МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

В.В. ИВАНОВ

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для обучающихся по основной образовательной программе высшего образования по направлению подготовки 11.03.03 Конструирование и технология электронных средств

С А М А Р А
Издательство Самарского университета
2019

УДК 004(075)
ББК 32.97я7
И20

Рецензенты: д-р техн. наук, проф. А. И. Д а н и л и н;
д-р техн. наук О. В. Г о р я ч к и н

Иванов, Владимир Васильевич

И20 **Микропроцессорная техника** : учеб. пособие / *В.В. Иванов.* – Самара: Изд-во Самарского университета, 2019. – 80 с.: ил.

ISBN 978-5-7883-1422-8

Учебное пособие охватывает разделы по архитектуре, системам команд, алгоритмам обращения к периферийным устройствам микроЭВМ, микропроцессоров и построенной на их основе техники.

Предназначено для студентов направления подготовки 11.03.03 Конструирование и технология электронных средств при изучении курса «Микропроцессорная техника».

Подготовлено на кафедре КТЭСиУ.

УДК 004(075)
ББК32.97я7

ISBN 978-5-7883-1422-8

ОГЛАВЛЕНИЕ

Глава 1. АРХИТЕКТУРА МИКРОЭВМ

1.1. Типовая структура микроЭВМ

Структура или архитектура – это, как говорит толковый словарь по вычислительным системам [2], описание (цифровой) вычислительной системы на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд и средств пользовательского интерфейса, организации памяти и системы адресации, операций ввода-вывода и управления и т. д. Реализация конкретной архитектуры на машинах данного семейства (*computer fami1y)* может быть различной, но все машины одного семейства должны быть способны выполнять одну и ту же программу. Реализации могут отличаться одна от другой как на уровне физических компонентов аппаратных средств, так и на уровне способов реализации подсистем. Например, может применяться микропрограммирование (*microprogramming)* вместо жесткой «зашитой» логики; однако чаще всего различия имеют место на обоих уровнях. Те или иные реализации могут существенно отличаться по производительности и стоимости. Детали реализации, невидимые для пользователя (*transparent*) (например, кэш-память), не оказывают влияния на архитектуру. Общность архитектуры разных ЭВМ обеспечивает их совместимость с точки зрения пользователя.

В контексте разработки вычислительной системы и проектирования ее аппаратных средств термин «архитектура» используется для описания принципа действия, конфигурации и взаимного соединения основных логических узлов ЭВМ (вследствие чего термин «архитектура» оказывается ближе к обыденному значению этого слова). Аппаратные средства – это, как правило, запоминающее устройство и его компоненты управления с аппаратурой, предназначенной для реализации требуемой стратегии управления, устройства, обеспечивающие нужную структуру, разрядность и функциональные возможности АЛУ, а также требуемые соединения входов и выходов (например, в форме звезды или общей шины). Описание архитектуры, кроме того, должно включать в себя разъяснение принципа действия и диапазона возможностей любого канального контроллера. Обычно частью, а зачастую и основой такого описания служит подробная структурная или принципиальная схема конкретной реальной, а не виртуальной машины.

1.2. Магистрально-модульный принцип построения ЭВМ

Одно ведущее устройство (master) передаёт по магистрали адрес ведомого устройства (slave), подключаемого к магистрали, и команду для него. По это команде выбранное устройство либо принимает данные с магистрали (режим записи), либо выдаёт на магистраль информацию (режим чтения). В подавляющем большинстве случаев ведущим устройством является процессор.

Каждый двоичный разряд кода адреса и кода данных передаётся по своему проводу. Все провода, по которым передаётся адрес, образуют шину адреса. Соответственно шина данных состоит из проводов, передающих данные. По проводам шины управления передаются команды и синхронизирующие сигналы. То есть полноразмерная магистраль состоит из шины адреса, шины данных и шины управления.

В микропроцессоре Intel 8086 шина данных и шина адреса использовали одни и те же контакты на корпусе. Вначале передаётся адрес, затем данные. При работе с внешней памятью микроконтроллеры семейства MCS-51 записывают в порт P0 младший байт адреса. Данные записываются или считываются позже через тот же порт. Такое решение называется «мультиплексирование по времени».

Мультиплексирование по времени позволяет сократить число проводов.

Скорость передачи уменьшается в число раз уменьшения количества проводов.

В предельном случае магистраль вырождается в два провода. По одному последовательно передаются адреса и данные, по второму – синхронизирующие импульсы, подтверждающие приход очередного импульса информации. Когда по первому проводу подряд идут несколько логических единиц или нулей, это очень важно.

Асинхронный режим приёма-передачи позволяет обойтись одним проводом. Однако скорость обмена снижается, так как к передаваемому байту добавляется, по крайней мере, два бита: стартовый и стоповый. Скорости передатчика и приёмника должны быть одинаковыми и высокостабильными.

В микроконтроллерах в асинхронном режиме работает последовательный порт UART (универсальный асинхронный приёмопередатчик).

1.3. Арифметико-логическое устройство

Операцию @ над двумя числами X и Y можно представить выражением:

$$X @ Y = Z.$$

X называют первым операндом, Y – вторым операндом, Z – результатом. Операция @ это сложение, вычитание, логические И, ИЛИ, исключающее ИЛИ, реже – умножение и деление.

Машинный код команды должен содержать код операции и три адреса: адрес первого операнда, адрес второго операнда и адрес результата. Трехадресные процессоры состоят из большого числа узлов, поэтому сложны и дороги.

Двухадресные команды сокращают длину команды и упрощают конструкцию процессора. Результат в двухадресных процессорах записывается по адресу первого операнда. Это сокращает количество элементов в процессоре.

В состав одноадресных процессоров входит специальный регистр, который называется аккумулятором либо рабочим регистром (например, в PIC контроллерах). В аккумуляторе расположен первый операнд для одноадресной команды. Одноадресная команда содержит только адрес второго операнда. Результат записывается в аккумулятор. Одноадресные процессоры проще других. Однако программа удлиняется из-за постоянного извлечения результата предыдущей команды из аккумулятора и записи первого операнда новой команды в аккумулятор.

В ЭВМ результат операции формирует арифметико-логическое устройство (рис. 1.1). Оно построено на формальной логике и имеет три входа и два выхода.

По шине данных нельзя передать два операнда одновременно. Поэтому на входах АЛУ устанавливают два буферных регистра (БР).

В начале цикла в один БР по шине данных записывается первый операнд, затем в другой – второй операнд. На третий вход подаётся код операции из регистра команд. В регистре команд хранится полученная процессором инструкция (команда).

Кроме результата АЛУ формирует признаки результата операции. Часто признаки называют флажками или флагами. Первыми в ряду признаков стоят два флажка – результат равен нулю и перенос со старшего значащего бита в старшие регистры результата (признак переполнения регистра). Признаки записываются в регистр признаков и используются в командах ветвления программы.

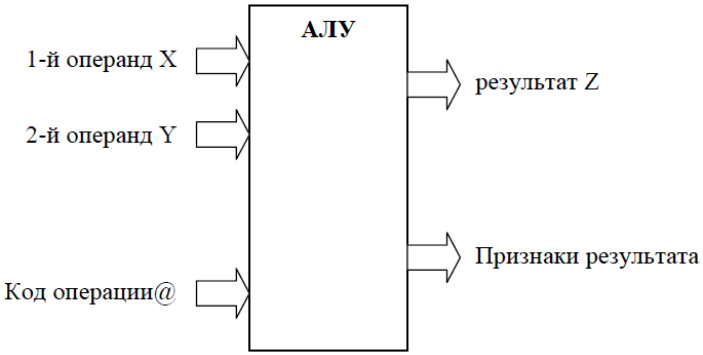

Рис. 1.1. Арифметико-логическое устройство

1.4. Структура вычислительной машины

Главными узлами ЭВМ являются процессор и память. Функционально память делят на память программ и память данных. Адрес ячейки памяти, в которой хранится код очередной команды, формирует процессор с помощью счетчика команд (программного счетчика). Полученный из памяти код команды хранится в регистре команд процессора. Значения операндов для выполнения команды процессор берёт из памяти данных. Выполнение команды заканчивается записью результата в память данных. Несколько ячеек памяти данных расположены непосредственно в процессоре. Эти ячейки называют регистрами общего назначения (РОН). Они сокращают длину адресной части команды. Другие их преимущества рассмотрим позднее.

Не путайте регистры общего назначения и кэш-память. Адреса ячеек внешней относительно процессора памяти и ячеек кэш одинаковы. Выигрыш в скорости возникает за счёт быстродействия кэш-памяти.

Связь между процессором и памятью показана на рис. 1.2.

Рис. 1.2. Обмен информацией между процессором и памятью

Электронные вычислительные машины (ЭВМ) строятся по гарвардской структуре или структуре, предложенной фон Нейманом. Разница этих структур определяется местонахождением в ЭВМ памяти программ.

В гарвардской структуре память программ и память данных разделены физически и логически, т.е. управляющие ими сигналы идут по разным проводам и команды обращения к ним имеют разные коды.

Фон Нейман первым описал ЭВМ, программа работы которой находилась в памяти машины, а не на перфоленте [1]. Обращение и к программам, и к данным в таких ЭВМ выполняется одними и теми же сигналами и командами.

Проще говоря, в неймановской структуре одна магистраль и одна память, а в гарвардской – по две. Одна магистраль связывает процессор с памятью программ, по второй магистрали процессор обменивается данными с памятью данных и периферийными устройствами.

1.4.1. Неймановская структура ЭВМ

Существует огромное разнообразие схем построения электронных вычислительных машин. В зависимости от области применения изменяется и структура ЭВМ. Рассмотрим обобщенную фон-неймановскую структуру, типичную для большинства компьютеров. Она приведена на рис. 1.3. Опишем элементы этой структуры.

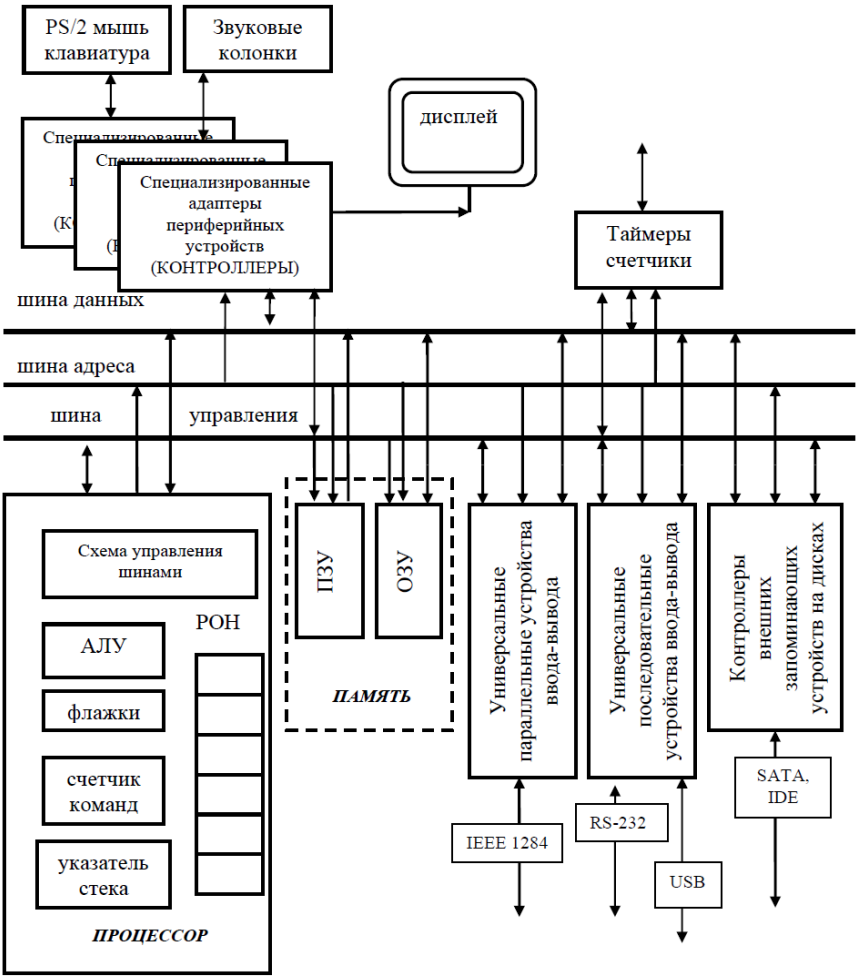

Рис. 1.3. Обобщенная структурная схема ЭВМ

Микропроцессор – ядро вычислительной машины. В его состав обязательно входят: арифметико-логическое устройство, программный счетчик, стековый регистр, регистр признаков, несколько регистров общего назначения и схема управления магистралью. Арифметико-логическое устройство выполняет с данными операции сложения, сравнения, сдвиг, перемещение и т.д. Данные могут располагаться как в реги-

страх микропроцессора, так и во внутренней памяти ЭВМ. Программный счетчик, часто называемый счетчиком команд, содержит номер текущей ячейки памяти программ. Из этой ячейки считывается код выполняемой операции. После каждой операции содержимое программного счетчика нормально увеличивается на единицу, однако оно может принимать и иное значение по команде условного или безусловного перехода. Схема управления шинами обеспечивает связь микропроцессора с памятью и устройствами ввода-вывода. В стековом регистре хранится адрес вершины стековой памяти. Принципы организации стековой памяти будут рассмотрены позднее. Регистр признаков, иначе называемый регистром условий или регистром флажков, используется при выполнении команд условного перехода.

В высокопроизводительных микропроцессорах устанавливается кэш-память, которая служит для увеличения скорости выполнения программ. Информацию из кэш-памяти процессор считывает быстрее, чем из внешнего запоминающего устройства.

Запоминающие устройства, входящие в состав памяти ЭВМ, хранят коды команд и результаты работы программ. Память делится на **оперативное** запоминающее устройство ОЗУ и **постоянное** запоминающее устройство ПЗУ. После выключения питания информация в ОЗУ теряется. ПЗУ состоит из микросхем памяти, в которых информация сохраняется и после выключения питания. В них хранят часто используемые программы. Для запуска машины после включения питания, когда в ОЗУ ещё ничего нет, все ЭВМ имеют в ПЗУ хотя бы одну программу для начальной загрузки. Следует иметь в виду, что ПЗУ в подавляющем большинстве могут использоваться только для чтения. Информацию в них заносят специальными программаторами до установки микросхем в машину.

Связь с периферийными устройствами осуществляется по **магистрали** через **унифицированные устройства ввода-вывода** или **контроллеры**. Эти устройства называют часто адаптерами. В контроллерах унифицированные форматы команд данных преобразуются в специальные форматы, управляющие коды и сигналы соответственно отдельным периферийным устройствам (дисплей, дисковод). Многие периферийные устройства, такие как принтер и мышь, соединены с шинами ЭВМ через унифицированные устройства ввода-вывода.

Магистраль состоит из трех шин: шины адреса, шины данных и шины управления.

Шина адреса. Все элементы ЭВМ, содержащие интересующую процессор информацию, имеют свой адрес. Код адреса передается по шине, подключенной ко всем устройствам. Шина адреса является однонаправленной, если лишь микропроцессор вырабатывает сигнал кода адреса а остальные устройства только опознают его. Так построена шина в простых микропроцессорных системах, где все устройства, кроме процессора, являются пассивными и только микропроцессор вырабатывает адреса данных, передаваемых в системе. В компьютерах для увеличения скорости обмена информацией некоторые устройства имеют прямой доступ к памяти. По их запросу процессор прекращает работу и отключается от магистрали. Полным хозяином ЭВМ становится устройство, запросившее прямой доступ к памяти. В подавляющем большинстве случаев таким устройством является контроллер дисководов.

Шина данных. По этой шине передаются данные, которыми обмениваются устройства ЭВМ. Для обеспечения обмена данными между различными устройствами шина реализуется двунаправленной. Иначе говоря, во время чтения данных из устройства к шине подключены его выходы, а при записи – входы.

Шина управления. По ней передаются управляющие сигналы, предназначенные для синхронизации и определения операций микропроцессорных систем. Эти сигналы задают начало и последовательность срабатывания различных устройств системы. Каждый микропроцессор имеет уникальную систему сигналов управления, поэтому описание всех линий шины управления можно найти в технической документации на конкретный микропроцессор.

1.4.2. Гарвардская структура

Из рис. 1.4 видно, что в состав ЭВМ гарвардской структуры входят все элементы неймановской структуры.

Две магистрали позволяют одновременно по одной – записывать результат операции, по другой – считывать следующую инструкцию. Постоянная связь программного счётчика и регистра команд с памятью программ упрощает схемотехнику ЭВМ.

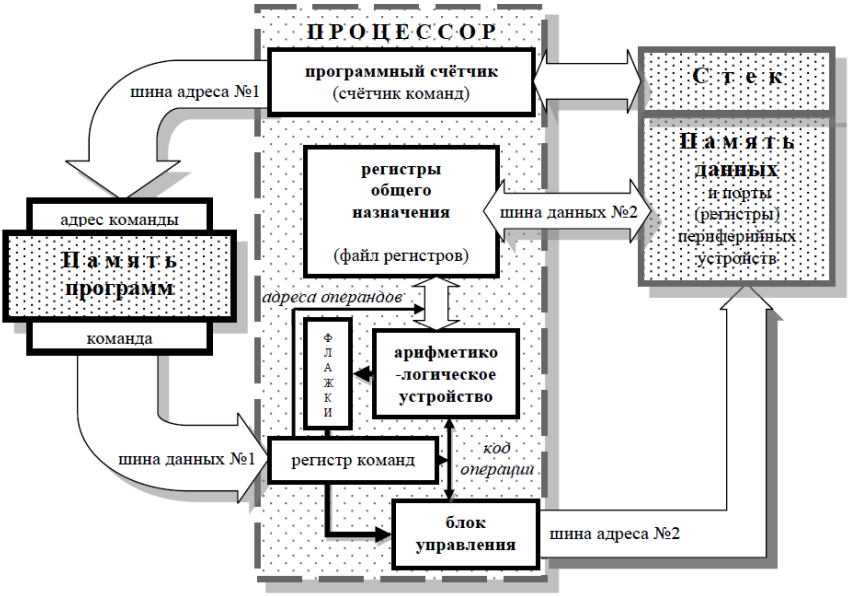

Рис. 1.4. ЭВМ гарвардской структуры

1.5. Компьютеры и контроллеры

Универсальные вычислительные машины, выполняющие как вычислительные, так и управляющее задачи, принято называть *компьютерами*.

Управляющие объектами, устройствами и процессами ЭВМ называют *контроллерами*. В них заложена обычно одна неизменяемая в процессе работы программа, поэтому программа хранится в недорогом постоянном запоминающем устройстве. Универсальные контроллеры, умещающиеся на одном кристалле, строят по гарвардской структуре. Это позволяет упростить управление и уменьшить размер наиболее сложного и дорогого элемента ЭВМ оперативного запоминающего устройства, хранящего изменяющиеся данные. В современных однокристальных микроконтроллерах разделены не только управляющие памятью программ и памятью данных сигналы, но и их шины данных и адресов. Выполнение операции, заканчивающееся записью данных, происходит одновременно со считыванием следующей команды и её дешифрированием, т.к. данные и коды команд проходят по разным магистралям.

1.6. Открытая архитектура

Персональные IBM PC совместимые компьютеры основываются на концепции “открытой архитектуры”.

Впервые принцип открытой архитектуры был использован в компьютере Apple II. От других компьютеров семидесятых годов он отличался наличием внутри системного блока расширительных гнезд. Вставляя в них различные устройства, пользователи могли изменять конфигурацию своих машин в соответствии со своими личными предпочтениями. Компания Apple опубликовала также подробную информацию об Apple II. Это дало возможность фирмам - производителям выпустить разнообразные платы расширения и адаптеры. Открытая архитектура сделала Apple II гибкой и потенциально мощной компьютерной системой.

Примерно в это время компания IBM начала создавать свой первый персональный компьютер. В новой разработке фирма позаимствовала у Apple II открытую архитектуру и спроектировала PC (Personal Computer) – персональный компьютер на основе более или менее готовых компонент, применив модульную конструкцию. IBM опубликовала также большой объем документации, в которой описывался новый ПК. Заинтересованные фирмы и частные лица получили доступ к аппаратным спецификациям и могли теперь проектировать платы для гнезд расширения. Кроме того, были опубликованы программные спецификации, что сделало возможным разработку программного обеспечения для новой машины. Именно вся эта документация привела, в конце концов, к организации производства клонов и совместимых с PC компьютеров.

Глава 2. МИКРОПРОЦЕССОРЫ

2.1. Структура и функционирование микропроцессора

Под процессором понимается функциональный блок ЭВМ, предназначенный для логической и арифметической обработки информации на основе принципа программного управления.

Изначально микропроцессоры делились на однокристальные и секционные. Характерными чертами однокристальных микропроцессоров являются фиксированный набор команд и фиксированная разрядность. В микропроцессорах секционного типа обрабатывающее устройство строилось из отдельных секций с возможностью наращивания разрядности шины данных. Систему команд таких микропроцессоров можно было изменять в зависимости от класса решаемых задач. В настоящее время практически не применяются.

Наибольшее распространение получили однокристальные микропроцессоры по причине простоты программирования и функциональной законченности. Они строятся на основе жесткой («зашитой») логики, либо с использованием микропрограммирования.

Жёсткая логика применяется в микропроцессорах с сокращённым набором команд (RISC) и гарвардской структурой ЭВМ. Биты команды сразу подключают через две шины нужные регистры и код операции к арифметико-логическому устройству. Это позволяет за один такт выполнить команду и во время записи результатов и флажков считать очередную команду из памяти программ.

Управляющее устройство в процессорах с ***микропрограммным управлением*** разбивает команду из памяти на несколько микрокоманд и выполняет эту микропрограмму за несколько, может быть даже десятков, тактов.

Рассмотрим структурную схему однокристального процессора с **одноадресной** системой команд и **микропрограммным** управлением.

Состав микропроцессора представлен на рис. 2.1. Разберем назначение его функциональных узлов.

Программный счетчик или счетчик адреса команд служит для хранения адреса команды, подлежащей выполнению. После выборки последнего байта команды содержимое счетчика увеличивается на единицу. При выполнении команд условного и безусловного перехода, а также при обращении к подпрограммам, следующий адрес команды

формируется путем загрузки в программный счетчик адресной части этих команд. Код адреса из программного счетчика передается в программную память через однонаправленную шину адреса.

Регистр команд принимает по шине данных выбранный из памяти код команды и хранит его в течение цикла выполнения команды. Если команда хранится в нескольких ячейках памяти, то для загрузки каждой части команды требуется отдельный машинный цикл выборки команды. При 8-битной шине данных 50-70% времени выполнения 2–3-байтных команд уходит на выборку команды из памяти.

Указатель стека хранит адрес последней занятой ячейки в области стековой памяти. Под стековой памятью понимают участок оперативной памяти, в которой запись или выборка слова производится по принципу: последний пришел – первый вышел. Адрес ячейки стековой памяти, по которой выполняется запись или чтение слова, называют вершиной стека. Начальный адрес стека (дно стека) устанавливается программным путём. Стековый регистр находится в процессоре, а сам стек – снаружи в оперативной памяти.

Регистры общего назначения являются сверхоперативной памятью вычислительной машины. В них хранят исходные данные (операнды) при выполнении логических и арифметических операций, а также промежуточные результаты вычислений. Их использование при выполнении команды позволяет повысить быстродействие микро-ЭВМ за счёт сокращения времени пересылок между процессором и памятью. Регистры общего назначения программно доступны и обращение к ним осуществляется посредством команд передачи данных.

Аккумулятор (рабочий регистр) используют в одноадресных процессорах в начале команды для хранения одного из операндов и результата выполнения команды в конце, т.е. аккумулятор является источником и приёмником информации. Обмен кода с периферийными устройствами также осуществляется через аккумулятор.

Арифметико-логическое устройство предназначено для арифметической и логической обработки данных, выполнения операций сдвига, формирования признаков результатов операций. Буферные регистры хранят коды операндов во время выполнения арифметико-логическим устройством операции над ними.

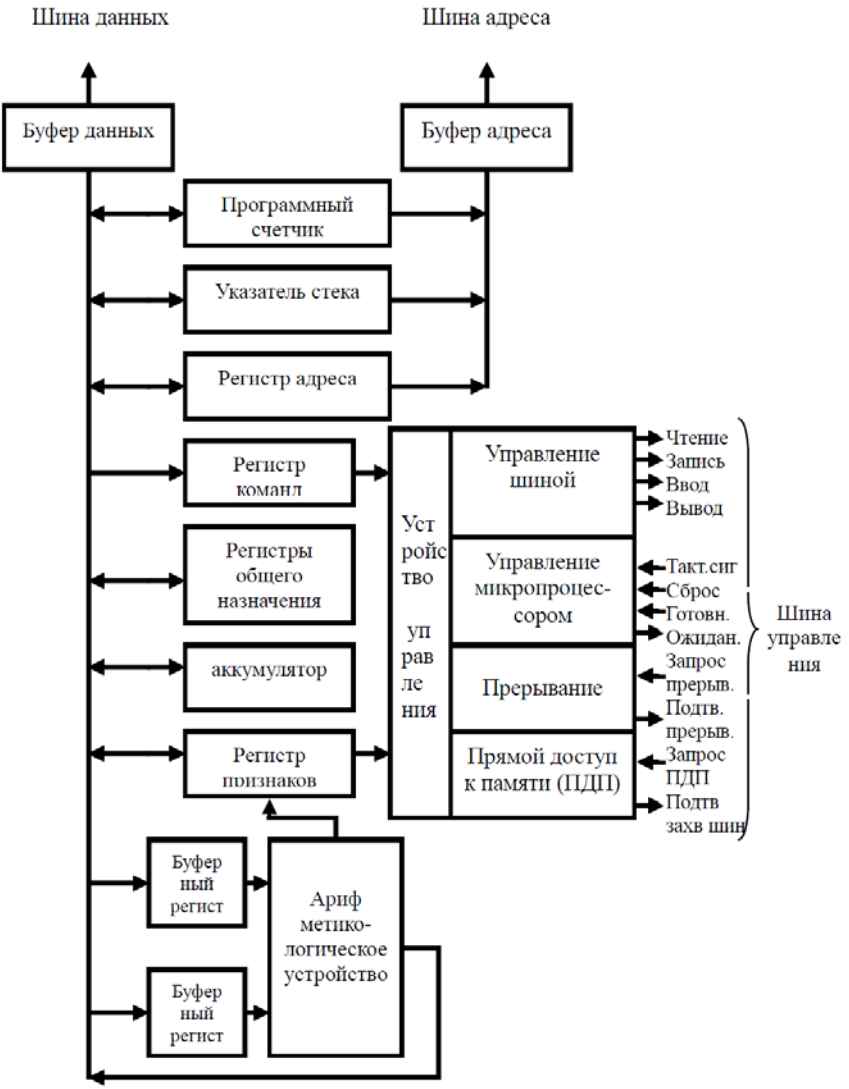

Рис. 2.1. Структурная схема одноадресного микропроцессора с микропрограммным управлением

Признаки результатов операций хранятся в ***регистре признаков*** или флажков. Эти признаки используются командами условного перехода при ветвлениях в программах.

Обмен кодами осуществляется через ***буферы шин***, которые изолируют внешние шины от внутренних.

Формирование последовательности внутренних и внешних управляющих сигналов из других устройств микроЭВМ выполняются ***устройством управления***. Оно функционально состоит из четырех блоков: блока управления шиной, блока управления микропроцессором, блока прерываний и блока прямого доступа к памяти (ПДП). Сигналы с устройства управления передаются к другим элементам ЭВМ и принимаются от них по шине управления.

Рассмотрим выполнение команды сложения ADD B. Она добавляет к содержимому аккумулятора содержимое регистра общего назначения B. Процессор с микропрограммным управлением разбивает её на микрокоманды. Для исполнения команды необходимы четыре машинных такта:

1. Адрес очередной команды из программного счётчика выставляется на шину адреса. По этому адресу из памяти считывается код команды и по шине данных записывается в регистр команд. Код дешифрируется управляющим устройством и разбивается на микрокоманды. Содержимое счётчика команд увеличивается. Всё происходит за один такт.
2. Содержимое аккумулятора по внутренней шине данных записывается в первый буферный регистр.
3. Содержимое регистра B по внутренней шине данных записывается во второй буферный регистр.
4. Результат и его признаки с выхода арифметико-логического устройства записываются в аккумулятор и регистр флажков.

Команда не разбивается на микрокоманды в процессорах с жесткой логикой. Такие процессоры используют сокращённый набор команд, что упрощает схемотехнику процессора и повышает их быстродействие.

Как за один такт выполняется команда в процессоре с ***жесткой логикой*** рассмотрим на примере двухадресного AVR-контроллера с сокращённым набором команд (RISC). Подключение арифметико-логического устройства показано на рис. 2.2. Микроконтроллер построен по гарвардской структуре. Память программ соединена с процессором по отдельным от памяти данных проводам (по отдельной магистрали).

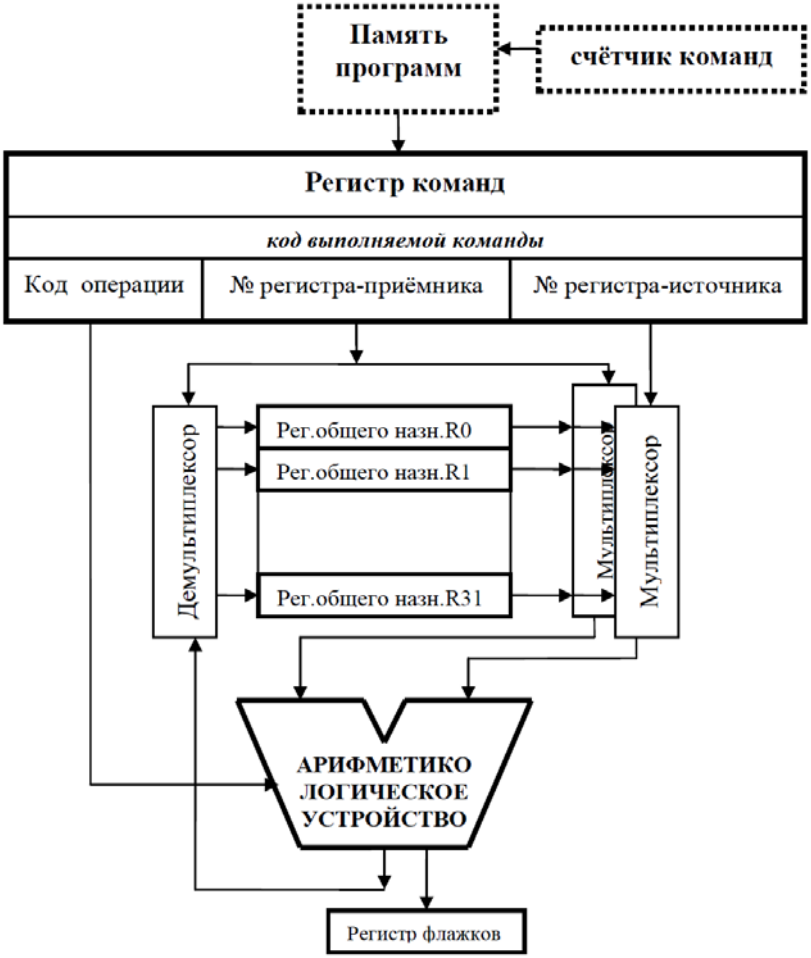

Рис. 2.2. Схема подсоединения регистров к АЛУ

Адрес очередной команды поступает на память программ со счётчика команд. Выбранный из памяти код команды фиксируется в регистре команд. Код команды имеет постоянную длину и состоит из инструкции и двух адресов регистров общего назначения (регистра-приёмника и регистра-источника). В этих регистрах расположены операнды команды.

В начале такта исполнения сразу после выборки команды из программной памяти к входам АЛУ через мультиплексоры подключаются регистр-приёмник и регистр-источник.

В конце такта результат с выхода АЛУ с помощью демультиплексора записывается в регистр-приёмник. В это же время в счётчике команд изменяется адрес очередной команды и из памяти программ извлекается её код. Микроконтроллер готов выполнить следующую команду.

2.2. Система команд

2.2.1. Методы адресации

Одним из важнейших показателей мощности системы команд любого микропроцессора является количество методов адресации. Под методом адресации понимается способ обращения команды к обрабатываемому операнду.

Непосредственная адресация. Операнд находится в памяти непосредственно за кодом операции.

Регистровая адресация. Операнд находится в регистре или в паре регистров общего назначения, расположенных в микропроцессоре.

Абсолютная (прямая) адресация. За кодом операции следуют два или больше байта, которые являются адресом.

Косвенно-регистровая адресация. Операнд находится в ячейке памяти, адрес которой содержится в одном из регистров, а в процессорах с восьмиразрядными регистрами – в одной из регистровых пар.

Относительная адресация. Этот вид адресации может использоваться командами условных и безусловных переходов. Адрес формируется как сумма содержимого программного счетчика, то есть начального адреса следующей команды, и смещения объемом в один, два или четыре байта. Смещение указывает положение следующей команды относительно текущей. В простейшем случае команда состоит из двух байтов: первый байт содержит код операции, а второй – смещение от –128 до +127. При использовании относительной адресации программа становится перемещаемой, т.е. не зависит от месторасположения в памяти ЭВМ.

Индексная адресация и базовая адресация аналогичны относительной адресации, только базой является не счётчик команд, а один из

индексных или базовых регистров. Смещение, непосредственно представленное в команде, складывается с содержимым индексного регистра для индексной адресации или с содержимым базового регистра для базовой адресации и образует адрес ячейки памяти, в которой находятся данные. Смещение размещается в одном или двух байтах.

Базово-индексная адресация. Адрес операнда образуется путем сложения содержимого базового и индексного регистров. В некоторых однокристальных микроЭВМ в качестве индексного регистра используют аккумулятор.

Последние три типа адресации удобно использовать при обращении к массивам и таблицам.

Строго говоря, индексная, базовая и базово-индексная адресация относятся к косвенным типам адресации, поэтому можно встретить такое название базово-индексной адресации – косвенная адресация по сумме базового и индексного регистров.

Битовая адресация используется в группе команд, работающих с отдельными битами. Использование таких команд требует наличия в микропроцессоре специального битового процессора, совершающего операции не с байтами, а с отдельными битами.

Микропроцессоры фирмы Intel, начиная с 8086, реализуют **сегментную** организацию памяти, при которой физический адрес ячейки памяти формируется путем сложения базового адреса сегмента и относительного адреса ячейки внутри сегмента.

Базовый адрес сегмента определяется содержимым шестнадцатиразрядного сегментного регистра и зависит от режима работы микропроцессора. В режиме обработки шестнадцатиразрядных данных (режим реальных адресов или режим виртуального микропроцессора 8086) 20-разрядный базовый адрес формируется сдвигом содержимого сегментного регистра на четыре разряда влево. В режиме обработки 32-разрядных данных (защищенный режим) 32-разрядный базовый адрес содержится в дескрипторе. Дескриптор выбирается из таблицы дескрипторов с помощью селектора, который находится в сегментном регистре микропроцессора. Защищенный режим есть в микропроцессорах 80286 и выше.

В этих микропроцессорах используются рассмотренные выше способы адресации с одной лишь оговоркой: под адресом в этом случае следует понимать относительный адрес ячейки внутри сегмента.

Примеры адресации в командах пересылок и сложения для восьмиразрядного микропроцессора Intel 8080 и микроконтроллеров семейства AVR и PIC18 приведены в табл. 2.1.

Таблица 2.1. **Типы адресации памяти**

Адресация	Intel 8080	AVR	PIC18
Непосредственная	**MVI B, K8** **ADI K8**	**LDI Rd, K8** НЕТ	**MOVLW K8** **ADDLW K8**
Регистровая	**ADD C**	**ADD Rd,Rr**	**ADDWF f,d,a**
Регистры	*A,B,C,D,E,H,L*	*с R0 по R31*	*256 регистров f*
Абсолютная (***прямая***)	**LDA** аааа	LDS Rd,аааа	НЕТ. ОЗУ разбито на 16 банков по 256 регистров
	JMP аааа	**JMP** аааа	**GOTO** аааа
Косвенно-регистровая (***косвенная***)	**MOV B,M**	**LD R1,X**	**MOVWF INFx,d,a**
	ADD M	**IJMP** **адрес в Z**	ADDWF INFx,d,a
регистры косвенной адресации	*H:L*	*X=R27:R26* *Y=R29:R28* *Z=R31:R30*	*три 12-разрядных регистра (FSRx): FSR0, FSR1, FSR2*
Относительная	НЕТ	**RJMP n**	**BRA n**
Индексная	НЕТ	LDD Rd,Z+n	**PLUSWx** Значение в регистре WREG используется как смещение

K8 – восьмибитовая константа;
f – номер регистра;
n – смещение;
аааа – адрес ячейки памяти.

2.2.2. Форматы команд

В системе команд простейших микропроцессоров и однокристальных микроЭВМ имеются однобайтные, двухбайтные и трехбайтные команды.

В 32-разрядных процессорах команда значительно длиннее и может содержать от одного до одиннадцати байт. Обобщенный вид формата команд показан на рис. 2.3.

Код операции	Адресация	Смещение	Операнд
1 или 2 байта	0,1,2 байта	0,1,2,4 байта	0,1,2,4 байта

Рис. 2.3. Общий формат команд процессоров Intel 486

Фиксированная длина и простой формат команды применяются в ***RISC-процессорах*. RISC** (*reduced instruction set computer)* – сокращённый набор компьютерных инструкций (команд). В таких процессорах быстродействие увеличивается за счёт упрощения декодирования несложных команд. Простые команды позволяют уменьшить количество транзисторов в процессоре.

В микроконтроллерах семейства AVR и PIC18 длина команды и шина, по которым они передаются в процессор, имеют 16 разрядов. В исключительных случаях команда может состоять из двух 16-разрядных слов.

ARM-процессоры используют 32-разрядные инструкции. Это упрощает декодирование за счет снижения плотности кода. Позднее режим Thumb с 16-разрядными командами повысил плотность кода программы.

2.2.3. Представление чисел в микропроцессорах

Область применения и конкретная задача, решаемая микроЭВМ, определяет формат представления чисел в микропроцессоре.

В простейших управляющих микропроцессорных системах широко используют способ ***представления целых чисел без знака*** в двоичных кодах (табл. 2.2).

Таблица 2.2. **Целое число без знака**

Имя бита	D7	D6	D5	D4	D3	D2	D1	D0
Вес бита	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1

Диапазон представления чисел: от 0 до 2^8-1 (255).

В ориентированных на обработку данных микропроцессорных системах, в которых выполняются операции вычитания, используется

представление чисел со знаком. Положительное число представляется модулем, а отрицательное – дополнительным кодом.

Представление отрицательных чисел в дополнительном коде позволяет выполнять арифметические операции с непосредственным использованием *переноса* при сложении и *заема* – при вычитании.

Отрицательное число в дополнительном коде получается прибавлением к обратному коду модуля отрицательного числа единицы. Обратный код – это двоичный код, все разряды которого инвертированы, то есть логические единицы заменены нулями, а нули – единицами.

Примеры такого представления чисел приведены в табл. 2.3.

Таблица 2.3. **Целое число со знаком**

Номер и вес разряда								Десятичный код числа
D7 Знак	D6 2^6	D5 2^5	D4 2^4	D3 2^3	D2 2^2	D1 2^1	D0 2^0	
1	0	0	0	0	0	0	0	-128
1	0	0	0	0	0	0	1	-127
1	1	1	1	1	1	1	1	-1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1
0	1	1	1	1	1	1	1	127

Диапазон представления чисел от –128 до +127.

Команда **NEG** (изменение знака) вычисляет дополнительный код, если число положительное, и наоборот. Она проводит следующее действие: **Rn=00-Rn.**

В микропроцессорных системах, требующих выполнения операций над числами по правилам десятичной арифметики, используют ***десятичный двоично-кодированный упакованный формат***. Чаще его называют двоично-десятичным кодом. Байт условно разбивается на два полубайта (по другому – две тетрады). В каждой из тетрад кодируется один разряд десятичного числа (табл. 2.4).

Диапазон представления чисел в одном байте от 0 до 99.

Двоично-десятичный код (по-английски *binary-coded decimal*) обозначается BCD или 8421-BCD.

Если числа представлены в двоично-десятичном коде, то после операции сложения и вычитания необходимо выполнить операцию десятичной коррекции.

Таблица 2.4. **Десятичный двоично-кодированный упакованный формат (8421-BCD)**

Номер и вес разряда								Десятичный код числа
D7 2^3*10	D6 2^2*10	D5 2^1*10	D4 2^0*10	D3 2^3	D2 2^2	D1 2^1	D0 2^0	
0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	18
0	1	1	1	0	1	0	1	75
1	0	0	1	1	0	0	1	99

Двоично-десятичный код упрощает вывод чисел на индикацию и умножение или деление на десять. К недостаткам следует отнести усложнение арифметических операций и увеличение требуемой памяти.

Шестнадцатеричные числа

Целые двоичные числа громоздки, поэтому при записи кодов команд и данных применяют шестнадцатеричный код, в котором четыре разряда двоичного кода соответствуют одному разряду шестнадцатеричного. Шестнадцатеричные цифры от 10 до 15 обозначаются буквами с A по F.

Числа с плавающей запятой

Числа с плавающей запятой удобны при представлении очень больших и очень малых чисел, они используют представление действительных чисел в виде мантиссы и порядка.

Числа с плавающей запятой бывают одной, двойной и расширенной точности. Их размер 4, 8 и 10 байтов соответственно.

Рассмотрим число одинарной точности. Это широко распространенный компьютерный формат представления вещественных чисел. Число занимает в памяти 32 бита (4 байта). Как правило, под ним понимают формат числа с плавающей запятой стандарта IEEE 754.

Числа одинарной точности с плавающей запятой обеспечивают относительную точность 7-8 десятичных цифр в диапазоне от 10^{-38} до, примерно, 10^{38}.

В современных компьютерах вычисления с числами с плавающей запятой поддерживаются аппаратным сопроцессором (FPU – Floating Point Unit). Если в составе ЭВМ нет FPU, работа с числами с плавающей запятой осуществляется программно.

Вид числа одинарной точности представлен ниже.

S	2^7	*Порядок*	2^0	$\mathbf{2^{-1}}$	**Мантисса**	$\mathbf{2^{-23}}$
B31	*B30*		*B23*	**B22**		**B0**

Знак – S. Для вычисления показателя степени из восьмиразрядного поля порядка вычитается смещение порядка равное $127_{10} = 7F_{16} = 01111111_2$. Так как в нормализованной двоичной мантиссе целая часть всегда равна единице, то в поле мантиссы записывается только её дробная часть. Для вычисления мантиссы к единице добавляется дробная часть мантиссы из 23-разрядного поля дробной части мантиссы.

2.2.4. Признаки, вырабатываемые АЛУ

Список признаков результата операции в различных микропроцессорах различен, однако типовой его состав характеризуется следующим набором (в скобках другое возможное обозначение):

- **S(F,N)** – признак отрицательного результата при работе с целыми числами **со** знаком;
- **N** – признак отрицательного результата при работе с целыми числами **со** знаком. **N** равен старшему биту;
- **Z(ZF)** – признак нулевого результата;
- **C(CF)** – перенос в старший байт (при восьмиразрядном АЛУ);
- **H(AF,DC)** – перенос из младшей тетрады (четырёх битов) в старшую;
- **V(OF,OV)** – признак переполнения разрядной сетки;
- **P(PF)** – флаг чётности.

Признак отрицательного результата **S** устанавливается в «1», когда при вычитании или сравнении первый операнд меньше второго, в противном случае сбрасывается в ноль. Следует помнить, АЛУ считает операнды целыми числами со знаком.

Признак отрицательного результата **N** устанавливается в «1», когда старший разряд результата **D7** равен единице, в противном случае сбрасывается в ноль. Применяется для работы с отрицательными числами, представленными в дополнительном коде.

Признак нулевого результата **Z** устанавливается в единицу при нулевом значении результата, при ненулевом результате сбрасывается в ноль.

***Перенос* C** устанавливается в единицу при переносе старшего разряда результата D7 в старший байт во время сложения и при заёме из старшего байта во время вычитания, в противном случае сбрасывается.

Дополнительный флаг переноса H устанавливается в единицу при переносе из разряда результата D3 в разряд D4. При отсутствии переноса сбрасывается в ноль. Флаг используется при обработке двоично-десятичных кодов.

***Признак переполнения разрядной сетки* V** используется при работе с операндами, представленными числами со знаком. Проблемы возникают, когда в результате операции сложения число становится больше +127 или меньше –128, что приводит к переносу в знаковый разряд D7. Знак числа становится неверным, о чем и предупреждает единица в бите признака переполнения разрядной сетки.

***Флаг чётности* P** устанавливается, если результат содержит чётное число единичных битов.

Инструкции передачи данных (подобных MOV, LD, ST) флажки не устанавливают. Проверка числа в регистре на ноль или отрицательность в микроконтроллерах семейства *AVR* проводится командой **TST** (тест).

Команды ветвления (команды передачи управления по другому адресу) также не влияют на флажки.

2.2.5. Упрощенный набор команд процессоров Intel

	Пересылка данных	
MOV a,b	пересылка	a <= b, b не меняется
	Стековые операции	
PUSH rm	пересылка в стек	заносит rm в стек (2 байта)
POP rm	извлечение из стека	извлекает из стека в rm (2 байта). и заносит в регистр или память
	Управление вводом-выводом	
IN A,port	ввод	в аккумулятор записывает содержимое порта, A <= port
OUT port,A	вывод	port <= A
	Арифметические операции	
ADD a,b	добавление	a <= a+b, b не меняется
ADC a,b	добавление с переносом	a <= a+b+c, b не меняется
SUB a,b	вычитание	a <= a-b, b не меняется
CMP a,b	сравнение	устанавливает флажки (как при вычитании), но a и b не меняются
INC rm	инкремент	rm <= rm +1
DEC rm	декремент	rm <= rm – 1
NOT rm	НЕ	инверсия битов, поразрядно
NEG rm	изменение знака	rm <= rm с изменённым знаком

Первые четыре арифметические команды работают с парами чисел (двуоперандные команды), которые обозначены a и b.
Последние 4 арифметические операции имеют только 1 операнд, который может быть содержимым либо регистра r, либо ячейки памяти m.

a и b могут означать следующее:
1) содержимое ячейки памяти m
MOV AX,A845H (AX – r, A845H – m)
2) содержимое регистра центрального процессора r
MOV AX,CX (AX,CX – r)
3) непосредственный аргумент
MOV AX,#A8FFH
4) регистр, в котором записан адрес ячейки памяти, содержимое которой перемещается (косвенная)
MOV AX,@BX

Логические операции и операции сдвига

AND a,b	логическая И	a <= a*b, поразрядно
OR a,b	логическая ИЛИ	a <= a+b, поразрядно
SAL rm	арифметический сдвиг влево	
SAR rm	арифметический сдвиг вправо	
SHL rm	логический сдвиг влево	
SHR rm	логический сдвиг вправо	
ROL rm	циклический сдвиг влево	
ROR rm	циклический сдвиг вправо	
RCL rm	циклический сдвиг влево через флаг переноса	
RCR rm	циклический сдвиг вправо через флаг переноса	

Команды управления

JMP label	передача управления	управление передаётся команде, помеченной меткой label
Jcc label	условная передача	управление передаётся команде, помеченной меткой label, если или когда флаг cc – истинна (лог.1)
CALL label	вызов	заносит в стек адрес следующей команды, передаёт управление команде, помеченной меткой label
RET	возврат управления	извлекает из стека адрес и передаёт управление по этому адресу
IRET	возврат управления из прерывания	извлекает адрес, восстанавливает флажки и передаёт управление по этому адресу
STI	инициализация	разрешает прерывание прерывания
CLI	сброс прерывания	запрещает прерывание

2.2.6. Работа команд управления стеком

Стек – это последовательный набор данных, организованный по принципу «последним пришёл – первым вышел». Стек нужен для работы с подпрограммами.

Стек расположен в обычном ОЗУ, а указатель стека – в регистре SP центрального процессора, обеспечивает возможность доступа к той ячейке памяти, которая является вершиной в данный момент времени.

Как изменяется содержимое ячеек стека и регистров процессора при записи в стек из регистра AX по команде ***PUSH AX*** и при чтении из стека в регистр BX по команде ***POP BX*** показано на рис. 2.4.

Для микропроцессоров фирмы INTEL стек состоит из 16-разрядных слов и, по мере занесения в него данных, растёт вниз в ОЗУ. Содержимое регистра SP автоматически декрементируется (уменьшается) на 2 перед каждой операцией ***PUSH*** и инкрементируется (увеличивается) на 2 после каждой операции ***POP***.

Дно стека – это адрес ячейки ОЗУ, с которой начинается стек после начальных установок.

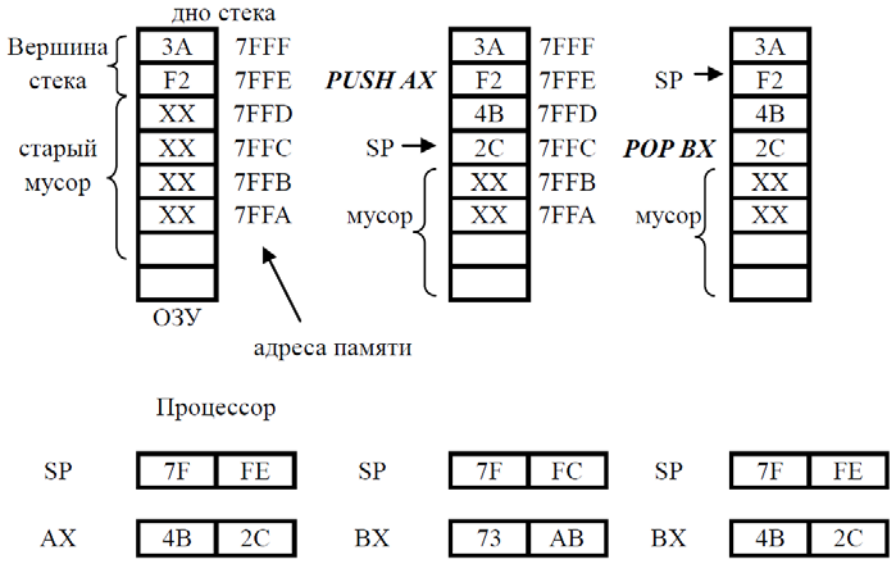

Рис. 2.4. Состояние стека и регистров при выполнении стековых команд

2.3. Методы повышения быстродействия процессоров

Повышение быстродействия за счёт быстродействия транзисторов процессора не рассматриваем.

2.3.1. Разрядность регистров и шин данных процессора

Размер регистров процессора определяет точность вычислений при выполнении одной команды. Число одинарной точности в x86 состоит из 32 двоичных разрядов, то есть 4 байтов. Соответственно в числе двойной точности 8 байтов. Для сложения двух чисел одинарной точности восьмиразрядным процессором нужно по крайней мере четыре команды сложения, не считая команд пересылок в случае одноадресного процессора.

Семейство x86 начиналось с 16-разрядных процессоров. Процессор содержал четыре двухбайтовых регистра общего назначения: AX, BX, CX, DX.

В 32-разрядных процессорах семейства регистры называются соответственно EAX, EBX, ECX, EDX, в 64-разрядных – RAX, RBX, RCX, RDX. Дополнительно в 64-разрядных расположены восемь регистров – R8 – R15 (рис. 2.5).

Изначально в семействе процессоров x86 математический сопроцессор для операций с плавающей запятой (FPU) имел восемь 80-разрядных регистров. В следующем семействе (x86-64) их уже шестнадцать.

Для работы с командами типа одна инструкция — множество данных (*Single Instruction, Multiple Data*) в процессор вводятся 128-разрядные (XMM) и 256-разрядные (YMM) регистры.

Широкая шина данных магистрали между процессором и ОЗУ увеличивает скорость обмена информацией при том же быстродействии модулей ОЗУ. Обычно это два канала по 64 разряда.

Процессоры с разъёмом LGA 2011 поддерживают четырехканальный режим работы с модулями оперативной памяти DDR3. То есть могут считывать или записывать 64x4=256 разрядов данных из ОЗУ одновременно.

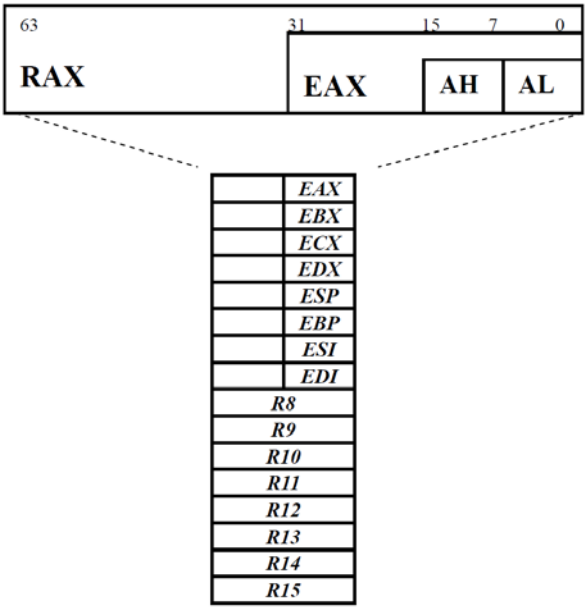

Рис. 2.5. Регистры общего назначения процессоров с x86 по x86-64

2.3.2. Конвейерный метод выполнения команд

Конвейерный *метод* начинался с конвейерного *принципа* в Inter 8086, который в современных процессорах сейчас не используется.

Микропроцессор с конвейерным ***принципом*** выполнения команд может быть условно разделён на две части: устройство сопряжения с магистралью и устройство обработки (рис. 2.6).

Отличие микропроцессора с конвейерным принципом выполнения команд заключается в следующем:

а) введение в устройство сопряжения регистров очереди команд (РОК);

б) разделение устройства управления на две части: устройство управления шин и устройство микропрограммного управления.

На рис. 2.6. приняты следующие сокращения:

УМУ – устройство микропрограммного управления;

УШ – устройство управления шин;

РОК – регистр очереди команд (6 штук);

УО – устройство обработки;

УС – устройство сопряжения.

Устройство сопряжения (УС) выполняет цикл выборки слова из памяти всякий раз, когда в очереди освобождается, по крайней мере, два байта.

Устройство обработки (УО) извлекает из РОК коды команд по мере необходимости.

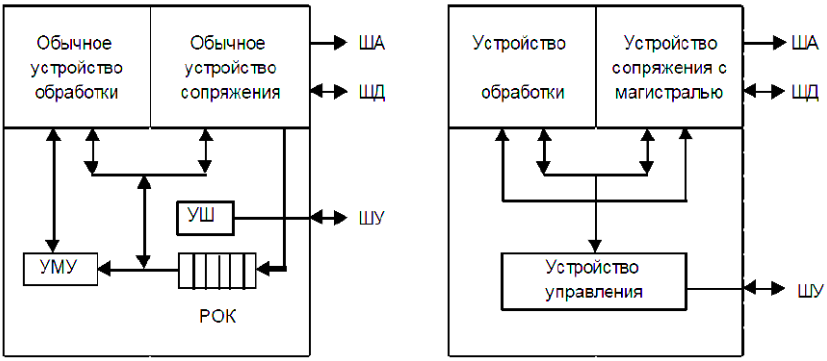

Рис. 2.6. Структура микропроцессора с конвейерным принципом выполнения команд и обычного микропроцессора

Очередь организована по принципу «первым пришёл, первым вышел». Шести регистров очереди считается достаточно, чтобы при шестнадцатиразрядной шине данных свести затраты выборки команд из памяти до минимума. В микропроцессорах с 32-разрядной шиной данных очередь команд содержит шестнадцать однобитных регистров.

В РОК находятся команды, которые хранились в ячейках памяти, непосредственно следующих за текущей командой. При передаче управления в другую ячейку памяти ход выполнения программы нарушается.

Устройство сопряжения очищает регистры очереди, выбирает команду по адресу перехода, передаёт его в устройство обработки, минуя очередь, и начинает новое заполнение регистров.

Дальнейшее развитие конвейерный принцип получил в ***конвейерном методе***.

Конвейерный метод основан на использовании времени простоя отдельных узлов процессора, выполнивших свою функцию и ожидающих прихода следующей команды. Процессор разбивается на N функ-

ционально законченных блоков, передающих друг другу процесс выполнения команд. Свою часть работы блок выполняет за один машинный такт. Освобождающийся блок сразу приступает к выполнению своей функции в следующей команде, не дожидаясь завершения текущей команды. В установившимся режиме одновременно работают N блоков. Результаты очередной команды появляются через один такт, хотя на команду тратится N тактов.

В процессоре Intel 486 выполнение команды разбивается на пять участков: выборка команды, декодирование, подготовка операндов, исполнение записи результатов.

Табл. 2.5 иллюстрирует конвейерный метод.

Таблица 2.5. **Последовательность выполнения команд**

Номер такта	1	2	3	4	5	6	7	8	9
Выборка команды	1-я ком	2-я ком	3-я ком	4-я ком	5-я ком	6-я ком	7-я ком	8-я ком	9-я ком
Декодирование		1-я ком	2-я ком	3-я ком	4-я ком	5-я ком	6-я ком	7-я ком	8-я ком
Подготовка операндов			1-я ком	2-я ком	3-я ком	4-я ком	5-я ком	6-я ком	7-я ком
Исполнение				1-я ком	2-я ком	3-я ком	4-я ком	5-я ком	6-я ком
Запись операндов					1-я ком	2-я ком	3-я ком	4-я ком	5-я ком

Суперскалярная архитектура использована в процессоре Pentium. Благодаря использованию суперскалярной архитектуры процессор может выполнять 2 команды за 1 такт. Такая возможность существует благодаря наличию двух конвейеров – u- и v-. u-конвейер – основной, выполняет все операции над целыми и вещественными числами; v-конвейер – вспомогательный, выполняет только простые операции над целыми и частично – над вещественными.

В процессорах фирмы AMD глубина конвейера K7. Целочисленный: 10 стадий, вещественно-численный: 15 стадий.

В микроархитектуре K8 используется конвейер с 12 стадиями, значительная часть которых приходится на декодер инструкций.

Микропроцессоры K8 являются суперскалярными, мультиконвеерными процессорами с предсказанием ветвлений и спекулятивным исполнением. Как и процессоры AMD K7 и Intel P6, они теоретически способны исполнять до 3 инструкций за один такт. Как и любой современный x86-процессор, K8 вначале перекодирует внешний сложный

CISC набор x86 инструкций во внутренние RISC-подобные микрооперации, которые, в свою очередь, уже идут на исполнение.

2.3.3. Микропроцессоры с архитектурой RISC

Такие микропроцессоры используют сравнительно небольшой набор наиболее употребительных команд. RISC – это аббревиатура английского выражения Reduced Instruction Set Computers (сокращенный набор команд). RISC-процессор обладает меньшим числом команд фиксированной длины. Упрощенная структура позволяет RISC-процессору развивать более высокую скорость. Все команды работают с операндами, размещёнными в регистрах процессора, и имеют одинаковый формат. Обращение к памяти выполняется специальными командами загрузки регистров и записи в память. Простота структуры и небольшой набор команд позволяют эффективно использовать конвейер.

Противоположностью аббревиатуре RISC является аббревиатура CISC (Complex Instruction Set Computing «вычисления со сложным набором команд»). Все члены семейства x86 – типичные представители CISC-процессоров со сложными, но удобными наборами команд.

В настоящее время многие архитектуры процессоров являются RISC-подобными, к примеру, ARM, DEC Alpha, SPARC, AVR, MIPS, POWER и PowerPC. Наиболее широко используемые в настольных компьютерах процессоры архитектуры x86 ранее являлись CISC-процессорами, однако новые процессоры, начиная с Intel 486DX, являются CISC-процессорами с RISC-ядром. Они непосредственно перед исполнением преобразуют CISC-инструкции x86-процессоров в более простой набор внутренних инструкций RISC.

После того, как процессоры архитектуры x86 были переведены на суперскалярную RISC-архитектуру, можно сказать, что большинство существующих ныне процессоров основаны на архитектуре RISC.

VLIW (англ.*very long instruction word* – «**очень длинная машинная команда**») – архитектура процессоров с несколькими вычислительными устройствами. Характеризуется тем, что одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно. Фактически это «видимое программисту» микропрограммное управление, когда машинный код представляет собой лишь немного свёрнутый микрокод для непосредственного управления аппаратурой.

В суперскалярных процессорах также есть несколько вычислительных модулей, но задача распределения между ними работы решается аппаратно. Это сильно усложняет дизайн процессора и может быть чревато ошибками. В процессорах VLIW задача распределения решается во время компиляции и в инструкциях явно указано, какое вычислительное устройство должно выполнять какую команду.

VLIW можно считать логическим продолжением идеологии RISC, расширяющей её на архитектуры с несколькими вычислительными модулями. Так же, как в RISC, в инструкции явно указывается, что именно должен делать каждый модуль процессора. Из-за этого длина инструкции может достигать 128 или даже 256 бит.

2.3.4. Внутренняя кэш-память

Кэш-память представляет собой быстродействующую буферную память ограниченного объёма, которая располагается между процессором и относительно медленной основной памятью.

В процессе работы отдельные блоки информации копируются из основной памяти в кэш-память, и когда процессор обращается за командой или данными, то сначала проверяется их наличие в кэш-памяти.

Если необходимая информация находится в кэш-памяти, то она быстро извлекается (кэш-попадания). Если информация отсутствует, то она извлекается из основной памяти и одновременно заносится в кэш-память (кэш-промах).

В большинстве случаев программа обращается к ячейкам памяти, расположенным вблизи от ранее используемых, поэтому кэш-попадания происходят чаще кэш-промахов, так как в кэш-память переписывается не одна ячейка, а блок памяти длиной 16 байтов.

Запись данных из процессора во внешнее оперативное устройство может осуществляться двумя способами: методом сквозной записи (write-through) и методом обратной записи (write-back). Информация в оперативной памяти при сквозной записи обновляется, только когда заменяется блок информации в кэш-памяти. Это снижает загрузку шины передачи данных. Кэш-память со сквозным методом записи пересылает данные в ОЗУ каждый раз, когда процессору необходимо сохранить информацию.

Внутренняя кэш-память первого уровня имеет объём 8, 16, 32 64 килобайта и выше.

Помимо внутренней кэш-памяти в системах INTEL 486 и Pentium допускается использование внешней кэш-памяти, построенной на быстродействующих микросхемах памяти статического типа. Для её управления микропроцессор вырабатывает специальные сигналы. В микропроцессорах Pentium-2 и выше кэш-память второго уровня расположена внутри процессора. Ее объем от 128 килобайт до 2 мегабайт.

Кэш-память первого уровня обозначают L1, второго – L2 (Level - уровень). Есть процессоры и с тремя уровнями.

Другой вид кэш-памяти применяется для буферизации данных при работе с жестким диском, с дисководом CD-ROM и другими устройствами с целью увеличения скорости обмена данными. Этот вид памяти формируется специальными системными программами, которые выделяют и используют часть ОЗУ как кэш-память. В данной памяти располагается используемая в данный момент часть содержимого дисков.

Способ повышения производительности за счет буферизации данных используют также некоторые контроллеры узлов и устройств ввода-вывода, прежде всего диски. Использование в составе встроенных контроллеров кэш-памяти позволяет значительно повысить скорость обмена данными.

2.3.5. Сопроцессоры

Сопроцессорами называют специализированные процессоры, рассчитанные на решение задач только в определённой области и работающих под управлением основного процессора. Они обрабатывают данные в формате, отличном от обычного представления целых чисел в двоичных кодах.

Главными в ряду специализированных процессоров являются **математический сопроцессор и векторный сопроцессор.**

Математический сопроцессор обрабатывает числа с плавающей запятой. Начиная с микропроцессора Intel 80486, сопроцессор располагается на одном кристалле с процессором. Модуль операций с плавающей запятой (или с плавающей точкой) по-английски обозначают аббревиатурой *FPU (floating point unit).*

Векторный процессор – это процессор, в котором операндами некоторых команд могут выступать упорядоченные массивы данных – векторы. В одном регистре процессора , например, располагаются сразу

все три составляющие вектора (X, Y и Z). Векторный процессор выполняет одну команду сразу над всеми элементами пары массивов в отличие от скалярных процессоров, которые могут работать только с парой операндов в единицу времени.

В большинстве современных микропроцессоров система команд имеет векторные расширения (SSE). Кроме того, современные видеокарты и физические ускорители можно рассматривать как векторные сопроцессоры.

К специализированным процессорам относится и ***цифровой сигнальный процессор*** (англ. *Digital signal processor DSP*). Он предназначен для цифровой обработки сигналов (обычно в реальном масштабе времени). В списке команд такого процессора должна быть операция «умножение с накоплением» для упрощения реализации быстрого преобразования Фурье.

2.3.6. Наборы инструкций SIMD

SIMD (англ. *single instruction, multiple data* – одиночный поток команд, множественный поток данных, ОКМД) – принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных.

SIMD-компьютеры состоят из одного командного процессора (управляющего модуля), называемого контроллером, и нескольких модулей обработки данных, называемых процессорными элементами. Управляющий модуль принимает, анализирует и выполняет команды. Если в команде встречаются данные, контроллер рассылает на все процессорные элементы команду и эта команда выполняется на нескольких или на всех процессорных элементах. Каждый процессорный элемент имеет свою собственную память для хранения данных. Одним из преимуществ данной архитектуры считается то, что в этом случае более эффективно реализована логика вычислений. До половины логических инструкций обычного процессора связано с управлением выполнением машинных команд, а остальная их часть относится к работе с внутренней памятью процессора и выполнению арифметических операций. В SIMD-компьютере управление выполняется контроллером, а «арифметика» отдана процессорным элементам.

SIMD-процессоры называются также векторными.

Использовать SIMD-процессоры фирма Intel начала с технологии **MMX.**

Аббревиатура MMX происходит от выражения MultiMedia eXtension – расширение для мультимедиа, которое реализовано фирмой Intel в своей серии процессоров MMX с тактовой частотой 166 и более МГц.

Процессор Pentium MMX отличается от «обычного» Pentium по шести основным пунктам:

1) добавлено 57 новых команд обработки данных;

2) увеличен в два раза объем внутреннего кэш (16 кб для команд и столько же – для данных);

3) увеличен объем буфера адресов перехода (Branch Target Buffer – BTB), используемого в системе предсказания переходов (Branch Prediction);

4) оптимизирована работа конвейера (Pipeline);

5) увеличено количество буферов записи (Write Buffers);

6) введено так называемое двойное электропитание процессора.

Набор из 57 новых команд и является основным отличием; остальные пять – не более, чем сопутствующие изменения. Хотя увеличенный объем кэш и внутренних буферов и оптимизированный конвейер несколько ускоряют работу любых приложений, однако основное увеличение производительности – до 60 % – возможно только при использовании программ, правильно применяющих технологию MMX в обработке данных.

MMX-команды позволяют значительно ускорить обработку только целочисленных данных и никак не используются при вычислениях с плавающей точкой. Но именно последние активно используются в 3D-приложениях, при выполнении которых загрузка процессора максимальна.

Рассмотрим подробнее процесс формирования компьютером 3D-изображения.

В формировании 3D-изображения участвуют два важнейших компонента компьютера – центральный процессор и графический адаптер, каждый из которых отвечает за свою часть вычислений. Процесс формирования 3D-изображения состоит из четырех этапов.

Первый этап – это физическое моделирование. Каждый объект описывается в виртуальном математическом пространстве. Важно заметить, что на этом этапе не учитывается взаимное перекрытие объектов, поскольку еще не определена точка взгляда (положение наблюдателя). Каждый объект существует как бы сам по себе – в своем пространстве

и в своей системе координат – и описывается строгими математическими формулами. В виде объектов просчитывается всё – все поверхности (стены, потолки, небо, земля и т.д.) и все действующие лица (люди, машины и и т.д.). Этот этап требует от процессора особенно интенсивных вычислений с плавающей точкой, поэтому он обычно выполняется центральным процессором системы.

Второй этап – геометрическое моделирование. На этом этапе все объекты собираются в едином виртуальном пространстве – единой системе координат. При этом учитывается взаимодействие объектов, формируются геометрические поверхности, рассчитывается освещенность каждого объекта. Именно на этом этапе из проволочной модели объектов формируются объемные поверхности, состоящие из треугольников. Так, шар превращается в набор треугольников, которые в совокупности выглядят как шар. Одновременно с этим происходит "клиппинг" – усечение частей объектов, скрытых другими объектами. Этот этап обработки также требует интенсивных вычислений с плавающей запятой, поэтому он тоже обычно производится центральным процессором.

Третий этап – треугольное проецирование. На этом этапе происходит перевод объемного виртуального мира в мир взгляда из одной точки. При этом активно используются вычисления как с целыми числами, так и с плавающей запятой. Обычно этот этап вычислений производится центральным процессором, однако некоторые наиболее "продвинутые" 3D-ускорители уже берут эту часть вычислений на себя.

И последняя операция – рендеринг. Именно во время рендеринга попиксельно вычисляется освещенность и цвет каждой точки изображения. В этот же момент происходит "натягивание" реалистичных текстур на объекты, что и позволяет получать настоящее трехмерное изображение. Для этого этапа характерны большие объемы целочисленных вычислений.

Первоначально предполагалось, что MMX-инструкции центрального процессора будут использоваться именно на этапе рендеринга, но в последнее время целочисленные вычисления на этом этапе выполняются графическим ускорителем.

Таким образом, при работе с 3D-графикой наиболее емкими по вычислительным затратам являются не реализованные в MMX операции по обработке целочисленных данных, а операции с плавающей запятой. Более того, возможности современных графических ускорителей вы-

росли настолько, что они берут на себя большую часть обработки целочисленных данных во время работы с 3D-графикой, а их специализированные чипы стали справляться с этой задачей значительно лучше, чем центральный процессор. В конце концов, "узким горлышком" всей системы при работе с 3D-графикой стала низкая скорость вычислений с плавающей запятой, выполняемых центральным процессором.

Для решения этой проблемы был предложен способ, аналогичный использованному при разработке технологии MMX. Поскольку обычно расчеты сводятся к однотипной обработке больших объемов однотипных же данных, то один из способов значительной экономии процессорного времени – создание таких инструкций процессора, при исполнении которых производится сразу несколько операций по обработке однотипных данных. Для целочисленных данных с этой целью был разработан набор MMX-инструкций (практически все они относятся к типу SIMD). Теперь надо было решить эту же проблему для чисел с плавающей запятой. Именно такое решение и предложила фирма AMD, разработав новую технологию 3DNow!, которая построена на основе набора SIMD-команд для вычислений с плавающей запятой.

Технология **3DNow!** имеет две особенности. Первая из них – уменьшение точности производимых вычислений. При расчете выводимых на экран пикселов нет никакой необходимости производить вычисления с высокой точностью. Так что принудительным образом ограничив точность 14 битами (по сравнению с 24-32 битами в традиционных вычислениях), можно получить значительное увеличение скорости расчетов (до трех тактов процессора по сравнению с 30 тактами для обычных команд, например, деления) при снижении точности, несущественном для данного типа вычислений. Вторая особенность — параллельное выполнение инструкций, то есть одновременно могут выполняться две инструкции из набора 3DNow!. В результате этих нововведений фирма AMD добилась весьма высоких результатов, нашедших свое отражение в первом чипе серии 3DNow! – процессоре AMD-K6-2.

Дальнейшее расширение наборов команд рассмотрено ниже.

SSE (англ. *Streaming SIMD Extensions*, потоковое SIMD-расширение процессора) – это SIMD набор инструкций, разработанный Intel и впервые представленный в процессорах серии Pentium III как ответ на аналогичный набор инструкций 3DNow! от AMD.

Технология SSE позволяла преодолеть 2 основные проблемы MMX – при использовании MMX невозможно было одновременно использовать инструкции сопроцессора, так как его регистры были общими с регистрами MMX, и возможность MMX работать только с целыми числами.

SSE включает в архитектуру процессора восемь 128-битных регистров и набор инструкций, работающих со скалярными и упакованными типами данных.

Преимущество в производительности достигается в том случае, когда необходимо произвести одну и ту же последовательность действий над разными данными. В таком случае блоком SSE осуществляется распараллеливание вычислительного процесса между данными.

Advanced Vector Extensions (AVX) – расширение системы команд x86 для микропроцессоров Intel и AMD, предложенное Intel в марте 2008.

AVX предоставляет различные улучшения, новые инструкции и новую схему кодирования машинных кодов.

Улучшения:

- Размер векторных регистров SIMD увеличивается с 128 (XMM) до 256 бит (регистры YMM0 – YMM15). Существующие 128-битные инструкции будут использовать младшую половину новых YMM регистров. В будущем возможно расширение до 512 или 1024 бит.
- Неразрушающие операции. Набор инструкций AVX позволяет использовать любую двухоперандную инструкцию XMM в трёхоперандном виде без модификации двух регистров-источников, с отдельным регистром для результата. Например, вместо a = a + b можно использовать c=a+b, при этом регистр a остаётся неизменённым. AVX не поддерживает неразрушающие формы операций над обычными регистрами общего назначения, такими как EAX, но такая поддержка, возможно, будет добавлена в последующих расширениях.
- Требования выравнивания данных для операндов SIMD в памяти ослаблены.

2.3.7. Увеличение числа ядер в процессоре

Ядра процессора выполняют работу независимо друг от друга. При условии, что вычислительную задачу можно распараллелить идеально, время вычисления будет обратно пропорционально числу задействованных ядер. Если **доля α от общего объёма** вычислений может быть

получена только последовательными расчётами, тогда **ускорение *S***, которое может быть получено на вычислительной системе из ***ρ* процессоров**, по сравнению с однопроцессорным решением не будет превышать величины

$$S_A = 1/(\alpha + (1 - \alpha)/\rho).$$

Закон сформулировал Амдал. Он показал, что рост производительности вычислительной системы с увеличением количества вычислителей ограничен объёмом вычислений, который может быть получен только последовательными расчётами.

Аналог закона Амдала – закон Густавсона – Барсиса. Оценку ускорения вычисляют по формуле:

$$S_B = \alpha + (1 - \alpha)\rho = \rho + (1 - \rho)\alpha.$$

Глава 3. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

3.1. Общие принципы ввода-вывода

Ввод-вывод с периферийных устройств в вычислительных машинах в отличие от чтения-записи памяти осуществляется командами IN (ввод) и OUT (вывод). Адреса регистров периферийных устройств располагаются в отдельном адресном пространстве и не влияют на размер адресного пространства ОЗУ.

Размер адресного пространства обусловлен числом проводов в шине адреса конкретного адресного пространства. То есть, сколько адресных проводов подходит к периферийному устройству.

Управляющие вычислительные машины (микроконтроллеры) не рассчитаны на большой объём вычислений и не требуют большой объём ОЗУ. Адреса регистров периферийных устройств в микроконтроллерах обычно располагаются в адресном пространстве ОЗУ. Команды ввода-вывода те же, что и для чтения-записи ячеек памяти, – MOV (перемещение).

Во всех вычислительных машинах, в том числе микроконтроллерах, есть хотя бы три типа периферийных устройств: таймеры-счётчики, параллельные и последовательные порты.

В ЭВМ применяют три режима ввода-вывода с периферийного устройства.

Программный ввод-вывод. Инициирование осуществляется процессором, а периферийные устройства играют сравнительно пассивную роль и сигнализируют только о своём состоянии, в частности о готовности к операции ввода-вывода.

Ввод-вывод по прерыванию. Инициирование осуществляется не процессором, а периферийным устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал, процессор переключается на подпрограмму, обслуживающую устройство, вызвавшее прерывание. Действия, выполняемые этой программой, определяются пользователем, а непосредственно вводом-выводом управляет процессор.

Прямой доступ к памяти. Используется, когда пропускной способности процессора недостаточно. В этом режиме работа процессора приостанавливается и он отключается от системных шин. Обмен данными между периферийным быстродействующим устройством и памятью осуществляет контроллер периферийного устройства.

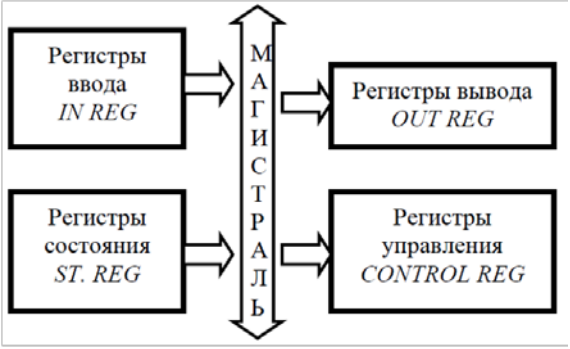

Рис. 3.1. Периферийное устройство с точки зрения программиста

С точки зрения программиста в общем случае периферийное устройство состоит из 4 групп регистров (рис. 3.1).

Периферийные устройства могут содержать в себе не все группы регистров. Регистр ввода и даже отдельный бит параллельного порта большинства микроконтроллеров с помощью управляющего регистра порта можно сделать выводящим. Регистров состояния (статуса) у параллельных портов практически нет. Например, в микроконтроллерах PIC18 есть только один бит **RBIF**, сигнализирующий об изменениях на одном из четырёх выводов (RB7-RB4) порта B.

3.2. Ввод-вывод в режиме прерывания

Этот режим обычно используется для медленно действующих устройств. Он позволяет резко сократить потери времени, возникающие за счёт циклов опроса готовности периферийного устройства при программном вводе-выводе.

Режим прерывания необходимо использовать для ввода информации, которая может быть потеряна при несвоевременном считывании. По прерыванию переходят к более важной или срочной программе.

Прерывания бывают аппаратные (внешние) и программные (внутренние) в зависимости от источника прерываний.

Аппаратные прерывания инициируются периферийными устройствами (мышь, клавиатура, таймеры, сетевая карта). Они могут возникнуть в любой момент (нажатие клавиши, перемещение мыши).

Внутренние прерывания, в свою очередь, делятся на синхронные и программные. Синхронные прерывания возникают непосредственно в процессоре при нарушении условий исполнения машинного кода (деление на ноль, обращение к недопустимым адресам памяти и так далее). Программные прерывания инициируются специальной инструкцией в коде программы (в процессорах x86 командой *INT nn*).

Общая последовательность реакции процессора на сигнал внешнего прерывания обычно содержит следующие действия:

1) Периферийное устройство генерирует сигнал прерывания, который передаётся на вход INT.

2) Процессор завершает текущую команду и, если прерывание разрешено, формирует сигнал INTA, подтверждающий прерывание. До получения этого сигнала устройство сохраняет активный уровень сигнала INT.

3) Осуществляется запоминание в стеке содержимого программного счётчика, аккумулятора и регистра флагов.

4) Процессор определяет прерывающее устройство для перехода к соответствующей подпрограмме.

5) Выполняется подпрограмма обслуживания прерывания. В начале подпрограммы запоминается в стеке содержимого некоторых регистров, используемых в программе прерывания.

6) Восстанавливается выполнение прерванной программы. Запомненное содержимое регистров извлекается из стеков.

7) Возобновляется выполнение прерванной программы. Это действие инициируется последней командой подпрограммы прерывания *RETI*. Команда возврата из прерывания возвращает из стека значения аккумулятора и регистра флагов. (В двухадресных процессорах аккумулятор отсутствует).

Пункты 3 и 6 называют контекстным переключением. Оно должно занимать минимальное время.

Аккумулятор в PIC-контроллерах называется рабочим регистром WREG. Стек служит только для хранения адресов возврата. Рабочий регистр (аккумулятор) WREG, регистр флагов STATUS, а также регистр указателя банков BSR временно сохраняются в трёх скрытых регистрах WS, STATUSS, BSRS.

Двухадресные процессоры не содержат в себе аккумулятор, поэтому в микроконтроллерах семейства AVR сохраняются и возвращаются только адрес возврата и регистр флагов.

3.2.1. Вектор прерывания

Закрепленный за устройством номер прерывания или связанный с этим номером адрес начала подпрограммы обслуживания устройства, вызвавшего прерывание, называется вектором прерывания.

Векторы прерываний объединяются в таблицу векторов прерываний, содержащую адреса обработчиков прерываний. Местоположение таблицы зависит от типа процессора.

Во многих микроконтроллерах адреса начала подпрограмм прерываний (вектора прерываний) расположены в начале программной памяти с нулевого адреса. В семействе AVR под каждый вектор отведёно четыре байта. В них записывается команда перехода (*JMP aaa)* по адресу *aaa* к подпрограмме обработки прерывания.

Всего два вектора прерываний имеет микроконтроллер PIC18. При возникновении прерывания с высоким приоритетом происходит переход по вектору 000008h, а при возникновении прерывания с низким приоритетом – 000018h.

3.2.2. Приоритеты прерываний и вложенные прерывания

Все устройства в зависимости от значимости имеют свой приоритет прерывания.

Запрещение прерываний на время обслуживания любого периферийного устройства может привести к потере прерываний быстродействующих высокоприоритетных устройств.

Прерывание программ обслуживания прерываний называется вложенным прерыванием (рис. 3.2).

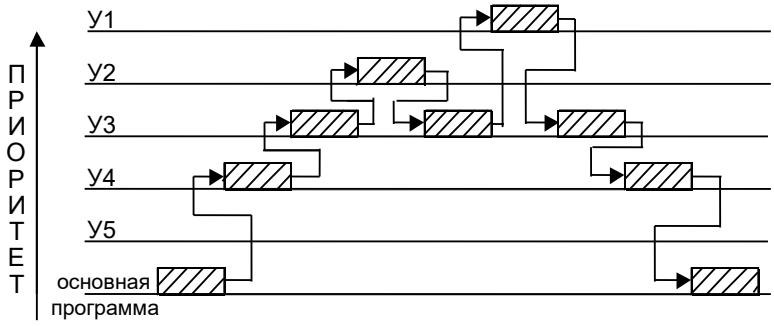

Рис. 3.2. Вложенные прерывания

В микропроцессоре бывает два вида прерываний: маскируемые INT и немаскируемые NMI (имеет самый высокий приоритет) прерывания.

Приоритет по входу INT можно устанавливать с помощью специального контроллера прерываний. В IBM PC существует 16 запросов прерываний: IRQ0….IRQ15.

Наибольший приоритет имеет IRQ0, затем 1, 2, 8…15, 3, 4…7.

Стандартно линии запросов прерываний подсоединены к элементам персонального компьютера PC/AT следующим образом:

- IRQ 0 – системные часы 18,2 Гц,
- IRQ 1 – клавиатура,
- IRQ 2 – блок прерывания,
- IRQ 8 – часы реального времени 1024 Гц,
- IRQ 10, 11, 12 – резерв,
- IRQ 13 – процессор,
- IRQ 14 – контроллер винчестера,
- IRQ 15 – резерв,
- IRQ 3 – COM2 или COM4,
- IRQ 4 – COM1 или COM3,
- IRQ 5 – LPT2,
- IRQ 6 – контроллер флоппи дисков,
- IRQ 7 – LPT1.

В интеловских микроконтроллерах семейства MCS-51 (советские МК-51) используется двухступенчатая структура приоритетов прерываний. Каждому источнику прерываний можно установить один из двух уровней приоритета: высокий и низкий. Запрос прерывания с высоким уровнем приоритета прерывает программу обработки прерывания с низким уровнем приоритета.

В пределах одного уровня приоритетов существует ещё одна структура приоритетов. В ней самый высокий приоритет имеет внешнее прерывание INT0. Затем идут прерывания TF0, INT1, TF1, TI+RI.

Микроконтроллеры семейства AVR используют аналогичную систему приоритетов прерываний.

PIC-контроллеры имеют простейшую систему приоритетов. Каждому прерыванию можно присвоить высокий или низкий приоритет. Подпрограмму с высоким приоритетом нельзя ничем прервать. В этот момент все прерывания запрещены. Прерывание низкого приоритета может прервать только запрос прерывания с высоким приоритетом. Как видим, вложенных прерываний в PIC-контроллерах практически нет.

Микроконтроллеры AVR и PIC в настоящее время выпускаются компанией Microchip

В семействе MCS-51 прерываниями управляют два регистра: IE и IP.

Регистр разрешения прерываний **IE** предназначен для разрешения или запрета прерываний.

IE

EA			ES	ET1	EX1	ET0	EX0

EA – разрешение/ запрет всех прерываний.

ES – разрешение/ запрет прерываний последовательного порта.

ET – разрешение/ запрет прерываний таймера счетчика.

EX – разрешение/ запрет INT0, INT1.

В микроконтроллере используется двухступенчатая структура приоритетов прерываний. Каждому источнику прерываний можно установить один из двух уровней приоритета: высокий и низкий. Запрос прерывания с высоким уровнем приоритета прерывает программу обработки прерывания с низким уровнем приоритета.

В пределах одного уровня приоритетов существует ещё структура приоритетов.

Источник	Приоритет внутри уровня
IE0	высший
TF0	
IE1	
TF1	
RI+TI	низший

Регистр приоритета прерываний IP предназначен для установки уровня приоритетов прерываний (1 – высокий уровень приоритета).

IP

			PS	PT1	PX1	PT0	PX0

PS – приоритет последовательного порта.

PT0, PT1 – приоритеты таймеров-счетчиков 0 и 1.

PX0 – приоритет запроса внешнего прерывания INT0.

PX1 – приоритет запроса внешнего прерывания INT1.

В начале памяти программ располагаются начала подпрограмм обслуживания прерываний. Обычно это команды переходов JMP(прыжок), потому что между векторами прерываний восемь ячеек программной памяти и целиком подпрограмма обслуживания не умещается. Если какое-либо прерывание не обслуживается, то по его вектору располагают команду возврата из прерывания на случай, если забыли это прерывание запретить. Макрокоманда ORG_<АДРЕС> макроассемблера ASM51 заставляет компилятор записать следующую за ней команду в программную память по адресу <АДРЕС>.

3.3. Ввод-вывод с прямым доступом к памяти

Разработаны две разновидности прямого доступа к памяти:

1. В режиме идентификации состояния памяти передача прямого доступа памяти (ПДП) выполняется без информирования контроллера. Для чего используются те интервалы машинного цикла, когда процессор не обращается к памяти, а выполняет внутреннее преобразование. Это уменьшает скорость передачи данных.

2. В режиме с пропуском тактов сигнал запроса ПДП заставляет процессор отключаться от системной шины на несколько тактов.

Первый режим использовался контроллером дисплея в домашних компьютерах Spectrum при чтении области видеопамяти оперативного запоминающего устройства.

В подавляющем числе случаев в персональных компьютерах для прямого доступа к памяти используется режим с пропуском тактов, при котором на вход процессора HOLD подается запрос ПДП (английская аббревиатура DMA). Если доступ разрешен, процессор выставляет на выводе HLDA активный уровень и отключается от магистрали.

Так как у процессора один вход запроса ПДП, то для увеличения линий запросов ставится контроллер. С помощью такого контроллера в IBM PC/XT число линий запросов прямого доступа к памяти увеличено до 4 (DRQ0..DRQ3), в IBM PC/AT – до семи.

Чаще всего используется канал 0. Каждые 15 микросекунд происходит регенерация оперативной памяти путем опроса следующей строки в матрице ячеек памяти в каждой микросхеме ОЗУ. В этот момент процессор по запросу DRQ0 отключается от системной магистрали и управление всеми шинами берет на себя контроллер ОЗУ.

Другие каналы прямого доступа к памяти используются дисководами для быстрого обмена данными с ОЗУ.

Глава 4. ПАМЯТЬ ЭВМ

4.1. Микросхемы запоминающих устройств

В микроЭВМ существует два типа памяти: ***постоянная и оперативная***. Поэтому конструктивно блок памяти состоит из постоянного запоминающего устройства и оперативного запоминающего устройства (ПЗУ и ОЗУ). Процессы записи и хранения информации в этих устройствах принципиально отличаются. В режиме чтения они неотличимы. При выключенном питании в ПЗУ информация не исчезает. В ОЗУ же для хранения данных и программ наличие напряжения питания обязательно, при его отсутствии информация теряется. Во время работы ЭВМ микропроцессор может записывать информацию только в ОЗУ, читать как из ПЗУ, так и из ОЗУ. ПЗУ используют для хранения программ запуска ЭВМ. Общее название таких программных средств – программа монитор. В IBM PC совместимых персональных компьютерах в ПЗУ находится BIOS – базовая система ввода-вывода. Записанные в BIOS программы, кроме контроля начальной загрузки компьютера, осуществляют управление устройствами ввода-вывода, входящими в состав ЭВМ.

Оперативное запоминающее устройство в англоязычных странах называют *random – access memory* (RAM) (запоминающее устройство с произвольной выборкой ЗУПВ), а постоянное запоминающее устройство *read only memory* (ROM) (память только для чтения). Однако, строго говоря, к устройствам с произвольной выборкой (произвольным доступом) относятся и ПЗУ, так как все его ячейки адресуемы и могут быть опрошены в произвольном порядке.

Ячейка, способная запоминать один бит информации, является основным элементом памяти. Микросхемы запоминающих устройств создают в виде двумерных матриц таких элементов. Отдельная ячейка однозначно определяется адресами строки и столбца. Адрес столбца получают декодированием старших разрядов адреса ячейки, адрес строки – младших.

ОЗУ – энергозависимые запоминающие устройства (*volatile memory*). Их используют для временного хранения программ и данных. Микросхемы ОЗУ подразделяют на ***статические и динамические***.

Микросхемы ***статического*** ОЗУ состоят из ячеек, построенных на триггерах, называемых также электронными защелками.

В ***динамических*** ОЗУ элементы памяти выполнены на основе электрических конденсаторов, сформированных внутри кристалла полупроводника. Из-за наличия токов утечки заряд конденсатора медленно изменяется и содержимое ячейки необходимо регулярно обновлять. Восстановление уровня сигналов в такой памяти осуществляется с помощью периодических циклов регенерации, во время которых информация в ячейках перезаписывается. Период регенерации обычно составляет одну - две миллисекунды. За один цикл регенерации происходит восстановление данных не в одной ячейке, а во всех элементах памяти, расположенных в строке. Поэтому за время, не превышающее период регенерации, необходимо перебрать все строки в микросхеме памяти.

Преимущество динамического ОЗУ по сравнению со статическим – относительно малая стоимость одного бита информации. Статическая же память обладает большим быстродействием, так как нет необходимости в циклах регенерации.

Постоянные запоминающие устройства по способу загрузки в них информации подразделяются на масочные (ПЗУМ), однократно программируемые (ППЗУ) и репрограммируемые (РПЗУ).

Функции элементов памяти в ПЗУ выполняют перемычки в виде проводников, диодов или транзисторов. В ***ПЗУ масочного типа*** перемычки формируют через маску на заключительной технологической стадии изготовления микросхемы. При объеме заказа более 10 тысяч микросхем масочные ПЗУ дешевле других типов постоянной памяти.

Однократно программируемые ПЗУ применяют только тогда, когда из-за размеров партии программирование на заводе-изготовителе нецелесообразно. Запись информации в ППЗУ осуществляет пользователь пережиганием электрическими импульсами легкоплавких перемычек, играющих роль элементов памяти.

ПЗУМ и ППЗУ (их английские аббревиатуры ROM и PROM) допускают лишь однократное программирование, что значительно сужает область их применения. Основная область применения такой памяти – память программ в однокристальных микроЭВМ и микроконтроллерах. В первых ноутбуках не было накопителей на жестких магнитных дисках и дисковая операционная система хранилась в однократно программируемых микросхемах памяти.

Многократное перепрограммирование допускают ***репрограммируемые ПЗУ*** – РПЗУ (EPROM). Элементом памяти в РПЗУ является МДП

– транзистор, обладающий свойством переходить в состояние проводимости под воздействием импульса программирующего напряжения и сохранять это состояние длительное время. Данный эффект обусловлен накоплением электрического заряда в подзатворном диэлектрике. Для стирания информации перед новым циклом программирования необходимо вытеснить накопленный под затвором заряд. РПЗУ по методу стирания подразделяют на два вида: со стиранием ***ультрафиолетовым светом*** и стиранием ***электрическим сигналом*** (EEPROM).

Следующим этапом развития электрически программируемой постоянной памяти являются микросхемы ***Flash памяти***. С целью упрощения и удешевления схемы запись в нее может производиться только постранично.

Для записи в современные EEPROM не требуется повышенного напряжения 27 или 12 В, поэтому запись и перезапись в них можно производить после установки микросхемы в аппаратуру ***(In-System Programmable Memory, ISP Memory)***.

Графические изображения микросхем памяти, взятые из библиотек символьных элементов САПР P-CAD 8.5, представлены на рис. 4.1– 4.5. Начертание несколько отличается от требований ГОСТов, что не удивительно.

Микросхемы памяти могут быть с одноразрядной организацией, когда в адресуемой ячейке памяти хранится один разряд данных (рис. 4.2), и со словарной организацией, при которой число входов-выходов данных больше одного (рис. 4.1).

Чтобы сократить число выводов микросхемы и упростить ее подключение к шине данных входы (I) и выходы (O) одного разряда памяти соединяют между собой (контакты I/O на рис. 4.1). Направление передачи данных, то есть входы или выходы памяти подключены к выводам I/O, определяется логическим уровнем на контакте W/R (запись/считывание).

Мультиплексирование кода адреса также позволяет уменьшить число выводов, однако при этом усложняется схема управления памятью.

Как используется мультиплексирование – рассмотрим на примере организации работы микросхемы 565РУ5 (рис. 4.2). Ввод адреса в нее происходит в два этапа. Вначале вводят код адреса строки A0-A7 и фиксируют его во входном регистре стробирующим сигналом RAS, затем вводят код адреса столбца A8-A15, фиксируя его сигналом CAS.

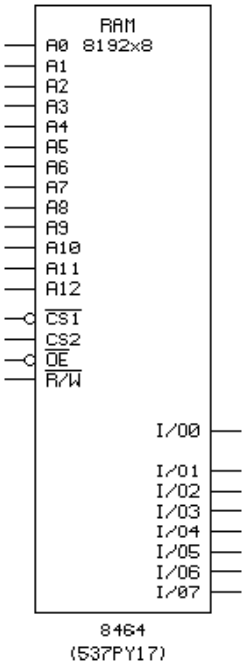

Рис. 4.1. Статическое ОЗУ
со словарной организацией

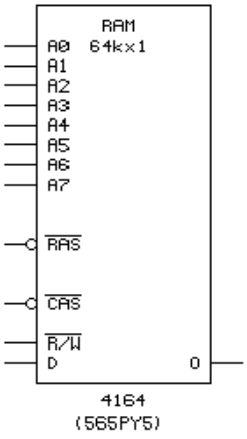

Рис. 4.2. Динамическое ОЗУ
с одноразрядной организацией
и мультиплексированием кода адреса

Контакт выбора микросхемы CS (Chip Select) необходим для создания памяти большего объема, чем позволяет разрядность шины адреса одной микросхемы. Он дает возможность подключать или полностью отключать микросхему от шин адреса и данных. Отключение и подключение выходов только к шине данных можно производить сигналом OE (Output Enable). Время задержки включения и выключения при этом меньше, чем при использовании сигнала CS.

В репрограммируемых постоянных запоминающих устройствах с электрическим стиранием чтение, стирание и запись выполняются с использованием одного источника питания.

Микросхемы могут быть с ***параллельным и последовательным*** вводом-выводом информации.

В первом случае (Parallel EEPROM) код адреса вводится, а код данных вводится и выводится параллельно. Для каждого разряда кода есть свой вывод (рис.4.3–4.5).

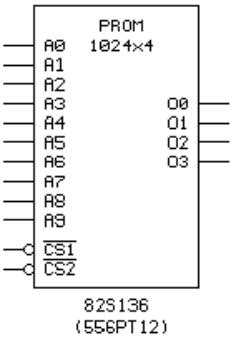

Рис. 4.3. Однократно программируемое пользователем ПЗУ

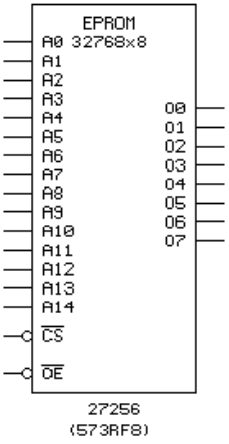

Рис. 4.4. Репрограммируемое ПЗУ с ультрафиолетовым стиранием

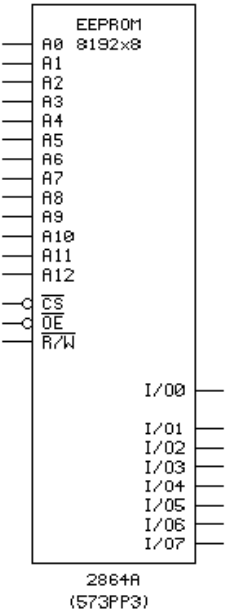

Рис. 4.5. Репрограммируемое ПЗУ со стиранием электрическим сигналом

Во втором случае (Serial EEPROM) коды адреса и коды данных, а также идущие первыми байты управления, вводятся последовательно, разряд за разрядом, с использованием одного вывода микросхемы. Код считанных из памяти данных выводится через тот же или другой вывод корпуса. Обмен информацией происходит под управлением синхросигнала, поступающего в микросхему извне.

Микросхемы памяти, имеющие **два вывода** для обмена информацией (2-Wire Serial EEPROM), используют интерфейс IIC (*Inter-Integrated Circuit*), чаще обозначаемый аббревиатурой **I2C** (рус. *ай-ту-си/и-два-цэ/и-два-си*).

Интерфейс **SPI** (*Serial Peripheral Interface*) применяется при передаче информации между микросхемой памяти и микроконтроллером **по трём проводам**. Такие микросхемы относят к классу SPI Serial EEPROM. С интерфейсом SPI также работают микросхемы Serial Data-

Flash. Интерфейс SPI имеет четыре варианта работы. Поэтому при разработке программы для микроконтроллера следует учитывать какой тип SPI (*SPI mode*) использует микросхема памяти для обмена.

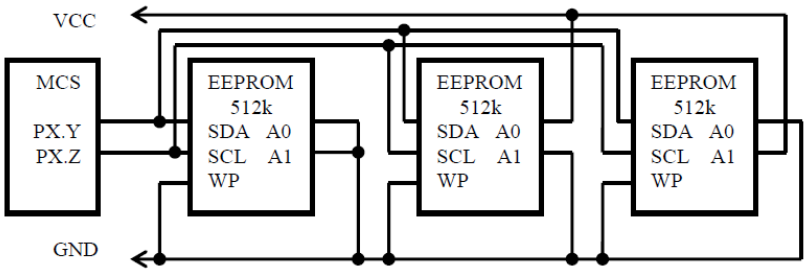

Рис. 4.6. Схема подключения трех микросхем 2-Wire Serial EEPROM типа AT24C512

На рис. 4.6 и 4.7 показано подключение к микроконтроллеру нескольких микросхем памяти с интерфейсами **I2C** и **SPI** соответственно.

Контакты микросхем используются для следующих функций:

- вход-выход SDA (Serial Data) для последовательного ввода-вывода битов;
- вход SCL или SCK (Serial clock) для ввода синхросигнала из микроконтроллера;
- вход SI (Serial Input) для последовательного приема битов;
- вход SO (Serial Output) для последовательной выдачи битов;
- вход WP (Write Protect) для запрета записи в память;
- вход CS (Chip Select) для перевода микросхемы в активное состояние.

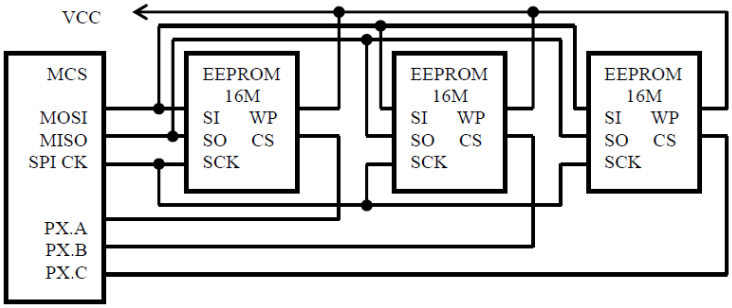

Рис. 4.7. Схема подключения трех микросхем типа AT45D161 к интерфейсу **SPI**

Входы A0 и A1 предназначены для присвоения микросхеме памяти номера, по которому микроконтроллер отличает ее от других микросхем памяти, подключенных к линиям SDA и SCL. Номер микросхемы в двоичном коде задается путем соединения входов A0 и A1 с шиной GND (0) или VCC (1). Возможное число одновременно подключенных к микроконтроллеру микросхем зависит от типа микросхемы памяти. Обычно можно подключить четыре или восемь чипов.

Микросхемы, использующие память Serial EEPROM, позволяют делать запись и перезапись после установки микросхемы в аппаратуру (In-System Programmable Memory, **ISP** Memory). Это широко используется для программирования памяти программ в микросхемах микроконтроллеров.

4.2. Модули оперативной памяти персональных компьютеров

В оперативной памяти компьютеров используются микросхемы динамической памяти (DRAM). Рабочая частота и тайминги являются основными характеристиками DRAM.

Адресное пространство модуля памяти разбивается на банки, пространство банка – на страницы. Страница структурно выполнена в виде матрицы. Контроллер памяти вначале кроме адресов банка и страницы передаёт на модуль памяти номер строк, затем – через промежуток времени – номер колонок матрицы. Большой промежуток времени тратится на открытие и закрытие банка. Время, потраченное на каждое действие, называется **таймингом**.

Основными **таймингами (задержками)** DRAM являются:

- задержка между подачей номера строки и номера столбца, называемая временем полного доступа (англ. *RAS to CAS delay*);
- задержка между подачей номера столбца и получением содержимого ячейки, называемая временем рабочего цикла (англ. *CAS Latency*);
- задержка между чтением последней ячейки и подачей номера новой строки (англ. *RAS Precharge Time*).

Чаще всего тайминги измеряются в тактах рабочей частоты и для краткости записываются ***тремя числами*** по порядку: *CAS Latency, RAS to CAS Delay* и *RAS Precharge Time*. Чем меньше величина тайминга в наносекундах, тем быстрее будет работать оперативная память.

Когда указывается только ***одно число*** (например, в характеристике модуля памяти указано, что **латентность** равна CL16), то имеется ввиду первый параметр, то есть *CAS Latency*.

Формула таймингов для памяти может состоять из ***четырёх цифр***, например 16-18-18-35. Последний параметр называется «DRAM Cycle Time Tras/Trc». Он определяет отношение интервала, в течение которого строка открыта для переноса данных (tRAS — RAS Active time), к периоду, в течение которого завершается полный цикл открытия и обновления ряда (tRC — Row Cycle time). «DRAM Cycle Time Tras/Trc» также называют циклом банка (Bank Cycle Time).

На планке памяти DIMM располагаются микросхемы динамической памяти и чип **SPD**. В микросхеме **SPD** хранится информация о рекомендуемых значениях таймингов для наиболее распространенных частот системной шины. Компьютер читает эти данные на этапе самотестирования, задолго до загрузки операционной системы, что позволяет настроить параметры обращения к памяти даже при одновременном наличии в системе разномастных модулей памяти и доступна чипсету.

Расширением SPD является технология **XMP** – **eXtreme Memory Profile** ("экстремальные профили памяти"). В специальной микросхеме записывают дополнительный набор данных о доступных для модуля частотах, таймингах и напряжениях. Материнская плата, на которой устанавливается память с **XMP**, должна поддерживать эту технологию.

Целостность данных в модуле памяти раньше проверялась **контролем чётности** (**Parity**). В настоящее время выявлением и исправлением ошибок занимается алгоритм **ECC (Error Correct Code).** В отличие от контроля чётности каждый бит в этом алгоритме входит более чем в одну контрольную сумму, что позволяет в случае возникновения ошибки в одном бите определить адрес ошибки и исправить ее. Как правило, ошибки в двух битах также детектируются, хотя и не исправляются. Для реализации этих возможностей на модуль устанавливается дополнительная микросхема памяти и он становится 72-разрядным в отличие от 64 разрядов данных обычного модуля.

Запись в память большой емкости занимает длительное время. Некоторые модули (как правило, для серверов) снабжаются специальной микросхемой (**буфером**), которая сохраняет поступившие данные относительно быстро, что освобождает контроллер. Модули с частичной буферизацией называются также "**регистровыми" (Registered),** а модули **с полной буферизацией (Full Buffered)** – FB-DIMM.

Во всех буферизованных модулях памяти при использовании большого количества микросхем памяти применяется **PLL** или Phase Locked Loop – (цепь автоподстройки частоты и фазы сигнала). **PLL** служит для снижения электрической нагрузки на контроллер памяти и повышения стабильности работы.

4.3. Типы компьютерной DRAM памяти

Раздел написан по интернет-источникам.

Страничная память

Страничная память (англ. Page Mode DRAM, **PM DRAM**) являлась одним из первых типов выпускаемой в начале 1990-х годов компьютерной оперативной памяти.

Быстрая страничная память

Быстрая страничная память (англ. Fast Page Mode DRAM, **FPM DRAM**) появилась в 1995 году. Увеличение скорости работы достигалось путём повышенной нагрузки на аппаратную часть памяти. Данный тип памяти в основном применялся для компьютеров с процессорами Intel 80486 или аналогичных процессоров других фирм. Память работала на частотах 25 и 33 МГц.

EDO DRAM – память с усовершенствованным выходом

С появлением процессоров Intel Pentium, у которого 64-битная шина данных, память FPM DRAM оказалась совершенно неэффективной. Для процессоров Pentium в 1996 году создали память с усовершенствованным выходом (англ. Extended Data Out DRAM**, EDO DRAM**). Её производительность оказалась на 10-15 % выше по сравнению с памятью типа FPM DRAM. Рабочая частота была 40 и 50 МГц. Эта память содержит регистр-защелку (англ. data latch) выходных данных, что обеспечивает некоторую конвейеризацию работы для повышения производительности при чтении.

SDR SDRAM – синхронная DRAM

В связи с увеличением частоты системной шины, стабильность работы памяти типа EDO DRAM стала заметно падать. Ей на смену пришла **синхронная** память – *Single Data Rate Synchronous Dynamic Random Access Memory* (**SDR SDRAM**). Новыми особенностями этого типа

памяти являлись использование тактового генератора для **синхронизации** всех сигналов и использование конвейерной обработки информации. Также память надёжно работала на более высоких частотах системной шины (100 МГц и выше).

Если для FPM и EDO памяти указывается время чтения первой ячейки в цепочке (время доступа), то для SDRAM указывается время считывания последующих ячеек. Цепочка – несколько последовательных ячеек. На считывание первой ячейки уходит довольно много времени (60-70 нс) независимо от типа памяти, а вот время чтения последующих сильно зависит от типа. Рабочие частоты этого типа памяти могли равняться 66, 100 или 133 МГц, время полного доступа – 40 и 30 нс, а время рабочего цикла – 10 и 7,5 нс.

Enhanced SDRAM (ESDRAM)

Для преодоления некоторых проблем с задержкой сигнала, присущих стандартной DRAM-памяти, было решено встроить небольшое количество SRAM в чип, то есть создать на чипе кеш.

ESDRAM – это, по существу, SDRAM с небольшим количеством SRAM. При малой задержке и пакетной работе достигается частота до 200 МГц. Как и в случае внешней кеш-памяти, SRAM-кеш предназначен для хранения и выборки наиболее часто используемых данных. Отсюда и уменьшение времени доступа к данным медленной DRAM.

Одним из таких решений являлась ESDRAM от Ramtron International Corporation.

Пакетная EDO RAM

Пакетная память **EDO RAM** (англ. *burst extended data output* DRAM, BEDO DRAM) стала дешёвой альтернативой памяти типа SDRAM. Основана на памяти EDO DRAM, её ключевой особенностью являлась технология поблочного чтения данных (блок данных читался за один такт), что сделало её работу быстрее, чем у памяти типа SDRAM. Однако невозможность работать на частоте системной шины более 66 МГц не позволила данному типу памяти стать популярным.

Video RAM

Специальный тип оперативной памяти – Video RAM (**VRAM**) – был разработан на основе памяти типа SDRAM для использования в видеоплатах. Он позволял обеспечить непрерывный поток данных в процессе обновления изображения, что было необходимо для реализации

изображений высокого качества. На основе памяти типа VRAM появилась спецификация памяти типа Windows RAM (WRAM), иногда её ошибочно связывают с операционными системами семейства Windows. Её производительность стала на 25 % выше, чем у оригинальной памяти типа SDRAM, благодаря некоторым техническим изменениям.

DDR SDRAM

По сравнению с обычной памятью типа **SDR SDRAM**, в памяти SDRAM с удвоенной скоростью передачи данных (англ. *Double Data Rate* SDRAM, DDR SDRAM или SDRAM II) была вдвое увеличена пропускная способность. Первоначально память такого типа применялась в видеоплатах, но позднее появилась поддержка DDR SDRAM со стороны чипсетов.

У всех предыдущих DRAM были разделены линии адреса, данных и управления, которые накладывают ограничения на скорость работы устройств. Для преодоления этого ограничения в некоторых технологических решениях все сигналы стали выполняться на одной шине. Двумя из таких решений являются технологии DRDRAM и SLDRAM. Они получили наибольшую популярность и заслуживают внимания. Стандарт SLDRAM является открытым и, подобно предыдущей технологии, SLDRAM использует оба перепада тактового сигнала. Что касается интерфейса, то SLDRAM перенимает протокол, названный SynchLink Interface и стремится работать на частоте 400 МГц.

Память DDR SDRAM работает на частотах в 100, 133, 166 и 200 МГц, её время полного доступа – 30 и 22,5 нс, а время рабочего цикла – 5, 3,75, 3 и 2,5 нс.

Так как частота синхронизации лежит в пределах от 100 до 200 МГц, а данные передаются по 2 бита на один синхроимпульс как по фронту, так и по спаду тактового импульса, то эффективная частота передачи данных лежит в пределах от 200 до 400 МГц. Такие модули памяти обозначаются DDR200, DDR266, DDR333, DDR400.

Специфическим режимом работы модулей памяти является двухканальный режим. Этот режим позволяет считывать информацию одновременно с двух модулей памяти, так как компенсирует разницу во время прихода данных с них.

Тип памяти **RDRAM** (Direct RDRAM или Direct Rambus DRAM) является разработкой компании Rambus. Высокое быстродействие этой памяти достигается рядом особенностей, не встречающихся в других типах памяти. Первоначальная очень высокая стоимость памяти RDRAM привела к тому, что производители мощных компьютеров

предпочли менее производительную, зато более дешёвую память DDR SDRAM. Рабочие частоты памяти – 400, 600 и 800 МГц, время полного доступа – до 30 нс, время рабочего цикла – до 2,5 нс. RDRAM используется в приставках PlayStation 2 и PlayStation 3.

Конструктивно иной тип оперативной памяти **DDR2 SDRAM** был выпущен в 2004 году. Основываясь на технологии DDR SDRAM, этот тип памяти за счёт технических изменений имеет более высокое быстродействие. Память может работать с тактовой частотой шины 200, 266, 333, 337, 400, 533, 575 и 600 МГц. При этом эффективная частота передачи данных соответственно равна 400, 533, 667, 675, 800, 1066, 1150 и 1200 МГц. Время рабочего цикла – от 5 до 1,67 нс.

Внешнее отличие модулей памяти DDR2 от DDR – 240 контактов (по 120 с каждой стороны).

DDR3 SDRAM тип памяти основан на технологиях DDR2 SDRAM со вдвое увеличенной частотой передачи данных по шине памяти. Отличается пониженным энергопотреблением по сравнению с предшественниками. Частота полосы пропускания лежит в пределах от 800 до 2400 МГц.

DDR4 SDRAM (англ. *double-data-rate four synchronous dynamic random access memory*) – четвёртое поколение DDR (2014 год) тип оперативной памяти. Развитие предыдущих поколений DDR (DDR, DDR2, DDR3).

В DDR4 удвоено число банков и скорость передачи. Она работает при напряжении питания от 1,2 V и 1.4 V с частотой от 800 до 2133 МГц. Введение механизма контроля чётности на шинах адреса и команд повышает надёжность работы. Поддерживает эффективные частоты от 2133 до 4266 МГц.

Стандарт позволяет емкость одного модуля DIMM DDR4 до 64 ГБ. Память DDR4 имеет 288-контактные DIMM-модули, схожие по внешнему виду с 240-контактными DIMM DDR-2/DDR-3. Контакты расположены более близко (0,85 мм вместо 1,0), чтобы разместить их на стандартном 5,25-дюймовом (133,35 мм) слоте DIMM. Высота увеличивается незначительно (31,25 мм вместо 30,35).

DDR4 модули SO-DIMM для портативных компьютеров содержат 260 контактов (а не 204, как у SO-DIMM DDR-3), которые расположены ближе (0,5 мм, а не 0,6). Модуль стал на 1,0 мм шире (68,6 мм вместо 67,6), но сохранил ту же высоту – 30 мм.

GDDR5 (англ.*Graphics Double Data Rate*) – 5 поколение ГРАФИЧЕСКОЙ памяти DDR SDRAM, спроектированной для приложений, требующих высокой полосы пропускания. В отличие от его предшественника, GDDR4, GDDR5 основан на памяти DDR3, которая имеет удвоенные по сравнению с DDR2, DQ (Digital Quest) каналы связи, но у GDDR5 также есть буферы предварительной выборки шириной 8 битов, как у GDDR4.

4.4. Форм-факторы модулей памяти

Оперативная память, устанавливаемая в первых персональных компьютерах, выполнялась либо в виде отдельных микросхем в корпусах типа **DIP** (*Dual In line Package* – корпуса с двухрядным расположением выводов), либо в виде модулей памяти типа **SIP** (*Single In line Package*)/**SIPP** (*Single In line Pinned Package* – модуль с одним рядом проволочных выводов).

Затем стали использовать модули типа **SIMM** (*Single In line Memory Module* – модуль памяти с одним рядом контактов) и **DIMM** (*Dual In line Memory Module* – модуль памяти с двумя рядами контактов).

Модули памяти представляют собой небольшие печатные платы с установленными на них микросхемами памяти в корпусах для поверхностного монтажа типа **SOP** (Small Outline Package). При этом для подключения к системной плате на **SIMM** и **DIMM**-модулях используется печатный (ножевой) разъем, а на **SIPP**-модулях – штыревой.

Модули памяти типа **SDRAM** (***S**ynchronous **D**ynamic **R**andom **A**ccess **M**emory* – синхронная динамическая память с произвольным доступом) наиболее распространены в виде 168-контактных DIMM-модулей, памяти типа **DDR SDRAM** – в виде 184-контактных, а модули типа **DDR2**, **DDR3** и **FB-DIMM SDRAM** – 240-контактных модулей. Память **DDR4** имеет 288-контактные DIMM-модули.

Для портативных и компактных устройств (материнских плат форм-фактора Mini-ITX, ноутбуков, таблетов и т. п.), а также принтеров, сетевой и телекоммуникационной техники и пр. широко применяются конструктивно уменьшенные модули **DRAM** (как **SDRAM**, так и **DDR SDRAM**) – **SO-DIMM** (Small outline DIMM) – аналоги модулей **DIMM** в компактном исполнении для экономии места. Модули **SO-DIMM** существуют в 72-, 100-, 144-, 200-, 204- и 260 -контактном исполнении.

4.5. Выбор модуля памяти компьютера

Параметры, по которым выбирается оперативная память:

- **Производитель.**
- **Объём памяти.**
- **Количество модулей в комплекте.**
- **Тип памяти,** например, ***Registered DDR4*** или ***DDR3 ECC.***
- **Стандарт**, например, ***PC4-25600 (DDR4 3200 МГц)*** означает ***25600*** – пропускная способность в Мбайт/с. Память типа ***DDR4*** с тактовой частотой ***3200 МГц.***
- **Тактовая частота памяти.**
- Цвет подсветки.
- **Memory rank**. На одной планке (плате) располагаются не один, а два или четыре модуля памяти. Используется в серверах для создания ОЗУ большого объёма при ограниченном количестве слотов.
- **VLP very low profile**. Маленькая высота модуля памяти.
- **Охлаждение памяти**. Есть радиатор или нет.
- **Двухсторонний модуль памяти**. Толщина модуля увеличена.
- **Тайминги** (см. раздел 4.2).
- **XMP.** Наличие в модуле специальной микросхемы с набором данных о доступных для модуля частотах, таймингах и напряжениях. Подробнее в разделе 4.2.
- **Напряжение питания**.

Глава 5. МАГИСТРАЛЬ

Магистраль состоит из набора проводников, по которым осуществляется обмен двоичными данными. В магистрали входят три шины:

- ***Шина адреса***. В простых микропроцессорных системах только процессор вырабатывает адрес. Шина адреса однонаправленная. Кроме микропроцессора адрес могут выставлять устройства, имеющие прямой доступ к памяти. Количества линий адреса соответствует разрядности кода адреса (16, 20, 24, 32 разряда и больше).
- ***Шина данных***. По шине данных передаются коды, которыми обмениваются устройства системы. Шина двунаправленная (можно читать и писать), её разрядность 8, 16, 32, 64.
- ***Шина управления и состояния.*** По шине управления передаются управляющие сигналы, предназначенные для синхронизации. Они задают последовательность и начало срабатывания различных устройств в системе.

5.1. Основные сигналы магистрали

Для того чтобы передать данные по магистрали, надо описать приемник (адрес), сами данные, момент, когда данные являются достоверными. Магистраль должна иметь ША, ШД и несколько линий синхронизации, которые сообщают о том, когда и в каком направлении передаются данные.

В шине управления есть, по крайней мере, два провода для синхронизации.

Синхронизацию можно осуществить двумя путями:

1. На одной линии выставляется **сигнал чтения/записи (R/W)**. Высокий уровень сигнала говорит о том, что процессор находится в состоянии чтения, а низкий – записи. По второй линии идёт **сигнал STROBE**, активный уровень которого сигнализирует устройству системы о готовности процессора к чтению или записи (рис. 5.1).

Рис. 5.1. Первый вариант управления чтением/записью

2. Сигнал чтения **RD** и сигнал записи **WR** идут по ***отдельным*** проводам (рис. 5.2).

Рис. 5.2. Второй вариант управления чтением/записью

Современные персональные компьютеры имеют в своем составе не одну, а несколько магистралей. Магистрали имеют разную скорость, а следовательно – сложность и цену.

По одним магистралям байты передаются параллельно, то есть байт целиком за один такт (восемь бит по восьми проводам), по другим – последовательно бит за битом по одному проводу.

Последовательная передача требует синхронизации. Наибольшая скорость последовательной передачи достигается введением специального второго провода, по которому передаются синхроимпульсы. Приход синхроимпульса говорит, что передаётся очередной бит информации по первой линии. Синхронная последовательная передача является одним из режимов порта **USART** (***U****niversal* ***S****ynchronous/****A****synchronous* ***R****eceiver-****T****ransmitter* – универсальный синхронный/асинхронный приемник – передатчик) в микроконтроллерах. На синхронной передаче построены интерфейсы **PS/2, SPI** (*Serial Peripheral Interface* – последовательный периферийный интерфейс) и **I2C** (*Inter-Integrated Circuit*).

В асинхронном режиме для синхронизации в посылку (обычно байт) добавляют стартовый и стоповый биты. Передача происходит по одному проводу на заранее оговорённой скорости. Кроме последовательного порта **USART/UART** в асинхронном режиме работают интерфейсы **USB, Fire Wire, RS-232, RS-485, SATA, PCI Express** и другие.

Асинхронный режим используют на больших скоростях передачи, потому что на гигагерцовых частотах трудно сделать две линии с одинаковым временем задержки. Информация и синхроимпульсы приходят в разное время.

Скорость в асинхронном режиме падает по сравнению с синхронным режимом на двадцать и более процентов из-за дополнительных неинформационных битов. Выигрыш – в вдвое меньшим числе проводов и проблематичности синхронной передачи данных на сверхвысоких частотах.

Конфигурация персонального компьютера с несколькими типами магистралей (рис. 5.3) позволяет сочетать высокую производительность компьютера с программной совместимостью широкого спектра контроллеров и узлов персонального компьютера, а также оптимизировать работу подсистем персональных компьютеров, обладающих разными скоростными и электрическими характеристиками.

Контроллеры магистралей конструктивно объединяют в большие интегральные схемы – северный и южный мосты. Быстродействующие контроллеры, входящие в состав северного моста, включаются в состав микропроцессоров. Поэтому на современных системных платах устанавливаются только южные мосты.

В компьютере магистрали часто называют шинами или интерфейсами. Их можно разбить на пять групп: шины расширения, интерфейсы накопителей, кабельные интерфейсы, видеоинтерфейсы и беспроводные интерфейсы.

Слоты **шин расширения** расположены на материнской плате. Через них подключатся платы, расширяющие возможности компьютера. В настоящее время применяется шина ***PCI Express*** с различным числом каналов последовательной передачи данных (от одного до шестнадцати).

Интерфейсы накопителей соединяют материнскую плату с дисководами оптических дисков и магнитных дискет, с накопителями на жёстких дисках и с твердотельными накопителями. Максимальная скорость у третей версии интерфейса ***Serial ATA***.

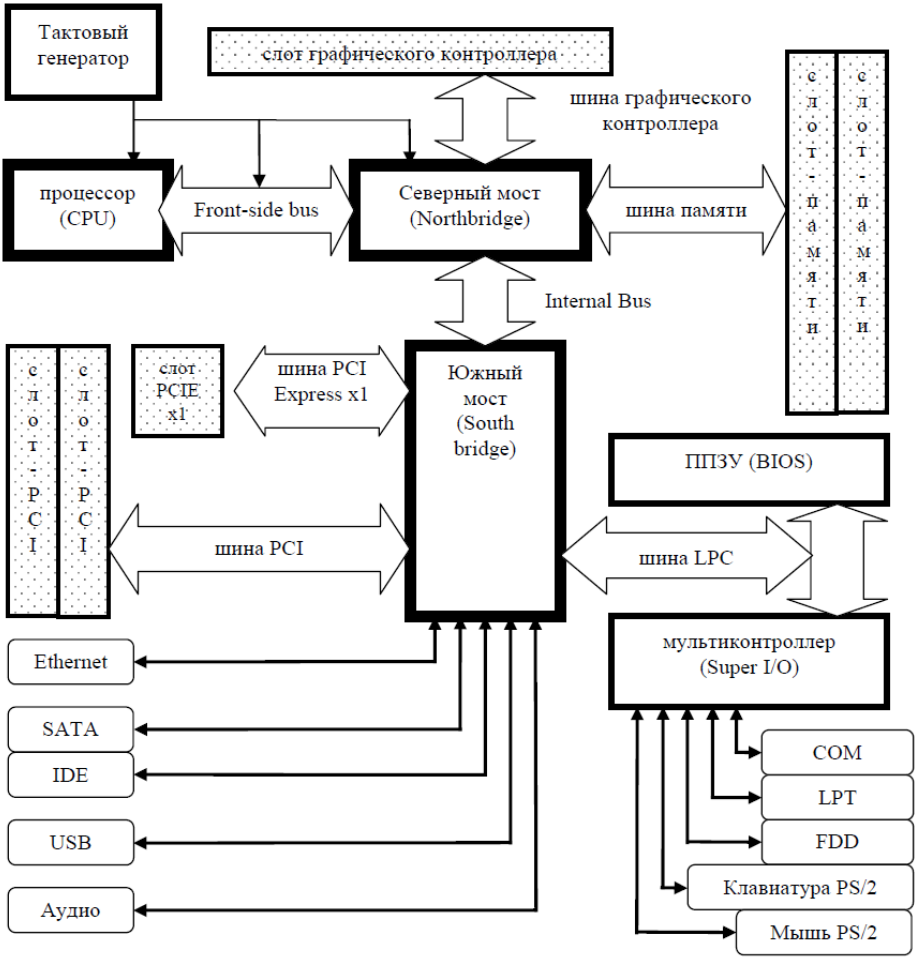

Рис. 5.3. Магистрали в персональном компьютере

Интерфейсы накопителей и шины расширения часто объединяют в одну группу под названием **“внутренние интерфейсы”.**

Через **кабельные интерфейсы** подключаются **внешние**, относительно компьютера, устройства, в том числе другие компьютеры. Универсальные устройства ввода-вывода преобразуют сигналы компьютера в сигналы кабельных интерфейсов. ***USB*** (универсальная последовательная шина) – самый распространённый кабельный интерфейс.

Беспроводные интерфейсы, как видно из названия, используются для беспроводного подключения к компьютеру различных устройств. В состав ЭВМ, в основном в ноутбуки и планшеты, входят приёмопередатчики с протоколами ***Bluetooth, Wi-Fi*** и *NFC*.

Видеоинтерфейсы связывают видеоадаптеры компьютеров с дисплеями.

5.2. Шины расширения

Шины расширения (Expansion Bus) предназначены для подключения различных адаптеров ПУ. Интерфейсы шин расширения PC ведут историю с 8-битной шины *ISA.* Ее открытость обеспечила появление широкого спектра плат расширений, позволивших использовать PC в различных сферах. С появлением AT-286 шина была расширена по разрядности и возможностям. Шина *EISA* была призвана сделать обмен еще более производительным и надежным. Она содержала прогрессивные идеи автоматизации конфигурирования (прототип PnP), позволяя устанавливать и ISA-адаптеры. Шина *MCA,* предложенная IBM, не была поддержана производителями PC, так как ее спецификация была закрытой. С появлением процессора 486 родилась высокоскоростная локальная шина *VLB.* Однако она являлась дополнением к слоту шины *ISA/EISA* и использовалась в основном лишь для графических карт и дисковых контроллеров. Принципиальная привязка *VLB* к шине процессора 486 не обеспечила ей долголетия. Современная скоростная шина *PCI* является стандартной для компьютеров с процессорами x86 всех поколений старше четвертого, она используется в Power PC и других платформах. Развитием шины *PCI*, нацеленным на дальнейшее ускорение обмена, явился порт *AGP,* предназначенный для подключения графических адаптеров.

Для блокнотных компьютеров (ноутбуков), имевших закрытую архитектуру, потребность в подключении периферии привела к появлению стандартизованной шины *PCMCIA,* впоследствии переименованной в *PC Card.*

Шины расширения системного уровня позволяют адаптерам максимально использовать системные ресурсы PC: пространства памяти и ввода/вывода, прерывания, каналы прямого доступа к памяти. Как следствие, изготовителям модулей расширения приходится точно сле-

довать протоколам шины, выдерживая жесткие частотные и нагрузочные параметры, а также временные диаграммы. Отклонения приводят к несовместимости с некоторыми системными платами. Если при подключении к внешним интерфейсам это приведет к неработоспособности только самого устройства, то некорректное подключение к системной шине может блокировать работу всего компьютера. Следует также учитывать ограниченность ресурсов PC. Самые дефицитные из них – линии запросов прерываний (каналы прямого доступа можно заменить активным управлением шиной PCI). Проблемы распределения ресурсов на шинах решаются по-разному, но чаще всего применяется технология PnP.

В табл. 5.1 из [3] даны параметры стандартных шин расширения до появления PCI Express, в табл. 5.2 – скорость интерфейсов.

Таблица 5.1. **Параметры шин расширения до PCI Express**

Шина	Пропускная способность, Мбайт/с*	Разрядность данных	Разрядность адреса (адресуемое пространство памяти)	Частота, МГц
ISA-8	4	8	20 (1 Мбайт)	8
ISA-16	16	16	24 (16 Мбайт)	8
EISA	33,3	32	32 (4 Гбайт)	8, 33
MCA-16	16	16	24 (16 Мбайт)	10
MCA-32	20	32	32 (4 Гбайт)	10
VLB	132	32/64 32 (4 Гбайт)	33-50(66)	
PCI	132/533	32/64	32/64 (4 Гбайт)	33(66)
AGP 1x/ 2x/4x/8x	266/ 533/ 1066/ 2132	32	32/64	66
PCI-x	533/ 4256	64	32/64	66-533
LPC	6, 7	8/ 16/ 32	32	33
PCMCIA (PC Card)	10/20	8/16	26 (64 Мбайт)	10
Card Bus	132	32	32	33

* Указана максимальная пропускная способность. Реальная примерно в 2 раза ниже за счет прерываний, регенерации и протокольных процедур.
** Поддержка автоматического конфигурирования. Для ISA PnP является позднейшей надстройкой, реализуемой адаптерами и ПО.

Таблица. 5.2. **Компьютерные интерфейсы (шины расширения)**

ISA 8 разрядов/4,77 МГц	9,6 Мбит/с	1,2 МБ/с
ISA 16 разрядов/8,33 МГц	42,4 Мбит/с	5,3 МБ/с
Low Pin Count	133,33 Мбит/с	16,67 МБ/с
PCI 1.0	640 Мбит/с	80 МБ/с
PCI 2.0	1 Гбит/с	133 МБ/с
PCI 64	2 Гбит/с	266 МБайт/с
PCI 66	4 Гбит/с	533 МБ/с
AGP 1.0 1x	2 Гбит/с	266 МБайт/с
AGP 1.0 2x	4 Гбит/с	533 МБ/с
AGP 2.0 4x	8 Гбит/с	1 ГБайт/с
AGP 3.0 8x	16 Гбит/с	2 ГБ/с
PCI Express 1.0 x1(одна линия и дуплекс)	4 Гбит/с	0,5 ГБ/с
PCI Express 2.0 x1	8 Гбит/с	1 ГБайт/с
PCI Express 3.0 x1	16 Гбит/с	2 ГБ/с
PCI Express 4.0 x1	32 Гбит/с	4 ГБ/с
PCI Express 2.0 x16 (16 линий, дуплекс)	128 Гбит/с	16 ГБайт/с
PCI Express 3.0 x16 (16 линий, дуплекс)	256 Гбит/с	32 ГБайт/с

Шины расширения конструктивно оформляются в виде щелевых разъемов (слотов) на системной плате для установки плат адаптеров. Унификация системных плат, корпусов и плат расширения обеспечивается:

- стандартизацией размеров, количества контактов и электрического интерфейса слотов шин расширения;
- фиксированным расстоянием от слота до задней кромки платы;
- фиксированным шагом между соседними слотами, а также их привязкой к крепежным точкам и разъему клавиатуры;
- определением максимальных габаритов (длины и высоты) карт расширения;
- определением геометрии нижнего края платы расширения, формы и размера фиксирующей скобки.

5.3. Кабельные интерфейсы

5.3.1. Параллельный интерфейс – LPT - порт

Порт параллельного интерфейса был введен в PC для подключения принтера, отсюда и пошло его название *LPT-порт* (Line PrinTer – построчный принтер). Традиционный, он же стандартный LPT-порт (так

называемый *SPP-порт),* ориентирован на вывод данных, хотя с некоторыми ограничениями позволяет и вводить данные. Существуют различные модификации LPT-порта – двунаправленный, EPP, ECP и другие, расширяющие его функциональные возможности, повышающие производительность и снижающие нагрузку на процессор. Поначалу они являлись фирменными решениями отдельных производителей, позднее был принят стандарт IEEE 1284.

С внешней стороны порт имеет 8-битную шину данных, 5-битную шину сигналов состояния и 4-битную шину управляющих сигналов, выведенные на разъем – розетку DB-25S. В LPT-порту используются логические уровни ТТЛ, что ограничивает допустимую длину кабеля из-за невысокой помехозащищенности ТТЛ-интерфейса. Гальваническая развязка отсутствует – схемная земля подключаемого устройства соединяется со схемной землей компьютера. Из-за этого порт является уязвимым местом компьютера, страдающим при нарушении правил подключения и заземления устройств. Поскольку порт обычно располагается на системной плате, в случае его «выжигания» зачастую выходит из строя и его ближайшее окружение, вплоть до выгорания всей системной платы.

С программной стороны LPT-порт представляет собой набор регистров, расположенных в пространстве ввода-вывода. Регистры порта адресуются относительно базового адреса порта, стандартными значениями которого являются 3BCh, 378h и 278h. Порт может использовать линию запроса аппаратного прерывания, обычно IRQ7 или IRQ5. В расширенных режимах может использоваться и канал DMA.

5.3.2. Последовательные интерфейсы UART

Существует несколько широко используемых стандартов последовательного асинхронного обмена информацией с протоколом **UART.** На рис. 5.4 показаны схемы подключения и характеристики этих интерфейсов.

Самая низкая помехозащищённость у ***RS-232***, так как по общему проводу может течь не только ток приёмника RS-232. Токи сторонних устройств за счёт падения напряжения на сопротивлении общего провода создают напряжение помехи. Скорость передачи 20 кбит/с. Длина кабеля 15 метров. Подключение линии в ***RS-423*** к дифференциальному входу приёмника с помощью дополнительного земляного провода ис-

ключает такой вид помехи и увеличивает допустимую длину соединения. Скорость передачи зависит от расстояния: 100 кбит/с (9 м), 10 кбит/с (90 м), 1 кбит/с (1200 м).

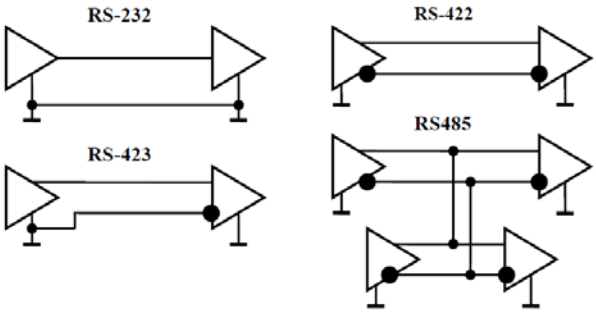

Рис. 5.4. Стандарты последовательных интерфейсов

Линии связи между передатчиком и приёмником делятся на симметричные и несимметричные линии. При синхронном подключении по одному проводу передаётся прямой сигнал, по другому – инвертированный (вместо логической единицы передаётся ноль и наоборот). Электрическая помеха действует одинаково на оба провода линии, добавляя синфазное паразитное напряжение на дифференциальном входе приёмника. Приёмник определяет разность входных напряжений (рис. 5.5). Полезный сигнал удваивается, сигналы помехи вычитаются друг из друга.

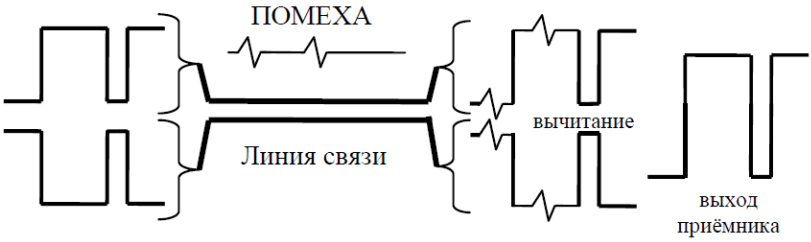

Рис. 5.5. Электрическая помеха в симметричной линии связи

Симметричные линии используют интерфейсы ***RS-422*** и ***RS-485.*** Линия связи у ***RS-422*** типа «точка-точка», у ***RS-485*** – типа «магистраль». Физическая среда передачи этих интерфейсов – витая пара. Перекрещивание проводов позволяет избавиться от магнитной составляющей внешней электромагнитной помехи. Скорости передачи – 10 Мбит/с (12 м), 1 Мбит/с (120 м), 100 кбит/с (1200 м).

5.3.3. USB

USB (Universal Serial Bus – универсальная последовательная шина) является промышленным стандартом расширения архитектуры PC, ориентированным на интеграцию с телефонией и устройствами бытовой электроники. Версия 1.0 была опубликована в начале 1996 года, большинство устройств поддерживает версию 1.1, которая вышла осенью 1998 года, – в ней были устранены обнаруженные проблемы первой редакции. Весной 2000 года опубликована спецификация USB 2.0, в которой предусмотрено 40-кратное повышение пропускной способности шины. Первоначально (в версиях 1.0 и 1.1) шина обеспечивала две скорости передачи информации: *полная скорость FS* (full speed) – 12 Мбит/с и *низкая скорость LS* (Low Speed) – 1,5 Мбит/с. В версии 2.0 определена еще и *высокая скорость HS* (High Speed) – 480 Мбит/с, которая позволяет существенно расширить круг устройств, подключаемых к шине. В одной и той же системе могут присутствовать и одновременно работать устройства со всеми тремя скоростями. Шина с использованием промежуточных хабов позволяет соединять устройства, удаленные от компьютера на расстояние до 25 м. Версии USB 3.0 и USB 3.1 работают в десять раз быстрее.

5.3.4. Шина IEEE 1394 - FireWire

Стандарт для высокопроизводительной последовательной шины (High Performance Serial Bus), получивший официальное название ШЕЕ 1394, был принят в 1995 году.

Основные свойства шины FireWire перечислены ниже.

- *Многофункциональность.* Шина обеспечивает цифровую связь до 63 устройств без применения дополнительной аппаратуры (хабов). Устройства бытовой электроники – цифровые камкордеры (записывающие видеокамеры), камеры для видеоконференций, фотокамеры, приемники кабельного и спутникового телевидения,

цифровые видеоплейеры (CD и DVD), акустические системы, цифровые музыкальные инструменты, а также периферийные устройства компьютеров (принтеры, сканеры, устройства дисковой памяти) и сами компьютеры могут объединяться в единую сеть.

- *Высокая скорость обмена и изохронные передачи.* Шина позволяет даже на начальном уровне (S100) передавать одновременно два канала видео (30 кадров в секунду) широковещательного качества и стереоаудиосигнал с качеством CD.
- *Низкая цена компонентов и кабеля.*
- *Легкость установки и использования.* FireWire расширяет технологию PnP. Система допускает динамическое (горячее) подключение и отключение устройств. Устройства автоматически распознаются и конфигурируются при включении/ отключении. Питание от шины (ток до 1,5 А) позволяет подключенным устройствам общаться с системой даже при отключении их питания. Управлять шиной и другими устройствами могут не только PC, но и другие «интеллектуальные» устройства бытовой электроники.

Fire Wire по инициативе VESA позиционируется как шина «домашней сети», объединяющей всю бытовую и компьютерную технику в единый комплекс. Эта сеть является одноранговой (peer-to-peer), чем существенно отличается от USB.

Сравнение шин USB и FireWire приведено в табл. 5.3.

Таблица 5.3. **Свойства USB и FireWire**

	USB	**FireWire (**IEEE 1394**)**
Скорость передачи до Мбит/с	480 (USB 2.0), 4800 (USB 3.0), 10000 (USB 3.1 G2)	98,304(S100); 196,608(S200); 393,216(S400); (S800)
Мощность шины питания	0,5 А (USB 2.0), 0,9А (USB 3.0), 100Вт (USB 3.1 G2)	1,5 А
Сеть	Ведущий – ведомый	Одноранговая (peer-to-peer) (равный-равному)
Адресное пространство	**127**	**63**

5.3.5. Ethernet

Ethernet в основном описывается стандартами IEEE группы 802.3. Стандарты Ethernet на физическом уровне модели OSI (*open systems*

interconnection basic reference model – базовая эталонная модель взаимодействия открытых систем) определяют проводные соединения и электрические сигналы, на канальном уровне – формат кадров и протоколы управления доступом к среде.

В настоящее время Ethernet в качестве передающей среды использует витую пару или оптический кабель. Оба типа среды обеспечивают гальваническую развязку подключаемых устройств. В табл. 5.4 приведены скорости разных стандартов Ethernet.

Таблица 5.4. **Скорости Ethernet**

Название	Скорость
Ethernet	10 Мбит/с
Fast Ethernet	100 Мбит/с
Gigabit Ethernet	1000 Мбит/с
10G Ethernet	10 Гбит/с

5.4. Интерфейсы накопителей

Таблица 5.5. **Интерфейсы для подключения накопителей (внутренние)**

Название	Тип	Скорость
Интерфейс НГМД ПК		0,062 МБайт/с
ATA-1 (DMA-0)	параллельный	4,2 МБайт/с
ATA-2 (DMA-1)	параллельный	13,3 МБайт/с
ATA-4 (UDMA-1)	параллельный	25,0 МБайт/с
ATA-4 (UDMA-2)	параллельный	33,3 МБайт/с
ATA-5 (UDMA-3)	параллельный	44,4 МБайт/с
ATA-6 (UDMA-5)	параллельный	100 МБайт/с
ATA-7 (UDMA-6)	параллельный	133 МБайт/с
SATA 1.x 1.5Gb/s	параллельный	150 МБайт/с
SATA 2.x 3Gb/s	параллельный	300 МБайт/с
SATA 3.x 6Gb/s	параллельный	600 МБайт/с

5.5. Видеоинтерфейсы

Таблица 5.6. **Интерфейсы дисплеев**

VGA - разъём	Аналоговый видеосигнал в стандарте VESA	400 МГц (−3 dB)
DVI-A	только аналоговая передача	400 МГц (−3 dB)
DVI-D	только цифровая передача	тактовая частота: от 21,96 МГц до 165 МГц
DVI-I	аналоговая и цифровая передача	
HDMI	цифровая передача видео и аудио	(4,9)-10,2 Гбит/с, кабель до (10)-3 метров
HDMI Type B	цифровая передача видео и аудио	2x10,2 Гбит/с, кабель до 3 метров
HDMI 2.0	цифровая передача видео и аудио	18 Гбит/с, кабель до 3 метров
DisplayPort	цифровая передача видео и аудио	10,2 Гбит/с, кабель до 15 метров
DisplayPort 1.2	цифровая передача видео и аудио	21,6 Гбит/с, кабель до 15 метров
Thunderbolt	объединяет PCI Express (PCIe) и DisplayPort (DP) и шину питания DC	20 Гбит/с, 3 м максимум, 10 Ватт питание

Ниже приведена созданная журналом CHIP хронология интерфейсов.

Внешние интерфейсы

1995 год: FireWire 400 **50 Мбайт/с**
1996 год: USB 1.0 **1,5 Мбайт/с**
2007 год: FireWire S3200 **400 Мбайт/с**
2013 год: USB 3.1 **1250 Мбайт/с**
2013 год: Thunderbolt 2 **2500 Мбайт/с**

Внутренние интерфейсы

1989 год: ATA-1 **8,3 Мбайт/с**
2000 год: Serial ATA 1.5 **150 Мбайт/с**
2001 год: ATA/ATAPI-7 **133 Мбайт/с**
2003 год: PCI Express 1.0 **250 Мбайт/с**
2009 год: Serial ATA 6.0 **600 Мбайт/с**
2016 год: PCI Express 4.0 **4000 Мбайт/с**

Беспроводные интерфейсы

1994 год: инфракрасный порт (IrDA 1.0) **0,014 Мбайт/с**
1997 год: WLAN (IEEE 802.11) **0,25 Мбайт/с**
1999 год: Bluetooth 1.0 **0,09 Мбайт/с**
2006 год: NFC **0,05 Мбайт/с**
2009 год: инфракрасный порт (Giga-IR) **125 Мбайт/с**
2010 год: Bluetooth 4.0 **3 Мбайт/с**
2013 год: WLAN (IEEE 802.11ac) **867 Мбайт/с**

Видеоинтерфейсы

1999: DVI **930 Мбайт/с**
2003: HDMI **495 Мбайт/с**
2009: DisplayPort 1.1 **1080 Мбайт/с**
2014: HDMI 2.0 **1080 Мбайт/с**
2014: DisplayPort 1.3 **4050 Мбайт/с**

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Язык компьютера / под ред. и с предисл. В.М.Курочкина; пер. с англ. М.: Мир, 1989. 240с.
2. Толковый словарь по вычислительным системам / под ред. В. Иллингуорта [и др.]; пер. с англ. А.К. Белоцкого [и др.] под ред. Е.К. Масловского. М.: Машиностроение, 1990. 560 с.
3. Гук М. Ю. Аппаратные интерфейсы ПК. Энциклопедия. СПб.: Питер, 2003. 528 с.
4. Гук М. Ю. Аппаратные средства IBM PC. Энциклопедия. 3-е изд. СПб.: Питер, 2008. 1072 с.
5. Гук М. Ю. Интерфейсы устройств хранения: ATA, SCSI и др. СПб.: Питер, 2007. 447 с.
6. Паттерсон Д., Хеннесси Дж. Архитектура компьютера и проектирование компьютерных систем. Классика computers Science. 4-е изд. СПб.: Питер, 2012. 784 с.
7. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ: учеб. пособие. М.: ИД «ФОРУМ»: ИНФРА-М, 209. 384 с.
8. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. М.: Стандартинформ, 2010.
9. Хоровиц П., Хилл У. Искусство схемотехники. М.: Мир, 1993.
10. Однокристальные микроЭВМ. М.: МИКАП, 1994. 400 с.
11. Однокристальные микроЭВМ: справочник / под ред. В.В. Карпова. М.: Бином, 1994.
12. Келим Ю.М. Вычислительная техника: учеб. пособие / Ю.М. Келим. 4-е изд., перераб. и доп. М.: Академия, 2008.
13. Магда Ю. Ассемблер для процессоров Intel Pentium. СПб.: Питер, 2006.
14. Юров В. Assembler: учебник для вузов. 2-е изд. СПб.: Питер, 2008.
15. Гуров. Микропроцессорные системы. учеб. пособие. М.: Инфра-м, 2016. 336 с.

16. Гусев В.Г. Электроника и микропроцессорная техника: учебник для вузов. М.: Кнопус, 2013. 800 с.

17. Макуха В.В., Микерин В.А. Микропроцессорные системы и персональные компьютеры: учеб. пособие для вузов. М.: Юрайт, 2017. 175 с.

18. Микушин А.В. Цифровые устройства и микропроцессоры: учеб. пособие. СПб.: БХВ-Петербург, 2010. 832 с.

19. Романов Ф.И., Шахнов В.А. Конструкционные системы микро- и персональных ЭВМ. М.: Высшая школа, 1991. 271 с.

20. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. М.: Энергоатомиздат, 1990. 224 с.

21. Титов М.А. Изделия электронной техники. Микропроцессоры и однокристальные микроЭВМ. М.: Радио и связь, 1994.

22. Разработка устройств сопряжения для персонального компьютера типа IBM PC: практич. пособие / Ю.В. Новиков [и др.]. М.: ЭКОМ., 1997. 224 с.

Учебное издание

Иванов Владимир Васильевич

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Учебное пособие

Редактор Н. С. К у п р и я н о в а
Компьютерная вёрстка А. В. Я р о с л а в ц е в о й

Подписано в печать 19.08.2019. Формат 60×84 1/16.
Бумага офсетная. Печ. л. 5,0.
Тираж 25 экз. Заказ . Арт. – 7(Р2У)/2019.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, Самара, Московское шоссе, 34.

Изд-во Самарского университета.
443086, Самара, Московское шоссе, 34.