

МИКРОСХЕМЫ ИНТЕГРАЛЬНЫЕ
1830BE32У, 1830BE52У, 1830BE32АУ

Руководство пользователя

Содержание

1 Введение	5
2 Назначение и основные технические характеристики микросхем 1830BE32У, 1830BE32АУ, 1830BE52У	7
2.1 Архитектурные характеристики микросхем	7
2.2 Конструктивные характеристики микросхем	8
2.3 Электрические характеристики микросхем	8
3 Общая характеристика микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У	15
4 Описание устройства	23
4.1 Структурная организация	23
4.2 Организация ОЗУ, ПЗУ и регистров микроконтроллеров	25
4.3 Организация портов ввода-вывода	31
4.4 Работа с внешней памятью	37
4.5 Внутренняя память программ ИС 1830BE52У	41
4.6 Устройство управления и синхронизации	42
4.7 Таймеры/счетчики.....	43
4.8 Система прерываний	47
4.9 Последовательный интерфейс	52
4.9.1 Регистр управления/статуса приемопередатчика SCON	55
4.9.2 Скорость приема/передачи последовательного порта	57
4.9.3 Особенности работы приемопередатчика в различных режимах.....	59
4.10 Работа приемопередатчика в мультимикропроцессорных системах	61
4.11 Сброс микроконтроллера	63
4.12 Сброс по включению питания	65
4.13 Режимы работы микроконтроллера с пониженным энергопотреблением	66
4.14 Режим холостого хода	67

4.15	Режим выключенного напряжения питания	69
4.16	Встроенный тактовый генератор.....	69
5	Общая характеристика системы команд микроконтроллеров.....	73
5.1	Группа команд пересылки данных.....	77
5.2	Группа команд арифметических операций	80
5.3	Группа команд логических операций	83
5.4	Группа команд операций над битами	85
5.5	Группа команд передачи управления	86
5.6	Общая характеристика команд.....	90
5.6.1	Арифметические команды	90
5.6.2	Команды битового процессора.....	91
5.6.3	Логические команды с байтовыми переменными	92
5.6.4	Команды передачи данных	92
5.6.5	Команды ветвления и передачи управления.....	93
6	Описание системы команд микроконтроллера	95
7	Адресация.....	100
7.1	Косвенно-регистровая адресация.....	100
7.2	Непосредственная адресация.....	100
7.3	Прямая адресация	101
7.4	Регистровая адресация	101
8	Особенности микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У по сравнению с ИС Н1830BE31, Н1830BE51	103
8.1	Внутренняя память данных.....	103
8.2	Пространство регистров специальных функций	104
8.3	Таймер 2	105
8.3.1	Режим захвата (фиксации)	107
8.3.2	Режим автоперезагрузки	108
8.3.3	Режим сигнала запрограммированной частоты	110
8.3.4	Режим задающего генератора последовательного порта	111
8.4	Программируемый массив счетчиков РСА.....	112

8.4.1 Модули фиксации-сравнения блока PCA.....	116
8.4.2 Режим фиксации блока PCA.....	117
8.4.3 Режимы сравнения блока PCA	118
8.4.4 Режим программируемого 16-разрядного таймера	119
8.4.5 Режим скоростного вывода.....	120
8.4.6 Режим сторожевого таймера.....	120
8.4.7 Режим широтно-импульсной модуляции	122
8.5 Система прерываний	123
8.6 Особенности приемопередатчиков в ИС 1830BE32У, 1830BE32AU, 1830BE52У	124
8.7 Режим пониженного энергопотребления	126
8.8 Специальный режим эмуляции ONCE	127
9 Описание инструментальных средств для микроконтроллеров.....	128
9.1 Интегрированный пакет разработки и отладки систем на базе микроконтроллеров семейства 8051	128
9.2 Внутрисхемный эмулятор 8-разрядных микроконтроллеров семейства 8051	129
9.3 Характеристики аппаратуры.....	130
9.4 Характеристики программного обеспечения.....	132
9.5 Компоненты эмулятора	132
9.6 Отладочные возможности PDS-52	135
9.7 Кросс-макроассемблер	136
10 Заключение.....	138
Лист регистрации изменений	139

1 Введение

В настоящем руководстве КФДЛ.431281.026 приведено описание архитектуры, функционального построения, системы команд и особенностей применения микросхем 1830BE32У, 1830BE32АУ, 1830BE52У, которые представляют собой однокристалльные интегральные микросхемы 8-разрядных микроконтроллеров с тактовой частотой 12 МГц для ИС 1830BE32У, 1830BE52У и 16 МГц для ИС 1830BE32АУ, встроенным мажорированным (с тройным резервированием) ОЗУ емкостью (256×8) бит с повышенной устойчивостью к одиночным сбоям, без встроенной памяти программ для ИС 1830BE32У, 1830BE32АУ и с масочным вариантом встроенной памяти программ объемом $(4К \times 8)$ бит для 1830BE52У, портом последовательного ввода-вывода и отвечающие требованиям группы 3У_с – 6У_с по ГОСТ РВ 20.39.414.2-98.

В настоящее время одним из основных факторов обеспечения конкурентоспособности отечественной радиоэлектронной аппаратуры и ее живучести является применение при ее разработке и производстве импортонезависимой элементной базы.

Импортозамещение электронных компонентов наиболее эффективно в случае использования полных функциональных аналогов изделий микроэлектроники. Появление отечественных вариантов высокопроизводительных 8-разрядных микроконтроллеров для цифровых систем управления призвано обеспечить импортонезависимость проектируемых на его основе устройств. К числу критичных технологий, характеризующих интеллектуальный потенциал нации, безусловно, относятся радиационно-стойкие технологии изготовления изделий электронной техники. Радиационная стойкость полупроводниковых структур, являющаяся одной из важнейших характеристик технологии, во многом определяет вероятность безотказной работы соответствующих систем и, как следствие, величину

срока активного существования аппаратуры в условиях повышенной радиации.

Одним из методов, значительно повышающих безотказность работы, является мажорирование оперативной памяти или метод тройного резервирования оперативной памяти. Блок исправления ошибок выбирает правильное значение схемой большинства голосов на основании выходов трех ОЗУ. Применение этого метода в микросхемах 1830BE32У, 1830BE32АУ, 1830BE52У позволяет устранять как одиночные сбои, так и сбои, произошедшие в нескольких ячейках ОЗУ.

Создание надежных высокоэффективных радиационно-стойких электронных устройств является важнейшей проблемой, требующей разработки методов и средств расчёта, моделирования, оптимизации схемотехнических и конструктивно-технологических решений, создания микроэлектронных приборов, схем и устройств, а также разработки методик для эффективной оценки потенциально ненадежных конструктивно-технологических решений для изучения последствий ионизирующих излучений. Микроконтроллеры 1830BE32У, 1830BE32АУ, 1830BE52У послужат цели развития отечественной элементной базы для применения в различных цифровых системах управления, радиосвязи и других изделиях, использующихся в условиях повышенной радиации (на объектах атомной энергетики).

Микроконтроллеры 1830BE32У, 1830BE32АУ, 1830BE52У "сверху вниз" программно и аппаратно совместимы с серийно выпускаемыми отечественными микроконтроллерами Н1830BE31 и Н1830BE51.

Настоящее руководство КФДЛ.431281.026 может служить практическим пособием по применению микроконтроллеров для разработчиков систем на основе микросхем 1830BE32У, 1830BE32АУ, 1830BE52У.

2 Назначение и основные технические характеристики микросхем 1830BE32У, 1830BE32АУ, 1830BE52У

Микросхемы предназначены для применения во встроенных цифровых системах управления комплексами радиосвязи, встроенных цифровых системах управления, бортовой аппаратуре, средствах оповещения, для управления робототехническими комплексами, в системах автоматизации технологических процессов, в системах автоматизированного управления электроприводом, оргтехнике, вычислительной технике, телекоммуникационной технике и т. п., к которым предъявляют высокие требования при работе в условиях повышенной радиации.

Изделия полностью совместимы по выводам, системе команд с серийно выпускаемыми в Российской Федерации микроконтроллерами Н1830BE31 и Н1830BE51, которые являются функциональными аналогами разработанных микросхем.

2.1 Архитектурные характеристики микросхем

Микросхемы имеют следующие архитектурные характеристики:

- Разрядность АЛУ, бит	8
- Встроенная память программ типа ROM (для микросхемы 1830BE52У, в микросхемах 1830BE32У, 1830BE32АУ отсутствует), бит	4К × 8
- Регистровое ОЗУ, бит	256 × 8
- Таймеры, бит	3 × 16
- Порты ввода-вывода, бит	4 × 8
- Последовательные порты	1
- Источники прерываний	7
- Режимы энергосбережения	да
- Напряжение питания для 1830BE32У, 1830BE52У, В	(5 ± 0,5)
- Напряжение питания для 1830BE32АУ, В	(3,3 ± 0,3)

- Тактовая частота для 1830BE32У, 1830BE52У, МГц	12
- Тактовая частота для 1830BE32АУ, МГц	16
- Длительность командного цикла для 1830BE32У, 1830BE52У, нс	1 000
- Длительность командного цикла для 1830BE32АУ, нс	750
- Корпус	Н16.48-2В
- Диапазон температур окружающей среды, °С	от минус 60 до 85
- Технология изготовления	КМОП-КНИ

2.2 Конструктивные характеристики микросхем

Микросхемы выполняются в металлокерамическом планарном корпусе с четырехсторонним расположением выводов Н16.48-2В и предназначены для ручной и автоматической сборки в соответствии с ГОСТ В 20.39.412 - 97.

Масса микросхем не более 3 г.

Показатель герметичности микросхем со свободным внутренним объемом по эквивалентному нормализованному потоку должен быть не более $6,65 \cdot 10^{-3}$ Па·см³/с.

Микросхема не имеет собственных резонансных частот ниже 100 Гц.

Более подробно конструктивные характеристики микросхем 1830BE32У, 1830BE32АУ, 1830BE52У приведены в технических условиях АЕЯР.431280.378ТУ.

Условное графическое обозначение микросхем приведено на рисунке 2.1.

Функциональное назначение выводов приведено в таблице 2.1.

2.3 Электрические характеристики микросхем

- Электрические характеристики микросхем при приемке и поставке приведены в таблице 2.2.

- Значения предельно допустимых электрических режимов эксплуатации в диапазоне рабочих температур приведены в таблице 2.3.

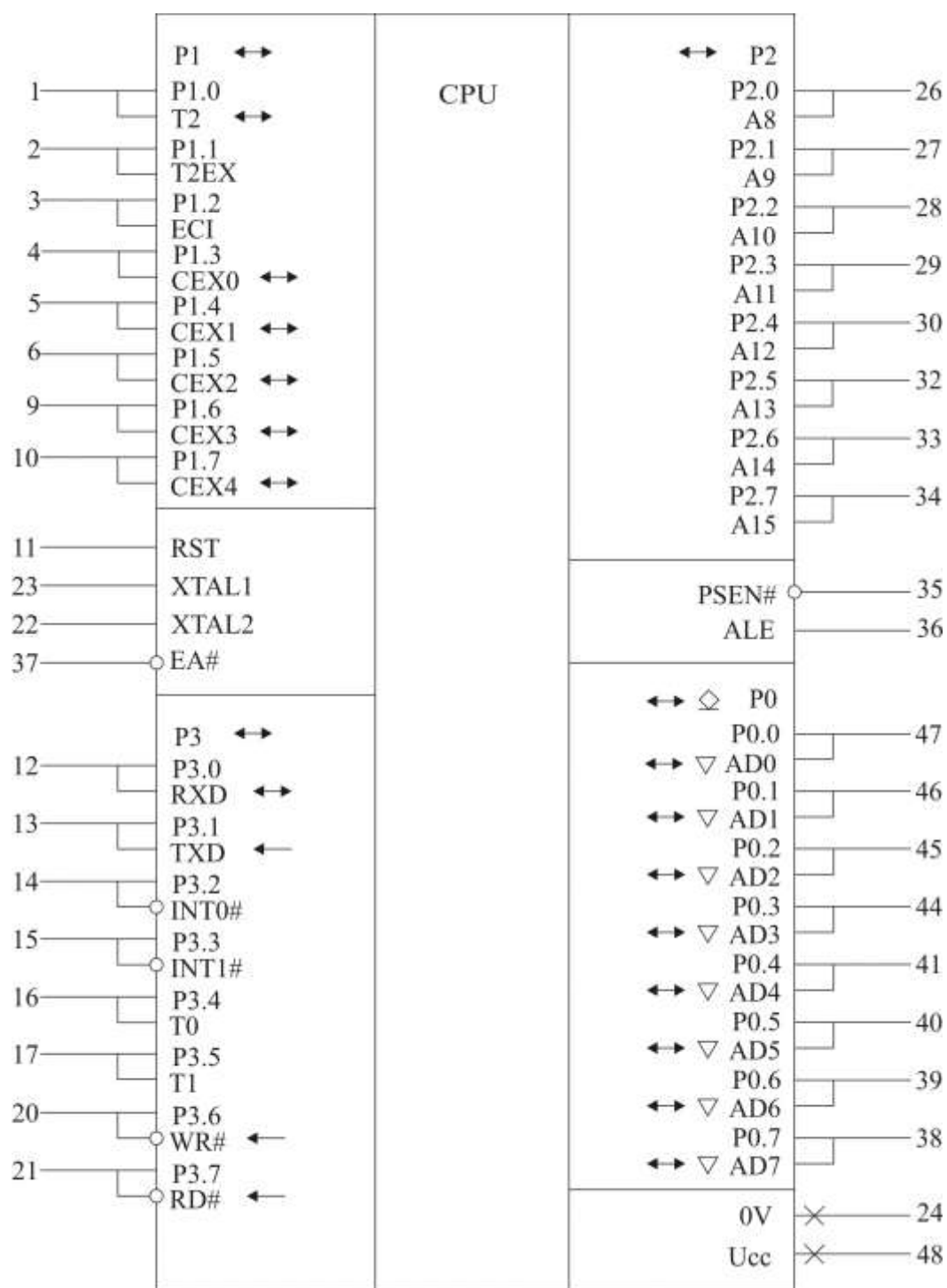


Рисунок 2.1 – Условное графическое обозначение микросхем 1830BE32У, 1830BE32АУ, 1830BE52У

Таблица 2.1 – Функциональное назначение выводов микросхем

Номер вывода	Обозначение		Функциональное назначение вывода	Тип вывода
	вывода	альтернативной функции вывода		
1	P1.0	T2	Вход/выход "порт 1, 0 разряд"	I/O
2	P1.1	T2EX	Вход/выход таймера/счетчика 2	I/O
3	P1.2	ECl	Вход/выход "порт 1, 1 разряд"	I/O
4-6, 9, 10	P1.3 - P1.7	CEX0 - CEX4	Вход триггера выборки-перезагрузки таймера/счетчика 2	I
			Вход/выход "порт 1, 2 разряд"	I/O
			Вход таймера массива программируемых счетчиков (PCA)	I
			Вход/выход "порт 1, 3-7 разряды"	I/O
			Вход-выход 0-4 каналов PCA	I/O
11	RST		Сигнал общего сброса	I
12	P3.0	RXD	Вход/выход "порт 3, 0 разряд"	I/O
13	P3.1	TXD	Вход-выход последовательных данных приемника	I/O
14	P3.2	INT0#	Вход/выход "порт 3, 1 разряд"	I/O
15	P3.3	INT1#	Выход последовательных данных передатчика	O
16	P3.4	T0	Вход/выход "порт 3, 2 разряд"	I/O
17	P3.5	T1	Вход внешнего прерывания 0	I
20	P3.6	WR#	Вход/выход "порт 3, 3 разряд"	I/O
21	P3.7	RD#	Вход внешнего прерывания 1	I
			Вход таймера/счетчика 0	I/O
			Вход/выход "порт 3, 4 разряд"	I/O
			Вход таймера/счетчика 1	I
			Вход/выход "порт 3, 5 разряд"	I/O
			Вход/выход "порт 3, 6 разряд"	I/O
			Выход стробящего сигнала при записи во внешнюю память данных	O
			Вход/выход "порт 3, 7 разряд"	I/O
			Выход стробящего сигнала при чтении из внешней памяти данных	O
22	XTAL2		Вывод для подключения кварцевого резонатора	-
23	XTAL1		Вывод для подключения кварцевого резонатора	-
24	0V		Вход тактового сигнала	I
26-30, 32-34	P2.0 - P2.7	A8 - A15	Общий вывод микросхемы	-
35	PSEN#		8-разрядный двунаправленный порт P2, 0 - 7 разряды	I/O
36	ALE		Выход адреса в режиме работы с внешней памятью	O
37	EA#		Выход "разрешение программной памяти"	O
47-44, 41-38	P0.0 - P0.7	AD0 – AD7	Выход "разрешение фиксации адреса"	O
48	U _{CC}		Вход "блокировка работы с внутренней памятью программ"	I
			8-разрядный двунаправленный порт P0, 0 - 7 разряды	I/O/2
			Шина адреса/данных при работе с внешней памятью	I/O/Z
			Вывод питания 5 В для ИС 1830BE32У, 1830BE52У и 3,3 В для ИС 1830BE32АУ	-
<p>Примечания</p> <p>1 Выводы 7, 8, 18, 19, 25, 31, 42, 43 не задействованы.</p> <p>2 В графе "Тип вывода": I - вход, O - выход, Z - третье состояние, 2 - режим открытого стока.</p>				

Таблица 2.2 – Электрические характеристики микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения		Буквенное обозначе- ние пара- метра	Норма параметра		Темпе- ратура среды, °C
			не менее	не более	
1 Выходное напряжение низкого уровня по выводам P1.0 – P1.7, P2.0 – P2.7, P3.0 – P3.7, В, I _{OL} = 1,6 мА	U _{CC} = 4,5 В	U _{OL1}	–	0,45	–60 ± 3 25 ± 10 85 ± 3
	U _{CC} = 3,0 В*				
2 Выходное напряжение низкого уровня по выводам ALE, PSEN#, P0.0/AD0 – P0.7/AD7, В, I _{OL} = 3,2 мА	U _{CC} = 4,5 В	U _{OL2}	–	0,45	
	U _{CC} = 3,0 В*				
3 Выходное напряжение высокого уровня по выводам P1.0 – P1.7, P2.0 – P2.7, P3.0 – P3.7, В, I _{OH} = –0,03 мА	U _{CC} = 4,5 В	U _{OH1}	U _{CC} – 0,7	–	
	U _{CC} = 3,0 В*				
4 Выходное напряжение высокого уровня по выводам ALE, PSEN#, P0.0/AD0 – P0.7/AD7, В, I _{OH} = –3,2 мА	U _{CC} = 4,5 В	U _{OH2}	U _{CC} – 0,7	–	
	U _{CC} = 3,0 В*				
5 Входной ток низкого уровня по вы- водам P1.0 – P1.7, P2.0 – P2.7, P3.0 – P3.7, мкА, U _I = 0,45 В	U _{CC} = 5,5 В	I _{IL}	–75	–	
	U _{CC} = 3,6 В*				
6 Токи утечки на входе по выводам P0.0/AD0 – P0.7/AD7, мкА, 0,45 В < U _I < U _{CC}	U _{CC} = 5,5 В	I _{LL} I _{LH}	–10	10	
	U _{CC} = 3,6 В*				
7 Динамический ток потребления в активном режиме, мА, f _{CI} = 12 МГц	U _{CC} = 5,5 В	I _{OCC1}	–	60	
	U _{CC} = 3,6 В*				30*
8 Динамический ток потребления в режиме пониженного потребления, мА, f _{CI} = 12 МГц	U _{CC} = 5,5 В	I _{OCC2}	–	21	
	U _{CC} = 3,6 В*				10*
9 Ток потребления в режиме хранения, мкА, U _{CC} = 2,0 В		I _{CCS}	–	100	
10 Функциональный контроль	U _{CC} = (4,5; 5,5) В, f _{CI} = (0,5; 12) МГц	ФК	–	–	
	U _{CC} = (3,0; 3,6) В*, f _{CI} = (0,5; 16) МГц*				
11 Время переключения сигнала на выходе ALE, нс, время нарастания время спада	U _{CC} = 4,5 В	t _r t _f	–	26	
	U _{CC} = 3,0 В*				
Примечание – Параметры I _{IL} , I _{LL} , I _{LH} , I _{CCS} при температуре минус 60 °C не измеряются, а гарантируются нормами при температуре (25 ± 10) °C.					
* Для ИС 1830BE32AU.					

Таблица 2.3 – Значения предельно допустимых электрических режимов эксплуатации в диапазоне рабочих температур от минус 60 до 85 °С

Наименование параметра режима, единица измерения		Буквенное обозначение параметра	Предельно допустимый режим		Предельный режим	
			не менее	не более	не менее	не более
1		2	3	4	5	6
1 Напряжение источника питания, В	1830BE32У, 1830BE52У	U_{CC}	4,5	5,5	-0,3	6,0
	1830BE32АУ		3,0	3,6	-0,3	6,0
2 Входное напряжение низкого уровня по выводам, за исключением вывода EA#, В		U_{IL1}	-0,5	$0,2U_{CC}-0,1$	-0,6	-
3 Входное напряжение низкого уровня по выводу EA#, В		U_{IL2}	0	$0,2U_{CC}-0,3$	-0,5	-
4 Входное напряжение высокого уровня по выводам, за исключением XTAL1, RST, EA#, В		U_{IH1}	$0,2U_{CC}+0,9$	$U_{CC}+0,5$	-	$U_{CC}+0,6$
5 Входное напряжение высокого уровня по выводам XTAL1, RST, EA#, В		U_{IH2}	$0,7U_{CC}$	$U_{CC}+0,5$	-	$U_{CC}+0,6$
6 Выходной ток высокого уровня, мА	P1.0 – P1.7, P2.0 – P2.7, P3.0 – P3.7	I_{OH}	-0,03	-	-0,05	-
	P0.0/AD0 – P0.7/AD7, ALE, PSEN#		-3,2	-	-5,0	-
7 Выходной ток низкого уровня, мА	P1.0 – P1.7, P2.0 – P2.7, P3.0 – P3.7	I_{OL}	-	1,6	-	2,5
	P0.0/AD0 – P0.7/AD7, ALE, PSEN#		-	3,2	-	5,0
8 Частота следования импульсов тактового сигнала, МГц	1830BE32У, 1830BE52У	f_{CI}	0,5	12	-	-
	1830BE32АУ		0,5	16	-	-
9 Емкость нагрузки по выводам P0.0/AD0 – P0.7/AD7, ALE, PSEN#, пФ		C_{L1}	-	100	-	200
10 Емкость нагрузки по выводам P1.0 – P1.7, P2.0 – P2.7, P3.0 – P3.7, пФ		C_{L2}	-	80	-	150
11 Длительность фронтов тактового сигнала, нс		t_{LH} t_{HL}	-	10	-	-
12 Длительность фронтов на других входах, нс		t_{LH} t_{HL}	-	20	-	-

В таблице 2.4 приведены сравнительные данные 8-разрядных микроконтроллеров разработки ФГУП "НИИЭТ".

Таблица 2.4 – Сравнительные характеристики 8-разрядных микроконтроллеров с архитектурой MCS-51

Наименование параметра, единица измерения	H1830BE31, H1830BE51 "Танго"	1882BE53У "Тема-3"	1830BE81Т, 1830BE91Т "Тостер"	1830BE32У, 1830BE32АУ, 1830BE52У "Танк-5"
1	2	3	4	5
Разрядность АЛУ, бит	8	8	8	8
Память программ (встроенная), бит	4К × 8 (маска)	12К × 8 (Flash)	2К × 8 (маска/Flash)	4К × 8* (маска)
Объем регистрового ОЗУ (встроенного), бит	128 × 8	256 × 8	128 × 8	256 × 8
Мажоритирование ОЗУ (тройное резервирование)	–	–	–	да
Память данных (встроенная, EEPROM), бит	–	2К × 8	–	–
Адресуемая память программ, Кбайт	64	64	2	64
Адресуемая память данных, Кбайт	64	64	–	64
Количество портов ввода-вывода	4 × 8	4 × 8	1 × 8, 1 × 7	4 × 8
Тактовая частота, МГц	12	24	24	12; 16**
Последовательный порт ввода-вывода (UART)	1	1	1	1
Последовательный периферийный интерфейс (SPI)	–	1	–	–
Число команд	111	111	107	111
Время выполнения команд, мкс:				
- сложение – регистр-регистр	1,0	0,5	0,5	1,0; 0,75**
- сложение – регистр-память	2,0	1,0	1,0	2,0; 1,5**
- умножение/деление	4,0	2,0	2,0	2,0; 1,5**
Количество источников прерываний	5	8	5	5
Число таймеров/счетчиков	2 × 16	3 × 16	2 × 16	3 × 16
Сторожевой таймер	–	1	–	–

Окончание таблицы 2.4

1	2	3	4	5
Программируемый массив счётчиков (РСА), каналов	–	–	–	5
Мощность потребления, мВт	132	275	100	145
Аналоговый компаратор	–	–	1	–
Стойкость к специальным видам воздействий	–	–	–	Да
Корпус	Н16.48-2В	Н16.48-2В	4153.20-5	Н16.48-2В
<p>* Для ИС 1830BE52У. ** Для ИС 1830BE32АУ.</p>				

3 Общая характеристика микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У

Микроконтроллеры 1830BE32У, 1830BE32АУ, 1830BE52У имеют следующие основные особенности:

- 8-разрядный процессор, оптимизированный для приложений управления;
- обширные возможности побитовой обработки;
- встроенная масочная память программ для 1830BE52У;
- двунаправленные и индивидуально адресуемые линии ввода-вывода;
- три 16-разрядных таймера/счётчика;
- полнодуплексный UART;
- разветвлённая структура прерываний (7 источников прерываний);
- встроенный тактовый генератор;
- режимы сохранения мощности: Idle и Power Down;
- программируемый массив счётчиков PCA (5 каналов).

Рассмотрим более подробно особенности архитектуры микроконтроллеров.

Отличительной особенностью микроконтроллеров является наличие блока PCA, предназначенного для выполнения различных операций счета и определения временных интервалов, в том числе при широтно-импульсной модуляции (таблица 2.4).

Для обеспечения экономии потребления энергии, микроконтроллеры имеют два программно управляемых режима работы с пониженной мощностью. В режиме Idle процессор выключен, в то время как оперативная память и встроенные периферийные устройства продолжают функционировать. В этом режиме потребление тока уменьшается приблизительно на 15 % от потребления полностью активного устройства. В режиме Power Down все устройства микроконтроллера выключены, од-

нако данные в оперативной памяти продолжают сохраняться. Кроме того, микроконтроллеры разработаны с применением статической логики, которая не требует непрерывной синхронизации. Поэтому частота тактового генератора может быть уменьшена до 0,5 МГц, или же он может быть остановлен в ожидании события, требующего обработки. Это также способствует снижению потребления по питанию.

Переход в режимы Idle и Power Down обеспечивается установкой соответствующих битов в регистре PCON пространства SFR.

В режиме Idle ($IDL = 1$) тактовый генератор продолжает работать, и обеспечивается работа периферийных устройств: таймеров, блока обработки прерываний и последовательного порта, при этом процессор микроконтроллера останавливается в ожидании поступления прерывания.

В режиме Power Down ($PD = 1$) останавливается тактовый генератор микроконтроллера, однако, содержимое встроенной памяти и регистров пространства SFR (регистров специальных функций) сохраняется. Выход из состояния Power Down возможен только при выполнении аппаратного сброса.

Организация памяти микроконтроллеров приведена на рисунке 3.1. Микроконтроллеры имеют отдельные адресные пространства для памяти программ и данных. Это позволяет обращаться к памяти данных 8-битными адресами, чем обеспечивается быстрота операций, выполняемых с памятью 8-разрядным процессором. Вместе с тем, с помощью регистра DPTR может быть сгенерирован 16-битный адрес данных. Таким образом, может быть адресовано до 64 Кбайт внешней памяти, для которой контроллер генерирует сигналы чтения и записи RD# и WR#. Для памяти программы обеспечивается только чтение. Непосредственно адресуется до 64 Кбайт памяти программ. Для чтения внешней памяти программ контроллер генерирует сигнал PSEN#. Внешняя память программ

и память данных могут быть объединены посредством объединения по логическому "И" сигналов контроллера RD# и PSEN#.

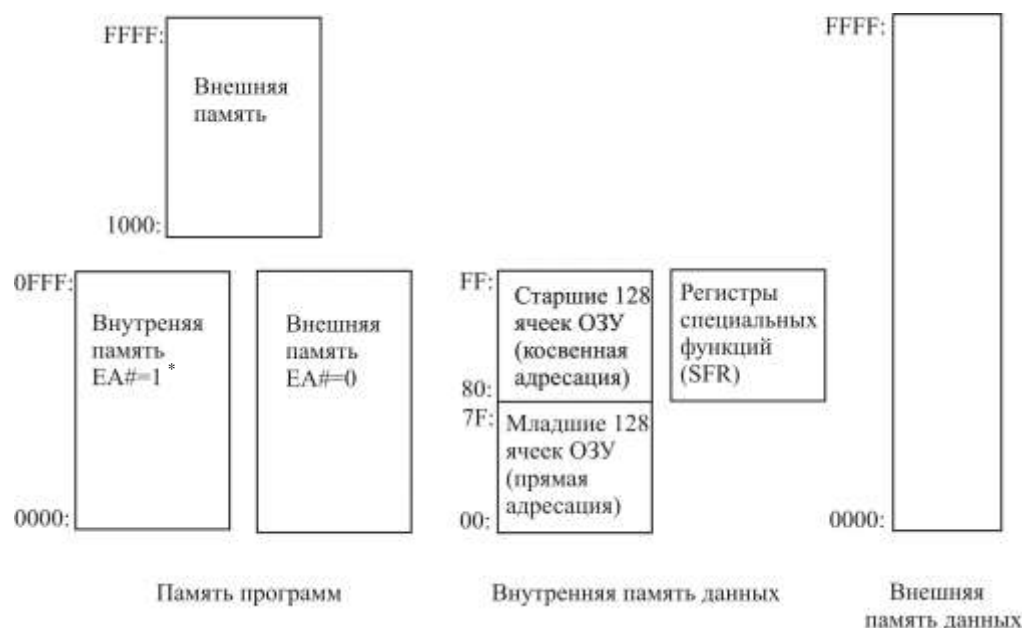
После выполнения процедуры сброса выполнение программы начинается с адреса 0000_H. С адреса 0003_H располагаются блоки обработки прерываний, занимающие по 8 байтов. Если процедура обработки прерывания занимает не более 8 байтов, она может располагаться в этом блоке. Процедуры обработки прерываний большего размера размещаются в других областях памяти программ, а управление передаётся им из блоков обработки прерываний командами безусловного перехода. Если прерывания в программе не используются, адреса, зарезервированные под блоки обработки прерываний, могут быть заняты кодом программы. Нижние адреса памяти программ могут адресовать как встроенную память программ, так и внешнюю память, в зависимости от того соединён ли вывод управления внешним доступом EA# с цепью питания или с общим проводом соответственно. Внешняя память программ может быть адресована 8-разрядным адресом с использованием порта ввода-вывода P0 для организации мультиплексированной шины адреса/данных, или 16-разрядным адресом с использованием портов ввода-вывода P0 и P2, причём последний передаёт старший байт адреса.

Внешняя память данных может иметь объём до 64 Кбайт. Адресуется она так же, как и память программ, 8 или 16-разрядными адресами. Пространство памяти данных разделено на 3 блока: нижние 128 байтов, верхние 128 байтов и область регистров специальных функций (Special Function Registers – SFR). Внутренняя память данных всегда адресуется одним байтом, что соответствует максимальному объёму памяти 256 байтов. Однако применение различных способов адресации позволяет использовать до 384 байтов внутренней памяти. Это связано с тем, что при прямой и косвенной адресации пространства свыше 7F_H адресуются в различные области памяти (верхние 128 байтов и область SFR).

В нижних 128 байтах внутренней памяти первые 32 байта заняты четырьмя банками по 8 регистров, адресуемых командами программы как R0, ..., R7. Выбор банка, в котором адресуются регистры, обеспечивается соответствующей установкой двух битов в слове состояния программы (PSW). Такая архитектура позволяет более эффективно использовать кодовое пространство, так как команды обращения к регистрам короче команд прямой адресации памяти. Следующие 16 байтов после банков регистров образуют блок битовой адресации. Система команд микроконтроллера включает широкий выбор команд битовой адресации, которые могут непосредственно адресовать 128 битов в этой области. Адресуемые биты имеют адреса $00_H \dots 7F_H$.

Все байты в нижних 128 байтах могут адресоваться прямым и косвенным методом адресации. К верхним 128 байтам, доступным в микроконтроллерах 1830BE32У, 1830BE32АУ, 1830BE52У с объёмом памяти 256 байтов, можно обращаться только с применением косвенной адресации. Пространство SFR (регистров специальных функций) включает порты ввода-вывода, регистры таймеров, регистры управления периферийными устройствами и т. д. Эта область может адресоваться только прямой адресацией. Структура пространства идентична структуре аналогичного пространства контроллеров семейства MCS-51™, однако имеются дополнительные регистры. 16 адресов в пространстве SFR могут адресоваться и побайтно и поразрядно.

Адреса разрядно адресуемых регистров заканчиваются тремя нулями. Адреса битов этих регистров имеют значения $80_H \dots FF_H$ и расположены эти регистры в области внутренней памяти данных (рисунок 3.1).



* Для ИС 1830BE52У.

Рисунок 3.1 – Организация памяти микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У

Система команд микроконтроллеров оптимизирована для 8-разрядных приложений управления, обеспечивая ряд быстрых способов адресации для доступа к внутренней оперативной памяти. Система команд обеспечивает обширную поддержку для однобитовых переменных как отдельного типа данных, позволяя выполнять прямое разрядное манипулирование в управлении и логических системах, которые требуют Булевой обработки.

Слово состояния программы PSW содержит биты состояния, которые отражают текущее состояние процессора, и размещается в пространстве SFR. Слово состояния программы содержит бит переноса CY, бит дополнительного переноса (для операций с BCD – двоично-десятичным кодированием) AC, биты выбора банка RS0 и RS1, флаг переполнения OV, биты контроля по четности P и два определяемых пользователем флажка.

Микроконтроллеры имеют команды прямой и косвенной адресации, регистровые команды и специальные команды для некоторых регистров. В последнем случае код команды непосредственно указывает на

регистр, с которым будет производиться операция. При прямой адресации операнд определён 8-разрядным полем адреса в команде.

Этим методом может быть адресована только внутренняя оперативная память (младшие 128 байт) и пространство SFR. При косвенной адресации в команде указан регистр, который содержит адрес операнда. Таким методом может адресоваться как внутренняя, так и внешняя оперативная память. В качестве регистра адреса для 8-разрядных адресов может быть или указатель вершины стека или регистры R0 или R1 выбранного банка.

Регистром адреса для 16-разрядных адресов может быть только 16-разрядный регистр указателя данных DPTR. К банкам регистров, которые содержат регистры R0, ..., R7, можно обращаться командами, коды операций которых включают 3-разрядную спецификацию регистра.

Команды, которые обращаются к регистрам этим способом, обеспечивают эффективное использование кода программы, так как при этом в команде отсутствует байт адреса. Банк, в котором текущей командой адресуется регистр, выбирается соответствующей установкой двух битов в слове состояния программы PSW.

Значение константы может следовать за кодом операции в памяти программ. 16-разрядный базовый регистр (DPTR или счётчик команд PC) указывает на начало текущей команды.

Машинный цикл микроконтроллеров состоит из 6 состояний S1, ..., S6, каждое из которых занимает два такта тактового генератора. Таким образом, длительность машинного цикла составляет 12 тактов тактового генератора или при тактовой частоте 12 МГц – 1,0 мкс. Команда программы может быть выполнена в течение одного или нескольких машинных циклов, например, команда MOVX занимает два машинных цикла.

Микроконтроллеры имеют 7 источников прерываний. Прерывание по каждому из источников может быть индивидуально разрешено или

запрещено путём установки или сброса соответствующих битов в регистре разрешения прерываний IE, расположенном в пространстве SFR.

В процессе выполнения программы состояние флагов прерываний считывается в состоянии 5-го машинного цикла и опрашивается в следующем цикле.

Для каждого из источников прерываний может быть запрограммирован один из двух уровней (от 0-го до 1-го) приоритета путём установки или сброса соответствующих битов в регистре приоритетов прерываний IP, расположенном в пространстве SFR. Низкоприоритетное прерывание может быть прервано высокоприоритетным прерыванием, но не другим низкоприоритетным прерыванием. Выполнение процедуры высокоприоритетного прерывания не может быть прервано никаким прерыванием.

Если одновременно поступило два запроса на прерывание с разными уровнями приоритета, то сначала выполняется процедура высокоприоритетного прерывания. При поступлении запросов на прерывание с одинаковым уровнем приоритета порядок выполнения процедур обработки прерывания определяется внутренней последовательностью опроса. В процессе обработки прерывания аппаратно сгенерированная процедура LCALL помещает содержимое счётчика команд PC в стек и загружает начальным адресом соответствующего блока обработки прерывания. Кроме счётчика команд автоматически в стеке не сохраняются никакие другие регистры.

За сохранение других необходимых регистров отвечает программист. В ряде случаев это позволяет сократить время обработки прерывания. В результате, много функций обработки прерываний, которые являются типичными в приложениях управления, – переключение вывода порта, перезагрузка таймера или чтение буфера последовательного порта, могут быть завершены скорее, чем это было бы возможно при другой архитектуре.

Многие приложения требуют больше уровней приоритетности прерываний, обеспечиваемых аппаратными средствами микроконтроллеров. В таком случае возможно применение простого программного кода, с помощью которого эмулируется следующий (четвертый) уровень приоритетности прерывания.

Последовательный порт микроконтроллеров – полнодуплексный, с буфером приёмника. Доступ к регистрам приёма и передачи осуществляется через регистр SBUF в пространстве SFR.

При выполнении записи в этот регистр загружается регистр передачи, чтение обеспечивает доступ регистру приёма. Для обеспечения стандартных скоростей обмена (1 200 – 19 200) бод необходимо тактировать микроконтроллер с частотой 11,059 МГц, при этом для получения необходимой скорости обмена используется один из таймеров.

При сбросе переинициализируются все регистры пространства SFR, однако содержимое внутренней памяти данных не изменяется. Аппаратный сброс запускает также тактовый генератор микроконтроллера. Сигнал сброса должен быть достаточно длительным для того, чтобы стабилизировалась работа тактового генератора (обычно не более 10 мс).

4 Описание устройства

4.1 Структурная организация

Микроконтроллеры имеют следующие аппаратные особенности:

- 8-разрядный процессор, оптимизированный для приложений управления;
- встроенная масочная память программ объемом 4 Кбайт для ИС 1830BE52У;
- мажоритированное ОЗУ 256×8 , с тройным резервированием, построенное по принципу выбора одного правильного значения из трех по большинству;
- блок ввода-вывода и исправления ошибок;
- двунаправленные и индивидуально адресуемые линии ввода-вывода;
- три 16-разрядных таймера/счётчика;
- полнодуплексный UART;
- блок прерываний с разветвленной структурой прерываний;
- встроенный тактовый генератор;
- встроенный блок PCA;
- режимы сохранения мощности: Idle и Power Down.

На рисунке 4.1 приведена структурная схема микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У. Для ИС 1830BE32У, 1830BE32АУ блок памяти программ ($4К \times 8$) бит отсутствует.

Функциональное назначение выводов микросхем приведено в таблице 2.1.

Принятые обозначения и сокращения.

0V – потенциал земли;

U_{CC} – напряжение питания;

XTAL1, XTAL2 – выводы для подключения кварцевого резонатора;

RST – сигнал общего сброса микроконтроллера;

PSEN# – сигнал разрешения внешней памяти программ; выдается только при обращении к внешнему ПЗУ;

ALE – сигнал разрешение фиксации адреса при работе с внешней памятью;

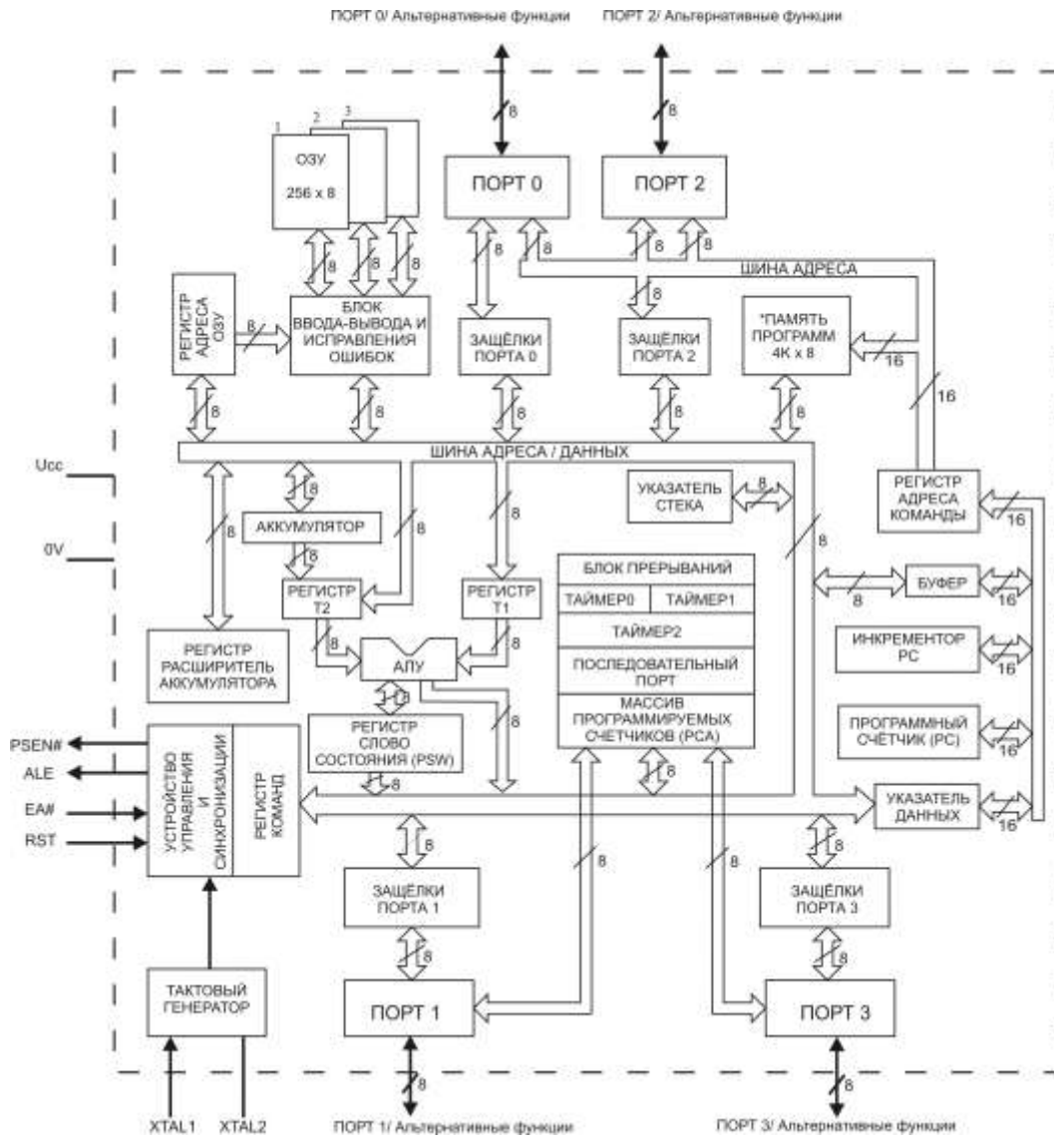
EA# – сигнал блокировки работы с внутренней памятью. Уровень 0 на этом входе заставляет микроконтроллер выполнять программу только внешнего ПЗУ, игнорируя внутреннее (если последнее имеется);

P0 – 8-битный двунаправленный порт ввода-вывода информации; при работе с внешними ОЗУ и ПЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных. При работе в режиме порта ввода-вывода необходимо внешнее подтягивание каждой линии порта к уровню логической "1";

P1 – 8-битный квазидвунаправленный порт ввода-вывода; каждый разряд порта может быть запрограммирован как на ввод, так и на вывод, независимо от состояния других разрядов; кроме того, выходы порта могут выполнять ряд альтернативных функций, которые описаны ниже;

P2 – 8-битный квазидвунаправленный порт, аналогичный P1; кроме того, выходы порта используются для выдачи адресной информации при обращении к внешней памяти программ или данных (если используется 16-битовая адресация последней). При обращении микроконтроллера к внешней памяти данных с использованием 8-разрядного адреса (команды MOVX@Ri) на выходы порта P2 выдается содержимое регистра специальных функций P2;

P3 – 8-битный квазидвунаправленный порт, аналогичный P1; кроме того, выходы порта могут выполнять ряд альтернативных функций, которые описаны ниже.



* Только для 1830VE52U.

Рисунок 4.1 – Структурная схема ИС 1830VE32U, 1830VE32AU, 1830VE52U

4.2 Организация ОЗУ, ПЗУ и регистров микроконтроллеров

Память программ и данных являются самостоятельными и независимыми друг от друга устройствами, адресуемыми различными командами и управляющими сигналами. Объем памяти программ, расположенной на кристалле микроконтроллера 1830VE52U, равен 4 Кбайт. При обращении к внешней памяти программ микроконтроллеры всегда используют 16-разрядный адрес, что обеспечивает им доступ к 64 Кбайт ПЗУ.

Микроконтроллер обращается к программной памяти при чтении кода операции и операндов (используя счетчик команд РС), а также при

выполнении команд переноса байта из памяти программ в аккумулятор. При выполнении команд переноса данных адресация ячейки памяти программ, из которой будут прочитаны данные, может осуществляться как с использованием счетчика РС, так и с использованием специального двухбайтового регистра-указателя данных DPTR.

Память данных, расположенная на кристалле, имеет объем 256 байт (регистровая память) и имеет повышенную защиту от одиночных сбоев. Она представляет собой блок мажоритированного ОЗУ и состоит из трех одинаковых ОЗУ объемом по 256 байт и блока ввода-вывода и исправления ошибок. Запись и чтение происходит одновременно для трех ОЗУ. При чтении схема исправления ошибок формирует данные по большинству. Для потребителя это одна память данных. Первые 32 байта организованы в четыре банка регистров общего назначения, обозначаемых, соответственно, банки 0, 1, 2, 3.

Каждый банк состоит из 8 регистров R0...R7. В любой момент времени программе доступен только один банк регистров, номер которого содержится в 3-м и 4-м битах слова состояния программы PSW (см. таблицу 4.2). Оставшееся адресное пространство может конфигурироваться разработчиком по своему усмотрению – в нем располагаются стек, системные и пользовательские области данных. Обращение к ячейкам памяти данных возможно двумя способами. Первый способ – прямая адресация ячейки памяти. В этом случае адрес ячейки является операндом соответствующей команды. Второй способ – косвенная адресация при помощи регистров R0 или R1 – перед выполнением соответствующей команды в один из этих регистров должен быть занесен адрес ячейки, к которой необходимо обратиться. К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR. Адреса, по которым расположены эти регистры, приведены в таблице 4.1. Отметим, что регистры занимают только часть 256-байтового адресного пространства.

Таблица 4.1 – Блок регистров специальных функций

Символ	Адрес	Наименование
1	2	3
* ACC	0E0 _H	Аккумулятор
* B	0F0 _H	Регистр-расширитель аккумулятора
* PSW	0D0 _H	Слово состояния программы
* P0	080 _H	Порт 0 (SRF P0)
* P1	090 _H	Порт 1 (SFR P1)
* P2	0A0 _H	Порт 2 (SFR P2)
* P3	0B0 _H	Порт 3 (SFR P3)
SP	081 _H	Регистр-указатель стека
DPH	083 _H	Старший байт регистра-указателя данных DPTR
DPL	082 _H	Младший байт регистра-указателя данных DPTR
TH0	08C _H	Старший байт Таймера 0
TL0	08A _H	Младший байт Таймера 0
TH1	08D _H	Старший байт Таймера 1
TL1	08B _H	Младший байт Таймера 1
TMOD	089 _H	Регистр режимов таймеров/счетчиков 0, 1
* TCON	088 _H	Регистр управления-статуса таймеров
T2CON	0C8 _H	Регистр управления Таймера 2
RCAP2H	0CB _H	Регистр перезагрузки Таймера 2, старший байт
RCAP2L	0CA _H	Регистр перезагрузки Таймера 2, младший байт
TH2	0CD _H	Старший байт Таймера 2
TL2	0CC _H	Младший байт Таймера 2
* IP	0B8 _H	Регистр приоритетов
* IE	0A8 _H	Регистр маски прерываний
SADDR	0A9 _H	Регистр спецфункций
SADEN	0B9 _H	Регистр маски спецфункций
PCON	087 _H	Регистр управления мощностью
* SCON	098 _H	Регистр управления приемопередатчиком
SBUF	099 _H	Буфер приемопередатчика

Окончание таблицы 4.1

1	2	3
CCON	0D8 _H	Регистр управления таймера/счетчика блока PCA
CMOD	0D9 _H	Регистр режима таймера/счетчика блока PCA
CCAPM0	0DA _H	Регистр режима работы модуля захвата/сравнения канала 0 блока PCA
CCAPM1	0DB _H	Регистр режима работы модуля захвата/сравнения канала 1 блока PCA
CCAPM2	0DC _H	Регистр режима работы модуля захвата/сравнения канала 2 блока PCA
CCAPM3	0DD _H	Регистр режима работы модуля захвата/сравнения канала 3 блока PCA
CCAPM4	0DE _H	Регистр режима работы модуля захвата/сравнения канала 4 блока PCA
CL	0E9 _H	Счетный регистр таймера-счетчика блока PCA, младший байт
CCAP0L	0EA _H	Регистр фиксации-сравнения канала 0 блока PCA, младший байт
CCAP1L	0EB _H	Регистр фиксации-сравнения канала 1 блока PCA, младший байт
CCAP2L	0EC _H	Регистр фиксации-сравнения канала 2 блока PCA, младший байт
CCAP3L	0ED _H	Регистр фиксации-сравнения канала 3 блока PCA, младший байт
CCAP4L	0EE _H	Регистр фиксации-сравнения канала 4 блока PCA, младший байт
CH	0F9 _H	Счетный регистр таймера-счетчика блока PCA, старший байт
CCAP0H	0FA _H	Регистр фиксации-сравнения канала 0 блока PCA, старший байт
CCAP1H	0FB _H	Регистр фиксации-сравнения канала 1 блока PCA, старший байт
CCAP2H	0FC _H	Регистр фиксации-сравнения канала 2 блока PCA, старший байт
CCAP3H	0FD _H	Регистр фиксации-сравнения канала 3 блока PCA, старший байт
CCAP4H	0FE _H	Регистр фиксации-сравнения канала 4 блока PCA, старший байт
Примечание – Регистры, имена которых отмечены знаком (*), допускают адресацию своих отдельных бит при выполнении команд из группы команд операций с битами.		

При обращении к внешней памяти данных микроконтроллера становятся доступными 64 Кбайт ОЗУ. Адресация упомянутых ячеек внешнего ОЗУ осуществляется методом косвенной адресации при помощи ре-

гистров R0 и R1 или при помощи вышеупомянутого двухбайтового регистра DPTR.

Аккумулятор является источником операнда и местом фиксации результата при выполнении ряда операций. Только с использованием аккумулятора могут быть выполнены операции сдвига, проверки на нуль и ряд других. Однако в отличие от большинства микропроцессоров, система команд подобна системе команд семейства 8051 (фирмы Intel) и содержит большое количество команд пересылок, логических операций и переходов, не использующих аккумулятор. В этом отношении упомянутая система команд достаточно гибка и универсальна.

При выполнении ряда команд в арифметико-логическом устройстве (АЛУ) формируются признаки операций – флаги, которые фиксируются в регистре PSW. Перечень флагов, их символические имена и условия их формирования приведены в таблице 4.2.

Таблица 4.2 – Формат регистра слова состояния программы PSW

Символ	Позиция	Имя и назначение																				
1	2	3																				
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратно в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе																				
-	PSW.1	Не используется																				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций																				
RS0 RS1	PSW.3 PSW.4	Биты выбора используемого банка регистров. Могут быть изменены программным путем. <table border="1"> <thead> <tr> <th>RS0</th> <th>RS1</th> <th>Банк</th> <th>Границы адресов ОЗУ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00_H - 07_H</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>08_H - 0F_H</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>10_H - 17_H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18_H - 1F_H</td> </tr> </tbody> </table>	RS0	RS1	Банк	Границы адресов ОЗУ	0	0	0	00 _H - 07 _H	1	0	1	08 _H - 0F _H	0	1	2	10 _H - 17 _H	1	1	3	18 _H - 1F _H
RS0	RS1	Банк	Границы адресов ОЗУ																			
0	0	0	00 _H - 07 _H																			
1	0	1	08 _H - 0F _H																			
0	1	2	10 _H - 17 _H																			
1	1	3	18 _H - 1F _H																			
F0	PSW.5	Флаг пользователя. Может быть установлен, сброшен или проверен программой пользователя																				
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заеме в бите 3 аккумулятора																				

Окончание таблицы 4.2

1	2	3
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается как аппаратно, так и программным путем

Регистр-указатель стека SP в микроконтроллерах рассматриваемого семейства 8-битный. Он может адресовать любую область внутренней памяти данных. Стек "растет вверх", т. е. перед выполнением команды PUSH или CALL содержимое SP инкрементируется, после чего производится запись информации в стек. Соответственно при извлечении информации из стека регистр SP декрементируется.

В процессе инициализации микроконтроллеров после сигнала сброса или при включении питающего напряжения в SP заносится код 07_H. Это означает, что первый элемент данных будет располагаться в ячейке памяти с адресом 08_H.

Регистр-указатель данных DPTR чаще всего используется для фиксации 16-битного адреса в операциях обращения к внешней памяти программ и данных. С точки зрения программиста он может выступать как в виде одного 16-битного регистра, так и в виде двух независимых регистров DPL и DPH.

Две регистровые пары с именами TH0, TL0 и TH1, TL1 представляют из себя регистры, обеспечивающие независимое функционирование двух программно-управляемых 16-битных таймеров/счетчиков. Режимы таймеров/счетчиков задаются с использованием регистра TMOD, а управление ими осуществляется при помощи регистра TCON для Таймеров 0, 1 и T2CON для Таймера 2. Для управления режимами энергопотребления микроконтроллеров используется регистр PCON. Регистры IP и IE управляют работой системы прерываний микроконтроллера, а регистры SBUF и SCON – работой приемопередатчика последовательного порта. Регистры SADDR и SADEN позволяют работать по последовательному порту при подключении к одному каналу нескольких микроконтролле-

ров. Описание таймеров/счетчиков, программируемого массива счетчиков PCA, системы прерываний и приемопередатчика последовательного порта будет приведено ниже.

4.3 Организация портов ввода-вывода

Все 4 порта микроконтроллеров – двунаправленные. Каждый из них содержит регистр-защелку (SFR P0...P3), выходную цепь и входной буфер. Оптимизированные схемы портовых буферов приведены на рисунке 4.2.

Выходные цепи P0 и P2 вместе с входным буфером P0 используются при обращении к внешней памяти. При этом на выходах P0 младший байт адреса внешней памяти мультиплексируется с вводимым-выводимым байтом. Выходы P2 содержат старший байт адреса внешней памяти, если адрес 16-разрядный. При использовании 8-разрядного адреса на выводах P2 присутствует информация из SFR P2.

Выводы порта P3, помимо обычного ввода информации и вывода ее из SFR, могут выполнять альтернативные функции. Описание этих функций приведено в таблице 4.3.

Как видно из рисунка 4.2, схема порта P1 микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У претерпела изменения по сравнению с портом P1 микроконтроллеров Н1830BE31, Н1830BE51. Это явилось следствием того, что выводы порта P1 стали выполнять альтернативные функции. Описание этих функций приведено в таблице 4.4.

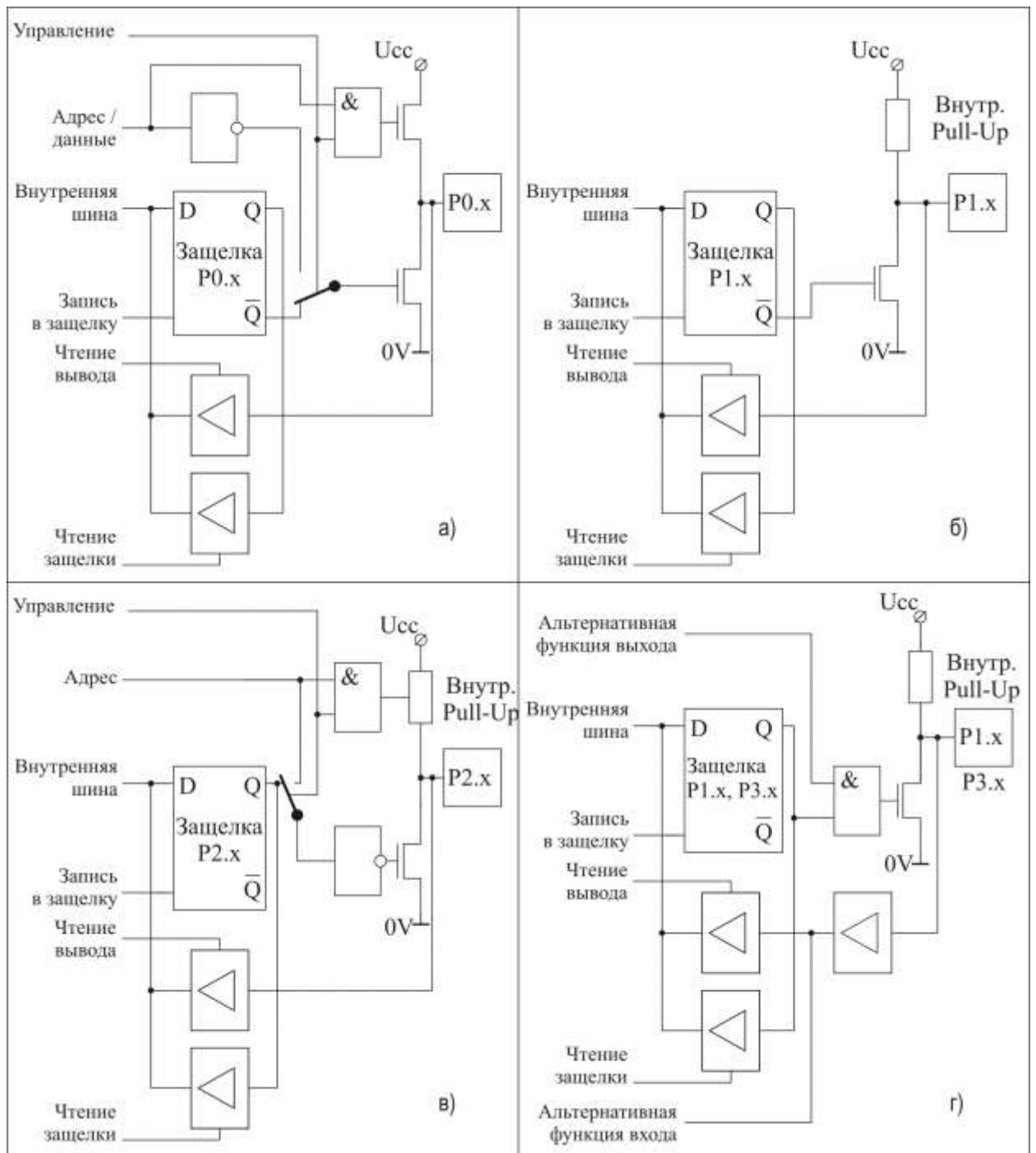


Рисунок 4.2 – Схемы портов микроконтроллеров:

а) P0.x для ИС 1830BE32У, 1830BE32АУ, 1830BE52У,

б) P1.x для ИС Н1830BE31, Н1830BE51,

в) P2.x для ИС 1830BE32У, 1830BE32АУ, 1830BE52У,

г) P1.x, P3.x для ИС 1830BE32У, 1830BE32АУ, 1830BE52У

Таблица 4.3 – Альтернативные функции порта P3

Вывод порта	Альтернативная функция
P3.0	RXD – вход последовательного порта
P3.1	TXD – выход последовательного порта
P3.2	INT0# – внешнее прерывание 0
P3.3	INT1# – внешнее прерывание 1
P3.4	T0 – вход таймера/счетчика 0
P3.5	T1 – вход таймера/счетчика 1
P3.6	WR# – строб записи во внешнюю память данных
P3.7	RD# – строб чтения из внешней памяти данных

Таблица 4.4 – Альтернативные функции порта P1

Вывод порта	Альтернативная функция
P1.0	T2 – вход-выход таймера/счетчика 2
P1.1	T2EX – вход триггера перезагрузки и выбора направления счета таймера/счетчика 2
P1.2	ECI – вход таймера программируемого массива счетчиков PCA
P1.3	CEX0 – вход-выход 0 канала фиксации-сравнения PCA
P1.4	CEX1 – вход-выход 1 канала фиксации-сравнения PCA
P1.5	CEX2 – вход-выход 2 канала фиксации-сравнения PCA
P1.6	CEX3 – вход-выход 3 канала фиксации-сравнения PCA
P1.7	CEX4 – вход-выход 4 канала фиксации-сравнения PCA

Альтернативные функции могут быть активированы только в случае, если в соответствующие биты SFR порта P3 предварительно занесены единицы.

Функциональные схемы регистров-защелок и буферов ввода-вывода каждого из четырех портов микроконтроллеров построены так:

Каждый из битов регистра-защелки SFR является по сути D-триггером, информация в который заносится с внутренней шины данных по сигналу "запись в SFR Pi" ($i = 0...3$) от центрального процессорного элемента (CPU). С неинвертированного выхода D-триггера информа-

ция может быть выведена на внутреннюю шину по сигналу "чтение SFR Pi" от CPU, а с вывода микросхемы ("из внешнего мира") – по сигналу "чтение выводов Pi". Одни команды активируют сигнал "чтение SFR Pi", другие – "чтение выводов Pi". Подробнее об этом будет сказано ниже при описании системы команд.

Поэтому выходные цепи P0 и P2 коммутируются сигналами CNTR, что обеспечивает вывод адресной информации и данных при обращении к внешней памяти. При обращении к внешней памяти в SFR P0 автоматически заносятся единицы во все биты. Информация в SFR P2 при этом остается неизменной. Следует отметить, что для того, чтобы выводы порта P3 выполняли альтернативные функции, необходимо в SFR P3 также занести единицы во все биты.

Выходы SFR регистров портов P1...P3 выполнены на МОП-транзисторах, имеющих внутреннюю нагрузку, в то время как выход SFR регистра P0 выполнен на транзисторах с открытым стоком. Каждая линия любого из портов может независимо использоваться как для ввода информации, так и для вывода (для линий портов P0 и P2 это справедливо тогда, когда они не используются для обращения к внешней памяти). Для использования любой линии портов P1...P3 в качестве входной необходимо в соответствующий разряд SFR занести единицу. При этом выходной полевой транзистор отключается. Внутренний нагрузочный резистор как бы "подтягивает" потенциал вывода к напряжению питания, в то время как внешняя нагрузка может сделать его нулевым.

Выводы порта P0 имеют иную структуру. Нагрузочный транзистор линии порта включен только тогда, когда порт выводит единицу при обращении к внешней памяти. В остальных случаях нагрузочный транзистор отключен. Таким образом, при работе в режиме обычного ввода-вывода информации (как, например, порт P1) выводы порта P0 представляют собой транзисторы с открытым стоком. Запись единицы в соответствующий бит SFR отключает и второй транзистор, что приводит к тому,

что вывод оказывается под плавающим потенциалом. Это позволяет использовать линии порта P0 как выводы с высокоимпедансным состоянием.

Поскольку выводы P1...P3 имеют внутреннюю нагрузку, то при переводе их в режим ввода они являются источниками тока для выхода микросхемы или транзистора, нагруженного на этот вывод. Поэтому порты P1...P3 получили название "квазидвунаправленные", в отличие от "истинно двунаправленного" порта P0, переводимого в режиме ввода информации в высокоимпедансное состояние.

Каждый из выводов портов P1...P3 может быть нагружен на 4 ТТЛ-входа. Они не требуют внешней нагрузки, но если в качестве выходов к этим выводам подключены ненагруженные выводы с открытым коллектором или стоком, время установления сигнала может затянуться. Выводы порта P0 при обращении к внешней памяти могут быть нагружены на 8 ТТЛ-входов. При использовании в качестве обычного порта каждый из выводов должен быть подключен к шине питания через резистор номиналом 10...20 кОм. При использовании P0 в качестве шины адреса/данных необходимость в этом отпадает.

Во время выполнения команд, изменяющих SFR портов, новые значения защелкиваются в триггеры SFR в самом последнем цикле выполнения команды. Но на выводах микроконтроллер данные появляются только в начале выполнения следующей команды.

Если это изменение представляет собой перепад из 0 в 1 линий портов P1...P3, то в начальный момент выполнения команды включается дополнительный нагрузочный транзистор в нагрузке транзистора выходного каскада порта. Дополнительная нагрузка в состоянии обеспечить ток переключения в 100 раз больший, чем номинальная, что резко повышает скорость переключения.

Следует отметить, что если на выводе микроконтроллера присутствовала единица, а внешняя нагрузка переключилась в ноль, то допол-

нительный нагрузочный транзистор может закрыться, и порт перейдет в высокоимпедансное состояние. Но при снятии причины, приведшей к этому переключению, основной р-канальный МОП-транзистор, быстродействие которого велико и сопоставимо с быстродействием п-канального, восстановит исходный потенциал.

Обращение к портам ввода-вывода возможно с использованием команд, оперирующих с байтом, отдельным битом и произвольной комбинацией бит. При этом в тех случаях, когда порт является одновременно операндом и местом назначения результата, автоматически реализуется специальный режим "чтение-модификация-запись". Этот режим обращения предполагает ввод сигналов не с внешних выводов порта, а из его регистра-защелки SFR, что позволяет исключить неправильное считывание ранее выведенной информации.

Подобный механизм обращения к портам реализован в следующих командах:

- ANL – логическое И, например, ANL P1, A;
- OPL – логическое ИЛИ, например, ORL P2, A;
- XRL – исключающее ИЛИ, например, XRL P3, A;
- JBC – переход, если в адресуемом бите 1, то последующее ее обнуление, например, JBC P1.1, LABEL;
- CPL – инверсия бита, например, CPL P3.3;
- INC – инкремент порта, например, INC P2;
- DEC – декремент порта, например, DEC P2;
- DJNC – декремент порта и переход, если его содержимое не равно нулю, например, DJNC P3, LABEL;
- MOV PX.Y, C – передача бита переноса в бит Y порта X;
- SET PX.Y – установка бита Y порта X;
- CLR PX.Y – сброс бита Y порта X.

Причиной, по которой оказалось необходимо осуществить команды типа "чтение-модификация-запись", является необходимость исклю-

чить неправильное считывание уровня сигнала на выводе микроконтроллера. Предположим, например, что линия "x" порта "y" соединяется с базой мощного транзистора и выходной сигнал на ней предназначен для его управления. Когда в данный бит записана единица, то транзистор включается. Если для проверки состояния исполнительного механизма, управляемого мощным транзистором, прикладной программе требуется прочитать состояние выходного сигнала в том же бите порта, то считывание сигнала с внешнего вывода порта, а не из D-триггера SFR порта, приведет к неправильному результату. Единичный сигнал на базе транзистора имеет относительно низкий уровень и будет интерпретирован CPU микроконтроллером как ноль. Команды типа "чтение-модификация-запись" реализуют считывание из регистра SFR, а не с внешнего вывода порта, что обеспечивает получение правильного значения единицы.

4.4 Работа с внешней памятью

Обращения к внешней памяти подразделяются на обращения к внешней памяти программ и обращения к внешней памяти данных. В первом случае для формирования сигнала, активирующего ПЗУ с программной памятью, используется сигнал PSEN#. Во втором случае используются сигналы RD# и WR#, активирующие ОЗУ с данными.

Если используется 16-битовый адрес, старшие 8 бит выводятся через порт P2, где они сохраняются в течение всего цикла обращения к внешней памяти. Следует отметить, что выходные буферы порта P2 имеют внутреннюю нагрузку, несколько отличную от P1 и P3, благодаря чему в SFR P2 при выводе адресной информации вовсе не обязательно защелкивать все единицы. Следует добавить, что при выводе адресной информации, информация из SFR P2 хоть и не присутствует на выводах микроконтроллера, но и не теряется, восстанавливаясь на выводах после окончания обращений к внешней памяти (если в процессе этих обращений SFR P2 не был модифицирован).

Если при обращении к внешней памяти данных используется 8-битный адрес, то на выводах порта остается та же информация, которая там была до начала обращения к внешней памяти. Это позволяет организовать постраничную адресацию внешней памяти данных.

Как уже отмечалось выше, на выводах порта P0 младший байт адреса мультиплексируется с данными. Сигналы адреса/данных задействуют оба полевых транзистора выходного буфера P0. Следовательно, в этом случае выводы P0 уже не являются выводами с открытым стоком и не требуют внешних нагрузочных элементов. Сигнал ALE используется для фиксации младшего байта адреса во внешнем регистре-защелке. Адресная информация достоверна в момент окончания сигнала ALE. Выводимый в цикле записи байт заносится в P0 после активации сигнала WR# и остается неизменным до окончания этого сигнала. В цикле чтения данные на выводах P0 для достоверного считывания должны быть установившимися к моменту окончания сигнала RD#.

Во время обращения к внешней памяти CPU записывает $0FF_H$ в SFR P0, разрушая таким образом хранимую там информацию. Следовательно, использовать для записи порт P0 при работе с внешней памятью надо с известной долей осторожности.

Когда CPU работает с внешней памятью программ, все линии P2 используются для вывода старшего байта адреса и не могут быть использованы для обычного ввода-вывода информации. При этом, как отмечалось выше, в SFR P2 может быть занесена любая информация, так как адресная информация, выводимая через P2, не зависит от состояния его SFR.

При обращении к внешней памяти данных используются команды, выполняющиеся за два машинных цикла. В первом машинном цикле сигнал ALE возникает в момент S1P2 и в момент S4P2. По спаду этих сигналов во внешний регистр-защелку должна быть занесена адресная информация, выводимая по линиям порта 0 (DPL OR RI OUT – вывод

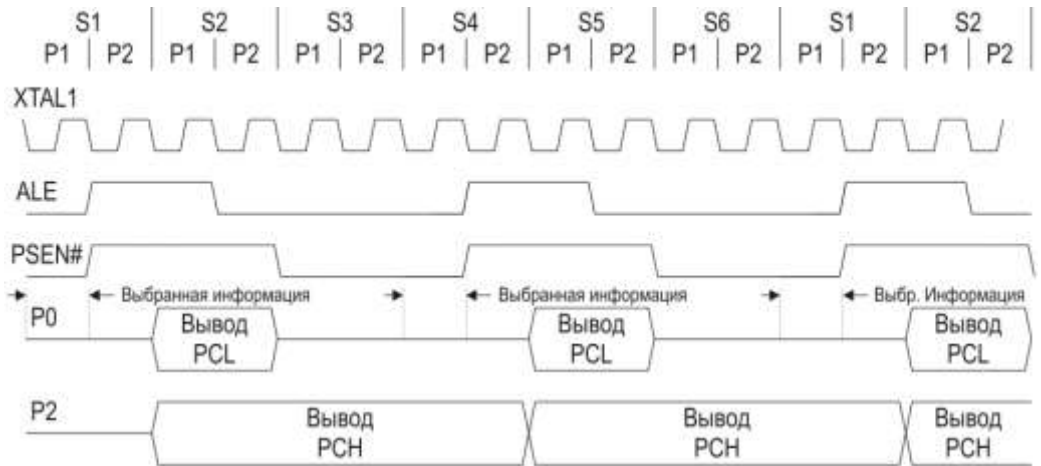
младшего байта из регистра DPTR либо байта из регистра R0 или R1). При обращении к внешней памяти с использованием регистра DPTR на линиях порта 2 при этом появляется старший байт из регистра DPTR (DPH). Если происходит обращение к внешней памяти данных с использованием регистров R0 или R1, то на линиях порта появляется информация его SFR (P2 SFR OUT).

При вводе информации через порты P0...P3 данные должны быть установившимися к моменту S5P1 и сохраняться неизменными до момента S5P2. При выводе информации из портов новые данные (NEW DATA) появляются на выводах микроконтроллера в момент S1P1 цикла, следующего за циклом, в котором осуществлялась команда вывода.

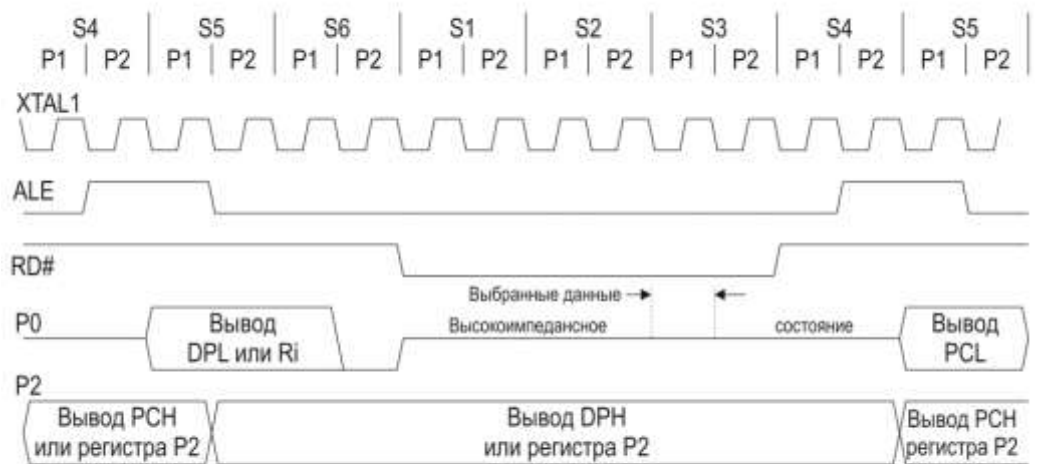
При работе микроконтроллеров используются в виде комбинаций следующие три типа магистральных циклов:

- ввод кода команд из внешней памяти программ без обращения к внешней памяти данных с длительностью 6 тактов или 12 периодов XTAL1 (рисунок 4.3а);
- чтение из внешней памяти данных с длительностью 12 тактов или 24 периодов XTAL1 (рисунок 4.3б);
- запись во внешнюю память данных с длительностью, как при чтении (рисунок 4.3в).

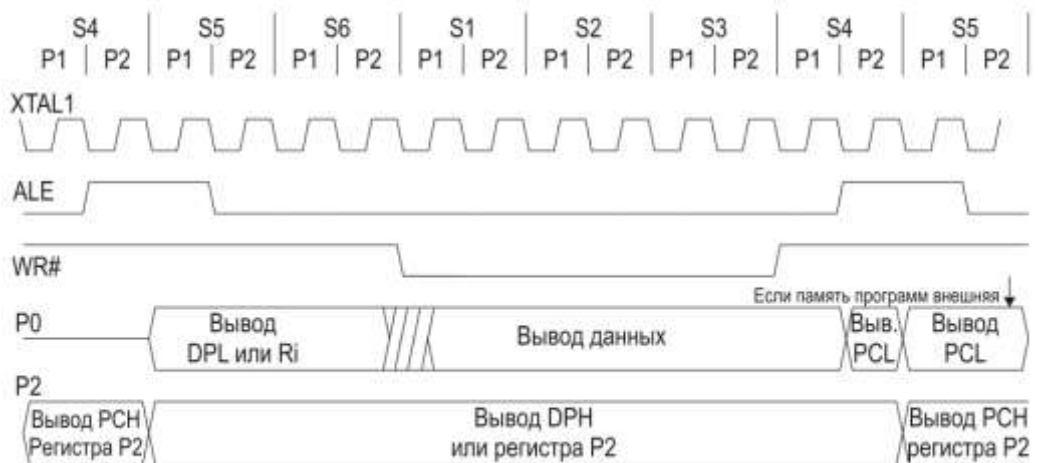
Особенностью архитектуры микроконтроллеров является то, что циклы всех команд формируются из циклов магистрали и имеют строго определенную длительность.



а) Выборка из внешней памяти программ



б) Чтение из внешней памяти данных



в) Запись во внешнюю память данных

Рисунок 4.3 – Временные диаграммы сигналов микросхем при работе с внешней памятью

4.5 Внутренняя память программ ИС 1830BE52У

Адресное пространство внутренней памяти программ приведено на рисунке 4.4.

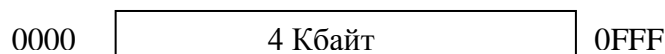


Рисунок 4.4 – Блок памяти программ

Память программ в масочном варианте состоит из двух массивов, содержащих ячейки, состоящие из информационных и адресных транзисторов и собранные в байтовые блоки. Такая конструкция массива памяти выбрана для уменьшения временных задержек, а также для уменьшения геометрических размеров. Выборка информационного байта производится по строкам и столбцам, разряды адреса $A_0...A_7$ в дешифраторе строк определяют соответствующую строку, а разряды $A_8...A_{11}$ в дешифраторе столбцов определяют соответствующий столбец. В результате информация с выбранного байта поступает на усилитель считывания и далее на основную внутреннюю шину данных. Требуемая информация записывается в массив памяти на стадии изготовления, при этом используется дополнительный сменный шаблон по карте заказа потребителя. Функциональная схема блока программной памяти приведена на рисунке 4.5.

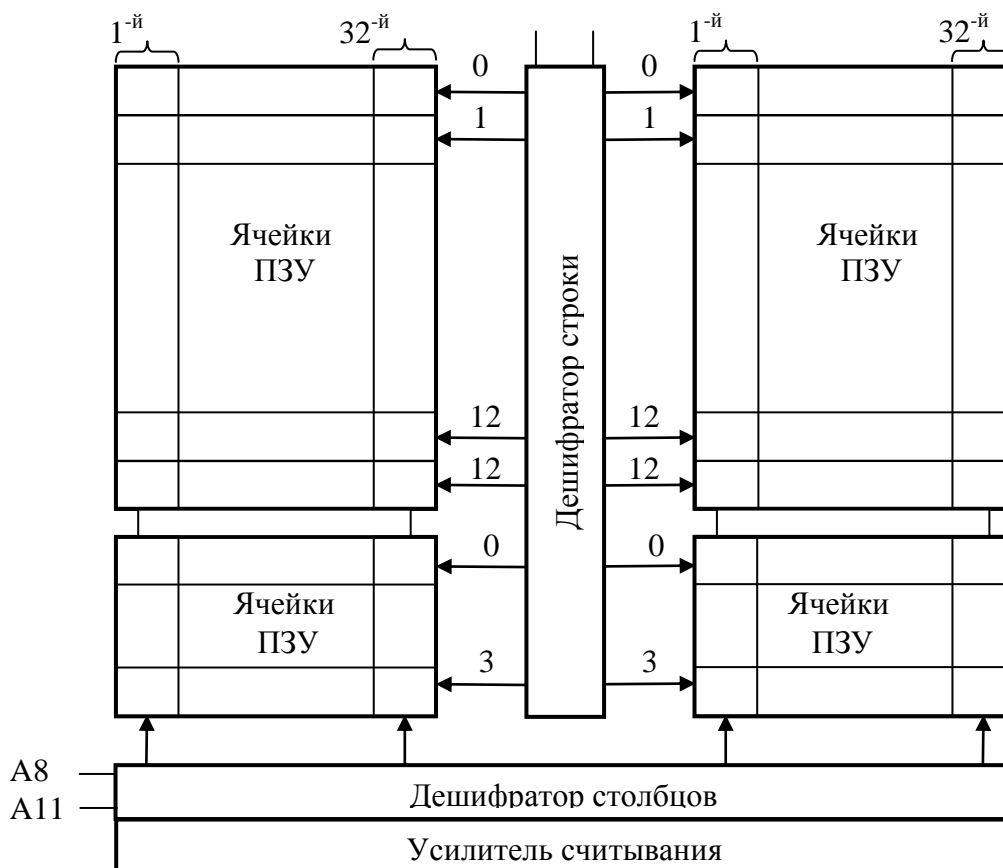


Рисунок 4.5 – Функциональная схема блока программной памяти ИС 1830BE52У

4.6 Устройство управления и синхронизации

К выводам XTAL1 и XTAL2 микроконтроллера подключается кварцевый резонатор, или к выводу XTAL1 подключается внешний тактовый сигнал. Об устройстве встроенного тактового генератора микроконтроллера будет рассказано ниже при описании режимов энергопотребления (см. подраздел 4.16). Пока отметим только то, что вырабатываемые им сигналы синхронизируют всю работу микроконтроллера.

Устройство управления микроконтроллеров формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора или шести состояниям управляющего устройства (S1...S6). Каждое состояние содержит две фазы сигналов резонатора (P1 и P2). В фазе P1, как правило, выполняется операция в АЛУ, а в фазе P2 выполняется межрегистровая передача. Весь машинный цикл состоит из 12 фаз, начиная от S1P1 и кончая S6P2. Так осуществляется работа управляющего устройст-

ва при выполнении команд различной степени сложности, от однобайтовых, выполняющихся за один машинный цикл, до двухбайтовых, работающих с внешней памятью.

Количество машинных циклов и тактов, необходимое микроконтроллеру для выполнения каждой команды, будет дано при описании системы команд. Сейчас отметим только, что микроконтроллер оперирует с 13 различными типами команд, имеющими длину от 1 до 3 байт и выполняющимися за один или два машинных цикла (за исключением двух однобайтовых команд умножения и деления, выполняющихся за четыре цикла). Таким образом, при работе на частоте 12 МГц подавляющее большинство команд выполняется микроконтроллером за 1,0...2 мкс.

4.7 Таймеры/счетчики

Два программируемых 16-битных таймера/счетчика (Т/С0 и Т/С1) могут быть использованы как в качестве таймеров, так в качестве счетчиков внешних событий. Следует отметить, что в микроконтроллере есть еще таймер/счетчик 2 (см. раздел 8). При работе в качестве таймера содержимое соответствующего таймера/счетчика инкрементируется в каждом машинном цикле, т. е. через каждые 12 периодов резонатора. При работе в качестве счетчика содержимое таймера/счетчика инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (Т0, Т1) вывод микросхемы. Опрос значения внешнего входного сигнала выполняется в момент времени S5P2 каждого машинного цикла. Содержимое счетчика будет увеличено на 1 в том случае, если в предыдущем цикле был считан входной сигнал высокого уровня (1), а в следующем – сигнал низкого уровня (0). Новое (инкрементированное) значение счетчика будет сформировано в момент S3P1 в цикле, следующем за тем, в котором был обнаружен переход сигнала из 1 в 0. Так как на распознавание перехода требуется два машинных цикла, то максимальная частота подсчета входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ог-

раничений сверху нет. Для гарантированного прочтения входного считываемого сигнала он должен удерживать значение 1 как минимум в течение одного машинного цикла ядра микроконтроллера. Описание регистров управления TMOD и TCON приведены в таблицах 4.5, 4.6.

Таблица 4.5 – Регистр TMOD режима работы таймеров/счетчиков 0, 1

Символ	Позиция	Имя и назначение
GATE	TMOD.7 для T/C1	Управление блокировкой. Если бит установлен, то таймер/счетчик "х" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то таймер/счетчик разрешается, как только бит управления "TRx" устанавливается
	TMOD.3 для T/C0	
С/Т	TMOD.6 для T/C1	Бит выбора режима таймера или счетчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если бит установлен, то работает счетчик от внешних сигналов на входе "Tx"
	и TMOD.2 для T/C0	
M1	TMOD.5 для T/C1	Режим работы (см. примечание)
	TMOD.1 для T/C0	
M0	TMOD.4 для T/C1	
	и TMOD.0 для T/C0	
Примечание		
M1	M0	Режим работы
0	0	Таймер BE48. "TLx" работает как 5-битный предделитель
0	1	16-битный таймер/счетчик. "THx" и "TLx" включены последовательно
1	0	8-битный автоперезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению
1	1	Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TL0 работает как 8-битный таймер/счетчик, и его режим определяется управляющими битами Таймера 0. TH0 работает только как 8-битный таймер, и его режим определяется управляющими битами Таймера 1

Таблица 4.6 – Регистр TCON управления/статуса таймеров/счетчиков 0, 1

Символ	Позиция	Имя и назначение
TF1	TCON.7	Флаг переполнения Таймера 1. Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно
TR1	TCON.6	Бит управления Таймера 1. Устанавливается/сбрасывается программой для пуска/останова
TF0	TCON.5	Флаг переполнения Таймера 0. Устанавливается аппаратно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	Бит управления Таймера 0. Устанавливается или сбрасывается программой для пуска/останова таймера/счетчика
IE1	TCON.3	Флаг фронта прерывания 1. Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1#. Сбрасывается при обслуживании прерывания
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается или сбрасывается программно для спецификации запроса INT1# (срез/низкий уровень)
IE0	TCON.1	Флаг фронта прерывания 0. Устанавливается по срезу сигнала INT0#. Сбрасывается при обслуживании прерывания
IT1	TCON.0	Бит управления типом прерывания 0. Устанавливается/сбрасывается программно для спецификации запроса INT0# (срез/низкий уровень)

Режим 0. Перевод T/C0 или T/C1 в режим 0 делает его похожим на таймер, на вход которого подключен 5-битный предделитель частоты на 32. Работу таймера/счетчика в режиме 0 на примере T/C1 иллюстрирует рисунок 4.6а. В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния "все единицы" в состояние "все нули" устанавливается флаг прерывания от таймера TF1. Входной синхросигнал Таймера 1 разрешен (поступает на вход T/C1), когда управляющий бит TR1 установлен в 1 и либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания INT1# поступает уровень 1.

Режим 1. Работа любого таймера/счетчика (0 или 1) в режиме 1 такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит (рисунок 4.6б).

Режим 2. В режиме 2 работа организована таким образом, что переполнение (переход из состояния "все единицы" в состояние "все нули")

8-битного счетчика TL1 приводит не только к установке флага TF1 (рисунок 4.6в), но и автоматически перезагружает в TL1 содержимое старшего байта (ТН1) таймерного регистра, которое предварительно было задано программным путем. Перегрузка оставляет содержимое ТН1 неизменным. В режиме 2 Т/С0 и Т/С1 также работают совершенно одинаково.

Режим 3. В режиме 3 Т/С0 и Т/С1 работают по-разному. Т/С1 сохраняет неизменным свое текущее содержимое. Иными словами, эффект такой же, как и при сбросе управляющего бита TR1 в нуль.

Работу Т/С0 в режиме 3 иллюстрирует рисунок 4.6г. В режиме 3 TL0 и ТН0 функционируют как два независимых 8-битных счетчика. Работу TL0 определяют управляющие биты Т/С0 (С/Т, GATE, TR0), входной сигнал INT0# и флаг переполнения TF0. Работу ТН0, который может выполнять только функции таймера (подсчет машинных циклов микроконтроллера), определяет управляющий бит TR1. При этом ТН0 использует флаг переполнения TF1.

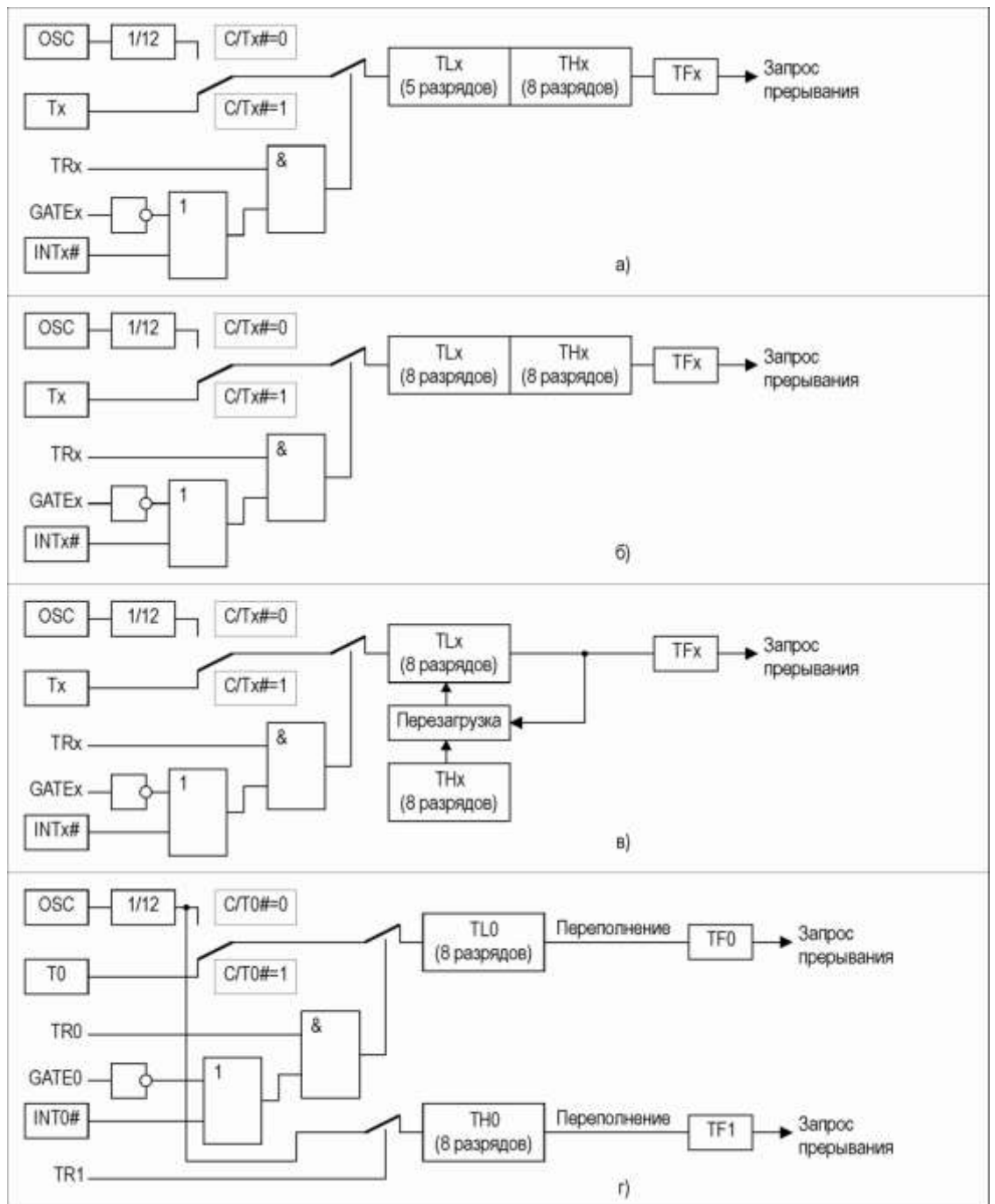


Рисунок 4.6 – Работа счетчиков/таймеров 0, 1: а) в режиме 0; б) в режиме 1; в) в режиме 2; г) в режиме 3

4.8 Система прерываний

Микроконтроллеры имеют 7 уровней прерываний:

- внешнее прерывание (INT0#);
- внешнее прерывание (INT1#);
- прерывание от Таймера 0 (INT T0);
- прерывание от Таймера 1 (INT T1);

- прерывание от Таймера 2 (INT T2);
- прерывание от последовательного порта (INT S);
- прерывание от блока PCA (INT PCA).

Внешние прерывания INT0# и INT1# могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах микроконтроллера в зависимости от значений управляющих бит IT0 и IT1 в регистре TCON. От внешних прерываний устанавливаются флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей программы обслуживания прерывания. Сброс этих флагов выполняется аппаратно только в том случае, если прерывание было вызвано по переходу (фронту) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага IE должна управлять соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

Флаги запросов прерывания от таймеров TF0 и TF1 сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются блоком управления приемопередатчика аппаратно, но сбрасываться должны программным путем.

Прерывания могут быть вызваны или отменены программой, так как все перечисленные флаги программно доступны и могут быть установлены/сброшены программой с тем же результатом, как если бы они были установлены/сброшены аппаратными средствами.

В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний и уровнями приоритета. Форматы этих регистров, имеющих символические имена IE и IP, описаны в таблицах 4.7 и 4.8 соответственно.

Таблица 4.7 – Регистр масок прерывания IE

Символ	Позиция	Имя и назначение
1	2	3
EA	IE.7	Снятие блокировки прерывания. Сбрасывается программно (EA = 0) для запрета всех прерываний, независимо от состояний IE.4 – IE.0
EC	IE.6	Запрещение запроса от блока PCA при EC = 0
ET2	IE.5	Запрещение запроса от Таймера 2 при ET2 = 0
ES	IE.4	Бит разрешения прерывания от приемопередатчика. Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI. Запрет при ES = 0
ET1	IE.3	Бит разрешения прерывания от Таймера 1. Установка/сброс программой для разрешения/запрета прерываний от Таймера 1. Запрет при ET1 = 0
EX1	IE.2	Бит разрешения внешнего прерывания 1 по входу INT1#. Установка/сброс программой для разрешения/запрета прерывания 1. Запрет при EX1 = 0
ET0	IE.1	Бит разрешения прерывания от Таймера 0. Установка/сброс программой для разрешения/запрета прерываний от Таймера 0. Запрет при ET0 = 0
EX0	IE.0	Бит разрешения внешнего прерывания 0 по входу INT0#. Установка/сброс программой для разрешения/запрета прерывания 0. Запрет при EX0 = 0

Таблица 4.8 – Регистр приоритетов прерываний IP

Символ	Позиция	Имя и назначение
–	IP.7	Не используется
PPC	IP.6	Бит приоритета блока PCA
PT2	IP.5	Бит приоритета Таймера 2
PS	IP.4	Бит приоритета приемопередатчика. Установка/сброс программой для присваивания прерыванию от приемопередатчика высшего/низшего приоритета
PT1	IP.3	Бит приоритета Таймера 1. Установка/сброс программой для присваивания прерыванию от Таймера 1 высшего/низшего приоритета
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания приоритета внешнему прерыванию INT1#
PT0	IP.1	Бит приоритета Таймера 0. Установка/сброс программой для присваивания приоритета прерыванию от Таймера 0

Окончание таблицы 4.8

1	2	3
PX0	IP.0	Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания приоритета внешнему прерыванию INT0#

Следует отметить, что если флаг прерывания был установлен, но по одному из перечисленных выше условий не получил обслуживания и к моменту окончания блокировки уже был сброшен, то запрос прерывания теряется и нигде не запоминается. Флаги прерывания опрашиваются в момент S5P2 каждого машинного цикла. Ранжирование прерываний по уровню приоритета выполняется в течение следующего машинного цикла. Система прерываний оформляет аппаратно вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована условиями, приведенными в таблице 4.9.

Таблица 4.9 – Условия блокировки программы обслуживания прерывания

Номер по порядку	Условия
1	В текущий момент обслуживается запрос прерывания равного или более высокого уровня приоритета
2	Текущий машинный цикл – не последний в цикле выполнения команды
3	Выполняется команда RETI или любая другая команда, связанная с обращением к регистрам IP и IE

Уровень приоритета конкретного источника определяется значением комбинации битов IP.x (где $x = 0, 1$). При одновременном появлении нескольких запросов прерывания одного уровня очередность обслуживания определяется с помощью внутренней процедуры поллинга (последовательного опроса), который производится в порядке фиксиро-

ванного старшинства источников внутри одного уровня приоритета. Этот порядок представлен в таблице 4.10.

Таблица 4.10 – Приоритеты прерываний при поллинге

Источник	Приоритет внутри уровня
1	2
INT0#	Высший
Таймер 0	-
INT1#	-
Таймер 1	-
Блок PCA	-
Последовательный порт	-
Таймер 2, флаг TF2	-
Таймер 2, флаг T2EX	Низший

По коду LCALL, сформированному аппаратно, система прерывания помещает в стек только содержимое счетчика команд (PC) и загружает в счетчик команд адрес вектора соответствующей подпрограммы обслуживания. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. Подпрограмма обслуживания в случае необходимости должна начинаться командами записи в стек (PUSH) слова состояния программы (PSW), аккумулятора, расширителя, указателя данных и т. д. и должна заканчиваться командами восстановления из стека (POP). Подпрограммы обслуживания прерывания должны завершаться командой RETI, по которой в счетчик команд перезагружается из стека сохраненный адрес возврата в основную программу. Команда RET также возвращает управление прерванной основной программе, но при этом не снимает блокировку прерываний, что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания.

Особенности запросов внешних прерываний

По входам INT0#, INT1# могут приниматься сигналы запросов, активными значениями которых является либо низкий уровень входного сигнала, либо перепад из высокого уровня сигнала в низкий. Тип активного значения определяется битами IT0, IT1 регистра TCON. При ITx = 0 запрос фиксируется по низкому уровню сигнала на соответствующем входе INTx#, а при ITx = 1 – по перепаду. Если запрос прерывания формируется перепадом сигнала на входе INTx#, высокий и низкий уровни сигнала должны удерживаться не менее одного машинного цикла каждый. При запросе в виде низкого уровня сигнала активное значение должно удерживаться на входе до начала обработки данного запроса. Далее сигнал должен стать пассивным до завершения процедуры обслуживания. Временная диаграмма запроса и обработки прерывания приведена на рисунке 4.7.



Рисунок 4.7 – Запрос и обработка прерывания

4.9 Последовательный интерфейс

Через универсальный асинхронный приемопередатчик осуществляется прием и передача информации, представленной последовательным кодом (младшими битами вперед), в полном дуплексном режиме обмена. В состав приемопередатчика, называемого часто последовательным портом, входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (SBUF) приемопередатчика. Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буфер-

ного регистра приемника позволяет совмещать операцию чтения ранее принятого байта с приемом очередного байта. Но если к моменту окончания приема байта предыдущий байт не был считан из SBUF, то он будет потерян.

Последовательный порт микроконтроллера может работать в четырех различных режимах, приведенных на рисунках 4.8 – 4.10.

Режим 0. В этом режиме информация и передается и принимается через внешний вывод входа приемника (RXD). Принимаются или передаются 8 бит данных. Через внешний вывод выхода передатчика (TXD) выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна 1/12 частоты резонатора.

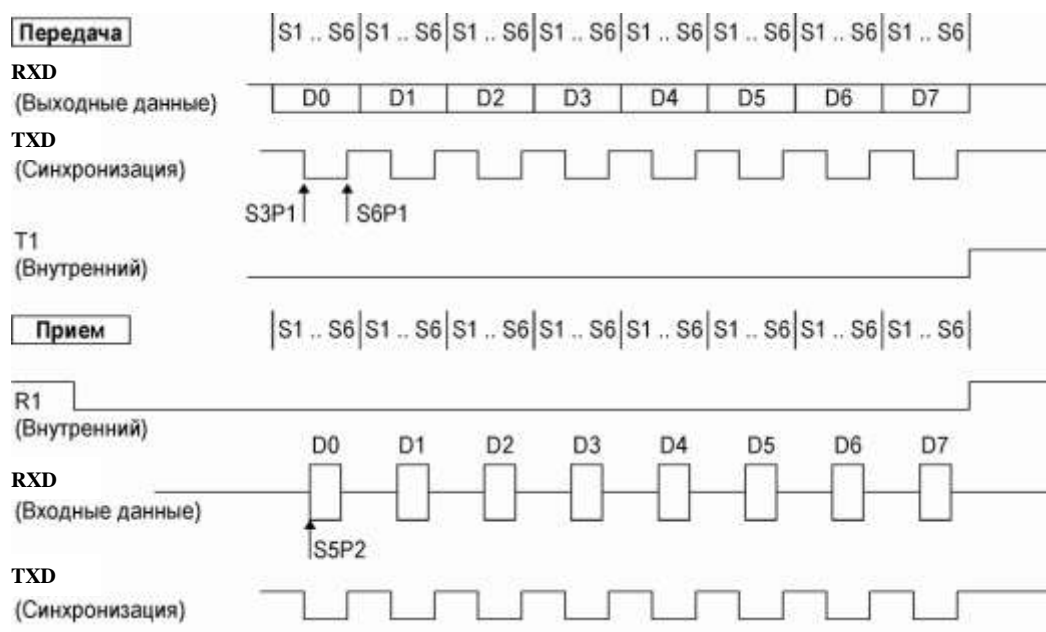


Рисунок 4.8 – Работа последовательного порта в режиме 0

Режим 1. В этом режиме передаются через TXD или принимаются из RXD 10 бит информации: старт-бит (0), 8 бит данных и стоп-бит (1). При приеме информации в бит RB8 регистра управления-статуса приемопередатчика SCON заносится стоп-бит. Скорость приема/передачи – величина переменная и задается Таймером 1 или Таймером 2.

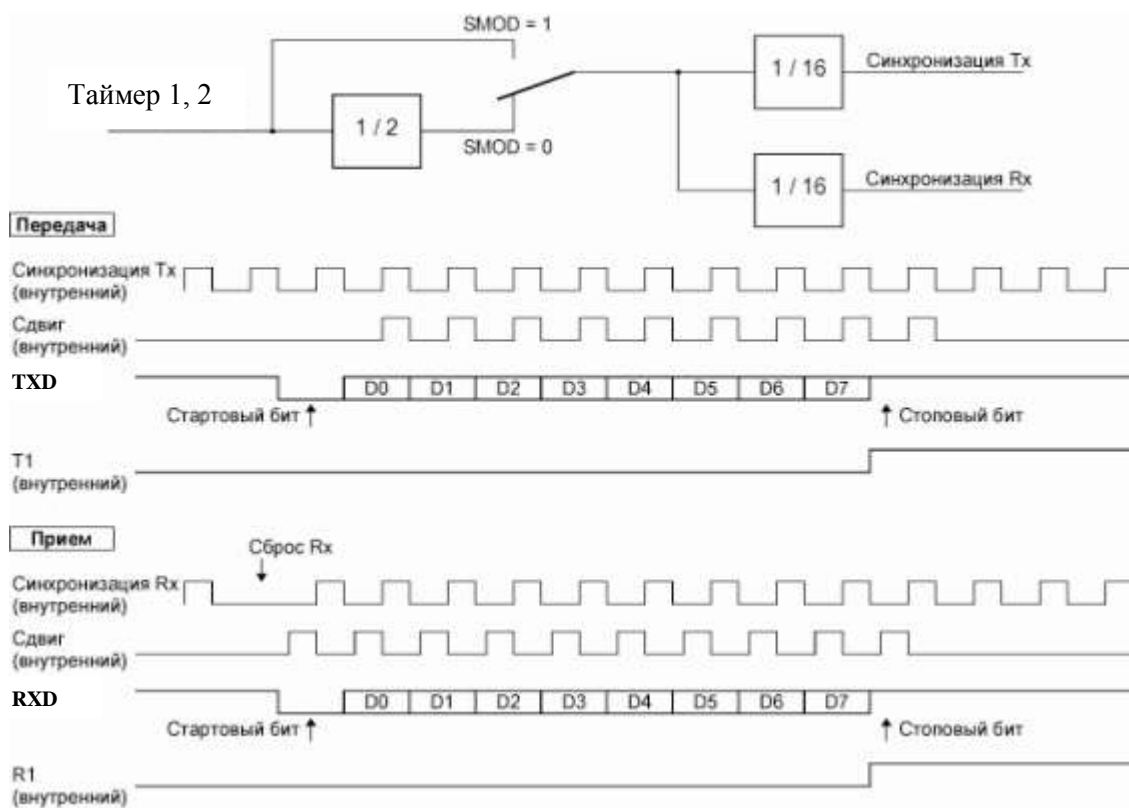


Рисунок 4.9 – Работа последовательного порта в режиме 1

Режим 2. В этом режиме через TXD передаются или из RXD принимаются 11 бит информации: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит данных может принимать значение 0 или 1, или, например, для повышения достоверности передачи путем контроля по четности в него может быть помещено значение признака паритета из слова состояния программы (PSW.0). При приеме девятый бит данных помещается в бит RB8 SCON, а стоп-бит, в отличие от режима 1, теряется. Частота приема/передачи выбирается программой и может быть равна либо $1/32$, либо $1/64$ частоты резонатора в зависимости от управляющего бита SMOD.

Режим 3. Режим 3 совпадает с режимом 2 во всех деталях, за исключением частоты приема/передачи, которая является величиной переменной и задается Таймером 1 или Таймером 2.

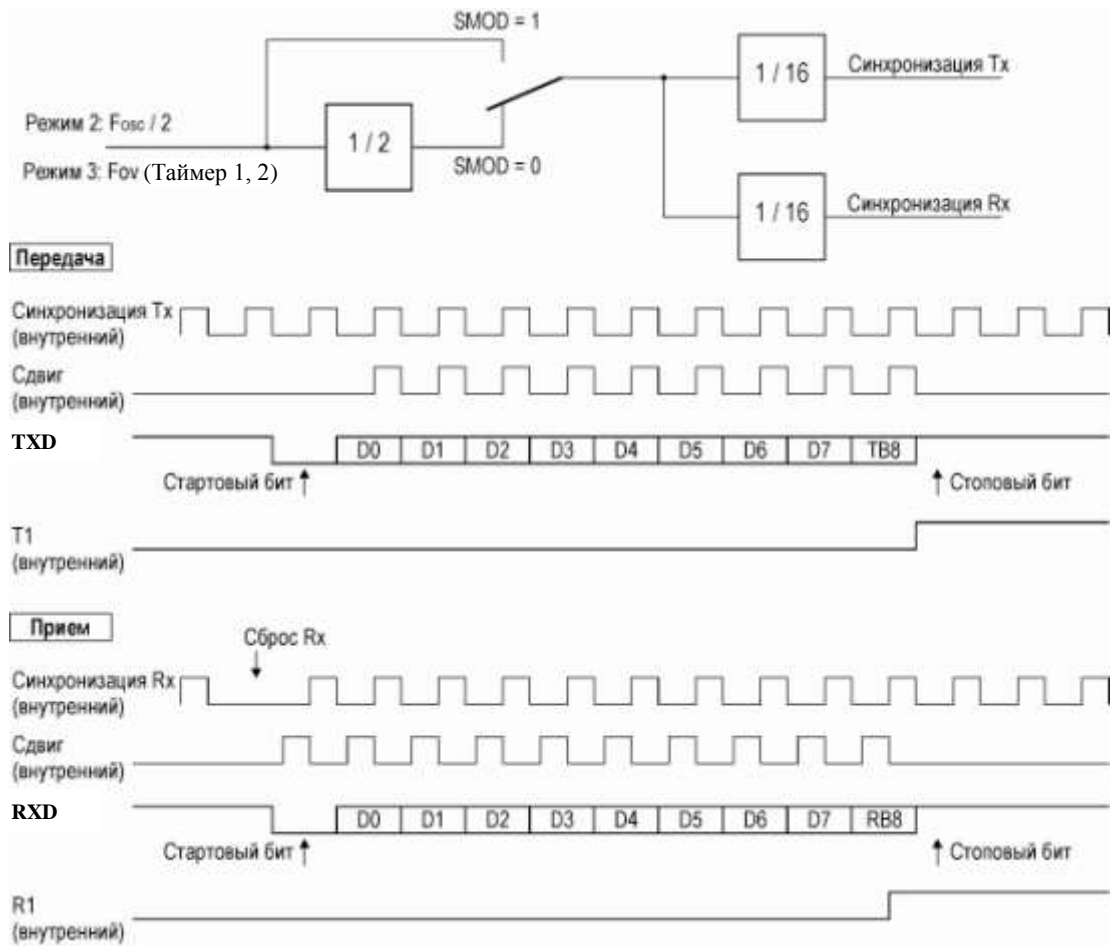


Рисунок 4.10 – Работа последовательного порта в режимах 2, 3

4.9.1 Регистр управления/статуса приемопередатчика SCON

Управление режимом работы приемопередатчика осуществляется через специальный регистр с символическим именем SCON. Этот регистр содержит не только управляющие биты, определяющие режим работы последовательного порта, но и девятый бит принимаемых или передаваемых данных (RB8 и TB8) и биты прерывания приемопередатчика (RI и TI).

Функциональное назначение бит регистра управления/статуса приемопередатчика приводится в таблице 4.11.

Таблица 4.11 – Регистр управления/статуса SCON

Символ	Позиция	Имя и назначение
SM0/FE	SCON.7	Биты управления режимом работы приемопередатчика. Устанавливаются/сбрасываются программно (см. примечание)
SM1	SCON.6	
SM2	SCON.5	Бит управления режимом приемопередатчика. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0
REN	SCON.4	Бит разрешения приема. Устанавливается программно для разрешения/запрета приема последовательных данных
TB8	SCON.3	Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме 9-битового передатчика
RB8	SCON.2	Прием бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме 9-битового приемника
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания
RI	SCON.0	Флаг прерывания приемника. Устанавливается аппаратно при приеме байта. Сбрасывается программно после обслуживания прерывания
Примечание		
SM0	SM1	Режим работы приемопередатчика
0	0	Режим 0. Сдвигающий регистр расширения ввода/вывода
0	1	Режим 1. 8-битовый асинхронный приемопередатчик. Изменяемая скорость передачи
1	0	Режим 2. 9-битовый асинхронный приемопередатчик. Фиксированная скорость передачи
1	1	Режим 3. 9-битовый асинхронный приемопередатчик. Изменяемая скорость передачи

В бите TB8 программно устанавливается значение девятого бита данных, который будет передан в режиме 2 или 3. В бите RB8 фиксируется в режимах 2 и 3 девятый принимаемый бит данных. В режиме 1 в бит RB8 заносится стоп-бит. В режиме 0 бит RB8 не используется.

Флаг прерывания передатчика TI устанавливается аппаратно в конце периода передачи стоп-бита во всех режимах. Соответствующая подпрограмма обслуживания прерывания должна сбрасывать бит TI.

Флаг прерывания приемника RI устанавливается аппаратно в конце периода приема восьмого бита данных в режиме 0 и в середине перио-

да приема стоп-бита в режимах 1, 2 и 3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

4.9.2 Скорость приема/передачи последовательного порта

Скорость приема/передачи, т. е. частота работы приемопередатчика в различных режимах, определяется различными способами.

В режиме 0 частота передачи зависит только от резонансной частоты кварцевого резонатора $f_{\text{РЕЗ}}$: $f = f_{\text{РЕЗ}} / 12$. За машинный цикл последовательный порт передает один бит информации. В режимах 1, 2 и 3 скорость приема/передачи зависит от значения управляющего бита SMOD в регистре специальных функций PCON (таблица 4.12). В режиме 2 частота передачи определяется выражением $f_2 = (2^{\text{SMOD}}/64)f_{\text{РЕЗ}}$. В режимах 1 и 3 формировании частоты передачи принимает Таймер 1 или Таймер 2. Частота передачи определяется частотой переполнения Таймера 1: $f_{1,3} = (2^{\text{SMOD}}/32)f_{\text{OVT1}}$. При этом прерывание от Таймера 1 должно быть заблокировано. Параметры частоты передачи, формируемой Таймером 2, приведены в разделе 8.

Таблица 4.12 – Регистр управления мощностью PCON

Символ	Позиция	Наименование и функция
1	2	3
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0. По сбросу SMOD = 0
SMOD0	PCON.6	Бит выбора SCON.7. Если SMOD0 = 0, то бит SCON.7 представляет собой флаг SM0, иначе бит SCON.7 представляет флаг FE
-	PCON.5	Не используется
-	PCON.4	Не используется
GF1	PCON.3	Флаги, специфицируемые пользователем (флаги общего назначения)
GF0	PCON.2	
PD	PCON.1	Бит пониженной мощности. При установке бита в 1 микроконтроллер переходит в режим пониженной потребляемой мощности
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микроконтроллер переходит в режим холостого хода

Окончание таблицы 4.12

Примечание - При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс содержимого PCON выполняется путем загрузки в него кода 0XXX0000.

В таблице 4.13 приводится описание способов настройки T/C1 для получения типовых частот передачи данных через приемопередатчик в режимах 2 и 3. Отметим, что скорость приема и передачи могут отличаться друг от друга.

Таблица 4.13 – Настройка Таймера 1 для управления частотой работы приемопередатчика

Частота приема/передачи (BAUD RATE)	Частота резонатора, МГц	Таймер/счетчик 1			
		SMOD	C/T	Режим (MODE)	Перезагру- жаемое число
Режим 0, макс: 1 МГц	12	X	X	X	X
Режим 2, макс: 375 кГц	12	1	X	X	X
Режим 1,3:	12	1	0	2	0FF _H
19,2 кГц	11,059	1	0	2	0FD _H
9,6 кГц	11,059	0	0	2	0FD _H
4,8 кГц	11,059	0	0	2	0FA _H
2,4 кГц	11,059	0	0	2	0F4 _H
1,2 кГц	11,059	0	0	2	0E8 _H
137,5 Гц	11,059	0	0	2	01D _H
110 Гц	6	0	0	2	072 _H
110 Гц	12	0	0	1	0FEEB _H

Предельно низких частот приемо-передачи можно достичь при использовании Таймера 1 в режиме с автоперезагрузкой (16-битовый таймер) и разрешении прерываний от таймера (старший полубайт TMOD = 0010_B). При этом частота передачи определяется выражением

$$f_{1,3} = (2^{\text{SMOD}}/32)(f_{\text{PEЗ}}/12)(256 - (\text{TH1})).$$

Перезагрузка 16-битового таймера должна осуществляться программным путем.

4.9.3 Особенности работы приемопередатчика в различных режимах

Режим 0

Данные передаются и принимаются через вывод RXD. Через вывод TXD выдаются синхросигналы сдвига. Передаются/принимаются 8 бит младшим битом вперед, частота обмена – фиксированная, равная 1/12 частоте резонатора.

Передача начинается любой командой, по которой в SBUF поступает байт данных. В момент времени S6P2 устройство управления микроконтроллера по сигналу "Запись в буфер" записывает байт в сдвигающий регистр передатчика, устанавливает триггер девятого бита и запускает блок управления передачей, который через один машинный цикл вырабатывает внутренний разрешающий сигнал "Посылка". При этом в момент S6P2 каждого машинного цикла содержимое сдвигающего регистра сдвигается вправо (младшими битами вперед) и поступает на выход RXD. В освобождающиеся старшие биты сдвигающего регистра передатчика записываются нули. При получении от детектора нуля сигнала "Передатчик пуст" блок управления передатчиком снимает сигнал "Посылка" и устанавливает флаг TI (момент S1P1 десятого машинного цикла после поступления сигнала "Запись в буфер").

Прием начинается при условии $REN = 1$ и $RI = 0$. В момент S6P2 следующего машинного цикла блок управления приемником формирует разрешающий сигнал "Прием", по которому на выход TXD передаются синхросигналы сдвига, и в сдвигающем регистре приемника начинают формироваться значения бит данных, которые считываются с входа RXD в моменты S5P2 каждого машинного цикла. В момент S1P1 десятого машинного цикла после сигнала "Запись в SCON" блок управления прием-

ником переписывает содержимое сдвигающего регистра в буфер, снимает разрешающий сигнал "Прием" и устанавливает флаг RI.

Режим 1

Через выход TXD приемопередатчик передает, а с входа RXD принимает 10 бит: старт-бит (0), 8 бит данных и стоп-бит (1). При приеме стоп-бит поступает в бит RB8 регистра SCON.

Передача инициируется любой командой, в которой получателем байта является регистр SBUF. Генерируемый при этом управляющий сигнал "Запись в буфер" загружает 1 в девятый бит сдвигающего регистра передатчика, запускает блок управления передачей и в момент времени S1P1 формирует разрешающий сигнал "Посылка". По этому сигналу на вывод TXD сначала поступает старт-бит, а затем (по внутреннему разрешающему сигналу "Данные") биты данных. Каждый период передачи бита равен 16 тактам внутреннего счетчика-делителя на 16.

Прием начинается при обнаружении перехода сигнала на входе RXD из состояния 1 в состояние 0. Для этого под управлением внутреннего счетчика вход RXD опрашивается 16 раз за период представления бита. Как только переход из 1 в 0 на входе RXD обнаружен, в сдвигающий регистр приемника загружается код $1FF_H$, внутренний счетчик по модулю 16 немедленно сбрасывается и перезапускается для выравнивания его переходов с границами периодов представления принимаемых бит. Таким образом, каждый период представления бита делится на 16 периодов внутреннего счетчика. В состояниях 7, 8 и 9 счетчика в каждом периоде представления бита производится опрос сигнала на входе RXD. Считанное значение принимаемого бита – это то, которое было получено по меньшей мере дважды из трех замеров (мажоритарное голосование по принципу "два из трех"). Если значение, принятое в первом такте, не равно 0, то блок управления приемом вновь возвращается к поиску перехода из 1 в 0. Этот механизм обеспечивает подавление ложных (сбойных) старт-бит. Истинный старт-бит сдвигается в регистре прием-

ника, и продолжается прием остальных бит посылки. Блок управления приемом сформирует сигнал "Загрузка буфера", установит RB8 и флаг RI только в том случае, если в последнем такте сдвига выполняются два условия: бит RI = 0 и либо SM2 = 0, либо принятый стоп-бит равен 1. Если хотя бы одно из этих двух условий не выполняется, то принятая последовательность бит теряется. В это время, вне зависимости от того, выполняются указанные условия или нет, блок управления приемником вновь начинает отыскивать переход из 1 в 0 на входе RXD.

Режимы 2, 3

Через вывод TXD приемопередатчик передает или с вывода RXD принимает 11 бит: старт-бит (0), 8 бит данных, программируемый девятый бит и стоп-бит (1). На временных диаграммах (рисунок 4.10) показана работа приемопередатчика при передаче и приеме данных в режимах 2 и 3. Как видно из них, в режимах 2 и 3 прием данных ничем не отличается от приема в режиме 1 за исключением того, что в бит RB8 в этих режимах заносится не стоп-бит, а девятый бит данных. Естественно, это никак не изменяет временные диаграммы в режимах 2 и 3 в сравнении с режимом 1. Необходимо также отметить, что несколько изменяются условия окончания цикла приема: блок управления приемником сформирует управляющий сигнал "Загрузка буфера", загрузит RB8 и установит флаг RI только в том случае, если в последнем такте сдвига выполняются два условия: бит RI = 0 и либо SM2 = 0, либо значение принятого девятого бита данных равно 1.

4.10 Работа приемопередатчика в мультимикропроцессорных системах

Описание приемопередатчика микроконтроллера будет неполным, если не упомянуть еще об одной возможности. Речь идет о работе приемопередатчика в системах децентрализованного управления. Такие системы используются для управления и регулирования в распределенных объектах. При этом возникает задача обмена информацией между мно-

жеством микроконтроллеров, объединенных в локальную вычислительно-управляющую сеть. Как правило, локальные сети на основе микроконтроллеров имеют магистральную архитектуру с отдельным моноканалом (коаксиальный кабель, витая пара, оптическое волокно), по которому осуществляется обмен информацией между микроконтроллерами.

В регистре специальных функций *SCON* микроконтроллера имеется управляющий бит *SM2*, который в режимах 2 и 3 приемопередатчика позволяет относительно простыми средствами реализовать межпроцессорный обмен информацией в локальных управляющих сетях.

Механизм межпроцессорного обмена информацией через последовательный порт микроконтроллера построен на том, что в режимах 2 и 3 программируемый девятый бит данных при приеме фиксируется в бите *RB8*. Приемопередатчик может быть запрограммирован таким образом, что при получении стоп-бита прерывание от приемника будет возможно только при условии $RB8 = 1$. Это выполняется установкой управляющего бита *SM2* в регистре *SCON*. Программа реализации протокола сетевого обмена информацией должна быть построена таким образом, чтобы при получении байта-идентификатора ($RB8 = 1$) во всех ведомых микроконтроллерах произошли прерывания прикладных программ и вызов подпрограммы сравнения байта-идентификатора с кодом собственного сетевого адреса. Адресуемый микроконтроллер сбрасывает свой управляющий бит *SM2* и готовится к приему блока данных. Остальные ведомые микроконтроллеры, адрес которых не совпал с кодом байта-идентификатора, оставляют неизменным состояние $SM2 = 1$ и передают управление основной программе. При $SM2 = 1$ информационные байты, передаваемые по моноканалу и поступающие в приемопередатчик ведомых микроконтроллеров, прерывания не вызывают, т. е. игнорируются. В микроконтроллере реализован так называемый "Расширенный UART", который выполняет все функции стандартного UART плюс обнаружение ошибок приема стоповых битов и автоматическое распозна-

вание адреса. Проверка на ошибку приема стопового бита выполняется в режимах 1, 2, 3. Если стоповый бит в определенное время не обнаружен, устанавливается флаг ошибки кадра FE. Этот флаг делит с битом управления SM0 один и тот же разряд SCON.7. Бит SMOD0 (PCON.6) указывает при обращении на бит управления или на флаг ошибки кадра. Флаг FE должен сбрасываться программно.

Для автоматического распознавания адреса введены дополнительно регистр спецфункций SADDR (адрес 0A9_H), хранящий индивидуальный адрес, и регистр SADEN (адрес 0B9_H), хранящий маску адреса. Именно маска дает возможность адресовать в каждый момент времени одно или несколько устройств.

4.11 Сброс микроконтроллера

Сброс микроконтроллера осуществляется единичным уровнем сигнала. Сигнал сброса подается на вывод RST.

Сброс достигается удержанием вывода RST в состоянии высокого уровня в течение, по крайней мере, двух машинных циклов (24 периодов колебаний) ПРИ РАБОТАЮЩЕМ ГЕНЕРАТОРЕ. CPU отвечает выработкой внутреннего сброса, алгоритм которого описан ниже (рисунок 4.11).

Внешний сигнал сброса асинхронен в отношении внутренних сигналов микроконтроллера. Вывод RST опрашивается в течение состояния S5 фазы P2 каждого машинного цикла. Поскольку выходы портов сохраняют свое текущее состояние в течение 19 периодов колебаний после обнаружения единицы на выводе RST, длительность сигнала сброса должна быть большей 19 периодов колебаний после сигнала внешнего сброса.

Пока вывод RST имеет единичный уровень, выходы ALE и PSEN# превращаются во входы, и потенциал на них медленно ползет вверх. После возврата RST в нулевой уровень требуется от одного до двух машинных циклов, чтобы ALE и PSEN# начали синхронизироваться. Поэтому

другие устройства нежелательно синхронизировать этими сигналами, если предполагается использовать сигнал сброса.

Подача на выходы ALE и PSEN# сигнала нулевого уровня в момент сброса может перевести микроконтроллер в неопределенное состояние.

Алгоритм внутреннего сброса записывает нули во все регистры специальных функций, кроме регистров-защелок портов ввода-вывода, указателя стека и SBUF. В защелках портов при инициализации установлено 0FF_H, в указателе стека – 07_H, а содержимое SBUF неопределенно. В таблице 4.14 приведен список некоторых регистров специальных функций и их значений после прохождения сигнала сброса. Полный список регистров и их состояний приведен в разделе 8.

Таблица 4.14 – Состояние регистров SFR после сброса

Наименование регистра	Значение после сброса
1	2
PC	0000 _H
ACC	00 _H
B	00 _H
PSW	00 _H
SP	07 _H
DPTR	0000 _H
P0...P3	0FF _H
IP	XXX00000 _B
IE	0XX00000 _B
TMOD	00 _H
TCON	00 _H
TH0	00 _H
TL0	00 _H
TH1	00 _H
TL1	00 _H

Окончание таблицы 4.14

1	2
SCON	00 _H
SBUF	XXXXXXXX _B
PCON	0XXX0000 _B
Примечание – X – значение бита не определено.	

Сброс не оказывает воздействия на состояние ячеек внутреннего ОЗУ микроконтроллера. Однако необходимо учитывать, что их состояние после включения питающего напряжения не определено.

4.12 Сброс по включению питания

Если подача питающего напряжения не сопровождается достоверным сбросом (т. е. удержанием единичного уровня на входе RST в течение 24 периодов резонатора), то микроконтроллер может начать выполнение программы до того, как в регистры специальных функций будут занесены значения. При этом нельзя гарантировать корректность выполнения программы. Следовательно, микроконтроллер должен иметь цепи, обеспечивающие автоматическое формирование сигнала сброса при включении питания.

Значение емкости для микроконтроллера по цепи сброса составляет значение порядка 1 мкФ.

При включении питания подобная цепь удерживает высокий уровень на входе RST в течение времени, которое зависит от значения емкости и уровня, до которого она заряжена. Для гарантии достоверного сброса этот единичный уровень должен сохраняться дольше, чем генератор выработает два машинных цикла. Практика показывает, что при включении питания питающее напряжение достигает своего номинального значения, как правило, в течение примерно 10 мс. Время запуска генератора зависит от его частоты. Для резонатора с частотой 10 МГц оно составляет обычно 1 мс, а для 1 МГц резонатора оно составит 10 мс. Следовательно, постоянная времени RC-цепи, соединенной с выводом RST, должна в не-

сколько раз превосходить указанные временные интервалы и составлять (40 – 80) мс.

В приведенной схеме резкое падение питающего напряжения вызывает мгновенное понижение напряжения на выводе RST ниже 0 В. Однако, микроконтроллеры снабжены внутренними ограничительными диодами, и эти броски напряжения не выводят их из строя.

Следует отметить, что пока не запустится генератор микроконтроллера и не выполнится описанный выше алгоритм сброса, выходы портов P0...P3 будут находиться в неопределенном состоянии.

4.13 Режимы работы микроконтроллера с пониженным энергопотреблением

Во многих вариантах использования микроконтроллеров энергопотребление является одним из основных параметров.

Микроконтроллеры 1830BE32У, 1830BE32АУ и 1830BE52У имеют 2 режима с пониженным потреблением: режим холостого хода и режим пониженного потребления тока или выключенного питания. Упрощенная схема аппаратной реализации режимов пониженного энергопотребления представлена на рисунке 4.12. Дословно второй режим, называемый в англоязычной литературе режимом "Power Down Mode", можно перевести так, как указано нами выше. Управление режимами энергопотребления осуществляется при помощи регистра PCON. Назначение битов этого регистра приведено в таблице 4.12. Бит PCON.0 называется IDL, бит PCON.1 называется PD. Биты регистра PCON, отвечающие за режимы энергопотребления приведены в таблице 4.15.

Таблица 4.15 – Биты энергопотребления регистра PCON

Символ	Позиция	Назначение
1	2	3
GF1	PCON.3	Флаги, специфицируемые пользователем (флаги общего назначения)
GF0	PCON.2	

Окончание таблицы 4.15

1	2	3
PD	PCON.1	Бит пониженной мощности. При установке бита в 1 микроконтроллер переходит в режим пониженной потребляемой мощности
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микроконтроллер переходит в режим холостого хода
Примечание – При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс содержимого PCON выполняется путем загрузки в него кода 0XXX0000 _В .		

В режиме холостого хода ($IDL = 1$) генератор микроконтроллера работает, подавая вырабатываемые тактовые сигналы на схему прерываний, последовательный порт и на таймеры/счетчики. Все регистры сохраняют свое значение, на выводах всех портов удерживается то логическое состояние, которое было на них в момент перехода в режим холостого хода. Однако синхросигнал генератора, синхронизирующий CPU, отключается. В режиме выключенного напряжения питания ($PD = 1$) генератор останавливается (рисунок 4.12). Прекращается тактирование не только CPU, но и последовательного порта, таймеров/счетчиков, схемы прерываний. Как и в режиме холостого хода, состояние регистров, резидентного ОЗУ и выводов портов остается неизменным. Адрес регистра PCON – 87_Н.

4.14 Режим холостого хода

В режим холостого хода IDLE микроконтроллер переводится любой командой, устанавливающей 1 в бите PCON.0 ($IDL = 1$). Эта команда оказывается последней в цепочке выполняемых команд – в режиме холостого хода выполнение программы приостанавливается, т. к. на CPU перестает подаваться сигнал с тактового генератора. Однако содержимое внутреннего ОЗУ и регистров специальных функций остается неизменным, выводы портов удерживают значения, которые были на них до перехода в режим холостого хода, на таймеры/счетчики, приемопередатчик

и на схему прерывания продолжают поступать тактовые сигналы. На выводах ALE и PSEN# устанавливаются сигналы единичного уровня.

Состояние выводов портов зависит от типа ОЗУ, с которым микроконтроллер обменивался информацией перед переходом в режим холостого хода. При работе с внутренним ОЗУ на выводах портов присутствуют данные из соответствующих SFR (естественно, если порт в режиме вывода информации). При работе с внешним ОЗУ выводы порта 0 переходят в высокоимпедансное состояние, а на выводах порта 2 сохраняется адресная информация. На выводах портов 1 и 3 присутствуют данные из SFR портов.

Прекратить холостой ход возможно двумя способами. Вызов любого из прерываний приведет к аппаратному стиранию PCON.0 (IDL = 0), прекращающему холостой ход. Прерывание будет обслужено, и следующей после RETI выполняемой командой будет та команда, которая следует за командой, приведшей к переходу микроконтроллера в режим холостого хода.

Флаги GF0 и GF1 регистра PCON могут использоваться для индикации того, произошло ли прерывание во время нормальной работы или во время холостого хода. Например, команда, запускающая холостой ход, может также устанавливать один или оба флага. Когда холостой ход прерывается, сервисная программа прерывания может проверять состояние флагов.

Другой способ прекращения холостого хода – с помощью аппаратного сброса. Поскольку синхрогенератор продолжает работать, аппаратный сброс должен поддерживаться в активном состоянии только в течение двух машинных циклов (24 периодов колебаний). Сигнал сброса стирает бит PCON.0 (IDL = 0). В этот момент CPU возобновляет выполнение программы с того места, где оно было остановлено; таким образом, следующая команда – та, что следует за командой, вызвавшей холостой ход. Как показано на рисунке 4.11, перед началом отработки алгоритма

внутреннего запуска могут иметь место два или три машинных цикла выполнения программы. Встроенное в микросхему устройство в это время препятствует доступу к внутреннему ОЗУ, но доступ к выводам порта не ограничен. Чтобы исключить возможность появления неопределенных выходных сигналов на выводах порта, команда, следующая за вызывающей холостой ход, не должна быть командой, записывающей информацию в SFR порта или во внешнее ОЗУ данных.

Следует помнить, что после аппаратного сброса содержимое SFR переопределяется.

4.15 Режим выключенного напряжения питания

Команда, устанавливающая в 1 бит PCON.1 ($PD = 1$), переводит микроконтроллер в режим выключенного напряжения питания Power Down. В этом режиме генератор микроконтроллера останавливается (рисунок 4.12). С остановкой синхрогенератора прекращают функционирование не только CPU, но и таймеры/счетчики, приемопередатчик, схема прерываний.

Единственный способ выйти из этого режима – аппаратный сброс. Он переопределяет содержимое всех SFR, но не меняет содержимого встроенного ОЗУ.

4.16 Встроенный тактовый генератор

Микроконтроллеры способны выключать свой генератор программным путем (записью 1 в бит PD регистра PCON). Кроме того, сигнал внешнего синхрогенератора должен быть подан на вход XTAL1 (вход XTAL2 не используется). Упрощенная схема подключения тактового генератора и использование его для включения режимов энергосбережения приведена на рисунках 4.12 и 4.13. Генератор можно использовать с внешними компонентами. Обычно, если элементом обратной связи является кристалл кварца, то $C1 = C2 = 30$ пФ, а если используется керамический резонатор, то $C1 = C2 = 47$ пФ. Рекомендуется использовать тактовый сигнал от внешнего тактового генератора.

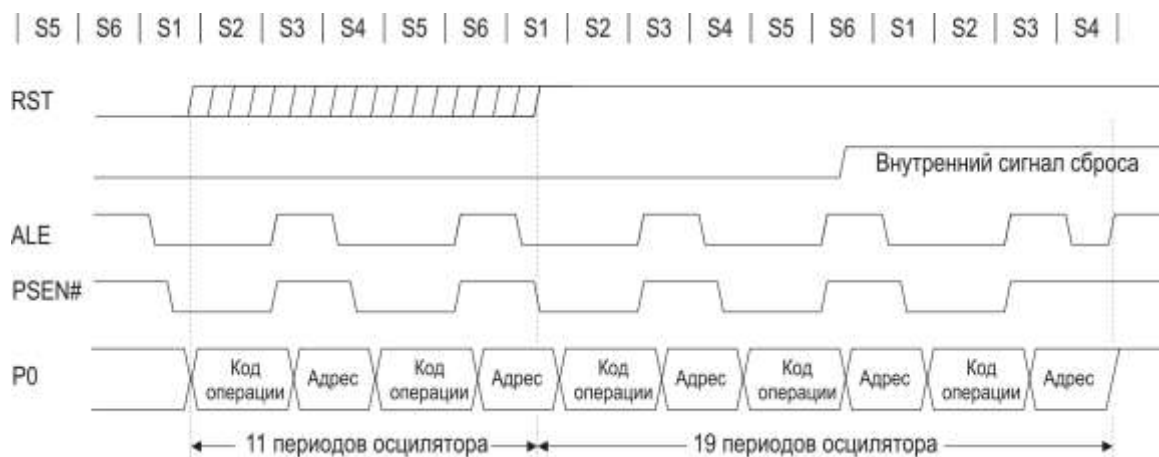


Рисунок 4.11 – Временные диаграммы сигналов микроконтроллеров при сбросе

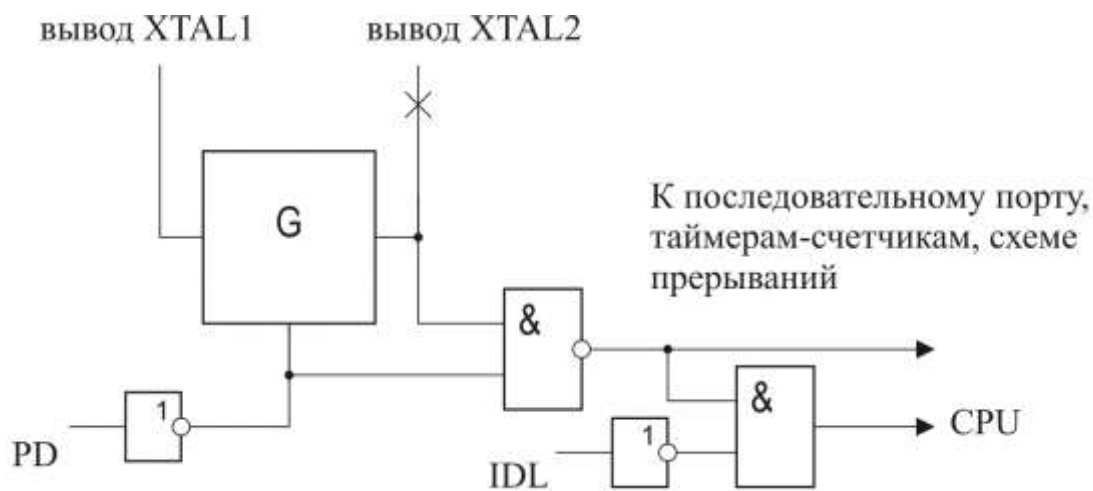


Рисунок 4.12 – Упрощенная схема аппаратной реализации режимов пониженного энергопотребления

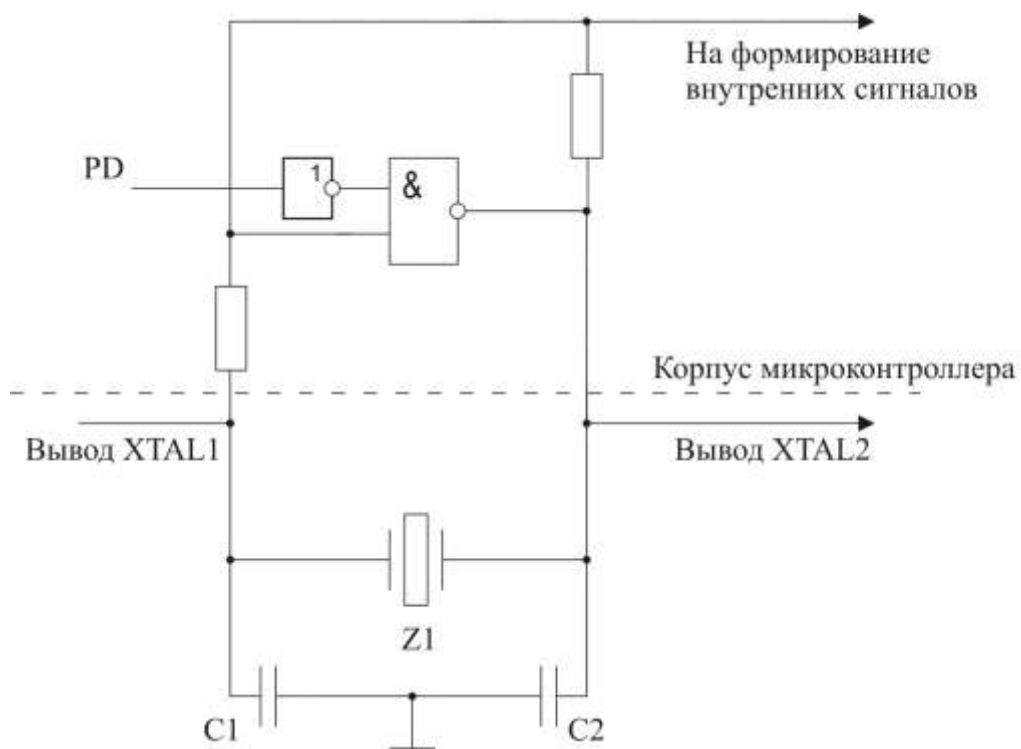


Рисунок 4.13 – Тактовый генератор микроконтроллера

Временная диаграмма работы тактового генератора приведена на рисунке 4.14.

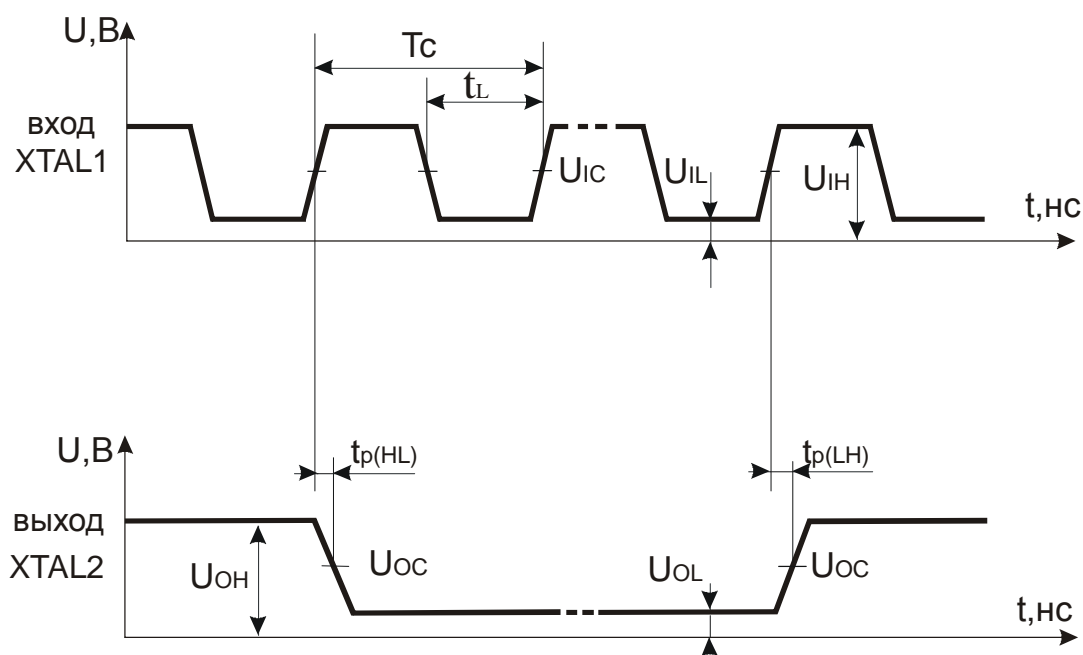


Рисунок 4.14 – Временная диаграмма работы тактового генератора

Тактовый генератор обеспечивает следующие параметры тактового сигнала на частоте 12 МГц для ИС 1830ВЕ32У, 1830ВЕ52У и 16 МГц для ИС 1830ВЕ32АУ (более подробно приведено в технических условиях АЕЯР.431280.378ТУ):

- $T_C = 80$ нс (для ИС 1830ВЕ32У, 1830ВЕ52У), $T_C = 62,5$ нс (для ИС 1830ВЕ32АУ); $t_L = 0,5T_C$;
- длительность фронтов на XTAL1 – не более 10 нс;
- t_p – время распространения сигнала: минимум 20 нс, максимум 110 нс;
- $U_{IL} = 0$ В; $U_{IH} = U_{CC}$;
- U_{OH} , U_{OL} – выходное напряжение высокого и низкого уровня;
- $U_{IC} = U_{OC} = 1,5$ В – уровни измерения t_p .

5 Общая характеристика системы команд микроконтроллеров

Система команд микроконтроллеров включает в свой состав 111 основных команд. Длина команд составляет 1, 2 или 3 байта, причем большинство команд (94 %) одно- или двухбайтные. Все команды выполняются за 1 или 2 машинных цикла (1,0 или 2,0 мкс при тактовой частоте 12 МГц), исключение составляют лишь команды MOVX A,@Ri, MOVX @Ri,A и LCALL addr16, которые выполняются за 3 машинных цикла (3,0 мкс). В качестве операндов команд микроконтроллеры могут использовать: отдельные биты, 4-х битные цифры, байты и двухбайтовые слова. Всего микроконтроллеры выполняют 13 типов команд (рисунок 5.1).

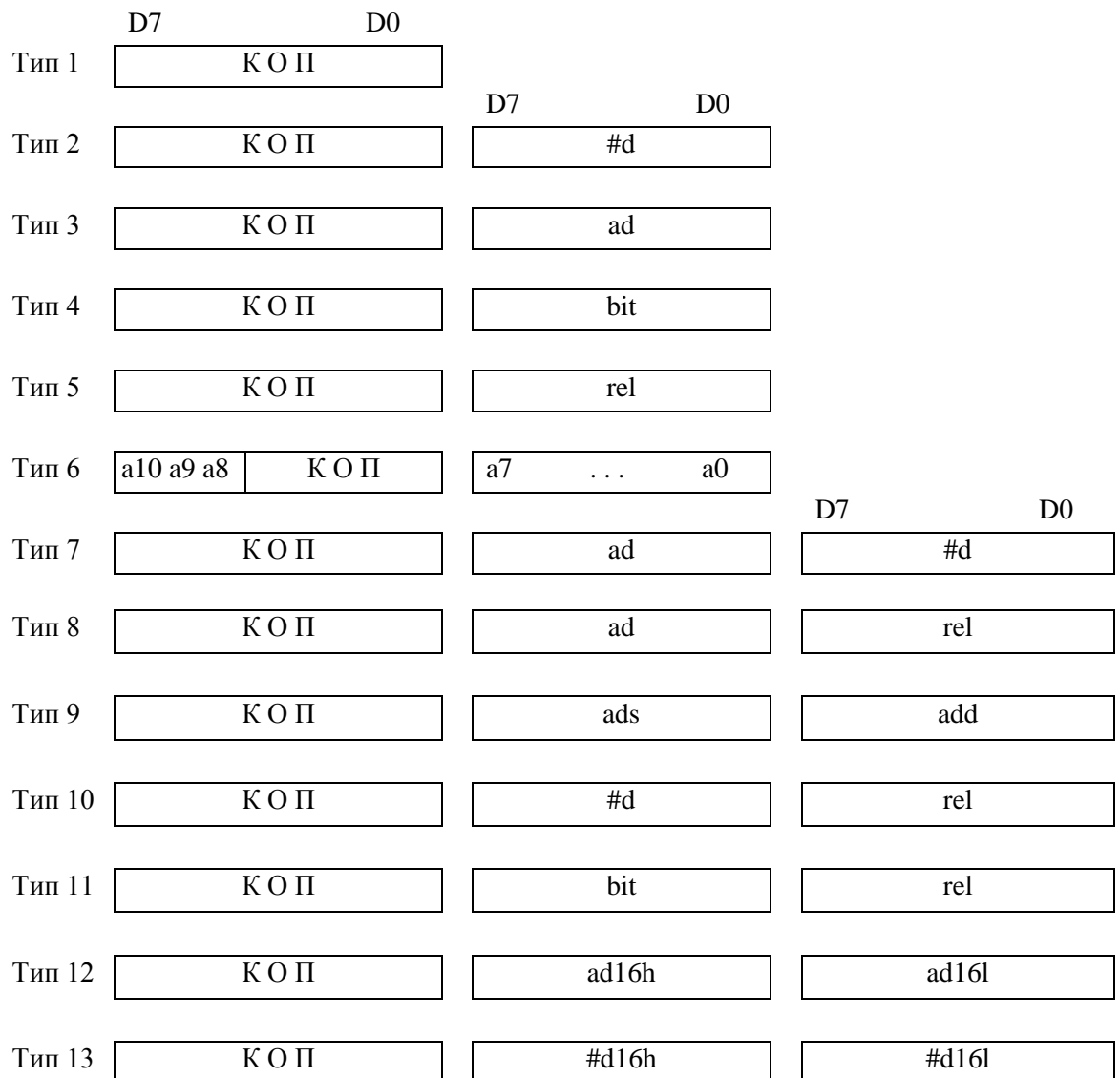


Рисунок 5.1 – Типы команд микроконтроллеров

Как следует из рисунка 5.1, первый байт команды всегда содержит код операции (КОП), а второй и третий байты (если они присутствуют в команде) содержат адреса операндов или непосредственные значения операндов.

Наиболее существенным является возможность адресовать отдельные биты в резидентной памяти данных. Кроме того, выше уже отмечалось, что отдельные регистры из блока регистров специальных функций так же допускают адресацию отдельных бит.

Карты адресов отдельных бит в резидентной памяти данных и в блоке регистров специальных функций приведены на рисунках 5.2 и 5.3.

Адреса

	D7				D0			
7Fh-								
30h-								
2Fh-	7F	7E	7D	7C	7B	7A	79	78
2Eh-	77	76	75	74	73	72	71	70
2Dh-	6F	6E	6D	6C	6B	6A	69	68
2Ch-	67	66	65	64	63	62	61	60
2Bh-	5F	5E	5D	5C	5B	5A	59	58
2Ah-	57	56	55	54	53	52	51	50
29h-	4F	4E	4D	4C	4B	4A	49	48
28h-	47	46	45	44	43	42	41	40
27h-	3F	3E	3D	3C	3B	3A	39	38
26h-	37	36	35	34	33	32	31	30
25h-	2F	2E	2D	2C	2B	2A	29	28
24h-	27	26	25	24	23	22	21	20
23h-	1F	1E	1D	1C	1B	1A	19	18
22h-	17	16	15	14	13	12	11	10

Рисунок 5.2, лист 1 – Карта адресуемых бит в резидентной памяти данных

21h-	0F	0E	0D	0C	0B	0A	09	08
20h-	07	06	05	04	03	02	01	00
1Fh-	банк 3							
18h-								
17h-	банк 2							
10h-								
0Fh-	банк 1							
08h-								
07h-	банк 0							
00h-								

Рисунок 5.2, лист 2

	D7				D0				Имя регистра
FFh-	• • •								
F0h-	F7	F6	F5	F4	F3	F2	F1	F0	B
	• • •								
E0h-	E7	E6	E5	E4	E3	E2	E1	E0	A
	• • •								
D0h-	D7	D6	D5	D4	D3	D2	D1	D0	PSW
	• • •								
B8h-	-	-	-	BC	BB	BA	B9	B8	IP
	• • •								
B0h-	B7	B6	B5	B4	B3	B2	B1	B0	P3
	• • •								
A8h-	AF	-	-	AC	AB	AA	A9	A8	IE

Рисунок 5.3, лист 1 – Карта адресуемых бит в блоке регистров специальных функций

	• • •								
A0h-	A7	A6	A5	A4	A3	A2	A1	A0	P2
	• • •								
98h-	9F	9E	9D	9C	9B	9A	99	98	SCON
	• • •								
90h-	97	96	95	94	93	92	91	90	P1
	• • •								
88h-	8F	8E	8D	8C	8B	8A	89	88	TCON
	• • •								
80h-	87	86	85	84	83	82	81	80	P0

Рисунок 5.3, лист 2

Все команды микроконтроллера можно разбить на 5 функциональных групп:

- команды пересылки данных;
- команды арифметических операций;
- команды логических операций;
- команды операций над битами;
- команды передачи управления.

При рассмотрении команд будут использоваться следующие обозначения:

R_n ($n = 0, 1, \dots, 7$) – регистр общего назначения в выбранном банке регистров;

@ R_i ($i = 0, 1$) – регистр общего назначения в выбранном банке регистров, используемый в качестве регистра косвенного адреса;

ad или direct – адрес прямоадресуемого байта;

ads – адрес прямоадресуемого байта-источника;

add – адрес прямоадресуемого байта-получателя;

ad11 или addr11 – 11-разрядный абсолютный адрес перехода;
 ad16 или addr16 – 16-разрядный абсолютный адрес перехода;
 rel – относительный адрес перехода;
 #d или #data – непосредственный операнд, 1 байт;
 #d16 или #data16 – непосредственный операнд, 2 байта;
 bit – адрес прямоадресуемого бита;
 /bit – инверсия прямоадресуемого бита;
 A – аккумулятор;
 PC – счетчик команд;
 DPTR – регистр указатель данных;
 () – содержимое ячейки памяти или регистра.

5.1 Группа команд пересылки данных

Данная группа команд представлена 28 командами, краткое описание которых приведено в таблице 5.1, где так же указаны тип команды (Т) в соответствии с рисунком 6.1, длина команды в байтах (Б) и время выполнения команды в машинных циклах (Ц).

Таблица 5.1 – Команды пересылки данных

Название команды	Мnemonic	КОП	Т	Б	Ц	Операция
1	2	3	4	5	6	7
Пересылка в аккумулятор из регистра (n = 0 – 7)	MOV A, Rn	11101rrr	1	1	1	(A) = (Rn)
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	11100101ad	3	2	1	(A) = (ad)
Пересылка в аккумулятор байта из РПД (i = 0, 1)	MOV A, @Ri	1110011i	1	1	1	(A) = ((Ri))
Загрузка в аккумулятор константы	MOV A, #d	01110100 #d	2	2	1	(A) = #d
Пересылка в регистр из аккумулятора	MOV Rn, A	11111rrr	1	1	1	(Rn) = (A)
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	10101rrr ad	3	2	1	(Rn) = (ad)
Загрузка в регистр константы	MOV Rn, #d	01111rrr #d	2	2	1	(Rn) = #d

Продолжение таблицы 5.1

1	2	3	4	5	6	7
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101 ad	3	2	1	(ad) = (A)
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr ad	3	2	1	(ad) = (Rn)
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101 ads add	9	3	1	(add) = (ads)
Пересылка байта из РПД по прямому адресу	MOV ad, @Ri	1000011i ad	3	2	1	(ad) = ((Ri))
Пересылка по прямому адресу константы	MOV ad, #d	01110101 ad #d	7	3	1	(ad) = #d
Пересылка в РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) = (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	01110011i ad	3	2	2	((Ri)) = (ad)
Пересылка в РПД константы	MOV @Ri, #d	0111011i #d	2	2	1	((Ri)) = #d
Загрузка указателя данных	MOV DPTR, #d16	10010000 ad15...ad8 ad7...ad0	13	3	2	(DPTR) = #d16
Пересылка в аккумулятор байта из ПП	MOVC A, @A+DPTR	10010011	1	1	2	(A) = ((A) + (DPTR))
Пересылка в аккумулятор байта из ПП	MOVC A, @A + PC	10000011	1	1	2	(PC) = (PC) + 1 (A) = ((A) + (PC))
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	11110001i	1	1	3	(A) = ((Ri))
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11110000	1	1	2	(A) = ((DPTR))
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	3	((Ri)) = (A)
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) = (A)
Загрузка в стек	PUSH ad	11000000 ad	3	2	2	(SP) = (SP) + 1 ((SP)) = (ad)
Извлечение из стека	POP ad	11010000 ad	3	2	2	(ad) = (SP) (SP) = (SP) - 1
Обмен аккумулятора с регистром	XCH A, Rn	11001rrr	1	1	1	(A) <-> (Rn)
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101 ad	3	2	1	(A) <-> (ad)
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A ₀₋₃) <-> ((Ri) ₀₋₃)

Окончание таблицы 5.1

1	2	3	4	5	6	7
Обмен младшей тетрады аккумулятора с младшей тетрадой байта РПД	XCHD A, @Ri	1101011i	1	1	1	(A ₀₋₃) <-> ((Ri) ₀₋₃)

По команде MOV выполняется пересылка данных из второго операнда в первый. Команда MOV не имеет доступа ни к внешней памяти данных, ни к памяти программ. Для этих целей служат команды MOVX и MOVC соответственно. Команда MOVX обеспечивает чтение-запись байт из внешней памяти данных, а команды MOVC – чтение байт из памяти программ.

По команде XCH выполняется обмен байтами между аккумулятором и ячейкой РПД, а по команде XCHD – обмен младшими тетрадами (битами 0-3).

Если сравнить группы команд пересылки данных рассматриваемых микроконтроллеров и микроконтроллеров семейства 8051 (фирмы Intel), то можно выделить два существенных отличия. Во-первых, в рассматриваемом микроконтроллере "исчезли" все команды для работы со специальными регистрами: PSW, таймером, портами ввода-вывода. Теперь доступ к ним, как и к другим регистрам специальных функций, осуществляется путем задания соответствующего прямого адреса, т. е. эти команды упрятаны в команды типа 3. Например, чтение PSW в аккумулятор может быть выполнено командой

MOV A, PSW,

которая преобразуется ассемблером к виду

MOV A, 0D0h (E5 D0),

где E5 - код операции, а D0 - операнд (адрес PSW).

Следует отметить, что большинство ассемблеров допускают символические имена для регистров специальных функций, а отдельные биты

этих регистров (конечно, если выбранный регистр допускает адресацию отдельных бит) могут адресоваться путем указания имени регистра и номера бита через точку. Например, к нулевому биту аккумулятора можно обратиться по имени ACC.0. Это означает, что в микроконтроллере аккумулятор имеет два различных имени в зависимости от способа адресации: А – при регистровой адресации (например, MOV A, R0) и ACC – при использовании прямого адреса. Первый способ предпочтительнее, однако, не всегда возможен.

Вторым существенным отличием является появление команд записи данных в стек PUSH и их чтения из стека POP. Размер стека теперь ограничен лишь размером резидентной памяти данных (РПД).

5.2 Группа команд арифметических операций

Данная группа команд состоит из 24 команд, краткое описание которых приведено в таблице 5.2.

Как следует из таблицы 5.2, микроконтроллер выполняет достаточно широкий набор команд для организации обработки целочисленных данных, включая команды умножения и деления. По результату выполнения команд ADD, ADDC, SUBB, MUL и DIV устанавливаются флаги регистра слова состояния программы (PSW), структура которого приведена на рисунке 5.4.

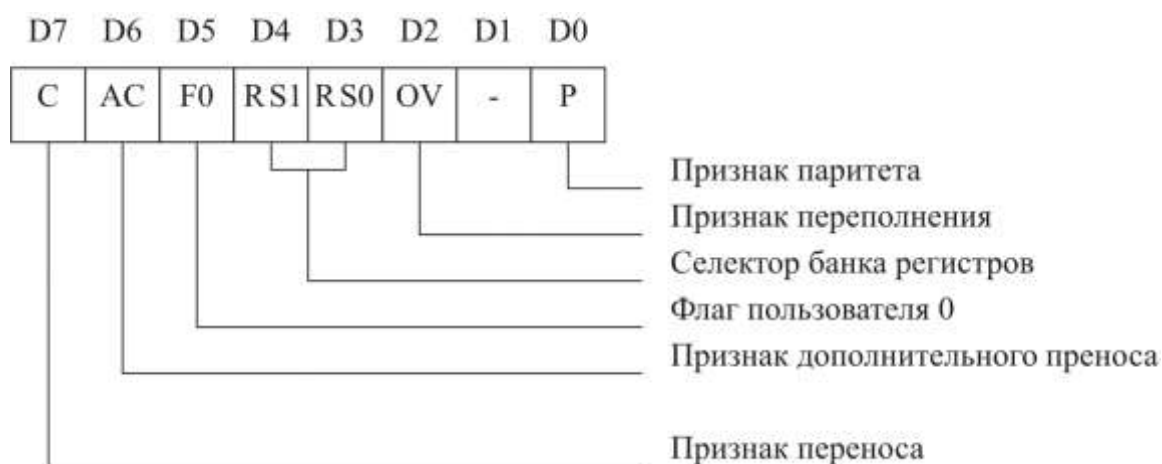


Рисунок 5.4 – Формат регистра PSW

Таблица 5.2 – Команды арифметических операций

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
1	2	3	4	5	6	7
Сложение аккумулятора с регистром (n = 0 - 7)	ADD A, Rn	00101rrr	1	1	1	$(A) = (A) + (Rn)$
Сложение аккумулятора с прямоадресуемым байтом	ADD A, ad	00100101 ad	3	2	1	$(A) = (A) + (ad)$
Сложение аккумулятора с байтом из РПД (i = 0, 1)	ADD A, @Ri	0010011i	1	1	1	$(A) = (A) + ((Ri))$
Сложение аккумулятора с константой	ADD A, #d	00100100 #d	2	2	1	$(A) = (A) + \#d$
Сложение аккумулятора с регистром и переносом	ADDC A, Rn	00111rrr	1	1	1	$(A) = (A) + (Rn) + (C)$
Сложение аккумулятора с прямоадресуемым байтом и переносом	ADDC A, ad	00110101 ad	3	2	1	$(A) = (A) + (ad) + (C)$
Сложение аккумулятора с байтом из РПД и переносом	ADDC A, @Ri	0011011i	1	1	1	$(A) = (A) + ((Ri)) + (C)$
Сложение аккумулятора с константой и переносом	ADDC A, #d	00110100 #d	2	2	1	$(A) = (A) + \#d + (C)$
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если $((A_{0-3}) > 9) \vee ((AC) = 1)$, то $(A_{0-3}) = (A_{0-3}) + 6$, затем, если $(A_{4-7}) > 9 \vee ((C) = 1)$, то $(A_{4-7}) = (A_{4-7}) + 6$
Вычитание из аккумулятора регистра и заема	SUBB A, Rn	10011rrr	1	1	1	$(A) = (A) - (C) - (Rn)$
Вычитание из аккумулятора прямоадресуемого байта и заема	SUBB A, ad	10010101 ad	3	2	1	$(A) = (A) - (C) - ((ad))$
Вычитание из аккумулятора байта РПД и заема	SUBB A, @Ri	1001011i	1	1	1	$(A) = (A) - (C) - ((Ri))$

Окончание таблицы 5.2

1	2	3	4	5	6	7
Вычитание из аккумулятора константы и заема	SUBB A, #d	10010100 #d	2	2	1	$(A) = (A) - (C) - \#d$
Инкремент аккумулятора	INC A	00000100	1	1	1	$(A) = (A) + 1$
Инкремент регистра	INC Rn	00001rrr	1	1	1	$(Rn) = (Rn) + 1$
Инкремент прямоадресуемого байта	INC ad	00000101 ad	3	2	1	$(ad) = (ad) + 1$
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	$((Ri)) = ((Ri)) + 1$
Инкремент указателя данных	INC DPTR	10100011	1	1	1	$(DPTR) = (DPTR) + 1$
Декремент аккумулятора	DEC A	00010100	1	1	1	$(A) = (A) - 1$
Декремент регистра	DEC Rn	00011rrr	1	1	1	$(Rn) = (Rn) - 1$
Декремент прямоадресуемого байта	DEC ad	00010101 ad	3	2	1	$(ad) = (ad) - 1$
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	$((Ri)) = ((Ri)) - 1$
Умножение аккумулятора на регистр B	MUL AB	10100100	1	1	2	$(B)(A) = (A) \times (B)$
Деление аккумулятора на регистр B	DIV AB	10000100	1	1	2	$(A).(B) = (A) / (B)$

Флаг C устанавливается при переносе из разряда D7, т. е. когда результат не помещается в 8 разрядов.

Флаг AC устанавливается при переносе из разряда D3 в командах сложения и вычитания и служит для реализации десятичной арифметики. Этот признак используется командой DA A.

Флаг OV (overflow) устанавливается при переносе из разряда D6, т. е. когда результат не помещается в семь разрядов и разряд D7 не может быть интерпретирован как знаковый. Этот признак служит для организации обработки чисел со знаком.

Флаг P устанавливается и сбрасывается аппаратно. Если число единичных бит в аккумуляторе нечетно, то $P = 1$, иначе $P = 0$.

5.3 Группа команд логических операций

Данная группа команд состоит из 25 команд, краткое описание которых приведено в таблице 5.3.

Таблица 5.3 – Команды логических операций

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
1	2	3	4	5	6	7
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) = (A) \wedge (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101 ad	3	2	1	$(A) = (A) \wedge (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) = (A) \wedge ((Ri))$
Логическое И прямоадресуемого байта и аккумулятора	ANL ad, A	01010010 ad	3	2	1	$(ad) = (ad) \wedge (A)$
Логическое И прямоадресуемого байта и константы	ANL ad, #d	01010011 ad #d	7	3	2	$(ad) = (ad) \wedge \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) = (A) \vee (Rn)$
Логическое ИЛИ аккумулятора и прямоадресуемого байта	ORL A, ad	01000101 ad	3	2	1	$(A) = (A) \vee (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) = (A) \vee ((Ri))$
Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100 #d	2	2	1	$(A) = (A) \vee \#d$
Логическое ИЛИ прямоадресуемого байта и аккумулятора	ORL ad, A	01000010 ad	3	2	1	$(ad) = (ad) \vee (A)$

Окончание таблицы 5.3

1	2	3	4	5	6	7
Логическое ИЛИ прямоадресуемого байта и константы	ORL ad, #d	01000011 ad #d	7	3	2	$(ad) = (ad) \vee \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) = (A) \nabla (Rn)$
Исключающее ИЛИ аккумулятора и прямоадресуемого байта	XRL A, ad	01100101 ad	3	2	1	$(A) = (A) \nabla (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) = (A) \nabla ((Ri))$
Исключающее ИЛИ прямоадресуемого байта и аккумулятора	XRL ad, A	01100010 ad	3	2	1	$(ad) = (ad) \nabla (A)$
Исключающее ИЛИ прямоадресуемого байта и константы	XRL ad, #d	01100011 ad #d	7	3	2	$(ad) = (ad) \nabla \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) = 0$
Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) = \text{NOT}(A)$
Сдвиг аккумулятора влево циклически	RL A	00100011	1	1	1	$(A_{n+1}) = (A_n),$ $n = 0 \dots 6, (A_0) = (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) = (A_n),$ $n = 0 \dots 6, (A_0) = (C),$ $(C) = (A_7)$
Сдвиг аккумулятора вправо циклически	RR A	00000011	1	1	1	$(A_n) = (A_{n+1}),$ $n = 0 \dots 6,$ $(A_7) = (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) = (A_{n+1}),$ $n = 0 \dots 6, (A_7) = (C),$ $(C) = (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0-3}) \leftrightarrow (A_{4-7})$

Как следует из таблицы 5.3, данная группа команд позволяет выполнять операции над байтами: логическое И (\wedge), логическое ИЛИ (\vee), исключающее ИЛИ (∇), инверсию (NOT), сброс в нулевое значение и сдвиг.

5.4 Группа команд операций над битами

Данная группа команд состоит из 12 команд, краткое описание которых приведено в таблице 5.4.

Как следует из таблицы 5.4, данная группа команд позволяет выполнять операции над отдельными битами: сброс, установку, инверсию бита, а так же логическое И (\wedge) и логическое ИЛИ (\vee).

В качестве "логического" аккумулятора, участвующего во всех операциях с двумя операндами, выступает признак переноса C (разряд D7 PSW). В качестве операндов могут использоваться 128 бит из резидентной памяти данных и регистры специальных функций, допускающие адресацию отдельных бит.

Таблица 5.4 – Команды операций над битами

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
1	2	3	4	5	6	7
Сброс переноса	CLR C	11000011	1	1	1	(C) = 0
Сброс бита	CLR bit	11000010 bit	4	2	1	(b) = 0
Установка переноса	SETB C	11010011	1	1	1	(C) = 1
Установка бита	SETB bit	11010010 bit	4	2	1	(b) = 1
Инверсия переноса	CPL C	10110011	1	1	1	(C) = NOT(C)
Инверсия бита	CPL bit	10110010 bit	4	2	1	(b) = NOT(b)
Логическое И бита и переноса	ANL C, bit	10000010 bit	4	2	1	(C) = (C) \wedge (b)
Логическое И инверсии бита и переноса	ANL C, /bit	10110000 bit	4	2	1	(C) = (C) \wedge NOT(b)

Окончание таблицы 5.4

1	2	3	4	5	6	7
Логическое ИЛИ бита и переноса	ORL C, bit	01110010 bit	4	2	1	$(C) = (C) \vee (b)$
Логическое ИЛИ инверсии бита и переноса	ORL C, /bit	10100000 bit	4	2	1	$(C) = (C) \vee \text{NOT } (b)$
Пересылка бита в перенос	MOV C, bit	10100010 bit	4	2	1	$(C) = (b)$
Пересылка переноса в бит	MOV bit, C	10010010 bit	4	2	1	$(b) = (C)$

5.5 Группа команд передачи управления

Группа команд передачи управления представлена командами безусловного и условного переходов, командами вызова подпрограмм и командами возврата из подпрограмм. Краткое описание команд группы приведено в таблице 5.5.

Команда безусловного перехода LJMP (L-long-длинный) осуществляет переход по абсолютному 16-битному адресу, указанному в теле команды, т. е. команда обеспечивает переход в любую точку памяти программ. Действие команды AJMP (A-absolute-абсолютный) аналогично команде LJMP, однако в теле команды указаны лишь 11 младших разрядов адреса. Поэтому переход осуществляется в пределах страницы размером в 2 Кбайт, при этом надо иметь в виду, что сначала содержимое счетчика команд увеличивается на 2 и только потом заменяются младшие 11 разрядов адреса. В отличие от предыдущих в команде SJMP (S-short-короткий) указан не абсолютный, а относительный адрес перехода. Величина смещения rel рассматривается как число со знаком, следовательно, переход возможен в пределах от минус 128 до плюс 127 байт относительно адреса команды, следующей за командой SJMP. Команда косвенного перехода JMP @A+DPTR позволяет вычислять адрес перехода в процессе выполнения самой программы.

Таблица 5.5 – Команды передачи управления

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
1	2	3	4	5	6	7
Длинный переход в полном объеме в памяти программ	LJMP ad16	00000010 ad15...ad8 ad7...ad0	12	3	2	(PC) = ad16
Абсолютный переход внутри страницы в 2 Кбайта	AJMP ad11	a ₁₀ a ₉ a ₈ 00001 a ₇ a ₆ a ₅ a ₄ a ₃ a ₂ a ₁ a ₀	6	2	2	(PC) = (PC) + 2 (PC ₀₋₁₀) = ad11
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000 rel	5	2	2	(PC) = (PC) + 2 (PC) = (PC) + rel
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	1	(PC) = (A) + (DPTR)
Переход, если аккумулятор не равен нулю	JNZ rel	01110000 rel	5	2	2	(PC) = (PC) + 2, если (A) ? 0, то (PC) = (PC) + rel
Переход, если перенос равен единице	JC rel	01000000 rel	5	2	2	(PC) = (PC) + 2, если (C) = 1, то (PC) = (PC) + rel
Переход, если перенос равен нулю	JNC rel	01010000 rel	5	2	2	(PC) = (PC) + 2, если (C) = 0, то (PC) = (PC) + rel
Переход, если бит равен единице	JB bit, rel	00100000 rel	11	3	2	(PC) = (PC) + 3, если (b) = 1, то (PC) = (PC) + rel
Переход, если бит равен нулю	JNB bit, rel	00110000 bit rel	11	3	2	(PC) = (PC) + 3, если (b) = 0, то (PC) = (PC) + rel
Переход, если бит установлен, с последующим сбросом бита	JBC bit, rel	00010000 bit rel	11	3	2	(PC) = (PC) + 3, если (b) = 1, то (b) = 0 и (PC) = (PC) + rel
Декремент регистра и переход, если не нуль	DJNZ Rn, rel	11011rrr rel	5	2	2	(PC) = (PC) + 2, (Rn) = (Rn) - 1, если (Rn) ? 0, то (PC) = (PC) + rel
Декремент прямоадресуемого байта и переход, если не нуль	DJNZ ad, rel	11010101 ad rel	8	3	2	(PC) = (PC) + 2, (ad) = (ad) - 1, если (ad) ? 0, то (PC) = (PC) + rel
Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	CJNE A, ad, rel	10110101 ad rel	8	3	2	(PC) = (PC) + 3, если (A) ? (ad), то (PC) = (PC) + rel, если (A) < (ad), то (C) = 1, иначе (C) = 0

Окончание таблицы 5.5

1	2	3	4	5	6	7
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100 #d rel	10	3	2	(PC) = (PC) + 3, если (A) ? #d, то (PC) = (PC) + rel, если (A) < #d, то (C) = 1, иначе (C) = 0
Сравнение регистра с константой и переход, если не равно	CJNE Rn, #d, rel	10111rrr #d rel	10	3	2	(PC) = (PC) + 3, если (Rn) ? #d, то (PC) = (PC) + rel, если (Rn) < #d, то (C) = 1, иначе (C) = 0
Сравнение байта в РПД с константой и переход, если не равно	CJNE @Ri, #d, rel	1011011i #d rel	10	3	2	(PC) = (PC) + 3, если ((Ri)) ? #d, то (PC) = (PC) + rel, если ((Ri)) < #d, то (C) = 1, иначе (C) = 0
Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	3	(PC) = (PC) + 3, (SP) = (SP) + 1, ((SP)) = (PC ₀₋₇), (SP) = (SP) + 1, ((SP)) = (PC ₈₋₁₅), (PC) = ad16
Абсолютный вызов подпрограммы в пределах страницы в 2 Кбайта	ACALL ad11	a ₁₀ a ₉ a ₈ 10001 a ₇ a ₆ a ₅ a ₄ a ₃ a ₂ a ₁ a ₀	6	2	2	(PC) = (PC) + 2, (SP) = (SP) + 1, ((SP)) = (PC ₀₋₇), (SP) = (SP) + 1, ((SP)) = (PC ₈₋₁₅), (PC ₀₋₁₀) = ad11
Возврат из подпрограммы	RET	00100010	1	1	2	(PC ₈₋₁₅) = ((SP)), (SP) = (SP) - 1, (PC ₀₋₇) = ((SP)), (SP) = (SP) - 1
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	(PC ₈₋₁₅) = ((SP)), (SP) = (SP) - 1, (PC ₀₋₇) = ((SP)), (SP) = (SP) - 1
Холостая команда	NOP	00000000	1	1	1	(PC) = (PC) + 1
Примечание – Ассемблер допускает использование обобщенного имени команд JMP и CALL, которые в процессе трансляции заменяются оптимальными по формату командами вызова (ACALL, LCALL) или перехода (AJMP, SJMP, LJMP).						

Команды условного перехода позволяют проверять следующие условия: JZ – аккумулятор содержит нулевое значение; JNZ – аккумулятор содержит не нулевое значение; JC – бит переноса C установлен; JNC – бит переноса C не установлен; JB – прямоадресуемый бит равен единице;

JNB – прямоадресуемый бит равен нулю;

JBC – прямоадресуемый бит равен единице и сбрасывается в нулевое значение при выполнении команды.

В отличие от микроконтроллеров семейства 8051 (фирмы Intel), все команды условного перехода содержат короткий относительный адрес, т. е. переход может осуществляться в пределах от минус 128 до плюс 127 байт относительно следующей команды.

Команда DJNZ предназначена для организации программных циклов.

Регистр Rn или байт по адресу ad, указанные в теле команды, содержат счетчик повторений цикла, а смещение rel – относительный адрес перехода к началу цикла. При выполнении команды содержимое счетчика уменьшается на единицу и проверяется на нуль. Если значение счетчика не равно нулю, то осуществляется переход на начало цикла, иначе выполняется следующая команда.

Команда CJNZ удобна для реализации процедур ожидания внешних событий. В теле команды указаны "координаты" двух байт и относительный адрес перехода rel. В качестве двух байт могут быть, например, использованы значения аккумулятора и прямоадресуемого байта или косвенно адресуемого байта и константы. При выполнении команды значения указанных двух байт сравниваются и в случае если эти значения не равны, осуществляется переход. Например, команда

```
WAIT: CJNE A, P0, WAIT
```

будет выполняться до тех пор, пока значения на линиях порта P0 не совпадут с аккумулятором.

Действие команд вызова процедур полностью аналогично командам безусловного перехода. Единственное отличие состоит в том, что эти команды сохраняют в стеке адрес возврата.

Команда возврата из подпрограммы RET восстанавливает из стека значение счетчика команд, а команда возврата из процедуры обработки

прерывания RETI кроме того разрешает прерывание обслуженного уровня. Команды RET и RETI не различают какой командой, LCALL или ACALL, подпрограмма была вызвана, т. к. и в том и в другом случае в стеке сохраняется полный 16 разрядный адрес возврата.

5.6 Общая характеристика команд

5.6.1 Арифметические команды

В наборе команд имеются следующие арифметические операции: сложение, сложение с учетом флага переноса, вычитание с заемом, инкрементирование, декрементирование, сравнение, десятичная коррекция, умножение и деление.

В АЛУ производятся действия над целыми числами без знака. В двухоперандных операциях: сложение (ADD), сложение с переносом (ADDC) и вычитание с заемом (SUBB) аккумулятор является первым операндом и принимает результат операции. Вторым операндом может быть рабочий регистр выбранного банка рабочих регистров, регистр внутренней памяти данных с косвенно-регистровой и прямой адресацией или байт непосредственных данных. Указанные операции влияют на флаги: переполнения, переноса, промежуточного переноса и флаг четности в слове состояния процессора (PSW).

Использование разряда переноса позволяет многократно повысить точность при операциях сложения (ADDC) и вычитания (SUBB).

Выполнение операций сложения и вычитания с учетом знака может быть осуществлено с помощью программного управления флагом переполнения (OV) регистра PSW. Флаг промежуточного переноса (AC) обеспечивает выполнение арифметических операций в двоично-десятичном коде.

Операции инкрементирования и декрементирования на флаги не влияют.

Операции сравнения не влияют ни на операнд назначения, ни на операнд источника, но они влияют на флаги переноса.

Существуют три арифметические операции, которые выполняются только на аккумуляторе: две команды проверки содержимого аккумулятора А (JZ, JNZ) и команда десятичной коррекции при сложении двоично-десятичных кодов. При операции умножения содержимое аккумулятора А умножается на содержимое регистра В и результат размещается следующим образом: младший байт в регистре В, старший – в регистре А.

В случае выполнения операции деления целое от деления помещается в аккумулятор А, остаток от деления – в регистр В.

5.6.2 Команды битового процессора

Битовый процессор является частью архитектуры микроконтроллеров, и его можно рассматривать как независимый процессор побитовой обработки. Битовый процессор выполняет набор команд, имеет свое побитово-адресуемое ОЗУ и свой ввод-вывод.

Команды, оперирующие с битами, обеспечивают прямую адресацию 128 битов (0-127) в шестнадцати ячейках внутреннего ОЗУ (ячейки с адресами 20_H-2F_H) и прямую побитовую адресацию регистров специального назначения, адреса которых кратны восьми.

Каждый из отдельно адресуемых битов может быть установлен в "1", сброшен в "0", инвертирован, проверен. Могут быть реализованы переходы: если бит установлен; если бит не установлен; переход, если бит установлен, со сбросом этого бита; бит может быть перезаписан в (из) разряда переноса. Между любым прямоадресуемым битом и флагом переноса могут быть произведены логические операции "И", "ИЛИ", где результат заносится в разряд флага переноса. Команды побитовой обработки обеспечивают реализацию сложных функций комбинаторной логики и оптимизацию программ пользователя.

5.6.3 Логические команды с байтовыми переменными

Система команд позволяет реализовать логические операции: "И", "ИЛИ", "ИСКЛЮЧАЮЩЕЕ ИЛИ" на регистре-аккумуляторе (A) и байте-источнике. Вторым операндом (байтом-источником) при этом может быть рабочий регистр в выбранном банке рабочих регистров; регистр внутреннего ОЗУ, адресуемый с помощью косвенно-регистровой адресации; прямоадресуемые ячейки внутреннего ОЗУ и регистры специального назначения; непосредственная величина.

Указанные логические операции могут быть реализованы на любом прямоадресуемом регистре внутреннего ОЗУ или регистре специального назначения с использованием в качестве второго операнда содержимого аккумулятора (A) или непосредственных данных.

Существуют логические операции, которые выполняются только на аккумуляторе: сброс и инвертирование всех восьми разрядов аккумулятора (A); циклический сдвиг влево и вправо; циклический сдвиг влево и вправо с учетом флага переноса; обмен местами старшей и младшей тетрады (ниблов) внутри аккумулятора.

5.6.4 Команды передачи данных

Таблицы символов (кодов), зашитые в ПЗУ программ, могут быть выбраны с помощью команд передачи данных с использованием косвенной адресации. Байт константы может быть передан в аккумулятор из ячейки памяти программ, адресуемой суммой базового регистра (PC или DPTR) и индексного регистра (содержимого A). Это обеспечивает, например, удобное средство реализации алгоритма преобразования кода ASCII в семисегментный код.

Любая ячейка 256-байтового блока внешнего ОЗУ данных может быть выбрана с использованием косвенно-регистровой адресации через регистры указатели R0 или R1 (выбранного банка рабочих регистров).

Ячейка внутри адресного пространства 64 Кбайт внешнего ОЗУ также может быть выбрана с использованием косвенно-регистровой адресации через регистр-указатель данных DPTR.

Команды передачи между прямоадресуемыми регистрами позволяют заносить величину из порта в ячейку внутреннего ОЗУ без использования рабочих регистров или аккумулятора.

В логическом процессоре любой прямоадресуемый бит может быть помещен в бит переноса и наоборот.

Содержимое аккумулятора может быть обменено с содержимым рабочих регистров (выбранного банка) и с содержимым адресуемых с помощью косвенно-регистровой адресации ячеек внутреннего ОЗУ, а также с содержимым прямоадресуемых ячеек внутреннего ОЗУ и с содержимым регистров специального назначения.

Младший нибл (разряды 3-0) содержимого аккумулятора может быть обменен с младшим ниблом содержимого ячеек внутреннего ОЗУ, выбираемых с помощью косвенно-регистровой адресации.

5.6.5 Команды ветвления и передачи управления

Адресное пространство памяти программ не имеет страничной организации, что позволяет свободно перемещать фрагменты программы внутри адресного пространства, при этом не требуется перезасылка (изменение) номера страницы.

Перемещение отдельных фрагментов программы обеспечивает возможность использования перемещаемых программных модулей различными программами.

Команды 16-разрядных переходов и вызовов подпрограмм позволяют осуществлять переход в любую точку адресного пространства памяти программ объемом 64 Кбайт.

Команды 11-разрядных переходов и вызовов подпрограмм обеспечивают переходы внутри программного модуля емкостью 2 Кбайт. В системе команд имеются команды условных и безусловных переходов отно-

сительно начального адреса следующей программы в пределах от минус 128 до плюс 127. Эти переходы унаследованы от семейства MCS48. Система команд микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У дополнительно выполняет "длинные" переходы с использованием 16-разрядного адреса. Группа команд передачи управления включает четыре команды безусловного перехода без возврата ("ультракороткая" SJMP, "короткая" AJMP, "длинная" LJMP и переход по значению A+DPTR в команде JMP), две команды безусловного перехода с возвратом ("короткая" ACALL и "длинная" LCALL), восемь команд условного перехода без возврата. Имеются две команды возврата: из обычной подпрограммы команда RET и из подпрограммы обработки прерывания команда RETI. Завершает эту группу и всю систему команд NOP – "нет операции". Команды передачи управления значения флагов в регистре PSW не изменяют, кроме команд условного перехода CJNE, влияющих на значение флага C. Команды проверки отдельных разрядов позволяют осуществлять условные переходы по состоянию "0" или "1" прямоадресуемых битов. Команды проверки содержимого аккумулятора на (ноль/не ноль) позволяют осуществлять условные переходы по содержимому A.

Косвенно-регистровые переходы в системе команд обеспечивают ветвление относительно базового регистра (содержимого DPTR или PC) со смещением, находящимся в аккумуляторе A.

В заключение следует отметить, что большинство ассемблеров допускают обобщенную мнемонику JMP для команд безусловного перехода и CALL для команд вызова подпрограмм. Конкретный тип команды определяется ассемблером исходя из степени "длины" перехода или вызова.

Полный перечень команд приведен разделе 6 "Описание системы команд микроконтроллера".

6 Описание системы команд микроконтроллера

Система команд микроконтроллера предоставляет большие возможности обработки данных, обеспечивает реализацию логических, арифметических операций, а также управление в режиме реального времени. Реализована побитовая, потетрадная (4 бита), побайтовая (8 бит) и 16-разрядная обработка данных.

Основные функциональные блоки микроконтроллера: АЛУ, ОЗУ, регистры специального назначения, ПЗУ и внешние шины имеют байтовую организацию. Двухбайтовые данные используются только регистром-указателем (DPTR) и счетчиком команд (PC). Следует отметить, что регистр-указатель данных может быть использован как двухбайтовый регистр DPTR или как два однобайтовых регистра специального назначения DPH и DPL. Счетчик команд всегда используется как двухбайтовый регистр.

Набор команд микроконтроллера имеет 42 мнемонических обозначения команд для конкретизации 33 функций этой системы.

Синтаксис большинства команд ассемблерного языка состоит из мнемонического обозначения функции, вслед за которым идут операнды, указывающие методы адресации и типы данных. Различные типы данных или режимы адресации определяются установленными операндами, а не изменениями мнемонических обозначений.

Систему команд условно можно разбить на пять групп:

- арифметические команды,
- логические команды,
- команды передачи данных,
- команды битового процессора.
- команды ветвления и передачи управления,

Существуют следующие типы адресации операндов-источников:

- регистровая адресация,
- прямая адресация,

- косвенно-регистрационная адресация,
- непосредственная адресация.

Таблица 6.1 – Обозначения и символы, используемые в системе команд

Обозначение, символ	Назначение
1	2
A	Аккумулятор
Rn	Регистры текущего выбранного банка регистров
r	Номер загружаемого регистра, указанного в команде
ad	8-битовый внутренний адрес ячейки данных (direct), который может быть ячейкой внутреннего ОЗУ данных (0-127) или SFR (128-255)
rel	8-разрядная константа, определяющая относительный адрес перехода
@Ri, где i=0, 1	Косвенно адресуемая 8-битовая ячейка внутреннего ОЗУ данных
#d	8-разрядная константа, входящее в КОП (#data)
#d16	16-разрядная константа, входящая в КОП (#data16)
ad11	11-битовый адрес назначения (addr11) в командах LCALL, LJMP
ad16	16-битовый адрес назначения (addr16) в командах LCALL, LJMP
bit	Бит с прямой адресацией, адрес которого содержит КОП, находящийся во внутреннем ОЗУ данных или SFR
a ₁₅ , a ₁₄ ...a ₀	Биты адреса назначения
(X)	Содержимое элемента X
((X))	Содержимое по адресу, хранящемуся в элементе X
(X)[M]	Разряд M элемента X
+ - * / AND OR XOR NOT	Операции: сложения вычитания умножения деления логического умножения (операция И) логического сложения (операция ИЛИ) сложения по модулю 2 (исключающее ИЛИ) инверсия

Мнемонические обозначения функций однозначно связаны с конкретными комбинациями способов адресации и типами данных. Всего в системе команд возможно 111 таких сочетаний. В таблице 6.2 приведен перечень команд, упорядоченных по алфавиту.

Таблица 6.2 – Перечень команд микроконтроллеров

Мнемоника	Функция	Флаги
1	2	3
Команда ACALL ad11	Абсолютный вызов подпрограммы	
Команда ADD A, <байт-источник>	Сложение	AC, C, OV
Команда ADDC A, <байт-источник>	Сложение с переносом	AC, C, OV
Команда AJMP ad11	Абсолютный переход	
Команда ANL <байт-назначения>, <байт-источника>	Логическое "И"	
Команда ANL C, <байт-источника>	Логическое "И" для переменных-битов	C
Команда CJNE <байт-назначения>, <байт-источник>, <смещение>	Сравнение и переход, если не равно	C
Команда CLR A	Сброс аккумулятора	
Команда CLR bit	Сброс бита	C, bit
Команда CPL A	Инверсия аккумулятора	
Команда CPL bit	Инверсия бита	C, bit
Команда DA A	Десятичная коррекция аккумулятора для сложения	AC, C
Команда DEC <байт>	Декремент	
Команда DIV AB	Деление	C, OV
Команда DJNZ <байт>, <смещение>	Декремент и переход, если не равно нулю	
Команда INC <байт>	Инкремент	
Команда INC DPTR	Инкремент указателя данных	
Команда JB bit, rel	Переход, если бит установлен	
Команда JBC bit, rel	Переход, если бит установлен и сброс этого бита	
Команда JC rel	Переход, если перенос установлен	

Продолжение таблицы 6.2

1	2	3
Команда JMP @A+DPTR	Косвенный переход	
Команда JNB bit , rel	Переход, если бит не установлен	
Команда JNC rel	Переход, если перенос не установлен	
Команда JNZ rel	Переход, если содержимое аккумулятора не равно нулю	
Команда JZ rel	Переход, если содержимое аккумулятора равно 0	
Команда LCALL ad16	Длинный вызов	
Команда LJMP ad16	Длинный переход	
Команда MOV <байт-назначения>, <байт-источника>	Переслать переменную-байт	
Команда MOV DPTR,#d16	Загрузить указатель данных 16-битной константой	
Команда MOVC A, @A+DPTR	Переслать байт из памяти программ	
Команда MOVX <байт приемника>, <байт источника>	Переслать во внешнюю память (из внешней памяти) данных	
Команда MUL AB	Умножение	C, OV
Команда NOP	Нет операции	PC
Команда ORL <байт-назначения>, <байт-источника>	Логическое "ИЛИ" для переменных-байтов	
Команда ORL C, <бит источника>	Логическое "ИЛИ" для переменных-битов	C
Команда POP ad	Чтение из стека	
Команда PUSH ad	Запись в стек	
Команда RET	Возврат из подпрограммы	
Команда RETI	Возврат из прерывания	
Команда RL A	Сдвиг содержимого аккумулятора влево	
Команда RLC A	Сдвиг содержимого аккумулятора влево через флаг переноса	
Команда RR A	Сдвиг содержимого аккумулятора вправо	
Команда RRC A	Сдвиг содержимого аккумулятора вправо через флаг переноса	C

Окончание таблицы 6.2

1	2	3
Команда SETB bit	Установить бит	C
Команда SJMP rel	Короткий переход	
Команда SUBB A, <байт источника>	Вычитание с заемом	AC, C, OV
Команда SWAP A	Обмен тетрадами внутри аккумулятора	
Команда XCH A, <байт>	Обмен содержимого аккумулятора с переменной-байтом	
Команда XCHD A, @Ri , где i=0, 1	Обмен тетradой	
Команда XRL <байт-назначения>, <байт-источника>	Логическое "ИСКЛЮЧАЮЩЕЕ ИЛИ" для переменных-байтов	

7 Адресация

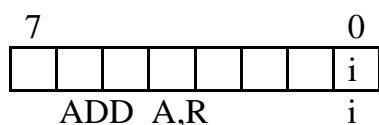
7.1 Косвенно-регистровая адресация

Косвенно-регистровая адресация используется для обращения к старшим 128 ячейкам внутреннего ОЗУ данных (128-256). В качестве регистров-указателей используются регистры R0, R1 выбранного банка регистров.

В командах PUSH и POP используется содержимое указателя стека (SP).

Косвенно-регистровая адресация используется также для обращения к внешней памяти данных. В этом случае с помощью регистров-указателей R0 и R1 (выбранного банка рабочих регистров) выбирается ячейка из блока в 256 байт внешней памяти данных. Номер блока предварительно задается содержимым порта P2.

Формат команды сложения ADD A,@Ri следующий:

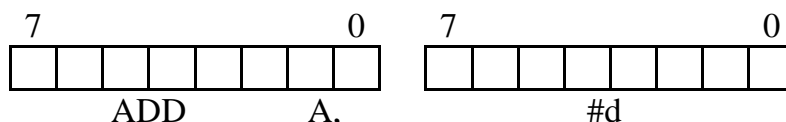


16-разрядный указатель данных (DPTR) может быть использован для обращения к любой ячейке адресного пространства внешней памяти данных объемом до 64 Кбайт.

7.2 Непосредственная адресация

Непосредственная адресация позволяет выбрать из адресного пространства памяти программ константы, явно указанные в команде. Операнд находится в отдельном байте команды, это константа. Таким образом, команды с использованием непосредственной адресации имеют минимум два байта, обращения за операндом к внешней памяти не требуется.

Формат команды сложения ADD A,#d следующий:



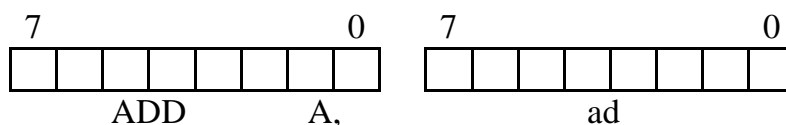
7.3 Прямая адресация

Прямая байтовая адресация используется для обращения к ячейкам внутренней памяти (ОЗУ) данных (0-127) и к регистрам специального назначения.

Прямая побитовая адресация используется для обращения к отдельно адресуемым 128 битам, расположенным в ячейках с адресами 20_H-2F_H и к отдельно адресуемым битам регистров специального назначения.

Старший бит байта кода прямого адреса выбирает одну из двух групп отдельно адресуемых битов, расположенных в ОЗУ или регистрах специального назначения. Прямо адресуемые биты с адресами 0-127 (00_H-7F_H) расположены в блоке из 16 ячеек внутреннего ОЗУ, имеющих адреса 20_H-2F_H . Указанные ячейки последовательно пронумерованы от младшего бита младшего байта до старшего бита старшего байта. Отдельно адресуемые биты в регистрах специального назначения пронумерованы следующим образом: пять старших разрядов адреса совпадают с пятью старшими разрядами адреса самого регистра, а три младших – определяют местоположение отдельного бита внутри регистра.

Формат команды сложения ADD A,ad следующий:

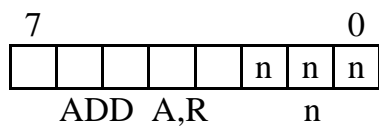


7.4 Регистровая адресация

Регистровая адресация используется для обращения к восьми рабочим регистрам выбранного банка рабочих регистров (эти же регистры могут быть выбраны с помощью прямой адресации и косвенно-регистровой адресации как обычные ячейки внутреннего ОЗУ данных).

Регистровая адресация используется для обращения к регистрам А, В, АВ (сдвоенному регистру), DPTR и к флагу переноса С. Использование регистровой адресации позволяет получать двухбайтовый эквивалент трехбайтовых команд прямой адресации.

Формат команды сложения ADD A,Rn следующий:



8 Особенности микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У по сравнению с ИС Н1830BE31, Н1830BE51

Основными особенностями микроконтроллеров являются:

- дополнительные специальные функциональные регистры;
- Таймер 2, имеющий режимы захвата (фиксации), таймера/счетчика со счетом на увеличение и уменьшение, программируемого синхросигнала, генератора скорости передачи в бодах;
- программируемый массив счетчиков, 5 каналов;
- расширенное мажоритированное ОЗУ;
- 7 источников прерывания;
- расширенный режим снижения потребляемой мощности;
- флаг отключения питания;
- режим ONCE;
- повышенная стойкость к специальным видам воздействий.

8.1 Внутренняя память данных

Микроконтроллеры имеют 256 байт внутреннего ОЗУ. Внутреннее ОЗУ мажоритированное, с тройным резервированием. Младшие 128 байт аналогичны внутреннему ОЗУ микроконтроллеров Н1830BE31, Н1830BE51. Старшие 128 байт занимают адресное пространство, параллельное пространству специальных функциональных регистров. Это означает, что они имеют те же адреса, что и регистры, но физически они являются независимыми от SFR и расположены в другой части кристалла микросхемы.

При выполнении команды обращения к внутреннему ОЗУ с адресами выше $7F_H$ CPU по режиму адресации, использованному в команде, микроконтроллер узнает, к чему именно это обращение: к старшим 128 байтам ОЗУ или к пространству SFR. Команды с прямой адресацией обеспечивают доступ к пространству SFR. Например, команда

`MOV 0A0H, #d ;(прямая адресация)`
обращается к SFR, расположенному по адресу 0A0_H (т. е. к P2). Команда с косвенно-регистровой адресацией обращается к старшим 128 байтам ОЗУ, например

`MOV @R0, #d ;(косвенная-регистровая адресация)`
где R0 содержит 0A0_H, обращается к байту с адресом 0A0_H в пространстве старших 128 байт ОЗУ. Следует отметить, что работа стека организована с применением косвенной адресации (адрес используемой под стек ячейки памяти размещен в регистре SP), так что старшие 128 байт данных ОЗУ прекрасно подходят для размещения в них стека микроконтроллера.

8.2 Пространство регистров специальных функций

Карта встроенной матрицы памяти, называемая пространством регистров специальных функций SFR, дана в таблице 8.1. Как видно из таблицы, появление новых функциональных блоков увеличило количество регистров специальных функций. К набору регистров добавлены регистры данных, перезагрузки и управления Таймера 2, регистры расширенного последовательного порта, дополнительный регистр прерываний, регистры управления блока программируемого массива счетчиков PCA. Ячейки, соответствующие незанятым адресам, физически отсутствуют на кристалле микросхемы. Чтение при обращении к этим адресам возвращает случайные данные, запись в такие ячейки также даст неопределенный результат.

Пользовательские программы не должны записывать единицы в эти незанятые позиции, т. к. они будут использованы в новых разработках. Гарантируется, что нулевые значения этих битов всегда будут означать отключение вновь введенных функций. Последнее означает, что программы, рассчитанные на использование в микроконтроллерах, не имеющих этих новых функций, будут корректно работать и в новых из-

делях, если они не активируют упомянутые биты. Дополнительно в таблице 8.1 указано побитное состояние каждого регистра SFR после старта.

Таблица 8.1 – Регистры специальных функций микроконтроллеров

Адрес	Регистры SFR и состояние их битов после сброса						
0F8 _H		CH 00000000	CCAP0H xxxxxxxx	CCAP1H xxxxxxxx	CCAP2H xxxxxxxx	CCAP3H xxxxxxxx	CCAP4H xxxxxxxx
0F0 _H	B 00000000						
0E8 _H		CL 00000000	CCAP0L xxxxxxxx	CCAP1L xxxxxxxx	CCAP2L xxxxxxxx	CCAP3L xxxxxxxx	CCAP4L xxxxxxxx
0E0 _H	ACC 00000000						
0D8 _H	CCON 00x00000	CMOD 00xxx000	CCAPM0 x0000000	CCAPM1 x0000000	CCAPM2 x0000000	CCAPM3 x0000000	CCAPM4 x0000000
0D0 _H	PSW 00000000						
0C8 _H	T2CON 00000000	T2MOD 00000000	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000	
0C0 _H							
0B8 _H	IP x0000000	SADEN 00000000					
0B0 _H	P3 11111111						
0A8 _H	IE 00000000	SADDR 00000000					
0A0 _H	P2 11111111						
098 _H	SCON 00000000	SBUF xxxxxxxx					
090 _H	P1 11111111						
088 _H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	
080 _H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 00xx00000

8.3 Таймер 2

Таймер 2 – это 16-битный таймер/счетчик, способный работать и как таймер, и как счетчик событий. Выбор производится битом C/T2# в SFR T2CON (таблица 8.2). Таймер может работать в трех различных режимах: режиме захвата (фиксации), режиме автоперезагрузки (при этом направление счета может быть как вверх, так и вниз, т. е. на увеличение

или уменьшение содержимого {TH2, TL2}) и в режиме генератора скорости передачи в бодах. Режимы выбираются битами в T2CON, как показано в таблице 8.3.

Таблица 8.2 – Регистр управления Таймера 2 T2CON

Бит	Обозначение	Функция
1	2	3
T2CON.7	TF2	Флаг переполнения таймера/счетчика 2. Сброс его осуществляется программным путем. Флаг не устанавливается, если либо RCLK либо TCLK установлены в единицы
T2CON.6	EXF2	Внешний флаг таймера/счетчика 2. Устанавливается при защелкивании информации в таймерных регистрах или при перезагрузке, происходящих под воздействием перехода из 1 в 0 на выводе P1.1 и при EXEN2 = 1. Сбрасывается только программным путем. При разрешенном прерывании от Таймера 2 вызывает подпрограмму обработки прерывания (если бит T2MOD.0 = 0).
T2CON.5	RCLK	Выбор Таймера 2 для задания скорости работы приемника в режимах 1 и 3. При RCLK = 1 используется Таймер 2, иначе – Таймер 1
T2CON.4	TCLK	Выбор Таймера 2 для задания скорости работы передатчика в режимах 1 и 3. При TCLK = 1 используется Таймер 2, иначе – Таймер 1
T2CON.3	EXEN2	Разрешение работы от внешнего сигнала. При EXEN2 = 1 перепад из 1 в 0 на входе P1.1 вызывает защелкивание информации или перезагрузку таймера/счетчика 2. При EXEN2 = 0 таймер/счетчик игнорирует сигналы на входе P1.1
T2CON.2	TR2	Запуск/остановка Таймера 2. TR2 = 1 запускает таймер/счетчик 2
T2CON.1	C/T2#	Бит выбора режима работы таймера/счетчика 2. При установке бита в 1 таймер/счетчик функционирует как счетчик перепадов из 1 в 0 на входе P1.0. При установке бита в 0 работает таймер
T2CON.0	CP/RL2#	Выбор захвата/перезагрузки. Если бит установлен в 1, то при EXEN2 = 1 перепад из 1 в 0 на P1.1 вызовет фиксацию содержимого TL2, TH2 в RCAP2L, RCAP2H. При нулевом бите таймер/счетчик работает в режиме автоперезагрузки при переполнении счетчика или при обнаружении перепада из 1 в 0 на P1.1 (последнее при EXEN2 = 1)

Примечание – При сбросе все биты регистра T2CON устанавливаются в 0.

Таблица 8.3 – Режимы работы Таймера 2

RCLK и TCLK	CP/RL2#	TR2	Режим
0 и 0	0	1	16-битный таймер-счетчик с перезагрузкой
0 и 0	1	1	16-битный таймер-счетчик с захватом текущего значения
1 или 1	X	1	Генератор приемопередатчика последовательного порта
X	X	0	Таймер 2 выключен

Таймер 2 состоит из двух 8-битных регистров TH2 и TL2, составляющих 16-разрядный регистр, и управляющей логики.

В режиме таймера его 16-разрядный регистр {TH2, TL2} инкрементируется в каждом машинном цикле. Так как каждый машинный цикл состоит из 12 периодов колебаний, скорость счета равна 1/12 частоты генератора. По сути дела в этом режиме происходит подсчет таймером выполненных машинных циклов.

В режиме счетчика регистр инкрементируется в ответ на перепад из 1 в 0 на соответствующем контакте внешнего ввода P1.0 (вход T2). В этом режиме внешний ввод анализируется в момент S5P2 каждого машинного цикла. Когда анализ показывает наличие единичного уровня в одном цикле и нулевого в следующем, содержимое счетчика инкрементируется.

8.3.1 Режим захвата (фиксации)

В режиме захвата есть два подрежима, выбираемых битом EXEN2 в T2CON. Если EXEN2 = 0, то Таймер 2 – это 16-битный таймер или счетчик, при переполнении которого устанавливается бит TF2 в T2CON. Битом C/T2# выбирается режим таймера или счетчика (см. таблицу 8.2). Бит TF2 затем может использоваться для вызова прерывания. Если EXEN2 = 1, то Таймер 2 продолжает делать то же самое, но с тем дополнением, что перепад из 1 в 0 на внешнем вводе P1.1 (сигнал T2EX) вызывает захват текущих значений TH2 и TL2 в RCAP2H и RCAP2L соответственно. Кроме того, этот перепад вызывает установку в 1 бита EXF2 в

T2CON. Бит EXF2, как и TF2, может вызвать прерывание. Работа Таймера 2 в режиме захвата показана на рисунке 8.1.

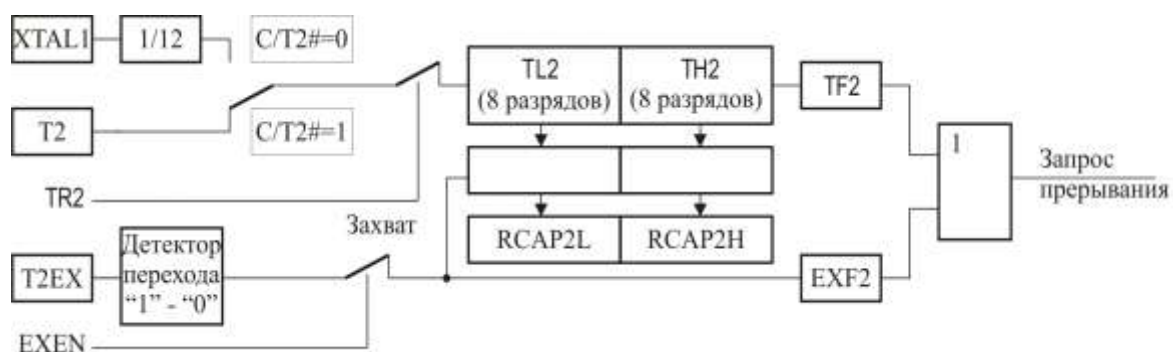


Рисунок 8.1 – Работа Таймера 2 в режиме захвата (фиксации)

8.3.2 Режим автоперезагрузки

Если Таймер 2 – в режиме 16-битного таймера/счетчика с автоперезагрузкой, то он может быть настроен на счет вверх или вниз (увеличение или уменьшение содержимого регистров TL2, TH2). Этот режим вызывается установкой в 1 бита DCEN (Down Counter Enable), расположенного в T2MOD (см. таблицу 8.4). При сбросе бит DCEN устанавливается в 0, так что Таймер 2 по умолчанию считает вверх. Когда DCEN установлен, Таймер 2 может считать вверх или вниз в зависимости от значения на выводе P1.1 (T2EX). Работа Таймера 2 в режиме автоперезагрузки показана на рисунке 8.2.

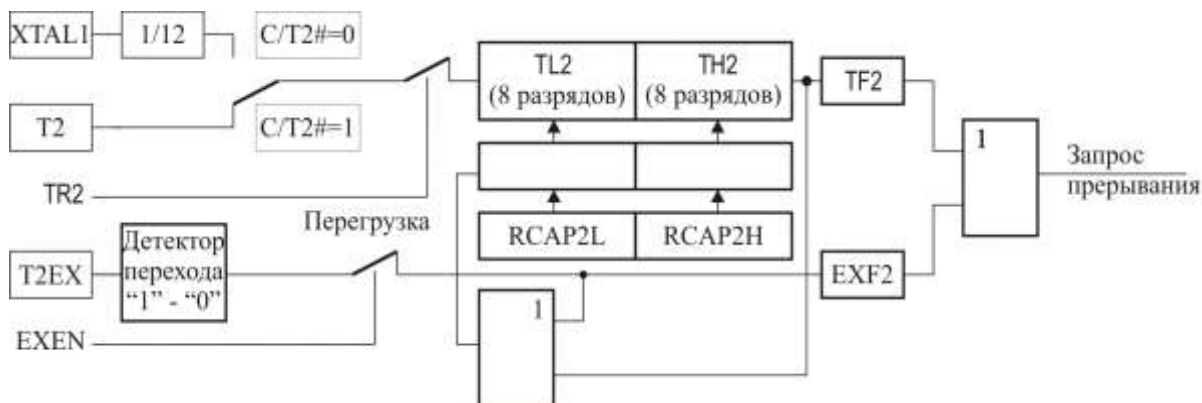


Рисунок 8.2 – Работа Таймера 2 в режиме автоперезагрузки

Таблица 8.4 – Регистр режима Таймера 2 T2MOD

Бит	Обозначение	Функция
T2MOD.7	-	Не используется
T2MOD.6	-	Не используется
T2MOD.5	-	Не используется
T2MOD.4	-	Не используется
T2MOD.3	-	Не используется
T2MOD.2	-	Не используется
T2MOD.1	T2OE	Бит разрешения выдачи сигнала программируемой частоты от Таймера 2 на вывод P1.0
T2MOD.0	DCEN	Бит разрешения счета Таймера 2 в режиме автоперезагрузки. При DCEN = 1 счет разрешен в прямом и обратном направлении в зависимости от уровня сигнала (T2EX) на выводе P1.1, при DCEN = 0 счет только в прямом направлении
Примечание – Биты регистра T2MOD не адресуются непосредственно командами работы с битами. По сбросу T2MOD.0 и T2MOD.1 устанавливаются в 0, значение остальных бит неопределено.		

Режим Таймера 2, считающего на увеличение при DCEN = 0. В этом режиме имеются 2 подрежима, выбираемых битом EXEN2 в T2MOD. Если EXEN2 = 0, то Таймер 2 считает вверх до 0FFFF_H и затем устанавливает в 1 бит переполнения TF2. Переполнение вызывает также перезагрузку регистров таймера 16-битным значением в RCAP2H и RCAP2L. Значения RCAP2H и RCAP2L предварительно должны быть установлены программно. При EXEN2 = 1 по перепаду из 1 в 0 на входе T2EX (вывод P1.1) или при переполнении включается 16-битная перезагрузка. Бит TF2 устанавливается при переполнении Таймера 2, а бит EXF2 устанавливается при перепаде из 1 в 0 на входе T2EX. Эти биты могут вызвать прерывание Таймера 2, если прерывание разрешено.

Установка бита DCEN в 1 переводит Таймер 2 в режим счета в прямом и обратном направлении в режиме автоперезагрузки. В этом режиме вывод P1.1 (сигнал T2EX) управляет направлением счета. Единичный уровень на P1.1 (сигнал T2EX) заставляет Таймер 2 считать в пря-

мом направлении. Таймер переполняется по достижении значения $0FFFF_H$ и устанавливает в 1 бит TF2. Это переполнение также вызовет перезагрузку 16-битного значения из RCAP2H и RCAP2L в регистры таймера TH2 и TL2 соответственно. При нулевом уровне на P1.1 счет идет в обратном направлении, при достижении значения, занесенного в регистры RCAP2H и RCAP2L, устанавливается флаг прерывания TF2 и осуществляется перезагрузка регистров TH2 и TL2 16-разрядным значением $0FFFF_H$.

8.3.3 Режим сигнала программированной частоты

У микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У на линию P1.0 в качестве второй альтернативной функции может быть осуществлен вывод сигнала программируемой частоты из Таймера 2 (выход T2). Для этого бит C/T2# (T2CON.1) должен быть очищен, а бит T2OE (T2MOD.1) установлен. Бит TR2 (T2CON.2) запускает и останавливает Таймер 2. Частота формируемого сигнала на выводе P1.0 определяется по формуле:

$$F_{OUT} = f_{РЕЗ} / [4 \times (65536 - \{RCAP2H, RCAP2L\})],$$

где $f_{РЕЗ}$ – частота резонанса кварцевого резонатора или частота внешнего тактового сигнала. Работа Таймера 2 в режиме сигнала программируемой частоты показана на рисунке 8.3.

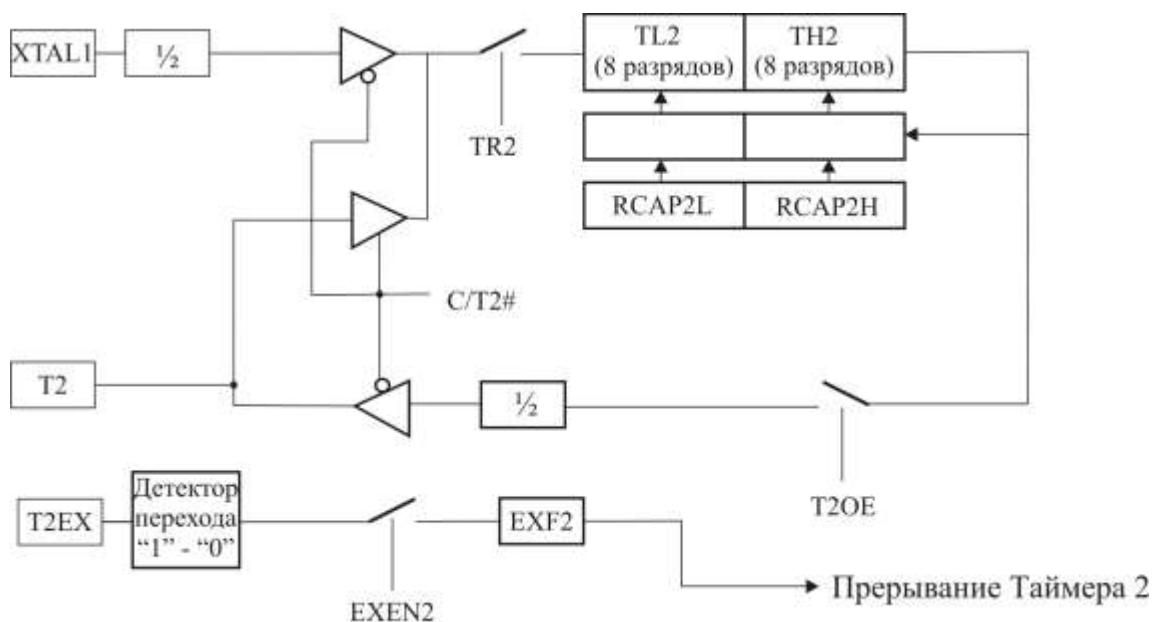


Рисунок 8.3 – Работа Таймера 2 в режиме сигнала программируемой частоты

8.3.4 Режим задающего генератора последовательного порта

В режиме задающего генератора (биты RCLK=1 или TCLK=1 регистра T2CON) приемник или передатчик последовательного порта тактируются синхроимпульсами, формируемыми при переполнении счетного 16-разрядного регистра Таймера 2 (рисунок 8.4). При C/T2# = 0 скорость работы последовательного порта (частота синхронизации) определяется формулой:

$$f_{1,3} = f_{\text{РЕЗ}} / [32 \times (65536 - \{ \text{RCAP2H}, \text{RCAP2L} \})].$$

Значение регистров RCAP2H, RCAP2L для нужной частоты синхронизации последовательного порта можно получить по формуле:

$$\{ \text{RCAP2H}, \text{RCAP2L} \} = 65536 - f_{\text{РЕЗ}} / (32 \times f_{1,3}).$$

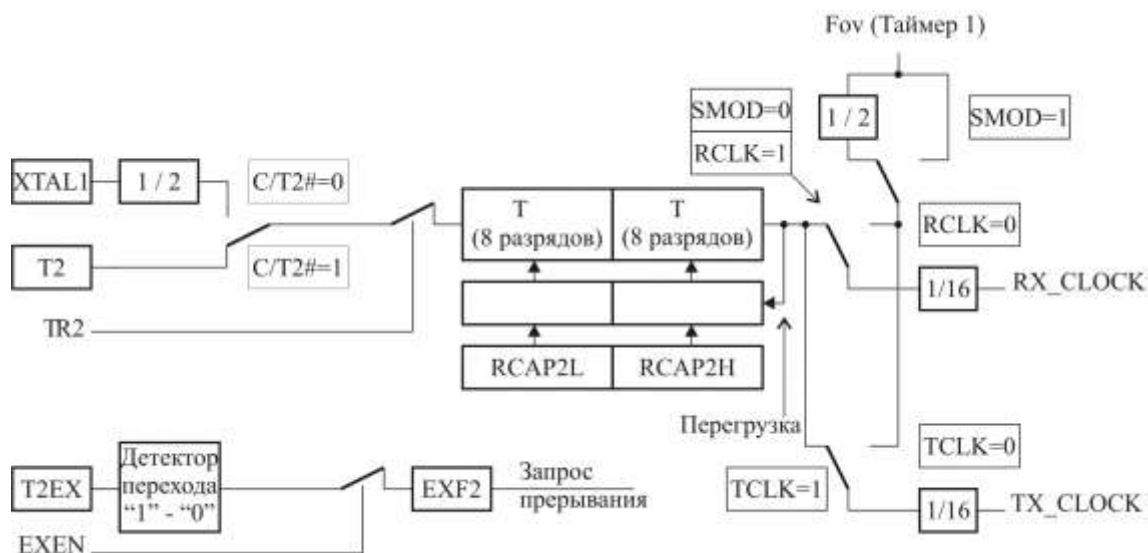


Рисунок 8.4 – Работа Таймера 2 в режиме задающего генератора

8.4 Программируемый массив счетчиков PCA

Основной особенностью микроконтроллеров является наличие блока PCA (programmable counter array – программируемый массив счетчиков). Это блок ввода-вывода, предназначенный для выполнения различных операций счета и определения временных интервалов, в том числе при широтно-импульсной модуляции. В таблице 8.1 приведён набор регистров специальных функций микроконтроллера. Блок PCA состоит из 16-разрядного таймера/счетчика и пяти 16-разрядных модулей фиксации-сравнения. Поэтому символ x в обозначениях регистров может принимать значения от 0 до 4. Таймер/счетчик является источником временной базы и счетчиком событий, значения его текущего отсчета передаются в модули фиксации-сравнения. Счетный регистр таймера/счетчика состоит из пары регистров специальных функций CH-CL, а значения, зафиксированные модулями, хранятся в пяти парах регистров SSCAPxH/SSCAPxL. Управление работой и режимами таймера/счетчика осуществляется при помощи регистров специальных функций SMOD (регистр режима) и SCON (регистр управления). Режимы работы модулей фиксации-сравнения определяются пятью регистрами SSCAPMx. Форма-

ты регистров CMOD и CCON приведены в таблице 8.5. В таблице 8.6 приведено функциональное назначение битов регистров CMOD и CCON. Формат регистров CСАРМх приведен в таблице 8.7. Функциональное назначение битов регистров CСАРМх приведено в таблице 8.8.

Таблица 8.5 – Формат регистров CMOD и CCON

CMOD	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF
CCON	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0

Таблица 8.6 – Функциональное назначение битов регистров CMOD и CCON

Имя бита	Номер бита	Функция
CIDL	CMOD.7	Бит разрешения функционирования блока PCA в режиме Idle: CIDL = 1 – функционирование запрещено CIDL = 0 – функционирование разрешено
WDTE	CMOD.6	Бит разрешения функции сторожевого таймера модуля 4: WDTE = 1 – функция разрешена WDTE = 0 – функция запрещена
—	CMOD.4	Зарезервирован
—	CMOD.3	Зарезервирован
CPS1	CMOD.2	Бит выбора источника синхросигнала
CPS0	CMOD.1	Бит выбора источника синхросигнала
ECF	CMOD.0	Бит разрешения прерывания блока PCA: ECF = 1 разрешает прерывание по флагу CF ECF = 0 – прерывание запрещено
CF	CCON.7	Флаг переполнения таймера/счетчика PCA. Вызывает прерывание, если установлен флаг ECF регистра CMOD
CR	CCON.6	Бит включения таймера/счетчика PCA: CR = 1 – счетчик включен CR = 0 – счетчик выключен
—	CCON.5	Зарезервирован
CCF4	CCON.4	Флаг прерывания модуля 4
CCF3	CCON.3	Флаг прерывания модуля 3
CCF2	CCON.2	Флаг прерывания модуля 2
CCF1	CCON.1	Флаг прерывания модуля 1
CCF0	CCON.0	Флаг прерывания модуля 0

Таблица 8.7 – Формат регистров ССАРМх

Номер бита	7	6	5	4	3	2	1	0
Имя бита	—	ЕСОМх	САРРх	САРНх	МАТх	ТОГх	РWMх	ЕССFх

Таблица 8.8 – Функциональное назначение битов регистров ССАРМх

Имя бита	Номер бита	Функция
—	ССАМРх.7	Зарезервирован
ЕСОМх	ССАМРх.6	ЕСОМх = 1 разрешает выполнение функции сравнения
САРРх	ССАМРх.5	САРРх = 1 разрешает сравнение по положительному фронту
САРНх	ССАМРх.4	САРНх = 1 разрешает сравнение по отрицательному фронту
МАТх	ССАМРх.3	При МАТх = 1 равенство значений таймера/счетчика и регистра модуля устанавливает соответствующий флаг ССFх
ТОГх	ССАМРх.2	При ТОГх = 1 равенство значений таймера/счетчика и регистра модуля переключает уровень сигнала на соответствующем выходе СЕХх
РWMх	ССАМРх.1	Бит включения режима широтно-импульсной модуляции. При РWMх = 1 модуль работает в режиме ШИМ, модулированный сигнал выдается на выход СЕХх
ЕССFх	ССАМРх.0	Бит разрешения прерываний по флагу ССFх регистра ССОН. При ЕССFх = 1 прерывания разрешены

Выводы порта 1 (P1.2 ÷ P1.7) обеспечивают ввод-вывод для блока РСА в качестве альтернативных функций. В таблице 8.9 приведено функциональное назначение выводов порта 1 с альтернативными функциями.

Таблица 8.9 – Альтернативные функции порта 1

Имя бита	Номер вывода	Альтернативная функция
СЕХ4	P1.7	Модуль 4. Вход при фиксации, выход при сравнении и ШИМ
СЕХ3	P1.6	Модуль 3. Вход при фиксации, выход при сравнении и ШИМ
СЕХ2	P1.5	Модуль 2. Вход при фиксации, выход при сравнении и ШИМ
СЕХ1	P1.4	Модуль 1. Вход при фиксации, выход при сравнении и ШИМ
СЕХ0	P1.3	Модуль 0. Вход при фиксации, выход при сравнении и ШИМ
ЕСI	P1.2	Вход таймера/счетчика

Таймер/счетчик PCA и пять модулей сравнения имеют единственный вектор прерывания. Если бит ECF (регистр SMOD) разрешения прерывания от PCA установлен, то установка флага CF приводит к генерации запроса прерывания от блока PCA.

Пара регистров CH-CL работает как 16-разрядный таймер/счетчик. Выбранный вход наращивает регистр младшего байта CL. Через два периода синхросигнала после переполнения CL наращивается регистр старшего байта CH. При переполнении CH устанавливается флаг CF регистра SCON и, если установлен бит ECF регистра SMOD, формируется запрос прерывания. Функциональная схема таймера/счетчика PCA приведена на рисунке 8.5.

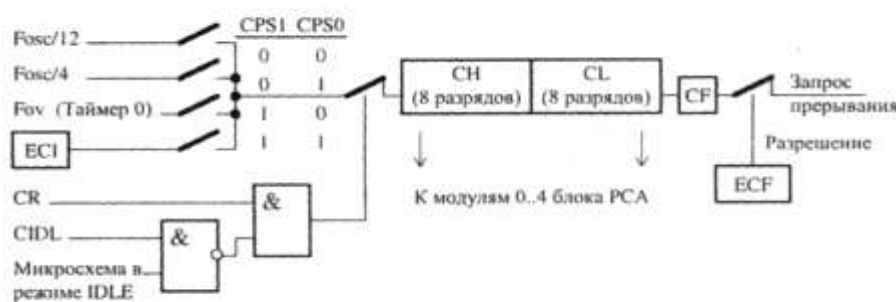


Рисунок 8.5 – Таймер/счетчик блока PCA

Биты CPS1 и CPS0 регистра SMOD выбирают один из следующих четырех сигналов в качестве входного для таймера/счетчика:

- Fosc/12. Сигнал активен в такте S5P2 каждого цикла;
- Fosc/4. Сигнал активен в тактах S1P2, S3P2 и S5P2 каждого цикла (в три раза чаще);
- переполнение Таймера 0. Регистр CL наращивается в такте S5P2 каждого цикла, если Таймер 0 переполнен. Это режим работы PCA с программируемой частотой на входе;

– внешний сигнал на линии P1.2/ECI. Центральный процессор проверяет вход ECI в тактах S1P2, S3P2 и S5P2 каждого цикла обмена. Если на линии ECI имел место переход "1"- "0", регистр CL наращивается. Максимальная частота переключений на входе ECI может быть $F_{osc}/8$.

Установка бита управления работой (CR в регистре CCON) включает таймер/счетчик PCA. Таймер/счетчик PCA продолжает работать в пассивном режиме, если не установлен бит CIDL регистра CMOD. Центральный процессор всегда может прочесть содержимое регистров CH и CL. Запись в эти регистры во время счета (когда установлен бит CR) запрещена.

8.4.1 Модули фиксации-сравнения блока PCA

Каждый из модулей фиксации-сравнения включает пару регистров CCAРxH/ССАРxL, 16-разрядный компаратор, логические вентили и селекторы сигналов. В регистрах запоминается время или значение счетчика, при котором внешнее событие произошло (фиксация) или должно произойти действие (сравнение). В режиме ШИМ регистр младшего байта управляет шириной выходного сигнала.

Конфигурация каждого модуля зависит от режима его работы. Каждый модуль может быть независимо запрограммирован на работу в одном из следующих режимов:

- фиксация 16-разрядного значения по положительному фронту, отрицательному фронту или по обоим фронтам сигнала на входе СЕХх;
- режимы сравнения: 16-разрядный программируемый таймер, 16-разрядный скоростной вывод, сторожевой таймер (только модуль 4), 8-разрядный ШИМ регулятор;
- нет операции.

Режим работы каждого модуля определяется комбинацией битов в регистрах режима ССАРМх. Возможные следующие комбинации приведены в таблице 8.10.

Таблица 8.10 – Режимы работы модулей PCA

Функция модуля	—	ECOM _x	CAPP _x	CAPN _x	MAT _x	TOG _x	PWM _x	ECCF _x
Фиксация значения по положительному фронту на входе СЕХ _x	x	x	1	0	0	0	0	x
Фиксация значения по отрицательному фронту на входе СЕХ _x	x	x	0	1	0	0	0	x
Фиксация значения по изменению значения на входе СЕХ _x	x	x	1	1	0	0	0	x
16-разрядный таймер	x	1	0	0	1	0	0	x
Скоростной вывод	x	1	0	0	1	1	0	x
Широтно-импульсный модулятор	x	1	0	0	0	0	1	0
Сторожевой таймер	x	1	0	0	1	x	0	x
Нет операции	x	0	0	0	0	0	0	0

Для работы модулей фиксации-сравнения необходима работа таймера/счетчика PCA. Он включается-выключается в соответствии со значением бита CR регистра SCON. Для запрещения работы любого модуля его следует перевести в режим "нет операции". При наличии события (фиксация, срабатывание таймера, скоростной вывод) устанавливается флаг CCF_x регистра SCON и формируется запрос прерывания от PCA (если в регистре CCAPM_x установлен бит разрешения). Центральный процессор всегда может прочитать или записать информацию в регистры CCAP_xH и CCAP_xL.

8.4.2 Режим фиксации блока PCA

Режим фиксации (рисунок 8.6) дает возможность измерять длительность импульсов, циклов, разность фаз по каждому из пяти входов. Входы СЕХ₀-СЕХ₄ проверяются на положительный или отрицательный перепад сигнала. Когда модуль фиксирует его наличие, он запоминает значение таймера/счетчика на этот момент в регистрах CCAP_xH/CCAP_xL и устанавливает флаг CCF_x регистра SCON. Разрешение составляет один период синхронизации таймера/счетчика.

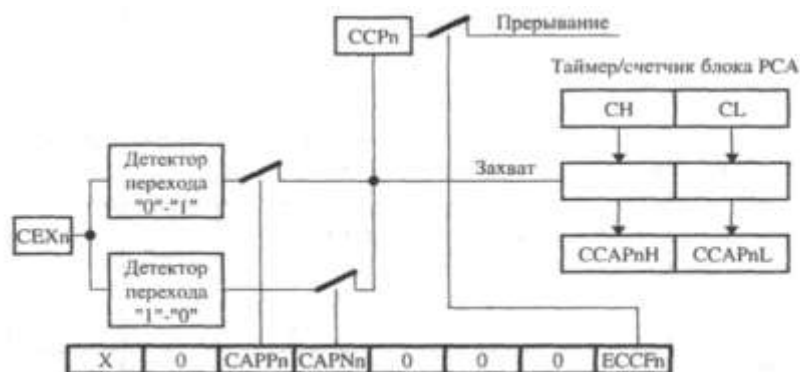


Рисунок 8.6 – Блок PCA в режиме фиксации

Если соответствующий бит разрешения ECCF_x регистра CCAPM_x установлен, блок PCA посылает соответствующий запрос прерывания.

Поскольку аппаратно при прерывании флаг события не очищается, пользователь должен сделать это программно. При следующем событии на этом же модуле значение таймера/счетчика в регистрах перезаписывается. Для сохранения зафиксированного значения его следует сохранить в ОЗУ в процессе обработки прерывания, пока не произошло следующее событие.

8.4.3 Режимы сравнения блока PCA

Функция сравнения обеспечивает четыре режима: режим 16-разрядного таймера, режим скоростного вывода, режим сторожевого таймера, режим ШИМ. В первых трех режимах модуль постоянно сравнивает содержимое таймера/счетчика со значением, загруженным предварительно в пару его регистров CCAP_xH/CCAP_xL. В режиме ШИМ модуль постоянно сравнивает содержимое младшего регистра таймера-счетчика CL со значением в его регистре CCAP_xL. Сравнение производится три раза за цикл обмена, т. е. с наибольшей возможной частотой ($F_{osc}/4$).

Функция сравнения для конкретного модуля выбирается установкой бита $ЕСОМx$ в регистре $ССАРМx$. Для использования модулей в режимах сравнения следует выполнить следующие шаги:

- выбрать режим работы модуля;
- выбрать входной сигнал для таймера/счетчика;
- загрузить значение эталона в пару регистров модуля;
- установить бит управления запуском таймера/счетчика;
- после прерывания очистить флаг события.

8.4.4 Режим программируемого 16-разрядного таймера

В этом режиме (рисунок 8.7) сравнивается текущее значение таймера/счетчика и предварительно загруженная в пару регистров $ССАРxH/ССАРxL$ величина. При совпадении устанавливается флаг события $ССFх$ регистра $ССОН$. Пользователь должен программно сбросить этот флаг при обработке прерывания. При обслуживании прерывания можно загрузить в регистры $ССАРxH/ССАРxL$ новое значение.

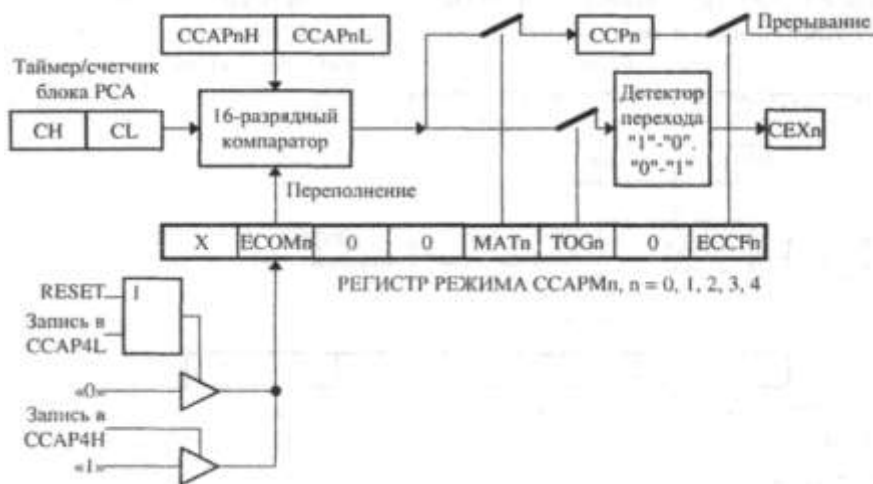


Рисунок 8.7 – Блок PCA в режиме 16-разрядного таймера и скоростного вывода

В процессе загрузки регистров рекомендуется вначале записывать данные в $ССАРxL$, а затем в $ССАРxH$. При записи в регистр младшего

байта очищается бит ECOMx, что запрещает выполнять сравнение. При записи в регистр старшего байта этот бит устанавливается, вновь разрешая сравнения. Эта последовательность защищает от ложных срабатываний.

8.4.5 Режим скоростного вывода

В режиме скоростного вывода (HSO) совпадение значений таймера/счетчика и величины, загруженной в регистры CCAPxH/CCAPxL, вызывает смену сигнала на выходе CEXx и установку флага CCFx (рисунок 8.7). Это обеспечивает более высокую точность, чем при программном переключении сигнала, т. к. переключение осуществляется до обслуживания запроса прерывания. Таким образом, интервал времени, связанный с обслуживанием прерывания, не вклинивается в диаграмму формирования выходного сигнала. Программно задавая уровень сигнала на выходе CEXx, пользователь задает тип перепада: из низкого уровня сигнала в высокий уровень или из высокого уровня сигнала в низкий. Флаг CCFx при обработке прерывания должен сбрасываться программно. Если в процедуре прерывания в регистры CCAPxH/CCAPxL новое значение не заносилось, то следующее совпадение произойдет через полный цикл таймера/счетчика.

8.4.6 Режим сторожевого таймера

В режиме сторожевого таймера (рисунок 8.8) модуль 4 блока PCA может быть запрограммирован на выполнение функции 16-разрядного сторожевого таймера (watchdog timer — WDT). В этом режиме при совпадении числа в таймере/счетчике с величиной, занесенной предварительно в регистры данных модуля, осуществляется сброс и инициализация микроконтроллера. Сброс микроконтроллера по истечении установленного времени является стандартным приемом выхода из зависаний программы управления.

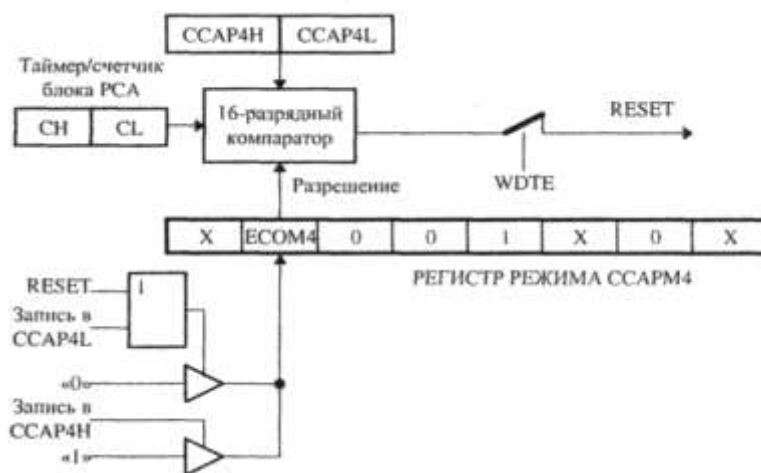


Рисунок 8.8 – Блок PCA в режиме сторожевого таймера

Чтобы перевести модуль 4 в режим сторожевого таймера, нужно установить биты ECOM4 и MAT4 регистра CCAPM4, а также бит WDTE регистра CMOD. Необходимо, также, выбрать нужный вход для таймера/счетчика с помощью битов CPS0, CPS1 регистра CMOD. Далее следует загрузить 16-разрядное число для сравнения в регистры CCAP4H, CCAP4L и 16-разрядное начальное значение в таймер/счетчик CH/CL (можно использовать значение 0000_H по сбросу). Разность между этими числами, умноженная на частоту входных импульсов PCA, определяет интервал времени, на который взведен сторожевой таймер.

Чтобы предотвратить сброс от сторожевого таймера PCA, имеется три возможности:

- периодически менять сравниваемое число в CCAP4H, CCAP4L так, чтобы совпадения не произошло;
- периодически менять значения в таймере/счетчике так, чтобы совпадения не произошло;
- очищать бит WDTE до совпадения, а затем вновь устанавливать его.

Второй вариант не рекомендуется применять, когда работают другие модули, поскольку все они пользуются временным отсчетом.

8.4.7 Режим широтно-импульсной модуляции

Все пять модулей блока PCA могут быть запрограммированы на режим широтно-импульсной модуляции ШИМ (рисунок 8.9). При этом на выходах CEXx выдаются модулированные сигналы, ширина импульсов которых определяется 8-разрядным разрешением. Это позволяет преобразовать цифровой код в аналоговый сигнал при помощи простой внешней схемы (например, интегрирующей цепочки).

В этом режиме младший байт таймера/счетчика (CL) постоянно сравнивается с содержимым регистра CCPxL. Если $CL < CCPxL$, то на выходе CEXx уровень сигнала низкий. При совпадении сигнал на выходе приобретает значение "1" и остается таким, пока счетчик не достигает конца (00_H). После этого выходной сигнал возвращается на низкий уровень, в регистр CCPxL загружается значение из регистра CCPxH и начинается новый цикл счета.

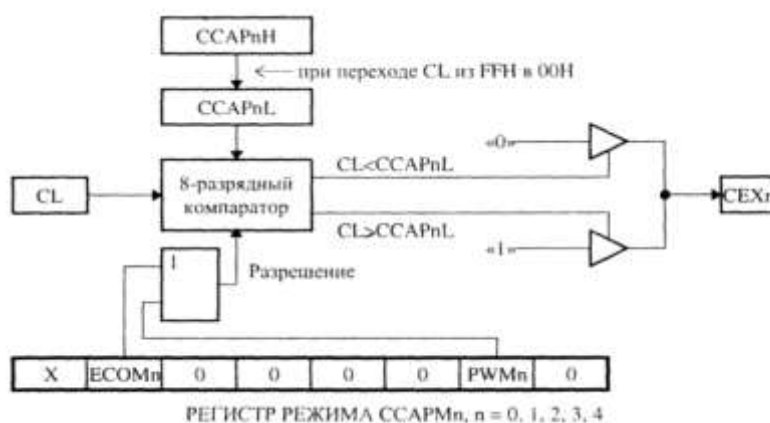


Рисунок 8.9 – Блок PCA в режиме широтно-импульсного модулятора

Число в ССАРxL определяет ширину импульса в текущем цикле, а число в ССАРxH определяет ширину импульса в следующем цикле. При ССАРxL = 0 ширина импульса составляет 100 %, а при ССАРxL = 225 она равна 0,4 %. Частота сигнала на выходе ШИМ равна частоте сигнала на входе таймера/счетчика, деленной на 256. Самая высокая частота имеет место при входе $F_{osc}/4$. Если $F_{osc} = 12$ МГц, то частота сигнала на выходе равна 11,7 кГц.

Для перевода модуля в режим ШИМ нужно установить биты ЕСОМx и РWМx регистра ССАРМx. Далее следует определить вход посредством комбинации битов СРС0, СРС1 регистра СМОД. Затем необходимо занести 8-разрядные числа в регистры ССАРxL и ССАРxH. Наконец следует установить бит СR управления запуском таймера/счетчика в регистре ССОН.

8.5 Система прерываний

Систему прерываний микроконтроллера образуют описанные ранее в подразделе 4.8 семь прерываний. По сравнению с микроконтроллерами Н1830ВЕ31, Н1830ВЕ51 добавлено два прерывания: от Таймера 2 и от блока программируемого массива счетчиков РСА. Флаги прерываний должны сбрасываться программным путем, аппаратного сброса они не имеют. Подпрограмма прерывания должна самостоятельно определить, какой из этих флагов установлен и сбросить его, выполняя после этого действия, предусмотренные в качестве реакции микроконтроллера на вызов прерывания.

Регистр приоритетов прерываний IP содержит 7 бит в соответствии с 7 источниками прерываний микроконтроллеров (см. таблицу 4.8). Каждый из этих битов является битом приоритета соответствующего прерывания. Таким образом, уровень каждого из прерываний определяется состоянием этих битов. Очевидно, при помощи одного бита можно закодировать 2 различных числа – 0_B и 1_B, что дает нам соответственно 2 уровня приоритета для каждого из источников – 0

и 1, причем нулевой уровень соответствует низшему приоритету, а первый – высшему.

8.6 Особенности приемопередатчиков в ИС 1830BE32У, 1830BE32АУ, 1830BE52У

Первая особенность приемопередатчиков – наличие устройства детектирования ошибок формата (Framing Error Detection). Это устройство позволяет последовательному порту проверять правильность стоп-битов в режимах 1, 2 или 3. Потеря стоп-бита может возникнуть, например, из-за шумов на линиях передачи или из-за одновременной передачи информации двумя процессорами мультипроцессорной системы.

Если бит остановки потерян, устанавливается бит ошибки формата (FE). Этот бит может быть проверен программно после каждого приема для обнаружения ошибок связи. Будучи однажды установлен, бит ошибки формата может быть сброшен только программным путем. Если после установки бита ошибки формата он не был сброшен программно, а затем в результате следующего цикла обмена информацией получен байт данных с правильным стоп-битом, сброса бита ошибки формата не произойдет.

Бит FE расположен в SCON и делит один адрес с SM0. Бит управления SMOD0 в регистре PCON (PCON.6) определяет, к какому биту обращается микропроцессор: к SM0 или к FE. Если SMOD0 = 0, то обращение – к SM0. Если SMOD0 = 1, то обращение – к FE.

Вторым отличием приемопередатчиков в ИС 1830BE32У, 1830BE32АУ, 1830BE52У от приемопередатчиков в ИС Н1830BE31, Н1830BE51 является автоматическое распознавание адреса. Автоматическое распознавание адреса уменьшает время, требуемое микроконтроллеру на обслуживание последовательного порта. Поскольку в этом случае CPU не прерывается при получении адресного байта, отличного от своего собственного адреса, издержки CPU на сравнение адресов исключаются. Эта особенность микро-

контроллера используется, как правило, в одном из 9-битовых режимов. При этом флаг прерывания от приемника (RI) устанавливается только тогда, когда полученный адресный байт совпадает с индивидуальным или с широковещательным адресом.

Эту особенность приемопередатчика можно использовать в мультипроцессорных системах следующим образом. Когда ведущий процессор "хочет" передать блок данных одному или нескольким подчиненным процессорам системы, он сначала посылает адресный байт, который идентифицирует нужный подчиненный компонент. Следует помнить, что в адресном байте 9-й бит установлен в 1, тогда как в байте данных 9-й бит установлен в 0. Во всех подчиненных процессорах их биты SM2 должны быть установлены в 1, так что они могут быть прерваны при приеме адресного байта. Способность автоматического распознавания адреса позволяет обеспечить прерывание только того подчиненного компонента, к которому адресуется ведущий процессор. В этом режиме сравнение адресов происходит аппаратно, а не программно. (В последовательном порту 1830BE31, 1830BE51 адресный байт прерывает все подчиненные процессоры для сравнения адресов).

Ведущий процессор может избирательно связываться и с группами подчиненных процессоров. Адресация ко всем подчиненным сразу или к какой-либо группе из них возможна при использовании адресов, хранящихся в двух специальных функциональных регистрах SADDR и SADEN.

Индивидуальные адреса подчиненных процессоров хранятся обычно в SADDR. SADEN – это масочный байт, определяющий незначащие биты для формирования заданного адреса. Эти незначащие биты придают гибкость определяемому пользователем протоколу для формирования адреса выбранного подчиненного процессора.

Ведущий процессор может также связываться со всеми подчиненными одновременно с помощью широковещательного адреса. Он форми-

руется логикой ИЛИ регистров SADDR и SADEN, при этом нули в SADEN по-прежнему определяют незначащие биты в адресном байте. Это также придает дополнительные возможности в выборе широковещательного адреса, но в большинстве применений широковещательным адресом выбирают $0FF_H$.

SADDR и SADEN расположены по адресам $0A9_H$ и $0B9_H$ соответственно. При сбросе регистры SADDR и SADEN устанавливаются в 00_H , что определяет заданный и широковещательный адреса как XXXX XXXX (все биты неопределенные). Это гарантирует совместимость сверху вниз последовательного порта.

8.7 Режим пониженного энергопотребления

Микроконтроллеры 1830BE32У, 1830BE32АУ, 1830BE52У имеют два режима энергопотребления, описанные в разделе 4. Микроконтроллеры выходят из режима пониженного энергопотребления как при аппаратном сбросе, так и при внешнем прерывании. Сброс переопределяет все регистры специальных функций, но не меняет содержимого внутреннего ОЗУ. Внешнее прерывание позволяет как регистрам (кроме бита PD в PCON), так и встроенному ОЗУ сохранять свои значения. Чтобы обеспечить правильный выход из этого режима, сброс или внешнее прерывание не должны подаваться прежде, чем U_{CC} восстановит свой нормальный рабочий уровень, и должны удерживаться достаточно долго, чтобы генератор перестартовал и стабилизировался (обычно менее 10 мс).

При использовании для вывода из режима пониженного энергопотребления внешних прерываний INT0# или INT1# схема прерываний должна быть конфигурирована на срабатывание по уровню соответствующего сигнала. Удержание вывода при нулевом потенциале перестартует генератор, а перевод вывода в единичный уровень завершит выход. После выполнения команды RETI в подпрограмме обслуживания прерывания следующей будет выполняться та команда, которая идет после той, что перевела микроконтроллер в режим пониженного энергопотребления.

8.8 Специальный режим эмуляции ONCE

Данный дополнительный режим позволяет выполнять тестирование и отладку систем, использующих микроконтроллеры без удаления последних из платы. Для активизации режима эмуляции необходимо выполнить следующую последовательность действий:

- 1 При высоких логических уровнях на RST и PSEN# подать уровень логического 0 на вывод ALE.

- 2 Удерживать ALE в состоянии логического 0 при деактивации сигнала сброса на RST (подаче на RST уровня логического 0), после чего перестать удерживать ALE в состоянии логического 0.

Во время нахождения микросхемы в режиме эмуляции выводы порта P0 переходят в высокоимпедансное состояние, а выводы портов P1-P3 и выводы ALE, PSEN# находятся в состоянии логической 1, которое обеспечивается с помощью внутренних высокоомных резисторов. Тракт генератора остается активным. В то время, когда микросхема находится в режиме эмуляции, для управления системой можно использовать внешний эмулятор или тестовый процессор. Нормальная работа схемы возобновляется после подачи нормального сигнала сброса на вход RST.

9 Описание инструментальных средств для микроконтроллеров

Инструментальные средства разработки и отладки систем на базе микроконтроллеров (ближайших аналогов разработанных микроконтроллеров, в данном случае H1830BE31, H1830BE51) поставляются многими фирмами.

Ниже приводится краткая информация о средствах отладки, поставляемых ООО "Фирма Фитон", г. Москва, обеспечивающего отладку систем на базе разработанных микроконтроллеров при условии модернизации жгута подключения (адаптера) под корпус H16.48-2В.

Реквизиты: ООО "Фирма ФИТОН", Россия, Москва, 123458, ул. Твардовского, д. 8, стр. 1, кор. Б, оф. 208, ТехноПарк «Строгино»

Тел/факс (495) 730-75-84 (многоканальный)

E-mail: PHYTON@phyton.ru

Полная конфигурация пакета описана ниже.

9.1 Интегрированный пакет разработки и отладки систем на базе микроконтроллеров семейства 8051

Пакет Project-52 – набор программно-аппаратных средств 4-го поколения, предназначенный для разработки и отладки систем на базе микроконтроллеров семейства 8051 фирмы Intel. Концепция Project-52 – объединение внутрисхемного эмулятора, программного отладчика-симулятора, компиляторов, текстового редактора, менеджера проектов и программатора в рамках единой интеллектуальной среды разработки. При наличии одного из программаторов PicProg+, ChipProg, ChipProg+ пакет поддерживает работу и с программатором. Программный интерфейс пакета унифицирован и поддерживает все этапы разработки про-

граммного обеспечения – от написания исходного текста программы до ее компиляции и отладки.

Пакет Project-52 ориентирован на отладку программ на языке высокого уровня по исходному тексту. Встроенные многооконный редактор, менеджер проектов и большое количество сервисных возможностей существенно облегчают труд разработчика, избавляя его от рутинных операций. Встроенный многооконный редактор предназначен для написания исходных текстов программ. Редактор поддерживает операции с блоками текста, поиск/замену, цветовое выделение синтаксических конструкций языка Си и ассемблера. Встроенный менеджер проектов поддерживает автоматическую компиляцию программ, написанных для компилятора Си и ассемблера.

Переход от редактирования исходного текста к отладке и обратно происходит прозрачно, т. е. менеджер проектов автоматически запускает компиляцию изменившихся исходных текстов, активизирует отладчик, осуществляет загрузку программ.

Полная конфигурация пакета называется Project-52/ESA и включает в себя:

- менеджер проектов,
- кросс-компилятор языка ассемблер MCA-51,
- отладчик-симулятор PDS-52,
- внутрисхемный эмулятор PICE-52.

9.2 Внутрисхемный эмулятор 8-разрядных микроконтроллеров семейства 8051

PICE-52 – эмулятор нового поколения, созданный с применением новых технологий разработки аппаратуры и программного обеспечения. Применение программируемых матриц большой емкости позволило значительно сократить размеры эмулятора без какого-либо ущерба его функциональным возможностям, свести к минимуму отличия в электрических и частотных характеристиках эмулятора от характеристик эмули-

руемого процессора и, тем самым, добиться максимальной точности эмуляции на частотах до 70 МГц при напряжениях питания от 1,8 В до 5,5 В. Перезагружаемая аппаратная структура эмулятора обеспечивает эмуляцию практически всех микроконтроллеров семейства Intel 8051. Программная поддержка PICE-52 работает в среде Windows-95/98/ME/NT/2000/XP и предоставляет пользователю обширный сервис как по разработке программ, так и по их отладке. Эмулятор состоит из основной платы размером (95×70×40) мм, сменного пода под определенную группу процессоров и сменного адаптера под конкретный тип корпуса. На основной плате реализованы трассировщик, процессор точек останова. Плата сменного пода содержит эмулирующий процессор под конкретный тип микроконтроллера. Сменные адаптеры обеспечивают установку эмулятора в колодки DIP, PLCC, а также на посадочные места QFP, SOIC и SSOP на плате пользователя. Питание эмулятора осуществляется от блока питания 3,3 В, 1 А или непосредственно от отлаживаемого устройства. Связь с компьютером – по гальванически развязанному каналу RS-232C на скорости 115 КБод или по каналу USB.

9.3 Характеристики аппаратуры

- Точная эмуляция – отсутствие каких-либо ограничений на использование программой пользователя ресурсов микроконтроллера.
- До 1 Мбайта эмулируемой памяти программ и 512 Кбайт памяти данных при использовании основной платы MR1-52-05 (до 64 Кбайт памяти программ и 64 Кбайт памяти данных при использовании основной платы MR1-52-03).
- Поддержка банкированной модели памяти – до 32 банков размером по 64 Кбайт.
- Распределение памяти между эмулятором и устройством пользователя с точностью до 256 байт.
- Анализатор использования памяти программ и данных. Отображение информации анализа в физическом и символьном форматах.

- Аппаратная поддержка отладки программ на языках высокого уровня.
- Трассировка 8 произвольных внешних сигналов.
- 4 выхода синхронизации аппаратуры пользователя.
- Трассировщик реального времени с буфером объемом до 64К фреймов по 128 битов при использовании основной платы MR1-52-05 (до 16К фреймов по 128 битов при использовании основной платы MR1-52-03) с доступом "на лету". Трассировка шины адреса/данных памяти программ, внутренних и внешних данных, EEPROM, сигналов управления, таймера реального времени и восьми внешних сигналов пользователя.
- Программируемый фильтр трассировки, до 1М триггеров запуска и останова трассировщика.
- До 1М аппаратных точек останова по доступу к памяти программ.
- До 512К аппаратных точек останова по доступу к внешней памяти данных. Точки останова по доступу ко всей памяти данных на кристалле: регистрам SFR, внутренней области XDATA, энергонезависимой EEPROM и пр.
- Аппаратный процессор точек останова с возможностью задания сложного условия останова эмуляции по комбинации сигналов адреса, данных, управления, восьми внешних сигналов, таймера реального времени, счетчиков событий и таймера задержки.
- Четыре комплексных точки останова, которые могут быть использованы независимо или в комбинациях по условиям AND/OR/IF-THEN.
- 48-разрядный таймер реального времени.
- Прозрачная эмуляция – доступ "на лету" к эмулируемой памяти, точкам останова, процессору точек останова, буферу трассировки, таймеру реального времени.

– Программируемый генератор тактовой частоты для эмулируемого процессора. Возможность плавного изменения тактовой частоты от 5 кГц до 70 МГц с шагом в 1 кГц и точностью 0,5 %. Реализация 12-, 6-, 4-, 2-тактного режимов работы.

– Последовательный канал связи с компьютером: оптоизолированный RS-232C со скоростью обмена 115К Бод или USB интерфейс формата 1.1, совместимый с USB 2.0.

– Программируемый источник питания эмуляционного кристалла в диапазоне 1,8 В до 5,5 В с двумя режимами работы: режим автоматического отслеживания напряжения питания на отлаживаемом устройстве и режим принудительной установки напряжения по команде пользователя.

– Встроенная система самодиагностики аппаратуры эмулятора.

9.4 Характеристики программного обеспечения

– Программное обеспечение работает в среде Windows-95/98/ME/NT/2000/XP.

– Поддерживается разработка программ на уровне ведения проектов для макроассемблера MCA-51 нашей фирмы, который входит в комплект поставки, а также для пакетов кросс-средств языка Си и ассемблера фирм IAR Systems, Keil Software и Raisonance. Помимо указанных пакетов, поддерживается полнофункциональная символьная отладка программ, созданных с помощью компиляторов фирм Avocet Systems, Hi-Tech Software, Tasking, Crossware, Intel.

– Автоматическое сохранение и загрузка файлов конфигурации аппаратуры, интерфейса и опций отладки. Обеспечивается совместимость файлов конфигурации с симулятором PDS-52. Обеспечена переносимость проектов между эмулятором PICE-52 и симулятором PDS-52.

9.5 Компоненты эмулятора

Для эмулятора PICE-52 существует несколько вариантов основной платы, различающихся по скорости, объему памяти и, соответственно, по

цене. Каждый вариант имеет свой номер, присутствующий в конце обозначения платы: MR1-52-XX. Минимальные параметры и цену обеспечивает основная для PICE-52 плата MR1-52-03. Название ПОДа состоит из следующих символов (слева направо): "P" – указывает, что это ПОД; "R1" – соотносит ПОД с основной платой; "52" – обозначает семейство микроконтроллеров (в данном случае 8051); "ARZ", "A5112", "ARX" – характеризует группу микроконтроллеров, поддерживаемую данным ПОДом. Если ПОД допускает установку нескольких эмуляционных кристаллов, в конце обязательно указывается и его аббревиатура (например, "ARX/W78", "ARX/ID2").

Название адаптера состоит из следующих символов (слева направо): "A" – указывает, что это адаптер; "R1" – соотносит адаптер с основной платой; "52" – обозначает семейство микроконтроллеров (в данном случае 8051); "ARZ", "A5112", "ARX" в сочетании с предшествующим "52" образует название ПОДа, с которым этот адаптер стыкуется (как правило, адаптеры могут стыковаться с несколькими разными ПОДами, но имя наследуют от наиболее характерного); "D", "L", "Q" – характеризуют тип корпуса эмулируемого микроконтроллера ("D" – соответствует DIP, "L" – соответствует PLCC, "Q" – соответствует QFP); "08", "44" и т. д. – указывают число выводов корпуса. Для эмуляции микроконтроллеров в корпусах SOIC и SSOP соответствующий адаптер для DIP корпуса укомплектовывается переходником из корпуса SOIC в корпус DIP стороннего производителя с распайкой один-в-один, в названии которого обязательно указывается число ног: PA-DSO-2003, PA-DSO-2803.

Комплект поставки эмулятора PICE-52:

- Руководство пользователя и паспорт (гарантийный талон).
- Компакт-диск с программным обеспечением и документацией.
- Аппаратура эмулятора.
- Кабель связи с компьютером (RS-232C или USB).
- Трассировочный кабель.

- Блок питания.
- Упаковочная коробка.

PDS-52 – это интегрированный комплекс профессиональных средств для разработки систем на базе семейства микроконтроллеров 8051 фирмы Intel, включающий среду разработки, макроассемблер, отладчик-симулятор, примеры программ и проектов, мощную систему контекстной помощи, электронные гипертекстные руководства по всем компонентам пакета, а также краткое руководство пользователя в печатном виде. PDS-52 работает в среде Windows-95/98/ME/NT/2000/XP.

С помощью PDS-52 можно эффективно разрабатывать и отлаживать программы, используя не только входящий в комплект макроассемблер MCA-51, но и кросс-средства фирм Keil Software и IAR Systems, для которых также предоставляется возможность разработки программ на уровне ведения проектов. Помимо указанных пакетов, PDS-52 обеспечивает полнофункциональную символьную отладку программ, созданных с помощью пакетов фирм Franklin Software, Avocet Systems, Hi-Tech Software, Tasking Software. Пользователю предоставляется обширный сервис по выполнению отлаживаемой программы в различных режимах, манипуляции различными типами точек останова, просмотру и модификации состояния ресурсов микроконтроллера. Поддерживается отладка программ по исходному тексту, а также просмотр и изменение значений сложных объектов языка высокого уровня – массивов, структур, указателей.

Среда разработки программ PDS-52 интегрирует в себе средства, используемые при разработке программ для микроконтроллеров 8051. Обеспечивается интерактивная поддержка всех этапов разработки от написания исходного текста до зашивки готовой программы в ПЗУ микроконтроллера, а именно:

- написание исходных текстов программ с помощью встроенного многооконного редактора;

- настройка опций кросс-средств, используемых для компиляции программы (ассемблера, компилятора Си, линкера, библиотекаря). Настройка производится с помощью диалогов, снабженных контекстной справочной информацией;

- компиляция и линковка программы. Если компилятор обнаруживает ошибки в исходном тексте программы, то строка с ошибкой в окне редактора подсвечивается и ошибки можно сразу же исправить;

- отладка программы;

- "зашивка" программы в ПЗУ микроконтроллера.

"Интегрированность" среды PDS-52 проявляется в том, что перечисленные этапы разработки связываются в одно целое. Самые трудоемкие этапы, а именно компиляция/линковка с диагностикой и исправлением ошибок, максимально упрощены. PDS-52 самостоятельно следит за изменениями, которые вносятся в исходные тексты программ. Например, исправив ошибку в исходном тексте, можно нажатием одной кнопки "выполнить программу до курсора" заставить PDS-52 перетранслировать изменившиеся модули, загрузить полученную программу в память отладчика и запустить ее до указанной строки. Переход от отладки к редактированию происходит так же прозрачно и быстро.

9.6 Отладочные возможности PDS-52

Симулятор PDS-52 представляет собой программно-логическую модель микроконтроллера, имитирующую (симулирующую) работу ядра архитектуры семейства Intel 8051 – памяти, АЛУ, системы команд, регистров (периферийные устройства не поддерживаются). Возможности PDS-52:

- отслеживание выполнения программы по ее исходному тексту;

- просмотр и изменение значений любых переменных;

- встроенный анализатор эффективности программного кода;

- точки останова по сложному условию;

- неограниченное количество точек останова по доступу к ячейкам памяти;

- просмотр стека вызовов подпрограмм и функций;
- встроенный строчный ассемблер;
- возможность выполнения программы "назад" на большое количество шагов, а также в непрерывном режиме. При этом состояние модели микроконтроллера полностью восстанавливается;
- точный подсчет интервалов времени и многое другое.

Основные достоинства программно-логической модели микроконтроллера, реализованной в PDS-52 – точная симуляция узлов микроконтроллера и возможность моделировать устройства, подключенные к микроконтроллеру "снаружи" (т. н. моделирование внешней среды), например, внешнюю логику, датчики, клавиатуру, исполнительные устройства (дисплеи), задавать периодические и непериодические воздействия и т. п.

9.7 Кросс-макроассемблер

Кросс-макроассемблер предназначен для трансляции исходных текстов программ для процессоров семейства 8051 фирмы Intel. Кросс-макроассемблер имеет следующие возможности:

- поддерживает все микроконтроллеры 8051 фирмы Intel;
- генерирует HEX-файл и подробный листинг;
- поддерживает широкий набор директив условной трансляции;
- предоставляет удобные средства работы с макросами;
- генерирует подробную символьную информацию для отладчиков;
- допускает использование русских букв в именах;
- поддерживает 16-битные арифметические и логические выражения;
- выполняет проверку перекрытия кода;
- выполняет проверку размещения данных в запрещенных областях;
- включает полный набор include-файлов;
- поставляется как в составе пакета Project-52, так и отдельно.

Макроассемблер МСА-51 поддерживает все известные на текущий момент процессоры семейства Intel 8051. Имеется возможность расширять номенклатуру поддерживаемых процессоров без обновления версии ассемблера. Использование русских букв в именах позволяет создавать исходные тексты программ, обладающие превосходной читаемостью. Генерируется подробный листинг, включающий не только текст программы и адреса инструкций, но также и таблицы символов, макросов, констант и т. п. с указанием имен, к которым не было ссылок в программе. Микропроцессоры семейства 8051 8-разрядные, тем не менее, МСА-51 позволяет объявлять в исходном тексте переменные, занимающие 2 и 4 байта, а также массивы байт, слов и двойных слов, если того требует логика организации данных в программе. При отладке программы такие переменные будут представлены в соответствии с их объявлением в программе, т. е. как слова, двойные слова и массивы, а не просто как цепочки байт.

10 Заключение

В настоящем руководстве КФДЛ.431281.026 приведено подробное описание архитектуры, функционального построения, системы команд и особенностей применения микроконтроллеров 1830BE32У, 1830BE32АУ, 1830BE52У, которые представляют собой 8-разрядную ОЭВМ с внутренней программной памятью объемом 4 Кбайт для 1830BE52У, масочный вариант и без ПЗУ для 1830BE32У, 1830BE32АУ.

Все значения электрических параметров микроконтроллеров приведены в технических условиях АЕЯР.431280.378. Значения параметров, приведенные в данном техническом описании, являются справочными.

Настоящее руководство может служить практическим пособием по применению микроконтроллеров для разработчиков систем с повышенной стойкостью к ионизирующим воздействиям на основе микросхем 1830BE32У, 1830BE32АУ, 1830BE52У, а также для программистов.

Применение радиационно-стойких микроконтроллеров в цифровых системах управления, для управления робототехническими комплексами, в системах автоматизации технологических процессов, в системах автоматизированного управления электроприводом, оргтехнике, вычислительной технике, телекоммуникационной технике и т. д. позволит создавать надежные изделия при эксплуатации в условиях повышенной радиации.

