

№102
7.04.15.


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПУТЕЙ СООБЩЕНИЯ»

Кафедра: « Железнодорожная автоматика, телемеханика и связь »

Факультет: « Транспортные средства »

Курсовой проект

по дисциплине: «Микропроцессорные информационно-управляющие системы »

1. Структурная схема
МТТ системы, анализ
не работ, временные диаграммы
2. Машинный алгоритм
3. Расчет
4. Программирование
К работе после проверки
замечаний
23.04.15


Выполнила: студентка 4-го курса

Шифр: 1110-СДс-0847

Тишакова Е. И.

Проверил: доц. Ермакова О.П.

Москва 2015 г.

Функциональное назначение выводов корпуса МК51

Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии и выпускается в корпусе БИС, имеющем 40 внешних выводов. Цоколевка корпуса МК51 и наименования выводов показаны на рис. 3.1. Для работы МК51 требуется один источник электропитания +5 В. Через четыре программируемых порта ввода/вывода МК51 взаимодействует со средой в стандарте TTL-схем с тремя состояниями выхода. Корпус МК51 имеет два вывода для подключения кварцевого резонатора, четыре вывода для сигналов, управляющих режимом работы МК, и восемь линий порта 3, которые могут быть запрограммированы пользователем на выполнение специализированных (альтернативных) функций обмена информацией со средой.

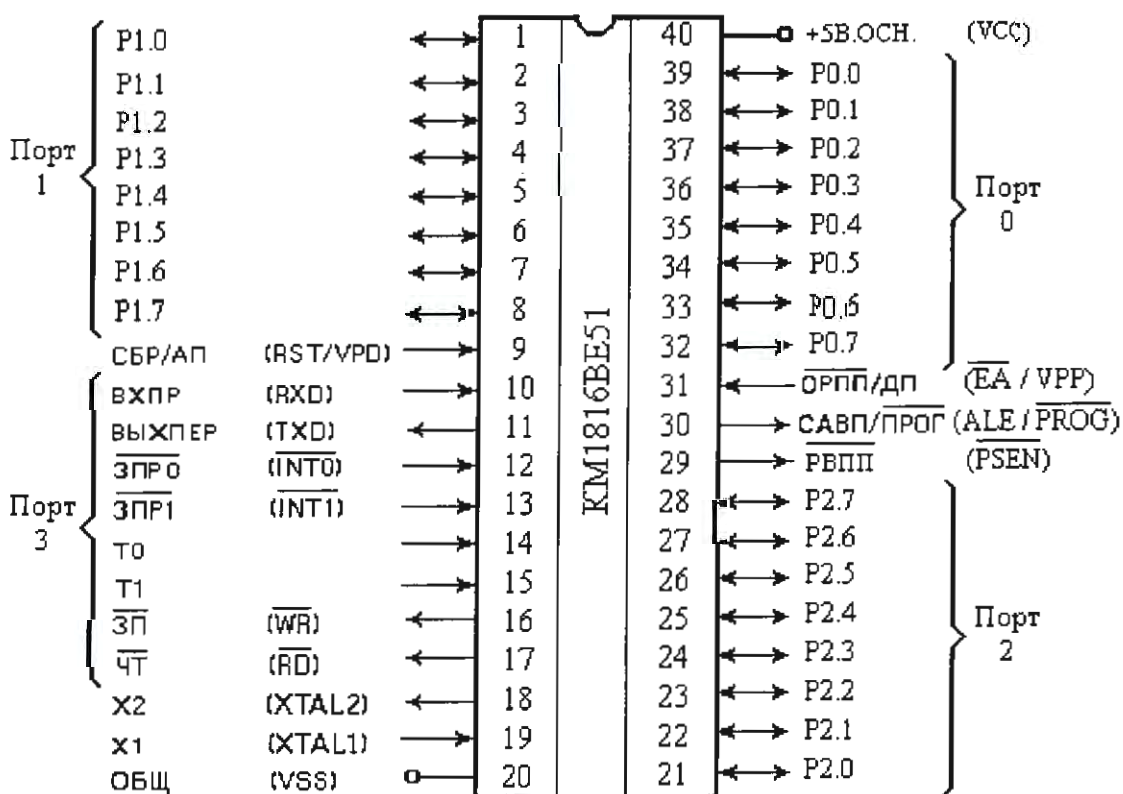


Рис. 3.1. Цоколёвка корпуса МК51 и наименование выводов

Структурная схема МК51

Основу структурной схемы МК51 (рис. 3.2) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой все основные узлы и устройства: резидентную память, АЛУ, блок регистров специальных функций, устройство управления и порты ввода/вывода. Рассмотрим основные элементы структуры и особенности организации вычислительного процесса в МК51.

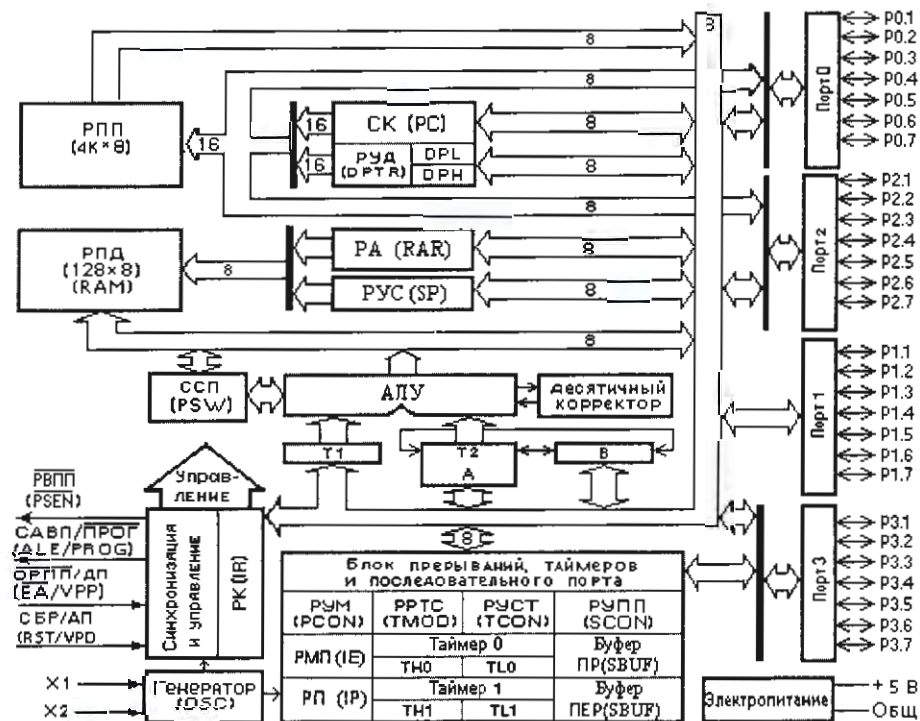


Рис.3.2. Структурная схема МК51

Арифметическо-логическое устройство

8-битное АЛУ может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. В АЛУ имеются программно недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции и схема формирования признаков.

Простейшая операция сложения используется в АЛУ для инкрементирования содержимого регистров, продвижения регистра-указателя данных и автоматического вычисления следующего адреса РПШ. Простейшая операция вычитания используется в АЛУ для декрементирования регистров и сравнения переменных.

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

Важной особенностью АЛУ является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Эта способность АЛУ, оперировать битами, столь важна, что во многих описаниях МК51 говорится о наличии в нем "булевого процессора". Для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевскими переменными (истина/ложь), реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Таким образом, АЛУ может оперировать четырьмя типами информационных объектов: булевскими (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В АЛУ выполняется 51 различная операция пересылки или преобразования этих данных. Так

как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования "операция/ режим адресации" базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

Резидентная память

Память программ и память данных, размещенные на кристалле МК5 физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции.

Память программ (ПЗУ или СППЗУ) имеет емкость 4 Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации, таблиц перекодировки входных и выходных сменных и т.п. РПП имеет 16-битную шину адреса, через которую обеспечивается доступ из счетчика команд или из регистра-указателя данных. Последний выполняет функции базового регистра при косвенных переходах по программе или используется в командах, оперирующих с таблицами.

Память данных (ОЗУ) предназначена для хранения переменных в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к адресному пространству РПД примыкают адреса регистров специальных функций (РСФ), которые перечислены в таблице 2.1.

Память программ, так же как и память данных, может быть расширена до 64 Кбайт путем подключения внешних БИС.

Аккумулятор и ССП. Аккумулятор является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того,

только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п.

Таблица 2.1. Блок регистров специальных функций

Символ	Наименование	Адрес
* ACC	Аккумулятор	0E0H
* B	Регистр-расширитель аккумулятора	0F0H
* PSW	Слово состояния программы	0D0H
SP	Регистр-указатель стека	81H
DPTR	Регистр-указатель данных (DPH)	83H
	(DPL)	82H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регистр приоритетов	0B8H
* IE	Регистр маски прерываний	0A8H
TMOD	Регистр режима таймера/счетчика	89H
* TCON	Регистр управления/статус таймера	88H
TH0	Таймер 0 (старший байт)	8CH
TL0	Таймер 0 (младший байт)	8AH
TH1	Таймер 1 (старший байт)	8DH
TL1	Таймер 1 (младший байт)	8BH
* SCON	Регистр управления приемопередатчиком	98H
SBUF	Буфер приемопередатчика	99H
PCON	Регистр управления мощностью	87H
Примечание. Регистры, имена которых отмечены знаком (*), допускают адресацию отдельных бит.		

При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре ССП. В таблице 2.2 приводится перечень флагов ССП, даются их символические имена и описываются условия их формирования.

Таблица 2.2. Формат слова состояния программы (ССП)

Символ	Позиция	Имя и назначение			
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается аппаратными средствами или программой при выполнении арифметических и логических операций			
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заем в бите 3			
F0	PSW.5	Флаг 0. Может быть установлен, сброшен или проверен программой как флаг, специфицируемый пользователем			
RS1	PSW.4	Выбор банка регистров. Устанавливается и сбрасывается программой для выбора рабочего банка регистров (см. примечание)			
RS0	PSW.3				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций			
—	PSW.1	Не используется			
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратуры в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе, т.е. выполняет контроль по четности			
Примечание.		RS1	RS0	Банк	Границы адресов
		0	0	0	00H-07H
		0	1	1	08H-0FH
		1	0	2	10H-17H
		1	1	3	18H-1FH

Наиболее "активным" флагом ССП является флаг переноса, которые принимает участие и модифицируется в процессе выполнения множеств операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции "булевого аккумулятора" в

командах, манипулирующих с битами. Флаг переполнения (OV) фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. АЛУ не управляет флагами селекции банка регистров (RS0, RS1), и их значение полностью определяется прикладной программой и используется для выбора одного из четырех регистровых банков.

Широкое распространение получило представление о том, что в микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя адресацию "по умолчанию" (неявную). В МК51 дело обстоит иначе. Хотя процессор в МК51 имеет в своей основе аккумулятор, однако, он может выполнять множество команд и без участия аккумулятора. Например, данные могут быть переданы из любой ячейки РПД в любой регистр, любой регистр может быть загружен непосредственным операндом и т.д. Многие логические операции могут быть выполнены без участия аккумулятора. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены (test) без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

Регистры-указатели. 8-битный указатель стека (PUS) может адресовать любую область РПД. Его содержимое инкрементируется прежде, чем данные будут запомнены в стеке в ходе выполнения команд PUSH и CALL. Содержимое PUS декрементируется после выполнения команд POP и RET. Подобный способ адресации элементов стека называют прединкрементным/постдекрементным. В процессе инициализации МК51 после сигнала СБР в PUS автоматически загружается код 07H. Это значит, что если прикладная программа не переопределяет стек, то первый элемент данных в стеке будет располагаться в ячейке РПД с адресом 08H.

Двухбайтный регистр-указатель данных (РУД) обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней памяти. Командами МК51 регистр-указатель данных может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

Таймер/счетчик. В составе средств МК51 имеются регистровые пары с символическими именами THO, TLO и THI, TLI, на основе которых функционируют два независимых программно-управляемых 16-битных таймера/счетчика событий.

Буфер последовательного порта. Регистр с символическим именем SBUF представляет собой два независимых регистра - буфер приемника и буфер передатчика. Загрузка байта в SBUF немедленно вызывает начало процесса передачи через последовательный порт. Когда байт считывается из SBUF, это значит, что его источником является приемник последовательного порта.

Регистры специальных функций. Регистры с символическими именами IP, IE, TMOD, TCON, SCON и PCON используются для фиксации и программного изменения управляющих бит и бит состояния схемы прерывания, таймера/счетчика, приемопередатчика последовательного порта и для управления мощностью электропитания МК51. Их организация будет описана ниже при рассмотрении особенностей работы МК51 в различных режимах.

Устройство управления и синхронизации

Кварцевый резонатор, подключаемый к внешним выводам X1 и X2 корпуса МК51, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации.

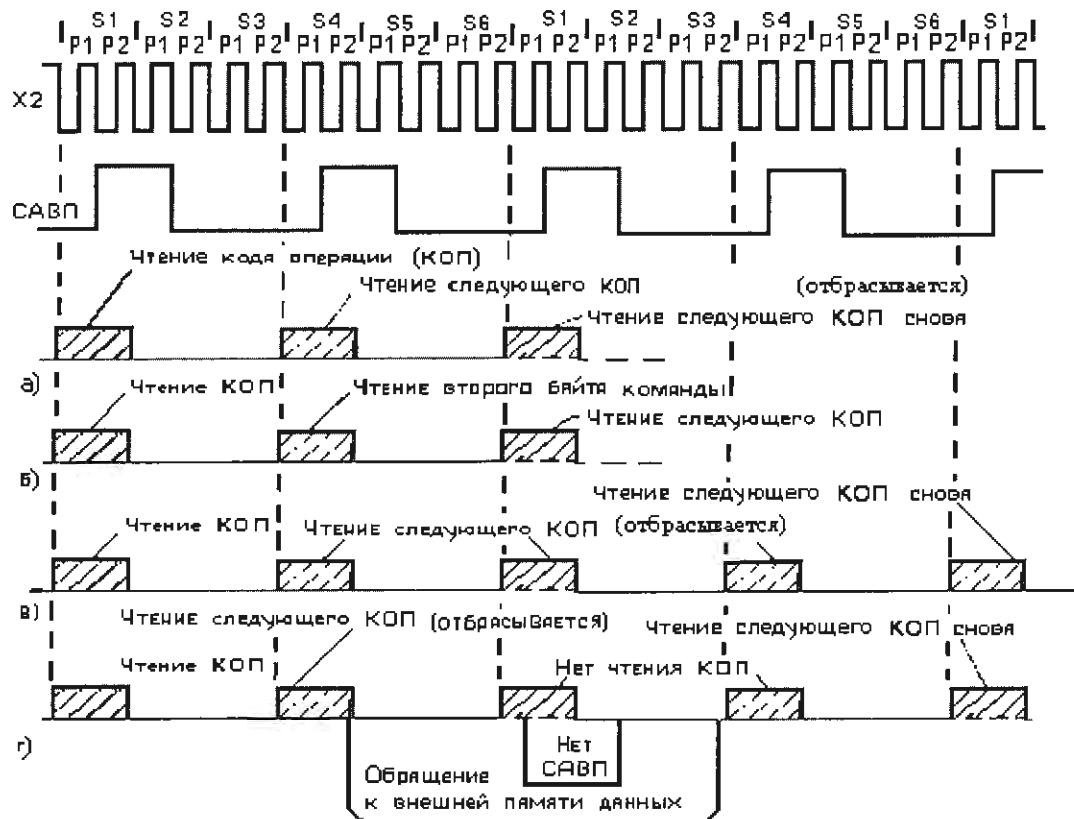


Рис. 3.3. Последовательности выборки и выполнения команд в МК51:
а - команда 1 байт / 1 цикл, например INC A ; б - команда 2 байта / 1 цикл, например ADD A, #d
в - команда 1 байт / 2 цикла, например INC DPTR ; г - команда 1 байт / 2 цикла, например MOVX

а - команда 1 байт/1 цикл, например INC A; б - команда - 2 байта/1 цикл, например ADD A, #d; в - команда 1 байт/2 цикла, например INC DPTR; г - команда 1 байт/2 цикла, например MOVX.

Устройство управления МК51 на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора или шести состояниям первичного управляющего автомата (S1_S6). Каждое состояние управляющего автомата содержит две фазы (P1, P2) сигналов резонатора. В фазе P1, как правило, выполняется операция в АЛУ, а в фазе P2 осуществляется межрегистровая передача. Весь машинный цикл состоит из 12 фаз, начиная с фазы S1P1 и кончая фазой S6P2, как показано на рис. 3.3. Эта временная диаграмма иллюстрирует работу устройства управления МК51 при выборке и исполнении команд различной степени сложности.

Все заштрихованные сигналы являются внутренними и недоступны пользователю МК51 для контроля. Внешними, наблюдаемыми сигналами являются только сигналы резонатора и строба адреса внешней памяти. Как видно из временной диаграммы, сигнал САВП формируется дважды за один машинный цикл (S1P2-S2P1 и S4P2-S5P1) и используется для управления процессом обращения к внешней памяти.

Большинство команд МК51 выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления МК51 производится расчет времени исполнения прикладных программ.

Общие сведения о системе команд

Система команд МК51 содержит 111 базовых команд, которые удобно разделить по функциональному признаку на пять групп: команды передачи данных, арифметических операций, логических операций, передачи управления и операций с битами.

Система команд МК51 много мощнее и шире системы команд МК48, так как кроме всех команд МК48 в ее состав входят команды умножения, деления, вычитания, операций над битами, операций со стеком и расширенный набор команд передачи управления. Большинство команд (94) имеют формат один или два байта и выполняются за один или два машинных цикла. При тактовой частоте 2 МГц длительность машинного цикла составляет 1 мкс. На рис. 3.19 показаны 13 типов команд МК51. Первый байт команды любых типа и формата всегда содержит код операции (КОП). Второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

Типы операндов. Состав операндов МК51 шире, чем МК48, и включает в себя операнды четырех типов: биты, 4-битные цифры, байты и 16-битные слова.

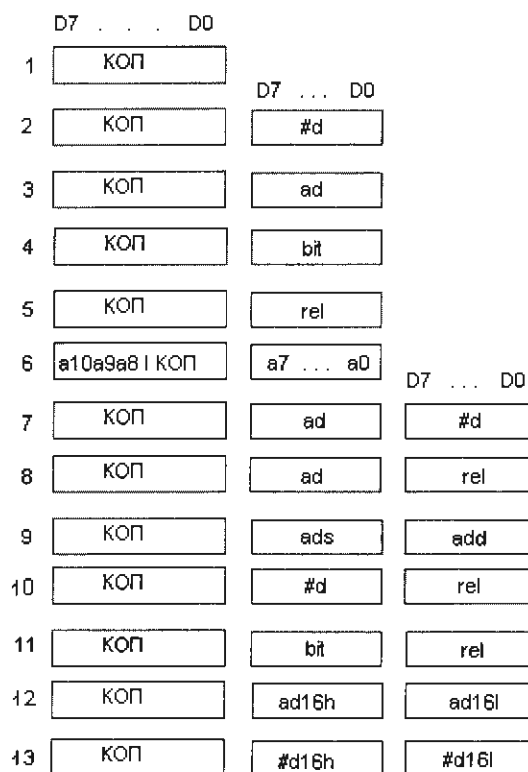


Рис.3.19. Типы команд МК51

В отличие от МК48, который имеет только три битовых флага, МК51 имеет 128 программно-управляемых флагов пользователя. Имеется также возможность адресации отдельных бит блока регистров специальных функций и портов. Для адресации бит используется прямой 8-битный адрес (bit). Косвенная адресация бит невозможна. Карты адресов отдельных бит представлены на рис.3.20 и рис.3.21. Четырехбитные операнды используются только при операциях обмена (команды SWAP и XCHD). Восьмибитным операндом может быть ячейка памяти программ или данных (резидентной или внешней), константа (непосредственный операнд), регистры специальных функций (РСФ), а также порты ввода/вывода. Порты и РСФ адресуются только прямым способом. Байты памяти могут адресоваться также и косвенным образом через адресные регистры (R0, R1, DPTR и PC). Двухбайтные

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

Способы адресации данных. В МК51 используются такие же способы адресации данных, как и в МК48: прямая, непосредственная, косвенная и неявная. Следует отметить, что при косвенном способе адресации РПД в отличие от МК48 используются все восемь бит адресных регистров R0 и R1.

Система команд МК51 по сравнению с МК48 допускает больше комбинаций способов адресации операндов в командах, что делает ее более гибкой и универсальной.

Флаги результата. Слово состояния программы (PSW) включает в себя четыре флага: C - перенос, AC - вспомогательный перенос, O - переполнение и P - паритет.

Флаг паритета (отсутствует в МК48) напрямую зависит от текущего значения аккумулятора. Если число единичных бит аккумулятора нечетное, то флаг P устанавливается, а если четное - сбрасывается. Попытки изменить флаг P, присваивая ему новое значение, будут безуспешными, если содержимое аккумулятора при этом останется неизменным.

Флаг AC устанавливается в случае, если при выполнении операции сложения/вычитания между тетрадами байта возник перенос/заем.

Флаг C устанавливается, если в старшем бите результата возникает перенос или заем. При выполнении операций умножения и деления флаг C сбрасывается. Флаг OV (отсутствует в МК48) устанавливается, если результат операции сложения/вычитания не укладывается в семи битах и старший (восьмой) бит результата не может интерпретироваться как знаковый. При выполнении операции деления флаг OV сбрасывается, а в случае деления на ноль

устанавливается. При умножении флаг *OV* устанавливается, если результат больше 255.

В табл. 3.1 перечисляются команды, при выполнении которых модифицируются флаги результата. В таблице отсутствует флаг паритета, так как его значение изменяется всеми командами, которые изменяют содержимое аккумулятора. Кроме команд, приведенных в таблице, флаги модифицируются командами, в которых местом назначения результата определены *PSW* или его отдельные биты, а также командами операций над битами.

Таблица 3.1. Команды, модифицирующие флаги результата

Команды	Флаги	Команды	Флаги
ADD	C, OV, AC	CLR C	C = 0
ADDC	C, OV, AC	CPL C	C = ^M C
SUBB	C, OV, AC	ANL C, b	C
MUL	C = 0, OV	ANL C, /b	C
DIV	C = 0, OV	ORL C, b	C
DA	C	ORL C, /b	C
RRC	C	MOV C, b	C
RLC	C	CJNE	C
SETB C	C = 1		

Символическая адресация. При использовании ассемблера МК51 (*ASM51*) для получения объектных кодов программ допускается применение в программах символических имен регистров специальных функций (*PCF*), портов и их отдельных бит (рис. 3.21).

Для адресации отдельных бит *PCF* и портов (такая возможность имеется не у всех *PCF*) можно использовать символическое имя бита следующей структуры:

Например, символическое имя пятого бита аккумулятора будет следующим: ACC.5. Символические имена РСФ, портов и их бит являются зарезервированными словами для ASM51, и их не надо определять с помощью директив ассемблера.

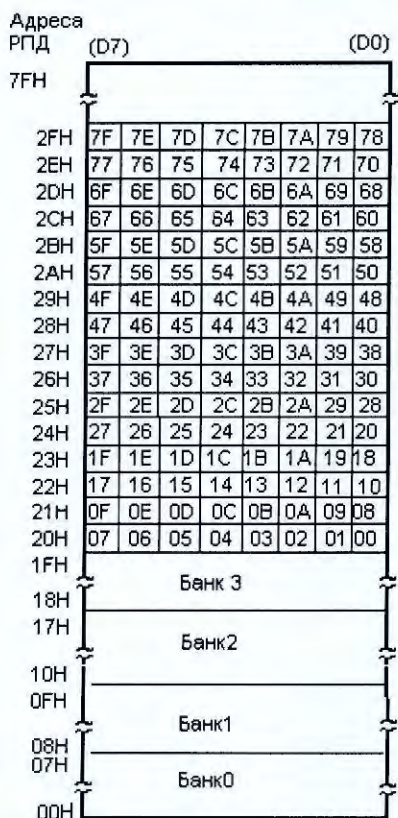


Рис. 3.20. Карта адресуемых бит в резидентной памяти данных

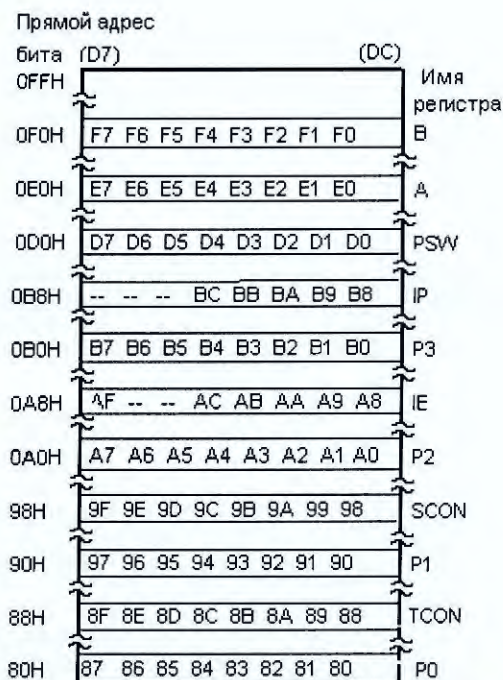


Рис. 3.21. Карта адресуемых бит в блоке регистров специальных функций

Порты ввода/вывода информации

Все четыре порта МК51 предназначены для ввода или вывода информации побайтно. Схемотехника портов ввода/вывода МК51 для одного бита показана на рис. 3.4 (порты 1 и 2 имеют примерно такую же структуру, как и порт 3). Каждый порт содержит управляемые регистр-зашелку, входной буфер и выходной драйвер.

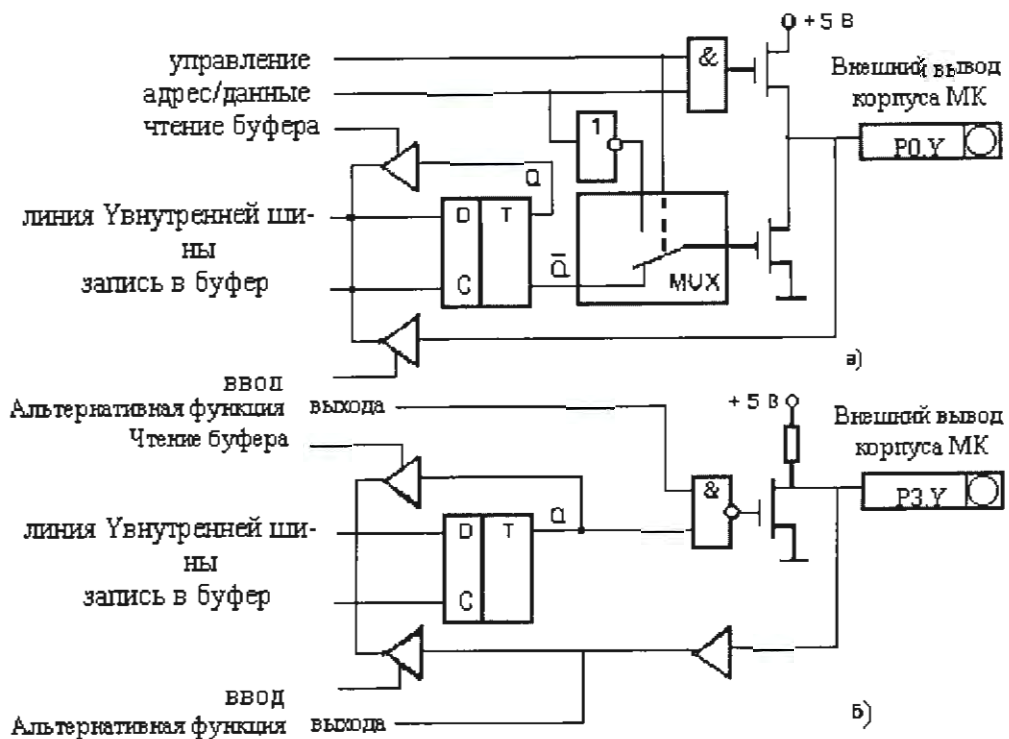


Рис. 3.4. Схематехника портов ввода/вывода МК51:
а- порт 0 ; б- порт 3

Выходные драйверы портов 0 и 2, а также входной буфер порта 0 используются при обращении к внешней памяти (ВП). При этом через 1 порт 0 в режиме временного мультиплексирования сначала выводится младший байт адреса ВП, а затем выдается или принимается байт данных. Через порт 2 выводится старший байт адреса в тех случаях, когда 1 разрядность адреса равна 16 бит.

Правила разработки программ на языке Ассемблер

Разработка прикладного программного обеспечения на языке Ассемблере - творческая задача, требующая от программиста отличного знания программистской модели МК-51, состава и формата команд, способов адресации операндов и т.д. Однако существуют формальные правила составления программ, соблюдение которых позволяет даже начинающему разработчику составлять работоспособные программы.

Подробно эти правила рассмотрены в [1], согласно которым для получения текста исходной программы необходимо выполнить следующую последовательность действий:

- составить подробное описание исходной задачи;
- выполнить инженерную интерпретацию задачи, желательно с привлечением того или иного аппарата формализации (сети Петри, графа автомата и т.п.);
- разработать блок-схемы алгоритма работы МПС;
- разработать детализированные блок-схемы алгоритмов отдельных процедур, выделенных на основе модульного принципа составления программ; распределить рабочие регистры и память МК-51;
- составить текст исходной программы. Исходный текст программы на языке Ассемблера имеет определенный формат, состоящий, обычно, из четырех полей:

МЕТКА МНЕМОНИКА ОПЕРАНД КОММЕНТАРИЙ.

Поля отделяются друг от друга произвольным числом пробелов. Метка. Метка - это имя, состоящее из букв латинского алфавита и стоящее в начале строки. После метки ставится двоеточие. В директивах Ассемблера EQU, DB и DW метка принимает значение выражения, стоящего за директивой. В остальных случаях метка принимает значение \$ (текущее значение счетчика команд). Перед директивой ORG использование меток запрещено.

Мнемоника. В поле Мнемоника записывается мнемоническое обозначение команды МК-51 или псевдокоманды ассемблера.

Операнды. В этом поле указываются операнды, участвующие в операции. Команды ассемблера могут быть без-, одно- или двухоперандными. Операнды разделяются запятой.

Операнд может быть задан непосредственно или в виде его адреса (прямого или косвенного). Непосредственному операнду обязательно должен предшествовать префикс (#). В качестве непосредственного

операнда можно указывать число или символическое имя. Прямой адрес операнда может быть задан мнемоническим обозначением, числом или символическим именем. Указанием на косвенную адресацию служит префикс @. В командах передачи управления операндом может быть число, метка, косвенный адрес или выражение.

Используемые в качестве операндов символические имена должны быть определены в программе, а числа представлены с указанием системы счисления, для чего используется суффикс (буква, стоящая после числа): В - для двоичной, Q - для восьмеричной, H - для шестнадцатеричной. Число без суффикса по умолчанию считается десятичным.

Комментарий. Поле комментария всегда начинается после символа (;) и игнорируется Ассемблером. В нем допускается использовать любые символы. Чаще всего это поле используется программистом для пояснения логической организации программы.

Директивы Ассемблера. Строка программы может содержать директиву или команду. Директивы, в отличие от команд, не исполняются МК-51 и предназначены, в основном, для управления трансляцией программы. Ниже приводится список директив Ассемблера, которые поддерживает редактор, встроенный в эмулятор микроконтроллера: ORG - изменение текущего значения счетчика команд;

После составления текста программы необходимо получить объектный код, т.е. набор двоичной информации, содержащий коды команд и данных. Для простых программ объектный код может быть получен вручную. Однако для более сложных программ требуются специальные средства, позволяющие осуществить трансляцию программы в автоматическом режиме. В настоящее время для МК-51 серии 1830 разработано значительное количество кросс - средств, которые не только обеспечивают ввод и трансляцию программ

составленных на языке Ассемблер, но и позволяющие проводить отладку программ.

Отладку разработанной программы рекомендуется выполнять с помощью эмулятора однокристальных микроконтроллеров Single-Chip machine.

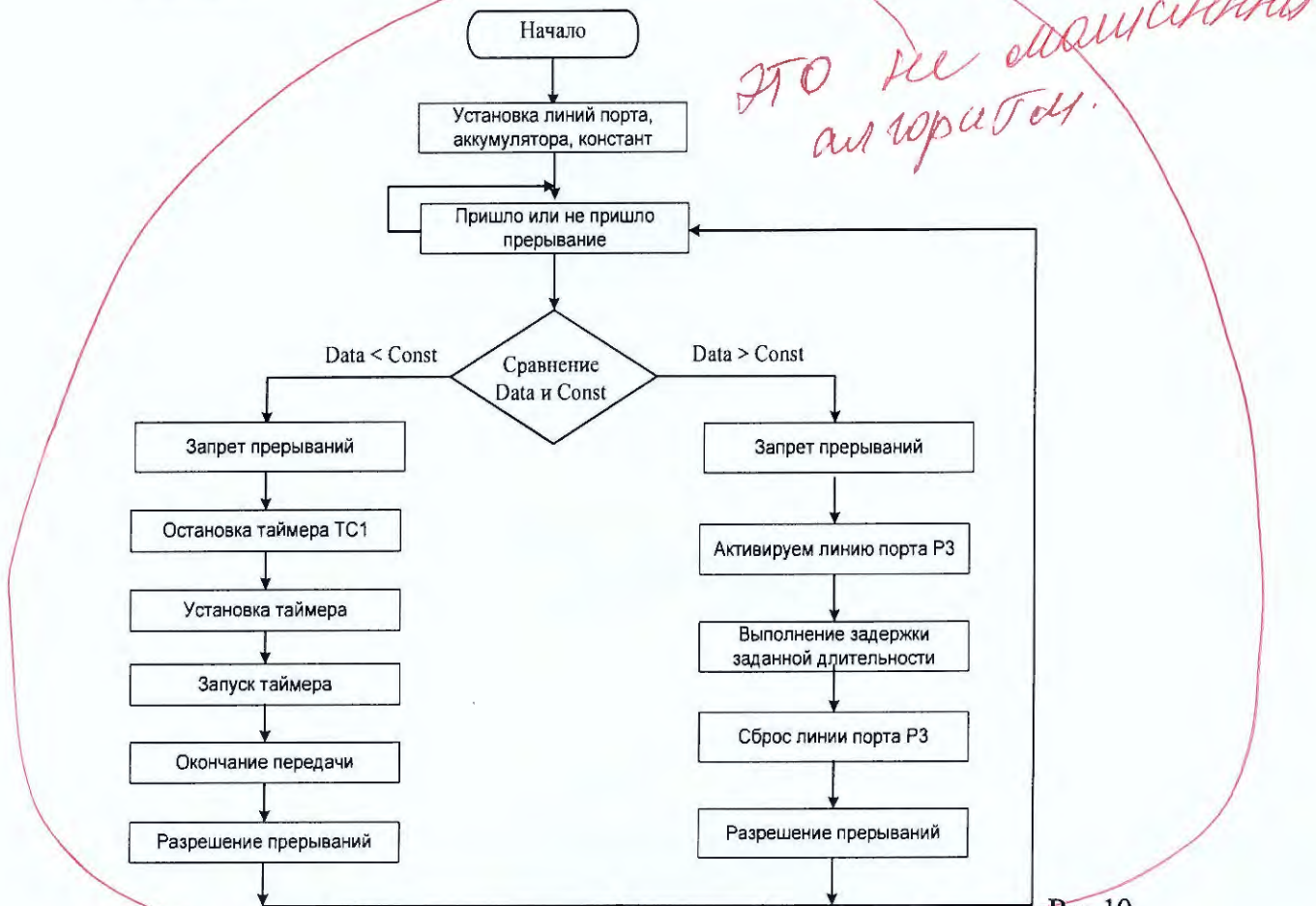


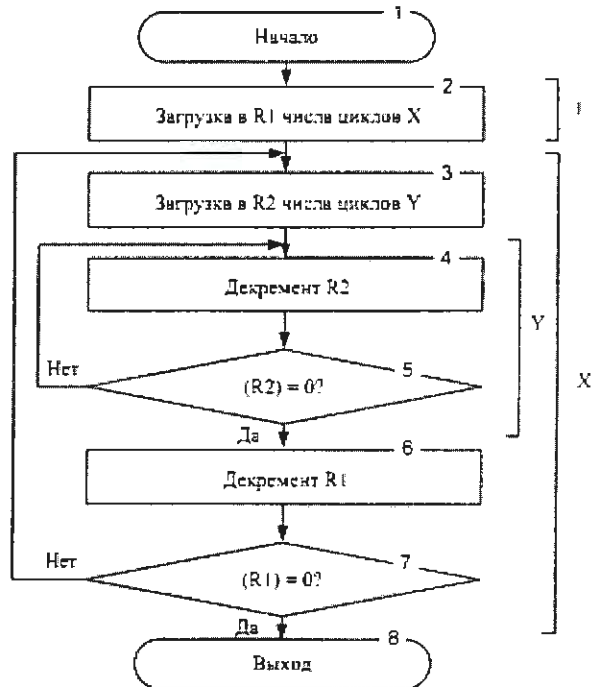
Рис.10

Машинный алгоритм выполнения программы

Последовательный порт МК-51 может использоваться в качестве универсального асинхронного приемо-передатчика (УАПП) с фиксированной или переменной скоростью последовательного обмена и возможностью дуплексного включения. Скорость последовательного обмена УАПП в режимах 1 и 3 определяется по формуле:

$$F_{n1} = F_{n3} = (2^{SMOD} \cdot f_{BQ}) / (32 \cdot 12 \cdot (256 - (TH))) \quad (1.1)$$

Во многих применениях МК-51 необходимо сформировать импульс заданной длительности на той или иной линии порта. Обычно для этих целей используется либо таймер/счетчик, либо регистры РПД. Блок-схема алгоритма формирования временной задержки большой длительности показана на рис. 11:



Данный алгоритм для формирования временной задержки использует так называемые вложенные циклы. На начальном этапе в регистр R1 загружается число “большого” цикла X (блок 2), а в регистр R2 – число “малого” цикла Y (блок 3). Эти две команды выполняются за один машинный цикл. С помощью блоков 4, 5 подпрограммы содержимое регистра R2 уменьшается до 0 и осуществляется переход к блоку 6 подпрограммы. Этот цикл удобнее организовать с помощью команды DJNZ R, \$, которая выполняется МК-51 за 2 машинных цикла. Аналогично организуется и “большой” цикл (блоки 6, 7).

Исходные данные:

число каналов ввода	28_{10}	$1C_{16}$
число каналов вывода	10_{10}	A_{16}
Объем памяти ПЗУ, Кбайт	8	
Объем памяти ОЗУ, Кбайт	16	
Частота синхро, МГц	10,5	
Длител. импульса, мс	25	
№ банка памяти	0	
№ линии порта P3	6_{10}	6_{16}
Скорость передачи, Кбайт	14400	
прерывание по	int1	

Последовательный порт МК-51 может использоваться в качестве универсального асинхронного приемо-передатчика (УАПП) с фиксированной или переменной скоростью последовательного обмена и возможностью дуплексного включения. Скорость последовательного обмена УАПП в режимах 1 и 3 определяется по формуле:

$$F_{n1} = F_{n3} = \frac{(2^{SMOD} \times f_{BQ})}{32 \times 12 \times [256 - (TH)]}, TH = 256 - \frac{(2^{SMOD} \times f_{BQ})}{32 \times 12 \times F_{n1}}$$

где SMOD - значение бита SMOD регистра управления мощностью PCON;

- fBQ - частота синхронизации МК-51;
- TH- десятичное значение содержимого регистра TH1.

$$TH = 256 - \frac{2^0 \times 10,5 \times 10^6}{32 \times 12 \times 14400} = 254_{10} = FE_{16}$$

Время задержки в секундах рассчитывается по формуле:

$$t_3 = \frac{1}{f_{BQ}} \times 12 [1 + (3 + 2 \times Y) \times X]$$

Так как задана длительность импульса, необходимо подобрать такие X и Y, чтобы выполнялось условие:

$$t_3 \leq t_{зад}$$

где tзад – заданная длительность импульса.

Методом подбора, определим значения X и Y:

$$t_3 = 25 \text{ мс}$$

$$12 + 12X(3 + 2Y) = t_3 f_{BQ}$$

$$12X(3 + 2Y) = t_3 f_{BQ} - 12$$

$$X = \frac{(t_3 f_{BQ} - 12)}{12(3 + 2Y)}$$

$$X = \frac{(25 \times 10^{-3} \times 10,5 \times 10^6 - 12)}{12 \times (3 + 2 \times 255)} = 42,6_{10} \quad 2A_{16}$$

число циклов должно быть не меньше.

Зная значения X и Y рассчитаем время задержки в секундах:

$$t_z = \frac{12 \times (1 + (3 + 2 \times 255) \times 42,6)}{10,5 \times 10^6} = 25,00 \text{ мс}$$

Для того чтобы МК-51 сформировал импульс с точностью плюс/минус 1 цикл в подпрограмму необходимо добавить команду “нет операции” (NOP), число которых можно рассчитать по формуле:

$$N = \frac{t_{зад} - t_z}{12} \times f_{BQ}$$

$$N = \frac{25 - 25}{12} \times 11 = 0$$

5. Листинг программы Single-Chip Machine.

```

data equ      1C  h
const equ     5   h
start:                ; начальная установка
clr A                ; очистка аккумулятора
mov R0,A            ; очистка регистра R0
mov R1,A            ; очистка регистра R1
mov R2,A            ; очистка регистра R2
mov TH1,A           ; очистка таймера-счетчика
mov PSW,A           ; очистка регистра состояния программы
clr P3. 6            ; сброс линии порта
mov P1,#FFh         ; настройка порта на ввод
mov IE,A            ; сброс регистра IE
mov IP,A            ; очистка регистра приоритета прерываний
mov TCON,A          ; очистка регистра TCON
mov TMOD,A          ; очистка регистра T/C
LOOP:                ; загрузка данных
mov TCON,#01h       ; прерывание по заднему фронту
mov IE,# 84  h      ; прерывание по int 1
mov IP,#0 4  h      ; установка приоритета прерывания
mov PSW,# 00  h     ; выбор банка памяти
Main:                ; метка
mov P1,# 1C  h      ; загрузка в порт числа
mov A, P1           ; загрузка числа в аккумулятор
mov R0,# 0 5  h     ; загрузка в регистр R0 константы
subb A,R0           ; сравнение R0 с аккумулятором
jnc Impuls          ; переход к формированию импульса
subb SBUF,A         ; передача в последовательный порт
clr TR1             ; остановка таймера T/C1

```

установка св. по дефолту где же операция?

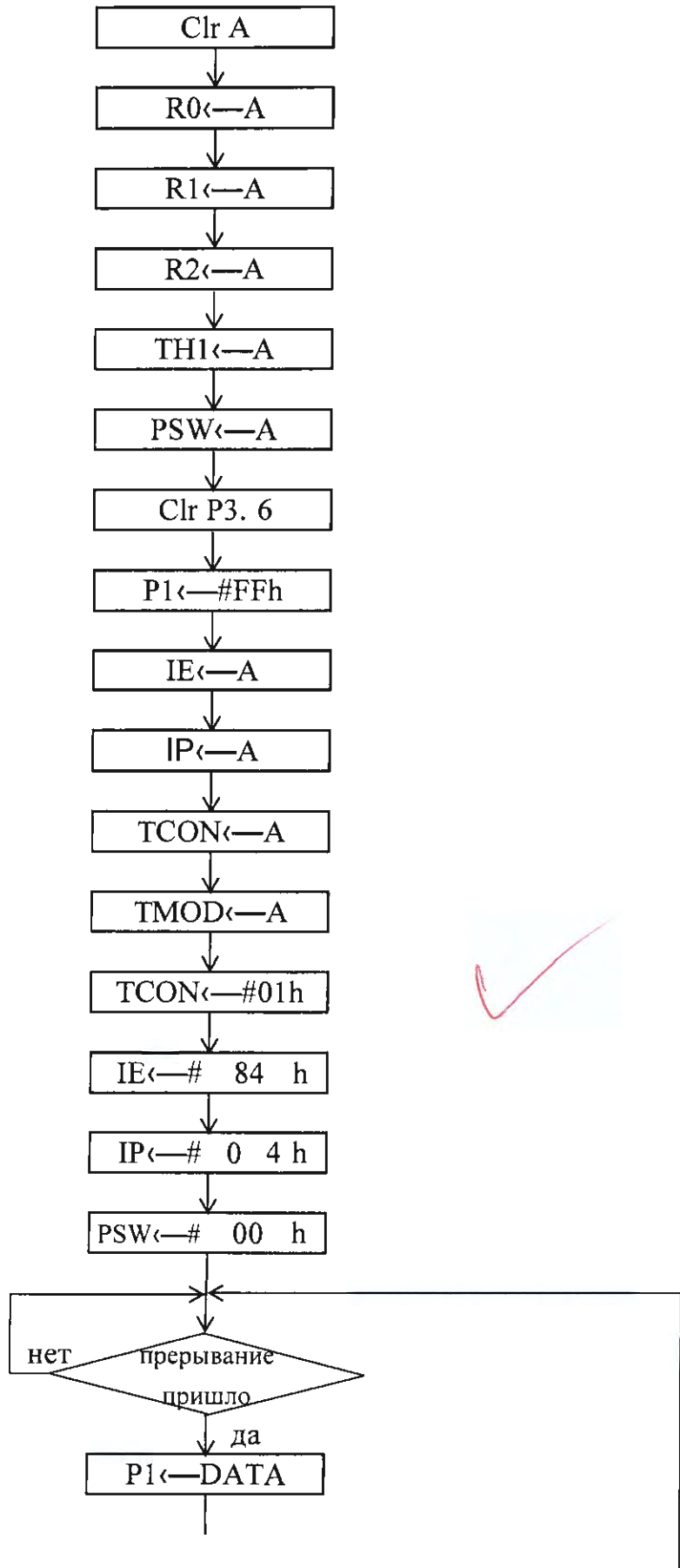
зачем загрузка в PSW даночк. 01

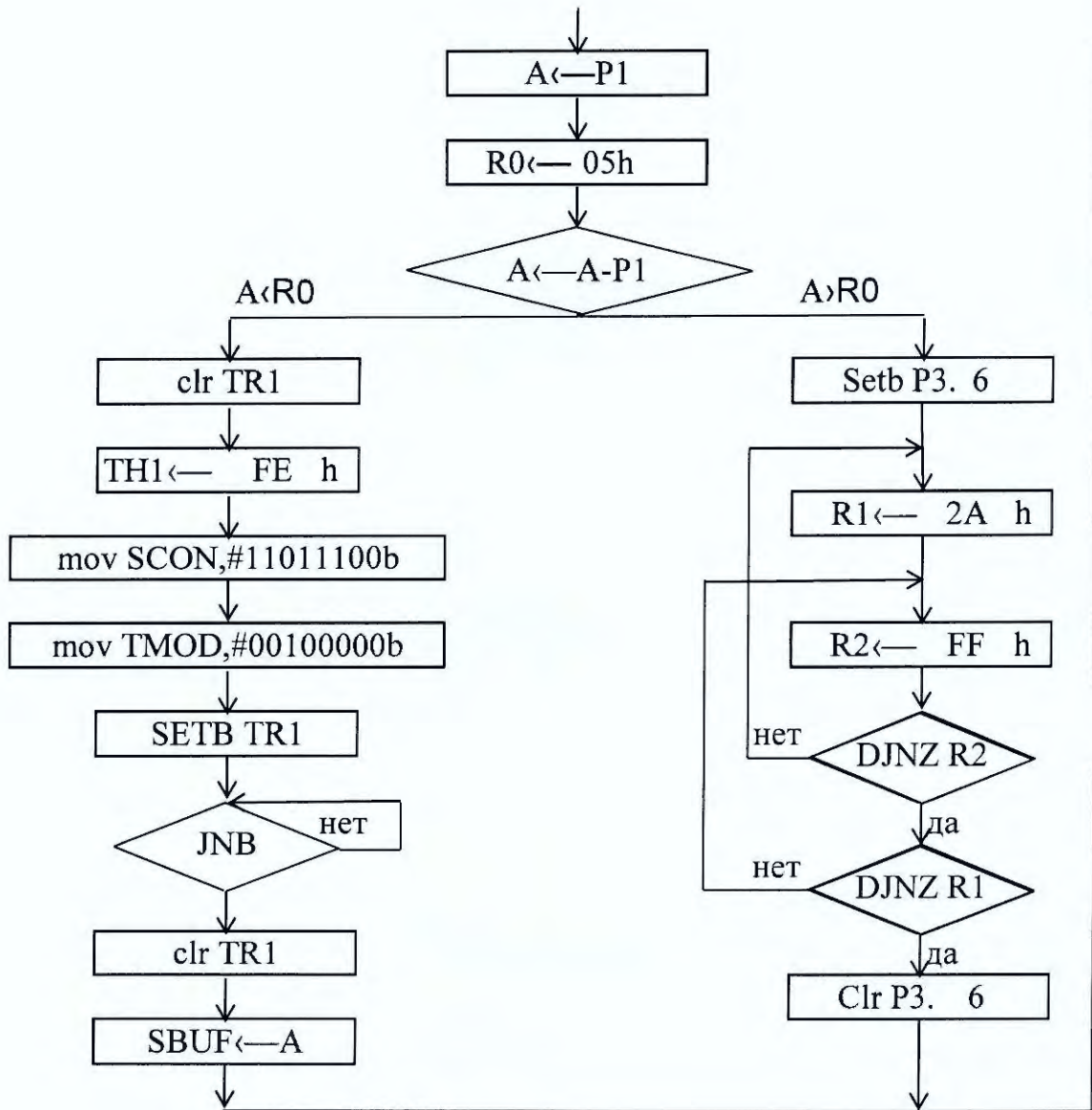
```

mov TH1,# FE h      ;загрузка значения скорости
mov SCON,#11011100b ;установка режима 3 УАПШ
mov TMOD,#00100000b ;установка режима 2T/C1
setb TR1            ;запуск таймера
label:
jnb T1,label        ;ожидание окончания передачи
clr T1              ;очистка флага передачи
Impuls:
Setb P1. 6
mov R1,# 2A h      ;загрузка значения X
M0:
mov R2,# FF h     ;загрузка значения Y
M1:
djnz R2,M1        ;декремент R2
djnz R1,M0        ;декремент R1
nop               ;пустая операция
Clr P3            ;сброс линий порта P3
ajmp Main         ; переход по метке Main

```

а далее см. алгоритм.





Машинный алгоритм выполнения программы

6. Список литературы.

1. Калабеков Б.А. Цифровые устройства и микропроцессорные системы. Учебник .- М.:Горячая линия – Телеком, 2002 г.
2. Токхайм Р. Микропроцессоры. Курс и упражнения/Пер. с англ., под. ред. В.Н. Грасевича. М.:Энергоатомиздат, 1988 – 336 с.
3. Каган Ю.М., Сташин В.В. Основы проектирования микропроцессорных устройств автоматики. М.: Энергоатомиздат, 1987-304 с.
4. Каган Б.М. Электронные вычислительные машины и системы. – М.: Энергоатомиздат, 1991. – 592 с.
5. А.В. Горелик, В.Ю. Горелик, А.Е. Ермаков, О.П. Ермакова «Микропроцессорные информационно-управляющие системы железнодорожного транспорта», Москва 2011 г.
6. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах.- М.: Энергоатомиздат, 1990 г.
7. Однокристальные микроЭВМ./ Боборыкин А.В., Липовецкий Г.П. и др. М.: МИКАП, 1994.
8. Русак И.М. Луговский В.П.Технические средства ПЭВМ: Справочник/ Под ред. И.М. Русака. – Мн.: Выш. шк., 1996. – 504 с.
9. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник: В 2 т. Н.Н. Аверьянов, А.И. Березенко и др.; Под ред. В.А. Шахнова. - М.: Радио и связь, 1988 г.
- 10.Шило В.Л. Популярные цифровые микросхемы. :Справочник. – М.: Металлургия, 1989.