**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ**

**Н.Г. ЧЕРНЫШЕВСКОГО»**

| **Микропроцессорная система на базе КР580ВМ80А** |
| --- |

**РЕФЕРАТ**

студента 4 курса 421 группы

направления 09.03.01 «Информатика и вычислительная техника»

факультет Компьютерных наук и информационных технологий

Наумова Никиты Игоревича

|  |  | |
| --- | --- | --- |
| Проверил:  д.ф.- м.н., профессор | подпись, дата  Саратов 2025 | А.А. Семёнов | |

**СОДЕРЖАНИЕ**

**Введение**…………………………………………………………………………..3

**1. Микропроцессор КР580ВМ80**……………………………………………….4

1.1. Устройство и принцип работы микропроцессора, блок-схема, характеристики……………………………………………………………………4

1.2. Режимы работы микропроцессора…………………………………………..8

1.3. Схема включения микропроцессора, формирование шин………………..10

1.4. Дешифрация адресного пространства микропроцессора (ОЗУ и ПЗУ 8192 байта). Карта памяти…………………………………………………………….12

1.5. Система команд микропроцессора, типы команд, группы команд по назначению……………………………………………………………………….14

1.6. Селекция и дешифрация пространства устройств ввода-вывода………..17  
**2. Периферийные устройства микропроцессорной системы**……………..19  
2.1. Параллельный периферийный адаптер (ППА) К580ВВ55……………….19  
2.1.1. Устройство и принцип работы ППА……………………………………..19  
2.1.2. Режимы работы ППА……………………………………………………..23  
2.1.3. Настройка ППА…………………………………………………………...25  
2.1.4. Схема включения ППА в микропроцессорную систему……………….27  
2.2. Программируемый интервальный таймер К580ВИ53……………………28  
2.2.1. Устройство и принцип работы интервального таймера………………...28  
2.2.2. Режимы работы интервального таймера………………………………...31  
2.2.3. Настройка интервального таймера………………………………………34   
2.2.4. Схема включения интервального таймера в микропроцессорную систему…………………………………………………………………………...35  
**3. Практическая часть**…………………………………………………………35

3.1. Лабораторная работа №1.7…………………………………………………36

3.2. Лабораторная работа №1.6…………………………………………………39

3.3. Заключение по практике……………………………………………………44

Заключение……………………………………………………………………...45

**Введение**

Появление микропроцессоров имело значительное влияние на развитие компьютерных технологий, обработки данных и управляющих систем, которые являются основой автоматизации в различных областях. Интерес к микропроцессорам сохраняется благодаря их надежности, компактности и высоким вычислительным возможностям, которые позволяют применять их даже в случаях, где раньше не рассматривалась целесообразность цифровой обработки информации.

Современные микропроцессоры представляют собой сложные устройства, работы которых нельзя полноценно описать вне контекста системы, в которой они используются. Изучение микропроцессорных систем остается актуальным, так как они широко применяются практически повсеместно.

КР580ВМ80, объект изучения, является устаревшим микропроцессором, основным компонентом микропроцессорного комплекта серии КР580. Разработанный Киевским НИИ микроприборов под руководством А. В. Кобылинского, этот микропроцессор выпускался с 1977 года по середину 1990-х. Не смотря на свою давность, этот микропроцессор характеризуется простотой, хорошей документацией и совместимостью с более современными процессорами.

Цель и задачи курса «Микропроцессорные систем» - это выработка навыков и умений построения микропроцессорной системы фактически на любом процессоре с любым набором сопутствующих интегральных схем. Разработка программного обеспечения микропроцессорной системы на языке ассемблера с конструированием программ реального времени.

**1 Микропроцессор КР580ВМ80**

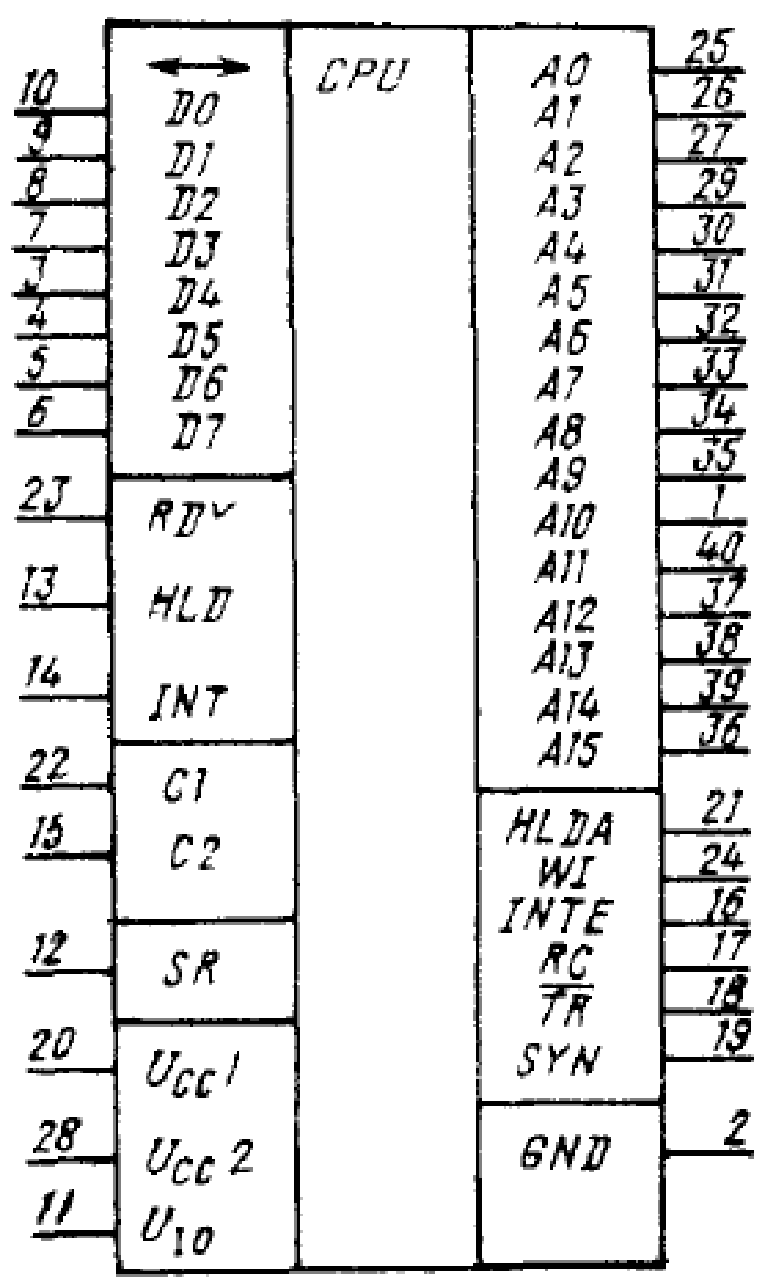
**1.1. Устройство и принцип работы микропроцессора, блок-схема, характеристики**

Микропроцессор (МП) KP580BM80 является аналогом известного МП 8080A, разработанного фирмой «Интел» (США). Микросхема КР580ВМ80А – функционально законченный однокристальный параллельный 8-разрядный микропроцессор с фиксированной системой команд, применяется в качестве центрального процессора в устройствах обработки данных и управления.

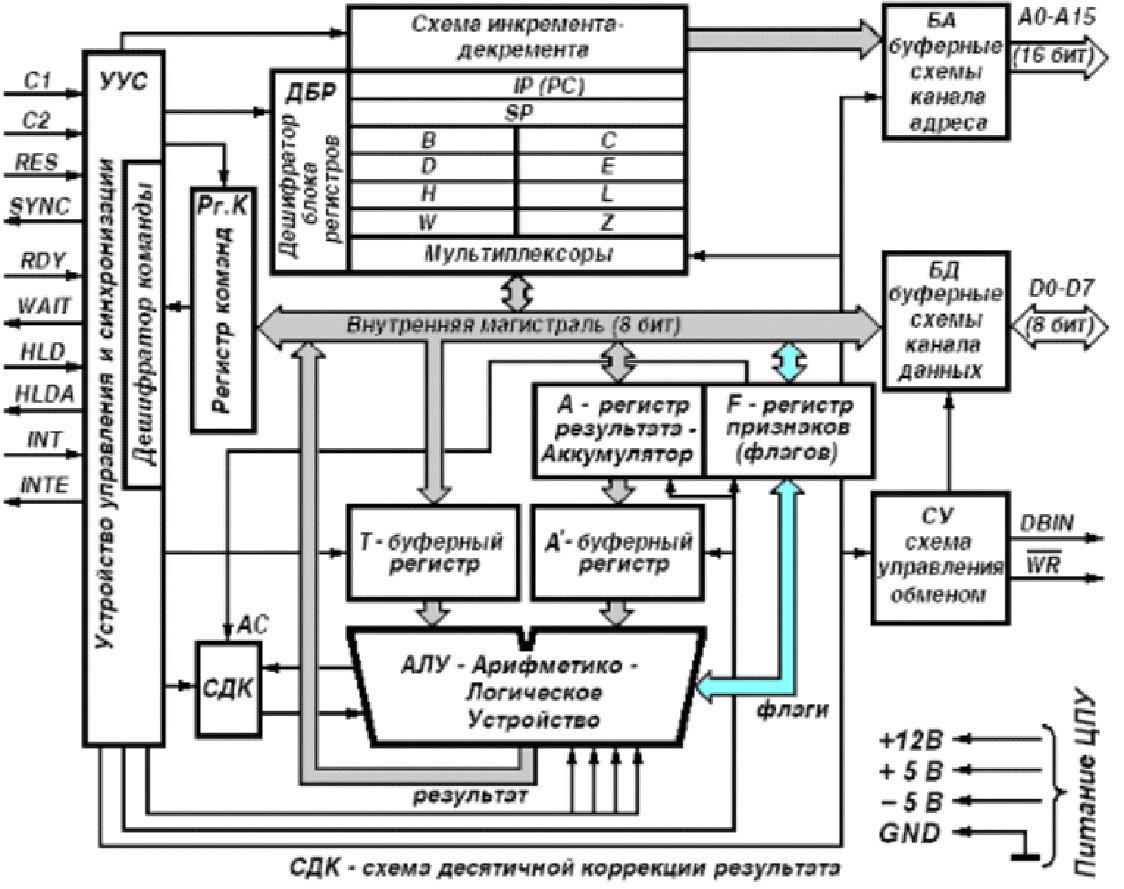
Микропроцессор выполняет команды по машинным циклам. Число циклов, необходимое для выполнения команды, зависит от ее типа и может быть от одного до пяти. Машинные циклы выполняются по машинным тактам. Число тактов в цикле определяется кодом выполняемой команды и может быть от трех до пяти. Минимальное время выполнения простых регистровых команд 4 такта. Номинальная частота процессора - 2 МГц, получаем период равный 500 нс. Максимальная тактовая частота для процессора - 3,5 МГц. Для процессоров военной приемки - 4 МГц.

Микропроцессор имеет раздельные 16-разрядный канал адреса и 8-разрядный канал данных.

Условное графическое изображение микропроцессора приведено на рисунке 1.

  
Рисунок 1 – Условное графическое изображение микропроцессора

Восьмиразрядное арифметико-логическое устройство микропроцессора обеспечивает выполнение арифметических и логических операций над двоичными данными, представленными в дополнительном коде, а также обработку двоично-десятичных упакованных чисел. В процессоре присутствует блок регистров общего назначения, в составе которого можно выделить регистры временного хранения WZ и шесть 8-разрядных регистров общего назначения (B, C, D, E, H, L). 8-ми разрядные регистры общего назначения могут объединяться в 16-ти разрядные регистровые пары. При этом пара называется по старшему разряду. В составе процессора есть 2 специальных 16-ти разрядных регистра - это программный счетчик, указатель стека - Stack Pointer. SP устанавливается программно и указывает на область оперативной памяти, куда мы можем довольно быстро сохранять промежуточные значения. Cтек работает по принципу FIFO. Стек устанавливаем по адресу 1000:LXI SP, 1000H. Процессор автоматически использует стек для сохранения адресов возврата из подпрограмм и прерываний. Стек всегда должен быть в ОЗУ и находится в области, которая не мешает основной программе. На рисунке 2 представлена структура микропроцессорной БИС (большая интегральная схема) КР580ВМ80.

  
Рисунок 2 – Структура микропроцессорной БИС КР580ВМ80

**1.2. Режимы работы микропроцессора**

Режим сброса – выполняется за четыре такта и за это время программный счетчик устанавливается в ноль прерывание запрещается. Режим прерывания - внешнее устройство обращается к микропроцессору с запросом на обслуживание аппаратным сигналом по входу INT. По старту микропроцессора прерывания аппаратно запрещены. Их разрешает команда EI и запрещает DI. Если прерывания разрешены, то микропроцессор получает от внешней схемы коды команд CALL ADDR или RST N, выполняет подпрограмму обслуживания прерывания и возвращается из неё на тот адрес, где выполнялась программа в момент возникновения прерывания.

Режим прямого доступа к памяти используется в тех случаях, когда в системе имеется устройство с высокой скоростью передачи больших массивов информации (например, накопители на гибких магнитных дисках или накопители на жестких дисках). Для работы с такими устройствами применяется специальная БИС контроллера Прямого Доступа к Памяти (ПДП), которая запрашивает у процессора доступ к шинам активным сигналом HOLD (Запрос Захвата), по которому микропроцессор переводит свои выводы в высокоимпедансное состояние и отдает управление контроллеру ПДП. Последний способен с высокой скоростью либо записать массив последовательно расположенных байт информацией из внешнего устройства, либо передать ему такой массив.

Суть режима останова заключается в том, что микропроцессор в процессе прохождения программы выполняет специфическую служебную команду HLT (HALT – Останов). Встретив такую команду, микропроцессор приостанавливает дальнейшее выполнение программы на адресе команды HLT. Вывести микропроцессор из состояния останова может прерывание, если прерывания разрешены, или сигнал начальной установки RESET.

Режим ожидания используется в тех случаях, когда быстродействие микропроцессора превышает скоростные характеристики микросхем памяти или подсоединенных в систему внешних устройств. В этом случае аппаратно формируется сигнал «неготовности» устройства, который подается на вход микропроцессора, и последний приостанавливает свою работу, начиная выполнять «холостые» такты ожидания, до тех пор, пока сигнал READY не сообщит о готовности. Этот режим удобен для аппаратной пошаговой отладки системы.

**1.3. Схема включения микропроцессора, формирование шин**

Рисунок 3 показывает схему включения МП типа КР580ВМ80. В состав устройства в данном случае, кроме самого МП, входят также тактовый генератор (G) КР580ГФ24, системный контроллер (SC) КР580ВК28 и два буферных регистра (F) типа КР580А87 или ВА86 (первый из них инвертирует сигналы, второй не инвертирует).

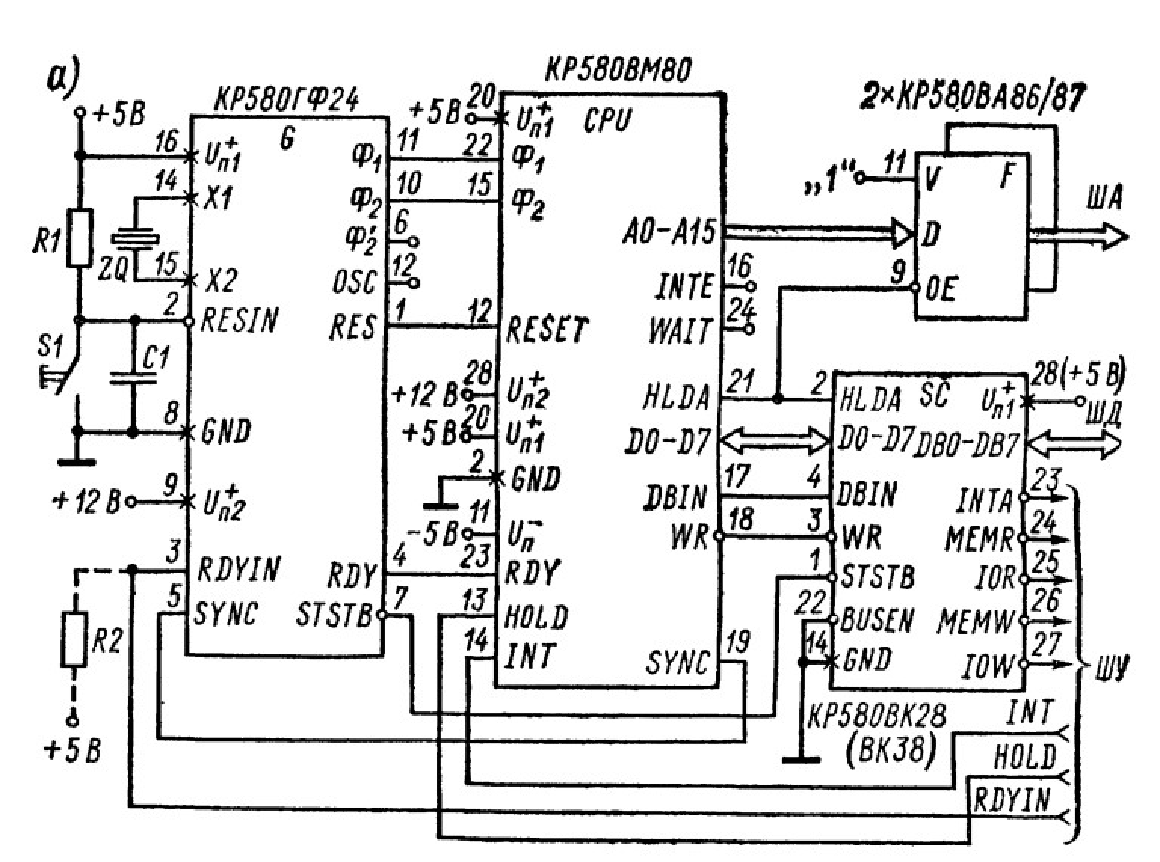


Рисунок 3 – Схема включения МП типа КР580ВМ80.

Тактовый генератор вырабатывает неперекрывающиеся положительные импульсы Ф1 и Ф2, имеющие амплитуду 12В и синхронизирующие работу МП-системы. Кроме того, вырабатываются импульсы Ф`2, совпадающие по времени с импульсами Ф2, но имеющие амплитуду, соответствующую ТТЛ-схемам. Эти импульсы при необходимости могут использоваться для синхронизации работы внешних устройств.

Частота тактовых импульсов стабилизируется кварцевым резонатором ZQ, резонансную частоту которого следует выбирать в 9 раз превышающей требуемую тактовую частоту fт- процессора. Обычное значение fт =2 МГц, для чего требуется кварцевый резонатор на 18 МГц.

Наряду с выдачей тактовых импульсов тактовый генератор осуществляет формирование и синхронизацию некоторых управляющих сигналов: сброса (RESIN-RES), готовности (RDYIN - RDY), строба (SYNC -STSTB). Сброс ЦПУ при включении питания реализуется с помощью RC-цепи (например, R1=10 кОм, C1=1 мкФ), присоединенной ко входу RESIN. В дальнейшем сброс может осуществляться с помощью кнопки S1.

Входной сигнал готовности RDYIN поступает на генератор от устройств памяти или ввода - вывода. Если эти устройства имеют достаточное быстродействие для того, чтобы своевременно реагировать на команды ЦПУ, то можно на вход DRYIN подать постоянный сигнал готовности, соединив этот вход резистором с источником питания -+5В.

Системный контроллер SC выполняет роль буфера шины данных и формирует сигналы управления для устройств памяти и ввода - вывода. Как упоминалось в предыдущем параграфе, часть сигналов управления передается из ЦПУ по шине данных в режиме разделения времени. Эти сигналы управления фиксируются системным контроллером в начале каждого цикла команды по сигналу строба STSTB. Кроме того, три сигнала управления: HLDA (подтверждение перехода в режим ПДП), DBIN (переключение шины данных на ввод) и WR (запись информации) - подаются на SC из ЦПУ непосредственно по отдельным проводам. На основе принятых сигналов системный контроллер вырабатывает управляющие сигналы INTA (подтверждение прерывания), MEMR (чтение из ячейки памяти), IOR (чтение из устройства ввода - вывода), MEMW (запись в ячейку памяти), IOW (запись в устройство ввода - вывода).

В шину управления входят также линии, подающие сигналы от указанных устройств на процессор. Это сигнал запроса прерывания INT, сигнал запроса на захват шин от контроллера прямого доступа к памяти HOLD, сигнал готовности устройства к выполнению операции RDYIN. Кроме упомянутых, ЦПУ вырабатывает еще два сигнала управления, которые могут быть использованы при построении МП-системы. Сюда относятся сигнал WAIT, свидетельствующий, что ЦПУ находится в режиме ожидания сигнала готовности RDY, и сигнал разрешения прерывания INTE, информирующий о том, что ЦПУ готово реагировать на сигнал прерывания INT.

Буферные каскады шины адреса могут быть реализованы не только с помощью микросхем КР580ВA86/87, но и с помощью микросхем серии ТТЛ, например, типа Қ155ЛП10/11, Қ155ЛНб.

**1.4. Дешифрация адресного пространства микропроцессора.** Дешифрация адресного пространства микропроцессора с ОЗУ и ПЗУ по 1024 байта представлена на рисунке 4.

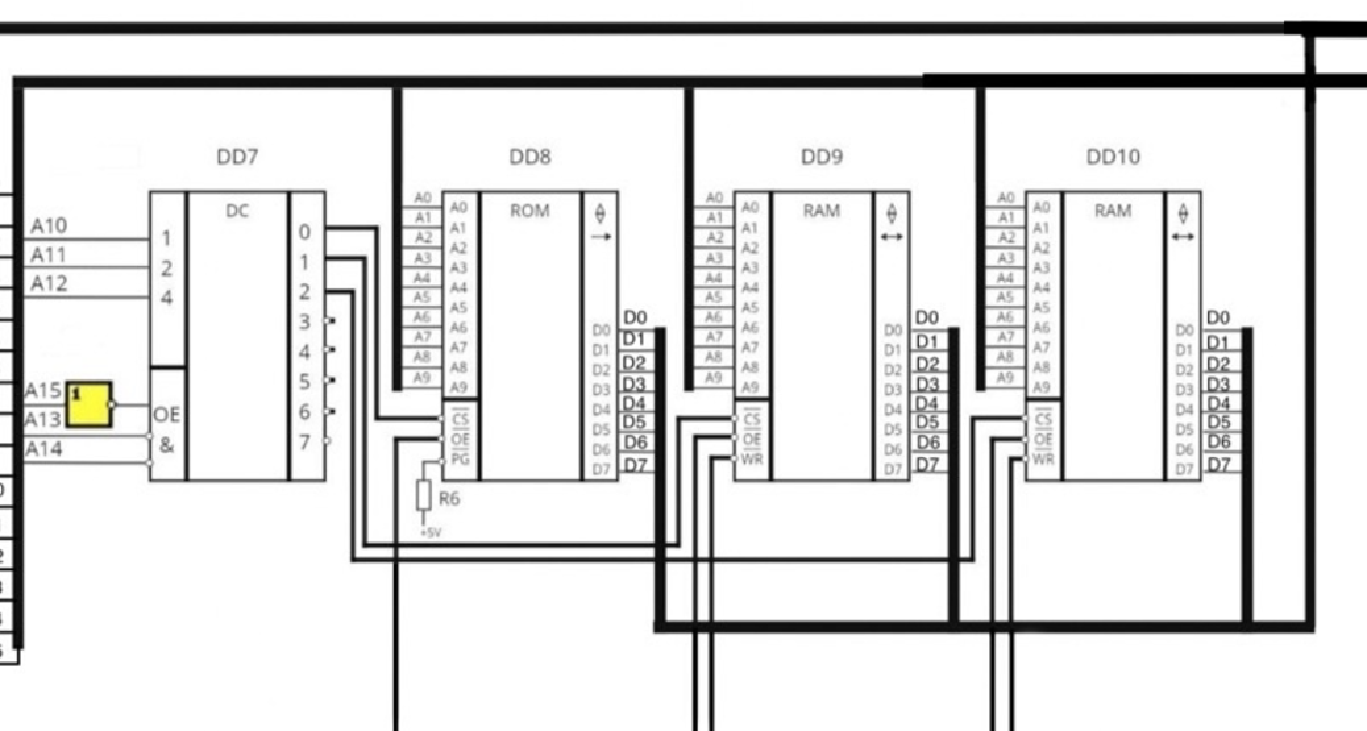


Рисунок 4 – Дешифрация пространства памяти МПС

**1.5. Система команд микропроцессора, типы команд, группы команд по назначению**

Система команд микропроцессора КР580ВМ80А состоит из 78 базовых команд, которые можно разделить на пять групп:

* Команды передачи данных – используются для передачи данных из регистра в регистр, из памяти в регистр, из регистра в память.
* Арифметические команды – используются для сложения, вычитания, инкремента или декремента содержимого регистров или ячейки памяти.
* Логические команды И, ИЛИ, исключающее ИЛИ, сравнение, сдвиги.
* Команды переходов – используются для условных и безусловных переходов, вызова подпрограмм и возврата из них.
* Команды управления, ввода/вывода и работы со стеком – используются для управления прерыванием, регистром признаков, ввода и вывода информации.

В микропроцессоре КР580ВМ80А принят формат информационного слова, представляющего собой 8 разрядное двоичное слово (байт). Формат информационного слова представлен на рисунке 5 где D7 – старший разряд слова, D0 – младший. Отрицательные числа хранятся в памяти в дополнительном коде.

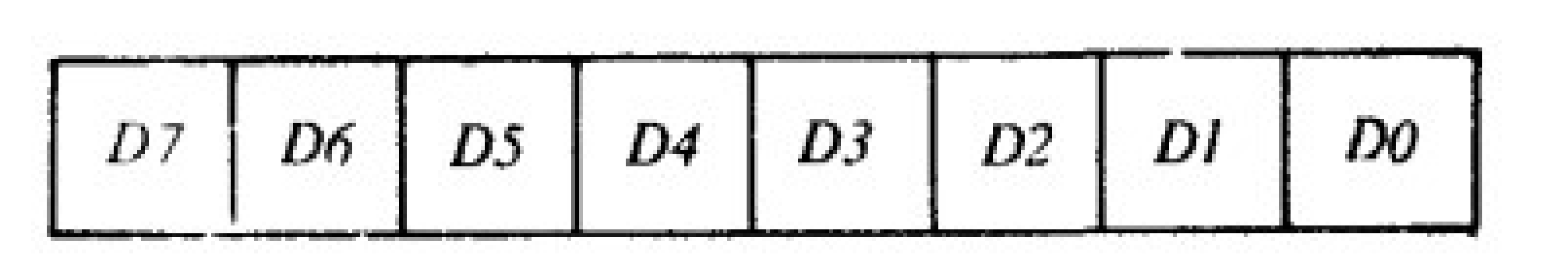


Рисунок 5– Информационное слово микропроцессора КР580ВМ80А.

Формат команды зависит от типа операции и может быть одно, двух и трехбайтовым. Байты двух и трехбайтовых команд должны храниться в ячейках памяти, следующих одна за другой. Адрес первого байта всегда является адресом кода операции. Формат команд микропроцессора представлен на рисунке 6



Рисунок 6 Формат команд микропроцессора КР580ВМ80А.

Рассмотрим какие команды относятся к группам команд микропроцессора КР580ВМ80А.

Команды передачи данных: MOV, MVI, LXI, LDA, STA, LDAX, STAX, IN, OUT, PUSH, POP, XCHG, XTHL, SPHL, LHLD, SHLD, SHLD.

Арифметические команды: ADD, ADC, ADI, SUB, SBB, SUI, SBI, INR, DCR, DAD, DAA.

Команды логических операций: ANA, ANI, ORA, ORI, XRA, XRI, CMA, CMP, CPI, RAL, RAR, RLC, RRC.

Команды перехода: JMP, PCHL, CALL, RET.

Команды условного перехода: J (F – NZ, N, P, M, PO, PE, NC, C), C(F - NZ, N, P, M, PO, PE, NC, C), RSTX.

**1.6. Селекция и дешифрация пространства устройств ввода-вывода**

Микропроцессор КР580ВМ80 имеет возможность адресации до 256 устройств ввода и 256 устройств вывода. При передаче адреса по общей шине адреса, ЦПУ определяет, что нужно активировать - ячейку памяти, устройство ввода или устройство вывода. Это решается сигналами управления: MEMR (чтение из памяти), MEMW (запись в память), IOR (чтение из устройства ввода-вывода), IOW (запись в устройство). По младшему байту адреса обращения к устройствам ввода-вывода дублируется на старший байт, поскольку для них достаточно одного байта адресного кода. Хотя устройства ввода-вывода могут адресоваться так же, как и ячейки памяти, им выделены отдельные сигналы управления. Это связано с тем, что устройства ввода-вывода могут быть размещены в ячейках памяти, но для работы им необходим двухбайтный дешифратор. При использовании устройств ввода-вывода как отдельных устройств, можно обойтись дешифратором на 8 входов. Сокращение адресации устройств ввода-вывода делает программы более компактными, а отдельные команды управления обеспечивают более ясные программы.

Дешифратор адресного пространства устройств ввода-вывода выделяет старшие 32 байта в верхних адресах FFE0 -- FFFF адресного пространства и разделяет по 4 байта.

На рисунке 7 изображен дешифратор и селектор пространства устройств ввода-вывода.

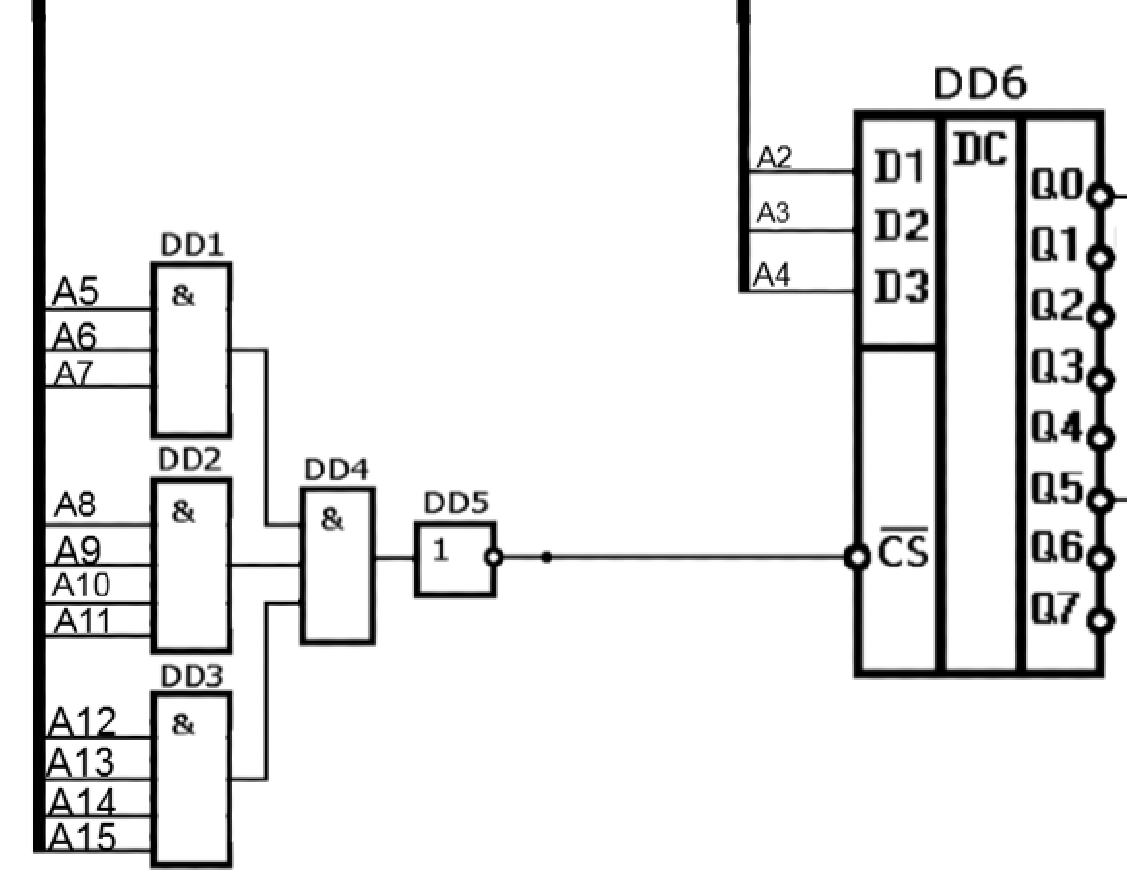


Рисунок 7 – Дешифрация адресного пространства устройств ввода-вывода (УВВ).

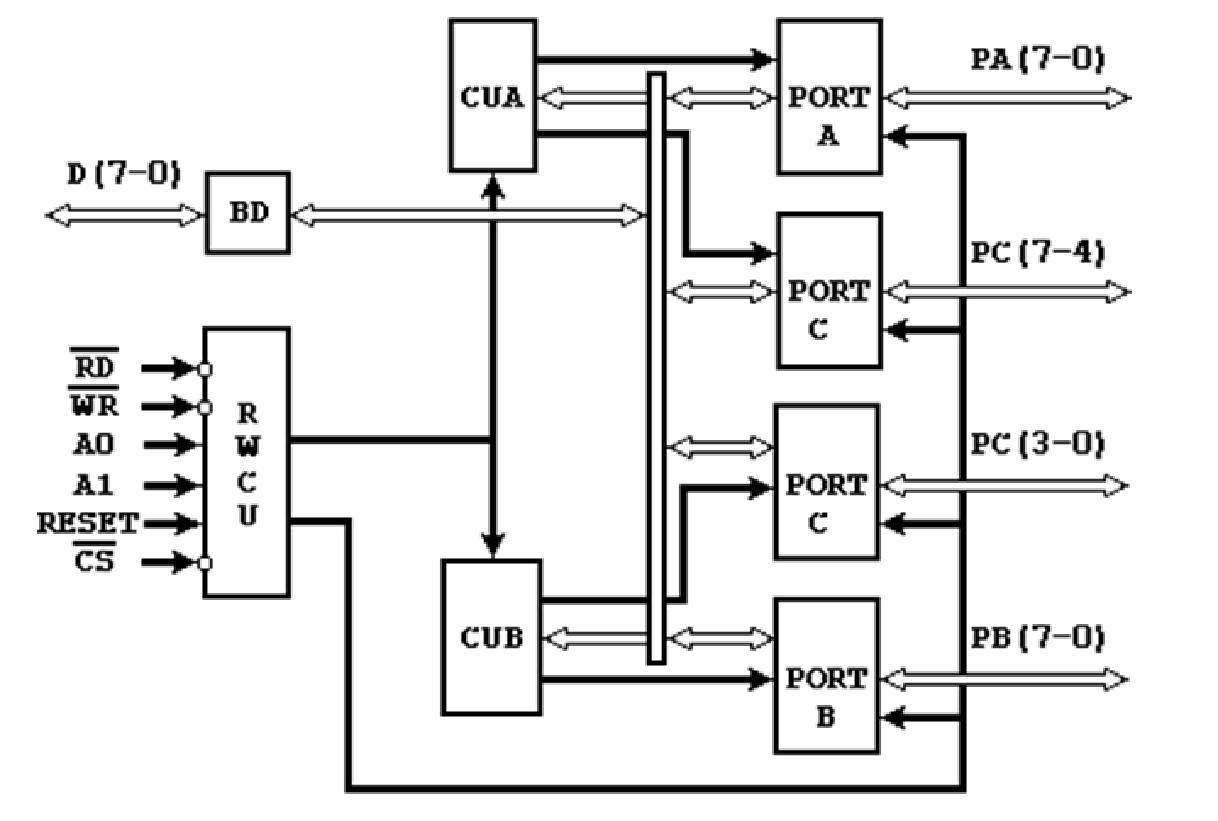
**2. Периферийные устройства микропроцессорной системы**

**2.1. Параллельный периферийный адаптер (ППА) К580ВВ55**

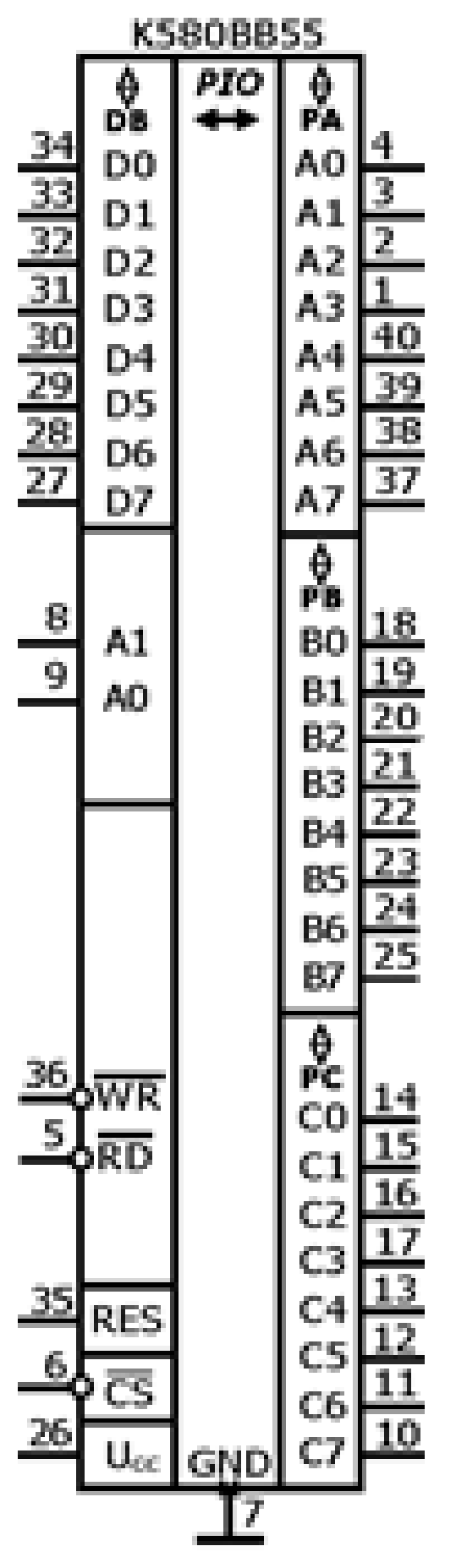
**2.1.1. Устройство и принцип работы ППА**

В комплекте микропроцессора К580 выполнение функций программируемого устройства ввода/вывода параллельной информации различного формата осуществляется с помощью микросхемы БИС параллельного периферийного адаптера К580ВВ55А. Эта микросхема предназначена для управления вводом и выводом данных в параллельном виде и поддерживает большинство стандартных протоколов обмена по параллельным каналам. Обычно БИС ППА используется для соединения микропроцессорной системы с обычным периферийным оборудованием (таким как цифровой дисплей, принтер, телетайп, внешние устройства хранения данных) через три 8-разрядных канала.

Как показано на рисунке 8, микросхема включает в себя: двунаправленный 8-разрядный буфер данных для связи с системной шиной данных; блок для управления записью/чтением, который регулирует передачу данных внутри и за пределами устройства, обработку управляющих слов и информацию о состоянии; три 8-разрядных канала ввода/вывода (PORT A, PORT B, PORT C) для обмена информацией с внешними устройствами; схему управления группой А, которая генерирует сигналы управления для канала А и старшие разряды канала С; схему управления группой В, которая генерирует сигналы управления для канала В и младшие разряды канала С.

  
Рисунок 8 – Структурная схема ППА

Условное графическое обозначение ППА представлено на рисунке 9.

  
Рисунок 9 – Условное графическое обозначение ППА

**2.1.2. Режимы работы ППА**

**Режим 0:** применяется при синхронном обмене или при программной организации асинхронного обмена. Микросхема может рассматриваться в этом режиме как устройство, состоящее из 4-х портов (два 8-разрядных и два 4-разрядных), независимо настраиваемых на ввод или вывод. Вывод информации осуществляется по команде OUT микропроцессора с фиксацией выводимой информации в регистрах каналов, а ввод по команде IN без запоминания информации. В совмещенной карте памяти возможно обращение к портам, как к ячейкам памяти соответствующими командами работы с памятью.   
 **Режим 1:** обеспечивает стробируемый однонаправленный обмен информацией с внешним устройством. Передача данных производится по каналам А и В, а линии канала С управляют передачей. Работу канала в режиме 1 сопровождают три управляющих сигнала. Если один из каналов запрограммировать на режим 1, то остальные 13 интерфейсных линий можно использовать в режиме 0.   
 Если оба канала запрограммированы на режим 1, то оставшиеся 2 интерфейсные линии канала С могут быть настроены на ввод или вывод. В режиме 1 для ввода информации используются следующие управляющие сигналы:

* строб приема (STB) – входной сигнал, формируемый внешним устройством, указывает на готовность ВУ к вводу информации;
* подтверждение приема (IBF) – выходной сигнал ППА, сообщающий ВУ об окончании приема данных в канал, формируется по спаду STB;
* запрос прерывания (INTR) – выходной сигнал ППА, информирующий МП о завершении приема информации в канале; Н-уровень сигнала устанавливается при STB=1, IBF=1 и INTE=1; сбрасывается спадом сигнала RD.

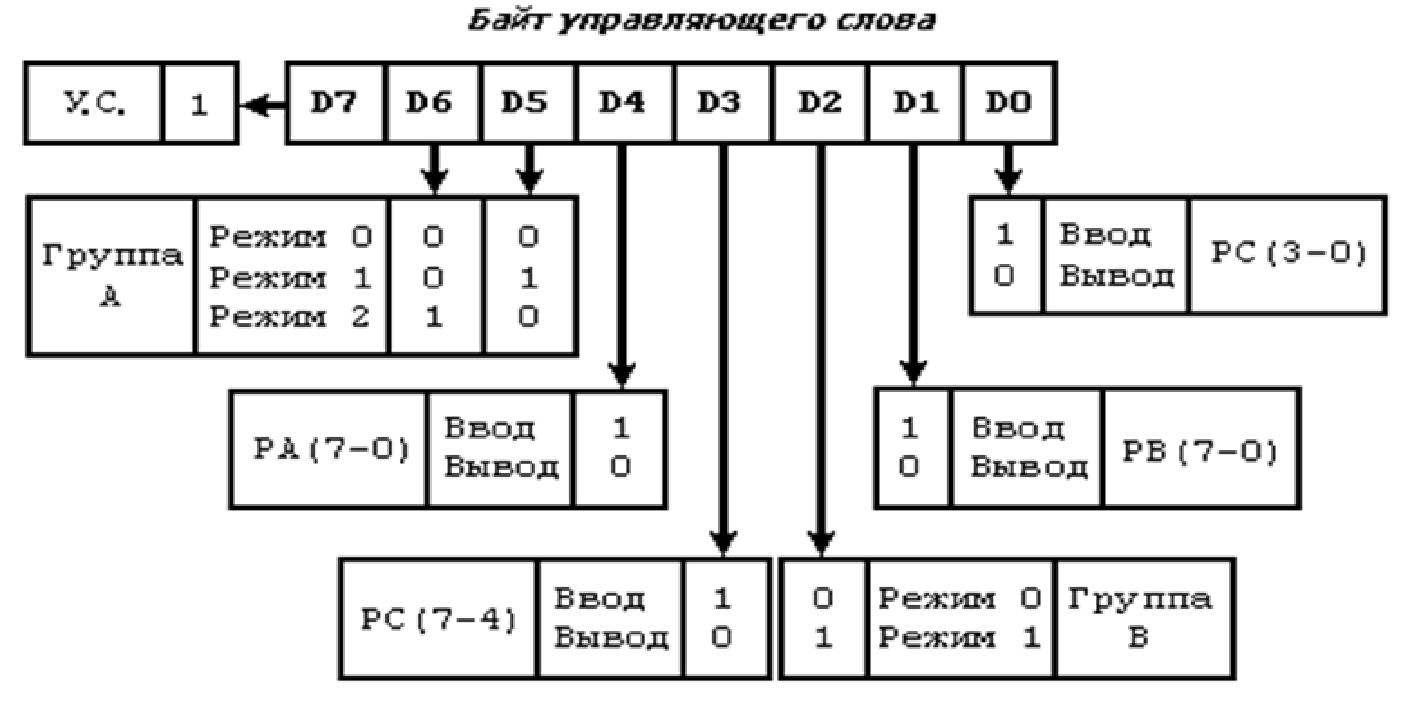
Для операции ввода управление сигналом INTE канала А осуществляется по линии PC4, а канала В – по линии PС2.   
 Для вывода информации в режиме 1 используются следующие управляющие сигналы:

* строб записи (OBF) – выходной сигнал, указывающий ВУ о готовности к выводу, формируется по фронту WR;
* подтверждение записи (ACK) – входной сигнал от ВУ, подтверждающий прием информации из ППА;
* запрос прерывания (INTR) – выходной сигнал ППА, информирующий МП о завершении операции вывода информации; Н-уровень сигнала устанавливается по фронту сигнала ACK при OBF=1 и INTR=1; сбрасывается сигналом WR.

Для операции вывода управление сигналом INTE канала А осуществляется по линии РС6, а канала В – по линии РС2.   
 **Режим 2**: обеспечивает двунаправленную передачу информации по каналу А к ВУ и обратно. Процесс обмена сопровождают 5 управляющих сигналов, подаваемых по линиям РС7-РС3. Оставшиеся 11 интерфейсных линий могут настраиваться на режим 0 или режим 1.   
 Функции управляющих сигналов аналогичны рассмотренным выше сигналам для режима 1. Управление установкой внутреннего сигнала INTE для операции ввода осуществляется по линии РС4, а для операции вывода - по линии РС6.

**2.1.3. Настройка ППА**

Для конфигурации в параллельном периферийном адаптере присутствует четвертый регистр - регистр управляющего слова, в который записывается байт для настройки всех трех портов одновременно. При запуске микропроцессорной системы все 24 линии ввода-вывода находятся в состоянии высокого сопротивления, что означает, что они не соединены ни с чем. После сброса все 24 линии установлены в режим ввода со значением 0.  
 Режим работы каждого из каналов ППА программируется с помощью управляющего слова.   
 Управляющее слово может задать один из трех режимов: – основной режим ввода/вывода (режим 0);   
– стробируемый ввод/вывод (режим 1);   
– режим двунаправленной передачи информации (режим 2).   
 Одним управляющим словом можно установить различные режимы работы для каждого из каналов. Формат управляющего слова представлен на рисунке 10.

  
Рисунок 10 – Формат управляющего слова

Канал А может работать в любом из трех режимов, канал В – в режимах 0 и 1. Канал С может быть использован для передачи данных только в режиме 0, а в остальных режимах он служит для передачи управляющих сигналов, сопровождающих процесс обмена по каналам А и В.   
 Разряд D7 управляющего слова определяет либо установку режимов работы каналов (D7=1), либо работу ППА в режиме сброса/установки отдельных разрядов канала С (D7=0). При поразрядном управлении каналом С разряды D3-D1 определяют номер модифицируемого разряда.   
 Разряд D0 задает сброс (D0=0) или установку (D0=1) модифицируемого разряда; разряды D6-D4 не используются. Сброс/установку разрядов канала С можно использовать для выработки сигналов запроса прерывания от ППА.   
 Для каждого из каналов А и В в ППА имеется триггер разрешения прерывания, установка/сброс которого осуществляется управляющим словом установки/сброса определенного разряда канала С. Если триггер разрешения прерывания соответствующего канала установлен (INTE=1), то ППА может сформировать сигнал запроса прерывания при готовности внешнего устройства к вводу или выводу.

**2.1.4. Схема включения ППА в микропроцессорную систему**

Параллельный адаптер подключается к шинам микропроцессора. Данные на одноименные данные, два адреса на соответствующие адреса.  
 На рисунке 11 изображена схема включения ППА в микропроцессорную систему.

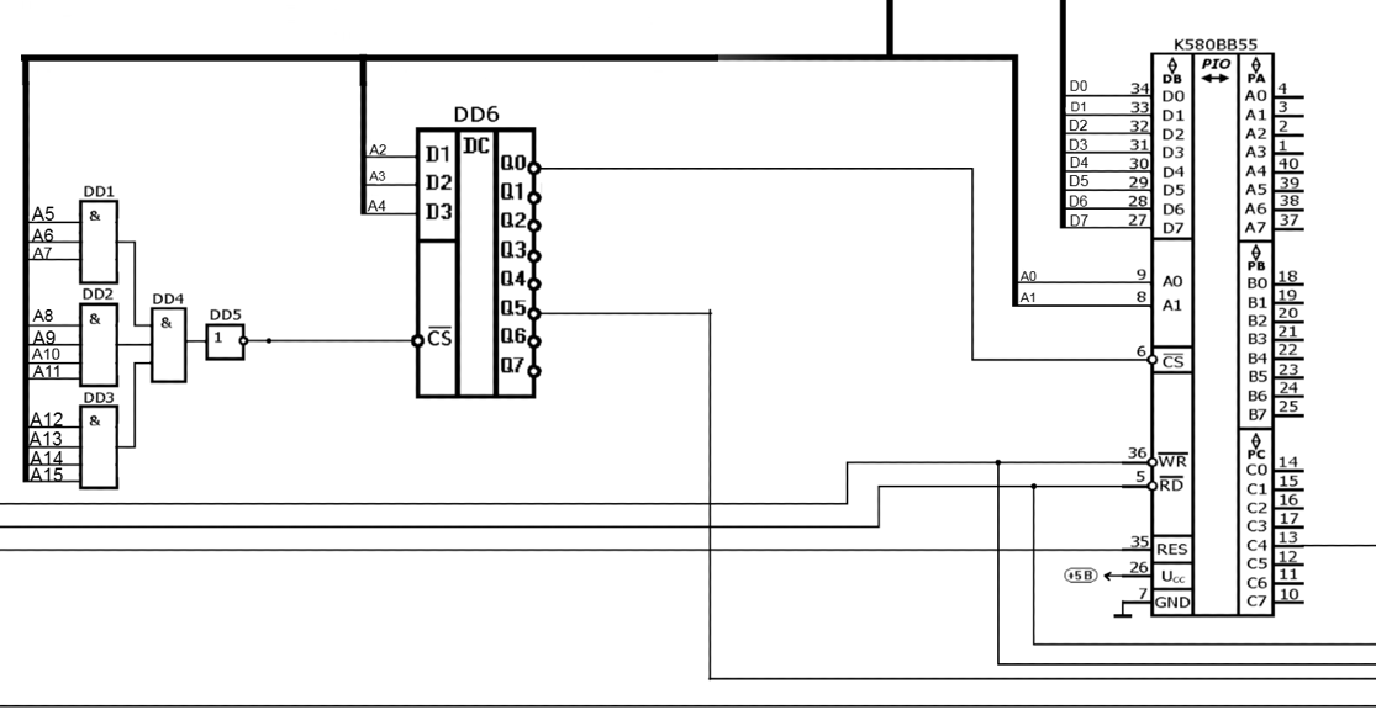


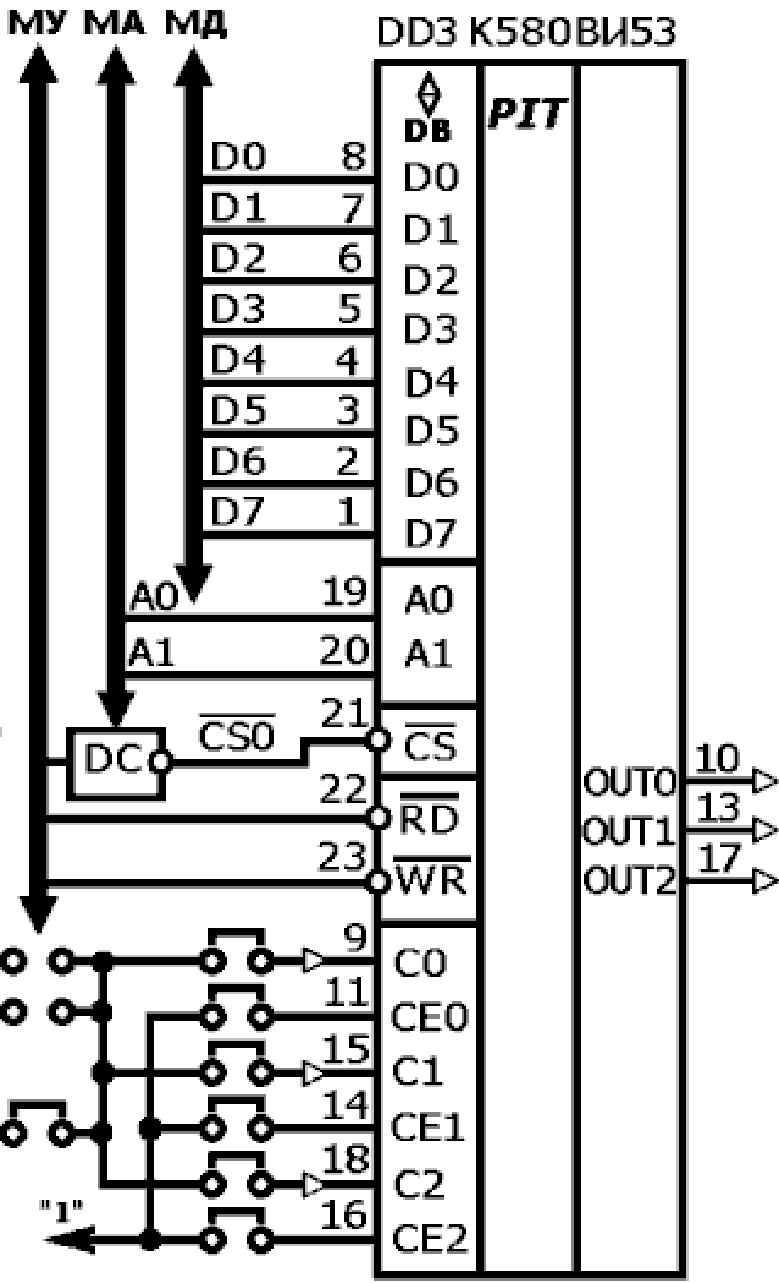
Рисунок 11 – Схема включения ППА в микропроцессорную систему

**2.2. Программируемый интервальный таймер К580ВИ53**

**2.2.1. Устройство и принцип работы интервального таймера**

В составе микропроцессорного комплекта К580 существует большая интегральная схема (БИС) программируемого интервального таймера КР580ВИ53 – трёхканальное программируемое устройство, предназначенное для организации работы микропроцессорных систем в режиме реального времени. Микросхема формирует сигналы с различными временными параметрами, задаваемыми программно.   
 С помощью ПТ процессор может формировать временные интервалы произвольной длительности, производить синхронизацию внешних устройств, организовывать счетчики событий, вести счет текущего времени и т. д.

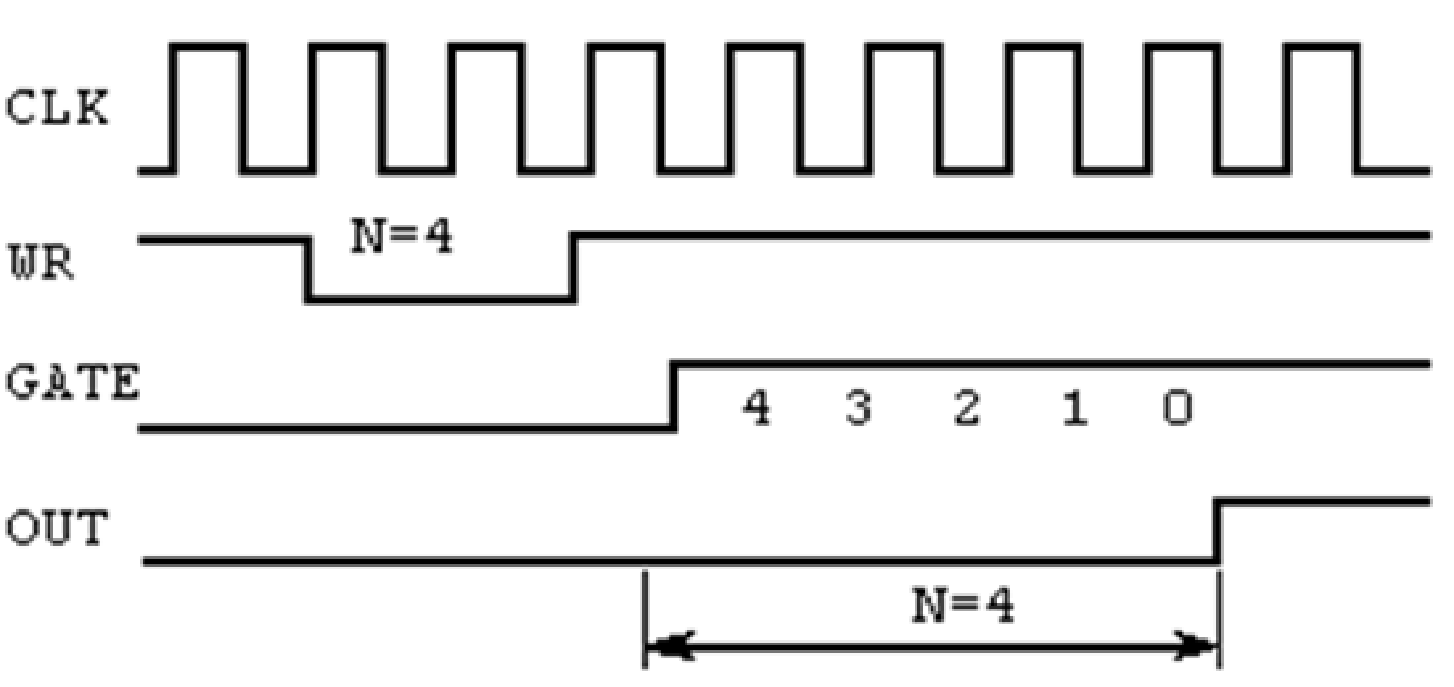
Условное графическое изображение таймера и подключение ПТ к шинам микропроцессора показано на рисунке 12.

  
Рисунок 12 – Условное графическое изображение таймера

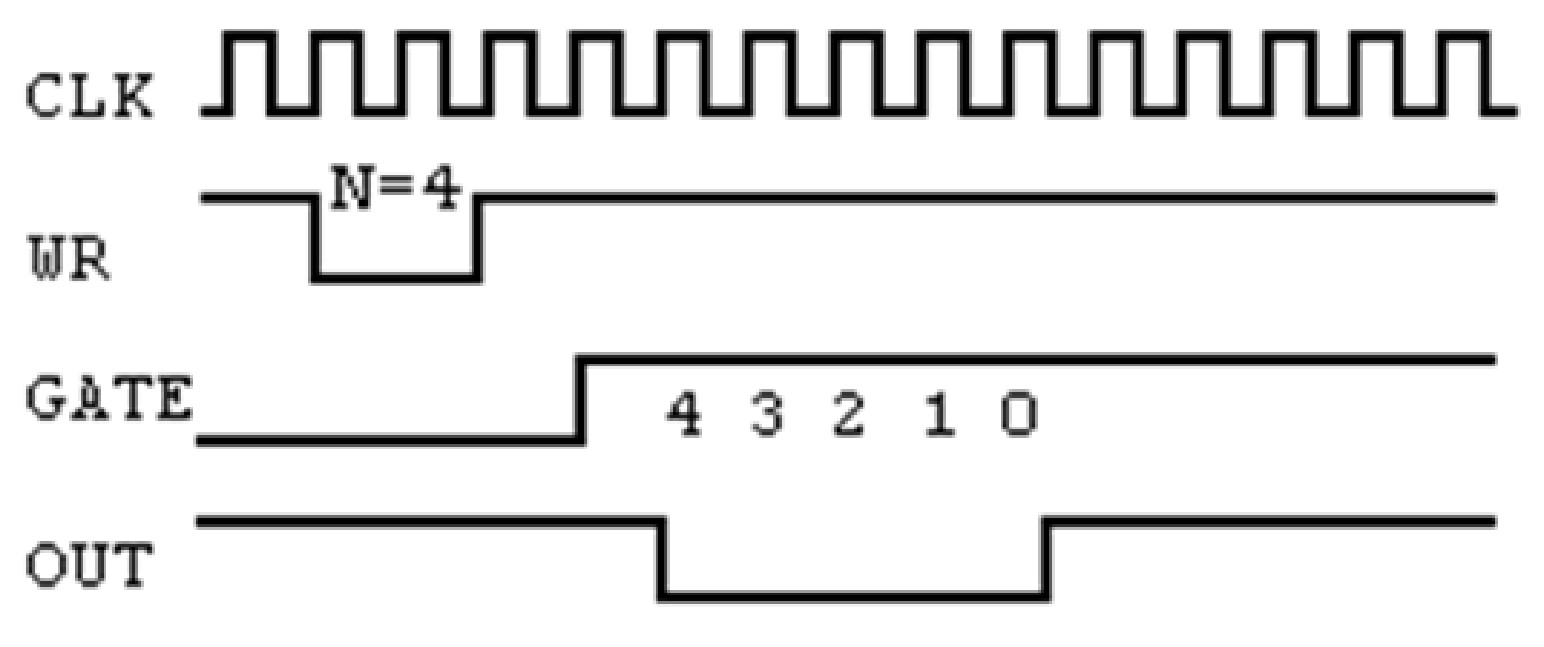
В состав ПТ входит буфер канала данных, логика чтения/записи, определяющая, к какому каналу обращается процессор, и трех независимых каналов. Каждый канал включает в себя 16-разрядный счетчик, регистр режима, схему управления и схему синхронизации.   
 В состав счетчика входят регистр хранения, буферный регистр и собственно счетчик. Регистр хранения содержит значение константы счета. В начале цикла работы канала константа счета из регистра хранения переписывается в счетчик, и затем по тактовым импульсам на входе CLK происходит декремент содержимого счетчика. Содержимое счетчика в любой момент времени может быть переписано в буферный регистр и прочитано процессором. В регистр режима записывается управляющее слово, определяющее режим работы канала. Схема управления синхронизирует работу отдельных схем в соответствии с запрограммированным режимом работы и работу канала с работой процессора. Схема синхронизации формирует серию внутренних импульсов с длительностью, определяемой внутренними времязадающими цепями, и периодом, равным периоду внешних тактовых импульсов.

**2.2.2. Режимы работы интервального таймера**

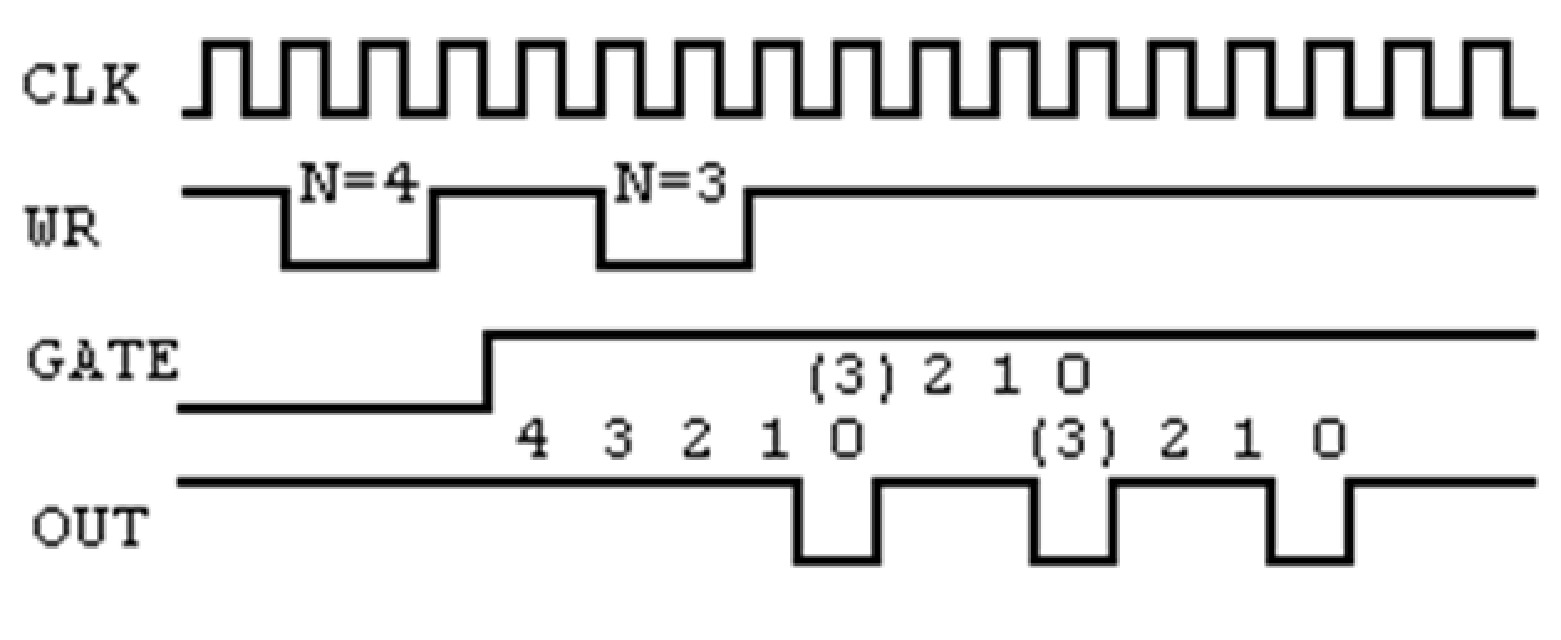
Существует шесть режимов работы каждого канала.   
 Режим 0: выдача сигнала прерывания по конечному числу (прерывание терминального счета). При работе в этом режиме на выходе канала появляется уровень "0" сразу же после установления режима работы. После загрузки числа в счетчик канала выход остается в "0" и счетчик начинает считать, если на входе разрешения установлен уровень "1". После того как достигается конечное число, на выходе устанавливается уровень "1" и остается до тех пор, пока канал не будет перезагружен режимом работы или новым числом. Диаграмма работы таймера в данном режиме представлена на рисунке 13.

  
Рисунок 13 – Диаграмма работы таймера в режиме 0.

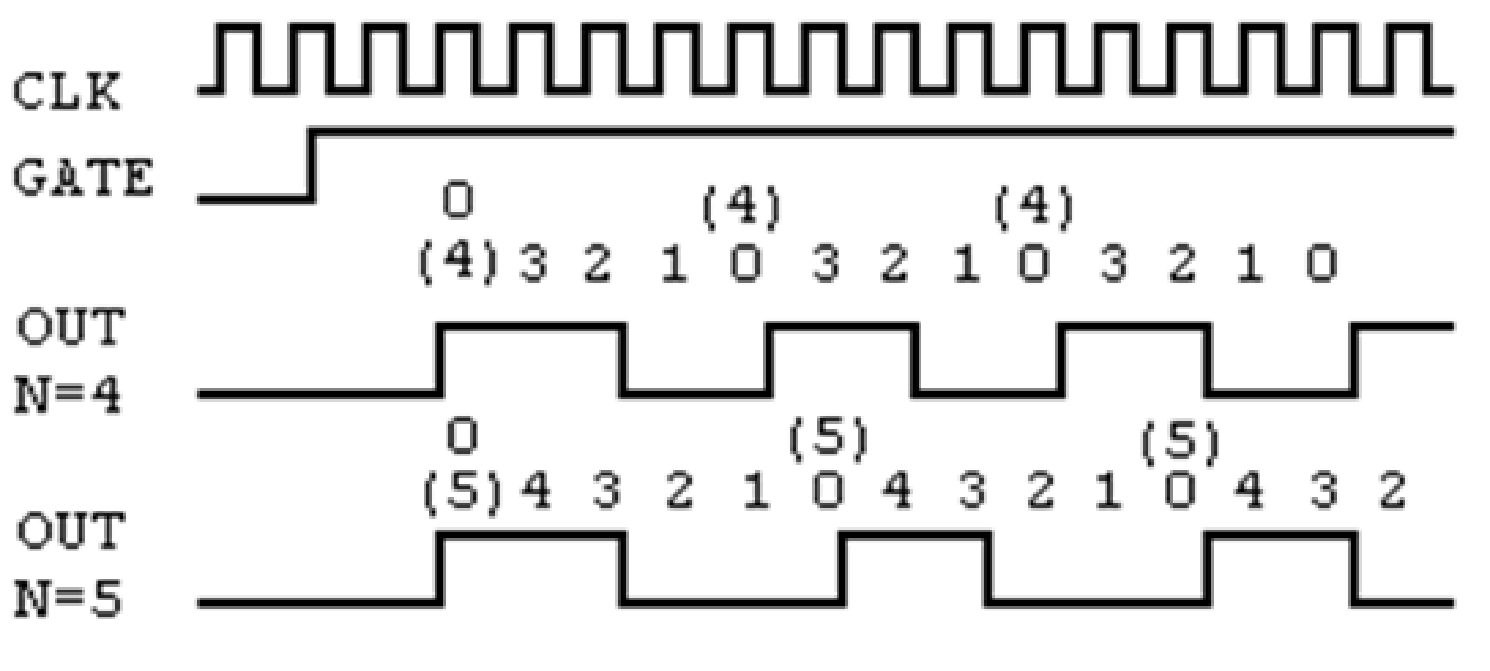
Режим 1: ждущий мультивибратор с программно-устанавливаемой длительностью сигнала. В этом режиме выход канала после загрузки числа в счетчик канала устанавливается в уровень "0" после первого тактового сигнала, следующего за передним фронтом на управляющем входе. Одновременно начинается счет, а при достижении конечного числа на выходе устанавливается уровень "1". Диаграмма работы таймера в данном режиме представлена на рисунке 14.

  
Рисунок 14 – Диаграмма работы таймера в режиме 1.

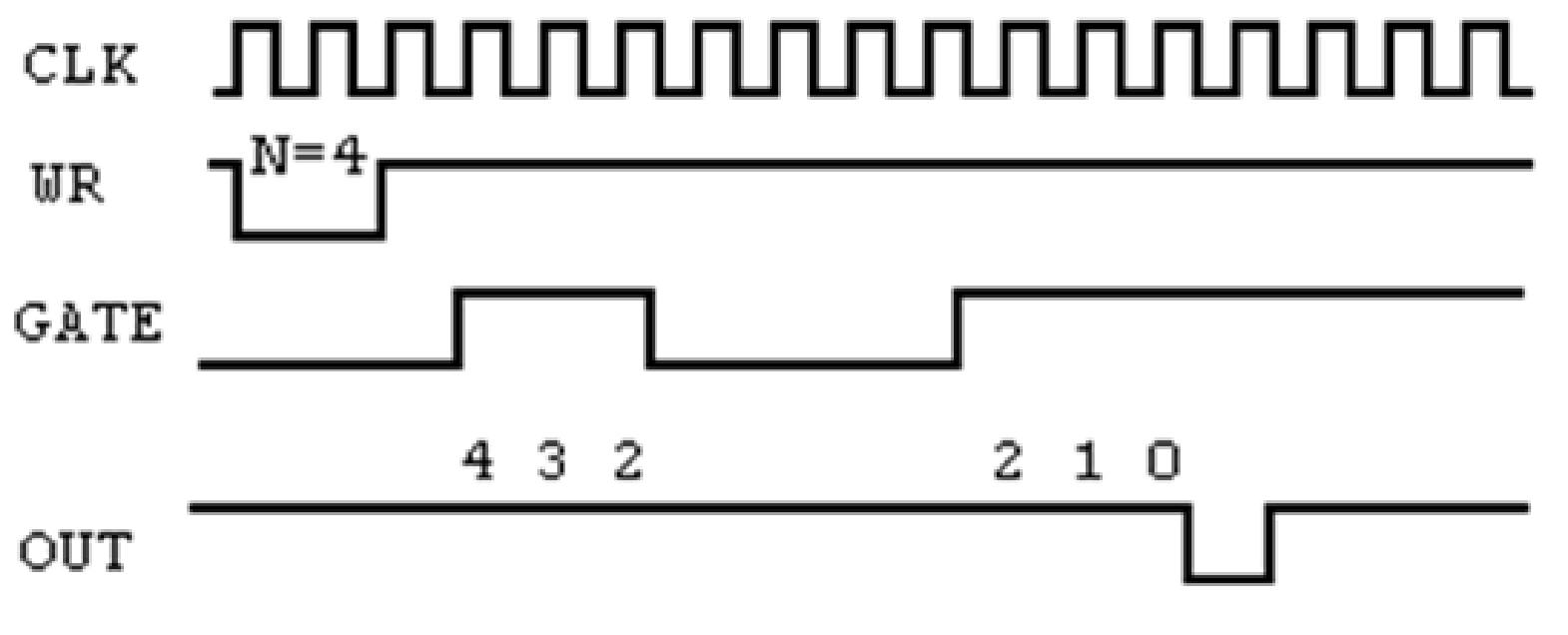
Режим 2: генератор тактовых сигналов. В этом режиме на выходе канала через число периодов тактовой частоты, записанное в счетчике канала, появляется уровень "0" длительностью в один период тактовой частоты. Диаграмма работы таймера в данном режиме представлена на рисунке 15.

  
Рисунок 15 – Диаграмма работы таймера в режиме 2.

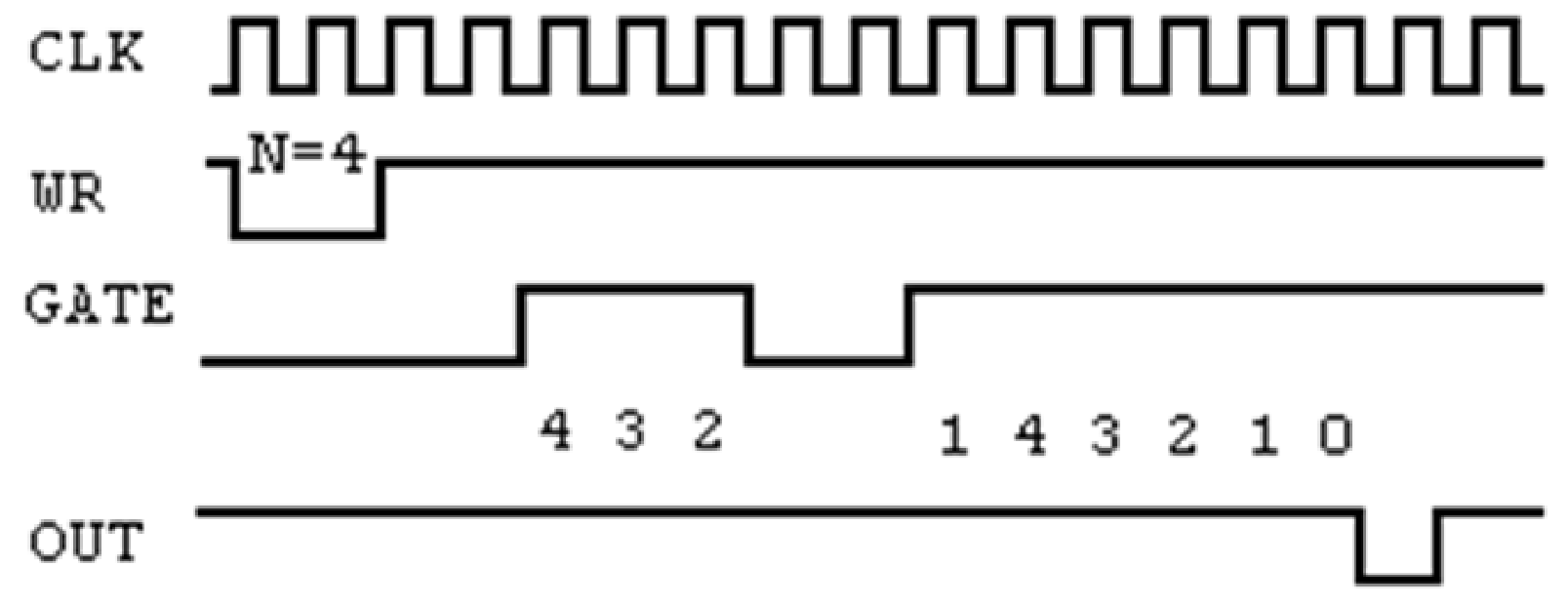
Режим 3: генератор прямоугольных сигналов. В этом режиме на выходе канала будет уровень "1" в течение первой половины интервала времени, определяемого числом в счетчике, и уровень "0" в течение второй половины. При нечетном числе длительность сигнала уровня "1" на один такт больше, чем для сигнала уровня "0". Диаграмма работы таймера в данном режиме представлена на рисунке 16.

  
Рисунок 16 – Диаграмма работы таймера в режиме 3.

Режим 4: программно-управляемый строб. После установки режима 4 на выходе канала появляется уровень "1". Когда число полностью загружено в счетчик канала и на управляющий вход подан уровень "1", начинается счет, и при достижении конечного числа на выходе появляется импульс уровня "0" длительностью в один период тактовой частоты. Диаграмма работы таймера в данном режиме представлена на рисунке 17.

  
Рисунок 17 – Диаграмма работы таймера в режиме 4.

Режим 5: схемотехнически управляемый строб. Работа канала в этом случае аналогична работе в режиме 4 с той лишь разницей, что счетчик канала после загрузки начинает счет только по переднему фронту на управляющем входе. Кроме того, если во время счета на управляющем входе снова появится передний фронт сигнала, то счет будет начат сначала. Диаграмма работы таймера в данном режиме представлена на рисунке 18.

  
Рисунок 18 – Диаграмма работы таймера в режиме 5.

**2.2.3. Настройка интервального таймера**

Каналы таймера полностью независимы друг от друга – каждый может иметь свой режим работы. Счетчик в каждом канале представляет собой 16-разрядный счетчик с предустановкой, работающий на вычитание в двоичном или двоично-десятичном коде. Таким образом, максимальное число при счете – 216 (при работе в двоичном коде) или 104 (при работе в двоично-десятичном коде) получается при загрузке всех нулей в счетчик канала.  
 При начальной настройке таймера все входы GATE устанавливаются в 0. После настройки в каждый канал надо записать делители и только после этого переключать входы GATE в 1.   
 Если таймер не может одним каналом посчитать длинный интервал времени, то каналы таймера можно соединить друг за другом.   
 Порядок программирования ПТ произвольный, т. е. можно сначала запрограммировать режимы работы всех каналов, а затем загрузить счетчики. Счетчик канала должен быть обязательно загружен именно тем количеством байтов, которое было запрограммировано в управляющем слове (значения разрядов D5 и D4). При загрузке содержимого счетчика двухбайтовым числом первым записывается младший байт, затем старший.

**2.2.4. Схема включения интервального таймера в микропроцессорную систему**

На рисунке 19 представлена схема включения интегрального таймера в микропроцессорную систему.

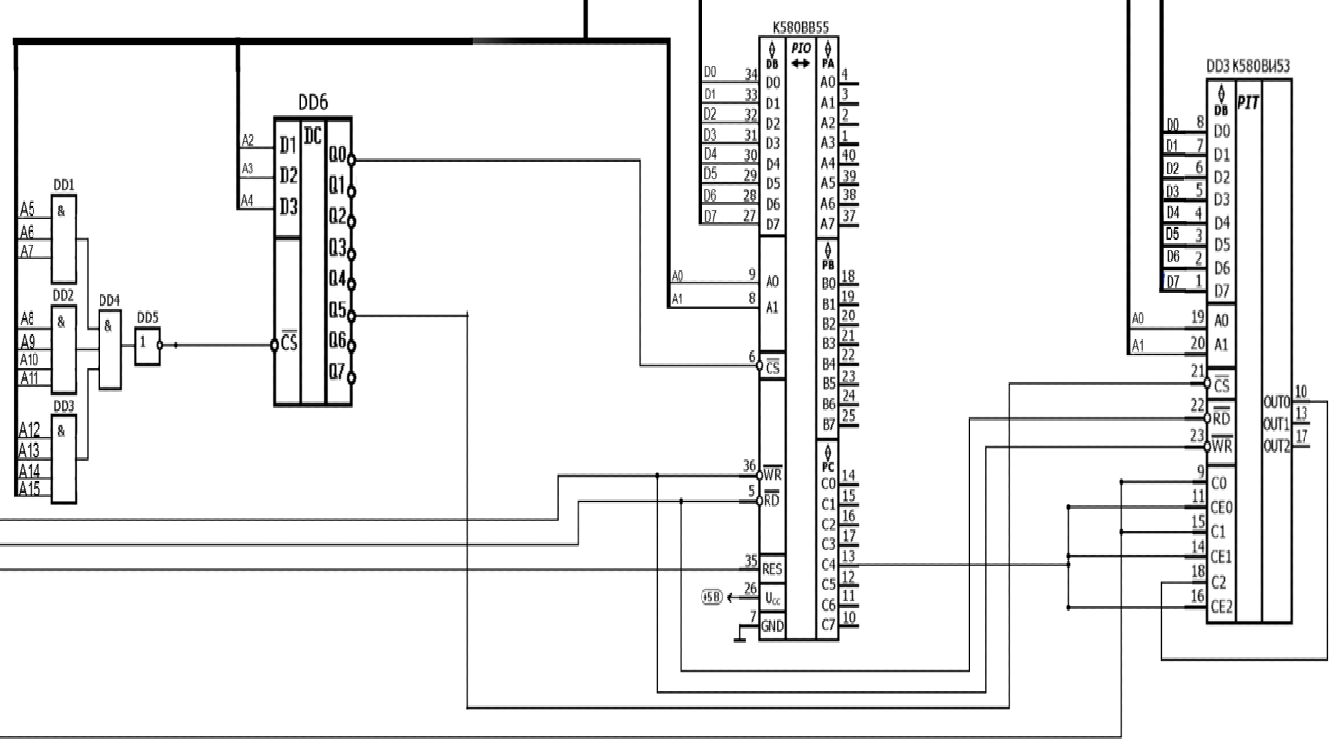


Рисунок 19 – Схема включения интервального таймера в микропроцессорную систему.

На рисунке 20 представлена полученная схема.

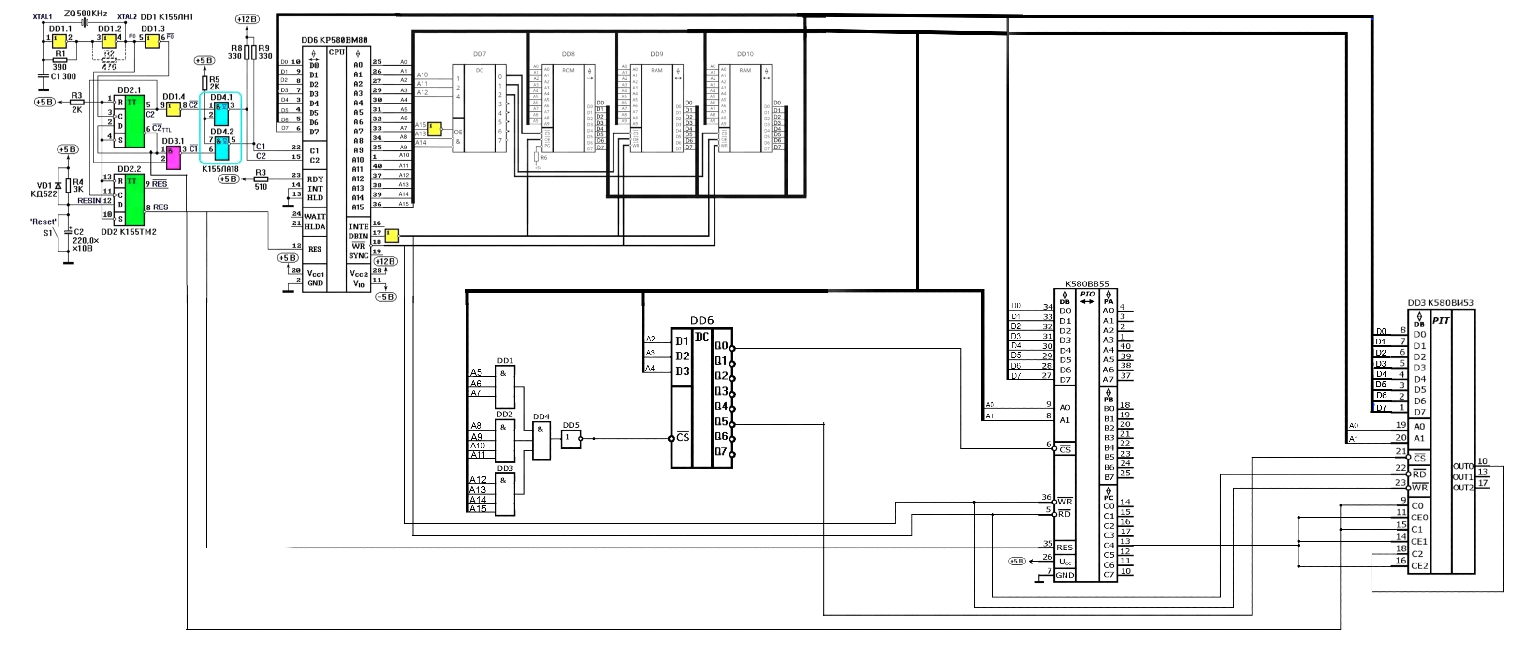


Рисунок 20 – Схема электрическая принципиальная разработанной МПС.

**3. Практическая часть  
3.1. Лабораторная работа №1.7**

**Задание:**

Реализовать программно линейный светодиодный индикатор с увеличением и уменьшением длины светящейся линии, используя параллельный перифе-рийный адаптер К580BB55A микропроцессорной системы.

К выводам каналов А и В ППА через буферный регистр формирователь

КР580ИР83 подключена линейка из шестнадцати светодиодов. Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - OFFE4H;

регистр В - OFFE5H;

регистр С - OFFE6H;

регистр управляющего слова - О7H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ПІПА:

режим 0 - простой ввод/вывод,

канал А - вывод,

канал В - вывод,

канал С - ввод.

Для выхода из программы использовать опрос разряда 1 канала В ППА по адресу OFFE1H. установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша "Shift") инициализирует безусловный переход в системный монитор по адресу OF800H.

**Решение**

На рисунке 21 изображена схема микропроцессорной системы в программном комплексе САПР.

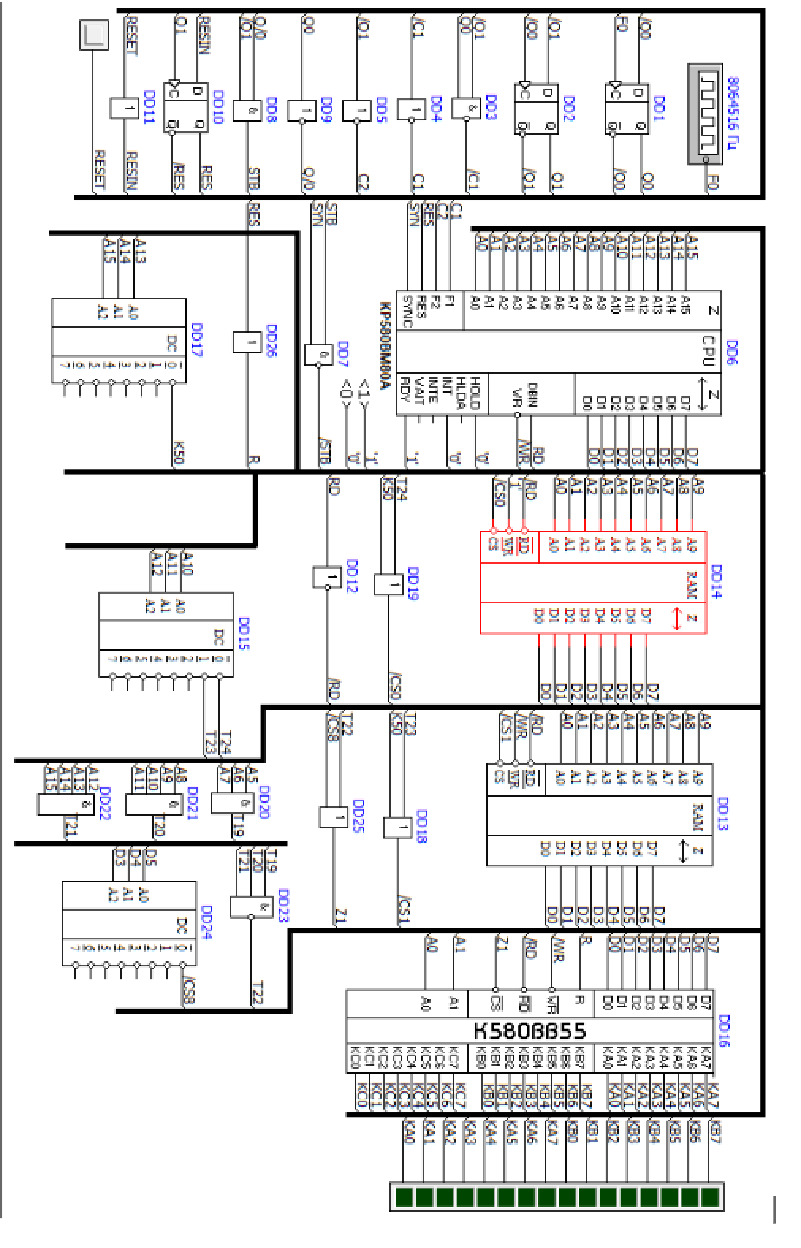


Рисунок 21 – схема микропроцессорной системы

**Код программы:**

ORG 0000H ;Начало программы в памяти по адресу 000H

:START

LXI SP, 800H ;Установка указателя стека на адрес 0800H

:INIT

MVI A,89H ;Настройка портов вывода А - Вывод, С - Ввод, В - Вывод

:WAITKEY

0 MVI E, 001H ;Инициализация счетчика порта А

MVI C, 000H ;Вспомогательный регистр

MVI B, 000H ;инициализация счетчика порта B

MVI D, 0FFH ;Максимальное значение для сравнения

MVI A, 001H ;Стартовый патерн для порта А

MVI L, 025H ;Инициализация счетчика задержки(25H = 37 циклов)

STA PORTOUT ;Запись начального значения порт А

:INCA

MOV A, D

CMP E ;Проверка достижений максимума

JZ INCB ;если да то переход к работе с портом В

MOV A, E

INR E ;Увеличение счетчика

ADD E ;Генерация паттерна

MVI L, 025H ;Установка времени задержки

CALL DELAY ;Вызов программы задержки

MOV E, A ;Сохранение нового паттерна

STA PORTOUT ;Обновление состояния порта A

JMP INCA

:INCB

MVI A, 0FFH

CMP B ;проверка достижений максимума

JZ DECB ;переход к фазе уменьшения

MOV A, B

INR B ;увеличение счетчика

ADD B ;генерация паттерна

MVI L, 025H ; Установка задержки

CALL DELAY

MOV B, A

STA PORTINP ;Обновление светодиодов

JMP INCB

:DECB

MVI A, 000H

CMP B ;Проверка на максимум

JZ DECA ;переход к уменьшению порта А

DCR B ;уменьшение счетчика

MOV A, B

RRC ;сдвиг паттерна вправо

MVI L, 025H

CALL DELAY

MOV B, A

STA PORTINP

JMP DECB

:DECA

MVI A, 000H

CMP E ;проверка на максимум

JZ INCA ;возвращение к увеличению

DCR E ;уменьшение счетчика

MOV A, E

RRC ;сдвиг паттерна вправо

MVI L, 025H

CALL DELAY

MOV E, A

STA PORTOUT

JMP DECA

:DELAY

DCR L ;уменьшение счетчика

JNZ DELAY ;повторять пока L != 0

RET ;возврат из подпрограммы

CONST PORTOUT 0FFE4H

CONST PORTINP 0FFE5H

**3.2. Лабораторная работа №1.6**

**Задание:**Реализовать программно линейный светодиодный индикатор с увеличением и уменьшением длины светящейся линии, используя параллельный перифе-рийный адаптер К580BB55A микропроцессорной системы.

К выводам каналов А и В ППА через буферный регистр формирователь

КР580ИР83 подключена линейка из шестнадцати светодиодов. Регистры ППА имеют следующие адреса в совмещенной карте памяти:

регистр А - OFFE4H;

регистр В - OFFE5H;

регистр С - OFFE6H;

регистр управляющего слова - ОFFE7H.

Доступное пользователю ОЗУ расположено по адресам 0000H÷7FFFH.

При выполнении задания реализовать следующую настройку ПІПА:

режим 0 - простой ввод/вывод,

канал А - вывод,

канал В - вывод,

канал С - ввод.

Для выхода из программы использовать опрос разряда 1 канала В ППА по адресу OFFE1H. установка которого в нулевое состояние через подключенную к нему замыкающую кнопку (клавиша "Shift") инициализирует безусловный переход в системный монитор по адресу OF800H.

**Решение**

На рисунке 22 изображена схема микропроцессорной системы в программном комплексе САПР.

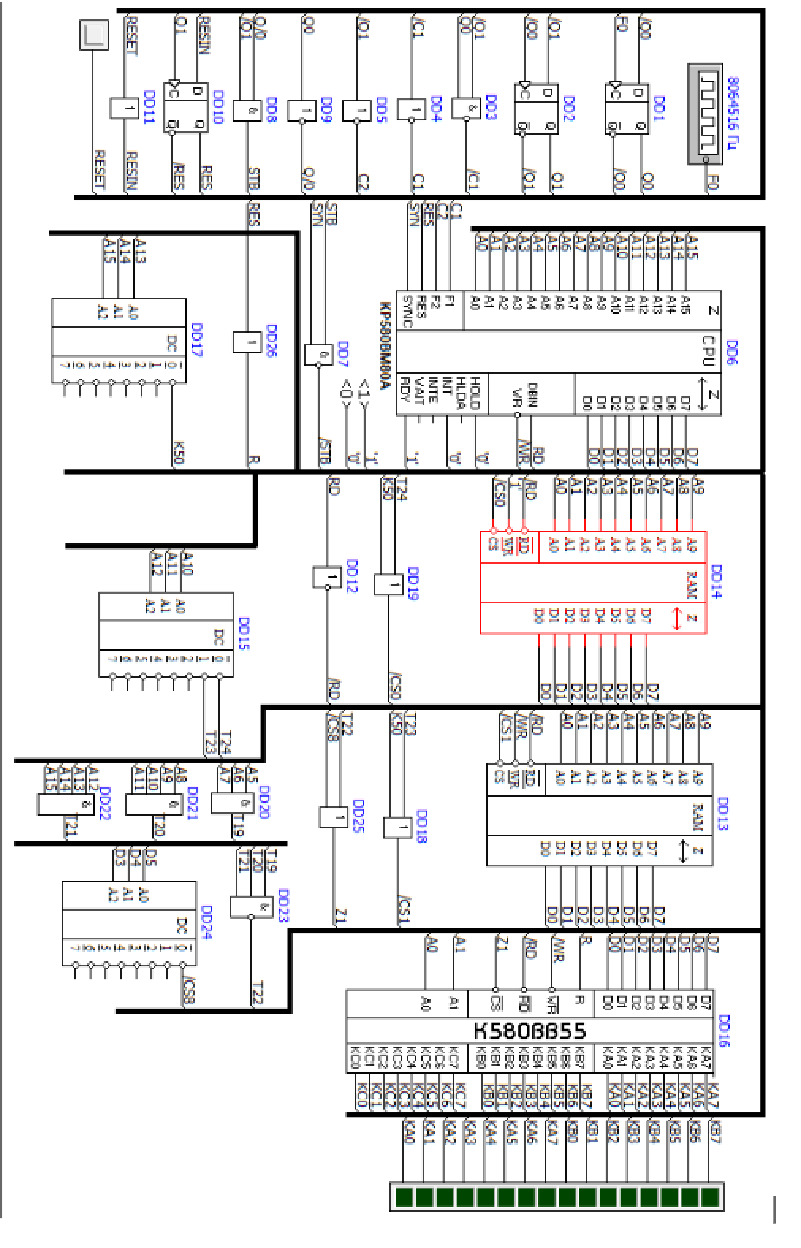


Рисунок 22 – схема микропроцессорной системы

**Решение:**  
ORG 0000H ; Начало программы по адресу 0000H

LXI SP, 7FFH ; Инициализация указателя стека

:START

MVI A, 89H ; Загрузка управляющего слова 89H в аккумулятор:

; - Режим 0

; - Порт A: вывод, Порт B: вывод, Порт C: вывод

STA 0FFE3H ; Загрузка управляющего слова в регистр ППА

:FIRST

MVI A, 00H ; Сброс аккумулятора

STA 0FFE1H ; Вывод в порт B

MVI A, 01H ; Загрузка начального значения

:LOOP1

STA 0FFE0H ; Вывод в порт A

RLC ; Циклический сдвиг влево

CALL DELAY

JC SECOND ; Если был перенос, переход к следующей фазе

JMP LOOP1 ; Повтор цикла

:SECOND

MVI A, 00H ; Сброс аккумулятора

STA 0FFE0H ; Выключение порта A

MVI A, 01H ; Повторная инициализация

:LOOP2

STA 0FFE1H ; Вывод в порт B

RLC ; Сдвиг влево

CALL DELAY

JC THIRD ; Переход при переносе

JMP LOOP2

:THIRD

MVI A, 00H ; Сброс

STA 0FFE1H ; Выключение порта B

MVI A, 80H ; Начальное значение

:LOOP3

STA 0FFE1H ; Вывод в порт B

RRC ; Циклический сдвиг вправо

CALL DELAY

JC FOURTH ; Переход при переносе

JMP LOOP3

:FOURTH

MVI A, 00H ; Сброс

STA 0FFE1H ; Выключение порта B

MVI A, 80H ; Начальное значение

:LOOP4

STA 0FFE0H ; Вывод в порт A

RRC ; Сдвиг вправо

CALL DELAY

JC FIRST ; Завершение цикла, возврат к первой фазе

JMP LOOP4

HLT ; Остановка

:DELAY

DCR D

JNZ DELAY

RET

**Заключение по практике.**

В ходе работы были приобретены навыки, умения и компетенции по созданию микропроцессорной системы на базе процессора КР580ВМ80А с применением системы команд на языке Ассемблера. Был реализован программно линейный светодиодный индикатор с увеличением и уменьшением длины светящейся линии, с использованием ППА К580ВВ55А.

**Заключение** В результате проделанной работы, мною были приобретены навыки и умения построения микропроцессорной системы с произвольным процессором и произвольным набором сопутствующих микросхем на основе микропроцессора КР580ВМ80. Получены навыки создания программного обеспечения для разработанной микропроцессорной системы с использованием системы команд на основе языка Ассемблер, а также получен опыт оптимизации, включая процессы отладки. Цель данной работы по дисциплине «Микропроцессорные системы» достигнута, задачи выполнены в полном объеме.