

Министерство образования Российской Федерации
Южно-Уральский государственный университет
Кафедра автоматизации механосборочного производства

681.325(07)
С506

Смирнов В.А.

СХЕМОТЕХНИКА МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Текст лекций

Челябинск
Издательство ЮУрГУ
2003

УДК 681.325.5.181.4(075.8)

Смирнов В.А. Схемотехника микропроцессорных систем: Текст лекций. — Челябинск: Изд-во ЮУрГУ, 2003. — 110 с.

В тексте лекций изложены вопросы построения микропроцессорных систем управления технологическим оборудованием. Показана общность подходов вне зависимости от используемых микропроцессоров или однокристальных микро-ЭВМ. Рассмотрено подключение к микропроцессорной системе типовых устройств систем управления.

Текст лекций предназначен для студентов специальности 210200 «Автоматизация технологических процессов и производств», может быть полезен студентами родственных специальностей.

Ил. 58, табл. 16, список лит. — 10

Одобрено учебно-методической комиссией
механико-технологического факультета.

Рецензенты: к.т.н. Андрианов В.Н., д.т.н. Юсупов Р.Х.

© Издательство ЮУрГУ, 2003.

1 Микропроцессорная система с тремя шинами

1.1 Общая структура микропроцессорной системы с тремя шинами

Курс «Схемотехника микропроцессорных систем управления» является логическим продолжением курсов электроники и автоматизации. Цель данного курса — дать слушателям представление о современном состоянии систем управления, научить создавать электронные системы, способные управлять теми или иными объектами (процессами).

В последнее время отчетливо проявляется тенденция построения систем управления с использованием элементов цифровой и микропроцессорной техники, которые позволяют сделать системы более гибкими, реализовать алгоритмы управления, недоступные аналоговым системам. Рассмотрим идеализированную схему управления некоторым объектом, которая включает в себя как аналоговые устройства, так и цифровые, рис. 1.

Основой данной системы управления является контроллер. Контроллером называют устройство, выполняющее функцию логического управления. Контроллер может быть создан на базе элементов с жесткой логикой, однако в настоящее время контроллеры создают с использованием микропроцессоров. Микропроцессор (МП) — это функционально законченное устройство, изготовленное в виде большой интегральной схемы (БИС) и выполняющее операции с данными в соответствии с управляющей программой, поступающей на его вход. Управляющая программа может содержать как логические, так и арифметические команды. Использование МП при создании контроллеров позволяет сделать эти устройства более гибкими с точки зрения их работы — контроллеры, управляющие различными устройствами могут быть абсолютно одинаковы по конструкции, differing только управляющими программами. В дальнейшем под контроллером будем понимать микропроцессорную систему, предназначенную для управления некоторым объектом.

Примерами МП, серийно выпускаемых отечественной промышленностью, могут служить МП КР580ВМ80А и КР1810ВМ86.

На рис. 2 показан вариант построения контроллера системы управления с выделением характерных модулей. Помимо МП в состав контроллера входят различные системные устройства, реализованные на интегральных схемах (ИС), в том числе на БИС. Системные устройства могут выполнять различные функции: ввода-вывода данных, измерения временных интервалов, хранения информации и

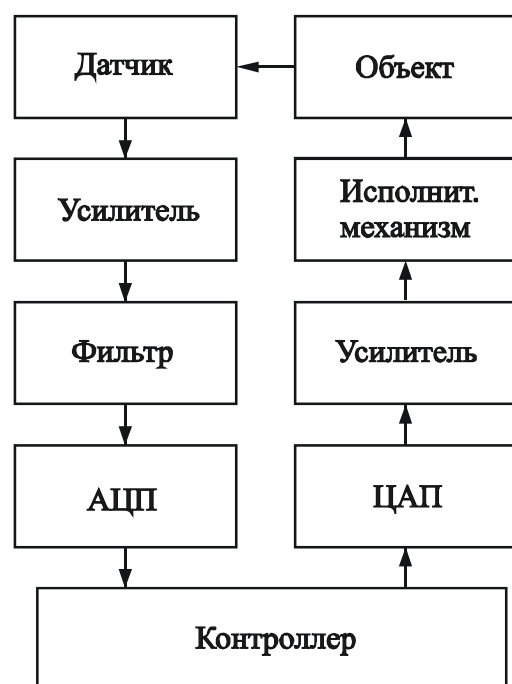


Рис. 1

т. п. Устройства хранения информации принципиально необходимы для функционирования МП, поэтому их выделяют в отдельный класс системных устройств, называемый обычно памятью микропроцессорной системы. Все остальные системные устройства называют периферийными. МП часто называют центральным микропроцессором (ЦМП), подчеркивая его главенствующую роль.

ИС, предназначенные для реализации некоторых периферийных устройств, совместно с микросхемой МП могут образовывать микропроцессорные наборы (комплекты). Микропроцессорный набор — это совокупность совместимых по сигналам и формату данных ИС, разработанных для расширения функциональных возможностей ЦМП. Отмеченные выше МП входят в микропроцессорные наборы КР580 и КР1810 соответственно. Помимо МП в эти наборы соответственно входят следующие микросхемы: КР580ГФ24 — системный генератор, КР580ИР82/83 — буферные регистры, КР580ВА86/87 — 8-ми разрядные шинные формователи, КР580ВК28/38 — системные контроллеры, КР580ВГ18 — контроллер шины, КР580ВГ59 — программируемый контроллер прерываний, КР580ВИ53 — программируемый таймер, КР580ВВ55А — программируемый периферийный адаптер, КР580ВВ51А — программируемый адаптер последовательного интерфейса, КР580ВВ79 — драйвер индикации и клавиатуры, КР1810ВМ87 — арифметический сопроцессор, К1810ГФ84 — системный генератор, К1810ВМ89 — процессор ввода/вывода и другие. Как правило, построение системы с использованием интегральных схем из одного набора требует минимального использования других микросхем, однако микросхемы из набора КР580 широко используются при построении систем на МП КР1810ВМ86. ИС памяти в микропроцессорные наборы не входят; в дальнейшем будем разделять системные устройства на периферийные и на устройства хранения информации, т. е. память.

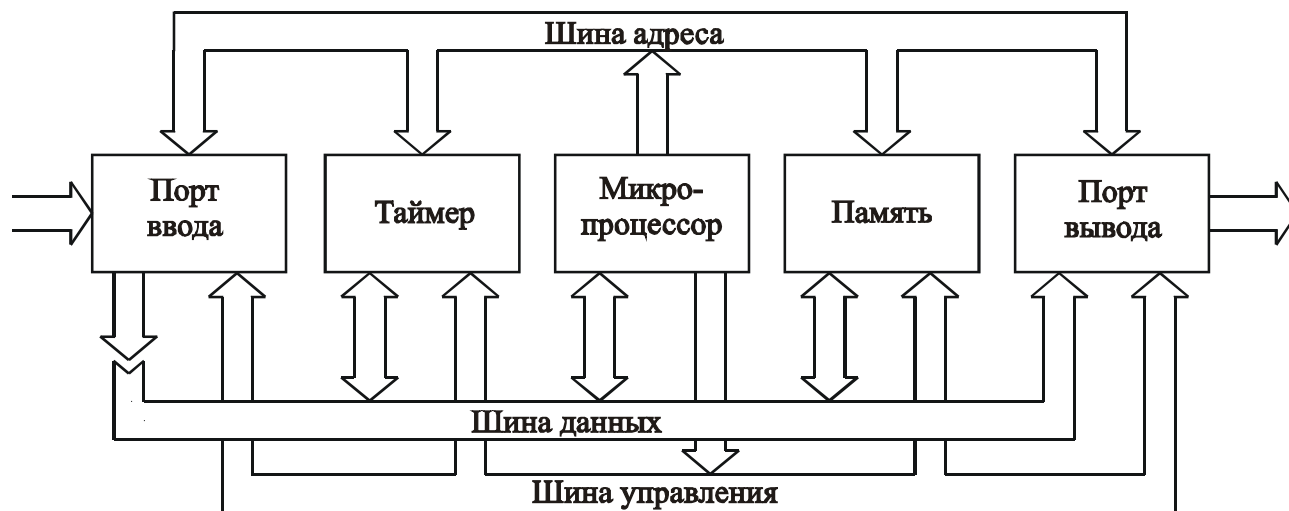


Рис. 2

Взаимодействие между модулями микропроцессорной системы происходит посредством сигналов, распространяющихся по шинам (магистральям). Шиной называют совокупность проводников, обеспечивающих параллельное соединение модулей в единую систему. В микропроцессорной системе выделяют три шины:

данных, адреса и управления, различающиеся по своим функциям и по количеству проводников.

Шина данных (ШД) представляет собой набор двунаправленных сигнальных линий, по которым в параллельном формате (каждый бит по отдельной линии) передается информация, называемая данными. Разрядность ШД может быть различна: относительно простые МП, например, КР580ВМ80А, имеют 8-разрядную ШД, более сложные 16- и 32-разрядную. Разрядность ШД определяет максимальную величину числа, с которым может оперировать МП. Очевидно, что 8-разрядные МП могут оперировать с числами в диапазоне от 0 до 255, а 16-разрядные — от 0 до 65535. Увеличение разрядности положительно сказывается на быстродействии МП. Разрядность ШД является основным критерием классификации МП — МП КР580ВМ80А относится к 8-разрядным микропроцессорам, КР1810ВМ86 — к 16-разрядным.

Шина адреса (ША) — это группа сигнальных линий, посредством которых ЦМП указывает устройство, в которое или из которого должна быть передана информация. Каждое двоичное число, поданное на ША, обозначает адрес устройства, которое, будучи выбрано, оказывается соединенным с ЦМП через ШД и может обмениваться с ним информацией. Под выбором устройства подразумевается его активизация при помощи сигналов на ША.

В качестве устройства, к которому может адресоваться МП может выступать как ячейка памяти, так и регистр, входящий, например, в устройство ввода или вывода. Разрядность ША определяет размер адресного пространства МП, т. е. количество устройств, с которыми МП может взаимодействовать. МП КР580ВМ80А имеет 16-разрядную ША, что позволяет ему выставлять адреса 65536 устройств. На самом деле данный процессор имеет возможность адресоваться к 65536 ячейкам памяти и 256 периферийным устройствам. Это объясняется тем, что для обращения к памяти и периферийным устройствам используются различные команды, формирующие различные сигналы на шине управления. В результате у данного МП выделяет 16-разрядную ША для работы с памятью и 8-разрядную ША для работы с периферийными устройствами.

Характерной особенностью современных МП является использование ША для передачи не только адресной, но и других видов информации за счет временного мультиплексирования шины. В частности, в МП КР1810ВМ86 реализована 20-разрядная ША, 16 младших разрядов которой используются как ШД. Это позволяет более экономно использовать внешние выводы БИС. Принципы временного мультиплексирования шин будут рассмотрены в дальнейшем.

Шина управления (ШУ) — это набор линий для передачи различных сигналов, вырабатываемых ЦМП и другими устройствами, входящими в микропроцессорную систему. Состав этих сигналов определяется типом используемого МП, сигналы могут выполнять разнообразные функции. Выходными сигналами ШУ называют сигналы, формируемые ЦМП для управления пересылкой информации в системе и как ответные на входные сигналы. Входные управляющие сигналы обычно служат для того, чтобы заставить ЦМП выполнить определенное дейст-

вие. Одной из функций шины управления является определение режима работы системы в данное время.

В общем случае микропроцессорная система может выполнять следующие операции:

- запись данных в память из ЦМП;
- считывание ЦМП данных из памяти;
- запись данных из ЦМП в периферийное устройство;
- считывание ЦМП данных из периферийного устройства;
- обработку прерываний;
- прямой доступ к памяти (ПДП) под контролем ЦМП.

Эти операции являются результатом выполнения команд управляющей программы; любую выполняемую микропроцессорной системой операцию можно отнести к одному из этих типов. Однако не обязательно, чтобы система выполняла все эти операции, существуют специализированные системы, выполняющие только некоторые из них.

1.2 Типовые представители системных устройств

1.2.1 Системные устройства хранения информации

Системные устройства хранения информации в микропроцессорных системах подразделяют на постоянные запоминающие устройства (ПЗУ) и оперативные запоминающие устройства (ОЗУ). ПЗУ предназначено для долговременного хранения информации и отличается от энергонезависимостью, т. е. способностью хранить информацию длительное время, от нескольких лет и дольше, после выключения питания устройства. Обычно в ПЗУ хранится управляющая программа, определяющая работу системы, поэтому ПЗУ называют памятью программ. Запись информации в ПЗУ средствами микропроцессорной системы, в которой оно используется, без специальных схемотехнических решений невозможна. ОЗУ не обладает свойством энергонезависимости, но позволяет микропроцессорной системе записывать в себя данные. Поэтому ОЗУ используется для краткосрочного хранения информации в процессе работы системы и называется памятью данных. По способу хранения информации ИС ОЗУ делятся на два типа — статические ОЗУ и динамические ОЗУ. В статических ОЗУ элемент памяти представляет собой набор триггеров (обычно 8 для хранения одного байта). Записанная в триггеры информация будет храниться в них сколь угодно долго пока ее не перепишут или отключат питание. В динамических ОЗУ в качестве базового элемента памяти выступает конденсатор. Т. к. любой конденсатор подвержен саморазряду, то информация в такой ячейке может храниться ограниченное время (обычно несколько секунд), после чего самопроизвольно стирается. Чтобы предотвратить стирание информации к ОЗУ динамического типа применяют операцию регенерации его содержимого, которая заключается в периодической подзарядке конденсаторов элементов памяти. До недавнего времени динамические ОЗУ превосходили статические по объему памяти и были более распространены. В некоторых МП специально закладывались возможности осуществления регенерации динамического

ОЗУ, подключенного к нему. В настоящее время появились относительно дешевые статические ОЗУ большого объема (512 кбайт и больше), поэтому в микропроцессорных системах управления динамическое ОЗУ практически не используется. По этой причине особенности построения систем с динамическим ОЗУ рассматриваться не будут.

Память программ и память данных могут использоваться ЦМП совместно или раздельно, что является его особенностью. Совместное использование означает, что к обеим разновидностям памяти МП обращается с использованием одинаковых команд и по одним шинам. Такое построение МП называется архитектурой фон Неймана. Если же МП построен таким образом, что обращение к памяти программ и данных ведется при помощи различных команд и по различным шинам, то говорят о гарвардской архитектуре и раздельном использовании памяти. Применительно к МП, архитектура фон Неймана является более распространенной.

Т. к. из ИС ПЗУ возможно только чтение данных, в ОЗУ данные могут быть записаны или же извлечены из него. Поэтому эти устройства часто называют ROM (Read Only Memory — память только для чтения) и RAM (Random Access Memory — память произвольного доступа), используя эти аббревиатуры в условно-графических обозначениях (УГО) ИС, рис. 3.

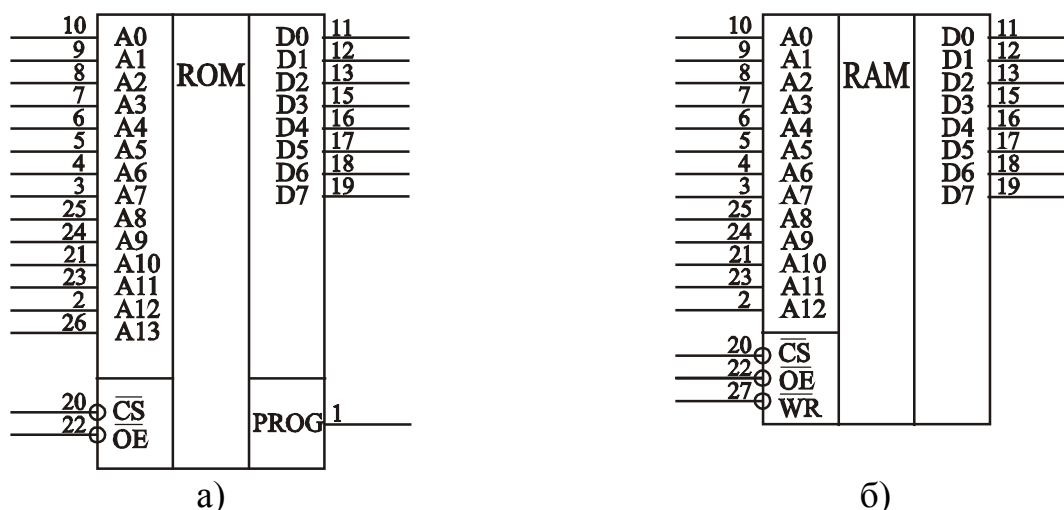


Рис. 3

С точки зрения использования в микропроцессорной системе ПЗУ и ОЗУ отличаются только наборов управляющих сигналов. Типичный набор сигналов управления ПЗУ следующий: сигнал выбора данной ИС \overline{CS} (Chip Select) и сигнал разрешения передачи хранимой информации на входные линии ИС \overline{OE} (Output Enable). В ОЗУ дополнительно необходим сигнал, определяющий направление передачи данных — сигнал \overline{WR} (Write). Линии A_i являются входными адресными линиями, по линиям D_i передаются данные. Количество адресных линий определяет количество ячеек памяти, имеющих в данной ИС, количество линий данных определяет разрядность каждой ячейки. Например, ПЗУ, показанное на рис. 3,а, имеет объем памяти $2^{14} = 16384$ 8-разрядных ячеек.

Взаимодействие ИС памяти с другими устройствами, например, МП, удобно иллюстрировать при помощи диаграмм, показывающих последовательность появления сигналов на линиях ИС. Т. к. предполагается, что сигналы изменяются с течением времени, то их принято называть временными без введения оси времени. Временные диаграммы могут отображать как качественный процесс появления сигналов, так и содержать количественную информацию о длительности сигналов и пауз между ними. На рис. 4,а показана временная диаграмма цикла чтения информации из ИС статического ОЗУ.

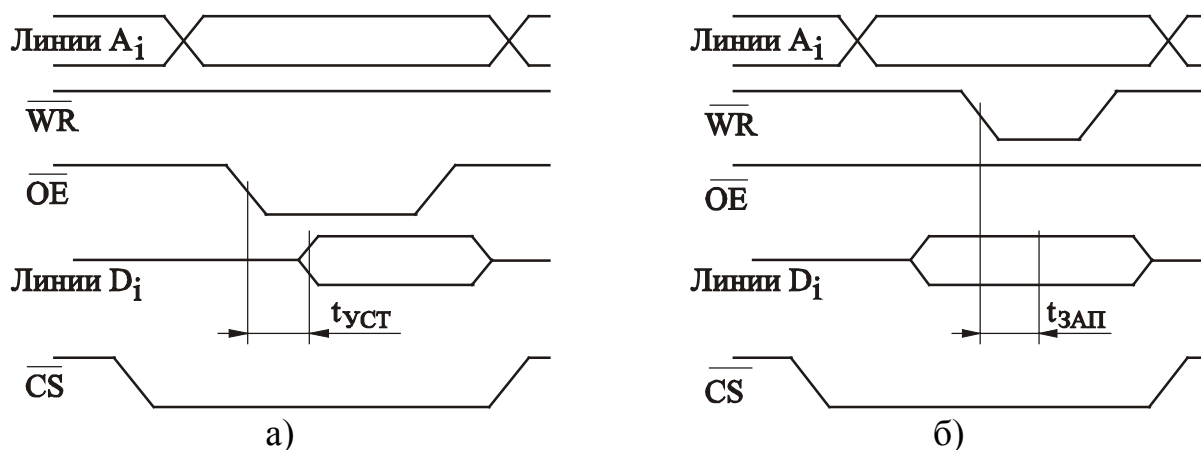


Рис. 4

Из временной диаграммы видно, что в первую очередь внешнее устройство должно установить на адресных линиях ИС памяти адрес той ячейки, информация из которой должна быть считана. Данный адрес будет использован внутренним дешифратором ИС для обращения к нужной ячейке. Далее необходимо сформировать управляющий сигнал \overline{WR} высокого уровня, как показатель того, что будет производиться операция чтения данных, а также управляющий сигнал \overline{OE} , по которому данные из ячейки, адрес которой установлен на входах A_i , будут переписаны во внутренний выходной регистр ИС памяти и появятся на ее выходных линиях D_i .

Основной динамической характеристикой ИС памяти является время установления $t_{уст}$, т. е. время от появления разрешающего сигнала до появления данных на выходных линиях ИС. Очевидно, что в качестве разрешающего сигнала будет выступать сигнал \overline{OE} .

Управляющий сигнал \overline{CS} собственно в операции чтения не участвует. Низкий уровень этого сигнала просто разрешает функционирование ИС памяти, поэтому данный сигнал должен иметь активный уровень при любом обращении к ИС. В противном случае выходные линии D_i будут находиться в высокоимпедансном состоянии.

Операции записи в ОЗУ соответствует временная диаграмма, показанная на рис. 4,б. Собственно запись информации в ОЗУ производится по сигналу \overline{WR} в течении времени $t_{зап}$. Однако, предварительно на адресных линиях должен быть установлен адрес той ячейки памяти, в которую необходимо поместить данные, а

на линиях D_i должны присутствовать данные, которые необходимо записать. Т. к. производится операция записи, сигнал \overline{OE} должен быть неактивен. Управляющий сигнал \overline{CS} должен иметь активный уровень. Время записи $t_{3АП}$ является важной характеристикой ОЗУ.

Операция чтения и записи применительно к ПЗУ будет протекать аналогично.

В качестве типовых представителей ИС памяти можно назвать ИС ПЗУ К573РФ4 объемом 8 кбайт и ИС статического ОЗУ К537РУ10 объемом 2 кбайта.

1.2.2 Программируемый периферийный адаптер КР580ВВ55А

Организация обмена микропроцессорной системы с внешними устройствами обычно строится с использованием периферийных устройств, выполняющих функции ввода или вывода. Эти устройства обычно называют портами ввода или вывода. Если устройство способно выполнять двунаправленный обмен, то его называют портом ввода/вывода. Порты ввода/вывода входят в состав специальных БИС, называемых программируемыми периферийными адаптерами (ППА). К числу ППА следует отнести следующие БИС: 8255, 8255А — ППА фирмы Intel (базовый и улучшенный варианты); МС6820 — ППА фирмы Motorola (базовый и улучшенный варианты); КР580ВВ55, КР580ВВ55А — отечественные аналоги БИС 8255, 8255А.

Программируемый периферийный адаптер КР580ВВ55А — это однокристальное программируемое устройство параллельного ввода/вывода информации произвольного формата. В состав осуществляемых им процедур входит параллельный обмен данными с квити́рованием или без него как в режиме программного управления, так и по прерываниям. При этом обеспечивается организация однонаправленного и двунаправленного ввода/вывода. Определение режимов работы ППА выполняется программными средствами при помощи специального регистра управляющего слова CW . На рис. 5 показано УГО БИС КР580ВВ55А.

В состав ППА входят три двунаправленных 8-разрядных порта, каждый из которых имеет буферный регистр. При построении микропроцессорной системы ППА должен рассматриваться как совокупность 4 регистров (3 буферных регистра портов и регистр CW). Обмен информацией между ЦМП и регистрами ППА осуществляется через линии D_i и управляется сигналами \overline{CS} , A_0 , A_1 , \overline{RD} и \overline{WR} . Адресные сигналы выбирают один из внутренних регистров, а сигналы \overline{RD} и \overline{WR} управляют направлением передачи согласно табл. 1. Сигнал \overline{CS} используется для выбора БИС. Если БИС не выбрана, то линии D_i находятся в высокоимпедансном состоянии.

34	D0	IOP	PA0	4
33	D1		PA1	3
32	D2		PA2	2
31	D3		PA3	1
30	D4		PA4	40
29	D5		PA5	39
28	D6		PA6	38
27	D7		PA7	37
9	A0	КР580ВВ55А	PB0	18
8	A1		PB1	19
	\overline{CS}		PB2	20
6	\overline{RD}		PB3	21
5	\overline{WR}		PB4	22
36			PB5	23
			PB6	24
		PB7	25	
35	RES		PC0	14
			PC1	15
			PC2	16
			PC3	17
			PC4	13
			PC5	12
			PC6	11
			PC7	10

Рис. 5

Таблица 1

A1	A0	\overline{RD}	\overline{WR}	\overline{CS}	Операция
0	0	0	1	0	$D_i \leftarrow$ порт А
0	1	0	1	0	$D_i \leftarrow$ порт В
1	0	0	1	0	$D_i \leftarrow$ порт С
1	1	0	1	0	Недопустимо
0	0	1	0	0	Порт А $\leftarrow D_i$
0	1	1	0	0	Порт В $\leftarrow D_i$
1	0	1	0	0	Порт С $\leftarrow D_i$
1	1	1	0	0	$CW \leftarrow D_i$
x	x	1	1	0	Нет операции
x	x	x	x	1	Нет операции

Вход RES (Reset) служит для аппаратного сброса БИС в начальное состояние. При этом все внутренние регистры ППА, включая регистр управляющего слова CW, устанавливаются в нулевое состояние. Дальнейшая настройка ППА выполняется программно с помощью специального управляющего слова MS (Mode Selection), которое назначает режим работы каждому каналу. Эти режимы могут быть изменены в любое время. Для хранения MS используется регистр CW.

Регистр управляющего слова ППА 7-разрядный. Запись в него осуществляется только при передаче в ППА управляющего слова, старший бит которого равен 1 (признак слова MS). При этом 7 младших битов управляющего слова определяют режим работы каждого порта в соответствии с форматом, приведенным табл. 2.

Таблица 2

Номер бита	Назначение	Режим работы					
		0	Режим 0	0	Режим 1	1	Режим 2
6	Выбор режима работы порта А	0	0	0	0	1	1
5		0	0	1	1	x	2
4	Выбор направления работы линий PA7...PA0 порта А	1	Ввод информации				
		0	Вывод информации				
3	Выбор направления работы линий PC7...PC4 порта С	1	Ввод информации				
		0	Вывод информации				
2	Выбор режима работы порта А	1	Режим 1				
		0	Режим 0				
1	Выбор направления работы линий PB7...PB0 порта В	1	Ввод информации				
		0	Вывод информации				
0	Выбор направления работы линий PC3...PC0 порта С	1	Ввод информации				
		0	Вывод информации				

При D7=0 управляющее слово применяется для независимой установки (сброса) разрядов выходного порта С и носит название BSR (Bit Set/Reset). Такая выбо-

рочная манипуляция содержимым буферного регистра порта С может потребоваться, если к данному порту подключены несколько исполнительных устройств, управление которыми осуществляется независимо. Пример использования BSR будет показан в дальнейшем. Формат BSR достаточно прост: бит D7=0, биты D6, D5, D4 не используются, биты D3, D2, D1 задают адрес разряда порта С, который будет изменен, бит D0 определяет характер изменения содержимого разряда — при D0=0 разряд будет сброшен, при D0=1 разряд будет установлен.

ППА поддерживает три режима работы портов: режим 0 — однонаправленный ввод/вывод без квитирования (применим к любому из трех портов); режим 1 — однонаправленный ввод/вывод с квитированием (применим к портам А и В); режим 2 — двунаправленный ввод/вывод (допускается только для порта А). При работе портов А и В в режимах 1 и 2 часть линий порта С используется для управления обменом с внешними микропроцессорными устройствами. Ограничимся рассмотрением только нулевого режима работы ППА, как наиболее простого и легко реализуемого на практике. Подробное описание остальных режимов можно найти в [10]

В режиме 0 осуществляется однонаправленный ввод/вывод через любой из 3 портов. Рассмотрим временную диаграмму операции чтения из порта А, рис. 6,а.

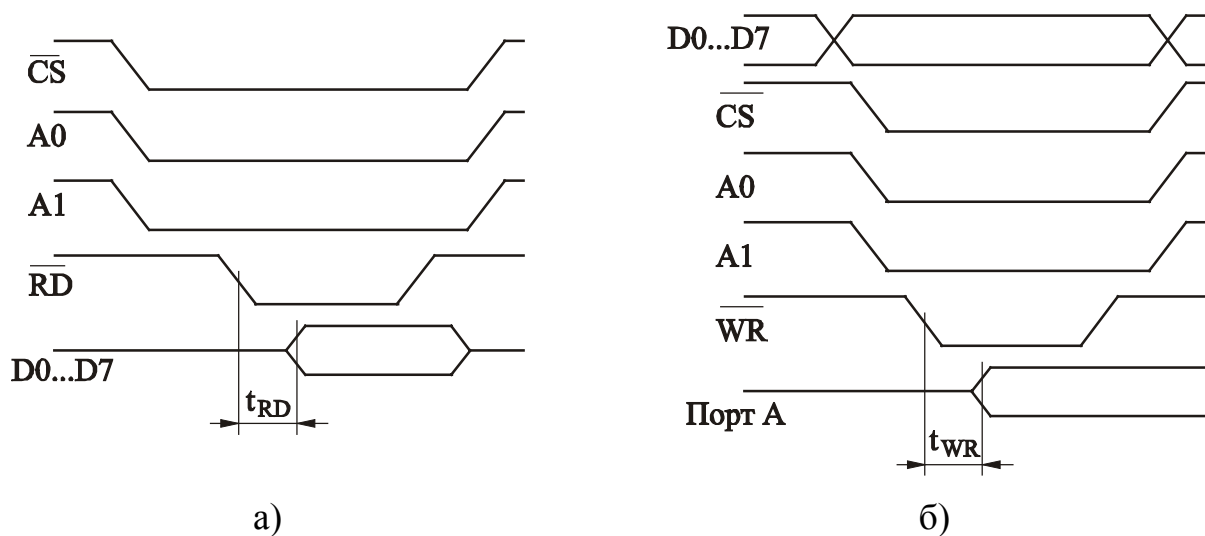


Рис. 6

Изначально порт, из которого предполагается считывание информации, должен быть выбран сигналами \overline{CS} , A0, A1. Считываемые данные появятся на линиях D_i через время t_{RD} после появления сигнала \overline{RD} активного уровня.

Операция вывода информации через порт протекает следующим образом, рис. 6,б. Изначально порт, из которого предполагается считывание информации, выбирается сигналами \overline{CS} , A0, A1. На момент прихода разрешающего вывод сигнала \overline{WR} на линиях D_i должны присутствовать данные, которые требуется вывести. Эти данные появятся на линиях выбранного порта через время t_{WR} после появления сигнала \overline{WR} и останутся там до нового цикла вывода.

1.2.3 Программируемый интервальный таймер КР580ВИ53

БИС КР580ВИ53 — функционально законченное однокристальное программируемое устройство, предназначенное для генерации точных временных интервалов под программным контролем. УГО таймера показано на рис. 7.

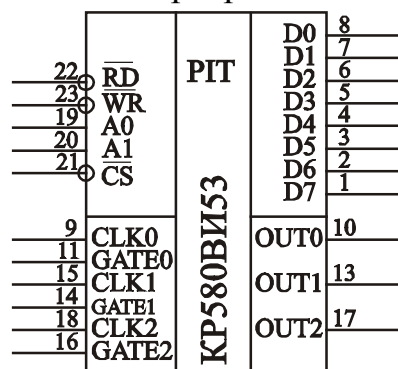


Рис. 7

В состав таймера входят три 16-разрядных вычитающих счетчика с частотой счета по входам CLK до 2 МГц. Каждый счетчик может работать в одном из шести задаваемых программно режимов независимо от других. Все счетчики программно доступны для записи и чтения с помощью слов данных DW и могут работать в двоичном, или в двоично-десятичном коде.

Управление режимами выполняется с помощью управляющего слова MS, которое хранится в регистре SW. Управляющее слово определяет код счета — двоичный или двоично-десятичный, номер режима, формат обмена данными с МП, номер счетчика и к которому относится очередное слово. Формат управляющего слова представлен в табл. 3.

Таблица 3

Номер бита	Режим работы												
	7	0	Выбор счетчика 0		0	Выбор счетчика 1		1	Выбор счетчика 2		1	Недопустимо	
6	0	1		1	0		0	1		1			
5	0	Защелкивание		0	Чт./зап. мл. байта		1	Чт./зап. ст. байта		1	Чт./зап. слова		
4	0	1		1	0		0	1		1			
3	0	Режим 0	0	Режим 1	0	Режим 2	0	Режим 3	1	Режим 4	1	Режим 5	
2	0		0		1		1		0		0		0
1	0		1		0		1		1		0		1
0	1	Двоично-десятичный режим работы											
	0	Двоичный режим работы											

Связь интервального таймера с МП осуществляется через линии данных D_i под управлением сигналов, поступающих на входы A0, A1, \overline{CS} , \overline{WR} , \overline{RD} . Назначение пяти управляющих входных линий таймера аналогично назначению соответствующих линий рассмотренного выше ППА КР580ВВ55А, с той разницей, что вместо регистров портов обращения идут к регистрам счетчиков, табл. 4. Временные диаграммы обмена этих устройств с МП одинаковы с точки зрения последовательности появления сигналов.

Следует отметить, что при двухбайтовом формате обмена данными операции со счетчиками выполняются в два приема: сначала записывается или считывается младший байт, затем — старший. Обслуживание счетчиков выполняется параллельно и независимо друг от друга.

Таблица 4

A0	A1	\overline{RD}	\overline{WR}	\overline{CS}	
0	0	0	1	0	$D_i \leftarrow \text{Счетчик } 0$
0	1	0	1	0	$D_i \leftarrow \text{Счетчик } 1$
1	0	0	1	0	$D_i \leftarrow \text{Счетчик } 2$
1	1	0	1	0	Нет операции
0	0	1	0	0	Счетчик 0 $\leftarrow D_i$
0	1	1	0	0	Счетчик 1 $\leftarrow D_i$
1	0	1	0	0	Счетчик 2 $\leftarrow D_i$
1	1	1	0	0	$CW \leftarrow D_i$
x	x	1	1	0	Нет операции
x	x	x	x	1	Нет операции

При подаче питания состояния и режим работы счетчиков оказываются неопределенными, поэтому перед началом работы каждый счетчик должен быть инициализирован индивидуально посылкой соответствующего управляющего слова MS. Каждое слово, за исключением операции защелкивания ($D5=D4=0$), сопровождается одним или двумя байтами слова данных DW начального состояния выбранного счетчика. Выполнение операции начинается только после загрузки последнего байта данных. Последовательность инициализации и переинициализации счетчиков произвольна. Допускается приостановка последовательности инициализации любого счетчика на неопределенное время с последующим ее продолжением или началом новой.

Рассмотрим режимы, в которых могут работать счетчики интервального таймера. Будем считать, что на вход CLK счетчика поступают прямоугольные импульсы с периодом следования $T_{CLK} = 1 \text{ мкс}$.

Режим 0 — прерывание по окончанию счета. По окончании записи в регистр CW управляющего слова MS на выходе OUT устанавливается сигнал низкого логического уровня. После загрузки слова данных DW счетчик производит декрементацию по каждому срезу сигнала CLK. При переходе счетчика в ноль на выходе OUT устанавливается высокий логический уровень, но работа счетчика при этом не останавливается. Перезапуск счетчика производится при загрузке в него новых данных: запись первого байта слова данных останавливает счет, запись второго запускает новый счет.

Вход GATE разрешает счет при высоком и запрещает при низком уровне напряжения.

Пусть в счетчик, работающий в нулевом режиме, загружено число $C350_{16}=50000$; сигнал $GATE=1$. Тогда через интервал времени $\Delta t = 50000 \cdot T_{CLK} = 0,5 \text{ с}$ после загрузки старшего байта низкий логический уровень на выходе OUT счетчика сменится высоким.

Режим 1 — программируемый одновибратор. По первому срезу сигнала CLK после фронта разрешающего сигнала GATE на выходе OUT появится сигнал низкого логического уровня и счетчик начнет считать. При переходе счетчика в ноль на выходе OUT устанавливается высокий логический уровень. Перезагрузка счетчика во время счета не изменяет длительности текущего импульса. Однако появление нового фронта сигнала GATE перезапускает счетчик с новым или старым значением. Чтение счетчика возможно в любое время.

Пусть в счетчик, работающий в первом режиме, загружено число $C350_{16}=50000$, и разрешающий сигнал GATE изменил свое состояние с низкого уровня на высокий. Тогда через интервал времени $\Delta t = 50000 \cdot T_{CLK} = 0,5$ с после прихода первого среза сигнала CLK низкий логический уровень на выходе OUT счетчика сменится высоким.

Режим 2 — генератор частоты. В данном режиме на выходе OUT присутствуют импульсы низкого уровня с периодом повторения, равным произведению загруженного в счетчик слова данных на период T_{CLK} сигнала CLK. Длительность выходного импульса равна одному периоду сигнала CLK. Подача на управляющий вход GATE сигнала низкого логического уровня сформирует выходной сигнал $OUT = 1$ — генерация импульсов прекратится. Фронт сигнала GATE запустит счетчик из начального состояния, что может быть использовано для аппаратной синхронизации генерируемых импульсов.

Пусть в счетчик, работающий во втором режиме, загружено число $C350_{16}=50000$; сигнал $GATE = 1$. Тогда на выходе счетчика появятся импульсы длительностью $t_{имп} = 1$ мкс и периодом $T_{имп} = 0,5$ с.

Режим 3 — генератор прямоугольных импульсов. На выходе OUT в данном режиме будут присутствовать импульсы прямоугольной формы с периодом повторения, равным произведению загруженного в счетчик слова данных на период сигнала CLK. При четном слове данных, загруженном в счетчик, скважность выходных импульсов будет равна 2. При нечетном DW, равном N, на протяжении $(N+1)/2$ тактов на выходе присутствует уровень лог. 1, а на протяжении $(N-1)/2$ тактов — уровень лог. 0.

Пусть в счетчик, работающий в третьем режиме, загружено число $C350_{16}=50000$; сигнал $GATE = 1$. Тогда на выходе счетчика появятся импульсы длительностью $t_{имп} = 0,25$ с и периодом $T_{имп} = 0,5$ с.

Режим 4 — программная задержка строба. После записи управляющего слова MS на выходе OUT счетчика, работающего в четвертом режиме, устанавливается сигнал высокого логического уровня. Запуск счета осуществляется загрузкой слова данных. При переходе счетчика в ноль на выходе OUT генерируется импульс низкого уровня длительностью в один период сигнала CLK, а счетчик продолжает работать. Сигнал $GATE = 0$ приостанавливает счет.

Пусть в счетчик, работающий в четвертом режиме, загружено число $C350_{16}=50000$; сигнал $GATE = 1$. Тогда через интервал времени $\Delta t = 50000 \cdot T_{CLK} = 0,5$ с после загрузки слова данных на выходе счетчика появится импульс низкого уровня длительностью $t_{имп} = 1$ мкс.

Режим 5 — аппаратная задержка строба. Режим подобен предыдущему, но запуск счетчика осуществляется по фронту сигнала GATE.

В ряде случаев необходимо контролировать текущее состояние счетчиков, например, когда какой-либо из них используется в качестве счетчика событий или реальных часов. Существуют два метода чтения содержимого счетчиков. Первый метод состоит в обычном чтении одного или двух байтов выбранного счетчика. Сначала читается младший, затем старший байт. Для устойчивого чтения текущего состояния счетчика функцию счета желательно заблокировать с помощью входа GATE или внешним запретом импульсов CLK.

Второй метод заключается в чтении содержимого счетчика «на ходу» без запрета его работы. Для этого в интервальный таймер посылается специальная команда защелкивания, см. табл. 4. При этом биты D7 и D6 управляющего слова задают счетчик, состояние которого считывается, а остальные разряды слова могут быть произвольными. По команде защелкивания текущее состояние выбранного счетчика записывается в специальный регистр, что не мешает работе счетчика. Следующая за командой защелкивания операция чтения извлечет информация из этого регистра.

1.3 Временные диаграммы операций обмена в системе с тремя шинами

Выше были перечислены операции, которые может выполнять микропроцессорная система, а также приведены типовые временные диаграммы обмена системных устройств с МП. Рассмотрим последовательность сигналов, которые должны появляться на шинах микропроцессорной системы при выполнении операций обмена.

Анализ временных диаграмм, приведенных на рис. 4 и рис. 6 показывает, что обмен МП с ИС памяти и ИС периферийных устройств строится по одинаковым принципам. При выполнении операций должны строго выдерживаться временные последовательности подаваемых сигналов, определяемые типом используемого системного устройства. Каждая шины микропроцессорной системы выполняет свою собственную функцию и может рассматриваться независимо от остальных. По шине адреса передается адрес системного устройства, с которым взаимодействует ЦМП. Шина данных служит физическим средством для передачи данных между выбранным устройством и ЦМП. В зависимости от выполняемой операции информация по шине данных может передаваться в двух направлениях: от ЦМП к другим устройствам системы (при записи) и от других устройств к ЦМП (при чтении). Для правильной работы микропроцессорной системы необходимо, чтобы в каждый момент времени сигналы на шину данных выставлялись только одним устройством — либо ЦМП, либо одним из системных устройств.

Важным этапом любой операции обмена является установка управляющего сигнала, разрешающего выполнение этой операции. Очевидно, что эти сигналы должны передаваться по шине управления. В микропроцессорной системе применительно к шине управления выделяют 4 управляющих сигнала, каждый из кото-

рых формируется при выполнении одной из операций обмена ЦМП с системными устройствами:

- $\overline{\text{MEMR}}$ — формируется при выполнении операции чтения данных из памяти;
- $\overline{\text{MEMW}}$ — формируется при выполнении операции записи данных в память;
- $\overline{\text{IOR}}$ — формируется при выполнении операции чтения данных из периферийного устройства;
- $\overline{\text{IOW}}$ — формируется при выполнении операции записи данных в периферийное устройство.

Следовательно, шина управления должна состоять как минимум из четырех линий, каждая из которых используется для управления при выполнении отдельной операции.

Во время выполнения операции чтения данных из памяти сигнал управления чтением $\overline{\text{MEMR}}$ не подается на шину управления до тех пор, пока микропроцессор не подаст стабильные сигналы на шину адреса, рис. 8,а.

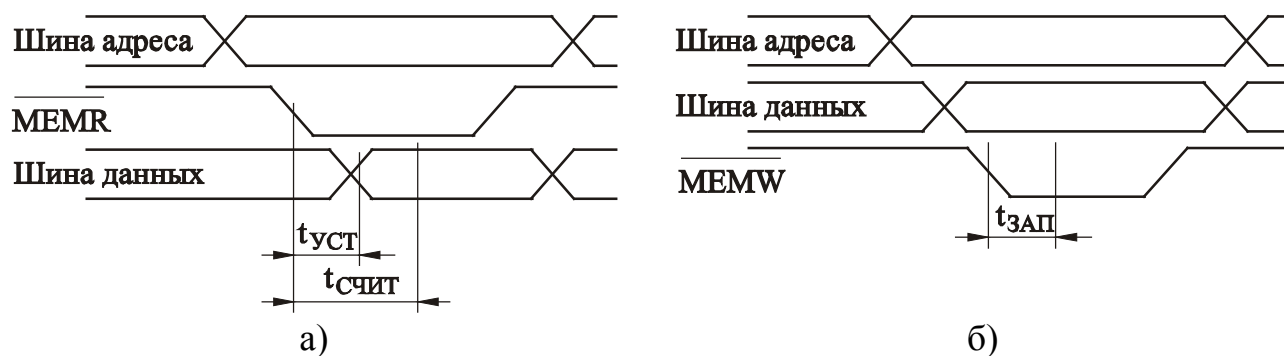


Рис. 8

Появление сигнала $\overline{\text{MEMR}}$ служит разрешением чтения собственно для ИС памяти. Через промежуток времени $t_{уст}$, определяемый характеристиками ИС памяти, на шине данных появляется информация, которую должен считать ЦМП. Прием информации ЦМП осуществляет в некоторый момент времени $t_{счит}$, причем необходимо обеспечить, чтобы $t_{счит} > t_{уст}$. Т. к. быстродействие современных ИС памяти достаточно велико ($t_{уст} \ll 1\text{мкс}$), то отмеченное выше неравенство в большинстве случаев выполняется автоматически. Через фиксированный промежуток времени, больший чем $t_{счит}$, сигнал $\overline{\text{MEMR}}$ сбрасывается, что служит признаком окончания операция чтения данных из памяти. Состояние шины адреса на интервале активности сигнала $\overline{\text{MEMR}}$ должно быть стабильным.

Во время выполнения операции записи данных в память сигнал управления записью $\overline{\text{MEMW}}$ не подается на шину управления до тех пор, пока микропроцессор не подаст стабильные сигналы на шины адреса и данных, рис. 8,б. Появление сигнала $\overline{\text{MEMW}}$ разрешает запись в ИС памяти, которая осуществляется в течение времени $t_{зап}$. Операция записи данных в память из ЦМП завершается сбросом сигнала $\overline{\text{MEMW}}$. Этот сигнал сбрасывается через фиксированный промежуток времени, больший $t_{зап}$. Величина промежутка времени, в течение которого

сигнал $\overline{\text{MEMW}}$ должен быть активен, задается типом ИС памяти, но определяется микропроцессором. Состояние шины адреса и данных в этот промежуток времени должно быть стабильным.

Временные диаграммы при записи данных в периферийное устройство и чтения данных из него фактически не отличаются от рассмотренных временных диаграмм операций обмена с ИС памяти. Очевидно, что отличия будут только в управляющих сигналах — при операции чтения из периферийного устройства используется сигнал $\overline{\text{IOR}}$, при операции записи в периферийное устройство используется сигнал $\overline{\text{IOW}}$, рис. 9,а и рис. 9,б соответственно.

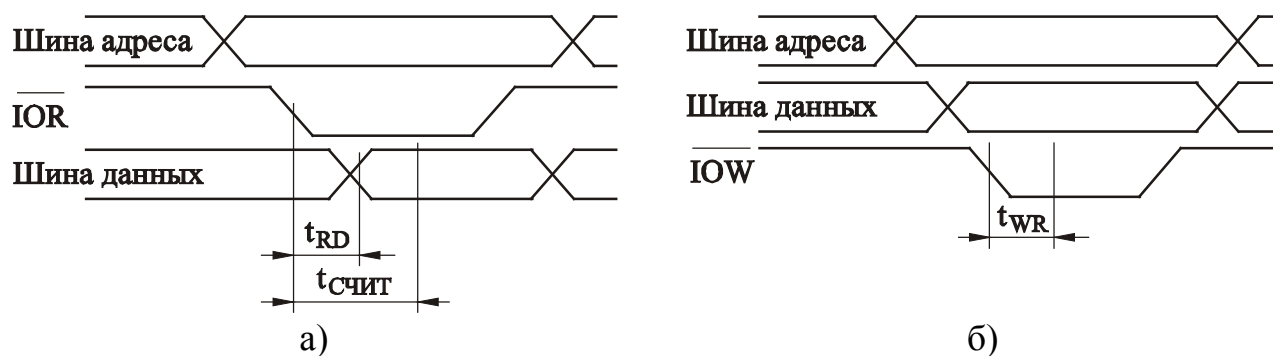


Рис. 9

Выполнением всех временных соотношений в микропроцессорной системе управляет ЦМП, т. е. при выполнении каждой операции микропроцессор будет вырабатывать в соответствующие моменты времени сигналы, подаваемые на каждую из трех шин.

2 Микропроцессор Z80

2.1 Общая характеристика

Как отмечалось выше, МП является основой микропроцессорной системы. Параметры используемого процессора определяют такие важные характеристики микропроцессорной системы, как ее быстродействие, разрядность, объем используемой памяти и количество периферийных устройств. Критерии выбора микропроцессора для построения системы управления с заданными параметрами достаточно неопределенны, и часто сводятся к субъективным оценкам конкретного разработчика. Кроме того, в настоящее время более целесообразно строить системы управления с использованием однокристальных микроЭВМ. Однокристальные микроЭВМ — это устройства, выполненные в виде БИС и содержащие в себе все основные элементы микропроцессорной системы: МП, ПЗУ, ОЗУ, порты ввода/вывода, таймеры и т. п. Однокристальные микроЭВМ принято называть микроконтроллерами, подчеркивая их функциональную завершенность и основное назначение — построение систем управления.

Исходя из современных тенденций, не будем уделять внимания вопросам выбора МП, а рассмотрим основные принципы и схемотехнические решения по-

строения микропроцессорных систем на примере МП Z80. Данный процессор был разработан в конце 70-х годов компанией «Zilog Corporation», применялся в контроллерах многих бытовых и офисных устройств, например, принтеров, а также в бытовых и персональных ЭВМ.

Укрупненно внутренняя структура МП Z80 показана на рис. 10.

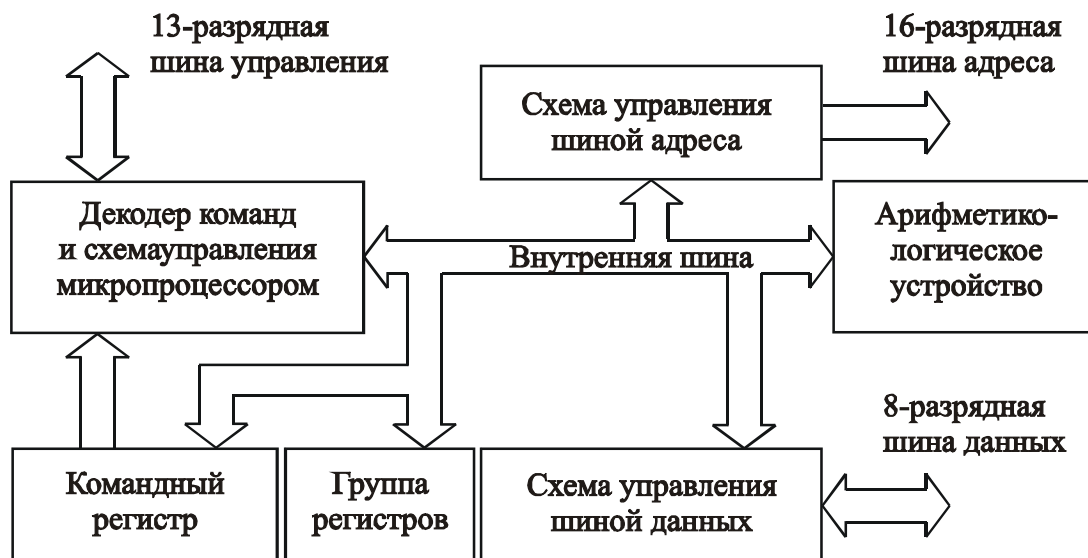


Рис. 10

Команда (инструкция), которую должен выполнить МП, считывается им по 8-разрядной шине данных из устройства, адрес которого предварительно установлен МП на своей шине адреса. Команда считывается как 8-разрядное двоичное число, которое называют кодом данной команды. Код команды по внутренней шине МП поступает в командный регистр, где и хранится до окончания выполнения этой команды.

Внутреннее строение МП принято называть его архитектурой. Одной из особенностей той или иной архитектуры является способ выполнения команд. Работа МП Z80 по выполнению каждой команды основана на принципе микропрограммного управления. Это означает, что каждая команда реализуется как последовательность микроопераций, приводящая к требуемому результату. Преобразование кода команды в последовательность микроопераций осуществляет декодер (дешифратор) команд. Кроме того, совместно со схемой управления декодер команд формирует и обрабатывает сигналы на линиях шины управления МП.

Помимо считывания команд, по шине данных МП способен считывать данные. Обработка данных производится в арифметико-логическом устройстве (АЛУ). Как следует из названия, АЛУ способно выполнять над данными как арифметические операции, так и логические.

Схемы управления шинами выполняют функции буферизирования и управления состоянием шин. В МП Z80, как и во многих других, шины данных и адреса могут принимать три состояния, шина данных является двунаправленной, а шина адреса — однонаправленной. Временное мультиплексирование шин в данном МП не реализовано. Все сигналы на линиях МП Z80 должны иметь уровень ТТЛ.

2.2 Регистры микропроцессора Z80

С точки зрения создания управляющей программы для микропроцессорной системы, построенной на базе МП Z80, наибольший интерес представляет группа регистров. Регистром называют элемент внутренней памяти МП, предназначенный для хранения данных. Регистры принято подразделять на регистры общего назначения (РОН) и регистры специального назначения (специальных функций).

В МП Z80 имеется две группы РОН — основные и вспомогательные. В группу основных РОН входят аккумулятор А, регистры В, С, D, E, H, L, флаговый регистр F; в группу вспомогательных РОН, соответственно, аккумулятор А', регистры В', С', D', E', H', L', флаговый регистр F'. Флаговые регистры 6-разрядные, остальные РОН — 8-разрядные. При обработке двухбайтных слов РОН объединяют в пары BC, DE, HL.

Аккумуляторы и флаговые регистры отличны от остальных РОН. Аккумулятор — это специальный однобайтовый регистр, отличающийся более широким использованием при выполнении команд. В МП Z80, как и во многих других, существуют операции над данными, рвняющимися в аккумуляторе — над данными, хранящимися в других РОН подобные операции невозможны.

Во флаговом регистре хранятся флаги — специальные признаки некоторых операций. В 6-разрядном регистре F хранятся 6 флагов, из которых 4 проверяемые и 2 непроверяемые. Проверяемые флаги — флаг переноса C (Carry), флаг нуля Z (Zero), флаг отрицательного числа S (Sign), флаг четности P (Parity). Непроверяемые флаги — флаги полупереноса и вычитания. Флаги автоматически устанавливаются и сбрасываются после выполнения команд. Считается, что флаг установлен, если значением соответствующего бита во флаговом регистре равно единице. Результат выполнения одной команды может повлиять на состояния нескольких флагов. Однако, существуют команды, не влияющих на флаги. Состояние проверяемых флагов может быть проверено программно и использовано для организации ветвлений в управляющей программе. Непроверяемые флаги программно не опрашиваются, а используются АЛУ МП при выполнении некоторых операций. Например, флаг вычитания используется для коррекции результатов операций над двоично-десятичными числами.

В каждый момент времени программисту доступны либо основные, либо вспомогательные РОН. Переключение между ними осуществляется программно. РОН сохраняют в себе информацию даже находясь в неактивном состоянии.

Регистры специального назначения, как следует из названия, используются для хранения специальной информации, которая используется при выполнении тех или иных операций. К регистрам специального назначения относятся: два 16-разрядных индексных регистра IX, IY, 16-разрядный счетчик команд PC, 16-разрядный указатель стека SP, 8-разрядный регистр векторов прерываний I и 8-разрядный регистр регенерации динамической памяти R. Рассмотрим только те из регистров специального назначения, которые в том или ином виде присутствуют в любом МП. Назначение регистров, являющихся особенностью МП Z80, рассмотрим в дальнейшем по мере необходимости.

Счетчик команд РС предназначен для хранения 16-разрядного адреса команды, которая должна выполняться за текущей командой. Содержимое счетчика команд автоматически изменяется в зависимости от того, какая команда выполняется. Например, пусть выполняется команда записи в регистр D содержимого аккумулятора, располагавшаяся в памяти по адресу 0005_{16} . Данная команда является однобайтной, т. е. состоит из одного байта — кода данной команды. Поэтому следующая команда управляющей программы будет находиться по адресу 0006_{16} , т. е. будет смещена на один байт; именно это число будет находиться в счетчике команд. Существуют команды длиной более одного байта. Например, при выполнении команды записи в регистр D числа 10, располагающейся в памяти по адресу 0005_{16} , в счетчике команд будет находиться число 0007_{16} . Объясняется это тем, что рассматриваемая команда двухбайтная — кроме кода операции (первый байт) в нее входит один операнд — число 10 (второй байт).

В рассмотренных примерах содержимое счетчика команд определялось прибавлением к адресу выполняемой команды ее длины. Однако, если в программе встречается переход, то содержимое счетчика команд изменяется иначе. Например, если текущей командой, располагающейся в памяти по адресу 0005_{16} , является команда безусловного перехода по адресу 2325_{16} , то содержимое счетчика команд будет 2325_{16} .

Указатель стека SP — это регистр, в котором хранится адрес текущей активной ячейки стека. Стеком называется особым образом организованный участок внешней по отношению к МП памяти, выделяемый программистом для временного хранения информации. В качестве информации, хранимой в стеке, обычно выступает содержимое РОН и счетчика команд.

Информация в стеке хранится по принципу LIFO (Last In First Out) — последним вошел, первым вышел. Информация в стек всегда заносится в стек и извлекается из него через активную ячейку, называемую вершиной стека. Стек заполняется в сторону уменьшения адресов — при занесении в стек однобайтного числа содержимое указателя стека уменьшается на единицу, двухбайтного — на два. Соответственно, при извлечении из стека однобайтного числа содержимое указателя стека увеличивается на единицу, двухбайтного — на два.

Стек является областью памяти с последовательным доступом — извлечь из стека информацию можно только после того, как извлечена вся информация, помещенная в стек после требуемой.

2.3 Назначение выводов

На рис. 11 УГО показано МП Z80. Рассмотрим назначение внешних линий этого МП.

A0...A15 — трехстабильная выходная адресная шина; активный уровень сигналов — высокий.

D0...D7 — трехстабильная двунаправленная шина данных; активный уровень сигналов — высокий.

$\overline{M1}$ (Machine cycle 1 — машинный цикл 1) — выход, активизирующийся при чтении кода операции из памяти; активный уровень сигнала — низкий.

\overline{MREQ} (Memory request — запрос памяти) — трехстабильный выход, активизирующийся при обращении МП к памяти; активный уровень сигнала — низкий.

\overline{IORQ} (Input/Output request — запрос ввода/вывода) — трехстабильный выход, активизирующийся при обращении МП к периферийному устройству; активный уровень сигнала — низкий.

\overline{RD} (Read — чтение) — трехстабильный выход, указывающий в активном состоянии, что МП готов к чтению данных из памяти или из периферийного устройства; активный уровень сигнала — низкий.

\overline{WR} (Write — запись в память) — трехстабильный выход, указывающий в активном состоянии, что на шине данных микропроцессора содержатся данные, предназначенные для записи в память или для вывода на периферийное устройство; активный уровень сигнала — низкий.

\overline{RFSH} (Refresh — регенерация) — выходной сигнал, определяющий начало операции регенерации динамической памяти; активный уровень — низкий.

\overline{HALT} (Halt state — состояние останова) — выходной сигнал с низким активным уровнем; указывает, что МП находится в состоянии останова, т. е. непрерывно выполняет «пустую» команду.

\overline{WAIT} (Wait — ожидание) — входной сигнал с низким активным уровнем; указывает МП, что адресуемое системное устройство не готово для выполнения передачи данных. Микропроцессор будет находиться в состоянии ожидания до тех пор, пока этот сигнал активен.

\overline{INT} (Interrupt request — запрос на прерывание) — входной сигнал с низким активным уровнем.

\overline{NMI} (Non maskable interrupt — немаскируемый запрос на прерывание) — входной сигнал с низким активным уровнем.

\overline{RESET} (Reset — сброс) — входной сигнал с низким активным уровнем, имеет самый высокий приоритет. Длительность сигнала сброса — не менее 3 периодов тактирующего сигнала. Во время действия сигнала сброса адресная шина и шина данных переводятся в состояние высокого импеданса, а для всех управляющих выходных сигналов устанавливается неактивный уровень или высокоимпедансное состояние. После снятия сигнала \overline{RESET} МП начинает выполнение программы, начало которой находится в ячейке памяти с нулевым адресом.

\overline{BUSRQ} (Bus request — запрос доступа к шине) — входной сигнал с низким активным уровнем. При обнаружении этого сигнала МП переводит линии шин

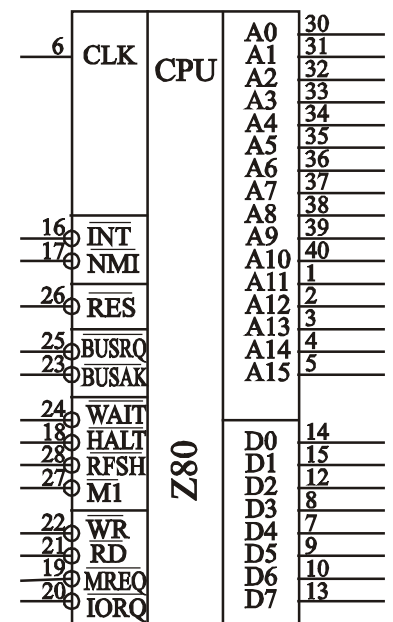


Рис. 11

адреса и данных, а также линий \overline{MREQ} , \overline{IORQ} , \overline{RD} , \overline{WR} в высокоимпедансное состояние, после чего шинами микропроцессорной системы может управлять другое устройство.

BUSAK (Bus acknowledge — предоставление доступа к шине) — выходной сигнал с низким активным уровнем; подается на запрашивающее доступ к шине внешнее устройство для подтверждения того, что линии шин адреса и данных, а также линий \overline{MREQ} , \overline{IORQ} , \overline{RD} , \overline{WR} переведены в высокоимпедансное состояние и внешнее устройство может управлять шинами микропроцессорной системы.

CLK (Clock — такт) — вход, предназначенный для приема последовательно-тактовых импульсов. Внутренняя система синхронизации МП обеспечивает работу всех его составляющих синхронно с перепадами тактовых импульсов. Интервал времени, соответствующий периоду следования этих импульсов называется машинным тактом и является важной характеристикой МП, т. к. длительность любой команды, выполняемой МП, равна строго определенному количеству машинных тактов.

Питание БИС МП Z80 осуществляется напряжением 4,75...5,25 В; минус питания подается на 20-й вывод БИС, плюс — на 40-й.

2.4 Временные диаграммы основных машинных циклов

МП осуществляет свою работу в соответствии с управляющей программой хранящейся в ПЗУ начиная с нулевого адреса. Управляющая программа представляет собой набор команд, каждая из которых заставляет МП выполнять вполне определенное действие.

Важной характеристикой команды является машинный цикл. Машинным циклом называют характерную операцию (микрооперацию), выполняемую микропроцессором. В перечень основных машинных циклов микропроцессора Z80 входят следующие операции: чтение операции — цикл M1; чтение данных из памяти; запись данных в память; чтение данных из периферийного устройства; запись данных в периферийное устройство; подтверждение доступа к шине; подтверждение прерывания; выполнение команды останова.

Каждая команда, выполняемая микропроцессором, состоит из строго определенного количества машинных циклов — от одного до шести. В свою очередь продолжительность каждого машинного цикла равна строго определенному количеству машинных тактов.

На рис. 12 показана временная диаграмма цикла чтения команды. Содержимое счетчика команд РС подается на адресную шину непосредственно в начале машинного цикла. Через половину периода следования тактовых импульсов подается сигнал \overline{MREQ} . Задний фронт этого сигнала может быть использован для отпирания кристалла ИС памяти. Активное состояние сигнала \overline{RD} показывает, что МП готов к приему информации по шине данных. Собственно чтение информации МП выполняет по переднему передний фронту синхронизирующего импульса, соответствующего такту T3. Во время тактов T3 и T4 цикла чтения команды

производится регенерация динамической памяти, о чем свидетельствует активный уровень сигнала \overline{RFSH} , а так же МП декодируют и выполняет считанную команду. Во время, соответствующее приходу заднего фронта импульса T2, МП анализирует состояние на входе \overline{WAIT} . Если в данный момент времени на линии \overline{WAIT} присутствует сигнал активного (низкого) уровня, то процессор не станет выполнять дальнейшие операции цикла выборки команды до тех пор, пока сигнал на линии \overline{WAIT} не станет пассивным.

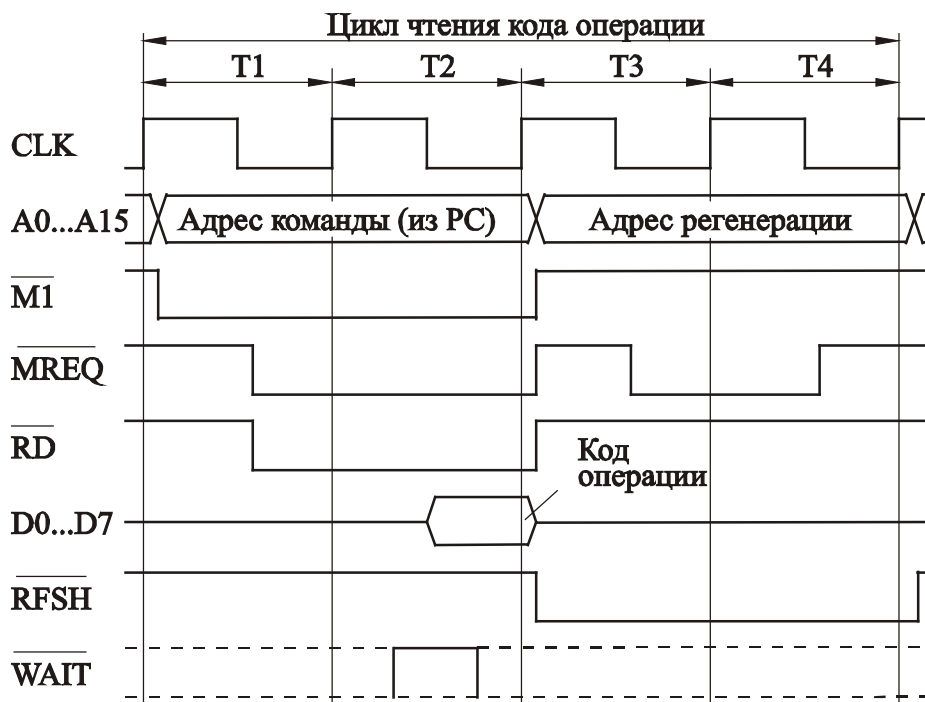


Рис. 12

На рис. 13,а показана временная диаграмма цикла чтения из памяти. Данная временная диаграмма отличается от временной диаграммы цикла M1, рассмотренной выше. Однако здесь, как и в цикле чтения кода операции, используются сигналы \overline{MREQ} и \overline{RD} . Чтение данных микропроцессором происходит во время третьего такта, если во время заднего фронта второго тактирующего импульса сигнал ожидания находится в неактивном состоянии. В противном случае микропроцессор чтения производить не будет, а вновь проверит состояние линии ожидания. Адрес ячейки памяти, из которой производится чтение, выставляется МП на шине адреса в самом начале цикла.

На рис. рис. 13,б показана временная диаграмма записи в память. Адрес ячейки памяти, в которую производится запись, выставляется МП на шине адреса в самом начале цикла. Сигнал \overline{MREQ} активизируется через половину тактового импульса, когда уровни сигналов на адресной шине стабилизировались. Поэтому он может быть непосредственно использован в качестве сигнала отпираания кристалла ИС памяти. Одновременно с активизацией сигнала \overline{MREQ} МП выставляет на шине данных информацию, которая должна быть записана в ячейку памяти.

Линия подачи сигнала \overline{WR} активизируется, когда данные на шине данных уже стабилизировались. Следовательно, этот сигнал можно непосредственно использовать в качестве разрешающего запись сигнала для ИС памяти.

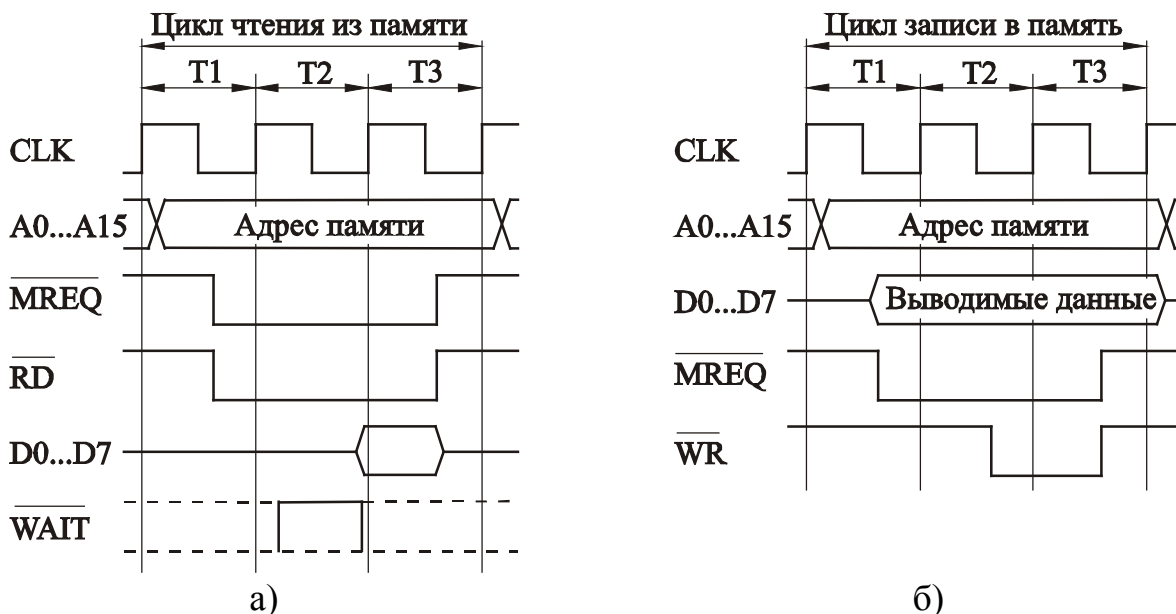


Рис. 13

На рис. 14,а показана временная диаграмма цикла ввода данных из периферийного устройства. Адрес устройства, из которого будет производиться чтение данных, выставляется МП на шине адреса в самом начале данного цикла. В начале такта T2 МП выставляет сигналы \overline{IORQ} и \overline{RD} , подтверждая готовность считывать данные из периферийного устройства. Эти сигналы могут использоваться для разрешения периферийному устройству установить на шине данных передаваемую в МП информацию. Во время выполнения рассматриваемого цикла чтения автоматически генерируется состояние ожидания, продолжающееся в течение времени T_w . Благодаря состоянию ожидания обеспечивается время, достаточное для дешифрирования адреса в периферийном устройстве и, при необходимости, для активизирования линии сигнала \overline{WAIT} .

Данные от внешнего устройства должны поступить на шину данных не позднее начала такта T3, т. к. в середине этого такта процессор считывает информацию с шины данных и переводит линии \overline{IORQ} и \overline{RD} в неактивное состояние.

На рис. 14,б показана временная диаграмма цикла вывода (записи) данных в периферийное устройство. Адрес устройства, в которое будет производиться запись данных, выставляется МП на шине адреса в самом начале данного цикла. Спустя временной интервал, равный половине периода тактовых импульсов, МП выставляет на шину данных информацию, которую требуется передать во внешнее устройство. В начале следующего такта T2 процессор выставляет сигналы \overline{IORQ} и \overline{WR} , которые могут использоваться как сигналы разрешения внешнему устройству считать информацию с шины данных. Во время выполнения рассматриваемого цикла также генерируется состояние ожидания, продолжающееся в те-

чение времени T_w . В середине такта T_3 МП переводит линии \overline{IORQ} и \overline{WR} в неактивное состояние, а в конце этого такта он освобождает шину данных.

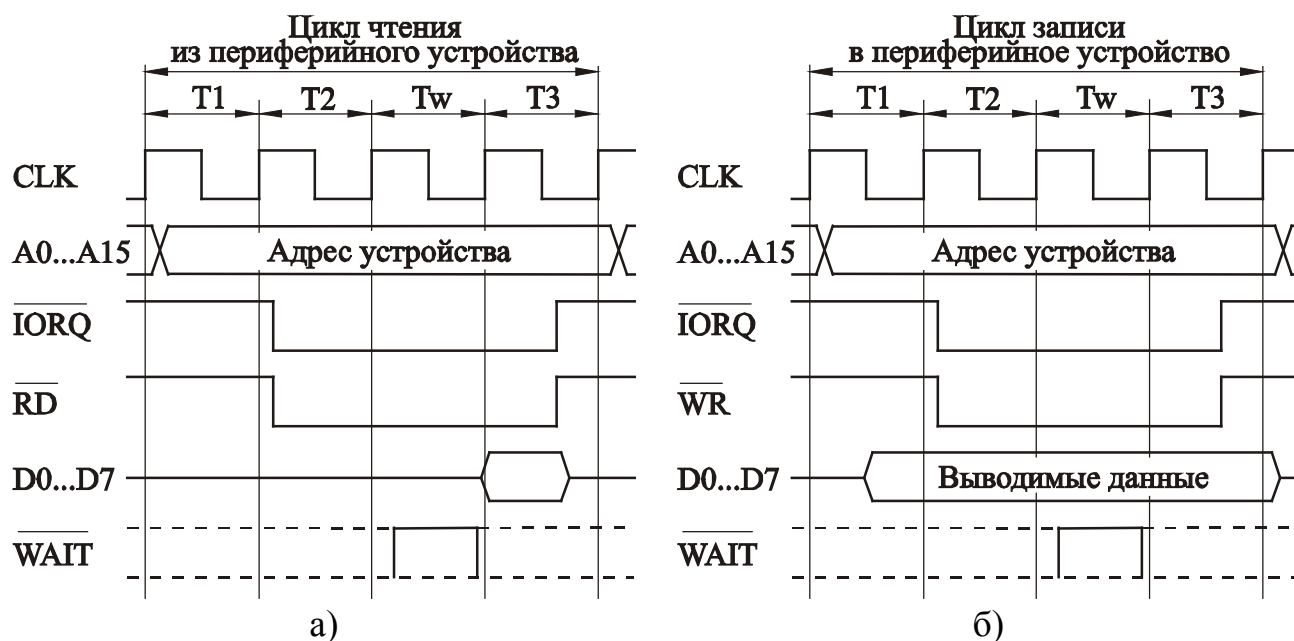


Рис. 14

Следует отметить, что при обращении к памяти МП Z80 использует все 16 адресных линий, а при обращении к периферийным устройствам — только 8 младших. Следовательно, МП Z80 может адресоваться к 64 кбайтам внешней памяти и к 256 периферийным устройствам.

2.5 Способы адресации в микропроцессоре Z80

Под способами адресации понимают различные варианты взаимодействия МП с памятью микропроцессорной системы по линиям шин адреса и данных, определяемые месторасположением данных, над которыми выполняется та или иная операция. В МП Z80 предусмотрены следующие способы адресации.

Выборка команды — осуществляется в цикле M1. Информация, поступающая в ходе этого цикла в МП по линиям данных, трактуется им как код команды, которая начинает выполняться после декодирования. Напомним, что команды МП Z80 бывают одно-, двух- и трехбайтными.

Непосредственная адресация — реализуется для двухбайтных команд. В этом случае ячейка ПЗУ, следующая за ячейкой со считанным кодом операции, содержит 8-разрядный операнд — число. Непосредственная адресация является наиболее экономичным способом хранения и поиска информации, поскольку необходимые данные содержит сама команда. Примером команды, для которой реализуется непосредственная адресация, может служить команда записи в аккумулятор МП числа 25: мнемоническое обозначение этой команды LD A, 5, в ПЗУ эта команда будет храниться в виде чисел $3E_{16}$ и 19_{16} .

Расширенная непосредственная адресация — реализуется для трехбайтных команд. В ячейках ПЗУ, следующих за ячейкой со считанным кодом операции,

при этом способе адресации находится 16-разрядный операнд — число. Примером команды, для которой реализуется непосредственная адресация, может служить команда записи в регистровую пару HL МП числа $289=0121_{16}$: мнемоническое обозначение этой команды LD HL, 5, в ПЗУ эта команда будет храниться в виде чисел 21_{16} , 21_{16} и 01_{16} .

Модифицированная адресация нулевой страницы — реализуется для однобайтной команды рестарта RST N. Выполнение данной команды переводит МП к выполнению подпрограммы, расположенной начиная с конкретной ячейки памяти. Использование команды рестарта будет рассмотрено далее при описании нулевого типа маскируемого прерывания.

Относительная адресация. В этом режиме задается однобайтовое смещение в диапазоне от +127 до -128 относительно текущего адреса плюс 2, что дает возможность обращения к расположенной поблизости области памяти с помощью команды длиной в 2 байта. Например, чтобы осуществить переход на ячейку памяти, смещенную на $20=14_{16}$ байт вперед относительно текущего адреса, то можно воспользоваться командой JR 20, которая в ПЗУ будет храниться в виде чисел 18_{16} , и 12_{16} .

Прямая адресация — реализуется для трехбайтных команд. При этом полный 16-разрядный адрес назначения определяется двухбайтовым операндом, следующим в команде за кодом операции. Этим способом можно обеспечить обращение к подпрограмме или переход по любому адресу памяти. Примером команды с прямой адресацией может служить команда вызова подпрограммы, расположенной по адресу $30657=77C1_{16}$. Мнемоническое обозначение данной команды имеет вид CALL 30657, в ПЗУ команда будет храниться в виде чисел CD_{16} , $C1_{16}$ и 77_{16} .

Индексная адресация — реализуется для команд, использующих индексные регистры IX и IY. Смещение, хранящееся в одной из ячеек ПЗУ, следующих за ячейкой с кодом команды, прибавляется к содержимому индексного регистра, образуя действительный адрес памяти. Пусть, например, в индексном регистре IX хранится число 4356, тогда команда ADD A, (IX+15) прибавит к содержимому аккумулятора число, которое хранится в ячейке памяти с адресом $4356+15=4371$. Рассмотренная команда будет храниться в ПЗУ в виде чисел DD_{16} , 86_{16} и $0F_{16}$.

Регистровая адресация. При регистровой адресации код операции содержит указание на регистр или пару регистров, в которых содержатся данные; при мнемонической записи команды в ней обозначения регистров присутствуют явно. Примером может служить однобайтная команда LD A, B (код 78_{16}), которая переписывает данные из регистра B в аккумулятор.

Неявная адресация, при которой регистр, хранящий информацию, однозначно определяется данной командой. Например, однобайтовая команда CPL (код $2F_{16}$) не имея явного указания на аккумулятор, выполняет операцию инвертирования над его содержимым.

Регистровая косвенная адресация. В этом случае информация, находящаяся в 16-разрядной регистровой паре, например HL, интерпретируется МП не как данные, а как 16-разрядный адрес ячейки памяти, в которой хранятся данные. Примером может служить команда LD A, (HL), которая имеет код $7E_{16}$. Пусть в регистре

Н хранится число 23_{16} , а в регистре L — число 76_{16} . Тогда после выполнения команды LD A, (HL) в аккумулятор будет занесена информация, хранящаяся в ячейке памяти, расположенной по адресу 2376_{16} .

Побитовая адресация — реализуется для операций, позволяющих работать с отдельными битами ячейки памяти. Три разряда кода такой операции определяют конкретный разряд ячейки памяти, над которым производится данная операция. Например, двухбайтная команда SET 1, (HL) установит в единичное состояние первый бит ячейки памяти, полный 16-разрядный адрес которой хранится в регистровой паре HL. Код этой команды двухбайтный: $СВСЕ_{16}$. Разряды D4...D6 второго байта команды $СЕ_{16}=11001110_2$ определяют номер устанавливаемого бита.

2.6 Система команд

Систему команд МП Z80 составляют 158 команд, полная совокупность которых описывается 699 кодами. Т. к. шина данных этого МП 8-разрядная, то он может распознать всего 256 однобайтных кодов. Поэтому некоторые команды МП Z80 имеют двухбайтный код.

С полным перечнем команд МП Z80 можно ознакомиться в [6]. В отличие от некоторых других источников, рекомендуемая книга содержит такую важную для программиста информацию, как длительность выполнения команд в тактах МП.

Приведем только общие сведения о системе команд МП Z80. Мнемонические обозначения, используемые при записи команд данного МП, просты и наглядны. Многие команды характерны для других МП.

МП Z80 имеет команды пересылки данных. Структура этих команд следующая: LD операнд-приемник, операнд-источник — данные пересылаются из операнда-источника в операнд-приемник. Примерами команд пересылки могут служить следующие команды: LD H, 10, LD B, C, LD C, (HL), LD A, (31476), LD D, (IX+15), LD (HL), A. Данные могут перемещаться из регистра в регистр, из памяти в регистр, из регистра в память или из одних ячеек памяти в другие.

Команды переходов предназначены для изменения последовательного выполнения команд управляющей программы. Команды перехода позволяют перейти к выполнению программы с указанного адреса безусловно — команда JP адрес, или по условию — команда JP условие, адрес. В командах условного перехода будет осуществлен переход на указанный адрес, если условие выполнено. В качестве условий выступают состояния проверяемых флагов регистра F МП, поэтому команды условного перехода могут иметь вид JP NZ, 25645 — переход на адрес 25645, если флаг Z сброшен, JP Z, 34890 — переход, если флаг Z установлен, JP P, 64731 — переход, если флаг S установлен, JP M, 47360 — переход, если флаг S сброшен. Флаги устанавливаются или сбрасываются при выполнении некоторых (не любых!) команд МП.

Аналогичную структуру имеют команды относительного перехода: JR 100, JR Z, -20 и т. п.

К командам перехода относятся также команды безусловного (CALL адрес) и условного (CALL условие, адрес) вызова подпрограмм, команды безусловного

(RET) и условного (RET условие) возврата из подпрограмм, а также команды RETN и RETI возврата из подпрограмм обработки прерываний.

МП Z80 способен выполнять над данными арифметические и логические операции. Для этого служат специальные команды, например ADD A, данные — прибавляет к содержимому аккумулятора данные, которыми могут являться число (ADD A, 10), содержимое регистра (ADD A, B), содержимое ячейки памяти (ADD A, (HL)). Предусмотрены команды арифметических операций над 16-разрядными данными, например, ADD HL, DE.

В качестве одного из операндов команд логических операций OR, AND, XOR выступает аккумулятор: OR A, 11, AND A, B, XOR A, (HL).

МП может осуществлять над данными, находящимися в регистрах МП или в памяти системы операции сдвига. Например, команда RLCA осуществляет циклический сдвиг содержимого аккумулятора влево, а команда RRC (HL) циклически сдвигает вправо содержимое ячейки памяти, чей адрес хранится в регистрах HL.

Мнемонические обозначения команд, выполняемых МП, используются при написании управляющих программ на языке ассемблера, что позволяет сделать ее более наглядной. Текст программы на языке ассемблера имеет определенный формат. Каждая команда представляет собой строку, состоящую из четырех полей: **МЕТКА ОПЕРАЦИЯ ОПЕРАНД(Ы) КОММЕНТАРИЙ**

Поля могут отделяться друг от друга произвольным числом пробелов. Все поля являются необязательными.

В поле метки размещается символическое имя ячейки памяти, в которой располагается отмеченная команда или операнд. Метка представляет собой буквенно-цифровую комбинацию, начинающуюся с буквы. Разрешается использовать буквы только латинского алфавита. Длина метки может ограничиваться требованиями конкретной программы-транслятора — программы, переводящий текст, написанный на языке ассемблера, в последовательность чисел, которые и составят собственно управляющую программу для МП. Длина метки в 6 символов допускается большинством трансляторов, поэтому превышать ее не рекомендуется. Метка всегда завершается двоеточием. В качестве символических имен и меток не могут быть использованы мнемонические обозначения команд, псевдокоманд и операторов ассемблера, а также мнемонические обозначения регистров и других внутренних блоков МП.

В поле операции записывается мнемоническое обозначение команды МП или псевдокоманды ассемблера. Для конкретного МП используется строго определенный набор мнемонических кодов. Любой другой набор символов, размещенный в поле операции, воспринимается транслятором как ошибочный.

В поле операндов определяются операнды (или операнд), участвующие в операции. Операнды разделяются запятой.

Операнд может быть задан непосредственно или в виде прямого или косвенного адреса. Непосредственный операнд представляется числом или символическим именем, которому соответствует определенное число. В командах передачи

управления операндом может также являться метка. Используемые в качестве операндов символические имена и метки должны быть определены.

Числа записываются с указанием системы счисления, для чего используется суффикс (буква, стоящая после числа): В — для двоичной, Q — для восьмеричной, D — для десятичной и H — для шестнадцатеричной. Число без суффикса по умолчанию считается десятичным. Если шестнадцатеричное число начинается с символа A, B, C, D, E или F, то перед этим символом располагают незначащий ноль: 0FD34H, 0AH.

Некоторые программы в процессе трансляции текста, написанного на языке ассемблера, способны обрабатывать выражения, записанные в поле операндов. Выражение представляет собой совокупность символических имен и чисел, связанных операторами ассемблера. Операторы ассемблера обеспечивают выполнение арифметических («+» — сложение, «-» — вычитание, «*» — умножение, «/» — целочисленное деление) и логических (OR, AND, XOR, NOT) операций.

Поле комментария может быть использовано программистом для текстового или символьного пояснения логической организации управляющей программы. Поле комментария полностью игнорируется при трансляции, потому в нем допустимо использовать любые символы. По правилам языка ассемблера поле комментария начинается после точки с запятой.

Текст, написанный на языке ассемблера, транслируется в последовательность чисел, которые и составят собственно управляющую программу для МП. Данная последовательность называется объектным кодом. Транслирующая программа берет на себя многие из рутинных задач, таких как присвоение действительных адресов вместо меток, преобразование чисел, присвоение действительных значений символьным переменным и т. п., программист все же должен указать ей некоторые параметры, например, начальный адрес размещения управляющей программы в ПЗУ, момент окончания программы и т. п. Для этой цели служат псевдокоманды ассемблера.

Псевдокоманда ORG задает адрес ячейки памяти, в котором будет располагаться после трансляции следующая за ней команда программы. Например, после трансляции следующего фрагмента

```
ORG 38H
LD A, (HL)
```

в ячейку с адресом 38_{16} будет занесен код операции LD A, (HL) — число $7E_{16}$.

Псевдокомандой EQU можно любому символическому имени поставить в соответствие некоторый операнд, обычно число. Примером может служить следующий фрагмент:

```
UGOL EQU 180
...
LD A, UGOL
```

После трансляции в некоторую ячейку памяти будет помещено число $3E_{16}$ — код операции LD A, число, а в следующую — число $B4_{16}=180$.

Псевдокоманды резервирования памяти и определения данных DB, DW позволяют размещать в памяти данные, представленные в виде байт или слов.

Псевдокомандой END транслятору дается указание завершить трансляцию программы.

2.7 Реализация некоторых операций

Рассмотрим реализацию некоторых операций, часто встречающихся в практике программирования. Т. к. рассматриваемые примеры характерны для программирования на языке ассемблера для любого МП, то в ряде случаев одна задача будет реализовываться несколькими способами, чтобы подчеркнуть особенности системы команд МП Z80.

Занести в ячейку с адресом 097_{16} число 10.

LD A, 10	; занести в аккумулятор число 10
LD HL, 097H	; занести в регистровую пару HL число 97_{16}
LD (HL), A	; занести содержимое аккумулятора в ячейку ; памяти, адрес которой находится в HL

Или

LD A, 10	; занести в аккумулятор число 10
LD (097H), A	; занести содержимое аккумулятора в ячейку ; памяти с адресом 97_{16}

Извлечь данные из ячейки с адресом 065_{16} , инвертировать их и занести в следующую ячейку.

LD A, (065H)	;занести в аккумулятор содержимое ячейки ;памяти с адресом 65_{16}
CPL	;инвертировать содержимое аккумулятора
LD (066H), A	;занести содержимое аккумулятора в ячейку ;памяти с адресом 66_{16}

Или

LD HL, 065H	;занести в регистровую пару HL число 65_{16}
LD A, (HL)	;занести в аккумулятор содержимое ячейки ;памяти, адрес которой находится в HL
CPL	;инвертировать содержимое аккумулятора
INC HL	;инкрементировать (увеличить на единицу) ;содержимое HL
LD (HL), A	;занести содержимое аккумулятора в ячейку ;памяти, адрес которой находится в HL

Увеличить вдвое данные, хранящиеся в ячейке с адресом 065_{16} .

LD A, (065H)	;занести в аккумулятор содержимое ячейки ;памяти с адресом 65_{16}
LD B, A	;сохранить содержимое аккумулятора в регистре B
ADD A, B	;сложить содержимое аккумулятора и регистра B
LD (065H), A	;занести содержимое аккумулятора в ячейку ;памяти с адресом 65_{16}

Или

LD HL, 065H	;занести в регистровую пару HL число 65_{16}
-------------	--

LD A, (HL) ;занести в аккумулятор содержимое ячейки
;памяти, адрес которой находится в HL
SLA A ;сдвинуть влево на разряд содержимое
;аккумулятора, нулевой разряд обнулится
LD (HL), A ;занести содержимое аккумулятора в ячейку
;памяти, адрес которой находится в HL

Или

LD HL, 065H ;занести в регистровую пару HL число 65_{16}
SLA (HL) ;сдвинуть влево на разряд содержимое
;ячейки памяти, адрес которой находится в HL

Увеличить содержимое регистра D в пять раз,

LD A, D ;сохранить содержимое регистре D в аккумуляторе
SLA D ;сдвинуть влево на два разряда содержимое
SLA D ;регистра D
ADD A, D ;сложить содержимое аккумулятора и регистра D
LD D, A ;сохранить содержимое аккумулятора в регистре D

Вычислить сумму чисел, хранящихся в 10 ячейках, начиная с адреса 0111_{16} .
Результат поместить в регистровую пару HL.

LD HL, 0111H ;занести в регистровую пару HL число 111_{16}
LD BC, 0 ;обнулить содержимое регистровой пары BC
LD DE, 0 ;обнулить содержимое регистровой пары DE
LD A, 10 ;занести в аккумулятор число 10

MET1:

LD C, (HL) ;извлечь в регистр C информацию из ячейки
;памяти, адрес которой находится в HL
EX DE, HL ;обменять содержимое регистровых пар DE и HL
ADD HL, BC ;сложить содержимое регистровых пар BC и HL
EX DE, HL ;обменять содержимое регистровых пар DE и HL
INC HL ;инкрементировать содержимое пары HL
DEC A ;декрементировать (уменьшить на единицу)
;содержимое аккумулятора
JP NZ, MET1 ;если содержимое аккумулятора неравно нулю и
;флаг Z сброшен, то перейти к повторному
;выполнению фрагмента программы, начиная с
;метки MET1, если содержимое аккумулятора
;равно нулю, то выполнять следующую команду
EX DE, HL ;обменять содержимое регистровых пар DE и HL

Проверить значение содержимого ячейки с адресом 0923_{16} ; если оно больше
нуля, то прибавить к нему единицу, если меньше нуля — вычесть единицу, если
равно нулю — не изменять.

LD A, (0923H) ;занести в аккумулятор содержимое ячейки
;памяти с адресом 923_{16} , команда не устанавливает
;флаги
OR A ;для установки флагов логически сложим

		;содержимое аккумулятора само с собой,
		;при этом содержимое аккумулятора не изменится,
		;но будут расставлены флаги
JP Z, MET1		;если содержимое аккумулятора равно нулю, то
		;выйти из программы по метке MET1
JP P, MET2		;если содержимое аккумулятора больше нуля, то
		;перейти на метку MET2
DEC A		;декрементировать содержимое аккумулятора
JP MET3		;перейти на метку MET3
MET2:		
INC A		;инкрементировать содержимое аккумулятора
MET3:		
LD (0923H), A		;занести содержимое аккумулятора в ячейку
		;памяти с адресом 923 ₁₆ ,
MET1:		
Сравнить значения в регистрах В и С; если содержимое регистра В больше, то поменять значения местами.		
LD A, B		;сохранить содержимое регистра В в аккумуляторе
CP C		;вычесть из содержимого аккумулятора число,
		;хранящееся в регистре С, команда не сохраняет
		;результат, а только расставляет флаги
JP M, MET1		;если по флаг S сброшен, то выйти из фрагмента
LD B, C		;сохранить содержимое регистра С в регистре В
LD C, A		;сохранить содержимое аккумулятора в регистре С
MET1:		
Проинвертировать бит D3 в ячейке с адресом 0123 ₁₆ .		
LD A, (0123H)		;занести в аккумулятор содержимое ячейки
		;памяти с адресом 123 ₁₆ ,
LD B, A		;сохранить содержимое аккумулятора в регистре В
AND 11110111B		;обнулить бит D3 аккумулятора
LD C, A		;сохранить содержимое аккумулятора в регистре С
LD A, B		;восстановить в аккумуляторе ранее сохраненное
		;в регистре В значение
AND 00001000B		;обнулить все биты аккумулятора, кроме D3, его
		;состояние останется неизменным
CPL		;проинвертировать содержимое аккумулятора
AND 00001000B		;обнулить все биты аккумулятора, кроме D3
OR C		;логически сложить содержимое аккумулятора и
		;регистра С
LD (0123H), A		;занести содержимое аккумулятора в ячейку
		;памяти с адресом 123 ₁₆ ,
Или		
LD HL, 0123H		;занести в регистровую пару HL число 123 ₁₆
LD A, (HL)		;занести в аккумулятор содержимое ячейки

AND 00001000B	;памяти, адрес которой находится в HL
JP Z, MET1	;обнулить все биты аккумулятора, кроме D3
	;если флаг Z установлен, т. е. содержимое
	;аккумулятора равно нулю, то перейти на MET1,
RES 3, (HL)	;в противном случае сбросить бит D3 в ячейке
	;памяти, адрес которой находится в HL
JP MET2	;выйти из фрагмента по метке MET2
MET1:	
SET 3, (HL)	;установить бит D3 в ячейке памяти, адрес которой
	;находится в HL

MET2:

Сформировать программно временную задержку в 100 мкс.

Задача формирования временной задержки часто возникает при управлении тем или иным объектом. Процедура реализации временной задержки использует метод программных циклов. При этом в некоторый регистр загружается число, которое затем в каждом проходе цикла уменьшается на единицу. Так продолжается до тех пор, пока содержимое регистра не станет равным нулю, что обычно воспринимается как выход из цикла. Время задержки определяется произведением числа, записанного в регистр, на длительность выполнения одного прохода цикла, а также временем выполнения команд, собственно в цикл не входящих.

При формировании временной задержки необходима информация о тактовой частоте, на которой работает процессор. Это позволит определить длительность машинного такта и длительность выполнения МП каждой команды.

Пусть МП имеет тактовую частоту $f_{\text{ТАКТ}} = 2$ МГц, что соответствует периоду машинных тактов в 0,5 мкс. В простейшем случае цикл строится следующим образом.

LD C, число	;длительность 7 тактов или 3,5 мкс
MET1:	
DEC C	;длительность 4 такта или 2 мкс
JP NZ, MET1	;длительность 10 тактов или 5 мкс

Тогда длительность выполнения одного прохода цикла составляет $2 + 5 = 7$ мкс. Для формирования требуемой временной задержки в 100 мкс необходимо выпол-

нить $\frac{100 - 3,5}{7} \approx 13,8$ проходов цикла. Очевидно, что в регистр C следует помес-

тить целое число 13, что позволит сформировать задержку в $13 \cdot 7 + 3,5 = 94,5$ мкс. Недостающие 5,5 мкс могут быть получены, если в программу добавить команду

(команды) длительностью $\frac{5,5}{0,5} = 11$ тактов. Обычно для этой цели используется

пустая операция NOP, которая длится 4 такта. Можно использовать три таких команды, получив погрешность в 0,5 мкс, или одну пустую команду и одну команду длительностью 7 тактов, например уже использованную в программе. В этом случае временная задержка будет сформирована без погрешности. Полностью фрагмент показан ниже.

```

LD C,13
LD C,13
MET1:
DEC C
JP NZ, MET1

```

При программном формировании более длительных временных интервалов может возникнуть необходимость использования в качестве счетчика регистровой пары. В этом случае следует иметь в виду, что операции инкрементации и декрементации содержимого регистровых пар не влияют на флаги.

3 Построение микропроцессорной системы

3.1 Шины в реальной микропроцессорной системе

Как отмечалось выше, взаимодействие между модулями микропроцессорной системы происходит посредством сигналов, распространяющихся по трем шинам: адреса, данных и управления.

Шина адреса в микропроцессорной системе однонаправленная. Эта шина формируется из адресных линий используемого МП, для МП Z80 это линии A0...A15. Для формирования двунаправленной шины данных используются соответствующие линии МП, для МП Z80 это линии D0...D7.

К сформированным шинам адреса и данных подключаются соответствующие линии системных устройств. Т. к. линии адреса и данных МП имеют ограниченную нагрузочную способность (по отдаваемому в нагрузку току и емкости нагрузки) то между ними и линиями системных устройств могут устанавливаться так называемые шинные формирователи. Задача шинных формирователей — снизить нагрузку на линиях МП, обеспечив тем самым формирование в системе сигналов требуемой формы. По своей сути шинные формирователи являются буферными элементами с повышенной нагрузочной способностью.

Т. к. шина адреса однонаправленная, в качестве формирователя шины адреса могут выступать различные ИС, обеспечивающие передачу сигналов без инверсии. Возможный критерий выбора в этом случае, помимо требуемой нагрузочной способности,— использование для формирования шины с заданным количеством разрядов минимального количества буферных ИС. Стандартным решением при

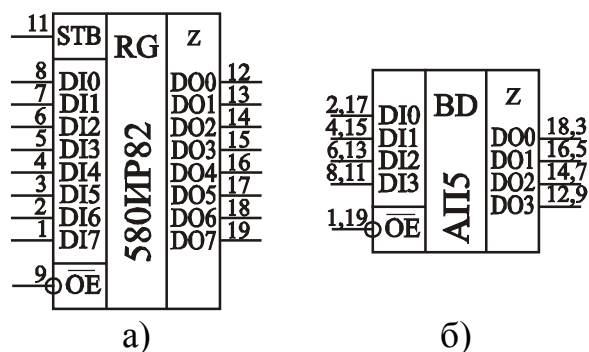


Рис. 15

формировании шины адреса является использование двух ИС КР580ИР82 — буферных регистров с прямыми выходами, рис. 15,а.

Данные в регистр записываются при STB = 1, в противном случае регистр находится в режиме хранения информации. Сигнал OE = 1 переводит выходные линии регистра в высокоимпедансное состояние.

Буферный регистр КР580ИР82 обеспечивает выходной ток до 32 мА и допускает подключение нагрузки емкостью до 300 пФ. Если шина нагружена слабее, то можно использовать другие ИС, например, КР1533АП5, рис. 15,б. Данная ИС представляет собой сдвоенный 4-разрядный шинный драйвер (BD — Bus driver), поэтому для формирования 16-разрядной шины потребуется 2 такие ИС. Эта ИС отличается от КР580ИР82 меньшей нагрузочной способностью (до 24 мА на нагрузку емкостью до 50 пФ), но сама потребляет существенно меньший, чем КР580ИР82, ток.

При использовании рассмотренных буферных ИС на их управляющий вход \overline{OE} следует подать сигнал низкого уровня, т. к. в большинстве случаев переводить шину адреса в высокоимпедансное состояние со стороны МП не требуется. Исключение составляют случаи, когда в системе предусмотрено управление шиной со стороны другого устройства (захват шин). В этом случае необходимо обеспечить перевод формирователя в высокоимпедансное состояние сигналом, подтверждающим освобождение шины МП — для МП Z80 это сигнал $\overline{BUSAК}$.

Для буферирования шины данных используют ИС приемопередатчиков (Transceiver, TR), название которых подчеркивает возможность двунаправленной передачи информации. В микропроцессорный набор КР580 входит ИС КР580ВА86 — 8-ми разрядный шинный формирователь, рис. 16,а. Направление передачи информации через эту ИС определяется сигналом на входе Т — при $T=1$ информация передается с линий А на линии В, в противном случае информация передается в обратном направлении. Сигнал $\overline{OE}=1$ переводит все линии данных в высокоимпедансное состояние. Нагрузочная способность ИС по линиям А и В различна: линии А способны обеспечить ток до 16 мА и работать на нагрузку емкостью 100 пФ, линии В — 32 мА и 300 пФ. При неизменяющемся сигнале на входе Т рассматриваемый шинный формирователь будет работать в однонаправленном режиме и может использоваться для буферирования шины адреса.

Другой ИС, которую можно использовать для формирования сигналов на шине данных, является КР1533АП6, рис. 16,б. По сигналам она идентична КР580ВА86, обеспечивает ток 24 мА как по линиям А, так и по линиям В.

Очевидно, что в микропроцессорной системе на линии Т рассмотренных ИС должен подаваться сигнал, определяющий направления протекания информации по шине данных, т. е. сигнал \overline{RD} или \overline{WR} . Выбор конкретного сигнала определяется подключением ИС приемопередатчика. Если его линии А подключены к МП, а к линиям В подключены соответствующие выходы системных устройств, то управление приемопередатчиком должен осуществлять сигнал \overline{RD} .

Если не требуется переводить ИС приемопередатчика в высокоимпедансное состояние, то на его вход \overline{OE} следует подать сигнал низкого уровня.

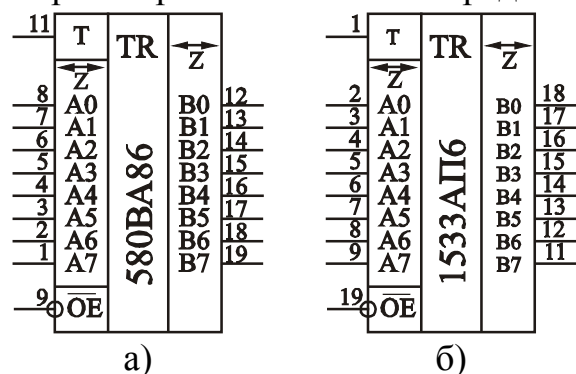


Рис. 16

Необходимость установки в микропроцессорной системе шинных формирователей определяется из анализа нагруженности линий адреса и данных используемого МП.

Как отмечалось ранее, шина управления микропроцессорной системы должна содержать сигналы $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, $\overline{\text{IOW}}$. Эти сигналы управляют операциями чтения и записи данных при работе с памятью и периферийными устройствами в соответствии с временными диаграммами работы выбранного устройства. Данные сигналы должны формироваться из сигналов управления используемого микропроцессора. Применительно к микропроцессорной системе на основе МП Z80 сигнал $\overline{\text{MEMR}}$ должен иметь активный (низкий) уровень, когда имеют низкий уровень сигналы $\overline{\text{MREQ}}$ и $\overline{\text{RD}}$, т. е. $\overline{\text{MEMR}} = \overline{\text{MREQ}} + \overline{\text{RD}}$. Аналогично можно записать, что $\overline{\text{IOR}} = \overline{\text{IOEQ}} + \overline{\text{RD}}$, $\overline{\text{MEMW}} = \overline{\text{MREQ}} + \overline{\text{WR}}$, $\overline{\text{IOW}} = \overline{\text{IOEQ}} + \overline{\text{WR}}$.

Сформированные стандартные сигналы шины управления подаются на управляющие входы ИС памяти и периферийных устройств: сигнал $\overline{\text{MEMR}}$ на вход $\overline{\text{OE}}$ ПЗУ и ОЗУ, $\overline{\text{MEMW}}$ на вход $\overline{\text{WR}}$ ОЗУ, сигналы $\overline{\text{IOR}}$ и $\overline{\text{IOW}}$ на входы $\overline{\text{RD}}$ и $\overline{\text{WR}}$ периферийных устройств соответственно. Дополнительная буферизация полученных стандартных сигналов не требуется.

3.2 Разделение адресного пространства

3.2.1 Адресация ИС памяти

Как отмечено выше, МП Z80 может адресоваться к 64 кбайтам внешней памяти и к 256 периферийным устройствам. Т. е. для данного МП существует два адресных пространства — адресное пространство памяти и адресное пространство периферийных устройств. Эти пространства не перекрываются, т. к. при обращении к памяти и периферийным устройствам используются различные сигналы — $\overline{\text{MREQ}}$ и $\overline{\text{IORQ}}$ соответственно, поэтому могут рассматриваться независимо.

Рассмотрим вопрос подключения к МП ИС памяти. Т. к. в системе присутствует память двух типов, то необходимо разделить полное адресное пространство памяти на область ПЗУ и область ОЗУ. Деление пространства осуществляется соответствующей коммутацией линий шины адреса микропроцессорной системы. Т. к. каждая линия определяет соответствующий двоичный разряд, то в целях упрощения коммутации следует выбирать объем памяти равным 2^N , где N — целое число. Проиллюстрировать деление адресного пространства можно с помощью карты распределения памяти. На карте памяти записывают адреса областей, относящихся к ПЗУ и ОЗУ, рис. 17.

Карты соответствуют системе, имеющей ПЗУ объемом 16 кбайт и ОЗУ объемом 8 кбайт, но размещение ОЗУ в этих картах различно. В связи с тем, что МП Z80 после сброса сигналом $\overline{\text{RESET}}$ начинает выполнять управляющую программу начиная с нулевого адреса, то местоположение ПЗУ в адресном пространстве фиксировано — постоянная память должна начинаться с нулевого адреса. УГО ИС памяти, которые могут быть использованы в системе, показаны на рис. 3. От-

метим, что ИС ПЗУ имеет 14 адресных линий, а ИС ОЗУ — 13. Очевидно, что эти линии должны быть подключены к соответствующим адресным линиям МП, тем самым будет организована шина адреса микропроцессорной системы. Для выбора конкретной ячейки в ИС ПЗУ должны использоваться 14 младших линий шины адреса, а для выбора конкретной ячейки в ОЗУ — 13 младших линий. Оставшиеся старшие линии могут использоваться для выбора либо ИС ПЗУ, либо ИС ОЗУ.

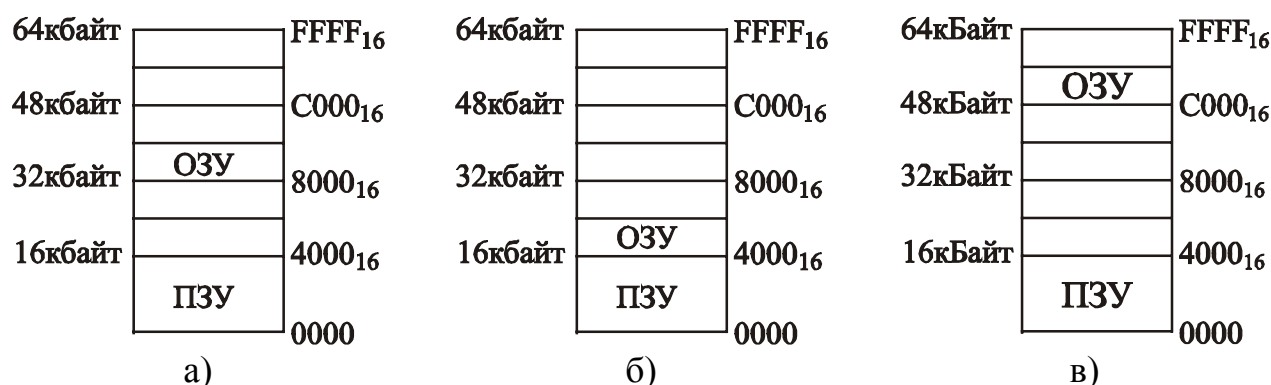


Рис. 17

Рассмотрим карту памяти, показанную на рис. 17,а. Запишем 16-разрядные адреса областей памяти, в которых размещаются ПЗУ и ОЗУ: ПЗУ от 0000000000000000_2 до 0011111111111111_2 , ОЗУ от 1000000000000000_2 до 1001111111111111_2 . Очевидно, что ИС ПЗУ должна выбираться, когда сигналы на линиях A15 и A14 шины адреса равны нулю. Для выбора ИС ОЗУ требуется несколько иная комбинация сигналов на старших линиях шины адреса: A15=1, A14=0, A13=0.

В качестве устройства, выбирающего ИС ПЗУ или ОЗУ в зависимости от состояния старших линий шины адреса, выступает селектор (дешифратор) адреса. Он может быть реализован, например, на логических элементах. Спроектируем данную логическую схему. Составим таблицу истинности для трех входных (A15, A14, A13) и двух выходных ($\overline{CS}_{ПЗУ}$, $\overline{CS}_{ОЗУ}$) переменных при условии, что ИС выбираются низким уровнем, табл. 5. Логические функции для выходных переменных, составленные по таблице истинности, будут иметь следующий вид:

$$\overline{CS}_{ПЗУ} = (A15 + A14 + A13)(A15 + A14 + \overline{A13}),$$

Таблица 5

$\overline{CS}_{ОЗУ} = (\overline{A15} + A14 + A13)$. На рис. 18,а показана схема на логических элементах, реализующая функцию выбора ИС памяти.

Селектор адреса можно выполнить на ИС дешифратора с тремя входами, например, КР1533ИД7, рис. 18,б. Особенность этой ИС являются инверсные выходные сигналы, что позволяет подключать выходные линии дешифратора к входам выбора кристалла ИС памяти непосредственно, без использования ин-

A15	A14	A13	$\overline{CS}_{ПЗУ}$	$\overline{CS}_{ОЗУ}$
0	0	0	0	1
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

верторов. Очевидно, что к селектору адреса, показанному на рис. 18,б, при необходимости легко могут быть подключены еще несколько ИС памяти. Схема на логических элементах такой гибкостью не отличается — введение дополнительной ИС памяти потребует перепроектирования схемы.

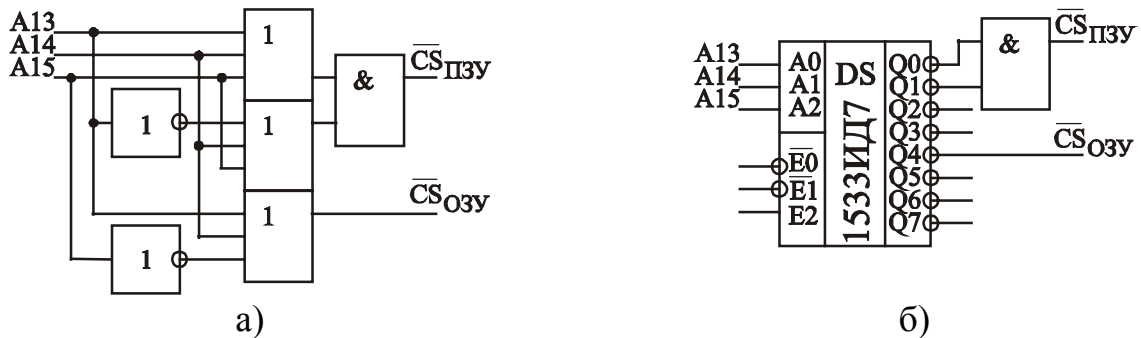


Рис. 18

В качестве селектора адреса можно использовать ИС ПЗУ. При этом адресные входы ПЗУ будут использоваться как входные линии селектора, а линии данных будут выполнять роль выходных линий селектора. Собственно процесс выбора будет определяться информацией, занесенной в ячейки памяти ПЗУ. Реализуем селектор на ПЗУ объемом 16 кбайт, рис. 19.

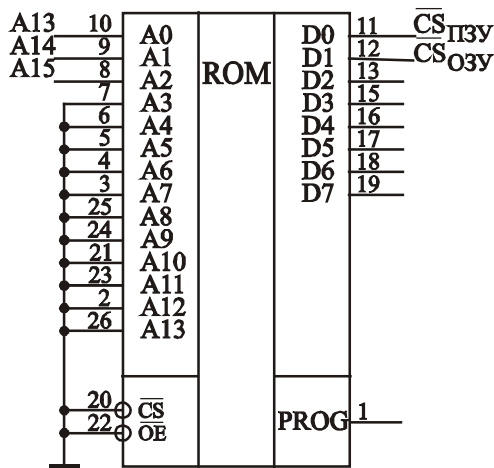


Рис. 19

Т. к. нас интересуют только три состояния линий шины адреса, то при показанном на рис. 19 подключении ПЗУ выбирающая ИС памяти информация должна находиться в ячейках с адресами (внутренними адресами ячейки памяти в ПЗУ) $0000000000000_2=0$, $0000000000001_2=1$, $0000000000100_2=4$. Информация в первых двух ячейках должна выбирать ПЗУ и не выбирать ОЗУ. Т. к. выбор осуществляется низким уровнем, а сигнал выбора ПЗУ снимается с линии D0, то в эти ячейки должно быть занесено двоичное число 1111110_2 . Соответственно, в ячейку с адресом 4 должно быть занесено двоичное число 1111101_2 , что позволит при появлении на стар-

ших линиях шины адреса комбинации $A15=1, A14=0, A13=0$ осуществить выбор ИС ОЗУ. Во все остальные ячейки ПЗУ следует занести числа 1111111_2 , это гарантирует правильный выбор ИС памяти при любых сигналах на шине адреса.

Селектор адреса, реализованный на ПЗУ, допускает подключение дополнительных выбираемых ИС (до восьми) без перепроектирования схемы, однако требует перепрограммирования ПЗУ.

Селекторы адреса для карт памяти, показанных на рис. 17,б,в, принципиально от рассмотренных не отличаются.

В ряд случаев построение селектора адреса можно существенно упростить, если сделать некоторые допущения о работе микропроцессорной системы. В корректно работающей системе области адресного пространства, к которому реально

обращается МП, жестко заданы. Если, например, управляющая программа имеет объем 14 кбайт и находится в ИС ПЗУ объемом 16 кбайт, то в корректно работающей системе МП никогда не обратится к 2 старшим кбайтам ИС ПЗУ. Следовательно, для МП нет принципиальной разницы между незадействованной областью адресного пространства с адресами от 4000_{16} до $7FFF_{16}$ (рассматривается карта памяти на рис. 17,а) и неиспользуемым пространством ПЗУ — и к той, и к другой области памяти МП никогда не обращается. Следовательно, можно считать, что вся область с адресами от нулевого до $7FFF_{16}$ является областью ПЗУ, но задействованы в работе только младшие 14 кбайт.

Аналогично, что область памяти выше адреса 8000_{16} относится к ОЗУ, но используется только 8 кбайт (реально может использоваться существенно меньше). Следовательно, адресной линией, делящей адресное пространство на области ПЗУ и ОЗУ будет являться старшая линия шины адреса — при $A_{15}=0$ должна выбираться ИС ПЗУ, при $A_{15}=1$ — ИС ОЗУ. Селектор адреса в данном случае строится следующим образом: вывод \overline{CS} ИС ПЗУ подключается к линии A_{15} шины адреса непосредственно, а вывод \overline{CS} ИС ОЗУ — через инвертор. Помимо простоты, данный способ позволяет легко наращивать объем ПЗУ и ОЗУ путем установки ИС большего объема — вплоть до 32 кбайт под каждый тип памяти без перепроектирования схемы селектора.

Аналогично разделение адресного пространства можно упростить для карты памяти, показанной на рис. 17,б. Но для карты, показанной на рис. 17,в, для разделения памяти ПЗУ и ОЗУ потребуется использовать линию A_{14} шины данных, объем ОЗУ заменой ИС может быть в этом случае увеличен до 16 кбайт.

Как показано выше, деление адресного пространства пополам на область ПЗУ и ОЗУ легко осуществимо на практике. Однако редко требуется иметь в микропроцессорной системе управления равный объем ОЗУ и ПЗУ. На практике редко требуется ОЗУ объемом более 8 кбайт; всю остальную область адресного пространства целесообразно отвести под ПЗУ. Рассмотрим, как может быть построен селектор адреса для системы, в которой ОЗУ имеет объем 2 кбайта, а под ПЗУ отведено оставшиеся 62 кбайта. Чтобы получить требуемый объем ПЗУ необходимо задействовать ИС объемом 32, 16, 8, 4, 2 кбайта. Нецелесообразность такого подхода очевидна, гораздо проще использовать, например, 4 ИС ПЗУ объемом 16 кбайт или 2 ИС ПЗУ объемом 32 кбайта, или 1 ИС ПЗУ объемом 64 кбайта, не используя 2 кбайта, выделенные под ОЗУ. Пусть требуется использовать ИС ПЗУ объемом 32 кбайта и под ОЗУ отведены старшие 2 кбайта адресного пространства. Следовательно, область ПЗУ располагается по адресам от нулевого до 0111111111111111_2 — для первой ИС ПЗУ, и от 1000000000000000_2 до 1111011111111111_2 — для второй ИС ПЗУ, область ОЗУ располагается по адресам от 1111100000000000_2 до 1111111111111111_2 . Для разделения адресного пространства требуется использовать 5 старших линий шины адреса: при $A_{15}=0$ должна выбираться первая ИС ПЗУ, при $A_{15}=A_{14}=A_{13}=A_{12}=A_{11}=1$ должна выбираться ИС ОЗУ, в остальных случаях должна выбираться вторая ИС ПЗУ. Использовать ИС дешифратора в рассматриваемом случае затруднительно, т. к. де-

шифратор с 5 входами отечественной промышленностью не выпускается. Селектор в данной системе легко может быть реализован с использованием ИС ПЗУ. Реализуем селектор адреса на логических элементах. Логические функции для сигналов выбора первой ИС ПЗУ и ИС ОЗУ будут иметь следующий вид: $\overline{CS}_{ПЗУ1} = A15$, $\overline{CS}_{ОЗУ} = \overline{A15} + \overline{A14} + \overline{A13} + \overline{A12} + \overline{A11} = \overline{A15 \cdot A14 \cdot A13 \cdot A12 \cdot A11}$. Т. к. вторая ИС ПЗУ должна выбираться, когда не выбрана ни первая ИС ПЗУ, ни ИС ОЗУ, то логическая функция для сигнала выбора второй ИС ПЗУ запишется как $\overline{CS}_{ПЗУ2} = \overline{\overline{CS}_{ПЗУ1} + \overline{CS}_{ОЗУ}} = \overline{\overline{CS}_{ПЗУ1} \cdot \overline{CS}_{ОЗУ}}$. Схема селектора адреса на логических элементах показана на рис. 20.

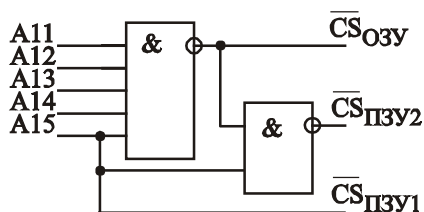


Рис. 20

Селектор адреса можно реализовать с использованием ИС цифровых компараторов: сравнивают состояние шины адреса с некоторым заданным числом. Однако на практике использование таких ИС нецелесообразно, т. к. схема селектора становится громоздкой — для построения селектора для 16-разрядной шины адреса потребуется 4 ИС четырехвходовых цифровых компараторов.

Выбор способа реализации селектора адреса — на логических элементах или с использованием ИС дешифраторов или ПЗУ зависит от сложности реализации схемы для заданной карты распределения памяти, а также от требуемого быстродействия схемы. Несмотря на то, что схемы с использованием ПЗУ наиболее просты и универсальны, по быстродействию они несколько проигрывают схемам на логических элементах. В ряде случаев это может оказаться решающим фактором при выборе схемы реализации селектора адреса.

3.2.2 Адресация периферийных устройств

Для обращения к периферийным устройствам МП Z80 использует 8 младших линий шины адреса, что позволяет ему взаимодействовать с 256 периферийными устройствами. Обычно такого большого количества периферийных устройств в системах управления не требуется.

Принципиальной разницы при обращении к конкретной ячейке памяти и к периферийному устройству нет. И в том, и в другом случае из сигналов, присутствующих на шине адреса должен быть сформирован сигнал доступа к интересующему устройству, т. е. должен быть реализован селектор адреса.

Пусть в микропроцессорной системе задействованы 4 периферийных устройства, выбираемых низким уровнем сигнала. Назначим адреса этим устройствам: первому — 00000000_2 , второму — 00000001_2 , третьему — 00000010_2 , четвертому — 00000011_2 . Отметим, что назначенные адреса не являются единственными — применительно к МП Z80 адресом периферийного устройства может служить любое 8-разрядное двоичное число.

При назначенных адресах селектор адреса легко реализуется с использованием ИС дешифратора, например, КР1533ИД7, рис. 21,а. Такой способ обращения к периферийному устройству является наиболее общим. Очевидно, что реализовав

дешифратор с 8 входами, можно обеспечить формирование сигналов выбора для всех 256 периферийных устройств, которые могут быть подключены к МП Z80.

При небольшом (до восьми) количестве периферийных устройств грамотный выбор адресов этих устройств позволяет вообще отказаться от использования селектора адреса. Для иллюстрации сказанного изменим адреса периферийных устройств: пусть первое устройство имеет адрес 00000001_2 , второе — 00000010_2 , третье — 00000100_2 , четвертое — 00001000_2 . Если устройства выбираются низким уровнем сигнала, то схеме их подключения к шине адреса будет иметь вид, показанный на рис. 21,б. Если же устройства выбираются высоким уровнем сигнала, то инверторы из схемы исключаются.

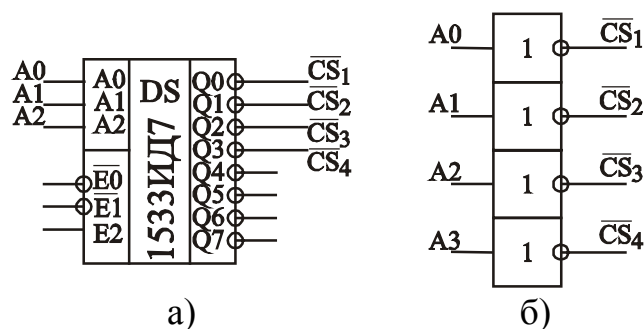


Рис. 21

От использования инверторов можно отказаться и для периферийных устройств, выбираемых низким уровнем, если принять для них адреса, инверсные к заданным выше: первое устройство будет иметь адрес 11111110_2 , второе — 11111101_2 , третье — 11111011_2 , четвертое — 11110111_2 . В этом случае входы разрешения работы периферийных устройств подключаются непосредственно к соответствующим линиям шины адреса — первого устройства к линии A0, второго — к A1, третьего — к A2, четвертого — к A3. Данный способ подключения называется линейным, каждая линия шины адреса в этом случае отвечает за выбор одного периферийного устройства. Способ наиболее просто реализуем на практике, но имеет ограничение по числу подключаемых устройств.

Отметим особенность назначения адресов устройств при линейном подключении. В рассмотренном примере было задействовано 4 периферийных устройства, которые выбирались 4 младшими линиями шины адреса. В общем случае состояния линий A4...A7 на выбор этих устройств не влияет, поэтому адреса можно было задать и иначе: первое устройство, например, могло иметь адрес 10001110_2 , второе — 01001101_2 , третье — 00101011_2 , четвертое — 00000111_2 . Однако целесообразно при назначении адреса назначить невливающим разрядам значения, соответствующие неактивному состоянию сигнала выбора — если периферийные устройства выбираются низким уровнем сигнала, то невливающим разрядам следует присвоить значение единицы. Это позволит в дальнейшем линейно подключать к системе дополнительные периферийные устройства без корректировки адресов ранее подключенных устройств.

Некоторые ИС содержат в себе несколько периферийных устройств. Примером может служить ранее рассмотренный ППА КР580ВВ55А, ИС которого содержит 4 устройства. Выбор собственно ИС осуществляется сигналом \overline{CS} , конкретное устройства (внутренний регистр) из совокупности выбирается встроенным дешифратором по сигналам A0 и A1. Такие ИС по способу адресации относятся к коммутируемым периферийным устройствам.

При подключении коммутируемых периферийных устройств линии шины адреса разделяют на две группы: линии выбора самой ИС и линии выбора внутренних регистров. Количество линий в каждой группе определяется количеством выбираемых внутренних регистров в устройстве. При подключении ИС КР580ВВ55А можно, например, линию А2 шины адреса использовать для формирования сигнала выбора ИС \overline{CS} , а линии А1 и А0 — для выбора конкретного внутреннего регистра ИС. В этом случае адрес порта А будет 1111000_2 , порта В — 1111001_2 , порта С — 1111010_2 , регистра СW — 1111011_2 . Если потребуется подключить еще одну ИС ППА, то для ее выбора может использоваться линия А3 шины адреса. Для выбора конкретного внутреннего регистра также можно использовать линии А1 и А0; адрес порта А второй ИС ППА будет 1110100_2 , порта В — 1110101_2 , порта С — 1110110_2 , регистра СW — 1110111_2 . Такой способ подключения коммутируемых устройств, когда за выбор каждой ИС отвечает одна из линий шины адреса, является разновидностью линейного подключения. Очевидно, что в этом случае имеется ограничение на количество подключаемых ИС — к МП Z80 можно этим способом только 6 ИС ППА КР580ВВ55А. Если требуется подключить большее количество коммутируемых устройств, то следует воспользоваться дешифраторами.

3.2.3 Адресация в системах с единым адресным пространством

Некоторые МП имеют единое адресное пространство, в котором при проектировании системы необходимо разместить как ПЗУ и ОЗУ, так и все периферийные устройства. Напомним, что в МП Z80 адресные пространства памяти и периферийных устройств разделены тем, что при обращении к памяти и периферийным устройствам используются различные сигналы — \overline{MREQ} и \overline{IORQ} соответственно. МП с единым адресным пространством формируют, как правило, только два сигнала, управляющих обменом с внешними устройствами \overline{RD} и \overline{WR} . В этом случае для формирования набора стандартных сигналов шины управления \overline{MEMR} , \overline{MEMW} , \overline{IOR} , \overline{IOW} задействуют линии шины адреса. Тем самым единое адресное пространство принудительно разделяют на область памяти и область периферийных устройств. Т. к. ПЗУ обычно располагают начиная с нулевого адреса, то область периферийных устройств располагают в верхней части адресного пространства.

Разделение единого пространства на области памяти и периферийных устройств предполагает построение селектора адреса, сигнал которого будет использован для формирования стандартного набора сигналов шины управления. Рассмотрим пример. Пусть в системе с единым адресным пространством в 64 кбайта требуется использовать два ППА КР580ВВ55А, ОЗУ объемом 4 кбайта, остальную область адресного пространства отвести под ПЗУ. Отведем под область периферийных устройств старшие 256 Байт адресного пространства, обеспечивая тем самым возможность использования, при необходимости, до 64 ППА. Для организации ПЗУ задействуем две ИС объемом 32 кбайта каждая, а для организации

ОЗУ — 2 ИС по 2 кбайта. Карта распределения адресного пространства в этом случае будет иметь вид, показанный на рис. 22,а.

Адреса, по которым располагаются системные устройства, следующие: ПЗУ1 — от нулевого до 0111111111111111_2 , ПЗУ2 — от 1000000000000000_2 до 1110111011111111_2 , ОЗУ1 — от 1110111100000000_2 до 1111011011111111_2 , ОЗУ2 — от 1111011100000000_2 до 1111111011111111_2 , периферийные устройства — от 1111111100000000_2 до 1111111111111111_2 . Очевидно, что селектор адреса, используемый для деления единого адресного пространства на области памяти и периферийных устройств, должен анализировать состояние 8 старших линий шины адреса — когда сигналы на этих линиях равны высокому уровню, предполагается обращение МП к периферийному устройству, в противном случае МП обращается к памяти. На рис. 22,б показано, как сформировать стандартные сигналы шины управления при использовании рассмотренного селектора адреса.

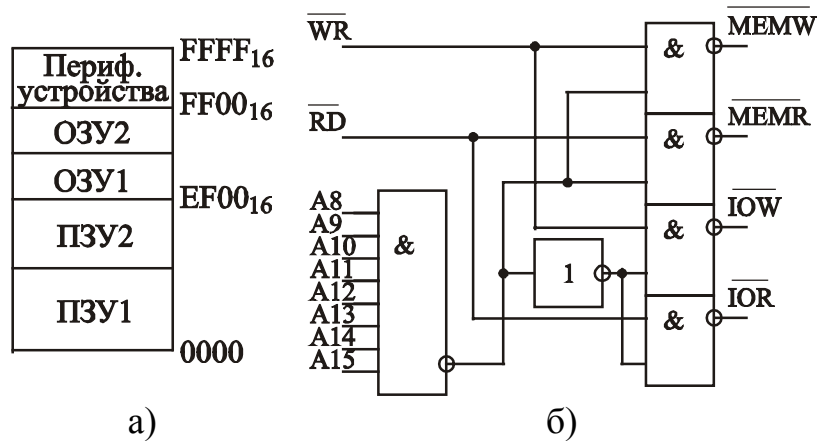


Рис. 22

Далее необходимо назначить адреса ИС ППА, например, следующим образом: порту А ППА1 — 111111111111000_2 , порту В ППА1 — 111111111111001_2 , порту С ППА1 — 111111111111010_2 , регистру СW ППА1 — 111111111111011_2 , порту А ППА2 — 1111111111110100_2 , порту В ППА2 — 1111111111110101_2 , порту С ППА2 — 1111111111110110_2 , регистру СW ППА2 — 1111111111110111_2 , и спроектировать селекторы адреса, выбирающие то или иное системное устройство.

Рассмотренный способ адресации может быть реализован и применительно к МП Z80. В этом случае сигналы \overline{MREQ} и \overline{IORQ} не используются.

3.3 Контроллер на базе микропроцессора Z80

Основываясь на изложенном выше материале спроектируем микропроцессорную систему на базе МП Z80. Система должна иметь 8 кбайта ОЗУ, 1 ППА КР580ВВ55А. Остальной объем адресного пространства используем под ПЗУ.

В целях упрощения проектирования используем в качестве ПЗУ одну ИС 27С512 объемом 64 кбайта. В качестве ОЗУ используем ИС КР537РУ17 объемом 8 кбайта.

Отведем под ПЗУ старшие 8 кбайта адресного пространства от адреса от 1110000000000000_2 до адреса 1111111111111111_2 . Тогда область ПЗУ будет располагаться от нулевого адреса до адреса 1101111111111111_2 . Для разделения адресного пространства используем 3 старших линий шины адреса: при $A15=A14=A13=1$ должна выбираться ИС ОЗУ, в остальных случаях должна вы-

бираться ИС ПЗУ. Реализуем селектор адреса на логических элементах. Логическая функция для сигналов выбора ИС ОЗУ будет иметь следующий вид: $\overline{CS_{\text{ОЗУ}}} = A_{15} + A_{14} + A_{13} = \overline{A_{15} \cdot A_{14} \cdot A_{13}}$. Т. к. ИС ПЗУ должна выбираться, когда не выбрана ИС ОЗУ, то сигнал выбора ПЗУ будет инверсен сигналу выбора ОЗУ.

Назначим периферийным устройствам следующие адреса: порту А ППА — 11111000_2 , порту В — 11111001_2 , порту С — 11111010_2 , регистру СW — 11111011_2 . При адресации ППА дополнительные ИС не использованы.

Проанализируем нагруженность линий адреса и данных МП. Эти линии способны обеспечить ток в 250 мкА в состоянии логической единицы и 1,8 мА в состоянии логического нуля. Емкость нагрузки каждой линии адреса не должна превышать 100 пФ, данных — 200 пФ. К шинам адреса и данных подключены соответствующие линии ИС 27C512, КР537РУ10, КР580ВВ55А. Т. к. эти ИС выполняются по n-MOP технологии, то отличаются малым (единицы мкА) током, потребляемым по линиям управления. Емкость входов этих ИС лежит в пределах 10...20 пФ. Таким образом, в использовании шинных формирователей нет необходимости. Схема подключения ИС памяти и ППА к МП Z80 показана на рис. 23.

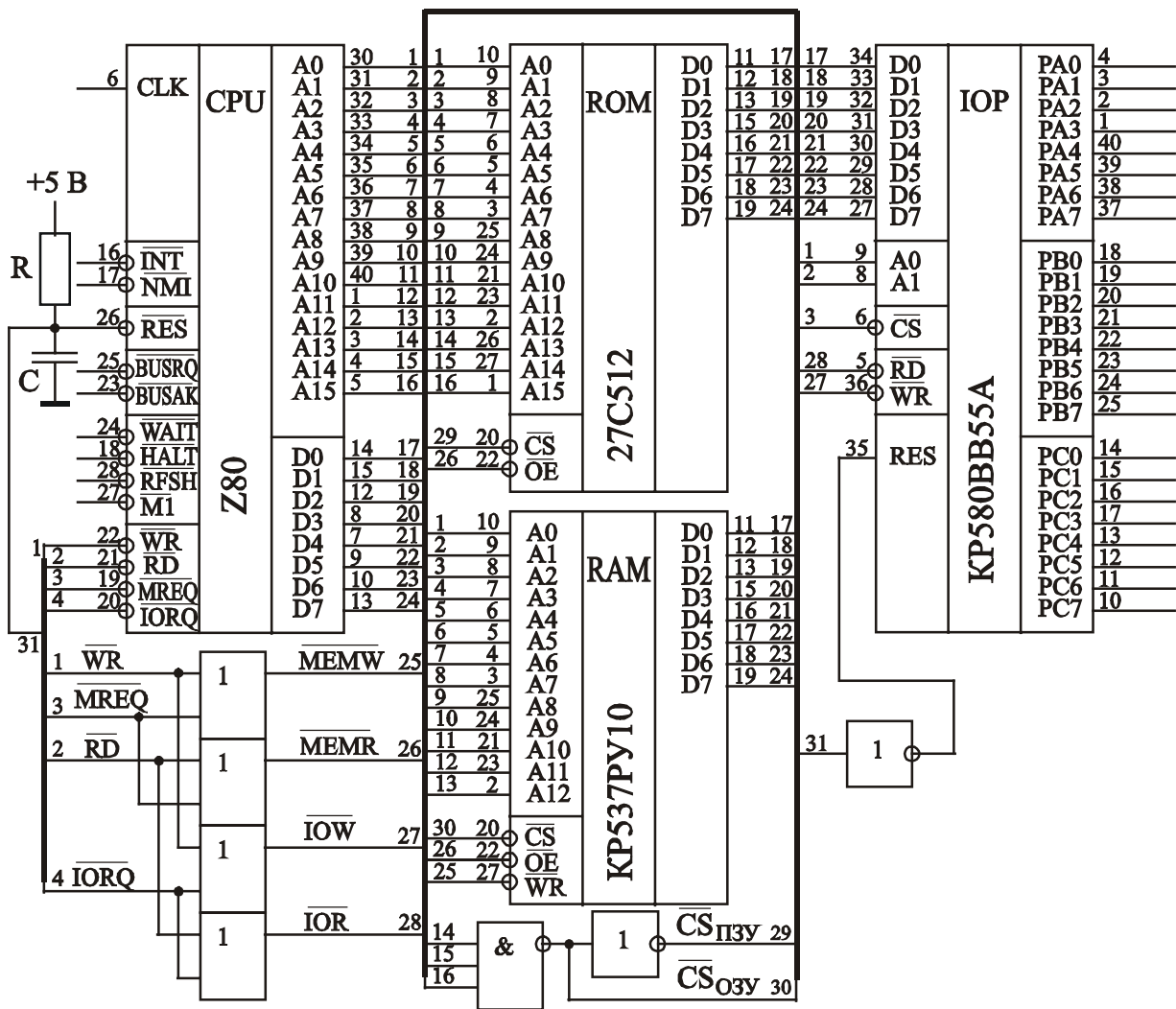


Рис. 23

На данной схеме не показаны цепи формирования тактирующих работу МП импульсов. На неиспользуемых в данном включении входах МП (\overline{INT} , \overline{NMI} , \overline{BUSRQ} , \overline{WAIT}) должны быть сформированы пассивные уровни. Сброс МП и ППА осуществляется RC цепью при подаче питания на схему. Параметры компонентов этой цепи выбирают из условия обеспечения требуемой длительности импульса сброса. Например, при частоте тактовых импульсов МП $f_{ТАКТ} = 2$ МГц длительность импульса сброса должна быть $T_{СБР} > \frac{3}{f_{ТАКТ}} = 1,5$ мкс, и можно выбрать $C = 1$ мк и $R = 10$ кОм — $RC = 10$ мкс $> 1,5$ мкс.

4 Прерывания в микропроцессорных системах

Прерыванием называется изменение порядка выполнения управляющей программы, вызванное воздействием на МП со стороны внешнего устройства. На рис. 24,а показан процесс выполнения некоторой управляющей программы, состоящей из пяти команд.

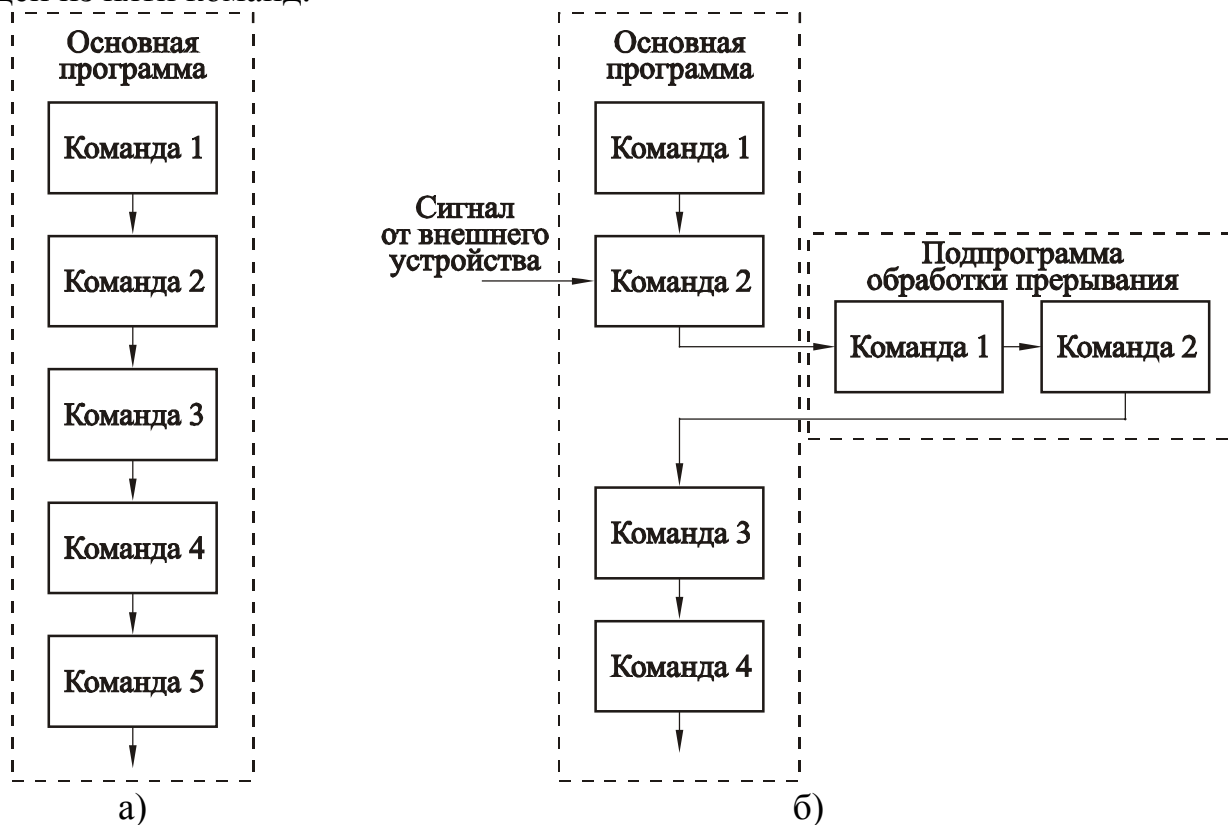


Рис. 24

При отсутствии прерывания команды выполняются последовательно, от первой до пятой. Система прерываний позволяет изменять ход выполнения программы на основании сигналов, поступающих в МП, рис. 24,б. Первые две команды программы выполняются МП так же, как и при непрерывном последовательном ее выполнении. Затем в МП поступает сигнал от внешнего устройства, называемый запросом прерывания. Если выполнение прерывания разрешено, то по за-

вершении выполнения текущей команды основной программы МП преступает к выполнению первой команды подпрограммы обработки прерывания. Как эта подпрограмма будет выполнена, МП вернется к прерванной основной программе и продолжит ее нормальное выполнение.

В показанном примере прерывание основной программы происходит между 2-й и 3-й командами. Однако оно могло бы произойти и между другими командами. Вообще, сигнал запроса прерывания, поступающий в МП, является асинхронным, т. е. он может прийти на любом этапе работы основной программы. Для системы не имеет значения, когда наступает прерывание, т. к. для обработки конкретного прерывания всегда используется одна подпрограмма. Адрес ячейки памяти, в которой располагается первая команда подпрограммы обработки прерывания, называется вектором этого прерывания. Переход МП к выполнению подпрограммы обработки прерывания называют переходом по вектору. Для некоторых МП векторы прерывания принято называть векторами рестарта с введением аббревиатуры RST.

Во МП используются прерывания двух типов: маскируемые и немаскируемые. Если прерывание является немаскируемым, то МП будет реагировать всякий раз, когда появится сигнал на его запрос. Реакция МП на маскируемое прерывание может быть разрешена или запрещена программными средствами.

Рассмотрим принципы взаимодействия МП и внешнего устройства, способного выдавать на МП (запрашивать) сигнал прерывания. В общем случае это взаимодействие состоит из следующих этапов.

1) Внешнее устройство подает на вход запроса прерывания МП сигнал активного уровня — запрашивает прерывание.

2) МП принимает запрос прерывания и выставляет сигнал подтверждения прерывания.

3) Получив сигнал подтверждения прерывания, внешнее устройство передает в МП вектор прерывания и снимает свой запрос.

4) МП прекращает выполнение основной программы и начинает выполнять подпрограмму прерывания, расположенную по полученному от внешнего устройства вектору.

Описанное взаимодействие может иметь свои особенности для конкретных МП и внешних устройств. Например, запрос прерывания от внешнего устройства может изначально формироваться в виде короткого импульса, т. е. не требуется использования сигнала подтверждения прерывания для снятия запроса прерывания внешним устройством. У некоторых МП линия подтверждения запроса прерывания отсутствует и этот сигнал может быть сформирован только при помощи внешних элементов. В ряде случаев передача вектора прерывания не требуется, т. к. адрес первой ячейки подпрограммы обработки прерывания жестко задан.

При реализации описанного взаимодействия необходимо обеспечить недопустимость поступления запроса на прерывание от внешнего устройства в период времени, когда идет выполнение подпрограммы обработки прерывания, запрошенного этим же внешним устройством. Наиболее просто это может быть реализовано для маскируемого прерывания: в начале подпрограммы обработки прерыва-

ния программно запрещается реагирование МП на это прерывание; в конце работы этой подпрограммы реагирование разрешается.

Совокупность прерываний, которые могут быть реализованы в устройстве, называют системой прерываний. Особое значение система прерываний имеет для систем управления. Системы управления относятся к системам реального времени, которые предполагают совершенно конкретные, рассчитываемые на этапе разработки интервалы времени реакции на внешние события. Задержки сверх расчетных здесь просто недопустимы, т. к. они могут приводить к катастрофическим последствиям.

Проблема быстрой реакции на события решается организацией системы прерываний. Она подразумевает, что для каждого такого события разрабатывается отдельная подпрограмма, который формирует реакцию устройства на него. При наступлении события должен формироваться запрос прерывания, который вызовет соответствующую подпрограмму.

Важной характеристикой системы прерываний является организация приоритетов прерываний. Понятие приоритет означает, что выполняемая подпрограмма прерывания может быть прервана другим событием только при условии, что оно имеет более высокий приоритет, чем текущее. В ином случае МП перейдет к обработке прерывания от нового события после окончания обработки предыдущего.

Для организации системы прерываний в микропроцессорной системе могут использоваться специальные ИС — контроллеры прерываний, например, КР580ВГ59. Однако современные МП сами имеют мощные системы прерываний.

Рассмотрим систему прерываний МП Z80. Как отмечено ранее, МП Z80 имеет два входа, предназначенные для приема запросов на прерывание: запрос на прерывание \overline{INT} и немаскируемый запрос на прерывание \overline{NMI} .

Немаскируемое прерывание имеет вектор рестарта, равный 066_{16} . Это значит, что, когда поступит сигнал запроса немаскируемого прерывания, МП начнет выполнение подпрограммы обработки прерывания, первая команда которой находится по указанному адресу. Переход на подпрограмму обработки немаскируемого прерывания в МП Z80 осуществляется следующим образом, рис. 25. Внутри МП вход \overline{NMI} соединен с триггером, состояние которого может быть изменено импульсным сигналом длительностью не менее 80 нс, поступающим на этот вход. В конце последнего цикла выполняемой МП команды он проверяет состояние этого триггера и, если оно изменилось, запоминает состояние маскируемого прерывания (разрешено или запрещено), запрещает его и формирует цикл подтверждения немаскируемого прерывания. В цикле подтверждения прерывания МП не реагирует на сигнал \overline{WAIT} . В след за циклом подтверждения прерывания обязательно следуют два цикла записи данных в память, в которые в вершину стека автоматически записываются старший и младший содержимого счетчика команд РС. Тем самым в стеке сохраняется адрес возврата, адрес команды, к выполнению которой МП приступит после окончания выполнения подпрограммы обработки прерывания. Автоматическое сохранение адреса возврата при работе с подпрограммами характерно для любого процесса.

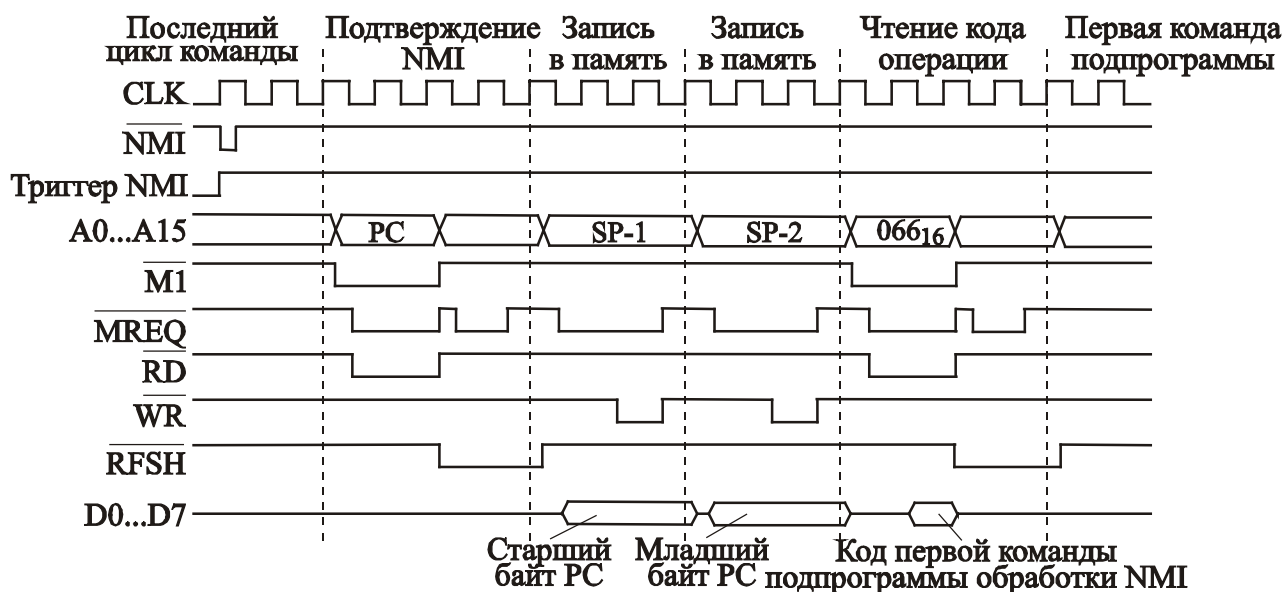


Рис. 25

Далее следует цикл чтения кода операции из ячейки памяти с адресом 066_{16} — код первой команды обработки прерывания.

Завершение подпрограммы обработки прерывания осуществляется по специальной команде возврата из прерывания, которая имеет мнемоническое обозначение RETN. После выполнения этой команды МП автоматически восстанавливает запомненное ранее состояние маскируемого прерывания, и считывает из вершины стека в PC два байта. МП будет интерпретировать содержимое PC как адрес возврата, однако это будет справедливо только в том случае, если при выполнении подпрограммы обработки прерывания количество занесенных в стек и извлеченных из него байт было одинаковым. В противном случае выполнение основной программы после возврата из подпрограммы будет некорректным.

В ходе выполнения подпрограммы обработки немаскируемого прерывания маскируемое запрещается, следовательно, немаскируемое прерывание имеет более высокий приоритет.

МП Z80 перейдет на подпрограмму обработки маскируемого прерывания в том случае, если это прерывание разрешено, а на входе \overline{INT} появляется сигнал низкого уровня. Т. к. МП анализирует состояние этой линии в конце последнего цикла выполняемой программы, то длительность сигнала запроса маскируемого прерывания должна быть не меньше времени выполнения самой продолжительной команды. Максимальная длительность этого сигнала не должна превышать времени выполнения подпрограммы обработки маскируемого прерывания. В противном случае один сигнал запроса прерывания вызовет неоднократное выполнение подпрограммы его обработки.

При переходе к выполнению подпрограммы обработки маскируемого прерывания МП формирует цикл подтверждения прерывания, рис. 26, в ходе которого из внешнего устройства считывается вектор прерывания. Сигналом, разрешающим внешнему устройству выставить на шину данных вектор прерывания, может выступать совокупность $\overline{M1}$ и \overline{IORQ} . Для того, чтобы внешнее устройство успело

выставить вектор прерывания, в цикле автоматически подтверждения формируются два такта ожидания T_w , количество которых может быть увеличено, исходя из анализа линии \overline{WAIT} .

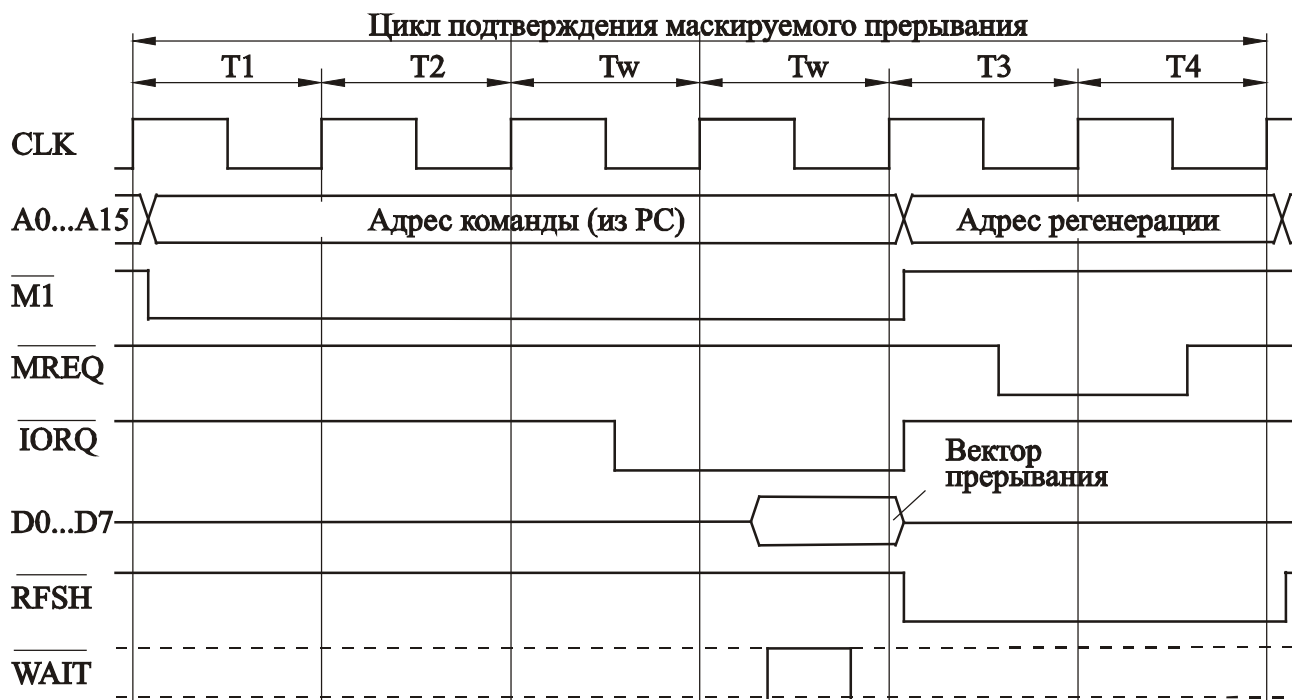


Рис. 26

За циклом подтверждения маскируемого прерывания МП автоматически формирует два цикла записи, сохраняя в стеке адрес возврата. Дальнейшие действия МП зависят от типа маскируемого прерывания.

В МП Z80 существуют маскируемые прерывания нулевого (IM 0), первого (IM 1) и второго (IM 2) типов. Тип прерывания задается программно.

Если разрешено маскируемого прерывания нулевого типа, то данные, принимаемые МП в цикле подтверждения, интерпретируются им как код первой команды обработки прерывания. В системе команд МП Z80 существуют однобайтные команды рестарта RST N, которые переводят МП к выполнению управляющей программы с заданного адреса. В табл. 6 показаны двоичные коды команд RST N и адреса, на которые они передают управления.

Таблица 6

Команда	Двоичный код команды	Адрес
RST 0	1 1 0 0 0 1 1 1	0_{16}
RST 1	1 1 0 0 1 1 1 1	8_{16}
RST 2	1 1 0 1 0 1 1 1	10_{16}
RST 3	1 1 0 1 1 1 1 1	18_{16}
RST 4	1 1 1 0 0 1 1 1	20_{16}
RST 5	1 1 1 0 1 1 1 1	28_{16}
RST 6	1 1 1 1 0 1 1 1	30_{16}
RST 7	1 1 1 1 1 1 1 1	38_{16}

Маскируемое прерывание первого типа практически аналогично немаскируемому прерыванию, но отличается от него вектором рестарта: 038_{16} вместо 066_{16} .

Маскируемые прерывания второго типа наиболее гибкие, т. к. с их помощью можно менять вектор прерывания как программно, так и аппаратно.

При этом виде прерываний вектор, полученный МП от внешнего устройства в цикле подтверждения прерывания, интерпретируется им как младший байт адреса. Старшим байтом адреса в этом случае будет являться содержимое регистра векторов прерываний I, информация в который заносится программистом. Сформировав такой 16-разрядный адрес, МП выполняет две операции чтения из памяти: из ячейки памяти с полученным адресом извлекается младший байт нового адреса, из следующей ячейки памяти — старший байт нового адреса. Новый адрес будет являться адресом перехода на собственно подпрограмму обработки маскируемого прерывания второго типа; в следующем цикле МП начнет выполнение этой подпрограммы.

Для иллюстрации принципов работы с прерыванием второго типа рассмотрим пример. Пусть в системе требуется реализовать 3 маскируемых прерывания, подпрограммы обработки которых начинаются с адресов 2358_{16} , $A546_{16}$, $BC31_{16}$. В некоторой области памяти, например, начиная с ячейки, расположенной по адресу $FD55_{16}$, разместим карту переходов к этим подпрограммам — начиная с указанной ячейки занесем в память адреса подпрограмм: 58_{16} в ячейку с адресом $FD55_{16}$, 23_{16} в ячейку с адресом $FD56_{16}$, 46_{16} в ячейку с адресом $FD57_{16}$ и т. д. Адреса подпрограмм заносятся в «обратном» порядке — сначала младший байт, затем старший. В регистр векторов прерываний I следует занести старший байт адреса первой ячейки карты переходов — число FD_{16} . Если в МП разрешены маскируемые прерывания второго типа и в цикле подтверждения он примет, например, вектор 46_{16} , то начнется выполнение второй подпрограммы обработки прерывания. Очевидно, что карта переходов и сами подпрограммы могут располагаться в любых областях памяти микропроцессорной системы.

В цикле подтверждения маскируемого прерывания МП автоматически запрещает это прерывание. В дальнейшем это прерывание должно быть разрешено программно. Команда разрешения прерывания обычно предшествует команде возврата из подпрограммы обработки маскируемого прерывания.

Возврат из подпрограммы осуществляется по специальной команде RETI, после выполнения которой МП извлекает из стека адрес возврата и возобновляет выполнение основной программы.

5 Микроконтроллеры

5.1 Общая характеристика

Как отмечалось выше, в настоящее время более целесообразно строить системы управления с использованием однокристальных микроЭВМ. Однокристальная микроЭВМ отличается от МП тем, что помимо центрального процессора содержит память и многочисленные периферийные устройства: аналого-цифровые пре-

образователи, последовательные и параллельные каналы передачи информации, таймеры реального времени, широтно-импульсные модуляторы (ШИМ), генераторы программируемых импульсов и т. д., рис. 27.

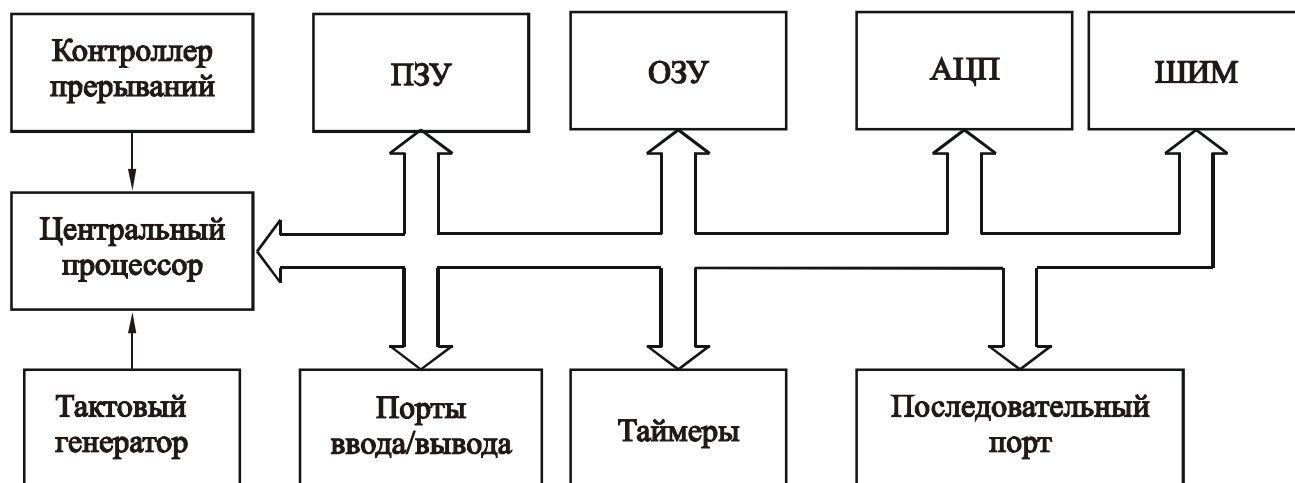


Рис. 27

По своей структуре и принципу функционирования однокристалльная микро-ЭВМ, в сущности, не отличается от микропроцессорной системы, но реализована в виде одной БИС. Подчеркивая функциональную завершенность и основное назначение однокристалльных микроЭВМ — использование в системах автоматического управления, встроенных в самые разные устройства, их называют микроконтроллерами (МК).

В настоящее время в мире выпускаются тысячи разновидностей МК. Они поставляются в корпусах с числом выводов от 8 до 356, работают при температуре от -55 до $+125^{\circ}\text{C}$, на частотах от 32 кГц до 200 МГц, способны функционировать при напряжении питания от 1,2 В, потребляя при этом ток, не превышающий единицы микроампер. Цена изделий также непрерывно снижается. Некоторые 8-разрядные МК стоят не дороже 50 центов, что сопоставимо со стоимостью одной ИС «жесткой логики». Все это привело к тому, что сегодня все труднее найти область человеческой деятельности, где бы МК не нашли применения.

Рассмотрим общие вопросы, связанные с архитектурой МК. Т. к. МК содержит внутри себя центральный процессор, то он имеет свою систему команд. В зависимости от числа используемых кодов операций системы команд подразделяют на две группы: CISC и RISC. Термин CISC означает сложную систему команд и является аббревиатурой английского определения Complex Instruction Set Computer. Термин RISC происходит от Reduced Instruction Set Computer и означает сокращенную систему команд. Несмотря на широкую распространенность этих понятий, следует отметить, что сами названия не отражают главного различия между этими системами команд. Основная идея RISC-архитектуры — это тщательный подбор таких комбинаций кодов операций, которые можно было бы выполнить за один период сигнала тактового генератора, т. е. за один такт. Основным выигрыш от такого подхода — существенное упрощение аппаратной реализации центрального процессора и возможность значительно повысить его производительность.

Первоначально реализовывать такой подход удавалось, лишь существенно сократив набор команд, отсюда и родилось название RISC. Например, система команд МК семейства Microchip PIC16 включает в себя всего 35 инструкций и может быть отнесена к типу RISC. Очевидно, что в общем случае одной команде CISC-архитектуры должны соответствовать несколько команд RISC-архитектуры. Однако выигрыш от повышения быстродействия в рамках RISC-архитектуры обычно перекрывает потери от менее эффективной системы команд, что приводит к более высокой эффективности RISC-систем. Например, самая быстрая команда МК 8051 выполняется за 12 тактов. Даже если для каждой инструкции потребуется выполнить три инструкции RISC-контроллера, то в итоге RISC-архитектура обеспечит четырехкратное увеличение производительности.

Кроме того при упрощении центрального процессора уменьшается число транзисторов, необходимых для его реализации, что ведет к снижению стоимости МК и потребляемой им мощности.

В настоящее время грань между этими двумя понятиями стирается. Например, МК семейства AVR фирмы Atmel имеют систему команд из 120 инструкций, что соответствует типу CISC. Однако большинство из них выполняется за один такт, что является признаком RISC-архитектуры. Поэтому принято считать, что основным признаком RISC-архитектуры является выполнение команд за один такт. Число команд само по себе значения уже не имеет.

Тактовый генератор вырабатывает импульсы для синхронизации работы всех узлов МК. Частоту их следования могут задавать внешние кварцевый резонатор или RC-цепь. В некоторых МК предусмотрен режим работы тактового генератора без применения внешних элементов.

Назначение ПЗУ — хранения программ, поэтому эту память называют памятью программ. До недавнего времени существовало две основных разновидности ПЗУ — масочные и программируемые.

В масочные ПЗУ информацию заносят в процессе изготовления МК с помощью технологических шаблонов — масок. Изменить ее после окончания производственного цикла невозможно. Такие ПЗУ используют лишь в случаях, когда качество программы не вызывает сомнений и существует массовая потребность в МК именно с этой программой. Достоинство масочных ПЗУ — самая низкая стоимость при массовом производстве.

В программируемые ПЗУ информацию записывают с помощью внешнего устройства, называемого программатором. МК с такими ПЗУ бывают двух типов: однократно и многократно программируемые (перепрограммируемые). Первые допускают только однократное программирование, после чего стереть информацию уже невозможно. МК с такой памятью используют в мелкосерийном производстве, когда применение масочных МК экономически не оправдано из-за затрат на разработку маски.

Перепрограммируемые МК подразделяются на оснащенные ПЗУ со стиранием ультрафиолетовым (УФ) излучением и МК с электрически перепрограммируемой памятью. МК, имеющие память, стираемую УФ излучением, выпускаются в корпусе с прозрачным окошком, через которое осуществляется стирание. Недостат-

ком таких МК — высокая стоимость и относительно небольшое число циклов записи/стирания, которое зависит от суммарной дозы облучения кристалла и обычно не превышает 15...20.

В настоящее время все более распространенной становится новая технология реализации ПЗУ — Flash-память. Ее главное достоинство в том, что она построена на принципе электрической перепрограммируемости, т. е. допускает многократное стирание и запись информации с помощью только электрических сигналов. Минимальное гарантированное число циклов записи/стирания обычно превышает несколько тысяч. Это существенно увеличивает жизненный цикл и повышает гибкость микроконтроллерных систем, т. к. позволяет вносить изменения в управляющую программу как на этапе разработки системы, так и в процессе его работы в реальном устройстве. В некоторых МК реализованная возможность перепрограммирования Flash-память программ без внешнего программатора.

Оперативное запоминающее устройство, реализованное в МК, обычно особенностей не имеет. Однако в ряде МК наряду с обычной энергозависимой памятью данных присутствует энергонезависимая память данных — разновидность Flash-памяти.

Большинство МК построено предполагает отдельное использование памяти программ и памяти данных, т. е. построено по гарвардской архитектуре.

Таймеры/счетчики — это построенные на основе счетчиков устройства, которые могут быть запрограммированы на выполнение целого ряда функций. Их можно использовать для точного формирования временных интервалов, подсчета импульсов на выводах МК, формирования последовательности импульсов, тактирования приемопередатчика последовательного канала связи. Таймеры/счетчики могут вырабатывать запросы прерываний. Поскольку основное применение МК находят в системах реального времени, таймеры/счетчики являются их обязательным элементом. В некоторых МК количество таймеров/счетчиков достигает 32.

Параллельные порты ввода/вывода также являются обязательным элементом любого МК. Обычно их используют для связи — датчиками и исполнительными механизмами.

Важная особенность параллельных портов МК — возможность программной настройки на выполнение нескольких функций. Например, в МК 8051 выводы портов P0 и P2 могут использоваться либо как обычные статические регистры ввода/вывода, либо в качестве шины адреса и данных для подключения внешних устройств, таких как дополнительная память программ, память данных, ППА. Порт P3 этого МК может использоваться как статический регистр ввода/вывода, либо выполнять специальные функции для работы последовательного канала, таймеров, контроллера прерываний и т. д. Возможность программной настройки позволяет с максимальной эффективностью задействовать в проектируемом устройстве все выводы МК.

Последовательный порт — важный канал информационного обмена МК с внешними устройствами. Такой канал занимает минимальное число выводов МК, обеспечивая связь на значительные расстояния с минимальными аппаратными затратами. В различных МК реализованы устройства, поддерживающие разнооб-

разные протоколы: RS-232C, что обеспечивает возможность организации связи МК с персональным компьютером, RS-485, I²C (двухпроводная двунаправленная шина), SPI (последовательный периферийный трехпроводный интерфейс), Bitbus (последовательная магистраль управления), CAN (межконтроллерный сетевой интерфейс), USB (универсальная последовательная шина) и некоторые другие. Практически для любого типа последовательного канала можно найти МК, имеющий в своем составе соответствующий последовательный порт.

Обычно, когда говорят о каком-либо МК, то всегда упоминают семейство, к которому он принадлежит. К одному семейству относят изделия, имеющие одинаковое ядро, под которым понимают совокупность таких понятий, как система команд, временные диаграммы работы ЦП, организация памяти программ и памяти данных, система прерываний и базовый набор периферийных устройств. Отличия между различными представителями одного семейства заключаются, в основном, в составе периферийных устройств и объеме памяти программ или данных. Поскольку диапазон задач, решаемых МК, чрезвычайно широк, их производители стараются выпустить столько модификаций, чтобы удовлетворить самые разнообразные запросы потребителей. В некоторых семействах число достигает ста и более.

Наиболее важная особенность семейства — программная совместимость на уровне двоичного кода всех входящих него МК. Это позволяет разработчикам систем заменять одни МК семейства другими без серьезной переделки имеющегося программного обеспечения. Естественно, чем большее число разновидностей входит в семейство, тем больше шансов выбрать оптимальный вариант, и тем привлекательнее это семейство для разработчика. Вопрос правильного выбора семейства МК для новой разработки является стратегическим, так как проблема переноса программного обеспечения между изделиями разных семейств чрезвычайно сложна и даже использование языков высокого уровня не всегда позволяет решить ее без больших потерь.

5.2 Микроконтроллер 8051 фирмы Intel

5.2.1 Краткое описание

В 1976 году фирма Intel выпустила первый МК — 8048. Помимо центрального процессора в него входила память программ 1 кбайт), память данных (64 Байта), 8-разрядный таймер/счетчик, 27 линий ввода/вывода. Отечественная промышленность выпускала аналог данного МК — прибор К1816ВЕ48. В настоящее время такие МК не выпускаются.

Развитием МК 8048 является более совершенный МК 8051. В составе этого МК имеется четыре 8-разрядных порта ввода/вывода, внутреннее ОЗУ 128 Байт, внутреннее ПЗУ 4 кбайта, два 16-разрядных таймера/счетчика, универсальный асинхронный последовательный приемопередатчик (UART), поддерживающий протокол стандарта RS-232C. Память данных и память программ в МК 8051 полностью разделены, т. е. он построен по гарвардской архитектуре. МК способен

обращаться к 64 кбайтам внешнего ОЗУ и 64 кбайтам внешнего ПЗУ. Максимальная частота тактовых импульсов — 12 МГц.

Данный прибор послужил основой целого семейства МК, выпускаемых фирмой Intel. Кроме того, ряд иностранных фирм выпускает полные аналоги этого МК, а также МК, ядро которых совместимо с 8051, но набор входящих в них устройств существенно превосходит имеющиеся в МК «родного» семейства.

Отечественная промышленность выпускает полный аналог МК 8051 — прибор К1816ВЕ51.

В связи с наличием МК, являющихся аналогами 8051, но имеющих различные способы реализации ПЗУ (с УФ стиранием, Flash и т. д.), будем рассматривать некоторый абстрактный прибор, который назовем МК51. По своим параметрам он соответствует 8051, но тип ПЗУ уточнять не будем.

На рис. 28 показано УГО прибора МК51. На данном рисунке приняты следующие обозначения:

0V — потенциал общего провода («земли»);

Ucc — основное напряжение литания +5 В;

BQ0, BQ1 — выводы для подключения внешнего кварцевого резонатора;

RST — вход общего сброса;

$\overline{\text{PME}}$ — разрешение внешней памяти программ; сигнал на этой линии выдается только при обращении МК51 к внешнему ПЗУ;

ALE — строб адреса внешней памяти;

DEMA — отключение внутренней программной памяти; низкий уровень сигнала на этом входе заставляет МК51 выполнять программу только из внешнего ПЗУ;

P0 — 8-разрядный двунаправленный порт ввода/вывода информации; при работе с внешними ОЗУ и ПЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных;

P1 — 8-разрядный квазидвунаправленный порт ввода/вывода;

P2 — 8-разрядный квазидвунаправленный порт; выводы этого порта дополнительно используются для выдачи старшего байта адреса при работе с внешними ОЗУ и ПЗУ;

P3 — 8-разрядный квазидвунаправленный порт; выводы этого порта могут выполнять ряд альтернативных функций: P3.0 — входная линия UART, обозначается RxD, P3.1 — выходная линия UART (TxD), P3.2 — вход запроса нулевого внешнего прерывания ($\overline{\text{INT0}}$), P3.3 — вход запроса первого внешнего прерывания ($\overline{\text{INT1}}$), P3.4 — счетный вход нулевого таймера/счетчика (T0), P3.5 — счетный вход первого таймера/счетчика (T1), P3.6 — запись ($\overline{\text{WR}}$), P3.7 — чтение ($\overline{\text{RD}}$).

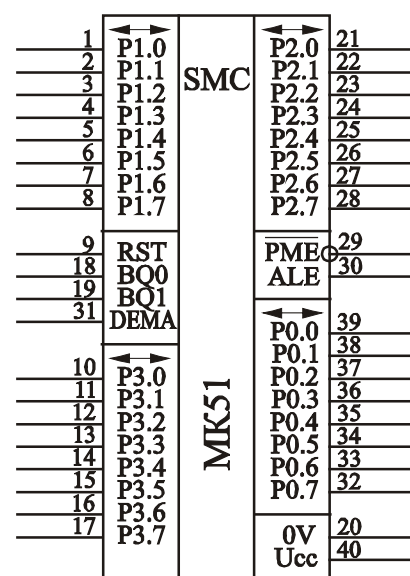


Рис. 28

Практически все команды МК51 выполняются за один или два машинных цикла. Исключение составляют команды умножения MUL A, B и деления DIV A, B, продолжительность выполнения которых составляет четыре машинных цикла. МК51 является типичным представителем CISC-контроллеров.

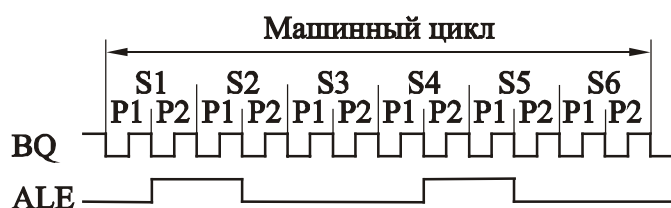


Рис. 29

Машинный цикл МК51 имеет фиксированную длительность и содержит шесть состояний (тактов) S1...S6. Каждое состояние состоит из двух временных интервалов, называемых фазами P1 и P2, рис. 29. Длительность фазы равна периоду следования сигнала синхронизации BQ.

Сигнал BQ вырабатывается либо встроенным тактовым генератором МК51 при подключении к ее выводам BQ1 и BQ2 кварцевого резонатора или LC-цепочки, либо внешним источником тактовых сигналов.

В каждом машинном цикле МК51 дважды формирует сигнал ALE, который используется при работе с внешними устройствами.

5.2.2 Блок таймеров/счетчиков

Таймеры/счетчики (Т/С) предназначены для подсчета внешних событий, для получения программно управляемых временных задержек и выполнения времязадающих функций микроконтроллера. МК51 имеет два таймера/счетчика — Т/С0 и Т/С1. В состав каждого Т/С входит 16-разрядный регистр ТС. Эти регистры выполняют функцию хранения содержимого счета. Каждый из них состоит из пары 8-разрядных регистров, соответственно ТН0, ТЛ0 и ТН1, ТЛ1. Регистры ТН0, ТН1 содержат старшие, а регистры ТЛ0, ТЛ1 — младшие 8 разрядов регистров ТС0 и ТС1. Каждый из 8-разрядных регистров имеет свой адрес и может быть использован как РОН, если соответствующий Т/С не используются.

Начальное значение величины счета заносится в регистры ТС программно. В процессе счета содержимое регистров инкрементируется. Признаком окончания счета является переполнение регистра ТС, т. е. переход из состояния «все единицы» в состояние «все нули». При окончании счета Т/С устанавливает флаг TF, что вызывает соответствующее прерывание, если оно разрешено. Флаги TF0 и TF1 представляют собой соответственно 5-й и 7-й биты 8-разрядного регистра управления TCON. Эти флаги могут программно считываться, устанавливаться или сбрасываться. Программная установка флага эквивалентна его аппаратной установке при переполнении Т/С, т. е. вызывает соответствующее прерывание. При вызове подпрограммы обработки прерывания флаг сбрасывается.

Все регистры ТН0, ТЛ0, ТН1, ТЛ1 доступны по чтению, что позволяет контролировать достижение требуемой величины счета программно.

Каждый Т/С может быть настроен на работу с качестве таймера или в качестве счетчика внешних событий. При работе в качестве таймера содержимое регистра ТС инкрементируется в каждом машинном цикле, т. е. Т/С является счетчиком

машинных циклов. Поскольку машинный цикл состоит из 12 периодов сигнала синхронизации частотой f_{BQ} , то частота счета в данном случае равна $\frac{f_{BQ}}{12}$.

При работе Т/С в качестве счетчика внешних событий содержимое регистра ТС инкрементируется в ответ на переход из «1» в «0» сигнала на счетном входе (вывод Т0 для Т/С0 и вывод Т1 для Т/С1). Счетные входы аппаратно проверяются в фазе S5P2 каждого машинного цикла. Когда проверки показывают высокий уровень на счетном входе в одном машинном цикле и низкий уровень в другом машинном цикле, регистр ТС инкрементируется. Новое значение заносится в регистр ТС в фазе S3P1 машинного цикла, непосредственно следующего за тем, в котором был обнаружен переход из «1» в «0» на счетном входе. Т. к. для распознавания такого перехода требуется 2 машинных цикла (24 периода сигнала синхронизации), то максимальная частота счета Т/С в режиме счетчика равна $\frac{f_{BQ}}{24}$.

Чтобы уровень сигнала на счетном входе был гарантировано зафиксирован, он должен оставаться неизменным в течение как минимум одного машинного цикла.

Настройку режимов работы Т/С осуществляют записью информации в специальный регистр TMOD. Обозначение разрядов регистра TMOD и их назначение приведено в табл. 7.

Таблица 7

Номер бита	Обозначение	Назначение	Режим работы							
			1	Разрешает управление						
7	GATE1	Разрешение управления Т/С1 от вывода $\overline{INT1}$	0	Запрещает управление						
			1	Разрешает управление						
6	С/Т1	Выбор варианта использования Т/С1	1	Т/С1 в качестве таймера						
			0	Т/С1 в качестве счетчика						
5	M1.1	Выбор режима работа Т/С1	0	0-й	0	1-й	1	2-й	1	3-й
4	M0.1		0		1		0		1	
3	GATE0	Разрешение управления Т/С0 от вывода $\overline{INT0}$	1	Разрешает управление						
			0	Запрещает управление						
2	С/Т0	Выбор варианта использования Т/С0	1	Т/С0 в качестве таймера						
			0	Т/С0 в качестве счетчика						
1	M1.0	Выбор режима работа Т/С0	0	0-й	0	1-й	1	2-й	1	3-й
0	M0.0		0		1		0		1	

Работа Т/С разрешается установкой 6 и 4 разрядов регистра управления TCON. Обозначение разрядов регистра и их назначение приведены в табл. 8. Младшие 4 разряда регистра TCON более подробно рассмотрим при анализе системы прерываний МК51.

Следует отметить, что для регистра TCON возможны побитовые операции, т. е. отдельные его биты могут быть установлены или сброшены командами SETB и RES. К разрядам регистра TMOD побитовые операции неприменимы.

Таблица 8

Номер бита	Обозначение	Назначение	Режим работы	
			1	0
7	TF1	Флаг переполнения T/C1	1	Счетчик переполнен
			0	Счетчик не переполнен
6	TR1	Включение T/C1	1	T/C1 включен
			0	T/C1 выключен
5	TF0	Флаг переполнения T/C0	1	Счетчик переполнен
			0	Счетчик не переполнен
4	TR0	Включение T/C0	1	T/C0 включен
			0	T/C0 выключен
3	IE1	Флаг запроса первого внешнего прерывания	1	Формирование запроса
			0	Отсутствие запроса
2	IT1	Вид сигнала запроса первого внешнего прерывания (на $\overline{INT1}$)	1	Прерывание по фронту
			0	Прерывание по уровню
1	IE0	Флаг запроса нулевого внешнего прерывания	1	Формирование запроса
			0	Отсутствие запроса
0	IT0	Вид сигнала запроса нулевого внешнего прерывания (на $\overline{INT0}$)	1	Прерывание по фронту
			0	Прерывание по уровню

Как видно из табл. 7, режим работы каждого T/C определяется значением битов M0, M1 регистра TMOD. T/C0 и T/C1 имеют четыре режима работы. Режимы работы 0, 1, 2 одинаковы для обоих T/C; T/C0 и T/C1 в этих режимах полностью независимы друг от друга. Работа T/C0 и T/C1 в режиме 3 различна, установка режима 3 в T/C0 влияет на режимы работы T/C1.

T/C в режиме 0 представляет собой устройство на основе 13-разрядного регистра, состоящего из 8 разрядов регистра TH и 5 младших разрядов регистра TL. Работу T/C в этом режиме иллюстрирует рис. 30.

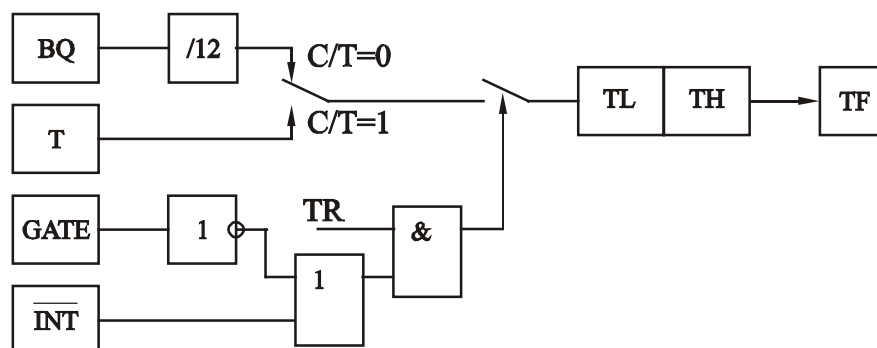


Рис. 30

На линии BQ присутствует частота f_{BQ} . Деление этой частоты на 12 определяет частоту счета при работе T/C в качестве таймера. Бит C/T регистра TMOD оп-

ределяет работу Т/С или в качестве таймера (С/Т=0), или в качестве счетчика (С/Т=1). Счет начинается при установке бита TR регистра TCON. При необходимости управления счетом извне производится установка бита GATE регистра TMOD. Тогда при TR=1 счет будет разрешен, если на входе INT присутствует сигнал высокого уровня, и будет запрещен, если сигнал на этом входе будет иметь низкий уровень. Сброс бита TR выключает Т/С независимо от состояния других битов. При переполнении Т/С устанавливается соответствующий флаг TF.

Режим 1 практически аналогичен режиму 0. Отличие состоит в том, что в этом режиме Т/С функционирует на основе 16-разрядного регистра, т. е. задействованы все разряды регистра TL.

Во 2-м режиме Т/С представляет собой устройство на основе 8-разрядного регистра TL. При каждом переполнении TL кроме установки в регистре TCON флага TF автоматически происходит загрузка в регистр TL содержимого регистра TH. Регистр TH загружается программно, загрузка в TL из TH не влияет на содержимое последнего. Работу Т/С в режиме 2 иллюстрирует рис. 31. Назначение битов управления TR, GATE, С/Т такое же, как в режимах 0 и 1.

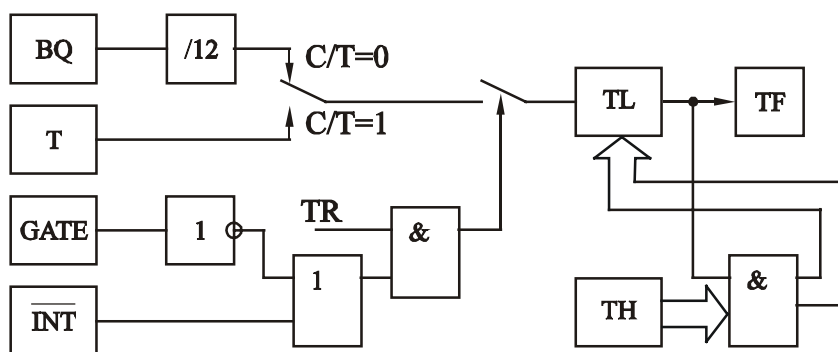


Рис. 31

Режим 3 различен для Т/С0 и Т/С1. Т/С1 в режиме 3 заблокирован и просто сохраняет значение кода в регистре TC1. Т/С0 в этом режиме представляет собой два независимых устройства на основе 8-разрядных регистров TL0 и TH0. Устройство на основе регистра TL0 может работать в режиме таймера и в режиме счетчика. За ним сохраняются все биты управления Т/С0, оно реагирует на воздействия по входам T0, INT0. При переполнении TL0 устанавливается флаг TF0.

Устройство на основе регистра TH0 может работать только в режиме таймера. Оно использует бит включения TR1, при переполнении TH0 выставляет флаг TF1. Других битов управления устройством на основе TH0 в этом режиме не имеет. Работу Т/С в режиме 3 иллюстрирует рис. 32.

Перевод Т/С0 в режим 3 лишает Т/С1 бита включения TR1. Поэтому, если Т/С0 находится в режиме 3, то Т/С1 в режимах 0, 1, 2 при GATE=0 всегда включен и при переполнении в режимах 0 и 1 Т/С1 обнуляется, а в режиме 2 перезагружается не устанавливая флаг. Следовательно, в таком режиме Т/С1 можно использовать в любых приложениях, не требующих формирования запроса прерывания, т. е. при программном отслеживании содержимого регистра TC1. Остановить счет

T/C1 можно сигналом $\overline{INT1}=0$ при $GATE1=1$, т. к. управление от входов $\overline{INT1}$, T1, биты управления C/T1, GATE1 для T/C1 не зависят от режима T/C0.

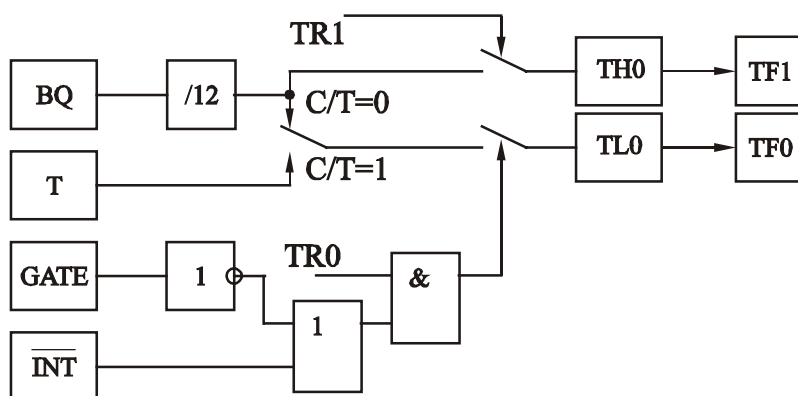


Рис. 32

В любом режиме работы выключение T/C (при помощи бита TR или сигнала $\overline{INT}=0$) не искажает код, находящийся в регистре TC. T/C можно выключить, через произвольное время вновь включить и счет начнется с той величины, которая была в регистре TC на момент выключения.

Новая загрузка T/C сразу же означает новую величину счета, а старая теряется. Если загрузка произведена при включенном T/C, счет продолжится с новой величины. Очередность загрузки регистров TL0, TH0, TL1, TH1 — произвольная.

Во всех режимах, кроме режима 2, после переполнения T/C счет продолжается с нулевой величины.

5.2.3 Организация портов ввода/вывода

Порты P0, P1, P2, P3 предназначены для обеспечения обмена информацией между МК51 и внешними устройствами. В совокупности порты образуют 32 линии ввода/вывода. Каждый из портов содержит 8-разрядный регистр, называемый фиксатором-защелкой. Вывод информации в порт осуществляется записью в фиксатор-защелку. К фиксатору-защелке применимы побитные операции, позволяющие выборочно установить или сбросить любой бит.

Помимо работы в качестве обычных линий ввода/вывода разряды портов P0...P3 могут выполнять ряд дополнительных функций. Рассмотрим эти функции.

При работе с внешней памятью программ и внешней памятью данных через порт P0 выводится младший байт адреса A0...A7. По этим же линиям МК принимает данные при работе с внешней памятью.

Если при обращении к внешней памяти МК формирует 16-разрядный адрес, то его старший байт A8...A15 выводится по линиям порта P2.

Каждая линия порта P3 имеет индивидуальную альтернативную функцию: P3.0 — RxD, вход последовательного порта, предназначен для ввода последовательных данных в приемник последовательного порта; P3.1 — TxD, выход последовательного порта, предназначен для вывода последовательных данных из передатчика последовательного порта; P3.2 — $\overline{INT0}$, используется как вход запроса нулевого внешнего прерывания; P3.3 — $\overline{INT1}$, используется как вход запроса

первого внешнего прерывания; P3.4 — T0, используется как вход счетчика внешних событий T/C0; P3.5 — T1, используется как вход счетчика внешних событий T/C1; P3.6 — \overline{WR} , строб записи во внешнюю память данных, выходной сигнал, сопровождающий вывод данных через порт P0 при обращении к внешней памяти данных; P3.7 — \overline{RD} , строб чтения из внешней памяти данных, выходной сигнал, сопровождающий ввод данных через порт P0 при обращении к внешней памяти данных. Альтернативная функция любой линии порта P3 выполняется только в случае, когда соответствующий этой линии бит фиксатора-защелки установлен.

На рис. 33 показана функциональная схема одного разряда порта P0. Разряд фиксатора-защелки на этом рисунке показан в виде D-триггера, управляемого внутренним сигналом «Запись в защелку», который вырабатывается при записи данных в порт. Прямой выход триггера может быть подключен на внутреннюю шину МК через буферный элемент B1 сигналом «Чтение защелки», что обеспечивает возможность программного чтения содержимого фиксатора. Значение сигнала непосредственно на линии порта может быть программно считано на внутреннюю шину МК через буферный элемент B2, управляемый внутренним сигналом «Чтение линии». Часть команд МК при чтении порта активизирует сигнал «Чтение защелки», т. е. считывает информацию из фиксатора, другая часть команд считывает информацию непосредственно с линий порта и активизирует сигнал «Чтение линии».

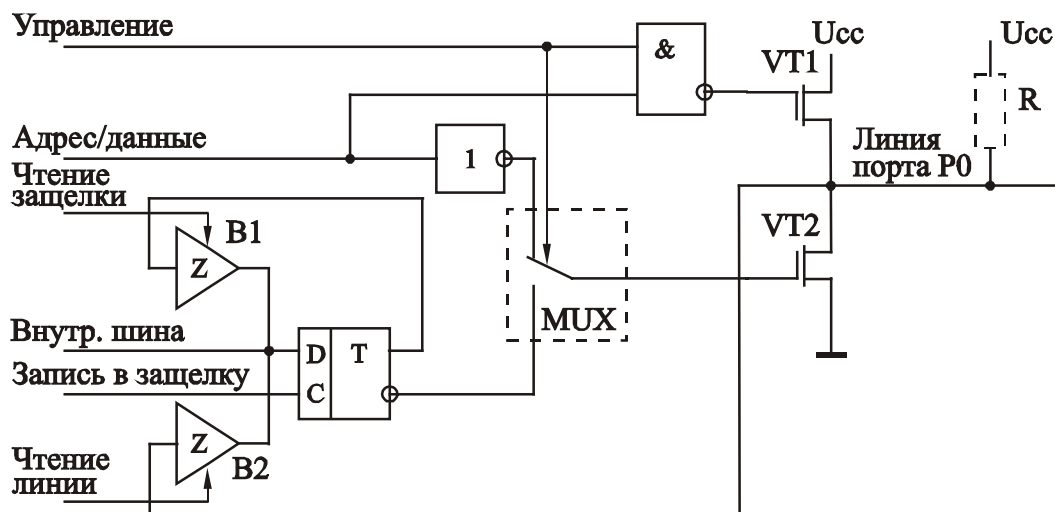


Рис. 33

Как показано на рис. 33, выходные каскады порта P0, образованные транзисторами VT1 и VT2, через мультиплексор MUX могут подключаться либо к выходу защелок, либо к внутренней шине адреса/данных. Последнее обеспечивается сигналом высокого уровня на линии «Управление» и используется при обращении к внешней памяти. Во время обращения к внешней памяти в защелки порта P0 всегда автоматически записываются «1» во все разряды, но это никак не сказывается на состоянии линии порта. Однако, информация в защелках теряется.

Если в процессе работы с внешней памятью по линии «Адрес/данные» передается сигнал высокого уровня, то транзистор VT1 откроется, а VT2 закроется.

Следовательно, на линии данного разряда будет присутствовать высокий уровень сигнала, близкий к напряжению питания. Соответственно, при выводе по линии «Адрес/данные» сигнала низкого уровня будет открыт транзистор VT2, а VT1 закроется. Это приведет к появлению на линии разряда сигнала низкого уровня.

В случае работы порта на ввод/вывод транзистор VT1 всегда будет закрыт сигналом «Управление» низкого уровня. Выходной каскад в этом случае будет представлять собой каскад с открытым стоком, транзистор VT2 через мультиплексор будет подключен к выходу защелки. Запись в защелку «0» приведет к открытию этого транзистора и появлению на линии разряда сигнала низкого уровня. При записи в защелку «1» транзистор VT2 закроется и выходной каскад будет переведен в высокоимпедансное состояние. Уровень сигнала на линии порта определяется в этом случае внешним устройством, присоединенным к этой линии и может быть прочитан МК при использовании команд, активизирующих сигнал «Чтение линии». Поэтому говорят, что запись в защелку «1» переводит соответствующую линию порта в режим ввода, хотя никаких переключений в схеме разряда не происходит.

Запись в защелку «1» может рассматриваться как попытка вывести на линию сигнал высокого уровня. Однако для линий порта внутренними средствами получить сигналы высокого уровня невозможно, т. к. запись «1» в разряд фиксатора-защелки переводит соответствующую линию в высокоимпедансное состояние. Для формирования выходного сигнала высокого уровня необходимо предусмотреть внешние «подтягивающие» резисторы, один из которых показан пунктиром на рис. 33. Таким образом, в зависимости от решаемой задачи, запись «1» в защелку может трактоваться как попытка сформировать на соответствующей линии выходной сигнал высокого уровня, либо как настройка разряда на ввод информации. Это справедливо для всех портов МК51.

Очевидно, что разряды порта могут использоваться на ввод или вывод независимо друг от друга, что качественно отличает порты МК51 от портов рассмотренного выше ППА КР580ВВ55А.

Порт P1, функциональная схема одного разряда которого показана на рис. 34, является наиболее простым из портов МК51, т. к. предназначен только для работы в качестве устройства ввода/вывода. Особенностью данного порта, равно как и портов P2 и P3, является наличие внутренних нагрузочных резисторов R, а также схемы ускоренного переключения разряда из состояния «0» в состояние «1».

При записи в защелку «0» закрывается транзистор VT1, а транзистор VT2 открывается. На линии разряда будет присутствовать низкий уровень сигнала.

При записи в защелку «1» транзистор VT2 сразу же закрывается. Из-за наличия в схеме управления транзистором VT1 элемента задержки DEL, обеспечивающего задержку длительностью в два периода сигнала BQ, в течение этого интервала времени транзистор VT1 будет открыт. В нагрузку, подключенную к разряду, будет отдаваться ток, в 100 раз больший тока, определяемого резистором R, что позволяет быстрее установить на линии порта сигнал высокого уровня. По истечении времени задержки транзистор VT1 закроется и выходной ток разряда

порта будет определяться резистором R . Таким образом, разряды порта $P1$ способны самостоятельно формировать сигналы высокого уровня.

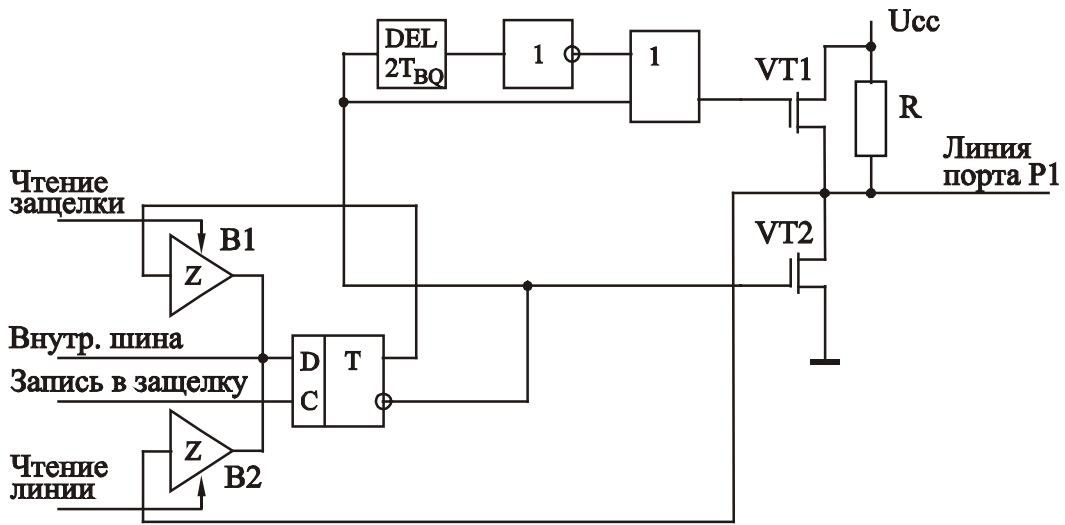


Рис. 34

Запись в защелку «1» может рассматриваться как перевод разряда в режим ввода. Однако, для элементов, подключенных к нему, разряд будет являться, источником тока, определяемого резистором R . Поэтому порты $P1$, $P2$ и $P3$, имеющие однотипные выходные каскады с нагрузочными резисторами, называют «квазидвунаправленными», в отличие от «истинно двунаправленного» порта $P0$, который в режиме ввода находится в высокоимпедансном состоянии. В большинстве случаев квазидвунаправленность портов несущественна.

Порт $P2$ практически не отличается по структуре выходного каскада от порта $P1$, но имеет мультиплексор, переключающий каскад на выдачу адреса, рис. 35.

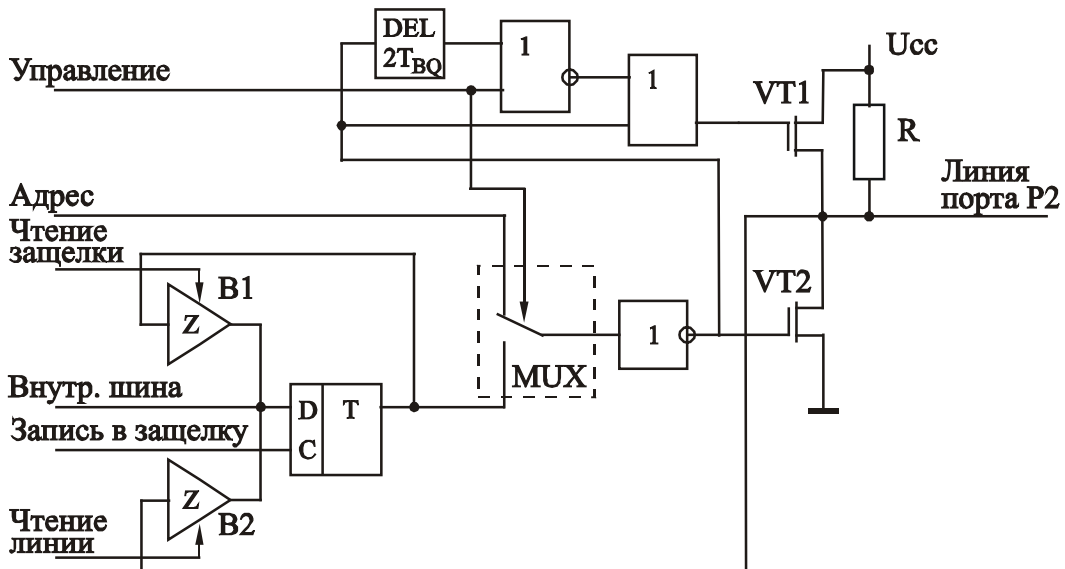


Рис. 35

Если через линию порта P2 выдается разряд адреса, содержащий «1», то выходное напряжение высокого уровня формируется мощным транзистором VT1, который в этом случае будет открыт в течение всего времени выдачи адреса.

Выдача адреса через порт P2 не влияет на содержимое его защелок. Если порт P2 не задействован на выдачу адреса, то на его выводах выставляется содержимое защелок.

Как отмечалось выше, разряды порта P3 способны выполнять альтернативные функции, поэтому его внутренняя структура отличается от других, рис. 36. Запись в разряд фиксатора-защелки «1» порта P3 приводит к тому, что выходной сигнал на соответствующей линии становится зависимым от внутреннего сигнала «Альтернативная функция выхода», а через буферный элемент B3 сигнал на линии может быть прочитан как альтернативная функция входа. Следовательно, запись в разряд защелки «1» может трактоваться как настройка соответствующего разряда на выполнение альтернативной функции. Однако, по аналогии с другими портами, это же действие позволяет сформировать на линии выходной сигнал высокого уровня и настроить линию порта на ввод.

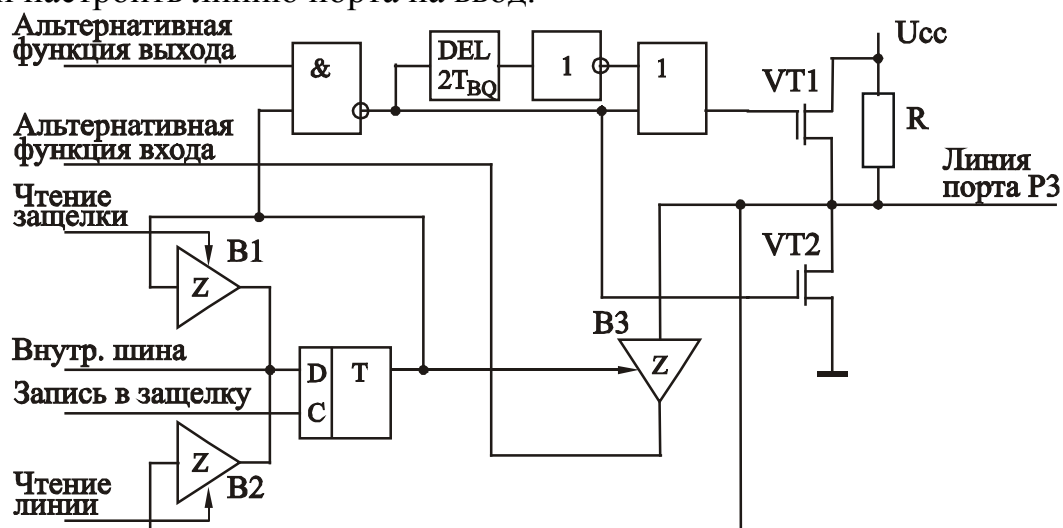


Рис. 36

При выполнении записи в любой порт новое значение заносится в защелку в фазе S6P2 последнего машинного цикла команды. Однако непосредственно на выводах порта новое содержимое защелки поступает в фазе S1P1 следующего машинного цикла.

Порты имеют идентичные характеристики. Уровни напряжений на линиях портов совместимы с ТТЛ. Величины втекающих и вытекающих токов различны для изделий разных производителей, поэтому должны рассматриваться применительно к конкретному прибору. С характеристиками некоторых МК семейства 8051 можно познакомиться в [2].

Данные, записанные в порты P1, P2 и P3, статически фиксируются и не изменяются до перезаписи. Если не используется работа с внешней памятью, то данные в порту P0 также фиксируются до перезаписи. Сигнал RST устанавливает все порты на прием входной информации, т. е. записывает с защелки «1».

Как было отмечено выше, часть команд МК51 при чтении порта считывает информацию из защелки, другая часть команд считывает информацию непосредственно с линий порта. Если в команде порт служит операндом-источником, то информация считывается непосредственно с линий порта. Например при выполнении команды ADD A,P1 содержимое аккумулятора складывается с информацией на выводах порта P1 и результат заносится в аккумулятор.

Во всех случаях, когда операндом и регистром назначения результата является порт или бит порта, команды считывают информацию с выходов защелок, а не с внешних выводов порта. Команды, считывающие информацию с выходов защелок, реализуют так называемый режим «чтение-модификация-запись», заключающийся в том, что команда считывает состояние защелки, при необходимости модифицирует полученное значение и записывает результат обратно в защелку. Ниже приведены команды, работающие в этом режиме.

ANL — логическое И, например, ANL P1,A.

ORL — логическое ИЛИ, например, ORL P2,A.

XRL — логическое ИСКЛЮЧАЮЩЕЕ ИЛИ, например, XRL P3,A.

JBC — переход, если бит равен «1» и очистка бита, например, JBC P1.1,LABEL.

CPL — инверсия бита, например, CPL P3.0.

INC — инкремент, например, INC P2.

DEC — декремент, например, DEC P2.

DJNZ — декремент и переход, если не ноль, например, DJNZ P3,LABEL.

MOV PX.Y, C — пересылка бита переноса в бит Y порта X.

CLR PX.Y — очистка бита Y порта X.

SETB PX.Y — установка бита Y порта X.

Не очевидно, что последние три команды в приведенном списке работают в режиме «чтение—модификация—запись», однако, это так. Указанные команды считывают с порта весь байт целиком, модифицируют адресуемый бит, после чего записывают полученный новый байт обратно в фиксатор-защелку порта.

Чтение информации с выходов защелок, а не с внешних линий порта позволяет исключить возможную в ряде случаев неправильную интерпретацию уровня сигнала на выводе порта. К примеру, вывод порта P1.0 может использоваться для управления ключевым каскадом на биполярном транзисторе, рис. 37. Т. к. данный порт имеет нагрузочные резисторы, которые ограничивают выходной ток, то в цепи базы дополнительный резистор можно не устанавливать. В этом случае при записи в защелку «1» транзистор откроется, но физически на линии данного разряда будет присутствовать сигнал низкого уровня, определяемый напряжением на эмиттерном переходе транзистора. Чтение же выходного сигнала из защелки покажет истинное логическое значение сигнала на выводе порта.

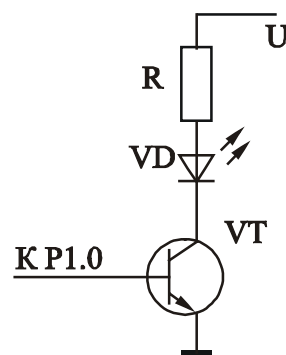


Рис. 37

5.2.4 Организация памяти в МК51

Все микроконтроллеры семейства МК51 имеют несколько адресных пространств, функционально и логически разделенных за счет разницы в механизмах адресации и сигналах управления записью и чтением: память программ, внутренняя память данных, внешняя память данных.

Память программ имеет 16-разрядное адресное пространство, ее элементы адресуются с использованием счетчика команд РС или команд перехода. Память программ доступна только по чтению, т. к. МК не имеет команд и управляющих сигналов, предназначенных для записи в эту память. Память программ имеет байтовую организацию и общий объем до 64 Кбайт. Ряд МК семейства имеют внутреннюю память программ различной емкости. В базовом варианте 8051 емкость внутренней памяти программ 4 Кбайт.

С точки зрения программиста имеется только одно пространство памяти программ объемом 64 Кбайт. Тот факт, что в ряде МК он образуется комбинацией массивов, находящихся внутри МК и вне его, для программиста неощутим, т. к. центральный процессор МК автоматически выбирает байт из соответствующего массива в соответствии с его адресом.

Сигналом, стробирующим выборку и ввод байта из внешней памяти программ в МК является сигнал \overline{PME} . Для МК, содержащих внутреннюю память программ, сигнал \overline{PME} формируется только в том случае, если адрес в счетчике команд превосходит максимальный адрес внутренней памяти программ. При выборке из внутренней памяти программ данный сигнал не формируется. Для МК, не имеющих внутренней памяти программ, сигнал \overline{PME} формируется при любом обращении к памяти программ.

Микроконтроллеры семейства МК51 имеют вывод DEMA, с помощью которого можно запретить работу внутренней памяти программ. Если подать на вывод DEMA сигнал низкого уровня, то внутренняя память программ отключается и все обращения происходят к внешней памяти программ с формированием сигнала \overline{PME} . Если на вывод DEMA подан сигнал высокого уровня, то работают и внутренняя и внешняя память программ. Для микроконтроллеров, не имеющих внутренней памяти программ, для нормальной работы всегда необходимо подавать на вывод DEMA сигнал низкого уровня.

Если центральный процессор осуществляет доступ к внешней памяти программ, сигнал \overline{PME} вырабатывается дважды во время каждого машинного цикла (исключение составляет команда MOVX) независимо от того, необходим или нет выбираемый байт для текущей команды. При выборке из внешней памяти программ всегда используется 16-разрядный адрес, младший байт которого выдается через порт P0, а старший — через порт P2. Байт из внешней памяти программ вводится в микроконтроллер через порт P0, который в этом случае используется как шина адреса/данных в режиме временного мультиплексирования.

На рис. 38 показана упрощенная временная диаграмма обращения МК к внешней памяти программ, а на рис. 39 — типовая схема подключения внешней памяти программ к МК51.

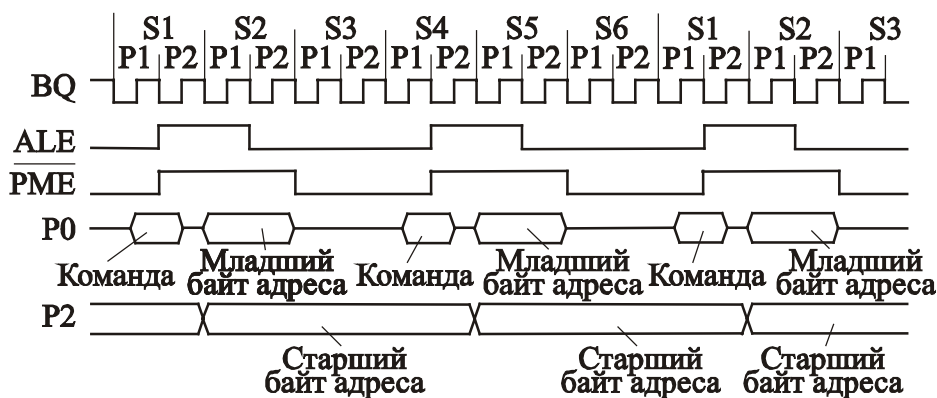


Рис. 38

Т. к. младший байт адреса присутствует на линиях порта P0 ограниченное время, для корректного обращения к ИС ПЗУ его необходимо сохранить в каком-либо внешнем устройстве. В качестве устройства временного хранения младшего байта адреса обычно используется 8-разрядный регистр. Сигнал ALE переходит из высокого уровня в низкий в момент, когда на линиях порта P0 гарантировано присутствует младший байт адреса используется для управления. Поэтому для записи этого байта в регистр используют задний фронт сигнала ALE, регистр должен иметь прямой статический или динамический вход записи.

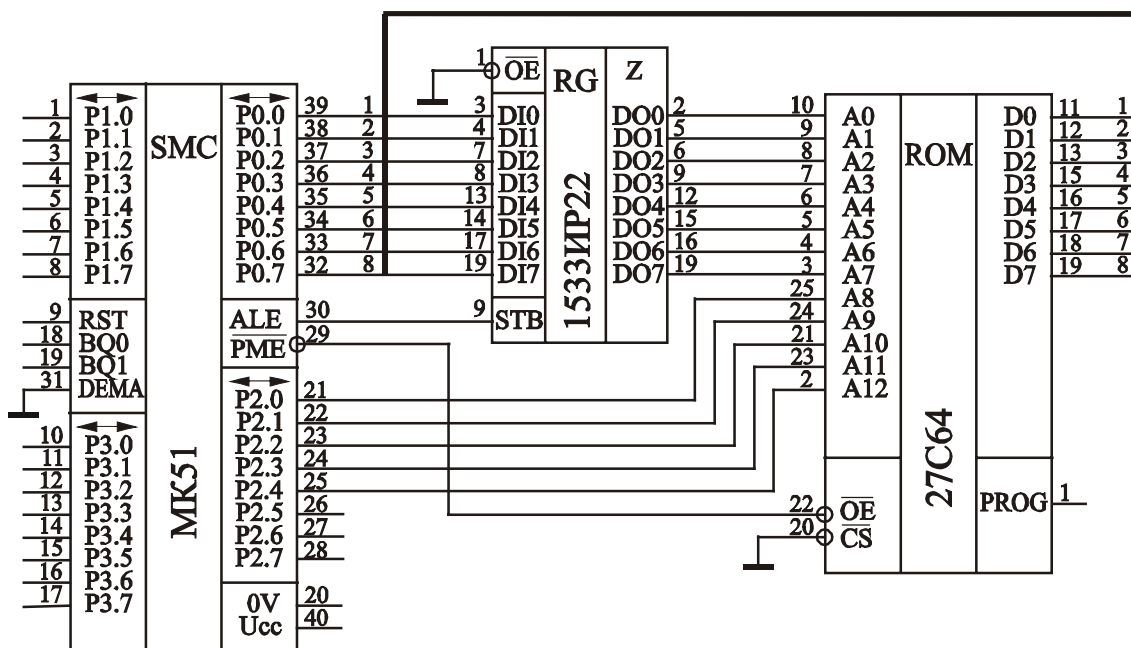


Рис. 39

Сигнал \overline{PME} выполняет для ИС ПЗУ роль сигнала разрешения выхода \overline{OE} . Если в схеме присутствует одна ИС ПЗУ, она может быть постоянна выбрана низким уровнем сигнал на входе \overline{CS} . При задействовании нескольких ИС ПЗУ следует производить их дешифрацию, как это было рассмотрено для микропроцессорной системы на МП Z80. Полную 16-разрядную шину адреса составляют выходные линии регистра и линии порта P2. В качестве шины адреса выступает

порт P0. Как будет показано ниже, эти шины используются при обращении к внешней памяти данных и другим устройствам.

Входные линии показанного на схеме регистра через внутренние резисторы соединены с цепью питания этой ИС. Поэтому нет необходимости дополнительно подключать к линиям порта P0 «подтягивающие» резисторы. Однако, не все совместимые с ТТЛ серии ИС имеют такие внутренние резисторы.

Внутренняя память данных МК51 состоит из двух областей: 128 байт оперативной памяти (ОЗУ) с адресами $0-7F_{16}$ и области регистров специальных функций, занимающей адреса $80_{16}-FF_{16}$. Физически внутреннее ОЗУ данных и область регистров специальных функций являются отдельными устройствами.

Все ячейки внутреннего ОЗУ могут адресоваться с использованием прямой и косвенной адресации. Младшие 32 байта внутреннего ОЗУ данных сгруппированы в 4 банка по 8 регистров в каждом. Команды программы могут обращаться к регистрам, используя их имена R0...R7. Переключение между банками регистров осуществляется программно при помощи битов RS0 и RS1 специального регистра состояния PSW, табл. 9. Т. к. остальные разряды этого регистра используются редко, то они не рассматриваются. Познакомиться с их назначением можно в [2]. К разрядам регистра PSW применимы побитовые операции.

Таблица 9

Номер бита	Обозначение	Назначение	Режим работы							
			0	0-й	0	1-й	1	2-й	1	3-й
4	RS1	Указатель банка рабочих регистров	0		0		1		1	
3	RS0		0		1		0		1	

Следующие после банков регистров внутреннего ОЗУ 16 байт (адреса $20_{16}...2F_{16}$) образуют область ячеек, к которым возможна побитовая адресация. МК51 содержит значительное количество команд, позволяющих работать с отдельными битами, используя при этом прямую адресацию. Составляющие рассматриваемую область внутреннего ОЗУ 128 битов имеют собственные адреса $00_{16}...7F_{16}$ и предназначены для работы с такими инструкциями. Установка или сброс этих битов позволяет пользователю хранить информацию, например, о состоянии 128 устройств.

Обращение к внутреннему ОЗУ данных всегда осуществляется с использованием 8-разрядного адреса. При включении питания содержимое ОЗУ будет иметь случайное значение.

Внешняя память данных формируется дополнительными ИС ОЗУ, подключаемыми к МК и может иметь емкость до 64 Кбайт. Пространства внутренней и внешней памяти данных не пересекаются, т. к. доступ к ним осуществляется с помощью разных команд. Для работы с внешней памятью данных существуют специальные команды MOVX, которые не влияют на внутреннюю память данных МК. Таким образом, в системе могут одновременно присутствовать внутренняя память данных с адресами $00_{16}...FF_{16}$ и внешняя память данных с адресами $00_{16}...FFFF_{16}$. Обращение к ячейкам внешней памяти данных осуществляется

только с использованием косвенной адресации по содержимому регистров R0 и R1 активного банка регистров (команды типа MOV @Ri) или по содержимому регистра специальных функций DPTR (команды типа MOV @DPTR). Команды MOVX @Ri, A и MOVX A, @Ri формируют 8-разрядный адрес, выдаваемый через порт P0. Команды MOVX @DPTR, A и MOVX @A, DPTR формируют 16-разрядный адрес, младший байт которого выдается через порт P0, а старший — через порт P2.

Байт адреса, выдаваемый через порт P0, должен быть зафиксирован во внешнем регистре по спаду сигнала ALE, т. к. в дальнейшем линии порта P0 используются как шина данных, через которую байт данных принимается из внешнего ОЗУ при чтении или выдается в него при записи. При выполнении операции чтения из внешнего ОЗУ МК вырабатывает управляющий сигнал \overline{RD} , при записи — сигнал \overline{WR} . На рис. 40 показана возможная схема подключения внешней памяти данных к МК51.

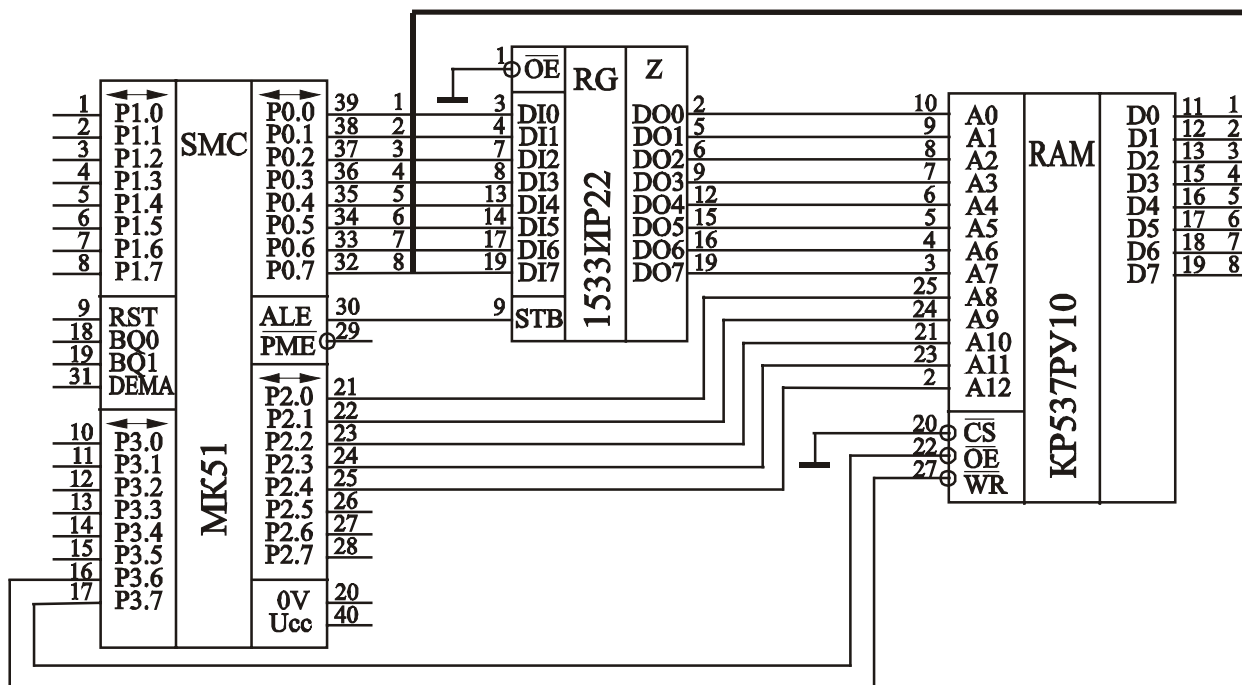


Рис. 40

По аналогии с показанным на рис. 40 подключением ИС ОЗУ к МК51 может быть подключено любое периферийное устройство, предназначенное для использования в микропроцессорных системах. Отдельного адресного пространства периферийных устройств МК51 не имеет.

5.2.5 Контроллер на основе МК51

На рис. 41 представлена схема системы, построенной на основе МК51. В системе используется внешняя память программ объемом 64кбайта. Внутренняя память программ отключена. Внешнее ОЗУ имеет объем 8 кбайт. Помимо внешнего ОЗУ в системе задействовано периферийное устройство — программируемый интегральный таймер КР580ВИ53. Применение данной ИС совместно с МК51 следует рассматривать как учебный пример, т. к. МК51 самостоятельно может вы-

полнять большинство функций интегрального таймера. На схеме не показаны элементы, задающие тактовые сигналы для МК51, а также элементы, обеспечивающие работу интегрального таймера.

Деление адресного пространства внешней памяти данных на области ОЗУ и периферийных устройств осуществляется при помощи инвертора. ОЗУ располагается по адресам от нулевого до $1FFF_{16}$. Адреса счетчиков интегрального таймера следующие: нулевого — $**1*****00_2$, первого — $**1*****01_2$, второго — $**1*****11_2$, регистра CW — $**1*****11_2$.

Система сброса МК51 в простейшем случае состоит из показанной на схеме RC-цепи. Номиналы элементов системы сброса рассчитываются из условия, что импульс сброса должен иметь длительность, как минимум 4 мс. Типовые значения при $f_{BQ} = 12$ МГц: $R=8,2$ кОм, $C=10$ мк.

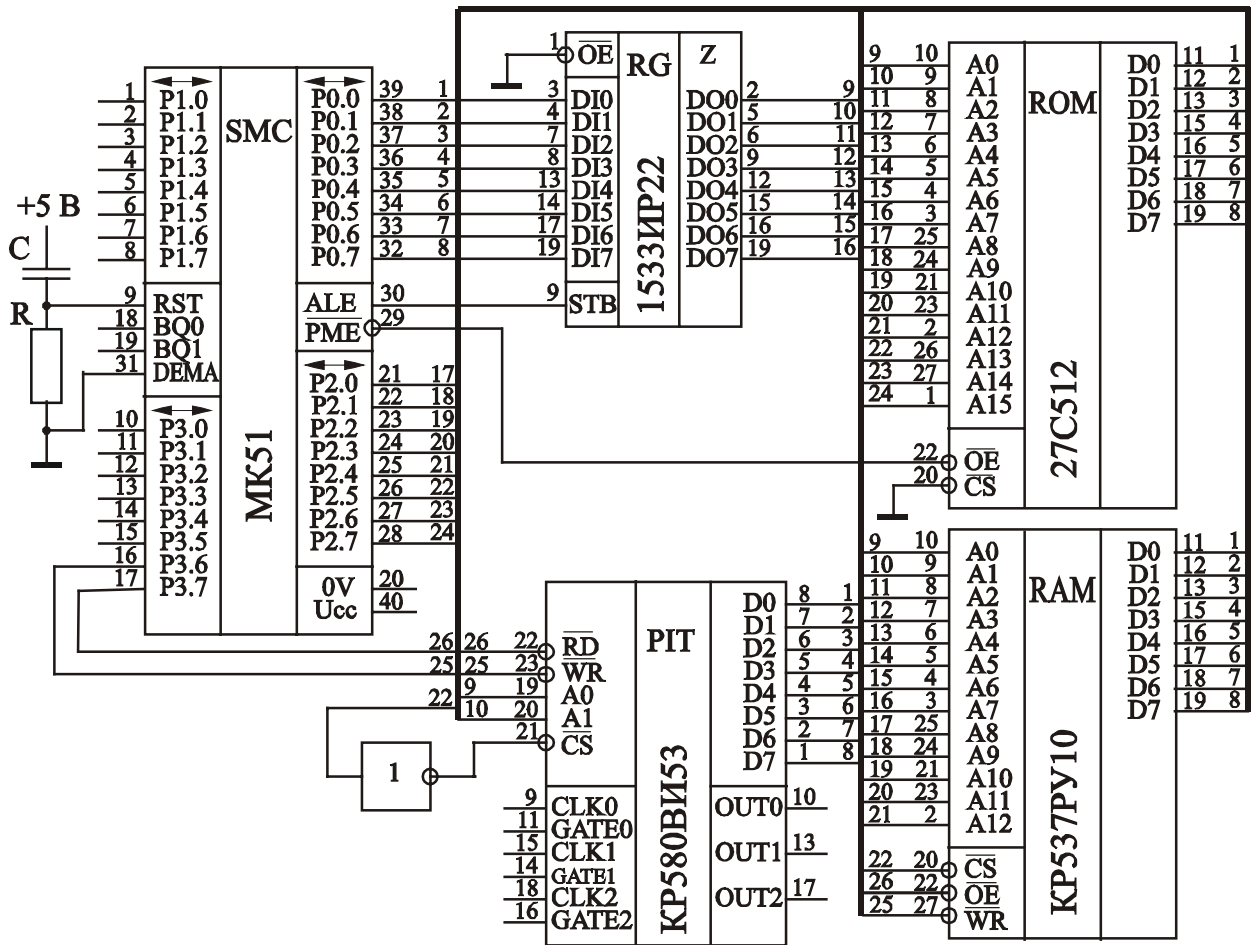


Рис. 41

5.2.6 Система прерываний в МК51

В МК51 имеет развитую систему прерываний с двумя уровнями приоритетов. В качестве источников прерываний могут выступать как внешние, так и внутренние (переполнение Т/С, завершение последовательного обмена) события.

МК имеет возможность обработки двух внешних прерываний — нулевого ($\overline{INT0}$) и первого ($\overline{INT1}$). Внешние прерывания запрашиваются по линиям P3.2 и P3.3. Каждое из этих прерываний может быть активизировано либо низким уровнем, либо задним фронтом сигнала на соответствующей линии. Способ активизации определяется состоянием битов IT0 и IT1 регистра TCON, см. табл. 8. При поступлении запроса внешнего прерывания устанавливается соответствующий бит IE регистра TCON. Если запрошенное прерывание разрешено, то установка бита вызовет подпрограмму обработки запрошенного прерывания. Если прерывание активизируется фронтом, то в момент вызова этой подпрограммы бит IE автоматически сбрасывается. Чтобы внешний сигнал запроса прерывания, активизируемого фронтом, был распознан, необходимо, чтобы его длительность была не меньше 12 периодов T_{BQ} . Для прерывания, активизируемого уровнем, запрашивающий сигнал должен удерживаться до начала работы подпрограммы обработки этого прерывания, и сниматься до завершения подпрограммы. Бит IE в этом случае сбросится в момент снятия запрашивающего сигнала. Программная установка бита IE эквивалентна появлению внешнего запрашивающего прерывания сигнала и приведет к вызову соответствующей подпрограммы.

Внутренние прерывания от Т/С вызываются установкой флагов TF0 и TF1 регистра TCON. Как рассмотрено выше, данные флаги устанавливаются при переполнении счетчиков. Сброс флагов осуществляется автоматически при вызове подпрограмм обработки прерываний.

Прерывания от последовательного порта вызываются установкой флага прерывания приемника RI или флага прерывания передатчика TI в регистре SCON. Данные флаги сбрасываются только программным путем.

В табл. 10 приведены векторы прерываний от перечисленных источников. Как видно из таблицы, установка флагов RI или TI приводит к вызову одной и той же подпрограммы. Эта подпрограмма должна определять, установкой какого флага она была вызвана, с тем, чтобы далее обработать прерывание от приемника или от передатчика.

Таблица 10

Источник прерывания	Флаг	Вектор прерывания
Нулевое внешнее прерывание $\overline{INT0}$	IE0	03 ₁₆
Нулевой таймер/счетчик T/C0	TF0	0B ₁₆
Первое внешнее прерывание $\overline{INT1}$	IE1	13 ₁₆
Первый таймер/счетчик T/C1	TF1	1B ₁₆
Последовательный порт	TI, RI	23 ₁₆

Любое из имеющихся в МК51 прерываний может быть разрешено или запрещено программно. За это отвечают биты регистра разрешения прерываний IE, табл. 11. К регистру IE применимы побитовые операции.

Таблица 11

Номер бита	Обозначение	Назначение
7	EA	Управление всеми прерываниями. При EA=0 все прерывания запрещены
6	x	Резервный
5	x	Резервный
4	ES	Управление прерываниями от последовательного порта. При ES=1 прерывание разрешено.
3	ET1	Управление прерыванием от первого таймер/счетчика. При ET1=1 прерывание разрешено.
2	EX1	Управление первым внешним прерыванием. При EX1=1 прерывание разрешено.
1	ET0	Управление прерыванием от нулевого таймер/счетчика. При ET0=1 прерывание разрешено.
0	EX0	Управление нулевым внешним прерыванием. При EX0=1 прерывание разрешено.

Каждому из источников прерываний может быть индивидуально присвоен высокий или низкий уровень приоритета. Это осуществляется установкой (высокий уровень приоритета) или сбросом (низкий уровень приоритета) соответствующих разрядов регистра приоритетов IP, табл. 12. К регистру IP применимы побитовые операции.

Таблица 12

Номер бита	Обозначение	Назначение
7, 6, 5	x	Резервные
4	PS	Установка уровня приоритета для прерывания от последовательного порта.
3	PT1	Установка уровня приоритета для прерывания от первого таймер/счетчика.
2	PX1	Установка уровня приоритета для первого внешнего прерывания.
1	PT0	Установка уровня приоритета для прерывания от нулевого таймер/счетчика.
0	PX0	Установка уровня приоритета для нулевого внешнего прерывания.

Подпрограмма обработки прерывания с низким уровнем приоритета может быть прервана запросом прерывания с высоким уровнем приоритета. Если одновременно приняты запросы прерываний с разными уровнями приоритета, то сначала будет обслужен запрос с высоким уровнем. Если одновременно запрошены прерывания с одинаковым уровнем приоритета, то порядок их обработки определяется последовательностью опроса флагов: IE0→TF0→IE1→TF1→RI, TI. Эта последовательность задает второй уровень приоритетов прерываний в МК51.

5.2.7 Последовательный порт

Последовательный порт МК51 представлен в виде линий P3.0 (RxD) и P3.1 (TxD). Последовательный порт может использоваться в виде регистра сдвига для расширения ввода-вывода или в качестве универсального асинхронного приемопередатчика с фиксированной или переменной скоростью последовательного обмена и возможностью дуплексного включения. Программный доступ к регистрам приемника и передатчика осуществляется обращением к регистру специальных функций SBUF. При записи в SBUF байт загружается в регистр передатчика, а при чтении SBUF байт читается из регистра приемника. Прием и выдача байта данных начинаются с младшего разряда и заканчиваются старшим. Для управления работой последовательного порта используется регистр SCON, табл. 13, и 7-й разряд (SMOD) регистра PCON. К регистру SCON побитовые операции применимы, а к регистру PCON нет.

Последовательный порт может быть запрограммирован на один из четырех режимов путем программирования разрядов SM0 и SM1 регистра SCON.

Таблица 13

Номер бита	Обозначение	Назначение	Режим работы								
			0	0-й	0	1-й	1	2-й	1	3-й	
7	SM0	Выбор режима работы последовательного порта	0	0	0	1	1	1	1	3-й	
6	SM1		0	1	0	0	1	1	1	3-й	
5	SM2	Разрешение многопроцессорной работы. В режиме 0 обеспечить SM2=0. Если SM2=1, то в режиме 1 флаг RI не устанавливается, если не принят стоп-бит, равный 1; в режимах 2 и 3 флаг RI не устанавливается, если девятый принятый бит данных равен 0.									
4	REN	Разрешение приема последоват. данных	1	Прием разрешен							
			0	Прием запрещен							
3	TB8	Девятый бит передаваемых данных в режимах 2 и 3.									
2	RB8	Девятый бит принятых данных в режимах 2 и 3.									
1	TI	Флаг прерывания передатчика	Устанавливается аппаратно в конце времени выдачи 8 бита в режиме 0 или в начале стоп-бита в других режимах.								
0	RI	Флаг прерывания приемника	Устанавливается аппаратно в конце времени приема 8 бита в режиме 0 или через половину интервала стоп-бита в режимах 1, 2, 3 при SM2=0.								

В режиме 0 последовательный порт работает как восьмиразрядный сдвиговый регистр. При этом 8 бит информации в последовательном коде принимаются и передаются через двунаправленный вывод RxD. На выводе TxD формируется

сигнал синхронизации, рис. 42. Байт передается или принимается начиная с младшего бита.

Передача начинается любой командой, которая выполняет операцию записи в регистр SBUF. При выполнении такой команды в фазе S6P2 вырабатывается внутренний импульс «Запись в SBUF», рис. 42,а, по которому предназначенный к передаче байт записывается в регистр сдвига передатчика и запускается блок управления передачей. Внутренняя система тактирования МК организована так, что между сигналом «Запись в SBUF» и началом передачи проходит один полный машинный цикл, после чего вырабатывается внутренний сигнал «Посылка», разрешающий выдачу содержимого регистра сдвига передатчика на выход RxD и импульсов синхронизации сдвига на выход TxD. Сигнал синхронизации имеет низкий уровень в состояниях S3, S4 и S5 каждого машинного цикла и высокий уровень в состояниях S6, S1 и S2. В фазе S6P2 каждого машинного цикла, в котором сигнал «Посылка» активен, формируется внутренний импульс «Сдвиг», по которому содержимое регистра сдвига передатчика сдвигается на одну позицию и на выходе RxD выставляется очередной бит передаваемой посылки. Всего формируется восемь импульсов «Сдвиг», после чего блок управления передачей снимает сигнал «Посылка» и устанавливает флаг прерывания передатчика TI.

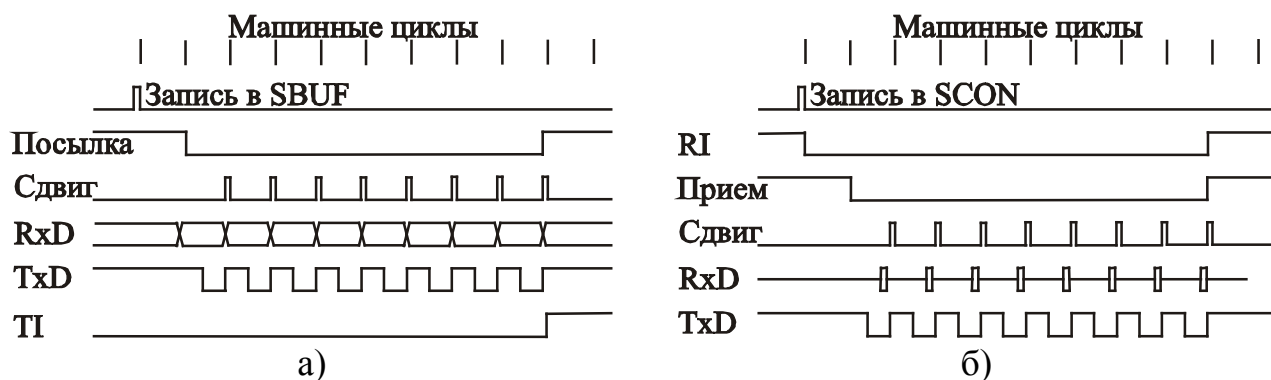


Рис. 42

Прием начинается при одновременном выполнении двух условий: $REN=1$ и $RI=0$. Условия выполняются записью необходимой информации в регистр SCON, при этом формируется соответствующий внутренний сигнал. В фазе S6P2 следующего машинного цикла блок управления приемом вырабатывает внутренний сигнал «Прием», разрешающий выдачу импульсов синхронизации на линию TxD. Биты принимаемой посылки, начиная с младшего, через вход RxD поступают в регистр сдвига приемника. Состояние входа RxD опрашивается в фазе S5P2. В фазе S6P2 каждого машинного цикла, в котором сигнал «Прием» активен, формируется внутренний импульс «Сдвиг» и содержимое регистра сдвига приемника сдвигается на одну позицию. Значение, которое при этом записывается в его старший разряд, является значением сигнала на входе RxD, полученным в фазе S5P2 этого же машинного цикла. Всего формируется восемь импульсов «Сдвиг», после чего блок управления приемом загружает содержимое регистра сдвига приемника в регистр SBUF. В фазе S1P1 10-го машинного цикла после записи в SCON, сбросившей RI, сигнал «Прием» сбрасывается, а флаг RI устанавливается.

Частота следования импульсов синхронизации, определяющая скорость приема/передачи, в режиме 0 постоянна и составляет $f_{BQ}/12$.

В режиме 1 прием/передача данных осуществляется в формате 8-разрядного универсального асинхронного приемопередатчика. Через TxD передаются, а через RxD принимаются 10 битов: старт-бит (активный уровень — низкий), 8 битов данных и стоп-бит (активный уровень — высокий). При приеме стоп-бит заносится в бит RB8 регистра SCON. Скорость приема/передачи определяется частотой переполнения T/C1 f_{TC1} , и составляет $f_{TC1}/16$ при SMOD=1 или $f_{TC1}/32$ при SMOD=0.

На рис. 43 показаны временные диаграммы, иллюстрирующие работу последовательного порта в режиме 1. Передача инициируется любой командой, использующей SBUF в качестве регистра назначения. Вырабатываемый при этом внутренний импульс «Запись в SBUF», рис. 43,а, загружает предназначенный к передаче байт в младшие 8 разрядов регистра сдвига передатчика и разрешает работу блока управления передачей. В 9-й разряд регистра сдвига заносится «1» (стоп-бит). Собственно передача начинается в начале машинного цикла, следующего за ближайшим после импульса «Запись в SBUF» импульсом «Синхр. Tx» (синхронизация передачи). Частота следования импульсов «Синхр. Tx» составляет $f_{TC1}/16$ или $f_{TC1}/32$ в зависимости от состояния бита SMOD. Период этого сигнала определяет время, в течение которого выдаваемый бит присутствует на выходе TxD.

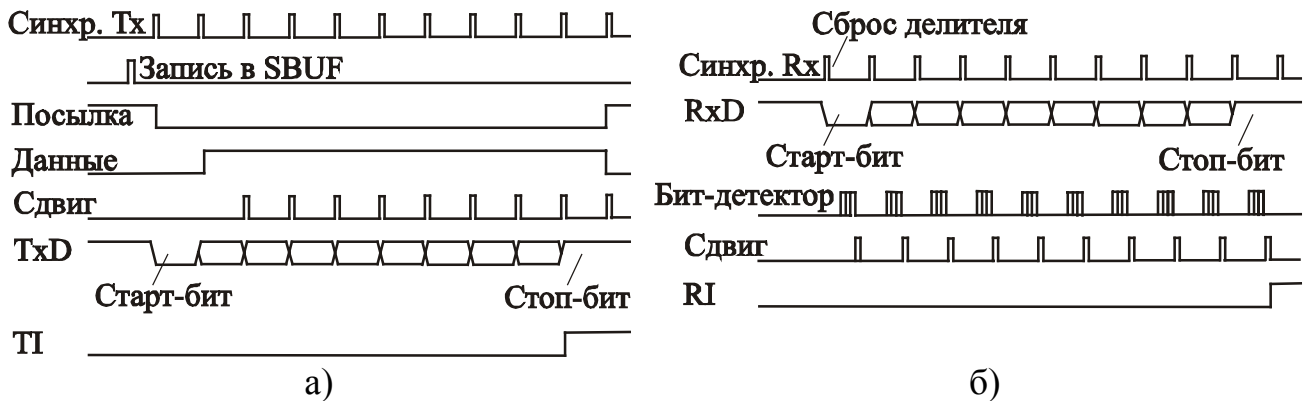


Рис. 43

Передача начинается установкой активного уровня внутреннего сигнала «Посылка», появление которого вызывает выдачу на выход TxD старт-бита. Следующий импульс «Синхр. Tx» активизирует сигнал «Данные», который разрешает выдачу содержимого регистра сдвига передатчика на выход TxD. При появлении сигнала «Данные» старт-бит на выходе TxD сменяется младшим битом регистра сдвига передатчика. Последующие импульсы «Синхр. Tx» формируют импульсы «Сдвиг», по которым содержимое регистра передатчика сдвигается на один разряд и на выходе TxD появляются новые биты. Всего формируется 9 импульсов «Сдвиг», в результате чего на выход TxD выдаются 8 битов данных и стоп-бит. По окончании выдачи всех битов посылки устанавливается флаг прерывания передатчика TI и снимаются сигналы «Посылка» и «Данные».

Прием начинается при обнаружении перехода сигнала на входе RxD из «1» в «0» при REN=1. Для отслеживания такого перехода вход RxD аппаратно опрашивается с частотой $f_{TC1}/16$ ($f_{TC1}/32$). Когда переход сигнала обнаружен, немедленно сбрасывается счетчик-делитель на 16 (32), в результате чего происходит совмещение импульсов «Синхр. Rx» с границами смены битов принимаемой посылки на входе RxD. Время, в течение которого бит принимаемой посылки присутствует на входе RxD, МК делит на 16 фаз. В фазах 7, 8 и 9 специальное устройство МК, бит-детектор, считывает с входа RxD три значения принимаемого бита, по мажоритарному принципу «2 из 3-х» выбирает из них одно и подает его на вход регистра сдвига приемника. Блок управления приемом при этом формирует внутренний импульс «Сдвиг», в результате чего содержимое регистра сдвига приемника сдвигается на один разряд и в регистр заносится принятый бит. Всего формируется 10 импульсов «Сдвиг». Т. к. регистр сдвига приемника в режиме 1 является 9-разрядным, то после 10-го импульса «Сдвиг» в регистре приемника находятся биты данных D0...D7 и стоп-бит. Если в момент генерации последнего (10-го) импульса «Сдвиг» при сброшенном флаге RI SM2=0 или стоп-бит равен «1», то данные из регистра сдвига приемника перегружаются в SBUF, стоп-бит из регистра сдвига приемника загружается в разряд RB8 регистра SCON и устанавливается флаг прерывания приемника RI. В противном случае принятая посылка безвозвратно теряется, а флаг RI не устанавливается.

Если мажоритарный отбор при приеме первого бита посылки (старт-бита) показывает ненулевое значение бита, все устройства блока приема сбрасываются и начинается отслеживание следующего перехода сигнала из «1» в «0» на входе RxD. Таким образом обеспечивается защита от некорректных старт-битов.

Режимы 2 и 3 — это режимы 9-разрядного универсального асинхронного приемопередатчика с постоянной и переменной скоростью обмена. В этих режимах через выводы TxD и RxD соответственно передаются и принимаются 11 битов в следующей последовательности: старт-бит, 9 битов данных, стоп-бит. Девятый бит данных при передаче определяется содержимым разряда TB8 регистра SCON. При приеме 9-й бит данных заносится в бит RB8 регистра SCON.

Скорость (частота) приема/передачи в режиме 2 программно настраивается на одну из двух возможных величин: $f_{BQ}/32$ или $f_{BQ}/64$. В режиме 3 скорость (частота) приема/передачи определяется частотой переполнения T/C1. Различие в скорости (частоте) приема/передачи является единственным отличием между режимом 2 и режимом 3. Во всем остальном эти два режима полностью идентичны.

На рис. 44 показаны временные диаграммы, иллюстрирующие работу последовательного порта в режимах 2 и 3. Как видно из рисунков, работа последовательного порта в этих режимах практически не отличается от работы в режиме 1. Однако, при передаче в режимах 2 и 3 на выход TxD выдается 9 битов данных: D0...D7 и TB8, а также стоп-бит, равный «1», см. рис. 44,а.

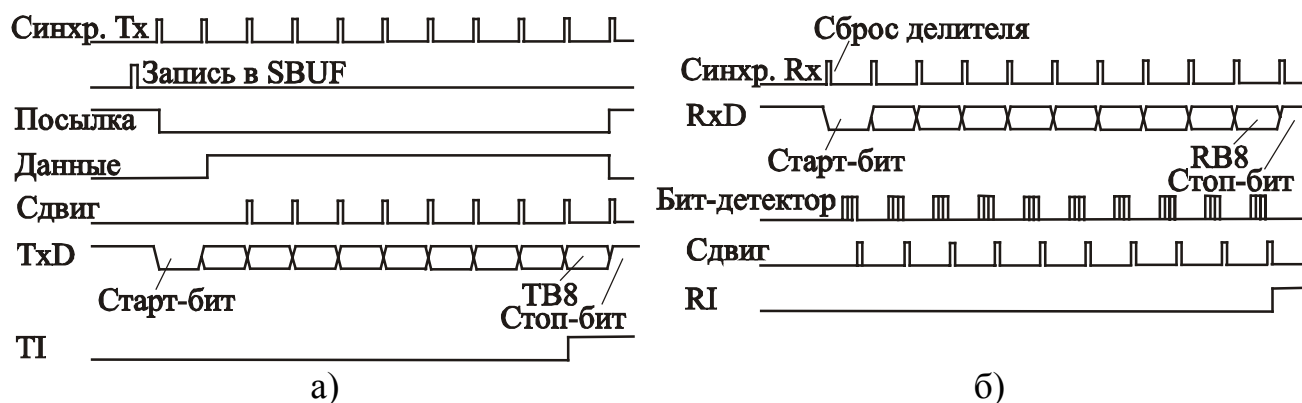


Рис. 44

При приеме в режимах 2 и 3 также осуществляется совмещение импульсов «Синхр. Rx» с границами смены битов принимаемой посылки на входе RxD и формирование внутренних импульсов «Сдвиг», см. рис. 44,а. Если в момент генерации 10-го импульса «Сдвиг» при сброшенном флаге RI SM2=0 или 9-й бит данных равен «1», то данные из регистра сдвига приемника перегружаются в SBUF, 9-й бит из регистра сдвига приемника загружается в разряд RB8 регистра SCON и устанавливается флаг прерывания приемника RI. В противном случае принятая посылка безвозвратно теряется, а флаг RI не устанавливается.

Тот факт, что при приеме в режимах 2 и 3 девятый бит данных может влиять на флаг RI, позволяет организовать работу МК51 в режиме многопроцессорной работы, подразумевающей обмен по последовательному каналу между несколькими МК. Настройка на этот режим осуществляется установкой бита SM2=1. При этом принятая МК посылка данных установит флаг RI только в том случае, если 9-й принятый бит равен «1». Работа в многопроцессорном режиме строится следующим образом. Изначально у всех МК бит SM2 установлен. Прежде чем передавать информацию, предназначенную для конкретного МК, ведущий МК передает посылку, которая содержит 8-разрядный идентификационный код и 9-й бит, равный «1». В результате все МК, принявшие посылку, вызывают подпрограммы обработки прерывания приемника. Эти подпрограммы анализируют принятый идентификационный код, тот МК, идентификационный код которого совпал с принятым, сбрасывает бит SM2. В следующей посылке ведущего МК 9-й бит должен быть равен «0», поэтому на нее отреагирует только МК, у которого SM=0.

Один из возможных вариантов использования последовательного порта — это организация обмена МК51 с персональным IBM-совместимым компьютером по интерфейсу RS-232. О том, как это осуществить, можно узнать из [8].

5.3 Обзор микроконтроллеров популярных серий

5.3.1 Микроконтроллеры фирмы Intel

Как отмечалось ранее, в настоящее время различными фирмами выпускается огромное количество разнообразных МК. Некоторые из них являются развитием рассмотренного МК51, другие имеют принципиально иную архитектуру. Приве-

дем краткое описание некоторых МК, ориентируясь, главным образом на новые по сравнению с 8051 внутренние устройства и функции.

Развитием МК51 являются МК группы 8052. МК 8052 имеет полную программную совместимость с 8051, т. е. все программы, написанные для 8051 будут без доработок исполняться МК 8052. МК группы 8052 имеют внутреннюю память программ объемом до 32 кбайт, внутреннюю память данных в 256 байт, дополнительный таймер/счетчик, развитую систему прерываний, реализующую 4 уровня приоритета. МК способны формировать на выводе P1.0 прямоугольные импульсы со скважностью 2 (меандр) и частотой до 4 МГц.

МК группы 8052 имеют два механизма защиты внутренней памяти программ от несанкционированного доступа: проверку зашифрованного содержимого памяти программ и биты защиты памяти программ. Первый механизм не обеспечивает абсолютную недоступность памяти программ, т. к. в этом случае она может читаться командами, выбранными из внешней памяти программ. Рассмотрим подробнее назначение битов защиты. В зависимости от модификации, МК этой группы содержат 1, 2 или 3 бита защиты. Каждый бит может быть запрограммирован или оставлен незапрограммированным. Если запрограммирован только 1-й бит защиты, то командам, выбранным из внешней памяти программ, запрещается доступ к внутренней памяти. Если запрограммированы 1-й и 2-й биты защиты, то в дополнение к запрету командам, выбранным из внешней памяти программ, обращаясь к внутренней памяти, запрещается проверка внутренней памяти программ внешними программирующими устройствами. Программирование всех битов защиты дополнительно полностью запрещает работу с внешней памятью программ.

Последовательный порт в МК группы 8052 имеет устройство автоматического распознавания адреса, что облегчает построение многопроцессорных систем. В специальном регистре SADDR каждого МК может находиться его адрес — идентификационный код. Когда ведущий МК передает адресную посылку, у которой 9-й бит равен «1», то остальные МК для анализа передаваемого идентификационного кода не формируют прерывание приемника. Устройство автоматического распознавания адреса в том МК, для которого принятый код совпал с хранимым в регистре SADDR, сбрасывает бит SM2 — этот МК готов к приему посылки, в которой 9-й бит равен «0». Детально ознакомиться с работой устройства автоматического распознавания адреса можно в [9].

Группа МК 8xC51Fх (80C51FA, 83C51FA и др.) фирмы Intel обладают широкими функциональными возможностями. Наиболее существенными отличиями по сравнению с МК семейства 8052 являются повышенная до 24 МГц тактовая частота и наличие массива программируемых счетчиков (РСА).

РСА состоит из 16-разрядного таймера/счетчика и пяти 16-разрядных модулей сравнения-защелки. Таймер/счетчик РСА служит в качестве базового для всех пяти модулей сравнения-защелки. Таймер/счетчик РСА может быть настроен на подсчет импульсов поступающих с делителей $f_{BQ}/4$ и $f_{BQ}/12$, сигналов переполнения T/C0 или внешних сигналов на линии P1.2. Информация о состоянии счетчика РСА хранится в регистрах CH и CL. Любой из модулей сравнения-защелки

независимо от остальных может быть запрограммирован на работу в следующих режимах: защелкивания по фронту и/или спаду импульса, программируемого таймера, высокоскоростного выхода, ШИМ. Каждый из модулей имеет в своем составе пару 8-разрядных регистров $SSAPnH$ и $SSAPnL$, где n — номер модуля.

В режиме защелкивания появление на входной линии $P1.3+n$ перепада, на отслеживание которого настроен модуль, приводит к запоминанию в регистровой паре $SSAPn$ текущего состояния счетчика PCA и вызову соответствующего прерывания. Тем самым обеспечивается возможность достаточно точного измерения периода следования сигналов, ширины импульса, его скважности, а также временных сдвигов между импульсами, приходящими по разным входным линиям. Точность измерения определяется главным образом временем, необходимым для запоминания информации в регистровой паре $SSAPn$. При тактовой частоте 12 МГц это время не превышает 10 мкс.

Режим программируемого таймера используется в случаях, когда некоторая подпрограмма должна выполняться с постоянным заданным интервалом. Это может потребоваться, например, при реализации часов реального времени.

В режиме высокоскоростного выхода происходит переключение сигнала на линии $P1.3+n$ с высокого уровня в низкий в момент совпадения информации в соответствующей регистровой паре $SSAPn$ с содержимым счетчика PCA . При этом может быть запрошено прерывание. Подпрограмма обработки этого прерывания должна изменить содержимое регистровой пары $SSAPn$. В противном случае новое изменение состояния линии $P1.3+n$ произойдет, когда счетчик PCA переполнится и начнет счет заново. Это может потребовать существенного времени — при тактовой частоте 16 МГц порядка 30 мс.

Выход ШИМ может использоваться для широтно-импульсного управления исполнительными устройствами систем автоматики, а также для преобразования цифровых сигналов в аналоговую форму. Выходной ШИМ-сигнал вырабатывается по следующим правилам: если содержимое регистра $SSAPnL$ меньше содержимого регистра CL , то на внешнем выводе $P1.3+n$ присутствует сигнал низкого уровня, в противном случае — высокого. Таким образом, значение, записанное в регистр $SSAPnL$, задает скважность формируемых импульсов, а сам модулятор является 8-разрядным. Т. к. при работе в этом режиме переполнение $SSAPnL$ приводит к записи в него информации из $SSAPnH$, то число, соответствующее требуемой скважности, следует хранить в $SSAPnH$. Содержимое $SSAPnH$ связано со скважностью следующим соотношением: $SSAPnH = 256 - \frac{256}{\text{скважность}}$, где результат округляется до целого.

Модуль 4 сравнения-защелки может использоваться в качестве сторожевого таймера (watchdog timer). Сторожевой таймер представляет собой устройство, которое вызывает сброс, если МК не пришлет ему сигнал подтверждения нормальной работы. Это устройство целесообразно задействовать в системах, длительно работающих в условиях мощных электрических помех. В этом режиме при совпадении содержимого счетчика PCA со значением, записанным в регистровой паре $SSAP4$, генерируется внутренний сигнал сброса. При задействованном стороже-

вом таймере предотвратить формирование этого сигнала можно периодическим изменением содержимого счетчика PCA и/или регистровой пары CCAP4 таким образом, чтобы до следующего изменения их значения не совпали. Очевидно, что в случае некорректной работы МК («зависания»), но при функционирующем сторожевом таймере в какой-то момент будет сформирован внутренний сигнал сброса, который восстановит работоспособность МК. Однако, информация, хранившаяся до «зависания» МК в его памяти данных скорее всего будет потеряна.

МК группы 8xC51GB имеют аппаратный сторожевой таймер WDT. Таймер всегда активен и работающем тактовом генераторе постоянно увеличивает содержимое своего 14-разрядного счетчика. Для предотвращения формирования внутреннего сигнала сброса пользовательская программа не реже одного раза за 16384 машинных цикла должна последовательно заносить в специальный регистр WDTRST два байта $1E_{16}$ и $A6_{16}$, что приведет к перезапуску сторожевого таймера. Не рекомендуется производить перезапуск при помощи подпрограммы обработки прерывания одного из таймеров/счетчиков, т. к. прерывания могут обрабатываться и при «зависшей» основной программе.

Другой особенностью МК группы 8xC51GB является наличие 8-канального 8-разрядного АЦП. АЦП может запускаться как внутренним, так и внешним сигналом. Результаты преобразования заносятся в 8 специальных регистров, из которых они могут быть считаны.

Все рассмотренные выше МК принято относить к семейству MCS-51. Развитием этого семейства стали МК семейств MCS-151/251. МК этих семейств (МК151/251) имеют полностью статическую структуру, что позволяет им работать при значениях тактовой частоты от 0 Гц. По составу внутренних устройств МК151/251 мало отличаются от МК группы 8xC51Fx семейства MCS-51. Они содержат четыре 8-разрядных параллельных порта ввода/вывода, последовательный порт с автоматическим распознаванием адреса, три таймера/счетчика и массив программируемых счетчиков, а также аппаратный сторожевой таймер.

ИС МК151/251 выпускаются в таких же корпусах, что и микросхемы MCS-51, и могут устанавливаться на платах вместо них без каких-либо изменений в схеме.

МК семейства MCS-151 (МК151) имеют такую же структуру и систему команд, что и МК семейства MCS-51. Объем внутренней памяти программ — до 16 Кбайт, внутренней памяти данных — 256 байт. Основное преимущество МК151 — существенное сокращение времени выполнения программ при той же тактовой частоте. Это достигается иной организацией процедуры обращения к внутренней памяти программ — машинный цикл с обращением к ней содержит два такта, при этом одновременно считываются два байта.

При обращении к внешней памяти программ МК151 может функционировать в режиме страничного обмена. В этом режиме во внешнем регистре запоминается старший байт адреса, а прием байта данных из ПЗУ осуществляется по линиям порта P2. В дальнейшем МК выдает старший байт адреса и защелкивает его в регистре, если он отличается от уже хранимого в регистре байта. В противном случае МК выдает только младший байт, тем самым увеличивая скорость обмена.

МК семейства MCS-251 (МК251) отличаются расширенной структурой внутреннего центрального процессора, расширенным адресным пространством и расширенной системой команд.

МК251 могут выполнять операции с данными, имеющими формат 8-, 16- или 32-разрядных двоичных чисел.

МК251 имеют адресное пространство объемом 128 кбайт. Дополнительный разряд адреса (A16) выдается через вывод P3.7, при этом чтение по всем адресам выполняется по сигналу PSEN. Внутренняя память программ может достигать 16 кбайт, внутренняя память данных — 1 кбайт.

МК251 разрабатывались как полностью программно совместимые с МК семейства MCS-51. Однако, существенные отличия в структуре потребовали расширения системы команд. МК семейства MCS-51 имеют 255 команд, код каждой из которых является 8-разрядным двоичным числом. Следовательно, для дополнительных команд может использоваться один оставшийся код — $A5_{16}$, и новые команды должны иметь двухбайтную кодировку. Очевидно, что такое построение команд должно существенно тормозить работу МК. Разработчики МК251 пошли по несколько иному пути. Система команд МК251 содержит 111 команд, входящих в систему команд семейства MCS-51 («старые» команды) и, кроме того, 157 «новых». Для различения «старых» и «новых» команд, у которых коды совпадают, используется префикс $A5_{16}$.

МК семейства MCS-251 могут выполнять программу в обычном или расширенном режиме. Выбор режима определяется значением одного из битов в байте конфигурации, который записывается в МК при программировании. В обычном режиме МК251 выполняет все команды подобно МК51, т. е. полностью совместим программно. Однако в этом режиме МК доступны и новые команды. Если при работе в этом режиме МК встречает код $A5_{16}$, то считает следующий код «новой» командой. В расширенном режиме у МК251 отсутствует полная программная совместимость с МК51, но существенно возрастает производительность.

Рассмотренные выше МК относятся к классу 8-разрядных устройств. В 1982 году фирма Intel начала выпуск 16-разрядных МК семейства MCS-196/296. Особенностью этих является регистр-регистровая архитектура, при которой отсутствует аккумулятор. Роль аккумулятора при выполнении команд может выполнять любой внутренний регистр. С одной стороны, это усложняет формат команд, с другой — делает их более гибкими. МК семейства MCS-196/296 отличаются многообразием встроенных периферийных устройств (порты ввода/вывода, таймеры, WDT, встроенные генераторы сигналов, в том числе для управления трехфазными электродвигателями и т. д.) и развитой системой команд, поддерживающей операции цифровой обработки сигналов. Это позволяет с успехом использовать эти МК при создании систем, реализующих сложные законы управления [1].

5.3.2 PIC — микроконтроллеры фирмы Microchip

Выпускаемые фирмой Microchip Technology Inc МК принято сокращенно называть PIC, от Peripheral Interface Controller. Фирма выпускает несколько семейств PIC: PIC15C/16C/17C. Приведенная ниже информация относится, в основном, к

МК семейства PIC16C, однако архитектура PIC характерна для МК всех семейств.

Подобно рассмотренным выше МК PIC строятся по гарвардской архитектуре с разделением памяти программ и памяти данных. Однако в PIC реализовано одно из главных достоинств такой архитектуры — возможность одновременного обращения и к памяти программ и к памяти данных. Тем самым повышается быстродействие PIC.

Другой особенностью PIC является его система команд. Все команды состоят из двух байт и большинство из команд выполняются за два машинных цикла. Цикл равен 4 периодам тактового сигнала. В PIC реализована конвейерная обработка команд, когда одновременно с исполнением текущей команды производится считывание следующей. В результате реальная длительность выполнения одной команды равна длительности одного машинного цикла. Следовательно PIC можно отнести к RISC-устройствам.

МК PIC16C не имеют внешних шин, что не позволяет задействовать внешнюю память. В этой серии выпускаются как МК с минимальным набором, включающим один 8-разрядный таймер-счетчик и 13 линий ввода/вывода, так и расширенные версии, содержащие до трех таймеров различной конфигурации с модулями сравнения-защелки и ШИМ, модуль последовательного интерфейса, поддерживающий несколько стандартов синхронной и асинхронной передачи данных, драйвер ЖК-дисплея и многоканальный АЦП. Рассмотрим особенности некоторых периферийных устройств.

Помимо стандартного блока универсального синхронно-асинхронного приемопередатчика, являющегося средством обеспечения последовательной связи многих современных МК, большинство МК семейства PIC16C содержат модуль синхронного последовательного порта (SSP), поддерживающий стандарты SPI и I²C. Интерфейс SPI (Serial Peripheral Interface) — это трехпроводная синхронная линия связи, предназначенная для подключения внешних устройств с последовательным доступом. Интерфейс I²C (Inter-Integrated Circuit), разработанный фирмой Philips, обладает более развитым по сравнению с SPI протоколом обмена данными и подходит как для простого соединения двух устройств, так и для объединения большого числа источников и приемников данных посредством двухпроводной двунаправленной линии. Простота использования шины I²C позволяет ей составить серьезную конкуренцию асинхронному интерфейсу RS-232, упрощенные версии которого наиболее часто применяют для обмена данными в простых микропроцессорных системах.

Некоторые PIC семейства PIC16C имеют в своем составе вспомогательный параллельный порт (Parallel Slave Port), который может использоваться при подключении нескольких устройств к общей 8-разрядной шине. При этом порт PIC работает в режиме 8-разрядной двунаправленной защелки данных с высокоимпедансным состоянием выходов и тремя дополнительными линиями \overline{CS} , \overline{RD} и \overline{WR} , по которым можно управлять режимами выборки порта, чтения и записи в порт с аппаратным формированием соответствующего прерывания. Для подключения, такого PIC, например, к шине ISA, имеющейся в персональных компьютерах, понадобится добавить только дешифратора адреса. При этом остаются доступными

для использования практически все выводы РС: входы АЦП, последовательных портов, линии управления таймеров и т. д. Очевидно, что скорость обмена данными при параллельном подсоединении РС будет существенно выше, чем при использовании последовательных интерфейсов.

Особенности РС позволяют рекомендовать их для построения относительно простых систем управления, в которых не требуется сколько-нибудь серьезная обработка данных.

6 Устройства обеспечения взаимодействия микропроцессорной системы с внешними объектами

6.1 Преобразователи сигналов

Как показано на схеме управления, см. рис. 1, контроллер взаимодействует с датчиками и исполнительными механизмами при помощи устройств, преобразующих аналоговые сигналы в цифровые и обратно. В качестве этих устройств обычно выступают АЦП и ЦАП. Как отмечалось выше, АЦП может являться встроенным периферийным устройством МК, а функцию преобразования цифрового сигнала в аналоговый может выполнять встроенный в МК ШИМ. Поэтому рассмотрим вопросы подключения преобразователей сигналов только применительно к микропроцессорной системе с 8-разрядной шиной данных.

Не рассматривая внутреннюю структуру и принцип работы существующих АЦП, их можно разделить на устройства параллельного и последовательного обмена. В первых выходной сигнал передается в параллельной форме по линиям, количество которых совпадает с разрядностью АЦП. Во вторых выходной сигнал передается в последовательной форме по одной линии.

Рассмотрим способ подключения АЦП параллельного обмена. Как отмечалось выше, для подключения некоторого устройства к шине данных микропроцессорной системы требуется, чтобы линии этого устройства могли переходить в высокоимпедансное состояние при отсутствии сигнала выбора этого устройства. Если предполагаемый к использованию АЦП не имеет такого свойства, то задача его подключения несколько усложняется. В качестве примера рассмотрим АЦП K1113PB1A, УГО которого представлено на рис. 45,а. Данная ИС представляет собой 10-разрядный АЦП поразрядного кодирования с представлением выходной информации в двоичном прямом коде. Входной аналоговый сигнал подается на вход IN. Запуск преобразования осуществляют задним фронтом сигнала на входе ST. По истечению времени преобразования t_s (30 мкс для данного АЦП) будет сформирован сигнал низкого уровня на выходе готовности данных RDY и на линиях D0...D9 появится двоичный код, эквивалентный входному напряжению, рис. 45,б. Как видно из этого рисунка, линии D0...D9 способны находиться в высокоимпедансном состоянии. Однако, перевод этих линий в активное состояние осуществляется внутренним сигналом АЦП RDY, а не внешним сигналом выбора. Поэтому между выходами АЦП и линиями шины данных необходимо дополнительное устройство — буфер с возможностью управляемого перевода линий в высокоимпедансно-состояние. Т. к. разрядность АЦП

высокоимпедансное состояние. Т. к. разрядность АЦП превышает разрядность шины данных, то считывание информации из АЦП следует производить в два приема: считать сначала состояние восьми младших разрядов, а затем оставшихся старших. Следовательно, потребуются два буферных устройства. В целом, при работе с АЦП МП будет взаимодействовать с 4 периферийными устройствами: устройством, в которое будет записываться сигнал начала преобразования, устройством, из которого будет считываться сигнал готовности АЦП и двумя буферными устройствами. Каждое из этих периферийных устройств должно иметь свой адрес, обращение к ним осуществляется через дешифраторы адреса.

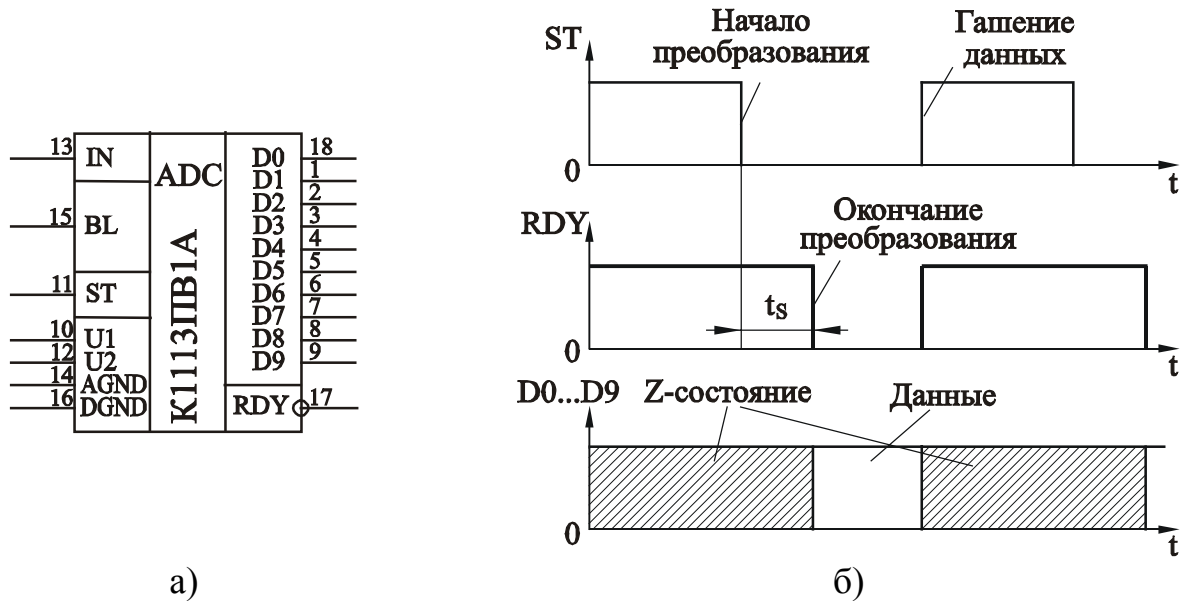


Рис. 45

Устройство, в которое будет записываться сигнал начала преобразования ST можно построить с помощью D-триггера, рис. 46,а. Схема спроектирована из условия, что дешифратор адреса при обращении к этому устройству вырабатывает сигнал низкого уровня. При выполнении команды записи в это устройство (по соответствующему адресу) на выходе триггера появится уровень, определяемый состоянием разряда D0 шины данных. Очевидно, что для запуска преобразования АЦП необходимо записать в это устройство любой байт, младший разряд которого равен «0».

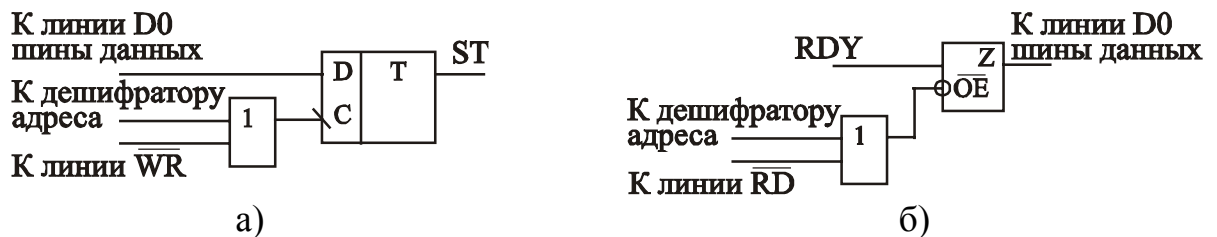


Рис. 46

Проверить состояние готовности АЦП следует, считывая данные из периферийного устройства, возможная схема которого показана на рис. 46,б. Однораз-

рядный буферный элемент с высокоимпедансным состоянием выходной линии пропустит сигнал RDY АЦП на младшую линию шины данных только при выполнении операции чтения из этого устройства.

После обнаружения сигнала готовности МП должен считать результаты преобразования. Периферийные устройства, из которых считываются результаты, по своей структуре аналогичны показанному на рис. 46,б, однако в них должны использоваться многоразрядные буферные элементы. В качестве таких элементов можно использовать регистр, например, КР580ИР82, или шинный драйвер, например, КР1533АП5, рис. 15.

Многие современные АЦП иностранного производства имеют вход \overline{CS} , низкий уровень сигнала на котором переводит выходные линии АЦП из высокоимпедансного состояния в активное. Подключение таких АЦП не требует буферных элементов между его выходами и шиной данных.

Подключение к микропроцессорной системе АЦП с последовательным обменом также требует дополнительных устройств, т. к. протокол обмена таких АЦП достаточно сложен и требует выдерживать на управляющих входах АЦП сигналы заданных уровней в течение определенного времени. Рассмотрим АЦП TLC548 фирмы Texas Instruments, рис. 47,а. Данное устройство представляет собой 8-разрядный АЦП, дополненный схемой выборки/хранения, тактовым генератором и системой последовательного ввода/вывода. На рис. 47,б показаны временные диаграммы работы рассматриваемой ИС.

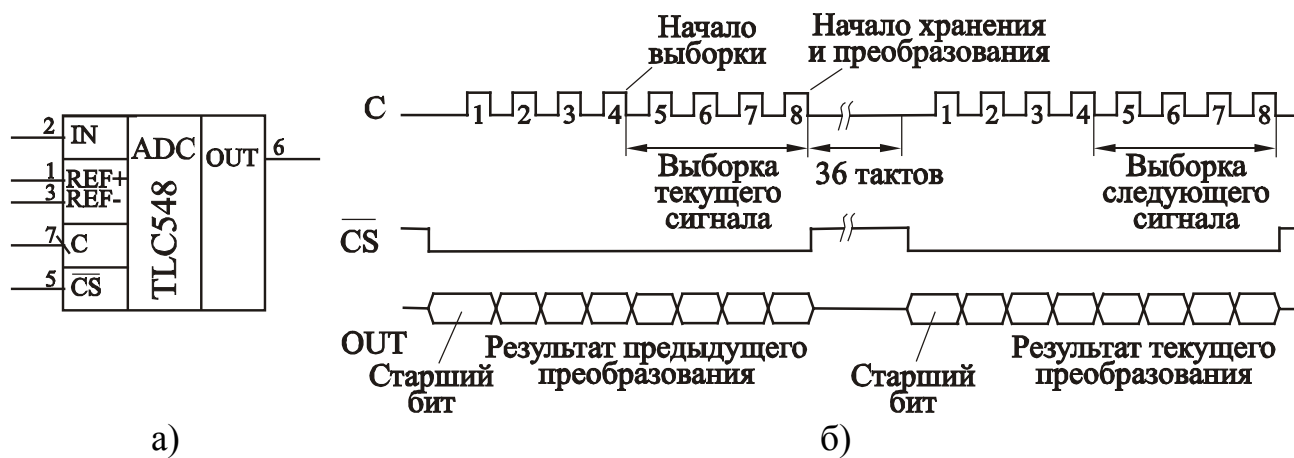


Рис. 47

Если на вход \overline{CS} АЦП TLC548 подан сигнал высокого уровня, то выход OUT находится в высокоимпедансном состоянии. Работа АЦП разрешена при $\overline{CS} = 0$. Появление такого сигнала автоматически выводит на выходную линию старший бит результата предыдущего преобразования. В дальнейшем состояние на выходной линии меняется по заднему фронту внешнего тактового сигнала.

Задний фронт 4-го внешнего тактового импульса запускает процесс выборки текущего состояния сигнала, а 8-го — процесс хранения и преобразования. Хранение и преобразование длится 36 тактов встроенного генератора, что составляет

примерно 17 мкс. Во время хранения и преобразования необходимо обеспечить $\overline{CS} = 1$. Выдача результата преобразования начнется при установке $\overline{CS} = 0$.

Если обмен с рассматриваемым АЦП предполагается вести через шину данных микропроцессорной системы, то необходимо реализовать три периферийных устройства, каждое со своим адресом. Устройства, в которые МП будет записывать сигналы, предназначенные для линий C и \overline{CS} АЦП должны содержать запинаяющие элементы — триггеры, и могут быть построены по схемам, аналогичным приведенной на рис. 46,а. Для считывания результатов преобразования потребуется устройство, аналогичное показанному на рис. 46,б.

Устройства, схемы которых показаны на рис. 46, достаточно универсальны и могут применяться при обращении МП к любым периферийным устройствам, в частности, к ЦАП.

В системах, построенных с использованием МК, подключение ИС АЦП и ЦАП целесообразно производить с использованием собственных портов ввода/вывода МК. При этом на линиях портов необходимо формировать управляющие сигналы в соответствии с временными диаграммами работы конкретных ИС.

6.2 Кнопочная клавиатура

Большинство микропроцессорных систем являются незамкнутыми со стороны пользователя устройствами, т. е. эти системы в обязательном порядке должны иметь устройства для поддержания взаимодействия с пользователем. В качестве устройства ввода информации от пользователя в микропроцессорную систему обычно выступает кнопочная клавиатура. Устройством вывода (отображения) информации обычно является некоторое устройство индикации, реализованное, например, на семисегментных индикаторах.

Рассмотрим вопросы, связанные с организацией в микропроцессорной системе кнопочной клавиатуры. В наиболее простом случае для ввода информации с кнопочной клавиатуры требуются только линии настроенного на ввод порта ввода/вывода, реализованного, например, с использованием ППА КР580ВВ55А или собственных линий ввода/вывода используемого МК. На рис. 48 показана схема простой 8-кнопочной клавиатуры, подключаемой к порту ввода.

Ввод информации при помощи этой клавиатуры осуществляется следующим образом. Пусть, например, кнопка $SB1$ не нажата. В этом случае на соответствующей линии порта будет присутствовать сигнал высокого уровня, формируемый источником напряжения U . В случае нажатия кнопки $SB1$ произойдет замыкание соответствующей линии порта на общий провод, в результате чего на линии будет сформирован сигнал с низким логическим уровнем. Резистор $R1$ служит для ограничения тока, потребляемого от источника U в этом случае. Исходя из этого величина данного резистора должна быть по возможности большой. Однако этот же резистор будет задавать ток, втекающий в линию порта в случае не нажатой кнопки. Поэтому величина резистора ограничена допустимым минимальным током. На практике используют резисторы сопротивлением 1...10 кОм. Если ис-

пользуемый порт имеет внутренние «подтягивающие» резисторы, то применять внешние нет необходимости.

Аналогичная ситуация возникает при нажатии любой кнопки данной клавиатуры: высокий активный уровень на соответствующей линии порта сменяется низким. Если программно организовать просмотр состояния линий порта, то микропроцессорная система отследит нажатие на соответствующую кнопку, т. е. произойдет ввод информации. Данное построение клавиатуры позволяет отслеживать одновременное нажатие нескольких кнопок.

Недостатком рассмотренного построения клавиатуры является задействование относительно большого количества линий порта — по одной на каждую кнопку. В случае незначительного количества кнопок или при наличии большого количества незадействованных линий портов ввода/вывода с этим можно мириться. Однако на практике не редки ситуации, когда клавиатура микропроцессорной системы должна иметь десятков и более кнопок, а количество линий портов ввода/вывода ограничено. В этом случае применяют иное схемотехническое решение. Клавиатуру организуют в виде сетки (матрицы), в узлах которой располагаются собственно кнопки, рис. 49,а.

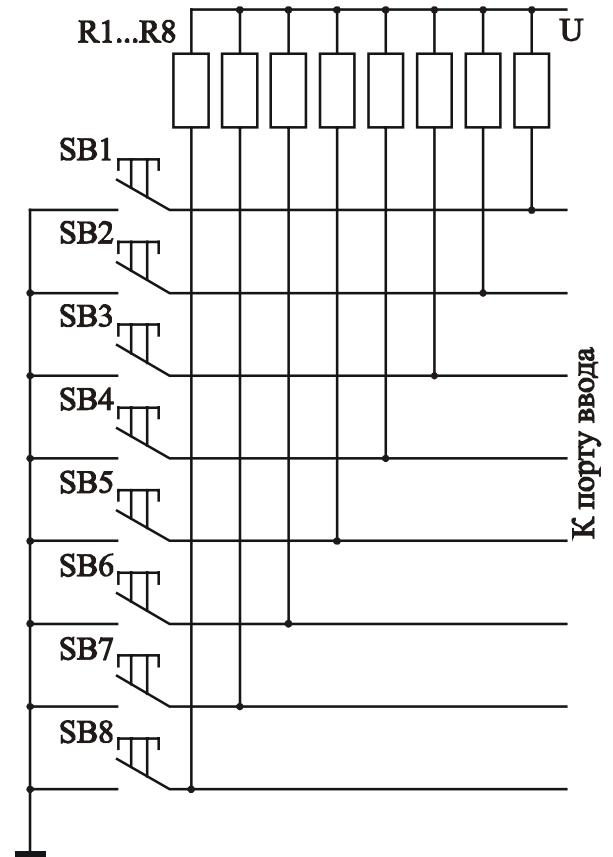


Рис. 48

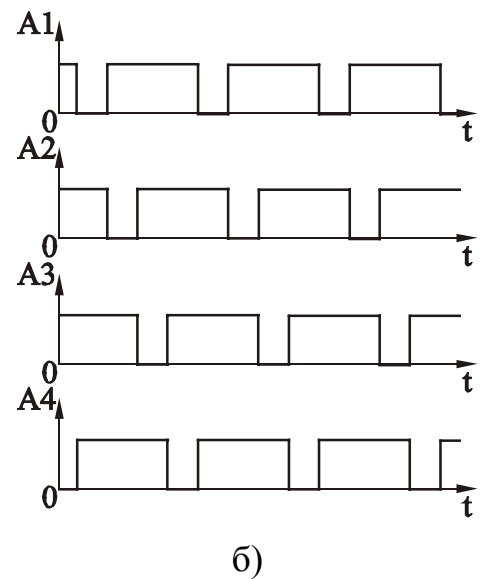
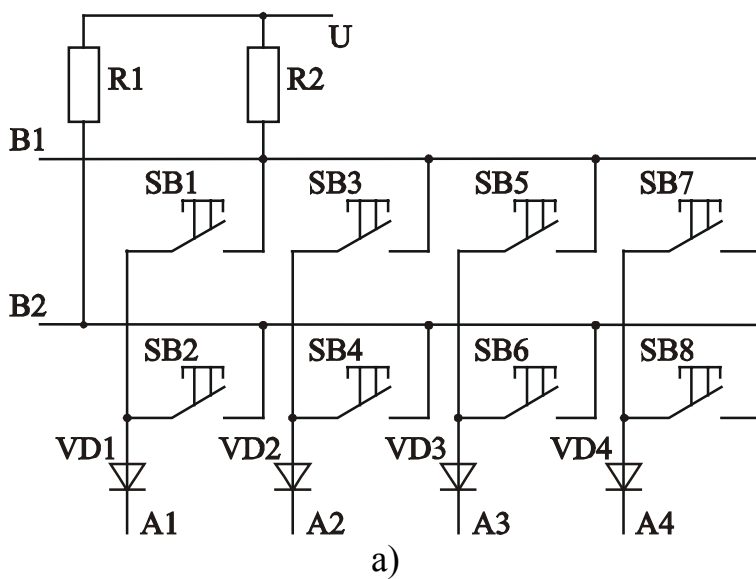


Рис. 49

Работа данной матричной клавиатуры строится следующим образом. На линиях В1, В4 будут присутствовать высокие уровни, т. к. эти линии подключены через резисторы к источнику напряжения. Пусть в некоторый момент на линии А1 присутствует сигнал низкого уровня, а на линиях А2...А4 — высокого. Данные уровни могут быть сформированы, например, портом вывода. Допустимый втекающий ток низкого уровня для этого устройства (порта вывода) должен быть больше тока, определяемого из выражения $I = \frac{U}{R}$, где $R = R1 = R2$. При нажатии

на кнопку SB1 на линии В1 установится сигнал низкого уровня, состояние сигнала на линии В2 не изменится. Если нажать кнопку SB2, то сигнал низкого уровня появится на линии В2. Очевидно, что нажатие кнопок SB3...SB8 не изменит уровни на линиях В_і. Таким образом, анализируя состояние линий В_і и зная, что на низкий уровень установлен на линии А1, можно определить, какая из кнопок SB1 или SB2 нажата.

Чтобы проанализировать состояние кнопок SB3 и SB4, необходимо установить сигнал низкого уровня на линии А2, а на линии А1 — высокого уровня. В этом случае появление сигнала низкого уровня, например, на линии В1 будет свидетельствовать о нажатии на кнопку SB3. Проверка состояния кнопок SB5...SB8 производится аналогично.

Линии А_і, состояние которых задается портом вывода, можно назвать входными линиями матричной клавиатуры. Соответственно, линии В_і можно назвать выходными линиями клавиатуры. Подключают эти линии к порту ввода.

Применительно к матричной клавиатуре можно говорить об активных и пассивных уровнях сигналах. Пассивным будет являться уровень сигнала, который присутствует на выходных линиях клавиатуры, когда не нажата ни одна кнопка.

В общем случае работа микропроцессорной системы по обслуживанию матричной клавиатуры строится следующим образом.

- На первой из входных линий устанавливается активный сигнал.
- Анализируется состояние выходных линий.
- При нахождении выходной линии, на которой присутствует активный уровень, делается вывод о нажатии соответствующей кнопки.
- Активный сигнал убирается с первой входной линии и устанавливается на вторую.
- Анализируется состояние выходных линий, и т.д. в бесконечном цикле.

Сигнал на каждой входной линии имеет вид последовательности прямоугольных импульсов, скважность которых пропорциональна количеству входных линий. Между собой сигналы на входных линиях сдвинуты во времени. На рис. 49,б показаны входные сигналы для случая четырех входных линий, причем активным является сигнал низкого уровня, т. е. для схемы клавиатуры, показанной на рис. 49,а. Входные сигналы такого вида часто называют сигналами сканирования линий клавиатуры.

Обычно между портом вывода и линиями A_i включают диоды, как это показано на рис. 49,а. Полярность включения диодов определяется уровнем активного сигнала. На рис. 49,а показан случай, когда активный сигнал имеет низкий уровень. Данные диоды защищают линии порта от чрезмерных нагрузок в случае нажатия нескольких кнопок, связанных с разными входными линиями, но с одной выходной. Пусть, например, нажаты одновременно кнопки SB1 и SB5, причем, активна линия A1. В этом случае выходной ток линии порта, соединенной с входной линией A2, потечет через кнопки SB2 и SB1 в линию порта, соединенную с входной линией A1, что, по сути, эквивалентно замыканию линии порта, соединенной с входной линией A2, на общий провод. Эта ситуация может привести к выходу из строя линий порта. Диоды VD_i предотвращают появление описанной цепи протекания тока, тем самым защищая линии порта.

С точки зрения программного обеспечения работы клавиатуры следует отметить, что в большинстве случаев требуется, чтобы система управления реагировала на нажатие той или иной кнопки в любом из своих режимов работы. Иными словами, проверка состояния клавиатуры должна производиться постоянно. Однако, система управления помимо обслуживания клавиатуры должна выполнять свои основные функции расчета и управления. Поэтому для работы с клавиатурой широко используют прерывания. Например, для простой клавиатуры, показанной на рис. 48, все линии дополнительно могут быть соединены с входами логического элемента 8И, выход которого может использоваться для формирования сигнала запроса прерывания. В этом случае программа анализа состояния клавиатуры должна быть реализована как подпрограмма обработки этого прерывания. Данная подпрограмма будет вызываться каждый раз, как только будет нажата любая кнопка; в функцию этой подпрограммы будет входить определение нажатой кнопки. Все остальное время анализ состояния клавиатуры производиться не будет и система управления сможет выполнять другие действия.

Применительно к матричной клавиатуре может быть использовано аналогичное схемотехническое решение, однако его реализация вызовет существенное усложнение схемы. Поэтому на практике обычно поступают следующим образом. Программа анализа состояния клавиатуры также реализуется в виде подпрограммы обработки прерывания, но само прерывание периодически запрашивается внешним устройством, например, таймером или генератором импульсов. Подпрограмма обработки прерывания должна не только анализировать состояние выходных линий клавиатуры, но и изменять сигналы на входных линиях клавиатуры.

При нажатии на кнопку сигнал на соответствующей линии принимает новое значение не сразу. Из-за неидеальности контактов возникает процесс, называемый дребезгом: в момент замыкания или размыкания контактов формируется последовательность импульсов, рис. 50,а. В большинстве случаев дребезг является нежелательным процессом, т. к. может восприниматься системой как многократное нажатие на одну кнопку. Длительность процесса дребезга индивидуально для каждой кнопки, но обычно не превышает десятков мс.

Негативное влияние дребезга контактов устраняют аппаратными или программными средствами. В первом случае между кнопкой и портом ввода устанавливают дополнительное устройство, например, показанное на рис. 50,б. RC-цепь в приведенном устройстве должна обеспечивать временную за-

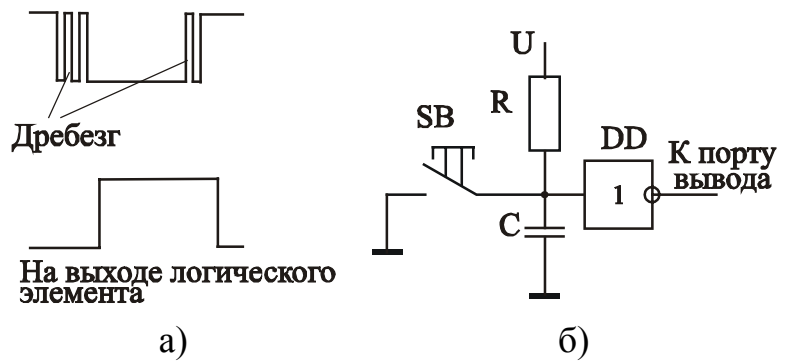


Рис. 50

держку переключения логического элемента после нажатия или отпущения кнопки, большую времени протекания процесса дребезга. Логический элемент формирует выходной сигнал с крутыми фронтами. Очевидно, что для матричной клавиатуры подобное устройство применить затруднительно.

При программном устранении дребезга контактов необходимо опрашивать состояние одной кнопки несколько (обычно 2 или 3) раз с интервалом, превышающим время процесса дребезга. Если в результате опросов зафиксированы различные состояния кнопки, т. е. наблюдается дребезг, то окончательное решение о состоянии кнопки не принимается, а организуется новая серия опросов.

В системах управления широко используются датчики, обычно нажимного действия, которые замыкают или размыкают цепь протекания тока. По своей сути такие датчики могут рассматриваться как кнопки, поэтому на них распространяются схемные решения, рассмотренные для кнопочной клавиатуры.

6.3 Устройства отображения информации

6.3.1 Табло на полупроводниковых светодиодах

Основным элементом отображения информации в микропроцессорных системах управления являются светодиоды, дискретные или включенные в состав семисегментных или матричных индикаторов. Использование дискретного светодиода обычно не вызывает затруднений: для управления им используется отдельная линия порта вывода, необходимый ток формируется либо ключом на биполярном транзисторе, либо логическим элементом, рис. 51.

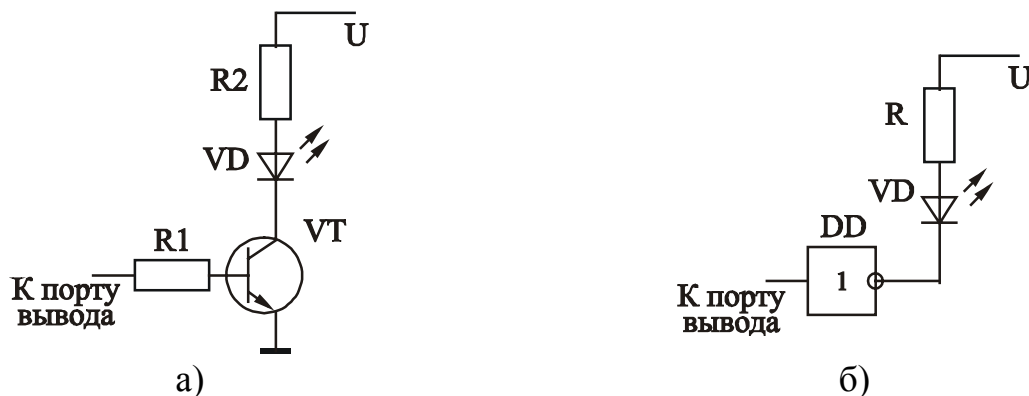


Рис. 51

Расчет ключевого транзисторного каскада сводится к выбору транзистора по заданному коллекторному току и определению сопротивлений $R1$ и $R2$ исходя из заданного тока через светодиод I_{VD} . Очевидно, что ток коллектора равен I_{VD} , и задается резистором $R2$. Сопротивление $R2$ определяется как $R2 \approx \frac{U - U_{VD}}{I_{VD}}$, где U_{VD} — падение напряжения на светодиоде. Сопротивление $R1$ определяется как $R1 \approx \frac{(U_{УПР} - 0,7)h_{21}}{I_{VD}}$, где $U_{УПР}$ — напряжение высокого уровня на выходе порта вывода, h_{21} — коэффициент передачи используемого транзистора.

Рассмотрим построение устройства отображения информации с использованием семисегментных индикаторов. Различают два режима работы светодиодных индикаторов в микропроцессорных системах — так называемые режимы статической и динамической индикации. Режим статической индикации наиболее прост в реализации — каждый светодиод индикатора через буферный усилитель подключается к линии порта вывода. Соответственно, количество задействованных линий портов соответствует количеству используемых элементов индикации. Поэтому данный режим применяется при незначительном количестве элементов индикации или при наличии нужного числа незадействованных линий портов. Расчет элементов, на которых реализована статическая индикация, сводится к выбору нужного элемента индикации, задания для него тока I_{VD} и расчету буферного усилителя, в качестве которого обычно выступают простейшие транзисторные ключевые каскады или логические элементы. Использование индикаторов в статическом режиме ничем не отличается от использования соответствующего количества дискретных светодиодов.

Характерной особенностью режима статической индикации является возможность одновременно зажечь все элементы индикации. Однако в некоторых многознаковых семисегментных индикаторах и всех матричных индикаторах принципиально невозможно зажечь все имеющиеся в них светодиоды, поэтому статическая индикация к данным приборам не применима.

При динамической индикации в каждый момент времени может быть зажжена только вполне определенная часть элементов индикации, которую назовем группой. Построение устройств динамической индикации в схемотехническом плане сходно с построением матричной клавиатуры, однако все линии в схеме индикации являются входными. На рис. 52 представлена схема устройства динамической индикации на четырех индикаторах. В схеме использованы индикаторы с общим анодом. Ключи на транзисторах $VT1...VT4$ отвечают за подачу напряжения на общие аноды индикаторов $HL1...HL4$ соответственно. Данные транзисторные ключи открываются при поступлении по линиям $IND1...IND4$ сигналов низкого уровня. Ключи на транзисторах $VT5...VT12$ отвечают за зажигание собственно сегментов индикаторов. Из схемы видно, что при подаче, например, напряжения высокого уровня по линии A_{IN} откроется транзистор $VT5$, и сегменты «А» всех индикаторов смогут зажечься. Однако загорится сегмент «А» только у тех инди-

катодов, у которых на общий анод будет подано напряжение U . Например, если подать сигнал низкого уровня только на линию IND1, обеспечив на линиях IND2...IND4 высокие уровни, то загорится сегмент «А» только индикатора HL1.

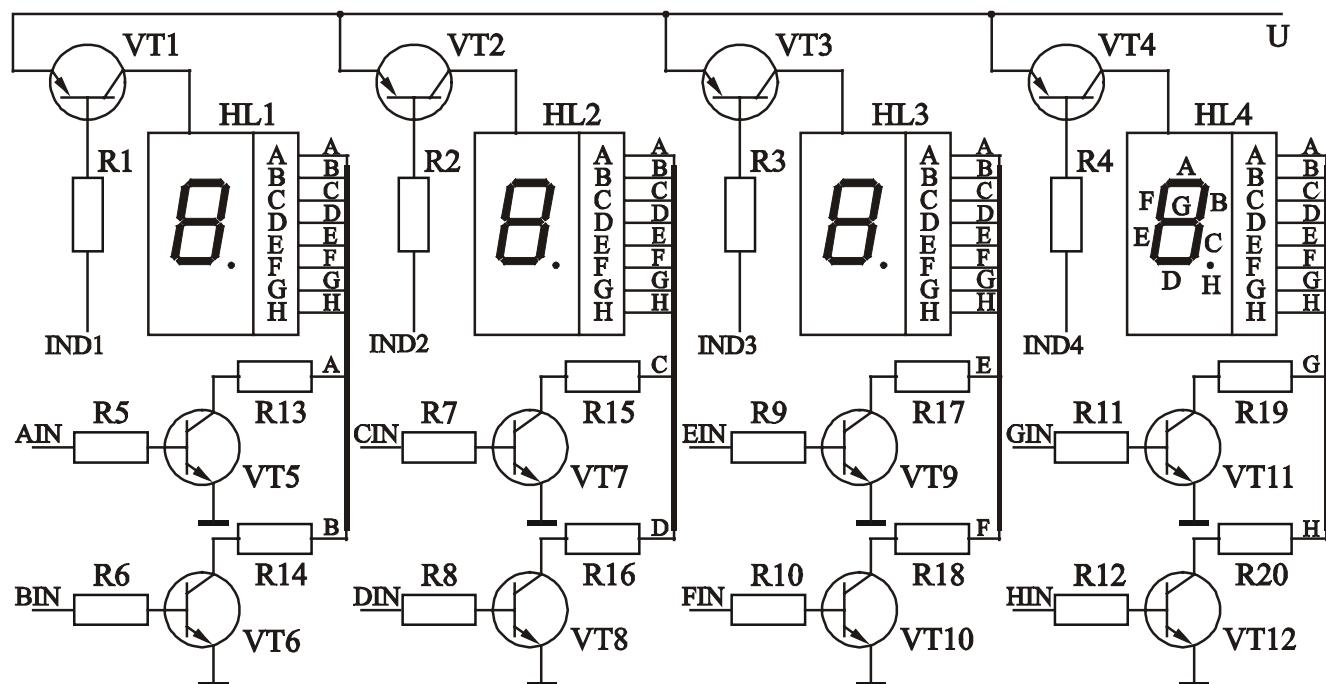


Рис. 52

Работу представленной схемы динамической индикации рассмотрим на примере формирования надписи «2102». Для того должны быть осуществлены следующие действия.

- В исходном состоянии на линии IND1...IND4 должно быть подано напряжение высокого уровня, что позволит погасить все элементы индикации.
- На линии AIN...HIN подаются сигналы, которые должны обеспечить зажигание на индикаторе HL1 цифры «2». В рассматриваемом примере требуемая совокупность сигналов может быть описана двоичным числом 11011010_2 , в котором старший разряд соответствует сигналу на линии AIN, младший — на линии HIN.
- На линию IND1 подается сигнал низкого уровня. Это вызовет появление на индикаторе HL1 цифры «2».
- Обеспечивается выдержка времени, в течение которого горит цифра на индикаторе HL1.
- На линию IND1 подается сигнал высокого уровня, что гасит цифру «2».
- На линии AIN...HIN подаются сигналы, описываемые числом 01100000_2 , которые должны обеспечить зажигание на индикаторе HL2 цифры «1».
- На линию IND2 подается сигнал низкого логического уровня. Это вызовет появление на индикаторе HL2 цифры «1».
- Обеспечивается выдержка времени, в течение которого горит цифра на индикаторе HL2.
- На линию IND2 подается сигнал высокого уровня, что гасит цифру «1».

- На линии АIN...НIN подаются сигналы, описываемые числом 1111100_2 , которые должны обеспечить зажигание на индикаторе HL3 цифры «0».
- На линию IND3 подается сигнал низкого логического уровня. Это вызовет появление на индикаторе HL3 цифры «0».
- Обеспечивается выдержка времени, в течение которого горит цифра на индикаторе HL3.
- На линию IND3 подается сигнал высокого уровня, что гасит цифру «0».
- На линии АIN...НIN подаются сигналы, описываемые числом 11011010_2 , которые должны обеспечить зажигание на индикаторе HL4 цифры «2».
- На линию IND4 подается сигнал низкого логического уровня. Это вызовет появление на индикаторе HL4 цифры «2».
- Обеспечивается выдержка времени, в течение которого горит цифра на индикаторе HL4.
- На линию IND4 подается сигнал высокого логического уровня, что гасит цифру «2».
- Далее вновь обеспечивается отображение цифры на индикаторе HL1 и цикл повторяется.

В рассмотренном устройстве динамической индикации одновременно могут гореть только элементы, объединенные в семисегментный индикатор, т. е. группой является сам индикатор. Несколько иначе строится схема, реализующая режим динамической индикации для матричных светодиодных индикаторов. Такой индикатор представляет собой набор размещенных в виде матрицы светодиодов, объединенных по строкам и столбцам. Размер матрицы может быть различным: индикатор ЗЛС340А имеет матрицу 5×7 , ЗЛС347А — 8×8 . При работе с матричными индикаторами группу составляют светодиоды горизонтальной или вертикальной линии матрицы, рис. 53. В приведенной схеме элементы строки могут светиться при подаче на соответствующую линию СТР_і сигнала высокого уровня; элементы столбца — при подаче на соответствующий вход СТБ_і сигнала низкого уровня. Элемент будет светиться, если на линии, соответствующие строке и столбцу, на пересечении которых он находится, подать активные уровни сигналов. В целом, работа схемы с матричным индикатором не отличается от работы схемы с семисегментным индикатором.

Устройства отображения информации на матричных индикаторах более универсальны, т. к. способны отображать любую информация, вплоть до графической, однако схемы управления ими сложнее из-за большого количества групп.

Если в системе предполагается использовать большое количество дискретных светодиодов при ограниченном количестве выходных линий портов, то их можно рассматривать как элементы матрицы светодиодов и, обеспечив соответствующее соединение, реализовать динамическую индикацию.

При построении системы динамической индикации важно правильно выбрать время отображения информации группой элементов $t_{от}$. Для элементов индикации, установленных на неподвижных устройствах, рекомендуется частота возоб-

новления информации $f_{\text{ВОЗ}} = 100 \text{ Гц}$. Следовательно, время отображения информации группой элементов $t_{\text{ОТ}} = \frac{1}{f_{\text{ВОЗ}}n}$, где n — количество групп. Занижение частоты возобновления может привести к эффекту мерцания индикаторов, чрезмерное увеличение может привести к общему снижению яркости свечения индикаторов из-за инерционности транзисторных ключей.

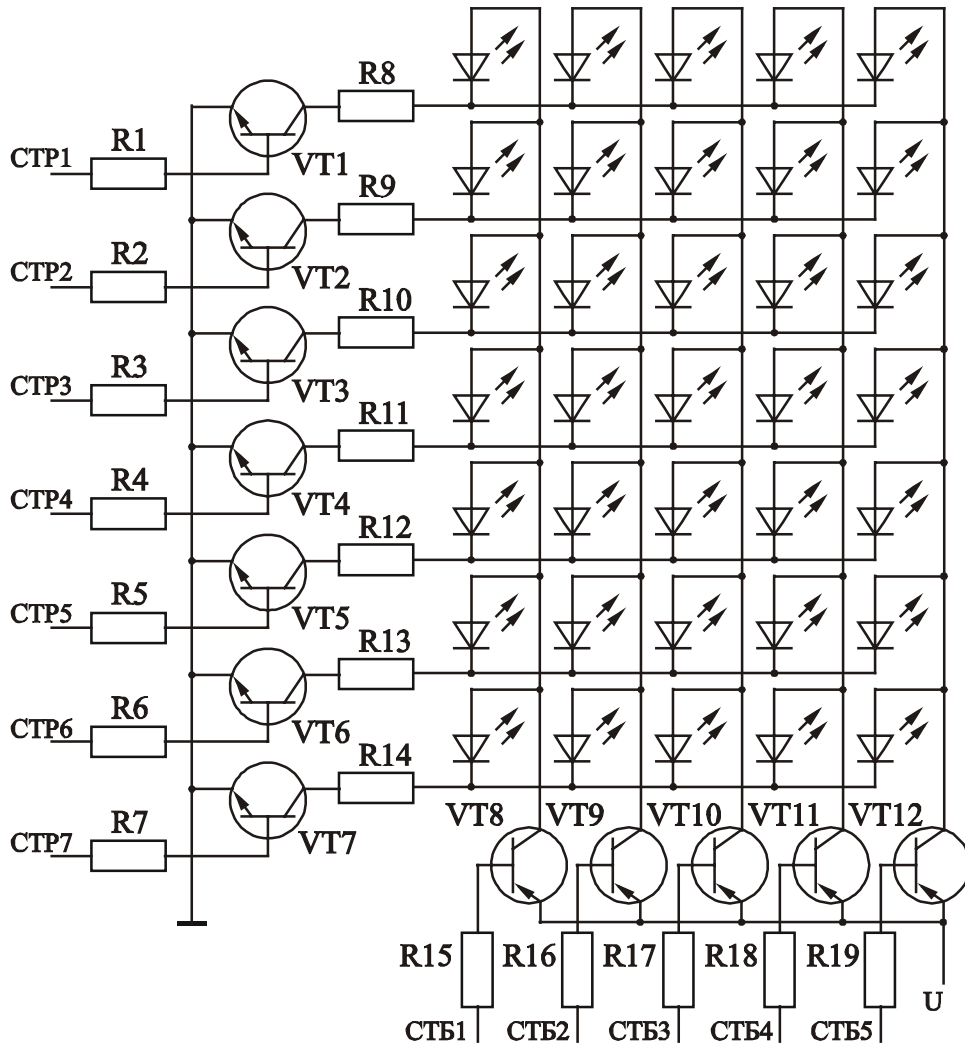


Рис. 53

Ток, текущий через элемент индикации, работающий в динамическом режиме, носит импульсный характер. Без учета интервала времени гашения всех групп можно считать, что этот ток имеет скважность, равную n ; средний ток равен $I_{\text{СР}} = \frac{I_{\text{ИМП}}}{\sqrt{n}}$, где $I_{\text{ИМП}}$ — максимальный ток в импульсе. Если $I_{\text{ИМП}}$ равен току $I_{\text{СТ}}$, текущему через элемент в статическом режиме, то будет очевидна экономичность режима динамической индикации — потребляемая мощность снизится в n раз. Однако, снижение среднего тока при динамической индикации приведет к снижению яркости свечения индикаторов, что может оказаться нежелательным. Для повышения яркости увеличивают ток $I_{\text{ИМП}}$, снижая экономичность. Многие полу-

проводниковые светодиоды имеют нелинейную зависимость яркости свечения от протекающего тока — увеличение амплитуды импульсного тока приводит к существенному увеличению яркости. Таким образом, при динамической индикации для обеспечения требуемой яркости, сопоставимой с яркостью индикаторов в статическом режиме, не требуется повышать импульсный ток в \sqrt{n} раз. Обычно достаточным бывает его значение, обеспечивающее $I_{CP} = \frac{I_{CT}}{2}$, что обеспечивает четырехкратное снижение потребляемой мощности.

При расчете ключевых каскадов, используемых в устройствах динамической индикации, следует ориентироваться на обеспечение требуемого тока $I_{ИМП}$. Этот ток определяется исходя из количества групп и заданного рабочего тока элемента индикации, используемого в расчетах как I_{CT} . Для индикаторов может быть задан максимальный ток в импульсном режиме работы, необходимо, чтобы $I_{ИМП}$ не превосходил этого значения.

Подробнее с принципами построения устройств с динамической индикацией можно ознакомиться в [4].

При задействовании в микропроцессорной системе полупроводниковых индикаторов, особенно матричных, и клавиатуры может возникнуть ситуация, когда количество требуемых линий портов вывода меньше имеющегося. Если невозможно увеличить количество имеющихся линий, то ряд схемотехнических решений позволяет уменьшить количество требуемых.

Если в микропроцессорной системе предполагается использовать матричную клавиатуру и устройство отображения с динамической индикацией, то можно объединить входные линии клавиатуры и одни из входных линий устройства индикации — те, которые отвечают за выбор группы элементов индикации. Сигналы на этих линиях очень похожи: и в клавиатуре и в устройстве динамической индикации в каждый момент времени только одна из рассматриваемых линий является активной. Применительно к рассмотренным схемам клавиатуры и индикации, см. рис. 49,а и рис. 52, можно объединить линии A_i и IND_i . В этом случае для обслуживания клавиатуры и устройства индикации потребуется 14 линий вместо 18.

Данное схемное решение является типовым и реализовано в некоторых специализированных ИС, например в ИС КР580ВВ79 — драйвере индикации и клавиатуры. Эта ИС может обслуживать клавиатуру, имеющую до 64 кнопок, и выводить информацию из внутренних регистров на 8- или 16-разрядное табло, построенное на полупроводниковых семисегментных индикаторах. ИС разработана для использования совместно с 8-разрядными МП. Подробное описание этой ИС можно найти в [5], а ее аналога — ИС 8279 — в [7].

Уменьшить количество требуемых для подключения индикаторов линий портов можно за счет использования дополнительных микросхем. Сигналы сканирования клавиатуры, см. рис. 49,б, легко могут быть получены при помощи дешифратора с инверсными выходами, например, КР1533ИД7. Для полного задействования N выходных линий дешифратора потребуется $\log_2 N$ выходных линий пор-

та, т. е. для формирования, например, 8 сканирующих сигналов потребуется всего 3 линии порта. Если дешифратор дополнить счетчиком, то можно обойтись всего двумя линиями порта: по одной линии будут передаваться счетные импульсы, передвигающие активный уровень на выходах дешифратора, по другой линии должен поступать импульс, обнуляющий счетчик в момент выбора первого индикатора. Вместо отдельных ИС дешифратора и счетчика можно использовать ИС счетчика-дешифратора, например, К561ИЕ8.

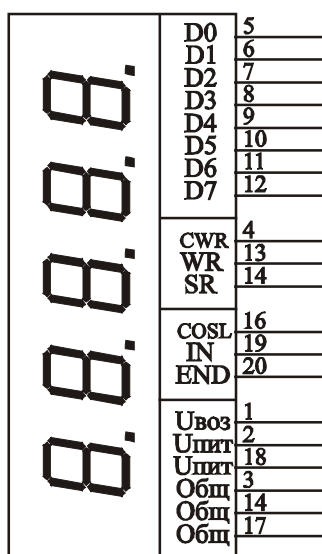
Существуют специальные ИС дешифраторов, предназначенные для управления семисегментными индикаторами, например, К514ИД1, которые можно использовать, если на индикаторах предполагается отображение только цифровой и некоторой псевдографической информации. Выходные сигналы данных ИС могут управлять табло по линиям АIN...GIN. Т. к. ИС управляется по 4 входам, то их использование позволит сэкономить еще 3 линии порта вывода.

Если в устройстве предполагается использовать простую клавиатуру, аналогичную по структуре показанной на рис. 48, то целесообразно задействовать ИС шифратора, уменьшив тем самым количество требуемых линий портов.

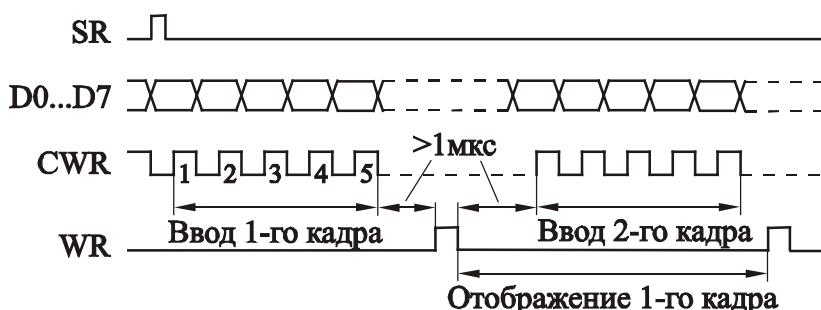
6.3.2 Табло на жидкокристаллических индикаторах

В настоящее время в качестве элементов устройств отображения информации используются жидкокристаллические индикаторы (ЖКИ). Особенность таких индикаторов является низкое энергопотребление, что позволяет использовать их в аппаратуре с батарейным питанием. С принципами формирования изображений в ЖКИ можно ознакомиться в [3].

Отечественная промышленность выпускает несколько типов ЖКИ, которые могут быть задействованы в микропроцессорных системах управления. Например, прибор ИЖЦ71-5/7 представляет собой 5-разрядный индикатор, предназначенный для отображения цифровой информации в семисегментном представлении. Для упрощения использования индикатора он снабжен встроенной схемой управления. УГО данного индикатора показано на рис. 54,а.



а)



б)

Рис. 54

На рис. 54,б показаны временные диаграммы, поясняющие операцию занесения информации в индикатор ИЖЦ71-5/7. Сигналом начала ввода в него информации является импульс на линии SR, который сбрасывает внутренние регистры индикатора. По переднему фронту следующего за ним тактового импульса CWR во входной регистр будет записана информация, находящаяся на линиях D_i . Данная информация определит состояние сегментов первого разряда. Каждая из линий D_i отвечает за зажигание соответствующего сегмента в разряде: линия D0 — за сегмент «А», линия D1 — за сегмент «В» и т. д. Сегмент может зажигаться как высоким уровнем сигнала на соответствующей линии D_i , так и низким, что определяется сигналом на линии IN. По фронту следующего импульса CWR во входной регистр будет занесена информация о состоянии сегментов второго разряда и т. д. Пятым импульсом CWR будет завершён ввод в индикатор 1-го кадра. Введенная информация будет отображена в соответствующих разрядах индикатора после прихода импульса записи WR. Этот импульс в дальнейшем играет роль сигнала начала ввода нового кадра — фронт следующего за ним импульса CWR перепишет во входной регистр информацию с линий D_i .

Во время ввода нового кадра старый будет отображаться на индикаторе вплоть до прихода нового импульса записи.

Индикатор питается напряжением 4,5...5,5 В, уровни входных сигналов соответствуют уровням ТТЛ. Для нормальной работы индикатора необходимо на вывод COSL подавать прямоугольные импульсы с периодом от 5 до 20 мс и скважностью 2. Частота тактовых импульсов не должна превышать 400 кГц.

Использовать рассмотренный ЖКИ в качестве устройства отображения информации в микропроцессорной системе целесообразно совместно с ППА КР580ВВ55А.

Описанный ЖКИ способен отображать только цифровую и простую псевдографическую информацию. Если необходимо отображать текстовую информацию, то следует использовать буквенно-цифровые матричные ЖКИ. Примером такого индикатора может служить прибор ИЖВ74 – 160×16 отечественного производства. На табло прибора размещены две основные строки длиной 149,1 мм, состоящие из 32 знакомест в каждой. Знакоместо имеет матричную структуру из 7×5 элементов прямоугольной формы. Размер элемента 0,8×0,6 мм, размер знакоместа 6,2×3,4 мм. Под каждой из строк расположена курсорная строка, состоящая из одинарного ряда элементов того же размера. Курсорные строки позволяют формировать перемещающиеся метки под тем или иным знаком основной строки. В общей сложности на табло имеется 2560 элементов, организованных в 160 столбцов и 16 строк.

Работа с индикатором ИЖВ74 – 160×16 строится аналогично работе с ИЖЦ71-5/7: процесс занесения информации о состоянии текущей строки тактируется импульсами CWR, новое состояние строки отображается после поступления импульса WR, этот же импульс начинает ввод информации о состоянии следующей строки. Информация в индикатор вводится по линиям D0...D7. Импульс SR осуществляет кадровую синхронизацию, т. е. определяет ввод информации с

левого элемента верхней строки. Информация записывается слева направо по строкам и сверху вниз по столбцам.

В целом работа с индикатором ИЖВ74 – 160×16 достаточно сложна, т. к. внешнее устройство должно хранить в памяти выводимую на него информацию в текстовом виде и преобразовывать ее для построчной передачи. При такой организации работы изменение состояния одного знакоместа — одной буквы в сообщении потребует перезаписи информации о состоянии всех 2560 элементов табло.

В настоящее время существует большое количество ЖКИ иностранного производства, которые имеют в своем составе более мощную по сравнению с ИЖВ74 – 160×16 систему управления. Занесение информации в такие индикаторы сводится обычно к записи данных в ячейки внутренней памяти, каждая из которых отвечает за определенное знакоместо.

Принципы работы с современными ЖКИ иностранного производства рассмотрим на примере прибора РС1602 фирмы POWER TIP. Прибор способен отображать две строки по 16 символов в каждой. Система управления этого ЖКИ построена на специализированной ИС — контроллере HD44780. Контроллер принимает от внешнего устройства, например, МП данные и сигналы управления, и формирует на табло соответствующую буквенно-цифровую информацию.

На рис. 55,а показано УГО ЖКИ РС1602.

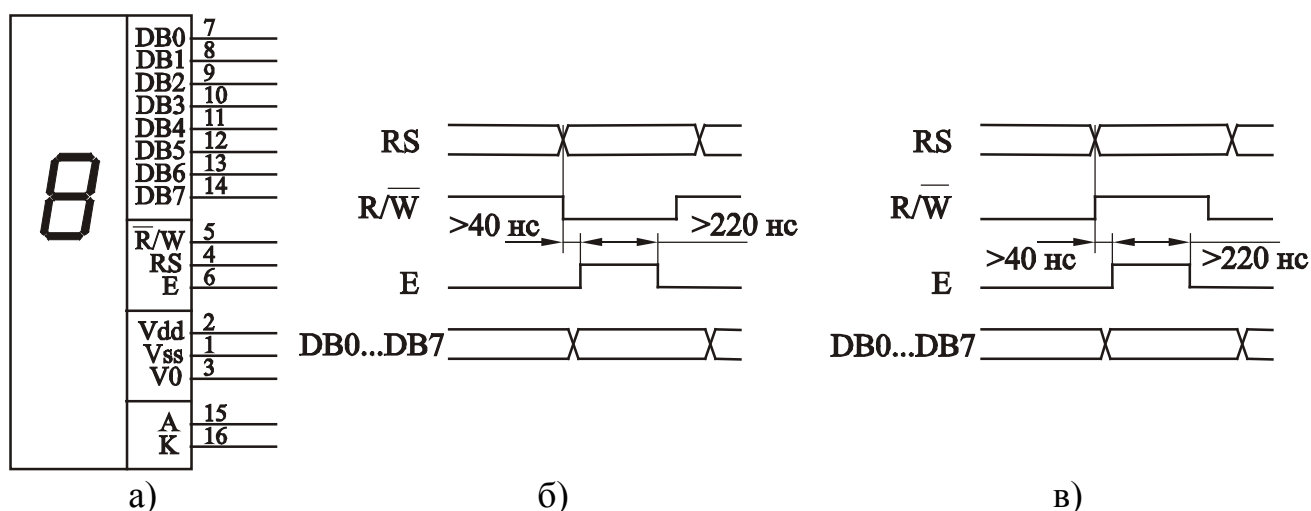


Рис. 55

Рассмотрим назначение линий ЖКИ: RS — выбор регистра команд ($RS = 0$) или регистра данных ($RS = 1$), R/\bar{W} — чтение/запись, E — разрешение чтения/записи, DB7...DB4 — старшие разряды шины данных, DB3...DB0 — младшие разряды шины данных, Vss — общий провод, Vdd — питание +5 В, V0 — управление жидкокристаллической панелью, A, K — питание подсветки. Разделение шины данных на младшие и старшие разряды обусловлено тем, что ЖКИ поддерживает как 8-разрядный, так и 4-разрядный режимы обмена с внешним устройством. При использовании 4-разрядного режима информация передается по старшим линиям шины данных DB7...DB4. Линии DB3...DB0 не используются. Цикл передачи одного байта между ЖКИ и внешним устройством занимает две

последовательные 4-битовые посылки, причем старшие биты передаются первыми. При 8-разрядном обмене данные пересылаются одной 8-битовой посылкой по линиям DB7...DB0.

Уровни сигналов по всем линиям совместимы с ТТЛ. При $E = 0$ линии данных переводятся в высокоимпедансное состояние.

Как отмечено выше, система управления ЖКИ реализована с использованием контроллера HD44780. Рассмотрим подробнее структуру этой ИС. Контроллер HD44780 имеет два 8-разрядных регистра: регистр данных DR и регистр инструкций IR. Тот или иной регистр выбирается сигналом RS. Регистр данных DR является промежуточным буфером для данных, записываемых во внутреннее ОЗУ или считываемых из него. Регистр IR запоминает коды инструкций (команд), таких как очистка дисплея, сдвиг курсора, а также адресную информацию для внутреннего ОЗУ. Регистр IR доступен только для записи. При чтении из регистра IR (т. е. при $RS = 1$) считывается состояние флага занятости BF (по линии DB7) и счетчика адреса AC (по линиям DB6...DB0).

Флаг занятости BF установлен, когда контроллер находится в режиме выполнения команды и не может считать следующую. Перед посылкой новой команды необходимо убедиться, что флаг занятости сброшен.

Счетчик адреса AC задает адрес ячейки внутреннего ОЗУ. При выполнении команды установки адреса адресная информация пересылается из регистра IR в счетчик адреса. После выполнения операции с внутренним ОЗУ счетчик адреса автоматически инкрементируется или декрементируется.

В контроллере реализовано внутреннее ОЗУ двух типов: внутреннее видеоОЗУ — 80 ячеек, в которых хранится информация, предназначенная для отображения на табло, а также внутреннее ОЗУ знакогенератора. Под знакогенератором понимается таблица соответствия информации, хранимой в ячейке внутреннего видеоОЗУ, изображению, формируемому контроллером в знакоместе. Основной знакогенератор находится во внутреннем ПЗУ и не может быть изменен. Знакогенератор внутреннего ПЗУ формирует знаки размером 5×7 или 5×10 точек, в соответствии 8-разрядными двоичными числами. Например, число 31_{16} позволит отобразить цифру «1», число 52_{16} — букву «R», число $7A_{16}$ — букву «z».

Дополнительный знакогенератор, находящийся в ОЗУ, позволяет отображать символы, не предусмотренные основным знакогенератором, а запрограммированные пользователем.

На рис. 55,б,в показаны временные диаграммы операций обмена внешнего устройства с контроллером HD44780. До появления переднего фронта сигнала E необходимо при помощи сигналов RS и R/\bar{W} сообщить контроллеру, какая операция будет выполняться. Собственно запись или чтение информации осуществляется по заднему фронту сигнала E .

Контроллер способен выполнять инструкции, приведенные в табл. 14.

Инструкция «Очистка табло» записывает код пробела (20_{16}) во все ячейки видеоОЗУ, устанавливает счетчик адреса видеоОЗУ в нулевое значение, сдвигается курсор в левый край первой строчки. Инструкция «Возврат на начало» действует аналогично, но не заносит код пробела в ячейки видеоОЗУ.

Таблица 14

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Инструкция
0	0	0	0	0	0	0	0	Очистка табло
0	0	0	0	0	0	1	x	Возврат в начало
0	0	0	0	0	1	I/D	S	Режим ввода
0	0	0	0	1	D	C	B	Вкл./выкл.
0	0	0	1	S/C	R/L	x	x	Сдвиг курсора/табло
0	0	1	D/L	N	F	x	x	Установка функций
0	1	Адрес в ОЗУ знакогенератора						Установка адреса ячейки ОЗУ знакогенератора
1	Адрес видеоОЗУ						Установка адреса ячейки видеоОЗУ	

Режим ввода задается разрядами I/D и S. При I/D=1 содержимое счетчика адреса автоматически инкрементируется после выполнения операции записи или чтения, при I/D=0 — декрементируется. Если S=1, то отображаемая на табло информация будет смещаться относительно неподвижного курсора при выполнении операции записи в видеоОЗУ. Сдвиг происходит на одну позицию вправо при I/D=0 и влево при I/D=1.

Инструкция включения/выключения позволяет управлять отображением информации на табло. При D=0 табло полностью выключено, однако информация в видеоОЗУ сохраняется и будет выведена на табло сразу после его включения установкой D=1. Курсор включен при C=1 и выключен при C=0. Установка B=1 обеспечит мерцание символа, расположенного над позицией курсора.

Инструкция «Сдвиг курсора/табло» позволяет сдвигать курсор или всю информацию на табло влево или вправо без выполнения операций записи или чтения, табл. 15.

Таблица 15

S/C	R/L	Функция	Примечание
0	0	Сдвиг курсора на позицию влево	Счетчик адреса декрементируется на единицу
0	1	Сдвиг курсора на позицию вправо	Счетчик адреса инкрементируется на единицу
1	0	Сдвиг всего табло на позицию влево, вместе с курсором	Счетчик адреса не меняется
1	1	Сдвиг всего табло на позицию вправо, вместе с курсором.	Счетчик адреса не меняется

Инструкция «Установка функций» позволяет задать разрядность шины данных: 8-разрядная при D/L=1, 4-разрядная при D/L=0, количество строк на табло: N=0 — одна строка, N=1 — две строки, размер знакоместа: 5×7 при F=0, 5×10 при F=1.

При установке адреса знакогенератора содержимое разрядов DB5...DB0 переписывается в счетчик адреса ОЗУ знакогенератора. Передаваемые затем данные будут адресованы именно к этой ячейке.

При установке адреса видеоОЗУ в качестве адреса ячейки этого ОЗУ, заносимого в соответствующий счетчик адреса, используются сигналы на линиях DB6...DB0. Передаваемые затем данные будут адресованы к этой ячейки видеоОЗУ. Необходимо отметить, что при $N = 0$ (однострочное табло) адрес видеоОЗУ должен находиться в пределах $00_{16}...4F_{16}$ включительно, при $N = 1$ (двухстрочное табло) — в пределах $00_{16}...27_{16}$ (первая строка) и $40_{16}...67_{16}$ (вторая строка) включительно.

После подачи питания на ЖКИ инициируется внутренняя система сброса контроллера HD44780. При инициализации выполняются следующие инструкции: «Очистка табло», «Установка функций» ($D/L = 1, N = 0, F = 0$), «Включение/выключение» ($D = 0, C = 0, B = 0$), «Режим ввода» ($I/D = 1, S = 0$). Флаг занятости остается установленным до конца инициализации.

Если время установления напряжения питания больше 10 мс, то внутренняя схема сброса может сработать некорректно. В этом случае инициализацию ЖКИ должно выполнить внешнее устройство.

Предположим, что внутренняя инициализация проведена корректно. В табл. 16 приведена последовательность инструкций, которая позволит отобразить на табло двухстрочную надпись.

Прибор PC1602 может быть задействован в микропроцессорной системе двумя способами. Во-первых, все его управляющие линии и линии шины данных могут быть подключены к ней с использованием портов ввода/вывода, например, ППА KP580BB55A, рис. 56,а, или непосредственно используемого МК. Данный способ легко реализуем практически, однако в контроллере с МП требует использовать дополнительную ИС ППА. Во-вторых, ЖКИ PC1602 может входить в систему как отдельное периферийное устройство. Такое подключение возможно, т. к. при отсутствии сигнала разрешение чтения/записи E линии DB7...DB0 переводятся в высокоимпедансное состояние. Рассмотрим второй способ подключения применительно к микропроцессорной системе, показанной на рис. 23.

Фактически внутри прибора находятся два периферийных устройства: регистры DR и IR, каждому из которых необходимо назначить свой адрес. Выбор того или иного регистра осуществляется сигналом RS, поэтому этот сигнал целесообразно формировать из их адресов. Например, линию RS ЖКИ можно соединить с линией A0 шины адреса микропроцессорной системы и назначить адреса регистров так, чтобы они отличались только состоянием младшего разряда. Тогда, при $A0 = 0$ произойдет обращение к регистру IR, а при $A0 = 1$ — к регистру DR.

ЖКИ PC1602 не имеет входа выбора кристалла. Поэтому необходимо внешними цепями обеспечить прохождение на ЖКИ активного уровня сигнала разрешение чтения/записи E только при обращении МП одному из регистров ЖКИ. Это легко обеспечить при помощи логического элемента ИЛИ. Собственно сигналом разрешение чтения/записи может являться проинвертированный сигнал \overline{IORQ} МП, т. к. его длительность превышает 220 нс. Сигналом низкого уровня,

разрешающим прохождение $\overline{\text{IORQ}}$ на вход Е ЖКИ, может являться сигнал на линии А2 шины адреса.

Таблица 16

Инструкция	Состояние дисплея	Комментарий
Установка функций 0 0 1 1 0 0 x x RS = 0, R / \overline{W} = 0		Устанавливает 8-разрядный режим обмена, 2-строчное табло, размер знакоместа 5 × 7.
Вкл./выкл. 0 0 0 0 1 1 1 0 RS = 0, R / \overline{W} = 0		Включает табло и курсор. Табло заполнено символами пробела после внутренней инициализации.
Режим ввода 0 0 0 0 0 1 1 0 RS = 0, R / \overline{W} = 0	—	Устанавливает автоматическую инкрементацию АС и сдвиг курсора вправо при выполнении операций с ОЗУ. Табло не сдвигается.
Запись данных 0 1 0 1 0 1 1 1 RS = 1, R / \overline{W} = 0	W	Записывает символ «W» в нулевую ячейку видеоОЗУ, которая заранее выбрана при инициализации. Содержимое АС инкрементируется.
Запись данных 0 1 1 0 0 1 0 1 RS = 1, R / \overline{W} = 0	We	Записывает символ «e» в следующую ячейку видеоОЗУ.
...
Запись данных 0 1 1 0 0 1 0 1 RS = 1, R / \overline{W} = 0	Wellcome	Записывает символ «e» в следующую ячейку видеоОЗУ.
Установка адреса видеоОЗУ 1 1 0 0 0 0 0 0 RS = 0, R / \overline{W} = 0	Wellcome	Устанавливает адрес 40 ₁₆ , что переводит курсор в первую позицию второй строки.
Сдвиг курсора/табло 0 0 0 1 0 1 x x RS = 0, R / \overline{W} = 0	Wellcome	Сдвигает курсор на одну позицию вправо.
Запись данных 0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 0	Wellcome 2102	Выводит во второй строке начиная со второго знакоместа сообщение «2102»

В МП Z80 при обращении к периферийным устройствам сигналы \overline{WR} и \overline{RD} формируются одновременно с сигналом $\overline{\text{IORQ}}$, см. рис. 14, поэтому необходимо

формировать сигнал R/\overline{W} иным способом. Данный сигнал можно сформировать, например, из адреса регистра, но каждый регистр будет иметь два адреса — один для чтения из него, второй для записи. Если задействовать линию A1 шины адреса, то в системе фактически появятся 4 периферийных устройства, расположенных по следующим адресам: 11111000_2 — регистр IR, работающий в режиме записи, 11111001_2 — регистр DR, работающий в режиме записи, 11111010_2 — регистр IR, работающий в режиме чтения, 11111011_2 — регистр DR, работающий в режиме чтения. Схема подключения ЖКИ к микропроцессорной системе показана на рис. 56,б.

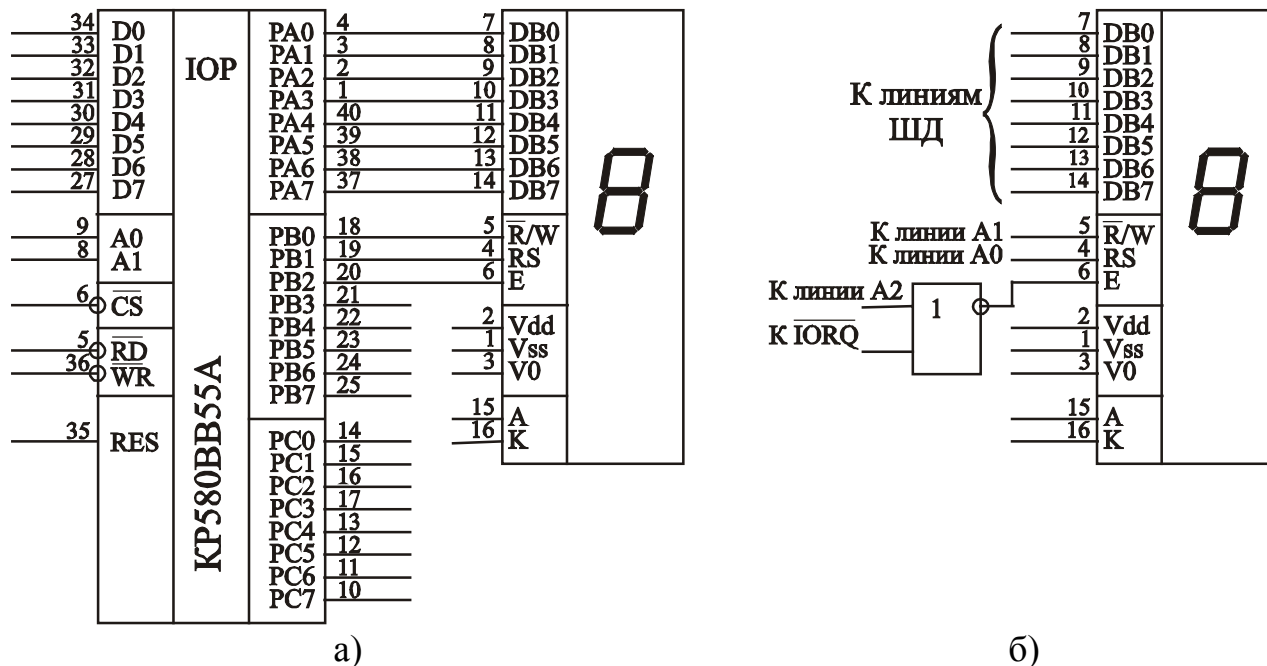


Рис. 56

Реализованный способ адресации к регистрам ЖКИ можно рассматривать как разновидность работы с коммутируемыми периферийными устройствами. Аналогичным образом ЖКИ PC1602 и подобные ему могут быть подключены к любой микропроцессорной системе.

7 Вспомогательные устройства микропроцессорных систем

7.1 Микросхемы памяти с интерфейсом I²C

Если микропроцессорная система должна сохранять некоторые данные после отключения питания, то возникает потребность включения в ее состав ПЗУ, которое может перепрограммировано самой системой. В настоящее время в качестве такого ПЗУ широко используют ИС многократно репрограммируемых ПЗУ с электрическим стиранием и записью данных (ИС EEPROM или Flash-памяти), обмен с которыми производится по шине I²C.

Разработанный фирмой Philips двухпроводный интерфейс I²C позволяет организовать двунаправленный обмен между несколькими десятками устройств. Дан-

ный интерфейс поддерживается многими МК и ИС различного назначения. Рассмотрим, как организуется обмен при помощи этого интерфейса.

Собственно шина состоит из двух линий: SCL (Serial Clock) и SDA (Serial DATA). Линии нагружены резисторами, соединенными с источником питания. К линиям подключаются одноименные входы/выходы устройств. Входы/выходы должны иметь открытый коллектор.

Одно из устройств служит ведущим (master), остальные — ведомыми (slave). Ведущее устройство генерирует синхроимпульсы SCL и управляет обменом. Ведомые принимают или передают данные только в ответ на запросы ведущего. На рис. 57 показан характер изменения сигналов на линиях SDA и SCL при пересылке информации. Исходное состояние шины — высокие уровни сигналов на линиях SDA и SCL. В начале сеанса обмена ведущий формирует команду СТАРТ: не меняя состояния линии SCL, устанавливает низкий уровень на линии SDA, а затем — такой же уровень на линии SCL. Передача информации — побитная. При низком уровне SCL передатчик (во время передачи первого байта это тоже ведущий, но в дальнейшем им может стать любой из ведомых) устанавливает на линии SDA уровень, соответствующий значению передаваемого бита: «1» — высокий, «0» — низкий. Затем ведущий устанавливает на линии SCL высокий уровень, а по истечении времени, отведенного приемнику для принятия информации, вновь сменяет его низким. Данная процедура повторяется для каждого передаваемого бита. Первым передают старший бит байта, последним — младший. Принявший байт устройство подтверждает факт приема, установив на линии SDA низкий уровень. Чтобы не мешать этому, после передачи последнего бита одного байта передатчик должен установить на линии SDA сигнал высокого уровня. После подтверждения приема байта начинается передача следующего или подается команда окончания сеанса СТОП — смена низкого уровня на линии SDA высоким при высоком уровне на линии SCL.

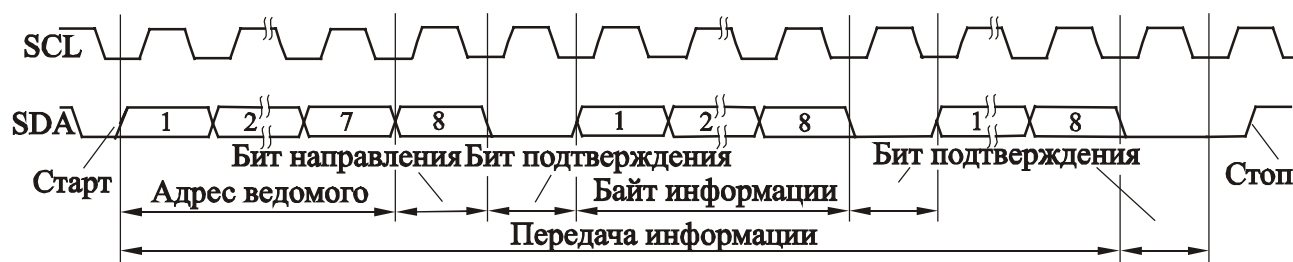


Рис. 57

Семь старших битов байта, обязательно передаваемого ведущим вслед за командой СТАРТ, представляют собой адрес ведомого, с которым он намерен установить связь. Если среди подключенных к шине есть исправное устройство с таким адресом, оно должно подтвердить прием и подготовиться к дальнейшим действиям. Все остальные ведомые, не опознав своего адреса, отключаются до следующей команды СТАРТ.

Младший восьмой бит первого байта — признак направления передачи. Если его значение «0», то информация в данном сеансе будет передаваться от ведущего

к ведомому, в противном случае ведущий будет читать данные, передаваемые ведомым. Число передаваемых за сеанс байт не ограничено, но смена направления передачи до завершения сеанса не предусмотрена.

На рис. 58 показана схема подключения к шине I²C восьми ИС EEPROM.

Сигналами, поданными на входы A0...A2, задают 3 младших разряда адреса ИС как ведомого шины I²C. Поэтому комбинации сигналов на этих входах у нескольких ИС, подключенных к шине, будут разными, см. рис. 58.

Старшие 4 разряда адреса ИС памяти всегда равны 1010₂.

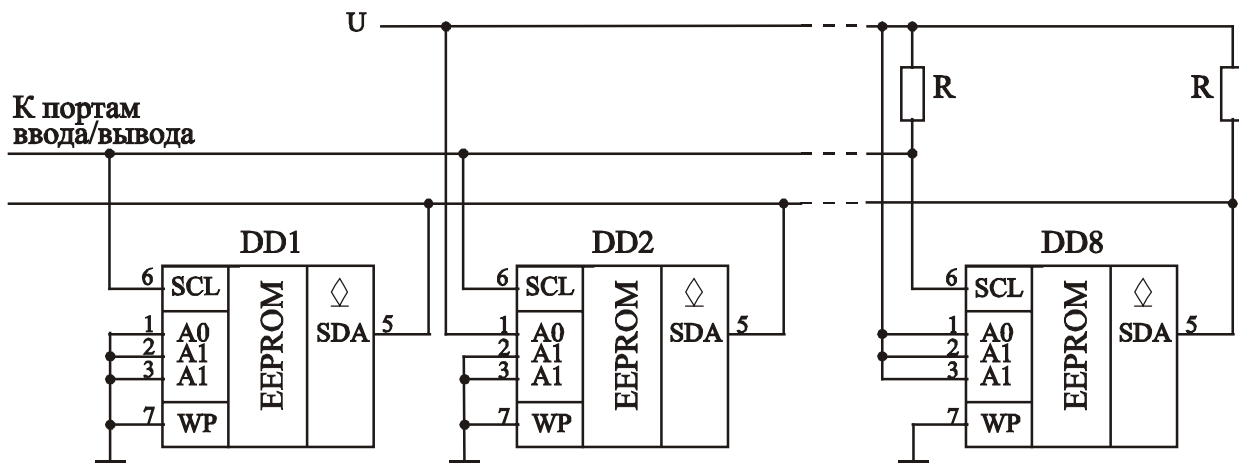


Рис. 58

Вход WP (Write Protect) предназначен для управления защитой записанных в ИС данных. Если этот вывод оставлен свободным или соединен с общим проводом, можно изменять содержимое любых ячеек. В противном случае весь массив памяти или его часть защищены от стирания и записи. Выпускают микросхемы, например, AT34C02, в которых предусмотрены участки памяти, запись в которые можно заблокировать командами ведущего.

Начало взаимодействия ведущего с ИС Flash-памяти следующее: команда СТАРТ, байт с адресом ведомого и установленным признаком записи («1» в младшем разряде), подтверждение приема. Далее ведущий посылает один или два байта адреса ячейки памяти, прием каждого из которых должен быть подтвержден ведомым. При двухбайтном адресе первым передается байт со старшими разрядами.

Дальнейшие действия зависят от того, намерен ли ведущий читать данные, хранящиеся в массиве памяти ведомого, или записывать их туда. Для записи одного или нескольких байт их достаточно передать вслед за адресом ячейки памяти. Первый попадет в заданную ячейку, после чего внутренний контроллер ИС памяти автоматически инкрементирует адрес и следующий байт будет направлен в следующую ячейку. Число последовательно переданных байт данных не ограничено.

После передачи и подтверждения приема всех данных ведущий подает команду СТОП, запускающую в ИС памяти внутренний автомат записи. Продолжительность выполняемой им процедуры не зависит от числа изменяемых байтов.

До окончания процедуры программирования ИС памяти не реагирует ни на какие внешние сигналы и в течение этого времени на повторные обращения ведущего по ее адресу не откликается.

Перед чтением данных не требуется обязательно указывать адрес ячейки. Если ведущий обращается к ИС памяти, установив в младшем бите байта адреса ведомого признак чтения, в ответ ему будет передан байт из ячейки, следующей за той, с которой выполнялась последняя операция записи или чтения, после чего счетчик адреса будет автоматически инкрементирован. Выдача ведомым данных на линию SDA синхронизируется импульсами SCL, генерируемыми ведущим. Он же обязан подтвердить прием.

При необходимости адрес читаемой ячейки задают в явном виде. Для этого ведущий обращается к ведомому с признаком записи и посылает ему один или два байта адреса ячейки. Получив подтверждение, он посылает новую команду СТАРТ, а за ней — адрес ведомого с признаком чтения и выполняет собственно чтение. Первым ему будет передан байт из ячейки с указанным адресом.

7.2 Супервизоры

В настоящее время при построении микропроцессорных систем широко используются специализированные ИС — супервизоры. Супервизоры предназначены для выполнения функций генерации сигнала сброса при включении питания, при понижении питания и во время выбросов и провалов напряжения сети, защиты ОЗУ от записи недостоверных данных, выдачи предупреждения о возможной аварии питания, переключения на питание от резервной батареи, сторожевого таймера. Эти функции жизненно необходимы микропроцессорным системам, т. к. гарантируют устойчивость от отказов, связанных со сбоями по цепям питания. Некоторые из этих функций реализованы в МК, но эти МК не всегда могут диагностировать свои собственные сбои. Чтобы система была эффективной и надежной, схема монитора питания должна находиться во внешней ИС супервизора.

Рассмотрим функции, выполняемые супервизором, более детально.

Приведенные выше схемы формирования сигнала сброса при подаче напряжения питания просты, см. рис. 23 и рис. 41, однако не всегда надежно выполняют свои функции. Если напряжение питания системы нарастает относительно медленно, то простая RC-цепь не сможет сформировать сигнал сброса с необходимой малой длительностью фронтов. Схема формирования сигнала сброса супервизора включает в себя источник опорного напряжения, компаратор, таймер и буферный элемент, формирующий сигнала сброса. При подключении системы к питанию схема начинает «отсчет» интервала задержки выдачи сигнала сброса только тогда, когда напряжение питания достигает своего минимально-допустимого уровня, что гарантирует надежный запуск микропроцессорной системы. Обычно супервизоры способны формировать импульс сброса как высокого, так и низкого уровней.

МП (МК) должен функционировать до тех пор, пока напряжение питания остается в допустимых для него пределах. Для гарантии надежной работы система

должна контролировать напряжение питания на предмет его понижения. Понижения напряжения питания редко обладают разрушительными аппаратными последствиями, но они могут вызывать непредсказуемые действия МП (МК), ведущие к сбою программы. После восстановления напряжения питания программа также будет выполняться некорректно, т. е. нормальное функционирование системы не восстановится. Следовательно, МП (МК) должен быть перезапущен сигналом сброса от внешнего устройства — супервизора.

При снижении питания и непредсказуемых действиях МП (МК) в ОЗУ системы может быть записана посторонняя информация — «мусор». Это вызывает потерю данных, которые не могут быть восстановлены последующим сбросом. Чтобы предотвратить эти потери, супервизор имеет возможность подавить сигнал разрешения работы памяти во время провалов или выбросов напряжения питания.

ИС ОЗУ обычно питаются от того же самого источника питания, что и МП (МК). Если требуется хранить большой объем данных после отключения питания, а использование Flash-памяти невозможно, то в системе предусматривают резервный источник питания ИС ОЗУ. В момент выключения микропроцессорной системы ОЗУ подключается к резервной батарее и сохраняет свое содержимое, потребляя от нее очень малый (до нескольких мкА) ток. Схема, которая переключает ОЗУ с главного питания на резервное, должна постоянно находиться в активном состоянии, чтобы выполнить обратное переключение. Поэтому супервизор также подключается к резервной батарее при выключении основного источника питания.

Надежная защита может потребовать большего, чем обнаружение пониженного напряжения и генерация сигнала сброса, т. к. при организованном выключении могут потребоваться другие действия до генерации сигнала сброса. Например, МП (МК) может потребоваться перед отключением или перезапуском сохранить содержимое какого-нибудь регистра в энергонезависимой памяти. Выполнение таких действий возможно, если перед стабилизатором источника питания находятся конденсаторы большой емкости. Напряжение на конденсаторах на несколько вольт больше напряжения питания системы, поэтому после отключения входного напряжения система может некоторое время работать за счет запасенной в них энергии. Контролируя напряжение на конденсаторах, можно получить сигнал раннего предупреждения о пропадании питания. Этот сигнал обычно подается на вход немаскируемого прерывания, что позволяет МП (МК) программно подготовиться к отключению.

Сторожевой таймер, реализованный в супервизоре, несколько отличается от подобных устройств, встроенных в МК. Если в МК для предотвращения срабатывания сторожевого таймера требуется периодически заносить информацию в специальный регистр, то супервизор должен периодически принимать от МП (МК) сигнал подтверждения нормальной работы.

Примерами ИС супервизоров могут служить КР1446СП1, МАХ69х, LTC1235 и др. Супервизоры обеспечивают выполнение всех перечисленных выше функций. Некоторые ИС выполняют только функции, связанные с контролем питающего напряжения. Такие ИС называют мониторами питания.

Библиографический список

1. Козаченко В.Ф. Микроконтроллеры: руководство по применению 16-разрядных микроконтроллеров Intel MCS-196/296 во встроенных системах управления. — М.: Издательство ЭКОМ, 1997. — 688 с.
2. Однокристалльные микроЭВМ/ Боборыкин А.В., Липовецкий Г.П., Литвинский Г.В. и др. — М.: МИКАП, 1994, — 400 с.
3. Пароль Н.В., Кайдалов С.А. Знакосинтезирующие индикаторы и их применение: Справочник. — М.: Радио и связь, 1989. — 128 с.
4. Применение полупроводниковых индикаторов/ Н.Н. Васерин, Н.К. Дадерко, Г.А. Прокофьев; Под ред. Е.С. Липина. — М.: Энергоатомиздат, 1991. — 200 с.
5. Проектирование цифровых устройств на однокристалльных микроконтроллерах/ В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. — М.: Энергоатомиздат, 1990. — 224 с.
6. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем: В 2-х кн. Кн. 1. Пер. с англ. — М.: Мир, 1988. — 312 с., ил.
7. Рафикузаман М. Микропроцессоры и машинное проектирование микропроцессорных систем: В 2-х кн. Кн. 2. Пер. с англ. — М.: Мир, 1988. — 288 с., ил.
8. Фрунзе А.В. Микроконтроллеры? Это же просто! Т. 1. — М.: ООО «ИД СКИМЕН», 2002. — 336 с.
9. Фрунзе А.В. Микроконтроллеры? Это же просто! Т. 2. — М.: ООО «ИД СКИМЕН», 2002. — 392 с.
10. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. — М.: Радио и связь, 1989. — 288 с.

Оглавление

1	Микропроцессорная система с тремя шинами	
1.1	Общая структура микропроцессорной системы с тремя шинами	3
1.2	Типовые представители системных устройств	
1.2.1	Системные устройства хранения информации	6
1.2.2	Программируемый периферийный адаптер КР580ВВ55А	9
1.2.3	Программируемый интервальный таймер КР580ВИ53	12
1.3	Временные диаграммы операций обмена в системе с тремя шинами.	15
2	Микропроцессор Z80	
2.1	Общая характеристика.....	17
2.2	Регистры микропроцессора Z80	19
2.3	Назначение выводов	20
2.4	Временные диаграммы основных машинных циклов.....	22
2.5	Способы адресации в микропроцессоре Z80	25
2.6	Система команд	27
2.7	Реализация некоторых операций	30
3	Построение микропроцессорной системы	
3.1	Шины в реальной микропроцессорной системе	34
3.2	Разделение адресного пространства.....	36
3.2.1	Адресация ИС памяти.....	36
3.2.2	Адресация периферийных устройств.....	40
3.2.3	Адресация в системах с единым адресным пространством	42
3.3	Контроллер на базе микропроцессора Z80.....	43
4	Прерывания в микропроцессорных системах	45
5	Микроконтроллеры	
5.1	Общая характеристика.....	50
5.2	Микроконтроллер 8051 фирмы Intel	
5.2.1	Краткое описание	54
5.2.2	Блок таймеров/счетчиков	56

5.2.3	Организация портов ввода/вывода.....	60
5.2.4	Организация памяти в МК51	66
5.2.5	Контроллер на основе МК51.....	69
5.2.6	Система прерываний в МК51.....	70
5.2.7	Последовательный порт	73
5.3	Обзор микроконтроллеров популярных серий	
5.3.1	Микроконтроллеры фирмы Intel	77
5.3.2	РIS — микроконтроллеры фирмы Microchip.....	81
6	Устройства обеспечения взаимодействия микропроцессорной системы с внешними объектами	
6.1	Преобразователи сигналов	83
6.2	Кнопочная клавиатура	86
6.3	Устройства отображения информации	
6.3.1	Табло на полупроводниковых светодиодах	90
6.3.2	Табло на жидкокристаллических индикаторах.....	96
7	Вспомогательные устройства микропроцессорных систем	
7.1	Микросхемы памяти с интерфейсом I ² C	103
7.2	Супервизоры.....	106
	Библиографический список	108