

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ

«КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

КУРСОВОЙ ПРОЕКТ

по дисциплине

«Архитектура компьютеров»

Руководитель

Ткаченко В.В.

Выполнил

студент группы ИВ-73

Захожий И. А.,

зачетная книжка №7308

Допущен к защите

___/_____ 2010 г.

Защищен

___/_____ 2010 г.

Киев-2010

ОПИСАНИЕ АЛЬБОМА

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

СХЕМА ЭЛЕКТРИЧЕСКАЯ
СТРУКТУРНАЯ

*СХЕМА ЭЛЕКТРИЧЕСКАЯ
ПРИНЦИПИАЛЬНАЯ*

*СХЕМА ЭЛЕКТРИЧЕСКАЯ
ФУНКЦИОНАЛЬНАЯ*

ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ

Содержание

Раздел	Лист
1 Назначение разработки	2
2 Основание для разработки	2
3 Состав устройств	2
4 Исходные данные	3
5 Технические требования	3
6 Перечень документации	4

					<i>ИАЛЦ.462637.002 ТЗ</i>		
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>Техническое задание</i>		
<i>Разработал</i>		<i>Захожий И. А.</i>					
<i>Провер.</i>		<i>Ткаченко В. В.</i>					
<i>Реценз.</i>							
<i>Н. Контр.</i>							
<i>Утвер.</i>		<i>Ткаченко В. В.</i>			<i>НТУУ "КПИ" ФИВТ зр. ИВ-73</i>		
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
						1	4

1. Назначение разработки

МикроЭВМ относится к микропроцессорным средствам вычислительной техники общего назначения. Разработанная ЭВМ специализирована для решения систем линейных уравнений методом отражений. Данная ЭВМ также может быть использована в других областях решения математических задач и управления техническими системами.

2. Основание для разработки

Основанием для разработки, послужило задание на курсовой проект по курсу "Архитектура компьютеров".

3. Состав устройств

В состав микроЭВМ входят:

- процессор*
- оперативное запоминающее устройство*
- внешние устройства*
- постоянное запоминающее устройство*
- блок питания*
- интерфейс внешних устройств*
- КПП*
- КПДП*

В состав процессора входят:

- блок микропрограммного управления*
- блок обработки данных*

					ИАЛЦ.462637.002 ТЗ	Лист
						2
Змн.	Лист	№ докум.	Подпись	Дата		

4. Исходные данные

Исходными данными являются технические характеристики разрабатываемой ЭВМ:

Объем внешней памяти програм : 64 Кбайт(2стр)

Объем внешней памяти данных : 2 Кбайт(12стр)

Количество внешних устройств: 80

Организация системной магистрали: с разделенными шинами адреса и данных

Контроллер прерываний: централизованный

Контроллер прямого доступа к памяти: централизованный

5. Технические требования

Общие требования:

Данная вычислительная машина может работать от сети переменного тока с напряжением 220В $\pm 15\%$ и частотой 50Гц $\pm 5\%$.

По устойчивости к климатическим и механическим воздействиям вычислительная машина должна удовлетворять следующим условиям:

- температура окружающего воздуха от 0 до +50 °C;*
- относительная влажность до 95% при температуре +35 °C;*
- воздействие ускорения до 10 м/с².*

Требования к надежности:

Средняя наработка вычислительной машины на отказ должна быть не менее 5000 часов;

Вычислительная машина должна сохранять заданные параметры надежности после проведения климатических испытаний.

					ИАЛЦ.462637.002 ТЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		3

Конструктивные требования:

Качество сборки и внешний вид блока должны соответствовать следующим требованиям:

Все детали, узлы должны быть прочно закреплены без перекосов, органы управления должны действовать плавно и обеспечивать надежность фиксации;

Все винты, гайки и детали, имеющие резьбу, не должны иметь повреждений и должны быть прочно закреплены;

Все надписи должны быть четкими и ясными;

Все покрытия должны быть прочными, ровными, без царапин и трещин, и обеспечивать защиту от коррозии.

Конструкция изделия должна обеспечивать максимальное удобство при работе, а также удобство доступа к элементам при ремонте.

Вычислительная машина должна быть выполнена в виде отдельного блока.

В конструкции должен максимально использоваться печатный монтаж.

6. Перечень документации

Текстовая документация

Опись альбома: ИАЛЦ 462637.001 ОА

Техническое задание: ИАЛЦ 462637.002 ТЗ

Ведомость проекта: ИАЛЦ 462637.003 ВД

Пояснительная записка: ИАЛЦ 462637.007 ПЗ

МикроЭВМ. Схема электрическая структурная: ИАЛЦ 462637.004 Э1

Схема электрическая функциональная: ИАЛЦ 462637.005 Э2

Схема электрическая принципиальная: ИАЛЦ 462637.006 Э3

					ИАЛЦ.462637.002 ТЗ	Лист
						4
Змн.	Лист	№ докум.	Подпись	Дата		

№ строк	Формат	Обозначение	Наименование	Кол. листов	№ экз.	Примечание			
1			Документация общая						
2									
3			Вновь разработанная						
4			Микропроцессорная система						
5	A4	ИА/ЛЦ. 462637 002 ТЗ	Техническое задание	4	-				
6	A4	ИА/ЛЦ. 462637 003 ВП	Ведомость технического проекта	1	-				
7	A1	ИА/ЛЦ. 462637 004 Э1	МикроЭВМ. Схема						
8			электрическая структурная	1	-				
9	A3	ИА/ЛЦ. 462637 005 Э2	МикроЭВМ. Схема электрическая						
10			функциональная	1	-				
11	A1	ИА/ЛЦ. 462637 006 Э3	МикроЭВМ. Схема электрическая						
12			принципиальная	1	-				
13	A4	ИА/ЛЦ. 467449 008 ПЗ	Пояснительная записка	59	-				
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
				ИА/ЛЦ. 467449.001					
		Изм.	Лист	№ документа	Подп.	Дата			
Инв.№ подл.		Разраб.	Захожий И. А.						
		Провер.	Ткаченко В. В.						
		Реценз.							
		Н.контр.							
		Утв.	Ткаченко В. В.						
		Описание альбома				Лит.	Лист	Листов	
								1	1
						НТУУ «КПИ» ФИВТ, гр. ИВ-73			

Содержание

Раздел	Лист
Введение	2
1. Обзор существующих решений	2
2. Архитектура МПС	6
2.1 Состав устройств	6
2.2 Структурная схема МК51	7
2.2.1 Функциональное назначение выводов корпуса МК51	7
2.2.2 Арифметико-логическое устройство	8
2.2.3. Резидентная память	10
2.2.4. Устройство управления и синхронизации	15
2.3. Система команд МК51	17
2.3.1. Общие сведения о системе команд	17
2.3.2. Группа команд пересылки данных	22
2.3.3. Группа команд арифметических операций	23
2.3.4. Группа команд логических операций	24
2.3.5. Группа команд операций с битами	24
2.3.6. Группа команд передачи управления	24
2.4 Организация памяти в МК51	26
2.5. Обмен информацией	36
2.5.1. Программный режим опроса готовности ВУ	37
2.5.2. Контроллер приоритетных прерываний	39
2.5.3. Режим прямого доступа к памяти (ПДП)	44
3. Разработка ПО	49
4. Разработка функциональной схемы	57
5. Разработка принципиальной схемы	58
6. Заключение	58
7. Список литературы	59

					ИАЛЦ.463627.007 ПЗ				
Змн.	Лист	№ докум.	Подпись	Дата	Пояснительная записка	Лит.		Лист	Листов
Разраб.		Захожий И. А.						1	59
Провер.									
Реценз.									
Н. Контр.									
Утвер.		Ткаченко В. В.							
						НТУУ "КПИ" ФИВТ гр. ИВ-73			

Введение

Микроконтроллеры (МК) представляют собой автономные функционально законченные устройства, которые состоят из одной или нескольких программноуправляемых интегральных схем (ИС), которые включают в себя все средства, необходимые для обработки информации и управления данными, и рассчитаны на совместную работу с устройствами памяти и ввода-вывода информации.

Для сокращения количества типов ИС, уменьшения их стоимости и сокращения время разработки используются многофункциональные и универсальные ИС с программным управлением, которые стали основой построения микропроцессоров. Через свою универсальность МК могут производиться большими сериями, что определяет их низкую стоимость.

Создание МК привело к широкому применению универсальных вычислительных средств в те отрасли техники, где любые другие вычислительные средства, созданные ранее, не могли быть использованы. По своей структуре и функциональной организации МК аналогичны процессорам цифровых ЭВМ. Главными отличительными чертами МК есть выполнение всех их структурных единиц в виде высокоинтегрированных микросхем, небольшая длина операндов, относительно небольшой объем внутренней оперативной памяти, сохранение программ и микропрограмм в постоянной памяти, мультиплексный режим передачи информации из внешних и внутренних каналов, простая система команд.

1. Обзор существующих решений

Можно выделить следующие направления развития МК: увеличение вычислительной мощности, снижение энергопотребления, специализация под конкретный класс задач, расширение спектра применения за счет интеграции большого количества периферийных модулей.

Первое направление реализуется как за счет повышения разрядности, так и за счет совершенствования процессов изготовления, позволяющих

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		2

наращивать тактовую частоту существующего ядра. Повышение разрядности – естественная тенденция развития микропроцессоров, но в случае с микроконтроллерами она наблюдается не столь явно: в общем случае кристаллы с меньшей разрядностью дешевле, имеют меньшее энергопотребление и иногда – более развитый набор периферии за счет того, что ядро меньшей разрядности оставляет на кристалле больше места под периферийные модули. Применение 16- и 32-битных ядер в общем случае оправдано для быстрой обработки большого количества данных, например в DSP-процессорах, в то время как для задач управления по заданному алгоритму, пускай и очень большой сложности, достаточно 8-битной, а иногда даже и 4-битной архитектуры. На сегодняшний день на рынке все еще представлено немало 4-битных решений, например серия PIC10 производства Microchip.

Немаловажно и то, что под каждую существующую архитектуру накоплено огромное количество программного обеспечения, хорошо оптимизированных библиотек подпрограмм, что нередко является определяющим фактором. Допустим, что существует некое серийно выпускаемое устройство с системой управления на базе микроконтроллера с ядром 8051. Функциональных возможностей исходного микроконтроллера достаточно для работы существующего устройства, но при модернизации потребовалось значительно повысить быстродействие системы управления. Конечно, можно переработать систему под ARM-7 микроконтроллер, но это, помимо всего прочего, повлечет за собой переписывание программного обеспечения (возможно, фактическую разработку «с нуля») и длительную его отладку. В таком случае для производителя существует альтернативный вариант – можно просто подобрать более современный чип с тем же ядром, но работающий на более высокой частоте. Как пример, новое семейство микроконтроллеров Signal C8051F12х. Эти микроконтроллеры имеют «классическое» ядро 8051, но работающее на

					ИАЛЦ.463627.007 ПЗ	Лист
						3
Змн.	Лист	№ докум.	Подпись	Дата		

частоте 100 МГц. При этом флэш-память чипа работает на частоте 25 МГц, и, чтобы снизить потери времени на ожидание очередной команды, используется механизм кеширования инструкций. Такой вариант может быть гораздо менее трудоемок.

Именно по причинам функциональной достаточности 8-битных микроконтроллеров для многих задач, при невысокой себестоимости и имеющихся резервах для роста производительности, эти устройства до сих пор занимают значительную долю рынка.

Второе направление развития – снижение энергопотребления – осуществляется как бы само собой, за счет все того же совершенствования технологических процессов и усилиями разработчиков-схемотехников. Достигнутые результаты не могут не впечатлить: например, чип MC9S08QE8 производства FREESCALE потребляет в активном режиме всего 7 мкА, а в Idle-режиме \square 300 нА. Микроконтроллеры с технологией nanoWatt производства Microchip, или PicoPower производства Atmel также позволяют создавать устройства, способные питаться год и более от одной батарейки типа CR2025, разумеется, при грамотном использовании энергосберегающих технологий, основанных на переключении «на лету» тактовой частоты кристалла и переводе его в «спящий режим» на максимально возможное время.

Третье и четвертое направление развития – по сути взаимоисключающие. Фактически, специализированные микроконтроллеры – это особые версии универсальных чипов, максимально оптимизированные под один конкретный класс задач. Специализация устройств для данного класса задач делает их невыгодными для других применений, особенно с учетом того, что для удешевления ненужные для данного класса задач периферийные модули и интерфейсы в таких микроконтроллерах производителем исключается, а те, что остаются, максимально оптимизируются под конкретное применение, теряя универсальность. То

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		4

есть, если микроконтроллер будет специализирован для применения в высокочастотном преобразователе с широтно-импульсным управлением – обычный модуль таймера/счетчика с функциями сравнения/захвата/ШИМ в нем будет переработан именно под применение в режиме генератора ШИМ будут добавлены аппаратные комплементарные выходы, автоматическая генерация dead-time, защитные функции, а вот другие режимы (обычного таймера/ счетчика, захвата, сравнения) могут быть урезаны или вообще отсутствовать.

Микроконтроллеры, предназначенные для широкого круга задач, напротив, обязаны иметь максимальное количество разнообразных периферийных модулей и интерфейсов. И чем эти интерфейсы и модули разнообразнее и более гибки, тем шире круг задач, для которых можно применить данный чип, и тем удобнее его применение. При этом производители универсальных микроконтроллеров всегда предлагают целую линейку продукции, позволяющую выбрать оптимальный вариант и не переплачивать за ненужную периферию, излишний объем FLASH-памяти и т.д.

В качестве примеров третьего направления можно привести микросхемы Z8FMC16100 производства Zilog, предназначенные специально для построения схем управления бесколлекторными электродвигателями постоянного тока, ряд кристаллов производства ChipCon и Nordic Semiconductor – трансиверы на различные частоты с встроенным ядром 8051, семейство Z-Link от Atmel – ZigBee-трансиверы с встроенным AVR-микроконтроллером, микроконтроллеры LightingAVR от Atmel, имеющие специализированный ШИМ для применений в силовой электронике, или USB AVR с встроенным интерфейсом USB, что определяет их как основу для периферийных компьютерных устройств, и многих других продуктов. Их характеризует довольно узкая направленность применения каждой конкретной серии. Причем производители обычно приводят для данных

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		5

микросхем примеры их оптимального применения, что позволяет быстро сконструировать работающее устройство. А вот на микросхемах, являющихся представителями четвертого направления развития – максимально универсальных приборов, ориентированных на широкий круг задач, хотелось бы остановиться подробнее, так как именно они являются демонстрацией современных возможностей 8-битных решений. Для этого рассмотрим самые современные микроконтроллеры, предлагаемые разработчикам компанией Atmel.

В этом году компания Atmel приступила к выпуску нового семейства микроконтроллеров общего назначения – XМega AVR. Семейство XМega AVR программно совместимо с 8-битными микроконтроллерами AVR предыдущих поколений, и является новым в линейке 8-битных микроконтроллеров с ядром AVR общего применения Classic AVR (сняты с производства), Tiny AVR, Mega AVR. Сравнивать их по функциональным возможностям следует именно со старшими кристаллами из семейства MegaAVR.

2. Архитектура МПС

2.1. Состав устройств

МикроЭВМ основанная на МК1816BE51 (функциональная схема ИАЛЦ.423637.005 Э1). Данная ЭВМ может быть использована для решения математических задач и управления техническими системами.

В состав микроЭВМ входят:

- процессор
- оперативное запоминающее устройство
- внешние устройства
- постоянное запоминающее устройство
- блок питания
- интерфейс внешних устройств

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		6

- КПП
- КПДП
- ППА ВВ55

Исходными данными являются технические характеристики разрабатываемой ЭВМ:

Система команд: комплексная (Intel)

Объем внешней памяти программ : 64 Кбайт(2стр)

Объем внешней памяти данных : 2 Кбайт(12стр)

Количество внешних устройств: 80

Организация системной магистрали: с разделенными шинами адреса и данных

Контроллер прерываний: централизованный

Контроллер прямого доступа к памяти: централизованный

2.2. Структурная схема МК51

2.2.1. Функциональное назначение выводов корпуса МК51

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

Основу структурной схемы МК51 (рис. 1) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой все основные узлы и устройства: резидентную память, АЛУ, блок регистров

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		7

специальных функций, устройство управления и порты ввода/вывода. Рассмотрим основные элементы структуры и особенности организации вычислительного процесса в МК51.

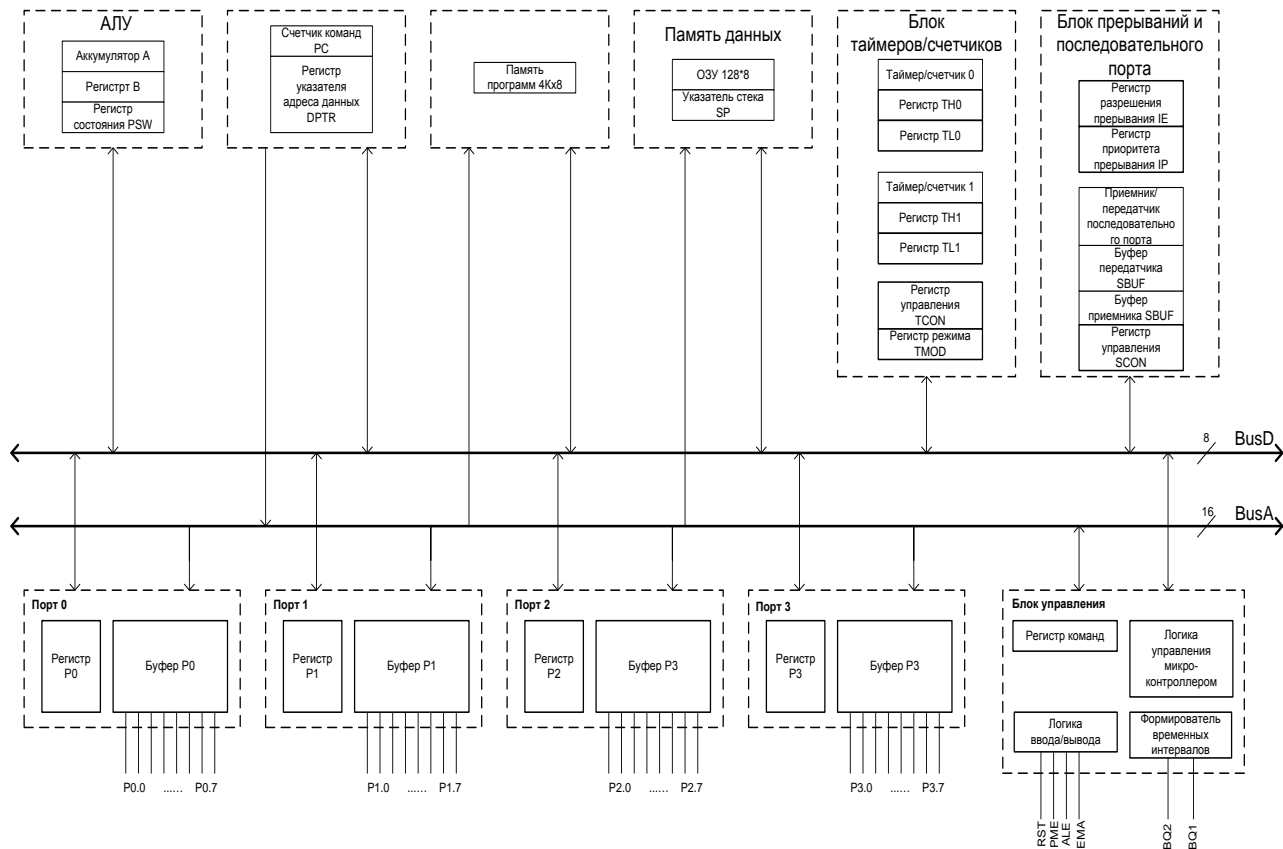


Рис. 1 Структурная схема МК51

2.2.2. Арифметико-логическое устройство

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

Простейшая операция сложения используется в АЛУ для инкрементирования содержимого регистров, продвижения регистра-указателя данных и автоматического вычисления следующего адреса РПП.

Простейшая операция вычитания используется в АЛУ для декрементирования регистров и сравнения переменных.

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

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

Таким образом, АЛУ может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными

(16 бит). В АЛУ выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования "операция/режим адресации" базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		9

2.2.3. Резидентная память

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

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

Память данных (ОЗУ) предназначена для хранения переменных в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к адресному пространству РПД примыкают адреса регистров специальных функций (РСФ), которые перечислены в табл. 1.

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

Аккумулятор и ССП. Аккумулятор является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п.

					ИАЛЦ.463627.007 ПЗ	Лист
						10
Змн.	Лист	№ докум.	Подпись	Дата		

Таблица 1. Блок регистров специальных функций

<i>Символ</i>	<i>Наименование</i>	<i>Адрес</i>
<i>* ACC</i>	<i>Аккумулятор</i>	<i>0E0H</i>
<i>* B</i>	<i>Регистр-расширитель аккумулятора</i>	<i>0F0H</i>
<i>* PSW</i>	<i>Слово состояния программы</i>	<i>0D0H</i>
<i>SP</i>	<i>Регистр-указатель стека</i>	<i>81H</i>
<i>DPTR</i>	<i>Регистр-указатель данных (DPH)</i>	<i>83H</i>
	<i>(DPL)</i>	<i>82H</i>
<i>* P0</i>	<i>Порт 0</i>	<i>80H</i>
<i>* P1</i>	<i>Порт 1</i>	<i>90H</i>
<i>* P2</i>	<i>Порт 2</i>	<i>0A0H</i>
<i>* P3</i>	<i>Порт 3</i>	<i>0B0H</i>
<i>* IP</i>	<i>Регистр приоритетов</i>	<i>0B8H</i>
<i>* IE</i>	<i>Регистр маски прерываний</i>	<i>0A8H</i>
<i>TMOD</i>	<i>Регистр режима таймера/счетчика</i>	<i>89H</i>
<i>* TCON</i>	<i>Регистр управления/статус таймера</i>	<i>88H</i>
<i>TH0</i>	<i>Таймер 0 (старший байт)</i>	<i>8CH</i>
<i>TL0</i>	<i>Таймер 0 (младший байт)</i>	<i>8AH</i>
<i>TH1</i>	<i>Таймер 1 (старший байт)</i>	<i>8DH</i>

<i>TL1</i>	<i>Таймер 1 (младший байт)</i>	<i>8BH</i>
<i>* SCON</i>	<i>Регистр управления приемопередатчиком</i>	<i>98H</i>
<i>SBUF</i>	<i>Буфер приемопередатчика</i>	<i>99H</i>
<i>PCON</i>	<i>Регистр управления мощностью</i>	<i>87H</i>
Примечание. Регистры, имена которых отмечены знаком (*), допускают адресацию отдельных бит.		

При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре ССП. В [табл.2](#) приводится перечень флагов ССП, даются их символические имена и описываются условия их формирования.

Таблица 2. Формат слова состояния программы (ССП)

<i>Символ</i>	<i>Позиция</i>	<i>Имя и назначение</i>
<i>C</i>	<i>PSW.7</i>	<i>Флаг переноса. Устанавливается и сбрасывается аппаратурными средствами или программой при выполнении арифметических и логических операций</i>
<i>AC</i>	<i>PSW.6</i>	<i>Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратурными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заем в бите 3</i>
<i>FO</i>	<i>PSW.5</i>	<i>Флаг 0. Может быть установлен, сброшен</i>

		или проверен программой как флаг, специфицируемый пользователем			
RS1	PSW.4	Выбор банка регистров. Устанавливается и сбрасывается программой для выбора рабочего банка регистров (см. примечание)			
RS0	PSW.3				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций			
—	PSW.1	Не используется			
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратуры в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе, т.е. выполняет контроль по четности			
Примечание.		RS1	RS0	Банк	Границы адресов
		0	0	0	00H-07H
		0	1	1	08H-0FH
		1	0	2	10H-17H
		1	1	3	18H-1FH

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

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

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

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

Двухбайтный регистр-указатель данных (PUD) обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней

					ИАЛЦ.463627.007 ПЗ	Лист
						14
Змн.	Лист	№ докум.	Подпись	Дата		

памяти. Командами МК51 регистр-указатель данных может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

Таймер/счетчик. В составе средств МК51 имеются регистровые пары с символическими именами TH0, TLO и TH1, TLI, на основе которых функционируют два независимых программно-управляемых 16-битных таймера/счетчика событий.

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

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

2.2.4. Устройство управления и синхронизации

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

					ИАЛЦ.463627.007 ПЗ	Лист
						15
Змн.	Лист	№ докум.	Подпись	Дата		

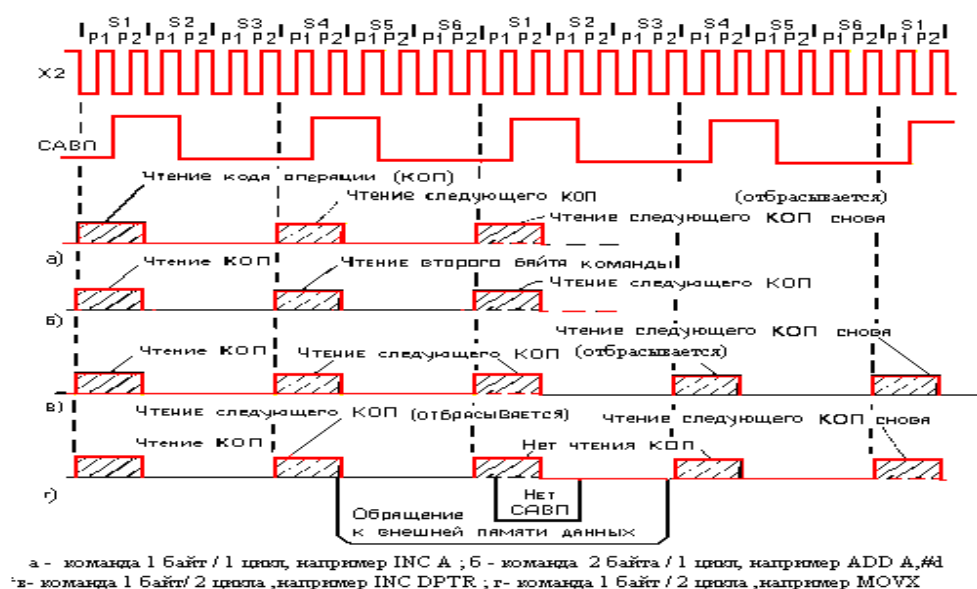


Рис. 2. Последовательности выборки и выполнения команд в МК51:
а - команда 1 байт/1 цикл, например INC A; б - команда - 2 байта/1 цикл, например ADD A, #d; в - команда 1 байт/2 цикла, например INC DPTR; г - команда 1 байт/2 цикла, например MOVX.

Устройство управления МК51 на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора или шести состояниям первичного управляющего автомата (S1-S6). Каждое состояние управляющего автомата содержит две фазы (P1, P2) сигналов резонатора. В фазе P1, как правило, выполняется операция в АЛУ, а в фазе P2 осуществляется межрегистровая передача. Весь машинный цикл состоит из 12 фаз, начиная с фазы S1P1 и кончая фазой S6P2, как показано на рис. 2. Эта временная диаграмма иллюстрирует работу устройства управления МК51 при выборке и исполнении команд различной степени сложности. Все заштрихованные сигналы являются внутренними и недоступны пользователю МК51 для контроля. Внешними, наблюдаемыми сигналами являются только сигналы резонатора и stroba адреса внешней памяти. Как видно из временной диаграммы, сигнал СВВП формируется дважды за один машинный цикл

(S1P2-S2P1 и S4P2-S5P1) и используется для управления процессом обращения к внешней памяти.

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

2.3. Система команд МК51

2.3.1. Общие сведения о системе команд

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

Система команд МК51 много мощнее и шире системы команд МК48, так как кроме всех команд МК48 в ее состав входят команды умножения, деления, вычитания, операций над битами, операций со стеком и расширенный набор команд передачи управления. Большинство команд (94) имеют формат один или два байта и выполняются за один или два машинных цикла. При тактовой частоте 2 МГц длительность машинного цикла составляет 1 мкс. На рис. 3 показаны 13 типов команд МК51. Первый байт команды любых типа и формата всегда содержит код операции (КОП). Второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

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

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		17

	D7 D0		
1	КОП		
		D7 D0	
2	КОП	#d	
3	КОП	ad	
4	КОП	bit	
5	КОП	rel	
6	a10a9a8 КОП	a7 a0	D7 D0
7	КОП	ad	#d
8	КОП	ad	rel
9	КОП	ads	add
10	КОП	#d	rel
11	КОП	bit	rel
12	КОП	ad16h	ad16l
13	КОП	#d16h	#d16l

Рис. 3 Типы команд МК51

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

Способы адресации данных. В МК51 используются такие же способы адресации данных, как и в МК48: прямая, непосредственная, косвенная и

неявная. Следует отметить, что при косвенном способе адресации РПД в отличие от МК48 используются все восемь бит адресных регистров R0 и R1.

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

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

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

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

Флаг C устанавливается, если в старшем бите результата возникает перенос или заем. При выполнении операций умножения и деления флаг C сбрасывается. Флаг OV (отсутствует в МК48) устанавливается, если результат операции сложения/вычитания не укладывается в семи битах и старший (восьмой) бит результата не может интерпретироваться как знаковый. При выполнении операции деления флаг OV сбрасывается, а в случае деления на нуль устанавливается. При умножении флаг OV устанавливается, если результат больше 255.

В табл. 3 перечисляются команды, при выполнении которых модифицируются флаги результата. В таблице отсутствует флаг паритета, так как его значение изменяется всеми командами, которые изменяют содержимое аккумулятора. Кроме команд, приведенных в таблице,

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		19

флаги модифицируются командами, в которых местом назначения результата определены PSW или его отдельные биты, а также командами операций над битами.

Таблица 3 Команды, модифицирующие флаги результата

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

Символическая адресация. При использовании ассемблера МК51 (ASM51) для получения объектных кодов программ допускается применение в программах символических имен регистров специальных функций (РСФ), портов и их отдельных бит (рис 5).

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

< имя РСФ или порта > . < номер бита >

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

Адреса РПД	(D7) (D0)							
7FH								
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00
1FH								
18H	Банк 3							
17H								
10H	Банк 2							
0FH								
08H	Банк 1							
07H								
00H	Банк 0							

Рис .3.20. Карта адресуемых бит в резидентной памяти данных

Рис. 4. Карта адресуемых бит

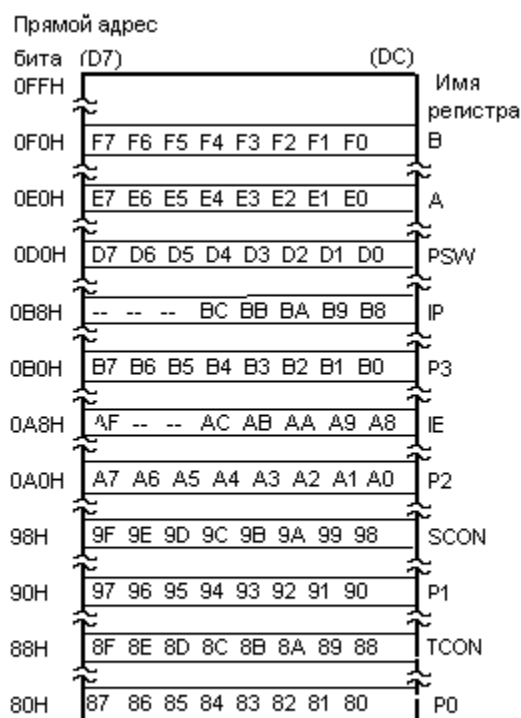


Рис.3.21. Карта адресуемых бит в блоке регистров специальных функций

Рис. 5. Карта адресуемых бит в резидентной памяти данных блоке регистров специальных функций

2.3.2. Группа команд пересылки данных

Большую часть команд данной группы составляют команды передачи и обмена байтов. Команды пересылки бит представлены в группе команд битовых операций. Все команды данной группы не модифицируют флаги результата, за исключением команд загрузки PSW и аккумулятора (флаг паритета).

Структура информационных связей. В зависимости от способа адресации и места расположения операнда можно выделить девять типов операндов, между которыми возможен информационный обмен.

В отличие от МК48 передачи данных в МК51 могут выполняться без участия аккумулятора.

Аккумулятор. В отличие от МК48 обращение к аккумулятору может быть выполнено в МК51 с использованием неявной и прямой адресации. В

зависимости от способа адресации аккумулятора применяется одно из символических имен: А или АСС (прямой адрес). При прямой адресации обращение к аккумулятору производится как к одному из РСФ, и его адрес указывается во втором байте команды.

Использование неявной адресации аккумулятора предпочтительнее, однако не всегда возможно, например при обращении к отдельным битам аккумулятора.

Обращение к внешней памяти данных. Режим косвенной адресации ВПД, имеющийся в МК48, реализован также и в МК51. При использовании команд MOVX @Ri обеспечивается доступ к 256 байтам внешней памяти данных.

Существует также режим обращения к расширенной ВПД, когда для доступа используется 16-битный адрес, хранящийся в регистре-указателе данных (DPTR). Команды MOVX @DPTR обеспечивают доступ к 65 536 байтам ВПД.

2.3.3. Группа команд арифметических операций

Данную группу образуют 24 команды, выполняющие операции сложения, десятичной коррекции, инкремента/декремента байтов. Дополнительно по сравнению с МК48 введены команды вычитания, умножения и деления байтов.

Команды ADD и ADDC аналогичны командам сложения МК48, но допускают сложение аккумулятора с большим числом операндов. Аналогично командам ADDC существуют четыре команды SUBB, что позволяет более просто, чем в МК48, производить вычитание байтов и многобайтных двоичных чисел. В МК51 реализуется расширенный (по сравнению с МК48) список команд инкремента/декремента байтов, введена команда инкремента 16-битного регистра-указателя данных.

					ИАЛЦ.463627.007 ПЗ	Лист
						23
Змн.	Лист	№ докум.	Подпись	Дата		

2.3.4. Группа команд логических операций

Данную группу образуют 25 команд реализующих те же логические операции над байтами, что и в МК48. Однако в МК51 значительно расширено число типов операндов, участвующих в операциях.

В отличие от МК48 имеется возможность производить операцию "исключающее ИЛИ" с содержимым портов. Команда XRL может быть эффективно использована для инверсии отдельных бит портов.

2.3.5. Группа команд операций с битами

Отличительной особенностью данной группы команд является то, что они оперируют с однобитными операндами. В качестве таких операндов могут выступать отдельные биты некоторых регистров специальных функций (РСФ) и портов, а также 128 программных флагов пользователя.

Существуют команды сброса (CLR), установки (SETB) и инверсии (CPL) бита, а также конъюнкции и дизъюнкции бита и флага переноса. Для адресации бита используется прямой восьмиразрядный адрес (bit). Косвенная адресация бита невозможна.

2.3.6. Группа команд передачи управления

К данной группе команд относятся команды, обеспечивающие условное и безусловное ветвление, вызов подпрограмм и возврат из них, а также команда пустой операции NOP. В большинстве команд используется прямая адресация, т.е. адрес перехода целиком (или его часть) содержится в самой команде передачи управления. Можно выделить три разновидности команд ветвления по разрядности указываемого адреса перехода.

Длинный переход. Переход по всему адресному пространству ПП. В команде содержится полный 16-битный адрес перехода (ad 16). Трех байтные команды длинного перехода содержат в мнемокоде букву L (Long). Всего существует две такие команды: LJMP – длинный переход и LCALL – длинный вызов подпрограммы. На практике редко возникает необходимость

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		24

перехода в пределах всего адресного пространства и чаще используются укороченные команды перехода, занимающее меньше места в памяти.

Абсолютный переход. Переход в пределах одной страницы памяти программ размером 2048 байт. Такие команды содержат только 11 младших бит адреса перехода (ad 11). Команды абсолютного перехода имеют формат 2 байта. Начальная буква мнемокода - А (Absolute). При выполнении команды в вычисленном адресе следующей по порядку команды $((PC) = (PC) + 2)$ 11 младших бит заменяются на ad11 из тела команды абсолютного перехода.

Относительный переход. Короткий относительный переход позволяет передать управление в пределах -128 - +127 байт относительно адреса следующей команды (команды, следующей по порядку за командой относительного перехода). Существует одна команда безусловного короткого перехода SJMP (Short). Все команды условного перехода используют данный метод адресации. Относительный адрес перехода (rel) содержится во втором байте команды.

Косвенный переход. Команда JMP @A + DPTR позволяет передавать управление по косвенному адресу. Эта команда удобна тем, что предоставляет возможность организации перехода по адресу, вычисляемому самой программой и неизвестному при написании исходного текста программы.

Условные переходы. Развитая система условных переходов предоставляет возможность осуществлять ветвление по следующим условиям: аккумулятор содержит нуль (JZ); содержимое аккумулятора не равно нулю (JNZ); перенос равен единице (JC); перенос равен нулю (JNC); адресуемый бит равен единице (JB); адресуемый бит равен нулю (JNB).

Для организации программных циклов удобно пользоваться командой DJNZ, которая работает аналогично соответствующей команде МК48.

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		25

Однако в качестве счетчика циклов в МК51 может использоваться не только регистр, но и прямоадресуемый байт (например, ячейка РПД).

Команда *CJNE* эффективно используется в процедурах ожидания какого-либо события. Например, команда

WAIT: CJNE A,P0,WAIT

будет выполняться до тех пор, пока на линиях порта *P0* не установится информация, совпадающая с содержимым аккумулятора.

Все команды данной группы, за исключением *CJNE* и *JBC*, не оказывают воздействия на флаги. Команда *CJNE* устанавливает флаг *C*, если первый операнд оказывается меньше второго. Команда *JBC* сбрасывает флаг *C* в случае перехода.

Подпрограммы. Для обращения к подпрограммам необходимо использовать команды вызова подпрограмм (*LCALL*, *ACALL*). Эти команды в отличие от команд перехода (*LJMP*, *AJMP*) сохраняют в стеке адрес возврата в основную программу. Для возврата из подпрограммы необходимо выполнить команду *RET*. Команда *RETl* отличается от команды *RET* тем, что разрешает прерывания обслуженного уровня.

2.4. Организация памяти в МК51

В архитектуре МК51 память данных и память программ разделены. Каждая из них может иметь размер до 64Кбайта, выбор одной из двух матриц памяти осуществляется сигналами *PME*, *WR*, *RD*. Организация памяти в микроконтроллерах семейства МК51 иллюстрируется на Рис. 6.

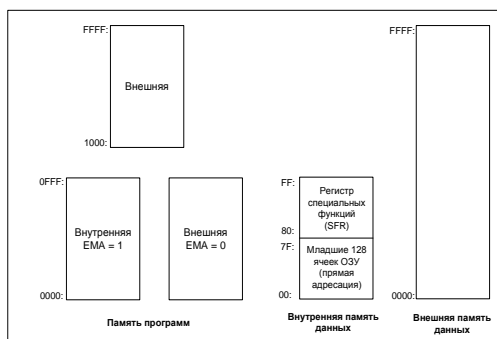


Рисунок 6. Организация памяти в архитектуре МК51

Память программ

Максимальное адресное пространство памяти программ составляет 64Кбайт. Память программ разделяется на резидентную (находящуюся внутри микросхемы) и внешнюю, для реализации которой требуются дополнительные микросхемы памяти. РПП представляет собой ПЗУ, емкостью 4Кбайт (адреса от 0 до 0FFFFh). РПП может быть отключена путем подачи низкого уровня на вход ЕМА. Заметим что в МК 1816ВЕ31 и 1830ВЕ31 внутренняя память программ отсутствует. Адреса 0, 3, 0Bh, 13h, 1Bh и 23h имеют специальное назначение:

00h – начальный адрес пуска;

03h – вектор внешнего прерывания INT0;

0Bh – вектор прерывания от таймера T/C0;

13h – вектор внешнего прерывания INT1;

1Bh – вектор прерывания от таймера T/C1;

23h – вектор прерывания от последовательного интерфейса.

Подключение внешней памяти программ показано на Рис. 7.

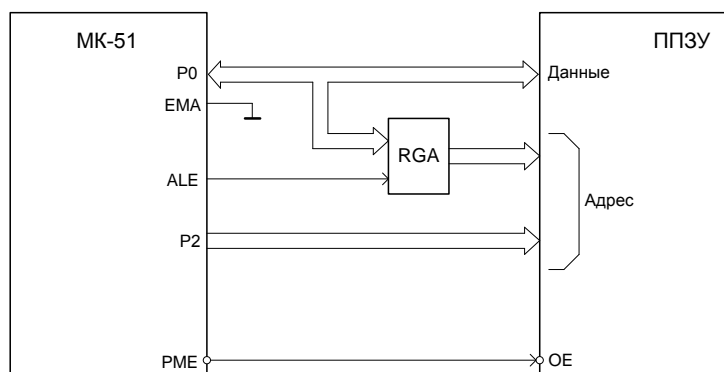


Рисунок 7. Схема включения МК-51 с внешней ППЗУ программ

При обращениях к внешней памяти программ всегда формируется 16-разрядный адрес, младший байт которого выдается через порт P0, а старший – через порт P2. При этом байт адреса, выдаваемый через порт P0, должен быть зафиксирован во внешнем регистре по спаду сигнала ALE, т. к. в дальнейшем линии порта P0 используются в качестве шины данных,

по которой байт из внешней памяти программ вводится в микроконтроллер.

На Рис. 5 показана функциональная схема включения МК51 с внешним ППЗУ программ. Порт P0 работает как мультиплексированная шина адрес/данные: выдает младший байт счетчика команд, а затем переходит в высокоимпедансное состояние и ожидает прихода байта из ППЗУ программ. Когда младший байт адреса находится на выходах порта P0, сигнал $\sim ALE$ защелкивает его в адресном регистре RG. Старший байт адреса находится на выходах порта P2 в течение всего времени обращения к ППЗУ. Сигнал $\sim PWE$ разрешает выборку байта из ППЗУ, после чего выбранный байт поступает на порт P0 МК51 и вводится в ОМЭВМ.

Адресация в памяти программ – непосредственная или косвенная базовая индексная. В первом случае из памяти программ выбирается константа, явно заданная в команде. Например, при выполнении инструкции `MOV R2, #15` в регистр пересылается константа 15. Во втором случае в качестве индексного регистра используется аккумулятор, а в качестве базового – регистр-указатель данных DPTR или счетчик команд PC. Чтение операндов выполняется командами `MOVC`.

`MOVC A, @A+DPTR` – Пересылка в аккумулятор байта из памяти программ $(A) \leftarrow ((A) + (DPTR))$.

`MOVC A, @A+PC` – Пересылка в аккумулятор байта из памяти программ $(PC) \leftarrow (PC) + 1$, $(A) \leftarrow ((A) + (PC))$.

Память данных

Память данных предназначена для приема, хранения и выдачи информации, используемой в процессе выполнения программы. Память данных, расположенная на кристалле ОМЭВМ, состоит из регистра адреса ОЗУ, дешифратора, ОЗУ и указателя стека.

					ИАЛЦ.463627.007 ПЗ	Лист
						28
Змн.	Лист	№ докум.	Подпись	Дата		

Регистр адреса ОЗУ предназначен для приема и хранения адреса выбираемой с помощью дешифратора ячейки памяти, которая может содержать как бит, так и байт информации.

ОЗУ представляет собой 128 восьмиразрядных регистров, предназначенных для приема, хранения и выдачи различной информации.

Указатель стека представляет собой восьмиразрядный регистр, предназначенный для приема и хранения адреса ячейки стека, к которой было последнее обращение. При выполнении команд LCALL, ACALL содержимое указателя стека увеличивается на 2. При выполнении команд RET, RETI содержимое указателя стека уменьшается на 2. При выполнении команды PUSH direct содержимое указателя стека увеличивается на 1. При выполнении команды POP direct содержимое указателя стека уменьшается на 1. После сброса в указателе стека устанавливается адрес 07H, что соответствует началу стека с адресом 08H.

Внутренняя память данных ОМЭВМ состоит из двух областей: оперативной памяти (ОЗУ) с адресами 0-7FH и области регистров специальных функций, занимающей адреса 80H-FFH. Распределение пространства внутренней памяти данных показано на Рис.5 - 4.

Физически внутреннее ОЗУ данных и область регистров специальных функций являются отдельными устройствами. Все ячейки внутреннего ОЗУ данных могут адресоваться с использованием прямой и косвенной адресации (режимы адресации описаны в описании системы команд ОМЭВМ). Кроме того, внутреннее ОЗУ данных имеет следующие особенности. Младшие 32 байта внутреннего ОЗУ данных сгруппированы в 4 банка по 8 регистров в каждом (БАНКО-БАНКЗ на Рис. 6). Команды программы могут обращаться к регистрам, используя их имена R0-R7. Два бита PSW (указатели банка рабочих регистров RS0 и RS1) определяют, с регистрами какого банка производятся манипуляции. Наличие такого механизма работы с ячейками ОЗУ позволяет экономить память программ, т. к.

команды, работающие с регистрами R0-R7, короче команд, использующих прямую адресацию. Следующие после банков регистров внутреннего ОЗУ данных 16 байт (адрес 20H-2FH) образуют область ячеек, к которым возможна побитовая адресация. Набор команд ОМЭВМ семейства МК51 содержит значительное количество инструкций, позволяющих работать с отдельными битами, используя при этом прямую адресацию. 128 бит, составляющих рассматриваемую область внутреннего ОЗУ данных, имеют адреса 00H-7FH и предназначены для работы с такими инструкциями. 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 микросхемам памяти, подключаемыми к ОМЭВМ 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 ОМЭВМ. Таким образом, в системе могут одновременно присутствовать внутренняя память данных с адресами 00H-FFH и внешняя память данных с адресами 0000H-FFFFH.

Обращение к ячейкам внешней памяти данных осуществляется только с использованием косвенной адресации по регистрам R0 и R1 активного банка регистров внутреннего ОЗУ (команды типа MOV @Ri) или по регистру специальных функций DPTR (команды типа MOV @DPTR). Соответственно в первом случае будет формироваться 8-разрядный, а во втором случае 16-разрядный адреса внешней памяти данных.

При обращениях к внешней памяти данных адрес выводится через порт P0 (младший байт) и порт P2 (старший байт) ОМЭВМ. Обмен байтом данных (запись и чтение) производится через порт P0 ОМЭВМ, т. е. порт P0 используется как шина адреса/данных в режиме мультиплексирования.

					ИАЛЦ.463627.007 ПЗ	Лист
						30
Змн.	Лист	№ докум.	Подпись	Дата		

Считывание данных из внешней памяти данных в ОМЭВМ производится с помощью выходного сигнала ОМЭВМ $\sim RD$, а запись данных из ОМЭВМ во внешнюю память данных с помощью выходного сигнала ОМЭВМ $\sim WR$.

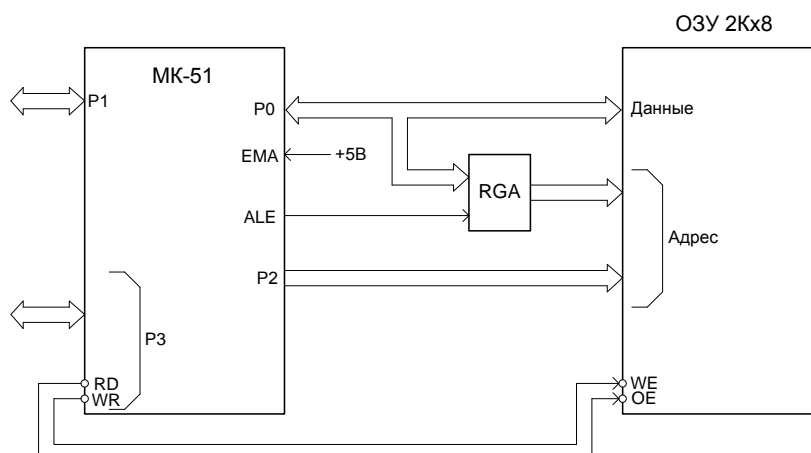


Рисунок 8. Страничная организация внешней памяти данных

На Рис. 8 показана страничная организация внешней памяти данных. Приведенная схема позволяет работать с памятью данных емкостью 2 Кбайт, используя команды типа $MOVX @Ri$. Порт P0 при этом работает как мультиплексированная шина адрес/данные, а три линии порта P2 адресуют страницы внешнего ОЗУ. Остальные 5 линий порта P2 могут использоваться в качестве линий ввода/вывода.

Протоколы обмена микроконтроллеров семейства МК-51 по магистралям реализованы в виде комбинаций следующих трех типов магистральных циклов.

- ввод кода команды из внешней памяти программ без обращения к внешней памяти данных;*
- чтение из внешней памяти данных;*
- запись во внешнюю паять данных.*

Доступ к ВПП осуществляется при помощи управляющего сигнала $\sim RVPP$, который выполняет функцию строб-сигнала чтения. Доступ к ВПД

обеспечивается управляющими сигналами $\overline{ЧТ}$ и $\overline{ЗП}$, которые формируются в линиях P3.7 и P3.6 при выполнении портом 3 альтернативных функций.

При обращении к ВПП всегда используется 16-битный адрес. Доступ к ВПД возможен с использованием 16-битного адреса ($MOVX A, @DPTR$) или 8-битного адреса ($MOVX A, @Ri$).

В любых случаях использования 16-битного адреса старший байт адреса фиксируется (и сохраняется неизменным в течение одного цикла записи или чтения) в регистре-защелке порта 2.

Если очередной цикл внешней памяти ($MOVX A, @DPTR$) следует не сразу же за предыдущим циклом внешней памяти, то неизменяемое содержимое регистра-защелки порта 2 восстанавливается в следующем цикле. Если используется 8-битный адрес ($MOVX A, @Ri$), то содержимое регистра-защелки порта 2 остается неизменным на его внешних выводах в течение всего цикла внешней памяти.

Через порт 0 в режиме временного мультиплексирования осуществляется выдача младшего байта адреса и передача байта данных. Сигнал $\overline{СABП}$ должен быть использован для записи байта адреса во внешний регистр. Затем в цикле записи выводимый байт данных появляется на внешних выводах порта 0 только перед появлением сигнала $\overline{ЗП}$. В цикле чтения вводимый байт данных принимается в порт 0 по фронту стробирующего сигнала $\overline{ЧТ}$.

При любом обращении к внешней памяти устройство управления МК51 загружает в регистр-защелку порта 0 код 0FFH, стирая тем самым информацию, которая могла в нем храниться.

Доступ к ВПП возможен при выполнении двух условий: либо на вход отключения резидентной памяти программ ($\overline{ОРПП}$) подается активный сигнал, либо содержимое счетчика команд превышает значение 0FFFH. Наличие сигнала $\overline{ОРПП}$ необходимо для обеспечения доступа к младшим 4К

адресам адресного пространства ВПП при использовании МК31 (микроконтроллера без резидентной памяти программ).

Временные диаграммы на рис.9 иллюстрируют процесс генерации управляющих сигналов САВП и РВПП при обращении к внешней памяти.

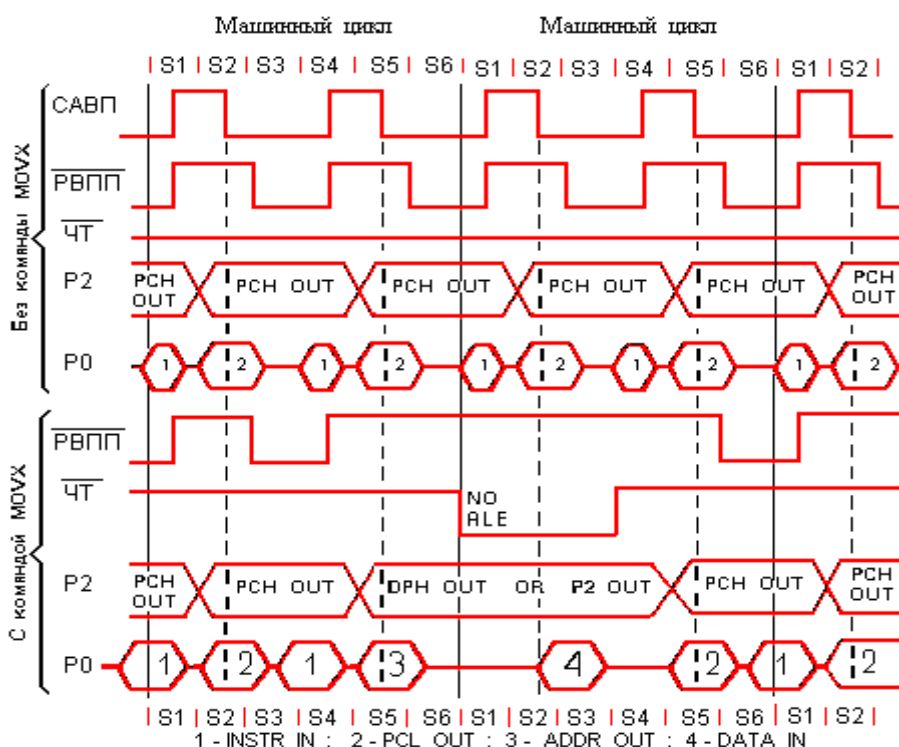


Рис. 9. Временные диаграммы операций с обращением к внешней памяти

Основная функция сигнала САВП — обеспечить временное согласование передачи из порта 0 на внешний регистр младшего байта адреса в цикле чтения из ВПП. Сигнал САВП приобретает значение 1 дважды в каждом машинном цикле. Это происходит даже тогда, когда в цикле выборки нет обращения к ВПП. Доступ к ВПД возможен только в том случае, если сигнал САВП отсутствует. Первый сигнал САВП во втором машинном цикле команды MOVX блокируется. Следовательно, в любой МК-системе, не использующей ВПД, сигнал САВП генерируется с постоянной частотой, равной 1/16 частоты резонатора, и может быть использован для синхронизации внешних устройств или для реализации различных временных функций.

При обращении к РПП сигнал РВПП не генерируется, а при обращении к ВПП он выполняет функцию строб-сигнала чтения. Полный цикл чтения ВПД, включая установку и снятие сигнала ЧТ, занимает 12 периодов резонатора.

Временные диаграммы на рис. 10 и рис. 11 иллюстрируют процесс выборки команды из ВПП и работу с ВПД в режимах чтения и записи соответственно.

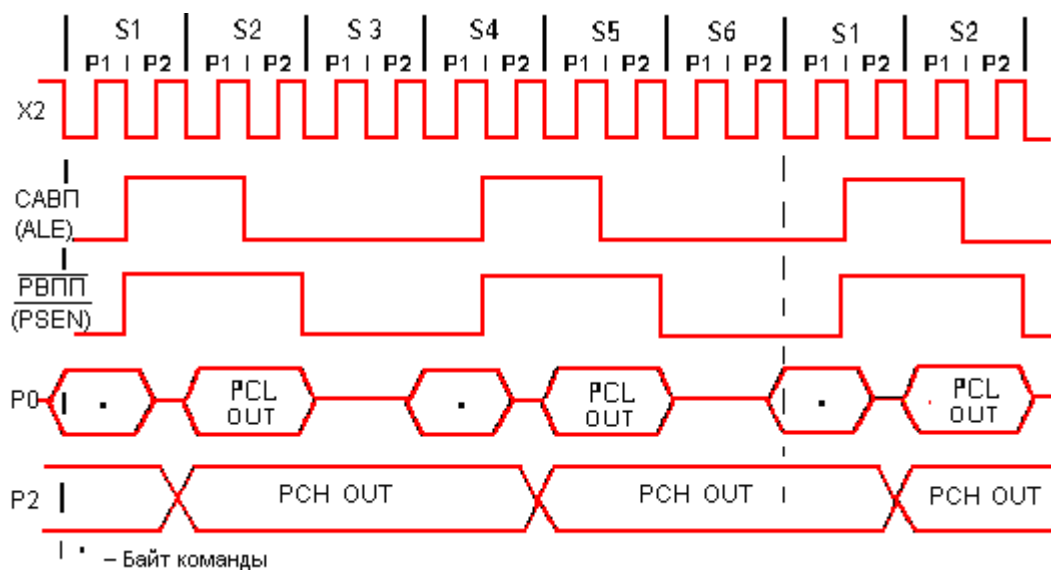


Рис. 10. Временная диаграмма выборки команды из ВПП

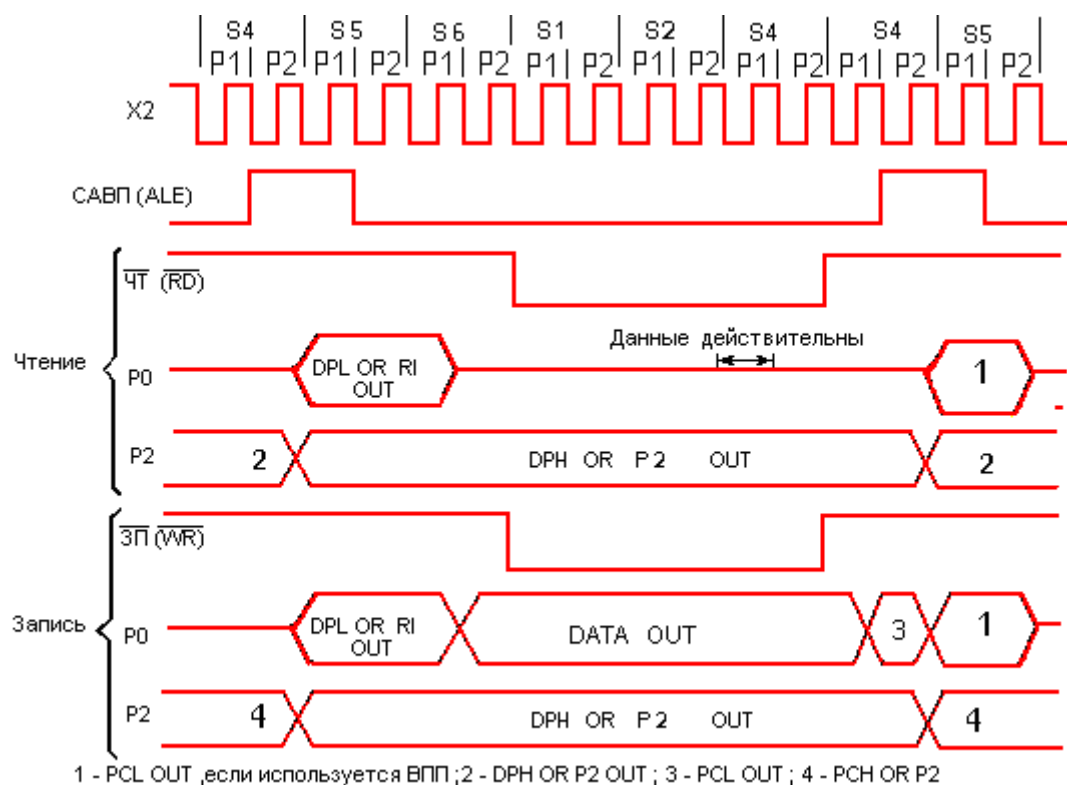


Рис. 11. Временная диаграмма работы с ВДП

Особый режим работы МК51. Содержимое памяти программ МК51 заполняется единожды на этапе разработки МК-системы и не может быть модифицировано в завершенном (конечном) изделии. По этой причине микроконтроллеры не являются машинами классической "фон-неймановской" архитектуры. Оперативная память данных (резидентная или внешняя) не может быть использована для хранения кодов программы, так как в МК выборка команд производится только из области адресов памяти программ. Эта особенность архитектуры МК объясняется тем, что в большинстве применений МК требуется наличие одной неизменяемой прикладной программы, хранимой в ПЗУ, наличие ОЗУ небольшой емкости для временного хранения переменных и эффективных, а следовательно, разных методов адресации памяти программ и памяти данных.

Однако на этапе разработки и отладки прикладных программ машина "фон-неймановского" типа оказывается очень удобной, так как позволяет

разработчику оперативно изменять коды прикладной программы, размещаемой в ОЗУ. С этой целью МК-система может быть модифицирована для совмещения адресного пространства ВПП и ВПД путем подключения внешней логики, как показано на рис. 12.

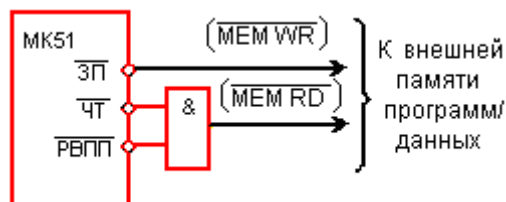


Рис. 12. Схема совмещения адресного пространства ВПП и ВПД

Здесь на выходе схемы И формируется строб-сигнал чтения, который может быть использован для объединения памяти программ и памяти данных во внешнем ОЗУ. При этом необходимо учитывать, что в МК51 на схемном уровне реализуются пять различных и независимых механизмов адресации для доступа к ПРР, РПД, ВПП, ВПД и блоку регистров специальных функций. Вследствие этого перемещаемая версия прикладной программы, которая отлаживается в среде внешней памяти программ/данных, будет отличаться от загружаемой в РП (окончательной) версии программы.

Подобный способ организации управления внешней памятью может быть использован в тех применениях МК51, где требуется оперативная перезагрузка или модификация прикладных программ (с помощью ЧВВ), как в ЭВМ классической архитектуры.

2.5. Обмен информацией

Концепция вычислительной техники предполагает обработку и хранение информации. Для обеспечения этих функций необходим обмен информацией между элементами ЭВМ. То есть необходимо обеспечить

взаимодействие между процессором, оперативной памятью и внешними устройствами.

Можно выделить три основных режима работы процессора с внешними устройствами:

- 1) программный режим опроса готовности ВУ (полинг),*
- 2) режим прямого доступа к общей памяти со стороны ВУ,*
- 3) режим прерывания программы.*

2.5.1. Программный режим опроса готовности ВУ

В данном режиме инициатором обмена является процессор. Для синхронизации используется бит готовности в регистре (порте) внешнего устройства. Этот бит устанавливается контроллером внешнего устройства, когда оно готово к обмену и сбрасывается при обращении к регистру (порту) данных.

Для обслуживания нескольких ВУ используется программный полинг, то есть опрос ВУ в соответствии с рангом их приоритетов. Полинг оформляется в виде подпрограммы. Возможен случай, когда сама программа процессора является полингом.

На рисунке 13 приведен алгоритм программы полинга для 56 ВУ (в соответствии с заданием). Приоритет ВУ уменьшается с увеличением их номеров.

					ИАЛЦ.463627.007 ПЗ	Лист
						37
Змн.	Лист	№ докум.	Подпись	Дата		

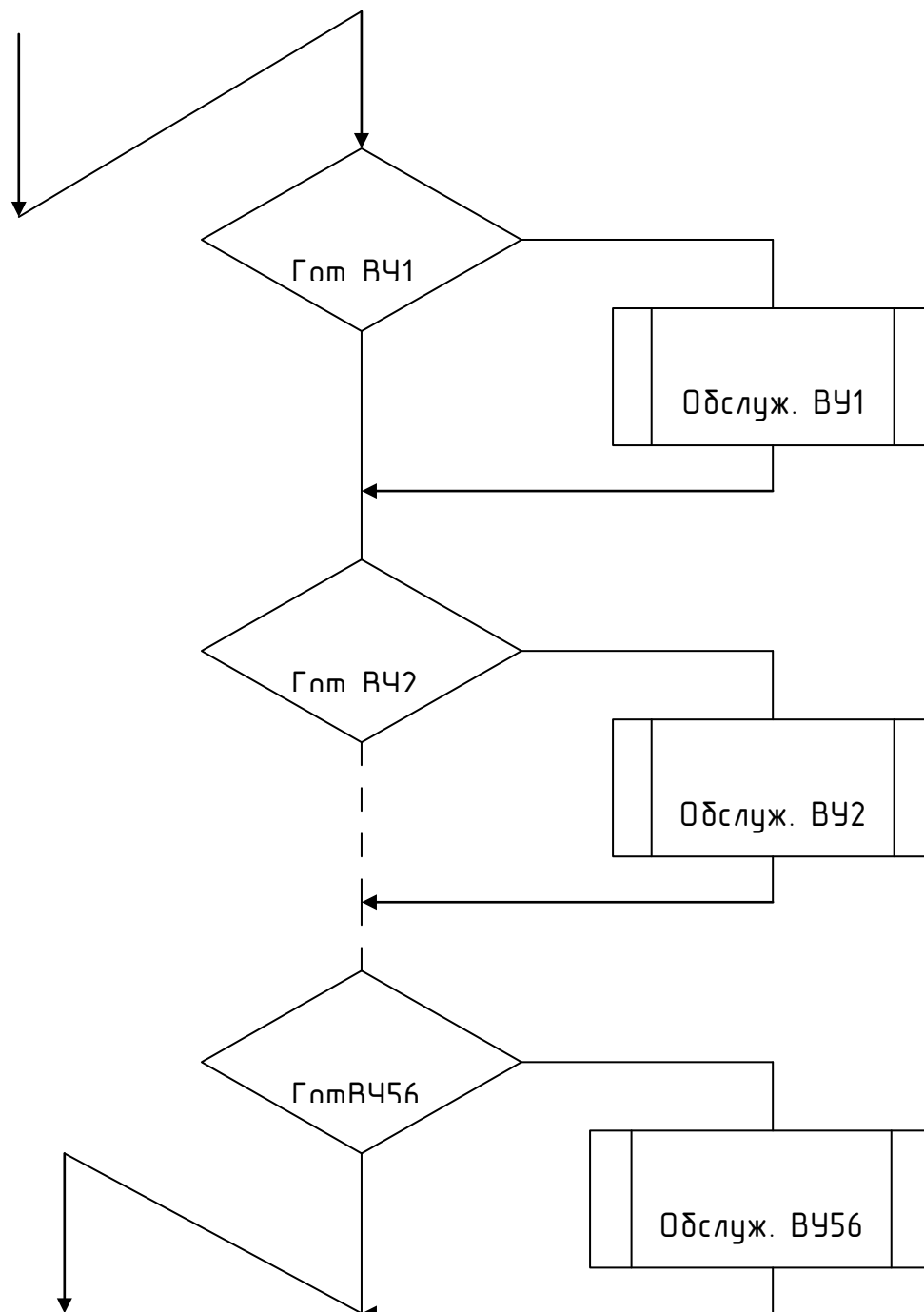


Рис. 13. Алгоритм полинга

Режим опроса готовности имеет преимущества и недостатки по сравнению с другими режимами.

Преимущества:

- 1) простота реализации интерфейсов ВУ,
- 2) в процессе выполнения программы можно менять приоритеты ВУ.

Недостатки:

- 1) снижение производительности за счет непроизводительного расхода команд процессора на опрос ВУ,
- 2) трудно предусмотреть аварийное или экстренное обслуживание некоторого ВУ.

2.5.2. Контроллер приоритетных прерываний

На структурной схеме ИАЛЦ.423637.003 Э1 использовался централизованный контроллер приоритетных прерываний (КПП).

Режим прерываний предполагает временный останов основной программы и переход на программу обработки прерываний с последующим возвратом в основную программу. Классификация прерываний приведена на рисунке 14.

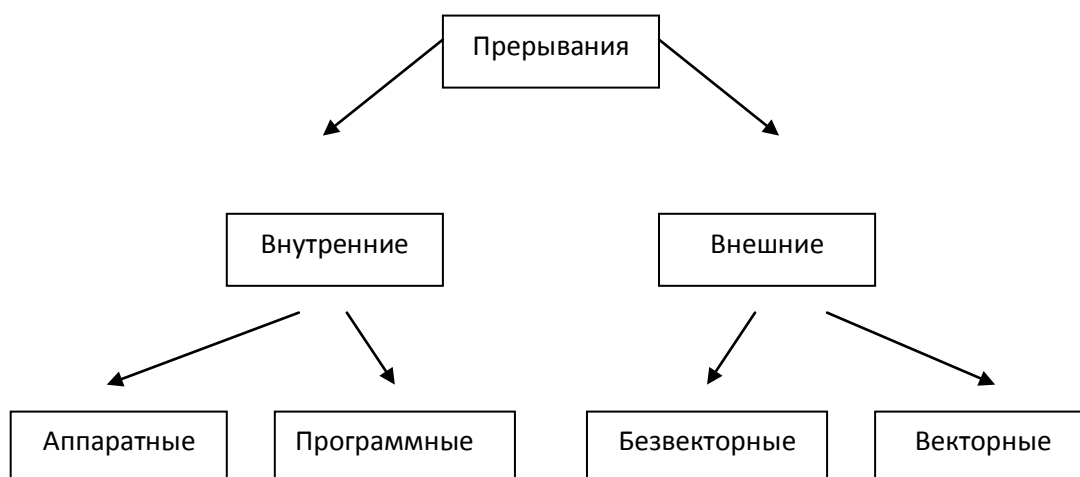


Рис. 14. Классификация прерываний

Внутренние прерывания

Аппаратные прерывания формируются определенными схемами процессора при наступлении определенных событий (деление на ноль, зависание при обращении к памяти или ВУ).

Программные прерывания вызываются при выполнении команд прерывания. Эти программы могут вводиться программистом в исходную программу или вставляться компилятором в процессе компиляции программы. Такие прерывания удобны в процессе отладки системы (они имитируют внешние прерывания), а также являются универсальным средством для вызова стандартных подпрограмм операционной системы.

Внутренние прерывания имеют фиксированные начальные адреса подпрограмм для их обслуживания.

Внешние прерывания

Существуют векторные и безвекторные внешние прерывания. Запросы на безвекторные прерывания поступают на специальные входы процессоров. Эти запросы являются более приоритетными, чем запросы на векторные прерывания. Механизм обработки безвекторных прерываний заложен в процессоре на микропрограммном или аппаратном уровне. В конечном счете, механизм обеспечивает переход к подпрограмме по определенному адресу (с запоминанием адреса возврата и состояния прерванной программы). Безвекторные прерывания, как правило, используются процессором на его локальной магистрали, причем, входы запросов имеют определенное назначение (отключение питания, срабатывание внешнего таймера и т.д.).

Внешние векторные прерывания являются важным средством синхронизации процессов в МВС. С помощью сигналов внешних прерываний устройства системы сообщают друг другу о готовности к передаче данных, о переключении режимов работы, то есть обмениваются управляющей информацией. Если для приема сигналов прерываний в процессорах преду-

					ИАЛЦ.463627.007 ПЗ	Лист
						40
Змн.	Лист	№ докум.	Подпись	Дата		

смотрены специальные входы, то для формирования сигналов требования прерываний могут понадобиться дополнительные средства.

Внешние векторные прерывания осуществляется с помощью специального контроллера следующим образом.

По запросам от внешних устройств контроллер с учетом системы приоритетов выдает на процессор сигнал требования прерывания *IRQ*. Условием формирования этого сигнала является готовность устройств к взаимодействию с процессором и отсутствие маскирования запросов со стороны процессора.

Получив сигнал *IRQ*, процессор завершает до конца выполнение очередной команды, выдает на контроллер сигнал подтверждения прерывания *IACK* и считывает с шины данных вектор, который выставляет на шину контроллер.

Процессор сохраняет (как правило, в стеке) адрес возврата и состояние прерываемой программы, вычисляет адрес перехода, используя для этого принятый вектор, и затем осуществляет переход на первую команду подпрограммы. Процессор выполняет подпрограмму. Последняя команда подпрограммы является специальной командой возврата из прерывания, которая восстанавливает состояние прерванной программы и передает ей управление.

Контроллеры прерываний могут быть централизованными и распределенными. Централизованные контроллеры выполнены в виде отдельного устройства. Распределенные контроллеры представляют совокупность нескольких блоков. Эти блоки включены в состав внешних по отношению к процессору активных устройств, которые могут работать в режиме прерываний.

Схема подключения централизованного контроллера прерываний (КП) показана на рис. 5.14. Внешние устройства (ВУ), в числе которых могут

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		41

быть и процессоры, формируют запросы на прерывания IRQ . Особенность такой схемы заключается в том, что подключение ВУ к магистрали процессора не является обязательным. Это объясняется тем, что вектор V передает в процессор КП, а не внешние устройства. Благодаря этому через прерывания могут взаимодействовать процессоры, подключенные к разным магистралям.

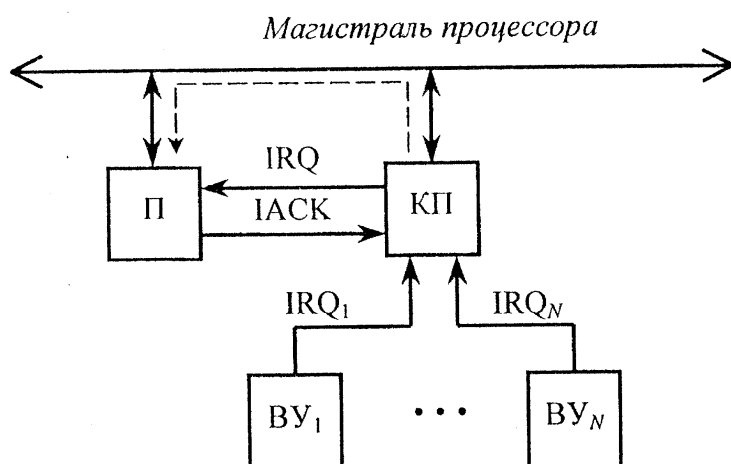


Рис. 15. Система с централизованным контроллером прерываний

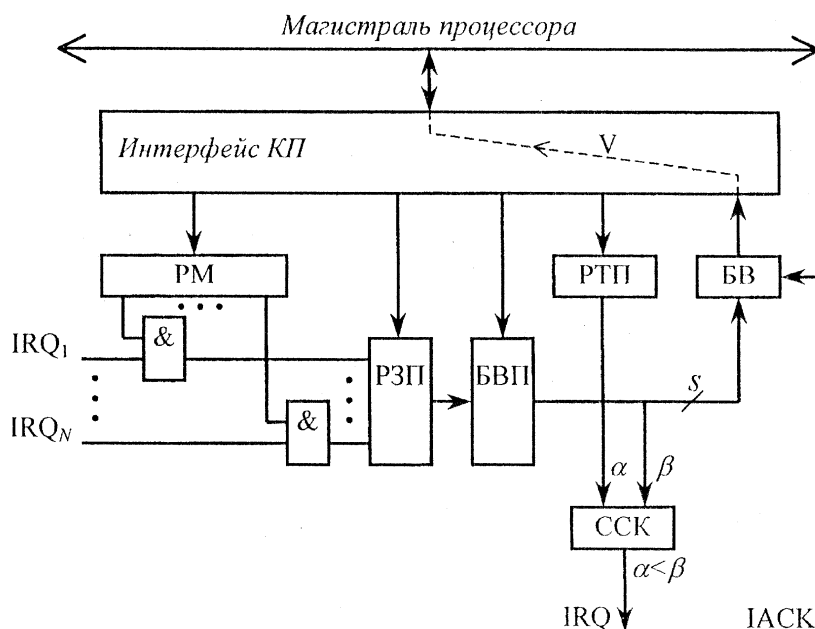


Рисунок 16. Централизованный КП

Возможный вариант построения КП показан на рисунке 15. Контроллер подключен к магистрали процессора через интерфейс, который обеспечивает процессору доступ к регистрам маски (РМ) и текущего приоритета (РТП). Адреса указанных регистров включены в адресное пространство процессора. Записью слова маски в РМ процессор может разрешить или запретить прерывания от определенных источников. Незамаскированные запросы IRQ, записываются в регистр запросов прерываний (РЗП). Блок выбора приоритета (БВП) формирует код самого старшего приоритета, который в схеме сравнения кодов (ССК) сравнивается с текущим приоритетом (приоритетом выполняемой процессором программы). Если запрашиваемый приоритет выше текущего, то формируется сигнал IRQ, а затем по ответному сигналу IACK вектор через буфер вектора (БВ) и интерфейс выдается в магистраль и принимается процессором. В РТП записывается новый код приоритета. В данном контроллере роль вектора исполняет код номера запроса.

К достоинствам централизованных контроллеров следует отнести следующее:

- возможность динамически изменять стратегию обслуживания заявок;
- быстрое выборочное маскирование запросов на прерывание.

Централизованный арбитр потенциально позволяет обеспечивать различные дисциплины обслуживания заявок, так как все заявки поступают в одно устройство. В простейшем случае в качестве БВП используется приоритетный шифратор, который формирует код старшего уровня приоритета. При этом все запросы имеют фиксированные приоритеты, уровень которых определяется номером входа КП. За счет усложнения схемы можно обеспечить другие дисциплины обслуживания заявок. Наиболее часто используют циклическое изменение уровней приоритетов, что обеспечивает гарантированное обслуживание любого запроса на

определенном промежутке времени, причем, независимо от интенсивностей запросов.

Так как регистр маски РМ включен в адресное пространство процессора, то при условии, что число запросов не превышает разрядности шины данных (что обычно выполняется), процессор может за одно обращение к этому регистру записать в него любую маску.

К недостаткам централизованных арбитров следует отнести:

- большое число линий запросов в шине управления (равное числу ВУ);
- ограничение на максимальное число источников прерываний;
- возможная неоднородность процессорных модулей.

Последний из указанных недостатков обусловлен тем, что в централизованных системах приемником прерываний обычно является управляющий процессор. Следовательно, при необходимости использования в системе резерва аппаратуры на уровне модулей потребуются разные типы резерва.

Указанные недостатки ограничивают возможности применения способа при построении однородной модульной МВС с открытой архитектурой.

2.5.3. Режим прямого доступа к памяти (ПДП)

В разработанной МПС на структурной схеме ИАЛЦ.423637.003 Э1 использован централизованный контроллер прямого доступа к памяти (КПДП).

Целью этого способа обмена является освобождение ЦП от необходимости управлять пересылкой слова или массива данных между ОП и ВУ, и предоставление процессору возможности выполнить в это время другие задачи.

Повышается скорость передачи данных. Это происходит за счет того, что КПДП не обращается через СМ за командами, в то время как ЦП постоянно читает команды при передаче слова (массива).

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		44

В КППД есть регистры, которые относятся к адресному пространству ВУ или ОП. Настройка КППД может осуществляться в программном режиме П обработки прерываний, т.е. указывается операция (направление передачи массива данных), длина массива, адрес начала передачи, в некоторых системах номер ВУ.

То есть определяется начальный адрес массива, количество слов в массиве и куда пересылать, тип передачи (слово, байт).

Направление передачи, тип передачи (блок или байт), бит пуск \square в RS; после получения команды П и КППД начинает захватывать на определенное время СМ независимо друг от друга. При блочной передаче данных КППД захват СМ на все время передачи массива.

В режиме одиночной передачи – на один цикл обращения к ОП и ВУ. В первом случае П простаивает все время, необходимое для передачи массива. Во втором случае П и КППД работают параллельно. П выполняет свою программу (обращение к ОП за командами и данными), а КППД захватывает циклы на передачу одного слова.

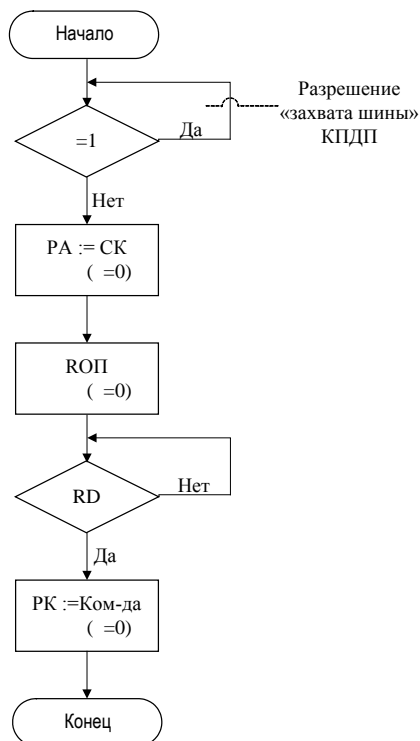


Рис. 2.15 Микроалгоритм выборки команды

Закончив работу (выборку команды) Пр. выставляет сигнал $\alpha=1$ и начинает выполнять свою программу.

Синхронизация процессов захвата СМ осуществляется следующим образом:

- ВУ вырабатывает сигнал Требования ПДП;
- КПДП вырабатывает Требования шины (HOLD);
- П захватывает очередной цикл обращения к СМ (только цикл, а не команду конца); (РПДП=1) ($\alpha=1$);
- При наличии ТШ П формирует сигнал Подтверждения шины, и переводит свои выводы, связанные с СМ в 3-е состояние (высокоомное). П находится в отключенном состоянии, пока действует сигнал ТШ;
- (для случая передачи одиночных данных из П в ВУ);
- КПДП читает слово из ОП в RD и не отпуская СМ, передает слово из RD во ВУ. При этом сигнал Подтверждения ПДП является активным;
- КПДП останавливает сигнал ТШ и ППДП, т.е. отдает СМ процессору; ($\alpha=0$);
- Процессор выполняет очередной цикл обращения к СМ.

При блочной передаче сигнал ТШ не снимается до конца передачи блока.

Блочная передача (если есть КПДП) обеспечивает передачу массива быстрее чем П, потому что он является устройством, не требующим считывания команд для передачи данных; управление на аппаратном уровне. Одиночная – выигрыш потому что параллельно работает П и КПДП.

КПДП и П обмениваются двумя сигналами ТПТШ и ПППШ (требование и подтверждение). П проверяет сигнал ТПТШ (от КПДП) после каждого цикла обращения к ОП. Завершив цикл обращения и если есть сигнал ТПТШ он выдает сигнал в КПДП ПППШ и отключается от шины, до тех пор, пока держится сигнал ТПТШ. (Т.е. ОШ захватывает КПДП). КПДП □ активное устройство.

После завершения цикла работы с каналом КПДП снимает сигнал ТПТШ, после чего активным устройством становится процессор и продолжает свою работу до следующего сигнала ТПТШ.

Рассмотрим схему: в БМУ (П) добавлен один разряд α , который показывает, что в данной МК происходит цикл обращения к ОП.

При $\alpha=1$, КПДП разрешается «захват шины», если $\alpha=0$ КПДП запрещает захват шины.

Сигнал β (считывается с ТзПД);

Если $\beta=1$ процессор отключен от ОШ (т.к. $\alpha=1$).

Если $\beta=0$ – процессор сам захватывает ОШ ($\alpha=0$).

При $\alpha=1$ подключается ТзПД, который формирует сигнал $\beta=1$ и ППДП. Этим сигналом отключается буфер P_r от СМ, т.е. П от ОШ, этим же сигналом блокируется работа БМУ, сигнал β анализируется вначале каждой команды.

Если в системе несколько КПДП то должен использоваться арбитраж (по своим функциям похожий на арбитраж INT и так же он может быть централизованным или распределенным).

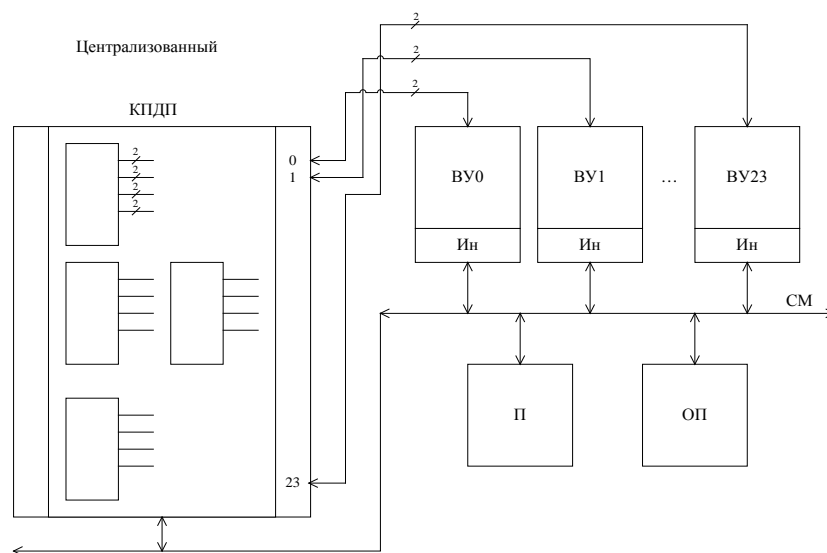


Рисунок 2.16 Централизованный КДП

Синхронизация процесса захвата системной магистрали

1. Внешнее устройство, одного или несколько, которым необходим прямой доступ к ОП выставляют на шину ТПДП свои запросы (шина виртуальное «И»). ТПДП (ТШ) снимается с элемента И1, Т2ПДП в этих устройствах устанавливается в единичное состояние.

2. В ответ на сигнал ТПДП (ТШ) процессор, после того, как закончит свой цикл обмена с ОП, выдаст сигнал Под ПДП на соответствующую шину ($\alpha=1$).

3. По сигналу Под ПДП устанавливается триггер ПВ (Т2) в единичное состояние во ВУ которое требует «захват шины». Этот триггер настраивает «дсузи-цепочку» в каждом ВУ.

4. Через некоторый промежуток времени П выдаст сигнал ПП ПДП. Сигнал ПП ПДП проходит через i -е ВУ, если устройство не выставило сигнал ТШ, в противном случае, сигнал ПП не проходит дальше, и это устройство «захватывает шину». Т.е. оно наиболее приоритетно, в нем замыкается «дсузи-цепочка».

5. Это происходит следующим образом. На входе элемента &3 появляется единичный сигнал, через $F_{пв}$ он поступает на шину подтверждения выборки (ПВ), (шина монтажное &). Этим сигналом снимается сигнал ТПДП (ТШ). (через &2)

6. Сигнал ПВ подтверждает «захват шины» на цикл обмена. Т.е. он держится до тех пор, пока ВУ i осуществляет пересылку одного слова или массива слов в ОП. В ответ на сигнал ПВ процессор отключается от системной магистрали сигналом ОЕ ПДП; ($\alpha=1$). Т2ПВ сбрасывается сигналом с БУ «сброс ПВ».

ППДП сбрасывает т2. Т2ПДП, но он уже не играет роль, т.к. ПВ уже установился, уже зафиксировано подключение этого устройства.

3. Разработка ПО

Программа чтения массива байт из внешней памяти

Допустим, требуется прочесть и каким-то образом обработать 30 байт из 7-й страницы внешней памяти данных, начиная с адреса 35h. Делается это следующим образом:

```
MOV R1, #30           ;количество считываемых байт
MOV DPTR, #0735       ;начальный адрес, откуда считываются данные
L1:
MOVX A, @DPTR         ;описание команды далее
;обработка байта из ВПД, находящийся в аккумуляторе
INC DPTR              ;сформировать адрес следующего байта
DJNZ R1, L1           ;R1:=R1-1 и переход на L1
END                   ;конец
```

Подробнее опишем, что происходит при выполнении команды MOVX. Младшие восемь разрядов адреса (которые в нашем варианте используется для адресации внутри одной страницы внешней памяти данных) записываются в порт P0. Заметим, что P0 работает как мультиплексированная шина адрес/данные. Поэтому сразу же подается сигнал ALE, по перепаду которого содержимое P0 записывается во внешний регистр адреса. Старшие разряды адреса (3 младших бита задают номер страницы) посылаются на порт P2. Так как в данном случае происходит чтения из ВПД, то генерируется сигнал R. Этот сигнал подается по одной из линий порта P3, биты которого имеют специальные функции. Считанные данные записываются в порт P0 и программа продолжает выполнение.

Программа записи массива байт во внешнюю память

Задача: требуется записать 40 байт в 2-ю страницу ВПД, начиная с адреса AEh. Программа почти аналогична предыдущей, за исключением нескольких строк.

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		49

```

MOV R1, #40      ;количество записываемых байт
MOV DPTR, #02AE   ;начальный адрес, куда записываются данные
L1:
    ;запись в A каким-то образом сформированного байта
MOVX @DPTR, A     ;описание команды далее
INC DPTR          ;сформировать адрес следующего байта
DJNZ L1           ;R1:=R1-1 и переход на L1
END              ;конец

```

Выполнение команды MOVX мало чем отличается от того, что она делает в предыдущей программе. 16 разрядов адреса пересылаются таким же образом, но после запоминания младших восьми разрядов из порта P0 во внешний регистр, в порт P0 записывается записываемый в ВПД байт. После этого генерируется сигнал W, который также посылается по одной из линий порта P3.

Подпрограмма выполнения операции умножения

Следует сказать, что система команд микроконтроллера BE51 уже содержит операцию умножения 8-разрядных чисел. Тем не менее, покажем, как реализовать умножение 16-разрядных чисел с формированием 32-разрядного результата при помощи подпрограммы. Умножение выполняется по второму способу.

Операционная схема умножения по второму способу показана далее.

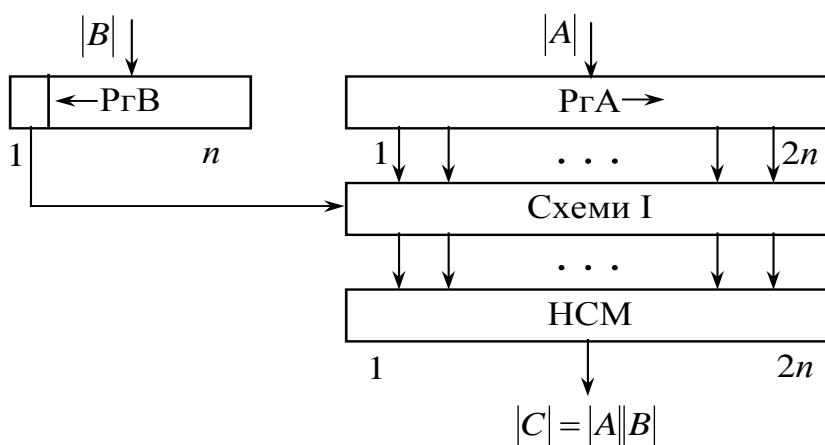


Рисунок 3.1 Операционная схема умножения по второму способу

Таблица 3.1 – Пример умножения по методу 2

Суматор HCM	Регістр PrA	Регістр PrB	Примітки
0000000000 + 0101000000 0101000000	0101000000 →	①0011 ←	П.с. HCM:=0; PrA:= $2^{-1} A $; PrB:= $ B $
+ 0000000000 0101000000	0010100000 →	①011_	$b_1=1$; HCM:=HCM+PrA;
+ 0000000000 0101000000	0001010000 →	①11__	PrA:=R1PrA; PrB:=L1PrB; $b_2=0$; HCM:=HCM+0;
+ 0000000000 0101000000	0000101000 →	①11__	PrA:=R1PrA; PrB:=L1PrB; $b_3=0$; HCM:=HCM+0;
+ 0000101000 0101101000	0000101000 →	①11__	PrA:=R1PrA; PrB:=L1PrB; $b_4=1$; HCM:=HCM+PrA;
+ 0000010100 0101111100	0000010100	①11__	PrA:=R1PrA; PrB:=L1PrB; $b_5=1$; HCM:=HCM+PrA

В каждом цикле к сумме частичных произведений добавляется или множное, если $= 1$, или ноль, если $= 0$, после чего сумма частичных произведений умножается на t , т.е. сдвигается на один разряд вправо. После завершения n -го цикла образуется окончательный результат умножения. Отсюда следует, что умножение начинается с младших разрядов множителя i сдвигается сумма частичных произведений вправо на один разряд в каждом цикле.

Для реализации данного метода умножения нужны n -разрядный регистр умножений РГА, n -разрядный регистр множителя РГО с кругами для сдвига влево, n схем И и $(2n + 1)$ -разрядный накопительные сумматор НСМ с кругами для смещения вправо. Здесь умножений всегда прилагается к n старших разрядов суммы частичных произведений. Один дополнительный разряд слева в НСМ необходимый для запоминания цифры переполнения, который может возникнуть в процессе добавления; при следующем сдвига эта цифра пойдет в старший из основных разрядов накопительные сумматора, так что в конечном результате переполнения не будет.

Программа, реализующая данный алгоритм выглядит следующим образом.

fmull:

MOV R7,#17

l1:

MOV A,R1

CLR C

RRC A

MOV R1,A

MOV A,R2

RRC A

MOV R2,A

MOV A,R3

RRC A

MOV R3,A

MOV A,R4

RRC A

MOV R4,A

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		52

CLR C

MOV A,R5

RLC A

MOV R5,A

MOV A,R6

RLC A

MOV R6,A

JNC L2

MOV R0,#4Ah

CLR C

MOV A,@R0

ADDC A,R4

MOV @R0,A

DEC R0

MOV A,@R0

ADDC A,R3

MOV @R0,A

DEC R0

MOV A,@R0

ADDC A,R2

MOV @R0,A

DEC R0

MOV A,@R0

ADDC A,R1

MOV @R0,A

L2:

DJNZ R7,I1

					ИАЛЦ.463627.007 ПЗ	Лист
						53
Змн.	Лист	№ докум.	Подпись	Дата		

```

MOV R0,#4Ah
MOV A,@R0
MOV R4,A
DEC R0
MOV A,@R0
MOV R3,A
DEC R0
MOV A,@R0
MOV R2,A
DEC R0
MOV A,@R0
MOV R1,A

```

Ret

Программа вычисления функции

Заданная по варианту функция выглядит следующим образом:

$$X=(\sqrt{A}+B)+ (C^2+\sqrt{A})$$

Org 0

Jmp Program_start ; прыжок в начало программы

Program_start; начало программы

MOV A, #05 ; X [0..7]

XCH A, R1

MOV A, #00 ; X[8..15]

XCH A,R2

MOV A, #08 ; y[0..7]

XCH A, R5

MOV A, #00 ; y[8..15]

XCH A,R6

CALL fnull

MOV R0,#5Ah

MOV A,R4

MOV @R0,A

DEC R0

MOV A,R3

MOV @R0,A

DEC R0

MOV A,R2

MOV @R0,A

DEC R0

MOV A,R1

MOV @R0,A

MOV A, #05 ; X [0..7]

XCH A, R1

MOV A, #00 ; X[8..15]

XCH A,R2

MOV A, #05 ; y[0..7]

XCH A, R5

MOV A, #00 ; y[8..15]

XCH A,R6

CALL fnull

MOV R0,#5Ah

MOV A,R4

ADD @R0,A

DEC R0

MOV A,R3

ADDC @R0,A

DEC R0

```

MOV A,R2
    ADDC @R0,A
    DEC R0
MOV A,R1
    ADDC @R0,A
    MOV A, #08 ; X [0..7]
XCH A, R1
MOV A, #00 ; X[8..15]
XCH A,R2
MOV A, #08 ; y[0..7]
XCH A, R5
MOV A, #00 ; y[8..15]
XCH A,R6
CALL fnull
MOV R0,#5Ah
    MOV A,R4
    ADD @R0,A
        DEC R0
        MOV A,R3
        ADDC @R0,A
        DEC R0
    MOV A,R2
        ADDC @R0,A
        DEC R0
    MOV A,R1
        ADDC @R0,A

    MOV A, #05 ; X [0..7]
XCH A, R1

```

```

MOV A, #08 ; y[0..7]
ADD A,R1
XCH A,R2
MOV A, #00 ; X[8..15]
XCH A, R5
MOV A, #00 ; y[8..15]
ADDC A,R5
XCH A,R6

```

```

MOV R0,#5Ah
MOV A,R2
ADD @R0,A

DEC R0
MOV A,R6
ADDC @R0,A
DEC R0
MOV A,#0
ADDC @R0,A
DEC R0
MOV A,#0
ADDC @R0,A

```

End

4. Разработка функциональной схемы

По заданию, взятому у преподавателя, нужно было разработать МПС которая включает в себя КПП,и ППА ВВ55 . Программируемый периферийный адаптер применяется для параллельного обмена данными с квитированием и без него как в режиме программного управления, так и по прерываниям.

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		57

При этом организация однонаправленного или двунаправленного обмена данными выполняется программированием ВВ55.

Результирующую схему можно увидеть на чертеже ИА/Ц 462637 004. Е2.

5. Разработка принципиальной схемы

По заданию, взятому у преподавателя, нужно было разработать МПС которая включает в себя КПП,и ППА ВВ55 . Так как принципиальная схема почти ничем не отличается от функциональной, кроме того, что подписаны все входы и выходы, то ее описание в принципе будет таким же. Программируемый периферийный адаптер применяется для параллельного обмена данными с квитированием и без него как в режиме программного управления, так и по прерываниям. При этом организация однонаправленного или двунаправленного обмена данными выполняется программированием ВВ55.

Результирующую схему можно увидеть на чертеже ИА/Ц 462637 005.

6. Заключение

В данном проекте была разработана МПС, предназначенная для решения специализированных задач. Для увеличения быстродействия путем освобождения процессорного времени в данной МПС были использованы такие расширения Фон-Неймановской архитектуры:

- Контроллер прямого доступа к памяти;*
- Контроллер приоритетных прерываний;*

В данном курсовом проекте была рассмотрен микроконтроллер 1816ВЕ51.

В завершение можно сказать, что применение устройств на базе программируемых микроконтроллеров имеет широкое применение сейчас (они содержатся практически в любом электронном устройстве, работа которой основана на определенной логике или алгоритме) и в будущем.

					ИА/Ц.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		58

Заменить их имеют возможность только все более дешевеющие универсальные микропроцессоры, которые позволяют строить гораздо более сложные системы.

7. Список литературы

- 1. Конспект лекций по курсу "Архитектура ЭВМ"*
- 2. В. І. Жадін, І.А.Жуков, В.В. Ткаченко, І.А. Клименко, Мікропроцесорні системи, 2009 – 492с.*
- 3. В.И.Жадин, В.В.Ткаченко, Логические основы и схемотехника ЭВМ. ВЕК 1997г.*
- 4. В.И.Жадин, В.В.Ткаченко, Архитектура однокристальных ЭВМ. ВЕК 1997г.*
- 5. В.И.Жадин, В.В.Ткаченко, Однокристальные микро-программируемые ЭВМ. Киев 1995г.*
- 6. И.В.Пацюра, В.И.Корнейчук, Л.В.Довбыш, Надёжность электронных систем.*
- 7. Самофалов К. Г., Корнейчук В. И., Тарасенко В. П., "Электронные цифровые вычислительные машины", К:1976. – 404 с.*

					ИАЛЦ.463627.007 ПЗ	Лист
Змн.	Лист	№ докум.	Подпись	Дата		59

