

Содержание

Главный редактор

Сергей Бирюков

Зам. главного редактора

Сергей Кузнецов

Редакционная коллегия

Павел Асташкевич
Александр Фрунзе
Виктор Йовчик
Юлия Герасимова

Дизайн и верстка

Ирина Галкина
Ирина Чикина

Отдел распространения

(095) 777-12-15
e-mail: sales@dian.ru
Юрий Царев
Сергей Лукин

Отдел рекламы

Юлия Суханова

Адрес редакции:

127015 Москва, ул. Бутырская, д. 41/47
«ИД Скимен»
тел./факс: (095) 777-12-15
www.dian.ru
e-mail: editor@dian.ru

Издатель и учредитель
ООО «ИД Скимен»

Отпечатано в ГУП «Чеховский
полиграфический комбинат»

Тираж 5 300 экз.
Заказ № 2971.

Журнал зарегистрирован
в Министерстве РФ по делам печати,
телерадиовещания и средств
массовых коммуникаций
Reg. № ПИ77-5262

Редакция не несет ответственности
за информацию, приведенную
в рекламных материалах

За содержание статьи
и ее оригинальность несет
ответственность автор

Полное или частичное
воспроизведение материалов
допускается только с разрешения
ООО «ИД Скимен»

Информацию о подписке
см. на последней странице журнала

КОМПОНЕНТЫ

- С. Бирюков.** Сетевые источники питания на микросхемах LinkSwitch-TN 2
- Автомат управления насосом на базе микроконтроллера HT48R05A-1 4
- О. Николайчук.** Анализ SFR-совместимости микроконтроллеров фирмы Silabs 7
- В. Зотов.** Применение ПЛИС семейства CoolRunner-II для реализации контроллера интерфейса I²C 10
- С. Алексеев.** Операционные усилители фирмы ROHM 14
- О. Вальпа.** Цифровые сигнальные процессоры 17

ИСКУССТВО СХЕМОТЕХНИКИ

- С. Лоцицкий.** Методы анализа операционных схем в частотной области 20
- Ю. Дмитриев.** Априорная оценка нелинейных искажений при проектировании усилителей на биполярных транзисторах 23
- А. Фрунзе, А. Фрунзе.** Микроконтроллеры? Это же просто! 26
- Б. Шевкопляс.** Скремблирование передаваемых данных 30
- Г. Волович, Е. Савченко.** Повышение точности измерительного трансформатора тока 34
- Г. Петин.** Генератор с шунтирующим диодом 35

СОФТ

- Г. Кардашев.** Компьютерное схемотехническое моделирование электронных устройств 36
- М. Бырдин.** Пиктограммы в области System Tray 38
- О. Вальпа.** Borland C++ Builder 6 для начинающих 40

ПРАКТИКА

- С. Белов.** Компьютер — измерительный комплекс 43
- И. Новоселов.** Автомобильный мультиметр-часы на PIC16F873 46
- В. Чулков.** Декодер телевизионного стереозвукового сопровождения формата NICAM NM2905 48
- Д. Онышко.** Коммутатор телевизионных сигналов 51
- С. Малахов.** Музыкальный звонок 52
- О. Вальпа.** Дополнительный порт LPT 54

CD-ROM с журналами «Схемотехника» (с. 3).

Семинары на DSPA-2005 (с. 6). Вниманию читателей (с. 9, 33).
Подписка-2005 (с. 56).

Библиограф
КАН 398

Сетевые источники питания на микросхемах LinkSwitch-TN

Расчет преобразователей следует начать с выбора схемы входного выпрямителя, сглаживающей цепи и элементов подавления помех. На рис. 18 приведены четыре варианта входных цепей. Вариант по схеме на рис. 18, а рекомендуется для преобразователей с выходной мощностью до 0,25 Вт, по рис. 18, б — 0,25...1 Вт, а по рис. 18, в и г — 1 Вт и более. Суммарная емкость входных сглаживающих конденсаторов должна быть не менее 1 мкФ на каждый ватт выходной мощности при мостовом выпрямителе (рис. 18, з) и в два раза больше при однополупериодном. Рекомендуемые номиналы и типы других элементов приведены в табл. 4.

Следующий этап — выбор схемы собственно преобразователя, часть возможных вариантов приведена в предыдущей части статьи.

Выбирают вариант работы — с непрерывным или прерывистым током дросселя — исходя из следующих соображений. В преобразователе с прерывистым током дроссель имеет меньшую индуктивность, он меньше по размерам и дешевле, менее жесткие требования по быстродействию к диоду D_{FW} (рис. 8), в

результате он также дешевле, выше коэффициент полезного действия, однако может потребоваться более дорогая микросхема с большим током ограничения. По совокупности факторов этот вариант, как правило, получается несколько дешевле варианта с непрерывным током дросселя.

Далее в зависимости от выходного напряжения и выходного тока по табл. 5 или 6 следует определить параметры дросселя — индуктивность и рабочий ток, выбрать мик-

росхему и номинал резистора R_{FB} . Табл. 5 предназначена для выбора параметров элементов преобразователей с последовательным включением дросселя (рис. 8—10, 13, 16, 17), табл. 6 — с параллельным (рис. 11, 12, 14). Другие элементы преобразователей могут иметь стандартные номиналы: R_{BIAS} — 2 кОм (рис. 8), C_{BP} — 0,1 мкФ (керамический на напряжение 50 В), конденсаторы C_{FB} — 10 мкФ и C_{OUT} — 100 мкФ с низким последовательным сопротивлением на напряжение, на 25 % превышающее выходное, резистор R_Z , шунтирующий светодиод оптрона, — 0,47...2 кОм (рис. 9, 10, 12).

Если выбран вариант с обратной связью без оптрона, необходима установка резистора начальной нагрузки R_{PL} , его сопротивление и мощность выбирают, исходя из протекающего через него тока 3 мА. Для варианта

Таблица 4

Схема (рис. 15)	R_{F1}		R_{F2}		L_{IN}		D_{IN}		
	Ом	Вт	Ом	Вт	мГн	А	А	В	Возм. тип
а	100...470	0,5	100...470	0,5	—	—	1	1000	1N4007
б	8,2	1	100	0,5	—	—	1	1000	1N4007
в	8,2	1	—	—	0,47...2,2	0,05...0,3	1	1000	1N4007
г	8,2	1	—	—	0,47...2,2	0,05...0,3	1	600	1N4005

Таблица 5

$U_{ВЫХ}$, В	$I_{ВЫХ}$, А	Режим	$L1$, мГн	I_{RMS} , МА	Микросхема	R_{FB} , кОм
5	≤65	Прер.	1200	70	LNK302	3,84
	80	Непр.	1200	80		
	120	Прер.	680	220	LNK304	
	160	Непр.	680	230	LNK305	
	175	Прер.	680	320		
	225	Непр.	680	340	LNK306	
	280	Прер.	680	440		
12	≤60	Прер.	1800	70	LNK302	11,86
	80	Непр.	2700	80	LNK304	
	85	Прер.	680	180		
	120	Непр.	1000	230	LNK305	
	160	Прер.	1500	320		
	175	Непр.	680	340	LNK306	
	225	Прер.	1000	440		
15	≤65	Прер.	2200	70	LNK302	15,29
	80	Непр.	3300	80	LNK304	
	70	Прер.	680	160		
	120	Непр.	1200	210	LNK305	
	160	Прер.	1800	210		
	175	Непр.	820	310	LNK306	
	225	Прер.	1200	310		
24	≤65	Прер.	3300	70	LNK302	25,6
	80	Непр.	4700	80	LNK304	
	50	Прер.	680	130		
	120	Непр.	1500	190	LNK305	
	160	Прер.	2200	180		
	175	Непр.	1200	280	LNK306	
	225	Прер.	1500	280		
24	280	Непр.	1200	350	LNK306	
	360	Прер.	2200	360		

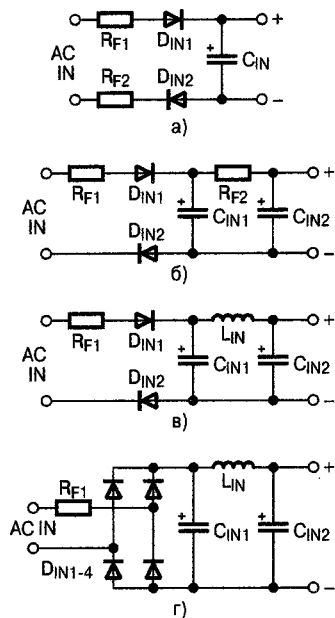


Рис. 18

Таблица 6

U _{вых} , В	I _{вых} , А	Режим	L1, мкГн	I _{РМС} , мА	Микросхема	R _{FB} , кОм
5	≤65	Прер.	1200	70	LNK302	3,84
	80	Непр.	1500	80		
	120	Прер.	680	220	LNK304	
	160	Непр.	680	230		
	175	Прер.	680	340	LNK305	
	225	Непр.	680	320		
	280	Прер.	680	440	LNK306	
360	Непр.	680	430			
12	≤55	Прер.	2200	70	LNK302	11,86
	80	Непр.	3300	90		
	70	Прер.	680	180	LNK304	
	120		1200	220		
	160	Непр.	1800	210	LNK305	
	175	Прер.	820	320		
	225	Непр.	1200	310	LNK306	
	280	Прер.	820	410		
360	Непр.	1800	410			
15	≤55	Прер.	2200	70	LNK302	15,29
	80	Непр.	3900	90		
	50	Прер.	680	180	LNK304	
	120		1500	220		
	160	Непр.	2200	220	LNK305	
	175	Прер.	1000	320		
	225	Непр.	1500	320	LNK306	
	280	Прер.	1200	400		
360	Непр.	2200	410			
24	≤65	Прер.	3300	70	LNK302	25,6
	80	Непр.	6800	100		
	50	Прер.	680	180	LNK304	
	120		2200	210		
	160	Непр.	3300	210	LNK305	
	175	Прер.	1800	300		
	225	Непр.	2200	290	LNK306	
	280	Прер.	1800	370		
360	Непр.	3300	410			

Таблица 7

Тип диода	U _{МАХ} , В	I _{МАХ} , А	T _{RR'} , нс	Оформление	Изготовитель
MUR160	600	1	50	Осевые выводы	Vishay
UF4005	600	1	75	Осевые выводы	Vishay
BYV26C	600	1	30	Осевые выводы	Vishay/Philips
FE1A	600	1	35	Осевые выводы	Vishay
STTA10 6	600	1	20	Осевые выводы	ST Microelectronics
STTA10 6U	600	1	20	Поверхн. монтаж	ST Microelectronics
US1J	600	1	75	Поверхн. монтаж	Vishay

с оптроном резистор начальной нагрузки не нужен.

Диод D_{FB} должен с запасом выдерживать пиковое напряжение сети и обладать хорошим быстродействием, годится, например, 1N4005GP (600 В, 1 А, время восстановления обратного сопротивления — 2 мкс)

или 1N4937GP (600 А, 1 А, 150 нс). Весьма важно правильно подобрать диод D_{FW}. Если выбран режим с непрерывным током дросселя, время восстановления обратного сопротивления диода должно быть не больше 35 нс, если прерывистый — 75 нс. В табл. 7 приведены типы и парамет-

ры некоторых диодов, пригодных для применения в рассматриваемых преобразователях.

Если требуется разработать преобразователь с отличными от предлагаемых в табл. 5 и 6 выходными параметрами, можно использовать вариант с ближайшим большим выходным током и/или напряжением. При этом рекомендуется проверить, что выходной ток меньше или равен половине минимального тока ограничения выбранной микросхемы для прерывистого режима или находится в диапазоне 0,5...0,8 минимального тока ограничения для непрерывного режима.

При отличном от приведенных в табл. 5 и 6 значении требуемого выходного напряжения сопротивление резистора R_{FB} может быть рассчитано по формуле

$$R_{FB} = \frac{V_{OUT} - V_{FB}}{\frac{V_{FB}}{R_{BIAS}} - I_{FB}} = \frac{(V_{OUT} - V_{FB})R_{BIAS}}{V_{FB} + I_{FB}R_{BIAS}} = \frac{(V_{OUT} - 1,65В) \times 2кОм}{1,748В}$$

Следует обратить внимание на то, что в табл. 5 и 6 для тока через дроссель указано его эффективное значение. Практически для расчета и самостоятельного изготовления дросселя необходимо знать максимальный ток через него. Этот ток соответствует максимальному току ограничения выбранной микросхемы (см. табл. 2).

Окончание следует

Сергей Бирюков,
г. Москва

Редакция подготовила CD-ROM с полным содержанием журнала «Схемотехника» за 2000—2002 г. Кроме него на диске записано программное обеспечение к статьям, опубликованным за все время существования журнала, годовое содержание в формате Excel за то же время, поправки ко всем замеченным опечаткам и ошибкам.

Вы можете подписаться на CD-ROM, переведя на расчетный счет редакции стоимость диска с пересылкой — 150 руб. Реквизиты редакции указаны на последней странице журнала. Не забудьте прислать нам по факсу, обычной или электронной почте копию квитанции об оплате с указанием полного адреса с индексом.

Автомат управления насосом на базе микроконтроллера HT48R05A-1

В этой статье автор рассматривает применение недорогого микроконтроллера в качестве основы для создания устройства, которое контролирует уровень воды в водонапорном баке и индицирует этот уровень с помощью светодиодов, а также управляет подачей в него воды.

Для жильцов современных домов слабый напор воды является насущной проблемой, особенно когда вода включается всеми потребителями одновременно (например, вечером во время рекламы на ТВ). Поэтому тот, кто имеет такую возможность, устанавливает на крыше водонапорный бак, в то время как основной бак с водой находится в подвале. Вода из основного бака перекачивается в водонапорный с помощью насоса. Это позволяет решить проблему слабого напора воды и более эффективно ее использовать.

Стандартные устройства могут регулировать уровень воды только между двумя значениями. Однако поскольку это устройство чаще всего устанавливается на водонапорном баке, оно наиболее подвержено коррозии и, соответственно, через какое-то время начинает работать со сбоями. Кроме того, если уровень воды в основном баке (в подвале) слишком низкий, это может привести к выходу из строя водяного насоса.

В обычном регуляторе в качестве датчиков уровня воды используются два поплавка в виде шаров. В соответствии с положением поплавков водяной насос будет включен или выключен. Использование в качестве датчиков уровня воды поплавков упрощает конструкцию и снижает стоимость. Но, как упоминалось выше, контакты выключателя, находящегося на водонапорном баке, будут корродировать, что в конечном итоге приведет к отказу в работе выключателя. Помимо этого, поскольку на основном баке с водой нет датчика уровня, возможен перегрев и выход из строя водяного насоса, если воды в основном баке будет слишком мало.

Так как обычный контроллер уровня воды обладает вышеперечисленными недостатками, многие разработчики пытаются решить эту проблему при помощи цифровых устройств. Кроме того, устанавливают датчик

уровня воды в основном баке — если уровень воды в этом баке становится ниже допустимого, питание насоса будет отключено и насос не выйдет из строя.

В цифровых устройствах для определения уровня воды не исполь-

ным насосом позволяет избавиться от этих недостатков.

Основные характеристики HT48R05A-1

Рабочее напряжение, В	3,3...5,5
Число выводов ввода/вывода	13
Разрядность таймера	8
Память программ, бит	512×14
Память данных, бит	32×8
Число уровней стека	2

HT48R05A-1 представляет собой недорогой быстродействующий восьмиразрядный микроконтроллер с RISC-архитектурой, имеющий 63 мощные команды и двухуровневый аппаратный стек. Блок-схема контроллера приведена на рис. 1, цоколевка — на рис. 2.

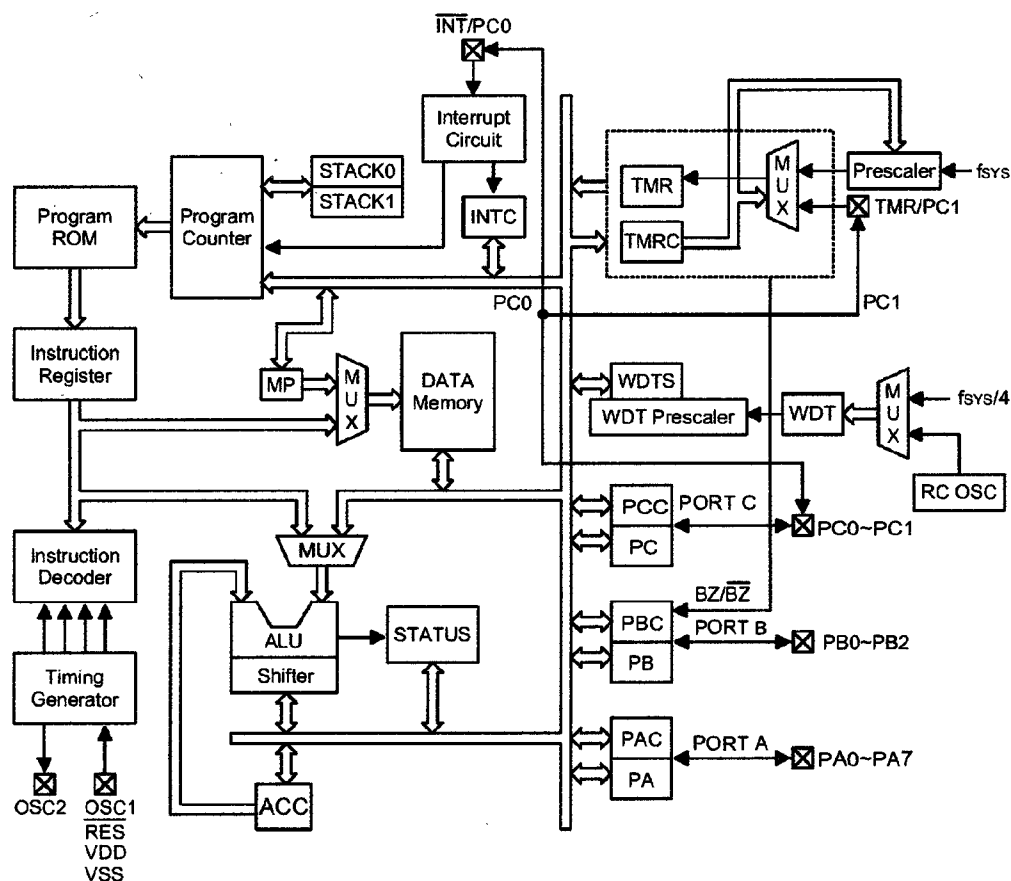


Рис. 1

зуются механические выключатели, следовательно, им не грозит коррозия и сбой в работе. Однако они обладают нередко следующими недостатками:

- нет индикации уровня воды в основном и водонапорном баках;
- когда уровень воды в основном баке низкий, отключается только питание насоса, а уровень воды в самом баке остается неизвестным;
- необходимо несколько цифровых микросхем, что усложняет разработку.

Применение микроконтроллера HT48R05A-1 в устройстве определения уровня воды и управления водя-

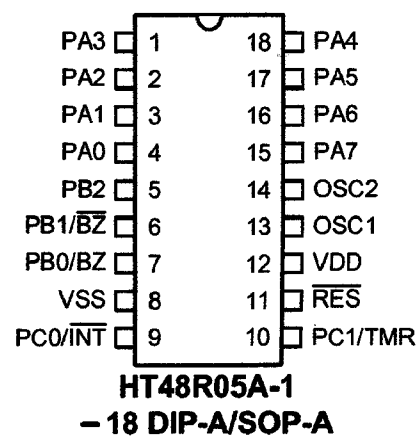


Рис. 2

Тактовая частота HT48R05A-1 определяется внешней RC-цепью или кварцевым резонатором. Микроконтроллер оформлен в корпусе 16SSOP-A, 18DIP или 18SOP-A (только для OTP).

Внешнее прерывание поступает на вывод PC0, внешний счетчик подключен к выводу PC1.

Сторожевой таймер (WDT) прерывает выполнение программы при случайном переходе в неопределенную область памяти, что может быть вызвано каким-либо неуправляемым внешним событием (например, электрическими помехами). Если это происходит, то сторожевой таймер перезапустит микроконтроллер для предотвращения сбоев в работе. Узел контроля напряжения питания автоматически выполнит перезапуск HT48R05A-1, если напряжение питания упадет ниже заданного уровня. Хотя объем памяти программ только 512×14 бит, этого достаточно для большинства применений. Память данных составляет 32×8 бит. Прерывания могут быть внешними и внутренними (от таймера/счетчика).

Тринадцать двунаправленных входов/выходов включают восемь выводов портов PA, три вывода портов PB и два вывода портов PC. Управление этими выводами может выполняться программным способом — вывод может быть определен как выход КМОП или как вход с триггером Шмитта, подключение подтягивающего резистора производится программно. Выводы PB0 и PB1 — выводы для подключения выводов зуммера BZ и \overline{BZ} (инверсный) соответственно. Для выводов INT (инверсный) и TMR предназначены порты PC0 и PC1. В том случае, когда порт PC0 надо установить на работу от внешнего прерывания, он должен быть сначала установлен как вход. Если требуется вход для внешнего таймера, то как вход должен быть установлен PC1.

В состав HT48R05A-1 входит восьмиразрядный таймер, который может работать от внешнего источника

тактового сигнала или использовать внутренний системный тактовый генератор. Если в качестве тактовой выбрана частота внутреннего генератора, она может быть пропущена через предварительный делитель, обеспечивая таймер восемью опорными частотами, которые находятся в пределах 1...1/128 частоты системного генератора.

АППАРАТНАЯ РЕАЛИЗАЦИЯ

На рис. 3 приведена принципиальная схема предлагаемого устройства. Автомат состоит из микроконтроллера DD1, четырех светодиодов HL1—HL4 индикации уровня воды в основном и водонапорном баках, полупроводникового реле SSR для включения водяного насоса M1 и нескольких дополнительных элементов.

Основная идея заключается в установке портов PA0—PA6 как входов детектора уровня. Как только вода достигает любого из датчиков, подключенного к портам PA0 — PA6, напряжение питания поступает на конкретный вывод. Считывая состояние портов, можно определить уровень воды. Порты PC0 и PC1 устанавливаются как выходы — PC0 для управления водяным насосом, а PC1 — для индикации уровня воды в основном баке.

Алгоритм работы устройства следующий:

- когда в основном баке воды достаточно, горит светодиод HL4. В этом случае может быть включен водяной насос;
- при достижении воды в водонапорном баке уровня B начнет мигать светодиод HL1, если вода достигнет уровня C, то светодиод HL1 будет постоянно светиться;
- при достижении уровня воды в водонапорном баке уровня D светодиод HL1 будет постоянно све-

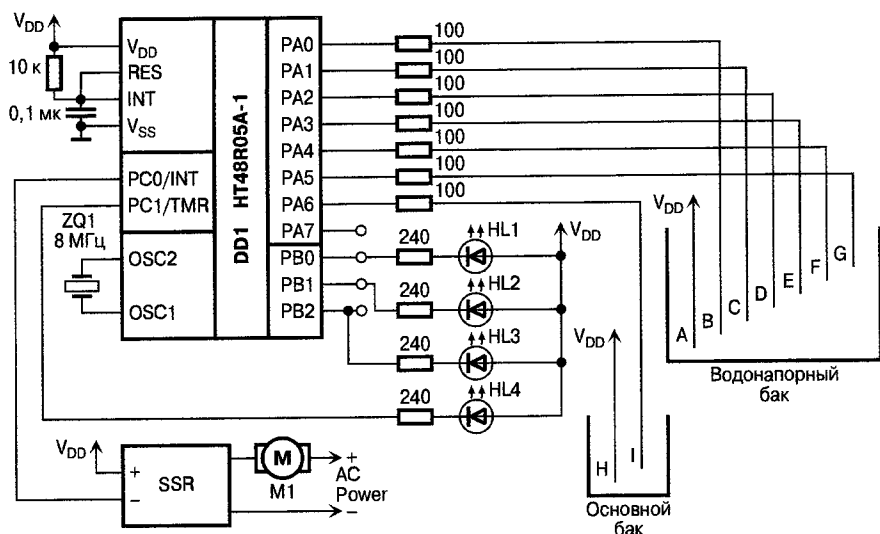


Рис. 3

CHIPSELECT

ПОСТАВКА ЗАКАЗОВ НА КОМПОНЕНТЫ

официальный
дистрибьютор Holtek

поставки
всего спектра
продукции
Holtek

HOLTEK

ВОЗМОЖНОСТЬ
монтажа
CHIP-ON-BOARD,
CHIP-ON-GLASS,
CHIP-ON-CABLE

(095) 267-47-33

(095) 263-96-64

(095) 263-27-37

(095) 101-37-65

(многоканальный)

info@chipselect.ru

www.chipselect.ru

тяться и начнет мигать светодиод HL2; если вода достигнет уровня E, то светодиоды HL1 и HL2 будут постоянно светиться;

- когда вода в водонапорном баке достигнет уровня F, то светодиоды HL1 и HL2 будут постоянно светиться и начнет мигать светодиод HL3; если вода достигнет уровня G (бак полон), то будут постоянно светиться все три светодиода HL1, HL2 и HL3;
- микроконтроллер включит водяной насос, если только уровень воды в водонапорном баке опустится ниже уровня B и уровень воды в основном баке будет выше уровня I, что позволит предотвратить выход из строя насоса;
- при работе насоса светодиоды будут мигать в последовательности HL1 → HL2 → HL3 → HL1 → HL2 → HL3;
- когда уровень воды в основном баке опустится ниже отметки I, насос отключится, погаснет светодиод HL4, а светодиоды HL1, HL2 и HL3 укажут уровень воды в водонапорном баке.

По сравнению с обычными контроллерами уровня воды предложенная система обладает следующими преимуществами:

- нет механических выключателей — нет проблем с коррозией;
- если уровень воды в основном баке меньше заданного, насос будет отключен, что предотвратит его выход из строя;

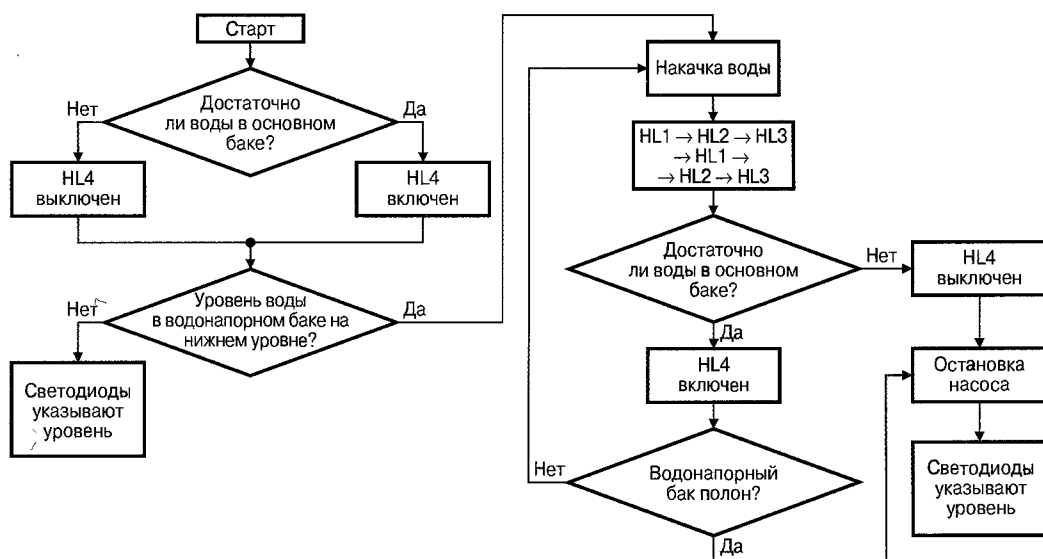


Рис. 4

- уровни воды в основном и водонапорном баках постоянно контролируются, а их состояние отображается светодиодами;
- применение микроконтроллера HT48R05A-1 позволяет уменьшить число внешних компонентов и тем самым повысить надежность системы.

Программное обеспечение достаточно простое и легко модифицируется для конкретной задачи. На рис. 4 приведена блок-схема программы.

Исходный текст программы можно найти на сайте журнала по адресу <http://www.dian.ru/programs>.

Wen-Ching Huang,
A Circuit for Water
Level Detection and Control.

Перевод Николая Раковича,
г. Москва.

От редакции

В статье есть неточность. Микроконтроллер HT48R05A-1 содержит подтягивающие резисторы, поэтому для изменения состояния его входов на них необходимо подавать лог. 0, а не лог. 1. Для этого электроды А и Н должны быть подключены к общему проводу, а не к плюсу питания. Если баки металлические, эти электроды не нужны, достаточно соединить их с общим проводом устройства. Подробные сведения о микроконтроллере можно найти на сайте изготовителя по адресу http://www.holtek.com/pdf/uc/48x0506_1v120.pdf.

СЕМИНАРЫ НА DSPA-2005

В рамках седьмой конференции и выставки «Цифровая обработка сигналов и ее применение — DSPA'2005» (16—18 марта 2005 г., Москва, м. Калужская, Институт проблем управления) состоятся следующие семинары:

- «Отечественные серии сигнальных контроллеров «Мультикор», преобразователей частоты «Мультифлекс», элементов системного сопряжения «Мультикор-конструктор» разработки ГУП НПП «ЭЛВИС», программное обеспечение, отладочные средства и модули обработки сигналов» (16 марта, 14.00—16.00).
- «Модули цифровой обработки сигналов и телекоммуникаций фирмы «СКАН Инжиниринг Телеком»» (17 марта, 10.00—14.00).
- «Процессоры, программное обеспечение и отладочные средства ANALOG DEVICES». Семинар проводят специалисты ANALOG DEVICES, Inc. и AUTEX Ltd. Как всегда, предоставляются раздаточные материалы; разыгрываются призы и подарки (17 марта, 11.00—18.00).

- «Современные средства сбора и цифровой обработки сигналов производства ЗАО «ИнСис»» (17 марта, 15.00—17.00).

Конференцию откроет Пленарное заседание, которое состоится 16 марта с 10.00 до 13.00, заседание Секции 7 «Проектирование и техническая реализация систем ЦОС» будет проведено 18 марта с 10.00 до 13.00.

Участие во всех мероприятиях бесплатное. Обеспечивается перевод на русский язык.

Во время семинаров, на заседаниях секций Конференции и на стендах компаний-участников Выставки вы сможете получить консультации по интересующим вас вопросам, касающимся проблематики ЦОС. **Ждем вас на DSPA'2005!**

По информации <http://www.autex.ru>

Продолжение. Начало — № 3/2004

Анализ SFR-совместимости микроконтроллеров фирмы Silabs

В рамках настоящей статьи цикла произведен системный анализ совместимости подсистем генераторов, регистров управления Flash-памятью, внешней памятью и SFR памятью.

ПОДСИСТЕМЫ ГЕНЕРАТОРОВ И УПРАВЛЕНИЕ ПАМЯТЬЮ

В перечне общесистемных подсистем SFR регистров микроконтроллеров фирмы SiLabs [1] остались нерассмотренными только четыре подсистемы:

- подсистема тактовых генераторов;
- подсистема регистров управления Flash памятью;
- подсистема управления интерфейсом внешней оперативной памяти;
- подсистема управления страницами SFR регистров.

Все SFR регистры вышеперечисленных подсистем приведены в табл. 1.

Подсистема тактовых генераторов

Подсистема тактовых генераторов достаточно проста. Все семейства микроконтроллеров имеют два основных SFR регистра управления встроенным (OSCICN) и внешним (OSCXCN) тактовыми генераторами. В ранних семействах (F00x, F01x, F02x и F2xx) никаких других SFR регистров не предусматривалось. В остальных семействах добавился еще регистр калибровки встроенного генератора OSCICL и регистра выбора генератора CLKSEL. Кроме того, в семействе F33x имеется второй встроенный низкочастотный генератор, который также имеет свой регистр управления OSCLCN. Помимо этого, в трех се-

мействах микроконтроллеров имеется умножитель тактовой частоты (F12x, F32x и F35x). В первом из названных семейств (F12x) он достаточно сложен, выполнен в виде отдельной подсистемы и внесен в состав цифровых периферийных подсистем (т. е. не будет рассматриваться в настоящей статье). В семействах F32x и F35x он более прост, для его управления применен всего один SFR регистр — CLKMUL, он будет рассмотрен.

Назначение битов регистра управления встроенным генератором OSCICN приведено в табл. 2.

Во-первых, отметим, что регистры OSCICN располагаются по разным адресам. В микроконтроллерах с одностраничной организацией SFR адрес регистра — 0xB2, а с многостраничной — 0x8A страницы F.

Во-вторых, как показано в табл. 2, даже назначение битов в различных семействах значительно отличается. Отсюда следует главный вывод — регистр OSCICN всегда требует индивидуальной настройки. Назначение битов регистра OSCICN приведено ниже.

MSCLKE (Missing Clock Enable Bit) — бит разрешения датчика отсутствия тактовой частоты (1 — включено/0 — выключено).

DC/DC преобразователи

для поверхностного монтажа
Полный ряд от 1 до 20 Вт

1W Серия TSM • Корпус SOIC-14/18 • Прочность изоляции 1000 В
• Диапазон рабочих температур -40 ... +85 °C

New:

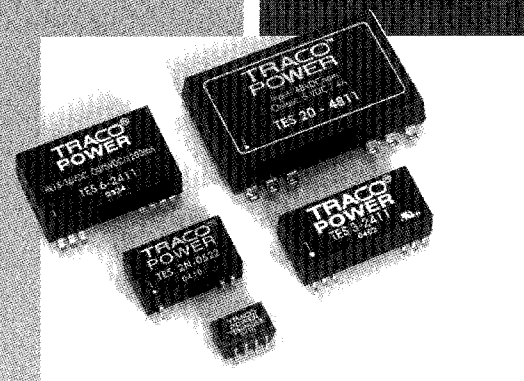
1W Серия TSV • Корпус SOIC-22 • Прочность изоляции 3000 В
• Диапазон рабочих температур -40 ... +85 °C

2W Серия TES-2N • Компактный корпус DIL-16
• Прочность изоляции 1500 В • Стабилизированный выход
• Широкий входной диапазон 4.5-9, 9-18, 18-36 и 36-75 В

3W Серия TES-3 • Корпус DIL-24 • Прочность изоляции 1500 В
• Широкий входной диапазон • Стабилизированный выход

6W Серия TES-6 • Корпус DIL-24 • Прочность изоляции 1500 В
• Соответствие стандартам ЭМС EN 55022, class A

20W Серия TES-20 • Компактное исполнение 2"x1,2" • Прочность изоляции 1500 В
• Входной диапазон 36 - 75 В • Выходные напряжения 2, 3.3, 5, 12, 15 В



Официальный дистрибьютор

ARGUSSOFT

<http://components.argussoft.ru>

ЗАО «АРГУССОФТ Компани»
ДЕПАРТАМЕНТ МИКРОЭЛЕКТРОНИКИ

Москва
Тел.: (095) 217 2487, 217 2519
components@argussoft.ru

Санкт-Петербург
Тел.: (812) 567 1867
spb@argussoft.ru

Екатеринбург
Тел.: (343) 378 3242
ural@argussoft.ru

Новосибирск
Тел.: (3832) 27 1155
nsk@argussoft.ru

Таблица 1

Название регистра	Назначение	Адрес SFR регистра											
		F00x	F01x	F02x	F04x	F06x	F12x	F2xx	F30x	F31x	F32x	F33x	F35x
Регистры генераторов													
OSCICL	Калибровка встроенного генератора	—	—	—	0xB3/F	0xB3/F	0xB3/F	—	0xB3	0xB3	0xB3	0xB3	0xB3
OSICN	Управление внутренним генератором	0xB2	0xB2	0xB2	0x8A/F	0x8A/F	0x8A/F	0xB2	0xB2	0xB2	0xB2	0xB2	0xB2
OSXCXN	Управление внешним генератором	0xB1	0xB1	0xB1	0x8C/F	0x8C/F	0x8C/F	0xB1	0xB1	0xB1	0xB1	0xB1	0xB1
OSCLCN	Управление НЧ генератором	—	—	—	—	—	—	—	—	—	—	0xE3	—
CLKSEL	Выбор генератора	—	—	—	0x97/F	0x97/F	0x97/F	—	—	0xA9	0xA9	0xA9	0xA9
CLKMUL	Регистр управления умножителем	—	—	—	—	—	—	—	—	—	0xB9	—	0xBE
Регистры Flash-памяти													
FLACL	Управление доступом к Flash	0xB7	0xB7	0xB7	0xB7/F	0xB7/F	0xB7/F	0xB7	0xB7	0xB7	0xB7	0xB7	0xB7
FLSCL	Временя доступа к Flash	0xB6	0xB6	0xB6	0xB7/0	0xB7/0	0xB7/0	0xB6	0xB6	0xB6	0xB6	0xB6	0xB6
PSCTL	Управление R/W к Flash	0x8F	0x8F	0x8F	0x87/0	0x87/0	0x87/0	0x8F	0x8F	0x8F	0x8F	0x8F	0x8F
FLSTAT	Flash статус	—	—	—	—	—	0x88/F	—	—	—	—	—	—
PSBANK	Управление Flash банками	—	—	—	—	—	0xB1/A	—	—	—	—	—	—
Регистры внешней памяти EMI													
EMIOCF	Конфигурация внешней памяти	—	—	0xA3	0xA3/0	0xA3/0	0xA3/0	—	—	—	—	—	—
EMIOCN	Интерфейс внешней памяти	0xAF	0xAF	0xAF	0xA2/0	0xA2/0	0xA2/0	0xAF	—	0xAA	0xAA	0xAA	0xAA
EMIOTC	Регистр временного контроля EMIF	—	—	0xA1	0xA1/0	0xA1/0	0xA1/0	—	—	—	—	—	—
Регистры SFR страниц													
SFRLAST	Предыдущая страница стека SFR	—	—	—	0x86/A	0x86/A	0x86/A	—	—	—	—	—	—
SFRNEXT	Следующая страница стека SFR	—	—	—	0x85/A	0x85/A	0x85/A	—	—	—	—	—	—
SFRPAGE	Регистр страниц SFR	—	—	—	0x84/A	0x84/A	0x84/A	—	—	—	—	—	—
SFRPGCN	Управление SFR	—	—	—	0x96/F	0x96/F	0x96/F	—	—	—	—	—	—

Таблица 2

Название регистра — OSICN, Internal Oscillator Control Register (Регистр управления внутренним тактовым генератором)								
SFR адрес — 0xB2		Значение после сброса — 0000100b (0x04)						
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F00x, F01x, F02x, F2xx	MSCLKE	—	—	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0
F04x, F06x, F12x, F31x, F33x, F35x	IOSCEN	IFRDY	—	—	—	—	IFCN1	IFCN0
F30x	—	—	—	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0
F32x	IOSCEN	IFRDY	SUSPEND	—	—	—	IFCN1	IFCN0

IFRDY (Internal Oscillator Frequency Ready Flag) — флаг готовности внутреннего генератора, устанавливается в 1 после нормального запуска внутреннего генератора на частоте, установленной битами IFCN0.

CLKSL (System Clock Source Select Bit) — бит выбора источника тактовых импульсов. Установка в 1 включает выбор внешнего генератора, в 0 — внутреннего.

IOSCEN (Internal Oscillator Enable Bit) — бит разрешения внутреннего генератора (1 — разрешен).

IFCN (Internal Oscillator Frequency Control Bits) — биты установки частоты внутреннего генератора.

В ранних моделях (F00x, F01x, F2xx):

- 00 — примерно 2 МГц;
- 01 — примерно 4 МГц;
- 10 — примерно 8 МГц;
- 11 — примерно 16 МГц.

В более поздних моделях:

- 00 — SYSCLK = F/8;
- 01 — SYSCLK = F/4;
- 10 — SYSCLK = F/2;
- 11 — SYSCLK = F/1.

SUSPEND — запись 1 останавливает внутренний генератор. Генератор запускается при следующей пересылке по интерфейсу USB или по прерыванию VBUS.

Регистр OSCXCXN также присутствует во всех семействах. В микроконтроллерах с одностраничной организацией SFR адрес регистра 0xB1, а с многостраничной — 0x8C страницы F. Назначение битов регистра OSCXCXN приведено в табл. 3.

Для всех семейств назначение битов регистра OSCXCXN совпадает по наименованию и расположению.

XTLVLD (Cristal Oscillator Valid Flag) — флаг готовности кварцевого генератора (работает только если XOSCMD = 1xx). При выходе генератора на нормальный режим устанавливается в 1.

XOSCMD2-0 (External Oscillator Mode Bits) — биты установки режима внутреннего генератора:

- 00x — выключен;
- 010 — внешний генератор, подключенный на вывод XLAT1;
- 011 — внешний генератор, подключенный на вывод XLAT1 с делением его частоты на 2;
- 10x — RC/C генератор с делением его частоты на 2;

- 110 — кварцевый резонатор;
- 111 — кварцевый резонатор с делением его частоты на 2.

XFCN (External Oscillator Frequency Control Bits) — код коэффициента установки частоты.

Код коэффициента XFCN различается для различных семейств и различных диапазонов тактовых частот. Значения кода обычно приводятся в специальной для каждого семейства таблице.

Регистр калибровки встроенного генератора OSCICL появился в составе подсистемы тактового генератора в семействах F04x, F06x, F12x, F30x, F31x, F32x, F35x. Он располагается по адресу 0xB3 во всех семействах, в которых он существует. Все его биты содержат калибровочный код, значение которого может отличаться для различных экземпляров микроконтроллеров. Код подбирается в заводских условиях. Пользователь может изменять этот код следующим образом.

Если после сброса тактовая частота равна f_{BASE} (период T_{BASE}), а требуемая частота — f_{DES} (период T_{DES}), например, $f_{BASE} = 24,5$ МГц, $f_{DES} = 20$ МГц, тогда

$$T_{DES} = 1/f_{DES} - 1/f_{BASE} = 0,005 \times 1/f_{BASE} \times OSCICL;$$

$$OSCICL = (f_{BASE} - f_{DES}) / (0,005 \times f_{DES}) = (24,5 - 20) / (0,005 \times 20) = 45.$$

Таблица 3

Название регистра — OSCXCXN, External Oscillator Control Register (Регистр управления внешним тактовым генератором)								
SFR адрес — 0xB1		Значение после сброса — 00110000b (0x30)						
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Все	XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	—	XFCN2	XFCN1	XFCN0

Регистр выбора генератора CLKSEL появился в составе подсистемы тактового генератора в семействах F04x, F06x, F12x, F31x, F32x, F35x. В микроконтроллерах с одностраничной организацией SFR адрес регистра — 0xA9, а с многостраничной — 0x97 страницы F. Назначение битов регистра CLKSEL приведено в табл. 4.

Назначение выводов регистра CLKSEL приведено далее.

CLKDIV (Output SYSCLK Divide Factor) — биты определения частоты выходного сигнала SYSCLK:

- 00 — SYSCLK = системная частота F;
- 01 — SYSCLK = F/2;
- 10 — SYSCLK = F/4;
- 11 — SYSCLK = F/8.

CLKSL (System Clock Source Select Bits) — бит выбора тактового генератора:

- 00 — включен внутренний тактовый генератор;
- 01 — включен внешний тактовый генератор;
- 10 — включена система PLL (F12x) или частота умножителя/2 (поделенная на два) для F32x или внутренний низкочастотный генератор (F33x, F35x).

USBCLK (USB Clock Source Select Bits) — биты выбора тактового генератора для USB. Для USB в режиме полной скорости частота (Full Speed) должна быть 48 МГц, для экономичного (Low Speed) режима — 6 МГц:

- 000 — умножитель на 4;
- 001 — внутренний генератор/2;
- 010 — внешний генератор;
- 011 — внешний генератор/2;
- 100 — внешний генератор/3;
- 101 — внешний генератор/4.

Регистр выбора режима низкочастотного генератора OSCLCN имеется только в семействе F33x, так что ни о какой совместимости говорить не приходится. Назначение битов регистра OSCLCN приведено в табл. 5.

Назначение выводов регистра OSCLCN приведено ниже:

OSCLEN (Internal L—F Oscillator Enable) — бит разрешения (1) низкочастотного генератора.

OSCLRDY (Internal L—F Oscillator Ready) — бит готовности (1) низкочастотного генератора.

OSCLF (Internal L—F Oscillator Frequency Control) — биты выбора частоты: 0000b соответствует максимальной частоте, 1111b — минимальной. В процессе фабричной калибровки максимальная частота устанавливается равной $f_{BASE} = 80$ кГц.

Таблица 4

Название регистра — CLKSEL, Oscillator Clock Selection Register (Регистр выбора тактового генератора)								
SFR адрес — 0xA9			Значение после сброса — 00000000b (0x00)					
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F00x, F01x, F02x, F2xx, F30x	—	—	—	—	—	—	—	—
F04x, F06x, F31x	—	—	—	—	—	—	—	CLKSL
F12x	—	—	CLKDIV1	CLKDIV0	—	—	CLKSL1	CLKSL0
F32x	—	USBCLK2	USBCLK1	USBCLK0	—	—	CLKSL1	CLKSL0
F33x, F35x	—	—	—	—	—	—	CLKSL1	CLKSL0

Таблица 5

Название регистра — OSCLCN, Internal L—F Oscillator Control Register (Регистр управления низкочастотным генератором)								
SFR адрес — 0xE3			Значение после сброса — 00000000b (0x00)					
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Таблица 6

Название регистра — CLKMUL, Clock Multiplier Control Register (Регистр управления умножителем)								
SFR адрес — 0xB9			Значение после сброса — 00000000b (0x00)					
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F32x, F35x	MULEN	MULINIT	MULRDY	—	—	—	MULSEL1	MULSEL0

OSCLD (Internal L—F Oscillator Divider Select) — биты определяют коэффициент деления для получения SYSCLK:

- 00 — F/8;
- 01 — F/4;
- 10 — F/2;
- 11 — F.

Регистр управления режимом умножителя CLKMUL имеется только в семействах F32x и F35x, причем находятся они по разным SFR адресам (0xB9 и 0xBE для семейств F32x и F35x соответственно). Назначение битов регистра CLKMUL приведено в табл. 6.

Назначение выводов регистра CLKMUL приведено ниже:

MULEN (Clock Multiplier Enable) — бит разрешения (1) умножителя.

MULINIT (Clock Multiplier Initialize) — бит инициализации. Бит должен быть 0 при установке MULEN. Бит аппаратно устанавливается в 1 при стабилизации умножителя.

MULRDY (Clock Multiplier Ready) — бит индикации состояния умножителя:

- 0 — не готов;
- 1 — готов и захват частоты произведен.

MULSEL — определяют источник частоты для умножителя:

- 00 — внутренний генератор;
- 01 — внешний генератор;
- 10 — внешний генератор/2.

Отметим, что, несмотря на простоту подсистемы тактовых генераторов микроконтроллеров фирмы SiLabs SFR регистры этой подсистемы практически несовместимы.

Это означает на практике, что для каждого из семейств микроконтроллеров необходимо создавать индивидуальные подпрограммы инициализации подсистемы тактовых генераторов.

Продолжение следует

ЛИТЕРАТУРА:

1. <http://www.silabs.com>

Олег Николайчук,
г. Кишинев, Молдавия

В статье А. Карпенко «Расчет трансформатора однотактного обратного источника питания», опубликованной в журнале № 1 за 2005 г. (с. 35—37), пропущены рис. 1 и 2. Приводим здесь эти рисунки и приносим свои извинения читателям.

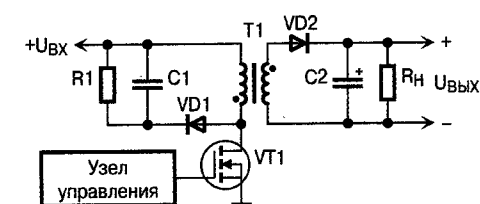


Рис. 1

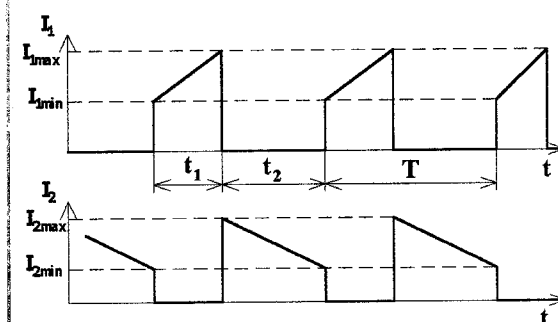


Рис. 2

Применение ПЛИС семейства CoolRunner-II для реализации контроллера интерфейса I²C

РЕГИСТРЫ, ВХОДЯЩИЕ В СОСТАВ КОНТРОЛЛЕРА ИНТЕРФЕЙСА МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ

Каждый из регистров специального назначения, входящий в состав контроллера интерфейса микропроцессорной системы, содержит восемь разрядов. При последующем рассмотрении структуры этих регистров используется нумерация разрядов, начинающаяся с нуля. Младший разряд считается нулевым, соответственно старший — седьмым.

В регистре адреса Address Register (MADR) младший разряд не используется. Остальные разряды этого регистра (с первого по седьмой) предназначены для хранения адреса, используемого контроллером интерфейса I²C в подчиненном (Slave) режиме.

Два младших разряда регистра управления Control Register (MBCR) являются резервными. Разряды со второго по седьмой содержат информацию об основных параметрах конфигурации контроллера интерфейса I²C. Значение бита, записанного во второй разряд регистра управления Repeated Start (RSTA), определяет возможность генерации повторного признака начала новой операции START на шине I²C. Нулевое значение этого бита соответствует обычному режиму работы шины I²C. Если контроллер работает в ведущем (Master) режиме, то при записи лог. 1 в разряд RSTA выполняется генерация повторного признака начала новой операции START на шине I²C. В третий разряд регистра управления записывается бит Transmit Acknowledge Enable (TXAK), значение которого определяет состояние сигнала на линии SDA в течение цикла подтверждения передачи данных. Когда в этом разряде регистра MBCR лог. 1, подтверждение передачи данных отсутствует. Четвертый разряд регистра управления используется для хранения бита Transmit/Receive Mode Select (MTX), который указывает направление пе-

редачи данных на шине I²C. Единичное значение бита MTX соответствует передаче данных ведущим (Master) устройством подчиненному (Slave), а нулевое — приему информации ведущим устройством от подчиненного. В пятый разряд регистра MBCR заносится бит Master/Slave Mode Select (MSTA), значение которого определяет режим работы контроллера интерфейса I²C. Если в данном разряде регистра управления присутствует лог. 1, контроллер функционирует в ведущем режиме. Нулевое значение этого разряда соответствует подчиненному режиму работы контроллера. При переключении бита MSTA из нулевого состояния в единичное контроллер вырабатывает признак начала новой операции START на шине I²C в ведущем режиме. При обратном изменении этого бита контроллер формирует признак завершения операции STOP и переключается в подчиненный режим (Slave). Шестой разряд регистра управления предназначен для записи бита Interrupt Enable (MIEN), значение которого определяет режим обработки прерываний. Если в этом разряде лог. 1, то обработка прерываний разрешена. Сброс бита MIEN в нулевое состояние приводит к запрету обслуживания прерываний. При этом не отменяются (не сбрасываются) те прерывания, обработка которых к моменту запрета была отложена. В седьмой разряд регистра управления заносится значение бита Controller Enable (MEN), который содержит информацию о разрешении или запрете работы контроллера интерфейса I²C. Значение этого разряда должно устанавливаться в первую очередь. Состояние остальных разрядов регистра MBCR изменяется только после перевода контроллера в разрешенное состояние. Если бит MEN принимает нулевое значение, то контроллер переводится в запрещенное состояние. Для переключения контроллера в активное (разрешенное) состояние в

седьмой разряд регистра управления необходимо записать лог. 1.

Регистр статуса Status Register (MBSR) предназначен для записи данных, отражающих текущее состояние контроллера интерфейса I²C. В нулевой разряд заносится бит Received Acknowledge Bit (RXAK), значение которого отражает состояние сигнала на линии SDA во время цикла подтверждения передачи данных. Нулевое значение этого бита указывает на то, что подтверждение передачи данных получено, лог. 1 говорит об отсутствии подтверждения. Первый разряд регистра MBSR содержит бит Interrupt Bit (MIF), отражающий наличие отложенного прерывания. Этот бит принимает единичное значение в том случае, когда возникает запрос обработки прерывания при наличии единичного значения бита MIEN (в шестом разряде регистра управления). Содержимое первого разряда регистра статуса сбрасывается в нулевое состояние программным способом во время выполнения процедуры обслуживания прерывания. Во втором разряде регистра MBSR хранится значение бита Slave Read/Write Bit (SRW). Этот разряд используется в том случае, когда контроллер интерфейса I²C функционирует в подчиненном (Slave) режиме. Нулевое значение бита SRW указывает на то, что ведущее устройство осуществляет операцию передачи (записи) данных к подчиненному устройству. Наличие лог. 1 во втором разряде регистра статуса говорит о том, что ведущее устройство выполняет операцию приема (чтения) данных от подчиненного устройства. Третий разряд регистра MBSR является резервным, четвертый предназначен для записи значения бита Arbitration Lost Bit (MAL). Данный разряд устанавливается в состояние лог. 1 аппаратными средствами при нарушении процедуры разрешения конфликтных ситуаций на шине I²C. Сброс данного бита в нулевое состояние производится программно. В пятый разряд регистра статуса заносится бит Bus Busy Bit (MBB), значение которого отражает состояние шины I²C. Нулевое значение данного бита соответствует свободному состоянию шины I²C, а единичное указывает на то, что эта шина занята выполнением некоторой операции. При обнаружении на шине I²C признака начала новой операции START в пятый разряд регистра MBSR

записывается лог. 1. Сброс бита MBB в нулевое состояние осуществляется после обнаружения признака завершения операции STOP. Шестой разряд регистра статуса используется для хранения бита Addressed as Slave Bit (MAAS), информирующего о том, что к контроллеру интерфейса I²C производится обращение как к подчиненному устройству. Данный бит устанавливается в 1 в случае, когда адрес, передаваемый по шине I²C, совпадает со значением, записанным в регистр MADR, при этом контроллер переключается в подчиненный режим. Содержимое седьмого разряда регистра MBSR отражает значение бита Data Transferring Bit (MCF), которое информирует о ходе выполнения процесса передачи байта данных по шине I²C. Нулевое значение этого бита указывает на то, что в текущий момент времени выполняется пересылка байта данных. Единица в бите MCF сообщает о завершении процесса передачи байта данных по шине I²C. В момент начала передачи очередного байта данных в седьмой разряд регистра статуса записывается нулевое значение. После пересылки этого байта во время цикла подтверждения передачи по фронту сигнала синхронизации SCL производится перезапись содержимого этого разряда регистра MBSR, в результате которой он устанавливается в состояние лог. 1.

Регистр данных Data Register (MBDR), предназначенный для побайтной записи информации, передаваемой и принимаемой по шине I²C, выполнен в виде пары восьмиразрядных регистров. В один из этих регистров заносится байт данных, который поступает со стороны микропроцессорной системы и должен быть передан на шину I²C. Второй регистр используется для записи информационного байта, который принимается от периферийного устройства, подключенного к шине I²C. Нулевой разряд этих регистров соответствует младшему биту данных (D0), а седьмой — старшему (D7).

ВНЕШНИЕ ЦЕПИ КОНТРОЛЛЕРА ИНТЕРФЕЙСА I²C, РЕАЛИЗУЕМОГО НА ОСНОВЕ ПЛИС СЕМЕЙСТВА COOLRUNNER-II XC2C256-5-TQ144

VHDL-описание контроллера интерфейса I²C, предназначенного для реализации на основе ПЛИС семейства CoolRunner-II, представлено в виде объекта I²C. Интерфейсные цепи (сигналы) контроллера описываются в форме портов этого объекта. Двух-

направленный порт SDA соответствует входу/выходу сигнала данных Serial Data интерфейса I²C. Двухнаправленный порт SCL представляет вход/выход тактового сигнала, подключаемый к линии синхронизации Serial Clock шины I²C. 24-разрядный входной порт ADDR_BUS, описываемый в форме вектора ADDR_BUS[23:0], соответствует входам адреса, подключаемым к шине адреса Address Bus микропроцессорной системы. Восемьразрядный двухнаправленный порт DATA_BUS, представленный в виде вектора DATA_BUS[7:0], описывает информационные входы/выходы рассматриваемого контроллера, сопрягаемые с шиной данных Data Bus. Порт AS соответствует входу сигнала стробирования адреса Address Strobe шины управления. Активным уровнем этого сигнала является лог. 0, который сигнализирует о достоверности значения адреса, представленного на адресных входах ADDR_BUS [23:0]. Порт DS соответствует входу сигнала стробирования данных Data Strobe шины управления. Наличие сигнала низкого логического уровня на этом входе указывает на достоверность информации, передаваемой по шине данных Data Bus. Порт R_W описывает вход сигнала управления Read/Write. Значение этого сигнала определяет тип выполняемой операции: чтения или записи. Лог. 1 сигнала Read/Write указывает на то, что выполняется операция чтения данных, а лог. 0 соответствует операции записи. Порт DTACK соответствует выходу сигнала подтверждения Data Transfer Acknowledge. Для этого сигнала активным состоянием считается лог. 0. Порт IRQ описывает выход сигнала запроса прерывания Interrupt Request, генерируемого контроллером интерфейса I²C. Активным уровнем этого сигнала является уровень лог. 0, который информирует о наличии запроса на прерывание со стороны контроллера шины I²C. Порт MCF представляет выход сигнала завершения передачи одного байта данных Data Transferring Bit. Порт CLK соответствует входу тактового сигнала контроллера Clock. Порт Reset описывает вход сигнала сброса контроллера интерфейса I²C Reset.

Информацию о номерах выводов микросхемы XC2C256-5-TQ144, к которым подключены все перечисленные выше внешние цепи контроллера интерфейса I²C, можно найти в файле отчета о результатах размещения и трассировки проекта в ПЛИС. Следу-

ет обратить внимание на то, что при повторном выполнении трансляции, размещения и трассировки проекта нумерация соответствующих функциональных выводов может измениться. Для того, чтобы зафиксировать расположение входов и выходов контроллера в ПЛИС, необходимо используя одну из систем проектирования серии Xilinx ISE создать и включить в состав проекта файл временных и топологических ограничений User Constraints File (UCF). В этом файле можно явно указать требуемую привязку внешних цепей контроллера к выводам микросхемы, после чего следует заново выполнить этапы трансляции, размещения и трассировки проекта в кристалле.

ПРОТОКОЛ ПЕРЕДАЧИ ДАННЫХ, РЕАЛИЗУЕМЫЙ КОНТРОЛЛЕРОМ ИНТЕРФЕЙСА I²C

Протокол обмена данными между микропроцессорной системой и периферийными устройствами, подключенными к шине I²C, который поддерживается контроллером, реализуемым на основе ПЛИС семейства CoolRunner-II, включает в себя следующие основные фазы:

- передача адреса устройства;
- ввод данных;
- завершение процесса передачи (записи или чтения) данных;
- окончание текущего цикла.

Рассмотрим подробнее эти операции. Очередной цикл чтения или записи данных начинается с передачи адреса устройства, с которым должен осуществляться информационный обмен. В начале этой фазы микропроцессор формирует сигнал управления Read/Write, который определяет направление передачи данных (тип операции — чтение или запись информации). В этом случае на шине адреса ADDR_BUS размещается код адреса соответствующего периферийного устройства и формируется активный уровень сигнала стробирования адреса Address Strobe. Если выполняется операция записи (передачи данных от микропроцессора к контроллеру интерфейса I²C), на шине данных DATA_BUS микропроцессор размещает байт передаваемых данных и вырабатывает активный уровень сигнала стробирования данных Data Strobe.

В начале второй фазы контроллер осуществляет декодирование адреса, присутствующего на адресных входах, которые подключены к шине адреса ADDR_BUS. После этого при выполнении операции записи происходит

фиксация информации, представленной на шине данных DATA_BUS. При реализации операции чтения информация размещается контроллером на шине данных. Далее контроллер выработывает активный уровень сигнала подтверждения передачи данных Data Transfer Acknowledge.

Процесс чтения данных, поступающих от контроллера интерфейса I²C, завершается фиксацией информации на шине данных DATA_BUS, после чего снимается активный уровень сигналов стробирования данных Data Strobe и адреса Address Strobe. Если микропроцессором выполнялась операция записи, то после этого переданная информация удаляется с шины данных.

На заключительной фазе при выполнении операции чтения контроллер интерфейса I²C удаляет переданную информацию с шины данных DATA_BUS. Текущий цикл информационного обмена заканчивается переключением сигнала подтверждения передачи данных Data Transfer Acknowledge на выходе контроллера в неактивное состояние. После этого может стартовать следующий цикл передачи данных.

СТРУКТУРА АРХИВА, СОДЕРЖАЩЕГО ИСХОДНЫЕ ФАЙЛЫ ПРОЕКТА КОНТРОЛЛЕРА ИНТЕРФЕЙСА I²C

Исходные файлы проекта контроллера интерфейса I²C представлены в двух вариантах, оформленных в виде архивов харр385.zip и харр385_ver.zip, которые включают в себя описания на языках VHDL и Verilog соответственно. В данной статье рассматривается вариант, в котором все исходные описания контроллера и его компонентов выполнены на языке VHDL.

Архив файлов проекта контроллера интерфейса I²C харр385.zip, предоставляемый пользователю, содержит следующие группы модулей:

- модули исходных описаний устройства и его компонентов на языке VHDL;
- файл описания параметров проекта контроллера в САПР серии Xilinx ISE;
- файлы, предназначенные для верификации контроллера с помощью системы моделирования ModelSim;
- файлы, содержащие результаты компиляции исходных VHDL-описаний;
- отчеты о выполнении операций размещения и трассировки проекта контроллера в ПЛИС;

- файл программирования ПЛИС.

Описание контроллера интерфейса I²C на языке VHDL выполнено в виде иерархической конструкции. Исходное VHDL-описание интерфейса и структуры объекта верхнего уровня иерархии I²C, представляющего рассматриваемый контроллер, содержится в файле i2c.vhd. Данный объект включает в себя два компонента, которые соответствуют функциональным блокам обобщенной структурной схемы контроллера интерфейса I²C, приведенной на рис. 2.

VHDL-описание структурного модуля uc_interface, реализующего функции шинного интерфейса микропроцессорной системы, совместимой с микроконтроллерами фирмы Motorola серии 68000, оформлено в виде файла uc_interface.vhd. Описание структурного блока, выполняющего функции контроллера интерфейса I²C в ведущем и подчиненном режимах, содержится в файле i2c_control.vhd. Кроме указанных основных компонентов архив файлов проекта контроллера включает также VHDL-описание двух вспомогательных элементов: восьмиразрядного регистра сдвига с параллельной загрузкой и четырехразрядного двоичного счетчика с асинхронным сбросом и синхронной загрузкой. Эти элементы используются в описании основных компонентов. Описание регистра сдвига Shift8 содержится в файле shift.vhd, а счетчика upcnt4 — в файле upcnt4.vhd. Суммарный объем указанных исходных файлов проекта достаточно велик, поэтому текст VHDL-описаний контроллера и его компонентов не приводится в настоящей статье. Ознакомиться с ним можно после получения архива проекта харр385.zip.

Описание всех параметров проекта контроллера интерфейса I²C для САПР серии Xilinx ISE содержится в файле i2c.npl. Формат этого файла соответствует одной из ранних версий среды проектирования, поэтому при открытии проекта i2c.npl в управляющей оболочке Project Navigator САПР более поздней версии пользователю предлагается автоматически привести этот файл в соответствие с используемой версией системы проектирования.

Для осуществления функционального и полного (временного) моделирования в архив включены файлы, содержащие описание списка соединений (netlist) контроллера интерфейса I²C, VHDL-описания программных средств тестирования (испытательных стендов, testbench) и сценарии моделирования.

Файл i2c_timesim.vhd содержит описание списка соединений контроллера, предназначенный для выполнения полного (временного) моделирования. Описание программных средств тестирования, предназначенных для моделирования контроллера, имеет иерархическую структуру. В файле micro_test.vhd содержится описание модуля верхнего уровня иерархии тестовой системы, используемое для функционального моделирования. В этой системе в качестве компонентов используются объекты micro_tb, pullup и i2c, VHDL-описания которых находятся в одноименных файлах с расширением vhd. В состав тестовой системы входят два экземпляра контроллера интерфейса I²C. Один из них функционирует в ведущем режиме, второй — в подчиненном. Описание модели подтягивающих резисторов, используемых в тестовой системе, содержится в файле pullup.vhd. Описание модуля верхнего уровня иерархии программных средств тестирования, предназначенное для полного (временного) моделирования, находится в файле micro_test_post.vhd. В этом модуле в качестве компонентов применяются объекты micro_master_tb, micro_slave_tb и pullup, описания которых представлены в одноименных файлах с расширением vhd. Объекты micro_master_tb и micro_slave_tb описывают полные модели контроллера интерфейса I²C, функционирующего в ведущем и подчиненном режиме соответственно.

Сценарий функционального моделирования контроллера в рамках пакета ModelSim XE Starter, который содержит набор директив управления этим процессом и описание тестовых воздействий, представлен в командном файле micro_test.do. В этом сценарии используется командный файл wave.do, определяющий конфигурацию окна временных диаграмм системы ModelSim при функциональном моделировании рассматриваемого устройства. Для проведения полного (временного) моделирования, которое осуществляется после размещения и трассировки проекта контроллера в ПЛИС, предназначен файл сценария micro_test_post.do. Данный сценарий содержит обращение к командному файлу wave_post.do, определяющему конфигурацию окна временных диаграмм системы ModelSim при временном моделировании контроллера.

Результаты компиляции исходных VHDL-описаний контроллера, его компонентов и программных средств тес-

тирования, выполняемой средствами синтеза XST и системы моделирования ModelSim, расположены в подразделе work.

Информация о результатах размещения и трассировки проекта контроллера в ПЛИС XC2C256-5-TQ144 содержится в файле отчета i2c.rpt. Кроме того, в архив включен файл i2c.sxt, сформированный на этапе размещения и трассировки, в котором представлена исходная информация для оценки потребляемой мощности проектируемого устройства с помощью программы XPower. Данная программа входит в состав систем проектирования серии Xilinx ISE.

Конфигурационные данные в формате JEDEC, определяющие назначение ресурсов ПЛИС для реализации функций контроллера интерфейса I²C, содержатся в файле i2c.jed.

ИСПОЛЬЗОВАНИЕ ФАЙЛОВ ПРОЕКТА КОНТРОЛЛЕРА ИНТЕРФЕЙСА I²C

Для реализации контроллера интерфейса I²C на основе ПЛИС XC2C256-5-TQ144 в форме автономного устройства в том виде, как он предоставляется пользователю, достаточно файла конфигурационной битовой последовательности i2c.jed.

Данный файл предназначен для программирования ПЛИС, которое осуществляется с помощью программы iMPACT, входящей в состав средств проектирования серии Xilinx ISE, и загрузочного кабеля [4].

При необходимости пользователь может внести требуемые изменения в файлы исходного описания контроллера и его компонентов. В этом случае после редактирования исходных файлов следует открыть проект i2c.npl в САПР Xilinx ISE и последовательно выполнить процедуры синтеза, трансляции, размещения и трассировки проекта в ПЛИС. В результате будет сформирована новая версия конфигурационного файла i2c.jed. Кроме того, рекомендуется провести функциональное и временное моделирование проекта после внесения изменений, чтобы обнаружить возможные ошибки до программирования микросхемы.

Представленный контроллер может также использоваться как один из компонентов разрабатываемой системы, реализуемой на основе ПЛИС семейства CoolRunner-II. Для этого нужно включить в состав проекта системы модули исходных описаний контроллера и его компонентов, перечисленные в предыдущем разделе.

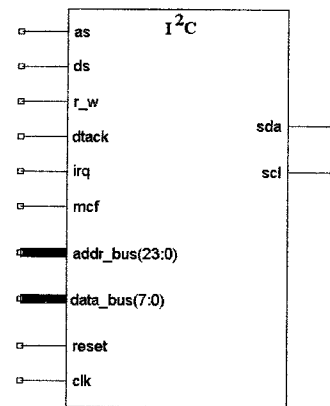


Рис. 4

Компонент I²C может использоваться как в текстовом (HDL), так и в схематехническом описании разрабатываемой системы. Условный графический образ (УГО) компонента, представляющего контроллер интерфейса I²C, может быть автоматически сформирован средствами САПР серии Xilinx ISE и при необходимости отредактирован пользователем. Один из вариантов УГО компонента I²C, предназначенный для использования в схематехническом редакторе Engineering Schematic Capture (ECS), показан на рис. 4.

Валерий Зотов,
г. Москва

ARGUSSOFT

http://components.argussoft.ru

ЗАО «АРГУССОФТ Компани»
ДЕПАРТАМЕНТ МИКРОЭЛЕКТРОНИКИ

ADuM130x/ADuM140x

Многоканальные изоляторы

- Двухнаправленная передача информации
- Замена оптронов и дискретных компонентов
- Снижение энергопотребления в 70 раз
- Скорость передачи данных до 100 Мбит/сек
- Температурный диапазон -40...+100°C

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

ANALOG DEVICES

Хотите попробовать?
Закажите бесплатные образцы у нас:

Москва
Тел.: (095) 217 2487, 217 2519
Факс: (095) 216 6642
E-mail: components@argussoft.ru

Санкт-Петербург
Тел.: (812) 567 1867
Факс: (812) 5671849
E-mail: spb@argussoft.ru

Новосибирск
Тел.: (3832) 27 1155
Факс: (3832) 22 4031
E-mail: nsk@argussoft.ru

www.platan.ru

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ ОТ ВЕДУЩИХ ПРОИЗВОДИТЕЛЕЙ

Автомобильные реле серии ETR-AD1

- используются при установке сигнализаций, иммобилайзеров, центральных замков, стеклоподъемников и других устройств автоматики и защиты в автомобилях всех типов
- имеют кронштейн для крепления к кузову автомобиля

Основные электрические параметры:

Коммутируемое напряжение/ток: 12 В / 40 А

Напряжение обмотки: 12 В

Потребляемая мощность: 1,8 Вт

Макс. кол-во циклов переключения: 10⁵ x 10⁵

Конфигурация контактов: SPDT (одна группа на переключ.)

Контактная колодка **ETR441** для подключения и установки реле серии **ETR-AD1**.

Москва, ул. Ивана Франко, д. 40, стр. 2
Тел./факс: (095) 73-75-999

Пента: 121351, Москва, в/п 100
E-mail: platan@aha.ru

Операционные усилители фирмы ROHM

Фирма ROHM выпускает широкий спектр полупроводниковых приборов, пассивных компонентов, радиоэлектронных модулей и электронно-оптических приборов для различных применений. В этой статье рассматриваются операционные усилители этой фирмы.

Номенклатура ОУ фирмы ROHM составляет восемь типов, каждый из которых выпускается в различных корпусах. Эти ОУ по электрическим параметрам и цоколевке соответствуют ОУ других производителей, ставших фактически стандартными приборами, но имеют меньшую стоимость.

Микросхемы BA10358 и BA10324A можно отнести к относительно мощным ОУ.

Первая из них содержит в корпусе два ОУ, допускает питание как от двуполярного источника $\pm 1,5 \dots \pm 16$ В, так и от однополярного $+3 \dots +32$ В, и оформляется в корпусах четырех типов — DIP8 (BA10358), SOP8 (BA10358F), SSOP-B8 (BA10358FV), SIP8 (BA10358N). Эта микросхема совместима с широко выпускаемыми различными фирмами ОУ LM358, с другими, содержащими в своем обозначении цифры 358, а также с КР1040УД1. Диапазон входных синфазных напряжений ОУ при однополярном питании составляет от нуля до напряжения, на 1,5 В меньше напряжения питания, такой же диапазон и у выходного напряжения.

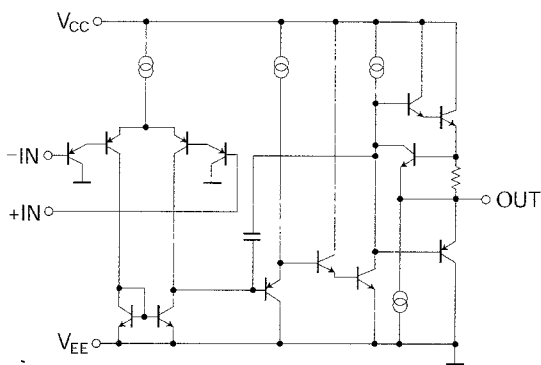


Рис. 1

Этот ОУ, как и все рассматриваемые далее, содержит цепи внутренней фазовой коррекции. Упрощенная принципиальная схема одного ОУ этой микросхемы приведена на рис. 1.

Электрические параметры ОУ микросхемы BA10358 и других, рассматриваемых далее, приведены в табл. 1. В таблице приняты следующие обозначения:

- N — число ОУ в корпусе;
- $K_{у\text{ мин}}$ — коэффициент усиления ОУ;
- $U_{\text{пит}}$ — диапазон питающих напряжений;
- $I_{\text{пит макс}}$ — максимальный ток потребления микросхемы;
- $U_{\text{см}}$ — типовое (максимальное) напряжение смещения нуля;
- $K_{\text{ос.сф}}$ — коэффициент ослабления входного синфазного напряжения;
- $K_{\text{вл.ип}}$ — коэффициент влияния нестабильности источников питания на напряжение смещения;

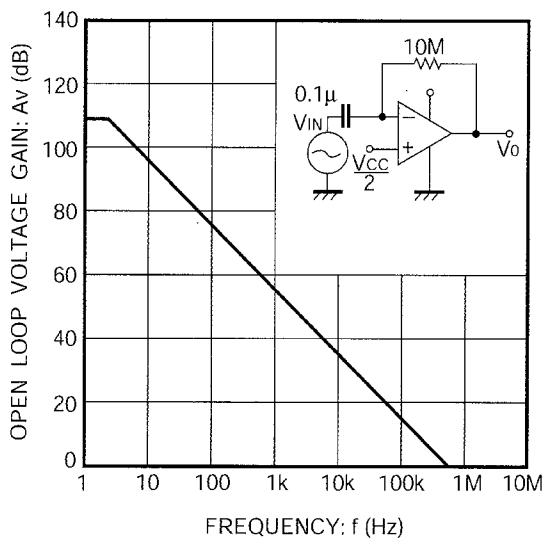


Рис. 2

- $I_{\text{вх}}$ — максимальный входной ток;
- f_1 — частота единичного усиления;
- $V_{\text{убых}}$ — максимальная скорость нарастания выходного напряжения;
- $U_{\text{ш}}$ — шумовое напряжение, приведенное к входу.

На рис. 2 приведена зависимость малосигнального коэффициента усиления ОУ от частоты, на рис. 3 — амплитудно-частотная характеристика ОУ на большом сигнале, на рис. 4 — зависимость разности между напряжением положительного источника питания и выходным от выходного вытекающего тока, на рис. 5 — выходное напряжение в зависимости от выходного втекающего тока.

Микросхема BA10324A содержит в корпусе четыре ОУ, выпускается в корпусах трех типов — DIP14 (BA10324A), SOP14 (BA10324AF), SSOP-B14 (BA10324AFV). Эта микросхема совместима с ОУ LM324 и К1401УД2. Основные свойства микросхемы соответствуют BA10358, есть некоторые отличия в параметрах (см. табл. 1).

К быстродействующим можно отнести микросхемы BA4560, BA15218, BA15532 и BA4510. В корпусе каждой микросхемы — по два ОУ. Все ОУ обладают высокой скоростью нарастания

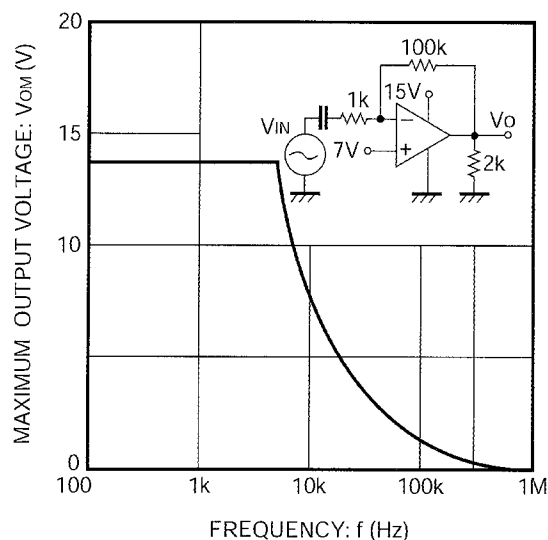


Рис. 3

Таблица 1

Тип ОУ	N	$K_{у\text{ мин}}$ ДБ	$U_{\text{пит}}$ В	$I_{\text{пит макс}}$ МА	$U_{\text{см}}$ мВ	$K_{\text{ос.сф}}$ ДБ	$K_{\text{вл.ип}}$ ДБ	$I_{\text{вх}}$ нА	f_1 МГц	$V_{\text{убых}}$ В/мкс	$U_{\text{ш}}$ мкВ	Число выводов
BA10358	2	87	3...32	1,2	2 (7)	80	100	250	0,5	0,2	—	8
BA10324A	4	87	3...32	2	2 (7)	75	100	250	0,25	0,2	—	14
BA4560	2	87	$\pm 4 \dots \pm 18$	4	0,5 (6)	90	90	500	10	4	2,2	8
BA15218	2	87	$\pm 2 \dots \pm 16$	8	0,5 (5)	90	90	500	10	3	1	8
BA15532	2	80	$\pm 3 \dots \pm 20$	16	0,5 (4)	100	100	800	20	8	1,5	8
BA4510	2	60	$\pm 1 \dots \pm 3,5$	7,5	1 (6)	80	80	500	—	5	—	8
BA14741	4	86	$\pm 2 \dots \pm 18$	7	1 (5)	100	100	300	2	1	4	14
BA4558	2	86	$\pm 4 \dots \pm 18$	4	0,5	90	90	500	—	1	—	8

выходного напряжения. Микросхемы характеризуются низким уровнем шума и высоким быстродействием.

Первые три микросхемы (BA4560, BA15218, BA15532) выпускаются в корпусах DIP8 (без суффикса), SOP8 (суффикс F), SIP8 (суффикс N) и находят широкое применение в звуковой аппаратуре, активных фильтрах и многих других случаях.

Упрощенная принципиальная схема одного ОУ микросхемы BA4560 показана на рис. 6. Для этой микросхемы входное синфазное напряжение не должно доходить ближе, чем на 3 В до уровней напряжения питания, примерно такой же диапазон выходного напряжения. На рис. 7 и 8 приведены зависимость малосигнального коэффициента усиления ОУ от частоты и амплитудно-частотная ха-

рактеристика ОУ на большом сигнале соответственно.

Свойства ОУ BA15218, BA15532 и их принципиальные схемы близки к BA4560, различаются они, в основном, электрическими параметрами.

Микросхема BA4510 выпускается в корпусах двух типов — SOP8 (BA4510F) и SSOP-B8 (BA4510FV). Ее параметры заметно отличаются от других рассматриваемых здесь приборов (табл. 1). Принципиальная схема одного ОУ приведена на рис. 9. Эта микросхема может работать при низком напряжении питания — $\pm 1 \dots \pm 3,5$ В, для нее входное синфазное напряжение не должно доходить ближе, чем на 1 В до уровней напряжения питания, выходное не доходит до этих уровней примерно на 0,5 В.

Стратегические инженерные решения

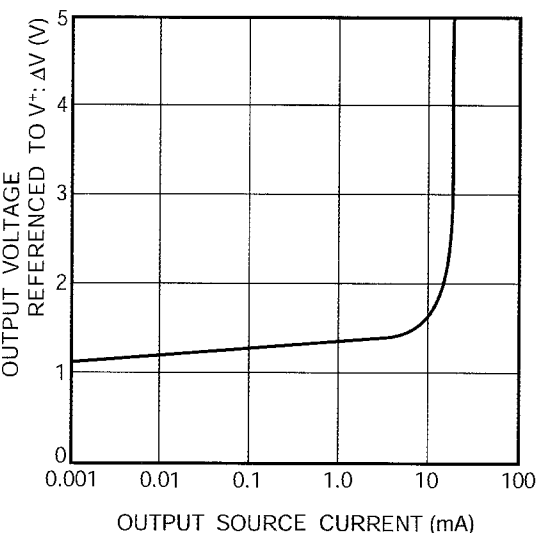
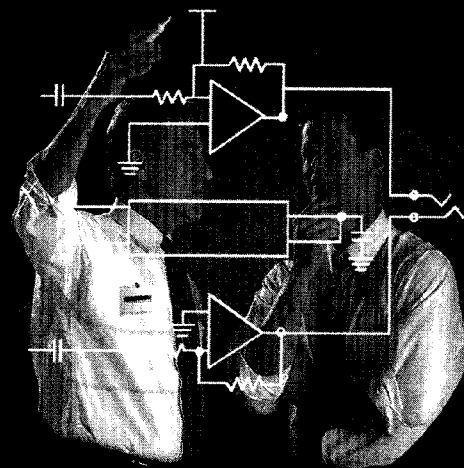


Рис. 4

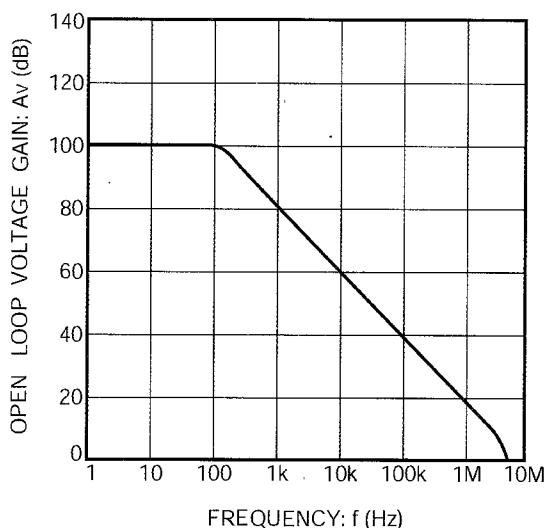


Рис. 7

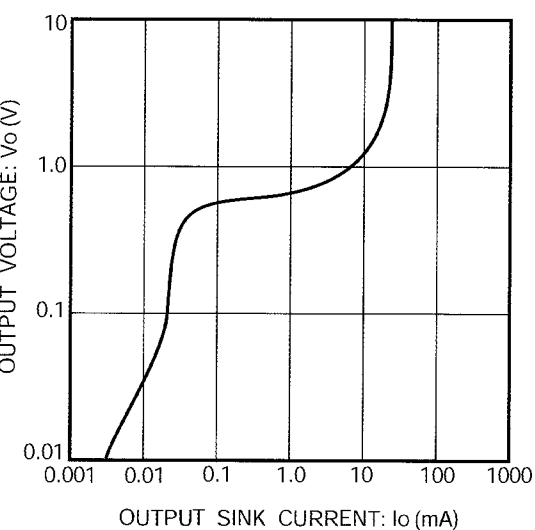


Рис. 5

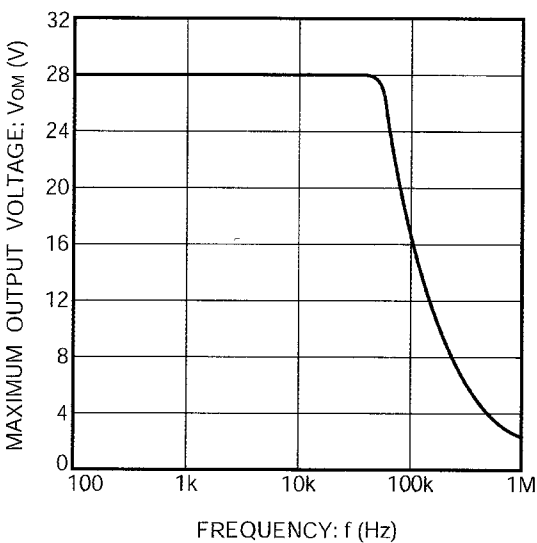


Рис. 8

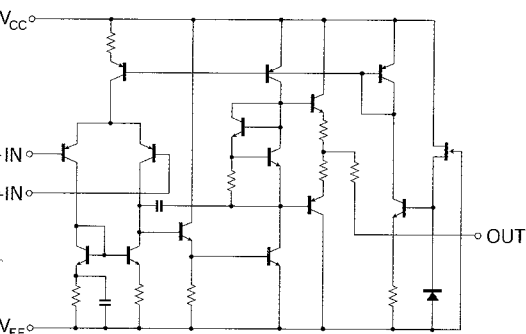


Рис. 6

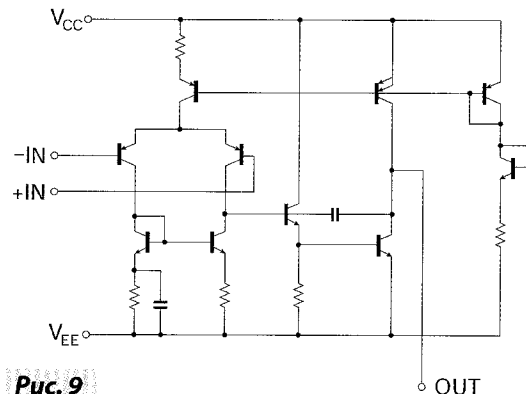


Рис. 9

- Мощные
- Быстродействующие
- Маломощные
- Многоканальные ОУ

с возможностью коммутации каналов

Связь и телекоммуникации

Энергетика

Транспорт

Торговля

Доступ и защита информации

Промышленность

Медицина

Информационные технологии

RAINBOW TECHNOLOGIES
www.rics.ru
www.ibutton.ru
www.rainbow.by

Москва (095) 797-8993, info@rainbow.msk.ru
 Санкт-Петербург (812) 324-0902, spb@rainbow.msk.ru
 Минск (+37517) 249-8273, chip@rainbow.by
 Киев (+38044) 490-9228, chip@rainbow.com.ua
 Екатеринбург (343) 269-4984, ural@rainbow.msk.ru
 Новосибирск (3832) 270-810, nsk@rainbow.msk.ru

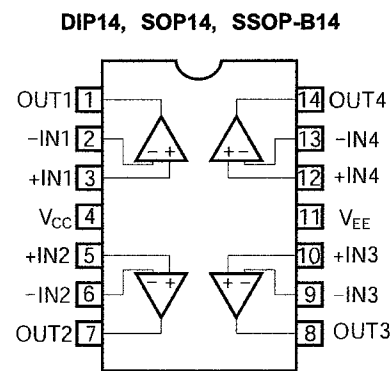
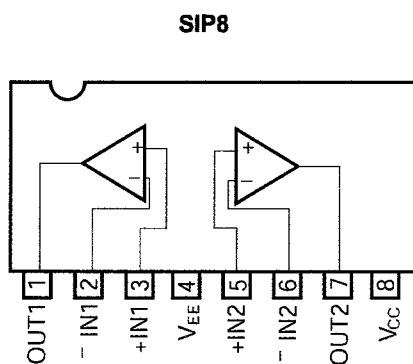
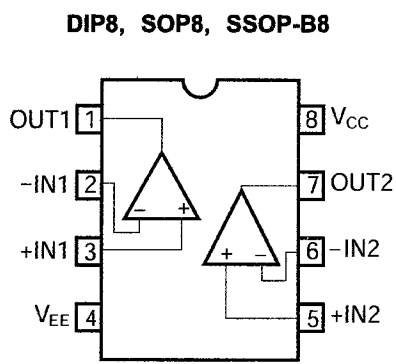


Рис. 10

К малошумящим усилителям можно отнести ОУ ВА14741, ВА4558.

Микросхема ВА14741 содержит четыре ОУ, совместимых с давно привычными μ A741 (К140УД7), она выпускается в корпусах DIP14 (ВА14741) и SOP14 (ВА14741F).

Два ОУ микросхемы ВА4558 обладают несколько лучшими параметрами (см. табл. 1). Эта микросхема выпускается в корпусах DIP8 (ВА4558), SOP8 (ВА4558F), SIP8 (ВА4558N). Принципиальная схема ОУ этой микросхемы соответствует ВА4560 (рис. 6).

У обеих из рассмотренных выше микросхем входной синфазный сигнал не должен подходить к уровням напряжений питания ближе, чем на 3 В.

Разводка выводов рассмотренных микросхем в корпусах восемью выводами приведена на рис. 10, с 14-ю

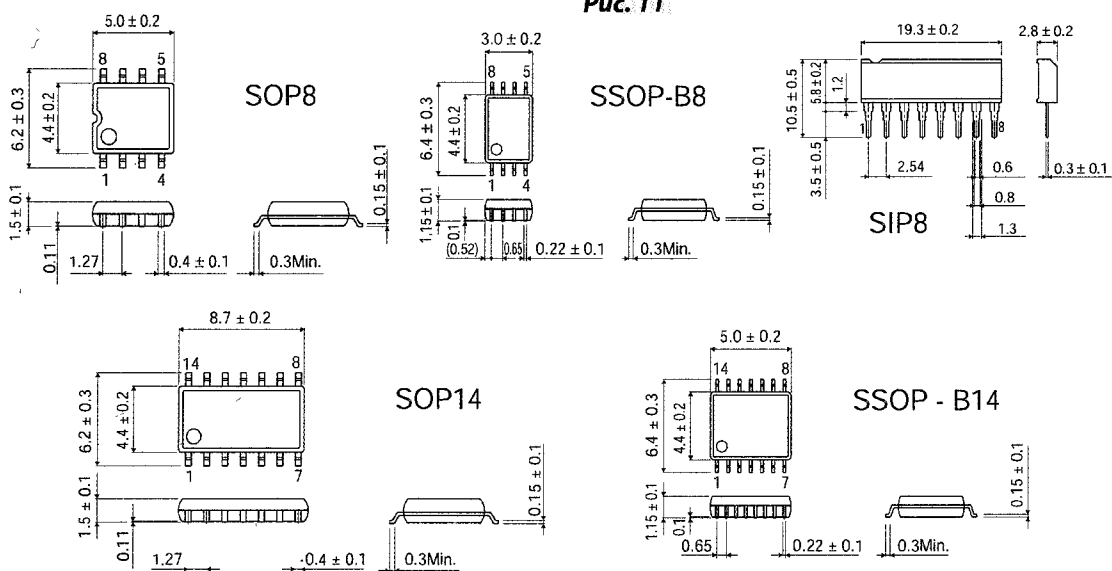


Рис. 12

выводами — на рис. 11, чертежи корпусов — на рис. 12.

Подробные сведения по микросхемам фирмы ROHM можно найти на сайте официального дистрибьютора этой фирмы — компании Rainbow

Рис. 11

<http://www.rctcs.ru> и на сайте фирмы-изготовителя <http://www.rohm.com>.

Сергей Алексеев,
г. Москва

ARGUSSOFT ЗАО «АРГУССОФТ Компани»
ДЕПАРТАМЕНТ МИКРОЭЛЕКТРОНИКИ
<http://components.argussoft.ru>

Новые E+ IGBT модули
1200 и 1700 Вольт
до 440 Ампер

IXYS

- NPT3 кристаллы с улучшенными характеристиками
- уменьшенное прямое падение V_{ces}
- сокращение потерь переключения
- рабочая частота до 20 КГц
- уменьшенное значение E_{off}
- стандартный корпус на медном основании со встроенным термодатчиком

Москва (095) 217 2437 217 2519
Санкт-Петербург (812) 567 1867
Екатеринбург (343) 378 3242
Новосибирск (3832) 27 1159
Факс: (095) 216 6642
E-mail: component@argussoft.ru spb@argussoft.ru ural@argussoft.ru nsk@argussoft.ru

ПЛАТАН www.platan.ru
ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ ОТ ВЕДУЩИХ ПРОИЗВОДИТЕЛЕЙ

Выпрямительные диоды
Выпрямительные диоды

- Диоды со стандартным временем восстановления в корпусах от SMD до Puk
- Лавинные диоды
- Снабберные диоды
- Диоды Шоттки с улучшенными электрическими параметрами (низкий ток утечки, низкое прямое падение напряжения, широкий температурный диапазон)
- Быстрые и ультрабыстрые диоды
- Мощные диоды на напряжение до 4500 В, 8000 А
- Диоды для FPC приложений, сварочной техники, телекоммуникаций

MAXIM DALLAS
Honeywell
EPCOS
IOR
MITSUBISHI ELECTRIC
PHILIPS
Infineon
BOURNS
ANALOG DEVICES
VISHAY
muRata
AMP
CRYDOM
DATA VISIOP
ST
Kingbright

Москва, ул. Ивана Франко, д. 40, стр. 2
Тел./факс: (095) 73-75-999
Почта: 121351, Москва, а/я 100
E-mail: platan@aha.ru

Цифровые сигнальные процессоры

Рассмотрим основные директивы ассемблера, которые применяются в текстах программ для сигнальных процессоров семейства ADSP.

Директива MODULE

Директива MODULE обозначает начало программного модуля и определяет его название. Как было сказано ранее, программа для ADSP-2181 может состоять из одного или нескольких модулей, которые ассемблируются отдельно и затем связываются вместе. Каждый файл с исходным текстом программы может содержать только один программный модуль. Директива имеет следующую форму записи:

```
.MODULE/ПАРАМЕТР1/ПАРАМЕТР2 ... ИМЯ_МОДУЛЯ;
```

Параметры могут быть следующими:

- RAM или ROM — тип памяти для размещения программы;
- ABS = адрес — абсолютный стартовый адрес (не используется вместе с директивой STATIC);
- SEG = сегмент — размещение программы в указанном сегменте;
- STATIC — статичное размещение модуля в памяти (не используется вместе с директивой ABS);

Параметр RAM (ОЗУ) или ROM (ПЗУ) определяет тип памяти процессора для размещения программы. Если этот тип не определен, по умолчанию принимается RAM.

Параметр ABS размещает коды модулей программ по определенным адресам памяти программ, что делает их перемещаемыми компоновщиком. Это означает, что компоновщик резервирует память для модулей по указанным адресам. Модули, которые не имеют параметра ABS, распределяются компоновщиком в памяти процессора автоматически.

Параметр SEG размещает модуль в сегмент памяти, который объявлен в файле системной конфигурации. Если определить оба параметра ABS и SEG и указать абсолютные адреса, которых нет в данном сегменте, компоновщик выдаст сообщение об ошибке.

Параметр STATIC сохраняет модуль в памяти программ, когда загружаются страницы начальной загрузки. Пример записи директивы:

```
.MODULE/RAM/ABS=0 demo;
{Определение имени demo программного модуля}
{с размещением в ОЗУ по абсолютному адресу 0}
```

Директива CONST

Директива CONST определяет соответствие между именем константы и ее значением. Данная директива имеет следующую форму записи:

```
.CONST имя_константы1 = значение1, имя_константы2 = значение2...;
```

Значения констант могут быть числовыми или символьными. Кроме того, допускается присваивать константам выражения. В выражениях разрешаются только арифметические или логические операции над двумя или более целыми константами. В качестве знаков выражений в ассемблере допускается использовать следующие символы:

- + — сложение;
- - — вычитание;
- × — умножение;
- / — деление;
- % — модуль (целая часть от деления);
- (— левая скобка;
-) — правая скобка;
- & — логическое И (AND);
- | — логическое ИЛИ (OR);
- ~ — исключающее ИЛИ (XOR);
- << — логический сдвиг влево;
- >> — логический сдвиг вправо.

Запись чисел в программах может осуществляться в нескольких форматах. Для записи шестнадцатеричных чисел используется префикс 0x (ноль и икс) или H#. Например:

```
0x24FF, H#CF8A.
```

Для восьмеричных чисел применяется префикс 0 (ноль):

```
0777, 0123, 07777.
```

Двоичные числа записываются с префиксом B#:

```
B#01110100.
```

Десятичные числа записываются в программе без префиксов, т. е. по умолчанию формат числа считается десятичным:

```
1024,-55,0.
```

Пример записи арифметического выражения:

```
X = (29+129)×(128-48)/3.
```

Пример логического выражения:

```
Y = 0x55&0x0F.
```

Пример записи директивы:

```
.CONST N=1024, LEN_BUF=N/2; {Число точек отсчета = 1024, длина буфера = 512}
```

Директива VAR

Директива VAR объявляет переменные и буферы переменных. Буфер переменных представляет собой набор ячеек памяти, расположенных упорядоченно друг за другом по смежным (соседним) адресам. Буфер может быть многомерным и одномерным, т. е. состоящим из нескольких или из одной переменной, он должен быть объявлен до использования в программе. Буферы удобны для организации приема и передачи блочных данных и часто используются в программах.

Директива VAR имеет следующую форму записи:

```
.VAR/параметр1/параметр2 ... имя_буфера[длина], ... ;
```

С директивой могут использоваться следующие параметры:

- PM или DM — тип памяти для размещения буфера;
- RAM или ROM — тип памяти процессора;
- ABS = адрес — абсолютный адрес (нельзя использовать вместе с директивой STATIC);
- SEG = сегмент — размещение буфера в сегменте, объявленном системным конфигуратором;
- CIRC — кольцевой буфер;
- STATIC предотвращает перезапись буфера во время загрузки начальной страницы.

Одна директива VAR позволяет объявить в одной строке длиной до 200 символов несколько буферов, разделенных запятыми.

Буферы могут быть линейными и кольцевыми, они могут быть произвольной длины — от единицы до

размера, не превышающего объем памяти процессора. Линейный буфер размещается в памяти с любого адреса, кольцевой — с некоторыми ограничениями, связанными с аппаратными особенностями памяти процессоров. Так, кольцевой буфер должен стартовать с базового адреса, который кратен 2^n , где n — число бит, требуемых для представления длины буфера в двоичном виде. Это необходимо учитывать при создании кольцевых буферов с параметром ABS.

При многочисленном объявлении линейных буферов в одной строке редактор связей размещает их в смежных областях памяти. Если при таком объявлении используется параметр CIRC, то создается единственный кольцевой буфер, а остальные объявленные в этой строке буферы будут линейными.

Буферы могут быть размещены как в памяти программ PM, так и в памяти данных DM. Тип памяти по умолчанию устанавливается в RAM.

Параметр ABS размещает буфер с указанного стартового адреса, делая его непеременяемым. Параметр SEG размещает буфер в указанном сегменте памяти, который был объявлен в файле системного конфигурирования.

Параметр CIRC определяет кольцевой буфер. Буфер будет организован как линейный, если не применен атрибут CIRC. Параметр STATIC предотвращает перезапись буфера, когда загружается страница начальной загрузки.

Пример объявления буфера:

```
.VAR/PM/RAM/SEG=segdata buffer[10];
```

Здесь линейный буфер объявлен в памяти программ RAM, перемещаемый в пределах сегмента с именем segdata. Буфер с именем buffer состоит из десяти ячеек в памяти программ. Длина буфера указывается в квадратных скобках.

Для объявления одномерных буферов, т. е. простых переменных, применяется директива VAR без указания длины буфера. Например:

```
.VAR/DM/RAM/ABS=0x000C xdata;
```

Данная директива объявляет однословную переменную с именем xdata в памяти данных RAM по шестнадцатеричному адресу 0x000C.

Ниже приведен пример объявления автоматически перемещаемого кольцевого буфера с именем buffer_c, длина которого определяется константой size, объявленной директивой CONST.

```
.CONST size = 15;
.VAR/DM/CIRC buffer_c[size];
```

Ниже приведены примеры, позволяющие понять, как можно размещать циклические буферы в памяти. Следующий оператор объявляет кольцевой буфер с именем abuf из шести ячеек памяти:

```
.VAR/CIRC abuf[6];
```

Поскольку ближайшим числом больше шести и кратным двум является число восемь, то базовый (стартовый) адрес буфера должен быть кратен восьми. Три младших значимых разряда (МЗР) этого адреса будут равны нулю.

Если в одной строке объявлены несколько буферов с параметром CIRC, то, как указывалось выше, будет создан один кольцевой буфер, вмещающий в себя все буферы. Первый из них будет кольцевым, а отдельные — простыми линейными. Например, следующая директива создает один кольцевой буфер abuf и два линейных буфера с именами bbuf и cbuf:

```
.VAR/CIRC abuf[6], bbuf[3], cbuf[5];
```

Для размещения всех этих буферов в памяти процессора потребуется $6+3+5 = 14$ ячеек памяти. Поскольку первым в директиве объявлен буфер abuf, он будет кольцевым. Ближайшим числом, большим 14 и кратным двум, является число 16, поэтому базовый адрес буфера abuf должен быть кратен 16 (четыре младших значащих разряда будут равны нулю). Базовый адрес буфера bbuf будет следовать сразу за адресом последнего элемента буфера abuf. Аналогично будет размещен и буфер cbuf. Размещение этих буферов в памяти процессора показано в табл. 1.

Следующий пример показывает использование трех директив для объявления трех различных кольцевых буферов:

```
.VAR/CIRC abuf[6];
.VAR/CIRC bbuf[3];
.VAR/CIRC cbuf[5];
```

Поскольку буферы объявлены отдельно, все они будут кольцевыми и не будут объединены. Правила размещения для каждого из этих буферов такие же, как в рассмотренном выше примере для одиночного буфера abuf. Размещение этих буфе-

Таблица 1

Имя буфера	Элемент буфера	Двоичный адрес
—	—	...
abuf	abuf[0]	XXXXXXXXXX0000
—	abuf[1]	XXXXXXXXXX0001
—	abuf[2]	XXXXXXXXXX0010
—	abuf[3]	XXXXXXXXXX0011
—	abuf[4]	XXXXXXXXXX0100
—	abuf[5]	XXXXXXXXXX0101
bbuf	bbuf[0]	XXXXXXXXXX0110
—	bbuf[1]	XXXXXXXXXX0111
—	bbuf[2]	XXXXXXXXXX1000
cbuf	cbuf[0]	XXXXXXXXXX1001
—	cbuf[1]	XXXXXXXXXX1010
—	cbuf[2]	XXXXXXXXXX1011
—	cbuf[3]	XXXXXXXXXX1100
—	—	XXXXXXXXXX1101
—	—	...

ров в памяти процессора показано в табл. 2.

Таблица 2

Имя буфера	Элемент буфера	Двоичный адрес
—	—	...
abuf	abuf[0]	XXXXXXXXXX0000
—	abuf[1]	XXXXXXXXXX0001
—	abuf[2]	XXXXXXXXXX0010
—	abuf[3]	XXXXXXXXXX0011
—	abuf[4]	XXXXXXXXXX0100
—	abuf[5]	XXXXXXXXXX0101
—	—	...
bbuf	bbuf[0]	XXXXXXXXXX0100
—	bbuf[1]	XXXXXXXXXX0101
—	bbuf[2]	XXXXXXXXXX0110
—	—	...
cbuf	cbuf[0]	XXXXXXXXXX1000
—	cbuf[1]	XXXXXXXXXX1001
—	cbuf[2]	XXXXXXXXXX1010
—	cbuf[3]	XXXXXXXXXX1011
—	—	...

Чтобы получить доступ к буферу abuf из программы, необходимо инициализировать индексные регистры DAG и регистры длины буфера следующими инструкциями:

```
I0 = ^abuf; {Присвоить индексному регистру I0 базовый адрес буфера abuf}
L0 = %abuf; {Присвоить L0 длину буфера abuf}
M0 = 1; {Присвоить модификатору M0 значение наращивания указателя адреса буфера abuf}
```

Далее в программе можно читать данные из буфера с помощью инструкции AR = DM(I0, L0) или записывать данные в буфер инструкцией DM(I0, L0) = AR. Вместо регистра AR могут применяться и некоторые другие регистры процессора. При размещении этих инструкций в цикле каждое обращение к буферу будет автоматически увеличивать индексный регистр адреса I0 на величину M0. При достижении регистром I0 величины, равной L0, индексный регистр I0 вновь загрузится базовым адресом

буфера. Таким образом, обращения к буферу будут кольцевыми. Для линейных буферов регистр длины буфера должен быть установлен в нуль.

Директива PMSEG и DMSEG

Данные директивы предназначены для размещения программ и данных в сегментах памяти. Директива PMSEG указывает редактору связей на необходимость разместить все программы и данные модуля в определенном сегменте памяти программ, а DMSEG — в определенном сегменте памяти данных. Сегменты должны быть предварительно определены в файле описания архитектуры системного конфигуризатора. Директивы PMSEG и DMSEG подобно параметру SEG директив MODULE и VAR имеют следующий формат:

```
.PMSEG имя_сегмента_pm;
.DMSEG имя_сегмента_dm;
```

Обычно чтобы расположить все программы и данные исходного модуля в определенном системным конфигуризатором сегменте памяти, необходимо повторить параметр SEG в директиве MODULE и всех директивах VAR внутри модуля. Директивы PMSEG и DMSEG используются для исключения многократного повторения параметров SEG и должны быть размещены в исходном файле программного модуля перед директивой MODULE. Ниже приводится пример, в котором модуль prog1, некоторые буферы и переменную var1 располагают в памяти данных в сегменте с именем seg1:

```
.DMSEG seg1;
MODULE/RAM prog1;
.VAR/DM/RAM/CIRC buf1[15];
.VAR/DM/RAM buf2[5];
.VAR/DM/RAM buf3[5];
.VAR/DM/RAM var1;
...
.ENDMOD;
```

Директива INIT

Директива INIT используется для инициализации переменных и буферов в ПЗУ. Редактор связей помещает данные инициализации в файл образа памяти, который затем используется разделителем программ (splitter) при подготовке данных для записи в ПЗУ. Разделитель трансформирует части этого файла в формат, совместимый с промышленным стандартом программатора ПЗУ.

Инициализирующие значения могут быть перечислены в директиве или указаны во внешнем файле. Директива INIT может иметь одну из следующих форм:

```
.INIT имя_буфера: значение1, значение2,
...;
.INIT имя_буфера: ^другой_буфер или %другой_буфер, ...;
.INIT имя_буфера: <имя_файла>;
```

Операторы ^ и % используются для инициализации буфера или переменной базовым адресом или длиной, либо другими буферами. Любые комбинации констант, указателей адресов буфера и величин длины буфера могут быть заданы через запятую.

Данный пример инициализирует переменную x шестнадцатеричной константой 0x3FFF:

```
.INIT x: 0x3FFF;
```

Следующая директива инициализирует буфер buf списком констант:

```
.INIT buf: 9,0,3,5,7;
```

А здесь переменная ab инициализируется указателем стартового адреса буфера buf:

```
.INIT ab: ^buf;
```

Допускается инициализировать только часть данных буфера, задавая смещение его базового адреса (индекса):

```
.INIT buf[2]: 3,5,7;
```

В результате инициализирующие величины будут размещены, начиная с элемента buf[2]. Здесь инициализируются второй, третий и четвертый элементы буфера buf значениями 3, 5 и 7 соответственно.

Третья форма директивы INIT указывает имя файла, который содержит инициализирующие величины. Ассемблер устанавливает указатель на этот файл, и данные присоединяются при запуске редактора связей. Следующий пример заставляет редактор связей инициализировать буфер sin содержимым файла sinus.dat:

```
.INIT sin: <sinus.dat>;
```

Если файл с данными находится в директории с программой, то необходимо указать в скобках только имя этого файла. Если файл находится в

другом каталоге, необходимо указать путь к этому каталогу и имя файла. Например, если файл init.dat для буфера с именем buff размещен в директории C:\ADSP2181\PROG1, тогда директива INIT должна быть применена следующим образом:

```
.INIT buff: < C:\ADSP2181\PROG1\init.dat>
```

Это позволит редактору связей найти файл. Данный способ широко используется для загрузки буферов данными, выработанными другими программами, такими, например, как нахождение коэффициентов фильтра. После того, как редактор связей считывает и присоединит содержимое файла, изменение данных потребует лишь выполнить повторную компоновку программы.

Переменные данных и буферов могут быть инициализированы с помощью семиразрядного кода ASCII. Следующий пример инициализирует один буфер input кодами ASCII для букв A, E, F, Z. ASCII коды размещаются в семи младших разрядах 16-разрядной памяти данных или 24-разрядной памяти программ. Символы необходимо заключать в апострофы.

```
.INIT input: 'AEFZ'; {Инициализировать буфер символами ASCII}
```

Специальный синтаксис директивы INIT24 позволяет сохранять 24-разрядные данные в памяти программ. Это дает возможность получить доступ к младшим восьми разрядам каждого 24-разрядного слова памяти программ при инициализации буферов данных или переменных в исходной программе.

Например, эта директива позволяет вычислить 16-разрядный адрес для переменной var:

```
.INIT var: ^buff + 17;
```

Следующая директива вычисляет 24-разрядный адрес для этой же переменной:

```
.INIT24 var: ^buff + 17;
```

Продолжение следует

Олег Вальпа,
г. Миасс Челябинской обл.

Методы анализа операционных схем в частотной области

3.4. Детальный анализ ПФ2

Пример 9. На рис. 39 изображена схема ПФ2 [7, 8], который в случае использования в нем идеального ОУ, характеризуется следующими параметрами:

- центральная частота полосы пропускания F_0 10 кГц;
- полоса пропускания при неравномерности АЧХ 3,0103 дБ $F_0/1,05...1,05 \times F_0$;
- коэффициент передачи на центральной частоте полосы пропускания H_{BP2} 9,992226;
- добротность Q 10,2439;
- частота полюса F_p 10 кГц.

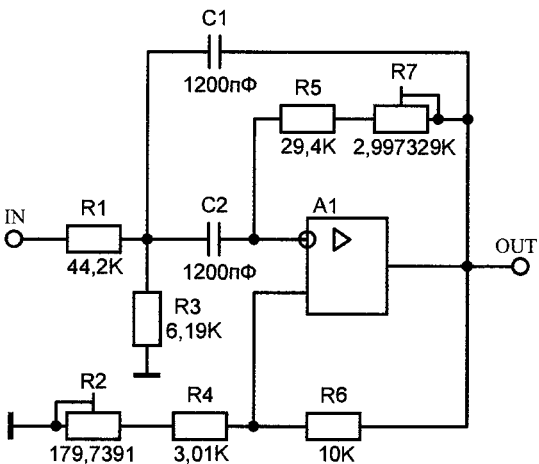


Рис. 39

В процессе всестороннего частотного анализа ОПС необходимо:

- выявить наиболее критичные параметры компонентов ПФ2, в наибольшей степени влияющих на нестабильность его параметров;
- дать рекомендации по повышению стабильности параметров ПФ2;
- сформулировать обоснованные требования к параметрам ОУ A1;
- получить числовые значения или графические зависимости всех основных параметров ПФ2, позволяющие прогнозировать его поведение при изменении условий эксплуатации.

В ПФ2 используется частотно-независимая ПОС (делитель R2R4R6) оптимальной глубины, при которой минимизируется активная чувствительность схемы. Однако использование ПОС делает фильтр потен-

циально неустойчивым и поэтому целесообразно на первом этапе анализа исследовать его устойчивость. На рис. 40 изображена схема анализа частотных свойств петлевого усиления ПФ2, полностью соответствующая рассмотренной выше функциональной схеме на рис. 38.

В схеме используется ММ ОУ ОРА_2P_PhMar (табл. 5) с типичными величинами параметров. Следует отметить, что фактические сопротивления подстроечных резисторов R2, R2a, R7, R7a, включенных по остататной схеме, равны их номиналу, умноженному на величину атрибута A, указанную на схеме рядом с соответствующим резистором ($A = 0...1$ — относительный угол поворота движка подстроечного резистора). В результате многовариантного анализа (варьировался параметр FT в диапа-

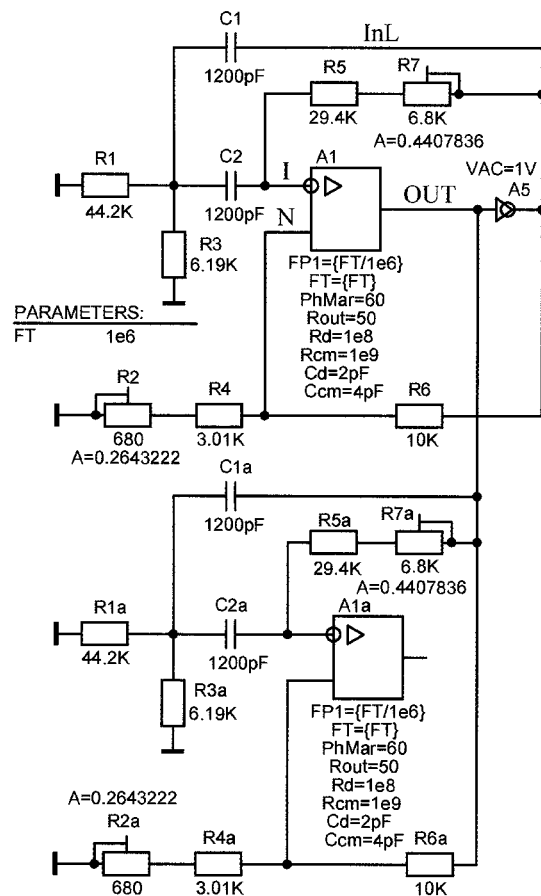


Рис. 40

зоне значений 0,2...2 МГц) получены зависимости (рис. 41):

- запаса устойчивости по фазе ОПС от частоты единичного усиления ОУ $P_M = f(F_T)$;

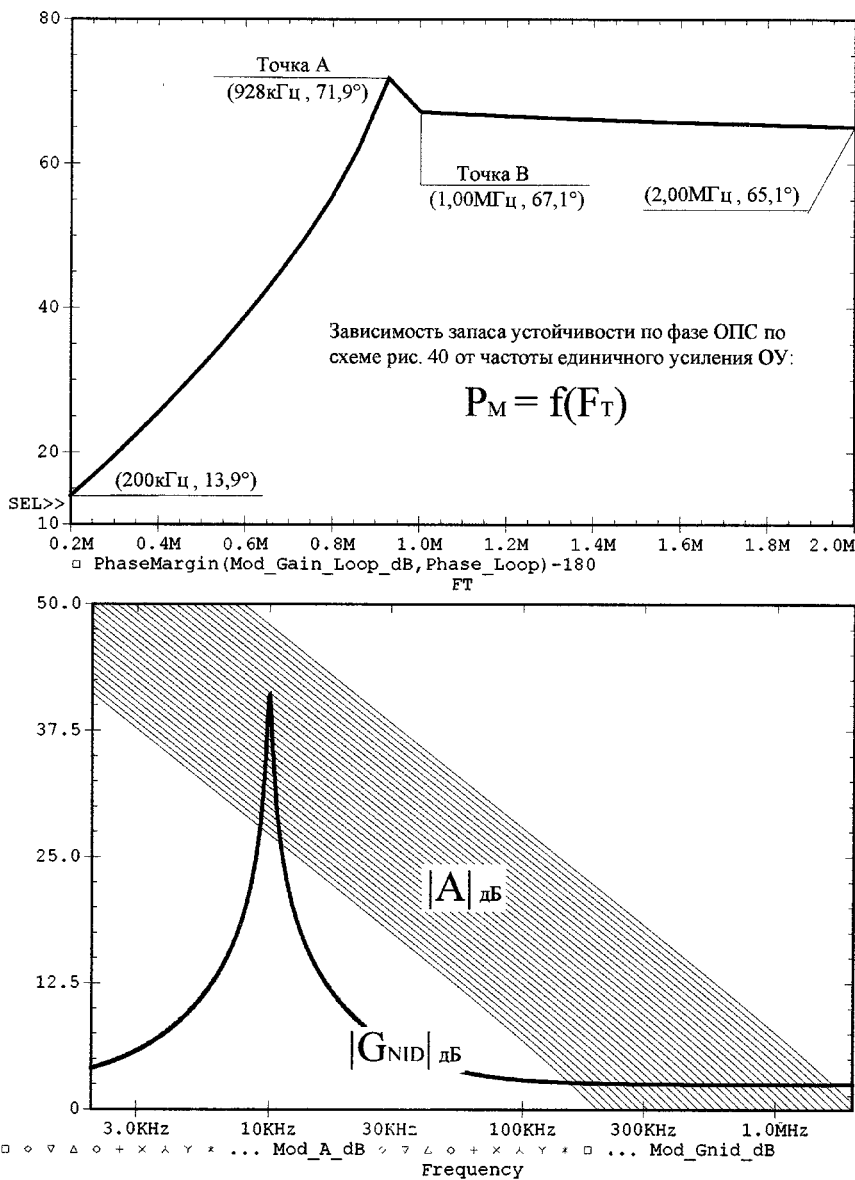


Рис. 41

- семейство ЛАЧХ ОУ $|A|_{дБ}$;
- ЛАЧХ передаточной функции $G_{NID}(S) |G_{NID}|_{дБ}$.

Для построения параметрической зависимости $P_M = f(F_T)$ используется целевая функция PhaseMargin, фактические параметры (трассы) которой задаются макросами Mod_Gain_Loop_dB и Phase_Loop (табл. 10). Наличие точек излома (точки А и В) графика $P_M = f(F_T)$ объясняется особенностями алгоритма поиска частотных точек, осуществляемых целевой функцией PhaseMargin. Целевая функция для каждого значения варьируемого параметра F_T находит первую (минимальную) частоту, на которой модуль петлевого усиления ОПС равен 0 дБ, и для найденного значения частоты по фазовой характеристике петли определяет величину P_M . Однако при $F_T \leq 370$ кГц петлевое усиление ОПС принимает значение 0 дБ три раза — два раза, когда резонансный всплеск графика $|G_{NID}|_{дБ}$ возрастая и убывая пересекает соответствующую кривую семейства графиков $|A|_{дБ}$, а третий раз, когда указанные кривые пересекаются на высокой частоте. При $F_T \geq 399$ кГц кривые $|G_{NID}|_{дБ}$ и $|A|_{дБ}$ пересекаются только на высоких частотах, т. е. петлевое усиление ОПС принимает значение 0 дБ только один раз. Из сказанного следует, что части графика P_M , расположенные левее точки А и правее точки В, определяются различными факторами. Совершенно очевидно, что практический интерес представляет только часть графика P_M , расположенная правее точки В и соответствующая величинам $F_T \geq 399$ кГц. Из сказанного следует, что при $F_T \geq 399$ кГц ОПС (ПФ2) имеет значительный запас устойчивости по фазе, величина которого даже несколько превышает величину запаса устойчивости по фазе используемого в нем ОУ.

На рис. 42 приведена схема анализа частотных параметров ПФ2, соответствующая функциональной схеме, изображенной на рис. 35.

Следует заметить, что коэффициент погрешности $T_{ERR}(S)$ редко используется для оценки ошибок реализации АЧХ и ФЧХ высокодобротных звеньев фильтров. Погрешности коэффициентов передачи звеньев фильтров второго порядка, обусловленные неидеальностью ОУ, принято характеризовать относительными отклонениями частоты полюса и добротности [7, 30]:

$$FP[\%] = (FP_{ACTUAL} / FP_{IDEAL} - 1) \times 100, \quad (107)$$

PARAMETERS:

```

NumVar 1
Vac1 {{if(NumVar==1,1V,0V)}}
Vac2 {{if(NumVar==2,1V,0V)}}
FT 1GHz
Cd 0.000pF
Ccm 0.000pF
Rout 1e-6
Rd 1e12
Rcm 1e12
    
```

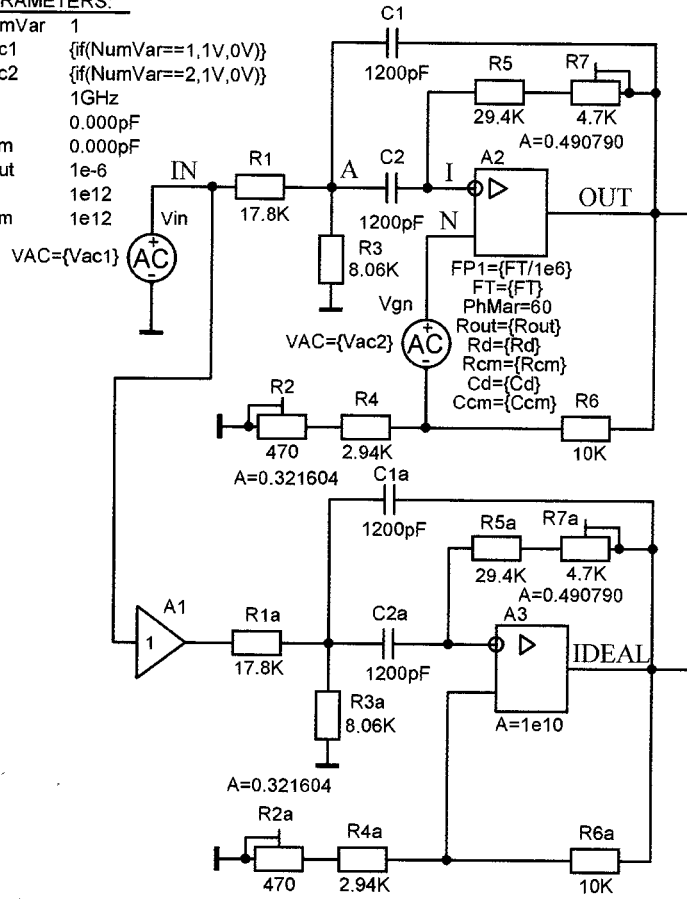


Рис. 42

$$Q[\%] = (Q_{ACTUAL} / Q_{IDEAL} - 1) \times 100. \quad (108)$$

В этих формулах Q_{IDEAL} , FP_{IDEAL} — добротность и частота полюса звена фильтра второго порядка, в котором

используется идеальный ОУ; Q_{ACTUAL} , FP_{ACTUAL} — фактическая добротность и частота полюса звена фильтра второго порядка, в котором используется реальный ОУ. Многовариантный анализ схемы на рис. 42 при вариации

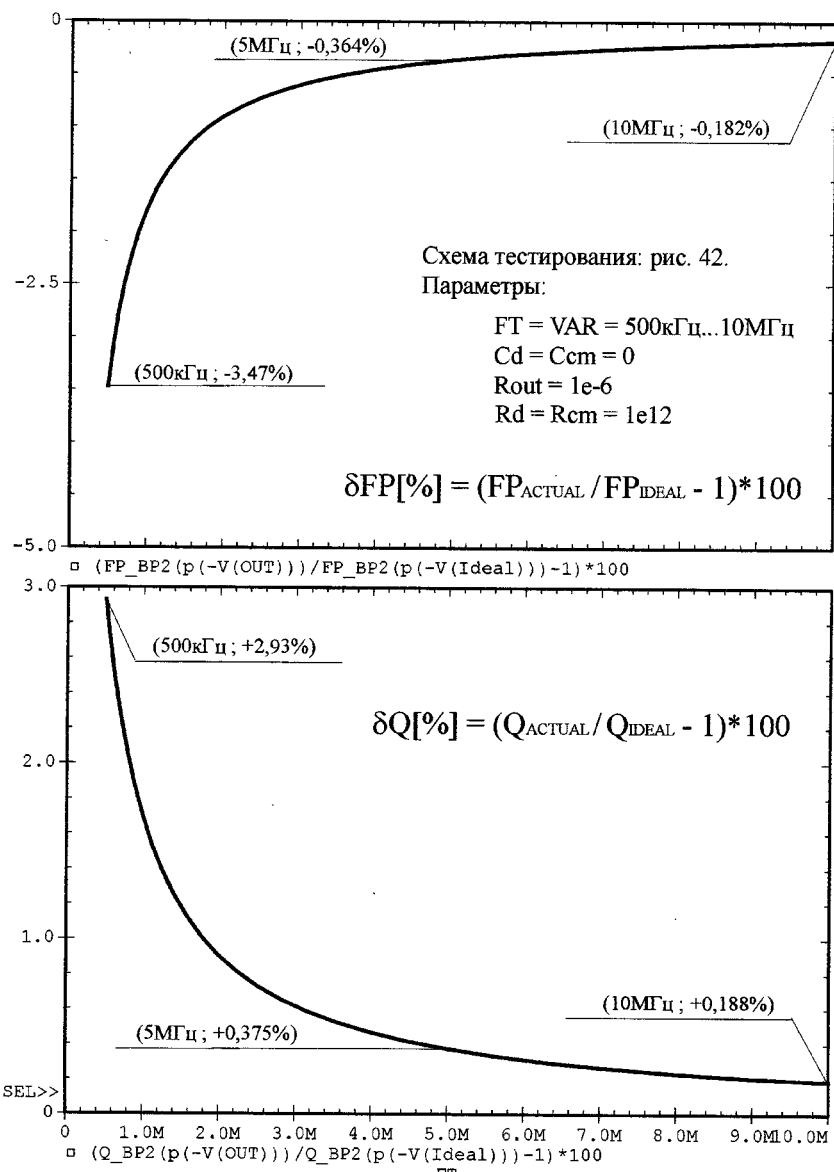


Рис. 43

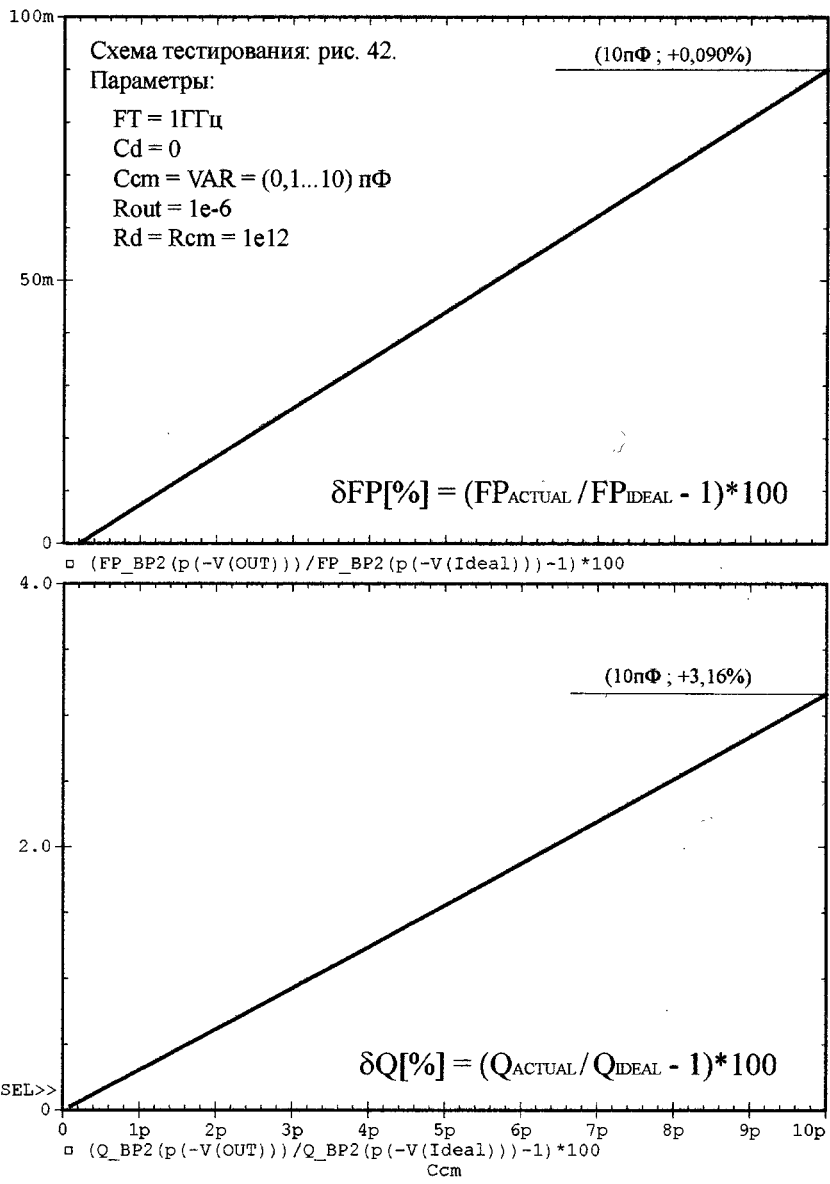


Рис. 44

параметров F_T , C_d , C_{CM} , R_d , R_{CM} , R_{out} показал, что:

- частота F_T единичного усиления ОУ существенно влияет на величины ошибок δFP и δQ (рис. 43);
- величина синфазной емкости C_{CM} существенно влияет на величину ошибки δQ (рис. 44);
- величина синфазного входного сопротивления R_{CM} существенно влияет на ошибку δQ и в меньшей степени на ошибку δFP (рис. 45);
- изменение параметров C_d (0...10 пФ), R_d (1...1000 МОм) и R_{OUT} (0...100 Ом) весьма незначительно влияет на величины ошибок δFP и δQ .

В процессе анализа, когда варьируется один из параметров ММ ОУ А2, величины других его параметров устанавливаются очень большими, либо очень малыми таким образом, чтобы в части этих параметров ОУ был близок к идеальному. Это гарантирует, что именно варьируемый параметр оказывает доминирующее влияние на величины ошибок δFP , δQ и позволяет по отдельности оценить влияние каждого из параметров ОУ на эти ошибки. Однако при таком подходе необходимо крайне

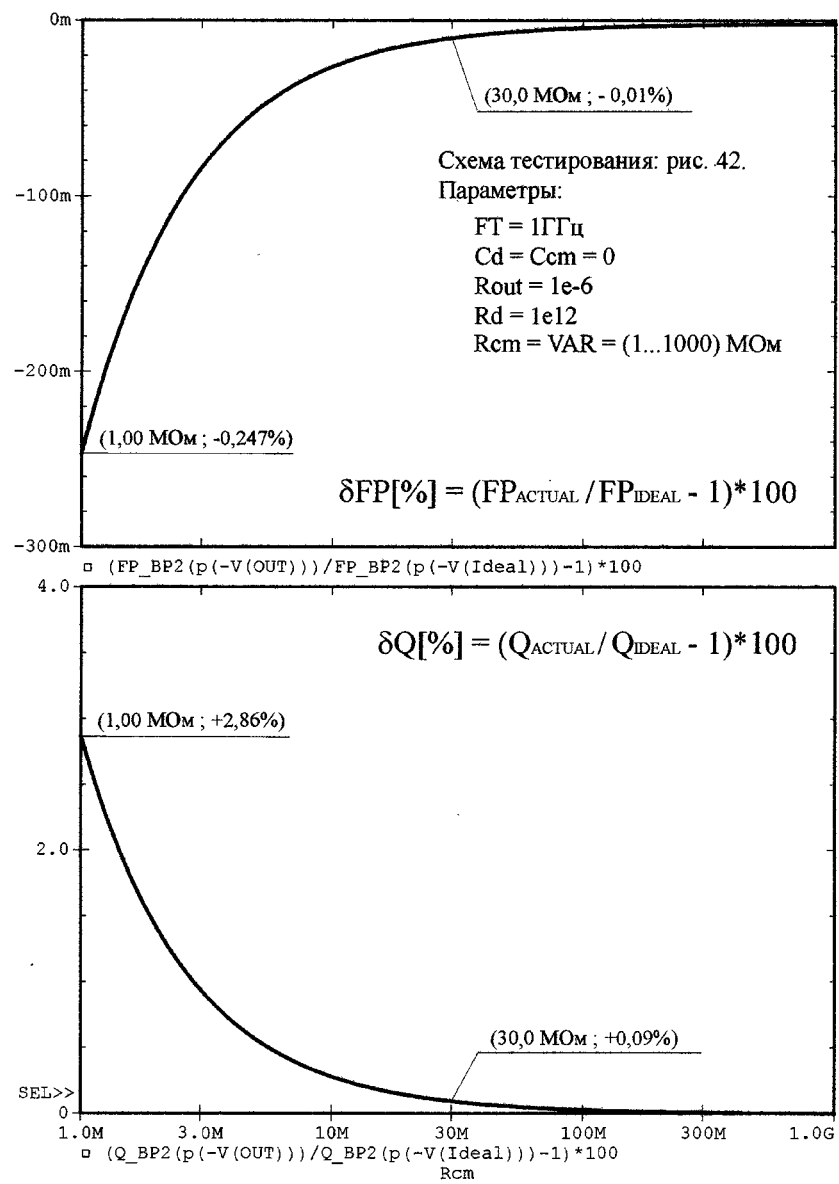


Рис. 45

осторожно делать обобщающие выводы на основании полученных результатов моделирования. Так, в частности, сделанный выше вывод о незначительном влиянии параметров C_d , R_d , R_{OUT} на ошибки δFP , δQ абсолютно верен только в том случае, когда в схеме ПФ2 используется ММ чрезвычайно широкополосного ОУ с $F_T \geq 1$ ГГц. Можно предположить, что при использовании такого ОУ влияние параметров C_d , R_d , R_{OUT} на ошибки δFP , δQ существенно подавляется глубокой ООС, которая в этом случае имеет место на частотах, близких к частоте F_p полюса ПФ2. Дополнительное тестирование показало, что при использовании в схеме ПФ2 ММ ОУ с $F_T = 2$ МГц влияние параметров C_d , R_d , R_{OUT} на ошибки ОПС существенно возрастает, но, тем не менее, остается достаточно малым. Так, например, при увеличении C_d с 0,1 до 10 пФ величина δQ возрастает с 0,906 % до 1,034 %, а увеличение R_{OUT} с 1 до 100 Ом вызывает увеличение δQ с 0,906 до 1,020 % (влияние этих параметров на величину δFP пренебрежимо мало).

Полученные результаты в части зависимости δFP , δQ от частоты F_T

единичного усиления ОУ (рис. 43) в целом соответствуют данным [7, 30]. А вот сведений о зависимости параметров ПФ2 от величины синфазной входной емкости ОУ в литературе обнаружить не удалось. Следует заметить, что влияние емкости C_{CM} на параметры ПФ2 снижается при уменьшении полного сопротивления (при уменьшении сопротивлений резисторов и пропорциональном увеличении емкостей конденсаторов) цепи ОС фильтра. Однако необходимо иметь в виду, что при этом уменьшается и входное сопротивление ПФ2.

Продолжение следует

ЛИТЕРАТУРА:

30. В. В. Масленников, А. П. Сироткин. Избирательные RC-усилители. — М.: Энергия, 1980, с. 215.

Сергей Лозицкий,
г. Брянск

Априорная оценка нелинейных искажений при проектировании усилителей на биполярных транзисторах

В статье показано, как используя классическую теорию электрических цепей и простые модели сигналов можно вычислить значение коэффициента нелинейных искажений для транзисторного усилительного каскада, а также для многокаскадного усилителя.

При проектировании транзисторных усилительных каскадов необходимо на начальном этапе оценить значение коэффициента нелинейных искажений (КНИ), чтобы последующий труд по расчету, сборке, настройке и испытанию усилителя не оказался напрасным.

Нелинейные искажения — это отклонение формы выходного сигнала усилителя от формы входного. У идеального усилителя формы входного и выходного сигнала должны совпадать. При подаче на вход усилителя синусоидального сигнала на выходе должны получить (пусть в другом масштабе) синусоидальный сигнал той же частоты, что и на входе. Различия формы временных зависимостей входного и выходного сигналов реального усилителя оцениваются с помощью КНИ, который определяют по формуле:

$$\text{КНИ} = \sqrt{\frac{A_2^2 + A_3^2 + A_4^2 + \dots}{A_1^2}}, \quad (1)$$

где A_i — амплитудные либо действующие значения отдельных гармонических составляющих (силы тока или напряжения) несинусоидального сигнала; i — номер гармонической составляющей.

Формула (1) удобна для оценки нелинейных искажений при наличии экспериментальных данных интенсивности отдельных гармонических составляющих (гармоник) выходного сигнала усилителя. При проектировании усилителя вычисление амплитуд всех отдельных гармоник для каждого конкретного случая оказывается нецелесообразным из-за громоздкости вычислений, поэтому необходимо иметь простые соотношения (удобные для инженерных расчетов), позволяющие оценить нелинейные искажения, создаваемые транзисторными усилительными каскадами.

Для каскадов с общим эмиттером и общей базой сквозная характеристика биполярного транзистора имеет вид, показанный на рис. 1 (кривая «без обратной связи»).

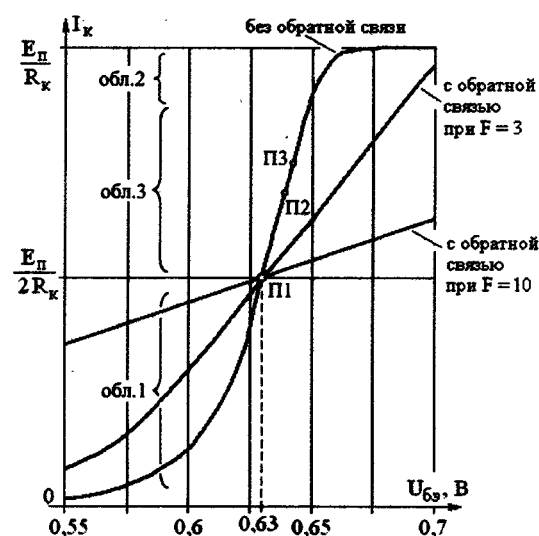


Рис. 1

Верхний загиб этой характеристики имеет меньший радиус кривизны, чем нижний. Источником такой асимметрии является различие причин искривления характеристики. В нижней части искривление обусловлено зависимостью базового тока I_B от напряжения база-эмиттер $U_{БЭ}$. На нижнем участке характеристики справедливо известное соотношение для коллекторного тока I_K [1]:

$$I_K = h_{21Э} I_B = I_S e^{\frac{U_{БЭ}}{\varphi_T}}, \quad (2)$$

где $h_{21Э}$ — коэффициент усиления по току используемого транзистора, I_S — постоянная величина (в данной статье принято значение $I_S \approx 10^{-13}$ А); φ_T — температурный потенциал р-п перехода, $\varphi_T = kT/q$; k — постоянная Больцмана ($k = 1,38 \times 10^{-23}$ Дж/град); T — абсолютная температура; q — заряд электрона, $1,6 \times 10^{-19}$ Кл; при $T = 17^\circ \text{C}$ $\varphi_T = 25$ мВ.

В верхней части характеристики искривление обусловлено ограни-

вающим действием коллекторного резистора R_K . Если сопротивление коллекторного резистора много меньше, чем динамическое сопротивление закрытого коллекторного р-п перехода (что справедливо для большинства практических случаев), то зависимость коллекторного тока от напряжения между коллектором и эмиттером $U_{КЭ}$ можно аппроксимировать (с достаточной для практики точностью) следующим соотношением:

$$I_K \approx h_{21Э} I_B \left(\frac{U_{КЭ}}{0,1 + U_{КЭ}} \right), \quad (3)$$

С помощью формулы (3) нетрудно получить выражение для определения коллекторного тока в каскадах с общим эмиттером:

$$I_{КЭ} = \frac{E_{П}}{R_K} + \frac{h_{21Э} I_B}{2} - \frac{E_{П} - 0,1}{2R_K} - \frac{1}{R_K} \times \sqrt{0,25(h_{21Э} I_B R_K - E_{П} + 0,1)^2 + 0,1 E_{П}}, \quad (4)$$

где $E_{П}$ — напряжение источника питания.

График сквозной характеристики транзистора, показанный на рис. 1, получен с помощью формулы (4) и хорошо согласуется с экспериментальными данными. Анализируя нелинейные искажения по методу пяти ординат [2] и ограничиваясь рассмотрением второй и третьей гармоник выходного несинусоидального сигнала, получаем зависимости КНИ от амплитуды входного синусоидального сигнала ($U_{ВХ.М}$), графики которых показаны на рис. 2.

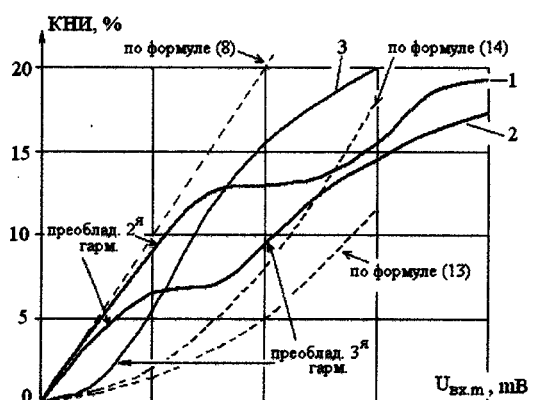


Рис. 2

Здесь кривая 1 — график КНИ при коллекторном токе покоя $I_{КП}$, соответствующем точке П1 на рис. 1 (т. е. $U_{БЭ \text{ пок}} \approx 0,63$ В). Кривая 2 — график КНИ при $I_{КП}$, соответствующем точке П2 ($U_{БЭ \text{ пок}} \approx 0,64$ В). Кривая 3 — это график КНИ при $I_{КП}$, соответствующем точке П3 ($U_{БЭ \text{ пок}} \approx 0,645$ В).

На начальных участках (при $U_{ВХ.М} < 10$ мВ) графиков 1 и 2 (рис. 2) в выходном несинусоидальном сигнала-

ле преобладает вторая гармоника, но при увеличении амплитуды входного сигнала интенсивность второй гармоники снижается и возрастает влияние третьей гармоники. Для графика 3 на начальном участке доминирует третья гармоника, но при увеличении $U_{ВХ.М}$ амплитуда второй гармоники возрастает и при $U_{ВХ.М} > 20$ мВ постепенно приближается к амплитуде третьей гармоники.

Для оценки нелинейных искажений целесообразно рассмотреть их возникновение в трех областях характеристики транзистора (рис. 1) по отдельности.

КНИ в области 1. Формирование нелинейных искажений в области 1 поясняется графиками на рис. 3.

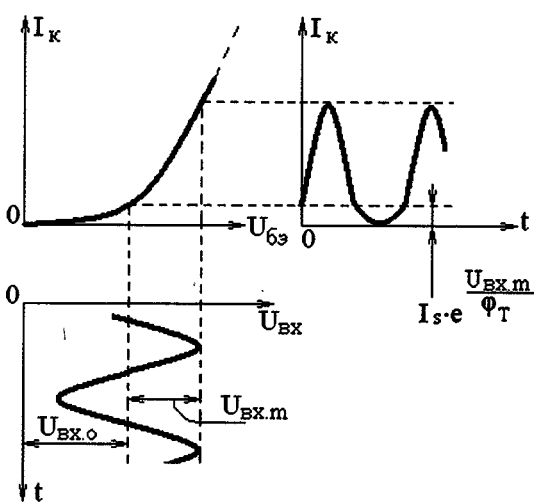


Рис. 3

Здесь видно, что для области 1 характерно «сплющивание» отрицательной полуволны коллекторного тока. Такой тип нелинейных искажений может быть объяснен наличием в коллекторном токе второй гармоники, что нетрудно увидеть на рис. 4.

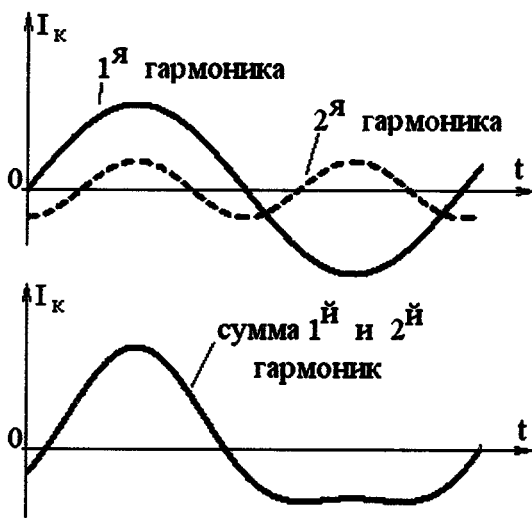


Рис. 4

Записывая формулу (2) в виде ряда Тейлора (для экспоненциальной функции) и учитывая, что

$$U_{бэ}(t) = U_{вх.0} + U_{вх.м} \sin \omega t, \quad (5)$$

где $\omega = 2\pi f$ — угловая частота, получаем:

$$I_k(t) = I_s e^{\frac{U_{вх.0}}{\phi_T}} \left(1 + \frac{U_{вх.м}}{\phi_T} \sin \omega t + \frac{U_{вх.м}^2}{2\phi_T^2} \cdot \frac{1 - \cos 2\omega t}{2} + \frac{U_{вх.м}^3}{6\phi_T^3} \cdot \frac{3\sin \omega t - \sin 3\omega t}{4} + \dots \right). \quad (6)$$

Из (6) видно, что временная зависимость коллекторного тока в области 1 содержит кроме основной (первой) гармоники четные и нечетные высшие гармоники. Ограничившись учетом второй и третьей гармоник получим формулу для КНИ в области 1 в следующем виде:

$$\text{КНИ.1} \approx \sqrt{\frac{\frac{U_{вх.м}^4}{16\phi_T^4} + \frac{U_{вх.м}^6}{576\phi_T^6}}{\left(\frac{U_{вх.м}}{\phi_T} - \frac{U_{вх.м}^3}{8\phi_T^3}\right)^2}}. \quad (7)$$

График зависимости КНИ.1 от $U_{ВХ.М}$ показан на рис. 5.

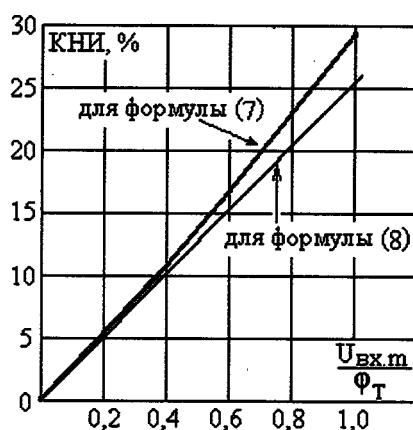


Рис. 5

Здесь видно, что зависимость КНИ.1 от $U_{ВХ.М}$ в области 1 достаточно близка к прямолинейной при $U_{ВХ.М} \leq 0,5\phi_T$. Если амплитуда входного напряжения ($U_{ВХ.М}$) не превышает $0,5\phi_T$, то с погрешностью не более 3% можно пренебречь влиянием третьей гармоники в формулах (6) и (7). Тогда в КНИ учитывается только вторая гармоника и в этом случае КНИ можно назвать коэффициентом нелинейных искажений по второй гармонике (КНИ.2) [4].

$$\text{КНИ.2} = \frac{A_2}{A_1} = \frac{U_{вх.м}}{4\phi_T} \approx 10U_{вх.м}. \quad (8)$$

При $U_{ВХ.М} = 1$ мВ (0,001 В) получаем $\text{КНИ.2} \approx 1\%$ (0,01).

Формула (8) очень проста и удобна для инженерных расчетов. Однако из рис. 2 видно, что формула (8) позволяет оценить только верхнюю границу ожидаемого значения КНИ для проектируемого усилительного каскада.

КНИ в области 2. Если усилительный каскад работает в области

2 (рис. 1), то «сплющивается» положительная полуволна коллекторного тока, как это видно из рис. 6. На рис. 7 показано, что такой тип нелинейных искажений может быть также объяснен наличием в коллекторном токе второй гармоники. Однако из сравнения рис. 4 и 7 видно, что вторые гармоники в областях 1 и 2 (рис. 1) противофазны. Для оценки КНИ в области 2 можно пользоваться формулой, аналогичной формуле (8), но с учетом большей кривизны сквозной характеристики:

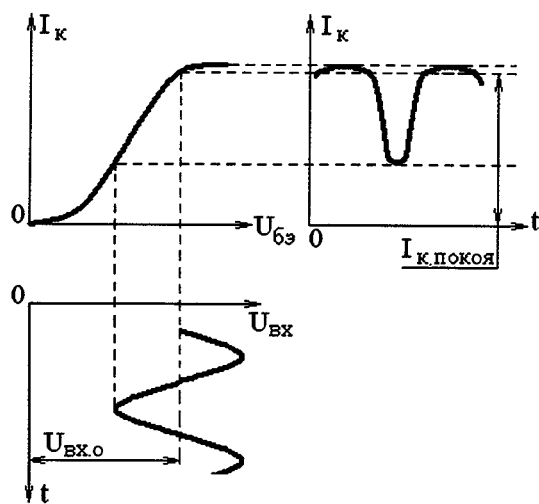


Рис. 6

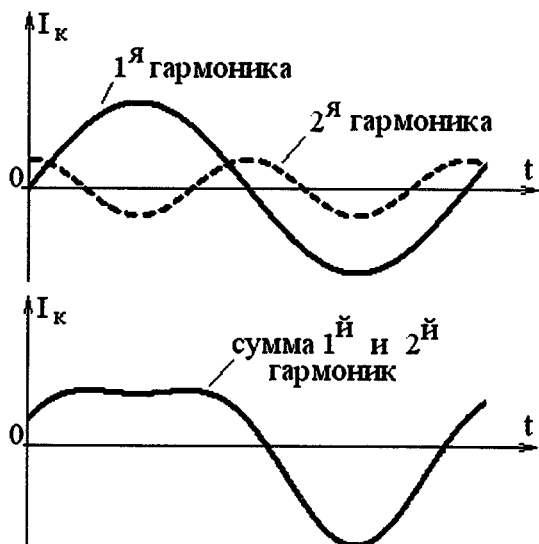


Рис. 7

$$\text{КНИ.2.2} \approx 60U_{вх.м}, \quad (9)$$

Формула (9) справедлива только для области (2), т. е. при заходе транзистора в область насыщения, поэтому данная формула не применяется автономно при оценке нелинейных искажений в тех случаях, когда в коллекторной цепи используется резистор R_K . Если в коллекторной цепи используется динамическая нагрузка, например, в виде нелинейного сопротивления закрытого коллекторного перехода комплементарного транзистора, то работа происходит именно в области 2 и КНИ следует оценивать по формуле (9).

КНИ в области 3. Область 3 отличается от областей 1 и 2 тем, что в области

3 вторые гармоники из областей 1 и 2 взаимно компенсируются и основную роль играет третья гармоника. Это заключение может быть проиллюстрировано графиками на рис. 8 и 9.

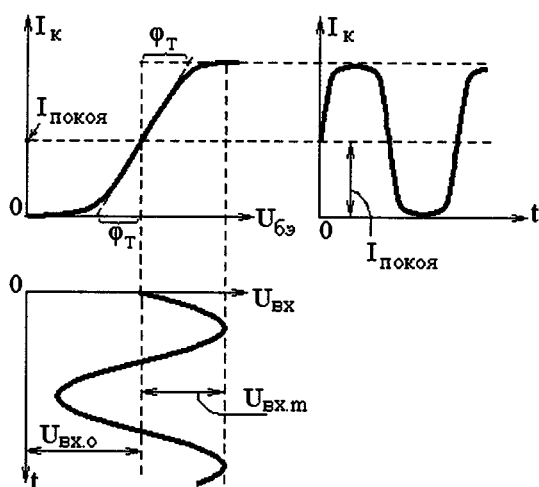


Рис. 8

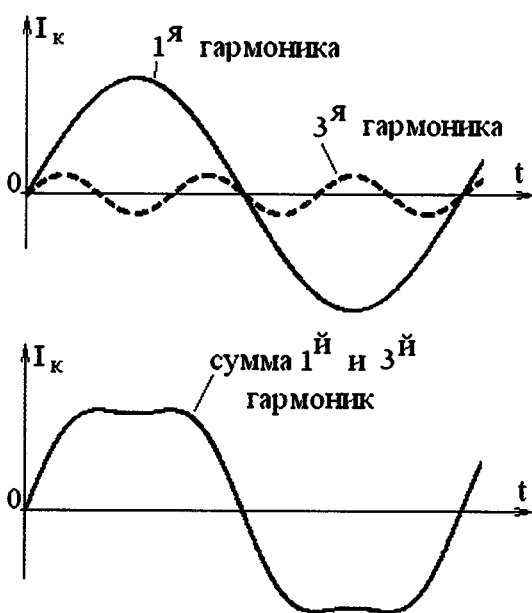


Рис. 9

На рис. 8 видно, что в области 3 временная зависимость коллекторного тока представляет собой синусоиду с двухсторонним ограничением. На рис. 9 показано, что такой вид нелинейных искажений может быть получен за счет наличия в коллекторном токе первой и третьей гармоник. Чтобы абстрагироваться от рассмотрения вторых гармоник в области 3, введем допущение о симметричности характеристики в этой области. Аппроксимируя симметричную сквозную характеристику транзистора функцией гиперболического тангенса, получим для области 3:

$$I_{к.з} \approx I_{покоя} \left(1 + \text{th} \frac{U_{бэ}(t) - U_{вх.о}}{\phi_T} \right), \quad (10)$$

где $I_{покоя} \approx E_{п}/2R_k$.

Представим функцию гиперболического тангенса в виде ряда Тейлора и ограничимся рассмотрением двух первых членов ряда. При этом погрешность вычислений, обусловленная ограничением числа членов ряда, в

интервале $0 \leq U_{вх.м} \leq \phi_T$ не превысит 6,8 %, что видно из табл. 1.

Для уменьшения погрешности вычислений, вызванной ограничением числа членов ряда, изменим коэффициент во втором члене усеченного ряда, как показано в табл. 1 для функции y_2 . В этом случае по-

Таблица 1

x	th x	$y_1 = x - x^3/3$	$(\text{th } x - y_1)/\text{th } x$	$y_2 = x - 2x^3/7$	$ \text{th } x - y_2 /\text{th } x$
0,5	0,462	0,458	0,008	0,464	0,001
1,0	0,716	0,667	0,068	0,714	0,003

Таблица 2

$U_{вх.м}/\phi_T$	$Z_1 = \text{КНИ.3 (12)}$	$Z_2 = \text{КНИ.3 (13)}$	$(Z_1 - Z_2)/Z_1$
0,25	0,019	0,02	-0,05
0,5	0,042	0,04	0,05
1,0	0,091	0,081	0,1

решность вычислений в интервале $0 \leq U_{вх.м} \leq \phi_T$ не превысит 0,3 %.

В результате усечения и преобразования ряда Тейлора получаем для области 3:

$$I_{к.з} \approx I_{покоя} \left(1 + \frac{U_{вх.м} \sin \omega t}{\phi_T} - \frac{2U_{вх.м}^3}{7\phi_T^3} \cdot \frac{3 \sin \omega t - \sin 3\omega t}{4} \right). \quad (11)$$

В формуле (11) фигурируют первая и третья гармоники, поэтому целесообразно рассмотреть только коэффициент нелинейных искажений по третьей гармонике (КНИ.3).

$$\text{КНИ.3.1} = \frac{-3}{1} = \frac{U_{вх.м}^2}{14\phi_T^2 - 3U_{вх.м}^2}. \quad (12)$$

График зависимости КНИ.3 от $U_{вх.м}$ показан на рис. 10.

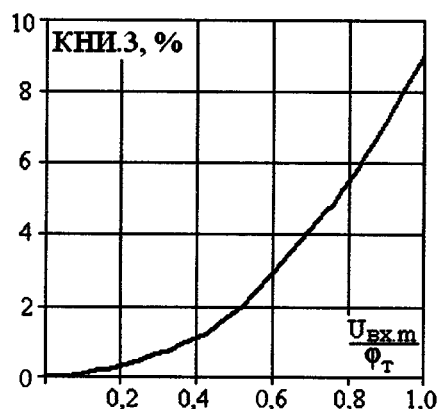


Рис. 10

Как видно из рис. 10, зависимость КНИ.3 от $U_{вх.м}$ можно аппроксимировать квадратичной параболой:

$$\text{КНИ.3.2} \approx 130U_{вх.м}^2. \quad (13)$$

В табл. 2 приведены результаты расчетов по формулам (12) и (13).

Из этой таблицы видно, что погрешность вычисления КНИ.3 по

простой формуле (13) не превышает $\pm 5 \%$ для $U_{вх.м} \leq 0,5\phi_T$.

Работа усилительного каскада в области 3 соответствует режиму класса А. В этом режиме нелинейные искажения минимальны. Действительно, однопроцентный КНИ.3 получается при $U_{вх.м} \approx 8,8 \text{ мВ}$. А в

режиме класса В, при такой же амплитуде входного напряжения, КНИ.2 составляет 8,8 %. Следовательно, режим А является более предпочтительным для минимизации нелинейных искажений. Заканчивая рассмотрение нелинейных искажений в области 3, следует отметить, что КНИ.3 показывает заниженное значение КНИ, ибо КНИ.3 не учитывает пятой, седьмой и других нечетных высших гармоник.

На рис. 2 штриховой линией показана зависимость КНИ.3 от $U_{вх.м}$ в соответствии с формулой (13). Из графиков на рис. 2 видно, что практические значения КНИ находятся в интервале от КНИ.3 до КНИ.2.

Так как в реальных усилителях (без отрицательной обратной связи) КНИ не должен превышать 10...12 %, можно рекомендовать для оценки нижней границы возможного значения КНИ следующее соотношение:

$$\text{КНИ.3.3} \approx 200U_{вх.м}^2. \quad (14)$$

Зная верхнюю и нижнюю границы возможного значения КНИ можно априорно оценить его ожидаемое значение как среднее арифметическое:

$$\begin{aligned} \text{КНИ.А} &\approx \frac{\text{КНИ.2} + \text{КНИ.3}}{2} = \\ &= 5U_{вх.м} + 100U_{вх.м}^2. \end{aligned} \quad (15)$$

Максимальные предельные значения КНИ.2 и КНИ.3

Простые и удобные для инженерных расчетов формулы (8), (13) и (15) справедливы для малых значений амплитуды входного напряжения (т. е. при

$U_{ВХ.М} \leq 0,5\Phi_T$). Для получения более полного представления о зависимости КНИ от $U_{ВХ.М}$ следует рассмотреть (хотя бы ориентировочно) вопрос о значениях КНИ при $U_{ВХ.М} \geq \Phi_T$.

При увеличении амплитуды входного напряжения на выходе усилительного каскада (в пределе) получим сигнал, форма которого показана на рис. 11.

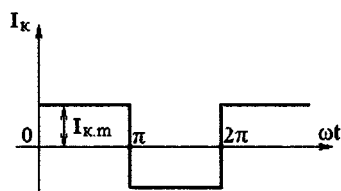


Рис. 11

Такой сигнал можно описать следующим рядом Фурье:

$$I_k(t) = I_{k.m} \frac{4}{\pi} \times \left(\sin \omega t + \frac{\sin 3\omega t}{3} + \frac{\sin 5\omega t}{5} + \dots \right). \quad (16)$$

Из формулы (16) легко выделить значения амплитуд первой и третьей гармоник, с помощью которых можно определить максимальное предельное значение КНИ.3:

$$\max \text{КНИ.3} = \frac{1}{3} = 0,33. \quad (17)$$

Зависимость КНИ.3 от $U_{ВХ.М}$ с учетом максимального предельного значения показана на рис. 12.

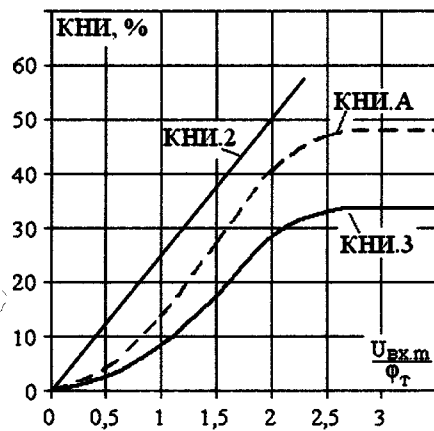


Рис. 12

При анализе предельных нелинейных искажений в области 3 можно оценить влияние пятой, седьмой и других нечетных высших гармоник на значение КНИ. Для ряда (16) предельный КНИ может быть определен по следующей формуле [3]:

$$\begin{aligned} \max \text{КНИ}_{\text{нечетн}} &= \sqrt{\sum_{n=1}^{\infty} \frac{1}{(2n-1)^2}} - 1 = \\ &= \sqrt{\frac{\pi^2}{8}} - 1 = 0,48. \end{aligned} \quad (18)$$

Сравнивая результаты вычислений по формулам (17) и (18) видим, что учет пятой и других нечетных вы-

сших гармоник приводит к увеличению значения КНИ в 1,45...1,5 раза. Распространяя действие множителя 1,45...1,5 на значения КНИ.3 при $U_{ВХ.М} \leq 3\Phi_T$, приходим к заключению о правомерности соотношения (15), что иллюстрируется графиками на рис. 12.

ЛИТЕРАТУРА:

1. У. Титце, К. Шенк. Полупроводниковая схемотехника: справочное руководство. Пер. с нем. — М.: Мир, 1982. — 512 с.
2. П. Шкритек Справочное руководство по звуковой схемотехнике. — М., Мир, 1991.
3. И. Н. Бронштейн, К. А. Семендяев. Справочник по математике для инженеров и учащихся Втузов. — М.: ГРФМЛ, 1981.
4. Ю. С. Ежков. Справочник по схемотехнике усилителей. — М.: Радио Софт, 2002.

Окончание следует

Юрий Дмитриев,
г. Самара

Продолжение. Начало — № 11/2004

Микроконтроллеры? Это же просто!

Как быть, если микроконтроллер вообще не имеет встроенных инструкций умножения, аналогичных MUL АВ? Например, их не имели некогда популярные микропроцессоры 8080/8085 (отечественные аналоги — КР580ВМ80 и КР1821ВМ85). Как быть в этом случае? Да очень просто. Нужно программно реализовать двоичное умножение, которое будет содержать лишь операции сложений и сдвигов и не потребует каких-либо средств для аппаратного умножения. Давайте посмотрим, как это можно выполнить.

Рассмотрим для начала простой пример, в которой осуществляется умножение двух трехразрядных десятичных чисел (рис. 26).

$$\begin{array}{r} \times 111 \\ 234 \\ + 444 \\ + 333 \\ + 222 \\ \hline 25974 \end{array}$$

Рис. 26

Здесь все очевидно: 111 умножается на каждую цифру 234, получаются частичные произведения 444, 333 и 222.

Далее к 444 прибавляется 333, сдвинутое на одну позицию влево (иными словами, умноженное на 10), после чего к этому прибавляется 222, сдвинутое на две позиции влево (т. е. умноженное на 100).

В общем, примерно то же, что и на рис. 24, только больше частичных произведений и их суммирований. Очевидно, что если бы мы умножали друг на друга не трехразрядные, а 12-разрядные десятичные числа, принцип формирования произведения был бы тем же, только объем вычислений был бы намного больше. Так вот, тот же принцип лежит в основе алгоритма умножения двоичных чисел, не использующего инструкцию MUL АВ. Давайте убедимся в этом.

Вначале заметим, что поскольку нам придется осуществлять умножение друг на друга двоичных цифр, нам нужно ознакомиться с двоичной таблицей умножения (табл. 8). Должен вам сказать, она несоизмеримо проще своего десятичного эквивалента.

Таблица 8

X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

Очевидно, что любая цифра (или число), умноженная на 0, даст нулевое произведение, и в то же время умножение цифры (или числа) на 1 не изменяет исходной цифры (числа). Пожалуй, стоит акцентировать на этом внимание, поскольку именно тот факт, что в двоичной системе всего две цифры 0 и 1, упрощает умножение. Нам не нужно ничего умножать на цифры 2, 3, 4 и т. д. Все, на что мы умножаем, это цифры 0 и 1. Да, и не забудьте, что сложения при двоичном умножении тоже двоичные.

Давайте теперь с учетом этой таблицы умножим 1010_2 на 101_2 (рис. 27).

$$\begin{array}{r} \times 1010_2 \\ \underline{101_2} \\ + 1010_2 \\ + 0000_2 \\ + 1010_2 \\ \hline 110010_2 \end{array}$$

Рис. 27

Видите? Если разряд множителя 101_2 равен 0, то прибавляется 0 (или ничего не прибавляется), а если разряд множителя равен 1, то прибавляется множитель 1010_2 , сдвинутый на соответствующее число позиций влево. Напомним, что когда мы сдвигаем двоичное число на разряд влево или приписываем к нему справа 0, это эквивалентно умножению числа на основание двоичной системы, т. е. на $10_2 = 2_{10}$. Можете проверить, что $1010_2 \cdot 101_2 = 110010_2$, произведя умножение эквивалентных десятичных чисел 10_{10} и 5_{10} . Равны ли 50_{10} и 110010_2 ?

Приведем еще несколько примеров.

Умножим 101_2 на 110_2 (рис. 28).

$$\begin{array}{r} \times 101_2 \\ \underline{110_2} \\ + 000_2 \\ + 101_2 \\ + 101_2 \\ \hline 11110_2 \end{array}$$

Рис. 28

Умножим 111_2 на 111_2 (рис. 29).

$$\begin{array}{r} \times 111_2 \\ \underline{111_2} \\ + 111_2 \\ + 111_2 \\ + 111_2 \\ \hline 110001_2 \end{array}$$

Рис. 29

Если в таких примерах, как этот, у вас не получается правильного произведения (здесь умножается 7_{10} на 7_{10} , должно получиться 49_{10}), возможны две ошибки: просто неверно выполненное сложение или одновременное сложение «в столбик» трех и более чисел. Старайтесь не складывать «в столбик» более двух чисел одновременно. Почему? Хотя бы потому, что в предыдущих главах операция сложения была введена как операция над двумя числами, а не тремя, и если вы попытаетесь сложить три ненулевых числа в столбик, то у вас, несомненно, где-то возникнет проблема с определением правильной цифры суммы и правильного значения переноса. Даже несмотря на то, что в приведенных

примерах умножение требует сложения трех и более ненулевых чисел, и при этом не приводится детального описания сложения, вы не должны складывать три и более чисел «в столбик», делайте сложение по два числа и не ошибетесь. Так, в вышерассмотренном примере правильно вычислять сумму частичных произведений следующим образом:

$$\begin{aligned} 111_2 + 1110_2 + 11100_2 &= (111_2 + 1110_2) + 11100_2 = \\ &= 10101_2 + 11100_2 = 110001_2. \end{aligned}$$

Подведем некоторый итог. В нашем микроконтроллере есть инструкция MUL AB, которая позволяет умножать два целых беззнаковых восьмибитных числа. А что делать, если в микропроцессоре вообще нет инструкции умножения? Ответ прост. Нужно просто умножить двоичные числа «в столбик», как было показано выше. При этом все, что понадобится делать, это сдвигать и иногда складывать. Ни больше, ни меньше.

Следующая подпрограмма (рис. 30) производит умножение двух целых беззнаковых двоичных 16-битных чисел «в столбик». Основным отличием от обсужденного алгоритма умножения является то, что не множимое число сдвигается влево перед сложением с накапливаемой суммой, а накапливаемая сумма сдвигается вправо перед сложением с множимым. Принципиальной разницы в том, что относительно чего сдвигается, конечно, нет. Главное, чтобы это было сделано правильно. Как и в предыдущем случае, подпрограмма ожидает множимое в регистрах R3 (старший байт) и R2 (младший байт), а множитель в регистрах R5 (старший байт) и R4 (младший байт). 32-битное произведение записывается в регистры R7 (старший байт), R6, R5 и R4 (младший байт). Флаг CY, как и раньше, устанавливается в 1, если произведение занимает больше 16 бит (т. е. R7 или R6 или оба не равны 0), в противном случае CY устанавливается в 0.

Итак, мы рассмотрели подпрограмму, осуществляющую умножение целых двоичных чисел путем сложений частичных произведений и сдвига вправо накапливаемой суммы после каждого цикла анализа разряда множителя. В то же время, рассматривая примеры на рис. 27—29, мы не накапливаемую сумму сдвигали вправо перед сложением с множимым, а множимое число сдвигали влево перед сложением с накапливаемой суммой. Как было отмечено, принципиальной разницы в том, что относительно чего сдвигать, конечно, нет. Однако каждый из вариантов являет собой самостоятельную вычислительную схему целочисленного умножения. Таких схем существует четыре. Рассмотрим их более подробно. Замечу только, что для тех, кто только осваивает технику программирования, приводимая отсюда и до конца этого раздела информация при первом знакомстве может быть пропущена без ущерба для понимания остального материала.

Как уже отмечалось, любой программный метод выполнения умножения (за исключением громоздкого табличного метода) основан на многократном выполнении операций сложения. Очевидно, что простейший способ умножения целых беззнаковых чисел $X \cdot Y = Z$ есть суммирование множимого X с накоплением его столько раз, сколько единиц содержит значение множителя Y . Этот способ при больших значениях Y требует значительных затрат времени на выполнение многократных операций сложения, и поэтому применение его ограничено.

```

MUL16A:
; Подпрограмма умножения двух целых беззнаковых
; 16-разрядных чисел
; Вход:
; R3R2 - множимое
; R5R4 - множитель
; Выход:
; R7R6R5R4 - 32-разрядное произведение
; Флаги:
; CY - признак переполнения 16-разрядного произведения
; Используемые регистры:
; A, B, R2-R7
; Требуется свободных байт в стеке:
; 2

```

```

MOV R6, #0
MOV R7, #0 ; обнулили СТС произведения
MOV B, #16 ; 16 циклов сложения и сдвиг

```

```

MUL16A_1:
MOV A, R4
RRC A
JNC MUL16A_2 ; пропустить сложение, если бит
множителя равен 0

```

```

MOV A, R6
ADD A, R2
MOV R6, A
MOV A, R7
ADDC A, R3
MOV R7, A ; добавили множимое к частичному
произведению

```

```

MUL16A_2:
MOV A, R7
RRC A
MOV R7, A
MOV A, R6
RRC A
MOV R6, A
MOV A, R5
RRC A
MOV R5, A
MOV A, R4
RRC A
MOV R4, A ; R7R6R5R4 сдвинуто вправо

```

```

DJNZ B, MUL16A_1 ; закичивание
MOV A, R7
ORL A, R6
CLR C
JZ MUL16A_3 ; CY=0, если результат помещился
в 16 бит
SETB C ; CY=1, если результат не поместился
в 16 бит

```

```

MUL16A_3:
RET

```

Сокращение числа операций сложения при реализации умножения основано на методах совместного анализа цифр множителя. Эти методы дробят процесс получения произведения на ряд шагов, связанных с формированием частичных произведений (ЧП) — произведений множимого на отдельные разряды или группы разрядов множителя — и их суммированием, т. е. формированием суммы частичных произведений (СЧП). Ускоренные методы умножения используют анализ одновременно двух и более цифр множителя для получения ЧП, а обычные (неускоренные) методы — анализ поочередно каждой отдельной цифры множителя. Ускоренные методы требуют специальных схемотехнических решений, которые, как правило, не используются в микроконтроллерах, и поэтому здесь мы их рассматривать не будем.

Методы умножения двоичных беззнаковых чисел основаны на представлении произведения в виде полинома:

$$\begin{aligned}
 Z = X \cdot Y &= X \left(\sum_{i=0}^{n-1} y_i \cdot 2^i \right) = \sum_{i=0}^{n-1} (X \cdot y_i) \cdot 2^i = \\
 &= \sum_{i=0}^{n-1} \text{ЧП}_i \cdot 2^i,
 \end{aligned} \tag{13}$$

где $y_i \in \{0,1\}$ — двоичные цифры множителя; $\text{ЧП}_i = X \cdot y_i$ — частичные произведения ($\text{ЧП}_i = X$, если $y_i = 1$, и $\text{ЧП}_i = 0$, если $y_i = 0$).

Напомним, что умножение $\text{ЧП}_i \neq 0$ на степень 2^i эквивалентно сдвигу множимого на i разрядов влево. Формирование произведения согласно формуле (13) представляется как последовательность таких действий:

- 1) обнуление СЧП;
- 2) анализ младшего разряда множителя: если $y_0 = 1$, выполняется сложение $\text{ЧП}_0 = X$ с СЧП и переход к п. 3; если $y_0 = 0$, сразу переход к п. 3;
- 3) сдвиг множимого влево;
- 4) анализ очередного разряда множителя: если $y_1 = 1$, выполняется сложение $\text{ЧП}_1 \cdot 2^1$ с СЧП и переход к п. 5; если $y_1 = 0$, непосредственно переход к п. 5;
- 5) сдвиг множимого влево;
- 6) анализ очередного y_2 разряда множителя и т. д.

После анализа старшего y_{n-1} разряда множителя осуществляется последнее сложение $\text{ЧП}_{n-1} \cdot 2^{n-1}$ с СЧП (если $y_{n-1} = 1$), и процесс прекращается. Результирующая СЧП является искомой. Очевидно, что неподвижную СЧП и сдвигаемое влево на $n-1$ разряд n -разрядное множимое необходимо размещать в $2n$ -разрядных форматах, причем операция сложения должна обрабатывать $2n$ -разрядные данные. Иными словами, 16-битное множимое ($n = 16$) мы должны разместить в четырех 8-разрядных регистрах или в четырех ячейках памяти, что обеспечит нам его хранение в процессе сдвига и сложения с СЧП. Соответственно, такое же число регистров или ячеек памяти нужно отвести под СЧП, и к тому же сложение в этом случае должно оперировать с четырехбайтными числами.

Форматы множимого и операции сложения можно ограничить n разрядами, если изменить процедуру обработки, определяемую формулой (13), следующим образом:

$$\begin{aligned}
 Z &= \sum_{i=0}^{n-1} X \cdot 2^i \cdot y_i = \sum_{i=0}^{n-1} (X \cdot 2^n) \cdot 2^{i-n} \cdot y_i = \\
 &= \sum_{i=0}^{n-1} V \cdot 2^{i-n} \cdot y_i,
 \end{aligned} \tag{14}$$

где $V = X \cdot 2^n$ — множимое, сдвинутое влево на n разрядов.

Преобразуем полученное выражение по схеме Горнера:

$$Z = (\dots((0 + V \cdot y_0) \cdot 2^{-1} + V \cdot y_1) \cdot 2^{-1} + \dots + V \cdot y_{n-1}) \cdot 2^{-1}. \quad (15)$$

Выражения в скобках в формуле (15) представляют собой последовательные значения СЧП_i, определяемые рекуррентной формулой

$$\text{СЧП}_i = \text{СЧП}_{i-1} \cdot 2^{-1} + V \cdot y_{i-1}, \text{СЧП}_0 = 0, i \in \{1, \dots, n\} \quad (16)$$

где $V \cdot y_{i-1} = \text{ЧП}_{i-1}$. Заметим, что умножение СЧП на 2^{-1} эквивалентно ее сдвигу вправо на один разряд.

В соответствии с формулами (15) и (16) процесс формирования произведения выглядит следующим образом:

- 1) обнуление СЧП;
- 2) анализ младшего разряда множителя: если $y_0 = 1$, сложение $\text{ЧП}_0 = X$ с СЧП и переход к п. 3; если $y_0 = 0$, непосредственно переход к п. 3;
- 3) сдвиг СЧП вправо;
- 4) анализ очередного разряда y_1 множителя и т. д. После анализа старшего разряда y_{n-1} множителя осуществляются последнее сложение ЧП_{n-1} с СЧП (если $y_{n-1} = 1$) и последний сдвиг СЧП вправо, после чего процесс прекращается.

Таким образом, данный процесс умножения требует для представления неподвижного множимого $V = X \cdot 2^n$ n -разрядного формата, поскольку младшие n разрядов множимого равны нулю, и такого же формата для операции сложения (сложение выполняется со старшими n разрядами СЧП). Формулы (13) и (15) определяют алгоритмы умножения с младших разрядов множителя соответственно при неподвижной и подвижной СЧП.

Аналогичные формулы имеют место при умножении со старших разрядов множителя:

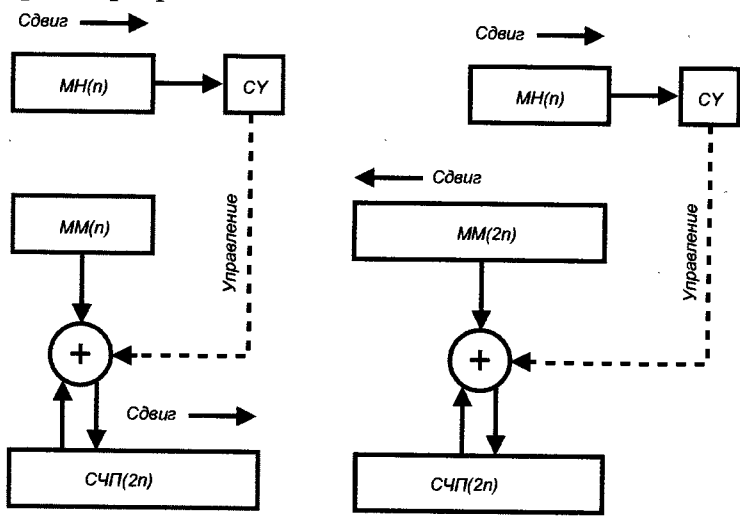


Рис. 31

Рис. 32

$$Z = (\dots(((0 \cdot 2^1 + X \cdot y_{n-1}) \cdot 2^{-1} + X \cdot y_{n-2}) \cdot 2^{-1} + \dots + X \cdot y_1) \cdot 2^{-1} + X \cdot y_0; \quad (17)$$

$$\text{СЧП}_i = \text{СЧП}_{i-1} \cdot 2^1 + X \cdot y_{n-i}, \text{СЧП}_0 = 0, i \in \{1, \dots, n\} \quad (18)$$

$$Z = \sum_{i=0}^{n-1} V \cdot y_{n-i-1} \cdot 2^{-(i+1)}, V = X \cdot 2^n. \quad (19)$$

Формулы (17—19) получены из полинома (13) по схеме Горнера и перенумерацией индексов. Выражения (17) и (18) определяют процесс умножения с неподвижным n -разрядным множимым и подвижной, сдвигаемой влево $2n$ -разрядной СЧП. Этот процесс реализуется такой последовательностью действий:

- 1) обнуление СЧП;

- 2) сдвиг СЧП влево;
- 3) анализ старшего разряда множителя: если $y_{n-1} = 1$, выполняется сложение $\text{ЧП}_{n-1} = X \cdot y_{n-1}$ с СЧП и переход к п. 4; если $y_{n-1} = 0$, прямой переход к п. 4;
- 4) сдвиг СЧП влево;
- 5) анализ очередного y_{n-2} разряда множителя и т. д. После анализа младшего разряда y_0 множителя выполняется последнее сложение $\text{ЧП}_0 = X \cdot y_0$ с СЧП (если $y_0 = 1$), и процесс прекращается.

Выражение (19) определяет процесс умножения с неподвижной СЧП и подвижным сдвигаемым вправо $2n$ -разрядным множимым. Этот процесс имеет следующую последовательность действий:

- 1) обнуление СЧП;
- 2) сдвиг множимого вправо;
- 3) анализ старшего разряда множителя: если $y_{n-1} = 1$, выполняется сложение $\text{ЧП}_{n-1} \cdot 2^{-1}$ с СЧП и переход к п. 4; если $y_{n-1} = 0$, прямой переход к п. 4;
- 4) сдвиг множимого вправо;
- 5) анализ очередного разряда y_{n-2} множителя и т. д. После анализа младшего разряда y_0 множителя осуществляется последнее сложение $\text{ЧП}_0 \cdot 2^{-1}$ с СЧП (если $y_0 = 1$), и процесс прекращается.

Рассмотренным четырем алгоритмам умножения соответствуют четыре вычислительные схемы, используемые при разработке программ умножения (рис. 31—34):

схема умножения 1 (рис. 31) соответствует выражениям (15), (16);

схема умножения 2 (рис. 32) — формуле (13);

схема умножения 3 (рис. 33) — соотношениям (17), (18);

схема умножения 4 (рис. 34) — формуле (19).

На схемах приведены следующие обозначения:

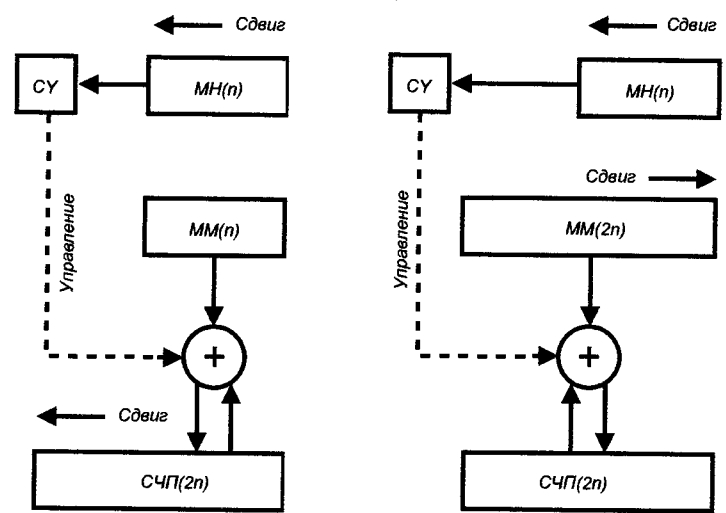


Рис. 33

Рис. 34

МН — множители, ММ — множимые (с указанием в скобках разрядности данных). Анализ разрядов множителя осуществляется путем сдвига его разрядов вправо (умножение с младших разрядов в схемах 1, 2) или влево (умножение со старших разрядов в схемах 3, 4) с модификацией признака переноса СУ, используемого для управления процессом накопления СЧП.

Продолжение следует

Александр Фрунзе,
Алексей Фрунзе,
г. Москва

Скремблирование передаваемых данных

ВЕРОЯТНОСТНАЯ синхронизация системы «скремблер-дескремблер» с изолированными генераторами псевдослучайных последовательностей битов

Как было показано, система «скремблер-дескремблер» с изолированными от линии связи генераторами псевдослучайных битовых последовательностей (рис. 3) обладает некоторыми преимуществами по сравнению с системой на основе неизолированных генераторов (рис. 2). Основное преимущество состоит в том, что ошибки, поступающие из линии, не размножаются. Кроме того, такая система более устойчива по отношению к неблагоприятным последовательностям битов, которые формируются в силу случайных стечений обстоятельств, либо по злому умыслу хакера. Однако есть и существенный недостаток — сложность обеспечения кодовой синхронизации между скремблером и дескремблером.

Это связано с тем, что в «классической» системе для поддержания синхронной работы сдвиговых регистров скремблера и дескремблера необходимо периодически прерывать передачу полезных данных и передавать по линии связи служебные информационные кадры, содержащие достаточно длинные цепочки синхронизирующих битов. Это уменьшает эффективную скорость передачи данных по линии, усложняет протокол обмена и требует значительного времени ожидания дескремблером служебного кадра в случае потери синхронизации. В течение этого времени передача пользовательских данных невозможна.

Предлагаемое решение (рис. 11) лишено этих недостатков. Синхронизация системы на основе изолированных генераторов выполняется автоматически и не требует введения в поток передаваемых данных каких-либо служебных кадров. Это позволяет повысить эффективную скорость передачи данных и исключить из системы программные средства установления и поддержания синхронизации. Кроме того, уменьшаются потери данных при восстановлении синхронизации в случае ее нарушения.

В общем виде идея построения системы такова — скремблер и де-

скремблер содержат изолированные от линии связи генераторы псевдослучайной последовательности битов с одинаковой структурой обратных связей. Скремблированный поток битов постоянно просматривается двумя одинаковыми анализаторами кодов, размещенными в скремблере и дескремблере, с целью отыскания в нем заранее заданных кодов. Обнаружение каждого такого кода скремблером и дескремблером приводит к одновременной установке обоих генераторов псевдослучайной последовательности битов в определенное состояние, соответствующее этому коду. Таким образом, генераторы в случайные моменты одновременно устанавливаются в одинаковые состояния по мере передачи полезных данных без применения какой-либо служебной процедуры установления синхронизации.

Если разрядность искомым кодов достаточно велика (она не связана с разрядностью регистров RG1 и RG2), то одновременная коррекция состояний генераторов происходит сравнительно редко по мере обнаружения этих кодов в случайном потоке битов. Другими словами, большую часть времени генераторы работают в режиме «естественного» последовательного перехода от предыдущего состояния генератора к последующему. Если кодовая синхронизация не была нарушена, то моменты одновременной установки генераторов в новые начальные состояния лишь подтверждает ее. Если кодовая синхронизация была ранее потеряна, то она восстанавливается при первом же обнаружении одного из заданных кодов в потоке скремблированных данных.

Рассмотрим предлагаемую систему подробнее (рис. 12).

Сдвиговые регистры RG1 и RG2 совместно с дешифраторами DC1 и DC2 представляют собой анализаторы кодов. Регистры RG1 и RG2 предназначены для временного хранения фрагментов SDATA и SDATA* потока

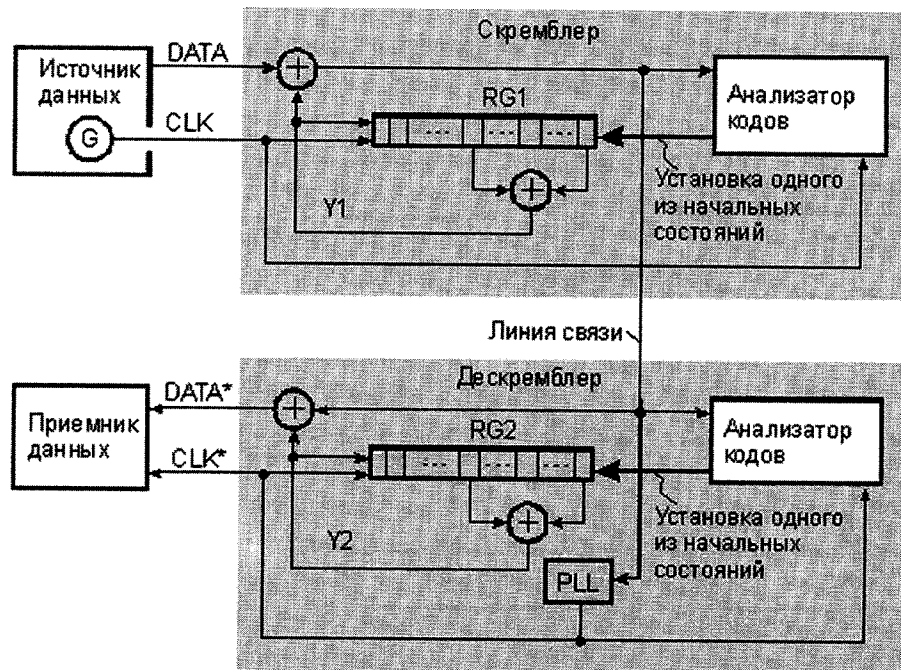


Рис. 11

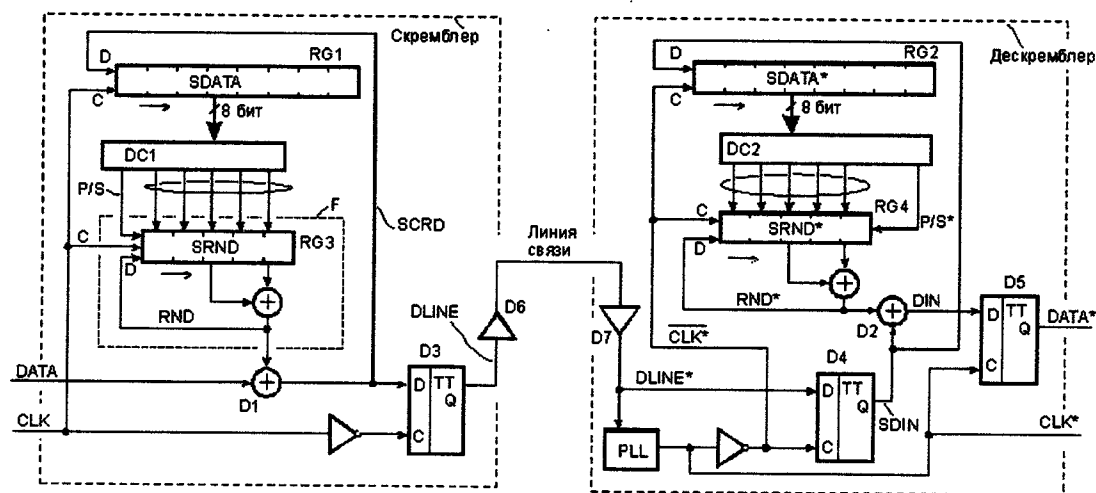


Рис. 12

скремблированных данных. В установленном режиме эти фрагменты одинаковы (совпадают с точностью до задержки передачи). Прием очередного бита в регистр RG1 (RG2) происходит по фронту сигнала на его синхронизирующем входе С. Одновременно с приемом очередного бита с входа D ранее хранимые данные сдвигаются на один разряд вправо (по стрелке). Выходящие из регистров биты теряются.

Разрядность регистра RG1 (RG2) выбрана равной восьми, хотя она может быть больше или меньше. Динамику работы регистра RG1 можно проследить по таблице его состояний (рис. 13).

SCRD	SDATA ₂	SDATA ₁₆
0	00010000	10
1	00001000	08
0	10000100	84
0	01000010	42
0	00100001	21
1	00010000	10
1	10001000	88
0	11000100	C4
1	01100010	62
0	10110001	B1
1	01011000	58
0	10101100	AC
0	01010110	56
1	00101011	2B
1	10010101	95
1	11001010	CA
0	11100101	E5
0	01110010	72

Рис. 13

Генератор F псевдослучайной последовательности битов скремблера построен на основе сдвигового регистра RG3, генератор дескремблера — на регистре RG4.

Сдвиговые регистры RG3 и RG4 предназначены для временного хранения псевдослучайных кодов SRND и SRND*. В установленном режиме эти коды одинаковы (совпадают с точностью до задержки передачи). Прием очередного бита в регистр RG3 (RG4) с входа D происходит по фронту сигнала на синхронизирующем входе С при условии, что на его управляющем входе P/S (P/S*), задающем режим параллельного или последовательного приема данных, присутствует сигнал лог. 0. Одновременно с приемом очередного бита с входа D происходит сдвиг ранее хранимого кода на один разряд вправо (по стрелке).

Если на управляющем входе P/S (P/S*) регистра RG3 (RG4) присутствует сигнал лог. 1, то по фронту сигнала на синхронизирующем входе С в регистр принимается параллельный код с группы входов, обведенной на рис. 12 овалом. В данном примере построения устройства разрядность

регистра RG3 (RG4) выбрана равной пяти, хотя она может быть больше или меньше. При этом точки подключения элемента «Исключающее ИЛИ» к регистру RG3 (RG4) выбираются в соответствии с таблицей, представленной на рис. 1, в.

В таблице (рис. 14, а) представлен список состояний генератора псевдослучайной последовательности битов на основе сдвигового регистра RG3. Диаграмма состояний этого генератора (рис. 14, б) отражает перемещение указателя А текущего состояния по кольцевому пути, линии В и С разделяют диаграмму на четыре примерно равных сектора.

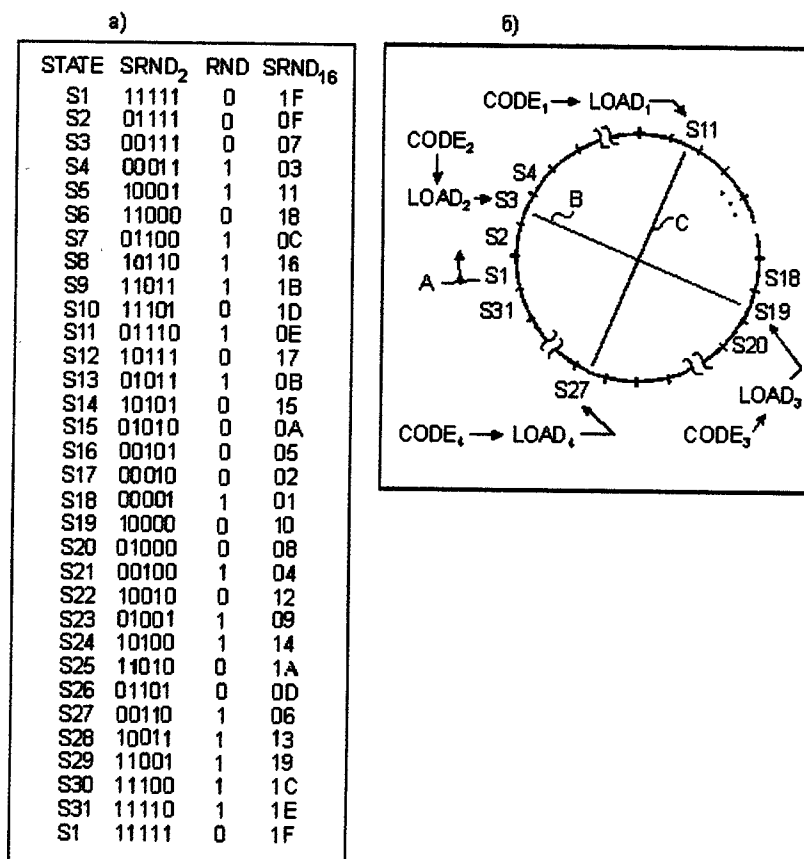


Рис. 14

Начальное состояние регистра RG3 может быть любым, в частности, нулевым. Выход из нулевого состояния происходит при записи в регистр ненулевого параллельного кода. Программа инициализации скремблера может предусматривать выдачу на его вход некоторого кода CODE₁, который распознается дешифратором DC1. Если в регистре RG3 первоначально присутствовал нулевой код, то код CODE₁ без изменения проходит через элемент D1 и последовательно загружается в регистр RG1.

Дешифратор DC1 реагирует на него переводом регистра RG3 в режим параллельной загрузки (P/S = 1) и формированием ненулевого кода LOAD₁ который затем принимается в регистр. Таким образом, генератор F выходит из запрещенного состояния 000...0. Если первоначальное состоя-

ние регистра RG3 было ненулевым, то выдача кода CODE₁ на вход скремблера оказывается бесполезной, но не приводит к каким-либо нежелательным последствиям. Возможна также и аппаратная установка регистра RG3 в ненулевое состояние (соответствующий вход установки начального состояния регистра не показан).

Начальное состояние регистра RG4 также может быть любым, в том числе, нулевым. Это состояние обновляется, т. е. становится заведомо ненулевым при обнаружении дешифратором DC2 в скремблированном потоке данных одного из заранее заданных кодов (CODE₁ и, возможно, других).

Элементы «Исключающее ИЛИ» D1 и D2 формируют скремблированный SCRД и дескремблированный DIN сигналы данных.

Триггеры D3, D4 и D5 принимают биты данных с входа D по фронту сигнала на входе синхронизации С. Триггеры D3 и D5 формируют выходные сигналы DLINE и DATA*, в которых на границах между битовыми интервалами сигнал может измениться только один раз, в то время как входные сигналы SCRД и DIN этих триггеров на границах между битовыми интервалами могут изменяться многократно из-за неодновременного протекания переходных процессов на входах каждого из элементов D1 и D2.

Триггер D4 практически полностью устраняет джиттер входного сигнала («дрожание» фронтов на границах между битовыми интервалами) бла-

годаря тому, что прием бита в этот триггер происходит в середине битового интервала, когда переходные процессы сигнала $DLINE^*$ уже закончились. Остаточный джиттер сигнала $SDIN$ на выходе триггера $D4$ определяется неидеальностью сигнала CLK^* на выходе генератора PLL с фазовой автоподстройкой частоты. Исходные состояния триггеров $D3—D5$ произвольны.

Генератор PLL предназначен для формирования высокостабильного синхросигнала CLK^* на основе непрерывного слежения за входным сигналом $DLINE^*$. Фронт сигнала CLK^* привязан к моментам изменения сигнала $DLINE^*$, спад CLK^* формируется в середине битового интервала сигнала $DLINE^*$, что соответствует его установившемуся значению.

Благодаря достаточной инерционности генератора PLL сигнал CLK^* практически нечувствителен к джиттеру сигнала $DLINE^*$ и иным его кратковременным искажениям, вызванным помехами в линии связи.

Дешифратор $DC1$ ($DC2$) предназначен для выделения в потоке скремблированных данных, проходящем через сдвиговый регистр $RG1$ ($RG2$), определенных кодов $CODE_1, CODE_2, \dots, CODE_J$. При обнаружении дешифратором $DC1$ ($DC2$) указанных кодов на его выходах (обведены овалом) формируется соответствующий M -разрядный код $LOAD_1, LOAD_2, \dots, LOAD_J$ для последующей параллельной загрузки сдвигового регистра $RG3$ ($RG4$). В данном примере построения системы $J = 4, M = 5$. При обнаружении любого кода $CODE_1, CODE_2, \dots, CODE_J$ дешифратор $DC1$ ($DC2$) формирует также единичный сигнал на входе P/S (P/S^*) управления режимом работы регистра $RG3$ ($RG4$), подготавливая его к параллельному приему данных по положительному фронту очередного синхроимпульса на входе C .

Усилитель $D6$ ($D7$) предназначен для передачи (приема) скремблированного сигнала данных в линию (из линии). Параметры усилителей $D6$ и $D7$ определяются типом линии связи, которая может быть выполнена в виде витой пары проводов, коаксиального или оптоволоконного кабеля и т. п.

Далее приведено описание работы системы.

На входы скремблера поступают данные $DATA$ и сопровождающий их сигнал синхронизации CLK . Фронты

сигнала CLK (моменты T_0, T_1, \dots, T_{18} на рис. 15) соответствуют границам между битовыми интервалами сигнала данных $DATA$. По фронтам сигнала CLK изменяется содержимое регистра $RG1$ (диаграмма сигнала $SDATA$), генератор F переходит в новое состояние. При этом формируется очередной псевдослучайный бит RND , который складывается по модулю два с битом данных $DATA$ и преобразуется в скремблированный бит данных $SCRD$. По окончании переходных процессов в момент формирования спада сигнала CLK бит $SCRD$ принимается в триггер $D3$ (диаграмма сигнала $DLINE$) и через усилитель $D6$ передается в линию связи.

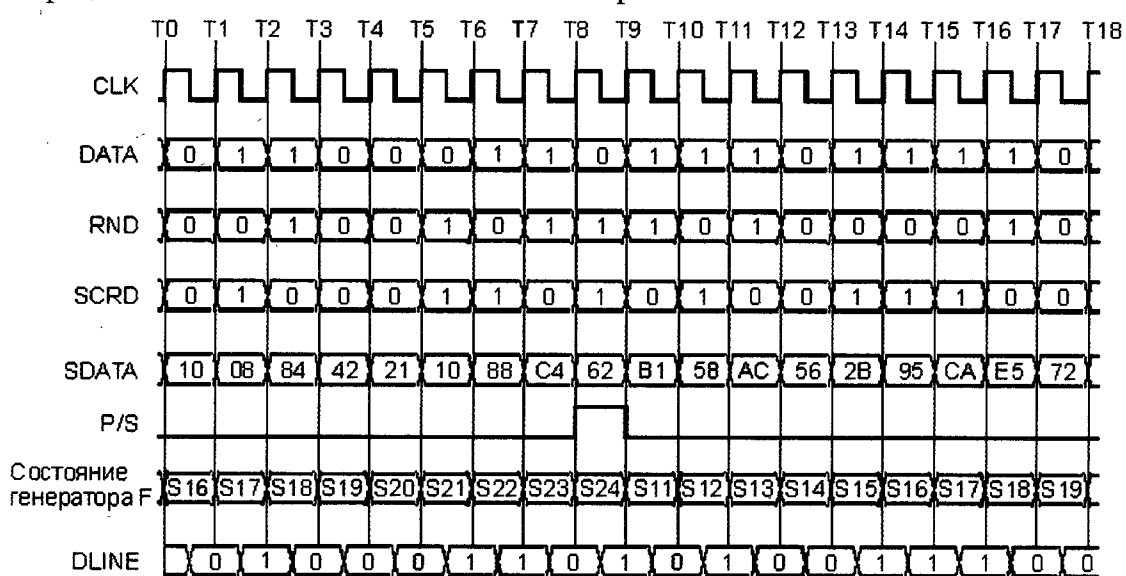


Рис. 15

В интервале времени $T_8—T_9$ дешифратор $DC1$ формирует сигнал лог. 1 на входе P/S управления режимом работы регистра $RG3$, подготавливая его к приему параллельных данных в момент T_9 .

В отсутствие параллельной загрузки генератор F псевдослучайной последовательности битов циклически проходит через ряд состояний $S_1, S_2, S_3, \dots, S_{31}, S_1, S_2$ и т. д., как показано в таблице и на диаграмме (рис. 14, а, б). В состоянии S_1 (первая строка таблицы, а также указатель A на диаграмме) в регистре $RG3$ хранится пятиразрядный двоичный код $11111_2 = 1F_{16}$, на выходе RND генератора F сформирован сигнал лог. 0. В следующем такте указатель A перемещается по часовой стрелке и фиксируется на соседней позиции, генератор F переходит в состояние S_2 , при котором $SRND = 01111_2 = 0F_{16}$, $RND = 0$ и т. д. Этот процесс циклически повторяется, указатель A вращается по кругу, последовательно проходя все возможные состояния S_i .

Параллельная загрузка регистра $RG3$ в произвольном такте приводит

к принудительной установке генератора в одно из заданных состояний, в данном примере в состояния S_3, S_{11}, S_{19} или S_{27} . Эти состояния предпочтительно выбираются так, чтобы на диаграмме (рис. 14, б) дуги $S_3—S_{11}, S_{11—S_{19}, S_{19—S_{27}}$ и $S_{27—S_3}$ имели примерно равную длину (линии B и C , которые разделяют окружность на четыре примерно равные части). В процессе работы системы указатель A сравнительно редко с равной вероятностью устанавливается в эти состояния, а в промежутках между такими установками продолжает равномерное перемещение по диаграмме в направлении по часовой стрелке.

Выбор нескольких (а не одного) заданных состояний, в которые генератор переходит в моменты его параллельной загрузки, целесообразен в тех случаях, когда число состояний генератора достаточно велико, и в течение полного оборота указателя A вероятность параллельной загрузки регистра $RG3$ близка к единице. Поэтому если указатель A периодически «срывается» с равномерного вращения в одно и то же заданное состояние, то вероятность того, что он успеет совершить хотя бы один полный оборот, становится невысокой. Иными словами, некоторые состояния генератора F будут использоваться реже, чем другие, а тогда отмеченные ранее при описании генератора (рис. 1) свойства «канонической» псевдослучайной последовательности битов будут в некоторой степени утеряны, что нежелательно. Наличие нескольких фиксированных точек установки, равномерно распределенных по диаграмме, выравнивает вероятности использования всех возможных состояний генератора F .

Как показано на диаграммах сигналов SDATA и P/S (рис. 15), одним из кодов, вызывающих принудительную установку генератора F в фиксированное состояние, является код SDATA = CODE₁ = 62₁₆ = 01100010₂. Этот код присутствует в регистре RG1 в интервале времени T8—T9 и, как уже отмечалось, дешифратор DC1 реагирует на него подготовкой регистра RG3 к приему параллельного кода LOAD₁. В данном примере этот код равен 0E₁₆ = 01110₂ и соответствует состоянию S11 генератора F (таблица на рис. 14, а). Таким образом, в момент T9 цепь последовательных переходов ... S16, S17, ..., S23, S24 разрывается и вместо перехода в очередное состояние S25 генератор F «перескакивает» в состояние S11. После этого формируется новая цепь последовательных переходов S11, S12, ..., S18, S19, ... — вплоть до возникновения очередной ситуации, при которой эта цепь разрывается, а затем образуется следующая цепь с одним из начальных состояний S3, S11, S19 или S27 и т. д.

Принятые из линии скремблированные данные DLINE* синхронизируют генератор PLL, в результате на его выходе формируется сигнал CLK* (рис. 16). Сигнал SDIN на выходе триггера D4 повторяет сигнал DLINE* с задержкой на половину периода синхросигнала, при этом сигнал SDIN, как уже отмечалось, практически не содержит фазовых искажений (джиттера). Скремблированные данные SDIN последовательно проходят через регистр RG2. После его заполнения данные SDATA* совпадают с данными SDATA в регистре RG1 скремблера.

Это следует из того, что, во-первых, источник данных для обоих регистров общий — выход элемента «Исключающее ИЛИ» D1 и, во-вторых, ничто не препятствует одновременному (с точностью до задержки передачи) заполнению обоих регистров RG1 и RG2 данными от этого источника. Так как дешифраторы DC1 и DC2 идентичны, а данные на их входах одинаковы, то сигналы на выходах этих дешифраторов также совпадают с точностью до задержки передачи. Поэтому рассмотренный ранее процесс установки генератора F в определенное состояние протекает также и в дескремблере. В интервале времени T8—T9 (рис. 16) на входе P/S* регистра RG4 формируется сигнал лог. 1, в момент T9 в регистр принимается параллельный код 0E₁₆, соответствующий состоянию S11.

Независимо от предыстории состояния генератора псевдослучайной последовательности битов дескремблера начиная с момента T9 этот генератор синхронизируется с генератором F скремблера в том смысле, что формируемые обоими генераторами последовательности битов совпадают. Неопределенные состояния и сигналы в начальный период, когда кодовая синхронизация между генераторами отсутствовала, помечены на диаграммах (рис. 16) символами X.

Начиная с момента T9 скремблирующая RND и дескремблирующая RND* последовательности битов совпадают, поэтому сигнал DIN дескремблированных данных совпадает с сигналом DATA на входе скремблера. Выходной сигнал DATA* данных, «очищенный» от возможных много-

кратных переключений на границах между битовыми интервалами, поступает на выход устройства и сопровождается сигналом CLK*. Таким образом, входные сигналы DATA и CLK преобразуются в совпадающие с ними (с точностью до задержки передачи) выходные сигналы DATA* и CLK*.

Частота следования моментов синхронной установки регистров RG3 и RG4 в одинаковые состояния (моментов синхронизации) зависит от скорости передачи данных, а также от разрядности и числа J кодов CODE₁, CODE₂, ..., CODE_J, распознаваемых дешифраторами DC1 и DC2.

При J = 1 и разрядности регистра RG1 (RG2), равной 20 бит, в скремблированном потоке данных в среднем в каждой цепи из 2²⁰ ≈ 10⁶ бит будет встречаться один искомый код, равный CODE₁. При скорости передачи данных, равной 1 Мбит/с, средняя частота следования моментов синхронизации составляет примерно 1 Гц. При J = 4 частота моментов синхронизации увеличивается в четыре раза.

Для уменьшения вероятности ложного распознавания кодов CODE₁, CODE₂, ..., CODE_J дешифратором DC2 дескремблера в связи с поступлением из линии связи ошибочных битов разрядность регистров RG1 и RG2 можно увеличить. Применение предлагаемой системы «скремблер–дескремблер» позволяет повысить скорость передачи полезных данных и уменьшить их потери при восстановлении нарушенной синхронизации благодаря исключению из потока данных служебной синхронизирующей информации.

Борис Шевкопляс,
г. Москва

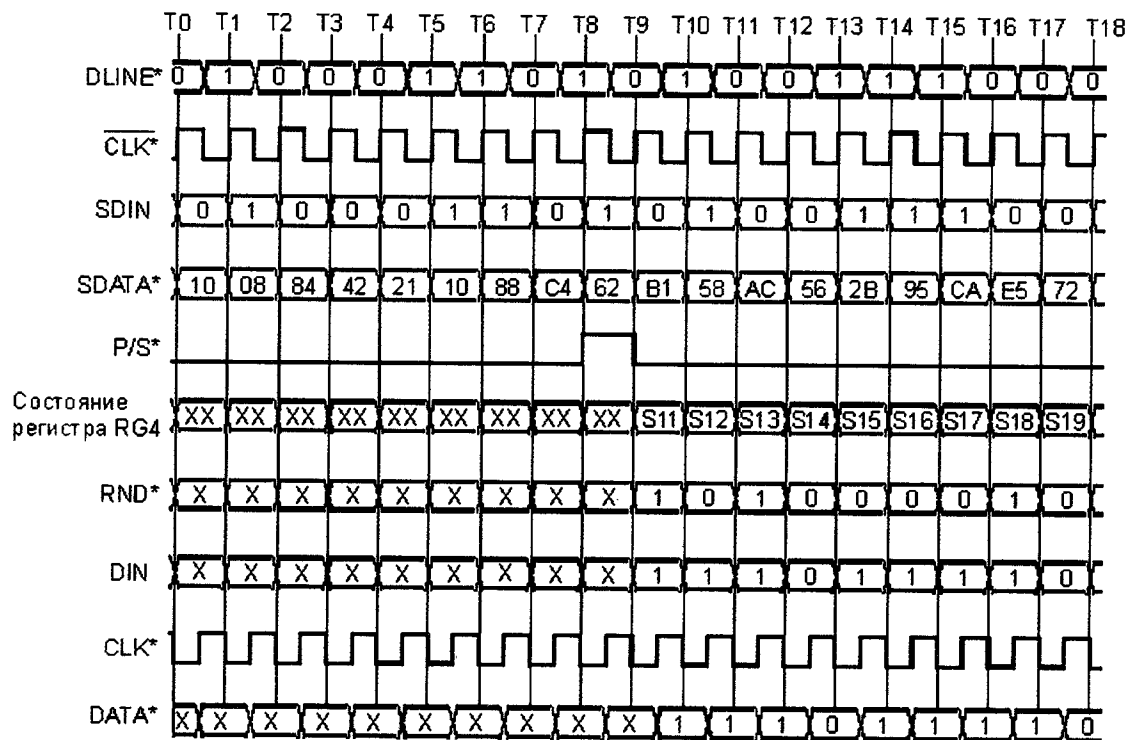


Рис. 16

Редакция журнала «Схемотехника»

приглашает авторов
к сотрудничеству

по всем вопросам обращаться
e-mail: editor@dian.ru
тел./факс (095)777-12-15

Требования по оформлению
статей см. в № 12, 2002, с. 44 и
на сайте редакции www.dian.ru.

Гонорары выплачиваются
авторам, проживающим на
территории СНГ.

Повышение точности измерительного трансформатора тока

В статье изложен простой способ снижения угловой и токовой погрешности датчика переменного тока на основе трансформатора тока путем введения следящей обратной связи по магнитному потоку.

Измерительные трансформаторы широко используются в качестве датчиков переменного тока вследствие сравнительно низкой стоимости, хорошей изоляции измерительной цепи и относительно высокой точности.

Угловая и токовая погрешности трансформатора обусловлены неточностью намотки обмоток (несоответствие числа витков первичной и вторичной обмоток) и потерями в ферромагнитном сердечнике. Неточность намотки может быть компенсирована путем калибровки с последующим запоминанием калибровочных коэффициентов, а потери в сердечнике нелинейно зависят от первичного тока и их учет оказывается затруднительным.

Схема трансформатора тока приведена на рис. 1. Трансформатор нагружен на сопротивление Z_H . Схема замещения трансформатора тока приведена на рис. 2.

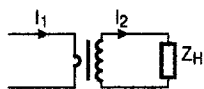


Рис. 1

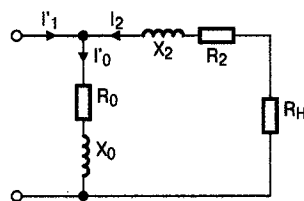


Рис. 2

Процессы в трансформаторе описываются следующей системой уравнений. Уравнение намагничивающих сил:

$$I_1 w_1 + I_2 w_2 = I_0 w_1 \text{ или } I_1 + I_2 = I_0'; \quad (1)$$

уравнение вторичной обмотки:

$$E_2 = I_2 (R_2 + jX_2) + I_2 (R_H + jX_H); \quad (2)$$

уравнение ЭДС вторичной обмотки:

$$E_2 = -j\omega w_2 \dot{\Phi} \text{ или } E_2 \approx -I_0' (R_0 + jX_0), \quad (3)$$

где w_1 и w_2 — число витков первичной и вторичной обмоток, Φ — магнитный поток, ω — круговая частота

измеряемого тока, R_0 и X_0 — параметры цепи намагничивающего тока схемы замещения трансформатора.

Этим уравнениям соответствует векторная диаграмма, представленная на рис. 3.

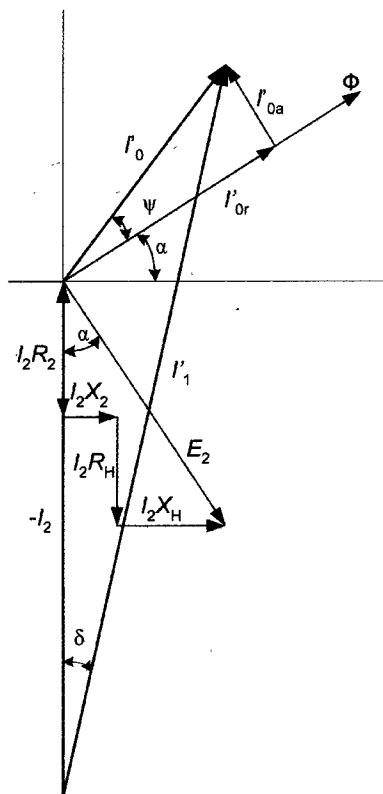


Рис. 3

Из векторной диаграммы видно, что относительную комплексную погрешность трансформатора тока можно найти по следующей формуле:

$$f = \frac{I_1' + I_2'}{I_1'} = \frac{I_0'}{I_1'}. \quad (4)$$

Угловая погрешность равна углу δ , а токовая погрешность определяется по формуле:

$$f_i = \frac{I_1' - I_2}{I_1'}$$

Как следует из векторной диаграммы, угловая и токовая погрешности непосредственно определяются током намагничивания I_0 , который нелинейно зависит от измеряемого тока и нагрузки трансформатора. Можно уменьшить этот ток, снизив сопротивление нагрузки. В этом случае, однако, уменьшится выходное напряжение датчика. Радикально снизить погрешность можно, уменьшив ток

намагничивания практически до нуля путем введения следящей обратной связи по магнитному потоку.

Схема трансформатора тока со следящей обратной связью приведена на рис. 4.

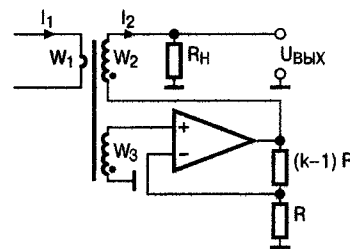


Рис. 4

В обмотке w_3 возникает ЭДС, пропорциональная магнитному потоку сердечника трансформатора. Усилитель в цепи отрицательной обратной связи создает в обмотке w_2 ток, намагничивающая сила которого стремится уменьшить магнитный поток. Этот ток равен

$$I_2 = \frac{kE_3 + E_2}{R_H + Z_2}, \quad (5)$$

где $Z_2 = R_2 + jX_2$ — комплексное сопротивление вторичной обмотки. Подставив в (5) значения E_2 и E_3 из формулы (3) получим после преобразований приближенное значение комплексной погрешности

$$- \frac{I_0'}{I_2} = \frac{w_2 (R_H + Z_2)}{(k w_3 + w_2) (R_0 + jX_0)} \approx f. \quad (6)$$

Из последней формулы видно, что увеличение коэффициента усиления k позволяет почти пропорционально снизить погрешность датчика тока. Следует иметь в виду, что формула (6) оценочная, т. к. R_0 и X_0 нелинейно зависят от магнитного потока сердечника.

Поскольку усилитель охвачен сложной обратной связью, он может возбуждаться. Простым средством стабилизации является включение между обмоткой обратной связи (w_3) и входом ОУ интегрирующей RC-цепочки, частота среза которой выбирается в диапазоне 10...40 кГц в зависимости от типа используемого усилителя.

Эксперименты, проведенные авторами с трансформаторами тока на частоте 50 Гц дали следующие результаты.

Трансформатор на кольцевом сердечнике из аморфного железа:

- первичная обмотка w_1 , вит. 1
- вторичная обмотка w_2 , вит. 1000
- обмотка обратной связи w_3 , вит. 300
- ток первичной обмотки, А 5
- угловая погрешность без обратной связи, град. 0,67

- угловая погрешность с обратной связью, град. $-0,012 \pm 0,004$.
 - коэффициент усиления усилителя обратной связи 100
- Трансформатор на броневом сердечнике Ч26 из феррита марки М2000НМ:
- первичная обмотка w_1 , вит. 1
 - вторичная обмотка w_2 , вит. 1000

- обмотка обратной связи w_3 , вит. 200
- ток первичной обмотки, А 5
- угловая погрешность без обратной связи, град. 1,3
- угловая погрешность с обратной связью, град. $-0,012 \pm 0,003$
- коэффициент усиления усилителя обратной связи 100

ЛИТЕРАТУРА:

1. *Isolated Current and Voltage Transducers. Characteristics — Applications — Calculations. Published by LEM Coporate Communications. — LEM Geneva, Switzerland, 1996.*

Георгий Волович,
Евгений Савченко,
г. Челябинск

Генератор с шунтирующим диодом

Генератор с шунтирующим диодом лет пятьдесят назад предложил известный специалист в области радиотехники Е. Г. Момот. В то время радиотехника была в основном электровакуумная. Тем не менее, данное схемотехническое решение применимо и в наши дни при современной элементной базе.

Во всяком генераторе, построенном на основе резонансного усилителя с колебательным контуром, после его включения наблюдается рост амплитуды колебаний. Амплитуда растет до тех пор, пока в генераторе не появятся какие-либо нелинейные явления, уменьшающие усиление используемого усилителя. Естественно, поскольку усилитель входит в нелинейный режим, появляется искажения формы генерируемого сигнала. На контуре искажения заметны мало, так как высокодобротный контур подавляет гармоники, но если сигнал снимается с другой точки генератора, они могут быть весьма заметными.

Идея предложенного Момотом генератора достаточно проста. Колебательный контур, задающий частоту генератора, шунтируется диодом, на который подано запирающее напряжение. Пока амплитуда колебаний на контуре меньше этого напряжения, диод закрыт и не оказывает влияния на работу генератора. Однако, как только амплитуда колебаний превысит это напряжение, диод начинает открываться, и в контур вносятся потери. Дальнейший рост амплитуды прекращается, происходит стабилизация амплитуды колебаний на колебательном контуре. При этом усилитель работает в линейном режиме, и сигнал, снимаемый не с колебательного контура, может иметь малые искажения. Кроме того, работа усилителя в линейном режиме позволяет повысить стабильность частоты генерируемых колебаний.

На рис. 1 показан пример схемы генератора, собранного по схеме ем-

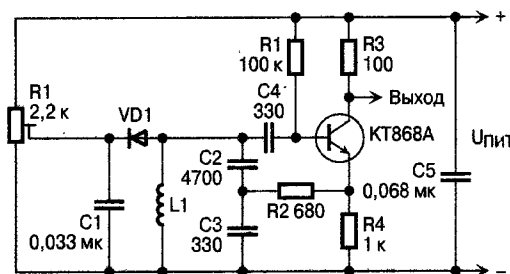


Рис. 1

костной трехточки с шунтирующим диодом.

Необходимое напряжение смещения на диоде регулируется построечным резистором R1. Выходной сигнал снимается с резистора R3 в цепи коллектора транзистора VT1 и его амплитуда может быть установлена в диапазоне 50...200 мВ. Подобное решение чаще всего используется в гетеродине супергетеродинного приемника, и сигнал с него обычно подается на смеситель. Как известно, крутизна преобразования и уровень шумов смесителя существенно зависят от амплитуды сигнала гетеродина. Применение гетеродина с шунтирующим диодом позволяет оптимизировать режим работы смесителя. Кроме того, стабильность частоты такого узла может быть выше, чем без шунтирующего диода, благодаря работе транзистора в линейном режиме. Так, исследование этого схемотехнического решения показало, что изменение напряжения питания в диапазоне 5...10 В при стабилизированном напряжении смещения на диоде, для чего на потенциометр подавалось напряжение от другого стабилизированного источника питания, приводило к из-

менению частоты на 0,025 %, а при отключенном диоде — на 0,5 %. С шунтирующим диодом амплитуда выходного напряжения изменялось на 1 %, без диода — более чем в два раза. Искажение формы выходного сигнала с диодом на экране электронного осциллографа почти незаметно, а без диода вершина синусоиды явно срезана.

На рис. 2 показана схема низкочастотного генератора, выполненного на гираторе с шунтирующим диодом.

Как известно, низкочастотный гиратор на своем входе имитирует высокодобротную индуктивность. В генераторе она совместно с конденсатором C2 образует колебательный контур. Добавление цепи положительной обратной связи через резистор R5 превращает его в генератор гармонических колебаний. Частота генерации определяется по формуле:

$$f = \frac{1}{2\pi\sqrt{R_3 R_6 C_2 C_3}}$$

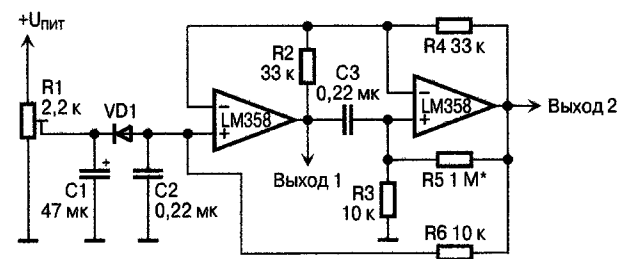


Рис. 2

Добавление шунтирующего диода дает возможность заметно уменьшить нелинейные искажения выходного сигнала. Без диода они имеют величину порядка 2 %, а с диодом уменьшаются до 0,2 %.

ЛИТЕРАТУРА:

1. Г. П. Петин. *Применение гиратора в схемах резонансных усилителей и генераторах.* — Радио, 1996, № 11, с. 33, 34.

Генри Петин,
г. Ростов-на-Дону

Компьютерное схемотехническое моделирование электронных устройств

В данной статье рассматривается моделирование простейших тиристорных регуляторов напряжения средствами программы Micro-Cap (MC).

МОДЕЛИРОВАНИЕ ТИРИСТОРНЫХ РЕГУЛЯТОРОВ НАПЯЖЕНИЯ В ПРОГРАММЕ MC

Поскольку тиристоры открываются при вполне определенном значении тока управляющего электрода, это используется для управления моментом их включения. Распространенным является фазовый способ управления.

Соберем простейшее устройство в программе MC (рис. 104), используя тиристор SCR, параметры которого заданы в окне на рис. 105, что соответствует текстовому заданию в виде:

```
MODEL SCR (50m, 40m, 1u, 55, 50MEG, 30u, 105, 1)
```

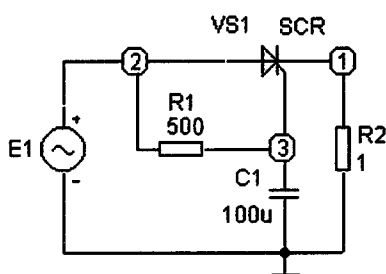


Рис. 104

На аноде тиристора действует переменное напряжение, заданное источником E1, параметры которого в окне Text таковы:

```
MODEL 50 SIN (F=50 A=50)
```

Зададим параметры режима моделирования установками в соответствии с окном на рис. 106.

Результат моделирования показан на рис. 107.

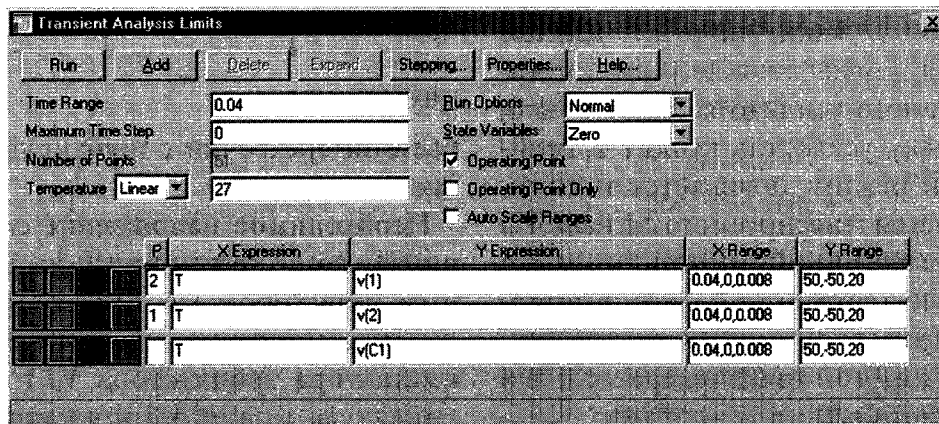


Рис. 106

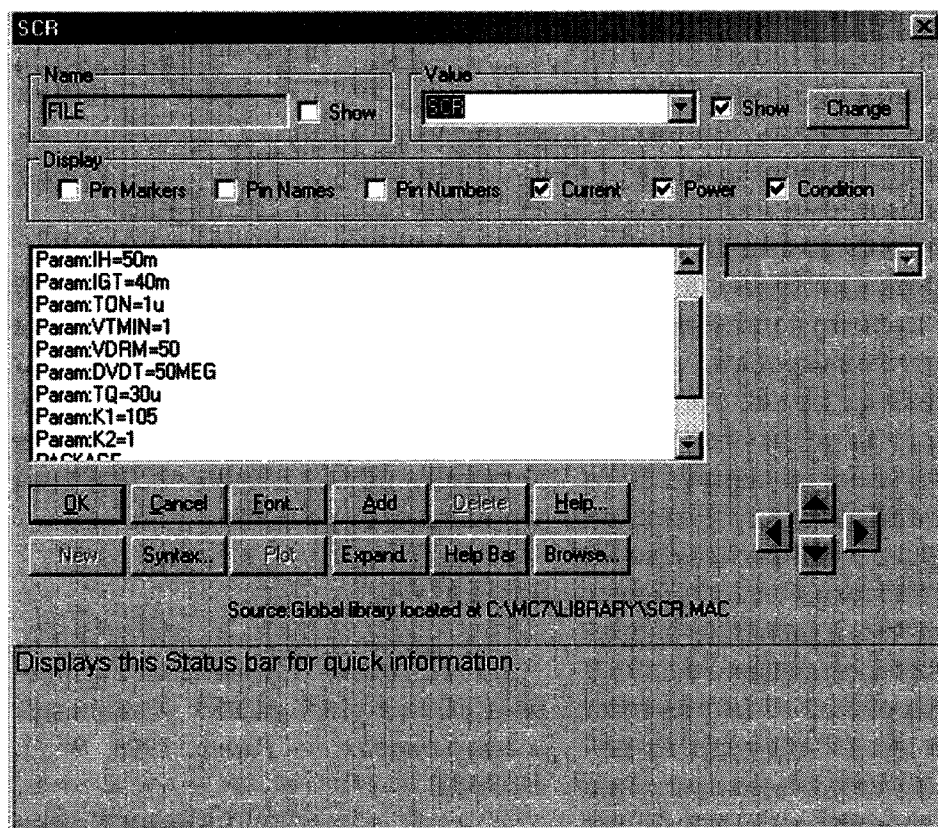


Рис. 105

В данной цепи (рис. 104) сигнал на управляющем электроде 3 по сравнению с сигналом на аноде тиристора 2 имеет временной (фазовый) сдвиг, зависящий от постоянной времени цепи R1C1. В результате этого момент открытия тиристора смещается во времени, и его можно регулировать, изменяя сопротивление резистора R1.

Проведем дополнительно моделирование, задав дискретные значения параметра R1 в окне Stepping по рис. 108.

На полученных осциллограммах (рис. 109) виден результат.

При регулировании как бы «отсекается» часть полуволн напряжения, что приводит к уменьшению его среднего значения на нагрузке R2, а, следовательно, регулируется средний ток и средняя мощность в ней. Сказанное наглядно проявляется на осциллограммах (рис. 109), где большему значению R1 соответствует перемещение вправо начала открытия тиристора в семействе кривых.

Тиристором можно также управлять (рис. 110), подавая в опреде-

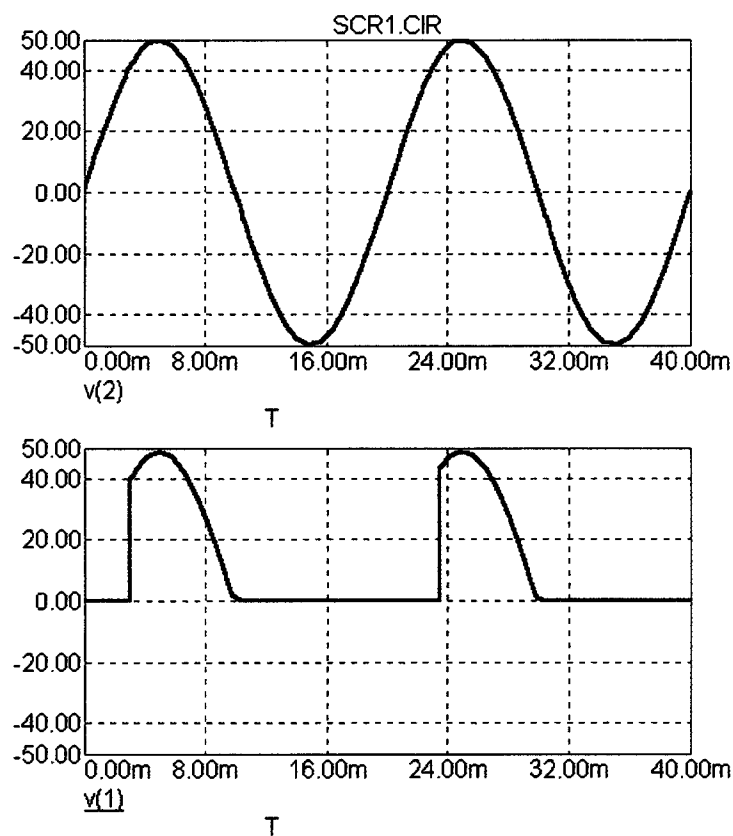


Рис. 107

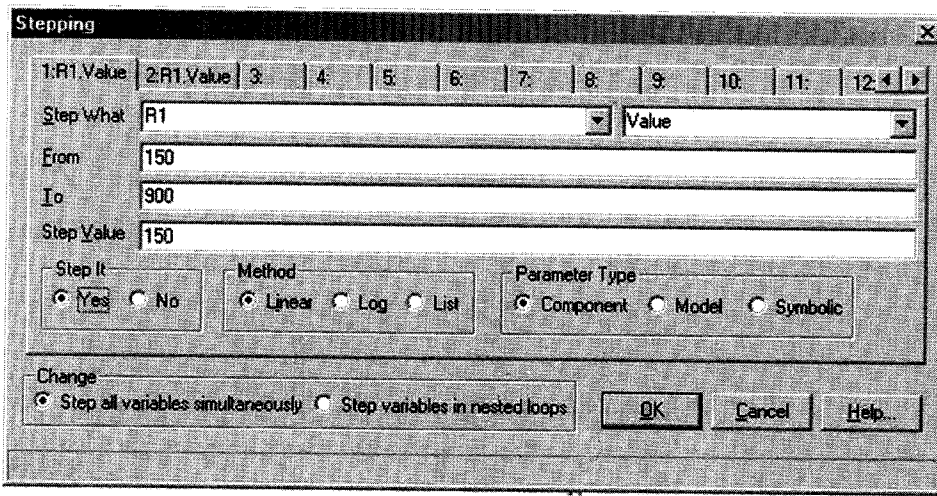


Рис. 108

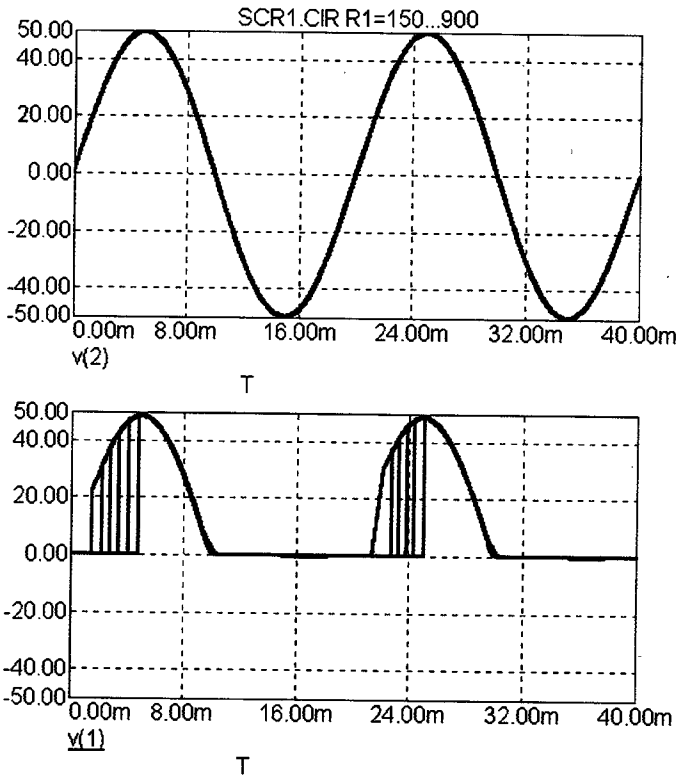


Рис. 109

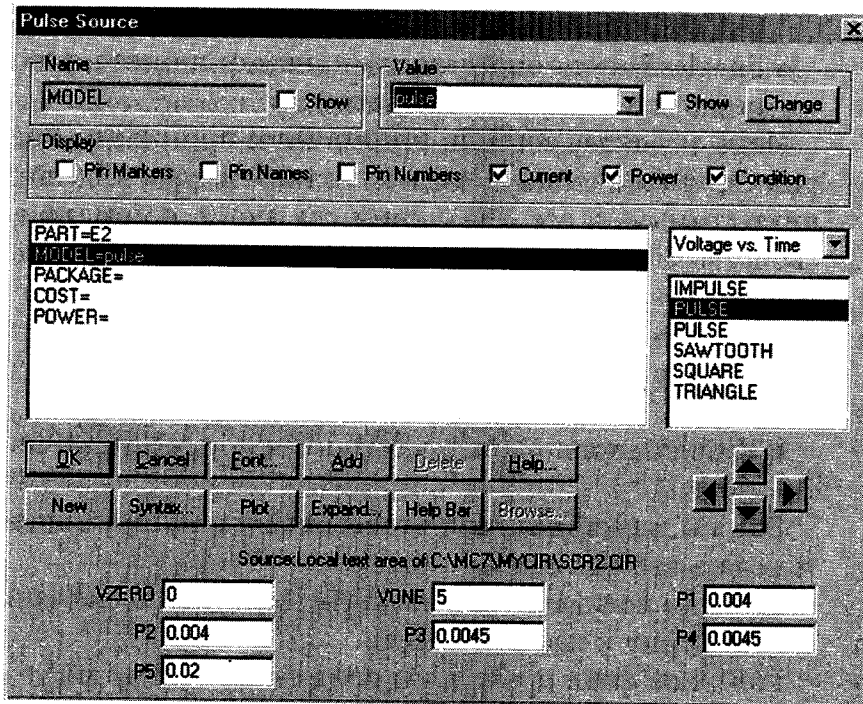


Рис. 111

ленные моменты времени открывающие его импульсы от источника E2 с параметрами, показанными на рис. 111.

Установки моделирования и осциллограммы показаны соответственно на рис. 112, 113.

Аналогично можно управлять и симистором (рис. 114), в качестве которого здесь использован библиотечный компонент Triac (рис. 115).

Для управления при отрицательных напряжениях использован импульсный источник E3, являющийся копией E2,

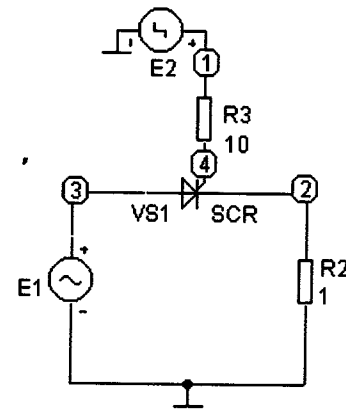


Рис. 110

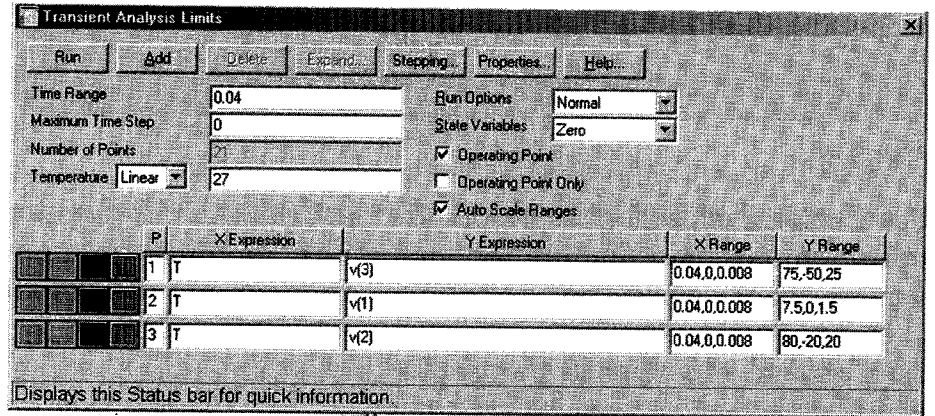


Рис. 112

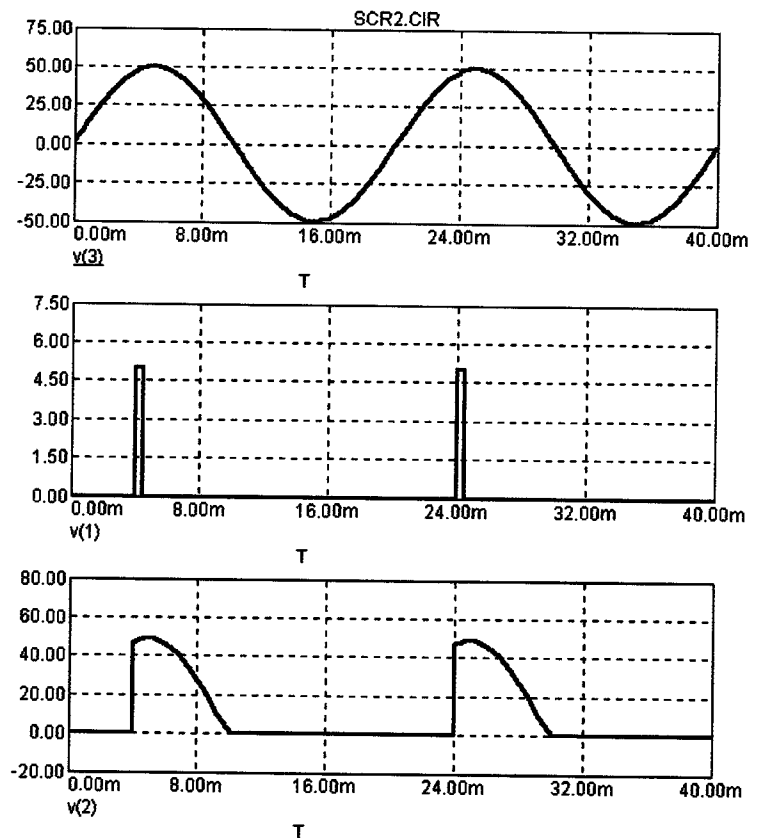


Рис. 113

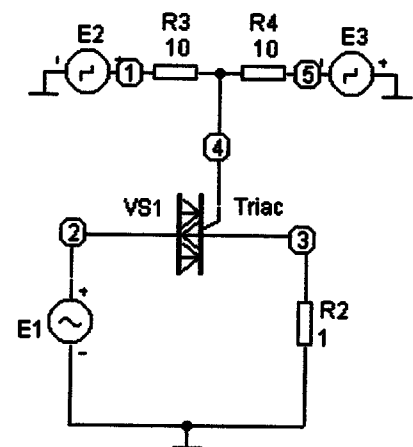


Рис. 114

включенного с обратной полярностью (положительный вывод питания соеди-

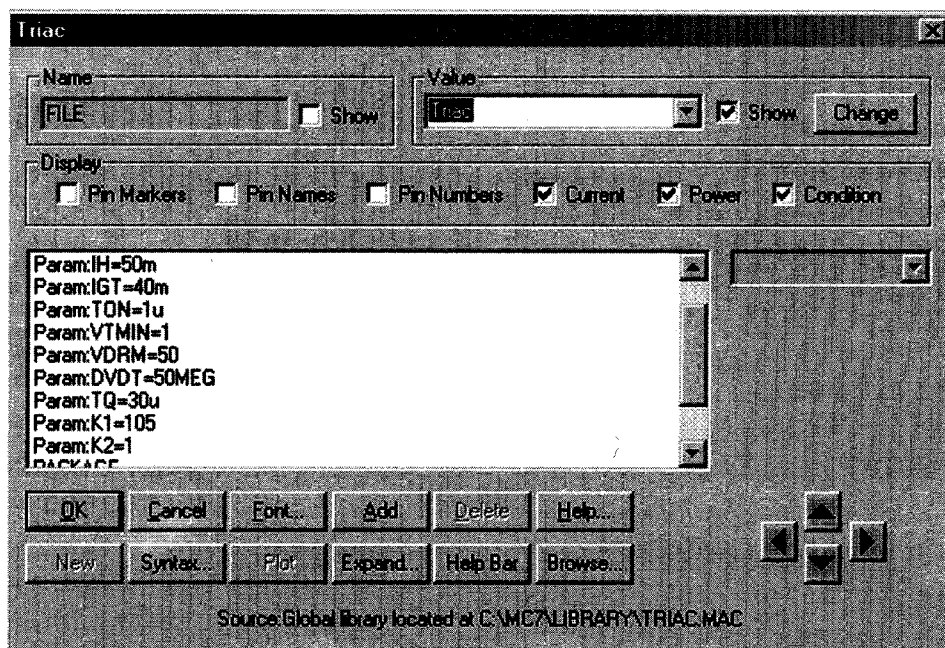


Рис. 115

нен с общим проводом). В результате происходит управление переменным напряжением в обоих полупериодах.

Осциллограммы в этой цепи показаны на рис. 116, причем на средней осциллограмме совмещены графики

импульсных источников E2 и E3, соответствующие их выходным напряжениям V(1) и V(5).

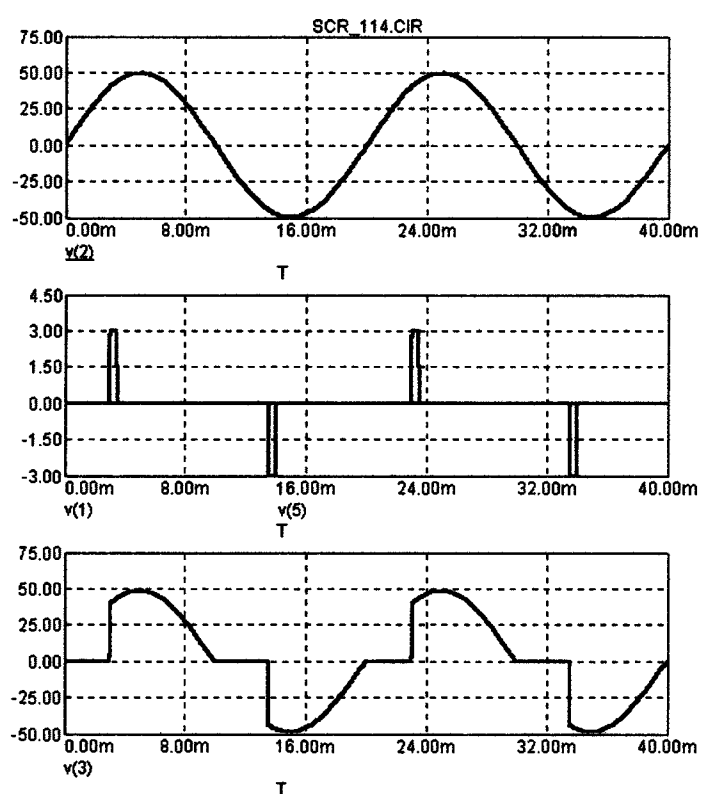


Рис. 116

Генрих Кардашев,
г. Москва

Продолжение следует

Информация: № 2-2005

Пиктограммы в области System Tray

Теперь давайте усложним задачу. Создадим полезную программу, например, CD-плеер. При запуске он сразу попадает в трей, начав проигрывать треки с музыкального диска. В этой программе мы создадим меню, которое будет открываться при нажатии правой кнопкой на пиктограмме.

Создание проигрывателя основано на компоненте MediaPlayer вкладки System. Его общий вид показан на рис. 4.



Рис. 4

Рассмотрим этот компонент. Он представляет собой набор кнопок. Их назначение:

- воспроизведение, btPlay — воспроизведение звука или видео;
- пауза, btPause — приостановка воспроизведения;
- стоп, btStop — остановка воспроизведения;
- следующий, btNext — переход к следующему кадру;
- предыдущий, btPrev — переход к предыдущему кадру;
- шаг, btStep — переход к следующему фрагменту (дорожке CD);
- назад, btBack — переход к предыдущему фрагменту (дорожке CD);
- запись, btRecord — запись;
- открыть/закрыть, btEject — открытие или закрытие дисководов.

Среди его свойств нужно выделить следующие:

- Display — определяет компонент, на поверхности которого будет воспроизводиться видеоролик;
- Filename — имя проигрываемого файла;

- AutoOpen — признак автоматического открытия файла сразу после запуска программы;
- DeviceType — тип устройства, с которым будет работать компонент;
- VisibleButtons определяет кнопки компонента, которые будут видны пользователю;
- EnabledButtons определяет кнопки компонента, которые будут доступны пользователю.

Итак, за основу возьмем предыдущую программу. Уберем с формы кнопки «Создать пиктограмму» и «Удалить пиктограмму» и все, что с ними связано, т. е. ссылки на свойство Enabled. Поместим на форму компонент MediaPlayer и изменим его свойства следующим образом:

- Name = MPlayer
- AutoOpen = True
- DeviceType = dtCDAudio
- VisibleButtons = [btPlay, btPause, btStop, btNext, btPrev, btEject]

Это мы делаем для того, чтобы связь с диском устанавливалась автоматически.

Еще нам понадобится компонент PopupMenu, добавим его на форму и назовем PMenu.

Головной файл программы приведем к такому виду:

```
program Player;
uses
  Forms,
  u_player in 'u_player.pas' {FMain};
{$R *.RES}
begin
```

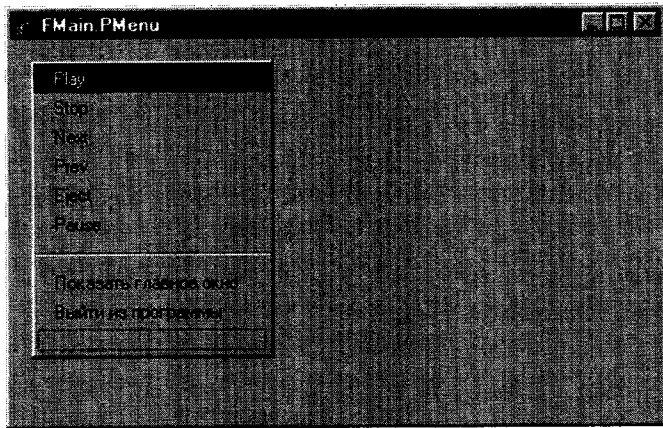


Рис. 5

```
Application.Initialize;
Application.CreateForm(TFMain, FMain);
FMain.Visible := false;
Application.ShowMainForm := false;
Application.Run;
end.
```

Для того, чтобы убрать форму с экрана, мы воспользовались первым из рассмотренных способов.

Обработчик нашего сообщения станет теперь таким:

```
procedure TFMain.Mess(var Msg: TMessage);
var P: TPoint;
begin
  If Msg.Msg = TrayIcon then
  If Msg.lParam = WM_LBUTTONDOWN then
  begin
    FMain.Show;
    SetForegroundWindow(Application.Handle);
  end else
  If Msg.lParam = WM_RBUTTONDOWN then
  begin
    GetCursorPos(P);
    PMenu.Popup(P.X, P.Y);
  end;
end;
```

Теперь здесь предусмотрен случай, когда пользователь нажал в области пиктограммы правую кнопку мыши и еще не отпустил ее. В этом случае появляется меню, но о нем позже.

Обработчикам событий OnCreate и OnClose формы придадим такой вид:

```
procedure TFMain.FormCreate(Sender: TObject);
begin
  with MyIcon do
  begin
    cbSize := SizeOf(TNotifyIconData);
    Wnd := Handle;
    uID := 1;
    uFlags := NIF_ICON or NIF_MESSAGE or NIF_TIP;
    uCallbackMessage := TrayIcon;
    hIcon := Application.Icon.Handle;
    szTip := 'Player';
  end;
  Shell_NotifyIcon(NIM_ADD, @MyIcon);
  MPlayer.Play;
end;
```

Здесь мы добавили автоматический запуск первого трека.

```
procedure TFMain.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  MPlayer.Stop;
  MPlayer.Close;
```



Рис. 6

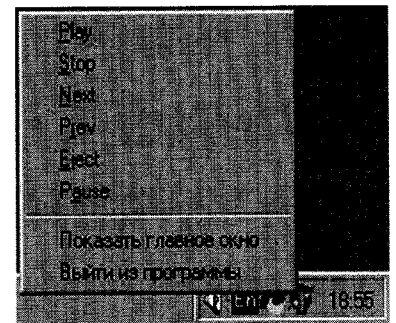


Рис. 7

```
Shell_NotifyIcon(NIM_DELETE, @MyIcon);
end;
```

Перед закрытием приложения программа перестает проигрывать треки.

Теперь займемся меню. Нажмите два раза левую кнопку мыши на компоненте PopupMenu, приведите меню к виду, показанному на рис. 5.

Создайте обработчики события OnClick для каждого пункта меню, нажав на каждом из них левую кнопку мыши два раза. Приведите их к такому виду:

```
procedure TFMain.N2Click(Sender: TObject);
begin
  FMain.Show;
  SetForegroundWindow(Application.Handle);
end;

procedure TFMain.N3Click(Sender: TObject);
begin
  FMain.Close;
end;

procedure TFMain.Play1Click(Sender: TObject);
begin
  MPlayer.Play;
end;

procedure TFMain.Stop1Click(Sender: TObject);
begin
  MPlayer.Stop;
end;

procedure TFMain.Next1Click(Sender: TObject);
begin
  MPlayer.Next;
end;

procedure TFMain.Prev1Click(Sender: TObject);
begin
  MPlayer.Previous;
end;

procedure TFMain.Eject1Click(Sender: TObject);
begin
  MPlayer.Eject;
end;

procedure TFMain.Pause1Click(Sender: TObject);
begin
  MPlayer.Pause;
```

Здесь мы дублируем кнопки компонента MediaPlayer и добавляем две новые кнопки.

Программа готова. Результат ее работы показан на рис. 6 и 7.

Михаил Бырдин,
г. Заречный Пензенской обл.

Borland C++ Builder 6 для начинающих

В этой статье рассказывается об основных свойствах визуальных компонентов.

СВОЙСТВА КОМПОНЕНТОВ

Рассмотрим основные свойства компонентов среды разработки Borland C++ Builder 6. При разработке программы удобно иметь справочную таблицу этих свойств в алфавитном порядке. Это позволяет оперативно находить нужное свойство для того, что бы узнать его назначение. В табл. 1 описаны основные свойства наиболее часто используемых компонентов Borland C++ Builder 6. Эти свойства сходны по смыслу и назначению для различных компонентов.

Большинство компонентов среды разработки VCB6 имеет одинаковые свойства, определяющие их имя, заголовки, размеры, скрытость и т. д. Познакомившись с одним из таких свойств на примере одного компонента, можно понять его назначение и для другого компонента. Поэтому мы будем рассматривать основные свойства компонентов, не привязываясь к конкретному их типу, а делая это лишь в исключительных случаях.

Самыми распространенными значениями, которые могут иметь большинство свойств, являются

True (истина) и False (ложь) или просто «да» и «нет», т. е. если свойство принимает значение True, оно активизируется. При значении False это свойство блокируется. Свойства любого компонента можно изменить во время создания программы с помощью инспектора объектов, при этом для большинства свойств эти изменения можно зрительно увидеть по тому, как изменится при этом сам компонент. В предыдущих примерах программ уже показывалось, как изменения свойств компонента влияют на его форму, цвет и т. п.

Все свойства объекта доступны не только на этапе проектирования, но и во время работы программы. Их значения можно изменять и в ходе ее выполнения в программном коде. Это также было показано в предыдущих примерах программ. Например, строка программы:

```
Image1->Picture->LoadFromFile(«PICTURE/1.bmp»);
```

Таблица 1

Свойство	Назначение
ActiveControl	Определяет, какой элемент управления на форме будет выделен при запуске программы.
Align	Выравнивание объекта внутри родителя. Принимает значения: alNone — нет выравнивания; alTop — по верхней границе; alBottom — по нижней границе; alLeft — по левой границе; alRight — по правой границе; alClient — по всему размеру родителя.
Alignment	Горизонтальное выравнивание: taLeftJustify — по левому краю; taRightJustify — по правому; taCenter — по центру.
Anchors	Определяет, как объект будет перемещаться или сжиматься при изменении размеров родителя. Например, если включить два подсвойства alLeft и alRight, то объект будет сжиматься при уменьшении его ширины. По умолчанию не используется.
AutoScroll	Если равно true, то разрешается автоматическое появление полос прокрутки, когда содержимое объекта не умещается в его видимой части.
AutoSize	Если равно true, то размер объекта будет автоматически подстраиваться под размер его содержимого.
BiDiMode	Способ поведения объектов с учетом национальной специфики — ввод текста и прокручивание информации слева направо (bdLeftToRight по умолчанию) или справа налево (bdRightToLeft).
BorderIcons	Набор значков в системном меню формы.
BorderStyle	Вид рамки объекта. Принимает значения bsNone — нет рамки; bsSingle — простая рамка; bsSizeable — рамка, позволяющая изменять размеры объекта мышью; bsDialog — рамка в стиле диалоговых окон; bsToolWindow — как bsSingle, но с небольшим заголовком; bsSizeToolWin — как bsSizeable, но с небольшим заголовком.
BorderWidth	Ширина рамки
Cancel	Определяет, будет ли происходить для данного объекта событие OnClick, когда пользователь нажмет клавишу Esc. Для этого значение Cancel должно быть равно true.
Caption	Заголовок. Для одних объектов применяется, чтобы задать заголовок в окне или надпись на кнопке, для других — описывает их содержимое (например, у полей надписи).
Checked	Свойство, определяющее состояние флажков (true — включен).
Color	Цвет объекта.
Columns	Число столбцов.
Constraints	Содержит четыре подсвойства, определяющие минимально и максимально допустимые размеры объекта.
Ctl3D	Вид объекта в объемном стиле (стандарт для Windows 95 и выше).
Cursor	Задаёт вид указателя, который отображается при наведении на объект.
Default	Определяет, будет ли происходить для данного объекта событие OnClick, когда пользователь нажмет клавишу Enter. Для этого свойство Default должно иметь значение true.
DockSite	Применяется для различных панелей, объектов группирования и т. д. Когда включено (т. е. true), позволяет использовать этот объект как базу для стыковки других объектов на форме при перетаскивании. Обеспечивает режим «прилипания» (см. свойство DragKind).
Down	Состояние кнопки. Нажата — true.
DragCursor	Задаёт вид указателя, который отображается при перетаскивании объекта мышью.
DragKind	Определяет, можно ли объект произвольно перетаскивать по окну (dkDrag) или же его можно перемещать как стыкуемый объект (dkDock), который сам определяет свою форму при стыковке с другими объектами.
DragMode	Определяет, можно ли объект перетаскивать (dmAutomatic) или нет (dmManual — по умолчанию) в соответствии с режимом, указанным в DragKind.
Enabled	Доступность объекта. Когда свойство Enabled имеет значение false, объект становится недоступным для пользователя (например, кнопка делается серой и щелкнуть на ней нельзя).
FileName	Имя файла. Используется при сохранении выбранного файла в стандартных диалоговых окнах, для указания имени файла с мультимедийным содержанием. Применяется для компонентов воспроизведения музыки или звука и т. д.
Flat	Вид границ объекта (true — плоские границы).
Font	Определяет шрифт, которым будут делаться все надписи внутри объекта. Содержит множество подсвойств.
FormStyle	Стиль формы. Может принимать значения: fsNormal — обычное окно; fsMDIChild — внутренне окно многооконного приложения; fsMDIForm — родительское окно многооконного приложения; fsStayOnTop — всегда на поверхности экрана.
Glyph	Графический файл, содержащий от одного до четырех (отпущено, недоступно, нажато, выделено) изображений графической кнопки.
Height	Высота объекта.

Продолжение таблицы 1

Свойство	Назначение
HelpContext	Номер экрана справочной системы, который будет вызываться при нажатии клавиши F1 при работе с объектом. Требует знания устройства справочной системы Windows.
HelpFile	Имя файла справочной системы для данного приложения или окна.
HorzScrollBar	Вид и стиль горизонтальной полосы прокрутки. Состоит из множества подсвойств.
Hint	Текст подсказки, которая всплывает при наведении указателя мыши на объект. Эта подсказка будет показываться, если свойство ShowHint установлено в true.
Icon	Имя файла со значком для приложения.
Images	Список картинок, которые будут использоваться для показа кнопок на панели инструментов, в пунктах меню и т. д.
ItemIndex	Текущий выбранный элемент в списке Items. Нумерация начинается с нуля. Если ничего не выбрано, то значение ItemIndex равно -1.
Items	Список объектов, хранящих основные данные. Используется во всевозможных списках, меню и наборах.
Kind	Вид (тип) объекта. Для разных компонентов может принимать разные значения.
Left	Левая координата объекта на родительском объекте.
Lines	Список строк. Используется в многострочных текстовых элементах. По структуре походит на свойство Items.
Max	Максимальное значение диапазона, используемого во всевозможных компонентах прокрутки, ползунках и т. п.
MaxLength	Максимально допустимая длина вводимой строки символов.
Min	Минимальное значение диапазона, используемого во всевозможных компонентах прокрутки, ползунках и т. п.
ModalResult	Значение, которое возвращает модальное диалоговое окно при его закрытии. Может принимать значения mrNone (по умолчанию), mrOk, mrCancel, mrAbort, mrRetry, mrIgnore, mrYes, mrNo и mrAll. Если во время работы диалога в это свойство записать значение, не равное mrNone, то диалоговое окно сразу закроется.
Name	Название объекта (имя переменной, которая будет использоваться в программе для обращения к этому объекту).
Options	Параметры настройки объекта. Различаются для разных компонентов.
Orientation	Ориентация объекта. Может быть горизонтальной (udHorizontal) или вертикальной (udVertical).
OwnerDraw	Характеризует, будет ли объект рисовать свой образ на экране самостоятельно (true) или использовать стандартный вид.
Picture	Картинка. Позволяет загрузить графический файл, например для компонента image.
PixelsPerInch	Число точек (пиксел) на дюйм экрана. Используется для масштабирования формы в зависимости от экранного разрешения. Будет учитываться, если свойство Scaled имеет значение true.
PopupMenu	Контекстное меню, связанное с объектом и вызываемое по щелчку правой кнопки мыши над этим объектом. Выбирается в раскрывающемся списке доступных меню и должно быть подготовлено заранее.
Position	Для формы — положение на экране. Принимает значения: poDesigned — положение окна во время работы программы соответствует положению формы на экране на этапе проектирования; poDefault — положение определяется Windows; poDefaultPosOnly — используются размеры, заданные разработчиком; poDefaultSizeOnly — используется положение, заданное разработчиком; poScreenCenter — положение в центре экрана для много мониторов систем; poDesktopCenter — положение в центре экрана. Для всевозможных компонентов прокрутки, движков и т. п. — текущее положение движка.
ReadOnly	Если данное свойство имеет значение true, то объект будет доступен в режиме только для чтения. Например, поле ввода с включенным свойством ReadOnly позволяет просматривать содержимое, но не позволяет изменять его.
Scaled	Если имеет значение true, то учитывается свойство PixelsPerInch.
Show/Hint	Определяет, надо ли показывать всплывающую подсказку, хранящуюся в свойстве Hint.
Sorted	Используется во всевозможных списках. Когда включено (true), содержимое списка будет автоматически сортироваться перед выводом на экран.
Style	Стиль объекта. Используется для разных компонентов (кнопок, списков, шрифтов и т. д.).
TabOrder	Определяет номер объекта при передвижении по элементам управления в родительском окне с помощью клавиши Tab. Начинается с 0.
TabStop	Определяет, будет ли происходить выделение данного объекта и остановка на нем при передвижении по элементам управления в родительском окне с помощью клавиши Tab.
Tag	Свойство, сделанное специально для разработчиков. Оно нигде не используется и предназначено только для хранения числа типа int. Tag можно рассматривать как пользовательское свойство и применять его для любых собственных нужд.
Text	Содержимое различных текстовых элементов управления (полей ввода и т. п.).
Title	Титул (заголовок) приложения, колонки, сообщения.
Top	Верхняя координата объекта на родителе.
Transparent	Прозрачность фона объекта. Например, если свойство Transparent имеет значение true для поля надписи, помещенного на объект-изображение, то отображаться на этом изображении будет только текст, а затирания изображения фоном не произойдет.
VertScrollBar	Вид и стиль вертикальной полосы прокрутки. Состоит из множества подсвойств.
Visible	Определяет, будет ли виден объект во время работы программы (по умолчанию — true).
Width	Ширина объекта.
WindowState	Статус окна при его открытии. Принимает значения: wsNormal — используются размеры и положение, заданные на этапе проектирования; wsMinimized — в свернутом виде; wsMaximized — в развернутом виде.

изменяет свойство **Picture** компонента **Image1** путем загрузки **LoadFromFile** графического файла **1.bmp** из каталога **PICTURE**.

В описании свойств часто встречается программный термин «родитель». Свойства компонентов, начинающиеся со слова **Parent** (родитель), определяют общение компонента со своим «родителем» — объектом, на котором он расположен. Например, когда поле ввода помещается на форму, то для него «родителем» будет форма. Если командная кнопка размещена на панели инструментов, то для нее

«родителем» будет панель и т. д. В большинстве прикладных задач эти свойства можно не учитывать.

Некоторые из перечисленных в таблице свойств имеют подсвойства. Примером такого свойства является **Font**. Ранее мы уже использовали данное свойство при создании программы обучения азбуке. Настройка подсвойств **Font** не представляет проблем, поскольку поддерживается оконным интерфейсом, знакомым пользователям Windows по приложениям **Word**, **Excel** и т. д.

Еще одним свойством с множеством подсвойств является **Anchor**

(якоря). Его подсвойства **akLeft**, **akTop**, **akRight** и **akBottom** со значениями **True** и **False** позволяют смещать (фиксировать) компонент соответственно влево, вверх, вправо или вниз.

Достаточно часто используемым свойством является **Color**. Поскольку современные графические средства позволяют передавать миллионы цветовых оттенков, число значений свойства **Color** довольно велико. Тем не менее, эти значения имеют индивидуальные названия, которые приведены в табл. 2 вместе с их описанием.

Таблица 2

Цветовая константа	Цвет	Цветовая константа	Цвет
clAqua	Цвет морской волны	clInactiveCaption	Цвет заголовка неактивного окна
clBlack	Черный	clMenu	Цвет фона меню
clBlue	Синий	clWindow	Цвет фона окон
clCream	Кремовый	clWindowFrame	Цвет рамок окон
clDkGray	Темно-серый	clMenuText	Цвет текста меню
clGray	Серый	clWindowText	Цвет текста окон
clGreen	Зеленый	clCaptionText	Цвет текста заголовка неактивного окна
clLime	Ярко-зеленый (липовый)	clActiveBorder	Цвет бордюра активного окна
clLtGray	Светло-серый	clInactiveBorder	Цвет бордюра неактивного окна
clMaroon	Бордовый	clAppWorkSpace	Цвет рабочей области компонента
clMedGray	Средне-серый	clHighlight	Цвет фона выделенного текста
clMoneyGreen	Денежный (долларовый)	clHighlightText	Цвет выделенного текста
clNavy	Цвет военно-морского флота	clBtnFace	Цвет поверхности кнопок
clOlive	Оливковый	clBtnShadow	Цвет теней кнопок
clPurple	Пурпурный	clGrayText	Цвет текста недоступных элементов
clRed	Красный	clBtnText	Цвет текста кнопок
clSilver	Серебряный	clInactiveCaptionText	Цвет текста заголовка неактивного окна
clSkyBlue	Небесно-голубой	clBtnHighlight	Цвет выделенной кнопки
clTeal	Цвет «чирок»	cl3DDkShadow	Для Windows95 и NT. Цвет темных теней трехмерных компонентов
clWhite	Белый	cl3DLight	Для Windows95 и NT. Цвет светлых частей трехмерного компонента
clYellow	Желтый	clInfoText	Для Windows95 и NT. Цвет текста всплывающих подсказок
clScrollBar	Цвет полос прокрутки	clInfoBk	Для Windows95 и NT. Цвет фона всплывающих подсказок
clBackground	Цвет фона рабочего стола	clGradientActiveCaption	Для Windows98 и Windows2000. Цвет правой стороны градиентной заливки названия активного окна.
clActiveCaption	Цвет заголовка активного окна	clGradientInactiveCaption	Для Windows98 и Windows2000. Цвет правой стороны градиентной заливки названия неактивного окна.

Для использования этих значений из программы необходимо присваивать их название свойству Color компонента. Например, следующая строка программы:

```
Label1->Color = clGreen;
```

изменяет свойство **Color** компонента **Label1** на **clGreen**, т. е. на зеленый цвет.

Продолжение следует

Олег Вальпа,
г. Миасс Челябинской обл.

velleman

www.platan.ru
ПЛАТАН

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ ОТ ВЕДУЩИХ ПРОИЗВОДИТЕЛЕЙ

Миниатюрный осциллограф HPS-40

Москва, ул.Ивана Франко, д. 40, стр. 2 • Тел./факс: (095) 73 75 999 • Почта: 121351, Москва, а/я 100 • E-mail: platan@aha.ru

Компьютер — измерительный комплекс

Сегодня практически в каждом доме или офисе есть компьютер, иногда даже несколько. С другой стороны, мало у кого есть генератор, вольтметр, осциллограф, анализатор спектра, измеритель нелинейных искажений. Дело в том, что эти приборы, как правило, дороги, занимают много места, а требуются далеко не каждый день. Нередко компьютер может с успехом заменить все эти приборы. В данной статье предпринята попытка показать некоторые возможности компьютера как мощного измерительного инструмента. При этом в конструкцию компьютера не потребуются вносить никаких изменений, дополнительно потребуются только шнуры для подключения, звуковая карта и, конечно, программы.

Можно считать, что в любом компьютере есть звуковая карта. Практически невозможно купить без нее новый компьютер. Компьютер без звука, как глухонемой человек — его возможности ограничены. Существуют сотни, а может быть и тысячи вариантов звуковых карт, встроенных и в виде отдельных плат. Все варианты и особенности учесть невозможно, поэтому мы будем ориентироваться на бюджетный (экономный) вариант — звуковую карту Sound Blaster Live! фирмы Creative Labs. Звуковая карта собрана на небольшой печатной плате с интерфейсом PCI. На рис. 1 показан современный вариант Sound Blaster Live!, фото взято с сайта <http://www.sblive.com>.

Сердце карты — специализированный процессор EMU10K1 с вычислительной мощностью, сопоставимой с процессором Intel Pentium 100. Все сигналы обрабатываются с точностью 32 бит 48 кГц с использованием запатентованной восьмиточечной интерполяции для уменьшения искажений. Де-факто она уже давно стала стандартом, когда говорят о компьютерном звуке, так и говорят — Sound Blaster. При невысокой цене (около \$30) она обладает весьма неплохими свойствами. На странице <http://www>.

ixbt.com/multimedia/sblivehomest.html приведены основные параметры, заявленные производителем и реально измеренные. Конечно, вторые несколько хуже, но все равно впечатляют:

- уровень выходного сигнала на аналоговом выходе, В 1;
- соотношение сигнал/шум, дБ 90;
- КНИ плюс шумы, % 0,007;
- частотный диапазон при уровне сигнала -20 дБ, кГц 0,02...20
- неравномерность частотной характеристики, дБ -0,05...+1.

При измерениях сигнал проходил через весь тракт передачи и приема звуковой карты, собственно сигнал с выхода звуковой карты еще лучше. Стабильность частоты определяется кварцевым резонатором на уровне $10^{-5} \dots 10^{-6}$.

Параметры семейства современных звуковых карт Audigy той же фирмы Creative Labs значительно выше, карты других производителей могут быть несколько лучше или хуже, но их параметры обычно того же порядка. Например, по данным сайта www.ixbt.com для звуковой платы Turtle Beach Montego гармонические искажения при уровне -3 дБ на частоте

1 кГц составляют 0,004 %, соотношение сигнал/шум 75 дБ; для звуковой платы Event GINA — 0,0022 % и 80 дБ, соответственно. Если не предъявлять высоких требований к выходному сигналу, годятся почти любые звуковые

карты. Главное, чтобы они поддерживали режим Full duplex (возможность одновременной работы на прием и передачу).

Так много внимания уделено именно звуковой карте потому, что от нее будут зависеть основные параметры наших измерительных приборов. Требования, которые предъявляет к компьютеру сама звуковая плата, весьма скромные — по заявлению фирмы-изготовителя достаточно Intel Pentium 133 и желательно 32 Мб оперативной памяти. Требования программ к компьютеру тоже небольшие, при необходимости они будут указаны. Конечно, большие объемы памяти и частота процессора будут полезны — работать становится быстрее и приятнее. Теперь перед бочкой с медом полагается ложка дегтя. Верхняя рабочая частота всех предлагаемых решений без внешних дополнительных устройств, как правило, не более 20...22 кГц. Есть исключения, когда этот диапазон увеличивается в несколько раз, но это неприципиально, поэтому рассматривать такие случаи не будем. Напрашиваются внешние преобразователи АЦП и ЦАП, что сразу снимает многие ограничения, но мы будем рассматривать только софт. Нельзя не отметить, что существуют профессиональные промышленные платы расширения для компьютера, которые предназначены именно для измерений. Они сочетают в себе высокую точность, широкий диапазон измеряемых величин, большой (по сравнению с осциллографами — просто огромный цветной дисплей компьютера 17...21" и более), возможность записи результатов измерений в файл и т. д. Есть один минус — стоимость, хотя и она нередко составляет всего несколько сотен у. е. Закончим с диапазоном частот, дальше начинается мед — частоты генерации, измерения и отображения могут быть очень низкими, всего несколько Гц, вплоть до постоянного тока.

Файлы с программами, описанными в данной статье, доступны на сайте журнала по адресу www.dian.ru/programs/ или по ссылкам на авторские сайты (некоторые программы являются платными и редакция не имеет право их распространять; в ряде случаев авторы требуют, чтобы их программы переписывались только с авторского сайта, в этом случае редакция вынуждена соблюдать авторские требования).

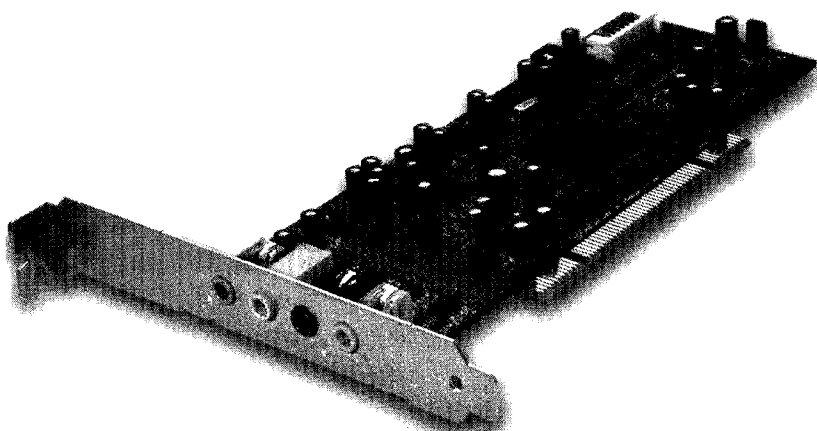


Рис. 1

Сначала рассмотрим приборы, в которых чаще бывает потребность и реализовать которые наиболее просто.

ГЕНЕРАТОРЫ

Любая звуковая плата имеет минимум два выхода (их может быть четыре, пять, даже восемь по мультимедийному стандарту для домашнего кинотеатра 7.1 Dolby Digital). Поэтому большинство программ выдает, как правило, сигнал звуковой частоты по двум каналом независимо друг от друга. Если они синхронны и противофазны, получаем симметричный относительно общего провода сигнал. Регулировка выходного сопротивления отсутствует, оно мало, т. к. допускается подключение головных телефонов. Входные и выходные разъемы — стандартные 3,5 мм (рис. 1).

Представлять программное обеспечение постараемся по принципу «от простого к сложному». Соответственно и возможности разные, сложность освоения тоже.

NCH Tone Generator

Это самая простая из найденных программ. После запуска файла `tnsetup.exe` на диске C: образуется каталог NCHTONE с файлами `nchtoner.exe` (сама программа), `nchtone.hlp` (файл помощи на английском языке), `uninst.exe` (файл для корректного удаления самой программы). Поскольку не для всех английский язык хорошо понятен, будут приведены краткие описания программ и порядок пользования ими.

Возможности достаточно скромные — диапазон генерируемых частот 1 Гц...20 кГц с шагом 1 Гц, форма сигнала — синусоидальная, прямоугольная, треугольная, пилообразная, импульсная, белый шум. Внешний вид окна программы показан на рис. 2.

Программа работает практически под любой версией операционной системой Windows, начиная от 3.1 и до XP. Предусмотрены дополнительные

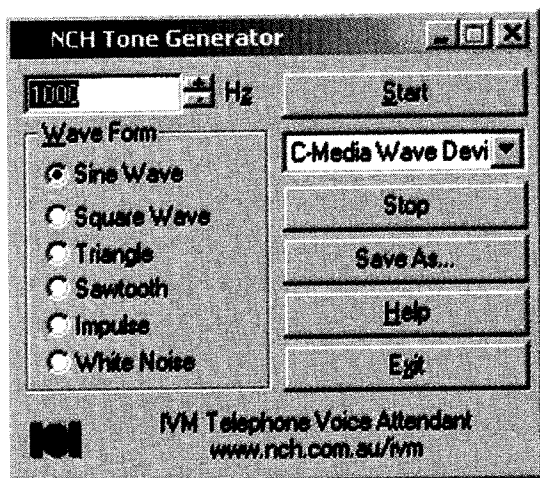


Рис. 2

удобства для музыкантов. Можно установить частоту 440 Гц, каждое нажатие клавиши + и - на дополнительной клавиатуре при этом добавляет или вычитает полутон, комбинации клавиш Ctrl+ и Ctrl- изменяют частоту точно на октаву. При нажатии кнопки Start на выходе звуковой карты появляется звуковой сигнал соответствующей формы и частоты (в режиме генерации белого шума установленная частота не учитывается). При нажатии кнопки Stop сигнал прекращается. Если полученный сигнал при прослушивании искажен, имеет место шум и щелчки, это указывает на недостаточную мощность компьютера (касается и других программ). Амплитуда сигнала в данной программе не регулируется, приходится пользоваться штатным регулятором выходного сигнала операционной системы или звуковой карты. Естественно, все эффекты звуковой карты (эхо, реверберация, хорус, фленджер и пр.) и эквалайзер должны быть выключены. Это касается и всех остальных программ, если не оговорено особо.

При нажатии кнопки Save As... открывается диалоговое окно (рис. 3) в котором можно указать необходимую длительность записываемого сигнала и имя файла. Файл будет сохранен в каталог NCHTONE, изменить расширение файла (`wav`) и каталог сохранения нельзя (да и не надо).

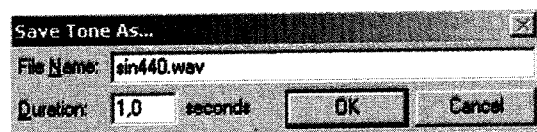


Рис. 3

Итак, даже эта простая и бесплатная для некоммерческого пользования программа обладает очень неплохими возможностями. Попробуйте установить, например, частоту ровно 440 Гц для проверки музыкального камертона с помощью обычного генератора, не говоря уже о стабильности частоты. Кстати, создать обычный генератор с частотой несколько герц и малыми нелинейными искажениями тоже совсем не просто. О возможности сохранения сигнала в файл следует упомянуть особо. Это открывает широкие возможности для создания специальных сигналов. Далее после обзора ряда программ мы рассмотрим, как изготовить свой уникальный измерительный компакт-диск.

Generator

Еще одна простая программа. Работает без инсталляции и создает всего

один исполняемый файл объемом 28 Кб. Формирует только синусоидальный сигнал. Внешний вид окна программы показан на рис. 4.

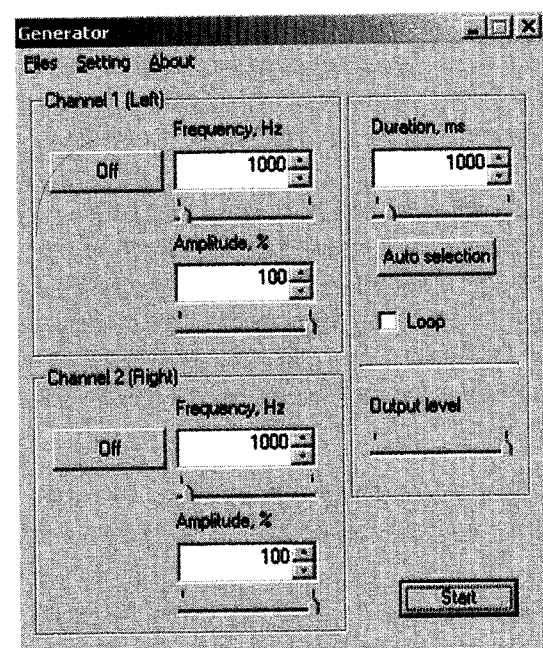


Рис. 4

Программа предназначена для работы под управлением операционной системы Windows95/98, 2000, XP. Требования к аппаратной части компьютера минимальные — процессор 80386 с сопроцессором (если вы умудритесь его найти) или любой процессор более поздней разработки. Есть возможность отдельной и независимой регулировки частоты и амплитуды выходного сигнала в двух каналах. Для запуска генератора надо нажать кнопку Start, для остановки повторно нажать эту же кнопку (на ней появится надпись Stop). Частота регулируется в диапазоне 1...22050 Гц с шагом 1 Гц, амплитуда 1...100 % с шагом 1 %. Нажатие кнопки Off отключает левый или правый канал. Длительность выходного сигнала не более 10 с (без функции Loop). При установке галочки в окне Loop (петля) сигнал формируется непрерывно, т. е. после окончания посылки она будет повторяться снова и снова. Так как заданная длительность сигнала, как правило, не кратна целому числу периодов сигнала, при повторении частотной посылки произойдет разрыв фазы. Для его устранения предназначена кнопка Auto selection. При ее нажатии происходит автоматическая подстройка длительности сигнала до величины, при которой выходной сигнал будет иметь целое число периодов, скачка фазы при использовании режима Loop не будет. Регулятор Output level позволяет регулировать уровень выходного сигнала от нуля до максимума, но уже только «на

глазок». Программа позволяет выбрать одно из имеющихся в системе устройств вывода звука. Для этого предназначен пункт меню Setting — Devices. Программа анализирует, какие устройства установлены в компьютере и выводит наименования обнаруженных устройств в список. Пользователю следует указать одно из устройств из списка. Для установки режима работы предназначен пункт меню Setting — Modes. Вы можете выбрать один из появившихся стандартных режимов, если он поддерживается выбранным устройством вывода звука. Рекомендуется режим — частота дискретизации 44100 Гц, 16 бит на отсчет, стерео. Если этого режима нет в списке, значит звуковая карта безнадежно устарела.

К сожалению, нет возможности записать звуковой сигнал в файл, функция File — Save сохраняет лишь настройки программы, но не сам сигнал. Впрочем, при желании эта проблема легко решается — существует много программ, которые записывают на жесткий диск сигнал с выхода звуковой карты.

Возможность одновременной генерации двух сигналов с разными частотами и амплитудами открывает новые возможности. Например, с помощью микшера можно получить сложный сигнал для измерения интермодуляционных искажений. Существует несколько разных методик измерения этих искажений. Обычно сигнал состоит из двух частот — низкой в диапазоне 50...150 Гц и высокой — в диапазоне 10...16 кГц. Еще один способ применения двухчастотного генератора — настройка тока подмагничивания кассетного или катушечного магнитофона. Сигнал опорной частоты записывают с уровнем -20 дБ от максимального уровня записи магнитофона и частотой, как правило, 315 или 400 Гц, и сигнал, близкий к максимальной частоте воспроизведения, обычно в диапазоне 14...18 кГц. При оптимальном токе подмагничивания уровни воспроизведения этих двух сигналов должны быть равны. В зависимости от способа контроля сигнала воспроизведения применяются разные способы записи. Если есть частотоизбирательный измерительный прибор, который способен отдельно измерить в одном канале уровень низкой и высокой частоты, входной сигнал при записи микшируется и подается в один канал записи. При отсутствии такого прибора до-

пустимо записать, например, в правый канал сигнал низкой частоты, в левый — высокой. При воспроизведении любым вольтметром добиваются равенства уровней сигналов низкой и высокой частоты, при этом АЧХ канала записи-воспроизведения магнитофона будет максимально близка к линейной.

Для смешивания сигнала из двух каналов можно использовать простые микшеры, как пассивные, так и активные. Схема простейшего микшера приведена на рис. 5.

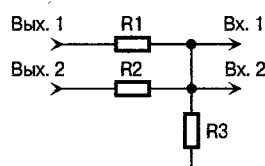


Рис. 5

Номиналы резисторов не критичны, $R1 = R2 \gg R3$ (обычно $R1, R2$ — десятки кОм, $R3$ — несколько кОм), зависят от входного сопротивления устройства, на которое подается сигнал. Пассивный микшер не требует электропитания, настройки, может быть размещен прямо на соединительном кабеле или в разъеме. При необходимости может быть использован активный микшер, который обеспечивает лучшее разделение каналов и необходимое усиление, но требует электропитания, лучше автономного, для минимизации помех и наводок.

Frequency master generator

Последняя рассматриваемая в данной статье простая программа-генератор. Ее особенность — отсутствие графической оболочки, т. е. для получения звуковых файлов формата wav необходимо вручную набрать в командной строке команду, например, Gen44k.exe FREQ=1000 SEC=1.0 out.wav. В результате в каталоге с программой будет создан файл out.wav с сигналом длительностью 1 с и частотой 1000 Гц. Это самый простой случай, однако программа позволяет при помощи специально написанных текстовых файлов создавать достаточно сложные и изощренные сигналы.

Приведем в качестве примеров ряд таких текстовых файлов. Чтобы с их помощью получить сигнал, необходимо в файловом менеджере, например, в Total Commander, выполнить несколько простых действий: выделить файл Gen44k.exe или Gen48k.exe, щелкнув по нему, нажать Ctrl+Enter, имя файла окажется в командной строке.

Аналогично перенести в командную строку текстовый файл с программой, дописать через пробел в командной строке имя выходного файла, например, out1.wav. При этом все файлы должны находиться в одном каталоге и соответствовать требованиям DOS, поэтому, если использовать авторские файлы с примерами, правильной будет команда Gen44k.exe stereo2.txt out33.wav, а команда Gen44k.exe Frequency changes.txt frequency.wav работать не будет.

Написать текстовый файл не так сложно, для примера приведем содержание простого и короткого авторского файла Nice_sounds.txt

```
// Nice sounds :)
##
freq=D-4 sec=1 vol=48 vol_to=20 pos= 10##
freq=E-4 sec=0.9 vol=48 vol_to=20
pos=32##
freq=F-4 sec=0.8 vol=48 vol_to=20
pos=48##
freq=G-4 sec=.7 vol=50 vol_to=20 pos=64##
freq=A-4 sec=.6 vol=50 vol_to=20 pos=80##
freq=B-4 sec=.5 vol=50 vol_to=20 pos=96##
freq=C-5 sec=.4 vol=50 vol_to=20
pos=120##
<END>
«Note using» demo for Frequency Master
Generator
(c) 2000 by Andrew Tkachyov.
```

Описание и файл помощи, несмотря на фамилию автора, на английском языке, поэтому вкратце опишем основные возможности.

Основные параметры:

- sine wave — синусоидальный сигнал;
- square — прямоугольный сигнал;
- saw — пилообразный сигнал;
- triangle — треугольный сигнал;
- drum — имитация ударных инструментов.

Дополнительные параметры:

- duration — длительность сигнала;
- volume — уровень сигнала;
- panning — азимут источника звука.

Специальные эффекты:

- volume smoothly changes — плавное изменение громкости;
- panning move — плавное изменение позиции источника звука;
- smoothly changes tone — плавное изменение частоты;
- vibration — девиация частоты;
- panning vibration — качание позиции источника звука;

- semicolon sign — ноты.
- Возможны паузы и задержки сигнала.
- Ограничения:
- максимальная частота 22050 Гц для частоты дискретизации 44,1 кГц и 24000 Гц для 48 кГц;
 - длительность сигнала до 10 с;
 - число сигналов в одном файле ≤ 40.
- Из перечисления этих возможностей видно, что внешняя простота программы обманчива и при умении

работать с текстом и командной строкой можно добиться неплохих результатов. Для большего удобства можно пользоваться командными файлами (расширение bat). Это файл, который удобно создавать и редактировать в простом редакторе типа Notepad, он может содержать одну строку, например, Gen44k.exe stereo2.txt out33.wav. Это избавит от необходимости работать с командной строкой. Содержимое этого файла элементарно редактируется, а запускать его, осо-

бенно при отладке, очень удобно. Для Windows 2000, XP и более новых вариантов при запуске командной строки или командного файла для того, чтобы успеть увидеть на экране ход выполнения задачи (особенно при каких-либо проблемах), необходимо держать нажатой клавишу Shift.

Продолжение следует

Сергей Белов,
г. Москва

Автомобильный мультиметр-часы на микроконтроллере PIC16F873

Микроконтроллеры PIC16F873 завоевали широкую популярность. На их базе можно изготовить множество простых, но многофункциональных изделий. Одно из них мы и предлагаем вашему вниманию.

Автомобильный мультиметр-часы позволяет водителю получать информацию о текущем времени, напряжении аккумулятора, температуре внутри салона и снаружи, частоте вращения коленчатого вала. Принципиальная схема устройства показана на рис. 1. Устройство обладает достаточно высокими эксплуатационными ха-

рактеристиками. Погрешность хода часов — не более 1,3 с в сутки, время хода при отключении от бортовой сети — 40...50 час. Погрешность измерения напряжения аккумулятора — не более 0,1 В, максимальное напряжение на входе вольтметра 25,5 В, температура может быть измерена в диапазоне -20...+100 °С с точностью не хуже 1 °С, рабочий диапазон тем-

пературы -40...+100 °С. На рис. 2 и 3 показан вид индикации при отображении положительной и отрицательной температуры.

При выходе контролируемой температуры за пределы указанного диапазона на индикаторе высвечиваются надписи «cold» (холод) или «hot» (жара) (рис. 4 и 5).

Погрешность измерения частоты вращения коленвала двигателя — 10 об/мин при установившемся режиме двигателя (усреднение за последние три секунды) и 30 об/мин при изменяющейся скорости вращения оборотах (усреднение за последнюю секунду).

Потребляемый ток в режиме измерения менее 100 мА, в режиме отключенной индикации менее 9 мА, при отключении от бортовой сети автомобиля от собственного источ-

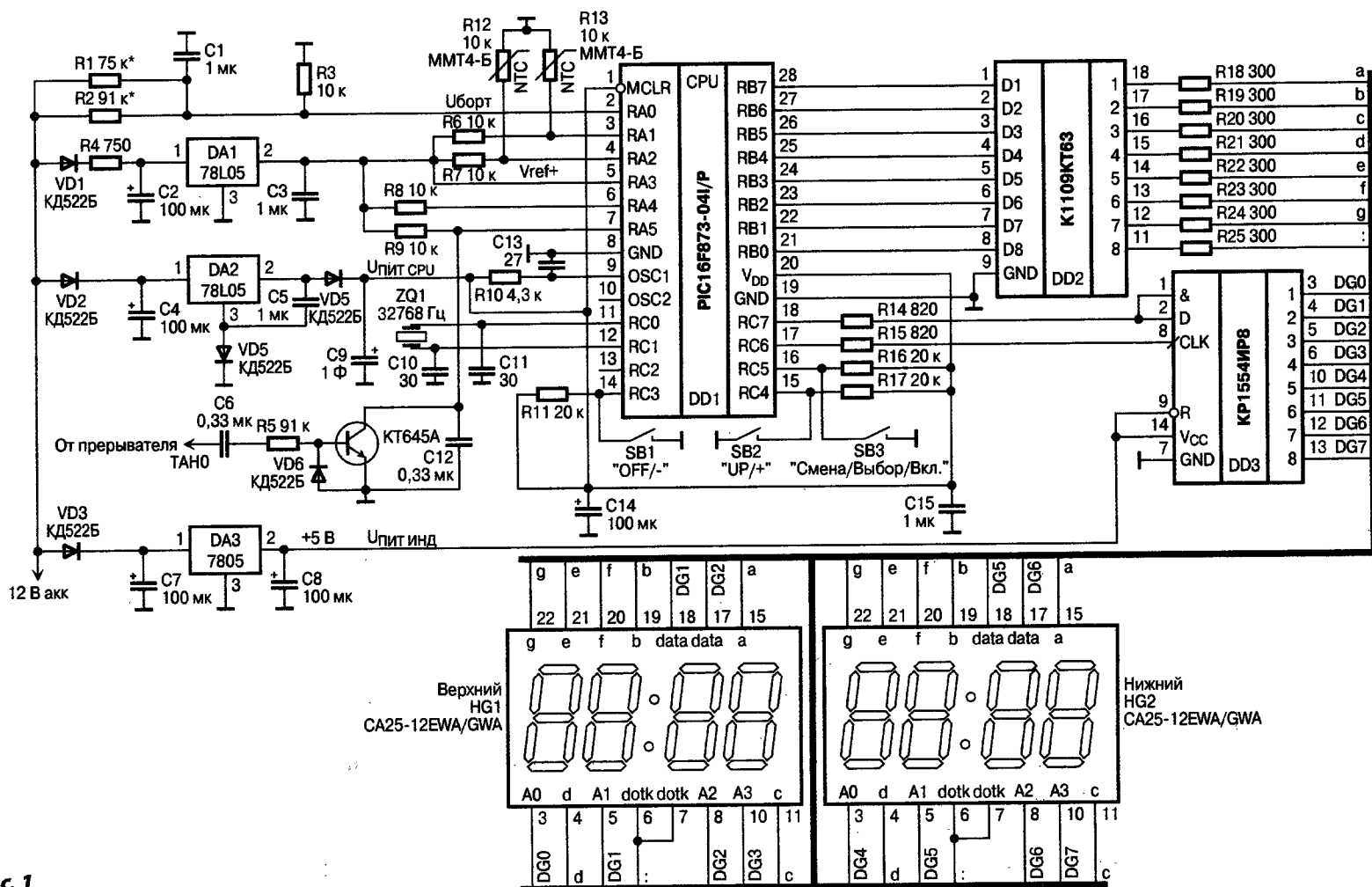


Рис. 1



Рис. 2



Рис. 3



Рис. 4

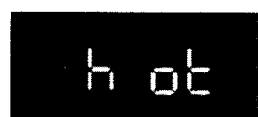


Рис. 5

ника питания (ионистора емкостью в 1 Ф) — около 30 мкА. При отсутствии ионистора устройство можно упростить, исключив нижний индикатор и кнопку SB1. Все измерительные функции прибора останутся, но на одном индикаторе, часы перестанут работать при отключенной массе автомобиля.

Индикация производится двумя светодиодными семисегментными индикаторами типа CA25-12. Каждый индикатор имеет четыре знакоместа и двоеточие (есть еще несколько неиспользуемых точек).

Управление прибором осуществляется тремя кнопками. Кнопка SB2 «UP/+» переключает на верхнем индикаторе индицируемый параметр по циклу Часы — Вольтметр — Термометр 1 — Термометр 2 — Тахометр — Часы.

Кнопка SB2 «Смена — Выбор — Вкл» может менять местами отображаемую информацию между нижним и верхним индикаторами, т. е. можно установить на любой индикатор любой параметр. При этом если на верхнем и нижнем индикаторах установлен одинаковый параметр, на нижнем произойдет фиксация последнего значения, если же на индикаторах разные параметры, индикация на них будет обновляться каждую секунду.

Кнопкой SB1 «OFF/—» при первом нажатии можно погасить верхний индикатор, при втором — оба. Включить индикацию можно кнопкой «Смена — Выбор — Вкл».

Для входа в режим установки часов необходимо кнопку «Смена — Выбор — Вкл» удерживать нажатой не менее трех секунд.

Можно изменить способ подключения устройства к бортовой сети. Для этого надо разъединить диоды устройства, подключенные к аккумуляторной батарее. Диод, питающий МК, остается подключен непосредственно к аккумуляторной батарее, остальные входы надо присоединить к замку зажигания так, чтобы питание на диоды поступало только при повороте ключа зажигания. В этом случае при

выключенном зажигании мультиметр переходит в режим низкого потребления тока (3 мА). Термодатчики на основе терморезисторов с отрицательным ТКС можно подключить к прибору с помощью коаксиального кабеля длиной до 1,5 м, оплетка кабеля должна иметь хороший электрический контакт с корпусом терморезистора. Общий провод от датчиков термометра не должен касаться металлических частей автомобиля.

Программы для микроконтроллера находятся на сайте журнала по адресу <http://www.dian.ru/programms/>.

Программа на ASM для PIC16F873 содержит подпрограмму для чтения таблицы соответствия между полученным кодом АЦП и измеренной температурой. Однако, так как все терморезисторы даже одного номинала имеют несколько различающиеся параметры, эту таблицу нельзя рассчитать на все случаи жизни. Для устранения этой проблемы написана программа под DOS, которая формирует таблицу по двум точкам зависимости температура—код АЦП.

Рекомендации по работе с программой:

- запустить программу `termometr.exe`, указать режим «k» — калибровка. Адрес начала таблицы — 800;
- текст полученного файла дописать к готовому HEX файлу мультиметра (без последней строки «00000001FF» — это конец файла) и сохранить под другим именем, например, `klbr.hex`;
- запустить программу `termometr.exe` в режиме «k», но уже с адресом C00 (адрес в шестнадцатеричной системе);
- полученный текст также добавляем в конец `klbr.hex` без последней строки файла `klbr.hex`. Новый файл `klbr.hex` должен содержать строку «00000001FF» один раз в конце файла;
- в готовое устройство установить PIC16F873 с прошивкой `klbr.hex`;
- после включения питания следует вывести на индикаторы показания обоих термометров. Они будут показывать какие-то цифры и символы градуса и градуса с подчеркиванием. Индикатор с подчеркиванием соответствует таблице по адресу 800, без подчеркивания — C00. Для калибровки туда занесены просто числа по порядку в десятичной системе;
- погрузить оба терморезистора в среду с 0 °С (вода со снегом),

подождать пока прекратятся изменения показаний индикаторов. Записать значения этих кодов АЦП при нулевой температуре;

- погрузить оба термометра в среду с другой (40...70 °С) температурой (главное, чтобы она была постоянной в течение замеров). Точным (электронным или жидкостным) термометром измерять эту температуру, фиксируем состояние АЦП для этой температуры;

- из исходного файла `mult.hex` аналогично формируем новый файл с помощью программы `termometr.exe`, но уже в режиме «t»;
- программа запросит значение температуры по Цельсию и коды АЦП для этих температурных точек, сформирует рабочую градуировочную таблицу термометра;
- полученный новый файл, состоящий из файлов `mult.hex+tabl.hex` для адреса 800 и `tabl.hex` для адреса C00 тоже должен содержать единственную строку «00000001FF» в конце файла.

Этот файл есть то, ради чего можно не покупать температурные датчики DS. Он заносится в микроконтроллер уже с готовыми градуировочными таблицами.

Описанная процедура кажется сложной, но занимает не более получаса.

Для калибровки устройства в режиме вольтметра следует подать на вывод питания «+12 В акк» напряжение 12 В, контролируемое цифровым вольтметром, и добиться аналогичных показаний подборкой резисторов R1 и R2.

Устройство успешно опробовано на двух автомобилях ВАЗ-2106, оно установлено вместо штатных часов. Температура в месте установки при работающей печке достигает 70 °С. В других автомобилях часы находятся в более щадящем температурном режиме.

Игорь Новоселов,
г. Екатеринбург

Декодер телевизионного стереозвукового сопровождения формата NICAM NM2905

Статья посвящена новой разработке МАСТЕР КИТ — декодеру телевизионного стереозвукового сопровождения формата NICAM NM2905. Декодер представляет собой готовое и настроенное устройство. Приводятся описание, технические характеристики, режимы работы, структурная и электрическая схемы декодера, схема подключения к телевизору. Этот материал должен заинтересовать работников ремонтного бизнеса, частных телемастеров и подготовленных радиолюбителей.

НЕМНОГО ИСТОРИИ

История попыток внедрения стереофонического звукового сопровождения телевидения в России, пожалуй, не менее длительна, чем само стереофоническое радиовещание. Специалисты знают, что отечественная система радиостереовещания с полярной модуляцией изначально разрабатывалась адаптированной к телевидению, поэтому в ней использована поднесущая частота 31,25 кГц, в точности равная удвоенной строчной частоте. Первые экспериментальные телевизионные передачи со стереозвуком были проведены в СССР еще в 1974 г. Однако в дальнейшем эта система так и не была внедрена. Позднее предлагалось еще несколько вариантов систем стереофонического звукового сопровождения телевидения, но ни один из них не был доведен до внедрения. За прошедшие несколько десятилетий отечественная наука и промышленность так и не смогли создать жизнеспособного стандарта стереофонического звукового сопровождения телевидения. Остальной мир в это время не стоял на месте, в Европе внедрились стандарт телевизионного стереовещания А2. Идея этого стандарта очень проста — помимо основной ЧМ поднесущей звука (5,5 МГц) в телевизионный радиосигнал вводится еще одна немного смещенная поднесущая (5,74 МГц), на которой передается стереоинформация. Для декодирования такого стереосигнала в телеприемник устанавливается дополнительный аналоговый ЧМ демодулятор звука, аналогичный основному, и стереоусилитель с двумя динамическими головками. Такая система стереофонического звукового сопровождения проста и незначительно увеличивает стоимость

телевизора, поэтому она получила широкое распространение. Однако система А2 сильно подвержена воздействию помех и не обеспечивает высокого качества стереосигнала звукового сопровождения.

Для устранения недостатков аналоговых систем стереовещания в Англии была разработана и внедрена принципиально новая цифровая система стереофонического звукового сопровождения телевидения NICAM-728. Были разработаны ее модификации, адаптированные под английский стандарт I и европейский стандарт В/G. В европейском варианте звук оцифровывается и передается в виде цифрового потока на поднесущей частоте 5,85 МГц. Система NICAM обеспечивает очень высокое качество стереофонического вещания, близкое к качеству CD, но декодер для нее сложнее и значительно дороже, чем декодер А2. Первоначально это оказалось тормозом для широкого внедрения этой системы. Установка декодера NICAM увеличивала стоимость телевизора на \$40...50. Однако сегодня ситуация изменилась. В современных телевизорах для декодирования звука используются специализированные цифровые сигнальные процессоры. В них поднесущая звука на входе процессора переводится в цифровую форму и вся дальнейшая обработка — преобразование, фильтрация, декодирование и др. — ведется программными методами. В таком процессоре реализация декодирования любого стандарта стереозвука обеспечивается простым переключением программных модулей, что практически не влияет стоимость телевизора. По этой причине использовать сегодня для вещания устаревшие аналоговые методы стереовещания, не



обеспечивающие высокого качества звука, нерационально.

Положение телеведущих сегодня иное чем во времена «развитого социализма». Телевизионные каналы, остро нуждаясь в дополнительных инструментах для привлечения телезрителей и устав ждать от нашей науки решения проблемы стереофонического телевидения, начали «явочным порядком» вводить стереовещание. Так, в конце 2003 г. «Первый канал» начал в Москве стереовещание в формате NICAM-728. Немного позднее на этот же формат перешел канал MTV. С начала 2005 г. ожидается начало стереовещания «Первого канала» в Санкт-Петербурге. В этом же стандарте вещает канал М1 в Киеве и ряд каналов в различных кабельных сетях по всей стране.

Так что интерес к стереофоническому звуку в телевидении в России есть и весьма значительный. Есть он и со стороны телезрителей. Об этом свидетельствуют многочисленные звонки в диспетчерские службы с вопросами о возможности стереоприема после начала эфирного стереовещания.

Опыт «Первого канала» показал, что дооснащение телевизионного передатчика для ввода цифрового стереовещания не является очень затратной задачей. Это обусловлено тем, что мощность дополнительного передатчика стереозвука NICAM должна относиться к мощности основного телевизионного передатчика в соотношении 1/500, т. е. для ТВ передатчика мощностью 10 кВт достаточно установить передатчик NICAM радиосигнала мощностью всего в 20 Вт. Дополнительно нужно на входе основного передатчика ТВ ограничить спектр модулирующего сигнала полосой 5,1 МГц, что представляется не очень сложной задачей.

Справедливости ради нужно признать, что решение «Первого канала» о начале стереовещания в формате D/K NICAM было революционным, но его нельзя считать авантюрным.

Дело в том, что ранее этот формат стереовещания был внедрен в Китае, Венгрии и Польше, где используется такой же стандарт телевидения, что и в нашей стране. Опыт этих стран доказал возможность использования сигнала NICAM в телевизионном сигнале стандарта D/K. Правда, для снижения уровня помех от несущей сигнала NICAM потребовалось уменьшить ее уровень в составе спектра полного радиосигнала до значения -25 дБ по отношению к уровню несущей изображения (в стандарте V/G -20 дБ), что несколько усложняет декодирование стереосигнала и уменьшает зону уверенного приема стереовещания. По этой причине после того, как «Первый канал» начал вещание со стереозвуковым сопровождением, Государственный НИИ Радио в начале 2004 г. провел экспериментальные исследования по определению зоны уверенного приема стереозвукового сопровождения [1]. По относительно большому объему экспериментального материала было определено: «усредненное (единое для всех сезонов и времени суток) положение внешней границы зоны уверенного приема сигнала стереозвука от передатчика «Первого канала» [круг] радиусом 86 км». Такое значение практически соответствует границе уверенного приема телевизионного сигнала, т. е. стереоприем обеспечивается практически во всей зоне покрытия телевизионного передатчика.

Более существенным фактором является то, что телевизоров, способных декодировать стереозвук NICAM, в нашей стране всего 2...5 % от общего числа, и все они импортные. Поэтому

говорить о готовности парка телевизоров России к приему стереофонического вещания в системе NICAM нельзя.

Сейчас положение быстро меняется, ввозится и продается все больше телевизоров и видеомагнитофонов со встроенными декодерами NICAM. Отечественные производители также начали выпускать телевизоры с такими декодерами, однако полное обновление приемного парка телевизионных приемников займет 10...15 и более лет. В результате существует потребность в технических решениях, которые позволят принимать на существующий парк телевизоров стереосопровождение формата NICAM.

Похожая задача стояла перед населением нашей страны в 80-х годах уже прошлого столетия, когда началось широкое внедрение бытовых видеомагнитофонов. Тогда практически все записи для них были в цветном стандарте PAL, а существовавший парк телевизоров мог принимать только сигналы в стандарте SECAM. Потребовалось разработать способ модернизации телевизоров для того, чтобы они могли декодировать сигналы SECAM/PAL. Очень быстро без участия государства энтузиастами было разработано множество вариантов плат PAL-декодеров и способов их установки в телевизоры. Появилось множество фирм, которые занимались изготовлением и установкой таких декодеров, и проблема модернизации приемного парка была решена в короткие сроки.

Аналогичным способом можно решить и проблему приема существующим парком телеприемников стереозвукового сопровождения телевидения

формата NICAM. По этому пути уже прошла Польша и Голландия. Там есть фирмы, выпускающие платы декодеров NICAM, которые потребитель может установить в свой телеприемник и получить возможность слушать стереозвуковое сопровождение. Россия, видимо, также пойдет по этому пути. Первый известный автору российский серийный декодер стереотелевизионного звукового сопровождения NICAM, предназначенный для установки в телевизоры, выпущен фирмой «Мастер КИТ». Блок пока выпускается в одной модификации для телевизоров с промежуточной частотой 38,9 МГц (большинство телевизоров, имеющих у населения) и имеет название NM2905. При его разработке пришлось решить ряд проблем, связанных со специфическими особенностями приема NICAM в нашей стране. Далее будет подробно описана конструкция декодера, его возможности и особенности.

ДЕКОДЕР NICAM СТЕРЕОСИГНАЛА NM2905

Внешний вид декодера показан на рис. 1. Он выполнен на базе специализированного цифрового сигнального процессора MSP3410D производства фирмы Micropas, по технологии поверхностного монтажа и имеет размеры 85×65×20 мм. При установке не требуется никаких дополнительных настроек. Небольшие размеры декодера позволяют установить его даже в малогабаритные телевизоры с размером экрана 14" и менее.

Использованная в блоке специализированная микросхема цифрового сигнального процессора позволяет получить хороший звуковой сигнал. В режиме приема NICAM устройство имеет следующие основные параметры:

Отношение сигнал/шум на выходе, не менее	72 дБ
Коэффициент гармоник, не более	0,1 %
Разделение каналов, не менее	80 дБ
Неравномерность АЧХ в полосе 20...15 000 Гц, не более	1 дБ

Микросхема также позволяет осуществлять декодирование звукового сопровождения телевидения в любом из существующих форматов телевидения — A2, D/K, V/G, SAT и др.

Упрощенная структурная схема декодера показана на рис. 2.

При создании этого декодера разработчикам пришлось решить несколько специфических проблем,

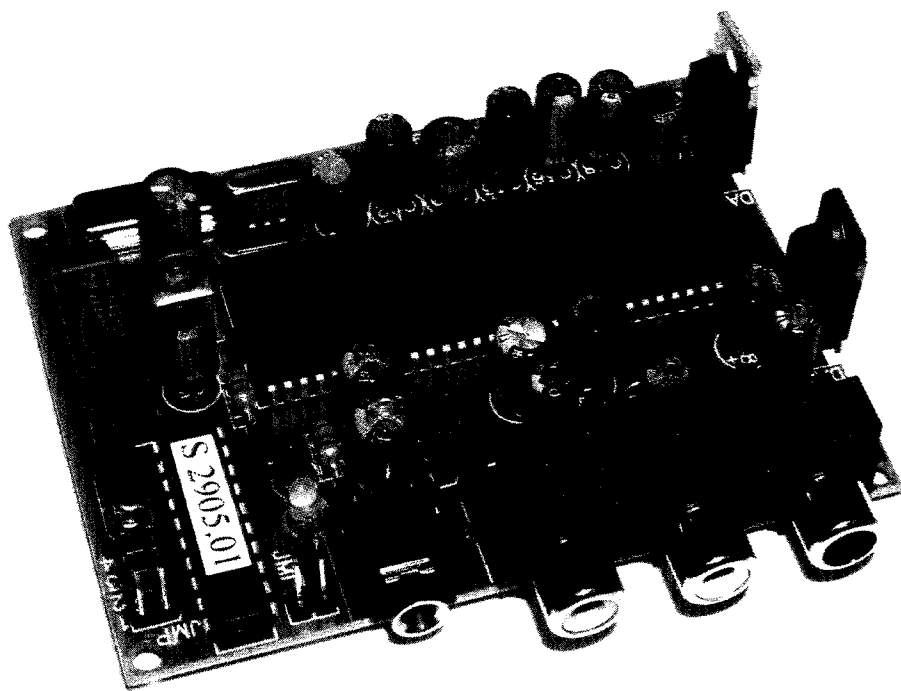


Рис. 1

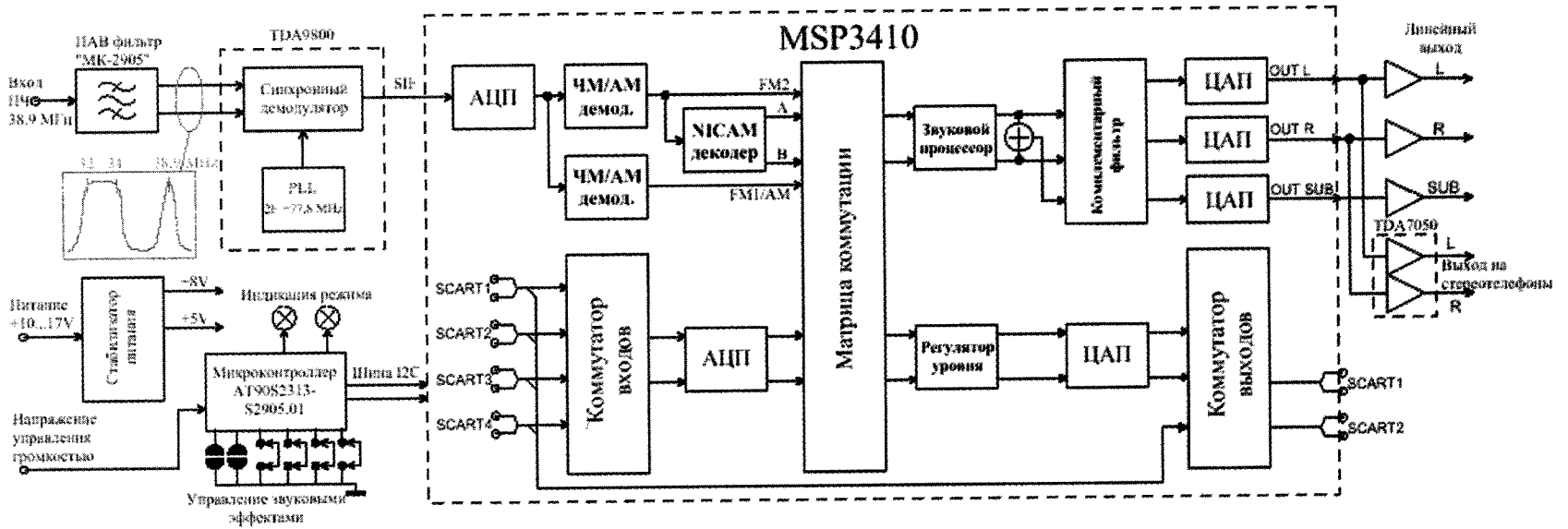


Рис. 2

связанных с особенностями реализации стереовещания NICAM в составе сигнала SECAM D/K. Как уже упоминалось выше, для снижения помех от поднесущей сигнала NICAM ее уровень был существенно снижен по сравнению с первоначальной спецификацией стандарта. Примерный состав спектральных компонент полного телевизионного радиосигнала NICAM D/K (B/G) показан на рис. 3. Радиосигнал вещательного телевидения в общем случае состоит из четырех основных спектральных компонент — несущей частоты изображения, сигнала цветовой поднесущей (4,43 МГц — PAL, 4,25/4,406 МГц — SECAM), поднесущей аналогового звука (5,5 МГц — B/G, 6,5 МГц — D/K) и поднесущей NICAM (5,85 МГц).

При демодуляции такого радиосигнала с использованием стандартных телевизионных фильтров (серая штриховая кривая) происходит перенос спектральных компонент в область низких частот и ослабление сигналов поднесущих звука и NICAM. В этом случае рядом с сигналом NICAM оказываются более мощные спектральные компоненты сигнала цветности. Нали-

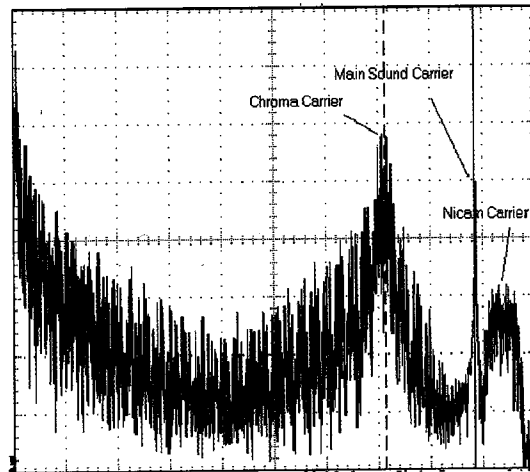
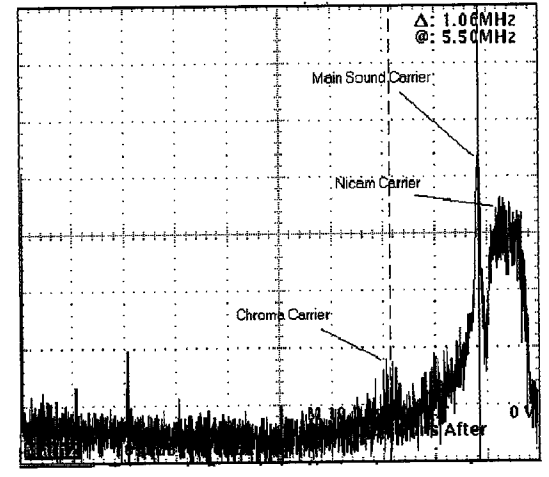


Рис. 4

чие таких компонент приводит к неустойчивой работе декодера NICAM, особенно при смене цветных сюжетов. Для устранения такого эффекта был разработан фильтр на ПАВ со специальной характеристикой, оптимизированной для приема сигналов NICAM B/G, D/K (голубая линия на рис. 3). Применение фильтра с такой характеристикой позволило значительно снизить уровень помех от поднесущей цветности и составляющих сигнала яркости, поднять уровень сигнала поднесущей NICAM в демодулированном сигнале и значительно улучшить устойчивость приема звука. На рис. 4 показаны спектрограммы сигнала на входе микросхемы демодулято-



ра NICAM без специализированного фильтра (а) и с ним (б).

Декодер работает в телевизоре в режиме квазипараллельного канала звука, т. е. на декодер подается сигнал первой ПЧ телевизора (38,9 МГц) с выхода телевизионного тюнера до телевизионного фильтра ПАВ. Такое построение декодера позволяет обеспечить наивысшее качество звука и максимально простое подключение к любому типу телевизионному приемнику.

ЛИТЕРАТУРА:

1. А. У. Жильцов и др. Экспериментальные исследования эфирного телевизионного радиосигнала со стереозвуковым сопровождением по системе NICAM 728. — Труды НИИР 2004.
2. Ю. А. Садиков. Особенности индивидуального приема ТВ программ со стереозвуком NICAM. — Схемотехника, 2004, № 3, с. 46, 47.

Окончание следует

Вячеслав Чулков,
г. Москва

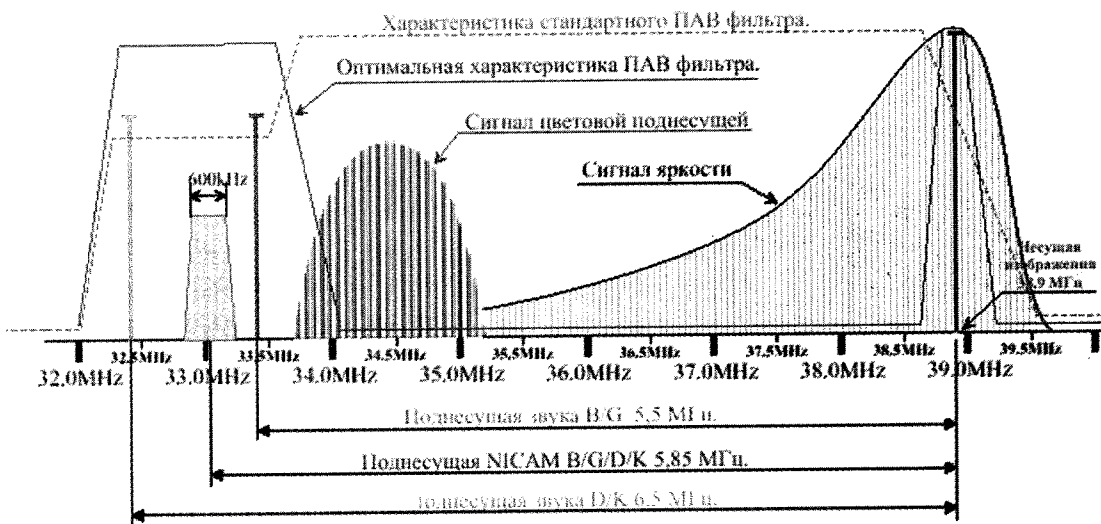


Рис. 3

См. также с. 55.

Коммутатор телевизионных сигналов

Для приема телевизионных программ телезрители зачастую применяют несколько антенн. В этом случае необходимы устройства, обеспечивающие подключение этих антенн к телевизионному приемнику. В статье рассматривается электронный коммутатор, обеспечивающий выбор необходимой антенны.

Расширение телевизионного вещания приводит к необходимости приема сигналов, входящих с разных направлений. В этом случае, как правило, применяют несколько направленных телевизионных антенн, ориентированных на источники сигналов.

Объединение сигналов в этом случае осуществляется как с помощью пассивных сумматоров, так и с помощью коммутаторов. Пассивные квадратурные и синфазные сумматоры наиболее удобны в эксплуатации, однако при сложении произвольных сигналов они вносят потери мощности и не обеспечивают оптимальной помехоустойчивости. Электромеханические коммутаторы лишены указанных недостатков, но они должны устанавливаться в удобном для пользователя месте, что не всегда возможно. Дистанционную коммутацию телевизионных сигналов можно обеспечить с помощью электронного устройства.

Схема электронного коммутатора телевизионных сигналов приведена на рис. 1.

Коммутация сигналов осуществляется микросхемой K561КТ3, представляющей собой четыре двунаправленных ключа. Согласно паспортным данным эта микросхема предназначена для коммутации сигналов с частотой не более сотен кГц, однако эксперименты выявили ее способность коммутации телевизионных сигналов. Для уменьшения потерь электронные контакты микросхемы соединены параллельно. Устройство обеспечивает коммутацию также и активных антенн, поскольку двунаправленные свойства переключателей микросхемы обеспечивают подачу напряжения питания на усилители антенн. Напряжение питания микросхем DD1 и DD2 обеспечивается блоком питания усилителя антенн с помощью цепочки R1C3, при этом потребляемая микросхемами мощность чрезвычайно мала.

Управление переключателем DD1

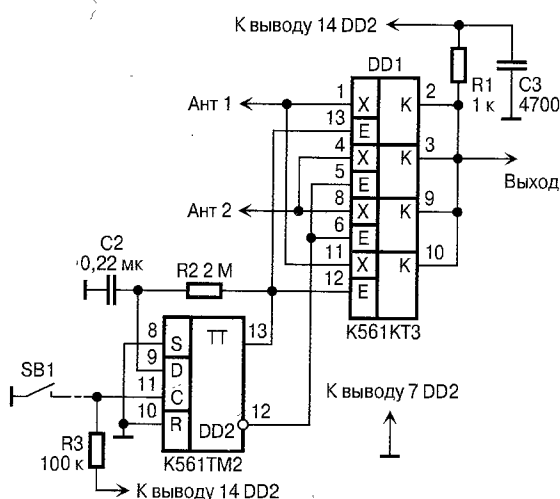


Рис. 1

осуществляется D-триггером DD2. При кратковременном нажатии на кнопку дистанционного управления SB1 состояние триггера изменяется на противоположное, что обеспечивает переключение цепей антенн. Узел на микросхеме DD2 обладает антидребезговыми свойствами. При поступлении очередного тактового импульса от кнопки SB1 на вход С триггера его состояние изменяется. Обратная связь на вход D осуществляется через цепочку R2C2 с достаточно большой постоянной времени, в течение которой D-триггер не реагирует на изменение сигнала на входе С.

Коммутатор установлен в корпус ранее использовавшегося пассивного сумматора.

Опытная эксплуатация коммутатора телевизионных сигналов показала высокую его эффективность. Он обеспечивает существенное снижение взаимных помех телевизионных сигналов, принимаемых с различных направлений, а также прием сигналов различных телевизионных станций, работающих на одном и том же телевизионном канале, но позволяющих осуществить пространственную селекцию. В случае необходимости число подключаемых антенн можно увеличить за счет некоторого усложнения схемы коммутатора.

Дмитрий Онышко,
г. Новочеркасск Ростовской обл.

AMPER 2005

13. международная ярмарка
электроники
и электротехники

5. - 8. 4. 2005

Прага - Чешская республика

750 участников

17 стран мира

35 000 м² выставочной
площади

- Электротехника
- Электроника
- Кабели
- Приводы
- Измерение
- Автоматизация
- Освещение
- Обеспечение

Теринвест тоо,
Легерова 15, 120 00 Прага 2
Чешская республика
тел: +420 221 992 134,126
www.amper.cz

Музыкальный звонок

В журнале «Радио» была опубликована схема звонка на 120 мелодий [1]. Автор статьи собрал один экземпляр и остался не удовлетворен его работой. Использование приведенных в статье рекомендаций, собственные доработки, консультации с автором схемы так и не смогли довести звонок до требуемого результата. Он периодически начинал воспроизводить несколько мелодий подряд, иногда завершая звучание неприятным «хрюканьем». Пришлось перерабатывать звонок практически полностью.

Основой звонка (рис. 1), как и в предыдущей конструкции, является микроконтроллер. В данной конструкции применен микроконтроллер фирмы Microchip. Аргументами для его выбора являлись специально предусмотренные аппаратные узлы, защищающие внутреннюю EEPROM от неправильной записи и осуществляющие сброс при снижении напряжения питания. Применение данного микроконтроллера позволило без каких-либо затрат решить проблему со сбоями информации во внутренней EEPROM, описанные автором предыдущей конструкции, а также проблему с «перескоком» мелодий при воспроизведении.

Изменен и способ хранения мелодий. Теперь мелодии хранятся во внешней микросхеме энергонезависимой памяти. Такая конструкция, во-первых, позволила сократить размеры устройства — вместо одной большой 40-выводной микросхемы удалось применить две в гораздо более миниатюрных 8- и 14-выводных корпусах, и, во-вторых, значительно снизить его стоимость. Кроме того, применение шины I²C позволяет без каких-либо доработок увеличить

объем памяти для мелодий до 64 Кб вместо 8 Кб. Подсчитать точное число мелодий, которое уместится в таком объеме, невозможно, т. к. оно зависит от их длины. Однако примерно можно предположить, что если в 8 Кб можно записать 120 мелодий (реально еще больше, т. к. в этой конструкции весь объем ПЗУ отведен под мелодии), то в 64 Кб их должно уместиться около 960 (!). При одном звонке в дверь в день для их прослушивания понадобится более чем 2,5 года. Однако возникает другая проблема — где их столько взять?

При разработке данной конструкции были максимально использованы как схемотехнические, так и программные разработки от предыдущей конструкции. Таблица нот и длительностей осталась прежней. Это позволило создать файл прошивки для микросхемы DD1 простым перекодированием ассемблерного файла мелодий mel.inc [1] в двоичный. Ноте с порядковым номером 1 («С-») соответствует код 01H, 2 («С#-») — 02H и т. д. Длительность кодируется аналогично: 1 соответствует 01H, 2 — 02H и т. д. Паузе

соответствует код FFH, после которого следует код, указывающий ее длительность. В конце мелодии ставится код FFH, а в конце последней мелодии — код FDH (код FFH в этом случае ставить не надо). Применение этих кодов позволяет применить ПЗУ любой емкости, однако принципиальные отличия в системе адресации ячеек ПЗУ объемом 128...2048 байт и 4096...65536 байт сужают допустимый диапазон замены микросхем. В данной конструкции возможно применение микросхем ПЗУ с шиной I²C и двухбайтовой адресацией ячеек. Этому условию соответствуют ПЗУ типа 24C32—24C512.

Алгоритм работы выглядит следующим образом. После подачи напряжения питания кратковременным замыканием контактов звонковой кнопки SB1 микроконтроллер производит настройку внутренних узлов и блокирует кнопку контактами реле K1. Затем он считывает из микросхемы DD1 байт и определяет, является ли он управляющим (FFH, FDH). Если нет, то считывается следующий байт — код длительности. Программный счетчик адреса ячейки в ПЗУ увеличивается на два (по числу считанных байтов), и микроконтроллер начинает воспроизводить ноту. Шестнадцатиразрядный Таймер 1 используется для генерации тона, а восьмиразрядный Таймер 0 определяет длительность. Так как разрядности Таймера 0 и подключенного к нему предварительного делителя не хватает для правильного отсчета длительности, то вместе с ними применяется еще и программный счетчик.

По окончании звучания ноты микроконтроллер считывает по уже подготовленному адресу следующую ноту и так далее до одного из управляющих кодов. Если встретится код конца мелодии (FFH), то микроконтроллер переписет адрес первой ноты следующей мелодии во внутреннюю энергонезависимую память и отключит питание звонка. Обработка кода конца всех мелодий (FDH) отличается от описанной выше только тем, что перед записью во внутреннюю память микроконтроллер сбросит счетчик адреса в нулевое значение.

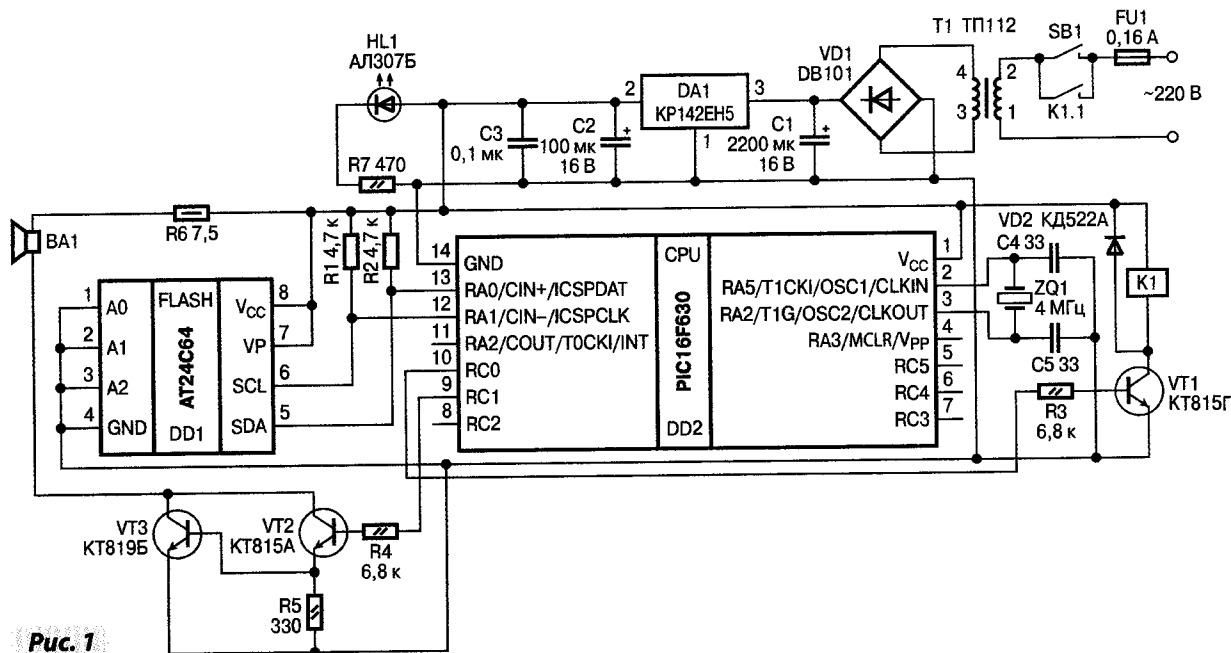


Рис. 1

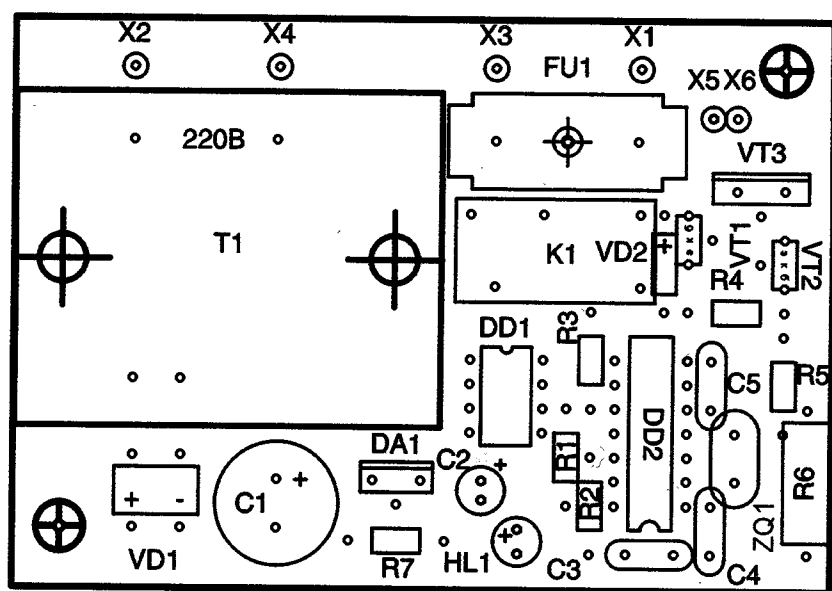


Рис. 2

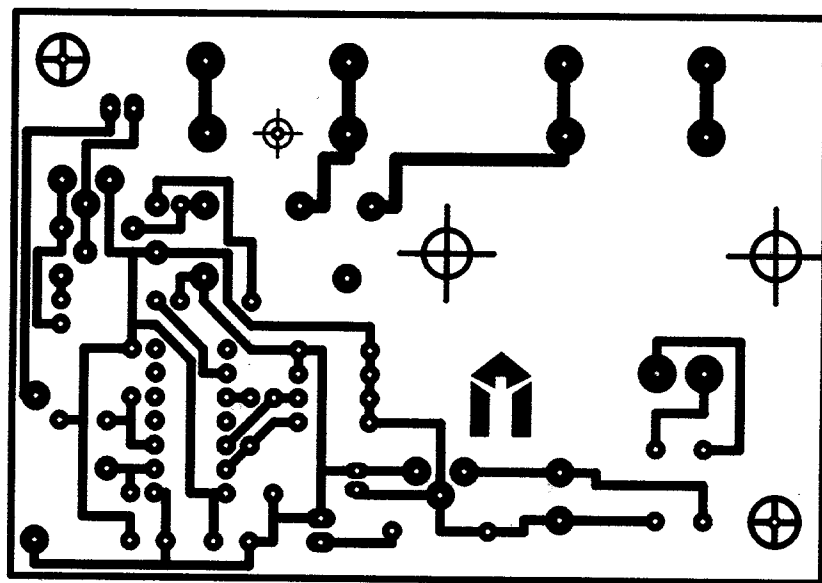


Рис. 3

Конструктивно звонок собран на печатной плате из односторонне фольгированного стеклотекстолита размером 86×60 мм. На рис. 2 показано размещение деталей, а на рис. 3 — топология печатных проводников.

Приобретение относительно нового микроконтроллера PIC16F630 может вызвать у радиолюбителей определенные трудности, в этом случае можно порекомендовать адаптировать программу к более распространенным микроконтроллерам PIC16F627/628. Для этого вместе с файлом прошивки *zvonok.hex* на сайте журнала по адресу www.dian.ru/programs и файл с исходным текстом — *zvonok.asm*. Программирование микроконтроллера производилось с помощью программы IC-Prog версии 1.05C [2] и программатора, опубликованного в [3]. Выводы микроконтроллера подключались к программатору в соответствии с таблицей, приведенной в [4].

В качестве ПЗУ мелодий в данной конструкции применена микросхема 24C64 объемом 8 Кб. Под этот

же объем рассчитан и файл мелодий *mel.bin*. Он представляет собой перекодированный по приведенному выше алгоритму файл *mel.txt* и является обычным двоичным файлом.

Если с программированием микроконтроллера никаких проблем не возникло, то запрограммировать ПЗУ оказалось достаточно сложно. Используемые чаще всего радиолюбителями программы IC-Prog и PonyProg согласно спецификации поддерживают данный тип (24C64), однако на практике они совершенно не хотели работать с ПЗУ, имеющим двухбайтовую адресацию, хотя микросхемы меньшего объема с однобайтовой адресацией программировали отлично. Чем это вызвано, трудно сказать, однако создается ощущение, что авторы этих программ просто забыли ввести поддержку двухбайтовой адресации. Пришлось самому создать программатор для микросхем ПЗУ, использующих шину I²C. Принципиальная схема адаптера, подключаемого к порту LPT, показана на рис. 4. А вот с программой, его поддерживающей, могут

быть определенные трудности, т. к. она использует “shareware” компонент TVicLPT (организующий работу с LPT портом) и поэтому при запуске программы появляется окно, напоминающее о том, что его надо купить. Для правильной работы программы необходимо этот компонент установить, т. к. вместе с ним устанавливается драйвер LPT порта, обеспечивающий доступ к его ресурсам, в том числе и под операционными системами Windows NT/2000/XP. Найти компонент можно по адресу <http://www.entechtaiwan.com/dev/lpt/index.shtml>.

Кому-то данные сложности с программированием могут показаться лишними, хотя при единичном использовании это не должно сильно мешать.

Следует также добавить, что данная методика программирования ПЗУ не является единственно правильной. Автор ни в коей мере не настаивает на ее использовании. Возможно, что к моменту выхода статьи, в Интернете появятся новые версии программаторов, поддерживающих все типы ПЗУ с шиной I²C.

ЛИТЕРАТУРА:

1. М. Белецкий. Музыкальный звонок на 120 мелодий. — Радио, 2004, № 2, с. 33—34.
2. IC-Prog Prototype Programmer. <http://www.ic-prog.com>.
3. А. Долгий. Разработка и отладка устройств на МК. — Радио, 2001, № 6, с. 24—26, № 7, стр. 19—21.
4. А. Долгий. Программаторы и программирование микроконтроллеров. — Радио, 2004, № 1, с. 53—55.

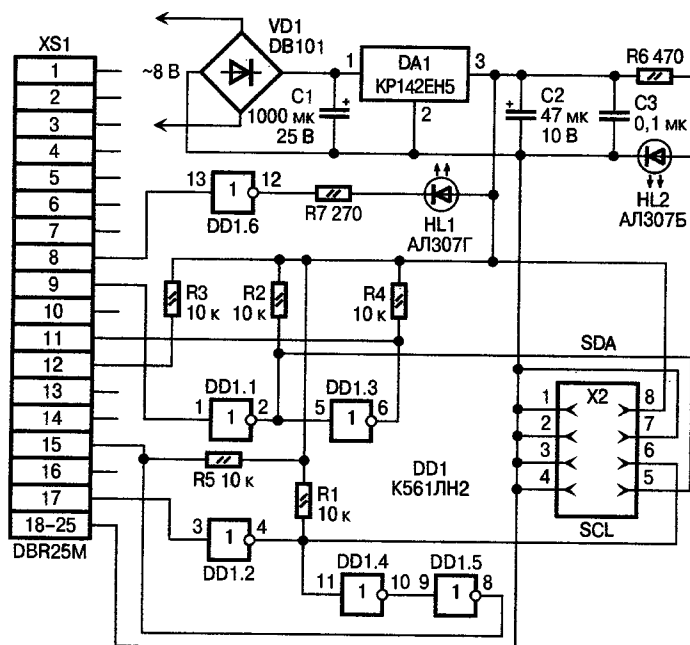


Рис. 4

Сергей Малахов,
г. Санкт-Петербург

Дополнительный порт LPT

Перед первым включением необходимо убедиться в правильности монтажа всех элементов и отсутствии замыканий. Отладку адаптера в статическом режиме можно выполнить на эмуляторе шины ISA [2]. После подачи питания необходимо запрограммировать микросхему D1. Файл прошивки для нее под названием isa-lpt.jed можно найти на сайте журнала по адресу <http://www.dian.ru/programs>. Перед подключением адаптера к компьютеру для избежания конфликтной ситуации необходимо убедиться, что адресное пространство, используемое адаптером, не занято никаким другим устройством. Если необходимые адреса все же заняты, требуется перепрограммировать адаптер на новый адрес, либо освободить эти адреса в самом компьютере переносом на свободные. Для изменения базового адреса в адаптере необходимо лишь перепрограммировать микросхему D1, изменив предварительно схему дешифратора адресов. Эти изменения очень просты и сводятся к удалению или установке инверторов в адресные линии дешифратора. Для облегчения этой задачи необходимо воспользоваться табл. 5, в которой показано, какие именно адресные линии должны содержать инвертор для соответствующего базового адреса адаптера. В соответствии с приведенной выше схемой программируемой логики D1 в адаптере будет запрограммирован базовый адрес 278h.

Таблица 5

Адресные линии адаптера										Базовый адрес адаптера
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	1	1	1	0	X	X	0x378
1	0	0	1	1	1	1	0	X	X	0x278*
1	1	1	0	1	1	1	1	X	X	0x3BC

Условные обозначения: 0 — наличие инвертора необходимо; 1, X — наличие инвертора не требуется; * — адрес, запрограммированный в схеме.

После выполнения перечисленных выше операций можно подключить адаптер и включить компьютер. Через несколько секунд после включения компьютера на экране монитора появится таблица устройств, обнаруженных BIOS. Для остановки

компьютера в этом месте необходимо нажать клавишу Pause. Обратите внимание на строку с описанием параллельных портов. Эта строка обычно содержит лишь один адрес. При подключенном к компьютеру адаптере эта строка должна иметь запись примерно следующего вида — Parallel Port(s): 378, 278. Это будет означать, что адаптер обнаружен программой BIOS и компьютер имеет теперь два порта LPT. После нажатия любой клавиши компьютер продолжит загрузку операционной системы.

В операционной системе DOS убедиться в наличии нового порта LPT можно с помощью стандартной утилиты диагностики msd.exe. После запуска этой утилиты на экране монитора должно появиться изображение, подобное тому, что приведено на рис. 7.

Если щелкнуть в этой программе мышкой по кнопке LPT Ports или просто нажать клавишу L, на экране должно появиться окно, изображенное на рис. 8. В нем дается подробная информация о числе обнаруженных в компьютере портов LPT и их состоянии.

При работе в операционной системе Windows 98 и в более поздних версиях Windows адаптер LPT обнаруживается, как правило, автоматически. Если этого не произошло, необходимо один раз выполнить процедуру поиска и установки порта LPT. Для этого необходимо выполнить следующие действия — Пуск→Настрой-

ка→Панель управления→Установка оборудования. После этого Windows предлагает автоматически произвести поиск новых устройств, обнаружить новый адаптер LPT и установить для него драйверы. Поскольку наш адаптер LPT будет обнаружен

как стандартное устройство, драйверы для него будут установлены из инсталляционного пакета Windows. В дальнейшем операционная система Windows будет осуществлять поддержку нового порта в полном объеме. В списке оборудования компьютера теперь будет присутствовать еще один порт LPT. Для того, чтобы убедиться в этом, необходимо посмотреть свойства компьютера. На рис. 9 показана копия экрана компьютера с операционной системой Windows98 и установленным адаптером LPT.

Тестирование нового адаптера в составе компьютера можно выполнить с помощью тестовой программы Checkit или Norton Diagnostic. Заглушки для этих тестовых программ можно изготовить, воспользовавшись табл. 6 и 7.

Кроме того, для проверки адаптера LPT и отладки устройств, подключаемых к нему, можно воспользоваться универсальной программой lpt-test, описанной в [3]. Найти эту программу можно на сайте журнала по адресу <http://www.dian.ru/programs/index.html> в архиве 2003_01_Valpa_test_lpt.

Для программной поддержки LPT портов в BIOS компьютера имеется специальное программное прерывание INT17h. Ниже дается описание его функций. Выбор номера функции этого прерывания задается в регистре процессора AH.

Функция: 0 — печать символа

Вход : AH = 0, AL = символ, DX = номер порта (0 = LPT1, 1 = LPT2 или 2 = LPT3)

Выход: AH = статус порта

Функция: 1 — инициализация порта

Вход : AH = 1, AL = пусто, DX = номер порта (0 = LPT1, 1 = LPT2 или 2 = LPT3)

Выход: AH = статус порта

Функция: 2 — чтение статуса порта

Вход: AH = 2, AL = пусто, DX = номер порта (0 = LPT1, 1 = LPT2 или 2 = LPT3)

Выход: AH = статус порта (1 в разряде)

0 - тайм-аут

1 - x

2 - x

3 - ошибка

4 - принтер выбран

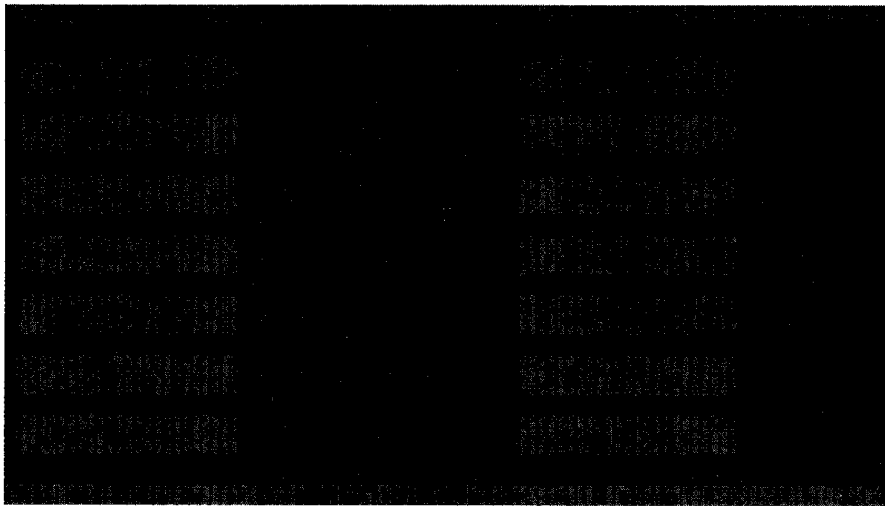


Рис. 7

Port	Port Address	On Line	LPT Ports				
			Paper Out	I/O Error	Time Out	Busy	ACK
LPT1:	0378H	No	No	No	No	No	No
LPT2:	0278H	Yes	Yes	No	No	No	No
LPT3:	-	-	-	-	-	-	-

OK

Рис. 8

Таблица 6

Сигнал	Направление	Вывод	Вывод	Направление	Сигнал
-Strobe	Выход	1	13	Вход	Select
Data 0	Выход	2	15	Вход	-Error
-Init	Выход	16	10	Вход	-Ack
-Slct In	Выход	17	11	Вход	Busy
-Auto LF	Выход	14	12	Вход	Paper End

Таблица 7

Сигнал	Направление	Вывод	Вывод	Направление	Сигнал
Data 0	Выход	2	15	Вход	-Error
Data 1	Выход	3	13	Вход	Select
Data 2	Выход	4	11	Вход	Busy
Data 3	Выход	5	10	Вход	-Ack
Data 4	Выход	6	12	Вход	Paper End

- 5 - конец бумаги
- 6 - подтверждение
- 7 - принтер не занят

Для примера, вызов функции 1 данного прерывания на Ассемблере выглядит следующим образом:

```
MOV AH, 01 ; функция 1
MOV DX, 01 ; номер порта 1 =
LPT2
INT 17 ; вызов прерывания 17h
```

Теперь в регистре AH хранится статус порта.

Ниже приведена последовательность действий для вывода на печать одного символа:

- вывести передаваемый байт в регистр данных;
- в цикле проверять состояние принтера до установки разряда регистра состояния (здесь возможно использование тайм-аута);
- проверить биты 3—5 регистра состояния на наличие ошибки;
- установить и сразу же сбросить нулевой бит регистра управления, для этого подходит следующая последовательность команд:

```
MOV DX, 0x037A ; адрес регистра управления
MOV AL, 00001101b ; установить биты 3, 2 и 0
OUT DX, AL ; установить нулевой
```

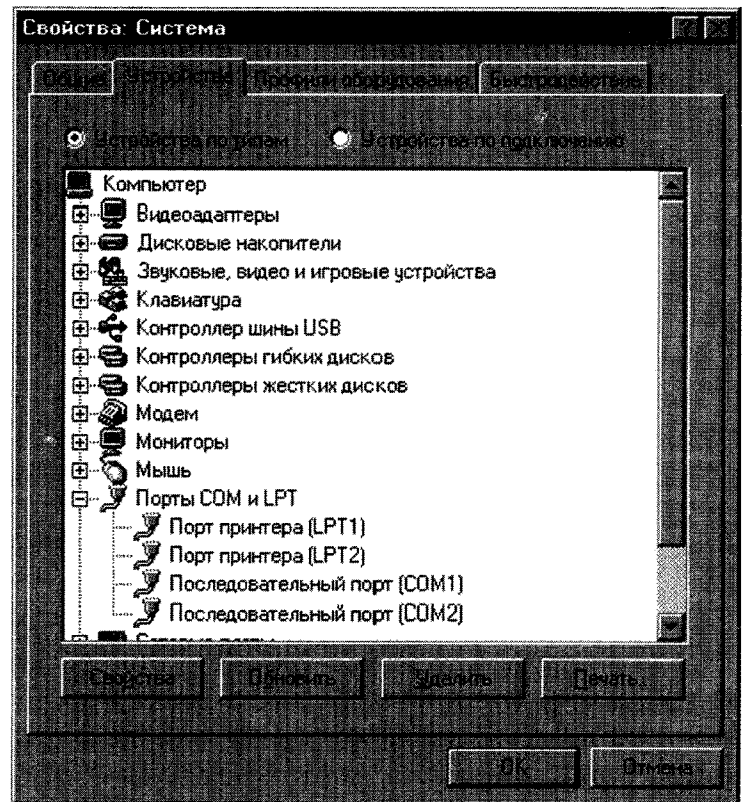


Рис. 9

бит регистра управления

XOR AL, 1 ; сбросить бит 0

OUT DX, AL ; сбросить нулевой

бит регистра управления

далее можно прочитать регистр состояния и перейти к печати следующего символа.

Как было показано выше, компьютер можно дополнить несколькими портами LPT. Изготовив два адаптера LPT с разными базовыми адресами и установив их в компьютер, можно получить компьютер с двумя новыми портами. Возможно также переработать рассмотренный здесь адаптер, превратив его сразу в двухпортовый. Второй разъем при этом придется устанавливать на дополнительную планку ввиду ограничения высоты планки на самом адаптере. Подключение второго разъема к адаптеру можно осуществить с помощью шлейфа с обжимным соединителем типа IDC-26, широко распространенным в компьютерной технике. На самом адаптере должна быть установлена ответная часть для такого разъема с сигнальными цепями второго порта LPT.

Олег Вальпа,
г. Миасс Челябинской обл.

«Мастер КИТ» предлагает «Декодер телевизионного стереозвукового сопровождения формата NICAM NM2905». В комплект поставки входит настроенный и полностью готовый к работе блок, а также подробная инструкция по его установке в телевизор.

Вся продукция «Мастер КИТ» представлена на сайте www.masterkit.ru. Пожелания и предложения сообщайте по e-mail: infomk@masterkit.ru, по тел.: (095) 234-77-66, 109044, Москва, а/я 19.

ПОДПИСКА — 2005

Журнал «СХЕМОТЕХНИКА»

www.dian.ru

Через каталог Агентства «Роспечать» (красный) — индекс 80724 (для РФ), 82177 (стран Балтии и СНГ).

Через Объединенный каталог «Пресса России» (зеленый) — индекс 41733.

Жители Украины могут подписаться через каталог агентства KSS (044-212-0050, 464-0220) — индекс 10540.

Жители Латвии могут подписаться через каталог агентства PKS (371-7320148).

Журнал «КОМПОНЕНТЫ и ТЕХНОЛОГИИ»

www.compitech.ru

Через каталог Агентства «Роспечать» (красный) — индекс 80743.

Через Объединенный каталог «Пресса России» (зеленый) — индекс 41734.

Жители Украины могут подписаться через каталог агентства KSS (044-212-0050, 464-0220) — индекс 10358.

Жители Латвии могут подписаться через каталог агентства PKS (371-7320148).

Оформить подписку с последующей доставкой можно также в любой стране через фирму МК-Periodica. Подробную информацию Вы можете узнать по телефону в Москве (7+095) 2819345, 2815715, 2813322, e-mail: info@periodicals.su

Условия подписки через редакцию:

Комплект журнала «Схемотехника» № 1—12 за 2005 г. (стоимость доставки включена) — 528 руб. (РФ)	Комплект журнала «Компоненты и Технологии» № 1—9 за 2005 г. (стоимость доставки включена) — 1080 руб.
Комплект журнала «Схемотехника» № 1—12 за 2004 г. (стоимость доставки включена) — 504 руб. (РФ)	Комплект журнала «Компоненты и Технологии» № 1—9 за 2004 г. (стоимость доставки включена) — 900 руб. (РФ)
Комплект журнала «Схемотехника» № 4—7, 10—12 за 2003 г. (стоимость доставки включена) — 475 руб. 20 коп. (РФ)	Вы можете выписать отдельные номера журнала «Компоненты и Технологии» за 2000—2004 г. (стоимость доставки включена) — 100 руб. О наличии номеров узнавайте в редакции по тел. (095) 777-1215, e-mail: podpiska@dian.ru
Вы можете выписать отдельные номера журнала «Схемотехника» за 2001—2004 г. (стоимость доставки включена) — 42 руб. О наличии номеров узнавайте в редакции по тел. (095) 777-1215, e-mail: podpiska@dian.ru	

- перечислите деньги на наш расчетный счет через Сбербанк по квитанции, либо через почтовое отделение почтовым переводом;
- отправьте квитанцию об оплате (или копию) и свой точный почтовый адрес (индекс обязательно) в редакцию по почтовому адресу: 127015, Москва, ул. Бутырская, д. 41/47, ООО «ИД Скимен», отдел подписки или по факсу (095) 777-1215, e-mail: podpiska@dian.ru.
- вы можете также подписаться на журнал непосредственно в редакции по адресу: г. Москва, ул. Бутырская, д. 41/47 (метро «Савеловская», во дворе д. 17, вход с торца здания) и на специализированных выставках.

НАШИ РЕКВИЗИТЫ:

ООО «ИД СКИМЕН» ИНН 7731195492 КПП 773101001
Р/с 40702810200000005646 в ОАО КБ «Национальный космический банк» г. Москва
К/с 30101810900000000278 БИК 044579278 ОКОНХ 84500, ОКПО 52744508

Внимание! К сведению подписчиков: отдел подписки не несет ответственности, если подписка оформлена через другие фирмы. В случае отмены заказчиком произведенной подписки деньги за подписку не возвращаются.

Информация о подписчике:

Название организации _____ Ф.И.О. _____

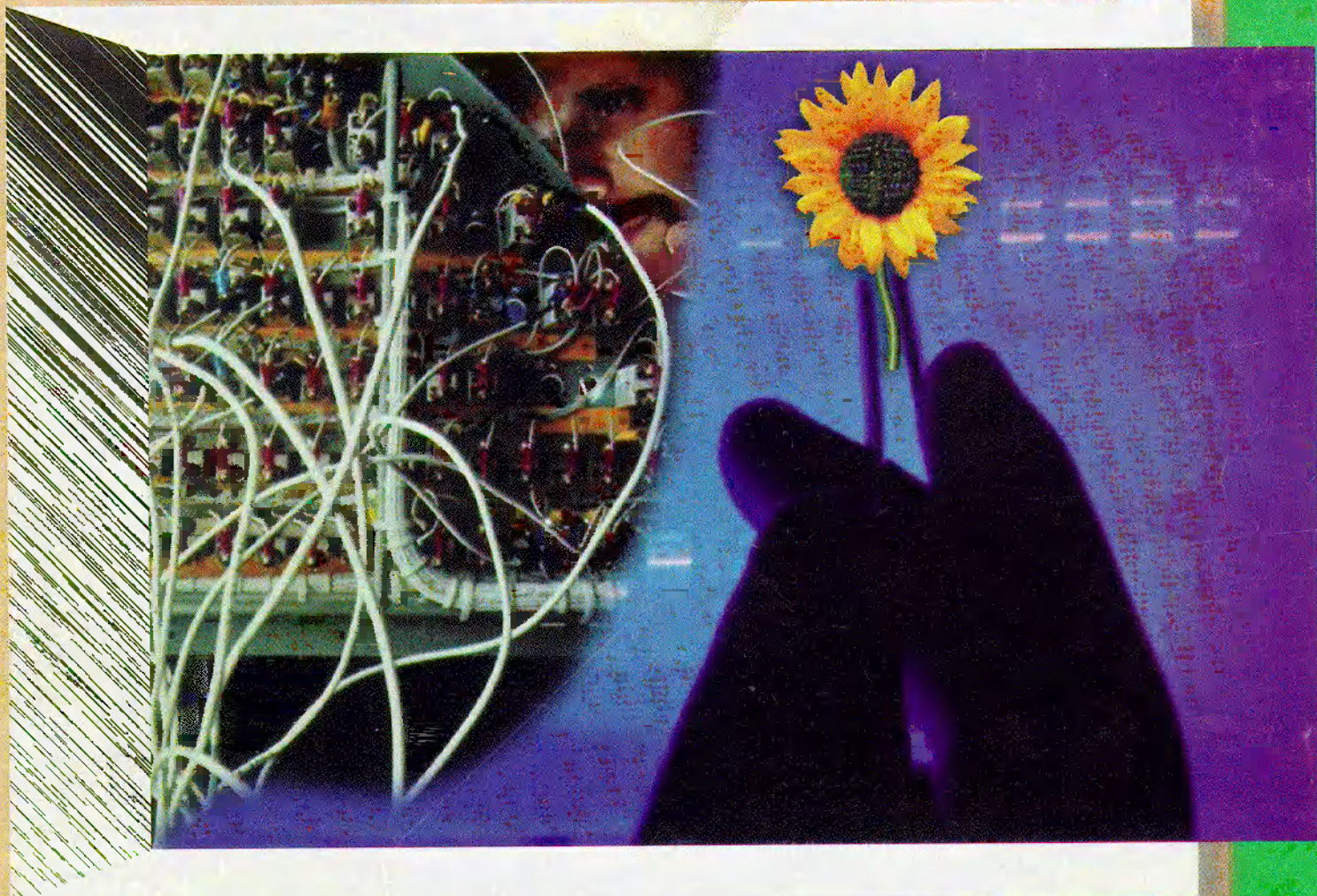
ИНН _____ КПП _____ (для организаций)

Почтовый адрес, индекс _____

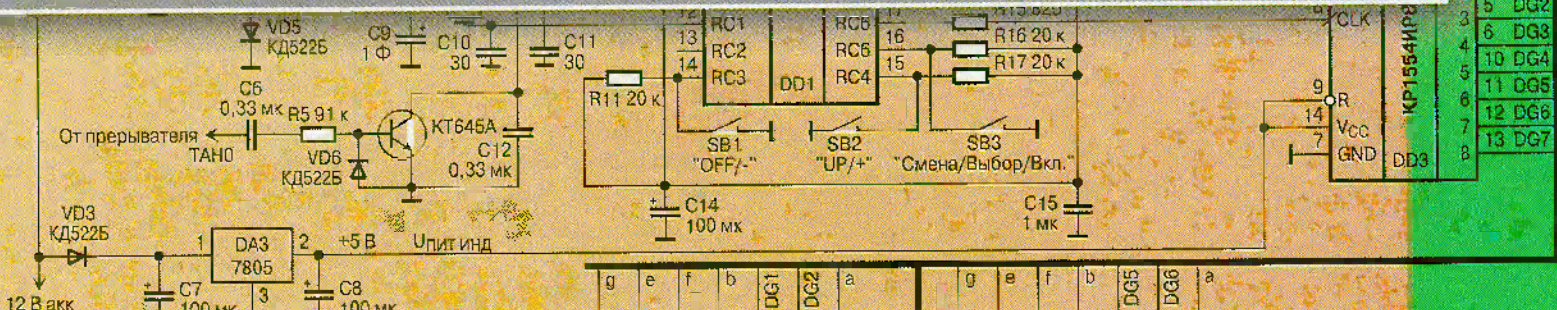
Телефон/Факс _____ E-mail _____

Подписка на журнал _____ № _____ за 200__ год

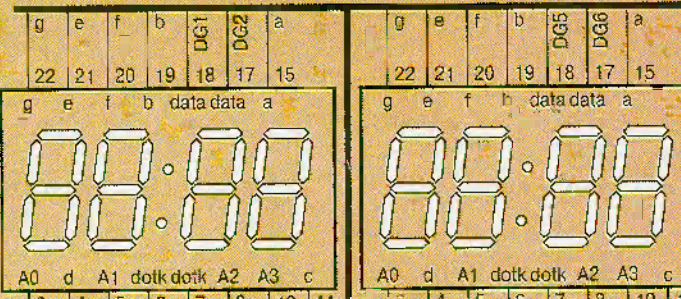
348



Автомат управления насосом на микроконтроллере HT48R05A
 Операционные усилители фирмы ROHM
 Повышение точности измерительного трансформатора тока
 Генератор с шунтирующим диодом
 Мультиметр-часы для автомобиля на PIC16F873
 ТВ стереозвуковой декодер для NICAM 728



a
b
c
d
e
f
g
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22



подписные индексы
 80724 . 82117
 41733