

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ВОЛЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

А.А. Силаев

Архитектура ЭВМ

Электронное учебное пособие для ВУЗов



Волжский
2018

УДК 004.2(07)
ББК 32
С 36

Рецензенты:

канд. тех. наук, доцент филиал г. Волжский МЭИ(ТУ)
Капля Е.В.,
канд. тех. наук, доцент ВолгГТУ
Макаров А.М.

Печатается по решению редакционно-издательского совета
Волгоградского государственного технического университета

Силаев, А.А.

Архитектура ЭВМ промышленности [Электронный ресурс] : учебное пособие /А. А. Силаев ; ВПИ (филиал) ВолгГТУ. – Электрон. текстовые дан. (1 файл: 2,13 МБ). – Волжский, 2018. – Режим доступа: <http://lib.volpi.ru>. – Загл. с титул. экрана.

ISBN 978-5-9948-3094-9

Ил. 12, библиограф Учебное пособие содержит базовые сведения об архитектуре ЭВМ. Отдельные разделы посвящены архитектуре микропроцессоров и организации основной памяти ЭВМ. Рассмотрены общие тенденции развития архитектуры ЭВМ.

Учебное пособие представляет собой конспект лекций.

Предназначено для студентов технических ВУЗов обучающихся по направлениям бакалавриата, связанных с вычислительной техникой всех форм обучения.

∴ 5назв.

ISBN 978-5-9948-3094-9

© Волгоградский государственный
технический университет, 2018
© Волжский политехнический
институт, 2018

Содержание

Введение	5
1 Введение в вычислительную технику	6
1.1 Базовые понятия	6
1.2 История развития компьютеров	6
1.3 Структурная схема персонального компьютера	7
1.4 Архитектура PC-совместимых компьютеров	12
1.5 Особенности функционирования ЭВМ	12
1.6 Выполнение команды	13
2 Микропроцессоры	15
2.1 Общие сведения	15
2.2 Структура микропроцессора	16
2.3 Принцип построения микропроцессоров	17
2.4 Основные функции микропроцессора	18
2.5 Архитектура микропроцессора	19
2.6 Регистровая модель микропроцессора	23
2.7 Архитектура процессора по формату используемых команд	25
2.8 Архитектура ЭВМ по способу организации выборки команд и данных	26
2.9 Конвейерный принцип выполнения команд	27
2.10 Суперскалярные процессоры	28
2.11 Основные классы микропроцессоров	30
3 Адресация команд и данных	33
3.1 Организация адресации в ЭВМ	33
3.2 Способы адресации	35
4 Система команд ЭВМ	40
4.1 Формат команды	40
4.2 Способы уменьшения формата команды	40

4.3	Система команд	42
4.4	Организация прерываний в микропроцессорных системах	44
5	Основная память компьютера	46
5.1	Оперативная память	46
5.2	Постоянная память	47
5.3	Иерархическая структура памяти	48
5.4	Классификация запоминающих устройств	49
5.5	Постоянные запоминающие устройства	51
5.5.1	Масочные ПЗУ	52
5.5.2	Программируемые ПЗУ (ППЗУ)	53
5.5.3	Репрограммируемые ПЗУ	54
5.5.4	Флеш-память	55
5.6	Статические ОЗУ	56
5.6.1	Основные виды микросхем статической памяти	58
5.7	Динамические ОЗУ	59
5.7.1	Работа элемента памяти в динамическом ОЗУ	61
5.7.2	Асинхронная и синхронная DRAM	62
5.7.3	Синхронные динамические ОЗУ	64
	Заключение	73
	Список литературы	74

Введение

С развитием вычислительной техники совершенствовалась и архитектура ЭВМ. В настоящее время существует большое разнообразие вычислительных систем, которые обладают разной архитектурой. Но не смотря на всё разнообразие ЭВМ можно выделить их общие базовые особенности.

В пособие наибольшее внимание уделено архитектуре персонального компьютера. Рассмотрены основные его компоненты.

Отдельная часть пособия посвящена архитектуре микропроцессора. Рассмотрены различные классификации архитектур процессоров и способы адресации команд и данных.

Отдельная часть посвящена организации основной памяти компьютера.

Учебное пособие предназначено для студентов технических ВУЗов, обещающихся по направлениям бакалавриата, связанных с вычислительной техникой и программированием.

1 Введение в вычислительную технику

1.1 Базовые понятия

ЭВМ (компьютер) представляет собой совокупность средств, предназначенных для цифровой обработки информации с выводом результатов в требуемой форме.

Компьютеры прошли долгий путь от простых механических вычислителей до сложных цифровых электронных систем, способных обрабатывать большие объемы информации.

Архитектура ЭВМ – концептуальная структура вычислительной машины, определяющая способы обработки информации и включающая методы преобразование информации в данные и принципы взаимодействия технических средств и программного обеспечения.

Среди имеющегося многообразия архитектур ЭВМ особое место занимает архитектура персонального компьютера (ПК), обладающего уникальными свойствами:

- обеспечивает пользователю комфортные условия интерактивного взаимодействия через различные устройства ввода-вывода;
- организует персональное личное пространство пользователя в цифровом мире;
- имеет широкие функциональные возможности.

1.2 История развития компьютеров

В истории развития вычислительной и компьютерной техники выделяют несколько поколений устройств, отличающихся элементной базой, назначением, функционально-логической организацией, конструктивно-технологическим исполнением, программным обеспечением, техническими и эксплуатационными характеристиками, пользовательским интерфейсом и другими показателями.

Первое поколение (1945–1955). Элементная база: электронные лампы и ферритовые сердечники. Устройства ввода-вывода: стандартная телеграфная аппаратура.

Второе поколение (1955–1965). Элементная база: полупроводниковые транзисторы и магнитные носители (ленты и барабаны). Устройства ввода/вывода: первые дисплеи с возможностью попиксельного отражения информации. Появились алгоритмические языки и программное обеспечение.

Третье поколение (1965–1980). Элементная база: интегральные схемы с многослойным печатным монтажом, представляющие собой законченный функциональный узел со сложной транзисторной схемой.

К особенностям следует отнести:

- появились первые микропроцессоры и микросхемы памяти;
- магнитные диски;
- появление удалённого доступа к ЭВМ с помощью сети;
- дальнейшее развитие операционных систем.

Четвертое поколение (1980–2000). Элементная база: большие и сверхбольшие интегральные схемы. Главная особенность данного поколения — это появление персональных компьютеров. В 1981 году появился первый компьютер фирмы IBM PC. Компьютеры получили широкое распространение.

Пятое поколение (2000 – по настоящее время). Элементная база: сверхбольшие интегральные схемы (многоядерные процессоры). Идёт дальнейшее совершенствование аппаратного и программного обеспечения. Развитие мобильных и сетевых технологий. Появление индустрии 4.0.

1.3 Структурная схема персонального компьютера

С точки зрения выполняемых функций ЭВМ, можно представить, как микропроцессорную систему, состоящую из четырёх блоков (рисунок 1):

- **центральный процессор** обеспечивает основную функцию: цифровую обработки информации;

- **блок основной памяти** предназначен для хранения и выдачу информации центральному процессору при исполнении программ;
- **блок периферийных (внешних) устройств** предназначен для организации взаимодействия ЭВМ с пользователем и внешней средой;
- **блок интерфейсов** обеспечивает взаимодействие всех устройств ЭВМ между собой на аппаратном и программном уровнях.

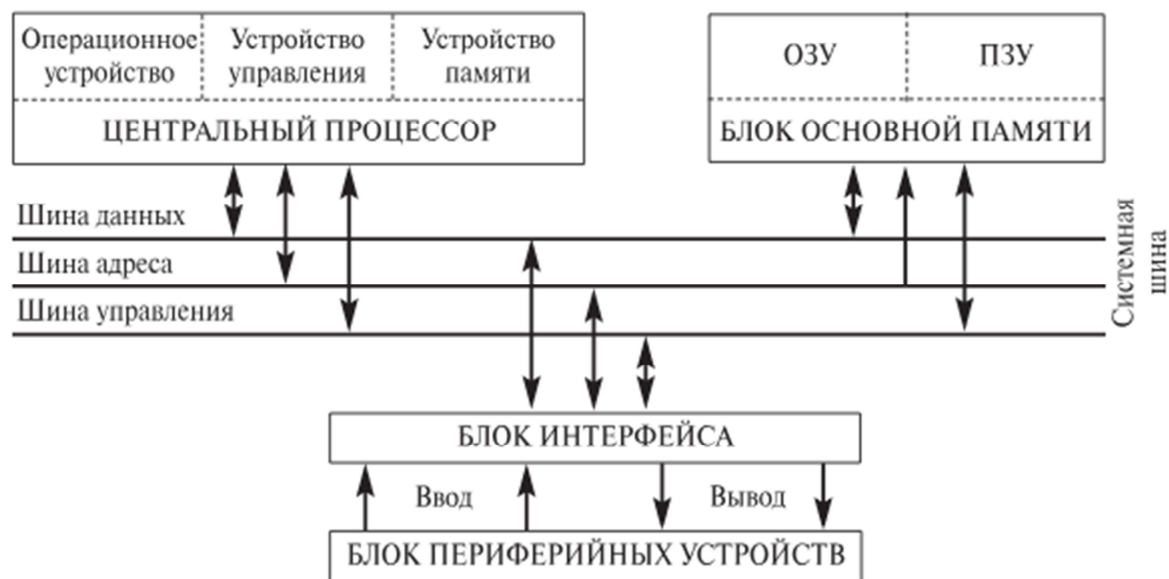


Рисунок 1 – Структурная схема персонального компьютера.

Для понимания обобщённой структуры ЭВМ рассмотрим её функциональные блоки более подробно.

Центральный процессор. В общем случае состоит из трех частей:

- **оперативное устройство** выполняет определённый набор команд (инструкций). Набор команд современных процессоров достигает нескольких сотен для реализации требуемых действий (выполнение целочисленных операций и операций с плавающей точкой, потоковая обработка данных и др.). В первых процессорах операционное устройство формировалось на основе арифметически-логического устройства (АЛУ);
- **устройство управления** предназначено для управления процессом выполнения команд, а именно для формирования требуемой

последовательности управляющих сигналов (микрокоманд, выполняемых на одном процессорном такте). Набор микрокоманд представляет собой микропрограмму для данной команды. При выполнении отдельных микрокоманд используются осведомительные сигналы (логические условия, признаки и флаги), поступающие со стороны операционного устройства. Осведомительные сигналы информируют управляющее устройство о состоянии операционного устройства;

– **устройство памяти** представляет собой набор регистров, образующий внутреннюю быстродействующую память процессора. Часть регистров доступна программисту и предназначена для хранения операндов и выполнения действий над ними, формирования адреса для взаимодействия с основной памятью и других действий. Хранение операндов во внутренних регистрах процессора значительно ускоряет выполнение программы, так как отсутствует обращение к основной памяти. Другая часть регистров используется процессором для служебных (системных) целей (доступ к этим регистрам может быть ограничен или запрещён). Дополнительно к устройству памяти относят и встроенную кэш-память современных процессоров.

Блок основной памяти. Этот блок состоит из двух устройств:

– **оперативное запоминающее устройство (ОЗУ)** предназначено для хранения команд выполняемой программы и данных, подлежащих обработке. Объём современных ОЗУ достигает десятки гигабайт. Обращение к ОЗУ по системной шине требует значительных затрат времени, поэтому для повышения скорости обмена данными дополнительно вводится быстродействующая буферная кэш-память, которая располагается в центральном процессоре. Оперативная память представляет собой единый массив памяти, непосредственно доступный процессору для записи и чтения данных, а также считывания программного кода. ОЗУ реализуется на модулях динамической памяти;

– **постоянное запоминающее устройство (ПЗУ)** предназначено для хранения стандартных (неизменяемых) программ и констант. В ПЗУ обычно записывают программы начальной загрузки системы, тестовые и диагностические программы и другое служебное программное обеспечение, которое не изменяется в процессе работы ЭВМ. В качестве ПЗУ используется память ROM (Read Only Memory) – память только для чтения и репрограммируемая память EEPROM – Electrically Erased Programmable Read Only Memory или флеш-память.

ОЗУ и ПЗУ логически располагаются в едином адресном пространстве, являются основной памятью ЭВМ, обеспечивая её нормальное функционирование.

Блок периферийных устройств. Периферийные (внешние) устройства – это устройства, не обладающие системными ресурсами (адресное пространство памяти, каналы запросов прерываний и прямого доступа к памяти), которые используются устройствами для обмена данными. Процессор может непосредственно обращаться к системным ресурсам и устройствам, но не может напрямую обращаться к периферийным устройствам. Поэтому периферийные устройства подключаются к интерфейсам системных устройств.

Пример. Накопитель на жёстких магнитных дисках (НЖМД), подключенный к контроллеру НЖМД, является периферийным устройством, так как не занимает отдельных системных ресурсов и процессор обращается к нему через ресурсы контроллера. Контроллер НЖМД – это системное устройство, так как имеет системные ресурсы в виде портов и линий прерываний, имеющих собственные адреса и процессор может непосредственно с ними работать.

Системная шина, состоит из трех отдельных шин:

– **шина данных** служит для передачи команд из основной памяти в устройство управления процессора, а также для обмена обрабатываемыми данными между процессором и ОЗУ или периферийными устройствами. В

первых ЭВМ числи линий данных шины соответствовало разрядности операндов, обрабатываемых процессором. В новых поколениях ЭВМ допустимо увеличение количества линий шины данных по сравнению с разрядностью операндов для одновременной передачи сразу нескольких операндов по шине;

- **шина адреса** предназначена для передачи адреса. Адрес формируется процессором для выбора нужной ячейки основной памяти или системного устройства. Разрядность шины адреса определяет максимальный объём памяти, доступной процессору.

Пример. 32-разрядная шина адреса обеспечивает адресацию памяти объёмом до 2^{32} байта = 4 Гбайта;

- **шина управления** передает различные управляющие сигналы, которые обеспечивают различные режимы работы памяти (чтение/запись), периферийных устройств (ввод/вывод информации) и процессора (запуск, тест запрос на обслуживание внешних устройств и др.).

Блок интерфейса. Это блок, который обеспечивает взаимодействие между процессором, основной памятью и периферийными устройствами (рисунок 1). Интерфейсные средства делятся на группы:

- **шинный интерфейс** – это шины расширения (шины ввода/вывода) и системная шина. Шины расширения позволяют увеличить функциональность ЭВМ за счёт возможности подключения периферийных устройств. Непосредственно через шины расширения к системной плате подключают различные адаптеры и контроллеры периферийных устройств. Адаптеры и контроллеры служат средством сопряжения периферийных устройств с конкретной шиной и обладают системными ресурсами (адресация процессора происходит к ним напрямую);

- **системная логика** представляет собой набор микросхем, находящийся на системной плате для организации обмена данными между процессором и

периферийными устройствами. Такой набор называют чипсетом, который включает интерфейс шины процессора, контроллеры памяти и др.

– **конструктивные интерфейсные средства** – это различные конструктивные части: разъемы, слоты, сокет, переключатели, кабели и др.

1.4 Архитектура РС-совместимых компьютеров

Архитектура ЭВМ представляет собой совокупность его аппаратных и программных средств, обеспечивающих обработку цифровой информации. В основу компьютера заложена транспортная архитектура ЭВМ, предложенная Дж. фон Нейманом в 1945г. Рассмотрим её основные особенности:

– ЭВМ содержит АЛУ, память, устройства управления (контроллеры периферийных устройств и часть центрального процессора) и устройства ввода/вывода;

– программы и данные хранятся в одной и той же памяти;

– выполняемые операции определяются АЛУ и устройством управления, которые составляют основу центрального процессора;

– при исполнении команд используется принцип последовательной передачи управления. Т.е. центральный процессор выбирает команды из памяти и исполняет их одну за другой, при этом адрес следующей команды определяется специальным счётчиком адреса.

Данная архитектура легла в основу большинства ЭВМ.

1.5 Особенности функционирования ЭВМ

Работа ЭВМ базируется на принципе дискретности. Для исполнения одной микрокоманды выделяется временной интервал (такт). Рассмотрим выполнение одной микрокоманды:

– в начальный момент времени ЭВМ находится в состоянии $A = \{a_i\}$, где a_i – состояния отдельных устройств системы, которые хранит память в виде последовательности нулей и единиц.

- поступившая на вход системы микрокоманда в виде совокупности сигналов $\{X_m\}$, переводит систему в другое состояние — $V = \{b_j\}$, где b_j – состояния отдельных устройств системы;

- новое состояние ЭВМ V запоминает и хранит память;

- по завершении выполнения всей предписанной последовательности микрокоманд из памяти считывается требуемая последовательность сигналов, которая подаётся на выбранное устройство ввода/вывода.

Упорядоченная во времени совокупность микрокоманд образует команду, а последовательность команд и данных – программу.

1.6 Выполнение команды

Время, затрачиваемое на выполнение команды, называется командным циклом. Командный цикл разбивается на машинные циклы. Машинным циклом называется промежуток времени между двумя последовательными обращениями процессора к ОЗУ или внешнему устройству по системной шине. Длительность машинного цикла состоит из трех и более системных тактов (периодов синхросигналов шины), которые требуются:

- для установки нужного адреса;
- для выдачи сигналов, определяющих вид цикла (чтение/запись);
- для получения сигнала готовности к обмену (от памяти ли внешних устройств);
- для передачи данных и команд.

Код команды представляет собой многоразрядное число, которое делится на две части:

- код операции определяет вид операции, выполняемой данной командой;
- код адресации операндов задает адреса источников операндов, над которыми производится данная операция, и адрес результата операции.

Коды команд хранятся в оперативной памяти.

Рассмотрим процесс выполнения команды, который разбивается на несколько этапов.

В начале первого машинного цикла по адресу, который задаётся содержимым программного счётчика РС (Program Counter), из ОЗУ считывается код подлежащей выполнению команды. Код команды поступает в регистр команд центрального процессора, входящий в состав его устройства управления.

Далее происходит дешифрация кода команды, где определяется:

- вид выполняемой операции и адрес необходимых операндов;
- необходимое число машинных циклов для выполнения команды.

Если для выполнения команды не требуется считывание операндов из памяти или запись результатов операции в память, то такая команда выполняется за один машинный цикл. Иначе требуется выполнение дополнительных циклов чтения/записи. В зависимости от разрядности обрабатываемых операндов и разрядности системной шины зависит число циклов необходимых для выполнения команды.

После считывания кода текущей команды содержимое программного счётчика РС автоматически увеличивается на один, если выполнялась операция перехода, то на более. Это позволяет обеспечить последовательную выборку команд в процессе выполнения программ. При выборке очередной команды содержимое РС поступает на шину адреса, обеспечивая считывание из ОЗУ следующей команды выполняемой программы. При реализации переходов (ветвлений) или других изменений последовательности выполнения команд происходит загрузка в счётчик РС нового содержимого, в результате чего осуществляется переход к другой ветви программы.

В соответствии с выполняемой операцией устройство управления формирует необходимые сигналы для реализации машинных циклов и требуемую последовательность микрокоманд в каждом цикле.

2 Микропроцессоры

2.1 Общие сведения

Микропроцессор – это программно-управляемое устройство в виде большой интегральной микросхемы, предназначенное для обработки цифровой информации. Микропроцессор – это универсальное устройство, которое находит применение не только в ЭВМ, но и в различных цифровых системах. Например, системы автоматического управления различными объектами и процессами, информационно-измерительные системы и др. Они находят широкое применение в промышленности, науке и технике.

Рассмотрим обобщённую структуру микропроцессора, состоящую из набора блоков:

центральное ядро состоит из операционного устройства (или нескольких) и устройства управления;

внутренняя память в виде регистров, кэш-памяти, модулей оперативной и постоянной памяти;

интерфейсный блок обеспечивает выход на системную шину и обмен данными;

периферийные устройства – специализированные контроллеры, таймерные модули, аналого-цифровые преобразователи; вспомогательные устройства (например, генератор тактовых импульсов).

Архитектура процессора представляет собой совокупность его аппаратных и программных средств, обеспечивающих выполнение программы и определяет его логическую организацию.

Архитектура процессора включает в себя набор программно-доступных регистров и операционных устройств, систему основных команд и способов адресации, объём и организацию адресуемой памяти, виды и способы обработки данных.

Например, 32-разрядные процессоры x86 с архитектурой IA-32 (Intel Architecture – 32 bit) имеют стандартный набор регистров, общую систему основных команд, одинаковые способы организации и адресации памяти, защиты памяти и обслуживания прерываний.

2.2 Структура микропроцессора

В структурно-функциональном отношении микропроцессор разделяется на две части: операционный и управляющий автоматы (см. рисунок 2).

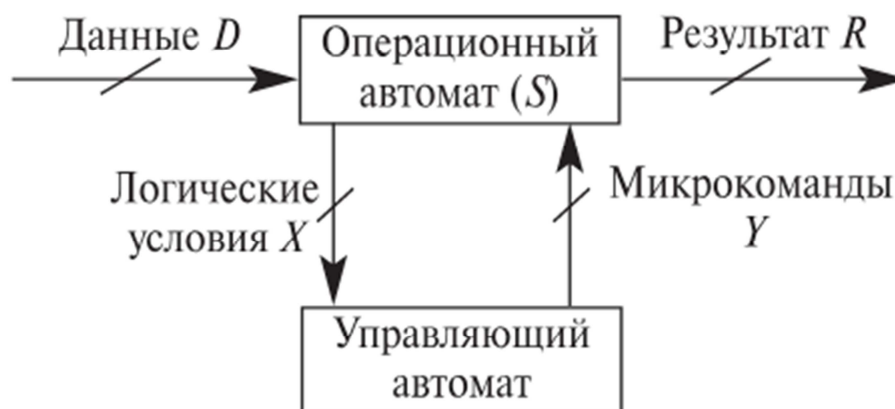


Рисунок 2 – Структурно-функциональная схема микропроцессора.

Операционный автомат предназначен:

- для хранения множества входных (D), выходных (R) и внутренних (S) слов;
- для выполнения набора микроопераций для получения результата R;
- для формирования множества осведомительных сигналов X, каждый из которых отождествляется с определенным логическим условием.

Микрооперации, реализуемые операционным автоматом, инициируются множеством управляющих сигналов $Y = \{y_1 - y_N\}$, каждый из которых соответствует определенной микрооперации.

Операционный автомат можно представить в виде трех функциональных модулей (рис. 3): памяти, комбинационной схемы микроопераций и комбинационной схемы логических условий.

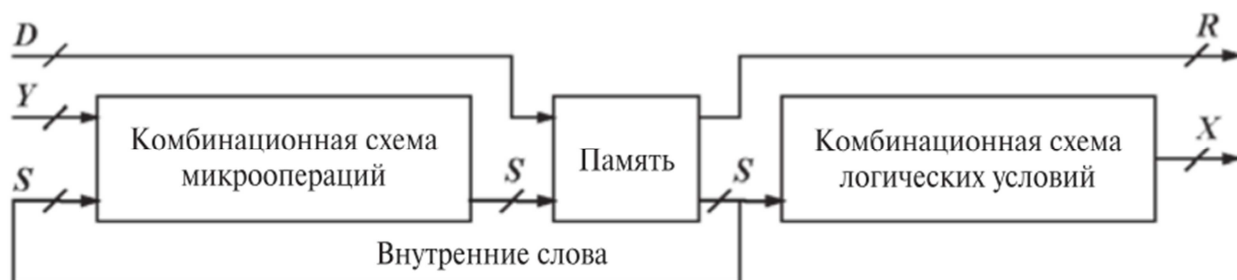


Рисунок 3 – Структура операционного устройства.

Управляющий автомат генерирует последовательность управляющих сигналов из множества Y , предписанную микропрограммой и соответствующую значениям логических условий X . При выполнении микропроцессором пакета микропрограмм на его входы последовательно подаются коды операций, которые соответствуют той или иной микропрограмме. На входы микропроцессора могут поступать внешние сигналы логических условий, а с выходов сниматься сигналы для управления периферийными устройствами.

Структура управляющего автомата во многом зависит от принципа его построения. В управляющих автоматах, созданных по принципу схемной логики, необходимая последовательность управляющих сигналов Y вырабатывается с помощью определенным образом соединенных логических элементов. При использовании принципа программируемой логики управляющие сигналы Y вырабатываются на основе микропрограммы, хранимой в постоянных запоминающих устройствах.

2.3 Принцип построения микропроцессоров

Принцип микропрограммного управления. Построение микропроцессоров базируется на принципе микропрограммного управления:

- любая операция, реализуемая микропроцессором, это сложное действие, которое разделяется на последовательные элементарные действия, называемые микрооперациями;
- для управления порядком следования микроопераций используются логические условия, которые отражают состояние микропроцессора после выполненных микроопераций;
- процесс выполнения операций в микропроцессоре описывается в форме алгоритма, представленного в терминах микроопераций и логических условий, и называется микропрограммой;
- микропрограмма используется как форма представления функции микропроцессора, на основе которой определяются его структура и порядок функционирования во времени.

2.4 Основные функции микропроцессора

При выполнении программы микропроцессор обеспечивает выполнение всех функций, предусмотренных программой:

- формирование адреса команд или данных, хранящихся в ОЗУ;
- выборка команд из памяти и их дешифрация;
- прием данных из основной (оперативной) памяти, выполнение над ними арифметических, логических и других операций, определяемых кодом команды, и передача обработанных данных во внешние устройства или память;
- формирование сигналов состояния (логических условий), управления и времени, необходимых для нормальной работы внутренних узлов, а также внешних устройств и памяти;
- временное хранение результатов выполненных операций, адресов, формируемых сигналов состояния и других данных;
- прием сигналов запроса от внешних устройств и их обслуживание. Для выполнения перечисленных функций процессор должен располагать необходимым набором аппаратных средств.

2.5 Архитектура микропроцессора

Структурная схема микропроцессора, представленная на рисунке 4, содержит четыре указанные выше составные части:

- операционный автомат, к которому можно отнести АЛУ и обслуживающие его регистры, блок десятичной коррекции и регистр признаков, формирующий логические условия;
- управляющий автомат, включающий регистр команд, дешифратор и блок управления;
- внутреннюю память, состоящую из регистров общего назначения, указателя стека, счетчика команд, регистра адреса и буферного регистра;
- средства коммуникации в виде внутренней шины, шины данных и шипы управления.



Рисунок 4 – Структурная схема микропроцессора

Арифметическо-логическое устройство (АЛУ) позволяет выполнить следующие операции над 8-разрядными операндами:

- арифметическое сложение двух операндов с передачей переноса в старший разряд (и без него) и вычитание с передачей заёма в младший разряд (и без него);
- логическое сложение, умножение, исключаящее ИЛИ и сравнение;
- четыре вида циклических сдвигов;
- арифметические операции над десятичными числами.

Блок десятичной коррекции. При суммировании десятичных чисел может потребоваться коррекция результата, для выполнения которой в микропроцессоре предусмотрен блок десятичной коррекции. При сложении каждый разряд десятичного числа (цифра) представляется 4-разрядным двоичным кодом (полубайтом, тетрадой). Сложение полубайтов происходит по правилам двоичной арифметики. Если сумма больше 9, то возникает необходимость в проведении коррекции. Коррекция осуществляется дополнительным прибавлением двоичного числа 0110 (6) к полученному результату. Это обусловлено тем, что вес 5-го разряда двоичного числа составляет 16 десятичных единиц, а вес старшего разряда десятичного числа – 10, т.е. разность равна 6. Если при сложении получен результат 10...15, то прибавление числа 6 (0110)₂ приведет к появлению 1 в 5-м разряде. Этот разряд "уходит" в старший полубайт и забирает с собой "добавку" (6), оставляя правильный результат. В том случае, когда результат составляет 16... 18, в 5-м разряде появляется единица, значение которой (до переноса) соответствует 16 в десятичном исчислении. После переноса единицы 5-го разряда в старший полубайт ее значение становится равным 10. Таким образом, перенос сопровождается уменьшением результата на 6 десятичных единиц, что также требует проведения его коррекции.

Регистр признаков. АЛУ непосредственно связано регистром признаков (флагов), в котором фиксируется результат выполнения некоторых арифметических и логических операций:

флаг переноса, вырабатывающий сигнал $C = 1$, если при выполнении операций сложения и сдвига появляется единица переноса из старшего разряда;

флаг дополнительного переноса, вырабатывающий сигнал $V = 1$, если при выполнении операции с двоично-десятичными кодами появляется единица из третьего разряда (старшего разряда младшего полубайта);

флаг нуля, вырабатывающий сигнал $Z = 1$, если результат операции равен нулю;

флаг знака, вырабатывающий сигнал $S = 1$, если результат операции – отрицательное число;

флаг четности, вырабатывающий сигнал $P = 1$, если результат операции содержит четное число единиц.

Указанные признаки обеспечивают выполнение в программе условных переходов. Например, если результат выполнения предыдущей операции равен нулю, то флаг нуля устанавливается в единичное состояние ($Z = 1$) и условие перехода в другую часть программы оказывается выполненным.

Регистры. Доступ к регистрам, в том числе к счетчику команд и указателю стека, осуществляется через мультиплексоры с помощью селектора регистров.

Регистры общего назначения (РОН) играют роль аккумуляторов, когда в них содержатся обрабатываемые данные, либо указателей, когда в них хранятся адреса операндов.

Программный счетчик (Program Counter – PC). Счетчик указывает адрес, где находится в памяти очередной байт команды. Байты команд обычно выбираются в порядке нарастания их адресов. Поэтому после выборки каждого очередного байта увеличивается содержимое программного счетчика на единицу. Перед началом выборки какой-нибудь команды в счетчик заносится адрес ее первого байта. Обычный порядок следования адресов может быть изменен. Для этого в счетчике предусмотрена возможность записи начального адреса той части программы, которая должна выполняться. Эту часть программы обычно называют подпрограммой.

Указатель стека (Stack Pointer – SP) представляет собой регистр, предназначенный для быстрой адресации особого вида памяти, именуемой стеком. Стековая память может быть выделена в основной памяти компьютера. Она используется при обслуживании прерываний. В стек заносится адрес возврата к прерванной программе на время обработки микропроцессором подпрограммы, содержимое аккумулятора, регистра признаков. Стековая память характеризуется тем, что обращение к ней осуществляется по принципу: первыми выбирают данные, поступившие последними.

Блок управления. Часть исполняемой команды записывается в регистр команд. В дешифраторе команд формируются сигналы, под действием которых в устройстве управления запускается микропрограмма для выполнения требуемой операции. Микропрограммы операций, определяемых набором команд микропроцессора, "защиты" в управляющую память. На входы устройства управления поступают:

- две неперекрывающиеся последовательности тактовых импульсов ($\Phi 1$, $\Phi 2$) с периодом T ;
- сигнал готовности (READY) внешних устройств и памяти к обмену информацией с микропроцессором;
- сигнал запроса от внешних устройств на прерывание (INT) выполнения основной программы и переход на выполнение подпрограмм обслуживания прерывания;
- сигнал запроса от внешних устройств на захват шин (HOLD), как правило, для организации обмена информацией по каналу прямого доступа к памяти;
- сигнал сброса (RESET), по которому происходит начальная установка микропроцессора.

С выхода устройства управления снимаются сигналы для управления внутренними узлами процессора и внешними устройствами. Для управления внешними устройствами вырабатываются:

- сигнал синхронизации (SYNC), указывающий на начало каждого машинного цикла – промежутка времени, необходимого для одного обращения процессора к внешним устройствам или памяти;
- сигнал приема (DB1N), подтверждающий готовность процессора к приему данных;
- сигнал ожидания (WAIT), свидетельствующий о том, что процессор находится в состоянии ожидания;
- сигнал подтверждения захвата (HLDA), подтверждающий, что шины находятся в высокоомном состоянии, и внешние устройства могут обращаться к памяти напрямую, минуя процессор;
- сигнал разрешения прерывания (INTE), свидетельствующий о том, что триггер разрешения прерывания в блоке управления находится в состоянии логической единицы, при котором возможен прием сигналов запроса;
- сигнал выдачи ($WR = 0$), указывающий на то, что процессор выдал информацию на шину данных для ее записи в память или передачи во внешние устройства.

Интерфейс микропроцессора. Обмен информацией между узлами процессора осуществляется по внутренней шине данных, а с внешней шиной данных (ШД) – через буферный регистр. Для адресации к памяти и внешним устройствам используются шина адреса (ША) и регистр адреса. К интерфейсу можно также отнести буферный регистр и регистр адреса.

2.6 Регистровая модель микропроцессора

При написании программ на низком уровне необходимо знать какие из регистров микропроцессора можно использовать для хранения операндов и управляющих сигналов. Совокупность таких регистров образуют регистровую модель микропроцессора.

Выделяют две группы регистров.

Регистры общего назначения (РОН), предназначенные для хранения операндов. Эта группа регистров образует внутреннюю память микропроцессора.

Внутренняя структура регистров общего назначения приведена на рисунке 5.

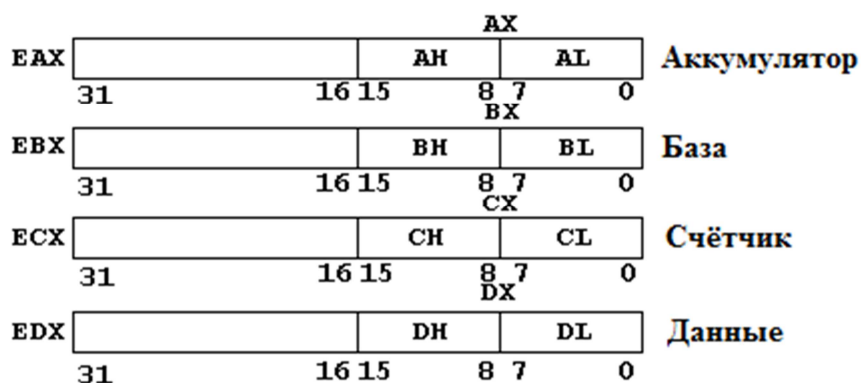


Рисунок 5 – Структура регистров общего назначения.

Обмен информацией между микропроцессором и внешними устройствами осуществляется через двунаправленный буферный регистр, а адресация к памяти и внешним устройствам – через регистр адреса. Особенность буферного и адресного регистров заключается в том, что предусмотрен режим прямого доступа к памяти внешним устройствам.

Служебные регистры, предназначенные для управления исполняемой программой, обеспечения требуемого режима работы процессора, организации обращения к памяти и выполнения других функций.

Основные служебные регистры:

- программный счётчик PC (или указатель команд IP);
- регистр состояния SR (Status Register) или регистр флагов (EFLAGS);
- регистры, реализующие сегментную и страничную организацию памяти;
- регистры, обеспечивающие отладку программ и тестирования.

Функционирование процессора можно представить, как процедуры изменения состояния регистров (регистровые пересылки) путём чтения/записи их содержимого. В результате таких пересылок обеспечивается адресация и

выборка команд и операндов из основной памяти, хранение и пересылка результатов, изменение последовательности команд и режимов функционирования процессора в соответствии с поступлением нового содержимого в служебные регистры, а также все другие процедуры, реализующие процесс обработки информации согласно заданным условиям.

2.7 Архитектура процессора по формату используемых команд

CISC-архитектура (Complex Instruction Set Computer) – архитектура процессора со сложным набором команд. Реализована во многих типах микропроцессоров (например, Pentium), выполняющих большой набор разнообразных команд с использованием многочисленных способов адресации. Система команд процессоров с CISC-архитектурой может содержать более 200 команд разной степени сложности (от 1 до 15 байт) и использовать десятки различных способов адресации. Недостаток CISC-архитектуры – дальнейшее её развитие связано с существенным усложнением структуры процессора, повышением его стоимости и увеличением времени выполнения команд.

RISC-архитектура (Reduced Instruction Set Computer) – архитектура процессора с сокращённым набором команд. RISC-процессоры реализуют около 100 команд, имеющих фиксированный формат длиной 4 байта, и используют небольшое число наиболее простых способов адресации (регистровую и индексную). Для уменьшения числа обращений к внешней памяти RISC-процессоры содержат большое количество регистров общего назначения. Обращение к внешней памяти используется только для обмена данными с внешней памятью. В результате упрощается структура микропроцессора, сокращаются его размеры и стоимость, повышается производительность. Поэтому в некоторых моделях CISC-процессорах используют RISC-ядро.

VLIW-архитектура (Very Large Instruction Word) – архитектура процессора с использованием сверхдлинных команд. Отдельные поля команды содержат

коды, обеспечивающие выполнение различных операций. Одна VLIW-команда может выполнить сразу несколько операций одновременно в различных узлах микропроцессора. Формирование такой команды производит специальный компилятор при трансляции программ, написанных на языках высокого уровня. Данная архитектура получила распространение в видеопроцессорах.

2.8 Архитектура ЭВМ по способу организации выборки команд и данных

Принстонская архитектура (архитектура Фон Неймана) имеет следующие характерные особенности:

общая оперативная память для хранения программ, данных и организации стека, что позволяет оперативно и эффективно перераспределять её объём в зависимости от решаемых задач в каждом отдельном случае применения процессора;

общая системная шина, по которой в процессор поступают команды и данные, а в оперативную память записываются результаты, что значительно упрощает контроль за функционированием системы, повышая её надежность. Но при этом использование общей шины ограничивает производительность цифровой системы.

Гарвардская архитектура. Основной особенностью которой является физическое разделение памяти на память команд и данных. Память команд и данных соединяются с процессором по разным шинам. Из-за разделения потоков команд и данных, а также совмещению операций их выработки обеспечивается более высокая производительность, чем при использовании принстонской архитектуры. К недостаткам гарвардской архитектуры относят: усложнение конструкции процессора; фиксированный объём памяти для команд и данных; увеличение общего объёма данных из-за невозможности её оптимального распределения между командами и данными.

Гарвардская архитектура получила широкое применение в микропроцессорных системах управления, и во внутренней структуре высокоскоростных процессоров в кэш-памяти с раздельным хранением команд и данных.

2.9 Конвейерный принцип выполнения команд

Процесс выполнения команды разбивается на ряд этапов:

выборка очередной команды (ВК);
декодирование выбранной команды (ДК);
формирование адреса операнда (ФА);
прием операнда из памяти (ПО);
выполнение операции (ВО);
размещение результата в памяти (РР).

Выполнение каждого этапа занимает один такт машинного времени.

Каждый из этапов соответствует выполнению одной микрооперации. Для каждой микрооперации используется отдельное устройство или ступень исполнительного конвейера. Для каждой команды должна соблюдаться очередность выполнения её отдельных этапов или микроопераций.

В конвейере одновременно может находиться несколько команд на разных этапах их выполнения. В идеальном варианте при полной загрузке конвейера однотипными командами на его выход в каждом такте будет поступать результат выполнения очередной команды. В этом случае производительность процессора (операций/с) будет равна его тактовой частоте (тактов/с).

Однако в реальных условиях отдельные ступени конвейера могут оказаться ещё в двух дополнительных состояниях:

- в состоянии ожидания (ОЖ), когда ступень не может выполнить требуемую микрооперацию из-за отсутствия необходимых данных;

в состоянии простоя (ПР), когда ступень вынуждена пропустить очередной такт, так как поступившая команда не требует выполнения соответствующего этапа.

Таким образом на эффективность использования конвейера влияет тип повторяющихся команд: однородные команды сокращают число состояний простоя и ожиданий в процессе их выполнения, разнородные команды, содержащие различное количество байтов, увеличивают число состояний простоя и ожидания.

На эффективность использования конвейера влияют также команды ветвления. При выполнении условия ветвления необходимо перезагружать конвейер командами из другой ветви программы, что ведет к значительному снижению производительности процессора. Для сокращения числа перегрузок конвейера используются различные способы предсказания ветвлений. Один из способов предсказания базируется на предположении, что при повторном обращении к команде условие ветвления сохраняется. Для реализации этого способа используется специальная память — буфер флагов ветвления (Branch Target Buffer), где хранятся адреса ранее выполненных условных переходов. При повторном поступлении команды ветвления предсказывается переход к ветви, которая была выбрана в предыдущем случае, и производится загрузка в конвейер команд из той же ветви. При правильном предсказании не требуется перезагрузка конвейера и эффективность его использования не снижается. Эффективность такого способа предсказания зависит от емкости буфера флагов ветвления.

2.10 Суперскалярные процессоры

Процессор, имеющий в своем составе несколько операционных устройств, обеспечивающих одновременное выполнение команд в исполнительных конвейерах, называется суперскалярным. Каждый конвейер обрабатывает одну из поступивших команд. В идеальном случае число одновременно

выполняемых команд (работающих конвейеров) равно числу операционных устройств. Однако при выполнении реальных программ трудно обеспечить полную загрузку всех исполнительных конвейеров. Суперскалярные процессоры содержат от 4 до 10 различных операционных устройств, параллельная работа которых обеспечивает выполнение за один такт в среднем от 2 до 6 команд.

В суперскалярных процессорах производится выборка нескольких десятков команд, которые декодируются, анализируются и группируются для одновременной (параллельной) загрузки в исполнительные конвейеры. В состав суперскалярного процессора входит:

- несколько устройств для выполнения целочисленных операции;
- одно или несколько устройств для обработки чисел с плавающей точкой;
- отдельные устройства для обработки специальных форматов видео- и аудиоданных;
- устройства формирования адресов и выборки операндов для загружаемых команд.

Направления повышения эффективности работы суперскалярных процессоров.

Организуется предварительная (спекулятивная) выборка данных, которые записываются в специальные регистры. Это позволяет заранее подготовить операнды для поступающих на исполнение команд.

Изменяется порядок следования команд. Команды выполняются не в порядке их выборки из памяти, а по мере готовности необходимых операндов и исполнительных устройств. Позже поступившие команды могут быть выполнены ранее выбранных из памяти команд. Таким образом, обеспечивается более полная загрузка исполнительных конвейеров. Для выдачи результатов в соответствии с исходной последовательностью поступления команд используется специальная буферная память.

Вводятся дополнительные регистровые блоки, дублирующие регистры процессора. Это позволяет реализовать одновременное обращение двух и более команд к одному регистру. При поступлении команд, которые обращаются к одному и тому же регистру, производится их переадресация к дублирующему регистровому блоку путем переименования регистров. В результате обеспечивается возможность одновременного выполнения команд обращения к одному и тому же регистру, что позволяет реализовать более эффективную параллельную работу исполнительных конвейеров.

2.11 Основные классы микропроцессоров

По функциональному признаку выделяют два класса микропроцессоров: это универсальные микропроцессоры (микропроцессоры общего назначения) и специализированные микропроцессоры, среди которых наиболее широкое распространение получили микроконтроллеры и цифровые сигнальные процессоры (Digital Signal Processor – DSP).

Микропроцессоры общего назначения. Этот класс процессоров предназначен для решения широкого круга задач обработки разнообразной информации и находит применение в персональных компьютерах, рабочих станциях, серверах и других цифровых системах массового применения. Некоторые типы микропроцессоров этого класса относят к CISC- процессорам, поскольку используют набор разноформатных команд с различными способами адресации. В их внутренней структуре может содержаться RISC-ядро, выполняющее преобразование поступивших команд в последовательность простых RISC-операций. Другие типы микропроцессоров этого класса непосредственно реализуют RISC-архитектуру.

Практически все современные универсальные микропроцессоры, используют гарвардскую архитектуру с разделением потоков команд и данных при помощи отдельных блоков кэш памяти. В большинстве случаев они имеют суперскалярную структуру с несколькими исполнительными конвейерами.

Микроконтроллеры. Этот класс специализированных микропроцессоров ориентирован на реализацию устройств управления, встраиваемых в разнообразную аппаратуру. Особенностью микроконтроллеров является размещение на одном кристалле центрального процессора, внутренней памяти и большого набора периферийных устройств. В состав периферийных устройств обычно входят несколько параллельных и последовательных портов ввода-вывода данных, таймерный блок, аналого-цифровой преобразователь, а также такие специализированные устройства, как блок формирования сигналов с широтно-импульсной модуляцией, контроллер жидкокристаллического дисплея и ряд других. Благодаря использованию внутренней памяти и периферийных устройств реализуемые на базе микроконтроллеров системы управления содержат минимальное количество дополнительных компонентов.

Для удовлетворения запросов потребителей выпускается большая номенклатура микроконтроллеров. Основными областями их применения являются промышленная автоматика, автомобильная электроника, измерительная техника, теле-, видео- и аудиотехника, средства связи, бытовая аппаратура. Для микроконтроллеров характерна гарвардская архитектура, что подразумевает наличие:

- отдельной внутренней памяти для хранения программ, в качестве которой используются однократно программируемое ПЗУ (PROM) или электрически репрограммируемое ПЗУ (EPROM, EEPROM или Flash);

- отдельной внутренней памяти для хранения данных, в качестве которой используется регистровый блок, организованный в виде ОЗУ.

Для повышения производительности во многих моделях микроконтроллеров реализованы принципы RISC-архитектуры, обеспечивающие выполнение большинства команд за один такт машинного времени.

Цифровые сигнальные процессоры. Этот класс специализированных микропроцессоров предназначен для цифровой обработки поступающих

аналоговых сигналов в реальном времени. Архитектура цифровых сигнальных процессоров (ЦСП) ориентирована на быстрое выполнение последовательности операций умножения-сложения с накоплением промежуточного результата в регистре-аккумуляторе, что обусловлено особенностью алгоритмов обработки аналоговых сигналов.

Значение оцифрованного аналогового сигнала может быть представлено в виде числа с фиксированной или с плавающей точкой. В соответствии с этим ЦСП делятся на два класса.

Процессоры, обрабатывающие числа с фиксированной точкой. К этому классу относятся более простые и дешевые ЦСП – их обычно обрабатывают 16- или 24-разрядные операнды, представленные в виде правильной дроби. Однако ограниченная разрядность в ряде случаев не позволяет обеспечить необходимую точность результатов.

Процессоры, обрабатывающие числа с плавающей точкой. Процессоры этого класса проводят вычисления над 32- и 40-разрядными операндами и обеспечивают более высокую точность результатов.

Для повышения производительности при выполнении специфических операций обработки сигналов в большинстве ЦСП реализуется гарвардская архитектура с использованием отдельных шин для передачи адресов, команд и данных.

3 Адресация команд и данных

3.1 Организация адресации в ЭВМ

Работу процессора можно представить, как непрерывную последовательность исполняемых им команд. Для выполнения каждой команды процессор должен знать:

- какие данные потребуются для исполнения команды, где находятся нужные данные и как к ним обратиться;
- где искать следующую команду.

Поэтому возникает задача адресации команд и данных. Часть из требуемых сведений содержится в коде команды, некоторые сведения поступают в процессе ее исполнения.

Адресация команд. При выполнении линейной части программы, когда команды расположены в памяти одна за другой, для их выборки из памяти используется программный счетчик (указатель команд), т.е. адресация полностью возлагается на аппаратные средства и адрес следующей команды в текущей команде не указывается. Необходимость указания адреса следующей команды в текущей возникает при передаче управления подпрограмме, расположенной в другом месте памяти. Такие случаи встречаются реже, и для их реализации предусмотрен набор команд (CALL, JMP, RET и др.), предназначенный специально для управления программой. В командах управления программой задан способ адресации, определяющий условия передачи управления, для реализации которого используются фиксированные указатели (в виде специальных, а не общих регистров). Например, при вызове подпрограммы новые значения адреса формируются путем загрузки селектора в сегментный регистр кода CS и смещения в программный счетчик, или указатель команды IP. При этом в стеке сохраняется пара CS:IP.

Адресация данных. В описании каждой команды должны быть указаны адреса мест расположения исходных операндов (источников), задействованных

при выполнении операции, и адрес места расположения операнда результата операции (приемника). Источниками и приемниками операндов могут служить регистровая память RSEG (Register Segment), память данных DSEG (Data Segment) или порты ввода-вывода IOSEG (Input/Out Segment). Адрес, определяющий место расположения операнда при выполнении команды, называется исполнительным адресом (Executive Address – EA), а способ формирования исполнительного адреса – способом адресации. Известны различные способы адресации, позволяющие:

- обеспечить не только эффективный доступ к структурированным данным (массивам, стекам, спискам, таблицам), но и перемещение программ при загрузке и выполнении;

- уменьшить длину программного кода и число обращений к системной шине;

- адресовать большой объем памяти при малой разрядности процессора. Для формирования исполнительного адреса используются сведения, заложенные в команде.

Сведения об адресе операнда могут быть представлены в следующих формах.

В явной форме, при которой сведения содержатся в адресном поле команды. Эта форма является более гибкой и эффективной, так как позволяет связывать с одной командой множество адресов. Однако ее использование приводит к увеличению длины кода команды.

В неявной форме, когда исполнительный адрес заложен в именной части команды, т.е. место расположения одного или всех операндов определяется кодом операции.

На практике применяются обе формы представления адресной информации. Например, в команде LDA addr, выполняющей операцию загрузки аккумулятора содержимым ячейки памяти по указанному адресу, адрес

операнда-источника (addr) задан в явной форме, а адрес приемника (аккумулятора) – в неявной (командой LDA).

3.2 Способы адресации

Рассмотрим основные способы адресации на примере загрузки аккумулятора A данными D.

Непосредственная адресация. При этом способе адресации в команде содержится не адрес операнда, а сам операнд (см. рисунок 6). Непосредственная адресация позволяет повысить скорость выполнения операции, так как в этом случае вся команда, включая операнд, считывается из памяти одновременно и на время выполнения команды хранится в процессоре в специальном регистре команд. Однако при использовании непосредственной адресации появляется зависимость кодов команд от данных, что требует изменения программы при каждом изменении непосредственного операнда.

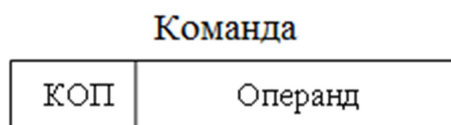


Рисунок 6 – Непосредственная адресация.

Прямая адресация. При прямом способе адресации в коде команды адрес операнда указывается непосредственно в виде некоторого значения (см. рисунок 7). Преимущество этого способа в том, что он самый простой, а недостаток — в том, что разрядность регистров общего назначения процессора должна позволять напрямую обращаться к памяти данных.

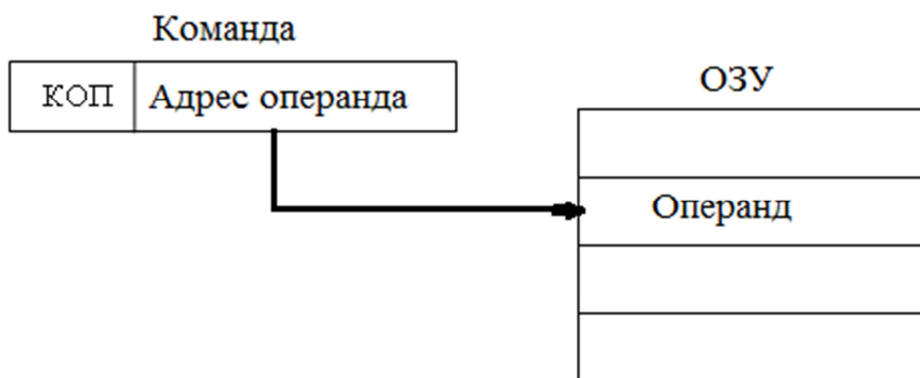


Рисунок 7 – Прямая адресация.

Косвенная адресация. При косвенной адресации в адресной части команды хранится адрес ячейки памяти (см. рисунок 8) или адрес регистра (см. рисунок 9), в которых содержится адрес операнда.

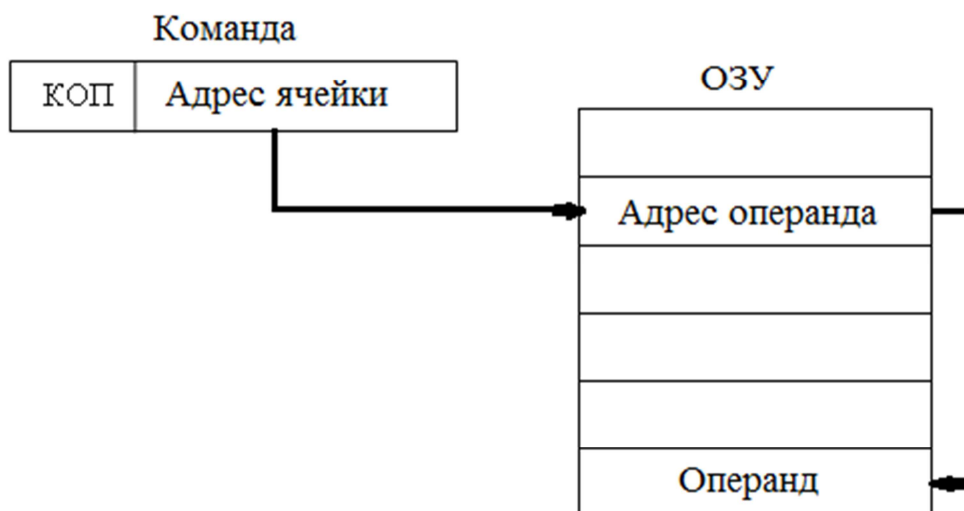


Рисунок 8 – Косвенная адресация через ячейку памяти.

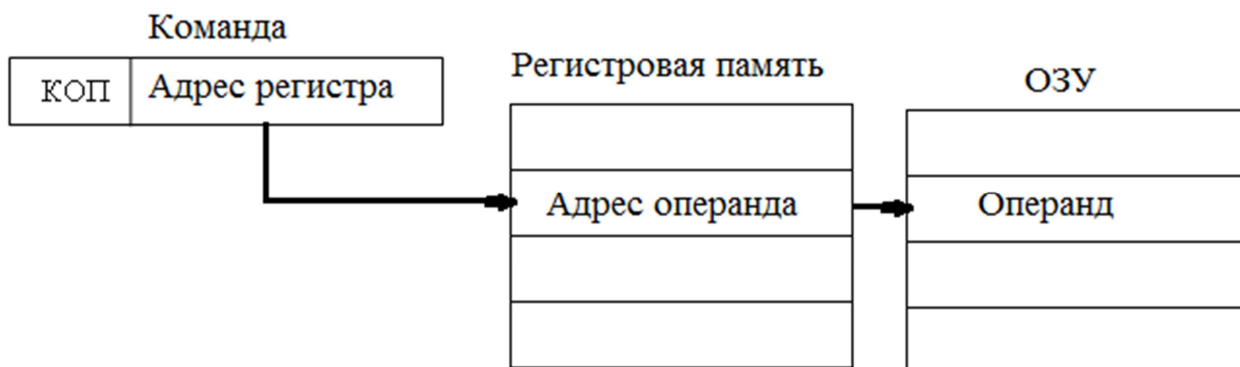


Рисунок 9 – Косвенная адресация через регистр.

Применение косвенной адресации операнда из оперативной памяти при хранении его адреса в регистровой памяти или оперативной существенно сокращает длину поля адреса, одновременно сохраняя возможность использовать для указания физического адреса полную разрядность регистра или ячейки памяти ОЗУ.

Недостаток этого способа – необходимо дополнительное время для чтения адреса операнда. Вместе с тем он существенно повышает гибкость программирования. Изменяя содержимое ячейки памяти или регистра, через которые осуществляется адресация, можно, не меняя команды в программе, обрабатывать операнды, хранящиеся по разным адресам. Так, адресация, при которой после каждого обращения по заданному адресу с использованием механизма косвенной адресации, значение адресной ячейки автоматически увеличивается на длину считываемого операнда, называют автоинкрементной. Адресация с автоматическим уменьшением значения адресной ячейки называется автодекрементной.

Стековая адресация. При стековой адресации текущий адрес ячейки памяти хранится в указателе стека. Этот способ адресации является косвенным и отличается от ранее рассмотренных тем, что необходимость обращения к указателю стека следует непосредственно из кода операции команд работы со стеком.

Относительная адресация. Этот способ используется тогда, когда память логически разбивается на блоки, называемые сегментами. В этом случае адрес ячейки памяти содержит две составляющих: адрес начала сегмента (базовый адрес) и смещение адреса операнда в сегменте. Адрес операнда определяется как сумма базового адреса и смещения относительно этой базы.

Для задания базового адреса и смещения могут применяться ранее рассмотренные способы адресации. Как правило, базовый адрес находится в одном из регистров регистровой памяти, а смещение может быть задано в самой команде или регистре. Рассмотрим оба случая.

1. Адресное поле команды состоит из двух частей (см. рисунок 10): в одной указывается номер регистра, хранящего базовое значение адреса (начальный адрес сегмента), а в другом адресном поле задается смещение, определяющее положение ячейки относительно начала сегмента. Именно такой способ представления адреса обычно и называют относительной адресацией.

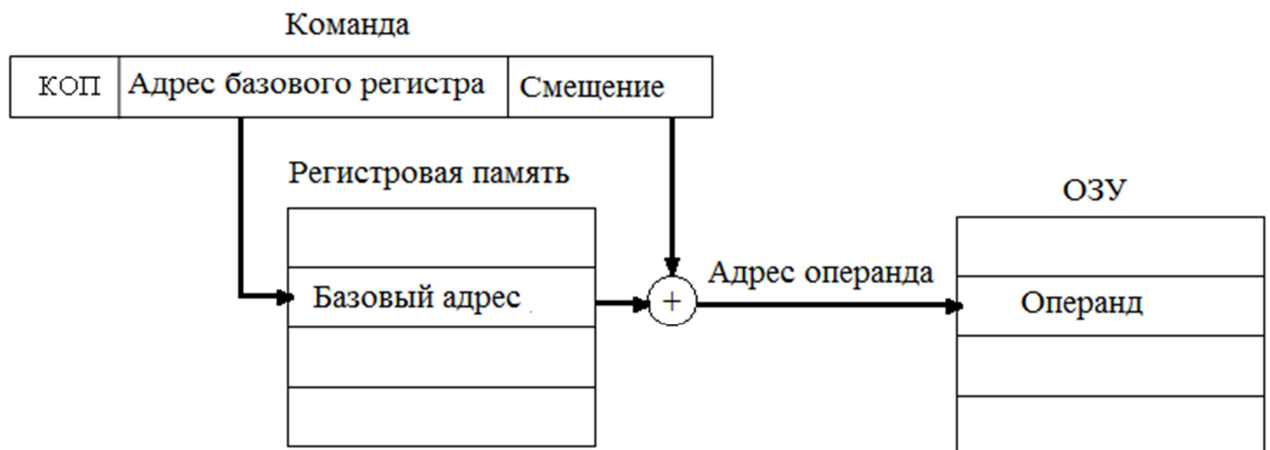


Рисунок 10 – Относительная адресация.

2. Первая часть адресного поля команды также определяет номер базового регистра, а вторая содержит номер регистра, в котором находится смещение (см. рисунок 11). Такой способ адресации чаще всего называют базово-индексным.

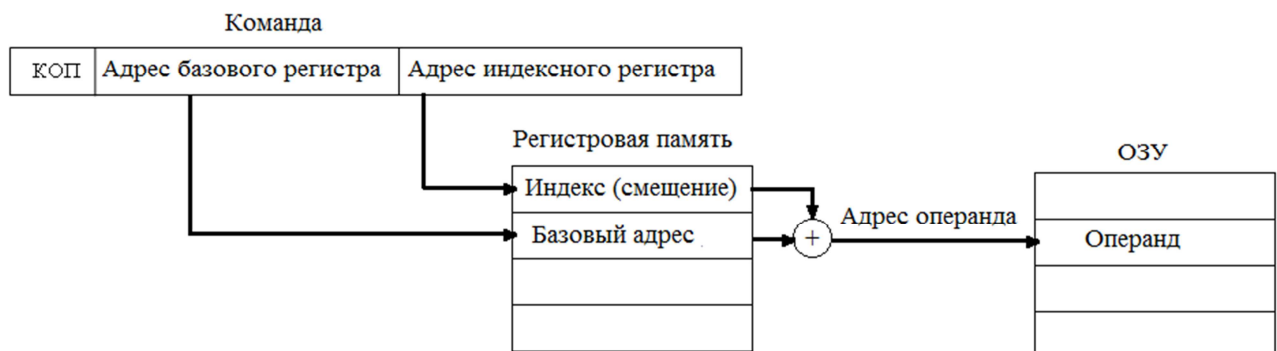


Рисунок 11 – Базово-индексная адресация.

Основной недостаток относительной адресации — это большое время вычисления физического адреса операнда. При этом преимущество этого способа адресации заключается в возможности создания "перемещаемых" программ — программ, которые можно размещать в различных частях памяти

без изменения команд программы. То же относится к программам, обрабатывающим по единому алгоритму информацию, расположенную в различных областях ЗУ. В этих случаях достаточно изменить содержимое базового адреса начала команд программы или массива данных, а не модифицировать сами команды. По этой причине относительная адресация облегчает распределение памяти при составлении сложных программ и широко используется при автоматическом распределении памяти в мультипрограммных вычислительных системах.

4 Система команд ЭВМ

Входное воздействие в виде двоичного кода, предназначенное для управления микропроцессором, называется командой. Команда предписывает шаги по реализации микропроцессором заданной операции, представляющей собой функционально завершенное действие, которое определяется типом используемых данных, источником их получения, операцией над ними, приемником размещения результата, источником получения следующей команды. Машинное представление команды в памяти, состоящее из ряда нулей и единиц, называется объектным кодом команды. Для лучшего восприятия команды используется ее символическое обозначение или мнемокод.

4.1 Формат команды

Каждая команда должна содержать сведения, необходимые для ее выполнения. Для каждой группы сведений выделяется свое поле. Совокупность полей, содержащих необходимые сведения для выполнения требуемой операции, называют форматом команды (см. рисунок 12). В формате команды должны быть определены:

- функциональное назначение операции в виде кода операции;
- адреса источников данных. В общем случае должны быть указаны адреса двух операндов;
- адрес места расположения результата;
- адрес следующей команды.

КО	АО1	АО2	АР	АСК
4 бита	12 бит	12 бит	12 бит	12 бит

Рисунок 12 – Формат команды.

4.2 Способы уменьшения формата команды

Рассмотрим следующий формат команды:

- поле кода операций (КО) занимает 4 разряда, что позволяет закодировать $2^4 = 16$ операций;

- под адреса двух операндов-источников, адрес места расположения результата, адрес следующей команды выделены поля АО1, АО2, АР, АСК (см. рисунок 12) по 12 разрядов, что позволяет в каждом случае адресовать $2^{12} = 4К$ ячеек памяти.

Как видно из рисунка 12, несмотря на скромные возможности команды, ее общая длина составляет достаточно большое число (52) бит. Для сокращения количества разрядов команды часть информации должна быть задана неявно и не должна зависеть от особенностей конкретной команды.

Наиболее употребительными являются следующие способы сокращения длины кода команды:

- использование специально предусмотренных для адресации регистров: программного счетчика, указателя стека и др., Например, при выполнении команд с последовательно возрастающими адресами программный счетчик автоматически считывает из памяти следующую команду. В этом случае в формате команды отсутствует поле адреса следующей команды;

- использование неявных способов адресации. Например, при использовании неявной регистровой адресации адрес следующей команды или операнда хранится в регистре, код которого содержит значительно меньшее число разрядов, чем код исполнительного адреса;

- совмещение источника одного из операндов с приемником результата. В этом случае в формате команды исключается поле адреса результата;

- использование команд с укороченной адресацией, т.е. части адресного пространства памяти;

- использование для некоторой группы операции одного регистра. Наиболее часто для этой цели используется аккумулятор. В операциях с аккумулятором не требуется его кодировка. Например, команда ADD В выполняет операцию сложения содержимого регистра В с аккумулятором А и в

него же помещает результат. Такая команда является одноадресной. Ее формат состоит из двух полей: кода операции и адреса операнда. Однако одноадресные команды требуют дополнительных команд для предварительной загрузки операндов в аккумулятор и последующего размещения результатов в памяти;

- использование нескольких аккумуляторов. В этом случае каждая команда имеет два адреса, однако адрес источника и места назначения может быть задан другим аккумулятором.

Следует отметить, что в одних процессорах все команды имеют одинаковую длину, в других – разную длину. Одинаковая длина всех команд упрощает декодирование, однако требует большего пространства, поскольку все команды должны быть такой же длины, как самая длинная. Команды могут быть короче слова, равными слову или длиннее слова. В процессорах с неймановской архитектурой команды и данные имеют одинаковую длину и поступают в процессор по шине данных. Поэтому для отличия команд от данных в процессоре предусмотрены средства, обеспечивающие:

- засылку команд (первого байта) в регистр команд с дальнейшей их дешифрацией для активизации устройства управления;
- поступление данных (последующих байт) в аккумулятор или другие регистры для обработки в операционном устройстве.

4.3 Система команд

Фиксированный набор команд конкретного микропроцессора называют системой команд. Функциональные способности процессора определяются совокупностью базовых команд с различными кодами операций.

Система команд представляется в виде таблицы. Таблица может иметь различную структуру, однако обычно она содержит следующие сведения о команде:

- мнемоническое обозначение команды, представляющее собой сокращенную запись названия команды;

- шестнадцатеричные коды команд;
- влияние выполненной команды на флаги регистра состояния программы;
- число байтов в команде и число машинных циклов и тактов, затрачиваемых на выполнение команды;
- описание выполняемой командой операции.

Часто для удобства систему команд разбивают на отдельные группы по функциональному признаку. Рассмотрим основные группы команд.

Команды пересылки. Группа содержит наиболее часто встречающиеся в программах команды пересылки данных, источниками и приемниками которых могут быть внутренние регистры процессора, основная память и внешние устройства. Команды не оказывают воздействия на флаги регистра состояния.

Арифметические команды. Набор команд арифметических команд позволяет выполнить:

- сложение и вычитание операндов;
- арифметическое сравнение содержимого аккумулятора А с содержимым одного из регистров общего;
- увеличение и уменьшение на 1 (инкремент и декремент) содержимого регистров и регистровых пар;
- десятичную коррекцию содержимого аккумулятора после выполнения арифметических операций в двоично-десятичном коде.

Команды логических операций. Логические операции являются поразрядными и выполняются независимо для каждого из бит операндов. Команды этой группы позволяют реализовать:

- бинарные логические операции умножения, сложения и исключающее ИЛИ;
- инвертирование содержимого аккумулятора А;
- циклических сдвиги влево и вправо.

Команды передачи управления. При передаче управления нарушается последовательный ход выборки содержимого ячеек памяти и процессор

адресуется в другую область памяти. Выделяют три вида команд передачи управления: команды перехода по заданному адресу, вызова подпрограммы и возврата из подпрограммы. К ним относятся безусловные и условные команды. Условные команды осуществляют переход, вызов подпрограммы и возврат из подпрограммы в зависимости от выполненного условия. Безусловные команды не содержат условие перехода.

Команды общего управления. Команды этой группы используются для задания режима работы микропроцессора:

- команда сброса;
- команда разрешения прерывания;
- команда запрещения прерываний;
- пустая команда используется в циклах программируемой задержки;
- команда останова вызывает прекращение выполнения программы.

4.4 Организация прерываний в микропроцессорных системах

Прерывание (Interruption) — это событие, вызывающее прекращение работы основной программы и переход к выполнению процедуры, или подпрограммы, предназначенной для его обработки. Совокупность аппаратных средств, команд и программ, обслуживающих прерывания, образуют систему прерываний. Прерывание инициируется специальным сигналом, при появлении которого система прерываний изменяет ход выполнения программы. Можно выделить три вида прерываний:

- аппаратные прерывания, которые инициируются внешними устройствами с помощью сигнала запроса на прерывание;
- программные прерывания, которые инициируются специальными командами (например, командой вызова подпрограммы);
- исключения, вызванные возникновением особых условий (случаев) выполнения текущей команды (например, деление на нуль).

Прерывания используются:

- для обмена информацией между процессором и внешним устройством;
- в аварийных ситуациях, например, при понижении напряжения питания;
- при исключительных условиях, таких, как переполнение при выполнении операции с плавающей точкой;
- для индикации аппаратных сбоев, приводящих к ошибкам при обработке данных (например, к ошибкам, обнаруженным при контроле на четность);
- при программных сбоях, т.е. если предпринимаются попытки выполнить неверную или неопределенную команду либо осуществить обращение к несуществующей или защищенной памяти;
- для координации работы в ЭВМ;
- для профилактики, ремонта, тестирования и отладки системы.

5 Основная память компьютера

По функциональному назначению компьютерную память можно разделить на две части:

- на память, непосредственно взаимодействующую с центральным процессором при исполнении программы. Эта память строится на полупроводниковой основе, ее часто называют основной или внутренней памятью компьютера;

- на память, с которой процессор взаимодействует через контроллеры с помощью специальных программ (драйверов). К ней относится внешняя память, выполненная на магнитных и оптических накопителях. Эта память предназначена для длительного хранения данных.

Рассмотрим основные виды внутренней памяти компьютера.

5.1 Оперативная память

В компьютере предусматривается память для хранения программного обеспечения, содержащего пользовательские программы и подлежащие обработке данные, с целью оперативного обмена информацией (командами и данными) между процессором, внешней дисковой памятью и периферийными подсистемами (ввод-вывод, графика, коммуникации и т.п.). К характерным особенностям оперативной памяти следует отнести:

- данные и программы в памяти сохраняются только при включенном компьютере, т.е. она является временным хранилищем на один сеанс работы с компьютером, или энергозависимой памятью;

- поддерживает три режима работы: запись, хранение и считывание данных;

- высокие быстродействие и производительность;

- большой объем памяти, обеспечивающий хранение программного обеспечения;

- использование емкостных элементов памяти, позволяющих получить наивысшую плотность упаковки микросхем;
- для доступа к памяти выделяется адресное пространство, т.е. оперативная память входит в состав основной памяти компьютера;
- высокая надежность хранения данных, так как ошибка даже в одном бите может привести к неточным результатам, искажению или полной потере данных.

Существует много разновидностей оперативной памяти, отличающихся способами построения, технологическими и конструктивными особенностями.

5.2 Постоянная память

В компьютере предусматривается память для постоянного хранения программного обеспечения, содержащего вспомогательные подпрограммы базовой системы ввода-вывода и драйверы для тех плат (например, для видеоадаптера), которые активизируются на начальном этапе его запуска. К характерным особенностям постоянной памяти следует отнести:

- два основных режима работы запоминающего устройства – считывание и хранение данных. Так как данные постоянно хранятся в памяти, такую память часто называют постоянной памятью;
- запись данных в память (или программирование памяти) осуществляется один раз на достаточно длительный промежуток времени. Запись занимает больше времени, чем считывание, так как требует подачи специальных команд записи и верификации (проверки правильности заносимой информации);
- постоянная память является энергонезависимой памятью, т.е. любые записанные в нее данные сохраняются при выключении питания;
- для доступа к памяти выделяется часть адресного пространства, т.е. постоянная память (как и оперативная) входит в состав основной памяти компьютера.

Существует много разновидностей постоянной памяти (ROM, PROM, EPROM, EEPROM, Flash Memory, FRAM и др.), отличающихся способами построения и областями применения.

5.3 Иерархическая структура памяти

При выполнении программы процессор не должен простаивать, т.е. основная память должна постоянно обеспечивать его командами и данными.

Для уменьшения времени доступа к командам и данным в компьютерах применяется многоуровневая иерархическая структура полупроводниковой памяти:

- внутренняя регистровая (быстродействующая) память процессора, используемая как регистры общего назначения для хранения команд, данных, адресов и промежуточных результатов;
- до 3 уровней кэш-памяти со временем доступа 1..5 тактов. В кэше временно хранятся наиболее часто используемые фрагменты программного кода и обрабатываемых данных. При этом кэш первых двух уровней обычно находится на кристалле микропроцессора и работает с его же тактовой частотой, а кэш 3-го уровня располагается вне кристалла микропроцессора и строится на базе быстродействующих микросхем памяти;
- оперативная память, в качестве которой используется относительно медленная динамическая память.

В иерархической многоуровневой памяти блоки программ и данных между уровнями памяти пересылаются в те промежутки времени, когда предшествующие блоки обрабатываются процессором, что позволяет существенно сократить простои процессора. Сокращение времени простоя тем больше, чем больше время обработки данных, извлекаемых процессором из буферной памяти, по сравнению со временем пересылки данных между буферной и основной памятью. Наибольший эффект достигается, когда для получения результата процессор многократно использует одни и те же данные.

Буферная кэш-память организована как ассоциативная память, в которой данные содержатся в совокупности с их адресами. Кэш-память позволяет гибко согласовывать структуры данных, требуемых в динамике вычислений, со статическими структурами данных оперативной памяти.

5.4 Классификация запоминающих устройств

В вычислительных системах используется широкий набор полупроводниковых ЗУ, который разделяется на отдельные классы, группы, типы по ряду отличительных признаков.

По принципу действия выделяют:

- устройства на емкостных элементах памяти;
- устройства на триггерных элементах памяти.

По функциональному назначению устройства разделяют на две группы:

- оперативные ЗУ (ОЗУ), или RAM (Random Access Memory – ЗУ с произвольной выборкой), предназначенные для работы в условиях, когда необходимо производить выборку и обновление информации в высоком темпе работы процессора цифрового устройства. По способу хранения информации ОЗУ разделяют на две подгруппы:

- на статические ОЗУ, в которых состояние элементов памяти при хранении информации остается неизменным;

- на динамические ОЗУ, в которых состояние элементов памяти (обычно полупроводниковых емкостей) не остается неизменным и требует периодического проведения процесса регенерации (восстановления) исходных уровней сигналов.

- постоянные ЗУ (ПЗУ), или ROM (Read Only Memory – память только для чтения), предназначенные для многократной выдачи постоянно хранящейся в них информации (даже при отключении электропитания вычислительной системы). По способу занесения информации (программирования) в полупроводниковых ПЗУ различают:

- масочные ПЗУ, в которых перемычки в накопителе формируются в заводских условиях на заключительной стадии изготовления микросхем памяти;

- программируемые ПЗУ, в которых пользователь имеет возможность с помощью специального устройства (программатора) один раз осуществить пережигание плавких перемычек, исходя из собственной программы или кода;

- репрограммируемые ПЗУ, допускающие многократное перепрограммирование.

По способу доступа к информации устройства памяти подразделяют на три группы:

- адресные ЗУ, в которых доступ к ячейке памяти обеспечивается с помощью адресного кода, являющегося по сути дела номером ячейки. К ним относятся оперативные ЗУ (ОЗУ), или ЗУ с произвольной выборкой, и постоянные ЗУ, обеспечивающие непосредственный доступ к любой ячейке памяти с заданным адресом как при записи, так и при считывании. В адресных ЗУ все ячейки памяти равнодоступны;

- ЗУ с последовательным доступом, в которых осуществляется последовательный просмотр всех ячеек памяти. К этой группе ЗУ относят видеопамять, буфер FIFO (First-In First-Out – "первым пришел – первым ушел") и стек, или LIFO (Last-In First-Out – "последним пришел – первым ушел");

- ассоциативные ЗУ, в которых поиск и извлечение информации производятся не по ее расположению в памяти (адресу или месту в очереди), а по некоторому характерному признаку самой информации. Таким признаком может служить, например, выделенная совокупность разрядов (поле, тег) слова. Поле входного слова сравнивается с полями всех слов, хранящихся в памяти. Если поля совпадают, то слово считывается из памяти. Основная область применения ассоциативного доступа к данным – кэш-память.

По организации использования памяти в цифровых системах вы делают:

– внутренние ЗУ, к памяти которых процессор вычислительной системы непосредственно обращается за командами и данными. К внутренним ЗУ относятся ОЗУ и полупроводниковые ПЗУ;

– внешние ЗУ, к памяти которых за командами и данными процессор непосредственно обратиться не может. Для использования хранящейся во внешних ЗУ информации ее предварительно передают во внутренние ЗУ. К внешним ЗУ относятся дисковые ЗУ на флэш-памяти.

5.5 Постоянные запоминающие устройства

Постоянные запоминающие устройства, хранят информацию, которая не изменяется после изготовления микросхемы памяти или изменяется редко не в оперативном, а в специальном режиме программирования. В ПЗУ хранится предварительно занесенная информация в виде стандартных подпрограмм, кодов физических констант, постоянных коэффициентов, букв русского и латинского алфавита и т.п. Поэтому ПЗУ работают только в режимах хранения и считывания. Одним из основных узлов ПЗУ является накопитель, имеющий матричную структуру. Функции элементов памяти выполняют программируемые перемычки (в виде полупроводниковых диодов или транзисторов), включенные между строками и столбцами матрицы. Наличие перемычки соответствует, например, логической единице, а ее отсутствие – нулю. Это придает свойство энергонезависимости ПЗУ, т.е. сохранения информации при отключении питающих напряжений.

Содержащаяся в ПЗУ информация обычно является наиболее важной для функционирования компьютера, поэтому ее сохранность и предотвращение несанкционированного изменения – важнейшие требования, предъявляемые к постоянной памяти. Другими важными параметрами энергонезависимой памяти выступает длительность хранения и устойчивость к электромагнитным воздействиям, а для перепрограммируемой памяти еще и гарантированное количество циклов перепрограммирования.

Микросхемы ПЗУ имеют словарную организацию, при которой информация считывается в форме многоразрядного кода (слова). Совокупность элементов памяти в матрице накопителя, хранящих слово, называют ячейкой памяти. У каждой ячейки памяти есть свой адрес. Число элементов памяти в ячейке определяет ее разрядность N . Если число адресных входов составляет M , то число ячеек равно $2M$, а информационная емкость микросхемы – $2M \times N$ бит. Таким образом, для ПЗУ характерны многоразрядная (словарная) организация, режим считывания как основной режим работы и энергонезависимость. Вместе с тем выделенные в отдельные группы ПЗУ имеют свои особенности, что делает целесообразным рассмотрение устройств каждой группы отдельно.

5.5.1 Масочные ПЗУ

В масочные ЗУ типа ROM данные заносятся при изготовлении микросхем фотолитографическим способом с помощью маски (шаблона) на завершающем этапе технологического процесса. Пользователь не может изменить содержимое памяти. Элементом памяти или связи между строками и столбцами могут быть полупроводниковые диоды, биполярные транзисторы и МОП-транзисторы. В матрице с диодными элементами в узлах, символизирующих логическую единицу, изготавливается полноценный диод, а в узлах, символизирующих логический ноль, присутствует неработоспособный диод. Это позволяет использовать один шаблон при изготовлении микросхемы и удешевить производство. Для матриц с МОП-транзисторами при изготовлении транзисторов, символизирующих логический ноль, увеличивают толщину подзатворного слоя. В таких транзисторах рабочие напряжения ЗУ не в состоянии открыть транзистор, что равноценно его отсутствию.

Достоинство масочных ЗУ – высокий уровень интеграции, высокое быстродействие, низкая стоимость при массовом производстве. Однако в компьютерах микросхемы ROM широкого применения не получили ввиду невозможности модификации содержимого (такая модификация

осуществляется только путем изготовления новых микросхем). Раньше они применялись в качестве знакогенераторов в некоторых моделях графических адаптеров.

5.5.2 Программируемые ПЗУ (ППЗУ)

Программируемая память, или память типа PROM (Programmable ROM), позволяет пользователю устранить или создать перемычки в исходной заготовке микросхемы памяти. Этот процесс называется программированием микросхемы памяти.

В микросхемах с устранимыми плавкими перемычками в исходной заготовке имеются все перемычки. В процессе программирования часть из них расплавляется импульсами тока достаточно большой амплитуды и длительности. Плавкие перемычки обычно включаются в электроды диодов или транзисторов.

В микросхемах с создаваемыми перемычками элемент памяти в исходной заготовке представляет собой два встречно включенных диода или тонкие диэлектрические слои и имеет очень большое сопротивление. В процессе программирования на элемент памяти подается повышенное напряжение, в результате чего происходит пробой элемента с образованием короткого замыкания.

Так как микросхемы PROM программируются один раз, их часто называют микросхемами OTP (One Time Programmable). При изменении программы старая микросхема должна быть заменена новой.

Микросхемы PROM после изготовления содержат двоичные единицы (или нули). В процессе программирования, который часто называется прожигом и выполняется с помощью специального программирующего устройства, в нее записываются нули.

В зависимости от емкости памяти микросхемы и применяемого алгоритма процесс программирования PROM занимает от нескольких секунд до нескольких минут.

Микросхемы PROM, как и масочные, практически нечувствительны к электромагнитным полям. Несанкционированное изменение их содержимого в устройстве исключено.

5.5.3 Репрограммируемые ПЗУ

Известны две разновидности репрограммируемых ПЗУ:

- с электрическим стиранием информации типа EEPROM (Electrical Erasable PROM – PROM с электрическим стиранием);
- со стиранием информации ультрафиолетовым излучением типа EPROM (Erasable PROM – стираемое PROM).

Их отличие обусловлено особенностями элементов памяти накопителей.

Элементом памяти в РПЗУ является полевой транзистор. Эти транзисторы под воздействием программирующего напряжения способны записать электрический заряд под затвором и сохранять его много тысяч часов без напряжения питания. Для того чтобы перепрограммировать такое ПЗУ, необходимо сначала стереть записанную ранее информацию. В РПЗУ на МНОП транзисторах стирание производится электрическим сигналом, который вытесняет накопленный под затвором заряд. В РПЗУ на ЛИЗМОП транзисторах стирание записанной информации происходит под воздействием ультрафиолетового излучения, которое облучает кристалл через специальное окно в корпусе микросхемы.

РПЗУ со стиранием ультрафиолетовым излучением имеют ряд недостатков, по сравнению с РПЗУ со стиранием электрическим сигналом. Так, например, для стирания информации ультрафиолетом необходимо вынимать микросхему из контактных устройств (панелек), что не совсем удобно. К тому же, наличие окна в корпусе обуславливает чувствительность микросхемы РПЗУ к свету, что увеличивает вероятность случайного стирания информации. Да и число циклов перепрограммирования меньше чем, у РПЗУ со стиранием электрическим сигналом.

5.5.4 Флеш-память

Разновидность полупроводниковой технологии электрически перепрограммируемой памяти (EEPROM).

Благодаря компактности, дешевизне, механической прочности, большому объёму, скорости работы и низкому энергопотреблению, флеш-память широко используется в цифровых портативных устройствах и носителях информации. Серьёзным недостатком данной технологии является ограниченный срок работоспособности носителей, а также чувствительность к электростатическому разряду.

Принцип работы. Основным компонентом в флеш-памяти является транзистор с плавающим затвором, который является разновидностью МОП-транзисторов. Его отличие в том, что у него есть дополнительный затвор (плавающий), расположенный между управляющим затвором и р-слоем. Плавающий затвор изолирован и хранимый в нём отрицательный заряд будет оставаться надолго. Путем изменения его состояния кодируется один бит данных. Обычно наличие заряда (электронов) на затворе соответствует логическому 0, а его отсутствие – логической единице. В исходном состоянии плавающий затвор не содержит электронов.

Различают устройства, в которых элементарная ячейка хранит один бит информации и несколько бит. В однобитовых ячейках различают только два уровня заряда на плавающем затворе. Такие ячейки называют одноуровневыми (single-level cell, SLC). В многобитовых ячейках различают больше уровней заряда; их называют многоуровневыми (multi-level cell, MLC). MLC-устройства дешевле и более ёмки, чем SLC-устройства, однако имеют более высокое время доступа и примерно на порядок меньшее максимальное количество перезаписей.

Обычно под MLC понимают память с 4 уровнями заряда (2 бита) на каждую ячейку. Более дешёвую в пересчёте на объём память с 8 уровнями (3 бита) чаще называют TLC (Triple Level Cell) или 3bit MLC (MLC-3). Существуют

экспериментальные устройства с 16 уровнями на ячейку (4 бита), 16LC или QLC (quad-level cell).

К 2016 году многоуровневая память доминирует на рынке. Тем не менее SLC-изделия, несмотря на многократно меньшую ёмкость, продолжают разрабатываться и выпускаться для особо ответственных применений.

5.6 Статические ОЗУ

Статическая память (Static Random Access Memory – SRAM) способна сколько угодно долго хранить данные в отсутствие обращений (при наличии питающего напряжения), т.е. в статическом режиме. Ячейки статической памяти строятся на элементах с двумя устойчивыми состояниями (триггерах). По сравнению с динамическими емкостными элементами памяти они проще в управлении и не требуют регенерации, однако являются более сложными в схемном отношении и занимают больше места на кристалле. Быстродействие и энергопотребление статической памяти определяются технологией изготовления и схемотехникой запоминающих ячеек. Самая экономичная КМОП-память пригодна для длительного хранения данных при питании от маломощной батареи. Она применяется в памяти конфигурации персональных компьютеров. Самая быстродействующая статическая память способна функционировать на частоте системной шины совместно с процессором, не требуя от него тактов ожидания.

Существуют три разновидности микросхем статической памяти: Async SRAM, Sync Burst SRAM и Pipelined Burst SRAM. Относительно высокая удельная стоимость хранения данных при низкой плотности упаковки не позволяет использовать SRAM в качестве основной памяти компьютеров.

Во избежание увеличения стоимости в компьютерах устанавливается небольшой объем высокоскоростной статической памяти SRAM, которая используется в качестве кэша. Кэш-память способна работать на тактовых частотах, близких или равных тактовым частотам процессора. Поэтому она непосредственно используется процессором при чтении и записи, что позволяет

сократить количество его простоев и увеличить быстродействие компьютера в целом. Контроллер кэша предугадывает потребности процессора в данных и предварительно загружает необходимые данные в высокоскоростную кэш-память. При выдаче процессором адреса памяти данные передаются не из медленной оперативной памяти, а из кэша.

Для сокращения времени ожидания и простоев процессора при считывании данных из низкоскоростной оперативной памяти в современных компьютерах предусмотрено до трех уровней кэша. При этом кэш-память первого и второго уровней может располагаться на одном кристалле с процессором. Использование синхронной работы с процессором и конвейерного пакетного режима способствует повышению быстродействия и эффективности кэш-памяти. Возможности и эффективность кэш-памяти предопределяет контроллер, который располагается в микросхемах (обычно North Bridge) системной логики или на плате процессора.

Таким образом, к основным особенностям статических ОЗУ следует отнести:

- способность при включенном компьютере сколь угодно долго хранить данные (информацию) в отсутствие обращений. Такая способность обеспечивается бистабильными ячейками памяти с двумя устойчивыми состояниями, которые выполняются на биполярных или КМОП-структурах;
- сравнительно высокое быстродействие микросхем на биполярных структурах, позволяющее работать синхронно с процессорами;
- низкое энергопотребление КМОП-микросхем, обеспечивающее длительное хранение параметров системы ввода-вывода;
- сравнительно большие габариты микросхем и высокая стоимость, что связано с большим числом транзисторов и кластеризованным их размещением;
- основная область применения – кэш-память и память конфигурации компьютера.

5.6.1 Основные виды микросхем статической памяти

Асинхронная статическая память (Async SRAM или просто SRAM). Интерфейс микросхем этой памяти включает шину адреса, шину данных и следующие сигналы управления:

- сигнал выборки кристалла (Chip Select – CS), активизирующий микросхему при $CS\# = 0$, где # – знак инверсии. При объединении микросхем внутренний сигнал CS# может быть получен из нескольких внешних (например, CS0#, CS1# и CS2#) с использованием схемы И;
- сигнал разрешения вывода (Output Enable – OE), открывающий при $OE\# = 0$ выходные буферы для считывания данных;
- сигнал разрешения записи WE# (Write Enable – WE) при $WE\# = 0$. При операции записи управление выходными буферами может производиться как сигналом OE#, так и сигналом WE#.

При использовании статической памяти в кэше процессор посылает адрес, после чего кэш производит поиск адреса и передает требуемые данные. В начале каждого обращения, как правило, используется дополнительный цикл для просмотра тегов.

Синхронная пакетная статическая память (Sync Burst SRAM) предназначена для пакетных операций обмена в кэш-памяти. Микросхема памяти содержит дополнительно счетчик адреса, не позволяющий перейти границу пакетного цикла, а помимо сигналов CS#, OE# и WE# для синхронизации с системной шиной используются сигнал CLC (Clock) и сигналы управления пакетным циклом, при этом:

- сигналы CADS# (Cache Address Strobe) и ADSP# (ADdress Status of Processor) являются стробами записи начального адреса цикла во внутренний регистр адреса. Этими сигналами процессор или кэш-контроллер отмечает фазу адреса очередного цикла. Любой из них инициирует одиночный или пакетный цикл обращения;

– сигнал ADV# (ADVance) используется для перехода к следующему адресу пакетного цикла.

Все сигналы, кроме сигнала управления выходными буферами OE#, синхронизируются по положительному перепаду сигнала CLC. Микросхемы Sync Burst SRAM обычно имеют сигнал, выбирающий режим счета адреса: чередование (для процессоров Intel) или последовательный счет (для Power PC).

Конвейеризированная пакетная память (Pipelined Burst SRAM – PB SRAM), поддерживающая конвейерный режим выполнения групповых операций чтения/записи, является дальнейшим усовершенствованием синхронной памяти. Конвейеризация реализуется добавлением выходного буфера, в который помещаются считанные из ячеек памяти данные. Буфер обеспечивает более быстрый доступ к данным при последовательных обращениях по чтению из памяти, так как не требуется обращение к матрице памяти для получения следующего элемента данных. Буферный регистр использует дополнительные такты только в пересылке первого элемента данных пакета, остальные элементы следуют без тактов ожидания.

5.7 Динамические ОЗУ

Название "динамические ОЗУ" обусловлено элементами памяти, в качестве которых используются конденсаторы небольшой емкости, способные хранить заряд. В реальных условиях конденсатор разряжается и требуется постоянная периодическая его подзарядка. Поэтому память на основе емкостных элементов является динамической памятью, чем она принципиально отличается от статической памяти, реализуемой на бистабильных ячейках, способных хранить информацию при включенном питании сколь угодно долго. Таким образом, динамическое хранение данных означает прежде всего возможность многократной записи информации в оперативную память, а также необходимость периодического обновления, или перезаписи данных.

При использовании емкостных элементов памяти удастся на одном кристалле размещать миллионы ячеек и получать самую дешевую полупроводниковую память достаточно высокого быстродействия с умеренным энергопотреблением. Благодаря этому динамические ОЗУ являются основной оперативной памятью компьютера.

Идеальный конденсатор представляет собой двухполюсник, заряд которого Q является линейной функцией напряжения U . Если к идеальному конденсатору C через ключ K подвести напряжение U от источника ЭДС, то на конденсаторе появится постоянный заряд Q . При постоянстве заряда ($Q = \text{const}$) ток в цепи не протекает, поэтому размыкание ключа не изменит состояния конденсатора C , т.е. на конденсаторе по-прежнему останутся $Q = \text{const}$ и $U = \text{const}$. Следовательно, конденсатор обладает способностью хранить заряд Q и напряжение U .

Реальные конденсаторы обладают потерями, кроме того, для реализации режимов записи и считывания к конденсаторам подключаются внешние цепи, которые также имеют потери. В процессе разряда конденсатор теряет свой заряд, и напряжение на его полюсах уменьшается. Поэтому, как отмечалось выше, использование конденсаторов в качестве элементов памяти требует периодического восстановления (регенерации) напряжения.

Основой для построения емкостных элементов памяти служат МОП-транзисторы. В настоящее время широкое распространение получили однотранзисторные структуры, которые помимо емкостного элемента памяти располагают средством подключения его к разрядной шине. Однотранзисторный емкостной элемент памяти проще элемента памяти статического ОЗУ. Благодаря тому, что на кристалле удастся разместить больше элементов памяти, динамические ОЗУ имеют значительно большую емкость памяти, чем их статические аналоги.

5.7.1 Работа элемента памяти в динамическом ОЗУ

Использование емкостных элементов памяти в ЗУ отражается на структуре накопителя. Помимо элементов памяти накопитель содержит дополнительные узлы и компоненты, обеспечивающие необходимые условия для нормального его функционирования.

При отсутствии напряжения на адресной шине транзистор заперт и конденсатор элемента памяти отключен от разрядной шины. Элемент памяти работает в режиме хранения.

При поступлении напряжения на адресную шину и, следовательно, на затвор транзисторного ключа элемент памяти подсоединяется к разрядной шине. В зависимости от значения сигнала чтения/записи возможно два режима работы емкостного элемента памяти.

В режиме записи с помощью управляющих сигналов, подаваемых на затворы транзисторных ключей, в элемент памяти можно записать соответственно логические нуль или единицу. При этом логическому нулю соответствует нулевое значение напряжения на конденсаторе, логической единице – напряжение, равное питанию.

В режиме считывания в силу большой протяженности разрядной шины и большого числа различных элементов, подключенных к ней, шина обладает емкостью, многократно превышающей емкость элемента памяти. Для считывания информации с разрядной шины при подключении к ней емкостного элемента памяти необходимо располагать точным значением напряжения на шине. Поэтому перед считыванием на разрядную шину подается фиксированное напряжение, равное напряжению источника питания, для подзаряда емкости. После этого элемент памяти подключается к разрядной шине.

В результате:

- при считывании на элементе памяти происходит падение напряжения, т.е. считывание является разрушающим процессом и требует восстановления исходной информации;

- напряжение на разрядной шине в режиме считывания изменяется в незначительных пределах, что затрудняет точную фиксацию хранимых в элементе памяти данных.

Для преодоления указанных недостатков принимают следующие меры:

- для восстановления заряда элемента памяти вводят циклы регенерации;
- увеличивают емкость элемента памяти;
- уменьшают емкость шины в два раза путем деления ее на две подшины;
- для считывания используют высокочувствительные дифференциальные усилители с положительной обратной связью – усилители-регенераторы.

5.7.2 Асинхронная и синхронная DRAM

Недостатки базовой динамической памяти RAM – низкие быстродействие и производительность – обусловлены принятой организацией обращения к памяти, основанной на использовании двух сигналов. Любой цикл обращения (чтения/записи) начинается после подачи стробов выборки адреса строки $RAS\# = 0$ и столбца $CAS\# = 0$. Так как стробы поступают последовательно во времени, занесение данных в память при записи и появление их на выходной шине данных при считывании происходит задержкой, связанной с ожиданием появления стробов. Поэтому при разработке новых типов (серий микросхем) динамической памяти основные усилия были направлены на повышение ее быстродействия и производительности.

В асинхронной памяти все процессы инициируются стробами выборки адресов строк и столбцов ($RAS\# = 0$ и $CAS\# = 0$). Их завершение происходит через определенный для данной микросхемы временной интервал. В течение всего временного интервала шина памяти занята, причем значительная часть времени проходит в ожидании данных.

Память типа FPM DRAM. Одним из путей сокращения времени ожидания является страничная организация памяти, при которой память разбивается на страницы длиной от 512 байт до нескольких килобайтов. При обращении к ячейкам памяти в пределах страницы (из-за ограниченного ее объема) уменьшается количество состояний ожидания. Если нужная ячейка памяти находится вне текущей страницы, то добавляется одно состояние ожидания для выбора новой страницы. В динамической памяти FPM DRAM используется быстрый страничный режим (Fast Page Mode – FPM). Повышение производительности в этом режиме достигается исключением фазы выдачи адреса строки из всех последующих циклов обращения, кроме первого. Быстрый страничный режим обеспечивается контроллером динамической памяти и может быть реализован в микросхемах базовой памяти DRAM. Обычно в этом режиме производится обращение к четырем соседним ячейкам. Дальнейшее повышение производительности памяти может быть достигнуто путем чередования банков.

Память типа EDO DRAM (Extended Data Output – память с расширенной выдачей данных). Основные особенности этого вида памяти по сравнению с FPM DRAM состоят в следующем:

- если линии ввода-вывода данных в памяти FPM DRAM отключаются от системной шины с появлением адреса следующего бита, то в памяти EDO DRAM они остаются подключенными до окончания ввода нового адреса и соответственно начала вывода следующего бита. Следовательно, память EDO позволяет одновременно считывать данные и задавать адрес следующих данных. Считывание выходных данных может производиться внешними схемами вплоть до появления следующего строба $CAS = 0$. Благодаря этому время цикла (длительность стробов CAS) внутри страницы для памяти уменьшается;

– память EDO содержит регистр-защелку выходных данных, который обеспечивает конвейеризацию работы и повышение производительности при чтении.

Память типа BEDO DRAM. Этот тип памяти (Burst EDO – BEDO) является разновидностью EDO DRAM. В микросхемы BEDO помимо регистра-защелки выходных данных введен внутренний счетчик адреса столбцов для пакетного цикла. Это позволяет выставлять адрес столбца в начале пакетного цикла, а в последующих передачах с помощью стробов CAS только запрашивать очередные данные. В результате удлинения конвейера (регистр-защелка + счетчик) выходные данные дополнительно отстают на один такт сигнала CAS. Последующие данные появляются без тактов ожидания процессора.

Все рассмотренные типы памяти (FPM, EDO, BEDO DRAM) для передачи данных из памяти в центральный процессор по современным высокочастотным системным шинам требуют подтверждения готовности. Если данные не готовы, то процессор вынужден осуществлять холостые циклы в ожидании данных. Поэтому асинхронная память в настоящее время не используется. Ее вытеснила синхронная память DRAM, которая стала новым стандартом оперативной памяти персональных компьютеров.

5.7.3 Синхронные динамические ОЗУ

В синхронной памяти все процессы при выполнении операций записи и чтения данных согласованы во времени с тактовой частотой центрального процессора (или системной шины), т.е. память и центральный процессор работают синхронно без циклов ожидания. Информация передается в пакетах, использующих высокоскоростной синхронизированный интерфейс.

Память типа SDRAM. Рассмотрим основные особенности синхронной динамической памяти SDRAM.

Состав и назначение сигналов. В состав сигналов синхронной памяти входят сигналы RAS#, CAS#, WE#, MA#, которые выполняют те же функции,

что и в асинхронной динамической памяти. Помимо приведенных сигналов используются сигналы, свойственные только динамической памяти SDRAM. К ним относятся:

- CLK (Clock) – тактовые импульсы синхронизации, действующие по переднему фронту;

- CKE (Clock Enable) – разрешение/запрещение синхронизации при CKE=1/0. Отсутствие синхроимпульсов уменьшает энергопотребление памяти. Переход в режим с пониженным энергопотреблением осуществляется с помощью специальных команд при CKE=0. Следует выделить три режима:

- режим пониженного потребления (Power Down Mode), реализующийся командами NOP или INHBT. В этих режимах микросхема памяти не воспринимает управляющих команд. Длительность пребывания в них ограничена периодом регенерации;

- режим приостановки синхронизации (Clock Suspend Mode), в котором отсутствует передача данных и не воспринимаются новые команды. В этот режим микросхема переходит во время выполнения команды чтения или записи при установке сигнала CKE=0;

- режим саморегенерации, в который микросхема переходит по команде Self Refresh. В этом режиме периодически выполняются циклы регенерации по внутреннему таймеру при отключенной внешней синхронизации;

- CS# (Chip Select) – выбор микросхемы. При CS# = 0 разрешается декодирование команд; при CS# = 1 декодирование команд запрещается, однако выполнение начатых команд продолжается;

- BSO, BSl (Bank Select) или BA0, BA1 (Bank Address) – выбор банка, к которому адресуется команда;

- Д[0:12] (Address) – сигналы мультиплексированной шины адреса. В циклах Bank/Activate задают адрес строки. В циклах Read/Write сигналы линий Л[0:9] и A1 задают адрес столбца, сигнал A10 = 1 включает режим

автопредзаряда. В циклах Precharge сигнал A10=1 включает режим предзаряда всех банков независимо от значений сигналов;

- DQ (Data Input/Output) – двунаправленные линии ввода-вывода данных;

- DQM (Data Mask) – маскирование данных. В цикле чтения при DQM= 1 шина данных через два такта отключается. В цикле записи при DQM – 1 запрещается запись текущих данных, при DQM = 0 разрешается запись без задержки.

Микросхемы SDRAM располагают двумя или более банками, а также счетчиками адреса столбцов. К достоинствам синхронного интерфейса SDRAM следует отнести то, что в сочетании с внутренней мультибанковой организацией он способен обеспечить высокую производительность памяти при частых обращениях.

В памяти SDRAM имеется возможность активизации строк в нескольких банках. Каждая строка активизируется своей командой ACT во время выполнения любой операции с другим банком. После активизации строки выбранного банка при записи и чтении строку можно закрывать не сразу, а после выполнения серии обращений к ее элементам. Для обращения к открытой строке требуемого банка используются команды чтения RD и записи WR, в которых указаны адрес столбца и номер банка. Можно так организовать процессы записи/чтения, что шина данных в каждом такте будет нести очередную порцию данных для серии обращений к разным областям памяти. Так как для обращений не требуются команды активизации, они будут выполняться быстрее. С помощью сигнала выборки микросхемы CS# можно держать открытыми строки в банках разных микросхем, объединенных общей шиной памяти.

С помощью счетчика весьма просто реализуется пакетный режим работы. При инициализации могут быть запрограммированы длина пакета (1, 2, 4, 8 элементов), порядок адресов в пакете (чередующийся или линейный) и операционный режим (пакетный режим для всех операций или только для

чтения). С помощью сигнала $DQM = 1$ в режиме записи осуществляется блокирование записи любого элемента пакета, а в режиме чтения – перевод в высокоимпедансное состояние буфера данных.

Благодаря исключению циклов ожидания, чередованию адресов, пакетному режиму, трехступенчатой конвейерной адресации удалось сократить время рабочего цикла микросхемы до 8...10 нс (1: 10 нс = 100 МГц) и повысить скорость передачи данных до 800 Мбайт/с при тактовой частоте системной шины 100 МГц.

Память типа DDR SDRAM (Dual Data Rate – удвоенная скорость данных). Основная особенность памяти DDR по отношению к обычной SDRAM состоит в том, что переключение данных производится по фронту и срезу тактовых импульсов системной шины. Это дает возможность выполнить два обращения за тактовый интервал и повысить быстродействие в два раза. При передаче данных по фронту и срезу импульсов синхронизации к временным параметрам управляющих сигналов и данных предъявляются повышенные требования. Для их удовлетворения приняты следующие меры:

- введен стробирующий сигнал DQS;
- используются два синхроимпульса CLK1 и CLK2, а также дополнительные аппаратные средства.

В отличие от обычных микросхем SDRAM, у которых данные для записи передаются одновременно с командой, в DDR SDRAM данные для записи подаются с задержкой на один такт (Write Latency). Значение CAS Latency может быть дробным ($CL = 2, 2,5, 3$).

На частоте 100 МГц DDR SDRAM имеет пиковую производительность 200 Мбит на один вывод, что в составе 8-байтных модулей DIMM соответствует производительности 1600 Мбайт/с. На частоте 133 МГц производительность составляет 2100 Мбайт/с.

Память типа RDRAM. В 1992 г. американская фирма Rambus приступила к разработке нового типа памяти, которая получила название RDRAM (Rambus

DRAM). Запоминающее ядро этой памяти построено на обычных КМОП-ячейках динамической памяти. Однако интерфейс памяти существенным образом отличался от традиционного синхронного интерфейса. Высокоскоростной интерфейс Rambus RDRAM обеспечивает возможность передачи данных со скоростью до 600 Мбайт/с через шину данных разрядностью 1 байт. Эффективная пропускная способность достигает величины 480 Мбайт/с, что в 10 раз превышает аналогичный показатель для устройств EDO DRAM. Время доступа к ряду ячеек памяти составляет менее 2 нс в расчете на байт, а время задержки (время доступа к первому байту массива данных) – 23 нс. При обмене большими массивами данных память Rambus является оптимальным вариантом в смысле отношения производительность/стоимость. Дальнейшим развитием стал интерфейс Direct DRAM, или просто DDRAM, с 16-разрядной (18-разрядной для микросхем с битами контроля) шиной данных. Память RDRAM используется в высокопроизводительных персональных компьютерах с 1999 г. и поддерживается в наборах микросхем системной логики.

Память RDRAM по отношению к другим типам памяти (FPM/EDO и SDRAM) имеет следующие отличительные особенности:

- является устройством с узким каналом передачи данных. Количество данных, передаваемых за один такт, составляет всего 16 бит, не считая двух дополнительных битов контроля по четности;

- благодаря небольшому числу (30) линий канала и специально принятым мерам по их расположению тактовая частота канала увеличена до 400 МГц, что обеспечивает производительность, равную $16 \times 400 \times 2/8 = 1600$ Мбайт/с (с учетом передачи данных по фронту и срезу синхроимпульсов). Для повышения производительности можно использовать двух- и четырехканальные RDRAM, которые позволяют увеличить скорость передачи данных до 3,2 или 6,4 Мбайт/с соответственно;

– передача адреса ячейки происходит по отдельным шинам: одна – для адреса строки, другая – для адреса столбца. Передача адресов осуществляется последовательными пакетами. В процессе работы RDRAM выполняется конвейерная выборка из памяти, причем адрес может передаваться одновременно с данными;

– для повышения производительности было предложено еще одно конструктивное решение: передача управляющей информации отделена от передачи данных по шине. Для этого предусмотрены независимые схемы управления и выделены две группы шин: адресные шины для команд выбора строки и столбца и информационная шина для передачи данных шириной 2 байта;

– потребляет мало энергии. Напряжение питания модулей памяти достигает 2,5 В. Напряжение низковольтного сигнала изменяется от 1,0 до 1,8 В, т.е. перепад напряжений равен 0,8 В. Кроме того, RDRAM имеет четыре режима пониженного потребления энергии и может автоматически переходить в режим ожидания на завершающей стадии транзакции, что позволяет еще больше экономить потребляемую мощность.

Память с виртуальными каналами – VC SDRAM. В компьютере доступ к оперативной памяти осуществляется различными устройствами. Одни из устройств (программы, которые выполняются параллельно в многозадачной операционной системе) бронируют для себя определенные области памяти. Такие устройства, как процессор, IDE- и SCSI-контроллеры, звуковые карты и видеокарты и другие, обращаются к оперативной памяти напрямую. При одновременном обращении к памяти нескольких устройств происходит задержка их обслуживания. Для устранения этого недостатка была разработана специальная архитектура модуля памяти, включающая в себя 16 независимых каналов памяти. Каждому устройству (программе) для обращения к памяти выделен отдельный канал.

Особенность архитектуры памяти с виртуальными каналами (Virtual Channel Memory Architecture) состоит в том, что между массивом запоминающих ячеек и внешним интерфейсом микросхемы памяти размещено 16 канальных буферов. В виртуальные каналы могут объединяться несколько буферов. По составу и уровням сигналов микросхемы VC SDRAM (Virtual Channel SDRAM) аналогичны обычным SDRAM (имеют внешнюю организацию по 4, 8 или 16 бит данных), однако отличаются структурой, системой команд и рядом других показателей. Микросхема содержит два банка (А и В), выполненных в виде квадратной матрицы. Каждая строка матрицы разбивается на 4 сегмента. Для микросхемы емкостью 128 Мбит размер матрицы составляет 8К x 8К, строка имеет объем 8К бит, а сегмент – 2К бит. Емкость канального буфера также составляет 2К бит. За одно обращение к матрице выполняется параллельная передача 2К бит данных между одним из буферов и сегментом выбранной строки. Микросхемы устанавливаются в 168-контактный модуль DIMM.

Операции обмена данными разделяются на две фазы:

- внешний обмен данными между источником информации и канальным буфером. Эта фаза обмена осуществляется через контроллер памяти и выполняется по командам чтения и записи (READ и WRITE), в которых указывается номер канала и адрес столбца. Обмен происходит в пакетном режиме. Длина пакета программируется и может составлять 1, 2, 4, 8 или 16 передач (элементов). Первые данные при чтении канала появляются с задержкой в 2 такта относительно команды чтения, следующие идут в каждом такте;

- внутренний обмен данными между каналами и массивом запоминающих ячеек. Обмен протекает в такой последовательности:

- с помощью команд предвыборки PRFA и сохранения RSTA, поступающих сразу после обращения к массиву памяти, автоматически осуществляется деактивизация строк (предварительный заряд). Для

деактивизации выбранного банка и обоих банков сразу можно использовать специальные команды;

- по команде ACT, которая задает банк (А или В) и адрес строки, активизируется требуемая строка матрицы;

- командами PRF (Prefetch) и RST (Restore) реализуется чтение массива в буфер и сохранение данных буфера в массиве. В командах указывается номер банка, номер сегмента и номер канала.

Обе фазы обмена выполняются по командам со стороны внешнего интерфейса почти независимо друг от друга.

DDR2 SDRAM – тип DRAM, основанный на DDR SDRAM и выпущенный в 2004 году. Память такого типа по сравнению с DDR SDRAM за счёт технических изменений обладала более высоким быстродействием. Работала на тактовых частотах шины 200, 266, 333, 337, 400, 533, 575 и 600 МГц. При этом эффективная частота передачи данных могла составлять 400, 533, 667, 675, 800, 1066, 1150 и 1200 МГц. Некоторые производители модулей памяти, помимо модулей, работающих на стандартных частотах, выпускали модули, работающие на нестандартных (промежуточных) частотах; такие модули предназначались для использования в разогнанных системах, где требовался запас по частоте.

DDR3 SDRAM – тип DRAM, основанный на DDR2 SDRAM, отличающийся удвоенной частотой передачи данных по шине памяти и пониженным энергопотреблением. Память такого типа обеспечивает большую пропускную способность по сравнению с ранее существовавшими типами памяти. Работает на частотах полосы пропускания в пределах от 800 до 2400 МГц (рекорд частоты – более 3000 МГц).

DDR4 SDRAM – тип DRAM, основанный на технологиях предыдущих поколений DDR и отличающийся повышенными частотными характеристиками, пониженным напряжением питания.

Основное отличие DDR4 от предыдущего стандарта (DDR3) заключается в удвоенном до 16 числе банков (в двух группах банков, что позволило увеличить скорость передачи). Пропускная способность памяти DDR4 в перспективе может достигать 25,6 ГБ/с (в случае повышения максимальной эффективной частоты до 3200 МГц). Надёжность работы DDR4 повышена за счёт введения механизма контроля чётности на шинах адреса и команд. Изначально в стандарте DDR4 был определён диапазон частот от 1600 до 2400 МГц с возможностью увеличения до 3200 МГц.

Заключение

Архитектура ЭВМ отражает концептуальную структуру вычислительной машины и отличительные черты в реализации.

По способу детализации различают архитектуру всей ЭВМ и архитектуру различных узлов ЭВМ: архитектура микропроцессора, памяти и т.д.

Также структура ЭВМ определяет способы адресации команд и данных, которые рассмотрены в третьей главе.

Отдельное внимание в пособие уделено организации основной памяти ЭВМ.

Материал, изложенный в учебном пособии, понадобится студентам для изучения таких дисциплин, как «Сети и телекоммуникации» и «Машинно-зависимые языки».

Список литературы

1. Новожилов, О. П. Архитектура ЭВМ и систем : учебное пособие для академического бакалавриата / О. П. Новожилов. — М. : Издательство Юрайт, 2018. — 527 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-02626-9. — Режим доступа : www.biblio-online.ru/book/C6CCB2DB-DD82-45E0-916D-B632CC9F39A9.
2. Мартиросян С.Т., Электронно-вычислительные машины и компьютерные сети. Учебное пособие по курсу «Организация ЭВМ и систем», для студентов специальности 220100 – Москва, МГИЭМ, 2011 – 552 с.
3. Жмакин А. П. Архитектура ЭВМ. — СПб.: БХВ-Петербург, 2006. — 320 с.
4. Авдеев, В.А. Организация ЭВМ и периферия с демонстрацией имитационных моделей [Электронный ресурс] : учебное пособие / В.А. Авдеев. — Электрон. дан. — Москва : ДМК Пресс, 2014. — 708 с. — Режим доступа: <https://e.lanbook.com/book/58704>. — Загл. с экрана.
5. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.

Электронное учебное издание

Алексей Александрович **Силаев**

Архитектура ЭВМ

Учебное пособие для ВУЗов

Электронное издание сетевого распространения

Редактор Матвеева Н.И.

Темплан 2018 г. Поз. № 47.

Подписано к использованию 15.10.2018. Формат 60х84 1/16.

Гарнитура Times. Усл. печ. л. 4,69.

Волгоградский государственный технический университет.
400005, г. Волгоград, пр. Ленина, 28, корп. 1.

ВПИ (филиал) ВолгГТУ.
404121, г. Волжский, ул. Энгельса, 42а.