

**Министерство образования и науки РФ
ФГБОУ ВПО «Сибирская государственная автомобильно-дорожная
академия (СибАДИ)»**

**Кафедра автоматизации технологических процессов
и электротехники**

**ПРИНЦИП РАБОТЫ
МИКРОПРОЦЕССОРА K580BM80
И ЕГО СИСТЕМА КОМАНД**

Методические указания к лабораторным работам

Составитель: А-й А. Руппель, А-р А. Руппель

**Омск
Издательство СибАДИ
2012**

1. ОСНОВНЫЕ ПАРАМЕТРЫ МП К580ВМ80

Микропроцессор КР58ОИК80 выполнен по n-МОП технологии, выпускается в корпусе, имеющем сорок выводов. Цоколевка корпуса и назначение выводов представлены на рис.1.

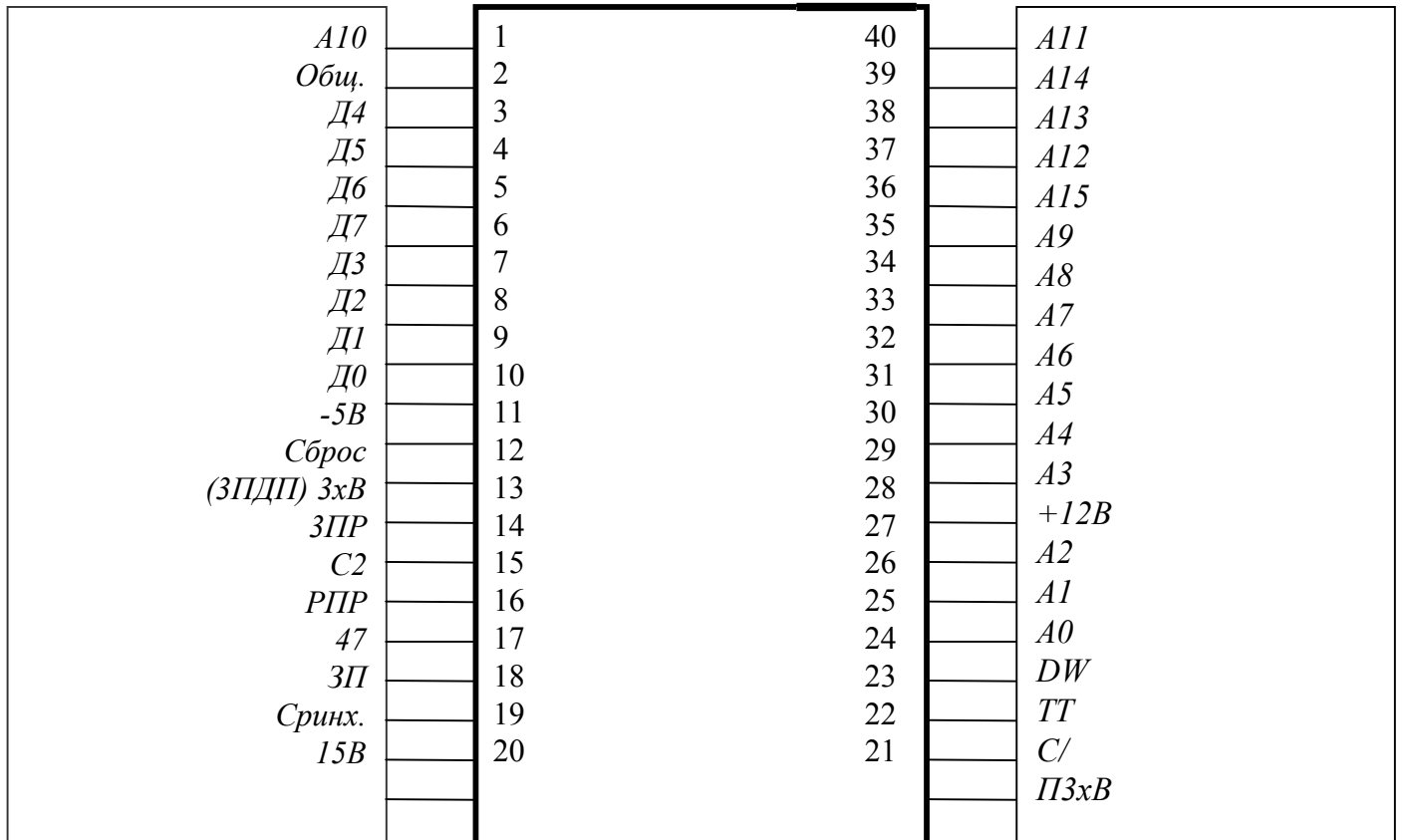


Рис.1. Назначение выводов МП КР580ИК80

Разрядность МП равна восьми битам. Это значит, что все регистры (кроме двух) и арифметико-логическое устройство — восьмиразрядные[1].

Электрические характеристики МП приведены в таблице. Из нее видно, что МП хорошо согласуется с ТТЛ-схемами. Для своей работы МП требует три источника питания, общая потребляемая мощность не превышает 1,5 Вт. МП имеет 16-разрядную адресную шину с тремя устойчивыми состояниями, с помощью которой можно адресовать 65536 байтов памяти и в до-

бавление к этому 256 портов вывода. МП обладает 8-разрядной двунаправленной шиной данных с тремя устойчивыми состояниями.

Таблица

Электрические параметры МП КР580ИК80

	минимальное	типичное	максимальное	
Низкий уровень синхросигналов, В	-1	0,5	0,8	При токе утечки
Высокий уровень синхросигналов, В	9	12	13	по входу не более
Низкий уровень входных сигналов, В	-1	0,4	0,8	± 10 мкА
Высокий уровень входных сигналов, В	3,3	4,5	6	
Низкий уровень выходного сигнала, В	0,2	0,3	0,45	Ток нагрузки не более 1,9 мА
Высокий уровень выходного сигнала, В	3,7	4,5	5	Ток нагрузки не более 150 мкА
Ток потребляемый от источников питания, мА: +5В -5В +12В	— — —	60 0,01 40	80 1,0 70	При τ $C_1 = 480$ нс

МП имеет ряд особенностей в своих электрических характеристиках: уровень лог.1 сигналов $C1$ и $C2$ должен быть не ниже 9В, остальных входных сигналов — не менее 3,3 В; нагрузочная способность всех выходов составляет 1,9 мА при лог.0 на выходе и 150 мкА при лог.1.

МП КР580ИК80 имеет средство для организации режимов прерывания, предоставления своих шин другому устройству и ожидания готовности внешних устройств, с которыми идет обмен информацией. Система команд МП насчитывает 78 основных команд. Время выполнения коротких команд

Блок регистров. МП имеет следующие внутренние регистры:

- шесть 8-разрядных регистров общего назначения (*РОН*) *B, C, D, E, H, L*;
- 16-разрядный счетчик команд (*СчК*) или программный счетчик (*ПС*);
- 16-разрядный указатель стека (*УС*);
- два 8-разрядных регистра временного хранения.

Счетчик команд служит для хранения адреса команд, его содержимое автоматически инкрементируется после выборки из памяти очередной команды.

Указатель стека служит для хранения адреса последнего загруженного в стек байта, т.е. адреса верхнего элемента стека. Начальное значение указателя стека может быть выбрано программистом, т. е. программист может по своему усмотрению располагать стек в любом месте памяти. При выполнении операции загрузки в стек очередного элемента содержимое указателя стека декрементируется, а при извлечении элемента из стека — инкрементируется.

Схема инкремента — декремента (*СхИД*) осуществляет операции инкрементирования и декрементирования содержимого любого регистра из матрицы регистров. В системе команд МП существуют специальные команды инкремента и декремента, при выполнении которых *СхИД* включается в работу.

Схема выбора регистра (*СхВР*) осуществляет выбор требуемого регистра для передачи его содержимого во внутреннюю шину данных или на схему инкремента-декремента, а также при обратных передачах информации в регистр.

В некоторых случаях регистры общего назначения могут попарно объединяться в 16-разрядные регистры *BC, DE, HL*. Такие попарно объединенные (конкатенированные) регистры называются регистровыми парами, соответственно *B, D* и *H*.

Арифметико-логическое устройство (АЛУ). Служит для выполнения арифметических и логических операций над 8-разрядными операндами. АЛУ обслуживается 8-разрядными регистрами временного хранения операнда (*PzBpX*), двухтактным регистром-аккумулятором (*PzA*), регистром признаков (*PzП*).

Регистр-аккумулятор выступает в качестве регистра операнда и регистра результата одновременно. Все команды арифметических и логических операций используют неявную адресацию первого операнда и результата. Во всех этих командах первый операнд — аккумулятор, результат выполнения операций записывается также в аккумулятор.

Регистр признаков (*PzП*) содержит следующие пять признаков результата: нулевого результата (*Z*); переноса займа (*CY*); единичного значения старшего разряда результата (*S*); четности числа единиц в байте результата (*P*); вспомогательного переноса между тетрами баята (*AC*).

Более детальное описание регистра признаков будет дано при рассмотрении системы команд.

Схема десятичной коррекции (*CxDK*) позволяет производить коррекцию результата при арифметических операциях над двоично-десятичными числами. В своей работе использует признак вспомогательного переноса. Десятичная коррекция производится по специальной команде МП (*DAA*).

Регистр команд (PzK). Служит для приема кода операции (*KOP*), расположенного в первом байте команды, и хранения его в течение всего времени выполнения команды. Содержимое регистра команд постоянно передается на входы дешифратора команд.

Буфер шины данных (БД). Служит для изоляции внутренней шины данных МП от внешней интерфейсной шины данных, реализует двунаправленный обмен данными между МП и внешними схемами, обеспечива-

ет необходимую нагрузочную способность, по сигналу захвата переключается в третье состояние.

Буфер шины адреса (БА). Служит для передачи адреса на внешние адресные линии интерфейса МП, обеспечивает необходимую нагрузочную способность, по сигналу захвата переключается в третье состояние. Передача адреса ведется всегда в одном направлении — от МП к внешним схемам, обратная передача невозможна.

Устройство управления (УУ). Служит для выработки внутренних сигналов синхронизации отдельных узлов МП, приема внешних управляющих сигналов, формирования и выдачи сигналов управления внешними схемами. Основу синхронизации МП составляют два внешних сигнала синхронизации, поступающие на входы *С1* и *С2* схемы интерфейсных сигналов. Именно эта схема обеспечивает выполнение таких процедур, как прерывание, захват шин МП, ожидание готовности внешнего устройства. Описание и временные диаграммы этих процедур отражены в разд. 5-8.

Дешифратор команд (ДшК) и схема управления машинным циклом. Код операции, хранящийся в регистре команд, поступает на дешифратор команд. Схема управления машинным циклом, используя результаты дешифрации команды, обеспечивает отработку МП определенных машинных циклов, необходимых для выполнения команды. Выходы дешифратора команд в комбинации с различными временными сигналами образуют управляющие сигналы для настройки различных узлов МП на выполнение требуемых операций.

Таким образом, всей работой МП управляют дешифратор команд совместно со схемой управления машинным циклом и схема интерфейсных сигналов, входящие в состав устройства управления (первичного управляющего автомата).

3. РАБОЧИЙ ЦИКЛ И СИНХРОНИЗАЦИЯ МП

Последовательность действий МП по выполнению одной команды называется циклом команды. Цикл команды состоит из одного или нескольких машинных циклов (максимум из пяти). Машинный цикл составляет основу всей работы МП (подробно машинный цикл описан в разд.5). Обычно в течение машинного цикла обязательно происходит передача байта информации между МП и внешними устройствами (под внешними устройствами понимается память и устройства ввода-вывода). Исключением является команда *DAD*, цикл команды которой включает в себя три машинных цикла, два из которых требуются для выполнения только внутренних операций суммирования.

Каждый машинный цикл состоит из трех, четырех или пяти машинных тактов. Машинный такт — период внешнего синхросигнала. Для синхронизации МП используются два сигнала — *C1* и *C2*. Временная диаграмма синхросигналов представлена на рис.3.

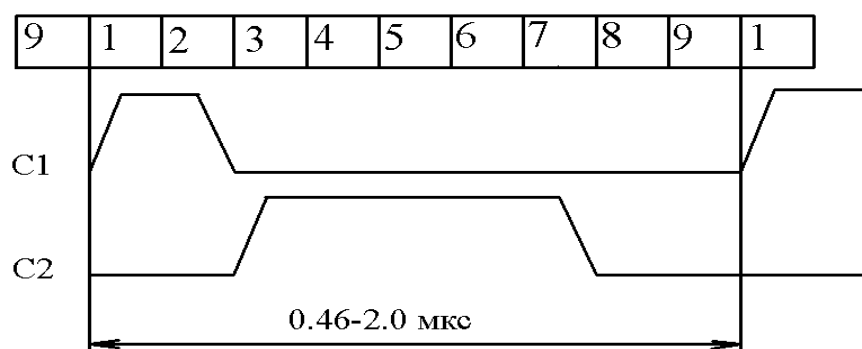


Рис.3. Временная диаграмма сигналов *C1* и *C2*.

Машинный такт принято отсчитывать от момента появления сигнала *C1*. Минимальное значение периода синхросигналов равно 480 нс., максимальное — 2 мкс.

Единичный уровень сигналов *C1* и *C2* должен быть не менее 9В. Обычно генератор, вырабатывающий синхросигналы, стабилизируется с помощью кварцевого резонатора. Это позволяет точно определять время выполнения команд, так как, известно что, сколько машинных тактов занимает, выполнение той или иной команды.

Цикл любой команды начинается с машинного цикла ВЫБОРКА (разд.4), в течение которого происходит выборка из памяти первого байта команды. Если команда имеет формат одного байта и операнды, располагаются во внутренних регистрах МП, то в этом же машинном цикле происходит и выполнение операции, заданной командой. Если же формат команды равен двум или трем байтам, то следующими будут выполняться машинные циклы чтения из памяти остальных байтов команды. Если операнды располагаются в памяти, то затем будут выполнены машинные циклы чтения операндов. Выполнение операций, предписанных командой, происходит по мере извлечения операндов в соответствующих машинных циклах. Иногда может потребоваться машинный цикл записи в память содержимого внутреннего регистра.

4. ТИПЫ МАШИННЫХ ЦИКЛОВ МП

Выполнение любой команды сводится к выполнению определенного числа определенных машинных циклов МП. Всего существует десять типов машинных циклов, все они связаны с передачей данных между МП и памятью или между МП и устройствами ввода-вывода.

В начале каждого машинного цикла МП выставляет на шину данных слово состояния машинного цикла (*ССМЦ*), которое несет всю информацию о типе текущего машинного цикла. Появление *ССМЦ* на шине данных

синхронизируется МП посредством специального сигнала СИНХР. На рис. 4 приведена временная диаграмма этого сигнала.

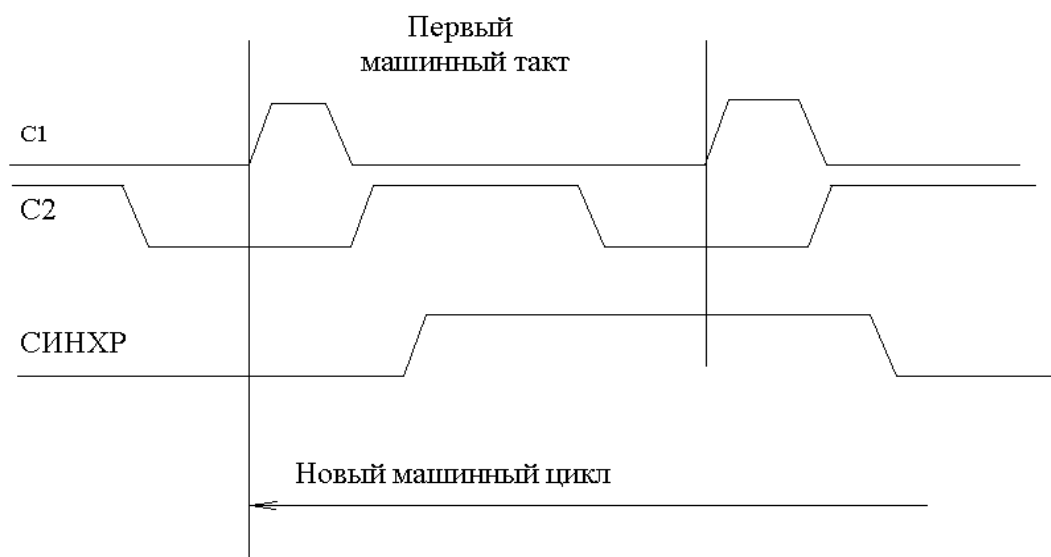


Рис.4. Временная диаграмма сигналов С1, С2 и СИНХР.

ССМЦ позволяет МП системе настроиться на выполнение определенного машинного цикла. *ССМЦ* присутствует на шине данных не постоянно, а только в течение определенного времени. Поэтому для сохранения *ССМЦ* на время всего машинного цикла необходимо запоминать *ССМЦ* в специальном регистре (*P₂ССМЦ*).

На рис.5 приведена временная диаграмма процесса записи *ССМЦ* в регистр *P₂ССМЦ*. Там же приведены функциональные обозначения отдельных разрядов *ССМЦ* и представлена кодировка всех десяти типов *ССМЦ*.

Разряды *ССМЦ* имеют следующие функциональные назначения:

ППР (D0) - подтверждение прерывания, используется для управления выдачей на шину данных кода команды *RST* (разд.7);

ЗПП (D1) - запись в память и вывод, указывает на то, что в машинном цикле будет выполняться операция записи данных в память или в устройство вывода;

СТЕК (Д2) - обращение к стеку, указывает, что в машинном цикле будет происходить запись в стек или чтение из стека, на адресной шине в этом машинном цикле находится содержимое *УС*,

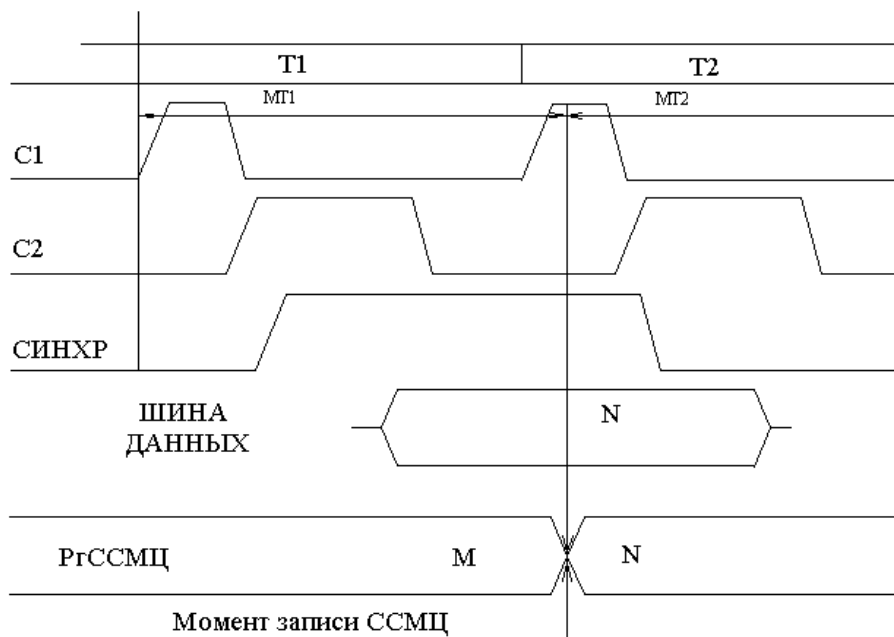
ПОСТ (Д3) - подтверждение останова, указывает на то, что МП перешел в состояние останова (по команде *HLT*);

ЗПВВ (Д4) - вывод, указывает на то, что в машинном цикле будет происходить операция записи данных в устройство вывода;

М1 (Д5) - выборка, указывает на то, что в машинном цикле будет происходить выборка первого байта команды из памяти;

ЧТВВ (Д6) - ввод, указывает на то, что в машинном цикле будет происходить чтение данных в МП из устройства ввода;

ЧТП (Д7) - чтение из памяти, указывает на то, что в машинном цикле будет происходить операция чтения из памяти;



Разряды ССМЦ	ШД	Выборка команды	Чтение из памяти	Запись в память	Чтение из стека	Запись в стек	Чтение из порта	Запись в порт	Прерывание	Останов	Прерывание в останове
Тип цикла		1	2	3	4	5	6	7	8	9	10
ППР	Д0	0	0	0	0	0	0	0	1	0	1
ЗПП	Д1	1	1	0	1	0	1	0	1	1	1
СТЕК	Д2	0	0	0	1	1	0	0	0	0	0
ПОСТ	Д3	0	0	0	0	0	0	0	0	1	1
ЗПВВ	Д4	0	0	0	0	0	0	1	0	0	0
М1	Д5	1	0	0	0	0	0	0	1	0	1
ЧТВВ	Д6	0	0	0	0	0	1	0	0	0	0
ЧТП	Д7	1	1	0	1	0	0	0	0	1	0

Рис. 5 Слово состояния машинного цикла: N, M – типы циклов МП

ССМЦ используется системным контроллером МП системы для выработки управляющих сигналов (подробно об этом см. разд.9).

5. МАШИННЫЙ ЦИКЛ МП

Машинный цикл может состоять из трех, четырех или пяти машинных тактов. На рис. 6 представлена диаграмма смены состояний МП в течение одного машинного цикла. Состояние МП в машинных тактах обозначены как $T1$, $T2$, $T3$, $T4$ и $T5$.

Кроме того, МП может находиться еще в трех особых состояниях: ожидания ($T_{ож}$), останова ($T_{ост}$), захвата ($T_{зах}$). Функционирование МП в этих состояниях будет рассмотрено ниже. Сейчас же предположим, что условия перехода в эти особые состояния не выполняются.

В состоянии $T1$ МП попадает после завершения очередного машинного цикла. Появление сигнала *СБРОС* также переведет МП в состояние $T1$. В этом состоянии МП передает на шину адреса адрес устройства ввода-вывода или ячейки памяти в зависимости от конкретного машинного цикла, на шину данных — слово состояния машинного цикла.

В состоянии $T2$ МП снимает с шины данных слово состояния машинного цикла, после чего шина данных переводится в режим ввода (для циклов чтения-ввода) или на шину данных выставляется информация (в случае циклов записи-вывода). В этом же состоянии МП проверяет наличие внешних сигналов *ГТ* и *ЗХВ* (*ЗПДП*) и внутреннего сигнала *ПОСТ* (подтверждение останова), а также производит инкрементирование программного счетчика.

В состоянии $T3$ осуществляется либо чтение-ввод данных, либо запись-вывод данных.

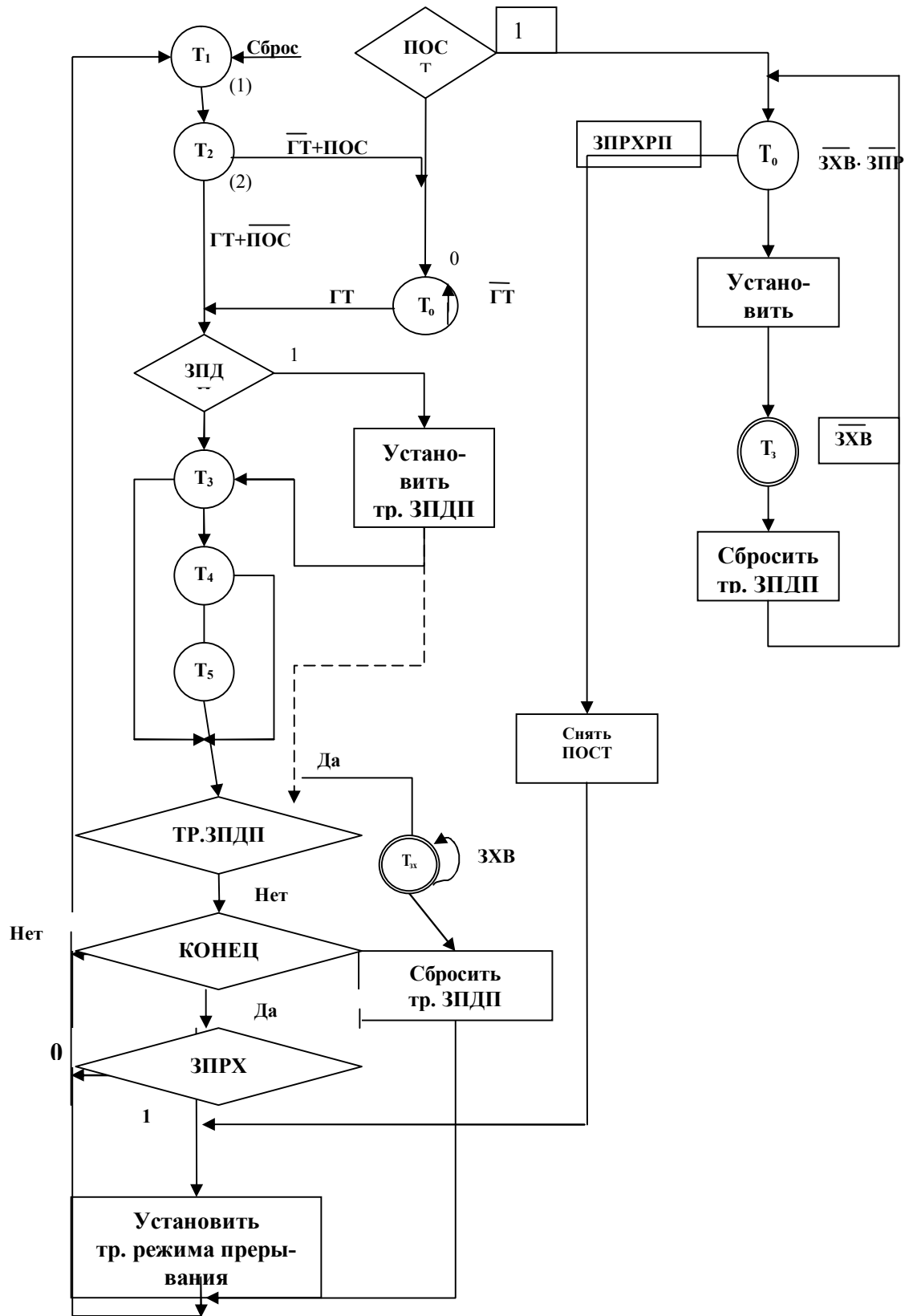


Рис.6. Диаграмма смены состояний МП при выполнении машинного цикла.

Состояния $T1$, $T2$, $T3$ обязательно присутствуют в каждом машинном цикле. Состояния $T4$ и $T5$ необходимы, если требуется выполнить действия по обработке данных внутри МП. Во время этих состояний могут выполняться внутренние пересылки или обработка данных в *АЛУ*. Однако, если этого не требуется, состояния $T4$ и $T5$ могут отсутствовать либо может иметь место только состояние $T4$.

После выхода из последнего состояния ($T3$, $T4$ или $T5$) МП переходит к выполнению нового машинного цикла; т. е. переходит в состояние $T1$. Работа МП при переходах в особые состояния будет рассмотрена ниже. Сейчас же главное уяснить, что в течение машинного цикла МП работает по жестко заданному алгоритму, последовательно проходя состояния $T1$ - $T5$. Любой из десяти различных машинных циклов выполняется по алгоритму, представленному на рис.6.

Выполнение любой команды сводится к выполнению необходимого числа определенных машинных циклов. Количество и типы машинных циклов однозначно определяются кодом операции команды. Таким образом, МП работает по жесткой схеме выполнения машинного цикла, а схема управления машинным циклом, используя сигналы с выхода дешифратора команд, указывает, какие машинные циклы, и в каком порядке должны быть выполнены. Внести изменения в этот алгоритм невозможно, поэтому данный МП называется МП с жесткой логикой управления. Все МП, обладающие фиксированным списком команд, имеют жесткую логику управления.

Рассмотрим временную диаграмму сигналов (рис.7) на линиях МП при выполнении машинного цикла чтения-ввода.

В первом машинном такте машинного цикла МП вырабатывает сигнал *СИНХР*, который принимает единичное значение с некоторой задержкой

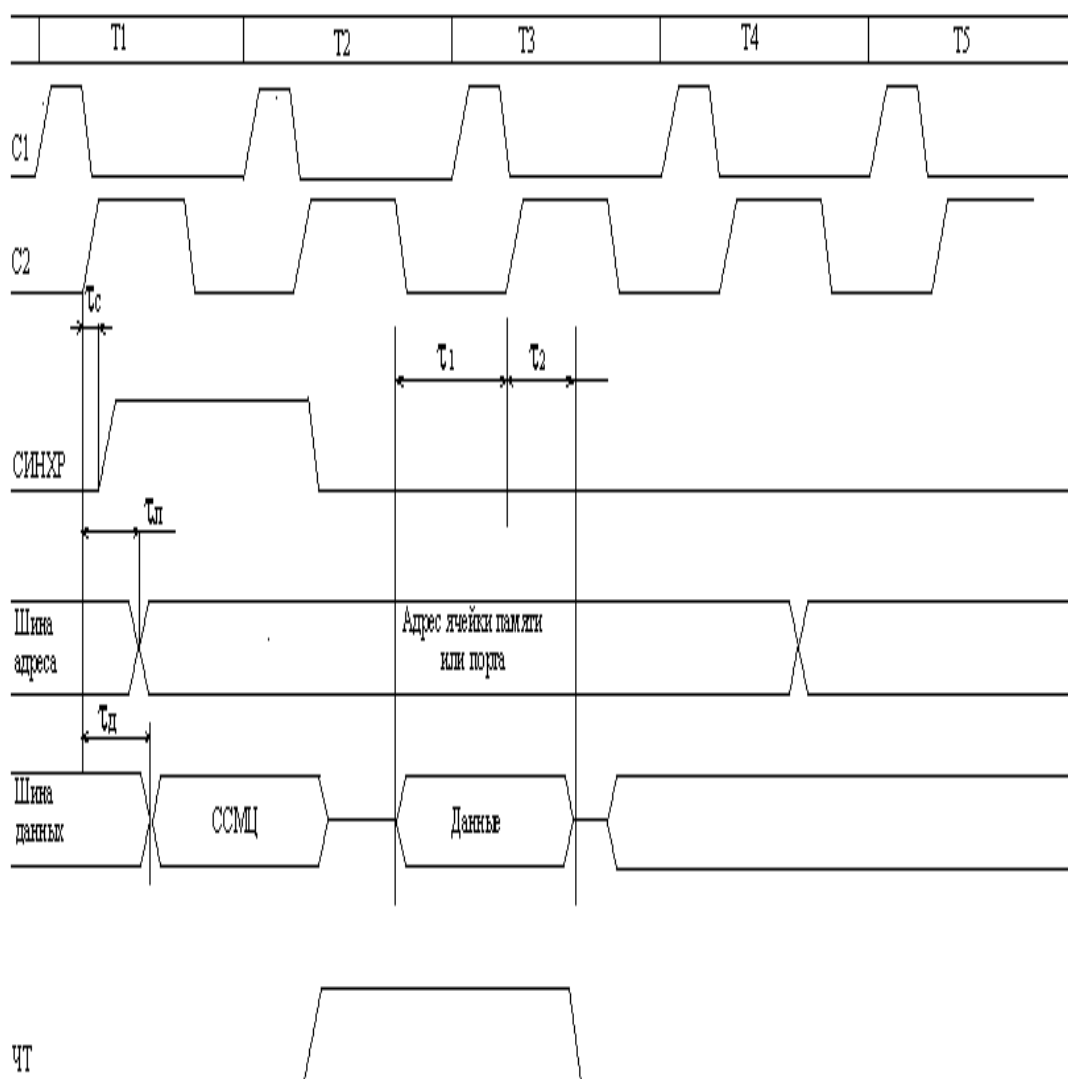


Рис.7. Временные диаграммы машинного цикла чтения данных ($\Gamma T=1$, $3xV=0$, $ЗПР=0$, $ПОСТ=0$)

τ_c относительно переднего фронта сигнала C_2 . Во втором машинном такте с той же задержкой относительно переднего фронта сигнала C_2 сигнал *СИНХР* примет нулевое значение.

Кроме сигнала *СИНХР* в первом машинном такте МП устанавливает информацию на шине данных и шине адреса. Информация на шине адреса устанавливается с некоторой задержкой τ_d (большей, чем τ_c) относительно

переднего фронта сигнала $C2$. Информация на шине данных установится чуть позже, чем на шине адреса (задержка τ_0 относительно переднего фронта сигнала $C2$). Информация на шине данных в это время представляет собой слово состояния машинного цикла.

В машинном такте $T2$ МП очищает шину данных и переводит её в режим ввода, это происходит с задержкой относительно переднего фронта сигнала $C2$. Одновременно с переводом шины данных в режим ввода вырабатывается сигнал CT , который сигнализирует внешним устройством о том, что МП готов принять данные. Сигнал CT должен использоваться внешними устройствами для синхронизации выдачи данных на шину.

В машинном такте $T3$ осуществляется чтение данных. Внешние устройства должны выставить данные на шину не позднее, чем за время τ_1 до прихода переднего фронта сигнала $C2$, и убрать данные с шины не ранее, чем через время τ_2 после прихода переднего фронта сигнала $C2$. Время перед установки данных τ_1 должно быть не меньше 150 нс, а время после удержания τ_2 - не менее 50 нс. Для исключения конфликтов на шине данных внешние устройства ввода должны подключаться к ней только по сигналу CT . Сигнал CT принимает нулевое значение спустя время τ_c после переднего фронта сигнала $C2$. Снятие сигнала CT должно приводить к отключению внешнего устройства от шины данных.

На рис.7 приведена временная диаграмма для случая, когда машинный цикл состоит из пяти состояний (тактов), однако состояние $T5$ или оба состояния $T4$ и $T5$ могут отсутствовать.

Состояние шины данных во время тактов $T4$ и $T5$ остается неопределенным, адрес снимается с шины адреса в такте $T4$, а если такт $T4$ отсутствует, то адрес заменяется на другой в первом машинном такте $T1$ следующего машинного цикла.

Для нормальной работы системы необходимо, чтобы внешние устройства обладали требуемым быстродействием для соблюдения времени перед установкой и после удержания. Если же внешние устройства не обладают таким быстродействием, то необходимо воспользоваться входом GT для перевода МП на требуемое время (пока не будут готовы данные) в состояние ожидания $T_{ож}$.

Временные диаграммы машинного цикла записи приведены на рис. 8. Отличительной особенностью является то, что во втором машинном такте $T2$ вслед за снятием с шины данных слова состояния машинного цикла МП выставляет на этой шине данные, которые подлежат записи во внешнее устройство (в память или устройство вывода). В машинном такте $T3$ с задержкой τ_c относительно переднего фронта сигнал CI МП вырабатывает сигнал $ЗП$ (активное значение - нулевой уровень). Сигнал $ЗП$ является идентификатором того, что адрес и данные, подлежащие записи, стабильны. Внешние устройства должны использовать сигнал $ЗП$ для синхронизации операции чтения данных с шины, именно в этом такте осуществляется запись данных во внешние устройства.

Диаграммы на рис.8 даны для машинного цикла записи, состоящего из трех машинных тактов. В первом такте следующего машинного цикла с задержкой τ_c относительно переднего фронта сигнала CI сигнал $ЗП$ сбрасывается. Следует заметить, что адрес и данные снимаются позже сброса сигнала $ЗП$, поэтому запись данных во внешнее устройство может производиться и по заднему, и по переднему фронтам сигнала $ЗП$.*

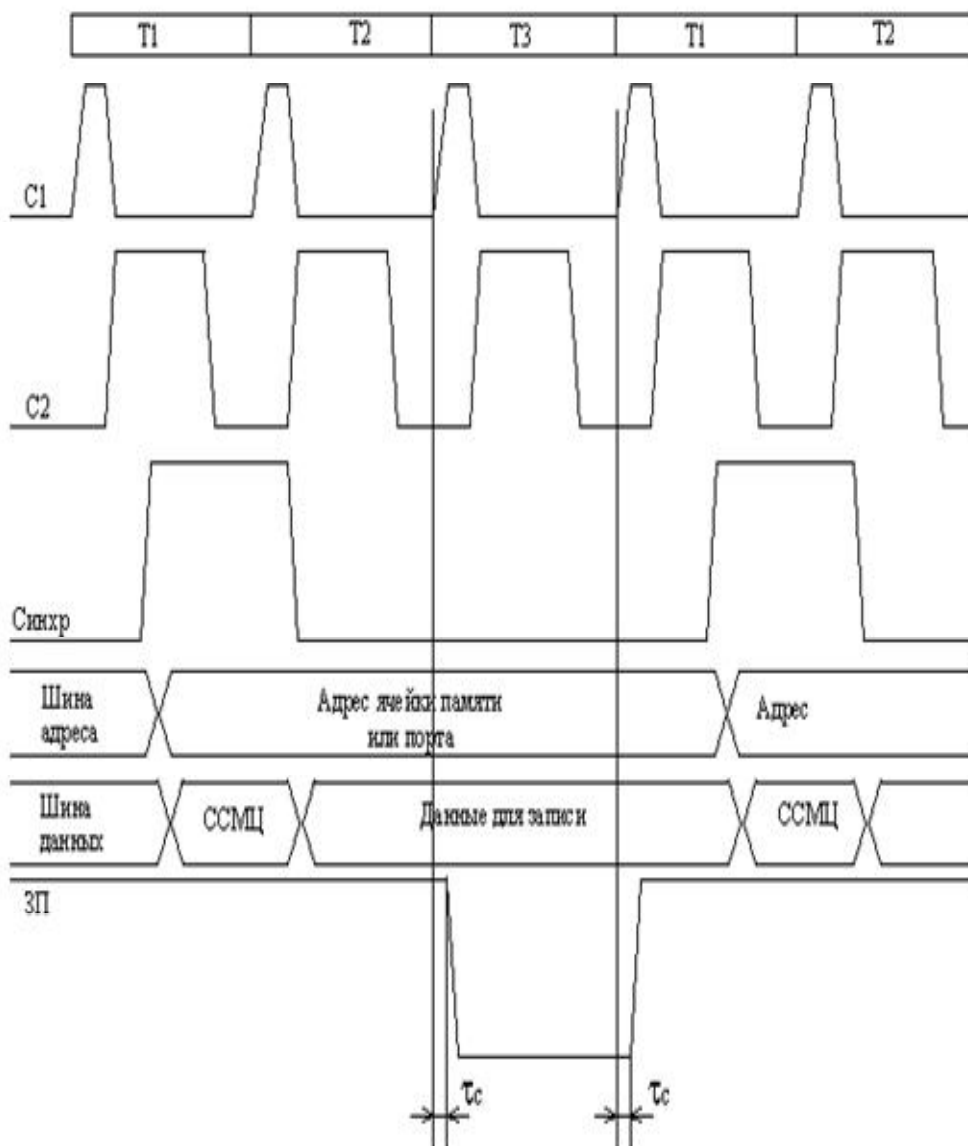


Рис.8. Временные диаграммы машинного цикла записи данных (ГТ=1; 3xВ=0; ЗПР=0, ПОСТ=0).

Для нормальной работы системы необходимо, чтобы внешние устройства обладали требуемым быстродействием для выполнения операции записи за время, равное длительности сигнала ЗП. Если же внешние устройства не обладают таким быстродействием, то необходимо воспользоваться входом ГТ для перевода МП на требуемое время в состояние ожидания ($T_{ож}$).

6. ОСОБЫЕ СОСТОЯНИЯ МИКРОПРОЦЕССОРА

Рассмотрим более подробно три особых состояния МП: ожидания, захвата (прямого доступа к памяти) и останова.

6.1. Состояние ожидания

МП может работать и без использования этого состояния, для чего достаточно, чтобы все внешние по отношению к МП устройства обладали достаточным быстродействием для выполнения операций обмена в темпе, который задается МП. В этом случае на входе GT нужно поддерживать постоянно единичный уровень сигнала. Тогда, как это видно на рис.6, МП никогда не попадет в состояние ожидания ($T_{ож}$); из состояния $T2$ всегда будет переходить в состояние $T3$ (при $ПОСТ=0$). Такой способ передачи данных называется синхронным.

В случае, если какое-либо внешнее устройство не может поддерживать темп, задаваемый микропроцессором, оно может растянуть машинный цикл на время; необходимое для выполнения операции передачи данных. Внешнее устройство может растянуть машинный цикл МП, выставив запрос. Для подачи на МП запроса на продление машинного цикла у МП существует специальный вход ГОТОВНОСТЬ (GT). При появлении нулевого уровня на входе GT МП переходит в режим ожидания (см. рис.6).

Время нахождения МП в состоянии ожидания определяется внешним устройством, выставившим запрос на ожидание. Пока запрос не будет снят, МП будет находиться в состоянии ожидания. Однако время нахождения МП в этом состоянии измеряется всегда целым числом машинных тактов. В этом случае организация передачи данных выглядит следующим образом. МП выставляет адрес внешнего устройства и данные (в случае опе-

рации записи-вывода). В ответ на это внешнее устройство выставляет запрос на продление машинного цикла ($GT=0$). МП переходит в состояние ожидания, а внешнее устройство в это время завершает операцию обмена, после чего внешнее устройство снимает запрос и МП завершает текущий машинный цикл. Такой способ передачи данных называется асинхронным. Используя состояние ожидания, МП может работать совместно с устройствами ввода-вывода, обладающими самыми разными скоростями обмена.

На рис.9 приведена временная диаграмма перехода МП в состояние ожидания. Внешнее устройство, получив свой адрес в первом такте, должно выставить запрос на продление машинного цикла не менее чем за 120 нс. (τ_{zm}) до заднего фронта сигнала $C2$ во втором такте. Если это условие выполнено, то, начиная со следующего такта, МП будет находиться в состоянии ожидания. В этом состоянии все сигналы, участвующие в операции передачи данных, будут сохранять свои значения постоянными. После того как внешнее устройство завершило операцию обмена, оно снимает запрос. Если единичный уровень на входе GT появляется не менее, чем за 120 нс. (τ_{zm}), то данный машинный такт будет последним тактом, в котором МП находился в состоянии ожидания. Следующим тактом будет машинный такт $T3$, в котором МП завершит операцию обмена. После этого восстанавливается нормальное выполнение машинного цикла. О входе в состояние ожидания МП извещает появлением единичного уровня сигнала ОЖИДАНИЕ ($OЖ$). Сигнал ОЖИДАНИЕ принимает единичное значение в первом такте состояния ожидания. После выхода из состояния ожидания сигнал на выходе $OЖ$ сбрасывается (в начале такта $T3$)

6.2. Состояние захвата (прямого доступа к памяти)

МП имеет средства для предоставления своих шин внешним устройствам. Это может потребоваться в случае, если необходимо передать данные между двумя внешними устройствами, минуя МП, например, для организации режима прямого доступа к памяти (*ПДП*). Режимом прямого доступа называется передача данных между устройствами ввода-вывода и памятью без участия МП. Режим *ПДП* может применяться, например, в случае, если МП не может обеспечить надлежащей скорости передачи данных между памятью и устройствами ввода-вывода. Тогда эта передача может осуществляться без его участия. Режим *ПДП* разгружает МП. Во время передачи данных по шинам микропроцессора он может выполнять внутренние операции обработки данных (вся фаза исполнения команды *DAD* может происходить параллельно с режимом *ПДП*).

Процесс перехода МП в состояние захвата поясняется на рис.6. Как только МП оказывается в состоянии *T2* (пусть $GT=1$, $POST=0$), он анализирует сигнал на входе *ЗХВ* (*ЗПДП*). Если *ЗХВ* равен единице, то происходит установка внутреннего триггера захвата ($Tr\ ZПДП=1$). После этого продолжает выполняться нормальный машинный цикл. После выхода из последнего состояния, если $Tr\ ZПДП$ установлен, осуществляется переход в состояние захвата и МП переводит линии адресной шины и шины данных в отключенное состояние.

Как видно из диаграммы на рис.6, время нахождения МП в этом состоянии определяется внешним устройством, выставившим запрос на захват шин. Пока запрос не будет снят, МП будет находиться в состоянии захвата. Однако время пребывания в этом состоянии будет измеряться целым числом машинных тактов. Пунктирная линия со сноской (3) озна-

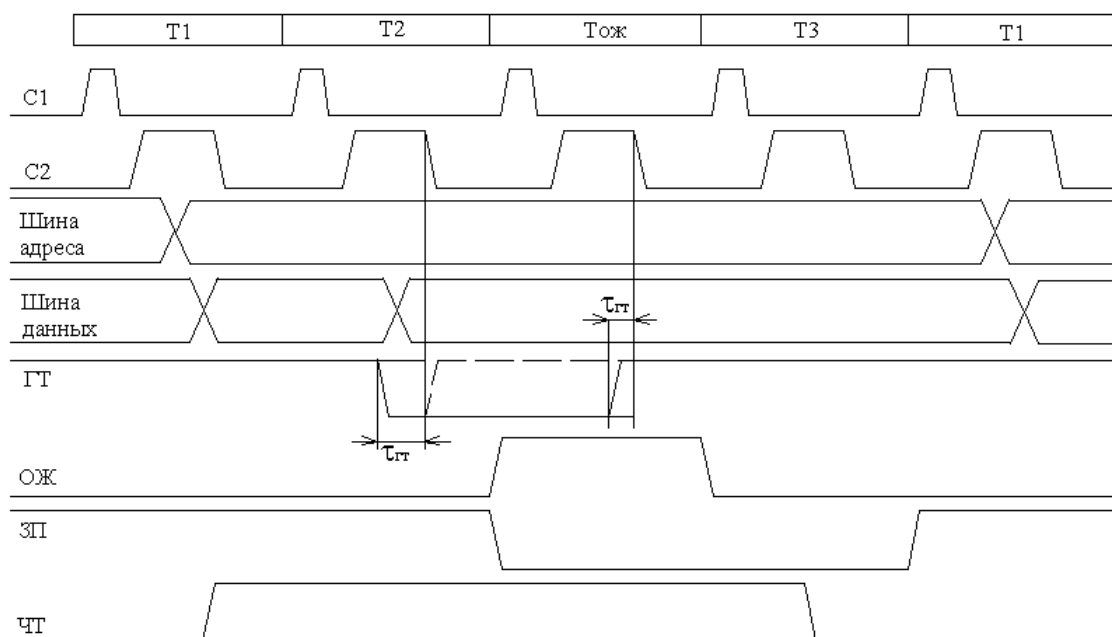


Рис.9. Временные диаграммы режима ожидания ($ZxV=0$, $ZPR=0$, $POST=0$).

чает, что, если машинный цикл состоит из четырех и более машинных тактов, такты $T4$ и $T5$ могут выполняться МП параллельно с режимом ПДП, поскольку в них МП выполняет только внутренние операции.

Таким образом, МП может перейти в состояние захвата его шин внешними устройствами в любом машинном цикле. При этом дальнейшее выполнение цикла команды приостанавливается. Переход в состояние захвата осуществляется всегда после завершения микропроцессором передачи данных в текущем машинном цикле: если выполняется машинный цикл чтение-ввод, то после завершения чтения-ввода в — третьем такте ($T3$); если же выполняется цикл записи-вывода то после завершения записи (вывода) — в такте, следующем за $T3$.

Временные диаграммы режима захвата шин МП приведены на рис.10.

Для того чтобы захват был осуществлен в данном машинном цикле, внешнее устройство должно выставить запрос (единичный уровень на входе ZXB) не позднее, чем за 140 нс. (τ_{zxB}) до переднего фронта сигнала

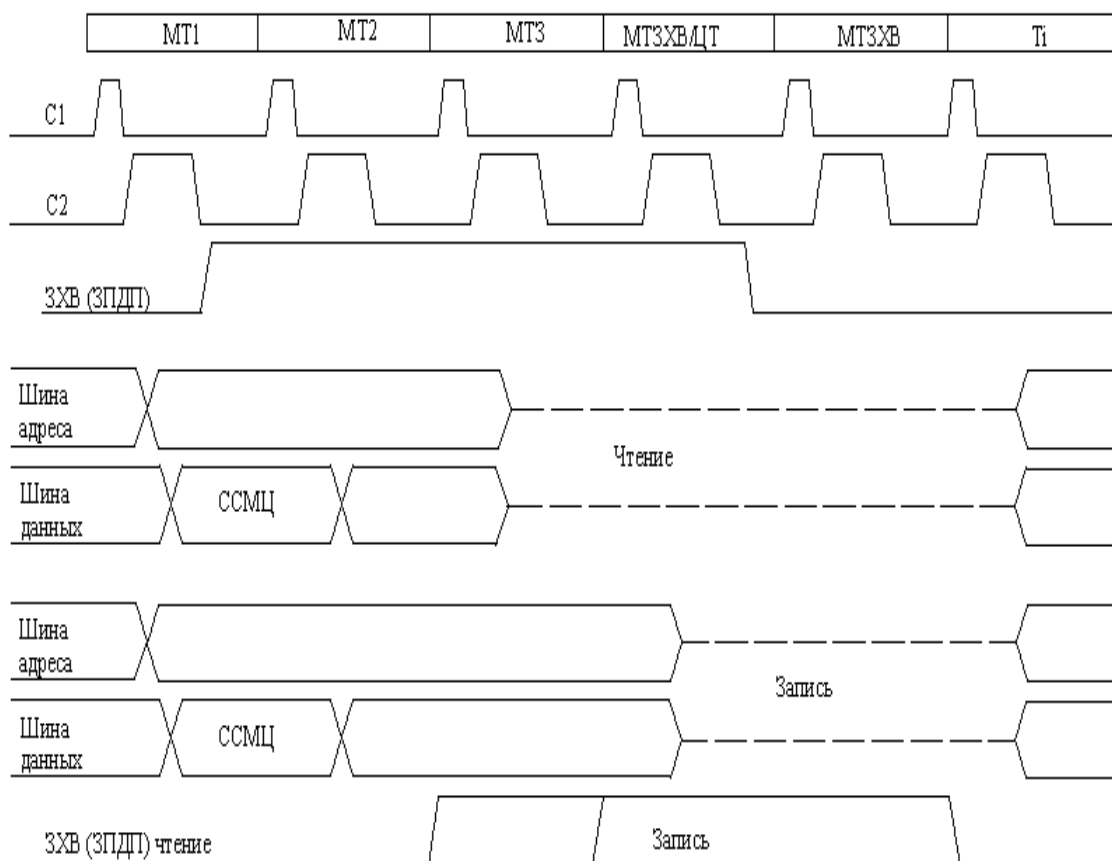


Рис.10. Временные диаграммы режима захвата (ZPR=0, POST=0, GT=1): третье состояние шины.

C2 во втором машинном такте (T_2). Если запрос был установлен, то после выполнения передачи данных в данном машинном цикле МП перейдет в состояние захвата и шины данных, и адреса будут освобождены. Это произойдет в середине такта T_3 , если выполнялась операция чтения-ввода, и в середине такта, следующего за T_3 , если выполнялась операция записывывода. Свой переход в состояние представления шин внешним устройством МП сопровождает выдачей единичного сигнала на выходе $PZXB$ (подтверждение захвата $ППДП$).

Однако существует некоторая задержка между появлением сигнала $PZXB$ и отключением шин.

Необходимо отметить, что выходы МП *ЧТ* и *ЗП* не переходят в третье состояние во время состояния захвата.

Данный машинный такт будет последним тактом состояния захвата, если запрос на захват будет снят не позднее чем за время t_{40} нс. ($\tau_{ЗХВ}$) до переднего фронта сигнала *С2* в этом такте. Если так, то со следующего такта МП возобновляет свою работу.

В состоянии захвата содержимое всех внутренних регистров остается неизменным.

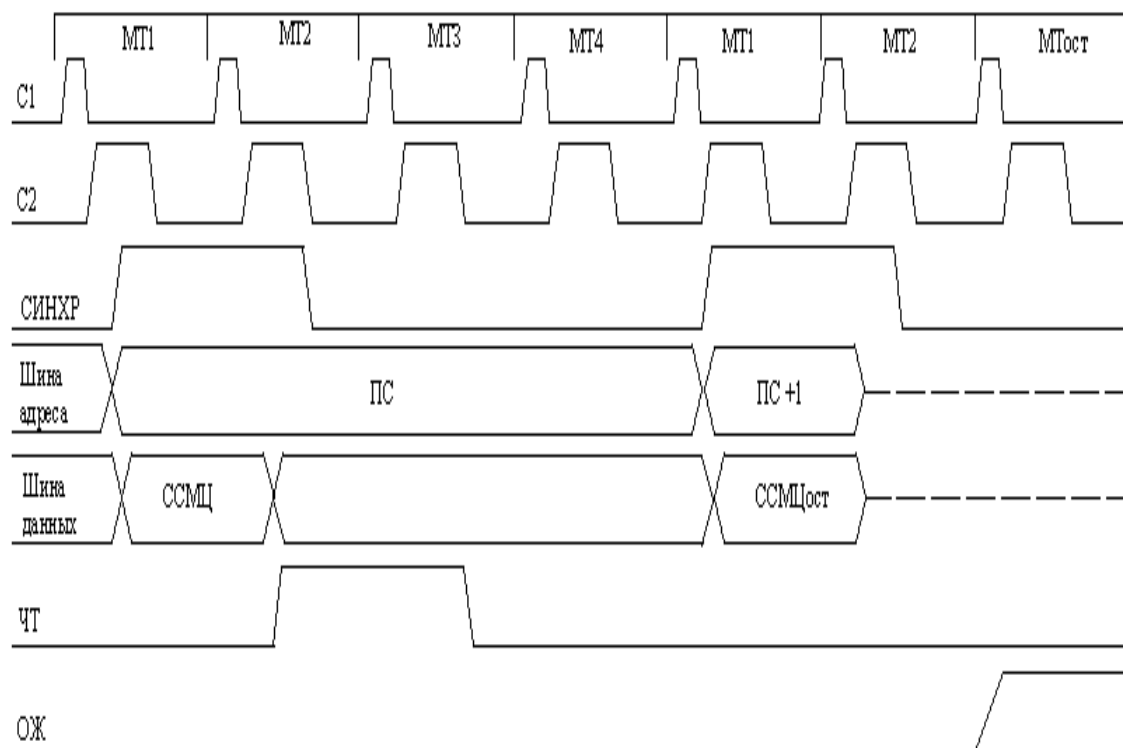
6.3. Состояние останова

В состояние останова $T_{ост}$ МП входит при выполнении команды *HLT*, а точнее — после такта *T2* следующего машинного цикла. В состоянии $T_{ост}$ МП может находиться неопределенно долго. Однако время пребывания МП в этом состоянии будет измеряться целым числом машинных тактов. В состоянии останова МП будет сохранять неизменным содержимое всех регистров.

Выйти из состояния останова МП можно только следующими способами:

1. По сигналу *СБРОС*, при этом обнуляется программный счетчик и МП переходит в состояние *T1*.
2. По запросу на захват шин (*ЗПДП*) МП перейдет из состояния $T_{ост}$ в состояние *TЗХВ*. После снятия сигнала запроса МП вернется в состояние останова по переднему фронту сигнала *С1*.
3. По запросу на прерывание (*ЗПП*), если прерывания разрешены ($TRPP=1$), МП выйдет из состояния останова и войдет в состояние *T1*. Подробно о прерываниях будет рассказано ниже. Временные диаграммы выполнения команды *HLT* приведены на рис.11, откуда видно, что выполнение команды занимает два машинных цикла.

Второй машинный цикл нужен только для выдачи слова состояния машинного цикла (*ОСТАНОВ*).



**Рис.11. Временная диаграмма режима останова
($ZxV=0$, $ZПР=0$, $ГТ=1$).**

Вход в состояние $T_{ост}$, и выходы из него подробно представлены на диаграмме смены состояний (см. рис.6).

Шина адреса и шина данных переключаются в третье состояние в первом машинном такте состояния останова.

7. ПРЕРЫВАНИЯ В МП

МП КР580ИК80 обладает средствами, позволяющими прерывать выполнение основной программы, перейти на обработку запроса прерывания, затем вернуться в точку прерывания основной программы и продолжить её выполнение. Процедура такого рода называется прерыванием.

Из диаграммы смены состояний МП (см. рис.6) видно, что запрос на прерывание воспринимается только после окончания выполнения текущей команды. Таким образом, прерывать выполнение какой-либо команды невозможно. Прерывание программы возможно только после окончания той команды, во время исполнения которой пришел запрос на прерывание.

Запрос от внешних устройств на прерывание должен подаваться на специальный вход МП-ЗПП.

МП имеет два внутренних триггера: разрешения прерывания (*ТППР*) и прерывания (*ТПР*).

Прерывание программы возможно только в том случае, если *ТППР* установлен.

Установка триггера разрешения прерывания производится только по специальной команде *E1*.

Первоначально, после включения питания, *ТППР* сброшен. *ТППР* может быть сброшен также выполнением специальной команды *D1*, а также по сигналу *СБРОС*. Кроме того, *ТППР* сбрасывается при восприятии запроса на прерывание.

Состояние триггера разрешения прерывания постоянно присутствует на специальном выходе МП-РПП. Уровень на этом выходе информирует внешние устройства о том, разрешены или запрещены прерывания: $RPP=1$ — прерывания разрешены; $RPP=0$ — прерывания запрещены.

Триггер прерывания устанавливается в единицу, если запрос на прерывание воспринят и сбрасывается при переходе на программу, обрабатывающую прерывание.

Рассмотрим подробно процесс прерывания. Допустим, что с помощью команды *E1* прерывания разрешены и при выполнении *i*-й команды на вход *ЗПП* поступает сигнал единичного уровня.

На рис.12 представлены временные диаграммы, поясняющие процесс прерывания. Для того чтобы запрос на прерывание был воспринят при выполнении i -й команды, он должен прийти не позже, чем за время $\tau_{3ПР}$ (120 нс.) до заднего фронта сигнала $C2$ в последнем такте последнего машинного цикла. Если это так, то триггер прерывания устанавливается в единицу и МП переходит в состояние $T1$, начиная выполнять новый машинный цикл. Этот машинный цикл называется ПЕРЫВАНИЕ и почти ничем не отличается от машинного цикла ВЫБОРКА КОМАНДЫ. В слове состояния этого цикла вместо бита $ЧТП$ единичное значение имеет бит $ППР$ (см. рис.5).

Следует сразу отметить, что в этом машинном цикле не происходит инкремент программного счетчика, что важно для организации возврата в точку прерывания программы. При выполнении этого машинного цикла МП не делает каких-либо особых действий.

В первом машинном такте на шину адреса выставляется адрес $(i+1)$ -й команды, на шину данных — слово состояния машинного цикла ПЕРЫВАНИЕ.

Во втором такте $ССМЦ$ снимается с шины данных и устанавливается сигнал $ЧТ$, который говорит о том, что шина данных МП перешла в режим ввода первого байта команды (так же как в машинном цикле ВЫБОРКА КОМАНДЫ). Однако из памяти этот байт не может быть выбран (в противном случае выбралась бы очередная команда, и прерывания бы не произошло), так как бит $ЧТП$ в слове состояния машинного цикла равен нулю. Требуемый байт должно "подсунуть" на шину данных внешнее устройство, запросившее прерывание.

В третьем машинном такте внешнее устройство должно передать на шину данных команду перехода на обрабатываемую программу. Так как обычные команды вывода подпрограмм имеют длину три байта, они не

подходят для этой цели. Команда вызова обрабатываемой программы должна иметь длину один байт.

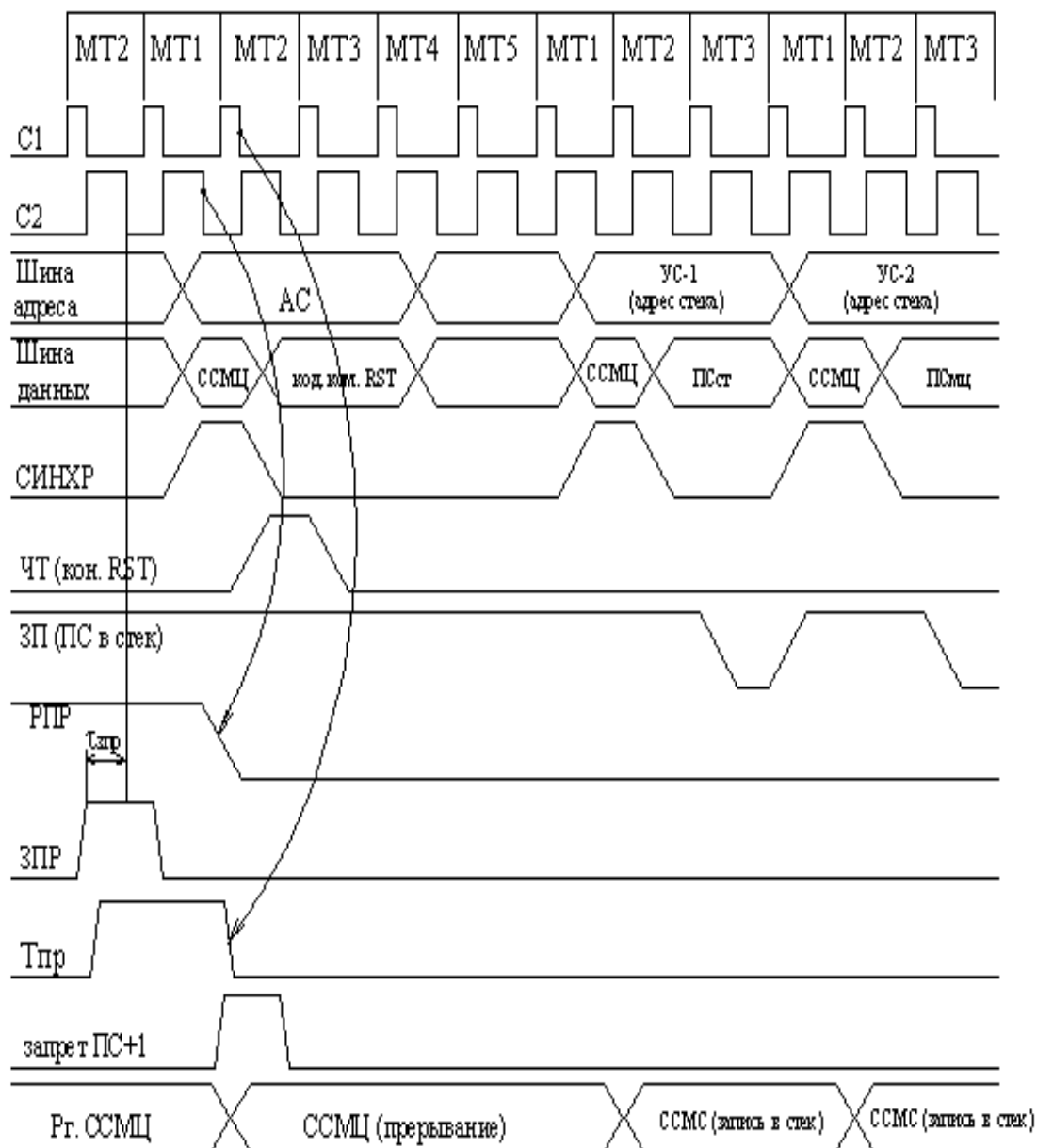


Рис.12. Временная диаграмма режима прерывания:

1 — если TПР=1, то RПР=0; 2 — если RПР=0, то TПР=0; 3 — см. разд. 6.2

В системе команд МП КР580ИК80 имеется такая команда, которая называется "Рестарт" *RST*. В теле команды (рис.13) имеется трех битовое поле,

от содержимого которого зависит адрес, на который осуществляется последующий переход. В зависимости от содержимого этого поля может быть осуществлен переход по восьми адресам: 0, 8, 16, 24, 32, 40, 48 и 56 (десятичные числа).

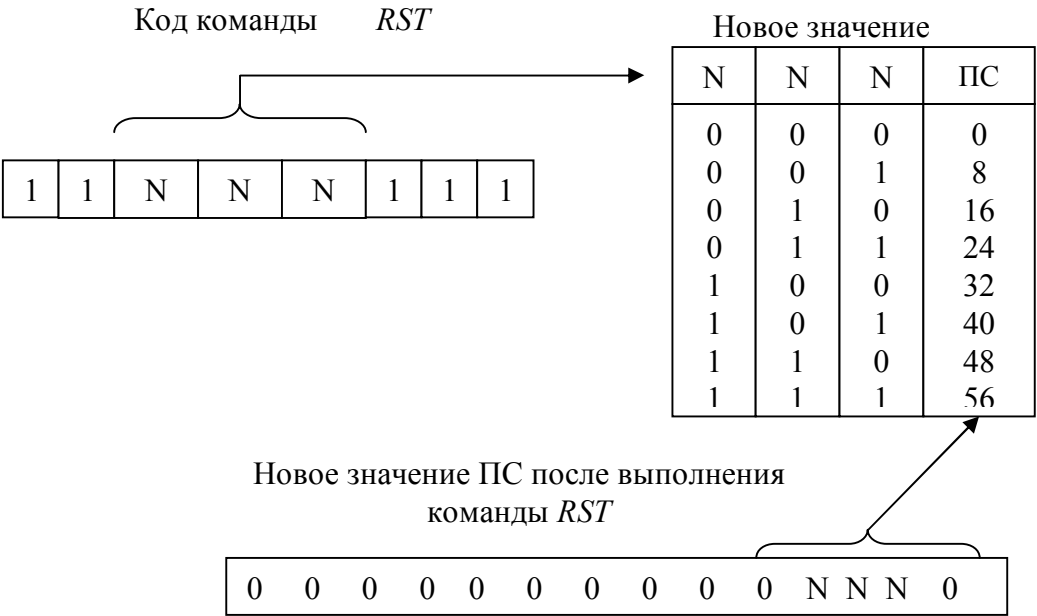


Рис.13. Команда RST.

Таким образом, передав в третьем машинном такте код команды *RST* (с нужным трех битовым полем), внешнее устройство обеспечивает прерывание основной программы и ее переход на соответствующую обрабатывающую программу. В тактах *T4* и *T5* формируется новое содержимое программного счетчика в соответствии с содержимым трех битового поля в команде *RST*. На этом машинный цикл ПЕРЫВАНИЕ заканчивается.

Далее следуют два машинных цикла ЗАПИСЬ В СТЕК, в течение которых старое содержимое программного счетчика записывается в стек для обеспечения возврата в основную программу. Последней командой в обрабатывающей программе должна быть команда *RST*, которая и обеспечит возврат (восстановит старое содержимое программного счетчика).

Остается добавить, что в первом машинном такте машинного цикла ПЕРЕРЫВАНИЕ по сигналу *C2* сбрасывается триггер разрешения прерывания. Это происходит не позднее 200 нс. относительно переднего фронта импульса *C2*. Во втором машинном такте сбрасывается внутренний триггер прерывания. Таким образом, дальнейшие запросы прерывания не будут восприниматься даже по возвращении в основную программу. Поэтому, если мы хотим, чтобы обрабатываемая программа могла быть прервана, необходимо выполнить команду *E1*.

8. ИНИЦИАЛИЗАЦИЯ РАБОТЫ МП

В первый момент времени после включения питания содержимое программного счетчика так же, как и всех остальных регистров, неопределенно. В программном счетчике может находиться любое число, причем после каждого включения оно может быть различным. Поэтому сразу же после включения питания на вход *СБРОС* МП должен быть подан сигнал единичного уровня длительностью не менее трех машинных тактов.

По этому сигналу произойдет обнуление программного счетчика и МП начнет выполнять программу, начиная с нулевой ячейки памяти. Следует помнить, что сигнал *СБРОС* воздействует только на программный счетчик. Поэтому состояние всех остальных регистров МП остается неизменным и должно быть определено программно.

Если МП должен начинать работать сразу же после включения питания, то рабочая программа должна размещаться с нулевой ячейки памяти.

Если же МП должен начинать работать только после прихода некоторого сигнала или по команде оператора, то в нулевую ячейку записываются команды *E1*, *HLLT*. МП, выполнив команду *HLLT*, переходит в состояние оста-

нова, из которого он может быть выведен прерыванием от внешнего сигнала или от кнопки на панели управления.

9. СТАНДАРТНАЯ АРХИТЕКТУРА МИКРОЭВМ НА ОСНОВЕ МП КР580ИК80

Стандартная архитектура микроЭВМ на основе МП КР580ИК80 представлена на рис.14 и состоит из следующих частей: модуль центрального процессора (МЦП) или ядро системы; память; устройство ввода-вывода; интерфейс.

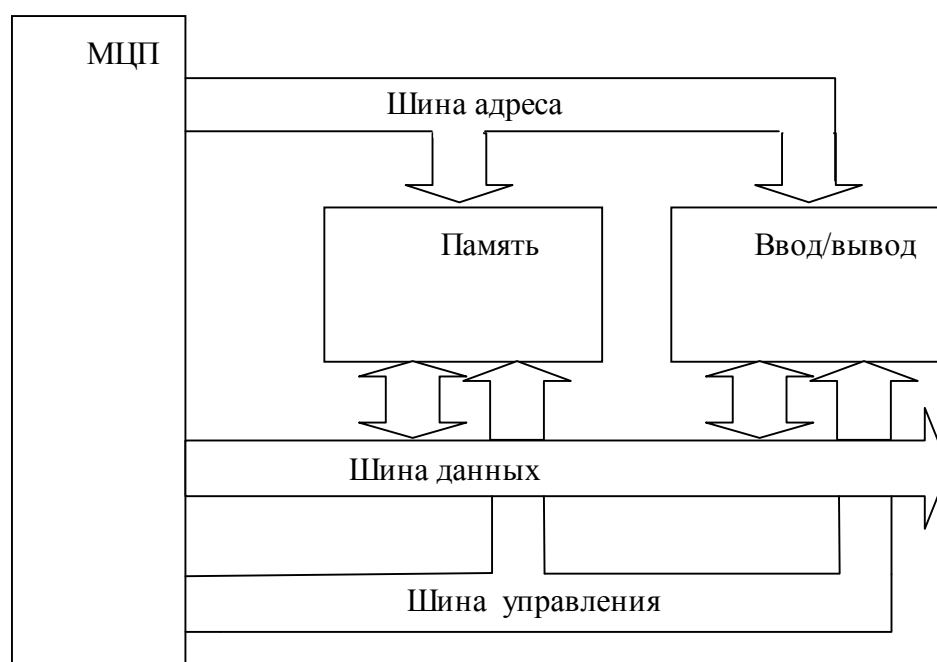


Рис.14. Архитектура микропроцессорной системы.

Модуль центрального процессора — главный элемент любой МП системы, осуществляющий управление всеми остальными элементами системы. Он включает в себя МП КР580ИК80, генератор синхросигналов, системный контроллер и буферы шины данных и адреса. Схема, стандартного МЦП приведена на рис.15.

Микропроцессорный набор КР580 содержит специальные микросхемы, позволяющие построить *МЦП*: КР580ГФ24 — генератор синхросигналов; КР580ВК28 — двунаправленный буфер шины данных и системный контроллер. Буфер шины адреса может быть построен с помощью шинных формирователей К589АП16 или многорежимных буферных регистров К589ИР12, которые обладают тремя состояниями. Буфер адресной шины служит только для повышения коэффициента разветвления адресных линий.

Генератор может быть построен с использованием микросхемы КР580ГФ24. Эта микросхема вырабатывает необходимые серии синхросигналов *C1* и *C2* с требуемой амплитудой. Напомним, что сигналы *C1* и *C2* должны иметь амплитуду не ниже *9В*, это вызывает некоторые трудности при реализации генератора на других микросхемах. Кроме того, генератор КР580ГФ24 вырабатывает и ряд других сигналов: сигнал *СБРОС*; сигнал *ГОТОВНОСТЬ (ГТ)*; стробирующий сигнал для загрузки слова состояния машинного числа в регистр системного контроллера (*СТРОБ*) и др.

Двунаправленный буфер шины данных служит для повышения коэффициента разветвления линий шины, а также для поднятия уровня логической единицы при передаче сигналов от внешних схем в шину данных микропроцессора. Напомним, что минимальное значение уровня логической единицы для входов МП составляет 3,3 В.

Системный контроллер служит для выработки управляющих сигналов. В шине управления присутствуют сигналы трех типов: для управления памятью, вводом-выводом и для считывания команды *RST*. Обычно используются следующие пять сигналов:

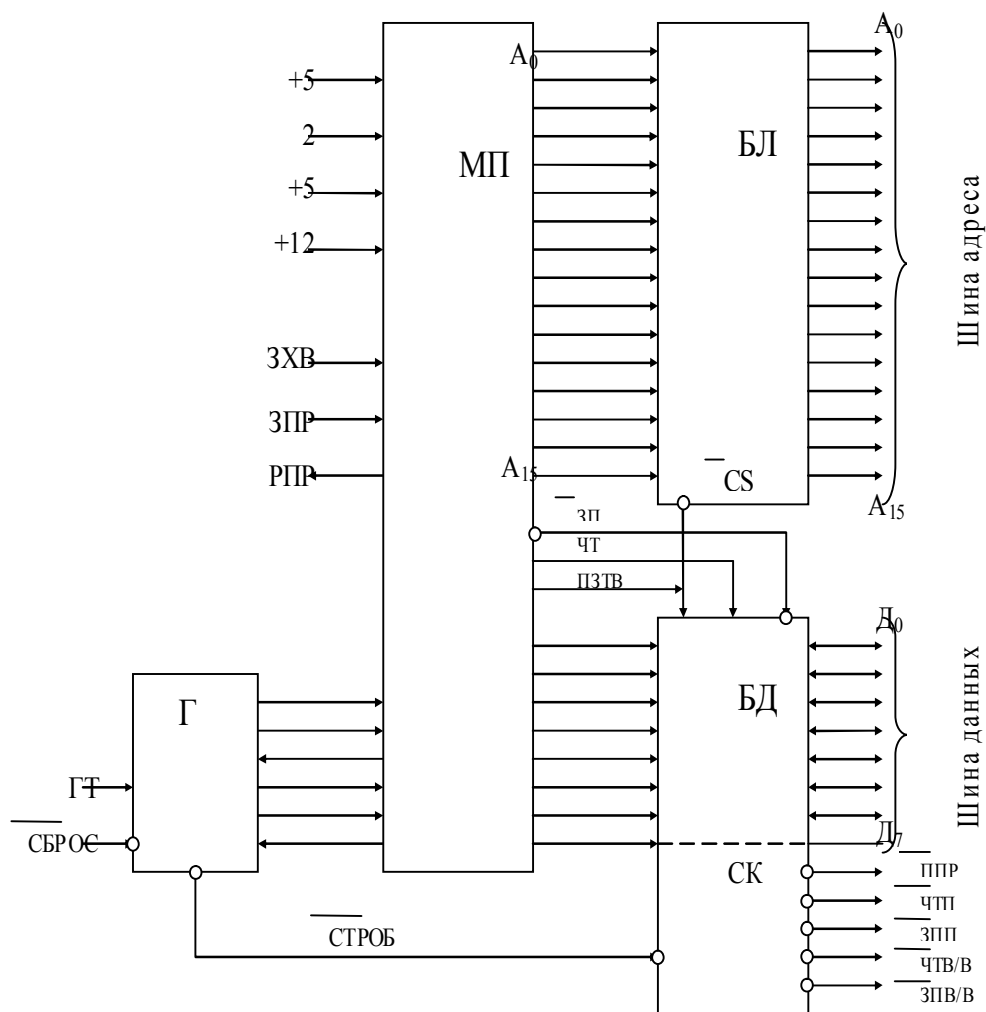


Рис.15. Стандартный модуль центрального процессора.

ППР — подтверждение прерывания, стробирует выдачу на шину данных команды перехода;

ЧТП — чтение из памяти, стробирует выдачу информации из памяти на шину данных;

ЗПП — запись в память, сообщает памяти, что данные для записи присутствуют на шине данных;

ЧТВ-В — чтение из устройства ввода-вывода, стробирует выдачу устройствами ввода-вывода данных на шину данных;

ЗПВ-В — запись в устройства ввода-вывода, сообщает устройствам ввода-вывода, что данные для записи присутствуют на шине данных.

Для реализации буфера данных и системного контроллера можно использовать одну микросхему КР580ВК28. Пять сигналов управления вырабатываются системным контроллером с использованием разрядов слова состояния машинного цикла и сигналов МП *ЧТ* и *ЗП*. Системный контроллер имеет регистра для фиксации слова состояния машинного цикла. Фиксация происходит по специальному сигналу *СТРОБ*, вырабатываемому генератором. Схема получения системных сигналов управления может быть такой, например, как на рис. 16, где приведены временные диаграммы этих сигналов.

Следует отметить, что это лишь один из вариантов организации шины управления. В более простых системах могут отсутствовать сигнал *ППР* или сигналы *ЗПВ-В*, *ЧТВ-В*.

В последнем случае обращение к устройствам ввода-вывода происходит как к ячейкам памяти, при этом все адресное пространство разделяется на две области: памяти и устройства ввода-вывода.

В более сложных системах можно ввести некоторые дополнительные сигналы, например сигналы записи (чтения) в (из) стек.

В этом случае стек будет располагаться в отдельной памяти и общее адресное пространство расширится. В простейшей системе системный контроллер может вообще отсутствовать, а в шине управления тогда будут присутствовать только два сигнала — *ЧТ* и *ЗП*, общие для памяти и устройства ввода-вывода. Прерывания в такой системе невозможны.

К памяти системы на основе МП КР580ИК80 не предъявляются какие-либо особые требования. Память, как правило, состоит из двух частей: постоянной и оперативной (в некоторых случаях оперативная память может отсутствовать). Постоянная память используется для хранения программ, а

оперативная, как правило, для стека и хранения промежуточных результатов.

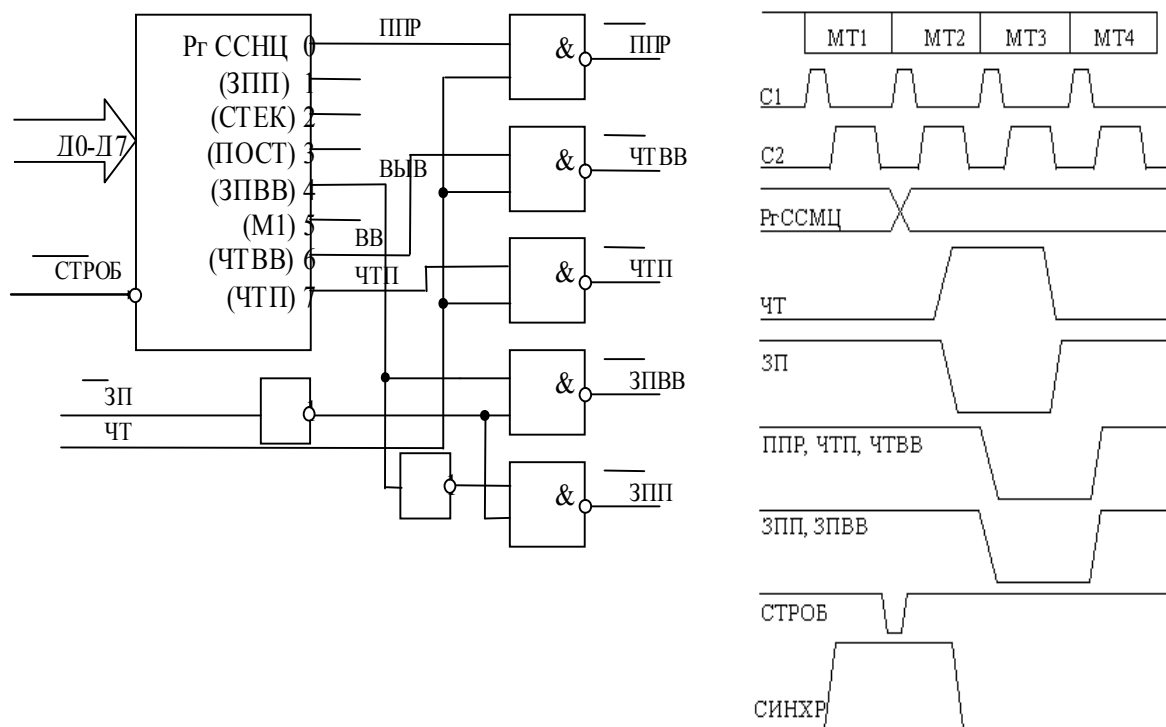


Рис.16. Схема получения системных управляющих сигналов и временные диаграммы.

В состав системы входят также устройства ввода-вывода. Простейшее устройство ввода-вывода — 8-разрядный регистра ввода-вывода, или порт. С помощью команд IN, OUT можно считывать из порта или записывать в порт информацию из аккумулятора. В качестве порта ввода вывода могут выступать многорежимный буферный регистр (МБР) К589ИР12 или специальные интерфейсные схемы из набора КР580: КР580 ИК55 — программируемый периферийный адаптер (ППА); КР580 ИК51 - программируемый связной адаптер (ПСА); КР580ИК53 - программируемый интервальный таймер (ПИТ).

Система команд МП.

Системой команд называют перечень инструкций команд, который может выполнить данный микропроцессор. Микропроцессоры разных типов имеют свои несколько отличающиеся друг от друга системы команд.

Рассмотрим систему команд однокристалльного микропроцессора КР 580 ВМ 80 А

Общие правила записи команд. Большинство команд состоит из двух частей. В первой части указывается на те действия которые должен выполнить микропроцессор. Эта часть команды называется кодом операции.

Вторая часть команды указывает на то где находятся данные или каков их адрес. Эта часть команды получила название операнд может быть записан тремя способами:

1. В виде числа (в двоичном, десятиричном или шестнадцатиричном формате)
2. В виде адреса ячейки памяти или порта, в которых находится число.
3. В виде места, где хранится число (например регистрВ)

Из трех перечисленных способов записи чаще всего пользуются адресом, что делает программу более универсальной и пригодной для многих аналогичных случаев.

Встречаются команды состоящие только из первой части, то есть не имеющие операнда. К таким командам относятся некоторые служебные команды, например EI, HLT.

Чтобы упростить написание и облегчить запоминание команды, ее первую часть операции записывают в виде сокращенных английских слов, указывающие выполненные действия. При этом употребляют сокращение до двух, трех или четырех букв.

Например команда ввода-вывода IN
 команда пересылки MOV
 команда вызова подпрограммы CALL

Для определения места где находятся данные, применяют условные обозначения. Расположение данных в РОН МП обозначают латинской буквой R под которой понимаются регистры (A, B, C, D, E, H, L). Если данные находятся в ячейке памяти, то пишут букву M.

Когда операнд записывают в виде адреса, то эту запись выполняют при помощи символа ADR или какого-либо условно символического слова, состоящего из четырех-пяти букв (например MULTI).

Для МП КР580ВМ80 употребляются команды в один, два и три байта.

В однобайтовых командах старшие разряды предназначаются для размещения кода операции, а младшие для указания кода регистров:

1 часть		2 часть					
1	0	1	0	1	0	1	0
Код операции		Код 1 операнда			Код 2 операнда		

В двухбайтовых командах первый байт полностью отводится под код операции, а второй – служит для записи данных (операнда) в виде числа, кода регистров или адреса

Команда	
Операция	Операнд
1 байт	2 байт

8 бит второго байта позволяют записать наибольшее число 255₁₀. При необходимости ввода чисел, превышающих 255 используют трех байтовую команду, у которой первый байт – код операции, а второй и третий байты предназначены для размещения шестнадцатиразрядных двоичных чисел или шестнадцатиразрядного адреса ячейки памяти.

Код операции	Операнд	Операнд
1 байт	2 байт	3 байт

При этом имеется возможность задания чисел до 65535.

При написании команд очень важно правильно выбрать способ адресации данных который требовал бы наименьший объем памяти для размещения данных и наименьшее время исполнения команды. В МП КР 580 ВМ 80 существуют следующие способы адресации:

1. прямой;
2. регистровый;
3. косвенный;
4. непосредственный;
5. неявный.

1. При прямой адресации адрес места, где хранятся данные помещают сразу же за кодом операции. Этим адресом может быть номер порта устройства ввода-вывода. Прямая адресация наиболее простая, но в тоже время и наиболее неэкономичная из-за достаточно длинных командных слов, приводящих к увеличению объема памяти.

2. При регистровой адресации вслед за кодом операции указывают код регистра, где хранятся данные. Этот код занимает три разряда для каждого регистра, поэтому команды получаются однобайтовые и выполняются довольно быстро.

3. При косвенной адресации указывают код регистров, в которых находится адрес ячейки памяти, содержащий данные. Косвенный способ адресации позволяет компактно адресовать пространство памяти при помощи коротких однобайтовых команд.

4. При непосредственной адресации данные в виде восьмиразрядного двоичного числа непосредственно размещают во втором байте команды. Если число имеет длину более восьми разрядов, то его записывают во втором и в третьем байтах. При этом младшие разряды числа заносят во второй байт, а старшие – в третий. Команды с непосредственной адресацией обозначают буквой I, написанной в конце мнемонического слова операции, а само слово уменьшают на одну букву. Например, вместо MOV пишут MVI, вместо ADD – ADI, и вместо ORA – ORI.

5. При неявной (подразумеваемой) адресации один из операндов должен находиться во внутреннем регистре микропроцессора, чаще всего в регистре A, называемом аккумулятором. Поэтому нет необходимости дополнительно сообщать его адрес. Команды с неявной адресацией получаются более короткими, что сокращает их написание и выполнение. Неявный способ адресации часто встречается в командах МП-580, который устроен так, что все логические и арифметические операции выполняются только через аккумулятор, когда один из операндов должен находиться в аккумуляторе еще до выполнения команды.

Существует также смешанный способ адресации. Например, команда CALL использует косвенную адресацию и прямую.

Система команд МП-580 насчитывает более 240 команд. Многие из них повторяют свое мнемоническое название, отличаясь друг от друга только операндом. Например команда MOV повторяется более 60 раз, указывая при помощи новых операндов разные места пересылки данных. Поэтому основных (ключевых) команд с непохожими мнемоническими наименованиями имеется около 70. из них более половины команд встречается при программировании очень редко. Таким образом для начала достаточно запомнить 20-30 мнемонических названий операций, чтобы приступить к написанию простейших команд на машинном языке или языке ассемблера.

Все команды МП-580 можно условно разделить на группы:

1. команды пересылок;
2. команды арифметических операций;
3. команды логических операций;
4. команды перехода;
5. специальные команды.

команды пересылок.

Команды пересылки чаще других встречаются при составлении программ и поэтому на них следует обратить особое внимание. Каждая пересылка данных всегда связана с местом, куда они пересылаются, а именно с приемником данных и тем местом, откуда эти данные поступают – ис-

точником данных. При написании операнда условились сначала указывать приемник, а затем источник. Наименование приемника и источника разделяют запятой.

К командам пересылок относятся следующие команды:

1.MOV

Данная команда пересылок позволяет обменивать данные между РОН и памятью. Синтаксис этой команды может выглядеть, как:

1.1 MOV R1, R2.

1.2 MOV R, M.

1.3 MOV M, R.

где R1, R2, R – регистры общего назначения.

M – ячейка памяти.

При обмене данными между внутренними регистрами A,B,C,D,E,H,L команда MOV повторяется 49 раз, указывая в своем операнде различные сочетания регистров (A и B, A и C, A и E, A и L, B и A и т. д.)

Например команда MOV A,B позволяет переслать 8-битные данные из регистра B в аккумулятор A, команда MOV H,E – из регистра E в регистр H, т.е. пересылка осуществляется из R2 в R1. При этом используется регистровый способ адресации и команда является однобайтовой в которой:

0	1	Код R1	Код R2
Код операции		Коды регистров	

Команды MOV R, M и MOV M, R позволяет обменивать данные между памятью и внутренними регистрами МП. Т. к. адрес ячейки памяти является шестнадцатиразрядным то перед выполнением этой команды в регистровую пару HL необходимо поместить адрес ячейки памяти в (из) которую пересылаются данные. В этом случае по данной команде МП пересылает данные из(в) ячейки памяти, адрес которой находится в регистровой паре HL в(из) данный регистр общего назначения.

Следующая команда пересылок:

2.MVI R,K

где R – регистр общего положения

K – произвольное 8-битное число

Команда предназначена для размещения в каком либо регистре РОН МП произвольного числа K. последняя буква мнемоники команды I указывает на то что в команде участвуют непосредственные данные. Команда MVI R,K является двухбайтовой. В первом байте указывается код команды

и код регистра приемника, во втором байте указывается непосредственно число К.

Например: по команде

MVI B, 12H

Число 12 записанное в шестнадцатеричном формате записывается в регистр В.

3. MVI M, K

Данная команда позволяет записать с ячейку памяти восьмиразрядное число К. При этом адрес ячейки памяти находится в регистровой паре HL. Например: необходимо в ячейке памяти М с адресом OFFOH записать число 17H.

Для этого необходимо выполнить следующие команды:

MVI H, OFH

MVI L, FOH

MVI M, 17H

Следующая команда

4. LXI P, K

Команда LXI P, K в отличии от команды MVI R, K позволяет загружать шестнадцатиразрядное, а не восьмиразрядное число К, причем запись осуществляется в регистровую пару Р, а не в регистр R.

Операнд Р обозначает одну из регистровых пар МП 580 BC, DE или HL, причем последние в этом случае обозначаются как:

BC – В

DE – D

HL – H

Команда LXI P, K – трехбайтовая.

Например: по команде LXI H, FFOOH, в регистровую пару HL загружается число FFOOH.

5. LDA ADR

С помощью команды LDA ADR можно загрузить аккумулятор А содержимым ячейки памяти с адресом ADR.

Команда LDA ADR – трехбайтовая.

Например: по команде LDA 2FF2H в аккумулятор А записывается число, находящееся в памяти по адресу 2FF2H.

6. STA ADR

Команда STA ADR, в отличие от команды LDA ADR позволяет переслать содержимое аккумулятора в ячейку памяти с адресом ADR.

7. LDAX P	8. STAX P
-----------	-----------

Команды LDAX и STAX выполняют те же операции что и команды LDA и STA, т.е. пересылают содержимое ячейки памяти M, в аккумулятор и обратно. Отличие этих команд заключается в том что при выполнении команд LDA ADR и STA ADR адрес ячейки памяти находится во 2 и 3 байтах самой команды, а при выполнении команд LDAX и STAX адрес ячейки памяти находится в регистровой паре P. Причем регистровой парой может быть только пара BC и DE. В регистровой паре HL размещать такой адрес нельзя. Команды LDAX и STAX – однобайтовые и эти команды в основном используются в том случае, когда по условию выполнения программы адрес ячейки памяти M по тем или иным причинам уже находится в регистровой паре BC или DE.

10. IN

Команда IN N служит для ввода данных в аккумулятор A из устройства ввода имеющего номер N. В этой команде приемником данных вида является аккумулятор, поэтому он и не указывается в операнде.

11. OUT

Команда OUT N служит для вывода данных из аккумулятора в порт вывода с номером N. Команды IN и OUT являются двухбайтовыми. Первый байт отводится для кода операции, а второй для записи адреса порта вывода-ввода.

12. PUSH	13. POP
----------	---------

По этим командам записывается (по команде PUSH) и считывается (по команде POP) в(из) стек содержимое либо регистровой пары B, D или H, либо слово состояния и содержимое аккумулятора. Начальный адрес области памяти, отведенной для стека находится в указателе стека SP микропроцессора.

Например: необходимо поместить в указатель стека содержание регистровой пары BC.

Для этого помещаем в указатель стека SP начальный адрес области памяти для стека LXI SP, ADR. Затем PUSH B.

14. XCHG

Осуществляется обмен содержимым регистровых пар DE и HL.

15. XTHL

Происходит обмен содержимого двух первых ячеек стека с регистровой парой HL.

16. SPHL

Пересылается содержимое регистровой пары в указатель стека SP.

17. LHLD

По команде LHLD ADR в регистр L передается содержимое первой ячейки памяти, адрес которой записан во второй и третий байты команды. В регистр H передается содержимое второй ячейки памяти, адрес которой на единицу больше адреса первой ячейки.

18. SHLD

Команда обратная команде LHLD ADR, когда загружаются две соседние ячейки памяти содержимым регистров H и L.

Команды арифметических операций.

Команды арифметических операций выполняются только через аккумулятор. При этом один из операндов должен находиться в аккумуляторе еще до выполнения команды. Таким образом складывая два числа, необходимо одно из них поместить в аккумулятор, а второе можно загрузить в регистр R или в ячейку памяти M. Только после этого можно выполнять операции арифметики.

После выполнения команды арифметической операции результат вычисления автоматически помещается в аккумулятор, а число, которое там было стирается.

МП-K580 выполняет команды сложения и вычитания, а операции умножения и деления осуществляются по особым подпрограммам с помощью набора команд сложения, сдвига и других преобразований в соответствии с

теми правилами которые были рассмотрены нами в разделе арифметики двойного счисления.

К группе команд арифметических операций относятся также команды, уменьшающие или увеличивающие на 1 содержимое регистров PОН или ячеек памяти.

19. ADD R
20. ADD M

Команда ADD R служит для сложения числа, содержащегося в аккумуляторе с числом, размещенным в регистре R. Сумма, которая образуется после выполнения этой команды, помещается в аккумулятор, а число, находившееся там стирается.

Команда ADD R – однобайтовая.

Пример:

Сложить два десятичных числа 13 и 7. и результат поместить в ячейке памяти F1F0H

```
MVI A, 13D
MVI B, 7D
ADD B
STA F1FO
```

Команда ADDM является разновидностью команды ADD R, отличающейся тем что число содержащееся в аккумуляторе складывается с числом находящимся в ячейке памяти M. адрес этой ячейки памяти находится в регистровой паре HL.

Команды ADD R и ADD M при выполнении операции арифметического сложения влияют на все флаги регистра признаков.

Так при сложении чисел может возникать:

1. единица переноса которая записывается в признак C.
2. при сложении двух двоично десятичных чисел единица в дополнительном признаке переноса AC.
3. признак знака S.
4. признак нуля Z.
5. признак четности P.

21. ADC

Команда ADC R и ADC M аналогична командам ADD R и ADD M с отличием в том что сложение двух чисел осуществляется с учетом признака переноса. Единица переноса прибавляется к результату вычислений в младшем разряде.

22. ADI K
23. ACI K

Эти команды позволяют сложить число K с числом находящемся в аккумуляторе. Для этого во второй байт команды нужно записать двоичное выражение числа K , после выполнения команды сумма вычислений остается в аккумуляторе.

Команды различаются тем что в ACI K число K складывается с учетом признака переноса.

24. SUB	26. SUI
25. SBB	27. SBI

Данные команды выполняют операции вычитания. По своей структуре и особенностям действий они аналогичны операциям сложения.

SUB – ADD
 SBB – ADC
 SUI – ADI
 SBI – ACI

Все что было сказано о командах сложения, действительно и для команд вычитания. Таким образом по команде SUB R выполняется вычитание из содержимого аккумулятора числа расположенного в регистре R, по команде SUB M выполняется вычитание числа расположенного в ячейке памяти M, а по команде SUI K – вычитание числа K , записанного во втором байте команды.

Единственное отличие команд вычитания от команд сложения стоит в том что в аккумуляторе должно находиться уменьшаемое число. В это же время в регистре R, в ячейке памяти M или во втором байте команды должно размещаться вычитаемое.

28. INR R	30. DCR R
29. INR M	31. DCR M

Команда INR R позволяет увеличить на единицу содержимое регистра R, а команда INR M – увеличить на единицу содержимое ячейки памяти M. аналогично команда DCR R позволяет уменьшить на единицу содержимое регистра R, а команда DCR M – уменьшить содержимое ячейки памяти M.

Команды INR R и DCR R однобайтовые. Перед выполнением команд INR M и DCR M, в которых участвуют ячейка памяти M, адрес ячейки памяти M нужно разместить в регистровой паре HL. Иногда указанные операции называют инкрементированием, а операцию уменьшения декрементированием.

При составлении программ команды INR и DCR чаще всего используют в сочетании с командами логических операций и переходов.

32. DAD P

Команда DAD P позволяет складывать два шестнадцатиразрядных числа. Одно из этих чисел помещают в регистровую пару HL, а другое в регистровую пару P (BC, DE или SP). Результат сложения размещается в регистровой паре HL. Команда DAD P однобайтовая.

Пример: сложить два числа 1263H и 0284H.

Решение: LXI H, 1263H

LXI B, 0284H

DAD B

33.DAA

Команда DAA служит для десятичной коррекции при выполнении особого способа двоично-десятичного сложения, при котором числа из двоичного кода переводятся в двоично-десятичный формат.

Команды логических операций.

Команды логических операций во многом схожи с командами арифметических операций. Они так же оперируют с двумя восьмиразрядными двоичными числами, одно из которых должно находиться в аккумуляторе, а второе – регистре R, в ячейке памяти M или во втором байте команды.

К группе логических команд условилось относить так же операции сравнения двух чисел и сдвига числа, находящегося в аккумуляторе.

Логические действия выполняются над числами поразрядно, то есть нулевой разряд одного числа логически сравнивается с нулевым разрядом другого числа, и результат сравнения располагается в нулевом разряде аккумулятора. Затем такое же сравнение производится над первыми разрядами чисел, потом над вторыми и т.д. до последних седьмых разрядов.

33. ANA R

34. ANA M

35. ANI K

Команды ANA R и ANI K выполняют логическое умножение (операцию И) над двумя восьми разрядными двоичными числами, одно из которых находится в аккумуляторе, а второе в регистре R (ANA R) или в памяти (ANA M) или во втором байте команды ANI K. Перед выполнением команды ANA M адрес ячейки памяти в которой хранится число должен быть занесен в регистровую пару HL.

Команды ANA R и ANA M – однобайтовые, а команда ANI K – двухбайтовая.

Результат операции И во всех трех командах размещается в аккумуляторе.

36. ORA R
37. ORA M
38. ORI K

Эти команды выполняют логическое сложение (операцию ИЛИ). По своей структуре и порядку выполнения они полностью соответствуют командам логического умножения

ORA R – ANA R
ORA M – ANA M
ORI K – ANI K

39. XRA R
40. XRA M
41. XRI K

Эти три команды выполняют логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ. Эти команды полностью соответствуют структуре и правилам выполнения уже рассмотренных команд И и ИЛИ.

Довольно часто команду XRA A используют для очистки аккумулятора, т.е. для записи в его разряды всех нулей. Так например эту операцию можно выполнить с помощью двухбайтовой команды

MVI A, 00H

А если использовать команду XRA A то она займет в памяти один байт.

Пусть в аккумуляторе было записано восьмиразрядное число 01010101. В. операция XRA A осуществляет следующее действие:

01010101
+
01010101
00000000

Таким образом в аккумуляторе записано число 00H.

42. CMA

Команда CMA выполняет логическую операцию НЕ. В результате этой операции все единицы во всех разрядах числа заменяются на нули, а все нули – на единицы, т.е. число инвертируется.

43. CMP R
44. CMP M
45. CPI K

С помощью этих трех команд выполняют сравнение двух чисел одно из которых находится в аккумуляторе а второе в регистре R, либо в ячейке памяти M, либо во втором байте команды.

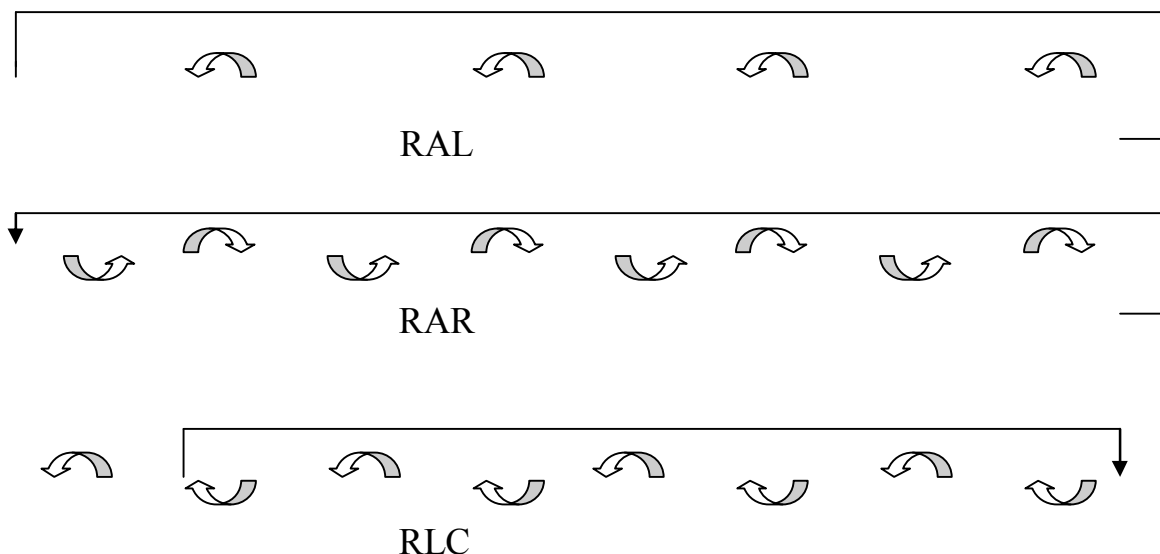
Сравнение выполняется путем внутреннего вычитания содержимого регистра R, ячейки памяти M или числа K из содержимого аккумулятора. Результат вычитания определяется по состоянию триггеров Z и AC. Если сравниваемые числа равны друг другу то триггер Z устанавливается в 1, а если не равны то сбрасывается в нуль. Когда сравниваемое число больше того, которое находится в аккумуляторе, то срабатывает триггер AC, устанавливаясь в единицу, а если меньше то устанавливается в нуль.

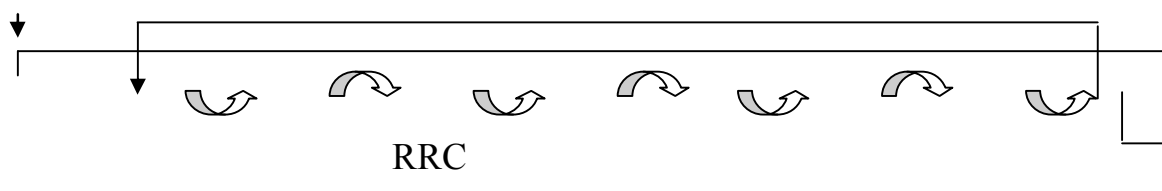
Команда CMP по своему действию соответствует команде вычитания SUB, с той лишь разницей что результат вычислений после выполнения команды CMP не размещается в аккумуляторе, не изменяет своего значения

46. RAL	48. RAR
47. RLC	49. RRC

Эти четыре команды выполняют операцию сдвига числа находящегося в аккумуляторе. Операцией сдвига называется перемещение каждого разряда числа на одну позицию влево (RAL и RLC) или вправо (RAR и RRA).

В операциях сдвига принимает участие триггер C регистра признаков F, расположенный левее старших разрядов аккумулятора и представляет собой как бы продолжение аккумулятора. На рисунке схематически показано как происходит сдвиг чисел:





Команды сдвига применяются для выполнения умножения деления и некоторых преобразований данных поступающих в микропроцессор.

50. STC

Перед выполнением некоторых логических и арифметических операций требуется предварительная установка в единицу триггера переноса С, входящего в состав регистра признаков.

В каком бы положении не находился триггер С после команды STC, он устанавливается в единицу.

Команды перехода.

К командам перехода в МП-580 относятся команды безусловного перехода, когда по команде происходит переход по указанному адресу, расположенному во втором и третьем байтах команды и команды условного перехода, когда переход по адресу осуществляется в зависимости от состояния специальных триггеров условий.

Команды безусловного перехода:

51. JMP ADR

По команде JMP ADR выполняется безусловный переход к выполнению программы, адрес которой указывается во втором и третьем байтах команды. Для возвращения к прежней программе в конце новой программы необходимо снова записать JMP ADR, указав в ней адрес того места программы, с которого начался переход.

52. PCHL

Команда PCHL осуществляет безусловный переход к выполнению программы, адрес которой находится в регистровой паре HL. Команда PCHL – однобайтовая.

Перед выполнением команды необходимо в регистровую пару HL поместить начальный адрес новой программы.

53. CALL ADR	54. RET
--------------	---------

Команда CALL ADR осуществляет безусловный переход к выполнению подпрограммы по адресу, указанному во втором и третьем байтах команды, а команда RET – безусловное возвращение к основной программе после выполнения подпрограммы. С этой целью адрес возврата размещается в ячейках памяти стека.

Команды условного перехода.

Напомним что команды условного перехода осуществляют переход в зависимости от условия, записанного в команде.

Так например, в языках высокого уровня команда условного перехода выглядит так:

IF< условие >GO TO ADR

Выполняется команда следующим образом: если условие записанное в операторе истинно, то выполняется оператор следующий за условием, а если условие ложно то выполняется следующий за оператором IF оператор.

Точно так же осуществляется выполнение команды перехода в МП-580.

53. J(F) ADR

По команде J(F) ADR осуществляется условный переход к выполнению программы, адрес которой записан во втором и третьем байтах команды, если условие F выполняется. Если условие F не выполняется то перехода не происходит, а продолжается выполнение команд основной программе. Различают восемь отдельных команд условного перехода J(F) ADR учитывающих признаки конкретного триггера, входящего в состав регистра F:

JNZ ADR – если не нулевой результат ($z=0$)

JN ADR – если нулевой результат ($z=1$)

JP ADR – если число положительное ($s=0$)

JM ADR – если число отрицательное ($s=1$)

JPO ADR – если нечетный результат ($p=0$)

JPE ADR – если четный результат ($p=1$)

JNC ADR – если нет переноса ($c=0$)

JC ADR – если есть перенос ($c=1$)

Другой разновидностью команды условного перехода является команда вызова подпрограммы при выполнении условия, учитывающего признаки триггеров.

54. C(F) ADR

55. R(F)

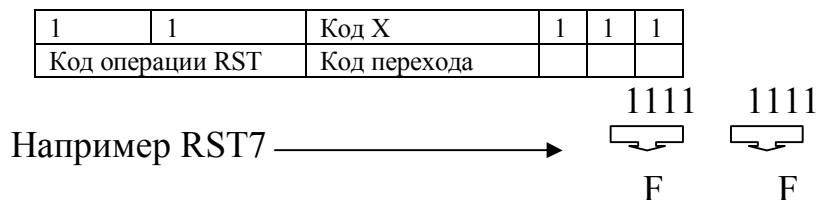
Команды C(F) ADR и R(F) аналогичны командам CALL ADR и RET и отличаются от них только тем, что переход к подпрограмме и возвращение из нее осуществляется в зависимости от состояния триггеров регистра признаков. Различают восемь команд C(F) ADR и R(F):

- CNZ ADR и RNZ, если $z=0$
- CZ ADR и RZ, если $z=1$
- CP ADR и RP, если $s=0$
- CM ADR и RM, если $s=1$
- CPO ADR и RPO, если $p=0$
- CPE ADR и RPE, если $p=1$
- CNC ADR и RNC, если $c=0$
- CC ADR и RC, если $c=1$

56. RSTX

Команда RSTX осуществляет переход к одной из восьми специальных подпрограмм режима прерывания, расположенных по фиксированному адресу X.

Команда RSTX является однобайтовым вариантом команды перехода с возвратом. Адрес очередной команды определяется тремя разрядами X содержащимися в коде операции:



Так например в ЭВМ при нажатии на клавиши в клавиатуре в МП поступает запрос на прерывание. МП останавливает выполнение основной программы и считывает из внешнего для него устройства – клавиатуры код команды RST X, сформированный самой клавиатурой. После чего МП приступает к выполнению подпрограммы обслуживания прерывания, адрес которой находится в коде команды RST X.

Специальные команды.

Эта группа команд не производит каких-либо действий над данными. Команды не имеют операнда, а только код операции. Большинство из них является служебными командами изменяющими режим работы МП.

57. EI	58. DI
--------	--------

Команда EI разрешает прерывание и устанавливает триггера разрешения прерывания и связанную с ним выходную линию МП в состояние 1.

Команда DI запрещает прерывание устанавливает соответственный триггер и линию в состояние 0.

В общем случае прерывание является крайней ситуацией для МП. Иногда они нужны, иногда нежелательны, поэтому управление МП командами EI и DI позволяет эффективно использовать возможности МП.

59. NOP

Команда NOP является пустой операцией. Код этой операции ООП. МП не выполняет здесь каких либо полезных действий по обработке данных, а просто затрачивает некоторое время для различных служебных переключений во время четырех машинных тактов. Зная длительность машинного такта, можно с помощью некоторого количества команд NOP получить нужную задержку по времени выполнения программы.

Команду NOP часто используют для пространственного заполнения программы. Если появляется необходимость ввести в программу дополнительную команду, то ее размещают взамен команд NOP, записанных в нужном месте программы

60. HLT

По команде HLT микропроцессор останавливает выполнение очередной команды и переходит в режим 1111111. вывести его из этого состояния можно подав сигнал сброса или прерывания.