

Микросхемы КР1820ВГ1 и КР1820ВП1 — периферийные устройства однокристалльных ЭВМ для управления жидкокристаллическим индикатором в режиме мультиплексирования по трем шинам и сохранения содержимого оперативной памяти при отключении питания

Малопотребляющее устройство обмена на основе универсального приемопередатчика содержит восемь последовательных каналов интерфейса для радиального подключения устройств ввода-вывода

Система СТАСИС для обработки данных с использованием статистических методов на IBM-подобных ПК

Однокристалльный контроллер дисплея и клавиатуры К1809ВТ3 заменяет контроллеры ЭЛТ прерываний, прямого доступа к памяти, контроллер клавиатуры, ПЗУ знакогенератора и ряд схем малой и средней степени интеграции

Отладочный комплекс для разработки и отладки программного обеспечения устройств и систем на базе МК БИС серии КР580 по принципу прямой связи с ЭВМ верхнего уровня

Опыт внедрения крупных технологических комплексов создания программ и методов их выбора



Microsoft Works 2.0 для персональных компьютеров IBM PC и совместимых с ними сделает Вас сильнее!

Microsoft Works имеет все, что необходимо для ежедневной работы. Русский Works содержит базу данных, «электронные таблицы» для ведения бухгалтерского учета, текстовый процессор, комплекс для деловой графики и даже электронный коммуникационный модуль. Microsoft Works — мощный, легкий и приятный в управлении программный продукт.

Мы добавили к системе Works такие компоненты, как калькулятор на экране, будильник, который будет напоминать Вам о предстоящих важных встречах, и специальную систему управления файлами, которая поможет освоить компьютерные команды. Добавьте к этому привлекательный внешний вид Works, высококачественную документацию и встроенную справочную систему.

В базе данных Works Вы сможете хранить информацию о товарах, списки клиентов или списки служащих фирмы. С ее помощью Вы можете за секунды произвести сортировку или сделать выборку из тысячи записей по таким критериям, как имя, номер и т. п.

База данных поможет Вам без особого труда составлять детальные обзоры, включающие статистические выводы. Works сделает это для Вас автоматически.

С помощью Works Вы сможете создавать профессионально оформленные письма, докладные записки и доклады любого объема. Вы можете воспользоваться различными стилями и размерами шрифта для того, чтобы улучшить внешний вид документа. Вы можете использовать Works для создания единого бланка письма или рассылки тысячам клиентов. С помощью специальной функции «Предварительный Просмотр» Вы сможете экономить время и бумагу, взглянув на свой документ в готовом виде перед тем, как его напечатать.

Электронные таблицы, являющиеся компонентой Microsoft Works — это более чем 70 встроенных математических, финансовых и статистических функций и операторов. Эти электронные советчики помогут Вам быстро рассчитать бюджет, проследить за направлением поставок и быть в курсе размеров прибыли. Электронные таблицы Microsoft Works рассчитаны на 256 колонок и 4096 строк.

С помощью функции построения графиков Microsoft Works дает возможность представлять цифровую информацию в наиболее наглядной форме. Работая с Microsoft Works Вы можете выбирать из большого числа разнообразных типов графиков.

Microsoft Works оснащена модемом — новейшим средством связи, используемым профессиональными бизнесменами. Используя Works оснащенный модемом, приспособлением, подключаемым к обычной телефонной сети, Вы сможете без труда посылать нужные файлы Вашим клиентам или коллегам. Информация пересылается через весь Советский Союз или через весь мир.

Покупая Microsoft Works, Вы не только получаете иллюстрированное руководство для пользователя и краткое справочное руководство, но и становитесь обладателем полной справочной системы, доступной Вам в любой момент. Ответ на Ваш вопрос появится на экране дисплея, да еще с цветными примерами, чтобы было легче понять. И, конечно, же, на русском языке. Works содержит пояснения по более чем 200 разделам.

Если Вы хотите получить дополнительную информацию о том, какие возможности предлагает Microsoft Works, запомните наш адрес:

**115598, Москва, ул. Ягодная, 17. СП «Диалог»
тел. 329-45-33 и телефакс: (095) 329-47-11.**

ММП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 3 / 1990 МОСКВА

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

В. А. Вишняков, А. С. Папков — Архитектура процессора логического вывода для 16-разрядной ПЭВМ 2

Е. А. Иванюта, Н. И. Климович, В. А. Кособрухов — Микросхема KP1820BF1 для управления мультиплексным ЖК-индикатором 4

П. Д. Банников, А. В. Брибыльский, В. С. Сякерский, Т. Г. Черноусова, Н. И. Шух — Устройство последовательного обмена на основе универсального асинхронного приемопередатчика 7

Г. А. Галуев — Программируемая нейтронно-конвейерная SIMD-архитектура для параллельной обработки изображений 11

А. В. Ананьев — Переносимость программ при работе с матричными печатающими устройствами 15

В. В. Коробченко, А. П. Дианов, Н. Н. Щелкунов — Программное обеспечение для программаторов ПЛМ 16

Л. И. Брусиловский, В. В. Скляров, Л. М. Хейфец — Инструментальная система MICROPOWER/PASSAL для разработки автономных микропроцессорных систем 22

В. С. Горбатов, В. А. Илларионов, А. А. Малюк, А. Е. Савин — Пакет программ для моделирования сложных динамических объектов на ПЭВМ 24

В. Ю. Зенинский, В. А. Убогов — Восстановление удаленных файлов в ОС RSX-11 28

А. И. Паниди — Дополнительный интерпретатор командных строк в ОС ОС-PB 29

А. В. Панфилов — Графические аппаратные средства персональных компьютеров семейства IBM PC 32

М. А. Алексеевский, И. А. Ельник, Е. И. Розенштейн — Операционная система реального времени ДС-86 36

С. Г. Шмалько, А. В. Винников — Язык программирования FORTH-88R 39

И. Б. Кириченко — Инструментальная система для создания АРМ 40

Л. Г. Осовецкий, А. А. Штрик — Анализ опыта внедрения крупных технологических комплексов создания программ и методов их выбора 43

В. И. Джиган — Управляющая микроЭВМ на основе микропроцессорного комплекта БИС серии K588 46

А. А. Григорьев, А. И. Федосова — Принципы сопряжения микроЭВМ с параллельными каналами ввода-вывода 47

А. А. Урсаев, С. Л. Соложников, С. А. Тарасенко — Система для регистрации и экспресс-анализа сигналов 50

Система технического зрения СТЗ-2М 54

Д. С. Коврига, С. М. Гладченко — Микропроцессорная приставка для станков с ЧПУ на базе микроЭВМ «Электроника БК» 57

С. Е. Богомолов — Быстрое получение полутоновых изображений для машиностроительных САПР 58

С. И. Вдовин, С. В. Цыганков — Получение и редактирование эскизных чертежей 61

А. Н. Седов, О. Н. Молчанов — Устройство формирования шины типа 8080 62

Ю. А. Орестов, Н. Н. Ивонтьев — Четырехканальный АЦП 64

М. А. Иванов — Многоканальные формирователи сигнала 69

В. Г. Арихов, А. А. Ланнэ, В. И. Пенской, Е. В. Стацюра — Измеритель периода основного тона речевого сигнала на основе цифрового процессора КМ1813BE1 71

В. Ф. Зелтиньш, Л. П. Лобанов, И. В. Пивоваров, В. А. Терсков, Г. С. Тимофеев — Разрешение конфликтов при обращении микропроцессоров к общей памяти мульти-микропроцессорных вычислительных систем 73

Ю. В. Новиков — Функциональные модули контрольно-измерительной системы на базе микроЭВМ «Электроника 60» 75

А. А. Бондаренко, В. Ф. Скороходов — Устройство контроля уровня выходных сигналов микросхем памяти 77

А. Л. Ковалев, П. В. Креневич, С. Л. Лизенко — Система меню для персональной ЭВМ 79

В. А. Зимнович, В. А. Сухман — Организация пользовательского интерфейса на основе системы меню РТК-микро 80

В. Д. Циделко, Ю. В. Хохлов, А. М. Литвин, С. Г. Богославский — Обучение микропроцессорной технике специалистов невычислительных специальностей 85

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Машинная графика

Инструментальные средства

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Робототехника

САПР

Устройства ввода-вывода

Измерительные системы, системы контроля, тестирование

УЧЕБНЫЙ ЦЕНТР

УДК 681.3

В. А. Вишняков, А. С. Папков

АРХИТЕКТУРА ПРОЦЕССОРА ЛОГИЧЕСКОГО ВЫВОДА ДЛЯ 16-РАЗРЯДНОЙ ПЭВМ

В последние годы многие ведущие фирмы увеличили капиталовложения в развитие практических применений искусственного интеллекта и в первую очередь в построение экспертных систем [1]. Поскольку экспертные системы включают элементы логического вывода, то возникает необходимость поддерживать его основные процедуры не только программно, но и аппаратно. Последнее приобретает особую важность при создании экспертных систем реального времени, в которых скорость логических выводов может составлять сотни клипс (тысяч логических выводов в секунду). Большая часть экспертных систем имеет блоки логического вывода, построенные на основе декларативных языков Пролог, Лисп.

Пролог — одна из наиболее широко используемых реализаций логического вывода, включающая машинные процедуры унификации и бэктрекинга, позволяющая распараллеливать базовые операции [2]. Для повышения скорости выполнения пролог-программ предложено не интерпретировать текст исходной программы, а компилировать его в набор команд виртуальной машины [3], реализуемой специальной программой.

Дальнейшее повышение скорости выполнения пролог-программ связано с аппаратным выполнением набора команд виртуальной машины Уоррена. В рамках этого направления и разрабатываются пролог-машины [4—7].

Архитектура пролог-машин

Одна из первых пролог-машин PSI аппаратно реализует предикаты языка KLO, представляющего собой расширенный Пролог [4]. В основе ее архитектуры лежит организация четырех стеков: локального, глобального, управляющего и трейлового. Процессор PSI включает блоки управления и обработки, оперативную и кэш-память, преобразователь адресов. Обработывающий блок содержит АЛУ, две входные шины и выходную, связанную с блоком регистров. Обработываемые 40-разрядные операнды представляют собой однобайтовый тег и 32-разрядные данные. Микропрограммное устройство управления параллельно выбирает и обрабатывает 64-разрядные микро-

команды с горизонтальной организацией. Рабочий цикл блока управления составляет 200 нс. К центральному процессору через шину ввода-вывода подключен консольный. Общая скорость выполнения программ PSI-машиной составляет около 30 клипс.

Интересна разработка японских специалистов — РЕК-машина, состоящая из основного и РЕК-процессоров. Основной процессор поддерживает интерфейс с пользователем, процедуры ввода-вывода, процесс компиляции программ [5]. РЕК-процессор взаимодействует с основным через командные регистры и общую память. Идеология работы РЕК-процессора совпадает с работой PSI-машины и многих аналогичных разработок логических процессоров, поддерживающих языки декларативного программирования.

Анализ известных архитектур процессоров логического вывода [4—7] позволяет заключить, что прямое их назначение — обеспечить ускоренное выполнение операций логического вывода, занимающих основное время процедурного процессора. Для этого используются аппаратные конфигурации, позволяющие оптимизировать доступ к структурам данных, порождаемым в ходе выполнения пролог-программы. Известные системы поддержки декларативных языков программирования представляют собой законченные автоматы, расширяющие возможности серийно выпускаемых процессоров за счет введения новых достаточно объемных в аппаратном смысле устройств.

При выполнении пролог-программы основная память системы захватывается пролог-процессором или ее содержимое дублируется в специальном образом организованной памяти, что позволяет исполнять фрагменты пролог-программ автономно. Общим в организации пролог-процессоров является то, что они не используют возможностей резидентных процедурных процессоров HOST-машины, остающихся незагруженными во время выполнения пролог-программы.

Для наилучшего использования аппаратных ресурсов серийных ПЭВМ в рамках общепринятой технологии программирования и сравнительно простого перехода от системы процедурного к системе логического программирования предлагается аппаратно-программный комплекс, совместимый со средой IBM PC. Разработаны программы, обеспечивающие взаимодействие аппаратно-программных средств про-

цессора, расширена область непосредственно адресуемой процессором области системной памяти, реализована аппаратная поддержка организации и сопровождения обработки структур данных и аппаратно-программная поддержка процедуры унификации и указателей областей памяти пролог-системы (см. рисунок).

Резидентный процессор взаимодействует с логическим через внутреннюю шину МП К1810ВМ86. Шина адреса-данных проходит «сквозь» аппаратные устройства, обеспечивающие моделирование пролог-системы. Специализированное устройство (контроллер шины) преобразует мультиплексированную шину обоих процессоров в системную шину ПЭВМ с отдельными шинами адреса и данных. Устройство, обеспечивающее быстрый доступ к структурам данных логического процессора и одновременно расширение объема непосредственно адресуемой системной памяти, назовем контроллером памяти. При работе с HOST-процессором оно «пропускает» адресную и информационные шины, полностью сохраняя логику работы МП К1810ВМ86. Взаимодействуя с логическим процессором, контроллер обеспечивает быстрый доступ к структурам данных пролог-системы и становится в какой-то мере частью блока унификации, моделируемого аппаратно-программными методами. Блоки управления и обработки составляют собственно аппаратную часть процессора пролог-системы и моделируют взаимодействие и работу внутренних регистров логического процессора. Область микропрограммной памяти блока управления в моделирующей системе резервируется как область системной памяти, что обеспечивает эффективную коррекцию алгоритмов управления аппаратной частью модели. Кроме того, внутренние регистры пролог-системы (ее аппаратной части) отображаются на область системной памяти ПЭВМ, что позволяет управлять логикой взаимодействия регистров пролог-процессора и алгоритмами их взаимодействия. Такая организация дает возможность проследить трассу выполнения пролог-программы.

Содержимое регистров от (к) пролог-процессора передается через вектор прерывания, предназначенный для инициализирования необходимой части программного ядра поддержки модели в HOST-системе.

Программные средства, обеспечивающие нормальную работу модели пролог-системы, имеют следующие компоненты:

базовую систему устройств ввода-вывода, совместимую с операционной системой MS DOS;

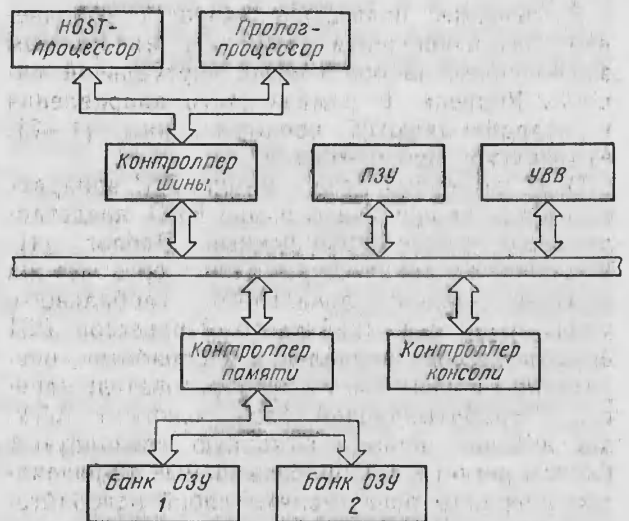
пакет программ для взаимодействия аппаратных средств процессора с ядром DOS;

пакет программ пролог-системы, адаптиро-

ванный к ПЭВМ, совместимым с IBM PC.

Контроллеры шины и памяти. Со стороны процессоров эти контроллеры получают мультиплексированную двунаправленную шину данных и адреса, слово состояния процессоров (SA0...SA2) и сигналы синхронизации доступа к системной шине ПЭВМ (RQ/GT0, RQ/GT1). Выходные сигналы контроллеров представляют собой сформированные шины адреса и данных, сигналы записи и чтения (RD/, WR/). Разрядность шины данных в зависимости от аппаратной конфигурации изменяется от 16 до 32, а шины адреса — от 22 (при 16-разрядной конфигурации) до 32 разрядов.

В качестве формирователей шин адреса и данных можно использовать серийные микросхемы формирователей шин К1810ИР82 и К1810ВВ86, позволяющие обеспечить простой переход от мультиплексируемой шины 16-разрядного МП К1810ВМ86 к системной шине ПЭВМ и 32-разрядной шине. Контроллер шины расширяет аппаратные возможности адресации МП К1810ВМ86, а его программная поддержка обеспечивает совместимость со структурами данных языка Пролог. Основным элементом контроллера памяти — регистр с разрядностью 16...32 бит (в зависимости от конкретного применения). Выходы регистра подключены ко входам регистра сдвига текущего значения данных на число разрядов, кратное четырем. Выходы регистра сдвига соединены с одной группой входов сумматора, другая группа — с выходами регистра сдвига следующего элемента. Покаскадное включение элементов контроллера памяти через сумматоры обеспечивает сколь угодно сложную адресацию любого элемента структуры данных пролог-программы наряду с



Структура модели пролог-системы

конвейерным вычислением адресов элементов структур данных [8].

Кроме аппаратной поддержки доступа к сложным структурам данных контроллер памяти организует доступ ЦП ПЭВМ к дополнительным банкам данных, вводимым в систему с помощью специально предусмотренного регистра адресации памяти. Необходимый банк данных устанавливается аппаратно-программным способом, для чего регистру адресации памяти присваивается определенный адрес. Совмещение сигнала обращения к банку данных с сигналом состояния процессора защищает данные в банке памяти от записи по адресу указателя стека и считывания ложных данных процессором при чтении команды.

Доступ к шине ПЭВМ синхронизируется сигналами управления локальной шиной ЦП ПЭВМ. При захвате шины ЦП блоки логического процессора находятся в режиме ожидания до тех пор, пока по внутренней магистрали блока процессоров не будет передан известный логическому процессору код префикса, указывающий на то, что следующая команда будет для него. После получения кода префикса логический процессор по локальной шине передает импульс, указывающий на его готовность работать с шинами адреса и данных ПЭВМ. ЦП генерирует ответный импульс, свидетельствующий об освобождении системных шин ПЭВМ. По окончании работы логический процессор по той же локальной шине генерирует еще один импульс, снова разрешающий ЦП доступ к системной шине ПЭВМ. Алгоритм

работы повторяется. Если необходим выход из пролог-программы по возникновению ошибки при выполнении с последующим обращением к интерпретатору пролог-системы, то логический процессор генерирует аппаратное прерывание, которое после передачи управления ЦП обеспечит обработку необходимой информации средствами интерпретатора пролог-системы.

Рассмотренные аппаратно-программные решения реализованы на ПЭВМ ЕС1840. Ядро логического процессора построено на базе МПК Ам29300. Скорость выполнения пролог-программы на модели достигает 105 клипс, что в 50 раз выше, чем без аппаратной поддержки. Телефон 39-89-17, Минск

ЛИТЕРАТУРА

1. Построение экспертных систем.— М.: Мир, 1987.— С. 441.
2. Логическое программирование.— М.: Мир, 1988.— с. 367.
3. Warren D. Implementing Prolog computing predicate logic programs.
4. Подробности проекта по компьютерам пятого поколения // Электроника.— 1984.— № 27.— С. 35—37.
5. Wada K., Miyatota M., Kio S. Intermediate code for sequantion Prolog mashine REK // Microprocessing and Microprogramming.— 1987.— № 21.— P. 275—282.
6. Компьютеры на СВИС.— М.: Мир, 1988.— Кн. 2.— С. 388.
7. Uchida S. Parallel Inferences Machines at ICOT // Future Generation Computer Systems.— 1987.— № 3.— P. 245—252.
8. Дементьев И. В., Вишняков В. А., Палков А. С. Аппаратные средства объектно-ориентированной, мультипроцессорной ПЭВМ // Обмен опытом в радиопромышленности.— 1988.— № 3.

Статья поступила 13.04.89

УДК 681.3.049

Е. А. Иванюта, Н. И. Климович, В. А. Кособрухов МИКРОСХЕМА КР1820ВГ1 ДЛЯ УПРАВЛЕНИЯ МУЛЬТИПЛЕКСНЫМ ЖК-ИНДИКАТОРОМ

Микросхема КР1820ВГ1 используется для управления 36-знаковым ЖК-индикатором (ЖКИ) в режиме мультиплексирования по трем шинам и в качестве периферийного устройства 4-рядных ОЭВМ серии КР1820. Допускается применение для работы с другими устройствами при условии соблюдения требований к эксплуатации.

Микросхема изготавливается по КМОП-технологии, выпускается в 20-выводном пластмассовом DIP-корпусе. Условное графическое обозначение приведено на рис. 1, назначение выводов — в табл. 1, электрические характеристики — в табл. 2, структурная схема — на рис. 2. На рис. 3, 4 показано типовое включение микросхемы.

Микросхема содержит встроенный генератор прямоугольных импульсов, ре-

зистивный делитель напряжения и делители частоты, с помощью которых формируются сигналы управления строками (общими электродами) и столбцами (сегментными электродами) ЖКИ в режиме 3-уровневого мультиплексирования (рис. 5). Одна микросхема имеет три выхода управления строками и 12 выходов управления столбцами. Предусмотрена возможность каскадирования схем, что позволяет использовать их для управления мультиплексным ЖКИ с числом знаков более 36.

Режимы работы микросхемы

Микросхема КР1820ВГ1 имеет четыре режима работы: одиночный, старший, младший, тестовый. В одиночном режиме одна микросхема управляет ЖКИ, емкостью 36 знаков, обеспечивая полную синхронизацию его работы. Стар-

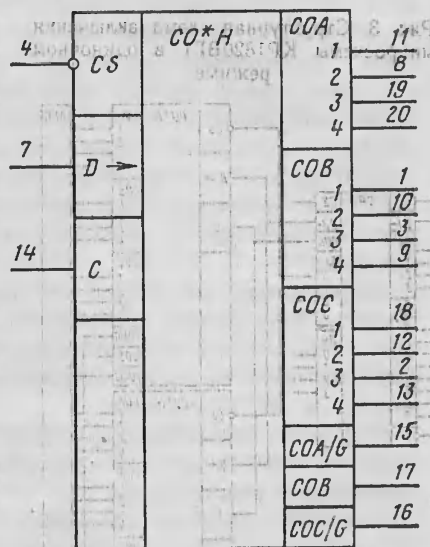
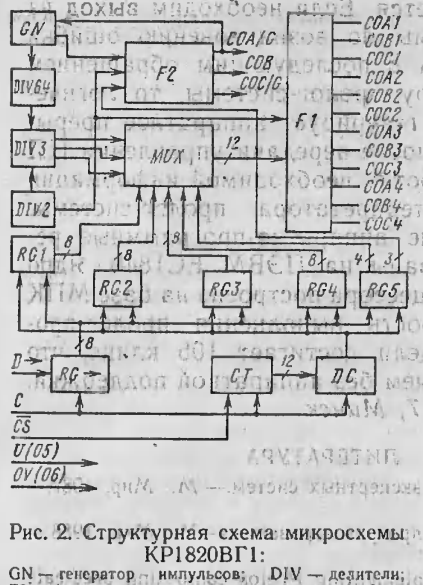


Рис. 1. Условное графическое обозначение микросхемы КР1820ВГ1

Назначение выводов микросхемы КР1820ВГ1



Вывод	Обозначение	Тип	Назначение
1...3	COB1, COB3, COB3	Выход	Управление столбцами В1, С3, В3
4	CS	Вход	Выбор кристалла
5	U _{CC}	—	Напряжение источников питания
6	GND	—	Общий
7	D	Вход	Данные
8...13	COA2, COB4, COB2, COA1, COC2, COC4	Выход	Управление столбцами А2, В4, В2, А1, С2, С4
14	C	Вход	Тактовый сигнал С
15	COA/G	Выход	Управление строкой А (вход генератора G)
16	COC/G	»	Управление строкой С (выход генератора G)
17	COB	»	Управление строкой В
18...20	COC1, COA3, COA4	»	Управление столбцами В1, А3, А4

Рис. 2. Структурная схема микросхемы КР1820ВГ1:

GN — генератор импульсов; DIV — делители; F2 — формирователь сигналов управления общими электродами ЖКИ; F1 — формирователь сигналов управления столбцами ЖКИ; MUX — мультиплексор «12 из 36»; RG1...RG4 — 8-разрядные регистры; RG5 — 7-разрядный регистр; RG — регистр сдвига; CT — счетчик импульсов; DC — децифратор.

Таблица 2

Основные электрические параметры микросхемы КР1820ВГ1

Параметр, единица измерения	Величина		
	не менее	не более	
1	2	3	4
Напряжение источника питания, В	U _{CC}	3 В ± 5 %	5 В ± 10 %
Диапазон температур, °С	T	-45	85
Тактовая частота, кГц	F		500
Период сканирования, мс	T _{SCAN}	4,8	25,6
Потребляемый ток, мкА (при U _{CC} =5,5 В)	I _{CC}		
Входное напряжение низкого уровня, В	U _{IL}	0	0,8
Входное напряжение высокого уровня, В (по выводу 15 в режиме «вход Gi»)	U _{IH1}	0,7U _{CC}	U _{CC}
Выходное напряжение высокого уровня (выбранный элемент ЖКИ), В	U _{OH1}	U _{CC} - 0,05U _{CC}	U _{CC}
Выходное напряжение низкого уровня (выбранный элемент ЖКИ), В	U _{OL1}	0	0,05U _{CC}
Выходное напряжение высокого уровня (невыбранный элемент ЖКИ), В	U _{OH2}	2/3U _{CC} - 0,05U _{CC}	2/3U _{CC} + 0,05U _{CC}
Выходное напряжение низкого уровня (невыбранный элемент ЖКИ), В	U _{OL2}	1/3U _{CC} - 0,05U _{CC}	1/3U _{CC} + 0,05U _{CC}
Входное напряжение высокого уровня, В (по выводам 4, 7, 14)	U _{IH2}	0,7U _{CC}	5,5

Рис. 3. Структурная схема включения микросхемы КР1820ВГ1 в одиночном режиме

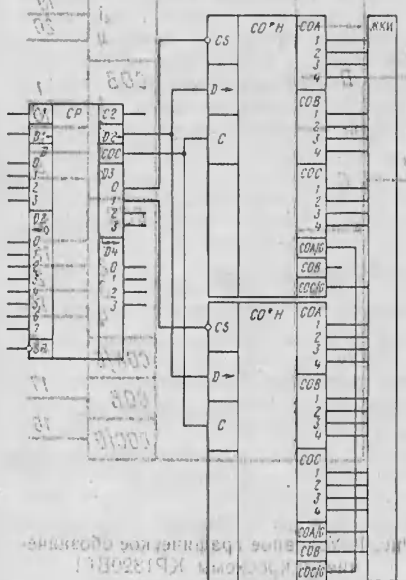


Рис. 4. Структурная схема включения микросхем КР1820ВГ1 в режимах старший (внизу) и младший (вверху)

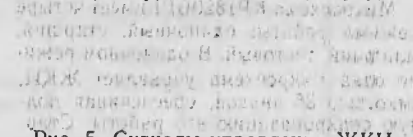
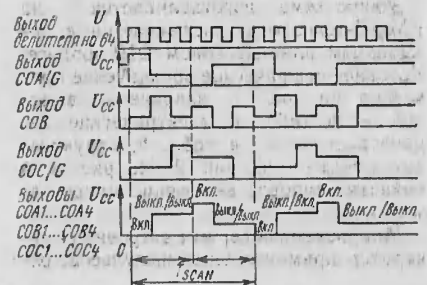


Рис. 5. Сигналы управления ЖКИ



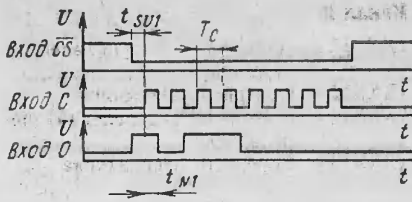


Рис. 6. Временная диаграмма работы микросхемы КР1820ВГ1:

$T_c \geq 2 \text{ мкс}$, $t_{h2} \geq 100 \text{ нс}$, $t_{su1} \geq 1 \text{ мкс}$,
 $t_{wh} \geq 0,8 \text{ мкс}$, $t_{N1} \geq 1 \text{ мкс}$, $4,8 \text{ мкс} \leq T_{SCAN} \leq 25,6 \text{ мс}$, $t_{su2} \geq 0,5 \text{ мкс}$

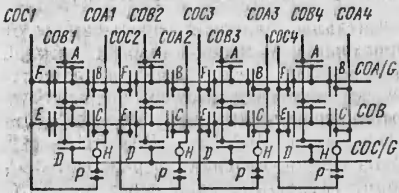


Рис. 7. Типовое расположение сегментов 4-разрядного ЖКИ и подключение к ним выводов управления строками и столбцами микросхемы КР1820ВГ1

ший и младший режимы предназначены для организации управления ЖКИ емкостью более 36 знаков, тестовый режим — для контроля качества микросхем в процессе изготовления.

Данные вводятся в микросхему в последовательном коде по входу D с синхронизацией записи фронтом тактовых импульсов по входу С (рис. 6). Код записываемых данных определяется конкретной схемой подключения шин управления строками и столбцами к сегментам ЖКИ (рис. 7), а также конфигурацией ЖКИ. Порядок следования битов в кодовой послылке для этого варианта подключения представлен в табл. 3.

Каждые восемь битов данных с номерами 1...32 имеют следующую структуру: А | В | С | D | E | F | G | Н. Первый бит соответствует сегменту Н первого разряда, восьмой — сегменту А первого разряда, девятый — сегменту Н второго разряда, 16-й — сегменту А второго разряда и т. д. Биты с 17-го по 24-й соответствуют сегментам цифры третьего разряда, с 25-го по 32-й — сегментам цифры четвертого разряда.

Биты с 33-го по 40-й имеют структуру: Q8 | Q7 | Q6 | X | P4 | P3 | P1 | P1. Данные, соответствующие битам 33...36, предназначены для управления специальными сегментами P1...P4. 37-й бит может принимать любое значение. 38-й и 39-й биты (Q6 и Q7) управляют режимом работы схемы согласно табл. 4. 40-й бит (Q8) предназначен для синхронизации работы двух и более микросхем при каскадировании.

Последовательность загрузки микросхемы в одиночном режиме. Для установления режима выполняется после-

Таблица 3

Порядок следования битов в кодовой послылке		
Номер бита	Вывод	Сегмент цифрового дисплея
1	COA1, СОС/Г	Н
2	COB1, COB	Г
3	COС1, COA/Г	F
4	COС1, COB	E
5	COB1, СОС/Г	D
6	COA1, COB	C
7	COA1, COA/Г	B
8	COB1, COA/Г	A
9	COA2, СОС/Г	Н
10	COB2, COB	Г
11	COС2, COA/Г	F
12	COС2, COB	E
13	COB2, СОС/Г	D
14	COA2, COB	C
15	COA2, COA/Г	B
16	COB2, COA/Г	A
17	COA3, СОС/Г	Н
18	COB3, COB	Г
19	COС3, COA/Г	F
20	COС3, COB	E
21	COB3, СОС/Г	D
22	COA3, COB	C
23	COA3, COA/Г	B
24	COB3, COA/Г	A
25	COA4, СОС/Г	Н
26	COB4, COB	Г
27	COС4, COA/Г	F
28	COС4, COB	E
29	COB4, СОС/Г	D
30	COA4, COB	C
31	COA4, COA/Г	B
32	COB4, COA/Г	A
33	COС1, СОС/Г	P1
34	COС2, СОС/Г	P2
35	COС3, СОС/Г	P3
36	COС4, СОС/Г	P4
37	Не используется	
38	Q6	Биты управления
39	Q7	
40	Q8	

довательность действий:
 установить на входе CS уровень Лог. 0,
 записать восемь битов данных для цифр первого-четвертого разрядов;
 записать четыре бита для специальных сегментов и четыре бита управления: 0 | 0 | 1 | 1 | P4 | P3 | P2 | P1;
 установить на входе CS уровень Лог. 1.

После установки микросхемы в нужный режим для последующей смены данных необязательно записывать все 40 бит информации. Перед проведением каждого нового цикла записи необходимо сбросить внутренние счетчики схемы. Для этого на входе CS должен появиться перепад уровней напряжения, т. е. произойти переход из состояния Лог. 1 в состояние Лог. 0.

Последовательность загрузки микросхем в старшем и младшем режимах. Данные записываются в следующем порядке:

1. Установить на входе CS уровень Лог. 0.
2. Записать 32 бит данных для «младшей» схемы
3. Записать четыре бита для специальных сегментов младшей схемы и биты управления: 1 | 1 | 1 | 1 | P4 | P3 | P2 | P1.

При подаче последней единицы обе микросхемы устанавливаются в младший режим, выходы COA/G обеих схем работают как выходы генератора. Происходит синхронизация работы микросхем.

4. Установить на входах CS обеих схем уровень Лог. 1.

5. Установить на входе CS, «старшей» схемы уровень Лог. 0.

6. Записать 32 бит данных для четырех цифр старшей схемы.

7. Записать четыре бита для специальных сегментов старшей схемы и биты управления: 0 | 0 | 0 | 0 | P4 | P3 | P2 | P1.

После этого вывод, COA/G старшей схемы начинает работать как выход управления строкой А, а вывод СОС/Г — как выход встроенного генератора. Импульсы с выхода генератора старшей схемы поступают на вход генератора COA/G младшей схемы, и оба кри-

Таблица 4

Состояние битов управления микросхемой

Бит		Режим работы	Вывод	
39 (Q7)	38 (Q6)		СОС/Г	COA/G
1	1	Младший	Выход управления строкой С	Вход генератора
0	1	Одиночный	То же	Выход управления строкой А
1	0	Тестовый	Выход внутреннего генератора	Выход управления строкой А
0	0	Старший		

сталла начинают работать синхронно от генератора старшей схемы.

8. На шходе CS установить уровень Лог. 1.

Чтобы записать во внутренние регистры-зашелки новые данные, нет необходимости сбрасывать обе схемы: достаточно записать данные по очереди во

внутренние регистры-зашелки каждой схемы. При этом в последний 40-й бит должен записываться ноль как для старшей, так и для младшей схем. Телефон 77-68-53, Минск

Статья поступила 13.04.89

УДК 681.325

П. Д. Банников, А. В. Брибыльский, В. С. Сякерский, Т. Г. Черноусова, Н. И. Шух

УСТРОЙСТВО ПОСЛЕДОВАТЕЛЬНОГО ОБМЕНА НА ОСНОВЕ УНИВЕРСАЛЬНОГО АСИНХРОННОГО ПРИЕМОПЕРЕДАТЧИКА

Малопотребляющее устройство, содержащее восемь последовательных каналов интерфейса для радиального подключения устройств ввода-вывода с последовательной передачей информации (ИРПС), разработано с использованием БИС универсального асинхронного приемопередатчика (УАПП) КР1818ВА19 и БИС серии К588. Устройство выполнено в виде кассеты GU3 европейского стандарта (220××233,4×20 мм) и предназначено для применения в системах управления технологическим оборудованием, работающих в составе гибких производственных систем и в системах с жесткими требованиями к энергопотреблению. Максимальный потребляемый устройством ток от источника питания 5 В — не более 85 мА, от источника питания 12 В — не более 160 мА. Малое значение тока потребления обеспечивается КМОП элементной базой. Устройство имеет интерфейс, совместимый по сигналам с каналом микроЭВМ «Электроника 60» (рис. 1).

Магистральные приемопередатчики (МПП) обеспечивают развязку сигналов устройства от канала ЭВМ, селектор адреса (СА) позволяет выбрать восемь групп адресов по четыре адреса в каждой группе и сформировать сигналы CS, WR и RD на БИС УАПП. Схема включения БИС СА отличается от типовой схемы [1]. На входы DA01...DA03 подаются из канала сигнала DA03...DA05, что позволяет получить восемь сигналов CS для каждой БИС УАПП. Адреса с шин AD01, AD02 по отрицательному фронту сигнала SYNC из канала запоминаются в адресном триггере и подаются параллельно на все восемь БИС УАПП. Разряд адреса с шины ADO не используется. Схема выбора адреса позволяет с помощью малогабаритных выключателей задавать положение адресов устройства в области внешних устройств ЭВМ. Адреса регистров устройства принимают следующие значения в диапазоне 160000...177776 (система счисления при указании адресов, адресов векторов и битов регистров состояния и буферных — восьмеричная):

- Канал 1**
- 1XXX00 — состояния (управления) приемника
 - 1XXX02 — буферный приемника
 - 1XXX04 — состояния (управления) передатчика
 - 1XXX06 — буферный передатчика

- Канал 2**
- 1XXX10 — состояния (управления) приемника
 - 1XXX12 — буферный приемника
 - 1XXX14 — состояния (управления) передатчика
 - 1XXX16 — буферный передатчика

- Канал 8**
- 1XXX70 — состояния (управления) приемника
 - 1XXX72 — буферный приемника
 - 1XXX74 — состояния (управления) передатчика
 - 1XXX76 — буферный передатчика.

Сигналы запросов прерывания от приемников и передатчиков с каждой БИС УАПП через инвертор подаются на свой контроллер прерывания, выполненный на БИС контроллера прерываний (КПРВ) К588ВН1 [2]. Цепь предоставления прерывания проходит последовательно через все БИС КПРВ, устанавливая тем самым физи-

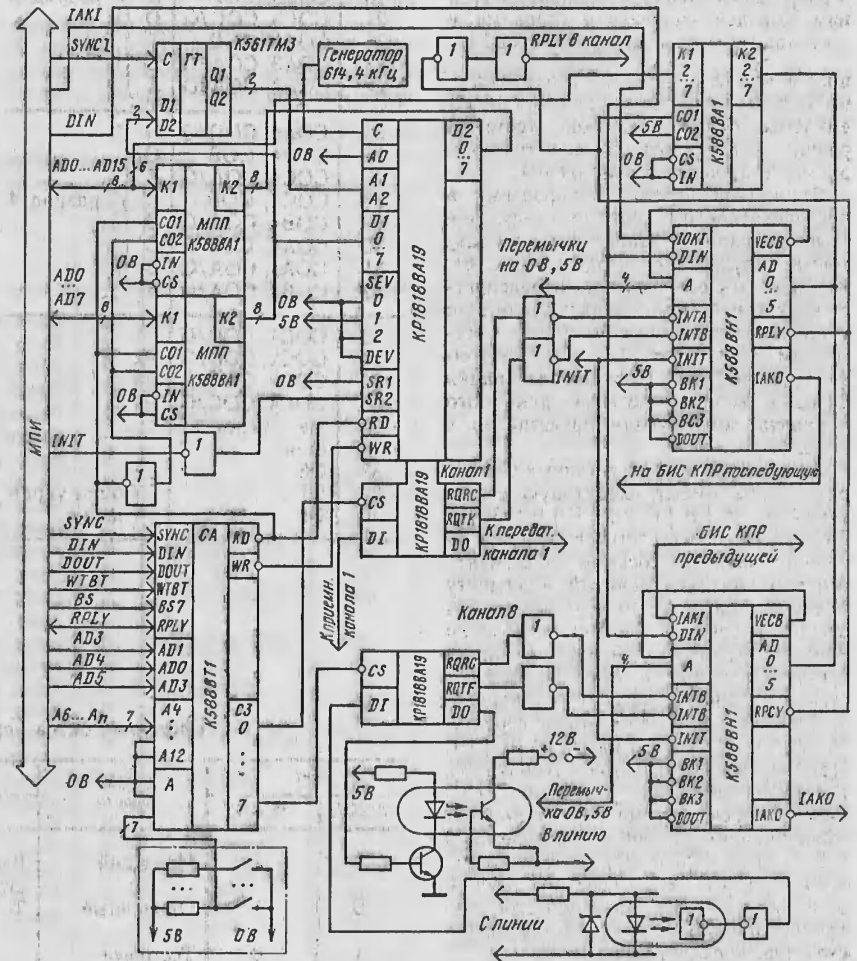


Рис. 1. Структурная схема устройства последовательного обмена

ческий приоритет каналов с меньшими адресами. Адрес вектора каждого канала задается с помощью перемычек на 0 или 5 В по входам, а БИС КПРВ для каждого канала — произвольным адресом в пределах 0..374 (для приемника ХХ0, для передатчика ХХ4). Генератор тактовой частоты, стабилизированный кварцем, вырабатывает меандр с частотой следования импульсов 614,4 кГц, которые подаются на входы БИС УАПП. Для каждого последовательного канала предусмотрены свои приемник и передатчик, обеспечивающие гальваническую развязку сигналов линии от сигналов устройства и преобразующие сигналы БИС УАПП в токовые сигналы и наоборот.

По сигналу INIT из канала БИС УАПП и КПРВ приводятся в исходное состояние. Далее ЦП загружает регистры состояния-управления приемников и передатчиков в цикле «выход». БИС СА формирует сигналы выборки CS на БИС УАПП и сигналы WR, RD. Адреса A1 и A2 по отрицательному фронту сигнала SYNG записываются в триггер и подаются на БИС УАПП, определяя конкретный регистр, к которому происходит обращение. При отсутствии посылки ток в линии соответствует Лог. 1 и находится в пределах 16..20 мА. БИС УАПП ожидает появления стартового импульса. При наличии посылки приемник каждого канала преобразует ее в последовательность импульсов ТТЛ-уровня, БИС УАПП выделяет стартовый импульс и принимает посылку. Если при установке режима для данного канала в регистре состояния-управления приемника RG SA/CORC запрещено прерывание, то ЦП определяет окончание приема по биту готовности (БИТ 200) в регистре RG SA/CORC и прочитывает байт информации из регистра данных приемника. Если в регистре RG SA/CORC разрешено прерывание от приемника, то по окончании приема посылки формируется сигнал требования прерывания от приемника, который посредством БИС КПРВ сообщает процессору о приходе посылки сигналом IRQ. Процессор при наличии разрешения на обработку внешних прерываний выдает на устройство сигналы IAKI, DIN и вводит адрес вектора прерывания, по коду которого определяется канал прихода посылки.

По готовности передатчика (БИТ 200 регистра RG SA/COTF равен единице) процессор выводит данные в регистр данных передатчика. БИС УАПП преобразует параллельный код в последовательный с ТТЛ-уровнями, а передатчик — в токовые посылки, причем в паузе между посылками и при передаче единиц в линии ток равен 16..20 мА (выходной транзистор открыт), а при передаче нулей — 0..3 мА (транзистор передатчика закрыт). Если в регистре RG SA/COTF не разрешено прерывание от передатчика (БИТ 100 равен нулю), то об окончании передачи посылки, точнее об освобождении

буферного регистра передатчика, процессор узнает по наличию в RG SA/COTF готовности передатчика (БИТ 200 равен единице). Если прерывание передатчику разрешено (БИТ 100 равен единице), то одновременно с готовностью передатчика БИС УАПП формирует сигнал запроса прерывания от передатчика, а БИС КПРВ устанавливает в канале ЭВМ сигнал IRQ. Дополнительно БИС КПРВ формирует сигнал VECB, и адрес вектора будет оканчиваться на четыре. Если при приеме посылки БИС УАПП идентифицировала ошибку, то ее признак можно прочитать из буферного регистра приемника, соответствующего канала.

Параметры и функции устройства определяются в основном БИС КР1818ВА19, выполненной по КМОП-технологии. УАПП принимает данные от ЦП в параллельном формате и выдает их в ВУ в последовательном формате; принимает последовательный поток данных от ВУ и передает его в параллельной форме в ЦП; вырабатывает сигналы, обеспечивающие прием и передачу; управляет скоростью передачи информации.

Условное графическое обозначение БИС КР1818ВА19 приведено на рис. 2. Обозначение и функциональное назначение выводов микросхемы представле-

4	0	D1	KP1818BA19	D2	0	12
5	1				1	13
6	2				2	14
7	3				3	15
8	4				4	16
9	5				5	17
10	6				6	18
11	7				7	19
21	A0			RQINARC		26
22	A1			RQINRTF		29
23	A2			RQINRHT		33
2	CS					
7	RD					
3	WR				DO	30
24	SR1					
39	SR2					
27	DI					
31	SEVO					
37	SEV1			INRG1		34
38	SEV2			INRG2		35
28	DEV			INRG3		36
32	C			INRG4		25
20	GND			U _{cc}		40

Рис. 2. Условное графическое обозначение БИС КР1818ВА19

Таблица 1

Назначение выводов БИС КР1818ВА19

Вывод	Обозначение	Рабочий уровень	Назначение
1	RD	H	Вход управления чтением
2	CS	H	Вход разрешения чтения или записи
3	WR	H	Вход управления записью
4..11	D1.O...D1.7	V	Двунаправленная магистраль данных
12..19	D2.0...D2.7	V	Выходная магистраль данных
20	OB	—	Общий вывод
21	AO	V	Вход адреса байта регистра
22, 23	A1, A2	V	Входы адресов регистров
24	SR1	V	Вход инициализации
25*	INRG4	V	Вход-выход делителя входной частоты на 768
26*	RQ INR RC	V	Вход-выход запроса прерывания приемником
27	DI	V	Вход последовательных данных
28	DEV	H	Вход запрета программирования скорости передачи
29*	RQ INR TF	V	Вход-выход запроса прерывания передатчиком
30	DO	V	Вход последовательных данных
31	SEVO	H	Вход выбора скорости передачи
32	C	V	Вход синхронизации
33*	PQ INP HLT	V	Вход-выход запроса прерывания по приему «пустого» слова
34	INRG1	V	Выход частоты синхронизации, с которой ведется обмен
35*	INRG2	V	Вход-выход делителя входной частоты на 12288
36*	INRG3	V	Вход-выход делителя входной частоты на 10240
37, 38	SEV1, SEV2	H	Входы выбора скорости передачи
39	SR2	V	Вход общего сброса
40	U _{cc}	V	Вход питания 5 В

* Выходы 25, 26, 29, 33, 35, 36 могут быть сброшены из состояния Лог. 1 в состояние Лог. 0 подачей извне в течение не менее 100 нс низкого уровня на соответствующие входы: H — низкий, V — высокий уровни.

но в табл. 1, структурная схема — на рис. 3. БИС УАПП включает следующие основные блоки: приемник RC, передатчик TF; схемы выбора скорости передачи данных SEV; блок управления приемопередатчиком COU; буфер шины данных BF.

Приемник RC принимает 8-битовое слово данных в последовательном формате, преобразует его в параллельную форму и загружает ЦП. В приемнике предусмотрены аппаратные средства для идентификации стартового и стопового битов, что позволяет определять границы начала и конца передаваемого слова данных. В качестве стартового бита принято значение Лог. 0 (низкий уровень), стопового — Лог. 1 (высокий уровень).

В состав приемника RC входят 16-разрядный регистр состояния-управления RG SA/CO RC; 16-разрядный буферный регистр RG BF RC; 10-разрядный сдвиговый регистр RG; схема управления приемником CO RC.

Регистр состояния-управления приемником RG SA/CO RC, формат которого представлен в табл. 2, содержит информацию, необходимую ЦП для организации обмена.

Триггер RC DONE (данные приняты), устанавливается после завершения передачи бита из сдвигового регистра приемника RG в буферный регистр RG BF RC. По окончании чтения содержимого буферного регистра на шину D1.0...D1.7, триггер RC DONE сбрасывается. Триггер RC ACT (активность приемника) находится в состоянии Лог. 1, если идет прием информации по последовательному каналу. В противном случае триггер сброшен. Установленный триггер RC IE (разрешение прерывания, приемником) — разрешает выдачу содержимого триггера RC DONE на выход RQ INR RC. Если триггер RC IE сброшен, то на выход RQ INR RC выдается низкий уровень. Триггер RC

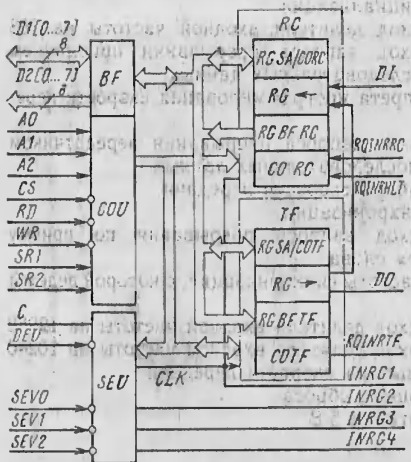


Рис. 3. Структурная схема БИС KPI818BA19

IE устанавливается и сбрасывается программно.

В младшем байте буферного регистра RG BF RC (табл. 2) хранится последнее принятое слово данных. В старший байт записывается информация о наличии ошибок ERR при передаче: кадрирования FR ERR, наложения OR ERR, а также информация о приеме «пустого» слова данных RC BRK. Ошибки кадрирования возникают, если приемник неправильно идентифицирует стартовый или стоповый бит. Ошибки наложения показывают, что на байт, находящийся в буферном регистре приемника, наложен байт данных из сдвигового регистра RG до того, как байт из буферного регистра был считан процессором. В этом случае предыдущее слово данных теряется. Триггер ERR (ошибка) устанавливается, когда обнаруживается ошибка кадрирования или наложения, и сбрасывается при устранении причин, вызвавших ошибку (рис. 4).

Триггер RC BRK — останов приема «пустой» посылки данных. Установка RC BRK в единицу сигнализирует о том, что приемник принял последова-

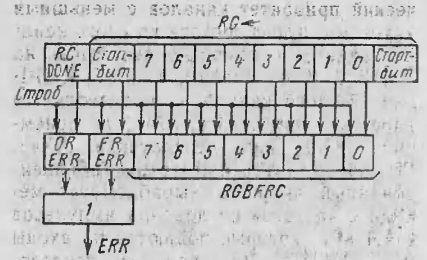


Рис. 4. Формирование сигналов ошибок

тельно не менее 11 нулевых битов и в передаче информации не участвует. RC BRK сбрасывается при переходе приемника в режим передачи.

После идентификации стартового бита в сдвиговый регистр RG по магистрали D1 последовательно загружаются стартовый бит, биты данных (младшими битами вперед) и стоповый бит. Наличие в приемнике двух регистров (сдвигового и буферного) обеспечивает прием данных, исключая при этом потерю информации. Пока принимается новое слово данных в сдвиговый регистр приемника, буфер-

Таблица 2
Формат регистров БИС KPI818BA19

Разряд регистра	Регистры							
	RG SA/CO RC		RG BF RC		RG SA/CO TF		RG BF TF	
	Обозначение триггеров	Функциональные возможности	Обозначение триггеров	Функциональные возможности	Обозначение триггеров	Функциональные возможности	Обозначение триггеров	Функциональные возможности
0	0	Чтение	RG BF RC	Чтение	TF BRK	Чтение-запись	RG BF TF	Чтение-запись
1	0				PBRE			
2	0				MAINT			
3	0				RBR0			
4	0				PBR1			
5	0	RBR2						
6	RC IE	Чтение — запись	0	0	0	0	0	0
7	RC DONE	0	0	0	0	0	0	0
8	0	Чтение	RC BRK	Чтение	TF RDY	Чтение	0	Чтение
9	0				0			
10	0				0			
11	RC ACT				0			
12	0				0			
13	0	FR ERR	0	0	0	0	0	0
14	0	OR ERR	0	0	0	0	0	0
15	0	ERR	0	0	0	0	0	0

* Не существующие триггеры читаются как нули

Коды выбора скорости передачи

Триггеры PBR			Внешние выходы SEV			Коэффициент деления тактовой частоты	Скорость передачи, Бод.
2	1	0	2	1	0		
H	H	H	B	B	B	128	300*
H	H	B	B	B	H	64	600
H	H	H	B	H	B	32	1 200
H	B	B	B	H	H	16	2 400
B	H	H	H	B	B	8	4 800
B	H	H	H	B	H	4	9 600
B	B	B	H	B	H	2	19 200
B	B	B	H	H	H	1	38 400

* Скорость передачи данных рассчитана для тактовой частоты 614,4 кГц.

ный регистр содержит ранее принятый байт данных, готовый к считыванию процессором. Если хранящиеся в буферном регистре данные не считаны процессором и поступает новое слово, то предыдущий байт данных теряется и формируется флаг ошибки наложения OR ERR.

Передачик TF является почти зеркальным отражением приемника — предназначен: для приема от процессора параллельного 8-разрядного слова данных; обрамления этого слова стартовым и стоповым битами; выдачи слова данных, стартового и стопового битов ВУ через последовательный канал передачи данных.

В состав передатчика TF входят 16-разрядный регистр состояния-управления передатчиком, RG SA/CO TF, 16-разрядный буферный регистр RG BF TF; 10-разрядный сдвиговый регистр RG; схема управления передатчиком CO TF.

Формат регистра RG SA/CO TF представлен в табл. 2. Младший байт регистра, за исключением бита TF RDY, устанавливается и сбрасывается программно.

Установленный триггер TF BRK запрещает выдачу данных на последовательный выход DO. Состояние выхода DO в этом случае не зависит от содержимого сдвигового регистра RG→, и на него выдается низкий уровень. Триггер PBRE определяет программный или аппаратный способ задания скорости передачи данных. Если триггер установлен, то скорость передачи задается программно содержимым триггеров PBR0...PBR2 или аппаратно внешними входами SEV0...SEV2 (табл. 3). Триггеры PBRE, PBR0...PBR2 сбрасываются внешним сигналом запрета скорости программирования передачи DEV. Триггер MAINT предназначен для самотестирования микросхемы. Если он установлен, что последовательный выход DO передатчика подсоединяется к последовательному входу приемника, отключая внешний вход DI (рис. 5). Триггер TF IE (разрешение прерывания передатчиком) управляет формированием сигнала запроса прерывания RQ INR TF (аналогично триггеру RC IE). При записи байта данных с шины DI в буферный регистр передатчика RG BF TF триггер TF RDY (готовность передатчика) сбрасывается и устанавливается, когда слово данных из буферного регистра RG BF TF передается в сдвиговый регистр RG→.

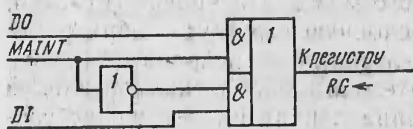


Рис. 5. Схема управления приемом информации

Таблица 4

Основные динамические параметры БИС КР1818ВА19

Параметр	Обозначение	Норма, нс
Время опережения информации, подаваемой на входы A0..A2, SC, относительно среза сигнала WR, не менее	t_{SU1}	100
Время опережения информации, подаваемой на входы D1.0...D1.7, относительно фронта сигнала WR, не менее	t_{SU2}	50
Время опережения информации, подаваемой на входы A0...A2, CS, относительно среза сигнала RD, не менее	t_{SUS}	50
Время удержания информации, подаваемой на входы A0...A2, CS, относительно фронта сигнала WR, не менее	t_{V1}	40
Время удержания информации, подаваемой на входы D1.0...D1.7, относительно фронта сигнала WR, не менее	t_{V2}	0
Время удержания информации, подаваемой на входы A0...A2, CS, относительно фронта сигнала RD, не менее	t_{V3}	0
Длительность низкой полки сигнала WR, не менее	t_{WL}	100
Время задержки распространения сигнала при переходе из состояния «Включено» в состояние высокого (низкого) уровня при управлении по входу RD, не более	t_{PZH} t_{PZL}	250

Буферный регистр передатчика RG BF TF, формат которого представлен в табл. 2, предназначен для временного хранения последнего слова данных, записанного в него с шины DI. Слово данных из буферного регистра, дополненное в передатчике стартовым и стоповым битами, загружается в сдвиговый регистр передатчика RG→. Содержимое сдвигового регистра RG→ последовательно выдается на выход DO.

Схемы управления приемником RC и передатчиком CO TF содержат счетчики импульсов с коэффициентами пересчета 10 и 16. Счетчики с коэффициентом 16 стробируют: в приемнике — прием информации в сдвиговый регистр RG— из канала DI; в передатчике — выдачу информации из сдвигового регистра RG→ в канал DO. Счетчики с коэффициентом 10 предназначены для подсчета числа при-

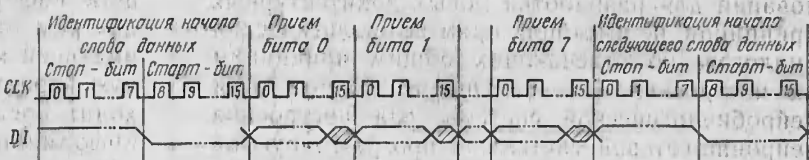


Рис. 6. Временная диаграмма приема информации по каналу DI

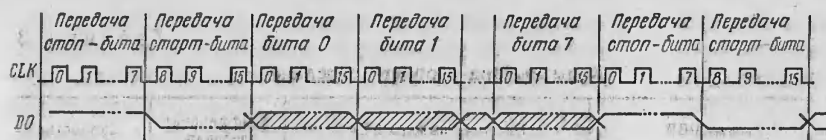


Рис. 7. Временная диаграмма передачи информации по каналу DO

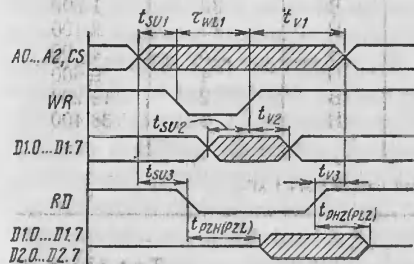


Рис. 8. Временная диаграмма чтения и записи информации по шинам D1 и D2

нятых или переданных битов данных. Блок выбора скорости передачи данных под управлением триггеров PBR0...PBR2 или внешних выводов SEV0...SEV2 делит входную частоту сигнала синхронизации f_C , поступающую на вход C, на один из коэффициентов деления, задаваемых в соответствии с табл. 3. Частота синхронизации приемника и передатчика определяется по формуле $f_{CLK} = f_C/D$, где D — коэффициент деления; скорость передачи информации $V_R = f_{CLK}/16$. Таким образом, входная частота f_C и скорость передачи информации V_R связаны соотношением $V_R = f_C/16$.

На внешний выход INR G1 выдается сигнал частотой f_{CLK} . Высокий уровень на входе SR2 вызывает опе-

рацию «Общий сброс», при которой выполняются следующие действия: инициализируются встроенные счетчики в приемнике и передатчике, регистры состояния-управления и буферные регистры; все выходы и входы-выходы микросхемы переключаются в высокоимпедантное состояние. После снятия сигнала SR2 (низкий уровень) микросхема УАПП работает в холостом режиме, пока пользователь программно не определит, какие действия она должна выполнять. Вход инициализации SR1 используется для начальной установки триггеров RC IE, TF IE, MAINT, и TF BRK.

Временные диаграммы, поясняющие работу БИС УАПП в режиме приема, приведены на рис. 6, в режиме чтения и записи информации — на рис. 8. Временные параметры представлены в табл. 4, основные электрические параметры в диапазоне рабочих температур $-10^\circ\text{C} \dots +70^\circ\text{C}$. Конструктивно БИС выполнена в пластмассовом 40-выводном корпусе 2123.40.5.

Телефон 77-28-51, 77-96-53, Минск

ЛИТЕРАТУРА

1. Бобков В. А., Чернуха Б. Н., Свиридович В. С., Ключников В. П. Расширенный микро-

Таблица 5
Основные электрические параметры БИС КР1818ВА19

Параметр	Норма
Напряжение питания U_{CC} , В	$5 \pm 5\%$
Входное напряжение низкого уровня U_{IL} , В, не более	0,8
Входное напряжение высокого уровня U_{IH} , В, не менее	$U_{CC} - 0,8$
Входной ток низкого уровня I_{IL} , мкА, не более	$ -20 $
Входной ток высокого уровня I_{IH} , мкА, не более	20
Выходное напряжение низкого уровня U_{OL} , В, не более	0,5
Выходной ток низкого уровня I_{OL} , мА, не более	4,0
Выходной ток высокого уровня I_{OH} , мА, не более	$ -0,2 $
Выходное напряжение высокого уровня U_{OH} , В, не менее	2,7
Потребляемый ток I_{CC} , мА, не более	1

процессорный комплект БИС серии К588 // Микропроцессорные средства и системы. — 1987. — № 1. — С. 6—7.

2. Свиридович В. С., Черноусова Т. Г., Чернуха Б. Н., Бобков В. А. Контроллер прерывания К588ВН1 // Микропроцессорные средства и системы. — 1987. — № 5. — С. 3—5.

Статья поступила 13.04.89

Г. А. Галуев

ПРОГРАММИРУЕМАЯ НЕЙРОННО-КОНВЕЙЕРНАЯ SIMD-АРХИТЕКТУРА ДЛЯ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Одно из многообещающих направлений в области развития параллельных систем обработки зрительной информации — изучение механизмов организации и функционирования нейронных структур мозга с целью их использования для разработки новых архитектурных принципов, не имеющих прямых биологических аналогов, но отвечающих общим принципам их построения [1—4]. Оценим возможности нейробиологической системы для построения нейронно-сетевой системы с программируемой нейронно-конвейерной SIMD-архитектурой для параллельной обработки изображений, обеспечивающей эффективную работу на всех уров-

нях и удовлетворяющей требованиям современной СБИС-технологии.

По современным нейрофизиологическим представлениям зрительный анализатор в общем виде состоит из сетчатки, зрительных нервов и зрительных центров головного мозга [5—7]. Оптическая система (глаз) создает изображение рассматриваемых предметов на сетчатке, в клетках которой обрабатываются световые сигналы. От клеток сетчатки берут свое начало зрительные волокна, объединяющиеся на выходе каждого глаза в зрительный нерв, передающий информацию от сетчатки к зрительным центрам головного мозга. На уровне сетчатки, имеющей многослойную структуру, образованную нервными клетками — нейронами, происходит предварительная обработка зрительной информации, направленная на улучшение изображения, формируемого фоторецепторами. Информация об этом изображении по зрительным нервам передается в нижние слои зритель-

ного центра для дальнейшей обработки. Согласно данным нейрофизиологических исследований [5, 7] эта обработка связана с выделением характерных локальных признаков изображения с помощью специальных нейронов — детекторов. По мере продвижения информации от нижних нейронных слоев зрительного центра к верхним выделяются все более сложные признаки изображения, соответствующие более высокому уровню абстракции. В верхних слоях зрительного центра по важнейшим информационным признакам, выделенным на предыдущих этапах обработки зрительной информации, распознается зрительный образ. Зрительный анализатор имеет четко выраженную многослойную иерархическую архитектуру, адекватную природной структуре процесса обработки зрительной информации. К **основным нейробионическим принципам** организации и функционирования зрительного анализатора, обеспечивающим эффективную реализацию процесса обработки зрительной информации, относятся следующие:

иерархии представлений, отвечающий за продвижение зрительной информации через множество различных уровней обработки от нижнего (входной сигнал) до верхнего (распознавание); параллельных вычислений, реализуемый на каждом уровне обработки зрительной информации. Этот принцип предполагает наличие на каждом уровне обработки некоторой нейронной сети, состоящей из множества взаимосвязанных нейронов и осуществляющей параллельные вычисления, соответствующие данному уровню обработки;

изменения эффективности синаптических связей между нейронами, лежащий в основе процессов обучения и функциональной перестройки нейронных сетей;

латерального взаимодействия, обеспечивающий селективные функции нейронов.

Эти принципы положены в основу цифровых нейроподобных процессоров (ЦНП), удовлетворяющих требованиям современной СВИСтехнологии и обеспечивающих возможность практической реализации нейронно-сетевых систем [8—11]. Методы построения таких систем на базе ЦНП для задач фильтрации изображений, выделения контуров и сегментации изображений, выделения важнейших локальных и глобальных признаков контурных изображений, распознавания зрительных образов рассмотрены в работах [12—15]. Нейронно-сетевую систему параллельной обработки изображений можно представить в виде иерархически организованной совокупности слоев ЦНП, каждый из которых связан с управляющей ЭВМ. Слой представляет собой сеть однородных, параллельных, взаимосвязанных ЦНП. Каждый ЦНП в цифровой форме реализует математи-

ческую модель динамического нейрона и позволяет воспроизвести основные функциональные свойства нервной клетки: процессы пространственного и временного суммирования входных сигналов, синаптическое взаимодействие, пороговость и генерацию выходных сигналов.

Для хранения основных параметров в ЦНП предусмотрена регистровая память. ЦНП можно перенастроить на режим функционирования формального нейрона, изменяя основные параметры [8, 9]. Функциональная пластичность ЦНП обеспечивает выполнение базисных арифметических и логических операций в процессе обработки изображений: свертки, скалярного произведения, численного интегрирования, матричных, векторных, логических, нелинейных. ЦНП обмениваются данными в последовательных кодах в фиксированные моменты времени, реализуя однотипные операции над множеством различных потоков данных. Иными словами, каждый слой ЦНП представляет собой полностью параллельный процессор с нейронной SIMD-архитектурой.

Структура связей между ЦНП внутри каждого слоя и между соседними слоями может устанавливаться жестко в соответствии с функциональным назначением данного слоя в процессе обработки изображений [14]. Однако часто возникает необходимость в изменении конфигурации синаптических связей между ЦНП в зависимости от алгоритма обработки. Для этого устанавливаются жесткие связи между ЦНП внутри слоя и между соседними слоями по типу каждый с каждым и задаются соответствующие матрицы синаптических коэффициентов [9]. Таким образом, в каждом слое ЦНП формируется структура связей, адекватная специфике выбранного алгоритма обработки информации.

Используя принцип модификации синаптических весов в верхних слоях ЦНП, предназначенных для реализации процедуры распознавания изображений, можно обеспечивать обучаемость системы распознаванию различных зрительных образов [15].

Архитектура системы программируется путем записи требуемых значений основных параметров ЦНП (в соответствующие регистры памяти) и задания структуры связей между ними [9]. Исходная информация об изображении вводится в память нижнего слоя ЦНП либо параллельно от матричного фотосенсорного устройства, либо параллельно-последовательно построчным считыванием элементов изображения с телевизионной камеры. В самом верхнем слое ЦНП в каждый фиксированный момент времени срабатывает один ЦНП, указывающий на принадлежность анализируемого изображения соответствующему классу [15]. По-

следнее обстоятельство, а также тот факт, что информация в процессе обработки не покидает систему до тех пор, пока не достигнет верхнего слоя ЦНП, позволяет снять проблему вывода больших массивных данных из системы и связанные с ней ограничения на производительность [1].

Движение информации в системе в вертикальном направлении от нижних слоев ЦНП к верхним позволяет организовать конвейерную обработку. Если каждый уровень обработки реализуется в одном слое ЦНП, то после заполнения конвейера время решения всей задачи обработки изображений для системы в целом будет определяться временем срабатывания одного ЦНП. Таким образом, архитектуру рассмотренной нейронно-сетевой системы можно определить как программируемую нейронно-конвейерную SIMD-архитектуру.

Однородность архитектуры позволяет выполнять ее на базе СВИС-технологии [1, 10, 11]. Например, можно разместить с помощью трехмерного монтажа микроэлектронные ЦНП в виде пачки кремниевых слоев [1]. Трехмерный монтаж по сравнению с двумерным значительно сокращает среднюю длину соединительных проводников, увеличивает плотность упаковки, снижает энергопотребление и по крайней мере на порядок уменьшает габариты системы.

347928; Таганрог, ГСП-284, Чехова, 2; НИИ МВС; тел. 6-25-75

ЛИТЕРАТУРА

1. СВИС для распознавания образов и обработки изображений / Пер. с англ.; Под ред. К. Фу.— М.: Мир, 1988, 248 с.
2. Соколов Е. Н., Шмелев Л. А. Нейробионика.— М.: Наука, 1983, 279 с.
3. Реальность и прогнозы искусственного интеллекта / Пер. с англ.; Под ред. Стефанюка В. Л.— М.: Мир, 1987, 247 с.

УДК 681.326-181.48

Б. И. Дубовик, А. А. Абакумов

УЧЕБНАЯ МИКРОЭВМ НА ОСНОВЕ КОНТРОЛЛЕРА «ЭЛЕКТРОНИКА МС2702»

Контроллер «Электроника МС2702» после доработки может быть использован в качестве лабораторного стенда для изучения работы микропроцессорных устройств и микроЭВМ. Для расширения доступа к внутренним элементам структуры контроллера создано устройство, обеспечивающее отображение состояния шин адреса, данных и регистра состояния процессора, а также возможность перехода на цикловый режим выполнения команд (на каждом шаге одного цикла текущей команды). Оно включает в себя шестизрядный семи-сегментный дисплей, четыре разряда которого

4. Каляев А. В., Галуев Г. А. и др. Современное состояние нейрокибернетических исследований и разработок в СССР и за рубежом / Деп. ВИНТИ, 05.05.88, № 3512, В 88.
5. Глезер В. Д. Зрение и мышление.— Л.: Наука, 1985.— 246 с.
6. Куффлер С., Николс Д. От нейрона к мозгу.— М.: Мир, 1979, 439 с.
7. Дудкин К. Н. Зрительное восприятие и память.— Л.: Наука, 1985, 208 с.
8. Каляев А. В., Чернухин Ю. В., Галуев Г. А. Адаптивные цифровые нейроноподобные элементы // Изв. Сев.-Кавказ. науч. центра высш. шк. Сер. Техн. науки.— 1980.— № 2.— С. 13—18.
9. Галуев Г. А. Цифровые нейроноподобные ансамбли и сети с перестраиваемой архитектурой // Многопроцессорные вычислительные структуры.— Таганрог: ТРТИ.— 1984.— № 6(XV).— С. 70—72.
10. Галуев Г. А., Ветер В. В. Способы реализации цифровых нейроноподобных элементов в интегральном исполнении // Многопроцессорные вычислительные структуры.— Таганрог: ТРТИ.— 1986.— № 8 (XVII).— С. 72—75.
11. Чернухин Ю. В., Брюхомицкий Ю. А., Галуев Г. А., Гайдученко Г. В. Принципы построения цифровых нейроноподобных структур на пластине // Деп. науч. работы.— М.— 1988.— № 2.— С. 12, 174 (Деп. 9.10.87, № 7179, В87).
12. Галуев Г. А. Параллельная обработка и распознавание образов на основе однородных цифровых нейроноподобных структур // Распараллеливание обработки информации: Тез. докл. VI Всесоюз. школы-семинара «РОИ-87».— Львов.— 1987.— С. 161—162.
13. Галуев Г. А., Брюхомицкий Ю. А. Параллельная обработка и анализ изображений на основе цифровых нейроноподобных структур // Математические методы распознавания образов (ММРО-Ш): Тез. докл. III Всесоюз. конф. Ч. II.— Львов.— 1987.— С. 62—63.
14. Галуев Г. А., Мильков Н. А. Обработка изображений на основе цифровых нейроноподобных сетей // Деп. науч. работы.— М., 1987.— № 2.— С. 18, № 304 (Деп. 22.10.86, № 7352, В86).
15. Галуев Г. А. Обучающаяся цифровая нейроноподобная система для распознавания образов // Изв. Сев.-Кавказ. науч. центра высш. шк. Сер. Техн. науки.— 1985.— № 3.— С. 89—92.

Статья поступила 22.08.88

КРАТКОЕ СООБЩЕНИЕ

используются для отображения в 16-ричном коде состояния шины адреса, два разряда — шины данных, точки индикаторов (сегменты h) — для вывода информации о процессоре в текущем цикле команды.

Конструктивно устройство выполнено в виде отдельного блока, который подсоединяется к ХС1-разъему контроллера. Полученное устройство — специализированный «тестер» для отыскания неисправностей. В этом случае необходимо использовать поцикловый режим работы и ПЗУ со специальными тестовыми программами.

300600, Тула, 26, пр. Ленина, 92, ТПИ; тел. 25-03-17

Статья поступила 27.06.88

УДК 681.3:519.85

В. Е. Кацман, В. Г. Малинов

ПАКЕТ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ЛП-МИКРО-2 ДЛЯ ПЭВМ

Пакет программ предназначен для решения транспортной задачи, линейного (ЛП) и квадратичного (КП) программирования. Все программы пакета написаны на языке БЕЙСИК версии 2.0 1985 г. для ПЭВМ «Искра 1030». БЕЙСИК-интерпретатор ПЭВМ совместим с интерпретатором БЕЙСИКа IBM PC. Рассматриваемая версия пакета содержит девять программ ЛП и КП, в том числе шесть программ с различными модификациями симплекс-метода ЛП, программы приближенного итеративного метода решения задач КП и транспортной задачи.

В программах IPSMA, LPSMVA, LP14A точных методов ЛП реализован простой, в программе LPDSMA — двойственный, в программах LPIBA и LPMGA — модифицированный симплекс-методы. С помощью программ LPSMA и LPIBA можно решать задачи с верхними ограничениями на аргументы путем расширения матрицы задачи, за исключением задач с ограничениями равенствами. В программе LPTRA решения транспортной задачи используется метод Хичкока, в программах LPMGA и LQPA, реализующих приближенный метод решения задач ЛП и КП, — метод сопряженных градиентов, во всех программах — диалоговый ввод данных задач.

Для программ LPIB и итеративных методов ЛП и КП применяется плотная запись матриц задач без нулевых элементов. В оперативной памяти (ОП) ПЭВМ «Искра 1030» для значений элементов входных, промежуточных, выходных массивов и идентификаторов отводится вся рабочая область БЕЙСИКа — 61301 байт. Тексты программ в ОП ПЭВМ хранятся вне зоны БЕЙСИКа.

После ввода данных задачи ЛП и КП решаются без вмешательства пользователя. Для программ точных методов ЛП предусмотрены четыре возможных окончания: 1) получено оптимальное решение; 2) система ограничений несовместна; 3) линейная форма неограничена; 4) превышено допустимое число итераций. Окончание по признаку 4 введено для предотвращения закликивания алгоритма.

Проблемные программы пакета могут быть загружены в ОП ПЭВМ как одноименные файлы при решении однотипных задач. По желанию пользователя выбор программы для счета и ее загрузка в ОП могут быть автоматизированы с помощью управляющей программы данного пакета. В ней для выбора проблемной программы при решении конкретной задачи используются следующие параметры: A1 — проблемной программы (0 — любой из точных алгоритмов ЛП, 1 — двойственный симплекс-метод, 2 — итеративный метод ЛП, 3 — итеративный метод КП, 4 — транспортная задача); A2 — вид постановки задачи ЛП (1 — задача канонического вида, 2 — задача канонического вида с ограничениями на аргументы сверху, 3 — основные ограничения задачи являются неравенствами); A3 — заполненность матрицы ограничений (0 — заполнена на 50 % и более, 1 — заполнена менее чем на 50 %). При A1 ≠ 0 значения параметров A2 и A3 могут быть произвольными и на выбор алгоритма не влияют.

Данные о решении тестовых задач одинаковой размерности для наиболее экономной по времени программы простого симплекс-метода LPSMA из данного пакета и про-

граммы LPSM для ПЭВМ «Искра 226» приведены в таблице. Для численных экспериментов задачи с квадратными

Результаты расчетов на ПЭВМ

Характеристика задачи	Плотность, %	«Искра 1030»				«Искра 226»	
		Простая точность		Двойная точность		T	K
Размерность матрицы		T	K	T	K	T	K
10×10	100	31	10	35	10	24	10
10×11	18,2	19	11	21	11	12	11
10×12	60	17	10	19	10	23,1	10
10×100	92	91	10	97	10	41,2	11
10×250	98,4	217	10	219	10	188	11
30×30	100	630	35	289	32	437	32
30×31	6,45	132	31	141	31	97	31
30×32	56,2	127	30	141	30	103	31
40×40	100	1343	45	1443	49	1013	44
40×41	4,9	227	41	242	41	161	41
40×42	54,8	219	40	244	40	171	41
60×61	3,21	493	61	519	61	332	61
70×71	2,81	665	71	705	71	444	71
70×72	53	648	70	720	70	474	71

Примечание. Плотность — содержание ненулевых элементов матрицы задачи ЛП в процентах; T — время решения задачи в секундах; K — число итераций (число пересчетов симплекс-таблицы)

матрицами генерировались с помощью датчика случайных чисел; задачи с неквадратными матрицами — целочисленные с детерминированными целочисленными коэффициентами.

Время решения задач при использовании БЕЙСИК-интерпретатора для ПЭВМ «Искра 1030» почти для всех задач больше, чем соответствующее время на ПЭВМ «Искра 226» с БЕЙСИК-2-системой.

Точность получения координат оптимального вектора наибольшая для ПЭВМ «Искра 1030» при счете с удвоенной точностью (вещественное число в ОП занимает 8 байт, сохраняется 17 значащих цифр). Например, для задачи 40×40 она составляет по сравнению с результатами для «Искры 226» соответственно 11 и 7 знаков после десятичной точки (вещественное число в ОП занимает 8 байт, сохраняется 13 значащих цифр). При счете на «Искре 1030» с простой точностью (вещественное число занимает четыре байта в ОП, сохраняется семь значащих цифр числа) погрешность в координатах оптимального вектора растет с увеличением размерности задачи, а точность их определения не превышает двух знаков в дробной части для задачи с матрицей 40×40.

Задачи наибольшей для программ пакета размерности, содержащие 110 основных ограничений и 112 переменных с полной записью матрицы в ОП (симплекс-таблица без искусственных переменных), позволяет решать программа двойственного симплекс-метода.

460352, ГСП Оренбург, пр-т Победы, 13, Политехнический ин-т, ОНИЛ АСУ; тел. 7-98-56, 7-97-40

Статья поступила 30.08.89

ПЕРЕНОСИМОСТЬ ПРОГРАММ ПРИ РАБОТЕ С МАТРИЧНЫМИ ПЕЧАТАЮЩИМИ УСТРОЙСТВАМИ

Большинство современных матричных печатающих устройств: D100 (СМ6325), ROBOTRON-6311 (СМ6329), EPSON FX-85, УВВПЧ-30 — позволяет отображать полный набор символов (русские, латинские, строчные, прописные), имеет несколько различных шрифтов (узкий, расширенный, жирный и т. п.) и дополнительные сервисные функции (подчеркивание, печать нижних (верхних) индексов, графический режим, псевдографика).

Одна из трудностей в работе с такими устройствами заключается в том, что прикладные программы (ПП) представляют данные в различных кодовых таблицах. В основном ПП оперируют кодовой таблицей КОИ-7 «Набор 2» (только прописные буквы), но многие редакторы текстов используют полный набор символов (КОИ-8 или КОИ-7 «Набор 0»/«Набор 1»). Если ПУ работает в КОИ-8, то печать информации в КОИ-7 с помощью «Набора 2» приведет к искаженному выводу символов (вместо прописных русских букв печатаются строчные латинские). В матричных ПУ используется несколько стандартных кодовых таблиц, сменяемых с помощью специальных переключателей или управляющих кодов. Но возможность работы с полным набором символов при переключении таблиц сохраняется не всегда. Поэтому, если аппаратно переключать ПУ в КОИ-7 «Набор 2», теряется возможность печати строчных букв, к тому же аппаратный способ смены таблиц не удобен в работе.

Таким образом, имеет смысл использовать ту кодовую таблицу, в которой можно работать с полным набором символов. Но и тогда различия между ПУ сохраняются. Например, D100 отображает полный набор символов В-битной кодовой таблицей (КОИ-8), в то время как ROBOTRON печатает с помощью двух 7-битных кодовых таблиц: одна для латинских (КОИ-7 «Набор 0»), другая для русских (КОИ-7 «Набор 1») букв, переключаемых специальными управляющими символами.

Управляющие символы и последовательности символов представляют собой еще одно препятствие при работе с матричными принтерами. Дело в том, что они для каждого типа ПУ свои, поэтому программы, использующие такие последовательности, могут работать только с тем ПУ, для которого они разработаны, что снижает их ценность.

Средством для преодоления подобных недостатков применительно к ОС RT-11 (РАФОС,

ОС ДВК) может служить универсальный драйвер печати, разработанный на вычислительном центре Московского института приборостроения, который можно использовать вместо стандартного системного драйвера печати (LP) RT-11. Драйвер работает с устройством любого типа и любыми ПП, позволяет сделать программы независимыми от управляющих последовательностей и типа ПУ. Это означает, что изменение аппаратных возможностей ПУ не требует изменений в ПП, т. е. программы корректно работают на любой модели ПУ.

При этом сам драйвер также не зависит от типа ПУ, т. е. аппаратные изменения ПУ не влекут за собой переассемблирование или замену на другой драйвер. Независимость обеспечивается благодаря преобразованию символов из кодовой таблицы ПП в кодовую таблицу ПУ и выдаваемых ПП управляющих последовательностей в управляющие последовательности конкретной модели ПУ. Драйверу при аппаратном изменении кодовой таблицы или замене ПУ с помощью SET-команды сообщается тип кодовой таблицы, в которой работает ПУ. Затем также SET-командой устанавливается тип кодовой таблицы, в которой должна выдаваться информация из ПП. Зная типы кодовых таблиц ПП и ПУ, не сложно выполнить преобразования, при которых вся информация будет печататься корректно.

Например, при работе на D100 пользователь вначале устанавливает в драйвере код этого ПУ — КОИ-8. Затем при необходимости напечатать текст, содержащий только прописные буквы (например, программу на БЕЙСИКе), он сообщает драйверу код выводимой информации (КОИ-7 «Набор 2»), которая после этого печатается. Если возникает потребность вывести на печать текст, набранный, к примеру, в редакторе EDIK, то надо изменить код выводимой информации на КОИ-7 «Набор 0» (печать также будет выполняться правильно).

Драйвер поддерживает все известные стандартные кодовые таблицы (КОИ-8, КОИ-7 «Набор 0», КОИ-7 «Набор 1», КОИ-7 «Набор 2») и специальную кодовую таблицу редактора МИКРОМИР. Управляющие последовательности преобразуются драйвером с помощью специальной таблицы замены, которая также задается SET-командами. В ней управляющим последовательностям, выдаваемым ПП, поставлены в соответствие последовательности ПУ. При печати драйвер «отлавливает» управляющие последовательности, идущие от ПП, и заменяет их соответствующими кодами из таблицы.

В драйвере реализован ограниченный алгоритм замены. Ограничение заключается в том, что управляющие последовательности, посту-

пающие из ПП, должны состоять из символа ESCAPE (восьмеричное 33) и произвольного символа (наиболее общий случай для ПУ). Если программа, созданная для одного типа ПУ, эксплуатируется на устройстве другого типа, необходимо ввести в драйвер таблицу, устанавливающую соответствие между их управляющими последовательностями. Это неудобно из-за необходимости менять таблицы при замене ПП и сложности представления в общем виде многообразия управляющих последовательностей разных моделей ПУ, которые варьируются от простых односимвольных кодов до четырех-пятибайтовых последовательностей с параметрами, и тем более невозможно добиться полной переносимости с помощью двухбайтовых ESCAPE-последовательностей, используемых в драйвере.

Избежать этих неудобств можно при выводе на печать из ПП не специальных управляющих последовательностей определенного ПУ, а некоторого промежуточного кода. Каждой такой промежуточной (виртуальной) управляющей последовательности легко поставить в соответствие коды конкретной модели ПУ. Необходимость смены таблицы при этом возникает лишь при замене ПУ. В качестве промежуточных последовательностей можно использовать код ESCAPE и некоторое число, записанное в одном байте.

Разрабатывая ПП с виртуальными последовательностями, программист имеет в виду, что каждая последовательность означает работу с определенной аппаратной возможностью ПУ (например, включение какого-либо шрифта). При этом, в драйвере виртуальной последовательности соответствует реальная управляющая последовательность. При переносе ПП на другую модель ПУ эта реальная последовательность изменяется в драйвере, но ПП при этом не меняется.

Если же возможность, подразумеваемая виртуальной последовательностью, в данной модели ПУ отсутствует, то при задании таблицы преобразования ей можно поставить в соответствие либо некоторую другую, похожую функцию (например, вместо шрифта двойной ширины — шрифт с фазовым смещением), либо просто заглушку (нулевой код), и тогда последовательность отрабатываться не будет.

В некоторых случаях полной независимости программ достичь не удается. Трудно применять последовательности с параметрами, которые устанавливают плотность печати в некоторых моделях ПУ. Отдельной, пока не решенной задачей является обеспечение на виртуальном уровне графического режима, аппаратная поддержка которого очень сильно изменяется в зависимости от модели ПУ (число иголок, установка ширины печати и т. п.).

Драйвер эксплуатируется с SJ-монитором RT-11 (ОС ДВК) на машинах типа ДВК и ПУ ROBOTRON, D100. Он поддерживает практически все команды, имеющиеся в стандартном системном драйвере печати LP RT-11. Кроме того, драйвер обладает дополнительными возможностями — постраничной печатью, вставкой в текст управляющих кодов в виде специальных символов.

107076. Москва, Стромынка, 20, Московский ин-т приборостроения; тел. 269-50-98.

Статья поступила 11.11.88.

УДК 681.325

В. В. Коробченко, А. П. Дидов, Н. Н. Шелкунов

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГРАММАТОРОВ ПЛМ

Разработчик микросистем (МС) знаком с выгодами использования программируемых логических матриц (ПЛМ) с плавкими перемычками [1]. Применение ПЛМ позволяет значительно сократить объем аппаратуры и унифицировать схему монтажа больших интегральных схем (БИС), сохранив при этом возможность изменения логики работы устройства. Из 39 БИС, на которых построена 16-разрядная одноплатная микроЭВМ ММС1213, пять — ПЛМ типа К556РТ1, К556РТ2. На их основе реализована не только классическая для данного случая логика выборки кристаллов, но и такой достаточно сложный автомат, как контроллер динамической памяти для ОЗУ К565РУ5, К565РУ7 [2], работающий в прозрачном для центрального процессора (ЦП) режиме без дополнительных тактов ожидания [3, рис. 6]. Контроллер гибкого диска (ГМД) для МС семейства ММС [3] удалось реализовать на шести БИС, две из которых — ПЛМ [4]. ПЛМ серий К556 были использованы при разработке новой 8-разрядной одноплатной микроЭВМ ММС1207 со встроенным контроллером ГМД, контроллера локальной сети Vitbus ММС2701, внутрисхемного эмулятора ММС9204 для однокристалльных микроЭВМ семейства МС51 [5].

Возможность применения ПЛМ в прикладных изделиях обусловлена развитыми инструментальными средствами (программаторами ПЛМ). Технические средства программирования ПЛМ [6] не находят широкого применения, так как нет хорошо проработанных программ по их обслуживанию. Система МПСР [7, 8] по программированию постоянных запоминающих устройств (ПЗУ) успешно внедрена благодаря мощному эффективному программному пакету МППС [9]. Для этой же системы был предложен периферийный модуль ММС9423 [8, рис. 7], рассчитанный на программирование ПЛМ серий К556 и КР556. Модуль ММС9423 аналогично другим периферийным модулям программирования логических схем [8, 10] подключается к входящему в состав МПСР интерфейсу проектирования И8102. Ориентация на специально предназначенный для таких целей интерфейс позволяет при переходе от ПЗУ к ПЛМ ограничиться разработкой лишь небольшого периферийного модуля, который во много раз проще, чем 2-платная система ММС9405 [6], рассчитанная на подключение непосредственно к внутрисистемной магистрали МС.

Универсальность пакета МППС — неизменность состава основных операций обслуживания ПЗУ и вида их карт прожига (размеры могут изменяться в значительных пределах). Основные параметры карты прожига определяются в файлах спецификации ПЗУ, поэтому пакет МППС удалось использовать и для обслуживания однокристалльных микроконтроллеров (МК) с электрически программируемым ПЗУ

Карта прожига ПЛМ КР556РТ1, КР556РТ2

	Входы (A _i)															Выходы (F _i)								
	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
00												1				0	1	1	1					1
01												0				1	1	1	1					1
02															1	0	1	1	1					1
03												1					1	1	1					1
04												0					1	1	1					1
05															0	1	1	1	1					1
06															1	0	1	1	1					1
07												1	0	1	1	1	1	1	1					1
08												0					1	1	1					1
09															1	0	1	1	1					1
0A															0	1	1	1	1					1
0B															1	0	1	1	1					1
0C												1	0	1	1	1	1	1	1					1
0D												0	1				1	1	1					1
0E												1	1	1	1	1	1	1	1					1
0F	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1
2F	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1

Примечание. Символы X, —, 0 и 1 в матрице И обозначают соответственно сохранность или удаление обеих перемычек, включение в конъюнктор входной переменной с инверсией или без нее; 1 и — в матрице ИЛИ — сохранность перемычки или ее удаление у соответствующего конъюнктора при формировании выходной переменной; 1 и 0 в матрице НЕ — высокий или низкий уровень активности выходной переменной.

[10]. Для этого потребовалось написать и отладить новый файл спецификации 87.ROM, включающий в себя всего 253 строки исходного текста на языке ASM80. Состав основных функций [11] и карта прожига ПЛМ (табл. 1) существенно отличаются от используемых при работе с ПЗУ и МК, поэтому для обслуживания ПЛМ потребовалось свое программное обеспечение.

В наиболее общем случае пакет программ по обслуживанию процесса программирования ПЛМ (рис. 1) должен обеспечить следующие функции:

- ввод текстового или графического описания логической схемы, предназначенной для записи в ПЛМ, в удобном для человека формате (создание исходных текстов);
- генерацию карты прожига ПЛМ по исходному описанию логической схемы с ее оптимизацией к конкретному типу ПЛМ (генерация загрузочных модулей);
- прожиг ПЛМ в соответствии с ее картой и создание карты прожига по готовым образцам ПЛМ для контроля за ее содержимым (программирование и чтение ПЛМ);
- возможность модификации карты прожига ПЛМ;
- создание выходных описаний ПЛМ в удобной для человека и совместимой с исходными текстами форме (генерация листингов).

Известен пакет программ автоматизации процесса программирования ПЛМ [12], рассчитанный на микровычислительный комплекс ДВК-2М.

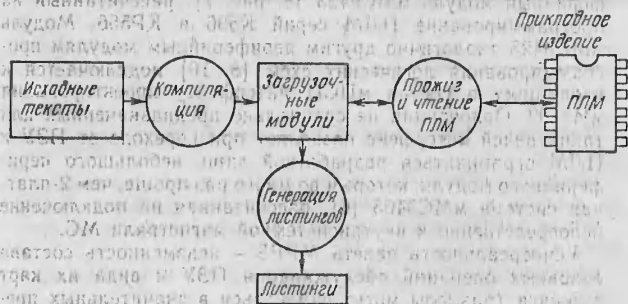


Рис. 1. Типовое ПО для программирования ПЛМ

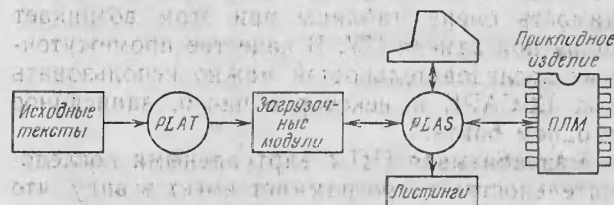


Рис. 2. Организация двухпрограммного пакета по обслуживанию ПЛМ

Существующая версия нового пакета, ориентированная на БИС типа КР556РТ1, КР556РТ2 и состоящая из двух программных модулей (рис. 2): оптимизирующего транслятора PLAT и диалоговой программы прожига и чтения PLAS — предназначена для другого класса МВК типа СМ1800, СМ1810. Рассмотрим работу пакета на примере проектирования 4-разрядного инкрементера (рис. 3): Вход INCI — линия разрешения счета. При INCI=1 устройство считает число импульсов на входе CLK, в противном случае переходит в режим хранения. Разрядность инкрементера наращивается с помощью выхода INCO, соединяемого со входом INCI следующей секции. ПЛМ КР556РТ1, КР556РТ2 не имеет встроенных регистровых схем, поэтому четырехразрядный регистр (рис. 3, а), реализуется внешними средствами. Логическая схема устройства (рис. 3, б) выполняется на базе ПЛМ.

В качестве исходных текстов используются символьные описания логики работы ПЛМ в виде булевых функций, которые менее наглядны, чем графические, однако средства их подготовки более просты. Для ввода символьных описаний используется текстовый (построчный или экраный) редактор, входящий в состав типового ПО микровычислительных комплексов общего назначения. Исходные описания логических схем состоят из логических выражений, разделяемых знаком ";". Предопределенные переменные A0...A15 и F0...F7 ссылаются на конкретные физические выходы БИС КР556РТ1, КР556РТ2 (табл. 1 и [11, рис. 1]). Для упрощения логических выражений допускается определение новых переменных. Новые переменные C0...C4

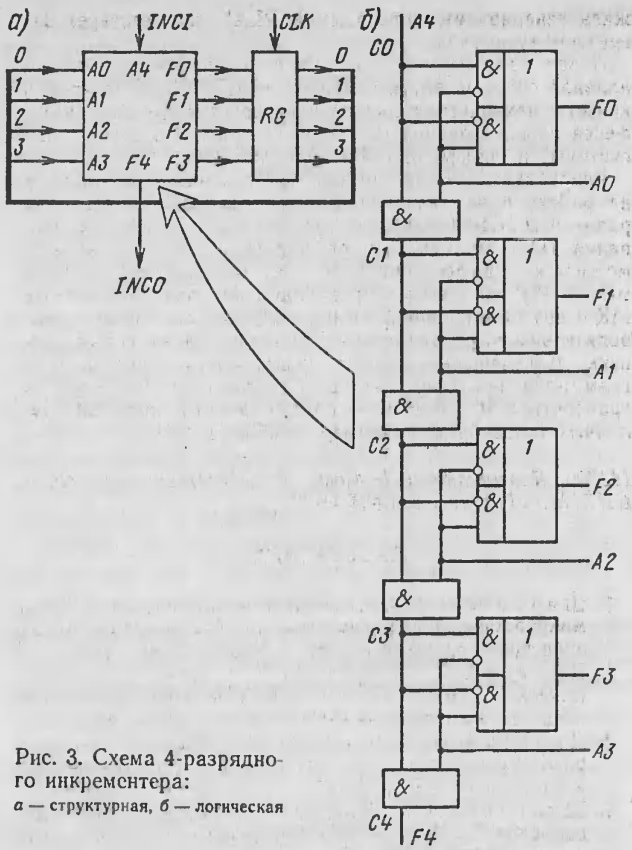


Рис. 3. Схема 4-разрядного инкрементера:
а — структурная, б — логическая

описания схемы рис. 3, б не только упрощают запись выражений, но и несут вполне конкретную смысловую нагрузку. Аналогично можно переименовать A0...A15 или F0...F7 в соответствии с их функциональным назначением в конкретном прикладном случае.

$$\begin{aligned}
 F0 &= A0 * C0 + A0 * C0; & C0 &= A4; \\
 F1 &= A1 * C1 + A1 * C1; & C1 &= A0 * C0; \\
 F2 &= A2 * C2 + A2 * C2; & C2 &= A1 * C1; \\
 F3 &= A3 * C3 + A3 * C3; & C3 &= A2 * C2; \\
 F4 &= C4; & C4 &= A3 * C3;
 \end{aligned}$$

На основании исходного модуля оптимизирующий транслятор PLAT создает загрузочный модуль (табл. 2), являющийся текстовым эквивалентом карты прожига (табл. 1). Перед генерацией загрузочного модуля транслятор оптимизирует исходные логические выражения, находя их минимальную дизъюнктивную нормальную форму [13].

Загрузочный и исходный модули имеют текстовый формат и могут обрабатываться обычными текстовыми редакторами. Однако структура текста в загрузочном модуле строго фиксирована, текст разбит на три части, разделяемые знаком '*'. В первой части описывается карта прожига матрицы И, один конъюнктор в каждой строке текста. Номер конъюнктора указывается явно после обязательного символа '&' в первой позиции строки. Порядок следования номеров конъюнкторов и их число не регламентированы. За номером конъюнктора через пробел следует его описание в форме, эквивалентной соответствующей строке матрицы И карты прожига (рис. 1). Во второй части модуля — карта прожига матрицы ИЛИ. Номер конъюнктора должен указываться после обязательного символа '!' в первой позиции строки. Правила кодирования строк матрицы ИЛИ соответствуют правилам их кодирования в карте прожига. Матрица НЕ описывается в последней части загрузочного модуля (одна строка):

Номер конъюнктора	Входные переменные															
&00	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&01	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&02	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&03	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&04	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&05	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&06	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&07	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&08	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&09	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&0A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&0B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&0C	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&0D	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&0E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
&0F	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
...&2F	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Таблица 2-2

Номер конъюнктора	Выходы															
!00	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!01	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!02	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!03	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!04	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!05	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!06	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!07	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!08	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!09	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!0A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!0B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!0C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!0D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!0E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!0F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
!...&2F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Таблица 2-3

Признак матрицы НЕ	Выходы							
^	1	1	1	1	1	1	1	1

Диалоговая программа PLAS — вторая часть пакета. Ее функция: генерация карт прожига, выводимых в табличной форме на консоль и принтер системы программирования ПЛМ; чтение загрузочных модулей и редактирование карт прожига в диалоговом режиме; программирование ПЛМ в соответствии с картами прожига; чтение текущего состояния ПЛМ и создание по нему карт прожига и загрузочных модулей. Они выполняются в диалоговом режиме с помощью набора команд (табл. 3). Конкретное содержимое карт про-

Состав команд PLAS Таблица 3

Команда	Назначение
H или ?	Вывод короткой справки о синтаксисе команд
P	Управление выводом эхо-файла на печать (результат — копия протокола диалога, карты прожига и их чистые бланки на принтере)
N	Вывод бланка карты прожига. Совместно с командой P обеспечивает получение на принтере чистого бланка карты прожига для его последующего заполнения при ручном кодировании
C	Очистка буфера. Его содержимое приводится к состоянию, эквивалентному чистой ПЛМ (предварительная подготовка буфера при ручном редактировании)
Li	Вывод содержимого буфера на консоль в форме карты прожига с ее разбивкой на i частей, i=1, 2, 3, 6. При нажатии любой клавиши, кроме X, вывод приостанавливается, X — прерывается с выходом на ввод новой команды (просмотр карты прожига)
SAnn	Редактирование матрицы И по строкам, pp — начальный номер конъюнктора
SOrn-pp	Редактирование матрицы ИЛИ по столбцам, m — начальный номер дизъюнктора, pp — начальный номер конъюнктора
SN	Редактирование матрицы НЕ
SLnn-A	Эквивалентно SAnn
SLnn-O	Редактирование матрицы ИЛИ по строкам, pp — начальный номер конъюнктора
SLnn-⟨CR⟩	Совместное редактирование матриц И и ИЛИ по строкам, pp — начальный номер конъюнктора, ⟨CR⟩ — возврат каретки.
E	Обмен содержимого основного буфера со вспомогательным (временно хранятся копии карты прожига в ОЗУ)
Dfile	Запись буфера в файл file в форме загрузочного модуля
Bfile	Чтение загрузочного модуля file в буфер
V	Контроль возможности программирования ПЛМ в соответствии с состоянием буфера (работа с не чистыми ПЛМ)
W	Прошивка ПЛМ в соответствии с состоянием буфера
Z	Сравнение состояния буфера и ПЛМ с выводом протокола разногласия
R	Чтение состояния ПЛМ в буфер
Q	Окончание работы, выход

жига генерируемых программой PLAS соответствует загрузочному модулю.

Пакет ориентирован на работу в составе профессиональной системы проектирования мПСП [7]. При этом в качестве технических средств используется адаптер интерфейса проектирования мМС8102 (входящий в состав этой системы) и периферийный модуль мМС9423.

Разработаны две версии пакета. Одна из них рассчитана на работу в рамках операционной среды СПО1800, вторая — в ДОС1800, ДОС1810 или ISIS-11. Пакет обслуживания ПЛМ значительно расширяет существующие возможности профессиональной системы проектирования мПСП [7] по программированию ПЗУ, однокристалльных МК и другой матричной логики (табл. 4), поднимая производительность разработчика МП-устройств на новый уровень. Дальнейшее развитие предложенных средств программирования ПЛМ связано с расширением возможностей транслятора PLAT и числа обслуживаемых системой различных типов БИС, например КР556РТ3 [12].

141700, Долгопрудный-1, Моск. обл., Институтский пер. 9 ЦНТТМ. «Физтех»; тел.485-44-77.

ЛИТЕРАТУРА

1. Щербakov О. А. Особенности применения ПЛМ в микропроцессорных системах // Микропроцессорные средства и системы.— 1986.— № 2.— С. 80—82.
2. Д и а н о в А. П., Щ е л к у н о в Н. Н. Организация динамической памяти микросистем // Микропроцессорные средства и системы.— 1987.— № 4.— С. 75—80.
3. Д и а н о в А. П., Щ е л к у н о в Н. Н. Одноплатные микроЭВМ семейства мМС // Микропроцессорные средства и системы.— 1990.— № 1.
4. Щ е л к у н о в Н. Н., Д и а н о в А. П. Организация контроллеров ГМД // Микропроцессорные средства и системы.— 1990.— № 3.
5. Щ е л к у н о в Н. Н., Д и а н о в А. П. Организация однокристалльных микроконтроллеров // Микропроцессорные средства и системы.— 1989.— № 5.
6. Д и а н о в А. П., Щ е л к у н о в Н. Н. Технические средства программирования логических схем // Микропроцессорные средства и системы.— 1986.— № 2.— С. 77—80.
7. Д и а н о в А. П., Щ е л к у н о в Н. Н. Система проектирования микропроцессорных устройств // Микропроцессорные средства и системы.— 1987.— № 5.— С. 83—86.
8. Д и а н о в А. П., Щ е л к у н о в Н. Н. Персональные модули программирования логических схем // Микропроцессорные средства и системы.— 1988.— № 1.— С. 40—44.

Средства программирования БИС в системе мПСП [7]

Таблица 4

Операционная среда	Основная программа	Файл спецификации	Основной модуль	Модуль спецификации	Состав обслуживаемых БИС ПЗУ, МК и ПЛМ
ДОС 1800, ДОС 1810 или ISIS-II	MPPS	573.ROM	мМС8102	мМС9411, мМС9412, мМС9412	K573PФ2, K573PФ5, K573PФ4, K573PФ6, 2764, 2764A, 2764, 27C64, 27128, 27128A, 27256, 27256, K558PP3, 27512, 9751H, 8744H, КР556РТ4, КР556РТ5, К155РЕ3, КР556РТ6, КР556РТ7, КР556РТ11...КР556РТ18, КР556РТ1, КР556РТ2, К556РТ1, К556РТ2
		27.ROM	мМС8102	мМС9412	
		558.ROM	мМС8102	мМС9412	
		Не создан	мМС8102	мМС9413	
		87.ROM	мМС8102	мМС9414	
		Не создан	мМС8102	мМС9421	
		Не создан	мМС8102	мМС9422	
		Не создан	мМС8102	мМС8424	
	PLAS и PLAT	Не нужен	мМС8102	мМС9423	

9. Щелкунов Н. Н., Дианов А. П. Организация программного обеспечения программаторов ПЗУ // Микропроцессорные средства и системы.— 1988.— № 2. С. 29—32.
10. Щелкунов Н. Н., Дианов А. П. Средства и методы программирования ОМК // Микропроцессорные средства и системы. 1989.— № 6.
11. Щелкунов Н. Н., Дианов А. П. Процедуры программирования логических матриц // Микропроцессорные средства и системы.— 1986.— № 2.— С. 71—76.
12. Система автоматизированного проектирования матричных БИС на базе микроЭВМ ДВК-2М / А. Г. Алексенко, А. А. Галицын, К. Н. Баландин, С. В. Нартов // Микропроцессорные средства и системы.— 1987.— № 2. С. 53—55.
13. Филиппов А. Г., Белкин О. С. Проектирование логических узлов ЭВМ — М.: Сов. радио, 1974.— 344 с.

Статья поступила 11.01.88

УДК 681.322.068

В. В. Марченко, С. Н. Стригина

ПРОГРАММАТОР РПЗУ СЕРИИ К573

Программатор предназначен для записи программ в РПЗУ с УФ-стиранием (К573РФ2, К573РФ21...К573РФ24, К573РФ5, К573РФ6) под управлением микроЭВМ «Электроника 60» или ДВК. С каналом микроЭВМ программа-

тор связан посредством устройства параллельного обмена Н2 через розетку Х5 (рис. 1). Напряжения источников питания 5 В и U_{PR} подаются через вилку Х4. Программируемая микросхема вставляется в колодку Х3. Для микросхем К573РФ23 или К573РФ24 предусмотрены специальные колодки Х1 и Х2. Перемычки между контактами КТ1...КТ12 установлены для записи программы в микросхемы К573РФ2, К573РФ5 или К573РФ23, К573РФ24. При записи данных в микросхему К573РФ21 переключкой соединяют контакты КТ7, КТ4; в микросхему К573КФ22 — контакты КТ7, КТ5 (при неизменном положении переключки на остальных контактах); в микросхему К573РФ6 — контакты КТ2, КТ3; КТ7, КТ6; КТ8, КТ10 (снять с контактов КТ11, КТ12). Программу пользователя загружают в оперативную память микроЭВМ с адреса IND, например 10000₈. В каждой ячейке оперативной памяти микроЭВМ размещают два байта программы с четным и нечетным адресами. Программу обслуживания программатора загружают в оперативную память микроЭВМ, например с адреса 1000₈.

Последовательность подачи кодов адреса

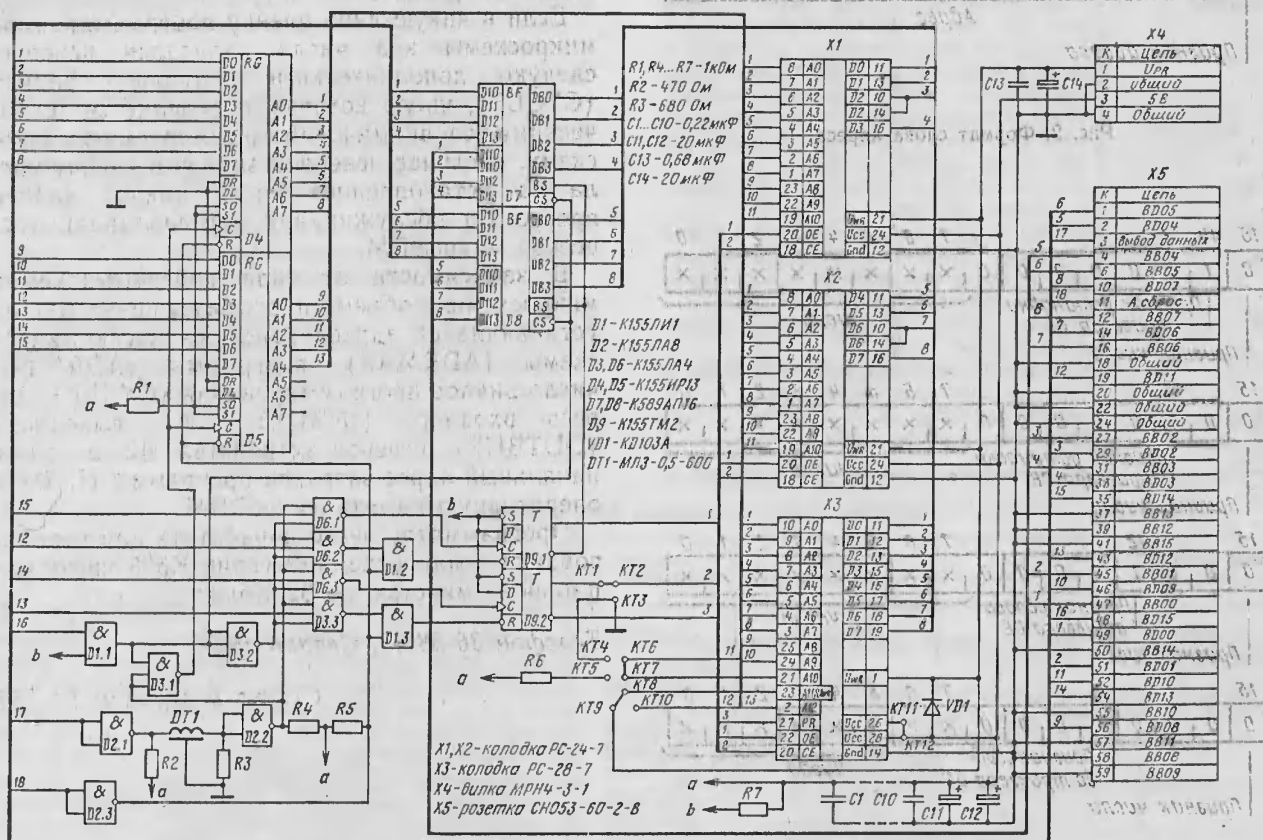


Рис. 1. Принципиальная схема программатора

программируемой ячейки микросхемы и записываемого числа (байта программы пользователя), установка и сброс сигналов управления (OE, CE или PR) путем записи и контроля в основном возложены на программу обслуживания программатора. Управление аппаратной частью программатора организуется в помощь слов адреса программируемой ячейки и записываемого числа (рис. 2, 3).

Процесс записи числа в выбранную ячейку состоит в генерации импульса длительностью $t_{CE} = 50 \pm 10\%$ мс с последующим контролем записанного числа. В режимах записи и контроля постоянно поддерживается напряжение программирования U_{PR} .

В цикле выдачи слово адреса, затем слово числа фиксируются в регистре выходного буфера устройства И2. Код адреса программируемой ячейки сигналом ВЫДАЧА ДАННЫХ заносится в регистр адреса (РА) D4, D5 программатора по признаку слова адреса. Одновременно на выводах выбора кристалла буферных микросхем D7, D8 формируется сигнал

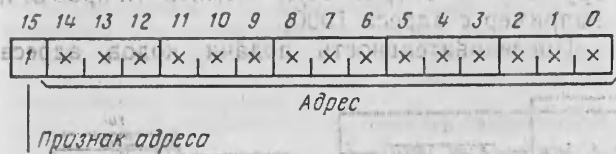


Рис. 2. Формат слова адреса

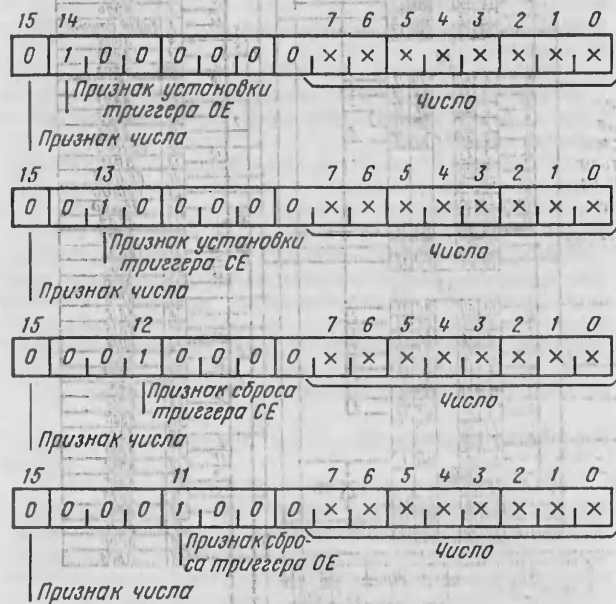


Рис. 3. Формат слова числа

запрета прохождения кода адреса на числовые выходы программируемой микросхемы. Признаком слова числа запрещается прохождение кода числа на РА и разрешается на числовые выходы программируемой микросхемы. Сигнал OE формируется триггером D9.1, сигнал CE — триггером D9.2. Устанавливаются и сбрасываются триггеры сигналом ВЫДАЧА ДАННЫХ. Разрешение на их установку и сброс обеспечивается программным путем: признаком слова числа и последовательной установкой Лог. 1 в разрядах 14...11 слова числа. Низким уровнем сигнала OE с инверсного выхода триггера D9.1 разрешается прохождение кода числа через буферные микросхемы D7, D8 на числовые выходы программируемой микросхемы. Задержка на 0,5 мкс сигнала ВЫДАЧА ДАННЫХ линией DT1 обеспечивает достоверность кодов, записываемых в РА и ячейку программируемой микросхемы.

В режиме контроля низким уровнем сигнала OE выходы D0...D7 программируемой микросхемы переводятся на выдачу кода записанного числа. Высоким уровнем с инверсного выхода триггера D9.1 разрешается прохождение кода числа через буферные микросхемы D7, D8 и входной буфер устройства И2 в канал микроЭВМ.

Если в какую-либо ячейку программируемой микросхемы код числа записался неверно, следуют дополнительные циклы записи (CYCLE), число которых определяется техническими условиями на программируемую микросхему. В случае неверной записи в ячейку числа за установленное число циклов записи программа обслуживания останавливает процессор микроЭВМ.

В зависимости от типа программируемой микросхемы и объема программы пользователь устанавливает адреса конечных ячеек микросхемы (ADRMAX), программы (ADRFIN), число циклов программирования (CYCLE), адреса входного (INBUF) и выходного (OUTBUF) буферов устройства И2, а также начальный адрес загрузки программы (IND) в оперативную память микроЭВМ.

Программатор легко доработать для использования с микросхемами серии K573 информационной емкостью до 32 Кбайт.

Телефон 36-38-53, Севастополь

Статья поступила 8.07.88

Л. И. Брусиловский, В. В. Складаров, Л. М. Хейфец

ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА MICROPOWER/PASSAL ДЛЯ РАЗРАБОТКИ АУТОНОМНЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Способы программирования автономных микропроцессорных систем

Автономные программные системы обычно создаются двумя способами:

с помощью автономного исполняющего монитора ОС, представляющего собой, как правило, усеченную версию монитора инструментальной системы; программирование на любом из языков инструментальной ОС. Этим способом созданы автономные ОС типа MRRT-11S или RSX-11S;

с помощью автономных исполняющих мониторов языков, работающих под управлением инструментальной ОС, выполняющей в этом случае роль кросс-системы. Исполняющий монитор языка программирования, на котором написано приложение, замещает монитор инструментальной ОС. Иллюстрируют этот способ автономные программы, написанные на языках FORTRAN VI, Modula 2, ADA, C, MP/P.

Первый способ создания автономных систем позволяет программировать приложения практически на любом языке или их смеси, но чреват избыточностью кода приложения. Отладка ведется в «родственной» инструментальной ОС с использованием отладочных средств ИВК (например, символьных интерактивных отладчиков).

Второй способ также может оказаться избыточным, если инструментальные средства не разработаны специально для создания оптимизированных автономных приложений. Основной недостаток данного способа — непродуманность средств отладки автономных программ.

Практически свободна от этих недостатков инструментальная система MicroPower/Pascal (MP/P), ориентированная на разработку автономных приложений для комплексов на базе микроЭВМ с системой команд LSI-11.

Структура инструментальной системы

Инструментальная система MP/P* разработана фирмой Digital Equipment Corporation (DEC) в 1981 г.

Автономные программы разрабатываются на ИВК под управлением инструментальной ОС, а работают на целевом комплексе в ОС MP/P.

Аппаратуру ИВК включает аппаратный диспетчер памяти, оперативную память на 128 Кбайт, сетевой или программируемый таймер, последовательный интерфейс для отладки на целевом компьютере по линии связи, дисковую память, достаточную для разработки приложения.

На аппаратуру целевого комплекса не накладывается никаких ограничений. Разрабатываемое приложение настраивается на конкретный тип целевой машины.

Первоначально в качестве инструментальной ОС использовалась только RT-11 XM, а в настоящее время наряду с ней — RSX-11M и VMS (можно применять ОС TSX-Plus и SHARE Plus).

Краткая характеристика языка программирования

MP/P как язык программирования представляет собой расширение стандартного языка Pascal, разработанного Йенсенем и Виртом, в сторону мультипрограммирования и программирования в реальном масштабе времени. Расширенные возможности MP/P доступны через программируемые запросы к исполняющей системе. В синтаксисе и семантике MP/P много общего с языками RTL-2, Modula-2, ADA.

К стандартным возможностям языка добавлены двоячные, восьмеричные и шестнадцатеричные целые, знаки $\$$ и

(_) в идентификаторах, структурные константы, тип данных UNSIGNED (беззнаковый), возможность перемещения текста из одного исходного файла в другой по команде % INCLUDE.

Атрибуты для подпрограмм, процессов, формальных параметров и переменных позволяют назначать граничные адреса для структур данных и переопределять выделяемый по умолчанию размер памяти для различных типов данных, задавать размер и имя стека.

Дополнительные директивы GLOBAL и EXTERNAL обеспечивают раздельное компилирование модулей, а FORWARD — предопределенное описание подпрограмм. Директива SEQ11 дает возможность генерировать стандартную для PDP-11 последовательность вызовов процедур и функций, написанных на других языках программирования (например, библиотеки научно-технических и инженерных расчетов на FORTRAN IV).

Наиболее существенные расширения стандартного языка Pascal связаны с мультипрограммированием и программированием в реальном времени. Мультипрограммирование обеспечивается определением параллельных процессов, синхронизованных через семафоры и кольцевые буферы. При этом различают статические процессы, определенные при инициализации приложения, и динамические, порожденные во время выполнения приложения. Активизация процессов, приостановка, возобновление, удаление, обмен между процессами организованы через встроенные программируемые запросы к исполняющей системе.

Программирование в реальном времени заключается в управлении внешними и внутренними прерываниями и исключительными ситуациями. Встроенные в язык MP/P программируемые запросы связывают вышеперечисленные события через определение структуры данных (векторы и ловушки прерываний) с подпрограммами и процессами для их планирования.

Исполняющая система

Исполняющая система MP/P — это набор стандартного программного обеспечения, создающего системную среду на целевом комплексе. Она состоит из системного ядра и совокупности системных процессов.

Хорошо структурированное системное ядро включает набор модулей с четко определенными функциями и интерфейсами, что позволяет оптимально настраиваться на различные приложения. Функционально компоненты ядра можно разделить на:

служебные подпрограммы-примитивы (38 примитивов); диспетчеры примитивов, исключительных ситуаций и внешних и внутренних прерываний, управляющие существенными событиями;

планировщик процессов по прерываниям и приоритетам; процедуру, инициализирующую структуру данных ядра и устанавливающую статические процессы во время старта или рестарта.

Системные процессы — это набор стандартных статических процессов, обслуживающих внешние устройства и работу с таймером. Пользовательские процессы получают доступ к системным, посылая им сообщения с запросами через примитивы семафоров очередей. Системные процессы включаются в целевое приложение по мере необходимости на этапе разработки приложения.

Разработка приложений

Цикл разработки автономных приложений на MP/P тщательно продуман, хотя достаточно сложен. Его можно разделить на фазы создания: системного ядра для выполнения сервисных функций; стандартных системных процессов, обеспечивающих работу с внешними устройствами и обслуживание таймера; пользовательских процессов (рис. 1).

Для упрощения работы пользователю предлагается специальный командный файл MPBLD.COM, автоматизирующий процесс разработки.

Приложение разрабатывается на инструментальном

* MicroPower/Pascal-RT Documentation, DEC, Moynard, Massachusetts, 1984.

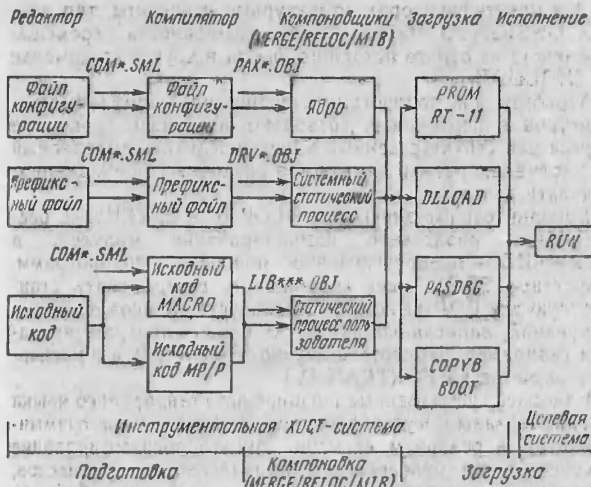


Рис. 1. Цикл разработки приложения на MP/P

ХОСТ-комплексе, отлаживается на целевом комплексе с помощью символического отладчика PASDBG с ХОСТ-комплекса по линии связи. Полученный исполняемый модуль загружается на целевой исполняющий комплекс через загружаемые внешние носители либо через ПЗУ.

Создание системного ядра связано с настройкой файла конфигурации на конкретное приложение, содержащее следующую информацию об аппаратуре целевого комплекса: тип процессора и его аппаратные опции; конфигурация памяти (объем, разделение на ОЗУ/ПЗУ); значения векторов и ловушек прерываний; объем памяти, выделенный в ядре для стека, пакетов сообщений, динамических структур данных.

После компилирования файл конфигурации сливают с одной из двух объектных библиотек — PAX*.OBJ — в зависимости от того, имеется ли на целевом комплексе аппаратное отображение памяти.

Для создания системных процессов используются соответствующие префиксные модули, которые компилируются и сливаются с одной из трех объектных библиотек хендлеров устройств — DRV*.OBJ. Префиксные модули статически выделяют область изменяемых данных для хендлера устройства, задают определенные специфические для данного устройства параметры (например, число контроллеров, адрес регистра управления и состояния, а также векторы прерываний для каждого контроллера и т. д.).

Дистрибутивный набор MP/P содержит префиксные модули хендлеров как на MACRO 11, так и на Pascal. Прикладные пользовательские процессы можно писать как на языке Pascal, так и на MACRO 11. В последнем случае используется одна из двух библиотек макроопределенной COM*SML (зависит от наличия отображения памяти). При компоновке используются четыре объективные библиотеки LIB***.OBJ, выбираемые в соответствии с системой команд микропроцессора.

Разработка приложений для систем на базе микроЭВМ MC1201, MC1212.

Для разработки приложений дистрибутивный набор MP/P дополнен с учетом конкретных интерфейсов. В первую очередь это относится к библиотеке префиксных модулей. Объектная библиотека DRV*.OBJ содержит следующие хендлеры устройств в объектном формате:

- CLOCK.OBJ — сетевой таймер;
- DDDRV*.OBJ — устройство типа DECtape-11;
- DXDRV*.OBJ, MXDRV*.OBJ, MYDRV*.OBJ — гибкие диски типа RX01, MX, MY; XLDRV*.OBJ, YADRV*.OBJ —

последовательный и параллельный асинхронные интерфейсы.

Имена системных процессов и префиксных модулей на исходном тексте приведены в таблице.

Имена системных процессов и префиксных модулей для микроЭВМ MC 1201

Процесс	Устройство	Префиксный модуль
Сетевой таймер	СК	CKPFX.MAC
DECtape 11	DD	DDPFX.MAC
Гибкий диск RX01	DX	DXPFX.MAC
Гибкий диск с интерфейсом	MX	MXPFX.MAC
	MY	MYPFX.MAC
Асинхронный интерфейс последовательный	XL	XLDPFX.MAC и XLDPFX.MAC
	YA	YAPFX.PAS

Разработка автономных многопроцессорных приложений

Начиная с версии 1.5 инструментальная система MP/P имеет средства для программирования многопроцессорных систем, состоящих из одного ведущего процессора типа LSI-11 и нескольких (до 14) ведомых периферийных процессоров типа KXT-11C, устанавливаемых на одной шине МПИ рис. 2. На шину можно добавить дополнительные контроллеры устройств ввода-вывода. Отличие KXT-11C от обычных устройств ввода-вывода состоит в том, что они программируются пользователем. Ведущий процессор LSI-11 выполняет в такой системе роль арбитра.

Периферийные сопроцессоры обеспечивают автономную загрузку задания, что значительно увеличивает производительность системы.

Арбитр и периферийные процессоры связаны по шине МПИ через двухпортовую память (TRX — Two Port RAM). Регистры управления, состояния и данных и вектора TPR предусмотрены в странице ввода-вывода арбитра. Программы имеют доступ к этим регистрам, как к устройствам ввода-вывода. Варианты конфигурации аппаратуры периферийного процессора определяют место TPR-области в странице ввода-вывода арбитра. Программная поддержка обмена осуществляется через КХ/КК-хендлеры.

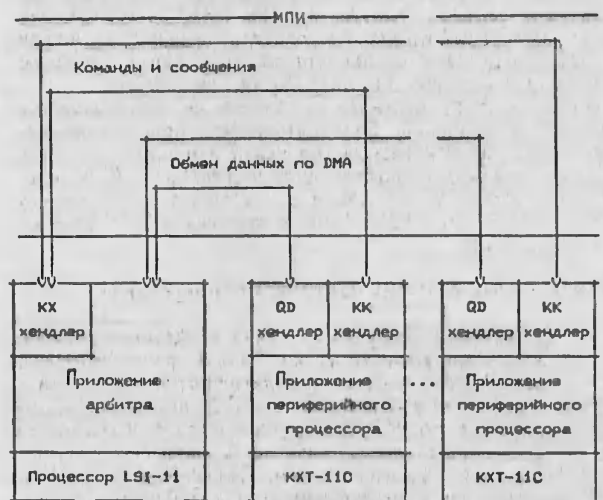


Рис. 2. Конфигурация прикладного программного обеспечения для многопроцессорной системы

Закключение

Язык МР/Р можно непосредственно использовать в качестве псевдокода для описания целевых систем автоматизации, так как в нем имеются синтаксические конструкции для мультипрограммирования и программирования в реальном времени. Компилятор МР/Р является многопроходным и оптимизирующим. Набор компонентов позволяет настроить приложение на определенную систему команд целевого микропроцессорного модуля. В системе МР/Р предусмотрен интерактивный отладчик PASDBG, работающий в лексике языка высокого уровня. Цикл разработки позволяет отдельно модифицировать системную (при изменении в составе аппаратуры) или прикладную части (при изменении спецификаций), что облегчает сопровождение.

Телефон 134-93-90, Москва

Статья поступила 18.05.89

УДК 681.3.06

В. С. Горбатов, В. А. Илларионов, А. А. Малюк, А. Е. Савин

ПАКЕТ ПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ ДИНАМИЧЕСКИХ ОБЪЕКТОВ НА ПЭВМ

Рассматриваемый пакет программ можно отнести к классу информационных систем, получивших название персональных баз данных (ПБД). Их классификация по функциональной направленности приведена на рис. 1. К специализированным ПБД относят системы, позволяющие решать вопросы частного, в основном, рутинного характера. Сужение их возможностей компенсируется большим дружелюбием, комфортностью и быстродействием. Для изучения сложных систем более интересны универсальные ПБД, обеспечивающие достаточно широкий круг функциональных возможностей.

Реляционный подход к построению универсальных традиционных ПБД основан на задании строгого формата документов (записей) в виде набора полей. На реляционной основе строится большинство практических ПБД. Недостаток такого подхода — необходимость введения специальных операций по связыванию данных [1] и недостаточно гибкий поиск, который в большинстве ПБД реляционного типа осуществляется по полям и предусматривает просмотр всего файла данного поля.

Среди средств, обеспечивающих формирование реляционных ПБД, можно назвать пакеты прикладных программ Pfl: life (IBM Filling Assistant) [2], dBASEII [3], Q&A [4], dBASEIII [5].



Рис. 1. Классификация персональных баз данных по функциональному назначению

Иерархическая модель данных в персональных системах основана на древовидном представлении, соответствующем структуре документа, разбитого на главы, разделы, параграфы, пункты и т. д. В результате достигается достаточно удобная «навигация» поиска необходимой информации, но значительно усложняется ввод и поиск данных, требуется большой объем внешней памяти. Формирование текстовой ПБД с иерархической надстройкой обеспечивает, например, пакет MaxThink [6].

Сетевые (распределенные) ПБД пока в чистом виде практически не используются, но им, без сомнения, принадлежит будущее.

Простейший тип текстовых ПБД — электронные карточки (CardFile Systems). Основное их назначение — выборка текста по ключевым словам. К пакетам, позволяющим формировать текстовые ПБД, можно отнести CardBox, DataFax, Free Form [7]. Для формирования электронных таблиц предназначены пакеты VisiCalc, What-if, Multiplan, «1—2—3», Super Calc III [8].

Существует множество русифицированных вариантов указанных пакетов, которые можно использовать на отечественных ПЭВМ. Однако нынешнее ничем не оправданное многообразие несовместимых моделей как самих ЭВМ, так и их программного обеспечения не позволяет провести достаточно точный обзор классификации отечественных инструментальных средств формирования и актуализации ПБД.

Поэтому возникает проблема создания оригинальных программных средств, рассчитанных на решение конкретной задачи, в частности моделирования сложных систем. Разработанный пакет представляет собой многоцелевой программный комплекс, обеспечивающий хранение и поиск информации, накопление вариантов моделирования и их анализ с помощью ПЭВМ «Искра 226». В основе комплекса — средства формирования ПБД. По нашему мнению, для эффективного моделирования структуры сложных систем необходимо использовать как реляционный, так и иерархический подход.

В состав разработанных средств входят информационный блок АС, позволяющий формировать иерархическую структуру с аналитической насадкой, блоки ПРИМА [9] реляционной ПБД и поддержки имитационного моделирования.

Блок АС совмещает поля с иерархической структурой. Поле вводится для целого поддерева, начинающегося с указанной вершины. При этом задаются его тип (числовое, символьное, типа даты) и формат. Значения полей задаются лишь на конечных (нижних) уровнях иерархии, называемых листьями. Кроме числовых значений в листьях может храниться и текстовая информация произвольной длины.

На форму дерева не накладывается никаких ограничений. Более того, имеющиеся средства дают возможность корректировать его структуру с различным числом уровней в ветвях и автоматически индексировать уровни, что облегчает поиск.

С помощью блока АС можно создавать и корректировать базы данных (БД), вести поиск информации, выдавать оглавление БД, анализировать данные, подготавливать диски.

БД создается путем левостороннего обхода дерева и ввода на каждом уровне названия и числа подуровней. На любом уровне можно задать новые поля, а на конечном уровне ввести значения полей, включая тексты. Средства реорганизации БД обеспечивают добавление, уничтожение, перемещение поддеревьев, коррекцию названий, редактирование информации.

Поиск информации проводится по дереву уровней, индексу, полям. Наиболее удобен поиск по индексу, однако в этом случае пользователь должен хорошо ориентироваться в БД. В противном случае поиск ведется по дереву путем выбора альтернатив при движении вверх или вниз. Добавляется поиск по полям, причем для числовых полей и дат можно выбрать в качестве усло-

вий сравнения следующие операции: «=»; «≠»; «<»; «>»; а для символьных полей — «=»; «≠».

Отличительная черта блока АС — наличие средств аналитической обработки, которая проводится в некотором поддереве БД. Для вычислений можно задать произвольную математическую формулу. Среди других возможностей блока АС — подготовка дисков, средства табличного отображения найденной информации и вычисленных показателей. Можно, используя один и тот же диск со структурой БД, работать с различными дисками информации. В целом блок АС обеспечивает хранение:

- структуры изучаемой системы (основное назначение данного блока);
- данных по конкретным объектам (как значений конкретных показателей) в соответствии со структурой системы;
- форматированных документов (как набора полей) с иерархическим оглавлением (здесь АС пересекается с другими блоками);
- текстовых документов произвольной длины с оглавлением.

В блоке ПРИМА документы представляются в виде набора полей, часть из которых поисковые. Все документы в БД имеют единый формат. Блок ПРИМА позволяет одновременно вести несколько БД со своими форматами документов. ПРИМА поддерживает поля текстовые, числовые и типа дата. Максимальное число поисковых полей в БД — 20.

Среди основных функций, реализованных в данном блоке, следует назвать:

- настройку системы на конкретное применение (создание новой БД);
- ввод и контроль документов;
- поиск и вывод информации;
- корректировку и просмотр БД;
- подготовку дисков.

Основной вид поиска в блоке ПРИМА — поиск по полям, причем в запросе может указываться несколько поисковых полей и условий сравнения. Возможен поиск по текстовым полям (поиск заданного слова внутри текста). Найденные документы выдаются в стандартном виде (в форме таблиц) с листанием вперед и назад, в форме электронной газеты (только значения текстовых полей), в графической форме (графики, прямоугольные и круговые диаграммы) в виде «твердой» копии на печатающем устройстве.

ПРИМА обеспечивает хранение форматированных документов как набора полей (основное назначение блока), конкретных данных по объектам (БД по узким конкретным областям), текстовых документов ограниченного объема (поиск по текстовым полям).

Отличительный блок рассматриваемого пакета — компонент поддержки имитационного моделирования на основе системной динамики. Блок включает (рис. 2) следующие БД:

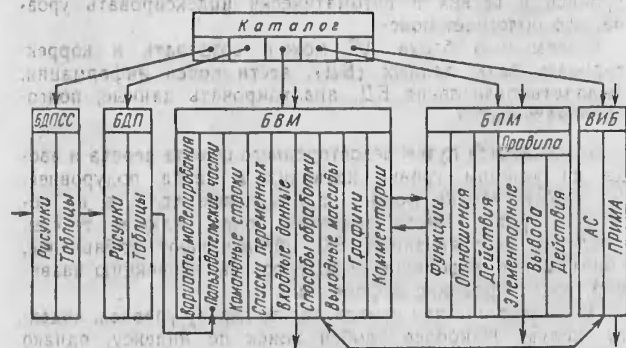


Рис. 2. Структура информационной базы данных имитационного моделирования

диаграмм причинно-следственных связей (БДПСС);
 диаграмм потоков (БДП);
 вариантов моделирования (БВМ);
 правил предметной области «Моделирование» (БПМ);
 вспомогательную информационную базу (ВИБ).

Предусмотрен общий каталог для всей базы в целом, имеющий иерархическую организацию. В БДПСС и БДП хранятся рисунки диаграмм и некоторые таблицы элементов и связей между ними. БДПСС содержит первичную информацию, вводимую пользователем. Поэтому особенно важно обеспечить простые и удобные средства ввода. Название каждой диаграммы хранится в каталоге.

Остановимся более подробно на организации БВМ. Хранимую в ней информацию можно разделить на два больших класса: компоненты модели и варианты моделирования.

- К компонентам моделей относятся:
- пользовательские части — наборы программных блоков, из которых строятся модели;
 - командные строки (модели) — последовательности блоков пользовательской части, задающие математическую модель процесса;
 - списки входов и выходов моделей — число входных и выходных переменных, а также их названия;
 - входные данные моделей — начальные условия, параметры (коэффициенты) моделей;
 - способы обработки моделей, в том числе и способы представления результатов;
 - выходные массивы (расчетные таблицы);
 - графики (результатирующие);
 - комментарии.

Каждая часть содержит множество файлов с названиями, каждое из которых хранится в своем разделе каталога. Каждый вариант моделирования строится из файлов компонентов модели восьми рассмотренных выше типов. Таким образом, любой вариант моделирования включает в себя пользовательскую часть, командную строку, список входов и выходов, входные данные и способ обработки модели, выходной массив, график и комментарий. Некоторые компоненты моделирования могут отсутствовать, тогда при работе с данным вариантом модели отсутствующий компонент будет задаваться не автоматически, а вручную. Любой компонент модели может иметь не одно, а несколько значений. Например, задавая два различных набора входных данных, мы можем последовательно просматривать как ведет себя модель при разных начальных условиях. Названия вариантов моделирования хранятся в специально отведенных разделах каталога.

БПМ представляет собой особую надстройку над БВМ, позволяющую проводить некоторый анализ полученных в процессе моделирования результатов, сравнивать модели, находить модели, удовлетворяющие определенным условиям, а также вносить необходимые изменения. По сути дела БПМ — это элементарная база знаний, основанная на правилах продукции.

ВИБ содержит дополнительную фактографическую информацию, которая может потребоваться пользователю при создании новых моделей или в процессе работы с ними. В качестве системных средств, обеспечивающих поддержку ВИБ, могут быть использованы ПБД АС и ПРИМА.

Основные информационные потоки показаны на рис. 2 стрелками. В соответствии с системно-динамическим подходом моделирование начинается с задания диаграммы причинно-следственных связей (ДПСС). На следующем этапе ДПСС переводится в диаграмму потоков (ДП), которая записывается в БД. Далее система переводит ДП в блок уравнений и сохраняет его в качестве варианта пользовательской части, которая является одним из компонентов модели в БВМ.

Работа с БВМ включает в себя несколько режимов. Это прежде всего режим корректировки, позволяющий создавать, уничтожать и корректировать как варианты

УДК 681.325.5-181.4

А. А. Камков, В. Ю. Лукоянов, Л. Н. Маренков
**ВСТРАИВАЕМЫЙ ДИСПЛЕЙ ДЛЯ СИСТЕМ
 УПРАВЛЕНИЯ ОБОРУДОВАНИЕМ**

Выделение контроллера дисплея в самостоятельную конструкцию позволяет разгрузить микропроцессорную систему от обслуживания экрана и клавиатуры, а также резко повысить ремонтпригодность. По формату изображения и системе команд дисплей совместимы с широко распространенным дисплеем «Электроника 15ИЭ-00-013».

Достоинства описываемого дисплея:
 применение трех форматов изображения с переключением по каналу ИРПС;
 полная двухрегистрационная клавиатура, защищенная профилированной резиновой пленкой;
 дополнительная функциональная клавиатура с 16 кнопками;

звуковой сигнал, управляемый по ИРПС;
 выделение информации на экране в виде негативного изображения, подчеркивая и мигая;
 встроенный «Рулон» изображения;
 одноплатная конструкция контроллера.

Контроллер выполнен на базе микропроцессорного комплекта КР580 на основе БИС КР580ВГ75, обеспечивающей все режимы работы экрана.

Функциональные узлы контроллера: тактовый генератор на ИС КР580ГФ24; центральный процессор КР580ВМ80А; системный контроллер КР580ВК28; последовательный канал связи на БИС КР580ВВ51А; схема опроса клавиатуры на БИС КР580ВВ55А; таймер КР580ВИ53; контроллер ЭЛТ на БИС КР580ВГ75; контроллер ПДП на БИС КР580ВТ57; ПЗУ программ на БИС К573РФ2; рабочее ОЗУ и видеоОЗУ на двух БИС КР537РУ10; знакогенератор на двух БИС К573РФ2; сдвиговый регистр на ИС К155ИР13; регистр управления режимами экрана, звуковым сигналом и светодиодами регистра клавиатуры; дешифратор адреса; схема формирования видеосигнала.

В контроллере использован один уровень прерывания RST 7, обслуживающий таймер и выход готовности приемника от контроллера канала связи. Схема управления форматом изображения содержит программируемый счетчик, понижающий частоту отображения точек в два или четыре раза, и переключает ПЗУ знакогенератора с разным размером шрифта.

Основные технические характеристики:

Число кнопок клавиатуры:	
основной	56
дополнительной	33
Форматы изображения, мм	80×24 40×24 20×18
Скорость передачи по каналу ИРПС. Ввод (определяется переключкой)	75...9600
Размеры плат, мм:	
клавиатуры	320×110
контроллера	300×110

Конструктивно дисплей состоит из двух плат с клавиатурой (кнопки ПКН-150), соединенных ленточным плоским кабелем с контроллером, и ТВ-монитора. Каждая из двух клавиатур защищена профилированной резиновой пленкой, обеспечивающей герметичность и хорошую тактильность кнопок.

Для питания дисплея (включая ТВ-монитор МС 6105) нужны два источника напряжения: +5В и +12В при токе 1А и 2А соответственно.

Тел. 534-94-59, Москва

Статья поступила 24.02.89

моделирования, так и отдельные компоненты моделей (пользовательские части, командные строки, входные данные и т. д.).

Основной для БВМ режим — работа с моделью. Пользователь либо задает один из хранимых вариантов моделирования, либо строит модель вручную непосредственно с экрана. С экрана вводятся также компоненты, отсутствующие в варианте моделирования. Ввод с экрана не означает, что пользователь не может работать с заранее подготовленной информацией (предусмотрена возможность загрузить файл компонента модели с диска по названию). Если какой-либо из компонентов модели имеет несколько значений, то данная модель «прокручивается» последовательно несколько раз. Система предоставляет также возможность сохранить сложившуюся последовательность действий в качестве нового варианта моделирования.

Работа с БПМ включает несколько режимов. Режим корректировки БПМ аналогичен соответствующему режиму работы с БВМ. В режиме вывода пользователь задает некоторое отношение и его аргументы (названия атомов). Система определяет истинно данное отношение или ложно. Таким образом, реализуется так называемый обратный порядок вывода. Вывод в рассматриваемой системе можно назвать элементарным, так как каждое отношение появляется в правой части правил только один раз. Другой режим работы с БПМ позволяет вычислять значения функций по значениям их аргументов (значениям атомов). В режиме модифицирования БД задается последовательность действий вместе с аргументами. Система последовательно проверяет их условия и выполняет действия, условия которых истинны. Наконец, предусмотрен режим, в котором пользователь задает отношение и часть его аргументов, а на выходе получает возможные наборы значений остальных аргументов (атомов), удовлетворяющие данному отношению.

Практически все режимы работы с БПМ используют и БВМ. Эти два информационные блока тесно связаны между собой и составляют информационное ядро системы в целом. ВИБ также связана с БВМ и БПМ.

Просмотр БД начинается с иерархического каталога. На нижних уровнях иерархии можно просмотреть, например, диаграмму, входные данные или график. В этом режиме организовано получение твердой копии различных видов рисунков, хранящихся в системе.

Описанный программный комплекс информационной поддержки моделирования эксплуатируется на микроЭВМ типа «Искра 226». В настоящее время он адаптируется к другим типам отечественных ПЭВМ.

Телефон 3-27-05, Путино Моск. обл.

ЛИТЕРАТУРА

1. Сенин Г. В. Персональные базы данных.— Тбилиси, НИИТИ, 1986.
2. O'Brien B. Pfs:fil and Pfs: report: Electronic File Cards // In: (4).— P. 178—181.
3. O'Brien B. Friday! A Long but Productive Day // In: (2).— P. 220—246
4. Whyte Ch. A Sense of Balance.— PC Word // Annual Review.— 1985.— P. 286—295
5. Hart G. A. dBASEIII: Advancing The Standart // In: (7).— P. 275—278
6. Hershey W. MaxThink // Byte.— 1985.— Vol. 10.— N 7.— P. 279—284
7. Rosenthal S. PC Filling: It's in the Cards // PC Magazine.— 1985.— Vol. 3.— N 7.— P. 349—350
8. Puuronen S. A Survey of Some PC Programs // Soviet Finnish Symposium on PC.— Tallinn.— 1985.— P. 6
9. Бабаян Т. Б., Горностаев Ю. М., Зиньков С. Н., Илларионов В. А. ИПС ПРИМА. Пакеты прикладных программ для микроЭВМ «Искра-226» // Методические материалы и документация по ППП.— М.: МЦНТИ.— 1985.— Вып. 28.— С. 148—184.

Статья поступила 22.11.88

АППАРАТУРНОЕ РАСШИРЕНИЕ ИНТЕРФЕЙСА ДВК

Попытка организовать на базе ДВК2 многофункциональные информационно-измерительные системы привела к необходимости аппаратного расширения интерфейса (рис. 1). Практическая схема расширенного интерфейса, реализованного в автоматизированной системе контроля

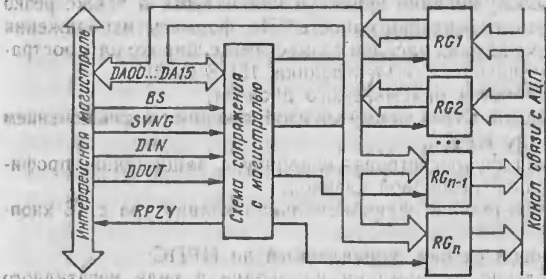


Рис. 1. Структурная схема интерфейса для связи с АЦП

транзисторных коммутаторов систем зажигания, приведена на рис. 2.

При обращении центрального процессора к АЦП на линии D00...D15 помещается адрес. При совпадении его базовой части (DA3...DA12) с настройкой дешифратора адреса (D1, D2.1...D2.5, D3.1) при активном уровне сигнала BS формируется сигнал DS (устройство выбрано). Базовый адрес интерфейса при включенном элементе D2.5 имеет значение 167770. По сигналу SYNG и активном уровне DS в зависимости от характера канальной операции (чтение — активен сигнал DIN, запись — DOUT) активизируется один из выходов дешифратора управления (D3.2, D2.6, D3.5, D10, D13). При этом стробируются бегеры либо приемные R61, R62 (D11, D14), либо передающие PG_{N-1} , RG_N (D9, D12) и осуществляется ввод или вывод данных. Изменяемая часть адреса (DA2 и DA1) позволяет инициализировать четыре буфера (на рис. 2 показаны два передающих и два приемных буфера), которые могут быть организованы для побайтного (D11, D14) и словарного (D9, D12) обменов. Число буферов можно увеличить, используя DA0 и соответствующую модификацию дешифратора управления.

310002, Харьков, ул. Фрунзе, 21, ХПИ им. В. И. Ленина, кафедра ИИТ, тел. 40-00-83

Сообщение поступило 27.06.89

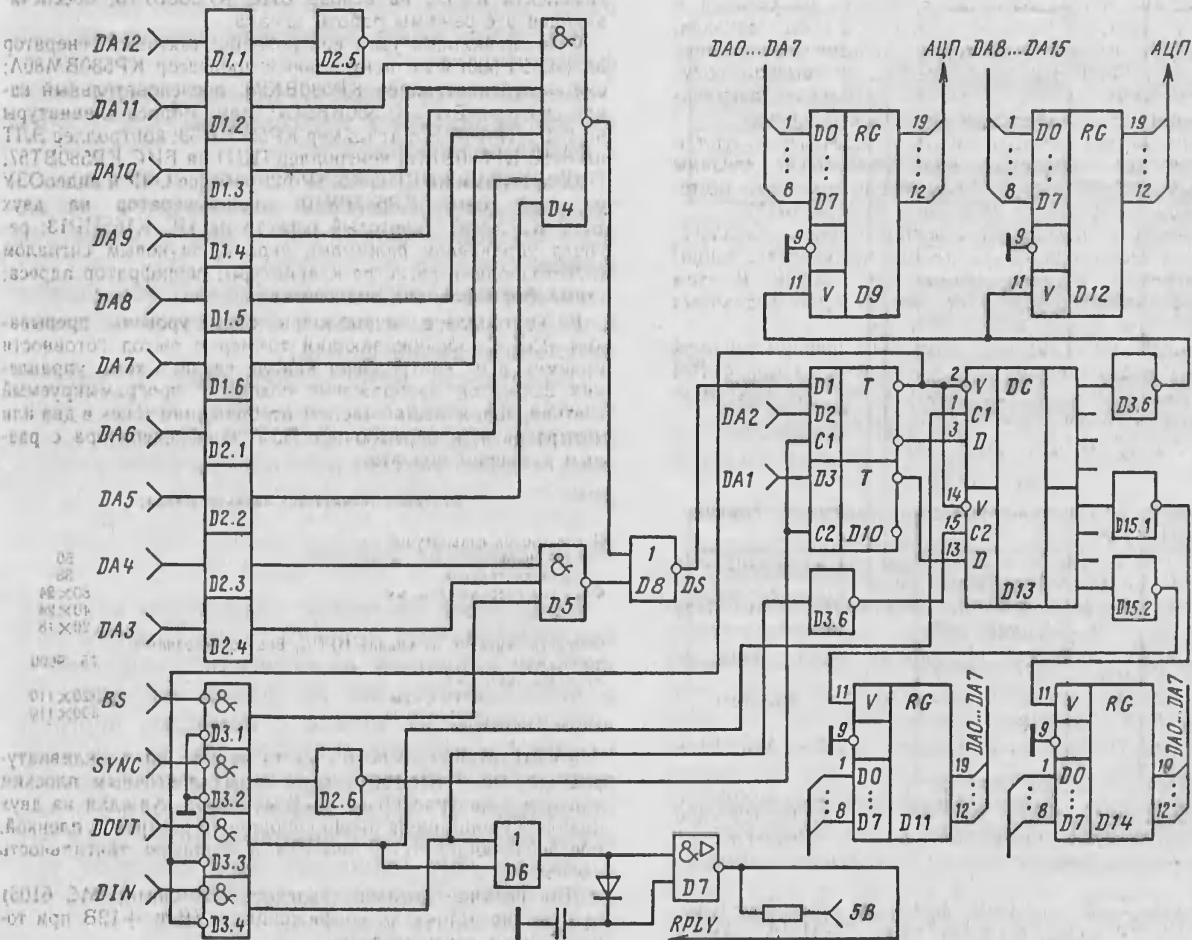


Рис. 2. Принципиальная схема интерфейса для связи с АЦП

ВОССТАНОВЛЕНИЕ УДАЛЕННЫХ ФАЙЛОВ В ОС RSX-11

Для восстановления случайно удаленных файлов в ОС РАФОС и ей подобных существуют команды монитора DIR/DEL и CREATE. Однако в ОС РВ или RSX-11 таких средств нет, и сложность файловой структуры не позволяет пользователю быстро и безошибочно отыскать на дисковом носителе потерянную информацию и восстановить ее.

Файловая структура FILES-11*

Основные элементы файловой структуры FILES-11 — индексный файл INDEXF. SYS, файл битовой карты тома BITMAP. SYS, оглавления пользователей XXXXXX. DIR, непосредственно пользовательские файлы FILNAM. EXT (рис. 1)

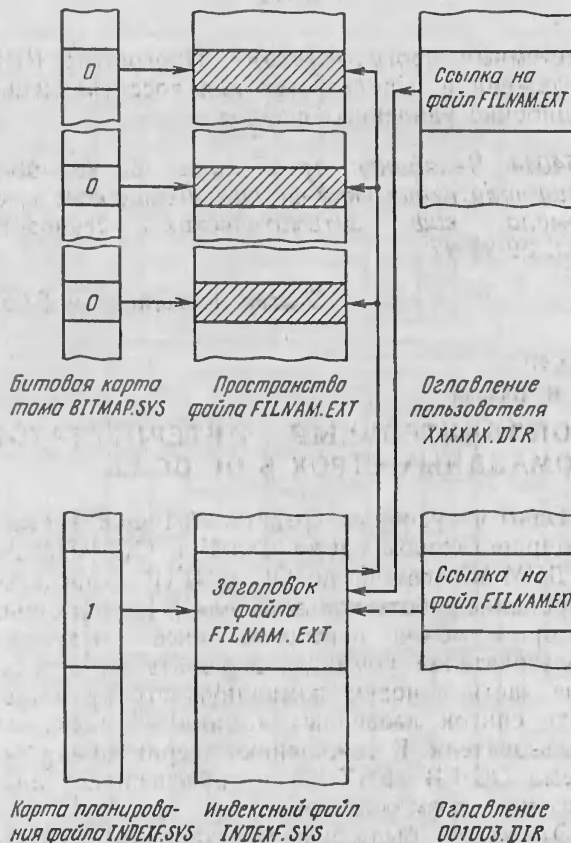


Рис. 1. Взаимосвязь элементов файловой структуры для файла FILNAM. EXT

* Операционная система ОС РВ СМ ЭВМ. — М.: Финансы и статистика, 1987.

Файл FILNAM. EXT занимает некоторое пространство на томе; поэтому для защиты пространства от повторной записи в битовой карте тома содержится информация о его занятости. Здесь каждый бит соответствует одному блоку, и если какой-либо бит равен нулю, то соответствующий блок считается занятым.

Чтобы получить информацию из пользовательского файла, необходимо знать, где конкретно он располагается. Для этого в индексном файле хранится заголовок, содержащий полную информацию о файле. Каждый заголовок занимает один блок индексного файла и включает номер файла, соответствующий номеру блока заголовка; последовательный номер, отражающий частоту использования блока заголовка; код владельца файла; имя, тип, версию, дату, время создания файла и прочие его атрибуты. Для определения занятости заголовков в индексном файле предусмотрена карта его планирования, в которой каждому биту соответствует один блок заголовка. Если бит в карте равен единице, то считается, что связанный с ним блок имеет заголовок существующего файла.

В оглавлении пользователя содержатся ссылки на минимально необходимую для работы с файлами информацию:

идентификатор файла (номер и последовательный номер), который используется для поиска полной информации о нем в индексном файле;

полное имя, по которому можно обращаться к файлу (это имя может отличаться от действительного имени файла).

Целостность этой информации, а именно: ссылка на файл, его заголовок, карта планирования индексного файла и битовая карта тома — обеспечивает нормальную работу с файлом.

Восстановление файлов

Из-за сбоев программ, аппаратуры или ошибок пользователя могут быть уничтожены файлы или нарушена целостность информации о них. Если ссылки на файлы потеряны, то восстановить их можно с помощью программы VEY с ключом /LO. Ссылки на все потерянные, но существующие файлы образуются в оглавлении [1, 3], их можно перенести в оглавление пользователя с помощью программы PIP с ключом /EN (синоним). При полном удалении файла признак удаления в его заголовке можно сбросить командой VEY/DE.

При полном удалении файла система управления файлами выполняет следующие операции:

из управления пользователя удаляет ссылку на файл;

в заголовке файла устанавливает бит признака удаления в байте H.SCHA, округляет поле номера файла H.FNUM и слово контрольной суммы блока заголовка H.CKSM;

в карте планирования индексного файла очищает бит, соответствующий блоку заголовка файла, и этот блок становится доступным для создания нового заголовка;

в битовой карте тома устанавливает, в единицу, все биты, соответствующие занимаемым файлом блокам, освобождая их для записи новой информации;

Остальная информация заголовка остается без изменений. Системное программное обеспечение позволяет восстановить битовую карту тома по информации индексного файла (VFY/UP) и образовать ссылки на потерянные файлы. Системных средств для восстановления блоков заголовков файлов не существует, но оставшейся без изменений информации в заголовке достаточно для восстановления недостающих ее элементов.

Для этого необходимо записать в поле номера заголовка удаленного файла H.FNUM его номер, определяемый из номера блока заголовка, вычислить контрольную сумму блока и записать ее в поле H.CKSM, установить бит в карте планирования индексного файла, соответствующий блоку заголовка. Эти операции можно выполнить с помощью программы ZAP, предварительно смонтировав том программой MOV с ключом /UNL, разрешающим коррекцию индексного файла. Остальные операции выполняются программой VFY.

Автоматизация восстановления

Для упрощения поиска заголовков удаленных файлов и их коррекции в системе RSX-11M V4.1 разработана специальная программа RDF (RESTORE DELETE FILES). Поиск заголовков удаленных файлов ведется по двум признакам: H.FNUM равно нулю и H.FSEQ не равно нулю. Имена обнаруженных таким образом файлов сравниваются со спецификацией файла, переданной в командной строке. Спецификация допускает поля типа «Дикая карта», что означает отсутствие сравнения данного поля с соответствующим полем заголовка. Для обращения к индексному файлу были использованы подпрограммы на языке MACRO II, вызываемые из основной программы, написанной на языке FORTRAN IV.

Фрагмент программы, модифицирующий индексный файл, приведен на рис. 2. По окончании работы RDF выполняются следующие операции: VEY/UP и VEY/LO (все операции восстановления файлов могут выполняться только

```

INTEGER*2    HEADER( 256 ),
1           H.FNUM,
2           H.CKSM,
3           H.SCHA,
4           N.BLK,
5           MAP( 256 )
C             ( ЗАГОЛОВОК ФАЙЛА.
              | НОМЕР ФАЙЛА.
              | КОНТРОЛЬНАЯ СУММА.
              | ХАРАКТЕРИСТИКИ ФАЙЛА.
              | НОМЕР БЛОКА ЗАГОЛОВКА.
              | КАРТА ПЛАНИРОВАНИЯ INDEXF.SYS.
C
C      EQUIVALENCE ( HEADER( 2 ), H.FNUM ),
1                ( HEADER( 7 ), H.SCHA ),
2                ( HEADER( 256 ), H.CKSM )
C
C      *          *          *
C      ЧТЕНИЕ БЛОКА ЗАГОЛОВКА С НОМЕРОМ N.BLK В HEADER
C
C      H.FNUM = N.BLK - 3
C      H.SCHA = H.SCHA .AND. '77777
C      H.CKSM = 0
C      DO      100  I = 1, 255
C              H.CKSM = H.CKSM + HEADER(I)
100     CONTINUE
C      ЗАПИСЬ HEADER НА ВРЕМЕННОЕ МЕСТО В INDEXF.SYS
C
C      ЧТЕНИЕ КАРТЫ ПЛАНИРОВАНИЯ INDEXF.SYS В MAP
C
C      I      = N.BLK - 4
C      J      = I / 16
C      K      = N.BLK - 4 - J
C      J      = J + 1
C      I      = ISHFT( I, K )
C      MAP(J) = MAP(J) .OR. I
C
C      *          *          *
C      ЗАПИСЬ MAP В КАРТУ ПЛАНИРОВАНИЯ INDEXF.SYS
C
C      *          *          *

```

Рис. 2. Фрагмент программы для восстановления заголовка файла

системным программистом). Программа RDF отлажена и используется для восстановления ошибочно удаленных файлов.

454044, Челябинск, пр. Ленина, 76, Челябинский политехнический ин-т им. Ленинского комсомола, каф. автоматических установок; тел. 39-93-42

Статья поступила 24.02.89

УДК 681.03

А. И. Паниди

ДОПОЛНИТЕЛЬНЫЙ ИНТЕРПРЕТАТОР КОМАНДНЫХ СТРОК В ОС ОС-PB

Одно из удобных средств, поддерживаемых интерпретатором языка Cshell в ОС ИНМОС и ДЕМОС (семейство ОС UNIX), — протоколирование работы пользователя [1], с помощью которого можно повторять ранее выданные пользователем команды, включать их отдельные части в новую командную строку, выдавать список введенных команд на терминал пользователя. К сожалению, операционная система ОС-PB (RSX-11) не обеспечивает аналогичных возможностей.

Эта задача была решена установкой в систему дополнительного интерпретатора командных строк CLI, в задачу которого входят: прием, сохранение и инициирование выполнения команд пользователя, распознавание и обработка запросов на повторение одной из ранее выданных команд, распознавание и про-

смотр списка введенных команд (HISTORY). Он написан на языке Фортран-IV и может быть использован в ОС-РВ с версии 3.0 или RSX-11 4.0, функционирующих на вычислительных комплексах серии CM, и в микроЭВМ «Электроника 60».

Рассмотрим общую схему работы интерпретатора. CLI получает команды из системной очереди по директиве GTCMSI (дать команду для CLI) и хранить их в специальном буфере. В отличие от интерпретатора Cshell CLI не создает дисковых файлов (процесс прохождения команд ускоряется). Расположения буфера команд в памяти может быть различным: в виртуальном адресном пространстве задачи CLI или в разделяемой общей области, которая прикомпоновывается к задаче в процессе построения. При хранении 20 командных строк длиной в 80 символов размер буфера (вместе с дополнительными структурами данных) составляет 1642 (10) байт на каждый обслуживаемый интерпретатором терминал.

При заполнении буфера каждая вновь поступающая команда «вытесняет» из него самую старую команду (хранятся последние 20 выданных пользователем команд). Поступающим командам присваиваются порядковые номера (номера команд и их тексты хранятся раздельно). С помощью специального массива, выполняющего роль указателя устанавливается однозначное соответствие между номером команды и адресом соответствующей командной строки в буфере команд. Такая структура данных позволяет достаточно просто и эффективно реализовать описанный выше механизм работы.

Если к моменту выдачи очередной директивы GTCMSI очередь оказывается пустой, то CLI прекращает работу. При этом возможны два варианта: интерпретатор остается активным, но переводится в состояние останова; осуществляется выход из CLI. Вариант выбирается при генерации интерпретатора и определяется состоянием специальной битовой маски. В первом случае расположение буфера команд не имеет значения, во втором — это существенно важно. При работе CLI с выходом на время отсутствия команд буфер команд не может быть в виртуальном адресном пространстве задачи интерпретатора, так как его содержимое будет потеряно. В разработанной версии интерпретатора буфер команд находится в разделяемой общей области, устанавливаемой в старших адресах системно-управляемого раздела GEN, на время отсутствия команд осуществляется выход из CLI.

Получив очередную команду, CLI в первую очередь определяет, является ли она командой HIS[TORY] и связана ли с требованием пользователя повторить одну из ранее выданных

команд. Если ни одно из указанных условий не выполняется, CLI копирует текст командной строки в буфер и передает ее на выполнение основному интерпретатору системы (MCR) по директиве RPOI (запросить выполнение и передать информацию о подключении). CLI не переводит синтаксического анализа команды и не ожидает ее завершения.

При получении команды HIS[TORY] CLI выводит на терминал пользователя список 20 последних команд с их порядковыми номерами. Команда HIS не копируется в буфер и, следовательно, отсутствует в списке команд. Признаком того, что пользователю необходимо повторить ранее выданную команду, служит символ «:», помещенный в первую позицию командной строки. Разработанный интерпретатор распознает следующие виды запросов:

— повторить последнюю команду;
:15 — повторить команду с порядковым номером 15;
:ABC — повторить последнюю команду, начинающуюся с символов ABC.

Любая из перечисленных команд может быть дополнена конструкцией вида ^C1^C2^ . В этом случае после того как соответствующая командная строка будет найдена в буфере, в ее тексте контекст C1 будет заменен на C2. После выборки требуемой команды и модификации ее текста (если это требуется) она вновь копируется в буфер в качестве последней команды и передается на выполнение MCR.

Установка CLI в операционной системе имеет ряд особенностей. В ОС-РВ интерпретаторы командных строк могут быть установлены с именами задач в двух формах: с прототипом имени задачи или с именем задачи по соглашению [2]. При массовом использовании разработанного CLI в пределах конкретной вычислительной системы целесообразно устанавливать его с именем задачи по соглашению. В этом случае все команды со всех терминалов, связанных с CLI, ставятся в очередь к единственной копии задачи — CLI (нагрузка на оперативную память значительно снижается).

При работе пользователя за терминалом, связанным с дополнительным CLI, при вводе символа CTRL/C появляется подсказка, соответствующая имени CLI. Кроме того, сам интерпретатор выдает сообщения, начинающиеся с текста «CLI--». Это может осложнить взаимодействие неопытного пользователя с системой. Для избежания подобных неудобств дополнительный CLI можно замаскировать под основной интерпретатор системы MCR (сообщения интерпретатора должны начинаться с символов «MCR--», а при инициализации CLI указывается ключ / CPR-«<15><12>/MCR/», в этом случае при получении символа CTRL/C терминальный драйвер выдает подсказку MCR».

Разработанный CLI реализует лишь основные возможности интерпретатора языка Cshell. Некоторые ограничения в работе с дополнительным интерпретатором связаны с тем, что в отличие от ОС ДЕМОС и ИНМОС практически все утилиты ОС-РВ обеспечивают интерактивный режим взаимодействия с пользователем. При этом команды пользователя не ставятся в очередь к интерпретатору, а направляются непосредственно к утилите. Следовательно, CLI не имеет доступа к этим командам и не может сохранить их в своем буфере. Таким образом, при интенсивном использовании режима интерактивной работы с утилитами эффективность CLI будет снижаться.

Другая особенность ОС-РВ — в очередь к CLI ставятся не только команды, вводимые с клавиатуры, но и команды, поступающие из косвенных командных файлов (ККФ). Очевидно, что при обработке командных файлов значительных размеров буфер интерпретатора будет заполнен командами ККФ, а команды пользователя — вытеснены. Это не всегда удобно, однако у пользователя появляется возможность повторять отдельные команды ККФ.

Разработанный интерпретатор командных строк может обслуживать несколько терминалов. Их число, число запоминаемых команд и максимальная длина командной строки определяются соответствующими параметрами в исходном тексте интерпретатора.

С целью уменьшения объема требуемой памяти возможно построение CLI с оверлейной структурой.

121355, Москва, ул. Молодогвардейская, 10, ГНИИХ; тел. 141-97-79

ЛИТЕРАТУРА

1. Тихомиров В. П., Давидов М. И. Операционная система ДЕМОС: инструментальные средства программирования. — М.: Финансы и статистика, 1986. — 206 с.
2. Операционная система ОС-РВ 3. Набор управляющих программ. Интерпретатор командных строк. Справочные материалы. 00086-01 97 02.

Статья поступила 9.03.89

УДК 681.3.06

М. Ю. Захаров

ПАКЕТ HALO

ФИРМЫ MEDIA CYBERNETICS

HALO — пакет подпрограмм [1], помогающих программистам в решении сложных задач обеспечения компьютерной графики. Его приложениями могут быть деловая и изобразитель-

ная графика, САПР, компьютерные игры, моделирование.

Библиотеки HALO созданы практически для всех основных языков программирования высокого уровня (BASIC-интерпретатор и компилятор, различные версии Фортрана, Паскаля и Си, Лисп) и ассемблера, работающих под управлением операционной системы MS DOS. Поддерживаются многие адаптеры растровой и векторной графики (более 20 типов), большое число принтеров и плоттеров.

HALO не претендует на соответствие какому-либо из существующих стандартов компьютерной графики: это компромиссная, ориентированная на конечного пользователя разработка, за основу которой взята часть функций из стандартов SIGGRAPH CORE и GKS и добавлены некоторые новые функции, облегчающие его использование.

В пакете HALO содержатся все основные примитивы компьютерной графики: построение точек, линий, прямоугольников (с заполнением и без), дуг, окружностей; перемещение частей изображения; запись изображения на диск и ввод его с диска; создание «твердой копии» изображения. Имеется большое число дополнительных средств: заполнение произвольных областей либо стандартными, либо определенными программистом типами штриховки; изменение толщины линий; определение программистом типов линий и т. п. Кроме того, в пакете есть подпрограммы, предоставляющие дополнительные возможности — динамическое распределение и освобождение памяти (не реализовано для Фортрана), копирование участков памяти (возможно со сжатием информации), большое число подпрограмм, специализированных для определенных устройств и позволяющих эффективно их использовать (например, работа с палитрами для адаптеров EGA и Qadram Palette Master). Однако пользоваться такими специализированными подпрограммами следует с осторожностью: хотя пакет HALO перенастраивается с одного типа графического адаптера на другой с большой легкостью (подключением средствами самого пакета нового драйвера экрана), в нем отсутствуют возможности определения типа установленного в конкретной ЭВМ графического адаптера. Поэтому вызов специфичной для другого адаптера подпрограммы или даже подпрограммы общего назначения, но с неверно выбранным драйвером экрана, приводит обычно к «зависанию» ЭВМ.

В пакете HALO хорошо развита работа с различными координатами на экране: целочисленными, определяемыми разрешением устройства, «мировыми», задаваемыми программистом, а также нормированными, соответствующими координатам устройства, приведенным к интер-

валу 0...1. В последних двух случаях координаты задаются вещественными числами. Имеется большой набор подпрограмм перехода из одной системы координат в другую, а также подпрограмм, позволяющих идентифицировать текущую систему координат. Кроме того, есть возможность выделять внутри мировой системы координат окна для изображения (такие окна автоматически «растягиваются» на весь экран) и независимо от них окна на экране. Это позволяет эффективно изменять масштаб изображения и создавать защищенные участки экрана.

В пакете HALO есть достаточно хороший выбор средств для вывода текстовой информации. Текст можно выводить по знакам (Fast text) или начиная с произвольного места графического экрана с определенным программистом размером символов в четырех ориентациях (Graphics text). Имеется также возможность вывода текста с использованием специальных шрифтов и в произвольной ориентации (Stroke text), но такие шрифты содержатся в отдельных файлах (вместе с пакетом поставляется всего один).

Удачно реализована запись изображения на диск — информация при записи достаточно сильно сжимается, что, однако, практически не влияет на скорость чтения-записи. Так, для стандартного цветного графического адаптера (CGA) с размером буфера экрана 16К, объем файлов с изображением получается порядка 2...4К и очень редко превышает 12К.

Но есть и недостатки. Например, программист должен сам позаботиться о том, чтобы на диске было достаточно свободного места, иначе изображение будет записано не полностью. Но основной недостаток состоит в другом. Невозможности средствами самого HALO вывести изображения, записанные на диск на другом типе графического адаптера, даже если адаптеры совместимы вплоть до структуры буфера экрана. Например, изображения, созданные на ЭВМ с CGA, нельзя прочесть средствами HALO на ЭВМ с платой EGA, в том числе, когда EGA работает в режиме эмуляции CGA и внутренняя структура файла с изображением позволяет сделать это без затруднений.

В состав HALO входит специальная программа LEARNHALO — интерпретатор подпрограмм пакета, помогающий быстро обучиться применению графических средств. Существует версия под названием Multi-HALO, отличающаяся от основной версии возможностью динамической загрузки драйверов экрана и внешних устройств для ввода-вывода графической информации.

Пакет HALO хорошо документирован, обладает высокой эффективностью и надежностью, что делает его удобным инструментом для разработки программного обеспечения систем ком-

пьютерной графики. Так, фирмой Media Cybernetics разработан пакет для подготовки публикаций HALO DPE [2], позволяющий совмещать текстовую и графическую информацию. Используя пакет HALO, удалось эффективно решить проблемы визуализации траекторий в задачах моделирования нелинейной динамики [3].

Телефон 333-24-88, Москва

ЛИТЕРАТУРА

1. HALO Graphics Primitives Functional Description Manual.— N. Y.: Lifeboat Association. 1984.
2. HALO DPE (Desktop Publishing Editor) Version 1.20.15. Users Guide.— Media Cybernetics, Inc., USA, 1987.
3. Chernikov A. A., Sagdeev R. Z., Usikov D. A., Zakharov M. Yu., Zaslavsky G. M. Minimal chaos and stochastic webs.— Nature.— 1987.— V. 326.— P. 559—563.

Статья поступила 5.11.88

УДК 681.326

А. В. Панфилов

ГРАФИЧЕСКИЕ АППАРАТНЫЕ СРЕДСТВА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ СЕМЕЙСТВА IBM PC

Графические устройства для компьютерного моделирования и проектирования (далее CAD/CAM/CAE) делятся на растровые и векторные с собственной вычислительной мощностью и простые. От обычных видеоадаптеров они отличаются повышенной разрешающей способностью и специальными аппаратными средствами, поддерживающими и ускоряющими процесс генерации изображения.

Одна из первых графических плат, пригодных для CAD/CAM-приложений, — популярная плата монохромной графики — HGC (Hercules Graphic Card), обеспечивающая разрешение 720×348 точек, имеющая 64 Кбайт видеопамати, совместимая в алфавитно-цифровом (А/Ц) режиме с IBM Monochrome и построенная на базе традиционного CRT-контроллера MC6845. Существует несколько типов HGC-совместимых устройств: Hercules In Color Card, HGCPlus, Hercules Color Card и др. Общее новое свойство названных адаптеров — загружаемый знак-генератор (Ram Font), существенно облегчающий адаптацию программного обеспечения, например русификацию. Кроме того, InColor Card и Hercules Color Card поддерживают графику в цвете.

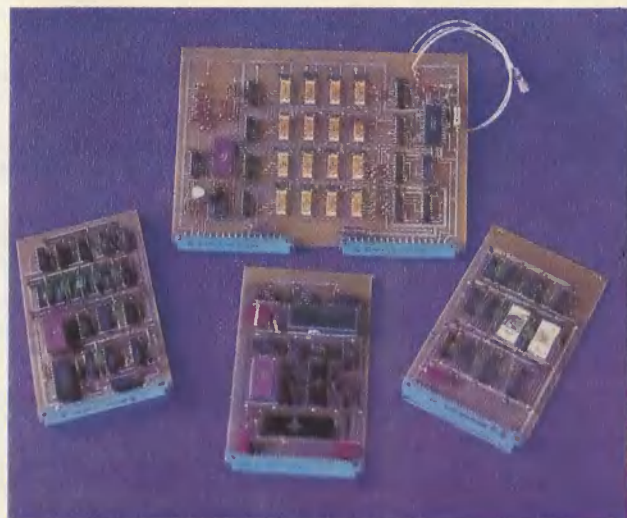
Графический режим HGC включается установкой бита 1 порта режима 3В8Н. Графическая память общей емкостью 64 Кбайт имеет упаковку видеоданных, отличную от CGA- и EGA-стандартов. Две страницы по 32 Кбайт разделяются каждая на четыре последовательных буфера по 8 Кбайт, биты в байте отображаются слева направо по одному биту на пиксел, а строки пакуются с интервалом в четыре строки. Например, в адресах В0000Н...В0059Н хранится первая строка, в В005АН...В00В4Н — четвертая и т. д. Начало буфера может задаваться по адресу В0000Н или В800СН программированием порта 3В8Н.

МИКРОКОНТРОЛЛЕР ДЛЯ ПОРТАТИВНЫХ УЛЬТРАЗВУКОВЫХ ДИАГНОСТИЧЕСКИХ ПРИБОРОВ

(к ст. Труханова А. И. и др.)

Для применения в ультразвуковых диагностических приборах эхоэнцефалоскопе, эхоофтальмометре и эхотомоскопе разработан ряд одноплатных специализированных модулей: микроконтроллера, памяти, знакогенератора, клавиатуры.

Основной из них — унифицированный модуль микроконтроллера (полуплата в центре фото) рассчитан на работу во всех приборах и предназначен для обслуживания функциональной клавиатуры, передачи данных на модуль знакогенератора, отображения алфавитно-цифровой информации на экране телевизионного монитора, управления перемещением визиров или маркеров по экрану монитора и расчета расстояний между ними, управления режимами работы прибора, расчета информативных параметров на основе табличных данных или фиксированных формул.



Основные характеристики микроконтроллера

Элементная база	ОЭВМ КМ1816ВЕ35
Объем памяти программ, Кбайт	2 (К573РФ5)
Число ТТЛ-совместимых линий ввода-вывода	35..40 (К580ВВ55А)
Потребляемая мощность, Вт, не более	3
Габаритные размеры, мм	160×100

Предусмотрена возможность измерения временного интервала (К580ВВ153).

Использование модулей позволило значительно расширить набор автоматизированных функций приборов, повысить их надежность, уменьшить габариты и вес.

УЧЕБНОЕ МНОГОФУНКЦИОНАЛЬНОЕ УСТРОЙСТВО

(к ст. В. Д. Цидялко и др.)

Разработано в Киевском политехническом институте для обучения студентов основам микропроцессорной техники. Это многофункциональный прибор со встроенным микроконтроллером на основе МП К580ВМ80.

На компактной информативной передней панели расположены клавиши управления, индикации и контрольные гнезда для наблюдения осциллограмм внутренних характерных точек микропроцессорной системы и измерительных модулей, дополнительный цифровой индикатор, имитирующий отсчетное устройство цифрового измерительного прибора. Двоичный светодиодный 16-разрядный индикатор с помощью переключателя режима может использоваться как линейный выходной индикатор или информационный индикатор состояния отдельных модулей устройства. Он указывает тип сигнала микропроцессорной системы, мультиплексирующийся на специально выделенное гнездо в поле контрольных точек. В устройстве предусмотрены девять гнезд контрольных точек для выведения 23 сигнальных линий, состояние которых визуальное контролируется осциллографом. Студенты наблюдают реальные временные диаграммы при выполнении программы и эпюры измерительного преобразования (например, процесс последовательного приближения в АЦП поразрядного уравнивания).

Наряду с режимом нормальной работы с номинальной скоростью выполнения программы предусмотрены отладочные режимы с остановкой после каждого машинного цикла или по достижении заданного адреса для детального изучения хода процесса вычислений и преобразований.

Конструктивно устройство выполнено в стандартном корпусе; допускается установка до 10 модулей одновременно; четыре определяют минимальную конфигурацию микропроцессорной системы, выбор шести зависит от курса лабораторного практикума.

Устройство можно использовать при обучении основам микропроцессорной техники, изучении схемотехники микропроцессорных комплектов и системы команд выбранного микропроцессора, исследовании структур измерительных приборов и систем, анализе алгоритмов коррекции погрешностей.



ВНИМАНИЮ ПОДПИСЧИКОВ!

С 1991 года наш журнал будет выходить под новым названием: «Средства и системы информатики».

В связи с повышением цен на бумагу, полиграфические услуги и особенно экспедирование, мы вынуждены поднять цену на журнал до 2 р. 20 к. за номер.

Выражаем уверенность в своевременности наших будущих встреч.

КОМПЛЕКС ПРОГРАММНО-АППАРАТНЫХ СРЕДСТВ ДЛЯ ОТЛАДКИ СИСТЕМ НА БАЗЕ ОЭВМ СЕРИИ К1816ВЕ31/51

УДК 681.3
А. Д. Цибульская

Комплекс ОКА51.3 построен на базе ПК IBM PC AT/XT и включает в себя внутрисхемный эмулятор ОКА51-ЭМ, пульт ручного управления ОКА51-ПУ и программу-отладчик ОКА51.3-ПО.

Эмулятор выполнен в металлическом корпусе с размерами 215×170×Х55 мм. Эмулируемая память программ пользователя — 8 Кбайт. Отлаживаемая программа может размещаться частично в эмуляторе, частично в ПЗУ системы — прототипа. Память данных пользователя может целиком располагаться в отлаживаемой системе или предоставляться эмулятором (2 Кбайт). Распределение памяти и конфигурации системы задаются программно. Эмулятор обеспечивает установку до 8 Кбайт контрольных точек.

Исполнительный процессор эмулятора — ОЭВМ К1816ВЕ31 выполняет программы либо пользователя, либо монитора. В мониторе пользователь имеет доступ к: памяти программ (предоставляемой эмулятором или расположенной в отлаживаемом устройстве), внешней и внутренней памяти данных с побитовой адресацией, памяти контрольных точек. Имеется возможность чтения и записи всех перечисленных ресурсов за исключением памяти программ, размещенной в ПЗУ, которую можно только читать. Следует подчеркнуть что, находясь в программе монитора, эмулятор обеспечивает полноценный доступ к элементам системы-прототипа, в том числе и к интерфейсным БИС. Это позволяет широко использовать эмулятор для диагностики готовых устройств.

Программа пользователя выполняется эмулятором в пошаговом режиме или реальном времени с остановом в контрольных точках. В случае необходимости программу можно остановить внешним сигналом, имитирующим контрольную точку в текущем адресе. Этот механизм используется для выдачи импульсов синхронизации при прохождении адресов, отмеченных контрольными точками (удобен для сигнатурного анализа и построения тестов устройств).

Для отработки прерываний в эмуляторе предусмотрено несколько режимов: прерывания запрещаются при входе в монитор и продолжают оставаться запрещенными при возврате в программу пользователя; аналогично предыдущему режиму, однако при возврате в программу пользователя восстанавливается состояние приоритетов и масок прерываний; прерывания в мониторе запрещены, но запросы на них фиксируются. Возврат в программу пользователя происходит после отработки зарегистрированных прерываний в соответствии с приоритетами, при этом восстанавливается состояние масок (прерываний).

Несколько режимов предусмотрено также для таймеров-счетчиков.

Пульт ручного управления обеспечивает автономную работу эмулятору, его программирование и полный доступ ко всем имеющимся ресурсам. Он реализован на базе ОЭВМ К1816ВЕ35 и связан с эмулятором последовательным каналом. Команды и данные вводятся с помощью 24-клавишной клавиатуры, отображаются тремя группами 7-сегментных индикаторов. Пульт выполнен в виде конструктивно-завершенного блока размерами 140×160×45 мм.

Программа-отладчик ОК051.3-ПО, работающая в среде ОС MS DOS, обеспечивает следующие возможности: символическое обращение к данным и адресам отлаживаемой программы и аппаратным ресурсам ОЭВМ; непосредственную отладку программ на ассемблере, языке PL/M-51 и встроенном языке высокого уровня DCL;

оперативное внесение изменений в программу без ее перетрансляции; формирование индивидуального отладочного окружения, в том числе экранного отображения.

Язык DSL грамматически близок языку PL/M. Его возможности:

введение собственных переменных для промежуточного хранения данных и управления процессом отладки;

создание отладочных процедур и вставок; замена участков программ фрагментами на ассемблере ОЭВМ и непосредственно на DCL.

выполнение команд ОЭВМ и операторов DCL при их непосредственном вводе с клавиатуры;

непрерывное и пошаговое выполнение программы с интерпретацией шага как команды ОЭВМ, оператора языка PL/M или процедуры PL/M;

организация квазинепрерывного режима, обеспечивающего расширенные возможности контроля программы;

трассировка программы.

Данный комплекс и аналогичные инструментальные комплексы ОКА51.1 на базе микроЭВМ типа ДВК и ОКА51.2 на базе микроЭВМ CM 1800/1810 поставляются НПК Пролог-88.

Телефон: 132-17-38, Москва



Отладочный комплекс ОКА51.3

Основные характеристики комплекса

Объем памяти:	
программ, Кбайт	64
внешней памяти данных, Кбайт	64
внутренней памяти данных, байт	256
Источники прерываний:	
внешние	2
внутренние	3
Встроенные 16-разрядные таймеры-счетчики	2
Встроенные порты ввода-вывода:	
последовательный	1
параллельный 8-разрядный	1
Максимальная тактовая частота, МГц	12

ДИАЛОГ

Если вы не можете найти общий язык со своим компьютером, язык BASIC фирмы Microsoft избавит Вас от ненужных хлопот!

Microsoft BASIC — это профессиональная система разработки программ для персональных компьютеров IBM и совместимых с ними!

Теперь все, что требуется для качественного программирования на языке BASIC, собрано в виде одного пакета. Новые возможности этого языка предназначаются для разработки программ коммерческого характера.

В Microsoft BASIC имеется много новых языковых возможностей, которые упрощают разработку прикладных программ. Наш новый пакет ISAM (индексно-последовательный метод доступа) полностью интегрирован в язык BASIC, поэтому Вы можете создавать быстро выполняемые прикладные программы с обширными возможностями для функционирования баз данных с помощью удобных синтаксических конструкций на языке BASIC — причем, без обращения к внешним библиотекам.

В языке Microsoft BASIC сняты все ограничения на объем с тем, чтобы предоставить Вам больше места для данных, больше памяти для программы на исходном языке и больше памяти для программы, полученной в результате компиляции.

Вы можете использовать расширенную память для увеличения места, которое требуется Вашей программе, поскольку Microsoft BASIC поддерживает в настоящее время спецификации расширенной памяти (EMS) версии 4.0. Вы можете иметь больше места для строк переменных, так как Microsoft BASIC теперь поддерживает строки типа «far». А поскольку Microsoft BASIC теперь поддерживает структуры наложений в процессе выполнения программ, то Вы можете писать программы, объем которых в результате компиляции достигает 16 Мбайт.

В наших библиотеках процесса выполнения повышена степень модульности программ, поэтому в Ваш выполняемый файл Вы можете включить только те их части, которые Вам совершенно необходимы.

Если же Вам нравится Microsoft Quick BASIC, то Вы полюбите расширенные возможности среды Microsoft Quick BASIC, — т. к. это те широкие возможности, которые Вам необходимы и которые предназначены, чтобы максимально увеличить Вашу производительность.

При условии поддержки полной спецификации расширенной памяти (EMS) расширенный Microsoft Quick BASIC может обрабатывать программы большего объема, чем просто Microsoft Quick BASIC. Нажатие на клавиши, меню, цвета и параметры компилятора — все выполняется по требованию заказчика. Объем отладчика увеличен вдвое.

Кроме того, Вы можете более эффективно обрабатывать ошибки, поскольку можете создавать программы для обработки ошибок как на уровне модуля, так и на уровне процедуры локально.

И, конечно, Microsoft BASIC — это все же только программный продукт BASIC, который поддерживает такие операционные системы, как MS-DOS, или PS-DOS, и системы OS/2, что позволяет Вам создавать программы на языке OS/2 как в режиме с защитой, так и в виде реальной программы.

Итак, если Вы хотите найти общий язык со своим компьютером и сэкономить при этом время — только Microsoft BASIC поможет Вам!

Наш адрес:

115598, Москва, ул. Ягодная, 17.

тел.: 329-21-74, 329-21-92

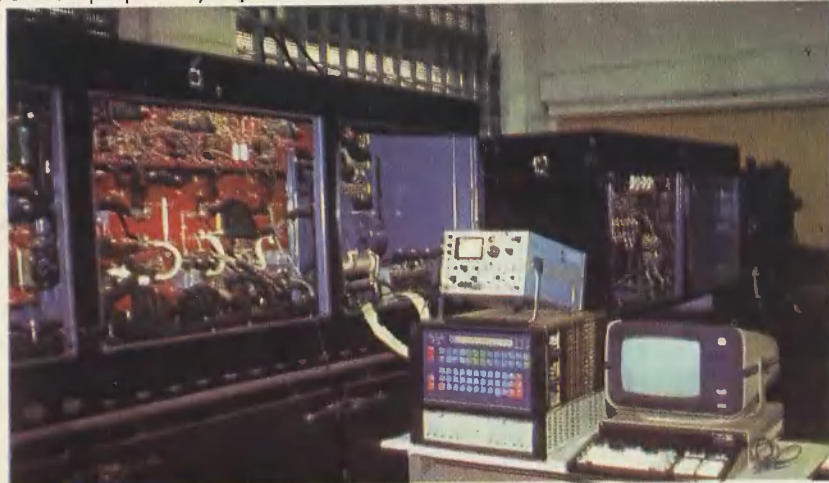
телефакс: (095) 329-47-11.

МИКРОПРОЦЕССОРНАЯ СИСТЕМА УПРАВЛЕНИЯ ЭЛЕКТРООБОРУДОВАНИЕМ МОТОРНЫХ ВАГОНОВ ЭЛЕКТРОПОЕЗДА

Система имеет двухуровневую структуру: на нижнем уровне решаются задачи управления тяговым электрооборудованием моторных вагонов; на верхнем — координации и контроля работы микроЭВМ нижнего уровня. Подсистема контроля расположена в головном вагоне. В целом система представляет собой локальную управляющую вычислительную сеть с моноканалом связи. Обмен данными в ней происходит только между центральной ЭВМ и микроЭВМ нижнего уровня. Последние информацией между собой не обмениваются, что значительно облегчает разработку протоколов об-

мена и дисциплины доступа к моноканалу.

Скорость обмена ограничена временем реакции системы на такие экстренные команды машиниста как «сброс» и «экстренное торможение» (не менее 0,1 с). Передаваемая информация включает закодированные команды машиниста, номера секций, значения тока уставки и скорости, а также различные признаки состояния. Всю информацию (с учетом помехоустойчивого кодирования) можно представить 5—6 байтами в посылке. Таким образом, процессы обмена информацией в системе можно отнести к категории низкоскорост-



Микропроцессорная система управления электрооборудованием моторных вагонов электропоезда

ных и реализовать в виде последовательной передачи информации по двухпроводному мультиплексному моноканалу. Такое решение дает ощутимый экономический эффект по сравнению с применяемой в настоящее время на электропоездах громоздкой многопроводной поездной магистралью. Например, на электропоезде ЭР200 экономится около 2,2 т медного провода стоимостью 4400 руб. При этом резко возрастают требования к достоверности передаваемой информации и, следовательно, надежности моноканала.

Подсистема управления электрооборудованием моторного вагона разработана на производственном объединении Рижский электромашиностроительный завод совместно с Томским НИИ автоматики и электромеханики (см. рисунок). Основу подсистемы составляют микросредства управляющей вычислительной техники серии В7 (МСУВТ В7), построенные на МПК БИС серий КР580 и К589. Выбор МСУВТ В7 объясняется удовлетворительным быстродействием указанных микросредств (500 тыс. опер/с), большой номенклатурой устройств связи с объектом и плат ввода-вывода. Подсистема ориентирована на электропоезда постоянного тока ЭР200 и ЭР30.

Комплекс программ управления включает модули плавного пуска, управления ослабления поля; электрического рекуперативного торможения, инициализации аппаратуры управления, ввода уставки тока и других параметров регулирования.

В перспективе на систему можно возложить задачи расчета оптимального тягового режима, расчета графика движения по перегону, задачи управления движением электропоездов, а также различные сервисные функции.

226010, Рига, Ганибу дамбис, 31, ПО «РЭЗ»; тел. 38-87-67.

УНИВЕРСАЛЬНОЕ СРЕДСТВО ВНЕШНЕЙ ДИАГНОСТИКИ

Устройство внешней диагностики (УВД) предназначено для диагностики и локализации неисправностей в любых цифровых блоках, узлах и устройствах микропроцессорной техники, а также выявления динамических неисправностей в реальном масштабе времени.

Аппаратные средства УВД представлены 48-разрядным логическим анализатором, состоящим из: ПЭВМ (ДВК-2М), интерфейсной платы в конструктиве полуплаты микроЭВМ «Электроника 60» с использованием семи управляемых вентиляционных матриц, корзины для установки тестируемых печатных плат и набора кабелей и заглушек.

ПО на ассемблере занимает четыре дискеты и под управлением ОС ДВК позволяет не только отбраковать проверяемый модуль, плату или устройство, но и (в случае одноплатной микроЭВМ) блокировать процессор тестируемого устройства и, эмулируя его работу средствами ДВК, исследовать работу отдельных узлов (ОЗУ, ПЗУ, интерфейс и т. д.).

В настоящее время тестовые программы написаны для печатных плат, входящих в состав ДВК2М, ДВК3, ДВК4, а также ПЭВМ отечественного производства, совместимых с IBM PC, Роботрон 1715, Ямаха.

Серийно выпускаемые УВД поставляются заводом по ремонту вычислительной техники ГКВТИ СССР.

Тел.: 299-62-53, Москва. Громыко А. Н.

Из монохромных адаптеров отметим два: WYSE-700 фирмы Sigma Technology и Laser View фирмы Sigma Design. Первый обеспечивает разрешение 1280×800 точек и поддерживает совместимость с CGA, используя четыре градации серого, второй — разрешение 1664×1200 точек при четырех градациях серого. Laser View поставляется с драйверами для ACAD, Windows, GEM, однако не совместим с существующими стандартами даже на уровне базовой системы ввода-вывода (BCBVB).

Графический адаптер EGA (Enhanced graphic adapter) предложен фирмой IBM в качестве нового видеостандарта для машин PC/XT/AT в 1983 г. Он поддерживает 16-цветную растровую графику с разрешением 640×350 точек из палитры в 64 цвета (640×200 точек — 16 цветов, монохромный — 640×350 точек) и эмулирует видеоадаптеры CGA и HGC. Все это относится к современной версии адаптера, имеющей 256 Кбайт видеопамати. Усложненная архитектура адаптера базируется на комплексе графических БИС в составе контроллера графических данных GDC, СРТ-контроллера, секвенсера и контроллера атрибутов. EGA состоит фактически из двух GDC (каждый управляет двумя из четырех планов графической памяти). GDC содержит простейшее ALU, поддерживающее булевы режимы записи и чтения планов памяти. Набор регистров контроля доступа позволяет динамически изменять поток графических данных.

В целом адаптер позволяет перемещать в видеобuffer или сравнивать четыре байта (восемь пикселей) видеопамати, поддерживает функции заполнения областей и быстрой заливки цветом, что частично снимает проблему разделения порта доступа к видеопамати (регенерация занимает 80 % пропускной способности интерфейса видеопамати), но сильно усложняет программирование в графических режимах из-за отсутствия прямого доступа к графическим данным. К достоинствам адаптера относятся возможности плавного скрола и панорамирования, аппаратная поддержка оконных режимов и загрузка пользовательских фонов.

Основные недостатки — сравнительно невысокое разрешение и медленное функционирование в графическом режиме повышенного разрешения (однако при эффективном использовании аппаратных возможностей адаптера можно получить сравнимое с CGA быстродействие). Фрагменты ассемблерных программ, демонстрирующие режимы чтения 0 и записи 2 для EGA- и VGA-адаптеров, приведены ниже.

; Базовые регистры управления EGA/VGA-адаптеров: индексный — 3 сег, данных — 3 сег.

; Режим чтения 0 задается установкой третьего бита регистра с индексом пять, план выбирается записью в регистр с индексом четыре
; На входе DS:SI определяются источник, ВЛ, передает номер плана,
; на выходе аккумулятор возвращает восемь значений восьми пикселей
; выбранного плана

```

READBYTE0 PROC NEAR
    MOV DX, 03CEH
    MOV AX, 0005
    OUT DX, AX
    MOV AH, BL
    MOV AL, 04
    OUT DX, AX
    MOV AL, DS:[SI]
    RET
READBYTE0 ENDP

```

; Запись данных в видеопамать организована по пикселям, т. е. значениями младшие четыре бита для 16 цветных режимов и весь байт для 256 цветных
; Регистр с индексом восемь определяет маску модификации восьми пикселей, режим записи устанавливается в регистре 5, операции с битами — в регистре 3 (в данной подпрограмме присваивается значение 0).
; На входе DS:SI — адрес в видеобufferе, ВЛ — маска, ВВ — значение пикселя.

```

WRITEBYTE2 PROC NEAR
    MOV AX, 0205H
    MOV DX, 03CEH
    OUT DX, AX
    MOV AH, BL
    MOV AL, 08
    OUT DX, AX
    MOV AX, 03
    OUT DX, AX

```

```

MOV BYTE BTR DS:[SI], ВВ
RET
WRITEBYTE2 ENP

```

Фирма IBM в составе семейства PS/2 предлагает два новых видеоконтроллера: мультицветный графический массив MCGA и графический видеомассив VCA. В отличие от IBM PC XT/AT и их видеоадаптеров, монтируемых в слот на системной плате, VGA и MCGA являются частью системы PS/2 [1, 2]. Модель 30 оснащена MCGA, модели 50, 60 и 80 используют VGA. Для совместимости и из коммерческих соображений IBM поставляет VGA-адаптер для модели 30, XT и AT.

Характеристики VGA и MCGA

Оба адаптера программно совместимы с видеоадаптерами CGA и EGA; MCGA эмулирует CGA на уровне портов ввода-вывода, CRT-контроллера MC6845 и карты памяти; VGA поддерживает полную программную совместимость с EGA-адаптером.

Улучшено разрешение систем в А/Ц и графическом режимах, увеличен набор цветов до 256 из палитры 262144 возможных. Для этого используется 18-битный ЦАП (6 бит на цвет) и аналоговый RGB-интерфейс. ВидеоЦАП обслуживают 256 18-битных регистра по 6 бит на первичный цвет; выбор регистров определяется содержимым видеопамати в данный момент времени [3]. Характеристика видеорежимов дана в табл. 1, наличие режимов в VGA- и MCGA-адаптерах показано в табл. 2.

Доступ к видеофункциям осуществляется посредством прерывания 10H с передачей параметров через регистры процессора. Расширения функций 10H прерывания приведены ниже. Функции прерывания 10H управляются засылкой их номера в AH и номера подфункции в AL с последующей генерацией прерывания.

Таблица 1

Характеристика видеорежимов адаптеров VGA и MCGA

Режим	Число цветов	Разрешение, символов (точек)	Формат экрана
0	16	40×25	А/Ц
2	16	80×25	»
11H	2	(640×480)	Графический
12H	16	(640×480)	»
13H	256	(320×200)	»

Таблица 2

Видеорежимы адаптеров VGA и MCGA

Номер	Режим	Адаптер	
		MCGA	VGA
0	320×400	+	
0	360×400		+
2	640×400	+	
2	720×400		+
11H	—	+	+
12H	—		+
13H	—	+	+

Новые функции прерывания 10H для VGA и MCGA

Функция 10H. Интерфейс цветовой палитры

AL=3: Переключение интенсивности/блик триггера А/Ц
AL=7: Чтение индивидуального регистра палитры (VGA)
AL=8: Чтение регистра цвета границы (VGA)
AL=9: Чтение всех регистров палитры и цвета границы (VGA)
AL=10H: Установка индивидуального регистра цвета ЦАП
AL=12H: Установка блока регистров цвета ЦАП
AL=13H: Выбор страницы цвета ЦАП (VGA)
AL=15H: Чтение индивидуального регистра цвета ЦАП
AL=17H: Чтение блока регистров цвета ЦАП
AL=1AH: Чтение статуса страниц цвета ЦАП (VGA)
AL=1BH: Просуммировать шкалу серого

Функция 11Н. Интерфейс генератора символов

AL=4: Загрузка A/Ц фонта 8×16
AL=14H: Установка A/Ц режима с использованием фонта 8×16
AL=24H: Загрузка графического фонта 8×16

Функция 12Н. Изменение предустановок

VL=30H: Выбор вертикального разрешения для A/Ц режима (VGA)
VL=31H: Разрешение-запрещение загрузки исходной палитры
VL=32H: Разрешение-запрещение видеодрессинга
VL=33H: Разрешение-запрещение исходного суммирования шкалы серого
VL=34H: Разрешение-запрещение эмуляции A/Ц курсора (VGA)
VL=35H: Интерфейс переключателя дисплея

Функция 1AH. Комбинационный код дисплея

AL=0: Чтение комбинационного кода
AL=1: Запись комбинационного кода

Функция 1BH. Функциональный статус

AL=0: Сохранить-восстановить видеосостояние (VGA)
AL=1: Определить размер видеобуфера для состояния
AL=2: Сохранить видеосостояние
AL=3: Восстановить видеосостояние

Функция 1AH позволяет определить тип используемого монитора, функция 1BH возвращает таблицу-дескриптор состояния видеоБСВВ: текущий видеорежим, активную страницу и размер доступной видеопамати и т. д. Эти функции удобны для резидентных программ при определении видеорежимов.

Алфавитно-цифровой режим. В этом режиме пользователь или БСВВ загружает в видеопамать четыре таблицы размерами 16×16 символов. Выбранные фонты копируются в две внутренние страницы знакогенератора. Изменяя исходный стандартный размер символа (8×16 точек на 8×8 точек), можно запрограммировать CRT-контроллер на отображение поля размерами 80×50 символов.

MCGA- и VGA-адаптеры настраиваются на режимы 400 и 480 линий вертикального разрешения соответственно. Кроме того, поддерживаются режимы 200 и 350 линий для совместимости с CGA- и EGA-адаптерами. Videобуферы всех адаптеров начинаются с адреса B800:0000 и имеют стандартные форматы записи символов и атрибутов.

Для адаптера VGA в отличие от MCGA нет ограничения на использование первых 16 регистров ЦАП, поэтому для него возможна любая из 262144 палитр. Кроме того, VGA обеспечивает управление цветом границы, а MCGA — нет из-за отсутствия в нем регистра выбора цвета I/03D9.

Графический режим. MCGA поддерживает два графических режима CGA и два новых. Карта памяти для режимов 11H и 13H (см. табл. 1) начинается с адреса A000:0000, распределение пикселей линейное — слева направо, сверху вниз. В режиме 11H один байт кодирует восемь пикселей, в режиме 13H — один пиксел.

VGA-адаптер организован аналогично EGA-адаптеру: содержит CRT-контроллер, секвенсор, контроллеры атрибутов и графический. Адреса внутренних регистров и функции у них совпадают. Режим 12H аналогичен режиму с разрешением 640×350 точек EGA-адаптера. Режимы чтения и записи также совпадают, но VGA поддерживает новый режим записи 3, аналогичный режиму 0, только битовая маска, определяющая активные пиксели, получается в результате операции логического умножения процессорного байта и значения регистра маски GDP. Это позволяет изменять маску битов, не перепрограммируя регистр маски.

Для VGA-адаптера каждые четыре бита атрибута (A/Ц режим) или 4-битное значение пикселя обрабатываются контроллером атрибутов и определяют один из 16 регистров палитры. Содержимое регистра (6 бит) комбинируется с 2-битным номером страницы цвета в регистр выбора цвета контроллера атрибутов. Полученный байт выбирает один из 256 18-битных регистров цвета, определяющих конечные значения первичных цветов. В случае использования монохромного монитора видеоБСВВ вычисляет значение шкалы серого перед загрузкой в заданный регистр цвета видеоЦАП по взвешенным средним значе-

ниям красного (0.3), зеленого (0.59), синего (0.11). В целом VGA-адаптер представляется несколько улучшенным вариантом EGA-адаптера и не дает заметного выигрыша в скорости или разрешения, однако уже сейчас становится новым промышленным стандартом.

Персональные компьютеры семейства PS/2 позволяют подключить через видеорасширение более мощную видеосистему (1024×768 точек, 256 цветов), сосуществующую с VGA-адаптером, что дает возможность использовать их как графические станции.

Интеллектуальные графические системы

Возрастающие потребности в сложной графике для компьютерного моделирования стимулировали разработку специализированных высокопроизводительных видеопроцессоров и систем, требующих для управления графикой минимального вмешательства ЦП.

Первой профессиональной графической платой с собственными вычислительными мощностями была IBM PGA (Professional Graphic Adapter), построенная на базе МП 8088. Она обеспечивает разрешение 640×480 точек при 256 цветах из палитры в 4096 цветов, эмулирует режимы CGA и занимает два слота компьютера. Интерфейс с ЦП реализуется на командном уровне через буфер в общем адресном пространстве. Прямой доступ ЦП к видеопамати невозможен, что иногда существенно усложняет программирование и несколько ограничивает диапазон применений системы. Однако PGA широко поддерживается ведущими программными фирмами в области CAD/CAM/CAE-продуктов (AutoCAD, MicroCADs, P-CCAD и т. д.). Современные версии PGA (Everex PGA) имеют значительно большую производительность и возможности.

Одной из первых дешевых графических систем с высоким разрешением и управляемой видеопроцессором для PC/XT/AT стала плата расширения ARTIST 1, выпускаемая фирмой Control Systems [2]. ARTIST 1 или 1+ представляет собой растровую графическую систему, спроектированную для машин IBM PC/XT/AT на базе графического процессора μ PD7220 фирмы NEC (далее GDC). GDC обслуживает четыре плана растровой видеопамати емкостью до 4 Мбит каждый и обеспечивает все сигналы синхронизации монитора. Максимальное разрешение системы 1024×1024 точек (возможны матрицы 1024×768, 960×720, 640×480 точек и др.).

Четыре плана памяти определяют текущее значение пикселя, задавая 4-битный адрес в таблице цвета, которая может быть запрограммирована ЦП на отображение 16 цветов из палитры в 4096 цветов.

ARTIST обеспечивает одновременное отображение текста и графики при неограниченном наборе пользовательских фонтов, хранящихся в системной памяти компьютера. GDC обслуживает дисплейную память по внутренней 16-разрядной мультиплексной шине адреса-данных, используя цикл чтение-модификация-запись, и поддерживает скорость рисования графических примитивов, равную 0,8 мкс/16 пиксел. Память организована в четыре массива по 64К 16-разрядных слов. GDC имеет 18-разрядную шину адреса (16 разрядов адресуют при отображении одновременно четыре 16-разрядных слова, обеспечивая 4 бит/пиксел; два дополнительных разряда определяют конкретный банк памяти).

ARTIST может работать в режиме внешней синхронизации как ведомый; в этом случае трехплатная конфигурация обеспечивает отображение 4096 цветов (по четыре бита на цвет). Система реализует аналоговый выход RGB-сигналов в соответствии со стандартом EIA RS-343, TTL-синхронизацию и световое перо.

Графический процессор GDC

Графический процессор существенно разгружает ЦП, выполняя следующие задачи:
рисование графических примитивов (дуга, прямоугольник, линия, графические символы);

полное управление растром и видеопамятью; скоростную обработку объектных данных от ЦП при параллельной генерации изображения; масштабирование и панорамирование изображения с коэффициентом 1...16;

генерацию изображения во время строчного бланкирования (режим может быть отменен).

ARTIST насчитывает шесть портов ввода-вывода с возможностью изменения адресов при помощи перемычек. Заводская установка перемычек для адресов 2EX, 3EX, 3CX в трехплатной конфигурации вызывает конкуренцию в адресном пространстве ввода-вывода с другими устройствами, например с EGA-адаптером. Можно порекомендовать установку адресов 2EX — красный, 2DX — зеленый, 2CX — синий. Такая комбинация поддерживается графическим программным обеспечением, разрабатываемым в ВЦ АН СССР, в частности системой «Редграф». Трехплатный вариант лучше монтировать непосредственно в шасси машины, а не в отдельном корпусе расширения системы, что позволяет использовать канал прямого доступа к памяти и повысить быстродействие системы.

Регистр масштаба ZOOM определяется базовым адресом со смещением 2. GDC взаимодействует с шиной посредством 16-байтного регистра FIFO и регистра состояния. Адресная линия A0 определяет запись команд (A0=1) или параметров (A0=0). Палитра в 4096 цветов получается путем программирования таблицы цвета размерами 16×2 (по 4 бит на первичный цвет) через порт выбора цвета; 4-битное значение пиксела определяет текущий адрес в таблице, выходы которой подключены к видеоЦАП.

GDC выполняет 20 команд, которые можно разделить на группы.

Команды режима:

- RESET: Сброс GDC.
- SYNC: Задание видеомасштаба.
- VSYNC: Выбор внешней синхронизации.
- CCHAR: Задание типа курсора.

Команды управления изображением:

- START: Разрешение сканирования
- BCTRL: Контроль бланкирования
- ZOOM: Задание масштаба
- CURS: Выбор пиксела в слове
- PRAM: Задание начального адреса, размера дисплейной области и символа
- PITCH: Задание масштаба X

Команды генерации изображения:

- WDAT: Запись байтов или слов данных в видеопамять
- MASK: Установка регистра маски
- FIGS: Задание параметров для процессора
- FIGD: Генерация заданной фигуры
- GCHRD: Генерация графического символа

Команды чтения данных:

- RDAT: Чтение байтов или слов данных из видеопамати
- CURD: Чтение позиции курсора
- LPRD: Чтение адреса светового пёра

Команды контроля ПДП

- DMAR: Запрос ПДП на чтение
 - DMAW: Запрос ПДП на запись
- GDC имеет 16-разрядное логическое устройство, позволяющее при модификации изображения замещать, дополнять, устанавливать или сбрасывать биты видеоданных в зависимости от содержимого регистров маски.

Основной недостаток ARTIST — отсутствие системной поддержки, невозможность использования в одномониторной конфигурации, а также невысокое быстродействие видеопроцессора, малая глубина FIFO и низкая «прозрачность» видеопамати для ЦП. Фирма сейчас выпускает плату ARTIST 10, 10/16, 12 на базе гораздо более производительного процессора Hitachi 63484 и новое семейство ARTIST HJ8/10/12 — на базе TMS34010, поддерживающее расширенную векторную графику (ARTIST 10/16-65 000 вект./с). Примечательно, что ARTIST 10/16 и ARTIST XJ используют 16-разрядную

шину (AT) и включают модули эмуляции режимов CGA или VGA.

Наряду с Hitachi 63484 получают распространение продукты на базе видеопроцессоров нового поколения: i82786, TMS34010/20, National 32016 и DP8500, NEC 7281 и др. Для новых систем характерно существенное увеличение производительности и гибкости при высокой разрешающей способности за счет введения целого ряда новшеств в архитектуру видеопроцессоров и памяти изображения.

Рядом фирм предложен новый тип видеопамати VRAM, используемой вместо стандартной однопортовой динамической памяти DRAM [3]. При обращении к однопортовой памяти изображения ЦП вынужден ждать свободного цикла (для адаптера EGA из 736 циклов на строку 640 циклов уходит на регенерацию изображения). VRAM-память имеет последовательный синхронный и независимый асинхронный порты, обеспечивающие прозрачный доступ ЦП к видеопамати и регенерацию. Широкое применение чипов VRAM сдерживается их высокой стоимостью и сравнительно небольшой емкостью.

Новые видеопроцессоры делятся на два основных класса — собственно видеопроцессоры с аппаратурной реализацией алгоритмов рисования графических примитивов и видеопроцессоры, работающие как сопроцессоры ЦП. Это i82786 и Hitachi 63484; для них не существует собственных программ, а лишь поток команд и данных для выполнения, получаемый от основного процессора. Ко второму классу относятся мощные микропроцессоры с сильно кооптированной параллельной архитектурой, имеющие универсальный, как правило типа RISC, набор команд (TMS34010/20, NEC 7281). Последний процессор относится к архитектурам не наймановского типа и реализует структуру управляющего потока данных с многоуровневым конвейером. Процессор оперирует структурами типа токенов, объединяющими данные, метки принадлежности, команды. До 14 процессоров могут быть соединены в дейзи-цепочку — конвейер.

Процессор i82786, спроектированный как мощный графический сопроцессор, в многозадачной системе на базе процессора i80386 устанавливает графические границы между задачами, создавая независимую карту памяти (окна) для каждого приложения. Он поддерживает разрешение 640×480×8 точек. Этот процессор использует для функционирования три независимых процессора: дисплейный и графический на кристалле и системный. Дисплейный обеспечивает интерфейс с монитором и регенерацию изображения, пользуясь дескрипторами запрограммированных окон, считывает видеоданные из заданных карт памяти с необходимой «глубиной» пиксела, т. е. поддерживает компактное кодирование.

Графический процессор поддерживает процесс генерации изображения и мощный набор графических примитивов. Приоритетное взаимодействие процессоров обеспечивает контроллер шинного интерфейса BIU. Кроме того, BIU управляет памятью типа VRAM и DRAM, поддерживает прямой доступ ЦП к графической памяти и доступ графического процессора к системной памяти. Набор команд процессора делится на четыре функциональные группы: контроля параметров рисования, общего контроля, генерации примитивов и блочных пересылок. Заметно, что i82786 не имеет команд условного ветвления, но позволяет выполнять подпрограммы, заданные ЦП по произвольным адресам. В целом процессор обеспечивает хорошую базу для создания мощной и дешевой графической системы для многозадачных приложений.

Графическую плату на основе i82786 для PC AT представила фирма Quadram. Система QuadHRG обеспечивает режим с разрешением 800×600×4 или 640×480×8 точек, BCIB поддерживает функции прерывания 10H, DGIS- и VDI-интерфейсы при одновременной генерации видеовыхода в HPG- и EGA-режимах.

TMS34010/20 — первый 32-битный графический системный микропроцессор, созданный специально для графических применений и поддерживающий пользовательскую

графику вместо встроенных графических примитивов. Процессор имеет 32-битное адресное пространство с возможностью адресации 1...32 бит в линейном и координатном масштабах.

Координаты X, Y — числа со знаком в диапазоне — 32768...32767. TMS 34010/20 — типичный RISC-процессор со скоростных ALU, параллельным 32 битным сдвижателем, 32-разрядной внешней шиной. Он содержит 31 32-битный регистр, разделенный на регистровые файлы A и B. Файл A включает регистры общего назначения, файл B — регистры, контролируемые графический контекст и параметры. Регистры ввода-вывода находятся в адресном пространстве TMS 34010 и контролируют глобальное взаимодействие в системе: параметры интерфейса локальной памяти, ЦП, прерывания и видеоконтроль. Системный процессор адресует графическую память косвенно через регистр адреса и получает заданное слово из регистра данных, что обеспечивает отображение 32-битного адресного пространства в четырех словах памяти или ввода-вывода РС. Набор из 120 команд универсален, поддерживает восемь типов адресации и четыре основных типа данных: массивы упакованных пикселей, X, Y — координаты, прямоугольные окна и битовые поля произвольной длины.

Система, базирующаяся на TMS 34010/20, в общем случае содержит видеомассив порядка 0,5 Мбайт VRAM и 1,5...3 Мбайт программной памяти DRAM. ПЗУ с программами инициализации, эмуляции предыдущих видеоадаптеров и графическими библиотеками. Взаимодействие такой системы с РС осуществляется на уровне геометрической модели объекта и условий наблюдения, что совсем недавно было возможно только на дорогих графических станциях.

Примером системы, реализованной на базе TMS34010/20, может служить плата расширения для PC ATGENESIS 1024 фирмы National Design, Inc. При разрешении 1024×768××4 точек плата обеспечивает скорость рисования до 48 Мпиксел/с, поддерживает выход в EGA-стандарте и поставляется с DGIS, CGI, AutoCAD и HPG (Harvard Presentation Graphics) совместимым интерфейсом.

Трехмерные графические системы

Основная задача трехмерной графики состоит в вычислении образа объекта по математическому представлению. Наиболее емко в плане вычислений процесс полутоновой штриховки 3D-поверхностей, требующий для представления в реальном времени (более 20 кадров/с) вычислительной мощности 30...500 MFLOPS.

Появление на рынке новых арифметических процессоров фирм AMD, Intel 860, AD, T132C030, Weitek 4167, Silicon Graphics с производительностью уже более 20 MFLOPS позволяет сделать предположение, что в

начале 90-х годов станут доступны широкому кругу пользователей одноплатные 3D-системы реального времени для персональных компьютеров.

Для семейства PC XT/AT предложено несколько различных плат, поддерживающих трехмерную графику, в частности интерес вызывают продукты фирм Densitron и Matrox, базирующиеся на геометрическом процессоре фирмы Silicon Graphics. Первая предлагает плату, включающую Hitachi 63483, National 32016, а также собственную СВИС теневого процессора, обеспечивающую производительность в 1500 тоновых 3D-полигональных поверхностей в секунду и до 7000 3D-вект./с при разрешении и цвете в стандарте VGA. Графическая плата M640/SM1281 фирмы Matrox Electronic Systems имеет близкие характеристики, поддерживает стандарты PGA, CGA и алгоритмы удаления скрытых линий-поверхностей.

Большие возможности для создания графических АРМ предоставляет новый 64-битный RISC-процессор i80860, имеющий пиковую производительность 120MIPS/80MELOPS, содержащий 3D-графическую подсистему на кристалле и встроенный процессор с плавающей точкой [4]. Набор команд оптимизирован для векторных операций и при параллельной работе подсистем возможно выполнение трех операций за один такт.

Возможности упомянутых систем огромны, однако вопросы переноса и совместимости программного обеспечения не только не снимаются, а, наоборот, обостряются. Поэтому такое большое значение приобретает проблема стандартизации графического интерфейса.

Математическое обеспечение, разрабатываемое в ВЦ АН СССР, поддерживает и развивает графический стандарт GKS VDI, взаимодействует с пользователем на уровне модели рабочей станции, облегчая перенос программного обеспечения с системы на систему. В данном случае адаптация существующих графических систем на новых аппаратных средствах сводится к написанию драйвера соответствующего устройства, реализующего функции конкретного графического стандарта.

117333, Москва, ул. Вавилова, 40, ВЦ АН СССР;
тел. 135-16-20

ЛИТЕРАТУРА

1. G. Michael Vose 'The technical Implications of the PS/2'
2. Richard Vilton 'PS/2 video programming' // Byte.— 1987, Extra Edition, Inside the IBM PCs
3. ARTIST. Color graphic controllers., Owners manual.
4. Demetrescu S. Moving Pictures // Byte.— 1985.— N 12.
5. The Intel 80860 // Byte.— 1989.— N 12.

Статья поступила 5.11.88

УДК 681.3.06

М. А. Алексеевский, И. А. Ельник, Е. И. Розенштейн

ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ ДС-86

Исполнительная ОС реального времени ДС-86 для управления процессами обработки и передачи данных в информационных и управляющих системах написана на языке ассемблера ЭВМ «Электроника 60». Система ориентирована на размещение в ПЗУ, что значительно повышает надежность изделия и является необходимым условием при разработке систем с жесткими ограничениями на весогабаритные характеристики, в частности одноплатных встраиваемых микроЭВМ.

ДС-86 проектировалась для одно-

платных микроЭВМ «Электроника С5-41» [1], переменного состава аппаратных средств, размещения программ в ПЗУ, минимизации объема используемого ОЗУ, возможности работы в реальном масштабе времени. При разработке ДС-86 использованы некоторые решения, реализованные в ОС RSX-11 и RMX/80.

Одно из основных требований к ОС реального времени — минимизация накладных расходов, связанных с их использованием, а именно времени процессора, а для малых систем — объемов потребляемой оперативной и

постоянной памяти.

ДС-86 построена по модульному принципу, что позволяет легко выбирать нужную конфигурацию системы, упрощает процесс ее сопровождения и обеспечивает максимальную экономию ресурсов ПЗУ и ОЗУ. Принятые принципы построения программ ядра ДС-86 дают возможность расширять ее функции за счет добавления новых модулей.

ДС-86 обеспечивает: централизованный механизм обработки прерываний;

параллельное выполнение задач в соответствии с приоритетами; временное диспетчерование; синхронизацию по времени и доступу к ресурсам;

управление обменом с внешними устройствами (ВУ) параллельно выполнению задач;

простой интерфейс между пользователем и системой;

формирование целевой системы в режиме начального пуска;

динамическое распределение оперативной памяти;

стандартизованный механизм работы с очередами.

Основная единица целевого программного обеспечения — задача. Целевая система может содержать до 256 задач, объединенных в группы (16 групп по 16 задач).

Задача может находиться в одном из следующих состояний (см. рисунок):

закрита: не установлена в системе (т. е. ДС-86 не располагает информацией о существовании задачи);

готова: установлена в системе;

активна: имеется запрос на выполнение задачи;

решается: занимает процессор;

прервана: решение прервано более приоритетной задачей;

ждет: решение приостановлено до свершения какого-либо события.

ДС-86 использует абсолютные статические приоритеты групп и относительные статические приоритеты задач в группах. Такая организация — пример компромисса между характеристиками системы и ограничениями на объем памяти. Основной единицей выделения ресурсов становится не задача, а группа, поскольку задачи одной группы друг друга не прерывают. Это дает возможность хранить не описатели задач, а описатели групп, иметь стеки групп, а не стеки задач и существенно экономить оперативную память. Однако при этом усложняется организация режима ожидания, так как перевод задачи в ждущее состояние означает невозможность решения других задач той же группы.

Группа имеет свой стек. Когда управление получает любая из задач группы, указатель стека определяет ссылку на начальный адрес этого стека. Кроме стеков групп имеется стек системы, начальный адрес которого задается при генерации системы.

Отдельной группе ставятся в соответствии статический и динамический описатели. Статический может распо-

лагаться как в ОЗУ, так и в ПЗУ. Он содержит адрес стека группы, максимальный номер задачи в группе и таблицу стартовых адресов задач. Динамический формируется в стеке группы и используется для запоминания общих регистров при прерывании и для хранения параметров.

Каждая задача может сообщить системе адрес ячейки, с которой система работает, как со шкалой, заноса при возбуждении задачи по К-му входу ($K=0, \dots, 15$) единицу в К-й разряд (механизм входов).

Механизм обработки прерываний организован по аналогии с принципами обработки прерываний в ОС RSX-11 [2]. Предусмотрены следующие уровни обработки прерываний: прерываний (УП); отложенных прерываний, или Ф-уровень (ФУ); задач.

Программы каждого из уровней подчиняются системным соглашениям по допустимому времени работы и доступу к ресурсам. На УП прерывания запрещены, и процессор не реагирует на внешние сигналы. Для снижения вероятности их потери (для исключения такой возможности) на УП допускается исполнение не более 10...20 команд.

Если необходимые действия по обработке прерывания не могут быть выполнены за это время, то система переходит на ФУ путем обращения к соответствующей подпрограмме. Программы Ф-уровня не прерывают друг друга и не прерываются задачами, т. е. это уровень более приоритетный, чем уровень задач. Запросы на выполнение программ Ф-уровня ДС-86 помещает в очередь отложенных прерываний (Ф-очередь), которая обрабатывается в соответствии с дисциплиной «первым пришел — первым обслужен». При переходе на ФУ устанавливается стек системы и исполняется 200...250 команд.

К программам всех уровней предъявляются определенные требования. Так, на УП запрещен доступ по записи к ресурсам, используемым на других уровнях. Программам ФУ запрещены обращение к диспетчерским директивам и доступ по записи к ресурсам, используемым на уровне задач. На уровне задач доступ к системным программам не ограничен.

Задаче предоставляются следующие механизмы синхронизации:

Запрещение прерываний. Запрещать прерывания можно не более чем на 10...20 команд, после чего они должны быть разрешены.

Запрещение отложенных прерываний, или переход на псевдо Ф-уровень (ПФУ). Задача, исполняемая на ПФУ, не может быть прервана ни задачами, ни программами Ф-уровня. Указатель стека при переходе на ПФУ не изменяется. На ПФУ допускается исполнение не более 200...250 команд. Переходов на ПФУ должно быть ровно столько, сколько уходов с него.

Запрещение прерываний задачами, или монополярный режим. Задача, решаемая в монополярном режиме, не может быть прервана другими задачами вне зависимости от их приоритетов. Задача, установившая монополярный режим, должна его снять. ДС-86 снимает этот режим при завершении задачи;

Использование семафорных примитивов.

Маскирование. Задача может запретить исполнение любой задачи. Если появится запрос на решение замаскированной задачи, она не получит управления, пока не будет размаскирована.

При появлении запроса на выполнение задачи ДС-86 помещает его в очередь групп и задач группы. Эти очереди построены, как шкалы, ряды в них распределены в соответствии с приоритетами задач или групп. Идентификатор задачи — номер группы, которой она принадлежит, и номер задачи в группе.

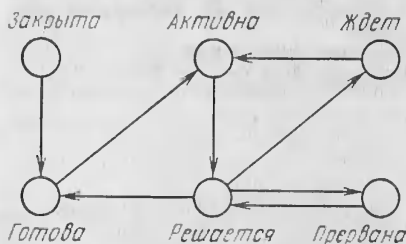
Предполагается, что для работы с ДС-86 пользователь применяет язык Макроассемблера. Взаимодействие реализуется посредством системных макрокоманд, подпрограмм и так называемых диспетчерских директив (ДД). ДД — это системные подпрограммы, получающие управление в результате программного прерывания и исполняемые на ФУ. Следствием выполнения ДД может стать передиспетчеризация процессов. Исходные данные для ДД называются макрокодами (МК). Адрес памяти, начиная с которого записан МК, — указатель макрокода (УМК). МК состоит из двух частей — стандартной (СтЧМК) и функциональной (ФЧМК). СтЧМК содержит системную информацию — номер МК, однозначно определяющий, какая ДД должна исполняться, и длину ФЧМК. ФЧМК отражает особенности конкретной ДД.

Вызывается ДД командой ЕМТ X. Расширение X определяет способ передачи макрокода. ДД передает задаче интегральный и расширенный коды возврата. Интегральный код показывает успешность или неуспешность выполнения ДД. Он передается в бите С слова состояния процессора. Расширенный код возврата передается в общем регистре R0. Все остальные регистры задачи сохраняются.

ДС-86 состоит из следующих программных блоков:

- супервизора задач;
- обработки времени;
- распределения динамической памяти;
- работы с очередями;
- супервизора ввода-вывода;
- инициализации.

Супервизор задач организует решение задач в соответствии с их приоритетами, строит Ф-очередь и выбирает из нее, возбуждает задачи и маскирует их, обеспечивает переход на ПФУ и в монополярный режим, стандартный выход из задачи, перевод за-



Граф состояний задачи

дачи в состоянии ожидания, механизм диспетчерских директив, а также возможность пользователю создавать свои, нестандартные, директивы, работающие на правах диспетчерских.

Блок обработки времени подсчитывает текущее время с точностью 0,1 с (при генерации системы пользователь может задать и другую точность), возбуждает задачи с заданным периодом в заданное астрономическое время или по истечении тайм-аута, а также отменяет запросы на их выполнение.

Блок распределения динамической памяти предоставляет возможность в процессе работы запрашивать и возвращать участки оперативной памяти. Адрес и длина пула динамической памяти устанавливаются пользователем в целевом начальном пуске обращением к соответствующей системной подпрограмме. При выделении памяти используется критерий «первый подходящий». При возвращении фрагменты памяти объединяются в односвязный список, упорядоченный по возрастанию адреса. Смежные фрагменты при этом «склеиваются». Если затребованный участок памяти может быть выделен, ДС-86 сообщает его адрес и длину, кратную четырем байтам. Если нет, то передается отрицательный код возврата. Стремление перевести в этом случае задачу, запрашивающую память, в состояние ожидания приводит к возникновению клинчей, распознавание и устранение которых требуют слишком больших затрат времени. Установление каких-либо дополнительных ограничений (например, отдать всю имеющуюся память перед тем, как запросить еще) приводит к неоправданному усложнению проектирования целевых программ. Предполагается, что в хорошо спланированной системе размеры пула будут определены так, чтобы в любой реальной ситуации памяти хватало.

Блок работы с очередями обеспечивает работу с беспriorитетными и приоритетными очередями, ставит в конец очереди FIFO, в очередь по приоритету, направляет элемент из начала очереди по адресу или образу.

Супервизор ввода-вывода ДС-86 можно рассматривать как составную часть системы ввода-вывода, содержащей драйверы ВУ. Каждому ВУ в системе ставится в соответствие описание и область ОЗУ, называемая подканалом (в определенном смысле они аналогичны статическому и динамическому описателям групп). Описание ВУ содержит адрес подканала, а также некоторые другие статические данные, используемые драйверами; подканал — динамически изменяемую рабочую память обмена с конкретным устройством.

Система ввода-вывода обеспечивает возбуждение обмена по требованию задачи или ВУ. Супервизор ввода-вывода принимает от задачи требование на обмен, организует очередь

к ВУ в случае его занятости (очередь приоритетная, приоритет задается задачей-инициатором обмена), реализует необходимые действия по завершению обмена. Число ВУ в системе не ограничено.

Драйвер ВУ принимает и обрабатывает прерывания от ВУ, инициирует операции ввода-вывода, запрошенные задачей, выполняет ориентированные на конкретное ВУ действия при истечении тайм-аута.

Состав драйверов определяется прикладной системой. Предполагается, что они разрабатываются пользователем и объединяются с ДС-86 на основе достаточно простых интерфейсных соглашений.

Задача для инициирования обмена обращается к ДД «Обмен», которая обрабатывает запрос с последующим инициированием драйвера. Драйвер инициирует операцию на устройстве. После ее завершения он вновь получает управление по прерыванию от ВУ для выполнения следующей операции или завершения ввода-вывода. По завершении обмена драйвер передает управление программе «Терминатор», входящей в состав супервизора ввода-вывода. Она выполняет необходимые действия, связанные с окончанием обмена и, если очередь к ВУ пуста, входит в задачу.

При обращении к ДД «Обмен» задача может потребовать от супервизора выполнения ряда функций:

по окончании обмена возбудить указанную задачу;
перевести задачу-инициатор обмена до его завершения в состояние «ждет»;
в случае занятости ВУ поставить запрос на обмен в очередь к ВУ;
поместить по указанному адресу код возврата обмена.

Блок инициализации подготавливает ДС-86 к работе, заносит в системные ячейки начальные значения и формируя векторы прерывания от таймера и по команде ЕМТ. Он содержит ДД «Создать систему», «Создать группу», «Создать задачу», «Подключить ВУ», позволяющие сообщать ДС-86 сведения о начальной конфигурации целевой системы или об изменении конфигурации. Эти сведения формируются пользователем и записываются в так называемый модуль описания системы (аналог конфигурационного модуля в RMX/80) [3]. В состав этого модуля входят статические описатели групп, описатели ВУ, таблицы адресов статических описателей групп и описателей ВУ. Среднее число исполняемых при переключении команд для ДС-86 равно 150.

Для работы ДС-86 в максимальной комплектации требуется около 1,5 Келов ПЗУ, а также 65 слов ОЗУ (с учетом 25 слов системного стека). Дополнительные ресурсы памяти полностью зависят от прикладной системы. Если она использует Т задач, разбитых на Р групп, и взаимодей-

ствует с У ВУ, то дополнительные расходы памяти составят по ПЗУ $2P+T+6Y$ слов, то ОЗУ — $16P+Y+11Y$ слов. Кроме того, каждая группа имеет свой стек, объем которого определяется пользователем.

Минимальная версия ДС-86 была скомпьютана для двух контроллеров, реализованных на микроЭВМ «Электроника С5-41» и предназначенных для автоматического управления технологическим оборудованием. В этих контроллерах ДС-86 занимает 240 слов ПЗУ и 32 слова системного ОЗУ (включая 20 слов стека системы). Система обслуживает только одно ВУ и содержит две группы (на восемь и одну задачу), допускается только одна частотная задача (остальные сервисные возможности не предоставляются).

Телефон 291-66-72, Ленинград

ЛИТЕРАТУРА

1. Гальперин М. П., Гинтер А. В., Городецкий В. В. и др. Одноплатные микроконтроллеры «Электроника С5-41» // Микропроцессорные средства и системы. — 1984. — № 2. — С. 20—23.
2. PDP-11 Handbook // Digital Equipment Corp.— 1979.— Section 4 System Software.— Chapt. 5.— P. 651—664.
3. Крамфус И. Р., Новик А. Г., Перцов Е. Е. Обзор операционной системы реального времени RMX/80. В сб.: Операционные системы реального времени для микроЭВМ.— М., 1984.

Статья поступила 17.01.89

Реклама

Внимание владельцев персональных компьютеров типа IBM PC/XT; AT Производственный кооператив «Модуль» предлагает базу данных по интегральным микросхемам советского и импортного производства. База содержит информацию по шести тысячам ИМС и поставляется на 15 магнитных дисках, 5,25 дюйма емкостью 360 Кбайт.

Информация записана на гибких магнитных дисках исполнителя. 167024, Сыктывкар, а/я 480, кооператив «Модуль»

Телетайп: 681147 БИС

Телефон: 2-64-54

ЯЗЫК ПРОГРАММИРОВАНИЯ FORTH-88R

Несмотря на общий рост популярности языка FORTH, отношение к нему специалистов остается крайне противоречивым. Определились две полярные точки зрения, согласно одной из которых FORTH — это оптимальный язык программирования, а согласно другой — типичная ветвь развития компьютерной лингвистики. На наш взгляд, эти мнения отражают односторонние подходы к языку, абсолютизируя его преимущества или недостатки. Практическая задача состоит в том, чтобы развивать преимущества этого языка как инструмента программирования для растущего парка отечественных микро- и мини-ЭВМ.

Характеристики языка в первую очередь определяются его структурой. В FORTH реализована структура «шитого» кода, т. е. дерева вызова подпрограмм. Эта его особенность обуславливает компактность и интерактивность языка — качества не заменимые при создании встраиваемых систем малых машин. Основная процедура языка — определение новых функциональных единиц — слов, компилируемых из элементов базового словаря или из ранее определенных программистом слов. Весь состав словаря служит для интерпретации набора символов на входе так называемого «входного потока». При этом интерпретатор может работать и в режиме компиляции. Параметры от слова к слову передаются через стек (стековый механизм наиболее удобен при древовидной записи алгоритма).

Словарь языка FORTH представляет собой связанный список, в узлах которого размещены «словарные статьи» (рис. 1). Каждая из них состоит из четырех полей: имени — идентификатора, связи — организатора списка, кода и параметров — для непосредственной организации шитого кода. Наиболее предпочтительна форма косвенно-шитого кода [1]. В этом случае в поле кода находится адрес машинных команд, обрабатывающих поле параметров; в поле параметров —

адреса полей вызываемых слов. В результате обеспечивается независимость структуры словарной статьи от аппаратной реализации языка и полная стереотипность работы компиляционного механизма.

Однако нельзя забывать, что любой язык, в частности язык программирования, — одна система, а множество контекстуально задаваемых систем с гибкими структурами элементарных единиц. Например, поля имени и связи словарной статьи языка необходимы при интерпретации входного потока и компиляции новых слов, но при интерпретации шитого кода в машинные команды их присутствие в словарной статье становится не только ненужным, но и вредным, так как приводит к неоправданно большому расходу оперативной памяти. Указанный дефект существующих версий языка FORTH может быть устранен, если ввести в словарную статью дополнительное поле, содержащее ссылку на поле параметров. В этом случае можно разделить словарную статью на две части: одну, содержащую поля имени, связи и ссылки, и другую — поля кодов и параметров. Их можно разместить в различных областях памяти, что позволяет удалять и повторно использовать содержимое первой области памяти.

Списковая структура словаря в стандарте языка предусматривает ссылку из поля связи на поле имени предыдущего слова. При этом существует неявная ссылка из поля имени на поле связи, обусловленная переменной длиной поля имени. В версии FORTH-88R поле связи, расположенное перед полем имени, указывает на поле связи другого слова (рис. 2). Таким образом, число связей уменьшается вдвое, скорость поиска и надежность работы FORTH-машины увеличиваются. Поле имени не имеет фиксированного размера, поэтому между ним и полем связи вставляется поле ссылки, позволяющее легко реализовывать извест-

ный способ уменьшения объема шитого кода и текста программ с помощью векторного поля кода [1]. Применение векторного поля кода в словарной статье означает появление в языке механизма слабой типизации. В результате создаются слова с несколькими точками входа, удобные для использования в прикладных задачах, например экономического характера. Недостатком существующих реализаций этой концепции следует считать наращивание векторного поля кода и поля параметров в сторону увеличения адресов, что приводит к конфликтам при попытке одновременного увеличения размеров поля параметров и числа полей кодов. В предлагаемой версии поля кодов могут наращиваться в сторону уменьшения адресов, а поля параметров, как обычно, в сторону увеличения.

Следующее изменение стандарта языка заключается в упорядочении сущности понятия «входной поток». По стандарту входной поток может располагаться в буферах двух типов: терминала, адресуемого переменной ТВ, и блока внешней памяти, адресуемой переменной ВЛК, указывающей номер блока. Если содержимое переменной ВЛК равно нулю, то происходит обращение к буферу терминала, иначе — к блоку внешней памяти. Значение рассматриваемой литеры во входном потоке задается с помощью смещения, хранящегося в переменной IN.

Такая структура входного потока препятствует расширению границ его использования, например интерпретации произвольного буфера, текстового файла и т. д. Эти ограничения можно снять, если рассматривать входной поток как текстовый буфер, ограниченный двумя переменными: IN — адресом интерпретируемой литеры, TILL — адресом конца буфера. Предложенное изменение позволяет усовершенствовать буферный пул. Прикладной программист работает с буферным пулом с помощью трех слов: BLOCK — вызов адреса оперативной памяти, по которому блок размещен системой; BUFFER — обращения к внешней памяти не происходит, если в пуле нет блока с данным номером; UPDATE — отмечает последний использованный блок как измененный (впоследствии запишется во внешнюю память). Эффективная работа с этими словами затрудняется тем, что система гарантирует расположение блока по заданному адресу только при однократном обращении к слову BLOCK. Для надежной работы программы приходится обращаться через BLOCK любое обращение к внешней памяти.

Повысить эффективность работы программ с внешней памятью можно, предусмотрев в программном интерфейсе четыре запроса: IN-B — выдать

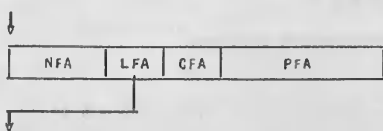


Рис. 1. Организация словаря по стандарту FORTH-83; NFA — поле имени (2... 32 байт); LFA — поле связи (2 байт); CFA — поле кода (2 байт); PFA — поле параметров (размер произвольный)

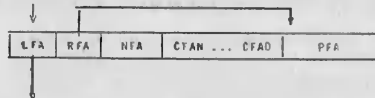


Рис. 2. Организация словаря FORTH-88R; LFA — поле связи (2 байт); RFA — поле ссылки на поле параметров (2 байт); NFA — поле имени (2... 32 байт); CFAN...CFAO — векторное поле кода (размер произвольный); PFA — поле параметров (размер произвольный)

адрес блока с заданным номером (при его отсутствии в буферном пуле происходит обращение к внешней памяти); OUT-B — освободить блок с последующей записью во внешнюю память; NEW-B — выдать адрес блоку без предварительного считывания из внешней памяти; FR-B — освободить блок без перезаписи во внешнюю память.

При подаче запросов IN-B и NEW-B блок «замораживается» в буферном пуле и освобождается только после поступления определенного числа запросов OUT-B и FR-B, равного поступившим запросам IN-B и NEW-B.

Описанная концепция буферного пула адаптирована к принятой структуре входного потока. Возможна декомпозиция ключевых слов пула существующего стандарта BLOCK, BUFFER и UPDATE на вновь определенные слова IN-B, OUT-B, NEW-B и FR-B.

Это суждение может быть проиллюстрировано примером решения параллельных задач. При использовании слов типа BLOCK возможны два варианта: в первом для каждой задачи организуется отдельный буферный пул, во втором — накладывается запрещение на прерывание работы во время действия операторов типа BLOCK. Недостаток первого варианта — повышенный расход оперативной памяти и сложность системной поддержки; второго — значительный рост инерционности системы, не приемлемый для многих практических случаев. Применение слов типа IN-B устраняет дефект второго варианта, так как появляется возможность работы с сохранением адреса буфера.

Наиболее явный недостаток существующего стандарта языка FORTH — отсутствие прислешей для численных расчетов арифметики. Для проведения расчетов типа $5237 \times 0,3$ приходится 5237 умножить на три и результат разделить на 10. В языке есть широкий набор операций с целыми числами размером два байта и ряд операций с целыми числами в четыре байта. Однако он неполон, так как нет операций умножения и деления чисел длиной четыре байта.

В предлагаемой версии языка кроме целочисленной арифметики, организованной в стеке данных, предусмотрена вычислительная арифметика, операции которой ориентированы на отдельный стек с элементом размером четыре байта. Конкретное представление числа может варьироваться в зависимости от характера прикладной задачи: с плавающей и фиксированной точками, двоично-десятичные (восемь десятичных знаков).

Сфера применения языка FORTH ограничивается жесткой ориентацией на 16-битную аппаратуру, в которой при реализации средних и крупных задач не хватает адресного пространства размером 64 Кбайт. Переход на

32-битную адресацию существенно уменьшит быстродействие и вдвое увеличит потребность в памяти.

Но можно ввести несколько страниц памяти с 16-битной адресацией, и в каждой странице разместить часть системы. Среди них страницы, содержащие машинные команды, заголовки словарных статей (поля связи, ссылки имени), штигтовый код, переменные и буферы. В некоторых страницах можно установить не побайтную, а пословную адресацию, что позволит увеличить в два раза адресное пространство. Эти приемы легко осуществимы в языке FORTH-88R. Стандарты языка FORTH не имеют стереотипной процедуры работы с файловой системой, что ограничивает переносимость написанных на них программ, а также возможность применения обширных массивов данных. В версии FORTH-88R использована привязка файлов ОС к заданному интервалу номеров блоков. Общее виртуальное пространство этих номеров равно 2^{15} , размер одного блока, принятый в FORTH, равен 1 Кбайт, поэтому допустимый объем базы данных весьма ограничен. Вместе с тем, унификация процедуры обращения к файлам снимает основные трудности в организации взаимодействия с внешней памятью при переносе прикладных программ на иные аппаратные средства. FORTH-88R реализован на ПЭВМ «Электроника 60» и «Искра 226».

В версии для «Электроники 60» обеспечена возможность генерации грузочного модуля с широким набором параметров (работа под управлением ОС RT-11 или без нее, с использованием набора команд EIS или без них и т. д.). Благодаря изменению алгоритма интерпретации входного потока скорость его трансляции возросла в три раза по сравнению с версией FIG-FORTH. Оптимизация слов ядра этой стандартной версии сократила время выполнения задач на 20%. Средний объем ядра грузочного модуля составляет 8...9 Кбайт (в зависимости

от параметров генерации). Ядро может расширяться с помощью пакетов «экранный редактор», «файловая система», «плавающая точка», «целевой компилятор», «параллельные задачи», «отладчик» и «ассемблер».

Реализация FORTH-88R для «Искры 226» выполнена на основе кодового штигтового кода в двух страницах памяти по 64 Кбайт каждая. В одной из них хранятся машинные команды и буферный пул, а другая занята, в основном, словарем. По сравнению с существующей версией для данной ПЭВМ, разработанной ИСЭП АН СССР в г. Ленинграде, предлагаемая версия при равных скоростях выполнения задач и объеме буферного пула увеличивает скорость трансляции в три раза, а допустимый объем загружаемых программ — в 2,5 раза. FORTH-88R поддерживает работу с файлами, совместимыми с БЕЙСИКом по формату каталога.

Язык FORTH-88R, расширенный системами бланкового ввода-вывода и реляционной базой данных, позволяет применять ПЭВМ не только в задачах управления аппаратурой, но и в нетрадиционной для них области экономических задач АСУП [2]. В настоящее время завершается адаптация FORTH-88R к ПЭВМ «Искра 1030», «Электроника 85».

344064. Ростов-на-Дону, пер. Технологической, 8, ОЭЗ НПО «Атомкотломаш»; тел. 34-81-88

ЛИТЕРАТУРА

1. Баранов С. Н., Ноздрин Н. Р. Язык ФОРТ и его реализации. — Л.: Машиностроение. — 1988, 157 с.
2. Шмалько С. Г., Винников А. В. Человеко-машинный диалог при решении экономических задач // Машиностроитель. — 1988. — № 9. — С. 30—31

Статья поступила 24.10.88

УДК 681.3.06

И. Б. Кириченко

ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ДЛЯ СОЗДАНИЯ АВТОМАТИЗИРОВАННЫХ РАБОЧИХ МЕСТ

В целях снижения затрат на проектирование, внедрение и сопровождение автоматизированных систем широко практикуется создание АРМ на основе типовых проектных решений (ТПР) с использованием настроек на условия конкретного производства. Для облегчения настройки предлагается инструментальная система на ПЭВМ ЕС1840, рассчитанная на АРМ различных категорий пользователей, задействованных в гибком автоматизированном производстве (ГАП).

Особенности системы

С точки зрения программной реализации система представляет собой ряд унифицированных компонентов (модулей), обеспечивающих диалог с пользователем, функциональную подсистему, взаимодействующую с прикладными пакетами (рис. 1).

Система включает в себя комплекс инструментальных средств, поддерживающих настройку на различные

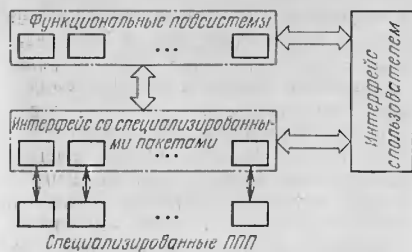


Рис. 1. Общая структура системы

проблемно-ориентированные области. Инженер-программист компонует модули и специализированные пакеты в соответствии с функциональным назначением АРМ, а непрограммист-профессионал (в дальнейшем — пользователь) применяет подготовленную таким образом систему в качестве инструмента в своей работе, настраивая ее на конкретные приложения.

Работа с системой не требует специальной подготовки. Трудности могут возникнуть лишь в процессе использования специализированных пакетов, подключаемых к системе, и определяются простотой или сложностью этих пакетов, а не самой системы.

Перед началом работы пользователь выбирает нужный режим с помощью графического меню.

Предположим, что гибкая автоматизированная система (ГАС) состоит из технической подготовки производства, технико-экономического планирования, оперативного управления производством, бухгалтерского учета, управления материально-техническим снабжением, сбыта и реализации готовой продукции, и рассмотрим, как можно использовать предложенный инструментарий в технико-экономическом планировании, а именно — каким образом с помощью имеющихся в инструментальной системе средств создать АРМ технолога-аналитика.

Типовые технологические процессы (ТПП) фиксируются в технологической документации, включающей следующие разделы:

ведомости применимости (область применения документации, объяснения некоторых терминов, особенностей изготовления изделия и т. д.);

указания по технологии процесса производства (наиболее общие требования по технологии, оборудованию и материалам, контролю качества, безопасности проведения работ);

указания по организации труда (рекомендации по составу бригады, расстановке и распределению работ между членами бригады, общие требования по квалификации рабочих и расценкам);

таблица трудовых затрат (данные о квалификации рабочих, нормах времени, расценках и т. д.);

основные технико-экономические показатели (трудоемкость работ, средний разряд рабочих, средняя заработная плата);

материальные затраты (сведения о необходимых машинах, оборудовании, инструменте, инвентаре, а также основных узлах изготавливаемого изделия);

маршрутные карты (содержание операций по переходам, оборудование, оснастка, квалификация рабочих, выполняющих операции, а также нормы времени на операции);

маршрутная карта может быть представлена таблицей, каждая строка которой содержит информацию, относящуюся к отдельной технологической операции; в некоторых случаях ее можно рассматривать как график проведения работ;

операционные карты — эскизы, поясняющие содержание операций;

чертежи заготовок;

технологические схемы;

программы для станков с ЧПУ [1].

Таким образом, технологическая документация — это графические, текстовые и табличные документы, определяющие технологический процесс изготовления изделия [2].

Инструментальные средства для работы с технологической документацией (рис. 2).

Графические документы формируются с помощью систем иллюстративной графики PC PAINT, PC PAINT BRUSH, графических процессоров ВГРАФ и ВИКОНТ, разработанных в ВЦ АН СССР; разнообразные виды текстовой документации — с помощью систем подготовки текстов (текстовых процессоров) MS WORD, ПОЛИТЕКСТ, ЛЕКСИКОН; табличные документы — с помощью систем управления базами данных dBase III или СПЕКТР. Система СПЕКТР удобна тем, что основана на фреймовом подходе к организации объектной базы.

Информация фрейма может рассматриваться с разных точек зрения, т. е. иметь разные виды. Например, для того чтобы получить информацию о графике выполнения технологического процесса, удобно использовать табличный вид фреймового объекта «маршрутная карта»; а для просмотра информации об отдельной операции лучше пользоваться трафаретным видом того же объекта [3].

Различные виды документов, относящихся к одной и той же технологии

предусматривается выделение необходимых ресурсов, широкое внесение изменения в документацию в соответствии с фактическими условиями, т. е. корректировка затрат труда и материальных ресурсов.

3. В соответствии с разработанной технологией составляется планировка веха, размещается оборудование, устанавливаются места хранения деталей, узлов, инструмента и т. н.

4. Помимо выбора технологии изготовления каждого вида изделия задаются рабочие места для каждой операции и относительное время начала ее выполнения. Эти характеристики определяют задание, выполняемое на основе оптимального календарного плана (расписания) [2].

Операционные карты
Чертежи заготовок
Технологические схемы

Рис. 2. Инструментальные средства для работы с технологической документацией

(каждый документ в машинном представлении составляет отдельный файл), объединяются общим именем, под которым эта технология (т. е. документация, описывающая некоторый технологический процесс) далее фигурирует в системе. Таким образом, создается база различных технологий, которую легко модифицировать: удалять технологию, добавлять новые, корректировать содержимое документов. База ведется системой: пользователю необходимо задать имя новой технологии, выбрать технологию из меню и определить (путем нажатия одной из функциональных клавиш) действия над ней. Табличные документы в системе задаются по правилам, о которых пользователь может узнать, ознакомившись с документацией к системе или нажав клавишу «подсказка» в процессе работы. Эти ограничения обеспечивают санкционированный доступ к информации таблиц из других модулей.

Привязка технологической документации к фактическим условиям производства

Для использования ТПП на конкретном предприятии необходимо скорректировать предусмотренные технологической документацией размещение оборудования, транспортные потоки, затраты труда и материальных ресурсов и др., т. е. осуществить привязку технологической документации к фактическим условиям производства.

Схематично работу по привязке технологической документации можно представить в виде нескольких самостоятельных этапов.

1. Прежде всего необходимо сопоставить имеющиеся варианты технологических процессов и выбрать из них наиболее удовлетворяющий некоторому заранее заданному критерию оптимальности. В качестве критериев могут быть выбраны максимальный валовый доход, минимальный объем затрат времени, трудовых и (или) материальных ресурсов, приближенное к более производительной поточной и др.

2. Для выбранного технологического процесса предусматривается выделение необходимых ресурсов, широкое внесение изменения в документацию в соответствии с фактическими условиями, т. е. корректировка затрат труда и материальных ресурсов.

3. В соответствии с разработанной технологией составляется планировка веха, размещается оборудование, устанавливаются места хранения деталей, узлов, инструмента и т. н.

4. Помимо выбора технологии изготовления каждого вида изделия задаются рабочие места для каждой операции и относительное время начала ее выполнения. Эти характеристики определяют задание, выполняемое на основе оптимального календарного плана (расписания) [2].

Работа со специализированными пакетами

В процессе привязки технологической документации и разработки календарного плана технолог-аналитик должен использовать различные пакеты, ориентированные на выполнение некоторых специальных функций. Например, выбор технологии, оптимальной с точки зрения максимизации валового дохода, сводится к канонической задаче линейного программирования; планировка цеха (участка) — к транспортной задаче; составление оптимального календарного плана — к одной из задач теории расписаний и т. д. Система подключает необходимые пакеты и обеспечивает удобное для пользователя обращение к ним (удобство достигается ценой значительного усложнения процесса настройки системы на конкретные приложения).

Подготовка входных данных и обработка полученных результатов для представления в удобном виде осуществляются самой системой без участия пользователя.

Некоторые режимы работы технолога-аналитика

Помимо модулей взаимодействия со специализированными пакетами (эти модули представляют собой, по сути, некоторый вспомогательный механизм) для организации работы технолога-аналитика в состав системы включены функциональные подсистемы.

Работа с пиктограммами (условными обозначениями). Каждому виду оборудования, используемому в цехе, ставится в соответствие некоторая пиктограмма (набор таких пиктограмм может быть подготовлен самим технологом или предварительно создан инженером-программистом с помощью графического процессора ВГРАФ). Система позволяет просматривать существующие пиктограммы, создавать новые, изменять и удалять пиктограммы, а также обращаться за получением дополнительной информации к базе данных «Основные фонды», где содержатся сведения об оборудовании, имеющемся в цехе.

Работа со схемами цехов. На схеме еще не фиксируется размещение оборудования в цехе, т. е. она представляет собой лишь некий шаблон, который будет использоваться на этапе составления планировки цеха. Схемы создаются с помощью графического процессора ВГРАФ их как пиктограммы, можно просматривать, изменять, удалять, задавать некоторые текстовые комментарии к ним.

Работа с планами цехов. План цеха — это схема цеха с расставленными на ней пиктограммами, соответствующими тому или иному оборудованию. Пиктограмма, привязанная к определенному месту на схеме цеха, становится условным обозначением рабочего места. Каждому рабочему месту присваивается номер (этот номер задает

пользователь в режиме диалога с системой).

Создание плана цеха — одна из наиболее сложных задач. Здесь целесообразно воспользоваться специализированным пакетом для решения транспортной задачи, позволяющим получить рекомендации по размещению оборудования, оптимальному с точки зрения транспортных потоков. Оборудование может (и должно) размещаться на основании технологических схем, имеющихся в документации, например технологических схем сборки. Независимо от способа, применяемого технологом, при составлении плана цеха, разумно воспользоваться советом экспертной системы, дающей рекомендации по наилучшей для конкретных условий планировке цеха.

Планы цехов в совокупности образуют базу, с которой можно работать по аналогии с базами пиктограмм и схем цехов, т. е. создавать планы, редактировать (изменять) их, просматривать, удалять, задавать текстовые комментарии к планам.

Работа в режиме модельного исполнения. Используя этот модуль, технолог-аналитик имеет возможность получить на экране дисплея наглядное представление о производственном процессе. Выбрав какой-то план цеха (как уже было сказано, их может быть несколько) и задав некоторую проекцию календарного плана (т. е. конкретное время начала работы, календарный план из числа имеющихся в соответствующей базе), технолог может для любого момента времени отслеживать ситуацию в цехе.

При работе в режиме модельного исполнения на экране выделяются рамками три основных поля: плана цеха, сообщений, временной шкалы (рис. 3). Поле плана цеха — главное, оно всегда занято выбранным планом. В поле сообщений, как правило, выводится информация, характеризующая ситуацию на конкретном рабочем месте в определенный момент времени. Поле временной шкалы располагается в нижней части экрана и содержит собственно временную шкалу и информационное поле. Вдоль временной шкалы можно двигаться с помощью стрелок, при этом

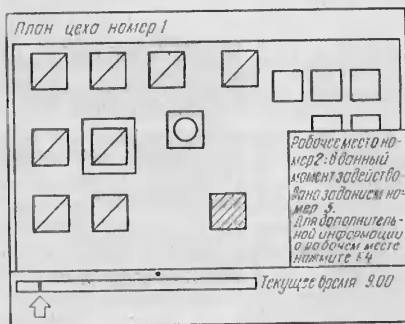


Рис. 3. Использование поверхности экрана при работе в режиме модельного исполнения

в информационном поле будет указываться соответствующий момент времени.

Варианты работы в режиме модельного исполнения.

Существует несколько вариантов работы в этом режиме. Выбрав интересующий его момент времени, пользователь нажимает клавишу «исполнение» и попадает в поле сообщений, куда выводится список заданий, которые в данный момент выполняются в цехе. Если пользователь выбирает одно из них и нажимает клавишу «исполнение», то пиктограммы на плане, соответствующие рабочим местам, занятым в данный момент времени данным заданием, начинают мигать.

Другой вариант работы заключается в том, что для любого момента времени можно выбрать на плане с помощью специального маркера пиктограмму и, нажав клавишу «исполнение», получить в поле сообщений информацию о том, какое задание в этот момент выполняется на соответствующем рабочем месте, кто именно работает здесь (т. е. в некотором роде динамические параметры рабочего места), что это за оборудование (т. е. статические параметры) и т. д. При этом происходит обращение к различным базам данных, в том числе таким, как «Основные фонды», «Трудовые ресурсы» для создания которых в системе также имеются специальные модули.

Еще один вариант состоит в следующем: задается некоторый масштаб времени, и ситуация в цехе отслеживается на протяжении всей длины расписания, но в указанном масштабе. Например, если масштаб 1 мин: 1 ч, а длина расписания — 8 ч, то за 8 мин пользователь, выбрав какое-то из заданий, может проследить, какие рабочие места будут последовательно заняты этим заданием: информация будет отображаться в главном поле на плане цеха миганием соответствующих пиктограмм.

Можно представить себе еще несколько вариантов работы в режиме модельного исполнения и еще несколько модулей, необходимых для создания действительно удобного АРМ технолога-аналитика. Особых затруднений для расширения набора модулей нет, так как вновь созданные модули легко включаются в систему или даже формируются с помощью уже имеющихся в системе блоков.

Оперативное управление производством

Результаты работы технолога-аналитика могут быть использованы в управлении технологическим процессом в реальном масштабе времени. Архитектура ПЭВМ семейства ЕС1840 позволяет организовывать такие автоматизированные системы управления с помощью дополнения ПЭВМ специальными электронными модулями сопряжения с управляемым технологическим процессом [5]. В каждый конкретный

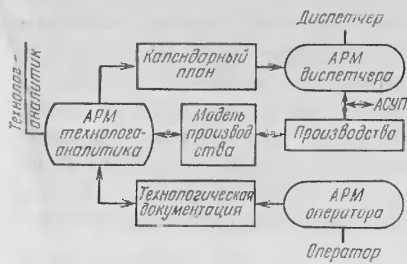


Рис. 4. Общая схема отношений между участниками процесса производства и компонентами информационного обеспечения ГАС

момент времени данные фактического состояния производственного процесса сопоставляются с соответствующими данными календарного плана, опреде-

УДК 681.3

Л. Г. Осовецкий, А. А. Штрик

АНАЛИЗ ОПЫТА ВНЕДРЕНИЯ КРУПНЫХ ТЕХНОЛОГИЧЕСКИХ КОМПЛЕКСОВ СОЗДАНИЯ ПРОГРАММ И МЕТОДОВ ИХ ВЫБОРА

Возрастающие требования к качеству программного обеспечения и необходимость сокращения затрат на их разработку заставляют руководителей программных проектов (ПП) обращать особое внимание на выбор инструментальных средств и серьезное обоснование разработок новых технологических комплексов (ТК).

Анализ опыта внедрения и эксплуатации ТК показывает, что наиболее существенными факторами, влияющими на выбор технологии, являются характеристики предметной области внедрения ТК и цели, которую требуется достичь в результате внедрения. Объективный предметно- и целеориентированный подход к оценке ТК позволяет с системных позиций выделить доминирующие факторы и определить их весовое значение в конкретной ситуации, сложившейся у пользователя, и влияние на цели, поставленные перед выбираемой технологией.

Основная цель статьи — на примере конкретной системы автоматизации разработки программного обеспечения (САРПО) РУЗА [1] проанализировать проблемы, возникающие при внедрении ТК, сформулировать рекомендации по организации процесса внедрения, а также рассмотреть подходы к выбору таких ТК в зависимости от реальных характеристик создаваемых программ и условий их разработки.

Анализ опыта внедрения САРПО РУЗА. Это настраиваемая кросс-система автоматизации разработки сложных комплексов программ, работающих преимущественно в реальном времени на различных типах встроенных ЭВМ; функционирующая на ЕС ЭВМ и обеспечиваю-

ются отклонения и заменяется задающая информация. Эта работа выполняется с помощью автоматической системы управления производством под контролем диспетчера.

Общая схема управления технологическим процессом производства на основе взаимосвязей участников производства и компонентов информационного обеспечения ГАС приведена на рис. 4.

117236, Москва, Ленинские горы, МГУ, 2-й уч. корп., фак. ВМиК, каф. алг. языков.; тел 939-18-80

ЛИТЕРАТУРА

1. Справочник технолога-машиностроителя: В 2 т. Т. 1.— М.: Машиностроение, 1972.— 694 с.

2. Справочник металлиста: В 5 т. Т. 5. / Под ред. Б. Л. Богуславского.— М.: Машиностроение, 1978.— 673 с.

3. Брябри В. М. Программное обеспечение персональных ЭВМ.— М.: Наука, 1988.— 272 с.

4. Макаров И. М. Робототехника и гибкие автоматизированные производства: настоящее и будущее // В кн.: Информатика и научно-технический прогресс.— М.: Наука, 1987.— С. 140—160.

5. МикроЭВМ: В 8 кн.: Практическое пособие / Под ред. Л. Н. Преснухина. Кн. 5. Персонально-профессиональные ЭВМ.— М.: Высшая школа, 1988.— 143 с.

Статья поступила 26.12.88

щая автоматизацию всех основных этапов создания программ, разрабатываемых на автокоде и макроязыке в рамках ПРОМЕТЕЙ-технологии [2]. Все необходимые сведения по системе приведены в [1, 3]. Ее разработка завершена в 1984 г. В течение последующих лет она серийно тиражировалась и поставлялась специализированным фондом алгоритмов и программ ПО АрмВТИ (Ереван), по прямым договорам разработчиков с пользователями.

К настоящему времени в фонде существуют три версии САРПО РУЗА, отличающиеся последовательным наращиванием функциональных возможностей и централизованной поставкой компонентов САРПО РУЗА, настроенных на наиболее распространенные типы ЭВМ («Электроника 60», микропроцессоры К580, КР1810ВМ86). По данным фонда система РУЗА поставлена более чем в 100 организаций страны.

Для объективного анализа возникающих трудностей с целью определения перспектив и оценки направлений развития современных ТК сознательно выбран один из первых примеров внедрения САРПО РУЗА с недостаточно высокой степенью отработки и небольшим опытом организации работ по внедрению и настройке. Первая версия включала 12 подсистем, охватывающих операции по разработке спецификаций и программ, их автономной отладке с помощью интерпретированного выполнения, структурного контроля, статического комплексирования, документирования, выпуска машинных носителей, а также средства диалоговой работы, базу данных проектирования, контроль за разработкой, средства автоматизированной настройки на характеристики ЭВМ и ПП.

Трудоёмкость настройки САРПО РУЗА на систему команд конкретной встроенной ЭВМ составила около 12 человеко-месяцев с учетом изучения и освоения системы, а также обнаружения и исправления в процессе работы определенного числа ошибок в САРПО [4].

В качестве конкретного ПП для внедрения САРПО РУЗА была выбрана разработка, имеющая перспективу развития по использованию различных видов встроенных ЭВМ. Результаты работы показали, что по сравнению с ранее применявшейся технологией работы на базе встроенной ЭВМ внедряемые технологические средства САРПО РУЗА повысили эффективность разработок и качество получаемых программ. Основным эффектом был достигнут за счет следующих факторов;

обязательности проведения работ по разработке спецификаций на программы (в большом объеме проектируемого ПП четко формулируется общая структура комплекса и управление разработкой);

использования эффективно организованного диалогового доступа к технологической ЭВМ, обеспечившего существенный рост пропускных способностей по сравнению с их последовательным прохождением на функциональной ЭВМ в прежней технологии;

наличия централизованной базы данных проектирования с развитыми средствами доступа к хранимой информации и работы с ней, средств интерпретации для проведения автономной отладки программ и группы программ на технологической ЭВМ без выхода на встроенную функциональную ЭВМ, а также средств контроля структурного построения отдельных программ и комплекса программы в целом;

использования средств контроля за ходом разработки для выявления узких мест в процессе проектирования программ, позволяющих информировать об этом все категории пользователей;

возможности применения общей технологии, поддерживаемой единичными инструментальными средствами для разных типов встроенных ЭВМ.

Наряду с работами по внедрению был проведен эксперимент по исследованию возможности использования ранее созданного задела программ, написанного на языке программирования, отличном от принятого в системе. Для этого разработан препроцессор перевода в автокод САРПО РУЗА. Однако проверка этого режима работы показала, что появление программ, написанных на другом языке программирования, нарушает регламентированный технологический процесс разработки, поддерживаемый в САРПО РУЗА. В частности, программы, разработанные на нештатном языке, не смогли пройти отладку и документирование средствами САРПО РУЗА. Таким образом, эксперимент подтвердил, что новая технология может быть эффективно внедрена только на вновь разрабатываемых ПП: попытки совместного использования прежних и вновь внедряемых технологий приводят к большим затратам.

Пользователь заинтересован в конечном ре-

зультате и проверке своих решений, поэтому его контакты с разработчиками ТК необходимы. Категория «пользователь» включает в себя не одиночного пользователя, а весь спектр потенциальных потребителей технологии, поэтому трудности согласования и формирования единого мнения становятся значительными. Важно четко организовать это взаимодействие. Его основные формы:

разработка и согласование технических заданий на ТК и его отдельные компоненты;

оперативное и квалифицированное рассмотрение предложений, возникающих в процессе разработки ТК;

выделение пользователем специалиста (с необходимыми полномочиями), ответственного за технологию разработки в проектируемом комплексе программ; участие пользователя в испытаниях ТК различного ранга.

Организационные и технологические проблемы внедрения: недостаточное энергичное применение правил структурного построения и принципов проектирования в широкой практике разработки программ, как следствие — сужение функциональных возможностей применяемых ТК; недооценка трудоемкости создания интерпретаторов встроенных ЭВМ и настройки системы отладки; перерасход ресурсов ЭВМ; отсутствие контроля за полнотой и достоверностью учетных данных при работе с ТК; психологические проблемы внедрения.

Подходы к выбору технологических комплексов.

Методология выбора и оценки ТК должна базироваться на сборе и анализе данных о результатах проведения аналогичных работ на достаточно большом числе программных проектов [5]. Получение таких данных затруднено из-за специфики разработки различных классов программ и отсутствия практики сбора и анализа результатов выбора и оценки ТК, а также обмена информацией между разработчиками о преимуществах и недостатках конкретных ТК в реальных условиях их эксплуатации.

Данные по 87 различным ПП, полученные на основе специальных опросных анкет, поступили в распоряжение авторов в виде экспертных оценок и позволили определить перечень наиболее употребляемых параметров и показателей, характеризующих технологию и ТК. Они могут быть условно разделены на три следующие категории:

характеристики предметной области внедрения технологии (табл. 1);

целей, которые ставятся перед внедряемой технологией;

технологических комплексов.

В свою очередь, характеристики предметной области составляют три группы. Перечень параметров области внедрения технологии програм-

Таблица 1

Характеристики предметной области внедрения технологии

Наименование	Единица измерения	Проекты выборки с указанными целями	
		Число	%
Ресурсы разработки			
Максимальное время разработки ПП	мес., год	67	77
Объем разработки ПП:			
уникальных собственной разработки	строка на исходном языке программируемая команда, байт	48	55
привлекаемых, адаптируемых, заимствованных		52	60
Число сотрудников, принимающих участие в разработке	чел.	28	33
Среднемесячная заработная плата на одного сотрудника без учета накладных расходов	руб.	12	14
Усредненные накладные расходы на предприятии	%	12	14
Затраты машинного времени на разработку ПО на технологических и объектной ЭВМ	ч.	11	11
Стоимость моделирующего отладочного стенда	руб.	10	11
Динамика изменения требований к одному программному заказу	шт./мес.	12	14
Параметры создаваемых программ			
Тип программного обеспечения	—		
Характеристики объектной (функциональной) ЭВМ, на которой должно функционировать разрабатываемое ПО	—		
Временные характеристики программных заказов	—		
Объем вспомогательного ПО, создаваемого в процессе разработки функционального (объектного) ПО	команда, строка, байт		
Ожидаемая степень использования памяти объектной ЭВМ	%		
Предполагаемая серийность разрабатываемого ПО	шт.		
Среда разработки			
Типы и число технологических ЭВМ	шт.		
Используемые средства автоматизации разработки ПО			
Время использования средств автоматизации, технологических систем	год, строка, команда, байт		
Объем существующего технологического обеспечения			
Состав существующего технологического программного обеспечения	—		
Степень удовлетворения требований разработчиков имеющимися технологическими средствами	—		
Среднее число лет работы специалиста в области разработки ПО	год.		

мирования может быть дополнен. Состав и номенклатура параметров свидетельствуют о необходимости проведения серьезного предпроектного обследования предметной области внедрения новой технологии. Однако такой оценки недостаточно для объективного выбора ТК: необходим анализ целей внедрения ТК (табл. 2). В зависимости от принятых критериев оценки эффективности и требований к со-

Таблица 2

Характеристика целей ТК

Наименование	Проекты выборки с указанными целями	
	Число	%
Сокращение трудозатрат на разработку ПО	67	77
Достижение требуемых характеристик:		
качества	48	55
надежности	52	60
эффективности	28	33
Эффективность использования ресурсов ЭВМ	12	14
Использование при разработке непрофессиональных программистов	12	14
Создание «дружественной» комфортной обстановки в коллективе разработчиков ПО	11	11
Сокращение затрат на разработку собственной технологии или адаптацию существующей	10	11
Другие цели	12	14

здаваемым программам необходима установка определенных весовых коэффициентов для характеристик ТК (выделяются доминирующие).

Анализ опыта разработки и внедрения крупных технологических комплексов автоматизации разработки программного обеспечения показывает, что повышение эффективности в области технологической организации разработки программ может быть достигнуто на основе системного исследования и анализа функциональных характеристик ПП, характеристик ТК автоматизации и целей их внедрения.

Однако, к сожалению, практика технологической организации разработки программ демонстрирует пока обратную картину — положительные результаты достигаются за счет эволюционного развития различных направлений технологического обеспечения разработки программ включая методы выбора технологии и ее внедрения.

190000, Ленинград, ул. Герцена, 67, ЛИАП; тел. 210-70-44

ЛИТЕРАТУРА

- Штрик А. А. РУЗА — технологическая кросс-система автоматизации разработки комплексов программ для управляющих и микроЭВМ // Микропроцессорные средства и системы. — 1984. — № 2. — С. 46—49.
- Липаев В. В. Промышленная технология разработки программных средств для встраиваемых микроЭВМ (технология ПРОМЕТЕЙ) // Вычислительная техника социалистических стран. — М.: Финансы и статистика. — 1985. — № 18. — С. 54—65.
- Каганов Ф. А., Корепанов Б. А., Минаев М. А. Автоматизация проектирования программ для управляющих ЭВМ и микропроцессоров на базе технологических ЕС ЭВМ — РУЗА // Автоматика и телемеханика. — 1984. — № 7. — С. 159—168.
- Технология проектирования комплексов программ АСУ / В. В. Липаев, Л. А. Серебровский, П. Г. Гаганов и др.; Под ред. Ю. В. Асафьева, В. В. Липаева. — М.: Радио и связь, 1983.
- Бозм Б. У. Инженерное проектирование программного обеспечения. — М.: Радио и связь, 1985.

Статья поступила 4.10.88

УДК 681.326.34
В. И. Джиган

УПРАВЛЯЮЩАЯ МИКРОЭВМ НА ОСНОВЕ МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА БИС СЕРИИ К588

МикроЭВМ семейства «Электроника» в настоящее время широко применяются в качестве устройств управления (УУ). Однако использование серийных микроЭВМ данного семейства в ряде случаев оказывается невозможным из-за высокого энергопотребления и сравнительно небольшого диапазона рабочих температур (+5...+40 °С). Например, мощность, потребляемая микроЭВМ «Электроника 60», равна 30,5 Вт, а микроЭВМ «Электроника МС 1201.01» — 12,6 Вт.

Предлагаемая малопотребляющая микроЭВМ, реализованная на основе микромошного МПК БИС серии К588 [1] разработана для решения задач управления фазированными антенными решетками (ФАР) (при наличии необходимых ВУ ее можно использовать также и для других целей). Устройство управления ФАР — 16-разрядная микроЭВМ (см. рисунок).

Процессор (ПР) выполнен в соответствии со структурной схемой, приведенной в [2]. Центральный процессор (ЦП) реализует систему команд микроЭВМ «Электроника 60». Умножитель (УМ) К588ВР2 подключается к внутренней магистрали ЦП, как ВУ. Для усиления сигналов системной магистрали ЦП используются магистральные приемопередатчики (МПП)

К588ВА1. В состав ПР входит также формирователь сигналов начального пуска и останова (ФСНПО).

В ОЗУ микроЭВМ с помощью контроллера запоминающего устройства (КЗУ) К588ВГ2 и МПП обеспечивается передача и прием информации 16-разрядными словами или байтами. Для этого ОЗУ выполнено в виде двух накопителей: младшего и старшего байтов (НМБ и НСБ). Адресация ячеек ОЗУ осуществляется с помощью регистра адреса (РА) и КЗУ. Объем ОЗУ выбран равным 32 Кбайт исходя из объемов рабочих программ во всех режимах функционирования; таблиц, используемых для повышения скорости выполнения некоторых алгоритмов управления ФАР; минимального объема ОЗУ, необходимого для проверки правильности функционирования узлов микроЭВМ с помощью тест-мониторной системы РАФОС, а также из возможности снижения потребляемой микроЭВМ мощности за счет перезаписи содержимого ПЗУ в ОЗУ с последующим отключением питания от БИС используемой постоянной памяти.

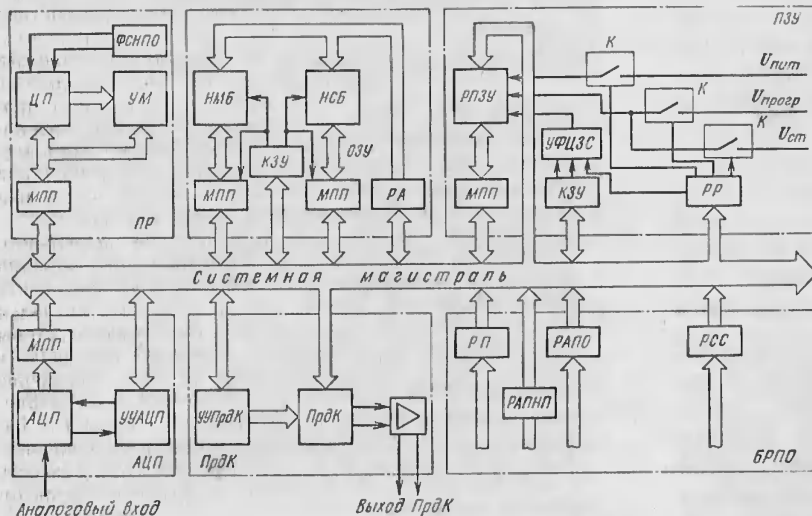
Модуль ПЗУ [3] выполнен на основе БИС электрически репрограммируемого ПЗУ (РПЗУ) КМ558РР3. Применение такой БИС обусловлено

необходимостью многократной смены информации на этапе отладки программ алгоритмов управления ФАР (в этом случае РПЗУ обладают преимуществом перед ПЗУ с однократным программированием). Из двух типов РПЗУ — с ультрафиолетовым или электрическим стиранием информации — предпочтение отдано РПЗУ последнего типа, поскольку для стирания в них информации не требуется такое оборудование, как источник ультрафиолетового излучения. Кроме того, при отладке программного обеспечения (ПО) в модуле ПЗУ целесообразно совместить функции программатора и собственно ПЗУ. Это позволяет менять информацию с помощью управляющей микроЭВМ без извлечения указанных БИС из модуля.

Адресация ячеек памяти БИС РПЗУ осуществляется внутренними регистрами БИС КМ558РР3, а управление работой модуля — адресуемым регистром режима (РР) К588ИР1, КЗУ и устройством формирования циклов записи и стирания (УФЦЗС). При включении питания элементы модуля устанавливаются в состояние, позволяющее считывать данные из РПЗУ. При записи соответствующего кода управления в РР и обращении в цикле вывода к ячейкам из адресного пространства РПЗУ инициализируются программно формируемые циклы записи и стирания информации.

Для репрограммирования БИС КМ558РР3 требуются напряжения программирования $U_{\text{прогр}} = 24$ В и стирания $U_{\text{ст}} = 18$ В, которые через ключи (К), управляемые с помощью РР, подаются от источников питания. Мощность, потребляемая модулем ПЗУ, в основном определяется БИС РПЗУ. В режиме хранения ее величина — 220 мВт, а в режиме обращения — 840 мВт. Мощность, потребляемая ОЗУ такого же объема на основе КМДП БИС, соответственно 20 и 260 мВт. Поэтому при перезаписи информации из ПЗУ в ОЗУ одинакового объема и отключении питания от БИС КМ558РР3 обеспечивается выигрыш в потребляемой мощности в режиме хранения в 11 раз, а в режиме обращения — в 3,2 раза.

Режимы начального пуска и останова ЦП задаются в блоке регистров пуска и останова (БРПО). В состав БРПО входят регистр пуска (РП) с адресом 17300₂, регистр адреса программы начального пуска (РАПНП) с адресом 17300₂, регистр адреса программы останова (РАПО) с адресом 16000₂ и регистр слова состояния (РСС) с адресом 16000₄. РП и



Структурная схема микроЭВМ для управления ФАР

РАППП представляют собой МПП, на входах которых коммутируется команда безусловного перехода и адрес перехода к программе начального пуска соответственно. Останов ЦП реализуется как прерывание по вектору, вводимому с адреса 160002. Пуск и останов ЦП в отладочном режиме осуществляются с помощью ФНСПО при подаче внешних сигналов, а запуск ЦП в рабочем режиме — при включении питания.

Требуемые специализированные ВУ микроЭВМ определяются особенностями алгоритмов управления ФАР. Одна из них — возможность реализации в ФАР без использования оценок сигналов, принятых отдельными антенными элементами. Для выполнения таких алгоритмов необходимо измерять с помощью аналого-цифрового преобразователя (АЦП) только уровень суммарного сигнала на выходе ФАР и передавать сигналы кодов фазовых сдвигов для управления ключами фазовращателей. Передача этих сигналов осуществляется передатчиком кодов (ПрдК) [4] в коде «Манчестер-II» с помощью БИС контроллера мультиплексной линии связи К588ВГ6. АЦП и ПрдК управляются соответствующими устройствами управления (УУАЦП) и (УУПрдК). В микроЭВМ используется одноканальный вариант АЦП [5], реализованный на основе БИС К1108ПВ1.

Управляющая микроЭВМ выполнена на печатных платах размерами 170×300 мм. Габаритные размеры конструкции микроЭВМ не превышают 250×325×160 мм. Потребляемая мощность микроЭВМ не более 2,5 Вт (при подаче напряжения питания $U_{пит}$ на БИС ПЗУ) и определяется в основном БИС ПЗУ и АЦП.

Температурный диапазон функционирования микроЭВМ также определяется указанными БИС и ограничивается значениями $-10...+70^{\circ}\text{C}$.

Методика отладки узлов микроЭВМ управления ФАР с помощью ПО микроЭВМ семейства «Электроника» подробно изложена в [6]. Основные устройства, использованные при отладке управляющей микроЭВМ, следующие: адаптер каналов, устройство межмашинного обмена, модуль интерфейсов стандартных ВУ (алфавитно-цифрового дисплея и накопителя на гибких магнитных дисках).

Адаптер соединяет каналы микроЭВМ управления ФАР и микроЭВМ из семейства «Электроника» в отладочном режиме. Алгоритмы отлаживаются и проверяются с помощью устройства межмашинного обмена в режиме, при котором управляющая микроЭВМ решает задачи управления ФАР, а модели ФАР и сигналов реализуются на языках высокого уровня в ЭВМ семейства «Электроника». В моменты времени, когда требуется передача кодов фазовых сдвигов в ФАР или оценка уровня ее выходного напряжения, соответствующая информация, вырабатываемая управляющей микроЭВМ или моделью ФАР, передается устройством межмашинного обмена в нужном направлении (ПО отлаживается без ФАР и других объектов управления, разработка которых, как правило, ведется параллельно).

Модуль интерфейсов стандартных ВУ позволяет использовать ПО тест-мониторной системы РАФОС для проверки работоспособности управляющей микроЭВМ, управляющую микроЭВМ в качестве персонального компьютера, а также репрограммировать БИС КМ558РРЗ модуля ПЗУ.

103498, Москва, МИЭТ, каф. микроэлектронных радиотехнических устройств и систем; тел. 534-09-08

ЛИТЕРАТУРА

1. Бобков В. А., Шиллер В. А. Комплект КМДП БИС для специализированных 16-разрядных микровычислительных систем с унифицированным интерфейсом // Электронная промышленность. — 1981. — № 4. — С. 32—35.
2. Черняковский Д. Н., Шиллер В. А., Юровский А. А. Процессор с системой команд и интерфейсом микроЭВМ «Электроника 60» // Электронная промышленность. — 1983. — № 9. — С. 14—17.
3. Джиган В. И., Лужецкий С. Г. Модуль электрически перепрограммируемого ПЗУ для микроЭВМ семейства «Электроника 60» // Микропроцессорные средства и системы. — 1988. — № 1. — С. 85—89.
4. Джиган В. И. Линия связи управляющей микроЭВМ с модулями антенной решетки // Микропроцессорные средства и системы. — 1989. — № 3. — С. 50—54.
5. Джиган В. И., Шербаков С. В. Аналого-цифровой преобразователь с интерфейсом типа МПИ // Микропроцессорные средства и системы. — 1990. — № 1.
6. Джиган В. И. Устройства отладки аппаратных средств и программного обеспечения микроЭВМ управления ФАР // Микропроцессорные вычислительные устройства управляющих систем: Межвузовский сб. — М.: Изд. МИЭТ, 1988. — С. 80—89.

Статья поступила 15.12.87

УДК 681.325

А. А. Григорьев, А. И. Федосова

ПРИНЦИПЫ СОПРЯЖЕНИЯ МИКРОЭВМ С ПАРАЛЛЕЛЬНЫМИ КАНАЛАМИ ВВОДА-ВЫВОДА

Интенсивное внедрение средств вычислительной техники в сферу автоматизации научных исследований и производственных процессов делает актуальной задачу сопряжения периферийных устройств (ПУ) с микроЭВМ. Для организации связи с ПУ широко применяются параллельные каналы передачи данных с протоколами асинхронного обмена. Сопряжение таких каналов с системным каналом

микроЭВМ требует разработки специальных интерфейсных модулей (адаптеров) [1].

Рассмотрим структуру параллельного асинхронного обмена. Предположим, что между источником (И) и получателем (П) информации (рис. 1) имеется параллельный канал связи DC, состоящий из n электрических линий, по каждой из которых передается один двоичный разряд данных. Наличия физического канала связи не достаточно для передачи данных, необходимо, чтобы моменты приема были синхронизованы во времени с интервалами передачи, а темп передачи согласован со скоростными характеристиками получателя.

Взаимное согласование действий устройств по приему и передаче данных осуществляется на так называемом

уровне протоколов обмена. В широком смысле под протоколом понимают совокупность правил, однозначно определяющую порядок взаимодействия устройств. Для обеспечения параллельного обмена данными наиболее распространенные варианты некоторого стандартного протокола асинхронного обмена, предполагающего наличие между устройствами двух дополнительных линий для передачи синхросигналов источника (СИ-И) и получателя (СИ-П).

В основе протокола параллельного асинхронного обмена лежит принцип диалога, в котором роль отдельных реплик играют фронты синхросигналов СИ-И и СИ-П. Протокол требует, чтобы источник и получатель формировали их строго поочередно (рис. 2). Соблюдение этого требования обеспечивает разбиение процесса обмена на фазы F1...F4, чередующиеся в правильной циклической последовательности. Длительность фаз F1, F3, заканчивающихся фронтом СИ-П, определяется получателем, F2, F4 — источником. Обмениваясь фронтами синхросигналов, источник и полу-

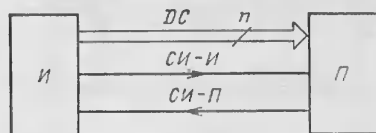


Рис. 1. Параллельный канал асинхронного обмена

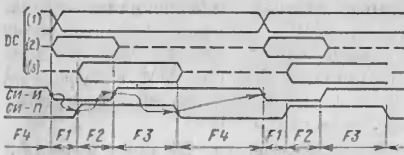


Рис. 2. Временные диаграммы асинхронного обмена

чателъ совместно формируют единую дискретную временную шкалу.

Можно выделить три варианта протокола асинхронного параллельного обмена, различающихся по способу использования канала:

с монополизацией канала источником; данные поступают в канал по спаду сигнала СИ-И и остаются активными до следующего момента выдачи;

с занятием канала по инициативе источника; данные передаются по низкому уровню сигнала СИ-И, который в этом случае играет роль stroba данных;

с занятием канала по требованию получателя; данные передаются по высокому уровню сигнала СИ-П, который играет роль требования получателя на передачу.

Прием данных из канала может быть организован по-разному. Универсальное решение — организация приема по нарастающему фронту stroba СИ-И, обеспечивающая уверенный прием при любом способе использования канала.

Во всех вариантах протокола F3 и F4 могут рассматриваться как интервалы ожидания по отсутствию готовности получателя и источника соответственно. Эти фазы обеспечивают согласование темпов передачи, приема и обработки данных.

Второй и третий варианты протоколов можно скомбинировать для организации двунаправленного асинхронного обмена между двумя ПУ (рис. 3). Пусть каждое ПУ может быть как источником, так и получателем данных. Предположим, данные от И1 к П2 передаются под управлением синхросигналов СИ-И1, СИ-П2 по третьему варианту протокола асинхронного обмена, а передача от И2 к П1 — по второму варианту. При этом оказывается, что порядок использования канала полностью определяется устройством ПУ2, на которое возлагается ответственность за безконфликтную передачу данных по единому каналу в обоих направлениях. Условие отсутствия конфликтов сводит-

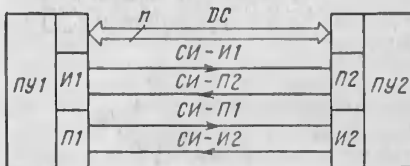


Рис. 3. Канал двунаправленного асинхронного обмена

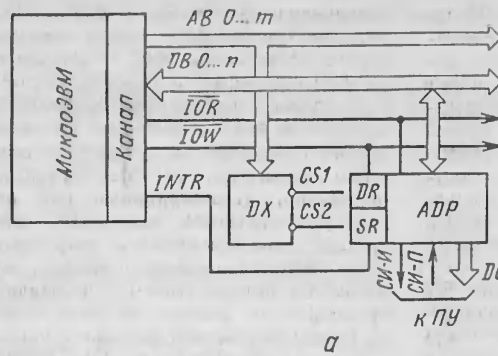


Рис. 4. СК микроЭВМ:

а — сопряжение с адаптером асинхронного канала; б — диаграммы циклов канала

ся к тому, что ПУ2 не должно выдавать низкий уровень stroba передаваемых данных СИ-И2 и высокий уровень требования на передачу принимаемых данных СИ-П2 одновременно.

Системный канал микроЭВМ. Предположим, что системный канал (СК) микроЭВМ имеет простейшую структуру (шины адреса (AB) и данных (DB) разделены, сигналы управления записью (IOW) и считыванием (IOR) независимы, рис. 4, а) Ввод-вывод данных осуществляется при выполнении определенных команд.

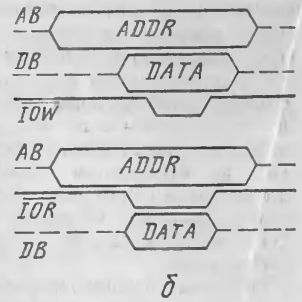
Канальный цикл ввода-вывода начинается с поступления на шину AB адреса абонента, с которым будет осуществляться обмен. На шину DB выдаются передаваемые абоненту данные, сопровождаемые активным низким уровнем сигнала записи IOW. По сигналу чтения IOR абонент должен выдать имеющиеся у него данные на шину DB для передачи в микроЭВМ.

СК сопрягается с асинхронным параллельным каналом передачи данных с помощью адаптера канала ADP (рис. 4), воспринимаемого ЭВМ как совокупность двух абонентов, называемых регистрами состояния SR и данных DR. Они выбираются сигналами CS1, CS2, формируемыми дешифратором DX, при наличии на адресной шине СК присвоенных этим регистрам адресов. Адаптер канала может формировать сигнал требования обслуживания для подачи на один из входов требования прерываний ЭВМ.

Адаптеры параллельных каналов. Адаптер асинхронного канала — это устройство, которое выполняет следующие функции:

электрически сопрягает канал передачи данных с шиной данных СК; ведет асинхронный обмен с ПУ, принимая входной и формируя выходной синхросигналы в соответствии с протоколом; реализует обращение к ЭВМ с требованием обслуживания.

Буферный регистр данных DR принимает данные из канала по нарастающему фронту сигнала СИ-И (STB) и передает их на шину данных СК



по низкому уровню сигнала чтения RD. В отсутствие сигнала RD выхода находятся в высокоимпедансном состоянии и не нагружают шину DB (рис. 5).

Участие адаптера в обмене на уровне протокола обеспечивается триггером IBF. Сигнал IBF (СИ-П) устанавливается по низкому уровню STB и сбрасывается по фронту сигнала RD (после передачи в ЭВМ данных при-

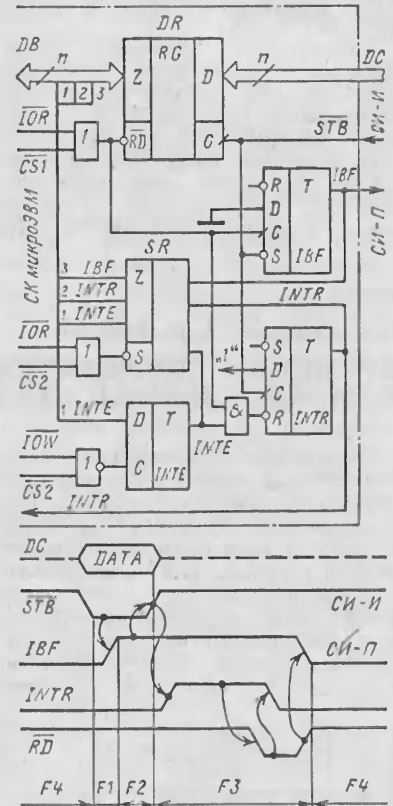


Рис. 5. Адаптер асинхронного канала ввода

нятых из канала). Низкий уровень IBF свидетельствует о готовности адаптера (получателя) к приему.

Адаптер участвует в асинхронном обмене как автономное устройство. В фазе отсутствия готовности (F3) адаптер должен обратиться к ЭВМ с требованием обслуживания. Возможность такого обращения на аппаратном уровне обеспечивается триггером сигнала требования обслуживания INTR и регистром SR. Трехразрядный регистр состояния позволяет микроЭВМ программно устанавливать и очищать внутренний триггер разрешения прерываний адаптера INTE, а также программно контролировать сигналы IBF, INTR, INTE.

Взаимодействие адаптера с микроЭВМ может быть организовано двумя способами. При программно управляемом обмене ЭВМ распознает факт наличия требования обслуживания от адаптера, программно контролируя это IBF в SR. Единичное значение этого бита — признак (флаг) требования. Другой способ организации взаимодействия — обмен по прерываниям. Если прерывания от адаптера разрешены (INTE=1), то по нарастающему фронту STB (в момент приема данных из канала) в адаптере устанавливается триггер требования

обслуживания INTR. Сигнал INTR поступает в систему прерываний микроЭВМ. При выполнении соответствующей программы обработки прерывания данные считываются из регистра DR. В момент считывания (по сигналу RD) триггер INTR сбрасывается.

Рассмотренный адаптер реализует передачу данных из параллельного канала в микроЭВМ за два этапа: данные принимаются из канала и фиксируются в промежуточном буферном регистре DR, а затем считываются из него.

Если порядок использования канала определяется источником (рис. 2), могут применяться адаптеры, реализующие передачу данных из канала в микроЭВМ в один этап. В таких адаптерах для обращения к микроЭВМ с требованием обслуживания используется фаза F1, предшествующая выдаче сигнала СИ-П подтверждения приема данных из канала. Сигнал подтверждения формируется только после ввода в микроЭВМ данных, находящихся в канале.

Участие адаптера асинхронного канала вывода (рис. 6) в обмене на

уровне протокола обеспечивается триггером OBF, который устанавливает активный сигнал СИ-И в момент записи данных от микроЭВМ в буферный регистр DR и снимает его по сигналу подтверждения СИ-П.

Буферный регистр DR принимает данные по сигналу WR и выдает их в канал по низкому уровню сигнала DE. Способ формирования сигнала DE зависит от используемого варианта протокола. В случае протокола с монополизацией канала (вариант 1) вход DE подключается к земле. Протокол с занятием канала по инициативе источника (вариант 2) реализуется при подаче на этот вход стопа данных с выхода триггера OBF. Наконец, при передаче данных по требованию получателя (вариант 3) на вход DE подается внутренний сигнал подтверждения ACK, инверсный по отношению к СИ-П.

В качестве примера рассмотрим БИС КР5801К55А (рис. 7) — универсальный программируемый адаптер параллельных каналов [3, с. 147].

БИС предоставляет три независимо

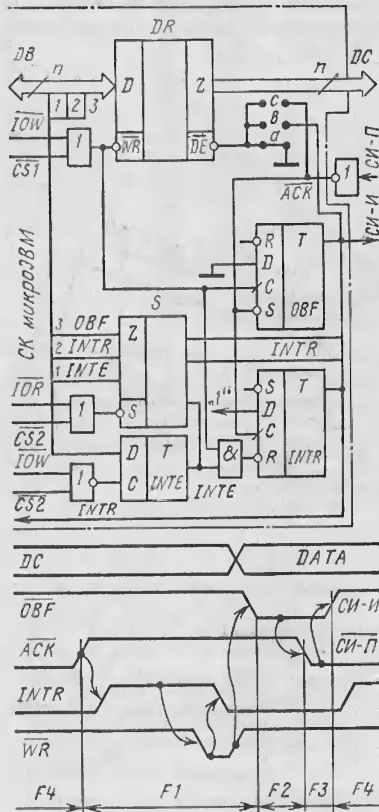


Рис. 6. Адаптер асинхронного канала вывода

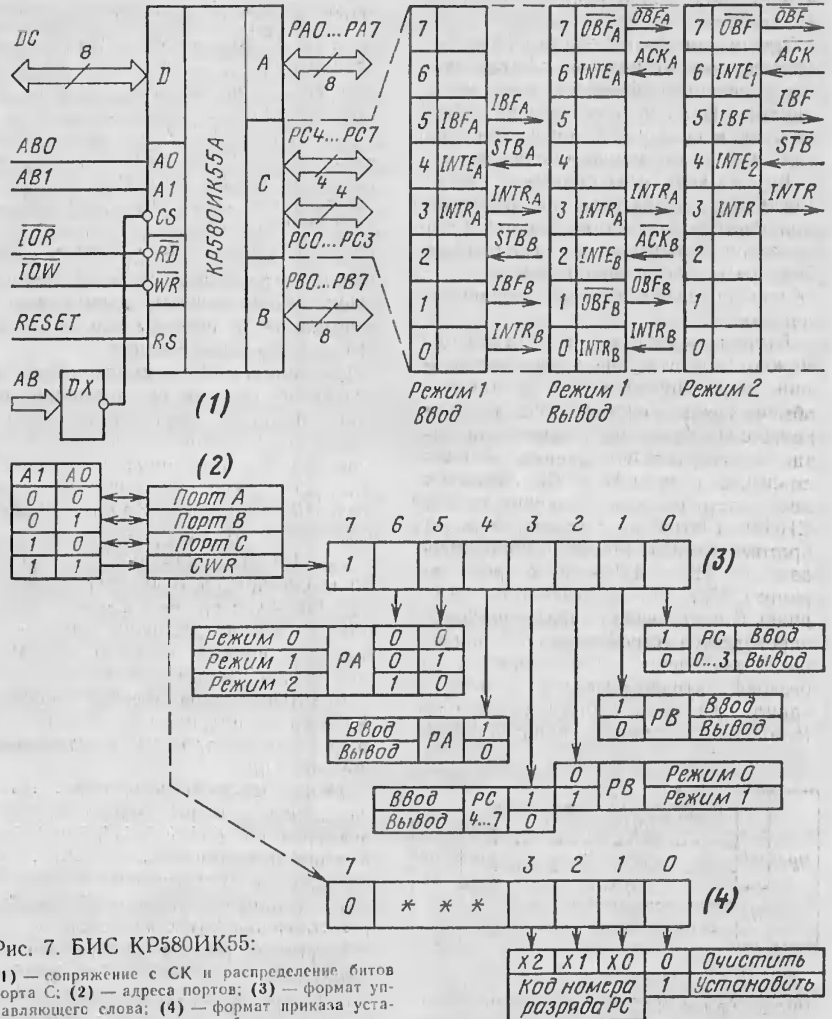


Рис. 7. БИС КР5801К55А:

(1) — сопряжение с СК и распределение битов порта С; (2) — адреса портов; (3) — формат управляющего слова; (4) — формат приказа установки-очистки битов

адресуемых порта ввода-вывода — А, В и С, режимы работы которых определяются программно записью слова определения конфигурации в формате (3) по адресу регистра управляющего слова (2). Наибольшими возможностями обладает порт А, который может быть запрограммирован на ввод или вывод в одном из трех режимов — 0, 1 и 2. В режиме 0 порт А функционирует подобно буферным регистрам адаптеров (рис. 5, 6), сопрягая 8-разрядный параллельный канал РА с СК на физическом (беспротокольном) уровне.

В режиме 1 порт А выполняет функции адаптера асинхронного канала ввода по схеме, приведенной на рис. 5, или адаптера канала вывода (рис. 6). В обоих случаях в качестве регистров состояния адаптеров используются группы разрядов порта С (1). По внешним выводам этих разрядов передаются сигналы управления.

В режиме 2 порт А с пятью разрядами порта С (1) выполняет функции адаптера двунаправленного асинхронного канала (рис. 3). Данные из адаптера выводятся в канал РА под управлением синхросигналов ОВФ, АСК по протоколу (3), а принимаются под управлением синхросигналов СТВ, ИВФ по протоколу (2). Таким образом, функции управления использованием канала рассматриваемый адаптер передает ПУ. В фазе 3 субканала ввода (рис. 5) и в фазе 1 субканала вывода (рис. 6) формируется единый сигнал требования обслуживания INTR. Биты INTE₁, INTE₂ регистра состояния позволяют независимо разрешать и запрещать прерывания от субканалов вывода и ввода соответственно.

Порт В допускает только два режима использования — 0 и 1, аналогичные соответствующим режимам порта А. Разряды порта С, незадействованные совместно с портами А и В в качестве регистров состояния адаптеров асинхронных каналов, могут быть запрограммированы независимо на ввод или вывод в режиме 0 (3). Имеется возможность независимого управления состоянием отдельных разрядов порта С путем записи приказа установочности битов в формате (4) по адресу CWR.

141700, Моск. обл., Долгопрудный, МФТИ, каф. радиотехники; тел 408-44-55

ЛИТЕРАТУРА

1. Лукьянов Д. А. КР580 — автоматизация без проблем! // Микропроцессорные средства и системы. — 1985. — № 1. — С. 85—90.
2. Панфилов Д. И., Шаронин С. Г., Яковлев С. Е. Организация обмена информацией с микроЭВМ в параллельном коде // Микропроцессорные средства и системы. — 1986. — № 5. — С. 85—89.
3. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и мини-ЭВМ. — Я.: Энергоатомиздат, 1984. Статья поступила 13.10.87

УДК 621.397.96:681.3

А. А. Урсатьев, С. Л. Сапожникова, С. А. Тарасенко

СИСТЕМА ДЛЯ РЕГИСТРАЦИИ И ЭКСПРЕСС — АНАЛИЗА СИГНАЛОВ

Для анализа отражений от элементов системы океан-атмосфера и морских радиолокационных (РЛ) целей разработана система на базе микроЭВМ «Электроника 60» или ДВК, регистрирующая эхо-сигналы, накапливающая экспериментальные данные на долговременных носителях информации и обрабатывающая их. Такие экспериментальные исследования необходимы до натурных испытаний [1, 2]. По статистическим характеристикам отражений от надводных объектов и поверхности моря оцениваются параметры радиолокационных эхо-сигналов и разрабатываются оптимальные алгоритмы обнаружения, а по сформированным банкам РЛ-изображений предварительно проверяется адекватность проектных решений реальным процессам.

Система работает совместно с импульсными некогерентными радиолокационными станциями (РЛС) (например, судовыми навигационными РЛС [3], используемыми в качестве датчиков информации в системах предупреждения столкновений). По своей структуре радиолокационная информация (рис. 1) подобна телевизионной (телевизионному кадру соответствуют N разверток видеосигнала по пространственной координате за обзор). Видеосигнал преобразуется в системе в цифровую форму; данные наблюдений накапливаются; на штатных индикаторах кругового обзора (ИКО) РЛС отображаются радиолокационные изображения, предварительно записанные в оперативное ЗУ системы или на внешние носители информации.

Прикладное программное обеспечение (ППО) системы ориентировано на

оперативную оценку по выборкам сравнительно небольшого объема основных статистических характеристик: корреляционно-спектральных и законов распределения вероятностей амплитуд эхо-сигналов. Такие оценки необходимы для выбора режимов съема данных в экспериментальных исследованиях новых или малоизученных явлений (процессов), идентификации регистрируемых данных выбранному объекту исследования, выработке гипотез относительно характера радиолокационных отражений. ППО реализует устойчивые алгоритмы статистической обработки сигналов [4] и при поддержке ОС RT11Sj требует для своего размещения 48 Кбайт оперативной памяти. При этом максимальный объем обрабатываемого массива данных составляет 14 Кбайт.

Для увеличения объема регистрируемой информации в системе предусмотрена возможность использования дополнительных ЗУ типа «Электронный диск» (по 512 Кбайт).

Диалоговый режим пользователя с системой позволяет выбрать следующие виды обработки:

просмотр результатов измерений по произвольному направлению двумерного массива данных;

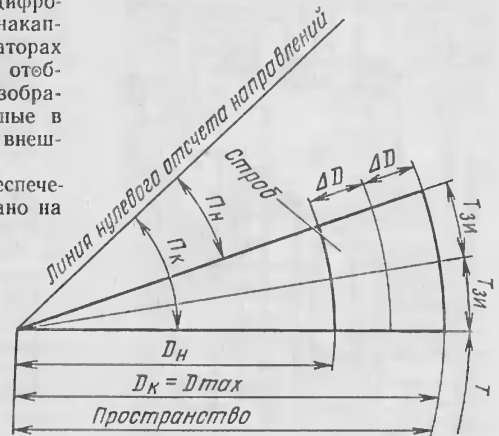


Рис. 1. Параметры строба регистрации РЛ — информации:

$T_{зи}$ — период излучения зондирующего импульса, P_n и P_k — пеленг начала и конца строба, D_n и D_k — дальность начала и конца строба

Технические характеристики системы:

Режимы регистрации РЛ-информации	Стробо (круговой обзор РЛС), остановленная антенна
Параметры строба	Программируются пользователем
Частота дискретизации, МГц	1,5...20,0
Полоса частот кодируемого сигнала, не более, МГц	7
Число уровней квантования	64 или 256
Число элементов разрешения по пространственной координате, не более	1024
Уровни сигналов (при работе на согласованный видеотракт с волновым сопротивлением 75 Ом), В	2,5 или 4,0
Синхронизация	От РЛС или РЛС от системы
Регистрация радиолокационных данных (частота измерений по временной координате), кГц	С частотой запуска РЛС или уменьшенной до значения 2^{16}

подавление аномальных наблюдений путем одномерной фильтрации исходных массивов данных;

получение оценок основных вероятностных характеристик сигналов (эмпирических плотностей распределения амплитуд эхо-сигналов, параметров распределения в семействе распределений Джонсона; корреляционных функций эхо-сигналов и спектров флюктуаций отражений).

Получаемые характеристики сигналов и исходные массивы отображаются на графическом дисплее «Электроника МС7401». Обрабатываемый массив экспериментальных данных в общем случае двумерный (угол α и начальная точка K_1). При его обработке выбираются данные, формируемые в одномерные массивы, по которым оцениваются требуемые статистические характеристики (рис. 2). Линии, соединяющие точки отсчетов показывают последовательность выбора данных при заданном угле. Реализации обрабатываются с объемами, не превышающими L при $|\alpha| < 45^\circ$ и K при $|\alpha| \geq 45^\circ$ (L и K — размерности массива данных по осям X и Y соответственно).

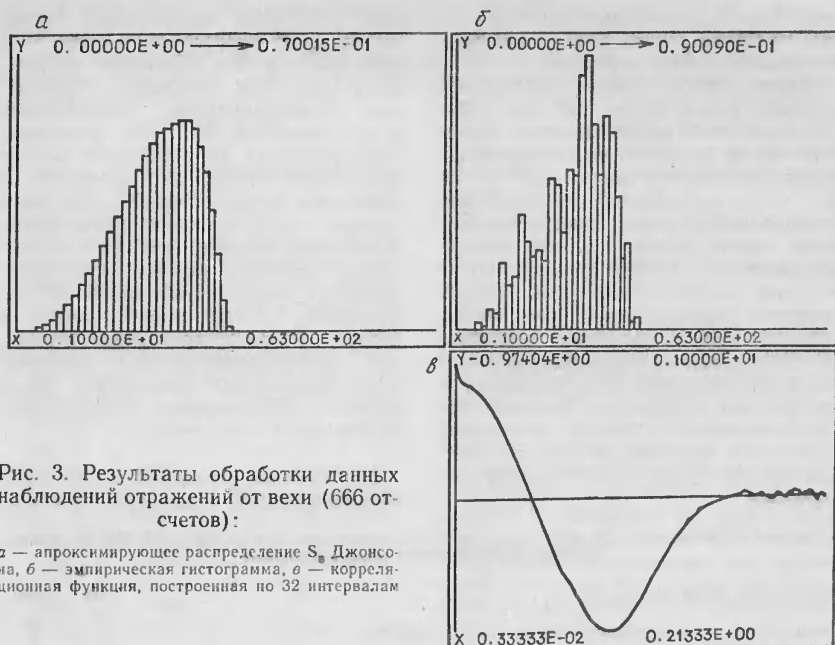


Рис. 3. Результаты обработки данных наблюдений отражений от вехи (666 отсчетов):

a — аппроксимирующее распределение S_B Джонсона, b — эмпирическая гистограмма, c — корреляционная функция, построенная по 32 интервалам

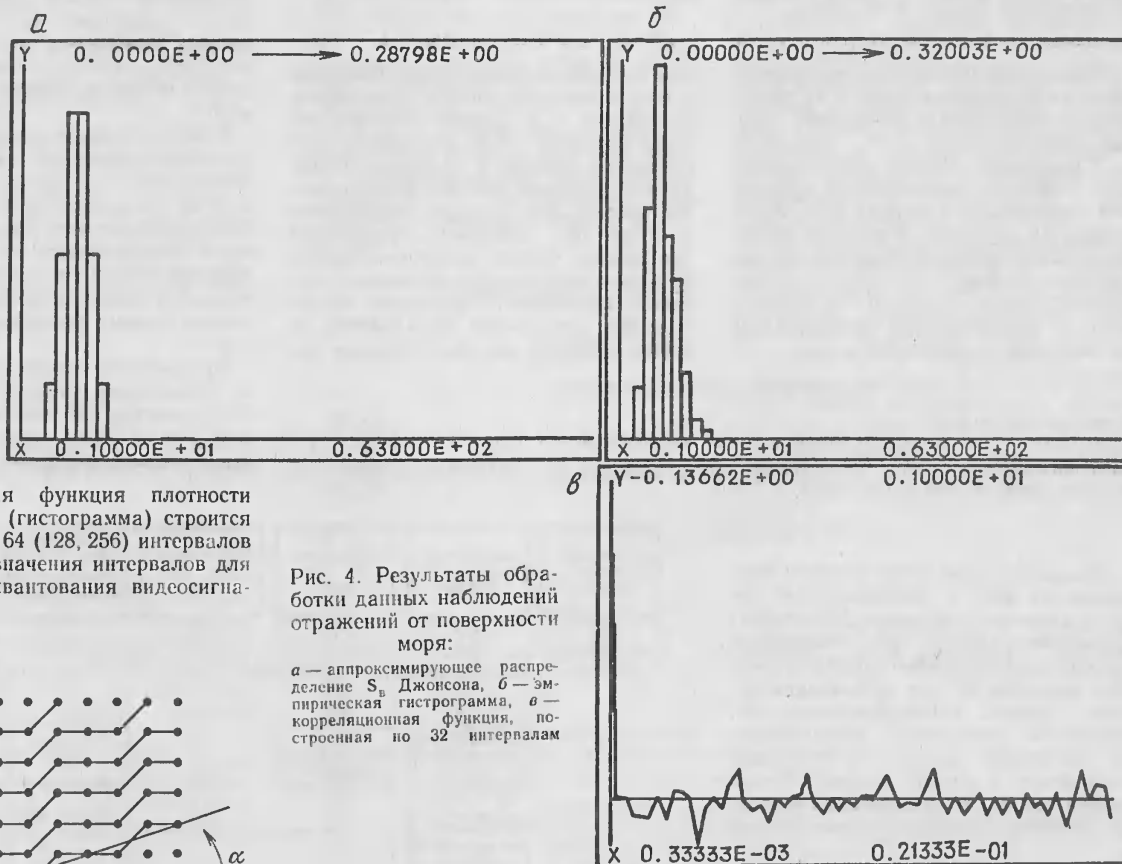


Рис. 4. Результаты обработки данных наблюдений отражений от поверхности моря:

a — аппроксимирующее распределение S_B Джонсона, b — эмпирическая гистограмма, c — корреляционная функция, построенная по 32 интервалам

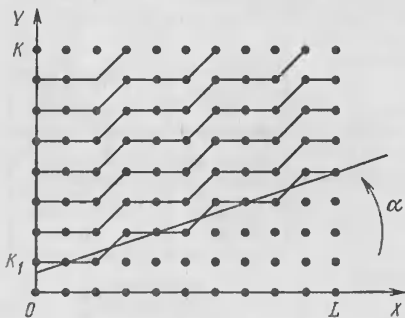


Рис. 2. Схема последовательности выбора данных

В качестве распределения, аппроксимирующего экспериментальные данные, используется семейство S_B Джонсона. Оценка параметров этого распределения производится по двум выборочным квантилям эмпирического распределе-

ния. Пары квантилей зафиксированы на следующих уровнях» (0,05; 0,95), (0,1; 0,9), (0,2; 0,8), (0,3; 0,7), (0,4; 0,6).

Дискретная корреляционная функция вычисляется не более, чем по 256 точ-

кам. Максимальный интервал корреляции ограничен длиной выбираемой реализации случайного процесса.

Энергетический спектр случайного процесса определяется методом БПФ его дискретной корреляционной функции. Число получаемых спектральных линий составляет $2^l (2^l \leq M < 2^{l+1})$, где M — число отсчетов используемой корреляционной функции. Значения интервалов корреляции и гармоник спектра определяются на основе информации о режимах съема и обработки данных (паспорта кадра) и индицируются пользователю в виде подписей к графикам соответствующих характеристики и сигналов. Устойчивость статистических характеристик исследуемых сигналов достигается предварительной процедурой одномерной фильтрации массива данных по направлению максимальной размерности.

Временные характеристики (в сек.) прикладного ПО, полученные при работе с ЦП М2 при разных объемах обрабатываемых массивов данных:

Гистограмма на 64 интервала	700	1400
Оценка параметров и построение функции распределения Джонсона	15	20
Корреляционная функция по 32 точкам	15	15
Спектр на 32 гармоника	105	210
Фильтрация массивов данных	5	5
	4	8

Рассмотрим результаты экспериментальных исследований (рис. 3, 4). Функции датчика сигнала выполняла судовая навигационная РЛС «Наяда-5» 3-см диапазона. Данные регистрировались в режиме остановленной антенны при синхронизации запуска РЛС от системы. Дискретность изменения интервала вычисляемых корреляций при отражении от вехи — $3,3 \cdot 10^{-3}$ с, от поверхности моря — $0,33 \cdot 10^{-3}$ с. Аппроксимация распределений производилась по квантилям уровней 0,2 и 0,8.

типа СМ5400; специальные блоки, выполненные в конструктиве микроЭВМ «Электроника 60». Последние включают в себя блок цифрового кодирования и декодирования видеосигнала и интерфейсный блок для согласования временных характеристик канала микроЭВМ, обладающего сравнительно невысоким быстродействием, с высоким темпом преобразования информации. Кроме того, интерфейсный блок обеспечивает желаемые режимы функционирования системы (регистрацию или отображение РЛ-информации, работу в стробе или при остановленной антенне, вид синхронизации и др.), пространственно-временные параметры съема данных и синхронизацию работы датчика сигнала с системой.

Центральный процессор обменивается информацией с интерфейсным бло-

ком с помощью программных операций с использованием средств прерывания программы и в режиме прямого доступа к памяти ЭВМ. По каналу программного обмена в регистрах блока задаются уставки, определяющие режимы работы. Эта функция реализуется в ответ на требование прерывания программы, которое инициализируется РЛС при прохождении ее антенной нулевого положения. Программа обслуживания прерывания выполняется во время действия импульса отсетки ну-

левого положения антенны (ИНПА). Таким образом, для каждого нового оборота антенны параметры стро-ба могут быть перепрограммированы изменением уставок соответствующих регистров. Уставки регистров рассчитываются по данным интерактивного диалога пользователя с системой на профессионально естественном языке.

Канал прямого доступа к памяти ЭВМ выполнен на основе стандартного интерфейса И3 15 КС-16-002. Интерфейсный блок (рис. 6) хранит данные измерений радиолокационного сигнала одной зондирующей посылки в течение времени, необходимого для их перезаписи в ОЗУ микроЭВМ. Процесс перезаписи данных должен быть завершён к следующему зондированию с периодом $T_{3и}$. В противном случае формируется специальный режим прерывания устройства, свидетельствующий о нарушении временных соотношений и идентифицирующий неисправность системы.

Время регистрации данных (T_p) определяется размером стро-ба по дальности и изменяется от одного периода тактовой частоты АЦП до 460 мкс (зависит от шкалы дальности РЛС). Число измерений за период ($T_{3и}$) должно выбираться из условия $T_p + T_{плд} \leq T_{3и}$, где $T_{плд}$ — время передачи данных по каналу прямого доступа к памяти.

В режиме воспроизведения РЛ-изображения данные из ОЗУ микроЭВМ с периодом $T_{3и}$ по каналу прямого доступа загружаются в интерфейсный блок, преобразуются в аналоговую форму и отображаются на ИКО РЛС. Для реализации этого режима программируются значения параметров, аналогичные режиму регистрации РЛ-сигналов.

Программное обеспечение поддержки технических средств системы (ПО ТС): средства интерактивного диалога, диагностика ТС системы, настройка на заданные режимы функционирования и

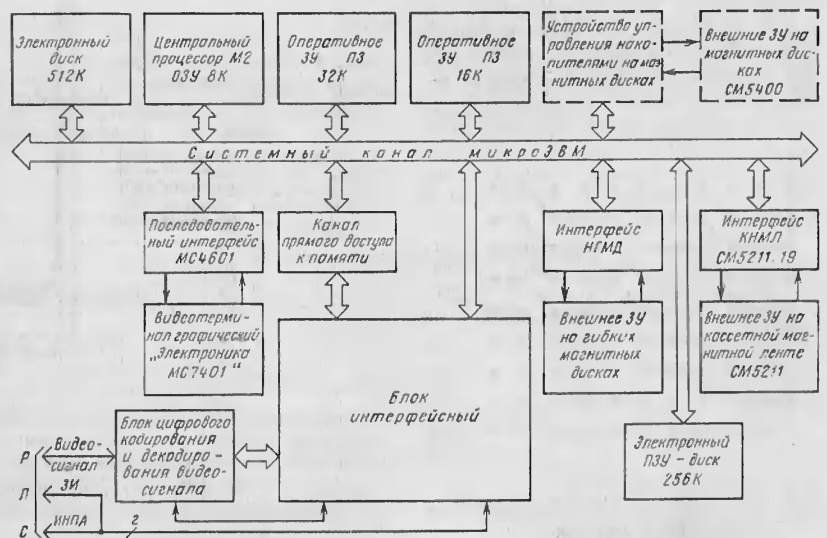
Характеристики режима съема данных:

Размер массива данных, отсчеты	6Х666
Число измерений видеосигнала по пространственной координате, шт.	6
Интервал между регистрируемыми данными, м	15
Число реализаций по каждому наблюдаемому элементу пространства	666
Интервал между отсчетами реализаций, с	$3,3 \cdot 10^{-4}$

Функциональная схема системы приведена на рис. 5. Нестандартные узлы — «электронный диск» (512 Кбайт); интерфейс КНМЛ, обеспечивающий подключение УВП КМЛ СМ5211 в качестве внешнего ЗУ для организации архивов данных РЛ-изображений [5]; устройство управления* накопителями на магнитных дисках, позволяющее подключать к каналу микроЭВМ при работе системы в стационарных условиях внешние ЗУ на магнитных дисках

* Устройство управления разработано СКБ НП «Оптика» Института оптики атмосферы Томского филиала СО АН СССР.

Рис. 5. Функциональная схема системы регистрации и экспресс-анализа РЛ-сигналов



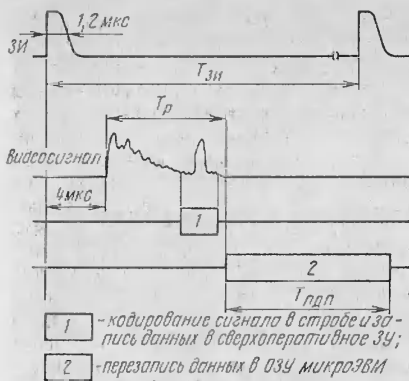


Рис. 6. Временная диаграмма регистрации РЛ-информации

их выполнение занимает 500 слов оперативной памяти и функционирует с любой из версий ОС РАФОС. При использовании ОС RT-11 пользователю для накопления данных предоставляется 15 Кслов оперативной памяти. Программные средства реального времени функционирования системы (синхронно с датчиком сигнала) выполнены на ассемблере; диалога, расчета всевозможных уставок для программирования ТС, поддержки периферийных устройств и ЗУ «Электронный диск» — на Фортране.

Надежность системы и эффективность ее эксплуатации в натуральных условиях обеспечивает энергозависимый электронный диск (ПЗУ-диск [6]).

УДК 681.3

Д. А. Киткин, В. Г. Копп

ПОСЛЕДОВАТЕЛЬНЫЙ КАНАЛ СВЯЗИ С ИСПОЛЬЗОВАНИЕМ КОДА «МАНЧЕСТЕР-II»

В распределенных системах управления технологическим оборудованием (СУ ТО) на базе микропроцессорных средств обмен информацией между локальными уровнями управления, как правило, происходит по последовательным каналам связи, в качестве которых обычно используются каналы типа ИРПС «Токовая петля» (модуль МПО-4 [1]).

Существенные недостатки таких средств связи — ограниченная разрядность передаваемых посылок (7—8 бит) и низкая скорость передачи данных

252207, Киев, пр. академика Глушкова, 20, Институт кибернетики имени В. М. Глушкова АН УССР, тел.: 265-30-35

ЛИТЕРАТУРА

1. Морская радиолокация / Под ред. В. И. Винокурова. — Л.: Судостроение, 1986.
2. Жерлаков А. В., Зимин Н. С., Кононов О. В. Радиолокационные системы предупреждения столкновения судов. — Л.: Судостроение, 1984.
3. Судовые радиолокационные станции. Атлас / Под ред. А. М. Байрашевского. — М.: Транспорт, 1986.
4. Корнильев Э. А., Прокопенко И. Г., Тарасенко С. А. Устойчивый алгоритм оценивания параметров сигнала // Тез. докл. Всесоюз. науч.-техн. конференции по статистическим методам в теории передачи и преобразования информационных сигналов, Львов, окт. 1988. — К., 1988.
5. Накопитель на кассетной магнитной ленте СМ5211 — внешнее ЗУ микроЭВМ «Электроника 60». В. Г. Бескровный и др. // Микропроцессорные средства и системы. — 1990. — № 1.
6. Бронштейн Р. А., Евтехов А. С. Электронный диск для микроЭВМ «Электроника 60», ДВК с энергозависимым хранением информации // Микропроцессорные средства и системы. — 1988. — № 2. — С. 82.

Статья поступила 1.08.89

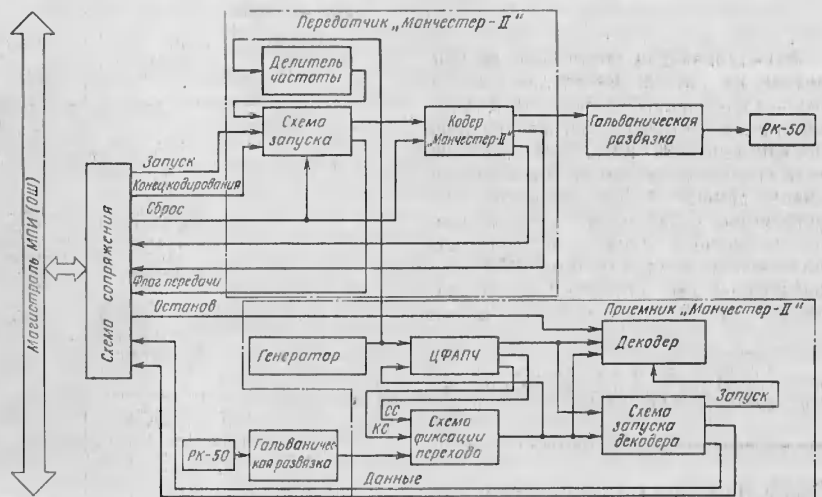
(до 60 КБод/с [2]). Один из способов увеличения разрядности сообщений и скорости их передачи без уменьшения длины линии связи основан на временном уплотнении цифровых сигналов с применением самосинхронизирующегося кода «Манчестер-II», содержащего в себе информационную последовательность и синхросигналы.

Манчестерский код отображает каждый двоичный разряд переходом уровня в середине отведенного каждому элементу временного интервала. Направление перехода определяет значение двоичной переменной. При замене низкого уровня высоким передается Лог. 0, при переходе от высокого уровня к низкому — Лог. 1. Информационный переход можно использовать для обнаружения ошибок передачи, его отсутствие рассматривается как недопустимый разряд, который необходимо фиксировать и учитывать при дальнейшей обработке ошибок.

Трудности, возникающие при применении интерфейсных микросхем серии К588, использующих код «Манчестер-II», полностью преодолены в предлагаемом последовательном канале связи (см. рисунок): не нужны специальные дополнительные устройства в качестве повторителей мультиплексного канала связи и разработка устройства, реализующего функции контроллера канала связи с соответствующим программным обеспечением для разрешения конфликтных ситуаций [5] и обмена между вычислительными устройствами с системной магистралью типа МПИ.

Передатчик и приемник работают по алгоритму, описанному в [3]. После запуска передатчик приступает к формированию последовательности, состоящей из восьми нулей в коде «Манчестер-II» и синхрипульса.

Выходные сигналы кодера синхронизируют работу внешних схем; в начале информационной последовательности формируются два нуля в качестве



Последовательный канал связи

Характеристики последовательного канала связи

Способ передачи информации	Синхронный
Режим обмена информацией	Дуплексный
Максимальная скорость передачи сообщений, Мбит / с	1,25
Максимальная разрядность информационной последовательности, бит	4096
Реализуемые протоколы обмена	ADCCP, SDLC, HDLC
Конструкция канала связи	Кабель РК-50
Максимальная длина линии связи, м	500
Гальваническая развязка	На основе трансформаторов ТИЛ-18

допустимых разрядов для запуска схемы декодирования в приемнике.

Передатчик останавливается по сигналу «Конец кодирования» от программируемого таймера. Время работы таймера выбирается пользователем программно по протоколу обмена. Окончание кодирования сообщений фиксируется сигналом «Флаг передачи», сбрасываемым после поступления сообщения. Начальная установка схемы передатчика осуществляется сигналом «Сброс».

Цифровая схема фазовой автоподстройки (ЦФАПИ) формирует два сигнала: СС, определяющий момент ожидания перехода, и КС — момент принятия решения (декодирования). При появлении перехода раньше момента ожидания (скорость передачи превышает ожидаемую) сигнал КС возникает на один такт раньше. При запаздывании информационного перехода КС вырабатывается на один такт позже. ЦФАПИ проверяет каждый переход и повторно синхронизирует декодер.

Ожидаемые моменты появления переходов известны, поэтому выделение частоты синхронизации декодера происходит в результате сопоставления истинных и требуемых переходов. Декодер запускается синхронным импульсом и

двумя Лог 0. Данные, синхронизованные частотой декодирования, поступают на выход схемы приемника. Время до останова приемника программируется по протоколу обмена. Средства сопряжения с магистралями МПИ и ОШ включают в себя следующие схемы:

распознавания и формирования флага при приеме и передаче информационной последовательности;

программируемого останова приемника и передатчика;

контроля ошибок при передаче (сравниваются передаваемая контрольная сумма и сформированная при приеме информационной последовательности); формирования контрольной суммы при передаче информационной последовательности;

контроллера прямого доступа в память систем с магистралями МПИ и ОШ;

четырёх восьмиразрядных регистров для приема и передачи полей адреса и управления в сообщении;

буферного ОЗУ приемника (4Кбит) для промежуточного хранения информации в последовательном коде при работе в дуплексном режиме. Технические средства конструктивно вы-

полнены на двух платах размерами 280×240 мм, которые устанавливаются в блоки с помощью соединителей РППМ-16-72.

Телефон 535-53-63, Москва

ЛИТЕРАТУРА

1. Волков С. В., Дудкин М. В. и др. Программное обеспечение коммуникационных процессоров в распределенных вычислительных комплексах // Микропроцессорные средства и системы. — 1986. — № 6. — С. 30—33.
2. Преснухин Л. Н., Белильников В. И. и др. Адаптер локальной вычислительной сети на базе БИС К1801ВП1-065 // Микропроцессорные средства и системы. — 1986. — № 6. — С. 57—58.
3. Сандерс Л. Однокристалльный кодер-декодер для преобразования кода без возвращения к нулю в манчестерский код // Электроника. — 1982. — № 15.
4. Хвощ С. Т., Смолов В. Б., Сухопаров А. И., Горовой В. В., Черняковский Д. Н. Комплект БИС для организации мультимплексных каналов обмена информацией // Микропроцессорные средства и системы. — 1984. — № 3. — С. 18—22.
5. Вейцман К. Распределенные системы мини- и микроЭВМ. — М.: Финансы и статистика, 1983.

Статья поступила 14.05.87

СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ СТЗ-2М

Система технического зрения СТЗ-2М* (рис. 1) предназначена для считывания, записи, хранения и предварительной обработки оптической информации, формируемой датчиками изображения на приборах с зарядовой связью (ПЗС). Модульное построение системы позволяет варьировать ее функциональные возможности в зависимости от конкретной решаемой задачи.

При разработке систем технического зрения (СТЗ) необходимо разграничение функций, выполняемых аппаратными средствами СТЗ, и функций программного обеспечения. Излишнее перекладывание функций системы СТЗ на программные средства приводит к снижению ее быстродействия, а ряд функций СТЗ без аппаратной поддержки просто невыполним. Адаптация СТЗ-2М, межкадровая обработка информации, и пространственная фильтрация изображений выполняются специализированными аппаратными модулями, и на выходе системы формируется массив предварительно обработанных отсчетов видеосигналов ПЗС-датчиков изображения, используемый для дальнейшей обработки с помощью ЭВМ.

СТЗ имеет следующие системные шины (рис. 2): 12-разрядную шину адреса (ША), три 8-разрядных шины данных (ШД, ШДТ, ШДМ); 24-разрядную шину синхрониза-

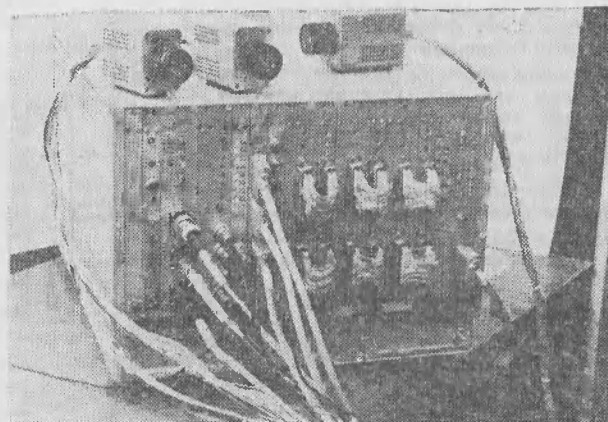


Рис. 1. Внешний вид системы технического зрения СТЗ-2М

* Разработана в Новгородском НИИ электромагнитных компонентов В. А. Стаценко, В. И. Васильевым, А. В. Балигиным, Г. В. Телешовым, В. Н. Глобовым, В. В. Шевченко, В. Б. Карпинским, Ю. В. Ковалевым.

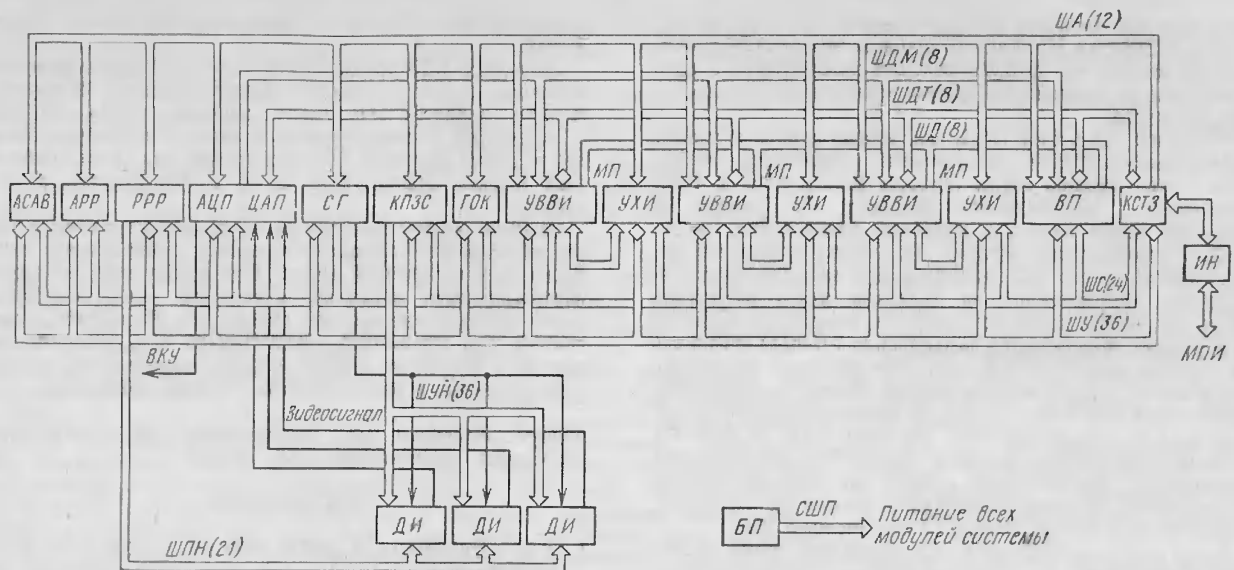


Рис. 2. Структурная схема системы технического зрения

ции (ШС); 16-разрядную шину управления (ШУ); системную шину питания (СШП), образованную потенциальными сигналами.

Кроме названных шин между платами устройств ввода-вывода информации и хранения информации модуля памяти (МП) размещены межблочные шины: по 76 разрядов на каждый модуль памяти, управляющих напряжений (ШУН) по 12 сигналов между контроллером ПЗС (КПЭС) и каждым датчиком изображения (36 сигналов), питающих напряжений (ШПН) по 7 сигналов между модулем регулировки режимов работы (РРР) и каждым датчиком изображения (21 сигнал).

Основные технические характеристики СТЗ-2М

Освещенность на объекте, лк	1...800
Разрядность выходного кода, бит	6
Скорость межкадровой обработки информации, Мбайт/с	4,6
Скорость обмена с внешней микроЭВМ «Электроника 60», Мбайт/с	0,09
Максимальная скорость обмена с внешней микроЭВМ, Мбайт/с	0,5
Число реализуемых фильтров пространственных частот, шт.	50
Объем микропрограммного обеспечения, Кбайт	16
Объем памяти, Мбайт	0,5
Формат окна, точ.	360×576
Время кадра, с	0,04...1
Число синхронных датчиков	3
Виды операций межкадровой обработки	16 логических, 16 арифметических
Потребляемая мощность, кВт	0,8
Габаритные размеры, мм:	
датчика изображения	52×74×94
печатной платы модуля системы	233,3×220 206×452×560

Модуль СТЗ (рис. 2)

Датчик изображения (ДИ) включает в себя матричный ПЗС-датчик К1200 ЦМ7, устройство электрического преобразования управляющих импульсов уровня ТТЛ до уровня, необходимых для работы ПЗС, и предварительный видеосигнальный усилитель. По сигналам управления он подключен к КПЭС и СГ, а по питающим напряжениям — к модулю РРР, а коаксиальный кабель для передачи видеосигнала изображения, сформированного датчиком, — к модулю АЦП-ЦАП.

Синхронизатор (СГ) формирует из тактовой кварцевой частоты 21 МГц набор опорных частот и синхронимпульсов, обеспечивающих через системную шину синхронизации согласованную работу модулей СТЗ как в телевизионном, так и малокадровых режимах, кратных телевизионному, с временем кадра 0,04—1 с; управляющие импульсы для микросхемы динамического ОЗУ серии К565 для управления модулями памяти.

С лицевой панели или по команде с внешней ЭВМ генератор «окно» (ГОК) выбирает прямоугольную связную область («окно») в изображении (размеры изменяются от одного элемента до полного кадра). Разработаны программы для ЭВМ типа «Электроника 60». Оператор управляет «окном», либо плавно изменяя его параметры, либо устанавливая их произвольно. Сигнал «окно», сформированный на экране ВКУ, может иметь несколько видов: темное или светлое окно, темный или светлый контуры окна — и используется для стабилизации амплитуды видеосигнала модулем АСАВ.

Аналого-цифровой и цифро-аналоговый преобразователь (АЦП—ЦАП) изменяет видеосигнал аналоговой формы с ПЗС-датчика в 6-разрядный двоичный код и обратно, формирует полный телевизионный сигнал для отображения на ВКУ и сигнала управления модулем.

Автоматическая стабилизация амплитуды видеосигнала (АСАВ) согласует диапазоны динамического видеосигнала и аналого-цифрового преобразователя; регулирует чувствительность ПЗС-датчика изображения при больших уровнях освещенности («пересвете») [1]. АСАВ производится как в телевизионном, так и в малокадровых режимах [2]. Чувствительность ПЗС-датчика определяется изменением времени накопления зарядовых пакетов в пределах времени кадра (25...100%). Регулировка режимов работы (РРР) предназначена для формирования регулирующих питающих напряжений датчиков изображения на ПЗС, которые могут задаваться либо оператором, либо автоматически с модуля АРР.

С помощью автоматической регулировки режимов работы (АРР) оптимизируется частотно-контрастная характеристика ПЗС-датчиков изображения путем изменения напряжения питания в малокадровом или телевизионном режимах и повышения точности настройки датчиков (устраняется влияние дестабилизирующих факторов). ПЗС-датчик настраивается модулем АРР поиском экстре-

муема целевой функции методом покоординатного спуска Гаусса — Зейделя [3]. Универсальность модуля АРР позволяет расширить его функциональные возможности и применить для автоматической фокусировки и диафрагмирования объектива.

Модули памяти (МП) считывают запись и хранят информацию отсчетов видеосигнала с датчиков изображения или внешней ЭВМ, которая фиксируется в телевизионном и малокадровом режимах. Он состоит из устройств ввода-вывода и хранения информации, соединенных межблочными шинами. Возможно проведение параллельного сдвига информации в двух взаимно перпендикулярных направлениях на заданное число элементов. Из-за ограниченного быстродействия микросхем К566РУ5 информация записывается параллельно в четыре страницы. Со стороны внешней ЭВМ обеспечен доступ к произвольной области любого модуля памяти. Время записи-считывания кадра информации в модуль памяти составляет 40 мс.

Видеопроцессор (ВП) на базе АЛУ (К531ИП3) со схемой ускоренного переноса (К531ИП4) обрабатывает информацию, поступающую с АЦП или модулей памяти, осуществляет 16 арифметических и логических операций, управляется от контроллера СТЗ или внешней ЭВМ.

Контроллер ПЗС (КПЗС) управляет тремя датчиками изображения. Его основные режимы работы: накопление информации под комбинацией фаз, задаваемых с внешней ЭВМ; пространственная фильтрация изображения с регулируемыми параметрами [4]; электронная расфокусировка изображения [5].

Контроллер СТЗ (КСТЗ) организует режимы работы СТЗ-2М установкой заданной комбинации кодов состояний модулей по командам с ЭВМ (ведомый режим) и в соответствии с собственной микропрограммой (ведущий режим). Объем памяти микропрограммного обеспечения КСТЗ — 16 Кбайт, скорость обмена информацией с внешней ЭВМ типа «Электроника 60» — 90 Кбайт/с. Интерфейс (ИН) сопрягает шины внешней ЭВМ (МПИ для ЭВМ типа «Электроника 60») и СТЗ.

Блок питания (БП) формирует стабилизированные напряжения всех модулей. Для обеспечения малых пульсаций он выполнен по трансформаторной схеме. В БП предусмотрена защита от короткого замыкания, а также по-

нижения или пропадания одного из питающих напряжений.

Состояние СТЗ определяется взаимодействием модулей, входящих в состав системы. Слово-состояние отдельного модуля задается по шине адресов (ША), системы с микроЭВМ и записывается в регистр состояния модулей во время кадровых гасящих импульсов. Сигнал приоритета модуля формируется КСТЗ и фиксируется по его адресу.

Режимы работы модулей АСАВ и АРР изменяются также по кадровому гасящему импульсу, однако время отработки больше времени кадра. При управлении СТЗ сбои контролируются сигналом «Четность», информирующим слово-состояние по сигналу приоритета. Разработана программа для микроЭВМ «Электроника 60», позволяющая изменять состояние любого модуля, производить начальную установку и распечатывать таблицу состояний.

173000, Новгород, ул. Дмитровская, 20, НИИ Электромагнитных компонентов; тел. 54-5-33

ЛИТЕРАТУРА

1. А. с. 1275788 СССР, МКИ Н04 № 5/1035, 5/16, 5/20. Устройство автоматической стабилизации амплитуды видеосигнала / В. А. Стаценко, Г. В. Телешов, А. В. Балягин, А. П. Афанасьев. — Оpubл. 1986. Бюл. № 45.
2. А. с. 1533014 СССР, МКИ Н04 № 5/20. Устройство автоматической регулировки амплитуды видеосигнала / В. А. Стаценко, В. Н. Васильев, А. В. Балягин, А. П. Афанасьев. — Оpubл. 1989. Бюл. № 48.
3. А. с. 1443204 СССР, МКИ Н04 № 3/14. Устройство формирования сигнала изображения / В. А. Стаценко, А. И. Кириллов, А. В. Балягин, А. П. Афанасьев. — Спубл. 1988. Бюл. № 45.
4. Стаценко В. А. Нерекурсивный фильтр на ПЗС // Радиотехника. — 1988. — № 10. — С. 11—13.
5. А. с. 1451744 СССР, МКИ О06 К 1100. Устройство для считывания информации / В. А. Стаценко, Г. В. Телешов, А. В. Балягин. — Оpubл. 1989. Бюл. № 2.

Статья поступила 30.01.89

УДК 621.865.8:681.586.5

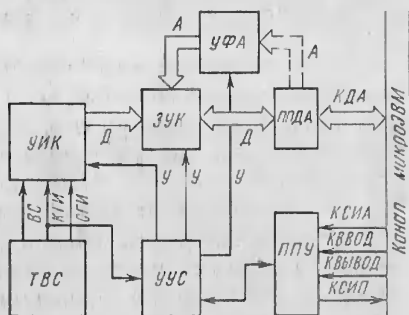
И. И. Дунин-Барковский, В. А. Клевалин

СИСТЕМА ВИЗУАЛЬНОГО ОЧУВСТВЛЕНИЯ ПРОМЫШЛЕННЫХ РОБОТОВ И РТК

Визуальное очувствление промышленных роботов (ПР) и роботизированных технологических комплексов (РТК) на основе систем технического зрения (СТЗ) открывает широкие перспективы повышения эффективности и надежности при расширении их функциональных возможностей [1]. При работе с достаточно контрастными объектами и хорошем освещении целесообразно использовать СТЗ бинарного типа, в которых изображения кодируются по двум градациям яркости. Ряд преимуществ имеет способ кодирования в виде последовательности строчных координат (построчное кодирование), при котором запоминаются координаты перепадов яркости бело-черного и черно-белого вдоль каждой телевизионной строки*. Код изображе-

ния при построчном кодировании, как правило, короче, чем при растровом, так как длина кода при построчном кодировании переменна и зависит от числа перепадов яркости на изображении. Например, при дискретизации изображения 256×256 пикселей построчный код сохраняет преимущество перед растровым, если число перепадов яркости на изображении не более 7000 или около 30 на одной строке. В среднем СТЗ ПР обрабатывает изображения, имеющие 1500...3000 фронтов, поэтому использование построчной системы кодирования вместо растровой дает экономию в требуемом объеме запоминающего устройства в 2,5...5 раз. Использование построчного кода приводит также к увеличению скорости обработки изображения в 1,5...2 раза, причем выигрыш тем больше, чем проще изображение (с меньшим числом перепадов яркости).

Структура СТЗ МСИ-1 с построчным кодированием координат перепадов яркости показана на рисунке. Система



Структурная схема СТЗ МСИ-1: ТВС — телевизионная система на основе малогабаритной телевизионной камеры на ПЗС — матрице 1200ЛМ7; УИК — устройство измерения координат; ЗУК — запоминающее устройство координат; УФС — устройство формирования адреса; ППДА — приемопередатчик данных и адреса; ППУ — устройство управления и синхронизации; КСА, КВВОД, КВЫВОД, КСИП — сигналы канала микроЭВМ; ВС — видеосигнал; КГИ, СГИ — кадровый и строчный гасящие импульсы А — адрес; Д — данные; У — сигналы управления

* Подобный метод кодирования изображения (RI-КОД) был предложен в работе [2], а аппаратная реализация описана в Институте технической кибернетики Словацкой АН [3].

аппаратно совместима с Q-шиной и выполнена в виде полуплаты конструкции «Электроника 60». Объем ЗУК составляет 4096 9-разрядных слов при дискретизации изображения 400×288 элементов. ЗУК доступно со стороны процессора микроЭВМ по считыванию и записи и включается в адресное пространство одним из трех способов: занимает адреса одного из банков ОЗУ (номер банка задается устанавливаемой перемычкой) или области внешних устройств 166000...167100 (доступно в страничном режиме); организовано по принципу стека (использует два изменяемых адреса в области адресов внешних устройств).

Способ включения ЗУК зависит от системы управления ПР и используемых программных средств (операционной системы и алгоритмов обработки изображений). Следует отметить, что возможность стекового варианта организации ЗУК — одно из достоинств способа кодирования изображения.

УУС управляет обменом информацией между системой и процессором микроЭВМ через приемопередатчики ППДА и ППУ, работам ЗУК при записи и считывании координат и УИК. Видеосигнал обрабатывается в УИК для выделения перепадов яркости и измерения их строчных координат, которые записываются сплошным массивом в ЗУК.

УФА формирует адреса записи и считывания для ЗУК, поэтому его организация зависит от способа включения ЗУК в адресное пространство микроЭВМ.

Для СТЗ МСИ-1 разработано программно-математическое обеспечение (ПМО), позволяющее вводить и обрабатывать изображения, сегментировать, описывать и распознавать объекты, в том числе соприкасающиеся и частично перекрывающиеся, задавать набор признаков, обходить контур объекта и вычислять его признаки (периметр, направление, кривизну).

ПМО написано на языке Си и построено по модульному принципу.

101472, Москва, Вадковский пер., За, Мосстанки, каф. «Промышленные роботы и РТС»; тел. 289-51-75

ЛИТЕРАТУРА

1. Кулешов В. С., Клевакин В. А., Дунин-Барковский И. И. Система технического зрения промышленного робота модели «Электроника НЦ ТМ-01» // Станки и инструмент. — 1987. — № 5.
2. Rosenfeld A., Kak A. C. Digital picture processing // AP. — N 4. — 1976.
3. Том Д. М. Представление знаний в человеко-машинных и робототехнических системах. — М.: ВИНТИ, 1984. — С. 165 и 241.

Статья поступила 24.02.89

УДК 681.327.23

Д. С. Коврига, С. М. Гладченко

МИКРОПРОЦЕССОРНАЯ ПРИСТАВКА ДЛЯ СТАНКОВ С ЧПУ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА БК»

В промышленности эксплуатируются станки, оснащенные системами ЧПУ типа NC (Numerical Control), обеспечивающие обработку деталей при непосредственном считывании управляющей программы (УП) с перфоленты через фотосчитывающее устройство (ФСУ). Сбои в работе ФСУ — источник многочисленных отказов системы ЧПУ. Для их устранения и расширения возможностей системы ЧПУ (Н22, Н33, Н55, Размер-2, Размер-4) разработана микропроцессорная приставка со следующими функциями:

ввода и хранения УП в ОЗУ ЭВМ через ФСУ станка, бытовой кассетный магнитофон, канал связи с ЭВМ верхнего уровня управления (контролируется формат адресов в кадре УП);

вывода УП в систему ЧПУ с первого или любого указанного кадра (для фрезерных станков автоматически формируются кадры подвода инструмента к детали, поэтому можно продолжить выполнение УП после прерывания обработки детали, вызванного износом или поломкой режущего инструмента);

экранного редактора (можно получить подсказку о назначении функциональных клавиш в виде графического изображения клавиатуры с краткой инструкцией по работе (табл. 1));

графического отображения в непрерывном и «пошаговом» (по кадрам УП) режимах с синхронным отображением текста УП в служебной строке экрана (в режиме «окна» можно выбрать

фрагмент и увеличить до максимально возможного на экране);

обмена с ЭВМ верхнего уровня; вывода УП на магнитную ленту.

Для облегчения обслуживания микроЭВМ «Электроника БК-0010» в состав ПО включена подпрограмма «Монитор команд оператора ЭВМ», язык которой максимально приближен к входному языку режима пультового терминала микроЭВМ «Электроника 60М» (табл. 2). Признак готовности монитора команд к приему очередной команды оператора — символ «a» в первой позиции строки экрана. Числа вводятся в 8-ричной системе счисления (набирать незначащие нули не обязательно).

Для стыковки с устройством ЧПУ используется параллельный интерфейс микроЭВМ (см. рисунок).

Таблица 1

Соответствие символов клавиатуры командам редактора

Символ клавиатуры	Команды
↑	Управления перемещением маркера на экране Сдвиг маркера вверх на одну строку Сдвиг маркера вниз на одну строку Сдвиг маркера влево на одну позицию Сдвиг маркера в начало следующей строки Сдвиг маркера вправо на одну позицию
↓	
←	
→	
↔	
←(ЗБ)	Редактирования Удаление предыдущего символа в строке Сдвиг текста вправо на одну позицию Сдвиг текста влево на одну позицию Вставка пустой строки Поиск начала УП Поиск следующей экранной страницы (24 строки) Поиск предыдущей экранной страницы Поиск по модели Конец работы с редактором
→	
↔	
BC	
CU/R	
+	
KT	

Таблица 2

Соответствие команд монитора выполняемым функциям

Команда	Функция
xxxxxx/ RX/	Открыть ячейку с 8-ричным адресом xxxxxx Открыть регистр «X», где X=0..7 для регистров общего назначения (РОН) или X=S для регистра состояния процессора (RS)
←	Закрывать ячейку или регистр (если было введено новое содержимое, оно будет записано)
↑	Закрывать ячейку или регистр (если был ввод нового значения, то с модификацией содержимого) и открыть предыдущую ячейку или регистр
↓	Закрывать ячейку или регистр (с модификацией) и открыть следующую ячейку или регистр
⊙	Закрывать ячейку или регистр (с модификацией); вычислить адрес ячейки, которую необходимо открыть по формуле: АДРЕС=содержимое текущей ячейки; открыть ячейку с этим адресом
СТОП	Остановить выполнение программы, вывести на экран содержимое РОН и RS
P	Продолжить выполнение программы с адреса, на котором она была прервана командой HALT или клавишей СТОП
ШАГ	Выполнить одну команду программы, вывести на экран содержимое РОН и RS и вернуться в режим монитора команд.
T	Вызвать тест-мониторную систему микроЭВМ

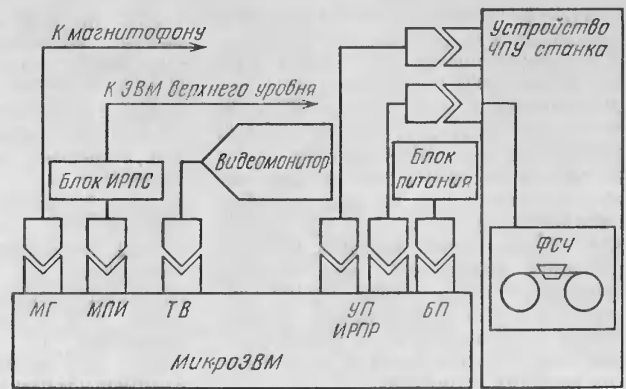


Схема подключения приставки на базе микроЭВМ «Электроника МС510 (МС513)» к станку с ЧПУ

блоков памяти БП7, серийно выпускаемых промышленностью, поэтому применение микропроцессорной приставки с разработанным ПО более предпочтительно.

З30064, Запорожье, ЗПО «Моторостроитель», отдел механизации и автоматизации ОМAB; тел. 61-43-17

ЛИТЕРАТУРА

1. Персональная ЭВМ «Электроника МС 0513» («Электроника БК-0011»). ПО. Драйвер-мониторная система. Руководство системного программиста 00008-01.32.01.
2. МикроЭВМ «Электроника МС 0510» («Электроника БК-0010Ш»). Руководство системного программиста. — Т. 1. — Кн. 2.

Статья поступила 22.4.89

САПР

УДК 681.3.06

С. Е. Богомолов

БЫСТРОЕ ПОЛУЧЕНИЕ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ ДЛЯ МАШИНОСТРОИТЕЛЬНЫХ САПР

Комплекс интерактивных систем двух- и трехмерного геометрического моделирования ГАМАЮН [1] для создания, хранения, визуализации и изменения геометрических моделей изделий был разработан в рамках ПО автоматизированного рабочего места конструктора (АРМ) в ОГК САПР ПО ЗИЛ в 1984—1988 гг.

Подсистема САПР ЗД позволяет создавать элементарные тела (параллелепипед, цилиндр, конус, сферу, пирамиду, призму, эллипсоид, параболоид, гиперболоид, тело вращения, профиль), размещать их в пространстве, создавать на их основе сложные тела с помощью операций объединения, вычитания и пересечения. Проекция и сечения трехмерных объектов изображаются либо в каркасном режиме,

либо с удалением невидимых линий.

В процессе эксплуатации системы ГАМАЮН (с учетом обзора соответствующей литературы) были выдвинуты следующие требования к подсистеме визуализации трехмерных геометрических моделей:

получение изображения трехмерных объектов в параллельной и центральной проекциях с удалением невидимых поверхностей и закраской видимых (в зависимости от освещенности и теней от точечного источника света); производительность при разрешении графического монитора 1024×1024—500...1000 тел/мин (одно тело должно изображаться в течение реального времени);

сопоставимость дополнительных затрат оперативной памяти с объемом

видеопамяти (1—2 Мбайт);

необходимость изображения элементарных тел, поверхностей и линий системы ГАМАЮН; результатов теоретико-множественных операций (объединения, пересечения, вычитания) над элементарными телами; разрезов моделей плоскостями;

легкость включения новых видов элементарных тел из некоторого класса (например, ограниченных кинематическими поверхностями);

быстрое выполнение не только изображения с необходимыми параметрами видовой операции (переноса, поворота, перемещения источника освещения и масштабирования), но и добавления, исключения тела или группы, изменения расположения отдельного тела или группы (заранее неизвестно, какое тело будет двигаться);

возможность выделения на экране указанного тела в течение секунды для сцены произвольной сложности;

получение каркасного изображения без удаления и с удалением невидимых линий;

быстрая выдача растрового изображения для интерактивной работы и линейризованного выхода для других подсистем.

Для сравнения эффективности предлагаемых алгоритмов приведем основные характеристики использованной аппаратуры. АРМ включает в себя растровый графический монитор, алфавитно-цифровой дисплей с клавиатурой и планшет с пером. Система реализована на двух типах АРМ: супермини-ЭВМ и рабочей станции.

Технические характеристики:

	Супермини-ЭВМ	Рабочая станция
Время выполнения оператора $A=A+B$, мкс:		
для целых переменных	3	1,5
для вещественных чисел	3	12,5
Графический монитор	Монохромный	Цветной
Число градаций яркости	7	256
Разрешающая способность, точек	1024×780	800×600
Интерфейс	ТЕКТРОНИХ 4014	Прямой доступ к видеосвязи
Скорость передачи данных, Бод	9600	

Цепочки. Все методы, рассмотренные в данной работе, основаны на использовании z-буфера. Первый алгоритм развивает метод пошагового задания линий, реализуемый в графопостроителях, для изображения поверхностей и тел. Линия в пространстве представляется в виде последовательности элементарных шагов на дискретной кубической сетке параллельно координатным осям и по диагонали. На кодирование каждого шага требуется всего 6 бит (по 2 бит на каждое измерение: вперед, нет движения, назад). Имея две линии (образующую и направляющую), можно построить поверхность (одна линия идет вдоль другой). Для этого, от каждой точки образующей строится линия, совпадающая по форме с направляющей. Совокупность таких линий образует сплошную поверхность (если есть три линии, можно построить тело). У такого тела шесть граней, некоторые из которых могут быть фиктивными. К этой категории относятся обобщенные цилиндрические поверхности.

Излишне вычислять координаты всех точек тела: при длине образующих в 300 точек потребуется вычислить $300 \times 300 \times 300$ (27 млн.) точек, в то время как видны будут только находящиеся на шести поверхностях $300 \times 300 = 540\,000$ точек (в 50 раз меньше).

Яркость каждой точки легко определить, так как направления изменений поверхности в каждой точке известны. Нормаль — векторное произведение двух направлений, а яркость — косинус угла между направлением на источник освещения и нормалью. Примем обычную систему координат [2]. Какие точки видны, тоже легко узнать: координаты каждой точки умножаются на произведение матрицы экзemplярного преобразования тела и видового преобразования всей сцены

(достаточно пересчитать координаты точек образующих). Для каждой точки экрана хранится яркость и z-координата. Битовая карта заполняется сначала фоновым цветом, а z-буфер — минимальным значением. Если координата z новой точки больше, чем значение z-буфера, то новая точка ближе к нам, чем старая, и ее параметры надо занести в z-буфер и битовую карту. После обработки всех объектов в ней расположены яркости каждой точки экрана.

Довольно просто реализуется идентификация (необходим дополнительный объем памяти). Для каждой точки



Направление в точке P

Направление в точке P с учетом окрестности 1

Направление в точке P с учетом окрестности 2

Направление в точке P с учетом окрестности 3

Рис. 1



Рис. 2



Рис. 3

экрана запоминается индекс объекта, который «виден» в ней. Получив координаты с планшета или другого устройства, достаем из таблицы индекс объекта. Можно организовать выделение идентифицированного объекта (стирая и рисуя все пиксели с данным индексом). Достаточно хранить индекс объекта не для каждой точки экрана, а для целого прямоугольника (необходимый объем памяти резко уменьшается). Каркасное и сеточное изображения как с удалением невидимых ребер, так и без него реализуются очевидным образом.

Важная особенность метода — каждый объект может иметь свою собственную матрицу экземплярных преобразований. Это упрощает и ускоряет процедуры генерации внутреннего представления. Для изменения расположения тела внутри сцены достаточно изменить его матрицу экземплярных преобразований. Этот прием использован в PHIGS [7].

В процессе реализации системы пришлось решать проблемы, связанные с методом, и бороться с недостатками аппаратуры. Во-первых, даже прямые линии рисуются зубренными. При каркасном изображении с этим еще можно мириться, однако при резких изменениях яркости плоская грань параллелепипеда превращается в пестрый наряд (хотя средняя яркость вычисляется правильно). Поэтому при расчете нормали нужно вычислять направления не в данной точке, а в некоторой ее окрестности (рис. 1). Можно сгладить яркость, взяв достаточно большой базис для вычисления направления. Вторая проблема — дырки, возникающие при изображении диагональных линий (рис. 2). При повороте изображения цепочка может разорваться. Шаги по диагонали можно запретить, заменив, например, шагами вперед и вверх на две по-

зиции (рис. 3). Однако линия утолщается и (как выяснилось из последующих экспериментов) все равно рвется, хотя дырки становятся единичными. Рассмотрим алгоритм «закрашивания» единичных дырок, работающий непосредственно на битовой карте. Вся битовая карта просматривается построчно снизу вверх (рис. 4), дырка заде-

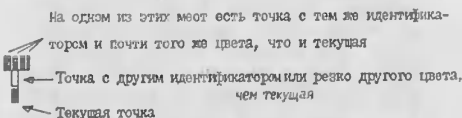


Рис. 4

Пошаговые преобразования

Самая главная проблема — класс изображаемых объектов очень беден. Невозможно нарисовать круг или сферу. Расширим модель следующим образом: поверхность (тело) по-прежнему формируется заметанием образующей линии (поверхности) по направляющей, вместо тела рассматриваются лишь шесть его граней, а вместо явных цепочек — конструктивные ломаные, определяемые начальной точкой, матрицей преобразования координат (координаты следующей точки ломаной получаются умножением координат текущей точки на эту матрицу шага) и числом шагов.

Окружность вокруг оси Z задается следующими параметрами: начальной точкой на оси X, матрицей шага (матрицей поворота вокруг оси Z на некоторый малый угол) и числом шагов, достаточным, чтобы описать полную окружность.

Понятие поверхности расширяется аналогично: можно не только двигать образующую линию по явно заданной направляющей, но и подвергать обра-

зующую линию пошаговым преобразованиями. Например, нарисуем внутренность круга в центре координат вокруг оси Z. В качестве образующей возьмем отрезок. Точка в центре координат один раз подвергается шаговому преобразованию — сместится по оси X на радиус круга (матрица такого преобразования очевидна). Образующая поворачивается вокруг оси Z на малый угол (матрица шага аналогична матрице, использованной при построении окружности) необходимое число раз.

Для удобства написания программ генерации внутреннего представления различных типов тел с образующей связывается матрица трансформации, применяемая для преобразования рассчитанных координат точек линии перед ее использованием при развертке поверхности. Например, эллипсоид вращения можно задать следующим образом: получить круг, подвергнуть получившуюся поверхность сжатию по оси Y (внутренность эллипса), а затем пошагово вращать его вокруг оси X. Матрица трансформации тела используется для размещения эллипсоида в определенном месте, под определенным углом и для трансформации в эллипсоид общего вида.

Класс конструируемых таким образом объектов достаточно широк, в частности, в него входят параллелепипед, шар, цилиндр, конус, тор, эллипсоид, винтовая поверхность, произвольное тело вращения или профиль, для которых можно задать сечение. Любое выпуклое сечение задается своей границей (в крайнем случае в виде явной цепочки) и матрицей шага, схлопывающей контур к диаметру сечения.

Нормаль в каждой точке вычисляется как векторное произведение двух направлений изменения поверхности. Эти направления выражены в вещественных числах, поэтому нормали можно не сглаживать. Однако недостаточно вычислить и нарисовать опорные точки поверхности, необходимо закрасить пространственные четырехугольники между опорными точками (рис. 5). Четырехугольник разбивается на два заведомо плоских треугольника, каждый из которых закрашивается.

Архитектура подсистемы визуализации изображена на рис. 6. При растровой развертке отрезок проверяется на невырождение в точку на плоскости OXY, производится отсечение по алгоритму Куэна — Сазерленда, затем пиксели отрезка генерируются по алгоритму Брезенхема, дополненному вы-

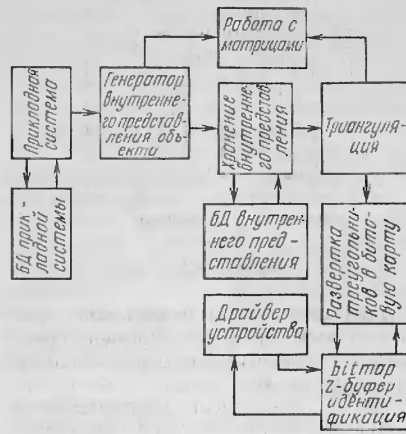


Рис. 6

числением Z — координаты для каждого пикселя.

При заливке треугольников проверяется возможность ее замены рисованием одной или двух его сторон (для очень узких треугольников). Предполагалось, что две стороны треугольника рисуются быстрее, чем заливается исходный узкий треугольник. Первоначально в качестве алгоритма отсечения был принят алгоритм Сазерленда — Ходгмана (соптимизированный для отсечения по прямоугольному окну), после которого используется алгоритм построчного сканирования. Горизонтальные ребра удаляются, остальные сортируются [2]. В нашем случае многоугольник выпуклый, поэтому в таблице активных ребер (ТАР) всегда два ребра, которые можно не укорачивать, в результате алгоритм поддержания и использования ТАР упрощается и значительно ускоряется. При генерации очередной точки для нее вычисляется z-координата из коэффициентов плоскости треугольника и x-, y-координат точки. Эволюция программы заливки треугольников

При создании изображений с радиусами сфер 25 и 50, радиусом цилиндра 25 и длиной 100 сцена преобразуется в 3794 треугольника, из которых 1167 оказываются вырожденными. При заливке треугольников делается 62 000 попыток поставить точку, из них 17 000 удачных. Однако на самом деле достаточно было поставить 10 000 точек. На расчет этой сцены и сцены с поворотом вокруг оси на 45 град центральным процессором (ЦП) было затрачено 21,6 с времени.

Сделано несколько попыток оптимизации программ. В качестве неожиданного результата обнаружено, что увеличение тщательности проверок на вырожденность треугольника уменьшает эффективность системы. Замена алгоритма отсечения Сазерленда — Ходгмана на алгоритм Ходорковского [3] дает выигрыш в 7 %.

Треугольники можно не подвергать предварительному отсечению. Тогда вместо алгоритма заливки выпуклого многоугольника можно использовать заливку треугольника с определением нижней и верхней строк и усложнением нахождения начала и конца каждой строки. Зато не надо делать отсечение, сортировать ребра, модифицировать ТАР. Действительно, существуют четыре случая расположения треугольника (рис. 7). Треугольник представляется в виде одного или двух углов. Угол — это треугольник с горизонтальной стороной. Есть два варианта углов с одинаковой программой заливки угла (рис. 8). Заливка треугольника превращается в заливку одного или двух углов (рис. 9), причем для второго угла надо пересчитывать формулу только одной границы. Затраты времени ЦП (10,8 с) раскладываются следующим образом: из внутреннего представления в треугольники — 24 %, из треугольников и отрезков в растр — 47 %, из растра в команды устройства — 27 %. Специализированная, квазиоптимальная программа для рисования сферы работает лишь в два раза быстрее. В дальнейшем за счет локальной оптимизации и перепибливания нескольких малых подпрограмм на ассемблере

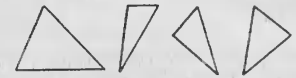


Рис. 7



Рис. 8

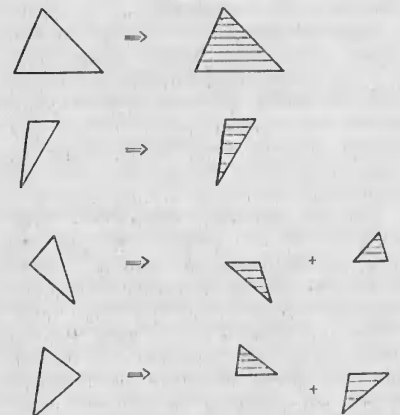


Рис. 9

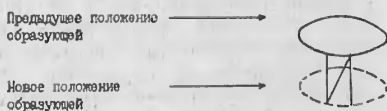


Рис. 5

(выполнено Сидоренко С. В.) удалось уменьшить затраты времени ЦП до 5,6 с.

Практическое применение

В январе 1988 г. подсистема визуализации была встроена в систему САПР ЗД (результат теоретико-множественных операций над телами не изображается).

В марте 1988 г. А. В. Ляпин добавил в САПР 2.5D [4] режим изображения объектов с раскраской поверхностей. Он использовал только часть подсистемы визуализации, а именно раскраску четырехугольников. Дело в том, что основной объект системы САПР 2.5D — это поверхность произвольной формы, и шаговая развертка здесь невозможна. Им был написан собственный генератор четырехугольников для сплайн-поверхностей. Результат превзошел все ожидания: для раскраски поверхностей с учетом освещенности и удалением невидимых поверхностей требуется в три раза меньше времени ЦП, чем для алгоритма удаления невидимых линий, взятого из системы ЗИГЗАГ [5] и использовавшегося в системе САПР 2.5D для получения каркасного изображения.

В апреле Н. Б. Стригина перенесла систему на рабочую станцию Беста-88. Куб $400 \times 400 \times 400$ точ. раскрашивается за 10 с.

Изображение теоретико-множественных операций над телами

Напомним, что одно из требований к подсистеме визуализации — изображение результатов теоретико-множественных операций (объединение, пересечение, вычитание) над элементарными телами, разрезов моделей плоскостями.

Объединение тел получается при использовании z-буфера естественным образом. Разрез одной или несколькими плоскостями (частями плоскостей) реализуется легко, хотя и увеличивает время заливки треугольников. Делается это так: после выяснения необходимости занесения точки в z-буфер проверяют ее положение перед плоскостью. При положительном ответе ставится z-координата и цвет плоскости. Если задняя поверхность тела находится перед плоскостью, цвет не изменяется.

Что же делать с вычитанием или пересечением произвольных тел? z-буфер здесь не поможет. На рис. 10 приведен простейший пример. Пусть тела поступают в следующем порядке: большой куб, малый куб, цилиндрическое отверстие в малом кубе. В этом случае, вычитая цилиндр из малого куба, в битовой карте необходимо восстановить цвета большого куба, информация о котором уже утеряна. Неизвестно даже, что он вообще был!

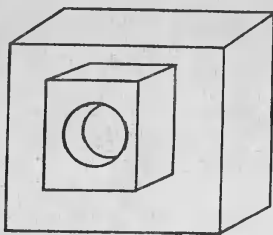


Рис. 10

Возможно несколько подходов к решению этой задачи. Например, предлагается использовать dexels: хранить не только самую большую z-координату, но и весь список z-интервалов с соответствующими цветами и индексами объектов [6]. Назовем такую структуру «интервальный z-буфер». Тогда вычитание тела — это вычитание интервалов, а пересечение тел — пересечение интервалов. При размере экрана 1024×1024 точек резервировать для каждой точки фиксированный объем памяти под хранение интервалов неразумно (где-то будет густо, где-то пусто), поэтому заведем линейный односвязный список. На адрес уйдет не менее 3 байт. В интервальном z-буфере для каждой точки будет храниться адрес ее первого интервала — 3 Мбайт. На каждый интервал пойдет 10 байт: начало — 2 байт, конец — 2 байт, Индекс — 2 байт, цвет — 1 байт, ссылка вперед — 3 байт. Допустим, что каждая точка описывается в среднем двумя интервалами, получим 23 Мбайт (эти данные можно не хранить между сеансами). Время постановки точки увеличивается на порядок.

С другой стороны, вычитание из одного тела другого (или пересечение тел) можно свести к операциям над границами тел. Например, в системе САПР ЗД поддерживается не только CSG-модель, но и граничное представление. В таком случае встает неприятная проблема раскраски поверхности с дырками произвольных форм.

И наконец, можно не хранить z-интервалы всех точек экрана. Если сразу

превратить все тела в полиэдры и хранить их в таком виде до следующего изменения сцены, можно устройство построения генерацию изображения и хранить интервальный z-буфер только для одной строки или для небольшой группы.

Тел. 277-96-02, Москва

ЛИТЕРАТУРА

1. Богомолов С. Е. Система геометрического моделирования ГАМА-ЮН // Практика автоматизированного проектирования в машиностроении: Сб. науч. тр. / Под ред. В. Д. Кальнера. — М.: Машиностроение, 1989. — Вып. 1. — С. 41—45.
2. Фолли Дж., Ван Дам А. Основы интерактивной машинной графики: В 2-х книгах. Пер. с англ. — М.: Мир, 1985. — Кн. 2.
3. Холорковский А. Н. Алгоритмизация решения геометрических задач машинного синтеза полутонных изображений: Дис. канд. техн. наук. — Киев, 1984.
4. Облаков И. В. Экспериментальная интерактивная графическая система разработки кузовных поверхностей // САПР в автомобилестроении. Сб. науч. тр. / НПО «НИИТавтопром». — М., 1988. — Вып. 1. — С. 13—16.
5. Смирнов А. С. Универсальный визуализатор пространственных геометрических моделей. — М., 1986. — Деп. в ВИНТИ АН СССР 17.06.86, № 4447-Б86.
6. Van Hook Tim Real-time shaded NC milling display // Computer Graphics, August. — 1986. — Vol. 20. — N 4. — P. 15—20.
7. International Organization for Standardization. Information processing systems-computer graphics-programmer's hierarchical interactive graphics system (PHIGS) functional description ISO (1986) Document ISO TC 97/SC 21/WG 2 N 446.

Статья поступила 26.08.88

УДК 681.3.06

С. И. Вдовин, С. В. Цыганков

ПОЛУЧЕНИЕ И РЕДАКТИРОВАНИЕ ЭСКИЗНЫХ ЧЕРТЕЖЕЙ В САПР

Эффективность эксплуатации САПР во многом зависит от того, в какой степени могут пользователи системы опираться на привычные для них принципы и приемы проектирования. Один из таких принципов, называемый далее эскизированием, заключается в том, что изображение технического объекта создают не по заданным или заранее вычисленным размерам, а с помощью глазомера.

Конструкторы во многих случаях предпочитают проводить линии «на глаз», визуально оценивая пропорции проектируемого объекта на соответствие функциональным требованиям и дизайну. То же самое можно делать при работе на ПЭВМ с растровым дисплеем, не имея компьютерной подготовки. Перемещая курсор и используя весьма малую часть клавиатуры, пользователь ПЭВМ может построить изображение

проекции или сечения несложной конструкции и нанести на него размеры в соответствии с правилами машиностроительного черчения.

Автоматизированное получение эскизных чертежей целесообразно не только в конструировании, но и при решении разнообразных технологических задач как альтернатива существующим методам кодирования геометрической информации об объекте производства. Перенос на экран дисплея плоский контур и значения размеров, содержащиеся в исходном документе — чертеже детали, технолог вводит тем самым необходимые данные для подготовки управляющей программы обработки детали, проектирования оптимального раскроя.

Процедура эскизирования реализована на ПЭВМ ДВКЗ в варианте, ориентированном на изображение осесимметричных листоштампованных деталей. Ее назначение — поддержка автоматизированного проектирования технологических процессов и оснастки для изготовления таких деталей. Пользователь может эскизировать образующую внутренней или наружной поверхности детали, содержащую не более 17 участков. Тип участка (отрезок или дуга), направление дуги, излом образующей или сопряжение на границах участков задаются в режиме диалога. Угол наклона образующей U в крайних точках участков можно задавать различными способами.

После ввода названных параметров на экране дисплея возникает изображение очередного участка в виде отрезка или дуги, длиной $L \approx 10$ мм или дуги радиусом $R \approx 10$ мм. Параметры U , L , R можно изменять нажатием клавиш «плюс» и «минус» с дискретностью либо 5° , либо 2,5 мм. Предусмотрена возможность стирания любого числа изображенных участков в случае ошибки.

При задании излома на границах i -го и $(i+1)$ -го участков после изображения последнего выдается запрос о необходимости скругления излома штамповочным радиусом. Положительный ответ служит обращением к подпрограмме скругления излома дугой, радиус которой можно корректировать вышеуказанным образом. Такой способ задания криволинейных участков удобен при изображении деталей, поверхности которых сопрягаются с помощью галтелей.

Заключив изображение образующей внутренней или наружной поверхности

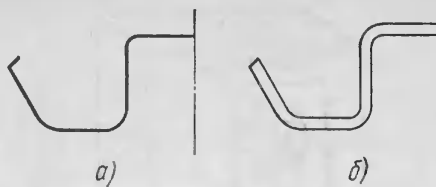


Рис. 1. Последовательность формирования изображения детали

детали, отмечают на ее краю отрезок, длина которого соответствует толщине листового материала (рис. 1, а). Эквидистантная (по отношению к изображенной образующей) линия наносится автоматически (рис. 1, б). Полученное таким образом сечение детали дополняется основными и штриховыми горизонтальными линиями (рис. 1, в), которые, во-первых, могут понадобиться для нанесения размеров, а, во-вторых, придают изображению детали наглядность. Эти линии заканчиваются на оси детали, положение которой предварительно указывается в правой части окна, а после изображения сечения может быть откорректировано.

Нанесение размеров курсором требует от пользователя указания мест размещения первой цифры размера; ввода одного из признаков: D — диаметра, R — радиуса, U — угла, G и B — горизонтального и вертикального размеров соответственно; ввода значения размера; указания точек, от которых автоматически начинают строиться размерные линии. Появление на экране дисплея размерной линии предшествует автоматический перебор крайних точек участков образующей и эквидистанты. Размерная линия идет от точки, координата которой (горизонтальная или вертикальная в зависимости от введенного признака размера) в наибольшей степени совпадает с соответствующей координатой курсора.

При изображении образующей возможно получение дуги, выходящей за пределы одного квадранта. В этом случае точка дуги, оказавшаяся на границе квадрантов, автоматически идентифицируется как граница участка, а вся дуга разбивается на два участка. С помощью этого приема можно получить размерную линию, касающуюся дуги не обязательно в крайней ее точке (например, размер 190 (рис. 2) $\varnothing 640$, размерные линии которого вводятся с признаками точки пересечения, и

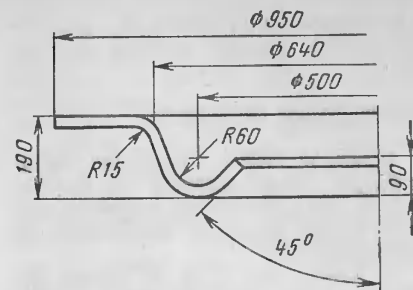


Рис. 2. Пример задания размеров детали

$\varnothing 500$ — точки центра). При этом перебор точек, о котором говорилось выше, организуется на множестве точек пересечения и центра, не принадлежащих образующей или эквидистанте.

Угловые размеры можно задавать не только от оси детали, но и между двумя любыми прямолинейными участками образующей или эквидистанты. Стрелка размера с признаком R принадлежит дуге, указанной курсором.

После нанесения всех необходимых размеров и ввода значения толщины материала можно дать команду на редактирование изображения детали.

Процедура функционирует в среде ОС RT-11 версии 5.0. Проблемный алгоритм выполнен на языке Паскаль. Вспомогательные алгоритмы геометрических построений и получения линий на экране дисплея реализованы на языках Паскаль и ассемблер. Особенности вспомогательного пакета — смешанное использование графического и алфавитно-цифрового экранов. Для исключения искажений, возникающих при неправильном вводе из-за смещения алфавитно-цифрового экрана при работе на последней строке, пакету передается реализация функции ввода с клавиатуры (отпадает проблема незапланированных символов).

Для уменьшения объема загрузочного модуля и увеличения скорости выполнения вспомогательный пакет реализует все построения только в целой арифметике. Весь алгоритм разбит на три задачи, вызываемые автоматически по окончании предыдущей. Загрузочные модули полностью размещаются на одной дискете одинарной плотности.

320197, Днепропетровск, ДГУ; тел. 39-12-85

УСТРОЙСТВА ВВОДА-ВЫВОДА

устройств управления, преобразования, мультиплексирования и т. д. с входной организацией шины типа 8080. К ним относятся программируемый параллельный интерфейс, программируемый таймер, последовательный интерфейс, программируемый контроллер клавиатуры и др.

УДК 681.326

А. Н. Седов, О. Н. Молчанов

УСТРОЙСТВО ФОРМИРОВАНИЯ ШИНЫ ТИПА 8080

Современное направление в области автоматизации научных исследований предполагает широкое использование различных исполнительных устройств, контроллеров,

Принципиальная схема устройства формирования шины типа 8080 для ПК, совместимых с IBM PC

Таблица 1
Карта прошивки ПЗУ КР556РТ4А (D6)

Адрес	Данные
26H	2
27H	9
Все остальные	В

Часто при многоканальности эксперимента от последних не требуется высокое быстродействие, а число их может достигать нескольких десятков.

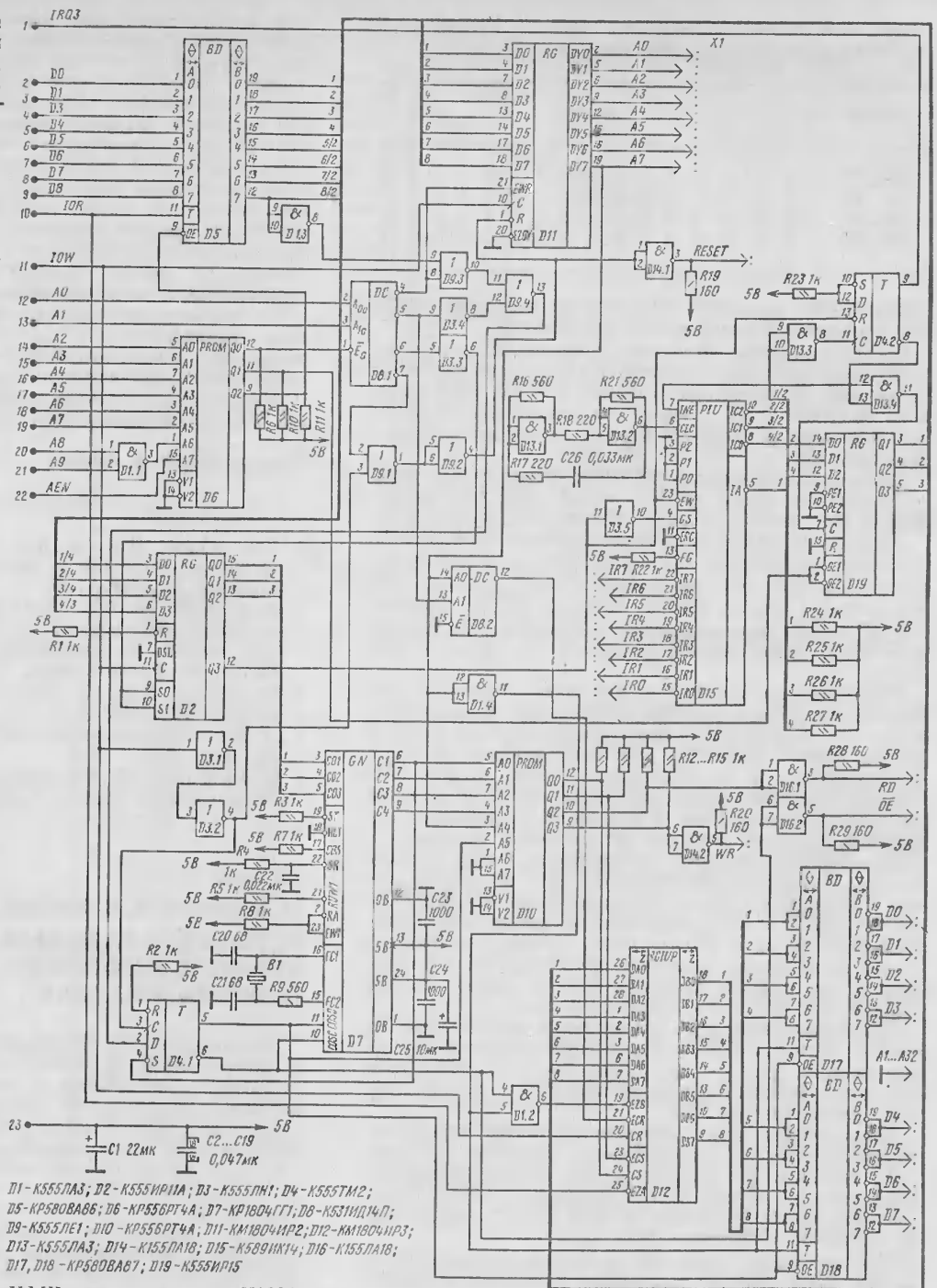
Ограниченность свободных адресов, а также значительные расстояния до исполнительных устройств затрудняют или делают невозможным использование шин персонального компьютера (ПК) для устройств ввода-вывода (УВВ). Не лучше обстоят дела и со входами запросов прерываний, которые, как правило, заняты.

Предлагаемое устройство формирования шины (УФШ) занимает восемь адресов адресного пространства ПК для УВВ и один вход запроса прерывания, предназначенный для второго последовательного интерфейса типа RS232 (COM2, входа IRQ3).

Использование в составе УФШ микросхем серии К1804 обеспечивает гибкость и работоспособность формирователя шины в широком диапазоне скоростей обмена с шиной ПК.

Техническая характеристика УФШ

Разрядность шины адреса	8
Разрядность данных	8
Число приоритетных прерываний	8
Время записи-чтения байта данных шины, мкс. с шагом 0,25	0,75...2,5
Длительность сигнала, мкс	
WR	1,25...3,0
RD	0,75...2,5



D1 - К555ЛА3; D2 - К555ИР11А; D3 - К555ИР1; D4 - К555ТМ2;
D5 - КР580ВА86; D6 - КР556РТ4А; D7 - КР1804ГТ1; D8 - К531ИД14П;
D9 - К555ЛЕ1; D10 - КР556РТ4А; D11 - КМ1804ИР2; D12 - КМ1804ИР3;
D13 - К555ЛА3; D14 - К185ЛА18; D15 - К589ИИ14; D16 - К155ЛА18;
D17, D18 - КР580ВА87; D19 - К555ИР15

Формирователь шины (см. рисунок) содержит: двунаправленный буфер данных D5 и дешифратор адресов шины ПК, реализованный на программируемом ПЗУ (табл. 1) и дешифраторе D8; формирователь признака чтения-записи на элементах D1.3, D9.3, D3.4, D9.4; триггер управления D4.1; регистр скорости обмена D2; регистр адреса D11; двунаправленный порт ввода-вывода D12; управляемый генератор фаз D7; формирователь сигналов управления и сигналов RD, WR, выполненный на программируемом ПЗУ D10 (табл. 2); двунаправленные буферные усилители данных D17, D18; усилители сигналов RESET, RD, WR, OE d D14 D16; блок приоритетных прерываний (БПП) D15; триггер прерываний D4.2; регистр кода пре-

Карта прошивки ПЗУ KP556PT4A(D10)

Адрес	Данные															
00...0F	3	3	3	В	3	3	3	В	3	3	3	В	3	3	3	В
10...1F	3	3	3	3	3	3	3	7	1	1	3	0	3	3	3	7
20...2F	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
30...3F	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

рываний D19; задающий генератор на элементах D13.1, D13.2 и элементы малой логики (остальные).

Работа устройства начинается с инициализации: в регистр скорости обмена записывается 3-разрядный код числа, определяющий длительность сигналов WR, RD, OE шины, а на вход EW БПП — сброс с активным низким уровнем (при этом на входе GS будет активный высокий уровень сигнала).

Сигналы A1, A0 определяют выбор одного из устройств формирователя шины: регистр адреса (00), порт ввода-вывода (01), регистр скорости обмена (10), RESET (11).

При обращении к регистру адреса на входе EWR последнего появляется уровень Log.0 и по положительному фронту сигнала IOW ПК записывается байт данных. Выход регистра адреса открыт сигналом EZDY. Старший бит A7 адреса шины определяет следующий этап обмена: если A7=0, то байт данных выводится; если A7=1 — вводится.

Формирователь признака чтения-записи генерирует сигнал «нужь» в двух случаях: выбран порт ввода-вывода и при этом A7=0 либо выбран регистр адреса и старший разряд данных (вывод 12 D5) равен единице. В том и другом случаях по положительному фронту сигнала IOW триггер управления переходит в состояние Log.0, изменяя состояние сигналов COS O и COS C управляемого генератора, который работает в пошаговом режиме.

Управляемый генератор вырабатывает четыре сигнала C1...C4. Под их воздействием ПЗУ D10 формирует сигнал WR (если A7=0) либо сигналы чтения RD, ECS, CS (A7=1). ECS и CS синхронизируют работу порта ввода-вывода на ввод. Чтение байта данных, записанного в порт ввода-вывода, возможно при обращении к последнему и при наличии активного низкого уровня сигнала IORPK.

При выводе байта шина данных порта ввода-вывода DB открыта сигналом EZB, который вырабатывается инверсным выходом триггера управления. Данные записываются при низком активном уровне сигнала ECR по положительному фронту сигнала IOW.

По окончании цикла синхросерии триггер управления устанавливается в первоначальное состояние (уровень Log.0 на выходе Q1) активным низким уровнем сигнала C1. Обращение по адресу RESET приводит к сбросу шины (высокий активный уровень сигнала RESET). Адрес шины находится в состоянии 00H.

При появлении одного или нескольких прерываний на входах БПП триггер прерывания переходит в единичное состояние и в регистр прерывания записывается код приоритетного прерывания. Выходы регистра находятся в состоянии высокого импеданса. После того как прерывание установлено, его код (три младших разряда байта) может быть считан на шину ПК. При выполнении подпрограммы обслуживания прерываний обязательен модуль инициализации и разблокировки БПП.

Сигнал OEd шины и сигнал A7 можно использовать для управления буферными усилителями данных на некотором расстоянии от ПК при значительной длине кабеля.

Программное обеспечение УФС включает подпрограммы инициализации, инсталляции и деинсталляции прерываний, RESET, ввода-вывода и обработки прерываний.

В начале программы пользователь должен вызвать подпрограммы инициализации и инсталляции прерываний. Подпрограмма инсталляции загружает по адресу, определяемому типом прерывания, указатель (или вектор) прерываний (двойное слово, состоящее из четырех байтов, где находятся указатель команд (первые два байта и базовый адрес подпрограммы обработки прерываний). Так как установленный тип прерываний в ПК может использоваться другими устройствами, подпрограмма инсталляции прерываний перед загрузкой вектора прерываний запоминает текущее содержимое вектора. Подпрограмма деинсталляции восстанавливает сохраненный в памяти предыдущий вектор прерываний, поэтому программа пользователя должна заканчиваться ее вызовом.

Последовательность команд, осуществляющая программное взаимодействие с портами ввода-вывода, должна выполняться полностью. В связи с этим в подпрограммах ввода-вывода организовано запрещение прерываний.

Подпрограммы написаны на языке ассемблера 8086/8088 для ПК, совместимых с IBM PC/XT/AT.

Телефон 218-93 19, Москва

ЛИТЕРАТУРА

1. Дж. Мик, Дж. Брик. Проектирование микропроцессорных устройств с разрядной модульной организацией. — М.: Мир, 1984.
2. Булгаков С. С., Мещеряков В. М., Новоселов В. В., Шумилов Л. А. Проектирование цифровых систем на комплектах микропрограммируемых БИС. — М.: Радио и связь, 1984.
3. Ю-Чжен Лю, Г. Гибсон. Микропроцессоры семейства 8086/8088. — М.: — Радио и связь, 1987.
4. Л Дао. Программирование микропроцессора 8088. — М.: Мир, 1988.
5. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и мини-ЭВМ. — Л.: Энергоиздат, 1984.

Статья поступила 26.01.89

УДК 681.327

Ю. А. Орестов, Н. Н. Ивонтьев

ЧЕТЫРЕХКАНАЛЬНЫЙ АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ

Предназначен для преобразования аналоговых сигналов в измерительных и управляющих системах в кодировый эквивалент. Применяется в составе микроконтроллера и позволяет гальванически развязать входные каналы от цепей межмодульной магистрали, обеспечивает подключение схем АЦП к центральному процессору.

Состоит из программируемого устройства ввода-вывода параллельной информации (микросхема KP580BB55A (D1) на рис. 1); гальванических узлов развязки A1...A4, A9...A14, включающих буферные каскады K555ЛН1 (D1, D3) и оптоэлектронные усилители АОТ101АС (D2) (рис. 2), уменьшающие влияние помех на цепи аналоговых сигналов и предохраняющие контроллер от случайного проникновения опасных электрических напряжений со стороны аналогового ввода; аналого-цифровых преобразователей A5...A8 на основе согласующих прецизионных операционных усилителей K143УД14А (DA1), имеющих малое напряжение и низкий температурный дрейф смещения, и АЦП K1108ПВ2А. Выводы одноименных разрядов соединены между собой. Схема одного узла преобразования приведена на рис. 3.

Микросхема K1108ПВ2А представляет собой 12-разрядный прецизионный, быстродействующий, функционально-законченный АЦП последовательного приближения с временем преобразования 2 мкс. БИС АЦП содержит на кристалле: ЦАП, компаратор, регистр последовательного при-

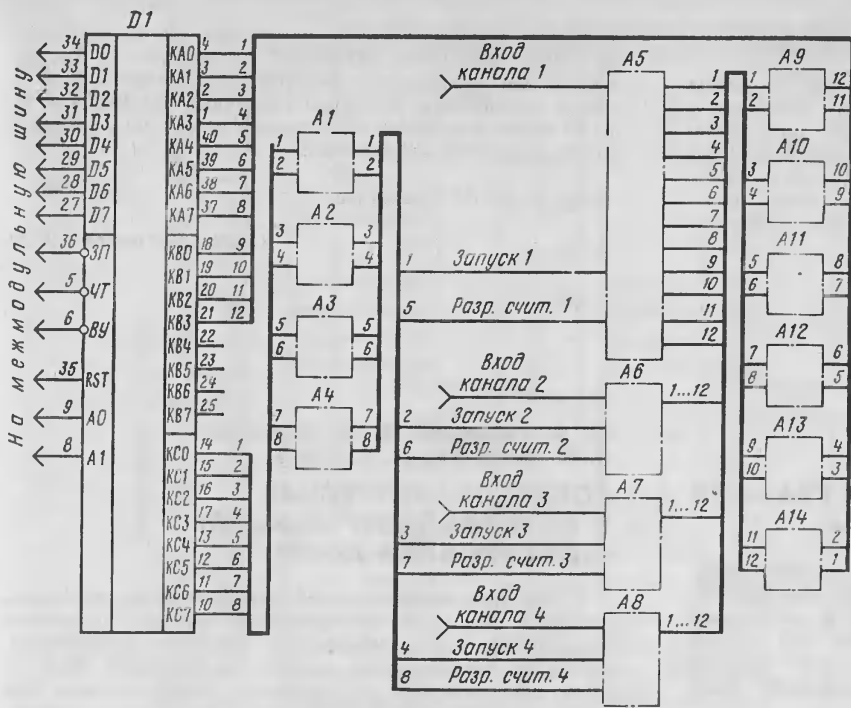


Рис. 1. Принципиальная схема 4-канального аналогоцифрового преобразователя

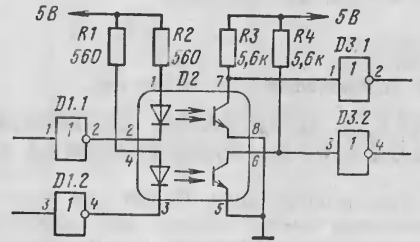


Рис. 2. Принципиальная схема узла гальванической развязки

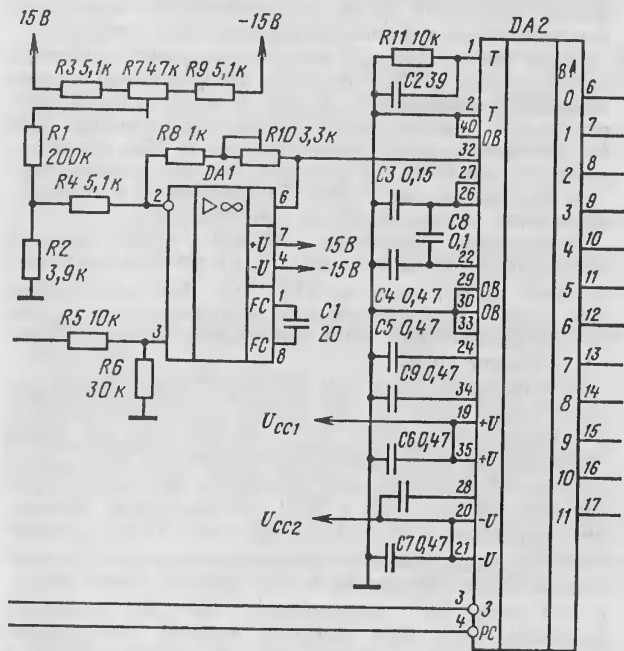


Рис. 3. Принципиальная схема узла преобразования



Рис. 4. Схема включения ИМС К1108PB2

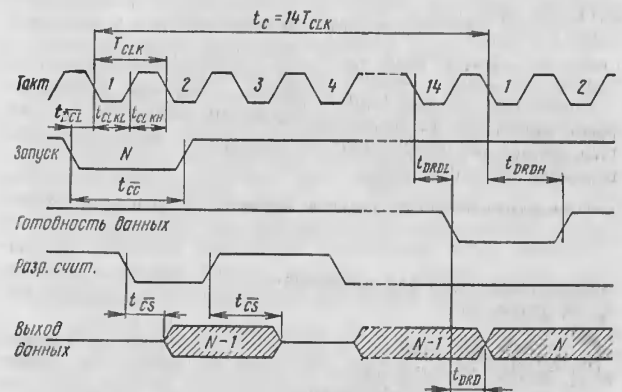


Рис. 5. Временные диаграммы работы ИМС К1108PB2

лижения, источник опорного напряжения, тактовый генератор, схему правления в выходной регистр на три состояния с хранением информации в течение последующего цикла преобразования. Включается по схеме с внутренним тактированием, внутренним опорным генератором, внешними сигналами «запуск» и «разрешение считывания», формируемыми программно (рис. 4). Временные диаграммы работы АЦП приведены на рис. 5.

Процесс преобразования аналоговых сигналов в код каждого канала АЦП складывается из трех циклов: последовательной подачи сигналов запуска и разрешения

считывания с порта КС (D1), запрограммированного на вывод информации;

преобразования аналогового сигнала в код; в момент преобразования одного АЦП выходы других находятся в высокоимпеданском состоянии и не нагружают друг друга;

считывания преобразованных данных на межмодульную шину данных контроллера через порт КА, КВ (D1), запрограммированные на ввод по сигналам «разр. счит.», которые переключают выходы схем АЦП из высокоимпеданского состояния в активное.

Время преобразования ограничено быстродействием оптронных усилителей АОТ101АС и составляет 70 мкс для одного канала; диапазон входного напряжения — 0...5 В, число устойчивых выходных двоичных разрядов — 11. АЦП имеет модульную конструкцию и размещен на одной плате размерами 220×232 мм.

Телефон 468-81-75, Москва

Статья поступила 6.07.88

КРАТКИЕ СООБЩЕНИЯ

УДК 681.3

В. Н. Чернявский, Н. И. Дегтярь

ПАКЕТ ПРОГРАММ ДЕЛОВОЙ ГРАФИКИ ДЛЯ ПЭВМ «ЭЛЕКТРОНИКА 85»

При использовании ПЭВМ для автоматизации учреждений работ полезно для конечного пользователя представлять зависимости между данными в графической форме. Поэтому в состав прикладного ПО ПЭВМ «Электроника 85» включен пакет деловой графики ГРАФ-85, функционирующий в операционной среде ИНМОС-85.

Исходные данные для построения графиков организуются средствами пакета в виде ячеек электронной таблицы.

Кроме того, пакетом ГРАФ-85 могут быть обработаны данные, подготовленные пакетами программ табличной обработки ТАБЛИЦА-85 и базами данных реляционного типа БАЗА-85.

Пакет ГРАФ-85 предоставляет пользователю следующие возможности: формировать и хранить исходные данные в виде таблицы; создавать различные типы графиков; оформлять графики, в том числе вывод заголовка, названий осей, установку геометрических параметров, ручное масштабирование, выделение указанного сектора круговой диаграммы; хранить описания графиков на магнитных носителях и быстро восстанавливать изображения графиков по имеющимся описаниям; корректировать описания графиков; выводить графики на экран дисплея и печатающее устройство. Кроме того, он обеспечивает создание следующих типов графиков: столбиковая диаграмма; составная столбиковая диаграмма; график рассеивания; точечный график в осях х/у; линейный график; круговая диаграмма.

Наличие развитых средств взаимодействия с пользователем в виде меню, функциональных клавиш подсказок и вывода помощи позволяет пользователю-непрограммисту быстро освоить работу с пакетом. Операции выбираются посредством меню и функциональной клавиатуры. В пакете реализована трехуровневая система меню. Головное меню пакета размещается в верхней части экрана и присутствует там постоянно. При выборе команды головного меню в левой части экрана появляется вспомогательное меню, детализирующее данную команду, которое удаляется с экрана после выполнения выбранной функции. На любом этапе работы с пакетом можно получить необходимые пояснения, используя функциональную клавишу вывода помощи.

ГРАФ-85 работает в среде ИНМОС-85 версии 1.2 и выше. Тексты программ пакета написаны на языке Си. Пакет может поставляться самостоятельно или входить в состав интегрированного пакета совместно с пакетом ТАБЛИЦА-85. С третьего квартала 1988 г. пакет ГРАФ-85 поставляется фондом алгоритмов и программ (ВС ФАП) при Бердянском филиале ИПИ АН СССР.

Телефон 3-63-83, Бердянск

Сообщение поступило 26.08.88

УДК 681.327.8

Ю. Ч. Гайдукевич, В. М. Марченко,
Н. И. Домаренок, И. Г. Мороз

СОГЛАСОВАНИЕ ЕС1840 С ИНТЕРФЕЙСОМ микроЭВМ «ЭЛЕКТРОНИКА ДЗ-28»

С развитием вычислительной техники в приборах, используемых в физических экспериментах, системах управления технологическими процессами и обработки информации, возникает необходимость замены управляющей ЭВМ на более совершенную с расширенными возможностями. Для этого интерфейс новой ЭВМ нужно согласовать с интерфейсом микроЭВМ ДЗ-28, что позволяет не производить конструктивную модернизацию физического прибора.

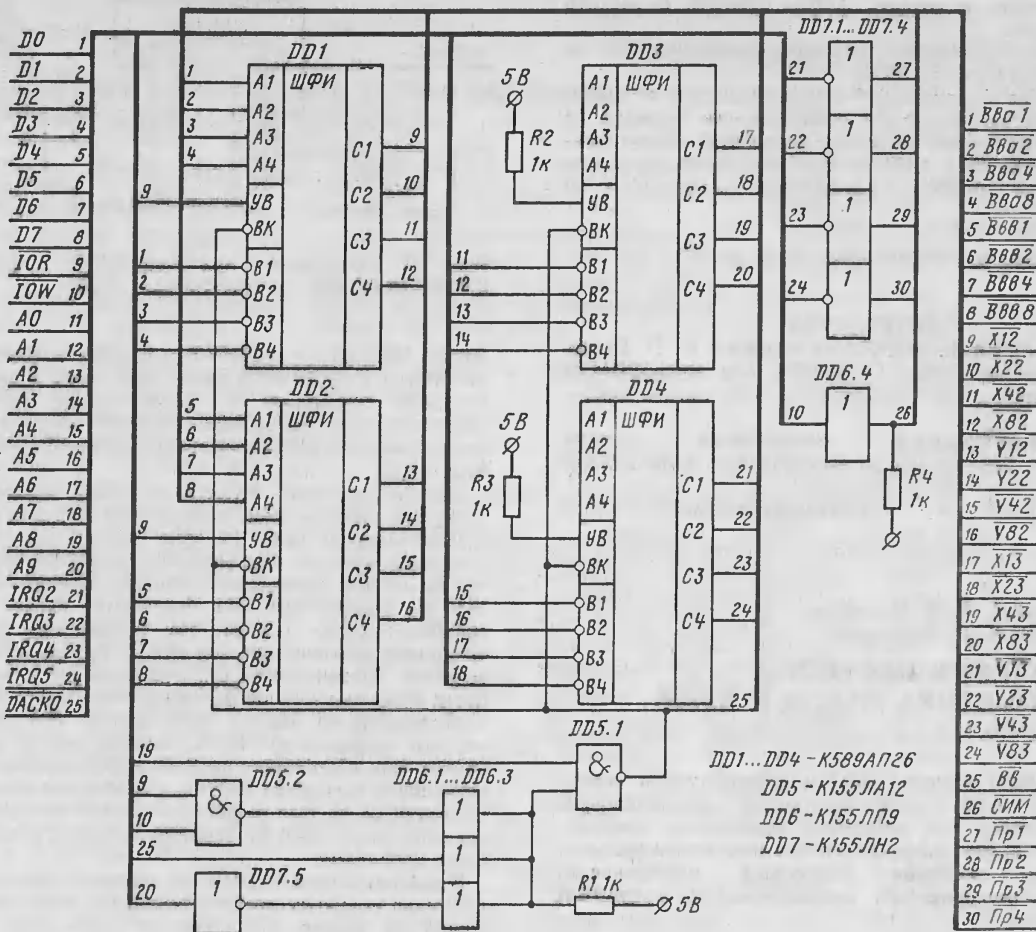
Такая задача возникла для тепловизионной параметрической системы (ТПС) ИИ-42Т, в которой использовалась микроЭВМ «Электроника ДЗ-28». В качестве новой вычислительной машины была выбрана персональная ЭВМ ЕС1840. Для согласования их интерфейсов была разработана переходная плата (см. рисунок), которая устанавливается в блок процессора ЕС1840 и подключается к системной шине через свободный разъем СМП34С-135.

Особенность интерфейса ЕС1840 — обмен данными происходит по одной шине D0...D7, а в ДЗ-28 по двум: ввод по ВвА1...ВвВ8, вывод по X12...Y82. Для согласования шин данных были использованы двунаправленные шинные формирователи (DD1, DD2). Направление передачи задается сигналом IOR.

В качестве сигналов X13...X83 (УПР) используются сигналы адреса порта A0...A9 (порты с адресами от 256 до 511 соответствуют УПР от 0 до 255). Сигнал адреса инвертируется на DD3, DD4. На микросхемах DD5.1, DD5.2, DD6.1...DD7.5 собран селектор адреса, куда через DD5.2 попадают сигналы IOR и IOW, определяющие программное обращение к портам, а через буфер DD6.3 — сигнал DACKO, фиксирующий отсутствие регенерации в данный момент. Таким образом, на выходе элемента DD5.1 формируется сигнал Вв, разрешающий выборку кристалла микросхем DD1, DD4 только в моменты программного обращения к портам 256...511, на элементе DD6.4 создается сигнал IOW из СИМ ДЗ-28 и вырабатываются сигналы прерывания IRQ2...IROQ5 на DD7.1...DD7.4 при инвертировании Pr1...Pr4.

МикроЭВМ «Электроника ДЗ-28» требует обязательного подтверждения готовности устройства сигналом СИП. В ЕС1840 готовность определяется по входному сигналу I/O CH RDY с временем ожидания не более 2,5 мкс. Если длительность сигнала I/O CH RDY больше этого интервала, информация в памяти разрушается из-за задержки цикла регенерации. Сигнал I/O CH RDY может отсутствовать, но при этом время отклика внешнего устройства не должно превышать 1 мкс, что не трудно обеспечить. Для устройств с большим временем ожидания сигнала готовности СИП можно завести на прерывание или подсоеди-

Интерфейс ЕС1840



Принципиальная схема блока согласования интерфейсов

Интерфейс микроЭВМ «Электроника ДЗ-28»

динить к биту порта ввода с последующим программным анализом.

Подключение ЕС1840 к ТПС ИИ-42Т позволило повысить ее быстродействие, использовать при разработке ПО языки высокого уровня, упростить процесс управления при работе в интерактивном режиме с экранним меню. Появилась возможность накапливать большое число изображений в памяти ЭВМ и записывать их на магнитный диск (44 изображения размерами 128×128 8-битных отсчетов на один гибкий диск), а также выводить тепловые изображения на печатающее устройство. Расширились графические возможности системы и пространственно-частотная фильтрация изображений.

220600, Минск, ул. П. Бровки, 6, МРИ, каф. ТРЭИ; тел. 39-88-62

Статья поступила 22.02.89

УДК 681.3.06

А. С. Зайка, И. В. Михайленко

АДАПТАЦИЯ СИСТЕМЫ QUASIC-2 ДЛЯ МИКРОЭВМ С КАССЕТНЫМ НАКОПИТЕЛЕМ

Широкое распространение в системах автоматизации эксперимента получила система программирования QUASIC, обеспечивающая разработку и выполнение программ на

микроЭВМ «Электроника 60», ДВК и им подобных [1]. Большая популярность системы объясняется использованием языка высокого уровня, обладающего возможностями ассемблера, и достаточно высокой скоростью выполнения программ. Система QUASIC позволяет работать на микроЭВМ с перфоленточным вводом-выводом, с гибким магнитным диском или на микроЭВМ, связанной линией связи с мини-ЭВМ СМ4.

Однако применение ЭВМ конфигураций накладывает ограничения на систему автоматизации эксперимента. В частности, применение перфоленточного ввода-вывода представляет известные трудности для оператора ЭВМ. Использование линии связи с мини-ЭВМ позволяет значительно повысить сервис и вычислительные возможности, но в связи с тем, что существует дефицит на это оборудование, использование данных конфигураций ограничено.

Для разрешения этих противоречий в целях обеспечения экспериментов в ФТИНТ АН УССР были использованы микроЭВМ «Электроника 60», оснащенные кассетным накопителем РК-1, под управлением системы QUASIC-2 в среде ТМОС (тест-мониторная ОС) [3]. Выбор системы ТМОС обусловлен удобством корректировки, сохранения и загрузки программ с кассетного магнитофона.

Для системы QUASIC разработана программа, реализующая простой интерфейс между программами пользователя на языке QUASIC, системой QUASIC и накопителем на магнитной ленте. Данная программа выполняет запись блока данных; пропуск файла назад и вперед; пропуск блока назад и вперед; запись файлового промежутка;

открыть, закрыть и создать файл; записать и считать текст программы.

Реализованные файловые операции ориентированы на файловую структуру ОС RT-1.

Эксплуатация предложенной системы показала ее значительные удобства. Благодаря совместимости формата на кассетных накопителях в составе микроЭВМ «Электроника 60» и мини-ЭВМ СМ4 можно переносить основные вычисления по обработке результатов эксперимента на мини-ЭВМ.

310080, Харьков, ул. Машинистов, д. 8, кв. 27;
тел. 30-03-57

ЛИТЕРАТУРА

1. Подольский Л. И., Лясковский А. П. Система программирования QUASIC-2 для микроЭВМ // Микропроцессорные средства и системы. — 1987. — № 2. — С. 9—11.
2. Тест-мониторная операционная система (ТМОС-2). Инструкция по эксплуатации. 4.098.166 ИЭ

Сообщение поступило 19.10.88

УДК 681.3.015

В. В. Глухенький, А. Н. Макеенко,
В. И. Пенской, В. Д. Ромасевич

ПОДКЛЮЧЕНИЕ ЦВЕТНОГО ВИДЕОМОНИТОРА MC6106 К ПЭВМ «ИСКРА 1030.11»

Многие ПЭВМ «Искра 1030.11» комплектуются черно-белыми дисплеями, хотя контроллер видеомонитора ПЭВМ позволяет без переделок подключить цветной. В настоящее время выпускаются цветные видеомониторы «Электроника MC6106» следующих модификаций: MC6106.01 — бескорпусный, встраиваемый в микроЭВМ

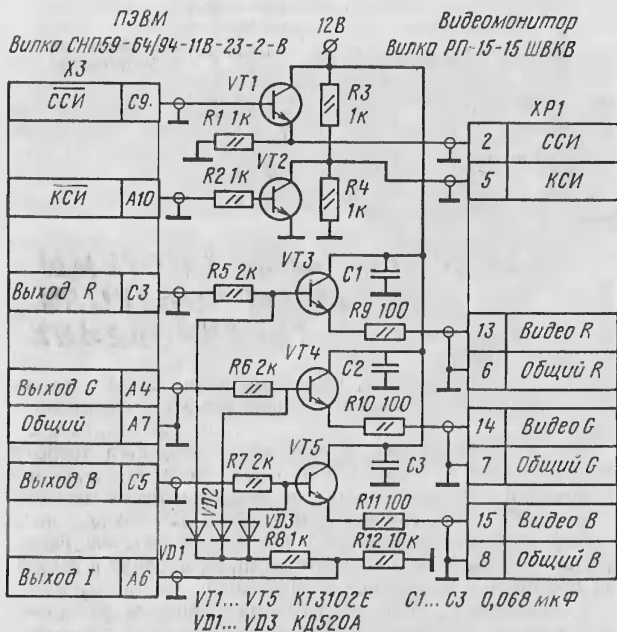


Рис. 1. Схема сопряжения видеомонитора «Электроника MC6106» с ПЭВМ «Искра 1030.11»

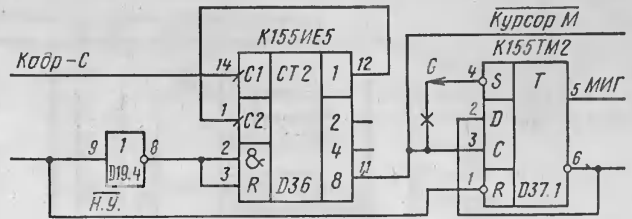


Рис. 2. Фрагмент принципиальной схемы платы КМОИ-25.104.026 ЭЗ контроллера МОИ

ДВК4; MC6106.2 и MC6106.3 — в корпусе. Видеомониторы MC6106.01 и MC6106.03 имеют интерфейс с пятью управляющими сигналами: R, G, B — цвета, КСИ, СИИ — синхронизации; MC6106.02 — полный видеовход, однако переключением двух перемычек его можно превратить в вариант MC6106.03.

Для подключения MC6106 к ПЭВМ «Искра 1030.11» требуется собрать схему сопряжения, легко встраиваемую в видеомонитор (рис. 1). Она состоит из пяти формирователей, которые обеспечивают требуемые уровни сигналов на входах управления MC6106. Эмиттерный повторитель на транзисторе VT1 буферизует сигнал «строчные синхрипульсы» (ССИ). На транзисторе VT2 сигнал «кадровые синхрипульсы» (КСИ) буферизуется и инвертируется. Транзисторы VT3...VT5 формируют видеосигналы цвета в зависимости от интенсивности (выход I) (яркость изображения на экране через диоды VD1...VD3 можно снизить наполовину). Цепь питания схемы сопряжения подключена к источнику питания видеомонитора. Название сигналов и цоколевка разъемов приведены согласно документации [2, 4]. Для устранения помех и взаимных наводок все сигналы от ЭВМ необходимо подключать экранированными проводами.

В видеомониторе MC6106 нет динамика. При размещении динамика в корпусе дисплея возникают искажения изображения на экране. Предлагается разместить динамик в корпусе ПЭВМ, а регулятор громкости вывести на ее заднюю панель.

При работе ПЭВМ «Искра 1030.11» с цветным видеомонитором обнаружен эффект мигания отдельных участков изображения на экране с частотой мигания курсора. Такой паразитный эффект можно устранить, если разъединить перемычку на печатной плате КМОИ-2 ПЭВМ «Искра 1030.11». На рис. 2 приведен фрагмент схемы с перемычкой между выводами 3 и 4 микросхемы D37.1.

Авторы статьи экспериментально установили, что подключение высококачественных фильтрующих емкостей 0,04...0,1 мкФ к цепям питания на платах видеомонитора значительно улучшает качество изображения.

252056, Киев, пр. Победы, 37, КПИ, каф. САПР;
тел. 441-14-53

ЛИТЕРАТУРА

1. Машина персональная профессиональная электронная вычислительная «Искра 1030.11». Техническое описание. 1.320.003.ТО.
2. Машина персональная профессиональная электронная вычислительная «Искра 1030.11.» Альбом схем.
3. Видеомонитор «Электроника MC 6106». (Эксплуатационная документация).
4. Видеомонитор «Электроника MC 6106». Альбом схем.

Статья поступила 29.03.89

УНИВЕРСАЛЬНЫЙ КОНТРОЛЛЕР ТЕРМИНАЛЬНЫХ УСТРОЙСТВ НА БАЗЕ ОЭВМ K1816BE35

Контроллер предназначен для подключения к интерфейсу ИРПС периферийных устройств с параллельным 8 битным интерфейсом (перфоратор, принтер, графопостроитель). Его принципиальная особенность — встроенная ОЭВМ (K1816BE35), диагностирующая подключенное к ней устройство по программе, записанной в РПЗУ (K573PФ2) (см. рисунок). Кроме схемы преобразования интерфейсов ОЭВМ имеет шину управления этим устройством (включение двигателя, опрос готовности, вывод на индикатор возникающих ошибок).

Универсальный синхронно-асинхронный приемопередатчик (УСАПП) обеспечивает прием байта информации в последовательном коде и его выдачу по прерыванию в ОЭВМ. Схема содержит тактовый генератор и делитель частоты синхронизации для УСАПП (от 1200 до 9600 Бод). В контроллере применяются либо приемники и передатчики интерфейса 20 мА петля, либо ИРПС.

Контроллер после включения выполняет программу инициализации: программирует режим работы УСАПП, устанавливает счетчик буфера и выходит в цикл ожидания поступления информации в буферное ОЗУ (занимается по прерыванию от УСАПП). Буфер проверяется на переполнение (выдается символ X-OFF), при его освобождении поддерживается стандартный протокол ИРПС (символ X-ON).

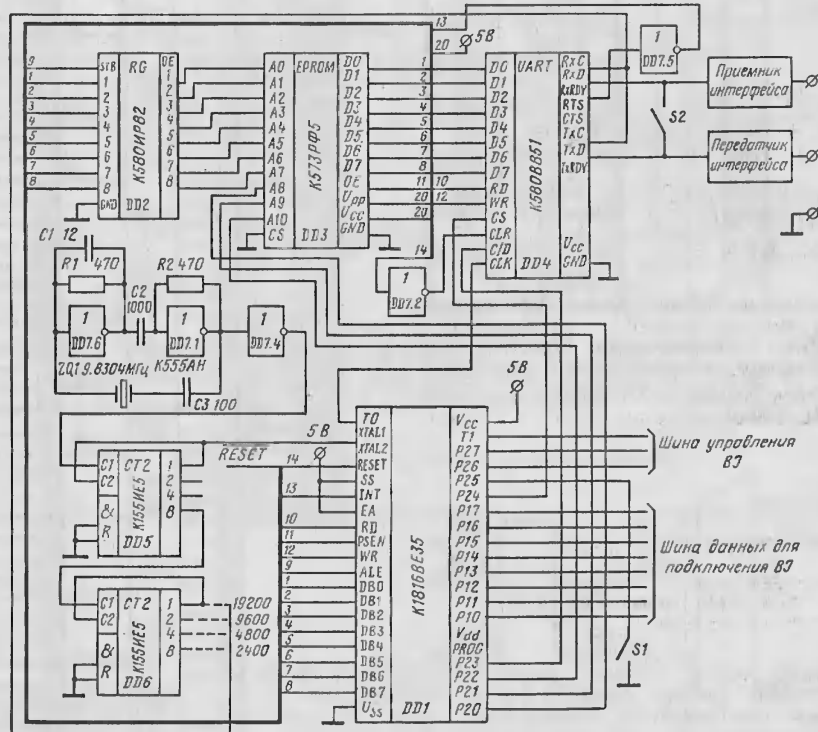
В контроллере возможно самотестирование. Для этого необходимо перед включением питания переключить ключ, коммутирующий шину порта на землю, вторым ключом соединить выводы приема передачи УСАПП.

Принципиальная схема контроллера

Программа вывода информации написана на ассемблере и занимает объем менее 1 Кбайт.

Программа работы ОЭВМ транслируется и отлаживается в ОС РВ с помощью кросс-средств, поэтому ее можно менять для конкретных реализаций, не изменяя при этом схему.

614600, Пермь, ГСП-624, МКБ, ОПЛРТИ



ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ, СИСТЕМЫ КОНТРОЛЯ, ТЕСТИРОВАНИЕ

МНОГОКАНАЛЬНЫЕ ФОРМИРОВАТЕЛИ СИГНАТУР

Для контроля многовыходных цифровых устройств (ЦУ) предлагается использовать формирователи сигнатур (ФС) [1—4], логика работы которых основана на алгоритме ускоренного деления многочленов (рис. 1—3). Сложность комбинационной части многоканального ФС (МФС) определяется числом ненулевых коэффициентов $\Phi(x)$. Величина N на рис. 3 выбрана равной 17. Это позволяет использовать

в качестве образующего примитивный трехчлен.

Данные устройства решают проблему упрощения МФС при большом числе контролируемых входов лишь частично, так как с ростом p растет сложность БПС и, таким образом, снижается быстрдействие формирователя. При $p > N_m$, где N_m — минимально необходимая разрядность сигнатуры, использование в качестве МФС устройства, со-

ЛИТЕРАТУРА

1. Воронин В. И., Макаров К. В., Старшова В. А. Контроллер клавиатуры на базе ОЭВМ K1816BE48 // Микропроцессорные средства и системы. — 1985. — № 3. — С. 36—38.
2. Кобылинский А. В., Липовцев Г. П. Однокристалльные микроЭВМ // Микропроцессорные средства и системы. — 1986. — № 1. — С. 10—19.
3. Петух А. М., Романюк А. Н., Подольский О. А. Преобразователь интерфейсов ИРПС — «Общая шина» // Микропроцессорные средства и системы. — 1985. — № 4. — С. 81—82.

Статья поступила 25.04.89

ответствующего многочлену $\Phi(x)$ степени n , с последующим анализом N_m разрядов регистра снижает эффективность контроля, так как нарушается требование зависимости контрольного кода от всех символов анализируемых последовательностей. В этом случае неисправности объекта, проявляющиеся на последних тестовых наборах и искажающие содержимое не анализируемых разрядов регистра МФС, будут пропущены. При использовании в качестве контрольного кода содержимого всех разрядов регистра МФС резко возрастает объем памяти для хранения эталонных сигнатур. Это усложняет и

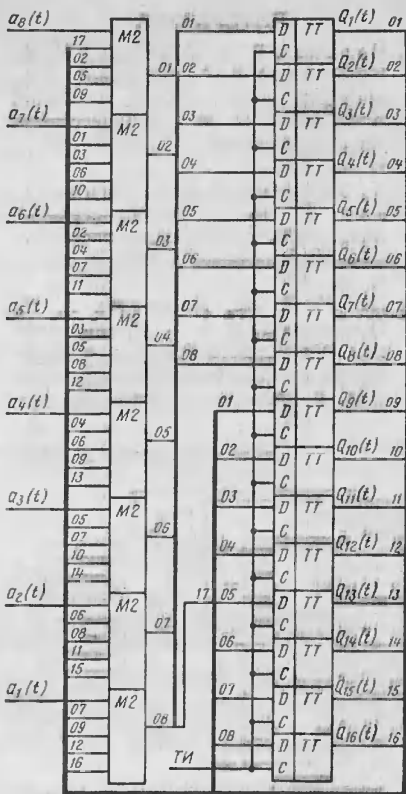


Рис. 1. 8-канальный формирователь сигнатур, соответствующий многочлену $\Phi(x) = x^{16} + x^{12} + x^9 + x^7 + 1$.

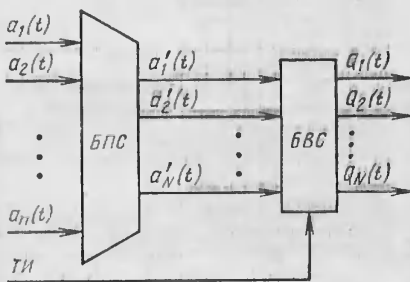


Рис. 2. Структурная схема многоканального формирователя сигнатур при $p > N$:

p — число анализируемых последовательностей;
 N — степень $\Phi(x)$; БПС — блок пространственной сжатия информации; БВС — блок временного сжатия информации

увеличивает стоимость контрольно-испытательной аппаратуры.

Предлагаемый сигнатурный анализатор (рис. 4) свободен от перечисленных недостатков. Сложность комбинационной части МФС минимальна при данном числе контролируемых входов, а разрядность сигнатуры в каждой конкретной ситуации можно выбрать

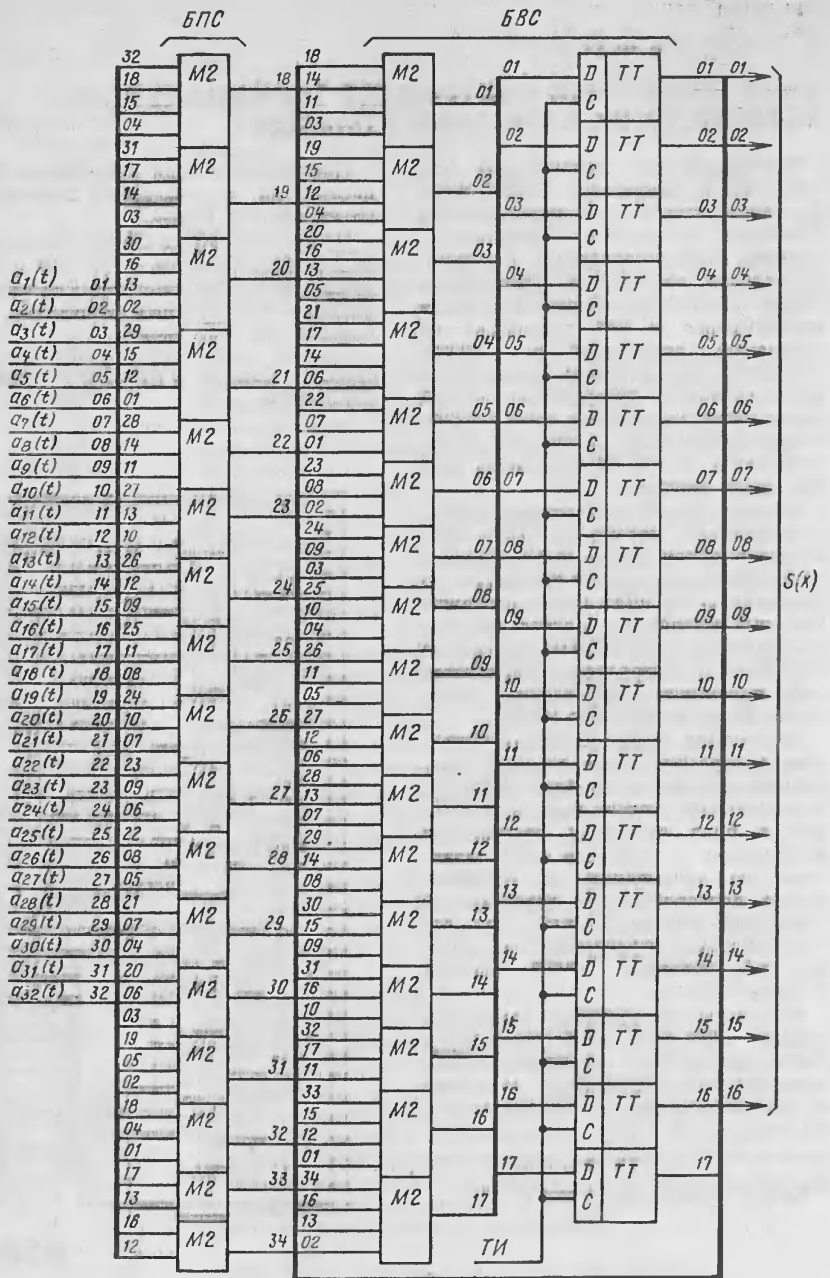


Рис. 3. 32-канальный формирователь сигнатур, соответствующий многочлену $\Phi(x) = x^{17} + x^{14} + 1$.

минимально необходимой для обеспечения требуемой достоверности контроля. Процесс формирования сигнатуры в регистре МФС продолжается и после закрытия «окна измерений». Процедура получения контрольного кода прекращается с возникновением сигнала переопределения на выходе счетчика. Коэффициент пересчета счетчика выбирается таким, чтобы при некорректности в объекте искажения разрядов регистра МФС, не являющихся

контрольными, проявились на выходе устройства. Таким образом, достигается зависимость получаемой сигнатуры от всех символов контролируемых последовательностей.

Рассмотренные формирователи сигнатур обеспечивают высокую достоверность и максимальное быстродействие при контроле и диагностировании многовыходных ЦУ. Они могут использоваться в качестве автономных и встроенных средств тестового и функ-

рис. 4. 32-канальный сигнатурный анализатор:

$$N_m = 16, N = 33, \Phi(x) = x^{33} + x^{26} + 1.$$

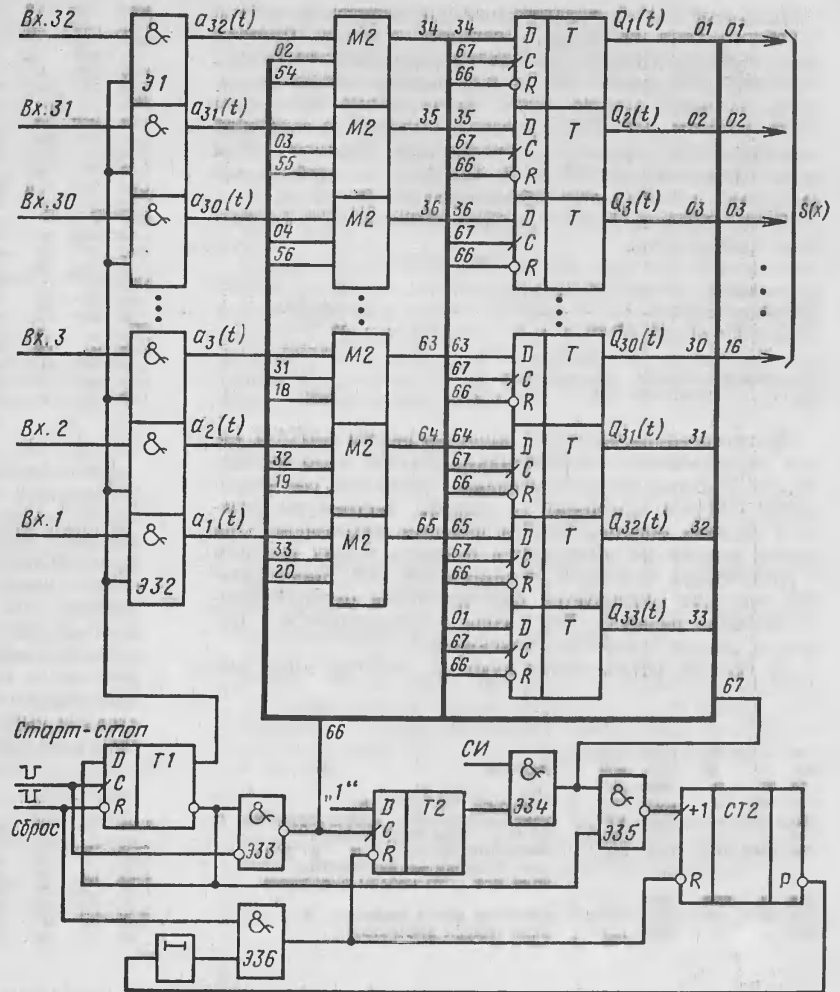
ционального диагностирования микро-
процессорных систем.

115409, Москва, Каширское шоссе, 31,
МИФИ, кафедра 12; тел. 324-34-35

ЛИТЕРАТУРА

1. Дисман А. М., Иванов А. А., Новик Г. Х. Вопросы построения параллельных структур сигнатурных анализаторов и генераторов псевдослучайных чисел // Вопросы радиоэлектроники. Сер. ОВР.— 1984.— № 11.— С. 106—117.
2. Многоканальный сигнатурный анализатор для проверки многовыходных цифровых устройств / В. И. Заславский, А. С. Календарев, А. П. Смирнов, Л. А. Шумилов // Вопросы радиоэлектроники. Сер. ТПО.— 1982.— № 1.— С. 114—119.
3. Ярмолик В. Н., Демиденко С. Н. Генерирование и применение генераторов псевдослучайных сигналов в системах испытаний и контроля / Под ред. П. М. Чеголина.— М.: Наука и техника, 1986.
4. Иванов М. А. Многоканальные анализаторы сигнатур.— Автоматика и вычислительная техника.— 1989.— № 2.— С. 84—92.

Статья поступила 25.04.88



УДК 621.391.037

В. Г. Артюхов, А. А. Ланнэ, В. И. Пенской, Е. В. Стацюра ИЗМЕРИТЕЛЬ ПЕРИОДА ОСНОВНОГО ТОНА РЕЧЕВОГО СИГНАЛА НА ОСНОВЕ ЦИФРОВОГО ПРОЦЕССОРА КМ1813ВЕ1

Измеритель периода основного тона (ОТ) является одной из составных частей вокодера, качество реализации которой во многом определяет технический уровень вокодера в целом.

Измеритель должен обеспечивать точность измерения около 1% в диапазоне частот 60...240 Гц [1]. Для этого в указанном диапазоне достаточно восьми двоичных разрядов.

Период ОТ измеряется целым числом периодов дискретизации. Наибольшему периоду, равному 1/60 Гц, целесообразно поставить в соответствие максимальное число, представляемое байтом, т. е. 255. Частота дискретизации при этом равна 15,36 кГц, а точность измерения — 0,4...1,5%. Так как программа в КМ1813ВЕ1 линейная, то частота дискретизации определяется ее длиной. При максимальной частоте синхронизации БИС 6,67 МГц время выполнения команды составляет 600 нс, поэтому длина программы должна быть равна 108 командам, а требуемая частота синхронизации БИС — 6,635 МГц.

Среди известных алгоритмов измерения периода основного тона широкое распространение получил алгоритм Рабинера и Гоулда, обладающий устойчивостью, хорошо работающий в шумах [1, 2]. Алгоритм основан на анализе экстремумов речевого сигнала во временной области. Отфильтрованный речевой сигнал анализируется шестью следящими обнаружителями основного тона, измеряющими временные интервалы между различными комбинациями минимумов и максимумов. Окончательное решение о периоде основного тона принимается по мажоритарному принципу (рис. 1).

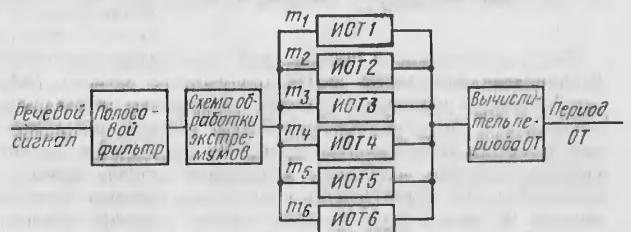


Рис. 1. Алгоритм оценки периода основного тона

Полосовой фильтр предназначен для выделения сигнала основного тона из полного речевого сигнала. Частоты среза выбираются исходя из максимальной частоты третьей гармоники ОТ, равной 720 Гц, и частотных свойств микрофона, который обычно имеет значительные искажения ниже частоты 150 Гц. Крутизна амплитудно-частотной характеристики фильтра определяется соотношениями между гармониками ОТ (-6.48дБ) и частотой первой форманты и выбирается обычно равной 24 дБ/окт.

Таким образом, фильтр должен удовлетворять следующим требованиям:

Динамический диапазон, дБ, не менее.	48
Полоса пропускания, Гц.	200...800
Подавление, дБ, не менее.	40 на частотах 65...1050 Гц
Неравномерность АЧХ, дБ, не более.	3
Частота дискретизации, кГц.	15,36

Проектирование фильтра с помощью пакета САПАМ [3] при использовании аппроксимации Кауэра дало передаточную функцию восьмого порядка. Программа, сгенерированная САПАМ, содержит 78 команд. Экстремумы речевого сигнала обрабатываются пиковым детектором. При поиске максимума вычисляется разность между текущим и предыдущим входными отсчетами XN—XP. Максимальное значение фиксируется при появлении отрицательно значения разности. Эта разность инвертируется и минимум ищется аналогично максимуму.

Программа, реализующая пиковый детектор, имеет вид

```
LDA R XN R00 NOP ;
SUB R XP R00 NOP ; R=XN-XP
LDA W R R00 NOP ;
SUB W W L01 NOP ; ИНВЕРСИЯ РАЗНОСТИ R
LDA DAR F R00 NOP ; ЗАНЕСЕНИЕ В DAR ФЛАГА МИНИМУМА F
LDA R W R00 CNDS ; R=-R, ЕСЛИ ИЩЕМ МИНИМУМ
LDA DAR R R00 NOP ; ЗАНЕСЕНИЕ РАЗНОСТИ R В DAR
LDA W K0 R00 NOP ; ЕСЛИ R<0, ТО НАЙДЕН ЭКСТРЕМУМ
LDA W KP0 R00 NOP ;
LDA W KM0 R00 CNDS ; ИНВЕРСИЯ ФЛАГА МИНИМУМА F,
XOR F W R00 NOP ; ЕСЛИ НАЙДЕН ЭКСТРЕМУМ
```

Для каждого экстремума формируются параметры M₁, M₂, M₃.

```
LDA M1 XP R00 CNDS ;
ABS M1 M1 R00 NOP ; M1=ABS(XP), ЕСЛИ НАЙДЕН ЭКСТРЕМУМ
LDA XP XN R00 NOP ; XP=XN, СМЕНА ПРЕДЫДУЩЕГО ЗНАЧЕНИЯ
LDA M2 M1 R01 CNDS ;
ADD M2 MPO R01 CNDS ; M2=(M2+MPO)/2, ЕСЛИ НАЙДЕН ЭКСТРЕМУМ
LDA W M1 R01 NOP ; MPO- ПРОШЛЫЙ ОБРАТНЫЙ ЭКСТРЕМУМ
SUB W MP R01 NOP ; M3=(M1-MP)/2, ЕСЛИ НАЙДЕН ЭКСТРЕМУМ
LDA M3 W R00 CNDS ; MP- ПРОШЛЫЙ ЭКСТРЕМУМ ТОГО ЖЕ ТИПА
LDA MP MPO R00 CNDS ;
LDA MPO M1 R00 CNDS ; MP=MPO, MPO=M1, ЕСЛИ НАЙДЕН ЭКСТРЕМУМ
ADD KK KP1 R05 NOP ; KK=KK+1, ПРИРАЩЕНИЕ СЧЕТЧИКА КОРРЕКЦИИ
LDA KK KP0 R00 CNDS ; KK=0, СБРОС СЧЕТЧИКА КОРРЕКЦИИ, ЕСЛИ НАЙДЕН
AND DAR F R00 NOP ; ЭКСТРЕМУМ
LDA M4 M1 R00 CNDS ;
LDA M5 M2 R00 CNDS ;
LDA M6 M3 R00 CNDS ;
LDA M1 KP0 R00 CNDS ;
LDA M2 KP0 R00 CNDS ; ПЕРЕЗАПИСЬ M1, M2, M3 В M4, M5, M6 И
LDA M3 KP0 R00 CNDS ; ОБНУЛЕНИЕ M1, M2, M3, ЕСЛИ ИЩЕТСЯ МИНИМУМ
```

Шесть элементарных измерителей периода основного тона представляют собой шесть однотипных детекторов с регулируемым значением времени блокировки и постоянной разряда. Для уменьшения числа команд, реализующих детектор, целесообразно вычислять напряжение в цепи разряда каждого детектора в каждом шестом проходе программы. Это эквивалентно увеличению периода дискретизации в шесть раз. В этом случае фильтр нижних частот можно реализовать одной командой SUBY(n)y(n) R05 NOP. Программа обработки сигнала элементарным измерителем имеет вид

```
LDA TB T0 R02 NOP ;
ADD TB T0 R03 NOP ; TB- ВРЕМЯ БЛОКИРОВКИ
ADD TB T0 R05 NOP ; TB=0.4*TO
LDA DAR TB R00 NOP ; TB=0, ЕСЛИ TB<0
SUB DAR KP7 R04 NOP ;
LDA TB KP7 R04 CNDS ; TB =14, ЕСЛИ TB<14
SUB TB K0 R00 NOP ; TB=TB-K(I)
LDA DAR TB R00 NOP ;
LDA W P0 R00 NOP ;
SUB W W R05 NOP ; PP(I)=P(I)-P(I)/32 - ЭКСПОНЕНЦИАЛЬНЫЙ СПАД
LDA P0 W R00 CNDS ; ЗАРЯД ДЕКТОРА, ЕСЛИ ВРЕМЯ БЛОКИРОВКИ
LDA M0 KP0 R00 NOP ; ИСТЕКЛО
LDA W V R00 NOP ;
SUB W W L01 NOP ; ИНВЕРСИЯ M(I)
LDA DAR W R00 NOP ;
ADD TB KK R00 CNDS ; TB=TB+KK, ЕСЛИ ПРИШЕЛ ЭКСТРЕМУМ (M(I)<>0)
LDA DAR TB R00 NOP ;
LDA M0 V R00 CNDS ; M0=M(I), ЕСЛИ ВРЕМЯ БЛОКИРОВКИ ИСТЕКЛО
LDA DAR P0 R00 NOP ;
SUB DAR M0 R00 NOP ; P(I)=P(I)-M0
LDA W K0 R00 NOP ;
SUB W KK R00 NOP ;
LDA T0 W R00 CNDS ; T(I)=K(I)-KK, ЕСЛИ M0 ПРЕВЫШАЕТ P0
LDA K0 KK R00 CNDS ; K(I)=KK,
LDA P0 V R00 CNDS ; P(I)=M(I),
```

Время между приходом экстремума и его обработкой учитывается счетчиком коррекции (СК), который сбрасывается при каждом появлении экстремума.

Так как время блокировки лежит в пределах 1,56...4,16 мс, а время обработки одного отсчета равно 65 мкс, то можно изменить шесть параллельных следящих измерителей основного тона одним с последовательной обработкой входных сигналов. Для этого однотипные параметры собраны в массивы, которые на каждом проходе программы циклически сдвигаются. При этом содержимое ячеек, используемых в детекторе, последовательно подается на его входы и выходы. Пример программы такого циклического сдвига массива оценок периода основного тона имеет вид

```
LDA W T0 R00 ; ИКЛИЧЕСКИЙ СДВИГ МАССИВА ОЦЕНОК
LDA T0 T1 R00 ; ПЕРИОДА ОСНОВНОГО ТОНА T(I)
LDA T1 T2 R00
LDA T2 T3 R00
LDA T3 T4 R00
LDA T4 T5 R00
LDA T5 W R00
LDA W P0 R00
```

Программа выделения оценок периода основного тона занимает 104 команды.

Блок статистической обработки вычисляет период основного тона путем сравнения каждого элемента строки с шестью элементами первого столбца [2].

Программа, реализующая алгоритм сравнения с элементами первой строки, имеет вид

```
LDA DAR T11 R00 NOP ; DAR1=T11
SUB DAR T R00 NOP ; DAR1=T11-T
ABS DAR DAR R00 NOP ; DAR1=ABS(T11-T)
SUB DAR KP1 R02 NOP ; DAR1=ABS(T11-T)-8
ADD 8 KP1 R05 CNDS ; 8=8+1, ЕСЛИ DAR1<0
; ПОДСЧЕТ ЧИСЛА СОВПАДЕНИЙ
```

После обработки все строки матрицы циклически сдвигаются на одну позицию. При этом обрабатываются элементы следующего столбца. Пример реализации такого сдвига для первой и второй строк приведен ниже.

```
LDA T11 W R00 CNDS ; ЦИКЛИЧЕСКИЙ СДВИГ ПЕРВОЙ СТРОКИ,
LDA W T26 R00 CNDS ; ЕСЛИ K<6
LDA T26 T25 R00 CNDS ;
LDA T25 T24 R00 CNDS ;
LDA T24 T23 R00 CNDS ;
LDA T23 T22 R00 CNDS ;
LDA T22 T21 R00 CNDS ;
LDA T21 W R00 CNDS ; ЦИКЛИЧЕСКИЙ СДВИГ ВТОРОЙ СТРОКИ,
LDA W T36 R00 CNDS ; ЕСЛИ K<6
LDA T36 T35 R00 CNDS ;
LDA T35 T34 R00 CNDS ;
LDA T34 T33 R00 CNDS ;
LDA T33 T32 R00 CNDS ;
LDA T32 T31 R00 CNDS ;
```

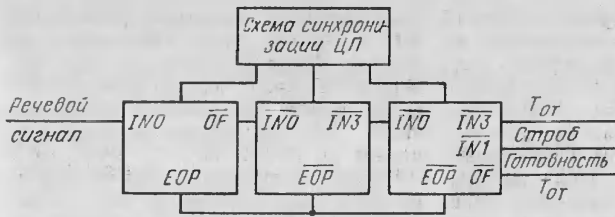


Рис. 2. Структурная схема измерителя основного тона

Сравнение со всеми 36 элементами матрицы происходит за шесть проходов программы, затем выбирается следующая оценка и обработка повторяется. После 36 проходов программы (2,3 мс) формируется новая матрица оценок (третья строка принимает значение второй, вторая — первой, а первая — значения из массива приема) и выводится окончательный период основного тона. Программа статистической обработки занимает 107 команд.

Общая длина программы измерителя периода основного тона составляет 289 команд, и требуется использование трех БИС КМ1813ВЕ1 (рис. 2). Исходя из анализа алгоритма программу целесообразно распределить следующим образом: БИС1 — полосовой фильтр (78 команд); БИС2 — формирователь оценок периода основного тона (104 команды); БИС3 — статистическая обработка оценок периода основного тона (107 команд).

Первая БИС принимает речевой сигнал в аналоговой форме. С последующими БИС она обменивается цифровыми кодами. При этом цифровой вывод БИС1 возможен только по выходу переполнения OF. Программа аналогового ввода совмещается с цифровыми полями полосового цифрового фильтра, а программа вывода по OF занимает 29 команд. Таким образом, длина программы полосового фильтра составляет 108 команд.

Программы цифрового ввода-вывода второй и третьей БИС также совмещаются с цифровыми полями команд ЦП.

УДК 681.3.06.02

В. Ф. Зельниш, Л. П. Лобанов, И. В. Пивоваров, В. А. Терсков, Г. С. Тимофеев

РАЗРЕШЕНИЕ КОНФЛИКТОВ ПРИ ОБРАЩЕНИИ МИКРОПРОЦЕССОРОВ К ОБЩЕЙ ПАМЯТИ МУЛЬТИМИКРОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Возникновение конфликтов при одновременном обращении нескольких микропроцессоров (МП) к общим неделимым ресурсам, особенно к общей памяти (ОП), снижает производительность мультимикропроцессорных вычислительных систем (ММПВС).

При использовании многоходовой памяти потери производительности уменьшаются, так как реализуется возможность одновременной выборки нескольких операндов. Однако возникают ситуации, когда один МП считывает из ячейки, а другой в то же время записывает в нее (из ОП считывается искаженное слово). Для разрешения такого типа конфликтов предлагается организовать процесс обращения к памяти следующим образом: цикл чтения будет частью цикла записи, который по времени опережает чтение [1]. В этом случае всегда будет считана «новейшая» информация, но быстроедействие памяти снизится, так как

цикл обращения увеличится в два раза. Операции записи составляют незначительную часть от общего числа обращений [2], поэтому увеличивать каждый цикл обращения к памяти нецелесообразно.

Необходимо обнаружить момент возникновения конфликта и только в это время произвести повторное считывание.

Запоминающее устройство с одновременной выборкой нескольких слов, определяющее момент возникновения конфликтов и их разрешение, состоит из матрицы запоминающих элементов, адресных блоков записи и чтения (регистров и дешифраторов); регистров направления чтения по числу направлений выборки с триггерами сбоя и элементами И [3]. Информация выбирается независимо по каждому направлению, запись ведется только по одному направлению. Считанная информация передается в соответствующий регистр направления чтения. Если в течение

Поэтому длина программ БИС2 и БИС3 составляет 108 команд, что удовлетворяет требованиям.

Разработанная программа моделировалась на кросс-системе проектирования прикладного программного обеспечения КМ1813ВЕ1 [4]. В результате моделирования правильная оценка периода основного тона получилась через 20/30 мс, что достаточно для вокодеров. Экспериментальные исследования измерителя в составе вокодера дали аналогичный результат. Конструктивно измеритель выполнен на плате размерами 150×160 мм, потребляет мощность 3,5 Вт.

Использование БИС КМ1813ВЕ1 дало выигрыш в аппаратных затратах (в сравнении с аналогичными решениями в виде цифрового автомата) при увеличении потребляемой мощности. Основным преимуществом предложенной реализации является возможность гибкого изменения используемых алгоритмов обработки речевого сигнала путем перепрограммирования БИС.

Телефон 441-14-53, Киев

ЛИТЕРАТУРА

1. Сапожников М. А., Михайлов В. Г. Вокодерная связь.— М.: Радио и связь, 1983.
2. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов.— М.: Мир, 1978.
3. Артюхов В. Г., Бритов А. А., Кондратюк В. А. Пакет прикладных программ автоматизированного проектирования цифровых фильтров на СМ ЭВМ. // Автоматизация проектирования в электронике: Респ. Межвед. науч.-техн. сб.— Киев: Техніка, 1985.— Вып. 32.— с. 73—79.
4. Артюхов В. Г., Макеенок А. Н. Автоматизация проектирования программного обеспечения цифровых процессоров сигналов как средство повышения эффективности инженерного труда.— Киев: Знание, 1985.

Статья поступила 31.03.88

выборки из ячейки в нее записывается новая информация, то возникает конфликт, фиксируемый триггером сбоя соответствующего направления, после этого считывание повторяется.

Определим производительность (П) ММПВС с учетом $P_{k,1}$ — вероятности нахождения системы массового обслуживания (СМО) в определенном состоянии [4, с. 74—75] при использовании S-входовой памяти, равной

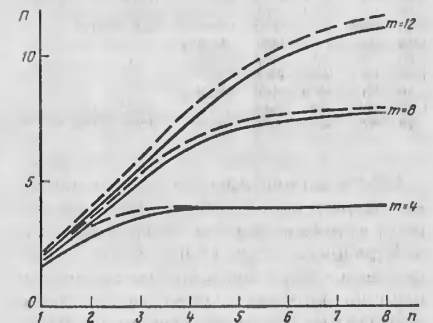


Рис. 1. Зависимость производительности ММПВС от числа МП и четырех входных блоков ОП

сумме вероятностей соответствующего состояния СМО при записи ($P_{k,1}^{3n}$ [4]) и чтении

$$P_{k,1}^{3n} = \begin{cases} P_{k,1} & \text{при } K > S \vee I = 0 \\ 0 & \text{в остальных случаях.} \end{cases}$$

Коэффициенты учитывают вероятность обращения МП в ОП для записи или чтения соответственно.

Максимальное число одновременно обслуживаемых в ОП запросов

$$h = \begin{cases} m & \text{при } n \geq m \wedge n \geq s \\ n & \text{при } n < m \wedge n \geq s \\ s & \text{при } n < m \wedge n < s \end{cases}$$

На рис. 1 приведены зависимости производительности ММПВС от числа МП (m), блоков ОП (n) и входов (S) при разрешении конфликтов (штриховая линия — предлагаемый метод непрерывная — метод, изложенный в [1].

Приоритетное разрешение конфликтов. В ММПВС наиболее часто используется жесткое закрепление приоритета за каждым МП, однако при высокой интенсивности поступления запросов от «старшего» МП «младший» долго не обслуживается. Неравномерность в работе МП приводит к снижению производительности, а в некоторых случаях к остановке вычислительного процесса. Предлагается дисциплина обслуживания «реже обращается — раньше обслужен» (РОРО), при которой предпочтение отдается тому МП, который реже обращается в блок ОП (устраняется недостаток жесткого закрепления приоритетов, а частота обслуживания каждого МП усредняется). Для формирования кода приоритета каждого МП необходимо накапливать число обращений каждого МП в соответствующем блоке ОП, хранить его во время выбора кода приоритета или ожидания обслуживания и выделять номер МП (из числа обратившихся), число обращений которого на момент формирования кода приоритета минимально.

Устройство приоритетного разрешения конфликтов содержит узлы формирования кодов приоритетов (УФП), состоящие из триггера фиксации запроса от МП, счетчика для подсчета поступивших запросов, регистра номера МП и узла выделения приоритетов — УВП (матрица из блоков опроса (БО), анализа разрядов кода приоритета и элементов ИЛИ, равных числу дополнительных разрядов кода приоритета, соответствующих номеру МП) [5].

Работа устройства. Сигналы запросов от каждого МП приходят на входы триггера и счетчика УФП, где формируются коды приоритетов, поступающие на соответствующие БО. На каждом шаге просматривается содержимое соответствующих разрядов кодов приоритетов. Если одноименные разряды всех кодов равны либо нулю, либо единице, то коды передаются в блок анализа для сравнения в следующем шаге; если значения разрядов разные, то коды, разряды которых равны нулю, дальше не рассматриваются. Под-

множество кодов, разряды которых равны единице, просматриваются, и на последнем шаге выделяется код номера процессора. Содержимое младших разрядов, выдаваемых с выходов элементов ИЛИ, указывает номер старшего МП, которому разрешено обращение в блок ОП. При равных значениях нескольких счетчиков УФП (несколько МП обратились одинаковое число раз) на выходе устройства устанавливается код МП с наименьшим номером. Триггер УФП возвращается в исходное состояние при получении кода номера старшего МП. Коды на каждом шаге сравниваются одновременно по разрядно, поэтому время выбора номера старшего МП зависит только от их разрядности.

Для ММПВС с разделенной памятью команд и данных методы разрешения конфликтов основаны на сдвиге начала циклов работы МП или блоков ОП [6]. Начало цикла каждого блока ОП сдвигается на время, необходимое для считывания и передачи информации в общих цепях связи. Недостаток метода — время ожидания необходимой информации, зависит от номера блока ОП, к которому обратился МП.

Этот недостаток устраняется при сдвиге циклов начала работы МП. Процесс выполнения команды можно разделить на несколько основных временных интервалов (циклов). Каждый МП обращается к ОП в разное время. Например, выполнение команды в МП типа К580 (i8080) можно разбить на пять машинных циклов [7, 8].

Число одновременных обращений к блоку ОП уменьшается при сдвиге начала работы МП на один-два машинных цикла. МП запускается либо программно командой HALT, либо аппаратно с помощью сигналов SYNG, READY, WAIT, HOLD.

Производительность ММПВС зависит от числа МП и блоков ОП. При

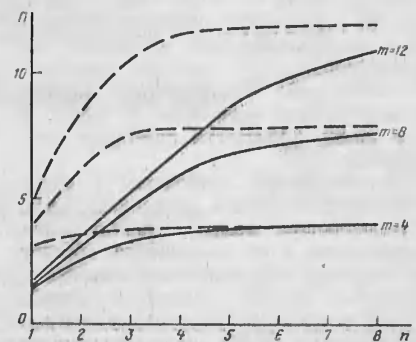


Рис. 2. Рассчитанная зависимость производительности ММПВС от числа МП и блоков ОП:

штриховые линии — предлагаемый метод, непрерывные линии — методы разрешения конфликтов не используются

одновременном обращении нескольких МП к одному блоку ОП возникает очередь. Число одновременно обратившихся МП равно ближайшему большему целому от деления реального числа МП на число независимых циклов [4, (10)].

Число циклов выполнения команд и их продолжительность не постоянны, а число МП может превышать число циклов. Рассчитанная производительность показана на рис. 2.

Телефон 472-94-22, Москва, Тимофеев Геннадий Сергеевич

ЛИТЕРАТУРА

1. Chang S. S. L. Multiple-read single Write memory and applications // IEEE Transactions Computers.— 1980.— Vol. 29, № 2.— P. 689—694. (Экспресс-информация. Сер. Вычислительная техника.— 1981.— № 17.— Ref. № 89. С. 1—4.)
2. Заболотный А. А., Недзельский Д. А. Анализ алгоритмов связи с главной памятью в мультимикропроцессорных системах // Приборы и системы управления.— 1976.— № 4.— С. 17—18.
3. А. С. 970464 СССР, МКИ⁴ G 11 C 11 / 00. Запоминающее устройство с одновременной выборкой нескольких слов. / В. Ф. Зельтинш, Л. П. Лобанов, В. И. Горбенко, Г. С. Тимофеев.— Опул. 1982. Бюл. № 40.
4. Зельтинш В. Ф., Лобанов Л. П., Терсков В. А., Тимофеев Г. С. Оценка производительности микропроцессорных вычислительных систем при конфликтах в общей памяти блочной структуры // Автоматика и вычислительная техника.— 1986.— № 2.— С. 74—80.
5. А. С. 1024921 СССР, МКИ⁴ G06F 9 / 46. Устройство для выбора по приоритету. / В. И. Горбенко, В. Ф. Зельтинш, Л. П. Лобанов, Г. С. Тимофеев.— Опул. 1983. Бюл. № 23.
6. Мультимикропроцессорные вычислительные системы. / Под ред. Я. А. Хетагурова.— М.: Энергия, 1980.
7. Микропроцессорные БИС и микроЭВМ. Построение и применение / А. А. Васенков, Н. М. Воробьев, В. Л. Духуян и др.; Под ред. А. А. Васенкова.— М.: Сов. радио, 1980.
8. Макглин Д. Р. Микропроцессоры. Технология, архитектура и применение / Пер. с англ. под ред. И. В. Прангишвили.— М.: Энергия, 1979.

Статья поступила 10.05.88

ФУНКЦИОНАЛЬНЫЕ МОДУЛИ КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Функциональные модули, входящие в состав разрабатываемой на базе микроЭВМ «Электроника 60» контрольно-измерительной и испытательной системы (КИС), обеспечивают выполнение следующих функций: генерацию управляющих (входных) воздействий, регистрацию ответных (выходных) реакций, временное согласование работы отдельных частей КИС, аналого-цифровое и цифро-аналоговое преобразование, электрическое и логическое сопряжение с объектом контроля, отладки и измерения, а также улучшение характеристик основных модулей (например, повышение быстродействия).

Комплект включает в себя модули: логического анализатора (МЛА), регистрирующего 1024 состояния 20 входных линий с тактовой частотой до 10 МГц [1]; адаптера [2], позволяющего поднять быстродействие МЛА; синхронизации по назначенным комбинациям уровней на линиях; буферные модули, модуль АЦП для регистрации совместно с МЛА аналоговых сигналов, генерации кодовых последовательностей и ряд других. Ниже подробно описывается организация перечисленных модулей.

Требования, предъявляемые к ним: специализация на выполнении одной или нескольких взаимосвязанных четко выделенных функций;

программное управление всеми режимами работы;

автономная работа одного модуля и совместное включение нескольких.

Модуль синхронизации по назначенным комбинациям (МСНК) уровней на линиях упрощает процесс контроля и отладки цифровых устройств, так как в МЛА не предусмотрена синхронизация процесса регистрации по назначенным комбинациям уровней на входных линиях [3]. С помощью МСНК запускаются процессы регистрации по комбинациям уровней на входных линиях МЛА (состояние которых регистрируется) или дополнительных линиях синхронизации, а также другие модули КИС, в частности генератор кодовых последовательностей. Для увеличения разрядности запускающего слова отдельные МСНК включаются параллельно; разряды, состояние которых не имеет значения, маскируются.

Состояния линий синхронизации и назначенной синхροкомбинации (запускающего слова) сравниваются компараторами кодов КК1—КК5, выходные сигналы которых объединены по схеме И (рис. 1). МСНК включаются параллельно через входы расширения — свободные входы элемента DD1. Сигнал совпадения кодов на входах компараторов записывается в триггер

DD4 по стробующему импульсу (сигналу на одной из входных линий МЛА или сигналу его тактового генератора). Его полярность задается внешним сигналом ПЛР. Например, при синхронизации по обращению к заданному адресу в канале микроЭВМ «Электроника 60» в качестве стробующего импульса (СИ) используется проинвертированный сигнал КСИАН (А0...А15 — линии синхронизации).

Линии синхронизации, не входящие в синхροкомбинацию, маскируются в схеме пропускания (СП), представляющей собой элементы 2И, на один вход которых поступает сигнал синхронизации, а на другой — разряд кода маски. Нулевой уровень выставляется в соответствующем разряде кода маски и

в этом же разряде запускающего слова. Код маски, запускающее слово и сигнал ПЛР подаются на МСНК со стандартной схемы интерфейса. В МСНК использованы микросхемы: КР531СП1 (КК1...КК5), КР531ЛИ1 (СП), КР531ТМ2 (DD4) и др.

Модуль генерации кодовых последовательностей (МГКП) задает входные управляющие воздействия в режиме реального времени (рис. 2). В роли МГКП может выступать процессор системы, выполняющий программу обмена с отлаживаемым модулем. Однако в этом случае по каналу микроЭВМ производятся обращения, не связанные непосредственно с процессом отладки (выборка кодов команд из системного ОЗУ, регенерация системного ОЗУ). Возможны неисправности отлаживаемого модуля, приводящие к сбоям в работе процессора. Требования, предъявляемые к автономному МГКП: высокое быстродействие, не уступающее быстродействию МЛА; использование внутреннего и внешнего тактовых сиг-

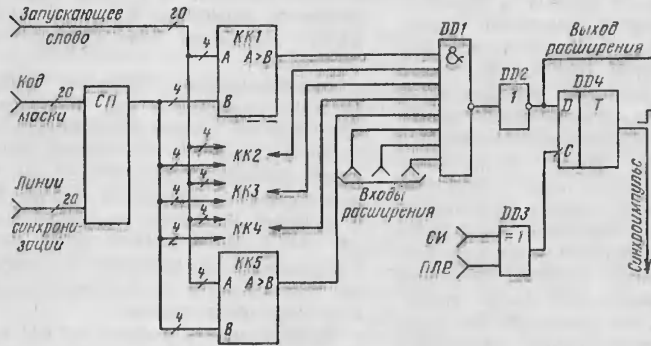


Рис. 1. Функциональная схема МСНК:

СП — схема пропускания, КК1...КК5 — компараторы кодов, СИ — стробующий импульс, ПЛР — сигнал управления полярностью СИ

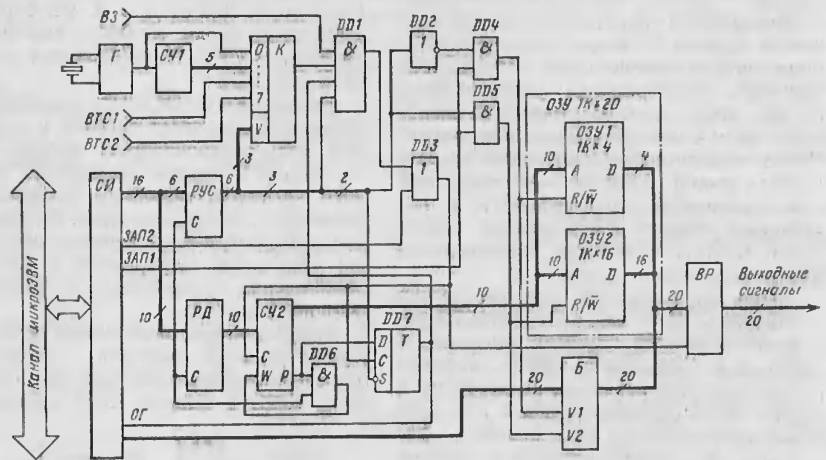


Рис. 2. Функциональная схема МГКП:

СИ — схема интерфейса, Г — кварцевый генератор, СЧ1 и СЧ2 — двойные счетчики, К — коммутатор, РДС — регистр управляющего слова, РД — регистр длины последовательности, В — буфер, ВР — выходной регистр, V3 — сигнал внешнего закрета, VТС1 и VТС2 — внешние тактовые сигналы, ЗАП1 и ЗАП2 — сигналы записи, ОГ — сигнал окончания генерации.

налов; большое число каналов и возможность их наращивания; программное задание числа тактов генерируемой последовательности; программный выбор режима запуска (однократный или периодический); достаточно большой объем внутренней памяти состояний выходных линий.

С помощью МГКП генерируются последовательности 20-разрядных кодов длительностью до 1024 периодов тактового сигнала: внутреннего с периодами 100, 200, 400, 800, 1600, 3200 нс или любого из двух внешних (ВТС1 и ВТС2) с частотой до 10 МГц. Возможна генерация последовательностей кодов при разовом или периодическом запуске. Предусмотрен сигнал внешнего запрета (ВЗ), останавливающий генерацию кодовых последовательностей и организующий параллельное или последовательное соединение нескольких МГКП.

Перед началом генерации заполняется внутреннее ОЗУ МГКП по сигналу ЗАП1. Разрядность ОЗУ (20) превышает разрядность шины данных микроЭВМ «Электроника 60» (16), поэтому процессы записи в ОЗУ1 и ОЗУ2 разделены во времени. Номер ОЗУ, в которое записывается информация, определяется состоянием одного из разрядов регистра управляющего слова (РУС). Каждое обращение к ОЗУ вызывает изменение состояния счетчика адресов (СЧ2). Такая организация процесса записи повышает оперативность смены информации в ОЗУ, позволяя, например, изменять последовательность состояний 16-разрядной шины адреса-данных (ОЗУ2), не меняя состояния 4-разрядной шины управления (ОЗУ1), при работе МГКЦ в качестве задатчика канала микроЭВМ «Электроника 60».

Режим работы МГКЦ задается 16-разрядным словом, записываемым в регистры модуля. При этом в 10 разрядах в регистр РД передается длина генерируемой последовательности (до 1024 тактов), а шесть разрядов образуют управляющее слово, заносимое в РУС. Три разряда управляющего слова определяют вид тактового сигнала (К); один разрешает или запрещает генерацию (DD1), один определяет режим запуска — разовый или периодический (триггер DD7), а последний задает номер памяти (ОЗУ1 или ОЗУ2), в которую производится запись.

При периодическом запуске счетчик адресов ОЗУ (СЧ2), работающий в режиме вычитания, отсчитав занесенное в РД и в СЧ2 число тактов, вырабатывает сигнал переноса, по которому в него перезаписывается код с РД, процесс повторяется. При разовом запуске по сигналу переноса триггер DD7, запрещающий дальнейшую генерацию, перебрасывается. Сигнал с его выхода выдается в схему интерфейса СИ и служит флагом окончания генерации (ОГ). Выходные коды ОЗУ записываются в выходной регистр

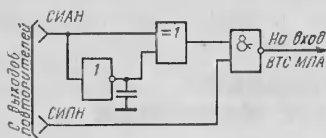


Рис. 3. Принципиальная схема формирователя тактового сигнала

(ВР) для выравнивания времен задержек изменения состояний различных линий.

МГКП конструктивно реализован на полуплате микроЭВМ «Электроника 60» и содержит 32 корпуса ИМС (ОЗУ — К541РУ2А, СЧ1, СЧ2 — КР531ИЕ17, РУС, РД, ВР — КР531ИР18, Б — К155ЛП11).

Буферный модуль (БМ) уменьшает влияние КИС на отлаживаемое устройство, снижает искажения формы передаваемых импульсов. Он содержит 20 (по числу каналов) повторителей входных сигналов КИС с малым входным током (пять микросхем К589АП16) и 20 повторителей выходных сигналов КИС (К155ЛП9). В составе КИС применяются несколько БМ для отладки разных устройств, отличающихся конструктивно.

Формирователь тактового сигнала (ФТС), входящий в БМ, фиксирует в памяти МЛА трассы выполняемой программы без учета временных соотношений между сигналами (рис. 3). При использовании выходного сигнала ФТС в качестве внешнего тактового сигнала (ВТС) МЛА в ОЗУ МЛА фиксируется адрес, по которому производится обращение, а также передаваемые и принимаемые данные.

Эффективность работы МГКП можно повысить, используя схему, приведенную на рис. 4. Она вырабатывает сигнал ВЗ для МГКП на время от выдачи сигналов К ВВОД Н или К ВЫВОД Н до поступления К СИП Н. Сигналы К ВООД Н и К ВЫВОД Н продлеваются до момента выработки сигнала К СИП Н (экономится объем ОЗУ МГКП).

Модуль АЦП (МАЦП) преобразует уровень аналогового сигнала в цифровой код, синхронизирует процесс регистрации и момент достижения входным сигналом заданного уровня, изменяемого программно (рис. 5). Наиболее распространенный режим обработки аналоговой информации в ЭВМ — нахождение экстремального значения входного сигнала.

Входной аналоговый сигнал преобразуется в коды выборки с частотой, определяемой внешним тактовым сигналом. Коды с выхода АЦП поступают

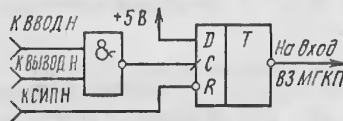


Рис. 4. Схема останова работы МГКП

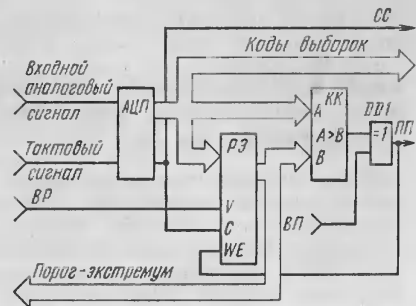


Рис. 5. Функциональная схема МАЦП: КК — компаратор кодов, РЭ — регистр экстремума, ВР — сигнал выбора режима, ВП — сигнал выбора полярности, СС — синхросигнал, ПП — сигнал превышения порога

на выход модуля, на КК, вырабатывающий сигнал превышения заданного порога, а также на регистр экстремума (РЭ), выходы которого имеют три состояния.

МАЦП может работать в двух режимах: регистрации формы аналогового сигнала с синхронизацией по превышению заданного порога и нахождения экстремума аналогового сигнала. В первом режиме выходы РЭ находятся в состоянии высокого импеданса (сигнал выбора режима). По двунаправленной шине Порог-экстремум на МАЦП со схемы интерфейса подается код порога. При превышении порога сигнал с выхода КК через элемент DD1, определяющий полярность синхрорежима входного сигнала через заданный порог (сигнал выбора полярности), поступает на синхровход МЛА. В качестве тактового сигнала МЛА используется сигнал окончания преобразования (готовности данных) с АЦП. По этому же сигналу выходной код АЦП заносится в РЭ в режиме экстремума. В этом режиме шина Порог-экстремум служит входной шиной МАЦП, по которой передается значение экстремума. Выходы РЭ переводятся в активное состояние. Запись в РЭ разрешается сигналом превышения порога (ПП), а сигнал ВП в этом случае определяет вид экстремума.

В качестве АЦП использована микросхема К1108ПВ1 [4] с временем преобразования не более 0,9 мкс (возможно применение и более быстрореагирующих микросхем АЦП, например К1107ПВ1 [5] или К1107ПВ2), в качестве РЭ — К155ИР15, КК — КР531СП1. Следует отметить, что разрядность входной шины МЛА позволяет организовать одновременную регистрацию двух или трех аналоговых сигналов.

Разработаны 28-канальная версия МЛА, 32-канальная версия МГКП и соответствующие программные средства.

С помощью комплекта специализированных функциональных модулей, включающего в себя описанные и некоторые дополнительные модули с аналоговой организацией (ЦАП, генератора с управляемой частотой, управ-

ляемого усилителя аттенюатора), можно строить достаточно дешевые гибкие системы измерения, контроля и управления, а также легко модернизировать их. Применение в составе таких систем средств отображения информации [6] позволяет организовать эффективное взаимодействие с пользователем.

115409, Москва, Каширское шоссе, 31, МИФИ, кафедра 3; тел. 324-95-65

ЛИТЕРАТУРА

1. Бородин С. М., Новиков Ю. В. Модуль логического анализатора для контрольно-измерительных систем на базе микроЭВМ // Микропроцессорные средства и системы.— 1987.— № 1.— С. 67—68.
2. Бородин С. М., Зайцев В. Л., Новиков Ю. В. Быстродействующий логический анализатор для аппаратно-программных комплексов разработки микропроцессорных систем // Микропроцессорные средства и системы.— 1987.— № 1.— С. 65.
3. Бирк Дж., Мерц М. Личные системы, ускоряющие тестирование и снижающие его стоимость // Электроника.— 1983.— № 7.— С. 27.
4. Игнатов Б. И., Рябов Е. А., Сотский Д. В. и др. Быстродействующий однокристалльный АЦП

ропроцессорные средства и системы.— 1987.— № 1.— С. 67—68.

5. Крылов Е. Т., Крылова Е. Э., Медведева Л. П. Быстродействующий аналого-цифровой преобразователь для микропроцессорных систем // Микропроцессорные средства и системы.— 1985.— № 1.—
6. Бородин С. М., Новиков Ю. В., Поддубный А. П., Томчук А. А. Средства отображения информации для микропроцессорных систем измерения, контроля и управления // Микропроцессорные средства и системы.— 1988.— № 3.— С. 76—79.

Статья поступила 13.11.87

УДК 681.326.3

А. А. Бондаренко, В. Ф. Скороходов

УСТРОЙСТВО КОНТРОЛЯ УРОВНЯ ВЫХОДНЫХ СИГНАЛОВ МИКРОСХЕМ ПАМЯТИ

Широко используемые в микропроцессорной технике РПЗУ с УФ-стиранием имеют следующий недостаток: уровень выходного сигнала данных с выхода микросхемы иногда оказывается между Лог.0 и Лог.1, т. е. между 0,4 и 2,4 В. Для объяснения этого явления необходимо обратиться к структуре запоминающего элемента данного типа микросхем [1]. Запоминающий элемент, выполненный на основе МОП-структур с плавающим (изолированным) затвором с использованием эффекта лавинной инжекции (ЛИЗМОП), представляет собой МОП-транзистор с индуцированным каналом, плавающий затвор кото-

рого изолирован от кристалла слоем диэлектрика (диоксида кремния).

При программировании микросхемы (запись Лог.0) на плавающем затворе в результате инжекции носителей через диоксид кремния накапливается некоторый заряд. При стирании микросхемы происходит разряд плавающего затвора за счет ионизации диоксида кремния. Таким образом, если плавающий затвор не полностью разрядился во время стирания микросхемы или накопленный заряд при программировании меньше необходимого, то вполне вероятно, что значение выходного напряжения на линии данных РПЗУ окажется между 0,4 и 2,4 В. Такой же эффект может произойти и при длительном хранении запрограммированной микросхемы из-за утечки заряда плавающего затвора через диэлектрик.

В известных программах микросхем РПЗУ [2—4] не предусмотрена возможность контроля уровня выходного сигнала, что приводит к труднообнаруживаемым ошибкам, так как выходной сигнал микросхем РПЗУ в программаторе может интерпретироваться как Лог.0, а в реальном устройстве — как Лог.1 или наоборот. Это объясняется различным уровнем порога считывающих элементов, различными нагрузками и т. д.

Устройство контроля уровня выходных сигналов микросхем РПЗУ К573РФ2 и К573РФ5 предназначено для работы с программатором, рассмотренным в [4], однако может использоваться с любым другим программатором (рис. 1). В рабочем режиме устройству необходимы три линии параллельного вывода и две линии ввода. В программаторе [4] в качестве линий вывода используется порт В2, ввода — младшая половина порта С2. Устройство состоит из аналогового мультиплексора 8→1 DD2 и двух компараторов напряжения DD3 и DD4. С помощью подстроечных резисторов R1 и R2 на входе одного из них задается уровень Лог.0—0,4 В, на входе другого — Лог.1—2,4 В.

Работает устройство следующим образом. На адресные входы подаются адрес проверяемого слова и необходимые

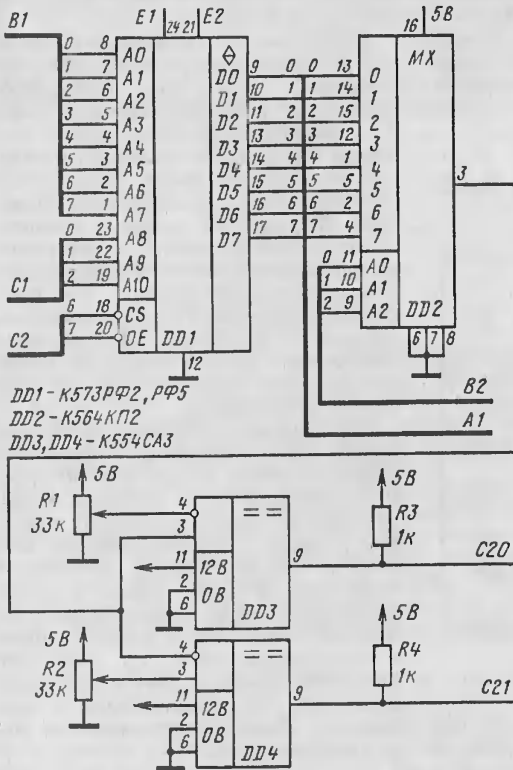


Рис. 1. Принципиальная схема устройства контроля микросхем РПЗУ

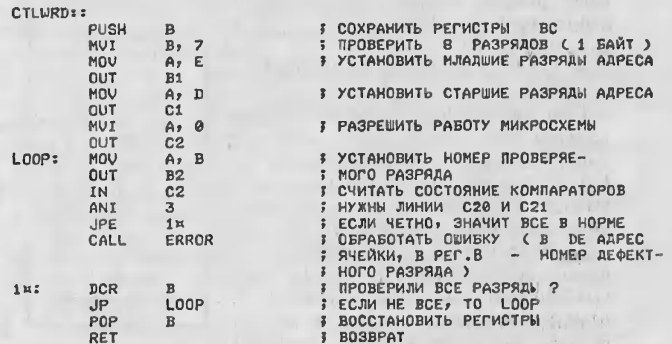


Рис. 2. Программа проверки одного слова РПЗУ: DE — адрес проверяемой ячейки

сигналы для режима чтения микросхемы. На линиях В20...В22 устанавливается номер проверяемого разряда, считывается состояние линий С20 и С21. Наличие на этих линиях сигналов Лог.0 означает, что в данном разряде проверяемого слова записан Лог.0. Если на линиях С20 и С21 будет Лог.1, то в данном разряде — Лог.1. Значение проверяемого разряда считается неопределенным (уровень выходного напряжения между 0,4 и 2,4 В), когда на одной линии установлен сигнал Лог.1, а на другой — Лог.0. Программа проверки одного слова микросхемы РПЗУ приведена на рис. 2.

454044, Челябинск, пр. Ленина, 76. Челябинский политехнический ин-т, каф. радиотехнических систем; тел. 39-92-87.

УДК 681.3.06

А. Н. Бычков, Р. В. Галанский, А. М. Никифоров, Р. Р. Аухадеев

ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ДЛЯ ПРОЕКТИРОВАНИЯ МП-СИСТЕМ

В инструментальный комплекс для проектирования МП-систем на базе БИС серии К580 входят микроЭВМ СМ1800 и универсальный программируемый контроллер «Электроника МС2702», выполняющий функции управления преобразователями информации, схемами регулирования и вычисления разрабатываемого устройства или системы (рис. 1).

Состав контроллера «Электроника МС2702»: канал последовательного связи, контроллер прерываний, трехканальный таймер, каналы параллельного обмена. Направление обмена информацией по параллельному каналу задается переключками в коммутационном поле платы контроллера в зависимости от конкретного типа подключаемого устройства и протокола его обмена. Жесткая структура ограничивает гибкость использования контроллера. Для устранения этого недостатка контроллер был доработан: подрезаны шины печатного монтажа и установлены новые переключки (рис. 2).

Использование одного порта (FН2) микросхемы параллельного ввода-вывода D59 для управления шинными формирователями D61...D68, D70, D71 позволило сделать каналы параллельного обмена двунаправленными. Шинные формирователи переходят в режим ввода или вывода при подаче в порт F2H микросхемы D59 соответствующих чисел FFH и F2H. Они объединены в группы D61, D62, D65, D66; D70, D71; D63, D64; D67, D68, поэтому можно изменять направление передачи для каждой из групп. В этой схеме контроллера направление передачи ка-

нала параллельного обмена и состояние выходных шин задаются программно. Для гальванической развязки микроЭВМ СМ1800 и контроллера «Электроника МС2702» по каналу последовательного обмена установлены адаптеры. Адаптер последовательного обмена — плата размерами 135×35 мм, размещенная в кожухе ответного (кабельного) разъема ХС2 МС2702; второй адаптер — плата размерами 50×50 мм, которая крепится в произвольном месте к СМ1800.

ПО комплекса включает в себя программу дистанционного отладчика, функционирующего под управлением ОС1800 (размещается в оперативной памяти СМ1800 с адреса D000H), и драйвер связи (200 байт). Драйвер может быть прошит в ПЗУ контроллера с адреса 5000H и запускаться либо директивой С5000, либо с нулевого

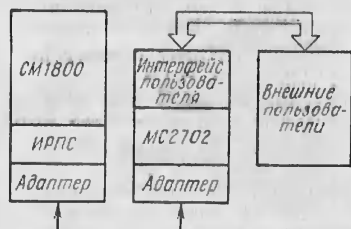


Рис. 1. Структурная схема комплекса

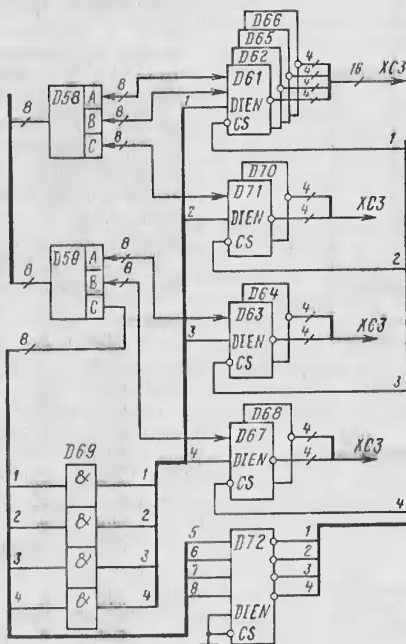


Рис. 2. Схема доработки контроллера «Электроника МС2702»

ЛИТЕРАТУРА

1. Косарев Ю. А., Виноградов С. В. Электрически изменяемые ПЗУ.— Л.: Энергоатомиздат, 1985.
2. Дианов А. П., Щелкунов Н. Н. Модуль программирования микросхем ПЗУ// Микропроцессорные средства и системы.— 1985.— № 3.— С. 80—83.
3. Лукьянов Д. А. Схемотехника универсальных программаторов ПЗУ// Микропроцессорные средства и системы.— 1985.— № 3.— С. 84—88.
4. Дианов А. П., Щелкунов Н. Н. Модули программирования логических схем// Микропроцессорные средства и системы.— 1988.— № 1.— С. 40—44.

Статья поступила 1.08.88

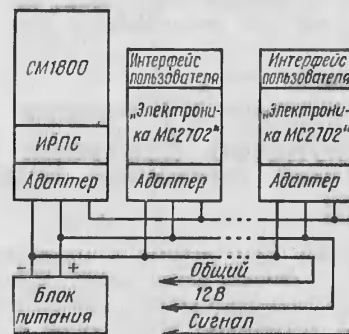


Рис. 3. Функциональная схема малой локальной вычислительной сети

адреса. Во втором случае контроллер может функционировать без пульта с клавиатурой. Обращение к драйверу осуществляется сразу после включения питания контроллера.

Директивы дистанционного отладчика: А — ввод данных в ассемблерной мнемонике, D — вывод содержимого памяти в шестнадцатеричном коде, F — запоминание адреса памяти, S — чтение и измерение содержимого области памяти, L — вывод содержимого области памяти в ассемблерной мнемонике, С — пуск по адресу с заданием до двух точек останова, Т — трассировка программы с выводом содержимого регистров микропроцессора, U — трассировка без промежуточного вывода содержимого регистров, M — передача содержимого области памяти по последовательному каналу связи из микроЭВМ в контроллер.

Режимы работы драйвера связи, размещаемого в контроллере: чтение из памяти и запись в нее, пуск по адресу с заданием точек останова, обмен данными по последовательному интерфейсу связи. В рассматриваемой системе к СМ1800 можно подключить через адаптер одной общей двухпроводной линии связи несколько контроллеров на расстоянии 10 м друг от друга и центральной ЭВМ. На этой базе разработан протокол обмена для малой локальной вычислительной сети (рис. 3). Линия запитывается от отдельного источника.

Телефон 74-35-02, Казань

УДК 681.3.06

А. Л. Ковалев, П. В. Кренивич, С. Л. Лизенко

СИСТЕМА МЕНЮ ДЛЯ ПЕРСОНАЛЬНОЙ ЭВМ

Система МЕНЮ РТК-микро [1] реализует пользовательский интерфейс типа меню для ПЭВМ и микроЭВМ «Электроника 60», ДВК2, ДВК3, «Электроника 85», а также мини ЭВМ СМ4 в среде ОС РАФОС (ФОДОС, ОС ДВК). Она обеспечивает естественный интерфейс пользователя с системными и прикладными программами через меню, которое состоит из выведенных на экран дисплея заголовка и перечня названий работ, составляемых самим пользователем:

СРЕДСТВА ПРОГРАММИРОВАНИЯ РТК-МИКРО
РЕДАКТИРОВАНИЕ ГРАФИЧЕСКИХ ПРОГРАММ
ТРАНСЛЯЦИЯ С ГРАФИЧЕСКОГО ПАСКАЛЯ
ТРАНСЛЯЦИЯ С ГРАФИЧЕСКОГО АССЕМБЛЕРА
ТРАНСЛЯЦИЯ С АССЕМБЛЕРА И КОМПОНОВКА ПРОГРАММЫ
ПЕЧАТЬ ГРАФИЧЕСКИХ ПРОГРАММ
ВЫВОД КАТАЛОГА РАБОЧЕГО УСТРОЙСТВА
РАБОТА С ФАЙЛАМИ
СПРАВКА ПО СИСТЕМЕ МЕНЮ
РЕДАКТИРОВАНИЕ МЕНЮ

Работы меню могут быть трех типов: команды монитора ОС, справка, ссылка на другое меню.

Команды монитора ОС. Работа меню этого типа позволяет выполнить заданную последовательность команд, при записи которых допускаются формальные параметры. Каждому параметру ставится в соответствие текст запроса, выдаваемый на экран дисплея при выполнении работы. В ответ с клавиатуры вводится соответствующее значение параметра. Кроме того, может быть задано значение формального параметра, используемое по умолчанию. Например, описание работы типа команды монитора:

```
R MACRO
& 1=&1
C
LINK &1, &2
RU &1
```

описание формальных параметров:

```
1=ИМЯ ПРОГРАММЫ
2=ИМЯ БИБЛИОТЕКИ
#2=LIBRTK
```

Данная работа предназначена для трансляции программы, написанной на Макроассемблере, компоновки ее с библиотекой и выполнения результирующей рабочей программы. Перед выполнением работы запрашиваются имена транслируемой программы и библиотеки, причем по умолчанию используется библиотека с именем LIBRTK.

Справка. С данной работой связан текстовый файл, который при ее выполнении выводится на экран и может просматриваться в режиме листания. Файл, содержащий текст справки, готовится любым текстовым редактором.

Ссылка на меню. Выполнение этой работы обеспечивает переход на другое меню, указанное в описании работы, связывая, таким образом, различные меню в многоуровневую систему.

Необходимая работа меню выбирается и выполняется путем перемещения указателя (стрелка на экране) с помощью функциональных клавиш «стрелка вверх», «стрелка вниз» и нажатия функциональной клавиши «выполнение». В меню может входить стартовая работа. Эта работа позволяет выполнить необходимые для других работ меню начальные действия, например установку режимов, назначе-

ние устройств, либо вывести на экран текст рекламного сообщения, справки о назначении и правилах использования данного меню и т. п.

Меню может быть защищено паролем, ограничивающим круг лиц, допускаемых к использованию данного меню.

С работами типа команды монитора и ссылки на меню можно соотнести текстовый файл, содержащий некоторую справку о данной работе, например текст помощи. Этот текст выводится на экран по нажатию клавиши «?».

В состав системы меню входят три программы: редактор (EDIMEN), интерпретатор (INTMEN), программа печати (PRIMEN).

Редактор меню позволяет выполнять следующие действия: ввод и редактирование заголовка меню; ввод, редактирование и удаление работы, включая и стартовую; перестановку работ; просмотр описаний работ; изменение пароля меню. Редактирование производится в экранном режиме с использованием оконного интерфейса и управляющих меню. При работе редактора на экран выводится редактируемое меню. После выбора в нем для редактирования работы в специальном окне экрана выводится управляющее меню, содержащее перечень допустимых при редактировании действий:

СРЕДСТВА ПРОГРАММИРОВАНИЯ
РТК МИ
РЕДАКТИРОВАНИЕ ГРАФИЧЕСКИХ
ПРОГР
ТРАНСЛЯЦИЯ С ГРАФИЧЕСКОГО
ПАСКАЛЯ
ТРАНСЛЯЦИЯ С ГРАФИЧЕСКОГО
АССЕМБ
ТРАНСЛЯЦИЯ С АССЕМБЛЕРА И
КОМПОН
ПЕЧАТЬ ГРАФИЧЕСКИХ ПРОГРАММ
ВЫВОД КАТАЛОГА РАБОЧЕГО
УСТРОЙСТ
РАБОТА С ФАЙЛАМИ
СПРАВКА ПО СИСТЕМЕ МЕНЮ
РЕДАКТИРОВАНИЕ МЕНЮ

Функции редактора меню
Ввод работы
Редактирование работы
Удаление работы
Считывание работы в буфер
Ввод стартовой работы
Редактирование заголовка меню
Изменение пароля
Запись меню в файл

При вводе и редактировании работы экран разделяется на окна, служащие для отображения и изменения отдельных компонент описания работы: имени и типа работы, команд монитора, формальных параметров и др. Доступ к редактированию компонент описания производится через управляющее меню, в котором перечислены их названия. Задание и изменение ряда компонент (тип работы, способ возврата в меню после выполнения работы) осуществляются путем выбора из возможных значений в дополнительном меню. Редактирование остальных компонент описания работы выполняется текстовым редактором в соответствующем окне на экране дисплея (см. таблицу). Ввод новой работы заключается в последовательном задании компонент ее описания в соответствующих окнах.

Интерпретатор меню. Созданное редактором конкретное меню хранится в виде специального файла, который интерпретируется при выполнении меню программой. Кроме выбора и выполнения работ меню интерпретатор обеспечивает движение по иерархии взаимосвязанных меню, а также позволяет, не выходя из меню, непосредственно вводить и выполнять команды монитора ОС.

Сервисная программа печати выводит описание меню, включая описания всех компонент входящих в него работ, в файл печати, который может служить документацией на данное меню.

Система меню позволяет создавать гибкий проблемно-ориентированный интерфейс взаимодействия человека с ЭВМ, важное достоинство которого — его простота, позволяющая практически за один сеанс освоить работу с систе-

Вид экрана при вводе и редактировании работы типа команды монитора

УДК 681.3.06

В. А. Зимнович, В. А. Сухман

ОРГАНИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА НА ОСНОВЕ СИСТЕМЫ МЕНЮ РТК-МИКРО

Успешное использование микро- и мини-ЭВМ непрофессиональными программистами (далее просто пользователями) зависит от степени освоения ими системных и прикладных средств ЭВМ. Одним из традиционных подходов к решению этой проблемы — специальная подготовка пользователей ЭВМ, отвлекающая их от основной работы на длительное время. Истоки подобного подхода заключаются в практически полном отсутствии для ЭВМ, например ДВК, средств, обеспечивающих «естественный» интерфейс пользователей с системными и прикладными программами.

Не менее актуальна и задача создания средств, позволяющих разгрузить профессиональных программистов от рутинной и трудоемкой работы. Кроме того, при разработке конкретных прикладных систем часто приходится заново создавать различные интерфейсные программы, так как существует острая нехватка подобных средств универсального назначения.

Альтернативный подход — разработка средств, закрывающих «внутреннюю кухню» работы с ЭВМ и обеспечивающих пользователей более удобными для него интерфейсами. Например, пакет прикладных программ под названием «система меню». Он разработан в ИК АН УССР в составе технологического комплекса РТК-микро, предназначенного для выполнения различных работ на микроЭВМ типа ДВК, «Электроника 85» и некоторых других ЭВМ в среде ОС ФОДОС или РАФОС [1,2].

Авторами данной статьи система меню применяется в течение четырех лет и позволяет эффективно решать следующие задачи: создание специализированных комплексов для обработки информации пользователями-непрограммистами; автоматизации разработки и ведения программной документации с отслеживанием требований ЕСПД; ведение фонда алгоритмов и программ, архивов программных разработок; сопровождение программных разработок; создание прототипа технологической среды для разработки программного обеспечения. Широкий спектр применения системы меню обусловлен тем, что она простыми средствами обеспечивает комплексирование различных прикладных программ в единую систему, организацию иерархического доступа к информации и технологических маршрутов обработки информации.

Рассмотрим примеры использования системы меню.

Введение библиографической информации. Система меню позволила объединить в единое целое программы обслуживания базы данных (БД) с библиографической информацией и технологию обработки этой информации. В качестве СУБД была использована РБД-микро, также входящая в состав РТК-микро [2]. В стартовый файл, запускаемый при загрузке ОС, был вставлен вызов системы меню, поэтому на терминал выводилось головное меню системы ведения библиографической информации и пользователь-библиограф сразу мог работать с ней. Названия работ в меню соответствовали их назначению и были составлены в терминах библиографической работы.

Процесс создания библиографической системы заключался в том, что специалист, хорошо знающий ОС ФОДОС, РБД-микро и систему меню, подготовил на основе предварительных консультаций с библиографом начальный вариант системы. На это ушло несколько часов. При этом была подготовлена БД с оговоренной структурой информации и головное меню с операциями заполнения БД, поиска и печати информации из БД. После этого библиограф был ознакомлен с библиографической системой и обучен работе с ней. На это ушло еще несколько часов. Причем на освоение работы с системой меню неподготовлен-

Компоненты работы → Имя работы Тип работы Способ возврата в меню	Тип работы: команды монитора Способ возврата в меню: безусловный
Команды монитора Формальные параметры Справка о работе Запись работы в меню	

Имя работы	ТРАНСЛЯЦИЯ С АССЕМБЛЕРА И КОМПИЛОВАНИЕ ПРОГРАММЫ
Команды монитора	R MACRO &1+&1 AC LIN K &1, &2 RU &1
Формальные параметры	1=ИМЯ ПРОГРАММЫ? 2=ИМЯ БИБЛИОТЕКИ? =2=LIBRTK

мой. При вызове системы меню из стартового командного файла головное меню появляется на экране дисплея сразу после загрузки ОС, полностью закрывая, таким образом, от неподготовленного пользователя командный язык ОС.

Программиста-профессионала система меню разгружает от многих рутинных работ, освобождает от необходимости запоминания десятков и сотен названий различных команд, программ, файлов, параметров и их значений, режимов работы, при этом существенно укрупняя операции взаимодействия с ЭВМ. Высокая оперативность, позволяющая за считанные минуты создать и модифицировать меню (в последнем случае даже не выходя из него), делает систему легко приспособляемой к повседневным нуждам разработки, обеспечивая в то же время возможность накопления опыта путем закрепления в виде меню сложившихся технологических приемов организации работ.

За пять лет существования система меню использовалась в различных организациях для создания информационно-справочных, обучающих, архивных, управляющих систем, систем подготовки и обработки документации и т. д.

Система меню реализована с помощью Р-технологии программирования на языках технологического комплекса РТК-микро [1] графический Паскаль и графический ассемблер. Общий объем системы — 30 Кбайт, из них объем интерпретатора меню — 5 Кбайт.

Телефон 266-00-97, Киев

ЛИТЕРАТУРА

1. Вельбицкий И. В., Ковалев А. Л. Графический стиль программирования для персональной ЭВМ // Микропроцессорные средства и системы.— 1985.— № 4.— С. 46—51.
2. Кочегарова С. И., Лолейт А. Т. Программные средства для микроЭВМ, входящие в ФАП МСВт // Электронная промышленность.— 1986.— № 9.— С. 31—34.
3. Кочегарова С. И., Лолейт А. Т., Слезко Л. С. Средства обработки текстовой информации на ДВК2М // Электронная промышленность.— 1986.— № 9.— С. 34—39.

Статья поступила 30.02.87

ного с точки зрения работы на ЭВМ библиографа ушло несколько минут. Основное время заняло изучение работы со средствами РБД-микром. Далее библиограф начал работать самостоятельно. Он заполнял БД, выполнял поиск и печать информации из БД, регулярно после каждого сеанса работы делал копию БД. В процессе работы выявлена необходимость изменения структуры БД. Реконфигурация БД проведена специалистом совместно с библиографом. На протяжении довольно длительного периода (нескольких месяцев) библиограф работал самостоятельно, изредка обращаясь к специалисту за консультацией.

Разработка и обработка различной документации. С точки зрения пользователя, меню позволяет обеспечивать: представление структуры разрабатываемого документа; сведение к минимуму числа команд программы форматирования документов, необходимых для работы пользователя, реализацию операций по обработке документа в виде работ меню.

В настоящее время разработаны средства для автоматического формирования файла сборки документа по структуре меню, его представляющей. Создана заготовка для разработки технического задания — «живой» руководящий материал. При этом структура технического задания формировалась в строгом соответствии с требованиями ЕСПД и с учетом достигнутого опыта разработки программного обеспечения. Заготовка позволила избежать неполноты документа с точки зрения наличия всех необходимых частей документа. Снизилась вероятность ошибок в командах форматирования документа, так как большинство команд генерировалось автоматически при сборке документа, а от оператора требовалось знание, как правило, только команд начала абзаца и перечисления.

До определенного момента у авторов статьи было мнение, что система меню удобна только для пользователей непрофессионалов и малоэффективна для целей автоматизации труда профессиональных разработчиков программного обеспечения. Это мнение коренным образом изменилось после того, как система меню была использована для ведения фонда алгоритмов и программ и разработок архивов. При этом система меню эксплуатировалась на мини-ЭВМ под управлением TS-монитора в многопользовательском режиме. Именно при работе на машинах с большей внешней памятью наиболее остро встает вопрос управления информацией, объем которой достигает значительных размеров. Под управлением TS-монитора в многопользовательском режиме проведена небольшая адаптация интерпретатора и редактора системы меню, исходные тексты программ которой были получены ее авторами.

Для ведения фонда и архивов разработок система меню использована как средство организации иерархического способа доступа к информации. При этом сама информация располагалась в виде виртуальных носителей и файлов, доступ к которым обеспечивался стандартными средствами ОС РАФОС, запускаемыми из меню. Таким образом, система меню обеспечивает при ведении фонда быстрый поиск информации (через иерархию меню) и автоматическую настройку на соответствующий ей виртуальный носитель. По такому же принципу организовывались и архивы разработок программного обеспечения.

Средства сопровождения программных разработок. Использование системы ведения фонда и архивов показало возможность сопровождения программных разработок этими же средствами. Система меню позволила представлять технологические маршруты (последовательность операций) при разработке программ в виде меню работ, гибко отслеживать особенности разрабатываемых программных систем и автоматизировать значительный объем рутинных и трудоемких операций при разработке. Это проявлялось в возможности отслеживания структуры программной разработки и принятой технологии программирования в виде соответствующего набора меню, связанных между собой.

Прототип мобильной технологической среды. Средства системы меню, дополнительные программы — прототип мобильной технологической среды, наборы разработанных

командных файлов и меню общего назначения составили основу пакета прикладных программ (ППП) для организации технологической среды разработчиков программного обеспечения под названием АРХИВ. ППП АРХИВ успешно используется разработчиками программного обеспечения в ряде организаций для поддержки различных технологий программирования.

Все разнообразие применения системы меню базируется на таких ее отличительных свойствах, как:

простота интерфейса, значительно снижающая напряжение пользователя при работе с системой и позволяющая практически «мгновенно» научить работать с системой меню неподготовленного пользователя;

«естественность» интерфейса, представленного названиями работ меню на естественном (русском) языке. Эти же свойства дают возможность создавать свой проблемно-ориентированный язык для работы с системой;

гибкость интерфейса, позволяющая оперативно добавлять в меню новые работы, изменять или удалять старые, сцеплять или разъединять меню между собой;

унификация интерфейса для решения различных задач.

Телефон 532-96-72, Москва

ЛИТЕРАТУРА

1. Вельбицкий И. В., Ковалев А. Л. Графический стиль программирования для персональной ЭВМ // Микропроцессорные средства и системы. — 1985. — № 4. — С. 46—51.
2. Ковалев А. Л. Три уровня инструментальной поддержки Р-технологии программирования на персональной ЭВМ // 2-я Всесоюз. конф. «Технология программирования»: Тез. докл. — Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1986, Ч. 2 — С. 177—180.

Статья поступила 30.01.88

УДК 681.3.06

Д. А. Черников, К. А. Черников

ПАКЕТ ПРОГРАММ ДЛЯ РАБОТЫ МИКРО-ЭВМ «ЭЛЕКТРОНИКА БК-0010» В КОМПЛЕКТЕ С ПЕЧАТАЮЩИМ УСТРОЙСТВОМ

Подключение печатающего устройства (ПУ) к БК-0010 существенно расширяет спектр возможностей компьютера. Авторами разработан пакет программ для работы БК-0010 в комплексе со стандартной термолетатью (в комплекте с ДВК-2) или другим ПУ с аналогичным интерфейсом. В первый момент было желание разработать одну универсальную программу, выполняющую все функции, которые могут потребоваться при печати с БК-0010. Но универсальная программа стала занимать довольно значительный участок и без того малой памяти, что заставило отказаться от этого варианта и создать пакет для работы в различных режимах (ФОКАЛЕ, Мониторе, БЕЙСИКЕ-87...) и печати текстовых файлов различных форматов (MIKRO8S, EDASP...). В пакет входят следующие программы:

DIR/PRI — печать каталогов файлов (с указанием имени и типа файла, а также его начального адреса и длины), записанных на внешний носитель. Все имена прочитанных файлов заносятся в буфер (в формате редактора текстов EDASP), который может быть записан на внешний носитель и в дальнейшем считываться из редактора EDASP как простой текстовый файл. Существует возможность многократной печати буфера (программа работает в диалоговом режиме).

PRI/FOC:DOC — печать текстов или результатов работы программ, написанных на ФОКАЛЕ. Печать производится с разбиением текста на страницы с указанием в начале каждой имени работающей (или распечатываемой) программы и номера страницы. Существует вариант этой программы и без разбиения на страницы. После

запуска программы с адреса ее загрузки управление передается ФОКАЛУ, затем, используя его стандартные команды, можно считать программу, текст или результаты работы которой требуется вывести на печать. Необходимо лишь следить за тем, чтобы текст программы, написанной на ФОКАЛЕ, не затер программу печати, для чего последнюю нужно загружать в дальние ячейки ОЗУ (программа занимает 1150 адресов) или, если программа на ФОКАЛЕ занимает всю область ОЗУ, переходить в режим расширенной памяти. Для включения режима печати надо указать команду X FCHR(1). Далее любой текст, выводимый на дисплей, будет дублироваться на ПУ. Выключается режим печати по команде X FCHR(2).

PRI/ED/K — печать файлов, написанных в редакторах EDASP и МИКРО8К, или других аналогичного формата. Печать осуществляется с разбиением на страницы и указанием в начале каждой имени печатаемой программы и номера страницы.

PRI/S — печать файлов, написанных в редакторе МИКРО8С, или других файлов аналогичного формата. Работа программы полностью аналогична PRI/ED/K.

PRI/MIK — дублирование на печать любой информации, выводимой на дисплей при работе в мониторе (например, распечатывает содержимое ячеек с адреса 0 по 1000), или печать нестандартных текстовых файлов (например, адресов меток операторов, получаемых при трансляции программы, написанной на МИКРО), или вывод любой информации, полученной после старта какой-либо программы из монитора (например, инструкций к некоторым играм, выполненных как законченные программы, а не в виде текстовых файлов). Вывод на ПУ начинается после ввода с клавиатуры или вставки в выводимый текст двух точек с запятой. После этого любая информация, выводимая на дисплей, будет дублироваться на ПУ. Выключается печать после ввода подряд двух звездочек — **. (Если в выводимом тексте встречаются две звездочки подряд, то символ окончания печати нужно выбрать другим. Это делается заменой кода звездочки на другой.)

PRI/BAS — печать шести текстов или результатов выполнения программ, написанных на БЕЙСИКе для БК-0010 со встроенным ФОКАЛОМ. Программа получилась настолько маленькой, что ее удалось разместить в области системного ОЗУ, которая при работе БЕЙСИКа остается незадействованной. Режим печати включается командой РОКЕ 24, 250, после чего любой текст, выводимый на дисплей (например, командой LIST) будет дублироваться на ПУ. Выключается печать командой РОКЕ 24, —32694. Особенности этих программ:

символы, декодировка которых отсутствует в ПУ, заменяются на *;

длина печатаемой строки — 80 символов. Если символов в строке больше 80, то оставшиеся символы переносятся на следующую строку;

одна страница текста содержит 64 строчки; вывод на ПУ ведется параллельно с выводом на дисплей. Работа программ без ПУ невозможна (за исключением DIR/PRI);

в программе предусмотрена печать всех букв русского и латинского алфавитов как заглавных, так и строчных.

Сопряжение с печатью программное, ПУ подключается к разъему пользователя. По младшим семи разрядам порта вывода на ПУ передаются данные, по восьмому разряду порта идет сигнал, который подается на контакт СИ4 разъема термопечати (для переключения режимов работы ПУ), девятый разряд — сигнал синхронизации. Сигнал СИЗ на разъеме термопечати подключен к выводу «Корпус», разряд готовности не используется (считается, что оба устройства всегда готовы к передаче данных). Остальные контакты разъема термопечати не используются.

Телефон 582-45-34, Москва

Сообщение поступило 26.05.88

УДК 681.322.042

В. Л. Лавровский

ГРАФИКА ЛОКАЛЬНОЙ СЕТИ МИКРОЭВМ БК-0010

Разработана локальная сеть, позволяющая работать на языке Паскаль с элементами формирования графических изображений на микроЭВМ БК-0010. Сеть промышленного исполнения функционирует без технических изменений. Работу сети в центральном комплексе обеспечивает программа связи с БК-0010 и операционная система ОС ДВК со средствами для работы на Паскале. В каждой из БК-0010 выполняется программа-монитор, содержащая редактор текста. Она преобразует команды оператора на трансляцию, загрузку исходного текста и оттранслированной программы в БК-0010 в соответствующие запросы к программе, исполняемой в центральном комплексе.

За основу модуля-ловушки прерываний взят модуль ST10 [1], недостатком которого можно считать невозможность выполнения многих команд ЕМТ БК-0010, так как все команды ЕМТ, не связанные с вводом-выводом на дисплей, воспринимаются им как ошибочные. Изменения, позволяющие устранить этот недостаток, свелись к преобразованию модуля ST10 от метки EXIT: до метки EXEMT:

EXIT:	CMPB RO, BNE HALT	#350 BKEMT
BKEMT:	MOV B BIC MOV MOV	RO.EMT #1,6(SP) (SP)+,R1 (SP)+,RO
EMT :	EMT JSR RTI	O PC, T60
EXEMT:		

Возможность использовать команды ЕМТ в программах позволила выводить графические изображения на экран монитора БК-0010. Для этого были написаны процедуры на Паскале, в которых применяются фрагменты на Макроассемблере. Процедуры оформлены, как внешние, оттранслированы отдельно и объединены в библиотеку GRAF.OBJ, расположенную на диске центрального комплекса. В состав библиотеки входят процедуры:

INIT — начальная инициация графического пакета.
POINT(X, Y:INTEGER; LIGHT:BOOLEAN) — формирование точки,
LINE(X1, Y1, X2, Y2:INTEGER; LIGHT:BOOLEAN) — формирование линии,
CIRCLE(X, Y, R:INTEGER; LIGHT:BOOLEAN) — построение окружности,
ARC(X, Y, R, LI, L2:INTEGER; LIGHT:BOOLEAN) — построение дуги,
BOX(XLD, YLD, XUR, YUR:INTEGER; LIGHT:BOOLEAN) — построение прямоугольника,
FILL(X, Y:INTEGER; LIGHT:BOOLEAN) — заполнение области,
CELL(I, J:INTEGER; VAR MAT:TMAT; LIGHT:BOOLEAN) — вывод на экран массива 8x8 точек, где тип TMAT-ARRAY [1..8] OF CHAR.

Эти процедуры подключаются к головной программе на этапе компоновки и должны быть описаны в ней, как внешние. Параметр LIGHT означает свечение или гашение точки: если LIGHT-TRUE, то точка с этими координатами засвечивается; если же LIGHT-FALSE, то точка гаснет.

В качестве примера рассмотрим реализацию процедуры LINE:

```
(* E+*)
PROCEDURE LINE (X1, Y1, X2, Y2:INTEGER; LIGHT:BOOLEAN);
BEGIN
  MOV (* C MOV #1,RO
      TSTB LIGHT(SP)
      BNE I
      MOV R0
      MOV X1 (SP),R1
      EMT X1(SP),R2
      MOV —03C
      MOV X2(SP),R1
      EMT Y2(SP),R2
      —032 *)
END;
```


Полезная особенность сети — возможность получения копии графического изображения с экрана БК-0010. Для этого содержимое ОЗУ БК-0010, соответствующее экранной памяти, записывается на диск центрального комплекса и затем распечатывается на матричном устройстве печати. Кроме того, возможна и немедленная (т. е. минуя диск) распечатка содержимого экранного ОЗУ.

Телефон 5-79-14, Орджоникидзе

ЛИТЕРАТУРА

1. Водянкин А. Г. Моисеенко В. И. Учебная локальная сеть микроЭВМ // Микропроцессорные средства и системы. — 1987. — № 4.
2. Математическое обеспечение микроЭВМ БК-0010. Руководство системного программиста. — 1985. — Т. 1, кн. 2.
3. Микрокомпьютерная графика / Д. Хирн, М. Бейкер. — М.: Мир, 1987.

Статья поступила 20.06.88

УДК 681.3.

В. Л. Баранов, Ю. Ф. Булаев, С. А. Павлючок

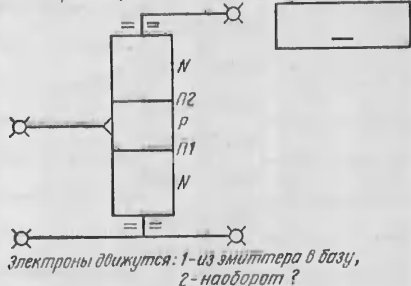
УЧЕБНАЯ ПРОГРАММА «ТРАНЗИСТОР»

Данная учебная Бейсик-программа предназначена для контроля и самоконтроля знаний с помощью ЭВМ «Искра 226» при изучении темы «Биполярный транзистор» в курсе «Промышленная электроника». Знания контролируются с помощью пяти вопросов и оцениваются по пятибалльной системе.

Правила работы с программой

1. Ввести программу в ЭВМ в соответствии с инструкцией по ее эксплуатации.
2. Прочитать на экране правила дальнейшей работы с программой и далее действовать в соответствии с ними, т. е. выполнять пункты 3—10.
3. Задать число учащихся.
4. Ознакомиться с контрольными вопросами и ввести время (в секундах) ответа на каждый из них.
5. Задать произвольные цифровые коды ответов с помощью любых цифр из ряда 0, 1, 2, ..., 9.
6. Познакомиться с содержанием вопросов и правилами ответов.

Биполярный транзистор N-P-N



7. Ввести в ЭВМ фамилию учащегося, после чего на экране появится учебная схема транзистора (см. рисунок).

```

10 REM ----- ЗАГЛОВОК -----
20 REM
                                     ИСКРА 226
30 REM BASIC 02 10.10.84
40 REM *****
50 REM * УЧЕБНАЯ ПРОГРАММА ТРАНЗИСТОР *
60 REM * LCST 1988 *
70 REM *****

80 DIM Y(20),Dx(20)16,K(1),L(3),X1(1),X2(1),X3(1):A=1:Z=0
90 PRINT HEX(0306):PRINT AT(1,20):PRINT "УЧЕБНАЯ ПРОГРАММА ТРАНЗИСТОР " ; "N-P-N"
100 PRINT " ДИАЛОГ С ПРЕПОДАВАТЕЛЕМ"
110 PRINT :PRINT " П' ; ПЕРЕД НАЧАЛОМ РАБОТЫ СЛЕДУЕТ ВВЕСТИ КОЛИЧЕСТВО УЧАЩИХСЯ (С
UT 1 ДО 20) И ВВЕСТИ ФАМИЛИЮ ПЕРВОГО УЧАЩЕГОСЯ, 'Г' ; 'О' ; ФАМИЛИЮ ОТВЕЧАЮЩЕГО ВВОДИТСЯ
IA ПЕРЕД ОТВЕТОМ ПО УКАЗАНИЮ ДИСПЛЕЯ."
120 PRINT " П' ; ОСЛЕ ТОГО, КАК ОТВЕТАТ ВСЕ УЧАЩИЕСЯ, НА ДИСПЛЕЕ ИЛИ АСПУ БУДУТ P
АСПЕЧАТАНЫ ВСЕФАМИЛИИ И ОЦЕНКИ." :PRINT
130 INPUT "ВВЕДИТЕ КОЛИЧЕСТВО УЧАЩИХСЯ (<=20) ,U
140 PRINT :PRINT " ПРЕДЛАГАЕМЫЕ ВОПРОСЫ : "
150 PRINT " 1. ОТКРЫТЫИ ИЛИ ЗАКРЫТЫИ ПЕРЕХОД П2? X1-ОТКРЫТЫИ , X2-ЗАКРЫТЫИ ?"
160 PRINT " 2. ЧТО БОЛЬШЕ : I' ; 'G' ; 'GX1' ИЛИ I' ; 'K' ; 'GX2' ?"
170 PRINT " 3. ЭЛЕКТРОНЫ ДВИЖУТСЯ: X2-ИЗ ЭМИТТЕРА В БАЗУ, X3-НАОБОРОТ ?"
180 PRINT " 4. УКАЖИТЕ ЭМИТТЕРС(X1),БАЗУ(X2),КОЛЛЕКТОРС(X3),НАЧИНАЯ С ВЕРХНЕЙ Ч
АСТИ ТР-РА"
190 PRINT " 5. ВВЕДИТЕ ПОЛЯРНОСТЬ НА БС1 ЗАКЛ1,КС1 ЗАКЛ1,ЗС2 ЗАКЛ1);ЕСЛИ + X3
,ЕСЛИ - X2':PRINT
200 PRINT "ВВЕДИТЕ ВРЕМЯ ОТВЕТА НА КАЖДЫИ ВОПРОС,НАЧИНАЯ С ПЕРВОГО,ЧЕРЕЗ ЗАПЯТУЮ
'INPUT L( )
210 INPUT "ВВЕДИТЕ КОДЫ ОТВЕТОВ НА ВОПРОСЫ - X1,X2,X3',X1K,X2K,X3K
220 FOR T=1TOU
230 PRINT HEX(0306):PRINT AT(1,20):PRINT "УЧЕБНАЯ ПРОГРАММА ТРАНЗИСТОР " ; "N-P-N"
240 PRINT :PRINT " ДИАЛОГ С УЧАЩИМСЯ"
250 PRINT "PRINT " В' ; НИЖЕЧАТЕЛЬНО ПРОЧИТАВ ВОПРОС,НАЖИМАЙТЕ НУЖНЫЕ ЦИФРОВЫЕ КЛ
АВШИИ ДЛЯ ОТВЕТА."
260 PRINT " Е' ; СЛИ ПРИ ОТВЕТЕ ТРЕБУЕТСЯ НАЖАТЬ НЕСКОЛЬКО КЛАВВИВ,ТО НАЖИМАЙТЕ
ОПРЕДЕЛЕНУЮ КЛАВВИВУ ТОЛЬКО ПОСЛЕ ВЫСВЕЧИВАНИЯ НА ДИСПЛЕЕ ПРЕДАДУЩЕЙ."
270 PRINT " В' ; ВЕДЕННЫЙ СИМВОЛ ИСПРАВЛЕНИЮ НЕ ПОДЛЕЖИТ."
280 PRINT " В' ; СЕГО ИМЕЕТСЯ 5 ВОПРОСОВ. 'Г' ; 'О' ; ВРЕМЯ ОТВЕТА НА КАЖДЫИ ВОПРОС УК
АЗАНО НА ДИСПЛЕЕ." :PRINT
290 LINPUT "ВВЕДИТЕ ФАМИЛИЮ",Dx(T)
300 PRINT HEX(0306):IF A=6THEN1210
310 PRINT AT(1,10):PRINT "ВЫПОЛНЯЕМЫИ П/П ТРАНЗИСТОР N-P-N ВРЕМЯ ОТВЕТА" ;L(A
);"СЕКУНД"
320 PRINT AT(10,53):PRINT "ВРЕМЯ"
330 PLOT (R,,),(<150,206.),(<40,,D),(<-40,-128.),(<40,,D),(<-40,0.),(<127,D),(<40,0.),
(<-127,D),(<-40,75.),(<40,,D),(<-40,-25.),(<40,,D)
340 PRINT AT(4,26):PRINT HEX(3D):PRINT AT(4,27):PRINT HEX(3D)
350 PRINT AT(16,26):PRINT HEX(3D):PRINT AT(16,27):PRINT HEX(3D)
360 PRINT AT(10,23):PRINT HEX(3C)
370 PLOT (<20,77.),(<12,D),(<0,0.),(<100,,D),(<-100,-142.),(<-12,D),(<-100,-1.),(<200
,,D),(<75,155.),(<80,,D),(<-30,D),(<-80,,D),(<30,D),(<-275,-77.),(<73,,D)
380 PRINT AT(10,12):PRINT HEX(24):PRINT AT(17,12):PRINT HEX(24):PRINT AT(17,41
):PRINT HEX(24):PRINT AT(3,41):PRINT HEX(24)
390 PRINT AT(10,30):PRINT HEX(50):PRINT AT(16,30):PRINT HEX(4E):PRINT AT(14,30)
:PRINT HEX(4E)
400 PRINT AT(9,30):PRINT HEX(F032):PRINT AT(11,30):PRINT HEX(F031)
410 ON AGOTO1030,1120,910,420,730
420 PRINT AT(9,22):PRINT HEX(3F):PRINT AT(5,22):PRINT HEX(3F):PRINT AT(15,22):
PRINT HEX(3F)
430 PRINT AT(20,5):PRINT "УКАЖИТЕ ЭМИТТЕРС';X1K;'БАЗУ' ;X2K;'К.КОЛЛЕКТОРС';X3K
;' ;НАЧИНАЯ С ВЕРХНЕЙ ЧАСТИ ТР-РА"
440 C=0:H=0:B=0:D=0:K=0:
450 FOR F=L(N)TOSTEP-1
460 KEYIN K,470,460:GOTO 540
470 C=C+1:PRINT AT(4,55+C):PRINT STR(KK,1,1)
480 IF K=X3THEN490:GOTO 500
490 H=H+1:GOTO 550
500 IF K=X2THEN510:GOTO 520
510 B=B+1:GOTO 550
520 IF K=X1THEN530:GOTO 540
530 D=D
540 GOSUB ' 2
550 IF C=3THEN570
560 NEXT F
570 IF H+B+D=3THEN580:GOSUB 700:GOTO 720
580 GOSUB 690:PRINT AT(9,22):PRINT HEX(E2):PRINT AT(5,22):PRINT HEX(EB):PRINT
AT(15,22):PRINT HEX(FC):GOSUB ' 1:GOTO 720
590 DEFFN ' 1
600 PRINT AT(1,10,60):PRINT AT(1,20):INPUT "ДЛЯ ПРОДОЛЖЕНИЯ НАЖМИ СР/ЛФ" ;-Dx
610 PRINT /10,HEX(0D):PRINT /05,HEX(0306)
620 KEYIN K,630,620:GOTO 630
630 RETURN
640 DEFFN ' 2
650 IF INT(F/2)<F/2THEN670
660 PRINT AT(5,55,5):PRINT AT(5,56+C):PRINT "-"
670 SELECT P6:PRINT AT(10,60):PRINT F:SELECT P:PRINT AT(5,55,5)
680 RETURN
690 PRINT HEX(07):PLOT (<0,115.),(<3,,C),(<.,,"ПРАВИЛЬНО") ;Z=Z+1:GOTO 710
700 PRINT HEX(07):PLOT (<0,115.),(<3,,C),(<.,,"ОШИБКА") ;GOSUB ' 1
710 RETURN
720 A=A+1:GOTO 300
730 PRINT AT(3,42):PRINT HEX(3F):PRINT AT(17,42):PRINT HEX(3F):PRINT AT(17,11)
:PRINT HEX(3F):PRINT AT(10,11):PRINT HEX(3F)
740 PRINT AT(20,5):PRINT "ВВЕДИТЕ ПОЛЯРНОСТЬ НА БС1 ЗАКЛ1,КС1 ЗАКЛ1,ЗС2 ЗАКЛ1);E
СЛИ + 'X3K' ;ЕСЛИ - 'X2K
750 C=0:H=0:B=10000:K=0:
760 FOR F=L(N)TOSTEP-1
770 KEYIN K,760,770:GOTO 830
780 D=C+1:PRINT AT(4,55+C):PRINT K;B=5/10
790 IF K=X3THEN800:GOTO 810
800 CONVERT K:ТОК:D=B/K:H=H+D
810 IF K=X2THEN820:GOTO 830
820 CONVERT K:ТОК:D=B/K:H=H+D
830 GOSUB ' 2
840 IF C=4THEN860
850 NEXT F

```

```

860 CONVERT X3*ТОХ3:CONVERT X2*ТОХ2:L=X3*1100+X2*11:IF H<L THEN B90:GOSUB 690
870 PRINT AT(13,12):PRINT HEX(35C2D):PRINT AT(10,41):PRINT HEX(556BC)
880 PRINT AT(3,42):PRINT HEX(2B):PRINT AT(17,42):PRINT HEX(2D):PRINT AT(17,11)
:PRINT HEX(2D):PRINT AT(10,11):PRINT HEX(2B):GOSUB ' 1:GOTO 900
890 GOSUB 700
900 A=A+1:GOTO 300
910 PRINT AT(20,5):PRINT "ЭЛЕКТРОНЫ ДВИЖУТСЯ":X2:--ИЗ ЭМИТТЕРА В БАЗУ, "X3:--
-HAДOPOT ?"
920 K=C: " :C=1:FOR F=L(A)TO9STEP-1
930 KEYIN K:940,930:GOTO 960
940 PRINT AT(4,57):PRINT K:
950 IF K=X2 THEN 700:GOSUB 700:GOTO 1020
960 GOSUB ' 2
970 NEXT F:GOSUB 700:GOTO 1020
980 GOSUB 690:PRINT AT(14,28):PRINT HEX(5E)
990 FOR S=1 TO 20:FOR I=1 TO 5:PRINT AT(16-I,27):PRINT HEX(4F):PRINT AT(16-I,27,1):N
EXT I
1000 FOR I=1 TO 4:PRINT AT(10,28-I):PRINT HEX(6F):PRINT AT(10,28-I,1):NEXT I
1010 FOR I=1 TO 6:PRINT AT(11-I,27):PRINT HEX(6F):PRINT AT(11-I,27,1):NEXT I:NEXT
S:GOSUB ' 1
1020 A=A+1:GOTO 300
1030 PRINT AT(20,5):PRINT "ОТКРЫТИЯ ИЛИ ЗАКРЫТИЯ ПЕРЕХОД П2:":X1:--"ОТКРЫТИЯ ,":
X2:--"ЗАКРЫТИЯ ?"
1040 K=C: " :C=1:FOR F=L(A)TO9STEP-1
1050 KEYIN K:1060,1050:GOTO 1080
1060 PRINT AT(4,57):PRINT K:
1070 IF K=X2 THEN 1100:GOSUB 700:GOTO 1110
1080 GOSUB ' 2
1090 NEXT F:GOSUB 700:GOTO 1110
1100 PLOT (7,10),(40,D),(-47,-10,):GOSUB 690:GOSUB ' 1
1110 A=A+1:GOTO 300
1120 PRINT AT(20,5):PRINT "ЧТО БОЛЬШЕ: I ' : ' : ' :X1M ' I ИЛИ I ' : ' : ' :X3M ' I ?"
1130 K=C: " :C=1:FOR F=L(A)TO9STEP-1
1140 KEYIN K:1150,1140:GOTO 1170
1150 PRINT AT(4,57):PRINT K:
1160 IF K=X1 THEN 1190:GOSUB 700:GOTO 1200
1170 GOSUB ' 2
1180 NEXT F:GOSUB 700:GOTO 1200
1190 GOSUB 690:PRINT AT(15,50):PRINT HEX(49DC3D49C22B49CB203D3E2049DC3E49CB):GOS
UB ' 1
1200 A=A+1:GOTO 300
1210 PRINT HEX(07):PLOT (50,100),(20,S),(3,D),(,,"СВОБОДЕН")
1220 PLOT (200,70),(2,D),(,,"ПРАВИЛЬНЫХ ОТВЕТОВ"):PRINT AT(7,55):PRINT Z
1230 PRINT AT(20,32):LINPUT "НАЖМИ CR/LF",-M:PRINT HEX(0306):PRINT /10,HEX(0D)
1240 Y(T)=Z:A=1:Z=0
1250 NEXT T:PRINT AT(1,10):INPUT "ВЫВОД ОЦЕНОК НА ДИСПЛЕИ(И) ИЛИ АЦПУ(0)":M:IF M
:THEN 1260:SELECT PRINT0C
1260 FOR T=1 TO V:IF Y(T):THEN 1270:Y(T)=2
1270 PRINT D(T):Y(T):NEXT T:SELECT PRINT0C

```

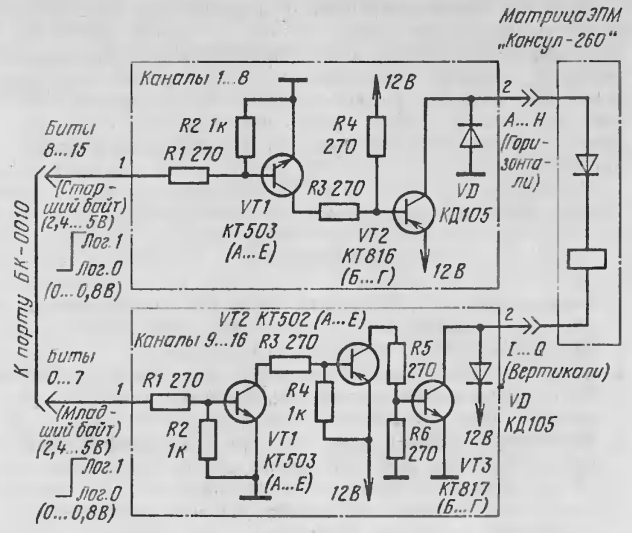


Рис. 1. Принципиальная схема переходного устройства

т. е. дают полную распечатку всей информации, выводимой на экран.

С драйверами АЦПУ работают в режиме «узкого» экрана, в котором емкость ОЗУ пользователя увеличена с 35776_в до 55776_в байт. Загрузка драйвера производится обычно, начиная с адреса 40000_в, хотя начальный адрес загрузки может быть любым.

При начальном запуске с адреса загрузки (в режиме «Монитор» — S<адрес загрузки>, а в режиме отладки «ТС» — G<адрес загрузки>) программа изменяет векторы

8. Ответить на вопросы путем нажатия на клавиши с цифровыми кодами, заданными в п. 5.
9. Прочитать на экране число правильных ответов.
10. Вывести результаты опроса на экран (код 1) или на АЦПУ (код 0). Пункты 1—5, 10 выполняются преподавателем, а пункты 6, 7 — учащимися.

Работоспособность и эффективность программы неоднократно проверена в условиях учебного заведения.

197227, Ленинград, Серебристый б-р, д. 18, кв. 226.
Павлуцкв С. А.

Сообщение поступило 15.06.88

УДК 681.3

Б. М. Лисенков

СОПРЯЖЕНИЕ ПИЩУЩЕЙ МАШИНЫ КОНСУЛ-260 С ПЭВМ ТИПА БК-0010

Для сопряжения электрифицированной пишущей машины (ЭПМ) «Консул-260» с ПЭВМ БК-0010 разработана схема переходного устройства и написана программа-драйвер АЦПУ, загружаемая в ПЭВМ с магнитной ленты при необходимости вывода информации на печать. Программа-драйвер написана на языке ассемблера в трех вариантах: в виде самостоятельной программы, загружаемой в любую область ОЗУ пользователя; дополнения («врезки») к интерпретатору языка ФОКАЛ, имеющемуся в ПЗУ машины; врезки в компилятор языка БЕЙСИК, при этом получившийся дополненный компилятор хранится на магнитной ленте и загружается вместо обычного компилятора для вывода информации на печать. Все разновидности драйвера обеспечивают вывод на печать текстов программ и данных, включая результаты решения задач,

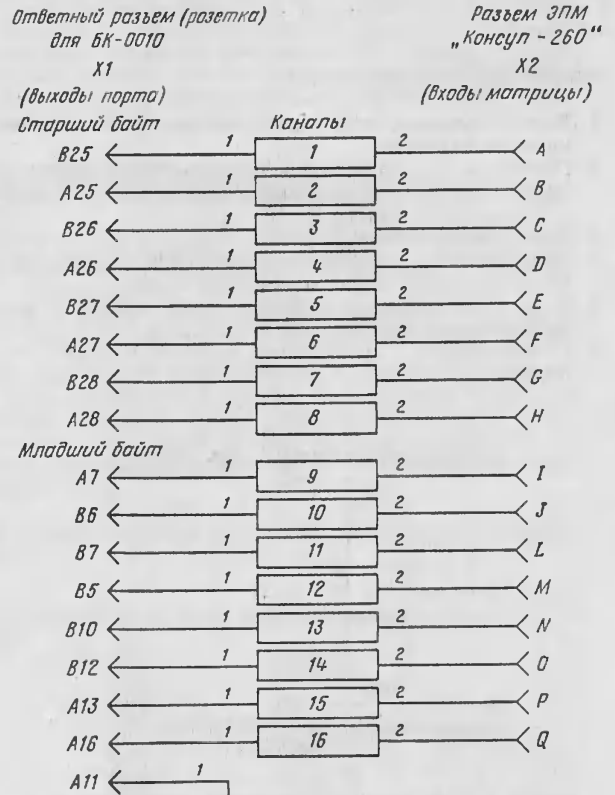


Рис. 2. Функциональная схема переходного устройства

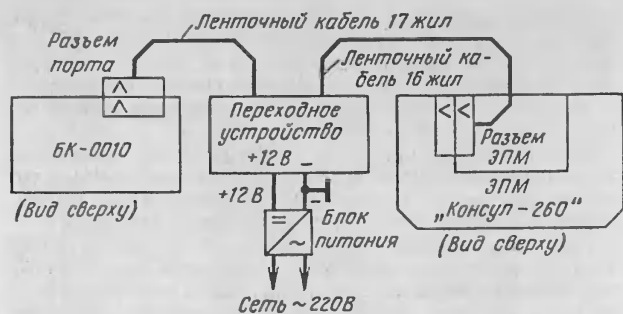


Рис. 3. Схема соединений переходного устройства

прерывания по клавиатуре (60) и ЕМТ (30) на адреса подпрограмм дополнительного анализа кодов символов клавиатуры и прерываний по ЕМТ. В самостоятельном варианте драйвера замена операторов NOP и HALT по адресам 14 и 16 на оператор JMP <начальный адрес> позволяет передавать управление после начального запуска на любую программу, расположенную в ОЗУ или ПЗУ. Такой механизм используется, в частности, в вариантех «врезок» в язык ФОКАЛ (оператор JMP 120000) и компилятор языка БЕЙСИК (оператор JMP 15174).

040000	012737	040066	000060	012737	040020	000030
040014	000240	000000	010546	016605	000002	014505
040030	042705	177400	022705	000016	001403	012605
040044	000137	100112	012605	002200	000021	001372
040060	012700	000040	000767	023727	177662	000015
040074	001406	023727	177662	000020	001410	000137
040110	101136	012737	040136	000030	004767	001020
040124	000002	004767	001066	000240	000002	010546
040140	016605	000002	014505	042705	177400	000167
040154	001062	000240	012605	000137	100112	012605
040170	022700	000214	001002	012700	000040	004767
040204	000002	000765	010346	010446	010546	004767
040220	000226	010704	004767	000650	126400	000510
040234	001444	005204	017505	010704	004767	000642
040250	126400	000600	001453	005204	017505	022700
040264	000012	000413	022700	000205	001410	012737
040300	077737	177714	004767	000160	004767	000194
040314	000451	012737	175577	177714	004767	000140
040330	004767	000114	004767	000140	000240	000240
040344	000435	012737	177177	177714	004767	000110
040360	004767	000064	004767	000310	000240	004767
040374	000072	004767	000046	000416	012737	176577
040410	177714	004767	000052	004767	000026	004767
040424	000252	000240	004767	000034	004767	000010
040440	012605	012604	012603	000207	012737	177777
040454	177714	012704	001000	000240	077402	000207
040470	012704	020000	077401	000207	012704	177777
040504	000240	077402	000207	000000	000000	000000
040520	077577	077737	077577	077773	077775	077776
040534	137737	137757	137167	137173	137175	137176
040550	157577	157737	157575	157167	157173	157175
040564	157776	167737	167757	167167	167173	167175
040600	167776	173737	173757	173167	173173	173175
040614	173776	173776	175737	175757	175767	175773
040630	175775	175776	176577	176737	176757	176767
040644	176773	176775	176776	177177	177337	177357
040660	177367	177373	177375	177376	177377	176774
040674	177373	177373	012704	000070	160904	060404
040710	090240	000240	060304	000506	006010	044050
040724	054120	024441	044537	054521	021033	041052
040740	051112	021532	041453	051513	022133	042074
040754	052114	133574	036445	120105	056525	023016
040770	043076	053116	007536	037447	076507	077527
041004	066500	067717	014430	164070	174360	034461
041020	164741	174761	031024	161072	171352	031772
041034	161473	171753	032373	162054	172354	137774
041050	026465	166745	176765	033162	163056	173356
041064	066376	027467	167747	160367	046501	047460
041100	012705	000070	162704	000014	000405	012705
041114	000070	162704	000034	000240	010403	000207
041130	000000	000000	016437	000310	177714	000207
041144	010045	010146	012701	000010	012700	000341
041160	104022	005201	012700	000343	104022	005201
041174	012700	000360	104022	005201	012700	000365
041210	104022	012601	012600	000207	012737	040020
041224	000030	010046	005000	104022	012600	000207
041240	022705	000016	001002	000167	176714	022705
041254	000020	001402	000167	176674	012605	010046
041270	010346	010203	000303	042702	177400	112100
041304	104016	120003	001401	077205	012603	012600
041320	000000	000000	000000	000000	000000	000000

Рис. 4. Дамп памяти драйвера печати

В составе программы-драйвера имеется несколько подпрограмм. Подпрограмма дополнительного анализа кодов символов клавиатуры служит для включения (клавиша ПЭВМ УСТ. ТАБ.) и выключения (СБР. ТАБ) режима печати. Используются также комбинации клавиш СУ с соответствующими алфавитными. Включение режима печати сопровождается появлением на служебной строке слова АЦПУ.

Подпрограмма анализа прерываний по ЕМТ обеспечивает вызов подпрограммы управления печатью при прерываниях по ЕМТ 16 (посимвольный вывод, применяется в самостоятельном варианте и с языком ФОКАЛ) или по ЕМТ 20 (вывод строки, используемый в основном с компилятором языка БЕЙСИК). При прерывании по ЕМТ 20 строка выводится путем многократного вызова в цикле оператора ЕМТ 16 и подпрограммы управления печатью. В любом случае подпрограмма управления печатью вызывается только при включенном режиме печати.

Подпрограмма управления печатью ставит в соответствие каждому печатному символу (на ЭПМ «Консул-260») комбинацию двоичных разрядов, соответствующую данному символу в коде матрицы включения электромагнитов ЭПМ, и заносит ее в выходной регистр 16-разрядного параллельного порта БК-0010 в виде числа, представляемого по одной единице в старшем и младшем байтах. Эти комбинации разрядов формируются с помощью таблицы, в которой каждому печатному символу поставлено в соответствие определенное двоичное число (выходной код), состоящее из указанных выше двух единиц. Вместо строчных букв, полуграфики и других символов, отсутствующих на ЭПМ «Консул-260», печатаются пробелы. Таблица сравнения символов побайтная, а таблица выходных кодов словная, так как каждый выходной код занимает 16 разрядов. Выходные коды хранятся в таблице в дополнительном коде в соответствии с принципом вывода информации на порт, принятым в ПЭВМ БК-0010.

Переходное устройство (рис. 1), включаемое между выходным регистром порта БК-0010 и входами матрицы электромагнитов ЭПМ, преобразует сигналы уровня ТТЛ, выдаваемые ПЭВМ, в сигналы высокого уровня, требуемые для возбуждения электромагнитов. Сопротивление электромагнитов ЭПМ постоянно току составляет 8 и 17 Ом, а ток необходимый для их срабатывания, должен быть в пределах 0,8...1,5 А. Напряжение питания цепей электромагнитов — 12 В. Переходное устройство содержит восемь «плюсовых» каналов, входы которых подключаются к выходам старшего байта порта БК-0010, и восемь «минусовых» каналов, входы которых подключаются к выходам младшего байта (рис. 2). Каждый канал содержит мощный ключ на транзисторе, управляемый соответствующими предвзательными каскадами, осуществляющими усиление и преобразование уровня сигнала. Входные каскады всех каналов одинаковы по схеме и параметрам. Схема соединений приведена на рис. 3, самостоятельный вариант разработанного драйвера — на рис. 4 в виде дампа памяти.

Переходное устройство эксплуатируется уже более полугодя и за это время показало себя надежным в работе.

Телефон 150-33-18, Москва

Статья поступила 8.07.88

УДК 681.321

В. Д. Циделко, Ю. В. Хохлов, А. М. Литвин, С. Г. Богославский

ОБУЧЕНИЕ МИКРОПРОЦЕССОРНОЙ ТЕХНИКЕ СПЕЦИАЛИСТОВ НЕВЫЧИСЛИТЕЛЬНЫХ СПЕЦИАЛЬНОСТЕЙ

В последнее время значительное внимание уделяется вопросу подготовки специалистов в области микропроцессорной техники, созданию методик и технических средств обучения. Одна из наиболее важных задач — определение

тематики и последовательности излагаемого материала, выбор принципов организации учебного процесса и его технической базы. Большинство авторов [1—6] придерживаются одинакового подхода, при котором практически не делается различие между обучением микропроцессорной (МП) технике и обычной подготовкой по вычислительной технике. Смешиваются два независимо развивающихся направления обучения: компьютерной грамотности и МП-технике. Один из основных критериев при выборе методики обучения — профессиональная ориентация обучающихся и характер их общения с МП-средствами. Можно выделить две категории специалистов — пользователи и разработчики.

Специалисты первой категории выступают в качестве потребителей изделий на уровне законченных устройств. Они обычно применяют в своей практической деятельности серийные микроЭВМ, поэтому и в учебном процессе целесообразно главное внимание уделить приобретению навыков работы с ними. Отметим, что практически все опубликованные работы по вопросам обучения МП-технике ориентированы на специалистов именно этой категории. Однако деятельность специалистов различных специальностей в областях измерительной техники, управления,

автоматики и так далее связана на практике не только с эксплуатацией готовых микроЭВМ. К их числу относится весьма многочисленная армия разработчиков радиоэлектронной аппаратуры различного назначения, использующих компоненты МП-техники на уровне элементной базы для построения специализированных микроконтроллеров (МК) и микроЭВМ, которые в наибольшей степени удовлетворяют требованиям конкретных технических заданий. Вопросам подготовки специалистов этой категории посвящена настоящая работа.

При обучении разработчиков аппаратуры со встроенными МП необходимо учитывать специфику их деятельности: последовательность этапов должна быть основана на принципе «от простого к сложному» по степени сложности излагаемого материала и структурной организации изучаемых микропроцессорных систем (МПС). Как в аналоговой технике трудно ожидать успешного применения операционных усилителей без знания работы транзисторных устройств, так и в данном случае использование и разработка сложных и высокоорганизованных систем со встроенными МП и МК будут более эффективными при базировании на глубоком и всестороннем изучении схемотехники микропроцессорных комплексов (МПК).

В общем виде можно предложить следующие этапы обучения: изучение организации, принципов функционирования и основных режимов работы конкретных типов МПК; системы команд (микрокоманд) МП и принятых способов адресации; принципов построения микроЭВМ на основе выбранного типа МПК и ее основных узлов; способов организации интерфейсной связи микроЭВМ и периферийными устройствами; особенностей построения, эксплуатации и программного обеспечения серийных микроЭВМ, построенных на основе изучаемого МПК.

Эти этапы положены в основу лабораторного курса обучения студентов на кафедре информационно-измерительной техники Киевского политехнического института. Основные цели курса:

знакомство студентов с принципами функционирования типичного МП с фиксированной системой команд;

обеспечение возможности изучения системы команд МП с визуальным наблюдением временных диаграмм работы каждой команды, принципов взаимодействия МП с органами управления и отображения цифровых измерительных приборов;

исследование особенностей построения измерительных приборов и систем со встроенными МП и алгоритмов коррекции погрешностей в микропроцессорных измерительных устройствах.

Обучение основам МП-техники проводится на примере МП КР580ИК80. При фиксированной системе команд процесс обучения намного упрощается и в некоторой степени

напоминает работу с программируемым калькулятором, поэтому главное внимание уделяется построению МПС и алгоритмам их функционирования, а не тонкостям микропрограммирования. Наличие развитой системы аппаратного и программного обеспечения этого комплекта делает его уникальным для учебного процесса.

Опыт обучения студентов и специалистов невычислительных специальностей основам МП-техники позволил выделить несколько характерных и неочевидных особенностей, влияющих на усвоение изучаемого материала.

Для лиц, впервые приступивших к изучению новой техники, необходимы простота и наглядность используемых технических средств обучения. МикроЭВМ отличаются друг от друга составом оборудования и разработанными мониторами. Возникает ситуация, когда изучаемому основу МП-техники приходится сначала предпринять усилия, чтобы усвоить специфическое программное обеспечение (принцип «от простого к сложному» нарушается). Применение директив монитора затрудняет понимание действия и этапов выполнения команд (структура исследуемой МПС становится неосознаваемой). В практической деятельности на этапе проектирования аппаратуры приходится создавать различные программы, а в ряде случаев даже мониторы и малые операционные системы, поэтому при обучении целесообразно предоставить всю область программируемой памяти в распоряжение пользователя, обеспечив его библиотечкой прикладных подпрограмм. Другая особенность — необходимость прямого общения обучаемого с МП. Непосредственный доступ к шинам и сигнальным линиям МПС и наблюдение реальных «живых» временных диаграмм ее работы важны прежде всего с психологической точки зрения: уменьшается существующее вначале чувство неизвестности и неуверенности, что позволяет не только убедиться в действительном соответствии теоретических и реальных диаграмм работы МП, но и глубже понять действие команд, визуально наблюдая этапы их выполнения.

Большое значение для разработчиков имеет наличие в технических устройствах обучения средств отображения и управления, которые имитируют аналогичные средства создаваемой аппаратуры. Подключение дисплея в качестве устройства отображения алфавитно-цифровой информации традиционно для универсальных микроЭВМ, однако нецелесообразно использовать его в качестве цифрового отсчетного устройства измерительного прибора. Средства управления должны не только обеспечивать все основные режимы работы МП, но и имитировать необходимые внешние сигналы и простейшие устройства ввода-вывода.

Анализ существующих средств обучения и функциональных характеристик серийных микроЭВМ позволил сделать вывод, что возможности их использования в рассматриваемом случае ограничены. Поэтому было принято решение разработать самостоятельно учебное микропроцессорное устройство, которое в наибольшей степени удовлетворяло бы поставленным целям [7].

Учебное микропроцессорное устройство (УМУ) — упрощенный вариант комбинированного цифрового измерительного прибора (ЦИП) со встроенным МК на основе МП КР580ИК80 для проведения лабораторного практикума по следующим темам: «Организация микроконтроллера на МПК КР580», «Система команд и программирование МП КР580ИК80», «Микропроцессорные ЦИП», «Алгоритмическая коррекция погрешностей в ЦИП с помощью МП».

Каждому функциональному узлу УМУ соответствует свой физический модуль. В состав МК входят центральный процессор (ЦП), модули ввода-контроля-отладки (ВКО), дисплей (ДПЛ) и клавиатура (КЛВ). Возможна установка дополнительных модулей — измерительных и памяти (ПМ).

Модуль ЦП выполнен на основе универсального МП типа КР580ИК80А, содержит ОЗУ статического типа на ИМС К565РУ2А (1 Кбайт) и РПЗУ на четырех схемах КР1601РРА (2 Кбайт). Такие объемы памяти вполне до-

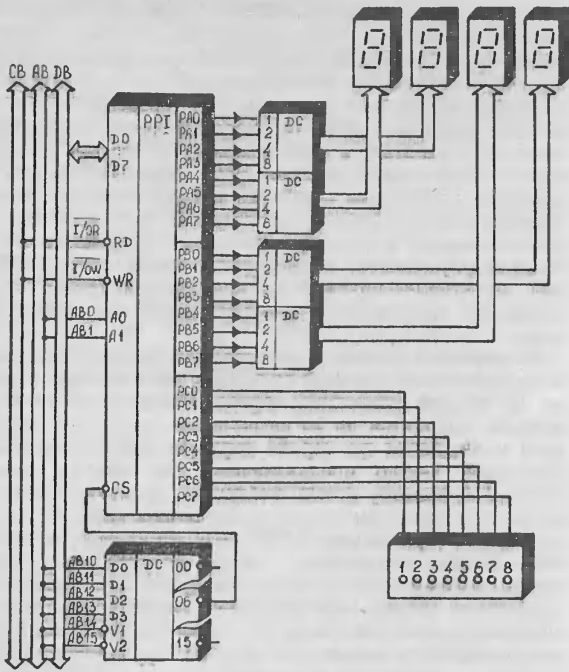


Рис. 1. Структурная схема дисплейного модуля

статочны для написания типичных учебных программ, которые заносятся и отлаживаются учащимися в оперативной памяти. В РПЗУ содержатся демонстрационные программы для проверки УМУ перед лабораторной работой, подпрограммы и таблицы констант.

Модуль ВКО позволяет обратиться к любой ячейке памяти или любому устройству ввода-вывода для записи/считывания информации, организовать непосредственный

доступ к шинам МК в режиме ПДП и контроль за ходом выполнения программ. Он содержит индикаторы состояния МП, устройства ПДП и задания режимов работы ЦП, регистры адреса (RGA), данных (RGD), схемы сопряжения с шинами МП и связан с клавиатурой управления на передней панели УМУ.

Дисплейный модуль — набор из четырех цифровых индикаторов, имитирующих цифровое отсчетное устройство ЦИП, и восьми светодиодов, образующих линейную шкалу (рис. 1), для исследования структур двоичных АЦП в качестве устройства индикации и контроля хода выполнения программ. Все индикаторы отображают информацию, записанную в регистры портов программируемого параллельного интерфейса КР580ИК55.

Модуль клавиатуры — блок из десяти переключателей на два положения со схемой устранения дребезга контактов (рис. 2). Переключатели В0...В7 используются для ввода числовой информации и формирования векторов прерывания IRQ0...IRQ7, подаваемых на программируемый контроллер прерываний КР580ВН59. Модули АЦП и частотомера-генератора — примеры измерительных модулей. Модуль АЦП (рис. 3) предназначен для проведения лабораторных работ по практической реализации аналого-цифрового преобразования средствами МП-техники поразрядного, следящего и развертывающего методов уравнивания. Он содержит интегратор, компаратор (С), ЦАП (DAC), аналоговые ключи (К), источники измеряемых напряжений (U_{x1} и U_{x2}), источник опорного напряжения ($U_{оп}$), дешифратор управления (DC). Для синхронизации и задания временных интервалов работы модуля в его состав введены генератор опорной частоты (G), интерфейс КР580ИК55 и таймер КР580ВИ53.

Частотомер-генератор (рис. 4) используется для организации работы устройства в режимах цифрового частотомера и синхронизатора сигналов сложной формы. Модуль включает в себя генератор опорной частоты (G), источники измеряемых частот (f_{x1} и f_{x2}), усилители (F), ЦАП (DAC) и таймер КР580ВИ53, управляемые модулем ЦП через порты интерфейса.

Передняя панель компактна и информативна, содержит необходимые средства управления, индикации и контрольные гнезда. В устройстве имеются два дополнительных

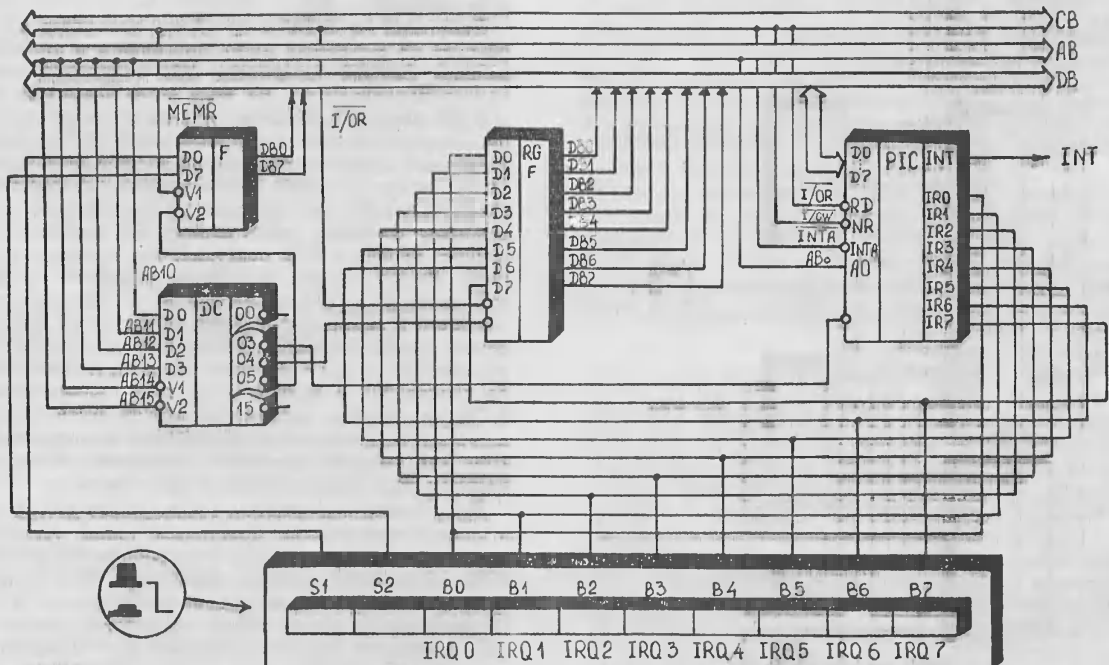
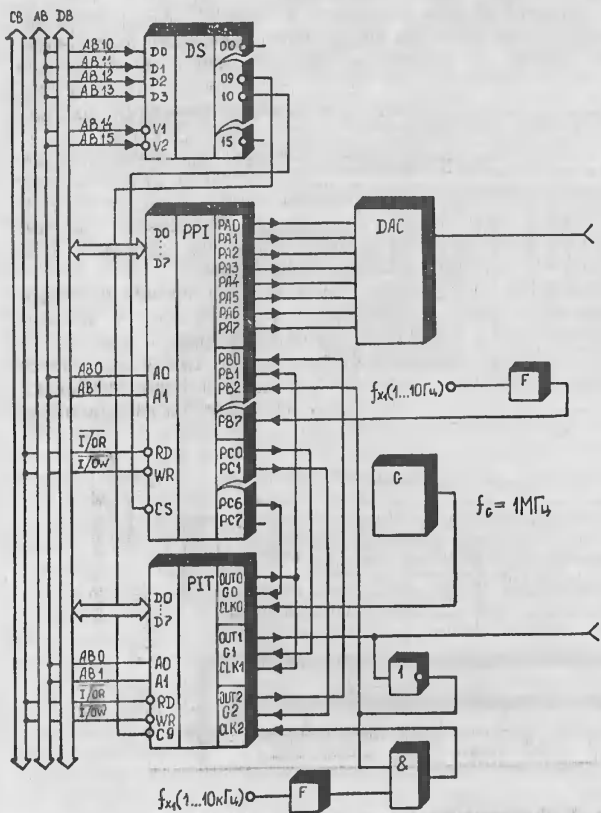
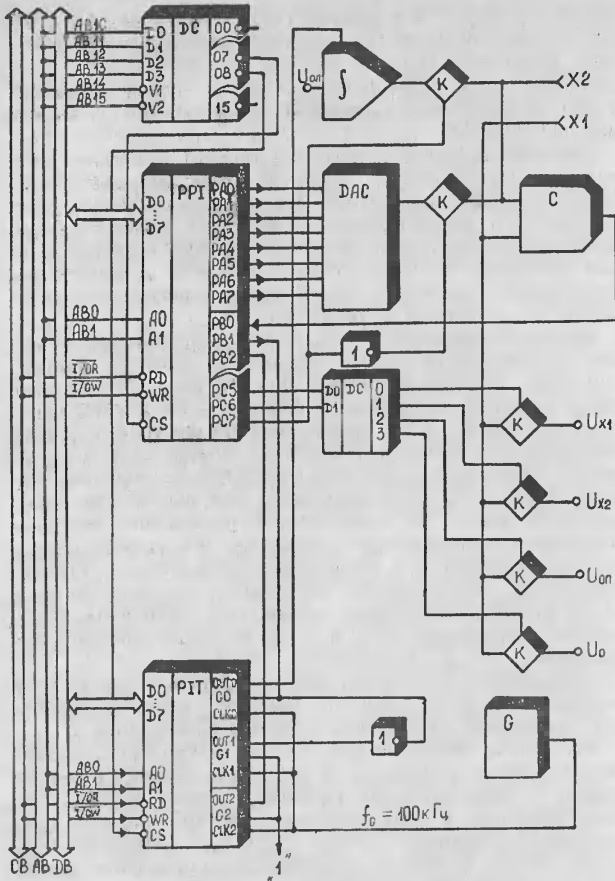


Рис. 2. Структурная схема модуля клавиатуры

Рис. 3. Измерительный модуль АЦП



индикатора. Цифровой 4-разрядный десятичный индикатор имитирует отсчетное устройство ЦИП и отображает содержание внутренних регистров МП (особенно удобно при отладке программ). Двоичный светодиодный 16-разрядный индикатор работает в двух режимах. В режиме А первые восемь светодиодов (считая слева направо) служат 8-разрядным двоичным линейным индикатором дисплейного модуля, вторые восемь — индицируют состояния МП, зафиксированные в регистре состояний. В режиме В этот индикатор указывает наименование одной сигнальной линии, мультиплексируемой на контрольное гнездо В, где визуально наблюдаются осциллограммы выбранного сигнала.

На передней панели переключаются режимы индикатора и передвигается световой указатель типа сигнала в режиме В; на ней размещены девять контрольных гнезд, на которые выводятся до 23 сигнальных линий для визуального наблюдения на экране двухлучевого осциллографа. Отдельное гнездо предназначено для вывода сигнала SYNC/M1, поэтому можно наблюдать временные диаграммы работы программ относительно начала каждого машинного цикла (при сигнале SYNC) или команды (при сигнале M1). Четыре контрольных гнезда соответствуют характерным точкам измерительных модулей для наблюдения аналоговых сигналов на различных этапах измерительного преобразования (например, в модуле АЦП это сигнал на выходе ЦАП в различных режимах уравнивания).

Основные режимы работы УМУ

Автоматическое выполнение программы — обычная работа МК и естественный ход (наблюдение этапов выполнения команд и работы узлов УМУ в контрольных точках). Пошаговое прохождение программы особенно удобно при устранении ошибок алгоритма, допущенных учащимися при его проектировании. Последовательный просмотр кодов команд, адресов и данных оказывает благоприятное психологическое воздействие на учащихся; становятся понятны принципы организации мультиплексированных шин и распределения потоков данных. Пошаговый режим обеспечивается аппаратными средствами, использующими режим готовности МП.

Установка по заданному адресу позволяет осуществить переход на желаемый адрес программы в автоматическом режиме (сигнал готовности МП снимается, вычисления приостанавливаются). На этом этапе возможно изменение режима работы (например, переход в режим с дальнейшим пошаговым выполнением программы). Во многих случаях начальный адрес программы не должен быть равен нулю. Такое состояние можно рассматривать как переход на произвольный адрес без выполнения команд (частный случай режима). На линии шины данных МП подается потенциал низкого уровня в МП, поступает код команды «нет операции» (NOP). Адрес остановки задается предварительной записью в регистр адреса RGA. Момент совпадения текущего и заданного адресов фиксируется схемой совпадения, выход которой, выведенный на специальное контрольное гнездо АДР, доступен оператору. Появление сигнала совпадения в данной точке будет наблюдаться также в любом другом режиме и может быть использовано для синхронизации осциллографа или логического анализатора при анализе временных диаграмм работы на произвольном участке выполняемой программы.

Ручной ввод информации обеспечивает доступ оператора к любому программно доступному узлу устройства для записи-чтения информации с ее одновременным отображением на соответствующем индикаторе. Обращение может быть с автоматическим приращением адреса и без него. Первый случай соответствует последовательному обращению к массиву памяти (например, при занесении учащи-

Рис. 4. Измерительный модуль частотомера-генератора

мися своих программ в ОЗУ в 16-ричном коде с клавиатуры передней панели). Регистры адреса (RGA) и данных (RGD) устанавливаются и сбрасываются в этом режиме и формируется сигнал подтверждения прерывания для контроллера KP580BH59.

При проектировании электронной аппаратуры, использующей МП KP580ИК80А, можно применять УМУ в качестве отладочного средства (в гнездо ЦП устанавливается переходная плата связи с МПС проектируемого изделия). При аналоговом выводе информации в поле контрольных точек с параллельным компарированием адресов УМУ выполняет функции упрощенного логического анализатора с представлением информации в виде временных диаграмм. УМУ полностью пригодно для изучения команд МК семейства K1816 и МП KM1821BM85A и систем на их основе. Единственное условие — замена модуля ЦП и добавление в коммутационную схему ограниченного числа сигнальных линий, выводимых через свободные контакты разъема.

252056, Киев, КПИ-2004; тел. 441-90-03

ЛИТЕРАТУРА

1. Грушвицкий Р. И., Коровацкий В. П., Преображенский А. В. Особенности построения и использования учебных микроЭВМ // Микропроцессорные средства и системы.— 1985.— № 2.— С. 75.
2. Гусев И. Т., Немчинов В. Н., Филиппов А. Г., Шагури И. И. Обучение специалистов применению и разработке микропроцессоров и микроЭВМ // Микропроцессорные средства и системы.— 1984.— № 2.— С. 88.
3. Мясников В. А. Подготовка специалистов по применению микропроцессорной техники // Микропроцессорные средства и системы.— 1984.— № 2.— С. 3.
4. Преснухин Л. Н., Фролов Г. И., Куправа Т. А., Безобразов В. С., Шахнов В. А. Учебный класс на основе диалоговых вычислительных комплексов // Микропроцессорные средства и системы.— 1985.— № 3.— С. 39.
5. Преснухин Л. Н., Кустов В. А., Зубарев П. В., Вильсон А. Л. Класс ДВК для изучения дисциплин «Радиоэлектроника» и «Импульсная техника» // Микропроцессорные средства и системы.— 1986.— № 4.— С. 66.
6. Широков Ю. Ф., Осипов Е. Н., Жданов В. И., Гутовец Н. И. Учебно-отладочное устройство «Электроника 580 // Микропроцессорные средства и системы.— 1985.— № 2.— С. 80.
7. Циделко В. Д., Павлишин Н. М., Хохлов Ю. В., Литвин А. М. Методические указания к лабораторным работам по курсу «Применение микропроцессоров в ИИТ». — Киев: КПИ, — 1987.

УДК 681.326.34

Статья поступила 22.10.88

В. И. Джиган, М. В. Мударисов

ПОВЫШЕНИЕ СКОРОСТИ ОБРАБОТКИ ИНФОРМАЦИИ В УПРАВЛЯЮЩЕЙ МИКРОЭВМ

При выполнении ряда прикладных задач с помощью микроЭВМ быстродействие имеющихся микропроцессорных средств может оказаться недостаточным. Возможны два решения этой проблемы — разработка новой микроЭВМ с требуемым быстродействием или дополнительных аппаратных средств, на которые перекладывается часть выполняемых микроЭВМ операций.

В настоящей работе рассматривается устройство, с помощью которого повышается скорость выполнения алгоритмов управления фазированной антенной решеткой (ФАР) в микроЭВМ семейства «Электроника», реализованной на основе МПК БИС серии K588. Возможное решение этой задачи — переложение часто встречающихся простых операций на аппаратные средства приемников

кодов управления фазовращателями, размещенными в модулях ФАР. Управляющая микроЭВМ при этом остается без изменений.

Приемник управляющего кода ориентирован на выполнение операций в симметричных двумерных ФАР с числом элементов 128 и 5-разрядными фазовращателями. Элементы ФАР можно представить в виде четырех подрешеток с номерами $(11)_2 \dots (00)_2$. Данные от управляющей микроЭВМ преобразуются в зависимости от номера подрешетки и в каждом из приемников кода выполняются операции по требуемым алгоритмам: строчно-столбцовому суммированию [1] компонент фазовых сдвигов в элементах решетки при управлении положением основного луча диаграммы направленности ФАР, возмущению фазового распределения в алгоритмах слежения за источником сигнала и по координатных градиентных алгоритмах адаптации ФАР к внешней помеховой обстановке [2]. Приемник реализован на основе БИС серии K588 и ИМС серии K564. Данные от управляющей микроЭВМ передаются и принимаются с помощью БИС контроллера мультиплексной линии связи (КМЛС) K588ВГ6 по двум сигнальным линиям: «Биполярный ноль» (Бип0) и «Биполярная единица» (Бип1) [3]. Для передачи используются посылки типа «Данные», а принятые и преобразованные данные перезаписываются в выходные регистры приемников кодов после обмена со всеми модулями решетки в режиме строчно-столбцового суммирования с помощью сигналов, формируемых КМЛС при приеме посылки типа «Команда».

Формат сообщений в этом режиме имеет вид: $(0 A4 A3 A2 A1 A0 Y4 Y3 Y2 Y1 Y0 X4 X3 X2 X1 X0)_2$, где 0 — признак данного режима, разряды $A0 \dots A4$ — адрес модуля в пределах одной подрешетки, $Y0 \dots Y4$ и $X0 \dots X4$ — компоненты фазового сдвига, вычисленные в управляющей микроЭВМ для подрешетки с номером $(00)_2$.

Выходной буфер КМЛС открывается по сигналу «Условие незаполненности регистра декодера» (УНРДК) после приема каждого из сообщений. При наличии низкого уровня этого сигнала на входе «Выборка кристалла регистра данных» (ВКД) выходной буфер КМЛС открывается независимо от вида принятой посылки. Схема распознавания (СР) анализирует сигналы УНРДК и «Команда» (рис. 1) и формирует импульсы отрицательной полярности с длительностью пауз между соседними сообщениями только после приема посылок типа «Данные» (рис. 2) (с их помощью информация записывается в промежуточный 16-разрядный регистр (Pr1). Регистр Pr1 позволяет выполнять операции в приемнике кода независимо от длительности пауз между поступлением очередных посылок, в течение которых на выходе открытого канала КМЛС хранится последнее принятое сообщение.

Сигналы разрядов $(Y0 \dots Y4$ и $X0 \dots X4)$ принятой информационной посылки поступают через каналы В мультиплексоров MX1 и MX2 и преобразователи кодов (ПК1) и (ПК2) на входы сумматора (См), где выполняются операции сложения во всех режимах работы приемника кодов. Преобразователи кодов изменяют знаки компонент

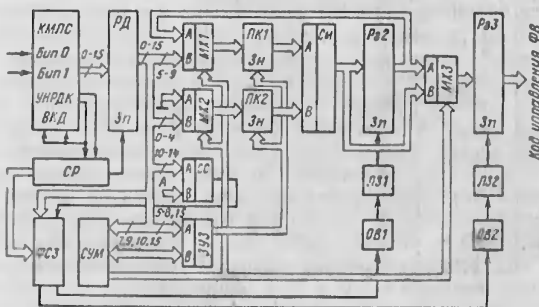


Рис. 1. Функциональная схема приемника кода

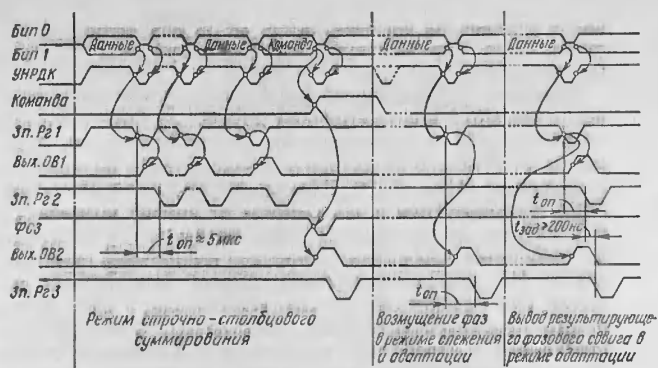


Рис. 2. Временные диаграммы работы приемника кода

фазовых сдвигов в зависимости от номера подрешетки с помощью запоминающего устройства знаков (ЗУЗ) при значении старшего (15) разряда информационной посылки, равном нулю.

Сумма записывается в Рг2 при совпадении адреса модуля решетки (А), заданного в пределах каждой из подрешеток на входах В схемы сравнения (СС), и разрядов (А0...А4) адресной части информационной посылки, поступающих на входы А СС. Информация записывается в регистр Рг2 по сигналу, вырабатываемому одновибратором (ОВ1). В режиме строчно-столбцового суммирования этот одновибратор запускается формирователем сигналов записи (ФСЗ) при уровне сигнала Лог.0 в 15-м разряде информационной посылки и Лог.1 — на выходе СС. Сигнал, формируемый одновибратором, задерживается линией задержки (ЛЗ1) на время выполнения операций преобразования кодов $t_{дл} \approx 5$ мкс (рис. 2). После передачи данных во все модули ФАР в режиме строчно-столбцового суммирования результат выполнения операций суммирования перезаписывается из Рг2 в выходные регистры Рг3 приемников кодов всех модулей ФАР. В этом режиме выходная Рг2 и входная Рг3 шины соединяются через канал А мультиплексора МХ3. Сигнал, формируемый одновибратором ОВ2, используется в качестве сигнала записи в Рг3. Одновибратор запускается ФСЗ при приеме посылки типа «Команда».

В режиме слежения за источником сигнала сообщения передаются в модуль ФАР в следующем формате: (1 *** **1 ЗНЗ ЗН2 ЗН1 ЗН0 ΔФ4 ΔФ3 ΔФ2 ΔФ1 ΔФ0)₂. Единица в старшем разряде указывает на выполнение режима, отличного от строчно-столбцового суммирования; звездочка — безразлично значение разрядов, единица в девятом разряде — признак выполнения операций приращений фазовых сдвигов в элементах ФАР на ΔФ4...ΔФ0 в соответствии с требуемыми знаками.

Знаки кодов приращений для подрешеток с номерами (11)₂—(00)₂ задаются соответственно значениями разрядов ЗНЗ...ЗН0 принятого сообщения. Низкий уровень сигналов этих разрядов говорит о приращении фазовых сдвигов в элементах подрешеток в соответствии со знаком числа (ΔФ0...ΔФ4)₂, а высокий — с его отрицательным значением. Знаки приращений фазовых сдвигов в элементах подрешеток в данном режиме выбираются также с помощью ЗУЗ (на шину А поступают сигналы ЗН0...ЗНЗ принятого сообщения). Текущие значения кодов управления фазовыми сдвигами, хранящиеся в Рг2, складываются со значением кода приращения фазового сдвига со знаком, определяемым номером подрешетки. Выходные сигналы ЗУЗ подключаются к управляющим входам преобразователей кодов при высоких уровнях сигналов принятой информационной посылки в 9-м и 15-м разрядах.

Результат выполнения операции в данном случае записывается сразу в выходной регистр Рг3. Сигнал перезаписи формируется, как в предыдущем режиме работы

приемника кодов, одновибратор ОВ2 запускается задержанным сигналом из ФСЗ при уровнях сигналов Лог.1 в 9-м и 15-м разрядах принятой информационной посылки.

Таким образом, в данном режиме результат приращений записывается одновременно в выходные регистры Рг3 во всех модулях ФАР.

Содержимое регистра Рг2 остается неизменным, так как может понадобиться в других режимах функционирования.

При выполнении алгоритмов адаптации информационных посылки передаются в модули решетки в форматах (1 А4 А3 А2 А1 А0 0 1 ПП ЗН1 ЗН0 ΔФ4 ΔФ3 ΔФ2 ΔФ1 ΔФ0)₂ и (1 А4 А3 А2 А1 А0 0 0 ПП ЗН1 ЗН0 КФ4 КФ3 КФ2 КФ1 КФ0)₂; в первом — команды на возмущение фаз сигналов в элементы подрешеток с адресами (А0...А4)₂, во втором по этому адресу выводятся значения фазовых сдвигов, рассчитанные в управляющей микроЭВМ. 8-й и 9-й разряды (01)₂ и (00)₂ указывают на признаки режима возмущения фаз сигналов и вывод новых значений фазовых сдвигов по адресу, разряд ЗН1 — знак изменений передаваемых значений кодов фазовых сдвигов (ΔФ4...ΔФ0 и КФ4...КФ0) в подрешетках (10)₂ и (11)₂, а разряд ЗН0 — в подрешетках (01)₂ и (00)₂. При уровне сигнала Лог.0 в разряде «Признак подрешетки» (ПП) рассматриваемые операции производятся в подрешетках с четными номерами, а единице — с нечетными.

При возмущениях фазового распределения значение кода (ΔФ4...ΔФ0) складывается с текущим значением кода фазового сдвига, хранящимся в регистре Рг2. Мультиплексор МХ2 отключается в подрешетках, не предусматривающих изменения фаз сигналов. Такое отключение обеспечивается схемой управления мультиплексорами (СУМ).

В режиме возмущения фазовых сдвигов результат выполнения операций заносится в выходной регистр Рг3 без дополнительной команды перезаписи. При приеме кода (КФ4...КФ0) его значение записывается в Рг2, а перезаписывается в Рг3 при анализе состояний 7-, 9-, 15-го разрядов и уровня выходного сигнала СС. Одновибраторы ОВ1 и ОВ2 запускаются ФСЗ практически одновременно. Данные поступают в Рг2, а затем содержимое этого регистра переписывается в Рг3, так как задержка сигналов в эти регистры по ЛЗ1 и ЛЗ2 выбирается разной (задержка, задаваемая ЛЗ2, должна быть больше задержки ЛЗ1 на суммарное время прохождения данных с выходной шины Рг2 через мультиплексор МХ3 и записи в Рг2).

Переложение на приемник кода простых операций управляющей микроЭВМ повышает скорость выполнения алгоритма строчно-столбцового суммирования [1] в четыре раза, по координатного градиентного спуска [2] — в два раза. В алгоритме слежения за источником сигнала число пересылок кодов фазовых сдвигов при выполнении возмущений фазового распределения уменьшается пропорционально числу элементов решетки.

Данный приемник может быть легко модифицирован при изменении соотношений между числом управляемых модулей ФАР и числом разрядов используемых фазовращателей подобно приемнику кодов [3].

103498, Москва, К-498, МИЭТ, кафедра микроразличных радиотехнических устройств и систем; тел. 534-09-08

ЛИТЕРАТУРА

1. Самойленко В. И., Шишов Ю. А. Управление фазированными антенными решетками. — М.: Радио и связь, 1983.
2. Джиган В. И., Незлин Д. В. Градиентные алгоритмы в задачах дискретной фазовой адаптации антенных решеток. — М., 1989. — 1 с. — Деп. в ВИНТИ 23.01.89, № 531—В89.
3. Джиган В. И. Линия связи управляющей микроЭВМ с модулями антенной решетки // Микропроцессорные средства и системы. — 1989. — № 3.

Статья поступила 1.04.88

УДК 681.325.63:681.3.025.06

А. Б. Китнер, Л. Д. Зорина

ЭФФЕКТИВНАЯ ПРОГРАММА ЗНАКОГЕНЕРАТОРА ДЛЯ МИКРОПРОЦЕССОРА K1810BM86

В растровых печатающих устройствах (ПУ) с построчным (телевизионным) методом формирования изображения знаки обычно генерируются из элементов точечной матрицы, хранящейся в ПЗУ знакогенератора. Алфавитно-цифровая информация из ЭВМ передается в буфер ПУ, и каждая текстовая строка преобразуется в последовательность растровых строк, которые, в свою очередь, передаются в механизм печати.

Первая растровая строка формируется из первых элементов точечных матриц всех знаков текстовой строки, вторая — из вторых и так далее до последней растровой строки. В итоге к каждому находящемуся в буфере ПУ знаку приходится обращаться столько раз, сколько растровых элементов (строк) содержит точечная матрица.

Время преобразования знаков из машинного представления (коды КОИ-7, КОИ-8, ДКОИ) в растровую форму существенно влияет на основную параметр ПУ — быстродействие. Поэтому знакогенераторы чаще всего реализуют аппаратно. Однако с введением МП в структуру контроллеров ПУ появилась возможность программной реализации знакогенератора, предпочтительную которую можно лишь в том случае, если не снижается быстродействие ПУ.

Ниже приведена подпрограмма формирования одной растровой строки для МП K1810BM86. Высокая скорость преобразования получена за счет нестандартной организации данных в ПЗУ, позволяющей 8-разрядным относительным адресом обращаться к любому элементу точечной матрицы, а значит, применять биструю команду XLAT. При такой организации (см. рисунок) одноименные элементы точечных матриц всех знаков (байты, пронумерованные 0...255) сгруппированы вместе, начиная с базового адреса V_i , где i — номер растровой строки.

Перед началом преобразования 8-битный код знака пересылается из буфера ПУ в аккумулятор AL и используется как индекс таблицы, адресуемой регистром BX, где находится базовый адрес V_i . Никаких промежуточных вычислений адреса уже не требуется. Число символов в строке находится в регистре CX, текущий адрес растрового

буфера — в DI, текущий адрес знака — в SI.

Здесь не приводится «обрамление», которое включает как перезагрузку регистров общего назначения перед формированием очередной растровой строки, так и подпрограммы управления узлами ПУ.

RASTER: LODSB

XLAT
MOV [DI], AL
INC DI
LOOP RASTER

Пересылка знака из буфера ПУ в аккумулятор AL, увеличение SI на единицу
Преобразование кодов
Запись раstra из AL в ОЗУ по адресу DI
Переадресация растрового буфера
Повторение цикла

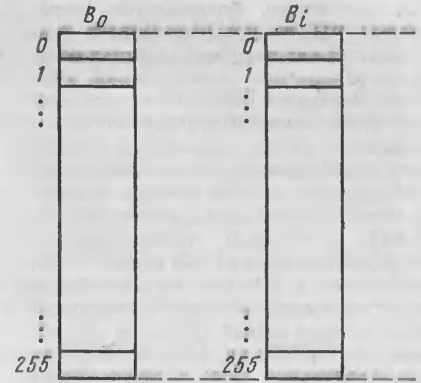
Время формирования одного знака в растровой строке равно 62 тактам, а строки из 132 знаков — 8179 тактам. При длительности такта 0,2 мкс это составит 1,64 мс. В расчете учтено время выборки из памяти первой команды LODSB при работе цикла, а также задержка в смене сегментных регистров в команде XLAT.

УДК 681.3.511

Б. К. Васильев, В. Г. Войнов

ИСПОЛЬЗОВАНИЕ ЯЗЫКОВ РАЗЛИЧНОГО УРОВНЯ В ЗАДАЧАХ ФИЛЬТРАЦИИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Язык программирования (ЯП), как и всякий язык, имеет свои законы развития, отличающиеся от законов естественных языков тем, что в них велика роль произвола создателей языка и компиляторов и почти незаметна роль «носителей» языка, т. е. пользователей. Лингвистический подход к ЯП приводит к необходимости расширения его синтаксиса, семантики и прагматики для реализации вновь возникающих задач [1]. Подразумевается, что языковая среда должна быть однородной. По нашему мнению, наиболее предпочтительно сочетание языков различного уровня в пределах одной программы. Использование всего арсенала уровней от надстроек для структурного программирования в ЯП до машинных кодов позволяет создать программу, оптимальную по трем измерениям: требуемой памяти, скорости выполнения, времени разработки (см. рисунок).



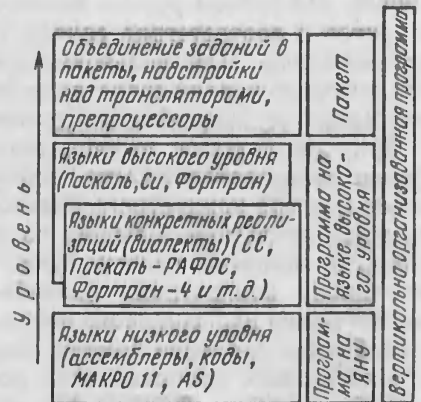
Организация данных в ПЗУ знакогенератора

Если элемент точечной матрицы состоит из более чем одного байта, то и в этом случае данные в ПЗУ

можно организовать с применением команды XLAT без промежуточных вычислений адреса. Таким образом в растровых ПУ во время перевода строки можно успеть сгенерировать очередную строку программным путем, сэкономив тем самым аппаратуру.

Телефон 485-92-69, 485-94-43, Киев

Статья поступила 15.03.88



Вертикальная организация в программировании

К сожалению, большинство современных ЯП не предусматривает возможности для подобной «вертикальной организации» программирования. «Чистый» Фортран и БЕЙСИК «без особых излишеств» лишили программиста возможности сократить время или сэкономить память, используя особенности ассемблера. Все же есть примеры удачного сопряжения языков высокого уровня (ЯВУ) с языками низкого уровня (ЯНУ): Фортран-2 в PS/8 (для PDP-8), допускающий вставки на ассемблере непосредственно в текст и имеющий обязательный второй проход на ассемблере [2]; система QUASIC-2, обладающая свойствами ЯВУ и включающая набор операций, являющихся прямым взаимодействием ассемблерных команд, имеющая программный интерфейс к аппаратным ресурсам [3]; Модуль-2 [4]; Паскаль-РАФОС, допускающий вставки на ассемблере в текст программы [5].

Сравним результаты вертикального программирования на Паскале-РАФОСе и на Паскале. Простая операция свертки в вещественных числах для массива в 1000 точек (операция свертки с гауссовой кривой) выполняется примерно за 4 мин на СМ4 с FIS-кодами при достаточно оптимальном алгоритме: заранее вычисле-

ны матрица ядра и простейшая квадратурная формула. Аналогичный результат сглаживания получается на ДВКЗ с использованием Паскаля-РАФОСа менее чем за 1 с. Столь значительный результат достигнут благодаря тому, что вычисления проводятся в целочисленной 32-разрядной арифметике, в качестве ядра используется 17-точечный фильтр Савицкого-Голэя [6, 7], а для хранения промежуточных результатов свертки применяется циклический буфер. Для обработки начального и конечного участков массива предусмотрен 5-точечный фильтр.

Приведенные результаты демонстрируют преимущества сочетания ЯВУ и ЯНУ в программе; при этом появляется возможность работы в реальном масштабе времени с минимальными требованиями к ОЗУ. Естественным логическим продолжением, еще не осуществленным авторами, является аппаратная реализация встроенного микропроцессорного устройства вычисления свертки, включающего микропроцессор, ПЗУ для команд и ядра уравнения, ОЗУ для хранения массива текущих данных.

690600, Владивосток, ул. Октябрьская, 27, НИФТИ при ДВГУ, Васильеву Б. К.; тел. 577-19.

1. Наумов Н. А. О некоторых подходах к расширению языков программирования // В сб.: Проблемы информационных систем. — М.: МЦНТИ. — 1988. — Вып. 4. — С. 24—39.
2. PS/8. Reference Manual. PDP-8 handbook series. V. 1. Introduction to programming. Ch. 9. — P. 1—174.
3. Подольский Л. И., Лясковский А. П. // Микропроцессорные средства и системы. — 1987. — № 2. — С. 9—11.
4. Tagg A. G. // Software-practice and experience. — 1988. — N 18. — P. 495—508.
5. ПАСКАЛЬ-2/РАФОС. Программное обеспечение СМ ЭВМ. ОС для организации высокопроизводительных систем с разделением функций РАФОС-ПЛЮС, 1985.
6. Savitsky A., Golay M. J. E. // Anal. Chem. — 1964. — V. 36. — 18. — P. 1627—1639.
7. Białkowski S. E. // Anal. Chem. — 1988. — V. 60. — P. 355—361 A; 403—413 A.

Сообщение поступило 21.03.89

УДК 681.3

А. А. Мостовский

МОНИТОР УПРАВЛЕНИЯ ЗАДАЧАМИ ПОЛЬЗОВАТЕЛЯ С ДВУХУРОВНЕВОЙ СИСТЕМОЙ ПАРОЛИРОВАНИЯ

Монитор управления задачами (МУЗ) предназначен для интерактивного диалога пользователей с многотерминальной вычислительной системой на уровне режимов меню и реализован в ОС ДИАМС (версии 3.0 и выше). Предполагаемая среда использования МУЗ — машины серий СМ4, СМ1420 и совместимые с ними.

Система паролирования делится на два уровня. На первом уровне конкретному паролю ставится в соответствие список задач данного пользователя. Для пользователей, решающих аналогичную задачу средствами МУЗ, этот список можно дублировать на другие пароли.

При доступности вычислительной системы группе пользователей часто необходимо ограничение доступа конкретного пользователя к данным внутри задачи. Поэтому на втором уровне (паролирование по данным МУЗ) каждой задаче пароля конкретному пользователю назначается номер разрешенного подразделения данных. Пустое значение номера указывает на доступность всех подразделений данных из этой задачи пароля. Доступность подразделения проверяется из прикладной задачи обращением к подпрограмме монитора с передачей ей номера введенного подразделения.

Пользователь работает с двухуровневым меню. Верхний уровень задач доступен данному пользователю (паролю). Указав номер задачи, пользователь получает возможность доступа в меню ее режимов, откуда запускается рабочая программа. Каждое вхождение в систему МУЗ конкретного пользователя и окончание работы фиксируется, поэтому можно получить статистику за месяц. Кроме того, МУЗ обрабатывает ошибки программ и ОС и учитывает их. Функции работы с паролями пользователей (копирование, удаление, переименование) переданы администратору системы МУЗ, в то же время его пароль, задаваемый при генерации, можно обрабатывать, как любой другой.

Ввод и корректировка задач, режимов и подразделений паролей реализованы таким образом, чтобы пользователь-администратор МУЗ достаточно просто и наглядно вводил и корректировал необходимые данные в режиме экранного редактирования. Число требуемых данных минимизировано: пароль, номер задачи и ее имя, номер режима и его имя, имя программы и раздел ее местонахождения. Состав МУЗ: семь базовых многофункциональных программ и 17 сопрограмм; объем оперативной памяти обслуживающих программ монитора — 5...8 К/слов, ОС — ДИАМС УЗ.0/3.3.

Тел. 53-73-63, Таллинн

Сообщение поступило 26.05.89

РАБОЧЕЕ МЕСТО УЧЕНИКА НА БАЗЕ МИКРОЭВМ С КАССЕТНЫМ НАКОПИТЕЛЕМ

Один из путей повышения эффективности использования вычислительной техники для обучения — совершенствование существующих образцов микроЭВМ. В Центре информатики и вычислительной техники Казанского Государственного университета разработано рабочее место ученика (РМУ) на базе микроЭВМ «Электроника БК-0010» и кассетного накопителя на магнитной ленте (КНМЛ) с автоматическим управлением. В состав РМУ входят также контроллер КНМЛ, модуль внешнего ОЗУ (8 Кбайт), телевизионный монитор и печатающее устройство. Модуль внешнего ОЗУ реализован на микросхемах серий КР588, КР537, КНМЛ — на К155, К555. Программы управления кассетным накопителем и каталог файлов размещаются во внешней памяти.

КНМЛ подключается к разъему УП микроЭВМ и может работать в автоматическом и ручном режимах. В предлагаемом варианте использован доработанный бытовой магнитофон «МАЯК-233». В этом случае затраты (стоимость магнитофона и деталей) на одно рабочее место составляют около 450 руб.

Работа с КНМЛ подобна работе с магнитным диском. При записи на ленту каждой программе присваивается имя и номер. Эта информация заносится в каталог, который пишется в начало ленты, а перед работой автоматически перегружается во внешнее ОЗУ. При записи и чтении файлов контроллер КНМЛ анализирует временные метки, поступающие с датчика оборотов, связанного с подкассетником. На самой ленте файлы никак не помечаются, поэтому надежность работы КНМЛ не зависит от ее качества.

Файлы можно выбирать в произвольном порядке на повышенной скорости в режиме «Перемотка». Аналогично программам на ленту записываются звуковые фрагменты.

На базе РМУ создан класс учебной вычислительной техники, включающий в себя: 24 стандартные микроЭВМ «Электроника БК-0010», радиально подключенные через пульт управления, содержащие коммутационные элементы и усилитель мощности (УМ). К магнитофонному каналу каждой машины (вход и выход объединены) подходит коаксиальный кабель. При этом обеспечивается двусторонняя передача информации. В каждом сеансе обмена РМП может принимать информацию только от одной микроЭВМ, а передавать через УМ в одну или несколько машин одновременно (УМ рассчитан на подключение не менее 32 машин при длине кабелей 15 м).

420008, Казань, ул. Ленина, 18, ЦИВТ;
тел. 39-80-04

ОСНАЩЕНИЕ ПЭВМ ЕС1840 АРИФМЕТИЧЕСКИМ СОПРОЦЕССОРОМ

Многие современные прикладные программные пакеты требуют наличия в системе арифметического сопроцессора. Как известно, микропроцессор i8088/i8086 позволяет иметь в системе арифметический сопроцессор i8087, который дает возможность значительно увеличить вычислительную мощность при выполнении операций как с плавающей запятой, так и повышенной точности [1, 2]. Время выполнения (в микросекундах) некоторых операций в системе с сопроцессором и без него приведено в табл. 1.

Таблица 1

Операция	Процессор i8086	Сопроцессор i8087
Сложение (вычитание)	1 600	17
Умножение	2 100	27
Деление	3 200	39
Сдвиг	1 300	9
Вычисление:		
квадратного корня	19 600	36
тангенса	13 000	90
экспоненты	17 000	100

Таким образом, ПЭВМ ЕС1840, оснащенная сопроцессором, представляется в новом качестве: из интеллектуальной пишущей машинки превращается в ПЭВМ с достаточно большой мощностью.

Подключение арифметического сопроцессора осложняется отсутствием в ЕС1840 соответствующих цепей, а печатные платы изготовлены по восьмислойной технологии, что практически исключает их переделку. Поэтому процессор, сопроцессор и необходимое число других компонентов установлены на дополнительной субплате. Соответствующие выводы процессора и сопроцессора соединены локальной шиной для обмена информацией и синхронизации работы. Соединение выводов процессора и сопроцессора представлено в табл. 2.

Таблица 2

i8086	i8087	i8086	i8087
AD0...AD15	AD0...AD15	BUSY	BUSY
A16...A19	A16...A19	QS0, QS1	QS0, QS1
CLK	CLK	S0...S2	S0...S2
RESET	RESET	RQ/GT1	RQ/GT0
READY	READY	BHE	BHE

Большинство прикладных пакетов, использующих сопроцессор, при возникновении деления на ноль производят программную обработку прерывания. В случае отсутствия в пакете такой внутрипрограммной обработки предусмотрена системная обработка прерывания. Для этого выходной сигнал INT сопроцессора через инвертор подается на вход схемы И (К555ЛА3) N12 на системной плате и далее стробируется триггером на вход NMI процессора.

Опробование всех доступных прикладных программных пакетов, требующих обработки чисел с плавающей запятой на ЕС1840 с арифметическим сопроцессором, дала положительный результат [3].

Телефон 196-76-32, Москва

ЛИТЕРАТУРА

- Бредли Д. Программирование на языке ассемблера для персональных ЭВМ фирмы IBM: Пер. с англ.— М.: Радио и связь, 1988.
- Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088: Пер. с англ.— М.: Радио и связь, 1987.
- Сурков Е. Л., Кузнецов Г. И. Тензор жесткости ядер и гигантский монополюсный резонанс // Ядерная физика.— 1989. Т. 49.— С. 393.

АССОЦИАЦИЯ «СТРОЙТЕХИНФОРМАТИКА» — ЭТО ГАРАНТ ВАШЕГО ПРОЦВЕТАНИЯ!

Ассоциация на коммерческой основе поставит «под ключ» уникальные высоконадежные автоматизированные технологические линии (АТЛ) и комплексы (АТК) по проектированию, координации, нормированию, подготовке производства, планированию, регулированию, учету, отчетности и анализу всех видов технологических процессов и уровней управления в строительстве.

Наша продукция: локальные сети на базе компьютеров РС/ХТ, РС/АТ; комплекты базового и специального программного обеспечения зарубежного и отечественного производства; пакеты методической и общегосударственной нормативной документации, услуги по настройке и вводу информационной технологии в условиях конкретного производства.

На всех автоматизированных местах предлагаемых линий специалисты могут работать автономно или в единой технологической цепочке.

АТЛ «Проект» — проектно-технологическая документация принципиально нового типа, повышающая уровень автоматизации проектных работ до 80—85 % и совмещающая в компактном и удобном для работы виде всю необходимую ресурсную и экономическую информацию.

АТЛ «План» — это высокое качество планирования, организация эффективной системы управления, стабильный темп выполнения работ и своевременный ввод объектов в эксплуатацию. Экономический эффект от внедрения линии — 200 тыс. руб. на 100 млн. руб./г., сокращение строительного этапа инвестиционного цикла в 2,2—2,8 раза.

АТЛ «Комплект» — оптимальная система управления материально-техническим обеспечением и производственно-технологической комплектацией (экономия — 250—350 тыс. руб. на 100 млн. руб.). «Комплект» включает в себя информацию о поставщиках; потребителях; транспортных средствах; мощностях технологических линий предприятий стройиндустрии и вспомогательных производств УПТК; государственные; ведомственные и производственные нормы и нормативы, необходимые для планирования и управления материально-техническим обеспечением объектов.

АТЛ «Учет» обеспечивает непрерывный учет выполненных работ и произведенных затрат, комплексный анализ и формирование отчетности, оперативный контроль за ходом выполнения договорных обязательств, оптимальные сроки оборачиваемости финансовых средств,

высокую точность финансовой и бухгалтерской документации (время обработки информации по данной линии сокращается в 10 раз по сравнению с традиционными методами).

Методические основы функционирования всех предлагаемых потребителям АТЛ заложены в комплекте документов **Методология**.

Автоматизированные технологии в строительстве и стройиндустрии предлагаются потребителям в виде АТК «Бетон», «ТВО», «Лаборатория».

«Бетон» — это высокое качество смесей при минимальном расходе вяжущих (автоматический режим всех технологических процессов на бетонно-растворном узле).

«ТВО» полностью исключает технологические потери энергоресурсов, автоматизируется управление процессом тепловлажностной обработки строительных изделий и конструкций.

«Лаборатория» — незаменимый инструмент технологов, реально связывающий науку с производством, базовая часть единой системы контроля и управления качеством, дисциплинирующая поставщиков сырья и стабилизирующая технологические процессы производства ЖБИ.

Высококвалифицированные специалисты Ассоциации подберут оптимальный состав технических средств для условий конкретного производства, установят программные средства, подготовят нормативную базу и эксплуатационную документацию, обучат в кратчайшие сроки специалистов профессиональной работе на АТЛ и АТК.

Продукция Ассоциации «Стройтехинформатика» поможет любой строительной организации стать процветающей проектно-строительной фирмой с мировым уровнем управления производством.

103051, Москва, Неглинная ул., 17, стр. 1, Информационно - коммерческий центр, тел. 200-43-07, 200-63-40.

Универсальные платы для РС АТ/ХТ/PS2

Значительно расширить возможности Вашего компьютера позволят универсальные макетные печатные платы с разъемами в стандарте IBM PC, производимые Центром НТТМ «Эврика».

Семь типов плат позволяют разместить от 40 до 120 корпусов микросхем типа DIP. По желанию заказчика устанавливаются схемы сопряжения с шиной и дешифратора адреса, конденсаторы развязки.

Максимальный срок выполнения заказа — два месяца; цена от 180 до 350 руб.

Принимаются также заказы на разработку фотошаблонов печатных плат для любых устройств по электрическим схемам и изготовление печатных плат по современной зарубежной технологии.

109387, Москва, ул. Ейская, д. 5/9, Центр НТТМ «Эврика»
Телекс: 411700 PTB SU EVRIKA, факс 200-22-17 EVRIKA
Телефон: 350-48-83, Шарунин П. А.

УДК 681.3

Вишняков В. А., Папков А. С. **Архитектура процессора логического вывода для 16-разрядной ПЭВМ // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 2.

Предложена структура аппаратно-программных средств для поддержки пролог-системы на 16-разрядных ПЭВМ. Рассмотрены возможности подключения пролог-процессора к системной шине ПЭВМ.

УДК 681.325

Банников П. Д., Прибыльский А. В., Сякерский В. С., Черноусова Т. Г., Шух Н. И. **Устройство последовательного обмена на основе универсального асинхронного приемопередатчика. // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 7.

Описана структура и приведены параметры мало потребляющего устройства, содержащего восемь последовательных каналов ИРПС, с использованием БИС универсального асинхронного приемопередатчика КР1818ВА19 и БИС серии К588.

УДК 681.007.57

Галуев Г. А. **Программируемая нейронно-конвейерная SIMD-архитектура для параллельной обработки изображений // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 11.

Анализируются принципы организации и функционирования зрительного анализатора и предлагается программируемая нейронно-конвейерная архитектура нейронно-сетевой СБИС-системы для параллельной обработки изображений.

УДК 581.3:519.85

Кацман В. Е., Малинов В. Г. **Пакет линейного программирования ЛП-МИКРО-2 для персональных ЭВМ // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 14.

Рассматривается пакет программ для решения задач линейного программирования на персональной ЭВМ «Искра 1030». Приводятся результаты численных экспериментов.

УДК 681.325

Коробченко В. В., Дианов А. П., Щелкунов Н. Н. **Программное обеспечение для программаторов ПЛМ // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 16.

Рассмотрена организация программного обеспечения для автоматизации процесса программирования ПЛМ. Для системы проектирования МПСП предложен двухпрограммный пакет программирования БИС КР556РТ1, КР556РТ2, который ориентирован для работы в ОС типа ДОС1800.

УДК 681.3.06

Алексеевский М. А., Ельник И. А., Розенштейн Е. И. **Операционная система реального времени ДС-86 // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 36.

Излагаются подход и принципы построения ПЗУ-ориентированной исполнительной операционной системы реального времени ДС-86, предназначенной для управления процессами обработки и обмена данными на микроЭВМ типа «Электроника 60».

УДК 681.3.06

Шмалько С. Г., Винников А. В. **Язык программирования FORTH-88R // Микропроцессорные средства и системы.**— 1990.— № 3.— С. 39.

Предложена модификация стандарта языка FORTH, предусматривающая изменение структуры словарной статьи, текстового интерпретатора, буферного пула, организацию вычислительной арифметики.

UDC 681.3

Vishnyakov V. A., Papkov A. S. **The logical 1/0 processor architecture for the 16-bit personal computer. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 2.

The structure of the hardware/software support of the PROLOG system for 16-bit personal computers is proposed. The possibility of PROLOG-processor connection to PC system bus is discussed.

UDC 681.325

Bannikov P.D., Prybylsky A. V., Syaker-sky V. S., Chernousova T. G., Shukh N. I. **Serial communication port using КР1818ВА19 UART. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 7.

The structure and parameters of the low-power consuming eight-channel serial communication adapter is described. The device is built using LSI UART unit type КР1818ВА19 and CMOS LSI of the K588 family.

UDC 681.007.57

Galuyev G. A. **A programmable neuron—conwayer SIMD architecture for parallel image processing. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 11.

The principles of organization and operation of image analyzer are discussed. The programmable neuron-conwayer architecture of the neuron—net VLSI system for parallel image processing is proposed.

UDC 681.3:519.85

Katzman V. E., Malinov V. G. **Linear programming software set MICRO-2 for personal computer. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 14.

The software toolbox for linear programming problems running on ISKRA 1030 IBM PC compatible computer is described. The results of numeric experiments are shown.

UDC 681.325

Korobchenko V. V., Dianov A. P., Shelkunov N. N. **Software tools or PAL programing. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 16.

The operation of the software for automation of PAL design and programming is described. The software package suitable for K556PT1 and K556PT2 PAL LSI programming in the microprocessor development system is proposed. The programs run under DOS1800 operating system.

UDC 681.3.06

Alekseevsky M. A., Elnik I. A., ROsenstein E. I. **A Real-time operating system DS-86. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 36.

The general principles and implementation of ROM-resident executive real-time operating system designed to support data processing and exchange on the "Electronika 60" (LSI-11) are described.

UDC 681.3.06

Shmalko S. G., Vinnikov A. V. **FORTH-88R programming language. // Microprocessor devices and systems.**— 1990.— N. 3.— P. 39.

The modification of the FORTH language standard is proposed, which implies new structure of the vocabulary entry, text interpreter, buffer pool, and calculator organization.

УДК 681.3

Осовецкий Л. Г., Штрик А. А. Анализ опыта внедрения крупных технологических комплексов создания программ и методов их выбора // Микропроцессорные средства и системы.— 1990 г.— № 3.— С. 43.

Рассмотрены вопросы внедрения крупных технологических систем автоматизации разработки программных средств. Обсуждаются различные стороны этого процесса на примере внедрения системы автоматизации САРПО РУЗА для разработки одного из конкретных программных проектов; подходы к осуществлению эффективного выбора технологических комплексов автоматизации и основные факторы, влияющие на этот выбор.

УДК 681.326.34

В. И. Джиган Управляющая микроЭВМ на основе микропроцессорного комплекса БИС серии К588 // Микропроцессорные средства и системы.— 1990.— № 3.— С. 46

Рассматривается микроЭВМ на основе МПК БИС серии К588, разработанная для решения задач управления фазированными антенными решетками. Она совместима с микроЭВМ семейства «Электроника». Максимальная мощность, потребляемая микроЭВМ, не превышает 2,5 Вт, а температурный диапазон ее функционирования ограничен значениями $-10...+70^{\circ}\text{C}$ и определяется используемыми БИС ПЗУ КМ558PP3 и АЦП К1108ПВ1

УДК 681.327.23

Коврига Д. С., Гладченко С. М. Микропроцессорная приставка для станков с ЧПУ на базе микроЭВМ «Электроника БК» // Микропроцессорные средства и системы.— 1990.— № 3.— С. 57.

В статье описан состав и характеристики программного обеспечения микропроцессорной приставки для станков с ЧПУ на базе микроЭВМ «Электроника БК-0010, БК-0011». Представлена принципиальная схема подключения приставки к станку с ЧПУ.

УДК 681.326

Седов А. Н., Молчанов О. Н. Устройство формирования шины типа 8080 // Микропроцессорные средства и системы.— 1990.— № 3.— С. 62.

Описано устройство формирования шины типа 8080 для персональных компьютеров, совместимых с IBM PC на основе быстродействующих комплектов серий К1804, К589. Рассмотрены особенности программного обеспечения.

УДК 681.327

Орестов Ю. А., Ивонтьев Н. Н. Четырехканальный аналого-цифровой преобразователь // Микропроцессорные средства и системы.— 1990.— № 3.— С. 64.

Рассмотрен одноплатный 4-канальный аналого-цифровой преобразователь на микросхемах К1108ПВ2А. Приведены технические характеристики, электрическая схема и краткое описание основных узлов.

UDC 681.3

Osovetsky L. G., Shtrik A. A. The analysis of large technologic systems for software creation and quideines for perfect choice. // Microprocessor devices and systems.— 1990.— N. 3.— P. 43.

The questions of large technologic software design support systems application are described. Various problems arising in this process are discussed on the practical example of RUZA programmers CAD application of one software project. The approaches for effective choice of technologic systems and main factors defining the perfect decision are shown.

UDC 681.325.34

Dzhigan V. I. Microprocessor controller using K588 CMOS LSI family. // Microprocessor devices and systems.— 1990.— N. 3.— P. 46.

The OEM microcomputer using K588 LSI family is designed for phased antenna array control. It is software-compatible with "Electronika" microcomputer family but its power consumption is less than 2.5W and the safe operation temperature range is $-10...+70^{\circ}\text{C}$. The temperature limitations are restricted by ROM КМ558PP3 and ADC К1108ПВ1.

UDC 681.327.3

Kovriga D. S., Gadchenko S. M. Microprocessor unit for digitally-controlled machine-tools using "Electronika-BK" home computer. // Microprocessor devices and systems.— 1990.— N. 3.— P. 57.

The configuration and main features of the software support for the microprocessor unit for machine-tools built using "Electronika-BK0010", BK0011 home computers are described. The circuit diagram showing unit connection to the machine-tool is included.

UDC 681.326

Sedov A. N., Molchanov O. N. The bus converter for IBM PC. // Microprocessor devices and systems.— 1990.— N. 3.— P. 62.

Converter card used for conversion of the IBM PC system bus into 8080 bus is described. The converter uses high-speed ICS of K1804 and K589 families. The software support is described.

UDC 681.327

Orestov Yu. A., Ivontiev N. N. Four-channel ADC. // Microprocessor devices and systems.— 1990.— N. 3.— P. 64.

The single-board four-channel ADC using K1108PW2A LSI is described. The technical specifications, circuit diagram and brief description are given.

Заместитель главного редактора
С. М. Пеленов

Номер подготовили:

Е. И. Бабич, Г. Г. Глушкова,
В. М. Ларионова, Е. П. Граве
Корректор Е. М. Кучерявенко
Технический редактор
Г. И. Колосова
Адрес редакции журнала:
103051, Москва, Малый
Сухаревский пер., д. 9А

Телефоны: 208-73-23, 208-19-94

Сдано в набор 01.11.89.

Подписано к печати 19.07.90.

Формат 84×108¹/₁₆

Офсетная печать Т-08989

Усл. печ. л. 10,08. Уч.-изд. л. 14,6.

Тираж 87.300

Заказ 6577 Цена 1 р. 10 к.

Орган Государственного
комитета СССР
по вычислительной технике
и информатике

Текст набран в Ордена

Трудового Красного Знамени

Чеховском полиграфическом

комбинате

Изготовлен в Московской типографии

№ 13

ПО «Периодика» ВО

«Союзполиграфпром»

Комитета СССР по печати

107005, Москва, Денисовский пер.,

дом 30. Заказ 209

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года



МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 4 / 1990 МОСКВА

СОДЕРЖАНИЕ

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА	Гальперин М. П., Блох Е. М., Шляхтин В. В.— Однокристалльные дисплейные контроллеры	2	
	Блох Е. М., Бодашков К. Б.— Однокристалльный контроллер дисплея и клавиатуры	5	
	Палташев Т. Т., Рахматулин О. А.— Графический сопроцессор для рабочих станций и ПЭВМ	10	
	Куценко А. А., Ковалевский В. К.— КМОП БИС КР1820ВР1 — периферийное устройство для однокристалльных ЭВМ	14	
	Руденко Ю. М.— Реализация плавающей арифметики и элементарных функций для микропроцессоров серий К1816, К1810, К580, К1821	16	
	Лопатин А. В., Позин Б. А.— Обращения к устройствам ввода-вывода средствами языка Паскаль М86	19	
	Лактионов Е. Ю.— Пакет программ БАЗА-85 для ПЭВМ	20	
	Домарацкий С. Н., Шраго И. Л.— Организация программной среды пользователя системы автоматизации исследований на базе ПЭВМ	21	
	Тищенко С. П., Якушин Ю. Е.— Электронный диск на базе КГД ДВК 3	25	
	Щелкунов Н. Н., Дианов А. П.— Организация программного обеспечения контроллеров ГМД	26	
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	Дедков А. Ф., Наумов Е. В., Щерс А. Л.— Средства работы с базами данных в языке Пролог	33	
	Операционные системы	Гаганов А. С., Анищенко В. П., Морозов С. А., Черкай А. Д.— Инструментальное программное обеспечение ОЭВМ КБ 1013	37
		Бальнич В. Н., Кройчик Г. В., Крол С. М., Палий О. Г., Фелезюк Р. В.— Семейство кросс-систем ПОМпа	39
	Прикладное программное обеспечение	Казменко Е. И., Заруднева С. В.— Средства форматного ввода информации в ЭВМ	42
		Брызгалова М. Г., Вельтмандер П. В.— Управляющая программа интеллектуального графического терминала «ГАММА-5»	45
		Дворский Э. Э., Черняховский А. Н., Лезник М. М.— Ввод изображений в микроЭВМ с помощью факсимильного аппарата «Штрих»	48
	Средства отладки	Нестеренко С. А., Кравцов В. А.— Микропроцессорная система проектирования и отладки микроконтроллеров	52
		Бедарев А. В., Гравов В. С.— Отладочный пульт для устройств на ОЭВМ	53
		Руденко Ю. М., Жиганов В. Г., Мосин А. Я.— Система тестирования и отладки ОЭВМ серии К1816	55
		Семенов В. С., Шипилов С. Г., Сафронов Е. А., Зиновейкин О. А.— Аппаратно-программный отладочный комплекс на базе контроллера «Электроника МС 2702-1»	57
Селицкий С. С., Сыркин М. Ю.— Комплект инструментальных средств отладки		59	
Иванова Л. В.— Информационно-поисковая система «Архив»		62	
Стрелков Н. В., Кокарев А. В.— Устройство защиты памяти микроЭВМ		65	
ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ	Щелкунов Н. Н., Дианов А. П.— Организация контроллеров ГМД	70	
	Буняк Ю. А.— Устройства сопряжения с шиной МПИ на основе БИС серии К1802	75	
	Герман Г. Б., Сачук А. А., Улыбин А. А.— Сопряжение микроЭВМ «Электроника 60» с микропроцессорными устройствами	77	
	Бири Ш., Ефремов А. А., Молнар И.— Интерфейс для управления и автоматического измерения на основе ПК типа IBM PC XT/AT	79	
	Усолкин А. А., Сороко Н. Н.— Программируемый интерфейс R 232C для микроЭВМ «Электроника 60»	80	
	Малахов Н. А., Пиляр Н. В.— Интерфейсы мышь и координатный шар к ПЭВМ типа IBM	83	
	Комиссаров Е. В., Кулинич П. А., Сидоркин В. В.— Интерфейс НМЛ СМ 5309 для ПК IBM PC	84	
	Бабкин П. А., Несмелов Е. Ю.— Многоканальное устройство прерываний для магистрали МПИ	86	
	Задеба А. А.— Команда RST в процедурах вызова подпрограмм	89	
	Владимирский М. М.— Тестирование ОЗУ с помощью псевдослучайной двоичной последовательности	91	
УЧЕБНЫЙ ЦЕНТР			

УДК 621.3.049.77:681.326.3

М. П. Гальперин, Е. М. Блох, В. В. Шляхтин

ОДНОКРИСТАЛЬНЫЕ ДИСПЛЕЙНЫЕ КОНТРОЛЛЕРЫ

Использование специализированных БИС дисплейных контроллеров значительно упрощает конструирование средств отображения информации, снижает нагрузку на центральный процессор (ЦП) системы, уменьшает затраты на создание программного обеспечения и повышает надежность аппаратуры. Цель этой статьи — обзор и сравнительный анализ однокристальных дисплейных контроллеров, выпускаемых отечественной промышленностью.

Первые специализированные БИС для систем отображения информации появились на мировом рынке в конце 70-х годов. За более чем 10-летний период развития подобных устройств можно выделить три поколения однокристальных дисплейных контроллеров.

Дисплейные контроллеры первого поколения решают задачу генерации изображения на экране электронно-лучевой трубки (ЭЛТ) на основе данных кадрового буфера памяти отображения. При этом формирование данных в кадровом буфере выполняется микропроцессором общего назначения.

Более сложные дисплейные контроллеры второго поколения содержат два основных блока: рисующий (или графический) процессор, генерирующий изображение в кадровом буфере, и дисплейный процессор воспроизводящий изображение на экране. Контроллеры этого поколения имеют развитую систему команд, программируемые форматы изображения, позволяют строить мощные терминальные системы для отображения алфавитно-цифровой и графической информации, но требуют значительных затрат времени ЦП на подготовку и загрузку данных.

В начале 1986 г. появились первые образцы специализированных графических БИС нового поколения — графических сопроцессоров. В новых БИС предусматриваются более высокие интеллектуальные возможности, усовершенствованные встроенные алгоритмы формирования графических образов, поддержка перспективных стандартов на графические интерфейсы, аппаратная реализация полиэкранного режима. Графические сопроцессоры решают свои задачи практически автономно, а ЦП выполняет общесистемные задачи.

Отечественная промышленность выпускает несколько БИС, относящихся к первому поколению дисплейных контроллеров: КР580ВГ75, К1809ВГ3 и КМ1809ВГ6, а также один дисплейный контроллер второго поколения — К1809ВГ4 (см. таблицу).

Дисплейный контроллер КР580ВГ75 — функциональный аналог микросхемы $i8275$, предназначен для конструирования экранных пультов с растровым сканированием на основе МПК БИС серии К580. Контроллер обеспечивает регенерацию изображения на экране на основе данных памяти отображения, которая входит в общее поле памяти ЦП. Для организации прямого доступа к памяти (ПДП) используется контроллер ПДП типа КР580 ВТ57. Дисплейный контроллер обеспечивает синхронизацию раstra маркера, обнаружение сигналов светового пера и предоставляет возможность сопряжения почти с любым индикатором на ЭЛТ с растровой разверткой при минимальных затратах внешних аппаратных и программных средств. Форматы экрана и символов являются программируемыми, что позволяет воспроизводить на экране до 64 строк по 80 символов в каждой, обеспечивая 1...16 телевизионных строк на одну строку символов. Контроллер имеет 11 программируемых визуальных атрибутов символов, хотя допускается не более 16 изменений атрибутов на одну строку. Дисплейный контроллер КР580ВГ75 не обеспечивает графический режим, однако возможна работа в режиме псевдографики. Более подробно микросхема КР580ВГ75 и особенности ее использования описаны в [1, 2].

Контроллер ЭЛТ (КЭЛТ) КМ1809ВГ6 (функциональный аналог микросхемы МС6845 фирмы Motorola [3]) также предназначен для управления растровыми дисплеями, но в отличие от КР580 ВГ75 может применяться в конструировании как алфавитно-цифровых, так и графических терминалов [4]. Контроллер обеспечивает оптимальное сочетание программных и аппаратных средств и в совокупности с распространяемыми микропроцессорами позволяет легко сконструировать плату управления терминалом. При этом функции управления клавиатурой, формирования образов в памяти отображения, управления курсором и редактирования выполняет терминальный микропроцессор, а КЭЛТ — функции регенерации изображения на экране и управления памятью отображения.

Важное преимущество КЭЛТ по сравнению

Характеристики дисплейных контроллеров

Параметр	КР580ВГ75	КМ1809ВГ6	К1809ВГ3	К1809ВГ4
1	2	3	4	5
Напряжение источника питания, В	5	5	5	5
Тактовая частота, МГц	2	2	5	5
Число контактов	40	40	48	40
Интерфейс растрового монитора:				
формирование видеосигналов	Нет	Нет	Есть	Нет
программирование синхронизации разверток и их типа	Есть	Есть	Нет	Есть
Интерфейс памяти отображения:				
шина	Общая с ЦП	Отделена от ЦП	Общая с ЦП	Отделена от ЦП
разрядность шины данных	8	8	16	16
объем адресуемой памяти, Кбайт	—	16	16	512
регенерация динамической памяти	Есть	Есть	Нет	Есть
Интерфейс светового пера	Есть	Есть	Нет	Есть
Интерфейс клавиатуры	Нет	Нет	До 256 клавиш	Нет
Контроллер ПДП	Нет	Нет	Есть	Нет
Символьный режим:				
число символов в строке	80	132	32,64	256
число строк	64	64	32	100
число атрибутов символа	11	—	5	5
программируемый курсор	Есть	Есть	Нет	Есть
встроенный знакогенератор	Нет	Нет	128	Нет
число линий раstra на ряд символов	16	32	16	32
Графический режим:				
разрешение, ЭО	Псевдо-графика	Есть	Есть	Есть
в одном цвете,	—	640×200	256×256	2048×2048
в 16 цветах	—	320×200	256×256	1024×1024
Совмещенный режим, полос	Нет	Нет	16	4

с двумя другими дисплейными контроллерами (КР580 ВГ75, К1809ВГ3) — двухпортовая

архитектура. Один порт предназначен для подключения к системной магистрали ЦП, а второй — для управления кадровым буфером. Такая структура позволяет вынести кадровый буфер из общего поля памяти ЦП, что особенно важно при работе микропроцессоров с ограниченным адресным пространством.

КЭЛТ имеет 18 внутренних регистров, программируемых через системную магистраль процессора. При этом обеспечивается:

плотность символьной информации на экране: 80×24, 72×64, 132×20 символов и др.

управление синхронизацией разверток и их типом (чересстрочная и прогрессивная);

аппаратная прокрутка изображения: постраничная, построчная, посимвольная;

управление форматом и мерцанием курсора, считывание положения курсора и светового пера.

В структуру КЭЛТ входят следующие программируемые блоки: синхронизирующие генераторы по горизонтали и вертикали, счетчик адресов памяти отображения, логическая схема маркера, регистр светового пера и управляющие схемы интерфейса с процессорной шиной данных. Микросхема синхронизируется тактовыми импульсами, поступающими от внешнего генератора.

Структура дисплейного контроллера позволяет применять его в алфавитно-цифровых, графических, монохромных и цветных дисплейных системах. При этом требуется соответствующая аппаратная поддержка: знакогенератор, схема формирования видеосигналов, регистры сдвига, дисплейная память необходимого объема и некоторые другие средства. КЭЛТ КМ1809ВГ6 обеспечивают функции отображения в персональных компьютерах ЕС1840, ЕС1841. При соблюдении стандартного аппаратного интерфейса не требуется дополнительных затрат на создание программного обеспечения, связанного с отображением.

Однокристалльный 16-разрядный дисплейный контроллер К1809ВГ3 предназначен для преобразования информации в памяти отображения в видеосигналы управления телевизионным монитором. В отличие от ранее описанных контроллеров он позволяет также управлять клавиатурой, содержащей до 256 клавиш. В составе микропроцессорной системы контроллер может использоваться для отображения алфавитно-цифровой и графической информации и формирования телеигрового изображения с возможностью перемещения графических блоков, а также для ввода данных с клавиатуры.

Микросхема состоит из трех независимых функциональных блоков: контроллеров отображения, канала и клавиатуры, связанных между собой общей внутренней магистралью, сигналами

лами синхронизации и управления. Режимы работы контроллера программируются пятью программно доступными регистрами.

Контроллер K1809BG3 позволяет отображать информацию в четырех различных режимах: символьном, графическом, совмещенном и слайдовом. Режимы переключаются программно через регистр управления. Каждому режиму соответствует своя структура представления данных в памяти отображения. Для максимального разрешения в цветном графическом режиме (256×256 элементов отображения в восьми цветах) требуется 16 Кбайт памяти. В символьном режиме в этой памяти могут быть размещены четыре страницы текста в формате 32 строки по 64 символа. В отличие от других рассмотренных дисплейных контроллеров K1809BG3 имеет совмещенный режим, допускающий одновременное отображение символьной и графической информации, а также слайдовый режим, дающий возможность эффективно программировать компьютерные игры.

Контроллер может использоваться для конструирования алфавитно-цифровых и графических терминалов, реализации функций отображения и ввода с клавиатуры персональных компьютеров, создания специализированных игровых (шахматных) автоматов. Эффективность использования этой микросхемы определяется тем, что в ее состав входят контроллеры ПДП и клавиатуры, встроенный знакогенератор, блок формирования видеосигналов, что обеспечивает значительную экономию аппаратных средств по сравнению с двумя другими рассмотренными дисплейными контроллерами.

К недостаткам K1809BG3 можно отнести ограниченное число форматов в символьном режиме, отсутствие схемы регенерации динамической памяти. Кроме этого, работа контроллера с кадровым буфером через общую магистраль с ЦП замедляет работу процессора, и часть адресного пространства ЦП выделяется под кадровый буфер.

Дисплейный контроллер K1809BG3 — оригинальная микросхема, не имеющая аналогов, поэтому при ее использовании требуется разработка специального программного обеспечения.

Графический дисплейный контроллер (ГДК) K1809BG4 (функциональный аналог микросхемы μ PD 7220 фирмы NEC) относится ко второму поколению дисплейных контроллеров и представляет собой интеллектуальный периферийный микропроцессор для работы в алфавитно-цифровых и графических терминалах [5]. ГДК располагается между памятью отображения и ЦП и выполняет функции генерации растровых изображений и управления памятью отображения объемом до 512 Кбайт.

Контроллер может использоваться в символьном, графическом и смешанном режимах.

В символьном режиме работы слово памяти отображения содержит код алфавитно-цифрового символа и атрибуты. Возможно отображение до 100 строк по 256 символов, каждая строка содержит до 32 строк телевизионной развертки. При использовании стандартного формата 80×24 символов память отображения позволяет хранить 34 текстовых кадра. В символьном режиме обеспечивается управление перемещением и атрибутами курсора. Изображение на экране можно запрограммировать в виде независимых областей-полос (до четырех). Каждая область определяется стартовым адресом памяти отображения и числом строк.

В графическом режиме контроллер позволяет иметь разрешение 2048×2048 элементов отображения (ЭО) монохромного изображения с кодированием ЭО одним битом. При разрешении 1024×1024 ЭО контроллер обеспечивает изображение в 16 цветах с кодированием четырьмя битами. В графическом режиме экран можно разделить на независимые верхнюю и нижнюю области. В смешанном режиме ГДК формирует от одной до четырех областей изображения. Области программируются независимо в графическом или символьном режиме.

Рисующий процессор ГДК обеспечивает аппаратную рисовку линий, дуг, прямоугольников, графических символов. Данные памяти отображения обрабатываются за один цикл «чтение-модификация-запись», причем операции модификации содержат установку, очистку, инвертирование и замену по образцу.

Память отображения изолирована от ЦП; контроллер освобождает его от создания графических образов. Задача микропроцессора — формирование последовательности команд для ГДК и запись их в буферное ЗУ контроллера. Контроллер имеет 18 команд, которые можно разделить на пять групп управления: изображение, экраном, вычерчиванием, прямым доступом к памяти и чтением данных.

ГДК обеспечивает гибкое управление форматами изображения, типом развертки, аппаратное увеличение изображения с коэффициентом до 16, панорамирование изображения, выдает независимые сигналы горизонтальной и вертикальной синхронизации. Скорость вычерчивания графических образов составляет не менее 800 нс на элемент изображения. Контроллер считывает данные из памяти отображения, положение курсора и адрес светового пера. При наличии контроллера ПДП он управляет прямым доступом к памяти отображения со стороны микропроцессора терминальной системы (скорость передачи данных 1,25 млн. байт/с).

Несколько ГДК достаточно легко синхронизируются для совместной работы в одной системе. Это позволяет увеличить скорость формирования графических образов за счет параллельной работы в различных областях памяти отображения. Отдельный ГДК может быть настроен на работу в алфавитно-цифровом режиме и независимой области памяти [6].

Недостатки контроллера: ограниченный набор команд, неудобный пользовательский интерфейс, медленная работа с цветным изображением, необходимость постоянного управления со стороны ЦП.

К сожалению, однокристалльные дисплейные контроллеры мало используются в серийно выпускаемых отечественной промышленностью терминалах и микроЭВМ. Так в ДВКЗ функции отображения реализуются на двух платах: микропроцессорном контроллере (алфавитно-цифровой терминал), и графическом адаптере, формирующем поточечное монохромное изображение. В ПЭВМ «Электроника МС0585» одноплатный видеоконтроллер выполнен на базе трех БИС на основе универсальных вентиляционных матриц. Одна матрица — видеогенератор, а две другие — процессор видеоданных. Последний предназначен для записи в память отображения вертикальной или горизонтальной строки точек (от 1 до 16) в соответствии с заданной функцией взаимодействия с содержимым памяти. Функции алфавитно-цифрового терминала, а также графические примитивы, реализуются программно. Несмотря на использование вентиляционных матриц, плата видеоконтроллера содержит 72 интегральные схемы. Использование однокристалльных дисплейных контроллеров в приведенных системах позволило бы упростить аппаратные средства и уменьшить стоимость системы.

В современных зарубежных персональных компьютерах широко используются графические БИС нового поколения. Наибольшее распространение получили графические сопроцессоры i82786, TMS 34010, Am 95C60. Сове-

ренный графический сопроцессор должен обладать следующими свойствами:

- высокой степенью автономности в работе;
- разрешением 4096×4096 ЭО;
- аппаратной реализацией многооконного режима, панорамированием, масштабированием;
- скоростью формирования изображения не менее 20 млн. ЭО в секунду;
- аппаратной реализацией функций пересылки блоков растровой информации;
- аппаратной реализацией быстрых алгоритмов основных графических примитивов;
- возможностью микропрограммной реализации требуемых функций;
- возможностью программной настройки для работы с распространенными графическими интерфейсами.

В условиях работы систем трех-мерной графики графический сопроцессор должен обеспечивать аппаратную интерполяцию линий по шести координатам (X, Y, Z и интенсивности цвета R, G, B) и аппаратное отсеечение невидимых поверхностей.

Телефон 291-67-41, 291-66-72, Ленинград

ЛИТЕРАТУРА

1. Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шарошин С. Г. Принципы организации и работы дисплеев на основе БИС КР580ВГ75 // Микропроцессорные средства и системы — 1985. — № 3. — С. 51—60.
2. Зеленко Г. В. Дисплей для бытовой персональной ЭВМ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 60—70.
3. МС6845 — CRT controller, Motorola Microprocessor data manual, 1981.
4. Петров П. Проектирование на системы, проектирование на ассемблер и экспериментирование с 6502 // Радио, телевизия, электроника. — 1986. — № 3. — С. 27—33.
5. Уайз Дж. Л., Шейнвальд Х. Дисплейный контроллер, упрощающий разработку сложных графических терминалов // Электроника. — 1981. — № 7. — С. 61—68.
6. Slade S. Multiple controller create high speed color graphics // Computer Design. — 1984. — V. 23, N 8. — P 137—144.

Статья поступила 8.06.89

УДК 681.326.3:681.327.2

Е. М. Блох, К. Б. Бодашков

ОДНОКРИСТАЛЛЬНЫЙ КОНТРОЛЛЕР ДИСПЛЕЯ И КЛАВИАТУРЫ

Микросхема К1809ВГЗ предназначена для создания устройств отображения на ЭЛТ и воспроизведения алфавитно-цифровой и графической информации на экране монохромного и цветного мониторов, ввода данных с клавиатуры. Перекрывая весь тракт формирования изображения от общей шины до видеосигналов, она заменяет несколько БИС серии КР580, в том числе контроллеры ЭЛТ

КР580ВГ75, прерываний КР580ВН59, прямого доступа КР580ВТ57, частично контроллер клавиатуры КР580ВГ79, микросхемы ПЗУ знакогенератора, а также ряд микросхем малой и средней степени интеграции, используемых для организации режима прямого доступа к памяти (ПДП), сдвигового регистра преобразования данных в видеосигнал, схемы управления атрибутами символов.

Структурная схема контроллера К1809ВГЗ приведена на рис. 1, наименование и назначение выводов показано в табл. 1.

Возможности отображения

Дисплейный контроллер (ДК) К1809ВГЗ имеет четыре основных режима отображения: алфавитно-цифровой, графический, совмещенный (алфавитно-цифровой и графический), слайдовый.

В алфавитно-цифровом, графическом и совмещенном режимах ДК при отображении последовательно опрашивает буфер кадра, расположенный в общем поле памяти. В алфавитно-цифровом

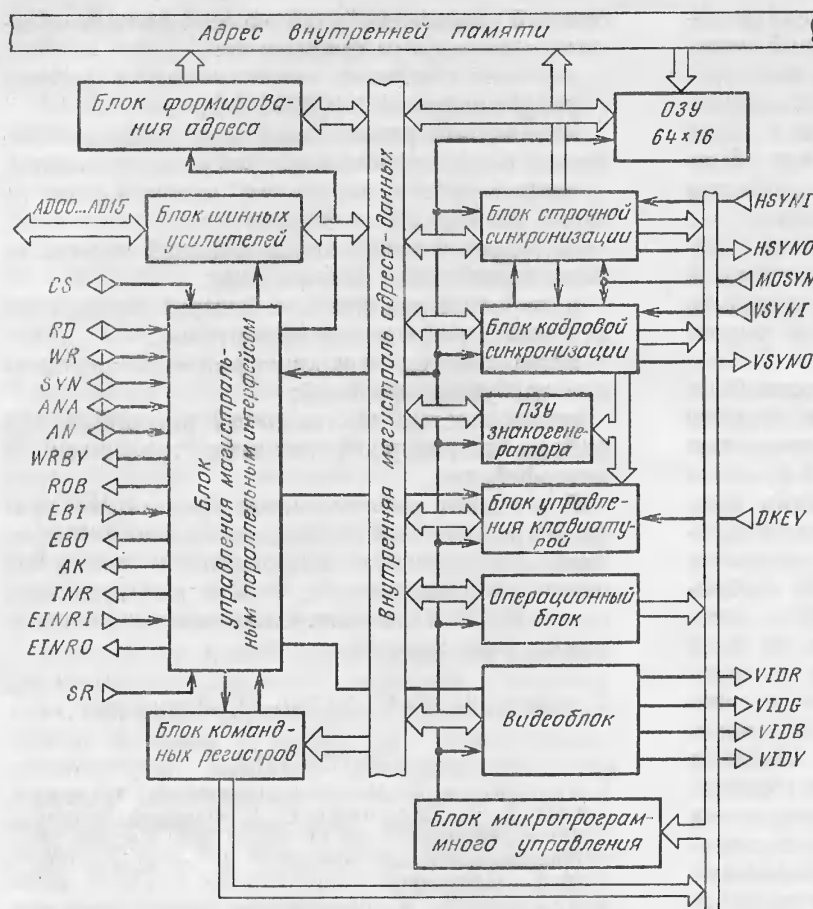


Рис. 1. Структурная схема БИС K1809BG3

точка изображения кодируется 2-разрядным кодом, что дает возможность представлять информацию в трех чистых цветах R, G, B. Первое битовое поле находится в младшем байте слова, второе — в старшем (табл. 3).

В режиме линейного пиксела в младшем байте слова размещаются данные о подсвете восьми линейно расположенных точек, разбитых на две группы (пиксели) по четыре точки в каждой, а в старшем байте — два 4-разрядных кода атрибутов пикселей, определяющих их цвет и мерцание (табл. 4).

Способ задания графической информации линейными пикселями при том же объеме памяти (2 бита на точку) дает возможность отображать информацию в восьми градациях цвета (или яркости) с заданием мерцания.

В режиме блоков используется пиксел размерами 4x4 точки с 4-разрядным кодированием одним цветом всех его точек. При таком способе формирования изображения при понижении разрешения до 64x64 точек цветовая палитра составляет 16 градаций цвета.

В совмещенном режиме алфавитно-цифровая и графическая информация отображается одновременно с разбиением экрана (по вертикали) на программно управляемые зоны символьной и графической информации (рис. 2).

Таблица 1

Вывод	Обозначение	Тип	Назначение	Примечание
1	2	3	4	5
25, 48	5 В	Вход	Напряжение источника питания	Максимальный ток потребления 230 мА
12, 35	0 В	»	Общий вывод	
24	BC	Выход	Напряжение подложки — 2 В	
27	C	Вход	Тактовая частота	5 МГц

Сигналы подключения к системной магистрали

№	Сигнал	Тип	Назначение	Примечание
1	RD	Вход-выход	Чтение данных	Вход — в режимах чтения или записи командных регистров
2	WP	»	Запись данных	
3	SYN	»	Синхронизация обмена	Выход — в режимах работы с кадровым буфером
4	ANA	»	Ответ по адресу	Вход — в режимах чтения или записи командных регистров
47	AN	»	Ответ устройства	Вход — в режимах работы с кадровым буфером
6		Вход	Выбор кристалла	
7...11	AD15...0	Вход-выход	Шина адреса-данных	
5	COB	Выход	Управление магистралью	
39	WRBY	»	Признак записи байта	
26	SP	Вход	Установка	Сброс командных регистров

режиме каждое слово буфера кадра содержит код знака по КОИ-7 (ГОСТ 13052-74, набор 2) или КОИ-8 (ГОСТ 19767-74), биты атрибутов цвета (R, G, B), мерцания (ME), инверсии подсвета (ИП), подчеркивания (ПДЧ) и признаки защиты памяти от редактирования (ЗП) (табл. 2).

Внутренний знакогенератор ДК генерирует символы в матрице 5x7 точек. Объем ПЗУ знакогенератора — 128 символов прописных букв русского и латинского алфавитов, цифр и знаков, а также 32 специальных символа. Число символов в строке 32 или 64 символа.

Число строк в кадре программируется методом задания числа линий развертки раstra на строку 9, 10, 12 или 16, что соответствует 32, 25, 24 или 16 строкам в кадре. При задании размерности точки по кадру в две линии раstra размеры символов по вертикали увеличиваются в два раза, при этом число строк в кадре изменяется на 16, 12, 12 и 8 соответственно.

В графическом режиме отображения используются следующие виды представления информации: битовой картой, линейными пикселями с разрешением 256x256, 256x128, и 128x128 точек и блоками с разрешением 64x64 точек.

В режиме битовой карты каждая

Продолжение табл. 1

Вывод	Обозначение	Тип	Назначение	Примечание
1	2	3	4	5
Сигналы управления ПДП				
40	EVI	Вход	Разрешение на захват магистрали	
41	AK	Выход	Подтверждение захвата магистрали	
42	RQB	Выход	Запрос магистрали	
44	EBO	»	Разрешение на захват магистрали	
Сигналы управления прерыванием от клавиатуры				
43	INR	Выход	Запрос на прерывание	
45	EINRI	Вход	Разрешение прерывания	
46	EINRO	Выход	Данные с клавиатуры	
38	DKEY	Вход	Данные с клавиатуры	
Сигналы управления видеомонитором				
28	VIDY	Выход	Видеосигнал дополнительный	
29	VIDB	»	Синий	
30	VIDG	»	Зеленый	
31	VIDR	Выход	Красный	
32	HSYNO	»	Строчная синхронизация или синхросмесь	$t=4,8 \text{ мкс}$
33	HSYNI	Вход	Внешняя строчная синхронизация	В режиме внутренней синхронизации соединять с 32
34	VSYNO	Выход	Кадровая синхронизация или синхросмесь	Синхросмесь задержана на 1,6 мкс
36	VSYNI	Вход	Внешняя кадровая синхронизация	При внутренней синхронизации соединить с шиной 5 В
37	MOSYN	»	Режим синхронизации	

Таблица 2

Байт	1								0							
Разряд	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Назначение	R	G	B	ME	ПДЧ	ИП	ЗП	ЗП	Код символа							

Таблица 3

Байт	1								0							
Разряд	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Точка	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1

Таблица 4

Байт	1								0							
Пиксел	2				1				2				1			
Разряд	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Точка	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
Назначение	R G B ME				R G B ME				Подсвет							



Рис. 2. Зоны отображения символической и графической информации

Максимальное число зон отображения — 16. В совмещенном режиме первые 16 ячеек буфера кадра используются для задания номера линии раstra, в которой необходимо переключить режим. Остальная часть буфера кадра разбивается на зоны, объем которых и тип отображаемой информации устанавливаются пользователем исходя из своих конкретных требований.

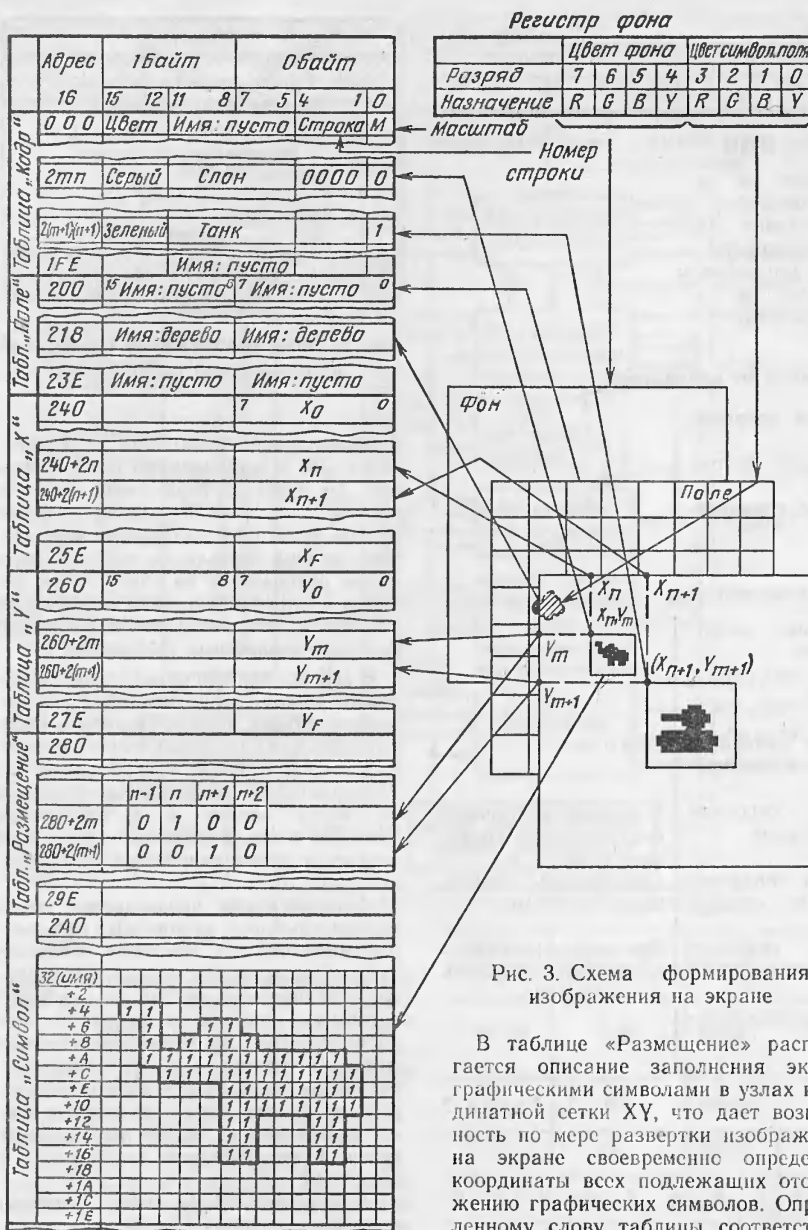
В ДК реализован специальный слайдовый режим, позволяющий в ограниченном объеме памяти создавать изображения для различных телеигр с плавным перемещением программируемых пользователем графических символов по всему экрану с дискретностью 256×256 точек. Изображение на экране создается путем наложения трех слайдов (рис. 3).

Нижний слайд представляет собой одноцветный фон, задаваемый из палитры, имеющей 16 цветовых оттенков; средний — поле, т. е. одноцветную картину, задаваемую из той же цветовой палитры и состоящую из равномерно расположенных по экрану графических символов (8 рядов по 8 символов); верхний — картину из программируемых по форме, цвету, размеру и месту положения на экране одноцветных графических символов из палитры в 16 цветов.

Полиграмма графических символов задается в матрице 16×16 точек с возможностью двукратного масштабирования. Два верхних слайда в точках, не занятых изображениями графических символов, прозрачны.

В слайдовом режиме описание изображения формируется центральным процессором (ЦП) в экранной памяти с использованием пяти таблиц: «Кадр», «X», «Y», «Размещение», «Символ».

В таблице «Кадр» размещается информация о графических символах, расположенных в узлах координатной сетки. Данные в ней располагаются последовательно по мере возрастания адресов в зависимости от порядкового номера координат: четыре младших разряда определяются номером координаты X, а четыре старших — номером координаты Y. Объем таблицы «Кадр» — 256 слов. В каждом слове таблицы задается цвет графического



символа, имя, номер строки символа, с которой начинается его отображение, а также бит масштаба, определяющий размер символа.

В таблице «Х» размещается упорядоченный по возрастанию кодов массив координат X, в таблице «Y» — координат Y графических символов (координаты отсчитываются от верхнего левого угла экрана).

Объем таблиц «Х» и «Y» — по 16 слов каждая. Это означает, что максимальное число отображаемых на экране графических символов с несовпадающими координатами также равно 16. При совпадении координат по оси X или Y число отображаемых графических символов увеличивается и в пределе может достигать 256, т. е. числа узлов в координатной сетке.

узле графического символа. Объем таблицы «Размещение» — 16 слов.

Полиграммы графических символов верхнего слайда программируются в матрице 16×16 точек (16 слов) и размещаются в общей для них таблице «Символ».

При экранном ЗУ 1К×16 в этой таблице можно расположить полиграммы 43 графических символов (для таблиц «Кадр», «Поле», «Х», «Y» и «Размещение» требуется в совокупности 336 ячеек памяти). С увеличением экранного ЗУ до 2К×16 число программируемых графических символов можно увеличить до 107, что вполне достаточно для синтеза довольно сложных изображений.

Интерфейс с процессором и памятью

Для сопряжения ДК с ЦП и памятью используется 16-разрядная магистраль типа МПИ. Дополнительно введен сигнал ответа по адресу АНА, что позволяет настроить микросхему К1809ВГ3 на максимальное быстродействие при передаче адреса на магистраль. Этот сигнал введен и в микропроцессоре К1801ВМ2.

ДК инициализируется в пассивном режиме загрузкой командных регистров (табл. 5) и подачей сигнала выборки кристалла CS с внешней схемы дешифрации адреса. Файл отображения из кадрового буфера, расположенного на общей магистрали, читается в режиме ПДП.

ДК допускает несколько вариантов работы в режиме ПДП, отличающихся интенсивностью обращения к общей памяти: монополюсно на прямой или обратный код по кадру, на строку раstra, а также на одиночные запросы.

Режим «монополюсно на обратный код по кадру» используется только при отделеении кадрового буфера от общей магистрали. При этом во время обратного хода по кадру дисплейный файл из общей памяти переписывается в кадровый буфер. Режимы работы ДК с памятью задаются регистром управления (PгУ) (табл. 6).

Гибкая регулировка режимов работы ДК с памятью дает возможность оптимально распределить вычислительные ресурсы разрабатываемой системы, а набор режимов запроса данных в алфавитно-цифровом режиме — настроить ДК на работу с магистралью в соот-

Таблица 5

Регистр	Адрес	Назначение	Команда
PгУ (управление)	0	Режим отображения и обмена на МПИ	WR
PгВек (вектора)	2	Номер вектора прерывания	WR
PгФ (фона)	4	Цвет фона и символов поля, форматы отображения	WR
PгА (установки адреса)	6	Зона памяти кадрового буфера	WR
PгКлв (клавиатуры)	0	Код клавиш	RD

Таблица 6

Разряд PpY	Назначение	Примечание
0	Включение экрана	0 — экран загашен, канал закрыт 1 — экран включен, канал открыт
6, 5	Режим ПДП	00 — монопольно на прямой ход по кадру 01 — одиночные запросы 10 — монопольно на строку раstra 11 — монопольно на обратный ход по кадру
4, 3	Режим отображения	00 — алфавитно-цифровой 01 — графический 10 — совмещенный 11 — слайдовый
2, 1	Режим запроса алфавитно-цифровой информации	00 — синхронно с отображением 01 — несинхронно, с гашением одной строки раstra 10 — несинхронно, с гашением двух строк раstra 11 — с буферизацией данных
7	Режим графики	0 — битовая 1 — пикселями

ветствии с форматом изображения алфавитно-цифровых данных. Внутренний буфер объемом 128 байт обеспечивает режим буферизации в течение времени отображения предыдущего ряда только при формате 32 символа в строке.

Использование регистра фона в слайдовом режиме показано на рис. 3, в графическом и символьном режимах — в табл. 7. Регистр уставки адреса задает зоны экранной памяти как при работе ДК и ЦП с общим полем памяти (разряды 0...3) так и при отследиении экранной памяти от общесистемной магистрали (разряды 4...7).

Интерфейс внутреннего буфера данных

В алфавитно-цифровом режиме каждая строка символов отображается определенным числом линий строчной развертки (от 9 до 32 в зависимости от формата).

Подлежащую отображению символьную информацию можно накапливать в буфере данных уже за время 1—2 строк линий раstra при развертке на экране

изображения каждой строки символов, что существенно облегчает ЦП доступ к общей памяти.

В слайдовом режиме первые 16 ячеек внутреннего буфера данных используются для буферизации данных таблицы «Кадр» на одну координатную строку «У», следующие 16 ячеек — для буферизации битовой карты на две строки символов таблицы «Поле» и еще 16 ячеек — для буферизации данных из таблицы «Х». Таким образом, внутренний буфер данных контроллера и в слайдовом режиме дает возможность существенно разгрузить общую магистраль от обращений контроллера.

Интерфейс клавиатуры

Встроенный в ДК блок управления клавиатурой обеспечивает поиск нажатой на клавиатуре клавиши, освобождает от дребезга контакты клавиши, кодирует ее, запоминает до четырех клавиш, находящихся в нажатом состоянии, и последовательно передает их

Таблица 7

Разряды регистра фона	Назначение	Примечание		
7...4	RGB и Y	Задание цвета фона RGB и дополнительного		
3	Размер точки по кадру	0 — 1 строка раstra 1 — 2 строки раstra		
2	Число символов в строке	0 — 32 символа 1 — 64 символа		
1,0	Задание формата по кадру	Код	Символьный рядов	Графический, точек
		00	32	256×256
		01	24	256×128
		10	16	128×128
		11	25	64×64, блоков

коды по командам ЦП. Нажатая клавиша определяется сканированием всей клавиатуры с частотой строчной развертки (цикл сканирования равен длительности одного кадра изображения).

Запросы и коды от клавиатуры передаются в режиме векторного прерывания в соответствии с диаграммой векторного прерывания МПИ. Номер вектора прерывания клавиатуры предварительно записывается в регистр вектора ДК, нулевой разряд которого используется для разрешения прерывания по вектору клавиатуры.

Контроллер работает без прерывания (запрос на прерывание должен завестись на цифровой вход и использоваться как сигнал готовности клавиатуры, снимаемый при завершении чтения кода клавиши с регистра клавиатуры).

Сигнал с клавиатуры принимается по одному входу, поэтому для организации сканирования клавиатурного поля сигналами строчной и кадровой разверток и мультиплексирования его выходов на один вход ДК необходима внешняя схема. Эта схема работает синхронно с внутренней схемой синхронизации контроллера. Число клавиш в клавиатуре может достигать 256, при этом дополнительная внешняя схема сканирования должна содержать 8-разрядный счетчик, дешифратор на 16 выходов и мультиплексор с 16 входов на один выход.

Интерфейс с видеомонитором

Контроллер обеспечивает формирование видеосигналов по четырем каналам (R, G, B, Y), содержащим сигналы гашения и подсвета, а также раздельных или совмещенных сигналов строчной и кадровой синхронизации для работы со стандартными телевизионными мониторами.

При тактовой частоте 5 МГц микросхема ДК формирует сигналы строчной синхронизации с частотой 15625 Гц и кадровой синхронизации с частотой 50 Гц для работы в режиме прогрессивной развертки.

ДК может работать и при внешней строчной и кадровой синхронизации, что создает возможность совмещения и наложения изображений от нескольких контроллеров и других источников телевизионного изображения.

Телефон 291-67-41, Ленинград

ЛИТЕРАТУРА

1. Зеленко Г. В. Дисплей бытовой персональной ЭВМ // Микропроцессорные средства и системы. — 1985, № 3. — С. 60—69.
2. А. С. 1292031 СССР. Дисплейный процессор / Е. М. Блох, О. С. Горбачев, А. Ф. Иоффе, А. О. Петров, Н. И. Петрова, В. С. Хорин. — Оpubл. 1987. Бюл. № 7.
3. А. С. № 1142825 СССР. Устройство для отображения информации на экране электронно-лучевой трубки / Е. М. Блох, К. Б. Бодашков. — Оpubл. 1985. Бюл. № 8.

Статья поступила 8.06.89

ГРАФИЧЕСКИЙ СОПРОЦЕССОР ДЛЯ РАБОЧИХ СТАНЦИЙ И ПЭВМ

Графический сопроцессор i82786 предназначен для использования в растровых графических терминалах САПР, а также в профессиональных ПЭВМ высокой производительности. БИС i82786 обеспечивает скоростную обработку графических и текстовых данных, возможность их комбинации при выполнении операций кадрирования, осуществляет их высококачественное отображение и поддерживает режим многозадачности. Микросхема выполнена по высококачественной КМОП-технологии в корпусе аналогично 16-разрядному микропроцессору i80286. Потребляемая мощность микросхемы — 1 Вт.

Режимы работы

Особенность БИС i82786 заключается в слабосвязанной системной архитектуре, поддерживающей режим сопроцессоров. В качестве центрального процессора (ЦП) можно использовать микропроцессоры от 16-разрядного i8086 до 32-разрядного i80386.

БИС работает с ЦП в следующих режимах:

ведомого — i82786 рассматривается как интеллектуальный периферийный контроллер динамического ЗУПВ;

ведущего — i82786 работает аналогично ЦП, управляя системной шиной и организуя доступ к системной памяти.

При поступлении от i82786 запроса на захват системной шины ЦП предоставляет управление графическому сопроцессору. В свою очередь i82786, используя собственную 22-разрядную шину адресов и 16-разрядную шину данных, осуществляет доступ в системную память и выбирает данные и команды, сформированные ЦП.

В наиболее распространенной системной конфигурации с сопроцессором обеспечивается одновременный доступ ЦП к системной памяти и сопроцессора i82786 к видеопамяти. ЦП может обращаться к видеопамяти, а i82786 — к системной, но при запрете одновременного доступа.

Архитектура графического сопроцессора

БИС i82786 содержит четыре основных модуля: графический и дисплейный процессоры, интерфейсные устройства шины и контроллер видеопамяти (рис. 1). Все модули работают относительно независимо, что обеспечивает

высокие скоростные характеристики микросхемы.

Графический процессор выполняет команды, размещенные в системной памяти, и формирует битовые карты для дисплейного процессора, располагая их в видеопамяти.

Дисплейный процессор преобразует битовое отображение, созданное графическим процессором, в растровые последовательности для видеоконтрольного устройства.

Интерфейсные устройства шины реализуют четыре логических интерфейса: внешний; доступа к видеопамяти для внутреннего графического процессора; доступа ЦП к регистрам дисплейного и графического процессоров и видеопамяти; запросов для регенерации видеоОЗУ.

Использование **видеопамяти**, содержащей отображаемые данные, для размещения бинарных масок символов и команд программирования модулей сопроцессора позволяет значительно уменьшить временные затраты на отображение информации.

Графический процессор формирует растровые образы графических примитивов вывода (ГПВ) в видеопамяти во взаимодействии с контроллером динамического ЗУПВ и интерфейсными устройствами шины. Он обладает развитыми аппаратными возможностями, позволяющими снизить до минимума число параметров в командах для формирования растровых образов отдельных ГПВ. Список графических команд и их параметров создается ЦП.

Отличительная особенность БИС i82786 состоит в том, что графические команды обрабатываются самим сопроцессором при обращении к системной памяти и последовательном выборе команд формирования ГПВ и их параметров из связанного списка графических команд через логику шинного интерфейса. Для синхронизации работы ЦП и сопроцессора используется флаг завершения формирования списка параметров. БИС i82786 начинает выполнение команды, если ЦП завершил формирование списка параметров к этой команде

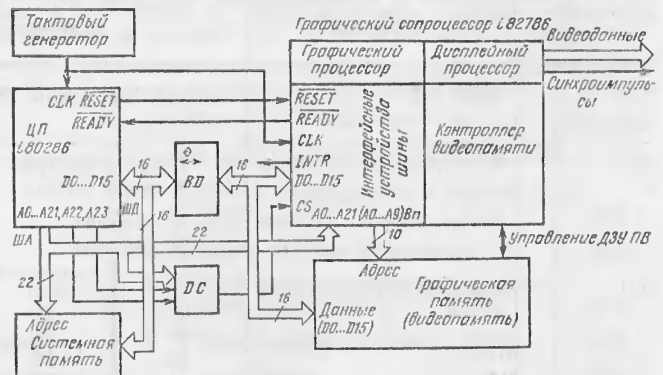


Рис. 1. Структура графического сопроцессора i82786

и установил соответствующий флаг. Такой режим обмена позволяет значительно повысить системную производительность.

Процесс выполнения команд начинается с загрузки старшего и младшего байтов адреса связи в управляющие регистры графического процессора. Графический процессор обрабатывает команды до тех пор, пока не обнаружит флаг остановки, определяемый нулевым разрядом регистра команды.

Графический процессор записывает сформированную битовую карту в соответствующую область видеопамати. Битовая карта представляет собой прямоугольную матрицу кодов элементов изображения (ЭИ) размерами, например 32К×32К ЭИ, с координатами, указанными в верхнем левом углу экрана. Коды ЭИ в битовой карте упаковываются последовательно в линейном порядке.

Производительность графического процессора зависит от частоты доступа к видеопамати (каждое записываемое слово требует выполнения цикла «чтение-модификация-запись»). Скорость работы графического процессора определяется также шириной полосы пропускания графической памяти и числом команд, выполняемых дисплейным процессором.

Имея специальные видеоЗУ, графический процессор может использовать до 99 % полосы пропускания, при обычных динамических БИС ЗУ—50—90 %.

Дисплейный процессор формирует синхросигналы и изображение на экране ВКУ, обеспечивает сканирование видеопамати, выдает 8-битовые коды ЭИ в канал видеовывода.

Особенность дисплейного процессора — возможность реализации полиэкранного режима, что предусматривает наличие развитых аппаратных средств его поддержки. Дисплейный процессор оптимизирован для данных, представленных в форме упакованных битовых карт. Он может переключаться в отдельном окне в режимы цветного графического адаптера IBM PC и адресации с перестановкой байтов.

Дисплейный процессор работает независимо от графического по собственной программе, описывающей таблицу конфигурации кадра, которая формируется ЦП и размещается в определенной области видеопамати. Обработывая битовые карты, дисплейный процессор обеспечивает выборку кодов ЭИ и воспроизводит их содержимое в нужных точках экрана, выполняет свои команды в течение вертикального бланкирования. Таким образом обеспечивается получение чистого, без мерцаний, изображения. Для увеличения гибкости системы интервальный таймер можно запрограммировать на выработку сигнала прерывания после вывода заранее заданного числа кадров.

Для синхронизации ЦП и дисплейного процессора используется механизм квитирования. После выполнения каждой команды дисплейный процессор устанавливает флаг, указывающий, что ЦП может загружать следующую команду.

Дисплейный процессор разделяет экранную область на горизонтальные полосы высотой в произвольное число строк. Полосы содержат блоки, которые относятся к различным сегментам окна (рис. 2, а). Структура управляющего блока дескриптора окна приведена на рис. 2, б.

Допускается до 16 горизонтальных окон-сегментов на каждую отдельную строку сканирования. Число вертикальных окон-сегментов определяется числом сканирующих строк. Минимальная ширина окна равна 1/16 длины строки. Максимальное число окон — 8192 (512 строк×16). Окна могут быть круглыми, эллипсовидными и прямоугольными. Вокруг окон, выведенных на экран, образуются цветные границы.

Таблица описаний дескрипторов полос содержит заголовок, в котором указаны число строк в полосе, адрес дескриптора следующей полосы, число блоков в текущей полосе. Затем идут данные, описывающие блоки и включающие указатели, ассоциированные с битовыми картами, и атрибуты. Таким образом для каждого кадра формируется таблица конфи-

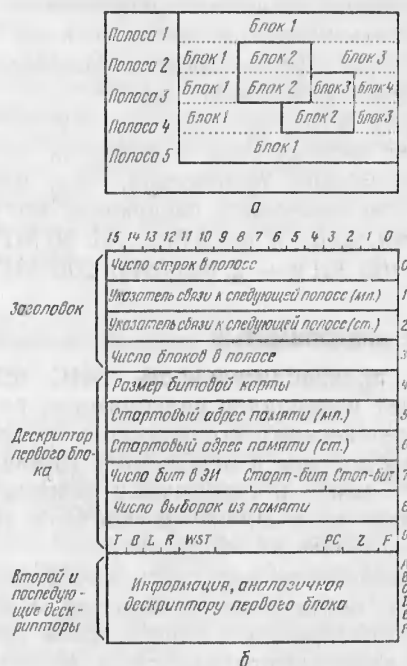


Рис. 2. Организация оконных сегментов (а) и структура управляющего блока дескриптора окна (б):

T, V, L, R — управляющие биты для верхней, нижней, левой и правой границ; WST — статус окна; Z — увеличение; F — фоновое поле

гурации в виде связанного списка команд дисплейного процессора.

Схема с использованием указателей связи позволяет дисплейному процессору перемещать, просматривать, панорамировать изображение путем считывания новых дескрипторов без формирования растровых образов в видеопамяти.

Для упорядочения расположения оконных сегментов ЦП модифицирует дескрипторы в таблице конфигурации кадра. Области экрана, не распределенные для отображения какого-либо окна, заполняются однородным фоновым цветом. Код цвета хранится в одном из регистров дисплейного процессора, поэтому нет необходимости выделять специальные области видеопамяти для хранения «пустого» изображения. Кроме того, снижаются временные затраты, на пересылку данных.

Дисплейный процессор может увеличивать масштаб изображения на экране дисплея по горизонтали и вертикали в 1...64 раз благодаря репликации ЭИ, применимой ко всем окнам. При нормальной частоте видеовывода 25 МГц и 8 бит/ЭИ можно разместить окна на экране с разрешающей способностью до единичного ЭИ. Можно установить курсор дисплея размерами 8×8 или 16×16 ЭИ.

Курсор может быть прозрачным, непрозрачным, блоком или визиром с пересечением всего экрана. Форма курсора — стрелка, квадрат, крест и т. д. Цвет курсора, его прозрачность и мерцание задаются программным способом. Программируются также сигналы синхронизации ЭЛТ с разрешающей способностью ЭИ до формата 4096×4096 ЭИ.

В зависимости от числа битов на ЭИ для одной частоты видеовывода изменяется режим работы дисплейного процессора. Так, например, можно использовать следующие частоты тактирования: 25 МГц при 8 бит/ЭИ; 50 МГц — 4 бит/ЭИ; 100 МГц — 2 бит/ЭИ; 200 МГц — 1 бит/ЭИ.

Управление видеопамтью

Высокую производительность БИС i82786 поддерживает контроллер видеопамяти, позволяющий графическому сопроцессору реализовать режимы доступа к отдельному 16-разрядному слову памяти и страничные режимы ускоренного доступа к динамическим ОЗУ (максимальная скорость до 40 Мбайт/с).

Область линейного адресного пространства графического процессора начинается с нулевого шестнадцатеричного адреса. Если реальный объем видеопамяти меньше 4 Мбайт, то для доступа по оставшимся адресам используется системная память.

БИС i82786 может работать с динамическими ОЗУ различного типа емкостью от

16 Кбит до 1 Мбит и разрядностью 1, 4, 8 бит с использованием страничного или скоростного страничного режима доступа для повышения производительности. Доступ к ОЗУ со скоростным страничным режимом выполняется за 100 нс; с обычным страничным режимом — за 200 нс. Интерфейс памяти поддерживает до 32 БИС динамических ОЗУ без внешней логики в пределах адресуемой памяти.

Использование переменного числа битов на кодирование одного ЭИ обеспечивает максимальную плотность упаковки битовых карт, экономит память и полосу пропускания. Приоритеты доступа к памяти по умолчанию устанавливаются в следующем порядке: регенерация ОЗУ, дисплейный и графический процессоры, ЦП.

Регенерация ОЗУ имеет наивысший приоритет, остальные приоритеты можно изменять программным способом и программировать уровни производительности сопроцессора.

Внутренние регистры и система команд графического сопроцессора

Графический сопроцессор i82786 программируется через регистровый блок, состоящий из 128 регистров [2]. Несмотря на то, что регистры распределены по всем модулям сопроцессора, они адресуются как единый блок в адресном пространстве памяти или ввода-вывода. Внутренние регистры можно разделить на группы интерфейсных устройств, графического и дисплейного процессоров. В группу интерфейсных устройств входят следующие регистры: управления интерфейсными устройствами; перемещений, обеспечивающий относительную адресацию остальных регистров интерфейсных устройств;

- управления регенерацией памяти;
- управления динамической памятью;
- приоритета для графического процессора, дисплейного процессора и управления внешним приоритетом.

Группа графического процессора содержит регистры

- кода операции;
- адреса связи;
- состояния;
- указателя команды.

В группу дисплейного процессора входят регистры

- кода операции;
- адреса видеопамяти;
- идентификатор прерывания;
- состояния;
- указателя отсутствия видеоданных.

Команды управления вычерчиванием поддерживают режимы «разрыва» формируемого растрового ГПВ, поиска объекта по курсору, управляемому от манипулятора типа «мышь»,

создания объектов более сложной формы. При этом графический процессор как бы заново формирует битовую карту, реально не обновляя содержимое видеопамати. При попадании объекта в прямоугольник с заданными размерами, координаты которого задаются курсором, вырабатывается сигнал прерывания, что позволяет идентифицировать графический объект.

Команды формирования растровых образов ориентированы на стандарт CGI. Аппаратными средствами графического процессора реализуются следующие ГПВ: точка, последовательность точек, линия, параллельные линии, дуга, окружность, прямоугольник, многоугольник, ломаная линия, открытый и закрытый линейный мультисегмент, мультисегмент многоугольников, поточечное построение.

Формат команд графического процессора определяется 16-разрядным полем, содержащим 8-разрядный код операции и флаг завершения формирования списка параметров, и полем параметров, включающим координаты точек привязки и геометрические параметры ГПВ (например, радиус окружности, координаты точек дуги).

Графический сопроцессор выполняет команды блочной пересылки битов Bit BLT (BBLT) со скоростью 24 Мбит/с. Одновременно с пересылкой осуществляются логические операции над кодами ЭИ и маскирование отдельных разрядных слоев от записи, что позволяет с высокой скоростью генерировать в видеопамати различные изображения путем копирования целых образов или отдельных частей фигур. Можно использовать режим обращения битов, позволяющий получать негативные изображения.

Растровые образы символов формируются командой Cha Bit.

Система команд графического сопроцессора делится на следующие группы:

- начальной инициализации сопроцессора;
- системного управления и обеспечения взаимодействия модулей сопроцессора;
- управления вычерчиванием;
- формирования растровых образов;
- обработки символов;
- управления отображением.

Команды начальной инициализации сопроцессора предназначены для подготовки графического сопроцессора к работе.

Команды системного управления используются для загрузки регистров графического процессора, сохранения содержимого регистров в системной памяти, подготовки внутренних регистров к программному обмену, вызова макрокоманд, генерации прерывания для ЦП.

Команды управления вычерчиванием используются для определения:

областей видеопамати, выделяемых для растровых образов;

координат границ этой области;
правил отсечения фрагментов изображения, выходящих за границы этой области;

числа битов, выделяемых для кодирования цвета и интенсивности каждого ЭИ;

цвета переднего плана и фона;

алгоритмов наложения областей, имеющих одинаковый цвет, на область другого цвета (объекты могут быть прозрачными и непрозрачными);

цвета формируемых линий, текстуры заполняемых областей, со скоростью 25 000 символов/с (одной командой можно выбрать любой набор символов). Предусматривается пропорциональное размещение символов, причем ширина и высота каждого символа определяются независимо. Расстояние между символами может изменяться вплоть до наложения символов. При отображении символов дисплейный процессор автоматически дополняет код ЭИ до полной цветности в соответствии с атрибутами окна.

Команды управления отображением обеспечивают управление режимом отображения в реальном масштабе времени. Они оперируют с 4016-разрядными регистрами как с единым блоком памяти. Одни внутренние регистры определяют различные режимы отображения включая координаты и форму курсора, цвет фона, границы окон, коэффициент аппаратного увеличения; другие непосредственно управляют сопряжением с ВКУ, работой видеотракта, определяют формат изображения, скорость выдачи ЭИ и другие параметры.

Начальное программирование БИС i82786 выполняется ЦП, который предварительно распределяет память, определяет конфигурацию видеопамати, набор параметров регенерации и приоритетов для модулей графического сопроцессора.

Производительность графического сопроцессора

Графический сопроцессор можно использовать в формате монохромного и цветного адаптеров ПЭВМ РС фирмы IBM. Разработчик может установить набор специальных битов и выбрать один из форматов IBM РС.

Сопроцессор i82786 поддерживает форматы IBM РС в 2 и 4 бит/ЭИ, поэтому программы для IBM РС могут отображать графические данные совместно с образами, сформированными БИС i82786. Работа i82786 обеспечивается системным тактированием 10 МГц и видеотактированием 25 МГц. БИС i82786 вычерчивает линии со скоростью 2,5 млн. ЭИ/с,

окружности — 2 млн. ЭИ/с, заполняет область — 30 млн. бит/с. Шинный интерфейс графического сопроцессора передает блоки данных со скоростью 24 млн. бит/с. Внутренняя аппаратура обрабатывает шрифты до 16×16 ЭИ. Более крупные шрифты требуют аппаратной и программной поддержки.

Сопроцессор i82786 может выполнять вертикальное и горизонтальное перемещения без дополнительной внешней аппаратуры, обеспечивать аппаратное увеличение с коэффициентом масштабирования до 64, одновременно отображать до 1024 цветов и поддерживать разрешающую способность от $640 \times 480 \times 8$ до $1024 \times 1024 \times 2$ бит с прогрессивной разверткой на частоте до 60 Гц.

Графический сопроцессор i82786 оптимизирован под микропроцессор i80286, который выполняет функции ЦП в графической системе. В этом случае интерфейсная логика, используемая для сопряжения ЦП и графического сопроцессора, минимальна. Внешняя и встроенная логика ЦП и сопроцессора обеспечивают разделение доступа к общесистемным ресурсам [3].

Графический сопроцессор i82786 позволяет значительно повысить производительность графических систем с многозадачным и полиэкранными режимами работы. Поэтому одно из перспективных применений БИС графического сопроцессора i82786 наряду с САПР и электронными издательскими системами — ПЭВМ высокой производительности, где сопроцессор эффективно поддерживает работу многооконных программных систем GEM и X — WINDOWS.

480013, Алма-Ата, ул. Космонавтов, 140, Спецкафедра КазПТИ; тел. 67-43-02

ЛИТЕРАТУРА

1. Nicholls B. Inside the 82786 graphics chip // Byte.— 1987.— V. 12.— № 9.— P. 135—141.
2. Randall M., Iohary A. Graphics coprocessor chip gives small systems a look at the big picture // Electronics Design.— 1986.— V. 29.— № 6.— P. 159—166.
3. Torres R. Coprocessor grafico 82786 // Elettron. oggi.— 1988.— № 63.— P. 67—68, 70—71, 73—75.

Статья поступила 23.03.89

УДК 681.323

А. А. Куценко, В. К. Ковалевский

КМОП БИС КР1820ВП1 — ПЕРИФЕРИЙНОЕ УСТРОЙСТВО ДЛЯ ОДНОКРИСТАЛЬНЫХ ЭВМ

Микросхема КР1820ВП1 обеспечивает хранение содержимого ОЗУ ЭВМ при отключении источника питания, формирует специальный сигнал управления для циклического отклю-

чения питания ОЭВМ, когда пауза между обращениями к микроЭВМ превышает заданное значение.

БИС изготовлена по КМОП-технологии с самосовмещенным поликремниевым затвором и изоляцией элементов толстым окислом. Выполнена в 14-выводном пластмассовом корпусе типа 201.14-1. Ориентировочная цена 2,7 руб.

Основные параметры микросхемы КР1820ВП1

Напряжение питания, В	5,0 ± 10 %
Потребляемая мощность, мВт, не более	0,165
Число режимов работы	2
Число каналов обмена	2
Разрядность команды	5
Число команд	6
Скорость обмена данными с внешним устройством, Мбит/с	0,25
Емкость ОЗУ, бит	256

Структурная схема микросхемы и временные диаграммы работы приведены соответственно на рис. 1, 2.

Генератор G предназначен для формирования тактовых импульсов, синхронизирующих работу делителя DIV1. К выводам генератора подключается кварцевый резонатор и элементы подстройки частоты.

Счетчик адреса СТА вырабатывает коды адресов ячеек, где хранятся данные, поступающие на вход DI от ОЭВМ.

Дешифратор команд DCINS1 предназначен для декодирования команд чтения (READ), записи (WRITE), запрета записи (WRDS), разрешения записи (WREN), отключения контроллера (SLEEP). Дешифратор DCINS2 декодирует команды флага начала отсчета временного интервала (TSEC).

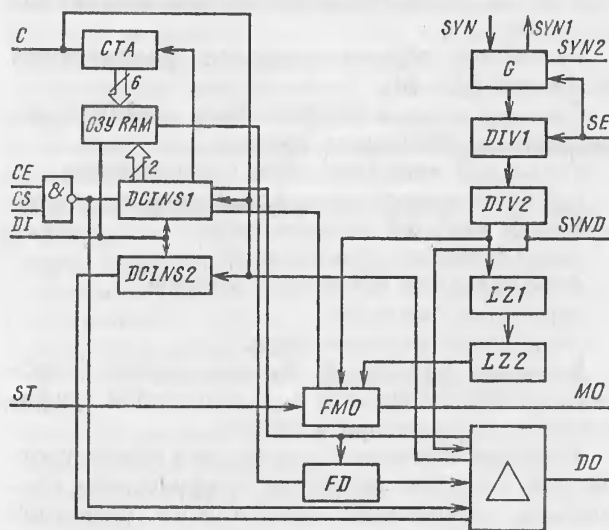


Рис. 1. Структурная схема БИС КР1820ВП1:

СТА — счетчик адреса; DCINS1,2 — дешифраторы команд; G — генератор; DIV1,2 — делители с переменными коэффициентами; LZ1,2 — линии задержки; FMO — блок формирования сигнала запуска контроллера; FD — блок формирования выходных данных; Δ — выходной усилитель

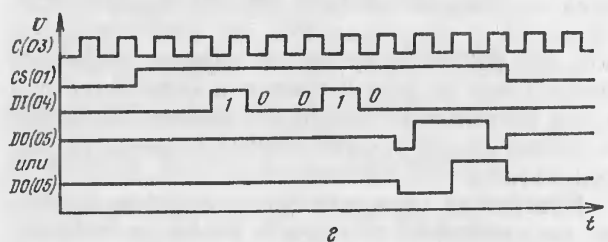
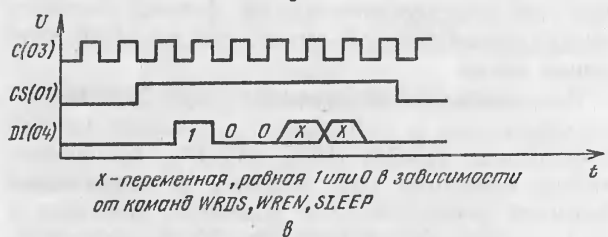
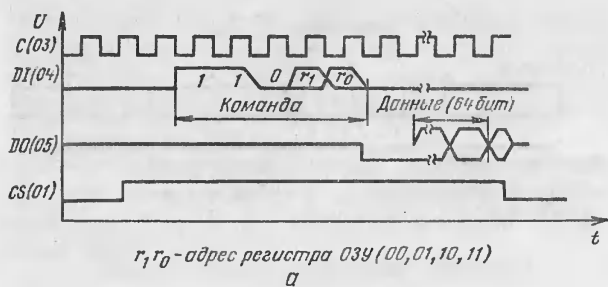


Рис. 2. Временные диаграммы работы микросхемы КР1820ВП1 при выполнении команд READ (а), WRITE (б), WRDS, WREN, SLEEP (в), TSEC (г)

Блок формирования сигнала запуска контроллера FMO формирует на выходе MO сигнал высокого уровня при поступлении команды SLEEP, отключая тем самым ОЭВМ. Блок формирования выходных данных FD обрабатывает данные, поступающие с выхода ОЗУ. Делитель DIV1 осуществляет предварительное деление синхронизирующей частоты и формирует синхриимпульсы для делителя DIV2. Он синхронизируется импульсами, поступающими с генератора G. Управляется этот делитель по входу SE. Делитель DIV2 вырабатывает тактовые сигналы для внешнего устройства и отсчитывает интервалы времени. Он синхронизируется импульсами, генератора G или делителя DIV1. Управляется DIV2 по выводу SE; результат его работы выдается на вывод SYND.

Линии задержки LZ1, LZ2 задерживают сигнал, поступающий с выхода делителя DIV2

на входы блока формирования сигнала запуска контроллера FMO.

Микросхема КР1820ВП1 работает в режимах: загрузки-выгрузки данных из ОЗУ внешней ЭВМ и отсчета интервала времени и запуска ЭВМ. Первый режим поддерживается командами READ, WRITE, WRDS, WREN, SLEEP (табл. 1), второй командой TSEC. Назначение

Таблица 1
Коды команд WRDS, WREN, SLEEP

Команда	Код
WRDS	10000
WREN	10011
SLEEP	10001

Таблица 2
Назначение выводов микросхемы КР1820ВП1

Вывод	Обозначение	Тип	Примечание
1	CS	Вход	Выбор кристалла
2	CE	»	Синхросигнал разрешения
3	C	»	Тактирование
4	DI	»	Ввод данных
5	DO	Выход	Вывод данных
6	SE	Вход	Выбор условия работы
7	GND	—	Общий
8	SIND	Выход	Синхронизация внешнего устройства
9	MO	»	Запуск контроллера
10	ST	Вход	Внешний запуск
11	SYN	»	Вход генератора
12	SYN 1	Вход-выход	Подключение кварцевого резонатора
13	SYN 2	Выход	Выход генератора с открытым стоком
14	U _{CC}	—	Напряжение источника питания

выводов микросхемы приведено в табл. 2. При подаче на вывод 1 сигнала низкого уровня сбрасывается регистр команд RGINS и счетчик адреса СТА, запрещается работа всех блоков микросхемы, за исключением генератора G и делителей DIV1, DIV2. Вывод 5 переходит в третье состояние. Сигнал высокого уровня разрешает работу счетчика адреса СТА и дешифратора команд DINS. Вывод 2 используется при переводе внешней ЭВМ в отключенное состояние, вывод 3 — для тактирования микросхемы в процессе записи и чтения последовательных данных.

По выводу 4 данные поступают ОЭВМ серии КР1820, по выводу 5 — из ОЗУ микросхемы КР1820ВП1 подаются на вход данных ОЭВМ. Активные логические уровни на выводе 5 формируются только по командам TSEC и READ. При выполнении остальных

команд вывод 5 (DO) находится в третьем состоянии.

Вывод 6 предназначен для управления режимом работы генератора G и коэффициентом деления делителя DIV. На выводе 8 (SYND) формируется последовательность импульсов с периодом 1 с (результат работы делителя DIV).

Сигнал высокого уровня, формируемый на выводе 9 запрещает, а сигнал низкого уровня разрешает работу ОЭВМ. Подача на вывод 10 сигнала высокого уровня блокирует возникновение сигнала запрета на выводе 9 (MO) при выполнении команды SLEEP. Выводы 12, 13 предназначены для подключения кварцевого резонатора и элементов подстройки частоты или тактирования от внешнего генератора.

БИС КР1820ВП1 взаимодействует с ОЭВМ серии КР1820 следующим образом: ОЭВМ при отсутствии к ней обращений в течение определенного времени вырабатывает команды WREN и WRITE и перезаписывает данные, хранящиеся в собственном ОЗУ и в ОЗУ БИС КР1820ВП1. Далее ОЭВМ формирует команду TSEC, в результате выполнения которой на вывод DO поступает сигнал специальной формы, являющийся для ОЭВМ флагом начала отсчета временного интервала регулируемой длительности. Затем выдается команда SLEEP, на выводе MO формируется Лог. 1 и ОЭВМ отключается от источника напряжения. После окончания регулируемого временного интервала состояние вывода MO изменяется на Лог. 0, ОЭВМ подключается к источнику и анализирует сообщение от оконечных устройств. При необходимости этот цикл повторяется.

БИС КР1820ВП1 позволяет использовать 4-разрядные однокристалльные микроЭВМ серии КР1820 в системах управления с автономным энергообеспечением, что существенно расширяет область их применения.

Телефон 77-68-53, Минск

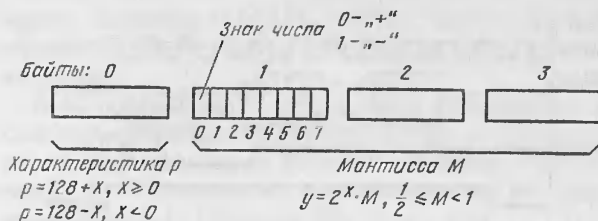
Статья поступила 4.07.89

УДК 681.3.06:326

Ю. М. Руденко

РЕАЛИЗАЦИЯ ПЛАВАЮЩЕЙ АРИФМЕТИКИ И ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ ДЛЯ МИКРОПРОЦЕССОРОВ СЕРИИ К1816, К1810, К580, К1821

Структура данных. Данное с плавающей точкой занимает четыре байта (см. рисунок): в байте 0 — порядок числа, в байтах 1...3 — мантисса. Из-за необходимости контроля за нормализацией числа оставлен явно присутствующим старший разряд мантиссы, хотя это и привело к потере точности в один двоичный разряд. Величина порядка смещена на 128,



Структура числа в плавающем формате

поэтому характеристика принимает только положительные значения. Мантисса хранится в прямом коде. В качестве нуля взят шестнадцатеричный код 01400000 вместо 80000000, принятого в первоначальном варианте [1]. Код 80000000 вызывает в отдельных случаях большие погрешности при выполнении арифметических операций, а также приводит к непредвиденным аварийным завершениям, так как по определению такая форма является экстракодом для выбранного способа представления чисел.

Разработаны программы преобразования используемых в библиотеке форматов данных к форматам ПЭВМ ДВК, ЕС, РС. Арифметические операции над числами в плавающем формате реализованы с помощью обычных в этих случаях алгоритмов [2]. Исходные числа хранятся в регистрах R0...R7 RB1. Регистры R0...R7 в отличие от первого варианта библиотеки не используются и оставлены для нужд потребителя. Операция деления засылает в ячейки 32...39 памяти промежуточные результаты.

Программа сложения (вычитания) использует выравнивание порядка в большую сторону, затем сдвиг мантиссы слагаемого вправо, сложение мантиссы и нормализацию результата. Формируются коды аварийных завершений: исчезновение мантиссы, исчезновение и переполнение порядков.

Программа вычитания выполняется с помощью программы сложения и имеет те же коды аварийных завершений.

Программа умножения определяет предварительный порядок произведения, выполняет 23 сдвига множителя на один разряд вправо (младшими разрядами вперед) и сложение мантиссы множителя с мантиссой произведения, если очередной выдвигаемый разряд множителя равен единице. При нулевом разряде множителя происходит сдвиг частичного произведения влево. Частичные произведения корректируются в зависимости от значений старшего отбрасываемого разряда остатка. Предусмотрены аварийные выходы: переполнение и исчезновение порядка.

Деление выполняется методом последовательного вычитания мантиссы делителя из мантиссы делимого и сдвига остатка влево.

Реализован метод без восстановления остатка. Текущий полученный разряд частного сразу записывается на соответствующее место, т. е. при получении нулевого остатка операция деления прекращается.

Предусмотрены аварийные выходы: деление на ноль, переполнение и исчезновение порядка частного. Карты размещения модулей приведены ниже.

Сложение

**** МОД..MAIN. ФайлаDK:PRZ.0.48
PRZ 0191
**** МОД..MAIN. ФайлаDK:NORM.0.48
NORM 01D7
**** МОД..MAIN. ФайлаDK:AD.0.48
AD 0234
**** МОД..MAIN. ФайлаDK,SHN.0.48
CHN 0241
**** МОД..MAIN. ФайлаDK:PR.0.48
PR 0252
**** МОД..MAIN. ФайлаDK:ADI.0.48
ADI 0264

Вычитание

**** МОД..MAIN. ФайлаDK:PRZ.0.48
PRZ 0191
**** МОД..MAIN. ФайлаDK:NORM.0.48
NORM 01D7
**** МОД..MAIN. ФайлаDK:AD.0.48
AD 0234
**** МОД..MAIN. ФайлаDK:CHN.0.48
CHN 0241
**** МОД..MAIN. ФайлаDK:PR.0.48
PR 0252
**** МОД..MAIN. ФайлаDK:ADI.0.48
ADI 0264
**** МОД..MAIN. ФайлаDK:B.0.48
SUB 0465

Умножение

**** МОД..MAIN. ФайлаDK:PRZ.0.48
PRZ 0191
**** МОД..MAIN. ФайлаDK:AD.0.48
AD 0234
**** МОД..MAIN. ФайлаDK:CHN.0.48
CHN 0241
**** МОД..MAIN. ФайлаDK:FMUN.0.48
FMUN 0470

Деление

**** МОД..MAIN. ФайлаDK:PRZ.0.48
PRZ 0191
**** МОД..MAIN. ФайлаDK:AD.0.48
AD 0234

**** МОД..MAIN. ФайлаDK:CHN.0.48
CHN 0241
**** МОД..MAIN. ФайлаDK:PR.0.48
PR 0252
**** МОД..MAIN. ФайлаDK:FDIV.0.48
EDIV 000C

Имена программ:

ADI — сложение; SUB — вычитание; FMUN — умножение; FDIV — деление; PRZ — проверка равенства числа нулю; NORM — нормализация числа; AD — сложение мантисс; CHN — обмен содержимого регистров (R0, R4), (R1, R4), ... (R3, R7); PR — проверка равенства нулю мантиссы.

Между программами оставлены «окна», в которые обычно помещаются другие программы, например преобразования целых чисел в вещественные, выделения целой части числа и др. Если используются только программы арифметики с плавающей точкой, то другие программы можно разместить вплотную друг к другу. Элементарные функции (логарифмические, тригонометрические, экспоненциальные и др.) вычисляются на основе приближения Паде [3], представляющего собой отношение двух многочленов. Для достижения точности порядка $10E-7$ достаточно взять многочлены степени $N=4, 5$ и задать аргумент X в интервале значений в соответствии с требованиями формулы Паде [3]. Функция извлечения квадратного корня реализуется с помощью формулы Герона [4] (для достижения точности примерно $10E-7$ достаточно 3, 4 итераций).

Степенная функция вычисляется по одной из двух программ. Если показатель степени целый, то с помощью умножения по схеме $R \times \times 2$, $R \times \times 4$, $R \times \times 8$ берется произведение $R \times R \times \times 2 \times R \times \times 8$. Таким образом, для вычисления $R \times \times 11$ требуется пять умножений (R — основание степени).

Для степенной функции с произвольным показателем используется формула в виде экспоненты произведения аргумента на натуральный логарифм основания. Погрешность вычисления по ней на всем интервале допустимых значений не превышает $10E-7$. Используя данные таблицы, можно всегда определить, какую из двух программ целесообразно применить в конкретном случае.

Если значение проектируемой функции приближается к асимптотическому или $k - \frac{1}{n}$, $+\frac{1}{n}$, где $\frac{1}{n}$ — символ бесконечности, то в аккумуляторе микропроцессора формируется код сообщения, а в регистрах R4...R7 RB1 — требуемое асимптотическое значение.

Символ $-\frac{1}{n}$ формирует шестнадцатеричный код FFFFFFFF, $+\frac{1}{n}$ — код FF7FFFFFFF.

Критическое приближение функции к асимптотическому значению завершается обычно аварийным сбоем одной из программ сложения (вычитания), умножения, деления внутри программы, реализующей рассматриваемую функцию, которая затем обрабатывается программой вычисления функции.

Все элементарные функции используют арифметические операции и другие вспомогательные функции, которые в общей сложности занимают 1 Кбайт памяти. Гиперболические функции вычисляются через экспоненту, десятичный логарифм — через натуральный и т. д.

В таблице приведена часть программ библиотеки. По договорам с предприятиями и организациями могут быть разработаны новые программы, учитывающие конкретные условия применения.

Характеристики программ

Программа	Время выполнения при тактовой частоте 3 МГц, мс	Объем памяти, байт	Объем данных, байт
Сложение	1	169	—
Вычитание	1	9	—
Умножение	3	181	—
Деление	4	263	8
Извлечение квадратного корня	20	169	10
Возведение в целую степень (P×11)	16	170	10
Возведение в произвольную степень	47	239	6
Натуральный логарифм	29	192	10
Десятичный логарифм	32	12	—
Экспонента	17	172	—
Синус	33	250	10
Тангенс	30	210	10
Арксинус	21	210	10
Арктангенс	22	215	10
Гиперболический синус	40	20	—

Опыт эксплуатации первого варианта библиотеки показал, что некоторые программные модули можно исключить из библиотеки, другие не оформлять как внешние. Это позволило сократить номенклатуру модулей. Так были исключены модули, реализующие плавающую арифметику над числами с двойной точностью (до пятнадцати десятичных разрядов). Введен статус внутренних для подпрограмм сдвига влево, вправо, получения дополнительных кодов, пересылки данных в памяти и т. д. Добавлены новые программы: вычисление функции ошибок, преобразования Фурье и гамма-функции.

Такая реорганизация библиотеки позволила сделать ее более обозримой, транспортабельной, удобной для пользователя и расширить область применения. Быстродействие программ воз-

росло в среднем на порядок, объем занимаемой памяти уменьшился примерно вдвое. Библиотека занимает три пяти- или восьмидюймовых дискеты. В комплект поставки входит описание каждой программы, составленное в соответствии с ЕСПД.

143890, Моск. обл., Нарофоминский р-н, пос. Кокошкино, ул. Дачная, 8, кв. 112; тел. раб. 434-94-65, 434-75-74 с 13 до 18 ч.

ЛИТЕРАТУРА

1. Руденко Ю. М., Чернова Т. Ф. Библиотека стандартных программ для ЭВМ серии К1816 // Микропроцессорные средства и системы.— 1987.— № 6.— С. 22—25.
2. Дроздов Е. А., Комарницкий В. А., Пятибратов А. П. Электронные вычислительные машины единой серии.— М.: Машиностроение, 1981.
3. Люк Ю. Специальные математические функции и их аппроксимации.— М.: Мир, 1980.— С. 41—117.
4. Математический анализ. Вычисление элементарных функций.— М.: Физматгиз, 1963.— 40 с.

Статья поступила 19.10.89

РЕКЛАМА

**ПРОИЗВОДСТВЕННО-ТЕХНИЧЕСКИЙ
КООПЕРАТИВ «АРКТУР»**

ПРЕДЛАГАЕТ ИНТЕРФЕЙСЫ ДЛЯ ЭВМ:

АДАПТЕРЫ приборного интерфейса (КОП, IEEE-488, GP-IB) для ЭВМ IBM PC XT/AT и ДВК IECmaster2. Предназначены для подключения до 15 приборов с КОП. Интерфейсы по желанию заказчика могут иметь дополнительные 8 битные порты ввода и вывода, а также комплектоваться модулями для подключения измерительных приборов с TTL-выходом без КОП. В комплект входят программы BIOS, а также демонстрационные программы цифровых вольтметров и осциллографов С9-8, С9-16.

Адаптер параллельного интерфейса для IBM PC XT/AT, по разьему и функциям совместимый с интерфейсом И2 «Электроники 60».

Универсальный макетный модуль METRA-board для IBM PC XT/AT, имеющий макетное поле и смонтированную канальную часть, 8-битные порты ввода-вывода, таймер i8253, интерфейс i8255 и логику прерывания и ПДП. В комплект входит пособие по созданию устройств и программированию аппаратного уровня, а также связи с языками высокого уровня.

Модули АЦП и ЦАП среднего быстродействия для IBM PC XT/AT.

Адрес для дополнительной информации: Москва, 125190, А190, А/Я 71, «АРКТУР».

УДК 681.3.06

А. В. Лопатин, Б. А. Позин

ОБРАЩЕНИЯ К УСТРОЙСТВАМ ВВОДА-ВЫВОДА СРЕДСТВАМИ ЯЗЫКА ПАСКАЛЬ М86

При реализации прикладных программ различного назначения на языке Паскаль М86, в среде ОС М86 возникают проблемы организации эффективной работы с устройствами ввода-вывода из-за того, что множество встроженных процедур и функций библиотеки компилятора Паскаль М86 ограничено. Обычно процедуры обращения к устройствам ввода-вывода пишут на языке ассемблера и включают в Паскаль-программу с помощью аппарата внешних процедур. Однако можно поступить иначе, используя средства самого компилятора Паскаль М86. В частности, разработан ряд процедур и функций, обеспечивающих работу с дисковой системой и клавиатурой, экраном, динамиком; управление принтером; переопределение назначения функциональных клавиш.

При работе с прерываниями используются переменные типа Result, определенные следующим образом:

```
type Result = record
```

```
  AX, BX, CX, DX, BR, SI, DI, DS, ES, Flags: Integer
```

```
end;
```

Функция Active — Disk позволяет получить номер активного дисковода, а процедура New — Disk — задать его номер. Их действие основано на использовании функций базовой системы ввода-вывода (BCBV), обращение к которым осуществляется через прерывание 224.

```
Function Active — Disk: Integer;
```

```
Var Register: Result;
```

```
Begin
```

```
  Register.CX := $19; {Запись в регистр CX номера функции}
```

```
  Intr (224, Register); {Обращение к функциям BCBV}
```

```
  Active — Disk := Register.AX + 1;
```

```
End;
```

```
Procedure New — Disk (C: Char);
```

```
Var Register: Result;
```

```
Begin
```

```
  Case C of {Запись в регистр DX номера диска}
```

```
    '1', 'A': Register.DX := 0;
```

```
    '2', 'B': Register.DX := 1;
```

```
    'M': Register.DX := 12; {M — диск}
```

```
  End;
```

```
  Register.CX := $0E; {Запись в регистр CX номера функции}
```

```
  Intr (224, Register); {Обращение к функциям BCBV}
```

```
End;
```

Задачу работы с клавиатурой решает процедура Key Board, использующая прерывание 16H. Она ожидает нажатие на клавиатуре и возвращает в первом параметре код символа, соответствующего данной клавише (если такое соответствие имеется), и скан-код нажатой клавиши — во втором параметре.

```
Procedure Key Board (var ASCII — Code, Scan — Code: byte);
```

```
Var Register: Result
```

```
  A: array [1..2] of Byte absolute Register;
```

```
Begin
```

```
  A [1] := 0; {Запись 0 в регистр AL}
```

```
  A [2] := 0; {Запись 0 в регистр AH}
```

```
  Intr ($16, Register);
```

```
  ASCII — Code := A [1];
```

```
  Scan — Code := A [2];
```

```
End;
```

Пример соответствия комбинаций клавиш и вырабатываемых кодов дан в табл. 1.

Таблица 1

Клавиша	ASCII — Code	Scan — Code
A	208	33
Упр-А	6	33
Доп-А	0	33
Ключ	27	1
Ввод	13	28
Ф5	0	63
Упр-Ф5	0	98
Доп-Ф5	0	108

Перед выдачей информации на печать полезно убедиться в готовности печатающего устройства с помощью процедуры Printer, возвращающей значение параметра OK=TRUE в случае готовности устройства и OK=FALSE при не включенном устройстве, конце бумаги, сбос, тайм-ауте и т. д.

```
Procedure Printer (var Ok: Boolean);
```

```
Var Register: Result
```

```
  A: array [1..2] of Byte absolute Register;
```

```
Begin
```

```
  A [1] := 0; {Запись 0 в регистр AL}
```

```
  A [2] := 0; {Запись 0 в регистр AH}
```

```
  Register.DX := 0; {Запись 0 в регистр DX}
```

```
  Ok := TRUE;
```

```
  Intr ($17, Register);
```

```
  if A [2] (>) 144 then Ok := FALSE;
```

```
End;
```

В некоторых случаях бывает удобно инициализировать распечатку содержимого экрана (например, при комплексной отладке, распечатке коротких таблиц и т. п.). Эта операция выполняется процедурой Print_Screen, также основанной на управлении прерыванием:

```
Procedure Print_Screen;
```

```
Var Register: Result
```

```
Begin
```

```
  Intr (5, Register);
```

```
End;
```

При решении некоторых задач необходимо выполнять определенные управляющие последовательности символов. В языке Паскаль М86 эти последовательности задаются с помощью оператора Write, при этом ни на экране, ни на печати не отображаются. Например, для отмены изображения курсора на экране используется функция с десятичным номером 110. Для ее выполнения в Паскаль-программе можно записать оператор

```
Write (#27, #110);
```

```
или эквивалентный ему
```

```
Write (#27, 'n').
```

Функции некоторых управляющих последовательностей приведены в табл. 2.

В процессе работы прикладной программы часто возникает необходимость в переопределении функциональных клавиш. Обычно эта статистическая операция выполняется с помощью команды ПАРМ86. Однако с использова-

Таблица 2

Символ	Код	Функция
1	49	Установить режим работы с отображением строки состояния
0	48	Установить режим работы без отображения строки состояния
H	72	Установить курсор в начало экрана
A	65	Переместить курсор вверх на строку
B	66	Переместить курсор вниз на строку
C	67	Переместить курсор вправо на позицию
D	68	Переместить курсор влево на позицию
j	106	Сохранить положение курсора
R	107	Восстановить положение курсора
I	74	Очистить экран от курсора до конца окна
o	111	Очистить строку от начала строки до курсора
N	78	Удалить текущий символ и сдвинуть оставшуюся правую часть строки
l	i08	Очистить текущую строку
m	109	Отобразить курсор
n	110	Отменить отображение курсора
p	112	Установить реверс изображения
q	113	Отменить реверс изображения
s	115	Установить режим мерцания изображения
t	116	Отменить режим мерцания изображения

нием управляющих последовательностей эту функцию можно задать непосредственно из программы с помощью следующей процедуры:

```
Procedure Fun_Key_Mode (N_Fun_Key: byte; S: Str 20);
```

```
Var i: byte;
```

```
Begin
```

```
  write (#27, #58); {Управляющая последовательность, задающая функцию изменения назначения ФК}
```

```
  write (char (N_Fun_Key+58)); {Задание номера изменяемой ФК}
```

```
  for i := 1 to Length (S) do write (S[i]); {Цикл посимвольного присвоения нового значения}
```

```
  write (#0); {Завершение переназначения}
```

```
End;
```

Str 20 — тип, соответствующий типу String [20];

N_Fun_Key — номер переназначаемой функциональной клавиши (должен быть в пределах 1...10). S — новое назначение функциональной клавиши (если нажата клавиша «ввод», то в строку заносится символ с кодом 13).

Управляющие последовательности могут использоваться и для управления режимами печати. Для установки высококачественного шрифта служит оператор

```
Write (LST, #27, 'M');
```

Для управления шрифтами предназначены следующие последовательности:

1) Выделенный шрифт:

```
#27, 'E' — установка
```

```
#27, 'F' — отмена
```

2) Высококачественный шрифт:

```
#27, 'M' — установка
```

```
#27, 'P' — отмена
```

3) Уплотненный шрифт:

```
#15 — установка
```

```
#18 — отмена
```

4) Двойная печать:

```
#27, 'G' — установка
```

```
#27, 'H' — отмена
```

Для управления плотностью строк по вертикали применяются следующие последовательности: #27, '0' — 1/8 дюйма, #27, '1' — 1/10 дюйма, #27, '2' — 1/6 дюйма, #27, '3' — подача бумаги на n/216 дюйма (0 <= n < 255), #27, 'A', #n — на n/72 дюйма (0 <= n < 85).

Для подачи звуковых сигналов, в частности короткого сигнала частотой 1000 Гц, в программе используется оператор Write (^G); Можно инсценировать «писк» печатающего устройства в момент начала печати с помощью оператора Write (LST, ^G);

Звуковые сигналы нужной тональности и продолжительности задаются с помощью процедуры Sound, основанной на использовании портов 42H, 43H и 61H ПЭВМ. Входные параметры процедуры Sound; Frequency — частота (Гц); Duration — длительность (мс) Procedure Sound (Frequency, Duration: Integer); Var Speaker_OFF, Speaker_ON: byte;

```
R: Real
```

```
Begin
```

```
  Speaker — OFF := Port [$61];
```

```
  Speaker — OFF := (Speaker — OFF div 4) * 4
```

```
  Speaker — ON := Speaker — OFF + 3
```

```
  R := 1.19 / Frequency * 1.0E6;
```

```
  Frequency := Round (R);
```

```
  Port [$43] := 182;
```

```
  Port [$42] := to (Frequency);
```

```
  Port [$42] := hi (Frequency);
```

```
  Port [$61] := Speaker — ON
```

```
  Delay (Duration);
```

```
  Port [$61] := Speaker — OFF;
```

```
End;
```

Для перехода к нотной записи мелодии служат процедура Play и константа Notes:

```
Type NoteRecord = record
```

```
  C, CF, D, DF, E, F, FF, G, GF, A, AF, B: Integer
```

```
end;
```

```
Const Notes: NoteRecord = C:1; CF:2; D:3; DF:4; E:5; F:6; FF:7; G:8; GF:9; A:10; AF:11; B:12;
```

Процедура Play имеет входные параметры Octave — номер октавы; Note — нота (в нотации NoteRecord); Duration — длительность (мс).

```
Procedure Play (Octave, Note, Duration: Integer);
```

```
Var Frequency: Real;
```

```
  I: Integer
```

```
Begin
```

```
  Frequency := 32.625;
```

```
  for i := 1 to Octave do Frequency := Frequency * 2;
```

```
  for i := 1 to Note — 1 do
```

```
    Frequency := Frequency * 1.059463094;
```

```
    Sound (Round (Frequency));
```

```
end;
```

Телефон 361-49-41, Москва, Лопатин Андрей Владимирович

Статья поступила 21.2.88

681.3.06

Е. Ю. Лактионов

ПАКЕТ ПРОГРАММ БАЗА-85 ДЛЯ ПЭВМ

Пакет ориентирован на пользователя-непрофессионала в области обработки данных с помощью ПЭВМ. Он включает достаточно развитые средства ввода, редактирования и поиска данных среднего объема (порядка тысяч и десятков тысяч записей).

База данных представляет собой совокупность отношений в одном каталоге файловой системы: отношение — прямоугольная таблица данных; элементы отношения — колонка, запись, поле данного; типы данных — целое, вещественное (двойной точности), текст (до 256 символов). Все поля колонки имеют одинаковый тип данных. Пакет работает с отношениями на жестком диске. Гибкие диски используются для копирования и хранения отношений.

Основные ограничения: объем отношения — до 400 Кбайт; число записей — до 32 тыс.; длина записи — до 2 Кбайт; число полей в записи (число колонок) — до 32. Число отношений в базе ограничено емкостью жесткого диска.

Экран пакета состоит из следующих областей: меню пакета, рабочая область, строки состояния, редактирования, сообщений. Рабочая область — окно в активное отношение (в строке состояния описано это отношение). Строка редактирования служит для ввода данных и команд, строка сообщений — для выдачи поясняющих и предупреждающих сообщений.

Команды пакета в основном задаются с помощью функциональной клавиатуры и команд меню (только команды сортировки и выборки вводятся через строку редактирования).

Данные вводятся и редактируются в режиме экранного редактирования, т. е. объект операции указывается на экране курсором, а затем задается действие (функциональной клавишей или через меню).

Меню пакета — ниспадающее, одноуровневое. Под клавишные команды задействована почти вся функциональная клавиатура, при этом группы команд организованы в группы клавиш. Имена отношений, значения данных, параметры команд вводятся через строку редактирования.

Для выдачи пользователю справок и пояснений в пакете организована помощь (в некоторой степени контекстно-зависимая). При обращении за помощью выдается подробная информация о работе пакета.

В БАЗЕ-85 предусмотрен также пакетный режим — можно запомнить в файле последовательность команд, а затем многократно выполнять эти команды. Операции можно объединить в следующие группы:

- создание и корректировка схемы отношения;
- ввод и редактирование данных;
- сортировка, поиск и выборка данных;
- оформление данных;
- подготовка к печати и печать;
- работа с каталогом;
- использование пакетного режима.

Схема отношения задается числом колонок и типом каждой колонки. Создание отношения заключается в последовательном формировании всех его колонок и задании типа данных для каждой колонки. Корректировка схемы сводится к удалению, вставке, копированию и перестановке колонок.

Эти функции реализуются с помощью следующего набора операций:

- перемещение курсора на позицию, кадр, границу отношения, указанную запись;
- поиска и замены значения поля внутри колонки;
- вставки, удаления, копирования, перестановки записей;
- редактирования (ввода), копирования значения поля;
- последовательного ввода значений полей записи (колонки).

Числовые поля при вводе проверяются «на цифру». В пакете предусмотрена функция копирования записей из одного отношения в другое, соответствующее первому по числу и типу колонок.

Ключ сортировки может содержать до пяти полей записи отношения. Сортировка проводится по возрастанию (убы-

ванию) каждого из них. Результат операции поиска и выборки — всегда новое отношение, содержащее отобранные данные.

Возможности операций поиска и выборки:

выборка-удаление записей из одного отношения по сложному логическому условию;

выборка-удаление из отношения записей, для которых есть (нет) соответствующие записи в другом отношении (записи с равными значениями полей в указанных колонках);

создание (вычисление) новой колонки (поля) в записи, являющейся результатом выражения над полями записи и константами;

вычисление агрегатных функций (суммы, среднего, максимума, минимума) по всему отношению и группам записей с одинаковым ключом.

Данные на экране оформляются с помощью следующих операций:

- шрифтового выделения колонки (жирного шрифта, подчеркивания);
- изменения ширины колонки;
- выравнивания, центрирования данных;
- задания числа знаков после десятичной точки при выводе.

На печать отношение выводится постранично. При подготовке к печати указываются размеры, заголовок страницы, отступы, порядок нумерации. Вывод на печать можно приостановить, возобновить, прервать.

Список отношений базы данных можно получить в специальном окне каталога. В нем можно переименовывать, копировать, удалять и загружать отношения в режиме непосредственного редактирования.

При работе с пакетом имеется возможность параллельно с выполняемыми действиями запомнить в файле на диске последовательность нажатия клавиш. Этот файл становится своего рода программой для пакета, его можно многократно исполнять. Особенно удобно запоминать и выполнять в пакетном режиме команды выборки, сортировки и целые связки команд, задающие часто повторяющиеся последовательности действий.

БАЗА-85 может передавать данные в пакет ТАБЛИЦА-85, т. е. отношение базы данных может быть прочитано и преобразовано в электронную таблицу пакетом ТАБЛИЦА-85. Обратной совместимости нет, таблица в отношении базы данных не преобразуется.

Совместно с пакетом БАЗА-85 поставляется библиотека, содержащая функции работы с отношениями. Используя эту библиотеку, пользователь может разработать на языке Си свои программы обработки отношений, совместимые с пакетом по данным.

БАЗА-85 работает на ПЭВМ «Электроника МС0585» под управлением ИНМОС-85 версии 1.2 и выше. Тексты программ пакета написаны на языке Си. Пакет поставляется фондом алгоритмов и программ (ВСФАП) при Бердянском филиале ИПИ АН СССР.

Телефон 3-63-83, Бердянск

Статья поступила 21.2.89

УДК 681.32.06

С. Н. Домацкий, И. Л. Шраго

ОРГАНИЗАЦИЯ ПРОГРАММНОЙ СРЕДЫ ПОЛЬЗОВАТЕЛЯ СИСТЕМЫ АВТОМАТИЗАЦИИ ИССЛЕДОВАНИЙ НА БАЗЕ ПЭВМ

Возможности современных ПЭВМ позволяют рассчитывать на получение нового качества в системах автоматизации научных исследований (АНИ). Поскольку в аппаратных средствах, применяющихся для АНИ, качественные сдвиги происходят заметно медленнее, чем в ВТ, основными слагаемыми получения нового качества становятся глубина обработки экспериментальных

данных в ПЭВМ и возможности построения экспертных систем. При этом большой вес имеют процессы интерактивного взаимодействия с экспериментатором и выбор альтернатив. Проверка собственных гипотез — наиболее ценная составляющая работы экспериментатора — часто не укладывается в рамки стандартных алгоритмических схем и методов математиче-

ской обработки измеряемых данных, определяемые коммерческими интерпретированными программными комплексами типа Lab Windows (National Instruments Inc), и вызывает необходимость разработки собственных программ. До последнего времени основным в АНИ считался подход, основанный на планировании последовательного запуска в пределах одного

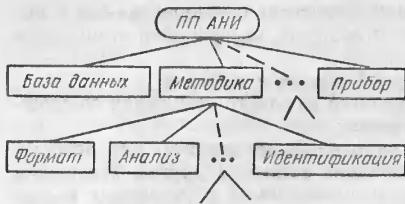


Рис. 1. Структура ПП АНИ

эксперимента разрозненных программ, в общем случае разработанных многими специалистами в различных языковых средах и в разное время. Функции монитора программного обеспечения эксперимента выполнялись самим исследователем. Для ПЭВМ это выглядит дорогостоящим анахронизмом.

Экспериментаторы в основном не являются профессиональными программистами, и вряд ли целесообразно требовать от них перепрограммирования ранее опробованных программ для формирования пакетов с развитыми средствами навигации в программной среде планируемого эксперимента. Это, помимо прочего, требует глубоких знаний ДОС. Правильнее предложить инструмент, позволяющий создавать иерархические интерактивные пакеты из готовых программ, не прибегая к дополнительному программированию. Унифицированный шаблон для создания монитора пользователя на базе ЭВМ СМ3, СМ4 и их аналогов описан, в частности, в ДИАНАД-4 [1].

Мы располагаем результатами разработки программной среды АНИ для ПЭВМ «Искра 1030» и других аналогов IBM PC XT, базирующейся на настраиваемом простом текстовым редактором супервизоре интегрированных многофункциональных пакетов программ (ПП) с иерархической структурой. Использована типовая структура пакета АНИ [2], а также понятие интерфейса пользователя [3]. За основу принят принцип «внешнего воздействия», запрещающий любое вмешательство средств организации пакета в целевые задачи АНИ. Это позволяет использовать готовые программы независимо от языка их реализации и улучшать качество работы с оригинальными программными средствами, интерфейс пользователя которых морально устарел.

Пример древовидного графа современного ПП АНИ показан на рис. 1. В узлах расположены альтернативы, выбираемые пользователем из меню. Это либо исполнительные модули ДОС, либо меню следующего уровня. Важный компонент обеспечения собственной эффективности пакета [3], в особенности при наличии в нем элементов экспертной системы, — существование контекстно-чувствительной системы текстовых подсказок, помогающей ориентироваться в среде ПП.

Итак, для ПП АНИ, строящихся из разрозненных программ, наиболее важны следующие возможности:

многоуровневая организация и древовидная структура; модифицируемость структуры при настройке на эксперимент; равнодоступность задач, режимов и функций всех уровней иерархии ПП; доступность в процессе работы ПП развернутых текстовых подсказок адекватных позиции на графе.

Реализация этих возможностей не затрагивает целевые функции программ и поэтому решается в рамках организующего супервизора. Для ПЭВМ имеются диалоговые интерфейсы, позволяющие объединять автономные программы в пакеты с общим меню. К наиболее распространенным относятся «Окна» (Microsoft), GEM (Digital Research) и TopView (IBM). Они обеспечивают полиэкранный интерфейс и межзадачный обмен. «Окна» поддерживают многозадачную операционную среду. Все эти системы — надстройки над ДОС. Недостаток их для АНИ — отсутствие средств создания иерархического управления экспериментом, большой расход оперативной (100 Кбайт сверх ДОС) и дисковой памяти («Окна» занимают на диске более 420 Кбайт), сложность создания системы подсказок, необходимость использования спе-

циального инструментария (библиотек, редакторов связей и т. д.), что является жестким требованием при создании целевых задач экспериментов и затрудняет использование старых программ. Возникают также вопросы правовой защиты ПП. Несмотря на эти недостатки при запуске задач в рамках единого пакета производительность труда пользователя повышается благодаря эффекту организованности ресурсов.

Разработанная авторами диалоговая Си-программа ФОРМИРОВАТЕЛЬ ДЕРЕВЬЕВ (ФОРМИРОВАТЕЛЬ) (рис. 2) лишена отмеченных недостатков. Программа написана под компилятор Microsoft C версии 3.0 и выше или Си ЛНПО Электронмаш. Она обеспечивает организацию пакетов программ с древовидной иерархической структурой и сопровождение их развернутой текстовой подсказкой посредством внешнего воздействия на целевые задачи пакета, подготовленные в различных языковых средах разными разработчиками, и требует значительно меньших расходов оперативной и дисковой памяти.

В качестве средства навигации по уровням иерархии использовано горни-

```

main() /* супервизор иерархических пакетов программ */
{
  int i, rc;
  FILE *fp;

  /* запросить память под первый (Корень дерева) элемент спуска меню */
  current_level = (MENU *) malloc(sizeof(*priv));
  /* нулевой указатель на предыдущий элемент - признак первого элемента */
  current_level->pointer = NULLPTR;
  /* предопределенное имя первого (Корневого) меню */
  strcpy(current_level->name, "формирователь");
  /* предопределенное имя файла корневого меню */
  strcpy(current_level->path, "t_former");

  /* загрузка Корневого меню
  в элемент спуска в ОЗУ ПЭВМ
  из файла T_FORMER.SET */
  menu_load();
  menu_level = 0; /* номер текущего уровня иерархии */

  /* перехват вектора прерываний от клавиатуры */
  get_keyboard();

  while(menu_level >= 0) /* пока номер уровня не отрицателен */
  {
    rc = play_menu(); /* навигация среди альтернатив текущего меню */
    if (rc == RC_ESC)
    {
      /* Возврат в меню предыдущего уровня */
      free_menu(); /* изменить указатель меню, освободить память */
      menu_level--;
      continue;
    }
    if (rc == RC_ENTER)
      switch (test_choice()) /* меню или задача? */
      {
        case TASK:
          run_task(); /* порождение подпроцесса целевой задачи ПП */
          break;
        case MENU:
          build_menu(); /* создать следующий элемент спуска меню */
          menu_load(); /* загрузить файл меню */
          menu_level++;
        }
      }
  }
  /* восстановление векторов прерываний,
  очищение экрана монитора
  выполнение Командного файла T_FORMER.BAT
  Возврат управления ДОС */
}
  
```

Рис. 2. «Скелет» программы ФОРМИРОВАТЕЛЬ ДЕРЕВЬЕВ

зонтальное «выпрыгивающее» (pop — op) меню с быстрой подсказкой и с применением принципов организации экрана монитора ПЭВМ из [2]. Для сокращения объема резидентной программы и упрощения организации перехода с уровня на уровень иерархии в памяти ПЭВМ хранится не весь древовидный граф описания структуры, а лишь ценный список уровней альтернатив пройденного по нему пути (т. е. альтернативы текущего уровня иерархии и все непосредственно следующие). Описания структур для каждого уровня иерархии пакета удобно хранить в виде отдельных файлов с расширением SET, состоящих из наборов троек символьных записей. Файлы меню, настраивающие ФОРМИРОВАТЕЛЬ на структуру (см. рис. 1), имеют следующий вид:

Уровень задач (файл T_FORMER.SET).

a maintit.ttl
База данных
Сопровождение наборов данных
fsio
Методика
Задачи обработки и анализа
research
...
Прибор
Управление прибором

имя файла заставки ПП
первая альтернатива в меню
первая краткая подсказка
первый вызов программы в АДОС
вторая альтернатива подуровня
вторая краткая подсказка
файл меню следующего уровня
прочие альтернативы
К-я альтернатива в меню
К-я краткая подсказка
К-й вызов программы в АДОС

Уровень режимов (файл research.set)

a research.ttl
Формат
Преобразование форматов
convert
...
Анализ
Формирование классов
class
...
Идентификация
Определение принадлежности
percy

файл заставки узла «Методика»
первая альтернатива в меню
первая краткая подсказка
первый вызов программы в АДОС
первый вызов программы в АДОС
вторая альтернатива в меню
вторая краткая подсказка
вторая вызов программы в АДОС
прочие альтернативы
К-я альтернатива в меню
К-я краткая подсказка
К-й вызов программы в АДОС

В основе алгоритма навигации по уровням иерархии ПП и создания иерархии взаимосвязи файлов описания меню лежит соглашение о совпадении полного имени файла меню подуровня с текстом вызова программы в АДОС в третьей строке тройки альтернативы. Си-описания меню уровня и текст процедуры загрузки меню в ОЗУ во время работы супервизора приведены на рис. 3.

Для управления экраном при организации меню можно использовать процедуры, приведенные в [2], с учетом замечаний по интерфейсу с языковыми средами. В момент запуска ФОРМИРОВАТЕЛЬ резервирует память под первый элемент (описание обратносвязного ценного списка уровней меню) и загружает в него файл главного меню пакета с фиксированным именем T_FORMER.SET (процедура menu_load), затем этот элемент используется в процедуре play_menu выбора альтернативы. Экспери-

ментатор, манипулируя клавишами управления курсором и ENTER, организует поиск по предоставляемым в меню альтернативам, среди которых могут быть альтернативы смены уровня иерархии, или возвращается на предыдущий уровень по нажатию клавиши ESC. В случае возврата на предыдущий уровень иерархии память, отведенная под элемент-описание меню текущего уровня, освобождается процедурой free_menu (рис. 4).

При выборе меню следующего уровня функцией test_choice повторяются действия по резервированию памяти и загрузке файла меню. Если выбранная альтернатива — задача, режим или функция пакета, о чем свидетельствует расширение COM или EXE, то она запускается на выполнение функцией run_task как порождаемый процесс

менты этого списка (обеспечивают последовательный доступ к экранам подсказки вперед, назад, прямой доступ к первому и последнему экранам. По завершении работы целевой программы ПП память, занятая подсказкой, освобождается и возвращается АДОС, а рабочий экран целевой программы восстанавливается. Восстановление экрана прерванной задачи обеспечивается во всех режимах контроллера цветного графического монитора. Таким образом достигается независимое от программ пакета, но согласованное с его целевыми задачами, полное информационное обеспечение.

Для создания пакета на базе ФОРМИРОВАТЕЛЯ необходимо иметь набор программ реализации функций эксперимента в виде исполняемых файлов АДОС с расширением EXE или COM и любой редактор текстов; текстовым редактором создать набор файлов меню уровней с расширением SET, начиная с T_FORMER.SET (аналогично приведенному выше примеру); этим же редактором создать, если требуется, файлы текстов оперативного информационного обеспечения задач пакета с именами программ пакета и расширением HLP. По завершении этих действий пакет готов к исполнению. Никаких других операций не требуется. Не накладывается ограничений и на программы — функции эксперимента. Их меню может быть построено любым способом, таким например, как в [1]. Для сохранения единого стиля авторы пользовались в своих программах экранной библиотекой из [2].

Для обеспечения большей гибкости ПП АНИ перед окончательным выходом в ДОС для выполнения операций общесистемного характера, например стирания промежуточных файлов или копирования файлов результатов, предусмотрена возможность вызова командного файла T_FORMER.BAT. Кроме того, с учетом изложенных в [2] соображений об организации экрана монитора в супервизоре реализованы следующие возможности:

вывод на экран монитора файла заставки — визитной карточки меню уровня;

создание перед запуском целевой программы и именованной рамки в верхней части экрана 2 (рис. 2, 3);

фиксация экрана до получения подтверждения перед возвратом в меню уровня для изучения данных, формируемых программами на экране монитора по завершении работы.

Организуемая программа, построенная на основе структуры, приведенной на рис. 2, в авторской версии допускает до 30 альтернатив на каждом уровне иерархии. Их число ограничено размерами экрана и соображениями об интерфейсе пользователя, а число уровней иерархии в ПП — размером свободной динамической памяти, предоставляемой исполняющей библиотекой Си.

```

}
/* описание типа элемента списка иерархии меню */
typedef struct menu
{
    struct menu *pointer; /* указатель на предыдущий элемент списка */
    char name[COMLENGTH]; /* имя меню */
    char path[70]; /* полное имя файла меню */
    char *title; /* имя визитной карточки меню */
    char *cmd[31]; /* имена альтернатив */
    char *file[30]; /* команды DOS, соответствующие альтернативам */
    char *expl[30]; /* поясняющие сообщения к альтернативам */
}MENU;
/* указатели на текущий и предыдущий элементы списка меню */
MENU *current_level, *priv;

int menu_load() /* процедура загрузки файла меню */
{
    int i;
    /* формирование имени файла меню */
    strcpy(buf, current_level->path);
    strcat(buf, ".set");

    if ((fp=fopen(buf, "r"))==NULL) /* открыть файл меню */
        return(RC_FAIL); /* файл меню не открыт, возврат ошибки */

    fgets(buf, COMLENGTH, fp); /* анализ первой строки файла меню на */
    if (buf[0]!='=') /* признак наличия заставки */
    {
        buf[strlen(buf)-1]=EOS; /* формирование имени */
        current_level->title=malloc(strlen(buf)); /* файла */
        strcpy(current_level->title, &buf[1]); /* заставку */
    }
    else
    {
        rewind(fp); /* позиционирование на начало файла */
        current_level->title=NULL; /* нет заставки */
    }
    i=0;
    while (fgets(buf, COMLENGTH, fp)!=NULL) /* чтение имени альтернатив */
    {
        /* запись имени альтернативы */
        buf[strlen(buf)-1]=0;
        current_level->cmd[i]=malloc(strlen(buf)+1);
        strcpy(current_level->cmd[i], buf);
        /* чтение и запись строки краткой подсказки */
        fgets(buf, EXPLENGTH, fp);
        buf[strlen(buf)-1]=0;
        current_level->expl[i]=malloc(strlen(buf)+1);
        strcpy(current_level->expl[i], buf);
        /* чтение и запись строки команды DOS, соответствующей альтернативе */
        fgets(buf, PROGLENGTH, fp);
        buf[strlen(buf)-1]=0;
        current_level->file[i]=malloc(strlen(buf)+1);
        strcpy(current_level->file[i], buf);
        i++; /* счетчик числа альтернатив меню */
    }
    current_level->cmd[i]=0; /* признак конца списка альтернатив */
    fclose(fp); /* закрыть файл меню */
    return(RC_SUCCESS); /* возврат из процедуры */
}

```

Рис. 3. Меню уровня и процедура его загрузки в ОЗУ

```

/* процедура уничтожения элемента списка меню */
free_menu()
{
    int i;
    i=0;
    while (current_level->cmd[i] !=NULLPTR) /* цикл по числу альтернатив меню */
    {
        free(current_level->cmd[i]); /* освободить память имени альтернативы */
        free(current_level->expl[i]); /* освободить память подсказки */
        free(current_level->file[i++]); /* освободить память команды DOS */
    }
    priv=current_level; /* сохр. указатель на текущий элемент меню */
    /* заменить указатель текущего элемента списка на предыдущий */
    current_level=current_level->pointer;
    free(priv->title); /* освободить память имени файла заставки */
    free(priv); /* освободить память старого элемента списка */
}

```

Рис. 4. Процедура free—menu

С применением ФОРМИРОВАТЕЛЯ создан ряд простых и удобных пакетов для различных предметных областей. Супервизор T__FORMER предназначен для операционной среды АДОС 3.3 на ПЭВМ «Искра 1030», но может использоваться на любом аналоге IBM PC в среде MS DOS, начиная с версии 2.0. На дискете он занимает 29 Кбайт, а в ОЗУ ПЭВМ — 32 Кбайт, в которые входит память, динамически распределяемая под описание меню уровней. Принцип внешнего управления позволяет реализовать программы, организуемые ПП АНИ практически в любой операционной среде.

Открытая текстовая настройка и основные принципы ФОРМИРОВАТЕЛЯ позволяют вызывать программы, изменяющие набор и внутреннее содержание файлов — описателей уровней иерархии (* .ET) в соответствии с информационной обстановкой в среде ПП, заложенными в них целями и желанием пользователя. Это дает возможность формировать ПП АНИ, оптимизирующие свою структуру во время эксперимента. Авторы видят в использовании такого свойства интересные перспективы для построения саморегулирующейся программной среды пользователя и создания экспертных систем в ряде областей знаний.

Телефон 252-69-89, Ленинград

ЛИТЕРАТУРА

1. Громов Г. Р. и др. ДИАНЭД-4 — инструмент для создания диалоговых систем. — Пушкино: НЦБИ АН СССР, 1981.
2. Домарацкий С. Н., Шраго И. Л. Организация экрана монитора ПЭВМ «Искра 1030» в программах автоматизации научных исследований // Микропроцессорные средства и системы. — 1988. — № 5.
3. James B. The User Interface // The Computer J.— The British Computer Society, London.— 1980.— V. 23.— N 1.— P. 25—29.

Статья поступила 8.02.89

РЕКЛАМА

Внимание руководителей государственных предприятий, СП, сервисных организаций, кооперативов!

ПО «Орловский завод УВМ им. К. Н. Руднева» освоил в производстве тестер цифровых и аналоговых блоков КД-1, осуществляющий поиск неисправностей в аналого-цифровых блоках и блоках ОЗУ, ПЗУ с комбинированной логикой, а также проверку аналого-цифровых узлов. Тестер КД-1 как средство измерения включен в Госреестр СССР.

Прибор имеет 288 логически программируемых дискретных входов-выходов или 84 и 64 дискретных входа-выхода.

Частота задания воздействий — от 10 до 10000 кГц. Видеоконтроль — функциональный и параметрический.

Среднее время проверки — 2 мин. Имеется язык высокого уровня для написания тестов проверки блоков.

Тестер комплектуется стандартным стыковочным узлом для подключения плат типа E2, контролирует электронные блоки персональных ЭВМ отечественного и импортного производства, блоки станков с ЧПУ.

Объединение принимает заказы по отдельному договору на разработку стыковочных узлов, программ проверки блоков и их ремонта.

Адрес: 302025, г. Орел, т. а. Нейрон 148216 и 148234; Телефон: 3-06-02.

ЭЛЕКТРОННЫЙ ДИСК НА БАЗЕ КГД ДВКЗ

В состав комплекса ДВКЗ входит контроллер графического дисплея (КГД), имеющий для хранения изображения оперативную память объемом 16 Кбайт. В программах, не работающих с графическим экраном, можно использовать эту память для промежуточного хранения данных. Непосредственное обращение к графической памяти через регистры контроллера делает вновь разрабатываемую программу зависимой от оборудования. Такие программы неработоспособны в отсутствие КГД.

Более гибкий способ использования памяти КГД — организация на ее основе электронного диска. В этом случае графическая память становится доступной многим пользователям и системным программам общего назначения, входящим в ОС ДВК.

Простейший драйвер такого диска имитирует носитель стандартной файловой структуры объемом в 32 блока по 512 байт на базе графической памяти. Однако даже при минимальном размере оглавления носителя восемь начальных блоков будут заняты служебной информацией, которая для электронного диска нужна в значительно меньшем объеме или совсем не нужна (например, области первичного и вторичного системных загрузчиков, резервные поля).

Ниже описан драйвер электронного диска GM (Graphic Memory), который, с точки зрения пользователя, работает как драйвер обычного диска, а фактически использует для хранения служебной информации лишь один блок. Остальная память КДГ (31 блок) отводится для записи файлов. Реализация драйвера потребовала разработки алгоритма сжатия служебной

информации. Соответствие логических адресов драйвера GM физическим адресам оперативной памяти КГД показано в таблице.

В приведенной таблице понятие «фиктивный адрес» обозначает, что при записи по нему информация теряется, а при чтении всегда считывается нулевой байт. Размер оглавления позволяет иметь записи о 27 файлах, что вполне достаточно для электронного диска с таким объемом памяти.

Для размещения служебной информации выбраны старшие адреса ОЗУ КГД, так как эта область памяти не отображается на графическом экране (последний из отображаемых адресов — 33733). Данное обстоятельство позволяет изменять драйвер GM для копирования изображения из оперативной памяти КДГ на внешний носитель и обратно. Например, если на экране какими-то средствами построено графическое изображение, то его можно сохранить на гибком диске (MX1), выполнив следующую последовательность команд ОС ДВК:

```
COPY / NOQUERY / DEVICE / PILE GM: /
START: 8 / END:35 PRIMER.PIC
```

Для восстановления изображения из файла MX1:PRIMER.PIC используются следующие команды:

```
.INIT/NOQUERY/SEGMENT:1GM:
.COPY PRIMER.PIC GM:
```

Для удобства работы с алфавитно-цифровым и графическим экранами дисплея дополнительно в драйвер GM введены SET-параметры: PIC/NOPIС и ТЕХТ/NOTЕХТ. Например, после копирования изображения можно отключить алфавитно-цифровую и открыть графический экран, выполнив команду:

```
.SET GM NOTЕХТ PIC
```

При использовании электронного диска для хранения файлов необходимо помнить, что при выключении питания ДВКЗ информация, хранящаяся на нем, будет потеряна. Перед первым использованием после включения питания диск необходимо инициализировать, как показано в примерах. В остальном работа с ним не отличается от работы с обычными дисками. Например, можно выполнить команду ОС ДВК DIRECTORY GM: и получить содержимое каталога. Быстродействие и надежность (в смысле износоустойчивости) у электронного диска выше.

Драйвер должен храниться на системном диске в файле GM.SVS, занимающем два блока. В работе драйвера есть некоторые ограничения, обычно несущественные в стандартных применениях:

GM нельзя инициализировать с числом сегментов каталога более одного;

число задействованных позиций в каталоге не должно превышать 27 (возможные свободные «окна» между файлами, входят в число задействованных позиций каталога);

Номер блока GM	Адреса в блоке GM	Физические адреса ОЗУ КГД	Назначение области
0	Все	Фиктивный адрес	Первичный системный загрузчик
1	0...203	»	Таблица замены дефектных блоков
	204...251	37 000...37 045	Используется программой DUP.SAV
	252...273	37 046...37 067	Используется программой BUP.SAV
	274...677	Фиктивный адрес	Системой не используется
	700...777	37 070...37 167	Общая информация о томе
2...5	Все	Фиктивный адрес	Вторичный системный загрузчик
6	0...607	37 170...37 777	Первый сегмент каталога (для GM он всегда единственный)
7	Все	Фиктивный адрес	»
8...38	»	0...36 777	Содержимое хранимых файлов

если для GM используется нестандартная (расширенная) структура каталога, то одновременно с увеличением длины одной позиции максимально уменьшается допустимое число позиций, задействованных в каталоге;

прикладные программы не должны пользоваться на физическом уровне старшими адресами памяти КГД.

249020, Калужская обл., Обнинск, ул. Курчатова, 21, ЦИПК; тел. 2-53-37.

Статья поступила 27.04.88

УДК 681.3

Н. Н. Щелкунов, А. П. Дианов

ОРГАНИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОНТРОЛЛЕРОВ ГМД

Хорошо разработанная аппаратура не может быть с успехом использована в микропроцессорных системах (микросистемах), если для нее не создано соответствующее программное обеспечение (ПО). Предлагается ПО *mDISK* для внутриплатного 6-кристального контроллера гибкого магнитного диска (ГМД) *mMC5301* [1] для гибких мини-дисков с одинарной (FM) или удвоенной (MFM) плотностью записи, который построен на базе БИС *i8272A* и ориентирован на системы типа *mMC* [2]. Пакет *mDISK* — основа базовой системы ввода-вывода (BIOS) новой одноплатной микроЭВМ *mMC1207*, совместной с *DOC1810*. Он обеспечивает хороший компромисс между нуждами системного и прикладного программистов, с одной стороны, и возможностями контроллера ГМД, с другой.

ПО контроллера ГМД построено в виде многоуровневой системы виртуальных устройств с четко определенными программными интерфейсами. Понятие виртуального устройства использовано для построения системы ввода-вывода (ВВ) монитора *mMON1200* [3], входящего в состав встроенного ПО одноплатных микроЭВМ *mMC1204...mMC1206* [2]. В системе определены три логических и пять физических устройств ВВ поточного типа, однако их строение во много раз проще, чем организация виртуальных устройств связи с ГМД.

Системный интерфейс контроллера ГМД

Ядро микросистемы (МС), состоящее из центрального процессора (ЦП) и основной памяти, соединяется с аппаратурой периферийного устройства (ПУ) рядом отдельных линий и портов ВВ, образующих системный интерфейс их взаимодействия. Интерфейс контроллера ГМД *mMC5301* (рис. 1) состоит из 8-разрядного двунаправленного порта данных DR

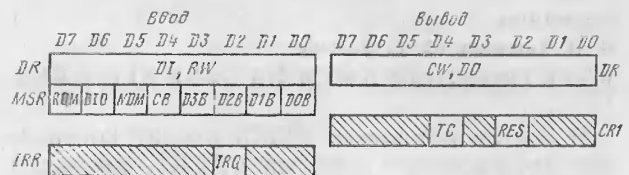


Рис. 1. Системный интерфейс контроллера *mMC5301*

для вывода команд *CW*, ввода-вывода данных *DI*, *DO* и ввода ответа *RW*; входного 8-разрядного порта основного состояния *MSR* и двух линий управления *RES* (сбор контроллера ГМД) и *TC* (конец операции чтения-записи), определенных во внешнем по отношению к контроллеру ГМД выходном порту *CR1*, а также подключенной к одному из входов контроллера прерываний *KP580BH59* (*K1810BH59A*) линии радиального запроса на прерывание *IRQ* (начало фазы ответа) [1].

Следует отметить, что однокристалльный контроллер ГМД *i8272A* поддерживает также операцию ВВ данных в фазе исполнения команды по каналу прямого доступа к памяти (ПДП), однако из-за отсутствия встроенного контроллера ПДП в базовой модели одноплатных микроЭВМ семейства *mMC* [2] в данной версии контроллера ГМД эта возможность не используется. Отказ от контроллера ПДП снижает стоимость аппаратуры, что важно для недорогих одноплатных МС (например, серии *mMC*). В этом случае начало ВВ очередного элемента осуществляется по сигналу радиального запроса на прерывание *IRQ*.

При обслуживании ВВ по прерываниям ЦП во время фазы исполнения команды ГМД должен выполнять очередную операцию ввода или вывода данных за 64 (FM) или 32 (MFM) мкс в случае работы с мини-дисками. ЦП на базе БИС серии *KP580*, работающему на частоте 2 МГц, этого времени недостаточно; предусматривается вариант программно-управляемого условного ВВ с занятием цикла (очередная операция ввода или вывода данных инициируется программным способом). Для этого программа должна непрерывно опрашивать состояние флажка запроса *RQM* из регистра основного состояния *MSR*, доступного для чтения в любое время. При работе контроллера ГМД в режиме, отличном от ПДП (*MSR:NDM=1*), во время фазы исполнения команды (*MSR:CB=1*) этот флажок в точности повторяет поведение сигнала *IRQ*.

Контроллер ГМД всегда отмечает начало фазы ответа установкой запроса на прерывание *IRQ*, который может быть воспринят либо через систему прерываний МС, либо с помощью программного опроса регистра запросов *IRR* контроллера прерываний *KP580BH59* (*K1810BH59A*) и выделения флажка *IRQ*. Ори-

ентация на второй способ приводит к простому программно-управляемому обслуживанию контроллера ГМД без использования каналов ПДП и системы прерываний. Именно такой способ обслуживания был использован в драйвере FD1207, разработанном для управления контроллера ГМД мМС5301 при его работе с 8-разрядными одноплатными микроЭВМ мМС1200, построенными на базе микропроцессора (МП) КР580ВМ80А. Этот драйвер вошел в качестве составной части «защитой» в ПЗУ базовой системы ВВ новой одноплатной микроЭВМ мМС1207 со встроенным контроллером ГМД мМС5301.

Драйвер ГМД принимает вызовы, требующие выполнения операций с контроллером ГМД от программ более высокого уровня и управляет их непосредственным выполнением (рис. 2). Обработка вызова должна оканчиваться возвратом в главную программу кода завершения, информирующего ее об удачном или неудачном окончании вызова (если надо, то с указанием причины неудачи). Набор вызовов драйвера ГМД, использованный в пакете mDISK, — хороший пример их типового состава (табл. 1).

Предполагается, что код возврата передается через заранее определенную ячейку памяти, в качестве которой в драйверах mDISK используется аккумулятор МП КР580ВМ80А.

Определены следующие значения кода возврата:

00H	Нормальное завершение
40H	Ошибка аппаратуры
53H	Ошибка завершения команды контроллера ГМД

Таблица 1

Вызовы драйвера ГМД

Вызов	Код	Функциональное назначение
RES	0	Сброс или инициализация контроллера ГМД при подаче программно-управляемого-импульса RES через порт CR
SEEK (H, C)	1	Позиционирование или установка головки стороны H привода ГМД на цилиндр C
FRM (H, C, D, I)	2	Форматирование или физическая разметка дорожки, расположенной на С цилиндре со стороны H с заполнением секторов байтом D (фактор смещения равен I)
REC	3	Рекалибровка или установка головок привода ГМД на 0-дорожку
RD (H, C, R, BUF)	5	Чтение сектора R цилиндра C стороны H в буфер BUF
WR (H, C, R, BUF)	6	Запись сектора R цилиндра C стороны H из буфера BUF

Анализ слов состояния контроллера ST0...ST3 [1] позволяет сформировать уточненный код ошибки. Например, в ДОС1810 [4] для этой цели зарезервированы коды:

01H	Удаленная запись
02H	Ошибка CRC в поле данных
03H	Неправильный адресный маркер
04H	Ошибка поиска
08H	Ошибка адреса
0AH	Ошибка CRC в поле идентификатора
0FH	Неправильные данные в адресном маркере
10H	Ошибка формата
20H	Попытка записи на защищенный диск
80H	Устройство не готово к работе

Таблица 2

Параметр	Функциональное назначение	Числовое значение или характеристика, определяющая его
DTL	Размер сектора всегда кратен 128 байт	0
DS	Выбор привода и диска на нем	Линия DRIVE SELECT, к которой подключен привод
EOT	Число секторов на дорожке	Формат разметки диска в соответствии с табл. 3
GPL	Длина промежутка GAP3	Тип привода и формат разметки диска (значение GPL для операций чтения-записи (GPLRW) и форматирования (GPLF) разные)
HDS	Выбор головки	H
HLT	Время прижатия головки	Тип привода
HUT	Время поднятия головки	Тип привода
MFM	Плотность записи	Формат разметки по табл. 3
MT	Переход через границу дорожки запрещен	0
N	Размер сектора	Формат разметки по табл. 3
ND	Режим, отличный от ПДП	1
SC	Число секторов на дорожке при форматировании	EOT
SK	Пропуск адресных меток удаленных данных запрещен	0
SRT	Скорость шага	Тип привода
STP	Фактор смещения при сравнении данных	Не используется

При необходимости указатель кода возврата может быть введен в состав параметров вызовов. В вызовах FRM, RD и WR предусмотрено встроенное предварительное позиционирование (для mDISK только в FRM). Вызовы драйвера

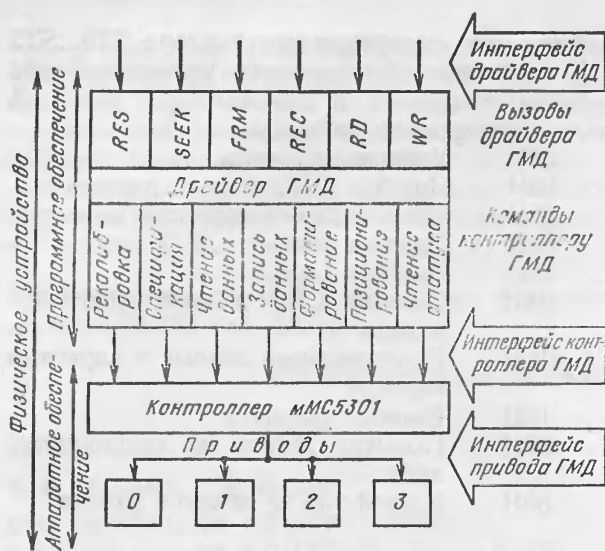


Рис. 2. Организация физического устройства внешней памяти на ГМД

ГМД транслируются в последовательности команд контроллера ГМД. Для этого определяется ряд дополнительных параметров, значения которых зависят от режима работы контроллера, типа привода и формата физической разметки диска. Состав этих параметров для контроллера ММС5301, а также правила их определения приведены в табл. 2. Полное определение всех параметров, команд и форматов контроллера ГМД дано в [1]. Следует отметить, что для реализации пяти основных вызовов драйвера ГМД требуется только семь команд из 15: рекалибровка, спецификация, позиционирование, форматирование, чтение и запись данных, чтение состояния прерывания.

Блок описания физических устройств

Драйвер ГМД со стороны вызывающих его программ рассматривается как некоторое виртуальное устройство внешней памяти, взаимодействие с которым реализуется через пять

строго определенных вызовов. Это устройство обычно называют физическим. Оно характеризуется рядом параметров, не изменяемых во время работы МС, которые жестко привязаны к конкретному типу привода и формату разметки диска. В ряде случаев это может оказаться неприемлемым, так как изменение типа привода или формата разметки диска приводит к перетрансляции драйвера ГМД.

Для повышения гибкости системы некоторые из параметров должны быть динамически изменяемыми, их значение может быть переопределено во время работы МС программным способом. Обычно параметры упаковывают в единый информационный блок описания устройства DUIB (табл. 4).

Данный DUIB описывает устройство для управления приводом с двусторонними (HEAD=2) 80-цилиндровыми (CYL=80) 133 мм ГМД с удвоенной плотностью записи (MFM=1) и 16 (EOT=16) 256-байтовыми (N=1) секторами на дорожке (табл. 3). Предполагается, что привод подключен к линии DRIVE SELECT 0 (DS=0). Временные параметры соответствуют приводам серии TM64 фирмы TANDOM.

В состав информационного блока кроме параметров команд контроллера ГМД входит ряд дополнительных взаимозависимых параметров, таких как HEAD, CYL, BYTES, BD, BT, TD и SEKTL, определяющих логическую организацию диска. Они позволяют проверить корректность адреса (H, C, R) в вызовах драйвера и, в случае ошибки, сформировать код возврата со значением 08H.

Программный интерфейс физических устройств внешней памяти

В системе может быть определено несколько DUIB и драйверов физических устройств к ним. Связь DUIB с устройством осуществляет дополнительный параметр кода устройств TYP, который служит индексом таблицы указателей драйверов DTBL (рис. 3). Эта таблица содержит

Таблица 3

Формат разметки ГМД

		203 мм						133 мм						89 мм		
FM	N	0	1	2	3	4	5	0	0	1	2	3	4	0	1	2
	EOT	26	15	8	3	2	1	18	16	8	4	2	1	15	9	5
	BLT	26	30	32	32	32	32	18	16	16	16	16	16	15	18	20
MFM	N	1	2	3	4	5	6	1	1	2	3	4	5	1	2	3
	EOT	26	15	8	4	2	1	18	16	8	4	2	1	15	9	5
	BLT	52	60	64	64	64	64	36	32	32	32	32	32	30	36	40

Примечания: FM и MFM — одинарная и двойная плотности записи; N — размер сектора (128×2^N байт); EOT — число секторов на дорожке; BLT — число 128-байтных блоков на дорожке.

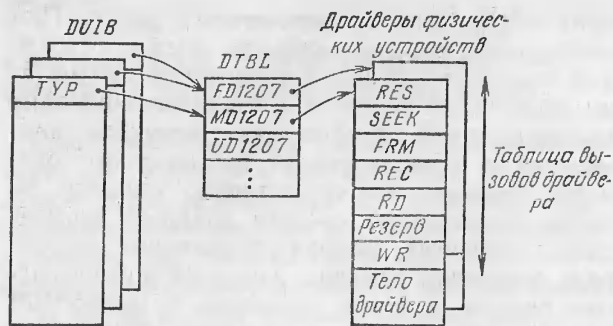


Рис. 3. Организация интерфейса с драйверами физических устройств внешней памяти

указатели на все устройства, используемые в данный момент времени в системе. Код 0 резервируется для фиксации факта отсутствия соответствующей программы управления устройством в системе. Программы управления начинаются со стандартной таблицы указателей стартовых точек всех вызовов.

Таблица 4

Имя	Длина, байт	Значение	Назначение
TYP	1	3	Код устройства
DS	1	0	Номер привода
—	1	0	Резерв (код привода)
—	1	0	Резерв (число фиксированных головок)
HEAD	1	2	Число головок (число сменных головок)
CYL	2	80	Число цилиндров на диске
EOT	1	16	Число секторов на дорожке
BYTES	2	256	Число байтов в секторе
—	1	0	Резерв (число альтернативных цилиндров)
—	1	0	Резерв (номер начального цилиндра)
BD	1	5120	Число блоков на диске
BT	2	32	Число блоков на дорожке
TD	2	160	Число дорожек на диске
SEKTL	1	8	Длина номера блока в секторе
DIRL	2	248	Число записей в директории
—	1	0	Резерв (разметка дорожки 0)
—	1	0	Резерв (разметка дорожки 1)
—	1	0	Резерв (разметка дорожек)
—	1	0	Резерв (смещение первого сектора)
—	1	0	Резерв (признак диска 203 мм)
MFM	1	1	Плотность записи
N	1	1	Размер сектора
GPLRW	1	20H	GPL для чтения-записи
GPLF	1	30H	GPL для форматирования
SRTHUT	1	0FFH	SRT, HUT — четыре старших и четыре младших разряда соответственно
HLT	1	31H	Время прижатия головки
—	1	0	Резерв (флаг включения мотора)

Таблица указателей DTBL, индексируемая параметром DUIB:TYP, совместно с таблицей стартовых точек вызовов позволяют изолировать вышестоящее ПО от абсолютных адресов размещения драйверов физических устройств, их числа и конкретной реализации. В качестве последних могут выступать не только драйверы ГМД, например FD1207 (код 3), но и программа управления ОЗУ-диском MD1207 (код 2) или любой другой драйвер пользователя UD1207 (код 7), удовлетворяющий интерфейсным требованиям на его организацию и вызов. В пакете mDISK определены также программы для управления ПЗУ-диском PD1207 (код 1) и жестким магнитным диском типа «винчестер» WD1207 (код 4). Возможен драйвер дистанционного доступа к системе внешней памяти, взаимодействующей с МС через гибкую связь или локальную сеть. При этом информационный блок DUIB — единственный элемент, связывающий эти программы с ПО.

Требование полной изоляции ПО от конкретных характеристик физического устройства приводит к понятию логического устройства. Например, файловая система ДОС1810 оперирует с логическими дисками, состоящими из ряда дорожек со 128-байтными секторами, называемыми блоками. Для адресации логической дорожки (LT) и блока (LB) используются 8-рядные указатели, поэтому максимальная информационная емкость диска составляет 256 дорожек по 256 блоков или 8 Мбайт.

Вызовы драйверов логического (табл. 5) и физического дисков аналогичны. Команды LRD и LWR предусматривают не только многоблочную пересылку (параметр NB), но и встроенное позиционирование. При возникновении ошибок вызовы повторяются до пяти раз, после чего код ошибки возвращается в главную программу.

Драйвер логического диска пересчитывает логический адрес (LT, LB) в физический (H, C, R, B) по схеме рис. 4. Размеры полей C, H, R и B меняются в зависимости от числа цилиндров (параметр CYL), типа физического диска (HEAD), числа секторов на дорожке (EOT) и размера сектора (N), значения которых определяются в DUIB. При CYL=77, HEAD=1, EOT=2 и N=0 (128-байтный сектор) копируется физический диск ДОС1800, в

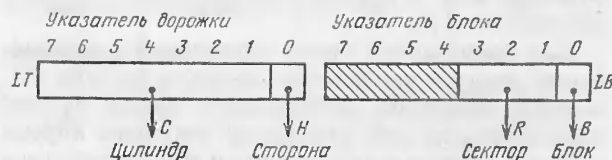


Рис. 4. Схема преобразования логического адреса диска в физический:

B — указатель положения блока внутри физического сектора

Таблица 5
Вызовы драйвера логического диска

Вызов	Код	Функциональное назначение
LRES (FX)	0	Сброс или инициализация устройства FX
LSEEK (FX, LT)	1	Позиционирование или установка устройства FX на дорожку LT
LFRM (FX, LT, D, I)	2	Форматирование дорожки LT устройства FX с заполнением ее байтом D. Фактор смещения равен 1
LREC (FX)	3	Рекальбровка или установка головок устройства FX на 0-дорожку
LRD (FX, LT, LB, NB, PTR)	4	Чтение с устройства FX (LT, LB) NB блоков с их пересылкой в буфер пользователя PTR
LWR (FX, LT, LB, NB, PTR)	6	Запись из буфера пользователя PTR на устройство FX по адресу (LT, LB) NB блоков
LSET (FX, PTR)	7	Связь информационной таблицы DUIB из буфера PTR с устройством FX
LCHK (FX, PTR)	8	Чтение DUIB устройства FX в память по адресу PTR
LBRES	9	Сброс буферов драйвера логического диска

котором отсутствуют средства динамического изменения параметров разметки диска, а структура логических дисков в точности соответствует структуре физических. Попытка применения другой схемы пересчета логического адреса диска в физический может привести к невыполнению операции форматирования.

При связывании логического диска DOS1810 с физическим при $N \geq 1$ возникает задача сборки и разборки сектора физического диска, которая решается с помощью вспомогательного буфера обмена BUF. Размер буфера выбирается равным максимально допустимому в системе размеру сектора физического диска (для MDISK — 256 байт). Сектор C, H, R сначала копируется в буфер, выполняется операция чтения или записи блока по адресу B и затем, при необходимости, содержимое буфера записывается на физический диск по своему же адресу. При этом управление обменом между буфером BUF и диском возлагается на драйвер физического диска.

Для сокращения числа обращений к физическому диску и увеличения скорости работы МС вызовы драйвера физического диска выполняются только при изменении текущего адреса сектора. Обращения к одному и тому же сектору приводят к многократному доступу к буферу обмена и только однократному (операция чтения) или двукратному (операция модифика-

ции) обращению к физическому диску. При необходимости размер области памяти, отводимой под обмен данными с физическими дисками, может быть увеличен, а число одновременно хранимых в ней секторов может меняться динамически в зависимости от их истинного размера. Конечно, это существенно увеличит ту часть программных средств драйвера логического устройства, которая ответственна за контроль состояния буферов. Хорошим компромиссом служит решение, принятое в микроЭВМ CM1810, где используются девять буферов по одному килобайту, каждый из которых может хранить только по одному сектору. Для инициализации буферов служит вызов LBRES.

В системе может быть одновременно определено несколько логических устройств. ПО MDISK и BIOS CM1810 поддерживают до десяти таких устройств :FX:, X=0...9. Процесс определения логического устройства отождествляется с фиксацией связанного с ним DUIB (прикрепление физического устройства). Для этого состав вызовов драйвера логического диска расширяется операцией LSET. Вызов LCHK возвращает DUIB логического устройства программам более высокого уровня, позволяя учитывать конкретные особенности устройств внешней памяти. Эта информация необходима для ряда таких команд DOS1810, как FORMAT или PATCH.

Усовершенствование драйвера логического диска может быть связано с вводом в действие механизмов отслеживания плохих дорожек, которые обнаруживаются во время форматирования диска. Информация о них хранится на самом диске и считывается драйвером при первом к нему обращении. Она используется при пересчете логического адреса в физический. Пакет MDISK эту функцию не поддерживает.

Над системой логических устройств может быть построен еще один уровень ПО, ответственный за связь с внешними устройствами ВВ, который называют файловой системой. Рассмотренные структуры совместимы с файловой системой DOS1810 и, следовательно, могут быть полезны в МС, ориентированных на ее использование.

Телефон, 408-62-22, Москва
ЛИТЕРАТУРА

- Щелкунов Н. Н., Дианов А. П., Организация контроллеров ГМД // Микропроцессорные средства и системы. — 1990. — № 4.
- Дианов А. П., Щелкунов Н. Н. Одноплатные микроЭВМ семейства мМС // Микропроцессорные средства и системы. — 1990. — № 1.
- Щелкунов Н. Н., Дианов А. П. Техника программирования 8-разрядных микроконтроллеров // Микропроцессорные средства и системы. — 1986. — № 6. с. 23—28.
- Дисковая операционная система для подготовки программ реального времени на CM1810 (DOS1810). Описание применения 2/Б/00132-01 31 01. — 1986.

Статья поступила 16.02.88

КОМПАКТНАЯ СТАТИСТИЧЕСКАЯ СИСТЕМА ДЛЯ IBM-ПОДОБНЫХ ПК

Обработка данных с использованием статистических методов — неотъемлемый компонент любой исследовательской деятельности, и доступность этого средства возрастает с распространением ПК. Однако среди известных зарубежных статистических пакетов для ПК (табл. 1, 2) мы не найдем хорошей системы для среднего пользователя. Программные «монстры» типа BMDP и SAS и их меньшие братья — гиганты SYSTAT и SPSS доступны только специалистам высокой квалификации. Они содержат широкий спектр статистических методов и их модификаций, большинство из которых предназначено для специального применения. Средства для оперирования данными, напоминающие ЯП, достаточно мощны, но требуют специального образования и навыка работы. Псевдографика для представления диаграмм данных и результатов неестественна, и получаемые рисунки не подходят для прямого использования в статьях и отчетах.

Есть компактные и простые в обращении статистические программы, подобные MICROSTAT и STATPAC, но они вовсе не содержат средств графического представления данных и результатов, а возможности диалога в них далеки от современных требований. Даже эргономически проработанные системы типа STATGRAPHICS с прекрасными графическими и диалоговыми возможностями в некоторых своих аспектах излишне сложны и избыточны для среднего пользователя. Кроме того, большинство зарубежных статистических систем трудно адаптировать для отечественных IBM-подобных ПК типа ЕС 1840, «Искра 1030», «Нейрон И9» из-за технических ограничений и различий.

Общие характеристики популярных статистических программ

Программа	BMDP	SYSTAT	STATGRAPHICS	SPSS	СТАТИС	MICROSTAT	STATPAC
Тип	Пакет	Пакет	Система	Пакет	Система	Пакет	Пакет
Язык общения	Директивный	ЯП	Меню	ЯП	Меню	Меню	Директивный
Максимальный размер выборки	>15000	>10000	4000	>10000	2000	1200	400
Редактор данных	Внешний строковый	Внешний экранный	Встроенный экранный	Внешний экранный	Встроенный строковый	Внешний строковый	Нет
Графика	Псевдо	Псевдо	Полн. Ч/Б	Псевдо	Полн. ЦВ	Нет	Нет
Число ГМД	80	5	6...9	14	1	1	1
Требует памяти.	640	>64	520	>64	128	64	64
Кбайт							
Жесткий диск, Мбайт	25	Желательно	10	Желательно	Нет	Нет	Нет
Средства программирования	Есть	Есть	Есть	Нет	Нет	Нет	Нет
Цена, \$	2500	600	690	1500	—	150	100

Примечание. ЯП — язык программирования.

Статистические методы и оценки

Программа	BMDP	SYSTAT	STATGRAPHICS	SPSS	СТАТИС	MICROSTAT	STATPAC
Преобразование	22	22	4	23	14	14	
Описательная статистика	26	10	15	14	14	14	8
Распределение	2		18		5	9	5
Параметрические тесты							
корреляция							
Пирсона	1		1		1		1
Хи-квадрат	3		1	1	3	1	1
Стьюдента	4		2	3	3	4	2
Фишера	1		1		1		
другие	2						
Непараметрические тесты							
знаков	1	1		1			
Вилкоксона	1	1	2	2	2	2	1
Мани-Уитни	1	1	1	1	1		1
Вальд-Вольфовича			1	1	1	1	
Колмогоров-Смирнова	1	2	2	2	2	2	
корреляция							
Спирмана	2	1			1		
конкордация							
Кендела		1	1	1		1	
медиана			1	2			
биномиальный				1			
Мак Немана	1			1			
Мозеса				1			
Кохрана				1			
Категориальный анализ							
кросс-табуляция							
сопряженность признаков	1	1	1	1	1		
признаков	25	13	1	9	1	2	1
многомерные таблицы	1						
лог-линейный	2	1	1				

Таблица 1

Продолжение табл. 2

Программа							
1	2	3	4	5	6	7	8
Дисперсионный анализ							
однофакторный							
двухфакторный	5		1	1	1	1	1
трехфакторный	2		2	3	3	2	1
многофакторный							
многомерный		9	2/12	2			
Крускал-Уоллиса	4	1					
Фридмана	1	1	1	1	1	1	
Регрессионный анализ							
однопараметрический	3		5	3	5	1	1
множественный линейный	2	1	2/17	1	1/30	1	1/20
+ селекция переменных	10		2	3			
полиномиальный	1	1	1		1/30		1/11
нелинейный	7		1				
многомерный	1						
Анализ временных серий							
автокорреляция	2		2	1	1		1
кросскорреляция	1		1	1	1		1
матричная корреляция	4	7/200	2/22	2	1/41	1	1
спектральный	2	1	2	1	1		
сглаживание		8	6		4		3
предсказание	1		5				1
Бокс-Дженкинса	4	4	4	4			
Многомерные методы							
кластерный	23	8	8	8	6		
факторный	5	2	2	1	2		
+ вращение факторов	4	3	3	3			
дискриминантный	2	1	1	1			
каноническая корреляция	1	1	1				
шкалирование	4						
Всего	183	106	106	101	79	59	32

Примечание. Пакет — набор отдельных подпрограмм с индивидуальным входом; система — связанная замкнутая программная система; п — максимальное число переменных.

Разработанная статистическая система СТАТИС, работающая на IBM-подобных ПК, включает следующие компоненты:

наиболее широко используемые статистические методы и оценки;
удобные и простые операциональные средства

в форме меню, подсказок и команд, выполняемых по одному нажатию ключа;

развитые и стандартизированные средства графического представления данных и результатов с возможностью оперативного вывода на монитор в цветном или черно-белом виде и получения качественной копии на принтере; способность к быстрому переходу от манипулирования данными к статистической обработке и графическому выводу.

Система СТАТИС компактна, проста в работе, оперирует с несколькими выборками и матрицами размерностью до 100 элементов, включает встроенный редактор данных и файловую систему, средства преобразования данных и цветной графики, адаптируемой к наиболее распространенным типам видеоконтроллеров. Ввод данных возможен с клавиатуры терминала, из числового или текстового файла. Результаты вводятся на экран и принтер в числовом или графическом виде, в дисковый файл — в текстовом виде, что позволяет использовать любой текстовый редактор.

Диалоговая инфраструктура системы удовлетворяет самым современным международным стандартам в этой области. Так, в нижней части экрана постоянно присутствует информационная панель, содержащая список допустимых команд и ключей (команды исполняются при нажатии на соответствующую клавишу). Нижняя строка экрана постоянно отведена под вопросы, которые система может задавать пользователю при исполнении команд. Остальная часть экрана предназначена для выдачи информации. На экране можно получить краткое описание каждой команды, что делает возможной работу в системе «с листа», т. е. без какого-либо предварительного изучения инструкции.

Средства графического представления данных позволяют получать график функциональных зависимостей Y от X , схемы распределения точек в двумерном пространстве, графики одной выборки или совместные графики двух выборок, матричные графики, в которых элементы каждой строки экспонируются в отдельной колонке с указанием положения среднего значения и разброса значений. При выполнении методов регрессионного и многомерного анализов, анализа временных рядов можно вывести графики результирующих функций. Каждый график, полученный в результате работы СТАТИС или подготовленный средствами внешнего текстового редактора, выводится на экран или принтер. Пользователю предоставлена возможность задавать требуемые размеры рисунка, перемещать графики в плоскости рисунка, добиваясь наилучшего расположения, строить совместные графики, выполнять подрисовочные надписи в любом доступном шрифте. Все это позволяет использовать полученные рисунки без каких-ли-

бо доработок при подготовке научных статей и отчетов. Сравнительные характеристики системы СТАТИС даны в табл. 1.2.

Телефон 939-50-05.

ЛИТЕРАТУРА

1. Dixon W. J., Endelman L., Frans J. W., Hill M. A., Jenerich R. I., Toporec J. D. BMDP Statistical Software // BMDP Inc., 1985.
2. Wilkinson L., SYSTAT: The System for Statistics. Evanston // IL: SYSTAT Inc., 1987.
3. STATGRAPHICS: Statistical Graphics System. User's Guide. Plus*Ware Product // STSC Inc., 1986.
4. SPSS/PC: SPSS for IBM PC/XT/AT // SPSS Inc., 1986.
5. MICROSTAT. Release 4.0. Ecosoft. Indianapolis, 1985.
6. STATPAC. Version 1.0. MPI, Barcing, 1985.

Статья поступила 2.12.88

УДК 681.3.06

А. Ф. Дедков, Е. В. Наумов, А. Л. Щер

СРЕДСТВА РАБОТЫ С БАЗАМИ ДАННЫХ В ЯЗЫКЕ ПРОЛОГ

С середины 70-х годов в тесной связи с работами в области представления знаний стали развиваться исследования по созданию так называемых дедуктивных баз данных [1], в которых новые факты (данные) могут быть выведены из уже известных и хранимых в явном виде. Эти базы данных (БД) опираются на методы математической логики и рассматриваются с позиции теории доказательств. Было показано, что такой подход в полной мере применим и к традиционным БД реляционного типа, которые обычно рассматриваются с позиции теории моделей [2].

В настоящее время одно из перспективных направлений в создании дедуктивных БД связано с языком логического программирования Пролог [3]. Полномасштабная реализация дедуктивных БД включает не только средства формулировки запросов, но и поддержку ограничений целостности выводимых фактов, а также эффективные методы работы с большими массивами данных на внешних носителях. Все это требует значительной модификации и расширения интерпретатора языка Пролог и соответствующих вычислительных ресурсов.

На практике целесообразно иметь интерфейс языка Пролог и одну из традиционных БД реляционного типа, сочетающие преимущества, получаемые пользователями каждой из систем в отдельности, с возможностью поэтапного наращивания дедуктивной составляющей БД и приобретения опыта работы с ней. Даже при непосредственной реализации очевидно большое ме-

тодологическое преимущество этого подхода, который и был принят за основу разработки системы Пролог-БД (средства работы с БД на языке Пролог) для ПЭВМ.

Основные положения

Пролог-БД представляет собой систему, объединяющую в единое целое язык логического программирования и систему управления БД. Интеграция выполнена на основе расширения интерпретатора языка Пролог-МР в направлении непосредственной работы с файлами БД dBASE II [4]. При этом пользователи языка и БД получают ряд дополнительных преимуществ.

Для пользователей Пролога они заключаются в контролируемом вводе и работе с большими массивами данных, хранимых на внешних запоминающих устройствах. Известно, что для задач практической значимости факты составляют обычно большую часть Пролог-программ.

Для пользователей dBASE II, в свою очередь, открывается возможность использования Пролога как языка запросов реляционных БД, эквивалентного известному языку QBE [5]. Обеспечивается обработка нескольких файлов БД (отношений) одновременно в рамках одного запроса, построение рекурсивных запросов, использование выведенных фактов непосредственно в среде dBASE II.

Согласно принципу, положенному в основу создания Пролог-БД, от пользователей Пролога и dBASE II не требуется никаких дополнительных усилий по реорганизации Пролог-программ или файлов БД для работы в интегрированной среде. Такая совместимость достигается благодаря введению уровня автоматического поиска фактов на внешнем запоминающем устройстве и сохранению механизма логического вывода языка Пролог на основе унификации структур.

С точки зрения языка Пролог файлы БД рассматриваются как набор фактов, имена структур которых определяются именем соответствующего файла, а сами структуры — структурой этого файла. Значения полей символьного и числового типа файлов БД соответствуют атомам и числам, которые унифицируются только с равными им значениями в запросе. Поля логического типа обрабатываются точно так же, как и поля символьного типа, за исключением интерпретации пустых значений БД. Унификация заканчивается успешно, если значение истинности логического поля соответствует символьному значению Т (истина) или F (ложь), заданному в запросе. В противном случае считается, что унификация закончилась неудачей.

При вводе данных в файл БД dBASE II значения некоторых полей могут быть не заданы. Система Пролог-БД поддерживает работу с та-

кими полями (поиск и выдача значений) в соответствии с принципами работы dBASE II. Унификация с пустыми значениями полей файла БД заканчивается успешно, если в запросе пользователя для символьных, числовых и логических полей заданы соответственно: " (один пробел), 0 (нуль), F (ложь).

Запросы к БД формулируются в полном соответствии с синтаксисом языка Пролог, а поиск осуществляется на основе унификации структур. При вычислении запросов интерпретатор организует поиск фактов сначала в БД Пролог, размещенной в оперативной памяти, и, если он заканчивается неудачей, переключается на поиск соответствующего файла БД на внешнем запоминающем устройстве. При этом используются только файлы с расширением DBF, в том числе и сортированные, для обработки которых предусмотрена эффективная процедура. При вычислении запросов с ключевыми параметрами используется нестандартная семантика унификации структур типа отношения (равно, больше и т. п.). Это вызвано желанием обеспечить возможность работы с отдельными полями файла, а область действия такой унификации строго ограничена запросами к БД.

При формулировке запросов могут использоваться факты и правила. Таким образом, можно порождать новые отношения, которые обобщают отношения, известные как «представления» (VIEWS) в традиционных БД, что непосредственно ведет к понятию дедуктивных БД. При этом могут использоваться все имеющиеся встроенные предикаты и стандартные процедуры языка Пролог-МР, а также процедуры, написанные пользователями. Все это значительно расширяет возможности работы с дедуктивными БД и характерно для принятого подхода.

Дополнительные инструментальные средства предоставляют пользователям ряд сервисных возможностей, в том числе встроенный редактор запросов, а также поддерживают определенную технологию работы с БД. Они позволяют представлять порождаемые с помощью правил вывода Пролог-программы отношения в виде стандартных файлов БД и обрабатывать их обычными средствами в среде dBASE II. Таким образом, технологическая цепочка БД — Пролог — БД полностью замыкается.

Заметим, что в части работы с БД система Пролог-БД предназначена главным образом для формулировки запросов, поэтому ввод, изменение, добавление, удаление данных, вывод на печать отчетов должны выполняться отдельными стандартными средствами dBASE II.

Запросы к базам данных

Элементарные запросы представляют собой структуру Пролога, имя функтора которой совпадает с именем некоторого файла БД. Эlemen-

ты структуры называются параметрами запроса и могут быть либо позиционными, либо ключевыми. Одновременное их использование в одном элементарном запросе не допускается.

Позиционные параметры представляют собой элементы структуры Пролога типа переменная, число, атом. Их число и порядок должны в точности совпадать с числом и порядком полей файла БД. Числа в запросе соответствуют значениями числовых полей файла БД, атом — значениями символьных или логических полей (их совокупность определяет условие выборки). Оно считается выполненным, если значения параметров запроса унифицируются с соответствующими значениями полей некоторой записи файла БД. При этом пользователю выдаются значения всех остальных полей данной записи, которым в запросе должны соответствовать переменные. Несовпадение типов параметров запроса и полей БД фиксируется как ошибка, о чем пользователь тут же уведомляется.

Возможности языка запросов будут иллюстрироваться на примерах БД «Поставщики — потребители» и отдельного файла (отношения) «Структура подчинения», приведенных ниже.

Структура БД «Поставщики — потребители»

ПОСТАВЩИК (НОМ ПОСТ, НАЗВ
ПРЕДПР, СТАТУС, МЕСТО_РАСП).

ИЗДЕЛИЙ (НОМ ИЗД, НАИМ_ИЗД,
ЦВЕТ, ВЕС, МЕСТО_ХРАН).

ПОСТАВКА (НОМ_ПОСТ, НОМ_ИЗД, КО-
ЛИЧЕСТВО).

Отношение «Структура подчинения»

СТРУКТ_ПОДЧ

НАЧ С10 С10 С10 С60 С80 С150 С150
С240

ПОДЧ С60 С70 С80 С180 С150 С160 С200
С240 С320

Приведенные БД позволяют сравнить запросы в системе Пролог-БД с аналогичными запросами на языке QBE. При этом предполагается использовать такой режим работы интерпретатора, который позволяет получить все ответы на запрос.

Пример. Выборка по условию. Получить полные данные о поставщиках, находящихся в Москве и имеющих статус 30.

?— поставщик (X1, X3, 30, Москва).

В результате выполнения запроса на дисплей выводятся полные данные об интересующих поставщиках (последовательные значения переменных X1 и X2). Позиционные параметры

удобно использовать, когда требуется простая или условная выборка всех полей БД.

Ключевые параметры представляют собой элементы структуры Пролога типа бинарное отношение (=, / =, >, <, > =, = <), левая часть которого является атомом, интерпретируемым как имя поля, а правая часть — переменной, числом или атомом. Для символьных и логических полей файлов БД допускается использовать только отношения «равно» и «не равно». Ключевые параметры можно задавать в произвольном порядке и только для тех полей, которые непосредственно интересуют пользователя. Несовпадение типа правой части ключевого параметра и соответствующего поля файла БД, определяемого левой частью, фиксируется как ошибка.

Пример. Выборка по условию. Получить номера поставщиков, находящихся в Москве и имеющих статус больше 20.

? — поставщик (ном_пост=X1, статус>20, место_расп=Москва).

Результатом выполнения этого запроса будет последовательная выдача на дисплей значений переменной X1, представляющей поле ном_пост (номер поставщика).

Запросы с использованием связей. Из элементарных запросов с помощью связи по переменным можно построить более сложные запросы к нескольким файлам БД (отношениям). Такие связи используются в Пролог-БД, как и в языке QBE. Там, где в языке SQL требуется вложенное отображение, в подязыке данных ALPHA — квантор существования, а в реляционной алгебре — операция JOIN.

Смысл связи заключается в том, что переменные, по которым она установлена и которые принадлежат разным элементарным запросам (подцелям языка Пролог), должны иметь одно и то же значение, чтобы этот запрос был удовлетворен.

Пример. Выборка с использованием связи внутри одного отношения. Получить номера поставщиков, которые поставляют изделия «и10» и «и20».

? — поставка (ном_пост=X1, ном_изд=i10),
поставка (ном_пост=X1, ном_изд=i20).

Этот запрос состоит из двух подцелей языка Пролог, которые решаются последовательно. Для каждой записи файла БД «Поставка» сначала проверяется условие ном_изд=i10 и в случае успеха делается попытка выполнить вторую подцель, т. е. проверяется условие ном_изд=i20 для текущего значения переменной X1. В случае неудачи осуществляется возврат назад и из файла БД «Поставка» выбирается следующая запись.

Пример. Выборка с использованием связи. Получить названия предприятий-поставщиков, изделия «и20».

? — поставщик (ном_пост=X1, назв_предпр=X2),
поставка (ном_пост=X1, ном_изд=i20).

В языке Пролог обычно выводятся значения всех переменных (в нашем примере X1 и X2). Если необходимо ограничить вывод некоторых переменных, то это можно сделать с помощью правил.

Запросы с использованием правил. Возможность использования правил языка Пролог при работе с БД является наиболее важной особенностью системы Пролог-БД. Во-первых, с их помощью решаются многие технические вопросы формулировки запросов, отмеченные выше. Во-вторых, они дают возможность при работе использовать все средства традиционного Пролога как языка программирования. И, наконец, служат основой построения дедуктивной составляющей БД, а также представления и работы со знаниями.

Пример. Выборка из нескольких отношений. Для каждого поставляемого изделия получить номер изделия и название города, из которого оно поставляется.

география_изд (X1, X2): —
поставщик (ном_пост=X3, место_расп=X2),

поставка (ном_пост=X3, ном_изд=X1).
? — география_изд (X1, X2).

Заметим, что сам запрос при этом ставится проще, а главное, что при его вычислении порождается новое отношение с именем «география_изд», которое, в свою очередь, можно использовать в других правилах для порождения необходимых пользователю отношений и т. д.

Запросы с использованием встроенных предикатов и стандартных процедур. Рассмотрим более сложные виды запросов, формулировка которых требует определенных навыков в программировании на языке Пролог. Встроенные предикаты реализуются непосредственно интерпретатором языка Пролог. Процедуры перед началом работы должны быть загружены в оперативную память пользователем. Набор стандартных процедур включает pot, max, min, count и др. Его легко расширить, а процедуры при необходимости модифицировать.

Пример. Выборка с использованием предиката cut. Получить названия предприятий-поставщиков, которые поставляют по крайней мере одно изделие красного цвета. Предикат управления логическим выводом cut позволяет отсечь ненужные ветви поиска, как только для очередного поставщика будет найдено первое

поставляемое им изделие красного цвета.

отсеч (X1): —

поставка (ном_пост=X1, ном_изд=
=X2),

изделие (ном_изд=X2, цвет=красный),

? — поставщик (ном_пост=X1, назв_предпр=X2),
отсеч (X1).

Пример. Выборка с использованием процедуры `not`. Получить номера поставщиков с наивысшим статусом.

? — поставщик (ном_пост=X1), статус=
=X2),
`not` (поставщик (статус>X2)).

Пример. Выборка, включающая ВСЕ, с использованием процедур `setof` и `subset`. Получить номера поставщиков всех тех изделий, которые поставляют «п20».

Процедура `setof` используется для построения списков (множеств) с неповторяющимися элементами, а процедура `subset` — для проверки того, что одно множество является подмножеством другого.

сравн_пост (X1, X2): —

`setof` (X3, поставка (X2, X3, —), X4),

поставщик (ном_пост=X1),

`setof` (X5, поставка (X1, X5, —), X6),

`subset` (X4, X6).

? — сравн_пост (X1, п20).

Запросы с использованием рекурсии. Рассмотрим операцию выборки в отношениях, имеющих древовидную структуру. Смысл любого кортежа этого отношения состоит в том, что указанный служащий с номером «подч» работает под руководством указанного начальника с номером «нач». Характерная особенность формулировки запросов в древовидных структурах на языке Пролог состоит в использовании рекурсивных процедур. Они позволяют строго в рамках логики предикатов формулировать запросы, которые нельзя выразить в реляционной алгебре или языках, эквивалентных ей по мощности. В языке QBE такие запросы строятся с помощью дополнительных языковых конструкций для работы с уровнями.

Пример. Выборка со всех уровней ниже заданного. Получить номера служащих, подчиненных служащему «с80» на любом уровне.

Вспомогательная рекурсивная процедура для реализации этого запроса и сам запрос формулируются следующим образом:

подчинение (X1, X2): — `структ_подч`
(X1, X2).

подчинение (X1, X3): — `структ_подч`
(X1, X2), подчинение (X2, X3).

? — подчинение (с80, X1).

Для решения этого и аналогичных ему примеров можно написать рекурсивную процедуру поиска и выбора данных со всех уровней древовидной структуры, удовлетворяющих не-

которому отношению (равно, меньше и т. п.) относительно заданного уровня.

Реализация и ограничения

Экспериментальная версия системы Пролог-БД разработана на базе интерпретатора языка Пролог ПЭВМ Роботрон 1715, имеющей восьмибитовый процессор, аналогичный процессору Z80A, оперативную память объемом 64 Кбайт и внешнюю память на двух дисководах объемом 780 Кбайт. Операционная система аналогична CP/M.

Интерпретатор Пролог-MP построен по классической схеме и реализует практически полный стандартный вариант языка Пролог [3]. Основные ограничения языка следующие. Операции динамической модификации программы разрешены только для фактов, но не для правил. Встроенный предикат ввода данных вводит только числа и атомы, но не произвольные структуры. Введен также и ряд значительных расширений набора встроенных предикатов.

Интерпретатор написан на языке Турбо-Паскаль и имеет объем менее 3000 строк исходного текста. При создании этого интерпретатора ставилась основная задача разработать максимально компактный переносимый интерпретатор, который мог бы служить «стендом» для различных экспериментов в области расширений на языке Пролог. Во время работы интерпретатор, имеющий оверлейную структуру, занимает всего 25 Кбайт и для размещения интерпретируемого кода Пролог-программы и стеков периода выполнения остается около 23 Кбайт памяти, что позволяет выполнять программы объемом в 100...150 правил. Число обрабатываемых фактов ограничено только объемом внешней памяти благодаря возможности использования для их хранения дисковых файлов. Скорость работы интерпретатора составляет 50...70 логических выводов в секунду. Разумеется, небольшой объем оперативной памяти машины не позволяет выполнять на ней сложные программы. Практическое применение интерпретатора представляет возможным для решения двух основных задач: обучения программистов языку Пролог и применения Пролога в качестве языка запросов к БД.

Основное ограничение системы Пролог-БД в работе с БД — размерность используемых в запросах и обрабатываемых чисел (диапазон — 32768...32767 определяется системой программирования Турбо-Паскаль). Иногда это ограничение можно обойти, выбирая подходящую размерность представления больших чисел в БД, т. е. за единицу измерения брать тысячи, десятки тысяч и т. д.

Экспериментальные исследования показыва-

ют, что система Пролог-БД обеспечивает эффективную работу пользователей в среднем с 7,8 файлами БД (отношениями) dBASE II. Время поиска по одному файлу БД в системе Пролог-БД примерно на 20 % больше, чем в системе dBASE II, однако с возрастанием сложности запросов, требующих просмотра нескольких файлов БД или двукратного просмотра даже одного файла, времена поиска полностью уравниваются. При этом построение соответствующего запроса в системе Пролог-БД оказывается несоизмеримо проще, чем написание и отладка программы для реализации того же самого запроса в системе dBASE II. Так, например, для реализации запроса «сравн_пост (X1, X2)» требуются программы с достаточно сложной логической структурой объемом порядка 50 операторов языка dBASE II.

Развитие системы Пролог-БД предусматривает снятие наиболее существенных ограничений и разработку дружественного интерфейса конечного пользователя. В настоящее время достигнута интеграция системы с БД dBASE III на ПЭВМ ЕС1840.

Телефон 235-81-23, Москва

ЛИТЕРАТУРА

1. Gallaire H., Minker J., Nicolas J.-M. Logic and databases: a deductive approach // Computing surveys.— 1984.— V. 16, № 2.— P. 153—185.
2. Reiter R. Towards a logical reconstruction of relational databases theory.— In: M. Brodie, J. Mylopoulos, J. W. Schmidt (Eds.) // On conceptual modelling. Berlin and New York: Springer — Verlag.— 1984.— P. 191—233.
3. Клоксин У., Меллиш К. Программирование на языке ПРОЛОГ.— М.: Мир, 1987.— 336 с.
4. Krumm R. Understanding and using dBASE — II * dBASE III.— Browie, Maryland: Prentice — Hall Publ. Comp., 1985.— 437 p.
5. Дейт К. Введение в системы без данных.— М.: Наука, 1980.— 464 с.

Статья поступила 1.07.88

УДК 681.3.06

А. С. Гаганов, В. П. Анищенко, С. А. Морозов, А. Д. Черкай

ИНСТРУМЕНТАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ОЭВМ КБ1013

Широкое внедрение 4-разрядных ОЭВМ в народное хозяйство позволяет разрабатывать новую бытовую технику, электронные игры для обучения детей, создавать комфортные условия при экономии электроэнергии, увеличивать ассортимент товаров повышенного спроса. Для быстрой, эффективной и качественной разработки прикладных программ необходимы максимально удобные пользователю инструментальные программные средства для отладки и проверки работоспособности на функционально-логической модели ОЭВМ. Инструментальное программное обеспечение (ИНПО) разработано для серии ОЭВМ КБ1013.

Технические характеристики ОЭВМ серии КБ1013

	БК 1-2	БК 4-2	БК 7-2
Объем памяти:			
программ (ПЗУ), байт	1827	2772	4032
данных (ОЗУ), полубайт	65	128	128
изображений, полубайт	18	32	32
музыкального ПЗУ, бит	—	—	256×6
Число выводов ЖКД	72	128	136
Число сегментов	36	32	34
Разрядность регистра вывода, бит	4	8	8
Число команд	58	53	59
Длительность машинного цикла, мкс	61		61/122
Тактовая частота, кГц	32,8		
Максимальный потребляемый ток, мкА	60		
Напряжение питания, В	3,0±10 %		

Серия ОЭВМ КБ1013 выгодно отличается от существующих устройств такого класса малым потреблением мощности, наличием встроенного контроллера жидкокристаллических дисплеев (ЖКД), музыкального автомата и встроенного таймера, а также низкой стоимостью [1, 2].

Комплекс ИНПО-1013 работает под управлением систем RT-11, ОС ДВК на микроЭВМ типа «Электроника 79», «Электроника 100/25», ДВК и совместимых с ними (возможен перенос на микроЭВМ других типов).

Его состав: транслятор с ассемблера, реасемблер машинных кодов, программы формирования управляющей магнитной ленты (МЛ) для изготовления фотошаблонов ПЗУ с разработанной программой пользователя, музыкального ПЗУ, программно-логическая модель, программа моделирования ЖКД, используемого в разрабатываемой прикладной программе, и комплект эксплуатационной документации на компоненты, подготовленные в соответствии с требованиями ЕСПД.

Процесс разработки прикладных программ состоит из следующих этапов:

- создание модели ЖКД, если вывод на ЖКД предусматривается алгоритмом решения задач;
- подготовки и трансляции исходного текста программы;
- отладки программы на программно-логической модели;
- подготовки исходного текста программы для музыкального ПЗУ.

ЖКД моделируется в интерактивном режиме по специальной программе, создающей образы сегментов ЖКД, запоминаемые в файле. Модель разрабатываемого ЖКД может создаваться как заново, так и на основе ранее разработанной модели, хранящейся в файле. Созданный файл модели используется на этапе отладки прикладной программы.

Исходный текст программы создается любым текстовым редактором операционной системы и обрабатывается транслятором. Он состоит из

директив ассемблера, машинных команд и комментариев. В результате работы транслятора создаются следующие файлы:

листинга с адресами ПЗУ и сообщениями об ошибках;

объектных кодов программы пользователя для формирования управляющей МЛ;

образа ПЗУ для программно-логической модели.

Программы отлаживаются в интерактивном режиме на программно-логической модели, отражающей архитектуру и систему команд ОЭВМ серии КБ1013. Входные данные для моделирования выполнения программы — файлы с моделью ЖКД, образом ПЗУ и соответствия клавиатуры входным сигналам ОЭВМ (создается любым текстовым редактором). Программно-логическая модель обеспечивает просмотр любой ячейки памяти и регистров моделируемой ОЭВМ, выполнение всей отлаживаемой программы или любой ее части, останов отлаживаемой программы в любой заданной точке и имитацию ввода-вывода.

Специальные методы проектирования позволяют имитировать работу отлаживаемой программы в масштабе времени, близком к реальному.

Для занесения отлаживаемой программы пользователя в ПЗУ конкретной ОЭВМ предназначена программа формирования управляющей МЛ, создающая файл образа ПЗУ в формате фотонаборной установки. Правильность занесения контролируется программой реассемблера и восстановленным исходным текстом из кодов ПЗУ.

Музыкальный автомат (ОЭВМ КБ1013ВК7-2) может программировать мелодии в диапазоне двух октав (от «до-диз» второй октавы до «до» четвертой октавы). Файл мелодий подготавливается любым редактором текстов, затем на его основе программа формирования управляющей магнитной ленты создает МЛ для изготовления фотошаблона музыкального ПЗУ.

Разработан пакет арифметических программ,

включающих следующие модули: сложение, вычитание, умножение и деление с фиксированной точкой; нормализация результата; сравнение и деление чисел с плавающей точкой; умножение чисел с плавающей точкой, эффективное с точки зрения объема памяти и времени выполнения.

Все операции производятся с числами, представленными в двоичной системе счисления (положительные числа в прямом, а отрицательные в дополнительном кодах). При умножении и делении чисел с плавающей точкой в специально выделенной ячейке формируется флаг знака результата. Результат получается в нормализованном виде (мантисса и порядок), заполняется полностью вся разрядная сетка мантиссы. Арифметика занимает в ОЗУ не более девяти страниц по 63 слова. Разработаны программные блоки вычисления специальных функций (\sqrt{x} , e^x , $\ln x$).

Инструментальные средства, состоящие из ассемблера, реассемблера, программно-логической модели и программы подготовки управляющей МЛ, хорошо зарекомендовали себя при создании электронных игр «Ну, погоди!», «Футбол», «Космический мост», «Биллиард», «Учитель арифметики», «Учитель английского языка», а в улучшенном варианте электронной игры «Ну, погоди!» позволили разместить в ПЗУ три игры вместо двух.

107207, Москва, Щелковское ш., 77. ЦНИИ «Циклон», Консультационно-технический центр; тел. 460-41-44.

ЛИТЕРАТУРА

1. Морозов С. А., Барановский Д. М., Минкин Л. К., Семикин А. П., Черкай А. Д. Однокристалльные ЭВМ серии КБ1013 // Микропроцессорные средства и системы. — 1987. — № 5. — С. 5—18.
2. Морозов С. А., Черкай А. Д., Минкин Л. К., Семичастнов О. Л., Кротков Б. В. Однокристалльная 4-разрядная микроЭВМ КБ1013ВК7-2 // Микропроцессорные средства и системы. — 1988. — № 3. — С. 20—28.

Статья поступила 22.03.89

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Анализ ядра МОС РВ СМ1800 показал, что из полтора десятков модулей, составляющих библиотечный файл ядра, большинство аппаратно-независимые т. е. могут работать в любой системе на основе МП серии КР 580, и лишь два модуля (IN8020, ТВ8020) аппаратно-зависимые. Они связаны с системой прерываний и системным таймером. Замена этих двух модулей ядра на новые обеспечивает его адаптацию к конкретной архитектуре микросистемы.

В СМ1800 прерывания выполняются с помощью микросхемы К589ИК14, а модуль тайме-

УДК 681.325.5—181.4

В. Д. Глизбург

ЯДРО СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ МИКРОСРЕДСТВ НА БАЗЕ МИКРОПРОЦЕССОРА КР580ВМ80

В состав программного обеспечения (ПО) комплексов СМ1800, основу которых составляет МП КР580ВМ80, входит система реального времени МОС РВ (аналогичная ей ОС РВ входит в МИКРО-ДАТ для КТС ЛИУС-2). Немало программистов имеют опыт работы с этими программными системами.

ра собран на микросхемах серий К155, К589, К555. В большинстве же микросистем на базе МП серии КР580 используются программируемый контроллер прерываний (ПКП) КР580ВН59 и таймер КР580ВИ53.

Именно для такого случая разработаны программы NEWIN и NEWTV аппаратно-зависимых модулей. Программы соответствуют наиболее простому варианту: в системе один ПКП, работающий в простом приоритетном режиме без регистра маски прерываний; выход системного таймера счетчика 0 микросхемы КР580ВИ53, работающей в режиме 0, связан с уровнем 1 запроса прерываний ПКП. Единица системного времени равна 5 мс.

Свойства полученного таким образом ядра и прикладных систем, построенных на его основе, почти не отличается от аналогичных свойств в рамках МОС РВ СМ1800.

Наиболее важное отличие адаптированного ядра от исходного связано с тем, что для МОС РВ СМ1800 значение приоритета задачи (интервал 0...127) в системе преобразуется в код «штурм», определяющий границу между маскируемыми и немаскируемыми прерываниями в момент, когда данная задача является текущей. В новом ядре подобное маскирование прерываний действует только до операции «неспецифический сброс» (EOI) регистра состояния ПКП КР580ВН59, выполняемой в конце системной подпрограммы обработки прерываний INTPUB. Таким образом, текущая задача с приоритетом в интервале 0...127 может прерываться возникающими низкоприоритетными по отношению к ней прерываниями.

Практическая реализация изложенного метода несколько затруднена из-за необходимости связывания новых модулей с помощью глобальных имен с остальными, аппаратно-независимыми модулями ядра, в которых эти имена (видимо с целью предотвращения постороннего доступа) недопустимы в рамках Макроассемблера ASM80, так как содержат больше шести символов (Макроассемблер оставляет в файле объектного кода в именах шесть символов, а «хвост» обрезает).

Изложенный подход использован для адаптации ядра реального времени на две различные микросистемы на основе МП серии КР580, включая серийно выпускаемую УЦМ-100. Помимо ядра разработаны системные программы, имеющие статус задач МОС РВ, например драйвер ИРПС канала. Прикладные системы реального времени на основе адаптированных ядер подготавливаются на СМ1800, загружаются в ОЗУ микросистемы по каналу связи (ИРПС) с помощью начального загрузчика и отлаживаются. Отлаженную прикладную систему можно загрузить в ПЗУ микросистемы.

Телефон 657-53-06, Ленинград

УДК 681.3.06

В. Н. Бальмич, Г. В. Кройчик, С. М. Крол,
О. Г. Палий, Р. В. Фелезюк

СЕМЕЙСТВО КРОСС-СИСТЕМ ПОМПА

Семейство диалоговых адаптивных кросс-систем автоматизированного проектирования ПОМПА составляет часть Прометей-технологии [1], обеспечивающей проектирование программных изделий (ПИ) на всех этапах жизненного цикла.

Кросс-системы семейства представляют собой функциональный ряд и могут настраиваться на конкретные типы целевых ЭВМ. Разработаны варианты систем для различных инструментальных ЭВМ (СМ4, СМ1420, ПЭВМ «Электроника МСО585»).

Кросс-системы ПОМПА наряду с традиционными средствами трансляции, компоновки и отладки программ включают средства ведения базы данных проектирования, контроля за ходом разработки, широкий набор сервисных программ-утилит, подсистему выпуска текстовой документации в стандартах ЕСКД и ЕСПД.

Общая структура систем ПОМПА для СМ ЭВМ представлена диаграммой на рисунке. Весь перечисленный набор средств включая компилятор с языка Ада должен составить наиболее полный вариант системы — ПОМаА-П. Идеология построения системы ПОМаА основана на понимании единого, ориентированного на окружение компилятора с языка Ада [2], набора технологических средств. К подмножеству таких средств можно отнести инструментальные средства, предназначенные для программирования на ассемблере. Так, вариант ПОМПА-С предлагает разработку ПИ на основе использования базы данных проектирования на языке



Состав и структура кросс-средств ПОМПА

Макроассемблера АЛВ. Минимальный вариант системы ПОМпА-М также ориентирован на язык АЛВ, но здесь компоненты проектируемого ПИ и текстовые документы размещаются непосредственно в файловой структуре ОС.

ПОМпА предоставляет возможность развития и наращивания состава инструментальных программных средств. Она независима от конкретной ОС, подвижна в управлении взаимодействующими задачами в среде ОС, в первую очередь в работе с программными моделями оборудования и внешней среды.

В основу системы положена методика, предусматривающая создание виртуального интерфейса для использования системных функций ОС и СУБД в инструментальных программах [3]. Виртуальный интерфейс на этапе разработки программ поддерживается набором стандартных процедур библиотек системы ПОМпА. Состав библиотек, а также все системные структуры данных описаны в документации, что позволяет разрабатывать программы на языках МАКРО П или Паскаль для последующего исполнения в рамках системы.

В процессе работы кросс-средств ПОМпА взаимодействие отдельных инструментальных программ (задач) координирует управляющая программа, обеспечивающая диалог, управление, синхронизацию, а также централизованное хранение данных. Отдельные копии управляющей и инструментальной задач связываются с каждым терминалом пользователя. Форматы диалога, реализуемые управляющей программой, синтаксически соответствуют операторам присвоения и вызова процедуры в языке Ада. Сосредоточение всего диалога в управляющей задаче позволило осуществить протоколирование и запоминание части информации, вводимой пользователем, с последующим автоматическим повторением фрагмента работы, определяемого этой информацией. К числу сервисных возможностей диалога относится также получение справочной информации.

Кроме того, в ходе диалога в системе ПОМпА-М пользователь может использовать любую команду или системную обслуживающую программу ОС РВ.

Технологический процесс проектирования в системах ПОМпА базируется на представлении проектируемого изделия и данных о ходе разработки в базе данных (БД). Поскольку внутренняя структура файлов БД не соответствует ОС РВ, то программные модули проектируемой системы (их текстовые, объектные и другие файлы) хранятся непосредственно в файловой структуре ОС, а сведения о структуре программных изделий и необходимые ссылочные связи — в структуре МИРИС [4].

Каждое ПИ (называемое далее проектом в соответствии с принятой в системах ПОМпА

терминологией) может иметь одну или несколько версий в зависимости от версий входящих в них объектов. Версии проекта связаны с соответствующими версиями модулей реестром, однозначно характеризующим данный проект.

В рабочих процедурах, выполняемых над модулями конкретного проекта, выбираются версии модулей, указанные в реестре, относящемся к проекту. В свою очередь, каждый модуль имеет паспорт, содержащий сведения о нем и ссылки на соответствующие файлы. Любое действие инструментальных программ, изменяющее состояние компонент проекта, отображается в структурах БД с учетом всех существующих связей. Содержимое структур БД можно изменить средствами интерактивной подсистемы ведения БД.

Работа в системах ПОМпА (для вариантов С и П) регламентируется подсистемой контроля за ходом разработки. Исходный момент проектирования — генерация БД, создание системных структур данных. К их числу относится таблица идентификации пользователей, содержащая фамилии, перечень средств и объектов, с которыми можно работать.

Сеанс работы каждого пользователя начинается с регистрации. Затем инициируется или создается средствами ведения БД требуемая для работы среда, в том числе указанный проект. В дальнейшем с помощью инструментальных средств можно преобразовывать отдельные объекты.

Начальные данные о плановых сроках и исполнителях отдельных работ по реализации проекта сохраняются в БД, здесь же отмечаются выполненные разработки. Таким образом система наряду с регламентацией доступа обеспечивает контроль за календарным графиком работ и анализ состояния разработки проекта.

Инструментальные средства в системах ПОМаП-С и ПОМпА-П работают в рамках описанного технологического процесса. В системе ПОМпА-М этот процесс существенно упрощен, в частности не обязательны регламентация работ пользователей и предварительное описание структуры проекта. В то же время сохранен принцип работы с объектами по именам (файл выбирается инструментальными программами).

Основное инструментальное средство системы — Макроассемблер АЛВ. Настройка его на конкретную целевую ЭВМ сочетает элементы параметрического метода (на основе формализованного описания языка ассемблера и архитектуры команд) с добавлением небольшой процедурной настройки в процессе генерации кода. Внутренние структуры данных транслируют программы для 8-, 16- и 32-разрядных ЭВМ в сегменте 32-разрядного адресного пространства.

Компоновщик систем ПОМпА формирует загрузочный модуль из отдельных компонент. Специальный формат его описания обеспечивает гибкость построения программы. Получаемая в результате компоновки программа отлаживается на кросс-средствах методом программного моделирования памяти и процессора микроЭВМ. Кроссотладчик наряду со стандартными командами просмотра и модификации элементов памяти в указываемой пользователем системе счисления предусматривает использование отладочных процедур, подсчет времени исполнения программы, распечатку областей памяти, обращение к элементам памяти по их символьным именам, обработку арифметических выражений, включающих символьные и числовые константы.

Для отладки взаимодействия программ с внешней средой и устройствами в кроссотладчике заданы четыре способа моделирования: модель-ОЗУ, модель-стандартная функция (замена моделируемого устройства ячейкой ОЗУ или стандартным устройством инструментальной ЭВМ соответственно), модель-подпрограмма (в этом случае модель устройства разрабатывается как одна из частей отлаживаемой программы) и модель-задача (автономная задача ОС, связанная с кроссотладчиком с помощью библиотек системы в рамках стандартных соглашений). Поскольку интерпретатор команд и эмулятор отладчика разделены, то допускается настройка на любой тип целевой микроЭВМ.

После завершения отладки загрузочный модуль используется в качестве исходных данных для программирования микросхем ПЗУ (или загрузки во внешнюю память целевой системы). Средства для вывода данных в программатор не включены в состав систем ПОМпА. В то же время формат файла загрузки описан в документации и для конкретного программатора нетрудно составить программу в рамках систем ПОМпА либо непосредственно в среде ОС.

Для удобства пользователя разработан ряд системных утилит — программ получения листинга в абсолютных адресах (после этапа компоновки), распечатки, просмотра и удаления отдельных файлов, относящихся к модулю.

Создаваемый в рамках системы ПОМпА-П компилятор с языка Ада обеспечит получение программы на языке Макроассемблера для дальнейшей обработки описанными инструментальными программами.

Текстовый редактор систем ПОМпА подобен известному семейству редакторов EDT и предназначен для вариантов ПОМпА-С и ПОМпА-П, в которых редактирование связано с коррекцией БД. В варианте ПОМпА-М с уровня системного диалога доступно использова-

ние любого редактора из числа работающих в ОС РВ.

Целям документирования служит подсистема выпуска текстовой документации, представляющая собой модифицированный вариант программы DOC [5], используемой в ОС РАФОС. Суть модификации состоит во введении дополнительной русской нменники директив, обеспечении полного соответствия получаемых документов требованиям стандартов ЕСПД, а также в возможности выпуска документации по ЕСКД с формированием требуемых форм и штампов. В вариантах ПОМпА, использующих БД, исходными для программ выпуска документации являются объекты БД документов (в варианте ПОМпА-М — файлы ОС РВ). В системе ПОМпА-М подсистему выпуска документации можно использовать автономно в среде ОС РВ.

Текущее состояние разработки системы: передана на серийное тиражирование базовая версия ПОМпА-М, включающая средства трансляции и отладки программ для микропроцессоров К1801ВМ1 и К1810ВМ86; разработана версия, содержащая средства трансляции и отладки программ для микропроцессора TMS-32010.

Вариант системы ПОМпА-С прошел тестирование, однако работы временно приостановлены в связи с тем, что объем оперативной памяти (256 Кбайт) стандартной конфигурации СМ ЭВМ слишком мал для совместной работы средств системы ПОМпА и СУБД МИРИС, что отрицательно сказывается на быстроте действия системы, в особенности при работе нескольких пользователей.

Создана версия ПОМпА-П с компилятором языка Ада, уточняется ее структура с учетом замечаний об использовании БД. Закончена разработка системы ПОМпА-Э, реализованной на ПЭВМ «Электроника МС0585», представляющей собой подмножество средств системы ПОМпА-М (исключена подсистема выпуска документации). Разрабатываются средства взаимодействия для систем ПОМпА-М и ПОМпА-Э на двухмашинном комплексе СМ ЭВМ — «Электроника МС0585» и переноса систем ПОМпА на ЭВМ типа СМ1700.

Телефоны 64-49-33, 63-04-94, Львов

ЛИТЕРАТУРА

1. Комплекс технологических систем автоматизации разработки программного обеспечения реального времени Прометей / Лицаев В. В., Штрик А. А., Минаев М. А. и др. // Вычислительная техника социалистических стран. — М.: Финансы и статистика. — 1986. — Вып. 19. — С. 121—128.
2. Пайл Я. Ада — язык встроенных систем / Пер. с англ.: Под ред. А. А. Красилова. — М.: Финансы и статистика, 1984. — 238 с.
3. Кройчик Г. В. Организация управления и взаимомикропроцессорные средства и системы» № 4, 1990 41

действия задач в кросс-системе разработки и отладки программ микропроцессоров ПОМнА // III Всес. конф. Автоматизация производства систем программирования. Тез. докл. ВЦАИ СССР, ВЦ СО АН СССР. Ин-т киберн. АН ЭССР.— Таллинн, 1986.— С. 75—77.

4. Кройчик Г. В., Коляда О. Н. Использование универсальной СУБД «Квант-М» при построении системы автоматизированной разработки программ для микропроцессорных систем // Всес. сем. Промышленная техно-

логия создания и применения программных средств в организационном управлении и НИОКР. Тез. докл. УНЦ АН СССР.— Свердловск, 1984. С. 185—186.

5. Валикова Л. И., Вигдорчик Г. В., Воробьева А. Ю., Лукин А. А. Операционная система СМ ЭВМ РАФСО / Под общ. ред. В. П. Семика.— М.: Финансы и статистика, 1984.— С. 72—74.

Статья поступила 13.03.89

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

УДК 681.3.06

Е. И. Казменко, С. В. Заруднева

СРЕДСТВА ФОРМАТНОГО ВВОДА ИНФОРМАЦИИ В ЭВМ

Темпы внедрения вычислительной техники во все сферы деятельности человека значительно превосходят рост числа квалифицированных пользователей ЭВМ. В этих условиях становится проблемой организация такой среды, в которой могли бы работать непрограммирующие пользователи. Необходимый минимум такой среды — средства ввода информации в ЭВМ. В ряде случаев, в частности при ведении простейших картотек, средства ввода и корректировки специальным образом структурированной информации могут являться функционально законченными приложениями. Кроме того, средства ввода — обязательная составная часть систем с большими (по сравнению с просто вводом) функциональными возможностями. В условиях стремительного роста парка персональных ЭВМ и отсутствия для них развитого программного обеспечения (ПО) средства ввода могут для некоторых приложений рассматриваться как стартовое подмножество базового ПО, ориентированного на непрограммирующего пользователя.

Для неспециалистов по автоматизированной обработке данных наиболее прост ввод данных в представленную на экране форму (форматный ввод) [1]. На основе предлагаемой архитектуры осуществлено несколько реализаций средств ввода, в том числе и для микроЭВМ [2].

Принцип непосредственного манипулирования при вводе

Считается, что непосредственное манипулирование [3] (непосредственное редактирование [4]) наиболее целесообразно для многих приложений. Для него характерны: наглядность, быстрая реакция системы на каждый введенный пользователем символ и наличие выделенных клавиш. Эти особенности позволяют организовать удобные для человека средства ввода: бланк на экране (пустографка), в который пользователь вводит информацию (набирает на клавиатуре). При построении средств форматного ввода допустимы отступления от принципа

непосредственного манипулирования: могут быть применены отличные от посимвольного (позкраниый или пореквизитный) режимы обработки информации при вводе [1]. Однако такие отступления ухудшают потребительские качества средств ввода. Наиболее перспективен посимвольный ввод, позволяющий осуществлять контроль вводимой информации после каждого нажатия клавиши пользователем [5].

Структура средств ввода. Конкретная система форматного ввода состоит из следующих основных составляющих: базового ПО (БПО), изменяемого при настройке средств ввода на конкретное приложение, для нетипового (специфического) контроля, вспомогательного ПО и метаданных, определяющих настройку БПО на ввод конкретной информации.

Представляет интерес случай, когда БПО реализует достаточно полные возможности по вводу данных, а структура метаданных допускает их разработку (и, соответственно, создание конкретной системы ввода информации) без привлечения программиста. Это очень существенно, так как жизнеспособность системы в целом в значительной степени зависит от лиц, проектирующих средства ввода (качество проектирования системы).

Рассмотрим взаимосвязь метаданных и вводимой информации. Обычно вводимые данные состоят из информационных полей (реквизитов), группируемых в записи (транзакции), размещенных в некотором архиве (базе данных, файле). Состав метаданных: описатели отдельных полей и записи в целом, описания архива (схемы базы данных).

В некоторых случаях средства форматного ввода допускают настройку на конкретное приложение при задании только одной из выделенных составляющих. Например, единственной выстраиваемой информацией может быть относящаяся к вводимой записи в целом двумерная экранная картинка. При этом ограничителя отдельных полей записи будут встроены в экранный бланк, а сама запись — формироваться выстраиванием вводи-

мых полей в память ЭВМ (друг за другом в последовательности их расположения на экране без учета защищенных частей экрана).

Другая крайность — наличие только метаданных, описывающих отдельные показатели записи. Описатель одного показателя будем называть словом описания реквизита (СОР). Совокупность СОР определяет структуру записи вводимой информации, в частности, в СОР может указываться относительное расположение реквизита внутри записи. С помощью СОР могут быть определены экранные метки, реквизиты записи которых заменяют экранную картинку.

Целесообразность ориентации средств ввода на метаданные пореквизитного уровня или описание записи в целом зависит от многих факторов. Рассмотрим общий случай, когда присутствуют обе составляющие метаданных. Уровень метаданных, описывающий характеристики всей совокупности вводимой информации, для простоты будем считать пустым.

Наиболее удобна архитектура, при которой метаданные состоят из экранной картинки (пустографки, выдаваемой на экран перед началом ввода) и совокупности (таблицы) СОР.

Вершины графа предлагаемой структуры БПО (рис. 1) соответствуют программным модулям, а стрелки указывают их подчиненность. Модуль ВВОД, ВВОЗА и ВВОРЕК — основные, реализующие непосредственно форматный ввод, остальные контролируют введенную информацию, передают готовые записи в архив и выполняют другие функции, не рассматриваемые в статье.

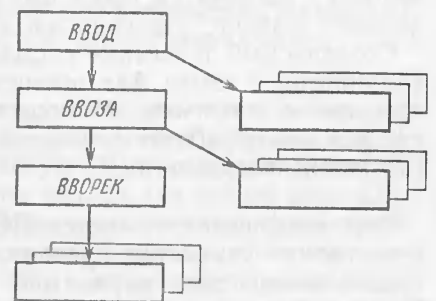


Рис. 1. Структура БПО

При работе пользователя со средствами ввода целесообразно выделить два основных состояния: командное, обрабатываемое головным модулем ВВОД, и ввода (редактирования) — ВВОЗА и ВВОРЕК. Пользователь, работая за экраном дисплея, должен визуально идентифицировать состояние, в котором находится программа ввода (требование непосредственного манипулирования).

Пользователь может определить текущее состояние программы по расположению экранного маркера (курсора). Например, в командном состоянии курсор располагается в специальном (командном) месте экрана: командная строка, верхний левый угол экрана или любое место вне пустографки. Состояние ввода идентифицируется местоположением курсора на одном из реквизитов пустографки.

Командное состояние — головное, при котором пользователь задает программе ввода команды выполнения. Основные из них: переход к редактированию записи, хранящейся в оперативной памяти, и ее вывод на внешние носители; ввод записи с внешнего носителя для корректировки; очистка буферов (и, соответственно, окон пустографки) от данных предыдущей записи; завершение сеанса ввода и др. [2].

Каждая команда инициируется пользователем нажатием соответствующей клавиши (например, команда перехода к редактированию клавишей ПРОБЕЛ).

Один из частных случаев алгоритма работы в командном состоянии и (рис. 2) может быть записан на алгоритмическом языке [7, 8] в следующем виде:

```

алгоритм ВВОД /*головной модуль*/
константы
СТОП /*значение командного символа «конец работы»*/
ПРОБЕЛ /*значение командного символа «ввод записи»*/
ВЫВОД /*значение командного символа «передача записи в архив»*/
переменные
А /*значение текущего командного символа*/
начало
вывод на экран пустографки на экран пока А ≠ СТОП

```

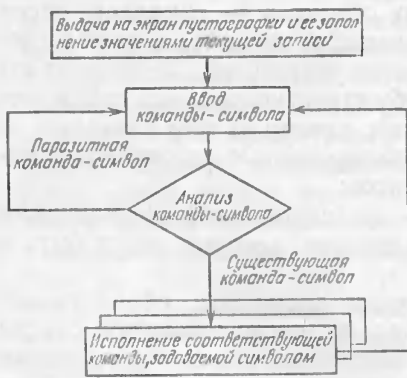


Рис. 2. Алгоритм работы средств ввода в командном состоянии

```

начало цикла
А := значение символа, введенного с клавиатуры
выбор
при А = ПРОБЕЛ: обратиться к подпрограмме редактирования записи
при А = ВЫВОД: обратиться к подпрограмме передачи готовой записи в архив
все /*конец выбора*/
конец цикла
конец

```

Подпрограмма редактирования записи работает с таблицей СОР. В терминах программирования эта таблица представляется как массив структур, где каждый элемент описывает один реквизит. Конкретное СОР состоит из параметров соответствующего реквизита. В число этих параметров могут входить: адреса реквизита в буфере ввода и сообщения, содержащего подсказку о правилах его заполнения; экранные координаты (два числа) начала отведенного под реквизит окна; длина (значность) реквизита; его тип (числовой, символьный и т. д.); адрес таблицы допустимых значений данного реквизита.

Конкретный пример описания таблицы СОР:


```

структура
/*определение нового типа переменной — структуры СОР*/
СОР ( АБ, /*адрес реквизита в буфере ввода*/
ЭХ, /*координаты реквизита*/
ЭУ, /*на экран*/
ДЛИНА /*длина реквизита*/
)

```

таблица СОР ТСОР 20 /*массив структуры/

Обязательная команда при работе со средствами форматного ввода в командном состоянии — переход к редактированию записи (рис. 3). Алгоритм подпрограммы позволяет выйти из нее в командное состояние; настроиться на следующий (предыдущий) реквизит или некоторый выделенный (полезно, если информация вводимых записей отличается только в нескольких реквизитах); получить подсказку об особенностях заполнения (ввода) текущего реквизита.

Конкретный алгоритм подпрограммы редактирования (ввода) записи:

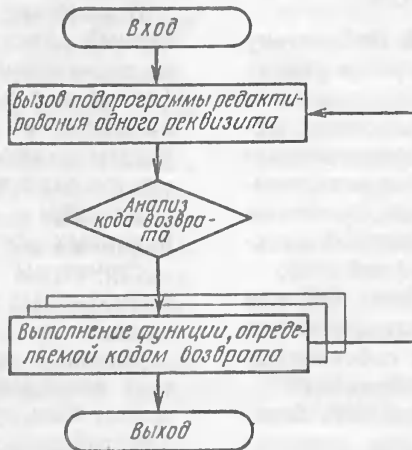


Рис. 3. Алгоритм подпрограммы редактирования записи

алгоритм ВВОЗА /*подпрограмма ввода (редактирования) записи*/

```

константы
СЛЕД. /*значение кода возврата, указывающее на необходимость перехода на следующий реквизит*/
ПРЕД. /*значение кода возврата перехода на предыдущий реквизит*/
ПЕРВЫЙ. /*значение кода возврата перехода на первый реквизит*/
ВЫХОД. /*значение кода возврата выхода в командное состояние*/

```

```

переменные
I /*номер текущего вводимого (редактируемого) реквизита*/
КВ /*код возврата*/
начало

```

```

I := 1
КВ := ПЕРВЫЙ
пока КВ ≠ ВЫХОД

```

```

начало цикла
обратиться к подпрограмме редактирования одного реквизита
выбор

```

```

при КВ = СЛЕД: I := I + 1 /* реально еще контролируется условие I больше числа реквизитов в записи */

```

```

при КВ = ПРЕД: I := I - 1 /* реально еще контролируется условие I < > I */

```

```

при КВ = ПЕРВЫЙ: I := 1
все /*конец выбора*/
конец цикла
возврат из подпрограммы ВВОЗА
конец

```

Подпрограмма редактирования отдельного реквизита (рис. 4) переносит информацию в буфер ввода:

символьная таблица БУФЕР [100], где 100 — длина записи в символах. При таком определении записи реквизит, входящий в нее, определяется как часть таблицы БУФЕР;

символьная таблица БУФЕР [10:14]. Этот реквизит может быть представлен в своем СОР значениями двух параметров: АБ = 10 (адрес буфера реквизита начинается с 10-й позиции записи) и ДЛИНА = 5.

Подпрограмме ввода одного реквизита доступны буфер ввода записи и все параметры соответствующего СОР:

```

алгоритм ВВОРЕК /*подпрограмма ввода одного реквизита*/

```

```

константы СЛЕД, ПРЕД, ПЕРВЫЙ, ВЫХОД, ПРОБЕЛ, ВЛЕВО

```

```

переменные I, КВ, /*эти переменные общие с подпрограммой ВВОЗА*/
/*указатель текущей позиции в буфере ввода*/

```

```

начало
установка указателя на начало реквизита (J := АБ) /*АБ [I]*/
установка курсора на первую позицию окна реквизита
безусловный цикл

```

```

начало цикла
КВ := значение символа, введенного с клавиатуры
выбор

```

```

при КВ = СЛЕД: выход из подпрограммы
при КВ = ПРЕД: выход из подпрограммы
при КВ = ПЕРВЫЙ: выход из подпрограммы
при КВ = ВЫХОД: выход из подпрограммы

```

```

/*предполагается, что ПРОБЕЛ имеет наименьший код
*из алфавитно-цифровых символов (АЦС)
*/

```

при КВ ≥ ПРОБЕЛ: /*обработка АЦС*/
 {БУФЕР [J] := КВ
 если J < (АВ + ДЛИНА)
 /*АВ, ДЛИНА зависят
 от I*/
 вывод значения символа
 на экран
 J: J + 1
 все
 при КВ = ВЛЕВО:
 если J ≠ 1
 J := J - 1
 смещение курсора влево
 на 1 позицию
 /*в буфере ввода и на
 экране*/
 все
 все /*конец выбора*/
 конец цикла
 конец

Предложенные средства ввода могут быть использованы практически на любой микроЭВМ. Программа [2] реализует больший набор функций, чем алгоритмы, приведенные в настоящей статье. При этом она занимает (без учета буфера ввода, таблицы СОР и располагающейся в памяти ЭВМ экранной картинке — пустографки) менее 1 Кбайт.

Важная особенность ПО форматного ввода — наращивание функциональных возможностей. В частности, рассмотренные в настоящей статье средства, обладая возможностями вызова уже введенной записи для ее модификации, являются ПО, предназначенным не только для ввода, но и для модификации (ведения различных архивов). При разделении полномочий (пользователи допускаются к разным записям), записей разных типов, отдельных реквизитов записей и реализации такого ПО не на отдельной ЭВМ, а на локальной вычислительной сети будет получена развитая система обработки экранных документов [10] или система обработки транзакций в другой терминологии.
 433510, Димитровград Ульяновской обл., НИИАР, ОВТА; тел. 3-23-24

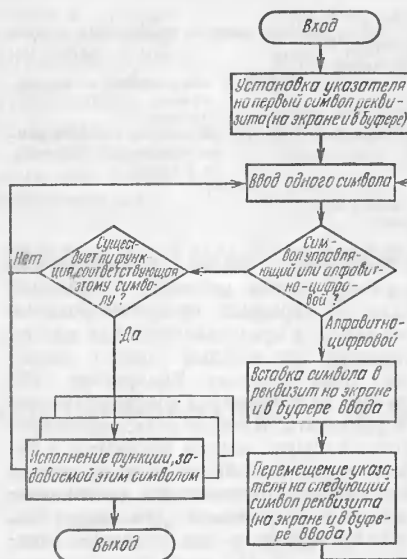


Рис. 4. Алгоритм редактирования отдельного реквизита

ЛИТЕРАТУРА

1. Казменко С. В. Системы форматного ввода информации в ЭВМ // УСИМ.— 1987.— № 1.— С. 98—102.
2. Казменко С. В., Шимбарова Л. А. Об использовании микроЭВМ в качестве интеллектуального устройства подготовки данных.— Препринт.— Димитровград, 1984.— 12 с.— (НИИАР; 15 (627)).
3. Schneiderman В. Direct manipulation: A step beyond programming languages // Computer, August.— 1983.— P. 57—69.

4. Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации // Микропроцессорные средства и системы.— 1986.— № 1.— С. 44—46, 58.
5. Thomas R. E. Maximander // Personal software.— 1983.— V. 2, N. 1.— P. 84—90.
6. Bradley В. Data-entry programming: it takes solid training // Office.— 1984.— V. 99, N. 2.— P. 81—82.
7. Основы информатики и вычислительной техники. Учебн. пособие для средн. учебн. заведений: В 2-х частях. Ч. 1 / А. П. Ершов, В. М. Монахов, С. А. Бешенков и др.; Под ред. А. П. Ершова, В. М. Монахова.— М.: Просвещение, 1985.
8. Основы информатики и вычислительной техники. Учебн. пособие для средн. учебн. заведений: В 2-х частях. Ч. 2 / А. П. Ершов, В. М. Монахов, А. А. Кузнецов и др.; Под ред. А. П. Ершова, В. М. Монахова.— М.: Просвещение, 1986.
9. Казменко С. В. Автоматизированная система разработки входных экранных документов (генератор программ форматного ввода информации АСУ) // УСИМ.— 1985.— № 2.— С. 88—93.
10. Казменко С. В. Экранные документы для общения конечного пользователя со средствами вычислительной техники. // Электронная техника. Сер. 9. Экономика и системы управления.— 1985. № 3(56).— С. 54—57.

Статья поступила 26.11.87

УДК 681.3.06

Ю. Е. Мошинский

МОБИЛЬНЫЙ ПАКЕТ ВЕДЕНИЯ ИНФОРМАЦИОННЫХ МАССИВОВ

Пакет DMF представляет собой библиотеку функций языка Си и набор программ-утилит для управления данными. Использование пакета расширяет возможности традиционных методов доступа, позволяет конструировать модели данных иерархической, сетевой и реляционной организаций, управлять разнообразными информационными объектами (от записей и текстов до множеств, списков, сетей и фреймов).

С помощью пакета DMF в файлах ОС или непосредственно на дисках создаются рабочие области, в которых организуются собственные системы файлов на индексных В-деревьях.

Файл представляет собой совокупность записей, логический порядок следования которых может определяться очередностью включения в файл и значениями элементов (ключей).

Запись состоит из одного или нескольких элементов данных (реквизитов), все или часть которых могут быть ключом (уникальным или дублируемым).

В качестве реквизита используется элементарный объект данных или агрегат, состоящий из элементарных объектов — символов, чисел (целые и с плавающей точкой), а также специальных объектов-указателей. Реквизиты-агрегаты по способу структурирования своих элементов относятся к одному из трех классов:

массивы — фиксированные совокупности однотипных элементов;

структуры — фиксированные совокупности разнотипных элементов, которые могут быть и массивами;

списки — цепочки элементов, обрабатываемых последовательно; число элементов списка может быть произвольным, в частности нулевым (пустой список); элементом списка может быть массив или структура.

Связи между записями устанавливаются с по-

мощью указателей — адресов записей. Адрес — это своего рода системный ключ быстрого доступа к записи. В течение всего времени существования записи в рабочей области ее адрес не изменяется: он может передаваться программе и запоминаться в базе данных наравне с числовой и символьной информацией, входить в ключ записи, быть элементом массива или списка.

Указатели — основной инструмент для построения схем баз данных. Например, отношение «главный — подчиненные» обеспечивается внесением в ключ подчиненных записей ссылки на их хозяина (записи того же самого или другого файла). Кортежи указателей на записи файлов-доменов образуют отношения, характерные для реляционных моделей данных. Ассоциативные связи, фреймы и подсхемы поддерживаются списками указателей.

Пакет DMF обеспечивает выборку, включение и удаление записей, позиционирование в файле по ключу, части ключа, номеру или адресу записи, поэлементную обработку, эффек-

тивную работу со списками, создание, переименование и уничтожение файлов, получение справочной информации о структуре записей, характеристиках файлов, содержании рабочих областей.

Программист может указать при открытии файла собственные подпрограммы специальной обработки, которые будут автоматически вызываться при чтении, включении и удалении записей, закрытии файла. С их помощью можно реализовать журнал транзакций, функции отката, динамический контроль непротиворечивости и полноты информации.

Программы пакета DMF написаны на языке Си, обладают высокими показателями мобильности и могут быть перенесены на любую ЭВМ и в среду любой ОС, где есть компилятор языка Си. В настоящее время DMF работает на ПЭВМ в средах MS DOS, Юникс-226; на СМ ЭВМ в средах ИНМОС, ДЕМОС и ОС РВ.

Телефоны 290-05-23, 290-82-47, Киев

Сообщение поступило 21.4.88

МАШИННАЯ ГРАФИКА

для хранения управляющей программы контроллера.

В ОЗУ статического типа объемом 8 Кбайт размещаются стек и буферы, хранятся значения рабочих величин микропрограммы. ВидеоОЗУ объемом 16 Кбайт служит для хранения поточечного представления изображения размерами $512 \times 256 \times 1$ бит, непрерывно сканируется схемами видеовывода и открыто для записи-чтения, как обычная оперативная память, во время обратного хода строчной развертки. Установкой перемычек видеоОЗУ можно сделать доступным процессору в любой момент времени, но в этом случае при одновременном обращении к видеоОЗУ процессора и схем видеовывода на экране появляются характерные помехи в виде «снега».

Порты ввода-вывода служат для связи контроллера с ЭВМ и дисплеем. Из аппаратных функций в контроллере предусмотрены средства горизонтального перемещения просмотрового окна размерами 34×256 точек с дискретностью в 8 точек и средства включения-выключения схем видеовывода, что приводит к появлению пропадания графического изображения на экране дисплея. Все остальные функциональные возможности дисплея определяются управляющей программой.

Функциональные возможности дисплея «ГАММА-5»

Контроллер находится в двух основных программно-переключаемых состояниях — «про-

УДК 681.3.01

М. Г. Брызгалова, П. В. Вельтмандер

УПРАВЛЯЮЩАЯ ПРОГРАММА ИНТЕЛЛЕКТУАЛЬНОГО ГРАФИЧЕСКОГО ТЕРМИНАЛА «ГАММА-5»

Интеллектуальный графический терминал «ГАММА-5» разрабатывался как компонент системы «массовой» интерактивной машинной графики, доводящей средства графического ввода-вывода до рабочего места массового пользователя.

Дисплей представляет собой одноплатную микроЭВМ (графический контроллер). Контроллер вставляется в свободное посадочное место в любой из алфавитно-цифровых дисплеев типа α ЕС7168 (VDT-52100-С. VDT-52130, «Электроника 15ИЭ-00-013») и несколькими монтажными доработками исходного дисплея подсоединяется к его источнику питания, схемам синхронизации, видеоусилителю и включается в разрыв последовательной линии связи между дисплеем и ЭВМ.

Основное стремление при разработке дисплея «ГАММА-5» — предоставить ему достаточные функциональные возможности для несложных приложений. Набор функций дисплея определялся исходя из накопленного опыта работы с высокооснащенными терминальными комплексами на базе мини-ЭВМ [1—8], в соответствии с возможностями разработанной для него микроЭВМ и с учетом потребностей пользователей.

Архитектура графического контроллера

Графический контроллер собран на микропроцессоре К580ВМ80, ПЗУ объемом 8 Кбайт

зрачности» и «непрозрачности». Переход из одного состояния в другое выполняется при получении от ЭВМ либо от дисплея переключающей последовательности из трех байтов.

На основе анализа кодов, используемых операционными системами ДИСПАК и ФЕЛИКС для БЭСМ-6, RSX-11M, RT-11, NTS — для мини- и микроЭВМ с системой команд «Электроника 60», ДОС4, СМ2, а также кодовых последовательностей, используемых различными программами в этих операционных системах, выбраны следующие переключающие последовательности:

007 016 017 — для перехода в непрозрачность; 007 016 006 — в прозрачность.

В состоянии прозрачности контроллер пропускает все коды от ЭВМ к терминалу и в обратном направлении без интерпретации, анализируется лишь появление переключающей последовательности.

В это состояние контроллер автоматически устанавливается после включения, следовательно, сохраняются все функции исходного дисплея, так что программы, не нуждающиеся в графическом сервисе, не должны учитывать его наличия.

В состоянии непрозрачности контроллер интерпретирует расширенный протокол дисплеев ТЕКТРОНИХ-4010. При этом предусмотрены три режима работы: алфавитно-цифровой, графического ввода и вывода.

В алфавитно-цифровом режиме практически полностью сохраняются возможности ввода-вывода исходного дисплея, за исключением нескольких служебных кодов, используемых для перемещений просмотрювого окна и графического перекрестия, задания графического режима.

В графическом режиме контроллер интерпретирует входной поток данных от ЭВМ и позволяет строить отрезки прямых, строки символов различных размеров и в нескольких ориентациях. Точность задания координат — 10 разрядов. Предусмотрена возможность задания одного из следующих типов линий: сплошные, точечные, пунктирные, штрих-пунктирные с двумя точками, разомкнутые.

Генерируемые изображения поточечно заносятся в видеоОЗУ и отображаются на экране совместно с изображениями символов исходного дисплея.

Предусмотрены два программно-выбираемых способа занесения очередной точки в видеоОЗУ: по операции ИЛИ и по операции Исключающее ИЛИ над старым и новым кодами точки с одинаковыми координатами. Результат той или иной операции заносится в видеоОЗУ.

В режиме графического ввода средства: опроса режима дисплея; разрешения перекрестия и ввода его координат

после нажатия любой алфавитно-цифровой клавиши;

опроса текущего положения перекрестия. Изображение перекрестия генерируется программно и всегда заносится в память операцией Исключающее ИЛИ. Управление позицией перекрестия может осуществляться с помощью клавиш управления символьным курсором стрелок.

Структура, состав технических средств и требования к функциональным возможностям определили структуру и состав встроенного программного обеспечения дисплея «ГАММА-5».

Программное обеспечение дисплея

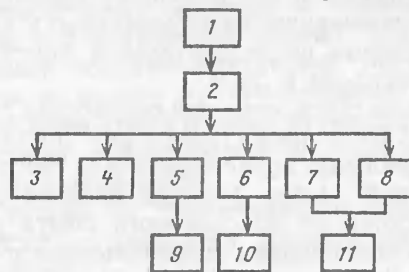
Управляющая программа дисплея «ГАММА-5» разделена на три логических уровня: обработки прерываний, интерпретации буфера ввода и интерпретации графического буфера.

На первом логическом уровне поддерживается функция ввода данных от дисплея и ЭВМ. Модули этого уровня обладают наивысшим приоритетом и служат для чтения поступающих данных и их занесения в буфер ввода.

Для части работ, например ликвидации ненужного эхо-отображения, важен порядок поступления данных от различных портов. В связи с этим выбран один буфер ввода. При поступлении данных от какого-либо порта в буфер ввода заносится байт идентификации порта и собственно байт данных.

Второй логический уровень (см. рисунок) представлен программой интерпретации содержимого буфера ввода (интерпретатором протокола). Эта программа выполняется как фоновая для работ первого уровня. Она и обеспечивает предусмотренный набор функциональных возможностей и свойств дисплея. Совокупность программ первого и второго уровней реализует транспортную функцию — передачу данных от дисплея к ЭВМ и наоборот.

В состоянии прозрачности данные из буфера ввода просто передаются соответствующему блоку вывода. В состоянии непрозрачности об-



Структура второго логического уровня программного обеспечения дисплея «ГАММА-5»:

1 — блок чтения буфера ввода; 2 — интерпретатор данных; 3 — блок управления; 4 — блок графического ввода; 5 — блок вывода порта 0 (ЭВМ); 6 — блок вывода порта 1 (терминал); 7 — блок вывода ломаных; 8 — регистр вывода порта 0; 9 — регистр вывода порта 1; 10 — регистр вывода порта 1; 11 — блок записи в графический буфер

работка данных определяется установленным режимом, активизируется соответствующий блок и данные из буфера ввода передаются выбранному блоку до обнаружения новой командной последовательности, изменяющей режим работы контроллера.

Блок управления осуществляет общее управление работой дисплея «ГАММА-5»: разрешение-запрет графического дисплея (непрозрачен-прозрачен), разрешение-запрет отображения графической информации, управление графическим окном и перекрестием, стирание изображения и текста.

Блок графического ввода служит для задания начальных координат перекрестия, опроса словосостояния графического контроллера, ввода координат перекрестия по опросу — перекрестие разрешается (появляется на экране), а значение его координаты передается в ЭВМ; ввод координат по запросу — перекрестие разрешается, затем перемещается оператором в нужную точку. После нажатия любой клавиши терминала в ЭВМ выдается код этой клавиши и координаты перекрестия.

Блок вывода ломаных активизируется при установлении графического режима и служит для пересылки команд позиционирования и построения отрезков в графический буфер.

Блок вывода графического текста при обнаружении команд задания атрибутов и собственно строк текста формирует и записывает данные в графический буфер. Предусмотрены задание ориентации вектора вертикали символа вдоль одного из четырех направлений осей координат и выбор одного из четырех направлений вычерчивания строки относительно вектора вертикали.

Третий логический уровень представлен программой-интерпретатором графического буфера, работающей как фоновая по отношению к программам первого и второго уровней. Содержимое графического буфера можно рассматривать как дисплейный файл, содержащий команды позиционирования, построения отрезков, вывода строк графического текста, задания атрибутов построений и команды управления режимом занесения в видеоОЗУ. Точность представления координат — 10 разрядов. Для полного задания координат точки требуются 4 байта (по 2 байта на координату); нулевые старшие разряды для координат концов отрезков ломаной могут не передаваться; при позиционировании задаются все, в том числе и нулевые разряды.

Протокол. Конкретная реализация протокола включает 45 команд. Команда состоит из кода операции и данных. Код операции — однобайтная, двухбайтная, трехбайтная последовательности. Используемая кодировка — КОИ-7.

Наряду с полной реализованы и частичные версии протокола, обеспечивающие преемственность дисплея по данным с дисплеями типа ТЕКТРОНИХ-4006, ТЕКТРОНИХ-4010. При наличии устройства растровой печати поддерживается получение твердой копии экрана.

Заключение. Разработано, отлажено и внедрено в эксплуатацию программное обеспечение интеллектуального графического дисплея «ГАММА-5», что позволило реализовать систему «массовой» интерактивной машинной графики. Дисплеи типа «ГАММА-5» установлены в ряде организаций различных ведомств.
Телефон 32-47-51, Новосибирск

ЛИТЕРАТУРА

1. Вельтмандер П. В. Интерактивная машинная графика в пакетах прикладных программ // *Материалы III Всесоюз. конф. «Диалог человек — ЭВМ».* — Серпухов, 1984.
2. Вельтмандер П. В. Аппаратно-ориентированные графпакеты. // *Проблемы машинной графики.* — Новосибирск: ВЦ СО АН СССР, 1982.
3. Бучнев А. А., Вельтмандер П. В., Кудряков В. Ф. Базовое программное обеспечение обработки изображений // *Тезисы докл. Всесоюз. конф. «Диалог человек — ЭВМ».* — Л., 1982. — Ч. 2.
4. Вельтмандер П. В., Кривошеев Е. А., Прошкин А. А., Сизых В. Г. Построение стереоизображений // *Там же.*
5. Бучнев А. А., Вельтмандер П. В. Аппаратно-независимые графпакеты // *Там же.*
6. Вельтмандер П. В. Распределенная система интерактивной машинной графики локальной вычислительной сети // *Материалы V школы-семинара «Интерактивные системы».* — Тбилиси, 1983.
7. Бучнев А. А., Вельтмандер П. В., Кудряков В. Ф., Мальячук А. М. Интерактивная система формирования и оперативного отображения машинных фильмов // *Материалы III Всесоюз. конф. «Диалог человек — ЭВМ».* — Серпухов, 1984.
8. Прошкин А. А. Интерактивная графическая система «Диакolor» // *Там же.*

Статья поступила 19.10.88

УДК 681.03

Б. К. Васильев

ЭКРАННЫЙ РЕДАКТОР РАСТРОВЫХ ЗАГОТОВОК

При работе с растровыми графическими системами удобно иметь готовые участки изображения при использовании команд либо логического, либо исключющего ИЛИ с изображением. Масштабирование такой заготовки и поворот ее на определенный угол должны обеспечиваться программным путем. Предлагаемая программа предназначена для создания и редактирования графических заготовок размера 8×8 или 16×16 элементов. Программа написана на Паскале, но содержит вставки на

↖ 7	↑ 8	↗ 9
← 4	Продви- жение	→ 6
↙ 1	↓ 2	↘ 3
Включение стирания 0		Рисовать •

Рис. 1. Функциональное назначение клавиш дополнительной цифровой клавиатуры

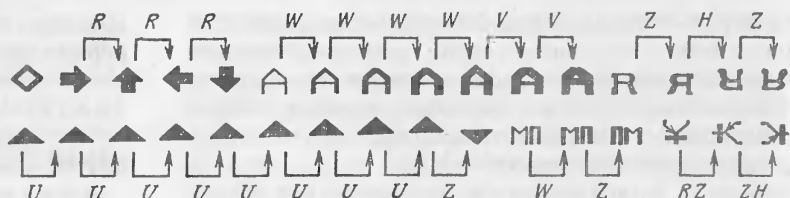


Рис. 2. Фрагмент печати таблицы заготовок: стрелками показаны заготовки, полученные преобразованиями поворота, уширения, отражения, смещения и их комбинациями

или рисунка. С помощью программы можно создавать библиотеки шрифтов и графических примитивов.

Телефон 5-77-19, Владивосток.

Статья поступила 10.10.88

УДК 681.327.12.621.397.7

Э. Э. Дворский, А. Н. Черняховский, М. М. Лезник

ВВОД ИЗОБРАЖЕНИЙ В МИКРОЭВМ С ПОМОЩЬЮ ФАКСИМИЛЬНОГО АППАРАТА «ШТРИХ»

Аппаратуру растрового считывания текстовых документов, схем, чертежей, карт можно создать на основе факсимильных аппаратов (ФА), применяемых для передачи неподвижных изображений по каналам связи. От выбора типа ФА (его технико-экономических характеристик) зависит эффективность функционирования всей системы обработки видеоданных. Оптические сканеры с механической разверткой (к которым относятся ФА) имеют ряд преимуществ по сравнению со сканерами, разработанными на иных принципах, например телекамерами. При выборе аппаратных средств ввода автоматизированных систем обработки растровой информации, считанной с бумажного носителя, такие свойства ФА, как практически постоянный коэффициент оптико-электрического преобразования по всему полю изображения, высокая линейность развертки, низкая стоимость, небольшие размеры и масса, простота эксплуатации, широкое применение, возможность работы по каналам связи, — определяющие. Известны разработки, в которых ФА использовались как устройства ввода растровой графической информации в ЭВМ. Так, в [1] описывается применение для этих целей факсимильного фототелеграфного аппарата (ФТАП) «Ладога», в [2] сообщается о системах обработки видеоданных, в состав которых входят ФТАП «Нева» и «Паллада». В настоящее время наибольшее распространение в стране получил факсимильный аппарат «Штрих» [3]. Сканер, разработанный на основе этого аппарата, обладает рядом существенных достоинств, к которым, в частности, относятся:

возможность изменения разрешающей способности сканирования по горизонтали;

ассемблере и вызовы подпрограмм библиотеки ОС РАФОС (RT-11), что делает экранный редактор зависимым от данной ОС; другие ОС могут применяться при условии написания процедур управления маркером и упаковки-распаковки рабочего массива с помощью средств конкретной операционной среды.

Данная версия программы рассчитана на работу с СМ7209 и принтером РОБОТРОН СМ6329.02М. При реализации на других типах терминалов и печатающих устройств также потребуется корректировка процедур графического вывода и позиционирования маркера на экране, написанных на ассемблере и присоединяемых как внешние подпрограммы.

Для управления маркером, перемещаемом по участку знакомест соответствующего размера, используется дополнительная цифровая клавиатура терминала (рис. 1). При этом после нажатия клавиш «5», «0», «.» продвигается маркер, пройденный участок заполняется кодом фона или точки. Процесс преобразования управляется алфавитными клавишами в соответствии с меню. Возможны следующие преобразования: смещение изображения влево, вправо, вверх и вниз; поворот на 90 град.; зеркальные отражения относительно горизонтальной и вертикальной осей; дублирование изображения со смещением на единицу по горизонтали и вертикали (уширение изображения). Созданную графическую заготовку можно поместить в таблицу или извлечь из нее для редактирования. Таблицу можно записать или вызвать с диска. На рис. 2 приведены примеры преобразований при заполнении таблицы.

Программу легко модифицировать для создания цветных графических заготовок (при наличии соответствующего терминала) при этом необходимо предусмотреть связь элементов массива А с атрибутами цвета и закрепление этих атрибутов за объектом R — цвет фона

управляемая подача по вертикали с оперативным изменением шага подачи;

высокая линейность и строгая горизонтальность строчной развертки;

возможность плавной или дискретной регулировки уровня контрастности считываемой информации;

выбор скорости сканирования.

Технические данные ФА «Штрих»:

Тип аппарата	Факсимильный для черно-белого изображения
Скорость развертки, об./мин	120 и 240
Тип развертки	Барабанный
Направление развертки	Левое
Шаг развертки, мм/об.	0,259
Разрешающая способность, линий/мм	3,8
Минимальная анализируемая толщина линий изображений оригинала, мм	0,1
Размеры передаваемого бланка, мм, не более	210×297
Полезное поле записи, мм, не менее	193×277
Напряжение, В	127 или 220
Потребляемая мощность, ВА, не более	110

При развертке барабанного типа, реализованной в ФА «Штрих», элементы растра выделяются за счет вращательного и поступательного движений (относительно светооптической системы) барабана с документом, закрепленным на его поверхности. Таким образом, за один оборот барабана считывается одна строка. Подача автоматически выключается, если весь документ прочитан. Время сканирования одного бланка при скорости считывания 240 строк/мин составляет около 6 мин.

Доработка электрической схемы ФА незначительна. Из тракта передачи видеoinформации отключаются блоки, необходимые для работы по каналу связи (модулятор, оконечный усилитель и др.). Видеосигнал снимается с узла формирователя видеосигнала блока передатчика (БПР), с помощью которого фотозлектронный умножитель (ФЭУ) согласуется с нагрузкой, следит за фоном подложки, преобразует аналоговый сигнал в дискретный по уровню и расширяет тошки линии ($0,1 \leq Z \leq 0,25$ мм) до 0,25 мм. При необходимости последнюю функцию можно исключить.

Сигнал «фазовый импульс» (ФИ), информи-

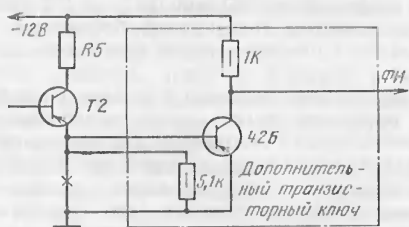


Рис. 1. Коррекция БФ ФА

рующий о начале считывания новой строки, снимается с блока фазирования (БФ) ФА. Длительность ФИ — 11 мс при скорости сканирования 120 об./мин и 5,5 мс при скорости 240 об./мин.

Для повышения линейности развертки в блоке синхронизации (БС) отключается внутренний задающий генератор и на схему синхронизации подаются импульсы частотой 400 Гц, полученные делением частоты внешнего кварцованного генератора (4 МГц).

Таким образом, необходимо доработать три блока ФА: в БС установить переключатель «синхронизация — внутренняя/синхронизация внешняя», в БФ — транзисторный ключ (рис. 1), в БПР изменить номиналы резисторов R79 и R80 на 610 кОм. Нагрузочная способность каскадов, формирующих видеосигнал и ФИ, повысится, поэтому режимы работы блоков ФА при подключении к ЭВМ не будут нарушаться.

В ФА «Штрих» не предусмотрена возможность прерывания работы с момента начала передачи информации. Для преодоления этого недостатка необходимо устранить кинематическую связь между строчной и кадровой развертками ФА (связь между вращательным и поступательным движениями документа при сканировании относительно светооптической системы). Для осуществления кадровой развертки на аппарат устанавливается дополнительный шаговый двигатель ЩДР-711 со схемой управления. Один шаг двигателя соответствует вертикальному смещению носителя относительно оптической системы на 0,0032 мм. Для получения стандартной разрешающей способности 0,259 мм/об. двигатель должен пройти 82 шага. Схема управления формирует на клеммах шагового двигателя диаграмму напряжений, соответствующую одному шагу, при появлении на ее входе одного импульса отрицательной полярности. Приводом строчной развертки по-прежнему служит синхронный двигатель ФА.

Для управления вводом графической растровой информации в микроЭВМ с шиной типа МПИ (рис. 2) разработан контроллер оптического сканера (КОС). Его основные функции: квантование видеосигнала по времени (изменение частоты квантования приводит к про-

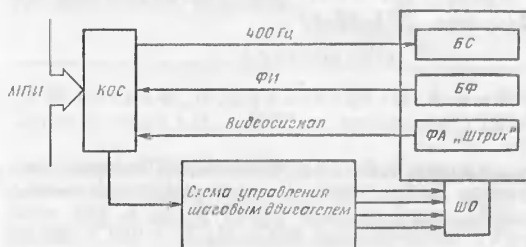


Рис. 2. Схема подключения ФА «Штрих» к микроЭВМ

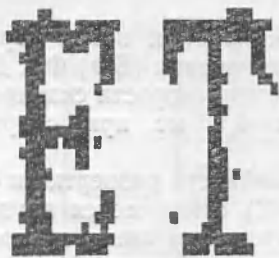


Рис. 3. Распечатка фрагмента раstra машинописного документа, введенного с ФА

УДК 681.3.06

П. А. Богомолов, А. А. Сутченков, А. И. Тихонов

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ДИАЛОГОВОЙ ГРАФИКИ ДЛЯ ДВК

порциональному изменению разрешающей способности по строке, например при скорости развертки 240 об./мин к частоте квантования 6,4 кГц разрешающая способность чтения по строке — 1600 эл./строку);

межстрочное фазирование ФА и КОС по сигналу ФИ (строчная развертка ФА синхронизируется частотой 400 Гц);

преобразование последовательного кода, полученного квантованием видеосигнала по времени, в последовательность 16-разрядных слов; управление шаговым двигателем кадровой развертки ФА;

обмен информацией с ЦП по прерываниям или по программному опросу.

Коэффициент деления частоты квантования, число шагов двигателя и момент пачала ввода очередной строки задаются программно.

Для микроЭВМ, имеющих интерфейс типа ИРПР, разработан адаптер «ФА-ИРПР». Аппаратная реализации варианта с применением адаптера значительно проще реализации системы ФА-КОС, однако уступает ей по эффективности использования процессорного времени, так как большинство функций ввода растровой информации решены программными средствами.

Система ввода графической растровой информации на основе ФА «Штрих» была использована в разработке макета читающего автомата для оконечных пунктов связи. Машинописные и рукописные текстовые документы сканировались с разрешающей способностью 0,13 мм по горизонтали и 0,19 мм по вертикали (рис. 3). Возможно применение ФА в качестве растрового принтера с большой разрешающей способностью или для организации передачи информации по каналам связи с последующим вводом ее в микроЭВМ.

252601, Киев, 110, ГСП, ул. Соломенская, 3, КОНИИС; тел. 271-44-11

ЛИТЕРАТУРА

1. Бутаков Е. А., Островский В. И., Фадеев И. Л. Обработка изображений на ЭВМ. — М.: Радио и связь, 1987.
2. Александров В. В., Горский Н. Д. Представление и обработка изображений. Рекурсивный подход. — Л.: Наука, 1985.
3. Аппарат факсимильный Ф5СБ-М1-71 и Ф5СБ-М2-71 «Штрих-М». Техническое описание и инструкция по эксплуатации.

Статья поступила 6.05.88

Комплекс инструментальных средств (библиотечных процедур на языке программирования Паскаль и диалоговых программ) существенно расширяет возможности известного графического пакета MED [1] для ДВК2М, ДВК3М, снабженных контроллером графического дисплея [2]. Он включает процедуры, позволяющие выводить на графический план дисплея штриховые штрих-пунктирные линии, а также линии произвольной толщины, выгружать в файл образы экранов, загружать их с дискеты, выводить на печатающие устройства УВВПЧ-30-004, D-100, Роботрон 6312, Роботрон 6329 графические копии экрана. Расширены функции графического курсора который может отображаться в виде небольшого перекрестия или стрелки. Предусмотрена возможность отрисовки размерных линий, ввода-вывода строк на графический план с редактированием в процессе ввода, последующим выравниванием по левому и правому краям и центрированием. Можно графически вводить целые и действительные числа в интерактивном режиме.

В составе инструментальных средств находятся также процедуры обработки меню, конструирования макетов экранов, диалогового ввода имени файла при загрузке-выгрузке графического образа экрана, задания типа принтера для печати твердой копии экрана.

Наряду с программным интерфейсом, позволяющим отрисовывать графические примитивы (отрезки прямых, дуги окружностей, спираль Архимеда, размерные линии, «заливка» ограниченной области на экране), разработан командный интерпретатор, входной информацией для которого служит описание картинки в виде набора команд, а выходной — графическое изображение на экране дисплея. Последнее позволило расширить реляционную базу данных МИКРО, созданную в ИК АН УССР, средствами работы с графической информацией.

Для повышения наглядности программ расчета электрических и тепловых режимов электротехнических устройств разработан комплекс программ визуализации электрических и электротепловых схем замещения, позволяющий в диалоге с пользователем создавать и редактировать графическое изображение схемы на экране дисплея. В каждый момент времени на экране отображается фрагмент схемы. При перемещении графического курсора, представляющего собой в данном случае инверсное изображение элемента схемы, за границу экрана отрисовывается очередной фрагмент. В библиотеку типовых элементов входят резисторы, конденсаторы, источники тока, перемычки; возможно расширение этой библиотеки. Программа отслеживает и нумерует узлы и ветви схемы.

Ограниченный объем оперативной памяти, отсутствие надежных средств отладки в значительной степени затрудняют разработку графического ПО, что привело к необходимости создания инструментального комплекса разработки графического ПО для ДВК на ЕС1840. Это достигнуто с помощью графической библиотеки, совместимой по вызовам с рассмотренными инструментальными средствами.

На основе перечисленных процедур разработана диалоговая программа представления экспериментальных данных, позволяющая отображать в графической форме функции одной переменной в линейном и логарифмическом масштабах, изменяемых в диалоге с пользователем. Одновременно можно вывести на экран до десяти графиков (данные вводятся либо с клавиатуры, либо из файла). Экспериментальные точки на графиках помечаются графическими маркерами; вводится и редактируется заголовок

рисунка, обозначения осей. Другая диалоговая программа отрисовывает поверхности, являющиеся графиками функций двух переменных в диметрической проекции; удаляет невидимые линии, позволяет вращать поверхность относительно вертикальной оси, масштабировать рисунок.

Диалоговый графический редактор кроме отрисовки графических примитивов сохраняет рисунок в виде графического образа экрана и текстового описания (редактор входит в комплекс диалоговых средств графической базы данных).

Рассмотренные инструментальные средства использованы при разработке на ДВК прикладных пакетов программ синтеза резистивных элементов потенциометров с нелинейными функциональными характеристиками, анализа тепловых режимов электротехнических устройств в условиях неопределенности, отрисовки кулачков для токарных автоматов.

Телефон 362-78-58, Москва

ЛИТЕРАТУРА

1. Бутомо И. Д., Самочадин А. В., Усанова Д. В. Программирование на алгоритмическом ПАСКАЛе для микроЭВМ.— Л.: Издательство Ленинградского университета, 1985.— 214 с.
2. Безобразов В. С., Димов В. А., Мякотин А. В., Сохранов В. Ю., Шишкевич А. А. Контроллер графического дисплея // Микропроцессорные средства и системы.— 1986.— № 3.— С. 50—52.

Сообщение поступило 25.04.89

КРАТКОЕ СООБЩЕНИЕ

УДК 681.3

О. А. Яворский, В. Н. Чернявский

ПАКЕТ ПРОГРАММ ОБРАБОТКИ ТАБЛИЧНОЙ ИНФОРМАЦИИ ДЛЯ ПЭВМ «ЭЛЕКТРОНИКА 85»

Эффективность применения ПЭВМ определяется в основном возможностями пакетов программ, получивших из всех компонентов ПО наибольшее развитие. Самые распространенные пакеты — пакеты программ табличной обработки — могут найти применение в различных сферах народного хозяйства, таких как бухгалтерский учет, плановые и статистические расчеты, анализ хозяйственной деятельности предприятий и организаций, обработка результатов экспериментов, инженерные расчеты.

Для профессиональной ПЭВМ «Электроника 85» разработан пакет программ обработки табличной информации, функционирующий в операционной среде ИНМОС-85. Пакет ТАБЛИЦА-85 включает средства, позволяющие выполнять следующие основные функции:

создания и изменения табличных структур с заданным числом колонок и строк, добавления и удаления колонок и строк, изменения ширины колонок. Максимальный размер таблицы 64 колонки и 256 строк;

ввода данных (текстов, чисел и формул) в ячейки таблицы и редактирования (удаление, замена данных). Имеется возможность ввода данных по маршруту и защиты их от случайного изменения;

выполнения вычислительных операций с данными. В любой из ячеек таблицы может находиться формула (описание вычислительных операций над числовыми константами, содержимым других ячеек для получения значения данной ячейки). Формулы могут содержать математические выражения, имена элементарных функций, ссылки

на значения данных, размещенных в других таблицах; упорядочения данных в колонках и строках;

копирования и перемещения данных в таблице и из других таблиц;

поиска данных по значениям в таблице с возможной их заменой;

формирования данных, включающих в себя выравнивание данных (центрирование, прижатие вправо или влево), представление чисел (с десятичной точкой, целое, с экспонентой), выбор шрифтов (нормальный, жирный, подчеркивание);

вывода таблицы или ее частей на печатающее устройство.

Данные для таблиц могут храниться на жестком магнитном диске типа «винчестер» и гибких магнитных дисках. Доступ к таблицам возможен через каталог устройств и файлов. В каталог файлов отбираются файлы пакетов ТАБЛИЦА-85, ГРАФ-85, БАЗА-85.

Пакет с развитыми средствами организации общения с пользователем ориентирован на пользователя-непрограммиста. «Дружественный» характер взаимодействия пользователя с пакетом имеет следующие особенности:

полиэкранный режим работы, заключающийся в создании нескольких рабочих зон на экране монитора, каждая из которых рассматривается как независимый экран для обработки таблиц или выполнения служебных функций. Рабочая область экрана — окно в таблицу. Можно перемещать таблицу в заданных направлениях, имитируя таким образом движение окна по таблице для ее просмотра, и выбирать необходимые для работы объекты;

в процессе работы пользователь видит объект, с которым собирается выполнять действия, и результаты выполнения каждой команды;

стандартные и часто используемые команды пакета программ реализуются через функциональную клавиатуру. Основные функции выполняются путем выбора их в меню операций, имеющем три уровня. Головное меню постоянно находится в первой строке экрана, а вспомогательные меню появляются в рабочей области экрана и исчезают после выполнения операции;

на любом этапе работы можно получить справочную и поясняющую информацию применительно к текущей ситуации. Это обеспечивает легкость обучения пользователя работе с пакетом программ;

при ошибках, связанных с действиями пользователя, можно получить информацию о их типе и способах устранения;

работа с клавишами движения в таблице, меню, каталоге устройств, каталоге файлов, а также в строке редактирования организована по единому принципу. Возможны пошаговые перемещения вправо, влево, вверх, вниз. Сочетания клавиш «Движение кадров», «Движение к границе» с клавишами пошагового движения позволяют ускорить перемещение по объекту и создают удобства для пользователя.

Пакет ТАБЛИЦА-85 совместим с пакетами деловой графики (ГРАФ-85), организации и ведения баз данных реляционного типа (БАЗА-85). Программы пакета написаны на языке Сп. С третьего квартала 1988 г. пакет ТАБЛИЦА-85 поставляется ведомственным фондом алгоритмов и программ ИПИ АН СССР.

Телефон 3-63-83, Бердянск

Сообщение поступило 26.08.88

РЕКЛАМА

НИИ Ядерной Физики МГУ выполняет работы по переносу данных из СУБД, работающих на ЭВМ ЕС, ДВК, Электроника, СМ и др., в СУБД, предназначенные для РС/XT, РС/АТ.

УДК 681.326

С. А. Нестеренко, В. А. Кравцов

**МИКРОПРОЦЕССОРНАЯ СИСТЕМА
ПРОЕКТИРОВАНИЯ И ОТЛАДКИ
МИКРОКОНТРОЛЛЕРОВ**

Система предназначена для разработки контроллеров различного прикладного назначения на основе ОЭВМ серии К1816 [1, 2]. Она позволяет отлаживать как программные, так и аппаратные средства разрабатываемого контроллера, а также выполнять финишную отладку спроектированного устройства [3].

В состав системы кроме базовой ПЭВМ входят отладочный модуль (ОМ), программатор и блок внутрисхемного эмулятора (рис. 1). ОМ предназначен для эмуляции работы программных и аппаратных средств проектируемого контроллера. Конструктивно он состоит из платы процессорного ядра и факультативной платы. Плата процессорного ядра содержит ОЭВМ, 4 Кбайт ОЗУ, эмулирующее ПЗУ программ контроллера, внешнее ОЗУ данных объемом 256 байт, схему сопряжения с магистралью МПИ, схему управления режимами работы (рис. 2). ОЗУ программ и ОЗУ данных выполнены по двухпортовой схеме, что позволяет обращаться к ним инструментальной ПЭВМ и ОЭВМ. Внутренняя магистраль и порты ОЭВМ процессорного ядра усилены и выведены на внешний разъем для подключения факультативной платы, на которой расположены периферийные схемы. Двухплатная структура ОМ дает возможность при неизменном процессорном ядре отлаживать аппаратную и программную части контроллеров различного функционального назначения благодаря подключению соответствующих факультативных плат.

Модуль внутрисхемного эмулятора необходим для финишной отладки программно-аппаратных средств проектируемого контроллера [3]. Он снабжен гибким кабелем с контакт-

ной колодкой, которая вставляется в гнездо для подключения ОЭВМ на плате спроектированного контроллера. Модуль имеет возможности по эмуляции любого из элементов процессорного ядра (ОЭВМ, память программ и данных), отсутствующих на плате контроллера во время наладки. Модуль программатора ориентирован на запись информации в РПЗУ серии К573, а также во внутреннее ПЗУ ОЭВМ серии К1816.

Описанные выше модули процессорного ядра, внутрисхемного эмулятора и программатора выполнены в виде одинарных плат конструктива микроЭВМ «Электроника 60», что позволяет размещать их резидентно в корпусе ПЭВМ ДВК2М. Факультативная плата подключается с помощью гибкого кабеля.

Разработаны платы процессорного ядра на базе ОЭВМ К1816ВЕ48 и К1816ВЕ51. Создан набор факультативных плат, включающий модули цифрового и аналогового ввода-вывода, преобразователя частоты и временных интервалов в код, последовательного интерфейса, устройства связи с клавиатурой и индикацией. Модуль внутрисхемного эмулятора разработан в двух модификациях, ориентированных на БИС К1816ВЕ48 и К1816ВЕ51 соответственно.

При использовании процессорного ядра и эмулятора на основе ОЭВМ К1816ВЕ48 ограничения по расположению программ пользователя отсутствуют, т. е. отлаживаемые программные модули могут располагаться в любой части адресуемой памяти ОЭВМ. Для ядра и эмулятора на основе ОЭВМ К1816ВЕ51 отлаживаемая программа должна исключить ячейки 19...21, а также старшие два килобайта адре-

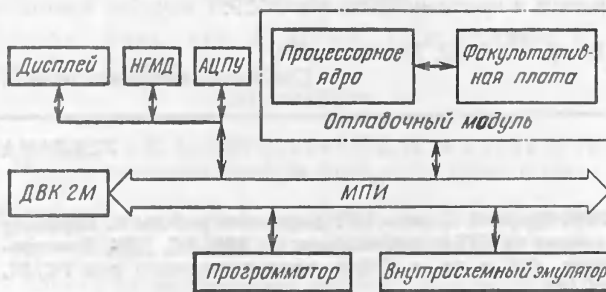


Рис. 1. Структура системы проектирования

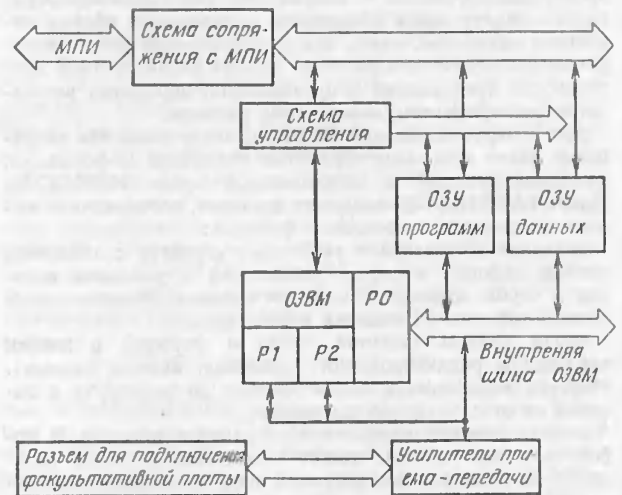


Рис. 2. Структура отладочного модуля

суммой памяти, что связано с нахождением в памяти отладочного монитора.

Программное обеспечение системы проектирования поддерживает работу аппаратуры, автономную разработку программного обеспечения проектируемого контроллера и предоставляет пользователю удобный интерфейс взаимодействия с разрабатываемым устройством.

Программный комплекс включает кросс-систему программирования, программу-отладчик, программы обеспечения процесса программирования ПЗУ и внутрисхемной эмуляции.

Кросс-система программирования содержит экранный редактор, кроссоасемблер и компоновщик МИКРОС. Программа-отладчик обеспечивает:

- загрузку с диска в память ПЭВМ;
- копирование из памяти ПЭВМ в ОЗУ программ процессорного ядра;
- запуск программы, загруженной в ОЗУ процессорного ядра на выполнение с любого адреса;
- установку адреса останова;
- пошаговое выполнение команды с индикацией результата на экране дисплея ПЭВМ (рис. 3);
- дизассемблирование программы, записанной в память ПЭВМ;
- просмотр и изменение содержимого внутренних элементов ОЭВМ (PSW, аккумулятора, регистров, стека и ячеек внутреннего ОЗУ данных);

просмотр и изменение содержимого ОЗУ программ в виде мнемокодов команд и в шестнадцатеричном виде;

просмотр и изменение содержимого внешнего ОЗУ данных в шестнадцатеричном виде;

копирование отлаживаемой программы из памяти ПЭВМ на диск и АЦПУ;

диагностику аппаратных компонент процессорного ядра и факультативной платы.

Выполнение отладчиком той или иной функции программируется в диалоговом режиме путем ввода команд с клавиатуры ПЭВМ. Команды представляют собой ключевой символ и параметры (адреса и данные в шестнадцатеричной форме или имена файлов). В качестве ПЗУ используются микросхемы серии К573. Исходными данными для программирования является файл, полученный с помощью отладчика программы, хранящийся на диске.

Программа внутрисхемной эмуляции имитирует ресурсы процессорного ядра (ОЭВМ, ПЗУ программ, ОЗУ данных), отсутствующие на плате пользователя и позволяет проверить техническое состояние аппаратуры, установленной на разрабатываемой плате, с помощью имеющихся диагностических процедур.

Программный комплекс написан на языке ассемблера МАКРО 11 и функционирует в среде ОС RT-11. Программа-отладчик занимает в за-

				PSW	ACC	P1	P2	OP	ADR
				---	---	---	---	---	---
РН	000	MOV	A, #02H	08	02	FF	FF		
РН	002	MOV	R0, #FEH	08	02	FF	FF	FE	
РН	004	MOVX	@R0, A	08	02	FF	FF	02	FE
РН	005	ANL	A, #07H	08	02	FF	FF		

Рис. 3. Фрагмент трассы отлаживаемой программы

грузочном формате 10 Кбайт оперативной памяти, программа программирования ПЗУ — 4 Кбайт, программа внутрисхемной эмуляции — 12 Кбайт. Программа трассировки, написанная на языке ассемблера ОЭВМ К1816 и обеспечивающая доступ к внутренним элементам ОЭВМ, входит в состав загрузочных модулей программ отладчика и эмулятора.

Разработанный комплекс программно-аппаратных средств представляет собой интеллектуальную систему проектирования для автоматизации разработки контроллеров на базе ОЭВМ серии К1816. Развитые функциональные возможности системы позволяют отлаживать как программную, так и аппаратную части проектируемого контроллера, проводить финишную отладку устройства непосредственно в конструктиве пользователя.

270044, Одесса, пр. Шевченко, 1, ОПИ;
тел. 28-87-00, 28-85-79

ЛИТЕРАТУРА

1. Кушнир В. А., Панфилов Д. И., Шаронин С. Г. Многофункциональный комплекс программно-аппаратных средств для семейства однокристалльных ЭВМ серии К1816. // Микропроцессорные средства и системы.— 1988.— № 4.— С. 3—6.
2. Домнин С. Б., Иванов Е. А., Муренко Л. Л. Средства комплексной отладки микропроцессорных устройств.— М.: Энергоатомиздат, 1988.
3. Нестеренко С. А. Методы внутрисхемной эмуляции при отладке микропроцессорных систем.— М.: ЦНИИ «Электроника», 1981.— 47 с.
4. Белов А. М., Иванов Е. А., Муренко Л. Л. Средства автоматизации программирования микропроцессорных устройств.— М.: Энергоатомиздат, 1988.

Статья поступила 21.03.89

УДК 681.326

А. В. Бедарев, В. С. Гравов

ОТЛАДОЧНЫЙ ПУЛЬТ ДЛЯ УСТРОЙСТВ НА ОЭВМ

Пульт предназначен для проверки работоспособности устройств на базе ОЭВМ КР1816ВЕ35, КМ1816ВЕ48, КМ1816ВЕ49. Он позволяет переводить устройство во время работы из режима непрерывного исполнения программы в покомандный и обратно, останавливать исполнение программы по заданному адресу. В покомандном режиме на индикаторах

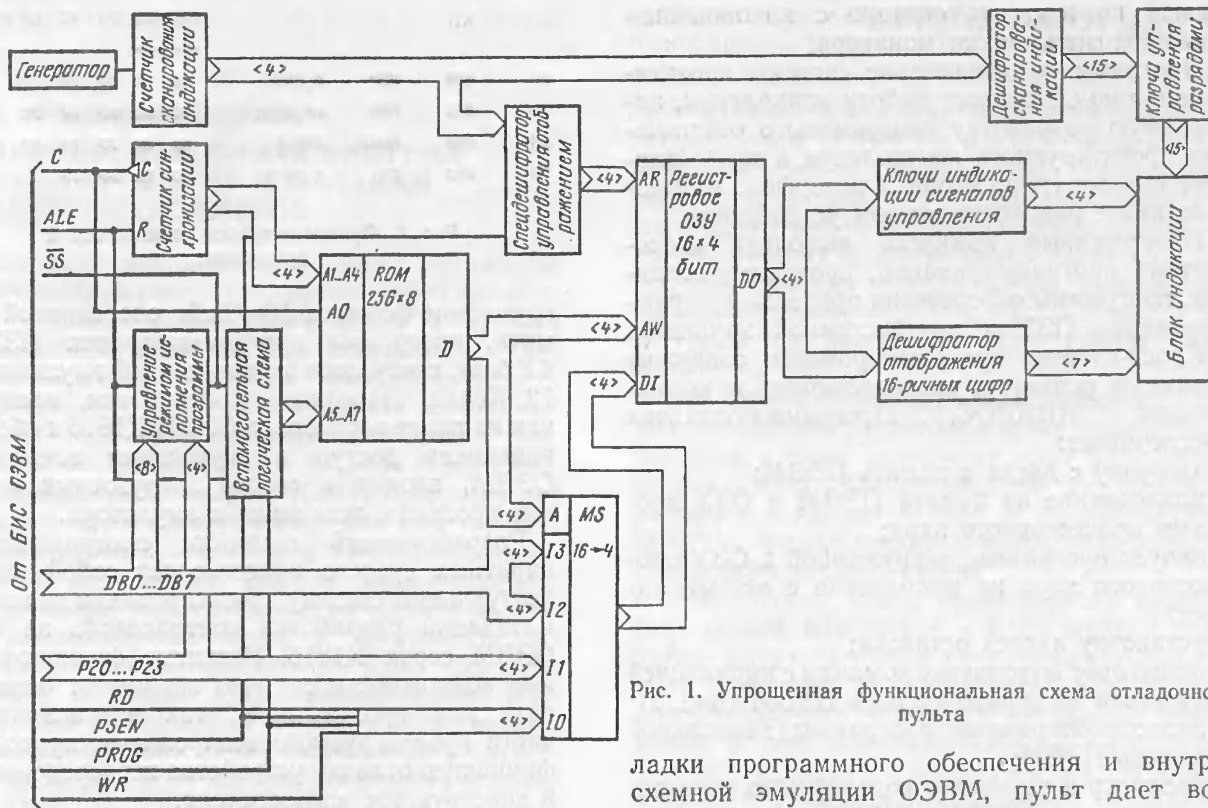


Рис. 1. Упрощенная функциональная схема отладочного пульта

пульта можно наблюдать адрес, данные и сигналы управления (PSEN, PROG, RD, WR) в каждом из циклов последней исполненной команды, а также адрес следующей команды. Общий объем отображаемой информации составляет 13 шестнадцатеричных цифр и восемь дискретных сигналов для двухцикловой команды или восемь шестнадцатеричных цифр и четыре дискретных сигнала для одноцикловой.

Пульт позволяет контролировать работу любого устройства на ОЭВМ с реальной программой, записанной во внешнем ЗУ программ, и может быть подключен к нему с помощью клипсы, надетой на БИС ОЭВМ, проходной панельки со шлейфом или любого разъема, соединенного непосредственно с выводами ОЭВМ, если этот разъем имеется в проверяемом устройстве.

Пульт фиксирует информацию, реально присутствующую на выводах шины данных, младшей части порта P2, управляющих сигналов ОЭВМ в каждом цикле команды. Эта информация весьма ценна при отладке устройства (рис. 1).

В отличие от отладочных устройств ОУ-48, ОУ-49*, предназначенных в основном для от-

ладки программного обеспечения и внутрисхемной эмуляции ОЭВМ, пульт дает возможность обнаружить и локализовать аппаратные неисправности, в том числе и тех частей устройства, которые не проверяются или не используются при работе с ОУ-48 и ОУ-49 (сама ОЭВМ, цепи подключения внешнего ЗУ программ, тактовый генератор). Таким образом пульт служит удачным дополнением существующих технических средств разработки и обслуживания устройств на ОЭВМ.

Пульт работает по следующему принципу: информация запоминается в определенные моменты времени при исполнении команды и после завершения исполнения команды воспроизводится на индикаторах пульта в виде, удобном для восприятия оператора.

В составе пульта имеется узел управления режимом исполнения программы, который используется для выбора непрерывного или покомандного исполнения программы, а также останова по адресу, заданному на 12-разрядном тумблерном регистре (рис. 2). Режимы ОЭВМ управляются посредством формирования в нужные моменты времени сигнала останова ОЭВМ SS путем установки в единичное состояние триггера D2. В покомандном режиме это происходит в первом цикле каждой команды по сигналу ALE в режиме прогона с остановом по заданному адресу — при фиксации факта совпадения реального адреса с заданным единичным сигналом на выходе триггера D2. Для продолжения программы следует нажать кнопку «пуск», записать единицу в

* Кушнир В. Е., Панфилов Д. И., Шаронин С. Г. Многофункциональный комплекс программно-аппаратных средств для семейства однокристалльных ЭВМ серии K1816 // Микропроцессорные средства и системы. — 1988. — № 4.

триггер D2 и снять сигнал \overline{SS} . Все анализируемые сигналы поступают через 4-канальный 4-разрядный мультиплексор на вход данных регистрового ЗУ (РЗУ) с независимым управлением записью и считыванием информации, имеющего организацию 16×4 бит. Момент регистрации информации, источник и адрес записи в РЗУ определяются счетчиком синхронизации и адресуемым ПЗУ с организацией 256×8 бит. Счетчик синхронизации устанавливается в начальное состояние сигналом ALE и фиксирует такты тактового генератора ОЭВМ проверяемого устройства. Его содержимое определяют пять младших разрядов адреса ПЗУ, остальные разряды генерируются узлом вспомогательной логики с учетом значений сигналов PSEN, PROG, ALE и порядкового номера цикла внутри команды.

Процесс записи информации с шины данных в РЗУ не зависит от наличия сигналов RD и WR ОЭВМ, поэтому при их отсутствии или неправильной генерации информация с шины данных будет правильно записана, зафиксирована и отображена на индикаторах пульта. Отображение информации функционально не связано с ее записью и осуществляется по принципу динамической индикации. Адрес индицируемой в данный момент ячейки РЗУ определяется счетчиком сканирования со специализированным дешифратором управления отображением, реализованным на ПЗУ.

Дополнительная функция дешифратора управления отображением — гашение части разрядов индикации при отображении процесса исполнения одноцикловой команды. Разряд на блоке индикации, в котором отображается информация из адресованной в данный момент ячейки РЗУ, определяется дешифратором сканирования индикации. Информация,

логически объединенная в полубайты (адреса, данные), дешифрируется при считывании из РЗУ дешифратором отображения шестнадцатеричных цифр и выводится на 7-сегментные индикаторы, а остальная проходит в блок индикации через ключи индикации сигналов управления и отображается на дискретных светодиодах. Частота сканирования индикации определяется генератором, входящим в схему пульта.

Пульт имеет один внешний источник питания 5 В с током до 2 А и реализован на 33 микросхемах серий K155, K555, K556, K589, 13 светодиодных матрицах и 9 дискретных светодиодах. Изготовлено и эксплуатируется в течение двух лет на кафедре вычислительной и информационно-измерительной техники Московского института нефти и газа им. И. М. Губкина.

Телефон 930-90-09, Москва

Статья поступила 13.04.89

УДК 681.3.06:326

Ю. М. Руденко, В. Г. Жиганов, А. Я. Мосин

СИСТЕМА ТЕСТИРОВАНИЯ И ОТЛАДКИ ОЭВМ СЕРИИ K1816

Система тестирования и отладки (СТО) однокристалльных ЭВМ (ОЭВМ) K1816BE48 и KM1816BE035 предназначена для проверки правильности выполнения команд и отладки программ на реальном образце. Программы представляются в виде абсолютных кодов, которые можно предварительно получить и отладить на логическом уровне с помощью кросс-средств, например «МИКРОСС 048» [1].

Имея информацию о правильности выполнения команд, можно говорить о возможности использования ОЭВМ в качестве элемента управления. В этом случае тестирование достаточно вести до первой неработающей команды. Но при изготовлении ОЭВМ или использовании дефектной требуются сведения о работоспособности каждой команды.

Возможность использования дефектной ОЭВМ определяется классом решаемых задач и набором неработающих команд. В качестве примера можно привести библиотеку стандартных программ [2], в которой задействовано около 40 % команд ОЭВМ. Рассматриваемая ниже система тестирования рассчитана на оба случая, хотя для первого она несколько избыточна (на выявление остальных дефектов после обнаружения первого тратится дополнительно около 1 с машинного времени и 225 байт памяти).

СТО рассчитана на ДВК2, 3, 4 или микроЭВМ «Электроника 60» и состоит из отладочного устройства (ОУ) и комплекса тестовых

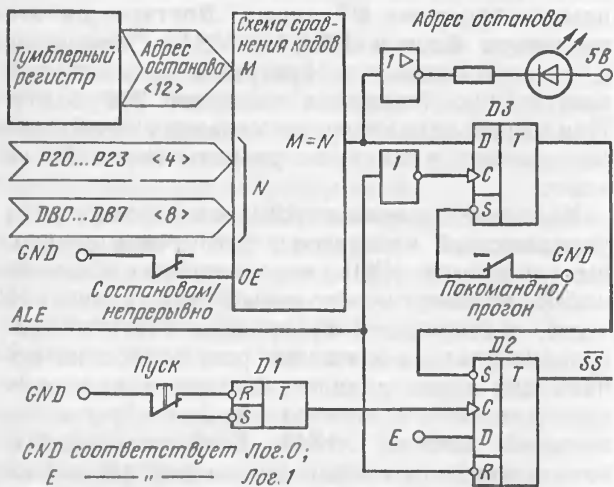
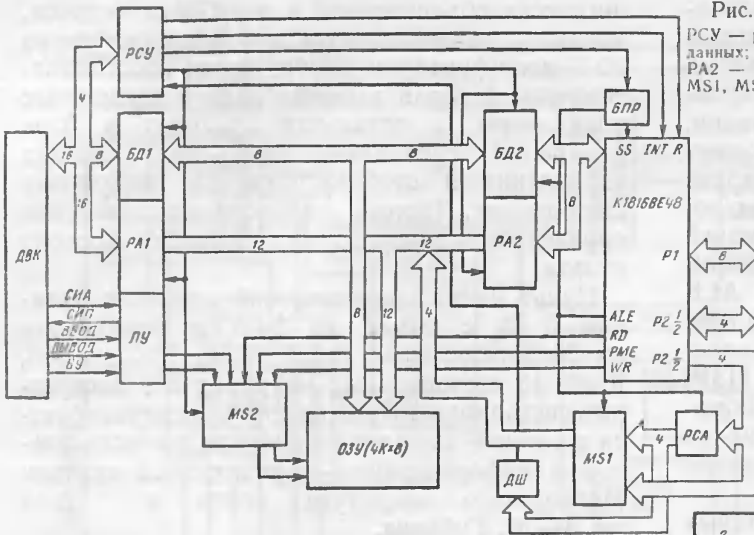


Рис. 2. Схема 12-разрядного тумблерного регистра

Рис. 1. Структурная схема отладочного устройства:

PCU — регистр состояния и управления; БД1, БД2 — 8-разрядные буферы данных; PA1 — 16-разрядный регистр адреса; ЛУ — логика управления; PA2 — 8-разрядный регистр адреса; PCA — регистр старших адресов; MS1, MS2 — мультиплексоры; БПР — блок пошагового режима; ДШ — дешифратор



программ (ТП) (рис. 1). Оперативная память доступна со стороны ДВК2 и ОЭВМ. Мультиплексор MS2 переключает сигналы управления памятью.

PCU содержит следующие разряды: P0 — сброс ОЭВМ (R); P1 — прерывание ОЭВМ (INT); P2 — переключение памяти; P3 — пошаговый режим-прогон; P4 — флаг ОЭВМ.

Разряды PCU P0...P4 доступны со стороны ДВК2 для записи и чтения (разряд P4 устанавливается ОЭВМ и считывается ДВК2). Адрес PCU — 177776.

На начальном этапе в разряде P0 устанавливается единица, в P2 — нуль. При этом вырабатывается сигнал «Сброс», поступающий на вход «P» ОЭВМ, и ОЗУ подключается к каналу ДВК2. Память занимает адресное пространство 160000...167777. Со стороны ДВК в ОЗУ заносится программа работы ОЭВМ. Четные адреса памяти ДВК 160000...167776 образуют младший банк, нечетные 160001...167777 — старший банк памяти ОЭВМ.

При записи в P0 нуля, в P2 единицы сигнал сброса ОЭВМ снимается и ОЗУ подключается к ОЭВМ, начинающей работу с адреса 0.

В пошаговом режиме после загрузки памяти со стороны ДВК в разряде P3 устанавливается единица и после переключения памяти организуется тактирование ОЭВМ. Разряд P1 дает возможность вызывать прерывание ОЭВМ в автономном режиме. Если при выполнении программы ОЭВМ необходимо обратиться к ДВК2, то с ее стороны записывается единица в разряд P4. Флаговый разряд устанавливается в том случае, если ОЭВМ обнуляет разряды 4...7 порта P2.

ОЗУ подключается к ДВК2 и ОЭВМ через элементы с третьим состоянием, переключаемые разрядом P2. Схема подключения позво-

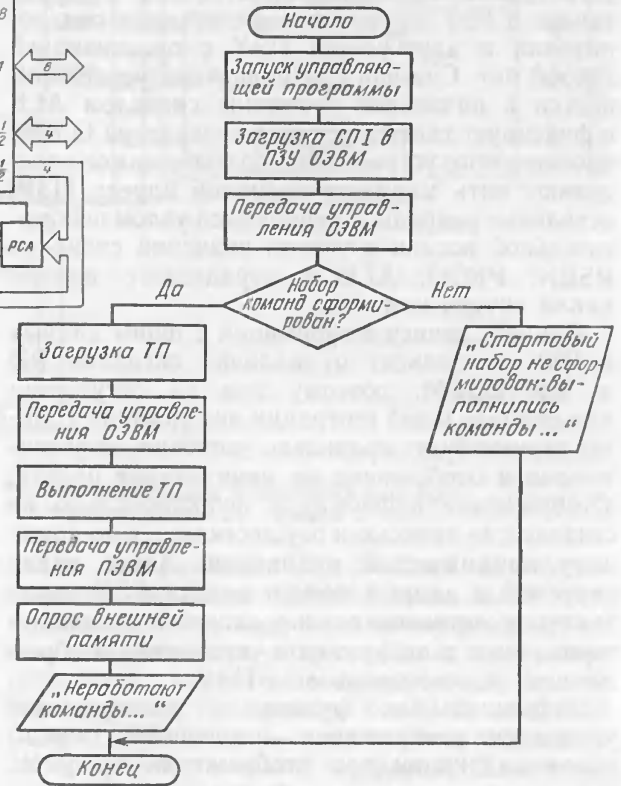


Рис. 2. Алгоритм тестирования ОЭВМ: СПИ — имя стартовой программы, соответствующей I-ой заменяемой команде (I=1,15).

ляет использовать весь объем ОЗУ в качестве памяти программ и данных. Достигается это введением блоков PCA и MS1. Обращение к памяти данных и программ организуется постранично (страница занимает 256 байт). При переходе на другую страницу необходимо записать в младшие разряды порта P2 ее адрес.

Комплекс программ тестирования состоит из управляющей программы, загрузчика стартовых программ (СП), определяющих базовый набор команд, необходимый для написания теста, тестирующей программы (ТП) и программы анализа и выдачи результатов на печать или экран дисплея. Исходная информация для работы последней фиксируется во внешней памяти ОЭВМ. В базовый набор входят команды считывания-записи данных из внешней памяти, команды засылки-чтения непосредственных данных, очистки, инвертирования аккумулятора, условные переходы по ну-

левому и ненулевому значениям аккумулятора, безусловному переходу и т. д.— всего 15 команд. Если в программе обнаружена неправильно работающая команда (программа выдает неверный результат или прерывается по таймеру-счетчику, то вызывается другая программа, в которой происходит замена какой-либо команды, начиная, например с первой. Замены последовательно до тех пор, пока не будет сформирован базовый набор команд или не отработают все стартовые программы.

Такая стратегия построения базового набора не исчерпывает всех возможностей его построения. Можно было бы попытаться пойти на некоторую дополнительную замену неработающей команды (ситуация, когда неработающая команда заменяется на неработающую) или выявить две-три неработающие команды в выбираемом базовом наборе. Но необходимо учитывать, что в этом случае объем требуемой памяти и затрачиваемое время тестирования катастрофически возрастают. Так, для поиска двух неработающих команд требуется 4200 байт памяти.

Если базовый набор команд сформирован, то ОЗУ тестируется по методике, описанной в [3]. Определяются участки памяти, не имеющие дефектов, и на них размещается ТП. Работа по созданию базового набора команд протоколируется во внешней памяти. ТП проверяет правильность выполнения каждой команды или, когда это невозможно, пары команд и формирует данные во внешней памяти ОЭВМ для работы программы анализа и выдачи результатов.

Наиболее сложны для тестирования команды условных и безусловных переходов (их тестирование вынесено в конец программы). Это объясняется тем, что в ТП последовательно проверяется очередная команда и в случае ее правильной работы соответствующий бит внешней памяти переводится в единицу. При заикливаниях программы и выходе из цикла по прерыванию таймера-счетчика часть команд может оказаться непроверенной.

При разработке ТП заманчивой казалась идея применить метод раскрутки, присоединяя к базовому набору команды, прошедшие тестирование. Но учитывая достаточно жесткие ограничения на ПЗУ ОЭВМ, пришлось отказаться от этого, так как каждое использование той или иной команды, не входящей в базовый набор, требует опроса состояния соответствующего бита внешней памяти. Выбран, на наш взгляд, более рациональный способ введения в работу команд, не входящих в базовый набор. В каждый участок программы, анализирующий правильность выполнения команды, при необходимости вводилась одна или две

команды. Для каждой вновь введенной команды таких участков было несколько. Они выбирались таким образом, чтобы программа анализа и выдачи результатов могла в большинстве случаев однозначно определить, какая команда работает неправильно (рис. 2).

Максимальный объем ПЗУ, занимаемый ТП составляет примерно 1,4 Кбайт. Время выполнения программ на ОЭВМ — порядка 3 с. Отладочное устройство изготовлено в виде стандартной полуплаты конструктива ДВК2. При отладке этого комплекса были выявлены ошибки в программно-логической модели, имитирующей выполнение команд RETR и JMPP.

Испытание комплекса на реальных образцах, имеющих в наличии у разработчиков, помогло обнаружить неработающую команду JMPP. Систему можно весьма эффективно использовать как средство проверки исправности ОЭВМ при серийном и массовом выпуске, а также приборов и устройств, включающих ОЭВМ в качестве управляющего элемента.

Телефон 434-94-65, Москва, с 14 до 18 ч.

ЛИТЕРАТУРА

1. Белов А. М., Иванов Е. А., Муренко Л. Л. Комплексы кросспрограмм «Электроника МИКРОСС» // Микропроцессорные средства и системы.— 1986.— № 3.— С. 27.
2. Руденко Ю. М., Чернова Т. Ф. Библиотека стандартных программ для БИС микроЭВМ серии K1816 // Микропроцессорные средства и системы.— 1987.— № 6.— С. 22—25.
3. Тюлькин С. П. Программа тестирования ОЗУ // Микропроцессорные средства и системы.— 1987.— № 1.— С. 54—56.

Статья поступила 25.04.89

УДК 681.327.2

В. С. Семенов, С. Г. Шипилов, Е. А. Сафронов,
О. А. Зиновейкин

АППАРАТНО-ПРОГРАММНЫЙ ОТЛАДОЧНЫЙ КОМПЛЕКС НА БАЗЕ КОНТРОЛЛЕРА «ЭЛЕКТРОНИКА МС 2702-1»

В настоящее время существует большое многообразие аппаратно-программных отладочных комплексов на основе микропроцессорных средств с фиксированной системой команд [1]: оценочные модули ограниченной мощности, разноразличные, универсальные.

В основу рассматриваемого отладочного комплекса (рис. 1) положен принцип прямой микропроцессорной связи [2]. Организация канала прямой связи обеспечивает перенос файлов из системы разделения времени в

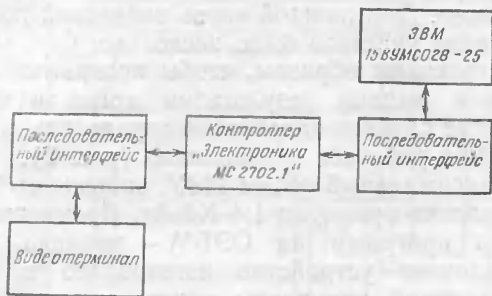


Рис. 1. Структурная схема комплекса

память контроллера. Контроллер «Электроника МС 2702-1» — одноплатная микроЭВМ, изготовленная на базе МПК БИС КР580 и предназначенная для управления различным технологическим оборудованием. Для повышения его функциональных возможностей разработана встроенная плата модуля ОЗУ и модифицирован монитор [3].

Комплекс работает в двух режимах: автономном (контроллер и терминал) и связи с ЭВМ верхнего уровня (полный состав). В автономном режиме комплекс позволяет вводить программы (данные с клавиатуры терминала), выводить на терминал и изменять содержимое ячеек ОЗУ и регистров МП, отлаживать программы, находящиеся в ОЗУ контроллера.

В режиме связи с ЭВМ верхнего уровня оператор имеет возможность работать за терминалом непосредственно с 15 ВУМС,

что позволяет создавать и редактировать исходные тексты программы, транслировать исходные тексты кроссассемблером, хранить файлы на устройстве прямого доступа (при необходимости оттранслированные программы можно загрузить в ОЗУ контроллера и инициировать их исполнение).

Для обеспечения возможностей работы с ЭВМ верхнего уровня разработан модуль ОЗУ и интерфейса (рис. 2). В его состав входят: ОЗУ в виде одной или двух страниц по 4 Кбайт; двунаправленный шинный формирователь; схема управления; универсальные синхронно-асинхронные приемопередатчики (USART1, USART2F); генератор тактовых импульсов, позволяющий работать с видеотерминалами, имеющими различное быстродействие (1200, 2400, 4800 и 9600 Бод); преобразователь уровня.

Системное программное обеспечение контроллера «Электроника МС 2702-1» включает в себя монитор (2 Кбайт), находящийся в ПЗУ. Он позволяет проверять и (или) изменять содержимое ячеек ОЗУ или регистров микропроцессора, загружать программы (данные) в ОЗУ вручную (автономный режим) или с ЭВМ верхнего уровня, выполнять программы с использованием или без использования точек прерывания.

Для работы монитора в основном (автономном) режиме необходимо иметь контроллер «Электроника МС 2702-1» и алфавитно-цифровой дисплей, в режиме связи с ЭВМ верхнего уровня — внешнюю мини- или микро-ЭВМ.

Для трансляции исходных текстов программ можно использовать как ассемблер (в частности, кроссмикроассемблер ASM85, который по своим основным характеристикам аналогичен резидентному ассемблеру ASM системы CP/M [4] и ASM80 системы ДОС1800 для микро-ЭВМ семейства CM1800 [5]), так и языки высокого уровня (например, Си и транслятор Rat с определенного подмножества этого языка [6]). Комплекс может быть дополнен внутрисхемным эмулятором, что существенно расширяет его функциональные возможности.

443100, Куйбышев, ул. Молодогвардейская, 244 КПТИ, имени В. В. Куйбышева, корп. 8, ком. 308, каф. вычислительной техники; тел. 37-12-86.

ЛИТЕРАТУРА

- Иванников А. Д. Аппаратно-программные комплексы для отладки микропроцессорных устройств // Микроэлектроника и полупроводниковые приборы / Под ред. А. А. Васенкова, Я. А. Федотова. — М.: Радио и связь, 1981. — Вып. 6. — С. 62—82.
- Денкер Дж. Прямая микропроцессорная связь для загрузки программ в системах разделения времени. Электроника, 1977. — Т. 50. — № 9. — С. 67—68.

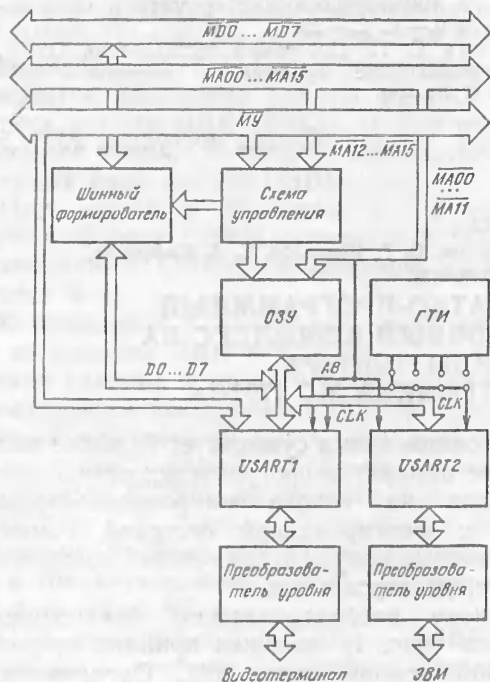


Рис. 2. Структурная схема модуля ОЗУ и интерфейса

3. Зеленко Г. С., Панов П. С., Попов С. В. Радиолюбителю о микропроцессорах // Радио, 1983.— № 11, 12.
4. Уэйт М., Ангермейер Дж. Операционная система CP/14.— М.: Радио и связь, 1986.— С. 321.
5. МикроЭВМ СМ1800.— М.: Финансы и статистика, 1984.
6. Берри Р., Микиз Б. Язык Си: введение для программистов / Пер. с англ.— М.: Финансы и статистика, 1988.

Статья поступила 17.10.88

УДК 681.3.06

С. С. Селицкий, М. Ю. Сыркин

КОМПЛЕКТ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ОТЛАДКИ

Система отладки ОЭВМ К1816ВЕ48 состоит из инструментальной ЭВМ общего назначения СМ1800 и внутрисхемного эмулятора ОУ-48 [1], работающего в режиме ON-LINE (дуплексной связи).

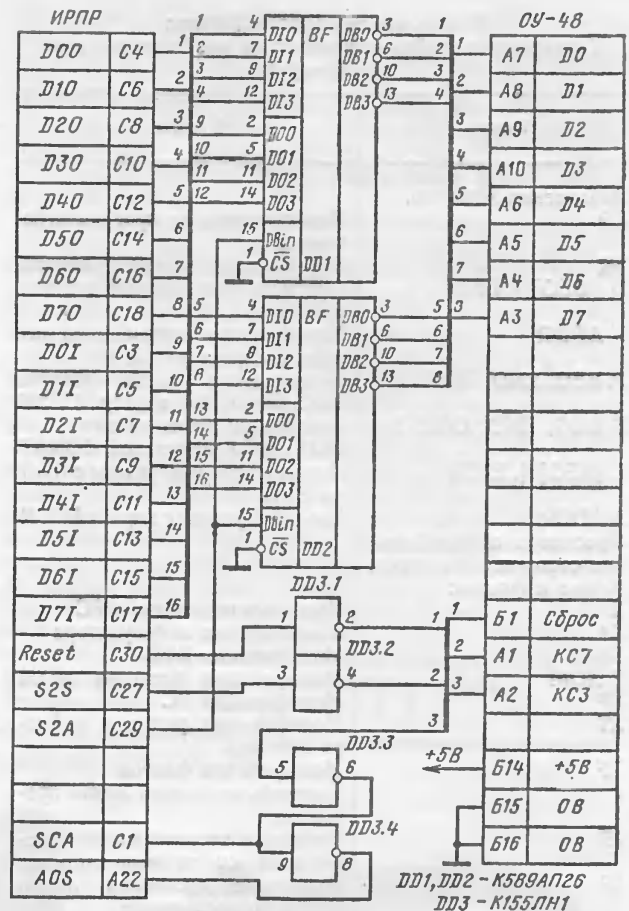
Для обеспечения замкнутого технологического цикла отладки ПО авторами разработан пакет кроссового ПО для ОЭВМ К1816ВЕ48. В него входят:

кроссотладчик DEB48, включающий в себя эмулятор, однопроходный ассемблер, дизассемблер, утилиты чтения-записи 16-ричного файла в формате INTEL. Кроссотладчик предназначен для автономной отладки программ и может быть поставлен на любую инструментальную машину с CP/M-совместимой ОС; программа IOU48, предназначенная для работы с отладочным устройством ОУ-48 в режиме дистанционного управления. ОУ-48 сопрягается с вычислительным комплексом по каналу ИРПР. Одни и те же команды пользователя в DEB48 работают с внутренней памятью эмулятора, а в IOU48 — с памятью отладочного устройства. Поэтому возможна как комплексная отладка программ, так и отладка программ реального времени;

программа вывода 16-ричного файла на перфоленту в формате отображения памяти (для последующего прожига в ПЗУ).

При подготовке программ для ОЭВМ можно пользоваться кроссассемблером XASM48 фирмы UVOCET SYSTEM, так как форматы его выходного файла и входных файлов программ DEB48 и IOU48 совместимы. Загрузочные модули хорошо читаются и редактируются в 16-ричном формате.

Программы DEB48 и IOU48 имеют идентичный пользовательский интерфейс, однако одна из них работает с отлаживаемой программой в режиме чисто программной эмуляции микроЭВМ К1816ВЕ48, а другая — в режиме ON-LINE со специализированным устройством ОУ-48. Для сопряжения инструментальной микроЭВМ СМ1800 с ОУ-48 используется стандартный модуль ИРПР СМ1800 с небольшой аппаратной доработкой в виде переход-



Принципиальная схема переходной платы

ной платы (см. рисунок), совмещенные шины ввода-вывода ОУ-48 стыкуются с разделенной шиной ввода-вывода модуля ИРПР СМ1800.

Меню команд DEB48 и IOU48 стандартное для отладочных комплексов такого класса (см. таблицу). Особенность кроссотладчика — возможность редактирования программ в мнемонической форме с помощью встроенного ассемблера и дизассемблера. Ассемблер использует «мягкий» формат строки (поля мнемоники команды могут разделяться произвольным числом табуляций и пробелов). Ошибочные строки помечаются сообщением об ошибке — «?», затем пользователю предлагается повторить ввод. В процессе ввода работает стандартный строчный редактор ОС CP/M. Выполнение длительных по времени команд кроссотладчика (например, дампа больших участков памяти) при необходимости можно прервать нажатием любой клавиши на клавиатуре (при нажатии клавиши I в режиме прерывания программы имитируется приход внешнего прерывания).

В отличие от существующих отладочных комплексов и кросс-систем, функционирующих под ОС РАФОС, ДВК, РВ или ДОС1800 [2],

**Список команд кроссотладчика
(квадратные скобки указывают на необязательность параметра)**

Команда	Функция
Управления памятью: BP	Переключение на программную память
BR	Переключение на память данных
D [BEG] [, END]	Просмотр содержимого памяти с BEG по END
S ADDR	Модификация содержимого памяти с адреса ADDR
M BEG, END, DEST	Перемещение блока памяти BEG...END по адресу DEST
F BEG, END, CONST	Заполнение блока памяти BEG...END константой CONST
L [BEG] [, END]	Вывод текста программы с BEG по END
A ADDR	Ввод программы с адреса ADDR
Просмотра и модификации состояния регистров, портов и флагов: X	Просмотр регистров и PC
XA	Модификация аккумулятора
XB	Модификация BUS
X NUM	Модификация регистра NUM
XP	Модификация PC
XT	Модификация регистра таймера-счетчика
XF	Модификация флагов
I	Контроль состояния линий прерывания
IT	Изменение состояния линии прерывания от таймера-счетчика
IE	Изменение состояния линии внешнего прерывания
P	Просмотр состояния портов
P NUM	Модификация состояния порта NUM
T	Просмотр T0, T1, TF
T0	Изменение состояния T0
T1	Изменение состояния T1
TF	Изменение состояния TF
Трассировки, работы с контрольными точками, загрузки и выгрузки файлов: G [BEG] [, END]	Выполнение программы с BEG по END
U [N]	Выполнение N команд в режиме трассировки
C ADDR, +	Установка признака контрольной точки
C [ADDR], -	Сброс признака контрольной точки (или всех)
C [ADDR]	Просмотр признака контрольной точки (или всех)
R	Загрузка HEX-файла
W BEG, END	Сохранение блока BEG...END в HEX-файле
E	Выход в ОС
?	Список допустимых команд

программный комплекс DEB48+IOU48 работает в среде ОС CP/M. При этом имеются возможности:

работы с текстами с помощью мощного экранного редактора ОС CP/M (ED, WM, WS); трансляции программ с ассемблера кросс-

ассемблером ASM48 фирмы AVOCET SYSTEM (поддерживаются метки, арифметические выражения и псевдокоманды);

отладки базовой логики программы в режиме эмуляции кроссотладчиком DEB48 и в режиме реального времени с использованием внутрисхемного эмулятора ОУ-48;

сохранения результатов (образа памяти) на диске в 16-ричном формате;

вывода программы на перфоленду для последующего прожига в ПЗУ или ее перевода непосредственно в РПЗУ кристалла ОЭВМ с помощью программатора ОУ-48.

Недостаток инструментального комплекса — невозможность компоновки программы из заранее оттранслированных модулей. Сборка программы производится на уровне исходных текстов (библиотек, подпрограмм поддержки) с использованием псевдокоманды макроассемблера INCLUDE или средств экранного редактора.

В качестве инструментальной машины вместо СМ1800 можно использовать любую машину с ОС CP/M (плата сопряжения проектируется в соответствии с используемым интерфейсом, в программе связи с внутрисхемным эмулятором изменяется драйвер обмена).

Москва, Волоколамское шоссе, 73, ВНИИСтройдормаш; тел. 490-06-23

ЛИТЕРАТУРА

1. Отладочное устройство ОУ-48. Руководство пользователя.
2. Каширин В. М., Понтак М. С., Рубенштейн Д. И., Смолкин В. С. Комплекс для отладки систем на базе однокристалльных ЭВМ серии K1816 // Микропроцессорные средства и системы.— 1987.— № 6.— С. 53—55.

Статья поступила 11.11.88

681.3.06

А. А. Ивлев, Е. В. Панкрац

СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЛОКАЛЬНОГО МНОГОМАШИННОГО КОМПЛЕКСА

Программное обеспечение рассчитано на работу совместимых ЭВМ «Электроника 79», «Электроника 85», ДВК4, СМ4, СМ1420 и т. п. в составе локального многомашинного комплекса под управлением ОС РАФОС, ФОДОС, RT11, NTS, TSX. Комплекс построен в соответствии с топологией типа «звезда» и включает одну центральную ЭВМ с развитым набором периферийных устройств и группу ЭВМ-сателлитов, для которых наличие дисковых ЗУ необязательно*. В качестве технических средств межмашинного обмена используются практически все серийные интерфейсы (мультиплексоры, ИРПС, ИРПР, СПИ-15 и т. п.), а также различные интерфейсы в стандарте КАМАК (MOD1, 350-305, 500A, 505 и т. п.). Обеспечивается доступ системных и пользовательских программ ЭВМ-сателлитов к устройствам и файлам на центральной ЭВМ при помощи стандартных команд оператора и стандартных программных запросов ввода-вывода ОС РАФОС, а также дистанционная загрузка прикладных программ

* Ивлев А. А., Панкрац Е. В. и др. Локальный многомашинный комплекс мини- и микроЭВМ с топологией связей типа «звезда» // Управляющие системы и машины.— 1987.— № 1.— С. 55—57. 88.

и (или) ОС в бездискую ЭВМ-спутник. Структурно ПО подразделяется на ПО центральной ЭВМ и ПО ЭВМ-спутника. В состав ПО центральной ЭВМ входят программа обслуживания запросов (ПОЗ) и драйверы линий связи. ПОЗ функционирует как оперативное или автономное задание и принимает при помощи драйверов линий связи запросы ЭВМ-спутников, которые обслуживаются по принципу: первый пришел — первый вышел. ПОЗ вместе с драйверами линий связи загружается и переходит в состояние пассивного ожидания, активизируясь только на время обслуживания запросов.

В состав ПО ЭВМ-спутников, оснащенных дисковыми ЗУ, входят драйверы удаленных устройств (дисков, принтеров, магнитных лент), которые транслируют запрос на центральную ЭВМ и получают результат исполнения. Для бездискующих ЭВМ-спутников возможна либо загрузка ОС и программ пользователя с устройств центральной ЭВМ (в этом случае вся дальнейшая работа ведется так же, как в варианте с дисками), либо загрузка (дистанционная или локальная) автономных программ, т. е. программ, не выдающих запросов к ОС. В последнем случае для взаимодействия ЭВМ-спутников с центральной ЭВМ используется пакет подпрограмм обмена.

Для работы под управлением ОС TSX или NTS набор выполняемых функций несколько расширен. Доработанная версия ПО позволяет ЭВМ-спутникам работать в режимах протокола и удаленного терминала.

В режиме протокола программа на ЭВМ-спутнике взаимодействует с ПОЗ на центральной ЭВМ; функции режима протокола наряду с операциями файлового и блочного обмена включают запуск автономных заданий на

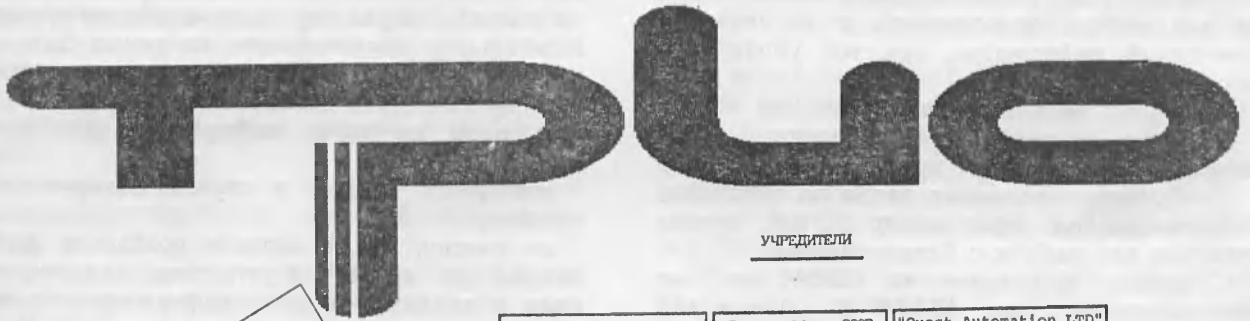
центральной ЭВМ по программным запросам ЭВМ-спутника и обмен сообщениями между программами на ЭВМ-спутнике и центральной. Для ЭВМ-спутника с дисковыми ЗУ этот режим реализуется с помощью запросов SPFUN к драйверу удаленного диска; для бездискующих ЭВМ-спутников — с помощью пакета подпрограмм обмена. Протокол обмена очень простой, поэтому скорость передачи данных ограничена быстродействием используемых интерфейсов связи.

В режиме удаленного терминала ЭВМ-спутник взаимодействует непосредственно с мультипрограммным монитором центральной ЭВМ. Работа в этом режиме поддерживается программой UTCOM, драйвером коммуникации XL, программой эмуляции терминала TEM. Оператор ЭВМ-спутника может чередовать режимы работы.

Созданы две версии ПОЗ: «терминальная звезда» и «смешанная звезда». В первом случае возможно взаимодействие с интерфейсом при поддержке терминальных драйверов ОС TSX и NTS (мультиплексоры, ИРПС и т. п.); во втором — с нестандартными интерфейсами, например модулями КАМАК. В ПОЗ терминальной звезды драйверами линий связи на центральной ЭВМ служат коммуникационные драйверы из состава ОС (например, для TSX это драйвер CL). Всем линиям связи доступны режимы удаленного терминала и протокола. В ПОЗ смешанной звезды каждой линии связи соответствует отдельно разработанный драйвер, а режимы удаленного терминала и протокола доступны только терминальным линиям. Для нестандартных линий возможен только режим протокола. ПО поставляется «под ключ» по договорным ценам.

Телефон 460-41-47, Москва. Ивлев А. А.

СОВМЕСТНОЕ СОВЕТСКО - БРИТАНСКОЕ ПРЕДПРИЯТИЕ



УЧРЕДИТЕЛИ

НПО "Химмаштехнология"

Промстройбанк СССР

"Quest Automation LTD"
Великобритания

СП "ТРИО" реализует:

- Высокоскоростные деловые персональные компьютеры "Arg1000" "Сетевой сборник"
- Рулонные и планшетные графопостройтели форматов А4-А1
- А4-А0
- Фотопостройтели форматов А4-А1
- Матричные и лазерные принтеры
- Графические мониторы высокого разрешения
- Сканеры высокого разрешения
- Дигитайзеры
- Телефаксы
- и другую разнообразную периферию

СП "ТРИО" специализируется на:

- поставке "под ключ" автоматизированных рабочих мест для машиностроительного проектирования; планово-экономических работ, редакторско-издательской деятельности и др. с полным программным обеспечением.
- установке локальных вычислительных сетей с пакетами программ "Novell" позволяющими "увязать" до 100 пользователей.

СП "ТРИО" обеспечивает:

- оптовые покупателям торговые скидки от персональной цены, а также возможность часть сумм оплатить в содружбных счетах со дня оплаты
- установку и наладку локальных вычислительных сетей
- обучение пользователей работе на них
- полное гарантийное обслуживание поштучной техникой в течение 2 лет.
- постгарантийное обслуживание в течение 5 лет за содружбной

СП "ТРИО" гарантирует:

- высококлассный сервис и индивидуальное выполнение ВСЕХ работ на себя обязательство и гарантии



ОПЛАТА

В свободно-конвертируемой валюте I категории (код 128) при заказе больших партий возможна оплата в иностранных валютах (индийская, филиппинская).

НАШ АДРЕС :

12312, Москва, ул. Выборгская 16
Телефон : 150-99-34, 150-88-82
Факс : 1509852
Телекс : 207325 "АМРА"

УДК 681.13

Л. В. Иванова

ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА «АРХИВ»

В настоящее время на 8-разрядных ПЭВМ прикладные системы для разных видов работ, связанных с обработкой информации, создаются в основном в виде автономных функционально-ориентированных систем в связи с ограниченным ресурсом этих машин.

Несмотря на автономность целесообразно при их разработке учитывать требования, которые предъявляются к интегрированным системам, а именно:

создаваемые системой файлы данных должны быть переносимы на другие прикладные системы, работающие на данной ПЭВМ;

единый стиль взаимодействия пользователя с большинством работающих на данной ПЭВМ систем предусматривает минимум манипуляций на клавиатуре, однотипность и наглядность выводимой информации, простое управление ходом диалога;

интерфейс пользователя на русском языке. Этому в основном удовлетворяют многие широко используемые прикладные системы.

Для решения основных задач по обработке информации на 8-разрядных ПЭВМ нужны средства для работы с базами данных.

С начала 80-х годов на ПЭВМ активно используется система dBASE II, однако для 8-разрядных ПЭВМ существуют определенные особенности ее применения.

Во-первых, dBASE II требует при активной работе около 40 Кбайт оперативной памяти. Поскольку оперативная память используется на пределе, в ряде случаев отмечается ненадежная работа системы, очевидно, из-за «наползания» области стека на область данных, что вызывает разрушение памяти и аварийный срыв работы.

Во-вторых, язык манипуляции данными dBASE II ориентирован на естественный английский язык. Из-за этого часто появляется необходимость участия программиста в создании пользовательской среды. Пользовательский интерфейс системы составлен на русском языке и продолжает интерфейс текстового процессора ТЕКСТ [1]: содержит типовые меню, те же команды перемещения курсора, работает со строковыми данными, имеет структуру файлов данных, которая согласуется с данными

dBASE II и систем программирования на языках BASIC, Паскаль. Система соответствует ресурсу 8-разрядных ПЭВМ класса Роботрон 1715, поскольку программа при работе занимает в ОЗУ не более половины (25 Кбайт) области транзитных программ, оставляя достаточно места для промежуточных данных системы.

Информационно-поисковая система (ИПС) «Архив» предназначена для создания и ведения персональных баз данных типа картотеки, архива и т. п. Она обеспечивает построение вводимых данных непосредственно на экране, ввод, проверку, быстрый поиск информации, хранение файлов на дискетах. Система ориентирована на пользователя-неспециалиста в области программирования; работа ведется в диалоговом режиме и сопровождается подробными пояснениями, которые выводятся на экран по желанию пользователя.

ИПС можно использовать в качестве средства ввода данных для программ, написанных на языках программирования высокого уровня. Наибольшая достоверность вводимых данных обеспечивается за счет того, что при построении формы можно назначить проверки:

выводом на экран информации для просмотра;

повторным вводом в случае обнаружения ошибки;

по списку, когда заранее создается файл данных со всеми допустимыми значениями поля и вводимая в поле информация проверяется на соответствие этим значениям. Если вводимое значение не обнаружено в указанном файле, оно отвергается.

Кроме того, маски редактирования позволяют вводить в определенные позиции поля только определенные символы. Ошибки при вводе обнаруживаются и исправляются. В выходной файл поступают только записи, прошедшие проверку.

Для проверки числовых полей данных предусмотрены:

контрольная цифра, обеспечивающая ввод только тех числовых данных, значения которых подчиняются правилу контрольной цифры (в данной системе проверяемые числовые значения должны делиться на 11 без остатка);

на соответствие вводимых данных области допустимых значений (min-max).

Высокая скорость обработки данных достигается за счет того, что в процессе ввода данных в новую форму можно:

использовать поля, заполняющиеся автома-

тически данными из уже существующих файлов или полученных в результате вычислений над другими полями формы, что также увеличивает достоверность вводимой информации;

копировать поля из предыдущей записи;

автоматически определять положение десятичной точки.

При вводе данных в базу можно назначить пакетный режим. Это дает возможность сначала закончить работу по вводу, а затем проверить весь пакет. В основной файл переносятся записи, прошедшие проверку, что увеличивает скорость ввода данных.

ИПС «Архив» обеспечивает выполнение арифметических и строчных (объединение) операций над полями формы, поддерживает несколько способов поиска и обновления информации в базе данных:

последовательный просмотр файла данных в порядке поступления записей в файл;

просмотр отсортированных записей;

выбор записи по заданному ключу.

При просмотре можно изменить или уничтожить запись, распечатать форму с данными или отдельно данные, установить маску просмотра для вывода на экран только определенных записей.

Система «Архив» состоит из трех программ:

«Форма» — для разработки формы, которая будет использоваться в работе с данными, и определения спецификаций для полей формы;

«Архив» — для ввода данных и работы с ними;

«Пакет» — для пакетной обработки данных и поддержки некоторых функций программы «Архив».

Программы «Форма» и «Архив» вызываются автономно из операционной системы или одна из другой.

Система «Архив» имеет два режима работы — создание формы и работа с базой данных, которые соответствуют работе программ «Форма» и «Архив» соответственно.

На первом этапе создается форма (файл определения формы), по которой будет формироваться база данных. На втором этапе создается база данных и организуется работа с ней.

Вид формы произвольный (определяется потребностями пользователя и видом данных). Размер создаваемой формы — до 255 символов в ширину и длину. Форма может содержать до 245 полей данных.

На первом этапе в распоряжении пользователя имеются четыре уровня помощи (страницы пояснений), которые устанавливаются последовательным вводом команды "И", начиная со страницы 4.

Страница 4 появляется на экране после ввода нового имени файла определения формы и дает общую информацию о процессе создания

формы и о работе программы «Форма». Если файл определения формы с указанным именем уже существует на диске, то на экран выводится страница 3 в виде меню с основными командами перемещения курсора, вставки-удаления символов и некоторыми другими. Дополнительные команды для включения и удаления полей данных выводятся в меню страницы 2 после ввода команды "И". Страница 1 показывает только строку состояний, которая дает информацию о положении курсора (номер строки и столбца) и поле данных (номер, длина, номер позиции, символы маски редактирования).

Форма — незаполненный бланк, который содержит пустые поля для заполнения и вспомогательный текст, поясняющий, какие данные следует вводить в каждое поле. Все слова и разделительные линии относятся к вспомогательному тексту. Он вводится в форму в нужные места и будет храниться в файле определения формы как часть формы. Поля данных создаются клавишей подчеркивания и определяют области, куда вводятся данные. Совокупность всех полей формы представляет запись файла данных. Только информация, введенная в поля данных, хранится в файле данных.

Работа с формой ведется в режиме экранного диалога. Экран условно разделен на три области: строку состояния, окно меню и окно формы.

Во время разработки формы верхняя строка экрана отображает текущее состояние системы — на какой строке, в каком столбце и поле формы находится курсор, длина текущего поля, позиция курсора в поле. При перемещении курсора по форме значения в строке состояния изменяются.

Все команды меню представляются управляющими символами, т. е. для ввода команды необходимо одновременно нажать клавишу <Упр. символ>.

Перемещая курсор по экрану и вводя в нужные места вспомогательный текст и поля, пользователь создает форму, удобную для работы с данными. Если имеется уже готовый вариант формы, его можно редактировать аналогично редактированию текста с помощью текстового процессора: вставлять или удалять символ, строку или столбец формы, расширять, сокращать или удалять поле формы и т. п.

Наиболее важный этап создания формы — назначение атрибутов полям. Для этого курсор устанавливается в нужное поле и вводится по команде "К или "Р.

По команде "К полю формы назначается атрибут ключевого поля, которое отмечается символами «*». Если атрибут ключевого поля назначен нескольким полям, то они объединяются в единый ключ, по которому и ведется индекс системы. При создании формы необхо-

димо назначить хотя бы одно ключевое поле.

По команде "Р система задает более двадцати вопросов относительно содержимого текущего поля.

Вопросы, как правило, требуют однозначного ответа — «да» или «нет» — и описывают порядок обхода полей в форме, тип поля, способ его заполнения, проверки или вычисления, шаблоны для заполнения поля и вывода данных на экран.

Каждому полю формы назначается свой номер. При вводе или редактировании данных курсор перемещается по полям формы в порядке назначения номеров. Ключевым полям также назначаются номера, в порядке следования которых несколько ключевых полей объединяются в единый ключ.

Содержимое поля можно скопировать из предыдущей записи или из файла данных, уже существующего на диске, атрибуты полей которого подобны атрибутам копируемых полей.

Назначается тип поля — числовой или символьный. Содержимое поля можно вычислить по формуле, аргументами которой служат другие поля формы, с точностью до 14 знаков.

Для уменьшения числа ошибок заполнения полей можно задать шаблон заполнения, а для чисел — указать диапазон величин, степень округления результатов вычисления, формат размещения в поле данных.

При назначении атрибутов всем полям записной книжки, в которые должна помещаться числовая информация, присвоим тип — числовой. Для полей даты рождения укажем диапазоны: первому — число дней 1...31, второму — номер месяца 1...12. Остальные поля будут содержать произвольную символьную информацию, и им можно назначить атрибуты по умолчанию.

Введенная информация проверяется системой на соответствие описаниям сразу после заполнения данными перед занесением их в файл или после заполнения промежуточного файла данных без проверки, но с последующей проверкой всей введенной информации.

При необходимости по каждому из вопросов системы пользователь может получить подробные пояснения, введя команду "И. В процессе назначения атрибутов по команде "Е можно вернуться назад к предыдущему вопросу или пропустить вопрос, введя <ВК>, приняв тем самым значение атрибута, указанное по умолчанию.

Команду "Р можно не вводить или же на все вопросы системы отвечать нажатием клавиши <ВК>, в результате в обоих случаях полю будут назначены атрибуты по умолчанию.

После создания и определения полей форма сохраняется командой "Ц. Если при создании формы имелись ошибки, то будут выданы сооб-

щения об их причинах и приглашение вернуться назад и внести исправления.

В режим работы с данными можно перейти из программы «Форма» или же вызвать программу «Архив» из системы. По каждому из режимов работы с данными справочная информация выдается по команде "И.

Если форма используется впервые и еще ни одна запись не занесена в систему, то доступен только режим ввода данных. В нем действуют команды перемещения курсора по полям формы, редактирования содержимого поля, распечатки на принтере содержимого форм и завершения ввода данных.

При пополнении базы данных можно не заполнять каждый раз заново поля, которые не изменяются от записи к записи. Для перезаписи достаточно выполнить команду "Ц. Это значительно ускоряет ввод данных и уменьшает число ошибок.

Для работы с базой данных нужно перейти в меню выбора режима, набрав команду "Е. В указанном меню перечислены все возможные виды работы с данными. Записи можно просматривать в порядке ввода (режим Д) или в порядке индексации (режим И). В последнем случае записи будут выдаваться в отсортированном виде.

В обоих случаях на экране имеется меню режима просмотра с указанными операциями над полями, доступными в текущем режиме. В процессе просмотра можно вызывать на экран содержимое следующей (команда "Н) или предыдущей (команда "П) записи.

При просмотре данных можно корректировать отдельные поля записей. Измененные записи заносятся в базу, а старые отмечаются как уничтоженные, и система их не воспринимает. По мере накопления данных возникает необходимость в очистке системы от записей, помеченных как уничтоженные, и в сортировке данных в базе в соответствии с порядком значений ключевых полей. Эти операции выполняются в режиме обслуживания, который вызывается командой "Ф.

Если необходимо быстро найти определенную запись, используется режим поиска по ключу (команда "К), указывается значение ключевого поля, и по индексному файлу в режиме непосредственного доступа система извлекает нужную запись. Если запись с указанным ключом не будет обнаружена, выдается соответствующее сообщение. В примере записной книжки ключом является фамилия.

Маска просмотра определяет записи, которые будут выводиться на экран при использовании одного из двух режимов просмотра. До редактирования маски просмотра на экран будут выводиться все записи в порядке, установленном соответствующим режимом про-

смотря. Если маска просмотра изменена или отредактирована, то выводятся только записи, соответствующие маске просмотра, а все остальные — пропускаются. Установив маску просмотра для полей даты рождения (например, 03,12), получим запись с нужной информацией. Накопленную в базе данных информацию можно распечатать целиком или выборочно.

Система «Архив» совместима с популярной прикладной системой управления базами данных DataStar производства фирмы MicroPro, распространенной на восьмиразрядных персональных ЭВМ с операционными системами класса CP/M.

Телефон 252-62-28, Москва

ЛИТЕРАТУРА

1. Система подготовки документации ТЕКСТ: Методические материалы и документация по пакетам прикладных программ.— М.: МЦМТИ.— 1987.— Вып. 42.
2. Data Star Users guide. MicroPro International Corp., 1980.
3. Борковский А. Б. Англо-русский словарь по программированию и информатике (с толкованиями).— М.: Рус. яз., 1987.
4. Уэйт М., Ангермейер Дж. Операционная система CP/M: Пер. с англ.— М.: Радио и связь, 1987.

Статья поступила 21.04.89

УДК 681.327

Н. В. Стрелков, А. В. Кокарев

УСТРОЙСТВО ЗАЩИТЫ ПАМЯТИ МИКРОЭВМ

Устройства защиты памяти микроЭВМ получили широкое распространение в связи с развитием микро мощных ОЗУ статического типа, применяемых в качестве электронных дисков ПЭВМ и буферных ОЗУ в контроллерах станков с ЧПУ, концентраторах информации, кассовых и банковских терминалах и т. д. Они позволяют разрешить противоречие между потребляемой мощностью микроЭВМ и надежностью защиты ее оперативной информации в условиях частого нарушения питания и при длительном отключении сети.

В настоящее время используются следующие способы защиты памяти: автономное (чаще аккумуляторное) питание микроЭВМ; аппаратное управление защитой информации; аппаратно-программный способ защиты информации; построение ОЗУ данных микроЭВМ на основе перепрограммируемых энергонезависимых кристаллов памяти [1].

Наиболее распространенный способ аппаратно-программной реализации защиты памяти [2, 3] состоит в том, чтобы в момент пропадания сети перевести микроЭВМ в режим выполнения подпрограммы прерывания с наивысшим при-

оритетом, которая предусматривает перезапись в стек ОЗУ данных содержимого СОЗУ центрального процессора, а затем подключить ОЗУ и схему управления защитой памяти к батарейному (или аккумуляторному) источнику питания, отключить сигнал выборки ОЗУ.

Существенный недостаток этого способа — нежелательная активизация процессора и памяти данных в условиях низкой стабильности питающего напряжения. Кроме того, в режиме прерывания процессора предусмотрен его возврат в прерванную точку программы, что нарушает работу программируемых контроллеров микроЭВМ и связанного с ними периферийного оборудования. В момент подачи сетевого напряжения микроЭВМ, не получив подтверждения внешних устройств, как бы «зависает» в прерванной точке программы, что требует дополнительного вмешательства оператора и фиксируется как сбой периферийного оборудования.

Предлагается комбинированный способ защиты памяти, позволяющий преодолеть вышеуказанные недостатки и повысить надежность защиты информации формированием пассивных режимов работы ОЗУ и процессора от быстродействующего энергонезависимого устройства управления в моменты включения-отключения питания микроЭВМ и программной поддержкой процессора при выполнении подпрограммы СБРОС.

Схема защиты памяти микроЭВМ может быть построена на цифровых ИС с микро мощным энергопотреблением (рис. 1). При первом включении питания процессор, обращаясь к ОЗУ, помещает в память данные или прикладную программу, записанные полностью либо частично. Порог нуля питающего напряжения устанавливается переменным резистором R от внешнего источника 4,75 В (перемычка S1 разомкнута). В рабочем состоянии перемычка S1 замыкается на внутренний источник 5 В. Процесс отключения электропитания микроЭВМ может длиться от единиц до десятков миллисекунд, что в основном определяется схемотехническими особенностями блока питания.

При снижении питающего напряжения на входе компаратора D1.1 ниже уровня установленного нуля (4,75 В) на его выходе формируется сигнал Лог. 1, а на выходе D1.4 — сигнал захвата (Зх), который передается на вход процессора. Процессор завершает выполнение текущей команды и переходит в режим «захват»: с выхода HLDA процессора на вход D2.2 выдается сигнал подтверждения захвата (ПЗх). Сигналом ПЗх (Лог. 1) элемент D2.1 защелкивается в состояние Лог. 1, сигнал сброса (СБ) при отключении питания не формируется (Лог. 0), с выхода D2.4 выдается управляющий сигнал отключения памяти (ОП, Лог. 1).

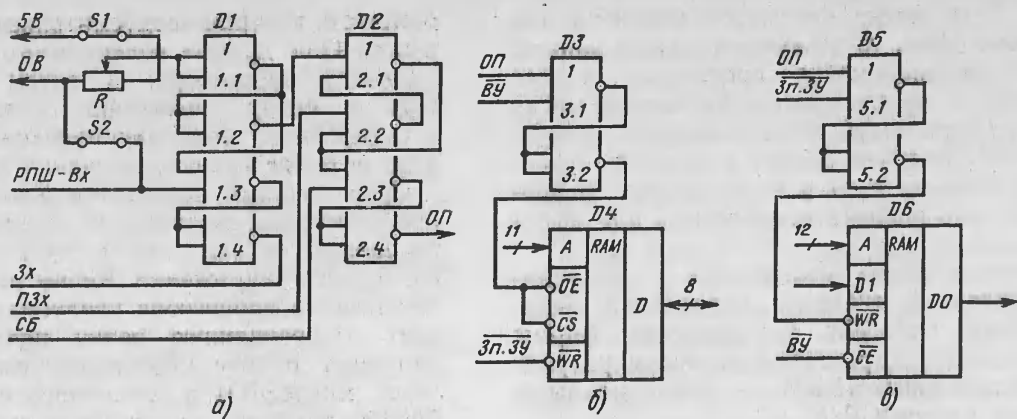


Рис. 1. Схема устройства защиты памяти микроЭВМ:

а — формирование сигнала отключения памяти; б — управление сигналом выборки устройства (ВУ); в — управление сигналом записи запоминающего устройства (Зп.ЗУ) D1, D3, D5 — К561ЛЕ5; D4 — КР537РУ10А (Б); D6 — КР537РУ2А (РУ3А); R — резистор СП5-3-470 Ом; S1, S2 — переключатель; ОВ — общий вывод

Сигнал ОП запрещает выборку ОЗУ (рис. 1, б), выходы D4:D переводятся в высокоимпедансное состояние ($OE = CS = \text{Лог. 1}$). Далее сигнал ОП переводит ОЗУ в режим чтения данных ($WR = \text{Лог. 1}$) (рис. 1, в). Таким образом обеспечивается защита памяти микроЭВМ в момент отключения питания и в режиме хранения информации.

При включении сети, как только уровень питающего напряжения на входе компаратора D1.1 превысит уровень установленной единицы (4,75 В), элемент D2.1 сигналом Лог. 1 переводится в состояние Лог. 0 (первый вход D2.3), а на втором входе D2.3 устанавливается Лог. 1 на время действия аппаратного сигнала СБ. По завершении $СБ = \text{Лог. 0}$ с выхода D2.4 выдается сигнал Лог. 0, который разрешает прохождение сигналов выборки устройства ВУ (рис. 1, б) и записи запоминающего устройства Зп.ЗУ (рис. 1, в) к ОЗУ данных. Таким образом обеспечивается защита памяти микроЭВМ в момент включения питания и в период ее нормальной работы.

Вход разрешения подключения шины (РПШ-Вх) используется в диагностическом режиме работы микроЭВМ. В рабочем режиме переключатель S2 замкнут на общий вывод (ОВ) питания — вход D1.3 в состоянии Лог. 0, поэтому сигнал захвата формируется только компаратором D1.1. В диагностическом режиме работы сигнал захвата формируется при поступлении РПШ-Вх потенциальным уровнем Лог. 1 (переключатель S2 разомкнут).

Программная поддержка устройства защиты памяти обеспечивает автоматическую обработку сбойной ситуации с возвратом микропроцессорной системы в начало прерванной программы. При этом содержимое базовых ячеек ОЗУ отражает ход выполнения основных операций микроЭВМ. По характеру инфор-

мации этих ячеек (признакам) определяется точка возврата.

При любом нарушении питания процессор повторно прерванную операцию автоматически, что обеспечивает надежную работу связанного с микроЭВМ периферийного оборудования, например печатающего устройства или каналов электросвязи.

Для инициализации микропроцессорной системы в целом используется открытый вход (порт) z, состояние которого анализируется при каждом включении питания после прохождения аппаратного сигнала СБ. Если $z = 1$, то процессор переходит в режим выполнения подпрограммы СБРОС с начальной установкой базовых ячеек ОЗУ (программа полного сброса).

Таким образом удается отличить факт случайного или преднамеренного отключения питающего напряжения от факта полной инициализации микроЭВМ. Фрагмент алгоритма программы СБРОС при инициализации каналов электросвязи (выполняемой процессором типа КР580ВМ80А) показан на (рис. 2). Программа запускается аппаратным сигналом СБРОС с адреса 0000H и включает следующие операции: установку указателя стека в исходное состояние;

анализ состояния порта z: если $z = 1$, то управление передается программе полного сброса, по которой все признаки рабочих ячеек ОЗУ и режимов переустанавливаются в исходное состояние; если $z = 0$, то анализируются базовые ячейки ОЗУ, устанавливаются программируемые контроллеры и переустанавливаются отдельные признаки рабочих ячеек ОЗУ; анализ состояния приема блока информации: если блок принят полностью, то в канал связи отсылается управляющий символ НЕТ и ожидается прием нового блока;

анализ состояния на ожидание, подтвержде-

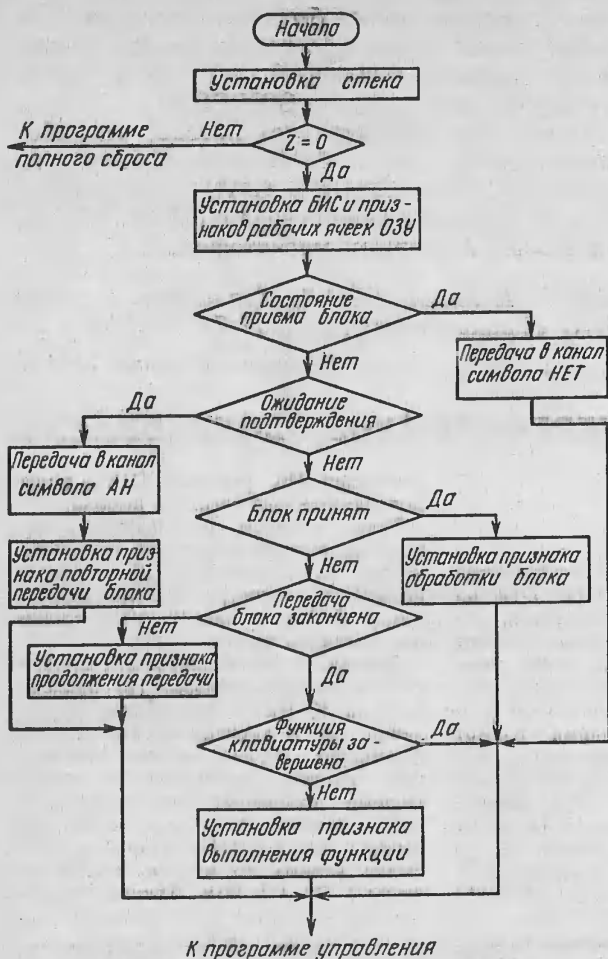


Рис. 2. Фрагмент алгоритма программы СБРОС при инициализации каналов электросвязи

ния приема блока информации: если блок передан полностью и ожидается прием подтверждения, то в канал связи передается управляющий символ АН (аннулирование предыдущей посылки) и устанавливается признак повторной передачи блока;

повторный анализ состояния приема блока информации: если блок принят полностью, то устанавливаются признаки для возврата процессора к началу обработки этого блока;

анализ состояния передачи блока информации: если передача блока не закончена, то все признаки, необходимые для передачи, остаются без изменений, а блок будет передаваться начиная с символа, который еще не был передан в канал связи; передача блока повторится заново, если он будет принят с ошибкой;

анализ выполнения функций клавиатуры: если была прервана одна из функций клавиатуры, то происходит возврат к началу выполнения данной функции;

переход к программе управления микроЭВМ: анализируются соответствующие признаки ра-

бочих ячеек ОЗУ и выполняются необходимые процедуры.

Схема управления и ОЗУ запитываются от двух источников: основного 5 В и резервного, развязанных через полупроводниковый диод. В качестве резервного источника используется блок из четырех аккумуляторов типа Д-0, 06Д (Д-0, 025Д), включенных последовательно. В режиме хранения схема, состоящая из трех ИС К561ЛЕ5 и двух КР537РУ10, потребляет не более 50 мкА.

Экспериментальная проверка устройства защиты памяти микроЭВМ в автономном режиме и в составе терминала производственного СИ6900 показала достаточную надежность защиты информации ОЗУ в процессе жесткой имитации включения-отключения питающего напряжения. Достигнуто быстродействие защиты по включению питания не менее 1 мкс, по выключению — 3 мкс. Получено положительное решение ВНИИГПЭ по заявке на изобретение № 4288034/24 с приоритетом от 21.07.—87

Телефон 3-21-37, Волгоградская обл., г. Волжский, Кокарев Александр Васильевич

ЛИТЕРАТУРА

1. Роберт Мак-Айвор. Электронные кредитные карточки // В мире науки.— 1986.— Вып. 1.— С. 74.
2. Заявка Японии № 61-30304. G06F12/16 1/00.— Изобретения стран мира.— 1987. Вып. 117.— № 7.— С. 147.
3. Моргенштерн Т. Квазинезависимый модуль КМОП ОЗУ и ППЗУ для микроЭВМ K1520 // Radio Fernsehen Elektronik / Пер. с нем.— 1984.— Т. 33.— Вып. 12.— С. 766—768.

Статья поступила 15.11.88

УДК 621.373

С. Е. Макаров

АРМ НА БАЗЕ КТС ЛИУС-2

Для автоматизации процесса контроля динамических характеристик промышленных изделий типа электроклапан (ЭК) с регистрацией результатов испытаний печатающим устройством в виде протокола испытаний и графиков процесса было разработано автоматизированное рабочее место (АРМ) на базе блоков КТС ЛИУС-2.

Устройство выполнено в виде автономного управляющего модуля с функциями микроЭВМ. В качестве дисплея используется видеотерминал ВТА 2000-30 с клавиатурой КТ-М3, печатающее устройство типа А521 (МП-4).

Программное обеспечение разработано с помощью кроссассемблера ОРИОН в среде ДОС АСПО для СМ2М (СМ1634) на языке макроассемблера микропроцессора КР580ВМ80.

Промышленная эксплуатация АРМ показала, что подобные комплексы могут эффективно применяться в различных отраслях. В на-

стоящее время ведутся работы по созданию на базе существующего АРМ комплекса контроля обжиговой печи, АРМ испытателя магнитных свойств трансформаторной стали. Широкий выбор блоков КТС ЛИУС-2 и разработанные универсальные программные блоки (драйверы ВТА и устройства печати, библиотеки подпрограмм, микротесты) позволяют достаточно просто изменять структуру и профиль использования описываемого программно-технического комплекса. Этому способствует и заложенная избыточность технических решений, так, например, число каналов обрабатываемых анало-

говых сигналов может быть увеличено до 32 и более, число линий программного телеуправления (включен-выключен) — до 16 и более, суммарный объем памяти ОЗУ-РПЗУ — до 64К.

Кроме того, представляет интерес разработанная линия связи АРМ с ЭВМ верхнего уровня (СМ2М, СМ1634, СМ4). Ведутся работы по созданию линий связи с ПЭВМ, НГМД и бытовым кассетным магнитофоном.

622607, Н-Тагил, ГСП-7, Уральская, 2, НТО НПО «Уралсистем»; тел. 2-17-74, 2-35-88

Сообщение поступило 21.11.88

ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ

УДК 681.327.8.06

Е. М. Алексеева, Э. С. Кривелева, И. В. Флейшман

МОДУЛЬ ПАРАЛЛЕЛЬНОГО ВВОДА-ВЫВОДА

Модуль параллельного ввода-вывода (МПВВ) предназначен для обмена информацией в виде 16-разрядного двоичного кода между ВУ и микроЭВМ «Электроника МС 1201.02». МПВВ состоит из устройства управления вводом-выводом информации и передачей информации*. Функцию устройства управления выполняет БИС К1801ВП1-033 (см. рисунок). Данные передаются с помощью четырех БИС К1801ВП1-034. Стандартная схема включения БИС К1801ВП1-034 в режиме передачи информации позволяет ввести 20 и вывести 16 двоичных сигналов. В описываемом модуле число сигналов ввода-вывода увеличено до 32 благодаря нескольким отличающемуся от стандартного включению БИС и применению дополнительных регистров, организованных на БИС К1801ВП1-034 и включенных в режиме буферов.

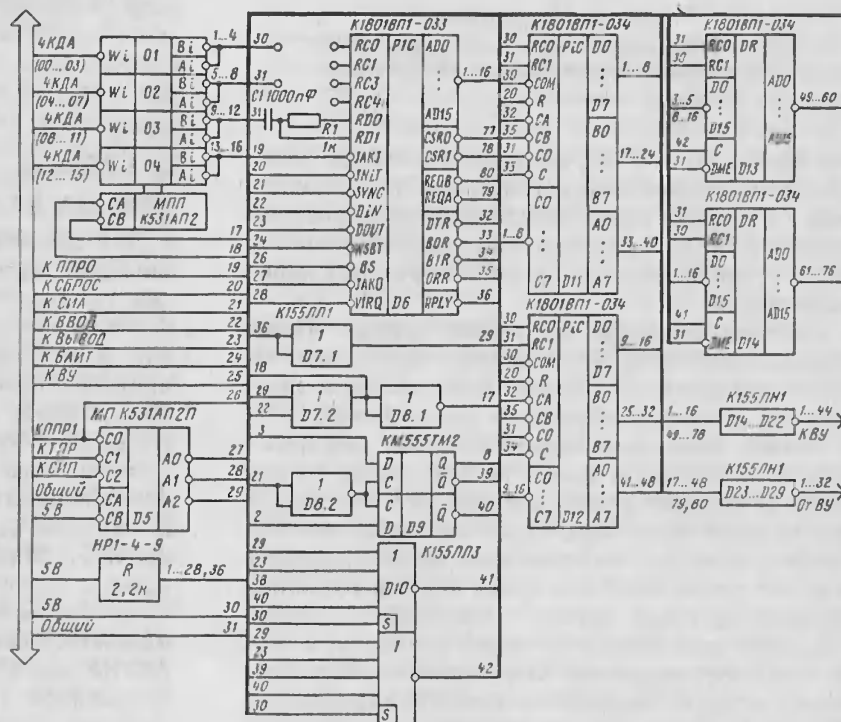
Для микроЭВМ модуль представлен тремя отдельными регистрами, программно доступными для записи и чтения. По каждому из трех адресов вводится или выводится, т. е. каждый адресуемый регистр осуществляет функции двух регистров ввода и вывода информации. Адреса регистров состояния, ввода и вывода задаются с помощью переключателей на Д6. При обращении микроЭВМ к МПВВ адрес из канала через приемопередатчик поступает на двуправленные входы-выходы Д6 и по приходу сигнала К СИА дешифрируется. Через определенное время микроЭВМ помещает данные в канал и вырабатывает сигнал К ВЫВОД, поступающий на вход Д6. По выбранному адресу и сигналу К ВЫВОД Д6 формирует сигналы «Вывод ст. байта» и «Вывод м. байта», стробирующие

данные из каналов С0...С7 устройства передачи информации Д11, Д12 на его входы Д0...Д7. Подтверждая, что данные приняты, Д6 вырабатывает сигнал К СИП, который через передатчик Д5 поступает в канал. Код адреса дополнительных регистров формируется двумя разрядами КДА01 и КДА02, дешифрируется адрес Д10. По совпадению кода адреса, сигналов К ВЫВОД и К СИП записывается 12-разрядный код (разряды КДА02...04, 07...15) в регистр Д11 и 16-разрядный код в регистр Д12. Регистру Д11 присвоен адрес регистра

состояния Д6, регистру Д12 — адрес дополнительного регистра вывода.

Таким образом, использование незанятых разрядов регистра состояния устройства управления Д6 и дополнительного регистра вывода информации Д12 увеличивает число двоичных сигналов до 44.

Данные из ВУ в микроЭВМ вводятся следующим образом. По приходу сигнала К ВВОД устройство управления Д6 в зависимости от адреса формирует сигнал «Ввод данных» или «Чтение выходного регистра». Первый разрешает мультиплексирование данных со входов А0...А7 на выходы С0...С7 Д11, второй — пересылку данных со входов В0...В7 на выходы С0...С7 Д12. Таким образом



Принципиальная схема модуля параллельного ввода-вывода

* Алексеева Е. М., Кривелева Э. С., Флейшман И. В. Микропроцессорное устройство для газоналитических систем контроля атмосферного воздуха // Тез. докл. Всесоюз. науч.-техн. конф. Современные методы и средства автоматического контроля атмосферного воздуха и перспективы их развития. — Киев, 1987

через передатчики D1...D4 информационный 16-разрядный двоичный код поступает в системный канал.

Длительность сигнала «Ввод данных» можно при необходимости изменить с помощью RC-цепи между выводами RC0, RC1 D6. В регистре состояния D6 предусмотрены разряды разрешения прерывания (REQA, REQB), которые устанавливаются ВУ и применяются в качестве флагов. При программном опросе регистра состояния D6 микроЭВМ по флагу вводит данные от ВУ. Сигналы CSR0, CSR1 — разряды 0, 1 регистра состояния D6 — могут использоваться для имитации запросов прерывания ВУ в режиме автономной проверки. Буферные

элементы для связи с ВУ выполнены на ИМС К155ЛН1. Если возникает необходимость в увеличении вдвое числа сигналов ввода-вывода, то следует подключить одну ИМС К1801ВП1-033, четыре ИМС К1801ВП1-034, 14 ИМС К155ЛЕ1.

Конструктивно такой модуль с дополнительным интерфейсом параллельного ввода-вывода можно выполнить на полуплате микроЭВМ «Электроника МС 1201.02».

252650, Киев, б. ул. Тверская, 6, КНПО «Аналитприбор»; тел. 269-99-44

Статья поступила 23.05.88

УДК 681.327.8.06

Е. М. Алексеева, Э. С. Кривелева, И. В. Флейшман

МОДУЛЬ ВВОДА-ВЫВОДА АНАЛОГОВЫХ СИГНАЛОВ

Модуль предназначен для преобразования информационных аналоговых сигналов в двоичный код, ввода его в микроЭВМ и вывода двоичных сигналов с последующим преобразованием в аналоговые сигналы для измерения и регистрации.

В качестве микроЭВМ может быть использована «Электроника МС 1201.02». Связь модуля с микроЭВМ осуществляется по асинхронному принципу 16-разрядными словами. Обмен информацией с каналом микроЭВМ управляет БИС К1801ВП1-033 (см. рисунок). Кроме интерфейсной части модуль содержит устройство аналогового ввода (УАВВ), устройство аналогового вывода (УАВД), источник опорных напряжений (ИОН) и таймер. Для микроЭВМ модуль представлен адресуемыми регистрами УАВВ, УАВД и регистром состояния. Адреса регистров устанавливаются с помощью перемычек.

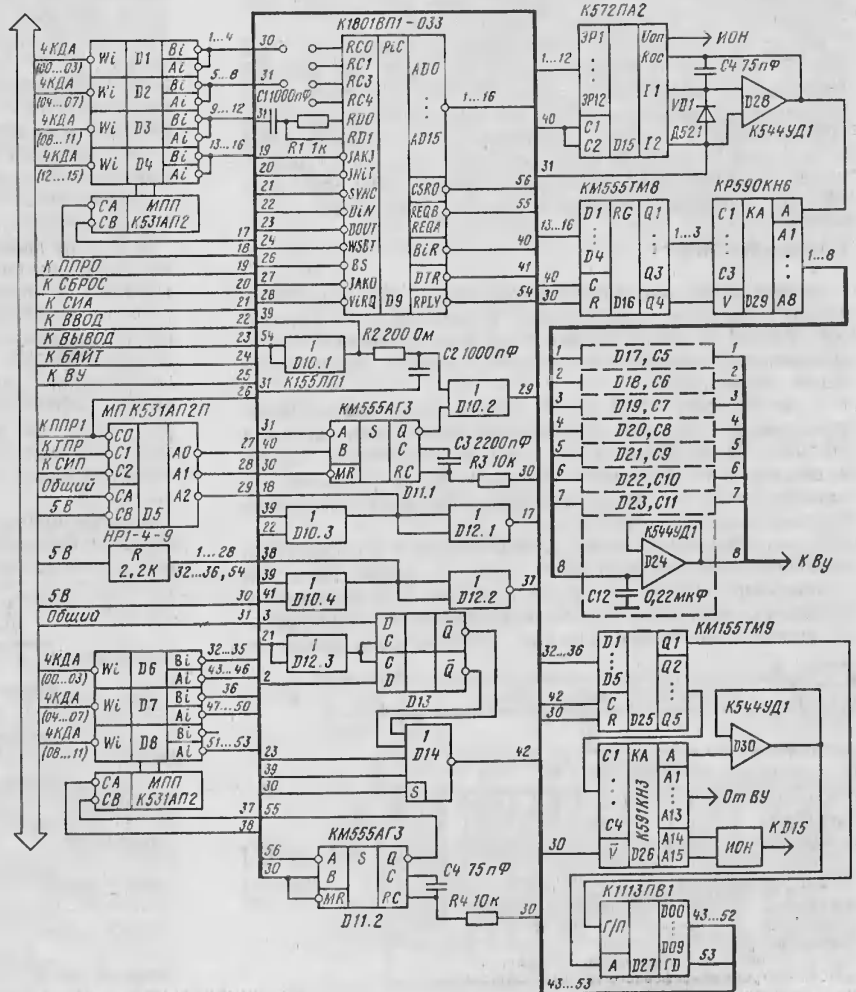
УАВВ предназначено для преобразования 16 аналоговых сигналов постоянного напряжения в 10-разрядный двоичный код. Содержит регистр адреса каналов, коммутатор, усилитель и АЦП. При обращении по адресу УАВВ устройство управления формирует сигнал записи, по которому регистр D25 запоминает код адреса номера канала. В качестве аналогового коммутатора использована ИМС К591КНЗ, которая пропускает один из 16 внешних аналоговых сигналов на вход усилителя D30, включенного в режиме повторителя для согласования выходов коммутатора со входом АЦП. АЦП выполнен на БИС функционально законченного 10-разрядного АЦП последовательного приближения К1113ПВ1А. Сигнал пуска АЦП формируется программно и поступает через D25 на вход АЦП. В конце преобразования на выходе АЦП появляется сигнал «Готовность данных». При программном опросе микроЭВМ по биту готовности вводит выходной код АЦП в канал 12-разрядный двоичный код пре-

роЭВМ по биту готовности вводит выходной код АЦП в канал.

12-разрядный двоичный код пре-

образуется в выходные аналоговые сигналы постоянного напряжения с помощью УАВД, включающего ЦАП, регистр адреса каналов, схему аналоговой памяти (САП). Основной элемент УАВД — 12-разрядный умножающий ЦАП с функцией записи и хранения цифровой информации, выполненный на БИС К572ПА2А. При обращении по адресу УАВД формирует сигнал записи информационного кода в регистр памяти ЦАП. Для достоверной записи данных (время записи составляет 15 мкс) введена задержка синхронпульса К СИП, выполненная на ИМС КМ555АГЗ. Аналоговый сигнал ЦАП с помощью коммутатора К590КНБ и САП преобразуется в 8-канальный.

САП выполнена на операционных усилителях К544УД1Б, включенных в режиме повторителей, и емкостях К73-17. Схема предназначена для запоминания мгновенного значения выходного сигнала ЦАП и хранения его в течение определенного времени. Наличие САП требует периодической регенерации, т. е. вывода данных в ЦАП и подключения его выходного сигнала



Принципиальная схема модуля ввода-вывода аналоговых сигналов

ко входам САП. Регенерация организована по прерыванию от сигналов внутреннего таймера, выполненного на одновибраторе КМ555АГЗ. Запускается таймер программно.

Основные инструментальные погрешности УАВВ и УАВД следующие: смещение нуля, изменение коэффициента передачи и нелинейность АЦП и ЦАП. Смещение нуля и коэффициента передачи УАВВ может корректироваться в программном режиме. ИОН формирует эталонные напряжения для коррекции смещения нуля

и коэффициента передачи АЦП и опорное напряжение ЦАП. Погрешность преобразования определяется стабильностью ИОН и нелинейностью АЦП и ЦАП*.

Время аналого-цифрового преобразования составляет не более 30 мкс, цифро-аналогового — не более 20 мс.

* Бородавка В. П., Кривелева Э. С., Флейшман И. В. Устройство ввода-вывода аналоговых сигналов // Приборы газодиагностического контроля атмосферы и промышленных выбросов: Сб. науч. тр. — Киев: ВНИИП, 1985.

Число внешних аналоговых сигналов ввода постоянного напряжения в диапазоне 0...10 В — 14, вывода — 8. Погрешность УАВВ — не более 0,2 %, УАВД — не более 0,05 %.

Конструктивно модуль выполнен на полуплате микроЭВМ «Электроника МС 1201.02».

252650, Киев, 6, ул. Тверская, 6, КНПО «Аналитприбор»; тел. 269-99-44

Статья поступила 23.05.88

УДК 681.325

Н. Н. Щелкунов, А. П. Дианов

ОРГАНИЗАЦИЯ КОНТРОЛЛЕРОВ ГМД

Прикладная ценность и эффективность многих микросистем (МС) определяется наличием в их составе встроенных средств связи с устройствами внешней памяти, которые можно использовать для длительного хранения больших массивов информации. Широко распространены относительно недорогие и компактные приводы гибких магнитных дисков (ГМД). Сфера профессиональных приложений бездисковых одноплатных микроЭВМ (например, семейство мМС 12XX [1]) может быть значительно расширена вводом в их состав таких средств.

Функцию физического сопряжения с приводами ГМД обычно выполняют контроллеры ГМД (рис. 1), которые освобождают центральный процессор (ЦП) от трудоемких обязанностей низкоуровневого управления приводом. Роль ЦП сводится к общему управлению контроллером ГМД с помощью команд и слов состояния высокого уровня. Программирование процесса ввода-вывода (ВВ) упрощается, общая производительность системы повышается.

Интерфейс ГМД

Задача проектирования контроллеров ГМД значительно упростилась с унификацией физического интерфейса привода [2]. В его состав входят от 8 до 22 линий с низким уровнем активности сигналов (обозначается звездочкой в конце имени).

К контроллеру ГМД могут быть подключены до четырех приводов, которые выбираются радиальным сигналом DRIVE SELECT I (I=0...3), разрешающим прием и выдачу всех остальных сигналов магистрального типа. При этом на передней панели привода загорается световой индикатор.

Перед обращением к выбранному приводу его двигатель включается по сигналу RUN, который может быть как радиального, так и магистрального типа. Снятие сигнала RUN останавливает двигатель. Для повышения времени жизни в некоторых приводах предусмотрена автомати-



Рис. 1. Связь привода ГМД с микроЭВМ

ческая остановка двигателя при отсутствии около 1 с обращения к устройству. Сигналы RUN и DRIVE SELECT могут быть совмещены.

О готовности к работе привод сообщает контроллеру сигналом READY (рис. 2). Автоматическая остановка двигателя или открытие дверцы привода снимает этот сигнал (в некоторых устройствах может отсутствовать). В этом случае для набора номинальной скорости вращения двигателя (после подачи сигнала RUN) следует выдержать паузу до 1 с.

Операция позиционирования состоит в установке головки привода на требуемую дорожку. Для этого головка должна выполнить ряд последовательных переходов с дорожки на дорожку. Движение головки управляется через шаговый двигатель двумя сигналами: DIRECTION отвечает за направление (0 — к краю, 1 — к центру диска), STEP — за импульс перехода с дорожки на дорожку. Операция перехода осуществляется по окончании импульса STEP. Обычно контроллер следит за текущим номером дорожки, над которым установлена головка, поэтому операция позиционирования заключается в выборе соответствующего направления движения и подаче необходимого числа импульсов STEP (рис. 2). Период их следования называется временем шага SRT.

В момент включения питания положение головки привода неопределенно. В этом, а также ряде других случаев выполняется операция рекалибровки (головка устанавливается на нулевую дорожку). Контроллеры ГМД контролируют выполнение операции, принимая специальный сигнал TRACK0. Операция рекалибровки состоит в выборе направления движения головки к краю диска, 00-дорожке (DIRECTION=0) и формировании серии импульсов STEP до появления сигнала TRACK0. Динамика формирования сигналов DIRECTION и STEP не связана с временными диаграммами включения и разгона основного двигателя.

Многие приводы имеют две головки считывания-записи, рассчитанные на двухсторонние диски. При обращении к такому приводу формируется сигнал TWO SIDED,

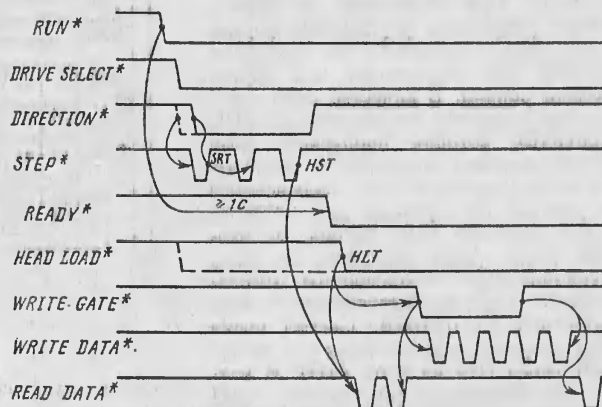


Рис. 2. Временные диаграммы интерфейса ГМД

сообщающий контроллеру о двух головках и рабочих поверхностях.

Головка выбирается сигналом HEAD SELECT и по сигналу HEAD LOAD прижимается к поверхности диска. Операцию чтения-записи данных можно выполнять через 30..50 мкс (время прижатия головки HLT). Сигналы HEAD LOAD и DRIVE SELECT одного привода могут совмещаться. Обычно операция позиционирования выполняется при прижатых головках, что значительно уменьшает время задержки начала операции доступа к данным (время успокоения головки HST).

Очень часто требуется защита дисков от записи на них какой-либо новой информации (например, эталонных и системных) от несанкционированного разрушения пользователем). Такие диски метятся наклейкой фольги в определенном месте конверта. Большинство приводов ГМД различают установку в него защищенного диска, сообщая об этом контроллеру генерацией сигнала WRITE PROTECT. Попытка записи на защищенный диск приводит к ошибке выполнения операции.

Сигнал WRITE GATE определяет тип доступа к данным (запись или чтение). При считывании (WRITE GATE=0) данные выводятся из привода по линии READ DATA вместе с сопровождающими их синхросигналами, при записи (WRITE GATE=1) поступают в привод ГМД по линии WRITE DATA. Каждый переход сигнала меняет направление тока через головку (на магнитную поверхность диска записывается один бит информации).

Ряд приводов предусматривает возможность уменьшения тока через магнитную головку во время записи данных. Величина тока записи понижается сигналом LOW CURRENT. Запись при уменьшенном токе рекомендуется проводить на внутренних дорожках диска ближе к центру.

Для указания физического начала дорожек гибкие диски имеют индексное отверстие, которое при вращении диска (около 5 об./с) попадает между светодиодом и фототранзистором и вызывает формирование импульсов, выводимых на линию INDEX. Индексные импульсы необходимы при операции форматирования диска (при его разметке контроллером на дорожки и секторы). Линия INDEX всегда возбуждена при отсутствии диска (воспринимается контроллером как неисправность привода).

Если во время операции доступа к данным в приводе возникает ошибка, генерируется специальный сигнал FAULT. В свою очередь контроллер в любой момент времени может прервать работу привода формированием управляющего сигнала FAULT RESET.

Использование однокристалльных контроллеров ГМД (табл. 1) значительно упрощает задачу проектирования встроенных в МС средств связи с приводами ГМД. Они заменяют большие объемы аппаратных средств, повышают компактность и надежность МС.

Системный интерфейс БИС 18272А (рис. 3) удовлетворяет требованиям стандарта P696-3 IEEE на 8-разрядную унифицированную микросхему с расширенным стробом записи WR [5]. В состав интерфейса входят три линии для организации обмена через канал прямого доступа к памяти

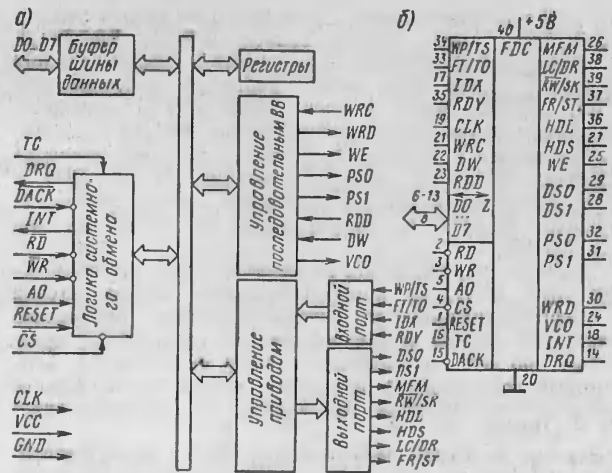


Рис. 3. Структурная схема контроллера ГМД 18272 (а) и его условное графическое обозначение (б)

(ПДП): DRQ — запрос на обмен, DACK — подтверждение обмена, TC — конец обмена и линия запроса на прерывание INT. Работа прибора синхронизируется импульсами CLK частотой 8 МГц при работе с обычными или 4 МГц с мини-дисками.

Периферийный интерфейс контроллера совместим со стандартным стыком привода ГМД. К однокристалльному контроллеру 18272А могут быть подключены либо два привода со свдвоенными, либо четыре привода с одиночными дисками, имеющими одинарную (FM) или двойную (MFM) плотность записи (рис. 4). Привод и диск на нем (сигналы DRIVE SELECT) выбираются с помощью линий DSO, DSI; головка для двухстороннего диска (HEAD SELECT) и ее прижим (HEAD LOAD) — линией HDS и HDL соответственно; направление передачи (WRITE GATE) — сигналом разрешения записи WE; информирование внешних узлов о режиме работы FM/MFM — по линии MFM.

Обычно контроллер работает либо в режиме позиционирования (SEEK), либо чтения-записи (RW), о чем свидетельствует потенциал на линии RW/SK. В зависимости от режима работы контроллера меняется функциональное назначение ряда линий его периферийного интерфейса. В режиме чтения-записи линия LC/DR используется для понижения тока на внутренних дорожках (LOW CURRENT), а в режиме позиционирования — для указания направления движения (DIRECTION). С помощью линии FR/ST формируется сигнал сброса ошибки (FAULT RESET) или генерация строба перехода с дорожки на дорожку (STEP). Такое совмещение функций позволяет сократить число физических выводов кристалла.

К множеству входных линий мультиплексируемых режимом RW/SK относятся WP/TS и FT/TO. По линии

Таблица 1

Однокристалльные контроллеры ГМД

Тип БИС	Функциональное описание
KP1818BG93	Программируемый контроллер ГМД с одинарной и удвоенной плотностями записи [3]
i8272, i8272A, KM1821BG72A	Программируемый контроллер ГМД с одинарной и удвоенной плотностями записи, управляющий четырьмя приводами
i82072	Высокоинтегрированный контроллер ГМД с одинарной и двойной плотностями записи [4]

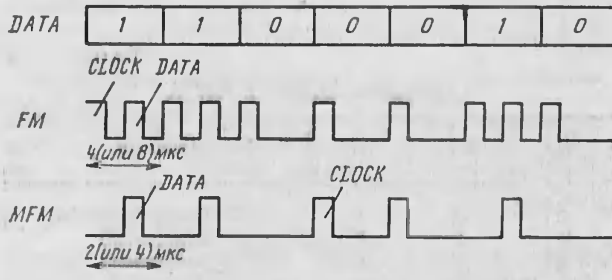


Рис. 4. Представление информации на ГМД: FM — одинарная плотность записи, MFM — двойная плотность для обычных и мини-дисков

WP/TS в режиме чтения-записи проходит сигнал защиты выбранного диска против записи WRITE PROTECT, в режиме позиционирования — сигнал о типе выбранного диска с одной или двумя рабочими поверхностями TWO SIDED. Через вторую линию FT/T0 устройство контролирует состояние ошибки привода FAULT в режиме чтения-записи и достижение приводом нулевого цилиндра TRACK0 в режиме позиционирования. Входы RDY и IDX контроллера i8272A принимают сигнал готовности привода READY и генерируемых им индексных меток INDEX.

Информация, считанная из привода ГМД, вводится через вход RDD, на который подаются данные чтения с сопровождающими их синхроимпульсами. Контроллер разделяет данные и синхроимпульсы внутренними средствами по дополнительному сигналу окна данных DW, который формируется внешними устройствами. Способ формирования сигнала DW зависит от плотности записи FM/MFM.

Данные из контроллера в привод ГМД в смеси с сопровождающими их синхроимпульсами выводятся по линии WRD. Темп их вывода и записи на диск определяется тактовыми импульсами, подаваемыми на вход WRC, частота следования которых зависит от режима (FM или MFM) и типа диска (обычный или мини). При использовании обычных дисков частота следования импульсов WRC равна 500 кГц для FM- и 1 МГц для MFM-режима; мини-дисков — 250 кГц и 500 кГц соответственно (длительность импульсов WRC — 250 нс).

При выводе данных записи в режиме MFM может потребоваться операция предкомпенсации (дополнительная задержка на время T или 2T, где T=150...250 нс в зависимости от сигналов PS0, PS1). Генератор, управляемый напряжением (ГУН), синхронизируется по линии VCO. Его работа разрешается только при VCO=1.

Центральный процессор и контроллер ГМД взаимодействуют через два 8-разрядных регистра (табл. 2). Регистр данных DR доступен для записи и чтения и используется для обмена командами, параметрами, непосредственно данными ВВ и дополнительной информацией о состоянии контроллера. Регистр состояния MSR доступен только для чтения и содержит основную информацию о состоянии контроллера (рис. 5). В отличие от DR доступ к MSR разрешен в любое время.

Контроллер реализует 15 команд (табл. 3), полный цикл выполнения которых можно разбить на три условные фазы: приема, исполнения и ответа. Каждая фаза может потребовать передачи многобайтовых последовательностей через регистр DR. В фазе приема контроллер получает всю информацию (C1, C2, ..., CN), необходимую для выполнения конкретной операции.

В фазе исполнения полученная команда выполняется непосредственно в одном из двух режимов (ПДП, не ПДП). В это время может потребоваться обмен массивами данных записи-считывания.

В фазе ответа контроллер выдает информацию о состоянии: адресные данные (C, H, R, N), определяющие текущее положение головки, и четыре дополнительных слова состояния ST0...ST3 (рис. 6).

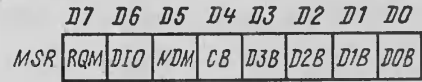


Рис. 5. Формат основного слова состояния i8272A:

RQM — готовность DR к обмену; DIO — направление обмена (0 — прием данных в DR, 1 — выдача из DR); NDM — режим не ПДП; CB — флаг занятости контроллера фазой исполнения операции чтения-записи; DNB — флаг занятости привода N операций позиционирования, где N=0...3; обеспечивает совмещение операций позиционирования и рекалибровки приводов

Фазы приема и ответа выполняются только в режиме программно-управляемого обмена (MSR:RQM=1). Контроллер не переходит к приему новой команды до тех пор, пока вся последовательность ответа не будет передана в ЦП.

Обмен данными во время фазы исполнения может выполняться по каналу ПДП (режим ПДП, MSR:NDM=0). В режиме не ПДП готовность к вводу-выводу очередного байта данных сопровождается генерацией сигнала прерывания INT, который снимается при получении очередного stroba (RD или WR). Период обслуживания ВВ при этом составляет 32 (MFM) или 16 (FM) мкс для обычных и 64 (MFM) или 32 (FM) мкс для мини-дисков. Если MC не успевает реализовать процесс обслуживания прерывания INT за отведенное для этого время, можно воспользоваться программно-управляемыми процедурами условного ВВ (роль флажка готовности выполняет бит RQM регистра состояния MSR). При выполнении контроллером операции чтения-записи (MSR:CB=1), поведение флагов RQM и INT одинаково. В режиме ПДП сигналы DRQ и DACK управляют обменом. В конце фазы исполнения (TC=1) генерируется сигнал прерывания INT (начало фазы ответа). После чтения первого байта сигнал INT сбрасывается.

Во время выполнения команды форматирования дорожки контроллер запрашивает массив данных по четыре байта на каждый физический сектор в следующей последовательности: C (номер сектора), H (номер головки), R (логический номер сектора) и N (размер сектора), что позволяет сформировать дорожку с разбросанными номерами секторов. Каждый логический сектор обычно размещается через некоторое фиксированное число физи-

ST0	IC	SE	ES	NR	HD	US		
ST1	EN	O	DE	OR	ND	NW	MA	
ST2	O	CM	DD	WC	SH	SN	BC	MD
ST3	FT	WP	RDY	TO	TS	HD	US	

Рис. 6. Формат дополнительных слов состояний i8272A:

Слово состояния ST0: IC — код завершения команды; 00 — нормальное, 01 — с ошибкой, 10 — запрещенная команда, 11 — с ошибкой из-за сброса готовности привода; SE — конец позиционирования; ES — ошибка привода поиска нулевой дорожки; NR — привод не готов; HD — адрес головки во время завершения; US — адрес привода во время завершения.

Слово состояния ST1: EN — попытка перехода через границу цилиндра; DE — ошибка CRC; OR — тайм-аут при передаче данных; ND — данные не могут быть найдены; NW — попытка записи на защищенный диск; MA — ошибка поиска адресного маркера.

Слово состояния ST2: CM — чтение сектора удаленных данных при SK=0; DD — ошибка CRC данных; WC — ошибка адреса цилиндра; SH — выполнение условия «равно» во время операции сравнения; SN — не найден сектор, удовлетворяющий необходимым условиям во время операции сравнения на цилиндре; BC — плохой цилиндр (адрес текущего цилиндра равен OFFH); MD — ошибка поиска адресного маркера данных.

Слово состояния ST3: FT — ошибка привода FAULT; WP — защита записи на диск WRITE PROTECT; RDY — готовность привода READY; TO — состояние сигнала нулевой дорожки TRACK0; TS — состояние сигнала двухстороннего диска TWO SIDED; HD — состояние сигнала выбора головки HEAD SELECT; US — состояние линий выбора привода DS0, DS1

Таблица 2

Операции доступа к i8272A

A0	RD	WR	Операция
0	0	1	D ← MSR
0	1	0	Запрещено
X	0	0	Запрещено
1	0	1	D ← DR
1	1	0	DR ← D

Формат команд контроллера ГМД i8272A

Команда	C1 — первый байт команды								C2	C3	Ответ													
	D7	D6	D5	D4	D3	D2	D1	D0			C4	C5	C6	C7	C8	C9	R1	R2	R3	R4	R5	R6	R7	
Чтение данных	MT	MFM	SK	0	0	1	1	0	HDS DS	C	H	R	N	EOT	GPL	DTL	ST0	ST1	ST2	C	H	R	N	
Чтение удаленных данных	MT	MFM	SK	0	1	1	0	0	HDS DS	C	H	R	N	EOT	GPL	DTL	ST0	ST1	ST2	C	H	R	N	
Запись данных	MT	MFM	0	0	0	1	0	1	HDS DS	C	H	R	N	EOT	GPL	DTL	ST0	ST1	ST2	C	H	R	N	
Запись удаленных данных	MT	MFM	0	0	1	0	0	1	HDS DS	C	H	R	N	EOT	GPL	DTL	ST0	ST1	ST2	C	H	R	N	
Чтение дорожки	0	MFM	SK	0	0	0	1	0	HDS DS	C	H	R	N	EOT	GPL	DTL	ST0	ST1	ST2	C	H	R	N	
Чтение состояния контроллера	0	MFM	0	0	1	0	1	0	HDS DS	—	—	—	—	—	—	—	ST0	ST1	ST2	C	H	R	N	
Форматирование	0	MFM	0	0	1	1	0	1	HDS DS	N	SC	GPL	D	—	—	—	ST0	ST1	ST2	C	H	R	N	
Сравнение данных на равенство	MT	MFM	SK	1	0	0	0	1	HDS DS	C	H	R	N	EOT	GPL	GPL	STP	ST0	ST1	ST2	C	H	N	
Сравнение данных на меньше или равно	MT	MFM	SK	1	1	0	0	1	HDS DS	C	H	R	N	EOT	GPL	STP	ST0	ST1	ST2	C	H	R	N	
Сравнение данных на больше или равно	MT	MFM	SK	1	1	1	0	1	HDS DS	C	H	R	N	EOT	GPL	STP	ST0	ST1	ST2	C	H	R	N	
Рекалибровка	0	0	0	0	0	1	1	1	DS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Чтение состояния прерывания	0	0	0	0	1	0	0	0	—	—	—	—	—	—	—	—	ST0	C	—	—	—	—	—	
Спецификация	0	0	0	0	0	0	1	1	SRT HUT	HLT ND	—	—	—	—	—	—	—	—	—	—	—	—	—	
Чтение состояния приводов	0	0	0	0	0	1	0	0	HDS DS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Позиционирование	0	0	0	0	1	1	1	1	HDS DS	C	—	—	—	—	—	—	—	—	—	—	—	—	—	
Запрещенная команда	Все остальные								—	—	—	—	—	—	—	—	ST0	—	—	—	—	—	—	

Примечания:

- C — номер цилиндра;
D — образец байта данных для заполнения сектора при форматировании;
DTL — число байтов данных, считываемых с сектора или записываемых на него при N=0 (при N≠0 DTL=OFFH);
DS — выбор привода и диска на нем DS=0, 1, 2 или 3 (программируется двумя младшими разрядами байта C2);
EOT — число секторов на дорожке;
GPL — длина промежутка GAP3 между секторами без поля синхронизации ГУН;
H — номер головки (0 или 1);
HDS — флажок выбора головки, HDS=H (программируется пятью старшими разрядами байта C2);
HLT — время прижатия головки (программируется семью разрядами байта C3 от 2 до 254 мс с шагом 2 мс при CLK=8 МГц);
HUT — время поднятия головки после окончания операции чтения-записи (программируется четырьмя младшими разрядами байта C2 от 16 до 240 мс с шагом 16 мс при CLK=8 МГц);
MFM — флажок одинарной (FM) или двойной (MFM) плотности записи;
MT — флаг многодорожечной операции, с его помощью в операциях чтения-записи можно пересекать границу дорожки (переход через границу цилиндра не допускается);
N — размер сектора в байтах, который связан с N по формуле $128 \cdot 2^N$, N=0, 1, ...;
ND — управление режимом работы (программируется младшим разрядом байта C3, 0 — режим ПДП, 1 — не ПДП);
R — номер сектора;
ST — число секторов на цилиндре;
SK — флажок пропуска адресной метки удаленных данных;
SRT — скорость шага (программируется четырьмя старшими разрядами байта C2 от 1 до 16 мс с шагом в 1 мс при CLK=8 МГц);
STP — фактор смещения при сравнении данных, указывает, через сколько физических секторов разменен следующий сектор, участвующий в операции сравнения (при STP=1 в операции сравнения физически участвуют соседние секторы).

ческих секторов, которое называется фактором смещения (последовательное размещение логических секторов соответствует фактору смещения, равному единице). Правила передачи данных форматирования и данных записи-считывания аналогичны.

Конец фазы исполнения всегда определяется сигналом ТС. Для режима ПДП этот сигнал принимается контроллером ПДП, для режима не ПДП ТС обрабатывается программным способом. Сигнал ТС обеспечивает работу контроллера ГМД в режиме многосекторного обмена. Важно отметить, что останов обмена между МС и контроллером до окончания текущего сектора не прекращает обмена между контроллером и приводом, который всегда завершается правильно с формированием и проверкой контрольных сумм. При записи сектора на диск его недостающая часть автоматически заполняется нулями.

В операциях декалибровки и позиционирования нет

фазы ответа. Корректность их завершения проверяется командой чтения состояния прерывания. Возвращаемый по этой команде ответ содержит текущий номер цилиндра, над которым в данный момент находится головка считывания-записи.

При обнаружении запрещенного кода команды контроллер из фазы приема переходит в фазу ответа без генерации каких-либо прерываний. С помощью этих кодов контроллер переводится во вспомогательный режим ожидания, в котором никакие команды не воспринимаются.

Контроллер ГМД мМС5301

Устройство связи мМС5301 построено на БИС i8272A (рис. 7) и рассчитано на системные магистрали типа И41. К этому классу относится системная магистраль ИИ1200/1210 [1] семейства одноплатных микроЭВМ

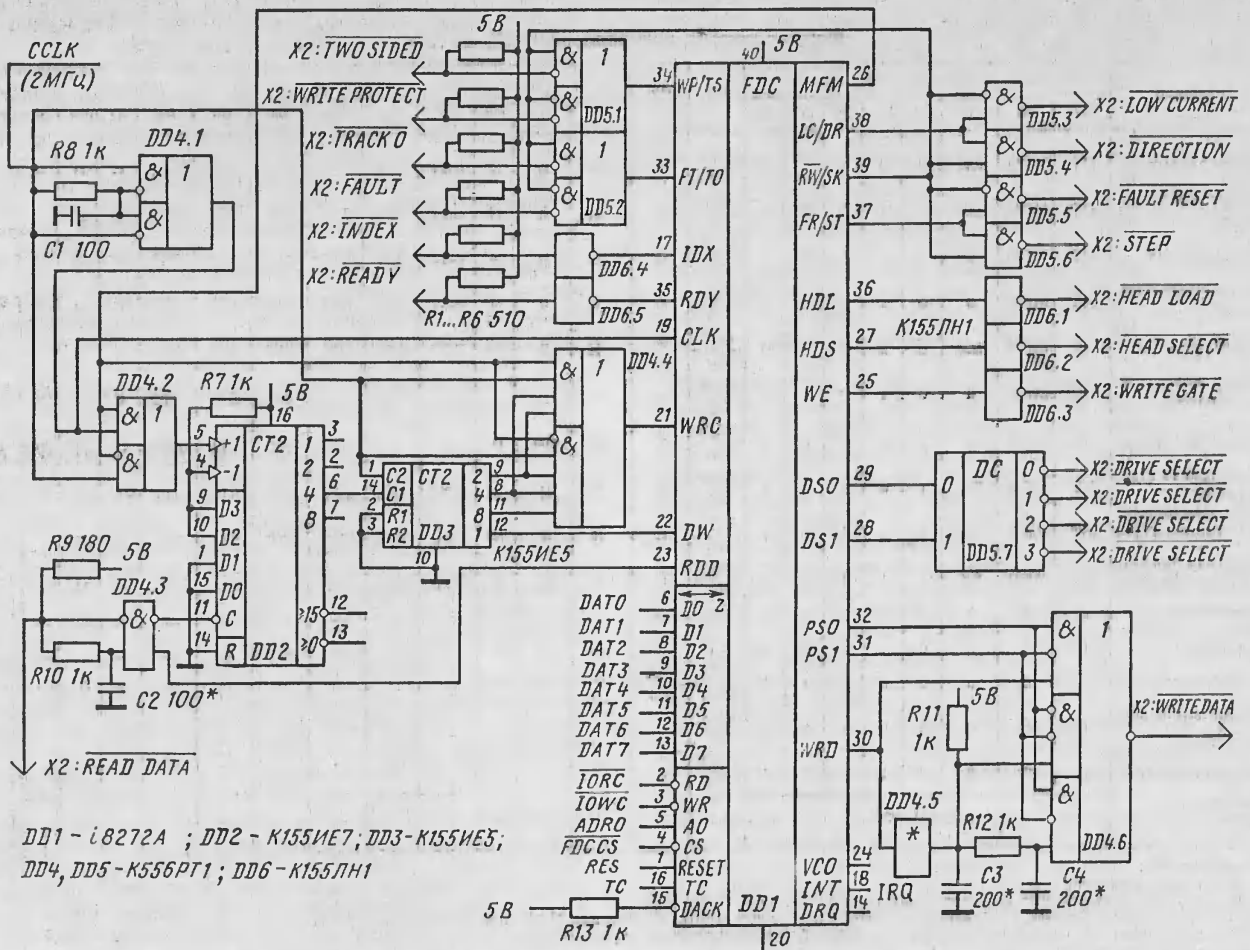


Рис. 7. Схема контроллера ГМД для микроЭВМ семейства мМС

мМС1200/1210. Данная схема расширяет их функциональные возможности.

Контроллер мМС5301 (рис. 7) поддерживает до четырех приводов мини-дисков. Его периферийный интерфейс удовлетворяет спецификациям на стандартный интерфейс привода ГМД. Необходимая для режима MFM операция предкомпенсации выполнена в виде линии задержки на дискретных RC-элементах и цифрового мультиплексора DD4.6.

Для синхронизации контроллера используется системная частота CCLK, типовое значение которой предполагается равным 2 МГц. Если значение CCLK отличается от 2 МГц, то можно использовать встроенный генератор тактовых импульсов. Тактовые импульсы CCLK поступают на удвоитель частоты DD4.1, поэтому основной синхросигнал CLK следует с частотой 4 МГц, что соответствует работе с мини-дисками. Частота записи данных WRC формируется с помощью делителя частоты DD3 и мультиплексора DD4.4. WRC=250 кГц при FM-режиме, 500 кГц при MFM (случай мини-дисков), окно данных формируется логикой DD2 и DD4.3. Схема DD4.3 генерирует входные сигналы READ DATA нормированной длительности. Эти импульсы устанавливают счетчик DD2 в состояние 1100 В (операция фазировки делителя). Затем счетчик DD2 начинает увеличивать свое значение со скоростью 2 МГц для FM- и 4 МГц для MFM-режима. Скорость счета подобрана таким образом, чтобы на один период входных данных READ DATA (8 мкс для FM- и 4 мкс для MFM-режима) попало 16 опорных меток.

В таких условиях работы перепад сигнала на выходе третьего разряда счетчика указывает моменты начала и конца окна данных DW, образование которого завершает реализованный на DD3 делитель на два. Отметим, что начальная фаза сигнала DW для 18272A не имеет значения. Сигнал VCO в схеме автоподстройки не используется.

Схема рассчитана на работу в режиме с программно-управляемым обменом, когда линии связи (DRQ и DACK) с контроллером ПДП не используются. Этот режим эффективен в недорогих однопользовательских системах, к которым относятся одноплатные микроЭВМ семейства мМС. Выход IRQ прибора 18272A подключается в одному из восьми радиальных входов контроллера прерываний KP580BH59 или K1810BH59A (обеспечивается обслуживающие операции чтения-записи по прерываниям). Если ЦП не успевает выполнить ВВ данных за отводимое для этого время (64 мкс в FM- и 32 мкс в MFM-режиме), то возможен прямой программный опрос флага RQM. В любом случае с помощью сигнала IRQ обнаруживается начало фазы ответа. При его программном опросе необходимо прочитать состояние регистра запросов на прерывание IRR контроллера прерываний KP580BH59 или K1810BH59A и выделить разряд, соответствующий входу IRQ.

В схеме используется чисто программный способ формирования сигналов начального сброса RES и окончания операции чтения-записи TC. Для этого линии RES и TC подключаются к программно-доступному для ЦП выходному порту. Сигнал FDS CS формируется логикой выборки кристаллов МС.

Схема реализована на шести кристаллах, среди которых две программируемые логические матрицы типа K556PT1 [6]. Недорогой и компактный контроллер ГМД, построенный на этих универсальных БИС, встроены в одноплатную микроЭВМ ММС1207. Эта же схема вошла в состав отдельной платы связи с внешней памятью ММС5402. Кроме четырех приводов ГМД плата поддерживает два привода жестких магнитных дисков типа винчестер. Новые модули дополняют семейство ММС, расширяя сферу их практического приложения на области, где требуется недорогая, но емкая вторичная память.

Телефон 408-62-22, Москва

ЛИТЕРАТУРА

1. Дианов А. П., Щелкунов Н. Н. Одноплатные микроЭВМ семейства ММС // Микропроцессорные средства и системы.— 1990.— № 1.

2. Мячев А. А., Иванов В. В. Интерфейсы вычислительных систем на базе мини- и микроЭВМ / Под редакцией Б. Н. Наумова.— М: Радио и связь, 1986.

3. Коваленко В. А., Олейник А. В., Пархоменко Л. П., Солдатенко Л. М. БИС контроллера КР1818ВГ93 для накопителей на гибком диске // Микропроцессорные средства и системы.— 1986.— № 3.— С. 3—8.

4. Microprocessor and Peripheral Handbook. Order Number 230843—004. Intel, 1987.

5. Форс. Стандартная микропроцессорная шина, упрощающая задачи разработчиков микрокомпьютеров // Электроника.— 1978.— № 5.— С. 33—41.

6. Щелкунов Н. Н., Дианов А. П. Процедуры программирования логических матриц // Микропроцессорные средства и системы.— 1986.— № 2.— С. 71—76.

Статья поступила 16.02.88

УДК 681.327

Ю. А. Буняк

УСТРОЙСТВА СОПРЯЖЕНИЯ С ШИНОЙ МПИ НА ОСНОВЕ БИС СЕРИИ К1802

Интерфейс МПИ [1] применяется в наиболее широко распространенных микроЭВМ «Электроника 60» и ДВК, в микроЭВМ и микроконтроллерах на основе МКК серий К581, К588, К1801 и К1806 [2]. Для подключения к каналу центрального процессора (ЦП) микроЭВМ внешних устройств необходимо адаптировать процессы обмена данными с протоколом МПИ с помощью интерфейсных модулей (ИМ). Применение для этих целей специализированных БИС, в частности БИС обмена данными (ОД) КР1802ВВ1 и управления интерфейсом (УИ) КР1802ВВ2 [2], позволяет значительно уменьшить объем аппаратных средств.

Например, интерфейсный модуль на основе четырех БИС КР1802ВВ1 (D4...D7) обеспечивает обмен с 9...128 внешними регистрами (рис. 1). БИС ОД содержит 4-рядные регистр-счетчик RG0 и три регистра RG1...RG3, подключаемых произвольно к четырем информационным каналам в соответствии с управляющими сигналами адреса (A0, A1) и режимов чтения (R), записи (W), включения (EC). Кроме регистров БИС ОД содержит схему сравнения содержимого RG0 и RG3. Равенство состояний этих регистров сигнализируется сигналом на выходе F. На схеме приведены только два информационных канала, предназначенных для подключения к шине адреса-данных К АД Н ЦП (магистральный канал DX) и формирования адреса ВР (DA). Каналы DB и DC могут быть использованы как шины ввода-вывода в соответствии с конкретной задачей. Работает интерфейс следующим образом. По фронту сигнала К СИА Н в RG0 записывается адрес, сформированный ЦП в адресном цикле обмена. Одновременно в RG3 (D5...D7) записывается устанавливаемый пользователем адрес устройства в канале ЦП. Канал DA, D4 и D5 переключается на выдачу принятого адреса, выход шинного формирователя D9 переходит в высокоимпедансное состояние. Если содержимые RG0 и RG3 равны, то элемент 3И D1 формирует сигнал выбора устройства (ВУ). Синхронно с сигналами К ВВОД Н и К ВЫВОД Н элементы 2И-НЕ D2.1 и D2.2 управляют обменом данными в канале DX. С помощью внешней схемы необходимо переключить канал DX на обмен с К АД Н через RG1 или RG2. Если обмен выполняется через RG3, то устройство выбирается R—S-триггером (S=ВУ, R=К СИА Н). Элементами D2.3 и D2.4 формируется сигнал подтверждения К СИП Н. В качестве элементов D2, D3 и D8 можно выбрать соответственно ИМС К559ИП1П, К559ИП2П и К155ЛП9.

ИНТЕРФЕЙСЫ

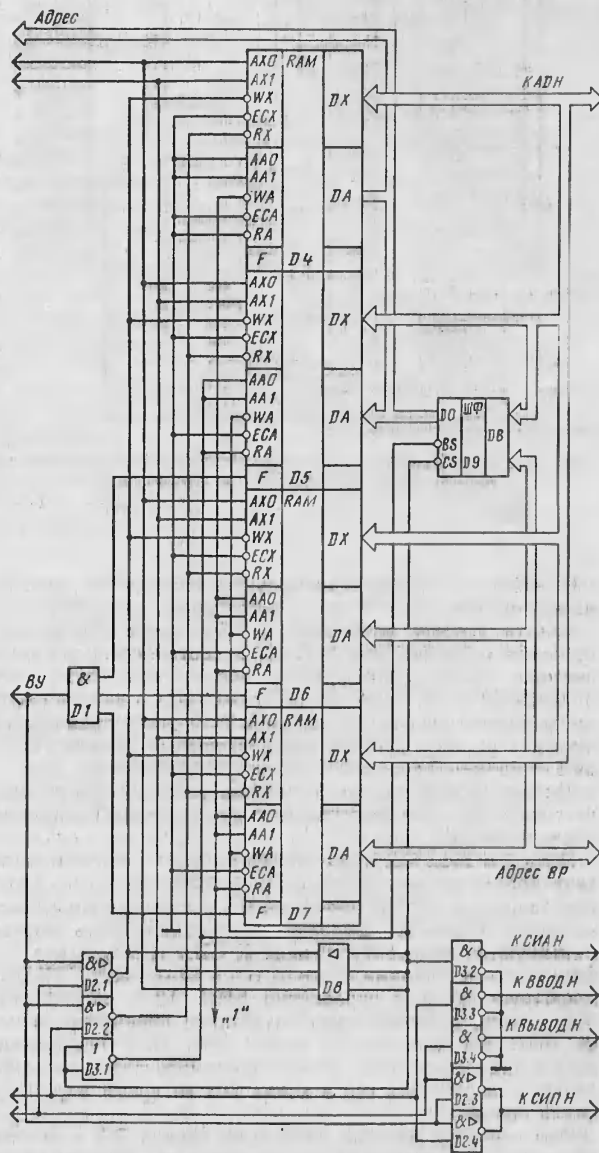


Рис. 1. Фрагмент схемы интерфейса на основе БИС КР1802ВВ1

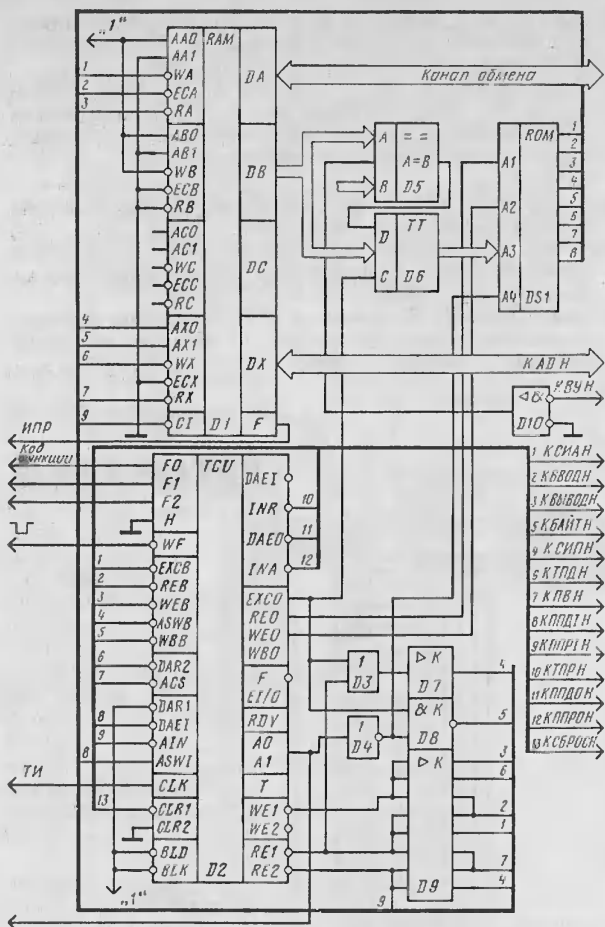


Рис. 2. Многофункциональное устройство сопряжения на основе БИС КР1802ВВ1 и КР1802ВВ2

С помощью многофункционального интерфейса (рис. 2) можно организовать обмен с регистрами, прямой доступ к памяти (ПДП) микроЭВМ и прерывания программы. Требования прерывания и ПДП в соответствии с интерфейсом МПИ адаптируются посредством БИС УИ КР1802ВВ2 (D2). Регистры БИС ОД (D1) в данной схеме распределены следующим образом: RG0 — счетчика адреса элемента массива данных, передаваемого в режиме ПДП, RG1 — ввода-вывода, RG2 — вектора прерывания, RG3 — конечного адреса массива в памяти, увеличенного на два. Честность адресов обеспечивается при смене мест старшего и младшего разрядов D1.

Регистры БИС ОД и устройства выбираются путем передачи адреса из канала DX в DB и сравнения его с адресом, заданным схемой сравнения D5, выполненной на основе ИМС К555СП1. Выборка и младшая часть адреса фиксируются по фронту сигнала К СИА Н в триггере D6. Синхронно с сигналами К ВВОД Н и К ВЫВОД Н ПЗУ DS1 генерирует сигналы управления БИС ОД и сигнал подтверждения, который через внутренние приемопередатчики БИС УИ поступает в канал ЦП. ПЗУ программируется так, чтобы было можно принимать адрес по каналу DX и передавать его в канал DB во время адресного цикла обмена.

При передаче массива данных из канала DA в память микроЭВМ в режиме ПДП формируется код функции обмена БИС УИ $\{F0, F1, F2\}_2 = \{0, 1, 1\}_2$, при передаче из памяти в канал DA — $\{F0, F1, F2\}_2 = \{0, 0, 1\}_2$. Каждый цикл ПДП синхронизируется импульсом на входе WF, БИС УИ формирует сигнал К Т ПД Н и при получении сигнала предоставления цикла К ППД Н устанавливает активный уровень сигнала К ПВ Н, управляя обменом данными. Цикл обмена сопровождается активным уровнем сигнала на выходе A1, который переключает выходы DS1 в пассивное состояние. Сигнал RE2 синхронизирует передачу в К АД Н адреса, RE1 — данных, WE1 — запись данных в канал DA. По окончании адресного цикла обмена счетчик RG0 увеличивает свое состояние на единицу. Каждый следующий цикл ПДП инициируется по фронту перехода сигнала на выходе A1 в пассивное состояние. Когда содержимое счетчика RG0 станет равным коду в RC3, на выходе F D1 формируется сигнал завершения обмена, который при необходимости может вызывать прерывание программы.

Для выполнения прерывания программы ЭВМ необходимо сформировать код функции $\{F0, F1, F2\}_2 = \{1, 0, 1\}_2$ и импульс WF. БИС УИ устанавливает сигнал К Т ПР Н и по завершении арбитража синхронно с сигналами К ППД Н и К ВВОД Н формирует сигналы чтения вектора RE1 и подтверждения К СИП Н.

На основе БИС КР1802ВВ1 и КР1802ВВ2 в виде платы изготовлен ИМ, согласующий промежуточный интерфейсный канал с каналом микроЭВМ типа «Электроника 60». ИМ выполняет следующие функции:

обмен ЭВМ с 32 внешними регистрами, четыре из которых принадлежат ИМ, а остальные — устройствам в интерфейсном канале;

прерывание программы по двум программно устанавливаемым векторам;

передачу массива данных произвольного объема из интерфейсного канала в память ЭВМ в режиме ПДП.

Прерывания программы по первому вектору выполняются по запросам от устройств в интерфейсном канале, по второму — при завершении передачи массива данных. Установлен следующий приоритет обслуживания запросов: прерывания по первому, второму векторам и ПДП. Обслуживание запросов сопровождается соответствующими сигналами подтверждения. Интерфейсный канал включает двустороннюю 16-разрядную шину данных, линии сигналов синхронизации ввода, вывода и выбора устройства, запросов и подтверждений прерывания, ПДП, а также линии для сигналов управления режимом работы внешних устройств, начальной установки и адресации регистров.

При проектировании встраиваемых микроконтроллеров не всегда можно создать магистральную структуру по типу микроЭВМ. В этом случае в состав платы контроллера целесообразно ввести средства управления интерфейсом. Например, если контроллер выполнен на основе МП К1801ВМ1 (К1801ВМ2), то могут быть использованы БИС УИ КР1802ВВ2 или БИС серий К588 [3, 4] и К1801 [5]. БИС УИ КР1802ВВ2 обладает тем преимуществом, что реализует все необходимые функции — прерывание программы и ПДП. К МП она подключается по схеме, аналогичной приведенной на рис. 2. Кроме того, БИС КР1802ВВ2 может работать как арбитр МПИ. Этот режим применен, например, в периферийном контроллере, работающем с описанным выше интерфейсом и выполняющем функции встраиваемого микропроцессорного блока в распределенных системах обработки данных и управления. Когда ЦП контроллера отключен, БИС УИ играет роль арбитра локальных шин и позволяет центральной микроЭВМ загрузить в его память программу, обмениваться данными с внешними устройствами. Если ЦП находится в активном состоянии, БИС УИ организует доступ к его шинам в режиме ПДП. Посредством БИС УИ контролируется выполнение программ контроллера. Обращением ЭВМ к одному из ее регистров в цикле вывода инициируется прерывание программы ЦП контроллера с одновременной передачей вектора. Контроллер выполнен в виде одноплатного модуля размерами 280×120 мм. Он включает ОЗУ и ПЗУ объемом по 4 Кбайт, селекторы адреса 16 внешних регистров и схемы сопряжения с интерфейсным каналом. Потребляемая мощность не превышает 5 Вт.

Рассмотренные функциональные схемы представляют собой два возможных варианта применения БИС ОД КР1802ВВ1 для сопряжения микроЭВМ с внешними устройствами. Совместное применение БИС ОД и УИ позволяет реализовать многофункциональный интерфейс с двумя или тремя независимыми каналами обмена данными.

286021, Винница, Хмельницкое ш., 93, ВПИ, каф. АИИТ; тел. 4-66-92

ЛИТЕРАТУРА

1. Межмодульный параллельный интерфейс МПИ. ОСТ 11.305.903-80.
2. Хвощ С. Г., Варлинский Н. Н., Попов Е. А. Микропроцессоры и микроЭВМ в системах автомати-

ческого управления.— Л.: Машиностроение, 1987.— 640 с.

3. Кушарев В. Н., Свиридович В. С., Чернуха Б. Н., Черняковский Д. Н., Бобков В. А. Контроллер прямого доступа к памяти К588ВТ2 // Микропроцессорные средства и системы.— 1988.— № 1.— С. 13—17.
4. Свиридович В. С., Черноусова Т. Г., Чернуха Б. Н., Бобков В. А. Контроллер прерываний К588ВН1 // Микропроцессорные средства и системы.— 1987.— № 5.— С. 3—5.
5. Микросхемы интегральные. Серия К1801. ОСТ 11 348.918-83.

Статья поступила 20.06.88

УДК 681.32

Г. Б. Герман, А. А. Сачук, А. А. Улыбин

СОПРЯЖЕНИЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60» С МИКРОПРОЦЕССОРНЫМИ УСТРОЙСТВАМИ

При создании радиотехнических систем часто возникает необходимость физического сопряжения серийно выпускаемых промышленностью и вновь разрабатываемых для данной системы устройств. Различия в сигналах и временных диаграммах работы устройств не позволяют обеспечить совместную работу без специальных блоков.

Описываемое устройство сопряжения (рис. 1) может быть использовано для организации программного обмена данными в комплексе; отличается схемотехнической простотой и надежностью работы.

Шинные формирователи усиливают сигналы шины адреса-данных и управления и передают информацию на

расстояние до 10 м; выполнены на микросхемах типа К559ИПЗ. Дешифраторы адреса регистров состояния и данных формируют сигналы выборки при обращении микроЭВМ к этим регистрам.

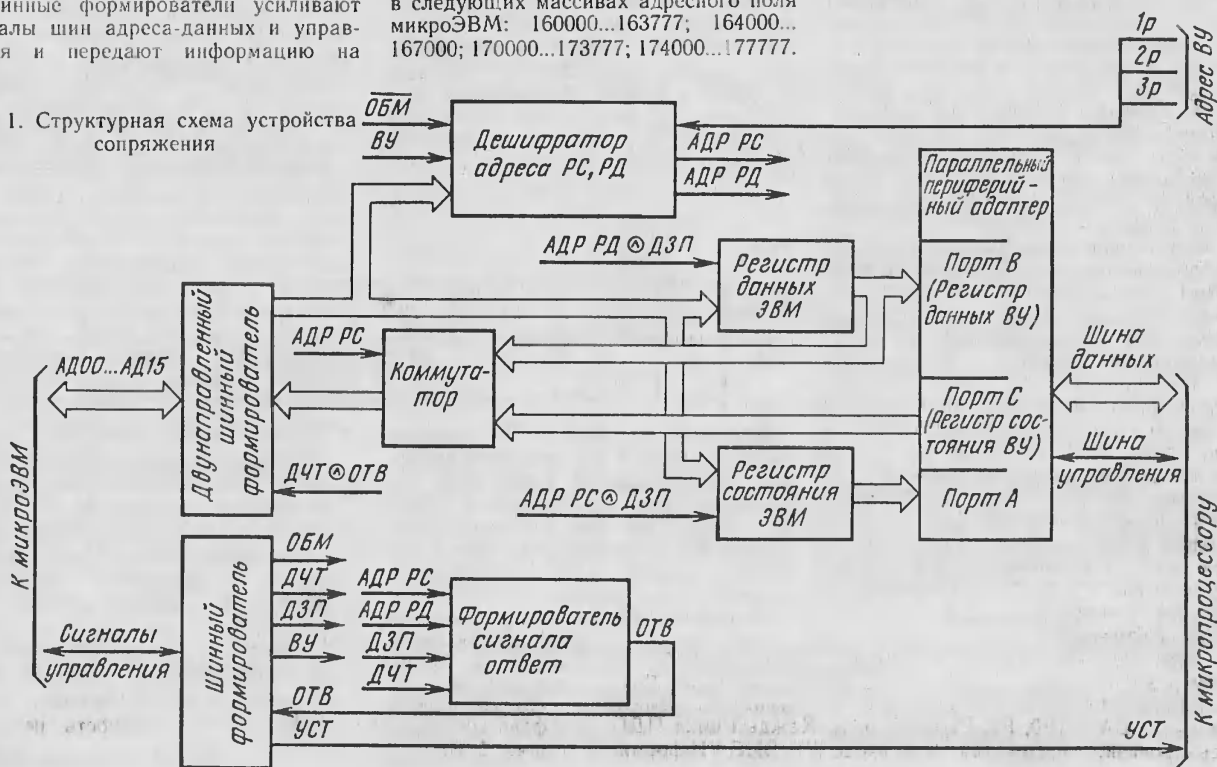
Адреса внешних устройств (ВУ) в адресном пространстве микроЭВМ «Электроника 60» — 160000...177777. При обращении к ним ЭВМ формирует сигнал ВУ.

Дешифратор может быть выполнен на ПЗУ и D-триггерах (рис. 2) и позволяет произвести дешифрацию любого из 256 адресов, продублированных в следующих массивах адресного поля микроЭВМ: 160000...163777; 164000...167000; 170000...173777; 174000...177777.

Младшие адреса АД01...АД10 и сигнал ВУ, поступающие на адресные входы и входы выборки ПЗУ, однозначно определяют в информационном пространстве ПЗУ адреса регистров данных и состояния, по которым программируется шестнадцатеричный код 00. По сигналу ОБМ регистр записывается в соответствующий D-триггер и запоминается на цикл обмена. Для расширения выборки пар адресов регистров ВУ до восьми используются три старших разряда ПЗУ, на которые может подаваться двоичный логический код в зависимости от места установки устройства сопряжения в системе.

При выборе адресов ВУ и программировании ПЗУ необходимо учитывать адреса штатных ВУ самой микроЭВМ (не должны совпадать).

Рис. 1. Структурная схема устройства сопряжения



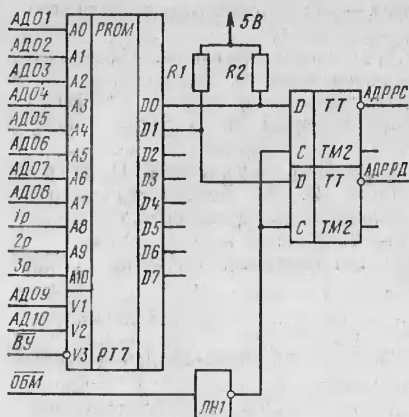


Рис. 2. Схема дешифратора адреса

Коммутатор подает информацию с регистра данных или состояния ВУ на общую шину ЭВМ в соответствии с адресом выбранного регистра. Он выполнен на микросхемах типа К530КП11.

ОТВЕТ (рис. 3) — синхронизирующий сигнал, возникающий при каждом цикле обмена данными между ЭВМ и ВУ. Формирователь собран на микросхемах К133ЛЕ1, К133АГ3 и К133ТМ2. Временная задержка определяет длительность задержки сигнала ОТВЕТ (1...3 мкс).

С помощью четырех 8-разрядных регистров производится обмен информацией: РДЭ — данные ЭВМ и РСЭ — состояния ЭВМ на микросхемах типа К585ИР12 с трехстабильным выходом, РДУ — данные ВУ и РСУ — состояния ВУ на микросхеме К580ВВ55 (параллельный периферийный адаптер — ППА). Порт В ППА в режиме 1 стробирует ввод данных с РДЭ при работе ВУ на прием и их вывод при передаче. Программно-аппаратные сигналы состояния ВУ и управления обменом формируются через порт С ППА. Через порт А в режиме 0 ВУ (микропроцессор) читает из РСЭ.

РСЭ и РДЭ — программно доступны по записи со стороны ЭВМ РСУ и РДУ — по чтению.

По сигналу УСТ, формируемому ЭВМ, ПО микропроцессора ВУ перезапускается (начальная установка).

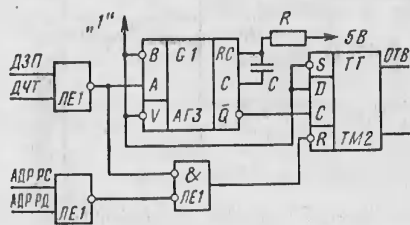


Рис. 3. Схема формирования сигнала ОТВЕТ

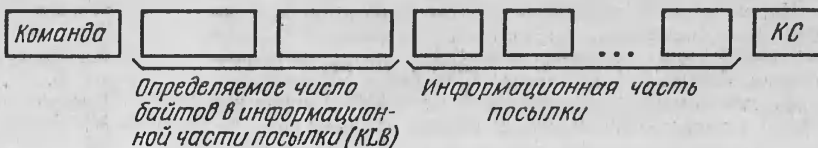


Рис. 4. Командно-информационная послылка:

КС — дополнительный байт для формирования контрольной суммы сообщения (сложение всех байтов по модулю 2^8 дает 16-ричный код А6)

Программный обмен данными между микроЭВМ «Электроника 60» и ВУ — их передача по инициативе и под управлением программ обмена — реализуется с помощью четырех регистров: РСУ, РСЭ, РДУ, РДЭ.

кабелей общей шины между ЭВМ и ВУ — 7,5 м. Сигналы общей шины согласованы со стороны ЭВМ и ВУ, подключенного к концу общей шины, резисторными делителями ($R_1=220$ Ом,

PCU

7	6	5	4	3	2	1	0
X	X	ВКЛ	ГОТ	ОШУ	ЧТУ/ЗПУ	ЗО/ЗАН	ДГУ/ДПУ

Аппаратные сигналы: ВКЛ — подача питающего напряжения на ВУ и ДГУ/ДПУ — синхронизации обмена. При ДГУ=1 ВУ подтверждает готовность формирования данных для передачи информации в ЭВМ, при ДПУ=0 — их прием.

Программные сигналы: ГОТ — готовность ВУ к работе по положительному результату прохождения встроенного теста; ОШУ — сигнал ошибки, формируемый ВУ при обнаружении информационных сбоев в процессе обмена; ЧТУ/ЗПУ — чтения-записи (при ЧТУ=1 ВУ принимает информацию, при ЗПУ=0 — передает); ЗО/ЗАН — запрос на обслуживание — занято (при ЗО=1 ВУ свободно и готово к обмену информацией, при ЗАН=0 — ВУ занято внутренним циклом работ); X — разряды, не анализируемые в процессе обмена.

$R_2=330$ Ом), включенными между шинами +5 В и корпусом.

Для повышения надежности работы применены программные методы увеличения достоверности передаваемой информации подсчетом байтов и контрольной суммы сообщений, а также трехкратного повторения обмена ЭВМ с ВУ при обнаружении случайных сбоев в связках управляющих сигналов.

290601, Львов, ул. Научная, ЛНИРТИ; тел. 63-83-90

ЛИТЕРАТУРА

1. Балашов Е. П., Григорьев В. Л., Петров А. Г. «Микро- и мини-ЭВМ», — Л.: Энергия, 1984.

РСЭ

7	6	5	4	3	2	1	0
X	X	X	X	ОШЭ	ЗПЭ/ЧТЭ	РАБ	ДПЭ/ДГЭ

Обмен между ЭВМ и ВУ производится командной или командно-информационной послылкой (один байт информации, рис. 4). Инициатор обмена — ЭВМ. При этом учитывается состояние ВУ: по сигналу ЗО в РСУ — ВУ готово к обмену, ЗАН — занято внутренним циклом работ.

Устройство обмена было реализовано и применено в многопроцессорном управляющем вычислительном комплексе. Центральная ЭВМ обеспечивала двусторонний обмен информацией с микропроцессорными ВУ со скоростью порядка 10 Кбайт/с. Суммарная длина

2. Лукьянов Д. А. ПЗУ — универсальный элемент цифровой техники // Микропроцессорные средства и системы. — 1986. — № 1 — С. 75.
3. МикроЭВМ. Семейство ЭВМ «Электроника 60». Практ. пособие в 8-ми кн. Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1988. Кн. 1.
4. Соучек Б. «Микропроцессоры и микроЭВМ», М.: Энергия, 1981.

Статья поступила 15.07.88

ИНТЕРФЕЙС ДЛЯ УПРАВЛЕНИЯ И АВТОМАТИЧЕСКОГО ИЗМЕРЕНИЯ НА ОСНОВЕ ПК ТИПА IBM PC XT/AT

Создание различных приборов и установок требует во многих случаях измерения распределения реальных физических величин (электрического или магнитного поля, температуры и т. д.) и сравнения их с расчетными. Так, например, при создании протяженных магнитных полей сложной конфигурации с большими градиентами для составления полной карты сформированного поля требуется провести измерения в нескольких десятках тысяч точек в трехмерном пространстве. Применение распространенных ПК IBM PC XT/AT открыло новые возможности в автоматизации измерений, обработке массивов данных, интерпретации, хранении и документировании результатов. Современное поколение традиционных измерительных приборов имеет стандартные интерфейсы (RS-232C, IEEE-488, HP-IB) подключения к компьютерам для дистанционного управления и обмена данными. А в измерительных системах необходимы также съём и обработка сигналов с различных датчиков, например для управления автоматическим перемещением детекторов.

Рассматриваемая измерительная система, построенная на основе ПК IBM PC XT/AT [1], была создана для автоматического измерения постоянного магнитного поля, обработки и документирования данных измерений. Интерфейс (рис. 1) осуществляет связь между IBM PC XT/AT и милливольтметром Щ1413 или ему подобным; кроме того, он через 8-разрядный вход принимает внешние сигналы, а через 8-разрядный мощный выход управляет внешними элементами (реле, двигателями) с помощью соответствующей управляющей программы.

Милливольтметр [2] работает в циклическом интегрирующем режиме, при этом циклы измерений и выдачи результатов на дисплей разделяются во времени. После команды «Пуск» (WRITE0) прибор выполняет один цикл измерения. Команда выдается по программе.

```
<WRITE0>-OUT &H300,0;
<READ0>=INP (&H300)
<READ1>=INP (&H301)
<READ2>=INP (&H302)
```

Интерфейс принимает сигналы (SW0...SW7) от датчиков через 8-разрядный входной порт (INPUTS). Сигналы, поступившие с датчиков электромеханики, управляющие движением детектора Холла в магнитном поле, информируют о его координатах.

В начале работы с интерфейсом программа инициализации должна разрешить интерфейсу в блоке (ITCONTROL) выдачу запросов прерывания и настроить контроллер прерываний INTEL 8259 ПК IBM PC XT/AT [3].

Если уровень выходного сигнала одного из датчиков (SW0...SW7) изменяется, через входной блок интерфейса (INPUTS) устанавливается Лог. 1 в соответствующий триггер (T0...T7) 8-разрядного регистра (REGISTERS). Сигнал с этого выхода регистра через блок (GATE) поступает на линию запроса прерывания (IRQ2) шины PC.

Источники запросов прерывания идентифицируются после считывания содержимого регистра (IT STATUS) (S0...S7). D-триггеры источников запросов сбрасываются при запуске Лог. 0 в регистр (IRQ CLEAR) (C0...C7), а сигнал разрешения выдачи запроса прерывания интерфейса в блоке (ITCONTROL) — при считывании регистра (IT STATUS).

```
<WRITE2>=OUT &H302,0
```

```
<READ3>=INP (&H303)
```

```
<WRITE3>=OUT &H303, DATA
```

8-разрядный выход (OUTPUTS) управляет реле (R0...R7). По команде

управляющей программы (WRITE1) данные (D0...D7) поступают в 8-битный регистр для включения выходных транзисторов (тип: KT815) с открытым коллектором. (DATA TRANSIVER), (FUNCTION DECODER) осуществляют связь между интерфейсом и шиной IBM PC XT/AT. Сигнал (RESET) предназначен для установки начальных условий.

```
<WRITE1>=OUT &H301, DATA
```

Интерфейс расположен на одной плате в стандарте IBM PC XT/AT. Плата содержит 22 микросхемы серии 74LS или K555. Потребление тока с шины +5 В составляет 0,5 А. Милливольтметр подключается с помощью разъема РП15-32ГВВ. Для подключения объекта управления используется разъем РП15-23ГВВ.

Программа управления и сбора данных разработана для ПЭВМ типа IBM PC/XT и состоит из трех основных частей: главная программа, подпрограмма обслуживания прерывания и подпрограмма измерений. Блок-схема общей части программы показана на рис. 2.

Главная программа начинается с разрешения уровня IRQ2 и прерывания интерфейса (IRQ2 ENABLE, (WRITE2)) после записи начального адреса подпрограммы прерывания в соответствующее место памяти [4] (SAVE IRA).

При включении внешнего устройства (WRITE1), отвечающего данной задаче, датчики через IRQ2-прерывание сообщают информацию о текущем положении детектора (в нашем случае —

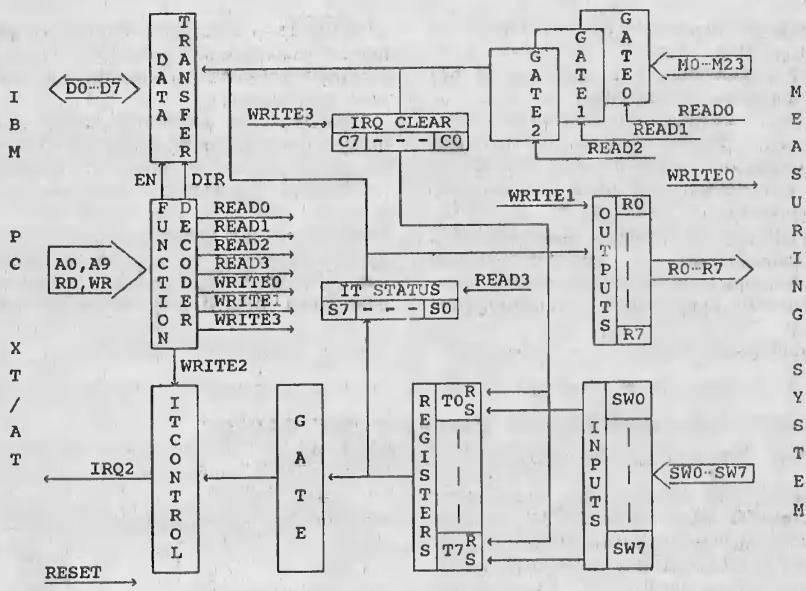


Рис. 1. Функциональная схема интерфейса. Блоки содержат следующие основные микросхемы:

(DATA TRANSFER) — K555AP6; (FUNCTION DECODER) — KP556PT4, K555ИД14, K555Л11; (ITCONTROL) — K555ЛП8, K555ТМ2; (GATE 0-2) — K555AP5; (OUTPUTS) — K555ИП23; (INPUTS) — K555ТЛ2; (REGISTERS) — K555ТМ2, K555Л11; (GATE) — K555ЛА2; (IT STATUS) — K555AP5; (IRQ CLEAR) — K555AP5

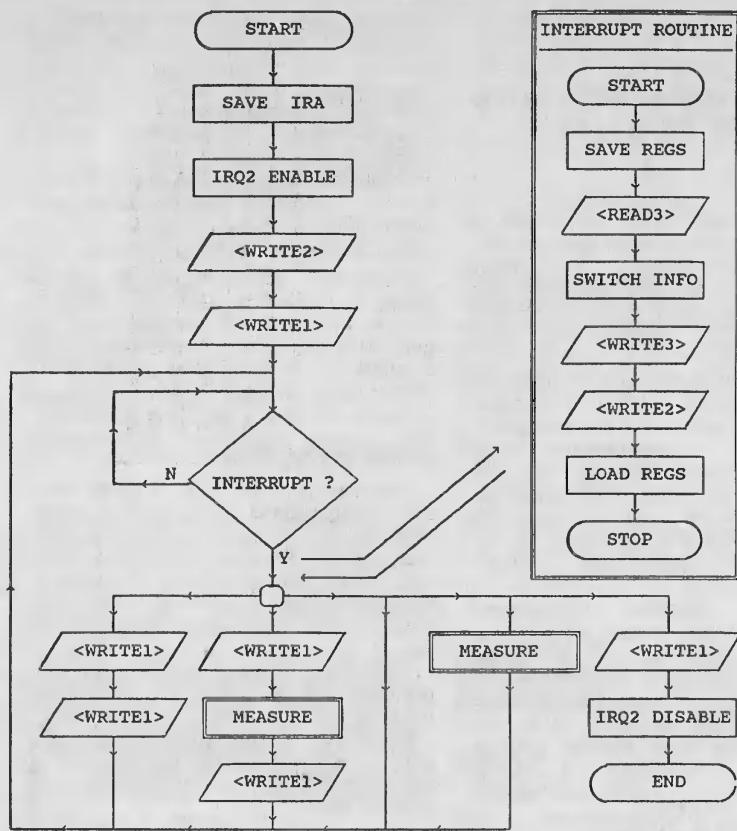


Рис. 2. Блок-схема части программы, обслуживающей интерфейс

о положении датчика Холла). Прерывание работы главной программы передает управление в подпрограмму обслуживания прерывания.

После сохранения системных переменных (SAVE REGS), считывания с (READ3) регистра (IT STATUS) и опознания датчиков, вызвавших прерывание (SWITCH INFO), (WRITE3) селективно вычеркиваются (сбрасываются соответствующие D-триггеры) и устанавливается разрешение прерывания интерфейса

(WRITE2). Затем вызывается сохраненное содержимое регистров (LOAD REGS) и управление передается в главную программу.

Принимая во внимание номер датчика, вызвавшего прерывания, и введенные начальные условия, главная программа производит одно из следующих действий: выключает обработавшее внешнее устройство и включает другое (другие); то же, но со считыванием информации (вызов программы измерения MEASURE); вызывает про-

грамму измерений или не производит никаких действий.

В зависимости от поставленных задач возможны разные комбинации. После выполнения любого из перечисленных действий программа переходит в режим ожидания следующего прерывания.

Подпрограмма измерений MEASURE начинается командой (WRITE0). Считывается циклически информация с порта (READ2) до тех пор, пока бит STATUS не сигнализирует об окончании измерений. Считывание портов (READ0), (READ1) (содержание порта (READ2) уже известно) позволяет, используя необходимые арифметические действия, получить действительное значение измеряемой физической величины (магнитного поля, температуры и т. п.), которое затем сохраняется в памяти или выводится на экран в текстовом или графическом виде.

Последний возможный ответ программы на прерывание — полное выключение всей установки. При этом отключаются все внешние устройства командой (WRITE1) и уровень прерывания (IRQ2) запрещается (рис. 2).

Следует отметить, что аппаратные возможности интерфейса шире реализованных в данной управляющей программе (например организация приоритета прерывания при одновременном получении нескольких входных сигналов (SW0...SW7)). Модульное построение управляющей программы позволяет изменить программу при решении конкретной задачи.

141980, Дубна Моск. обл., ОИЯИ; тел. 64-253

ЛИТЕРАТУРА

1. Персональный компьютер «Правец-16». Техническое описание. КМТ.— Правец, Болгария, 1986.
2. Ц1413 милливольтметр, Техническое описание, 1980.
3. Eggebrecht L. C. Interfacing to the IBM Personal Computer, Howard W. Sams Co. Indianapolis, USA, 1986.
4. Turbo Pascal 4.0 Owner's Handbook, Borland Int., 1987.

Статья поступила 16.06.89

УДК 681.325.5

А. А. Усолкин, Н. Н. Сороко

ПРОГРАММИРУЕМЫЙ ИНТЕРФЕЙС RS232C ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Интерфейс обеспечивает сопряжение системной магистрали МПИ с многочисленными отечественными и зарубежными устройствами и системами, имеющими выход на RS232C. Посредством этого интерфейса можно подключить микроЭВМ «Электроника 60» к модему и создать таким образом распределенные системы управления и территориально распределенные сети ЭВМ. Основные параметры интерфейса приведены ниже.

Максимальная скорость передачи, Бод	
RS232C	19 200
ИРПС	38 400
Тактовая частота, кГц	4608
Напряжение источника питания, В	5, 12, —12
Потребляемый ток, А, не более	
5 В	1
12 В	0,05
—12 В	0,05
Длина линии связи, м, не более	
RS232C	15
ИРПС	500

Модуль можно использовать и в качестве интерфейса ИРПС, если при этом универсальный синхронно-асинхронный приемопередатчик КР580ВВ51А (УСАПП) запрограммировать на асинхронную передачу данных, а вывод приемника соответствующей перемычкой соединить с оптроном. Перечень сигналов обмена и соответствие кодеров в различных стандартах приведены в табл. 1.

В состав модуля (см. рисунок) входят шинные формирователи, компаратор адреса, автомат прерываний, УСАПП, генератор тактовых импуль-

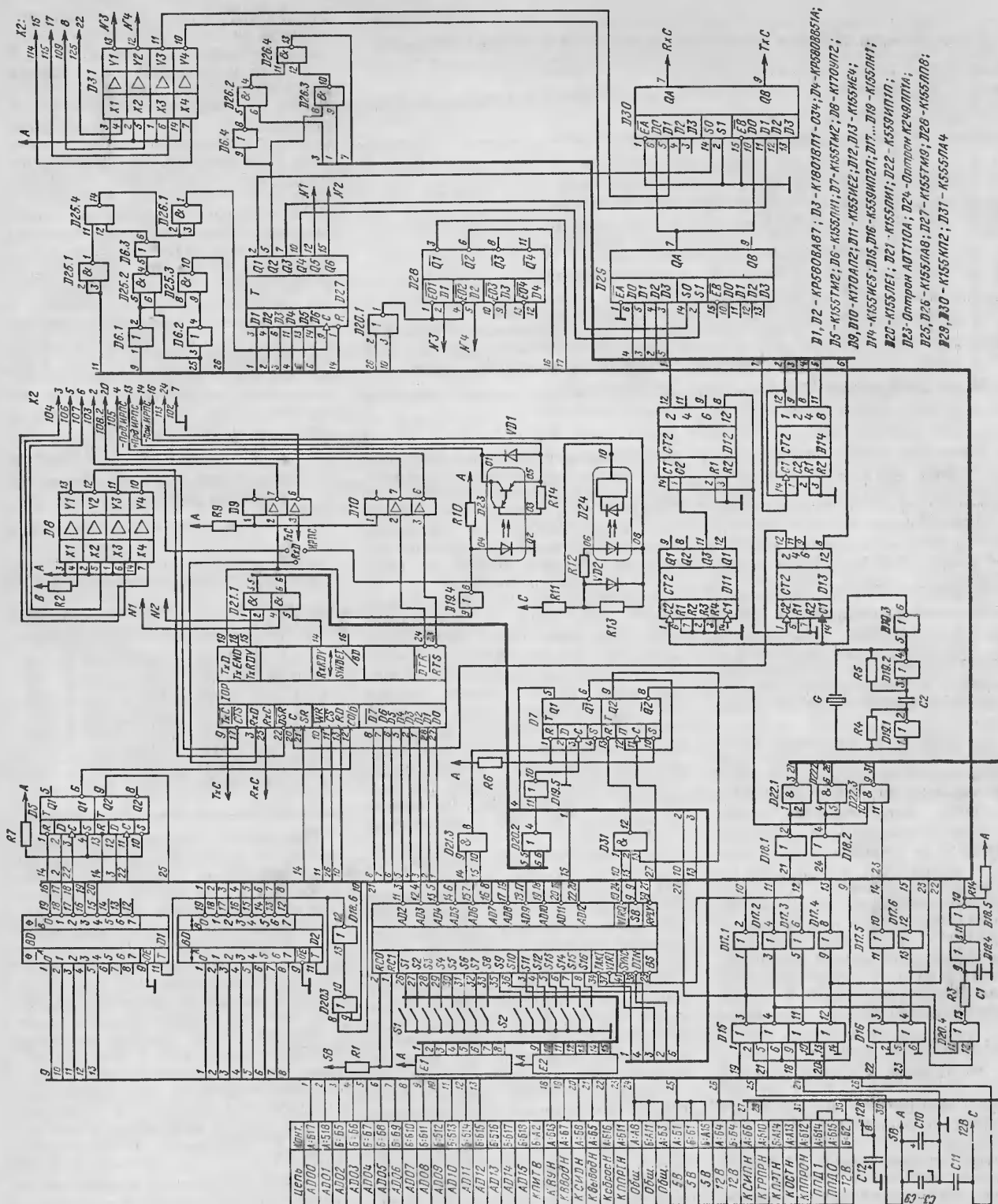


Таблица 1

Перечень сигналов обмена и соответствие кодировок в различных стандартах

Контакт	Кодирование в интерфейсе RS232C	Эквивалент ССИТ V.24	Соответствие сигналов УСАПП
1	AA	101	PG — защитное заземление
7	AB	102	SG — сигнальное заземление
2	BA	103	TXD — передаваемые данные
3	BB	104	RXD — принимаемые данные
4	CA	105	CTS — готовность к передаче
6	CC	107	DSP — готовность ПУ
20	CD	108.2	DTP — готовность компьютера
22	CE	125	Кольцевой индикатор
8	CF	109	Сигнальный детектор приемной линии
24	DA	113	Синхронизация элементов передаваемого сигнала DTE
15	DB	114	Синхронизация элементов передаваемого сигнала DCE
17	DD	115	Синхронизация элементов принимаемого сигнала

сов, блок частотной синхронизации, блок выборки внутренней и внешней синхронизации, регистр режима, преобразователи уровней, оптронная развязка. Шинные формирователи D1, D2, D15, D16, D17, D22 предназначены для согласования модуля с магистралью.

Компаратор адреса выполнен на БИС K1801ВП1-034 (D3). Адрес модуля устанавливается на выводах S1...S10D3 микропереключателями S1.1...S1.8 и S2.1, S2.2 (он должен находиться в адресном пространстве, отведенном для регистров внешних устройств, исключая стандартные адреса устройств ввода-вывода).

В адресной части машинного цикла состояния S1...S10 и AD3...AD12 сравниваются при наличии сигнала K BY H. Если состояния S1...S10 и AD3...AD12 аналогичны, вырабатывается сигнал SB, который запоминается в триггере на время всего машинного цикла, пока не снят сигнал CIA, и разрешает работу всего интерфейса.

Автомат прерываний собран на D3, триггерах D7.1, D7.2 и микросхемах D19.5, D20.2, D21.3, D31. Если в регистр режима в разряды 4 и 5 занесены Лог. 0, то прерывание маскируется элементами D21.1, D22.2. В противном случае при требовании прерывания УСАПП триггер защелкнется, низкий активный уровень пройдет на вывод VIRI микросхемы K1801ВП1-034, что вызовет обмен данными в режиме прерывания программы в стандарте «Электроника 60». Триггер D7.2 запрещает или разрешает прохождение сигнала ППРО в зависимости от присутствия или отсутствия сигнала ТПР.

Адрес вектора прерывания, задаваемый пользователем, устанавливает соответствующие состояния на выводах S12...S16 микропереключателем S2.3...S2.8 (адрес вектора прерывания приемника будет меньше адреса вектора передатчика на 48).

УСАПП программируется на выполнение различных протоколов последова-

тельной передачи данных и работает в синхронном и асинхронном режимах, путем записи в соответствующие регистры слов инструкции режима, служебных синхросимволов и инструкции команд. Инструкции записываются при наличии высокого уровня на входе CO/D (управление данных).

Блок управления выполнен на триггерах D5.1, D5.2 и микросхемах D6.1, D6.2, D25.1, D25.2, D25.6, D25.4, D6.3, D26.1. В адресной части машинного цикла триггеры защелкивают на время всего цикла разряды адреса A2 по фронту сигнала CIA. В зависимости от комбинаций A1 и A2 выбирается регистр команд, данных или режима (адреса регистров соответственно INNN0, INNN2, INNN4, где N — разряды, соответствующие состояниям на выводах S1—S10 микросхемы K1801ВП1-034).

Блок частотной синхронизации, состоящий из четырех счетчиков D11...D14, мультиплексора D29 и микросхем D6.4, D26.2, D26.3, D26.4, предназначен для получения частоты 4608 кГц, генерируемой тактовым генератором, набора частот 1600, 3200, 4800, 6400, 9600, 12800, 19200, 38400 Гц, необходимых для задания скоростей обмена интерфейса. Частота синхронизации приемника и передатчика устанавливается программно разрядами 1...3 регистра режима.

Мультиплексор D30 выбирает внутреннюю или внешнюю синхронизацию, обеспечивает внешнюю раздельную синхронизацию приемника и передатчика от ВУ или внутренней синхронизацию набором частот, вырабатываемых блоком частотной синхронизации. В последнем случае частоты синхронизации приемника и передатчика будут одинаковы. Если разряд 0 регистра режима установлен в 0 — синхронизация внутренняя, если в 1 — внешняя.

Регистр команд имеет формат 09|08|07|06|0|04|03|02|01|00 Первые восемь разрядов — разряды регистра

БИС УСАПП. Разряд 8 — обнаружение несущей, разряд 9 — индикатор вызова (доступны только по чтению).

Регистр режима D27 имеет формат 05|04|03|02|01|00. Разряд 0 — внутренняя и внешняя синхронизация, разряды 1...3 — частота синхронизации, разряд 4 — разрешение работы приемника по прерыванию. Если разряд установлен в 1, то прерывание разрешено, если в 0 — запрещено. Разряд 5 — разрешение работы передатчика по прерыванию. Если разряд установлен в 1, то прерывание разрешено, если в 0 — запрещено. Все разряды доступны только по записи, сбрасываются в 0 по сигналу СБРОС.

При работе в асинхронном режиме УСАПП имеет три подрежима, которые устанавливаются программно и отличаются друг от друга различным соотношением численных значений частот синхронизации передачи — приема к скорости передачи (табл. 2).

Таблица 2

Соответствие скоростей обмена разрядам регистра режима

Разряд регистра режима	Частота синхронизации, Гц			Скорость обмена, Бод		
				Подрегим		
				1:1	1:16	1:64
3	2	1				
1	1	1	1600	1600	100	25
0	0	1	3200	3200	200	50
1	1	0	4800	4800	300	75
0	1	1	6400	6400	400	100
0	0	0	9600	9600	600	150
1	0	1	12800	12800	800	200
0	1	0	19200	19200	1200	300
1	0	0	38400	384	2400	600

При работе в подрежиме 1:1 комбинация 100 разрядов 1...3 регистра режима запрещена, если модуль используется в качестве RS232C (максимальная скорость передачи RS232C не должна превышать 20 кбит/с), и разрешена, — в качестве ИРПС. В синхронном режиме работы скорости обмена соответствуют подрежимы 1:1.

Преобразователи уровней D31, D8, D9, D10 трансформируют напряжение с уровней TTL к уровням стандарта RS232C. Оптоны D23, D24 предназначены для гальванической развязки интерфейса с внешним устройством при использовании модуля в качестве ИРПС. Сигналы обмена обозначены на схеме в соответствии с стандартом ССИТ V.24

Модуль выполнен на печатной плате размерами 252×143×12 мм в стандарте микроЭВМ «Электроника 60». В качестве соединителя используется 25-штырьковый двухрядный разъем ОНП/П/-ЖГ-79/33,3 12.3 — В53.

167000, Ускольвар, ул. Свободы, 10, кв. 133, Сыктывкар, А. А.

Статья поступила 17.08.89

Н. А. Малахов, Н. В. Пиляр

ИНТЕРФЕЙСЫ МЫШЬ И КООРДИНАТНЫЙ ШАР К ПЭВМ ТИПА IBM

При вводе или коррекции графической информации, такой как принципиальные схемы, разводка плат, рисунки — мышь или координатный шар (КШ) незаменимы [1]. В КШ перемещение маркера задается вращением шара, закрепленного в неподвижном корпусе; в мышке пропорционально её перемещению по поверхности стола (механическое перемещение преобразовывается прерывателями и фотодиодами в последовательность импульсов, число которых пропорционально перемещению [2]). Большинство пакетов программ и системный драйвер в DOS 3.30 (MSMOUSE.SYS) ориентированы на устройство, которое работает специальными послылками через последовательный интерфейс RS-232 со следующими параметрами:

Скорость, Бод 1200
 Информация без контроля четности, бит. 8
 Стоп-бит 1
 Распределение байтов в одной информационной послылке:

- 1 — служебный (содержит информацию о состоянии трех кнопок);
- 2 — величина изменения координаты dX;
- 3 — величина изменения координаты dY;
- 4 — величина изменения координаты dX;
- 5 — величина изменения координаты dY;

Вторая пара dX, dY содержит информацию о перемещении, произведенном за время передачи первой пары, все байты dX и dY — знак перемещения в старшем бите.

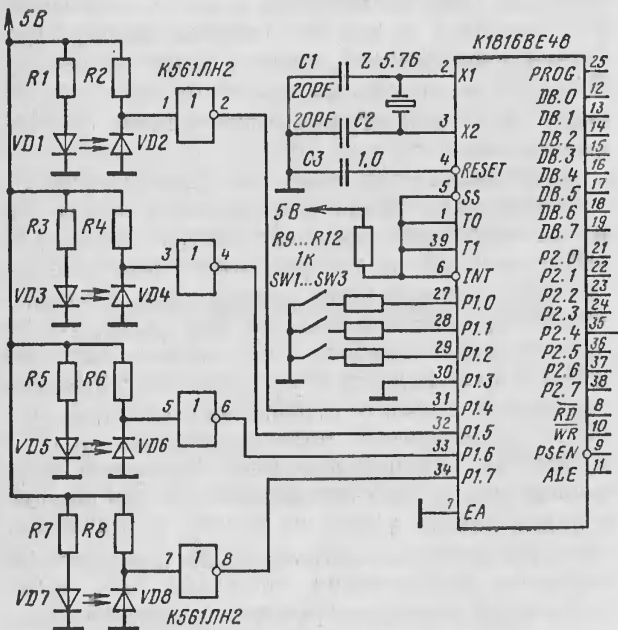


Рис. 1. Интерфейс на основе K1816BE48

За основу было решено взять ОЭВМ серии K1816, компактность, габаритные размеры и энергопотребление которой удовлетворяют всем требованиям. Работа по слежению за движением устройства, нажатие кнопок и организация передачи информации на ПК ведется программно.

Мы применили ОЭВМ K1816BE48 [2] с внутренним таймер-счетчиком, двумя портами ввода-вывода, внутренними ОЗУ и РПЗУ (рис. 1). Информация с фотодиодов (два на перемещение по X и два на перемещение по Y) через высокоомный буфер на K561ЛН2 подается на входы порта 1 микроЭВМ. На этот же порт подключены три кнопки через защитные резисторы (по ТУ на микроЭВМ).

МикроЭВМ работает по алгоритму, изображенному на рис. 2. При включении производится начальная инициализация портов, таймера и прерывания, затем опрос первого порта и слежение за любыми изменениями на нем. При появлении какого-либо изменения от фотодиодов при перемещении датчика или от кнопок поступившая информация анализируется (изменяется содержимое счетчиков dX и dY и организуется передача одной послылки из пяти байтов на линию связи), устанавливается флаг передачи, инициализируется таймер и разрешаются прерывания от него. Таймер в системе передачи определяет скорость передачи информации на ПК. При его переполнении работа микроЭВМ прерывается. Период счетных импульсов и начальное значение счетчика подбираются таким образом, чтобы частота прерываний была равна 1200 раз/с. На каждое преры-

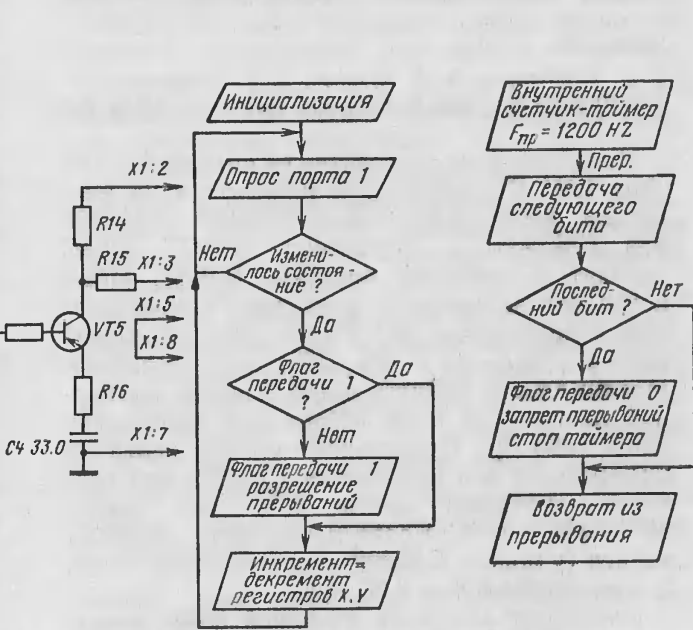


Рис. 2. Алгоритм работы микроЭВМ

вание микроЭВМ передаются пять информационных байтов на выход 1 второго порта. После передачи последнего бита флаг передачи сбрасывается, таймер останавливается и прерывания запрещаются.

Выходные импульсы формируются специальной схемой, использующей напряжение, подаваемое по выходной линии из ПЭВМ при отсутствии по ней передачи. Хочется отметить, что применяя ОЭВМ, изготовленную по КМОП-технологии, мощности сигнала ПЭВМ достаточно для питания всей схемы и не нужно напряжения питания +5 В.

Описанная схема подходит как для КШ, так и для мышки, она очень компактна и потребляет всего 15 мА.

Вместо микросхемы К1816ВЕ48 можно применить микроЭВМ КР1816ВЕ035, в которой нет внутренней памяти программ, добавив регистр адреса на К555ИР23 и РПЗУ программ на К573РФ2. Потребление такой схемы — 150 мА, +5 В. Ее габаритные размеры больше, и хотя плата легко размещается внутри корпуса координатного шара, для мышки не подходит. 141980, Дубна Моск. обл., ОИЯИ; тел. 6-48-95, 6-43-05

ЛИТЕРАТУРА

1. Фоли Дж., Дэм А. Основы интерактивной машинной графики. Пер. с англ. — М.: Мир, 1985.
2. Воробьев Н. В., Безобразов В. С. Графический манипулятор «мышь» для персональных ЭВМ // Микропроцессорные средства и системы. — 1988. — № 3. — С. 57—60.
3. Микропроцессоры и микропроцессорные комплекты интегральных микросхем. — М.: Радио и связь, 1988. — Т. 1.

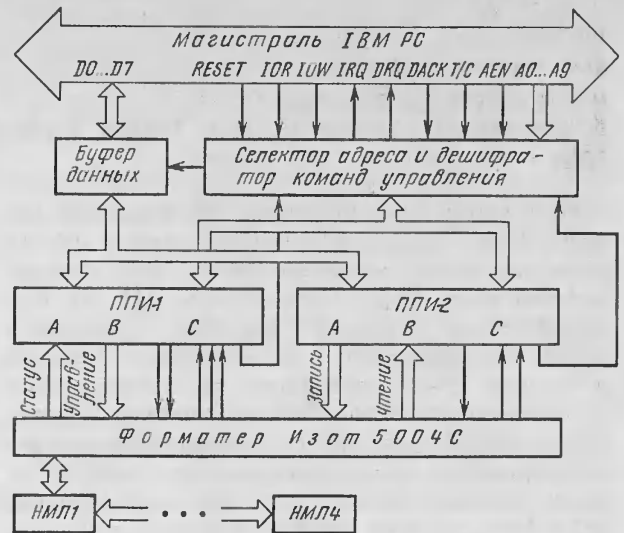
Статья поступила 17.05.89

УДК 681.326.3

Е. В. Комиссаров, П. А. Кулинич, В. В. Сидоркин ИНТЕРФЕЙС НМЛ СМ5309 ДЛЯ ПК ИВМ РС

Предлагаемое устройство — интерфейс ПК типа ИВМ РС ХТ/АТ для накопителя на магнитной ленте (НМЛ) типа СМ5309 (СМ5306 или СМ5308) с форматером ИЗОТ 5004С (входит в комплект поставки НМЛ). НМЛ позволяет записывать и считывать информацию с плотностью 800 и 1600 Вр/с с помощью БВН-1 [NRZ—1] и ФК [PE] соответственно. Максимальная ёмкость одной кассеты магнитной ленты — 20 и 40 Мбайт при плотностях 800 и 1600 Вр/с. Предельная скорость передачи информации для накопителя СМ5309 при скорости протяжки ленты 1,14 м/с равна 36 Кбайт/с и 72 Кбайт/с при двух способах записи (у модели СМ5306 скорость ленты равна 2 м/с, у СМ5308 — 0,63 м/с).

Интерфейс выполнен на одной плате конструктива ИВМ РС/ХТ и вставляется в свободный разъем расширения ПК. Он соединяется с фор-



Функциональная схема интерфейса магнитофона

матером кабелем из скрученных пар и обеспечивает подключение до четырех НМЛ. Питание платы осуществляется от источника питания ПК.

Форматер включается между интерфейсом и накопителями, управляет движением ленты магнитофонов, контролирует запись и чтение согласно стандартам. Он производит подготовку к записи и синхронизацию данных по разным дорожкам во время чтения. Наличие форматера существенно облегчает функции интерфейса и упрощает его конструкцию.

Схема интерфейса выполнена на основе двух БИС программируемого параллельного интерфейса (ППИ) КР580ВВ55 и микросхем серий ТТЛ средней и малой степени интеграции. Обмен информацией производится по каналу ПДП, при этом используются контроллер ПДП i8237 и контроллер прерываний i8259А, имеющиеся в составе ПК.

Функциональная схема интерфейса приведена на рисунке. Обмен информацией между ПК и магнитофоном осуществляется через два ППИ, каждый из которых имеет 8-разрядную двунаправленную шину данных и три 8-разрядных канала ввода-вывода (А, В и С). Подготовка интерфейса к работе, контроль его состояния и НМЛ производится с помощью программного обращения к восьми портам ввода-вывода двух ППИ. Дешифратор адреса выполнен на ПЗУ КР556РТ4. Двунаправленный буферный формирователь данных активизируется при распознавании адреса одного из портов, а также при передаче данных по каналу ПДП. Направление передачи определяется сигналами IOR, IOW.

Сигналы состояния НМЛ и интерфейса вводятся по каналу А и части канала С ППИ-1; управления НМЛ выводятся по В и остав-

шейся части С. Запрос прерывания к процессору (IRQ) по окончании операции НМЛ (если имеется разрешение в регистре управления ППИ-1) вырабатывается по сигналу С3 канала С. Информация по каналу ПДП при записи и чтении данных передается каналам А и В ППИ-2 соответственно. Линии канала С этой микросхемы используются для синхронизации обмена данными и, частично, для управления состоянием накопителя.

Работа с интерфейсом. Обмен информацией инициируется программно со стороны ЭВМ. Для этого в порты контроллера ПДП i8237 записываются управляющие коды, адрес первого элемента буфера и число пересылаемых слов, а в регистр расширения адреса — старшие биты адреса (при операциях чтения и записи). При работе по прерыванию необходимо программно инициализировать контроллер прерываний i8259A, подготовить подпрограмму обслуживания и дать разрешение на прерывание, установив соответствующий бит канала С в ППИ-1. Код команды и номер магнитофона по каналу В ППИ-1 подаются в форматер, который с помощью двух линий канала С активизируется, формируя сигнал СТАРТ. Для разрешения работы интерфейса по каналу ПДП (при чтении и записи) нужно также установить бит С5 ППИ-2.

Временная синхронизация пересылки данных между форматером и шиной ПК осуществляется импульсами сопровождения воспроизведения или записи (ИСВ, ИСЗ). При этом форматер и канал ППИ-2 для временного хранения байта данных обмениваются информацией. Во время операций чтения или записи по сигналу запроса к контроллеру ПДП (DRQ) начинается пересылка данных между памятью ПК и соответствующим каналом ППИ-2. Об окончании обмена информацией можно узнать по прерыванию, или проверяя программно состояние сигнала С3 ППИ-1. По статусной информации каналов А и С ППИ-1 анализируются текущее состояние интерфейса, магнитофона и возможные ошибки после окончания выполнения любой операции.

Назначение отдельных разрядов статусного слова:

- 0 — Идентификатор ФК
- 1 — Защита записи
- 2 — Корректируемая ошибка
- 3 — Маркер «конец файла»
- 4 — Состояние перемотки
- 5 — Состояние БВН-1
- 6 — Накопитель не готов
- 7 — Некорректируемая ошибка
- 8 — Начало ленты
- 9 — Конец ленты
- 10 — Задержка магистрали ЭВМ

Программное обеспечение реализовано в виде пакета подпрограмм нижнего уровня, тестовых программ и диалоговой системы ведения архи-

вов. Подпрограммы пакета написаны на ассемблере и позволяют выполнять набор стандартных операций для магнитофона: чтение-запись информации и ленточного маркера, пропуск рекордов вперед-назад, перемотку магнитной ленты на точку загрузки, а также, служебные функции (получение статусной информации о состоянии магнитофона и интерфейса, определение начального адреса массива данных в странице памяти). Подпрограммы вызываются из программ на языках Фортран-77 и Си.

Программные вызовы для работы с интерфейсом:

CALL MTR (IBUF, NBYTE, MT, NOK)	Чтение
CALL MTW (IBUF, NBYTE, MT, NOK)	Запись
CALL SKIPF (NRECORD, MT, NOK)	Пропустить вперед
CALL SKIPB (NRECORD, MT, NOK)	Пропустить назад
CALL REWMT (MT, NOK)	Перемотать
CALL EOF (MT, NOK)	Запись EOF
CALL ADR (IBUF, IADR)	Адрес в странице

Параметры:

IBUF	Массив из NBYTES
NBYTES	Число байтов информации для чтения-записи (INTEGER*2 или INTEGER*4)
MT	Номер выбираемого магнитофона (INTEGER*2 или INTEGER*4) (допустимы 0, 1, 2 и 3)
NOK	Число считанных-записанных байтов (INTEGER*4), если оно больше 0 (или рекордов для операций ПРОПУСК) Наличие метки Конец файла, если NOK=0 Сохраняемое статусного регистра, если NOK<0
NRECORD	Число рекордов для ПРОПУСКА (INTEGER*2 или INTEGER*4)
IADR	Адрес первого байта массива IBUF

В IBM PC контроллер ПДП работает в пределах только одной страницы памяти (вся память разбита на страницы по 64 Кбайт), поэтому изменить ее адрес можно только повторной установкой регистра расширения. Необходимо разместить массив IBUF внутри одной страницы так, чтобы выполнялось условие: $IADR + NBYTE \leq 65536$.

Тестовое обеспечение предназначено для настройки и отладки аппаратной части интерфейса и НМЛ. С его помощью проверяется правильность работы портов ввода-вывода и корректность записи-чтения различных типов данных на магнитной ленте во всех режимах, контролируются все основные операции НМЛ.

Сервисное обеспечение позволяет создавать на магнитной ленте резервные копии (Backup) для «твердых» и «гибких» дисков, файлов и групп файлов. Возможны также операции поиска, сравнения и восстановления (Restore) файлов. Диалог с пользователем организован в форме экранного меню.

141980, Дубна Моск. обл., ОИЯИ, лаб. ЛЯП;
тел. 6-25-20, 6-28-78

Статья поступила 30.10.89

УДК 681.324

П. А. Бабкин, Е. Ю. Несмелов

МНОГОКАНАЛЬНОЕ УСТРОЙСТВО ПРЕРЫВАНИЙ ДЛЯ МАГИСТРАЛИ МПИ

При управлении сложными объектами с помощью ЭВМ эффективно используется механизм прерываний программы центрального процессора. Обслуживание прерываний от большого числа источников затруднительно из-за линейного увеличения сложности оборудования в зависимости от числа запросов. Так, для обслуживания 16 векторных прерываний в ЭВМ МС1201, МС1212 необходимо около 40 корпусов ИС [1].

Восьмиканальный мультиплексор последова-

тельной связи (МПС) предназначен для управления распределенным оборудованием. В качестве элементной базы выбраны доступные микросхемы серии КР580. Сопряжение БИС КР580ВН59 с магистралью МПИ позволяет обрабатывать 15 запросов прерывания от восьми последовательных присоединителей (рис. 1). МПС выполнен на типовой полуплате ЭВМ «Электроника 60М».

Приведенный способ сопряжения отличается гибкостью программно задаваемых режимов маскирования и приоритетов, а также возможностью программного назначения кодов векторов прерываний [2, 3]. В общем случае возможно каскадное наращивание числа запросов до 64 [4].

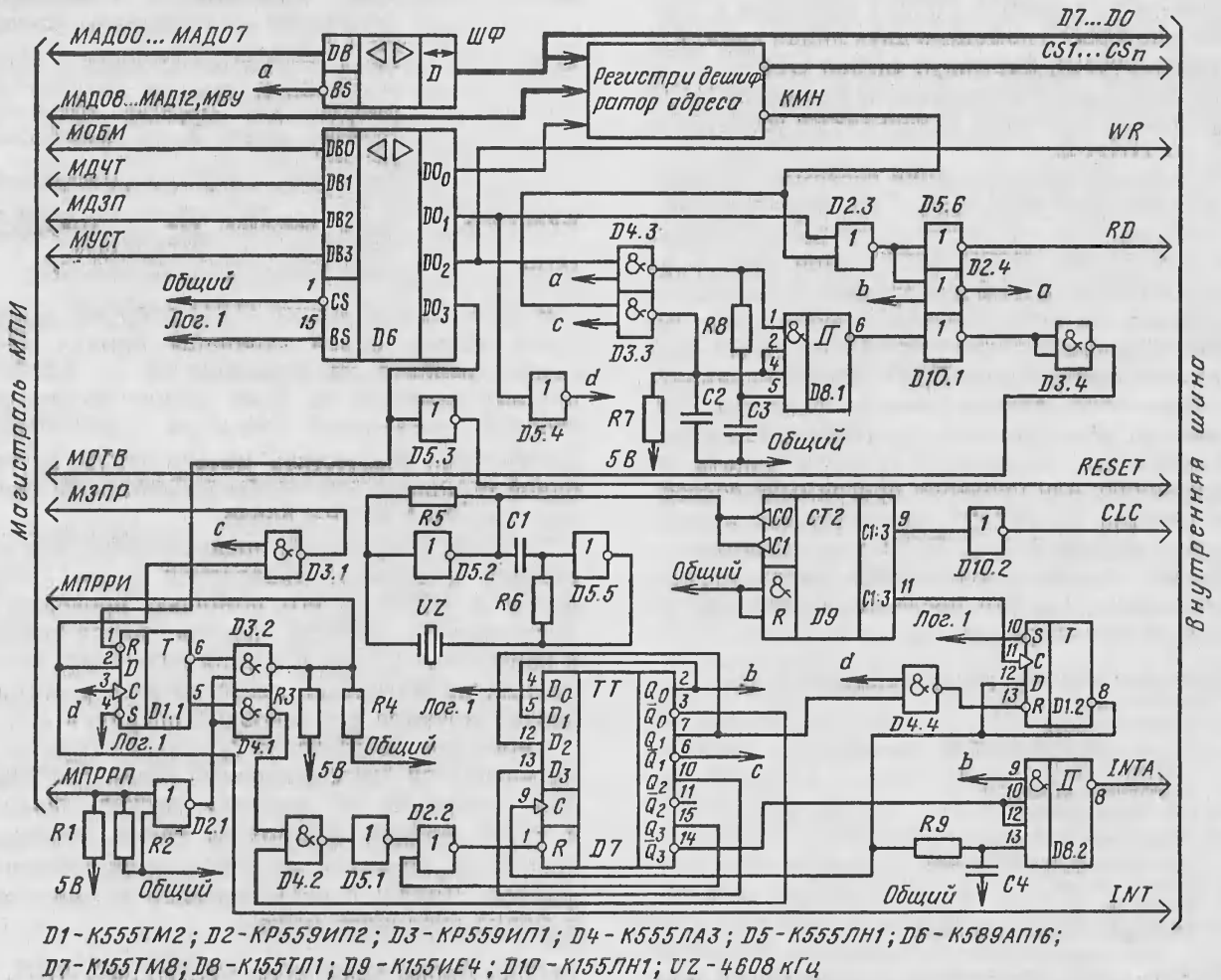


Рис. 1. Электрическая схема контроллера прерываний

Через шинные формирователи (ШФ) осуществляется обмен данными между магистралью ЭВМ и внутренней шиной МПС, выдача в магистраль ЭВМ вектора прерывания, а также адресация МПС. Направление передачи определяется сигналом на выводе BS, поступающим с выхода элемента D2.4.

Дешифратор адреса вырабатывает сигналы $CS_1...CS_n$ выбора БИС МПС и сигнал выбора МПС КМН (низкие активные уровни).

Сигнал чтения RD появляется на внутренней шине МПС только в циклах чтения. Элемент D2.3 запрещает формирование сигнала RD при вводе вектора прерывания, так как совместное появление сигналов INTA и RD на входах БИС КР580ВН59 приводит к нарушению ее функционирования.

Формирователь магистрального сигнала MOTB выполнен на элементах D3.3, D3.4, D4.3, D8.1 и D10.1. БИС КР580ВН59 подключается непосредственно к внутренней шине МПС. Через элемент D3.1 запрос прерывания INT от БИС КР580ВН59 выдается в магистраль ЭВМ (МЗПР). Узел, выполненный на элементах D1.1, D2.1, D3.2, D4.1, обеспечивает требуемую последовательность прохождения входного и выходного сигналов предоставления прерывания МППРП и МППРИ, а также формирует

на выходе элемента D4.1 внутренний сигнал предоставления прерывания, разрешающий работу формирователя внутренних сигналов подтверждения прерывания INTA.

Формирователь сигналов INTA, собранный на элементах D1.2, D2.2, D4.2, D4.4, D5.1, D7., D8.2, обеспечивает генерацию по внутреннему сигналу предоставления прерывания (выход D4.1) трех последовательных импульсов, что требуется для передачи БИС КР580ВН59 трехбайтного кода. Центральным процессором ЭВМ в качестве вектора прерывания принимается второй байт. Триггер D1.2 задерживает второй импульс INTA до завершения передачи вектора прерывания и снятия процессором сигнала МДЧТ (рис. 2).

Формирователь тактируется от встроенного генератора (элементы D5.2, D5.5, D9, D10.2). Счетчик D9 вырабатывает тактовые импульсы для формирования сигналов INTA (вывод 11 D9) и тактирования БИС приемопередатчиков. Частота импульсов задающего генератора 4608 кГц. Инвертор D10.2 позволяет максимально использовать быстродействие тактируемых БИС КР580ВН53 в программируемых генераторах скорости передачи.

Телефон 26-04-89, Новосибирск

ЛИТЕРАТУРА

1. Гойхман П. А. и др. Устройство приоритетного прерывания для микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1986.— № 1.— С. 73—75.
2. Аюпсв Р. М., Чабан С. Д. Восьмиканальный модуль последовательного обмена для микроЭВМ «Электроника 60» // Микропроцессорные средства и системы — 1987.— № 2.— С. 80—82.
3. Лукьянов Д. А. КР580 — автоматизация без проблем! // Микропроцессорные средства и системы.— 1985.— № 1. С. 87.
4. ОСТ П 348.917-82. Микросхемы интегральные полупроводниковые. Серия КР580. Руководство по применению.

Статья поступила 24.02.89

УДК 681.325.54

М. Г. Меш

ПРОГРАММНЫЙ СЧЕТЧИК ПРЕРЫВАНИЙ ПО ТАЙМЕРУ ДЛЯ ПЭВМ БК-0010

Очень часто пользователю ПЭВМ БК-0010 приходится отказываться от разработки прикладных программ на языке высокого уровня Фокал, входящего в состав программного обеспечения данной ПЭВМ, из-за ограниченного быстродействия интерпретатора. Но при этом теряется возможность пользоваться встроенными функциями языка Фокал, в том числе функцией FCLK, которая позволяет работать с таймером, не вникая в подробности процесса обслуживания прерывания по таймеру

«Микропроцессорные средства и системы» № 4, 1990 87

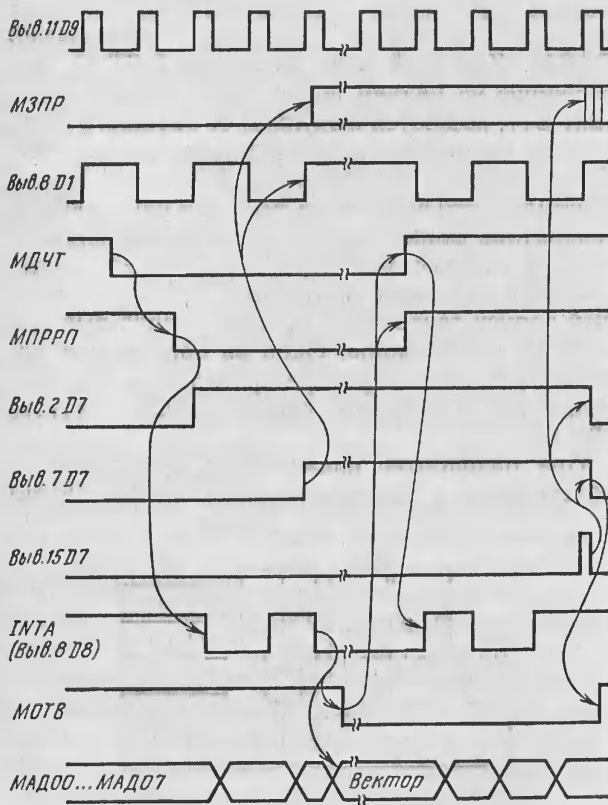


Рис. 2. Временная диаграмма работы формирователя сигналов INTA

и подсчета количества прерываний. При разработке программ в машинных кодах эти вопросы вплотную встают перед программистом.

Запросы на обслуживание прерываний по таймеру формируются внешним генератором импульсов и поступают на порт ввода-вывода (контакт В1).

Обслуживание прерываний по таймеру начинается через вектор 100_8 при условии равенства нулю разряда 7 слова состояния процессора. Подпрограмма обслуживания прерывания по таймеру расположена в памяти с адреса 132224_8 по адрес 132236_8 . В ОЗУ выделены две ячейки специального назначения: ячейка с адресом 1746_8 (счетчик прерываний), содержание которой увеличивается на 400_8 при каждом обслуживании прерывания, и ячейка с адресом 1750_8 (счетчик переносов), в которой суммируются переносы, возникающие при заполнении счетчика прерываний.

Для работы в реальном времени и формирования временных интервалов пользователь должен организовать программный счетчик прерываний по таймеру. Предварительно в ОЗУ произвольно резервируются две ячейки с адресами STMO (для хранения начального значения счетчика переносов MO) и STNO (для хранения начального значения счетчика прерываний NO).

Запуск программного счетчика прерываний по таймеру осуществляется вызовом подпрограммы TO. Для удобства эта и другие программы записаны в мнемонике языка ассемблера.

Подпрограмма KT позволяет в регистрах R0, R1 сформировать с удвоенной точностью количество прерываний по таймеру, обработанных с момента запуска программного счетчика (вычислить количество «тиков» таймера).

Проиллюстрировать работу с программным счетчиком можно на следующем примере. После

включения прибора датчик опрашивается в течение одной минуты. Если датчик за это время сработал, осуществляется переход к выполнению основной программы MAIN, если не сработал, работа заканчивается. Частота поступления запросов на прерывание по таймеру — 50 Гц.

Алгоритм программы, реализующей эту задачу, приведен на рисунке.

Емкость программного счетчика прерываний при частоте запросов на обслуживание прерываний 50 Гц позволяет работать с ним после включения микроЭВМ более 46 ч. Этого времени вполне достаточно для выполнения рабочих программ БК-0010.

Телефон 350-86-69, Ленинград

Статья поступила 12.07.88

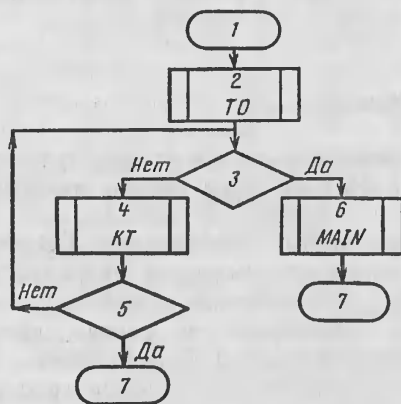
УДК 681.3

Е. А. Рудометов, А. Ю. Коленников

ПРОГРАММНО-АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ФУНКЦИИ ТАЙМЕРА ПЭВМ БК-0010

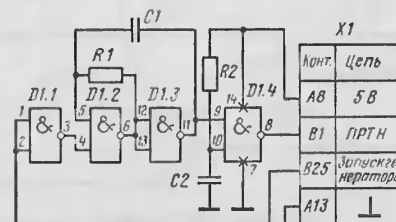
В автоматизированных системах научных исследований и управления технологическими процессами возможно применение ПЭВМ БК-0010 [1, 2], которое ограничено отсутствием таймера. Реализовать эту функцию можно с помощью программно-аппаратного обеспечения системы обработки внешних прерываний ПЭВМ. Для этой цели на линию ПРТ Н, выведенную на разъем порта ввода-вывода (контакт В1), подаются импульсы от внешнего генератора, схемотехническое решение которого выбирают исходя из характера решаемых задач. Учитывая, что включение ЭВМ при работающем генераторе может вызвать блокировку клавиатуры в режиме монитора, необходимо предусмотреть задержку включения либо программный запуск генератора от порта ввода-вывода, либо их комбинацию. Один из простейших вариантов схемы генератора приведен на рисунке. Значения элементов соответствуют частоте 100 Гц.

При разработке прикладного программного обеспечения с использованием языка Фокал



Алгоритм программы:

1 — начало; 2 — запуск программного счетчика; 3 — датчик сработал?; 4 — подсчет числа «тиков» таймера; 5 — одна минута прошла? (5760 «тиков» таймера); 6 — основная программа; 7 — останов



Упрощенная схема генератора:

R1—200; R2—6,8к; C1—6,8мкФ; C2—470,0; X1—СНП58—64/95×9 р; D1—K155ЛАЗ

число импульсов, поступивших по линии ПРТ Н, можно получить, применяя встроенную функцию FCLK ().

Для функции таймера в программах, разработанных на языке Фокал, частота задающего генератора выбирается с учетом времени непрерывной работы таймера и точности представления переменных, которая для данного языка составляет 5,6 значащих цифр.

Транслятор языка БЕЙСИК и драйвер-мониторная система, включенные в состав резидентного математического обеспечения ПЭВМ, не содержат программных средств, аналогичных оператору FCLK () языка Фокал. Для обеспечения функции таймера при программировании на БЕЙСИКЕ и ассемблере необходима программа в кодах, разработка которой выполняется с помощью соответствующих средств системного монитора ПЭВМ или математического обеспечения, совместимых по системе команд ЭВМ.

Трансляция на мини-ЭВМ СМ1420 в системе NTS, а проверка и отладка — на ПЭВМ БК-0010. Представленная программа в соответствии с частотой внешнего генератора обеспечивает расчет и вывод текущего времени (часы, минуты, секунды) в поле информационной строки. При этом данные значения сохраняются в трех ячейках памяти и доступны для программ пользователей. Считывание информации о текущем времени позволяет фиксировать произвольные моменты времени, что необходимо при использовании микроЭВМ в контуре управления.

Основные параметры программы: адрес загрузки — 37472, частота внешнего генератора (по умолчанию равна 100 Гц) — 37724, «тики» — 37734, «секунды» — 37736, «минуты» — 37740, «часы» — 37742. Для запуска программы достаточно установить необходимые значения начального времени, частоту генератора (если она не равна 100 Гц), записать в ячейку 100 начальный адрес — 37472 и запустить генератор таймера. Работоспособность программы сохраняется при ее загрузке в любую область памяти.

Данную программу можно присоединить к программам на БЕЙСИКЕ. Для этого программа в кодах должна иметь имя с расширением BIN: TIME.BIN. В этом случае ввод осуществляется командой BLOAD TIME с последующей защитой области памяти оператором CLEAR.

Программу таймера можно включать в текст программы на БЕЙСИКЕ с помощью операторов DATA с последующей пересылкой в требуемую область оперативной памяти оператором непосредственного доступа к памяти РОКЕ.

Программа запускается командой RUN. После ввода параметров в режиме диалога память, занимаемую программой БЕЙСИКА,

можно очистить с помощью команд CLEAR и NEW и использовать для размещения других программ. При этом оператор ввода-вывода приостанавливает работу таймера на время обмена с накопителем из магнитной ленте.

После останова ПЭВМ с помощью клавиши СТОП происходит потеря информации в ячейке 100. Это не только вызывает останов программы таймера, но и приводит к невозможности ее перезапуска. Однако работа данной программы возобновляется после восстановления содержимого ячейки 100.

Длительная эксплуатация приведенных программно-аппаратных решений в системах управления измерительными комплексами показала их высокую надежность и эффективность. Снижение производительности ПЭВМ, вызванное обработкой прерываний, при частоте генератора, равной 100 Гц, составило приблизительно 1,5 %.

Телефон 585-29-84, Ленинград

ЛИТЕРАТУРА

1. Косенков С. М., Полосин А. Н., Счепиский З. А., Дябин М. И., Половянюк А. И. Бытовая персональная микроЭВМ «Электроника БК-0010 // Микропроцессорные средства и системы 1985.— № 1.— С. 22—25.
2. Обновленский П. А., Рудометов Е. А., Фокин А. Л., Харазов В. Г., Капустина Н. В. Применение микроЭВМ «Электроника БК-0010» в системе управления производством кварцевого стекла. // Микропроцессорные средства и системы.— 1986.— № 2.— С. 58—60.

Статья поступила 12.01.89

УДК 681.3.06

А. А. Задеба

КОМАНДА RST В ПРОЦЕДУРАХ ВЫЗОВА ПОДПРОГРАММ

Программное обеспечение программируемых контроллеров (ПК), применяемых для управления различными технологическими процессами, создается в виде совокупности отдельных подпрограмм, инициализируемых командами обработки сигналов, поступающими от объекта управления, таймеров и т. д. Процесс управления любым объектом в реальном масштабе времени сводится, как правило, к двум задачам: первая — измерение временных интервалов, запись или стирание признаков (флагов) и подсчет наступивших событий; вторая — численная или логическая интерпретация заданного алгоритма управления объектом, формирование необходимой совокупности управляющих сигналов и выдача их в соответствующие интерфейсные устройства.

Временные интервалы в ПК измеряются схемным, схемно-программным и программным способами. В существующих контроллерах

чаще всего используются схемно-программные методы. Например, в ПК, построенных на основе МПК БИС К580, применяется программируемый таймер КР580ВИ53, управляемый программным путем, а сигналы с него обрабатываются, как прерывания от внешних устройств [1].

Если используется программный таймер, то сигналы от генератора временных интервалов заводятся непосредственно на схему обработки запросов на прерывание, которая в свою очередь в соответствии с установленным приоритетом инициализирует подпрограмму счетчика. В этом случае схема ПК несколько упрощается, но усложняется ее программное обеспечение. Для управления системой со множеством датчиков, срабатывание которых необходимо подсчитывать, также требуются программные счетчики.

События в процессе функционирования системы фиксируются запоминанием или стиранием соответствующих флагов. Подпрограммы записи (стирания) флагов и подпрограммы программных счетчиков должны быть универсальными и инициализироваться с соответствующими идентификаторами номеров флагов или счетчиков. Если флаги и счетчики встречаются в основной программе больше одного раза, то целесообразно инициализировать эти подпрограммы с помощью процедур вызова подпрограмм, например команды CALL, в ПК на основе МП КР580ИК80А.

Перед процедурой вызова подпрограммы необходимо установить значение идентификатора номера флага или счетчика, а для подпрограммы счетчика — адрес точки возврата из подпрограммы в случае достижения заданного числа заполнения счетчика.

Вызов подпрограммы счетчика состоит из следующих команд:

Команда	Операнд	Комментарий
MVIB data		Запись в регистр В номера счетчика (значение идентификатора)
MVIC data		Запись в регистр С смещения, необходимого для формирования адреса точки возврата
MVID data		Запись в регистр заданного значения счетчика
CALL data		Вызов подпрограммы счетчика

Если в процессе составления подпрограммы необходимо многократно вызывать одни и те же подпрограммы, то использование процедуры вызова, состоящей из четырех команд и занимающей 9 байт памяти, не эффективно.

В ПК, где в качестве схемы обработки сигналов на прерывание применяется БИС КР580ВИ59 или другая, в которой не используется команда RST (повторный старт), ее

можно задействовать в качестве команды инициализации подпрограммы.

Эта команда по своему действию аналогична команде CALL, но отсылает только к восьми адресам, т. е. можно организовать вызов восьми подпрограмм. Поскольку адресуемые командой RST ячейки отстоят друг от друга на восемь адресов и подпрограмма целиком не уместится в этом интервале, то к остальной части подпрограммы можно перейти с помощью команды безусловного перехода JMP.

Для реализации такого метода необходимо выполнить два условия:

начало соответствующих подпрограмм разместить в начале адресного пространства, так как

Таблица 1

Адреса хранения флагов

1000 1001	Младший байт—20 Старший байт—FF	Адреса хранения флагов 1...8;
1002 1003	Младший байт—21 Старший байт—FF	Адреса хранения флагов 9...16;

Таблица 2

Идентификация программных счетчиков

OF00 OF01	Младший байт—01 Старший байт—FF	Адреса хранения счетчика 1
OF02	Код предельного значения заполнения счетчика 1.	
OF03 OF04	Младший байт Старший байт	Коды смещения для формирования адреса точки перехода при заполнении счетчика 1
OF05 OF06	Младший байт—03 Старший байт—FF	Адреса хранения счетчика 2

Таблица 3

Области ОЗУ для хранения промежуточных результатов счета

FF01	Область хранения значения счетчиков						
FF02	Счетчик 1						
	Счетчик 2						
	Область хранения флагов, записанных в позиционном коде						

	Флаги							
FF20	8	7	6	5	4	3	2	1
FF21	16	15	14	13	12	11	10	9

команда RST отсылает к восьми адресам: 0000H, 0008H, 0010H...0038H.

в программе должны быть таблицы соответствия номеров флагов и счетчиков адресам их размещения в ОЗУ (табл. 1—3), а также предельные значения заполнения счетчиков и коды смещения для формирования адресов точек возврата из подпрограмм. Для вызова подпрограммы с помощью команды CALL необходимы девять байтов памяти, а с помощью команды RST — три байта.

Затраты памяти для организации программных счетчиков в зависимости от их числа, количества процедур вызова и применения в них соответственно команд CALL и RST показаны в табл. 4.

Таблица 4

Вызов подпрограмм

Число программных счетчиков	Команда CALL			Команда RST			
	Длина подпрограмм, байт	Длина процедуры вызова, байт	Объем памяти, байт	Длина подпрограмм, байт	Длина процедуры вызова, байт	Объем таблицы, байт	Объем памяти, байт
Однократный							
1	23	9	32	31	3	5	39
10	23	90	113	31	30	50	111
20	23	180	203	31	60	100	191
30	23	270	293	31	90	150	271
40	23	360	383	31	120	0	351
50	23	473	473	31	150	250	431
Двукратный							
1	23	18	41	31	6	5	42
10	23	180	203	31	60	50	141
20	23	360	383	31	120	100	251
30	23	540	563	31	180	150	361
40	23	720	743	31	240	200	471
50	23	900	923	31	300	250	581

Применение команды RST в процедуре инициализации подпрограммы наиболее эффективно при большом числе идентичных подпрограмм и многократном их вызове.

Телефон 365-02-26, Москва

ЛИТЕРАТУРА

1. Торгов Ю. И. Программируемый таймер КР580ВИ53 и его применение // Микропроцессорные средства и системы. — 1984. — № 1. — С. 77—84.
2. Хвоц С. Т., Варлинский Н. Н., Попов Е. А. Микропроцессоры и микроЭВМ в системах автоматического управления. Ленинград: Машиностроение, 1987.
3. Давыдов Н. П. Система управления технологическими процессами на основе БИС серии КР580 // Микропроцессорные средства и системы. — 1984. — № 3. — С. 72—78.

Статья поступила 15.04.88

УДК 681.327.664.4

М. М. Владимирский

ТЕСТИРОВАНИЕ ОЗУ С ПОМОЩЬЮ ПСЕВДОСЛУЧАЙНОЙ ДВОИЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Псевдослучайные двоичные последовательности (ПСДП) широко используются для тестирования различных устройств из-за хороших статистических характеристик, жесткой регулярности, простоты генерации и проверки*.

Предлагаемый алгоритм формирования ПСДП в одноименных разрядах смежных ячеек ОЗУ обеспечивает независимость сложности его реализации от степени порождающего полинома ПСДП, хотя и усложняется с ростом числа его членов. Простейший порождающий полином ПСДП — $1+X^Q+X^P$ (наибольший известный автору цикл ПСДП при $Q=20$, $P=33$). Использование команд пересылки, сравнения и поразрядного исключаящего ИЛИ позволяет параллельно формировать и обрабатывать τ независимых ПСДП (τ — разрядность ячейки ОЗУ), что существенно сокращает время тестирования ОЗУ.

Программа исключает обращение к системному ОЗУ и предполагает назначение указателя стека оператором. Это усложняет программу, так как нельзя пользоваться командами прямой адресации, но обеспечивает ее выполнение вне зависимости от состояния системного ОЗУ основной программы. Также необходима предварительная загрузка первых R -ячеек тестируемой памяти опорной информацией из задаваемой оператором области ПЗУ для исключения попадания в вырожденный цикл ПСДП и (или) формирования двух или более ПСДП с нулевым циклическим сдвигом.

Программа включает в себя два этапа тестирования ОЗУ. На предварительном этапе ОЗУ ПСДП заполняется опорной информацией с последующей проверкой ее сохранности (выявляются возможные сообщения между разрядами слова ОЗУ и повреждениями в шине адреса, укорачивающими цикл проверки основного этапа). На основном этапе осуществляется N -кратное последовательное заполнение ОЗУ ПСДП с предварительной проверкой правильности ранее записанной информации в каждой его ячейке (число циклов N определяет вероятность необнаружения ошибки $P_N=2^{-N}$).

Алгоритм программы включает в себя следующие операции:

1. Ввод и установка SP
2. Ввод: N ; AO; AN; AK
3. $I:=0$
4. $(AO+I) \rightarrow (AK-I)$
5. $I:=I+1$
6. Если $P - I > 0$, то к 4

* Хоровиц И. П., Хилл У. Искусство схемотехники. — М.: Мир, 1983. — Т. 2.

7. $AI := AK - P$
8. $(AI + P) \oplus (AI + P - Q) \rightarrow (AI)$
9. $AI := AI - 1$
10. Если $AI < AN$, то к 8
11. $I := 0$
12. Если $(AO + I) = (AK - I)$, то к 14
13. Вывод: $AK - I$; $(AK - I)$; $(AO + I)$
14. $I := I + 1$
15. Если $P - I > 0$, то к 12
16. $AI := AK$
17. Если $(AI) = (AI - P) \oplus (AI - Q)$, то к 19
18. Вывод: AI ; (AI) ; $(AI - P) \oplus (AI - Q)$
19. $(AI + P) \oplus (AI + P - Q) \rightarrow (AI)$
20. $AI := AI - 1$
21. Если $AI \geq AN$, то к 17
22. $N := N - 1$
23. Если $N > 0$, то к 16
24. Конец

Примечания: АО — начальный адрес области ПЗУ, информация из которой выбирается в качестве опорной; АН(АК) — начальный (конечный) адрес тестируемой области ОЗУ; Р, Q — степени образующего полинома ПСДП; \oplus , \rightarrow — операции поразрядного исключающего ИЛИ и пересылки соответственно.

$$AX = \begin{cases} AX, & \text{если } AN \leq AX \leq AK \\ AX + AK - AN, & \text{если } AX < AN \\ AX - AK + AN, & \text{если } AX > AK \end{cases}$$

Предварительный этап включает в себя с первого по пятнадцатый операторы, основной — с шестнадцатого до конца. Время выполнения программы при заданной достоверности результата (N) пропорционально объему тестируемого ОЗУ в словах и не зависит от длины слова. ПСДП, используемые для тестирования, практически независимы, так как коэффициенты взаимной корреляции между различными разрядами ПСДП (за малым исключением) близки к нулю.

Последнее утверждение несправедливо для разрядов ПСДП, разнесенных на их циклический сдвиг. Определение вероятности, что циклический сдвиг между хотя бы двумя ПСДП, используемыми для тестирования, будет меньше числа слов ОЗУ, сводится к решению элементарной задачи нахождения вероятности, что при случайном делении окружности длиной L (аналог цикла ПСДП) на τ частей (аналог разрядности слова ОЗУ) ни одна из дуг не будет меньше величины M (аналог числа слов ОЗУ).

$$P_{\tau} \{ < M \} = 1 - (1 - \tau M / L)^{\tau - 1}$$

при естественном $\tau M < L$.

Программа, составленная по изложенному алгоритму, для процессора K580BM80 имеет следующие характеристики:

объем менее 180 байт при обращении к мониторным подпрограммам ввода-вывода информации, несовпадениях и сообщениях об окончании программы;

требуется не менее десяти ячеек ОЗУ для стека;

время однократной обработки одной ячейки ОЗУ около 100 мкс;

вероятность появления циклического сдвига хотя бы между двумя ПСДП менее 60 000 не превосходит $0,4 \cdot 10^{-3}$.

Следует отметить, что при незагруженном системном ОЗУ и процессорах с большим числом регистров характеристики программы существенно улучшаются.

196128, Ленинград, Варшавская, 11, ЛОНИИС;
тел. 297-75-14

Статья поступила 28.02.89

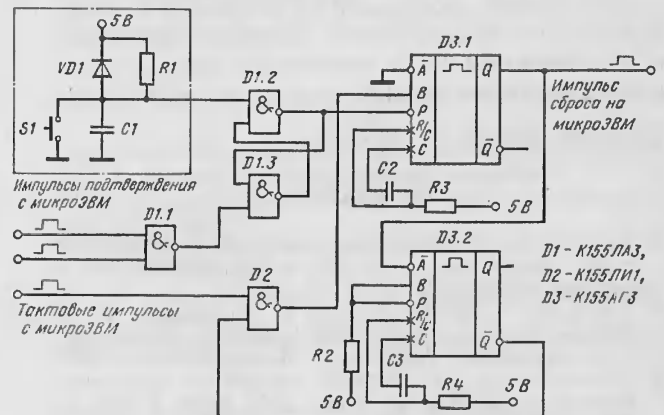
УДК 681.325

В. Д. Савелов

НАДЕЖНОЕ УСТРОЙСТВО ДЛЯ НАЧАЛЬНОГО ЗАПУСКА МИКРОПРОЦЕССОРА

Проблема надежного начального запуска микропроцессорной системы в промышленных условиях очень актуальна, так как при сбоях сетевого электропитания происходят не только значительные отклонения напряжения от номинального значения, но и его кратковременное пропадание.

Обычно микропроцессор запускается по сигналу, подаваемому на вход начальной установки (сброса). По этому сигналу в программный счетчик записывается начальный адрес, с которого микропроцессор начинает выборку команд подпрограммы инициализации, по которой специальные регистры микропроцессора устанавливаются в исходное состояние и задаются режимы работы программируемых БИС интерфейсов ввода-вывода, прерывания, прямого доступа к памяти и т. д. Из-за переходных процессов в сети питания, вызванных, например, кратковременными провалами напряжения, наибольшее число сбоев программы приходится на начальные моменты времени после появления номинального значения напряжения сетевого питания. Подпрограмма инициализации наиболее подвержена сбоям.



Устройство для включения и перезапуска микропроцессора

Известные устройства начального запуска [1, 2] в отличие от предлагаемого устройства (см. рисунок) не позволяют в случае сбоя подпрограммы инициализации автоматически произвести повторный перезапуск микропроцессора.

Техническое решение описанного устройства Госкомитетом СССР по делам изобретений и открытий признано изобретением [3].

470014, Караганда, ОПКБ НПО «Черметавтоматика»; тел. 51-21-95

УДК 681.326

А. С. Баранов, В. Д. Безгура, Н. А. Перепелица

ЭМУЛЯТОР ПОСТОЯННЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ

Один из способов решения проблемы отладки программного и аппаратного обеспечения встраиваемых микропроцессорных модулей, работающих по программам, записанным в постоянные запоминающие устройства (ПЗУ), — применение эмулятора ПЗУ совместно с логическим анализатором.

Исходные требования при разработке эмулятора ПЗУ: возможность реконфигурации шины в пределах 8, 16, 24, 32 разрядов и настройки на различные комбинации нулей и единиц на входах выборки ИС, время выборки — не хуже 125 нс, выдача сигнала для запуска внешнего устройства при совпадении значения адреса на шине ПЗУ с заданным.

Эмулятор ПЗУ (см. рисунок) выполнен на основе двуportового быстродействующего ОЗУ (32 К), представляющего собой четыре линейки по восемь ИС КМ132РУ5А и ОЗУ признаков (8К×2). Адрес байта заносится в регистр адреса, а затем байт по шине данных записывается в ОЗУ (воз-

можно запись данных в ОЗУ 16-разрядными словами). Такой способ доступа медленнее прямой адресации по шине, однако при этом остается свободным все системное адресное пространство.

Строб-импульсы поступают из ОЗУ признаков при совпадении адресов ПЗУ с заданными адресами, запускают логический анализатор или осциллограф. С их помощью пользователь может остановить программу по заданным адресам (например, сбрасывая готовность процессора).

Пользователь может применять ИС ПЗУ с различным числом входов выборки (CS) и комбинаций нулей и единиц на этих входах.

Дешифратор выборки представляет собой восемь компараторов К531СП1, непрерывно сравнивающих коды на входах с кодами, заданными переключателями. Конфигурация переключек определяет нужную комбинацию нулей и единиц. На вход эмулятора подключается до восьми каналов по четыре линии

1. Алексенко А. Г. и др. Проектирование радиоэлектронной аппаратуры на микропроцессорах. Программирование, типовые решения, методы отладки. — М.: Радио и связь, 1984.
2. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы. — М.: Радио и связь, 1981.
3. А. С. № 1488807 СССР, МКИ G 06 F 11/22 Устройство для включения и перезапуска микропроцессора / В. Д. Савелов — Опубл. 1989. Бюл. № 23.

Статья поступила 16.06.89

CS1...CS4 (принято, что у микросхемы ПЗУ не более четырех входов выборки). Таким образом, возможна эмуляция восьми ИС ПЗУ.

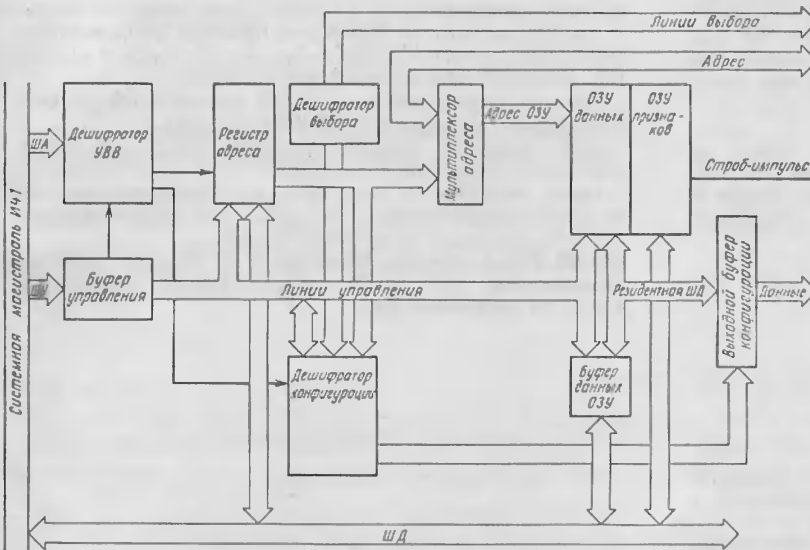
Предполагается, что ПЗУ имеют байтовую организацию, поэтому при 8-разрядной ИС выбираются последовательно, 16-разрядной — одновременно две ИС, 24-разрядной — три, 32-разрядной — четыре.

Дешифратор конфигурации программируется в соответствии с архитектурой ПЗУ эмулируемого объекта, типом и емкостью используемых ПЗУ. В режиме эмуляции он вырабатывает сигналы выбора линейки ОЗУ данных и признаков, открывания выходных буферов данных, формирует недостающие адресные разряды. Его работа может быть проиллюстрирована следующим примером: емкость ИС ПЗУ пользователя 1024 байт, следовательно, от платы пользователя приходят на эмулятор десять адресных линий; группы линий выборки CS переходят в активное или пассивное состояние через границы 1024-байт адресных массивов. Емкость одной линейки ОЗУ эмулятора (4 Кбайт) больше 1024 байт, следовательно, все эмулируемое ПЗУ пользователя может храниться в одной линейке. Дешифратор конфигурации ПЗУ в этом случае программируется так, чтобы при активных состояниях любой линии CS пользователя генерировать сигнал выбора первой линейки ОЗУ и адресные сигналы на двух адресных линиях. Таким образом, дешифратор конфигурации ПЗУ обеспечивает непрерывность массива данных, записанных в ОЗУ эмулятора; он выполнен на основе быстродействующего ОЗУ КМ185PV8Б.

Разработано следующее ПО: кросс-ассемблеры для КМ1810ВМ86, КР181БВЕ51, КР1816ВЕ48, специализированный кроссассемблер для разрядно-модульных систем AMDASM, программа-загрузчик объектных файлов в ОЗУ эмулятора. Все программы работают в ОС МИКРОДОС.

Для отладки систем на основе разрядно-модульных процессоров с шириной микрослова более 32 бит эмуляторы могут подключаться параллельно. Конструктивно эмулятор выполнен в виде двух печатных плат, подключаемых к шине, логически совместимой с ИЧ1.

Телефон 34-03-97, Запорожье



Структурная схема эмулятора ПЗУ:

ША — шина адреса, ШД — шина данных, ШУ — шина управления

НОВЫЕ ВЫСОКОСКОРОСТНЫЕ МОДЕМЫ ДЛЯ IBM СОВМЕСТИМЫХ МАШИН ЗАПОЛНЯЮТ ПРОБЕЛ НА ОТЕЧЕСТВЕННОМ РЫНКЕ

Разработано два варианта встраиваемого модема: «С-9600 М1», «С-9600 М2» для IBM PC XT/AT и для ДВК-2/3/4 соответственно. Оба модема поддерживают одинаковый протокол обмена, позволяющий подстраиваться под состояние телефонной линии.

Модемы выполнены в виде полуплат и вставляются в свободные разъемы расширений компьютеров:

прямо включаются в телефонную линию;
реализован автовызов и автоответ;
максимальная скорость работы 9600 бит/с в условиях отличного состояния местной или выделенной телефонной линии;

средняя скорость (в Москве, а также по междугородным линиям) составляет 2400 бит/с;

протокол обмена и коррекции ошибок обеспечивают 100 % достоверность передачи информации;

поставляются с программным пакетом «Комплинк» для IBM PC XT/AT или «Компмодем» для ДВК-2/3/4; пакет программ содержит перестраиваемую базу данных, совместимую с dBase III+.

Гарантийное обслуживание 3 года.

Как известно, телефонные линии стран — членов СЭВ не соответствуют международным стандартам, что не приводит к трудностям при выборе модема для Ваших целей. Рекомендованное устройство позволяет обеспечивать не только достоверность передачи, но и ее высокую скорость, удовлетворяет требованиям самого взыскательного пользователя.

Модемы можно приобрести по адресам:

Москва, Нахимовский проспект 30/43, магазин — салон «Приборы и вычислительная техника», тел. 125-09-36, 124-20-42; Ленинград, Ленинградский проспект 148, магазин — салон, тел. 290-55-34.

Подробные справки по телефону в Москве: 353-68-58 (с 14 до 19 ч).

ВАМ НЕ СЛЕДУЕТ ТРАТИТЬСЯ НА ПОКУПКУ ЛЕНТ ДЛЯ ПРИНТЕРОВ!

МАКЛИНКЕР — автоматическое устройство для прокраски лент позволит быстро и эффективно восстановить качество печати вашего принтера. МАКЛИНКЕР в 60—100 раз продлит срок эксплуатации ленты (именно столько раз можно обновлять ее красящий слой, прежде чем лента придет в негодность).

МАКЛИНКЕР укомплектовывается баллоном тонера на 500 прокрасов. Стоимость комплекта — 1000 рублей. Поставщик обязуется производить допоставку тонера в необходимых количествах по льготным расценкам. Телефон: 353-68-58, Москва

ЖДЕМ ВАШИХ ЗАЯВОК!

РЕМОНТ ИМПОРТНОЙ ВЫЧИСЛИТЕЛЬНОЙ И МНОЖИТЕЛЬНОЙ ТЕХНИКИ

Заправка тонером картриджей ксероксов и лазерных принтеров. Перепрокраска лент матричных принтеров в черный и другие цвета.

Поставка ленты для всех типов принтеров. Все услуги по льготным ценам, исполнение на месте. Оплата наличными в рублях или в валюте.

Телефон: 353-68-58, Москва

ТЕХНОЛОГИЧЕСКИЙ МОДУЛЬ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ (ТМООП) ДЛЯ ПЭВМ

Технологический модуль объектно-ориентированного программирования (ТМООП) предназначен для создания в объектно-ориентированной среде прототипов программ, разработки небольших промышленных программ и для практического ознакомления широкого круга пользователей с объектно-ориентированным программированием.

ТМООП предоставляет следующие возможности.

ИНТЕРФЕЙС С ПОЛЬЗОВАТЕЛЕМ:

перекрывающиеся окна на экране дисплея, иерархические выставляемые меню, цветной или черно-белый режим, управление курсором с помощью мыши, программируемая функциональная клавиатура, средства изучения системы.

ЯЗЫК ПРОГРАММИРОВАНИЯ:

близкий к естественному русскому языку программирования типа Смолток, богатый набор встроенных средств, абстрактные типы данных (классы), скрытие реализации, наследование свойств и поведения объектов, динамическое связывание имени со значением.

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММ:

формулировка задачи на языке взаимодействия объектов, выполняемая спецификация программ, высокая модульность, отладка на уровне исходного языка, адаптация к нуждам и вкусам пользователя, надежность внесения изменений.

Документация на ТМООП состоит из четырех частей: «Описание применения» — многооконный интерфейс, меню и работа с клавиатурой и мышью, правила запуска и эксплуатации ТМООП; «Описание языка» — синтаксис и семантика исходного языка.

«Руководство пользователя» — наиболее употребительные классы и примеры написания реальных программ. «Описание базовых классов» — справочное руководство по всем классам.

ТМООП работает на ПЭВМ типа IBM PC AT/XT с памятью 640 Кбайт, цветным или черно-белым монитором и мышью (по желанию) под управлением MS DOS.

Программные средства ТМООП поставляются на двух 5-дюймовых дискетах по 360 Кбайт каждая.

Для генерации ТМООП требуется около 1,5 Мбайт свободной памяти на винчестере.

Цена ТМООП — 5 тыс. руб. с сопровождением со стороны разработчика, 3 тыс. руб. — без сопровождения.

Адрес предприятия — разработчика:

117900, ГСП-1, Москва, Вавилова 30/6, Институт проблем информатики АН СССР, тел. 135-30-17 [дирекция], 930-05-16 [разработчики]

УДК 621.3.049.77:681.326.3

Гальперин М. П., Блох Е. М., Шляхтин В. В. **Однокристальные дисплейные контроллеры** // Микропроцессорные средства и системы.— 1990.— № 4.— С. 2.

Дана характеристика однокристальных дисплейных контроллеров: КР580ВГ75, КМ1809ВГ6, К1809ВГ3, К1809ВГ4, сделан их сравнительный анализ, указаны некоторые особенности применения. Сформулированы требования к современному дисплейному контроллеру.

УДК 681.326.3:681.327.2

Блох Е. М., Бодашков К. Б. **Однокристалльный контроллер дисплея и клавиатуры** // Микропроцессорные средства и системы.— 1990.— № 4.— С. 5.

Описана БИС К1809ВГ3, предназначенная для построения дисплеев на ЭЛТ, обеспечивающая воспроизведение алфавитно-цифровой и графической информации на экранах монохромных и цветных мониторов, а также ввод данных с клавиатуры.

УДК 681.32.06

Домарацкий С. Н., Шраго И. Л. **Организация программной среды системы автоматизации исследований на базе ПЭВМ** // Микропроцессорные средства и системы.— 1990.— № 4.— С. 21.

Рассмотрены методология и конкретные результаты разработки программной среды систем автоматизации научных исследований на базе ПЭВМ «Искра 1030», базирующейся на супервизоре пакетов с иерархической структурой. Иерархические интерактивные пакеты создаются из готовых программ (EXE-файлов), разработанных на любом языке в разное время. Вносить изменения в программы не требуется.

УДК 681.3.06

Дедков А. Ф., Наумов Е. В., Шерс А. Л. **Средства работы с базами данных в языке Пролог** // Микропроцессорные средства и системы.— 1990.— № 4.— С. 33.

Рассматривается расширенная версия интерпретатора языка логического программирования Пролог для ПЭВМ Роботрон 1715, которая обеспечивает непосредственную работу пользователей с файлами СУБД dBASE-II и дает практическую возможность использования Пролога как языка запросов реляционных и средства построения дедуктивных баз данных.

УДК 681.3.06:321

Руденко Ю. М., В. Г. Жиганов, А. Я. Мосин. **Система тестирования и отладки ОЭВМ серии К1816** // Микропроцессорные средства и системы.— 1990.— № 4.— С. 55.

Рассмотрены технические и программные средства тестирования ОЭВМ и отладки программ непосредственно на кристалле, представленных в виде абсолютных кодов. Описаны структурная схема отладочного устройства и алгоритм тестирования команд.

УДК 681.32

Герман Г. Б., Сачук А. А., Улыбин А. А. **Сопряжение микроЭВМ «Электроника 60» с микропроцессорными устройствами** // Микропроцессорные средства и системы.— 1990.— № 3.— С. 77.

Описано устройство сопряжения микроЭВМ «Электроника 60» с внешними микропроцессорными устройствами. Представлены структурная схема устройства сопряжения и принципиальные схемы отдельных узлов. Приведены результаты испытаний описываемого устройства сопряжения в комплексе.

UDC 621.3.049.77:681.326.3

Galperin M. P., Blokh E. M., Shlyahutin V. V. **Single-chip display controllers** // Microprocessor devices and systems.— 1990.— N 4.— P. 2.

General description, comparison and application hints of single-chip display controllers КР580ВГ75, КМ1809ВГ6, К1809ВГ3, К1809ВГ4 is given. The specifications for the state-of-art display controller are discussed.

UDC 681.326.3:681.327.2

Blokh E. M., Bodashkov K. B. **A single-chip display and keyboard controller.** // Microprocessor devices and system.— 1990.— N 4.— P. 5.

The К1809ВГ3 LSI for CRT displays is described. The IC displays alphanumeric and graphic information on monochrome and color CRT monitors and also supports data input from keyboard.

UDC 681.32.06

Domaratsky S. N., Shrago I. L. **The software environment structure of the personal scientific research automation system on PC** // Microprocessor devices and systems.— 1990.— N 4.— P. 21.

The methods and practical results of interactive program environment project for scientific automation on ISKRA-1030 PC is discussed. The concept is based on the usage of batch supervisor with hierarchic structure. The multi-level interactive program packages are built out of ready. EXE files previously developed in any language. No changes of source code are necessary.

UDC 681.3.06

Dedkov A. F., Naumov E. V., Shers A. L. **Data-base support in the PROLOG programming language** // Microprocessor devices and systems.— 1990.— N 4.— P. 33.

The extended version of PROLOG interpreter running on ROBOTON 1715 PC is discussed, which enables direct access to dBASE-II file structure in user programs and permits practical application of PROLOG as a query language for relation databases and as a tool for building deductive databases.

UDC 681.306:321

Rudenko Yu. M., Zhiganov V. G., Mosin A. Ya. **A test and development system for K1816 /8048 single-ship computer** // Microprocessor devices and systems.— 1990.— N 4.— P. 55.

The hardware and software tools for microprocessor testing and on-chip program development on the instruction code level are discussed. The structure of the debug unit and instruction test algorithm are described.

UDC 681.32

German G. B., Sachuk A. A., Ulybin A. A. **Interfacing "Electronika 60" to microprocessor devices.** // Microprocessor devices and systems.— 1990.— N 3.— P. 77.

The interface adapter for connecting "Electronika 60" to external microprocessor devices is described. The skeletal diagram of the unit is shown and description of specific blocks is given. The results of adapter test in the system are given.

УДК 681.327

Бири Ш., Ефремов А. А., Молнар Й. Интерфейс для управления и автоматического измерения на основе ПК типа IBM PC XT/AT // Микропроцессорные средства и системы.— 1990.— № 4.— С. 79.

Описан интерфейс и программа его обслуживания для автоматизации измерений протяженных магнитных полей сложной конфигурации, обработки и документирования данных измерений. Интерфейс осуществляет связь между компьютером IBM PC XT/AT и измерительным прибором, имеет 8-разрядный вход для приема внешних сигналов и 8-разрядный мощный выход. Он построен в стандарте IBM PC XT/AT и содержит 22 микросхемы серии 74LS или K555.

УДК 681.326.3

Комиссаров Е. В., Кулинич П. А., Сидоркин В. В. Интерфейс НМЛ CM 5309 для ПК IBM / Микропроцессорные средства и системы.— 1990.— № 4.— С. 84.

Описывается интерфейс накопителя на магнитной ленте типа CM5309 (CM5306 или CM5308) к ПК совместимому с IBM PC XT/AT. Он подключается через форматтер ИСОТ 5004 С, поставляемый вместе с накопителем, и позволяет реализовать набор стандартных операций на НМЛ с плотностью записи — чтения данных 800 и 1600 Врi. Информация передается по каналу прямого доступа к памяти процессора. Интерфейс выполнен на одной плате конструктива ПЭВМ и вставляется в свободный разъем расширения. Программное обеспечение разработано в виде пакета подпрограмм нижнего уровня, тестовых программ и диалоговой системы ведения архивов. Подпрограммы для IBM PC вызываются из программ на языках Фортран-77 и Си.

УДК 687.327

Biri Sh., Efremov A. A., Molnar J. Data acquisition and control card for the IBM PC XT/AT // Microprocessor devices and systems.— 1990.— N. 4.— P. 79.

The interface card and software support for complex configuration contiguous magnetic field measurement, data processing and documentation are described. The interface performs 8-bit data transfer between IBM PC and external instrument, and has 8-bit input and 8-bit power output ports. The standard IBM PC slot card is built using 22 TTL ICs.

УДК 681.3

Komissarov E. V., Kulnich P. A., Sidorkin V. V. The magnetic tape drive CM5903 interface for IBM PC // Microprocessor devices and systems.— 1990.— N. 4.— P. 84.

The IBM PC interface card for 1/2" tape drives type CM5306, CM5308 and CM5309 is described. The tape drive is connected via data formatter type ISOT 5004C, supplied with tape drive, and performs standard I/O operations at 800 and 1600 BPS. The single XT-slot card is installed in any empty slot and uses DMA transfer. The software support includes basic low-level I/O support, test programs and dialogue archive utility. The BIOS tape operations are called from Fortran-77 and C programs.

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

Извещаем, что с 1991 года наш журнал (индекс по каталогу Союзпечати «Газеты и журналы» 70588) будет выходить под новым названием — «СРЕДСТВА И СИСТЕМЫ ИНФОРМАТИКИ». Изменится и цена одного номера — теперь он будет стоить 2 руб. 20 коп. Что касается тематики журнала, то она останется прежней.

**НЕ ЗАБУДЬТЕ СВОЕВРЕМЕННО И ТОЧНО
ОФОРМИТЬ ПОДПИСКУ**

РЕКЛАМА

ПО «Орловский завод управляющих вычислительных машин им. К. Н. Руднева» заканчивает разработку и приступает к освоению интегрированной комплексной системы автоматизации рабочих мест (ИКАР).

Область ее применения — от учебных классов до систем массового обслуживания, от простейших АСУ технологическими процессами в сельском хозяйстве до АСУ ТП сложных химических производств, от конторских АРМ до САПР проектирования БИС.

В состав системы ИКАР входят:
8-разрядная универсальная ЭВМ;
две 16-разрядных — для управления и построения АРМ;
три типа УСО-МСКУ, МКСО, С601М;
сервисное оборудование для ремонта и метрологической проверки электронных блоков.

В системе ИКАР используются:
единая операционная система МДСС, совместимая с MS DOS 3.3;

унифицированные аппаратные средства;
полная переносимость программного обеспечения, разработанного ранее для линии УВМ РС;
выносное УСО трех типов, позволяющих автоматизировать практически любые производства;

сетевые средства, поддерживаемые аппаратно и программно;
программные продукты, поставляемые по заказу пользователя сервисного оборудования;

Адрес: 302025, г. Орел, т. а. Нейрон 148216 и 148234;
Телефон: 3-06-02.

Заместитель главного редактора С. М. Пеленов

Номер подготовили:

Е. И. Бабич, Г. Г. Глушкова,
Корректор Т. Ф. Ершова
Технический редактор Г. И. Колосова
Адрес редакции журнала:
103051, Москва, Малый Сухаревский пер., д. 9А
Телефон: 208-73-23, 208-19-94

Сдано в набор 01.07.90. Подписано к печати 13.08.90.
Формат 84×108 1/16. Офсетная печать. Усл. печ. л. 10,08.
Уч.-изд. л. 14,3. Тираж 87 300 экз. Заказ № 6087.
Цена 2 р. 20 к.

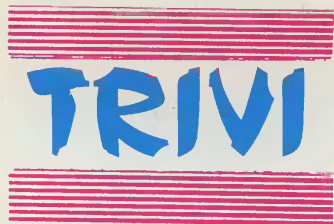
Орган Государственного комитета СССР
по вычислительной технике и информатике

Текст набран в Ордена Трудового Красного Знамени
Чеховском полиграфическом комбинате
Государственного комитета СССР
по печати
142300, г. Чехов Московской обл.

Изготовлен в Московской типографии № 13
ПО «Периодика» Государственного комитета СССР
по печати
107005, г. Москва, Денисовский пер., 30.
Заказ 209.

МЕЖОТРАСЛЕВОЕ ГОСУДАРСТВЕННОЕ ОБЪЕДИНЕНИЕ "МАШПРИБОРСЕРВИС"

по комплексному централизованному обслуживанию сложного технологического оборудования с применением микропроцессорной техники.



МГО-это предприятия и участки в 40 городах по всей территории СССР, свыше 4,5 тысяч квалифицированных специалистов и высококачественное обслуживание электронного и электро-механического оборудования.

Телефон московского филиала - 210-09-87

ПОСЛЕДОВАТЕЛЬНО ПЕЧАТАЮЩИЙ МЕХАНИЗМ КОНСУЛ 2112-10 ЕС 7934-02

Предназначен для непрерывной печати алфавитно-цифровых данных на бумагу по электрическим сигналам с УУ. Используется в системе отображения данных ЕС 7920, а также в ЭВМ и устройствах обработки данных 3 и 3,5 поколения.

Скорость печати - 160 зн/с
Число знаков в строке - 132
Код - КОИ-8
Растр - 9x9

ПОСЛЕДОВАТЕЛЬНО ПЕЧАТАЮЩИЕ УСТРОЙСТВА КОНСУЛ 212-10, КОНСУЛ 212-20, КОНСУЛ 212-30

Предназначены для печати алфавитно-цифровых и графических данных в устройствах включенных в ЕС ЭВМ и СМ ЭВМ, ИВМ РС и их аналогами.

Интерфейсы: параллельный ППУ Консул 2111 и Консул 2112 (можно выбирать также интерфейс, совместимый с Центроникс), параллельный ИРПП, последовательные V 24 и ИРПС

Скорость печати - 160 зн/с
Число знаков в строке - 132
Код - КОИ 7, КОИ 8, ДКОИ
Основной растр - 11x9

КОНСУЛ 2111 и КОНСУЛ 2112 - печать в двух направлениях.

КОНСУЛ 212-30 - односторонняя двухпроходная печать при повышенном качестве и графике (144x144 точек на дюйм)

ПОСЛЕДОВАТЕЛЬНО ПЕЧАТАЮЩЕЕ УСТРОЙСТВО КОНСУЛ 2011 (СМ 6339)

ППУ предназначены для вывода алфавитно-цифровой и графической информации в устройствах, работающих с СМ ЭВМ, ИВМ РС и их аналогами

Интерфейсы: параллельный ИРПП-М (Центроникс), последовательные С 2 (V 24) и ИРПС

Скорость печати - 160 зн/с

Растр - основной 9x9

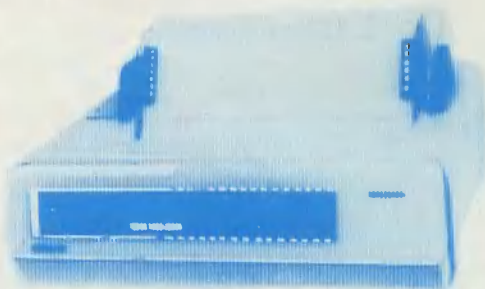
Число знаков в строке - 80

Код - КОИ 7, КОИ 8

Печать знаков в двух направлениях, графической информации в одном направлении.

Срок поставки: 1990 г. - по желанию заказчика

КОНСУЛ 212-10



**EXPORT
IMPORT
KOVO**
PRAHA
CZECHOSLOVAKIA

Внешнеторговое объединение:

KOVO
ул. Янковцова № 2
170 88 г. Прага
ЧСФР
телефон: 874 11 11
телекс: 121 481

Торговое представительство

ЧСФР Москва СССР
ул. Юлиуса Фучика 17/19
телефон: 250 20 60
телекс: 414 480



НПО «ВЕРТЕР» ПРЕДЛАГАЕТ:

1. ПТК «РС-7920», позволяющий осуществлять обмен информацией между персональными ЭВМ РС XT/AT/AT386/PS-2, ЕС1840/41, Нейрон и ЕС ЭВМ, а также использовать ПЭВМ в качестве терминала, совместимого с ЕС7920 в режиме полного экрана.

Плата адаптера устанавливается в один из свободных слотов ПЭВМ и через коаксиальный разъем подключается к устройству управления ЕС7922/ЕС7921 аналогично устройству ЕС7927.

Программное обеспечение комплекса:

- эмуляция терминала ЕС7927 со скоростью 720 кбит/с;
- передача файлов в среде VM(CBM), PRIMUS, TSO (скорость — 120 кбит/с);
- эмуляция дополнительного «винчестера» на дисках ЕС ЭВМ в среде VM(CBM), PRIMUS;
- программный интерфейс.

2. ПТК «ДВК 7920» для подсоединения ДВК2(3) и ЕС ЭВМ через ЕС7920. Возможности сходны с ПТК «РС-7920».

3. ПТК «СМ 7920», позволяющий осуществлять высокоскоростной обмен файлами между СМ4, СМ1420, СМ1700 и ЕМ ЭВМ через ЕС7920. Операционные системы ЕС ЭВМ: CBM, VM/SP, TKS, MVS; СМ ЭВМ: RSX, VMS.

4. ПТК «ПЭВМ 7970», позволяющий осуществлять полноэкранный диалог и передачу файлов при подключении ПЭВМ типа IBM PC, ЕС1840, ДВК, «Нейрон» к ЕС ЭВМ через ЕС7970.

5. Для микропроцессорных плат «Комета»: новое ПО для VM (CBM), по возможностям и характеристикам схоже с «РС-7920».

6. ПОСТАВЛЯЕТ:

- платы с 4-я и 8-ю разъемами RS232C (с комплектом ПО для работы плат в простейших сетях типа «звезда», ПО для ОС XENIX-286(386);
- одно- и четырехканальные устройства последовательного обмена (скорость до 19200 бит/с) по стыку ИРПС для IBM PC;
- платы BSC для IBM PC в комплекте с ПО;
- платы параллельного порта «CENTRONICS» для IBM PC, устанавливаются вместо или дополнительно к стандартной плате;
- модемы (на микропроцессорной базе);
- платы ОЗУ 256К;
- устройство аналогового ввода-вывода ЦАП/АЦП для IBM PC.

7. Комплексная поставка локальных сетей:

- типа «звезда» по стыку RS232C. Подключается до 16 ЭВМ, скорость передачи данных — 9600 бит/с;
- древовидной структуры с децентрализованным управлением, на основе моноканала, скорость передачи данных — 2 Мбит/с, максимальное расстояние — 750 м, для ЭВМ с шинами MULTIBUS, ОШ, МПИ.
- типа ETHERNET со скоростью передачи данных 10 Мбит/с;
- типа ArcNET;

8. Подключение к РС через параллельный интерфейс: ЕС 5012/5017/5025, СМ 5300, АЦПУ ЕС 7032/7036/7038/7040, АЦПУ СМ С315, DZM-180 и др.

Дополнительную информацию о поставляемой продукции, ее стоимости и сроках поставки можно получить, направив запрос с указанием Ваших реквизитов и телефонов по адресу: 103009, Москва, а/я 160.

Телефон: 297-55-82.