

## 3. ОРГАНИЗАЦИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

### 3.1. Организация памяти в микропроцессорных системах

#### 3.1.1. Назначение и виды памяти

Поскольку микропроцессор работает по программе, представляющей собой последовательность наборов двоичных кодов, то для хранения программы необходимо запоминающее устройство – память. В памяти хранятся также и обрабатываемые данные. Вся информация в памяти хранится в виде двоичных кодов определенной длины. Основой памяти является регистр требуемой разрядности - ячейка памяти. В настоящее время общепринята побайтовая организация памяти из восьмиразрядных ячеек, хранящих двоичный код в восемь бит или один байт. Информацию в ячейки памяти можно записывать и считывать. Считывание информации из ячейки памяти не нарушает содержимого последней. Каждая ячейка памяти состоит из элементов памяти, хранящих один разряд двоичного кода (рис. 3.1).

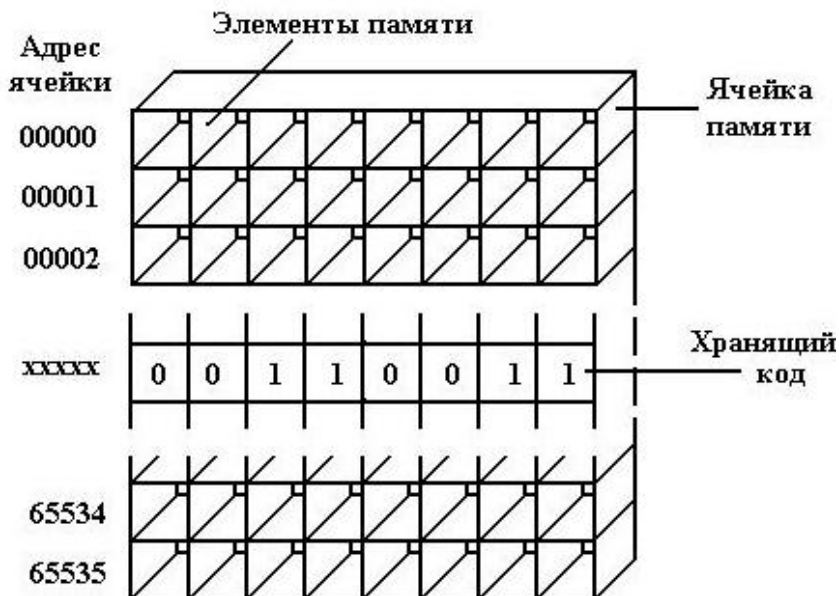


Рис. 3.1. Логическая организация памяти

Чтобы можно было выбрать нужную ячейку памяти для записи или считывания информации, каждая ячейка памяти снабжается адресом, представляющим собой двоичный код определенной разрядности, который подается на вход переключателя ячеек - селектора адреса памяти и обеспечивает подключение к шине данных памяти выбранной ячейки памяти. Для 8-разрядных микропроцессоров и микроЭВМ используются 16-разрядные двоичные адресные коды, что позволяет адресовать  $2^{16} = 65536$  ячеек памяти.

Разделение памяти по видам иллюстрируется на рис. 3.2. Регистры представляют собой вспомогательную память для временного хранения информации и имеют более простую адресацию, что повышает быстродействие записи информации в регистры и считывания ее. Память в виде



Рис. 3.2. Виды памяти ЭВМ

регистров входит в состав микропроцессора (ее мы рассмотрели выше).

Основная память управляется микропроцессором и предназначена для хранения выполняемой программы и данных. В процессе работы можно адресоваться к любой ячейке основной памяти. Разделение основной памяти на память данных и память программ обычно осуществляется программным путем, т.е. при записи программы выделяют одну часть памяти (адресуют ячейки) для хранения кодов команд программы, а для записи данных - другую часть памяти.

Внешняя память используется для долговременного хранения программ и данных с целью накопления. В ней информация хранится записанной на магнитные диски, магнитную ленту, лазерные диски, перфоленту и т.д. При работе с внешней памятью нет возможности адресоваться к содержимому отдельной ячейки, поскольку нет деления внешней памяти на ячейки. Для использования информации, хранящейся во внешней памяти, ее сначала необходимо переписать в основную память.

Основная память реализуется на быстродействующих элементах в виде запоминающих устройств (ЗУ). В микропроцессорных устройствах она, как правило, реализована на электронных элементах и оформлена в виде БИС. Различают оперативные запоминающие устройства (ОЗУ) и постоянные запоминающие устройства (ПЗУ). ОЗУ допускают двухсторонний оперативный обмен информацией между памятью и микропроцессором, т.е. чтение и запись информации. ПЗУ в процессе работы обеспечивают только чтение информации и служат для хранения постоянных программ, которые либо не меняются в процессе эксплуатации системы, либо меняются редко. Перезапись информации в ПЗУ или технически невозможна, или связана с необходимостью дополнительных мероприятий и производится вне вычислительной системы.

Запоминающие устройства могут быть статического типа и динамического. В ЗУ статического типа представление информации осуществляется потенциальными сигналами логического нуля и логической единицы, в памяти динамического типа сигналы представлены в виде импульсов и требуют постоянного обновления при считывании.

Оперативная память может быть энергозависимой и энергонезависимой. В энергонезависимой памяти информация сохраняется при отключении питания, в энергозависимой памяти информация теряется. С целью обеспечения энергонезависимости памяти для ее питания используются автономные батареи или аккумуляторы, которые автоматически подключаются к памяти в момент исчезновения напряжения питания.

Запоминающие устройства, реализованные на основе БИС, изготавливаются:

- в виде ПЗУ, программируемых на заводе-изготовителе (так называемые масочные ПЗУ);
- в виде ПЗУ, допускающих однократное программирование потребителем (ППЗУ);
- в виде ПЗУ, допускающих стирание записанной информации и многократную электрическую запись другой информации (РППЗУ);
- в виде статических и динамических ОЗУ.

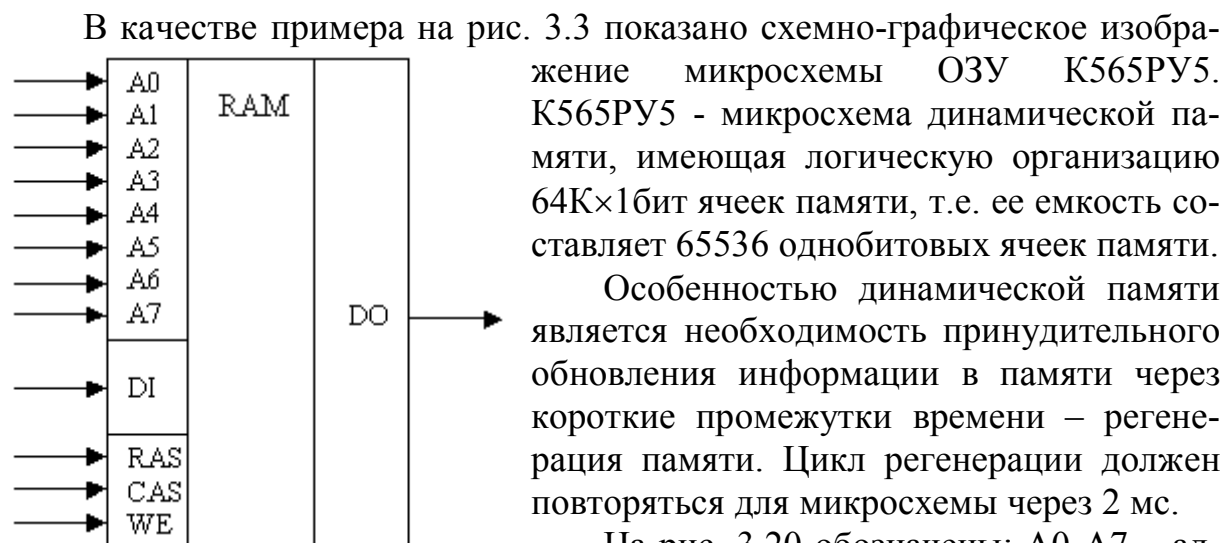


Рис. 3.3. Микросхема K565PY5

На рис. 3.20 обозначены: A0-A7 – адресные входы, DI – вход записи информации, DO – выход считывания информации, RAS – выбор микросхемы – выбор строки матрицы памяти, CAS – выбор столбца матрицы памяти, WE – сигнал управления записью информации.

В целях сокращения числа адресных выводов все адресное пространство микросхемы представлено в виде матрицы 8×8 ячеек памяти. При выборе ячейки памяти сначала на адресные входы подается 8-битный адрес строки матрицы. Этот адрес запоминается во внутреннем регистре адреса.

Затем на эти же входы подается адрес столбца матрицы. Подача адреса столбца стробируется сигналом CAS. В результате формируется полный 16-разрядный адрес ячейки памяти.

### 3.1.2. Взаимодействие памяти с микропроцессором

Запоминающее устройство подключается к микропроцессору с помощью системных шин. При подключении ОЗУ предусматривается запись и чтение информации, при подключении ПЗУ - только чтение информации. На рис. 3.4 показано подключение запоминающего устройства к микропроцессору. По шине адреса микропроцессор задает код адреса требуемой ячейки памяти, и селектор адреса ЗУ подключает эту ячейку к буферу данных. После подготовки ЗУ к обмену информации оно подает сигнал готовности Г микропроцессору. Микропроцессор подачей логических сигналов ВД или П на соответствующие линии шины управления определяет направление передачи информации, после чего происходит требуемая передача информации (код с шины данных переписывается в ячейку памяти при записи или с выходов ячейки памяти через буферный регистр поступает на линии шины данных при чтении).

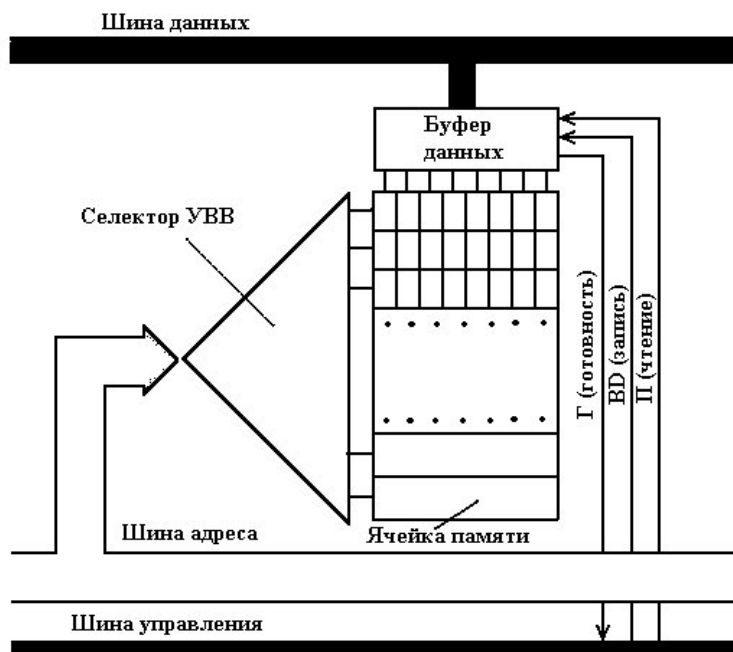


Рис. 3.4. Подключение памяти

чаемых блоков памяти – страниц. В последнем случае для доступа к ячейке памяти необходимо сначала адресовать страницу памяти, а затем – ячейку памяти, указав её адрес внутри страницы.

В качестве примера на рис. 3.5 показано подключение к системным шинам 8-разрядного микропроцессора микросхемы ОЗУ типа К537РУ10.

Микросхема представляет собой оперативную память статического типа и имеет 2048 (2 К) 8-разрядных ячеек памяти. Емкость памяти составляет 2 Кбайт

Входы-выходы данных D0...D7 соединены с линиями шины данных ШД микропроцессора. Для адресации ячеек памяти служат адресные входы микросхемы A0...A10. Эти входы соединены с соответствующими линиями адресной шины. Используемый для адресации 11-разрядный двоичный код позволяет адресовать 2048 ячеек памяти.

Входы-выходы данных D0...D7 соединены с линиями шины данных ШД микропроцессора. Для адресации ячеек памяти служат адресные входы микросхемы A0...A10. Эти входы соединены с соответствующими линиями адресной шины. Используемый для адресации 11-разрядный двоичный код позволяет адресовать 2048 ячеек памяти.

Входы-выходы данных D0...D7 соединены с линиями шины данных ШД микропроцессора. Для адресации ячеек памяти служат адресные входы микросхемы A0...A10. Эти входы соединены с соответствующими линиями адресной шины. Используемый для адресации 11-разрядный двоичный код позволяет адресовать 2048 ячеек памяти.

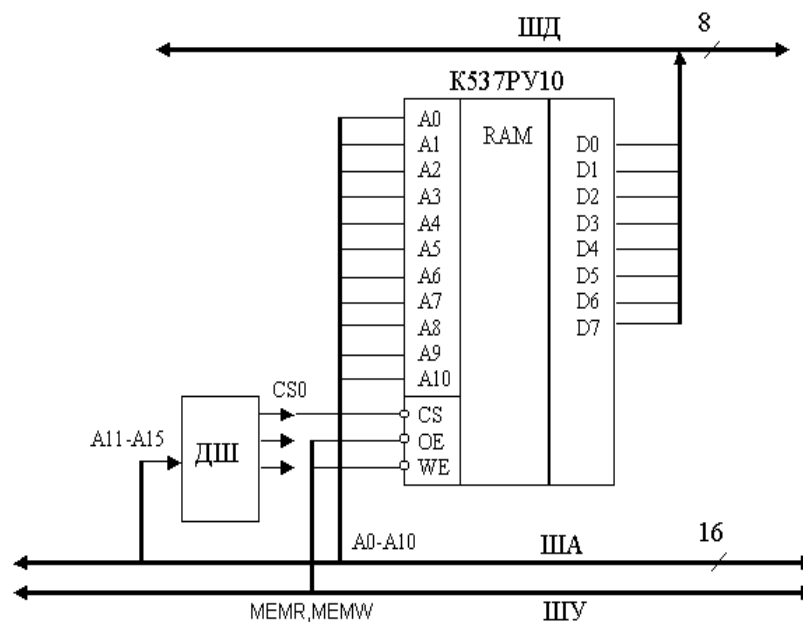


Рис. 3.5. ОЗУ на микросхеме K537PY10

Поскольку адресная шина микропроцессора имеет 16 разрядов, то старшие адреса A11...A15 используются для адресации нескольких микросхем памяти. Для этого введен дешифратор ДШ старших разрядов адреса. В принципе, используя эти разряды (5 бит), можно адресовать 32 микросхемы памяти, обеспечив полный объем памяти 64 Кбайта. В зависимости

от кода на линиях адреса A11...A15 на одном из выходов дешифратора создается сигнал разрешения работы микросхемы CS0, CS1, CS3 и т.д.

Сигнал разрешения работы микросхемы поступает на вход выбора микросхемы CS. Активный уровень этого сигнала разрешает работу микросхемы. При пассивном уровне сигнала CS выходы микросхемы переходят в высокоимпедансное состояние, и микросхема отключается от шин микропроцессора.

Направление передачи информации (запись в ячейку памяти или чтение из ячейки памяти) определяется сигналами на входах управления OE и WE. Для управления этими входами использованы сигналы шины управления: MEMR – чтение из памяти и MEMW – запись в память.

При создании памяти с определенной разрядностью ячейки памяти могут использоваться микросхемы памяти с разной организацией, что приводит к необходимости параллельного использования нескольких микросхем. Пример построения памяти емкостью 64 Кбайт на микросхемах ОЗУ типа K565PY5, имеющих организацию 64 К×1 бит, показан на рис. 3.6.

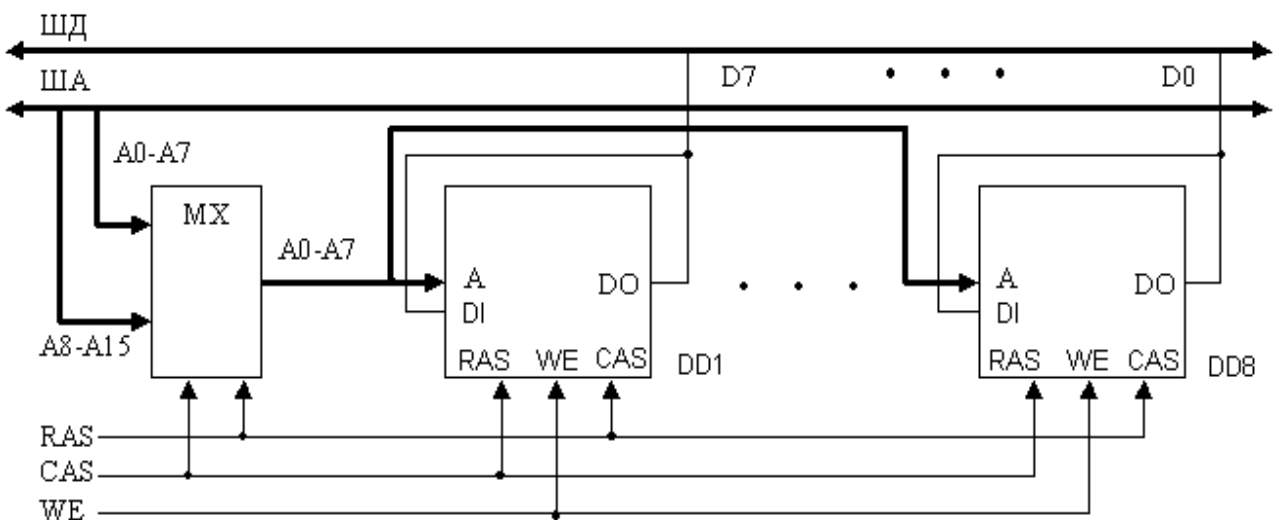


Рис. 3.6. ОЗУ на микросхемах K565PY5

Для получения 8-битной ячейки памяти параллельно работают 8 микросхем DD1...DD8. Входы записи DI и чтения DO каждой микросхемы объединены и присоединены к соответствующей линии шины данных ШД D7...D0 микропроцессора. Направление передачи данных определяется сигналом WE, формируемым схемой управления из системных сигналов MEMR и MEMW. Поскольку микросхема K565PY5 для адресации ячейки памяти нуждается в последовательной адресации столбца и строки матрицы памяти, то в схеме предусмотрен мультиплексор MX, который последовательно передает на адресные входы A0...A7 микросхемы либо младший A0...A7, либо старший байт A8...A15 шины адреса ША. Управляется мультиплексор сигналами RAS и CAS, формируемыми схемой управления.

Эти сигналы одновременно поступают на входы выбора столбца и строки микросхемы памяти RAS и CAS.

Пример страничной организации постоянной памяти на микросхемах ПЗУ типа К573РФ5 показан на рис. 3.7. Микросхема имеет организацию 2 К×1 байт и емкость 2 Кбайт. Младшие линии адресов А0...А10 шины адреса осуществляют выбор ячейки памяти в пределах 2 Кбайт микросхемы. Каждая микросхема может рассматриваться как страница памяти.

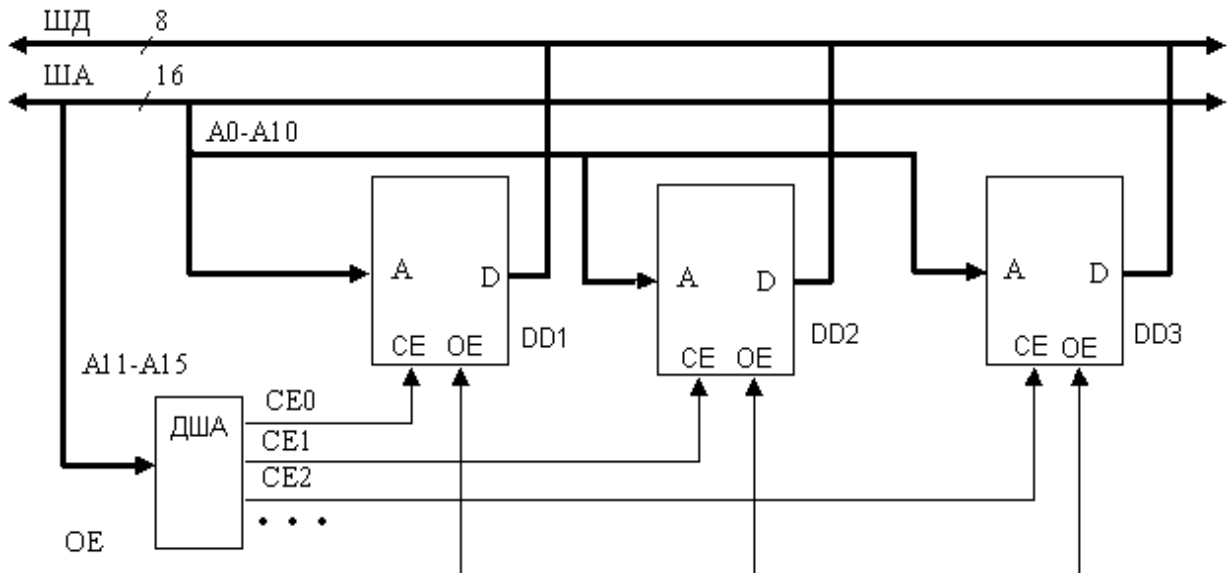


Рис. 3.7. ПЗУ на микросхемах К573РФ5

Старшие биты А11...А15 шины адреса подаются на дешифратор ДША, который в общем случае осуществляет преобразование 5-битного кода в 32 сигнала выбора микросхемы. В каждый момент времени активен один из выходов CE0, CE1, CE2,... дешифратора. Сигнал дешифратора поступает на вход CE микросхемы памяти DD1, DD2, DD3,..., разрешая её работу. Таким образом, к шине данных поочередно можно подключать 32 микросхемы памяти и реализовать ПЗУ на 2Кбайта×32=64Кбайт. Управляет чтением информации сигнал OE.

### 3.2. Устройства ввода-вывода информации в МПС

Для ввода данных в микропроцессор и вывода результатов обработки информации используются модули ввода-вывода. Основой модуля ввода-вывода является регистр, разрядность которого соответствует разрядности шины данных. Такой регистр называют портом. Если входы регистра подключены к линиям шины данных, а выходы - к внешнему устройству, тогда регистр служит для вывода данных и называется портом вывода. Для ввода информации используются порты ввода.

В каждый момент времени обмен данными возможен только с одним портом ввода или вывода. Для выбора нужного порта служит селектор портов, который в соответствии с кодом адреса, поступившего по адресной шине, выбирает нужный порт и подключает его к шине данных. Естественно, что каждый порт должен иметь свой адрес в виде двоичного кода адреса. Поскольку портов ввода-вывода обычно меньше, чем ячеек памяти, то используются адреса меньшей разрядности, что несколько упрощает программирование и сокращает объем программы микропроцессора. Для 8-разрядного микропроцессора применяется адрес портов длиной в один байт, что позволяет адресовать  $2^8 = 256$  портов ввода-вывода.

Модули ввода-вывода подключаются к микропроцессору с помощью системных шин. Схема соединения портов ввода-вывода с микропроцессором показана на рис. 3.8. В рассматриваемом примере модуль ввода-вывода содержит два порта ввода и один порт вывода. При вводе информации двоичный код данных от внешнего устройства должен поступить в регистр порта ввода и временно в нем храниться.

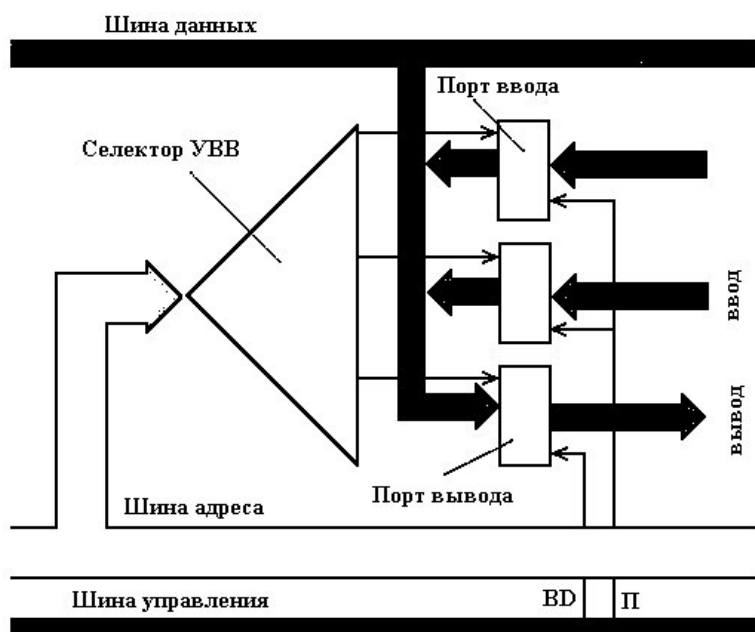


Рис. 3.8. Порты ввода-вывода

Микропроцессор при исполнении команды ввода данных от внешнего устройства выдает на шине адреса код нужного порта ввода, селектор устройства ввода-вывода УВВ подключает этот порт к шине данных. По сигналу чтения "П", передаваемому микропроцессором по соответствующей линии шины

управления, происходит передача кода данных с выходов регистра порта ввода на линии шины данных и далее в микропроцессор. Вывод данных производится аналогично через порт вывода с использованием сигнала управления "ВД". Обращение микропроцессора к портам ввода-вывода осуществляется по определенным командам, входящим в систему команд микропроцессора.

Для реализации портов ввода-вывода предназначен широкий ассортимент специальных микросхем различных серий. Интерфейсные микро-



схемы включены в микропроцессорные комплекты K580, K1810, K1801 и др. В качестве примера рассмотрим программируемый параллельный адаптер KP580BB55, предназначенный для реализации 8-разрядных портов ввода-вывода.

### 3.2.1. Контроллер параллельного обмена K580BB55

В настоящее время во многих микропроцессорных комплектах выпускают специальные интерфейсные БИС, существенно расширяющие (по сравнению с использованием регистров) возможности разработчиков при организации параллельного обмена в МПС. Наиболее популярной БИС этого класса можно считать контроллер K580BB55.

БИС. BB55 представляет собой трехканальный байтовый интерфейс и позволяет организовать обмен байтами с периферийным оборудованием в различных режимах. Внутренняя структура. BB55 показана на рис. 3.9., а подключение его к системной шине МПС – на рис 3.10.

Контроллер параллельного обмена K580BB55 (далее - "контроллер") включает в себя оборудование трех 8-разрядных каналов ввода/вывода (регистр и буферную схему), буфер шины данных, 8-разрядный регистр управления Y и блок управления.

Контроллер подключается к системной шине МПС, причем адресация его внутренних объектов осуществляется (младшими) разрядами шины адреса следующим образом:

Таблица 3.1

A <sub>1</sub>	A <sub>0</sub>	Адресуемый объект	Примечание
0	0	Канал А	
0	1	Канал В	
1	0	Канал С	
1	1	Регистр управления	Только запись

В МПС контроллер размещают, как правило, в пространстве адресов ввода/вывода. Поэтому в качестве стробов чтения и записи используются сигналы RDIO, WRIO, а для селекции контроллера дешифрируются старшие разряды адреса.

При подаче сигнала SR все регистры контроллера, в том числе и регистр управления Y, устанавливаются в состояние 00h.

В каждый момент времени процессор может поддерживать связь только с одним каналом, в зависимости от состояния линий A[1:0] (см. табл. 6.1). Функцию каждого канала и режим его работы определяет состояние управляющего слова Y, помещенного в регистр управления.

Каналы контроллера могут работать в одном из трех режимов:

- Режим "0" - синхронный однонаправленный ввод/вывод;
- Режим "1" - асинхронный однонаправленный ввод/вывод;

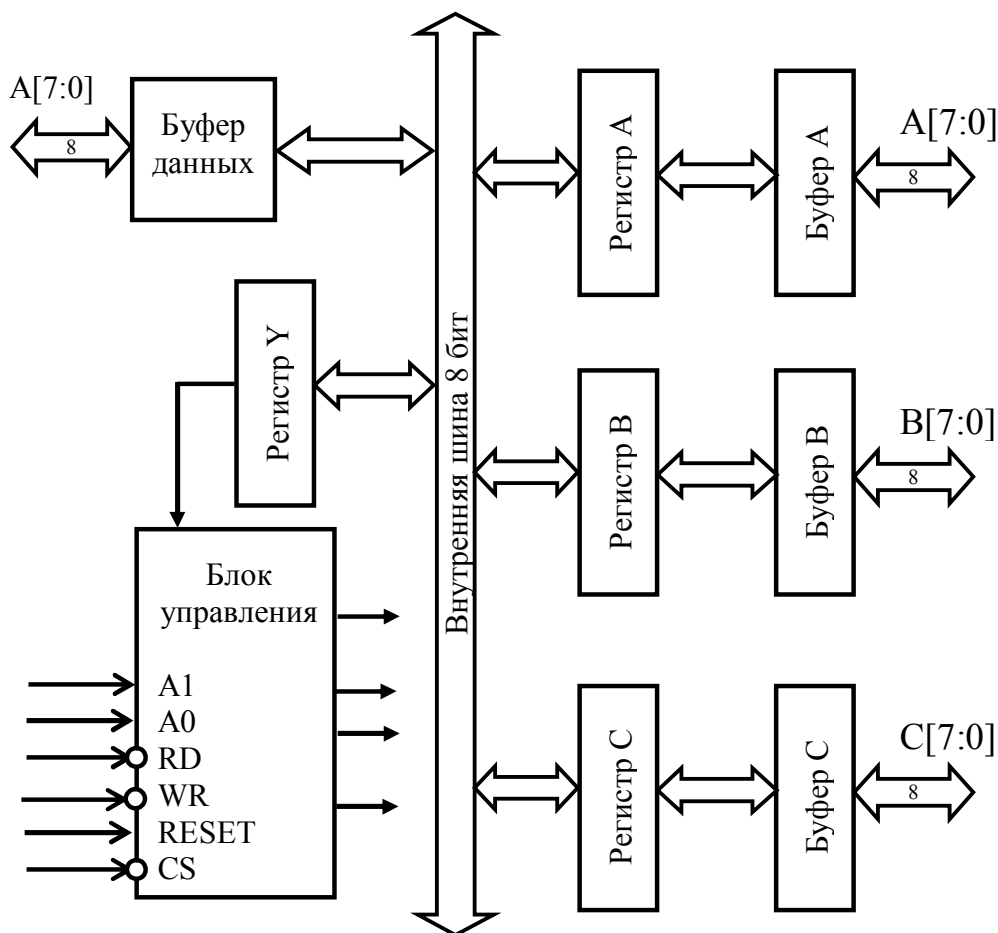


Рис.3.9. Внутренняя структура контроллера K580BB55

- Режим "2" - асинхронный двунаправленный ввод/вывод.

Кроме трех перечисленных, существует еще режим сброса-установки произвольного разряда канала С.

**В режиме "0"** контроллер может работать как четыре порта ввода/вывода: A[7:0], B[7:0], C[7:4], C[3:0], причем каждый порт может быть независимо запрограммирован на ввод или на вывод. При этом к порту, определенному как выходной, нельзя обращаться по чтению (будет прочитан "обрыв" - FFh), а на входной порт нельзя выводить информацию.

В стробируемом однонаправленном **режиме "1"** могут работать только каналы А и В, причем соответствующие линии (см. ниже) канала С придаются каналам А и В для передачи управляющих сигналов. Как и в режиме "0", каналы А и В программируются на ввод или вывод (независимо).

**В режиме "2"** может работать только канал А, к которому в этом случае можно обращаться как по записи, так и по чтению (двунаправленный стробируемый обмен).

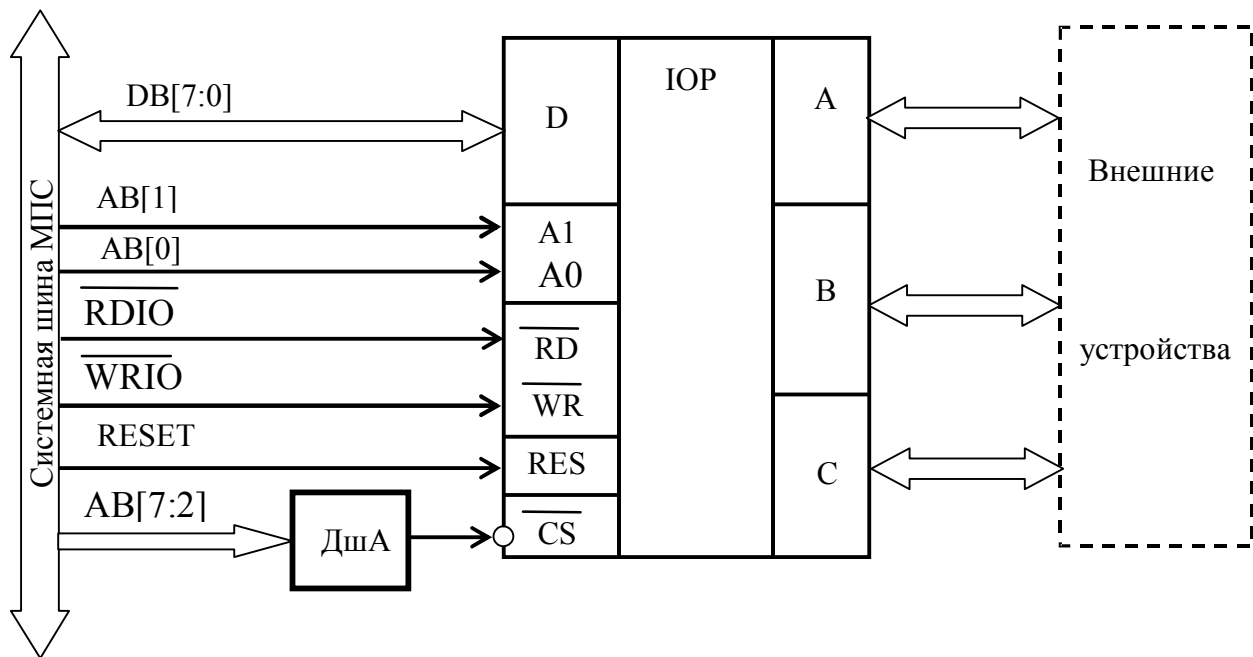


Рис. 3.10. Подключение контроллера BB55 с системной шине

Выбор режимов каналов и направления передачи данных в них осуществляется загрузкой в регистр Y управляющего слова, формат которого представлен на Рис.3.11.

1	D6	D5	D4	D3	D2	D1	D0	
	Режим A		На-	Направ	Режим	На-	Направ-	Направление: 1 - ввод 0 - вывод
	00 - "0"		прав-	ление	B	прав-	ление C[3:0]	
	01 - "1"		ление		0-"0"	ление		
	1x - "2"		A	C[7:4]	1-"1"	B		

Рис.3.11. Формат управляющего слова. BB55

Загрузка управляющего слова осуществляется путем записи его в регистр Y по адресу A[1:0] = 11, причем в старшем (D7) разряде такого слова должна стоять логическая единица.

Если при записи по адресу 11 D7 = 0, то такое слово не будет воспринято, как управляющее и не будет помещено в регистр Y. Такие действия обеспечивают установку или сброс одного разряда регистра C, причем номер изменяемого разряда записывается в разрядах D[3:1], а значение записываемого бита - в D0. Состояние разрядов D[6:4] - безразлично (Рис.3.12).

7	6	5	4	3	2	1	0
0	x	x	x	Номер разряда C		0/1	

В разряде 0 - значение устанавливаемого бита порта C

Рис.3.12. Управление битами канала C

Рассмотрим подробнее работу контроллера в различных режимах.

**Режим "0"**. При работе в этом режиме на вывод информация, поступающая с системной шины DB, запоминается в буферном регистре канала, адресуемого A[1:0] и сразу же передается на его выход. При вводе информации данные с линий соответствующего канала, минуя регистр, поступают через буфер данных на системную шину данных DB.

В режиме "0" канал C может быть разбит на два подканала, один из которых запрограммирован на ввод, а другой - на вывод. Обращение к обоим подканалам осуществляется по адресу A[1:0] = 10, причем в цикле ввода на DB подключаются только 4 входные линии, на остальных четырех - код 1111. При выводе информация с DB попадает только в 4 разряда регистра канала C, объявленные как выходные.

**Режим 1** обеспечивает однонаправленную асинхронную передачу информации между МП и ВУ. При этом каналы A и B используются как регистры данных, а канал C - для приема и формирования сигналов, сопровождающих асинхронный обмен, причем каждый разряд канала C имеет строго определенное функциональное назначение. В Таблица 3.2 показано использование линий канала C для передачи управляющих сигналов в режимах "1" и "2".

Таблица 3.2

Разряд канала C	Режим 1		Режим 2
	Ввод	Вывод	
0	ГТВ В	ГТВ В	-
1	ППР В	КГТ В\	-
2	СТР В\	ППРД В\	-
3	ГТВ А	ГТВ А	ГТВ А
4	СТР А\	-	СТР А\
5	ППР А	-	ППР А
6	-	ППРД А\	ППРД А\
7	-	КГТ А\	КГТ А\

В Таблица 3.2 использованы следующие обозначения:

СТР - строб записи в регистр канала (вход);

ППР - подтверждение приема (выход);

ГТВ - готовность (выход), может использоваться в качестве запроса на прерывание;

ППРД - подтверждение передачи (вход);

КГТ - канал готов к обмену (выход).

Символ "\" после имени сигнала обозначает, как и ранее, что активный уровень этого сигнала - низкий.

ВУ, будучи готовым выдавать информацию в контроллер, выставляет стробирующий сигнал, поступающий на вход СТР (С2 - для канала A, С4 - для B). По спаду строба байт данных с входных линий A или B "защелкивается" в регистр соответствующего канала. При этом контроллер форми-

рует управляющий сигнал ППР (подтверждение приема), который высоким уровнем запрещает ВУ вырабатывать новый строб.

Информация, попавшая в регистр канала, должна быть передана в МП. Контроллер вырабатывает по фронту СТР при ППР = 1 сигнал ГТВ, который поступает в качестве запроса на прерывание в подсистему прерываний и инициирует процедуру чтения байта из канала. Сигнал ГТВ формируется контроллером только при условии установления в "1" внутреннего триггера разрешения прерывания по каналу (разряды 2 и 4 регистра порта С для каналов А и В соответственно). Эти триггеры могут быть установлены и сброшены программно с помощью команд установки/сброса разряда порта С.

МП может не сразу удовлетворить запрос контроллера, т.к. занят обслуживанием более приоритетного запроса. В этом случае контроллер поддерживает активный уровень на выходе ГТВ и высокий уровень на ППР, запрещая выработку нового строга. Промежуток времени  $t_1$  (**Ошибка! Источник ссылки не найден.**) может быть произвольно большим и характеризует время ожидания реакции МП.

При появлении сигнала RD\ по соответствующему адресу содержимое регистра канала передается на шину данных DB и далее - в МП. По спаду RD\ снимается запрос ГТВ, а по фронту RD\ снимается ППР и ВУ получает возможность формировать новый строб записи данных. Промежуток времени  $t_2$  характеризует быстродействие ВУ и может, как и  $t_1$ , иметь произвольную длительность.

Таким образом, в описанном режиме осуществляется асинхронный ввод информации в МП с анализом готовности вводимой информации и исключением потери информации (в контроллере). Подсистема прерываний может быть выключена (сброшены триггеры разрешения прерывания или сигнал ГТВ не заведен в систему), однако МП может осуществлять программный опрос состояния каналов, считывая значение регистра порта С и анализируя значения разрядов готовности (С3 и С0 для каналов А и В соответственно). Модификация управляющих сигналов на линиях порта С может осуществляться программно с помощью соответствующих команд МП и аппаратно со стороны контроллера.

По стробу WR\ производится запись байта с системной шины данных в регистр канала. Одновременно снимается сигнал запроса на прерывание ГТВ. По фронту WR\ контроллер посылает на ВУ сигнал КГТ\, подтверждая, что данные для ВУ записаны в регистр и присутствуют на выводах канала. После получения КГТ ВУ начинает прием данных (отрезок времени  $t_2$ ), а закончив прием - формирует сигнал подтверждения передачи ППРД\ . Получив ППРД\ , контроллер формирует для МП сигнал ГТВ, сообщая, что можно выводить следующий байт данных. Время ожидания реакции МП - отрезок  $t_1$ . Программная установка/сброс триггеров разреше-

ния формирования сигналов ГТВ осуществляется с помощью команд переключения разрядов С6 и С2 - соответственно для каналов А и В.

Как и в случае режима ВВОД возможна реализация программно-управляемой процедуры асинхронного обмена.

В режиме "1", как и в режиме "0", недопустимо обращение по чтению к каналу, объявленному выходным и по записи к входному каналу. Для организации двунаправленного обмена предназначен режим "2", в котором может работать только канал А, которому для передачи управляющих сигналов придается 5 линий порта С (см. Таблица 3.). Канал В может при этом работать в режиме "1" - совместно с оставшимися линиями порта С, или в режиме "0", тогда С[2:0] будут работать тоже в "0"-режиме.

Направление передачи по каналам В и С определяется разрядами управляющего слова Y.

Процедуры ВВОД и ВЫВОД в режиме "2" осуществляются аналогично соответствующим процедурам в режиме "1".

### 3.2.2. Последовательный обмен данными в МПС

При организации последовательного обмена ключевыми могут считаться две проблемы:

- 1) синхронизацию битов передатчика и приемника;
- 2) фиксацию начала сеанса передачи.

Различают два способа передачи последовательного кода: *синхронный* и *асинхронный*.

При **синхронном** методе передатчик генерирует две последовательности - информационную TxD и синхроимпульсы CLK, которые передаются на приемник по разным линиям (Рис. 3.13).

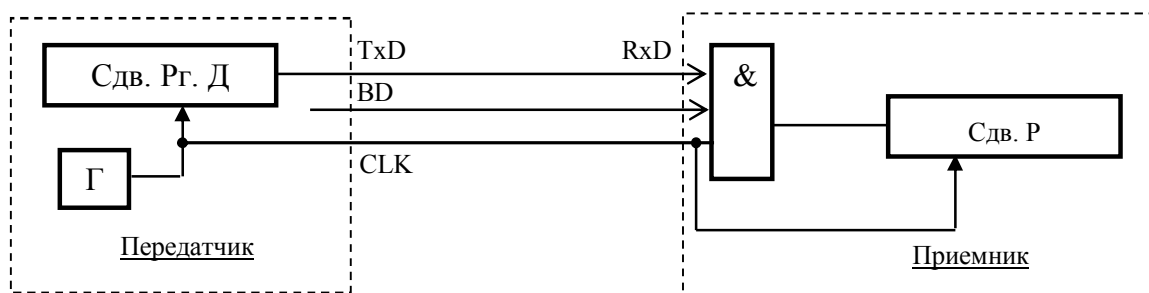


Рис.3.13. Последовательный синхронный обмен с внешней синхронизацией

Синхроимпульсы CLK обеспечивают синхронизацию передаваемых бит, а начало передачи отмечается по-разному. При организации *внешней синхронизации* сигнал начала передачи BD генерируется передатчиком и передается на приемник по специальной линии.

В системах с *внутренней синхронизацией* отсутствует линия BD, а на линию данных генерируются специальные коды длиной 1-2 байта - "сим-

волы синхронизации”. Для каждого приемника предварительно определяются конкретные синхросимволы, таким образом можно осуществлять адресацию конкретного абонента из нескольких, работающих на одной линии. Каждый приемник постоянно принимает биты с RxD, формирует символы и сравнивает с собственными синхросимволами. При совпадении синхросимволов последующие биты поступают в канал данных приемника.

**Асинхронный** способ обеспечивает передачу информации по единственной линии. Для надежной синхронизации обмена в асинхронном режиме

- 1) передатчик и приемник настраивают на работу с одинаковой частотой;
- 2) передатчик формирует стартовый и стоповый биты, отмечающие начало и конец посылки;
- 3) передача ведется короткими посылками (5..9 бит), а частоты передачи выбираются сравнительно низкими.

#### *Универсальный последовательный приемопередатчик KP580BB51*

Универсальный последовательный приемопередатчик (далее - *контроллер последовательного обмена* или просто **контроллер**) предназначен для аппаратной реализации протокола последовательного обмена между микропроцессором и внешним устройством. При передаче контроллер .BB51 преобразует параллельный код, получаемый от процессора, в последовательный поток символов со служебными битами и выдает этот поток в последовательный канал связи с выбираемой скоростью. При приеме осуществляются обратные преобразования.

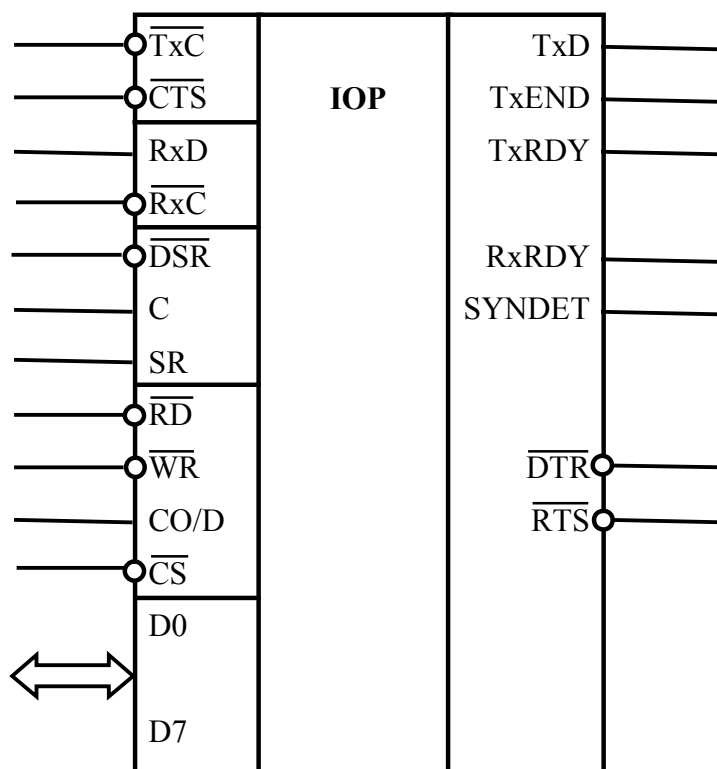


Рис. 3.14. Обозначение контроллера .BB51 на функциональных (принципиальных) схемах

Назначение выводов контроллер .BB51 приведено в Таблица 3.3, а внутренняя структура показана на Рис.3.15.

Таблица 3.3

№ вывода	Русск.	Лат.	Тип вывода	Назначение вывода
8,7,6,5,2 1,28,27	D(7:0)	D(7:0)	вх/вых	Шина данных
20	С	CLK (C)	вх	Синхронизация
21	УСТ	RESET (SR)	вх	Уст. в исходное состояние
12	ПНУ/Д\	C/D\	вх	Признак управление /данные
10	РПМ\	WR\	вх	Запись-разрешение приема с шины данных в контроллер
13	РВД\	RD\	вх	Чтение - разрешение выдачи на шину данных
11	ВД\	CS\	вх	Выбор микросхемы
22	СОИ1\	DSR\	вх	Готовность ВУ передать данные
24	СОИ2\	DTR\	вых	Запрос на передачу данных
17	ПГТ\	CTS\	вх	Готовность ВУ принять данные
23	СОИ3\	RTS\	вых	Запрос на прием данных



№ вывода	Русск.	Лат.	Тип вывода	Назначение вывода
16	С1	SYNDET/BD	вх/вых	Вид синхронизации (различный смысл в разных режимах)
25	СПМ\	RxC\	вх	Синхронизация приемника
14	ПГТПМ	RxRDY	вых	Готовность приемника
3	ВХ ПМ	RxD	вх	Вход данных приемника
9	СВД\	TxC\	вх	Синхронизация передатчика
15	ПГТПЧ	TxRDY	вых	Готовность передатчика
19	ВЫХ ПЧ	TxD	вых	Выход данных передатчика
18		TxEND	вых	Конец передачи
26	Епит	Ucc	-	+ 5 В
4	Общий	GND	-	Общий

**Контроллер работает** в двух режимах - инициализации и приема/передачи.

Режим *инициализации* обеспечивает настройку контроллера. .BB51 на определенный протокол обмена путем загрузки управляющего слова, символов синхронизации (один или два - только для синхронного режима) и подачи ряда команд. Инициализация обязательно начинается с начальной установки контроллера, которая может выполняться аппаратно сигналом SR или подачей команды начальной установки (см. ниже).

После начальной установки загружается управляющее слово, формат которого приведен на Рис. 3.15.

D7	D6	D5	D4	D3	D2	D1	D0
CC		Ч	К	Длина слова		Реж/Час	

Рис. 3.15. Формат управляющего слова контроллера. .BB51

Ниже приводятся значения разрядов управляющего слова и их смысл для различных режимов работы контроллера. .BB51.

*Режим работы* контроллера определяется разрядами D0, D1, причем для трех разновидностей асинхронного режима это поле определяет коэффициент деления частоты синхроимпульсов CLK для получения частоты асинхронного приема/передачи.

D1	D0	
0	0	Синхронный режим
0	1	1 : 1
1	0	1 : 16

Асин-  
хронный

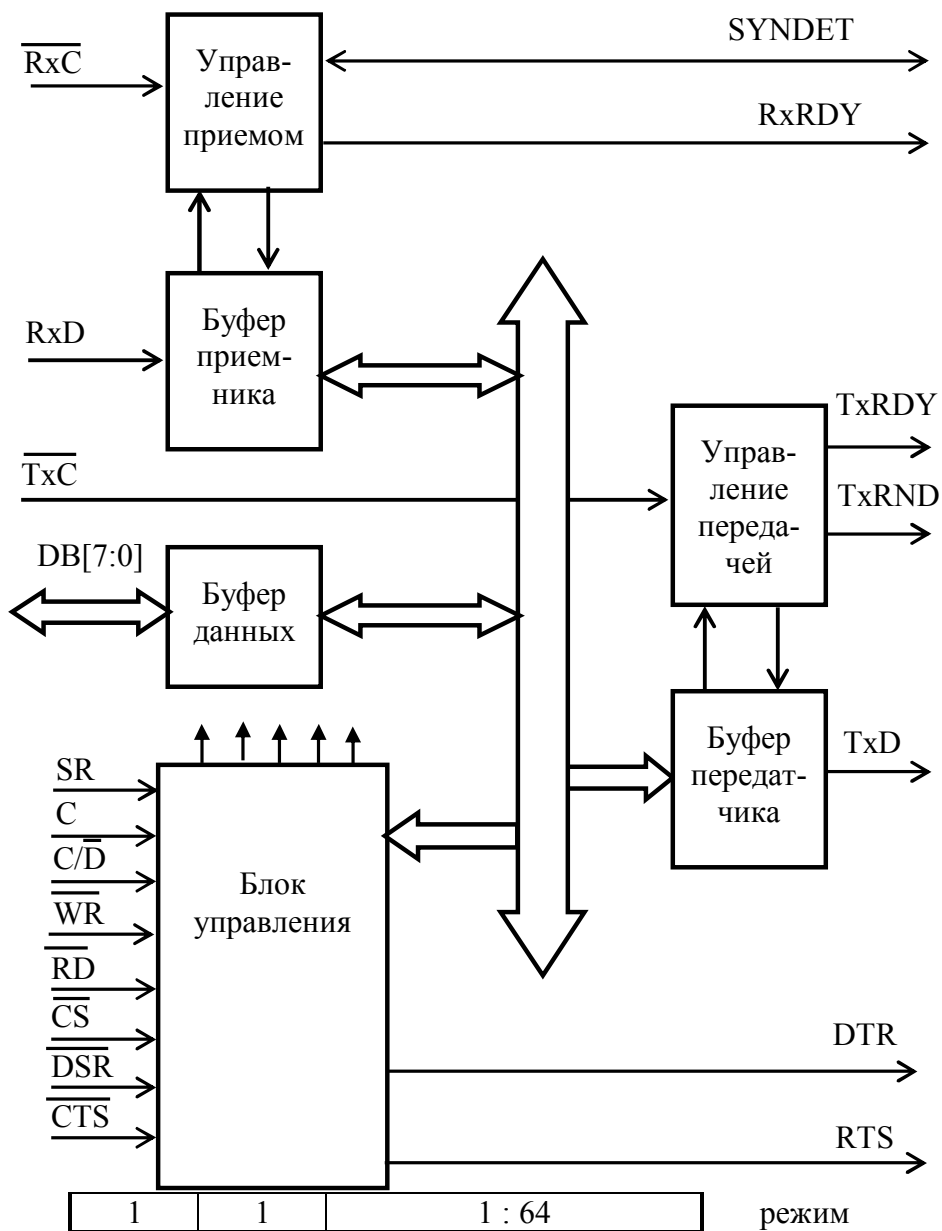


Рис.3.16. Структура контроллер K580VBBB51

Длина слова (кадра) определяется значением разрядов D3 и D2:

D3	D2	Длина кадра (бит)
0	0	5
0	1	6
1	0	7
1	1	8

Разряды D4 и D5 определяют режим контроля на четность:

- |        |                  |        |                           |
|--------|------------------|--------|---------------------------|
| D4 = 0 | - нет контроля;  | D5 = 0 | - контроль по нечетности; |
| D4 = 1 | - есть контроль; | D5 = 1 | - контроль по четности.   |

Разряды D6 и D7 имеют различный смысл для синхронного и асинхронного режима. В **синхронном** режиме D6 задает *вид синхронизации*, а D7 - *число синхросимволов*:

D6 = 0 - внутренняя;	D7 = 0 - два синхросимвола;
D6 = 1 - внешняя;	D7 = 1 - один синхросимвол.

В **асинхронном** режиме поле D7D6 определяет *длительность стоп-бита* в периодах частоты передачи:

D7	D6	Длина стоп-бита
0	0	Не используется
0	1	1
1	0	1,5
1	1	2

Последовательность программирования контроллера ..BB51 и соответствующие входные управляющие сигналы приведены в Таблица 3.4.

Таблица 3.4

№	Последовательность программирования	Сигналы на входах			
		CO/D	WR\	CS\	SR
1	Установка исходного состояния	X	X	X	1
2	Запись инструкции режима	1	0	0	0
3	Запись синхросимвола	1	0	0	0
4	Запись синхросимвола	1	0	0	0
5	Запись команды	1	0	0	0

Данные, передаваемые по шине D[7:0], в отличие от команд, инструкций режима и синхросимволов сопровождаются низким уровнем на линии CO/D.

Каждый бит команды, установленный в "1", имеет свой смысл, независимый от остальных разрядов команды:

- D0 - разрешена передача;
- D1 - запрос о готовности приемника;
- D2 - разрешен прием;
- D3 - пауза;
- D4 - сброс триггеров ошибок;
- D5 - запрос о готовности передатчика;
- D6 - программный сброс;
- D7 - поиск синхросимволов

Прием/передача может осуществляться в одном из двух режимов - *синхронном* или *асинхронном*.

**Синхронный режим** предназначен для передачи массива слов. В зависимости от способа синхронизации (с внутренней или внешней синхронизацией) определенным образом отмечается начала приема, после чего все поступающие на вход RxD биты упаковываются в слова заранее определенного формата. Способ фиксации момента начала приема определяет режим синхронного обмена.

Различают два режима синхронного приема. В режиме *приема с внутренней синхронизацией* приемник начинает работу с поисков символа (символов) синхронизации. Информация принимается по входу RxD на первый регистр приемника и непрерывно сравнивается с содержимым регистра первого синхросимвола. Если содержимое двух регистров не одинаково, то в регистр приемника принимается следующий бит и сравнение повторяется. Когда содержимое сравниваемых регистров становится одинаковым, контроллер заканчивает поиск и переходит в режим синхронизации. При этом, если не запрограммирован контроль по четности (нечетности), на выводе SYNDET/BD, работающем как выход, во время приема последнего бита синхросимвола устанавливается напряжение высокого уровня, сигнализируя ВУ, что произошел захват синхронизации. Если контроллер .BB51 запрограммирован с двумя символами синхронизации и/или контролем на четность, то указанная выше ситуация произойдет во время приема последнего бита второго синхросимвола или бита контроля.

В режиме *синхронного приема с внешней синхронизацией* на вывод SYNDET/BD, работающий как вход, подается напряжение синхронизации, которое разрешает прием информации по входу RxD со скоростью синхросигналов, поступающих на вход RxC\ . Если в управляющем слове определен прием в синхронном режиме с внешней синхронизацией, то цепи внутренней синхронизации блокируются установкой внутреннего триггера, сброс которого возможен по сигналу SR или после чтения состояния контроллера.

*Синхронная передача* начинается при подаче на вход CTS\ низкого уровня. Передатчик начинает трансляцию битов буферного регистра на выход TxD со скоростью синхроимпульсов, поступающих на вход TxC.

**Асинхронный режим** характеризуется одиночными посылками информации, инициализация которых определяется либо процессором системы, либо внешним устройством. Синхронизация передатчика и приемника обеспечивается равенством частот передачи и приема (частоты задаются программно). Для контроллера .BB51 возможны три частоты передачи/приема: 1:1, 1:16 и 1:64 относительно частоты синхронизации на входе CLK. Неактивному состоянию передатчика соответствует высокий уровень на линии данных.

В начале каждой посылки устанавливается низкий уровень на выходе TxD данных передатчика т.н. "*старт-бит*", длительность которого равна периоду заданной передающей частоты. Старт-бит служит для ввода в

синхронизацию передатчика и приемника. В конце посылки выдается "стоп-бит", длительность которого программируется на 1, 3/2 или 2 периода передатчика.

*Асинхронная передача* начинается с записи в контроллер данных и при условии, что в D0 инструкции команды записана "1" и на входе CTS\ - низкий уровень. Передатчик формирует на выходе TxD стартовый бит, заданное число бит данных (5..8, начиная с бита d0), если необходимо - контрольный бит и - обязательно - стоповый бит. Если контроллер не содержит информации для передачи, то на линии TxD удерживается высокий уровень, а если в инструкции запрограммирован режим "Пауза", то на TxD выдается низкий уровень.

*Асинхронный прием.* Обнаружение на RxD перехода с высокого уровня на низкий интерпретируется как приход старт-бита. Истинность этого бита проверяется вторично через 1/2 периода приема и если наличие низкого уровня подтверждается, то запускается счетчик битов, который позволяет определить конец битов данных, бит контроля (если он запрограммирован) и стоп-бит. Если же низкий уровень в середине периода не подтверждается, то приемник переходит в исходное состояние.

Схема управления и синхронизации приемника предохраняет от ошибочного запуска счетчика битов, если на выводе RxD присутствует напряжение низкого уровня, вызванное командой D3 "Пауза". Регистр приемника обнаруживает паузу и на выводе SYNDET/BD устанавливает высокий уровень.

Если при приеме обнаруживаются ошибки четности или длительности стоп-бита, то соответствующие триггеры ошибок устанавливаются в единичное состояние. Принятые данные передаются через внутренние шины в выходной регистр данных, и на выводе RxRDY устанавливается высокий уровень. Если предыдущий символ еще не считан процессором, то он теряется, и устанавливается триггер ошибки переполнения.

Наличие единиц в триггерах ошибок не останавливает работу контроллера. Триггеры ошибок сбрасываются командой с D4 = 1.

Состояние контроллера отображается в регистре состояний, доступном по чтению процессору (при CS = 0, RD = 0, CO/D = 1). Формат регистра состояний приведен ниже:

7	6	5	4	3	2	1	0
DSR	SYN/BD	D5	D4	D3	TxEND	RxRDY	TxRDY

Разряды 7, 6, 2, 1, 0 дублируют состояния соответствующих выводов контроллера (см. табл. 6.3); D5 - триггер ошибки стоп-бита, D4 - триггер ошибки переполнения (устанавливается, если предыдущий символ не прочитан процессором до записи в буферный регистр нового принятого символа), D3 - триггер ошибки четности.

### 3.3. Подсистема прерываний в МПС

Подсистема прерываний - совокупность аппаратных и программных средств, обеспечивающих реакцию программы на события, происходящие вне программы. Такие события возникают, как правило, случайно и асинхронно по отношению к программе и требуют прекращения (чаще временного) выполнения текущей программы и переход на выполнение другой программы (подпрограммы), соответствующей возникшему событию.

#### 3.3.1. Внутренние и внешние прерывания

Различают *внутренние* и *внешние* (по отношению к процессору) события, требующие реакции подсистемы прерываний. К *внутренним* событиям относятся переполнение разрядной сетки при выполнении арифметических операций, попытка деления на 0, извлечение корня четной степени из отрицательного числа и т.п., появление несуществующего кода команды, обращение программы в область памяти, для нее не предназначенную, сбой при выполнении передачи или операции в АЛУ и многое другое. Внутренние прерывания должны обеспечиваться развитой системой аппаратного контроля процессора, поэтому они не получили широкого распространения в простых 8- и 16-разрядных МП.

*Внешние прерывания* могут возникать во внешней по отношению к процессору среде и отмечать как аварийные ситуации (кончилась бумага на принтере, температура в реакторе превысила допустимый уровень, исполнительный орган робота дошел до предельного положения и т.п.), так и нормальные рабочие события, которые происходят в случайные моменты времени (нажатие клавиши на клавиатуре, исчерпан буфер принтера или ВЗУ и т.п.). Во всех этих случаях требуется прервать выполнение текущей программы и перейти на выполнение другой программы (подпрограммы), обслуживающей это событие.

С точки зрения реализации внутренние и внешние прерывания функционируют одинаковым образом, хотя при работе подсистемы с внешними прерываниями возникают дополнительные проблемы идентификации источника прерывания. Поэтому ниже будут рассмотрены внешние прерывания.

Анализ состояния внешней среды можно осуществлять путем программного сканирования - считывания через определенные промежутки времени слов состояния всех возможных источников прерываний, выделения признаков отслеживаемых событий и переход (при необходимости) на прерывающую подпрограмму.

Однако, такой способ не обеспечивает для большинства применений приемлемого времени реакции системы на события, особенно при необходимости отслеживания большого числа событий. К тому же при коротком

цикле сканирования большой процент процессорного времени тратится на проверку (чаще безрезультатную) состояния внешней среды.

Гораздо эффективней организовать взаимодействие с внешней средой таким образом, чтобы всякое изменение состояния среды, требующее реакции МПС, вызывало появление на специальном входе МП сигнала прерывания текущей программы. Организация прерываний должна быть обеспечена определенными аппаратными и программными средствами, которые мы и называем "**подсистемой прерываний**".

### ***3.3.2. Функции подсистемы прерываний и её реализация***

Подсистема прерываний должна обеспечивать выполнение следующих функций:

- 1) обнаружение изменения состояния внешней среды (запрос на прерывание);
- 2) идентификация источника прерывания;
- 3) разрешение конфликтной ситуации в случае одновременного возникновения нескольких запросов (приоритет запросов);
- 4) определение возможности прерывания текущей программы (приоритет программ);
- 5) фиксация состояния прерываемой (текущей) программы;
- 6) переход к программе, соответствующей обслуживаемому прерыванию;
- 7) возврат к прерванной программе после окончания работы прерывающей программы.

Рассмотрим варианты реализации в МПС перечисленных выше функций.

(1) Фиксация изменения состояния внешней среды может осуществляться различными средствами: двоичными датчиками, компараторами, схемами формирования состояний и т.п. Будем полагать, что все эти средства формируют в конечном итоге логические сигналы запроса на прерывание  $z$ , причем для определенности будем считать, что активное состояние этого сигнала передается уровнем логической единицы ( $H$  - уровень).

Количество источников запросов в МПС может быть различно, в том числе и довольно велико. Дефицит внешних выводов МП исключает возможность передачи запросов от ВУ по "собственным" линиям интерфейса. Обычно на одну линию запроса подключается несколько источников прерываний (по функции ИЛИ), а иногда и все источники системы - на единственный вход (как в i8080).

Различают два типа входов запросов на прерывания - *радиальные* и *векторные*. Процессор анализирует состояние входов запросов в конце каждого машинного цикла.

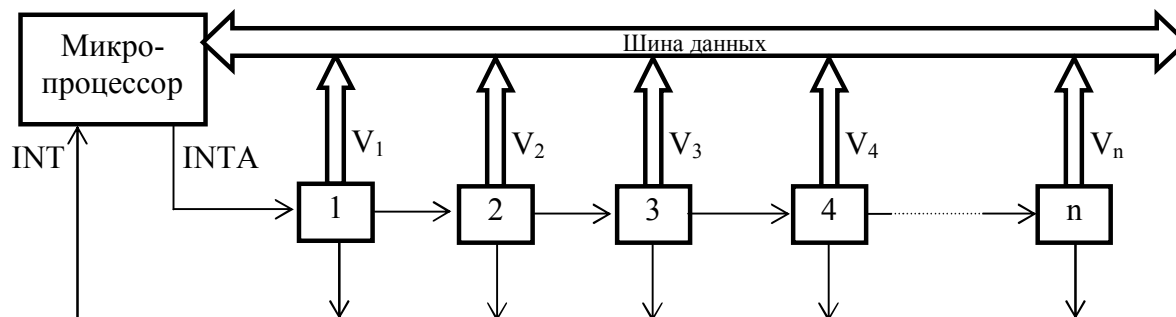


Рис. 3.17. Организация векторного прерывания

(2) Получив запрос на прерывание, процессор должен идентифицировать его источник, т.е. в конечном счете, определить начальный адрес обслуживающей это прерывание программы. Способ идентификации зависит от типа входа, на который поступил запрос.

Каждый радиальный вход связан с определенным адресом памяти, по которому размещается указатель на обслуживающую программу или сама программа. Если на радиальный вход поступает несколько запросов, то необходимо осуществить программную идентификацию источника путем последовательного (в порядке убывания приоритетов) опроса всех возможных источников прерывания. Этот способ не требует дополнительных аппаратных затрат и одновременно решает проблему приоритета запросов (3), однако время реакции системы на запрос может оказаться недопустимо велико, особенно при большом числе источников прерываний.

Гораздо чаще в современных МПС используется т.н. "векторная" подсистема прерываний (рис. 3.17). В такой системе микропроцессор, получив запрос на векторном входе INT, выдает на свою выходную линию сигнал подтверждения прерывания INTA, поступающий на все возможные источники прерывания. Источник, не выставивший запроса, никак не реагирует на сигнал INTA. Источник, выставивший запрос, получая сигнал INTA, выдает на системную шину данных "вектор прерывания" – свой номер или адрес обслуживающей программы или, чаще, адрес памяти, по которому расположен указатель на обслуживающую программу. Время реакции МПС на запрос векторного прерывания минимально (1..3 машинных цикла) и не зависит от числа источников.

(3) Для исключения конфликтов при одновременном возникновении нескольких запросов на векторном входе ответный сигнал INTA подается на источники запросов не параллельно, а последовательно – в порядке убывания приоритетов запросов. Источник, не выставивший запроса,



транслирует сигнал INTA со своего входа на выход, а источник, выставивший запрос, блокирует дальнейшее распространение сигнала INTA. Таким образом, только один источник, выставивший запрос, получит от процессора сигнал INTA и выдаст по нему свой вектор на шину данных.

Более гибко решается проблема организации приоритетов запросов при использовании в МПС специальных контроллеров прерываний (см. ниже).

Конфликты на радиальном входе исключаются самим порядком программного опроса источников.

(4) Прерывание в общем случае может возникать не только при решении "фоновой" задачи, но и в момент работы другой прерывающей программы, причем не всякую прерывающую программу допустимо прерывать любым запросом. В фоновой задаче так же могут встречаться участки, при работе которых прерывания (все или некоторые) недопустимы. В общем случае в каждый момент времени работы процессора должно быть выделено подмножество запросов, которым разрешено прерывать текущую программу.

В МПС эта задача решается на нескольких уровнях. В процессоре обычно предусматривается программно-доступный флаг разрешения/запрещения прерывания, значение которого определяет возможность или невозможность всех прерываний. Для создания более гибкой системы приоритетов программ на каждом источнике прерываний может быть предусмотрен специальный программно-доступный триггер разрешения формирования запроса. В таком случае возможно формирование произвольного подмножества разрешенных в данный момент источников прерываний.

(5..7) В разделе **Ошибка! Источник ссылки не найден.** упоминалась иерархия процессов в МП: машинный такт - машинный цикл - командный цикл. Рассмотрим возможность прерывания программы по окончании различных процессов. Учитывая, что прерванная программа должна быть запущена по окончании работы прерывающей с того места, где она была прервана, подсистема прерываний МПС должна обеспечить фиксацию полного состояния прерываемой программы на момент прерывания.

При прерывании после текущего машинного такта требуется запоминать не только состояние всех регистров процессора (программно-доступных и системных), но и состояние первичного управляющего автомата. Реализация процедуры фиксации состояния и последующего восстановления потребует значительных затрат дополнительного оборудования и/или времени.

Значительный объем информации требуется запоминать и при прерывании программы после текущего машинного цикла (выбранный фрагмент или всю команду, выбранные операнды или сформированные адреса).

Поэтому в большинстве МП прерывание может осуществляться после выполнения очередной команды. Состояние программы в этом случае ха-

рактируется содержимым счетчика команд (адрес следующей команды), а так же содержимым РОН и регистра флагов. Процедура перехода к прерывающей программе и последующего возврата из нее может быть полностью идентична действиям, выполняемым по командам ВЫЗОВ и ВОЗВРАТ. Состояние программного счетчика (а иногда и регистра флагов или PSW) аппаратно фиксируется в стеке, а значение РОНов - при необходимости программно в самом тексте прерывающей программы. Учитывая, что большинство команд МП являются короткими, время реакции МПС на запрос прерывания при анализе запросов по завершению текущего командного цикла не бывает большим.

### 3.3.3. Микропроцессорные контроллеры прерываний

Внешние устройства, включенные в подсистему прерываний, должны реализовать несколько функций, связанных с работой в этой подсистеме - формирование запроса, анализ ответа процессора, выдачу вектора прерывания. Кроме того, в подсистеме необходимо обеспечить дисциплину обслуживания запросов. Перечисленные функции могут быть реализованы на специальных устройствах - контроллерах прерываний, которые выпускаются в виде БИС в составе многих микропроцессорных комплектов.

На Рис. 3.18 приведена структурная схема простого контроллера прерываний K589ИК14, применяемого совместно с МП K580ВМ80 (на схеме показаны лишь основные блоки и выходы контроллера).

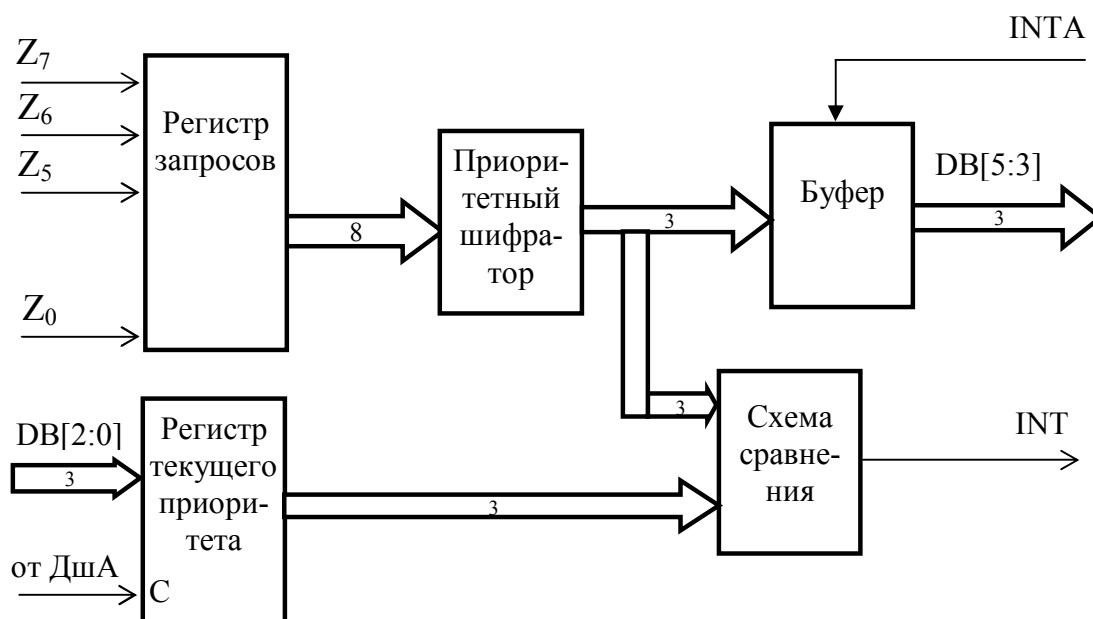


Рис. 3.18. Контроллер K589ИК14

Контроллер включает в себя следующие блоки:

- RгЗП - регистр запросов, на который поступают запросы на прерывание z0..z7 от внешних источников;
- ПШ - приоритетный шифратор, формирующий на выходе трехразрядный код номера возбужденного входа (соответствующего разряда RгЗП), причем, если на входе ПШ присутствуют несколько единиц, то на выходе будет сформирован код большего номера;
- Буф. - буферная схема, подключающая выход ПШ к линиям системной шины данных при подаче на управляющий вход сигнала подтверждения прерывания INTA;
- RгТП - регистр текущего приоритета, программно-доступный в пространстве ввода/вывода и хранящий трехразрядный код приоритета текущей программы;
- СС - схема сравнения, формирующая запрос на прерывание INT процессору при условии, что код наибольшего из поступивших запросов больше кода текущего приоритета.

МП проверяет состояние входа INT в конце каждого командного цикла при условии установки в "1" триггера разрешения прерывания TI. Если INT = 1, то МП вместо машинного цикла M1 следующего командного цикла выполняет цикл обслуживания прерывания, который отличается от M1 только тем, что вместо сигнала RDM\ вырабатывается сигнал INTA\ и программный счетчик PC не инкрементируется. Поэтому не происходит чтение команды из памяти (хотя ее адрес и выставлен на AB[15:0]), а по сигналу INTA\ контроллер выставляет на DB[5:3] трехразрядный номер запроса. На остальных линиях DB сохраняются лог. "1" (высокоимпедансное состояние). В такте T<sub>3</sub> цикла обслуживания прерывания код с DB[7:0] записывается в регистр команд МП.

Код 11nnn111 представляет собой код одной из восьми команд RST 0..RST 7, которые выполняются аналогично командам CALL, но с фиксированными адресами вызываемых подпрограмм:

```
RST 0 = CALL 0000h
RST 1 = CALL 0008h
RST 2 = CALL 0010h
. . . . .
RST 7 = CALL 0038h
```

Таким образом, каждому запросу соответствует фиксированный адрес обслуживаемой его подпрограммы.

К недостаткам рассмотренного выше контроллера прерываний K589ИК14 относится жесткая система приоритетов запросов, не позволяющая обеспечить равновероятное обслуживание источников прерываний. Кроме того, контроллер не позволяет выделить произвольное подмножество разрешенных в данный момент прерываний из множества поступающих на его входы запросов. Возможно лишь задать определенный уровень, начиная с которого запросы будут разрешены.

Использование ..ИК14 накладывает ограничение на расположение обслуживающих прерывания подпрограмм или указателей на них в адресном пространстве памяти. Команды RST вызывают подпрограммы из начальной области адресов памяти (0000..0038), что не всегда удобно при организации МПС.

Программируемый контроллер прерываний K580BH59 позволяет организовать более гибкую и эффективную подсистему прерываний в МПС.

Отличительной особенностью контроллера ..BH59 является то, что он генерирует код команды CALL, поэтому область векторов прерываний (по-прежнему компактная) может располагаться по произвольным адресам памяти. Структурная схема контроллера прерываний представлена на Рис. 3.19.

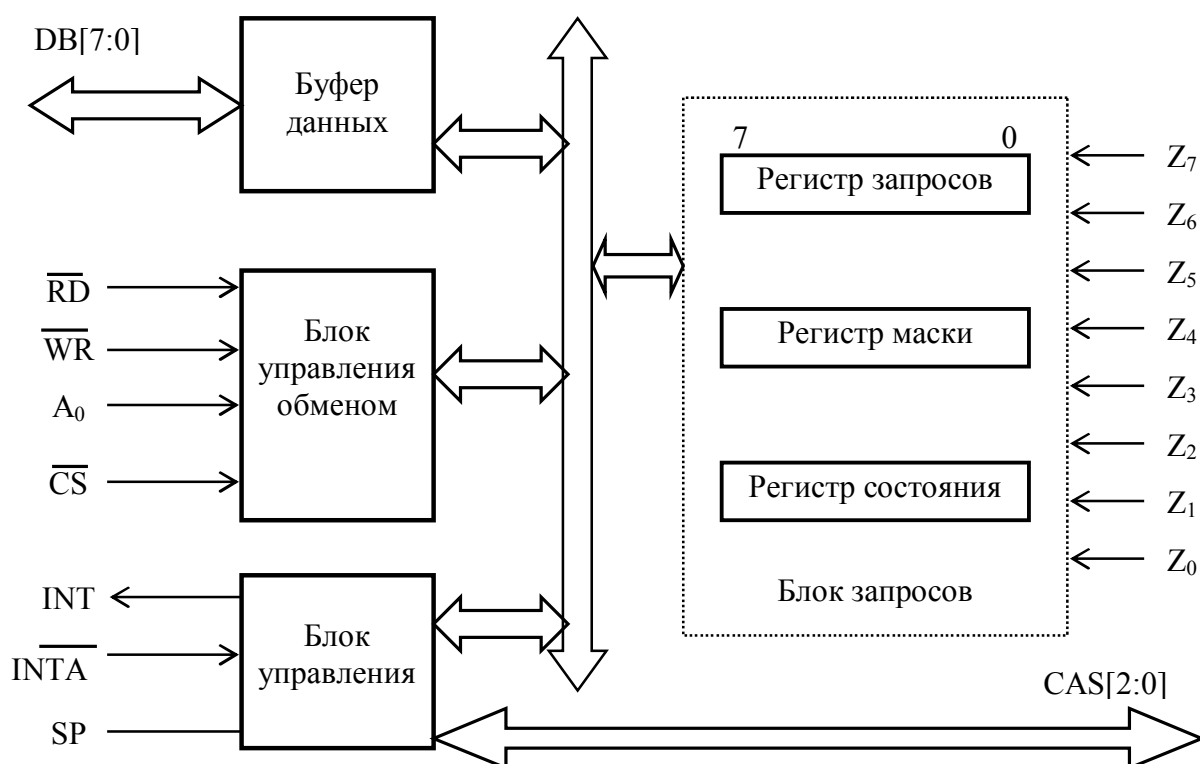


Рис. 3.19. Структура контроллера прерываний K580BH59

Программируемый контроллер прерываний включает в себя блоки связи с системной шиной, управления и запросов. Блок запросов содержит три 8-разрядных регистра - регистр запросов RgZ, в котором фиксируются запросы от источников прерываний, регистр маски RgM, определяющий подмножество источников, которым разрешены прерывания и регистр состояний RgC, в котором фиксируются запросы, принятые на обслуживание.

Состав управляющих линий контроллера включает стандартные линии подключения к системной шине : D[7:0], A0, WR, RD, CS; линии, передающие запрос на прерывания процессору и ответ МП INT и INTA\ соответственно.

Линии запросов z7..z0 соединяют контроллер с источниками прерываний. Контроллер может обслуживать до 8 источников прерываний. При большем числе источников возможно каскадное включение ..BH59, причем один из контроллеров будет ведущим, а остальные (не более 8) - ведомыми. Для назначения роли контроллера в системе предназначен вход SP (H - уровень соответствует ведущему контроллеру). Для организации взаимодействия каскадированных контроллеров прерываний предназначены линии CAS[2:0].

**Программирование контроллера** осуществляется путем загрузки в специальные регистры двух или трех управляющих слов, форматы которых показаны на Рис. 3.20. Загрузка этих слов осуществляется командами инициализации в начале работы системы.

Команда	A0	7	6	5	4	3	2	1	0
КИ1	0	A7	A6	A5	1	0	F	S	0
КИ2	1	A15	A14	A13	A12	A11	A10	A9	A8
КИЗ-1	1	z7	z6	z5	z4	z3	z2	z1	z0
КИЗ-2	1	x	x	x	x	x	n	n	n

Рис. 3.20. Форматы команд инициализации контроллера ..BH59

В команде КИ1 разряд S определяет количество контроллеров прерываний в системе : "1" - один контроллер, "0" - более одного (имеет место каскадирование). Разряд F задает шаг между соседними адресами подпрограмм, обслуживающих прерывания : "1" - 4 байта, "0" - 8 байт.

Разряды A7, A6, A5 вместе со всеми разрядами КИ2 образуют старшие 11 бит адреса подпрограммы. При значении F = 1 вектор прерывания (адресная часть команды CALL) формируется следующим образом :

$$A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 n n n 0 0,$$

где nnn - номер обслуживаемого запроса на прерывание (номер входа z). Если задан шаг между адресами 8 байт (F = 0), то разряд A5 игнорируется и вектор генерируется след. образом :

$$A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 n n n 0 0 0.$$

Третья команда инициализации подается (и принимается контроллером) только при наличии нескольких контроллеров в системе (S = 0), причем формат этой команды различен для ведущего и ведомых контроллеров.

Ведущему контроллеру командой КИЗ-1 указывается, к каким его входам z подключены ведомые контроллеры (в соответствующем разряде КИЗ-1 установлена "1"). "0" в разряде КИЗ-1 означает, что на соответствующий вход z подключен обычный источник прерываний.

Каждому ведомому контроллеру в трех младших разрядах КИЗ-2 указывается номер входа ведущего контроллера, к которому подключен данный ведомый. Состояние разрядов КИЗ-1[7:3] безразлично.

Помимо команд инициализации, выполняемых один раз в начале работы МПС, контроллер в любое время может выполнить команды управления, позволяющие установить маску запросов (КУ1), задать режим приоритета запросов (КУ2) или определить будущую операцию. Форматы команд управления приведены на рис. 3.21.

Команда КУ1 устанавливает новое значение в регистр маски, причем если разряд РгМ установлен в "1", то соответствующий вход z считается замаскированным и прерывание от этого источника запрещено.

Команда	A0	7	6	5	4	3	2	1	0
КУ1	1	m7	m6	m5	m4	m3	m2	m1	m0
КУ2	0	ЦП	К	СВ	0	0	k2	k1	k0
КУ3	0	x	СМ1	СМ0	0	1	СП	Р1	Р0

Рис. 3.21. Форматы команд управления контроллера. .ВН59

КУ2 задает тип приоритета запросов (ЦП = 0 - фиксированный приоритет, z0 -высший, z7 -низший; ЦП = 1 - циклический приоритет). При циклическом приоритете этой же командой задается номер входа (в разрядах k[2:0]), которому присвоен низший приоритет. После обслуживания *любого запроса* приоритеты входов циклически меняются на одну позицию. Разряд СВ КУ2 позволяет сбросить бит регистра РгС, номер которого задан битами k[2:0] КУ2. Разряд К определяет, используется ли в команде поле k[2:0].

Команда КУ3 позволяет установить или сбросить режим спецмаскирования, при котором на обслуживание принимаются запросы с приоритетом, ниже текущего. КУ3[6:5] = 11 - установить режим спецмаскирования, КУ3[6:5] = 10 - снять режим спецмаскирования, КУ3[6:5] = 0x - не воздействует.

Эта же команда подготавливает в следующем такте один из допустимых к чтению объектов: регистр запросов при КУ3[1:0] = 10; регистр состояний при КУ3[1:0] = 11; номер запроса с наивысшим приоритетом по формату Рис. 3.22 при КУ3[2:0] = 10x. Считывание осуществляется командой IN при A0 = 0, WR = 0. Признак INT отмечает наличие запроса, а код w[2:0] определяет номер незамаскированного запроса с наивысшим приоритетом. Результатом чтения информации по формату Рис. 3. можно воспользоваться при организации программной идентификации источника прерывания.

7	6	5	4	3	2	1	0
INT	x	x	x	x	w2	w1	w0

Рис. 3.22. Формат слова номера запроса с наивысшим приоритетом

Считывание содержимого регистра маски может осуществляться командой  $IN$  при  $A0 = 1$  без предварительной загрузки команды КУЗ.

При организации векторных прерываний контроллер, получив запросы на входах  $z0..z7$ , фиксирует их в соответствующих разрядах  $PгЗ$ . При наличии незамаскированных запросов контроллер формирует для МП сигнал  $INT$ , а, получив ответ  $INTA\backslash$ , принимает на обслуживание наиболее приоритетный из незамаскированных запросов, для чего

- (1) устанавливает в "1" соответствующий разряд  $PгС$ ;
- (2) выдает на шину данных  $DB$  код команды  $CALL$  (константу  $CDh$ );
- (3) по второму и третьему сигналу  $INTA\backslash$  выдает на  $DB$  адрес подпрограммы обслуживающей выбранное прерывание, сгенерированный по правилам, описанным выше.

**Использование нескольких контроллеров прерываний.** Если необходимо обслуживать в МПС более 8 источников прерываний, то можно использовать несколько контроллеров  $..BH59$ , причем один из них будет ведущим (для него  $SP = 1$ ), а остальные (до 8) - ведомые ( $SP = 0$ ). Все контроллеры связываются линиями каскадирования  $CAS[2:0]$  по принципу общей шины, причем линии  $CAS$  ведущего контроллера являются выходами, а ведомых – входами (Рис.3.23).

Все контроллеры подключаются к системной шине одинаковым образом, только выходной сигнал  $INT$  используется по не одинаково:  $INT$  ведущего контроллера подается по системной магистрали на вход  $INT$  МП, а  $INT$  ведомых подключаются на входы запросов  $z$  ведущего контроллера. Некоторые входы  $z$  ведущего контроллера и все входы ведомых могут использоваться для приема запросов на прерывание. Максимальное число контроллеров в системе - 9, т.к. к входам ведомых нельзя подключать третий "слой" контроллеров. Максимальное число обслуживаемых запросов - 64.

Все контроллеры инициализируются тремя командами, как было пояснено выше. Если запрос поступает на ведущий контроллер, то он обслуживается так же, как и при единственном контроллере в МПС. Запрос, поступивший на ведомый контроллер, обрабатывается по стандартным правилам и преобразуется в сигнал  $INT$ , который в качестве запроса поступает на ведущий контроллер.

Ведущий контроллер обрабатывает все поступившие на его входы запросы согласно заданным приоритетам и формирует сигнал  $INT$  для МП. Если запрос, принятый ведущим на обслуживание, поступил от ведомого контроллера (ведущий "знает" об этом из информации, содержащейся в  $КИЗ-1$ ), то ведущий в ответ на первый сигнал  $INTA\backslash$  выдает на  $DB$  константу  $CDh$  (код команды  $CALL$ ) и на линии каскадирования  $CAS[2:0]$  - номер обслуживаемого входа. Все ведомые контроллеры сравнивают со

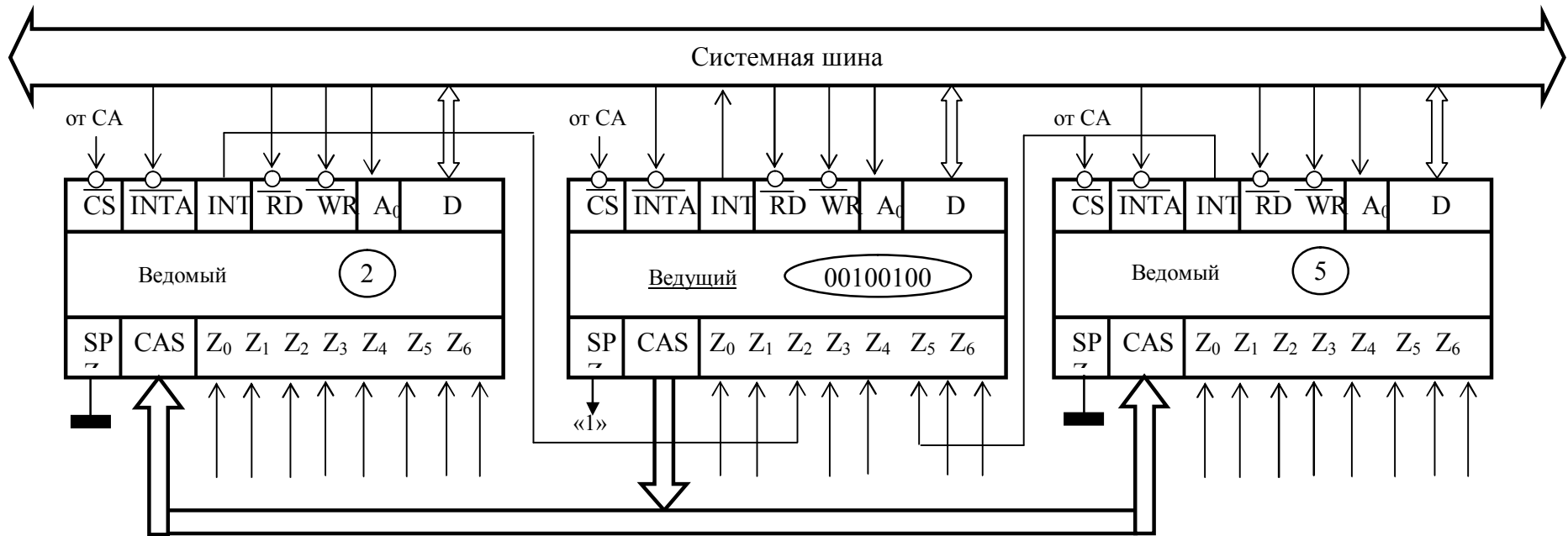


Рис.3.23. Каскадирование контроллеров прерывания





стояние линий CAS со своим номером, заданным командой КИЗ-2. Ведомый контроллер, опознавший свой номер на линиях CAS, в ответ на второй и третий сигналы INTA\ выдает соответственно младший и старший байт команды CALL - адрес подпрограммы, обслуживающей выбранное прерывание.

### 3.4. Подсистема прямого доступа в память

Рассмотрим вариант программно-управляемого обмена между памятью и внешним устройством в МПС на базе МП i8080. Пусть необходимо передать массив данных длиной L, начиная с адреса ADR на ВУ с адресом AIO. Положим, что начальный адрес массива загружен в регистровую пару HL, а длина массива - в регистр C. Тогда фрагмент программы обмена может иметь следующий вид:

Таблица 3.5

Метка	Мnemonic	Комментарий	Количество машинных циклов
LM:	MOV A,M	Чтение байта в Акк.	2
	OUT AIO	Выдача байта на ВУ	3
	INX H	Модификация адреса	1
	DCR C	Модификация счетчика	1
	JNZ LM	Переход, если массив не исчерпан	3
Всего:			10

Таким образом, для того, чтобы в рамках процедуры копирования массива данных из памяти в ВУ переслать один байт данных потребуется десять машинных циклов (МЦ). Процессоры с более совершенной системой команд могут использовать для этой цели меньшее число МЦ, но все равно их будет более одного.

Управляя обменом, микропроцессор "ведет" два счетчика - адресов массива и количества переданных байт и формирует сигналы управления на магистраль. Если снабдить ВУ аппаратными счетчиками и схемой формирования управляющих сигналов (т.н. "канал прямого доступа в память"), то передачу одного байта (слова) можно осуществить за один МЦ без участия процессора. Необходимо лишь на время передачи данных под управлением канала ПДП блокировать работу процессора, отключив его от системной шины. Для этого служит вход захвата шины HOLD. Если подать на него высокий уровень, то МП по окончании текущего МЦ безусловно перейдет в режим ожидания, переведя все свои выходные линии, кроме HLDA и WAIT, в высокоимпедансное состояние, а выходы HLDA и WAIT - в состояние лог. "1". Выходной сигнал HLDA используется для отключе-

ния МП от системной шины - перевода шинных формирователей, включенных между локальной и системной шиной, в высокоимпедансное состояние.

Если в МПС используется несколько ВУ, снабженных каналом ПДП, то целесообразно использовать специальный контроллер ПДП, который обеспечивает программирование каналов ПДП, подключение их к системной шине и дисциплину обслуживания.

### 3.4.1. Контроллер прямого доступа в память

Микросхема K580BT57 предназначена для управления передачей информации между ВУ о ЗУ в режиме ПДП. Основное назначение контроллера - формирование последовательности адресов и управляющих сигналов.

Контроллер ПДП имеет четыре независимых канала обмена. Блок управления каждого канала включает два 16-разрядных регистра : регистр начального адреса РА и регистр управления РУ, причем РУ[13:0] хранит длину передаваемого массива (количество передаваемых байт), а РУ[15:14] - режим работы канала. Структурная схема контроллера и схема подключения его к МПС показана на **Ошибка! Источник ссылки не найден.**25. Контроллер включает в себя, помимо четырех каналов ПДП, схему управления, обеспечивающую связь с системной шиной, разрешение конфликтных ситуаций в соответствии с заданной системой приоритетов и управление работой контроллера в различных режимах.

Схема управления содержит два восьмиразрядных регистра, общие для всех каналов - регистр режимов РгР и регистр состояний РгС, причем РгР доступен процессору только по записи, а РгС только по чтению. Форматы регистров РгР и РгС показаны на Рис.3.4.

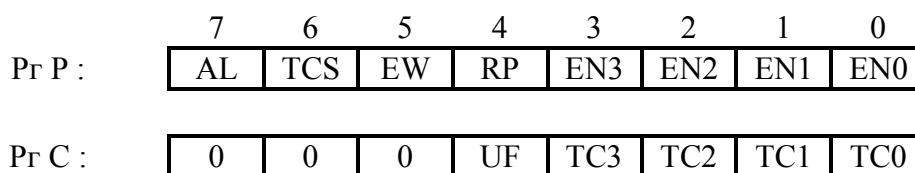
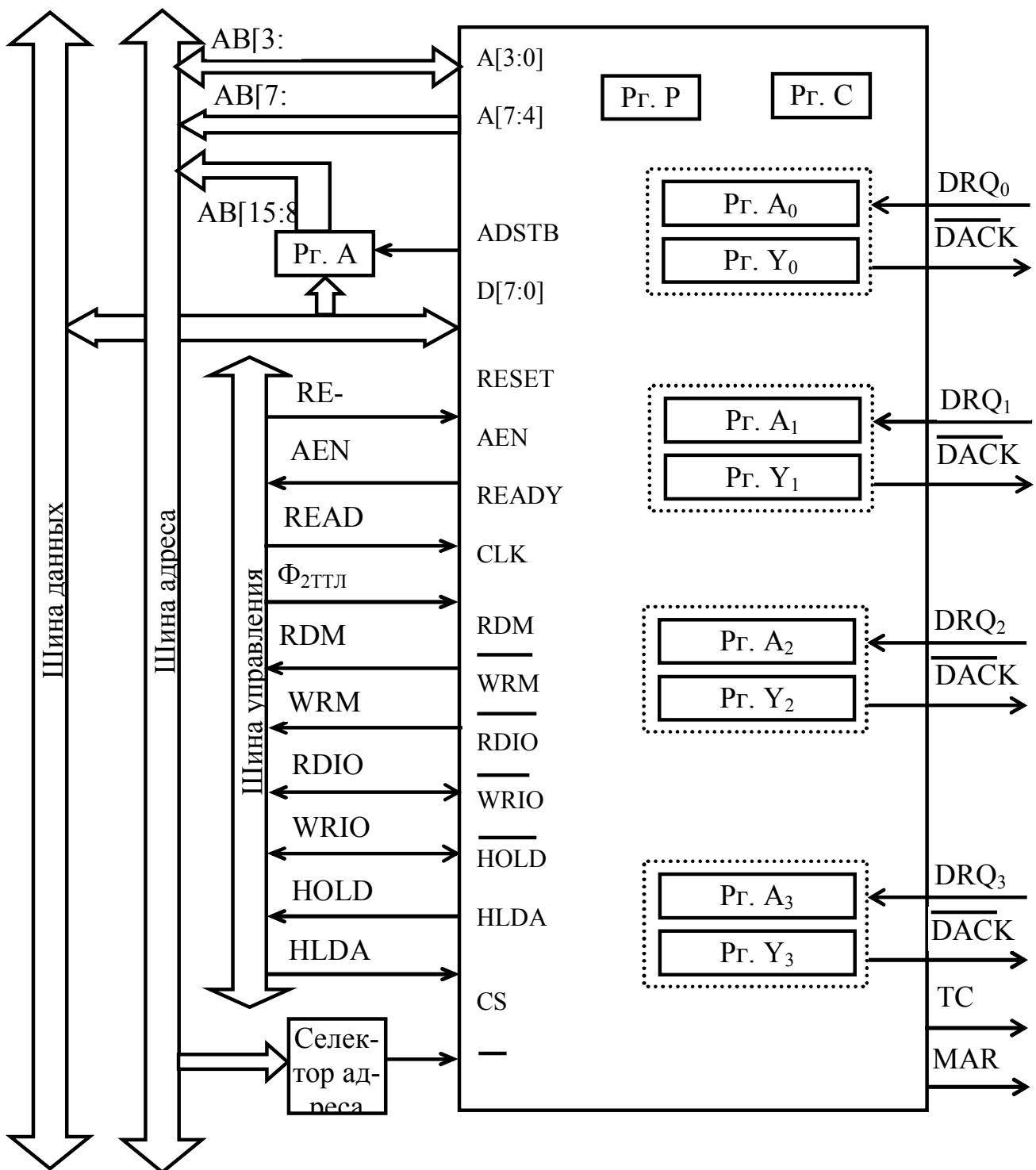


Рис.3.24. Форматы регистров режима и состояния

Рис. 3.25. Контроллер прямого доступа в память

В регистре режимов разряды имеют следующий смысл:



- AL - автозагрузка;
- TCS - отключение канала;
- EW - расширенная запись;
- RP - циклический приоритет;
- EN[3:0] - разрешение работы каналов 3..0.

В регистре состояния:

- UF - флаг обновления;

- TC[3:0] - конец счета в каналах 3..0.

Выводы контроллера можно разделить на две группы : (а) линии связи с ВУ и (б) линии связи с системной шиной МПС.

а)

DRQ[0:3] - запросы от ВУ (каналов) на прямой доступ в ЗУ;

DACK[0:3] - разрешения на ПДП внешним устройствам;

TC - конец счета, сигнал Н-уровня на этом выходе отмечает последний цикл передачи блока данных;

MARK - маркер - сигнал Н-уровня указывает, что до конца передаваемого блока необходимо выполнить число циклов обмена, кратное 128.

б) Линии связи с системной шиной передают стандартные сигналы управления, которые используются традиционно : RESET, READY (контроллер анализирует состояние готовности и при необходимости, как и МП, вставляет такты ожидания  $T_w$ ), RDM\, WRM\, RDIO\, WRIO\, HRQ, HLDA. Контроллер синхронизируется внешним системным синхросигналом, в качестве которого чаще всего используется Ф2тл. Сигнал ADSTB не поступает на системную шину и служит для "защелкивания" старшего байта адреса, передаваемого по шине данных в режиме ПДП (см. ниже). Сигнал AEN отмечает время действия цикла ПДП и может быть использован в системе для предотвращения захвата шины другими устройствами. Вход CS\ обеспечивает выбор микросхемы при адресной селекции.

Перед началом работы контроллера по сигналу RESET очищаются регистры всех каналов и регистр режимов RgP. Очистка RgP приводит к запрету работы всех каналов во всех режимах и предотвращает конфликты на шинах при подаче напряжения питания. Разрешение работы каналов не должны включаться, пока в регистры адресов и управления соответствующих каналов не будут загружены необходимые числа. В противном случае запрос DRQ от ВУ может вызвать циклы ПДП и порчу данных в памяти.

**Контроллер ПДП может работать в двух режимах:** в режиме программирования и в режиме ПДП.

В режиме программирования контроллер является пассивным устройством и процессор осуществляет с ним связь как с обычным ВУ. При этом линии A[3:0] и RDIO\, WRIO\ являются входными. Линии адреса A[2:1] при A[3] = 0 используются для селекции канала (код соответствует номеру канала), а тип регистра в канале идентифицируется разрядом A[0]: "0" - RgA, "1" - RgY. Поскольку RgA и RgY - двухбайтовые регистры, их загрузка осуществляется двумя циклами записи по одному адресу, причем в первом цикле передается младший байт.

При A[3:0] = 1000 осуществляется запись в RgP или чтение из RgC.

Разряды 0..3 регистра режимов RgP разрешают или запрещают работу соответствующих каналов ("1" - разрешено, "0" - запрещено).

Разряд  $RgP[4]$  устанавливает тип приоритета работы каналов. При  $RgP[4] = 0$  установлен фиксированный приоритет (канал 0 - высший, канал 3 - низший). При  $RgP[4] = 1$  устанавливается циклический приоритет каналов. В этом режиме после каждого цикла ПДП (но не запроса) приоритет каждого канала изменяется. Канал, который только что был обслужен, получает низший приоритет, а остальные каналы - по кольцу 0 - 1 - 2 - 3 - 0 -.

Циклический сдвиг приоритетов предотвращает монополизацию одного канала; последовательность циклов ПДП будет обслуживать различные каналы, если их работа не запрещена. Все операции начинаются с присвоения каналу 0 высшего приоритета.

Разряд  $RgP[5] = 1$  определяет режим опережающей записи. В этом случае длительность сигналов  $WRM\backslash$  и  $WRIO\backslash$  увеличивается за счет более ранней их активизации в цикле ПДП. Это в некоторых случаях позволяет обойтись без тактов ожидания  $T_w$ .

Состояние "1" в разряде  $RgP[6]$  определяет режим автоматического отключения канала по сигналу  $TC$ . Этот сигнал возникает когда  $RgYi[13:0] = 0$ ; в этом случае сбрасывается бит разрешения работы активного канала и запустить его вновь в работу можно только путем программной перезагрузки  $RgP$ . При  $RgP[6] = 0$  появление  $TC$  не запрещает дальнейшую работу канала.

При наличии "1" в  $RgP[7]$  устанавливается режим автозагрузки, позволяющий каналу 2 многократно передавать массив данных без программного вмешательства. При программировании канала 2 в этом режиме его параметры ( $RgA$  и  $RgY$ ) автоматически дублируются в соответствующих регистрах канала 3 и по сигналу  $TC$  переписываются из канала 3 в канал 2, восстанавливая первоначальное значение  $RgA2$  и  $RgY2$ , причем эта передача сопровождается установкой флага обновления  $UF$  в  $RgC[4]$ . Заметим, что возможности режима  $TCS$  (отключение каналов) не распространяются на канал 2 в режиме автозагрузки.

Если не запретить работу канала 3 в режиме автозагрузки, то он может работать как обычно при отсутствии запроса  $DRQ2$ . Однако, следует помнить, что работа канала 3 изменит значения параметров, которые должны передаваться в канал 2.

Перед началом работы каждый канал программируется путем занесения в  $RgA$  начального адреса массива, в  $RgY[13:0]$  - количества передаваемых байт, а в  $RgY[15:14]$  - режима работы канала (00 контроль, 01 - запись в ЗУ, 10 - чтение из ЗУ).

Режим контроля может использоваться ВУ для контроля принятых данных, т.к. в этом режиме не производится передача данных (не вырабатываются сигналы  $RDM\backslash$ ,  $RDIO\backslash$ ,  $WRM\backslash$ ,  $WRIO\backslash$ ), а все остальные функции ПДП сохраняются.

В режим ПДП контроллер переходит при поступлении запроса DRQ незамаскированного канала. Он вырабатывает запрос микропроцессору HRQ на захват шины и, получив подтверждение HLDA, передает сигнал активизации DACK\ выбранному ВУ и начинает циклы ПДП.

В цикле ПДП контроллер выдает содержимое PгА на линии A[7:0] (младший байт) и линии D[7:0] (старший байт в сопровождении строба ADSTB) и, в зависимости от режима работы канала, одну из пар управляющих сигналов RDM\ + WRIO\ или RDIO\ + WRM\ (в режиме контроля управляющие сигналы не выдаются). Режим ПДП завершается при снятии сигнала DRQ.

### **3.5. Клавиатура и индикация в МПС**

Устройства ввода/вывода МПС для АТП можно разделить на три разновидности:

1. УВВ для связи с объектом управления (ОУ);
2. УВВ для связи с ЦВМ другого (например, верхнего) уровня;
3. УВВ для связи с оператором.

Организация УВВ для связи с объектом управления целиком зависит от конкретного ОУ. В рамках данной главы остановимся на средствах связи МПС с оператором.

В многоуровневых МПС верхнего уровня обычно включают в себя разнообразные устройства ввода и вывода информации для оператора и поддерживающее их программное обеспечение. Сюда можно отнести клавиатуру, дисплей, принтер, а порой и более экзотические ВУ. На нижнем уровне МПС или в простейших МПС часто ограничиваются клавиатурой – ключевой или матричной и цифровой индикацией – двоичной или сегментной.

#### ***3.5.1. Двоичная индикация и ключи***

Подключение двоичной индикации для вывода информации и ключевых схем для ввода обычно сводится к тривиальному параллельному обмену с использованием буферных регистров, портов параллельного обмена или встроенных портов микроЭВМ.

Для устройств вывода в зависимости от вида конечного индикатора (светодиод, лампа накаливания, газоразрядный или жидкокристаллический индикатор, пьезокерамический динамик и др.) следует рассчитать соответствующую выходную цепь, обеспечивающую требуемые параметры нагрузки.



Для устройств ввода проблема согласования нагрузки обычно не стоит, здесь следует, прежде всего решить проблему защиты от дребезга (программно или аппаратно).

Встроенные порты параллельного обмена обычно бывают двунаправленными и допускают подключение, как ключей, так и двоичных индикаторов, однако, следует учитывать особенности схемотехники используемых портов.

Обмен с двоичными индикаторами осуществляется обычно в синхронном режиме.

### 3.5.2. Матричная клавиатура

Если клавиатура разрабатываемой МПС включает в себя большое число клавиш, целесообразно организовать ее в виде прямоугольной матрицы и идентифицировать клавишу номером строки и столбца матрицы.

На Рис.3.26 показан один из возможных способов подключения матричной клавиатуры к МПС. Регистр сканирования Рг.Скан. является устройством вывода и содержит унитарный код номера столбца. При опросе клавиатуры в этот регистр последовательно записываются коды, содержащие единицу только в одном разряде. Если нажата клавиша (замкнут контакт) в столбце, соответствующем единичному разряду Рг.Скан., то (с учетом инверсии выхода этого регистра) на линии строки, соответствующей нажатой клавише, установится потенциал «логический 0», на остальных линиях строк матрицы сохранятся «логические 1». Состояние строк матрицы через регистр ввода Рг.Клав. вводится в процессор. Позиции «1» в Рг.Скан. и «0» в Рг.Клав. однозначно определяют положение нажатой клавиши в матрице клавиатуры.

Для увеличения размерности матрицы клавиатуры можно использовать дешифратор на выходе Рг.Скан. и шифратор на входе Рг.Клав. (Рис.3.27). В этом случае возможно обслуживание матрицы клавиатуры размерностью до  $2^n \times 2^n$ , где  $n$  – разрядность регистров ввода и вывода.

Длительность нажатия клавиши обычно не бывает менее 0,1 сек. Для надежной фиксации нажатия необходимо за это время просканировать все столбцы матрицы. Очевидно, это не предъявляет слишком высоких требований к быстродействию МПС.

Проблема исключения дребезга клавиатуры может решаться на программном уровне, для чего достаточно после обнаружения нажатия клавиши (любой, пока можно не идентифицировать нажатую клавишу) осуществить программно задержку на время, превышающее длительность

дребезга<sup>1</sup>, и осуществить повторное сканирование клавиатуры с определением нажатой клавиши.

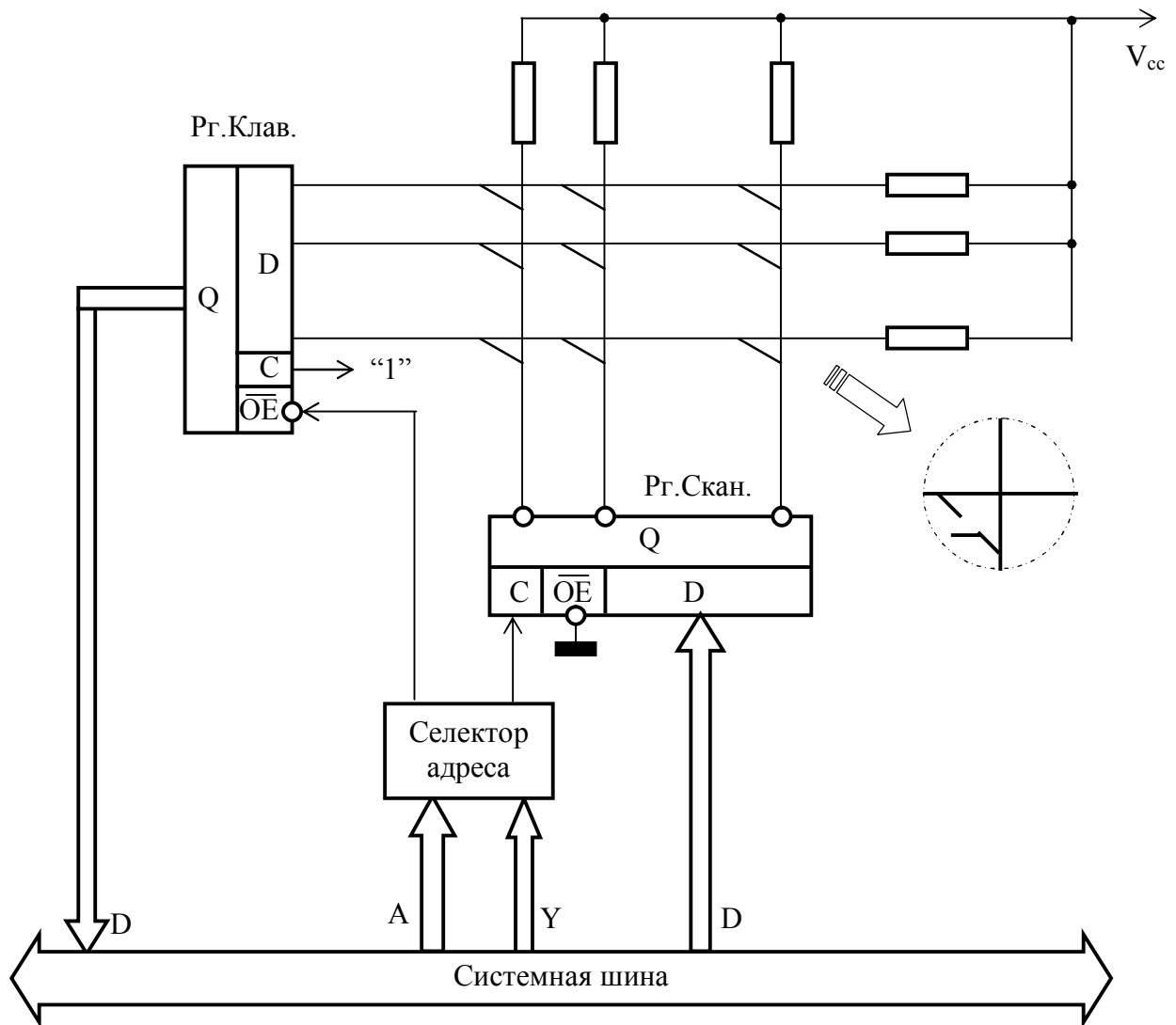
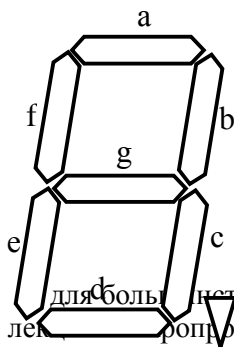


Рис.3.26. Подключение матричной клавиатуры

### 3.5.3. Сегментная индикация



Для индикации символов (прежде всего – десятичных цифр) в простейших МПС часто используется сегментная индикация. В этом случае каждому разряду (знакоместу) ставится в соответствие группа

для большинства контактов составляет единицы миллисекунд  
Курс лекции по процессорным устройствам автоматики

сегментов, реализованных каждый как светодиод, электролюминисцентный, газоразрядный или жидкокристаллический индикатор, причем у индикаторов одной группы имеется, как правило, общий электрод (анод или катод). Каждый сегмент разряда можно включить независимо от другого при условии, что на общий электрод будет подан соответствующий потенциал.

Количество сегментов разряда зависит от номенклатуры индицируемых символов, требуемого «качества» отображения и допустимых аппаратных затрат. Наибольшее распространение получили семисегментные индикаторы (восьмисегментные, восьмой электрод – запятая).

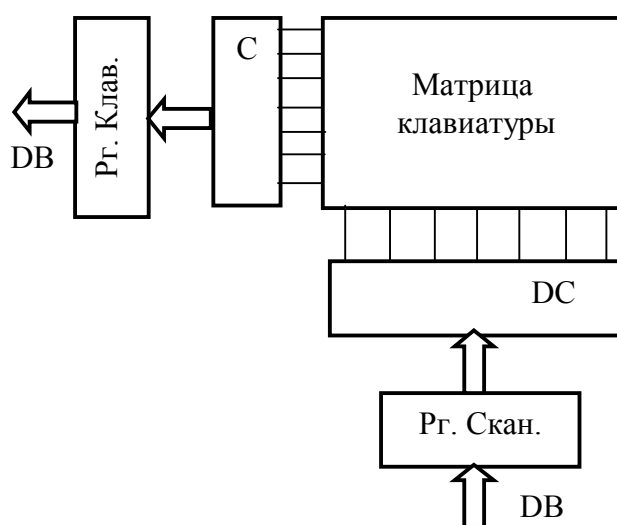


Рис.3.27. Использование шифратора и дешифратора для увеличения размерности матрицы клавиатуры

Различают два способа организации сегментной индикации – *статический* и *динамический*. Схема организации статической индикации приведена на рис 3.28 Для обслуживания каждого индикатора требуется собственный знакогенератор (ЗГ), преобразующий двоичный код символа в семисегментный. Обычно номенклатура индицируемых на семисегментном индикаторе символов не выходит за пределы десятичных (реже – шестнадцатеричных) цифр, поэтому для их кодирования достаточно четырехбитового поля<sup>2</sup>. Код символа поступает на вход знакогенератора (ЗГ) – комбинационной схемы или ПЗУ, преобразующей позиционный код символа в семисегментный.

<sup>2</sup> состояние сегмента запятой определяется в этом случае отдельной линией  
Курс лекций Микропроцессорные устройства автоматики

Рис.3.28. Статическая сегментная индикация

К недостаткам статической индикации можно отнести значительные затраты оборудования и большое число выводов (линий интерфейса), требуемых для ее обслуживания. Особенно это проявляется при большом числе индикаторов. В схеме рис. 3.28 можно исключить знакогенераторы, обеспечивая программное преобразование кодов символов. Однако такое решение приведет к значительному (почти двукратному) увеличению требуемого числа линий интерфейса.

Для сокращения числа линий интерфейса блока индикации и объема его оборудования при большом числе индикаторов в системе обычно используется динамическая индикация (Рис.3.29).





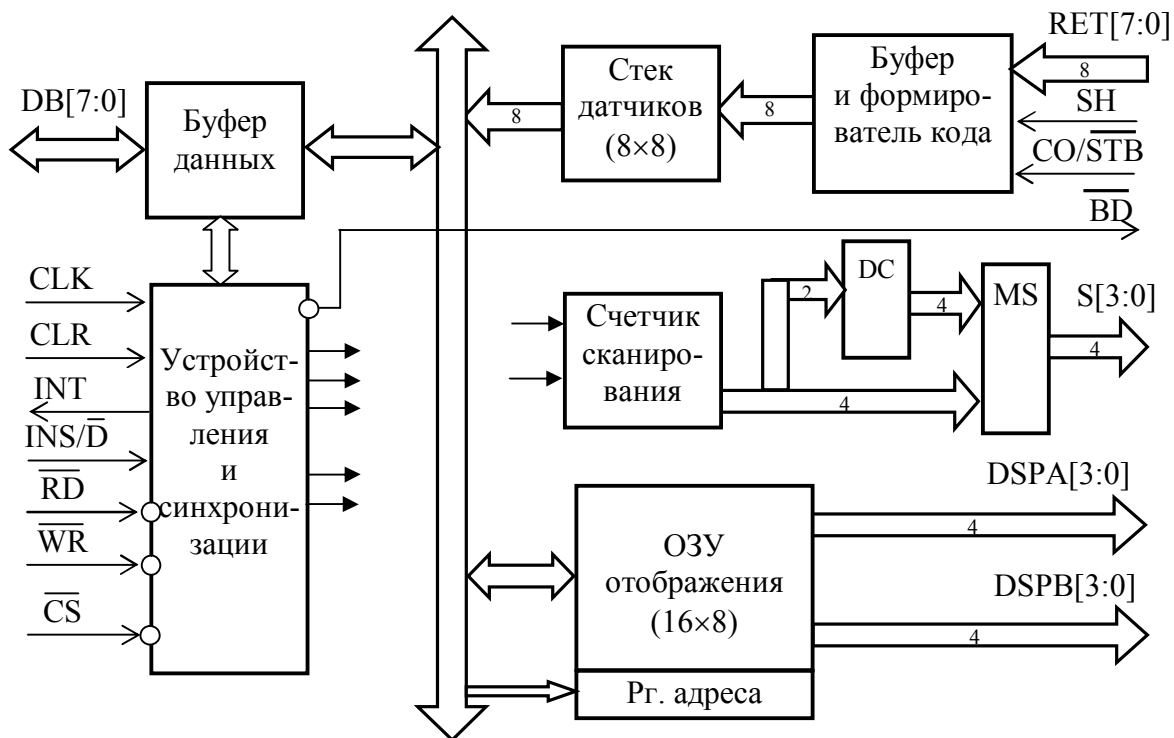
Рис.3.29. Динамическая индикация

При реализации динамической индикации все «одноименные» сегменты всех индикаторов объединяются гальванически и подключаются к выходам единственного регистра символов или единственного знакогенератора. Общие электроды индикаторов подключаются к источнику питания через ключи, управляемые регистром сканирования, причем в регистре сканирования должен размещаться унитарный код. Таким образом, в каждый момент времени активным является только один индикатор.

Микропроцессорная система должна программно обеспечить синхронную смену кода символа на входе знакогенератора и позиции символа в регистре сканирования. Для того, чтобы индицируемые символы на всех индикаторах воспринимались человеком как единое изображение, частота сканирования всего изображения должна составлять не менее 30 .. 50 Гц.

При работе динамической сегментной индикации МПС должна постоянно ее обслуживать, выводя коды и позиции символов. Для проведения других операций МПС может использовать промежутки времени между циклами индикации (около 20 мс) или, если этого времени недостаточно, останавливать индикацию на время проведения других операций. Если остановка индикации недопустима, а ресурсов времени недостаточно, можно использовать в МПС специальный контроллер, который берет на себя функции обслуживания динамической индикации, а заодно – и клавиатуры.

#### 3.5.4. Контроллер клавиатуры и индикации K580BB79



Контроллер K580BB79 (Рис.3.30) предназначен для обслуживания сегментной индикации (максимально – два блока по 16 индикаторов) и матричной клавиатуры (до четырех матриц размером 8×8 клавиш) или линейки двоичных датчиков (2×8). Структура контроллера приведена на рис. 5. Контроллер включает в себя следующие блоки:

- *ОЗУ отображения* объемом 16 8-разрядных слов, предназначенное для хранения кодов индицируемых символов;
- *ОЗУ датчиков* (клавиатуры) объемом 8 8-разрядных слов с *буферной схемой*, предназначенное для размещения кодов нажатых клавиш или состояний линейки датчиков;
- *четырёхразрядный счетчик сканирования с дешифратором* состояния двух младших разрядов;
- *буфер данных* для связи с МПС;
- *устройство управления и синхронизации*.

Рис.3.30. Контроллер клавиатуры и индикации K580BB79

Внешние выходы микросхемы имеют следующее назначение:

DB[7:0]	– системная шина данных;
CLK	– сигнал синхронизации (тактовая частота);
CLR	– сброс (установка микросхемы в исходное состояние);
INT	– выход запроса на прерывание;
INS/D	– вход идентификации типа данных:

H-уровень определяет запись команды или чтение слова состояния;

	L-уровень определяет запись или чтение данных;
RD	– вход строба чтения;
WR	– вход строба записи;
CS	– вход выбора микросхемы;
RET[7:0]	– входы линий возврата для приема сигналов от датчиков или матричной клавиатуры
SH	– вход сигнала идентификации для режима сканирования клавиатуры;
CO/STB	– то же / строб записи состояния датчиков в стек датчиков;
BD	– выход сигнала гашения индикаторов;
S[3:0]	– выходы разрядов счетчика или выходы дешифратора младших разрядов;
DSPA[3:0]	– выходы <u>старших</u> разрядов ОЗУ отображения;
DSPB[3:0]	– выходы <u>младших</u> разрядов ОЗУ отображения.

Контроллер позволяет обеспечить одновременное управление сегментными индикаторами и блоком матричной клавиатуры или линейкой датчиков. Схема подключения контроллера K580BB79 к МПС и блокам клавиатуры и индикации приведена на Рис.3.31.

Работа контроллера осуществляется следующим образом. МПС загружает в ОЗУ отображения коды выводимых на индикацию символов и больше не обращается к контроллеру до тех пор, пока не потребуется изменить выводимую на индикацию информацию.

Контроллер инкрементирует счетчик (с частотой CLK/f, где f – один из 16 программируемых коэффициента деления) и синхронно выводит на линии DSP содержимое ячейки ОЗУ отображения, адрес которой соответствует значению счетчика. При работе с внешним дешифратором возможно управление 16 сегментными индикаторами при условии, что в ячейках ОЗУ отображения хранятся семисегментные коды символов.

Если хранить в ОЗУ отображения двоичные позиционные коды символов, то в каждой ячейке можно разместить коды двух символов и, при наличии внешних знакогенераторов, можно управлять 32 индикаторами.

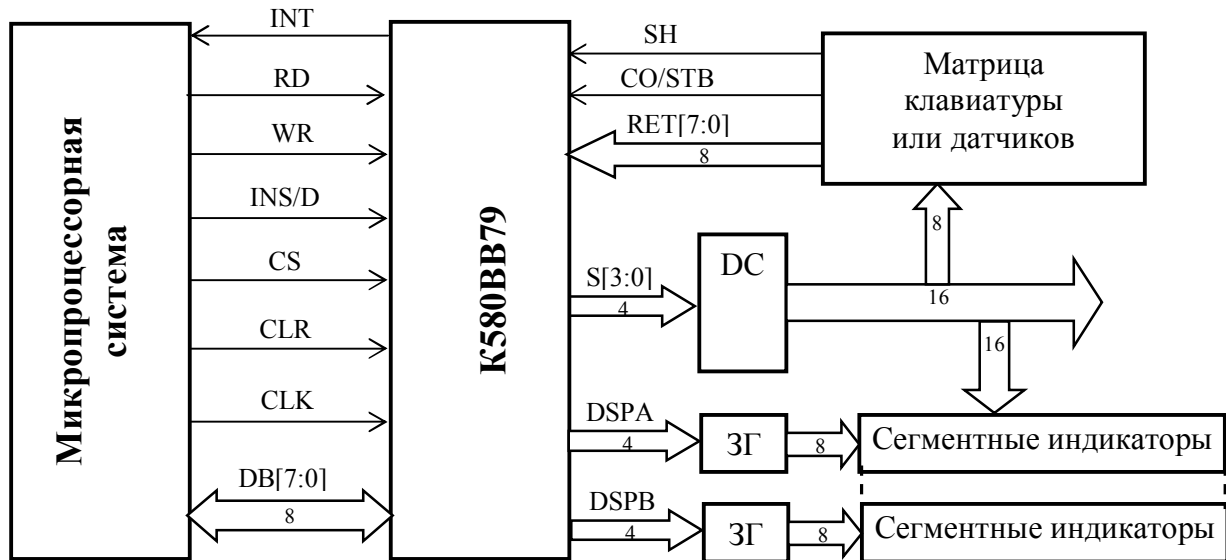
Рис.3.31. Подключение контроллера K580BB79 в составе МПС

Восемь младших выходов дешифратора можно одновременно использовать для сканирования матрицы клавиатуры или матрицы датчиков. В первом случае контроллер в каждом такте работы счетчика проверяет состояние линий RET и при обнаружении ненулевого разряда помещает в стек FIFO код нажатой клавиши, который формируется в следующем формате:

7	6	5	4	3	2	1	0
CO/STB	SH	Номер строки			Номер столбца		

Рис.3.32. Формат кода нажатой клавиши

где номер строки матрицы клавиатуры определяется как номер позиции единицы в слове RET, а номер столбца – состоянием трех младших разря-



дов счетчика. Два старших бита копируют состояние соответствующих входных линий, которые таким образом могут идентифицировать одну из четырех матриц клавиатуры 8×8. Занесение байта в стек сопровождается выработкой сигнала прерывания INT.

В режиме сканирования матрицы датчиков состояние входов RET по-байтно вводится в стек с частотой сканирования или по внешнему стробу, подаваемому на вход CO/STB.

Управление режимами работы контроллера осуществляется с помощью системы команд, включающую 8 однобайтовых команд, которые могут записываться в контроллер по шине данных при высоком уровне на



входе INS/D. Операция МПС при обращении к контроллеру определяются набором управляющих сигналов (Таблица 3.6).

Таблица 3.6

CS	RD	WR	INS/D	Операция
1	x	x	x	Z-состояние
0	1	0	1	Запись команды в контроллер
0	1	0	0	Запись данных в контроллер
0	0	1	1	Чтение слова состояния контроллера
0	0	1	0	Чтение данных из контроллера

Все команды контроллера содержат в трех старших разрядах код операции, а пять младших бит используются для задания параметров команд. Ниже приводятся **форматы команд контроллера K580BB79**.

1. Команда установки режимов:

7	6	5	4	3	2	1	0
0	0	0	Режим дисплея	Режим клавиатуры			

Режим дисплея		Режим клавиатуры	
00	8-разрядный дисплей, ввод слева	000	Кодированное сканирование клавиатуры с обнаружением двойных нажатий
01	16-разрядный дисплей, ввод слева	001	Дешифрованное сканирование клавиатуры с обнаружением двойных нажатий
10	8-разрядный дисплей, ввод справа со сдвигом	010	Кодированное сканирование клавиатуры с обнаружением N нажатых клавиш
11	16-разрядный дисплей, ввод справа со сдвигом	011	Дешифрованное сканирование клавиатуры с обнаружением N нажатых клавиш
		100	Кодированное сканирование матрицы датчиков
		101	Дешифрованное сканирование матрицы датчиков
		110	Ввод по стробу, кодированное сканирование дисплея
		111	Ввод по стробу, дешифрованное сканирование дисплея

Термины, используемые при описании режимов микросхемы, имеют следующий смысл:

- *кодированное сканирование* предполагает выдачу на выходы S[3:0] состояние разрядов счетчика сканирования;
- *дешифрованное сканирование* обеспечивает выдачу на выходы S[3:0] унитарного кода дешифрации двух младших разрядов счет-

чика сканирования. Такой режим можно использовать, если число объектов сканирования не превышает четырех (4 индикатора и матрица клавиатуры 4×8);

- *сканирование клавиатуры* (режимы 0 .. 3) обеспечивает занесение в стек датчиков кодов нажатых клавиш в формате Рис.3.3;
- *обнаружение двойных нажатий* (и более) игнорируются, никакой код в стек датчиков не вводится;
- *обнаружение N нажатых клавиш* обеспечивает ввод в стек кодов всех обнаруженных нажатых клавиш последовательно, в порядке сканирования;
- *сканирование матрицы датчиков* позволяет поместить по внутреннему тактовому сигналу состояние входного слова RET[7:0] в стек датчиков;
- *ввод по стробу* помещает RET[7:0] в стек датчиков по внешнему стробу на входе CO/STB, причем по его срезу информация фиксируется внутренними схемами БИС, а по фронту – переписывается в стек. Время между срезом и фронтом строба должна быть не менее периода тактового импульса;
- *8-разрядный дисплей* реализует последовательный вывод на линии DSP только восьми первых ячеек ОЗУ отображения в соответствии со значением трех младших разрядов счетчика сканирования (значение четвертого разряда при адресации ОЗУ принимается равным 0);
- *16-разрядный дисплей* реализует последовательный вывод всех 16 ячеек ОЗУ отображения в соответствии со значением счетчика сканирования;
- *ввод слева*<sup>3</sup> – как на обычных алфавитно-цифровых дисплеях;
- *ввод справа со сдвигом*<sup>1</sup> – как у микрокалькулятора.

## 2. Команда программирования частоты синхронизации

7	6	5	4	3	2	1	0
0	0	1	Коэффициент деления частоты CLK				

Частота сканирующих импульсов после сброса по сигналу CLR автоматически устанавливается 1/31 от частоты синхронизации БИС. С помощью команды "Программирование синхронизации" коэффициент деления может быть установлен пользователем от 2 до 31. Частота сканирования выбирается так, чтобы период сканирования был больше удвоенной длительности дребезга.

---

<sup>3</sup> Заданный порядок учитывается только при автоинкрементной записи в ОЗУ отображения.

Следующие три команды должны применяться перед обращением к ОЗУ отображения или чтением из ОЗУ-стека датчиков. Команды определяют адрес и вид будущего обращения. В случае, когда требуется обращение в последовательные ячейки памяти, достаточно предварительно сформировать одну команду с начальным адресом и установленным битом А – автоинкремента (бит 4 в командах 3, 4, 5).

### 3. Чтение ОЗУ датчиков

7	6	5	4	3	2	1	0
0	1	0	A	x	Адрес		

### 4. Чтение ОЗУ отображения

7	6	5	4	3	2	1	0
0	1	1	A	Адрес			

### 5. Запись ОЗУ отображения

7	6	5	4	3	2	1	0
1	0	0	A	Адрес			

### 6. Гашение – запрет записи отображения

7	6	5	4	3	2	1	0
1	0	1	x	D3	D2	D1	D0

D3 = 1 – запрет записи в ОЗУ отображения по входам D7..D4 для канала А;

D2 = 1 – запрет записи в ОЗУ отображения по входам D3..D0 для канала В;

D1 = 1 – гашение выходов DSPAi;

D0 = 1 – гашение выходов DSPBi.

С помощью команды "Гашение - запрет записи" пользователь может организовать запись в ОЗУ как целыми байтами, так и отдельными тетрадами:

### 7. Сброс

7	6	5	4	3	2	1	0
1	1	0	x	Режим сброса ОЗУ дисплея	D1	D0	

D0 = 1 – программный сброс;

D2 = 1 – сброс слова состояния и сигнала прерывания;

Режим сброса ОЗУ дисплея определяет, в какое состояние при сбросе будут установлены ячейки ОЗУ отображения: 00h, 20h или FFh или какое значение будет удерживаться на выходах DSPA, DSPB (Таблица 3.7).

Таблица 3.7

D4	D3	D2	Код в ОЗУ отображения	D4	D3	D2	DSPA	DSPB
0	0	x	0 0 0 0 0 0 0 0	0	0	x	0 0 0 0	0 0 0 0
0	1	0	0 0 1 0 0 0 0 0	0	1	0	0 0 1 0	0 0 0 0
0	1	1	1 1 1 1 1 1 1 1	0	1	1	1 1 1 1	1 1 1 1

### 8. Сброс прерывания – установка режима обнаружения ошибки

7	6	5	4	3	2	1	0
1	1	1	D4	x	x	x	x

Команда сбрасывает запрос на прерывание и, кроме того, при D4 = 1 устанавливает режим обнаружения ошибки в клавиатурных режимах, а при D4 = 0 – сбрасывает этот режим.

Состояние контроллера характеризует **слово состояния**, формат которого имеет следующий вид:

7	6	5	4	3	2	1	0
D7	D6	D5	D4	Число символов в ОЗУ датчиков			

D7 = 1 – доступ в ОЗУ отображения прекращен;

D6 = 1 – ключ замкнут (режимы матрицы датчиков) / ошибка двойного нажатия (режим клавиатуры с обнаружением ошибок);

D5 = 1 – ошибка переполнения;

D4 = 1 – ошибка переопустошения.

### 3.5.5. Работа контроллера K580BB79

Контроллер состоит из двух слабо связанных между собой частей – дисплейной и клавиатурной, программирование режимов которых осуществляется независимо.

#### Управление клавиатурой

Выше отмечалось, что поле [2:0] команды 1 установки режимов определяет режимы работы клавиатурной части контроллера. На самом деле младший разряд команды управляет мультиплексором линий S[3:0] и следовательно, одинаково влияет на сканирование клавиатуры (датчиков) и индикации.

D0 = 0 – кодированное сканирование, на линии S[3:0] выдаются значения разрядов счетчика сканирования;

D0 = 1 – дешифрованное сканирование, на линии S[3:0] выдаются выходы дешифратора двух младших разрядов счетчика сканирования (унитарный код).

Разряд D2 команды 1 установки режимов определяет режим работы клавиатурной части: D2= 0 – режим клавиатуры; D2= 1 – режим датчиков.

**Режим клавиатуры** предполагает фиксацию нажатия клавиши и размещение в стеке кода этой клавиши в формате Рис.3.3, причем разряд D1 команды 1 установки режимов определяет, будут ли фиксироваться коды

одновременно нажатых нескольких клавиш (см. описания формата команды 1). В режиме *с обнаружением двойных нажатий* двух- и более кратные нажатия будут просто игнорироваться и никакой код не будет введен в ОЗУ датчиков.

В режиме *с обнаружением N нажатых клавиш* при нажатии нескольких клавиш все они опознаются и их коды (Рис.3.3) вводятся в стек датчиков в порядке сканирования, однако, если предварительно прошла команда 8, такая ситуация будет вызывать установку флага ошибки – разряда D6 слова состояния..

**Режим датчиков** имеет две разновидности. *Режим матрицы датчиков* позволяет вводить в стек строку RET[7:0], которая соответствует состоянию сканируемой строки матрицы датчиков. При этом в стеке не сохраняется информация о позиции строки. Входы SH и CO/STB в этом режиме не используются. К линиям RET[7:0] можно подключать не только ключи, но и любую логическую схему, которой можно управлять с помощью выходов сканирования S[3:0]. Например, восемь мультиплексированных каналов ввода, которые сканируются с помощью БИС K580BB79. Информация, считанная из ОЗУ датчиков, будет инверсной.

В режиме *ввода по стробу* состояние входов RET[7:0] записывается в стек – ОЗУ датчиков по стробирующему импульсу на входе CO/STB, длительность которого должна быть не менее одного периода сканирования. Вход SH в этом режиме не используется.

Стек – ОЗУ датчиков предназначен для хранения с последующим считыванием кодов клавиш, состояния ключей в матрице датчиков, а также информации, вводимой по стробу. Блок памяти 8×8 бит выполняет две функции. В режимах сканирования клавиатуры или ввода по стробу память работает как магазин с дисциплиной FIFO (ввод в последовательные ячейки и последующее считывание в том же порядке). Если стек не пуст, формируется сигнал запроса на прерывание.

В режиме сканирования матрицы датчиков память работает как обычное ОЗУ, когда каждая ячейка ОЗУ загружается состоянием соответствующей (по счетчику сканирования) строки матрицы датчиков, причем запрос INT устанавливается только в том случае, если обнаружено изменение состояния датчиков.

Специальная схема отслеживает текущее состояние ОЗУ датчиков – формирует значение числа символов в стеке в поле [3:0] слова состояния и устанавливает флаги переполнения и переопустошения стека, а также формирует запрос прерывания INT.

### ***Управление дисплеем***

ОЗУ дисплея объемом 16 байт можно организовать в вдвоенное ОЗУ 2×16×4, используя команду 6 «Гашение – запрет записи отображения».

Команда 7 «Сброс» позволяет устанавливать все ячейки ОЗУ при очистке в состояние 00h, FFh или 20h. Регистр адреса ОЗУ отображения устанавливается в 0 по сбросу (аппаратному или программному), загружается произвольным значением командами 4 и 5 и инкрементируется при каждом обращении в ОЗУ со стороны процессора, если включен флаг автоинкремента, устанавливаемый теми же командами.

В режиме 8-разрядного дисплея сканируется только 8 ячеек ОЗУ с младшими адресами. Если запрограммирован режим дешифрованного сканирования, то будут воспроизводиться на выходах DSP только четыре младшие ячейки.

В контроллере K580BB79 предусмотрено два режима ввода информации в ОЗУ дисплея. *Ввод слева* соответствует простейшему формату отображения, когда каждой позиции дисплея соответствует определенная ячейка ОЗУ, а именно – адресу 0 соответствует крайний слева символ дисплея, а адресу 7 (15) – крайний справа. Ввод символов начиная с нулевого адреса вызывает заполнение дисплея слева направо, причем символ 8 (16) будет вводиться снова в крайнюю левую позицию и т.д. Для того, чтобы заменить один символ в существующем изображении, достаточно переписать содержимое одной соответствующей ячейки памяти.

*Ввод справа со сдвигом* принят в калькуляторах. Всякий вновь вводимый символ помещается в крайнюю правую позицию, предварительно все изображение сдвигается на один символ влево, а крайний левый символ теряется.

В контроллере предусмотрен выходной сигнал BD – гашение отображения. Он формируется длительностью не менее 150 мкс для гашения изображения на момент смены символов или при поступлении команды 6 «Гашение – запрет записи отображения». Если в этой команде  $D0 = D1 = 0$ , то сигнал BD равен нулю в течение всего действия команды, если  $D0 \oplus D1 = 1$ , то сигнал BD равен нулю в течение не менее 150 мкс.

### 3.6. Организация процессорного модуля и системного интерфейса

Для обеспечения функционирования микропроцессора к нему обязательно следует подключить тактовый генератор и буферные схемы, увеличивающие нагрузочную способность системных шин.

Системная шина управления может быть в простейшем случае образована двумя выходными сигналами МП - RD и WR. Однако, в большин-

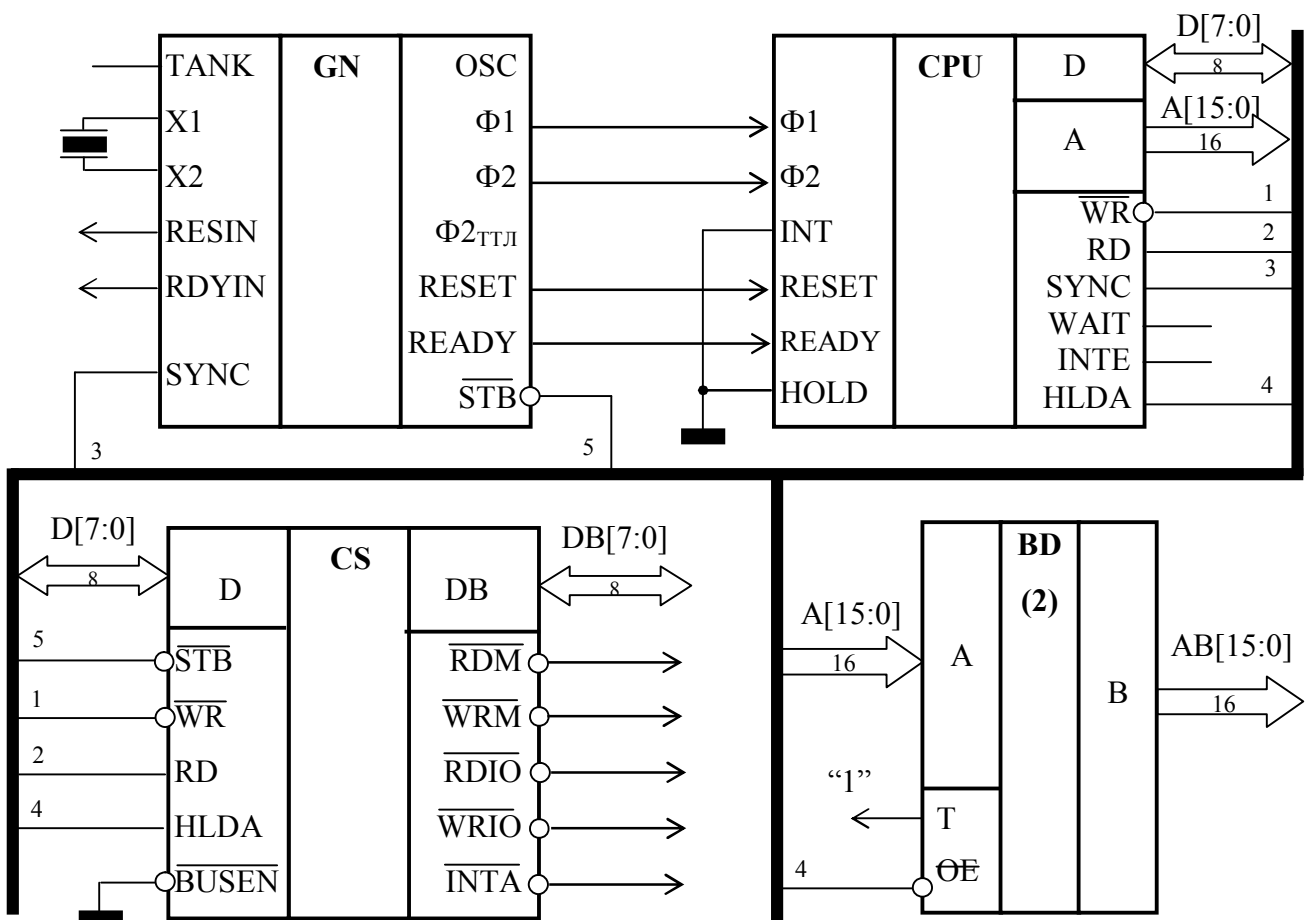
стве МПС фиксируются разряды (все или часть) PSW и на их основе формируется расширенный состав линий управления.

Типичная структура процессорного модуля показана на Рис. 3.33, в схеме которого, наряду с МП К580ВМ80, использованы СИС серии К580:

Рис. 3.33. Структура процессорного модуля на базе микропроцессора i8080

генератор тактовых импульсов К580ГФ24, системный контроллер К580ВК28 и шинные формирователи К580ВА86.

**Тактовый генератор. .ГФ24 (Ошибка! Источник ссылки не найден.34),** наряду с выработкой тактовых сигналов  $\Phi_1$ ,  $\Phi_2$ , стробирует внеш-



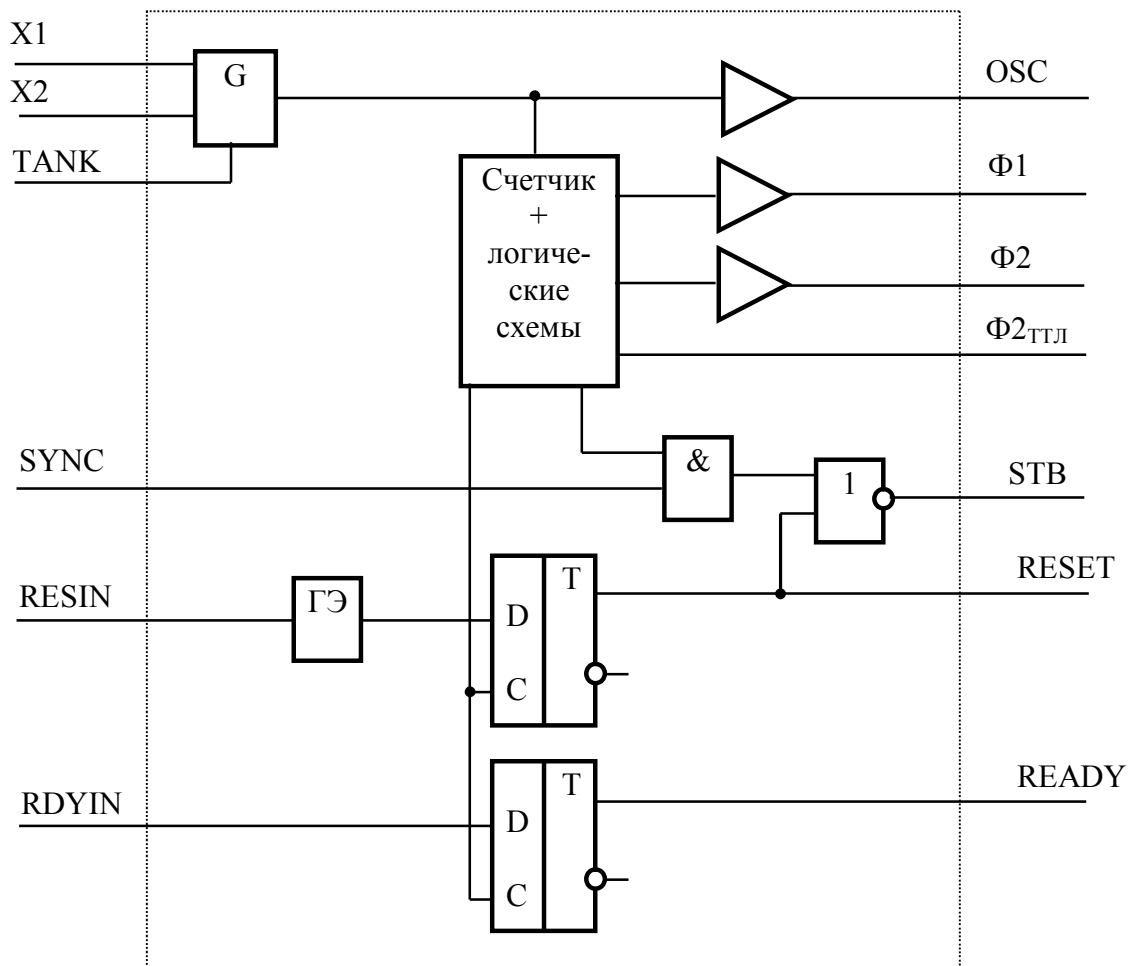
ние асинхронные сигналы RESIN, RDYIN по фазе  $\Phi_2$  и формирует сигнал строга  $STB\setminus$  для защелкивания PSW:  $STB\setminus = SYNC \& \Phi_1$ . Сигналы  $\Phi_1$ ,  $\Phi_2$  имеют амплитуду 12В; кроме того выдается сигнал  $\Phi_{2ТТЛ}$  уровня ТТЛ-схем.

**Системный контроллер. .BK28** (Рис. 3.335) буферизирует шину данных, образуя системную шину данных DB[7:0] с нагрузочной способностью порядка 30 нагрузок ТТЛ. В состав .BK28 входит регистр-защелка, фиксирующий 5 разрядов PSW (0,1,4,6,7) по стробу STB\, и логическая схема, формирующая сигналы на 5 управляющих линиях системного интерфейса:

- RDM\ - чтение из памяти;
- WRM\ - запись в память;
- RDIO\ - чтение из внешнего устройства;
- WRIO\ - запись во внешнее устройство;
- INTA\ - подтверждение прерывания.

Входной сигнал HLDA высоким уровнем переводит все выходы системного контроллера в высокоимпедансное состояние, то же делает и внешний сигнал BUSEN\ - "разрешение работы шины".

Рис. 3.34. Тактовый генератор K580ГФ24





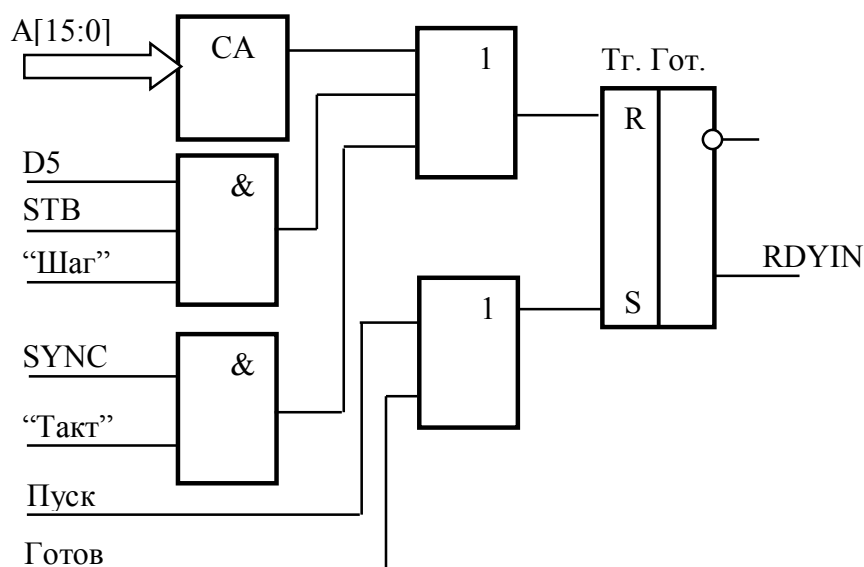


Рис. 3.36. Системный контроллер K580BK28

Шинные формирователи. BA86 буферизуют однонаправленную шину адреса, а сигнал HLDA, поступающий на вход OE\ переводит высоким уровнем системную шину AB[15:0] в высокоимпедансное состояние.

Внешний сигнал RDYIN формируется обычно на специальном триггере, что позволяет реализовать, с одной стороны – асинхронный режим обмена микропроцессора с «медленными» устройствами, с другой стороны – пошаговый и потактовый режимы работы системы. На рисунке ↑ сигнал готовности RDYIN сбрасывается: (1) селектором адреса «медленного» устройства; (2) сигналом  $M1 = D5 \& STB$  в пошаговом режиме; (3) сигналом SYNC в потактовом режиме. Установка RDYIN осуществляется сигналом готовности устройства или кнопкой «Пуск».

### 3.7. Структура микроЭВМ

Общая структура микроЭВМ показана на рис. 3.37, где ЦП – микропроцессор, ПЗУ – постоянная память для хранения программ, ОЗУ – оперативная память, УВВ – устройства ввода-вывода, Г – генератор тактовых импульсов. Отдельные модули соединены с помощью шины данных, шины адреса и шины управления. Количество модулей можно увеличивать, т.е. структура открыта и допускает расширение.

Процесс обработки данных в микроЭВМ представляет собой последовательность дискретных состояний рассматриваемой структуры. Ряд состояний образует определенный машинный цикл. Процесс выполнения каждой команды рабочей программы распадается на последовательность машинных циклов.

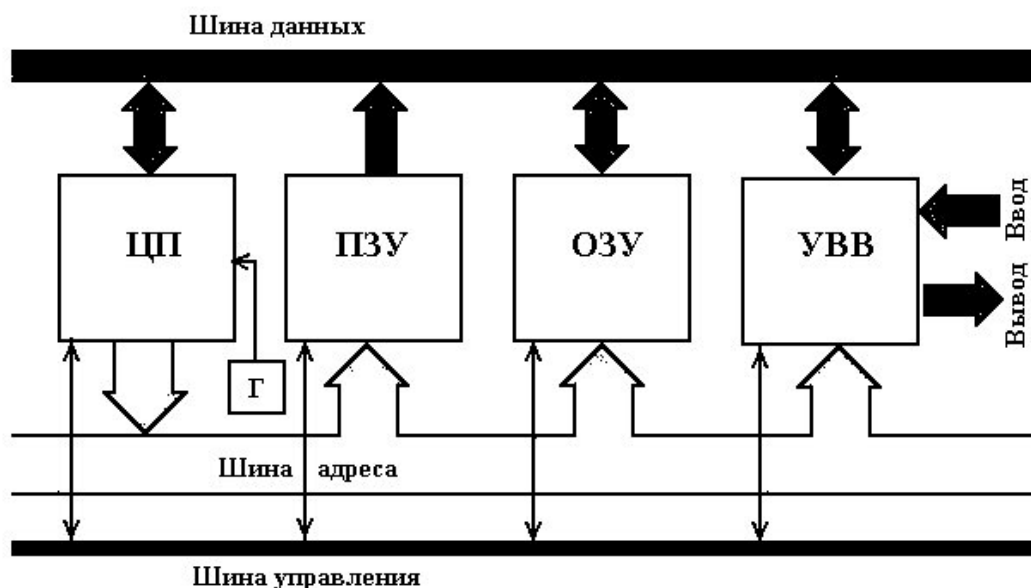


Рис. 3.37. Структура микроЭВМ

Переход микроЭВМ из одного состояния в другое происходит под воздействием тактовых импульсов, поступающих на тактовые входы микропроцессора. Для генерации последовательности тактовых импульсов в структуре микроЭВМ используется генератор Г тактовых импульсов, который вырабатывает последовательности прямоугольных импульсов. Частота тактовых импульсов для микропроцессора КР580ИК80А составляет 2 МГц и, следовательно, длительность одного состояния микроЭВМ равна 0,5 мкс.

Машинный цикл может включать разное число состояний, как и команда, может состоять из разного числа машинных циклов. Следовательно, длительность выполнения команды зависит от ее вида и необходимого числа последовательных состояний микроЭВМ при выполнении команды.

Работу микроЭВМ удобно рассмотреть на примере выборки из ячейки памяти очередной команды программы, с которой начинается цикл любой команды. Для 8-разрядной микроЭВМ с микропроцессором КР580ВМ80А процесс выборки команды состоит из одного машинного цикла, включающего три состояния.

1. Содержимое счетчика команд передается в регистр адреса.
2. Содержимое счетчика команд увеличивается на единицу (адрес следующей команды).
3. Код адреса передается в селектор ячеек памяти программ, подключает нужную ячейку к шине данных, и по шине данных код команды

переносится в регистр временного хранения информации ТЕМ и в регистр команды.

Поскольку выборка команды совершается за три состояния, то эта операция длится 1,5 мкс. Поступивший в регистр команды код дешифрируется, и дальнейшие состояния микроЭВМ определяются устройством управления в зависимости от выполняемой команды. Выборка команды завершается за один машинный цикл, и с этого цикла начинается выполнение каждой очередной команды программы.

#### 8-разрядная микроЭВМ

МикроЭВМ на основе комплекта К580 строится по модульному принципу, который обеспечивает подключение к единой магистрали модулей ЦП (центрального процессора), ЗУ (запоминающего устройства) и УВВ (устройства ввода-вывода).

Принципиальная схема микроЭВМ приведена на рис. 3.38. В микроЭВМ используются микропроцессор КР580ВМ80, микросхема системного

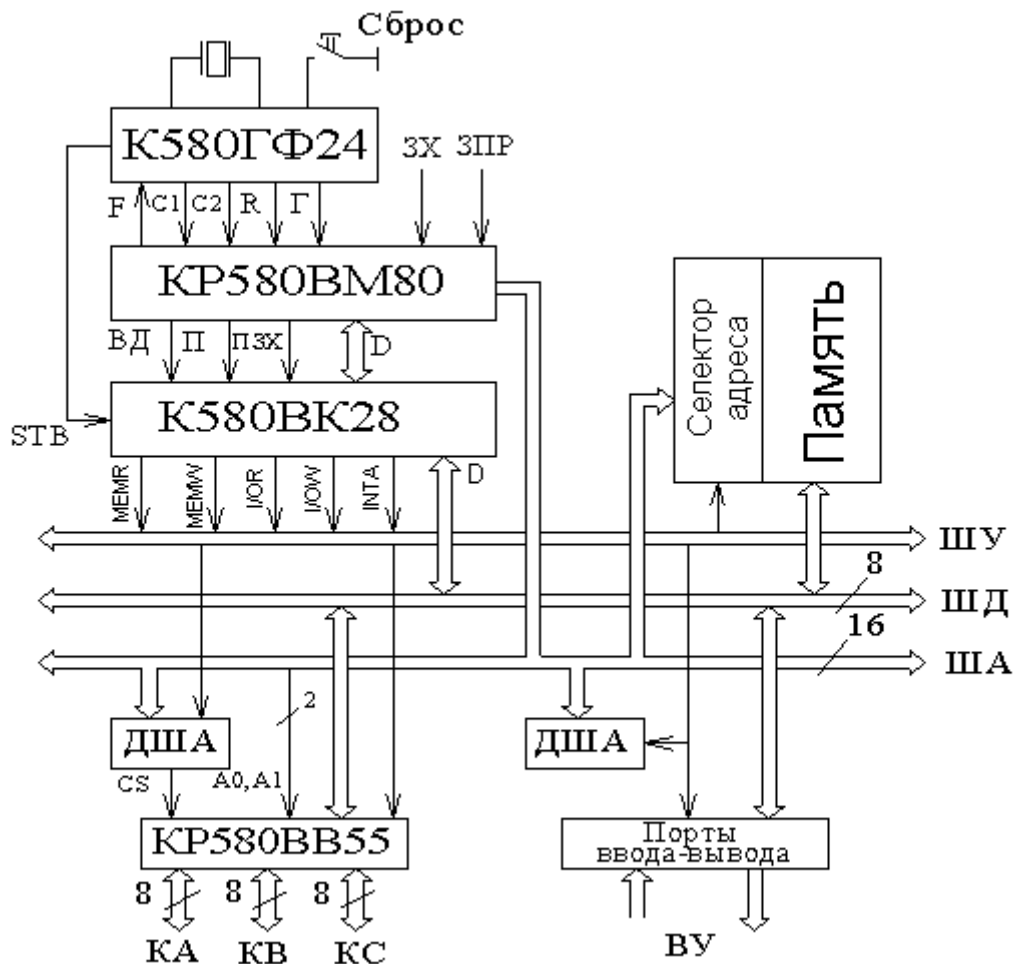


Рис. 3.38. 8-разрядная микроЭВМ

генератора тактовых импульсов К580ГФ24, микросхема системного контроллера К580ВК28, микросхема программируемого параллельного адаптера КР580ВВ55, а также дешифраторы адреса ДША и модуль памяти.

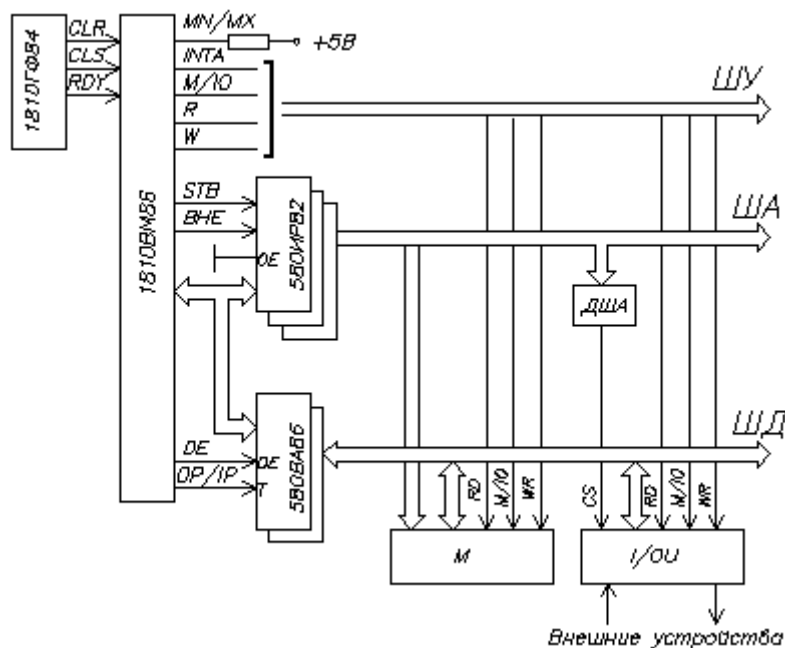
Генератор тактовых импульсов К580ГФ24 предназначен для синхронизации микропроцессорной системы и вырабатывает тактовые импульсы частотой 2 МГц для синхронизации МП, а также других схем (в последнем случае используются тактовые импульсы ТТЛ уровней). Одновременно эта микросхема обеспечивает формирование некоторых управляющих сигналов (например, сигнал сброса R).

Системный контроллер КР580ВК28 предназначен для фиксации слова-состояния МП, выработки системных управляющих сигналов, буферизации шины данных МП и управления направлением передачи по шине данных.

Внешние устройства ВУ подключаются к микроЭВМ через порты ввода-вывода. Внешними устройствами могут быть клавиатура, монитор, накопитель на гибком магнитном диске, объекты управления и др. На схеме показана реализация трех портов ввода-вывода с использованием программируемого параллельного адаптера КР580ВВ55. Адаптер обеспечивает три порта КА, КВ и КС, через которые 8-разрядные двоичные коды могут как вводиться в ЭВМ, так и выводиться из нее.

#### 16-разрядная микроЭВМ

Основу таких ЭВМ составляют 16-разрядные микропроцессоры. Широкое распространение получили микропроцессоры i8086, аналогом которых являются отечественные микропроцессоры К1810ВМ86. Микропроцессорный комплект БИС К1810 предназначен как для построения простейших одноплатных микроЭВМ общего назначения, так и для мульти-



процессорных систем. В состав комплекта входят микропроцессор и вспомогательные БИС, позволяющие строить ЭВМ различной конфигурации. Комплект К1810 совместим с комплектом КР580.

Схема построения 16-разрядной микроЭВМ на основе микропроцессора К1810ВМ86 показана на рис. 3.39. Для син-

Рис. 3.39. 16-разрядная микроЭВМ

хронизации работы микропроцессора использована БИС генератора тактовых импульсов 1810ГФ84. Генератор выдает последовательность тактовых импульсов CLS, а также формирует сигнал сброса микропроцессора CLR и сигнал готовности внешнего устройства RDY.

Микропроцессор используется в минимальной конфигурации, что обеспечивается подачей потенциала 5 В на вход выбора конфигурации MN/MX. Сигналы микропроцессора INTA (разрешение прерывания), M/IO (обмен с памятью/устройствами ввода-вывода), R (чтение данных) и W (запись данных) образуют шину управления ШУ микроЭВМ.

Для формирования шины адреса ША разрядностью 20-бит используются три 8-разрядных буферных регистра К580ИР82. Это параллельные регистры с тристабильными выходами. Они используются для фиксации адреса, выдаваемого микропроцессором. Запись соответствующих байтов адреса в регистры управляется сигналами микропроцессора STB (строб адреса) и VHE (разрешение старшего байта). Регистры адреса постоянно включены, что обеспечивается подачей логического нуля на вход выбора микросхем ОЕ.

16-разрядная шина данных формируется с помощью шинных формирователей (микросхемы КР580ВА86). Эти микросхемы являются 8-разрядными параллельными двунаправленными приемопередатчиками с тристабильными выходами. Управляются микросхемы сигналами микропроцессора DE (разрешение передачи данных) и OP/IP (направление передачи данных). Последний сигнал поступает на вход T выбора направления передачи данных через шинный формирователь.

Буферные регистры и шинные формирователи использованы в схеме микроЭВМ для повышения нагрузочной способности шин. К системным шинам присоединяются модуль памяти М и регистры ввода-вывода данных I/OU. Ячейки памяти адресуются по шине адреса. Управляется память системными сигналами RD (чтение), WR (запись) и M/IO (память/устройства ввода-вывода).

Для адресации устройств I/OU ввода-вывода по шине адреса использован дешифратор адреса ДША. При обращении к адресуемому порту ввода-вывода на выходе ДША генерируется сигнал CS выбора микросхемы, который включает регистр порта. Управление портами ввода-вывода производится теми же сигналами, что и памятью. Передача данных между микропроцессором и памятью, а также устройствами ввода-вывода происходит по шине данных ШД.