

# МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS-51

<b>1. ОБЩИЕ ОСОБЕННОСТИ УПРАВЛЯЮЩИХ МИКРОКОНТРОЛЛЕРОВ.....</b>	<b>5</b>
1.1. Четырехразрядные микроконтроллеры. ....	5
<b>2. УПРАВЛЯЮЩИЕ ВОСЬМИРАЗЯДНЫЕ МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS-51 ФИРМЫ INTEL И СОВМЕСТИМЫЕ С НИМИ.....</b>	<b>6</b>
<b>2.1. Структурная организация микроконтроллера i8051.....</b>	<b>6</b>
2.1.1. Общие характеристики.....	6
2.1.2. Арифметико-логическое устройство .....	7
2.1.3. Назначение выводов микроконтроллера 8051. ....	8
<b>3. ОРГАНИЗАЦИЯ ОЗУ, ПЗУ И РЕГИСТРОВ МИКРОКОНТРОЛЛЕРА 8051.....</b>	<b>10</b>
3.1.1. Память программ (ПЗУ). ....	10
3.1.2. Память данных (ОЗУ).....	10
3.1.3. Регистры специальных функций. ....	11
3.1.4. Регистры специальных функций. ....	11
3.1.5. Регистр флагов (PSW). ....	12
3.1.6. Устройство управления и синхронизации.....	13
<b>3.2. Организация портов ввода вывода микроконтроллера 8051.....</b>	<b>13</b>
3.2.1. Общие сведения. ....	13
3.2.2. Альтернативные функции. ....	13
3.2.3. Устройство портов. ....	14
3.2.4. Особенности электрических характеристик портов .....	15
<b>3.3. Таймеры / счетчики микроконтроллеров семейства 8051.....</b>	<b>15</b>
3.3.1. Регистр режима работы таймера/счетчика TMOD.....	16
3.3.2. Регистр управления/статуса таймера TCON. ....	16
3.3.3. Режимы работы таймеров-счетчиков. ....	16
<b>3.4. Последовательный порт микроконтроллера 8051.....</b>	<b>18</b>
<b>3.5. Регистр управления/статуса приемопередатчика SCON.....</b>	<b>18</b>
3.5.1. Функциональное назначение бит регистра управления/статуса приемопередатчика SCON. ....	19
3.5.2. Скорость приема/передачи информации через последовательный порт. ....	19
3.5.3. Регистр управления мощностью PCON.....	19
<b>3.6. Система прерываний микроконтроллера 8051.....</b>	<b>20</b>
3.6.1. Регистр масок прерывания (IE). ....	21
3.6.2. Регистр приоритетов прерываний (IP).....	21
3.6.3. Выполнение подпрограммы прерывания. ....	21
<b>3.7. Работа с внешней памятью микроконтроллера 8051.....</b>	<b>22</b>
<b>3.8. Режимы микроконтроллера 8051 с пониженным энергопотреблением.....</b>	<b>22</b>
3.8.1. Режим XX.....	23
3.8.2. Режим ВМП. ....	23
<b>4. СИСТЕМА КОМАНД МИКРОКОНТРОЛЛЕРА СЕМЕЙСТВА 8051. ....</b>	<b>24</b>
4.1.1. Общая характеристика.....	24
4.1.2. Типы команд .....	24

4.1.3.Типы операндов.....	24
4.1.4.Группы команд.....	25
4.1.5.Обозначения, используемые при описании команд.....	26
4.1.6.Команды пересылки данных микроконтроллера 8051.....	26
4.1.7.Команды арифметических операций 8051.....	27
4.1.8.Команды логических операций микроконтроллера 8051.....	28
4.1.9.Команды операций над битами микроконтроллера 8051.....	29
4.1.10.Команды передачи управления микроконтроллера 8051.....	29
<b>5.ДОПОЛНИТЕЛЬНЫЕ УСТРОЙСТВА В РАСШИРЕННЫХ ВАРИАНТАХ МИКРОКОНТРОЛЛЕРА.....</b>	<b>31</b>
5.0.1.Расширения микропроцессоров семейства MCS-51/52.....	31
5.0.2.8052.....	31
5.0.3.8XC51FA.....	32
5.0.4.8XC51GB.....	32
5.0.5.80C152.....	32
5.0.6.Маркировка микроконтроллеров фирмы Intel.....	33
<b>5.1.PCA микроконтроллера 8051.....</b>	<b>34</b>
<b>5.2.16-разрядный таймер-счетчик устройства PCA микроконтроллера MCS-51.....</b>	<b>35</b>
5.2.1.Регистр режимов PCA таймера-счетчика CMOD.....	35
5.2.2.Регистр управления PCA таймером-счетчиком CCON.....	36
<b>5.3.Модули сравнения-захвата PCA микроконтроллеров MCS-51.....</b>	<b>36</b>
5.3.1.Регистр режимов модуля сравнения захвата CCAPMn.....	36
5.3.2.Режимы работы PCA.....	37
<b>5.4.Режимы работы PCA микроконтроллеров семейства MCS-51.....</b>	<b>37</b>
5.4.1.Режим захвата.....	37
5.4.2.Режим 16-разрядного программируемого таймера.....	37
5.4.3.Режим скоростного вывода.....	38
5.4.4.Режим сторожевого таймера (watchdog timer).....	38
5.4.5.Режим генерации импульсов заданной скважности.....	39
<b>5.5.Аналого-цифровой преобразователь микроконтроллеров семейства MCS-51.....</b>	<b>39</b>
5.5.1.ADCON - Регистр управления преобразователем.....	40
5.5.2.ADDAT - регистр результатов преобразования.....	40
5.5.3.DAPR - регистр программирования опорных напряжений АЦП.....	40
5.5.4.Синхронизация АЦП и время преобразования.....	41
<b>5.6.Таймер счетчик T/C2 микроконтроллера 8052.....</b>	<b>42</b>
5.6.1.Регистр управление таймера/счетчика 2 T2COM.....	42
5.6.2.Режимы работы таймера/счетчика 2.....	42
5.6.3.Регистр режима таймера/счетчика 2 T2MOD.....	43
5.6.4.Дополнительный регистр приоритетов прерываний IPH.....	44
<b>6.СЕМЕЙСТВО MCS-251.....</b>	<b>45</b>
<b>7.ОДНОКРИСТАЛЬНЫЕ МИКРОКОНТРОЛЛЕРЫ INTEL MCS-96.....</b>	<b>47</b>
<b>7.1.Общая характеристика.....</b>	<b>47</b>
7.1.1.Структура микроконтроллера.....	47
<b>7.2.Периферийные устройства.....</b>	<b>48</b>
7.2.1.Устройства ввода и вывода данных.....	48

7.2.2.Устройство ввода и вывода дискретных сигналов.....	48
7.2.3.Устройства ввода и вывода аналоговых сигналов .....	49
7.2.4.Устройства обмена данными с другими микроконтроллерами и центральным процессором.....	49
7.2.5.Устройства приема и обслуживания запросов прерывания.....	50
7.2.6.Устройства контроля правильности функционирования микроконтроллера. ....	50
7.2.7.Характеристики микроконтроллеров подсемейств. ....	50
7.2.8.Почему 80C196 быстрее, чем 8051? .....	51
<b>8.ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ МИКРОКОНТРОЛЛЕРОВ И ПРОЦЕССОРОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ.....</b>	<b>52</b>
8.1.1.Общие особенности. ....	52
<b>8.2.Типы инструментальных средств разработки и отладки программ для микроконтроллеров и процессоров цифровой обработки сигналов.....</b>	<b>52</b>
<b>8.3.Внутрисхемные эмуляторы. ....</b>	<b>52</b>
8.3.1.Принцип работы.....	52
8.3.2.Классификация внутрисхемных эмуляторов. ....	52
8.3.3.Функциональные возможности внутрисхемных эмуляторов.....	53
8.3.4.Достоинства и недостатки внутрисхемных эмуляторов. ....	53
<b>8.4.PICE-51.....</b>	<b>53</b>
8.4.1.Внутрисхемный эмулятор 8-разрядных микроконтроллеров семейства 8051. ....	53
8.4.2.Характеристика аппаратуры. ....	54
8.4.3.Характеристики программного обеспечения . ....	55
8.4.4.Структурная схема эмулятора PICE-51.....	55
8.4.5.Варианты комплектации эмулятора PICE-51.....	56
8.4.6.Сравнительные характеристики некоторых эмуляторов для микроконтроллеров семейства 8051 ...	56
<b>8.5.Программные симуляторы.....</b>	<b>57</b>
<b>8.6.Платы развития.....</b>	<b>58</b>
<b>8.7.Отладочные мониторы. ....</b>	<b>58</b>
8.7.1.Принцип работы.....	58
8.7.2.Достоинства и недостатки отладочных мониторов.....	58
<b>8.8.Эмуляторы ПЗУ.....</b>	<b>59</b>
<b>8.9.Типичные функциональные модули средств разработки и отладки. ....</b>	<b>59</b>
<b>8.10.Отладчик.....</b>	<b>60</b>
<b>8.11.Узел эмуляции микроконтроллера.....</b>	<b>60</b>
<b>8.12.Эмуляционная память.....</b>	<b>60</b>
<b>8.13.Подсистема точек останова. ....</b>	<b>60</b>
<b>8.14.Процессор точек останова. ....</b>	<b>61</b>
<b>8.15.Трассировщик.....</b>	<b>61</b>
<b>8.16.Профилировщик . ....</b>	<b>61</b>
<b>8.17.Интегрированная среда разработки. ....</b>	<b>61</b>

<b>9.ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА MCS-51. ...</b>	<b>63</b>
<b>9.1.Дизассемблеры MCS-51.....</b>	<b>63</b>
<b>9.2. Введение в интегрированную отладочную среду ProView для микроконтроллеров семейства MCS-51, 251, XA.....</b>	<b>63</b>
9.2.1.Оптимизирующий кросс-компилятор C51.....	64
9.2.2.Макроассемблер A51.....	64
9.2.3.Компоновщик L51.....	64
9.2.4.Отладчик/симулятор WinSim51.....	64
<b>9.3.Быстрый старт.....</b>	<b>64</b>
9.3.1.Запуск ProView и создание файла проекта.....	65
9.3.2.Добавка файла с исходным текстом и его редактирование.....	65
9.3.3.Компиляция и компоновка.....	66
9.3.4.Тестирование и отладка.....	66
9.3.5.Пошаговый режим и выход из отладчика.....	68
9.3.6.Следующий шаг.....	68
<b>9.4.Интегрированная отладочная среда mVision2.....</b>	<b>71</b>
<b>9.5..Компиляторы с языка ассемблер для микроконтроллеров семейства MCS-51... 71</b>	<b>71</b>
<b>10.ОСНОВНЫЕ ХАРАКТЕРИСТИКИ МИКРОКОНТРОЛЛЕРОВ MCS 51 / MCS 251.....</b>	<b>72</b>
<b>11.МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS51 И ЕГО АНАЛОГИ.....</b>	<b>74</b>
<b>12.СПИСОК ЛИТЕРАТУРЫ.....</b>	<b>75</b>

# 1. Общие особенности управляющих микроконтроллеров.

Основные требования, которые потребители предъявляют к управляющим блокам приборов можно сформулировать следующим образом:

- низкая стоимость,
- высокая надежность,
- высокая степень миниатюризации,
- малое энергопотребление,
- работоспособность в жестких условиях эксплуатации;
- достаточная производительность для выполнения всех требуемых функций.

В отличие от универсальных компьютеров к управляющим контроллерам, как правило, не предъявляются высокие требования к производительности и программной совместимости.

Выполнение всех этих довольно противоречивых условий одновременно затруднительно, поэтому развитие и совершенствование техники пошло по пути специализации и в настоящее время количество различных моделей управляющих микроконтроллеров чрезвычайно велико.

Однако можно выделить некоторые черты архитектуры и системы команд, общие для всех современных микроконтроллеров, это:

- так называемая Гарвардская архитектура – то есть отдельные области памяти для хранения команд (программы) и данных. Они могут иметь разную разрядность, в системе команд для обращения к ним предусмотрены различные команды и т.д.
- интеграция в одном корпусе микросхемы (на одном кристалле) практически всех блоков, характерных для полнофункционального компьютера – процессора, ПЗУ, ОЗУ, устройств ввода-вывода, тактового генератора, контроллера прерываний и т.д. Поэтому в русскоязычной литературе подобные устройства часто называются однокристалльные ЭВМ (ОЭВМ).

Микроконтроллеры обычно классифицируют по разрядности обрабатываемых чисел

- четырехразрядные- самые простые и дешевые,
- восьмиразрядные - наиболее многочисленная группа (оптимальное сочетание цены и возможностей), к этой группе относятся микроконтроллеры серии MCS-51 (Intel) и совместимые с ними, PIC (MicroChip), HC68 (Motorola), Z8 (Zilog) и др.
- шестнадцатиразрядные - MCS-96 (intel) и др. - более высокопроизводительные но более дорогостоящие
- тридцатидвухразрядные - обычно являющиеся модификациями универсальных микропроцессоров, например i80186 или i386EX.

## 1.1. Четырехразрядные микроконтроллеры.

Четырехразрядные микроконтроллеры являются очень простыми и дешевыми устройствами, предназначенными для замены несложных схем на "жесткой" логике в системах с невысоким быстродействием. Типичные случаи применения - часы, калькуляторы, игрушки, простые устройства управления в промышленных устройствах и бытовой технике.

Типичные характеристики четырехразрядных микроконтроллеров -

- ОЗУ - объем - 16...64 четырехразрядные ячейки;
- ПЗУ - объем - 0.5 ...1 К восьмиразрядных ячеек, тип - масочное ПЗУ (наиболее дешевое);
- система команд - количество - 30...50;
- тактовая частота - 100 КГц...1 МГц;
- периферийные устройства - 2...4 четырехразрядных параллельных порта, иногда контроллер жидкокристаллического индикатора;
- стоимость - порядка 0.1\$

Примеры четырехразрядных микроконтроллеров - отечественные серии КР145ВМ1405, КР145ВМ1406, КР1834 и т.д.

Одним из крупнейших производителей четырехразрядных микроконтроллеров - "Ангстрем" г. Зеленоград (и экспортер в страны Юго-Восточной Азии).

## **2.Управляющие восьмиразрядные микроконтроллеры семейства MCS-51 фирмы Intel и совместимые с ними.**

Несмотря на непрерывное развитие и появление все новых и новых 16- и 32-разрядных микроконтроллеров и микропроцессоров, наибольшая доля мирового микропроцессорного рынка и по сей день остается за 8-разрядными устройствами. Согласно данным компании Semico Research Corp., Phoenix, в 1996 году общий мировой объем продаж микроконтроллеров всех типов составил \$11,4 миллиарда, при этом \$5,56 миллиарда (или 48.6%) пришлось на долю 8-разрядных кристаллов. Это в 2,5 раза больше объема продаж ближайших конкурентов: 16-разрядных микроконтроллеров (\$2.1 млрд.) и DSP (\$2.4). По всем прогнозам аналитических компаний на ближайшие 5 лет лидирующее положение 8-разрядных микроконтроллеров на мировом рынке сохранится.

В настоящее время среди всех 8-разрядных микроконтроллеров - семейство MCS-51 является несомненным чемпионом по количеству разновидностей и количеству компаний, выпускающих его модификации. Оно получило свое название от первого представителя этого семейства - микроконтроллера 8051, выпущенного в 1980 году на базе технологии HMOS. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке. С точки зрения технологии микроконтроллер 8051 являлся для своего времени очень сложным изделием - в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре 8086.

Важную роль в достижении такой высокой популярности семейства 8051 сыграла открытая политика фирмы Intel, родоначальницы архитектуры, направленная на широкое распространение лицензий на ядро 8051 среди большого количества ведущих полупроводниковых компаний мира.

В результате на сегодняшний день существует более 200 модификаций микроконтроллеров семейства 8051, выпускаемых почти 20-ю компаниями. Эти модификации включают в себя кристаллы с широчайшим спектром периферии: от простых 20-выводных устройств с одним таймером и 1К программной памяти до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров-счетчиков, аппаратными 16-разрядными умножителями и 64К программной памяти на кристалле. Каждый год появляются все новые варианты представителей этого семейства. Основными направлениями развития являются: увеличение быстродействия (повышение тактовой частоты и переработка архитектуры), снижение напряжения питания и потребления, увеличение объема ОЗУ и FLASH памяти на кристалле с возможностью внутрисхемного программирования, введение в состав периферии микроконтроллера сложных устройств типа системы управления приводами, CAN и USB интерфейсов и т.п.

Все микроконтроллеры из семейства MCS-51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

Основными производителями клонов 51-го семейства в мире являются фирмы Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и ряд других.

В рамках СССР производство микроконтроллера 8051 осуществлялось в Киеве, Воронеже (1816BE31/51, 1830BE31/51), Минске (1834BE31) и Новосибирске (1850BE31).

Микроконтроллеры данного семейства выпускаются в PLCC, DIP и QFP корпусах и могут работать в следующих температурных диапазонах:

- коммерческий (0°C — +70°C);
- расширенный (-40°C — +85°C);
- для военного использования (-55°C — +125°C).

Примерами микроконтроллеров семейства MCS-51 с расширенными возможностями могут Расширения микроконтроллеров MCS-51/52служить 8XC51FA, 8XC51GB, 80C152.

### **2.1.Структурная организация микроконтроллера i8051.**

#### **2.1.1.Общие характеристики.**

Микроконтроллер семейства 8051 имеют следующие аппаратные особенности:

- внутреннее ОЗУ объемом 128 байт;
- четыре двунаправленных побитно настраиваемых восьмиразрядных порта ввода-вывода;
- два 16-разрядных таймера-счетчика;
- встроенный тактовый генератор;
- адресация 64 Кбайт памяти программ и 64 Кбайт памяти данных;
- две линии запросов на прерывание от внешних устройств;
- интерфейс для последовательного обмена информацией с другими микроконтроллерами или персональными компьютерами.

Микроконтроллер 8751 снабжен УФ ПЗУ объемом 4 Кбайт.

Функциональная схема микроконтроллера семейства 8051.

Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы (рис. 1) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0-P3).

### 2.1.2. Арифметико-логическое устройство

8-битное арифметико-логическое устройство (ALU) может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. К входам подключены программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции (DCU) и схема формирования признаков результата операции (PSW).

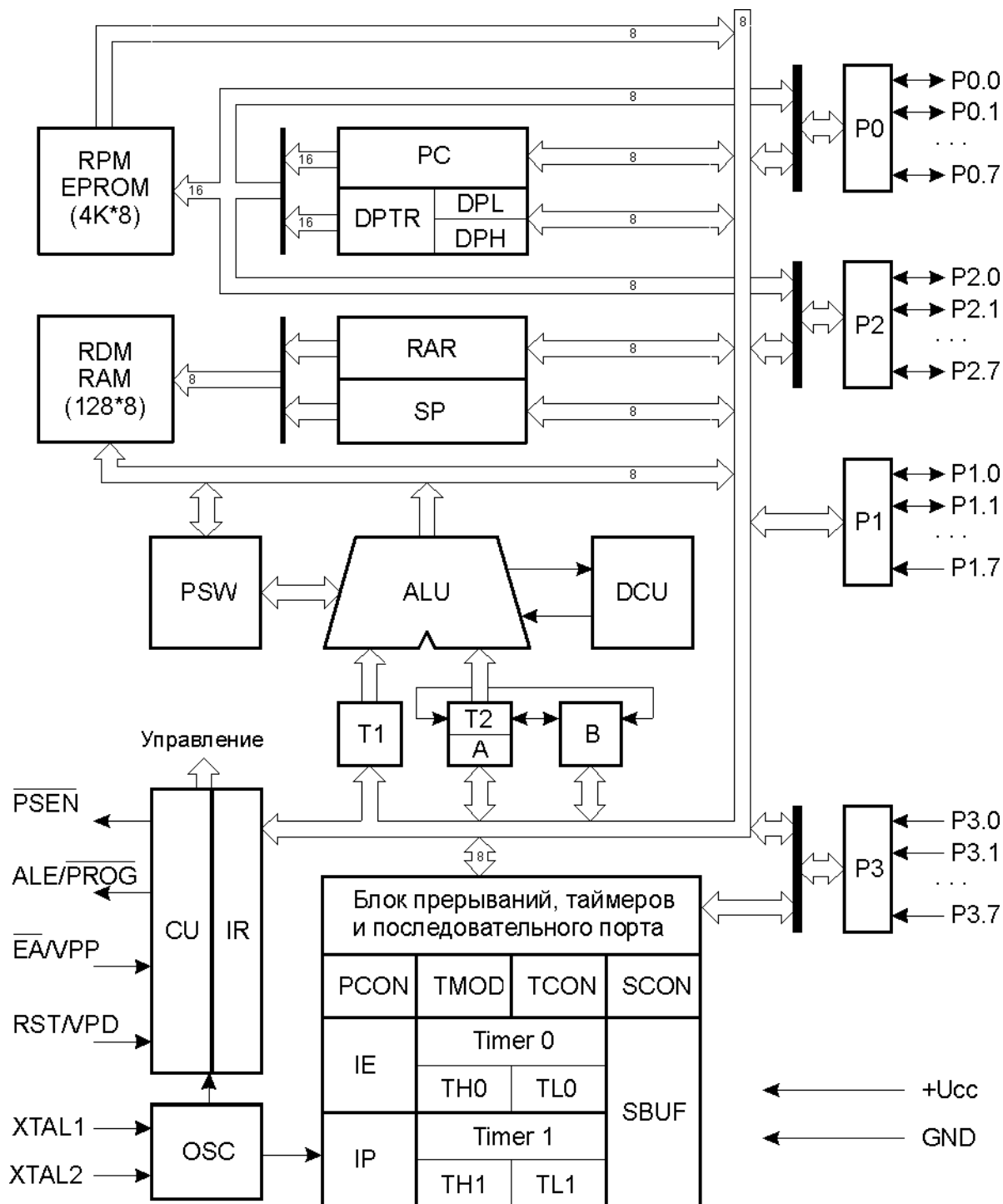


Рис. 1. Структурная схема микроконтроллера KM1816BE51

Простейшая операция сложения используется в ALU для инкрементирования содержимого регистров, продвижения регистра-указателя данных (RAR) и автоматического вычисления следующего адреса

резидентной памяти программ. Простейшая операция вычитания используется в ALU для декрементирования регистров и сравнения переменных.

Простейшие операции автоматически образуют “танделы” для выполнения таких операций, как, например, инкрементирование 16-битных регистровых пар. В ALU реализуется механизм каскадного выполнения простейших операций для реализации сложных команд. Так, например, при выполнении одной из команд условной передачи управления по результату сравнения в ALU трижды инкрементируется счётчик команд (PC), дважды производится чтение из RDM, выполняется арифметическое сравнение двух переменных, формируется 16-битный адрес перехода и принимается решение о том, делать или не делать переход по программе. Все перечисленные операции выполняются всего лишь за 2 мкс.

Важной особенностью ALU является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Эта способность достаточно важна, поскольку для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевыми переменными, реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Таким образом, ALU может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В ALU выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования операции и режима адресации базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

### 2.1.3. Назначение выводов микроконтроллера 8051.

P1.0	□ 1	40	□ Vcc
P1.1	□ 2	39	□ P0.0 (AD0)
P1.2	□ 3	38	□ P0.1 (AD1)
P1.3	□ 4	37	□ P0.2 (AD2)
P1.4	□ 5	36	□ P0.3 (AD3)
P1.5	□ 6	35	□ P0.4 (AD4)
P1.6	□ 7	34	□ P0.5 (AD5)
P1.7	□ 8	33	□ P0.6 (AD6)
RESET	□ 9	32	□ P0.7 (AD7)
(RxD) P3.0	□ 10	31	□ EA/Vpp
(TxD) P3.1	□ 11	30	□ ALE/ $\overline{\text{P}}\text{ROG}$
(INT0) P3.2	□ 12	29	□ PSEN
(INT1) P3.3	□ 13	28	□ P2.7 (A15)
(T0) P3.4	□ 14	27	□ P2.6 (A14)
(T1) P3.5	□ 15	26	□ P2.5 (A13)
(WR) P3.6	□ 16	25	□ P2.4 (A12)
(RD) P3.7	□ 17	24	□ P2.3 (A11)
XTAL2	□ 18	23	□ P2.2 (A10)
XTAL1	□ 19	22	□ P2.1 (A9)
Vss	□ 20	21	□ P2.0 (A8)

**Рис. 2.** Назначение выводов 8051.

Обозначения на этом рисунке:

- $U_{ss}$  — потенциал общего провода ("земли");
- $U_{cc}$  — основное напряжение питания +5 В;
- X1, X2 — выводы для подключения кварцевого резонатора;
- RST — вход общего сброса микроконтроллера;
- PSEN — разрешение внешней памяти программ; выдается только при обращении к внешнему ПЗУ;
- ALE — строб адреса внешней памяти;
- EA — отключение внутренней программной памяти; уровень 0 на этом входе заставляет микроконтроллер выполнять программу только внешнего ПЗУ; игнорируя внутреннее (если последнее имеется);
- P1 — восьми битный квази двунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов;



- P2 — восьми битный квази двунаправленный порт, аналогичный P1; кроме того, выводы этого порта используются для выдачи адресной информации при обращении к внешней памяти программ или данных (если используется 16-битовая адресация последней). Выводы порта используются при программировании 8751 для ввода в микроконтроллер старших разрядов адреса;
- P3 — восьми битный квази двунаправленный порт, аналогичный P1; кроме того, выводы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, порта последовательного ввода-вывода, контроллера прерываний, и внешней памяти программ и данных;
- P0 — восьми 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.

### 3. Организация ОЗУ, ПЗУ и регистров микроконтроллера 8051.

#### 3.1.1. Память программ (ПЗУ).

Как и у большинства микроконтроллеров, у микроконтроллеров семейства 8051, память программ и память данных являются самостоятельными и независимыми друг от друга устройствами, адресуемыми различными командами и управляющими сигналами.

Объем **встроенной памяти программ**, расположенной на кристалле микроконтроллера 8051 и 8751, равен 4 Кбайт. При обращении к **внешней памяти программ** все микроконтроллеры семейства 8051 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 PC), 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 с использованием как счетчика PC, так и специального двухбайтового регистра-указателя данных DPTR.

#### 3.1.2. Память данных (ОЗУ).

Объем **расположенной на кристалле памяти данных**—128 байт. Объем **внешней памяти** данных может достигать 64 Кбайт. Первые 32 байта организованы в четыре банка регистров общего назначения, обозначаемых соответственно банк 0 — банк 3. Каждый из них состоит из восьми регистров R0 — R7. В любой момент программе доступен только один банк регистров, номер которого содержится в третьем и четвертом битах слова состояния программы PSW (см. ниже).

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 с помощью регистров R0 или R1: перед выполнением соответствующей команды в один из них должен быть занесен адрес ячейки, к которой необходимо обратиться.

Для обращения к внешней памяти данных используется только косвенная адресация с помощью регистров R0 и R1 или с помощью 16-разрядного регистра-указателя DPTR. Он относится к группе регистров специальных функций, и с его помощью можно адресовать все 64 Кбайта внешней памяти.

Часть памяти данных представляет собой так называемую битовую область, в ней имеется возможность при помощи специальных битовых команд адресовываться к каждому разряду ячеек памяти. Адрес прямо адресуемых битов может быть записан либо в виде (Адрес Байта).(Разряд), например выражение 21.3 означает третий разряд ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса. Соответствие этих двух способов адресации можно определить по таблице.

Адрес байта	Адреса битов по разрядам								
	Adr	D7	D6	D5	D4	D3	D2	D1	D0
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	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	

**Таблица.1.** Адреса битовых областей памяти микроконтроллера 8051

**ПРИМЕЧАНИЕ.** Адрес прямо адресуемых битов может быть записан либо в виде выражения (Адрес Байта).(Разряд), например выражение 21.3 означает адрес третьего разряда ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) 0B.

### 3.1.3.Регистры специальных функций.

Адрес байта	Адреса битов по разрядам								Имя регистра
Adr	D7	D6	D5	D4	D3	D2	D1	D0	Name
F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
...	...								...
E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
...	...								...
D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
...	...								...
B8H	-	-	-	BC	BB	BA	B9	B8	IP
...	...								...
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
...	...								...
A8H	AF	-	-	AC	AB	AA	A9	A8	IE
...	...								...
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

**Таблица.2.** Карта адресуемых битов в блоке регистров специальных функций

**ПРИМЕЧАНИЕ.** Адрес прямо адресуемых битов может быть записан либо в виде выражения (Название Регистра).(Разряд), например выражение SCON.3 означает адрес третьего разряда регистра SCON, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) 9В. Кроме того, некоторые биты управляющих регистров имеют собственные названия, так например данный бит имеет название ТВ8

### 3.1.4.Регистры специальных функций.

К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR (Special Function Register).

Адрес	Символ	Наименование
0E0H	*ACC	Аккумулятор (Accumulator)
0F0H	*B	Регистр расширитель аккумулятора (Multiplication Register)
0D0H	*PSW	Слово состояния программы (Program Status Word)
080H	*P0	Порт 0 (SFR P0)
090H	*P1	Порт 1 (SFR P1)
0A0H	*P2	Порт 2 (SFR P2)
0B0H	*P3	Порт 3 (SFR P3)
081H	SP	Регистр указатель стека (Stack Pointer)
083H	DPH	Старший байт регистра указателя данных DPTR (Data Pointer High)
082H	DPL	Младший байт регистра указателя данных DPTR (Data Pointer Low)
08CH	TH0	Старший байт таймера 0 ()
08AH	TL0	Младший байт таймера 0 ()
08DH	TH1	Старший байт таймера 1 ()
08BH	TL1	Младший байт таймера 1 ()
089H	TMOD	Регистр режимов таймеров счетчиков (Timer/Counter Mode Control Register)
088H	*TCON	Регистр управления статуса таймеров (Timer/Counter Control Register)
0B8H	*IP	Регистр приоритетов (Interrupt Priority Control Register)
0A8H	*IE	Регистр маски прерывания (Interrupt Enable Register)
087H	PCON	Регистр управления мощностью (Power Control Register)
098H	*SCON	Регистр управления приемопередатчиком (Serial Port Control Register)
099H	SBUF	Буфер приемопередатчика (Serial Data Buffer)

**Таблица.3.** Размещение регистров специальных функций в пространстве SFR

*Примечание.* Регистры, символ которых отмечен знаком (\*), допускают адресацию своих отдельных бит при использовании команд из группы команд операций над битами.

Адреса, по которым расположены эти регистры, приведены в таблице.

Отметим, что регистры занимают только часть 128-байтового адресного пространства. Те ячейки памяти с адресами 80H-0FFH, которые не заняты регистрами, физически отсутствуют, на кристаллах микроконтроллеров семейства 8051 при обращении к ним можно прочитать лишь код команды возврата.

Регистры специальных функций управляют работой блоков, входящих в микроконтроллер.

- Регистры-защелки **SFR** параллельных портов **P0...P3** - служат для ввода-вывода информации.
- Две регистровые пары с именами **TH0, TL0** и **TH1, TL1** представляют собой регистры, двух программно-управляемых 16-битных таймеров-счетчиков.
- Режимы таймеров-счетчиков задаются с использованием регистра **TMOD**, а управление ими осуществляется с помощью регистра **TCON**.
- Для управления режимами энергопотребления микро-ЭВМ используется регистр **PCON**.
- Регистры **IP** и **IE** управляют работой системы прерываний микро-ЭВМ,
- регистры **SBUF** и **SCON** — работой приемопередатчика последовательного порта.
- Регистр-указатель стека **SP** в микро-ЭВМ рассматриваемого семейства — восьми битный. Он может адресовать любую область внутренней памяти данных. В отличие от микропроцессора KP580BM80, у микро-ЭВМ семейства 8051 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 после извлечения информации. В процессе инициализации микро-ЭВМ после сигнала сброса или при включении питающего напряжения в **SP** заносится код 07H. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом 08H.
- Регистр-указатель данных **DPTR** чаще всего используют 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**
- Аккумулятор (**ACC**) является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на ноль, формирование флага паритета и т.п. В распоряжении пользователя имеются 8 регистров общего назначения **R0–R7** одного из четырёх возможных банков. При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре **PSW**.
- Регистр **B** используется как источник и как приемник при операциях умножения и деления, обращение к нему, как к регистру **SFR**, производится аналогично аккумулятору.
- При выполнении ряда команд в арифметико-логическом устройстве (АЛУ) формируются признаки операций — флаги, которые фиксируются в регистре **PSW**.

### 3.1.5.Регистр флагов (PSW).

Символ	Позиция	Имя и назначение																				
P	PSW.0	Флаг приоритета. 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" style="margin-left: auto; margin-right: auto;"> <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>00H - 07H</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>08H - 0FH</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>10H - 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H - 1FH</td> </tr> </tbody> </table>	RS0	RS1	Банк	Границы адресов ОЗУ	0	0	0	00H - 07H	1	0	1	08H - 0FH	0	1	2	10H - 17H	1	1	3	18H - 1FH
RS0	RS1	Банк	Границы адресов ОЗУ																			
0	0	0	00H - 07H																			
1	0	1	08H - 0FH																			
0	1	2	10H - 17H																			
1	1	3	18H - 1FH																			
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																				
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается как аппаратно, так и программным путем																				

**Таблица.4.** Перечень флагов, их символические имена и условия формирования

Наиболее “активным” флагом PSW является флаг переноса, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции “булева аккумулятора” в командах, манипулирующих с битами. Флаг переполнения (OV) фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. ALU не управляет флагами селекции банка регистров (RS0, RS1), их значение полностью определяется прикладной программой и используется для выбора одного из четырёх регистровых банков.

В микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя неявную адресацию. В Intel 8051 дело обстоит иначе. Хотя процессор имеет в своей основе аккумулятор, он может выполнять множество команд и без его участия. Например, данные могут быть переданы из любой ячейки RDM в любой регистр, любой регистр может быть загружен непосредственным операндом и т.д. Многие логические операции могут быть выполнены без участия аккумулятора. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

### 3.1.6. Устройство управления и синхронизации.

Кварцевый резонатор, подключаемый к внешним выводам микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации. Устройство управления (CU) на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора. Большинство команд микроконтроллера выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления производится расчёт времени исполнения прикладных программ.

На схеме микроконтроллера к устройству управления примыкает регистр команд (IR). В его функцию входит хранение кода выполняемой команды.

Входные и выходные сигналы устройства управления и синхронизации:

- PSEN – разрешение программной памяти,
- ALE – выходной сигнал разрешения фиксации адреса,
- PROG – сигнал программирования,
- EA – блокировка работы с внутренней памятью,
- VPP – напряжение программирования,
- RST – сигнал общего сброса,
- VPD – вывод резервного питания памяти от внешнего источника,
- XTAL – входы подключения кварцевого резонатора.

## 3.2. Организация портов ввода вывода микроконтроллера 8051.

### 3.2.1. Общие сведения.

- Количество портов - 4. Название - P0...P3, они адресуются как регистры специальных функций.
- Разрядность - 8 с возможностью побитной адресации разрядов.
- Направление обмена информацией через порты - все порты двунаправленные, причем имеется возможность в каждом порту часть разрядов использовать для ввода данных, а часть для вывода.

### 3.2.2. Альтернативные функции.

Из-за ограниченного количества выводов корпуса ИМС микроконтроллера, большинство выводов используется для выполнения двух функций - в качестве линий портов и для альтернативных функций

- **Порты P0 и P2** 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 восьмиразрядного адреса портом P2 можно пользоваться для ввода-вывода информации обычным образом. При обращении к внешней памяти в P0 автоматически заносятся 1 во все биты. Информация в P2 при этом остается неизменной.
- **Порт P3** помимо обычного ввода и вывода информации используется для формирования и приема специальных управляющих и информационных сигналов. Разряды порта (все или частично) при этом могут выполнять следующие альтернативные функции:

Альтернативные функции могут быть активированы только в том случае, если в соответствующие биты порта P3 предварительно занесены 1. Неиспользуемые альтернативным образом разряды могут работать как обычно.

### 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

### 3.2.3. Устройство портов.

Каждый из портов содержит регистр-защелку (SFR P0 — SFR P3), выходную цепь и входной буфер.

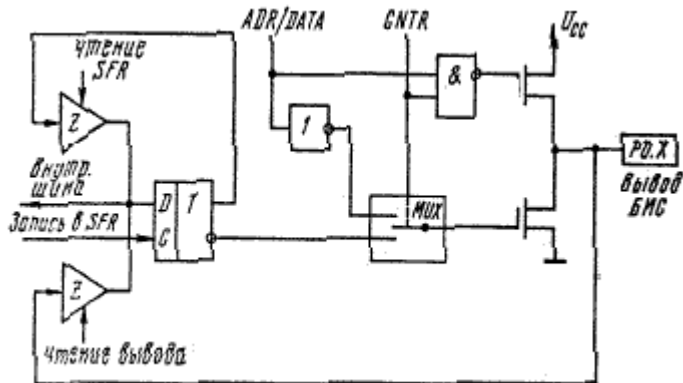


Рис.3. Порт P0

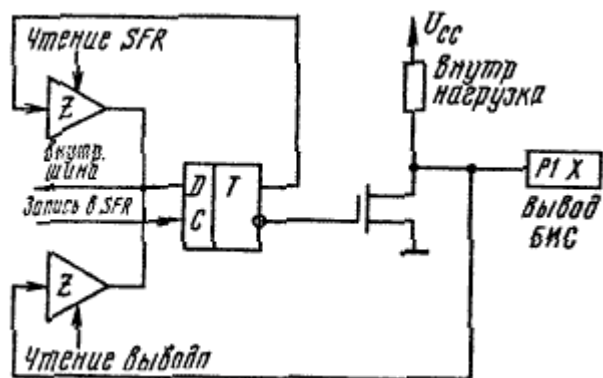


Рис.4. Порт P1

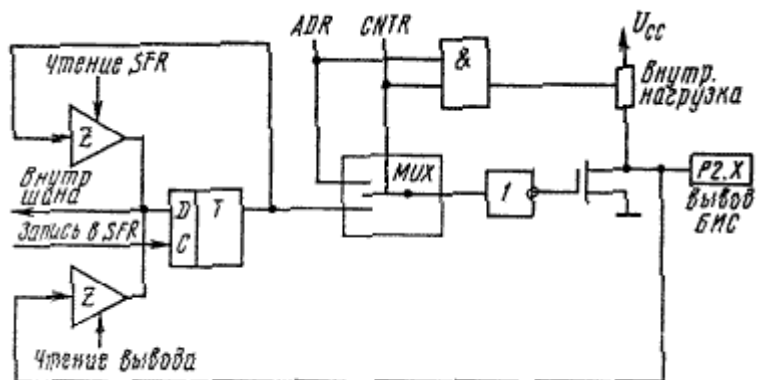


Рис.5. Порт P2

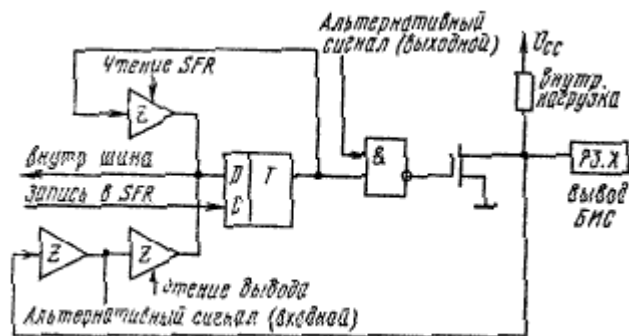


Рис.6. Порт P3

На рисунке изображены функциональные схемы регистров-защелок и буферов ввода-вывода всех портов микро-ЭВМ 8051. Каждый из разрядов регистра-защелки SFR является D-триггером, информация в который заносится с внутренней шины данных микроконтроллера по сигналу «Запись в SFR Px» (x= 0, 1, 2, 3) от центрального процессорного элемента (CPU). С прямого выхода D-триггера информация может быть выведена на внутреннюю шину по сигналу «Чтение SFR Px» от CPU, а с вывода микросхемы («из внешнего мира») по сигналу «Чтение выводов Px». Одни команды активизируют сигнал «Чтение SFR PI», другие - «Чтение выводов PI».

### 3.2.4. Особенности электрических характеристик портов.

Выходные каскады триггеров SFR портов P1 — P3 выполнены на полевых транзисторах с внутренней нагрузкой, в то время как аналогичные каскады триггеров SFR P0—на транзисторах с открытым стоком. Каждая линия любого из портов может независимо использоваться как для ввода, так и для вывода информации (для линий портов P0 и P2 это справедливо тогда, когда они не используются для обращения к внешней памяти).

Для перевода любой линии портов P1 — P3 в режим ввода информации необходимо в соответствующий разряд SFR занести 1. 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. Выходные каскады порта P0 имеют иную структуру. Нагрузочный полевой транзистор линии порта включен только тогда, когда порт выводит 1 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) выходные каскады порт P0 представляют собой ступени на транзисторах с открытым стоком. Запись 1 в соответствующий бит SFR отключает и второй транзистор, что приводит к тому, что вывод БИС оказывается под «плавающим» потенциалом. Это позволяет использовать линии порта P0 как выводы с высоко импедансным состоянием. Поскольку выходные каскады портов P1 — P3 имеют внутреннюю нагрузку, при переводе в режим ввода информации они становятся источниками тока для микросхемы или транзистора, нагруженных на данный.

### 3.3. Таймеры / счетчики микроконтроллеров семейства 8051.

В базовых моделях семейства имеются два программируемых 16-битных таймера/счетчика (T/C0 и T/C1), которые могут быть использованы как в качестве таймеров, так и в качестве счетчиков внешних событий. В первом случае содержимое соответствующего таймера/счетчика (далее для краткости T/C) инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов колебаний кварцевого резонатора, во втором оно инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (T0, T1) вывод микро-ЭВМ 8051. Так как на распознавание периода требуются два машинных цикла, максимальная частота подсчета входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входной сигнал должен удерживать значение 1, как минимум, в течение одного машинного цикла микро-ЭВМ.

Для управления режимами работы T/C и для организации их взаимодействия с системой прерываний используются два регистра специальных функций (TMOD и TCON), описание которых приведено ниже.

### 3.3.1.Регистр режима работы таймера/счетчика TMOD

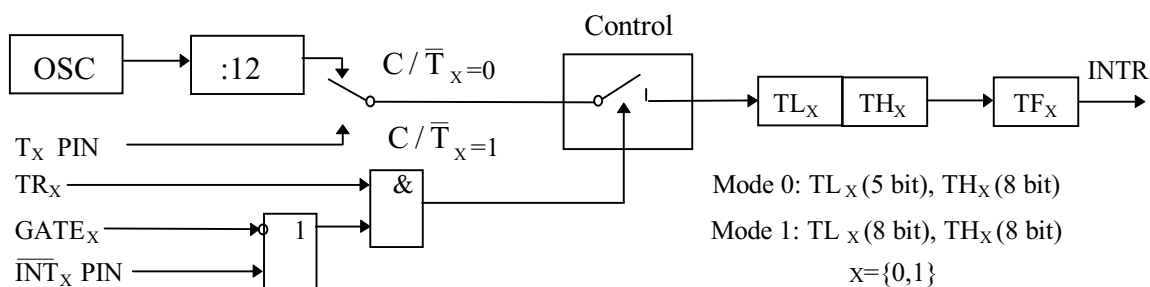
Символ	Позиция	Имя и назначение	
GATE	TMOD.7 для T/C1 и TMOD.3 для T/CO	<b>Управление блокировкой.</b> Если бит установлен, то таймер/счетчик "x" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается	
C/T	TMOD.6 для T/C1 и TMOD.2 для T/CO	<b>Бит выбора режима таймера или счетчика событий.</b> Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если; установлен, то работает счетчик от внешних сигналов на входе "Tx"	
M1	TMOD.5 для T/C1 и TMOD.1 для T/CO	<b>Режим работы</b>	
		<b>M1M0</b>	
		0 0	Таймер BE48. "TLx" работает как 5-битный предделитель
		0 1	16 битный таймер/счетчик. "THx" и "TLx" включен последовательно
		1 0	8-битный авто перезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению
M0	TMOD.4 для T/C1 и TMOD.0 для T/CO	1 1	Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TLO 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

### 3.3.2.Регистр управления/статуса таймера TCON.

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

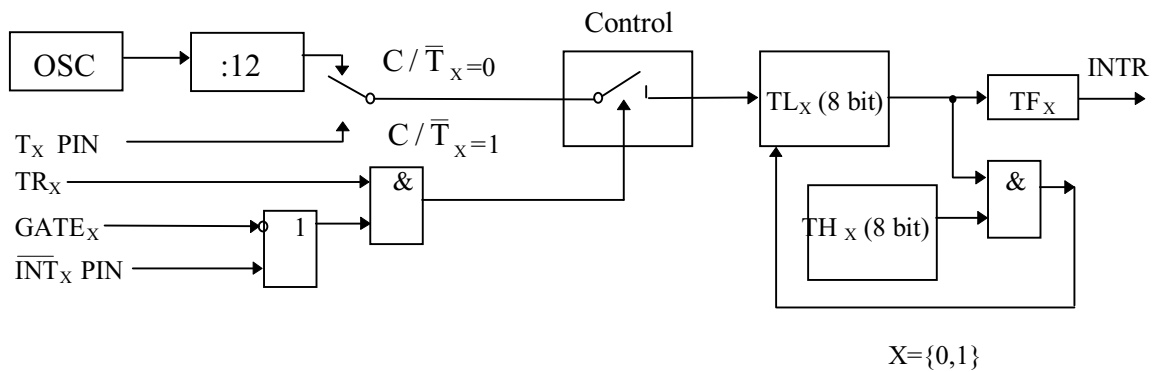
### 3.3.3.Режимы работы таймеров-счетчиков.

Как следует из описания управляющих бит TMOD, для обоих T/C режимы работы 0, 1 и 2 одинаковы. Режимы 3 для T/CO и T/C1 различны. Рассмотрим кратко работу T/C в каждом из режимов.

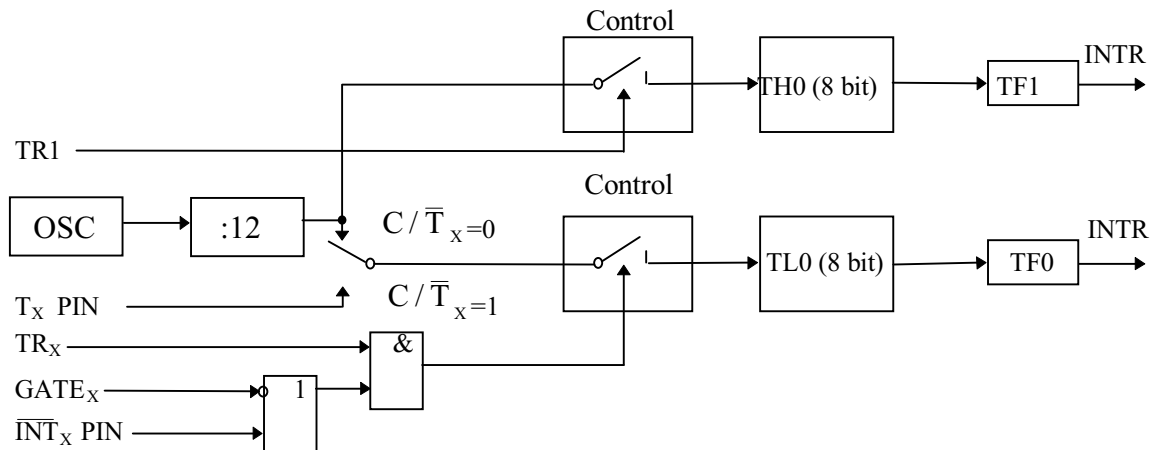


а - логика работы T/CO и T/C1 в режимах 0 и 1





б - логика работы T/C0 и T/C1 в режиме 2



в - логика работы T/C0 в режиме 3

Рис.7. Логика работы T/C0 и T/C1 в режимах 0, 1, 2 и 3

- Режим 0.** Перевод любого T/C в режим 0 делает его похожим на таймер KM1816BE48 (восьми битный счетчик), к входу которого подключен пяти-битный предделитель частоты на 32. Работу T/C в режиме 0 на примере T/C1 иллюстрирует рис а. В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния "все единицы" в состояние "все нули" устанавливается флаг прерывания от таймера TF 1. Входной синхросигнал таймера 1 разрешен (поступает на вход T/C1), когда управляющий бит TR1 установлен в 1 либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания INT1 поступает уровень 1. Отметим попутно, что установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала подаваемого на вход запроса прерывания.
- Режим 1.** Работа любого T/C в этом режиме такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.
- Режим 2.** В этом режиме работа организована таким образом, что переполнение (переход из состояния "все единицы" в состояние, "все нули") восьмибитного счетчика TL1 приводит не только к установке флага TF1 (см. рис. б), но и автоматически перезагружает в TL1 содержимое старшего байта (TH 1) таймерного регистра, которое предварительно было задано программным путем. Перегрузка оставляет содержимое TH1 неизменным. В режиме 2 T/C0 и T/C1 также работают совершенно одинаково.
- Режим 3.** В режиме 3 T/C0 и T/C1 работают по-разному. T/C1 сохраняет неизменным свое текущее содержимое. Иными словами, эффект такое же как и при сбросе управляющего бита TR1 в 0. Работу T/C0 иллюстрирует рис. в. В режиме 3 TL0 и TH0 функционируют как два независимых восьмибитных счетчика. Работу TL0 определяют управляющие биты T/C0 (C/T, GATE TR0), входной сигнал INT0 и флаг переполнения TF0. Работу TH0, который может выполнять только функции таймера (подсчёт машинных циклов микро-ЭВМ), определяет управляющий бит TR1. При этом TH0 использует флаг переполнения TF1. Режим 3 используется в тех случаях, когда требуется наличие дополнительного восьми битного таймера или счетчика событий. Можно считать, что в этом режиме микро-ЭВМ 8051 имеет в своем составе три таймера/счетчика. В случае же, если T/C0 используется в режиме 3, T/C1 может быть или выключен, или переведен в режим 0, 1 или 2, или может быть использован последовательным портом в качестве генератора частоты передачи.

В модернизированных моделях микроконтроллеров семейства MCS-51 может иметься третий таймер счетчик T/C2 и (или) блок программных счетчиков PCA, которые тоже могут быть использованы для отсчета временных интервалов.

### 3.4. Последовательный порт микроконтроллера 8051.

Через универсальный асинхронный приемопередатчик UART (Universal Asynchronous Receiver-Transmitter) осуществляются 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.

Кроме того, работой последовательного порта управляют два служебных регистра -

- Регистр управления/статуса приемопередатчика SCON
- Бит SMOD регистра управления мощностью PCON

Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буферного регистра приемника позволяет совмещать операцию чтения ранее принятого байта с приемом очередного. Но если к моменту окончания приема байта предыдущий не был считан из SBUF, то он будет потерян.

Последовательный порт 8051 может работать в четырех различных режимах.

- **Режим 0.** Информация и передается, и принимается через вывод входа приемника (RXi TXi). Принимаются или передается 8 бит данных. Через вывод выхода передатчика (TXD; выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна 1/12 частоты кварцевого резонатора
- **Режим 1.** В этом режиме передаются через вывод TXD или принимаются через 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.
- **Режим 2.** В этом режиме через вывод TXD передаются или через 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.** совпадает с режимом 2 во всех деталях, за исключением частоты приема/передачи, которая является величиной переменной и задается таймером.

Во всех случаях передача инициализируется инструкцией, в которой данные перемещаются в SBUF. Прием инициализируется при обнаружении перепада из 1 в 0 на входе приемника. При этом в режиме 0 этот переход должен сопровождаться выполнением условий  $R1 = 0$  и  $REN = 1$  (см. табл. 8), а для остальных режимов -  $REN = 1$ .

### 3.5. 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 (R1 и T1).

Прикладная программа путем загрузки в старшие биты регистра SCON двухбитного кода определяет режим работы приемопередатчика. Во всех четырех режимах работы передача инициализируется любой командой, в которой буферный регистр SBUF указан как получатель байта. Как уже отмечалось, прием в режиме 0 осуществляется при условии, что  $R1 = 0$  и  $REN = 1$ , в остальных режимах - при условии, что  $REN = 1$ .

В бите TB8 программно устанавливается значение девятого бита данных, который будет передан в режиме 2 или 3. В бите RB8 в этих режимах фиксируется 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 TL.

Флаг прерывания приемника RI устанавливается аппаратно в конце периода приема восьмого бита данных в режиме 0 и в середине периода приема стоп-бита в режимах 1, 2 и 3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

### 3.5.1. Функциональное назначение бит регистра управления/статуса приемопередатчика SCON.

Символ	Позиция	Имя и назначение															
SM0	SCON.7	Биты управления режимом работы приемопередатчика. Устанавливаются/сбрасываются программно см. примечание 1															
SM1	SCON.6	<table border="1"> <thead> <tr> <th>SM0</th> <th>SM1</th> <th>Режим работы приемопередатчика</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Сдвигающий регистр расширения ввода/вывода</td> </tr> <tr> <td>0</td> <td>1</td> <td>8 битовый приемопередатчик, изменяемая скорость передачи</td> </tr> <tr> <td>1</td> <td>0</td> <td>9 битовый приемопередатчик. Фиксированная скорость передачи</td> </tr> <tr> <td>1</td> <td>1</td> <td>9 битовый приемопередатчик, изменяемая скорость передачи</td> </tr> </tbody> </table>	SM0	SM1	Режим работы приемопередатчика	0	0	Сдвигающий регистр расширения ввода/вывода	0	1	8 битовый приемопередатчик, изменяемая скорость передачи	1	0	9 битовый приемопередатчик. Фиксированная скорость передачи	1	1	9 битовый приемопередатчик, изменяемая скорость передачи
		SM0	SM1	Режим работы приемопередатчика													
		0	0	Сдвигающий регистр расширения ввода/вывода													
		0	1	8 битовый приемопередатчик, изменяемая скорость передачи													
1	0	9 битовый приемопередатчик. Фиксированная скорость передачи															
1	1	9 битовый приемопередатчик, изменяемая скорость передачи															
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/сбрасывается 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															

### 3.5.2. Скорость приема/передачи информации через последовательный порт.

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  $f_{PE3}$ :  

$$f = f_{PE3}/12.$$

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.

### 3.5.3. Регистр управления мощностью PCON.

Символ	Позиция	Наименование и функция
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0. По сбросу SMOD = 0.
	PCON.6	Не используется
	PCON.5	Не используется
	PCON.4	Не используется
GF1 GF0	PCON.3 PCON.2	Флаги, специфицируемые пользователем (флаги общего назначения)
PD	PCON.1	Бит пониженной мощности. При установке бита в 1 микро-ЭВМ переходит в режим пониженной потребляемой мощности
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микро-ЭВМ 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 режиме 2 частота передачи определяется выражением

$$f = 2^{SMOD} f_{PE3}/64.$$

Иными словами, при SMOD = 0 частота передачи равна 1/64 частоты  $f_{PE3}$ , а при SMOD = 1 — 1/32 частоты  $f_{PE3}$ .

В режимах 1 и 3 в формировании частоты передачи, кроме управляющего бита SMOD, принимает участие таймер 1. При этом частота передачи  $f$  зависит от частоты переполнения  $f_{OVL1}$  определяется следующим образом:

$$f = 2^{SMOD} f_{OVL1}/32$$

Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам же таймер может работать как в режиме таймера, так и в режиме счетчика. Номер режима (0, 1, 2) роли не играет. Наиболее типично использование его в режиме таймера с автоперезагрузкой (старшая тетрада TMOD = 0010B). При этом частота передачи определяется выражением:

$$f = 2^{\text{SMOD}} \cdot f_{\text{PE3}} / (32 \times 12 \times (256 - \text{TH1})).$$

Частота приема/передачи (BAUD RATE)	Частота резонатора МГц	Таймер/счетчик 1			
		SMOD	C/T	Режим (MODE)	Перезагружаемое число
Режим 0, макс: 1 МГц	12	X	X	X	X
Режим 2, макс: 375 КГц	12	1	X	X	X
Режим 1, 3: 62,2 КГц	12	1	0	2	0FFH
19,2 КГц	11,059	1	0	2	0FDH
9,6 КГц	11,059	0	0	2	0FDH
4,8 КГц	11,059	0	0	2	0FAH
2,4 КГц	11,059	0	0	2	0F4H
1,2 КГц	11,059	0	0	2	0F4H
137,5 Гц	11,059	0	0	2	1DH
110 Гц	6	0	0	2	72H
110 Гц	12	0	0	1	0FEEBH

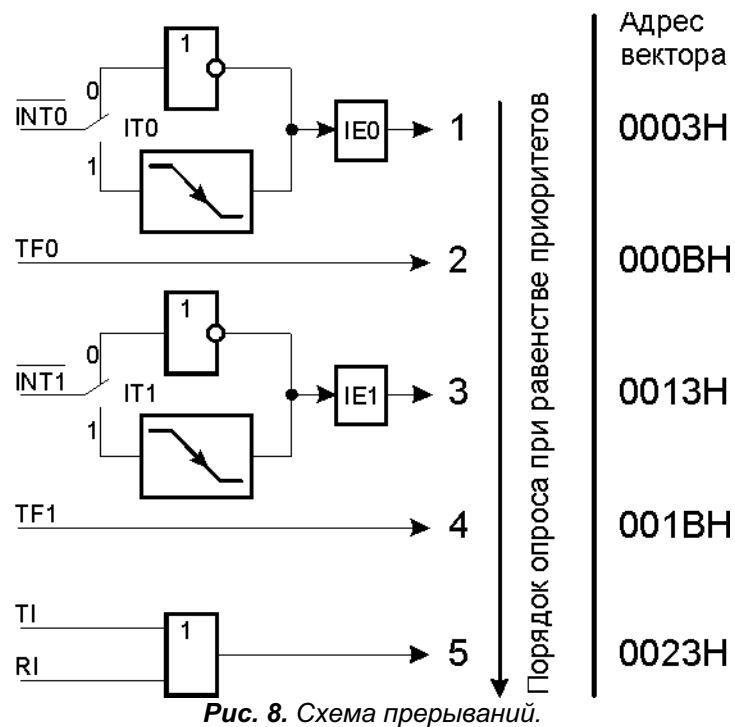
**Таблица.5.** Настройка таймера 1 для управления частотой работы приемопередатчика.

Отметим, что скорости приема и передачи могут различаться.

Предельно низких частот приемопередачи можно достичь при использовании таймера в режиме 1 (16-битный таймер) и разрешении прерываний от таймера (старший полубайт TMOD = 0001B). Перезагрузка 16-битного таймера должна осуществляться программным путем.

### 3.6. Система прерываний микроконтроллера 8051.

Упрощенная схема прерываний микро-ЭВМ 8051 показана на рисунке.



**Рис. 8.** Схема прерываний.

Внешние прерывания INT 0 и INT 1 могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах 8051 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 (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага I 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 названные 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 IE и уровнями приоритета IP. Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний 8051 исключительно гибкой.

В более сложных модификациях микроконтроллеров семейства MCS-51 количество периферийных устройств увеличено, что приводит к необходимости использовать один вектор прерывания для нескольких устройств (разделение подпрограмм обслуживания прерываний в этом случае необходимо реализовать программно), либо добавить еще два регистра - режима (маски) и приоритета прерываний.

### 3.6.1.Регистр масок прерывания (IE).

Символ	Позиция	Имя и назначение
EA	IE.7	Снятие блокировки прерывания. Сбрасывается, программно для запрета всех прерываний независимо от состояний IE.4 - IE.0
	IE.6	Не используется
	IE.5	Не используется
ES	IE.4	Бит разрешения прерывания, от приемопередатчика Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI .
ET1	IE.3	Бит разрешения прерывания от таймера. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерывания 1
ET0	IE.1	Бит разрешения прерывания от таймера 0. Установка/сброс программой для разрешения/запрета прерываний от таймера 0 .
EX0	IE.0	Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерывания 0

### 3.6.2.Регистр приоритетов прерываний (IP).

Символ	Позиция	Имя и назначение
-	IP.7 - IP.5	Не используется
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 высшего/низшего приоритета внешнему прерыванию INT1
PT0	IP.1	Бит приоритета таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета
PX0	IP.0	Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT0

### 3.6.3.Выполнение подпрограммы прерывания.

Система прерываний формирует аппаратный вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована одним из следующих условий:

- в данный момент обслуживается запрос прерывания равного или высокого уровня приоритета;
- текущий машинный цикл — не последний в цикле выполняемой команды;
- выполняется команда RETI или любая команда, связанная с обращением к регистрам IE или IP

Отметим, что если флаг прерывания был установлен, но по одному из указанных выше условий не получил обслуживания и к моменту окончания блокировки уже сброшен, то запрос прерывания теряется и нигде не запоминается.

По аппаратно сформированному коду LCALL система прерывания помещает в стек только содержимое счетчика команд (PC) и загружает в него адрес вектора соответствующей подпрограммы обслуживания. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. В случае необходимости она 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 снимают блокировку прерываний,

что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания.

### 3.7. Работа с внешней памятью микроконтроллера 8051.

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 программой, используется сигнал PSEN, во втором — сигналы RD и WR, активизирующие ОЗУ с данными.

Если используется 16-битовый адрес, старшие восемь бит выводятся через порт P2, где они сохраняются в течение всего цикла обращения к внешней памяти. Отметим, что выходные каскады порта P2 имеют внутреннюю нагрузку, несколько отличающуюся от P1 и P3, благодаря чему в SFR P2 при выводе адресной информации вовсе не обязательно защелкивать все единицы. Добавим также, что при выводе адресной информации информация из SFR P2, хотя и не присутствует на выводах микроЭВМ, но и не теряется, восстанавливаясь на них после окончания обращений к внешней памяти (если в процессе этих обращений SFR P2 не был модифицирован).

Если при обращении к внешней памяти данных используется восьми 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, на выводах порта P0 младший байт адреса мультиплексируется с данными. Сигналы адреса/данных задействуют оба полевых транзистора выходного каскада порта P0. Таким образом, 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 непосредственно перед активацией 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 0FFH в SFR P0, уничтожая, таким образом, хранимую там информацию. Таким образом, использовать для записи порт P0 при работе с внешней памятью надо с известной долей осторожности.

Обращение к внешней памяти программ возможно в двух случаях:

- когда сигнал EA активен, т.е. имеет нулевой уровень,
- когда программный счетчик PC содержит число больше 0FFH.

Следовательно, при использовании микро-ЭВМ, не имеющей встроенного ПЗУ или не использующей его, на входе EA должен присутствовать сигнал с нулевым уровнем.

Когда CPU работает с внешней памятью программ, все линии порта 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 — адресная информация, выводимая через P2, не зависит от состояния его SFR.

### 3.8. Режимы микроконтроллера 8051 с пониженным энергопотреблением.

Во многих вариантах применения микро-ЭВМ энергопотребление является одним из основных параметров. В этих случаях целесообразно использовать КМОП версии микро-ЭВМ. В них предусмотрены дополнительные возможности снижения энергопотребления, отсутствующие в стандартных n-МОП изделиях. Ранее выпускались и n-МОП версии микро-ЭВМ, имевшие режимы работы с пониженным энергопотреблением. В настоящее время их выпуск почти повсеместно прекращен. Однако время от времени они все же будут попадать в руки радиолюбителей, поэтому трудно гарантировать наличие или отсутствие этих режимов в тех или иных конкретных n-МОП изделиях.

КМОП микро-ЭВМ имеют два режима с пониженным потреблением тока: режим холостого хода (XX) и режим выключенного напряжения питания (ВНП). Последний иногда называют режимом пониженного энергопотребления. По мнению авторов настоящей статьи, это вряд ли корректно, так как режим XX также характеризуется пониженным энергопотреблением в сравнении с обычным режимом работы микро-ЭВМ. Дословно второй режим, называемый в англоязычной литературе режимом "Power Down Mode", можно перевести так, как указано выше. Этот перевод корректнее отражает суть режима - в нем допустимо вообще отключить от микро-ЭВМ питающее напряжение, подаваемое на вход Ucc (вывод 40 микро-ЭВМ). для сохранения содержимого внутренней ОЗУ в этом случае нужно запитать от резервного источника. Резервное питающее напряжение подают через вход RST (вывод 9).

В режиме XX (IDL = 1) генератор микро-ЭВМ работает, тактовые сигналы поступают на систему прерываний, последовательный порт и таймеры/счетчики. Все регистры сохраняют свое содержимое, на выводах всех портов удерживаются логические состояния, которые были на них в момент перехода в режим XX. Однако сигнал генератора синхронизирующий CPU, отключается. В режиме ВНП (PD = 1) генератор останавливается. Прекращается тактирование не только CPU, но и последовательного порта, таймеров/счетчиков, системы прерываний. Как и в режиме XX, состояние регистров, резидентного ОЗУ и выводов портов остается неизменным.

Режимы ХХ и ВНП активизируются при установке соответствующих битов в специальном функциональном регистре-регистре управления мощностью PCON. Адрес этого регистра 87H, его описание приведено в табл.

Отметим следующие особенности этого регистра. Если одновременно установлены в 1 биты IDL и PD, то последний имеет преимущество - микро-ЭВМ переходит в режим ВНП. Содержимое регистра PCON после сброса - 0XXX0000.

В n-МОП версиях микроЭВМ регистр PCON содержит, как правило, только SMOD. Остальные четыре бита присутствуют только в КМОП устройствах. Пользовательские программы никогда не должны заносить 1 в незанятые биты (PCON4 - PCON6), так как они могут использоваться в следующих модификациях микроЭВМ. Последнее распространяется на все регистры и на все адреса в области регистров специальных функций, которые не заняты регистрами. Разработчики предполагают использовать их в новых изделиях. Однако для достижения совместимости уже разработанного программного обеспечения с новыми изделиями дополнительные возможности последних будут включаться установкой в 1 битов в соответствующих регистрах. Поэтому пользовательская программа, устанавливающая в 1 неиспользуемые биты, будет нормально работать на микро-ЭВМ, имеющихся в наличии сегодня, но вовсе не обязательно будет работать на новых микро-ЭВМ, несмотря на их полную программную совместимость с семейством 8051.

### 3.8.1.Режим ХХ.

В этот режим микро-ЭВМ переводится любой командой, устанавливающей в 1 бит PCON.0. Она оказывается последней в цепочке выполняемых команд: в режиме ХХ выполнение программы приостанавливается, так как на CPU перестает поступать сигнал тактового генератора. Однако содержимое внутреннего ОЗУ и регистров специальных функций остается неизменным, выходы портов удерживают значения, которые были на них до перехода в режим ХХ, на таймеры/счетчики, приемопередатчик и на систему прерываний продолжают поступать тактовые сигналы. На выводах ALE и PSEN устанавливаются сигналы единичного уровня.

Состояние выводов портов зависит от типа ОЗУ, с которым микро-ЭВМ обменивалась информацией перед тем как перейти в режим ХХ. 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.

Вывести микро-ЭВМ из режима ХХ можно двумя способами. Так, вызов любого из прерываний приведет к аппаратному стиранию бита PCON.0, прекращающему ХХ. Прерывание будет обслужено, и очередной после RETI выполняемой командой будет та, которая следует за командой, приведшей к переходу микро-ЭВМ в режим ХХ. флаги GF0 и GF1 могут использоваться для индикации того, произошло ли прерывание во время нормальной работы или во время ХХ. Например, команда, запускающая этот режим, может также устанавливать один или оба флага. Когда ХХ прекращен прерыванием, сервисная программа прерывания может проверять состояние флагов.

Другой способ прекращения ХХ - 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. В этот момент CPU возобновляет выполнение программы с самого начала. Как показано на рис. 14, перед началом обработки 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 ХХ, не должна быть командой, записывающей информацию в SFR порта или во внешнее ОЗУ данных. Напомним, что после аппаратного сброса содержимое SFR переопределяется.

### 3.8.2.Режим ВНП.

Команда, устанавливающая в 1 бит PCON.1, переводит микро-ЭВМ в режим ВНП. В нем генератор микро-ЭВМ, как отмечалось, останавливается, прекращает функционирование не только CPU, но и таймеры/счетчики, приемопередатчик, система прерываний. При наличии основного или резервного источника питающего напряжения встроенное ОЗУ и регистры SFR сохраняют свое содержимое. Состояние портов не отличается от состояния при переходе в режим ХХ (см. выше). Однако в отличие от режима ХХ на выводах ALE и PSEN устанавливаются сигналы с нулевым уровнем.

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

В рассматриваемом режиме напряжение питания Ucc может снижаться вплоть до 2 В. Следует, однако, позаботиться, чтобы снижалось оно не раньше перехода микроЭВМ в режим ВНП и восстанавливалось до прежнего значения прежде, чем она выйдет из этого режима. Сигнал сброса, возвращающий микро-ЭВМ в обычный режим, не должен подаваться раньше, чем Ucc достигнет своего рабочего значения, и должен поддерживаться в активном состоянии достаточно долго, чтобы генератор успел запуститься и его колебания стабилизировались (обычно - не менее 10 мс).

## 4. Система команд микроконтроллера семейства 8051.

### 4.1.1. Общая характеристика.

Микро-ЭВМ рассматриваемого семейства являются типичными микропроцессорными устройствами с архитектурой SISC - со стандартным набором команд. Поэтому их система команд довольно обширна и включает в себя 111 основных команд. Их длина – один, два или три байта, причем большинство из них (94%) – одно- или двухбайтные. Все команды выполняются за один или два машинных цикла (соответственно 1 или 2 мкс при тактовой частоте 12 МГц), исключение – команды умножения и деления, которые выполняются за четыре машинных цикла (4 мкс). Микро-ЭВМ семейства 8051 используют прямую, непосредственную, косвенную и неявную, адресацию данных

В качестве операндов команд микро-ЭВМ семейства 8051 могут использовать отдельные биты, четырехбитные цифры, байты и двухбайтные слова.

Все эти черты обычны для набора команд любого SISC-процессора и по сравнению с RISC набором команд обеспечивает большую компактность программного кода и увеличение быстродействия при выполнении сложных операций.

В то же время, набор команд семейства 8051 имеет несколько особенностей, связанных с типичными функциями выполняемыми микроконтроллерами - управлением, для которого типичным является оперирование с одноразрядными двоическими сигналами, большое число операций ввода вывода и ветвлений программы.

Наиболее существенная особенность системы команд рассматриваемых микро ЭВМ это возможность адресации отдельных бит в резидентной памяти данных. Кроме того, как отмечалось, некоторые регистры блока регистров специальных функций также допускают адресацию отдельных бит. Карты адресов отдельных бит в резидентной памяти данных и в блоке регистров специальных функций.

### 4.1.2. Типы команд

Всего микро-ЭВМ выполняют 13 типов команд, они приведены в таблице. Как следует из нее, первый байт команды всегда содержит код операции (КОП), а второй и третий (если они присутствуют в команде) – адреса операндов или их непосредственные значения.

Тип команды	Первый байт D7...D0	Второй байт D7...D0	Третий байт D7...D0
тип 1	коп		
тип 2	коп	#d	
тип 3	коп	ad	
тип 4	коп	bit	
тип 5	коп	rel	
тип 6	коп	a7...a0	
тип 7	коп	ad	#d
тип 8	коп	ad	rel
тип 9	коп	ads	add
тип 10	коп	#d	rel
тип 11	коп	bit	rel
тип 12	коп	ad16h	ad16l
тип 13	коп	#d16h	#d16l

Таблица. 6. Типы команд

### 4.1.3. Типы операндов

Состав операндов включает в себя операнды четырех типов: биты, 4-битные цифры, байты и 16-битные слова.

Микроконтроллер имеет 128 программно-управляемых флагов пользователя. Имеется также возможность адресации отдельных битов блока регистров специальных функций и портов. Для адресации битов используется прямой 8-битный адрес (bit). Косвенная адресация битов невозможна. Карты адресов отдельных битов представлены на рис. 6 .

Четырехбитные операнды используются только при операциях обмена SWAP и XCHD.

Восьмибитным операндом может быть ячейка памяти программ (ПП) или данных (резидентной (РПД) или внешней (ВПД)), константа (непосредственный операнд), регистры специальных функций, а также порты ввода/вывода. Порты и регистры специальных функций адресуются только прямым способом. Байты памяти могут адресоваться также и косвенным образом через адресные регистры R0, R1, DPTR и PC.

Двухбайтные операнды - это константы и прямые адреса, для представления которых используются второй и третий байты команды.



Адреса	(D <sub>7</sub> )								(D <sub>0</sub> )
7FH									
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	
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									

**Рис. 9.** Карта адресуемых битов в резидентной памяти данных

#### 4.1.4. Группы команд.

Система команд семейства MCS-51 содержит 111 базовых команд, которые по функциональному признаку можно подразделить на пять:

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

Формат команд - одно-, двух- и трехбайтовый, причем большинство команд (94) имеют формат один или два байта. Первый байт любых типа и формата всегда содержит код операции, второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

Состав операндов включает в себя операнды четырех типов: биты, ниблы (4 разряда), байты и 16-битные слова. Время исполнения команд составляет 1, 2 или 4 машинных цикла. При тактовой частоте 12 мГц длительность машинного цикла составляет 1 мкс, при этом 64 команды исполняются за 1 мкс, 45 команд - за 2 мкс и 2 команды (умножение и деление) - за 4 мкс.

Набор команд MCS-51 поддерживает следующие режимы адресации.

**Прямая адресация (Direct Addressing).** Операнд определяется 8-битным адресом в инструкции. Эта адресация используется только для внутренней памяти данных и регистров SFR.

**Косвенная адресация (Indirect Addressing).** В этом случае инструкция адресует регистр, содержащий адрес операнда. Данный вид адресации может применяться при обращении как к внутреннему, так и внешнему ОЗУ. Для указания 8-битных адресов могут использоваться регистры R0 и R1 выбранного регистрового банка или указатель стека SP.

Для 16-битной адресации используется только регистр "указатель данных" (DPTR - Data Pointer).

*Регистровая адресация (Register Instruction).* Данная адресация применяется для доступа к регистрам R0+R7 выбранного банка. Команды с регистровой адресацией содержат в байте кода операции трехбитовое поле, определяющее номер регистра. Выбор одного из четырех регистровых банков осуществляется программированием битов селектора банка (RS1, RS0) в PSW.

*Непосредственная адресация (Immediate constants).* Операнд содержится непосредственно в поле команды вслед за кодом операции и может занимать один или два байта ( $data_8$ ,  $data_{16}$ ).

*Индексная адресация (Indexed Addressing).* Индексная адресация используется при обращении к памяти программ и только при чтении. В этом режиме осуществляется просмотр таблиц в памяти программ. 16-битовый регистр (DPTR или PC) указывает базовый адрес требуемой таблицы, а аккумулятор указывает на точку входа в нее. Адрес элемента таблицы находится сложением базы с индексом (содержимым аккумулятора).

Другой тип индексной адресации применяется в командах "перехода по выбору" (Case Jump). При этом адрес перехода вычисляется как сумма указателя базы и аккумулятора.

*Неявная адресация (Register-Specific Instructions).* Некоторые инструкции используют индивидуальные регистры (например, операции с аккумулятором, DPTR), при этом данные регистры не имеют адреса, указывающего на них; это заложено в код операции

#### 4.1.5. Обозначения, используемые при описании команд.

Rn (n = 0, 1, ..., 7) – регистр общего назначения в выбранном банке регистров;

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

ad – адрес прямоадресуемого байта;

ads – адрес прямо адресуемого байта-источника;

add – адрес прямо адресуемого байта-получателя;

ad11 – 11-разрядный абсолютный адрес перехода;

ad16 – 16-разрядный абсолютный адрес перехода;

rel – относительный адрес перехода;

#d – непосредственный операнд;

#d16 – непосредственный операнд (2 байта);

bit – адрес прямо адресуемого бита;

/bit – инверсия прямо адресуемого бита;

A - аккумулятор;

PC – счетчик команд;

DPTR – регистр указатель данных;

() – содержимое ячейки памяти или регистра,

#### 4.1.6. Команды пересылки данных микроконтроллера 8051.

Эта группа представлена 28 командами, их краткое описание приведено в таблице, где также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц).

Таблица.7. Команды передачи данных

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
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	11100101	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	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	3	2	2	(Rn) ← (ad)
Загрузка в регистр константы	MOV Rn, #d	01111rrr	2	2	1	(Rn) ← #d
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101	3	2	1	(ad) ← (A)
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr	3	2	2	(ad) ← (Rn)
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101	9	3	2	(add) ← (ads)
Пересылка байта из РПД по прямому адресу	MOV ad, @Ri	1000011i	3	2	2	(ad) ← ((Ri))
Пересылка по прямому адресу константы	MOV ad, #d	01110101	7	3	2	(ad) ← #d
Пересылка в РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)
Пересылка в РПД константы	MOV @Ri, #d	0111011i	2	2	1	((Ri)) ← #d
Загрузка указателя данных	MOV DPTR, #d16	10010000	13	3	2	(DPTR) ← #d16
Пересылка в аккумулятор байта из ПП	MOVC A, @A+DPTR	10010011	1	1	2	← ((A) +(DPTR))
Пересылка в аккумулятор байта из ПП	MOVC A, @A+PC	10000011	1	1	2	(PC) ← (PC)+1, (A) ← ((A)+(PC))

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	1110001i	1	1	2	(A) ← ((Ri))
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11100000	1	1	2	(A) ← ((DPTR))
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	2	((Ri)) ← (A)
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) ← (A)
Загрузка в стек	PUSH ad	11000000	3	2	2	(SP) ← (SP) + 1, ((SP)) ← (ad)
Извлечение из стека	POP ad	11010000	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	3	2	1	(A) ↔ (ad)
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмен младших тетрад аккумулятора и байта РПД	XCHD A, @Ri	1101011i	1	1	1	(A <sub>0...3</sub> ) ↔ ((Ri) <sub>0...3</sub> )

По команде MOV выполняется пересылка данных из второго операнда в первый. Эта команда не имеет доступа ни к внешней памяти данных, ни к памяти программ. Для этих целей предназначены команды MOVX и MOVC соответственно. Первая из них обеспечивает чтение/запись байт из внешней памяти данных, вторая – 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).

Команды PUSH и POP предназначены соответственно для записи данных в стек и их чтения из стека. Размер стека ограничен лишь размером резидентной памяти данных. В процессе инициализации микро-ЭВМ после сигнала сброса или при включении питающего напряжения в SP заносится код 07H. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом 08H.

Группа команд пересылок микроконтроллера имеет следующую особенность - в ней нет специальных команд для работы со специальными регистрами: PSW, таймером, портами ввода-вывода. Доступ к ним, как и к другим регистрам специальных функций, осуществляется заданием соответствующего прямого адреса, т.е. это команды обычных пересылок, в которых вместо адреса можно ставить название соответствующего регистра. 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).

Кроме того, следует отметить, что в микро-ЭВМ аккумулятор имеет два различных имени в зависимости от способа адресации: A – при неявной адресации (например, MOV A, R0) и ACC – при использовании прямого адреса. Первый способ предпочтительнее, однако, не всегда применим.

#### 4.1.7. Команды арифметических операций 8051.

В данную группу входят 24 команды, краткое описание которых приведено в таблице. Из нее следует, что микроЭВМ выполняет достаточно широкий набор команд для организации обработки целочисленных данных, включая команды умножения и деления.

В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц).

**Таблица.8. Арифметические операции.**

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
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	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	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	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	2	2	1	(A) ← (A) + #d + (C)
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если (A <sub>0...3</sub> ) > 9 или ((AC)=1), то (A <sub>0...3</sub> ) ← (A <sub>0...3</sub> ) + 6, затем если (A <sub>4...7</sub> ) > 9 или ((C)=1), то (A <sub>4...7</sub> ) ← (A <sub>4...7</sub> ) + 6
Вычитание из аккумулятора регистра и заёма	SUBB A, Rn	10011rrr	1	1	1	(A) ← (A) - (C) - (Rn)
Вычитание из аккумулятора прямоадресуемого байта и заема	SUBB A, ad	10010101	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))
Вычитание из аккумулятора константы и заема	SUBB A, d	10010100	2	2	1	(A) ← (A) - (C) - #d
Инкремент аккумулятора	INC A	00000100	1	1	1	(A) ← (A) + 1

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Инкремент регистра	INC Rn	00001rrr	1	1	1	$(Rn) \leftarrow (Rn) + 1$
Инкремент прямоадресуемого байта	INC ad	00000101	3	2	1	$(ad) \leftarrow (ad) + 1$
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
Инкремент указателя данных	INC DPTR	10100011	1	1	2	$(DPTR) \leftarrow (DPTR) + 1$
Декремент аккумулятора	DEC A	00010100	1	1	1	$(A) \leftarrow (A) - 1$
Декремент регистра	DEC Rn	00011rrr	1	1	1	$(Rn) \leftarrow (Rn) - 1$
Декремент прямоадресуемого байта	DEC ad	00010101	3	2	1	$(ad) \leftarrow (ad) - 1$
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
Умножение аккумулятора на регистр B	MUL AB	10100100	1	1	4	$(B)(A) \leftarrow (A) \cdot (B)$
Деление аккумулятора на регистр B	DIV AB	10000100	1	1	4	$(B).(A) \leftarrow (A)/(B)$

По результату выполнения команд ADD, ADDC, SUBB, MUL и DIV устанавливаются флаги PSW, структура которых приведена в таблице.

Флаг C устанавливается при переносе из разряда D7, т. е. в случае, если результат не помещается в восемь разрядов; флаг AC устанавливается при переносе из разряда D3 в командах сложения и вычитания и служит для реализации десятичной арифметики. Этот признак используется командой DAA.

Флаг OV устанавливается при переносе из разряда D6, т. е. в случае, если результат не помещается в семь разрядов и восьмой не может быть интерпретирован как знаковый. Этот признак служит для организации обработки чисел со знаком.

Наконец, флаг P устанавливается и сбрасывается аппаратно. Если число единичных бит в аккумуляторе нечетно, то  $P = 1$ , в противном случае  $P = 0$ .

#### 4.1.8. Команды логических операций микроконтроллера 8051.

В этой группе 25 команд, их краткое описание приведено в таблице. Нетрудно видеть, что эти команды позволяют выполнять операции над байтами: логическое И ( $\wedge$ ), логическое ИЛИ ( $\vee$ ), исключающее ИЛИ ( $\oplus$ ), инверсию (NOT), сброс в нулевое значение и сдвиг. В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц).

Таблица.9. Логические операции

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) \leftarrow (A) \text{ AND } (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101	3	2	1	$(A) \leftarrow (A) \text{ AND } (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) \leftarrow (A) \text{ AND } ((Ri))$
Логическое И аккумулятора и константы	ANL A, #d	01010100	2	2	1	$(A) \leftarrow (A) \text{ AND } \#d$
Логическое И прямоадресуемого байта и аккумулятора	ANL ad, A	01010010	3	2	1	$(ad) \leftarrow (ad) \text{ AND } (A)$
Логическое И прямоадресуемого байта и константы	ANL ad, #d	01010011	7	3	2	$(ad) \leftarrow (ad) \text{ AND } \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) \leftarrow (A) \text{ OR } (Rn)$
Логическое ИЛИ аккумулятора и прямоадресуемого байта	ORL A, ad	01000101	3	2	1	$(A) \leftarrow (A) \text{ OR } (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) \leftarrow (A) \text{ OR } ((Ri))$
Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100	2	2	1	$(A) \leftarrow (A) \text{ OR } \#d$
Логическое ИЛИ прямоадресуемого байта и аккумулятора	ORL ad, A	01000010	3	2	1	$(ad) \leftarrow (ad) \text{ OR } (A)$
Логическое ИЛИ прямоадресуемого байта и константы	ORL ad, #d	01000011	7	3	2	$(ad) \leftarrow (ad) \text{ OR } \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) \leftarrow (A) \text{ XOR } (Rn)$
Исключающее ИЛИ аккумулятора и прямоадресуемого байта	XRL A, ad	01100101	3	2	1	$(A) \leftarrow (A) \text{ XOR } (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) \leftarrow (A) \text{ XOR } ((Ri))$
Исключающее ИЛИ аккумулятора и константы	XRL A, #d	01100100	2	2	1	$(A) \leftarrow (A) \text{ XOR } \#d$
Исключающее ИЛИ прямоадресуемого байта и аккумулятора	XRL ad, A	01100010	3	2	1	$(ad) \leftarrow (ad) \text{ XOR } (A)$
Исключающее ИЛИ прямоадресуемого байта и константы	XRL ad, #d	01100011	7	3	2	$(ad) \leftarrow (ad) \text{ XOR } \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) \leftarrow 0$
Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) \leftarrow \text{NOT}(A)$
Сдвиг аккумулятора влево циклический	RL A	00100011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0\div 6, (A_0) \leftarrow (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0\div 6, (A_0) \leftarrow (C), (C) \leftarrow (A_7)$
Сдвиг аккумулятора вправо циклический	RR A	00000011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0\div 6, (A_7) \leftarrow (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0\div 6, (A_7) \leftarrow (C), (C) \leftarrow (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0...3}) \leftrightarrow (A_{4...7})$

#### 4.1.9. Команды операций над битами микроконтроллера 8051.

Группа состоит из 12 команд, краткое описание которых приведено в таблице. Эти команды позволяют выполнять операции над отдельными битами: сброс, установку, инверсию бита, а также логические И ( $\wedge$ ) и 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.

В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц).

**Таблица.10. Операции с битами**

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сброс переноса	CLR C	11000011	1	1	1	(C) ← 0
Сброс бита	CLR bit	11000010	4	2	1	(b) ← 0
Установка переноса	SETB C	11010011	1	1	1	(C) ← 1
Установка бита	SETB bit	11010010	4	2	1	(b) ← 1
Инверсия переноса	CPL C	10110011	1	1	1	(C) ← NOT(C)
Инверсия бита	CPL bit	10110010	4	2	1	(b) ← NOT(b)
Логическое И бита и переноса	ANL C, bit	10000010	4	2	2	(C) ← (C) AND (b)
Логическое И инверсии бита и переноса	ANL C, /bit	10110000	4	2	2	(C) ← (C) AND (NOT(b))
Логическое ИЛИ бита и переноса	ORL C, bit	01110010	4	2	2	(C) ← (C) OR (b)
Логическое ИЛИ инверсии бита и переноса	ORL C, /bit	10100000	4	2	2	(C) ← (C) OR (NOT(b))
Пересылка бита в перенос	MOV C, bit	10100010	4	2	1	(C) ← (b)
Пересылка переноса в бит	MOV bit, C	10010010	4	2	2	(b) ← (C)

#### 4.1.10. Команды передачи управления микроконтроллера 8051.

Группа представлена командами безусловного и условного переходов, командами вызова подпрограмм и командами возврата из подпрограмм.

В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц).

**Таблица.11. Команды передачи управления**

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Длинный переход в полном объеме ПП	LJMP ad16	00000010	12	3	2	(PC) ← ad16
Абсолютный переход внутри страницы в 2 Кб	AJMP ad11	a <sub>10</sub> a <sub>9</sub> a <sub>8</sub> 00001	6	2	2	(PC) ← (PC) + 2, (PC <sub>0-10</sub> ) ← ad11
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000	5	2	2	(PC) ← (PC) + 2, (PC) ← (PC) + rel
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	2	(PC) ← (A) + (DPTR)
Переход, если аккумулятор равен нулю	JZ rel	01100000	5	2	2	(PC) ← (PC) + 2, если (A)=0, то (PC) ← (PC) + rel
Переход, если аккумулятор не равен нулю	JNZ rel	01110000	5	2	2	(PC) ← (PC) + 2, если (A) ≠ 0, то (PC) ← (PC) + rel
Переход, если перенос равен единице	JC rel	01000000	5	2	2	(PC) ← (PC) + 2, если (C)=1, то (PC) ← (PC) + rel
Переход, если перенос равен нулю	JNC rel	01010000	5	2	2	(PC) ← (PC) + 2, если (C)=0, то (PC) ← (PC) + rel
Переход, если бит равен единице	JB bit, rel	00100000	11	3	2	(PC) ← (PC) + 3, если (b)=1, то (PC) ← (PC) + rel
Переход, если бит равен нулю	JNB bit, rel	00110000	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	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	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	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	8	3	2	(PC) ← (PC) + 3, если (A) ≠ (ad), то (PC) ← (PC) + rel, если (A) < (ad), то (C) ← 1, иначе (C) ← 0
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100	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	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 РГД с константой и	CJNE @Ri, #d, rel	1011011i	10	3	2	(PC) ← (PC) + 3, если ((Ri)) ≠ #d, то

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
переход, если не равно						$(PC) \leftarrow (PC) + rel, \text{если } ((Ri)) < \#d, \text{ то } (C) \leftarrow 1, \text{ иначе } (C) \leftarrow 0$
Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	2	$(PC) \leftarrow (PC) + 3, (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{0..7}), (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{8..15}), (PC) \leftarrow ad16$
Абсолютный вызов подпрограммы в пределах страницы в 2 Кб	ACALL ad11	a <sub>10</sub> a <sub>9</sub> a <sub>8</sub> 10001	6	2	2	$(PC) \leftarrow (PC) + 2, (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{0..7}), (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{8..15}), (PC_{0-10}) \leftarrow ad11$
Возврат из подпрограммы	RET	00100010	1	1	2	$(PC_{8..15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1, (PC_{0..7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	$(PC_{8..15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1, (PC_{0..7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Пустая операция	NOP	00000000	1	1	1	$(PC) \leftarrow (PC) + 1$

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 2 Кбайт, при этом надо иметь в виду, что сначала содержимое счетчика команд увеличивается на 2 и только потом заменяются 11 разрядов адреса.

В отличие от предыдущих команд, в команде SJMP (S – short – короткий) указан не абсолютный, а относительный адрес перехода. Величина смещения rel рассматривается как число со знаком, а, следовательно, переход возможен в пределах – 128...+127 байт относительно адреса команды, следующей за командой SJMP.

Команда косвенного перехода JMP @A+DPTR позволяет вычислять адрес перехода в процессе выполнения самой программы.

Командами условного перехода можно 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 — прямо адресуемый бит равен 1
- JNB — прямо адресуемый бит равен 0;
- JBC — прямо адресуемый бит равен 1 и сбрасывается в нулевое значение при выполнении команды.

Все команды условного перехода рассматриваемых микро-ЭВМ содержат короткий относительный адрес, т. е. переход может осуществляться в пределах—128... +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 1 и проверяется на 0. Если значение содержимого счетчика не равно 0, то осуществляется переход на начало цикла, в противном случае выполняется следующая команда.

Команда CJN 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 использованы, например, значения содержимого аккумулятора и прямо адресуемого байта или косвенно адресуемого байта и константы. При выполнении команды значения указанных двух байт сравниваются и в случае, если они не одинаковы, осуществляется переход. Например, команда

WAIT: CJNE A, P0, WAIT

будет выполняться до тех пор, пока значения на линиях порта P0 не совпадут со значениями содержимого аккумулятора.

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

Команда возврата из подпрограммы RET восстанавливает из стека значение содержимого счетчика команд, а команда возврата из процедуры обработки прерывания RETI, кроме того, разрешает прерывание обслуживаемого уровня. Команды RET и RETI не различают, какой командой – LCALL или ACALL – была вызвана подпрограмма, так как и в том, и в другом случае в стеке сохраняется полный 16-разрядный адрес возврата.

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 "длины" перехода или вызова.

## 5.Дополнительные устройства в расширенных вариантах микроконтроллера.

В процессе совершенствования появились усовершенствованные модели микроконтроллеров семейства MCS-51, оснащенные дополнительными периферийными устройствами.

### 5.0.1.Расширения микропроцессоров семейства MCS-51/52.

Наряду с созданием сложных и высокоинтегрированных схем, Intel также совершенствует микросхемы, выпуск которых был освоен давно, например, 8-раз-рядные микроконтроллеры или однокристалльные ЭВМ из семейства MCS-51 (отечественный аналог КМ1816ВЕ51). Эти микросхемы хорошо зарекомендовали себя в недорогих и сравнительно несложных устройствах. Основными направлениями модернизации данных Микроконтроллеров являются:

- увеличение внутренней памяти программ до 32К, причем она может быть масочной, однократно программируемой или с ультрафиолетовым стиранием;
- снижение потребляемой мощности путем применения КМОП-технологии и специальных режимов пониженного энергопотребления;
- увеличение тактовой частоты до 20 МГц;
- модификация режимов работы счетчиков-таймеров и последовательного порта;
- размещение на кристалле дополнительного оборудования.

Эти микроконтроллеры выпускаются в PLCC, DIP и QFP корпусах и могут работать в следующих температурных диапазонах:

- коммерческий (0°C — +70°C);
- расширенный (-40°C — +85°C);
- для военного использования (-55°C — +125°C).

Все микроконтроллеры из семейства MCS-51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

### 5.0.2.8052.

В состав этого семейства входят микро-ЭВМ 80С52, 80С54, 80С58 с масочно программируемым ПЗУ версии 87С52, 87С54 и 87С58 с ПЗУ, стираемым УФ облучением, а также микро-ЭВМ 80С32, не имеющая внутреннего ПЗУ. Между собой они различаются также корпусами, рабочими интервалами температур, предельно допустимой тактовой частотой и рядом других параметров, отражаемыми в буквенно-цифровой информации после обозначения типа микро-ЭВМ. Эту информацию можно получить из фирменных руководств Intel, AMD и других производителей микро-ЭВМ рассматриваемого семейства.

В отличие от 8051 микро-ЭВМ семейства 8052 имеют:

- встроенное ПЗУ объемом 8 (80С52), 16 (80С54) и 32 Кбайт (80С58);
- встроенное ОЗУ объемом 256 байт;
- дополнительные специальные функциональные регистры;
- таймер/счетчик 2 (далее для краткости — Т/С2), способный работать в режимах защелки, таймера/счетчика, допускающего счет как на увеличение, так и на уменьшение, и генератора скорости передачи в бодах;
- программируемый последовательный интерфейс с детектированием ошибок передачи и автоматическим распознаванием адреса;
- шесть источников прерываний;
- расширенный режим снижения потребляемой мощности; — флаг отключения питания; — режим ONCE.

Микро-ЭВМ 8052 используют стандартный набор команд семейства 8051, их выводы взаимно однозначно соответствуют выводам этих микро-ЭВМ. Отличие заключается лишь в том, что, помимо ввода\вывода информации, выводы P1.0 и P1.1 8052 могут выполнять альтернативные функции: первый из них играет роль внешнего входа для Т/С2, а второй управляет перезагрузкой/защелкиванием информации в регистры Т/С2.

### 5.0.3.8XC51FA .

В качестве одной из перспективных моделей MCS-51 можно считать микросхему 8XC51FA. В ее состав входят:

- четыре 8-битных параллельных порта;
- модуль PCA;
- последовательный порт;
- три 16-битных счетчика— таймера.

Микроконтроллеры с внутренней памятью программ позволяют защищать свои программные коды от копирования. Для этого используется схема блокировки внутренней памяти программ, которая состоит из специальных битов (Lock bits) и кодирующего массива (Encryption Array). Запрограммировав один или несколько таких битов, можно полностью или частично заблокировать эту память. При полной блокировке будет невозможно: — чтение с внешней шины внутренней памяти программ; — дальнейшее программирование кристалла; — выполнение команд и внешней памяти программ. При частичной блокировке возможно запретить или разрешить вышеперечисленные действия по отдельности. Кодирующий массив используется для поразрядного выполнения логической операции XNOR над байтами из внутренней памяти программ и байтами из этого массива при верификации, если она разрешена.

Основным отличием моделей 8XC51PA от отечественных аналогов является наличие модуля PCA (Programmable Counter Array) .

Это устройство состоит из 16-разрядного счетчика-таймера и пяти модулей сравнения-захвата. В качестве входных импульсов для счетчика-таймера могут служить сигналы:

- частота резонатора /12;
- переполнение от Timer 0;
- частота резонатора /4;
- внешний сигнал на контакте P1.2.

Каждый из пяти модулей сравнения-захвата может работать в следующих режимах :

- захват положительного или отрицательного фронта;
- программный таймер;
- скоростной вывод;
- генератор прямоугольных импульсов с заданной скважностью.

Четвертый модуль имеет также режим Watchdog Timer .

PCA рекомендуется использовать для измерения таких параметров, как ширина импульса, разность фаз, скважность и частота, а также для формирования на внешних выводах микроконтроллера прямоугольных сигналов. В принципе, для этих целей можно использовать счетчики-таймеры, которые имеются на кристалле. Однако при использовании PCA повышается точность за счет того, что счетчик-таймер, входящий в состав PCA, может изменять свое значение трижды за машинный цикл. Отметим, что обычные счетчики-таймеры могут изменять свое значение лишь один раз за машинный цикл. Кроме того, PCA требует значительно меньшего вмешательства процессора.

### 5.0.4.8XC51GB.

Большой интерес для разработчиков электронной аппаратуры могут представлять микроконтроллеры 8XC51GB. На кристалле этого устройства имеется следующее оборудование:

- шесть 8-битных параллельных портов;
- два модуля PCA ;
- три 16-битных счетчика— таймера;
- детектор падения частоты;
- два последовательных порта;
- отдельный Watchdog Timer;
- 8-канальный, 8-битный АЦП поразрядного приближения .

### 5.0.5.80C152 .

Развитие коммуникационных возможностей MCS-51 нашло отражение в микроконтроллере 80C152, где наряду с обычным последовательным портом появляется GSC (Global Serial Channel). Это устройство поддерживает стандартные протоколы SDLC и применяемый в сетях Ethernet CSMA/CD. Также возможно использование протоколов, определенных пользователем. GSC обеспечивает работу на физическом и



канальном уровнях согласно базовой модели открытых систем ISO. Для передачи информации используются NRZI и манчестерский коды. Кроме GSC микроконтроллер 80C152 имеет пять 8-битных параллельных портов для 48-выводного DIP корпуса (семь для 68-выводного PLCC), два канала ПДП и два счетчика-таймера.

### 5.0.6. Маркировка микроконтроллеров фирмы Intel.

Для маркировки микросхем фирмой INTEL применяется система обозначений из нескольких полей:

1	2	3	4
X	XX	XXXXXXXXXXXXXXXXXX	XXXXXX

Первое поле содержит однобуквенный префикс, отражающий температурный диапазон микросхемы:

- A (Automotive), автомобильное исполнение для расширенного температурного диапазона (-40/+125°C)
- M (Military), исполнение по военным стандартам (-55/+125°C)
- Q или C (Commercial), "коммерческий" температурный диапазон (0/+70°C) с (160±8)-часовой динамической термотренировкой;
- L или E (Extended), "расширенный" температурный диапазон (-40/+85°C) с (160±8)-часовой динамической термотренировкой;
- T (Extended), "расширенный" температурный диапазон (-40/+85°C) без термотренировки;
- I (Industrial), исполнение по промышленным стандартам.

Второе поле содержит одно- или двухбуквенный префикс, указывающий на вариант исполнения корпуса микросхемы (Package Type). Различных типов корпусов микросхем на сегодняшний день несколько десятков, поэтому в качестве примера приведем лишь некоторые обозначения:

- A Ceramic Pin Grid Array, (PGA);
- C Ceramic Dual In-Line Package, (CDIP);
- K Ceramic Quad Flatpack Package, (QFP);
- KD Plastic Quad Flatpack Package, Fine Pitch, Die Down, (PQFP);
- KU Plastic Quad Flatpack Package, Fine Pitch, Die Up, (PQFP);
- N Plastic Leaded Chip Carrier, (PLCC);
- P Plastic Dual In-Line Package, (PDIP);
- SM Single In-Line Leadless Memory Module, (SIMM);
- U Plastic Dual In-Line Package, Shrink Dip, (PDIPS);
- Z Zigzag In-Line Package, (ZIP).

Третье поле может содержать до 15 цифровых и буквенных символов, указывающих на тип конкретного устройства, расположенного на кристалле.

Четвертое поле может включать до шести цифровых и буквенных символов, отражающих различные особенности и варианты исполнения микросхем.

Дополнительную информацию по типам корпусов и их конструктивному исполнению можно найти в книге: *Packaging Order Number 240800*.

Применительно к описываемым микроконтроллерам семейства MCS-51, первый символ третьего поля традиционно (для фирмы Intel) равен "8". Второй символ третьего поля обычно указывает на тип встроенного ПЗУ:

- 0 масочное ПЗУ программ; кристалл без ПЗУ (для поздних версий кристаллов);
- 1 масочное ПЗУ программ (Standard ROM Code, Firmware);
- 3 масочное ПЗУ (для поздних версий кристаллов), (Customizable ROM Code);
- 7 УФРПЗУ или однократно-программируемое ПЗУ (EPROM or OTP ROM);
- 8 ЭСППЗУ (Flash - память на кристалле)

Далее может следовать буква, указывающая на технологические особенности изготовления:

- отсутствие буквы технология NMOS, питание 5В;
- C технология CHMOS, питание 5В;
- L технология CHMOS, питание 3В;

Следующими символами третьего поля для микроконтроллеров семейства MCS-51 являются номера (например, 31,32,51,54,58,152) и от одной до четырех букв, которые отражают функциональные особенности кристаллов (например, объем ПЗУ, специфику группы кристаллов, наличие системы защиты памяти программ от несанкционированного доступа, возможность использования более совершенного алгоритма программирования "Quick Pulse" и тому подобное).

В оригинальной технической документации фирмы Intel все микроконтроллеры семейства MCS-51 скомпонованы по группам ("Product Line"), каждая из которых объединяет наиболее близкие по своим функциональным возможностям и электрическим параметрам версии кристаллов. Поскольку наименования микросхем одной группы различаются незначительно, то для обозначения каждой отдельной группы

применяется обобщенная символика, образованная из маркировки конкретных микросхем, путем замены различающихся символов на "X". Таким образом, можно выделить следующие группы микроконтроллеров.

**Группа 8X5X** (8051 Product Line и 8052 Product Line):8031АН, 8051АН, 8751Н, 8051АНР, 8751Н-8, 8751ВН, 8032АН, 8052АН, 8752ВН.

**Группа 8XC51** (80C51 Product Line):80C31ВН, 80C51ВН, 87C51.

**Группа 8XC5X** (8XC52/54/58 Product Line):80C32, 80C52, 87C52, 80C54, 87C54, 80C58, 87C58.

**Группа 8XC51FX** (8XC51FA/FB/FC Product Line):80C51FA, 83C51FA, 87C51FA, 83C51FB, 87C51FB, 83C51FC, 87C51FC.

**Группа 8XL5X** (8XL52/54/58 Product Line):80L52, 87L52, 80L54, 87L54, 80L58, 87L58.

**Группа 8XL51FX** (8XL51FA/FB/FC Product Line):80L51FA, 83L51FA, 87L51FA, 83L51FB, 87L51FB, 83L51FC, 87L51FC.

**Группа 8XC51RX** (8XC51RA/RB/RC Product Line):80C51RA, 83C51RA, 87C51RA, 83C51RB, 87C51RB, 83C51RC, 87C51RC.

**Группа 8XC51GB** (8XC51GX Product Line):80C51GB, 83C51GB, 87C51GB.

**Группа 8XC152JX** (8XC152 Product Line):80C152JA, 83C152JA, 80C152JB, 80C152JC, 83C152JC, 80C152JD.

**Группа 8XC51SL** (8XC51SL Product Line):80C51SL-BG, 81C51SL-BG, 83C51SL-BG, 80C51-АН, 81C51SL-АН, 83C51SL-АН, 87C51SL-АН, 80C51SL-АL, 81C51SL-АL, 83C51SL-АL, 87C51SL-АL.

## 5.1.РСА микроконтроллера 8051.

Несмотря на бурное развитие микроэлектроники за последнее время, по-прежнему во всем мире остаются популярными микроконтроллеры семейства MCS-51 фирмы Intel. Отличительной чертой последних моделей этих микросхем является наличие у них нового устройства РСА (Programmable Counter Array). Оно обеспечивает большие "временные" возможности по сравнению с обычными счетчиками-таймерами, которые имеются на кристалле, при этом достигается большая точность с меньшими затратами ресурсов ОЭВМ. РСА рекомендуется использовать для измерения таких параметров, как ширина импульса, частота, скважность, а так же для формирования прямоугольных импульсов на внешних выводах микроконтроллера. Это устройство имеется у моделей 80C51FA, 80C51FB, 80C51PC и 80C51GB.

РСА состоит из 16-разрядного таймера-счетчика и пяти 16-разрядных модулей сравнения-захвата (compare-capture), как показано на рисунке.

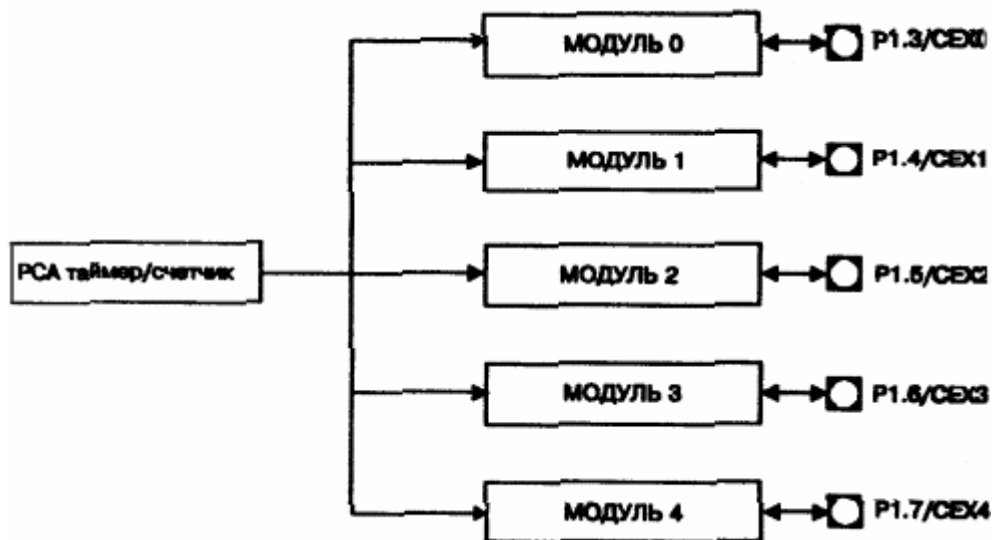


Рис.10. Таймер РСА

Таймер РСА является базой времени для пяти модулей и единственным таймером, обслуживающим РСА. На его счетный вход могут подаваться следующие сигналы:

- тактовая частота/12,
- тактовая частота/4,
- сигнал переполнения от таймера 0,
- внешний сигнал с контакта ЕС1(P1.2).

Каждый модуль сравнения-захвата может работать в одном из следующих режимов:

- захват по фронту и/или спаду,
- программируемый таймер;
- скоростной вывод,
- генератор импульсов с заданной скважностью.

Кроме этого, четвертый модуль может работать в режиме сторожевого таймера (watchdog timer). Во время работы модуля в каком-либо режиме может формироваться запрос на прерывание. При этом все пять модулей и сигнал переполнения от PCA таймер-счетчика совместно используют один вектор прерывания. Внешние контакты устройства PCA и параллельного порта 1 совмещены. Контакты, которые не задействованы при работе PCA, могут использоваться для обычного ввода-вывода.

### 5.2.16-разрядный таймер-счетчик устройства PCA микроконтроллера MCS-51.

16-разрядный таймер-счетчик PCA имеет два регистра CH и CL - старший и младший байты. Запись и чтение этих регистров возможно в любой момент времени. На рисунке представлена структурная схема PCA таймера-счетчика.

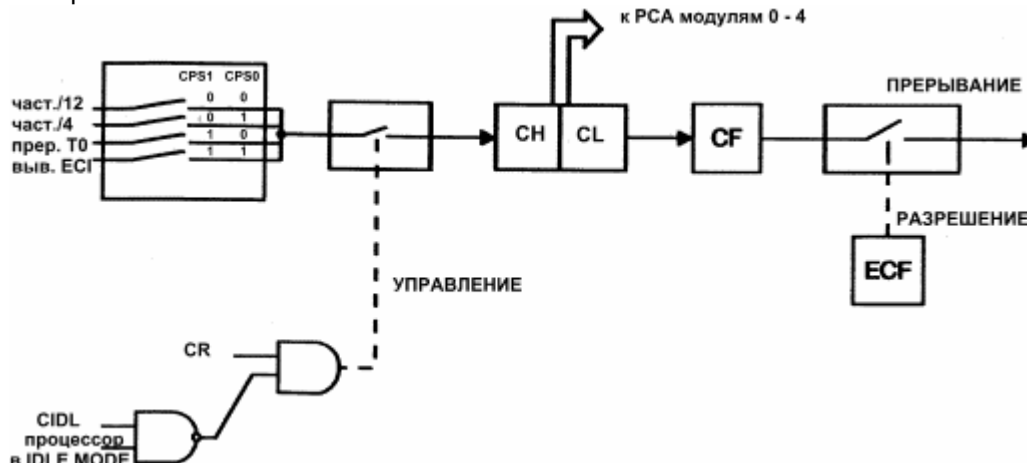


Рис.11. Структурная схема PCA таймера-счетчика.

На него могут подаваться следующие сигналы:

- **тактовая частота/12** (регистр CL за каждый машинный цикл увеличивает свое значение в момент S5P2, при внешнем резонаторе 16МГц это происходит каждые 750 нс.);
- **тактовая частота/4** (регистр CL за каждый машинный цикл увеличивает свое значение в моменты S1P2, S3P2, S5P2, при внешнем резонаторе 16МГц это происходит каждые 250 нс.);
- **переполнение от таймера 0** (регистр CL за каждый машинный цикл увеличивает свое значение в момент S5P2 при переполнении таймера 0);
- **внешний сигнал**(регистр CL после обнаружения перепада 1-0 на контакте ECI(P1.2) увеличивает свое значение в первый из моментов S1P2, S3P2, S5P2, максимальная частота входного сигнала в этом режиме равна тактовой частоте/8).

CH увеличивает свое значение после двух тактов при переполнении CL.

Для задания режимов работы PCA таймера-счетчика имеются регистр режимов PCA таймера-счетчика (CMOD) и регистр управления PCA таймером-счетчиком (CCON).

#### 5.2.1.Регистр режимов PCA таймера-счетчика CMOD.

Адрес=0D9H

Значение после сброса 00XX X000B адресация к отдельным битам невозможна

Символ	Позиция	Имя и назначение
CIDL	CMOD.7	CIDL=0 разрешает работу PCA в "режиме холостого хода" (Idle Mode), CIDL=1 запрещает работу PCA в режиме "холостого хода".
WDTE	CMOD.6	WDTE=0 запрещает режим сторожевого таймера (watchdog timer) четвертому модулю сравнения-захвата, WDTE=1 разрешает этот режим.
-	CMOD.5	-
-	CMOD.4	-
-	CMOD.3	-
Источник сигналов для таймера-счетчика PCA		
CPS1	CMOD.2	CPS1 CPS0 сигнал на счетном входе PCA таймерасчетчика
		0 0 тактовая частота/12
CPS0	CMOD.1	0 1 тактовая частота/4
		1 0 переполнение от таймера 0
		1 1 внешний сигнал на контакте EC1/P1.2
ECF	CMOD.0	ECF=1 разрешает флагу CF в CCON генерировать запрос на прерывание. ECF=0 запрещает данную функцию.

## 5.2.2.Регистр управления PCA таймером-счетчиком CCON.

Адрес=0D8H

Значение после сброса 00X0 0000B возможна адресация к отдельным битам

Символ	Позиция	Имя и назначение
CF	CCON.7	Флаг переполнения PCA таймера-счетчика, устанавливается аппаратно при переполнении счетчика. CF генерирует запрос на прерывание только при установленном бите ECP в CMOO. CP можно установить как программно, так и аппаратно, но сбрасывается он только программно.
CR	CCON.6	Бит запуска PCA таймера-счетчика. При программной установке включает PCA таймер-счетчик. При программном сбросе выключает.
-	CCON.5	-
CCF4	CCON.4	Флаг прерывания модуля 4. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.
CCF3	CCON.3	Флаг прерывания модуля 3. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.
CCF2	CCON.2	Флаг прерывания модуля 2. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.
CCF1	CCON.1	Флаг прерывания модуля 1. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.
CCFO	CCON.0	Флаг прерывания модуля 0. Устанавливается аппаратно, когда происходит соответствующее событие. Сбрасывается программно.

## 5.3.Модули сравнения-захвата PCA микроконтроллеров MCS-51.

Каждый модуль может выполнять любую из следующих функций:

- захват 16-разрядного значения по фронту,
- захват 16-разрядного значения по спаду;
- захват 16-разрядного значения и по фронту и по спаду,
- 16-разрядный программируемый таймер;
- скоростной вывод;
- генератор импульсов заданной скважности.

Четвертый модуль может также работать в режиме сторожевого таймера.

У каждого модуля имеется два 8-разрядных регистра CCAPnH и CCAPnL. В этих регистрах или фиксируется или задается момент наступления некоего события. В режиме широтно-импульсной модуляции CCAPnH определяет скважность импульсов. Режим работы модуля сравнения захвата определяет регистр CCAPMn (n=0,1,2,3,4).

### 5.3.1.Регистр режимов модуля сравнения захвата CCAPMn.

Значение после сброса X000

Адреса CCAPMn (n=0..4)

CCAPM0	0DAH
CCAPM1	0DBH
CCAPM2	0DCH
CCAPM3	0DDH
CCAPM4	0DEH

адресация к отдельным битам не возможна

Символ	Позиция	Имя и назначение
-	CCAPMn.7	-
ECOMn	CCAPMn.6	ECOMn=1 разрешение сравнения.
CAPPn	CCAPMn.5	CAPPn=1 разрешение захвата по фронту.
CAPNn	CCAPMn.4	CAPNn=1 разрешение захвата по спаду.
MATn	CCAPMn.3	Если MATn=1, то при совпадении значений в PCA таймере-счетчике и в регистрах CCAPnH, CCAPnL устанавливается флаг CCFn в регистре CCON.
TOGn	CCAPMn.2	Если TOGn=1, то при совпадении значения PCA в таймере-счетчике и в регистрах CCAPnH, CCAPnL. на внешнем контакте CEXn происходит переключение сигнала (0-1 или 1-0).
PWMn	CCAPMn.1	PWMn=1 разрешает использовать внешний контакт CEXn для генерации импульсов заданной скважности.
ECCFn	CCAPMn.0	Разрешение флагу CCFn в регистре CCON генерировать запрос на прерывание.

### 5.3.2. Режимы работы PCA.

В таблице представлены комбинации битов регистра ССАРМn, соответствующие различным режимам работы модуля сравнения-захвата.

	ЕСОМn	САРРn	САРNn	МАТn	ТОGn	РWМn	ЕССF	Функции модуля
X	0	0	0	0	0	0	0	нет операций
X	X	1	0	0	0	0	X	захват по фронту на СЕХn
X	X	0	1	0	0	0	X	захват по спаду на СЕХn
X	X	1	1	0	0	0	X	захват по перепаду на СЕХn
X	1	0	0	1	0	0	X	программируемый таймер
X	1	0	0	1	1	0	X	скоростной вывод
X	1	0	0	0	0	1	0	модулирование длительности импульса
X	1	0	0	1	X	0	X	сторожевой таймер

### 5.4. Режимы работы PCA микроконтроллеров семейства MCS-51.

#### 5.4.1. Режим захвата.

При обнаружении фронта или спада на внешнем выводе СЕХn в этом режиме осуществляется запись значения PCA таймера-счетчика в регистры ССАРnH, ССАРnL (см. рисунок). Этот режим рекомендуется использовать при измерении периодов, длительности, скважности импульсов, а также разности фаз между различными входами. Установка битов САРРn и/или САРNn в регистре ССАРМn определяет перепад (0-1 или 1-0), по которому будет осуществляться захват. При захвате устанавливается флаг события модуля ССFn в регистре ССОН, и генерируется запрос на прерывание, если установлен бит ЕССFn в регистре ССАРМn. Флаг события сбрасывается программно. До наступления следующего события значение в регистрах ССАРnH, ССАРnL должно быть сохранено в ОЗУ подпрограммой обработки прерывания. В противном случае, в регистрах ССАРnH, ССАРnL будет записано новое значение.

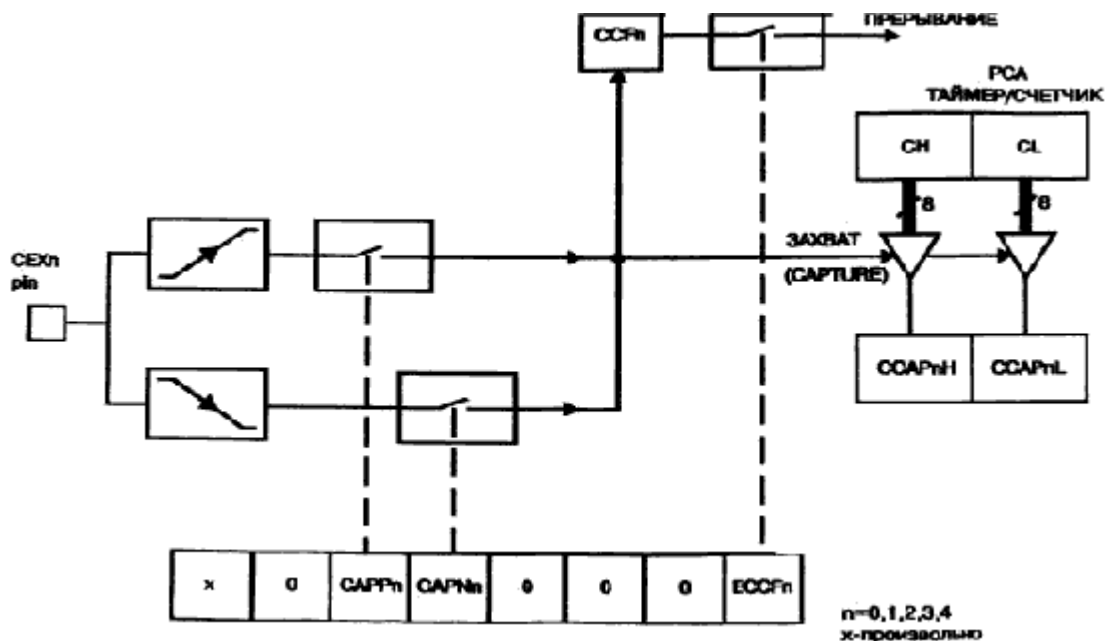


Рис.12. Режим захвата.

#### 5.4.2. Режим 16-разрядного программируемого таймера.

В этом режиме 16-разрядное значение PCA таймера-счетчика сравнивается с 16-разрядным значением, которое было предварительно загружено в регистры ССАРnH, ССАРnL одного из модулей сравнения-захвата. Это происходит трижды за машинный цикл. Установка бита ЕСОМn в регистре ССАРМn разрешает функцию сравнения, как показано на рисунке. Для перехода в режим программируемого таймера также должен быть

установлен бит МАТn. Когда происходит совпадение значений PCA таймерасчетчика и регистров ССАРnH, ССАРnL, устанавливается флаг событий ССFn. При этом будет генерироваться запрос на прерывание, если бит ЕССFn установлен. Математическое обеспечение должно сбрасывать флаг событий ССFn перед тем, как произойдет следующее совпадение.

Подпрограмма обработки прерывания может записать новое 16-разрядное значение в регистры ССАРnH, ССАРnL. Однако следует помнить, что запись в ССАРnL сбрасывает ЕСОМn бит, тем самым

запрещая функцию сравнения на время пока происходит изменение содержимого этих регистров, что позволяет избежать неверного совпадения. Запись ССАРnH устанавливает бит ECOMn и возобновляет работу компаратора. По этой причине математическое обеспечение должно записывать сначала ССАРnL, затем ССАРnH.

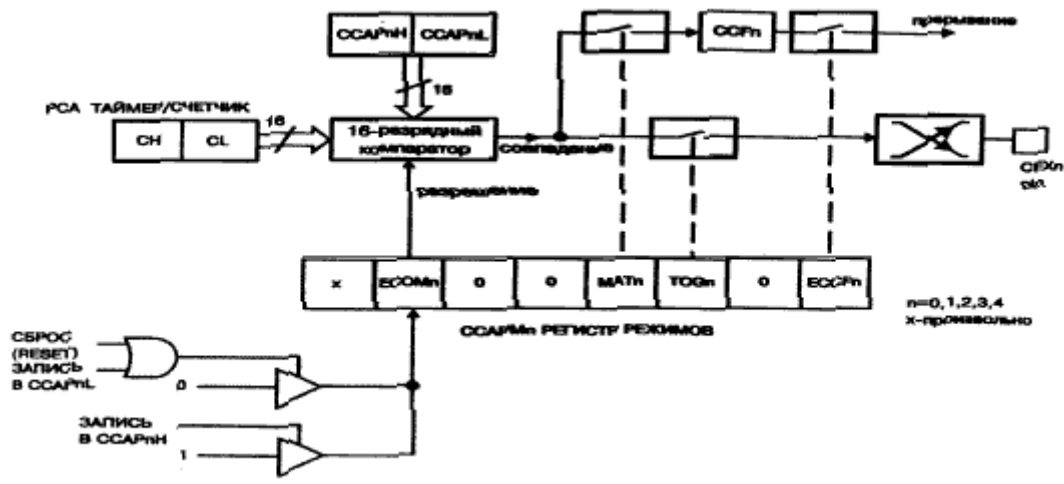


Рис.13. Режим 16-разрядного программируемого таймера.

### 5.4.3.Режим скоростного вывода.

В этом режиме формируется сигнал на внешнем выводе СЕХn, когда происходит совпадение PCA таймера со значением, которое было предварительно загружено в регистры ССАРnH, ССАРnL одного из модулей. В этом режиме бит TOGn должен быть установлен дополнительно к битам ECOMn и MATn, как показано на рисунке в предыдущем разделе. Устанавливая или сбрасывая этот бит в своей программе, вы можете формировать на внешнем выводе СЕХn или фронт, или спад. Кроме того, при совпадении может генерироваться запрос на прерывание, если бит ECCFn установлен. Режим скоростного вывода является более точным по сравнению с переключением контактов параллельного порта программным путем, поскольку формирование сигнала на внешнем выводе происходит до обращения к подпрограмме обработки прерывания. Это значит, что задержка, связанная с выполнением подпрограммы обработки прерывания, не будет влиять на формирование сигнала. Если подпрограмма обработки прерывания не изменит содержимого регистров ССАРnH, ССАРnL, то следующий сигнал "совпадение" будет сформирован после того, как значение PCA таймера-счетчика снова совпадет со старым сравниваемым значением.

### 5.4.4.Режим сторожевого таймера (watchdog timer).

Сторожевой таймер - это схема, которая автоматически сбрасывает микроконтроллер, если не получает от управляемой системы сигнала, который подтверждает, что не произошло никакого сбоя. Такое устройство используется в системах, (где есть электрические помехи или сбои по питанию и, где нужно обеспечить большую надежность. В режиме сторожевого таймера может работать только четвертый модуль. Сигнал сброса срабатывает всякий раз, когда происходит совпадение значения PCA таймера-счетчика со значением регистров ССАР4H, ССАР4L (см. рисунок). Бит WDTE в регистре СMOD устанавливает этот режим. При этом четвертый модуль должен находиться или в режиме программируемого таймера, или скоростного вывода. Сторожевой таймер сбрасывает ОЭВМ также, как при подаче высокого уровня на девятую "ножку" микросхемы. Предотвратить сброс можно тремя способами:

1. периодически изменять значение в регистрах ССАР4H, ССАР4L, так что бы оно никогда не совпало с PCA таймером-счетчиком;
2. периодически изменять регистры CH, CL PCA таймера-счетчика, чтобы они никогда не совпали со значением в регистрах ССАР4H, ССАР4L;
3. выключить режим охранного таймера путем сброса бита WDTE перед тем, как произойдет совпадение и затем снова включить его.

Первые два способа являются более надежными, поскольку третий способ трудно реализовать. Второй способ не рекомендуется применять, если таймер PCA используется другим модулем. Если сторожевой таймер не нужен, то четвертый модуль может работать в любом другом режиме.

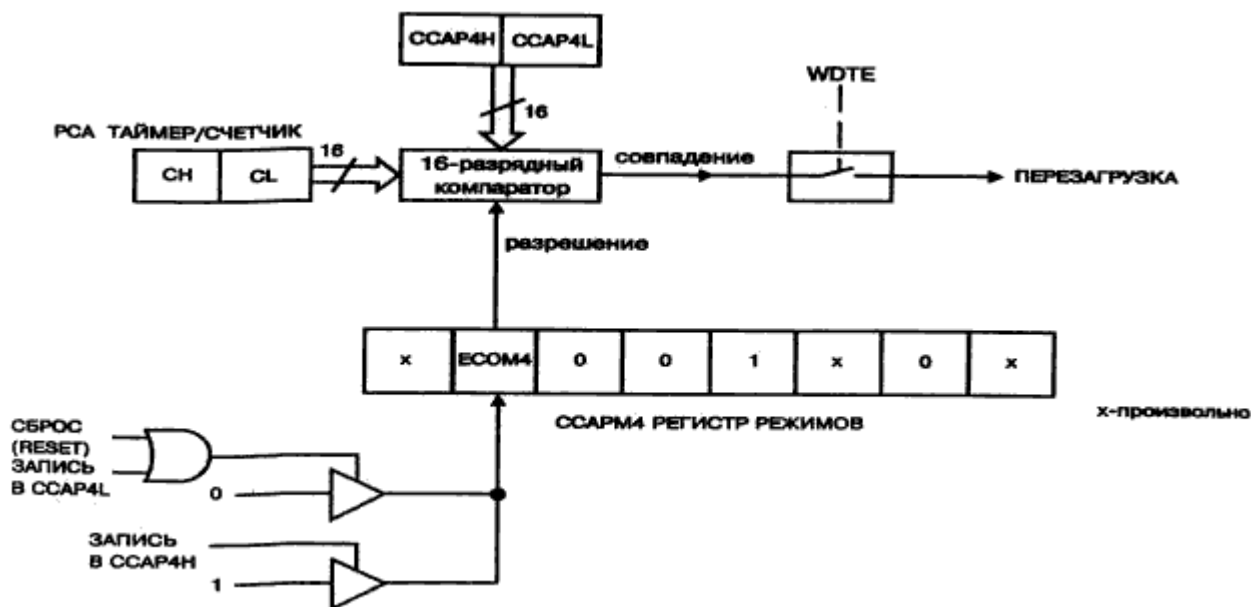


Рис.14. Сторожевой таймер.

#### 5.4.5. Режим генерации импульсов заданной скважности.

Любой из пяти модулей может быть использован как генератор импульсов заданной скважности. Частота генерируемых импульсов непосредственно зависит от частоты сигналов на счетном входе PCA таймера-счетчика. При внешнем резонаторе 16МГц, максимальная возможная частота генерируемых импульсов будет 15.6КГц. В этом режиме происходит сравнение регистра CL (младший байт PCA таймера-счетчика) с регистром ССАРnL (см. рисунок).

Когда  $CL < ССАРnL$  на внешнем контакте будет сигнал низкого уровня, при  $CL \geq ССАРnL$  на выходе будет сигнал высокого уровня. Значение в ССАРnL задает скважность импульсов. Для того, чтобы во время изменения значения ССАРnL на выходе не возникло помех, нужно новое значение записывать в регистр ССАРnH. Затем это значение аппаратно загрузится в ССАРnL при переходе CL из значения 0FFH в 00H, что будет соответствовать началу следующего периода. Изменяя значение в ССАРnH от 0 до 255 можно задавать скважность от 100% до 0.4%.

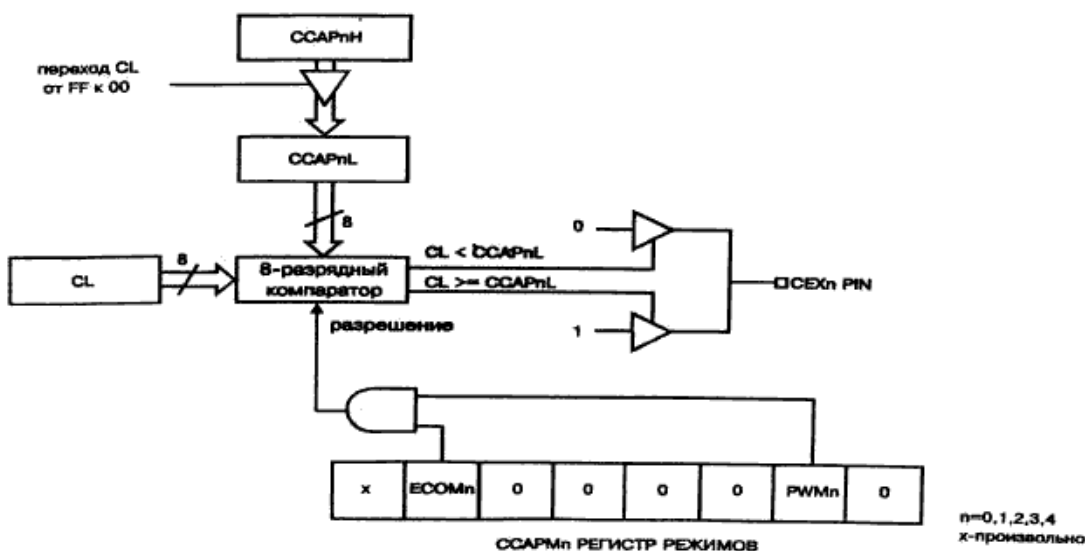


Рис.15. Режим генерации импульсов заданной скважности.

### 5.5. Аналого-цифровой преобразователь микроконтроллеров семейства MCS-51.

Аналого-цифровой преобразователь микроконтроллера семейства MCS-51/52 (например, типа SAB 80515 фирмы Siemens или 80C51GB) обеспечивает 8 битное преобразование и имеет восемь мультиплексных каналов аналогового входного сигнала "на чипе". Кроме того, аналого-цифровой преобразователь имеет схему выборки-хранения и возможность программирования опорных напряжений, что позволяет увеличивать точность преобразования, сужая пределы измерения. Преобразование осуществляется методом последовательного приближения с использованием конденсаторной цепи. Длительность цикла преобразования от 15 до 29 машинных циклов.

В аналого-цифровом преобразователе имеются три доступных для пользователей специальных функциональных регистра:

- ADCON- регистр управления аналого-цифрового преобразователя,
- ADDAT- регистр данных аналого-цифрового преобразователя, и
- DAPR- регистр программирования опорных напряжений.

### 5.5.1.ADCON - Регистр управления преобразователем.

(адрес - 0D8H, возможна побитовая адресация)

BD	CLK	-	BSY	ADM	MX2	MX1	MX0	Бит
0DFH	0DEH	0D0H	0DCH	0DBH	0DAH	0D9H	0D8H	Адрес

Регистр ADCON используется, чтобы

- выбрать один из восьми каналов аналогового входного сигнала, которые будут преобразованы,
- определять однократное или циклическое преобразование, и
- проверять бит состояния BSY, который сообщает, происходит ли преобразование или нет.

Символ	Позиция	Функция																																													
MX0	ADCON.0	Выбор канала аналогового входного сигнала, см. таблицу.																																													
MX1	ADCON.1																																														
MX2	ADCON.2	<table border="1"> <thead> <tr> <th>MX2</th> <th>MX1</th> <th>MX0</th> <th>Выбранный Канал</th> <th>Вывод ИМС</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Аналоговый входной сигнал 0</td> <td>AN0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Аналоговый входной сигнал 1</td> <td>AN1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Аналоговый входной сигнал 2</td> <td>AN2</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Аналоговый входной сигнал 3</td> <td>AN3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Аналоговый входной сигнал 4</td> <td>AN4</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Аналоговый входной сигнал 5</td> <td>AN5</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Аналоговый входной сигнал 6</td> <td>AN6</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Аналоговый входной сигнал 7</td> <td>AN7</td> </tr> </tbody> </table>	MX2	MX1	MX0	Выбранный Канал	Вывод ИМС	0	0	0	Аналоговый входной сигнал 0	AN0	0	0	1	Аналоговый входной сигнал 1	AN1	0	1	0	Аналоговый входной сигнал 2	AN2	0	1	1	Аналоговый входной сигнал 3	AN3	1	0	0	Аналоговый входной сигнал 4	AN4	1	0	1	Аналоговый входной сигнал 5	AN5	1	1	0	Аналоговый входной сигнал 6	AN6	1	1	1	Аналоговый входной сигнал 7	AN7
		MX2	MX1	MX0	Выбранный Канал	Вывод ИМС																																									
		0	0	0	Аналоговый входной сигнал 0	AN0																																									
		0	0	1	Аналоговый входной сигнал 1	AN1																																									
		0	1	0	Аналоговый входной сигнал 2	AN2																																									
		0	1	1	Аналоговый входной сигнал 3	AN3																																									
		1	0	0	Аналоговый входной сигнал 4	AN4																																									
		1	0	1	Аналоговый входной сигнал 5	AN5																																									
1	1	0	Аналоговый входной сигнал 6	AN6																																											
1	1	1	Аналоговый входной сигнал 7	AN7																																											
ADM	ADCON.3	Режим аналого-цифрового преобразования. Если ADM = 1 - непрерывное преобразование. Если ADM = 0, преобразователь останавливается после одного преобразования.																																													
BSY	ADCON.4	Флаг занятости. Этот флаг указывает, происходит преобразование (BSY = 1) или нет (BSY = 0).																																													
-	ADCON.5	Зарезервирован (должен быть 0).																																													
CLK	ADCON.6	(Используется не для управления АЦП) Включение синхронизации системы. Если равно 1, синхросигнал с $t/12$ частотой генератора подается на вывод P1.6/CLKOUT. CLK = 0 блокирует синхронизирующий вывод.																																													
BD	ADCON.7	(Используется не для управления АЦП) Включение режима передачи со скоростью в бодах. Если равно 1, то осуществляется прием в режиме 1 и 3 последовательного порта из внутреннего генератора скоростей в бодах.																																													

### 5.5.2.ADDAT - регистр результатов преобразования.

Специальный функциональный регистр ADDAT, фиксирует результат преобразования (8 бит). Данные сохраняются в ADDAT, пока результат не замещается данными следующего преобразования. Новое значение появляется в ADDAT на 15-ом машинном цикле после того, как преобразование было начато. ADDAT может читаться и записываться программным образом. Если АЦП не используется, регистр ADDAT может использоваться и как дополнительный универсальный регистр.

### 5.5.3.DAPR - регистр программирования опорных напряжений АЦП.

Регистр DAPR позволяет менять внутренние опорные напряжения IVAREF и IVAGND. Они могут программироваться с шагом в 1/16 относительно внешних опорных напряжений (VAREF-VAGND). Биты с 0 по 3 регистра DAPR определяют IVAGND, биты с 4 по 7 определяют IVAREF. Для нормальной работы АЦП требуется как минимум разность в 1 вольт между внутренними опорными напряжениями. Поэтому, внутреннее опорное напряжение IVAREF должны всегда программироваться на четыре шага выше, чем IVAGND (относительно внешнего образцового напряжения VAREF, которое равно  $5B \pm 5\%$ ).

Значения IVAGND и IVAREF определяются по формуле:

$$IVAGND = VAGND + DAPR(0 - 3) / 16 (VAREF - VAGND)$$

причем  $DAPR(0 - 3) < 0$  и  $DAPR(0 - 3) < 13$ ;

$$IVAREF = VAGND + DAPR(4 - 7) / 16 (VAREF - VAGND)$$

причем  $DAPR(4 - 7) > 3$ ;



где DAPR (0 - 3) - содержание младшего полубайта, и DAPR (4 - 7) - содержание старшего полубайта DAPR, принимаемого как десятичное целое число без знака.

Если DAPR (0 - 3) или DAPR (4 - 7) = 0, внутренние опорные напряжения соответствуют внешним опорным напряжениям соответственно VAGND и VAREF .

Если VAINPUT > IVAREF, то результат преобразования будет равен 0FFH, если VAINPUT < IVAGND, то результат преобразования - 00H (VAINPUT - напряжение аналогового входного сигнала).

Рисунок показывает, назначение разрядов регистра специальной функции DAPR.

*Регистр программирования АЦП DAPR (0DAH)*

7	6	5	4	3	2	1	0	Бит
Цифровое Значение для IVAREF				Цифровое Значение для IVAGND				

Если используются внешние опорные напряжения VAGND = 0 V и VAREF = + 5V (относительно GND и VCC), то при помощи регистра DAPR могут быть установлены следующие значения внутренних опорных напряжений IVAGND и IVAREF, как показано в таблице.

Степень	DAPR (0-3)	DAPR (4-7)	IVAGND (V)	IVAREF (V)
0	0000	0000	0.0	5.0
1	0001	0001	0.3125	-
2	0010	0010	0.625	-
3	0011	0011	0.9375	-
4	0100	0100	1.25	1.25
5	0101	0101	1.5625	1.5625
6	0110	0110	1.875	1.875
7	0111	0111	2.1875	2.1875
8	1000	1000	2.5	2.5
9	1001	1001	2.8125	2.8125
10	1010	1010	3.125	3.125
11	1011	1011	3.4375	3.4375
12	1100	1100	3.75	3.75
13	1101	1101	-	4.0625
14	1110	1110	-	4.375
15	1111	1111	-	4.6875

**Таблица.12. Программирование внутренних опорных напряжений**

Комбинации отмеченные '-' не допускаются поскольку IVAREF должен быть, по крайней мере, четыре степени выше, чем IVAGND.

#### 5.5.4. Синхронизация АЦП и время преобразования.

Преобразование начинается после записи стартового бита в регистр DAPR. Эта операция начнет новое преобразование, даже если текущее преобразование не завершено. Преобразование начинается со следующего машинного цикла. Флаг занятости будет устанавливаться в том же самом машинном цикле. Если значение, записанное в DAPR - 00H, это означает, что никакая корректировка внутренних опорных напряжений не желательна, и преобразование будет происходить в течение 15 машинных циклов, до полного окончания. Таким образом, время преобразования - 15 мкс при 12 МГц частоте тактового генератора. Для каждой корректировки внутренних опорных напряжений преобразование требуется дополнительно время 7 мкс. Таким образом, если должно программироваться только одно опорное напряжение, общее время преобразования будет занимать 22 машинных цикла, если же должны программироваться оба опорных напряжения, то время преобразования будет продолжаться 29 машинных циклов.

После того, как преобразование было запущено записью в соответствующий бит DAPR, аналоговое напряжение в выбранном входном канале выбирается в течение 5 машинных циклов (5 мкс при 12 МГц частоте генератора). Это напряжение будет оставаться неизменным на протяжении остальной части времени преобразования. Внешний аналоговый источник должен обеспечить ток достаточный, чтобы зарядить емкость выборки-хранения, равную 25pF, за 5 машинных циклов.

Преобразование выбираемого аналогового напряжения происходит между 6-ым и 15-ым машинным циклом после того, как была завершена операция выборки сигнала. В 15-ом машинном цикле преобразованный результат перемещается в ADDAT, флаг занятости (BSY) очищается, генерируется запрос на прерывание от АЦП и устанавливается флаг IADC (бит 0 в регистре управлении прерывания IRCON). Если установлено непрерывное преобразование, то следующее преобразование, автоматически начнется в следующем машинном цикле.

## 5.6. Таймер счетчик T/C2 микроконтроллера 8052.

T/C2 — это 16-битный таймер/счетчик, способный работать и как таймер, и как счетчик событий. Выбор производится битом CD2 в SFR T2CON . T/C2 может работать в режимах защелки, автоперезагрузки (при этом направление счета может быть как вверх, так и вниз, т. е. на увеличение или уменьшение содержимого TL2, TH2) и генератора скорости передачи в бодах. Режимы выбираются битами в T2CON .

T/C2 состоит из двух 8-битных регистров: TH2 и TL2. В режиме таймера его 16-разрядный регистр TL2, TH2 инкрементируется в каждом машинном цикле. Поскольку цикл состоит из 12 периодов колебаний, скорость счета равна 1/12 тактовой частоты (т. е. частоты кварцевого резонатора  $f_{рез}$ . По сути дела, в этом режиме таймер подсчитывает выполненные машинные циклы.

### 5.6.1. Регистр управление таймера/счетчика 2 T2COM.

Символ	Позиция	Имя и назначение
TF2	T2COM.7	Флаг переполнения таймера/счетчика 2. Сбрасывается программным путем. Флаг не устанавливается, если либо RCLK, либо TCLK (см. ниже) установлены в 1
EXF2	T2CON.6	Внешний флаг таймера/счетчика 2. Устанавливается при защелкивании информации в таймерных регистрах или при перезагрузке, происходящих под воздействием перепада из 1 в 0 на выводе P1.1 и при EXEN2 = 1. Сбрасывается только программным путем. При разрешенном прерывании от таймера/счетчика 2 вызывает подпрограмму обработки прерывания (если бит T2MOD.0 = 0)
RCLK	T2CON.5	Выбор таймера/счетчика 2 для задания скорости работы приемника в режимах 1 и 3. При RCLK = 1 используется таймер/счетчик 2, при RCLK = 0 - таймер/счетчик 1
TCLK	T2CON.4	Выбор таймера/счетчика 2 для задания скорости работы передатчика в режимах 1 и 2. При TCLK = 1 используется таймер/счетчик 2, при TCLK = 0 - таймер/счетчик 1
EXEN2	T2CON.3	Разрешение работы от внешнего сигнала. При EXEN2 = 1 перепад из 1 в 0 на выводе P1.1 вызывает защелкивание информации или перезагрузку таймера/счетчика 2. При EXEN2 = 0 таймер/счетчик 2 игнорирует сигналы на выводе P1.1
TR2	T2CON.2	Запуск/остановка таймера. TR2 = 1 запускает таймер/счетчик 2
C/T2	T2CON.1	Бит выбора режима работы таймера/ счетчика 2. При установке бита в 1 функционирует как счетчик перепадов из 1 в 0 на выводе P1.0. При установке бита в 0 работает таймер
CP/RL2	T2CON.0	Выбор режима защелки/перезагрузки. Если бит установлен в 1, то при EXEN2 = 1 перепад из 1 в 0 на P1.1 вызовет защелкивание содержимого TL2. TH2 а BCAP2L, RCAP2H. При нулевом бите таймер/счетчик 2 работает в режиме автоперезагрузки при переполнении счетчика или при обнаружении перепада из 1 в 0 на P1.1 (последнее при EXEN2 = 1)

После сброса все биты регистра T2ON устанавливаются в 0.

### 5.6.2. Режимы работы таймера/счетчика 2.

RCLK и TCLK	CP/RL2	TR2	Режим
0 и 0	и	1	16-битный таймер/счетчик с перезагрузкой
0 и 0	1	1	16-битный таймер/счетчик с защелкиванием информации
0 и 1, 1 и 0. 1 и 1	Любое	1	Генератор приемопередатчика
Любое	Любое	0	Выключен

**В режиме счетчика** регистр инкрементируется в ответ на перепад из 1 в 0 на входе P1.0. Состояние этого входа анализируется в момент S5P2 каждого машинного цикла. Если анализ показывает наличие единичного уровня в одном цикле и нулевого в следующем, содержимое счетчика инкрементируется. Его новое значение появляется в регистре в момент S3P1 цикла, следующего за тем, в котором был обнаружен перепад. Так как обнаружение этого перепада занимает два машинных цикла (24 периода колебаний), максимальная скорость счета равна 1/24 частоты  $f_{рез}$ . Чтобы микро-ЭВМ успела идентифицировать заданный уровень, он должен T/C2 — это 16-битный таймер или счетчик, при переполнении которого устанавливается бит TF2 в T2CON. Этот бит затем может использоваться для вызова прерывания. Если же EXEN2 = 1, то T/C2 продолжает делать то же самое, однако в этом случае перепад из 1 в 0 на выводе P1.1 вызывает защелкивание текущих значений TH2 и TL2 в RCAP2H и RCAP2L соответственно. Кроме того, этот перепад вызывает установку в 1 бита EXF2 в T2CON. Как и TF2, этот бит может вызвать прерывание. Режим защелки иллюстрирует рис. 1, содержимое счетчика растет вверх до 0FFFFH и затем устанавливает в 1 бит переполнения TF2. Переполнение вызывает также перезагрузку регистров T/C2 16-битным значением в RCAP2H и RCAP2L, которое предварительно должно быть установлено программно. Если же EXEN2 = 1 16-битная; перезагрузка может произойти как от переполнения, так и от перепада из 1 в 0 на выводе P1.1. Этот перепад также устанавливает в 1 бит EXP2. Оба бита - TP2 и EXP2 — могут вызвать прерывание, если оно разрешено.

Установка в 1 бита DCEN переводит T/C2 в режим счета вверх или вниз, как показано на рис.

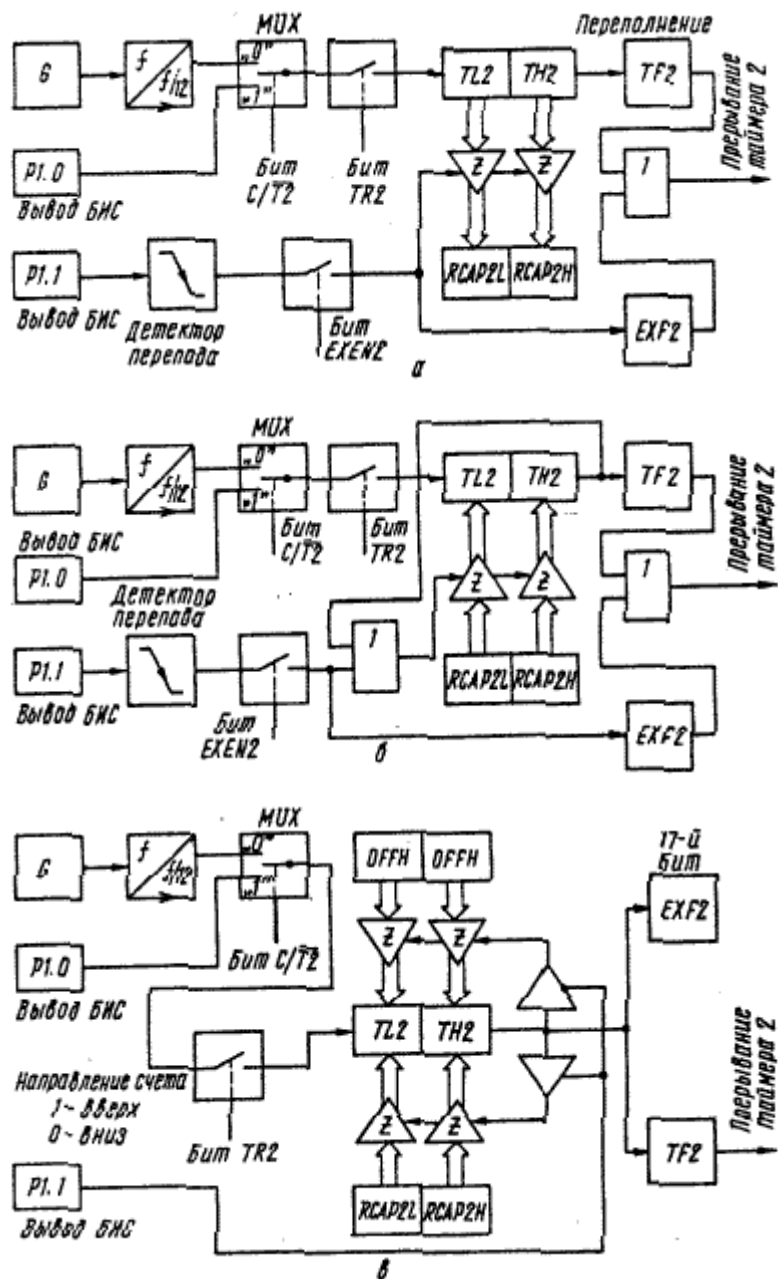


Рис.16. Работа Таймера/счетчика 2 в режиме : захвата (а), автоперезагрузки (б), задающего генератора (в) .

**Режим автоперезагрузки.** На рис. 1, б показана структурная схема Т/С2, считающего в сторону увеличения при DCEN = 0. В этом режиме возможны два подрежима, выбираемых регистром T2MOD

### 5.6.3. Регистр режима таймера/счетчика 2 T2MOD.

Символ	Позиция	Имя и назначение
	T2MOD.7	Не используется
	T2MOD.6	Не используется
	T2MOD.5	Не используется
	T2MOD.4	Не используется
	T2MOD.3	Не используется
	T2MOD.2	Не используется
T20E	T2MOD.1	При установке бита на выводе P1.1 формируется последовательность прямоугольных импульсов со скважностью 2
DCEN	T2MOD.0	При установке бита таймер/счетчик 2 конфигурируется на счет как вверх, так и вниз, в зависимости от уровня сигнала на выводе P1.1

Биты регистра T2MOD не адресуются непосредственно командами работы с битами. После сброса T2MOD.0 и T2MOD.1 устанавливаются в 0, значение остальных бит не определено.

#### 5.6.4. Дополнительный регистр приоритетов прерываний IPH.

Символ	Позиция	Имя и назначение
	IPH.7	Зарезервирован
	IPH.6	Зарезервирован
PT2H	IPH.5	Старший бит приоритета таймера/счетчика 2
PSH	IPH.4	Старший бит приоритета приемопередатчика
PT1H	IPH.3	Старший бит приоритета таймера/счетчика 1
PX1H	IPH.2	Старший бит приоритета внешнего прерывания 1
PTOH	IPH.1	Старший бит приоритета таймера/счетчика 0
PXOH	IPH.0	Старший бит приоритета внешнего прерывания 0

После сброса младшие шесть бит устанавливаются в 0, состояние старших - не определено.

## 6. Семейство MCS-251

Изначально наиболее "узкими" местами архитектуры MCS-51 были восьмиразрядное АЛУ на базе аккумулятора и относительно медленное выполнение инструкций (для самых "быстрых" из них требуется 12 периодов тактовой частоты). Это ограничивало применение МК семейства в устройствах, требующих повышенного быстродействия и сложных вычислений (16- и 32-битных). Насущным стал вопрос принципиальной модернизации старой архитектуры. Проблема осложнялась тем, что к началу 90-х годов уже была создана масса наработок в области программного и аппаратного обеспечения, и одной из основных задач разработки новой архитектуры стала реализация аппаратной и программной совместимости со старыми разработками на базе MCS-51. Для решения этой задачи была создана совместная группа из специалистов компаний Intel и Philips, но позднее пути этих двух фирм разошлись. В результате в 1995 г. появилось два существенно различающихся семейства: MCS-251/151 у Intel и 51XA у Philips .

Основные характеристики архитектуры MSC-251:

24-разрядное линейное адресное пространство, обеспечивающее адресацию до 16 Мбайт памяти; регистровая архитектура, допускающая обращение к регистрам как к байтам, словам и двойным словам;

страничный режим адресации для ускорения выборки инструкций из внешней программной памяти; очередь инструкций;

расширенный набор команд, включающий 16-битные арифметические и логические инструкции;

расширенное до 64 Кбайт адресное пространство стека;

выполнение самой "быстрой" инструкции за два такта;

совместимость на уровне двоичного кода с программами для MCS-51.

Система команд MCS-251 построена на базе двух наборов инструкций: первый является копией системы команд MCS-51, а второй состоит из расширенных инструкций, реализующих преимущества архитектуры MSC-251. Перед использованием МК его необходимо сконфигурировать, т. е. с помощью программатора "прожечь" конфигурационные байты, определяющие, какой из наборов инструкций станет активным после включения питания. Если установить набор инструкций MCS-51, то MSC-251 будет совместим с MCS-51 на уровне двоичного кода (режим Binary Mode). Расширенные инструкции в этом режиме также доступны, но через "форточку" - зарезервированный код инструкции 0A5H. Естественно, длина каждой расширенной инструкции увеличивается в таком случае на 1 байт.

Если же изначально установить набор расширенных инструкций, то программы, написанные для MCS-51, потребуют перекомпиляции на кросс-средствах для MCS-51, так как теперь уже стандартные инструкции будут доступны через ту же "форточку" 0A5H и длина их также увеличится на 1 байт. Такой режим называется Source Mode. Он позволяет с максимальной эффективностью использовать расширенные инструкции и достигнуть наибольшего быстродействия, но требует переработки программного обеспечения.

Для пользователей, ориентированных на применение MCS-251 в качестве механической замены MCS-51, фирма Intel выпускает МК MCS-251 с уже запрограммированными битами конфигурации в состоянии Binary Mode. Такие МК получили обозначение MCS-151.

Помимо Intel, МК MCS-251 по ее лицензии выпускает компания Temic Semiconductors. Подробную информацию о ее продукции можно получить на [web-site фирмы](#). Основные технические характеристики МК семейства MCS-251 приведены в таблице

В настоящее время Intel, устремленная на рынок Pentium-процессоров, прекратила производство кристаллов MCS-51. В целом для конкретного разработчика этот факт может остаться и не замеченным, если только он не использует 8xC51GB и 80C152Jx - эти кристаллы не имеют своих точных аналогов среди изделий других фирм. Что же касается всех остальных МК семейства MCS-51, то они многократно растиражированы другими компаниями и уход Intel с рынка никак на нем не скажется.

МК	ROM/ EPROM, Кбайт	RAM, байт	Таймеры/ счетчики	Последова- тельные каналы	Корпус (тип, число выводов)
8xC251SA	8	1024	3 + PCA + WDT	UART	D40, L44
8xC251SB	16	1024	3 + PCA + WDT	UART	D40, L44
8xC251SP	8	512	3 + PCA + WDT	UART	D40, L44
8xC251SQ	16	512	3 + PCA + WDT	UART	D40, L44
TSC8xC251G1	16	1024	3 + WDT	UART, I <sup>2</sup> C, SPI	L40, Q44
TSC8xC251A1	24	1024	2 + WDT	UART	D40, L44, Q44
8xC151SA	8	256	3 + PCA + WDT	UART	D40, L44
8xC151SB	16	256	3 + PCA + WDT	UART	D40, L44

Таблица.13. МК семейства MCS-251 фирмы Intel

**Примечание.** Максимальная тактовая частота всех модификаций – 16 МГц, число линий ввода/вывода – 32. МК TSC8xC251A1 имеет четырехканальный восьмиразрядный АЦП. Напряжение питания всех МК – 4,5...5,5 В, рабочий интервал температур – от –40 до +85 °С.

**Принятые сокращения:** PCA – массив программируемых счетчиков; WDT – сторожевой таймер; UART универсальный асинхронный последовательный приемопередатчик; I<sup>2</sup>C – двухпроводная двунаправленная шина; SPI – последовательный периферийный интерфейс. Корпус: D – DIP, L – PLCC, Q – QFP.

## 7.Однокристальные микроконтроллеры Intel MCS-96.

### 7.1.Общая характеристика.

В семейство MCS-96 фирмы Intel (иногда будет использоваться и название 80C196) входит более 30 разновидностей микроконтроллеров. Это 16-разрядные, быстродействующие ИС высокой степени интеграции, ориентированные на решение задач управления процессами в реальном масштабе времени. Типичные области применения для этих микроконтроллеров - управление двигателями, модемы, безъюзовые тормозные системы, контроллеры жестких дисков, медицинское оборудование.

История MCS-96 насчитывает более 12 лет. За это время специалисты фирмы Intel увеличили адресное пространство с 64 КБайт до 6 Мбайт, повысили тактовую частоту с 10 до 50 МГц, улучшили быстродействие в 16 раз и добились понижения цены на базовый кристалл\* примерно в 4 раза.

По сравнению с восьмиразрядными однокристальными микроконтроллерами, данное микроконтроллерное семейство позволяет существенно расширить область применения встраиваемых микроконтроллеров в первую очередь за счет более высокой скорости и точности вычислений, а также за счет использования расположенных на кристалле новых периферийных устройств, обеспечивающих более высокую скорость обработки сигналов в управляющей системе и более высокую надежность функционирования системы.

Микроконтроллеры 80C196 фактически стали промышленным стандартом для 16-разрядных встроенных систем управления, обеспечивая сочетание высоких технических показателей и экономической эффективности. Например, именно благодаря этим микроконтроллерам, установленным в системе управления зажиганием, специалистам концерна Ford удалось существенно снизить потребление топлива, уменьшить выбросы вредных веществ и одновременно повысить скоростные характеристики своих машин.

#### 7.1.1.Структура микроконтроллера.

Микроконтроллеры семейства MCS-96 являются микропроцессорными устройствами синхронного типа. Выполнение всех действий в микроконтроллере привязано во времени к тактовым сигналам, вырабатываемым внутренним генератором тактовых импульсов. Частота следования тактовых импульсов стабилизируется с помощью внешнего кварцевого резонатора. Высшее значение тактовой частоты ( $F_{max}$ ) у микроконтроллеров разных типов может иметь значение 10, 12, 16 и 20 МГц.

Основными функциональными частями микроконтроллера являются процессор, память и периферия (набор периферийных устройств).

В состав процессора входят арифметико-логическое устройство (АЛУ, ALU) и регистровое оперативное запоминающее устройство (РОЗУ, RRAM).

**АЛУ** В отличие от микроконтроллеров других семейств АЛУ микроконтроллера семейства MCS-96 не имеет регистра-аккумулятора. В качестве регистра-аккумулятора может использоваться любой регистр РОЗУ.

На частоте 16 МГц АЛУ выполняет 2 млн. оп/с при выполнении элементарных операций над знаковыми/беззнаковыми данными длиной 1 или 2 байт. Для этих чисел имеются также и операции умножения и деления (быстродействие: 580 тыс. умножений/сек, 330 тыс. делений/сек).

**РОЗУ** у микроконтроллеров разных типов может содержать 232, 360, 488 или 1000 восьмиразрядных регистров. Регистры РОЗУ используются для хранения только данных.

**Память** представлена постоянным запоминающим устройством (ПЗУ, ROM). У микроконтроллеров некоторых типов в состав памяти входит оперативное запоминающее устройство (ОЗУ, RAM). Ячейки памяти в ОЗУ и ПЗУ могут использоваться для хранения данных и команд программы.

**ПЗУ** у контроллеров разных типов может содержать 8К, 12К, 16К, 24К или 32К восьмиразрядных ячеек памяти. В ПЗУ имеется область, предназначенная для хранения специальных данных (векторы прерывания, ключ защиты ПЗУ и другие специальные коды).

**ОЗУ** у контроллеров разных типов может иметь 128, 256 или 512 восьмиразрядных ячеек памяти. При использовании ОЗУ для размещения команд программы открывается возможность выполнять модификацию команд в процессе выполнения программы.

Процессор обращается к памяти через контроллер памяти (КП, MC). Через контроллер памяти осуществляется также обращение к внешней памяти, реализованной с помощью микросхем ОЗУ и ПЗУ. Контроллер памяти позволяет при одном обращении к памяти считывать или записывать как восьмиразрядные, так и шестнадцатиразрядные коды.

Максимальный суммарный объем внешней и внутренней памяти (без РОЗУ) у микроконтроллеров большинства типов составляет 64Кх8 бит. У микроконтроллеров подсемейства NT суммарный объем памяти может быть доведен до 1Мх8 бит. Микроконтроллеры, в обозначении типа которых на втором месте стоит цифра 0 (X=0), не имеют внутреннего ПЗУ. Его функции реализуются с помощью микросхем ПЗУ, входящих в состав внешней памяти.

**ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА МИКРОКОНТРОЛЛЕРА** семейства MCS-96 по выполняемым функциям могут быть отнесены к одной из шести групп:

1. устройства ввода и вывода данных, представленных многоразрядными двоичными кодами;
2. устройства ввода и вывода отдельных дискретных сигналов (включено-выключено);

3. устройства ввода и вывода аналоговых сигналов;
4. устройства обмена данными с другими микроконтроллерами и центральным процессором системы;
5. устройства приема и обслуживания запросов прерывания;
6. устройства контроля правильности функционирования микроконтроллера.

Для управления работой периферийных устройств и определения их состояния используются регистры специальных функций (Special Function Registers -SFR).

**ТАЙМЕРЫ** Два 16-разрядных таймера TIMER1 и TIMER2 обеспечивают синхронизацию работы устройства ввода-вывода импульсных сигналов (HSIO, High Speed In/Out unit) с реальным временем и внешними событиями. TIMER1 синхронизируется изнутри, тогда как TIMER2 синхронизируется снаружи.

**CODE RAM** Это дополнительное ОЗУ, в котором можно размещать исполняемый код. Этот код будет выполняться очень быстро, так как Code RAM имеет 16-разрядный интерфейс с нулевым циклом ожидания. Code RAM может принести существенную пользу в задачах, где требуется максимально быстрое выполнение только небольших фрагментов кода, позволяя при этом использовать сравнительно медленное и дешевое 8-битное ПЗУ для хранения остальной части программы. Конечно, эту память можно использовать и для размещения данных или стека.

**ЭНЕРГОПОТРЕБЛЕНИЕ** Общее потребление - не более 75 мА на частоте 16 МГц. Имеются режимы с пониженным энергопотреблением: IDLE (30 мА) и POWER DOWN (0,1 мА).

**ТЕМПЕРАТУРНЫЙ ДИАПАЗОН, КОРПУСА** Существует четыре разновидности по температурному диапазону работы: коммерческий (0...+70 градусов), расширенный (-40...+85), автомобильный (-40...+125) и военный. Кроме того, микроконтроллеры могут быть подвергнуты динамической электротермотренировке. ИС устанавливаются в корпуса типов: PLCC-68, QFP-80, керамический LCC-68, и керамический PGA-68.

## 7.2. Периферийные устройства.

### 7.2.1. Устройства ввода и вывода данных.

Ввод и вывод данных, представленных многоразрядными двоичными кодами, осуществляется через параллельные порты. В микроконтроллерах семейства MCS-96 используются восьмиразрядные и четырехразрядные порты. При этом микроконтроллер может иметь от четырех до восьми портов.

Два восьмиразрядных порта (P3 и P4) предназначены для подключения внешней памяти. Использование этих портов для ввода и вывода данных возможно лишь при ее отсутствии.

тип линии порта	кол-во в порту 0	кол-во в порту 1	кол-во в порту 2	кол-во в портах 3 и 4	кол-во в порту HSIO	всего
двунаправленная		8	2	16/0	2	28/12
только вход	8		4		2	14
только выход			2		4	6
				порты 3/4 заняты если используется внешняя шина		48/32

Отдельные выводы параллельных портов могут выполнять альтернативные функции (прием запросов прерывания, вывод сигналов управления и др.). Для перевода выводов портов в режим альтернативных функций необходимо заслать определенное управляющее слово в соответствующий регистр специальных функций.

### 7.2.2. Устройство ввода и вывода дискретных сигналов.

Дискретные сигналы (включено-выключено) широко используются в системах управления.

Изменение значения дискретного сигнала называется событием.

В микроконтроллерах семейства MCS-96 для обработки входных и формирования выходных событий используются специальные периферийные устройства, осуществляющие быстрый ввод и быстрый вывод без непосредственного участия процессора.

Быстрый ввод заключается в обнаружении события определенного типа на определенном входе микроконтроллера и запоминание времени его наступления в заданной системе отсчета времени. Быстрый вывод заключается в формировании события определенного типа на заданном выходе микроконтроллера в заданный момент времени.

Для выполнения операций быстрого ввода и вывода в микроконтроллерах разных типов используются или блок быстрого ввода-вывода (HSIO), или блок процессоров событий (EPA).

В обоих блоках для формирования текущего значения времени используются шестнадцатиразрядные таймеры-счетчики, на счетные входы которых подаются сигналы времени от внутреннего генератора или от внешнего источника.

В блоке быстрого ввода-вывода (HSIO) для обработки входных событий и формирования выходных событий используются специализированные модули для ввода и для вывода, а в блоке процессоров событий (EPA) содержится набор универсальных модулей, каждый из которых при программировании настраивается на работу или в режиме быстрого ввода (capture-захвата) или в режиме быстрого вывода (compare-сравнения).



По результатам обработки входных событий могут вычисляться параметры импульсных последовательностей на входах микроконтроллера – период следования импульсов, их длительность, сдвиг во времени между импульсами на разных входах и другие параметры.

Блоки HSIO и EPA кроме операций быстрого ввода и вывода могут использоваться для формирования временных задержек (режим программного таймера) формирования сигналов специальной формы (например, сигнала с широтно-импульсной модуляцией), запуска аналого-цифрового преобразователя и выполнения некоторых других функций.

### **7.2.3. Устройства ввода и вывода аналоговых сигналов**

У микроконтроллеров большинства типов в число периферийных устройств входит многоканальный аналого-цифровой преобразователь (ADC). Число каналов может быть равно 4, 6, 8, 13 или 14. Входное напряжение в канале может изменяться в пределах от 0 до 5 (5,12)В. В результате преобразования формируется восьмиразрядный или десятиразрядный двоичный код.

Запуск преобразования в канале может производиться по команде в программе или по сигналу из блока HSIO или EPA в заранее заданное время. Некоторые преобразователи могут работать в режиме сканирования входов.

На частоте 16 МГц время преобразования - 19,5 мкс. Имеется схема выборки/хранения и отдельные входы опорного напряжения и аналоговой земли.

Преобразование цифровых данных в аналоговый сигнал выполняется с использованием широтно-импульсного модулятора (PWM). Широтно-импульсный модулятор формирует последовательность прямоугольных импульсов, следующих с постоянным периодом. Длительность импульса пропорциональна числу, преобразуемому в значение аналогового сигнала. Получаемая импульсная последовательность с выхода микроконтроллера с переменной скважностью подается на внешнюю интегрирующую схему, с выхода которой снимается аналоговый сигнал.

Диапазон изменения скважности импульсов - 256 градаций. Период импульсов может быть равен 256 или 512 тактам (31,25 или 15,625 кГц соответственно, для частоты 16 МГц).

В микроконтроллерах подсемейства MC кроме двух широтно-импульсных модуляторов имеется специальный блок содержащий три широтно-импульсных модулятора, работающих совместно. Этот блок, называемый генератором периодических колебаний (WG), имеет три пары выходов. Разность напряжений на выходах одной пары представляет собой синусоидоподобный ступенчатый сигнал. Сигналы, снимаемые с трех пар выходов, могут быть использованы для питания трехфазных индукционных двигателей переменного тока. Блок позволяет также формировать сигналы для управления вентильными двигателями постоянного тока, шаговыми двигателями и для некоторых других целей.

### **7.2.4. Устройства обмена данными с другими микроконтроллерами и центральным процессором.**

Обмен данными с другими микроконтроллерами в управляющей системе, содержащей несколько совместно работающих микроконтроллеров, может осуществляться по последовательному каналу или путем совместного использования внешней памяти.

Обмен данными по последовательному каналу выполняется с использованием последовательного порта (SP). Обмен производится путем отправки отдельных кадров, каждый из которых содержит стартовый бит, семь или восемь информационных битов и один стоповый бит. В состав кадра может быть включен дополнительный бит, который используется для контроля по четности правильности пересылки данных или для различения кадров, содержащих адреса абонентов, и кадров содержащих данные, при включении контроллера в простейшую локальную сеть.

Последовательный порт может также осуществлять последовательный ввод или вывод байтов с использованием внешних сдвигающих регистров, которые в этом случае выполняют функции дополнительных параллельных портов ввода или вывода.

У микроконтроллеров некоторых типов в число периферийных устройств входит второй последовательный порт (SSIO), с помощью которого осуществляется непосредственный обмен байтами между двумя микроконтроллерами путем последовательной передачи байта и сопровождающей серии импульсов сдвига. Порт SSIO содержит два последовательных канала, каждый из которых может работать в режиме передачи или в режиме приема. Максимальная скорость обмена (на частоте 16 МГц): в асинхронном режиме - 1 Мбод; в синхронном режиме - 4 Мбод.

Микроконтроллеры почти всех модификаций имеют аппаратные средства, обеспечивающие совместное использование внешней памяти несколькими микроконтроллерами. Согласование работы микроконтроллеров при обращении к внешней памяти реализуется с помощью сигналов HOLD, HLDA, BREQ и дополнительной внешней аппаратуры.

У микроконтроллеров некоторых типов имеется "подчиненный" порт (Slave Port), предназначенный для обмена данными с центральным процессором в иерархической управляющей системе. Через "подчиненный" порт микроконтроллер подключается непосредственно к системной магистрали микропроцессорной системы. Обмен данными происходит под управлением центрального процессора, который обращается к микроконтроллеру, как к собственному порту ввода и вывода. При появлении необходимости передать данные в центральный процессор микроконтроллер посылает запрос прерывания.

### 7.2.5. Устройства приема и обслуживания запросов прерывания.

Запросы прерывания текущей программы могут поступать от внешних источников или формироваться внутри микроконтроллера в различных периферийных устройствах. Общее число источников запросов прерывания у микроконтроллеров разных типов может быть 21, 28 или 37.

Запросы прерывания могут маскироваться путем отправки кодов маски в соответствующие регистры специальных функций. В микроконтроллерах всех типов имеется программный контроллер прерываний (PIC). Обслуживание запроса прерывания с использованием PIC заключается в переходе от выполнения текущей программы к выполнению другой определенной программы, составленной разработчиком программного обеспечения.

Адрес первой команды каждой прерывающей программы (вектор прерывания) хранится в определенной паре ячеек ПЗУ в области памяти, отведенной для хранения специальных данных. После завершения выполнения прерывающей программы происходит возврат к прерванной программе.

Прерывающая программа в свою очередь может быть прервана при поступлении любого незамаскированного запроса прерывания вне зависимости от соотношения приоритетов запроса, вызвавшего переход к данной программе, и нового запроса прерывания.

В микроконтроллерах некоторых типов кроме программного контроллера прерываний имеется микропрограммный контроллер прерываний (PTS). Любой запрос прерывания, кроме нескольких особых запросов, может быть направлен для обслуживания или в PIC или в PTS.

Обслуживание запроса прерывания с использованием PTS заключается в выполнении типовой микропрограммы, при этом выполнение операций по микропрограмме совмещается во времени с выполнением команд текущей программы. Микропрограммы PTS охватывают, в основном, пересылки данных. Прерывания, обслуживаемые PTS, обрабатываются быстрее, чем те, которые обслуживаются обычным способом. Однако, программировать PTS непросто, а отлаживать еще сложнее.

### 7.2.6. Устройства контроля правильности функционирования микроконтроллера.

Все микроконтроллеры семейства MCS-96 имеют сторожевой таймер (WDT). Сторожевой таймер по прошествии определенного интервала времени переводит микроконтроллер в состояние сброса. Правильно работающая программа должна предотвращать сброс микроконтроллера от WDT путем периодического сброса в нулевое состояние самого WDT. При сбое в ходе программы сторожевой таймер своевременно не сбрасывается, и при его переполнении микроконтроллер переводится в состояние сброса, что предотвращает появление и развитие опасных ситуаций в системе управления.

Микроконтроллеры некоторых типов имеют схему обнаружения падения частоты генератора тактовых импульсов (OFD). При снижении частоты ниже определенного уровня OFD вырабатывает сигнал сброса и переводит микроконтроллер в состояние сброса. Это предотвращает появление опасных комбинаций сигналов на выходах микроконтроллера, которые могут возникнуть при остановке генератора тактовых импульсов в произвольный момент времени в процессе выполнения программы.

### 7.2.7. Характеристики микроконтроллеров подсемейств.

К числу основных функциональных характеристик микроконтроллера относятся:

- емкость расположенных на кристалле регистрового оперативного запоминающего устройства (RRAM), постоянного запоминающего устройства (ROM), оперативного запоминающего устройства (RAM);
- максимальная тактовая частота ( $F_{max}$ );
- число команд в системе команд ( $N$ );
- состав периферийных устройств.

Основные функциональные характеристики базовых микроконтроллеров подсемейств приведены в таблице.

Емкость запоминающих устройств указана в байтах.

По значению тактовой частоты может быть определено быстродействие микроконтроллера. У микроконтроллеров подсемейства 8X9Y команды коротких операций выполняются за 12 периодов тактовой частоты. При тактовой частоте 12 МГц микроконтроллеры данного подсемейства имеют быстродействие 1000000 коротких операций в секунду. У микроконтроллеров остальных подсемейств команды коротких операций выполняются за 8 периодов тактовой частоты, и при тактовой частоте 16 МГц обеспечивается быстродействие 2000000 коротких операций в секунду.

При этом следует иметь в виду, что короткие операции в микроконтроллере семейства MCS-96 по своему содержанию существенно отличаются от коротких операций в микроконтроллере с регистром-аккумулятором. Так, например, одной короткой операции "сложение" в микроконтроллере семейства MCS-96 при представлении данных в формате "байт" соответствует последовательность из трех коротких операций в микроконтроллере семейства MCS-51, а при представлении данных в формате "слово" - соответствует последовательность из шести коротких операций.

Кроме того, в систему команд микроконтроллеров семейства MCS-96 входят команды умножения и деления чисел в формате "слово". В микроконтроллерах других семейств такие операции выполняются по подпрограммам, что резко увеличивает время их выполнения.

Отмеченные особенности существенно сокращают время вычислений в микроконтроллерах семейства MCS-96 по сравнению с микроконтроллерами других семейств.

В систему команд микроконтроллеров, изготавливаемых по КМДП технологии, входят различные дополнительные команды, в числе которых имеется команда перевода микроконтроллера в энергосберегающие режимы - режим холостого хода и режим пониженного энергопотребления. В режиме холостого хода программа не выполняется, но функционируют все периферийные устройства, при этом потребление энергии от источника питания уменьшается на 60%. В режиме пониженного энергопотребления прекращаются все процессы в микроконтроллере, но сохраняются данные в РОЗУ и ОЗУ. При этом ток потребления составляет единицы микроампер.

С появлением на рынке цифровых интегральных схем микроконтроллеров семейства MCS-96 фирмы Intel перед разработчиками систем, содержащих встроенные микроконтроллеры, открываются новые большие возможности по созданию высоко совершенных, малогабаритных, экономичных и надежных систем, приборов и устройств различного назначения.

Кристалл	Частота (МГц)/ быстродействие (MIPS)	Адресное пр-во	ПЗУ	Регист- ры	Доп. ОЗУ	Тайме- ры	Каналы АЦП	Линии в/в	HSIO/ EPA	Унив. посл. порты	Синхр. посл. порты	PTS	ШИМ
Device	Clock (Mhz)/ Speed (MIPS)	Address Space	ROM	Registers	Code RAM	Timers	A/D Channels	I/O Pins	HSIO/ EPA	UARTs	SSIO	PTS	PWM
8X96BH	12/1	64K	8K	232	нет	2	8	48	HSIO	1	нет	нет	1
8XC196KB	16/2	64K	8K	232	нет	2	8	48	HSIO	1	нет	нет	1
8XC198	16/2	64K	8K	232	нет	2	4	48	HSIO	1	нет	нет	1
8XC196KC	20/2.5	64K	16K	488	нет	2	8	48	HSIO	1	нет	да	3
8XC196KD	20/2.5	64K	32K	1000	нет	2	8	48	HSIO	1	нет	да	3
8XC196KR/KQ	16/2	64K	16K/12K	488/360	256/128	2	8	56	10 EPA	1	1	да	нет
8XC196JR/JQ	16/2	64K	16K/12K	488/360	256/128	2	6	41	EPA	1	1	да	нет
8XC196KT/KS	16/2	64K	32K/24K	1000	512/256	2	8	56	10 EPA	1	1	да	нет
8XC196JT/JS	16/2	64K	32K/24K	1000	512/256	2	6	41	EPA	1	1	да	нет
8XC196JV	20/2.5	64K	48K	1.5K	512	2	6	41	EPA	1	1	да	нет
8XC196MC	16/2	64K	16K	488	нет	2	13	53	EPA	нет	нет	да	нет
8XC196MD	16/2	64K	16K	488	нет	2	14	64	EPA	нет	нет	да	нет
8XC196MH	16/2	64K	32K	744	нет	2	8	50	EPA	2	нет	да	нет
8XC196CA	20/2.5	64K	32K	1000	256	2	6	44	EPA	1	1	да	нет
8XC196NT	20/2.5	1M	32K	1000	512	2	4	56	10 EPA	1	1	да	нет
8XC196CB	20/2.5	1M	56K	1.5K	512	2	8	56	10 EPA	1	1	да	нет
8XC196NP	25/3	1M		1000	нет	2	нет	32	EPA	1	нет	да	3
80C196NU	50/6	1M	нет	1000	нет	2	нет	32	EPA	1	нет	да	3
8XC296SA	50/16	6M	2K	512	2K	2	нет	32	EPA	1	нет	нет	3

Таблица.14. Номенклатура MCS-96

### 7.2.8. Почему 80C196 быстрее, чем 8051?

Кристаллы 80C196 изготавливаются по более современной технологии (с меньшим размером элементов на кристалле), поэтому достигаются более высокие тактовые частоты. Например, кристалл 80C196NU имеет тактовую частоту 50 МГц, а наиболее быстродействующие из семейства 8051 - 24 МГц

Все 232 внутренних регистра 80C196 имеют статус " аккумуляторов " - к ним можно непосредственно применять все необходимые арифметические и логические операции. У 8051 для достижения тех же результатов зачастую необходимо выполнять дополнительные пересылки в аккумулятор и из него.

У 80C196 можно использовать 16-разрядную внешнюю шину. Кроме того, цикл шины 80C196 в 3-4 раза короче, чем у 8051. В результате, 80C196 в 6-8 раз быстрее работает с внешней памятью. Отметим, что, для того, чтобы снизить стоимость изделий, можно вводить в шину циклы ожидания и сократить ее ширину до 8-ми бит; но даже в этом случае 80C196 будет иметь преимущество в 2-3 раза.

В задачах, требующих 16- и 32-разрядных вычислений, 80C196 примерно на порядок быстрее, поскольку имеет полноценный набор 16-разрядных арифметических инструкций.

Одним из самых эффективных способов сокращения времени разработки программ для микроконтроллеров является применение языка Си. Язык Си базируется на широком использовании стека и указателей. Однако для 8051 использование Си затруднено и ведет к большим накладным расходам, и вот почему. Поскольку 8051 имеет небольшой стек, Си-компиляторы для 8051 генерируют дополнительный код, эмулирующий большой стек во внешней памяти данных. Вдобавок 8051 имеет всего один 16-разрядный указатель - DPTR, и компиляторам также приходится генерировать дополнительный код, чтобы компенсировать этот недостаток. Все это приводит к замедлению программ и увеличению их размера. У 80C196 таких проблем нет - стек имеет размер до 64 КБайт, а в качестве указателя можно использовать любое из 116 слов встроенной регистровой памяти.

## **8. Особенности программирования микроконтроллеров и процессоров цифровой обработки сигналов.**

### **8.1.1. Общие особенности.**

Особенность написания и отладки программного обеспечения для однокристальных микро-ЭВМ (микроконтроллеров) и процессоров цифровой обработки сигналов (ЦПОС) состоит в том, что для этого, как правило, совершенно недостаточно иметь системы, состоящей только из программируемого микроконтроллера или ЦПОС. Это связано с тем, что,

- во-первых, как правило, ресурсов микроконтроллера (объема памяти, быстродействия) не достаточно для размещения и функционирования даже простейших сервисных программ (редактора текста, транслятора и отладочного монитора) необходимых для написания и отладки программы, если она даже будет написана на Ассемблере;
- во вторых, некоторые архитектурные особенности (раздельные области памяти для хранения программ и данных, устройства защиты памяти программ) микроконтроллеров затрудняют или делают просто невозможным редактирование (написание, отладку) программ, по которым они работают.

Все это заставляет при разработке программ для микроконтроллеров использовать специальные средства - называемыми инструментальными средствами разработки и отладки .

В случае построения средств разработки и отладки на базе универсального компьютера становится возможным существенно облегчить разработку программ - использовать языки высокого уровня - С, Паскаль, построить дружелюбный интерфейс, использовать принципы объектного и визуального программирования и пр. Рассмотрим различные варианты построения инструментальных средств разработки и отладки.

### **8.2. Типы инструментальных средств разработки и отладки программ для микроконтроллеров и процессоров цифровой обработки сигналов.**

К числу основных инструментальных средств отладки относятся:

- Внутрисхемные эмуляторы;
- Программные симуляторы;
- Платы развития;
- Мониторы отладки;
- Эмуляторы ПЗУ.

Данный список не исчерпывает всех типов существующих инструментальных средств отладки. Кроме указанных, существуют и комбинированные устройства и наборы, которые позволяют компенсировать недостатки основных средств, взятых порознь.

### **8.3. Внутрисхемные эмуляторы.**

#### **8.3.1. Принцип работы.**

Внутрисхемный эмулятор - программно аппаратное средство, способное замещать собой эмулируемый (моделируемый) процессор в реальной схеме. Внутрисхемный эмулятор - это наиболее мощное и универсальное отладочное средство.

По сути дела, «хороший» внутрисхемный эмулятор делает процесс функционирования отлаживаемого контроллера прозрачным, т.е. легко контролируемым, произвольно управляемым и модифицируемым по воле разработчика.

Обычно стыковка внутрисхемного эмулятора с отлаживаемой системой производится при помощи эмуляционного кабеля со специальной эмуляционной головкой. Эмуляционная головка вставляется вместо микроконтроллера в отлаживаемую систему. Если микроконтроллер невозможно удалить из отлаживаемой системы, то использование эмулятора возможно, только если этот микроконтроллер имеет отладочный режим, при котором все его выводы находятся в третьем состоянии. В этом случае для подключения эмулятора используют специальный адаптер-клипсу, который подключается непосредственно к выводам эмулируемого микроконтроллера.

#### **8.3.2. Классификация внутрисхемных эмуляторов.**

Функционально внутрисхемные эмуляторы делятся на

- стыкуемые с внешней вычислительной машиной (обычно это бывает IBM PC), и
- функционирующие автономно.

Автономные внутрисхемные эмуляторы имеют индивидуальные вычислительные ресурсы, средства ввода-вывода, не требуют для своей нормальной работы стыковки с какими-либо внешними вычислительными средствами, но за это пользователю приходится расплачиваться либо существенно более высокой ценой, либо пониженными функциональными и сервисными возможностями по сравнению с аналогичными моделями, стыкуемыми с IBM PC.

### 8.3.3. Функциональные возможности внутрисхемных эмуляторов.

Набор функциональных возможностей, которые предоставляют разработчику внутрисхемные эмуляторы весьма широк и включает в себя практически все разнообразие функциональных модулей средств разработок.

Также существенно облегчить работу разработчика может наличие в программной оболочке эмулятора встроенного редактора, встроенного менеджера проектов и системы управления. Тогда стирается грань между написанием программы, ее редактированием и отладкой. Переход от редактирования исходного текста к отладке - началу работы собственно эмулятора и обратно происходит «прозрачно» и синхронно с активизацией соответствующих окон, менеджер проектов автоматически запускает компиляцию по мере необходимости и активизирует соответствующие окна программного интерфейса.

При работе внутрисхемного эмулятора в составе интегрированной среды столь же просто можно осуществить и переход к отладке проекта с помощью имеющегося отладчика-симулятора или приступить к занесению в ПЗУ микроконтроллера отлаженной программой.

Некоторые модели внутрисхемных эмуляторов могут предоставлять пользователям и другие дополнительные возможности. Среди них отметим одну, хотя и достаточно специфическую, но в ряде случаев имеющую принципиальное значение: возможность построения многоэмуляторных комплексов, необходимых для отладки мультипроцессорных систем. Отличительной особенностью такого комплекса является возможность синхронного управления (с одного компьютера) несколькими эмуляторами.

### 8.3.4. Достоинства и недостатки внутрисхемных эмуляторов.

К достоинствам внутрисхемных эмуляторов следует отнести

- широкий набор функциональных возможностей, что делает внутрисхемные эмуляторы наиболее мощным и универсальным средством отладки;
- работу внутрисхемного эмулятора в реальной схеме электронного блока, в котором предполагается работа микроконтроллера или ЦПОС;
- большая гибкость моделирования временных и электрических характеристик микроконтроллера, что связано с преимущественно программным методом их моделирования

Однако внутрисхемные эмуляторы имеют и недостатки.

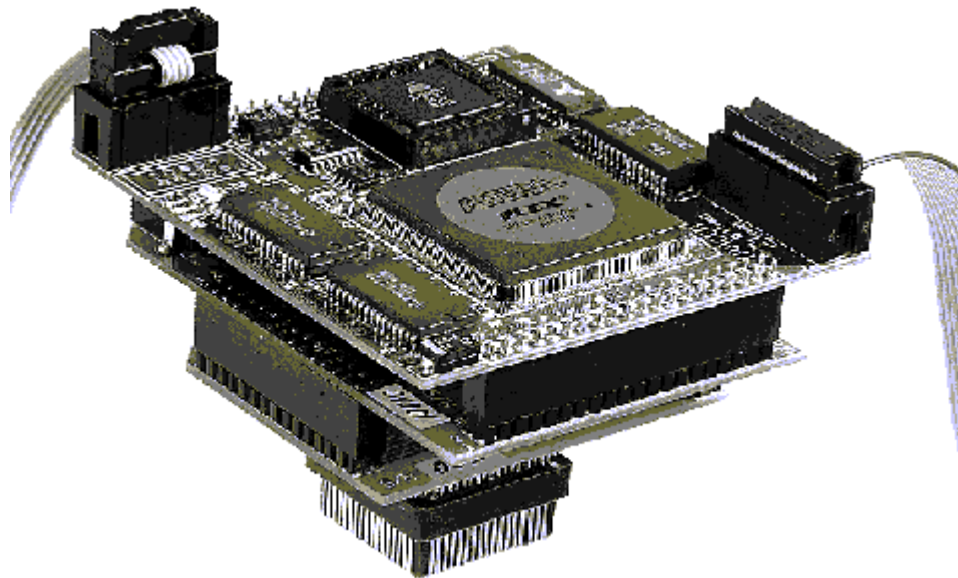
- Основным из них является трудность программного моделирования электрических сигналов на выводах микроконтроллера в реальном масштабе времени. Для адекватного моделирования быстроедействие моделирующего процессора или компьютера должно быть существенно выше, чем эмулируемого микроконтроллера, что достижимо далеко не всегда, особенно в случае эмуляции современных высокопроизводительных ЦПОС и микроконтроллеров.
- Кроме того, даже в случае работы в замедленном масштабе времени, различные модели внутрисхемных эмуляторов могут иметь разного рода ограничения по контролю и управлению функционированием отлаживаемых устройств, что связано трудностью их моделирования. Например, это может быть некорректное обрабатывание прерываний в пошаговом режиме, или запрет на использование последовательного порта и т.п..

## 8.4. PICE-51.

Возможности "реального" внутрисхемного эмулятора проиллюстрировано на примере модели PICE-51.

### 8.4.1. Внутрисхемный эмулятор 8-разрядных микроконтроллеров семейства 8051.

**PICE-51** - эмулятор нового поколения, созданный с применением новых технологий разработки аппаратуры и программного обеспечения.



**Рис.17. Внутрисхемный эмулятор PICE-51**

Применение программируемых матриц большой емкости позволило резко сократить размеры эмулятора без какого-либо ущерба его функциональным возможностям, минимизировать отклонения электрических и частотных характеристик эмулятора от характеристик эмулируемого процессора и, тем самым, добиться максимальной **точности эмуляции** на частотах до **30 МГц** при **напряжениях питания от 3.3В до 5В**.

Перезагружаемая аппаратная структура эмулятора обеспечивает эмуляцию практически всех микроконтроллеров семейства 8051 как отечественного производства, так и фирм: **Intel, Philips, Siemens, Atmel, Dallas, Temic, OKI, AMD, MHS** и других.

Мощный программный интерфейс в среде **Windows**, представляет собой интегрированную среду разработки, поддерживающую все этапы разработки программного обеспечения от написания исходного текста программы до ее компиляции и отладки. Программа поддержки эмулятора ориентирована на отладку программ на **языке высокого уровня по исходному тексту**.

Эмулятор состоит из **основной платы** размером 80x76мм, сменного **адаптера** под конкретный процессор и сменной **эмуляционной головки** под конкретный тип корпуса. На основной плате реализованы: **трассировщик, процессор точек останова**. Плата сменного адаптера содержит эмулирующий процессор под конкретный тип микроконтроллера. Эмуляционные головки обеспечивают установку эмулятора в колодки DIP и PLCC на плате пользователя. Питание эмулятора осуществляется от блока питания +5В, 0,5А или непосредственно от отлаживаемого устройства. Связь с компьютером - по **гальванически развязанному каналу RS-232C** на скорости 115 КБод..

#### **8.4.2. Характеристика аппаратуры.**

- Точная эмуляция - отсутствие каких-либо ограничений на использование программой пользователя ресурсов микроконтроллера.
- До 256К эмулируемой памяти программ и данных. Поддержка банкированной модели памяти. Распределение памяти между эмулятором и устройством пользователя с точностью до 1-го байта.
- До 512К аппаратных точек останова по доступу к памяти программ и данных.
- Аппаратная поддержка для отладки программ на языках высокого уровня.
- Трассировка 8 произвольных внешних сигналов.
- 4 выхода синхронизации аппаратуры пользователя.
- Трассировщик реального времени с буфером объемом от 16К до 64К фреймов по 64 бита с доступом "на лету". Трассировка адреса, данных, сигналов управления, таймера реального времени и 8-ми внешних сигналов пользователя.
- Программируемый фильтр трассировки.
- Аппаратный процессор точек останова с возможностью задания сложного условия останова эмуляции по комбинации сигналов адреса, данных, управления, 8-ми 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.
- Управляемый генератор тактовой частоты для эмулируемого процессора. Возможность плавного изменения тактовой частоты от 500 кГц до 40 МГц.
- Гальванически развязанный от компьютера канал связи RS-232C со скоростью обмена 115 КБод.
- Встроенная система самодиагностики аппаратуры эмулятора.

### 8.4.3. Характеристики программного обеспечения .

- Программное обеспечение ориентировано на работу в среде Windows на IBM-совместимых компьютерах с процессорами типа 386/486/Pentium;
- Встроенный многооконный **редактор** предназначен для написания исходных текстов программ. Редактор поддерживает операции с блоками текста, поиск/замену, цветное выделение синтаксических конструкций языка ассемблера и Си;
- Встроенный **менеджер проектов** обеспечивает автоматическую компиляцию программ. Все опции задаются в диалоговой форме. Переход от редактирования исходного текста к отладке и обратно происходит "прозрачно", т.е. менеджер проектов автоматически запускает компиляцию проекта при необходимости;
- PICE-51 обеспечивает символьную отладку и отладку по исходному тексту для программ, созданных с помощью следующих компиляторов:
  - ассемблер **ASM51** фирмы **Intel**;
  - ассемблер **MCA-51** фирмы **Фитон/МикроКосм**;
  - компилятор **PL/M** фирмы **Intel**;
  - ассемблер и компилятор **Си** фирмы **IAR Systems**;
  - ассемблер и компилятор **Си** фирмы **Avocet Systems Inc./HiTech**;
  - ассемблер и компилятор **Си** фирмы **Keil Software Inc.**;
- Автоматическое сохранение и загрузка файлов конфигурации аппаратуры, интерфейса и опций отладки. Обеспечивается совместимость файлов конфигурации с симулятором PDS-51. Обеспечена переносимость проектов между эмулятором PICE-51 и симулятором PDS-51;
- Возможность настройки цветов, шрифтов и других параметров для всех окон одновременно и для каждого окна в отдельности;

Эмулятор снабжен печатным **руководством по эксплуатации** и **контекстным электронным руководством**, в которых детально описаны его принципы работы, команды, меню, горячие клавиши.

### 8.4.4. Структурная схема эмулятора PICE-51.

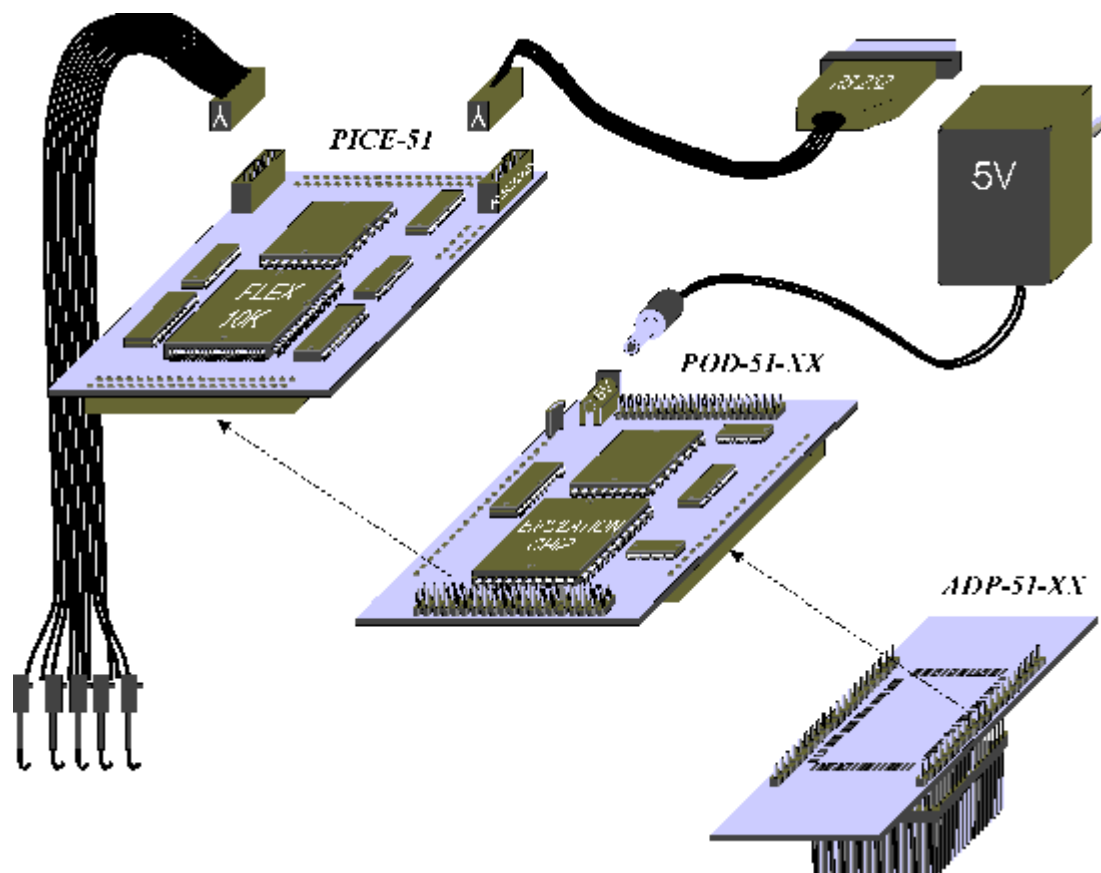


Рис. 18. Структурная схема эмулятора PICE-51

#### 8.4.5. Варианты комплектации эмулятора PICE-51.

Эмулируемый микроконтроллер	Эмуляция внутренней памяти программ	Комплект поставки		
		Основная плата	Сменный адаптер	Эмуляционная головка
<b>Intel:</b> 80C31/32, 80C51/52, 80L/C51FA, 80C51RA; <b>Philips:</b> 80C31/32, 80C51/52, 80C51FA; 80C51RA+, 80CL410, 80C524/528/550/652/654/575/576/851; <b>Atmel:</b> 89C51, 89C52, 89C55, 89S8252, 89S53; <b>Siemens:</b> SAB501/502/504/505/511/513; <b>MHS:</b> 80C31, 80C51, 80C32, 80C52, 80C154; <b>Oki:</b> 80C31, 80C51, 80C154; <b>AMD:</b> 80C31, 80C51, 80C52;	Нет	PICE-51	POD-51-31	ADP-51-DIP40 или ADP-51-LCC44
<b>Intel:</b> 80C31/32, 8XC51/52/54/58, 8XL/C51FA/FB/FC, 8XC51RA/RB/RC; <b>Philips:</b> 80C31/32, 8XC51/52/54/58, 8XC51FA/FB/FC; 8XC51RA+/RB+/RC+/RD+, 8XC524/528/550/652/654/575/576/851; 89C535/536/538; <b>Atmel:</b> 89C51, 89C52, 89C55; Siemens SAB501; <b>MHS:</b> 80C31, 8XC51, 80C32, 8XC52, 8XC154; <b>Oki:</b> 80C31, 8XC51, 8XC154; AMD: 80C31, 8XC51, 8XC52;	64K	PICE-51	POD-51-RX	ADP-51-DIP40 или ADP-51-LCC44
<b>Atmel:</b> 89C4051, 89C2051, 89C1051;	4K	PICE-51	POD-51-31 или POD-51-RX	ADP-51-2051
<b>Philips:</b> 80C451, 80C453; 87C451, 87C453	64K	PICE-51	POD-51-453	ADP-51-LCC68
<b>Philips:</b> 80C552, 80C562; 80C554 <b>Philips:</b> 80C552, 80C562; 80C554, 87C552, 87C562, 87C554	Нет 64K	PICE-51 PICE-51	POD-51-552 POD-51-554	ADP-51-LCC68 ADP-51-LCC68
<b>Intel:</b> 80C51GB	Нет	PICE-51	POD-51-GB	ADP-51-LCC68
<b>Dallas:</b> DS80C310, DS80C320, DS8XC520	64K	PICE-51	POD-51-DS530	ADP-51-DIP40 или ADP-51-LCC44
<b>Dallas:</b> DS8XC530	64K	PICE-51	POD-51-DS530	ADP-51-DS530

#### 8.4.6. Сравнительные характеристики некоторых эмуляторов для микроконтроллеров семейства 8051

Модель, Поставщик	EMUL-51 Nohau Corporation	USP-51 Signum Systems	iceMASTER-8051 MetaLink	PICE-5 Фитон1
Эмулируемые микроконтроллеры семейства 8051	Все известные разновидности	Все основные разновидности	Все основные разновидности	Все основные разновидности
Максимальная частота эмуляции	42 MHz	40 MHz	24 MHz	33 MHz
Максимальный объем эмулируемой памяти	320K	256K	128K	512K
Возможности перераспределения памяти между эмулятором и устройством пользователя	Блоками по 4K	Блоками по 256 байт	Блоками по 16 байт	С точностью до 1-го байта



Модель,Поставщик	EMUL-51 Nohau Corporation	USP-51 Signum Systems	iceMASTER-8051 MetaLink	PICE-5 Фитон1
Буфер трассировки	До 16К фреймов по 48 бит	До 32К фреймов по 80 бит	До 4К фреймов по 48 бит	До 64К фреймов по 64 бита
Доступ "на лету" к эмулируемой памяти и трассировщику	Да	Да	Да	Да
Интерфейс с компьютером	Карта в ISA слот	RS-232C,до 115 Kbaude	RS-232C, до 115 Kbaude	Изолированный RS-232C, до 115 Kbaude
Конструкция и размеры	Две карты в ISA формате, кабели 2 метра, ПОДы	Корпус 260x260x64 mm, кабель, ПОД	Корпус 178x140x25 mm, кабель, ПОД	Весь эмулятор - в эмуляционной головке размером 80x76x50 mm
Цена за сопоставимую конфигурацию поставки: поддержка 80C51, 25 MHz, 128K RAM, буфер трассировки объема 16К фреймов	\$5200(Nohau price list)	\$5300(Signum price list)	\$4000 (MetaLink price list)	\$990

Более полная информация, а так же демо-версия эмулятора, представлена на Website: <http://www.phyton.ru>

Конечно, столь широкий набор функциональных возможностей, делает внутрисхемные эмуляторы наиболее мощным и универсальным средством отладки

## 8.5. Программные симуляторы.

Симулятор - программное средство, способное имитировать работу микроконтроллера и его памяти. Как правило, симулятор содержит в своем составе:

- Отладчик;
- Модель ЦПУ и памяти.

Более продвинутые симуляторы содержат в своем составе модели встроенных периферийных устройств, таких, как таймеры, порты, АЦП, системы прерываний.

Симулятор должен уметь загружать файлы программ во всех популярных форматах, максимально полно отображать информацию о состоянии ресурсов симулируемого микроконтроллера, а также предоставлять возможности по симуляции выполнения загруженной программы в различных режимах. В процессе отладки модель «выполняет» программу, и на экране компьютера отображается текущее состояние модели.

Загрузив программу в симулятор, пользователь имеет возможность запускать ее в пошаговом или непрерывном режимах, задавать условные и безусловные точки останова, контролировать и свободно модифицировать содержимое ячеек памяти и регистров симулируемого микропроцессора. С помощью симулятора можно быстро проверить логику выполнения программы, правильность выполнения арифметических операций.

В зависимости от класса используемого отладчика, различные симуляторы могут поддерживать высокоуровневую символьную отладку программ.

Некоторые модели симуляторов могут содержать ряд дополнительных программных средств, таких, например, как: интерфейс внешней среды, встроенную интегрированную среду разработки.

В реальной системе микроконтроллер обычно занимается считыванием информации с подключенных внешних устройств (датчиков), обработкой этой информации и выдачей управляющих воздействий на исполнительные устройства. Чтобы в симуляторе не обладающем интерфейсом внешней среды смоделировать работу датчика, нужно вручную изменять текущее состояние модели периферийного устройства, к которому в реальной системе подключен датчик. Если, например, при приеме байта через последовательный порт взводится некоторый флажок, а сам байт попадает в определенный регистр, то оба эти действия нужно производить в таком симуляторе вручную. Наличие же интерфейса внешней среды позволяет пользователю создавать и гибко использовать модель внешней среды микроконтроллера, функционирующую и взаимодействующую с отлаживаемой программой по заданному алгоритму.

Очевидной особенностью программных симуляторов является то обстоятельство, что исполнение программ, загруженных в симулятор, происходит в масштабе времени, отличном от реального. Однако, низкая цена, возможность ведения отладки даже в условиях отсутствия макета отлаживаемого устройства делают программные симуляторы весьма эффективным средством отладки. Отдельно необходимо подчеркнуть, что существует целый класс ошибок, которые могут быть обнаружены только при помощи симулятора.

## 8.6. Платы развития.

Платы развития, или как принято их называть в зарубежной литературе - оценочные платы (Evaluation Boards), являются своеобразными конструкторами для макетирования прикладных систем. В последнее время, при выпуске новой модели кристалла микроконтроллера, фирма-производитель обязательно выпускает и соответствующую плату развития. Обычно это печатная плата с установленным на ней микроконтроллером, плюс вся необходимая ему стандартная обвязка. На этой плате также устанавливаются схемы связи с внешним компьютером. Как правило, там же имеется свободное поле для монтажа прикладных схем пользователя. Иногда имеется уже готовая разводка для установки дополнительных устройств, рекомендуемых фирмой. Например, ПЗУ, ОЗУ, ЖКИ-дисплей, клавиатура, АЦП и др. Кроме учебных или макетных целей, такие доработанные пользователем платы стало выгодно (экономия времени) использовать в качестве одноплатных контроллеров, встраиваемых в мало серийную продукцию (5..20 шт.).

Для большего удобства, платы развития комплектуются еще и простейшим средством отладки на базе монитора отладки. Однако, здесь проявились два разных подхода: один используется для микроконтроллеров, имеющих внешнюю шину, а второй - для микроконтроллеров, не имеющих внешней шины.

В первом случае отладочный монитор поставляется фирмой в виде микросхемы ПЗУ, которая вставляется в специальную розетку на плате развития. Плата также имеет ОЗУ для программ пользователя и канал связи с внешним компьютером или терминалом. Примером здесь может служить плата развития фирмы Intel для микроконтроллера 8051.

Во втором случае, плата развития имеет встроенные схемы программирования внутреннего ПЗУ микроконтроллера, которые управляются от внешнего компьютера. В этом случае, программа монитора просто заносится в ПЗУ микроконтроллера совместно с прикладными кодами пользователя. Прикладная программа при этом специально должна быть подготовлена: в нужные ее места вставляют вызовы отладочных подпрограмм монитора. Затем осуществляется пробный прогон. Чтобы внести в программу исправления, пользователю надо стереть ПЗУ и произвести повторную запись. Готовую прикладную программу получают из отлаженной путем удаления всех вызовов мониторных функций и самого монитора отладки.

Важно отметить, что, плюс к монитору, иногда платы развития комплектуются еще и программами отладки, которые запускаются на внешнем компьютере в связке с монитором. Эти программы в последнее время заметно усложнились и зачастую имеют высоко-профессиональный набор отладочных функций, например, отладчик-симулятор или различные элементы, присущие в чистом виде интегрированным средам разработки. В состав поставляемых комплектов могут входить и программы прикладного характера, наиболее часто встречающиеся на практике.

Возможности по отладке, предоставляемые комплектом "плата развития плюс монитор", безусловно, не столь универсальны, как возможности внутрисхемного эмулятора, да и некоторая часть ресурсов микропроцессора в процессе отладки отбирается для работы монитора. Тем не менее, наличие законченного набора готовых программно-аппаратных средств, позволяющих без потери времени приступить к монтажу и отладке прикладной системы, во многих случаях является решающим фактором. Особенно если учесть, что стоимость такого комплекта несколько меньше, чем стоимость более универсального эмулятора.

## 8.7. Отладочные мониторы.

### 8.7.1. Принцип работы.

**Отладочный монитор - специальная программа, загружаемая в память отлаживаемой системы.** Она вынуждает процессор пользователя производить, кроме прикладной задачи, еще и отладочные функции:

- Загрузку прикладных кодов пользователя в свободную от монитора память;
- Установку точек останова;
- Запуск и останов загруженной программы в реальном времени;
- Проход программы пользователя по шагам (часть функций трассировщика);
- Просмотр, редактирование содержимого памяти и управляющих регистров.

Программа монитора обязательно должна работать в связке с внешним компьютером или пассивным терминалом, на которых и происходит визуализация и управление процессом отладки. Повторим, что отладочные мониторы используют тот процессор, который уже стоит на плате пользователя.

### 8.7.2. Достоинства и недостатки отладочных мониторов.

Достоинством этого подхода являются очень малые затраты при сохранении возможности вести отладку в реальном времени.

Главным недостатком является отвлечение ресурсов микроконтроллера на отладочные и связанные процедуры, например: монитор занимает некоторый объем памяти, прерывания, последовательный канал. Объем отвлекаемых ресурсов зависит от искусства разработчика монитора. В последнее время появились изделия, которые практически не занимают аппаратных ресурсов процессора, о них рассказано в разделе эмуляторы ПЗУ.

Как правило каждая фирма-разработчик семейства микроконтроллеров или ЦПОС выпускает и вариант отладочного монитора, он обычно поставляется вместе с платами развития.

## 8.8. Эмуляторы ПЗУ.

Эмулятор ПЗУ - программно аппаратное средство, позволяющее замещать ПЗУ на отлаживаемой плате, и подставляющее вместо него ОЗУ, в которое может быть загружена программа с компьютера через один из стандартных каналов связи. Это устройство позволяет пользователю избежать многократных циклов перепрограммирования ПЗУ. Эмулятор ПЗУ имеет смысл только для микроконтроллеров, которые в состоянии обращаться к внешней памяти программ. Это устройство сравнимо по сложности и по стоимости с платами развития. Оно имеет одно большое достоинство: универсальность. Эмулятор ПЗУ может работать с любыми типами микроконтроллеров.

Ранние эмуляторы ПЗУ позволяли только загружать программу, запускать ее и останавливать, используя общий сброс. Затем появились усложненные модели с аппаратной выработкой сигналов трассировки по достижении определенного адреса на осциллограф. Эмулируемая память в таких изделиях была доступна для просмотра и модификации, но очень важный контроль за внутренними управляющими регистрами микроконтроллера был до недавнего времени невозможен.

Однако появились модели интеллектуальных эмуляторов ПЗУ, которые позволяют «заглядывать» внутрь микроконтроллера на плате пользователя и вообще, по управлению отладкой, стали похожими на внутрисхемный эмулятор. Фирма Sactus даже представляет свой фактически интеллектуальный эмулятор ПЗУ, как внутрисхемный эмулятор ряда микропроцессоров, настолько невозможно отличить работу с тем и другим. В действительности, процессор здесь не замещается, а используется тот, что стоит на плате пользователя.

Интеллектуальные эмуляторы ПЗУ представляют собой гибрид из обычного эмулятора ПЗУ, монитора отладки и схем быстрого переключения шины с одного на другой. Этим создается эффект, как если бы монитор отладки был установлен на плате пользователя и при этом он не занимает у микроконтроллера никаких аппаратных ресурсов, кроме небольшой зоны программных шагов, примерно 4К. Например, такое устройство разработала фирма «Фитон» для всех существующих и будущих микроконтроллеров, которые имеют ядро от 8051, но дополнительно насыщены различными устройствами ввода\вывода. Это устройство поддерживает множество самых разных микроконтроллеров фирм Philips, Siemens, OKI.

## 8.9. Типичные функциональные модули средств разработки и отладки.

Любое из перечисленных инструментальных средств состоит из нескольких взаимодействующих (программных либо аппаратных) функциональных модулей. Каждый из них обеспечивает определенный круг сервисных услуг при разработке и отладке программ. Некоторые из модулей специфичны для того или иного типа инструментальных средств разработки, другие используются во практически во всех вариантах систем разработки программ для микроконтроллеров.

Как минимум, система разработки содержит следующие функциональные блоки

- Отладчик;
- Узел эмуляции микроконтроллера;
- Эмуляционная память;
- Подсистема точек останова.

Более продвинутые модели могут содержать дополнительно:

- Процессор точек останова;
- Трассировщик;
- Профилировщик (анализатор эффективности программного кода);
- Таймер реального времени;
- Программно-аппаратные средства, обеспечивающие возможность чтения и модификации ресурсов эмулируемого процессора «на лету», т.е. в процессе выполнения программы пользователя в реальном времени;
- Программно-аппаратные средства, обеспечивающие синхронное управление, необходимое для эмуляции в мультипроцессорных системах;
- Интегрированную среду разработки.

## 8.10. Отладчик.

Отладчик является своеобразным мостом между разработчиком и отладочным средством. Состав и объем информации, проходящей через средства ввода-вывода, доступность ее для восприятия, контроля, и, при необходимости, для коррекции и модификации напрямую зависят от свойств и качества отладчика.

Хороший отладчик позволяет осуществлять:

- Загрузку отлаживаемой программы в память системы;
- Вывод на монитор состояния и содержимого всех регистров и памяти, и при необходимости, их модификацию;
- Управление процессом эмуляции.

Более мощные отладчики, обычно их называют высокоуровневыми (High-Level Debuggers), помимо этого, позволяют:

- Вести символьную отладку, благодаря тому, что отладчик "знает" адреса всех символьных переменных, массивов и структур (за счет использования специальной информации, поставляемой компилятором). При этом пользователь может оперировать более приемлемыми для человека символьными именами, не утруждая себя запоминанием их адресов;
- Контролировать и анализировать не только дисассемблированный текст, но и исходный текст программы, написанной на языке высокого уровня, и даже с собственными комментариями.

Такой отладчик позволяет пользователю одновременно контролировать ход выполнения программы и видеть соответствие между исходным текстом, образом программы в машинных кодах, и состоянием всех ресурсов эмулируемого микроконтроллера.

Следует отметить, что высокоуровневый отладчик обеспечивает выполнение всех своих функций только в том случае, если используется кросс-компилятор, поставляющий полную и правильную отладочную информацию (не все компиляторы, особенно их пиратские версии, поставляют такую информацию) и при этом формат ее представления должен быть "знаком" отладчику.

## 8.11. Узел эмуляции микроконтроллера.

Узел эмуляции микроконтроллера - модуль, позволяющий моделировать микроконтроллер.

Данный блок необходим в системах разработки на основе внутрисхемных эмуляторов и симуляторов, в других вариантах средств разработки в системах присутствует реальный микроконтроллер и, поэтому его эмуляция не нужна.

Как правило, при эмуляции микроконтроллера предусматривается возможность запуска программ, их останова и выполнения с различной скоростью, в том числе и в пошаговом режиме. Также обычной является функция просмотра и изменение содержимого внутренних регистров микроконтроллера и состояния его внешних выводов.

## 8.12. Эмуляционная память.

Наличие эмуляционной памяти дает возможность использовать ее в процессе отладки вместо ПЗУ в отлаживаемой системе, и более того, отлаживать программу без использования реальной системы или ее макета. При необходимости внесения изменений в отлаживаемую программу достаточно загрузить новую или модифицированную программу в память эмулятора, вместо того чтобы заниматься перепрограммированием ПЗУ.

Существуют модели эмуляторов, которые позволяют пользователю «подставлять» вместо ПЗУ эмуляционную память не только целиком, но и поблочко (в некоторых моделях минимальный размер блока может достигать одного байта), в порядке, определенном пользователем. Для этого пользователю достаточно задать распределение памяти данных и памяти программ, в соответствии с которым процессор будет получать доступ и к содержимому ПЗУ в отлаживаемой системе, и к содержимому эмуляционной памяти внутрисхемного эмулятора. Такая память обычно называется памятью с возможностью мэппинга.

## 8.13. Подсистема точек останова.

Подсистема точек останова - набор средств, управляющий процессом выполнения программы. Он позволяет останавливать выполняемую в реальном (или приближенном к реальному) масштабе времени программу, при выполнении команды, размещенной по заданному адресу. Частный случай работы системы точек останова - пошаговое выполнение. Другие, часто используемые случаи, - останов при проведении операций ввода-вывода.

В том или ином виде данный модуль присутствует как в системах с эмуляцией или симуляцией микроконтроллера, так и в системах с реальным микроконтроллером. В последнем случае при достижении точки останова микроконтроллер останавливается или (и) переводится на выполнение специальной мониторинговой программы, при помощи которой можно зафиксировать или изменить состояние микроконтроллера перед последующим стартом.

Более развитый набор сервисных функций аналогичного назначения имеет процессор точек останова.

#### **8.14.Процессор точек останова.**

Процессор точек останова позволяет останавливать выполнение программы или выполнять иные действия, например, запускать или останавливать трассировщик при выполнении заданных пользователем условий. В отличие от механизма обычных точек останова, процессор точек останова позволяет формировать и отслеживать условия практически любой степени сложности, и при этом эмулируемый процесс не выводится из масштаба реального времени.

#### **8.15.Трассировщик.**

В сущности, трассировщик представляет собой логический анализатор, работающий синхронно с процессором и фиксирующий поток выполняемых инструкций и состояния выбранных внешних сигналов. Существуют модели внутрисхемных эмуляторов, которые позволяют трассировать не только внешние сигналы, но и состояния внутренних ресурсов микроконтроллера, например, регистров. Такие эмуляторы используют специальные версии микроконтроллеров (эмуляционные кристаллы).

#### **8.16.Профилировщик .**

Профилировщик (иначе анализатор эффективности программного кода) позволяет получить по результатам прогона отлаживаемой программы следующую информацию:

- Количество обращений к различным участкам программы;
- Время, затраченное на выполнение различных участков программы.

Анализ статистической информации, поставляемой профилировщиком, позволяет легко выявлять "мертвые" или перенапряженные участки программ, и в результате оптимизировать структуру отлаживаемой программы.

#### **8.17.Интегрированная среда разработки.**

Интегрированная среда разработки - это совокупность программных средств, поддерживающая все этапы разработки программного обеспечения от написания исходного текста программы до ее компиляции и отладки, и обеспечивающая простое и быстрое взаимодействие с другими инструментальными средствами (программным отладчиком-симулятором, внутрисхемным эмулятором, эмулятором ПЗУ и программатором).

Строго говоря, интегрированные среды разработки не относятся к числу средств отладки, тем не менее обойти вниманием данный класс программных средств, существенно облегчающий и ускоряющий процесс разработки и отладки микропроцессорных систем было бы неправильно.

При традиционном подходе, начальный этап написания программы строится следующим образом:

1. Исходный текст набирается при помощи какого-либо текстового редактора. По завершении набора, работа с текстовым редактором прекращается и запускается кросс компилятор. Как правило, вновь написанная программа содержит синтаксические ошибки, и компилятор сообщает о них на консоль оператора.
2. Вновь запускается текстовый редактор, и оператор должен найти и устранить выявленные ошибки, при этом сообщения о характере ошибок выведенные компилятором уже не видны, так как экран занят текстовым редактором.

И этот цикл может повторяться не один раз. Если программа имеет большой объем, собирается из различных частей, и подвергается длительному редактированию или модернизации, то даже этот начальный этап может потребовать много сил и времени. После этого наступает этап отладки программы и к редактору с компилятором добавляется эмулятор или симулятор, за работой которого хотелось бы следить прямо по тексту программы в текстовом редакторе.

Избежать большого объема однообразных действий и тем самым существенно повысить эффективность процесса разработки и отладки позволяют т.н. интегрированные среды (оболочки) разработки (Integrated Development Environment, IDE).

Работа в интегрированной среде дает программисту:

- Возможность использования встроенного многофайлового текстового редактора, специально ориентированного на работу с исходными текстами программ;
- Диагностика выявленных при компиляции ошибок, и исходный текст программы, доступный редактированию, выводятся одновременно в многооконном режиме;

- Возможность организации и ведения параллельной работы над несколькими проектами. Менеджер проектов позволяет использовать любой проект в качестве шаблона для вновь создаваемого проекта;
- Перекомпиляции подвергаются только редактировавшиеся модули;
- Возможность загрузки отлаживаемой программы в имеющиеся средства отладки, и работы с ними без выхода из оболочки;
- Возможность подключения к оболочке практически любых программных средств.

В последнее время, функции интегрированных сред разработки становятся стандартной принадлежностью программных интерфейсов эмуляторов и отладчиков-симуляторов.

Подобные функциональные возможности, в сочетании с дружественным интерфейсом, в состоянии существенно увеличить скорость разработки программ для микроконтроллеров и процессоров цифровой обработки сигналов.

## 9. Программные средства для микроконтроллеров семейства MCS-51.

- Дisasсемблеры для MCS-51
- Интегрированная отладочная среда ProView для микроконтроллеров семейства MCS-51, 251, XA
- Интегрированная отладочная среда mVision2
- Интегрированная отладочная среда для ADuC812
- Компиляторы с языка ассемблер для микроконтроллеров семейства MCS-51
- [Библиотека для работы с числами с плавающей запятой](#) (2,5 Kb)

### 9.1. Дisasсемблеры MCS-51.

Дisasсемблер предназначен для преобразования исполняемого кода микроконтроллеров MCS-51 в текст программы на языке ассемблера.

Форматы входных данных: HEX, OBJ, BIN.

Работа производится с помощью системы меню.

Поддерживаются процессоры:

- 8051/8031
- 8052/8032
- 80512/80532
- 80(C)515/80(C)535
- 80C517/80C537
- 80C515A/80C535A
- 80C517A/80C537A
- 8035/8048/8041
- z80

а также их модификации.

- [Дisasсемблер для MCS-51](#) (32 Kb)
- [Дisasсемблер для 8051/8052, z80, 8035/8048/8041 с исходниками](#) (266Kb)
- [Дisasсемблер INTEL 8051](#) (24Kb)

### 9.2. Введение в интегрированную отладочную среду ProView для микроконтроллеров семейства MCS-51, 251, XA

( [ProView V3.1](#) (13 681 Kb)

ProView фирмы Franklin Software Inc. – интегрированная среда разработки программного обеспечения для однокристальных микроконтроллеров семейства Intel 8051 и его клонов. Она включает в себя всё, что нужно для создания, редактирования, компиляции, трансляции, компоновки, загрузки и отладки программ:

1. стандартный интерфейс Windows,
2. полнофункциональный редактор исходных текстов с выделением синтаксических элементов цветом,
3. организатор проекта,
4. транслятор с языка C,
5. ассемблер,
6. отладчик,
7. встроенную справочную систему.

Среда разработки подобна Visual C++ Microsoft и Borland C++ для Windows. Пользователи, знакомые с любым из этих изделий, будут чувствовать себя в ProView, как дома.

Первый этап разработки программы – запись её исходного текста на каком-либо языке программирования.

Затем производится компиляция или трансляция его в коды из системы команд микроконтроллера, используя транслятор или ассемблер. Трансляторы и ассемблеры – прикладные программы, которые интерпретируют текстовый файл, содержащий исходный текст программы, и создают объектные файлы, содержащие объектный код.

После компоновки объектных модулей наступает этап отладки программы, устранения ошибок, оптимизации и тестирования программы.

ProView объединяет все этапы разработки прикладной программы в единый рекурсивный процесс, когда в любой момент времени возможен быстрый возврат к любому предыдущему этапу.

ProView имеет следующие компоненты.

### 9.2.1. Оптимизирующий кросс-компилятор C51.

Язык C - универсальный язык программирования, который обеспечивает эффективность кода, элементы структурного программирования и имеет богатый набор операторов. Универсальность, отсутствие ограничений реализации делают язык C удобным и эффективным средством программирования для широкого разнообразия задач. Множество прикладных программ может быть написано легче и эффективнее на языке C, чем на других более специализированных языках.

C51 - полная реализация стандарта ANSI (Американского национального института стандартов), насколько это возможно для архитектуры Intel 8051. C51 генерирует код для всего семейства микроконтроллеров Intel 8051. Транслятор сочетает гибкость программирования на языке C с эффективностью кода и быстродействием ассемблера.

Использование языка высокого уровня C имеет следующие преимущества над программированием на ассемблере:

8. глубокого знания системы команд процессора не требуется, элементарное знание архитектуры Intel 8051 желательно, но не необходимо;
9. распределение регистров и способы адресации управляются полностью транслятором;
10. лучшая читаемость программы, используются ключевые слова и функции, которые более свойственны человеческой мысли;
11. время разработки программ и их отладки значительно короче в сравнении с программированием на ассемблере;
12. библиотечные файлы содержат много стандартных подпрограмм, которые могут быть включены в прикладную программу;
13. существующие программы могут многократно использоваться в новых программах, используя модульные методы программирования.

### 9.2.2. Макроассемблер A51.

Ассемблер A51 совместим с ASM51 Intel для всего семейства микроконтроллеров Intel 8051. Ассемблер транслирует символическую мнемонику в перемещаемый объектный код, имеющий высокое быстродействие и малый размер. Макросредства ускоряют разработку и экономят время, поскольку общие последовательности могут быть разработаны только один раз. Ассемблер поддерживает символический доступ ко всем элементам микроконтроллера и перестраивает конфигурацию для каждой разновидности Intel 8051.

A51 транслирует исходный файл ассемблера в перемещаемый объектный модуль. При отладке или при включенной опции "Include debugging information" этот объектный файл будет содержать полную символическую информацию для отладчика/имитатора или внутрисхемного эмулятора.

### 9.2.3. Компоновщик L51.

Компоновщик объединяет один или несколько объектных модулей в одну исполняемую программу. Компоновщик размещает внешние и общие ссылки, назначает абсолютные адреса перемещаемым сегментам программ. Он может обрабатывать объектные модули, созданные транслятором C51, ассемблером A51, транслятором PL/M-51 Intel и ассемблером ASM51 Intel.

Компоновщик автоматически выбирает соответствующие библиотеки поддержки и связывает только требуемые модули из библиотек. Установки по умолчанию для L51 выбраны так, чтобы они подходили для большинства прикладных программ, но можно определить и заказные установки.

### 9.2.4. Отладчик/симулятор WinSim51.

Отладчик исходных текстов используется с транслятором C51, ассемблером A51, транслятором PL/M-51 Intel и ассемблером ASM51 Intel. Отладчик/симулятор позволяет моделировать большинство особенностей Intel 8051 без наличия аппаратных средств. Можно использовать его для проверки и отладки прикладной программы прежде, чем будут изготовлены аппаратные средства. При этом моделируется широкое разнообразие периферийных устройств, включая последовательный порт, внешний ввод - вывод и таймеры.

## 9.3. Быстрый старт.

"Быстрый старт" – это обычный приём разработчиков современных программных средств. Цель состоит в том, чтобы, не углубляясь пока в подробности, дать новичку или достаточно опытному пользователю первое представление о программном средстве, дать возможность быстро получить конкретный результат. Полное представление, знания и умения появятся позже в процессе работы и изучения справочных материалов.

В качестве примера возьмём простейшую программу, с которой начинают изучение языков программирования многие поколения студентов. "Hello World" - программа из папки \Fs\Examples\Hello\,



которая выдаёт в последовательный порт (UART) микроконтроллера строку символов “Hello World” (“Привет Мир”). Весь исходный текст программы содержится в файле hello.c:

```
/******  
/* YOUR FIRST 8051 PROGRAM */  
/******  
  
#include <reg51.h> /* special function register declarations */  
/* for the intended 8051 derivative */  
#include <stdio.h> /* prototype declarations for I/O functions*/  
/******  
  
/* main program */  
/******  
  
void main (void) { /* execution starts here after stack init */  
    SCON = 0x50; /* SCON: mode 1, 8-bit UART, enable rcvr */  
    TMOD |= 0x20; /* TMOD: timer 1, mode 2, 8-bit reload */  
    TH1 = 0xf3; /* TH1: reload value for 2400 baud */  
    TR1 = 1; /* TR1: timer 1 run */  
    TI = 1; /* TI: set TI to send first char of UART*/  
    printf ("Hello World\n"); /* the 'printf' function call */  
    while (1) { /* An embedded program does not stop and */  
        ; /* ... */ /* never returns. We've used an endless */  
    } /* loop. You may wish to put in your own */  
} /* code were we've printed the dots (...) */
```

Прежде чем начать разработку проекта, скопируйте папку \Fsi\Examples\Hello\ в свою личную папку. В этой папке находится всего лишь один файл hello.c.

### 9.3.1. Запуск ProView и создание файла проекта.

ProView запускается из стартового меню Windows подобно остальным приложениям. Если необходимо запустить программу из командной строки, её синтаксис имеет вид: PV32 [projectfile], где projectfile - имя файла проекта с расширением [.PRJ].

Любая новая работа в ProView, как и во всех современных компиляторах, начинается с создания нового файла проекта. Файл проекта содержит имена всех исходных файлов, связанных с проектом, а также установки компиляции, трансляции и связывания файлов, чтобы генерировать выполняемую программу.

Для того чтобы создать новый файл проекта, выберите New из меню Project. Откроется диалоговое окно New Project. Используйте кнопку Browse, чтобы войти в свою папку. Найдите папку \Hello и нажмите кнопку [OK]. Затем выберите “8051” как тип проекта.

Когда менеджер проекта открывает файл проекта, окно проекта показывает включенные исходные файлы. В данном случае пока нет никаких исходных файлов. Имеется только один исходный файл, который необходимо подключить - hello.c.

### 9.3.2. Добавка файла с исходным текстом и его редактирование.

Теперь можно добавить hello.c к проекту. Выберите Add file из меню Project. Откроется диалоговое окно Add File . Выберите hello.c из списка.

Наш проект имеет только один исходный файл. В дальнейшем Ваши проекты, возможно, будут состоять из множества исходных файлов. Диалог Add File позволит Вам выбрать и добавить несколько файлов сразу. Для этого используют комбинацию клавиши [CTRL] и указателя мыши. Когда Вы нажмёте [Open], исходные файлы будут добавлены к проекту в выбранном порядке.

Теперь можно редактировать текст из файла hello.c. Выберите hello.c из окна Project (рис. 19). Нажмите его правой кнопкой мыши и выберите View source file, или просто дважды щёлкните мышью для того, чтобы просматривать файл в окне редактирования.

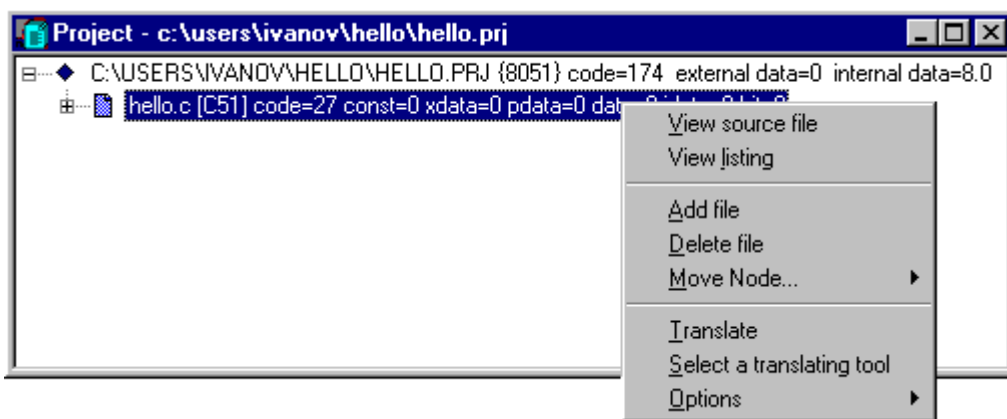


Рис. 19. Диалоговое окно Project

ProView загружает и показывает содержание hello.c в окне, где можно редактировать файл. Окно редактирования (рис. 20) - полнофункциональный редактор исходного текста, предлагающий такие возможности, как высвечивание синтаксических элементов и контекстный поиск. Если выбрать "printf" и нажать клавишу [F1], ProView откроет систему справки и перейдет к разделу справки о "printf".

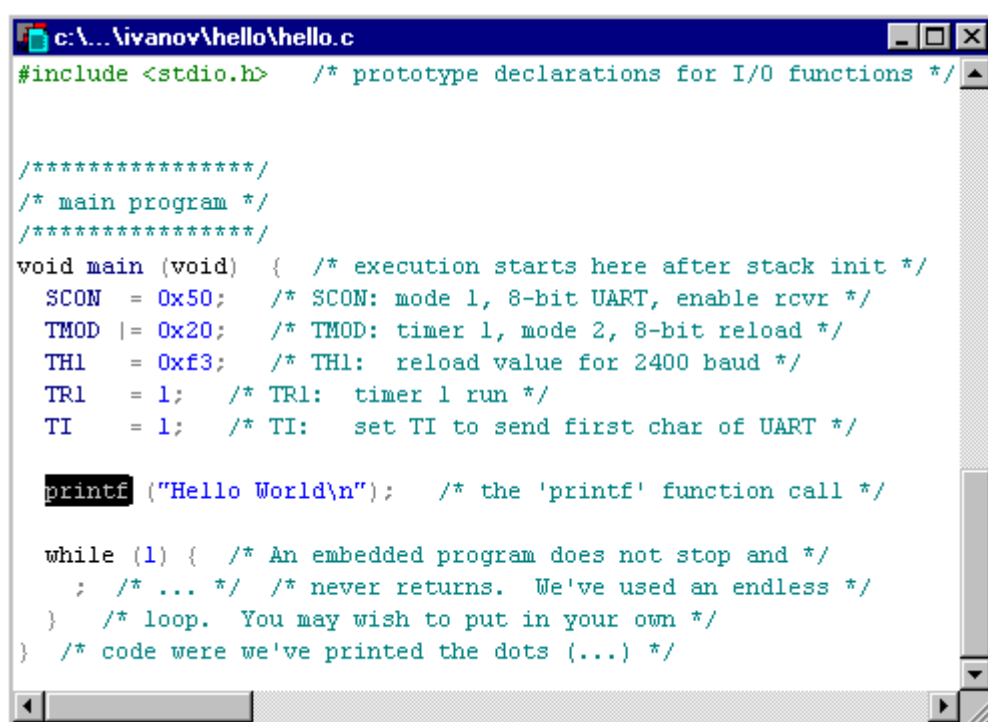


Рис. 20. Окно редактирования

### 9.3.3. Компиляция и компоновка.

Этот процесс компилирует, связывает hello.c с библиотеками и создает абсолютный объектный модуль, который мы сможем проверить в отладчике WinSim.

Выберите Make из меню Project. ProView отображает окно, показывая текущее состояние процесса. Когда процесс компиляции закончится, в окне Message (рис.21) отображается сообщение завершения. Если были обнаружены какие-нибудь ошибки, о них сообщается здесь же.

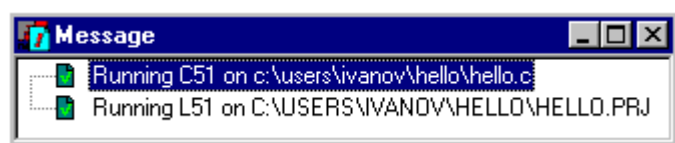
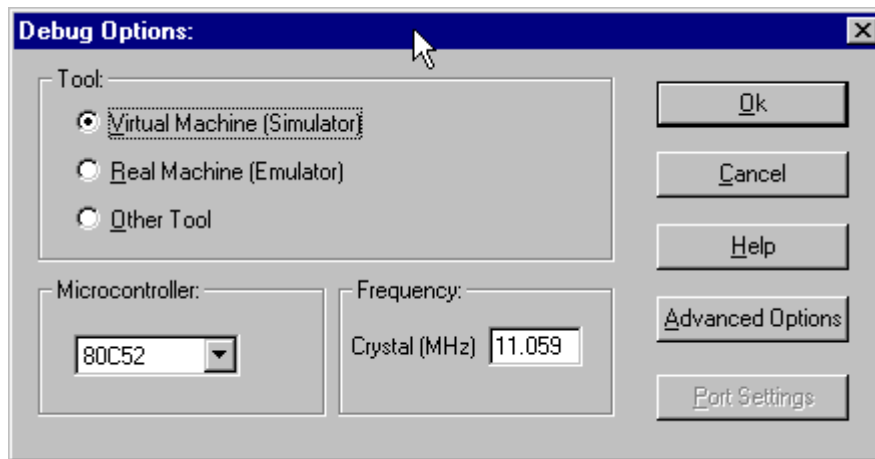


Рис. 21. Окно сообщений

### 9.3.4. Тестирование и отладка.

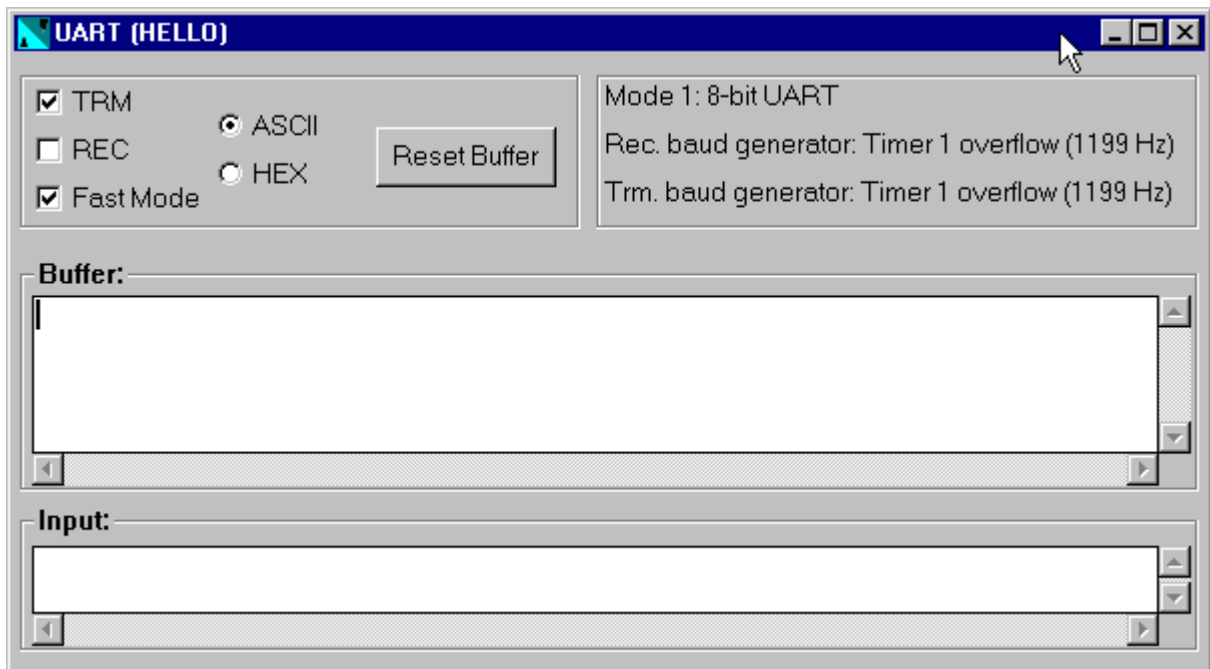
Выполним отладку программы. Если проект новый, откроется диалоговое окно Debug Options (рис. 22), где Вы можете изменять установки отладчика. В дальнейшем можно установить опции отладчика, выбрав Debug из меню Options. Наш проект использует значения по умолчанию.



**Рис. 22.** Окно диалога опций отладчика

Выберите Start из меню Debug.

Выберите Hardware (аппаратные средства) из меню View. Выберите UART, откроется окно последовательного порта (рис. 23). В дальнейшем при работе программы здесь можно будет увидеть всё, что выводит микроконтроллер в последовательный порт.



**Рис.23.** Окно последовательного порта

**GO**

Выберите Run из меню Debug или нажмите кнопку **Run**.

Рис. 24 показывает, как выглядит экран отладчика WinSim при выполнении программы. Обратите внимание, что в окне UART выведен текст "Hello World".

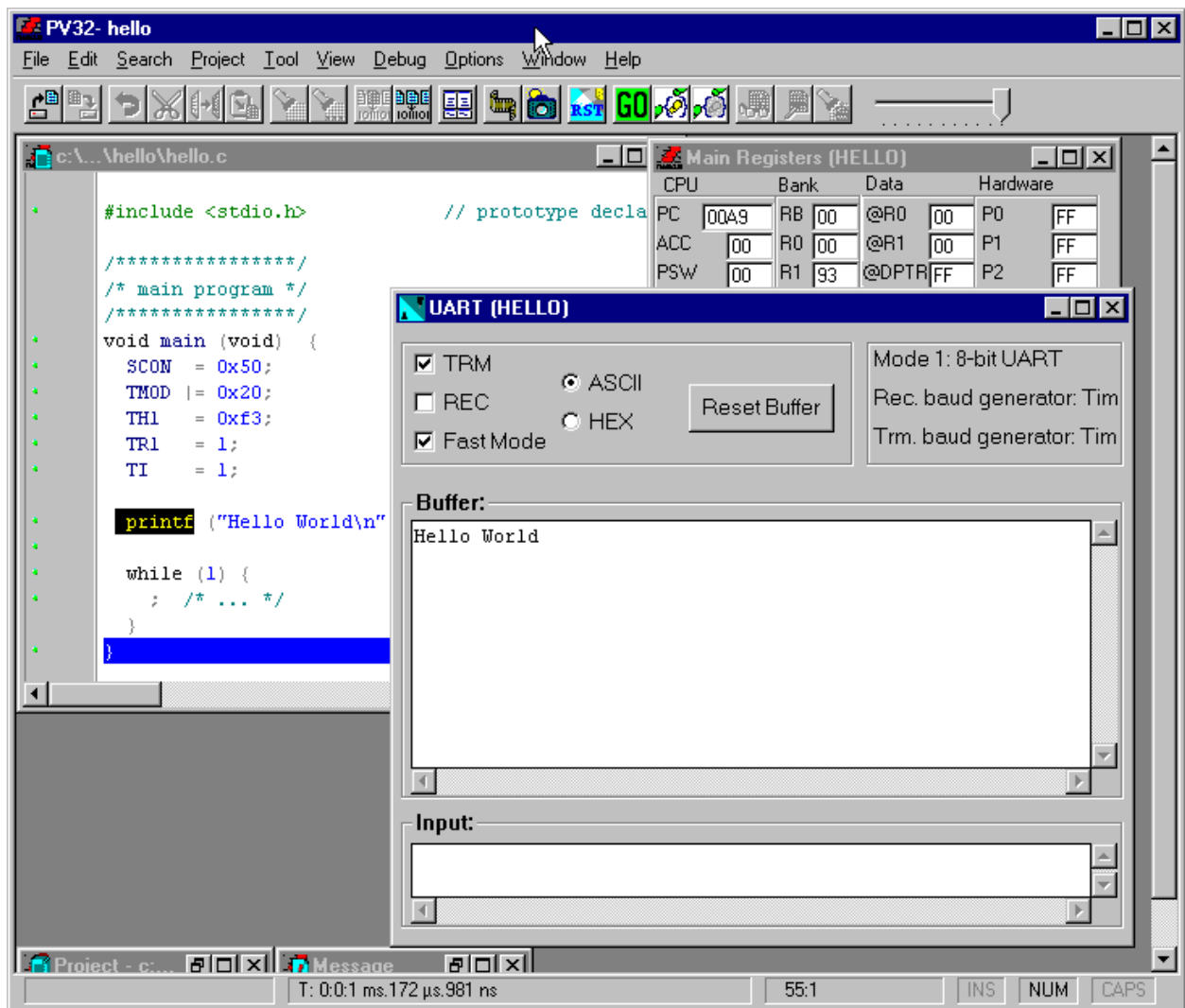




Рис. 24. Экран отладчика при выполнении программы

При выводе символов в порт начинается выполнение бесконечного цикла. Вы можете остановить выполнение программы, выбрав Stop из меню Debug. С помощью регулятора  при нажатой  кнопке

Animate на панели инструментов можно менять скорость работы отладчика. Строка состояния показывает текущее реальное время.

### 9.3.5. Пошаговый режим и выход из отладчика.

Вы можете использовать отладчик, чтобы перемещаться по программе. Выберите Reset из меню Debug (эта команда сбросит моделируемый процессор) и выберите Step Into и Step Over из меню Debug.

Команды Step позволяют “шагать” по каждой строке исходного текста. Текущая команда высвечивается на каждом шаге. Step Into позволяет войти в вызываемую функцию, Step Over – перешагнуть через неё, не входя во внутрь.

Для завершения работы с отладчиком в любой момент времени Вы можете выбрать Terminate из меню Debug и возвратиться в режим редактирования.

### 9.3.6. Следующий шаг.

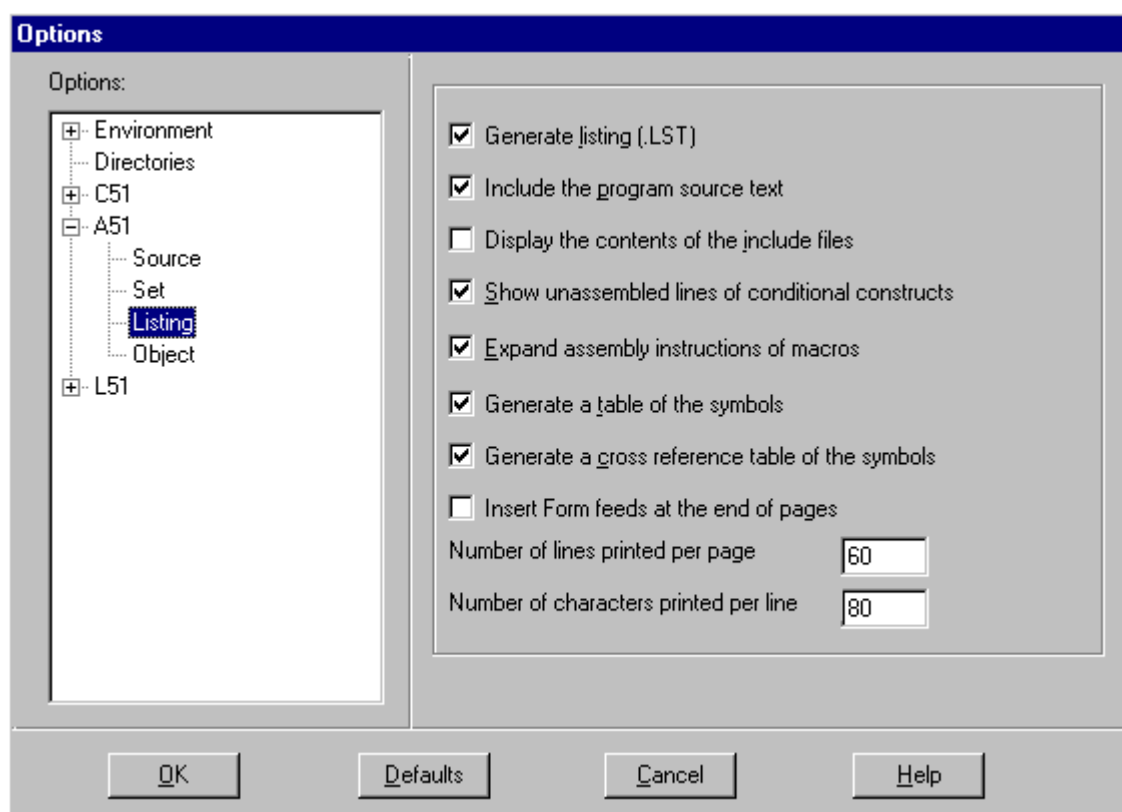
Обратите внимание, что в режиме отладки на экране видны ещё два окна. Первое – окно кода (рис. 25), где в пошаговом режиме параллельно с исходным текстом на языке C идёт трассировка текста на ассемблере.

Address	Symbol	Code	Mnemonic
0089:		6C	db 6C ; 'l'
008A:		6F	db 6F ; 'o'
008B:		20	db 20 ; ' '
008C:		57	db 57 ; 'W'
008D:		6F	db 6F ; 'o'
008E:		72	db 72 ; 'r'
008F:		6C	db 6C ; 'l'
0090:		64	db 64 ; 'd'
0091:		0A	db 0A
0092:		00	db 00
##_46	SCON = 0x50; /* SCON: mode 1, 8-bit UART, enable rcvr */		
0093:	main	759850	MOV SCON,#50
##_47	TMOD  = 0x20; /* TMOD: timer 1, mode 2, 8-bit reload */		
0096:		438920	ORL TMOD,#20
0099:		858989	MOV TMOD,TMOD
##_48	TH1 = 0xf3; /* TH1: reload value for 2400 baud */		
009C:		758DF3	MOV TH1,#F3
##_49	TR1 = 1; /* TR1: timer 1 run */		
009F:		D28E	SETB TR1
##_50	TI = 1; /* TI: set TI to send first char of UART */		
00A1:		D299	SETB TI
##_52	printf ("Hello World\n"); /* the 'printf' function call */		
00A3:		7B05	MOV R3,#05
00A5:		7A00	MOV R2,#00
00A7:		7986	MOV R1,#86
00A9:		120066	LCALL ?PRINTF0
##_57	} /* code were we've printed the dots [...] */		
00AC:		80FE	SJMP 00AC
00AE:		FF	db 0FF
00AF:		FF	db 0FF

**Рис. 25.** Окно кода

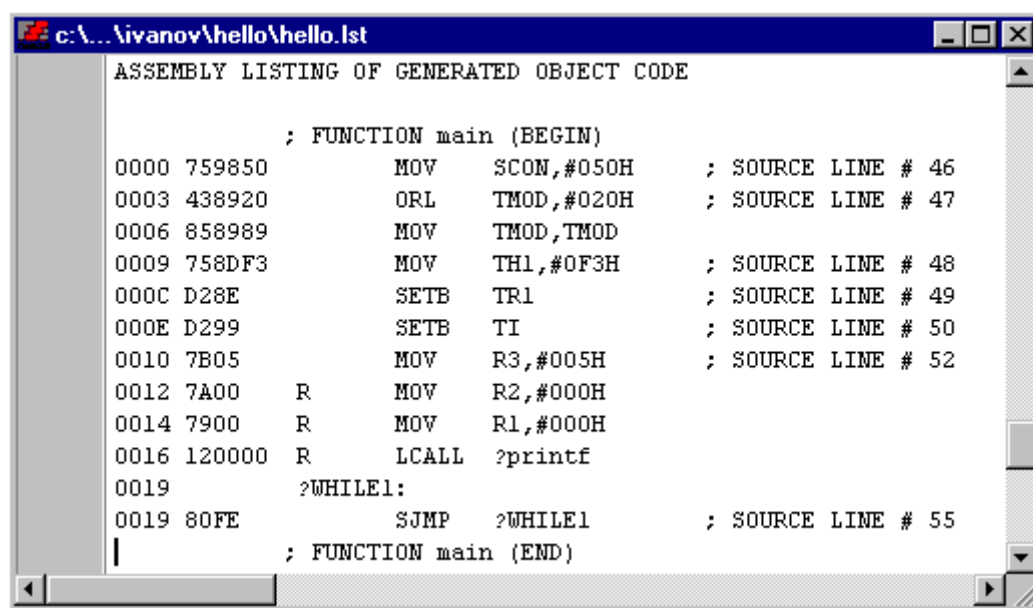
Прокрутите окно кода и изучите ассемблерный аналог исходного текста. С символов “##” начинаются строки, с помощью которых легко сопоставить ассемблерный текст и текст на языке C. Обратите внимание на то, сколько кода пришлось бы написать, если проектировать программу на ассемблере.

Ассемблерный аналог текста сохраняется в файле hello.lst, если в опциях проекта (Project из меню Options) отмечено Generate Listing (рис. 26). Здесь же можно указать, какую информацию включать в листинг.

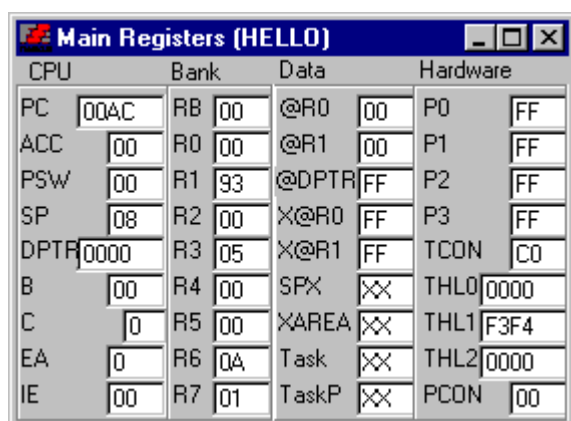


**Рис.26.** Диалог опций проекта

Изучите смысл других опций проекта в разделах Environment, C51, A51, L51. Откройте файл листинга (рис. 27) с помощью View listing из меню View.



**Рис. 27.** Окно файла листинга



Второе окно, которое присутствует на экране во время отладки, – Main Registers (рис. 28).

В этом окне постоянно отображается текущее состояние всех программно-доступных регистров микроконтроллера. Более того, содержимое регистров можно менять во время отладки.

С помощью пункта Data dump из меню View можно посмотреть содержимое памяти различного типа в режиме отладки. Попробуйте это сделать.

**Рис.28.** Окно регистров

## 9.4. Интегрированная отладочная среда mVision2.

**mVision2** - новая отладочная среда фирмы Keil Software для микроконтроллеров семейства MCS-51. Она включает средства управления проектами, мощный текстовый редактор и многофункциональный отладчик в удобной программной оболочке. В комплект входит подробное руководство, в котором есть справочная информация по всем вопросам и раздел для быстрого освоения программы.

Поддерживаются микроконтроллеры фирм: Analog Devices, AMD, Atmel, Dallas Semiconductor, Infineon, Intel, OKI, Philips, Temic, Winbond.

- [Интегрированная отладочная среда mVision2](#) (6842 Kb)

## 9.5..Компиляторы с языка ассемблер для микроконтроллеров семейства MCS-51.

Архитектура микроконтроллеров 8051 была разработана около 20 лет назад, и в настоящее время на рынке предлагается несколько сотен микроконтроллеров, которые выпускаются разными фирмами и совместимы с 8051. В данном разделе находятся компиляторы с ассемблера для таких микроконтроллеров.

Компилятор позволяет преобразовывать исходный текст программы на языке ассемблера в объектный код и код микроконтроллера для ПЗУ. Программы для разных модификаций микроконтроллеров следует писать с учетом особенностей конкретного кристалла и его периферийных модулей. Это можно выяснить по документации на микроконтроллер.

Мы рекомендуем использовать ассемблер 2500AD с документацией на русском языке, если нет специализированного компилятора для выбранного микроконтроллера.

- [Ассемблер 2500AD](#) (298 Kb)
- [Ассемблер для MCS-51](#) (63 Kb)
- [Ассемблер для MCS-51](#) (138 Kb)
- [Intel 8051 кросс-ассемблер](#) (46 Kb)
- [Ассемблер для 8031/8051 с исходниками](#) (54 Kb)
- [Кросс ассемблер для 8051](#) (32 Kb)

## 10. Основные характеристики микроконтроллеров MCS 51 / MCS 251.

Обозначения	Резидентная память программ		Резидентное ОЗУ	Таймеры/счетчик	Послед. порт	Аналог. входы	Число линий В/В	Тактовая частота	Тип корпуса	Секретность	Ключевые особенности
<i>Серия 8051 (Не рекомендуется к использованию в новых разработках)</i>											
8031AH	нет		128	2	1	0	32	12	D,N,P	нет	Процессор булевых функций
8051AH	4K	ROM	128	2	1	0	32	12	D,N,P	P	Процессор булевых функций
8751H	4K	EPROM	128	2	1	0	32	12	D	L1	Один уровень блокировки ЗУ
8751BH	4K	OTP ROM	128	2	1	0	32	12	N,P	L2	Два уровня блокировки ЗУ
<i>Серия 8052 (Не рекомендуется к использованию в новых разработках)</i>											
8032AH	нет		256	3	1	0	32	12	D,N,P	нет	Три таймера-счетчика
8052AH	8K	ROM	256	3	1	0	32	12	D,N,P	нет	Три таймера-счетчика
8752BH	8K	OTP/EPROM	256	3	1	0	32	12	D,N,P	L2	Два уровня блокировки ЗУ
<i>Серия 80C51</i>											
80C31BH	нет		128	2	1	0	32	12,16	D,N,P,S	нет	Режимы управления потреблением
80C51BH	4K	ROM	128	2	1	0	32	12,16	D,N,P,S	P	Режимы управления потреблением
87C51	4K	OTP/EPROM	128	2	1	0	32	12,16,20,24i	D,N,P,S	L3	Три уровня блокировки ЗУ
<i>Серия 8X52 /54 /58</i>											
80C32	нет		256	3	1	0	32	12,16,20,24i	N,P,S	нет	Реверсивный таймер-счетчик
80C52	8K	ROM	256	3	1	0	32	12,16,20,24i	N,P,S	L1	Реверсивный таймер-счетчик
87C52	8K	OTP/EPROM	256	3	1	0	32	12,16,20,24i	D,N,P,S	L3	Реверсивный таймер-счетчик
80C54	нет		256	3	1	0	32	12,16,20,24i	N,P,S	L1	Реверсивный таймер-счетчик
87C54	16K	OTP/EPROM	256	3	1	0	32	12,16,20,24i	D,N,P,S	L3	Реверсивный таймер-счетчик
80C58	нет		256	3	1	0	32	12,16,20,24i	N,P,S	L1	Реверсивный таймер-счетчик
87C58	32K	OTP/EPROM	256	3	1	0	32	12,16,20,24i	D,N,P,S	L3	Реверсивный таймер-счетчик
<i>Серия 8XL52 /54 /58</i>											
80L52	8K	ROM	256	3	1	0	32	12,16,20*	N,S	L1	V <sub>CC</sub> от 2.7В до 3.6В
87L52	8K	OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V <sub>CC</sub> от 2.7В до 3.6В
80L54	16K	ROM	256	3	1	0	32	12,16,20*	N,S	L1	V <sub>CC</sub> от 2.7В до 3.6В
87L54	16K	OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V <sub>CC</sub> от 2.7В до 3.6В
80L58	32K	ROM	256	3	1	0	32	12,16,20*	N,S	L1	V <sub>CC</sub> от 2.7В до 3.6В
87L58	32K	OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V <sub>CC</sub> от 2.7В до 3.6В
<i>Серия 8XC51FA /FB /FC</i>											
80C51FA	нет		256	3	1	0	32	12,16	N,P,S	нет	Матрица программируемых счетчиков (PCA)
83C51FA	8K	ROM	256	3	1	0	32	12,16	N,P,S	L1	Матрица программируемых счетчиков (PCA)
87C51FA	8K	OTP/EPROM*	256	3	1	0	32	12,16,20,24i	D,N,P,S	L3	Матрица программируемых счетчиков (PCA)
83C51FB	16K	ROM	256	3	1	0	32	12,16,20,24i	N,P,S	L1	Матрица программируемых счетчиков (PCA)
87C51FB	16K	OTP/EPROM*	256	3	1	0	32	12,16,20,24i	D,N,P,S	L3	Матрица программируемых счетчиков (PCA)
83C51FC	32K	ROM	256	3	1	0	32	12,16,20,24i	N,P,S	L1	Матрица программируемых счетчиков (PCA)
87C51FC	32K	OTP/EPROM*	256	3	1	0	32	12,16,20,24i	D,N,P,S	L3	Матрица программируемых счетчиков (PCA)
<i>Серия 8XL51FA /FB /FC</i>											
80L51FA	нет		256	3	1	0	32	12,16,20*	N,S	нет	V <sub>CC</sub> от 2.7В до 3.6В, PCA
83L51FA	8K	ROM	256	3	1	0	32	12,16,20*	N,S	L1	V <sub>CC</sub> от 2.7В до 3.6В, PCA
87L51FA	8K	OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V <sub>CC</sub> от 2.7В до 3.6В, PCA
83L51FB	16K	ROM	256	3	1	0	32	12,16,20*	N,S	L1	V <sub>CC</sub> от 2.7В до 3.6В, PCA
87L51FB	16K	OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V <sub>CC</sub> от 2.7В до 3.6В, PCA
83L51FC	32K	ROM	256	3	1	0	32	12,16,20*	N,S	L1	V <sub>CC</sub> от 2.7В до 3.6В, PCA



Обозначения	Резидентная память программ	Резидентное ОЗУ	Таймеры/счет	Послед. порт	Аналог.	Число линий	Тактовая частота	Тип корпуса	Секретность	Ключевые особенности
87L51FC	32K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V <sub>CC</sub> от 2.7В до 3.6В, PCA
<i>Серия 8XC51RA /RB /RC</i>										
80C51RA	нет	512	3	1	0	32	12,16,20,24i	N,P	нет	Расширенное ОЗУ, PCA, аппаратный WDT
83C51RA	8K ROM	512	3	1	0	32	12,16,20,24i	N,P	L1	Расширенное ОЗУ, PCA, аппаратный WDT
87C51RA	8K OTP ROM	512	3	1	0	32	12,16,20,24i	N,P	L3	Расширенное ОЗУ, PCA, аппаратный WDT
83C51RB	16K ROM	512	3	1	0	32	12,16,20,24i	N,P	L1	Расширенное ОЗУ, PCA, аппаратный WDT
87C51RB	16K OTP ROM	512	3	1	0	32	12,16,20,24i	N,P	L3	Расширенное ОЗУ, PCA, аппаратный WDT
83C51RC	32K ROM	512	3	1	0	32	12,16,20,24i	N,P	L1	Расширенное ОЗУ, PCA, аппаратный WDT
87C51RC	32K OTP ROM	512	3	1	0	32	12,16,20,24i	N,P	L3	Расширенное ОЗУ, PCA, аппаратный WDT
<i>Серия 8XC51GB</i>										
80C51GB	нет	256	3	1+ SE P	8	48	12,16	N1	нет	8-разрядный АЦП, 2 PCA, 6 портов В/В
83C51GB	8K ROM	256	3	1+ SE P	8	48	12,16	N1	L1	8-разрядный АЦП, 2 PCA, 6 портов В/В
87C51GB	8K OTP ROM	256	3	1+ SE P	8	48	12,16	N1	L3	8-разрядный АЦП, 2 PCA, 6 портов В/В
<i>Серия 8XC152 — Коммуникационные контроллеры</i>										
80C152JA	нет	256	2	1	0	40	12,16	P1,N1	нет	Многопротокольный послед. канал, 2 DMA
80C152JB	нет	256	2	1	0	56	12,16	N1	нет	Многопротокольный послед. канал, 2 DMA
83C152JA	8K ROM	256	2	1	0	40	12,16	P1,N1	нет	Многопротокольный послед. канал, 2 DMA
<i>Серия 8XC51SL — Контроллеры клавиатуры</i>										
80C51SL-AH	нет	256	2	1	4	24	16	KU	нет	Управление клавиатурой потребителя
81C51SL-AH	16K ROM1	256	2	1	4	24	16	KU	нет	Управление клавиатурой потребителя
83C51SL-AH	16K ROM	256	2	1	4	24	16	KU	нет	Управление клавиатурой потребителя
87C51SL-AH	16K OTP	256	2	1	4	24	16	KU	нет	Управление клавиатурой потребителя
80C51SL-AL	нет	256	2	1	4	24	16	SB	нет	V <sub>CC</sub> от 3В до 3.6В, 8-разрядный АЦП
81C51SL-AL	16K ROM1	256	2	1	4	24	16	SB	нет	V <sub>CC</sub> от 3В до 3.6В, 8-разрядный АЦП
83C51SL-AL	16K ROM	256	2	1	4	24	16	SB	нет	V <sub>CC</sub> от 3В до 3.6В, 8-разрядный АЦП
87C51SL-AL	16K OTP	256	2	1	4	24	16	SB	нет	V <sub>CC</sub> от 3В до 3.6В, 8-разрядный АЦП
<i>MCS251 — Новая архитектура</i>										
80C251S B	нет	1K	3	1	0	32	12,16	N	нет	Расшир. архитектура, PCA, аппаратный WDT
83C251S B	16K ROM	1K	3	1	0	32	12,16	N	L1	Расшир. архитектура, PCA, аппаратный WDT
87C251S B	16K OTP ROM	1K	3	1	0	32	12,16	N	L3	Расшир. архитектура, PCA, аппаратный WDT

## 11. Микроконтроллеры семейства MCS51 и его аналоги.

Type	Manufacturer	Max. Clock Rate	ROM (Byte)	RAM / XRAM (Byte)	Max. I/O Lines	Timer (16-Bit)	Serial I/O	ADC-In/Res	Others	PDF-Size (Data Sheet)
AT-89C51	Atmel	24	4k Flash	128	32	2	1	-	-	<a href="#">122k</a>
AT-89C52	Atmel	24	8k Flash	256	32	3	1	-	-	<a href="#">173k</a>
AT-89C1051	Atmel	24	1k Flash	64	15	2	1	-	Analog Comp.	<a href="#">219k</a>
AT-89C2051	Atmel	24	2k Flash	128	15	2	1	-	Analog Comp.	<a href="#">246k</a>
DS-80C310	Dallas	33	-	256	32	3	1	-	2 Data Pointer	<a href="#">327k</a>
DS-80C320	Dallas	33	-	256	32	3	2	-	Low Power, WDT	<a href="#">397k</a>
DS-80C323	Dallas	18	-	256	32	3	2	-	Low Power, WDT	<a href="#">397k</a>
DS-87C520	Dallas	33	16k	256 / 1k	32	3	2	-	WDT	<a href="#">470k</a>
DS-87C530	Dallas	33	16k	256 / 1k	32	3	2	-	RTC, WDT	<a href="#">343k</a>
I-8XC51	Intel	24	4k	128	32	2	1	-	-	<a href="#">288k</a>
I-8XC51FX	Intel	20	8k-32k	256	32	2	1	-	PCA	<a href="#">284k</a>
SC-8XC51	Philips	33	4k	128	32	2	1	-	-	<a href="#">411k</a>
P-8XC52	Philips	33	8k	256	32	3	1	-	-	<a href="#">380k</a>
SC-8XC451	Philips	16	4k	128	56	2	1	-	Parallel Interface	<a href="#">315k</a>
P-8XC550	Philips	16	4k	128	24 +8in	2	1	8 / 8	WDT	<a href="#">292k</a>
PCB-8XC552	Philips	24	8k	256	40 +8in	2	2	8 / 10	I2C, WDT, PWM	<a href="#">403k</a>
PCB-8XC562	Philips	16	8k	256	40 +8in	3	1	8 / 8	WDT, PWM, CAPCOM	<a href="#">247k</a>
P-8XC575	Philips	16	8k	256	32	3	1	-	WDT, PCA, 4Analog Comp.	<a href="#">508k</a>
P-8XC576	Philips	16	8k	256	30	3	1	6 / 10	WDT, PWM, 4Analog Comp.	<a href="#">615k</a>
SAB-80C5X5	Siemens	20	8k	256	40 +8in	3	1	8 / 8	WDT, PWM, WDU	
SAB-80C515A	Siemens	18	-	256 / 1k	40 +8in	3	1	8 / 10	WDT, PWM, WDU	
SAB-80C5X7	Siemens	16	8k	256	56 +12in	4	2	12 / 8	WDT, PWM, WDU	
SAB-80C517A	Siemens	18	-	256 / 1k	56 +12in	4	2	12 / 10	WDT, WDU, PWM	
SAB-C501	Siemens	40	8k	256	32	3	1	-	-	
SAB-C502	Siemens	20	16k	256 / 256	32	3	1	-	progr. XRAM-Start Adr, WDU	
SAB-C504	Siemens	40	16k	256 / 256	32	3	1	8 / 10	PWM for motor, WDU	
SAB-C509	Siemens	16	-	256 / 3k	64 +15in	5	2	15 / 10	WDT, PWM, WDU	
SAB-C511	Siemens	12	2,5k	128	32	2	1 SSC	-	-	
SAB-C511A	Siemens	12	4k	256	32	2	1 SSC	-	-	
SAB-C513	Siemens	12	8k	256	32	3	2	-	SSC	
SAB-C513A	Siemens	12	16k	256 / 256	32	3	2	-	SSC, EEPROM available	
SAB-C515C	Siemens	10	ROMless / 64k	256 / 1k	49 +8in	3	2	8 / 10	Full CAN Interface, PWM, SSC	

## 12.Список литературы.

1. Справочник. Микроконтроллеры : архитектура, программирование, интерфейс. Бродин В.Б., Шагурин М.И.М.:ЭКОМ, 1999.
  2. [Документация на микроконтроллеры фирмы Intel MCS- 51/151/251.](http://www.intel.com/design/mcs51/docs_mcs51.htm) (каталог pdf файлов)  
[http://www.intel.com/design/mcs51/docs\\_mcs51.htm](http://www.intel.com/design/mcs51/docs_mcs51.htm)
  3. [Описания микроконтроллеров семейства MCS-51](http://www.phyton.ru/cgi-bin/control/noframe.cgi?PAR1=3&PAR2=4&PAR3)  
<http://www.phyton.ru/cgi-bin/control/noframe.cgi?PAR1=3&PAR2=4&PAR3>
  4. [Микроконтроллеры:](http://me.tusur.ru/~radio/micro/)(подборка документации и софта)  
<http://me.tusur.ru/~radio/micro/>
  5. [Однокристалльные микроЭВМ MCS-96](http://ce.cctpu.edu.ru/msclub/LITERATU/gusev/oglav.htm) А.В. Гусев, О.В. Мироненко, Уральский политехнический университет, кафедра ТПРА (Екатеринбург). В книге дается подробный обзор архитектуры, системы команд и применений 16-разрядных микроконтроллеров семейства Intel MCS-96. (учебник в pdf файлах)  
<http://ce.cctpu.edu.ru/msclub/LITERATU/gusev/oglav.htm>
  6. [Документация на микроконтроллеры фирмы Intel MCS- 96/296.](http://ce.cctpu.edu.ru/msclub/LITERATU/gusev/oglav.htm) (каталог pdf файлов)  
<http://ce.cctpu.edu.ru/msclub/LITERATU/gusev/oglav.htm>
- А так же много информации можно найти на сайтах : <http://www.phyton.ru> и <http://www.gaw.ru/>