

Министерство общего и профессионального образования Российской Федерации

Новосибирский Государственный Технический Университет

**В.Н. Веприк, В.А. Афанасьев, А.И. Дружинин,
А.А. Земсков, А.Р. Исаев, О.В. Малявко**

МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS-51

Учебное пособие по курсам
"Микропроцессорные системы"
и "Проектирование микропроцессорных систем"
для студентов старших курсов
факультета автоматики и вычислительной техники
всех форм обучения

Новосибирск
1997

В.Н. Веприк, В.А. Афанасьев, А.И. Дружинин, А.А. Земсков, А.Р. Исаев, О.В. Малявко.
Микроконтроллеры семейства MCS-51: Учебное пособие. - Новосибирск.

Предлагаемое Вашему вниманию учебное пособие содержит общее описание архитектуры, функциональных возможностей и системы команд семейства однокристальных микроконтроллеров (Embedded Microcontrollers) MCS-51, производимых фирмой INTEL. Во второй части пособия приводится описание учебного микропроцессорного контроллера УМПК-51, предлагаемого студентам в качестве объекта исследования при выполнении цикла лабораторных работ.

Материал пособия может использоваться при курсовом и дипломном проектировании, а также может быть полезным для инженеров - схемотехников, занимающихся разработкой и эксплуатацией электронной аппаратуры.

Авторы выражают глубокую признательность АОЗТ "Новые технологии"- официальному дистрибьютору фирмы INTEL за предоставление материалов, на основе которых выполнена данная работа.

Табл.15, ил.25, список лит. 12 назв.

Рецензенты: Е.Д. Баран, Г.Г. Матушкин.

Работа подготовлена на кафедре
вычислительной техники



Новосибирский государственный
технический университет

ВВЕДЕНИЕ.

Семейство 8-разрядных однокристальных микроконтроллеров MCS-51 появилось на мировом рынке в начале восьмидесятых годов. Первые модификации кристаллов (около 7) были выполнены по высококачественной n-МОП (NMOS) технологии и являлись функционально завершенными однокристальными микроЭВМ гарвардской архитектуры, один из основных принципов которой состоит в логическом разделении адресных пространств памяти программ и данных. С развитием полупроводниковой технологии последующие версии микросхем MCS-51 стали изготавливать по более совершенной и низкочастотной КМОП (CMOS) технологии (в активном режиме потребление кристаллов было доведено до 10 ± 50 мА).

Система команд MCS-51, ориентированная на реализацию различных цифровых алгоритмов управления, при сохранении некоторой внешней схожести с системой команд предыдущего семейства MCS-48, качественно расширилась, в ней появились принципиальные нововведения:

- битово-ориентированные операции и адресуемые в памяти данных битовые поля, что дало возможность говорить о реализации на кристалле битового процессора;
- реализовано исполнение команд умножения, деления и вычитания;
- усовершенствована работа со стеком;
- расширена группа команд передачи управления;

Система команд стала выглядеть более симметричной, то есть менее зависимой от пересылок данных через аккумулятор.

Функциональные возможности встроенных периферийных устройств также расширились за счет введения:

- двух 16-разрядных таймеров-счетчиков;
- аппаратного последовательного дуплексного порта;
- двухуровневой системы прерываний;
- четырех 8-битовых портов ввода-вывода.

Принципиальные изменения в структуре временного цикла работы процессора привели к ускорению работы с внешней памятью программ и данных, а также реакций на внешние и внутренние прерывания. Суммарный размер адресного пространства внешней памяти программ и данных увеличился до 128 Кбайт. 16-разрядные регистры счетчика команд (Program Counter) и указателя данных (Data Pointer) позволили напрямую обращаться ко всему диапазону адресов, что дало разработчикам возможность реализации алгоритмов быстрой обработки крупных массивов данных.

Все программно-доступные узлы микроконтроллера были сведены в специальную область памяти данных (Special Function Register), что позволило обращаться к ним почти так же, как и к обычным ячейкам резидентного ОЗУ.

В более поздних модификациях кристаллов усовершенствование шло по пути наращивания дополнительных функциональных возможностей с сохранением полной программной совместимости с более ранними версиями. Особенности последних модификаций микроконтроллеров семейства MCS-51 являются:

- полностью статический дизайн;
- 3- и 5-вольтовые версии кристаллов;
- широкий спектр встроенных периферийных устройств;
- максимальная тактовая частота - 24 мГц; для отдельных групп кристаллов - 33 мГц.

В настоящее время в состав MCS-51 входит около 60 версий кристаллов, кроме того, имеется и доступна подробная фирменная документация (к сожалению, пока мало переведенная на русский язык). Для подготовки математического обеспечения микроконтроллеров MCS-51 используются в основном языки "ASM-51", "C", для которых существуют ряд достаточно хорошо зарекомендовавших себя компиляторов, библиотек стандартных подпрограмм и программных эмуляторов, производимых различными зарубежными и отечественными фирмами.

Несмотря на достаточную "древность" семейства (более 15 лет) и появление на мировом рынке за последние годы однокристалльных микроконтроллеров большей производительности и усовершенствованной архитектуры - MCS-51, MCS-251, MCS-96, контроллеры MCS-51 еще достаточно долго будут широко использоваться в сравнительно простых встроенных системах управления [1,4,6,10,11].

1. СИСТЕМА ОБОЗНАЧЕНИЙ КОМПОНЕНТОВ ФИРМЫ INTEL И ФУНКЦИОНАЛЬНЫЙ СОСТАВ СЕМЕЙСТВА MCS-51

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 \oplus C)
- M ⌚ (Military), исполнение по военным стандартам (-55/+125 \oplus C)
- Q или C ⌚ (Commercial), "коммерческий" температурный диапазон (0/+70 \oplus C) с (160 \oplus 8)-часовой динамической термотренировкой;
- L или E ⌚ (Extended), "расширенный" температурный диапазон (-40/+85 \oplus C) с (160 \oplus 8)-часовой динамической термотренировкой;
- T ⌚ (Extended), "расширенный" температурный диапазон (-40/+85 \oplus 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 [6,12].

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.

1. **Группа 8X5X** (8051 Product Line и 8052 Product Line):
8031АН, 8051АН, 8751Н, 8051АНР, 8751Н-8, 8751ВН, 8032АН, 8052АН, 8752ВН.
2. **Группа 8XC51** (80C51 Product Line):
80C31ВН, 80C51ВН, 87C51.
3. **Группа 8XC5X** (8XC52/54/58 Product Line):
80C32, 80C52, 87C52, 80C54, 87C54, 80C58, 87C58.
4. **Группа 8XC51FX** (8XC51FA/FB/FC Product Line):
80C51FA, 83C51FA, 87C51FA, 83C51FB, 87C51FB, 83C51FC, 87C51FC.
5. **Группа 8XL5X** (8XL52/54/58 Product Line):
80L52, 87L52, 80L54, 87L54, 80L58, 87L58.
6. **Группа 8XL51FX** (8XL51FA/FB/FC Product Line):
80L51FA, 83L51FA, 87L51FA, 83L51FB, 87L51FB, 83L51FC, 87L51FC.
7. **Группа 8XC51RX** (8XC51RA/RB/RC Product Line):
80C51RA, 83C51RA, 87C51RA, 83C51RB, 87C51RB, 83C51RC, 87C51RC.
8. **Группа 8XC51GB** (8XC51GX Product Line):
80C51GB, 83C51GB, 87C51GB.
9. **Группа 8XC152JX** (8XC152 Product Line):
80C152JA, 83C152JA, 80C152JB, 80C152JC, 83C152JC, 80C152JD.
10. **Группа 8XC51SL** (8XC51SL Product Line):
80C51SL-BG, 81C51SL-BG, 83C51SL-BG, 80C51-АН, 81C51SL-АН, 83C51SL-АН, 87C51SL-АН, 80C51SL-AL, 81C51SL-AL, 83C51SL-AL, 87C51SL-AL.

Первая группа микроконтроллеров включает в себя младшие модели семейства, выполненные по n-МОП технологии и не рекомендуемые к использованию в новых разработках, все остальные группы выполнены по современной КМОП технологии.

Микросхемы второй, третьей и четвертой групп являются на сегодняшний день классическими представителями семейства MCS-51. В пятую и шестую группы входят 3-вольтовые версии кристаллов (Low-Voltage). Кристаллы седьмой группы оснащены расширенным ОЗУ (Expanded RAM), объем которого равен 512 байт. Микросхемы восьмой, девятой и десятой групп представляют собой специализированные по применению микроконтроллеры (Application Specific). Многие современные приложения требуют высокопроизводительных управляющих микроконтроллеров, использующих расширенные возможности адресации, регистровую архитектуру, большой объем внутреннего ОЗУ и стека, а также эффективно поддерживающих программирование на языке высокого уровня. К таким микроконтроллерам относятся микроконтроллеры новой архитектуры (New Architecture) семейств MCS-5 и MCS-251, к производству которых компания Intel приступила в 1995 году. Функциональный состав и ключевые особенности микроконтроллера MCS-51/MCS-251 приведены в приложении.

2. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ БАЗОВОЙ МОДЕЛИ СЕМЕЙСТВА МИКРОКОНТРОЛЛЕРОВ MCS-51.

Базовой моделью семейства микроконтроллеров MCS-51 и основой для всех последующих модификаций является микроконтроллер I-8051. Его основные характеристики следующие:

- восьмиразрядный ЦП, оптимизированный для реализации функций управления;
- встроенный тактовый генератор;
- адресное пространство памяти программ - 64 К;
- адресное пространство памяти данных - 64 К;
- внутренняя память программ - 4 К;
- внутренняя память данных - 128 байт;
- дополнительные возможности по выполнению операций булевой алгебры (побитовые операции);
- 32 двунаправленные и индивидуально адресуемые линии ввода/вывода;
- 2 шестнадцатиразрядных многофункциональных таймера/счетчика;
- полнодуплексный асинхронный приемопередатчик;
- векторная система прерываний с двумя уровнями приоритета и шестью источниками событий [2-4,7,8].
- Структурная схема I-8051 показана на рис.1, назначение выводов микросхемы - на рис.2.

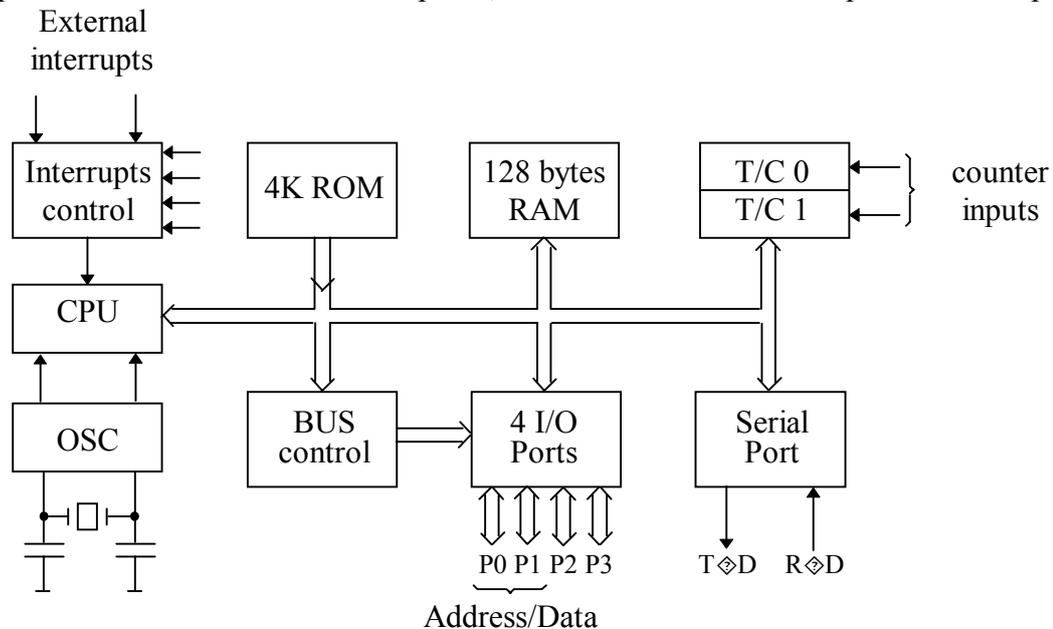


Рис.1. Структурная схема I-8051

Вся серия MCS-51 имеет гарвардскую архитектуру, то есть отдельные адресные пространства памяти программ и данных. Организация памяти изображена на рис.3.

Объем внутренней (резидентной) памяти программ (ROM, EPROM или OTP ROM), располагаемой на кристалле, в зависимости от типа микросхемы может составлять 0 (ROMless), 4К (базовый кристалл), 8К, 16К или 32К. При необходимости пользователь может расширять память программ установкой внешнего ПЗУ. Доступ к внутреннему или внешнему ПЗУ определяется значением сигнала на выводе EA (External Access):

EA= V_{cc} (напряжение питания) - доступ к внутреннему ПЗУ;

EA= V_{ss} (потенциал земли) - доступ к внешнему ПЗУ.

Для кристаллов без ПЗУ (ROMless) вывод EA должен быть постоянно подключен к V_{ss} .

Строб чтения внешнего ПЗУ - \overline{PSEN} (Program Store Enable) генерируется при обращении к внешней памяти программ и является неактивным во время обращения к ПЗУ, расположенному на кристалле.

Область нижних адресов памяти программ используется системой прерываний, архитектура микросхемы 8051 обеспечивает поддержку пяти источников прерываний:

- двух внешних прерываний;
- двух прерываний от таймеров;
- прерывания от последовательного порта.

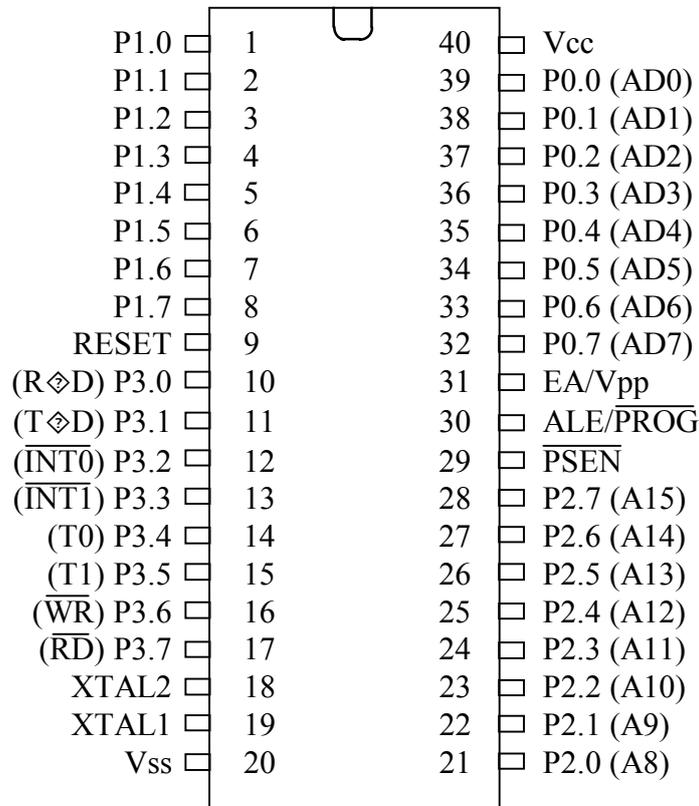


Рис.2. Назначение выводов I-8051

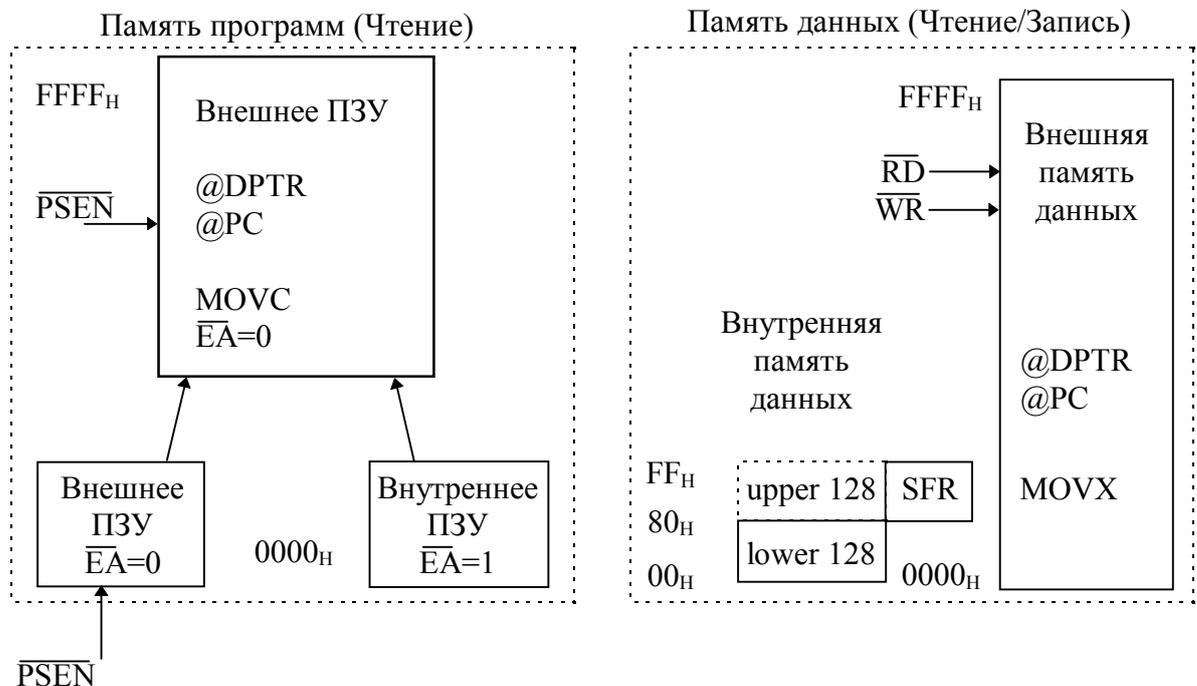


Рис.3. Организация памяти семейства MCS-51

На рис.4 изображена карта нижней области программной памяти.

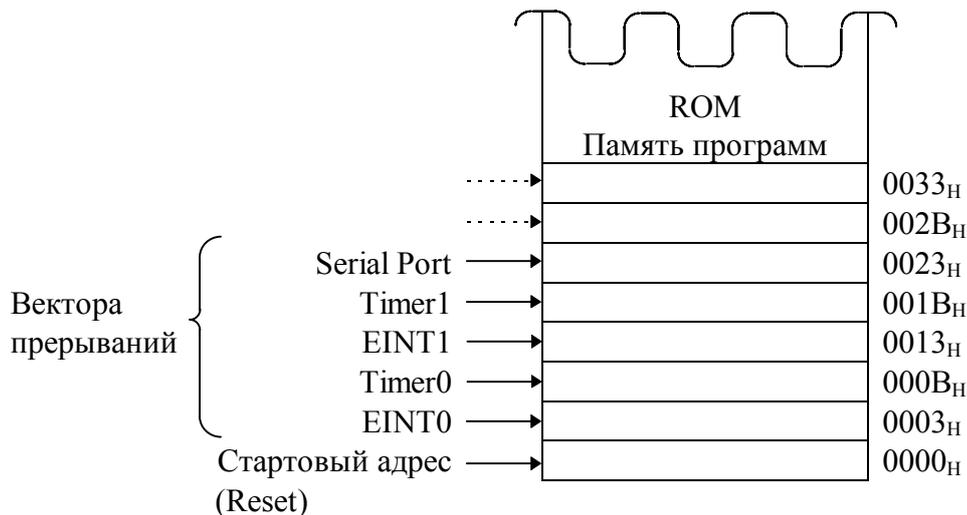


Рис.4. Программная память

Адреса векторов прерываний расположены с интервалом в 8 байт:

- 0003_H ⌚ внешнее прерывание 0 (External Interrupt 0) - вывод $\overline{INT0}$;
- 000B_H ⌚ прерывание от таймера 0 (по флагу переполнения таймера - $TF0$);
- 0013_H ⌚ внешнее прерывание 1 (External Interrupt 1) - вывод $\overline{INT1}$;
- 001B_H ⌚ прерывание от таймера 1 (по флагу переполнения таймера - $TF1$);
- 0023_H ⌚ прерывание от последовательного порта (Serial Interrupt = Receive Interrupt or Transmit Interrupt);

и так далее.

Память данных отделена от памяти программ. В этой области возможна адресация 64К внешнего ОЗУ. При обращении к внешней памяти данных ЦП микроконтроллера генерирует соответствующие сигналы чтения (\overline{RD}) или записи (\overline{WR}), взаимодействие с внутренней памятью данных осуществляется на командном уровне, при этом сигналы \overline{RD} и \overline{WR} не вырабатываются.

Внешняя память программ и внешняя память данных могут комбинироваться путем совмещения сигналов \overline{RD} и \overline{PSEN} по схеме "логического И" для получения строга внешней памяти (программ/данных).

Нижние 128 байт внутренней памяти данных (lower 128) присутствуют на всех кристаллах MCS-51 и показаны на рис.5.

Первые 32 байта представляют собой 4 банка (Register Bank) по 8 регистров (R7...R0). Регистры R0 и R1 в любом из банков могут использоваться в качестве регистров косвенного адреса.

Следующие за регистровыми банками 16 байт образуют блок побитно-адресуемого пространства. Набор инструкций MCS-51 содержит широкий выбор операций над битами, а 128 бит в этом блоке адресуются прямо и адреса имеют значения от 00_H до 7F_H.

Все байты в нижней 128-байтной половине памяти могут адресоваться как прямо, так и косвенно.

Верхняя 128 байтная половина памяти ОЗУ (upper 128) в микросхеме I-8051 отсутствует, но имеется в версиях кристаллов с 256 байтами ОЗУ. В этом случае область "Upper 128" доступна только при косвенной адресации. Область SFR (Special Function Register) доступна только при прямой адресации.

Размещение регистров специальных функций в пространстве SFR показано на рис.6. Они включают в себя регистры портов, таймеры, средства управления периферией и так далее.

7F _H	Побайтно-адресуемая область ОЗУ							
30 _H	(direct, indirect)							
2F _H	7F _H	7E _H	7D _H	7C _H	7B _H	7A _H	79 _H	78 _H
2E _H	77 _H	76 _H	75 _H	74 _H	73 _H	72 _H	71 _H	70 _H
	Побитно-адресуемая область ОЗУ							
	(direct)							
21 _H	0F _H	0E _H	0D _H	0C _H	0B _H	0A _H	09 _H	08 _H
20 _H	07 _H	06 _H	05 _H	04 _H	03 _H	02 _H	01 _H	00 _H
1F _H	RB3							
18 _H	RB2							
17 _H	RB2							
10 _H	RB1							
0F _H	RB1							
08 _H	RB1							
07 _H	★ SP после RESET							
00 _H	RB0(R7+R0)							

Рис.5. Нижние 128 байт внутреннего ОЗУ.

	побитовая адресация							
	↓							
	8 байт							
F8 _H								FF _H
F0 _H	B							F7 _H
E8 _H								EF _H
E0 _H	ACC							E7 _H
D8 _H								DF _H
D0 _H	PSW							D7 _H
C8 _H								CF _H
C0 _H								C7 _H
B8 _H	IP							BF _H
B0 _H	P3							B7 _H
A8 _H	IE							AF _H
A0 _H	P2							A7 _H
98 _H	SCON	SBUF						9F _H
90 _H	P1							97 _H
88 _H	TCON	TMOD	TL0	TL1	TH0	TH1		8F _H
80 _H	P0	SP	DPL	DPH			PCON	87 _H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F

Рис.6. Размещение регистров специальных функций в пространстве SFR.

Для 16 адресов в пространстве SFR имеется возможность как байтовой, так и битовой адресации. Для побитно-адресуемых регистров шестнадцатеричный адрес заканчивается на "0_H" или на "8_H". Битовые адреса в этой области имеют значения от 80_H до FF_H.

Вся серия кристаллов семейства MCS-51 имеет базовый набор SFR, как и в микросхеме I-8051, расположенный по тем же адресам. Однако в кристаллах, представляющих собой дальнейшее развитие семейства в область SFR, добавляются новые регистры для расширения

функциональных возможностей, например, таймер 2, матрица программируемых счетчиков (Programmable Counter Array - PCA), сторожевой таймер (Watchdog Timer - WDT), регистры, обслуживающие прямой доступ к памяти, аналого-цифровое преобразование, коммуникационный последовательный канал и так далее.

Регистры специальных функций, доступные при побитной адресации, даны в таблице 1, а регистры специальных функций, доступные при байтовой адресации, в таблице 2.

Назначение регистров области SFR и отдельных битов этих регистров приведены ниже.

Аккумулятор А (Accumulator, адрес E0_H).

Команды архитектуры MCS-51 используют аккумулятор как источник и как приемник при вычислениях и пересылках. Кроме обращения к аккумулятору командами, использующими мнемонику "A", имеется возможность побитовой или побайтовой адресации, как SFR-регистра.

Регистр В (Multiplication Register, адрес F0_H).

Регистр В используется как источник и как приемник при операциях умножения и деления, обращение к нему, как к регистру SFR, производится аналогично аккумулятору.

Слово состояния программы PSW (Program Status Word, адрес D0_H).

Данный регистр содержит биты, отражающие результаты выполнения операций, биты выбора регистрового банка и бит общего назначения, доступный пользователю. PSW отображен на область SFR и содержит:

- PSW.7 - бит переноса (Carry Flag, CY);
- PSW.6 - бит вспомогательного переноса (Auxiliary Carry Flag, AC);
- PSW.5 - пользовательский флаг (General Purpose Status Flag, F0);
- PSW.4, PSW.3 - биты выбора регистрового банка (Register Bank Select Bits, RS1 и RS0);
- PSW.2 - бит переполнения (Overflow Flag, OV);
- PSW.1 - зарезервирован (Reserved);
- PSW.0 - бит четности (Parity Flag, P).

Регистры SFR при побитной адресации.

Имя SFR	Адрес SFR	Адрес и мнемоника битов								Значение при сбросе
		87 _H P0.7	86 _H P0.6	85 _H P0.5	84 _H P0.4	83 _H P0.3	82 _H P0.2	81 _H P0.1	80 _H P0.0	
P0	80 _H									FF _H
TCON	88 _H	8F _H TF1	8E _H TR1	8D _H TF0	8C _H TR0	8B _H IF1	8A _H IT1	89 _H IE0	88 _H IT0	00 _H
P1	90 _H	97 _H P1.7	96 _H P1.6	95 _H P1.5	94 _H P1.4	93 _H P1.3	92 _H P1.2	91 _H P1.1	90 _H P1.0	FF _H
SCON	98 _H	9F _H SM0	9E _H SM1	9D _H SM2	9C _H REN	9B _H TB8	9A _H RB8	99 _H TI	98 _H RI	00 _H
P2	A0 _H	A7 _H P2.7	A6 _H P2.6	A5 _H P2.5	A4 _H P2.4	A3 _H P2.3	A2 _H P2.2	A1 _H P2.1	A0 _H P2.0	FF _H
IE	A8 _H	AF _H EA	AE _H ⌚	AD _H ⌚	AC _H ES	AB _H ET1	AA _H EX1	A9 _H ET0	A8 _H EX0	0XX00000 _B
P3	B0 _H	B7 _H P3.7	B6 _H P3.6	B5 _H P3.5	B4 _H P3.4	B3 _H P3.3	B2 _H P3.2	B1 _H P3.1	B0 _H P3.0	FF _H
IP	B8 _H	BF _H ⌚	BE _H ⌚	BD _H ⌚	BC _H PS	BB _H PT1	BA _H PX1	B9 _H PT0	B8 _H PX0	XXX00000 _B
PSW	D0 _H	D7 _H CY	D6 _H AC	D5 _H F0	D4 _H RS1	D3 _H RS0	D2 _H OV	D1 _H ⌚	D0 _H P	00 _H
A	E0 _H	E7 _H ⌚	E6 _H ⌚	E5 _H ⌚	E4 _H ⌚	E3 _H ⌚	E2 _H ⌚	E1 _H ⌚	E0 _H ⌚	00 _H
B	F0 _H	F7 _H ⌚	F6 _H ⌚	F5 _H ⌚	F4 _H ⌚	F3 _H ⌚	F2 _H ⌚	F1 _H ⌚	F0 _H ⌚	00 _H

Битами RS1, RS0 выбираются ячейки памяти, которые будут служить активным банком регистрового файла (регистры R0+R7).

RS1	RS0	Банк	Адрес
0	0	0	00 _H +07 _H
0	1	1	08 _H +0F _H
1	0	2	10 _H +17 _H
1	1	3	18 _H +1F _H

Регистры SFR при байтовой адресации.

Имя SFR	Адрес SFR	Мнемоника битов								Значение при сбросе	
SP	81 _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚	00000111 _B	
DPL	82 _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚	00 _H	
DPH	83 _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚	00 _H	
PCON	87 _H	SMOD	⌚	⌚	⌚	⌚	GF1	GF0	PD	IDL	0XXXXXXX _B 0XXX0000 _B
TMOD	89 _H	GATE1	$C/\overline{T1}$	M1.1	M0.1	GATE0	$C/\overline{T0}$	M1.0	M0.0		00 _H
TL0	8A _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚		00 _H
TL1	8B _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚		00 _H
TH0	8C _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚		00 _H
TH1	8D _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚		00 _H
SBUF	99 _H	⌚	⌚	⌚	⌚	⌚	⌚	⌚	⌚		XXXXXXXX _B

Регистры портов P0+P3 (адреса 80_H, 90_H, A0_H, B0_H).

Каждый порт является фиксатором - защелкой и может адресоваться как побайтно, так и побитно. Помимо работы в качестве обычных портов ввода/вывода, линии портов могут выполнять ряд альтернативных функций:

- через порт 0 (в мультиплексном режиме) выводится младший байт адреса, а также выдается и принимается в микроконтроллер байт данных при работе с внешней памятью программ/данных. Задаются данные при программировании внутренней памяти программ и читается ее содержимое;
- через порт 2 выводится байт старший байт адреса внешней памяти программ и данных, а также задаются старшие разряды адреса при программировании и верификации УФРПЗУ;
- порт 1 предназначен для задания младшего байта адреса при программировании и проверке ПЗУ микросхемы. Линии порта используются также для обслуживания работы T/C2 и модуля PCA (для тех версий кристаллов MCS-51, в составе которых имеются эти функциональные узлы);
- порт 3 имеет следующие альтернативные функции:
 - P3.7 - строб чтения из внешней памяти данных (Read Data for External Memory, \overline{RD});
 - P3.6 - строб записи во внешнюю память данных (Write Data for External Memory, \overline{WR});
 - P3.5 - внешний вход T/C1 (Timer/Counter 1 External Input, T1);
 - P3.4 - внешний вход T/C0 (Timer/Counter 0 External Input, T0);
 - P3.3 - вход внешнего прерывания 1 (External Interrupt 1 Input Pin, $\overline{INT1}$);
 - P3.2 - вход внешнего прерывания 0 (External Interrupt 0 Input Pin, $\overline{INT0}$);
 - P3.1 - выход данных передатчика последовательного порта (Serial Port Transmit Pin, TxD);
 - P3.0 - вход данных передатчика последовательного порта (Serial Port Receive Pin, RxD).

Указатель стека SP (Stack Pointer, адрес 81_H).

Используется для указания на вершину стека в операциях записи в стек и чтения из него. Неявно используется такими командами, как PUSH, RET, RETI, POP. По аппаратному сбросу от ЦП устанавливается в значение 07_H (область стека в этом случае начинается с адреса внутренней памяти данных 08_H) и инкрементируется при каждой записи в стек. Запись в SFR-регистр-SP (с

использованием байтовой адресации) производится для предопределения положения стека во внутренней памяти данных.

Указатель данных DPTR (Data Pointer, адреса 82_H, 83_H).

Команды архитектуры MCS-51 используют DPTR для пересылки данных, пересылки кода и для переходов (JMP@A+DPTR). DPTR состоит из двух регистров: младшего - DPL и старшего - DPH, обращение к ним - только байтовое.

Регистр управления энергопотреблением PCON (Power Control Register, адрес 87_H).

Для кристаллов, выполненных по NMOS-технологии, данный регистр имеет только один значащий бит - SMOD, который управляет скоростью работы последовательного порта. Для кристаллов, изготовленных по CHMOS-технологии, SFR-регистр PCON имеет следующее назначение битов:

PCON.7 - бит удвоения скорости передачи (Double Baud Rate Bit, SMOD);

PCON.6+PCON.4 - зарезервированы (Reserved);

PCON.3, PCON.2 - флаги общего назначения (General Purpose Flags, GF1, GF0);

PCON.1 - бит режима микропотребления (Power Down Bit, PD);

PCON.0 - бит холостого режима (Idle Mode Bit, IDL).

При установке холостого режима производится отключение ЦП при сохранении работоспособности внутрикристалльных периферийных устройств и ОЗУ.

В режиме микропотребления приостанавливаются все операции при сохранении данных в ОЗУ (потребление тока микроконтроллером в этом случае менее 10 мкА). Выход из режима микропотребления осуществляется аппаратным сбросом длительностью не менее 10 мс (время восстановления работы задающего генератора), выход из холостого режима может быть произведен активизацией любого разрешенного прерывания, либо по сигналу RESET длительностью не менее двух машинных циклов.

Регистры таймеров/счетчиков TL0, TL1, TH0, TH1 (адреса 8A_H, 8B_H, 8C_H, 8D_H).

Образуют 16-битные (Low/High) регистры таймеров/счетчиков "T/C0" и "T/C1". Обращение к регистрам только байтовое. Подробно описаны в подразделе "Организация таймеров/счетчиков".

Регистр режима таймеров/счетчиков TMOD (Timer/Counter Mode Control Register, адрес 89_H).

Регистр управления таймеров/счетчиков TCON (Timer/Counter Control Register, адрес 88_H).

Предназначены для управления работой таймерной секции микроконтроллера. Подробно описаны в подразделе "Организация таймеров/счетчиков".

Буфер последовательного порта SBUF (Serial Data Buffer, адрес 99_H).

Представляет собой два отдельных регистра. При записи в SBUF загружается "буфер передачи" последовательного порта, при чтении SBUF считывается содержимое "буфера приема" последовательного порта.

Регистр управления последовательным портом SCON (Serial Port Control Register, адрес 98_H).

Предназначен для управления работой последовательного порта. Обращение к данному регистру может быть как байтовым, так и побитным.

Подробное описание регистров SBUF и SCON дано в подразделе "Организация последовательного интерфейса".

Регистр разрешения прерываний IE (Interrupt Enable Register, адрес A8_H).

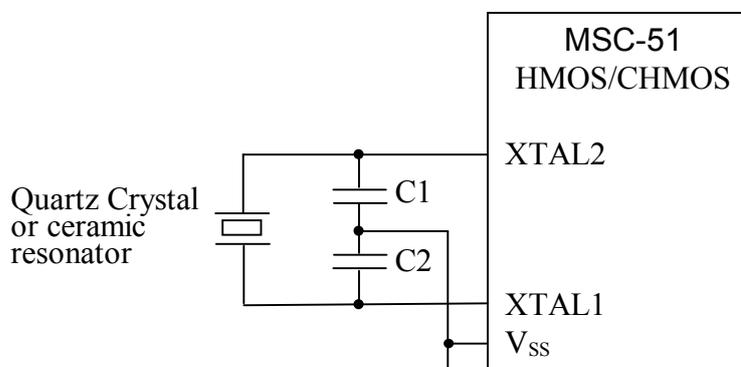
Регистр управления приоритетом прерывания IP (Interrupt Priority Control Register, адрес B8_H).

Поддерживают работу системы прерываний микроконтроллера. Подробное описание работы с регистрами дано в подразделе "Организация прерываний".

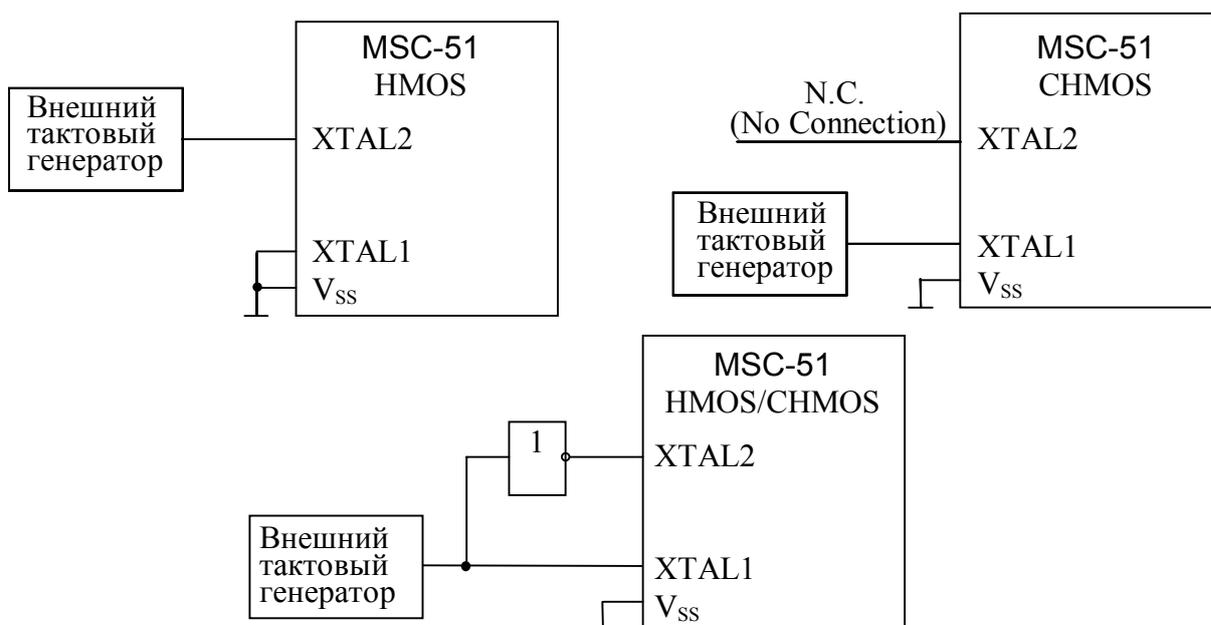
2.1. Синхронизация микроконтроллера.

Источником синхронизации для микроконтроллеров семейства MCS-51 может быть внешний тактовый генератор или внутренний генератор с внешним кварцевым резонатором. В случае использования встроенного генератора к выводам XTAL1 и XTAL2 подключается либо кварцевый, либо керамический резонатор с частотой, находящейся в пределах рабочего диапазона для конкретного кристалла номиналом в $50 \text{ pF} \pm 10\%$, подключенные к общей шине (рис.7). Для различных версий кристаллов частота тактового генератора может находиться в пределах от 3,5 до 33 МГц (в режиме программирования тактовая частота должна быть в пределах от 4 до 6 МГц).

Примеры использования внутреннего и внешнего тактового генератора показаны на рис.7.



а - использование внутреннего тактового генератора



б - использование внешнего тактового генератора

Рис.7. Примеры использования внутреннего и внешнего тактового генератора.

Отметим, что в NMOS-кристаллах сигнал внешнего генератора подается на вывод XTAL2, а в CHMOS - на вывод XTAL1. Это связано с тем, что встроенный генератор реализован на одном инверторе (вывод XTAL1 является его входом, а XTAL2 - выходом), причем в NMOS-микросхемах тактовая частота снимается с выхода этого инвертора, а в CHMOS - со входа.

Внутренний тактовый генератор необходим для синхронизации последовательности состояний ЦП, образующих машинный цикл, а также для правильной работы всех внутренних защелок и выходных буферов портов. Машинный цикл состоит из шести последовательных состояний (States) от S1 до S6, каждое из которых, в свою очередь, подразделяется на две фазы: фазу 1 (Phase 1 - P1) и фазу 2 (Phase 2 - P2). Таким образом, машинный цикл может быть определен как последовательность временных интервалов S1P1, S1P2, S2P1, ..., S6P2. Длительность фазы равна периоду следования тактовых импульсов, поэтому машинный цикл занимает 12 тактовых периодов.

2.2. Организация таймеров/счетчиков.

Таймеры/счетчики (T/C0 и T/C1) предназначены для подсчета внешних событий (выводы T0 и T1), организации программно-управляемых временных задержек и измерения временных интервалов. Таймер 1 может также служить генератором скорости передачи для последовательного порта.

Таймер/счетчик, работая в режиме таймера, ведет подсчет тактов деленной системной частоты (запрограммированный промежуток времени) и выдает запрос прерывания. Регистр таймера инкрементируется один раз в каждом периферийном цикле. Поскольку цикл состоит из 12 тактов, то скорость счета таймера равна $F_{OSC}/12$.

В режиме счетчика регистр таймера ведет подсчет (предустановленного числа событий) отрицательных перепадов сигнала на внешнем входе и по окончании счета выдает запрос прерывания. Поскольку распознавание отрицательного перехода внешнего сигнала занимает 24 периода тактовой частоты (2 цикла), то максимальная скорость счета равна $F_{OSC}/24$. Ограничений на рабочий цикл не накладывается, но чтобы гарантировать опрос конкретного уровня сигнала хотя бы один раз до момента его смены, он должен удерживаться на входе хотя бы в течение одного полного периферийного цикла.

Программное управление функционированием T/C0 и T/C1 обеспечивают SFR-регистры TMOD и TCON. Возможны 4 режима работы T/C микроконтроллера, которые определяются установкой соответствующих битов регистра TMOD. Режимы 0 (13-битовый таймер), 1 (16-битовый таймер) и 2 (8-битовый таймер с автоперезагрузкой) полностью идентичны для обоих T/C. В режиме 3 (два 8-битовых регистра) работает только T/C0, T/C1 в этом режиме заблокирован ("лишен" бита управления запуском TR1 и флага переполнения TF1) и сохраняет содержимое своих регистров TL1 и TH1. Логика работы T/C0 и T/C1 в режимах 0,1,2,3 показана на рис.8. Путем соответствующего программирования регистров TMOD и TCON осуществляется включение и выключение таймеров/счетчиков, выбор источника их тактирования и установка определенного режима их работы. Функциональное назначение разрядов этих регистров следующее.

Регистр TMOD.

TMOD.7, TMOD.3 - Timer 1 Gate, Timer 0 Gate.

Если $GATE_x=1$ и $TR_x=1$, то включение и выключение соответствующего таймера осуществляется внешним сигналом на входе INT_x . Когда $GATE_x=0$, бит управления запуском $TR_x=1$ разрешает прохождение входных сигналов от выбранного источника тактирования.

TMOD.6, TMOD.2 - Timer 1 Counter/Timer Select, Timer 0 Counter/Timer Select.

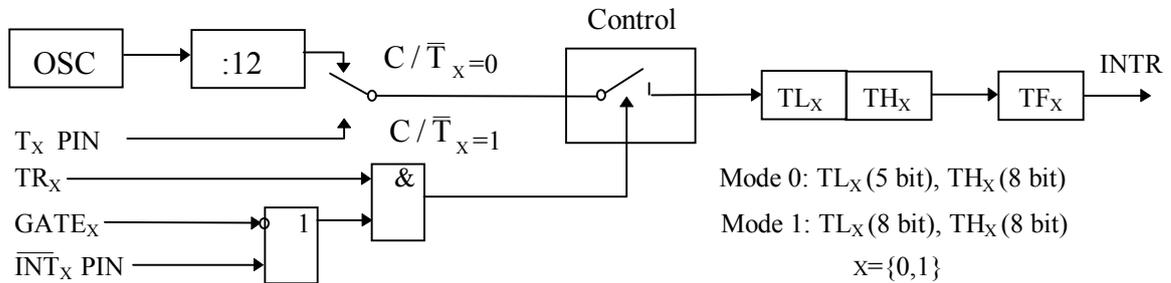
$C/\overline{T_x} = 0$ выбирает функцию таймера (подсчет импульсов деленной системной частоты);
 $C/\overline{T_x} = 1$ выбирает функцию счетчика (подсчет отрицательных переходов сигнала на внешнем выводе T_x).

TMOD.5, TMOD.4 - Timer 1 Mode Select,

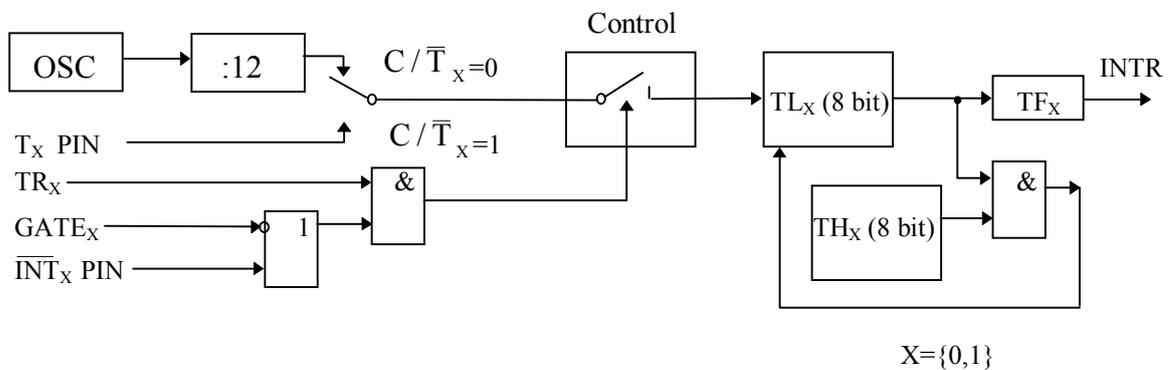
TMOD.1, TMOD.0 - Timer 0 Mode Select.

Mx.1 Mx.0

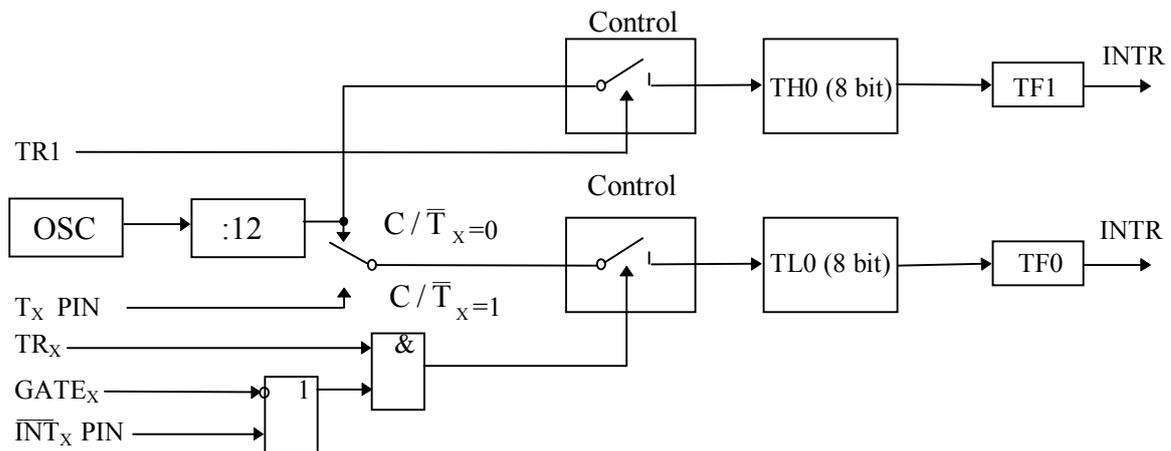
- 0 0 - Mode 0: 8-битовый таймер/счетчик (ТНх) с 5-битовым прескалером (ТLх);
- 0 1 - Mode 1: 16-битовый таймер/счетчик;
- 1 0 - Mode 2: 8-битовый автоперезагружаемый таймер/счетчик (ТLх).
Константа перезагрузки предварительно заносится в ТНх.
- 1 1 - Mode 3: ТL0 - это 8-битовый таймер/счетчик; ТН0 - 8-битовый таймер,
использующий биты ТR1 и ТF1.



а - логика работы Т/С0 и Т/С1 в режимах 0 и 1



б - логика работы Т/С0 и Т/С1 в режиме 2



в - логика работы Т/С0 в режиме 3

Рис.8. Логика работы Т/С0 и Т/С1 в режимах 0, 1, 2 и 3

Регистр TCON.

TCON.7, TCON.5 - Timer 1 Overflow Flag, Timer 0 Overflow Flag.

TFx устанавливается при переполнении соответствующего таймера (при переходе из FF_H в 00_H); при разрешении прерывания от T/Cx установка флага вызывает прерывание; очищается аппаратно, когда процессор переходит на подпрограмму обработки прерывания.

TCON.6, TCON.4 - Timer 1 Run Control Bit, Timer 0 Run Control Bit.

TRx устанавливается и сбрасывается программно при включении/выключении T/Cx.

TCON.3, TCON.1 - External Interrupt 1 Edge Flag, External Interrupt 0 Edge Flag.

IE_x устанавливается аппаратно, когда обнаружено внешнее прерывание (по фронту или уровню сигнала) на выводе INT_x; сбрасывается аппаратно во время обработки прерывания только в том случае, когда прерывание было вызвано фронтом сигнала.

TCON.2, TCON.0 - Interrupt 1 Type Control Bit, Interrupt 0 Type Control Bit.

IT_x определяет тип воспринимаемого сигнала на входе INT_x; для выбора срабатывания по фронту сигнала (высокий и низкий) нужно установить этот бит, для срабатывания по уровню (активный низкий уровень) нужно сбросить этот бит.

2.3. Организация прерываний.

Архитектура системы управления прерываниями для базовой модели (I-8051) показана на рис.9.

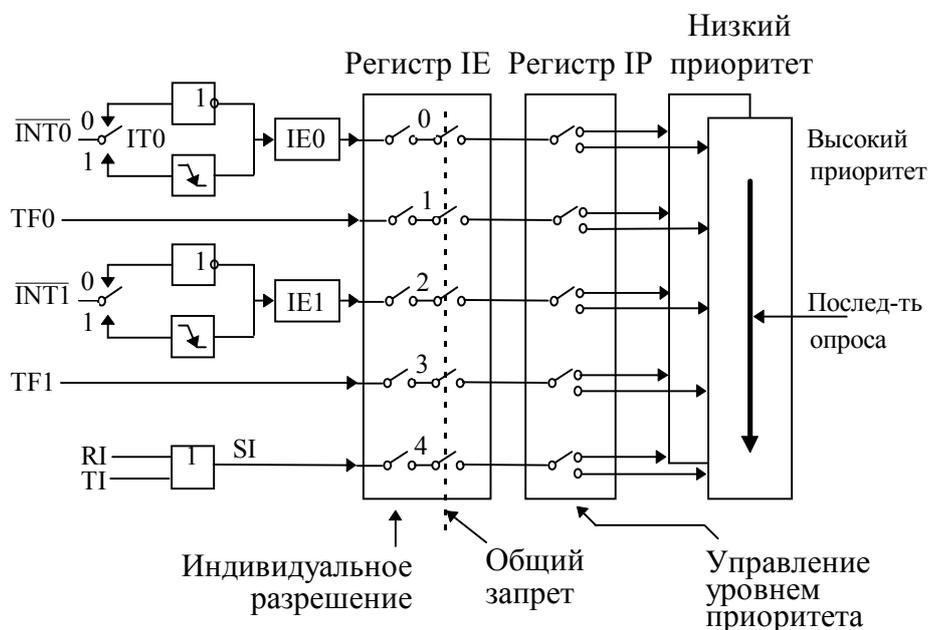


Рис.9. Система управления прерываниями

Каждый вид прерывания индивидуально разрешается или запрещается установкой или очисткой соответствующих бит SFR-регистра IE (Interrupt Enable). Регистр содержит также бит общего запрещения, при котором блокируются все прерывания. Назначение битов регистра IE следующее:

- IE.7 - Enable All Interrupts (EA), разрешение прерываний от всех источников;
- IE.6, IE.5 - reserved, зарезервированы для дальнейшего использования;
- IE.4 - Enable Serial Port Interrupt (ES), разрешение прерывания от последовательного порта;
- IE.3 - Enable Timer 1 Interrupt (ET1), разрешение прерывания по переполнению T/C1;
- IE.2 - Enable External Interrupt 1 (EX1), разрешение прерывания по внешнему сигналу на входе $\overline{INT1}$;
- IE.1 - Enable Timer 0 Interrupt (ET0), разрешение прерывания по переполнению T/C0;
- IE.0 - Enable External Interrupt 0 (EX0), разрешение прерывания по внешнему сигналу на входе $\overline{INT0}$.

Вид воспринимаемого внешнего сигнала (Interrupt Type) по входам $\overline{INT0}$, $\overline{INT1}$ задается программированием соответствующих бит в регистре TCON (TCON.0, TCON.2), а разряды TCON.1, TCON.3 (Interrupt Edge) выполняют роль флагов, фиксирующих поступившие запросы.

Прерывание по последовательному порту вырабатывается функцией логического "ИЛИ" битов RI и TI в регистре SCON.

Каждому типу прерывания может быть программно присвоен высокий или низкий приоритетный уровень путем установки или очистки соответствующего бита в SFR-регистре IP (Interrupt Priority), который имеет (для кристалла I-8051) следующее назначение битов:

IP.7, IP.6, IP.5 - reserved, зарезервированы для дальнейшего использования;

IP.4 - Priority of Serial Port Interrupt (PS), приоритет последовательного порта;

IP.3 - Priority of Timer 1 Interrupt (PT1), приоритет T/C1;

IP.2 - Priority of External Interrupt 1 (PX1), приоритет внешнего прерывания $\overline{INT1}$;

IP.1 - Priority of Timer 0 Interrupt (PT0), приоритет T/C0;

IP.0 - Priority of External Interrupt 0 (PX0), приоритет внешнего прерывания $\overline{INT0}$.

Низкоприоритетное прерывание может прерываться высокоприоритетным, но никогда не прерывается запросом того же уровня приоритета. Поэтому, если одновременно возникают два прерывания с различным уровнем приоритета, то сначала выполняется высокоприоритетное. Если же подобная ситуация складывается для прерываний с одинаковым уровнем приоритета, то последовательность их обработки определяется специальной последовательностью опроса флагов прерываний (Interrupt Polling Sequence). В процессе работы "фиксация" всех флагов прерываний происходит в состоянии S5 каждого машинного цикла, а опрос - во время следующего машинного цикла.

2.4. Организация последовательного интерфейса.

Последовательный порт ввода-вывода обеспечивает синхронный и асинхронный режимы связи и предназначен для поддержки взаимодействия микроконтроллера с модемами и другими внешними устройствами. Он работает как универсальный асинхронный приемопередатчик (UART) в трех полнодуплексных режимах (режимы 1, 2 и 3), причем асинхронная передача и прием могут вестись одновременно и с одинаковой или разной скоростью. Блок UART поддерживает обнаружение однобитовых ошибок кадра передачи, мультипроцессорный обмен и автоматическое распознавание адреса. Последовательный порт работает также и в синхронном режиме (режиме 0).

В состав UART входят передающий и принимающий сдвигающие регистры, буфер приемника и передатчика SBUF и блок управления работой порта с регистром управления SCON. Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи. Наличие буферного регистра приемника позволяет совмещать операцию чтения из SBUF ранее принятого байта с приемом очередного. Если к моменту окончания приема очередного байта предыдущий не был считан из SBUF, то он будет потерян.

Скорость передачи/приема (частота пересылки битов) зависит от запрограммированного режима работы порта и значения бита SMOD в регистре PCON. Как указывалось ранее, последовательный порт может работать в одном из четырех режимов.

Режим 0. Информация и передается и принимается через внешний вывод RxD (линия порта P3.0). Через выход передатчика TxD (P3.1) выдаются тактовые импульсы, стробирующие передаваемые или принимаемые биты. Формат посылки - 8 бит. Сдвиг данных (младшим значащим битом вперед) происходит на последней фазе (S6P2) каждого периферийного цикла, поэтому частота приема и передачи равна $F_{osc}/12$.

Режим 1. Информация передается через вывод TxD, а принимается через RxD. Формат посылки - 10 бит (стартовый - "0", 8 информационных и стоповый - "1"). При приеме стоп-бит поступает в разряд RB8 регистра SCON. Частота обмена задается программированием T/C1.

Режим 2. Последовательные данные выдаются на вывод TxD, а принимаются по выводу RxD. Кадр данных состоит из 11 бит (старт-бит, 8 информационных, программируемый девятый бит и стоп-бит). При передаче девятый бит транслируется из SCON.TB8, а при приеме - передается в SCON.RB8. Девятый бит используется по усмотрению программиста, например, как бит контроля информации по четности или как флаг ("control/data" или "address/data") идентификации передаваемого или принимаемого кадра. Частота обмена равна $2^{SMOD} \cdot F_{OSC} / 64$.

Режим 3. Идентичен режиму 2 с тем отличием, что частота обмена задается программированием T/C1 (как и в режиме 1) и равна $2^{SMOD} \cdot F_{OV} / 32$, где F_{OV} - частота переполнений (overflow) таймера 1. В большинстве приложений таймер 1 конфигурируется по схеме режима 2 (8-битовый таймер с автоперезагрузкой), для которого F_{OV} определяется выражением:

$$F_{OV} = \frac{F_{OSC}}{12 \cdot [256 - (TH1)]}$$

где (TH1) - десятичный код содержимого TH1.

Схема синхронизации работы последовательного порта в режимах 1, 2 и 3 показана на рис.10.

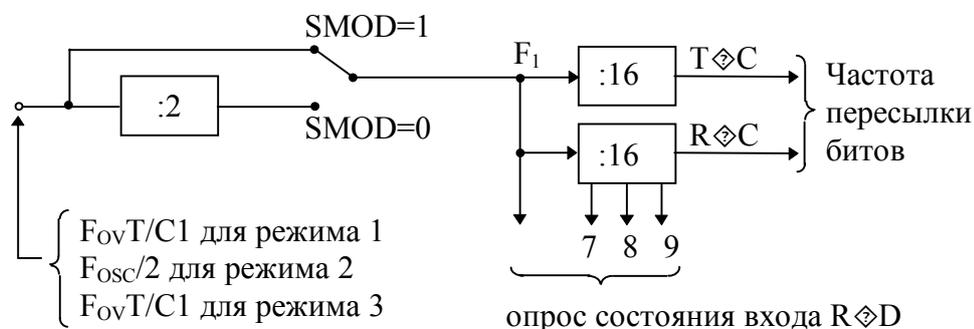


Рис.10. Схема синхронизации работы последовательного порта

Отметим особенности организации приема. Прием начинается при обнаружении перехода сигнала на входе R/D из "1" в "0". Для отслеживания такого перехода вход R/D аппаратно опрашивается с частотой F_1 . Когда переход сигнала на входе R/D из "1" в "0" обнаружен, то немедленно сбрасывается счетчик-делитель на 16 в цепи сигнала R/C, в результате чего происходит совмещение моментов переполнения этого счетчика-делителя с границами смены битов принимаемого кадра. 16 состояний счетчика-делителя делят время, в течение которого каждый принимаемый бит кадра присутствует на входе R/D, на 16 фаз. В фазах 7, 8 и 9 специальное устройство микроконтроллера, бит детектор, считывает со входа R/D три значения принимаемого бита и по мажоритарному принципу (два из трех) выбирает одно из них и подает его на вход регистра сдвига приемника.

Если мажоритарный отбор при приеме первого бита кадра (старт-бит) показывает ненулевое значение бита, то все устройства блока приема сбрасываются и начинаются и начинается отслеживание следующего перехода из "1" в "0" на входе R/D. Таким образом обеспечивается защита от сбойных (ложных) старт-битов.

Настройка последовательного порта на определенный режим работы осуществляется путем программного задания соответствующего управляющего слова в регистр SCON, функциональное назначение битов которого отражено ниже:

- SCON.7 ⌚ Serial Mode Control Bit 0 (SM0), бит 0 выбора режима работы;
- SCON.6 ⌚ Serial Mode Control Bit 1 (SM1), бит 1 выбора режима работы;
- SCON.5 ⌚ Serial Mode Control Bit 2 (SM2), бит 2 выбора режима работы;
- SCON.4 ⌚ Receiver Enable Bit (REN), бит разрешения приема;
- SCON.3 ⌚ Transmit Bit 8 (TB8), девятый бит передаваемых данных в режимах 2 и 3;

- SCON.2 ⌚ Receive Bit 8 (RB8), девятый бит принимаемых данных в режимах 2 и 3, в режиме 1 (при SM2=0) в RB8 фиксируется значение принятого стоп-бита;
- SCON.1 ⌚ Transmit Interrupt Flag (TI), флаг прерывания передатчика, устанавливается аппаратно в конце времени выдачи последнего бита кадра, очищается программно;
- SCON.0 ⌚ Receive Interrupt Flag (RI), флаг прерывания приемника, устанавливается аппаратно в конце времени приема последнего бита кадра, очищается программно (при SM2=1 процедура использования флага RI изменяется).

Задание режима работы последовательного порта дано в табл.3.

Таблица 3.

Режимы работы УАПП.

SM0	SM1	SM2*	Режим	Характеристика режима
0	0	0	0	Сдвигающий регистр. Скорость обмена - $F_{OSC} / 12$
0	1	1/0	1	8-бит. УАПП. Скорость обмена задается таймером 1 и равна $2^{SMOD} \cdot F_{OV} / 32$
1	0	1/0	2	9-бит. УАПП. Скорость обмена - $2^{SMOD} \cdot F_{OSC} / 64$
1	1	1/0	3	9-бит. УАПП. Скорость обмена задается таймером 1 и равна $2^{SMOD} \cdot F_{OV} / 32$

*) Бит SM2 - бит разрешения мультипроцессорной работы.

В режимах 2 и 3 при SM2=1 флаг RI не активизируется, если девятый принятый бит не равен 1, а в режиме 0 SM2 должен быть равен 0.

Во всех четырех режимах работы передача инициируется любой командой, определяющей буферный регистр SBUF как получатель байта (например, MOV SBUF,A). Прием в режиме 0 осуществляется при условии, что REN=1 и RI=0, а в режимах 1,2,3 прием начинается с приходом старт-бита, если REN=1.

В мультипроцессорных системах, где требуется обмен информацией между ведущим и несколькими ведомыми микроконтроллерами по одному последовательному каналу, использование бита SM2 и одного из 9-битных режимов последовательного порта предоставляет пользователю возможность решения таких задач, как:

- идентификация адресных или командных (control) кадров и кадров данных (data);
- распознавание "собственного" и "широковещательного" адреса;
- обнаружение ошибок при приеме кадра.

Функция автоматического распознавания адреса включается, когда разрешен мультипроцессорный обмен (установлен бит SM2 в регистре SCON), при этом процедура взаимодействия микроконтроллеров в системе может быть организована следующим образом. Когда ведущий микроконтроллер передает блок данных одному (заданный адрес) или нескольким (широковещательный адрес) ведомым, то он вначале выставляет на передачу адресный кадр, который идентифицирует требуемого адресата. В качестве признака "кадр адреса/кадр данных" может использоваться содержимое девятого бита (режимы 2 и 3 последовательного порта). Ведомый микроконтроллер проверяет на предмет совпадения адреса кадры с установленным девятым битом (признаком адреса). Если адрес приемника совпадает, то приемник аппаратно устанавливает бит RB8 и бит RI в регистре SCON, что вызывает прерывание (биты ES и EA в регистре IE должны быть установлены, чтобы бит RI сгенерировал прерывание). После этого программа подчиненного процессора-адресата очищает бит SM2 в регистре SCON и готовится к приему блока данных. Остальные "Slave"- устройства безразличны к этим кадрам данных (девятый бит=0), потому что они реагируют только на собственный адрес.

Данный механизм работает и в случае нахождения приемопередатчика микроконтроллера в 8-битном режиме (режим 1) с той лишь разницей, что стоп-бит занимает место девятого бита данных. Бит RI устанавливается только тогда, когда принимаемый "control"- кадр совпадает по адресу с адресом "Slave"- устройства и оканчивается действительным стоп-битом.

Функция обнаружения битовых ошибок кадра (в режимах 2 и 3 последовательного порта) может быть реализована следующим образом.

Для повышения достоверности передачи в качестве девятого бита (разряд TB8 регистра SCON) выбирают бит четности из регистра PSW, используя следующую процедуру:

<i>MOV</i>	<i>C, P</i>	бит P * в TB8
<i>MOV</i>	<i>TB8, C</i>	
<i>MOV</i>	<i>SBUF, A</i>	- инициализация передачи

При приеме девятый бит данных поступает в разряд RB8 регистра SCON и используется в качестве контрольного для обнаружения одиночных ошибок в принятом байте.

2.5. Система команд семейства MCS-51.

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

- команды пересылки данных (Data Transfer);
- арифметические команды (Arithmetic);
- логические команды (Logic);
- команды передачи управления (Control Transfer or Program Branching);
- команды обработки битовых переменных (Boolean Variable Manipulation).

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 регистравого 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 (data₈, data₁₆).

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.

2.5.1. Команды пересылки данных.

Данную группу команд можно подразделить на следующие подгруппы:

- команды передачи данных, использующие внутреннюю память данных;
- команды передачи данных, использующие внешнюю память данных;
- команды работы с таблицами.

В табл.4 показаны команды, использующие внутреннее ОЗУ и применяемые при этом режимы адресации.

Таблица 4.

Команды передачи данных, использующие внутреннее ОЗУ.

Мнемоника	Операция	Режимы адресации			
		Dir	Ind	Reg	Imm
MOV A, <src>	A=<src>	x	x	x	x
MOV <dest>, A	<dest>=A	x	x	x	
MOV <dest>, <src>	<dest>=<src>	x	x	x	x
MOV DPTR, #data ₁₆	DPTR=16-bit immediate constant				x
PUSH <src>	INC SP:MOV"@SP",<src>	x			
POP <dst>	MOV <dest>, "@SP": DEC SP	x			
XCH A, <byte>	ACC and <byte> exchange data	x	x	x	
XCHD A, @Ri	ACC and @Ri exchange low nibbles		x		

В приведенной таблице используются стандартные обозначения: <src>, (source) - источник, <dest>, (destination) - приемник. Команда MOV <dest>, <src> позволяет пересылать данные между ячейками внутреннего ОЗУ или SFR без использования аккумулятора. При этом работа с верхними 128 байтами внутреннего ОЗУ может осуществляться только в режиме косвенной адресации, а обращение к регистрам SFR - только в режиме прямой адресации.

Во всех микросхемах MCS-51 стек размещается непосредственно в резидентной памяти данных чипа и растет в сторону увеличения адресов. Поскольку по отношению к SP используется косвенная адресация, то стек может попасть и в область "Upper 128", но не в SFR. В тех кристаллах, где "Upper 128" отсутствует, увеличение стека за границу 128 байт приведет к потере данных.

Операция XCH A, <byte> применяется для обмена данными (exchange) между аккумулятором и адресуемым байтом. Команда XCHD A, @Ri аналогична предыдущей, но выполняется только для младших тетрад (ниблов), участвующих в обмене.

В табл.5 приведены команды для работы с внешней памятью данных.

Таблица 5.

Команды передачи данных, использующие внешнюю память данных

Мнемоника	Операция	Разрядность данных
MOVX A, @Ri	Read external RAM @Ri	8 bits
MOVX @Ri, A	Write external RAM @Ri	8 bits
MOVX A, @DPTR	Read external RAM @DPTR	16 bits
MOVX @DPTR, A	Write external RAM @DPTR	16 bits

Отметим, что при любом доступе к внешней памяти данных используется только косвенная адресация и обмен информацией осуществляется исключительно через аккумулятор. В случае 8-разрядных адресаций используется R0 или R1 текущего регистрового банка, а для 16-разрядных - регистр DPTR. Сигналы чтения и записи (\overline{RD} и \overline{WR}) активизируются только во время выполнения команд MOVX.

В табл.6 отражены две команды, предназначенные для чтения табличных данных, размещенных в программной памяти.

Таблица 6.

Команды для работы с таблицами.

Мнемоника	Операция
MOVC A, @A+DPTR	Read Program Memory at (A+DPTR)
MOVC A, @A+PC	Read Program Memory at (A+PC)

С помощью этих команд, осуществляющих доступ исключительно к памяти программ, возможно только чтение таблиц, но не их изменение, мнемонически эти команды выглядят как MOVC ("move constant"). Если таблица расположена во внешней программной памяти, то чтение байта из нее сопровождается стробом \overline{PSEN} (Program Store Enable).

Первая команда MOVC предназначена для обращения к таблице с максимальным числом входов - до 156 (от 0 до 255). Номер требуемого входа в таблицу загружается в аккумулятор, а регистр DPTR устанавливается на точку начала таблицы. Вторая команда MOVC работает аналогичным образом за исключением того, что в качестве указателя базы используется счетчик команд - PC и обращение к таблице производится из подпрограммы. Вначале номер требуемой точки входа загружается в аккумулятор, затем вызывается подпрограмма:

```
MOV A, Entry Number
CALL Table
```

Подпрограмма "Table" будет выглядеть следующим образом:

```
Table: MOVC A, @A+PC
RET
```

Таблица должна находиться в памяти программ непосредственно за инструкцией RET, причем такая таблица может иметь до 255 точек входа (от 1 до 256). Номер 0 не может быть использован, потому что во время выполнения инструкции MOVC A, @A+PC счетчик команд содержит адрес инструкции RET, и значением точки входа "0" будет сам код этой инструкции.

2.5.2. Команда обработки данных.

Команды обработки данных включают в себя арифметические, логические и булевские (битовые) операции. Перечень арифметических операций дан в табл.7.

Таблица 7.

Арифметические операции.

Мнемоника	Операция	Режимы адресации			
		Dir	Ind	Reg	Imm
ADD A, <byte>	A=A+<byte>	x	x	x	x
ADDC A, <byte>	A=A+<byte>+C	x	x	x	x
SUBB A, <byte>	A=A-<byte>-C	x	x	x	x
INC A	A=A+1	Accumulator only			
INC <byte>	<byte>=<byte>-1	x	x	x	
INC DPTR	DPTR=DPTR+1	Data Pointer only			
DEC A	A=A-1	Accumulator only			
DEC <byte>	<byte>=<byte>-1	x	x	x	
MUL AB	B:A=B⊙A	ACC and B only			
DIV AB	A=Int[A/B], B=Mod[A/B]	ACC and B only			
DA A	Decimal Adjust	Accumulator only			

Логические операции являются поразрядными. Их список приведен в табл.8.

Микросхемы MCS-51 содержат в своем составе "булевый" процессор, который можно рассматривать как независимый процессор побитовой обработки, имеющий свое побитово-адресуемое ОЗУ, свой ввод-вывод и выполняющий свой набор команд. Внутреннее ОЗУ имеет 128 прямоадресуемых бит, пространство регистров SFR также поддерживает до 128 битовых полей. Побитно-адресуемыми являются все порты ввода-вывода, каждая линия которых может рассматриваться как однобитовый порт.

Таблица 8.

Логические инструкции.

Мнемоника	Операция	Режимы адресации			
		Dir	Ind	Reg	Imm
ANL A, <byte>	A=A.AND.<byte>	x	x	x	x
ANL <byte>, A	<byte>=<byte>.AND.A	x			
ANL <byte>, #data	<byte>=<byte>.AND.#data	x			
ORL A, <byte>	A=A.OR.<byte>	x	x	x	x
ORL <byte>, A	<byte>=<byte>.OR.A	x			
ORL <byte>, #data	<byte>=<byte>.OR.#data	x			
XRL A, <byte>	A=A.XOR.<byte>	x	x	x	x
XRL <byte>, A	<byte>=<byte>.XOR.A	x			
XRL <byte>, #data	<byte>=<byte>.XOR.#data	x			
CLRA	A=00 _H	Accumulator only			
CPLA	A=.NOT.A	Accumulator only			
RLA	Rotate ACC Left 1 bit	Accumulator only			
RLCA	Rotate Left through Carry	Accumulator only			
RRA	Rotate ACC Right 1 bit	Accumulator only			
RRCA	Rotate Right through Carry	Accumulator only			
SWAPA	Swap Nibbles in A	Accumulator only			

Набор булевых инструкций перечислен в табл. 9.

Каждый из отдельно адресуемых бит может быть установлен в "1", сброшен в "0", инвертирован, передан в разряд "Carry" или принят из него. Между любым прямоадресуемым битом и флагом переноса могут быть произведены логические операции "И" и "ИЛИ". Кроме того, по результату анализа состояния адресуемого бита возможно осуществление короткого (Short) условного перехода в соответствии с байтом смещения (relative).

Таблица 9.

Команды побитовой обработки.

Мнемоника	Операция
ANL C, bit	C=C.AND.bit
ANL C, /bit	C=C.AND.(.NOT.bit)
ORL C, bit	C=C.OR.bit
ORL C, /bit	C=C.OR.(.NOT.bit)
MOV C, bit	C=bit
MOV bit, C	bit=C
CLRC	C=0
CLR bit	bit=0
SETB C	C=1
SETB bit	bit=1
CPL C	C=.NOT.C
CPL bit	bit=.NOT.bit
JC rel	Jump if C=1
JNC rel	Jump if C=0
JB bit, rel	Jump if bit=1
JNB bit, rel	Jump if bit=0
JBC bit, rel	Jump if bit=1; CLR bit

2.5.3. Команды передачи управления.

В табл.10 дан список команд безусловных переходов.

В приведенной таблице показана только одна команда "JMP addr", хотя фактически их имеется 3 варианта - SJMP, LJMP и AJMP, различающихся форматом адреса перехода. Мнемоника JMP используется в том случае, когда программист не уверен, какой вариант необходимо применить и предоставляет выбор Макроассемблеру.

Таблица 10.

Команды безусловных переходов.

Мнемоника	Операция
JMP addr	Jump to addr
JMP @A+DPTR	Jump to A+DPTR
CALL addr	Call subroutine at addr
RET	Return from subroutine
RETI	Return from interrupt
NOP	No operation

Для короткого перехода (Short JMP) адрес перехода определяется смещением (второй байт команды), при этом "дальность" перехода ограничена диапазоном -128/+127 байт относительно инструкции, следующей за JMP.

В инструкции длинного перехода (Long JMP) используется адрес назначения в виде 16-битной константы, что дает возможность перехода по любому адресу из 64К памяти программ.

Команда AJMP (Absolute JMP) является двухбайтной и использует 11-битную константу адреса. При выполнении инструкции младшие 11 бит PC замещаются 11-битным адресом из команды, старшие 5 бит остаются неизменными.

Инструкция JMP @A+DPTR предназначена для выполнения перехода "по выбору". Адрес назначения вычисляется как сумма значения регистра DPTR и аккумулятора. Обычно DPTR устанавливается на адрес таблицы переходов, а аккумулятор содержит индекс этой таблицы.

Команда "CALL addr" (аналогично команде "JMP addr") имеет две формы - LCALL и ACALL. Инструкция LCALL использует 16-битный адресный формат, ACALL - 11-битный.

Подпрограмма завершается инструкцией RET, позволяющей вернуться к инструкции, следующей за командой CALL.

Инструкция RETI используется для возврата из обработчиков прерываний, причем при выполнении RETI управляющая система уведомляется о том, что обработка завершилась.

В табл.11 дан список условных переходов.

Таблица 11.

Инструкции условных переходов.

Мнемоника	Операция	Режимы адресации			
		Dir	Ind	Reg	Imm
JZ rel	JMP if A=0	Accumulator only			
JNZ rel	JMP if A \neq 0	Accumulator only			
DJNZ <byte>, rel	Decrement and Jump if not zero	x		x	
CJNE A, <byte>, rel	JMP if A \neq <byte>	x			x
CJNE <byte>#data,rel	JMP if <byte> \neq #data		x	x	

Все условные переходы определяют адрес назначения как относительное смещение (rel) с длиной перехода, находящейся в пределах от -128 до +127 байт (относительно инструкции, следующей за условным переходом).

В PSW отсутствует флаг нуля, поэтому инструкции JZ и JNZ проверяют условие "равен нулю" тестированием данных в аккумуляторе.

Инструкция DJNZ (Decrement and Jump if Not Zero) используется для организации циклов.

Инструкция CJNE (Compare and Jump Not Equal) также может использоваться для управления циклами. Другим применением данной инструкции является проверка условий "больше чем", "меньше чем". Два байта в поле операндов представлены как "беззнаковое целое". Если первый операнд меньше, чем второй, то бит переноса "C" устанавливается в "1"; если больше или равен, то флаг "C" очищается.

Коды команд MCS-51 приведены в приложении.

3. УЧЕБНЫЙ МИКРОПРОЦЕССОРНЫЙ КОНТРОЛЛЕР УМПК-51

3.1. Описание УМПК-51

УМПК-51 предназначен для изучения архитектуры, системы команд, порядка функционирования и методов программирования ОЭВМ КР1816ВЕ51, а также для исследования методов сопряжения микропроцессорных систем (на основе MCS-51) с внешними цифровыми и аналоговыми устройствами [5].

Кроме того, УМПК-51 может быть использован при отладке программного и аппаратного обеспечения вновь разрабатываемых контроллеров.

Резидентное программное обеспечение и встроенные средства управления и индикации модуля обеспечивают пользователю:

- обмен информацией с инструментальной ЭВМ типа IBM PC;
- запуск программ в реальном масштабе времени и в пошаговом режиме;
- аналого-цифровое и цифроаналоговое преобразование;
- ввод информации (программ и данных) вручную (с помощью функциональной и цифровой клавиатуры и посредством ключей входного регистра) и аппаратно-программный по последовательному каналу связи (в стандарте RS-232);
- просмотр с помощью средств визуального отображения информации, а также изменение содержимого: ячеек внешней и внутренней памяти, программно-доступных регистров и битов битового процессора.

Структурная схема УМПК-51 показана на рис.11.

Схема включает в себя следующие функциональные блоки:

- 1 ⌚ блок ОЭВМ К1816ВЕ51;
- 2 ⌚ селектор адресов;
- 3 ⌚ блок клавиатуры и дисплея;
- 4 ⌚ блок внешней памяти программ и данных;
- 5 ⌚ блок ЦАП и АЦП;
- 6 ⌚ блок последовательного интерфейса (RS-232);
- 7 ⌚ имитатор внешних устройств.

Размещение на плате модуля органов управления, элементов индикации и функциональных блоков микроЭВМ показано на рис.12.

В блок ОЭВМ входят: ОЭВМ, буфер магистрали данных, регистр адреса, буфер магистрали управления и схема синхронизации и сбросов. Тактовая частота работы однокристалльной микроЭВМ определяется резонансной частотой кварцевого резонатора и равна 6 мГц. Начальная установка микроЭВМ осуществляется нажатием на клавишу "RS" (Reset).

Запрос прерывания на вход $\overline{INT0}$ может быть подан от следующих источников: нажатием кнопки "IO" или подачей нулевого уровня на вход $\overline{INT0}$ от внешнего устройства. Запрос прерывания по входу $\overline{INT1}$ поступает от контроллера клавиатуры и дисплея КР580ВВ79.

Буфер данных КР580ВА86 служит для организации резидентной магистрали данных микроЭВМ. Буфер переключается на ввод данных сигналами \overline{PSEN} или \overline{RD} при чтении команд или данных.

Регистр адреса КР580ИР82 "защелкивает" младший байт адреса по сигналу ALE.

Буфер магистрали управления реализован на К555ЛП8.

Селектор адресов формирует сигналы выборки для следующих устройств:

- четыре выборки на ИМС ОЗУ (КР537РУ8) и ПЗУ (К573РФ2) объемом по 2 Кбайт каждое. Выборки на микросхемы ПЗУ формируются при обращении по адресам $0000_H \dots 07FF_H$ и $0800_H \dots 0FFF_H$, выборки на микросхемы ОЗУ - при обращении по адресам $1000_H \dots 17FF_H$ и $1800_H \dots 1FFF_H$;
- выборку для контроллера клавиатуры и дисплея при обращении к ВУ по адресам 8400_H и 8401_H ;
- строб записи в регистр ЦАП (КР580ИР82) при записи информации по адресу 8100_H ;
- строб записи в триггер запуска АЦП (К555ТМ2) при записи по адресу 8300_H ;
- выборку чтения буфера АЦП (КР580ВА86) при чтении по адресу 8000_H ;

- выборку чтения бита готовности АЦП (бит D1) и выхода компаратора (бит D0) при чтении по адресу 8200_H.

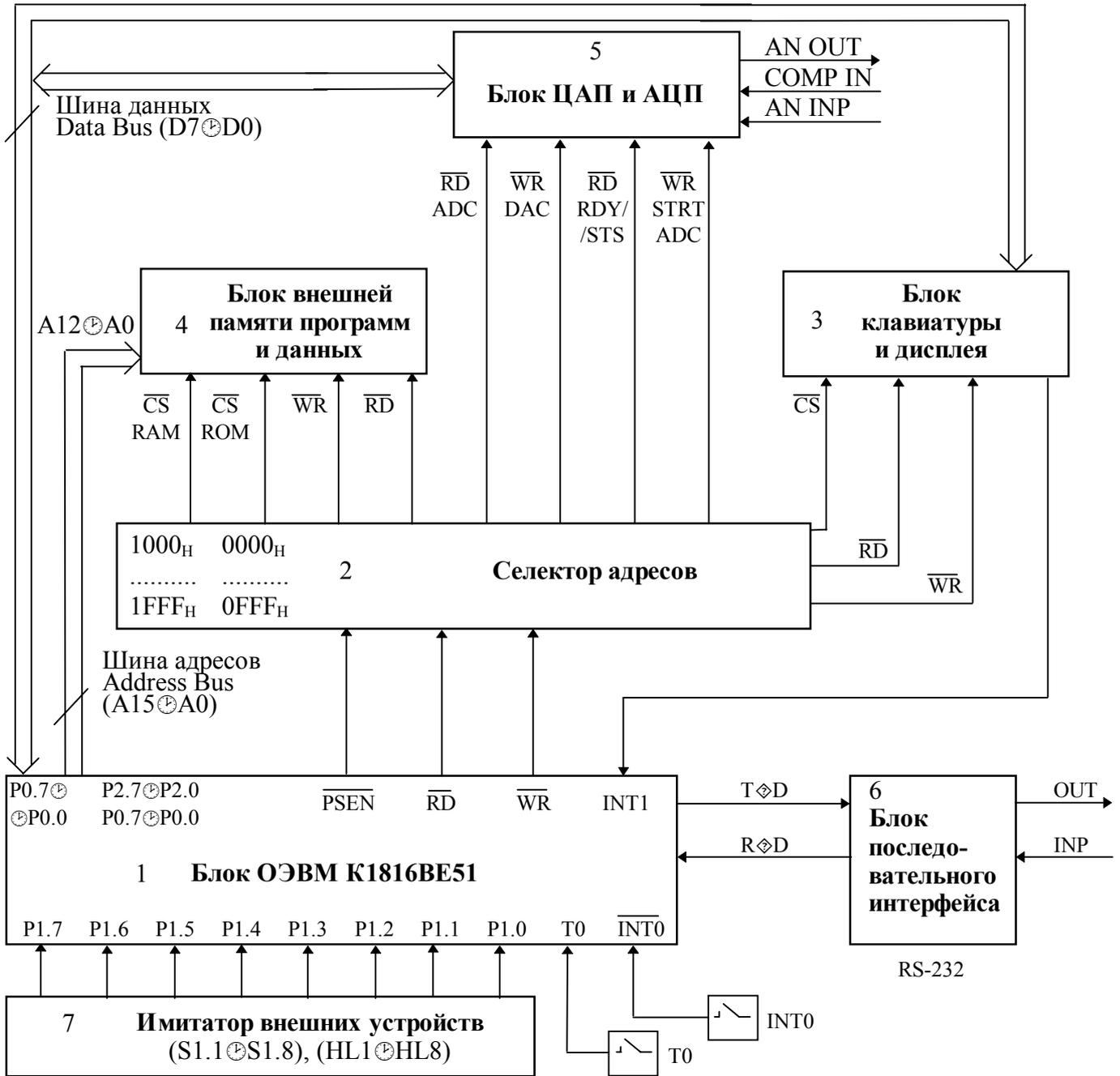


Рис.11. Структурная схема УМПК-51

Блок клавиатуры и дисплея построен на основе КР580ВВ79 по стандартной схеме включения с внешним дешифратором (К155ИД4). При нажатии клавиши на выходе \overline{IRQ} контроллера формируется сигнал прерывания, который подается на вход $\overline{INT1}$ ОЭВМ.

Блок ПЗУ содержит управляющую программу-монитор, выборка кода из блока памяти команд осуществляется сигналами \overline{PSEN} или \overline{READ} при чтении команд или данных по адресам 0000_H...0FFF_H. Чтение ОЗУ осуществляется посредством этих же сигналов, что позволяет использовать ОЗУ как в качестве памяти команд, так и в качестве памяти данных. ОЗУ занимает адресное пространство 1000_H...1FFF_H.

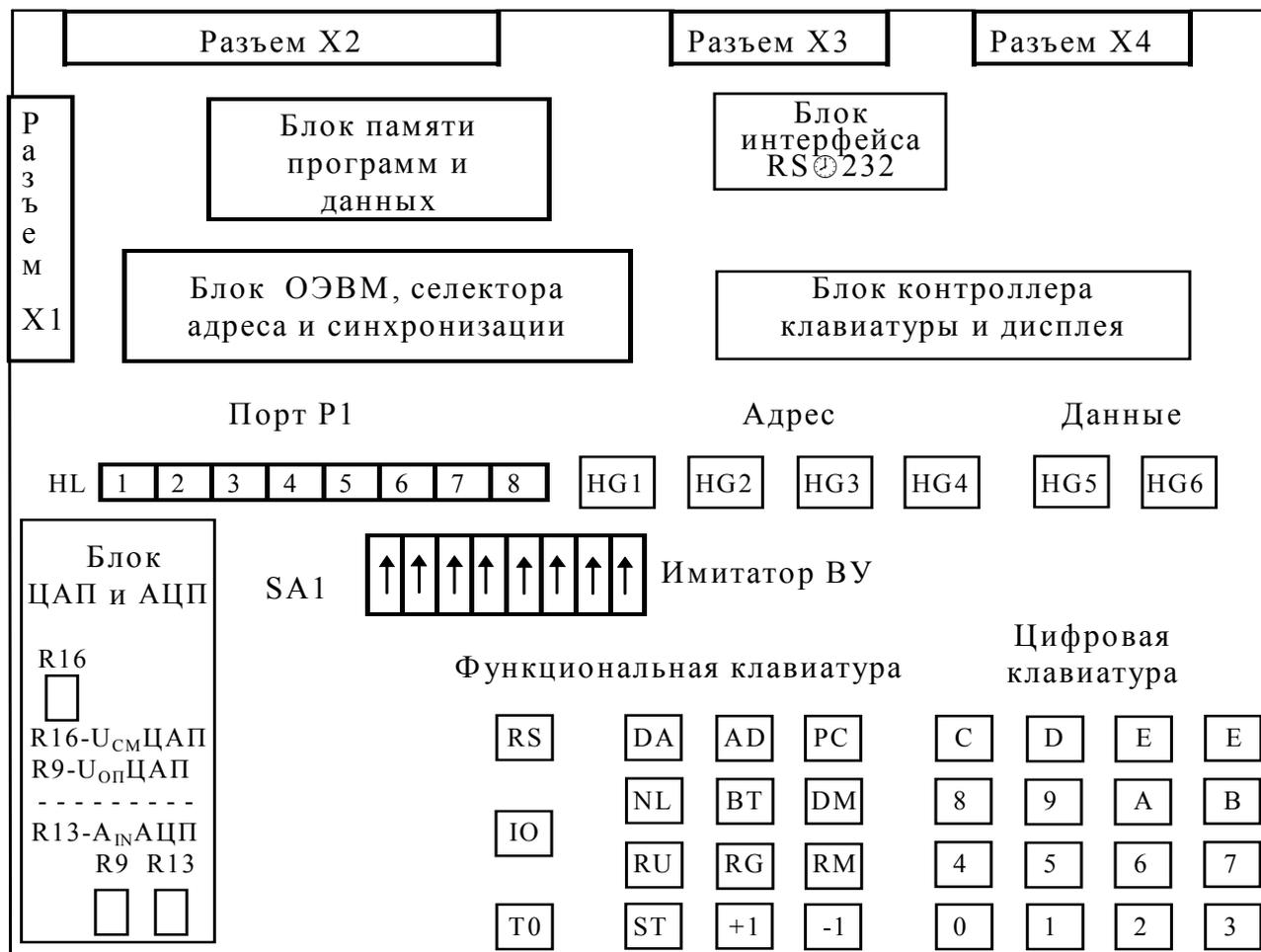


Рис.12. Размещение на плате УМПК-51 органов управления, элементов индикации и функциональных блоков микроЭВМ

- Примечания. X1, X2 - разъемы для подключения внешних устройств.
 X3 - разъем для сопряжения модуля с инструментальной ЭВМ.
 X4 - разъем для подключения источника питающих напряжений.

Блок ЦАП построен на основе КР572ПА1 и реализует преобразование 8-разрядного кода, записываемого в регистр (КР580ИР82), а аналоговые напряжения в диапазоне от -10,24 В до +10,16 В.

Блок АЦП построен на основе К1113ПВ1 и реализует преобразование внешнего аналогового напряжения от -10,24 В до +10,16 В в цифровую форму.

Блок последовательного интерфейса преобразует выходной сигнал ОЭВМ - T \diamond D с уровнем -15 В/+15 В, а также преобразует сигнал с линии RS-232 в сигнал с ТТЛ-уровнем, подаваемый на вход R \diamond D ОЭВМ.

Схема имитатора ВУ включает в себя переключатели SA1.1...SA1.8, клавиши T0 и IO и светодиоды HL1...HL8 и предназначена для формирования воздействий на микроЭВМ и отображения состояния порта P1. Кодовая комбинация, набранная с помощью переключателей, может быть считана через порт P1. Светодиоды HL1...HL8 отображают состояние выходов порта P1 (при разомкнутых переключателях SA1). Клавиша T0 используется при изучении работы внутреннего таймера/счетчика событий, а клавиша IO предназначена для формирования сигнала запроса прерывания на вход INT0 микроЭВМ.

3.2. Порядок работы с УМПК-51.

Подключить кабель от источника питания к разъему X4 микроЭВМ и подать питающие напряжения (включить источник питания).

Провести начальную установку микроЭВМ нажатием на клавишу "RS" (возможна автоматическая начальная установка микроЭВМ при включении питания). Начальная установка модуля может быть проведена как с инициализацией (обнулением) содержимого памяти, так и без инициализации в зависимости от положения кнопки "IO" в момент нажатия кнопки "RS". В нажатом положении кнопки "IO" проводится инициализация ОЗУ, в отпущенном - не проводится. При начальной установке выдается звуковой сигнал и на дисплей выводится сообщение "rEADY", после чего модуль готов к приему директив резидентного монитора. Директивы вводятся нажатием на "PM", "PC", "DM", "RU", "ST", "BT", "RG", "NL", "AD", "DA" с последующим вводом требуемых параметров.

3.2.1. Режим просмотра и изменения содержимого внешней памяти программ.

3.2.1.1. После нажатия на клавишу "PM" ("Program memory") в первом, втором, третьем и четвертом разрядах дисплея загораются нижние сегменты, означающие возможность ввода адреса памяти. Адрес ячейки вводится последовательным нажатием четырех цифровых клавиш (правое наборное поле клавиатуры), после чего на дисплей выводится адрес открытой ячейки памяти команд и ее содержимое.

3.2.1.2. Просмотр памяти команд вперед осуществляется с помощью клавиши "+1" ("Запись/Увеличить"), просмотр назад - нажатием клавиши "-1" ("Уменьшить"). Ввод нового значения ячейки по выбранному адресу производится нажатием на соответствующие цифровые клавиши. При этом в шестом разряде дисплея загорается запятая, являющаяся признаком записи. При вводе новой цифры число на дисплее отодвигается на один разряд влево, что позволяет не набирать незначимые старшие нули и корректировать неправильно набранные значения. Введенное значение записывается в память команд клавишей "+1", при этом осуществляется автоматический переход к следующей ячейке. Отказ от записи набранного значения может быть произведен с помощью клавиши "-1" или вводом новой директивы.

Примечание. Не рекомендуется производить запись в ячейки с адресами 1750_H...17FF_H, поскольку они используются управляющей программой.

3.2.2. Вывод на дисплей содержимого программного счетчика.

При нажатии на клавишу "PC" ("Program Counter") на дисплей выводится значение программного счетчика на момент последнего выхода из программы пользователя и содержимое памяти команд по этому адресу. После этого возможен просмотр и изменение содержимого памяти аналогично вышеописанному (п.3.2.1.2).

При начальной установке значение программного счетчика устанавливается равным 1000_H.

3.2.3. Просмотр и изменение содержимого внутренней памяти данных.

Для ввода адреса внутренней памяти данных необходимо нажать клавишу "DM" ("Data memory"), при этом на дисплей выводится сообщение

A ⊙ ☿☿ ☿☿ ,

затем необходимо ввести две цифры требуемого адреса, после чего на дисплее индицируется адрес и содержимое ячейки внутренней памяти данных. Просмотр и изменение содержимого памяти данных осуществляется аналогично п.3.2.1.2.

Примечание. При вводе адреса, превышающего 7FH, доступ осуществляется непосредственно к регистрам SFR (Special Function Registers).

3.2.4. Просмотр и изменение содержимого битов битового процессора.

Для ввода адреса бита необходимо нажать клавишу "BT" ("Bit"). На дисплей выводится сообщение

b 00 00 ,

затем необходимо ввести две цифры требуемого адреса, при этом на дисплее индицируется адрес и содержимое бита (0 или 1). Просмотр и изменение содержимого битов осуществляется аналогично п.3.2.1.2.

Примечание. При вводе адреса, превышающего $7F_H$, доступ осуществляется непосредственно к регистрам SFR.

3.2.5. Просмотр и изменение содержимого внутренних регистров ОЭВМ.

После нажатия на клавишу "RG" ("Registers") на дисплей выводится мнемоническое обозначение регистра и его содержимое в следующем порядке:

Acc	⌚	аккумулятор;
B	⌚	регистр B;
FL	⌚	регистр флагов (PSW);
DPH	⌚	старший байт регистра DPTR (Data Pointer);
DPL	⌚	младший байт регистра DPTR;
SP	⌚	указатель стека SP;
PCN	⌚	старший байт программного счетчика;
PCL	⌚	младший байт программного счетчика;
b0r0	⌚	регистр R0 нулевого банка регистров;
.....		
b3r7	⌚	регистр R7 третьего банка регистров.

Просмотр содержимого регистров вперед в указанном порядке осуществляется с помощью клавиши "+1", назад - нажатием "-1", изменение содержимого выбранного регистра может быть произведено набором требуемого значения с помощью цифровых клавиш и нажатием на клавишу "+1".

3.2.6. Запуск программы пользователя в автоматическом или пошаговом режиме.

После ввода программы пользователя (п.3.2.1) и, если необходимо предварительной установки внутренней памяти (п.3.2.3), битов (п.3.2.4) и регистров (п.3.2.5) может быть произведен запуск программы пользователя в одном из двух режимов: автоматическом или пошаговом. Для запуска программы пользователя необходимо предварительно ввести стартовый адрес программы с помощью директив "PM" или "PC".

3.2.6.1. Запуск программы в пошаговом режиме осуществляется нажатием на клавишу "ST" ("Step"). При этом выполняется одна команда и на дисплей выводится следующее значение программного счетчика.

3.2.6.2. Запуск программы в автоматическом режиме осуществляется нажатием на клавишу "RU" ("Run").

Выход из программы пользователя после запуска в автоматическом режиме производится нажатием на любую клавишу. При этом сохраняются внутренняя память данных и регистры ОЭВМ и на дисплей выводится адрес останова. В этом случае возможен просмотр всех необходимых программно-доступных ресурсов ОЭВМ и соответственно результатов выполнения программы. Продолжить выполнение программы с адреса останова можно, выполнив директивы "PC" и "RU" или "ST".

3.2.7. Аналоговый вывод.

После нажатия на клавишу "DA" ("Digital-to-analog") на дисплей выводится сообщение

A out 80 ,

это означает, что в буферный регистр (KP580IP82) аналоговой подсистемы предлагается записать код 80_H, который сформирован в регистре R4 внутренней памяти данных и соответствует нулевому выходному напряжению. Выбор произвольного кода осуществляется цифровыми клавишами, запись кода (передача из R4 в буферный регистр) выполняется нажатием клавиши "+1".

Соответствие кода и выходного напряжения приведено ниже:

00_H ☉ -10,24 В;

01_H ☉ -10,16 В;

.....

7F_H ☉ -0,08 В;

80_H ☉ -0,00 В;

81_H ☉ +0,08 В;

.....

FE_H ☉ +10,08 В;

FF_H ☉ +10,24 В;

3.2.8 Аналоговый ввод.

При нажатии на клавишу "AD" ("Analog-to-digital") на дисплей выводится сообщение

☉ A in XX ,

где XX - шестнадцатеричный код (двоичный смещенный дополнительный код) входного сигнала в соответствии с вышеприведенной таблицей.

Следующее преобразование индицируется нажатием на "+1" и так далее. Нажатие на клавишу "C" ("Cycle") включает циклический режим, при котором преобразование входной аналоговой величины производится автоматически и информация на дисплее будет периодически обновляться.

Нажатие на клавишу "D" ("Decimal") приводит к отображению информации в десятичном формате (от -10,24В до +10,16В), превращая УМПК-51 в цифровой вольтметр. В режиме десятичного отображения также может быть включен циклический режим (клавишей "C").

Порядок выбора режимов "C" и "D" произвольный и эти режимы могут быть отменены клавишей "-1".

3.2.9. Организация обмена с инструментальной ЭВМ.

Выбор режима работы с последовательным интерфейсом осуществляется клавишей "NL" и индицируется сообщением

rS ☉232,

Для организации этого режима инструментальная ЭВМ должна иметь последовательный канал связи типа RS-232C. Вид обмена - асинхронный, длина слова - 8 бит, без контроля четности, 2 стоп-бита, скорость обмена - 2400 бод. После запуска программы обмена и вывода сообщения (rs-232) УМПК-51 ожидает управляющую посылку от инструментальной ЭВМ. Формат управляющей посылки следующий:

первый байт ☉ направление обмена;

- 52_H ☉ передача из ОЗУ УМПК-51 в ЭВМ;

- 57_H ☉ передача из ЭВМ в ОЗУ УМПК-51;

второй и третий байты ☉ начальный адрес ОЗУ (Addr_L, Addr_H);

четвертый и пятый байты ☉ конечный адрес ОЗУ (Addr_L, Addr_H);

шестой байт ☉ первая контрольная сумма (CRC - Cyclic Redundancy Check).

В случае безошибочного приема управляющей посылки УМПК-51 выдает байт ответа (06_H) и начинает прием или передачу данных в указанных адресах. Данные оканчиваются второй контрольной суммой (CRC').

В случае безошибочного приема данных УМПК-51 выдает байт ответа (86_H) и выходит из подпрограммы, что индицируется сообщением

⊙ rEADY.

Примечание. Во время приема (передачи) на дисплей УМПК-51 выводится адрес текущей ячейки ОЗУ в формате

XXXX ⊙ ⊙

3.3. Подсистема аналогового вывода УМПК-51.

На рис.13 изображена функциональная схема подсистемы аналогового вывода на основе использования 10-разрядного интегрального ЦАП К572ПА1, подключенного через буферный регистр К580ИР82 к шине данных УМПК-51. Метод преобразований, используемый в ИС К572ПА1, предполагает суммирование в соответствии с заданным значением двоичного кода всех разрядных токов, взвешенных по двоичному закону. Распределение двоичных токов на токовые выходы I₁, I₂ ЦАП осуществляется с помощью двухпозиционных токовых ключей таким образом, что единичное значение какого-либо разряда определяет ток этого разряда в составе выходного тока I₁, а нулевое значение разряда определяет соответствующую составляющую тока на выходе I₂. Стандартное включение операционного усилителя Y1.1, показанное на рис.13, имеет целью преобразовать тип аналоговой величины: выходной ток I₁ в напряжение U₁. Напряжение U₁ при этом описывается выражением, приведенном в [9 с.51].

$$U_1 = -\frac{U_R R_{OC}}{2^n R} \sum_{j=1}^n \beta_j 2^{n-j} = -\frac{U_R}{2^n} \sum_{j=1}^n \beta_j 2^{n-j}, \quad (4.1)$$

в котором U_R - напряжение источника опорного напряжения (ИОН); R - эквивалентное сопротивление резистивной матрицы R-2R ЦАП (R ⊙ 10кОм); R_{OC}=R - сопротивление, размещенное на кристалле микросхемы и являющееся сопротивлением обратной связи усилителя Y1.1; β_j - двоичное значение j-го разряда (увеличение номера индекса уменьшает старшинство разряда во входном коде); n - разрядность ЦАП (n=10).

Изменив нумерацию индексов на обратную β_j = a_{N-j} и приняв N-j=i, приведем формулу (4.1) в более привычном виде, когда старшему разряду a_i соответствует больший номер в индексе

$$U_1 = -\frac{U_R}{2^n} \sum_0^{n-1} a_i \cdot 2^i = -\frac{U_R}{2^n} N, \quad (4.2)$$

где N - 10-разрядное беззнаковое число.

Рассмотренный фрагмент схемы обеспечивает так называемое униполярное преобразование (с обратным знаком) для беззнаковых цифровых кодов, содержащих только цифровые двоичные разряды. Однако на практике часто требуется биполярное преобразование для чисел, представленных дополнительным кодом. Правда, чтобы не вносить каких-либо изменений в структуру униполярного преобразователя К572ПА1 (и ему подобных) дополнительный код числа N_{доп} заменен специальной модификацией - дополнительном кодом с инверсным знаковым разрядом или смещенным дополнительным кодом N_{доп.см}.

Существуют два способа преобразования с использованием N_{доп.см}, различающихся друг от друга точностными характеристиками (точнее, стабильностью нулевого уровня). В УМПК-51 использована более простая схема, применение которой ограничивается разрядностью преобразуемого кода на уровне 8+10 бит.

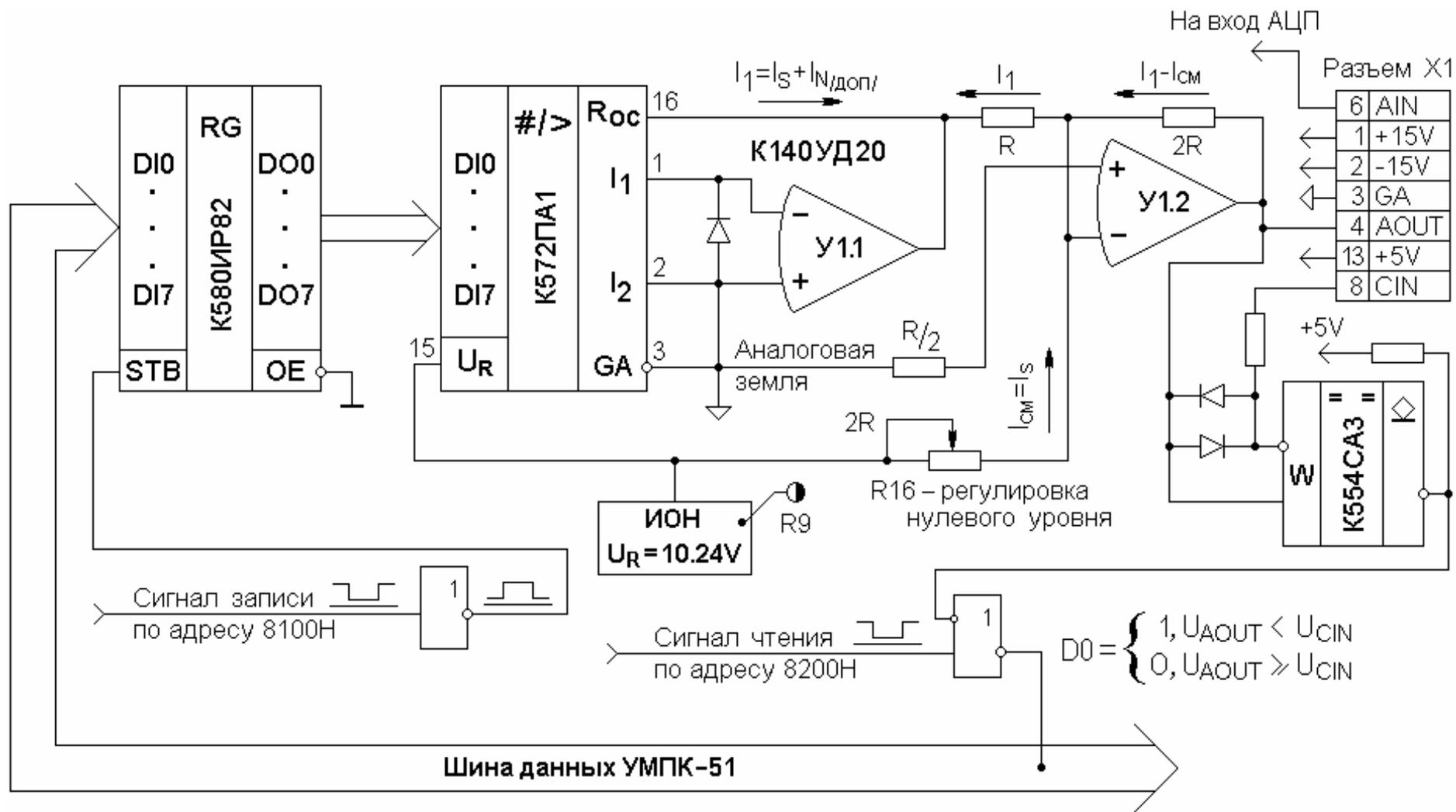


Рис.13. Функциональная схема подсистемы аналогового вывода

Проведем анализ предложенной схемы биполярного ЦА-преобразования, разделив при этом выходное напряжение U_1 на две части: составляющую, обусловленную действием знакового разряда S (управляет старшим разрядным ключом ЦАП), и составляющую, создаваемую всеми цифровыми разрядами или модулем смещенного дополнительного кода

$$\begin{aligned} |N_{\text{доп.см}}| &= |N_{\text{доп}}| = \sum_0^{n-2} a_i \cdot 2^i ; \\ U_1 &= U_{18} + U_{|N_{\text{доп}}|} = -\frac{U_R}{2^n} \cdot (S \cdot 2^{n-1} + |N_{\text{доп}}|) \end{aligned} \quad (4.3)$$

здесь S - значение знакового разряда $N_{\text{доп.см}}$.

Напряжение $U_{\text{вых}}$ суммирующего усилителя $Y1.2$ с учетом указанных на схеме значений входных сопротивлений усилителя

$$U_{\text{вых}} = -2U_1 - U_R. \quad (4.4)$$

С учетом (4.4) выражение (4.3) приводится к виду

$$U_{\text{вых}} = U_R \cdot \left(S - 1 + \frac{|N_{\text{доп}}|}{2^{n-1}} \right). \quad (4.5)$$

Из последнего соотношения следует, что дополнительное смещение уровня выходного усилителя, введенное через сопротивление R_{16} , равно по своему значению действию знакового разряда (разумеется, когда $S=1$). Кстати, эта особенность схемы и объясняет нестабильность нулевого уровня ЦАП, так как при $|N_{\text{доп}}| = 0$ нулевое значение $U_{\text{вых}}$ создается вычитанием значительных величин, каждая из которых характеризуется своим температурным коэффициентом изменения.

Приведем выражение (4.5) к более удобному для практических целей виду. Используя очевидные соотношения

- а) $N \geq 0, S = 1, |N_{\text{доп}}| = |N|$;
 б) $N < 0, S = 0, |N_{\text{доп}}| = 2^{n-1} - |N|$,

получим в окончательном виде

$$U_{\text{вых}} = U_{\text{AOUT}} = \frac{U_R}{2^{n-1}} (-1)^{\bar{S}} |N|, \quad \text{где } |N| = \sum_0^{n-2} a_i \cdot 2^i \quad (4.6)$$

На рис.14 представлена выходная характеристика биполярного ЦА-преобразования, а в табл.12 - соответствие между входным кодом $N_{\text{доп.см}}$ и аналоговой величиной U_{AOUT} для случая $n=8$ и $U_R=10,24\text{В}$.

Изготовителем УМК-51 не приведены технические характеристики для подсистемы аналогового ввода/вывода, однако проведенный выше анализ системы и собственные технические характеристики интегрального ЦАП 572ПА1 позволяют сделать некоторые заключения.

1. Рассматриваемый способ биполярного преобразования а основе использования униполярного ЦАП 572ПА1 уменьшает разрядность ЦА-преобразования до величины $n-1$.

2. Байтный формат данных МК-51 предопределил в качестве "лишних" два младших цифровых входа, уменьшив тем самым фактическую разрядность ЦА-преобразования до 7, что определило разрешающую способность R (или ступень квантования h на рис.14), равной

$$R = \frac{U_{\text{шк}}}{2^7 - 1} = 0,08\text{В} \quad (80 \text{ мВ}),$$

что соответствует 0,78% от полной шкалы $U_{\text{шк}} = 10,16 \text{ В}$.

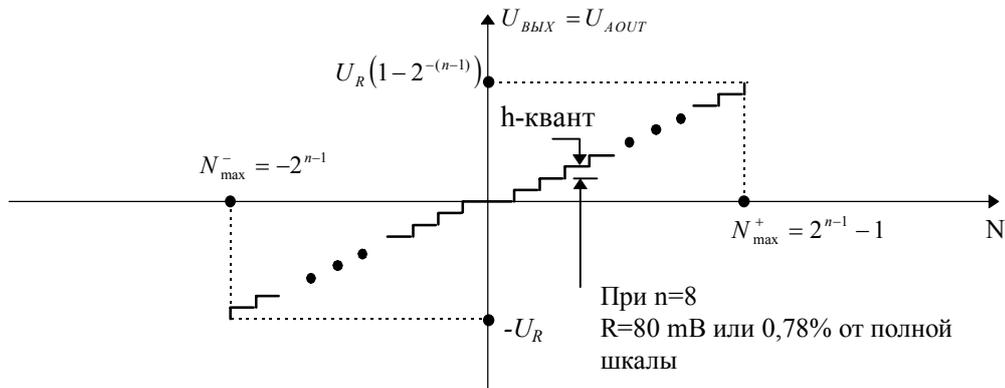


Рис.14. Выходная характеристика биполярного ЦАП (идеальный вариант)

Таблица 12

Соответствие между входным кодом $N_{ДОП.СМ}$ и аналоговым сигналом $U_{АОУТ}$ для $n=8$ и $U_R=10,24$ В

$N_{ДОП.СМ}$		$N_{ДОП}$	Десятичное число N	$U_{АОУТ} = \frac{U_R}{128} (-1)^S \cdot N $
S	$ N_{ДОП.СМ} = N_{ДОП} $			
1	111 1111 (FF _H)	0111 1111	127	10,16
1	111 1110 (FE _H)	0111 1110	126	10,08
1	111 1101 (FD _H)	0111 1101	125	10,00
	•			
	•			
	•			
1	000 0001 (81 _H)	0000 0001	1	0,08
1	000 0000 (80 _H)	0000 0000	0	0
0	111 1111 (7F _H)	1111 1111	-1	-0,08
	•			
	•			
	•			
0	000 0001 (01 _H)	1000 0001	-127	-10,16
0	000 0000 (00 _H)	1000 0000	-128	-10,24

Выбор в качестве "лишних" младших разрядов ЦАП имеет то преимущество (в сравнении с выбором в этом качестве двух старших разрядов), что основные статистические погрешности преобразования - нелинейность характеристики $\underline{\alpha}_L$ и дифференциальная нелинейность $\underline{\alpha}_{LD}$ - остаются неизменными, то есть определяются своими номинальными значениями для 10-разрядного ЦАП. Значения величин $\underline{\alpha}_L$ и $\underline{\alpha}_{LD}$ даются, как правило, в единицах ступени квантования, отнесенные к шкале, и для 10-разрядных ЦАП составляют величину $\pm 0,1 \oplus 0,15\%$.

3. Абсолютные погрешности ЦА-преобразования в начальной и конечной точках выходной характеристики в значительной степени компенсируются подстройкой сдвига нуля (резистор R16) и регулировкой крутизны характеристики преобразования - резистором R9. Фактическое значение этих погрешностей требует проведения специальных измерений.

4. Важнейшим из динамических параметров ЦАП является время установления t_s выходного сигнала в зоне $\pm h$ с момента изменения цифрового кода и составляет для токового выхода I_1 (I_2) ЦАП 572ПА1 величину, равную 5 мкс. Ясно, что последовательное включение двух операционных усилителей после ЦАП увеличивает этот показатель для выходного сигнала U_{AOUT} до уровня ≈ 10 мкс.

В заключение отметим некоторые особенности, присущие аналоговой подсистеме (рис.13).

Буферный регистр ЦАП К580ИР82 всегда находится в активном состоянии ($\overline{OE} = 0$), поэтому смена значений U_{AOUT} происходит одновременно с изменением содержимого регистра, которое осуществляется только в момент загрузки нового кода $N_{ДОП.СМ}$ командой передачи данных по адресу 8100 (MOVX @DPTR,A).

Напряжение ЦАП $U_{ВЫХ}$ выводится на четвертый контакт разъема X1 и, кроме того, подается на неинвертирующий вход компаратора К554СА3, что позволяет сравнивать выходное напряжение аналогового вывода U_{AOUT} с внешним напряжением $U_{СИН}$ (восьмой контакт разъема X1), подаваемым на инвертирующий вход компаратора. Состояние компаратора (логический "0" или логическая "1") можно опросить через буферный элемент К155ЛП8 как бит D0 байта, считываемого из внешней памяти по адресу 8200H.

3.4. Подсистема аналогового ввода УМПК-51.

Функциональная схема подсистемы аналогового ввода показана на рис.15 и структурно состоит из трех элементов: входного буферного каскада на усилителе Y2.1, преобразующего диапазон входного аналогового сигнала U_{AIN} ($\pm 10,24$ В) в диапазон изменения сигнала ($\pm 5,12$ В) на входе AI интегрального АЦП К1113ПВ1 (второй элемент) и шинного формирователя К580ВА86, который транслирует цифровой код на шину данных МК-51. 10-разрядный интегральный АЦП К1113ПВ1 относится к типу функционально законченного АЦП, содержащего в себе все узлы, необходимые для реализации АЦ-преобразования методом последовательных приближений [9]. Для лучшего понимания термина "функционально законченного" обратимся снова к подсистеме аналогового вывода. В ней для реализации ЦА-преобразования понадобилось использование, помимо ЦАП К572ПА1 с токовым выходом, двух операционных усилителей и источника опорного напряжения ИОН, в то время как интегральный АЦП К1113ПВ1 включает в себя биполярный ЦАП как одну из составных частей.

Переключение режима работы АЦП производится по входу V: если $V=0$, то преобразователь работает в униполярном режиме с входным сигналом $0...10,24$ В, а при "открытом" входе V (рис.15) АЦП работает в биполярном режиме ($U_{IN}=-5,12...+5,12$ В). В биполярном режиме преобразователя выходным кодом также является дополнительный смещенный код $N_{ДОП.СМ}$, как и в аналоговой подсистеме вывода.

В представленной на рис.15 схеме включения АЦП отсутствует только устройство регулировки нулевого уровня, формирующее сигнал на входе GA (земля аналоговая), что вызвано, очевидно, низкой разрешающей способностью АЦ-преобразования, оперирующего 8-разрядным кодом результата (два младших цифровых выхода не используются).

Технические характеристики аналоговой подсистемы ввода полностью определяются параметрами преобразователя К1113ПВ1, кроме шага квантования h , равном

$$h = \frac{5,12B}{127} = 40mB,$$

где $127 = 2^7 - 1$, 7 - число разрядов цифрового кода.

Другие характеристики:

Нелинейность $\underline{\alpha}_L$, %	$\pm 0,1...0,2$;
Дифференциальная нелинейность $\underline{\alpha}_{LD}$, %	$\pm 0,1...0,2$;
Напряжение смещения нуля на входе, мВ	± 30 ;
Время преобразования, t_C , мкс	30

Регулировка коэффициента преобразования (крутизна характеристики преобразования) осуществляется резистором R13 на входе буферного усилителя.

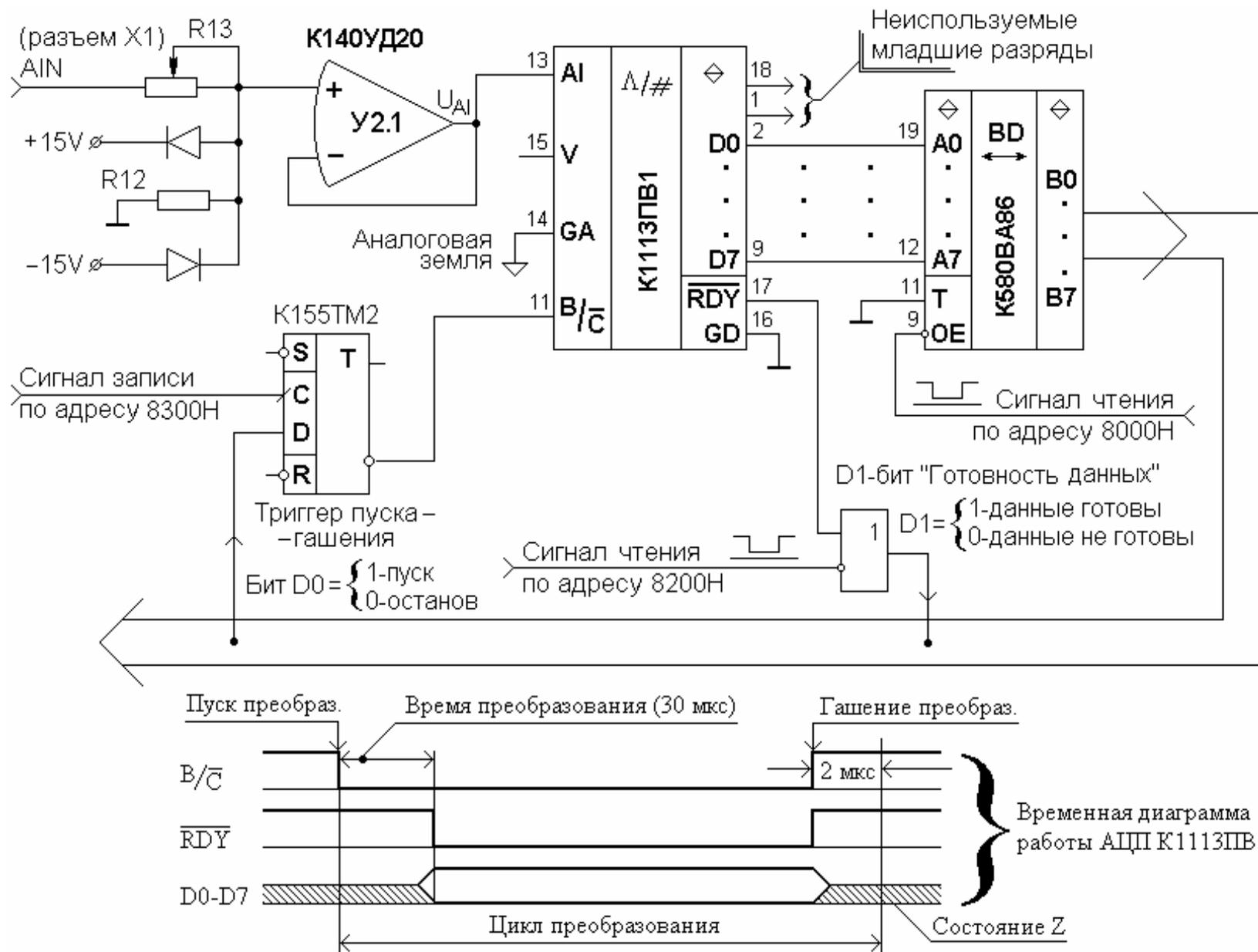


Рис.15. Подсистема аналогового ввода на основе АЦП К1113PB1 и временная диаграмма его работы

Процесс преобразования аналогового сигнала в код представлен на временной диаграмме (рис.15) и начинается по срезу сигнала на входе B/\overline{C} (гашение - преобразование). По окончании преобразования на выходе \overline{RDY} (готовность) появляется нулевой сигнал, одновременно с которым на цифровых выходах преобразователя устанавливается информация (результат преобразования), которая подается на входы шинного формирователя K580BA86. В этот интервал времени процессор МК-51 должен осуществить считывание информации с АЦП на шину данных путем перевода выходов буфера K580BA86 в активное состояние формированием сигнала $\overline{OE}=0$. Приняв данные с буфера, микропроцессор должен установить на входе B/\overline{C} уровень логической "1", который "гасит" информацию в регистре последовательных приближений и переводит цифровые выходы АЦП снова в состояние Z. Спустя задержку в 2 мкс, можно начать новый цикл преобразования.

Формирование управляющих сигналов временной диаграммы работы АЦП осуществляется командами передачи типа MOVX @DPTR, A по адресам внешней памяти данных, указанных на схеме рис.15. Значения битов аккумулятора также очевидны из работы АЦП и приведены в табл.13.

Таблица 13

Управляющие команды МК-51 для подсистемы аналогового ввода.

Действия процессора МК-51	Содержание командного байта в Асс	Адрес
Пуск преобразователя	Асс.0(D0)=1	8300H
Опрос готовности АЦП	Асс.1(D1)=0	8200H
Считывание данных с АЦП на шину данных	⌚	8000H
Гашение АЦП	Асс.0(D0)=0	8300H

3.5. Программное обеспечение организации обмена информацией через последовательный порт.

Прием/передачу информации через последовательный порт можно осуществить либо на основе опроса флагов УАПП (TI и RI), либо на основе программной обработки прерываний, вызванных этими флагами. В любом случае требуется как минимум три программных модуля:

1. Инициализация УАПП на определенный режим работы посредством задания управляющего слова в регистре SCON, а также настройка таймера/счетчика 1 на обеспечение требуемой скорости передачи.

2. Модуль загрузки данными буферного регистра передачи SBUF, а также бита TB8 регистра SCON (2 и 3 режимы).

3. Модуль считывания данных из буферного регистра приемника SBUF по мере их поступления и контроля правильности передачи/приема.

В целом ряде случаев, помимо указанных модулей, программа передающего МК может содержать блок вычисления или формирования данных, предназначенных для передачи, или блок обработки данных.

3.5.1. Инициализация УАПП и таймера/счетчика 1.

Практически важные значения управляющих слов регистра SCON приведены в табл.14, где дана и их краткая характеристика, достаточная для осуществления выбора.

Таблица 14

Управляющие слова регистра SCON,
используемые при инициализации УАПП

Режим работы	Биты регистра SCON								
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
	Управляющие слова, назначение и характеристики								
1	50 _H	⌚	только прием						
	42 _H	⌚	только передача						
	52 _H	⌚	передача/прием						
	72 _H	⌚	передача/прием. Используется для контроля занесения стоп-бита при приеме в бит RB8. Если в конце приема RB8=0, то флаг RI не устанавливается в "1".						
2 (3)	90 _H (D0 _H)	⌚	только прием)*						
	82 _H (C2 _H)	⌚	только передача						
	92 _H (D2 _H)	⌚	передача/прием)*						
)* После приема целесообразно провести контроль RB8=P, где P-бит паритета PSW.						
	B2 _H (F2 _H)	⌚	передача/прием. RI устанавливается в "1" по окончании приема только тогда, когда бит RB8 устанавливается в "1".						

Пример инициализации таймера/счетчика 1 с автоматической перезагрузкой при переполнении (режим 2) приведен ниже. Константы перезагрузки для регистра TH1 определяют частоту переполнения f_{OV} , которая синхронизирует передачу/прием битов УАПП. Значения этих констант для различной скорости приема/передачи приведены в табл.15. Ясно, что в этом режиме использования таймера/счетчика 1, флаг TF1 не должен вызывать прерывания, и они должны быть запрещены.

Пример использования таймера/счетчика 1 для работы в режиме 2

```

mov  PCON, #80H (00H) ; бит SMOD =1 (0)
mov  TMOD, #20H      ; T/C1 в режиме 2
mov  TL1, data8      ; константы перезагрузки
mov  TH1, data8      ; таймера/счетчика 1
CLR  ET1 (IE3)      ; запрет прерываний по флагу переполнения TF1 таймера/счетчика 1
SETB TR1 (TCON.6)  ; запуск таймера/счетчика на счет

```

Таблица 15

Настройка таймера 1 для управления скоростью работы УАПП

Режим УАПП	Скорость приема передачи, Кбод	SMOD	Перегружаемое число (TH1)=(TL1)	Примечание
2	187,5 93,75	1 0	Таймер/счетчик 1 не используется	Высокоскоростной асинхронный режим
1,3	31,250 15,625 10,416 7,812 6,250 2,403 1,201 0,222 0,110	1 1 1 1 1 1 1 1 0	FF _H FE _H FD _H FC _H FB _H F3 _H E6 _H 00 _H 72 _H	$f_T = 6$ мГц

3.5.2. Прием/передача на основе опроса флагов TI и RI.

Независимо от режима УАПП передача или прием начинается с опроса состояния соответствующего флага (TI или RI). Единичное значение флага TI говорит о готовности УАПП к передаче нового кадра, а флага RI - что в буфере SBUF находится принятый байт, который необходимо считать до окончания приема следующего байта. Ниже приведены соответствующие подпрограммы управления вводом-выводом для режима 1 и 2(3) соответственно.

Режим 1. SPOUT1

SPOUT1: JNB TI, \$; проверка флага TI
CLR TI ; сброс флага
MOV SBUF, A ; выдача символа
RET

SPIN1

SPIN1: JNB RI, \$; проверка флага RI
CLR RI ; сброс флага
MOV A, SBUF ; чтение символа
RET

Режим 2(3) SPOUT2

SPOUT2: MOV C, P ; занесение флага паритета P
MOV TB8, C ; в бит TB8 регистра SCON
JNB TI, \$; проверка флага TI и его
CLR TI ; сброс
MOV SBUF, A ; выдача байта
RET

SPIN2

SPIN2: JNB RI, \$; проверка флага RI и его
CLR RI ; сброс
MOV A, SBUF ; прием байта
MOV C, RB8 ; девятый бит данных во флаг C
JNB P, M1 ; исключаящее ИЛИ между
CPL C ; флагами P и C
RET

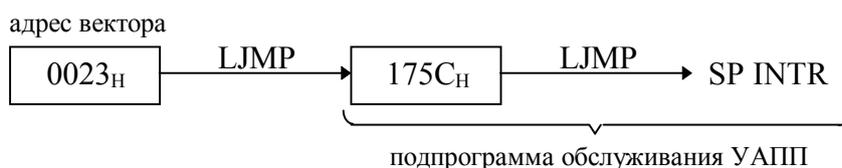
После выхода из подпрограммы SPIN2 проверяется флаг C. Если C=0, то прием/передача проведена правильно, при C=1 - допущен сбой.

Характерной особенностью применения данного способа организации работы последовательного порта является строгая регламентация протокола обмена информацией между микроконтроллерами, либо между МК-51 и периферийным устройством.

3.5.3. Прием/передача на основе обработки прерываний УАПП.

Такая система организации приема/передачи через последовательный порт считается основной, так как освобождает основную программу от постоянного контроля за готовностью передатчика к передаче очередного символа (флаг TI=1) и состоянием регистра SBUF (пустой или полный при RI=1) приемника. Кроме того, подпрограмма обслуживания прерываний в состоянии осуществлять не только одиночные передачи/прием, но и передачу/прием пачек символов, если это требуется из условий работы основной программы.

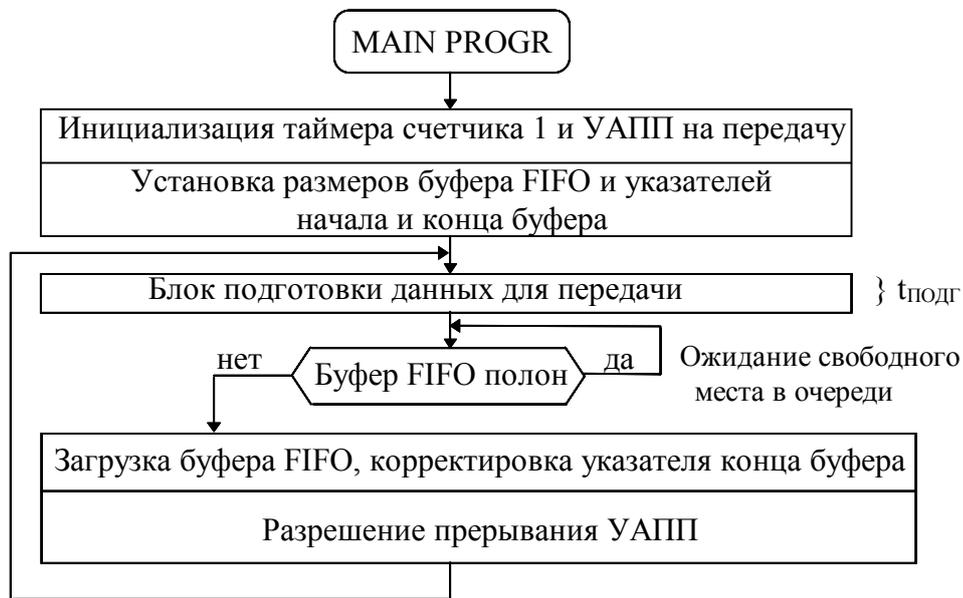
Порядок обработки прерывания от УАПП отвечает общей схеме обработки других запросов и предусматривает формирование по окончании текущей команды (это не относится к команде RETI и командам, связанным с обращением к регистрам IE и IP, после которых выполняется минимум одна команда текущей программы) аппаратного вызова подпрограммы обслуживания (типа команды LCALL), которая помещает в стек содержимое программного счетчика PC и загружает в него адрес вектора 0023_H подпрограммы обслуживания УАПП. По адресу вектора расположена команда безусловного перехода LJMP к начальному адресу подпрограммы обслуживания 175C_H. Поскольку адрес 175C_H граничит со служебной областью памяти, рекомендуется в качестве первой команды подпрограммы обслуживания опять же использовать LJMP для перехода в область ОЗУ пользователя с меньшими адресами, что соответствует следующей схеме



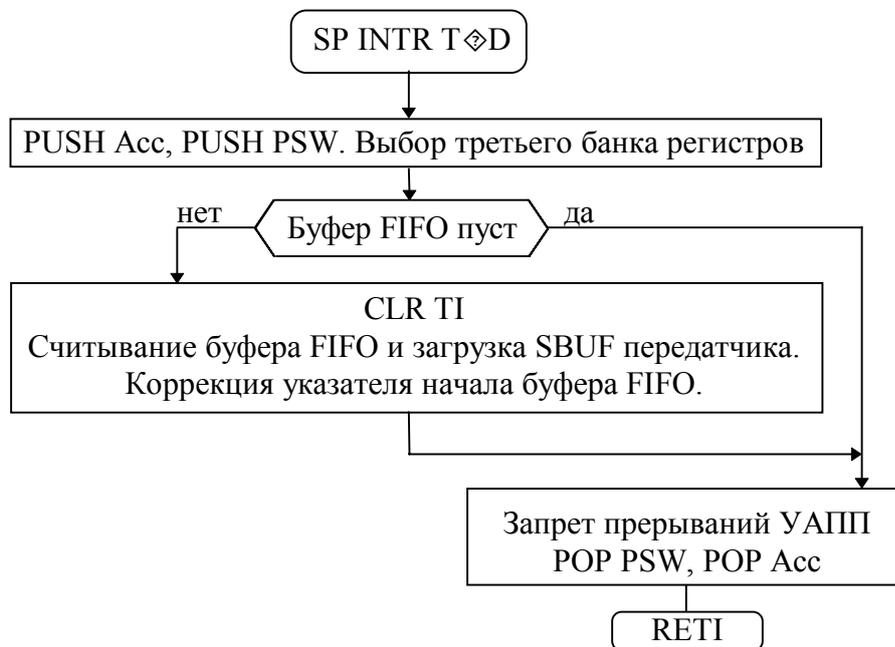
Прежде чем перейти к рассмотрению конкретных программ, отметим еще одну важную особенность работы последовательного порта МК-51, заключающуюся в его независимой параллельной с микропроцессором МК-51 работе (выводе через T \diamond D данных из SBUF передатчика и заполнение SBUF приемника данными со входа R \diamond D). При этом работа подпрограммы занимает всегда короткое время: она или инициирует передачу (загружает SBUF) или считывает из SBUF принятый байт данных, или то и другое последовательно во времени. В то же время работа самого последовательного порта над передачей или приемом байта данных (время T_{КАДР}) занимает, как правило, значительно большее время. Учет этих особенностей и определяет специфику построения программ передачи/приема информации через последовательный порт на основе подпрограмм обработки прерываний.

На рис.16 приведена структура программы передачи с использованием промежуточного буфера передаваемых данных типа FIFO (структура данных переменного размера типа ОЧЕРЕДЬ). Роль синхронизатора основной программы MAIN PROGR и программы обслуживания прерывания SP INTR T \diamond D по выводу данных буфера FIFO здесь выполняет бит разрешения прерывания от УАПП (IE.4). Подпрограмма SP INTR T \diamond D вызывается, когда передатчик готов для передачи следующего байта данных. Прежде всего она определяет, имеются ли данные для передачи и если они имеются, то записываются в буфер передатчика SBUF и производится корректировка указателя. По окончании передачи байта информации подпрограмма запрещает прерывания УАПП и передает управление основной программе. Программа MAIN PROGR снова разрешает прерывание, если появляются новые данные. Следует отметить, что при выполнении условия "t_{ПОДГ}<T_{КАДР}" будет наблюдаться заполнение буфера FIFO (под t_{ПОДГ} понимается время, необходимое основной программе для подготовки очередного байта данных к передаче).

Вообще реализация принципа передачи управления от основной программы к подпрограмме обслуживания прерывания и обратно (протокол взаимодействия или интерфейс) заслуживает самого пристального внимания. Управление работой УАПП на основе прерываний для МК-51 усложняется тем обстоятельством, что вектор прерывания является общим для принципиально различных запросов, его вызвавших (прием или передача). Более того, может оказаться так, что после перехода к обслуживанию прерывания оба флага запросов RI и TI будут находиться в единичном состоянии (это объясняется независимой работой регистров передачи в дуплексном режиме), что требует дополнительного анализа. Можно также говорить о зависимости протокола взаимодействия от временного соотношения между параметрами t_{ПОДГ}<T_{КАДР}.



а - основная программа



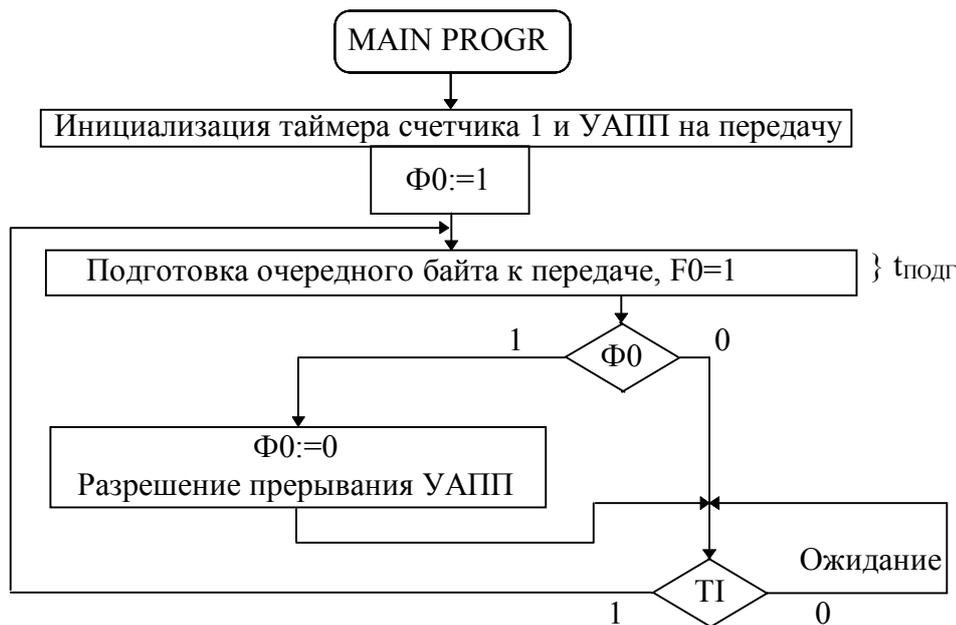
б - подпрограмма прерывания

Рис.16. Блок-схема программы буферизации данных, выводимых через УАПП

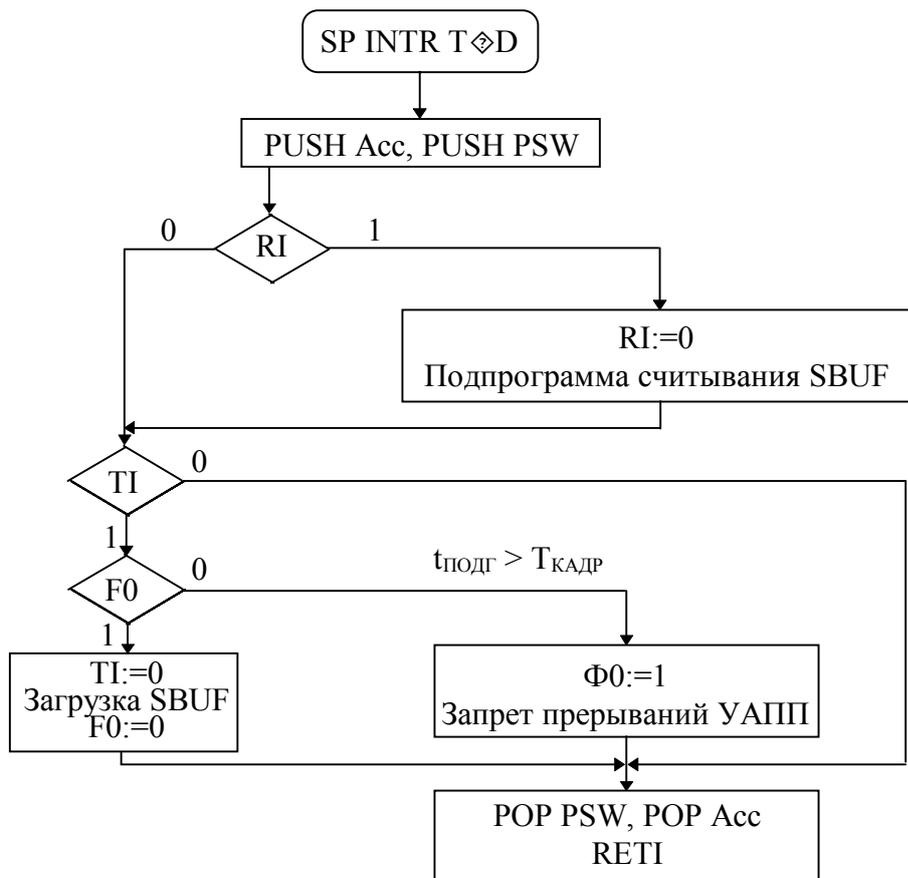
Остановимся на рассмотрении более сложного и наиболее часто встречающегося на практике случая, когда $t_{\text{подг}} \gg T_{\text{кадр}}$. На рис.17 и 18 представлены два варианта одной и той же программы, выполняющей как прием, так и передачу данных: без буферизации данных и с буферизацией соответственно.

Программа, по возможности, должна "мгновенно" и в произвольный момент реагировать на запрос по приему данных. Это требование будет удовлетворяться, если в основной программе MAIN PROGR будут постоянно разрешены прерывания от УАПП, а подпрограмма обработки прерывания SP INTR T◇R◇D будет начинаться с анализа флага RI. Характерной особенностью программы рис.17 является наличие простоя или ожидания в основной программе, так как темп подготовки данных опережает возможность их передачи через последовательный порт ($t_{\text{подг}} \gg T_{\text{кадр}}$). Поэтому в данном случае межпрограммный интерфейс использует рассмотренный ранее принцип опроса готовности флага передатчика TI. При этом передача

ведется при условии $TI \& F0 = 1$. В программах на рис.17 и 18 используются пользовательские флажки с произвольным обозначением: $F0=0$ означает, что очередной байт к передаче не подготовлен, а $\Phi 0=0$ - режим работы программы с разрешенным прерыванием. Запрет прерывания УАПП вводится лишь тогда, когда $t_{\text{ПОДГ}} > T_{\text{КАДР}}$, что соответствует значению $TI=1$ и $F0=0$.



а - основная программа

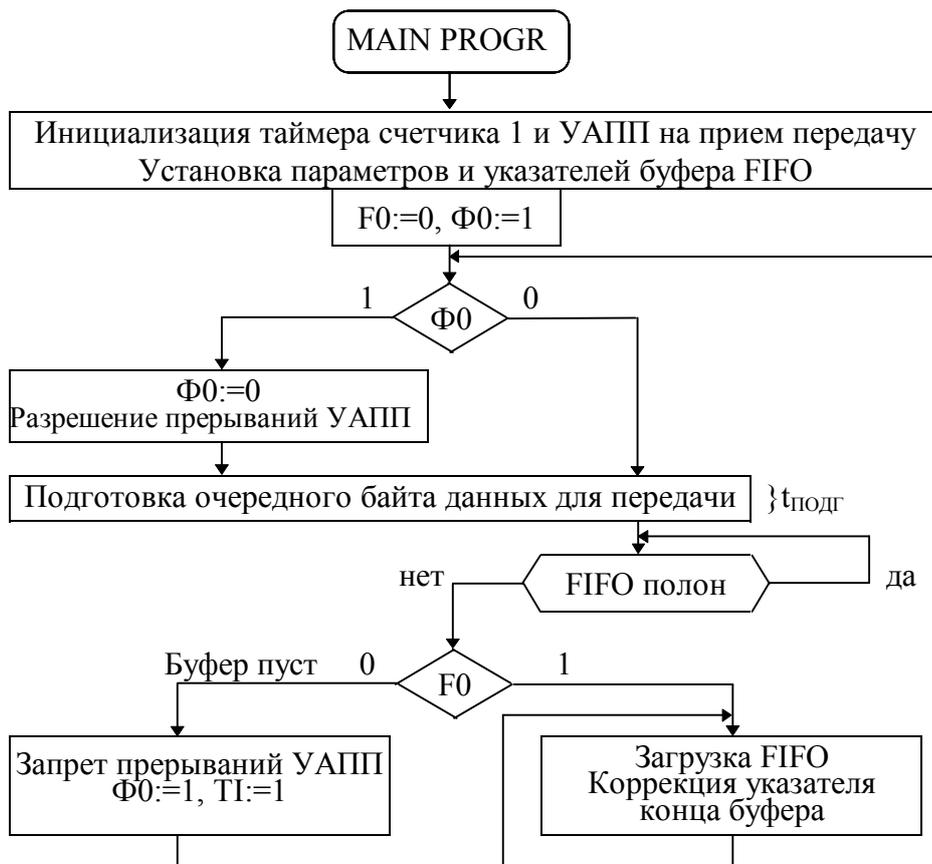


б - подпрограмма обслуживания

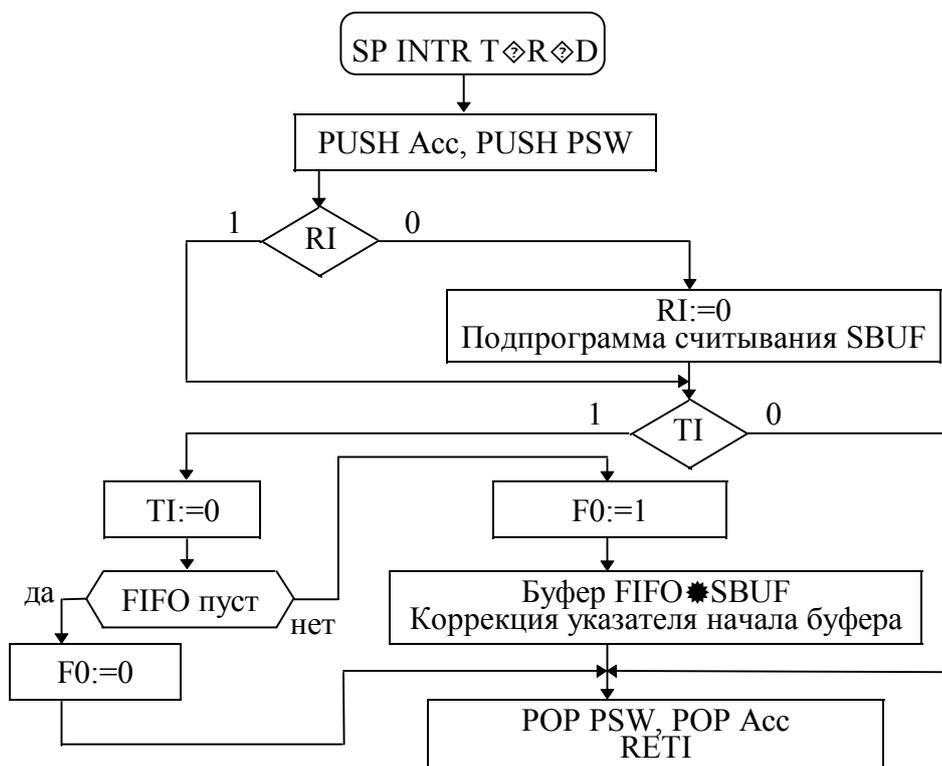
Программа приема/передачи, ориентированная на условие $t_{\text{ПОДГ}} \bullet T_{\text{КАДР}}$

Рис.17.

Другим, альтернативным, вариантом реализации программ, когда $t_{\text{подг}} > T_{\text{КАДР}}$, является использование промежуточного буфера данных типа FIFO (рис.18). В этом случае основная программа будет находится в ожидании лишь при переполнении буфера FIFO. Как и в предыдущей реализации программы, запрет на прерывание УАПП вводится, когда подготовка данных для передачи будет занимать значительное время (то есть $t_{\text{подг}} > T_{\text{КАДР}}$).



а - основная программа



б - подпрограмма обслуживания

Рис.18. Программа приема/передачи с буферизацией данных для передачи

4. ТЕМЫ ЛАБОРАТОРНЫХ РАБОТ И ВАРИАНТЫ ЗАДАНИЙ

Лабораторная работа №1

Изучение структуры и функциональных возможностей УМПК-51.
Изучение архитектуры и системы команд микроконтроллера семейства MSC-51.

Варианты заданий

1. Ввод информации с клавиатуры и отображение ее на дисплее с возможностью "стирания" введенных символов.
2. Вывод на дисплей "бегущего" сообщения.
3. Собственные варианты демонстрационных программ.

Лабораторная работа №2

Универсальный асинхронный приемопередатчик и
программирование процедур обмена информацией

Варианты заданий

Вариант 1. Программа передачи данных с использованием промежуточного буфера данных типа FIFO.

За основу может быть взята структура, приведенная на рис.16. Режим работы УАПШ - первый. Скорость передачи - переменная, от 10 Кбод до 100 бод. Размер буфера FIFO (рис.19) составляет 8 ячеек ОЗУ. Визуальную регистрацию передачи символов информации произвести с помощью осциллографа.

Очередь в буфере FIFO может расти и сокращаться. После включения каждого нового элемента в буфер указатель конца (УК) увеличивается на 1. Когда УК очереди достигнет зарезервированного в памяти конца буфера, он снова приобретает начальное значение (структура замкнута в кольцо). Буфер полон, если $УК+1=УН$ и пуст при $УК=УН$. В исходном состоянии $УК=УН=АДРЕС НАЧАЛА$ буфера.

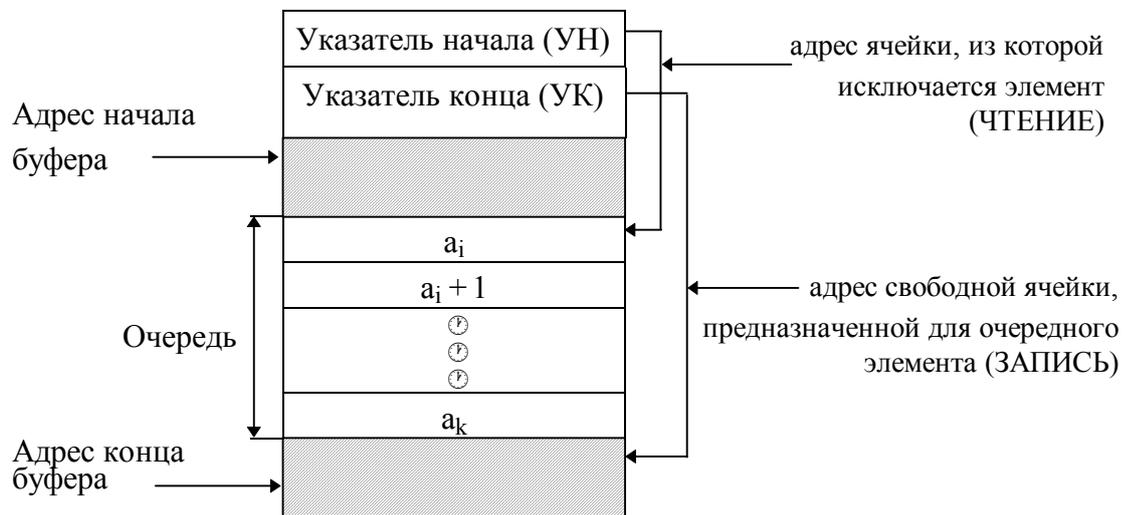


Рис.19 Структура буфера FIFO

Вариант 2. Обмен данными через последовательный порт между МК-51 в режиме "ВЕДУЩИЙ↻ВЕДОМЫЙ".

Микропроцессорный комплекс УМПК-51 содержит в своем составе необходимые технические средства для обеспечения асинхронной передачи/приема информации в стандарте RS-232C. На рис.20 представлена схема коммутации двух УМПК-51 при обмене информацией друг с другом. Требуется разработать программное обеспечение обмена, включающее две

программы, которые назовем по аналогии с выполняемыми функциями контроллеров: Ведущий и Ведомой.

Вид обмена - асинхронный по первому УАПП со скоростью передачи 2400 бод.

Запуск обеих управляющих программ (каждая на своем УМПК-51) производится в произвольный момент времени и сопровождается выводом сообщения "RS-232" на экран дисплея каждого УМПК. Вывод данного сообщения подтверждает факт готовности к реализации соответствующих функций приема или передачи.

Начало передачи инициируется нажатием произвольной клавиши клавиатуры ВЕДУЩЕГО. При этом он посылает управляющее сообщение (УС) из четырех байт В1⊕В4, определяющих начальный и конечный адреса буфера памяти для последующей передачи ВЕДОМОМУ массива данных:

В1 и В2 - младший и старший байты начального адреса буфера;

В3 и В4 - младший и старший байты конечного адреса.

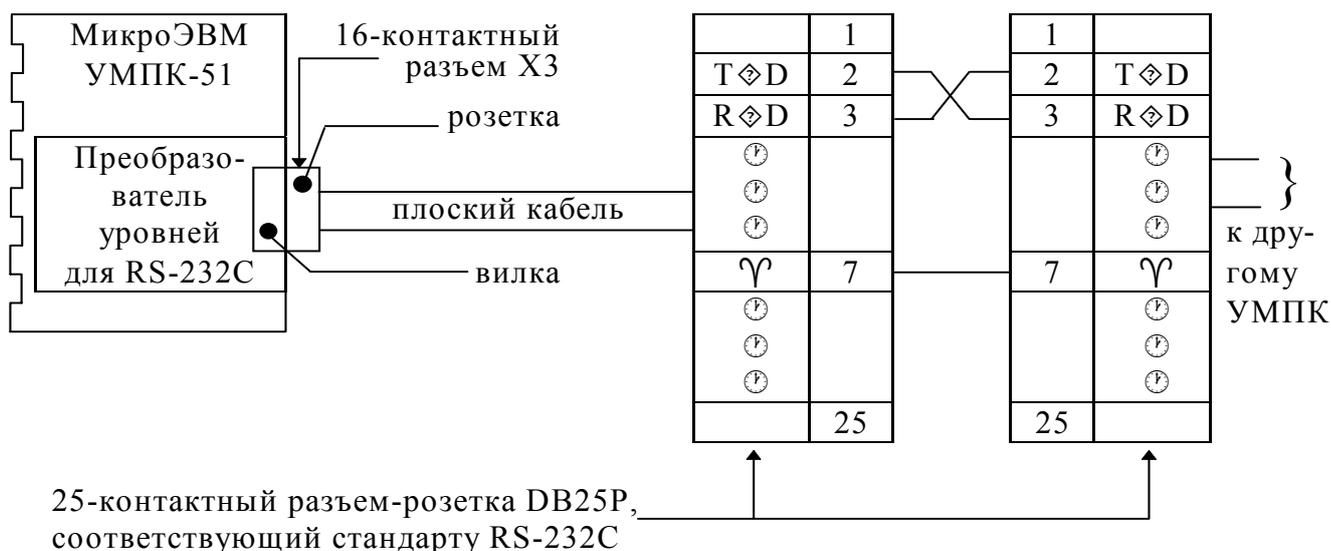


Рис.20. Схема информационной коммутации двух УМПК-51

В ходе приема ВЕДОМЫМ байтов УС осуществляется занесение их в одноименные по номеру регистры внутренней памяти данных с подсчетом контрольной суммы (КС) путем циклического сложения без учета переносов и переполнений.

$$КС = В1 + В2 + В3 + В4.$$

По окончании приема УС ВЕДОМЫЙ принимает пятый В5 контрольный байт (вычисленный уже ВЕДУЩИМ аналогичным образом), который сравнивается с КС. В случае безошибочного приема/передачи ВЕДОМЫЙ переходит в режим ожидания приема основной информации и занесения ее в ОЗУ по полученным в ходе приема УС адресам, в противном случае, он передает ошибочное значение КС ВЕДУЩЕМУ и переходит к точке программы, с которой можно совершить повторный прием УС.

ВЕДУЩИЙ, анализируя факт наличия/отсутствия передачи КС от ВЕДОМОГО (вводится короткая задержка), также либо повторяет передачу УС, либо переходит к передаче массива информации.

В ходе передачи информации как ВЕДУЩИЙ, так и ВЕДОМЫЙ УМПК-51 индицируют адрес ячейки памяти на дисплее. С этой целью байтовые послылки разделяются искусственно введенной задержкой длительностью 0,5⊕1 с. Подсчет КС и выявление ошибок приема/передачи на этом этапе не производятся (с целью сокращения объема учебных программ).

По окончании обмена оба УМПК-51 выводят на дисплей строку символов "End".

На рис.21 и 22 приведены блок-схемы управляющих программ ВЕДУЩАЯ и ВЕДОМАЯ соответственно. Приведенные блок-схемы программ ориентированы на организацию приема/передачи на основе опроса флагов TI и RI, но будут полезны и при разработке программ обслуживания прерываний.

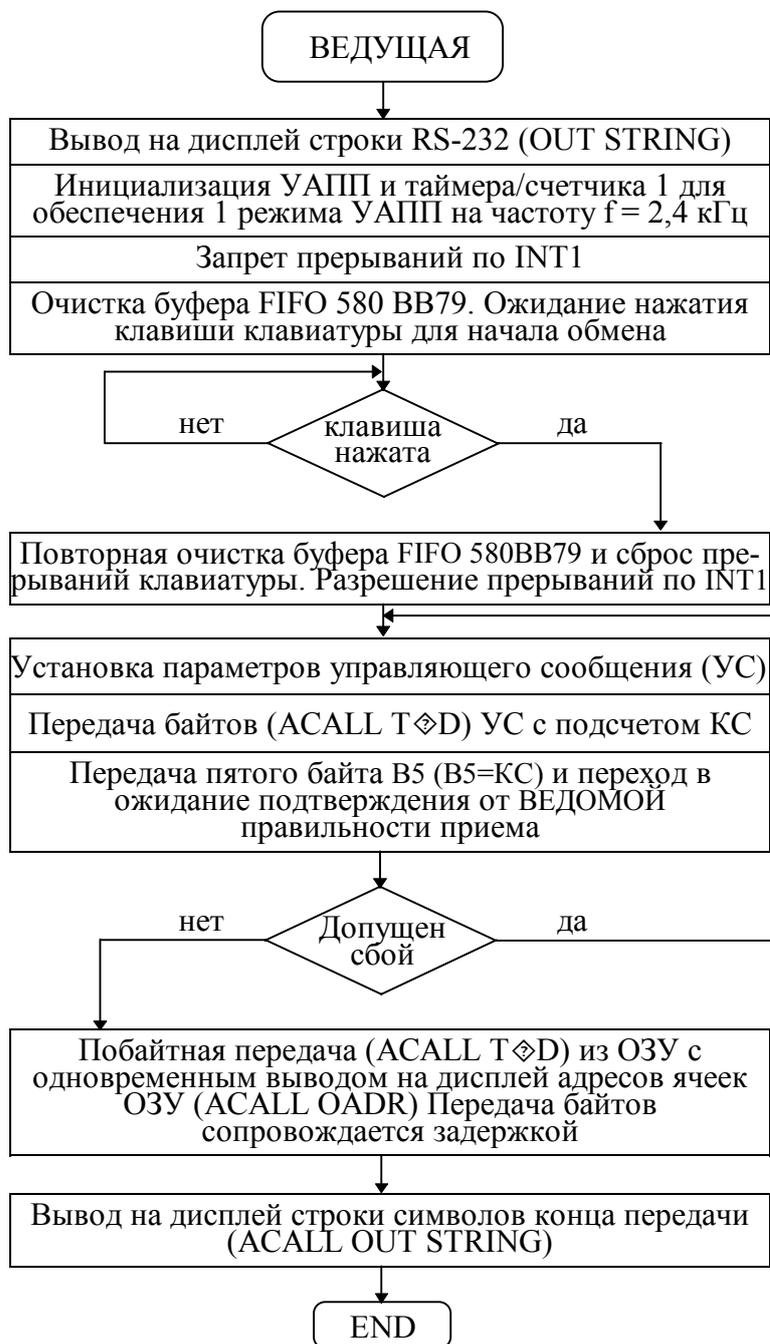


Рис.21. Структура ВЕДУЩЕЙ программы

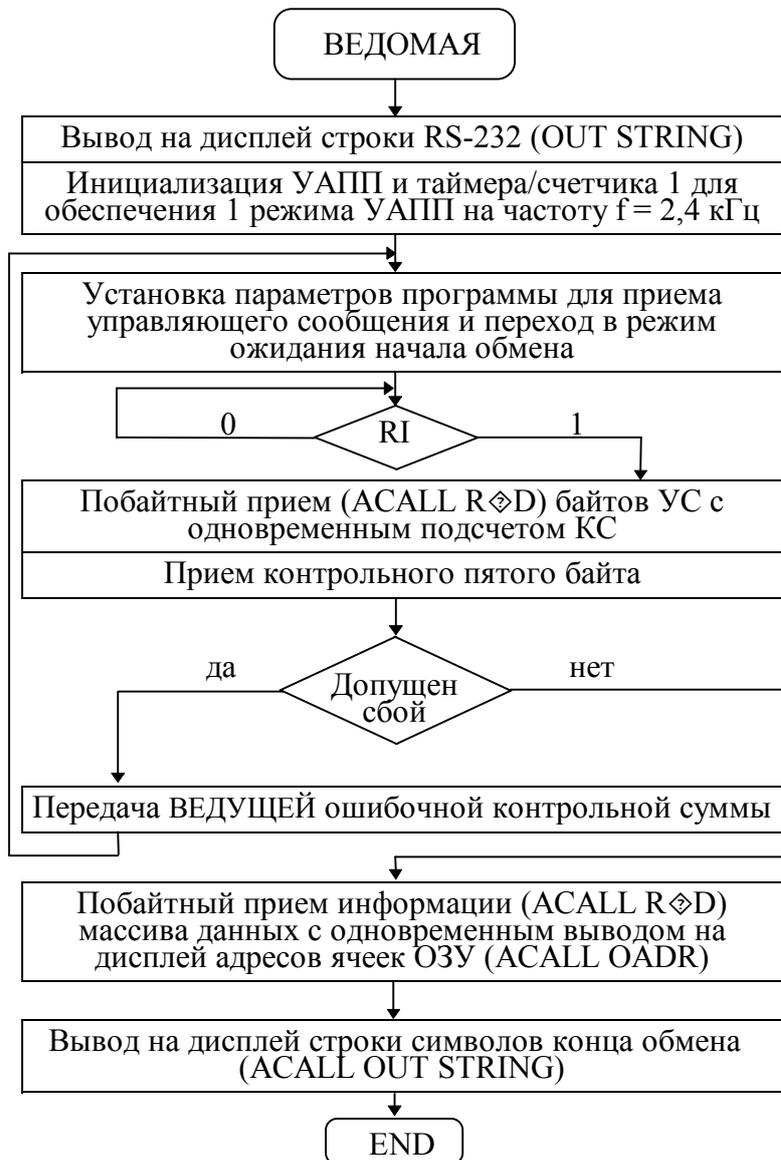


Рис.22. Структура ВЕДОМОЙ программы

Лабораторная работа №3

Подсистемы аналогового ввода/вывода

На рис.23 приведена схема коммутации двух УМПК-51 для проведения данной лабораторной работы.

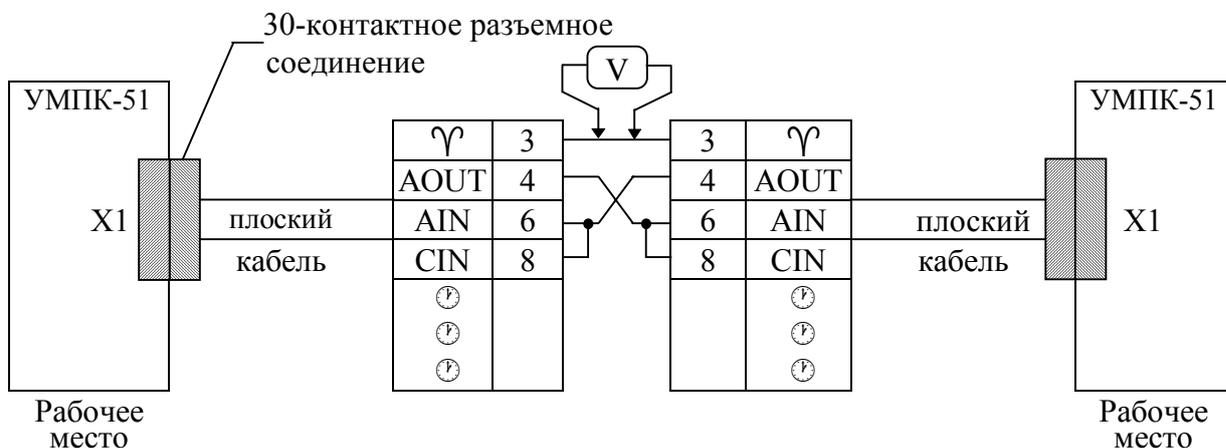


Рис.23. Схема коммутации УМПК-51

Варианты заданий

Вариант 1. Разработка и отладка управляющей программы для подсистемы аналогового вывода.

В качестве основы для разработки программы рекомендуется применять блок-схему программы PROGR AOUT, представленную на рис.24, логика которой практически копирует работу соответствующей управляющей программы монитора (клавиша "DA"). Отметим отличия в их функционировании.

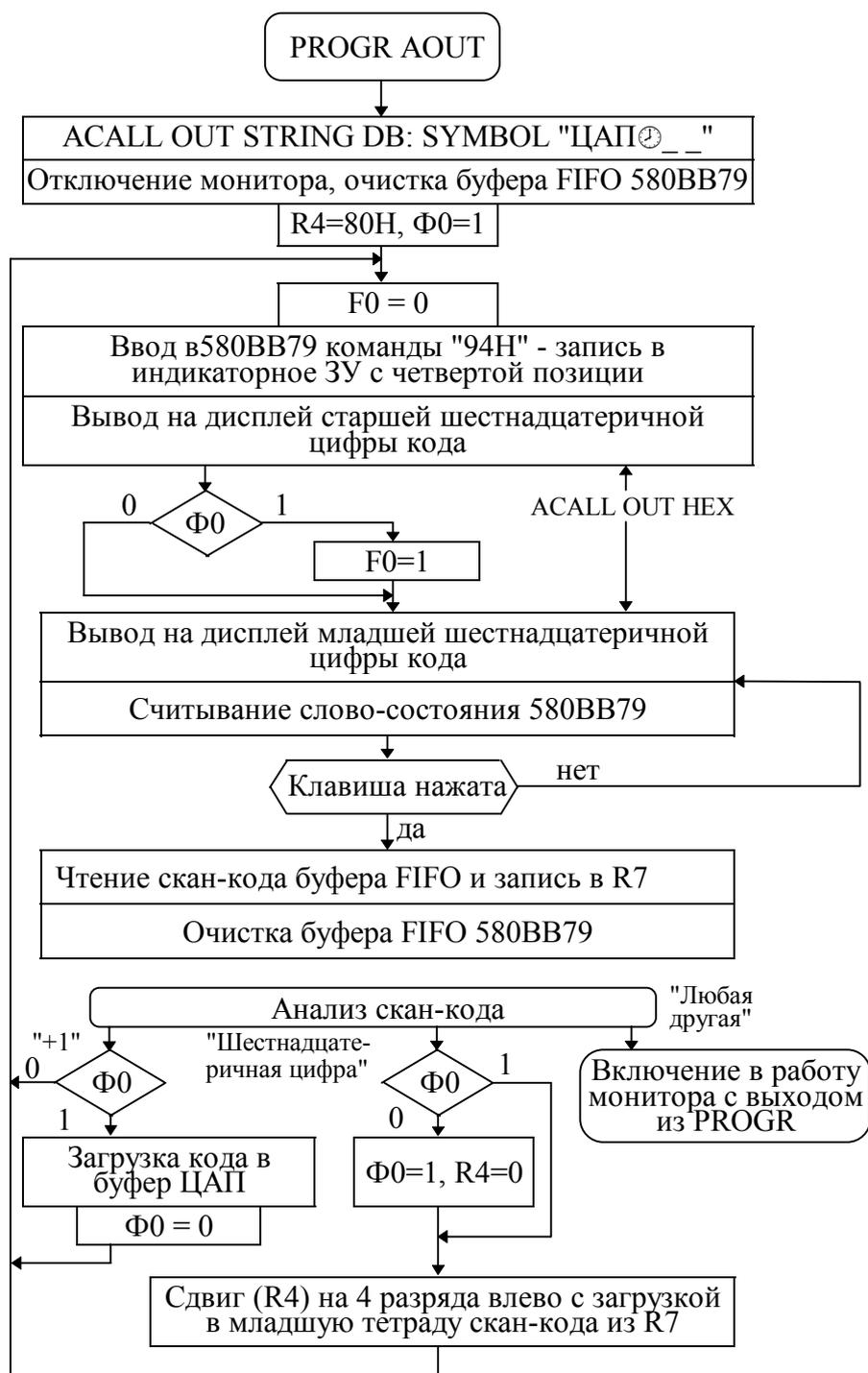


Рис.24. Блок-схема программы обслуживания подсистемы аналогового вывода

1. **PROGR AOUT** использует вывод на дисплей другого сообщения, идентифицирующего ее как учебную программу.

2. Работа учебной программы требует отключения монитора (запрет прерываний по входу INT.1), так как реализация ее функций управления осуществляется посредством клавиш клавиатуры, задействованных монитором.

При выполнении работы значения аналогового вывода U_{AOUT} можно фиксировать (рис.23):

1) с помощью измерительного прибора;

2) косвенно, в виде значений цифрового кода $N_{доп.см}$ на дисплее другого УМПК-51, который работает в режиме аналогового ввода. В последнем случае критерием правильности работы обоих типов аналоговых подсистем будет совпадение кодов $N_{доп.см}$ на дисплеях используемых УМПК-51.

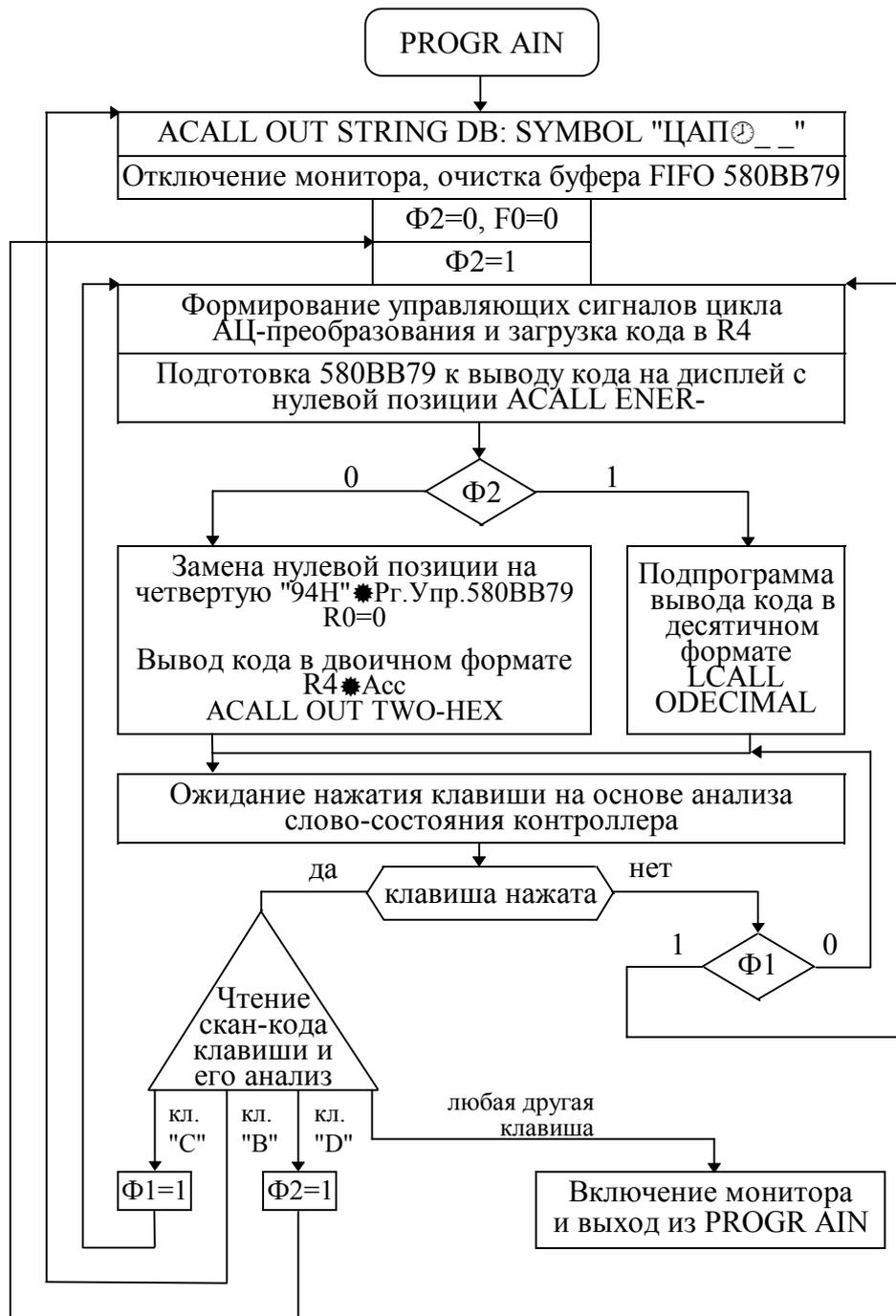


Рис.25. Блок-схема программы обслуживания подсистемы аналогового ввода

Вариант 2. Разработка и отладка управляющей программы для подсистемы аналогового ввода.

Блок-схема учебной программы PROGR AIN, предназначенной для обслуживания аналоговой подсистемы ввода, приведена на рис.25. Как и в предыдущем случае, с подсистемой аналогового вывода, эта программа также очень похожа на управляющую программу монитора (клавиша "AD"), хотя имеются и различия. Отметим их:

1. PROGR AIN имеет свое, идентифицирующее ее выводимое на дисплей сообщение "АЦП⊕XX " и работает при отключенном мониторе.

2. В учебной программе изменено назначение и уменьшено число управляющих клавиш, что упростило внутреннюю логику программ без потери ее функциональных возможностей в управлении.

Программа реализует следующие режимы работы:

Режим 1. Однократное преобразование (одно преобразование - одно нажатие управляющей клавиши) с выводом кода в двоичном формате $N_{\text{доп.см}}$ (клавиша "B").

Режим 2. Однократное преобразование с выводом кода в десятичном формате и диапазоном представления чисел от -10,24 до 10,16 с дискретностью 0,08 (режим цифрового вольтметра).

Режимы 1 и 2 переводятся в циклический характер преобразований входной аналоговой величины нажатием клавиши "C". Возвращение к однократному преобразованию осуществляется простым нажатием клавиши "B" или "D".

Реализация учебной программы PROGR AIN в полном объеме требует выполнения большой работы, особенно в той части, которая связана с преобразованием двоичного кода $N_{\text{доп.см}}$ в десятичную дробь, и выходит за рамки данной лабораторной работы.

В силу этого задание по варианту 2 распадается на два по созданию управляющих программ в усеченном виде:

- программы с представлением кода в двоичном формате (управляющие клавиши "B" и "C");
- программы с представлением кода в десятичном формате (управляющие клавиши "D" и "C") с использованием необходимой для этого подпрограммы ODECIMAL из состава монитора УМПК-51.

Подпрограмма ODECIMAL расположена в области резидентного ПЗУ с начальным адресом 0577_H и занимает 83 ячейки памяти.

При своей работе PROGR AIN использует регистры R0, R2, R3, R4, R5 нулевого банка RB0 и флаги: F0 - флаг вывода на дисплей цифры с запятой, Ф1 - работа АЦП в циклическом режиме (режим измерения), Ф2 - режим вывода выходного кода АЦП в десятичном формате.

Порядок выполнения работ

1. Подготовить контроллеры УМПК-51 к соответствующей лабораторной работе.
2. Ввести и отладить подготовленные тексты программ.
3. Оформить индивидуальный отчет, включающий необходимые иллюстрационные материалы (функциональные схемы исследуемых подсистем, блок-схемы алгоритмов разработанных программ, временные диаграммы и т.п.) и листинги отлаженных программ.

5. ПОДПРОГРАММЫ, РЕКОМЕНДУЕМЫЕ К ИСПОЛЬЗОВАНИЮ В ЛАБОРАТОРНЫХ РАБОТАХ

1. Подготовка индикаторного ОЗУ КР580ВВ79 к вводу информации с нулевого адреса:

```
ENER:  MOV    P2, #84H    ;установка адреса регистра управления
        MOV    R0, #01H  ;контроллера 580ВВ79
        MOV    A, #90H   ;команда ЗАПИСЬ в индикаторное ОЗУ
        MOVX   @R0, A    ;с нулевого адреса
        DEC    R0        ;установка адреса регистра данных
        RET
```

2. Вывод на дисплей строки символов:

DB: КОНСТАНТЫ "STRING"

```

OUT STRING: ACALL  ENER
             MOV    R6, #06      ;число знакомест
             POP    DPH          ;извлечение адреса
             POP    DPL          ;первой константы
M1:  CLR     A
             MOVC  A,@A+DPTR    ;вывод на дисплей
             MOVX  @R0, A       ;очередной константы
             INC   DPTR
             DJNZ  R6, M1
             CLR   A
             JMP   @A+DPTR      ;адрес, след. за посл. константой из семейства "DB: STRING"
    
```

3. Передача байта УАПП с последующей паузой для индикации адреса:

```

T◇D:  JNB    TI, $
       CLR   TI
       MOV   SBUF, A
       ACALL DELAY
       RET
DELAY: PUSH  R1
       MOV   R0, #00H
M2:   MOV   R1, #00H
       DJNZ  R1, $
       DJNZ  R0, M2
       RET
    
```

4. Вывод на дисплей текущего адреса ячеек массива данных:

```

OADR: ACALL  ENER
       MOV   A, DPH      ; загрузка старшей пары цифр адреса
       ACALL OUT TWO-HEX ; вывод старшей пары на дисплей
       MOV   A, DPL      ; загрузка младшей пары цифр адреса
       ACALL OUT TWO-HEX ; вывод младшей пары на дисплей
       CLR   A
       MOVX  @R0, A      ; вывод пробела на пятое и шестое
       MOVX  @R0, A      ; знакоместа дисплея
       RET
    
```

5. Вывод двух шестнадцатеричных цифр на дисплей:

```

OUT TWO-HEX: MOV    R7, A
              SWAP  A
              ACALL OUT HEX      ; вывод шестнадцатеричной цифры
              MOV   A, R7        ; на дисплей
OUT HEX:     ANL   A, #07H
              ADD   A, #07H
              MOVC  A, @A+PC
              JNB   F0(PSW.5), M3 ; при F0=1 вывод цифры
              ORL   A, #80H      ; сопровождается изображением
M3:  MOVX  @R0, A      ; запятой (код 80H)
       RET
    
```

DB - коды символов

```

DB:  3FH(0);  06H(1);  5BH(2);  4FH(3);  66H(4);  6DH(5);
      7DH(6);  07H(7);  7FH(8);  6FH(9);  77H(A);  7CH(B);
      39H(C);  5EH(D);  79H(E);  71H(F);
    
```

Машинные команды ОЭВМ 1816BE51 (по группам операций)

Арифметические операции.

Мнемоническое обозначение маш. команды	Код ком.	К-во байт
ADD A, R0	28	1
ADD A, R1	29	1
ADD A, R2	2A	1
ADD A, R3	2B	1
ADD A, R4	2C	1
ADD A, R5	2D	1
ADD A, R6	2E	1
ADD A, R7	2F	1
ADD A, #data	24	2
ADD A, direct	25	2
ADD A, @R0	26	1
ADD A, @R1	27	1
ADDC A, R0	38	1
ADDC A, R1	39	1
ADDC A, R2	3A	1
ADDC A, R3	3B	1
ADDC A, R4	3C	1
ADDC A, R5	3D	1
ADDC A, R6	3E	1
ADDC A, R7	3F	1

Мнемоническое обозначение маш. команды	Код ком.	К-во байт
ADDC A, #data	34	2
ADDC A, direct	35	2
ADDC A, @R0	36	1
ADDC A, @R1	37	1
SUBB A, R0	98	1
SUBB A, R1	99	1
SUBB A, R2	9A	1
SUBB A, R3	9B	1
SUBB A, R4	9C	1
SUBB A, R5	9D	1
SUBB A, R6	9E	1
SUBB A, R7	9F	1
SUBB A, #data	94	2
SUBB A, direct	95	2
SUBB A, @R0	96	1
SUBB A, @R1	97	1
INC A	04	1
INC R0	08	1
INC R1	09	1
INC R2	0A	1
INC R3	0B	1
INC R4	0C	1

Мнемоническое обозначение маш. команды	Код ком.	К-во байт
INC R5	0D	1
INC R6	0E	1
INC R7	0F	1
INC direct	05	2
INC @R0	06	1
INC @R1	07	1
DEC A	14	1
DEC R0	18	1
DEC R1	19	1
DEC R2	1A	1
DEC R3	1B	1
DEC R4	1C	1
DEC R5	1D	1
DEC R6	1E	1
DEC R7	1F	1
DEC direct	15	2
DEC @R0	16	1
DEC @R1	17	1
INC DPTR	A3	1
MUL AB	A4	1
DIV AB	84	1
DA A	D4	1

Логические операции

Мнемоническое обозначение маш. команды	Код ком.	К-во байт
ANL A, R0	58	1
ANL A, R1	59	1
ANL A, R2	5A	1
ANL A, R3	5B	1
ANL A, R4	5C	1
ANL A, R5	5D	1
ANL A, R6	5E	1
ANL A, R7	5F	1
ANL A, #data	54	2
ANL A, direct	55	2
ANL A, @R0	56	1
ANL A, @R1	57	1
ANL direct, A	52	2
ANL direct, #data	53	3
CLR A	E4	1
CPL A	F4	1
SWAP A	C4	1

Мнемоническое обозначение маш. команды	Код ком.	К-во байт
XRL A, R0	68	1
XRL A, R1	69	1
XRL A, R2	6A	1
XRL A, R3	6B	1
XRL A, R4	6C	1
XRL A, R5	6D	1
XRL A, R6	6E	1
XRL A, R7	6F	1
XRL A, #data	64	2
XRL A, direct	65	2
XRL A, @R0	66	1
XRL A, @R1	67	1
XRL direct, A	62	2
XRL direct, #data	63	3
RL A	23	1
RLC A	33	1

Мнемоническое обозначение маш. команды	Код ком.	К-во байт
ORL A, R0	48	1
ORL A, R1	49	1
ORL A, R2	4A	1
ORL A, R3	4B	1
ORL A, R4	4C	1
ORL A, R5	4D	1
ORL A, R6	4E	1
ORL A, R7	4F	1
ORL A, #data	44	2
ORL A, direct	45	2
ORL A, @R0	46	1
ORL A, @R1	47	1
ORL direct, A	42	2
ORL direct, #data	43	3
RR A	03	1
RRC A	13	1

Команды вызова подпрограмм

и возврата

Мнемоническое обозначение машинной команды	Код ком.	К-во байт
LCALL addr16	12	3
ACALL 000H-0FFH	11	2
ACALL 100H-1FFH	31	2
ACALL 200H-2FFH	51	2
ACALL 300H-3FFH	71	2
ACALL 400H-4FFH	91	2
ACALL 500H-5FFH	B1	2
ACALL 600H-6FFH	D1	2
ACALL 700H-7FFH	F1	2
RET	22	1
RETI	32	1

Команды безусловных

переходов

Мнемоническое обозначение машинной команды	Код ком.	К-во байт
LJMP addr16	02	3
AJMP 000H-0FFH	01	2
AJMP 100H-1FFH	21	2
AJMP 200H-2FFH	41	2
AJMP 300H-3FFH	61	2
AJMP 400H-4FFH	81	2
AJMP 500H-5FFH	A1	2
AJMP 600H-6FFH	C1	2
AJMP 700H-7FFH	E1	2
SJMP rel	80	2
JMP @A+DPTR	73	1

Команды условных переходов.

Мнемоническое обозначение машинной команды	Код ком.	К-во байт
CJNE A, direct, rel	B5	3
CJNE A, #data, rel	B4	3
CJNE R0, #data, rel	B8	3
CJNE R1, #data, rel	B9	3
CJNE R2, #data, rel	BA	3
CJNE R3, #data, rel	BB	3
CJNE R4, #data, rel	BC	3
CJNE R5, #data, rel	BD	3
CJNE R6, #data, rel	BE	3
CJNE R7, #data, rel	BF	3
CJNE @R0, #data, rel	B6	3
CJNE @R1, #data, rel	B7	3

Мнемоническое обозначение машинной команды	Код ком.	К-во байт
DJNZ R0, rel	D8	2
DJNZ R1, rel	D9	2
DJNZ R2, rel	DA	2
DJNZ R3, rel	DB	2
DJNZ R4, rel	DC	2
DJNZ R5, rel	DD	2
DJNZ R6, rel	DE	2
DJNZ R7, rel	DF	2
DJNZ direct, rel	D5	3
JZ rel	60	2
JNZ rel	70	2
NOP	00	1

Команды обработки отдельных бит данных.

Мнемоническое обозначение машинной команды	Код ком.	К-во байт
CLR C	C3	1
CLR bit	C2	2
SETB C	D3	1
SETB bit	D2	2
CPL C	B3	1
CPL bit	B2	2
ANL C, bit	82	2
ANL C, /bit	B0	2

Мнемоническое обозначение машинной команды	Код ком.	К-во байт
ORL C, bit	72	2
ORL C, /bit	A0	2
MOV C, bit	A2	2
MOV bit, C	92	2
JC rel	40	2
JNC rel	50	2
JB bit, rel	20	3
JNB bit, rel	30	3
JBC bit, rel	10	3

Команды пересылок данных.

Мнемоническое обозначение маш. команды	Код ком.	К-во байт	Мнемоническое обозначение маш. команды	Код ком.	К-во байт	Мнемоническое обозначение маш. команды	Код ком.	К-во байт
MOV A, R0	E8	1	MOV R6, #data	7E	2	MOV direct, direct	85	3
MOV A, R1	E9	1	MOV R7, #data	7F	2	MOV direct, @R0	86	2
MOV A, R2	EA	1	MOV R0, direct	A8	2	MOV direct, @R1	87	2
MOV A, R3	EB	1	MOV R1, direct	A9	2	MOV DPTR, #data16	90	3
MOV A, R4	EC	1	MOV R2, direct	AA	2	MOVC A, @A+DPTR	93	1
MOV A, R5	ED	1	MOV R3, direct	AB	2	MOVC A, @A+PC	83	1
MOV A, R6	EE	1	MOV R4, direct	AC	2	MOVX A, @R0	E2	1
MOV A, R7	EF	1	MOV R5, direct	AD	2	MOVX A, @R1	E3	1
MOV A, #data	74	2	MOV R6, direct	AE	2	MOVX @R0, A	F2	1
MOV A, direct	E5	2	MOV R7, direct	AF	2	MOVX @R1, A	F3	1
MOV A, @R0	E6	1	MOV @R0, A	F6	1	MOVX A, @DPTR	E0	1
MOV A, @R1	E7	1	MOV @R1, A	F7	1	MOVX @DPTR, A	F0	1
MOV R0, A	F8	1	MOV @R0, #data	76	2	PUSH direct	C0	2
MOV R1, A	F9	1	MOV @R1, #data	77	2	POP direct	D0	2
MOV R2, A	FA	1	MOV @R0, direct	A6	2	XCH A, R0	C8	1
MOV R3, A	FB	1	MOV @R1, direct	A7	2	XCH A, R1	C9	1
MOV R4, A	FC	1	MOV direct, A	F5	2	XCH A, R2	CA	1
MOV R5, A	FD	1	MOV direct, R0	88	2	XCH A, R3	CB	1
MOV R6, A	FE	1	MOV direct, R1	89	2	XCH A, R4	CC	1
MOV R7, A	FF	1	MOV direct, R2	8A	2	XCH A, R5	CD	1
MOV R0, #data	78	2	MOV direct, R3	8B	2	XCH A, R6	CE	1
MOV R1, #data	79	2	MOV direct, R4	8C	2	XCH A, R7	CF	1
MOV R2, #data	7A	2	MOV direct, R5	8D	2	XCH A, direct	C5	2
MOV R3, #data	7B	2	MOV direct, R6	8E	2	XCH A, @R0	C6	1
MOV R4, #data	7C	2	MOV direct, R7	8F	2	XCH A, @R1	C7	1
MOV R5, #data	7D	2	MOV direct, #data	75	3	XCHD A, @R0	D6	1
						XCHD A, @R1	D7	1

Условные обозначения.

Rn - один из РОН-ов R0 - R7 (в текущем регистровом банке).

direct - 8-битный *прямой* адрес внутренней памяти данных.

@Ri - 8-аёóíúé êñâáííúé ääðñ ÿ÷:áéèè áíóððáííáé èèè áíáøíáé ìàìüðè äáííúð. (i = 0, 1).

#data, #data16 - 8 или 16-разрядный *непосредственный* операнд.

addr16 - 16-разрядный адрес перехода .

000_h-7FF_h - 11-разрядный адрес перехода.

rel - 8-разрядный *относительный* адрес перехода.

bit - адрес *прямоадресуемого бита* из области внутренней памяти данных или из SFR-области.

Основные характеристики микроконтроллеров MCS 51 / MCS 251

Таблица П1

Обозначение	Резидентная память программ	Резидентное ОЗУ	Таймеры/счетчики	Послед. порт	Аналоговые 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 _{CC} от 2.7В до 3.6В
87L52	8K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V _{CC} от 2.7В до 3.6В
80L54	16K ROM	256	3	1	0	32	12,16,20*	N,S	L1	V _{CC} от 2.7В до 3.6В

Обозначение	Резидентная память программ	Резидентное ОЗУ	Таймеры/счетчики	Послед. порт	Аналоговые входы	Число линий В/В	Тактовая частота	Тип корпуса	Секретность	Ключевые особенности
87L54	16K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V _{CC} от 2.7В до 3.6В
80L58	32K ROM	256	3	1	0	32	12,16,20*	N,S	L1	V _{CC} от 2.7В до 3.6В
87L58	32K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V _{CC} 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 _{CC} от 2.7В до 3.6В, PCA
83L51FA	8K ROM	256	3	1	0	32	12,16,20*	N,S	L1	V _{CC} от 2.7В до 3.6В, PCA
87L51FA	8K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V _{CC} от 2.7В до 3.6В, PCA
83L51FB	16K ROM	256	3	1	0	32	12,16,20*	N,S	L1	V _{CC} от 2.7В до 3.6В, PCA
87L51FB	16K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V _{CC} от 2.7В до 3.6В, PCA
83L51FC	32K ROM	256	3	1	0	32	12,16,20*	N,S	L1	V _{CC} от 2.7В до 3.6В, PCA
87L51FC	32K OTP ROM	256	3	1	0	32	12,16,20*	N,S	L3	V _{CC} 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

Обозначение	Резидентная память программ	Резидентное ОЗУ	Таймеры/счетчики	Послед. порт	Аналоговые входы	Число линий В/В	Тактовая частота	Тип корпуса	Секретность	Ключевые особенности
<i>Серия 8XC51GB</i>										
80C51GB	нет	256	3	1+SEP	8	48	12,16	N1	нет	8-разрядный АЦП, 2 PCA, 6 портов В/В
83C51GB	8K ROM	256	3	1+SEP	8	48	12,16	N1	L1	8-разрядный АЦП, 2 PCA, 6 портов В/В
87C51GB	8K OTP ROM	256	3	1+SEP	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 _{CC} от 3В до 3.6В, 8-разрядный АЦП
81C51SL-AL	16K ROM1	256	2	1	4	24	16	SB	нет	V _{CC} от 3В до 3.6В, 8-разрядный АЦП
83C51SL-AL	16K ROM	256	2	1	4	24	16	SB	нет	V _{CC} от 3В до 3.6В, 8-разрядный АЦП
87C51SL-AL	16K OTP	256	2	1	4	24	16	SB	нет	V _{CC} от 3В до 3.6В, 8-разрядный АЦП
<i>MCS251 — Новая архитектура</i>										
80C251SB	нет	1K	3	1	0	32	12,16	N	нет	Расшир. архитектура, PCA, аппаратный WDT
83C251SB	16K ROM	1K	3	1	0	32	12,16	N	L1	Расшир. архитектура, PCA, аппаратный WDT
87C251SB	16K OTP ROM	1K	3	1	0	32	12,16	N	L3	Расшир. архитектура, PCA, аппаратный WDT

Примечания:

Технология изготовления микроконтроллеров серий 8051 и 8052 - NMOS (n-МОП), все остальные выполнены по технологии CHMOS (КМОП).

ROM	— масочное ПЗУ, программируемое на заводе-изготовителе математическим обеспечением заказчика
ROM1	— масочное ПЗУ, фирменным обеспечением заказчика
OTP ROM	— ПЗУ, программируемое на заводе-изготовителе
EPROM	— ПЗУ, программируемое на заводе-изготовителе
OTP/EPROM	— EPROM, OTP ROM

Тип корпуса

D	— 40-выводной керамический DIP
P	— 40-выводной пластмассовый DIP
P1	— 48-выводной пластмассовый DIP
N	— 44-выводной пластмассовый PLCC
N1	— 68-выводной пластмассовый PLCC
S	— 44-выводной пластмассовый QFP
SB	— 100-выводной пластмассовый SQFP
KU	— 100-выводной пластмассовый QFP

Частота (мГц)	24i — 24 мГц - внутренняя частота функционирования
	20* — 20 мГц - только для коммерческого диапазона температур

Секретность	L1 — 1 бит блокировки
	L2 — 2 бита блокировки
	L3 — 3 бита блокировки
	P — защита

PCA	— широтно-импульсный модулятор, модуль захвата/сравнения, высоко-скоростные выходы, сторожевой таймер
------------	---

SEP	— сторожевой таймер
------------	---------------------

Список литературы

1. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. - М.: Радио и связь, 1989.
2. Однокристальные микроЭВМ. Семейство МК48. Семейство МК51. Техническое описание и руководство по применению. /Липовецкий Г.Н., Литвинский Г.В., Оксинь О.Н., Проценко Л.В., Петренко Н.В., Сивобородов П.В. - М.: МП "Бином", 1992.
3. Сташин В.В. и др. Проектирование цифровых устройств на однокристальных микроконтроллерах. /Сташин В.В., Урусов А.В., Мологонцева О.Ф., - М.: Энергоатомиздат, 1990.
4. Однокристальные микроЭВМ / Боборыкин А.В., Липовецкий Г.Н., Литвинский Г.В., Оксинь О.Н., Прохорчик С.В., Проценко Л.В., Петренко Н.В., Сергеев А.А., Сивобородов П.В. - М.: МИКАП, 1994.
5. Модуль УМПК-51 /ВМ. Паспорт 1990.
6. Нерода В.Я., Торбинский В.Э., Шлыков Е.Л. Однокристальные микроЭВМ MCS-51. Архитектура. - М.: Дижитал Компонентс, 1995.
7. MCS 51. Microcontroller Family User's Manual. Order NO.: 272383-002, February 1994.
8. Embedded Microcontrollers. Order NO.: 270646, 1994.
9. Федорков Б.Г., Телец В.А. Микросхемы ЦАП и АЦП: Функционирование, параметры, применение. - М.: Энергоатомиздат, 1990.
10. Встраиваемый микроконтроллер 8XC251SB. Руководство пользователя. - Киев: "Квазар-Микро", 1995.
11. Однокристальный микроконтроллер семейства MCS-51 фирмы INTEL 8X51GB. Микросхемы FLASH памяти 28F512/28F256A. Микросхемы статической памяти UM6264B/UM62256B. - Томск: ТОО "SDD", 1995.
12. Packaging. Order NO.: 240800, 1994.