Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Севастопольский государственный университет»

В.С.ЧЕРНЕГА

конспект лекций по дисциплине

ТЕХНИЧЕСКИЕ СРЕДСТВА ИНФОРМАЦИОННЫХ СИСТЕМ

для студентов очной и заочной форм обучения

по направлению

09.03.02 “Информационные системы и технологии”

Севастополь 2019

Технические средства информационных систем. Конспект лекций по дисциплине "Технические средства информационных систем" для студентов дневной и заочной формы обучения / Сост. В.С.Чернега. — Севастополь: Изд. СевГУ, 2019. — 199 с.

В конспекте приведена общая характеристика информационных систем и технических средств, рассмотрены технические средства преобразования, ввода и вывода информации, а также средства проектирования и отладки программного обеспечения микропроцессорных информационных систем.

Рецензент: канд. техн. наук, доц. Бондарев В.Н.

Содержание

Введение …….. 5

1. Общая характеристика информационных систем и технических средств …… 6

1.1. Классификация информационных систем и виды ИС ……. 6

1.2. Технические средства преобразования аналоговых сигналов в цифровые …… 10

1.3. Цифро-аналоговые преобразователи 17

2. Технические средства обработки информации …… 22

2.1. Обобщенная структурная схема компьютера …… 22

2.2. Неймановская и гарвардская архитектура компьютеров ……. 24

2.3. Архитектура однокристального микропроцессора …….. 25

2.4. Структурная схема 8-разрядного однокристального МП …… 28

2.5. Функционирование и временные диаграммы МП ……... 30

2.6. Программируемый контролер прерываний …….. 34

2.7. Программируемый контроллер прямого доступа к памяти … 38

2.8. Подключение клавиатуры и устройств индикации к микро-ЭВМ ……. 41

3. **Запоминающие устройства информационных систем** …….. 46

3.1. Иерархическая организация памяти компьютера …… 46

3.2. Основные характеристики полупроводниковых ЗУ …… 47

3.3. Статические ОЗУ с произвольным доступом …….. 48

3.4. ОЗУ динамического типа …… 50

3.5. Постоянные запоминающие устройства ..……. 51

3.6. Кэш-память и ее организация ……. 53

4. **Архитектура 16-разрядных процессоров** ……. 56

4.1. Линейная и сегментная адресация ……. 56

4.2. Сегментация памяти и вычисление адресов ……. 57

4.3 Архитектура 16-разрядного процессора первого поколения 59

4.4. Регистры процессора Intel 8086 ……... 62

4.5. Система команд 16-разрядного процессора первого поколения ……. 65

4.6. Способы адресации памяти и устройств ввода/вывода …… 72

4.7. Структура и функционирование 16-разрядной микро-ЭВМ … 76

4.8. Защита памяти в процессорах второго и последующих поколений ……. 78

4.9. Поддержка многозадачности и виртуальной памяти …… 80

4.10 Архитектура процессоров второго поколения ……. 83

4.11. Программирование 16-разрядных микропроцессоров на Ассемблере …… 88

5. Архитектура 32-разрядных процессоров …….. 96

5.1. Структурная схема и регистры процессоров …… 96
5.2. Страничная организация памяти …… 99
5.3. Суперскалярные и мультискалярные микропроцессоры ……. 104
5.4. Архитектура суперскалярных процессоров типа Pentium ….. 111
6. Архитектура 64-разрядных и многоядерных процессоров 116
6.1. Общая характеристика 64-разрядных процессоров ……. 116
6.2. Процессор Athlon 64 ……..... 117
6.3. Многоядерные процессоры AMD и Intel …… 119

7. Архитектура персональных компьютеров …… 126
7.1. Обобщенная структурная схема компьютера …… 126
7.2. Распределение адресного пространства ПЭВМ ……. 133
7.4. Особенности архитектуры серверных компьютеров …… 139
7.5. Клавиатура компьютера и ее взаимодействие с процессором 142
7.6. Ввод информации с клавиатуры в приложениях Windows …….. 145
7.7. Видеосистема компьютера ……... 146
7.8. Аудиосистема компьютера …….. 153
7.9. Последовательный интерфейс персонального компьютера …. 157
7.10. Универсальный интерфейс USB ……. 159
7.11. Мультимедийный интерфейс высокого разрешения …… 163
7.12. Источники питания компьютеров …….. 165

8. Технические средства ввода информации в компьютер .. 169
8.1. Преобразователи изображений в электрический сигнал …….. 170
8.2. Сканеры …… 174
8.3. Сенсорные экраны …… 177
8.4. Оптическая мышь …… 186

9. Технические средства вывода информации …… 189
9.1. Струйные принтеры …… 189
9.2. Лазерные принтеры …… 193

Список рекомендованной литературы …….. 198

Введение

Реализация технологического процесса материального производства осуществляется с помощью различных технических (технических) средств, к которым относятся: оборудование, станки, инструменты, конвейерные линии и т.п.

По аналогии и для информационных систем и технологий существует нечто подобное. Такими инструментальными средствами создания и обработки информации являются аппаратное, программное и математическое обеспечение этих процессов. С их помощью производится переработка первичной информации в информацию нового качества, выполняется разработка новых информационных технологий, осуществляется поддержка технических средств и информационного технологического процесса. В связи с этим инструментарий технологического процесса ввода, обработки, хранения и выдачи информации подразделяется на аппаратный и программный.

Учебным планом подготовки бакалавров по направлению обучения «Информационные системы и технологии» дисциплинам, посвященным созданию математического обеспечения и программных средств предусмотрено существенно большая часть времени, а аппаратному инструментарию уделено недостаточно внимания. В связи с этим в данной дисциплине рассматриваются преимущественно аппаратные технические средства, обеспечивающие реализацию информационных систем и технологий.

1. Общая характеристика информационных систем и технических средств

1.1. Классификация информационных систем и виды технических средств

Информационная система (ИС) — по законодательству РФ — совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств. В более расширенном понимании **информационная система —** это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи информации и принятия управленческих решений. Обобщенная структура информационной системы, как совокупность обеспечивающих подсистем, изображена на рис. 1.1.

Рис. 1.1- Структура информационной системы как совокупность обеспечивающих подсистем

Информационное обеспечение - совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации или предприятии, а также методология построения баз данных. Назначение подсистемы информационного обеспечения состоит в современном формировании и выдаче достоверной информации для принятия управленческих решений.

Техническое обеспечение — комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы. В комплекс технических средств входят:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;

- эксплуатационные материалы и др.

Математическое и программное обеспечение — совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств. К средствам *математического обеспечения* относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.
- В состав *программного обеспечения* входят общесистемные и специальные программные продукты, а также *техническая документация*.

Организационное обеспечение - совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;
- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Правовое обеспечение - совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации. Главной целью правового обеспечения является укрепление законности.

В информационной системе происходят следующие процессы:

- ввод информации из внешних и внутренних источников;
- обработка входящей информации;
- хранение информации для последующего ее использования;
- вывод информации в удобном для пользователя виде;
- выработка управляющих воздействий, т.е. представление информации, переработанной в данной организации, для корректировки входящей информации.

Информационная система предназначена для своевременного обеспечения определенных людей необходимой информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области.

Информационные системы классифицируют по различным признакам (рис. 1.2).

Рисунок 1.2 − Классификация информационных систем

Информационные системы **организационного управления** предназначены для автоматизации функций управленческого персонала. К этому классу относятся информационные системы управления как промышленными фирмами, так и непромышленными объектами: гостиницами, банками, торговыми фирмами и др. Основными функциями подобных систем являются: оперативный контроль и регулирование, оперативный учет и анализ, перспективное и оперативное планирование, бухгалтерский учет, управление сбытом и снабжением и другие экономические и организационные задачи.

ИС управления технологическими процессами (ТП) служат для автоматизации функций производственного персонала. Они широко используются при организации для поддержания технологического процесса в металлургической и машиностроительной промышленности.

ИС автоматизированного проектирования (САПР) предназначены для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии. Основными функциями подобных систем являются: инженерные расчеты, создание графической документации (чертежей, схем, планов), создание проектной документации, моделирование проектируемых объектов.

Интегрированные (корпоративные) ИС используются для автоматизации всех функций организации (предприятия) и охватывают весь цикл работ от проектирования до сбыта продукции. Создание таких систем весьма затруднительно, поскольку требует системного подхода с позиций главной цели, например получения прибыли, завоевания рынка сбыта и т.д. Такой подход может

привести к существенным изменениям в самой структуре фирмы, на что может решиться не каждый управляющий.

Типовой архитектурой информационной системы является клиент-серверная, структура которой изображена на рисунке 1.3. Основными аппаратными компонентами такой системы являются серверный и клиентский компьютеры, устройства ввода и отображения информации.

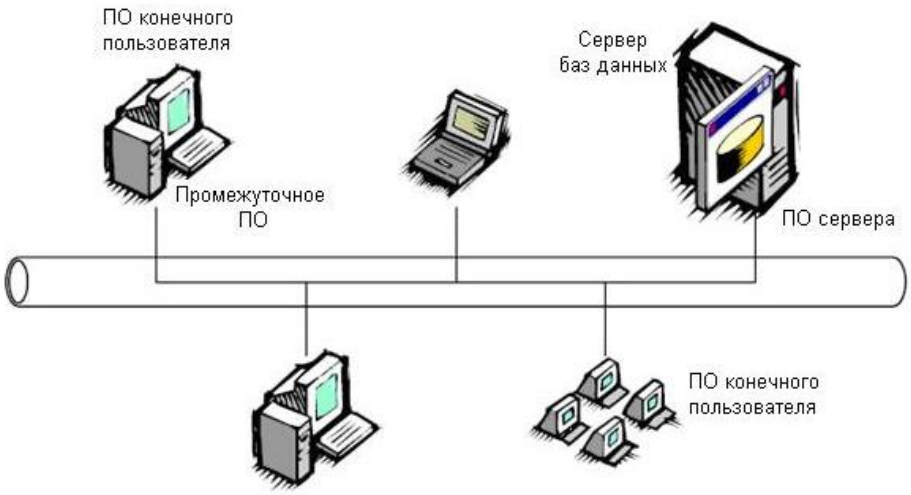


Рисунок 1.3 – Структура клиент-серверной ИС

К основным программным средствам относятся программное обеспечение сервера; программное обеспечение конечного пользователя; промежуточное программное обеспечение. Программное обеспечение сервера, кроме управления базами данных обеспечивает обслуживание клиентов. В таких СУБД предусмотрены механизмы блокировки и элементы управления многопользовательским доступом, которые обеспечивают защиту данных от рисков, присущих параллельному доступу. Кроме этого, серверу баз данных приходится защищать данные от несанкционированного доступа, оптимизировать запросы к базе данных, обеспечивать целостность данных и контроль завершение транзакций.

К программному обеспечению конечного пользователя относятся средства разработки прикладных программ и генераторы отчетов, в том числе электронные таблицы и текстовые процессоры С помощью этого программного обеспечения пользователи устанавливают связь с сервером, формируют запросы, которые автоматически генерируются в запросы на языке SQL и отправляются на сервер. Сервер принимает и обрабатывает запросы, а затем передает полученные результаты клиентам. Промежуточное программное обеспечение — часть системы клиент-сервер, которая связывает программное обеспечение конечного пользователя с сервером.

1.2. Технические средства преобразования аналоговых процессов в цифровые

Обработка данных в информационных системах осуществляется обычно цифровыми устройствами, в то время как большинство природных процессов, отображающих информационные сообщения, являются непрерывными (аналоговыми). Поэтому перед вводом информации в компьютер осуществляется аналого-цифровое преобразование непрерывных процессов.

Аналого-цифровое преобразование (АЦП) заключается в преобразовании аналогового (непрерывного) процесса (напряжения или тока, сопротивления, емкости, температуры, давления и пр.) в последовательность двоичных чисел — цифровой код**. Цифро-аналоговое преобразование** (ЦАП) призвано выполнять обратную задачу, т.е. преобразовывать последовательность чисел, представленных в виде цифрового кода, в эквивалентный аналоговый сигнал. АЦП и ЦАП являются одним из основных аппаратных технических средств, используемых в информационных системах.

Наиболее часто входной величиной, подаваемой на АЦП, является напряжение. Все другие величины перед подачей на АЦП необходимо преобразовать в напряжение.

В общем случае напряжение характеризуется мгновенным значением $U(t)$ или средним за выбранным промежуток времени T значением:

$$U_{ср} = U = \frac{1}{T}\int_0^T U(t)dt$$

В связи с этим все типы АЦП можно разделить на две группы: АЦП мгновенных значений напряжения и АЦП средних значений.

Выходной величиной АЦП является цифровой код, т.е. последовательность цифр, с помощью которой представляются дискретные квантовые величины. В АЦП используют четыре основных типа кодов: натуральный двоичный, десятичный, двоично-десятичный и код Грея.

В процессе преобразования напряжения в цифровой код осуществляются три независимые операции:

- дискретизация по времени;
- квантование по уровню;
- кодирование.

В процессе дискретизации непрерывный сигнал преобразуется в последовательность импульсов, амплитуда которых равна мгновенным значениям сигнала в моменты дискретизации, т.е. осуществляется амплитудно-импульсная модуляция (АИМ). Мгновенные значения сигналов называют отсчетами сигнала. АИМ в системах передачи данных получила название импульсно-кодовая модуляция (ИКМ).

Наиболее распространенной формой дискретизации является равномерная, в основе которой лежит теорема отсчетов (теорема Котельникова). Согласно этой теореме отсчеты сигнала $U(t_j)$ следует брать в дискретные моменты времени $t_j = j\Delta t$, а период дискретизации выбирать из условия:

$$\Delta t = 1/(2F_m),$$

где F_m - максимальная частота спектра преобразуемого сигнала.

Затем отсчеты квантуются по уровню, которым ставится в соответствие числовой эквивалент, представляющий собой двоичное отображение номера уровня квантования.

Дискретный сигнал можно представить в виде произведения исходного сигнала U(t) и дискретизирующей последовательности P(t)

$$U_\partial(t) = U(t)P(t) .$$

Дискретизирующая последовательность состоит из очень коротких импульсов. При теоретическом описании эта последовательность представляется δ – импульсами, которые следуют с частотой дискретизации $f_o = 1/T_o$

$$P(t) = \sum_{n=-\infty}^{\infty} \delta (t - nT_o)$$

Временные диаграммы процессов дискретизации, квантования и кодирования показаны на рисунке 1.4.

Основными параметрами и характеристиками АЦП являются:

Шаг квантования h **—** разность между двумя соседними значениями квантованной величины.

$$h = U_{вх.макс} / (2^N-1),$$

где $U_{вх.макс}$ **—** максимальное входное напряжение АЦП (напряжение полной шкалы), соответствующее максимальному значению выходного кода, N **—** разрядность АЦП. В стандартных АЦП максимальное входное напряжение обычно равно 5 В. Если напряжение источника сигналов превышает эту величину, то применяют входной делитель, выполненный на прецизионных резисторах.

Количество разрядов. Данное понятие применимо к аналого-цифровым преобразователям, вырабатывающим любые числовые коды. Для наиболее распространенных двоичных АЦП число разрядов равно двоичному логарифму максимального числа возможных кодовых комбинаций на выходе АЦП. Применительно к АЦП, вырабатывающим другие числовые коды, вводят число эквивалентных двоичных разрядов, соответствующее увеличенному до целого двоичному логарифму номинального числа возможных значений выходного кода.

Характеристика преобразования АЦП, представляющая собой зависимость между напряжением на его аналоговом входе и множеством возможных значений выходного кода. Характеристика может быть задана в виде таблицы, графика или формулы. Идеальная статическая характеристика преобразования 3-разрядного АЦП приведена на рисунке 1.5. Как видно из рисунка 1.5, в процессе преобразования возникает ошибка, не превышающая по величине половины величины шага квантования. Эта ошибка называется шумом квантования.

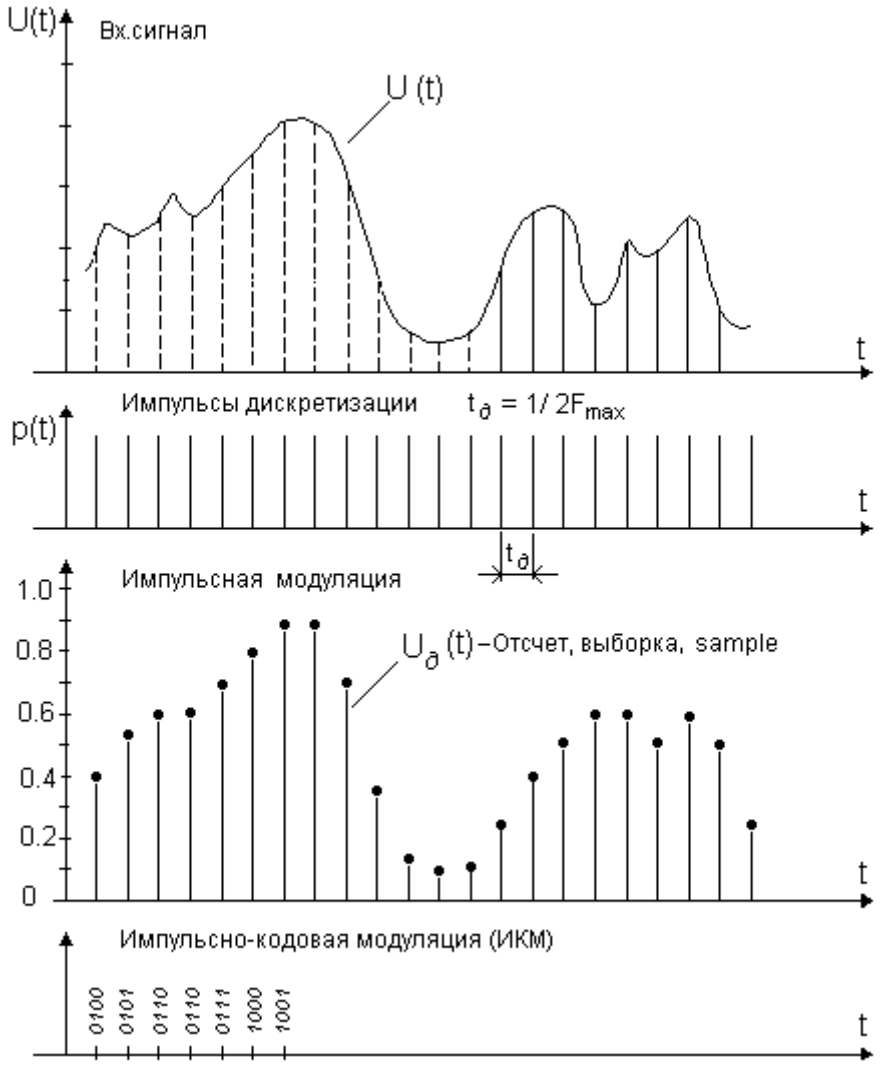


Рисунок 1.4 – Временные диаграммы импульсно-кодовой модуляции

Разрешающая способность — величина, обратная максимальному числу кодовых комбинаций на выходе АЦП. Чем больше разрядность преобразователя, тем выше его разрешающая способность. Разрешающая способность выражается в процентах, разрядах или децибелах и характеризует потенциальные возможности АЦП с точки зрения достижимой точности. Например, 12-разрядный АЦП имеет разрешающую способность 1/4096, или 0,0245% от полной шкалы, или минус 72,2 дБ.

Время преобразования $t_{пр}$ **—** это время, отсчитываемое от начала импульса дискретизации или начала преобразования до появления на выходе устойчивого кода, соответствующего данной выборке. Для одних типов АЦП, например, последовательного счета или многотактного интегрирования, эта величина является переменной, зависящей от значения входного сигнала, для других, таких как параллельные или последовательно-параллельные АЦП, а также АЦП последовательного приближения, примерно постоянной.

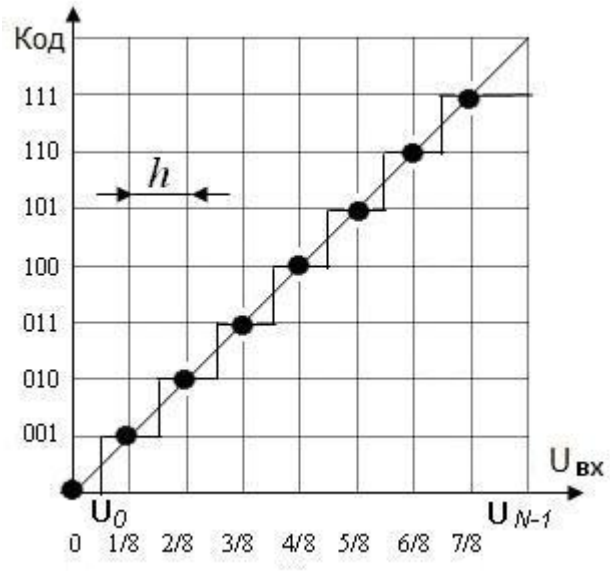


Рисунок 1.5 – Статическая характеристика преобразования АЦП

Наряду со временем преобразования используют и другую динамическую характеристику АЦП – ***максимальную частоту преобразования***. В общем случае *максимальной частотой преобразования* называют наибольшую частоту дискретизации входного сигнала, при которой выбранный параметр АЦП не выходит за заданные пределы.

Динамический диапазон АЦП (DR — *Dynamic Range*) называется отношение максимального воспринимаемого уровня входного напряжения к минимальному, выраженное в дБ:

$$DR = 20\lg(U_{max} / \Delta U) = 20\lg(2^N).$$

Например, для 12-разрядного АЦП DR=72 дБ.

Погрешность смещения нуля — значение $U_{вх}$, при котором выходной код АЦП равен нулю. Является аддитивной составляющей полной погрешности. Обычно определяется по формуле

$$\varepsilon_{см} = U_{вх.01} - h/2 ,$$

где $U_{вх.01}$ — значение входного напряжения, при котором происходит переход выходного кода из 0 в 1. Часто указывается в милливольтах или в процентах от полной шкалы:

$$\delta_{см} = (\varepsilon_{см} / U_{max})100\% .$$

Существуют различные методы аналого-цифрового преобразования, различающиеся между собой по точности и быстродействию. В большинстве случаев эти характеристики противоположны друг другу. В настоящее время большое распространение получили такие типы преобразователей как АЦП последова-

тельного счета, последовательного приближения, параллельные, параллельно-последовательные и с промежуточным преобразованием в интервал времени.

Структурная схема АЦП последовательного счета (рисунок 1.6) содержит компаратор, при помощи которого выполняется сравнение входного напряжения с напряжением обратной связи. На прямой вход компаратора поступает входной сигнал $U_{вх}$, а на инвертирующий — напряжение обратной связи. Работа преобразователя начинается с приходом импульса «ПУСК» от схемы управления, который замыкает ключ S. Через замкнутый ключ S импульсы U_1 от генератора тактовых импульсов поступают на счетчик, выходы которого соединены со входами цифро-аналогового преобразователя (ЦАП). В результате последовательного увеличения выходного кода счетчика N происходит последовательно-ступенчатое увеличение выходного напряжения U_5 ЦАП.

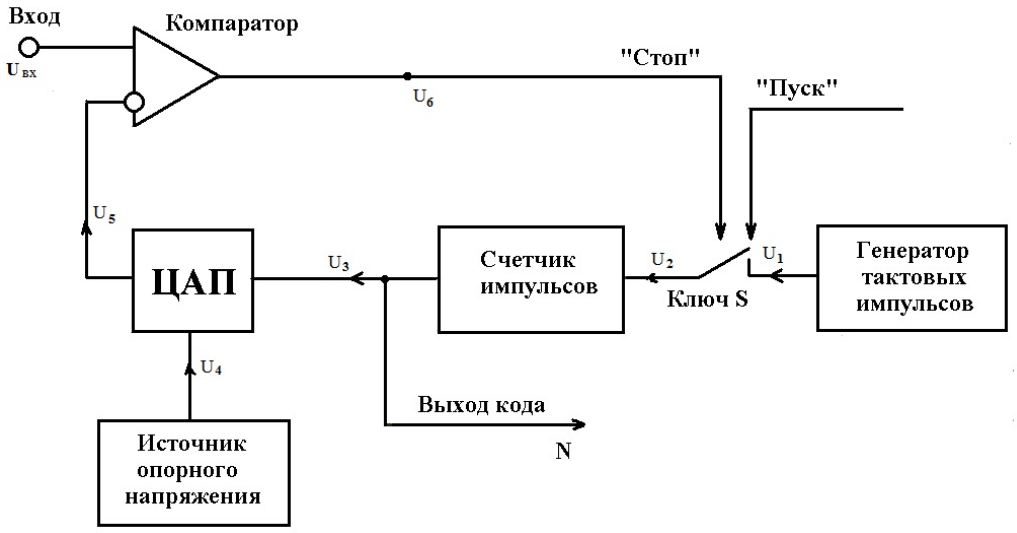


Рисунок 1.6 - Структурная схема АЦП последовательного счета

Когда выходное напряжение ЦАП сравняется со входным напряжением, произойдет переключение компаратора, и по его выходному сигналу «СТОП» разомкнется ключ S. В результате импульсы от генератора перестанут поступать на вход счетчика. Выходной код, соответствующий равенству $U_{вх} = U_s$, снимается с выходного регистра счетчика.

При числе двоичных разрядов счетчика, равном n и периоде следования счётных импульсов T максимальное время преобразования можно определить по формуле

$$T_{пр} = (2^n - 1)T.$$

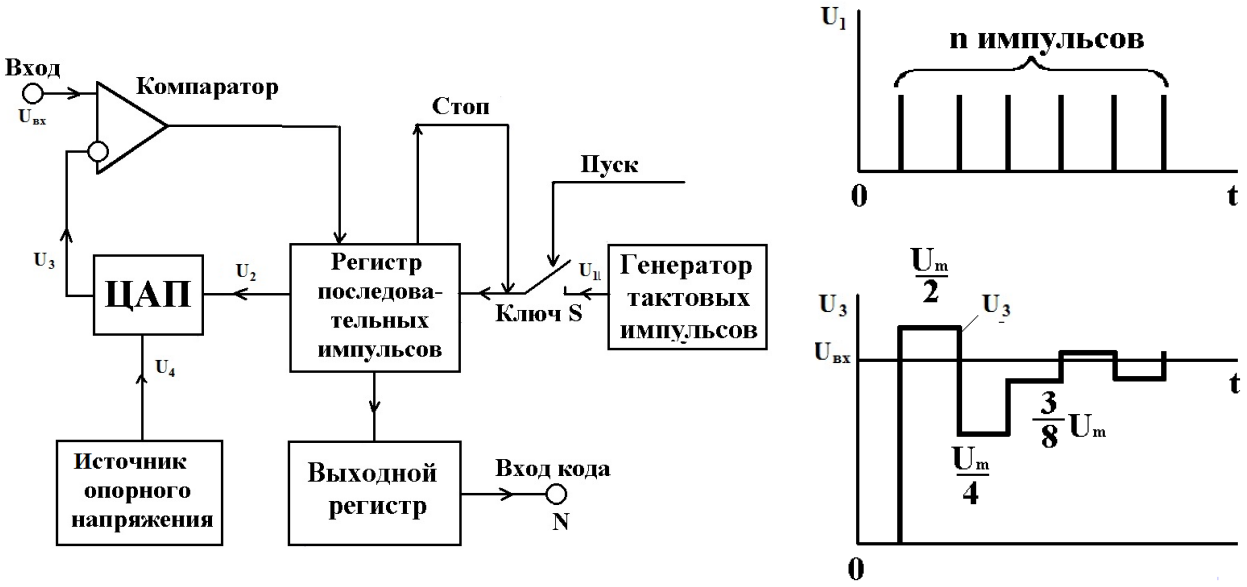
Уравнение преобразования АЦП последовательного счета можно записать в виде

$$Kh = U_{вх},$$

где $0 \le K \le n$ – число ступеней до момента сравнения;

h – шаг квантования.

Структурная схема АЦП последовательного приближения (рисунке 1.7*а*) отличается от структурной схемы последовательного счета тем, что вместо счетчика импульсов включен регистр последовательных приближений (РПП). В основе РПП лежит принцип дихотомии, т.е. последовательного сравнения преобразуемого напряжения $U_{вх}$ с соответствующими долями возможного максимального его значения U_m: ½, ¼, 1/8 и т.д. Это позволяет для *n*-разрядного АЦП выполнить весь процесс преобразования за *n* последовательных шагов (итераций) вместо (2^{n-1}) при использовании последовательного счета и получить существенный выигрыш по быстродействию.



а) *б*)

Рисунок 1.7 – Структурная схема АЦП последовательного приближения

АЦП функционирует следующим образом. На каждом шаге производится определение одного разряда, начиная со старшего. При первом сравнении определяется, больше или меньше напряжение $U_{вх}$ чем $U_m/2$. На следующем шаге определяется, в какой четверти диапазона находится $U_{вх}$. Каждый последующий шаг сужает область возможного результата. При каждом сравнении компаратор формирует импульсы, соответствующие состоянию «больше - меньше» (1 или 0), управляющие регистром последовательных приближений. График процесса последовательного приближения приведен на рисунке 1.7, *б.* Таким образом, АЦП последовательного приближения требуется один внутренний такт преобразования для каждого разряда, или *N* тактов для *N*-разрядного преобразования. Этот тип преобразователей применяется, когда необходимо разрешение 12, 14 или 16 разрядов и не требуется высокая скорость преобразования, а определяющими факторами являются невысокая цена и низкое энергопотребление. Такие АЦП чаще всего используется в разнообразных измерительных приборах и в системах сбора данных.

Структурная схема параллельного АЦП приведена на рисунке 1.8. В параллельном преобразователе используется массив компараторов, каждый из которых сравнивает входное напряжение с индивидуальным опорным напряжением. Такое опорное напряжение для каждого компаратора формируется на встроенном прецизионном резистивном делителе. Значения опорных напряжений начинаются со значения, равного половине младшего значащего разряда, и увеличиваются при переходе к каждому следующему компаратору с шагом, равным $U_{оп}/2^N$. В результате для 3-х разрядного АЦП требуется 2^3-1 или семь компараторов. А для 8-разрядного параллельного АЦП нужно уже 255 компараторов.

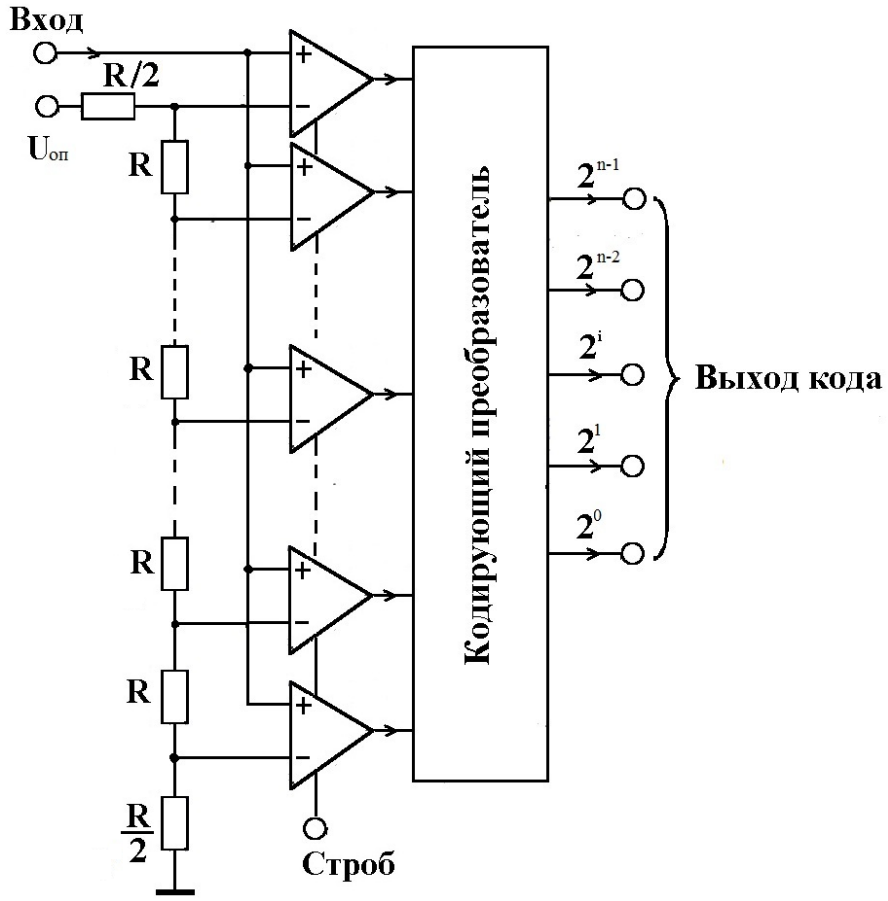


Рисунок 1.8 — Структурная схема параллельного АЦП

На выходах компараторов формируется квантованный сигнал, представленный в унитарном коде. Для преобразования унитарного кода в двоичный или двоично-десятичный используется кодирующий преобразователь. При работе в двоичном коде все резисторы имеют одинаковые сопротивления R. Время преобразования такого преобразователя составляет один такт, т.е. $T_{пр} = T$.

1.3. Цифро-аналоговые преобразователи

Цифро-аналоговый преобразователь (ЦАП) − это устройство для преобразования цифрового кода в аналоговый сигнал по величине, пропорциональной значению кода. ЦАП применяются для связи цифровых управляющих систем с устройствами, которые управляются уровнем аналогового сигнала. Также ЦАП является составной частью во многих структурах аналого-цифровых устройств и преобразователей.

Основной характеристикой ЦАП является **функция (характеристика) преобразования**. Она связывает изменение цифрового кода с изменением напряжения или тока. Функция преобразования ЦАП выражается следующим образом

$$U_{ВЫХ} = \frac{U_{MAX}}{N_{MAX}} N_{ВХ},$$

где $U_{вых}$ - значение выходного напряжения, соответствующее цифровому коду $N_{вх}$, подаваемому на входы ЦАП.

$U_{мах}$ - максимальное выходное напряжение, соответствующее подаче на входы максимального кода $N_{мах}$

Рисунок 1.9 – Функция преобразования цифро-аналогового преобразователя

Отношение $K_{цап} = U_{MAX} / N_{MAX}$ называют **коэффициентом цифро-аналогового преобразования**. Несмотря на ступенчатый вид характеристики, связанный с дискретным изменением входной величины (цифрового кода), считается, что ЦАП являются линейными преобразователями.

Если величину $N_{вх}$ представить через значения весов его разрядов, функцию преобразования можно выразить следующим образом

$$U_{вых} = K_{цап} \sum_{1}^{n} A_i U_i ,$$

где i - номер разряда входного кода $N_{вх}$; A_i - значение i-го разряда (ноль или единица); U_i – вес i-го разряда; n – количество разрядов входного кода (число разрядов ЦАП).

Разрешающая способность (вес разряда) определяется для конкретной разрядности, и вычисляется по следующей формуле

$$U_i = \frac{U_{ОП}}{2^n} ,$$

где $U_{ОП}$ - опорное напряжение ЦАП.

Абсолютная погрешность преобразования **—** максимальное отклонение выходного напряжения $U_{ВЫХ}$ в точке пересечения с идеальной характеристикой на уровне максимального напряжения U_{MAX} (напряжения полной шкалы) при максимальном значении входного кода N_{MAX}. Другими словами, это относительная разность между реальным и идеальным значениями предела шкалы преобразования при отсутствии смещения нуля.

Погрешность смещения нуля - значение $U_{ВЫХ}$, когда входной код ЦАП равен нулю.

Принцип работы большинства ЦАП состоит в суммировании долей аналоговых сигналов (веса разряда), в зависимости от входного кода.

ЦАП можно реализовать с помощью суммирования токов, суммирования напряжений и деления напряжений. В первом и втором случае в соответствии со значениями разрядов входного кода, суммируются сигналы генераторов токов и источников Э.Д.С. Последний способ представляет собой управляемый кодом делитель напряжения. Два последних способа не нашли широкого распространения в связи с практическими трудностями их реализации.

Способы реализации ЦАП с взвешенным суммированием токов.

Схема простейшего ЦАП с взвешенным суммированием токов изображена на рис.1.10.

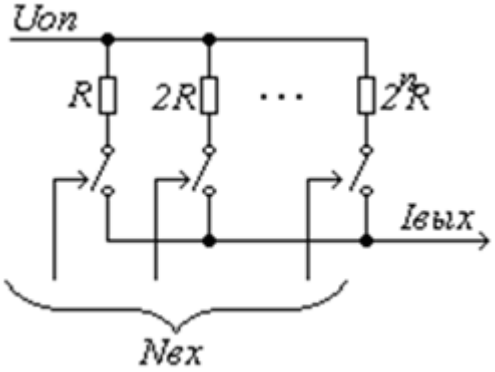


Рисунок 1.10 – Схема ЦАП с суммированием токов

Этот ЦАП состоит из набора резисторов и набора ключей. Число ключей и число резисторов равно количеству разрядов *n* входного кода. Номиналы резисторов выбираются в соответствии с двоичным законом. Если R=3 Ом, то 2R= 6 Ом , 4R=12 Ом, и так и далее, т.е. каждый последующий резистор больше предыдущего в 2 раза. При присоединении источника напряжения и замыкании ключей, через каждый резистор потечет ток. Значения токов по резисторам, благодаря соответствующему выбору их номиналов, тоже будут распределены по двоичному закону. При подаче входного кода $N_{вх}$ включение ключей производится в соответствии со значением соответствующих им разрядов входного кода. Ключ замыкается, если соответствующий ему разряд равен единице. При этом в узле суммируются токи, пропорциональные весам этих разрядов и величина вытекающего из узла тока в целом будет пропорциональна значению входного кода $N_{вх}$.

Сопротивление резисторов матрицы выбирают достаточно большое (десятки кОм). Поэтому для большинства практических случаев для нагрузки ЦАП играет роль источника тока. Если на выходе преобразователя необходимо получить напряжение, то на выходе такого ЦАП устанавливается преобразователь "ток-напряжение", например, на операционном усилителе (рис.1.11).

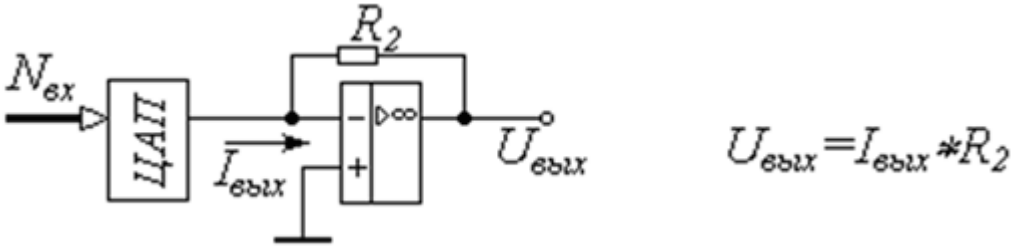


Рисунок 1.11 – ЦАП с преобразователем тока в напряжение

Однако при смене кода на входах ЦАП меняется величина тока, отбираемая от источника опорного напряжения. Это является главным недостатком такого способа построения ЦАП**.** Такой метод построения можно использовать только в том случае, если источник опорного напряжения будет с низким внутренним сопротивлением**.**

В другом случае в момент смены входного кода изменяется ток, отбираемый у источника, что приводит к изменению падения напряжения на его внутреннем сопротивлении и, в свою очередь, к дополнительному напрямую не связанному со сменой кода изменению выходного тока.

Сопротивления весовых резисторов могут отличаться в тысячи раз, что затрудняет реализацию таких резисторов в полупроводниковых ИС. Помимо этого, сопротивления резисторов старших разрядов могут быть соизмеримы с сопротивлением замкнутого ключа, а это приведет к погрешностям преобразования.

От всех указанных выше недостатков свободны структуры на основе резистивных *R-2R* матриц (рис.1.12). В матрице резисторов R-2R формируется ряд напряжений, отличающихся друг от друга ровно в два раза. Рассмотрим этот механизм. В конце резистивной цепочки находятся два резистора с сопротивлением

2R. Эти резисторы одним концом соединены друг с другом, другие концы присоединены к корпусу схемы, то есть резисторы соединены параллельно. В результате их общее сопротивление равно R. При соединении резистора R и параллельного соединения двух резисторов 2R образуется делитель напряжения с коэффициентом деления 2. В результате напряжение на его выходе будет в два раза меньше напряжения на его входе.

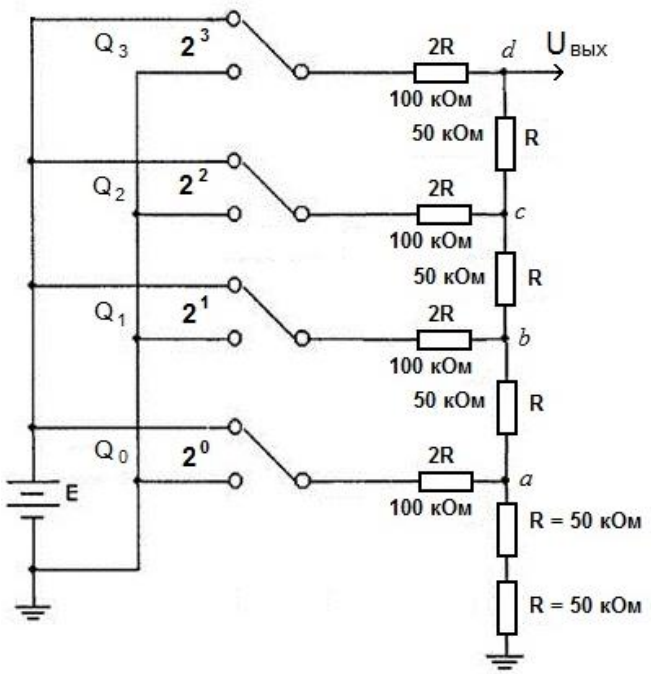


Рисунок 1.12 – Схема ЦАП с матрицей R-2R

Общее сопротивление делителя составляет 2R, так как сопротивления R в нем соединены последовательно. В результате в следующем звене матрицы ситуация повторяется. Снова образуется параллельное соединение двух резисторов 2R и снова образуется делитель напряжения в два раза. Так как напряжения в узлах матрицыR-2R отличаются друг от друга ровно в два раза, то и ток через резисторы 2R будет отличаться ровно в два раза, то есть подчиняться двоичному закону. Если теперь эти токи подавать или не подавать на вход аналогового сумматора на ОУ в зависимости от входного двоичного числа, то мы получим цифроаналоговый преобразователь.

Наличие только двух номиналов резисторов в матрице позволяет достаточно просто осуществлять подгонку их значений.

Другим вариантом использования резистивной матрицы R-2R является так называемый умножающий ЦАП (рис.1.13). Выходной ток для такой структуры пропорционален одновременно не только величине входного кода, но и величине опорного напряжения. Часто говорят, что он пропорционален произведению этих двух величин. Поэтому такие ЦАП называют умножающими. Такими свойствами будут обладать все **ЦАП,** в которых формирование взвешенных значений токов, соответствующих весам разрядов, производится с помощью резистивных матриц.

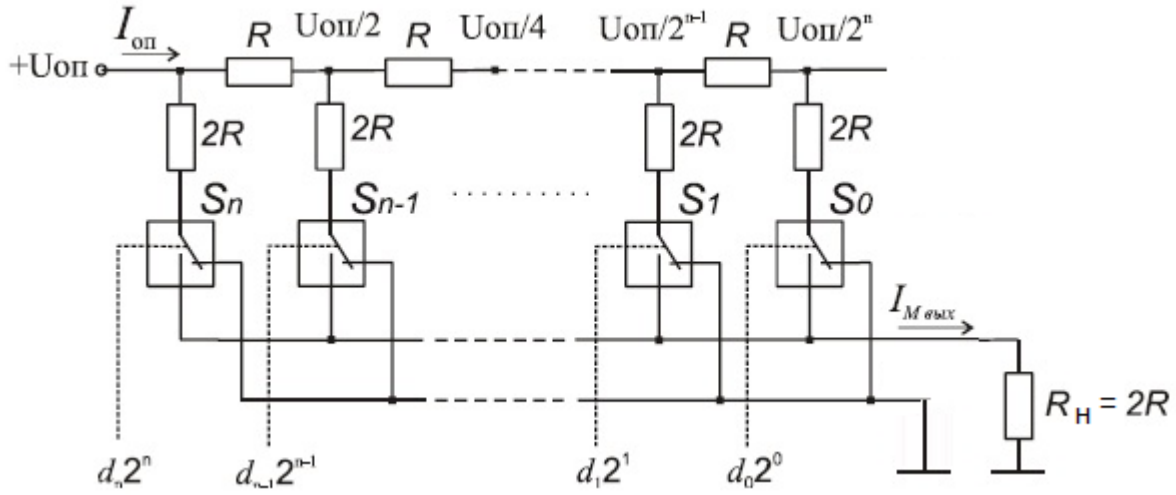


Рисунок 1.13- Умножающий R-2R цифро-аналоговый преобразователь

Цифро-аналоговым преобразователи на основе R-2R матрицы обладают рядом преимуществ, по сравнению с ЦАП с весовыми сопротивлениями. Первое заключается в том, что используются всего два номинала сопротивлений резисторов: R и 2R. Второе – нагрузка источника опорного напряжения ($R_{вх}$) не изменяется при любом числе последовательно включенных элементов матрицы и остается равной 2R. Третье – на выходе каждого элемента матрицы получается вдвое меньшее напряжение, чем на его входе.

Коммутация цепей резистивной матрицы в реальных преобразователях выполняется не механическими устройствами, а электронными ключами, выполненными по схемам КМОП, ТТЛ или ЭСТЛ. Токовые ключи должны иметь высокое быстродействие и не вносить заметных погрешностей в разрядные токи за счет собственного сопротивления в открытом состоянии. Для преобразователей среднего и низкого быстродействия широко применяются ключи на КМОП транзисторах, имеющих к тому же малое потребление энергии. Ключи для быстродействующих ЦАП строятся на биполярных транзисторах.

2. Технические средства обработки информации

Основным инструментальным программно-аппаратным средством обработки данных в информационных системах является электронно-вычислительная машина (ЭВМ) **—** компьютер. За более чем полувековый период развития компьютер превратился из громоздкого, энергоемкого, дорогостоящего вычислительного средства в миниатюрное устройство, реализованного в виде персональной ЭВМ или микро-ЭВМ, выполненной в виде сверхбольшой интегральной микросхемы (СБИС), встраиваемой в информационные системы и устройства.

2.1.Обобщенная структурная схема компьютера

Компьютером (цифровой электронно-вычислительной машиной — ЭВМ), называется цифровая программно управляемая система, содержащая взаимосвязанные между собой процессор (П), запоминающее устройство (ЗУ), устройства ввода и вывода и программное обеспечение, предназначенная для арифметической и логической обработки и отображения данных. Процессор в свою очередь состоит из арифметико-логического устройства (АЛУ) и устройства управления (УУ).

Под **архитектурой компьютера** в общем случае понимают концептуальную структуру вычислительной машины, определяющей способы преобразования и обработки информации, принципы взаимодействия технических средств и программного обеспечения. В более подробную характеристику архитектуры входят структурная схема ЭВМ, средства и способы доступа к функциональным элементам схемы компьютера, организация и разрядность шин и интерфейсов ЭВМ, набор и доступность внутренних регистров, организация памяти и способы её адресации, набор и формат машинных команд процессора, способы представления и форматы данных, правила обработки прерываний.

Обобщенная структурная схема ЭВМ изображена на рисунке 2.1.

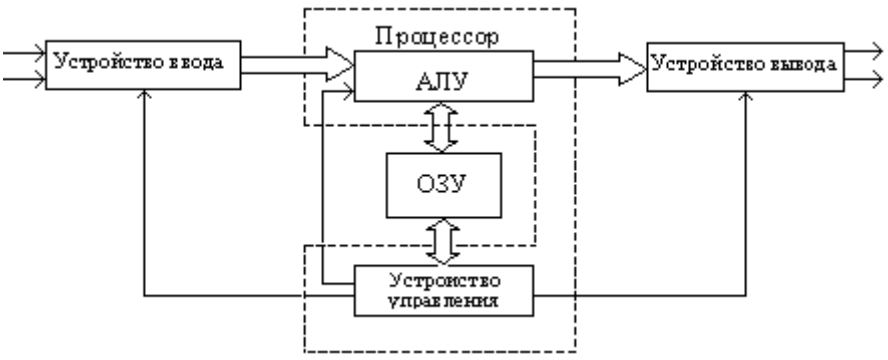


Рисунок 2.1 – Обобщенная структурная схема ЭВМ

Процессор *(П)* - часть ЭВМ для выполнения операций обработки данных. *Процессор* состоит из операционного блока - арифметико-логического устройства (АЛУ), устройства управления, регистров промежуточного хранения и шин сопряжения.

Отдельные составные части ЭВМ соединяются между собой группами линий одного функционального назначения – ***шин***, по которым передаются управляющие сигналы, данные и команды.

Устройства ввода предназначаются для занесения программы и входных данных в ЭВМ. Устройствами ввода-вывода могут быть, например, клавиатура, накопитель на магнитном диске, сканер, модем и др.

Память ЭВМ предназначена для хранения кодов программ, входных и выходных данных, результатов расчетов и промежуточных данных вычислений. Такая память называется оперативной (ОЗУ), поскольку функционирование ЭВМ определяется программой, хранимой в этом же запоминающем устройстве. Информация в ОЗУ хранится в виде отдельных слов, которые имеют размерность от 8 до 64 двоичных разрядов.

АЛУ (операционный блок) предназначен для выполнения в соответствии с программой, хранимой в памяти, арифметических, логических и сдвиговых операций над данными, находящимися в этой же памяти. Арифметическое устройство состоит обычно из накапливающего сумматора, регистров общего назначения, дешифраторов и других логических схем. Основной частью АЛУ является сумматор, поскольку все арифметические операции в процессоре выполняются посредством операций сложения или вычитания.

Арифметическое устройство (АЛУ) функционирует непосредственно под воздействием устройства управления (УУ). Оно может обрабатывать информацию последовательно, параллельно, либо параллельно-последовательно. АЛУ осуществляет операции, как правило, только над двумя числами (операндами). Если потребуется выполнить операции с большим количеством чисел, то применяется несколько циклов.

Устройство управления (УУ) в соответствии с алгоритмом, определяемым схемой этого устройства, последовательно производит выборку из ОЗУ отдельных команд программы и затем в соответствии с полученными командами вырабатывает управляющие сигналы, которые управляют составными частями ЭВМ. УУ организует также работу устройств ввода/вывода компьютера. Схемотехнически УУ содержит генераторы тактовых сигналов, счетчики, регистры, дешифраторы и различные комбинационные логические схемы.

Устройство вывода (печатающее устройство, графопостроитель, дисплей, и др.) обеспечивают перевод полученных результатов вычислений в форму, удобную для восприятия человеком, а также осуществляют документирование данных.

Функционирование компьютера в упрощенном виде можно описать следующим образом. После включения питания устройство управления процессора отправляет по шине адреса на ОЗУ определенный (на стадии проектирования компьютера) номер ячейки памяти, в которой хранится первая команда программы управления работой компьютера. Затем устройство управления генерирует

сигнал чтения памяти, по получении которого ОЗУ выставляет на шину код первой команды. Устройство управления процессора фиксирует этот код в регистре команды, дешифрирует ее и вырабатывает последовательность управляющих сигналов, требуемых для выполнения данной команды арифметико-логическим устройством. По завершении выполнения первой команды устройство управления формирует адрес следующей ячейки памяти и процесс получения и выполнения очередной команды повторяется. Команды управляют не только обработкой данных, но и вводом и выводом информации. Процесс чтения и выполнения команд циклически повторяется до получения команды останова.

2.2. Неймановская и гарвардская архитектуры компьютеров

В настоящее время наибольшее распространение в ЭВМ получили 2 типа архитектуры: *Неймановская* и *Гарвардская*. Неймановская архитектура компьютера основывается на следующих принципах.

Принцип однородности памяти.

Программы и данные хранятся в одной и той же памяти. Поэтому ЭВМ не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Принцип адресуемости памяти.

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к хранящимся в них значениям можно было бы впоследствии обращаться или менять их в процессе выполнения программы с использованием присвоенных имен.

Принцип последовательного программного управления.

Предполагает, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Принцип жесткости архитектуры.

Неизменяемость в процессе работы компьютера его структуры и списка команд.

Совместное использование шины для памяти программ и памяти данных приводит к ограничению пропускной способности между процессором и памятью по сравнению с объёмом памяти.

Для гарвардской архитектуры компьютера характерны следующие признаки:

1) хранилище команд (инструкций) и хранилище данных представляют собой разные физические устройства, в связи с чем машина гарвардской архитектуры имеет различные адресные пространства для команд и данных.

2) канал команд и канал данных также физически разделены.

В компьютере с использованием гарвардской архитектуры процессор может читать инструкции и выполнять доступ к памяти данных в одно и то же самое время. Благодаря этому компьютер с гарвардской архитектурой работает

быстрее. Недостатком гарвардской архитектуры является более высокая сложность аппаратной реализации компьютера.

Существуют гибридные архитектуры, сочетающие достоинства как Гарвардской так и н Неймановской архитектур. Так современные универсальные процессоры обладают раздельной кэш-памятью 1-го уровня для инструкций и данных, что позволяет им за один рабочий такт получать одновременно и команду, и данные для её выполнения. То есть процессорное ядро, формально, является гарвардским, но программно оно Неймановское, что упрощает написание программ.

2.3. Архитектура однокристального микропроцессора

Микропроцессор (МП) – это устройство для выполнения программно управляемых арифметических и логических операций, содержащее АЛУ, устройство управления, регистры и шины ввода-вывода информации, изготовленное в виде одной или нескольких больших интегральных схем (кристаллов). Компьютер, выполненный на основе микропроцессора, часто называют микро-ЭВМ.

Основная последовательность операций микро-ЭВМ следующая:

1) передача адреса очередной команды из МП в оперативную память (память программы);
2) считывание из памяти и декодирование команды в МП;
3) выполнение команды в МП;
4) формирование адреса следующей команды.

Последовательность этих операций называется циклом команды.

МП является основной частью — ядром микро-ЭВМ. Его архитектура в значительной степени определяет характерные параметры микро-ЭВМ. Типичная структурная схема МП имеет вид, изображенный на рисунке 2.2. В ее состав входят следующие функциональные узлы.

Аккумулятор – это регистр, в котором содержится операнд, подлежащий обработке в АЛУ. Его можно считать основным рабочим регистром, в котором записываются данные из памяти и результаты операций, заносимых обратно в память или в устройство ввода-вывода. Например, в соответствии с заданной командой в АЛУ подается команда выполнения операции сложения содержимого аккумулятора и другого регистра и записи полученной суммы в *Ak.*

Счетчик команд (СК). Команды, образующие программу, хранятся в памяти программы в определенной последовательности. В счетчике команды содержится адрес выполняемой в текущий момент команды. Обычно в процессе выполнения команд содержимое СК увеличивается на единицу. Однако в зависимости от вводимых данных либо от результатов выполнения операций может возникнуть необходимость изменения последовательности порядка выполнения команд. Для этого необходим переход к

новому адресу команды. С этой целью в СК непосредственно записывается адрес команды, к которой необходим переход. Для того чтобы затем была возможность обратного перехода в последовательности прежде выполнявшихся команд, необходимо запомнить адрес, предшествовавший переходу на новую последовательность команд. Этот адрес сохраняется в регистре стека.

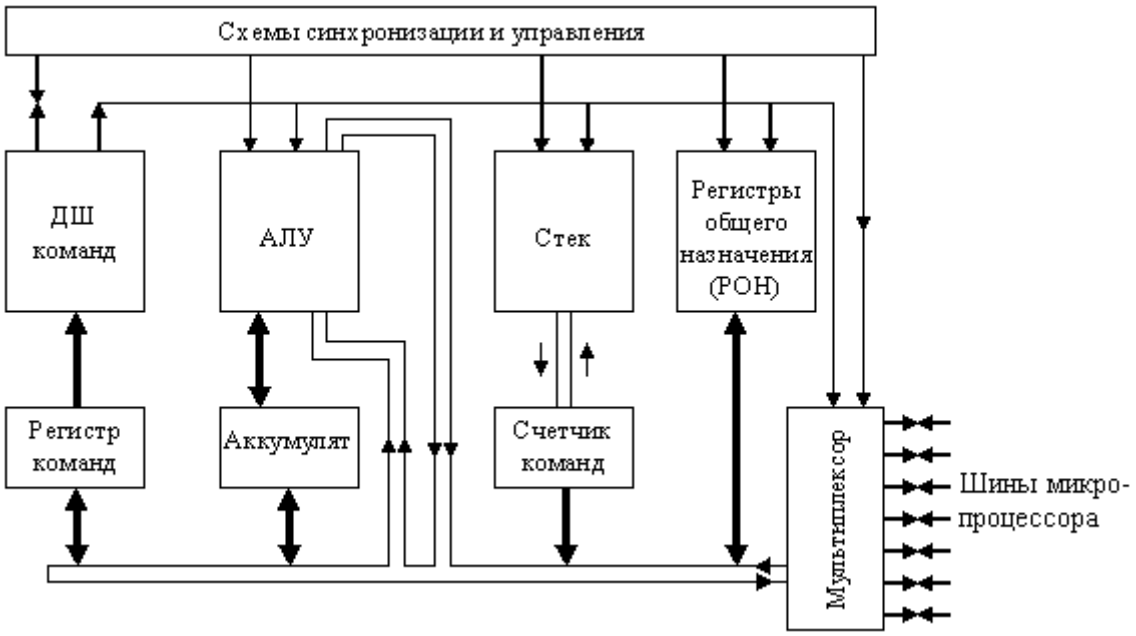


Рисунок 2.2 – Обобщенная структурная схема однокристального микропроцессора

Стек. Стек – это набор регистров или ячеек оперативной памяти, в котором данные или адреса выбираются “сверху” по принципу: “первый – поступивший последним”. При записи в стек очередного слова все ранее записанные слова смещаются на один регистр “вниз”. Например, если идет запись в последовательности A_1, A_2, A_3, A_4, то при считывании информация появляется в обратном порядке A_4, A_3, A_2, A_1. Т.е. нельзя извлечь раньше A_2, чем A_3 и т.п. Стек обычно используется в МП для хранения адресов возврата при обращении к подпрограммам, а также для запоминания состояния внутренних регистров при обработке прерываний. Важным параметром является число регистров стека (информационная емкость). При попытке записать в стек большее число слов, чем имеющаяся информационная емкость первое слово будет утеряно. Т.е. происходит переполнение стека. Для указания адреса последнего записанного в стек адреса применяется *указатель стека* (УС), являющийся специальным регистром.

Регистр команд. Команда, принимаемая МП из памяти, хранится в регистре команд (РК). Длина (формат) команды (т.е. число разрядов) зависит от типа МП. Простые МП имеют 8-разрядный код операции. Таким образом, всего может быть образовано 2^8=256 различных команд. Для практических целей такое общее число команд более чем достаточно. Несмотря на это, многие МП имеют

переменную длину команды, состоящую из одного, двух или трех байтов. С помощью таких команд осуществляется не только идентификация требуемой операции, но и задание кода одного или более адресатов для выбора данных и записи результата.

Дешифратор команд (ДШК) предназначен для определения операции, которую должен выполнить МП. Сигналы, образующиеся на выходах ДШ, управляют передачей информации между отдельными блоками и задают функции, выполняемые этими блоками. Как правило, команды делятся на группы, причем в каждой из групп выполняются аналогичные операции. Эти группы команд, например, “Передача данных”, “Арифметические операции”, “Логические операции” и т.д. различаются по четырем старшим разрядам КОП. Такая группировка значительно упрощает декодирование команд.

Регистры общего назначения (РОН) – сверхоперативная память. Эти регистры применяются в качестве временного запоминающего устройства для различной информации (адресов и данных), которую можно извлечь просто и с большей скоростью, чем из ОЗУ. Поэтому блок памяти РОН называют сверхоперативной памятью. Обращение к РОН – адресное. Эти регистры допускают считывание и запись информации, в связи с чем содержат входную и выходную шины, адресную шину и управляющие входы, информация на которых задает режим работы регистра: запись, чтение или хранение.

Арифметико-логическое устройство – (АЛУ) входит в состав всех процессоров, хотя его принципиальная схема, функции, быстродействие и т.д. могут быть существенно различными. АЛУ, как составная часть МП, должно выполнять по крайней мере следующие операции:

- сложение с переносом;
- вычитание с переносом (заемом);
- сдвиг влево и вправо;
- счет в прямом и обратном направлении;
- логическое умножение и сложение (И, ИЛИ);
- сравнение кодов.

Более сложные АЛУ могут выполнять и другие функции.

Схемы синхронизации и управления – (ССУ) совместно с ДШК называют устройством управления. Устройство управления расшифровывает поступающую команду и в соответствии с ней вырабатывает необходимую последовательность сигналов, управляющих работой всех остальных блоков МП. Следует заметить, что последовательность управляющих сигналов зависит также от характеристик промежуточных состояний (вычислений) и наличия дополнительных сигналов (прерывание, запрос и т.д). В состав микропроцессора входит также *регистр состояния,* не показанный на схеме. Он состоит из одного или нескольких триггеров, называемых “флажками”. Эти триггеры предназначены для хранения информации о состоянии МП и индикации этого состояния. Например, индикация нулевого содержимого аккумулятора, признак наличия единицы переноса при выполнении арифметических операций, знака содержимого аккумулятора и т.д. Данный регистр состояния обеспечивает возможность интерпрета-

ции результатов, полученных при вычислениях, и часто используется для реализации условных переходов.

2.4. Структурная схема 8-разрядного однокристального МП

В качестве примера рассмотрим организацию и особенности функционирования МП типа Intel 8080 (отечественный аналог К580ВМ80) – основного элемента микропроцессорного комплекта серии К580. Структурная схема МП представлена на рисунке 2.3.

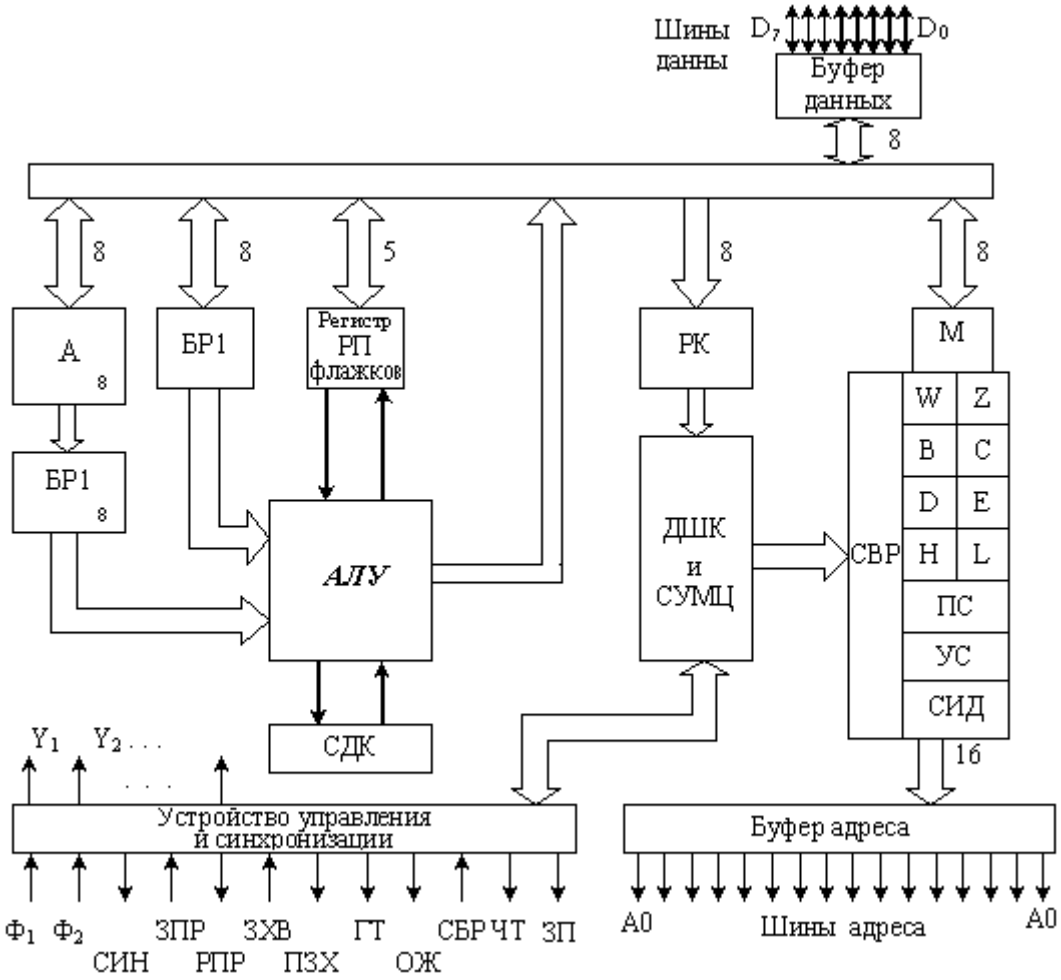


Рисунок 2.3 – Структурная схема однокристального микропроцессора Intel 8080

МП содержит шесть 8-разрядных регистров общего назначения РОН (B,C,D,E,H и L) с мультиплексором регистров M, восьмиразрядный аккумулятор A, четыре 8-разрядных буферных регистра БР1, БР2, W и Z, а также 5-разрядный регистр признаков РП. В состав МП входят также регистр команд РК, дешифратор команд ДШК, схема управления машинным циклом СУМЦ, схема десятичной коррекции СДК, схема выборки регистров СВР, программный счетчик ПС, указатель стека УС (в качестве стека используется часть ОЗУ), схема

инкрементации-декрементации СИД, устройство управления, а также буферные регистры данных и адреса.

Определяющим для описания любого МП является состав его внешних выводов. МП Intel 8080 размещен в корпусе с 48 выводами, но используются только 40 из них. Внешние выводы имеют следующее назначение:

$D_0 – D_7$ − двунаправленная шина данных;

$A_0 – A_{15}$ – шина адреса, допускающая подключение памяти объемом 2^{16} = 16Кбайт;

СИН (SYNC) – выход синхроимпульса, вырабатываемого МП в начале каждого машинного цикла;

СБР (RESET) – “Сброс”, вход установки микропроцессора в начальное (нулевое состояние);

ЧТ (DBIN) – “Чтение”, выход, сигнал на котором обозначает, что МП принимает информацию с шины данных. Используется для отпирания вентильных схем с целью передачи информации на шину данных из памяти или устройств ввода;

ЗП (WR) – “Запись” – выход, сигнал на котором обозначает, что МП выдает информацию на шину данных. Используется для синхронизации записи информации с шины данных в память или ее передачи в устройства вывода. Сигнал имеет инверсное значение;

ГТ (READY) – “Готовность” – вход сигнала готовности от устройства памяти или ввода. Низкий уровень на этом входе после выработки МП сигнала ЧТ указывает на то, что память или устройство ввода не готовы к передаче информации в МП. Микропроцессор в этом случае входит в состояние “Ожидание”;

ОЖ (WAIT) – “Ожидание” – выход, признак нахождения МП в состоянии ожидания;

ЗХВ (HOLD) – вход, сигнал на котором заставляет МП войти в режим ЗАХВАТ, когда адресная шина и шина данных МП переходят в состояние высокого сопротивления. Это позволяет внешнему устройству получить управление обеими шинами, например, для прямого доступа в память;

ПЗХВ (HLDA) – “ПРИЗНАК ЗХВ”, выход, сигнал на котором информирует о нахождении МП в режиме ЗАХВАТ;

ЗАПР (INT) – вход запросов на прерывание;

РПР (INTE) – выход сигнала, указывающего на разрешение прерывания;

$Ф_1$, $Ф_2$ – входы для подачи тактовых сигналов;

+12; +5; -5; КОРПУС – входы для подключения питания.

Функциональные блоки имеют следующее назначение.

1) *Матрица РОН и схемы адресации*:

а) *Матрица регистров* представляет собой статическое ОЗУ, состоящее из шести 16-разрядных регистров. *Пары регистров B-C; D-E и H-L* являются шестью 8-разрядными регистрами сверхоперативной памяти, которые можно использовать как шесть одинарных 8-разрядных регистров или как три пары 16-разрядных регистров. Пара регистров временного хранения W-Z предназначена для внутренних команд и не находится в распоряжении программиста;

б) *Программный счетчик (Счетчик команд).* 16-разрядный счетчик, предназначен для хранения адреса текущей команды выполняемой программы; после выборки любой команды происходит увеличение хранимого адреса на единицу;

в) *Указатель стека*. 16-разрядный УС содержит адрес очередного уровня стека в памяти. Содержимое УС уменьшается на единицу при поступлении данных в стек, а при выборке из стека - соответственно увеличивается. УС может использовать любую область памяти ОЗУ, что обеспечивает практически неограниченные возможности вложения подпрограмм;

г) 16-разрядный *фиксатор адреса*, загружаемый данными любой из трех пар регистров сверхоперативной памяти, обеспечивает прямую адресацию памяти через буферы адреса.

2) *АЛУ* – служит для выполнения арифметических и логических операций, а также операций циклического сдвига. При реализации операций с АЛУ используется аккумулятор, буферные регистры БР1, БР2 и 4-разрядный регистр признаков (флажков).

3) *Регистр команд РК* представляет собой 8-разрядный регистр, предназначенный для передачи данных от внутренней шины к ДШК и схемам управления. Выходные сигналы ДШК и сигналы управления от внешних устройств подаются на устройство управления и синхронизации.

4) *Устройство управления и синхронизации* принимает из ДШК и СУМЦ информацию о поступившей команде (из памяти) и в соответствии с сигналами на управляющих входах ЗАХВ, ЗАПР и ГТ обеспечивает ее выполнение выработкой необходимых управляющих сигналов *Yi* для всех блоков МП. Устройство управления и синхронизации вырабатывает также сигналы для управления памятью и устройствами ввода-вывода.

2.5. Функционирование и временные диаграммы микропроцессора

Формат команды МП580ВМ80 содержит от одного до трех байт. Время, затрачиваемое на извлечение 1 байта информации или выполнение команды, определяемой одним машинным словом, называют *машинным циклом* (М). Каждая команда требует для выборки и выполнения от одного до пяти машинных циклов. Машинные циклы именуются M_1, M_2, M_3, M_4, M_5.

Выполнение каждой команды в МП происходит в строгой последовательности, определяемой кодом команды, и синхронизируется сигналами $Ф_1$ и $Ф_2$ тактового генератора. Период синхросигналов $Ф_1$ или $Ф_2$ называется *машинным тактом* (Т). Любой машинный цикл включает от трех до пяти тактов: T_1, T_2, T_3, T_4, T_5. Каждый такт длится в течение одного периода синхросигнала (длительность такта при частоте 2 МГц = 0,5 мкс). Имеется три состояния, которые могут длиться неограниченное число тактов: WAIT (Ожидание), HOLD (Захват), HALT (Останов).

Время выполнения команды определяется процессом получения, декодирования и ее выполнения. В зависимости от вида команды это время может состо-

ять от 1 до 5 М. Для микропроцессора 580ВМ80 существует десять различных типов циклов:

1) M_1 – извлечение кода команды;
2) M_2 – чтение данных из памяти;
3) M_3 – запись данных в память;
4) M_4 – извлечение из стека;
5) M_5 – запись данных в стек;
6) M_6 – ввод данных из внешних устройств;
7) M_7 – запись данных во внешние устройства;
8) M_8 – цикл обслуживания прерывания;
9) M_9 – останов;
10) M_{10} – обслуживание прерывания при работе МП в режиме “Останов”.

Цикл M_1 – это всегда цикл выборки команды, он длится от 4 до 5 тактов. Циклы M_2, M_3, M_4, M_5 обычно состоит из 3-х тактов каждый. На рисунке 2.4. показаны временные диаграммы функционирования МП.

В течение такта T_1 содержимое программного счетчика ПС выдается на адресную шину, а на выходах СИН вырабатывается высокий потенциал. На шину данных подается 8-разрядный код, характеризующий выполняемый цикл. На первом такте каждого машинного цикла МП указывает тип выполняемого цикла с помощью 8-разрядного слова состояния цикла, выдаваемого на шины данных. Слово состояния выдается на шины данных лишь во время импульса СИНХР (такты T_1 и T_2), а используется на протяжении всего машинного цикла. Поэтому его необходимо записывать в специальный регистр слова состояния РгСС. Запись его осуществляется в момент совпадения сигналов СИНХР и $Ф_1$ на втором такте (рисунок 2.4).

Слово состояния в последующем используется для формирования сигналов раздельного обращения к памяти и внешним устройствам, так как в процессоре такие сигналы отсутствуют (например, Чт (Прием) относится как к памяти, так и к внешним устройствам).

За T_1 всегда следует такт T_2, в течение которого проверяется наличие сигналов подтверждения ГТ и ЗАХВАТ, а также проверяется не находится ли МП в состоянии останова HALT. Если на входе READY имеется сигнал готовности (высокий уровень), то МП переходит к такту T_3, в противном случае – в состояние ОЖИДАНИЕ (такт T_w) и находится в нем до тех пор, пока не появится сигнал готовности. Таким образом, сигнал ГОТ позволяет синхронизировать МП с памятью с любым временем доступа или с любым внешним устройством. Более того, сигнал ГОТ позволяет осуществить пошаговое выполнение программы.

Во время T_2 слово состояния цикла (PSWC) записывается в регистр состояния. Передним фронтом $Ф_2$ заканчивается формирование сигнала СИН, и вырабатывается единичный сигнал *Прием*, позволяющий поступить байту на вход МП через ШФ. В этом же такте T_2 из сигнала *Прием* и D_7 *PSWC* формируется сигнал *Чт Память*, позволяющий поступать данным из памяти на ШД микропроцессора. Изменения данных в этом такте восприниматься не будут так как их запись в МП осуществляется в фиксированные моменты времени в такте T_3.

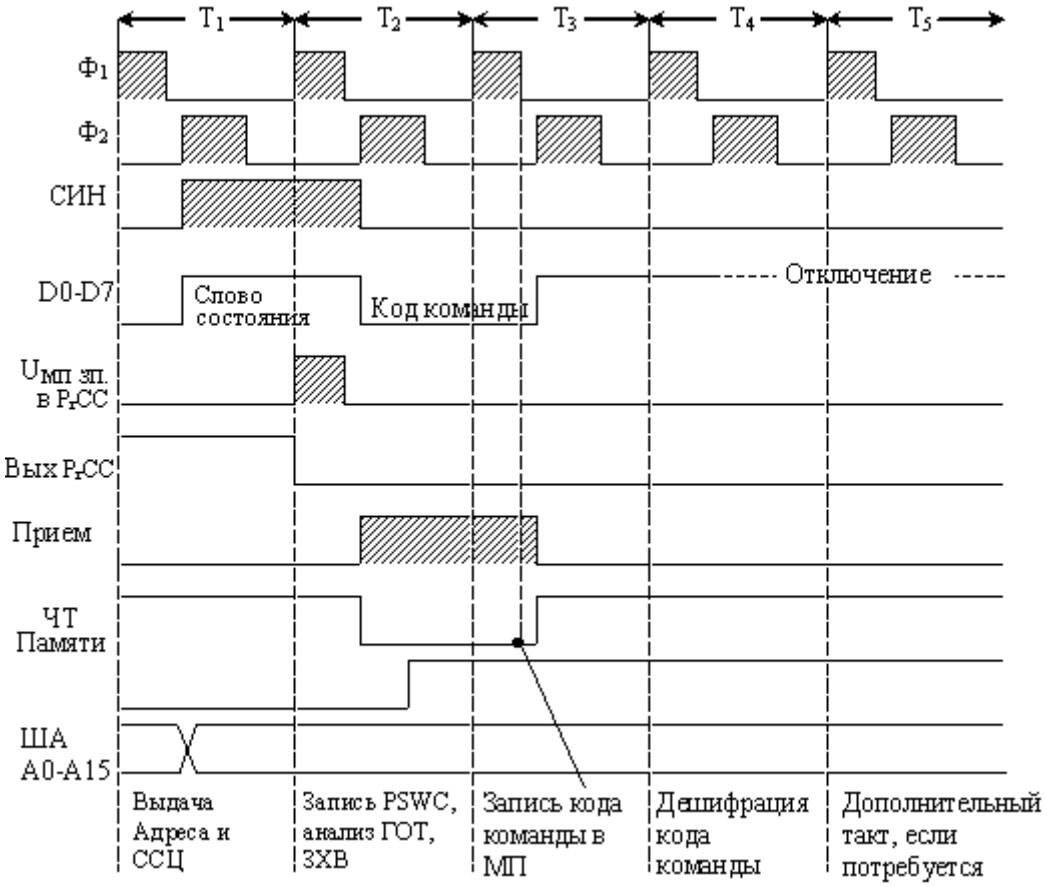


Рисунок 2.4 – Временные диаграммы функционирования процессора i8080

В такте T_3 во время заднего фронта $Ф_1$ производится запись кода команды во внутренний регистр кода команды. Положительным фронтом $Ф_2$ оканчивается сигнал ПРИЕМ на выходе МП и сигнал ЧТ Память. Импульс на выходе Прием формируется в машинных циклах: чтение команды, ЧТ данных из памяти, прерывания, чтение из стека или внешнего устройства.

На основании декодирования команды ДШК схема управления формируют сигналы управления и синхронизации для внутренних пересылок данных, а также соответствующие дешифрируемой команде машинные циклы.

На последующих тактах T_4 и T_5 ДШК расшифровывает код команды, определяет количество байтов в команде, формирует команды на внутренние пересылки данных и подготавливает МП к выполнению следующих машинных циклов.

В конце последнего машинного цикла выполнения каждой команды анализируется наличие запроса прерывания на входе ЗПР. Если запрос присутствует и прерывания разрешены (команда *EI*), то МП входит в специальный цикл M_1, во время которого содержимое ПС не изменяется, формируется признак начала об-

работки прерывания INTA, а прерывающее устройство посылает в МП код команды RST с адресом прерывающей программы.

Самые *простые команды*, не требующие обращения к памяти, выполняются в течении одного машинного цикла *за четыре такта,* т.е. за 2 мкс, *самые длинные* – на протяжении 5 машинных циклов – за *18 тактов*, т.е. 9 мкс.

Выборка команд длиной 2 и 3 байта производится соответственно за два или три машинных цикла, при этом *первый байт* команды заносится в регистр команд *РК*, второй в программно недоступный регистр *W* , а третий – в регистр *Z*.

В начале каждого машинного цикла на шину данных выдается байт состояния (в течение действия сигнала СИН). Назначение каждого разряда слова состояния D_i следующее:

INTA	- D_0	- Сигнал подтверждения прерывания. Используется для синхронизации передачи в МП из прерывающего устройства адреса прерывающей программы;
WO	- D_1	- Признак того, что в данном машинном цикле будет выполняться запись в память или вывод информации (WO=0). В противном случае будет выполняться ввод или чтение из памяти;
STACK	- D_2	- Указывает, что адресная шина содержит адрес одной из ячеек зоны ОЗУ, используемой в качестве стека;
HLTA	- D_3	- Подтверждение выполнения процессором операции останова (HALT);
OUT	- D_4	- Указывает, что на адресной шине, содержится адрес устройства вывода, на шинах данных – выводимая информация;
M1	- D_5	- Признак машинного цикла выборки первого байта команды
INP	- D_6	- Указывает, что на адресной шине находится адрес устройства ввода, а ввод будет производится на шину данных по сигналу Прием на выводе DBIN;
MEMR	- D_7	- Признак того, что в данном машинном цикле будет выполняться чтение из памяти.

Как видно из схемы восьмиразрядного микропроцессора, у него нет раздельных сигналов чтение/запись памяти и чтение/запись устройств ввода/вывода, а лишь общий сигнал Чтение /Запись. Для формирования таких сигналов и используется информация о слове состояния цикла МП. На рисунке 2.5 изображена упрощенная схема микро-ЭВМ на основе 8-разрядного МП.

Схемы совпадения служат для формирования на основе слова состояния раздельных сигналов записи и чтения в память или внешние устройства.

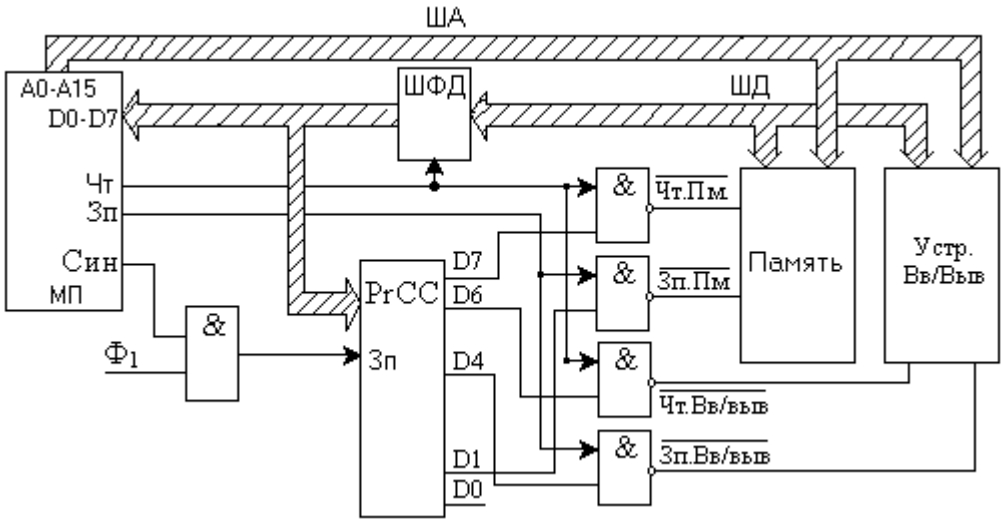


Рисунок 2.5 – Схема компьютера на основе 8-разрядного микропроцессора

2.6. Программируемый контролер прерываний

В первых персональных компьютерах IBM PC в качестве контроллера прерываний использовалась микросхема i8259 (отечественный аналог — БИС КР580ВН59). Данный контроллер прерываний представляет собой законченное устройство, которое позволяет реализовать восьмиуровневую векторную систему прерываний с возможностью маскирования и динамического изменения дисциплины обслуживания.

Для перехода к подпрограммам обслуживания прерываний контроллер формирует и подает на шину данных микропроцессора код команды CALL. За счет каскадного включения КР580ВН59 число обслуживаемых уровней прерывания может быть увеличено до 64. Контроллер может использоваться как для организации обмена информацией в режиме прерывания, так и для организации программно-управляемого обмена. В первом случае БИС ВН59 на приоритетной основе формирует запрос на прерывание для микропроцессора и адрес подпрограммы обслуживания. Во втором случае процессор считывает слово состояния контроллера, и определят устройство с наивысшим приоритетом, готовое к обмену. Микросхема размещена в пластмассовом корпусе с 28 выводами и потребляет мощность 1 Вт при напряжении питания +5 В.

Контроллер позволяет реализовать простой приоритетный режим и режим циклического приоритета обслуживания прерываний. При реализации простого приоритетного режима всем восьми входам запросов на прерывание присваиваются фиксированные приоритеты, причем наивысшие приоритет присваивается входу ЗП0, наименьший ЗП7. В режиме циклического приоритета после окончания обслуживания любого устройства приоритет входов контроллера циклически изменяется таким образом, что устройству, обслуженному последним, при-

сваивается низший приоритет. Кроме того, в режиме циклического приоритета низший приоритет может быть присвоен любому входу запроса программным способом.

Структурная схема контроллера прерываний показана на рисунке 2.6. Запросы на прерывание от внешних устройств подаются на входы ЗП0 – ЗП7 и запоминаются в регистре запросов. В регистре состояния содержатся все запросы на прерывания, обслуживаемые в данный момент. Регистр маски содержит единицы в разрядах, соответствующих маскируемым в настоящий момент входам запросов. Установка в единицу того или иного разряда регистра маски блокирует передачу запроса на прерывание.

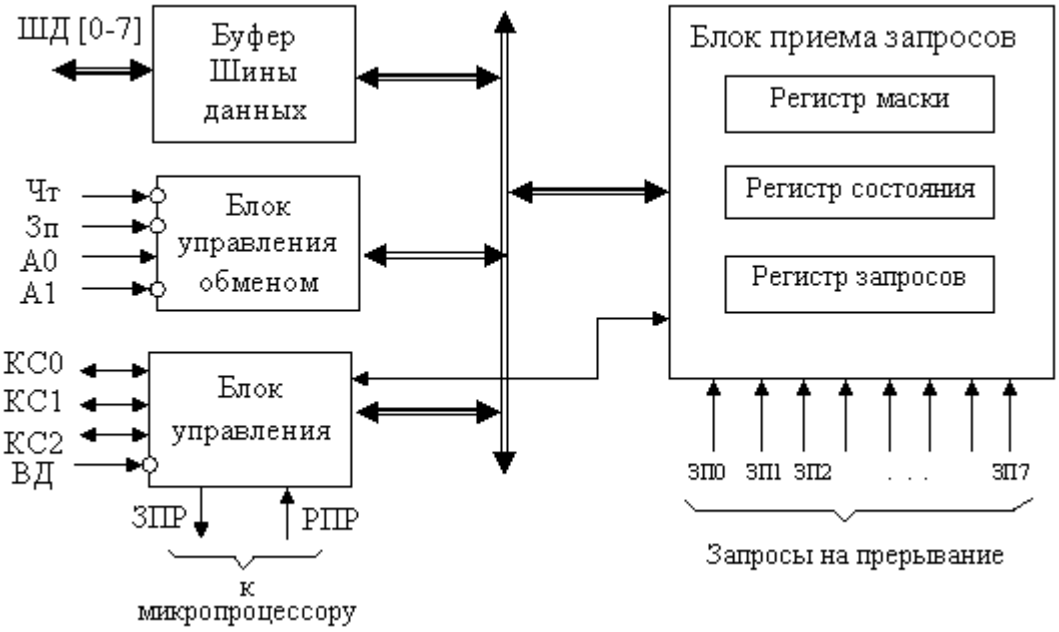


Рисунок 2.6 – Схема программируемого контроллера прерываний

Запросы на прерывание по любому входу могут быть поданы в потенциальной или импульсной форме. Однако каждый последующий запрос на прерывание воспринимается контроллером только после выполнения подпрограммы обслуживания текущего запроса по данному входу и сброса соответствующего разряда регистра состояния, что осуществляется специальной командой программным способом. *Сигналы управления контроллера прерываний КР580ВН59* имеют следующее значение:

ЗП — запись в БИС управляющих слов;

ЧТ — считывание из БИС содержимого внутренних регистров;

А0 — адресация регистров;

ВМ — выбор микросхемы;

ЗП0-ЗП7 — запросы на прерывание от внешних устройств;

ЗПР — запрос на прерывание, выдаваемый контроллером на микропроцессор;

РПР — разрешение прерывания. После поступления этого сигнала от микропроцессора контроллер осуществляет ввод в МП команду CALL;

ВДМ — ведомый; сигнал управления каскадирования. Высокий уровень — если контроллер является ведущим, низкий - ведомым;

КС0-КС2 — каскадирование, линии являются выходными, если контроллер ведущий, и входными — если ведомым.

Значение сигналов КС0 – КС2 ведущего контроллера, подаваемые на соответствующие входы КС0 – КС2 подчиненных, указывает подчиненный контроллер, который формирует и выдает в микропроцессор адрес своей подпрограммы обслуживания.

Программирование контроллера осуществляется двумя типами команд: *командами инициализации* и командами *управления режимом*. Ввод команд для контроллера прерываний осуществляется микропроцессором, как правило, командой OUT. Однако в системе может быть организовано обращение к контроллеру как к ячейкам ЗУ. **Команды инициализации** подаются перед началом работы контроллера. Эти команды задают стартовые адреса подпрограмм обслуживания прерываний, расстояниями между соседними стартовыми адресами и указывают, если необходимо, на наличие других контроллеров в системе. **Команды управления режимом** (*операциями*) служат для оперативного изменения режимов обслуживания прерываний и могут подаваться в любое время в процессе работы контроллера. **Команда управления операциями** (*OCW1*) осуществляет установку или сброс разрядов *регистра маски*. Установка определенного разряда регистра маскирования приводит к запрету прерывания по соответствующему входу. Команда управления *OCW2* осуществляет циклический *сдвиг приоритета запроса.* Команда управления операциями *OCW3* позволяет задать режим специального маскирования, при котором можно *выборочно* устанавливать *приоритеты прерывания*. Этой же командой можно задать режим опроса и произвести считывание состояния ПКП, при котором считывается значение регистра обслуживаемых прерываний либо регистра запроса прерываний.

При поступлении запроса на прерывание по одному из входов ЗП0-ЗП7 он фиксируется в регистре запросов, а на выводе ЗПР (INT) формируется сигнал запроса на прерывание, который поступает на соответствующий вход микропроцессора. Если запросы на прерывание не запрещены (специальной командой), то процессор завершает текущий цикл, сохраняет в стеке состояние программного счетчика и выдает контроллеру сигнал разрешения прерывания РПР (INTA). По этому сигналу контроллер выставляет на шину данных код команды вызова подпрограммы CALL, который поступает в регистр команд процессора. После дешифрации этой команды процессор последовательно выдает еще два сигнала РПР. По первому из них контроллер выдает младший байт адреса подпрограммы обслуживания прерываний, а по второму – старший. Эти байты заносятся в программный счетчик и процессор начинает выполнять команду, начиная из адреса (вектора), поступившего от контроллера прерываний. Подпрограмма обслуживания прерываний должна заканчиваться командой возврата RET, по которой два байта из стека загружаются в программный счетчик и процессор продолжает выполнять прерванную программу.

В процессоре Pentium был добавлен расширенный контроллер прерываний APIC (*Advanced PIC*). Он состоит из модуля, встроенного в сам процессор (в случае многоядерной системы — в каждое ядро), называемого локальный контроллер прерываний (англ. local APIC), и центрального модуля, выполненного в одном экземпляре даже на многоядерном оборудовании, обычно как часть микросхем обрамления процессора (англ. IO APIC).

Проводники IRQ от прерывающих устройств подсоединены к IO APIC. Для общения локального APIC и IO APIC, а также локальных APIC различных ядер друг с другом, используется системная (*frontside*) шина многопроцессорной системы, также используемая для соединения процессоров и контроллера памяти.

Преимуществом расширенного контроллера прерываний является следующее:

1) возможность реализации межпроцессорных прерываний — сигналов от одного процессора к другому;
2) поддержка до 256 входов IRQ, в отличие от 16 на классической IBM PC
3) очень быстрый доступ к регистрам текущего приоритета прерывания и подтверждения прерывания.

В настоящий момент наблюдается тенденция к отказу от IO APIC, как и проводников IRQ, и переходу на прерывания, инициируемые сообщениями MSI — (*Message Signaled Interrupts*).

MSI — альтернативная форма прерываний: вместо присваивания номера запроса на прерывание, прерывающему устройству разрешается записывать сообщение по определённому адресу системной памяти. Для записи сообщения используется механизм прямого доступа в память, причем устройство может иметь от одной до тридцати двух уникальных областей памяти. Все прерывания шины PCI Express всегда доставляются как MSI, даже при использовании эмуляции традиционных номеров проводников прерываний.

Достоинства MSI состоят в следующем:

- возможность передачи некоторых данных вместе с информацией о наступлении события, что зачастую избавляет обработчик прерывания от необходимости читать данную информацию из регистров состояния устройства, что уменьшает загрузку шины;
- возможность полного отказа от проводников INT# от устройств и разъемов PCI до главного контроллера прерываний (IO APIC), а также от самого главного контроллера прерываний, что упрощает системную плату;
- в многопроцессорных и многоядерных системах устройства получают возможность самостоятельно выбирать процессор/ядро для обработки конкретного прерывания, причем делать это полностью на уровне аппаратуры без исполнения программного кода, что позволяет оптимизировать работу путем размещения большей части структур драйвера устройства и связанного с ним программного обеспечения (сетевых протоколов и т.д.) в кэше конкретного процессора.

2.7. Программируемый контроллер прямого доступа к памяти

Контроллер прямого доступа к памяти (ПДП) служит для организации обмена данными между внешними устройствами и памятью, без участия центрального процессора. В персональных микро-ЭВМ используется контроллер ПДП типа КР580ВТ57, который имеет четыре независимых канала обмена, каждый из которых адресует свою область внешней памяти путем последовательного инкрементирования выбранного адреса. Контроллер ПДП имеет приоритетную логику, реализующую запросы от четырех периферийных устройств и производит счет циклов ПДП каждого канала.

Каждый канал ПДП имеет регистр адреса и регистр количества циклов (оба 16 разрядов), шину адреса и шину данных. Шина адреса разделена на две части (А3 – А0) – двунаправленная часть шины адреса, при работе которой во входном режиме указанные разряды адреса используют для инициализации определенного канала, а в выходном режиме – как младший полубайт адреса внешней памяти. А7 – А4 являются выходами и используются для выдачи адреса на внешнюю память. Шина данных (D7 – D0) обеспечивает двунаправленный обмен между микропроцессором и КПДП. По этой шине принимаются управляющие слова и выдается старший байт адреса внешней памяти.

Сигнал Чт В/В разрешает (во входном режиме) чтение регистра состояния (РС) или записанных в ЗУ контроллера *начального адреса и числа циклов* передачи данных (ПД) любого из каналов. В выходном режиме Чт В/В разрешает выдачу информации из внешнего устройства ввода – вывода. Сигнал Зп В/В разрешает во входном режиме загрузку *регистров установки режимов*, начального адреса и количества циклов для любого канала ПДП. В выходном режиме Зп В/В разрешает запись информации во внешнее устройство ввода – вывода. Выходные сигналы Чт П и Зп П предназначены для внешнего ЗУ.

Входные сигналы ЗПД3 – ЗПД0 поступают асинхронно с внешних устройств и воспринимаются БИС как запросы на обмен с ЗУ. Выходные сигналы ППД3 – ППД0 являются ответными по отношению к ЗПД. Они вырабатываются контроллером в соответствии с приоритетом внешних устройств. Сигнал КС – вырабатывается при установке в нуль шестнадцатиразрядного регистра количества циклов, и указывают периферийному устройству, что данный цикл ПД последний. Сигнал на выводе М128 – появляется на каждом 128 цикле от конца массива, а также во время действия сигнала КС.

Структурная схема БИС КР580ВТ57 показана на рисунке 2.7. Схема приема запросов (СПЗ) предназначена для приема и формирования ответных сигналов запроса в соответствии с приоритетом внешних устройств. Внутреннее запоминающее устройство (ВЗУ) служит для хранения начального адреса и числа циклов ПД для каждого канала. Для этого в нем расположены для каждого канала шестнадцатиразрядный регистр адреса РгА и регистр циклов РгЦ. РгА загружается адресом первой ячейки памяти, к которой будет обращение. Младшие 14 разрядов РгЦ указывают число циклов минус 1 до конца счета (сигнал КС). Разряды 14 и 15 РгЦ указывают на вид обмена 00 – контроль, 01 – запись в ЗУ, 10 – чтение ЗУ, 11 – запрещенная комбинация.

Схема формирователя сигналов Зп/Чт обеспечивает обмен информацией между микропроцессором и БИС контроллер с одной стороны, и БИС и ОЗУ и прерывающее устройство – с другой.

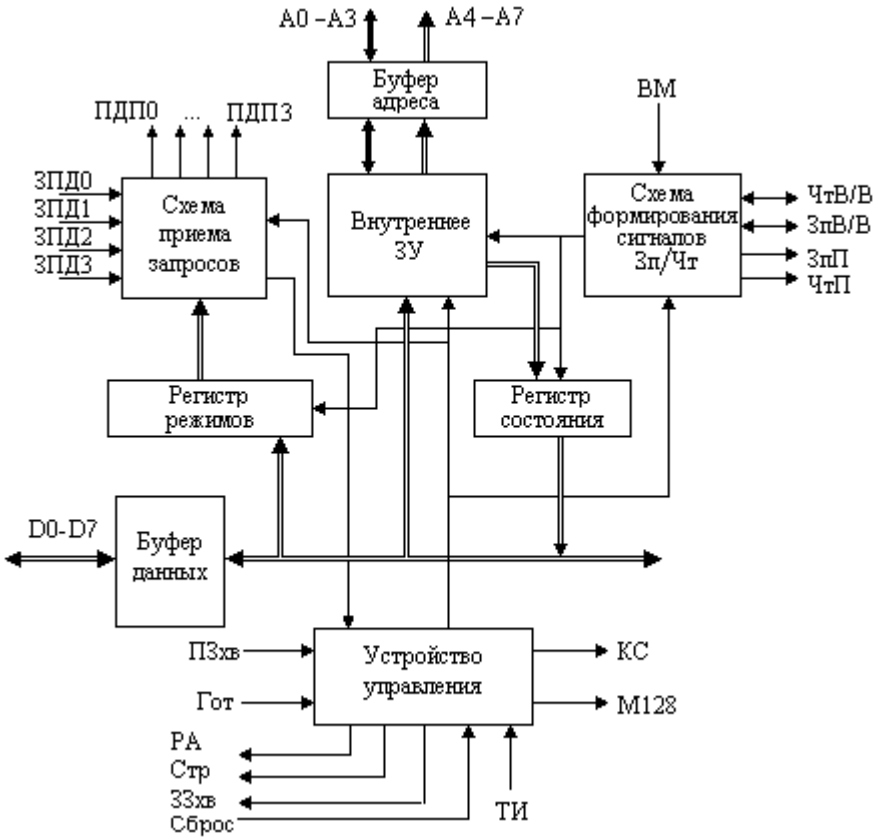


Рисунок 2.7 – Структурная схема контроллера прямого доступа в память

Регистр установки режимов РгР хранит информацию о режимах работы БИС, к которым относятся «Автозагрузка», «Конец счета-стоп», «Удлиненная запись», «Обычная запись», «Циклический сдвиг приоритета» и «Фиксированный приоритет». Разряды 3÷0 РгР разрешает работу соответствующего канала, а 7÷4 –обеспечивает необходимый режим контроллера. *Регистр состояния каналов РгС* указывает номер канала, который достиг конца счета.

Остальные линии БИС имеют следующее назначение:

A0 – A7 – инициализация канала или младший полубайт адреса.

Стр А – указывает, что на шине данных выдан старший байт адреса внешнего ЗУ.

РА – разрешение адреса для блокировки адресных шин в невыбранных устройствах, указывает, что протекает цикл ПД

ТИ – тактовый импульс Ф2ТТЛ –уровень.

Гот – для обеспечения совместной работы ПДП и медленных внешних устройств.

Микросхема КР580ВТ57 может находиться в одном из следующих состояний: исходном, программирования, ожидания, обслуживания.В исходное состояние микросхему переводит внешний сигнал установки. В этом состоянии маскируются все запросы признаков ПД, а буферы А3 ÷ А0 переводятся в состояние приема информации. В состоянии программирования микропроцессор имеет доступ к внутренним регистрам выбранного канала в соответствии с состоянием управляющих сигналов. В состоянии ожидания БИС находится либо от момента окончания программирования до выдачи сигнала ПДП, либо в промежутках между циклами ПД.

При получении от внешнего устройства сигнала запроса на ПДП (ЗПД0-ЗПД3), например, на вывод данных из памяти, контроллер вырабатывает сигнал запрос захвата ЗЗхв, который поступает на соответствующий вход микропроцессора. МП на каждом втором такте машинного цикла проверяет состояние этого входа и при активном сигнале ЗЗхв переводит свои шины в третье (высокоимпедансное) состояние, т.е. отключается от шин данных, адреса и управления и выдает сигнал подтверждение захвата ПЗхв. В таком состоянии процессор находится, пока на его входе присутствует активный сигнал ЗЗхв. Управляемыми остаются только линии процессора ЗЗхв и ПЗхв.

После получения от микропроцессора сигнала ПЗхв контроллер выдает сигнал ППД, соответствующему внешнему устройству и переходит в состояние обслуживания ПДП, в котором системные шины находятся под управлением контроллера ПДП. При этом на шину адреса по линиям А0-А7 котроллер выставляет младший байт адреса, а на шину данных – старшую часть адреса. Присутствие адреса на шине данных сопровождается стробом адреса СтрА, который используется для записи этой части адреса во внешний регистр. Вследствие этого на память поступает 16-разрядный адрес ячейки с которой начинается прямой доступ. После снятия адреса с шины данных контроллер вырабатывает сигнал чтения содержимого ячейки памяти ЧтП, в результате чего память выставляет на шину данных восьмиразрядное слово, которое поступает на внешние устройства. После завершения переходного процесса установки данных на шине контроллер вырабатывает сигнал записи во внешнее устройство ЗпВ/В. Очевидно, что данные будут восприняты только тем устройством, которое получило одновременно сигналы ППД и ЗпВ/В.

Затем контроллер ПДП уменьшает счетчик циклов на 1 и, если счет не закончился, повторяет цикл чтения из памяти и запись во внешнее устройство. Запись в память в режиме ПДП осуществляется аналогично, но при этом контроллер формирует сначала сигнал ЧтВ/В, а потом ЗпП. Если содержимое счетчика циклов стало равным нулю, то контроллер формирует сигнал конец счета КС и снимает сигнал ЗЗхв. Процессор возвращается в рабочее состояние и продолжает управление шинами самостоятельно.

Особенности программирования БИС контроллера ПДП:

1. При записи информации в регистры РА и РЦ выполняются по два обращения к регистрам, причем вначале записывается младший байт, а затем – старший. Запись осуществляется по одному и тому же адресу. Операцию записи

шестнадцатиразрядного слова в регистры необходимо всегда выполнять полностью.

2. В микропроцессорной системе, работающей с прерываниями, следует запрещать прерывания перед выполнением двойного обращения к регистрам.

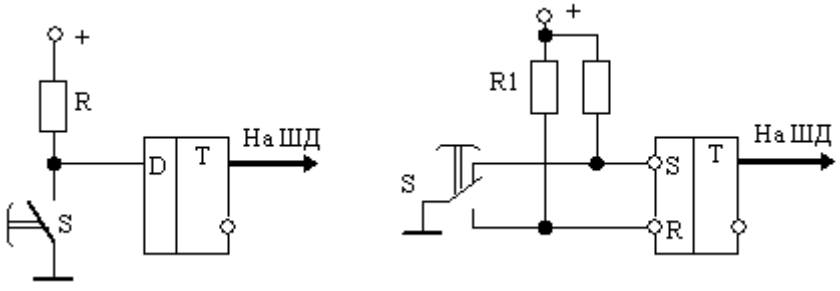
3. После сигнала Сброс в регистре режимов РР содержатся нули и работа всех каналов запрещена. Запись байта в РР необходимо всегда выполнять после загрузки параметров в РА и РЦ.

4. После каждого запроса ПДП контроллер необходимо перепрограммировать, если не был установлен режим автозагрузки.

В контроллерах ПДП для работы с 20- 24-разрядной шинами адреса добавляется 8-битовый регистр страниц. Для 20-разрядной ША используется 4 бита этого регистра, а для 24-разрядной – все 8.

2.8. Подключение клавиатуры и устройств индикации к микро-ЭВМ

Клавиатура компьютера представляет собой набор механических контактов (открытых или герконовых), контактов на основе токопроводящей резины, емкостных датчиков или датчиков на основе эффекта Холла. Состояние кнопки фиксируется в триггере, выход которого присоединяется к одной из линий шины данных микропроцессора. На рисунке 2.8,а показана схема подключения контакта к шине микропроцессора. Из схемы видно, что в отпущенном состоянии на входе D–триггера присутствует уровень логической 1 и триггер принимает состояние «1». При нажатии кнопки триггер принимает нулевое состоянии. Недостатком такой схемы является наличие эффекта «дребезга» контакта за счет того, что в момент нажатия ключа подвижный пружинистый контакт несколько раз отскакивает от неподвижного, прежде чем установиться в устойчивое состояние. В результате «дребезга» наблюдается расщепление единичного потенциала в момент смены состояния и возможно неверное фиксирование состояния кнопки. Дребезг контакта может быть устранен программно путем считывания состояния триггера с некоторой задержкой после замыкания контакта. На рисунке 2.8 б) показана схема бездребезгового формирования состояния кнопки клавиатуры.



а) б)

Рисунок 2.8 - Схемы подключения контакта клавиатуры к МП

Если клавиатура имеет N контактов, то для фиксации ее состояния требуется N триггеров или N/8 регистров. Организация клавиатуры, при которой на каждый контакт требуется свой входной триггер, носит название «линейного»

подключения. Более экономной, с точки зрения затрат клавиатурных регистров, является матричная организация клавиатуры, которая применяется практически во всех типах клавиатур.

При этом контакты клавиатуры располагаются в узлах матрицы размером n×m, где n,m – соответственно количество строк и столбцов. Схема подключения клавиатуры в виде матрицы показана на рисунке 2.9.

В качестве устройств вывода Увыв и ввода Увв используются регистры, выполненные на D–триггерах. Если выходные шины Увыв имеют нулевой потенциал, то схема подключения контакта кнопки ничем не отличается от рисунка 2.6а. При разомкнутых контактах клавиатуры триггеры регистра Увв находятся в состоянии «1». При замыкании одного из контактов он соединяет вход одного из соответствующего триггера УВв с нулевым потенциалом и тот переходит в нулевое состояние.

Во всех случаях при организации ввода информации с клавиатуры в ЭВМ решается ряд задач к которым можно отнести:

1) определение факта нажатия клавиши на клавиатуре;

2) определение номера нажатой клавиши;

3) осуществляется передачу управления на соответствующую программу.

Определение факта нажатия клавиши может быть осуществлено с помощью последовательных операций.

1. Записать нули в разрядные ячейки выходного устройства.
2. Считать содержание разрядов входного устройства.
3. Повторять слова, если во всех разрядах УВВ записаны единицы.

Вариант программы определения факта нажатия на одну из клавиш имеет вид:

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	3E F8	M1	MVI A, 11111000	Занести 0 в младшие три разряда аккумулятора
0802	D3 KBDOT		OUT KBDOT	Записать 0 в выходное устройство с адресом KBDOT
0804	DB KBDIN		IN KBDIN	Получить число со входного устройства с адресом KBDIN
0806	E6 07		ANI 00000111	Очистить старшие пять разрядов аккумулятора
0808	FE 07		CPI 00000111	Проверить, есть ли в младших трех разрядах аккумулятора 0
080A	CA 0008		JZ M1	Если нет, то идти на M1
080B	C3 0D08	DONE	JMP DONE	Конец

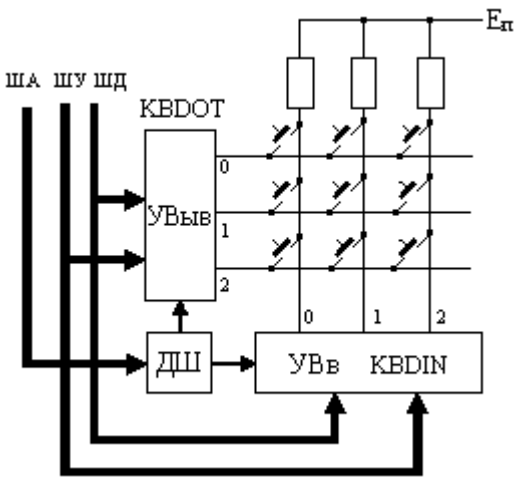


Рисунок 2.9 – Схема матричного подключения клавиатуры к микропроцессорной системе

Для определения номера нажатой клавиши необходимо нулевой потенциал подать только на одну из горизонтальных линий матрицы (путем записи единицы в соответствующий триггер Увыв), а на остальных должна быть «1» и определять ячейку регистра УВв, которая приняла состояние «1». Затем по очереди перемещать «0» в Увыв (сканировать УВыв) и определять разряд Увв с нулевым значением.

Вывод данных на восьмисегментный дисплей.

Восьмисегментный дисплей представляет собой микросхему с 8-мю светодиодами, выполненными в виде прямоугольных полосок (сегментов) и расположенных в пространстве в виде цифры 8 с точкой. Сегменты диодов могут в качестве общего электрода иметь либо анод или катод. В первом случае для высвечивания символа на электроды индикатора следует подавать нулевые потенциалы, а во втором – положительные.

Вывод на восьмисегментные дисплеи в микропроцессорных системах может осуществляться статическим или динамическим способом.

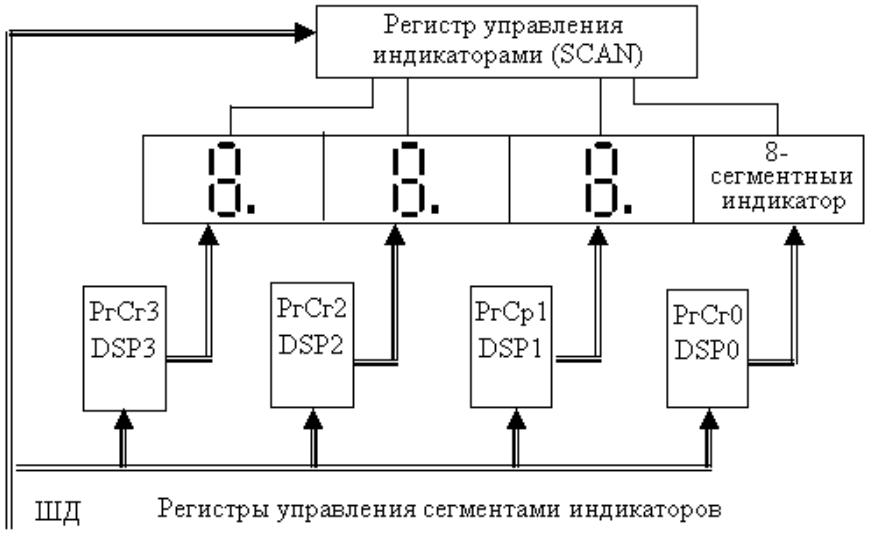


Рисунок 2.10 - Схема статической индикации данных

При статическом способе выводы сегментов каждого из индикаторов подключаются к своему регистру. Для управления разрешением высвечивания сим-

вола на индикаторе используется отдельный регистр, причем общий электрод каждого из индикаторов подсоединяется к соответствующему выходу этого регистра (рисунок 2.10). Программа управления выводом информации на дисплей состоит из операции выдачи кода символа на соответствующий индикатор (регистр DSP) и вывода разрешающего сигнала на этот индикатор (Регистр SCAN).

Одноименные сегменты каждой ячейки индикатора связаны общим проводом и соединены с соответствующим разрядом регистра сегментов РгСг. Выходы анодов каждого из индикаторов подключены к регистру сканирования РгСк.Наличие уровня логической единицы в соответствующем разряде регистра сканирования *РгСк* приводит к высвечиванию символа в соответствующем индикаторе дисплея при наличии информации на шине данных. Вариант программы включения сегментов второй ячейки с помощью кода, задаваемого со входного регистра (порта ввода) имеет вид:

Адрес	Машинный код	Метка	Мнемокод	Комментарий
0800	3E04		MVI A,04	Поместить в Акк число 00000100
2	D328		OUT SCAN	Вывести число на РгСк и включить цифру 2
4	DB20	M1	IN 20	Ввести данные в Акк из входного регистра
6	D338		OUT DSP2	Записать их в регистр сегментов РгСг дисплея
8	C30408		JMP M1	Продолжить с метки M1

1. Программа размещается в ОЗУ, начиная с ячейки с адресом 0800;
2. РгСг присвоено имя SCAN (адрес 28);
3. Входной регистр имеет адрес 20;
4. РгСг2 присвоено имя DSP2 (адрес 38).

Организация динамического режима работы дисплея. Схема подключения дисплея в динамическом (мультиплексном) режиме показана на рисунке 2.11. В этом режиме вывод информации на каждый индикатор дисплея выводится микро-ЭВМ последовательно. Сначала в RGDSP выводится код отображаемого символа, а в регистр RGSCAN – разрешающий потенциал для высвечивания только одного индикатора (например, 5-го).

Цифра или символ на индикаторе высвечивается некоторый промежуток времени, задаваемый подпрограммой задержки. Затем индикатор гасится, выставляется в RGDSP код, который должен быть отображен 4-м индикатором, и подается управляющий сигнал в RGSCAN, разрешающий светиться только этому индикатору. Ниже приведен вариант программы обеспечивающей мультиплексный режим работы дисплея.

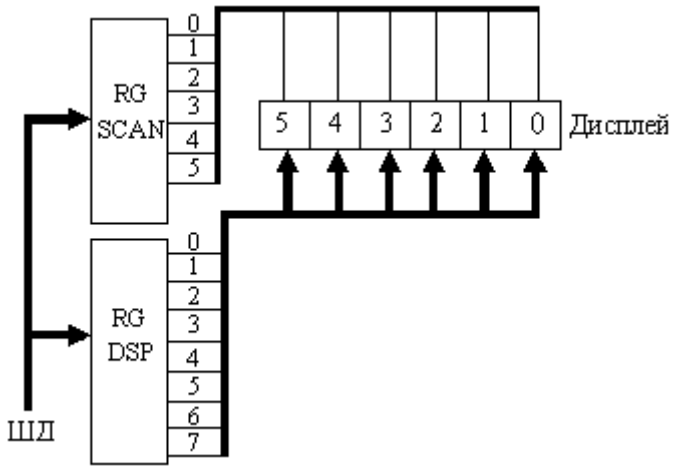


Рисунок 2.11 — Схема подключения индикаторов к микропроцессорной системе

Цифра или символ на индикаторе высвечивается некоторый промежуток времени, задаваемый подпрограммой задержки. Затем индикатор гасится, выставляется в RGDSP код, который должен быт отображен 4-м индикатором, и подается управляющий сигнал в RGSCAN, разрешающий светиться только этому индикатору. Ниже приведен вариант программы обеспечивающей мультиплексный режим работы дисплея.

Код цифр для вывода на каждую ячейку хранится в последовательных ячейках памяти с адресами 0900 – 0905. Код цифры для нулевой ячейки индикатора хранится в ячейке с адресом 0900. Начальный адрес подпрограммы временной задержки 0430.

Адрес	Метка	Мнемокод	Комментарий
0800		LXI B, 0100	Загрузить в регистр B, C длительность задержки
03		XRA A	Очистить аккумулятор
04	M1	LXI H, 0905	Указать на адрес кода цифры 5
07		MVI D, 20	Загрузить указатель цифры в регистр D
09	M2	MOV A,M	Получить из ОЗУ код очередной цифры
0A		OUT DSP	Записать его в РгСг
0C		MOV A,D	Загрузить в аккумулятор указатель цифры
0D		OUT SKAN	Включить нужную цифру
0F		RAR	Указать на следующую цифру
10		MOV D, A	Сохранить указатель цифры в регистре D
11		CALL DE2B	Вызвать подпрограмму временной задержки
14		XRA A	Очистить аккумулятор
15		OUT SKAN	Выключить цифры
17		DCR L	Уменьшить на 1 содержимое регистра L
18		ORA D	Все ли сообщения выведены?
19		JNZ M2	Если нет, то продолжить
1C		JMP M1	Если да, то начать сначала

3. Запоминающие устройства информационных систем

3.1. Иерархическая организация памяти компьютера

Память – важнейший ресурс информационно-вычислительной системы, требующий эффективного управления. Несмотря на то, что в наши дни память среднего домашнего компьютера в тысячи раз превышает память больших ЭВМ 70-х годов, программы увеличиваются в размере быстрее, чем память.

Системы памяти характеризуются двумя важными параметрами:

- время запаздывания (время отклика) - это полное время, прошедшее с момента завершения запроса или команды компьютерной системе до начала ответа
- пропускная способность – количество элементов данных, которые могут быть отправлены процессору за единицу времени.

При реализации систем памяти современных компьютеров проектировщики стремятся обеспечить ее максимальную производительность и емкость при разумной цене. Достичь таких показателей возможно на настоящее время только при иерархической многоуровневой структуры памяти (рисунок 3.1).

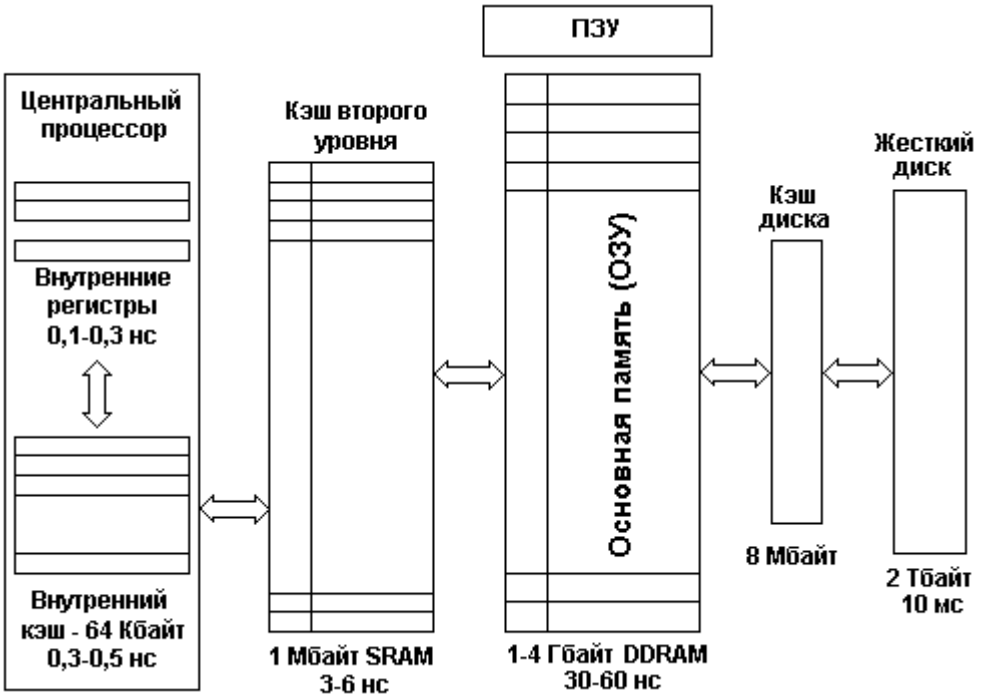


Рисунок 3.1 – Иерархия памяти компьютера

Память, ближайшая к процессору, делается относительно небольшой и быстрой, которая характеризуется высокой стоимостью в расчете на один бит. Этот уровень получил название "кэш-памяти". Реальная память, называемая также основной или оперативной памятью, имеет большой объем, работает медленнее и характеризуется меньшей стоимостью бита данных. Самый низкий уровень в иерархии памяти обычно представлен накопителем на жестком магнитном диске, который отличается самым большим временем отклика и самой

низкой пропускной способностью. Однако эта память может быть очена велика и дешева в расчете на бит. Обратите внимание на представленную иерархию памяти. При перемещении слева направо происходит следующее:

- снижается стоимость бита;
- возрастает емкость;
- возрастает время доступа и снижается частота обращений процессора к памяти.

3.2. Основные характеристики полупроводниковых ЗУ

Основная (внутренняя) память компьютера состоит из ЗУ двух видов: оперативного (ОЗУ) и постоянного (ПЗУ). ОЗУ предназначено для хранения переменной информации и допускает изменение своего содержимого в ходе выполнения процессором вычислительных операций. При выключении питания содержимое оперативной памяти теряется.

ПЗУ содержит информацию, которая не должна изменяться в процессе всего времени существования компьютера. Такую информацию составляют стандартные программы тестирования компьютера, драйверы устройств, значения физических констант и пр. Эта информация заносится в ПЗУ заранее, в процессе изготовления компьютера. ПЗУ является энергонезависимым устройством. Т.е. при выключении питания компьютера информация в ПЗУ не разрушается.

В современных компьютерах основная память строится преимущественно на полупроводниковых приборах. Полупроводниковая память имеет большое число характеристик и параметров. Важнейшими из них являются:

1) Емкость памяти – число бит хранимой информации (1024 бит, 4 Кбит). Важной характеристикой является информационная *организация* кристалла памяти размером M×N, где M – число слов, N – разрядность слова. Например, кристалл емкостью 16 Кбит может иметь различную организацию: 16 К × 1; 4 К ×4 или 2 К × 8.

2) Время обращения к памяти – время от подачи сигнала на запись или считывание до того момента, когда закончатся все действия, связанные с выполняемой операцией и устройство готово реализовать следующую операцию. Длительность времени обращения называется длительностью цикла.

По способу доступа память различается на:

- ЗУ с произвольной выборкой (ЗУПВ);
- ЗУ с последовательным доступом;
- стековую (первый записан - последний считан);
- ассоциативную.

Элементы полупроводниковых ЗУ изготавливаются в виде больших интегральных схем (БИС). Типовая БИС ЗУ с произвольной выборкой с организацией 1024×4 имеет следующие выводы (рисунок 3.2).

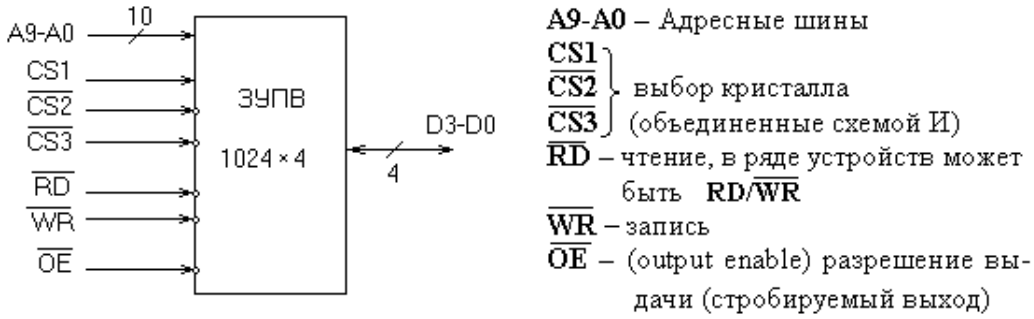


Рисунок 3.2 – Типовая схема БИС ЗУ с произвольной выборкой

Операция считывания начинается с установления на ША сигналов адреса требуемой ячейки памяти; после этого подается сигнал выбора кристалла CS. Через некоторое время содержимое ячейки выдается на шину данных D3-D0. При операции записи на ША устанавливается адрес требуемой ячейки, а затем (или одновременно) на ШД выставляется записываемое слово. После этого подаются импульсы выбора микросхемы CS и записи WR.

3.3. Статические ОЗУ с произвольным доступом

В статическом ОЗУ информация в ячейках памяти хранится до тех пор, пока подается напряжение питания. Типовым представителем статического ОЗУ с произвольным доступом (RAM – *random access memory*) являются микросхемы памяти серии КР537. В качестве примера рассмотрим структуру БИС КР537ЗУ3, условное обозначение которой изображено на рис.3.3,а.

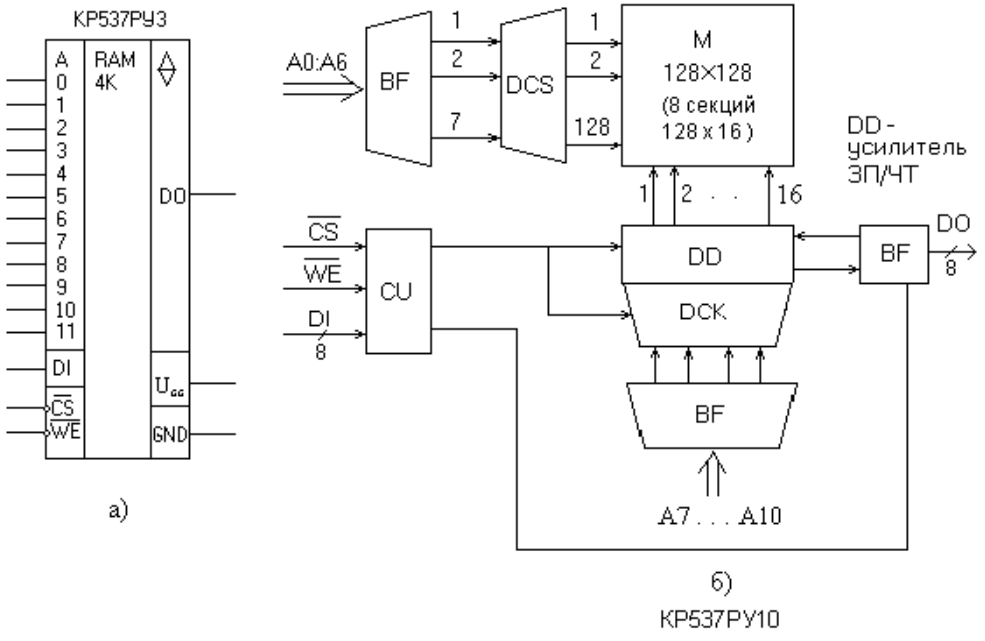


Рисунок 3.3 – Условное обозначение и структура БИС ОЗУ с произвольным доступом

Как видно из схемы ОЗУ, в нем содержится 2^{12} = 4К однобитовых ячеек памяти. Выходная линия данных может быть переведена в состояние высокого импеданса (значок ◊) путем подачи высокого уровня на вход CS. Запись данных в ячейку памяти осуществляется при наличии нулевого уровня на входе WE (*Write Enable*).

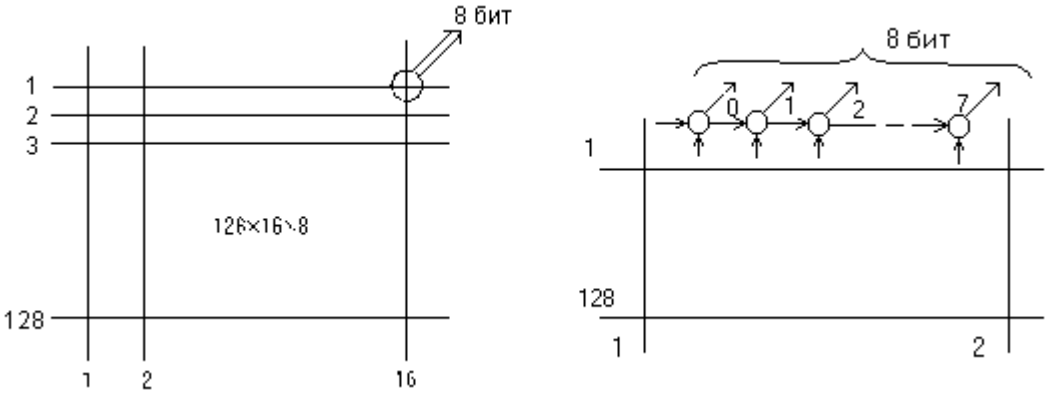


Рисунок 3.4 - Схема организации матрицы запоминающих ячеек

ОЗУ серии К537 выполнены на основе КМОП – технологии. В состав серии входят БИС: К537РУ2…РУ10. БИС РУ3 совместима с ТТЛ-схемами как по входам, так и по выходам. Особенности ОЗУ является способность сохранять информацию при пониженном напряжении источника питания.

Микросхемы памяти КР537РУ8 и РУ10 имеют организацию 2048×8 бит. Структурная схема ОЗУ с организацией 2048×8 бит показана на рисунке 4.3. Запоминающие ячейки ОЗУ организованы в виде матрицы размером 128 × 128 ячеек, состоящей из 128 строк и 16 столбцов, выбирающих по 8 триггеров. Структура матрицы запоминающих элементов показана на рисунке 5.4.

В состав БИС входят также адресные буферы BF строк и столбцов и соответствующие им дешифраторы DCS и DCK, усилители записи/чтения DD и блок управления CU. Схема однобитовой ячейки памяти статического ЗУ изображена на рисунке 3.5.

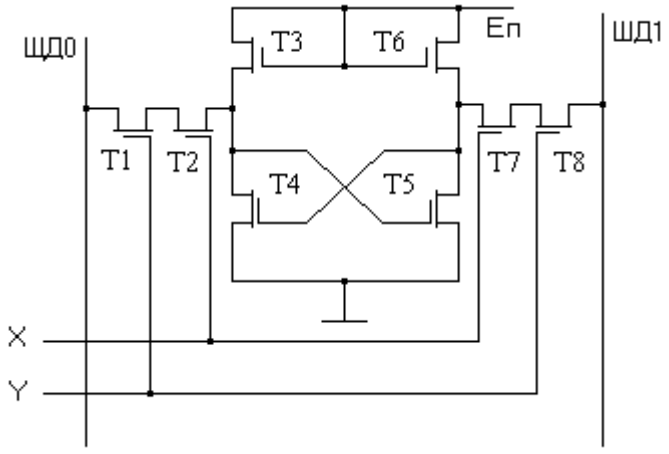


Рисунок 3.5 - Схема ячейки памяти статического ОЗУ

Она представляет собой статический триггер, состоящий из четырех МДП-транзисторов Т3-Т5, плечи которого через ключевые транзисторы Т1,Т2 и Т7,Т8 подключены к шинам данных ШД0 и ШД1. Подключение ячейки к шине данных происходит только при одновременной подаче единичных сигналов на линии выбора ячейки Х и У. Данные записываются в ячейку памяти и считываются из нее в парафазном коде (наличие уровней «0» и «1» одновременно). Работа ячейки подробно рассматривалась в курсе электроники.

3.4. ОЗУ динамического типа

В запоминающих устройствах динамического типа информация хранится в виде заряда на конденсаторе. Поэтому питание на ОЗУ подается не постоянно, а только в очень короткие промежутки времени. Оно используется для восстановления заряда на конденсаторах матрицы ОЗУ. Благодаря импульсному питанию динамические ОЗУ потребляют в тысячи раз меньше мощности, чем аналогичные по емкости статические.

В микросхемах динамической памяти функции запоминающих элементов выполняют электрические конденсаторы, образованные внутри МДП-структуры. Поскольку время сохранения заряда на конденсаторе ограничено, необходимо предусмотреть восстановление (регенерацию) записанной информации. Период регенерации для динамических ОЗУ равен нескольким миллисекундам (для микросхем серии К565 время регенерации 2 мс).

Микросхемы (МС) большинства динамических ОЗУ с целью уменьшения количества выводов построены с мультиплексированием кода адреса: вначале в МС вводят код строки А0 – А7, фиксируя его во входном регистре стробирующим сигналом RAS (*Row Address Strobe*), а затем код адреса столбца А8 – А13, фиксируя его во внутреннем регистре стробирующим сигналом CAS (*Column Address Strobe*).

В режиме регенерации микросхема ОЗУ изолируется от информационных входа и выхода за счет подачи сигнала CAS = 1. Следовательно, адресуются только строки, т.к. регенерация информации происходит во всех элементах памяти строки одновременно.

Перебирая адреса строк, устройство регенерации обеспечивает восстановление информации во всей матрице накопителя. Условное обозначение БИС динамического ОЗУ типа К565РУ5 и временная диаграмма функционирования показаны на рисунке 3.6.

Схема динамической ячейки памяти на 8 транзисторах показана на рисунке 3.7. Она отличается от аналогичной ячейки статического ОЗУ только тем, что затворы транзисторов Т3 и Т6 соединены с генератором импульсов регенерации, а не с источником питания. Для хранения информации используются не специально встроенные конденсаторы, а конденсаторы, образованные паразитными емкостями затвор-сток и исток-сток транзисторов Т4 и Т5.

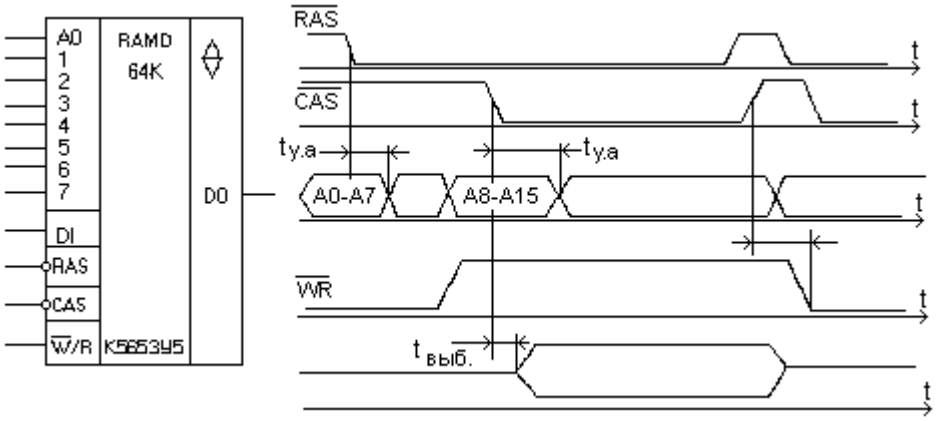


Рисунок 3.6 – Условное обозначение и временная диаграмма функционирования динамического ОЗУ

Такая ячейка памяти использовалась только в первых образцах динамических ОЗУ. Очевидно, что для хранения одного бита достаточно одного конденсатора. Поэтому впоследствии схема ячейки была существенно упрощена и в настоящее время имеет вид, изображенный на рисунке 3.8. При записи на вертикальную линию подается напряжение уровня данных, а на горизонтальную – импульс выбора строки.

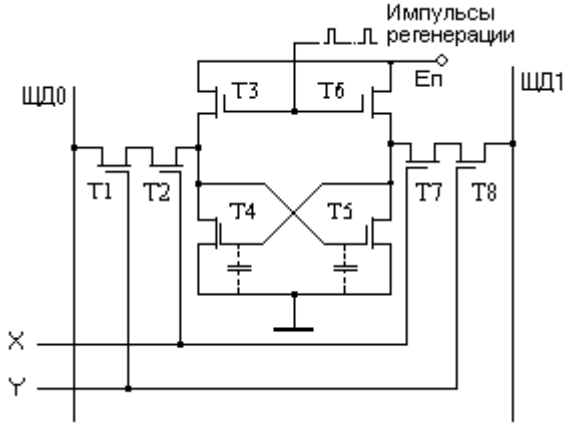


Рисунок 3.7 - Схема ячейки памяти динамического ОЗУ

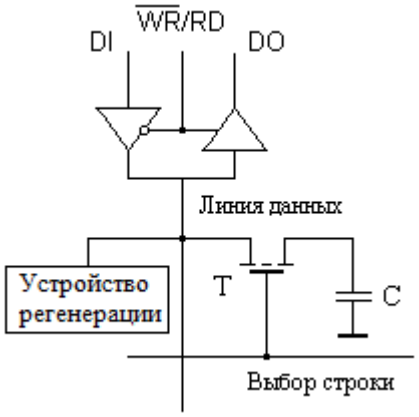


Рисунок 3.8 – Схема однотранзисторной ячейки динамического ОЗУ

За счет уменьшения количества транзисторов на одну ячейку удалось существенно увеличить емкость динамической памяти, располагаемой на одном кристалле и снизить потребление энергии от источника питания.

3.5. Постоянные запоминающие устройства

Постоянные запоминающие устройства (ПЗУ) являются энергонезависимыми устройствами, служащими для хранения цифровых данных. ПЗУ могут быть построены на пассивных элементах (плавких перемычках П или диодах D)

или активных (транзисторах). Схема ПЗУ представляет собой матрицу (рисунок 3.9) количество горизонтальных линий равно разрядности хранимого слова, а число вертикальных – количеству хранимых слов.

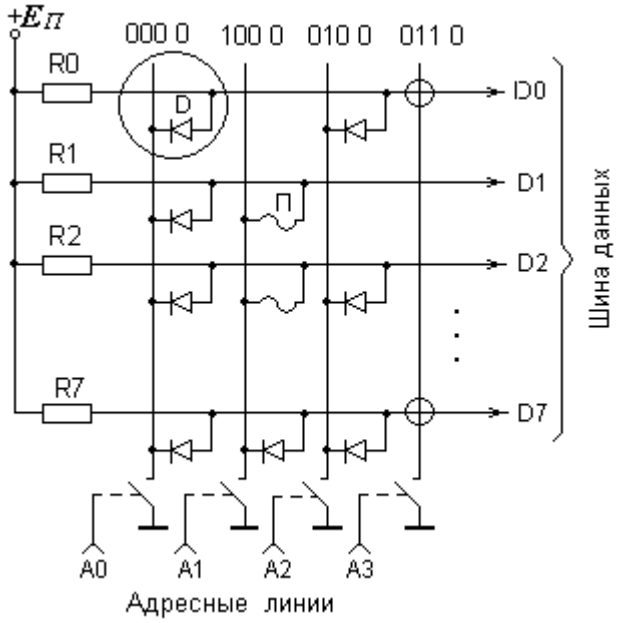


Рисунок 3.9 – Схема матричного ПЗУ

Как видно из схемы, при активации адресной линии вертикальная шина соединяется с сигнальной землей и диоды, подключенные к этой шине, шунтируют линии данных на "землю". Таким образом, если горизонтальная линии данных соединена с вертикальной через диод (или перемычку), то при выборе адресной линии, на выходе линии данных будет потенциал близкий к нулю, т. е. логический 0. Если диод или перемычка отсутствуют в данном узле, то на соответствующем выходе линии данных присутствует высокий потенциал, близкий к $E_П$, т.е. логическая 1.

Обычно такие ПЗУ изготавливаются со всеми диодами (или плавкими перемычками) в узлах матрицы. В тех узлах, в которых диод или перемычка должны отсутствовать, их убирают путем выжигания. Эта процедура выполняется в процессе программирования ПЗУ и называется "прожиганием ПЗУ"

Запись информации в ПЗУ осуществляется пословно (побайтно). Для занесения информации в ячейку ПЗУ необходимо на линии данных, в которых должна быть "1", подать высокий потенциал (≈ 25 В) и выбрать соответствующую адресную линию, т.е. соединить ее с сигнальной землей. Протекающий ток расплавляет диод или плавкую перемычку, исключая тем самым шунтирующую цепь соответствующей линии данных.

Недостаток рассмотренной схемы ПЗУ состоит в том, что после занесения информации в это устройство ее нельзя изменить. То есть, при изменении программы, подлежащей хранению в ПЗУ, необходимо запрограммировать новое устройство. Для устранения этого недостатка разработаны полупостоянные электрически перепрограммируемые постоянные запоминающие устройства (ЭППЗУ). Схема ЭППЗУ подобна ПЗУ на основе МОП транзисторов, однако транзисторы в таком устройстве имеют "плавающий" затвор, который электрически изолирован оксидным слоем полупроводникового материала. Схема ЭППЗУ изображена на рисунке 5.10. При подаче на "плавающий" затвор (ПЗ) положительного потенциала по отношению к стоку транзистора на ПЗ индуцируется электрический заряд, который за счет высококачественной изоляции может сохраняться до 10 лет и более. Благодаря этому заряду транзистор находится в открытом состоянии, при котором сопротивление Сток-Исток становится близким к нулю.

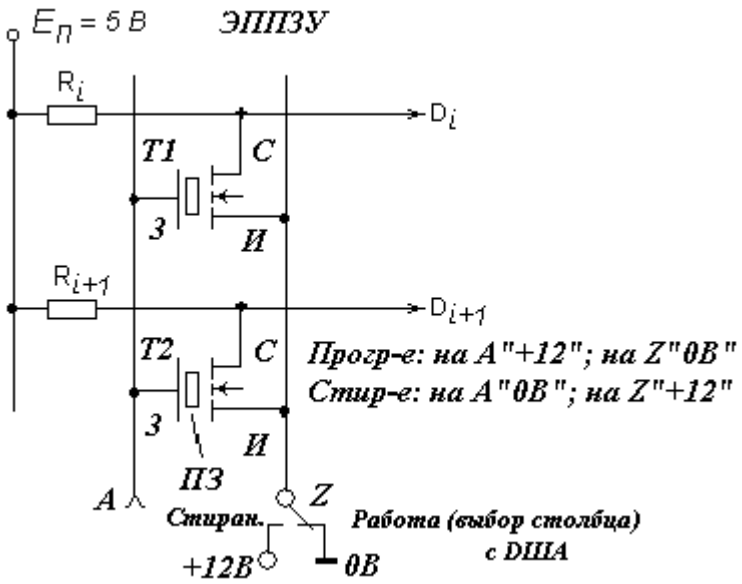


Рисунок 3.10 – Схема электрически перепрограммируемого запоминающего устройства

За счет шунтирования соответствующей линии данных на ее выходе устанавливается потенциал логического нуля. При отсутствии заряда на ПЗ транзистор остается в закрытом состоянии и на выходе данных имеется потенциал. близкий к E_n (уровень логической 1).

Полярность и значение напряжений при записи и стирании информации в ячейки ЭППЗУ показаны на схеме.

3.6. Кэш-память и ее организация

Кэш-память — это быстродействующая память, располагаемая между МП и основной памятью. Она предназначена для повышения производительности компьютеров. Быстродействующая память обычно является дорогостоящей, поэтому повышать быстродействие всей основной памяти большой емкости не выгодно. В связи, с чем предлагается реализовать память небольшой емкости с предельным быстродействием — кэш-памяти. В кэш-память помещают текущие программы и данные, хранящиеся обычно в основной памяти. При выполнении программы процессор сначала передает адресный сигнал в кэш-память, с помощью которого можно осуществить запись–считывание. Наличие этого адреса в кэш-памяти называется совпадением КЭШа, а отсутствие — несовпадением.

Кэш-память представляет собой быстродействующую память, в которую из основной памяти с умеренной скоростью переносятся команды и данные (выбранные участки ОЗУ копируются в кэш). Структурная схема системы подключения кэш-памяти в ЭВМ изображена на рисунке 3.11.

Кэш-память была предложена, когда появились процессоры, работающие значительно быстрее ОЗУ. Они вынуждены были ожидать, пока из памяти будут введены требуемые элементы данных. В системах с кэшированием информация пересылается из вторичной памяти (с жесткого диска) в основную память, а затем в кэш-память. Чтобы данные попали в кэш, нужно выполнить следующие действия:

В начале работы программа, обращаясь к своему адресному пространству, передает устройству управления памятью или микропрограмме *виртуальный адрес.* Если страница или сегмент, содержащие этот адрес, находятся в основной памяти, виртуальный адрес преобразуется в *адрес основной памяти*. В противном случае возникает *отказ страницы*, и страница или сегмент загружается в основную память.

После этого аппаратно проверяется, не находится ли требуемый блок уже в кэш-памяти. Если это так, то содержимое адресуемой ячейки извлекается из кэша (либо результат вычисления заносится в ячейку). В противном случае формируется отказ кэш-блока (аналогичный отказу страницы) и информация копируется из основной памяти в КЭШ. После загрузки блока в кэш-память команда выполняется повторно. Схема загрузки из основной в кэш-память управляется внешней (по отношению к ЦП) схемой – кэш-контроллером и без участия ОС.

Следует заметить, что использование кэш-памяти является возможным только потому, что команды и данные программ в большинстве случаев расположены в близлежащих ячейках памяти (локальность программы).

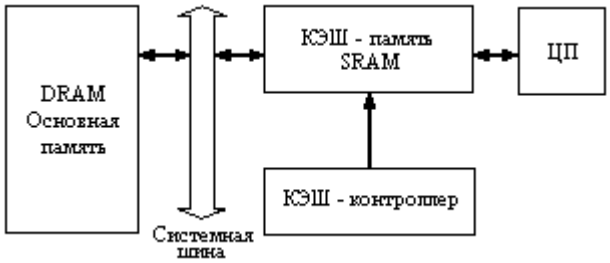


Рисунок 3.11 – Структурная схема подключения кэш-памяти в ЭВМ

В последующих шагах, когда процессору требуется следующая команда или следующий элемент данных, он сначала обращается к кэш-памяти. Доступ, при котором данные обнаруживаются в кэш-памяти, называется *кэш-попаданием*. В противном случае *фиксируется кэш-промах*. Доля обращений, при котором имелось кэш-попадание, называется *коэффициентом кэш-попаданий.* На величину этого коэффициента влияют размеры и организация КЭШа, алгоритмы поиска и характеристики выполняемой программы. Кэш-память снабжается контроллером, который управляет ее ресурсами и обеспечивает максимальную эффективность доступа. Контроллер разбивает основную память на блоки с типичным размером 2, 4, 8 или 16 байт. 32 – разрядный процессор обычно использует блоки размером 2 или 4 слова. Когда фиксируется кэш-промах, кэш-контроллер перемещает в кэш-память из основной памяти требуемое слово с целым блоком, содержащим это слово. Обращение к любому байту блока приводит к копированию в кэш всего блока.

Существуют три способа организации кэша: полностью ассоциативный, с прямым отображением, частично ассоциативный.

Полностью ассоциативная кэш-память. В этом случае в кэше содержатся наиболее часто используемые процессором блоки данных и их полные адреса. Когда процессор запрашивает данные, кэш-контроллер сравнивает адрес требуемых данных с каждым адресом в кэш-памяти. Для ускорения поиска используют схемы с параллельным поиском, которые содержат компараторы и контроллеры.

Кэш-память с прямым отображением. Каждый блок из основной памяти может занимать *одно единственное место* в кэше. Это уменьшает число шагов поиска и сравнений, необходимых для фиксации попаданий и промахов. Основной недостаток – *конфликтование* в кэш-памяти ячеек ОЗУ, которые имеют одно и тоже место хранения в кэше. В этом случае каждый раз нужно будет осуществлять попеременно подкачку этих ячеек. Однако такие случаи встречаются довольно редко и эта кэш-память обладает приемлемой производительностью при невысокой стоимости.

Частично ассоциативная кэш-память. Этот способ кэширования является промежуточным между рассмотренными выше.

Частично ассоциативная кэш-память характеризуется наличием нескольких наборов блоков с прямым отображением, которые функционируют параллельно. Когда из основной памяти поступает блок данных, он может быть записан на соответствующее этому блоку место любого набора. При выборе нового блока кэш-контроллер должен решить, на место которого блока в кэш-памяти его записать. При этом может быть реализован:

- случайный выбор;
- запись в последовательном порядке;
- на место блока, который не адресовался дольше других.

При использовании КЭШ-памяти имеется две копии данных – одна в КЭШ, а другая – в основной памяти. Одна из них может быть модифицирована, а другая нет, т.е. возникает два набора данных с одним адресом памяти.

Наличие двух копий требует механизма обновления, который предотвращал бы использование устаревших данных. Разработано несколько способов записи «новых» данных:

- **сквозная запись** – контроллер сразу же после записи данных в кэш переносит их в основную память (потеря производительности);
- **буферизированная сквозная запись** – запросы на запись в основную память буферизируются. Это позволяет процессору, не дожидаться окончания записи в основную память, приступить к новому циклу;
- **обратная запись** – имеется поле признака каждого блока в кэш-памяти, которое включает бит, показывающий, был ли изменен блок. В основную память переносится только измененный блок. Этот способ обеспечивает максимальную скорость обмена.

4. Архитектура 16-разрядных процессоров

4.1. Линейная и сегментная адресация

В 16-разрядных МП длина слова равна 16 бит, адресуемое пространство памяти составляет 1-16 Мбайт, оперативный блок состоит из 16 регистров общего назначения. Шина данных является 16-разрядной, используется расширенный набор команд, включающий команды умножения, деления и др, отсутствующие в 8-разрядных МП. Адреса обычно представляют собой 20-24-разрядные слова. Среди устройств этого класса имеются процессоры с раздельными шинами данных и адресов Motorola (MC68000) с большим количеством выводов (64) и процессоры с совмещенными шинами данных и адресов фирмы Intel и Zilog (8086 и Z8001), с меньшим количеством выводов (40 и 48 соответственно).

Если число разрядов адреса превышает длину информационного слова, возникают вопросы: в какой форме поместить такой адрес в регистр и каким способом осуществлять вычисление адреса? Для решения этих задач используют два способа адресации: *линейную* и *сегментную.*

При *линейной адресации* адрес представляет собой отдельное целочисленное значение. Вычисление адреса осуществляется с помощью операций сложения, приращения 16-ти разрядов слова двойной длины .В 16-разрядных МП выполняются аналогичные операции, т.е. адреса хранятся в 32-разрядных регистрах двойной длины. Архитектура такого процессора довольно сложная, но зато удобная для использования, т.к. все адресное пространство используется, как единое целое.

При *сегментной адресации* все пространство адресов делится на множество сегментов, т.е. пространство является сегментированным. Начальный адрес сегмента называют базовым. Порядок разбиения на сегменты может быть произвольным и после того, как он установлен, адрес можно представить с помощью номера (базового адреса) сегмента и смещения. Т.е. можно использовать два 16-разрядных регистра. Следовательно, максимальный размер одного сегмента составит 64Кбайт, а вычисления адреса сводится только к вычислению смещения. Для вычисления смещения можно воспользоваться тем же 16-разрядным АЛУ, что значительно упрощает структуру МП.

В МП MC68000 используется линейная адресация. 24-разрядный код адреса позволяет обращаться к 16 Мбайтам адресного пространства. Адреса хранятся в 32-разрядных регистрах. Операционный блок, кроме 16-го АЛУ, содержит два 16-разрядных сумматора, с помощью которых одновременно осуществляется вычисления 32-разрядных адресов. В МП 8086 и Z8001 применяется сегментная адресация. В 8086 используется базовый адрес сегмента и смещения, а в Z8001 – номер сегмента (7 старших бит) и 16 разрядов – смещение.

Память в микро-ЭВМ на базе МП Intel 8086 (отечественный аналог КР1810ВМ86) логически организована как одномерный массив *байтов*, каждый из которых имеет адрес в диапазоне *0000 – FFFFF*. Любые два смежных байта могут рассматриваться как 16-битовое слово. Младший байт имеет

меньший адрес, старший – больший. *Адресом слова считается адрес его младшего байта.*

Полная информация, необходимая для определения его физического адреса содержится в адресном объекте *"сегмент-смещение*", который называется указателем адреса и содержит адрес сегмента и внутрисегментное смещение. Для запоминания указателя адреса требуется *два слова памяти*, причем слово с *меньшим адресом* всегда содержит *смещение*, а слово с большим адресом – *базовый адрес сегмента*.

Команды, байты и слова данных можно свободно размещать по любому адресу, что позволяет экономить память, благодаря ее плотной упаковки. Однако, для экономии времени выполнения программ, целесообразно размещать слова данных в памяти по *четным адресам*, т.к. МП передает такие слова за один машинный цикл. Слово с четным адресом называется *выровненным на границе слов*. Слова с нечетными адресами (не выровненные) также допустимы, но для их передачи требуется два цикла шины, что снижает производительность МП. Следует отметить, что интерфейсный блок процессора (BIU) инициирует необходимое для выборки слова число обращений к памяти автоматически, так что двукратное обращение к памяти не требует специального указания в программе. Особенно важно иметь выровненные слова для операций со стеком, так как в них участвуют только слова. Следовательно, указатель адреса стека SP необходимо *всегда* инициализировать на четный адрес.

Команды всегда выбираются словами по четным адресам, за исключением первой выборки после передачи управления по нечетному адресу, когда выбирается один байт. Поток команд разделяется на байты при заполнении конвейера команд внутри МП. Т.о. выравнивание команд не влияет на производительность и потому не используется.

4.2. Сегментация памяти и вычисление адресов

Пространство памяти емкостью 1 Мбайт представляется как набор сегментов, определяемых программным путем. Сегмент состоит из смежных ячеек памяти и является независимой и отдельно адресуемой единицей памяти емкостью 64 Кбайт. Каждому сегменту программой назначаются начальный (базовый) адрес, являющийся адресом первого байта сегмента в пространстве памяти. Начальные адреса четырех сегментов, выбранных в качестве текущих, записываются в сегментные регистры CS, DS, SS, ES. Для обращения к команде и данным, находящимся в других сегментах, необходимо изменить содержимое сегментных регистров, что позволяет использовать все пространство памяти емкостью 1 Мбайт. Частный случай загрузки всех сегментных регистров нулями приводит к организации памяти характерной для МП580ВМ80, т.е фактически к отказу от сегментации памяти.

В сегментном регистре хранится 16 старших битов 20-разрядного начального адреса сегмента. Четыре младших бита принимаются равными нулю и дописываются справа к содержимому сегментного регистра при вычислении фи-

зических адресов ячеек памяти. Поэтому начальные адреса сегментов всегда кратны 16. Поскольку других ограничений на размещение сегментов в памяти нет, *сегменты могут быть соседними (смежными), неперекрывающимися, частично или полностью перекрывающимися*. Физическая ячейка памяти может принадлежать одному или нескольким сегментам.

Физический адрес сегмента памяти представляет 20-битовое число в диапазоне 00000-FFFFF, которое однозначно определяет положение каждого байта. *Логический адрес* ячейки памяти состоит из двух 16-битовых беззнаковых значений: *начального адреса сегмента*, который называют *базой или сегментом*, и *внутрисегментного смещения*, определяющего расстояние от начала сегмента до этой ячейки. Для вычисления физического адреса база сегмента сдвигается на 4 бита влево и суммируется со смещением. Перенос из старшего бита, который может возникнуть при суммировании, игнорируется (рисунок 4.1).

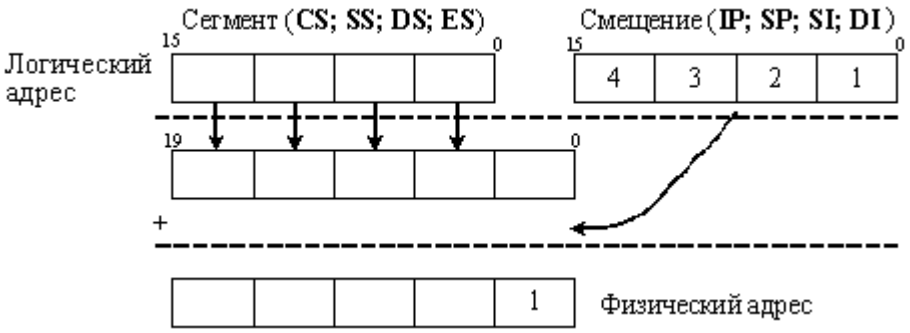


Рисунок 4.1 — Схема вычисления физического адреса

Это приводит к так называемой кольцевой организации памяти, при которой за ячейкой FFFFF следует ячейка с нулевым адресом. Аналогичную кольцевую организацию имеет и каждый сегмент.

Команды всегда выбираются из текущего сегмента кода в соответствии с логическим адресом CS:IP. Стековые команды всегда обращаются к текущему сегменту стека по адресу SS:SP. Если при вычислении эффективного адреса EA используется регистр BP, то обращение производится также к стековому сегменту, а ячейки стекового сегмента рассматриваются как ОЗУ с произвольной выборкой. Операнды, как правило, размещаются в текущем сегменте данных, и обращение к ним организуется по адресу DS:EA. Однако программист может заставить МП обратиться к переменной, находящейся в другом текущем сегменте.

По умолчанию предполагается, что цепочка-источник находится в текущем сегменте данных, а ее смещение задается регистром SI. Цепочка - *получатель* обязательно располагается в текущем дополнительном сегменте, а *смещение* берется из регистра DI. Команды обработки цепочки автоматически модифицируют содержимое индексных регистров SI и DI по мере продвижения по

цепочке в направлении, задаваемом флагом DF. Источники логического адреса для различных типов обращения к памяти приведены в табл. 4.1.

Таблица 4.1 – Источники логического адреса

Тип обращения к памяти	Сегмент (умолчание)	Вариант	Смещение
Выборка команд	CS	нет	IP
Стековые операции	SS	нет	SP
Переменная	DS	CS, SS, ES	EA
Цепочка-источник	DS	CS, SS, ES	SI
Цепочка-приемник	ES	нет	DI
как базовый регистр	ES	CS, SS, DS	EA

Смена сегментного регистра осуществляется с помощью однобайтового префикса замены сегмента 001SR110, который ставится первым байтом команды. Двухбитовое поле SR содержит код сегментного регистра, используемого для вычисления физического адреса в данной команде: 00 – регистр EA; 01 – CS; 10 – SS; 11 – DS.

Сегментная структура памяти обеспечивает возможность создания позиционно независимых или динамически перемещаемых программ, что необходимо в мультипрограммной среде для эффективного использования оперативной памяти. Чтобы обеспечить позиционную независимость все смещения в программе должны задаваться относительно фиксированных значений, содержащихся в сегментных регистрах. Это позволяет произвольно перемещать программу в адресном пространстве памяти, *изменяя только содержимое сегментных регистров*.

4.3. Архитектура 16-разрядного процессора первого поколения

Однокристальный 16-разрядный МП Intel 8086 (К1810ВМ86) имеет мультиплексную 20-разрядную ША и 16-разрядную ШД и рассчитан на работу как в одно, так и в многопроцессорных системах. Процессор 8086 на языке ассемблера совместим с МП 8080, регистры и систему команд которого можно рассматривать как подсистему регистров и команд МП 8086.

Эффективность работы МП существенно повышена за счет введения команд математических операций (включающих *умножение* и *деление*) над 8- и 16-разрядными числами, команд побитовой обработки чисел, команд *работы с массивами данных*, расширения видов прерываний работы МП, а также реализации *конвейерного* типа выполнения команд в самой БИС. МП может работать

с памятью объемом до 1 Мбайт, обмениваться информацией с 64 Кбайт внешних устройств, имеет 256 типов различных прерываний.

Упрощенная структурная схема БИС К1810ВМ86 приведена на рисунке 4.2. Блок сопряжения с шинами BIU производит все пересылки данных и кодов для EU. Пересылки между памятью или внешними устройствами осуществляется по требованию EU. В то время как EU занят выполнением команды, блок BIU получает последующие в программе коды из памяти. *Блок выполнения команд* EU имеет 16-разрядные АЛУ с регистром состояния и флажками управления, а также РОНы. Все регистры и внутренние магистрали блоки 16-разрядные. Блок не имеет связи с внешними шинами МП.

На АЛУ поступают коды команд из конвейера команд, расположенного в BIU. Если в результате дешифрации кода команд в АЛУ необходимо получение одного или нескольких операндов по внешним магистралям МП, то EU запрашивает BIU на получение и размещение необходимых данных в BIU.

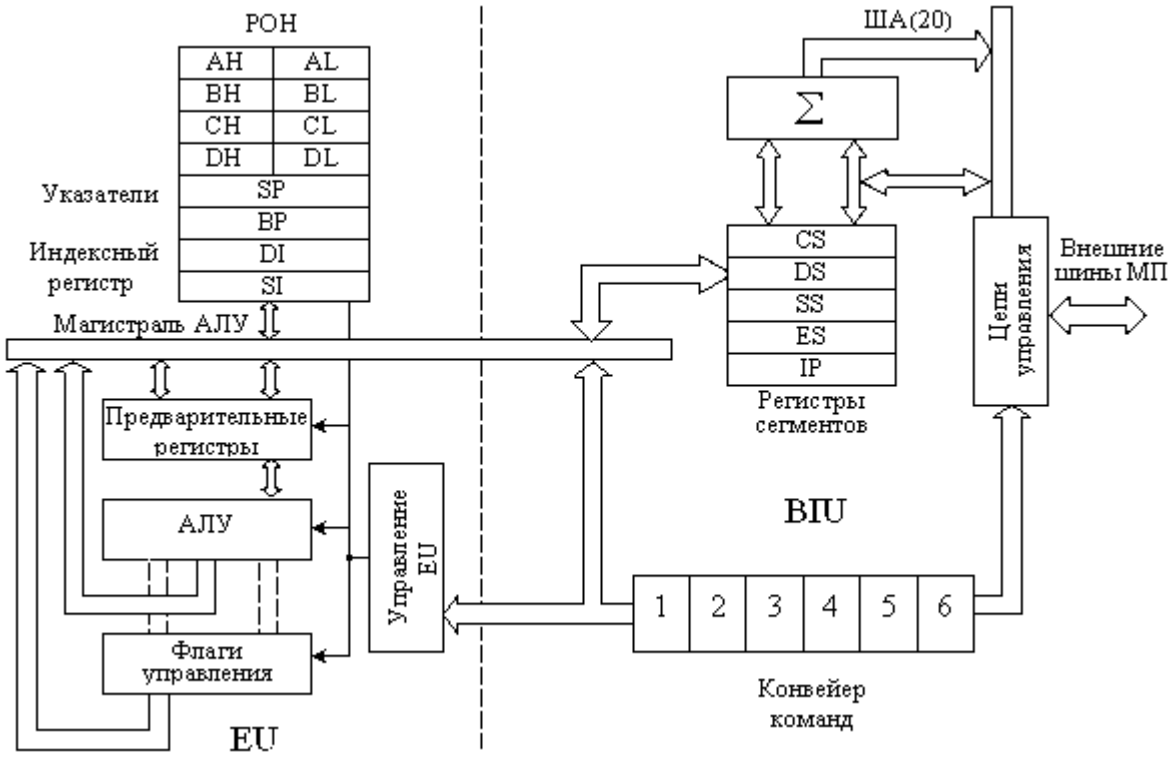


Рисунок 4.2 – Структурная схема 16-разрядного микропроцессора 8086

Несмотря на то, что все адреса, с которыми оперирует EU, 16-разрядные, BIU производит необходимые преобразования адресов так, чтобы EU имел возможность обращаться по всему возможному адресному пространству (1 Мбайт) МПС. BIU считывает команды с памяти и сохраняет их в конвейере команд, где может быть размещено до 6 инструкций. Это позволяет BIU выдавать их в EU по мере надобности без дополнительной загрузки внешней шины. BIU организует получение нового кода команды как только два байта из конвейера будут переданы в EU.

В большинстве случаев в BIU находится хотя бы одна команда и EU не простаивает, пока очередная команда будет извлечена из памяти. Коды подаются в EU последовательно, так как они записаны в программе. Если EU выполняет команду передачи управления в другое место программы, то BIU очищает конвейер команд, получает код из нового адреса, передает его в EU и начинает заполнять конвейер заново. Если EU требует обращения к внешнему устройству, то BIU приостанавливает процесс получения команд в конвейер и организует необходимый цикл обмена данными.

Любая ячейка памяти МП имеет два типа адресов: физический и логический. Физический адрес представляется 20-разрядным числом и однозначно определяет любую из 1 Мбайт ячеек памяти. В 16-разрядной системе адреса расположены в диапазоне от 0 до FFFFF. Весь обмен информацией МП с памятью осуществляется с использованием физических адресов. Программы же больше используют логические или физические адреса и позволяют записывать команду без предварительных знаний места, где эта команда будет размещена в памяти. Это дает программисту возможность маневрировать при распределении памяти, увеличивает гибкость программного обеспечения.

Логический адрес состоит из двух основных частей: значения *базы сегмента* и значения *смещения в сегменте*. Базовый адрес и смещение в сегменте отображаются 16-разрядными числами. Как только BIU обращается к памяти, базовый адрес формирует физический по принципу: Значение базы сегмента смещается на четыре разряда влево, и полученное 20-разрядное число (с четырьмя нулями в младших четырех разрядах) складываются со значением смещения в сегменте. Таким образом, база сегмента (с четырьмя нулями, добавленными в качестве младших разрядов) задает для памяти сегменты длинной 64 Кбайт, а значение сегмента в смещении – расстояние от начала сегмента до искомого адреса памяти. Максимально возможное смещение в сегменте равно 64 Кбайт. В любой момент времени программа может осуществлять доступ к одному из четырех сегментов:

сегменту текущего кода (*Current Code Segment* – CS);
сегменту текущих данных (*Current Date Segment* - DS);
сегмент текущего значения стека (*Current Stack Segment* – ES);
дополнительный сегмент текущих значений (*Current Extra Segment* – ES).

Каждый из этих сегментов может быть задан с помощью записи числа в соответствующий 16-разрядный регистр сегментов МП БИС. В большинстве случаев в МПК К1810 сегменты могут быть заданы в программе произвольно по усмотрению программиста.

В зависимости от команд, блок BIU получает информацию о логическом адресе памяти из различных регистров МП. *Коды команд* всегда извлекаются из адреса памяти, определяемого *содержимым регистра текущего кода* и регистра *указателя команд* IP (*Instruction Pointer*). В IP записано смещение в сегменте текущего кода. *Команды со стеком* всегда используют для адресации SS и регистр – *указатель стека* SP (*Stack Pointer*), где записано смещения в сегменте текущего значения стека. *Данные или переменные* в командах извлекаются из памяти, адреса которых расположены в *сегменте текущих данных*, хотя

по директивам команд это можно сделать и с помощью всех остальных сегментов. При вычислении физического адреса смещение в сегменте задается в EU, так как в нем происходит дешифрация кода команд и определение сегмента памяти, с которых будет работать БИС в текущий момент времени.

4.4. Регистры процессора Intel 8086

МП 8086 содержит три группы регистров. К *первой группе* относятся РОН, используемые для хранения промежуточных результатов. Ко *второй* группе относятся *указатели и индексные регистры*, предназначенные для размещения или извлечения данных из выбранного сегмента памяти. Содержание этих регистров определяет значение смещения в сегменте при задании логического адреса. К *третьей* группе относятся *регистры сегментов*, задающие начальные адреса (базы) самих сегментов памяти. МП имеет регистр флаговых разрядов, используемых для указания состояния АЛУ при выполнении различных команд и управления его работой.

Таким образом в МП располагается тринадцать 16-разрядных регистров и девять флаговых разрядов АЛУ. Последний, тринадцатый регистр, называется указателем команд *IP (Instruction Pointer*), выполняет функции, аналогичные функциям программного счетчика в МП КР580ИК80 и не является программно доступным регистром. Доступ к его содержимому может быть осуществлен с помощью команд передачи управления. КР580ИК80 и один добавленный 8-разрядный регистр для того, чтобы объединить все регистры в четыре 16-разрядные пары (рисунок 4.3).

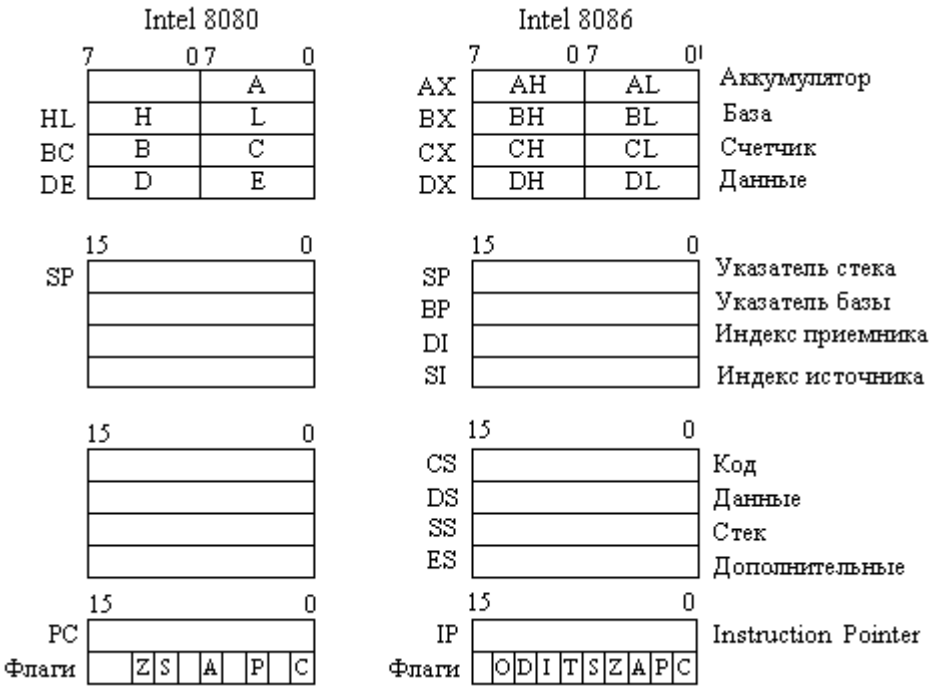


Рисунок 4.3 - Регистры процессоров 8080 и 8086

Группа РОН включает в себя семь 8-разрядных регистров МП К ним может быть организован программный доступ как к 8- или 16-разрядным регистрам. 16-разрядные регистры обозначаются AX, BX, CX и т.д. При обращении к ним, как к 8-разрядным регистрам, их обозначают AL, AH, BL, BH и т.д. Все эти регистры могут быть использованы при выполнении арифметических и логических команд. Однако, есть команды, использующие определенные регистры для специфических целей, тогда применяют мнемонические обозначения: аккумулятор (*accumulator*), база (*base*), счет (*count*), данные (*data*).

Группа указателей и индексных регистров состоит из четырех 16-разрядных регистров.

Регистры указатели: SP (Stack Pointer); BP (Base Pointer).

Индексные регистры: SI (Source index); DI (Destination index).

Обычно эти регистры содержат информацию о смещении по адресам в выбранном сегменте и позволяют компактно писать программы каждый раз непосредственно, не приводя используемого адреса. С их помощью производится вычисление адресов программ. Чаще всего в регистрах *указателях* записано адресное смещение по отношению к *стековому* сегменту, а в *индексных* регистрах – адресное смещение по отношению к *сегменту данных*.

Группа регистров сегментов состоит из четырех 16-разрядных регистров: CS; SS;DS и ES. Извлечение кодов команд производится из сегментов текущего кода с использованием адресного смещения, задаваемого указателем команд IP. Сегмент для извлечения операндов команды обычно указывается путем записи специального однобайтового префикса перед командой. Префикс имеет специальную кодировку, позволяющую МП отличить его от кода команды.

Регистр состояния (Флаговый регистр) содержит шестнадцать триггеров, из которых используется только 9. Эти триггеры отображают состояние процессора при выполнении последней арифметической или логической команды. Пять триггеров аналогичны МП 580-й серии. Дополнительные триггеры сигнализируют:

OVERFLOW – переполнение (при выполнении операций с числами МП имеет дополнительные флаги управления;

DIRECTION – направление – указывает направление работы с массивами (автоматическое увеличение или уменьшение на единицу адреса массива);

INTERRUPT – прерывание – определяет для МП возможность реагирования на прерывание;

TRAP – (западня, ловушка) устанавливает для МП пошаговый режим.

Кроме основных функций, соответствующих названию регистров, общие регистры выполняют специальные функции, указанные в таблице 4.2.

Для задания режима работы используется вывод $\overline{MN/MX}$. Подача на этот вход напряжения +5 В указывает на то, что в МПС имеется лишь одна БИС МП (минимальная конфигурация системы). При этом МП вырабатывает на своих электродах все необходимые управляющие сигналы для связи с памятью и периферийными устройствами. При максимальной конфигурации МПС

$\overline{MN/MX}$ =0. МП выдает дополнительно на три вывода, подсоединяемые к контроллеру магистралей, управляющие сигналы.

Таблица 4.2.

Регистр	Название	Специальная функция регистра
AX	Аккумулятор	Умножение, деление, ввод-вывод слов
AL	Аккумулятор - младший байт	Умножение, деление, ввод-вывод байтов, преобразование байтов, десятичная арифметика
AH	Аккумулятор - старший байт	Умножение и деление слов
BX	Базовый регистр	Адресация по базе; преобразование адресов
CX	Счетчик	Подсчет циклов; подсчет элементов цепочек
CL	Счетчик (младший байт)	Реализация параметрических сдвигов
DX	Регистр данных	Умножение и деление слов; косвенный ввод-вывод
SP	Указатель стека	Операции с использованием стека
BP	Указатель базы	Базовый регистр
SI	Индекс источника	Указатель цепочки-источника, индекс, регистр
DI	Индекс приемника	Указатель цепочки-приемника, индексный регистр

$\overline{LOCK}$ - (запрет на пользование магистралями) – информирует устройства системы, что они не должны пытаться запрашивать шину.

$\overline{RQ}/\overline{GT0}$ $\overline{RQ/E0}$ } Запрос и получение подтверждения от основной МП БИС на возможность доступа к магистралям.
$\overline{RQ}/\overline{GT1}$ $\overline{RQ/E1}$ }

При работе максимальной конфигурации МП выдает восемь сигналов состояния процессора (ST0-ST7), которые могут использоваться внешними устройствами.

ST2 ST1 ST0 - указывают тип машинного цикла
0 0 0 Обслуживание прерываний
0 0 1 Чтение внешних устройств
0 1 0 Запись во внешние устройства
0 1 1 Останов
1 0 0 Извлечение кода команд
1 0 1 Чтение памяти
1 1 0 Запись в память
1 1 1 Не используется

Выводы ST3 и ST4 указывают на тип регистра сегментов, используемых при вычислении физического адреса.

0 0 ES
0 1 SS

1	0	CS (или никакой)
1	1	DS

Выводы QS0 QS1 Выполняется текущая операция с конвейером команд.

0	0	Нет операции
0	1	Извлекается первый байт команды из конвейера
1	0	Конвейер очищается
1	1	Очередной байт извлекается из конвейера

WAIT – ожидание; HLD – прямой доступ в память (запрос).

4.5. Система команд 16-разрядного процессора первого поколения

Система команд BM86 состоит из 91 мнемокоманд и позволяет совершать операции над байтами, двухбайтовыми символами, отдельными битами, а также цепочками байтов и слов.

По функциональному признаку система команд МП 8086 разбивается на 6 групп:

1) Пересылка данных;
2) Арифметические операции;
3) Логические операции и сдвиги;
4) Передача управления;
5) Обработка цепочек;
6) Управление процессором.

Команды пересылки данных составляют 4 подгруппы:

1) общие;
2) обращение к стеку;
3) ввода-вывода;
4) пересылка цепочек.

Эти команды, за исключением POPF и SAHF, не влияют на флаги. Команда MOV осуществляет пересылку содержимого источника src в получатель dst.

Команда **MOV r_1/m, r_2/m** обеспечивает пересылки регистр-регистр/память-память при использовании любого общего регистра и любого способа адресации.

Команда **MOV r/m, d** позволяет передать непосредственные данные в общий регистр или ячейку памяти.

MOV sr, r/m и **MOV r/m, sr** осуществляют пересылки между сегментным регистром и регистром или памятью. При этом передаются только слова, а ячейка памяти может быть определена с помощью любого допустимого способа адресации.

MOV *a*c, m и **MOV m, *a*c** предназначены для загрузки и запоминания содержимого аккумуляторов AL и AX при использовании прямой адресации.

Обращение производится к текущему сегменту данных, и адрес, указанный в команде, представляет смещение в этом сегменте. Если пересылаются два байта, то младший располагается по указанному адресу, а старший – по следующему. Так как не существует непосредственной загрузки сегментных регистров, то команда MOV sr, r/m используется для инициализации регистров SS, DS и ES, т.е. для определения соответствующих сегментов памяти. Если, например, в регистр DS необходимо загрузить число 8000, то используют две команды:

```
MOV AX, 8000H
MOV DS, AX
```

При этом в качестве промежуточного регистра обычно используют AX, т.к. команда MOV*ac*, d короче более общей команды MOV m/r, d.

Возможна также инициализация сегментных регистров из программной памяти при использовании префикса замены сегмента для замены DS на CS при вычислении адреса EA в следующих командах с прямой адресацией. При замене сегмента новый сегментный регистр указывается явно.

```
MOV DS, CS : ADS; Инициализация DS
MOV ES, CS :  AES; инициализация ES
MOV SS, CS :  ASS; инициализация SS
```

Следует помнить, что в МП BM86 обеспечивается защита процесса инициализации регистров SS и SP, состоящего из двух команд, от прерываний, чтобы гарантировать правильную работу стека.

```
MOV SS, CS : ASS
MOV SP, CS : ASP
```

Операнды ADS, AES, ASS, ASP обозначают адреса ячеек памяти, в которых хранятся базовые адреса соответствующих сегментов.

Команда

```
XCHG dst, sre; dst ↔sre
```

осуществляет обмен данными между источником и получателем и имеет два формата. Общий формат позволяет произвести обмен содержимым любой пары общих регистров, а также между общим регистром и ячейкой памяти при любом допустимом способе адресации. Указанный формат осуществляет обмен любого РОН и аккумулятора AX.

Команда XCHG AX,AX, код которой 90H, используется как команда операции NOP, обеспечивающая задержку на время 3T.

Однобайтная команда XLAT с кодом операции D7 предназначена для быстрого преобразования кодов и заменяет содержимое AL на байт из 256-байтовой таблицы, начальный базовый адрес которой содержится в регистре BX, т.е. содержимое AL используется как индекс таблицы, находящейся в сегменте данных и адресуемой регистром BX. При выполнении этой команды, к содержимому BX прибавляется содержимое AL, а полученный результат используется как смещение относительно DS. Адресуемый таким образом байт из памяти пересылается в AL.

Команды LEA, LDS, LES отличаются от других команд пересылки тем, что при их выполнении в адресуемый регистр (регистры) передаются не собственно данные из памяти, а адреса. Основное назначение этих команд – инициализация регистров перед выполнением цепочечных команд, или перед вызовом подпрограммы.

LEA r,m обеспечивает вычисление эффективного адреса EA ячейки памяти в соответствии с указанным способом адресации и загрузку EA (а не содержимого адресуемой ячейки памяти) в указанный РОН. Такая операция может потребоваться, например, для загрузки начального адреса таблицы в регистр BX перед выполнением команды XLAT.

Стек, как обычно, организуется в ОЗУ и его положение определяется содержимым регистров SS и SP. Регистр SS хранит базовый адрес текущего сегмента стека, а регистр SP указывает на вершину стека в стековом сегменте. При каждом обращении к стеку пересылается одно слово, а содержимое SP модифицируется автоматически, при записи в стек оно уменьшается на 2, при чтении из стека – увеличивается на 2.

При всех достоинствах организации памяти ВМ86 она имеет некоторый недостаток, заключающийся в трудности манипуляции физическими адресами при необходимости их программной обработки.

Команды ввода-вывода. Ввод-вывод данных может осуществляться двумя способами: с использованием адресного пространства ввода-вывода и с использованием общего с памятью адресного пространства. При первом способе применяются команды IN, OUT, которые обеспечивают передачу данных между аккумуляторами AL или AX и адресуемыми портами.

При выполнении этих команд вырабатывается сигнал M/IO=0, который вместе с сигналами WR и RD позволяет сформировать системные сигналы IWO и IOR для управления операциями записи данных в порт или чтения из порта.

Команды IN и OUT могут использовать прямую адресацию (аналог с ВМ80) и косвенную, когда адрес порта содержится в регистре DX. В первом случае можно адресовать 256 портов. Во втором – до 64К 8-битовых или 32К 16-битовых портов. Косвенная адресация позволяет вычислять адреса портов при выполнении программы, и удобна при организации вычислительных циклов для обслуживания нескольких портов с помощью одной процедуры. Ячейки с адресами F8 – FF зарезервированы для системных целей, и использовать их в прикладных программах не рекомендуется.

При втором способе адресации обращение к портам не отличается от обращения к ячейкам памяти, т.е. для ввода-вывода можно использовать любую команду обращения к памяти при любом способе адресации. Однако следует учитывать, что команды обращения к памяти имеют больший формат и выполняются дольше, чем простые команды IN и OUT. Кроме того, усложняется декодирование 20-битового физического адреса порта и сокращается число адресов, которые могут использоваться для ячеек памяти.

МП может передавать по ШД байт или слово из ВУ. Чтобы слово передавалось за один цикл шины, адрес ВУ должен быть четным. Адрес байтового ВУ может быть четным или нечетным, и соответственно порты этих ВУ должны

подключаться к линиям младшего и старшего байта ШД. Для раздельного обращения к этим портам дешифрирование адресов осуществляется с учетом сигналов на линиях BHE и A0.

Цепочные команды (обработки строк). Под цепочкой (строкой) понимают последовательность любых контекстно-связанных байт или слов, находящихся в смежных ячейках памяти. В системе команд МП К1810ВМ86 имеется пять однобайтных команд, предназначенных для обработки одного элемента цепочки за прием. Цепочной команде может предшествовать специальный однобайтовый префикс повторения REP. Число повторений задается регистром CX. Например, последовательность команд

```
MOV CX, 500
REP  MOVS DST, SRC
```

заставит МП выполнять команду MOVS 500 раз, уменьшая значение регистра CX после каждого повторения, до тех пор, пока <CX> не станет равным 0.

Остальные префиксы повторения, решение о продолжении или прекращении повторений принимают в зависимости от значения флага ZF:

REPE (repeat while equal) пока равно;

REPZ (repeat while zero) повторять пока нуль;

REPNE

REPNZ

Команда LODS *(load string)* пересылает операнд строка-источник, адресованный регистром SI, из сегмента данных в регистр AL или AX, а затем изменяет SI так, чтобы он указывал на следующий элемент строки. Его значение увеличивается, если флаг направления DF равен 0, и уменьшается, если DF=1. Формат команды имеет вид

LODS SRC ; <AC>←<SRC>.

Допускается использование мнемокодов LODSB и LODSW, указывающих тип строки (цепочки).

Пример:

Сравнить строки DST и SRC длиной 500 байтов каждая. В случае обнаружения первого несовпадения, элемент строки LODS загрузить в регистр AL.

```
CLD
LEA   DI,  ES : DST
LEA   SI,  SRC
MOV CX, 5000
REPE CMPSB
  JCXZ  M1          ; Выйти, если CX=0, на метку M1
  DEC  SI           ;  Подправить регистр SI
  LODS  SRC         ; считать элемент в AL
  …….
M1:  …….
```

Команда сохранения строки STOS (stove string) служит для запоминания содержимого аккумулятора в элементе строки, адресуемом регистром DI. После выполнения команды содержимое DI увеличивается на 1. Формат команды

STOS DST ; DST:=<AC>

Сегментный адрес для этой команды всегда находится в сегменте ES, а префикс замены сегмента не используется. Тип элементов строки можно указывать с помощью мнемокодов STOSB и STOSW.

Пример: Просмотреть строку W_STRING длиной в 200 слов. Если обнаружен не нулевой элемент, то он и следующие за ним 5 слов обнуляются.

```
CLD
LEA   DI,  ES : W_STRING
MOV AX, 0
MOV CX, 2000
REPNE SCASW
JCXZ  ALLO
SUB    DI, 2
MOV  CX, 6
REP      STOS W_STRING
ALLO:  …….
```

Команда пересылки строк MOVS используется для копирования байта или слова из одной части памяти в другую. Она имеет формат

MOVS DST, SRC ; DST:=< SRC>

Здесь строка-источник находится в сегменте данных DS, а строка приемник – в дополнительном сегменте ES. Байт или слово источника, адресуется регистром SI, а получатель DI. После выполнения команды значения SI и DI модифицируются для указания следующих элементов строк.

При флаге DF=0 осуществляется автоинкрементация, а при DF=1 – декрементация.

Пример: Скопировать 100 байтов из строки SRC в строку DST.

```
CLD                  ;  Установить DF=0 для обработки строки слева направо
LEA   SI,  SRC       ;  Занести смещение адреса SRC в SI, а
LEA   DI,  ES: DST ;  смещение адреса DST в DI
MOV CX, 100        ;  Установить счетчик элементов
REP  MOVS DST, SRC    ;  Скопировать байты.
```

Обычно ассемблер (в зависимости от версии) допускает использование для передачи цепочки мнемокоды MOVSB (move byte string) или MOVSW (move word string), которые определяют тип элементов строки 8-байт w-слово. При этом операнды в команде могут отсутствовать.

Команды сравнения строк CMPS (*compare string*). Формат команды имеет вид:

CMPS SRC, SRC ; < SRC > = < DST > ?

Эта команда производит вычитание байта или слова строки DST, адресуемой регистром DI из байта или слова цепочки SRC, адресуемой регистром SI. В зависимости от результата вычитания устанавливаются флажки, но сами операнды не изменяются.

Когда перед CMPS находится префикс REPE или REPZ операция интерпретируется как «сравнивать, пока не достигнут конец строки», (или пока элементы строки не будут равны). При наличии префикса REPNE или REPNZ

операция интерпретируется как «сравнивать, пока не достигнут конец строки» (или пока элементы строки будут равны).

Например, команды

```
        CLD
      MOV CX, 100
REPE   CMPS DST, SRC
```

будут сравнивать до 100 пар элементов строк SRC - DST, пытаясь найти два несовпадающих элемента.

Команды

```
        CLD
        MOV CX, 100
REPNE CMPS DST, SRC
```

будут сравнивать до 100 пар элементов строк SRC и DST, пытаясь найти два совпадающих элемента. Для явного указания типа элементов строки обычно допускаются мнемокоды CMPS и CMPSW, при этом операнды в команде отсутствуют.

Команда сканирования SCAS имеет формат:

```
SCAS DST  ; <ac> - <dst>
```

По этой команде вычитается элемент строки DST (байт или слово), адресуемое регистром DI, из содержимого аккумулятора AL или AX. В соответствии с полученной разностью устанавливаются флажки, но значение операндов не изменяется.

Для выполнения действий более чем над одним элементом строки, надо воспользоваться префиксами повторения REPE (REPZ) или REPNE (REPNZ).

Например, с помощью последовательности команд

```
        CLD
        LEA   DI,  ES: B_STRING
        MOV AL,
        MOV CX, 100
REPE   SCAS B_STRING
```

можно просмотреть до 100 элементов строки байтов B_STRING в поисках элемента, отличного от пробела. Для явного указания типа элементов строки, обычно используются мнемокоды SCASB и SCASW.

Команды безусловной и условной передачи управления.

Команды передачи управления используются в разветвляющихся и циклических программах, а также при вызове подпрограмм и возврате из них.

Сегментная организация памяти определяет два основных типа команд передачи управления: внутрисегментные NEAR (близкие) и межсегментные FAR (дальние). При выполнении команды типа NEAR модифицируется только регистр IP, и адрес переходов представляется одним словом и даже байтом, если используется короткий вариант перехода с ограниченным диапазоном адресов. При выполнении команды типа FAR изменяется содержимое регистров IP и CS и адрес перехода представляется двумя словами (сегмент:смещение), что позволяет перейти в любую точку адресного пространства памяти.

Ассемблер поддерживает большое число команд условного перехода, которые осуществляют переход в зависимости от состояния флагового регистра. Существуют знаковые и беззнаковые команды условного перехода. Использование команд определяется типом данных, над которыми производятся операции. В рассматриваемую группу команд входят команды безусловных и условных переходов, вызовов, возвратов, управления циклами и прерываний.

Команды безусловных переходов JMP производят модификацию регистра IP или регистров IP и CS без сохранения прежних значений этих регистров. Имеется три формата команды JMP типа NEAR, осуществляющих переход в пределах текущего кодового сегмента, и два формата типа FAR, осуществляющих переход в любую точку адресного пространства.

В двухбайтовой команде JMP dispL во втором байте содержится смещение, которое интерпретируется как знаковое целое. Это смещение добавляется (с предварительным расширением знака до 16 бит) к содержимому IP, которое соответствует адресу команды, находящейся после данной команды. Диапазон смещения dispL составляет от –128 до +127, причем при положительном смещении осуществляется переход вперед, а при отрицательном – назад. Данная команда реализует короткий переход, который в мнемоническом обозначении отмечается указателем SHORT (например, JMP SHORT COUNT – короткий переход к метке).

Трехбайтная команда JMP disp производит такое же действие, как и предыдущая команда, но содержит 16-битовое смещение (ассемблерный указатель NEAR PTR). Это смещение интерпретируется как знаковое число в диапазоне –32 768 до +32 767, что обеспечивает переход в любую точку кодового сегмента.

Команда JMP r/m осуществляет косвенный внутрисегментный переход (ассемблерный указатель WORD PTR), при котором в регистр загружается содержимое регистра или ячейки памяти.

Команда JMP m реализует косвенный межсегментный переход (ассемблерный указатель DWORD PTR) через содержимое двух ячеек памяти.

Команда JMP addr реализует прямой межсегментный переход (ассемблерный указатель FAR PTR) и содержит 4 байта адреса перехода: два байта сегментное смещение загружается в IP, а два байта – в регистр CS.

Команды вызова подпрограмм CALL имеют такие же форматы, как и команда JMP и выполняются аналогичным образом, за исключением того, что автоматически запоминается адрес возврата (т.е. адрес команды, следующий за командой CALL). С этой целью при внутрисегментных вызовах в стеке запоминается содержимое регистра IP, а при межсегментных вызовах – сначала содержимое IP, а затем CS.

Команды возвратов RET из подпрограмм возвращают управление программе, осуществившей вызов. Такая передача управления осуществляется путем извлечения из стека адреса возврата. Поэтому команды возврата не содержат никакой адресной информации.

Тип команды возврата выбирается в соответствии с типом команды CALL, осуществивший вызов данной подпрограммы. Каждая подпрограмма должна

обязательно содержать команду возврата. Команды условных переходов осуществляют передачу управления в зависимости от значения флаговых регистров. Имеется 18 команд условных переходов, которые представлены единым двухбайтовым форматом, позволяющим осуществлять короткие (в пределах от –128 до +127) переходы относительно указателя команд IP.

4.6. Способы адресации памяти и устройств ввода/вывода

Команды микропроцессора Intel 8086 реализуют разнообразные способы адресации, что упрощает организацию и использование сложных структур данных, а также расширяет возможности отдельных команд и повышает гибкость их применения.

1. **Регистровая адресация**. Операнд находится в одном из общих регистров МП, а в некоторых командах в одном из сегментных регистров.

Примеры:

```
MOV AX, SI  ; <SI>→<AX>
ADD DI, BX  ;  <BX>+<DI>
AND CL, AX  ;  Ошибка, несоответствия размеров регистров
XOR AL, AH  ; XOR <AL> и <AH>.
```

2. **Непосредственная адресация**. Операнды представляют собой константы длиной 8 или 16 бит, содержащиеся в командах. В МП нет команд непосредственной загрузки регистров.

```
SUB AL, 30H          ; <AL>- 48   (30H = 48D)
MOV CL, 10           ;  (10→<CL>)
AND AX, 0F000H       ; Выделить старших 4 бита в AX
XOR DH, 1            ; Инвертировать младший бит в  DH
CMP BL,40H           ; Сравнить содержимое BL с числом 64.
```

3. **Прямая адресация.** Эффективный адрес берется из поля смещения команды:

```
MOV AX, GAMMA   ;  В Акк AX загружено содержимое ячеек памяти
                ;  с адресом, полученным суммированием
                ;  <DS> сдвинутого на 4 разряда с адресом
                ;   переменной GAMMA, определенном в ячейке
программы.
        ADD TEMP, BL  ;  <BL>+ <<DS>↑4+TEMP>
```

4. **Косвенная регистровая**. Эффективный адрес (EA) находится в одном из базовых или индексных регистров. В косвенной адресации могут использоваться только регистры BX, SI, DI. Косвенные регистровые операнды заключают в квадратные скобки.

```
        ADD AX, [DI]   ;   К <AX> прибавляется содержимое ячейки
памяти,
                       ;  адрес который находится  в DI.
```

; исполнительный адрес: <DS>↑4+ <DI>.
MOV [SI], CL ; <<SI>> ←< CL>

5. **Базовая адресация** (база + смещение). ЕА определяется суммой значения смещения, указанного в команде и содержимого регистров BX или BP.

К операциям в памяти можно адресовать, указывая *прямой адрес,* т.е. называя имя соответствующей области памяти, либо *косвенные* - через регистры-указатели, или индексные регистры. При прямой адресации 16-ричное смещение автоматически складывается с базовым адресом соответствующего сегмента. При косвенном обращении участвуют один или два из четырех регистров. Они указываются в квадратных скобках [ ] – признак косвенной адресации BX, BP, SI, DI. Если указывается переменная, за ее именем следует выражение в квадратных скобках, которое задает базовые или индексные регистры.

В случае косвенной адресации может быть указан либо только базовый регистр, либо только индексный, либо оба регистра и может быть также указано 8 или 16-битовое смещение.

MOV AX, [BX] ; переслать слово из памяти в AX. Слово находится в сегменте
; данных, адрес этого сегмента в регистре DS, смещение
; относительно этого адреса в регистре BX.

К операндам, находящимся в памяти, можно обратиться одним из четырех способов:

Указанием прямого 16-разрядного смещения

MOV REPORT, AL ;в байт памяти с именем REPORT пересылается содержимое AL
; ([DS]↑4+[REPORT])→AL

Использованием косвенного обращения через базовый регистр, содержимое которого суммируется с 8 или 16-разрядным смещением.

MOV ON[BX+2], AL
MOV BL, ON[BP]

Использованием косвенного обращения через индексный регистр, содержимое которого суммируется со смещением

MOV CL, ITEM[SI+1]
MOV ON[DI+1], CL

Использованием косвенного обращения через базовый и индексный регистры, содержимое которых суммируется со смещением

MOV AH, ITEM[BX+1] [SI+1]
MOV ON[BX+1][DI+1]

При определении BP в качестве базового регистр обращение осуществляется к текущему сегменту стека SS (если нет префикса замены сегмента). Это делает базовую адресацию с регистром BP очень удобным средством обращения к данных, находящимся в стеке. Обычно базовый регистр BREG указывает на начало структуры данных, а требуемый элемент адресуется с помощью смещения (расстояния) от базы.

Для обозначения базовой адресации используют два представления:

1) [BREG] DISP BREG – базовый регистр (BX или BP)
2) [BREG+ DISP]

Пример записи команд с базовой адресацией:

MOV AX, [BP]10 ; Обе команды передают шестое слово массива, адресуемое BP,

MOV AX, [BP+10] ; в Аккумулятор,

ADD [BX]TEMP,CX ; Прибавить <CX> к слову TEMP в массиве, адресуемом BX.

6. **Индексная адресация** (смещенная база + индекс). Этот вид адресации называют адресацией с индексированием. Эффективный адрес вычисляется как сумма смещения, находящегося в команде, и содержимого индексного регистра DI или SI. Адресация удобна при доступе к элементам таблицы (массива), когда смещение указывает на начало таблицы (массива), а индекс – на элементы в таблице (массиве). По существу индексная и базовая адресация в МП К1810 аналогичны. Это объясняется тем, что базовые и индексные регистры имеют одинаковую длину. Индексная адресация обозначается в виде TABL[IREG]. Здесь TABL – 16-битовое смещение (адрес начала таблицы).

Примеры:

```
MOV ADRM [SI], AX   ; Передать <AX> в элемент массива с начальным
                    ; адресом ADRM
ADD CX, MASS [DI]   ; Прибавить к <CX> элемент массива.
                    ; MASS – смещение, указанное в команде.
```

Пример: Загрузить 3-й элемент массива в аккумулятор AL.

```
table DB 10, 20, 30, 40
MOV DI, 2  ; загрузить в индексный регистр номер выбираемого байта
           ; минус 1 (т.к. массив начинается с нулевого элемента)
MOV AL, TABLE [DI] ; загрузить 3-й байт таблицы в AL.
```

Примеры записи базового индекса адресации:

```
MOV AX, [BX+2+DI]  ; Операнды можно заключать в скобки
MOV AX, [DI+BX+2]  ; в любом порядке, а сдвиг можно
MOV AX, [BX+2][DI]  ; сочетать с любым из регистров
MOV AX, [BX][DI+2]
```

7. **Базово-индексная адресация** (по базе с индексированием). Эффективный адрес равен сумме содержимого базового регистра, индексного регистра и, возможно, смещения, указанного в команде. Этот способ целесообразно использовать при работе с двумерными таблицами. В этом случае базовый регистр содержит начальный адрес массива, а значения смещения и индексного регистра является смещением по строке и столбцу. В ассемблере МП 1810ВМ86 базово-индексная адресация представляется в виде: [BREG]ADR16[IREG].

Пример: загрузить в AX 16-разрядный элемент таблицы, состоящей из 4-х столбцов и 3-х строк, находящийся в третьей строке на третьей позиции (3,3) [a_{22}, если в таблице считать 0-й столбец и 0-ю строку].

```
TABLE DW 1024, 1048, 2048, 3600    ; Задание таблицы в начале ассемблерной
     DW 4100, 5000,   600, 2000        ; программы
     DW    80,   300, 4000, 5000       ;
  VALUE DB 2          ; указание номера элемента в строке минус 1 (т.к. считается с 0).
  ……..
  MOV BX, TABLE ; в базовый регистр BX заносится начальный адрес таблицы
  MOV DI, 16            ; в индексный регистр DI заносится смещение в байтах (ячейка
                        ; памяти) адреса начала третьей строки от начала таблицы.
  MOV BX, VALUE [BX][ DI] ; загрузка элемента (a22)=4000 в AX.
```

8. **Относительная адресация**. Эффективный адрес вычисляется как сумма фиксированного смещения, находящегося в команде и текущего значения программного счетчика PC. При этом значение PC равно адресу байта, следующего за текущей командой. В МП К1810 относительная адресация применяется только в командах условных и безусловных переходов, вызова подпрограмм и управления итерациями (или циклами). Следует отметить, что программист в ассемблерных программах указывает не значение смещения, а абсолютный адрес перехода, т.е. метку команды, которой необходимо передать управление. Значение смещения выполняется автоматически программа - ассемблер.

9. **Адресация цепочек**. Для обращения к операциям цепочечных команд, используются индексные_регистры. Регистр SI адресует первый байт (слово) цепочки источника, а регистр DI – первый байт (слово) цепочки получателя. В повторяющихся цепочных операциях МП автоматически изменяет содержимое регистров SI и DI.

10. **Адресация портов ввода-вывода.** Существует прямая и косвенная адресация портов. В прямой адресации номер порта представляет собой 8-битовый непосредственный операнд, находящийся во втором байте команды, что обеспечивает обращение к фиксированным портам 0-255.

При косвенной адресации номер порта находится в регистре DX и имеет диапазон 0-65535. С помощью предварительной инициализации регистра DX одна и та же команда может обращаться к любому порту в адресном пространстве ввода-вывода.

Примеры:

```
IN AL, 40H    ; Ввод байта из порта номер 40H
OUT DX, AX ; Вывод слова в порт с адресом, хранящемся в DX
IN AX, DX     ; Ввод слова из устройства, адрес которого хранится в регистре DX.
```

4.7. Структура и функционирование 16-разрядной микро-ЭВМ

Принципы построения 16-разрядных микро-ЭВМ имеют много общего с построением 8-разрядных машин. Микропроцессор обменивается информацией с внешними устройствами (ВнУ) и памятью по 16-разрядной шине данных с использованием 20-разрядной адресной шины, шины управления и шины состояния МП. Структурная схема микро-ЭВМ на основе микропроцессора КР1810ВМ86 показана на рисунке 4.4, а временные диаграммы функционирования машины в минимальном режиме – на рисунке 4.5.

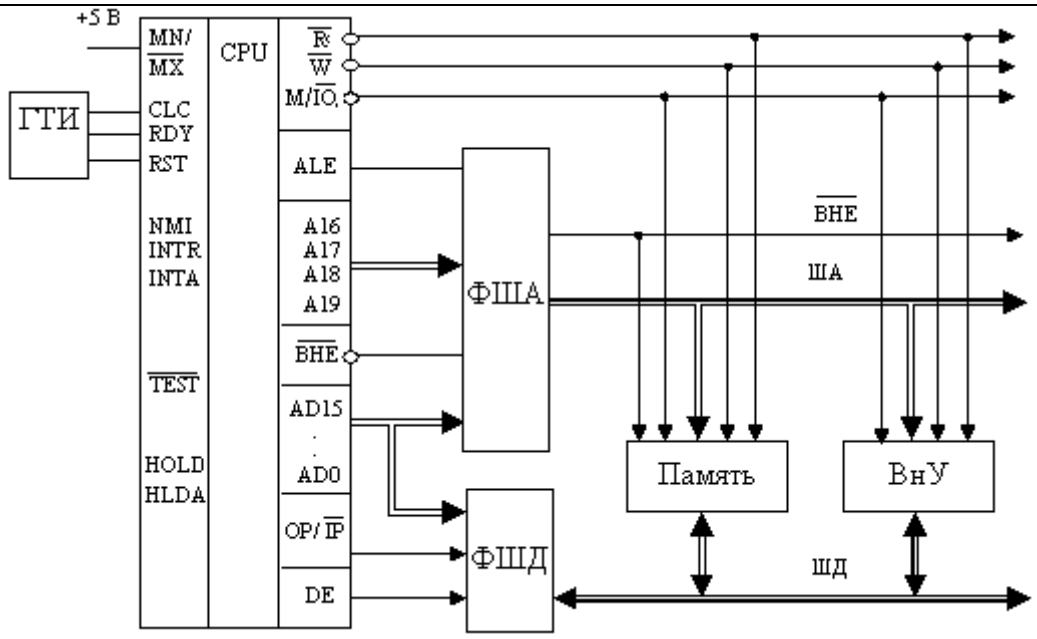


Рисунок 4.4 – Схема микро-ЭВМ на базе процессора КР1810ВМ86

Минимальный режим работы процессора задается путем подачи высокого уровня сигнала на вывод микросхемы MN/MX. При минимальном режиме управляющие сигналы для памяти и внешних устройств генерирует сам процессор, а в максимальном режиме для управления шинами используется специальный системный контроллер, который формирует управляющие сигналы на основании значения линий состояния S0-S2. Для генерирования последовательности тактовых импульсов CLC, сигнала готовности RDY, а также сигнала начальной установки RST используется функциональный генератор типа К1819ГФ84, входящий в состав микропроцессорного комплекта серии 1810.

Минимальный цикл обмена информацией микро-ЭВМ состоит из четырех машинных тактов. Цикл начинается с формированием на такте Т1 сигнала М/IO, определяющего тип устройства (ОЗУ или ВнУ), к которому производится обращение для пересылки данных. Длительность сигнала М/IO равна длительности цикла шины, и он используется для селекции адресуемого устройства. В такте Т1 и в начале такта Т2 микропроцессор выставляет адрес ОЗУ на линии А19-16 и АД15-0, либо адрес ВнУ, а также вырабатывает сигнал BHE, который

вместе с A0 определяет передачу слова или одного из байтов. Одновременно с этим МП выдает строб адреса ALE, по спаду которого адрес фиксируется во внешних регистрах-защелках.

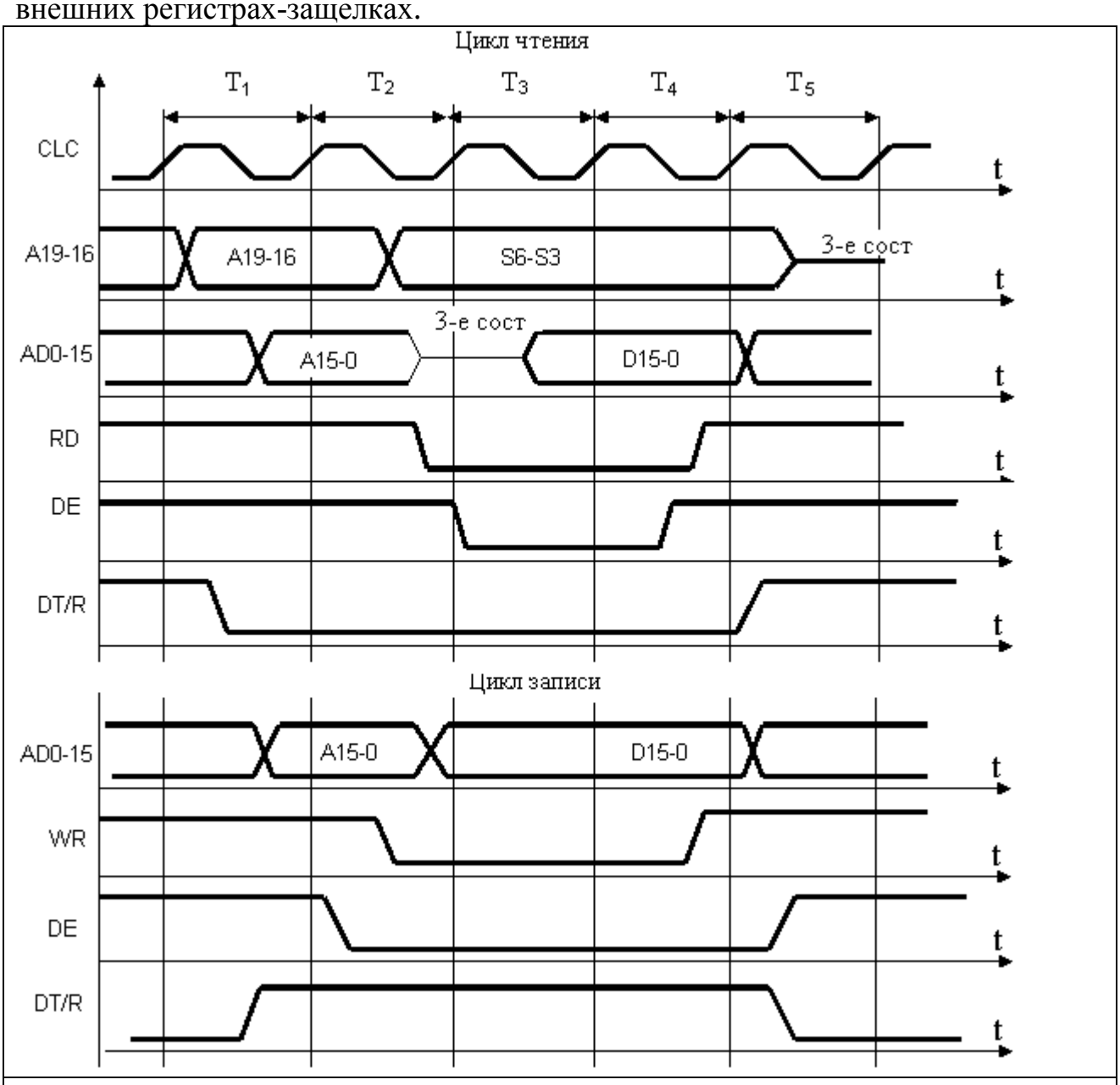


Рисунок 4.5 – Временные диаграммы функционирования 16-разрядной ЭВМ

В такте Т2 происходит переключение шин: на линии A19/S6 – A16/S3 поступают сигналы состояния S6-3, которые сохраняются до конца такта Т4. Значения этих сигналов зависят от вида выполняемого действия процессора. В минимальном режиме работы процессора эти сигналы практически не используются. В цикле чтения в такте Т2 линии АД15-0 переводятся в третье состояние, давая тем самым процессору перейти из режима записи (выдачи адреса) к режиму приема команды или данных. В тактах Т2-Т4 вырабатывается сигнал чтения RD=0, который указывает адресуемому устройству на необходимость выдачи слова. Для управления формирователем шины данных (ФШД), который

подключается к линиям АД15-0, в тактах Т2-Т4 формируется сигнал DE (Data Enable), разрешающий прием данных, действующий в течение всего цикла.

После выполнения чтения и установления сигнала RD=1 микропроцессор заканчивает такт следующим образом: линии АД15-0 переключаются в высокоомное состояние, сигналы M/IO, DE, DT/R, S7-3 переходят в неактивное состояние. Шинные формирователи данных отключены от канала.

Цикл записи отличается от чтения не только активными значениями сигналов RD или WR и состоянием DT/R, но и тем, что в цикле записи сигналы DE и WR становятся активными раньше и имеют большую длительность, чем в цикле чтения. Линии АД15-0 переключаются с адреса на данные без промежуточного перехода в третье состояние. Соответственно данные при записи имеют большую длительность, чем при чтении.

4.8. Защита памяти в процессорах второго и последующих поколений

В составленной программе, как правило, имеются ошибки. Ошибочный переход к области данных может привести их к уничтожению. Еще более опасна запись в область программы. При разрушении области операционной системы (ОС) обычно возникает системный отказ. Для предотвращения таких ситуаций в 16-разрядных процессорах второго поколения введена защита памяти. Кроме этого, в этих процессорах внедрена поддержка режима виртуальной памяти (ВП) и мультипрограммного режима. Для реализации этих нововведений в микропроцессорах используется устройство управления памятью (УУП).

Для осуществления защиты памяти кроме базового адреса вводится *указатель размера* и *атрибут сегмента*. В этом случае ситуация, когда величина смещения превышает размер сегмента, считается аварийной, т.к. возможно проникновение в область соседнего сегмента. Атрибуты сегментов могут быть самыми разными. Наиболее широко применяются следующие:

- разделение на **системную область** и **область пользователя**;
- разделение на **область программ** и **данных**;
- в случае работы с областью программы производят разделение ее на участки, допускающие только считывание данных или также и запись.

Процессор в каждом цикле вырабатывает сигналы состояния, показывающие, является ли данный цикл считыванием или записью. УУП при каждом обращении к памяти сравнивает атрибуты сегментов с этими сигналами. Если в размере сегмента или в его атрибуте обнаруживается ошибка, УУП выдает в ЦП сигнал сегментной ошибки, вызывающий специальное прерывание.

Для хранения базового адреса, размера и атрибута сегмента в УУП введены специальные регистры – **дескрипторы сегментов**. Число таких регистров определяется количеством используемых сегментов. При этом если номер сегмента является n–разрядным, то возможно использовать 2^n регистров. С целью ускорения поиска регистра они строятся на принципах ассоциативной памяти.

Для этого в составе каждого регистра имеется схема сравнения искомого номера регистра с хранящимся в регистре его номером.

Для защиты памяти также введена система привилегий (PL), с помощью которой осуществляется защита сегментов. Система привилегий регулирует доступ к тому или иному сегменту, в зависимости от уровня его защищенности и от степени важности (привилегированности) запроса. В МП Intel x286 и всех последующих моделях установлены четыре уровня привилегий PL, которые задаются номерами от 0 до 3 (рисунок 4.6). Наиболее привилегированным является уровень с меньшим номером. Степень защищенности сегмента также имеет четыре уровня, которые схематически представляются в виде вложенных колец защиты. В процессорах фирм Motorola (MC68000) и Zilog (Z8001) предусмотрено **два** уровня привилегий: в системном и пользовательском режимах.

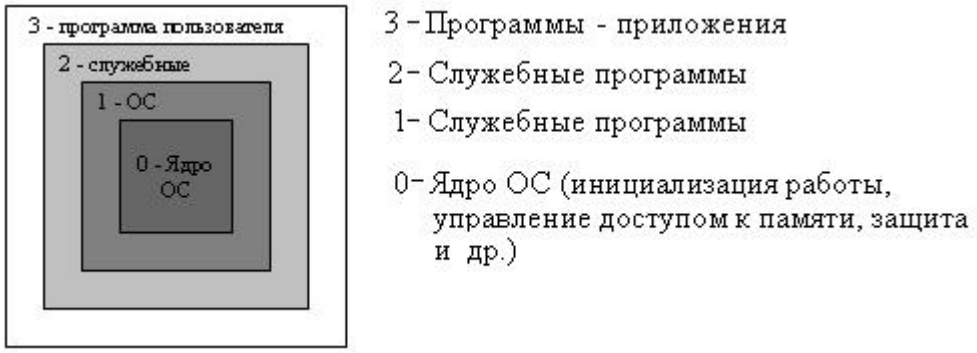


Рисунок 4.6 - Уровни привилегий и защиты

В соответствии с уровнями привилегий и защищенности установлены следующие правила доступа для сегментов программ и данных:

1) Данные из сегмента, имеющий уровень привилегий могут быть выбраны программой такого или более высокого уровня привилегий.

2) Сегмент программ, имеющий уровень защиты PL, может быть вызван программой, имеющий такой или более низкий уровень привилегий.

3) Уровень защиты и привилегий определяется двумя битами, значение которых указывает номер кольца защиты или уровня. Эти биты размещаются в байте доступа дескриптора (для привилегий дескриптора) или в селекторе (уровня привилегий запроса).

4) Выполнение команды ветвления ограничивается сегментами в пределах одной программы. В командах вызова и перехода допускается переход в сегменты другой программы при условии, что уровень ее привилегий равен текущему уровню.

Следует заметить, что процессор разрешает менее привилегированной процедуре вызвать более привилегированную, но ограничивает доступ разрешенными точками входа. Допустимые точки входа идентифицируются специальными дескрипторами, называемыми шлюзами вызова. Привилегия в шлюзе задается достаточно низкой, что позволяет обращаться в точку входа операционной системы. Размер смещения в шлюзе не учитывается.

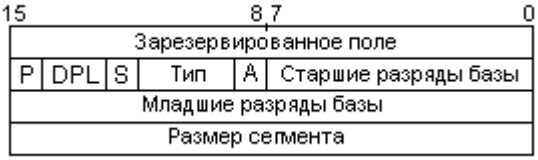


Рисунок 4.7 – Формат дескриптора сегмента процессора 8086

Дескрипторы сегментов располагаются в памяти ПЭВМ. Из четырех слов дескриптора одно не используется. Формат дескриптора сегмента процессора 8086 показан на рисунке 4.7.Оно зарезервировано для последующего поколения процессоров (386 и т.д.). В дескрипторе содержится 24-разрядный базовый адрес, 16 разрядов размера сегмента, 3 разряда типа, определяющие атрибут сегмента.

Кроме них имеются битовые поля: P – присутствие в основной памяти; A– обращение к сегменту; DPL – уровень привилегий дескриптора; S – бит системного сегмента. S=0 — системный; S=1 — сегмент программы или данных. В поле "Тип" указывается, кодовый это сегмент или данных, разрешена запись в него или запрещена, расширение сегмента осуществляется вниз (для стека) или обычное.

Существуют 4 уровня привилегий сегмента: 0 – высший; 3 – низший. Процессор разрешает менее привилегированной процедуре вызвать более привилегированную, но ограничивает доступ разрешенными точками входа. Допустимые точки входа идентифицируются в процессоре специальными дескрипторами, называемыми шлюзами вызова. Привилегия в шлюзе задается достаточно низкой, что позволяет обращаться в точке входа к операционной системе.

Группа дескрипторов сегментов, расположенных в глобальном адресном пространстве называется *глобальной таблицей дескрипторов* (GTD), а сегментов, расположенных в локальном адресном пространстве – *локальной таблицей дескрипторов* (LDT).

Если число задач, выполняемых ЭВМ, равно N, то и число локальных адресов также равно N и соответственно количество LDT также равно N.

GTD и N-LTD располагаются в глобальном адресном пространстве (рис.6.8).

Следует помнить, что сама LDT также представляет собой отдельный сегмент и для его описания требуется дескриптор. Дескриптор LDT хранится в GTD.

4.9. Поддержка многозадачности и виртуальной памяти

В компьютерах, построенных на основе 16-разрядных процессоров первого поколения, работавших под управлением DOS, одновременно выполнялась только одна задача. Процессоры второго поколения получили возможность поддержки **многозадачного** (мультипрограммного) режима. Обычно в памяти ЭВМ находится одновременно несколько программ: прикладная, ОС, библиотечные подпрограммы и др. При запуске прикладной программы выполняются не только ее команды, но и команды ОС и библиотечных прикладных про-

грамм. В процессе выполнения работы до ее окончания осуществляется переход от одной программы к другой. Такое выполнение последовательности команд называется *процессом*, а выполнение команд одной программы – *задачей*.

В обычном компьютере имеется только один процессор, поэтому в каждый момент времени он может выполнять команды только одной задачи. Поэтому в многозадачном режиме команды различных задач должны выполняться в режиме разделения времени.

Выполнение задачи может быть приостановлено на любой команде, поэтому необходимо обеспечить перезапуск соответствующей программы. Каждой задаче отводится отдельная область памяти, в том числе и стек для хранения всех данных, используемых в задаче. Кроме того, в памяти должны быть предусмотрены ячейки для хранения содержимого программного счетчика, регистра флагов ФР и регистров общего назначения РОН. Когда выполнение задачи приостанавливается, в них помещается начальные значения указанных регистров. При возобновлении задачи содержимое этих ячеек пересылается в ПС, ФР и РОН процессора, после чего начинается выполнение задачи. Совокупность этих ячеек называется *блоком управления задачей*. В случае приостановки выполнения команды задачи, информация, необходимая для ее перезапуска, должна сохраняться в *блоке управления задачей*, который размещается в *сегменте состояния задачи* (ССЗ – TSS). Для этого сегмента также используется свой дескриптор, который хранится в ГТД. Для выполнения N задач соответственно используется N сегментов состояния задачи. Таким образом, в ГТД хранятся N дескрипторов ЛТД и N дескрипторов ССЗ (TSS).

Когда некоторая задача находится в состоянии прогона, в соответствующую часть регистра задачи (TR) помещается селектор, с помощью которого осуществляется считывание дескриптора ССЗ. Автоматически считанный из ГТД дескриптор ССЗ помещается в регистр дескрипторов сегментов TR, в котором содержится базовый адрес ССЗ.

Сегмент состояния задачи состоит из 22-х слов (44 байта). В нем, помимо начальных значений всех регистров, хранятся также начальные значения указателей стеков 0 -, 1 – и 2-го уровней привилегий. В конце сегмента ССЗ располагается *селектор ЛТД данной задачи*. При запуске задачи восстанавливается содержимое регистров процессора, а содержимое ЛТД автоматически помещается в селекторную часть РгЛТД.

В верхней части (с начала) сегмента состояния задачи располагаются данные (2 байта) о связи с задачей, которая выполнялась непосредственно перед этой задачей.

При переключении на другую задачу осуществляются следующие действия:

1) Состояние выполняемой задачи помещается в ССЗ;
2) Селектор новой задачи – в селекторную часть TR;
3) Дескриптор ССЗ новой задачи в регистр дескрипторов TR;
4) Состояние новой задачи посылается в процессор. Кроме того, в процессор посылается селектор ЛТД, с помощью которого можно обратиться к новой ЛТД.

В 80286 эта процедура выполняется аппаратно, на что затрачивается ~180 тактовых импульсов (при f=10 МГц это составляет 18 мкс).

В ССЗ хранится информация об обратной связи, на основании которой указывается задача, вызвавшая вложенную задачу, Т.е., если переключение происходит в результате прерывания, то автоматически устанавливается связь с задачей, выполняемой в момент возникновения прерывания. Таким образом, в случае возврата автоматически осуществляется возвращение к прежней задаче.

Планирование выполнения задач может быть осуществлено в программе управления задачами в ОС. Для этого при запуске программы управления задачами информацию о связи со следующей задачей следует поместить в поле обратной связи ССЗ. (Имеются команды загрузки LTR и сохранения STR регистра задачи.)

С помощью вентилей задачи системные программы могут ограничивать право переключаться на специфические задачи. Вентиль – логический элемент, пропускающий только определенные процессы.

Для реализации **мультипрограммного режима** в ОС должны быть предусмотрены программы, выполняющие специальные функции. Одной из них является *программа управления задачей*, которая управляет переходом задачи из одного состояния в другое. Выбор, какая задача будет выполняться следующей, решается *планированием задач.* Замена задач в состоянии выполнения называется *переключением задач*. В этом случае в блок управления задачей необходимо передать содержимое ПС, ФР и РОН, а из блока управления новой задачей считать содержимое этих регистров. Однако для реализации этих действий программно требуется много времени. В целях сокращения времени, основные операции целесообразно выполнять аппаратными средствами.

Выбор, какая задача будет выполняться следующей, решается *планированием задач.* Замена задач в состоянии выполнения называется *переключением задач*. В этом случае в блок управления задачей необходимо передать содержимое ПС, ФР и РОН, а из блока управления новой задачей считать содержимое этих регистров. Однако для реализации этих действий программно требуется много времени. В целях сокращения времени, основные операции целесообразно выполнять аппаратными средствами.

В процессорах, начиная со второго поколения и выше, реализована аппаратная поддержка виртуальной памяти. Виртуальная память – это способ организации основной памяти большой емкости с помощью внешней памяти. Она позволяет при составлении программы распоряжаться всем пространством адресов, зарезервированных в процессоре. Такие адреса называются *виртуальными*.

При выполнении программы устройство управления памятью (УУП) преобразует виртуальные адреса в физические. Обычно в основной памяти размещается небольшое количество сегментов, к которым в данной части программы осуществляется обращение, т.е. размещаются только необходимые в данное время сегменты. Индикатором нахождения данного сегмента в основной памяти является бит P дескриптора сегмента. При ссылке на несуществующие сегменты выполнение программы приостанавливается и производится замена со-

держимого основной памяти. При этом сегменты, которые не предполагается использовать, посылают во внешнюю память, а на их место размещают требуемые сегменты. Затем возобновляется выполнение приостановленной программы. Функция замены осуществляется ОС.

Для отбора сегментов, которые используются наименее часто применяется бит А.При обращении к сегменту биту А присваивается значение 1. ОС в фиксированный момент времени проверяет значение А, и если оно равно 1, производит приращение содержимое строки справочной таблицы и сбрасывает А. Таким образом, с помощью справочной таблицы ОС отбирает сегменты, которые реже всего используются и поэтому в первую очередь подлежат замене.

4.10 Архитектура процессоров второго поколения

Структурная схема 16-разрядного процессора второго поколения, разработанного фирмой Intel, изображена на рисунок 4.8. Процессор состоит из четырех блоков: *адресного* AU, *шинного* BU, *исполнительного* EU и *командного* IU, причем все блоки могут работать параллельно.

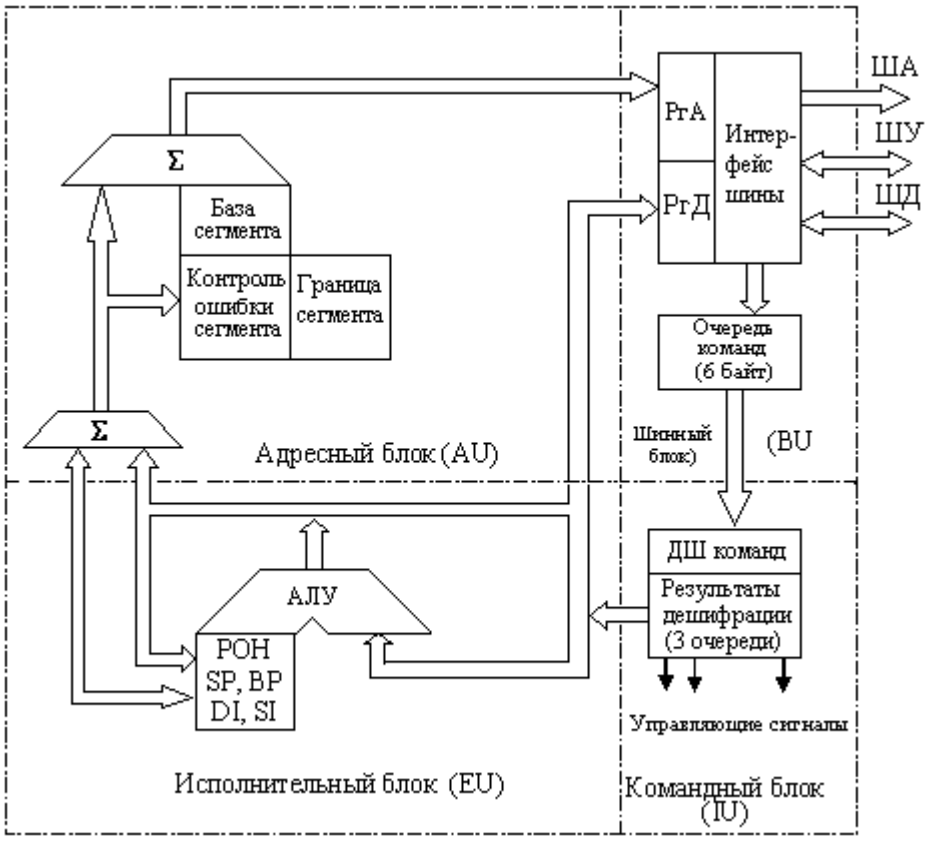


Рисунок 4.8 – Структурная схема 16-разрядного процессора второго поколения

Шинный блок осуществляет считывание памяти и портов ввода/вывода. Адресный блок вычисляет все адреса и формирует физические адреса. За счет независимой параллельной работы блоков производительность возрастает в 2-3 раза по сравнению с процессором 8086 (при том, что тактовая частота у него возросла всего до 12,5 МГц по сравнению с 5 МГц в 8086). Для использования шин с максимальной эффективностью применяется 6-байтовая очередь команд. При этом осуществляется упреждающее считывание команд, время выполнения которых в АЛУ велико. Результаты дешифрации помещаются в очередь результата. Поэтому командный блок после выполнения одной команды сразу же переходит к выполнению следующей. При наличии команд переходов очередь сбрасывается.

Процессор может работать в двух режимах: **реальном** и **защищенном.** Реальный режим используется для выполнения программ МП предыдущего поколения и инициализации регистров защищенного режима. Защищенный режим обеспечивает возможность поддержки виртуальной памяти, реализовать механизм привилегий с целью защиты памяти и многозадачную работу.

После подачи питания или сигнала сброса процессор устанавливается в реальный режим. В этом режиме процессор x286 имеет такую же базовую архитектуру, что и 8086, которая работает с большой скоростью. Различие двух режимов состоит в способе вычисления базового адреса.

БИС процессора 80286 имеет встроенное устройство управление памяти УУП. Это устройство оказывает поддержку виртуальной памяти и мультипрограммного режима. Регистровая структура МП имеет вид, изображенный на рисунке 4.9. В защищенном режиме в сегментных регистрах располагается не информация о физическом адресе, а *селекторы*, указывающие на описатели сегментов. С помощью селекторов, расположенных в сегментных регистрах, центральный процессор получает из локальной или глобальной таблиц дескрипторов описатели, характеризующие размещение и длину используемых сегментов.

Если процедура получения дескрипторов будет осуществляться при выполнении каждой команды, то это приведет к существенному снижению быстродействия. Дескрипторы находятся в ОЗУ и при поступлении нового селектора считываются в регистры дескрипторов. Таким образом, информация о физическом адресе сегмента и его длине при загрузке сегментного регистра загружается в так называемый *теневой регистр*, который имеется для каждого сегментного регистра. Т.е. каждый сегментный регистр содержит кроме видимой части – 16 битового селектора – другую невидимую часть, имеющую ширину 48 бит (для каждого из четырех селекторов). Аналогичные теневые регистры имеются для регистров глобальной и локальной таблиц дескрипторов.

Загрузка теневых регистров относительно медленная операция. Так для выполнения команды **MOV DS, AX** в реальном режиме требуется только два такта, то в защищенном режиме 18.

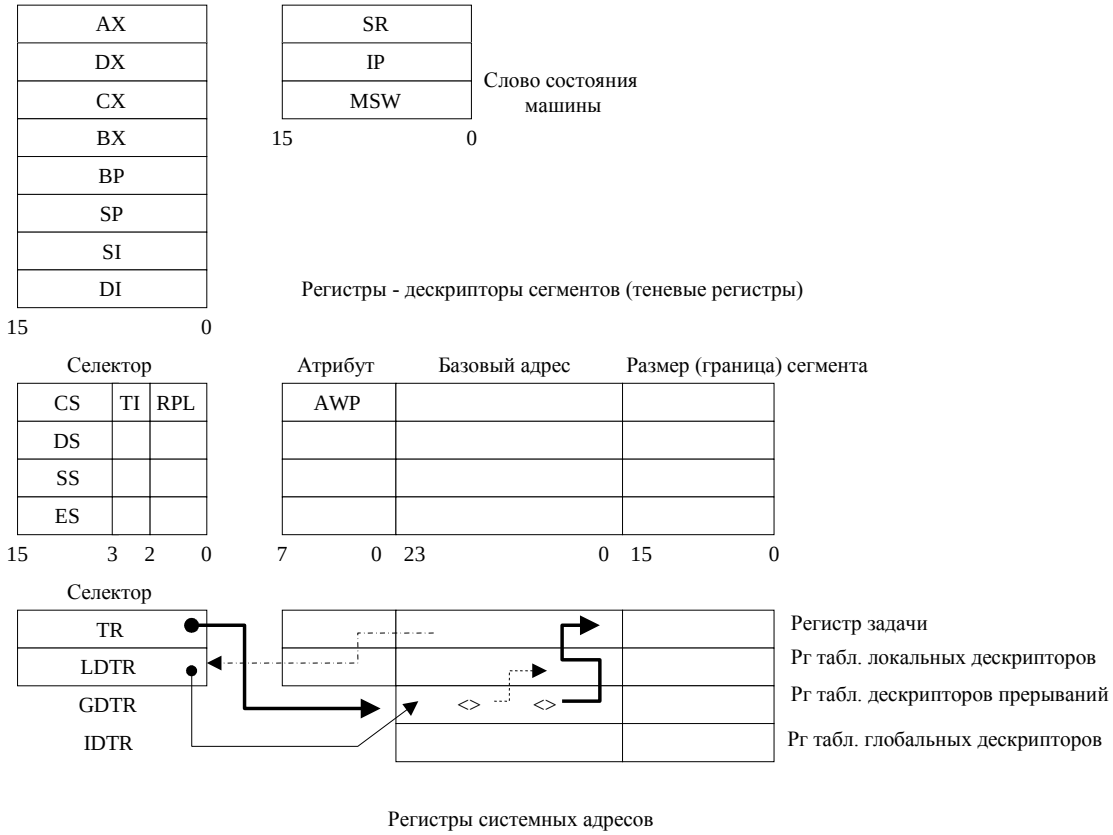


Рисунок 4.9 - Регистровая структура процессора Intel 80286

TI – Table Indicator; RPL – Requested Privilege Level.

По сравнению с МП 8086 структура регистров CS, DS, SS и ES, предназначенных для управления памятью, расширена за счет дополнительных регистров, используемых в качестве *дескрипторов сегментов*. Содержимое дескрипторов не используется непосредственно в программе. Их содержимое автоматически записывается центральный процессор и используется для управления памятью.

Регистр задачи TR указывает задачу, выполняемую в данный момент.

Все пространство виртуальных адресов состоит из *локальных* адресных пространств, отдельных для каждой задачи, и *глобального* адресного пространства, общего для всех задач. Системные программы (ОС и др.) хранятся в глобальном адресном пространстве. Пространство виртуальных адресов каждой задачи занимает объем 1 Гбайт $=2^{13}$ сегментов $\times$ 2^{16} – объем сегмента. Поле **TI** (Table Indicator) состоит из одного бита. Если этот бит равен нулю, для преобразования адреса используется так называемая глобальная таблица дескрипторов GDT (Global Descriptor Table), в противном случае - локальная таблица дескрипторов LDT (Local Descriptor Table).

Физические адреса, выдаваемые МП во внешние блоки, являются 24-разрядными (т.е. объем основной памяти может достигать 16 Мбайт). С помощью 24-разрядной базы физического адреса, хранящейся в дескрипторе сегмента, вычисляется физический адрес, который равен базе физического адреса + смещение сегмента.

Одновременно на основании атрибута и размера сегмента, хранящемся в этом же регистре, обнаруживаются сегментные ошибки и ошибки памяти. При их возникновении инициируется обработка исключительной ситуации.

Селектор – это 16-разрядный указатель, с помощью которого выбираются дескрипторы. В логическом адресе селекторная часть определяет индивидуальный дескриптор.

Селектор имеет следующие поля:

- *Индекс* – биты 15-3. Индекс выбирает один из 8192 дескрипторов в таблице дескрипторов; 8192 □ 8 (число байт в дескрипторе) = 64К – емкость сегмента;
- *Индикатор* таблицы TI – бит 3 определяет таблицу, на которую ссылается селектор: 0 – означает GDT, а 1 – текущую таблицу LDT;
- Запрашиваемый *уровень привилегий* RPL (2-0).

Регистры системных адресов GDTR, IDTR, LDTR и TR служат для обращения к элементам, которые управляют механизмом сегментации памяти.

Такими элементами являются таблицы и сегменты, входящие в структуру защиты памяти МП 286/386:

GDT – таблица глобальных дескрипторов

IDT – таблица дескрипторов прерываний

LDT – таблица локальных дескрипторов

TSS – сегмент состояния задачи.

Регистры GDTR и IDTR содержат 24–разрядные линейные адреса базы и 16-разрядные величины границ GDT и IDT, которые являются глобальными по отношению ко всем задачам. Перед переходом в защищенный режим программа должна создать в ОЗУ таблицу GDT и загрузить регистр GDTR при помощи специальной команды LGDT.

Регистр таблицы локальных дескрипторов LDTR и регистр задачи TR содержит 16-разрядные селекторы сегментов LDT и TSS, которые определены для конкретной задачи. С каждым из них связан программно недоступный регистр дескрипторов сегмента.

В МП 80286 входит регистр *слова состояния машины* (это не регистр флагов! Он имеется само собой) В его состав входят следующие поля:

- PE (Protected Mode Enable) – «защита разрешена». Устанавливается в 1 при работе в виртуальном режиме, 0 – в реальном;
- MP – бит присутствия сопроцессора. Если MP=1, то 286 будет учитывать команды WAIT, ESC;
- EM – режим эмуляции. Если EM=1, то МП при встрече команды ESC генерирует особый случай отсутствия сопроцессора. Процедура обработки особого случая в случае отсутствия сопроцессора может эмулировать команды сопроцессора программно;
- TS – бит “задача переключена”. Устанавливается автоматически, когда МП осуществляет переключение задач.

Процессор имеет различные виды адресного пространства: физическое, логическое и линейное.

Физическое – это реальные адреса, используемые для выбора ячеек микросхем физической памяти, содержащих данные.

Логический адрес состоит из *селектора* и *относительного адреса внутри сегмента*.

Линейный адрес – это адрес, сформированный добавлением относительного адреса к базовому адресу сегмента.

Допустим, что процессор выполняет задачу, в которой используется некоторое локальное адресное пространство и ЛТД сегментов, расположенных в нем. Дескрипторы ЛТД-сегментов хранятся в ГТД. При инициировании этой задачи селектор ЛТД, служащий для выбора дескриптора сегмента, автоматически помещается в соответствующее поле регистра ЛТД (РгЛТД). Индекс, содержащийся в селекторе, дает смещение относительно базового адреса ГТД. Дескриптор сегмента имеет длину 4 слова или 8 байт. Индекс помещается в старшие 13 бит селектора. Поэтому смещение в памяти равно:

Смещение=Числовое_значение_индекса□8 .

При таком способе обращения автоматически считываются дескрипторы сегментов ЛТД, хранящиеся в ГТД, и помещаются в свой РгЛТД. С помощью содержимого этого регистра можно обратиться в ЛТД. Схема преобразования виртуальных адресов в физические изображена на рисунок 4.10.

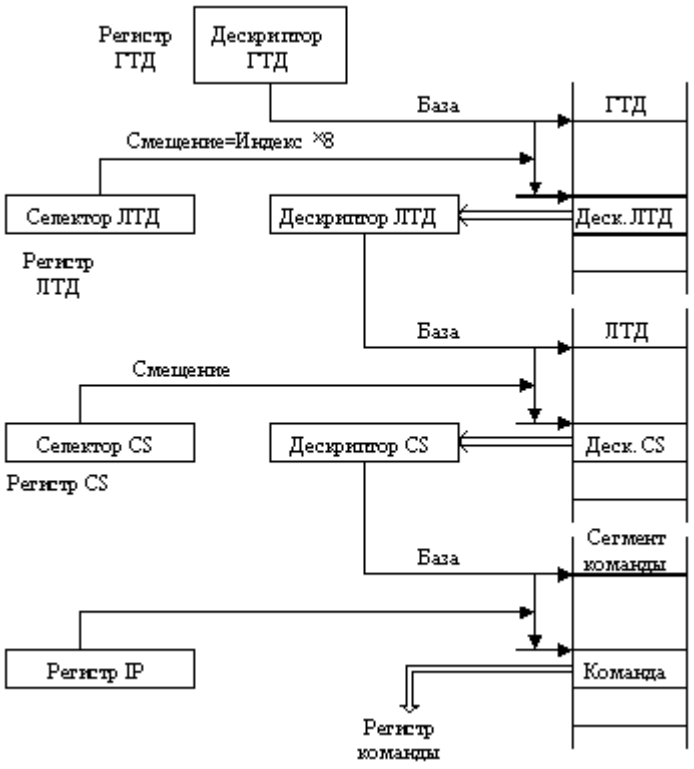


Рисунок 4.10 – Схема извлечения команды

В селекторной части регистра CS хранится селектор сегмента программы. Дескриптор сегмента программы, указываемый этим селектором, автоматически считывается из ЛТД и помещается в регистр дескрипторов сегментов CS. Содержимое его принимается за базу, а содержимое IP – за смещение, и на основании их значений осуществляется обращение к памяти. Процессор может одновременно обратиться к четырем сегментам. В этих сегментах уже задается смещение, поэтому формирование адреса может осуществляться с высокой скоростью.

Система команд процессора 80286 включает все команды 8086-го процессора и дополнена рядом новых команд:

- сохранение константы в стеке, сохранение и восстановление всех регистров PUSHA (Push All – все регистры), POPA;
- целочисленное умножение на константу;
- сдвиги с указанием счетчика в константе;
- вход и выход из процедур ENTER, LEAVE;
- контроль диапазона BOUND;
- ввод/вывод байта (слова) в строку, одиночный INSB, INSW, OUTSB, OUTSW, или по счетчику в регистре CX, повторение строковых команд по префиксу REP;
- сброс флага переключения задач CTS;
- команды управления защитой LGDT, SGDT, LIDT, LMSW, SMSW, доступные в обоих режимах и только в защищенном режиме: LLDT, SLDT, LTR, STR, ...

Попытка выполнения недействительной команды (или в реальном режиме команды, предназначенной только для защищенного режима) вызывает исключение 6.

4.11. Программирование 16-разрядных микропроцессоров на Ассемблере

Допустимые символы языка ассемблера состоят из прописных и строчных букв (латинских), цифр, специальных знаков +, -, *, /, =, (), [], ’, ’’, ., ;, @, &, ?, <, >, % и символов: перевод строки ПС (OAH), возврат каретки ВК (ODH), табуляции (O9H). Любой другой символ воспринимается как пробел. Наименьшей конструкцией модуля является идентификатор – последовательность букв и цифр (не более 31), начинающийся с буквы. К зарезервированным именам относятся имена регистров, операций, псевдокоманд. Модуль представляет собой последовательность операторов языка, записанных в одной строке и заканчивающихся возвратом каретки и переводом строки. Если в первой позиции оператора стоит символ &, то оператор является продолжением предыдущего.

Операторы разделяются на командные и директивы. Команды порождают одну машинную команду. Директивы (псевдокоманды) содержат управляющую информацию для ассемблера. Оператор команды имеет вид:

{метка:} {префикс} {мнемоника} {операнд(ы)}; { комментарий}.

Значения *метки* являются текущим значением счетчика в данном сегменте кода, то есть, представляет собой адрес команды. *Префикс* позволяет формировать байты блокировки LOOK или повторения REP. *Мнемоника* идентифицирует тип генерируемой команды. В зависимости от функции команды может быть один операнд, два или ни одного. Более двух операндов указывается в макрокоманде. *Комментарии* поясняют смысл команды.

Директивы ассемблера имеют несколько другой формат:

{имя} директива {операнд(ы)} {; комментарий}

Имя директивы имеет другой смысл по сравнению с меткой и не заканчивается двоеточием. В ряде директив имя отсутствует.

Директивы используются для распределения памяти, связей между модулями, манипуляции с символами и т.д. В отдельных директивах допускаются списки операндов. Операнды могут быть ключевыми словами в директивах PROG, SEGMENT. Для определения макрокоманд используется оператор вида:

MACROCODE имя {операнд(ы)}; {комментарии}

Переменная – это единица данных, имеющая имя. Она имеет три атрибута: **сегмент, смещение** и **тип.** *Сегмент* SEG определяет сегмент, содержащий переменную. *Смещение* OFFSET, расстояние от начала сегмента до переменной, *тип* – число байтов переменной (1,2 или 4).

Метка, представляющая имя ячейки памяти, имеет атрибутами сегмент, смещение, расстояние. *Константа* отличается от переменной и метки тем, что она определяет только число. Символьные цепочки заключаются в апострофы и обычно имеют длину до 255 знаков.

Для определения и инициализации данных предназначены директивы:

DB - определить байт (Define Byte)
DW - определить слово (Define Word)
DD - определить двойное слово (Define Double Word).

Формат директивы:

имя DX <начальное значение>, [< начальное значение>]

Пример определения переменных:

```
Z1 DB 0ABH            ; один байт, равный AB
Z2 DW 1000H           ;  одно слово, равное 1000
Z3 DD 1235H; 1000H    ; младшее слово 1000, старшее 1235.
```

Для указания произвольного значения ячеек памяти используется символ ? (значение переменной не оговаривается, резервирование ячейки). Для распределения и инициализации нескольких ячеек памяти используется конструкция DUP:

```
DB    100 DUP(0)              ; сто нулевых байтов (Dublicate)
DW    20 DUP(?)               ; 20 слов без значения

ADR   DD   10 DUP(ADR)    ; десять полных адресов
DB    10 DUP(80DUP)         ; десять слов, содержащих 80 пробелов
```

В директивах DW, DD допускаются символьные цепочки длиной 1 и 2 символа, например:

DW ’K1’; DD ’G’.

В выражениях, где запрашивается тип используемой информации, применяются операторы TYPE, LENGTH, SIZE. TYPE (тип) сообщает число байт, отведенных для переменной (1,2,4). LENGTH (длина) определяет число единиц памяти переменной (байт, слов, двойных слов). SIZE (размер) сообщает размер переменной в байтах, причем размер переменной равен длине, умноженной на тип.

В ассемблере вводится понятие логического сегмента, под которым понимается часть программы, которая может включать сегменты для машинного кода, данных и стека. Каждый логический сегмент должен начинаться с директивы SEGMENT и заканчиваться директивой ENDS. Логическому сегменту присваивается имя, данное программистом, и список параметров (атрибутов), которые не обязательны, но необходимы в случае программы, включающие несколько модулей.

Пример определения простого сегмента:

```
DATA  SEGMENT  [<список атрибутов>]

    F1    DB ?

    F2    DW ?
    F3    DD ?
DATA ENDS
```

До использования сегментного регистра в формировании адреса он должен быть инициализирован, для чего используется имя логического сегмента.

Пример:

```
MOV AX, DATA
MOV DS, AX
```

До использования стека необходимо инициализировать регистры SS и SP. В общем случае в программе первыми командами являются команды инициализации регистров DS, SS, SP. В языке ассемблера К1810 допускаются вложенные сегменты. Это означает, что если определен некоторый сегмент S1 (парой директив SEGMENT и ENDS), затем определяется сегмент S2, а потом в новой директиве SEGMENT снова появляется сегмент с именем S1, то содержащиеся в новом сегменте данные и (или) команды будут автоматически размещаться за операторами, которые находятся в старом сегменте с именем S2. Вложенный сегмент должен заканчиваться раньше внешнего сегмента.

Параметры директивы SEGMENT.

1. **Выравнивание**. Определяет границу начала сегмента. Обычное значение - PARA, по которому сегмент устанавливается на границу параграфа. В этом случае адрес кратен 16. При отсутствии этого операнда ассемблер по умолчанию принимает PARA; адрес сегмента XXX0. Бывает: PAGE=XX00; WORD=XXXE (четная граница); BYTE=XXXX – любая шестнадцатеричная цифра.

2. **Объединение**. Определяет, объединяется ли данный сегмент с другими сегментами в процессе компоновки после ассемблирования. Возможны следующие типы объединений:

STACK, COMMON (общий), PUBLIC (общедоступный), AT- и MEMORY.

Все **PUBLIC** - сегменты, имеющие одинаковое имя или класс, загружаются компоновщиком в *смежные области*. Все такие сегменты имеют один общий базовый адрес.

Для сегментов **COMMON** с одинаковыми именами и классами компоновщик устанавливает один *общий базовый адрес*. При выполнении происходит наложение одного сегмента на другой. Размер общей области определяется самым длинным сегментом.

AT-параграф: он должен быть определен предварительно. Данный операнд обеспечивает определение меток и переменных по фиксированным адресам в фиксированных областях памяти, таких как ROM или таблица векторов прерываний. Например, для определения адреса дисплейного видеобуфера используется

VIDEO_RAM SEGMENT AT 0B800h.

Для сегмента стека используется объединение STACK.

Если программа не должна объединяться с другими программами, то указание типов объединения должно быть опущено (за исключением STACK).

3. Класс. Данный элемент, заключенный в апострофы, используется для группирования относительных сегментов при компоновке. Он может содержать любое имя и используется ассемблером для обработки сегментов, имеющих одинаковые имена и классы.

Типичными примерами являются классы ‘STACK’ и ‘CODE’. (См. П. Абель с. 55 и 396).

Директива ASSUME (присвоить). Необходимая ассемблеру информация о значении сегментных регистров сообщается в директиве ASSUME, которая имеет формат:

ASSUME <SR: базовое значение>, [<SR: базовое значение>]…

Директива ASSUME DS: data1 требуется для транслятора, указывая ему, что сегмент сопоставляется с регистром DS. Описание в директиве ASSUME соответствия сегментного регистра DS сегменту данных избавляет нас от необходимости указывать в каждой строке, содержащей ссылку на имя данных, в каком сегментных регистров находится сегментный адрес этих данных. По умолчанию будет подразумеваться регистр DS. Если регистр ES в директиве ASSUME не описан, его следует в явной форме указывать во всех строках программы, где выполняется адресация к дополнительному сегменту данных:

Mov BX, ES: mas[DI]

Однако, и в этом случае занесение в регистр ES требуемого сегментного адреса должно быть сделано в явном виде:

```
mov ax, data1
mov ex, ax
```

Поле SR содержит имя одного из сегментных регистров CS, DS, SS, ES, а базовое значение указывает на начало области памяти, адресуемой через этот регистр. Одним из наиболее часто используемых типов базового значения является имя сегмента, например

ASSUME DS: DATA

Такая директива сообщает ассемблеру, что к определенным в сегменте DATA переменным можно обратиться через сегментный регистр DS. Директива ASSUME необходима перед использованием сегментных регистров и перед каждой точкой в программе, в которой может модифицироваться содержимое сегментных регистров. Если какой-либо сегмент не будет использован в модуле, то директива будет иметь вид ASSUME NOTHING.

Директива ORG (origin - начало). Основной внутренней переменной ассемблера является счетчик адресов, который при ассемблировании выполняет функцию, аналогичную функции программного счетчика при выполнении программ. Этот счетчик сообщает ассемблеру адрес следующей ячейки памяти (имеется в виду *смещение* в сегменте), которая предназначена для размещения следующего байта команды или данных.

Первое появление директивы <имя> SEGMENT определяет начало сегмента с заданным именем. При этом организуется *новый* счетчик ячеек, в который загружается нулевое значение, так как первый байт в сегменте имеет нулевое смещение. При распределении каждого следующего байта производится инкремент счетчика ячеек.

Директива ENDS с тем же именем отключает данный счетчик до тех пор, пока этот же сегмент не будет открыт еще одной директивой SEGMENT. В этом случае счетчик продолжает счет распределяемых байт со старого значения.

Текущее значение счетчика адресов может быть принудительно изменено программистом с помощью директивы ORG <выражение>. При выполнении директивы ORG вычисляется значение выражение и полученный результат загружается в счетчик ячеек (адресов).

Ассемблирование следующих команд или элемента данных производится по полученному адресу (смещению в текущем сегменте), который изменяется в диапазоне 0-65535 (вычисление осуществляется по модулю 64К).

Директива EQU. Директива позволяет определить символические имена для часто используемых выражений. Формат директивы:

<имя> EQU <выражение>

Поля выражения может определять константы, адреса, регистры и даже мнемокоды макрокоманд. Именам целесообразно придавать содержательный

смысл. Допустим, в программе необходимо многократно применять размер таблицы. Тогда его можно определить:

SIZE TABL EQU 100; 100 – размер таблицы.

Затем можно использовать имя SIZE TABL в любой команде, где требуется использовать этот размер. Такой прием улучшает понимание программы, а при расширении таблицы достаточно заменить в директиве EQU число 100 другим размером и произвести повторное ассемблирование программы. Ассемблер автоматически заменит каждое появление в новой команде SIZE TABL новым размером.

Присвоенные имена сохраняют одно и тоже значение во всей программе, если не будут отменены директивой PURGE, которая имеет формат

PURGE <имя>, [<имя>]…

Если какое-то имя удалено, то с помощью директивы PURGE, его можно переопределить. Пример использования директивы EQU:

```
CR EQU ODH              ; численная константа
LF EQU OAH
BET EQU ALPHA [SI]+3    ; адресное выражение
COUNT EQU CX            ; регистр
```

Имена, присвоенные директивами EQU, можно использовать в любых операторах исходного модуля, например:

```
MOV CL, LF   ; загрузить в CL значение OA
INC COUNT    ; инкремент содержания регистра CX,
             ; используемого в качестве счетчика.
```

Процедура (замкнутая подпрограмма) представляет собой законченную командную последовательность, которая приводится в действие командой вызова CALL. Процедура является одним из средств разработки модульных программ. Каждая процедура допускает автономную отладку, что позволяет ускорить разработку и отладку всей прикладной программы.

Команда CALL содержит метку одной из команд процедуры. Метка в процедуре, которой передает управление команда CALL, называется *точкой входа процедуры*. Процедура выполняется до тех пор, пока не встретится команда возврата RET.

Процедура обычно разрабатывается как функциональный блок, который формирует набор выходных данных путем преобразования четко определенных входных данных, называемых *параметрами*. Поэтому суть действий с процедурами сводится к передаче им параметров и получении из них результатов. Для организации процедур в языке ассемблера предназначены директивы PROC и ENDP. Директива PROC отмечает точку входа процедуры, а директива ENDP – окончание процедуры.

Формат этих директив имеет вид:

```
[имя]  PROC  [тип]
. тело процедуры
[имя]  ENDP
```

Имя представляет точку входа в процедуру. *Тип* процедуры (NEAR или FAR) (по умолчанию NEAR) используется ассемблером для определения вида генерируемой команды CALL для вызова процедуры. Если указан тип NEAR, то процедура находится в том же сегменте кода, в котором находятся все команды CALL, вызывающие эту процедуру. В этом случае ассемблер генерирует команду внутрисегментного вызова, то есть машинная команда CALL содержит только смещение точки входа.

Если указан тип FAR, процедура находится в сегменте, отличающимся от того сегмента кода, в котором находятся команды вызова CALL. То есть ассемблер должен генерировать длинную команду межсегментного вызова, которая содержит базу и смещение точки входа. При передаче управления в стеке приходится запоминать, а затем восстанавливать содержимое регистров CS и PC.

В соответствии с двумя форматами команд вызова необходимы и две разновидности команды возврата RET. Ассемблер определяет генерируемую разновидность команды RET на основе типа, указанного в директиве PROC.

Наиболее употребляемые директивы ассемблера МП 8086 приведены в табл.4.1

Таблица 4.1 – Директивы ассемблера

Название (назначение)	Запись/Пример	Комментарии
Определение данных	[имя] Dn операнд; [комментарий] X1 DB 25 Y2 DB ? Z DB 11, 12, 13 *DD ?* DT ? MON DB 01, ‘JAN’, 02, ‘FEB’, 03, ‘MAR’	X1:=25 У2 не определен Z:=11; Z+1:=12; Z+2:=13
Определение повторяющихся констант	[имя] Dn число повтор. UP(выражение) DW 10 DUP(?) DB 5 DUP(14) DB 3 DUP (4DUP(8))	10 неопредел. слов 5 байт=OEH 12 восьмерок
Задание сегмента Завершение сегмента	[имя] SEGMENT [параметры] STACKSG SEGMENT PARA STACK ‘Stack’ DATASG SEGMENT PARA ‘Data’ CODESG SEGMENT PARA ‘Code’ . имя ENDS	
Процедура Завершение процедуры	[имя процедуры] PROC FAR или NEAR . .. RET Имя процедуры ENDP	

Присвоение	ASSUME операнды ASSUME SS::имя_стека,CS::имя_сегмента, DS::имя_ сегмента, ES::имя_сегмента	
Установка текущего значения IP	ORG выражение ORG100H	Программа размещается с адреса 100H.
Управление листингом	PAGE кол. строк, кол. символов PAGE 60,132 TITLE текст (напр., имя программы)	По умолчанию PAGE 66,80 Вверху каждого листа будет напечатан текст.
Присвоение имени програмы	NAME имя	

Пример определения процедуры:

```
SRED PROC NEAR
MOV CX, AX;        передать <AX> в <CX>

ADD CX, DX;        прибавить <DX> к <CX>

RCR CX,1;          сдвинуть <CX> вправо (разделить на 2)
RET
SRED ENDP
```

Эта процедура определяет среднее значение содержимого регистров DX и AX помещает его в CX. Для вызова процедуры используется команда CALL SRED.

К переменным или меткам внутри тела процедуры можно обращаться из любого места программы. Для предотвращения случайного выполнения процедуры без вызова, ее рекомендуется размещать выше тех программ, которые ее вызывают, а также избегать вложенных определений PROG - ENDP.

.

5. Архитектура 32-разрядных процессоров

5.1. Структурная схема и регистры процессоров

Структурная схема 32-разрядного процессора Intel 80386 изображена на рисунке 5.1. Процессор 80386 состоит из следующих шести функциональных блоков, которые работают параллельно:

- интерфейсный блок;
- блок предварительной выборки команд;
- блок предварительной дешифрации команд;
- блок центрального процессора;
- блок сегментации;
- блок управления страницами.

Центральный процессор включает в себя операционное ОУ и управляющее УУ устройства. ОУ состоит из АЛУ и восьми 32–разрядных РОН. Особенностью АЛУ является наличие *64 – разрядного сдвигателя,* используемого при быстрых арифметических и циклических сдвигах, умножении и делении. В связи с этим 32–разрядное умножение выполняется менее чем за 1 мкс.

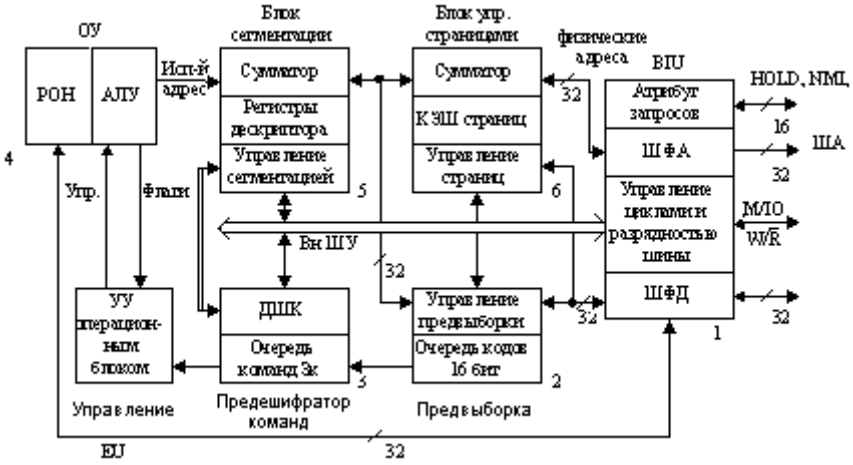


Рисунок 5.1 – Структурная схема 32-разрядного процессора 80386

Подсистема выборки команд реализует двухступенчатый алгоритм конвейеризации и состоит из блоков *предвыборки* кодов и *преддешифрации* команд. Первый из них осуществляет заполнение очереди команд длиной 16 байт.

Во втором блоке производится преддешифрация, определяется тип и формат команд, выделяется поле относительного смещения, содержимое которого поступает в блок сегментации для вычисления линейного адреса.

Команды, подготовленные к выполнению, хранятся в *очереди команд,* куда помещается в среднем 3 команды.

Диспетчер памяти (MMU – Memory Management Unit) состоит из блока **сегментации** и блока **управления страницами**. Осуществляет двухступенчатое формирование физического адреса ячеек памяти.

Наличие диспетчера памяти определяет два режима работы МП:

- режим реальных адресов (реальный режим);
- и режим защищенных виртуальных адресов.

В *реальном режиме* МП 386 работает как очень быстрый 8086, но при необходимости с расширением разрядности операндов и адресов до 32.

В *защищенном режиме* могут осуществляться переключения и выполнения нескольких задач, предназначенных для режима виртуального МП 8086.

Блок управления страницами действует на более низком уровне, по сравнению с сегментированием. Разбиение на страницы возможно только в защищенном режиме. Каждый сегмент делится на страницы фиксированного размера по 4 Кбайта каждая.

Блок магистрального интерфейса реализует циклы обмена с памятью, сопроцессором, контроллерами. Обмен осуществляется с помощью 32-й ШД, 34-й ША и 16-разрядной й шиной управления. Особенность шины данных является возможность динамического изменения ее разрядности. За один цикл могут быть переданы 8, 16 или 32 бита (сигналы BE(0 – 3)# - Byte Enable – указывают на используемые шины данных). Здесь значок # означает, что активным сигналом является низкий уровень (инверсный сигнал). Структура и состав регистров процессора Intel 80386 показаны на рисунке 5.2.

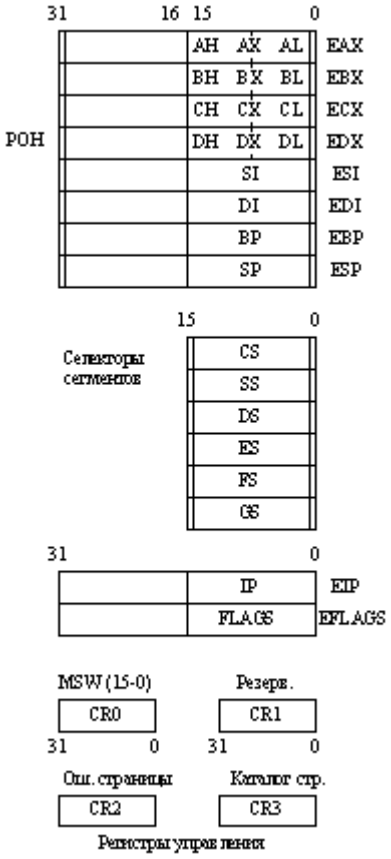


Рисунок 5.2 –Регистры процессора 80386

Набор регистров включает:

- РОН;
- сегментные регистры;
- указатель команд и регистр флагов;
- регистры управления;
- регистры адреса системы;
- регистры отладки;
- регистры тестирования.

Все 16 – разрядные регистры МП 8086 и 80286 содержатся в 32 – разрядных регистрах МП 80386. Состав сегментных регистров расширен за счет введения регистров дополнительных сегментов данных FS и GS.

Содержимое регистров определяется текущей задачей, т.е. регистры автоматически загружаются новыми значениями при переключении задач. РОН поддерживают работу с 1, 8, 16, 32 и 64 битовыми операндами. Адреса имеют размер 16 и 32 бита. Младшие 16 разрядов этих регистров доступны отдельно при использовании имен AX, BX и т.д. При операциях с байтами можно обращаться отдельно к младшему AL либо старшему AH байту.

В состав процессора входят также шесть 16–разрядных сегментных регистров CS, SS, DS, ES, FS и GS содержат значения *селекторов сегментов*, ука-

зывающих на текущие адресуемые сегменты памяти. С каждым из них связан программно-недоступный *регистр дескриптора сегмента*.

В *защищенном режиме* каждый сегмент может иметь размер от 1 байта до 4 Гбайт. В режиме реальных адресов максимальный размер сегмента 64 Кбайта. Селекторы CS указывает текущий сегмент кода команд, SS – стека, а селекторы DS, ES, FS и GS – текущие сегменты данных.

Каждый *регистр дескриптора* содержит 32 разрядный базовый адрес сегмента, 32 – разрядную границу сегмента, другие необходимые атрибуты (рис.5.3). Когда в регистр сегмента загружается новое значение селектора, содержимое соответствующего регистра дескриптора автоматически корректируется.

В режиме реальных адресов непосредственно обновляется только базовый адрес (он получается путем сдвига значения селектора на 4 разряда влево), т.к. максимальный размер и атрибуты сегмента в реальном режиме фиксированы.

В защищенном режиме корректируются все параметры. При каждом обращении к памяти регистр дескриптора сегмента, связанный с выбранным сегментом, автоматически вовлекается в эту операцию.

Базовый адрес сегмента (32 разрядный) становится компонентом вычисления линейного адреса. 32 – разрядная граница используется для операции контроля размера. Атрибуты проверяются на соответствие типа памяти.

Рисунок 5.3 – Регистры селекторов и дескрипторов процессора 80386

Имя 32–разрядного флагового регистра EFLAGS. Кроме флагов C, P, A, Z, S, T, I, D, и O добавлены новые флаги:

- IOPL – уровень привилегии ввода – вывода (биты 12 – 13);
- NT – вложенная задача (бит 14);
- RF – флаг итога (бит 16) используется в пошаговом режиме при отладке. Если бит установлен, то любая ошибка отладки в следующей команде игнорируется;
- VM – виртуальный режим 8086 (бит 17).

В процессоре имеется четыре 32 – разрядных регистра управления CR0, CR1, CR2 и CR3, используемые для фиксации общего состояния процессора, в частности:

- CR0 – регистр управления машины. Младшие 16 разрядов называют словом состояния (MSW). Другие биты используются для управления страницами, переключения задач, эмуляции сопроцессора, управление сопроцессором, включение защиты;
- CR1 – зарезервирован;
- CR2 – линейный адрес ошибки страницы. Содержит адрес, который вызвал последнюю ошибку страницы;
- CR3 – базовый адрес каталога страниц.

В состав процессора входят также регистры системных адресов, которые служат для обращения к элементам, управляющих механизмом сегментации памяти:

GDTR – регистр таблицы глобальных дескрипторов;
IDTR – регистр таблицы дескрипторов прерываний;
LDTR – регистр таблицы локальных дескрипторов;
TR – регистр задачи.

Кроме того, МП содержит 10 регистров тестирования, два из которых зарезервированы. Эти регистры определяют точки останова и показывают текущее состояние МП при остановах (R6), а также используются при тестировании памяти.

5.2. Страничная организация памяти

Страничная организация памяти обеспечивает более эффективное заполнение памяти по сравнению с сегментной, однако, требует дополнительного времени и специальных аппаратных средств для преобразования адресов. Страничная организация имеет место, если в регистре управления CR3 бит 31 имеет значение PG=1.

Страница (*кадр страницы*) – это блок в 4 Кбайт физической памяти. Страницы начинаются на границах 4 Кбайт областей памяти и фиксированы по размеру. Они не имеют непосредственного отношения к логической структуре программы. При страничной организации *сегмент* разбивается на отдельные *разделы*, число которых может достигать $2^{10}=1024$ (рисунок 5.4). Раздел может содержать 1024 страницы объемом по 4 Кбайта каждая. Границы страниц жестко фиксированы.

Адрес страницы указывает физический начальный адрес и его младшие 12 байтов всегда равны 0. Таким образом, начальные адреса страниц имеют значения от 00000000H до FFFFF000H. Начальные адреса страниц данного раздела хранятся в соответствующей *таблице страниц*, содержащейся в памяти.

Обращение к этой таблице производится с помощью *каталога*, в котором содержатся *адреса таблиц* для всех *разделов*. В *каталоге* страниц адрес кадра страницы – это адрес таблицы (раздела) страниц. В таблице страниц адрес кадра страницы является адресом кадра той страницы, который содержит нужный операнд в памяти.

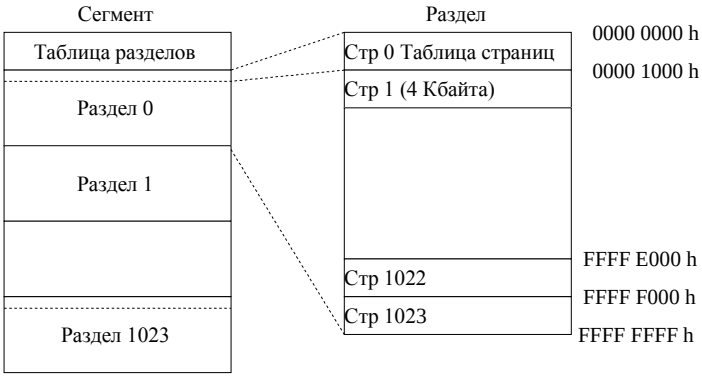


Рисунок 5.4 – Схема страничной организации памяти

Таблица страниц – это массив 32 – разрядных спецификаторов страниц. Таблица также является страницей и занимает 4 Кбайта или до 1 Кбайта 32 – разрядных кодов (слов). Каждое слово содержит базовый адрес страницы (12 бит), а также 12 битовых полей используемых для защиты страниц (бит пользователь/супервизор, чтение/запись, бит присутствия, бит обращения и др.).

Таким образом, страницы, содержащие отдельные фрагменты программ или данных, могут быть рассеяны по разным частям памяти, а их размещение определяется содержанием каталога разделов и таблиц страниц. При этом границы страниц и сегментов могут не совпадать.

Страничная организация памяти обеспечивает более эффективное использование памяти по сравнению с сегментной, однако, требует дополнительного времени и специальных аппаратных средств для преобразования адресов. *Линейный* 32 – разрядный адрес при этом является исходной информацией для формирования физического адреса с помощью *каталога разделов* и *таблиц страниц*. Базовый физический адрес сегмента загружается из дескриптора сегмента по его селектору в регистр CR3.

Линейный адрес при страничной организации рассматривается как совокупность трех полей (рисунок 5.5):

Поле **TABLE** (Раздел) включает разряды A31 – A22 линейного адреса, указывает относительный адрес таблицы страниц выбираемого раздела в каталоге.

Поле **PAGE** (A21 – A12) задает относительный адрес требуемой страницы раздела.

Поле **BYTE** (A11 – A0) содержит относительный адрес выбираемого на странице байта.

Каталог занимает одну страницу памяти, где для каждого из 1024 возможных разделов содержатся 32 – разрядные указатели входа в таблицу страниц этого раздела. Каждая из **таблиц страницы** также занимает одну страницу, где для каждой из 1024 страниц раздела даются 32 – разрядные указатели входа в нее.

Содержимое регистра управления CR3 задает старшие 20 разрядов адреса (A31 –A12) ячейки памяти, содержащий указатель входа в таблицу страниц (оглавления) раздела, т.е. адрес начала каталога раздела сегмента.

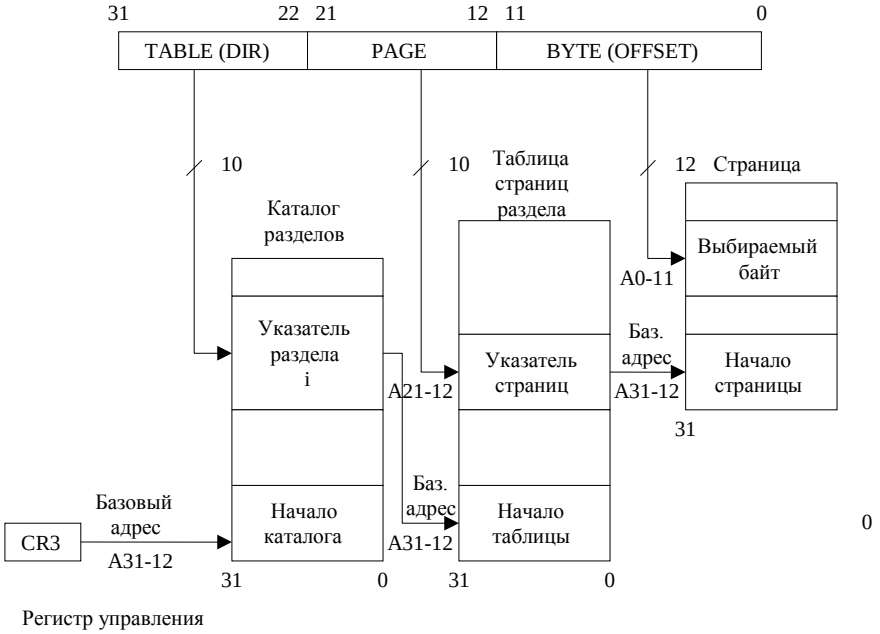


Рисунок 5.5 – Обращение к памяти при страничной организации

Разряды A11 – A2 адреса этой ячейки составляют относительный адрес, содержащегося в поле TABLE линейного адреса. Старшие из дескриптора 20 разрядов указателя задают базовый адрес, определяющий начало таблицы страниц.

PAGE определяет разряды A11 – A2 адреса ячеек памяти, хранящей указатель входа в кадр выбираемой страницы. Этот указатель имеет такой же формат, как и указатель входа в таблицу. Его старшие 20 разрядов служат базовым адресом, определяющим адрес начала страницы (первого байта). Добавление к базовому адресу поля BYTE, позволяет получить физический адрес выбираемого байта.

Таким образом, при страничной организации памяти 32 – разрядный (физический) адрес формируется как сумма базового адреса задаваемого указателем входа в кадр страницы, и относительного адреса, содержащегося в поле BYTE линейного адреса.

Помимо базового адреса указатели входа в таблицу страниц и кадр страницы содержат дополнительную информацию, определяющую порядок их использования. Эта информация задается значениями определенных битов, которые:

- разрешают или запрещают обращение к соответствующему разделу или странице (бит A0);
- определяют право доступа к соответствующим разделам или страницам для программ пользователя, имеющих минимальные привилегии (бит A1);
- фиксируют обращение к данному разделу или странице – запись или чтение (бит A5);
- фиксируют процесс записи на данную страницу (бит A6);

- определяют время последнего обращения к данному разделу или странице. Эта информация используется для определения разделов и страниц, подлежащих замене из внешней памяти (биты A11 – A9).

При каждом обращении к оперативной памяти аппаратными средствами выполняются следующие действия:

1) на основании начального адреса таблицы страниц (содержимое регистра адреса таблицы страниц), номера виртуальной страницы (старшие разряды виртуального адреса) и длины записи в таблице страниц (системная константа) определяется адрес нужной записи в таблице,
2) из этой записи извлекается номер физической страницы,
3) к номеру физической страницы присоединяется смещение (младшие разряды виртуального адреса).

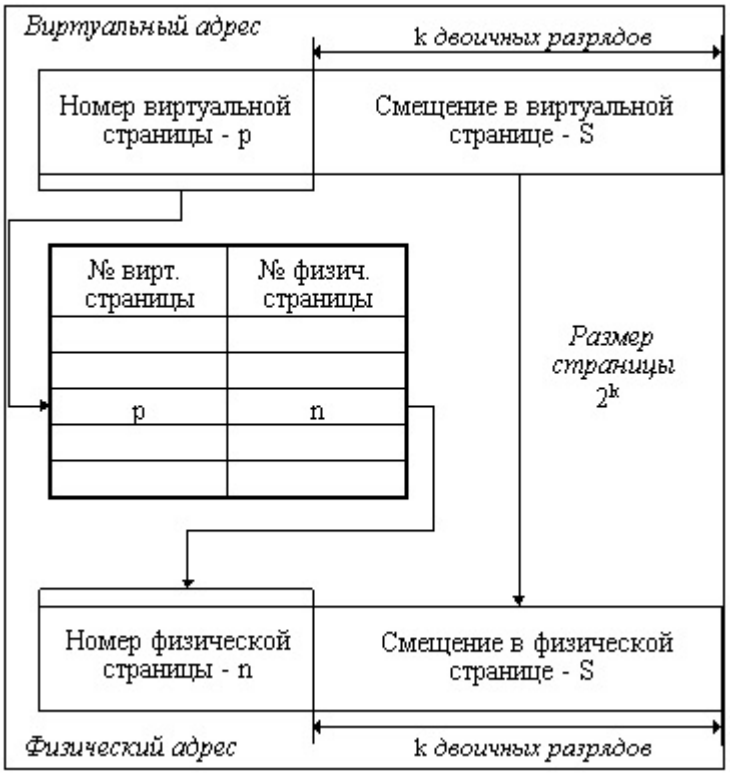


Рис.5.5а - Схема преобразования виртуального адреса в физический при страничной организации памяти

В 32-разрядных процессорах последующих поколений был введен ряд существенных дополнений:

- введен внутренний кэш первого уровня (*Cashe Level* 1) размером 8 Кбайт единый для данных и команд и предусмотрены все необходимые средства для построения памяти с двухуровневым кэшированием, допускающей работу и в мультипроцессорных системах;

- повышена производительность локальной шины – введены пакетные циклы, позволяющие передавать очередное слово данных в каждом такте шины (а не через такт, как в обычном режиме);
- наряду с использованием) полной системы команд CISC (*Complete Instruction Set Computers*) применено RISC–ядро *(Reduced Instruction Set Computers)*, позволяющее наиболее часто встречающиеся команды выполнять за один такт, за счет чего средняя производительность МП по сравнению с первыми 32-рязрядными процессами удвоилась;
- в состав БИС процессора введен высокопроизводительный математический сопроцессор;
- увеличена очередь команд до 16 байт;
- введены буферы отложенной записи;
- расширены средства тестирования (введены регистры TR3, TR4, TR5 для тестирования внутреннего кэша);
- добавлены средства тестирования процессора извне по интерфейсу JTAG;
- введено умножение тактовой частоты системной платы (кратность частоты может быть увеличена в 2, 2.5 и 3 раза);
- добавлены новые команды (BSWAP, XADD – для прикладных задач; CMPXHG, INVLD, WBINVD – для управления кэшем);
- введены раздельные 32-разрядные ШД и ША.

Шина адреса позволяет адресовать 4 Гб физической памяти (00000000-FFFFFFFFh) в защищенном режиме и 1 Мб из области младших адресов в реальном режиме. При выполнении инструкций ввода-вывода процессор адресуется к области в 64 Кб пространства ввода вывода (00000000-0000FFFFh).

Линии **A[31:2]** идентифицируют адрес с точностью до двойного слова, а в пределах этого слова сигналы **BE[0:3]** непосредственно указывают, какие байты используются в данном цикле (BE0 – 1 байт; BE3 – четвертый).

Шина данных D[31:0] допускает как 32-разрядный режим обмена, так и 16-и 8-разрядный, в зависимости от состояния входных сигналов BS16# и BS8#. Сигналы BS16# или BS8# вводятся внешней схемой, если текущий цикл адресуется к 16- или 8-битному устройству, подключенному к младшей части шины данных, BS8# имеет более высокий приоритет. Для связи с системной шиной, имеющей возможность подключения 8-, 16- и 32-битных устройств (шины ISA и EISA) и полную шину адреса, включающую сигналы A0, A1 и SBHE#, недостающие сигналы генерируются из сигналов BE [0:3].

Каждый байт шины данных имеет бит паритета DP[3:0]. Схемы паритета генерируют корректные контрольные биты в циклах записи, а в циклах чтения в случае ошибки паритета только вырабатывается сигнал ошибки на выходе **PCHK#,** никак не влияющий на работу процессора. Он может использоваться внешними схемами по усмотрению разработчика системной платы. Контроль четности (число единиц в байте вместе с битом паритета должно быть четным) выполняется для всех байт, участвующих в конкретном цикле.

Тип цикла локальной шины определяется сигналами **M/IO#, D/C#, W/R#** и **LOCK#** во время активности сигнала **ADS#** .

Для управления шинными циклами процессор имеет входные сигналы готовности RDY# и BRDY#. Сигнал BRDY# вырабатывает устройства, поддерживающие пакетный режим обмена. На возможность пакетного продолжения начатого цикла процессор указывает выходным сигналом BLASTS.

5.3. Суперскалярные и мультискалярные микропроцессоры

В современных микропроцессорах широко используется принцип конвейерного выполнения отдельных элементарных операций. Конвейеризация внутренних процессов позволяет выполнять команду за каждый процессорный цикл. Процесс выполнения команды можно разделить на следующие фазы:

1) **ВК- выборка команды** (*Instruction Fetch*): течение этой фазы по адресу команды она извлекается из ОЗУ (или кэша) в буфер команды, а программный счетчик увеличивается;
2) **ДК- декодирование**: на основании кода операции вырабатываются (устройством управления) сигналы, управляющие операционным блоком;
3) **ВА**- вычисление адреса операнда;
4) **ВО-выборка операнда**(ов): чтение операнда(ов) из регистров процессора или памяти;
5) **ИК-исполнение команды** в АЛУ;
6) **ЗР-запись результата** в регистр(ы).

В зависимости от команды отдельные фазы могут отсутствовать либо обойдены. Деление машинной операции на отдельные фазы осуществляется специальном блоке, называемом конвейером команд (*Pipeline*).

Дальнейшее внедрение принципов конвейеризации привело к появлению класса *суперскалярных* микропроцессоров. Их отличительной особенностью является возможность выполнения нескольких команд *параллельно*, т.е. за один процессорный цикл. Такой режим выполнения программы стал возможным благодаря наличию в процессорах нескольких исполнительных устройств.

Есть два подхода к отображению внутреннего параллелизма обработки данных на архитектурном уровне в системе команд. Первый подход состоит в том, что никакого указания на параллельную обработку внутри процессора система команд не содержит. Такие процессоры относятся к классу *суперскалярных*.

Второй подход - открывает все возможности параллельной обработки. В специально отведенных полях команды каждому из параллельно работающих обрабатывающих устройств предписывается действие, которое устройство должно совершить. Такие процессоры называются *процессорами с длинным командным словом* (**VLIW**).

Типичный суперскалярный процессор выбирает команды и исследует их по мере выполнения. Исследование проводится с целью выявления и обработки

команд перехода, идентификации типа команды для ее дальнейшего направления на соответствующий исполнительный блок или в буфер памяти. Выполняются также некоторые действия для смягчения зависимостей по данным, например переименование регистров. VLIW процессор возлагает на компилятор статическую реализацию тех функций, которые в суперскалярном процессоре выполняются динамически.

Такты→	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Команда 1	ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔								
Команда 2		ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔							
Команда 3			ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔						
Команда 4				ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔					
Команда 5					ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔				
Команда 6						ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔			
Команда 7							ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔		
Команда 8								ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔	
Команда 9									ВК ↔	ДК ↔	ВА ↔	ВО ↔	ИК ↔	ЗР ↔

Основная идея, определяющая развитие суперскалярных микропроцессоров, состоит в построении возможно большего количества параллельных структур при сохранении традиционных последовательных программ. Это означает, что компиляторы и аппаратура микропроцессора сами, без вмешательства программиста, обеспечивают загрузку параллельно работающих функциональных устройств микропроцессора.

В соответствии с моделью последовательного программирования, программы пишутся в предположении, что команды будут выполнены в том же

порядке, в каком они представлены в программе. Однако с целью достижения большей эффективности современные процессоры пытаются выполнять несколько команд одновременно и в некоторых случаях в порядке, отличном от их исходной последовательности в программе. Это переупорядочение может быть выполнено в трансляторе и (или) в аппаратных средствах во время выполнения. Суперскалярные и VLIW-процессоры принадлежат классу архитектур, которые используют параллельность уровня команды (**ILP**).

ILP-процессоры и компиляторы обычно преобразуют полностью упорядоченное множество команд исходной программы в частично упорядоченное множество, структурированное зависимостями по данным и управлению (командам). Зависимости по управлению (которые проявляются как переходы по условию) представляют главное препятствие высокопараллельному выполнению потому, что эти зависимости должны быть установлены прежде, чем будут выполнены все последующие команды.

Текст последовательной программы, представленной на языке высокого уровня, компилируется в машинный код, отражающий статическую структуру программы, т. е. упорядоченное множество команд (инструкций) в памяти компьютера. Процесс выполнения программы с конкретными наборами входных данных может быть представлен динамической структурой программы, т. е. множеством последовательностей инструкций в порядке их исполнения.

Повысить степень параллелизма программы можно путем изменения соответствующим образом ее статической или динамической структуры. Поскольку статическая структура программы однозначно соответствует ее исходному тексту (в предположении неизменности компилятора), то изменение статической структуры сводится к модификации исходного кода, что, в общем случае, не всегда возможно. Динамическая же структура программы может быть изменена при неизменной статической структуре. И главной целью такого изменения должно быть повышение степени параллельного исполнения команд.

Допустимые границы преобразования динамической структуры программы задают существующие на множестве инструкций отношения: зависимость по управлению и зависимость по данным. При описании архитектур суперскалярных процессоров часто используется модель *окна* исполнения. При исполнении программы микропроцессор как бы продвигает по статической структуре программы окно исполнения. Команды в окне могут исполняться параллельно, если между ними нет зависимости.

Для устранения зависимостей, вызванных командами переходов, используется метод предсказания, позволяющий извлекать и условно исполнять команды предсказанного перехода. Если позднее обнаруживается, что предсказание было сделано верно, то результаты условно исполненных команд принимаются. Если предсказание было ошибочным, состояние процессора восстанавливается на момент принятия решения о выполнении перехода.

Команды, помещенные в окно исполнения, могут быть зависимы по данным. Эти зависимости обусловлены использованием одних и тех же ресурсов памяти (регистров, ячеек памяти) в разных командах. Поэтому для правильного

исполнения программы необходимо использование этих ресурсов в предписываемом программой порядке.

Все виды зависимостей по данным могут быть классифицированы по типу ассоциаций: RAR - "чтение после чтения", WAR - "запись после чтения" и WAW - "запись после записи", RAW - "чтение после записи". Некоторые из зависимостей по данным могут быть устранены. RAR, по сути дела, соответствует отсутствию зависимостей, поскольку в данном случае порядок выполнения команд не имеет значения. Действительной зависимостью является только "чтение после записи" (RAW), так как необходимо прочитать предварительно записанные новые данные, а не старые.

Лишние зависимости по данным появляются в результате "записи после чтения" (WAR) и "записи после записи" (WAW). Зависимость WAR состоит в том, что команда должна записать новое значение в ячейку памяти или регистр, из которых должно быть произведено чтение. Лишние зависимости появляются по нескольким причинам: не оптимизированный программный код, ограничение количества регистров, стремление к экономии памяти, наличие программных циклов. Важно отметить, что запись может быть произведена в любой свободный ресурс, а не только тот, который указан в программе.

После удаления лишних зависимостей по управлению и данным команды могут исполняться параллельно. Формирование расписания параллельного выполнения команд возлагается на аппаратные средства микропроцессора. Это расписание учитывает существующие зависимости между командами и имеющиеся функциональные модули процессора.

Типовая архитектура суперскалярного микропроцессора включает:

- блок выборки команд и предсказания переходов;
- блок декодирования команд и анализа зависимостей между ними;
- блок переименования и диспетчеризации;
- блоки регистров и обрабатывающих устройств с плавающей и фиксированной точками;
- блок управления памятью;
- блок упорядочения выполненных команд.

Ниже рассмотрены основные приемы повышения быстродействия в суперскалярных микропроцессорах.

Предварительная выборка команд и предсказание переходов. Поскольку при суперскалярной обработке необходимо извлекать из памяти несколько команд за один такт для загрузки параллельно работающих функциональных модулей, повышенные требования предъявляются к пропускной способности интерфейса микропроцессор-память. В современных микропроцессорах применяются многоуровневые раздельные кэш-памяти данных и команд.

Для уменьшения потерь процессорных циклов, связанных с промахами при обращении к кэш-памяти в случае выполнения команд ветвления, в состав системы кэширования введены средства предсказания переходов, основное

назначение которых - повысить вероятность наличия в кэш-памяти требуемой команды.

Исполнение условных ветвлений состоит из следующих этапов:

- распознание команды условного ветвления;
- проверка выполнения условия перехода;
- вычисление адреса перехода;
- передача управления, в случае перехода.

На каждом этапе используются специальные приемы повышения производительности.

1. Для быстрого декодирования используются либо дополнительные биты в поле команды, либо *преддекодирование* команд при выборе из кэш - памяти команд.

2. Часто, когда команда уже выбрана в кэш, условие перехода еще не вычислено. Чтобы не задерживать поток команд в данном случае используется предсказание перехода по одной из нескольких возможных схем. Некоторые предсказатели используют статическую информацию из двоичного кода программы или специально выработанную компилятором. Например, определенные коды операций чаще вырабатывают ветвление, чем другие коды, или ветвление более вероятно (при организации циклов), или компилятор может устанавливать флаг, указывающий направления перехода. Может также использоваться статистическая информация, полученная при трассировке программы.

Другие предсказатели используют динамически формируемую информацию в процессе исполнения программы. Обычно это информация, касающаяся истории выполнения данного ветвления, сохраняемая в таблице ветвлений или в таблице предсказаний ветвлений. Таблица предсказания ветвлений организуется по ассоциативному принципу, подобно кэш-памяти, ее элементы доступны по адресу команды, ветвление которой предсказывается. В некоторых реализациях элемент таблицы предсказания ветвления является счетчиком, значение которого увеличивается при правильном предсказании и уменьшается при неправильном. При этом значение счетчика определяет преобладающее направление ветвлений.

В момент определения действительного значения условия ветвления, вносится изменение в историю ветвления. Если предсказание было неверным, то должна инициироваться выборка правильных команд. Результаты команд, которые были условно выполнены, должны быть аннулированы.

3. Для определения адреса ветвления обычно требуется выполнить целочисленное сложение, прибавляющее к текущему значению счетчика команд смещение, заданное в поле команды ветвления. И хотя это не требует дополнительных циклов для обращения к регистрам, ускорение вычисления адреса может быть достигнуто благодаря использованию буфера, содержащего ранее использованные адреса переходов.

Если требуется осуществить смену значения счетчика команд, то необ-

ходим, по крайней мере, один такт для распознания команды ветвления, модификации счетчика команд и выборки команды по заданному значению счетчика команд. Эти задержки вызывают пустые такты в конвейерах процессора. Более сложные решения используют буферы, содержащие наборы команд для двух возможных результатов ветвлений.

Возможно также использование "отложенных переходов", когда одна или несколько команд после команды ветвления выполняются безусловно.

Декодирование команд, переименование ресурсов и диспетчеризация. В этой фазе определяются существенные зависимости (RAW) по данным между командами и преодолеваются несущественные (WAW, WAR), производится распределение команд по буферам команд функциональных устройств.

При декодировании команды создается одна или несколько упорядоченных троек, каждая из которых включает: 1) исполняемую операцию, 2) указатели на операнды, 3) указатель на место помещения результата.

Для преодоления лишних WAR и WAW зависимостей, возникающих в результате ограниченности логических ресурсов (ячеек памяти, регистров), используется механизм динамического отображения определяемых текстом программы логических ресурсов на физические ресурсы микропроцессора. При данном подходе с одним логическим ресурсом может быть связано несколько значений в различных физических ресурсах, каждое из которых соответствует значению логической величины в один из моментов времени последовательного выполнения программы.

Когда команда создает новое значение для логического регистра, физический ресурс, в который помещается это значение, получает имя. Последующие команды, использующие это значение, снабжаются именем физического ресурса. Данная процедура называется *переименованием регистров*.

Для этого в микропроцессоре создается физический файл регистров размером больше логического. При необходимости переименования из списка свободных физических регистров берется один и ему сопоставляется соответствующее логическое имя. Если список свободных регистров пуст, диспетчеризация команд приостанавливается до момента появления свободных физических регистров.

Рассмотрим пример реализации данного способа переименования. Пусть требуется выполнить команду sub r3, г3, 5 (из значения регистра г3 вычесть константу 5 и поместить результат в регистр r3). Логические имена регистров начинаются со строчной буквы, а физические — с прописной. Пусть также в момент исполнения команды в таблице регистру r3 соответствует R 1. Первым регистром в списке свободных пусть является R2. Поэтому в поле результата команды sub r3, r3, 5 регистр r3 заменяется на R2. Исполнимая команда приобретает вид sub R2, R1, 5. Любая следующая за sub команда, использующая ее результат, должна использовать в качестве операнда R2.

Остается вопрос о возвращении физических регистров в список свободных

после того, как из них считаны данные в последний раз. Один из способов связывает счетчик с каждым физическим регистром. Счетчик увеличивается при каждом переименовании операнда в командах, использующих этот физический регистр. Соответственно при использовании операнда значения счетчика уменьшается на 1. При достижении счетчиком нуля физический ресурс должен быть переведен в список свободных.

Исполнение команд. После формирования для каждой команды упорядоченных троек, состоящих из кода операции, физических операндов - источника и результата, и размещения их в буферах, наступает фаза динамической проверки готовности значений операндов для исполнения команды.

В идеале команда готова к исполнению, как только готовы ее входные операнды. Однако есть ряд ограничений, связанных с доступностью физических ресурсов, таких как исполнительные устройства, коммутаторы и порты регистровых файлов (или переупорядочивающего буфера). Для организации окна исполнения используются различные методы: одной очереди, многих очередей или метод резервирующей станции.

Если имеется одна очередь, то переименование регистров не требуется, так как доступность значений операндов может отмечаться битом резервирования, сопоставленным каждому регистру. Регистр резервируется, когда модифицирующая его команда назначается на исполнение. И регистр освобождается, когда заканчивается исполнение команды. Если для команды ресурсы не были зарезервированы, то она приостанавливает свое исполнение.

В методе многих очередей каждая очередь организуется для команд одного типа. Например, очередь команд с плавающей точкой или очередь команд работы с памятью.

Третий метод предполагает использование резервирующей станции, состоящей из совокупности элементов, каждый из которых содержит позиции для размещения кода операции, наименования первого операнда, самого первого операнда, признака доступности первого операнда, наименования второго операнда, самого второго операнда, признака доступности второго операнда и наименования регистра результата. Когда команда завершает исполнение и вырабатывает результат, то наименование результата сравнивается с наименованиями операндов в резервирующей станции.

Если в резервирующей станции обнаруживается команда, ждущая этого результата, то данные записываются в соответствующую позицию и устанавливается признак их доступности. Когда у команды доступны все операнды, инициируется ее исполнение. Резервирующая станция следит за доступностью операндов. Когда команда при диспетчеризации попадает в резервирующую станцию, все готовые операнды из регистрового файла переписываются в поля этой команды. Если все операнды готовы, команда исполняется. Иногда резервирующая станция содержит не сами операнды, а указатели на них в регистровом файле или переупорядочивающем буфере.

Завершение выполнения команды. Завершающей фазой исполнения команды является фаза изменения состояния процессора в соответствии с выполненной командой. Назначение этой фазы - сохранение последовательной модели исполнения программы, при реальном параллельном выполнении отдельных команд и условном выполнении команд ветвления.

Мультискалярные процессоры используют *агрессивную стратегию* выполнения кода с целью извлечения параллелизма уровня команд из последовательной программы, представленной на языке высокого уровня. В соответствии с данной стратегией программа разбивается на *совокупность задач* с помощью программных и аппаратных средств. Задача – это часть программы, выполнению которой соответствует непрерывная область динамической последовательности команд (например, часть базисного блока, базисный блок, множество базисных блоков, одиночная итерация цикла, полный цикл, обращение к функции, и т.д.).

На каждом шаге обхода программы мультискалярный процессор назначает одну задачу на один из процессорных элементов (ПЭ) для выполнения, без учета фактического содержания задачи, и продолжает обход. Задача назначается для выполнения некоторому процессорному элементу, передачей ему начального значения программного счетчика. Множество инициированных таким образом задач выполняется параллельно на процессорных элементах, результатом чего является выполнение множества команд за один процессорный такт.

Каждый из совместно используемых процессорных элементов выбирает и выполняет команды, принадлежащие выделенной ему задаче. Значения разделяемых процессорными элементами регистров копируются в каждый ПЭ. Результат модификации содержимого регистров динамически направляется множеству параллельных ПЭ в соответствии с генерируемыми компилятором масками.

Доступ к памяти осуществляется *спекулятивно* (условно, по предположению) без знания последовательности предшествующих команд загрузки или сохранения. Обращение к данным производится параллельно многими ПЭ, а обработка приостанавливается только в случае истинной зависимости данных.

5.4. Архитектура суперскалярных процессоров типа Pentium

Рассмотрим особенности архитектуры суперскалярных процессоров на примере процессоров Pentium III и IV. В этих процессорах наряду с внутренним КЭШем первого уровня (L1-кэш) размером 64 Кбайт (32 Кбайт для данных + 32 Кбайт для команд), введен кэш второго уровня (L2-кэш) объемом 512 и более Кбайт.

Ядро процессора Pentium III состоит из двух частей: декодера и RISC–ядра (рисунок 5.7). Декодер получает из кэша команд ассемблерные команды, представляющие собой полный комплект команд компьютера (CISC-команды) и

преобразует их в микрооперации сокращенного набора команд (RISC-команды). В декодере имеется три преобразователя кодов.

RISC–ядро выполняет полученные микрооперации (мкоп) параллельно несколькими блоками обработки (АЛУ) в произвольном порядке и выдает результат обработки в порядке, заданном исходной ассемблерной программой. Pentium III обладает *десятью* блоками обработки, которые объединены в 5 групп. За один машинный такт на блоки обработки одновременно может быть подано максимум 5 микроопераций, причем на каждую их групп не более одной микрооперации. Первые две группы содержат по одному ММХ-АЛУ и блок генерации адреса. Кроме того, в них имеется вычислительные блоки для мультимедийных команд с плавающей точкой (ММХ пл.тчк). Оставшиеся три блока обработки служат для выполнения операций обращения к памяти: чтения и записи операндов или промежуточных результатов.

Операнды располагаются (с большой вероятностью) в двухпортовом кэше данных. Для одновременной записи в кэш адреса и данных блоки генерации адреса и передачи данных выполнены раздельно и работают параллельно.

Устройство функционирует следующим образом. CISC–декодер загружает x86-команды без предварительной обработки в 32-байтный буфер команд. Из этого буфера команды извлекаются предварительным декодером (предекодером), который определяет длину команды и группу, к которой она принадлежит. Кроме этого предекодер выделяет команды ветвления и заносит их адреса в буфер адресов ветвления емкостью 512 слов. Если обнаружена команда ветвления, то буфер команд заполняется командами, адреса которых определены блоком предсказания.

За один такт машинного цикла процессора на три преобразователя кодов ПК могут быть поданы максимум три x86-команды. Задачей ПК является преобразование x86-команд в RISC–микрооперации. Преобразователи кодов отличаются тем, что два из них обрабатывают только «простые» команды (MOV, INR,…) и генерируют одну RISC –операцию. Любая микрооперация имеет фиксированную длину 118 бит.

Третий преобразователь предназначен для «сложных» команд и на каждую x86-инструкцию генерирует от 1 до 4-х микроопераций. Однако среди сложных команд имеются такие, которые для их выполнения требуют целую микропрограмму. Такие команды преобразуются в блоке микропрограммного управления.

За один такт из трех ПК на RISC–ядро процессора могут быть подано максимум 6 микроопераций. Они предварительно заносятся в 6-словный буфер микроопераций и оттуда подаются в командный пул (буфер динамически выделяемой памяти).

После того, как по соответствующим статусным битам установлено завершение команды, происходит обратное переименование «внутренний регистр - архитектурный регистр». Затем блок завершения команд восстанавливает результаты выполненных команд в порядке, установленном исходной программой.

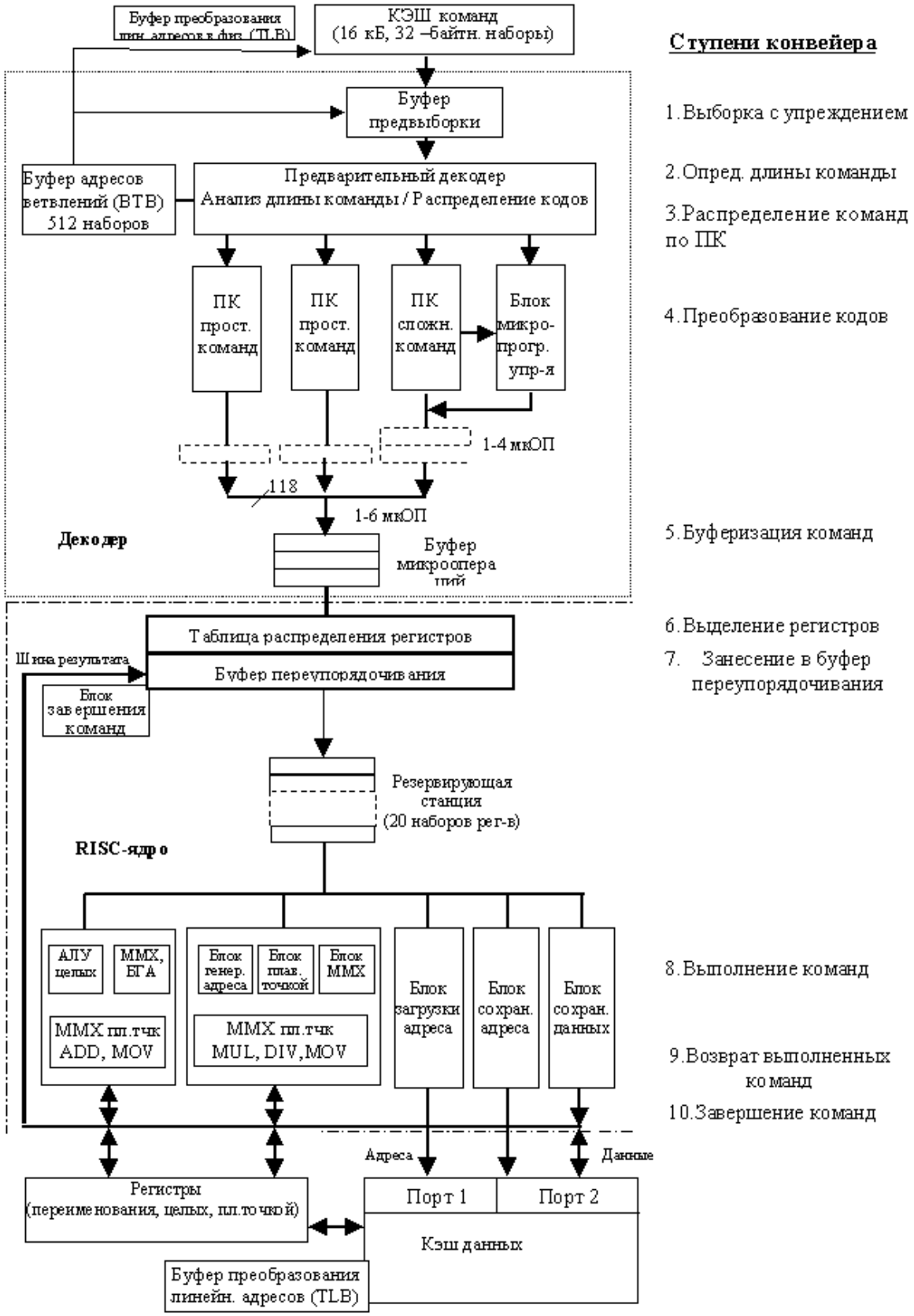


Рисунок 5.7 – Структурная схема процессора Pentium III

Для быстрого преобразования виртуальных адресов в физические в процессоре аппаратно реализован буфер быстрого преобразования адресов **TLB**

(*Translation look-aside buffer*). Он состоит из двух частей: буфера преобразования адресов команд (Instruction TLB) и буфера преобразования адресов данных (Data TLB).

Одним из недостатков такой архитектуры процессоров является то, что при выполнении программных циклов происходит преобразование одних и тех же ассемблерных команд в RISC-команды в каждом цикле. Это приводит к снижению производительности. Отличие архитектуры Pentium 4 от Pentium III состоит в том, что внутренний кэш команд размещен за блоками преобразования команд и в нем помещаются уже преобразованные команды (RISC-микрооперации). Вследствие этого исключаются дополнительные непроизводительные преобразования и повышается быстродействие процессора в целом. Обобщенная структурная схема процессора Pentium 4 изображена на рисунке 5.8.

Кроме этого, отличительными особенностями Pentium 4 от его предшественников являются:

1) введение гиперконвейера (Hyper Pipeling), позволяющего размещать и выполнять одновременно 20 микроопераций;
2) усовершенствование динамического исполнения позволяет в единицу времени выполнять в 3 раза больше микроопераций;
3) частота процессора превысила 3 ГГц, а системной шины 800 МГц;
4) в два раза повышена скорость выполнения операций в АЛУ целых чисел;
5) введено дополнительно 144 новых мультимедийных команд.

В процессоре P-4 большинство операций целочисленной арифметики выполняется в устройствах, работающих на двойной частоте (Double Pumped ALU). Таких устройств два. Латентность этих операций (время выполнения в зависимых цепочках) составляет всего 0.5 такта. Для того чтобы обеспечить такую низкую задержку каждое их этих двух 32-битных арифметико-логических устройств реализовано в виде двух отдельных 16-битных блоков, обрабатывающих соответственно младшие и старшие 16 разрядов операндов. Из-за возможного переноса разрядов из младшей половины слова блок обработки старшей половины работает с временным сдвигом на полтакта. Таким образом, старт-стопное время выполнения операции составляет целый такт (два полутакта). Однако выполнение новой операции, зависящей от результатов текущей, начинается немедленно после получения младшей половины слова. Это правило распространяется также на логические операции, команды пересылки и чтения из памяти.

Таким образом, операция обращения к памяти начинает выполняться, не дожидаясь готовности старших 16 разрядов адреса — поиск данных в L1-кэше начинается на основе младших 16 разрядов. Благодаря этому, полное время считывания данных из кэша составляет всего 2 такта. Операции сдвигов обрабатываются отдельно, в так называемом медленном устройстве. Их задержка (латентность) составляет 4 такта, а темп выполнения — 1 операция за такт. Также в медленном устройстве исполняются целочисленные умножения, деления и другие редкие операции.

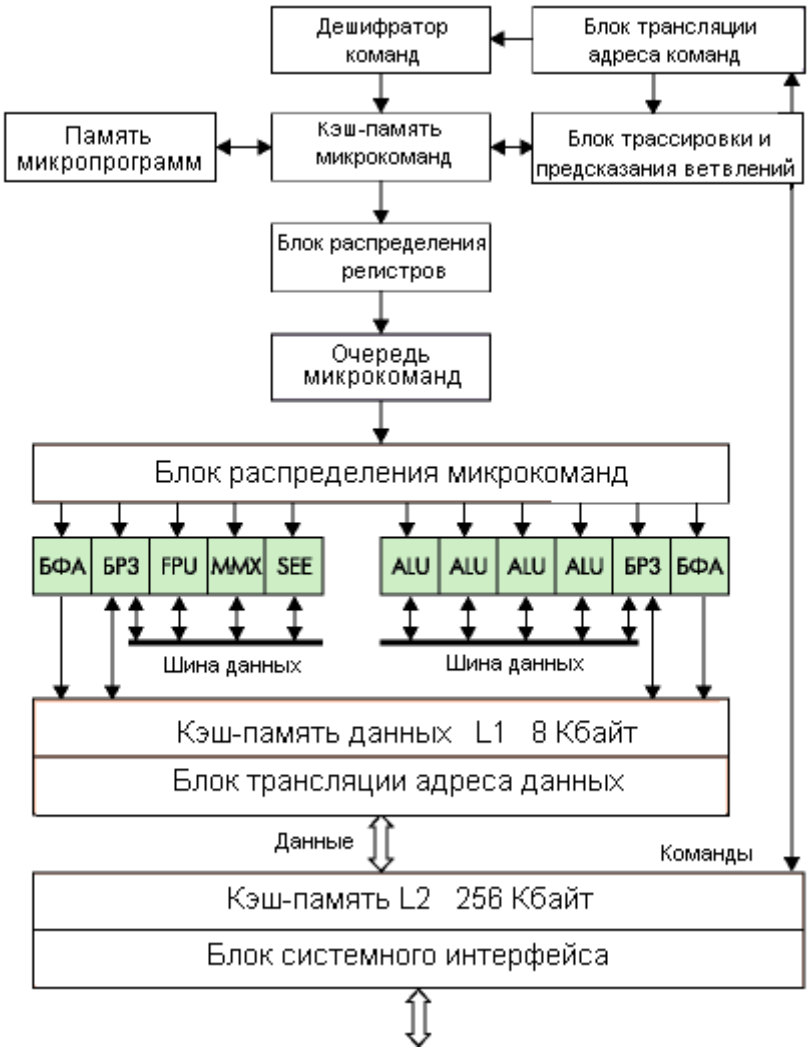


Рисунок 5.8 – Обобщенная структура процессора Pentium 4

6. Архитектура 64-разрядных и многоядерных процессоров

6.1. Общая характеристика 64-разрядных процессоров

Преимуществом 64-битных процессоров над своими 32-битными аналогами является расширение адресного пространства, увеличение разрядности и увеличение числа регистров общего назначения.

Расширенное 64-битное адресное пространство теоретически позволяет процессору работать с 16 экзабайт (2^{64}) физической памяти в рамках плоской модели организации. И хотя современные 64-разрядные процессоры на практике могут обеспечить доступ лишь к 1 терабайту (2^{40}) памяти, данный показатель всё равно уже значительно превосходит возможности 32-битной адресации. Увеличение объема доступной памяти в свою очередь, даёт возможность исключить или сильно сократить количество крайне медленных операций по подкачке данных с диска.

Увеличение числа и разрядности регистров позволит процессору одновременно работать с большими участками памяти, более эффективно работать с переменными и массивами, передавать аргументы функций в регистрах вместо использования для этой цели стека.

Стоит помнить, что для получения реального прироста производительности на 64-разрядном процессоре необходимо транспонировать программу с применением 64-битной версии компилятора, учитывая изменение модели данных (новые размерности типов). Запуск неадаптированного для 64-битной платформы приложения наоборот может, в зависимости от особенностей архитектуры используемого процессора, привести к существенным потерям в производительности.

Наибольший же прирост производительности от перехода на 64-разрядную платформу получат приложения, манипулирующие большими массивами данных — это системы управления баз данных, программы для работы с цифровым мультимедиа сообщениями, прикладные научные приложения. Прирост производительности для ПО данного класса может составить сотни процентов**.**

64-битное расширение классической 32-битной архитектуры IA32 было предложено в 2002 году компанией AMD (первоначально называлось x86-64, сейчас — AMD64) в процессорах семейства K8. Спустя некоторое время компанией Intel было предложено собственное обозначение — EM64T (Extended Memory 64-bit Technology). Но, независимо от названия, суть новой архитектуры одна и та же: разрядность основных внутренних регистров 64-битных процессоров удвоилась (с 32 до 64 бит), а 32-битные команды x86-кода получили 64-битные аналоги. Кроме того, за счет расширения разрядности шины адресов объем адресуемой процессором памяти существенно увеличился.

6.2. Процессор Athlon 64

Первыми серийными процессорами, основанными на 64-разрядной архитектуре, стали чипы семейства K8 фирмы AMD под названием *Opteron*, официально анонсированные 22 апреля 2003 года. **AMD Opteron**, ориентированные на применение в серверах и рабочих станциях, стали первыми 64-разрядными процессорами, способными выполнять не только инструкции AMD64, но и традиционные 32-разрядные инструкции x86, без использования каких бы то ни было специальных режимов эмуляции. Именно сохранение обратной совместимости, позволяющей говорить о возможности плавного перехода с 32- на 64-разрядное программное обеспечение, и является главным, но не единственным достоинством всех процессоров семейства K8. Через полгода были анонсированы процессоры **Athlon 64/Athlon FX**, ориентированные на применение в настольных ПК и ноутбуках.

Структура процессора Athlon 64 показана на рисунке 6.1.

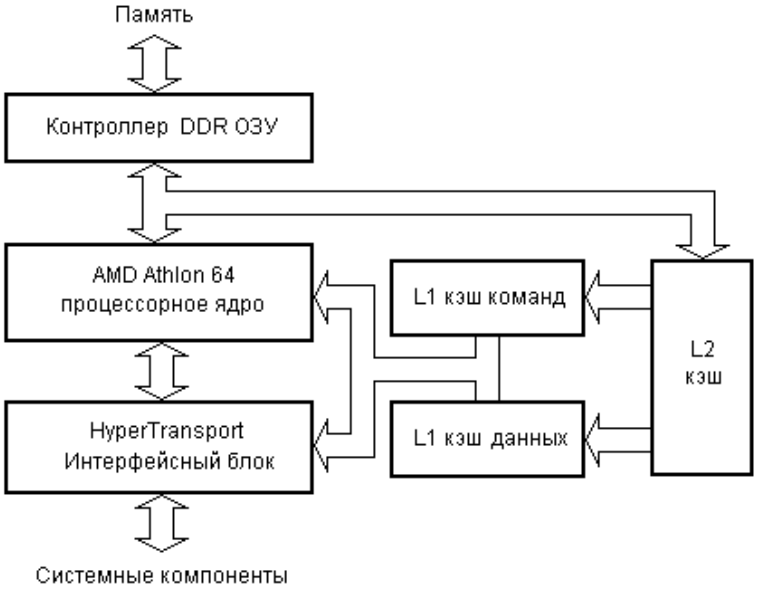


Рисунок 6.1 – Структура процессора Athlon 64

По сравнению с предшествующей моделью Athlon XP, новые процессоры обладают целым рядом качественных нововведений. Процессоры семейства Athlon 64 поддерживают **новый набор 64-разрядных инструкций AMD64,** наряду с 32-разрядными инструкциями x86. Это позволяет Athlon 64/Athlon 64 FX выполнять и современные 32-битные, и новые 64-битные приложения.

Контроллер памяти DDR 400 интегрирован непосредственно в ядро процессора. До сих пор на настольных платформах и AMD, и Intel применяли схему, которая предусматривала размещение контроллера памяти в микросхеме северного моста чипсета. AMD перенесла контроллер на ядро, стараясь увеличить скорость обмена данными между процессором и системной памятью. Процессор Athlon 64 FX обладает двухканальным контроллером, в то время как "обычный" Athlon 64 - одноканальным. Увеличение **кэш-памяти второго**

уровня до 1 Мб дает возможность Athlon 64 и Athlon 64 FX обеспечивать большую производительность в широком круге приложений, в том числе архиваторах, играх, офисных пакетах. Оригинальная структура кэш-памяти, примененная AMD впервые еще в процессорах Athlon/Duron, позволяет хранить лишь одну копию данных в кэшах первого и второго уровней, что увеличивает общий эффективный объем кэш-памяти до 1152 Кб.

Кэш инструкций 2-х канальный частично-ассоциативный, имеет размер 64КВ. Кэш данных, также 2-х канальный и тоже частично-ассоциативный.

Для связи с другими системными компонентами и, в первую очередь, с микросхемами набора системной логики, ответственными за работу с AGP и другими устройствами, применена **универсальная шина HyperTransport**. Контроллер HyperTransport интегрирован на ядро процессора. Шина **HyperTransport (HT)**— это двунаправленная последовательно/параллельная компьютерная шина, с высокой пропускной способностью и малыми задержками. Она работает на частотах от 200 МГц до 2,6 ГГц. HyperTransport поддерживает автоматическое определение ширины шины, от 2-х битных линий до 32-х битных. Шина HyperTransport основана на передаче пакетов. Пакеты HyperTransport передаются по шине последовательно. Увеличение пропускной способности влечёт за собой увеличение ширины шины. HyperTransport может использоваться для передачи служебных сообщений системы, для передачи прерываний, для конфигурирования устройств, подключенных к шине и для передачи данных. Схема ядра процессора Athlon 64 показана на рисунке 6.2.

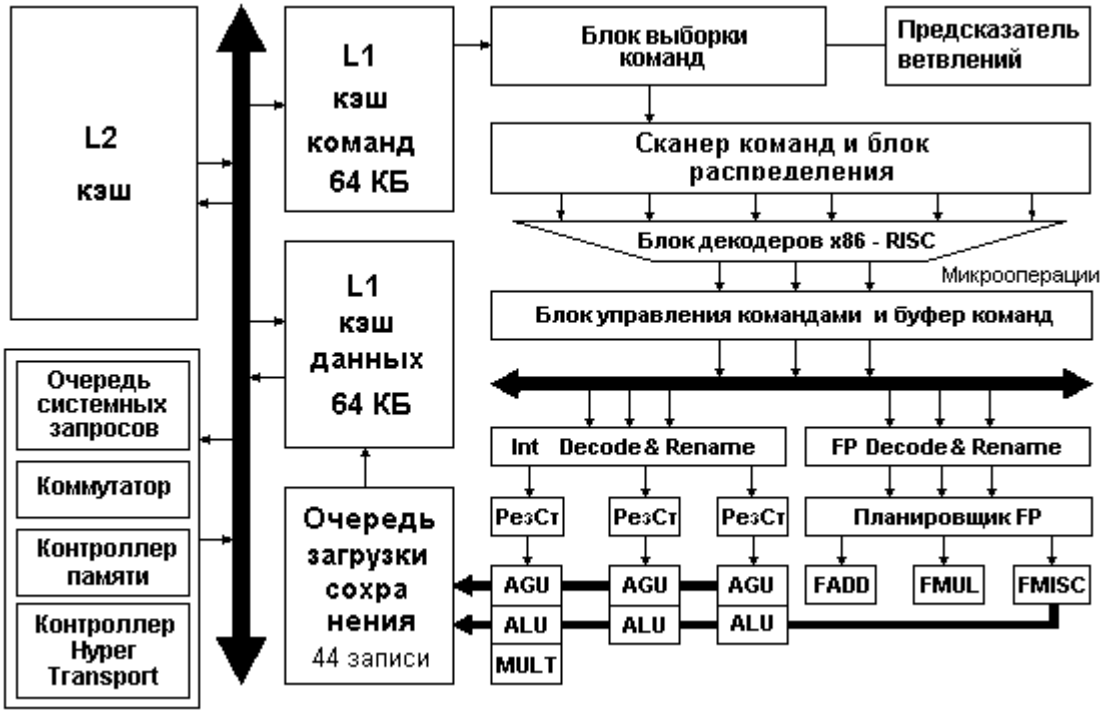


Рисунок 6.2 – Схема процессорного ядра Athlon 64

Блок декодеров превращает команды x86 переменной длины в RISC-микрооперации фиксированной длины. Причем за один такт **каждый** декодер

может обработать инструкцию длиной до **16 байт** и отправить планировщику на исполнение микрооперации, которые затем упаковываются в группы по три.

В ядре установлены три 8-входовые станции резервирования (РезСт). Все вместе они образуют 24-входовый планировщик заданий. Исполнительные устройства ядра располагают тремя целочисленными блоками ALU, 3-мя блоками генерации адресов и загрузки AGU, тремя блоками с плавающей точкой. Длина целочисленных конвейеров имеет 12 ступеней, FPU конвейеров — 17. Все данные 64-битовые. Арифметические и логические операции, кроме умножения, выполняются за один такт. На 32-битное умножение расходуется три такта и пять тактов на 64-битное умножение.

6.3. Многоядерные процессоры AMD и Intel

К началу 2005 года классические одноядерные процессоры практически полностью исчерпали резервы роста производительности за счет повышения рабочей частоты. Камнем преткновения стало не только слишком высокое тепловыделение процессоров, работающих на высоких частотах, но и проблемы с их стабильностью. Дальнейшее повышение производительности процессоров может быть достигнуто преимущественно за счет расширения концепции параллелизма работы, в частности перехода от параллельных цепочек конвейеров до параллельно работающих процессорных ядер. Первыми представителями процессоров, реализовавших на практике данную концепцию, стали двуядерные процессоры.

По сути своей двуядерный процессор почти не отличается от обыкновенной двухпроцессорной системы, в которой установлено два независимых процессора. В результате получают все преимущества двухпроцессорных систем без необходимости использования сложных и дорогих двухпроцессорных материнских плат. Однако в отличие от двухпроцессорных систем в двуядерных более остро стоит проблема использования общих ресурсов, доступ к которым нужно распределять между двумя ядрами.

С точки зрения программного обеспечения двуядерный процессор система рассматривает как два независимых. Естественно, что двуядерный процессор способен одновременно выполнять два потока инструкций, в то время как обычный CPU производит вычисления строго по очереди. В приложениях, оптимизированных под многопроцессорность, можно получить ощутимый — вплоть до двукратного — прирост производительности.

Число транзисторов в новых процессорах составляет около 230 млн.

Существуют две архитектуры двуядерных процессоров: фирмы *AMD* (**Athlon 64 X2**) и фирмы *Intel* **Pentium D** и **Pentium 4 Extreme Edition 840**.

При всей несхожести архитектур ядер, оба процессора имеют одну общую черту – одинаково реализованную двуядерность: **каждое ядро имеет собственный кэш второго** уровня (до 2048 Кбайт на каждое ядро у Pentium D 9xx и до 1024 Кбайт у Athlon 64 X2). В процессоре Pentium D для связи ядер используется внутренняя шина (передача данных между ядрами возможна и через

системную шину FSB). Такое решение не самое удобное в том случае, когда одному ядру надо получить данные, содержащиеся в КЭШе другого ядра. Ведь нередко ядра работают над одной и той же задачей, просто выполняя разные её части. Отсюда неминуемые задержки в доступе к информации, а также возможен простой одного из ядер по той причине, что его кэш полностью заполнен данными, которые всё ещё нужны второму ядру. Логичное решение этой проблемы – использование общей кэш-памяти. И именно таким образом работает двуядерный процессор, получивший название Core Duo.

Каждое из ядер Athlon 64 X2 также обладает собственным набором исполнительных устройств и выделенной кэш-памятью второго уровня; контроллер памяти и контроллер шины HyperTransport - общие (рисунок 6.3). А вот взаимодействие каждого ядра с разделяемыми ресурсами происходит посредством специального перекрестного коммутатора (Crossbar Switch) и интерфейса системных запросов (System Request Interface), в котором формируется очередь системных запросов (System Request Queue). На этом же уровне организовано и взаимодействие ядер, благодаря чему снимается дополнительная нагрузка на системную шину и шину памяти.

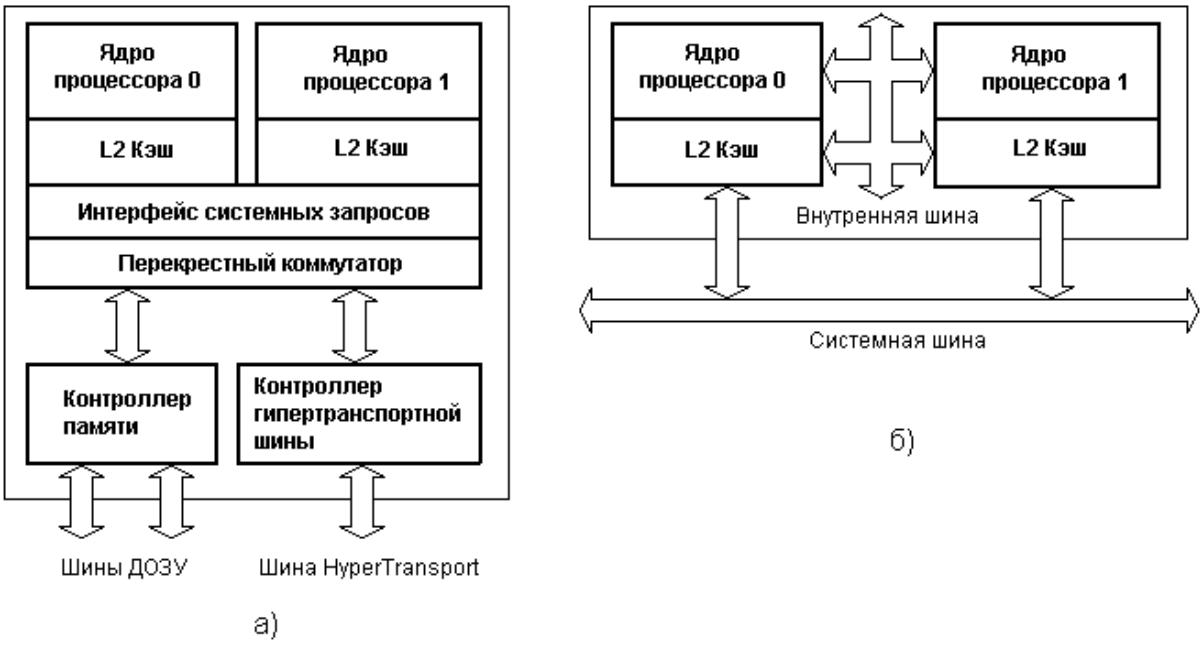


Рисунок 6.3 – Структура двуядерного процессора типа **Athlon 64 X2** (а) и **Pentium D9xx** (б)

Шина HyperTransport работает на частотах от 200 МГц до 2,6 ГГц (сравните с шиной PCI и её 33 или 66 МГц). Кроме того, она использует память DDR, что означает, что данные посылаются как по переднему, так и по заднему фронтам сигнала синхронизации. Это позволяет осуществлять до 5200 миллионов посылок в секунду при частоте сигнала синхронизации 2,6 ГГц; частота сигнала синхронизации настраивается автоматически.

HyperTransport поддерживает автоматическое определение ширины шины, от 2-х битных до 32-х битных линий. Полноразмерная, полноскоростная 32-

х битная шина в двунаправленном режиме способна обеспечить пропускную способность до 20800 МБ/с (2*(32/8)*2600), являясь, таким образом, самой быстрой шиной среди себе подобных. Шина может быть использована как в подсистемах с высокими требованиями к пропускной способности (оперативная память и ЦПУ), так и в подсистемах с низкими требованиями (периферийные устройства). Данная технология также способна обеспечить низкие задержки для других применений в других подсистемах.

Шина HyperTransport основана на передаче пакетов. Каждый пакет состоит из 32 разрядных слов, вне зависимости от физической ширины шины (количества информационных линий). Первое слово в пакете — всегда управляющее слово. Если пакет содержит адрес, то последние 8 бит управляющего слова сцеплены со следующим 32-битным словом, в результате образуя 40 битный адрес. Шина поддерживает 64 разрядную адресацию — в этом случае пакет начинается со специального 32 разрядного управляющего слова, указывающего на 64 разрядную адресацию, и содержащего разряды адреса с 40 по 63 (разряды адреса нумеруются, начиная с 0). Остальные 32-х битные слова пакета содержат непосредственно передаваемые данные. Данные всегда передаются 32-х битными словами, вне зависимости от их реальной длины (например, в ответ на запрос на чтение одного байта по шине будет передан пакет, содержащий 32 бита данных и флагом-признаком того, что значимыми из этих 32 бит являются только 8).

Пакеты HyperTransport передаются по шине последовательно. Увеличение пропускной способности влечёт за собой увеличение ширины шины. HyperTransport может использоваться для передачи служебных сообщений системы, для передачи прерываний, для конфигурирования устройств, подключенных к шине и для передачи данных.

Операция записи на шине бывает двух видов — *posted* и *non-posted*. Posted-операция записи заключается в передаче единственного пакета, содержащего адрес, по которому необходимо произвести запись, и данные. Эта операция обычно используется для обмена данными с высокоскоростными устройствами, например, для DMA передачи. Non-posted операция записи состоит из посылки двух пакетов: устройство, инициирующее операцию записи, посылает устройству-адресату пакет, содержащий адрес и данные. Устройство-адресат, получив такой пакет, проводит операцию записи и отсылает устройству-инициатору пакет, содержащий информацию о том, успешно ли произведена запись. Таким образом, posted-запись позволяет получить максимальную скорость передачи данных (нет затрат на пересылку пакета-подтверждения), а non-posted-запись позволяет обеспечить надёжную передачу данных (приход пакета-подтверждения гарантирует, что данные дошли до адресата).

Шина HyperTransport поддерживает технологии энергосбережения, а именно ACPI. Это значит, что при изменении состояния процессора (C-state) на энергосберегающее, изменяется также и состояние устройств (D-state). Например, при отключении процессора НЖМД также отключаются.

Электрический интерфейс HyperTransport/LDT — низковольтные дифференциальные сигналы (Low Voltage Differential Signaling (LVDS)), с напряжением 2,5 В.

Первые двухядерные процессоры AMD, в отличие от аналогичных представителей Intel, вначале не были предусмотрены для работы с DDR2. Отсутствие поддержки современных типов памяти с высокой пропускной способностью объяснялось тем, что компания в первую очередь стремилась сохранить совместимость Athlon 64 X2 с существующими платформами. Зато, в отличие от двуядерных процессоров Intel, для которых необходим новый чипсет, эти CPU могли устанавливаться на те же материнские платы, что и обычные Athlon 64. В современных многоядерных процессорах AMD реализована поддержка памяти типа DDR2.

Линейка процессоров Pentium D представлена тремя моделями: 820, 830 и 840 с частотами, соответственно, 2.8, 3.0 и 3.2 ГГц. Для повышенных требований разработана модель Pentium Extreme Edition 840, процессорные ядра которой работают на частоте 3,2 ГГц. Отличие экстремального двуядерного процессора от остальных заключается в разблокированном коэффициенте умножения и в присутствии технологии Hyper-Threading, которая отключена в моделях линейки Pentium D. Таким образом, Pentium Extreme Edition будет определяться операционной системой как четыре логических процессора.

Двухядерному процессору присущ весь набор современных технологий: поддержка 64-битных расширений EM64T, технология безопасности Execute Disable Bit, а также полный набор средств Demand Based Switching для управления тепловыделением и энергопотреблением.

Проблеме снижения тепловыделения двуядерным процессором уделяется значительное внимание. Введены средства, позволяющие «на лету» изменять тактовую частоту процессора и напряжение на ядре в зависимости от необходимой на данный момент производительности. В новом процессоре реализован и ряд других, не менее интересных и полезных энергосберегающих технологий, таких, например, как Dynamic Power Coordination. Суть этой технологии состоит в том, что ядра могут независимо друг от друга менять энергопотребление в зависимости от текущей нагрузки на процессор. В том числе, возможна ситуация, когда одно ядро работает, а другое находится в состоянии Deep Sleep, в котором потребление энергии близко к минимальному.

Фактически, получается, что второе ядро вовсе не означает двухкратного увеличения энергопотребления и тепловыделения, ведь оно работает только тогда, когда это действительно нужно. Компания Intel называет это качество Dual-Core Performance on Demand – «производительность двуядерного процессора по требованию». Такое решение является весьма рациональным: с одной стороны, в случае необходимости процессор может потреблять мало энергии, работая в «одноядерном режиме», а с другой стороны, способен мгновенно перейти в режим высокой производительности, задействовав второе ядро.

Введено еще одна технология снижения энергопотребления, получившая название Dynamic Cache Sizing – «динамическое изменение размера кэш-

памяти». Суть этой технологии заключается в отключении простаивающих блоков кэш-памяти. Кроме того, если информация, содержащаяся в кэше, в течение какого-то времени не используется, то она переносится в оперативную память, а блоки кэша опять-таки отключаются. Учитывая, что объём кэша у ядра достаточно велик – целых 2 Мбайта – полностью он используется не так уж часто, особенно при невысокой нагрузке на систему, как это обычно и бывает при работе от аккумуляторной батареи.

Компания Intel отказалась от имени Pentium. Двуядерные процессоры фирмы Intel получили официальные имена Core Solo и Core Duo – одноядерный и двуядерный процессор соответственно.

Вслед за двуядерными процесорами фирмой Intel был выпущен четырёхядерный процессор, вышедший под кодовым именем Kentsfield, который представляет собой объединение двух кристаллов Core Duo, сделанное в единой процессорной упаковке. Промышленная модель Kentsfield получила название **Core 2 Extreme QX6700**.

В октября 2011 года появился шестиядерный процессор Intel Core i7-3960X на базе архитектуры Sandy Bridge-E, являющийся на сегодняшний день самым быстрым CPU от компании Intel для домашних пользователей. Тем временем AMD существенно доработала свой четырехядерный Phenom X4, увеличив объем кеш-памяти и освоив 45-нанометровый технологический процесс, а в апреле 2010 года анонсировала «шестиядерник» AMD Phenom II X6, который позволил не отпустить Intel слишком далеко вперед.

Одним из самых важных недостатков у шести- и восьмиядерных процессоров является внушительное энергопотребление, а значит, сильное тепловыделение и высокие температуры чипа при работе под нагрузкой. Производители борются с этим, осваивая все более «тонкие» технологические процессы и разрабатывая более совершенные схемы питания. Также тормозит массовое развитие «многоядерников» дефицит соответствующего программного обеспечения: большая часть потенциала микрочипа остается попросту нереализованной. Кроме того, себестоимость многоядерных процессоров пока обуславливает отнюдь не привлекательную для рядового пользователя цену, которая тоже сдерживает спрос.

В процессорах Intel Core i7 использована совершенно новая микроархитектура, но оставлен прежний рыночный бренд – Core. Четырехъядерные процессоры **Core i7** основаны на микроархитетуре под названием **Nehalem,** произведены по 45 нм технологическим нормам, на площади кристалла расположен 731 млн. транзисторов. Под эти чипы разработан новый сокет – **Socket B** (LGA 1366).

Основной принцип построения процессоров на основе **Nehalem** – модульность, которая позволит варьировать количество ядер и изменять оснащенность процессорной системы прочими блоками (в т. ч. графическими), в зависимости от предназначения и требуемой производительности.

Core i7, также как и процессоры AMD имеют встроенный контроллер памяти, что позволяет ядрам напрямую связываться с DDR модулями на систем-

ной плате, значительно ускоряя обмен данными между процессором и оперативной памятью. Встроенный контроллер памяти имеет максимальную пропускную способность 25,6 Гбайт/с. При этом, вместо традиционной двухканальной системы обмена использована трехканальная, с поддержкой модулей DDR3 1066 МГц. Структурная схема ядра процессора изображена на рисунке 6.4.

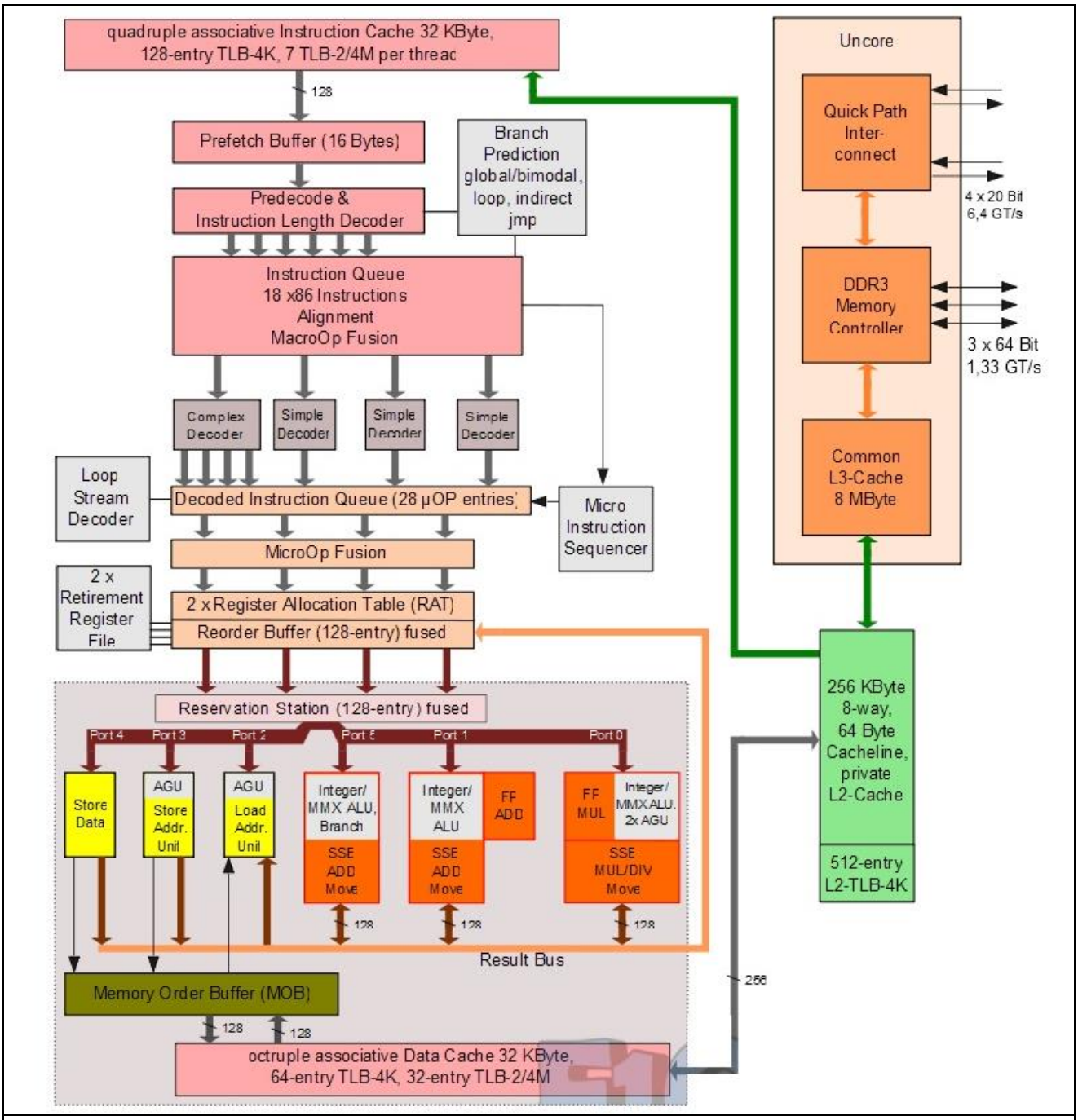


Рисунок 6.4 – Структурная схема ядра процессора Intel Core-i7

Традиционная процессорная FSB шина параллельного типа **Quad Pumped** заменена последовательным интерфейсом, который уже давно исполь-

зуется в процессорах AMD **Core i7**общается с элементами чипсета через последовательную шину с фирменным названием **QuickPath Interconnect (QPI)**.

В процессоре реализована технологиия многопоточности **SMT** (англ. **Simultaneous Multi-Threading**). Эта технология позволяет каждому из четырех ядер одновременно выполнять две задачи. Таким образом, для операционной системы новые процессоры становятся, как бы восьми ядерными, и это ускоряет работу многопоточных приложений, за счет одновременного выполнения большего количества задач. SMT является по сути реанимацией технологии Intel Hyper-Threading, разработанной некогда для одно ядерных процессоров.

7. Архитектура персональных компьютеров

В 1972 г. фирмой IBM была представлена совершенно новая конструкция персонального компьютера (ПК). Характерным ее признаком было модульное исполнение. Благодаря гибкости конструкции, возможности ее расширения, она оказалась победителем рынка персональных компьютеров.

Компьютер в отдельности сначала использовал ограниченный круг пользователей, в связи с тем, что его программирование требовало больших затрат на создание своих вычислений и управляющих программ. Популярность ПК резко подскочила после разработки фирмой Microsoft операционной системы для ПК. Эта ОС получила название MS – DOS (Микрософт - дисковая операционная система) и очень быстро распространилась во всем мире, превратившись в отдельную ветвь мирового рынка. Одновременно разрабатывался широкий спектр пользовательского программного обеспечения (ПО), без которого ОС не имеет смысла. Большое значение в расширении популярности ПК явилась разработка дискет, которые стали стандартными и совместимыми по структуре данных и форматам.

Первым IBM-PC был компьютер XT-класса (*eXtended Technologies* – расширяемая технология). В нем использован 16-битный процессор с 8-разрядной шиной данных. В первых XT PC использовался МП 8088 или 8086 фирмы Intel. Затем начали применяться процессоры фирмы NEC, улучшенной модификации 8086 с обозначением V20. Этот ПК работал с тактовой частотой 4.77 МГц, которая в так называемой Turbo-XT модификации была поднята до 15 МГц.

Следующей была разработана модель AT – серии (*Advanced Technologies* передовая технология). Ядром этой ПЭВМ был интеловский процессор 80286. Он располагал 16 – или 32–х разрядной внутренней ШД, и 16 – разрядной внешней ШД. Естественно в его состав может входить и 8 – разрядная шина ввода/вывода. Тактовая частота была поднята от 8 до 21 МГц. Это позволило существенно улучшить производительность ПК.

Затем появились 32-разрядные и 64-разрядные процессоры тактовые частоты которых превысили 3 ГГц, а на их базе были созданы высокопроизводительные компьютеры. На настоящее время наиболее перспективными являются компьютеры на основе двух- и многоядерных процессоров.

7.1. Обобщенная структурная схема компьютера

Структурная схема классического системного блока ПК типа IBM PC изображена на рисунке 7.1. Системный блок состоит из основной панели и расширяемой системной шины. Ядром IBM PC является *микропроцессор* фирмы Intel 80xxx, 8088 или 8086; затем 80x86, Pentium и т.д. Совместно с каждым процессором может быть включен *сопроцессор* соответствующего типа 8087, 80287 ... 80487. Наличие его не являлось обязательным. Для этого на основной (материнской – Mother Board) плате в ПК до 386 модели в непосредственной близости от ЦП, была установлена панелька (сокет), куда мог быть вставлен

сопроцессор. В последующих моделях сопроцессор интегрирован в БИС основного процессора.

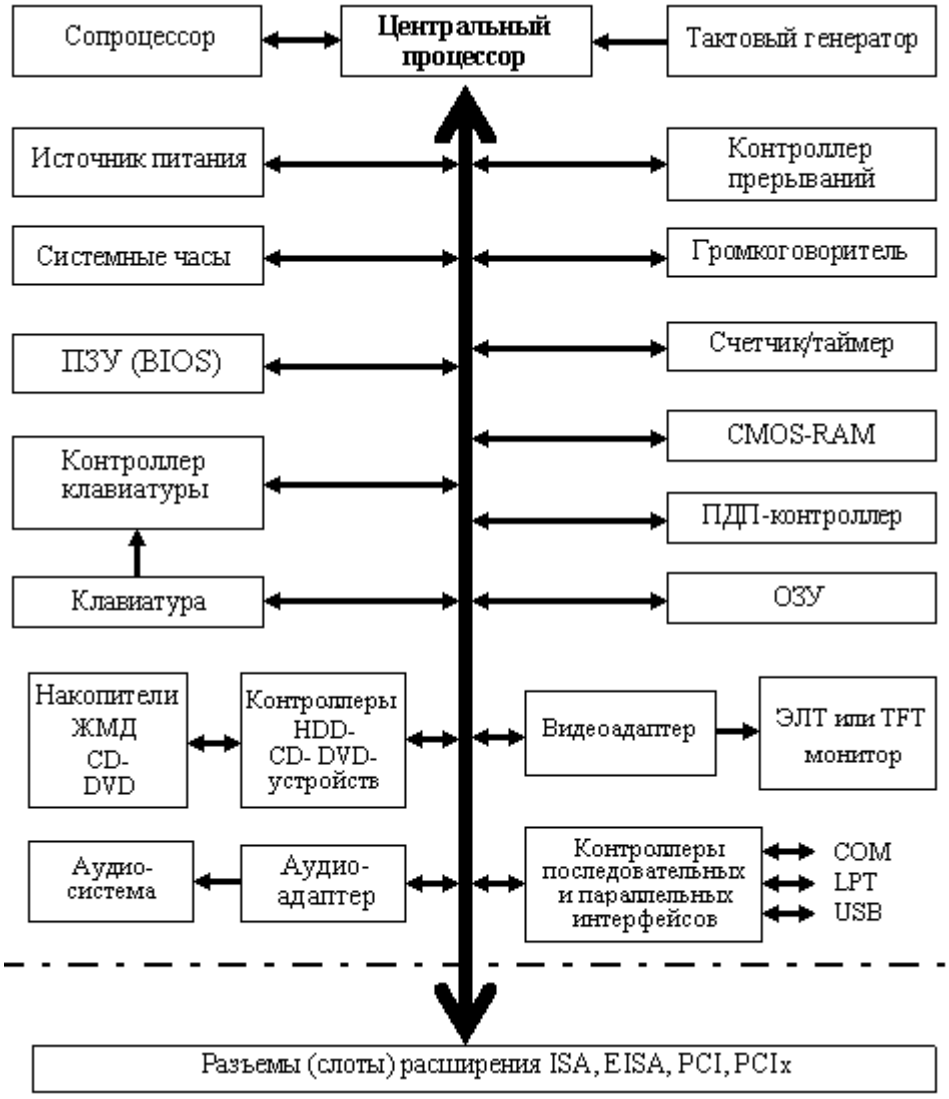


Рисунок 7.1 – Структурная схема персонального компьютера

Особенностью такого системного блока явилось то, что практически каждый функциональный узел компьютера выполнялся в виде одной или нескольких больших интегральных схем (БИС). В последующих модификациях системного блока большинство из функциональных модулей были интегрированы в несколько сверхбольших БИС.

Тактовый генератор определяет рабочую частоту ПК. Стандартный компьютер XT работал на частоте 4.77 МГц, AT 486 до 66 МГц, ПК на основе Pentium-процессоров работают с частотой свыше 3 ГГц.

BIOS (ПЗУ) – содержит базовую программу ввода/вывода. Она служит для выполнения следующих функций:

1) Автоматическое тестирование аппаратных компонентов при включении; Вызов блока начальной загрузки ДОС, т.е. сначала BIOS загружает с системного диска в память специальный блок начальной загрузки, а затем переда-

ет ему управление, а тот в свою очередь осуществляет загрузку других модулей ДОС;

Обслуживание системных вызовов или прерываний. *Контроллер прерываний.* В качестве контроллера использовались БИС 8259 (580ВН59). В XT – 1 шт, в AT – 2. В XT имелось 8 линий запроса на прерывание IRQ0 – IRQ7 и 8 дополнительных в PC – AT и последующих моделях IRQ8 – IRQ15.

Таймер 8253(54) (Отечественный аналог 1810ВИ54). Один из каналов таймера зарезервирован для задания импульсной последовательности для контроллера ПДП для регенерации динамической памяти. Второй канал осуществляет прерывание с частотой 18,2 Гц для программно организованных часов. Третий канал генерирует сигнал для управления (динамическим громкоговорителем – спикером).

В первых моделях ПК для задания различных режимов использовались мини – переключатели типа DIP (*Dual In-line Package*). В связи с большим многообразием конфигурации в AT комплексе ручная конфигурация больше не используется. Для задания параметров конфигурации и их хранения применяется CMOS – RAM. Для сохранения информации в ОЗУ используется питание от аккумулятора, питающего также системные часы. Из этой ИС загружается SETUP – информация.

Контроллер ПДП 8237 (1810ВТ37). В модели XT – 1 шт, в AT – 2 шт. Основное задание – регенерация динамической памяти и обмен данными между устройствами ввода/вывода и ОЗУ (НГМД или НЖМД - Винчестер).

ОЗУ. Используется ОЗУ динамического типа, сгруппированные в несколько банков. 8 – разрядная организация + дополнительный контрольный бит.

Контроллер клавиатуры. Для XT использовался параллельный порт 8255, а для AT микроконтроллер 8042. С помощью контроллера 8042 можно не только считывать данные, но и передавать приказы клавиатуре. Клавиатуры XT и AT различаются. Основное различие состоит в форматах передачи данных. Некоторые клавиатуры позволяют с помощью переключателя менять формат с целью использования их в обоих типах ПК.

Источник питания обеспечивает подачу напряжений +3,3; +5; +12; –5 и –12 V на ПК. Напряжение +12 V используется преимущественно приводными двигателями дисководов и вентиляторов, -5 и –12 применяются для дополнительного напряжения специальных микросхем. Стандартный источник питания XT выпускался мощностью 150 Вт, блоки питания AT имеют мощность 200 – 350 Вт, в последующих моделях мощность была увеличена до 500 Вт и выше. Типовые значения допустимых токов следующие: +12V (5,5A; 7,6A; 8,5A), +5V (15A; 20A; 22A), –5V, -12V (по 0,5A).

На системной плате компьютера имеется группа разъемов, позволяющая подключать к системной шине дополнительным сменные блоки, которые необходимы для управления работой различных устройств ЭВМ. Эти блоки носят название *контроллер* или *адаптер*. Следует заметить, что во многих современных ПК большинство контроллеров выполнены в виде одной или нескольких БИС, которые расположены на системной плате компьютера.

В состав первых ПК входил контроллер гибких магнитных дисков (ГМД) - *контроллер дисководов*. Он мог управлять двумя дисководами. Для управления жестким диском (ЖМД) применяется АТ-системный контроллер, состоящий из контроллера системной шины с дополнительной логикой.

Жесткие диски (ЖД) различаются размерами диска и емкостью. Диаметр диска указывается в дюймах (1 дм = 2,54 см). В старых типах ЖД имели диаметр 5.25”, в современных - 3.5”. Конструктивно во многих компьютерах контроллеры магнитных дисков располагались на одной плате расширения. В настоящее время они интегрированы в специальную БИС (Южный мост).

Разъемы (шины) расширения. Стандартный ПК обладал до 8-ми разъемами (*слотами*) расширения. Сюда могли вставляться различные цифровые и измерительные карты. В персональных компьютерах IBM XT с процессором 8086 или 8088 использовалась 8-битовая шина расширения.

Компьютеры IBM AT имели 16–битовую шину расширения, которая допускает вставлять в слоты 8–битовые платы. Эта шина расширения получила название **ISA** (*Industry Standard Architecture*). 16-битовые карты, выполняющие аналогичные функции с 8-битовыми, являются существенно более быстродействующими. Кроме 16 – разрядной ШД в ISA использовалась 24-разрядная ША, что позволяла адресовать 16 Мб памяти. $F_{T\ шины}$=8 МГц. Шина ISA являлась тормозящим фактором на пути передачи сигналов между ЦП и другими устройствами (видеоадаптером, ввода/вывода).

В качестве альтернативы была разработана шина расширенной промышленной стандартной архитектуры EISA (*Extended ISA*). Это 32-разрядная шина с нисходящей совместимостью с существующими платами расширения, т.е. с ней можно использовать 8- и 16-битовую плату ISA. Контакты на EISA – разъеме располагались один под другим. Они подключаются к более глубокому соединителю, чем ISA-платы.

PCI (*Peripheral Component Interconnect*) – 32-разрядная шина, является мостом между системной шиной процессора и шиной ввода/вывода, функционирующая со стандартной частотой 33 или 66 МГц и пропускной способностью 132 Мбайт/с. Разработана для компьютеров с процессором Pentium. Шина обычно содержит 2 или 4 слота расширения, соединенных с мостом системных шин. Шина поддерживает ведущие устройства, которые могут генерировать свои собственные циклы работы с шиной (IDE – контроллеры, MPEG – кодеры, сетевые контроллеры). Производительность PCI интерфейса была впоследствии значительно повышена за счет использования пакетных циклов обмена по шине.

При каждом обмене по шине (***транзакции***) участвуют два устройства — *инициатор обмена* (*Initiator или Master*) и целевое устройство — *устройство назначения* (*Target или Slave*). Каждая транзакция начинается фазой (циклом) адреса, за которой может следовать одна или несколько фаз данных. Для адреса и данных используются общие мультиплексированные линии AD. Четыре мультиплексированные линии C/BE[3:0] используются для кодирования команд в фазе адреса и разрешения байт в фазе данных. В начале транзакции инициатор по шине AD передает целевой адрес, а по линиям C/BE# информацию о типе тран-

закции (команде). Адресованное целевое устройство отзывается сигналом DEVSEL#, после чего инициатор может указать на свою готовность к обмену данными сигналом IRDY#. Значок # в обозначении сигнала означает, что активным уровнем является инверсное значение сигнала. Когда к обмену данными будет готово и устройство назначения, оно установит сигнал готовности. Данные по шине AD могут передаваться только при одновременном наличии сигналов готовности инициатора и устройства назначения.

Шина **PCI Express**, или PCIe, использующая программную модель шины PCI и высокопроизводительный физический протокол, основанный на последовательной передаче данных. В отличие от PCI, использующей для передачи данных общую шину, PCI Express функционирует по принципу пакетной сети с топологией типа звезда. Устройства PCI Express взаимодействуют между собой через коммутатор, при этом каждое устройство напрямую связано соединением типа точка-точка с коммутатором.

Кроме того, шиной PCI Express поддерживается горячая замена карт расширения, осуществляется контроль целостности передаваемых данных, управление энергопотреблением, гарантируется полоса пропускания (QoS);

Шина PCI Express (PCIe) предназначена на использование только в качестве локальной шины. Так как программная модель PCI Express во многом унаследована от PCI. Высокая пиковая производительность шины PCI Express позволяет использовать её вместо шин AGP и тем более PCI и PCI-X. Де-факто PCI Express заменила эти шины в персональных компьютерах. Связь между двумя устройствами PCI Express осуществляется по одной (x1) или нескольким (x2, x4, x8, x12, x16 и x32) двунаправленным последовательным линиям. Каждое устройство должно поддерживать соединение по крайней мере по одной линии (x1). Плата PCI Express помещается и корректно работает в любом слоте той же или большей пропускной способности (например, карта x1 будет работать в слотах x4 и x16);

Разработаны более производительные версии этой шины, в частности PCI Express 2.0 и PCI Express 3.0 со скоростью передачи 5 и 10 Гбит/ с соответственно. Кроме повышенной пропускной способности в шине предусмотрено динамическое управление скоростью передачи, оповещение о пропускной способности и ряд других функций.

В последующих моделях персональных компьютеров почти все функциональные узлы и контроллеры были интегрированы в две сверхбольших интегральные схемы (рисунок 7.2): системный контроллер и функциональный контроллер.

Для связи центрального процессора с северным мостом в 1990-х годах была разработана системная шина **FSB** (*Front Site Bus*). FSB разработана компанией Intel и впервые использовалась в компьютерах на базе процессоров Pentium. Частота работы шины FSB является одним из важнейших параметров работы ЭВМ и во многом определяет производительность всей системы. Обычно она в несколько раз меньше частоты работы процессора.

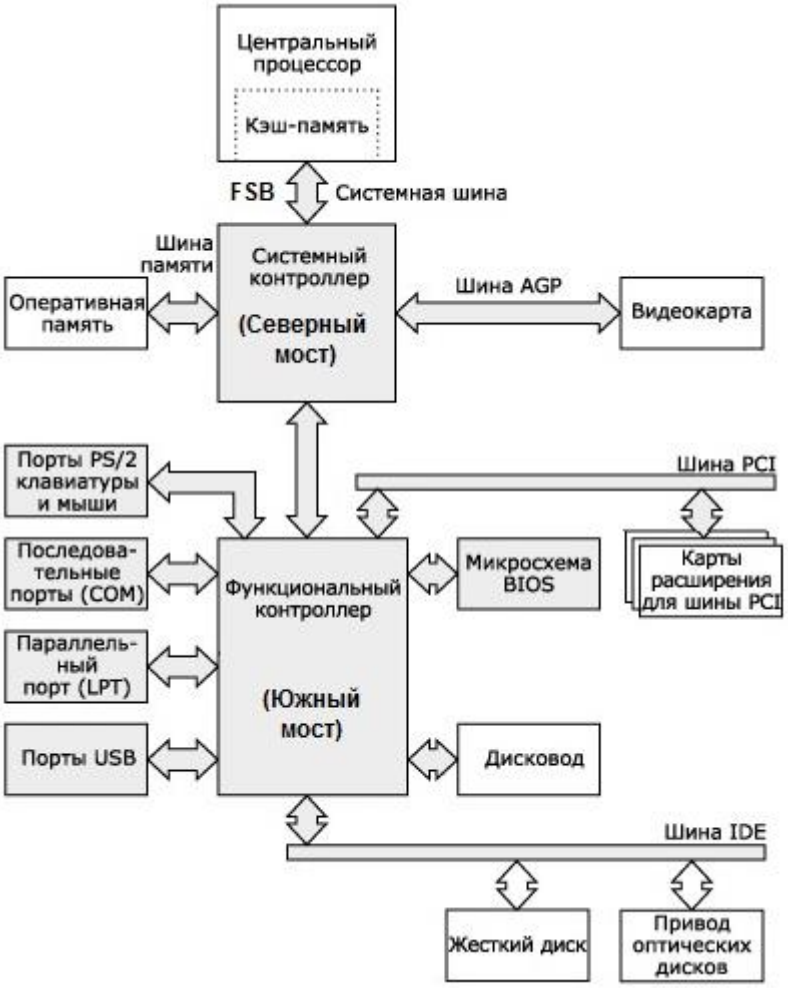


Рисунок 7.2 – Структура компьютера на основе системного и функционального контроллеров

Частоты, на которых работают центральный процессор и системная шина, имеют общую опорную частоту и в упрощенном виде рассчитываются, как Fп = F_0*k, где Fп – частота работы процессора, F_0-опорная частота, k – множитель. Обычно в современных системах опорная частота равняется частоте шины FSB.

Частота системной шины FSB постепенно возрастала с 50 МГц, для процессоров класса Intel Pentium и AMD K5 в начале 1990-х годов, до 400 МГц, для процессоров класса Xeon и Core 2 в конце 2000-х. При этом пропускная способность возрастала с 400 Мбит/с до 12800 Мбит/с.

Шина FSB использовалась в процессорах типа Atlon, Celeron, Pentium, Core 2, и Xeon вплоть до 2008 года. На данный момент эта шина вытеснена системными шинами DMI, QPI и Hyper Transport.

Видеоадаптер (видеокарта) в таких компьютерах сначала подключался к шине расширения PCI. В дальнейшем для повышения быстродействия для него была разработана специальная шина AGP (*Accelerated Graphics Port*).

Для повышения быстродействия обмена данными с оперативной памятью и освобождением места на кристалле центрального процессора, контроллер динамического ОЗУ был интегрирован в кристалл ЦП. Кроме того, параллельный

интерфейс с жестким диском PATA (IDE) и был заменен на последовательный **SATA** (*Serial ATA*, ATA - *AT Attachment*).

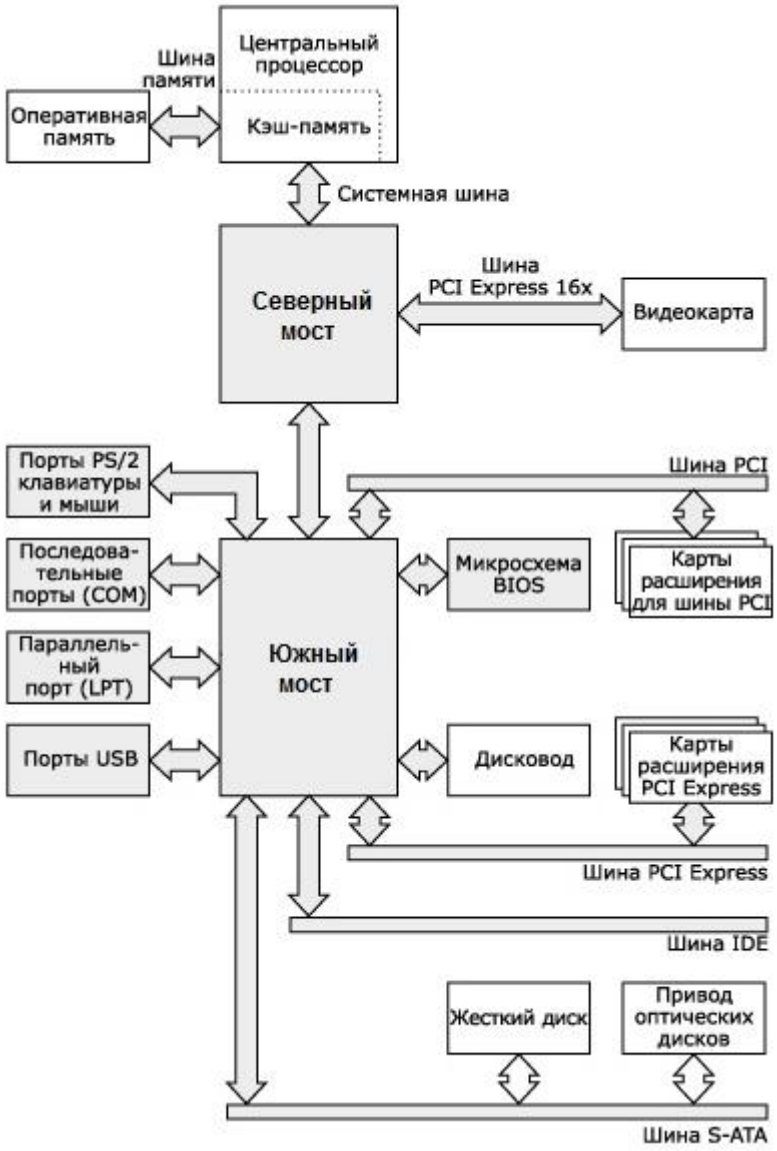


Рисунок 7.3 – Структура компьютера на основе последовательных шин и встроенного в процессор контроллера динамического ОЗУ

Несмотря на то, что последовательный интерфейс принципиально медленнее параллельного, интерфейс SATA работает быстрее стандарта IDE (*Integrated Drive Electronics* — встроенный интерфейс накопителей) за счет использования более высоких тактовых частот. Каждый SATA-накопитель подключается отдельным кабелем к отдельному разъему на материнской плате. Поэтому для данного интерфейса проблема конфликта Slave/Master отсутствует. Поскольку для каждого SATA-устройства полагается отдельный кабель, одновременно могут работать несколько устройств. Структура такого компьютера показана на рисунке 7.3.

В последних моделях многоядерных персональных компьютеров на основе

процессоров микроархитектуры Haswell с сокетом LGA 2011-v3 произошла дальнейшая интеграция системной логики и переход на новые, более производительные шины. Так как ряд функций северного моста реализован на кристалле центрального процессора, то на смену северному и южному мостов пришел однокристальный контроллер (сначала Intel P55 Express, затем X79 и X99).

Чипсет Intel X99 поддерживает до 14 портов USB из которых до 6 портов могут быть портами USB 3.0. Центральный процессор содержит 4-канальный контроллер ДОЗУ с поддержкой памяти типа DDR4, общий объем которой может достигать 64 ГБ. Кроме того, в процессор введен блок поддержки функций управления и системы BIOS. Упрощенная структурная схема такого компьютера показана на рисунке 7.4.

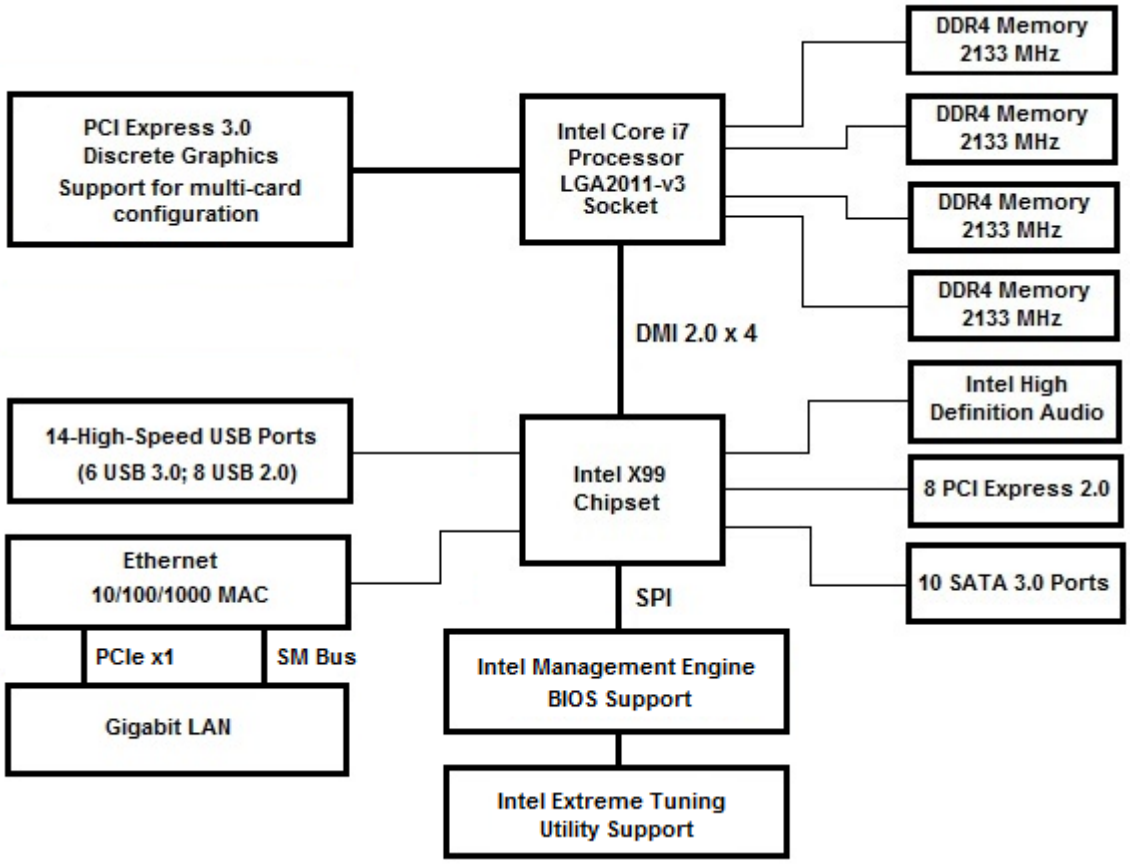


Рисунок 7.4 – Схема компьютера с однокристальным контроллером

7.2. Распределение адресного пространства ПЭВМ

Адресное пространство IBM – совместимых компьютеров, функционирующих под управлением дисковой операционной системы (DOS), логически можно разделить на три области (рис. 2.5):

1) стандартная память (CMA) *Conventional Memory Area*;
2) верхняя память (UMA) *Upper Memory Area*;
3) расширенная память (XMA) *Extended Memory Area*.

Весь объем памяти может быть разбит на 16 блоков адресного пространства по 64 КБ. Адреса таких блоков: 00000h – 0FFFFh, 1xxxx, 2xxxx и т.д.

Первые десять блоков RAM (сегменты от № 0 до 9) выделены для пользователя – так называемая **стандартная память.** Очевидно, что объем стандартной памяти не может превышать 640 Кбайт.

В диапазоне от 640 Кб до 1 Мб расположены стандартная и расширенная *видеопамять*, а также расширение ПЗУ и BIOS. Собственно BIOS занимает последние 64 КБ ячеек. Компьютеры AT – класса могут использовать расширенную RAM – область свыше 1 Мб.

Определенные области памяти имеют специальное назначение и используются отдельными командами и подпрограммами ОС. Это следующие области:

1) *Таблица векторов прерывания*, которая определяет место расположения прикладных программ по обработке этих прерываний. Она занимает 1024 байта, определяя тем самым 256 различных прерываний. Адреса от 0 до 400h.

Каждый адрес вектора состоит из 4-х байт. Адрес любого вектора находится путем умножения номера прерывания на 4.

Векторы в памяти расположены в обратном порядке: смещение (младший байт затем старший байт) и сегмент (младший байт и старший байт). Например, содержимое первых четырех ячеек E8 4E 9A 01 является сегментированным адресом 01 9A:4E E8.

В таблице векторов расположены три основных вида адресов. Они указывают адреса расположения:

- Базовой Системы ввода/вывода ПЗУ BIOS – ROM;
- Расположение основной памяти;
- Адреса DOS.

Некоторые адреса могут быть нулевыми, что означает, что эти адреса в данный момент не обрабатываются.

2) *Область данных BIOS.* Она занимает адреса от 00400h до 004FFh. В этой области хранятся разнообразные данные, используемые программами BIOS в процессе управления периферийным оборудованием, в частности, здесь размещается:

- входной буфер клавиатуры с указателями;
- адреса последовательных и параллельных портов;
- данные настройки видеосистемы (форма курсора, его нахождение на экране, текущий видеорежим и т.п.);
- ячейки для отсчета текущего времени;
- область межзадачных связей.

Область данных BIOS заполняется в процессе начальной загрузки компьютера и динамически модифицируется операционной системой по мере необходимости. Многие прикладные программы обращаются к этой области для чтения или модификации содержащейся в ней информации.

3) *Рабочая область DOS и Бейсика*, занимает 256 байтов в диапазоне адресов от 500h до 600h. В ней содержатся некоторые системные данные DOS.

<table>
<tr><td>Размер и адрес</td><td colspan="2">Название области памяти</td></tr>
<tr><td>до 15 Мб
10FFF0h</td><td>XMS Расширенная память</td><td>Расширенная память
(XMS-eXtended Memory Area)</td></tr>
<tr><td>64 Кб
100000h</td><td>HMA Высокая память</td><td>Высокая память
(HMA-High Memory Area)</td></tr>
<tr><td>128 Кб
E0000h</td><td>ПЗУ BIOS</td><td rowspan="6">Область верхней памяти
(UMA-Upper Memory Area)</td></tr>
<tr><td>64 Кб
D0000h</td><td>Верхняя память
(UMB-Upper Memory Block)</td></tr>
<tr><td>64 Кб
C0000h</td><td>ПЗУ - расширения BIOS</td></tr>
<tr><td>32 Кб
B8000h</td><td>Текстовый буфер EGA</td></tr>
<tr><td>32 Кб
B0000h</td><td>UMB Верхняя память</td></tr>
<tr><td>64 Кб
A0000h</td><td>Графический буфер EGA</td></tr>
<tr><td rowspan="3">640 Кб</td><td>Command.com</td><td rowspan="7">7FFF -9FFFh Stack DOS

Стандартная память

(CMA-Convertional Memory Area)</td></tr>
<tr><td>Свободная память для прикладных программ</td></tr>
<tr><td>Command.com
(резидентная часть)</td></tr>
<tr><td></td><td>Загружаемые драйверы</td></tr>
<tr><td>0700h</td><td>IO.SYS и MSDOS.SYS</td></tr>
<tr><td>0500h</td><td>Область данных DOS</td></tr>
<tr><td>0400h</td><td>Область данных BIOS</td></tr>
<tr><td>0000h</td><td>Векторы прерываний</td></tr>
</table>

Рисунок 7.5 - Типичное распределение адресного пространства PC DOS

4) Далее следует *память пользователя*. Она охватывает десять блоков от 0-го до 9-го. Базовая конструкция PC выделяет только 10 из 16-ти блоков, адресного пространства, что составляет немногим более 60% от всей памяти. В ней располагается операционная система, загружаемая из файлов IO.SYS и MSDOS.SYS (IBMBIO.COM и IBMDOS.COM).

Файл **IO.SYS** – файл загрузки операционной системы (ОС), содержит расширение базовой системы ввода/вывода и является интерфейсом между ОС и BIOS. Он обращался к файлу конфигурации (Config.sys), содержащему сведе-

ния о драйверах устройств. Файл Config.sys вместе со всеми указанными драйверами должен находиться на загрузочном диске. После того, как файл IO.SYS получит необходимые данные, он загружает указанные драйверы в память и связывает их с BIOS.

Файл **MSDOS.SYS** является в некотором смысле набором программ обработки прерываний, в частности прерывания INT 21H. Это тело операционной системы.

Командный процессор COMMAND.COM предназначен для организации диалога с оператором. Он анализирует вводимые оператором команды и организует их выполнение. Так называемые внутренние команды (DIR, COPY и т.д.) обрабатываются именно командным процессором.

Если в файле CONFIG.SYS включены директивы DEVICE = загрузки установленных драйверов (ADM.SYS, SMARTDRV.SYS, EMM386.EXE, ANSI.SYS), то они загружаются вслед за системой.

Выше драйверов размещается резидентная часть командного процессора COMMAND.COM, занимающего около 3 Кбайт.

В функции резидентной части COMMAND.COM входит обработка комбинаций клавиш <Ctrl/C>, <Ctrl/Break> и критических ошибок, вывод сообщений об ошибках, завершение текущей задачи, загрузка транзитной части COMMAND.COM.

Транзитная часть COMMAND.COM размещается в самом верху стандартной памяти и затирается при загрузке больших программ. После завершения выполняемой программы она должна загружаться с диска снова.

5) Оставшаяся часть 384 Кбайта - **верхняя память** (**UMA** – *Upper Memory Area*) первоначально была предназначена для размещения ПЗУ и видеопамяти, а также постоянной памяти адаптеров периферийных устройств.

Блоки A и B этой части памяти предназначенные для работы дисплея. Основная причина такого размещения данных заключается в том, что она позволяет нашим программам очень быстро манипулировать на дисплее данными. Для повышения скорости работы PC к этой части могут получить доступ ЦП и Дисплей, не конфликтуя друг с другом.

Область C, D, E имеет специальное назначение и служит для «расширения ПЗУ». Используется в различных целях, которые возникают в процессе эволюции семейства ПК. Когда к PC добавляются новые устройства, требующие встроенного программного обеспечения, то дополнительные программы размещаются здесь.

Другим назначением области расширения является поддержка «расширенной памяти» - *Expanded Memory*.

Блок F используется для хранения встроенных программ ПЗУ и BIOS. Сюда входят тестовые и инициализирующие программы; базовые программы контроля устройств ввода/вывода; в ряде PC – встроенный Бейсик.

В конце блока F заносится дата создания BIOS (для IBM PC) код типа компьютера FF – базовая модель IBM PC, FE – XT, FC – AT.

Верхняя память вначале использовалась не полностью. В ней имелись «дыры», которым позже было найдено применение. Однако, начиная с версии

DOS 5.0, в операционную систему включен драйвер поддержки HIMEM.SYS, который позволяет эффективно управлять UMB, загружая в них устанавливаемые драйверы устройств, а также резидентные программы расширения DOS (APPEND.EXE, DOSKEY.COM, KEYB.COM и др.).

Загрузка в UMB системных программ освобождает от них стандартную память, увеличивая ее транзитную область.

6) **Высокая память** (*High Memory Area*). При проектировании МП 8086 была допущена ошибка, в результате которой он мог адресовать в реальном режиме на 64 Кбайт – 16 байт памяти больше, чем 8086. Чтобы не изменять сам процессор, эта ошибка устранена специальным внешним узлом (адаптером) блокирующим адресацию памяти за границею 1 Мбайт. С помощью специальным программным управлением этим адаптером в реальном режиме можно осуществить доступ к первым 64 Кбайтам базовой расширенной памяти. Функция управления адаптером возложена на HIMEM.SYS. С его помощью образуется «высокая память» HMA (DOS=HIGH – в файле Config.sys).

Высокая память может хранить данные или одну выполняемую программу. Чаще всего эту область использует DOS, и тогда часть ее резидентных кодов и структур данных перемещаются в HMA, что освобождает стандартную память для других целей.

7) В состав ПК наряду со стандартной памятью 640 Кбайт входит **расширенная** память объемом до 15 Мбайт, располагаемая за пределами первого мегабайта с адресами: 100000H – FFFFFFH.

Функционирование расширенной памяти подчиняется «спецификации расширенной памяти» XMS – *eXtended Memory Area,* то эту память часто называют XMS – памятью. Доступ к ней возможен только в защищенном режиме, поэтому для MS – DOS, работающей только в реальном режиме, расширенная память не доступна.

Первые 64 Кбайта – 16 байт носят название «высокой, или старшей памятью». (HMA – High Memory Area). Хотя она находится за пределами 1 Мбайта, к ней можно обратиться в реальном режиме МП, если определить сегмент, начинающийся с адреса FFFF0H. Первые 16 байт этого сегмента заняты ПЗУ, а область со смещением 0010H – FFFFH может быть использована под программы и данные.

Для управления расширенной памятью используется драйвер HIMEM.SYS, входящий в состав DOS.

Поскольку спецификация XMS предусматривает лишь пересылку данных из стандартной памяти в расширенную и обратно, расширенная память может быть использована только для:

- хранения данных, необходимых выполняемым программам;
- создание виртуальных дисков;
- организация КЭШа для ускорения доступа к дискам.

Загрузка в расширенную память программ для выполнения спецификаций XMS не поддерживается. Как и отображаемая, расширенная память автоматически каждой программе не предоставляется. Программа должна ее запросить.

После подачи питания на компьютер управление передается базовой системе ввода/вывода BIOS. Она выполняет проверку наличия и типа аппаратных узлов компьютера, формирует начальную часть таблицы векторов прерываний, инициализирует устройства и начинает процесс загрузки операционной системы.

В защищенном режиме компьютер работает с виртуальной памятью, которая формируется эмулированием оперативной памяти путем использования как физической оперативной, так и дисковой памяти. Благодаря этому емкость виртуальной памяти получается больше реальной. Исполняемый в настоящее время программный код должен обязательно храниться в физической оперативной памяти, остальной же код может временно располагаться в ее «продолжении» на жестком диске. Когда управление передается фрагменту программы, хранящемуся на диске, он загружается в физическую память. Если места в ней не достаточно, то из памяти на диск выгружается пассивная часть программы. Процесс загрузки и выгрузки программного кода называется свопингом или подкачкой.

На рис.7.6. в качестве примера показана карта виртуальной памяти 32-разрядных процессоров, работающих под управлением операционной системы Windows 32. Максимальная ее емкость составляет 2^{32} = 4 Гбайт. Нижний мегабайт виртуального адресного пространства используется для активной виртуальной машины MS-DOS. Кроме этого, каждая виртуальная машина MS-DOS имеет в своем распоряжении еще и участок виртуальной памяти между вторым и третьим гигабайтами, чтобы Windows могла обращаться к памяти пассивных виртуальных машин MS-DOS.

Каждый процесс для Windows изолирован от других подобных процессов и обладает собственным адресным пространством, размер которого достигает почти 2 Гбайт.

Между третьим и четвертым гигабайтами располагаются компоненты системы, располагающиеся ниже ядра Windows (Диспетчер виртуальных машин, устанавливаемые файловые системы, диспетчер конфигурации, драйверы устройств). Эта зона виртуальной памяти является недоступной со стороны исполняемых процессов.

Виртуальная память имеет страничную организацию. Задача непосредственного управления виртуальной памятью возлагается в Windows на модуль подкачки страниц памяти. Он обеспечивает преобразование виртуальных адресов в физические и осуществляет свопинг страниц виртуальной памяти. Таким образом, в Windows выгружаются и загружаются не зоны виртуальной памяти, отведенные потокам или целым процессам, а небольшие по размеру страницы.

В ОС Windows существенно изменилась роль ПЗУ, расположенного на системной плате. Программы ПЗУ системной платы необходимы лишь для запуска системы, начальной инициализации программно-управляемых устройств компьютера, проверки пароля для входа в систему и загрузки ОС. В процессе загрузки ОС современные 32- и 64-разрядные драйверы загружаются непосредственно с жесткого диска, заменяя все драйверы в ПЗУ системной платы. После

успешной загрузки 32- и 64-разрядных драйверов и ядра операционной системы ПЗУ отключается.

Низкоуровневые компоненты Windows (Модули ядра. Работают в кольце защиты с номером 0)	4 Гбайт 3 Гбайт
DLL Win32, другие совместно используемые процессы	2 Гбайт
Прикладные программы Win32 Виртуальные машины Win16	4 Мбайт
Почти не используется	1 Мбайт
Память MS-DOS	0 Мбайт

Рисунок 7.6 – Карта виртуальной памяти IBM PC

7.4. Особенности архитектуры серверных компьютеров

Аппаратным сервером называется компьютер, выделенный из группы персональных компьютеров (или рабочих станций) для выполнения какой-либо сервисной задачи без непосредственного участия человека. Сервер — самый мощный компьютер в сети, берет на себя нагрузку всех компьютеров локальной сети. Сервер и рабочая станция могут иметь одинаковую аппаратную конфигурацию, так как различаются лишь по участию в своей работе человека за консолью.

Серверное оборудование, как правило, предназначено для обеспечения бесперебойного обслуживания пользователей в круглосуточном режиме 7 дней в неделю. Для обеспечения высокой надежности (0,99999) серверы обычно комплектуется дублирующими узлами, позволяющими обеспечить такую надежность при времени восстановления системы не более 10 мин в год, поэтому при разработке серверов создаются специальные решения, отличные от создания обычных компьютеров, а именно:

- память обеспечивает повышенную устойчивость к сбоям за счет введения коррекции ошибок ECC (Error Checking and Correction,);
- вводится дублирование процессоров;
- используется горячее резервирование (англ. Hot-swap) важных компонентов.

- дублирование накопителей на жестких магнитных дисках в составе массива RAID, контроллеров дисков и блоков питания;
- дублирование групп вентиляторов, обеспечивающих охлаждение компонентов сервера.

Кроме этого, в серверных компьютерах осуществляется аппаратный мониторинг, для реализации которого вводят дополнительные каналы для контроля большего количества параметров сервера:

- датчики температуры контролируют температурные режимы всех процессоров, модулей памяти, температуру в отсеках с установленными жёсткими дисками;
- электронные счётчики импульсов встроенные в вентиляторы выполняют функции тахометров и позволяют, в зависимости от температуры, регулировать скорость их вращения;
- постоянный контроль напряжения питания компонентов сервера позволяет сигнализировать об эффективности работы блоков питания;
- сторожевой таймер не позволяет остаться незамеченным зависанию системы, автоматически производя принудительную перезагрузку сервера.

Серверы (и другое оборудование), которые требуется монтировать в некоторое стандартное шасси (например, в 19-дюймовые стойки и шкафы) изготавливаются по стандартным размерам и снабжаются необходимыми крепежными элементами.

Серверы, не требующие высокой производительности и большого количества внешних устройств, зачастую уменьшают в размерах. Часто это уменьшение сопровождается уменьшением наличных ресурсов сервера.

Особый вид составляют серверы в промышленном исполнении, в которых корпус сервера, кроме уменьшенных размеров, имеет бóльшую прочность, защищенность от пыли (снабжается сменными фильтрами), влажности и вибрации, а также имеет конструкцию кнопок, предотвращающий случайные нажатия.

Конструктивно аппаратные серверы могут исполняться в настольном, напольном и стоечном вариантах. Последний вариант обеспечивает наибольшую плотность размещения вычислительных мощностей на единицу площади, а также максимальную масштабируемость. Всё большую популярность в системах высокой надёжности и масштабируемости получили так называемые блейд-серверы (от англ. blade — лезвие), выполняемые в виде компактных модульных устройств. Такое исполнение позволяет сократить расходы на электропитание, охлаждение, текущее обслуживание и т.п.

Важным показателем серверного компьютера является аппаратная масштабируемость. Масштабируемость — это возможность увеличить производительности сервера или операционной системы (в частности, их способности выполнять больше операций или транзакций за определённый период времени, либо запускать больше различных служб) за счёт установки большего числа процессоров, оперативной памяти и т. д. или их замены на более производительные. Изначально серверы в продаже идут в базовой комплектации, но с заложенным потенциалом к «апгрейду» — аппаратной масштабируемости. К

примеру, базовый набор сервера имеет один процессор, два модуля памяти, например 2х2 Гб и дисковый массив из двух жёстких дисков, емкостью, например, 2 Тб. По мере потребности можно доустановить ещё один процессор, память или добавить диски в массив.

Масштабируемость бывает вертикальная и горизонтальная. Под вертикальной масштабируемостью подразумевается создание одной системы с множеством процессоров, а под горизонтальной — объединение компьютерных систем в единый вычислительный ресурс. Каждый из этих подходов рассчитан на использование в различных областях. Так, горизонтальное масштабирование лучше всего подходит для балансировки нагрузки Web-приложений, а вертикальное масштабирование лучше всего подходит для больших баз данных, управлять которыми на одной системе проще и эффективнее. Кроме аппаратной используется также программная масштабируемость.

Многие современные серверные и клиентские компьютеры строятся на основе многоядерных процессоров Intel Xeon серии 55xx. К основным новшествам процессоров Xeon 55xx относятся:

- Реализация технологии Intel HyperThreading;
- Поддержка до 8 вычислительных потоков;
- Возможность расширение до 8 сокетов с помощью шины QPI;
- Интегрированный в процессор контроллер памяти;
- Расширяемый буфер и каналы памяти;
- Три канала DDR3;
- Более высокая пропускная способность;
- До 16 слотов памяти на один процессор;
- Поддержка функции MCA Recovery (Machine Check Architecture) – контроль и изоляция аппаратных ошибок, могущих привести к краху системы;
- Поддержка технологии Intel Turbo Boost.— автоматического увеличения тактовой частоты в зависимости от текущих параметров системы (напряжения, тока, температура, состояние ОС).

Новые серверы DELL PowerEdge обеспечивают также улучшенные технологии сбережения энергии, такие как:

- Выбираемые пользователем пороги максимального энергопотребления;
- Улучшенное энергосбережение систем по питанию;
- Точные датчики температуры в БП;
- Переключаемые регуляторы напряжения;
- Закрытый цикл охлаждения;
- Улучшенный обдув задней секции серверов;
- PWM-вентиляторы с двумя секциями лопастей и зависимой от конфигурации скоростью вращения лопастей;
- Использование памяти DDR3 с более низким напряжением и потреблением по сравнению с DDR2;
- Динамическая регулировка напряжения на ЦПУ;
- Статическое изменение фаз регулятора напряжения памяти;

- Произвольное время запуска каждого сервера в стойке;
- Секция опций Power/Performance в BIOS;
- Возможность снижения частоты и отключения некоторых модулей памяти;
- Возможность отключения питания на неиспользуемых портах Ethernet или PCI-E;
- Высокоэффективные блоки питания с КПД 88-92%.

7.5. Клавиатура ПЭВМ и ее взаимодействие с процессором

Особенности работы клавиатуры персональной ЭВМ PC–AT состоит в следующем:

a) После нажатия клавиши ее код фиксируется в буфере клавиатуры и подается сигнал компьютеру, о том, что в клавиатуре произошло некоторое событие. Этот сигнал выдается в виде прерывания 09h (IRQ1→ Int 09h);

b) Клавиатура фиксирует отдельно события: когда клавиша *нажимается* и когда ее *отпускают*. Коды нажатого и отжатого состояния клавиши отличаются XT клавиатуре единицей в старшем разряде (отжата), в клавиатуре AT при отжатии клавиши сначала посылается байт F0h, а затем код клавиши;

c) Если клавиша не отпускается в течение времени более 0.5 с, то клавиатура генерирует повторные коды нажатой клавиши.

Для управления клавиатурой (рисунок 7.9) в XT и AT использовался контроллер на основе однокристальной микро-ЭВМ Intel 8048, в более поздних моделях – контроллер 8042, выполняющий, по сути, те же функции, что и 8048.

Основная задача БИС 8048 – отслеживать за клавишами и выдавать сообщение ROM – BIOS при нажатии и отпускании клавиши. В контроллере имеется буфер символов (порты клавиатуры 060H – 063H), в котором может храниться до 20 кодов клавиш, не обработанных ЦП.

Каждый раз, когда нажимается или отпускается одна из клавиш, схема клавиатуры генерирует однобайтовое число, называемое *скен – кодом*, которое однозначно идентифицирует перемещение клавиши. Клавиатура выдает различные скен – коды при нажатии и отпускании клавиши. При нажатии байт скен – кода содержит число в диапазоне от 1 до 83 (в стандартной клавиатуре XT). При отпускании генерируется скен – код на 128 (80H) больше, чем скен – код при нажатии клавиши. Например, при нажатии клавиши Z скен – код 44, а при отпускании 172 = 44 + 128.

Нажатие или отжатие клавиши сообщается ROM – BIOS посредством прерывания INT9, которое вызывает прикладную программу обработки прерываний. При этом читается состояние порта 96 (**60H**), через который осуществляется связь клавиатуры с BIOS.

Передача кодов с клавиатуры производится в последовательном старт-стопном коде (линии: данные и синхронизация, "Сброс", +5, земля). Прикладная программа получает скен-код и переводит его в 2-х байтный код. *Младший байт* этого кода содержит *ASCII*-код клавиши, а *старший – скен-код* клавиши.

Специальные клавиши F1 – F10, Ins, Del и др., а также клавиши дополнительной цифровой клавиатуры имеют в младшем байте 0, а в старшем байте скен – код.

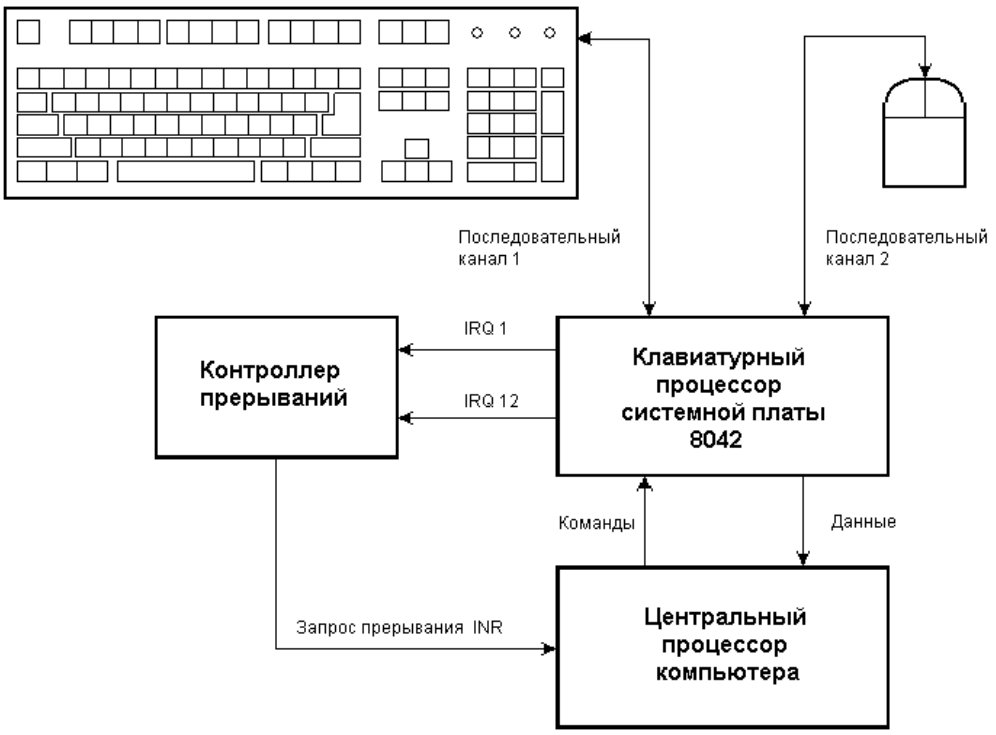


Рисунок 7.9 – Схема подключения клавиатуры и мышки

Затем прикладная программа BIOS помещает оттранслированный код в очередь (буфер), находящуюся в младших адресах памяти с 0000:041E. Работа по переводу скен – кодов усложняется тем, что клавиатура IBM имеет несколько операций по изменению регистров, меняющих значение нажатых клавиш: Shift – c = C, Ctrl – C «break». Также меняется значение клавиши при нажатии клавиши Alt. Кроме этих регистровых клавиш есть еще CapsLock и NumLock.

Информация о состоянии клавиш регистров и клавиш переключателей хранится ROM – BIOS в младших адресах памяти: ячейки 417H и 418H. Когда нажимается одна из этих клавиш, BIOS устанавливает в байтах состояния определенный разряд. Как только BIOS получает код отжатия клавиши, она переключает соответствующий разряд в исходное состояние.

Всякий раз, когда BIOS получает скен – код для нажатия обычной клавиши, то первым делом проверяется состояние регистров, а затем транслируется скен – код в соответствующий 2-й байтный код. В процессе трансляции скен – кодов BIOS постоянно проверяет их на определенные комбинации клавиш: Ctrl+Alt+Del – перезагрузка; Shift – PrintScreen; Int5 – печать экрана; Ctrl – NumLock – приостановка работы, пока не будет нажата любая клавиша; Ctrl – Break Int27 – прерывание. Прямой ввод кодов ASCII может быть осуществлен с дополнительной цифровой клавиатуры при нажатой клавише Alt. (Нажать Alt и вводить цифры кода. При отпускании Alt на экране отображается соответствующий символ ASCII – кода).

После того как действие клавиши оттранслировано, оно записывается в пару байтов в буфере BIOS. *Младший* из них является *главным*, а *старший – вспомогательным*. Когда главный байт содержит ASCII-символ, то это означает, что была нажата одна из клавиш стандартных символов клавиатуры. Для этих ASCII-символов вспомогательный байт содержит скен– код для нажатой клавиши. В простейшем случае этот байт не используется. Он может быть использован, когда необходимо различать клавиши с одинаковыми символами, но различными скен-кодами. При вводе символов со вспомогательной клавиатуры, скен-код во вспомогательном байте равен нулю.

Когда главный байт равен нулю, то это означает, что была нажата какая-то из специальных клавиш: функциональные, управление курсором, либо комбинации с клавишами Ctrl, Alt, Shift. Каждому нажатию специальной клавиши, или комбинации клавиш соответствует определенное значение вспомогательного байта.

Схема взаимодействия клавиатуры с ЭВМ изображена на рис.7.10.

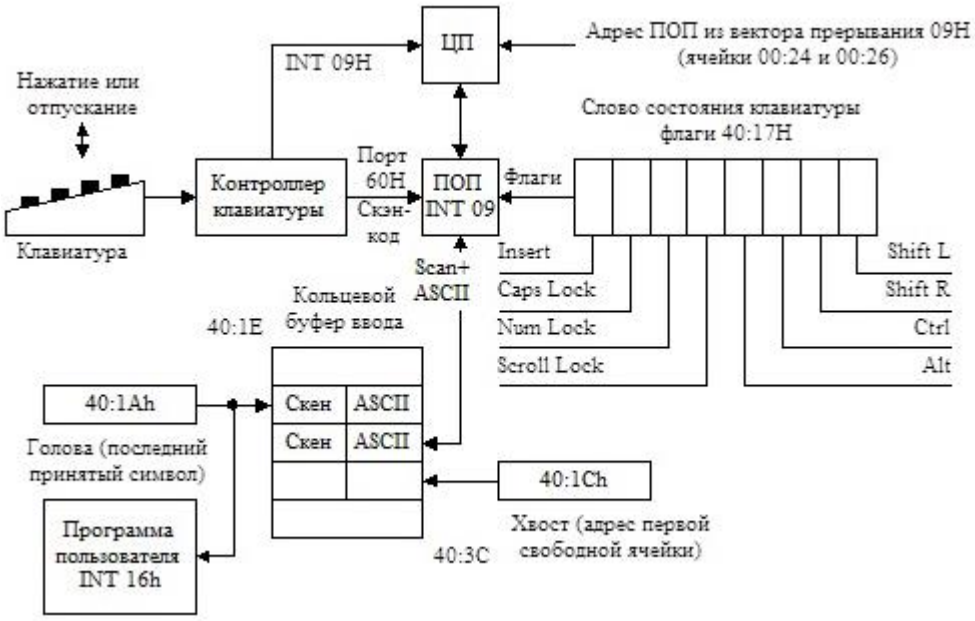


Рисунок 7.10 – Схема взаимодействия клавиатуры с ЭВМ

Нажатие или отпускание любой клавиши вызывает сигнал аппаратного прерывания Int09h, которое заставляет процессор перейти на программу обработки прерываний (ПОП). Адрес ПОП находится в ячейке памяти 9×4=36=24h. Контроллер клавиатуры распознает номер нажатой (отпускаемой) клавиши и помещает ее скен-код в регистр, имеющий адрес 60h.

Программа INT 09 помимо порта 60h работает еще с двумя областями ОЗУ: кольцевым буфером ввода, емкостью 15 слов (адреса от 40:1Eh до 40:3Dh). Каждая ячейка буфера хранит два байта: ASCII клавиши и слово состояния клавиатуры (флаги) – состояние управляющих клавиш. Программа INT 09, получив управление, считывает из порта 60h скен-код и анализирует

его значение. Если скен – код принадлежит одной из управляющих клавиш и, к тому же, представляет собой код нажатия, то соответствующий флаг клавиатуры устанавливается в 1.

При нажатии любой другой клавиши программа INT 09 считывает из порта 60h ее скен – код и по таблице трансляции скен – кодов в коды ASCII формирует двухбайтовый код. Старший байт содержит скен-код, а младший – код ASCII. Поскольку нажатой клавише соответствует два скен – кода (строчные и прописные буквы), то соответственно образуется и два ASCII – кода (a и A, r и R и т.д.).

Для управления светодиодами, расположенными на клавиатуре, необходимо в порт 60h записать код команды 0EDh, а вслед за ней в этот же порт следует занести байт управления: Бит 0 = "1" – включение светодиода Scroll Lock; 0 – выключение; Бит 1 = "1" – включение Num Lock, 0 – выключение; Бит 2= "1" – включение Caps Lock, 0 –выключение.

7.6. Ввод информации с клавиатуры в приложениях Windows

Все приложения, написанные для Microsoft Windows, способны осуществлять ввод данных с клавиатуры. В основу работы приложения с клавиатурой положена следующая модель (рисунок 7.11).

Рисунок 7.11 – Схема взаимодействия клавиатуры с ОС Windows

Windows обеспечивает независимую от устройств поддержку клавиатурного ввода с помощью установки **драйвера клавиатуры**. Также поддерживается языковая независимость входного потока.

При возникновении событий ввода (нажатии клавиш, перемещении мыши) события обрабатываются соответствующим драйвером и помещается в системную очередь аппаратного ввода. В системе имеется особый поток необработанного ввода, называемый **RIT** (*Raw Input Thread*), который извлекает события из системной очереди и преобразует их в сообщения. Полученные сообщения помещаются в конец очереди виртуального ввода одного из потоков (виртуальная очередь потока называется **VIQ** – Virtualized Input Queue). При этом RIT сам выясняет, в очередь какого конкретно потока необходимо поместить событие. Для событий мыши поток определяется поиском окна, над которым расположен курсор мыши. Клавиа-

турные события отправляются только одному потоку – так называемому активному потоку (т.е. потоку, которому принадлежит окно, с которым работает пользователь).

Специализированная задача обработки сообщений выбирает их из очереди и рассылает окнам-абонентам приложений. Выбор окна-абонента осуществляется в соответствии с временным свойством окна, называемым **фокусом**, поскольку общий клавиатурный поток должен быть разделен между всеми окнами.

Задача может вызвать функцию **GetFocus**, чтобы определить, какое окно имеет в настоящий момент фокус ввода. Передача фокуса выполняется функцией **SetFocus.**

При передаче фокуса от одного окна другому система посылает сообщение **WM_KILLFOCUS** тому окну, которое теряет фокус, и **WM_SETFOCUS** окну, получающему фокус.

Нажатие на клавишу приводит к помещению в очередь сообщений **WM_KEYDOWN** или **WM_SYSKEYDOWN**, которые передаются окну, имеющему фокус ввода.

Сообщения о нажатиях и отжатиях клавиш передаются в общем случае **парами**, однако, если пользователь держит клавишу нажатой в течение продолжительного времени так, что включается функция автоповтора, система генерирует множество сообщений **WM_KEYDOWN** или **WM_SYSKEYDOWN** и одно **WM_KEYUP** или **WM_SYSKEYUP.**

Сообщения о нажатиях на клавиши предоставляют разнообразную информацию, но не включают символьный код клавиши. Чтобы получить символьный код, приложение должно вызывать функцию **TranslateMessage** в цикле обработки сообщений.

При этом проверяется виртуальный код и, если он соответствует символу, возвращается кодовый эквивалент с учетом состояния **SHIFT** и **CAPS LOCK** клавиш. Генерируется символьное сообщение, которое помещается в вершину очереди сообщений. Следующая итерация цикла обработки сообщений забирает его и передает ожидающему ввода окну.

7.7. Видеосистема компьютера

Видеосистема компьютера состоит из *видеоадаптера* (видеокарты) и *монитора* (дисплея). Видеоадаптер – это устройство, осуществляющее сопряжение устройства отображения информации – дисплея с компьютером. Дисплей осуществляет преобразование электрических сигналов, поступающих из видеоадаптера в видимое изображение. Основным блоком монитора является электронно-оптическое устройство – электронно-лучевая трубка *(ЭЛТ),* либо плоские экраны на жидкокристаллических элементах. Кроме этого в состав монитора входит отклоняющая система, блок формирования высоковольтного напряжения (только в мониторах на основе ЭЛТ), усилители развертки, схема синхронизации и управления.

Все мониторы используют растровый принцип формирования изображения, в соответствии с которым изображение на экране образуется группой близко расположенных горизонтальных линий или строк (около 1000 линий на экран), называемой *растром*. Электронный луч ЭЛТ последовательно проходит каждую строку слева направо, начиная с левого верхнего угла. Когда луч проходит по строке, цвет и яркость каждой из точек строки (*пикселя*) изменяется и весь растр представляется как связанное изображение (рис.7.12).

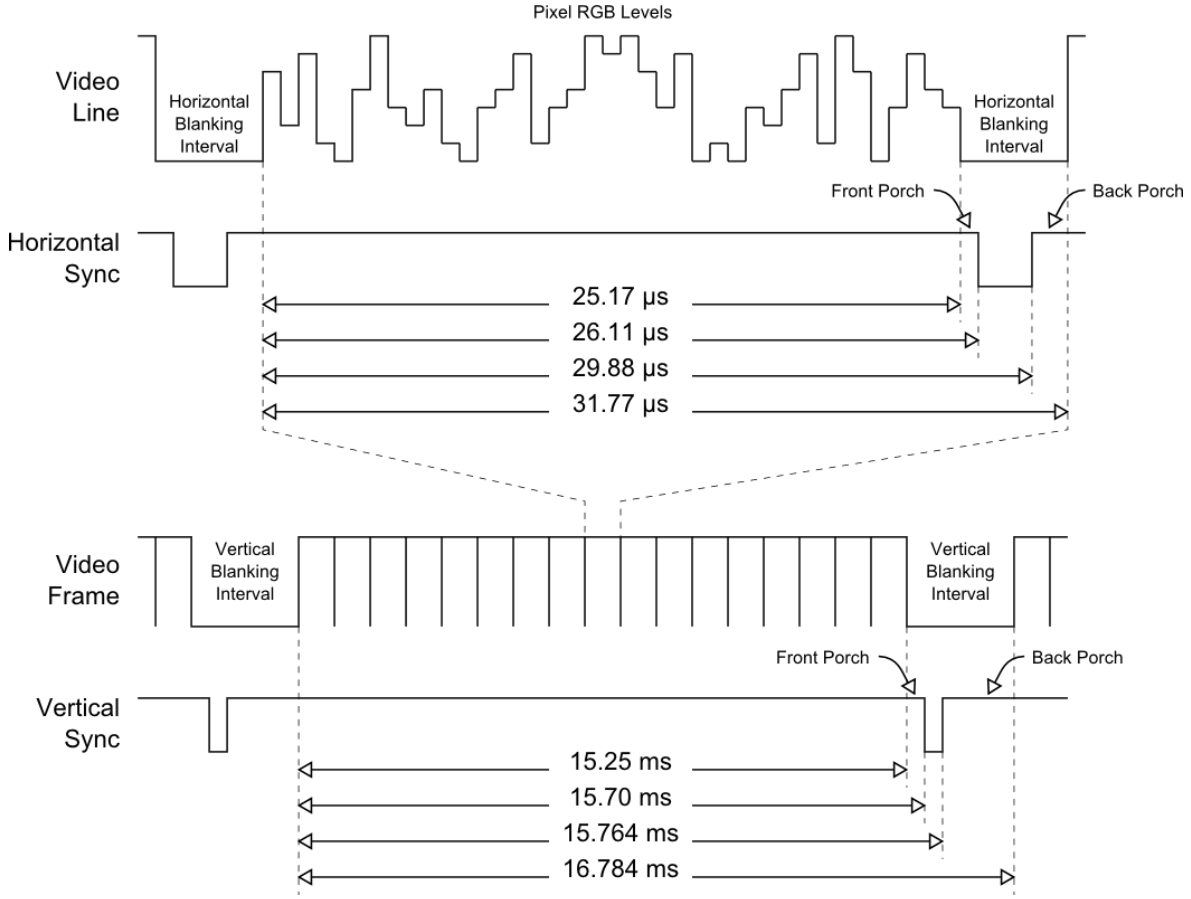


Рисунок 7.12 – Схема видеосигнала

После прохождения очередной строки слева направо луч смещается вниз к началу следующей строки, и так происходит до тех пор, пока луч полностью не сформирует растр. Качество изображения, получаемое на экране монитора, зависит от параметров электронно-оптического устройства и управляющего им видеоадаптера. К основным параметрам относятся: размеры экрана и светящейся точки «зерна», определяющие количество отображаемой информации и возможную степень ее детализации; скорость обновления изображения (частота кадров), определяющая степень подавления мерцания.

Размер экрана монитора подразделяются на 14-дюймовый (36 см), 15-дюймовый (39 см), 17-дюймовый (44 см), 19-дм (49 см). 21 дм (54 см) и т.д. Соответствующие цифры в дюймах (см) указывают размер экрана по диагонали. Важным параметром, определяющим качество изображения – размер пикселей («зерен») его экрана (0.26, 0.28, 0.29 и др.). Чем меньше зерно, тем лучше каче-

ство изображения. Частота кадров в первых мониторах равнялась 50 Гц. В современных мониторах она повышена до 75 – 100 Гц. Размер экрана является главным параметром монитора, наиболее сильно влияющим на качество изображения.

В настоящее время наиболее широко применяются мониторы с использованием жидкокристаллических (ЖК) панелей, которые вытеснили мониторы на основе электронно-лучевых трубок (ЭЛТ). В английском языке для обозначения таких мониторов применяется аббревиатура LCD (*Liquid-Cristal Display*). В производстве LCD известны две технологии: с использованием пассивных матриц (PMLCD-STN) и активных матриц (AMLCD-TFT).

STN-матрица (*Super Twisted Nematic*), состоит из ЖК-элементов с изменяемой прозрачностью, в которых используется *твист-эффект*. Суть этого эффекта состоит в изменении угла вращения плоскости поляризации проходящего света под воздействием электрического поля в так называемых нематических типах жидких кристаллов. Молекулы таких ЖК скручены в спираль.

В активной матрице на тонкопленочных транзисторах — TFT (*Thin Film Transistor*), каждый пиксель управляется отдельным транзистором. Практически все современные ЖК-мониторы используют панели на тонкопленочных транзисторах, обеспечивающих более яркое и четкое изображение большего размера. По сравнению с пассивной матрицей, TFT-мониторы имеет более высокую контрастность, насыщенность, меньшее время переключения (нет "хвостов" у движущихся объектов).

Thin Film Transistor — тонкопленочный транзистор, есть не что иное, как обычный транзистор, но очень тонкий (толщина от 0,1 до 0,01 микрона). В одиночку TFT- элементы не изготавливаются. В едином технологическом процессе создается "пленка", на которой расположена матрица из таких устройств. Так для создания трехцветной матрицы размером 800×600 пикселей понадобится 1440000 отдельных транзисторов.

Принцип действия и многослойная структура всех LCD TFT-дисплеев примерно одинаковы. Свет от лампы подсветки (неоновая или светодиоды, рисунок 7.13) проходит через первый поляризатор и попадает в слой жидких кристаллов, над которыми размещены пластины миниатюрных конденсаторов, управляемых тонкопленочными транзисторами (рисунок 7.14).

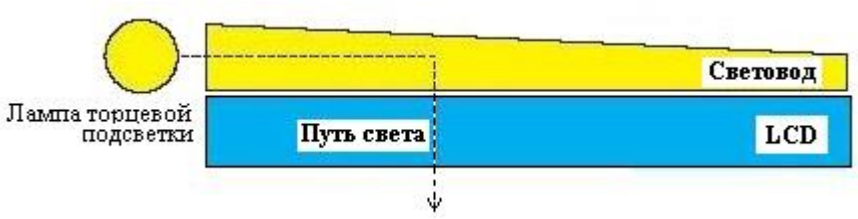


Рисунок 7.13 – Схема свечения LCD-экрана

Транзистор с конденсатором создает электрическое поле, которое формирует ориентацию жидких кристаллов. Пройдя такую структуру, свет меняет

свою поляризацию и будет или полностью поглощен вторым поляризационным фильтром (экран становится черным), или не будет поглощаться (экран - белый), или поглощение будет частичным (одна из градаций яркости цвета).

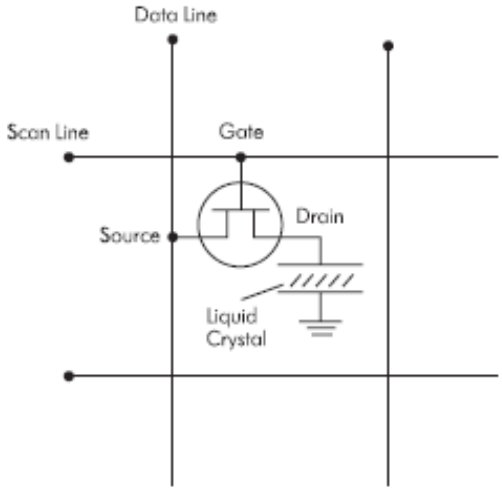


Рисунок 7.14 – Схема ячейки пиксела

Цвет изображения определяют цветовые фильтры. Каждый пиксель матрицы состоит из трех субпикселей - красного, зеленого и голубого. В TFT – мониторах используются различные схемы расположения триад фильтров (рисунок 7.15,б).

Жидкокристаллический дисплей на основе TFT-элементов устроен следующим образом. Поперечное сечение панели на тонкопленочных транзисторах представляет собой многослойную конструкцию (рисунок 7.15,а). Крайние слои сторон конструкции выполнены из стекла. Между этими слоями расположен тонкопленочный транзистор, панель цветного фильтра, обеспечивающая нужный цвет – красный, синий или зеленый, и слой жидких кристаллов. Вдобавок ко всему существует флуоресцентная подсветка, освещающая экран изнутри.

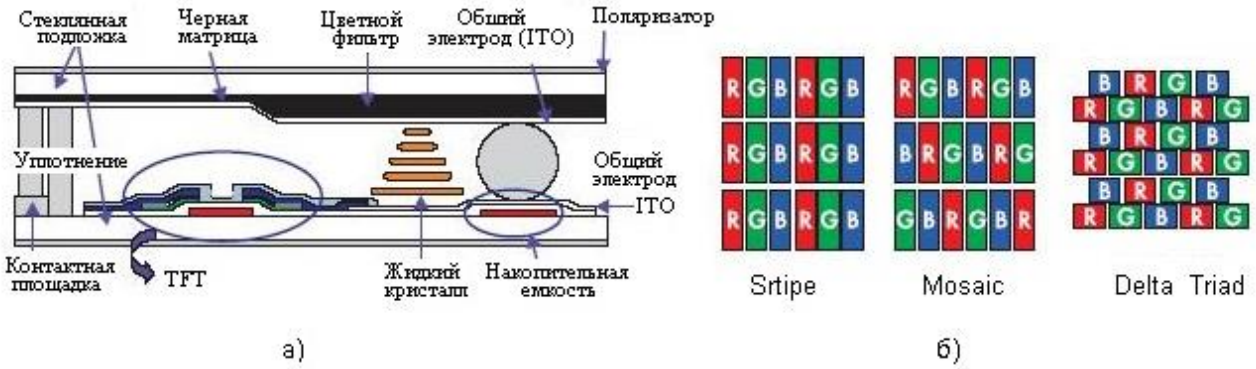


Рисунок 7.15— Структура TFT LCD-панели (а) и структура цветного фильтра (б)

При нормальных условиях, когда нет электрического напряжения, жидкие кристаллы находятся в аморфном состоянии. Количеством света, проходящего через жидкие кристаллы, можно управлять с помощью электрических напряжений, вследствие чего изменяется пространственная ориентация кристаллов.

Управление яркостью в жидкокристаллическом дисплее основано на поляризации света. Свет поляризуется с определенным углом поляризации, проходя через поляризационный фильтр. При этом наблюдатель видит только снижение яркости света (почти в 2 раза). Если за этим фильтром поставить еще один такой фильтр, то свет будет полностью поглощаться, при условии, что угол поляризации второго фильтра перпендикулярен углу поляризации первого, или полностью проходить, если углы поляризации совпадают. При плавном изменении угла поляризации второго фильтра интенсивность проходящего света будет также плавно изменяться.

Пиксель формируется из трех участков – красного, зеленого и синего. А различные цвета получаются в результате изменения величины соответствующего электрического потенциала на пластинах конденсатора, что приводит к повороту вектора поляризации кристаллом и изменению яркости проходящего светового потока. Каждый элементарный участок LCD-панели обслуживается своим тонкопленочным транзистором. При отсутствии сигнала тонкопленочный транзистор закрыт и свет проходит свободно (экран светится белым цветом). При наличии активного уровня сигнала транзистор открывается, соответствующий жидкокристаллический элемент меняет свою ориентацию и при достаточном напряжении система полностью не пропускает свет. Таким образом, варьируя подачу напряжения на TFT транзистор, можно регулировать пропускание света компонентом матрицы.

За каждым элементом стоит свой светофильтр (R,G,B). Поэтому, регулируя освещенность каждой из трех компонент пикселя (точки), можно получить необходимый оттенок цвета пикселя. При этом пиксель может иметь белый цвет (у каждого из трех элементов закрыт транзистор – свет проходит – все три элемента RGB светятся по максимуму – в сумме белый цвет). Если все три транзистора пикселя открыты по максимуму, то свет не проходит и свечение отсутствует, что дает черный цвет. TFT-экран состоит из целой сетки таких пикселей, где работой каждого цветового участка каждого пикселя управляет отдельный транзистор. Количество транзисторов определяет разрешающую способность экранной матрицы, которая является одним из важнейших параметров монитора.

В жидкокристаллических (ЖК) мониторах персональных компьютеров также используется растровый принцип построения изображения. Растровое изображение представляет собой сетку пикселей или цветных точек (обычно прямоугольную) на компьютерном мониторе, бумаге и других отображающих устройствах и материалах (растр). Растровый дисплей можно рассматривать как матрицу дискретных ячеек (точек), каждая из которых может быть подсвечена. Таким образом, оно является точечно-рисующим устройством.

Для подсветки ячеек используется линейная развертка, т. е. развертка с постоянной скоростью вдоль строк и по кадру. При перемещении светящейся точки по горизонтали на экране монитора прочерчиваются **строки** растра, а перемещением развертывающего элемента по вертикали из совокупности строк образуется растр. Количество точек на экране LCD-монитора (разрешающая способность) фиксировано и определяется размером экранной панели: так для

15-дюймовых LCD-панелей разрешающая способность равна 1024×768 пикселей, а для 17” панелей —1280×1024. Структурная схема жидкокристаллического дисплея изображена на рисунке 7.16.

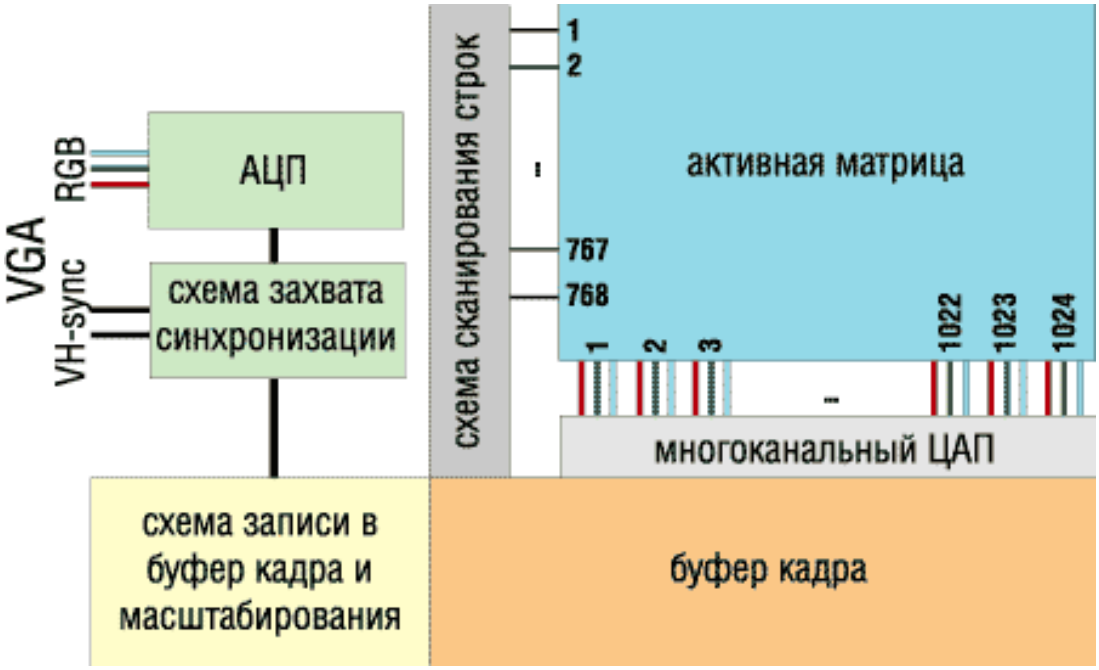


Рисунок 7.16 – Структура ЖК-дисплея

В состав TFT-монитора на основе жидкокристаллической LCD-панели, кроме собственно LCD-панели, входит контроллер формирования управляющих сигналов, который последовательно построчно формирует сигналы управления транзисторами RGB-матрицы, образуя тем самым растровое изображение. Подсветка соответствующего пикселя осуществляется подачей управляющих сигналов на усилители вертикальных и горизонтальных строк и далее на управляющие электроды TFT. Для осуществления подсветки с торца панели используются специальные электролюминисцентные лампы фоновой подсветки (*Backlight Lamp*). Для них требуется повышенное напряжение, вырабатываемое преобразователем напряжения, который генерирует напряжения VBL (*Voltage Back Light*) для двух ламп. На вход монитора с видеокарты компьютера подаются сигналы синхронизации горизонтальной и вертикальной разверток, RGB-сигналы управления цветностью и ряд вспомогательных сигналов.

Важным параметром является минимальное время послесвечение пикселя. Так, минимальное время свечения пикселя у TFT-мониторов достигает на настоящее время 10…15 мс (для ЭЛТ-мониторов — 3…5 мс). Т.е быстродействие TFT –монитора ниже, чем ЭЛТ, однако для большого ряда применений этого быстродействия вполне хватает.

К другим параметрам монитора относятся цветовая битность матрицы и размер экрана по диагонали. Все современные панели для нормального воспроизведения цветов должны поддерживать режим True Color (24 bit).

Для того чтобы на экране монитора формировалось изображение, информация о каждой точке (код цвета точки) должна храниться в видеопамяти ком-

пьютера. В графических режимах дисплея программно можно управлять цветом любого выводимого пикселя, что позволяет строить на экране монитора сложные графические образы. Видеосистемы персональных компьютеров хранят информацию о пикселе в виде *группы битов*, представляющих значение пикселя. Цвет каждого пикселя определяется непосредственно или косвенно по значению группы пикселей. Формат расположения битов (битовая карта) в видеобуфере зависит как от количества битов, необходимых для представления каждого пикселя, так и от архитектуры видеопамяти. Количество цветов, которое может отображаться в данном графическом режиме одновременно, определяется числом битов, представляющих каждый пиксель.

Для соединения видеоадаптера с жидкокристаллическим монитором или с мультимедийным проектором разработан специальный стандартный цифровой видеоинтерфейс **DVI** (*Digital Visual Interface*) стандарт на интерфейс и соответствующий разъём. В отличие от мониторов на электронно-лучевой трубке, имеющих аналоговый интерфейс, LCD-мониторы, по природе своей, цифровые. Их использование с аналоговым интерфейсом – временное вынужденное решение: в видеокарте цифровой сигнал преобразуется в аналоговый, передается на монитор, где опять преобразовывается в цифровую форму. Поэтому возникновение цифрового видео-интерфейса было совершенно очевидным. Цифровой видеоинтерфейс может поддерживать одинарный и двойной режимы.

Одинарный режим DVI (*Single link*) использует четыре витых пары проводов (красный, зелёный, синий, и clock), обеспечивающих возможность передавать 24 бита на пиксель. С ним может быть достигнуто максимальное возможное разрешение 1920×1200 (60 Гц) или 1920×1080 (75 Гц).

Двойной режим DVI-D (*Dual link*) — удваивает пропускную способность и позволяет получать разрешения экрана 2560×1600 и 2048×1536. Поэтому для самых крупных LCD мониторов с большим разрешением, таких, как 30" модели, обязательно нужна видеокарта с двухканальным DVI-D Dual-Link выходом.

Существует несколько разновидностей интерфейсов DVI (рис.2.13):

DVI-A — только аналоговая передача.

DVI-I — аналоговая и цифровая передача.

DVI-D — только цифровая передача.

В Single Link DVI присутствует четыре канала передачи данных: три из них соответствуют информации об основных цветах B (*Channel* 0), G (*Channel* 1) и R (*Channel* 2), четвертый же несет в себе сигнал тактовой частоты «*Clock*» (*Channel* C). Физически кабель DVI состоит из соответствующего количества витых пар.

В Dual link скоростные возможности DVI достигнуты за счет алгоритма кодирования сигналов, специально разработанного для этой цели. Называется данный алгоритм TMDS – *Transition Minimized Differential Signaling*, или дифференциальная передача сигналов с минимизацией перепадов уровней, что можно было бы еще назвать «сверхплотным архивированием данных без потерь». При этом дифференциальный, или балансный, способ передачи, когда по каждому проводнику витой пары проходит один и тот же прямой и инвертированный сигнал, обеспечивает эффективную защиту данных от синфазных по-

мех.

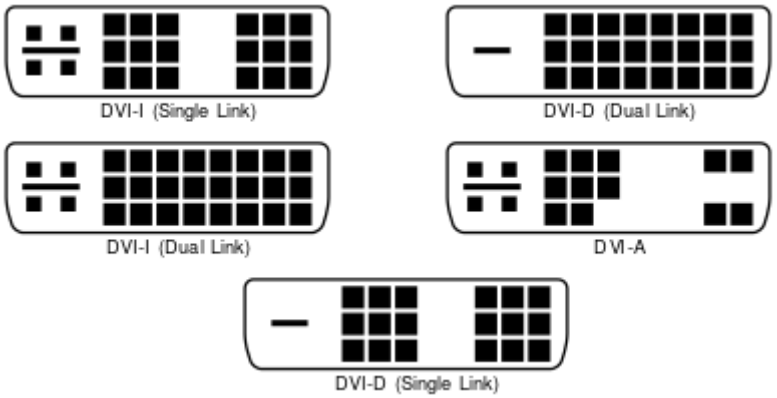


Рисунок 7.17 – Виды разъемов видеосистемы

Кроме каналов цветности имеется Канал DDC (*Display Data Channel*), предназначенный для передачи процессору информации о дисплее, который на основании ее, выдает оптимальные для данного дисплея сигналы с нужным разрешением и экранными пропорциями.

7.8. Аудиосистема компьютера

Звуковая система компьютера состоит из звукового адаптера (звуковой карты) и электроакустических преобразователях звуковых колебаний (микрофона и звуковых колонок).

Звуковые карты выполняют следующие функции:

- дискретизацию аналоговых сигналов с частотами 11,025 кГц, 22,05 и 44,1 кГц. Первая частота относится к 8 битовым картам, другие – к 16 битовым;
- 8- или 16– битовое квантование, кодирование и декодирование с использованием линейной импульсно-кодовой модуляции (ИКМ);
- одновременно производить запись и воспроизведение звуковой информации (режим Full duplex);
- ввод сигналов через монофонический микрофон с автоматическим регулированием уровня входного сигнала;
- ввод и вывод аудиосигналов через линейный вход/выход;
- микширование (смешивание) сигналов от нескольких источников и выдача суммарного сигнала в выходной канал. В качестве источников используются:
  - а) аналоговый выход CD-ROM;
  - б) ЦАП;
  - в) музыкальный синтезатор;
  - г) внешний источник, подключенный к линейному входу.
- управление уровнем суммарного сигнала и сигнала каждого из каналов в отдельности;
- обработка стереофонических сигналов;

- синтез звуковых колебаний с использованием частотной модуляции (FM) и волновых таблиц (WT).

Звуковая карта должна использовать не более 13% ресурсов процессора ЭВМ при частоте дискретизации 44,1 кГц и не более 7% - при $f_д = 22{,}05$ кГц. В звуковой карте осуществляется обработка аналоговых и цифровых сигналов. В соответствии со спецификацией AC-97 (*Audio Codec 97 Component Specification*), разработанной фирмой Intel в 1997 году, обработка звуковых сигналов разделена между двумя устройствами:

звуковым кодеком (*AC-audio codec*) и
цифровым контроллером (*DC – digital controller*).

Расширенная модификация БИС звукового кодека дополнительно выполняет функции модема.

Продолжением спецификации AC-97стал стандарт звука высокой четкости HD Audio (*high definition audio*), предложенным компанией Intel в 2004 году. Он обеспечивает воспроизведение большего количества каналов с более высоким качеством звука, чем при использовании интегрированных аудиокодеков AC'97. Аппаратные средства, основанные на HD Audio, поддерживают 24-разрядное качество звучания (до 192 кГц в стереорежиме, до 96 кГц в многоканальном режимах — до 8 каналов).

В упрощенном виде схема аудиосистемы PC может быть представлена следующим образом (рисунок 7.18). Микрофон (М) осуществляет преобразование акустических колебаний в электрический, а громкоговоритель (Гр.) преобразование электрических колебаний в акустические. Входной сигнал с микрофона усиливается, а с линейного входа подается непосредственно на аналого-цифровой преобразователь. Выходной сигнал также может формироваться с помощью синтезатора звуковых колебаний.

Рисунок 7.18 – Структура звуковой карты

Синтез звуковых сигналов предназначен для генерации звуков музыкальных инструментов, соответствующие определенным нотам, а также создавать „немузыкальные" звуки: шум ветра, выстрела и т.п.

Одна и та же нота, воспроизводимая на музыкальном инструменте, звучит по-разному (скрипка, труба, саксофон). Это вызвано тем, что хотя определенной ноте соответствует колебание конкретной частоты, звуки различных инструментов, кроме основного тона (синусоиды), характеризуются наличием дополнительных гармоник – *обертонов.* Именно обертоны определяют тембровый окрас голоса музыкального инструмента.

Созданный с помощью музыкального инструмента звуковой сигнал состоит из трех характерных фрагментов – фаз. Так, например, при нажатии клавиши рояля амплитуда звука сначала быстро растет до максимума, а затем немного спадает (рисунке 7.19).

Рисунок 7.19 – Форма воспроизведения звукового сигнала

Начальная фаза звукового сигнала называется атакой. Длительность атаки для различных музыкальных инструментов варьируется от единиц до десятков и даже сотен мс. После атаки начинается фаза „поддержки", в течение которой звуковой сигнал имеет стабильную амплитуду. Слуховое ощущение высоты звука формируется как раз на стадии поддержки. Далее следует участок с относительно быстрым затуханием уровня сигнала. Огибающая колебаний во время атаки, поддержки и затухания называется амплитудной огибающей. Различные музыкальные инструменты имеют разные амплитудные огибающие, тем не менее, отмеченные фазы характерны практически для всех музыкальных инструментов, за исключением ударных.

Для создания электронного аналога реального звука, т.е. для ***синтеза*** звука, необходимо воссоздать огибающие гармоник, из которых состоит реальный звук. Существует несколько методов синтеза. Одним из первых и наиболее изученных является аддитивный синтез. Звук в процессе синтеза формируется путем сложения нескольких исходных звуковых волн. Этот метод использовали еще в классическом органе. Специальной конструкцией клапанов при нажатии клавиши заставляли звучать сразу несколько труб. При этом звучащие трубы были настроены либо в унисон или в одну две октавы. При нажатии клавиши первыми начинали звучать короткие трубы, дающие высокие обертоны, затем вступала средняя секция и последними – басы.

При цифровом аддитивном синтезе отдельно формируется N гармоник с частотами от f_1 до f_N и амплитудами от $A_1(t)$ до $A_N(t)$. Затем эти гармоники складываются.

Второй метод является разновидностью нелинейного синтеза. Для получения одного музыкального звука используется сигнал одного генератора. Гармоническую окраску получают в результате нелинейных искажений исходного сигнала. Для этого синусоидальный сигнал, формируемый генератором, управляемым кодом (ГУК) с амплитудой A_1 и частотой f_1 (рисунок 7.16 а) пропускают через нелинейный элемент с некоторой характеристикой $K(x)$ (рисунок 7.20 б). Зная амплитуду сигнала A_1 и вид характеристики $K(x)$, можно вычислить спектр сигнала на выходе (рисунок 7.20 в).

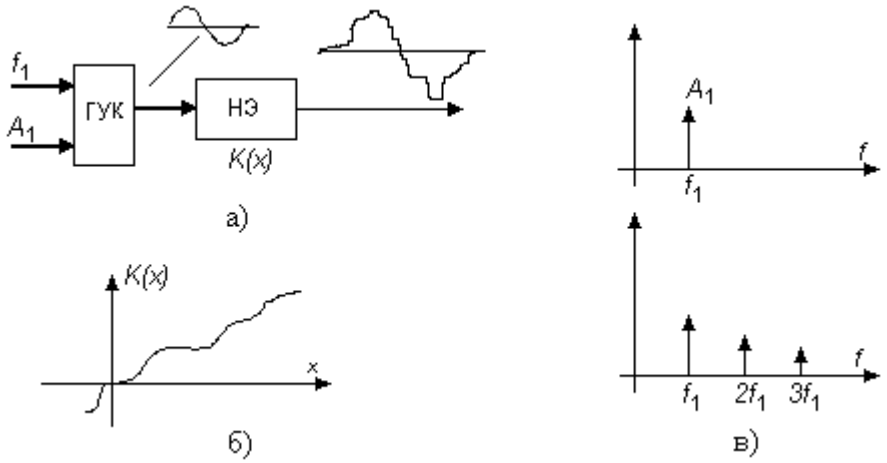


Рисунок 7.20 – Синтез звуковых сигналов на основе нелинейного преобразования

Следующим широко распространенным методом является синтез на основе ***частотной модуляции*** (широко используется в ЭМИ фирмы Yamaha). При частотной модуляции осуществляется изменение частоты f_0 несущего колебания $U(t) = A\sin(2\pi f_0 + \varphi)$ по закону модулирующего колебания $x(t)$. Выражения для частотно-модулированного колебание имеет вид

$$U(t) = A\sin(\omega_o t + \Delta\omega\int[2x(t) - 1]dt),$$

Величина изменения частоты несущего колебания $\Delta\omega_0=2\pi f_0$ называется девиацией частоты, а отношение отклонения Δf_0 частоты модулированного колебания к частоте модулирующего колебания f_m называется индексом частотной модуляции $m_f = \Delta f_0/f_m$. Изменяя индекс модуляции можно изменять спектр сигнала на выходе модулятора и тем самым достичь качества синтезируемого звука, близкого к естественному звучанию. Выражения для частотно-модулированного колебание при синусоидальном модулирующем колебании $x(t) = \sin\omega_o t$ имеет вид

$$U(t) = A\sin[2\pi f_o t + m_f \sin(2\pi f_m t)].$$

Спектр модулированных сигналов при различных индексах модуляции изображен на рисунке 7.21.

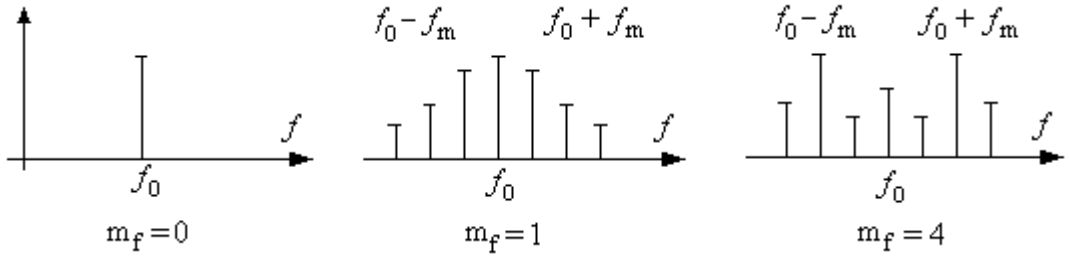


Рисунок 7.21 – Спектры сигналов с различным индексом модуляции

На рисунке 7.21 слева изображен спектр чистого тона. Варьируя индекс модуляции, можно изменять спектральный состав звукового колебания и тем самым изменять окраску звука.

7.9. Последовательный интерфейс персонального компьютера

Персональный компьютер имеет ряд внешних интерфейсов для подключения дополнительных устройств. К ним относится клавиатура, “мышь”, печатающее устройство, модем и т.д. Интерфейсы можно разделить на две группы: последовательные и параллельные.

Последовательный интерфейс RS-232/V.24.

Интерфейс RS-232C предназначен для синхронной и асинхронной передачи данных в дуплексном режиме работы. Введен в 1969 г. в США (RS-*Recommended Standard*). Первоначально скорость передачи до 20 Кбит/с. Цепи основных линий интерфейса приведены в табл.2.1.

Международный консультативный комитет по телефонии и телеграфии (МККТТ), разрабатывающий стандарты в области передачи данных, ввел свой вариант этого стандарта. Он получил название V.24. Нумерация цепей в этом стандарте цифровая и изменяется от 102 до 125. Для передачи данных и управляющих воздействий в цепях RS-232C и V.24 используются разнополярные сигналы. В частности логическому значению “0” соответствует напряжение от +3 до +15 В, а для передачи “1” напряжение от −3 до −15 В. В 1991 г. была введена модификация E (RS-232E), позволяющая передавать данные с более высокой скоростью (до 345 Кбит/с).

Таблица 7.1 – Цепи последовательного интерфейса RS-232

Название цепи	Название цепи (англ)	Обозначение RS-232	Номер контакта	Обозначение V.24
Передаваемые данные	Transmit Data	TxD	2	103
Принимаемые данные	Receive Data	RxD	3	104
Запрос передачи	Request to Send	RTS	4	105
Готов к передаче	Clear to Send	CTS	5	106
Передатчик готов	Data Set Ready	DSR	6	107
Приемник готов	Data Terminal Ready	DTR	20	108
Детектор линейного сигнала	Data Carrier Detect	DCD	8	109
Индикатор вызова	Ring Indicator	RI	22	125
Сигнальная земля	Signal Ground		7	102

Для реализации интерфейса вначале была создана БИС универсального синхронного-асинхронного приемопередатчика (УСАПП) типа 8250, затем были разработаны ее модификации 16450 и 16550 с наличием FIFO – буферизации. Схема последовательного адаптера ПЭВМ IBM PC изображена на рисунке

7.22. Он носит название COM-порт. С точки зрения программиста адаптер представляет собой восемь регистров, базовый адрес для COM1 3F8h.

Линия INTR подключается к линии IRQ компьютера через ключ, управляемый сигналом $\overline{OUT2}$ УСАПП. $\overline{OUT1}$, $\overline{OUT2}$-биты регистра управления модемом, их можно изменять программно.

Интерфейсную БИС можно запрограммировать на разрешение или запрещение прерываний. Возникновение одной из следующих ситуаций приводит к формированию сигнала INTR:

- очередной или несчитанный символ находится в буферном регистре приемника;
- регистр передачи пуст;
- возникла ошибка на линии (ошибка четности, формата, переполнения);
- УСАПП обнаруживает изменение состояния линий с модема (CTS, DSR, DCD, RI).

Следует заметить, что последовательный адаптер не реализует полный интерфейс RS-232, так как в конкретных применениях (например, при подключении модема к последовательному порту ПЭВМ) полный набор может не понадобиться, и обычно реализуется только часть этих возможностей.

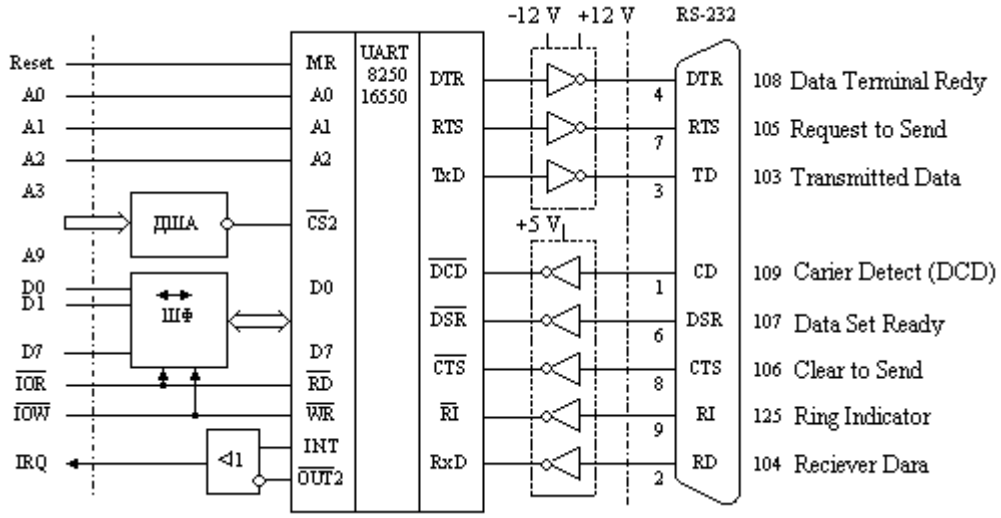


Рисунок 7.22 – Структурная схема последовательного адаптера ПЭВМ

В настоящее время в компьютерах более распространенным является синхронный последовательный интерфейс типа **SSI** (*Synchronous Serial Interface*), в основе которого лежит интерфейс RS-422, улучшенный вариант интерфейса RS-232. Передача данных осуществляется по четырехпроводной симметричной линии. Скорость передачи зависит от длины линии связи и достигает 1,5 Мбит/с. Еще одна пара проводов используется для передачи тактовых импульсов.

7.10. Универсальный интерфейс USB

Интерфейс **USB** (*Universal Serial Bus*) — это новый интерфейс, который является универсальным последовательным интерфейсом. Используется в компьютерах с 1999 года. Предназначен для подключения к ПК любой периферии со скоростью обмена 12 Мбит/с (полная скорость), либо с низкой скоростью 1,5 Мбит/с.

Помимо универсальности этот интерфейс позволит избавиться от шины ISA. До тех пор пока в ПК существует COM– и LPT–порты и интерфейс для подключения флоппи – дисков в материнской плате должен существовать интерфейс ISA, со всеми присущими ему недостатками: конфликты прерываний и адресов портов, дополнительные дорожки на плате для разводки этих портов.

Архитектурой USB предусматривается топология "*звезда*". Система должна состоять из одного **ведущего** (корневого) **концентратора** с хост-контроллером (Host Controller), управляемым операционной системой, некоторого количества концентраторов и узлов (периферийных устройств). Хост-контроллер входит в состав системного блока компьютера. На шине USB допускается наличие только одного хоста. На системном блоке персонального компьютера может располагаться несколько хостов, каждый из которых управляет собственной шиной USB.

Концентратор (Hub, хаб) – ключевой элемент системы Plug-and-Play в архитектуре USB. Он служит для создания дополнительных портов. Хаб является кабельным концентратором, а точки подключения к нему называются портами хаба.

Концентраторы могут каскадироваться, образуя древовидную структуру с поддеревьями (рисунок 7.23). Архитектурой интерфейса предусмотрено каскадное соединение не более 5 хабов. У каждого хаба имеется один **восходящий** порт (Upstream Port), предназначенный для подключения к компьютеру (хосту) или хабу верхнего уровня. Остальные порты являются **нисходящими** (Downstream Ports) и предназначены для подключения функциональных устройств или хабов нижнего уровня.

В интерфейсе введено понятие функции, под которым понимается периферийное устройство или отдельный блок периферийного устройства, способный передавать и принимать информацию по шине USB. Каждая функция предоставляет конфигурационную информацию, характеризующую возможности периферийного устройства. Еще одним понятием в интерфейсе является **конечная точка**. Она представляет собой часть USB-устройства, имеющего уникальный идентификатор и являющегося получателем или отправителем информации. Обычно роль конечной точки исполняет один или несколько регистров или буфер памяти. Любое устройство имеет конечную точку с нулевым номером. Логическое соединение между конечной точкой и программным обеспечением хоста получило название **канала** (pipe, англ. труба).

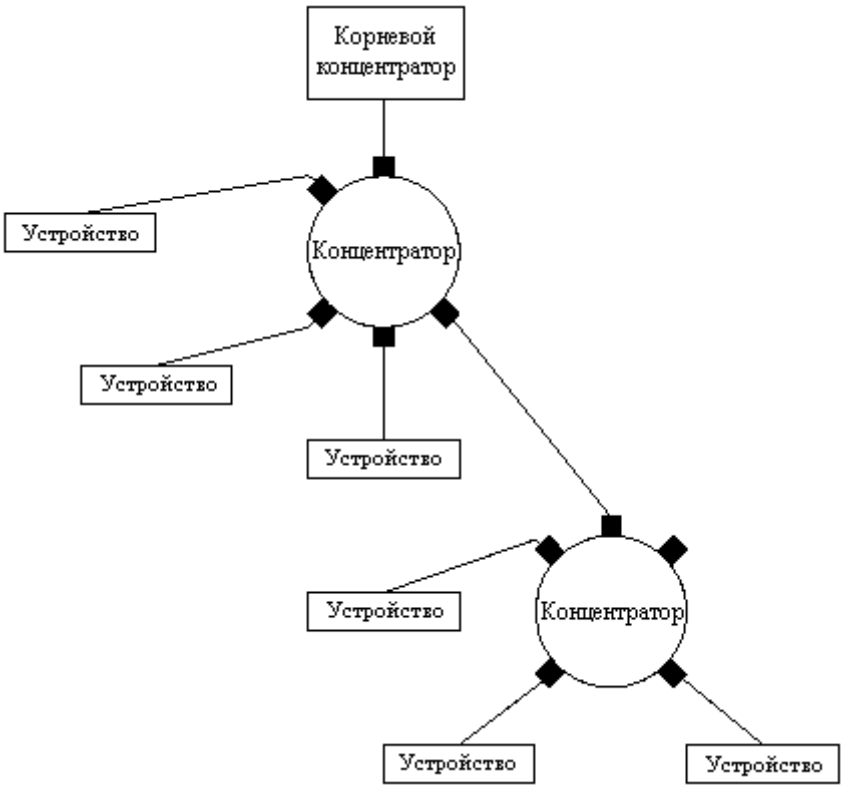


Рисунок 7.23 – Древовидная структура UBS-интерфейса

Устройства подключаются к концентраторам. Всего устройств может быть до **127,** концентратор также считается устройством. На практике такое количество устройств вряд ли используется, а топология чаще остается линейной. В устройства, подключаемые к шине USB, часто встраивается концентратор с единственным выходным портом, что позволяет объединять устройства в цепочку. При построении цепочки следует иметь в виду, что наиболее скоростные устройства (монитор) следует подключать ближе к корневому концентратору, а наименее скоростные (клавиатура) – в конец цепочки. Это обеспечит приоритетное обслуживание высокоскоростных устройств.

Для соединения устройств и концентраторов USB используется 4-х жильный кабель: по двум его проводам подается электропитание (+ 5V) на устройства, не имеющие своего источника, а два других используется для передачи информации.

Одним из преимуществ USB является возможность “горячего” подключения без перезагрузки системы. При обнаружении на шине нового устройства концентратор оповещает об этом корневой концентратор. Затем система опрашивает вновь подключенное устройство о его возможностях и потребностях и конфигурирует его. Вдобавок при этом загружаются необходимые драйверы.

USB поддерживает ряд режимов передачи, как однонаправленных, так и двунаправленных. Передача осуществляется между ПО компьютера (хоста) и конкретной конечной точкой устройства. Устройство может иметь несколько конечных точек, связь с каждой из них (канал) устанавливается независимо от других.

Архитектура USB допускает четыре базовых типа передачи данных:

- Управляющие посылки, используемые для конфигурирования во время подключения и в процессе работы для управления устройствами. Длина поля данных управляющей посылки не превышает 64 байт на полной скорости и 8 байт на низкой.
- Сплошная передача небольших пакетов без жестких требований ко времени доставки (например, от сканера или к принтеру). Длина пакетов от 8 до 64 байт.
- Прерывания – для передачи типа вводимых символов или координат.
- Изохронная передача – непрерывная передача в реальном времени, имеющая заданную задержку доставки сообщения. В случае обнаружения ошибки изохронные данные передаются без повтора – недействительные пакеты игнорируются. Типичным примером изохронной передачи является цифровая передача речевых сообщений.

После включения или сброса устройства оно устанавливает нулевой адрес, используемый для возможности его конфигурирования. Для работы с компьютером каждое устройство должно иметь уникальный адрес (от 1 до 127), задаваемый ему при начальной конфигурации корневым концентратором.

Общая схема обмена управляющими сообщениями и данными по USB интерфейсу изображена на рисунке 7.24.

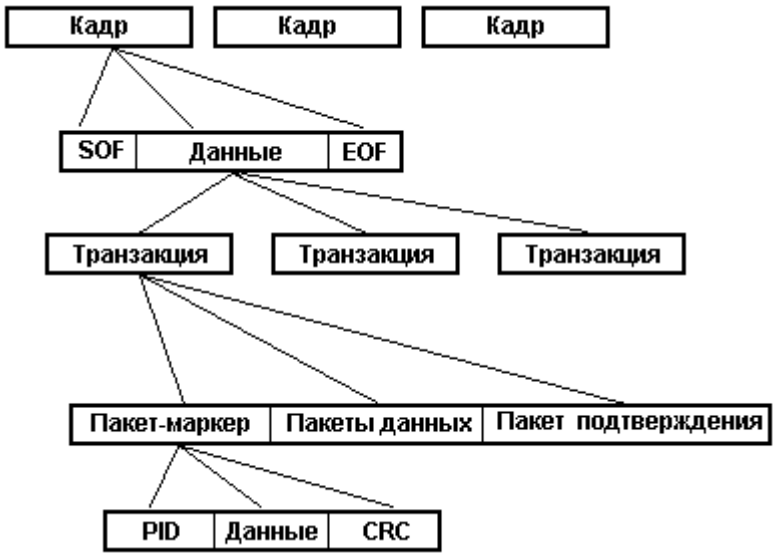


Рисунок 7.24 – Общая схема процесса обмена данными по протоколу USB

Любой обмен данными по шине USB инициируется хост-контроллером. Хост-контроллер формирует кадры из наиболее приоритетных данных. Каждый кадр начинается стартовой посылкой SOF (*Start of Frame*) и завершается посылкой EOF (*End of Frame*). Любая передача состоит из одной или нескольких транзакций. Все транзакции по USB интерфейсу состоят из трех пакетов. Каждая транзакция начинается по инициативе контроллера, который посылает *па-*

кет-маркер, описывающий тип и направление передачи, адрес устройства USB и номер конечной точки. В каждой транзакции возможен обмен только между адресуемым устройством (его конечной точкой) и ЭВМ. Адресуемое маркером устройство USB распознает свой адрес и подготавливается к обмену. Источник данных (определенный маркером) передает *пакет данных* или уведомление об отсутствии данных, предназначенных для передачи. Каждый пакет начинается с синхронизирующей кодовой комбинации SYNC, за которой следует идентификатор пакета PID (Packet Identifier), собственно данные и контрольная последовательность пакета CRC.

После успешного приема пакета получатель данных посылает *пакет подтверждения* (*Handshake Packet).* Для защиты передаваемой информации от ошибок используется помехозащищенное кодирование циклическим кодом. В случае обнаружения ошибок контроллер автоматически производит повторную передачу (до трех раз). Если повторы безуспешны, сообщение об ошибке передается клиентскому ПО для программной обработки.

Для передачи данных по двухпроводной линии применяются двухполярные линейные сигналы без возврата к нулю (рисунок 7.25) типа NRZI (*Non-Return to Zero Inverted*).

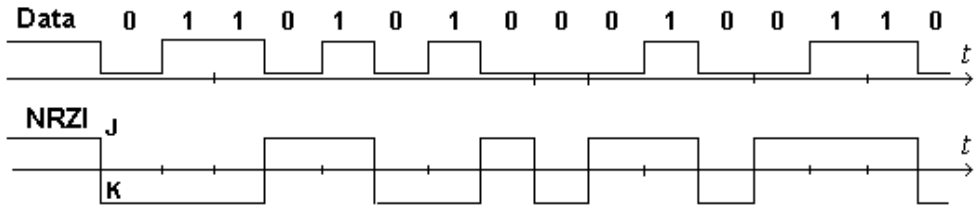


Рисунок 7.25 – Линейные сигналы интерфейса USB

В этих сигналах каждый логический ноль данных приводит к изменению полярности сигнала, а при передаче логической "1" полярность сигнала остается прежней. Такое линейное кодирование позволяет избавиться от длинных последовательностей нулей и улучшить синхронизирующие свойства сигналов.

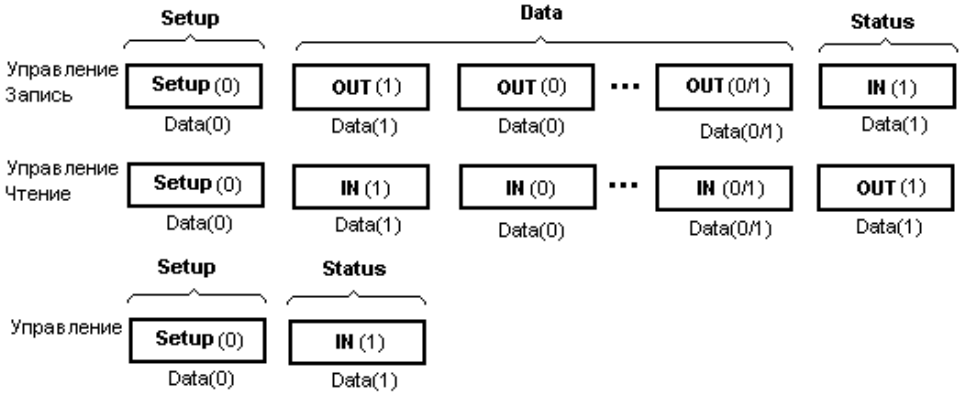


Рисунок 7.26 – Управляющие последовательности при записи и чтении данных

На рисунке 7.26 показаны последовательности пакетов управления и данных при записи и чтении информации.

На рисунке 7.27 показаны последовательности трехпакетной транзакции при записи и чтении данных.

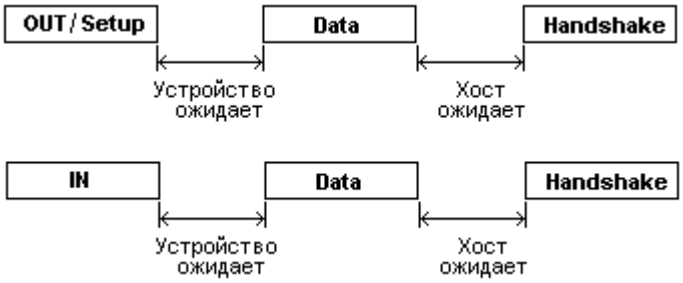


Рисунок 7.27 – Последовательности передачи пакетов

В 2002 году был введен улучшенный стандарт USB 2.0 (IEEE 1394) который превосходит по быстродействию стандартный интерфейс USB в 40 раз (480 Мбит/с, 12Мбит/с или 1,5Мбит/с). Скорость обмена определяется автоматически в диалоговом режиме. Быстродействие этого интерфейса позволило использовать его для подключения внешних жестких дисков, видеокамер, DVD-драйвов и пр.

В конце 2008 году утвержден новый стандарт SuperSpeed USB (USB 3.0), обеспечивающий скорость передачи данных 5 Гбит/сек, что в 10 раз превосходит быстродействие USB 2.0. SuperSpeed USB использует технологию Sync-N-Go, которая минимизирует время ожидания пользователя. Данный интерфейс полностью совместим с оборудованием USB 2.0.

Основной целью интерфейса USB 3.0 является повышение доступной пропускной способности, однако новый стандарт эффективно оптимизирует энергопотребление. Интерфейс USB 2.0 постоянно опрашивает доступность устройств, на что расходуется энергия. Напротив, у USB 3.0 есть четыре состояния подключения, названные U0-U3. Состояние подключения U0 соответствует активной передаче данных, а U3 погружает устройство в «сон». Если подключение бездействует, то в состоянии U1 будут отключены возможности приёма и передачи данных. Состояние U2 идёт ещё на шаг дальше, отключая внутренние тактовые импульсы. Соответственно, подключённые устройства могут переходить в состояние U1 сразу же после завершения передачи данных, что, как предполагается, даст ощутимые преимущества по энергопотреблению, если сравнивать с USB 2.0.

7.11. Мультимедийный интерфейс высокого разрешения

Мультимедийный интерфейс высокого разрешения HDMI (*High-Definition Multimedia Interface*) позволяет передавать цифровые видеоданные высокого разрешения и многоканальные цифровые аудио-сигналы с защитой от копиро-

вания HDCP (*High Bandwidth Digital Copy Protection*). Мультимедийный интерфейс HDMI имеет пропускную способность в пределах от 4,9 до 10,2 Гбит/с. Максимальная длина HDMI кабеля может составлять 15 метров, но обычно рекомендуется использовать кабели длиной не более 1.5-2 метров. Интерфейс HDMI предназначен для соединения источников цифрового аудио и видео с устройствами отображения — цифровыми ресиверами, усилителями звука, телевизорами, плазменными панелями и является новым стандартом подключения оборудования высокой четкости. HDMI позволяет передавать видео, как стандартного разрешения, так и высокой чёткости, а также передавать многоканальный звук.

Разъём HDMI имеет 19 контактов и обеспечивает цифровое соединение нескольких устройств с помощью соответствующих кабелей. В состав интерфейса входят следующие цепи:

- TMDS (*Transition-Minimized Differential Signaling*). Использует три канала (TMDS Data0 - TMDS Data2), передающие потоки аудио/видео и дополнительные данные (TMDS Clock), с пропускной способностью до 3,4 Гбит/с на канал.
- CEC (*Consumer Electronics Control*). Используются для передачи команд и управляющих сигналов между участниками связи. Функции CEC встраиваются по желанию производителя. Если все участники связи будут поддерживать HDMI CEC, то можно, например, посылать команды с пульта ДУ всей подключённой технике. Среди команд есть включение/выключение, воспроизведение, переход в режим ожидания, запись и другие.
- SCL (*Serial Data Clock*). Применяется для сигналов синхронизации передачи данных.
- SDA (*Serial Data Access*). Для доступа передачи данных.
- DDC (*Display Data Channel*). Служит для передачи спецификации дисплея, в частности, название производителя, номер модели, поддерживаемые форматы и разрешения и т.д.
- Цепи питания (+ 5 В), контроля подсоединения и экранов.

7.12. Источники питания компьютеров

Источник питания компьютера служит для преобразования напряжения 220 -250 В однофазной сети переменного тока в ряд вторичных напряжений постоянного тока, а также поддержания этих напряжений в заданных пределах при колебаниях напряжения питающей сети и изменении нагрузки. Типовыми вторичными напряжениями источника питания являются (+3,3 В; 20 А), (+5 В; 32 А), (+12 В; 16А), (-5 В; 0,5 А), (-12 В; 0,5 А).

В настоящее время в компьютерах применяются источники питания только импульсного типа, преимуществами которых являются малые габариты при большой мощности и независимость от изменения частоты и номинального напряжения сети. Принцип работы импульсных блоков питания заключается в выпрямлении сетевого напряжения с последующим преобразованием его в переменное высокочастотное напряжение прямоугольной формы, которое понижается трансформатором до нужных значений, выпрямляется и фильтруется и поддерживается (стабилизируется) в заданных пределах при возмущающих воздействиях. Структурная схема импульсного блока питания изображена на рисунке 7.28.

Рисунок 7.28 – Структурная схема импульсного блока питания

Входной фильтр предназначен для устранения кратковременных скачков питающего напряжения и импульсных помех, вызванных работой расположенных поблизости импульсных устройств (ВЧ помехи), а также для устранения помех, проникающие в сеть непосредственно от данного источника питания. В его состав входят обычно дроссели и высокочастотные конденсаторы.

Сетевой выпрямитель преобразует переменное напряжение электросети (110/230 вольт) в постоянное пульсирующее напряжение. Он представляет собой четыре диода, соединенных по мостовой схеме. Выпрямленное напряжение величиной порядка 310 В поступает на сглаживающий фильтр, состоящий из нескольких электролитических конденсаторов и дросселя.

Принципиальная схема выпрямителя типового блока питания компьютера с цепями фильтрации показана на рисунке 7.29. Конденсаторы C1-C3 и двухобмоточный дроссель на ферритовом сердечнике L1 образуют фильтр, защищающий компьютер от помех, которые могут проникнуть по сети и одновременно этот фильтр защищает сеть от помех, создаваемых компьютером. Резисторы R1, R4 и R5 выполняют функцию разрядников для конденсаторов фильтра после того как блок питания отключен от сети. Варистор R3 защищает блок

питания от перенапряжений в сети, а терморезистор R2 защищает входные цепи БП в момент включения.

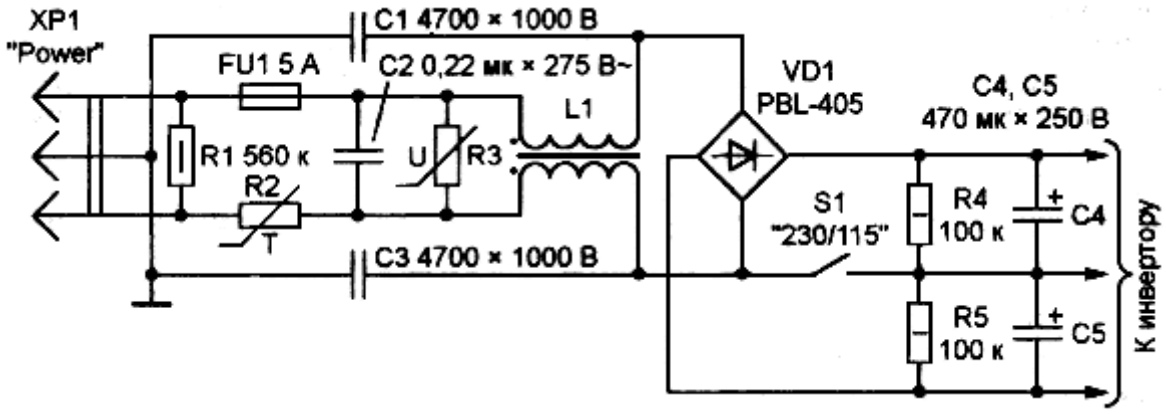


Рисунок 7.29 – Принципиальная схема выпрямителя с фильтрами

Выключатель S1 (**"230/115"**) предназначен для возможности работать от сети с напряжением 110...127 вольт. При замыкании контактов выпрямитель работает по схеме с удвоением напряжения и на его выходе образуется напряжение вдвое больше сетевого.

Отфильтрованное постоянное напряжение поступает на высоковольтный транзисторный ключ (инвертор), который переключается схемой управления с частотой несколько десятков килогерц (30 – 100 кГц). Из курса электротехники известно, что чем выше частота переменного тока в первичной обмотке трансформатора, тем меньше необходимый размер сердечника и число витков обмотки. Импульсы напряжения поступают на импульсный понижающий трансформатор, на вторичных обмотках которого и формируются напряжения для каналов +3,3 В, +5 В, +12 В, -5 В, -12 В. Каждый из каналов содержат двухполупериодный выпрямитель и LC-фильтр.

Инвертор (высокочастотный преобразователь) преобразует постоянное напряжение, полученное от выпрямителя в последовательность высокочастотных импульсов прямоугольной формы. Преобразователь состоит из мощных транзисторов, работающих в ключевом режиме. На управляющие электроды подаются прямоугольные импульсы положительной полярности и с высокой частотой следования (десятки кГц). С этой частотой транзисторы как ключевые элементы открываются и закрываются. Ключевые транзисторы устанавливаются на радиатор для принудительного охлаждения во время работы, а сам радиатор обдувается вентилятором.

К высокочастотному преобразователю относится и силовой понижающий импульсный трансформатор TV. Он понижает высокочастотное переменное напряжение от преобразователя до напряжений, требуемых для питания электронных узлов компьютера. Кроме этого, трансформатор обеспечивает гальваническую развязку от силовой электросети. Частота колебаний выбирается порядка 30 – 100 кГц. Благодаря высокой частоте размеры трансформатора существенно снижаются.

Схема управления формирует импульсы управления мощным инвертором, обеспечивая стабилизацию выходных напряжений, защиту от короткого замыкания на выходе и пр. Схема построена на основе широтно-импульсного модулятора (ШИМ). В ШИМ частота импульсного (прямоугольного) сигнала постоянная, а скважность (отношение периода следования импульса к его длительности) переменная. С помощью изменения скважности импульсов можно менять среднее напряжение на выходе ШИМ (рисунок 7.30).

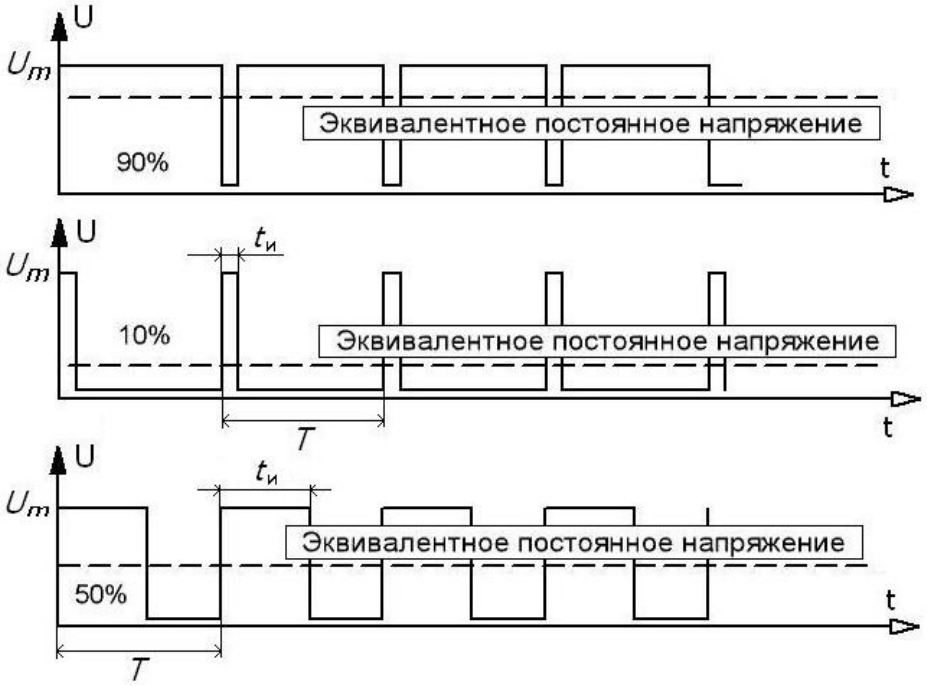


Рисунок 7.30 – Диаграммы сигналов ШИМ

Среднее значение выходного напряжения преобразователя при широтно-импульсном регулировании связано с напряжением питания соотношением

$$U_{ср} = (t_и/T)U_m = gU_m, \qquad (7.1)$$

где $g = t_и/T$ – коэффициент регулирования.

В соответствии с (2.1) диапазон регулирования выходного напряжения ИП с ШИР составляет от нуля ($t_и = 0$, $g = 0$) до U_m ($t_и = T$, $g = 1$).

Схема управления на основе ШИМ, применяемая в блоках питания компьютеров выполнена в виде интегральной схемы. На схему управления подаются тактовые импульсы с задающего генератора, а также сигналы отклонения выходного напряжения от опорного, формируемые измерительной цепью. Процесс регулирования выходного напряжения состоит в изменении (модуляции) ширины управляющих импульсов, подаваемых на ключевые транзисторы инвертора.

Выходные выпрямители выполняют преобразование переменного низковольтного напряжения в постоянное. Здесь же происходит фильтрация выпрямленного напряжения.

Для питания микросхемы ШИМ-контроллера, а также формирования дежурного напряжения +5 В, которое используется компьютером, когда он выключен, в схеме БП установлен отдельный преобразователь, на выходе которого имеется стабилизатор напряжения на +5 В.

Основными показателями качества подаваемого на компьютер питания являются:

- номинальные значения напряжений и параметры их допустимых отклонений, которые обычно не превышают 5%;
- величина пульсаций на выходе питающих каналов. Стандарт ATX допускает пульсации в пределах 120 мВ для шины 12 В и 50 мВ – для шины 5 В. Различают высокочастотные пульсации (на удвоенной частоте ключа основного преобразователя) и низкочастотные (на удвоенной частоте питающей сети).

Данные параметры качества не должны выходить за установленные пределы при максимальной нагрузке (токах), указанных в паспорте на блок питания и на его корпусе.

8. Технические средства ввода информации в компьютер

В качестве источников информации в информационных системах могут быть преобразователи (датчики) физико-химических процессов (температура, скорость, давление, влажность, соленость, количество кислорода и пр.), микрофоны, видеокамеры, а также текстовые и графические изображения, представленные на бумажных и других видах носителей.

Измерительные преобразователи подразделяются на **первичные** и **вторичные**. Измерительный преобразователь, на который непосредственно воздействует измеряемая физическая (или иная) величина называют первичным измерительным преобразователем. *Вторичный измерительный преобразователь* выполняет функцию звена, связывающего первичный измерительный преобразователь со вторичными устройствами (ЭВМ, сигнализаторами, регуляторами и др.). Для ввода контролируемых параметров в компьютер они должны быть сначала преобразованы в электрические сигналы (изменение напряжения или тока) и представлены в цифровом виде. Так как в информационных системах большинство входных параметров являются непрерывными процессами, а вторичным устройством является цифровая ЭВМ, то роль вторичного преобразователя выполняет аналого-цифровой преобразователь (АЦП).

В промышленности применяется огромное разнообразие первичных датчиков физических величин, каждый из которых имеет свой выходной сигнал. Чтобы избежать такого же разнообразия вторичных измерительных и регулирующих приборов, датчики оснащаются нормирующими преобразователями. Нормирующие преобразователи служат для:

- усиления слабых сигналов и формирования унифицированных уровней сигналов напряжения или тока;
- выполнения (при необходимости) линеаризации нелинейных характеристик первичного преобразователя;
- осуществления термокомпенсации, если первичный преобразователь подвержен сильному влиянию температуры, как, например, в случае с термопарами и емкостными датчиками влажности;
- снижения влияния электромагнитных помех.

Конструктивно нормирующие преобразователи размещают либо непосредственно в корпусе датчика, либо выполняют в виде независимых модулей.

Одна из основных задач нормирующих преобразователей состоит в том, чтобы преобразовать различные сигналы первичных преобразователей (термопар, термопреобразователей сопротивления, влажности, давления, веса, pH и проч.) в унифицированные сигналы постоянного тока или напряжения. Такими унифицированными сигналами являются сигналы, изменяющиеся в диапазоне по напряжению:

0 – 0,01 В; 0 – 1 В или 0 – 10 В;

по току:

0 – 5 мА; 0 – 20 мА или 4 – 20 мА.

Среди стандартных сигналов наиболее удобным и популярным является токовый сигнал 4-20 мА. Это связано с тем, что сигналы первичных преобразователей, как правило, очень малы. Например, сигналы термопар обычно меньше 50 мВ. В промышленных условиях сильные электромагнитные излучения могут создавать паразитные колебания, в сотни и тысячи раз превышающие полезные сигналы. По этой причине относительно большие токовые сигналы диапазона 4 - 20 мА являются более помехоустойчивые. Кроме того, могут быть снижены требования к величине сопротивления соединительных проводов. При работе с токовым сигналом от 4 до 20 мА также легко обнаружить обрыв линии связи (ток при обрыве будет равен нулю, т.е. выходить за пределы установленного диапазона).

8.1. Преобразователи изображений в электрический сигнал

Преобразователи изображений в электрический сигнал широко используются при вводе в информационные системы текстовых и неподвижных графических сообщений, а также подвижных видеоизображений. В качестве простейшего преобразователя оптического излучения, который преобразует попавший на его фоточувствительную область свет в электрический ток, используется фотодиод. Фотодиоды могут работать в одном из двух режимов – без внешнего источника электрической энергии (режим фотогенератора) либо с внешним источником электрической энергии (режим фотопреобразователя). Фотодиоды, работающие в режиме фотогенератора, преимущественно применяют в качестве источников питания, преобразующих энергию солнечного излучения в электрическую.

При работе фотодиода в фотопреобразовательном режиме источник питания Е включается в цепь в запирающем направлении (рисунок 8.1). Изменение светового потока вызывает пропорциональное изменение тока I в измерительной цепи и соответственно падения напряжения U на нагрузочном резисторе $R_н$.

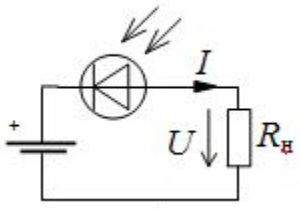
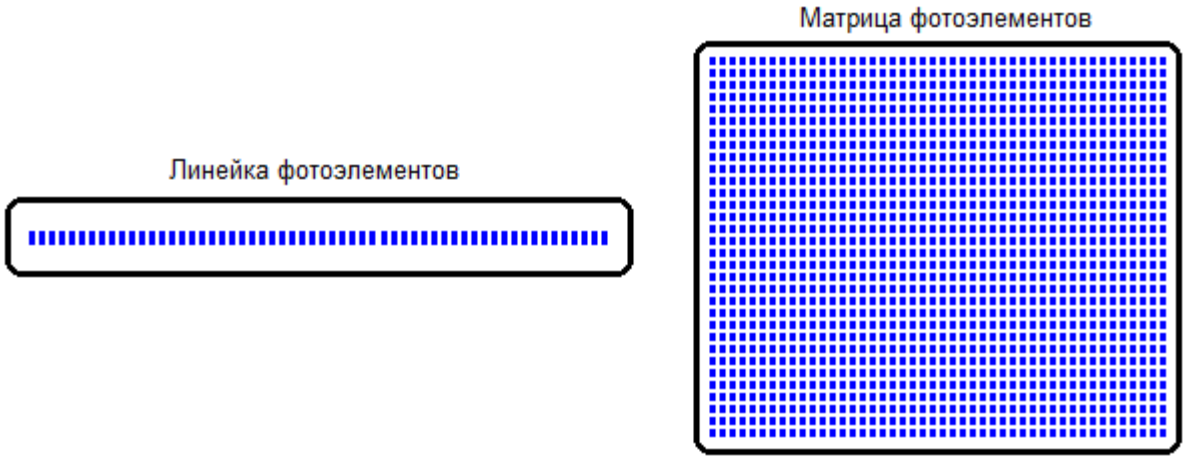


Рисунок 8.1 - Схема включения фотодиода в измерительную цепь

Кроме фотодиодных преобразователей в настоящее время широко используются светочувствительные устройства на основе приборов с зарядовой связью (**ПЗС**) (англ. CCD — *Charge-Coupled Device)* и КМОП-транзисторов (комплементарные металл-оксидные полупроводники). ПЗС **—** общее обозна-

чение класса полупроводниковых приборов, в которых применяется технология управляемого переноса заряда в объеме полупроводника. ПЗС- и КМОП-сенсоры представляют приборы, способные преобразовывать световую энергию в электрический заряд. В КМОП-сенсорах используются фотодиоды и полевые транзисторы с изолированным затвором с каналами разной проводимости. В информационных системах для повышения скорости ввода видеоинформации элементарные фотопреобразователи (фотосенсоры) объединяются в светочувствительную линейку (рис.8.2,а) или матрицу (рис.8.2,б), состоящих из элементарных фотоприемников (пикселей).



а) б)

Рисунок 8.2 – Расположение светоприемных элементов в линейке и матрице

В связи с тем, что ток любого фотосенсора пропорционален только яркости света, то изображение получается черно-белым. Для формирования цветного изображения на ячейки матрицы накладываются цветные фильтры. Каждый пиксел цветного изображения состоит из трех сенсоров, перед каждым из которых находится собственный цветофильтр - красный, синий и зеленый (RGB). В связи с тем, что человеческий глаз наиболее чувствителен к зеленому цвету. на матрице фильтры располагаются группами по четыре, таким образом, что на два зеленых приходится по одному синему и красному. Световые лучи разного спектра имеют разную длину волн, поэтому фильтр пропускает в ячейку лучи только своего цвета. Полученная картинка состоит только из пикселов красного, синего и зеленого цвета – именно в таком виде записываются файлы формата RAW (сырой формат).

Полупроводниковая КМОП-матрица снабжена системой микролинз, каждая из которых располагается непосредственно над пикселем и фокусирует падающий свет на фотодиод. С целью предотвращения попадания этого излучения на подложку, что может сгенерировать дополнительные заряды и привести к нежелательным наводкам, подложка защищена специальным непрозрачным металлическим слоем. Маленький электрический заряд, генерируемый фотоди-

одом, накапливается в конденсаторе. Заряд слишком слабый, чтобы быть использованным для самостоятельного использования, и нуждается в усилении до приемлемого для работы с ним уровня. Каждый пиксель в КМОП-сенсоре имеет свою собственную усилительную схему, поэтому усиление сигнала происходит еще до считывания изображения (рис.8.3).

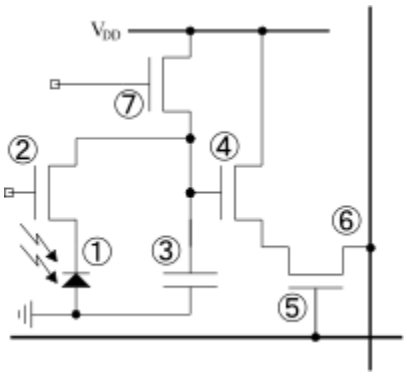


Рисунок 8.3 – Схема ячейки КМОП-матрицы

Цифрами на рис. обозначены следующие элементы: 1 — светочувствительный элемент (фотодиод); 2 — затвор МОП-транзистора; 3 — конденсатор, сохраняющий заряд с диода; 4 — усилитель; 5 — шина выбора строки; 6 — вертикальная шина, передающая сигнал процессору; 7 — сигнал сброса.

Процесс преобразования изображения осуществляется в такой последовательности:

- до съёмки подаётся сигнал сброса;
- во время экспозиции происходит накопление заряда фотодиодом;
- в процессе считывания происходит выборка значения напряжения на конденсаторе.

КМОП-сенсоры часто имеют дополнительную схему обработки изображения, включая аналого-цифровые преобразователи и процессоры сжатия прямо на чипе, делая процесс считывания и обработки изображения более быстрым и простым. Все это выражается в менее мощном чипе, увеличенной надежности, уменьшенном потреблении энергии и более компактном дизайне.

Появилась возможность произвольного доступа к каждому пикселю сенсора - подобно тому, как работает оперативное запоминающее устройство (по параллельной схеме). Такой доступ позволяет КМОП-сенсору считывать не всю матрицу целиком, а лишь некоторые области. Этот метод называется *методом оконного считывания*

Светочувствительная ячейка ПЗС-матрицы состоит из поликремниевого канала p-Si, расположенного на кремниевой подложке, и трех поликремнивых затворов G, а для изоляции затворов от поликремниевого канала применяется пленка окиси кремния SiO_2 (рис.8.4). При подаче комбинации напряжений на поликремневые затворы изменяются электрические потенциалы вблизи этих электродов, приводящие к образованию в канале потенциальной ямы.

До начала процесса экспонирования (засветки проецируемым изображением ПЗС-матрицы) подачей определённой комбинации напряжений на электроды затворов происходит сброс всех ранее образовавшихся в элементарных ячейках (пикселях) зарядов и приведение всех элементов в идентичное состояние.

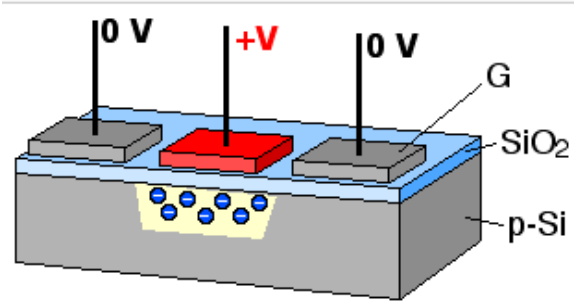


Рисунок 8.4 – Схематичное изображение ПЗС-ячейки

Далее комбинация напряжений на электродах создаёт потенциальную яму. Фотоны, проникающие в кремний в результате воздействия света при экспонировании изображения, приводит к генерации электронов, которые притягиваются потенциальной ямой и остаются в ней. Чем интенсивнее световой поток во время экспозиции, тем больше накапливается электронов в потенциальной яме, соответственно тем выше итоговый заряд данного пикселя.

После завершения экспонирования определенные последовательные изменения напряжений на электродах затворов формируют в каждом пикселе и рядом с ним распределение потенциалов, которое приводит к перетеканию заряда в заданном направлении, к выходным элементам матрицы.

После экспонирования осуществляется считывание зарядов ПЗС-элементов строки матрицы с помощью параллельно-последовательных регистров сдвига, которые преобразовывают строку зарядов на входе в серию импульсов на выходе. Данная серия представляет собой аналоговый сигнал, который в дальнейшем поступает на усилитель.

Параллельно-последовательный регистр сдвига в ПЗС-матрицах реализуется с помощью тех же самых ПЗС-элементов, объединённых в строку. Работа такого устройства базируется на способности **п**риборов с **з**арядовой **с**вязью ПЗС обмениваться зарядами своих потенциальных ям. Обмен осуществляется благодаря наличию специальных электродов переноса, расположенных между соседними ПЗС-элементами. После считывания всех зарядов из регистра на его параллельные входы подключается следующая строка матрицы, затем следующая и таким образом формируется дискретный аналоговый сигнал на основе двумерного массива ПЗС-ячеек.

Светочувствительные матрицы на основе ПЗС- и КМОП-структур имеют свои преимущества и недостатки. Основные преимущества КМОП-матриц перед ПЗС-матрицами заключаются в следующем:

- минимальное потребление энергии;
- высокая степень интеграции и размещение на базе чипа дополнительных блоков для обработки изображения;
- небольшая стоимость;
- возможность выборочного чтения определенной области матрицы.

Однако, при всех перечисленных преимуществах КМОП-сенсоров, ПЗС-системы формируют намного лучше изображение и по чёткости и по контрастности и по цветопередаче, они имеют меньший уровень шумов и более широкий динамический диапазон, что ограничивает применение в специальных областях недорогих КМОП-сенсоров.

Для концентрации и фокусировки изображения на плоскости светочувствительной матрицы в большинстве устройств ввода видеоинформации используется оптическая система (рис.3.5).

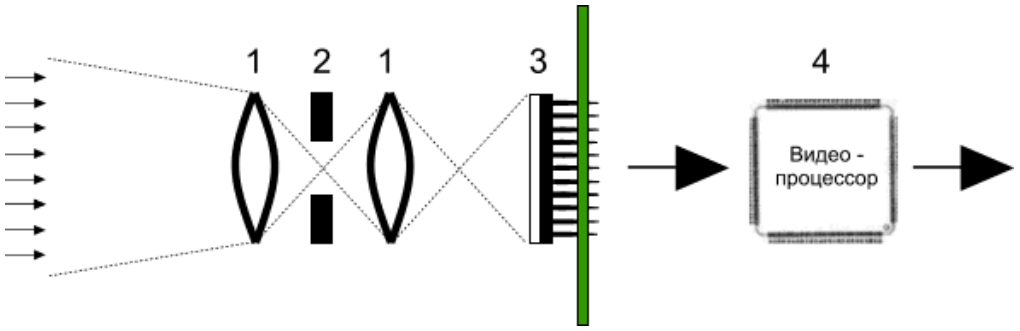


Рисунок 8.5 – Схема оптической системы видеокамеры

Оптическая система состоит и нескольких фокусирующих линз (1) и диафрагмы (2). Диафрагма позволяет изменять интенсивность светового потока путем изменения диаметра отверстия в непрозрачном диске. Сфокусированный видеопоток проецируется на фотоприемную ПЗС- или КМОП-матрицу (3). Далее, в светочувствительной матрице изображение преобразуется в электрический сигнал, который поступает уже в электронную часть (4) видеокамеры и там подвергается обработке. Электроника видеокамеры может, в свою очередь, управлять диафрагмой для регулировки яркости изображения.

3.2. Сканеры

Сканер — устройство, которое создаёт электронный цифровой образ неподвижного изображения. Полученное электронной изображение может быть сохранено как графический файл, или, если оригинал содержал текст, распознанный посредством программы распознавания текста и сохраненный как текстовый файл.

В зависимости от способа сканирования объекта и самих объектов сканирования существуют следующие виды сканеров:

Планшетные — наиболее распространённые, поскольку обеспечивают максимальное удобство для пользователя — высокое качество и приемлемую скорость сканирования. Представляет собой планшет, внутри которого под прозрачным стеклом расположен механизм сканирования.

Ручные — в них отсутствует двигатель, следовательно, объект приходится сканировать вручную, единственным его плюсом является дешевизна и мобильность, при этом он имеет массу недостатков — низкое разрешение, малую скорость работы, узкая полоса сканирования, возможны перекосы изображения, поскольку пользователю будет трудно перемещать сканер с постоянной скоростью.

Листопротяжные — лист бумаги вставляется в устройство и протягивается по направляющим роликам внутри сканера мимо лампы. Имеет меньшие размеры, по сравнению с планшетным, однако может сканировать только отдельные листы. Многие модели имеют устройство автоматической подачи, что позволяет быстро сканировать большое количество документов, причем в ряде моделей – с двух сторон за один прогон.

Планетарные — применяются для сканирования книг или легко повреждающихся документов. При сканировании нет контакта со сканируемым объектом (как в планшетных сканерах).

Барабанные — применяются в полиграфии, имеют большое разрешение (около 10 тысяч точек на дюйм). Оригинал располагается на внутренней или внешней стенке прозрачного цилиндра (барабана).

Слайд-сканеры — служат для сканирования плёночных слайдов; выпускаются как самостоятельные устройства, так и в виде дополнительных модулей к обычным сканерам.

Сканеры штрих-кода — предназначены для сканирования штрих-кодов товара на складах и в магазинах.

Основными характеристиками сканеров являются следующие.

Формат сканируемой поверхности: A4 (стандартный печатный лист), A3, слайд-сканеры под формат пленки 13x18 и 18x24 мм.

Оптическое разрешение — измеряется в точках на дюйм (dots per inch — dpi). Указывается два значения, например 600×1200 dpi, горизонтальное — определяется матрицей оптических датчиков, вертикальное — определяется количеством шагов двигателя на дюйм.

Скорость работы. Измеряется в страницах в минуту, при этом имеются в виду страницы определенного формата и определенное разрешение сканнера, из числа возможных.

Глубина цвета. Определяется качеством ПЗС-матрицы (CCD) и разрядностью АЦП. Измеряется количеством оттенков, которые устройство способно распознать. 24 бита соответствует 16777216 оттенков. Современные сканеры выпускают с глубиной цвета 24, 30, 36 бит. Несмотря на то, что графические адаптеры пока не могут работать с глубиной цвета больше 24 бит, такая избыточность позволяет сохранить больше оттенков при преобразованиях картинки в графических редакторах.

Рассмотрим принцип действия планшетных сканеров, как наиболее распространённых моделей. Упрощенная схема планшетного сканера показана на рис.8.6.

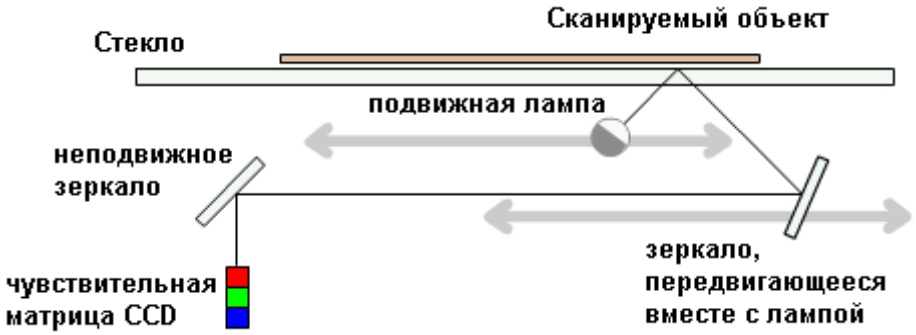


Рисунок 8.6 – Схема планшетного сканера

Сканируемый объект кладётся на стекло планшета сканируемой поверхностью вниз. Под стеклом располагается подвижная лампа, движение которой осуществляется шаговым двигателем. Свет, отражённый от сканируемого объекта, через систему зеркал и объектив попадает на чувствительную матрицу фотоприемников далее на АЦП и передаётся в компьютер. За каждый шаг двигателя сканируется полоска страницы, потом все полоски объединяются программным обеспечением в общее изображение.

Основной деталью планшетного сканера является считывающая головка, двигающаяся вдоль сканируемого изображения. Важнейшей частью считывающей головки является фотоприемник. На сегодняшний день наиболее распространены два типа фотопринимающей матрицы: ПЗС-матрица (прибор с зарядовой связью, в английских обозначениях — CCD, *Couple-Charged Device*) и КДИ-матрица (контактный датчик изображения, в анг. — CIS, *Contact Image Sensor*), построенная на основе КМОП-транзисторов.

Основой элемента ПЗС-матриц является фототранзистор, выполненный по технологии МОП (металл-оксид-полупроводник). ПЗС-матрица состоит из множества миниатюрных датчиков, преобразующих падающий на них свет в пропорциональный его интенсивности электрический заряд. Эта технология используется и во многих других приборах для считывания изображений, от мощнейших телескопов до приборов ночного видения.

Принципиального различия между КДИ- и ПЗС матрицами нет. КДИ-сканеры отличаются от ПЗС-сканеров тем, что в них матрица постороена на основе КМОП-транзисторов и растянута на всю ширину рабочей области, поэтому в них полностью отсутствует оптическая система.

В КДИ-модификациях сканеров источник освещения заменяется светодиодами. При этом для цветного сканирования возникает необходимость в трех светодиодах на пиксел, в соответствии со стандартным разложением цвета RGB. Зеркала и объектив в КДИ-сканерах не представлены, так как эта техно-

логия обеспечивает прямую проекцию полной поверхности рабочей области прямо на считывающую матрицу.

Излучение, идущее от светодиодов, отражается от оригинала и, пройдя через линзу, фокусируется на датчике изображения. Датчик изображения - фототранзисторы, сделанные на основе МОП–технологии. В результате получается аналоговый сигнал, который усиливается в видеоусилителе и идет в АЦП.

Для сканирования полноцветного изображения используются три светодиода на один элемент датчика: красный, зеленый и синий, — которые при сканировании включаются по очереди.

ПЗС-сканеры обладают рядом преимуществ, в частности:

- Лучшая глубина резкости. Глубина резкости КДИ-сканеров ±0,3 мм, тогда как для сканеров с ПЗС она равна ±3 мм. Это означает, что трехмерные предметы, находящиеся на расстоянии 3 мм от общего уровня, будут нормально отсканированы ПЗС-сканером, а изображение, полученное КДИ-сканером, будет нерезким и размытым. На практике такими предметами зачастую являются развернутые толстые книги.

- Дольше срок службы. Сканер на основе ПЗС обеспечивает стабильное и неизменное качество в течение 10 000 часов работы, тогда как у КДИ-сканеров после 500 часов работы происходит падение яркости на величину до 30%.

8.3. Сенсорные экраны

Сенсорный экран — устройство ввода информации, представляющее собой экран, реагирующий на прикосновения к нему. Сенсорные экраны представляют собой наиболее удобный способ взаимодействия человека с компьютером. Применение сенсорных экранов имеет ряд преимуществ, недоступных при использовании любых других устройств ввода: увеличенный размер экрана за счет уменьшения количества других управляющих элементов на корпусе; повышенную надежность, устойчивость к жестким внешним воздействиям, более удобный и красочный интерфейс с использованием пиктограмм на рабочем столе. Преимуществ использования сенсорных технологий в портативных устройствах множество, некоторые из них очевидны:

Сенсорный экран состоит из следующих элементов: собственно экран (панель), контроллер и интерфейс. Панель представляет собой прозрачный многослойный экран, плоский или повторяющий форму поверхности монитора. С внутренней стороны имеется поддерживающее стекло, придающее конструкции необходимую жесткость. По периметру экрана расположены элементы механического крепления и контакты для съема электрических сигналов.

Контроллер — блок, преобразующий исходный сигнал (аналоговый или цифровой) к виду, удобному для дальнейшей обработки.

Интерфейс — узел контроллера, состоящий из разъема, соединительного кабеля, а также драйвера (например, конвертора сигналов ТТЛ в RS-232 и обратно). Он предназначен для передачи информации от контроллера к главному

управляющему узлу системы, например, к компьютеру. Наиболее часто экраны оснащаются интерфейсами USB, в некоторых устройствах используются устаревшие интерфейсы RS-232 или RS-485. Для подключения к управляющему микроконтроллеру в специализированных приборах сенсорные экраны оснащаются последовательными интерфейсами типа I^2C или SPI.

Существуют несколько типов построения сенсорных экранов:

- Резистивные:
  - 4-проводные;
  - 5-проводные;
  - 8-проводные;
- Емкостные;
- Цифровые;
- На поверхностных акустических волнах (ПАВ или SAW);
- Инфракрасные.

Резистивные экраны — наиболее популярный и отработанный в технологическом плане вид экранов. Исторически — это самый первый тип сенсорных панелей. За годы, прошедшие с момента появления первых экземпляров, конструкция резистивных экранов претерпела много изменений и ныне является надежной и самой дешевой. Наибольшую популярность имеют 4- и 5-проводные конструкции.

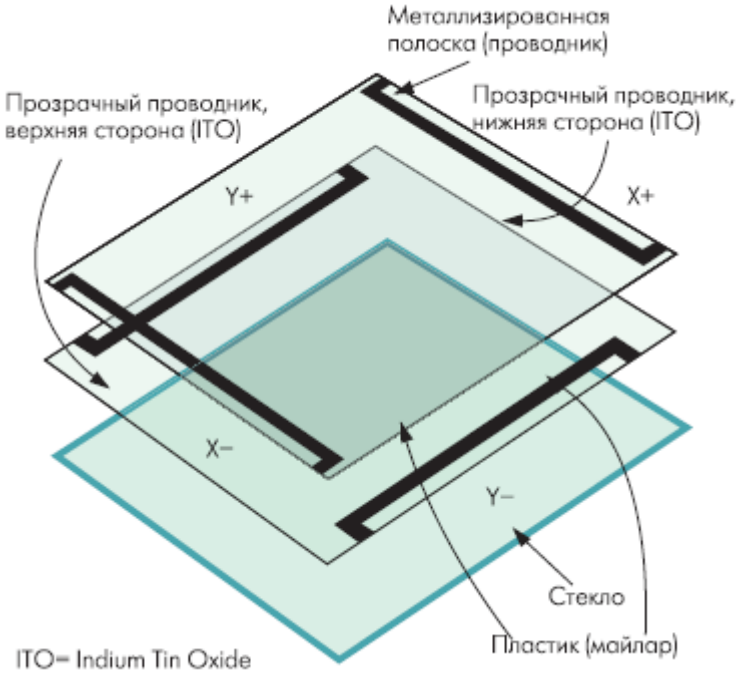


Рисунок 8.7 – Конструкция 4-проводного сенсорного экрана

4-проводная панель устроена следующим образом. Два слоя прозрачного и прочного пластика (обычно полиэстер или майлар) покрываются прозрачной токопроводящей пленкой на основе двуокиси индия и олова (**ITO**-*Indium Tin*

Oxide). Эти пластины устанавливаются таким образом, чтобы проводящие слои на каждом из них были обращены друг к другу. Между ними вносятся изолирующие упругие микроскопические шарики (спейсеры), не позволяющие поверхностям соприкасаться друг с другом при отсутствии внешних сил (рис.8.7). Токопроводящие покрытия обладают электрическим сопротивлением. При нанесении их стараются сделать максимально однородными по всей плоскости, чтобы тем самым обеспечить равномерность распределенного сопротивления. Если теперь на электроды одной плоскости (на металлизированные полоски по краям пластика) подать напряжение, то оно распределится между полюсами так же равномерно и однородно. Эквивалентная схема проводящих слоев экрана изображена на рис.8.8в,.

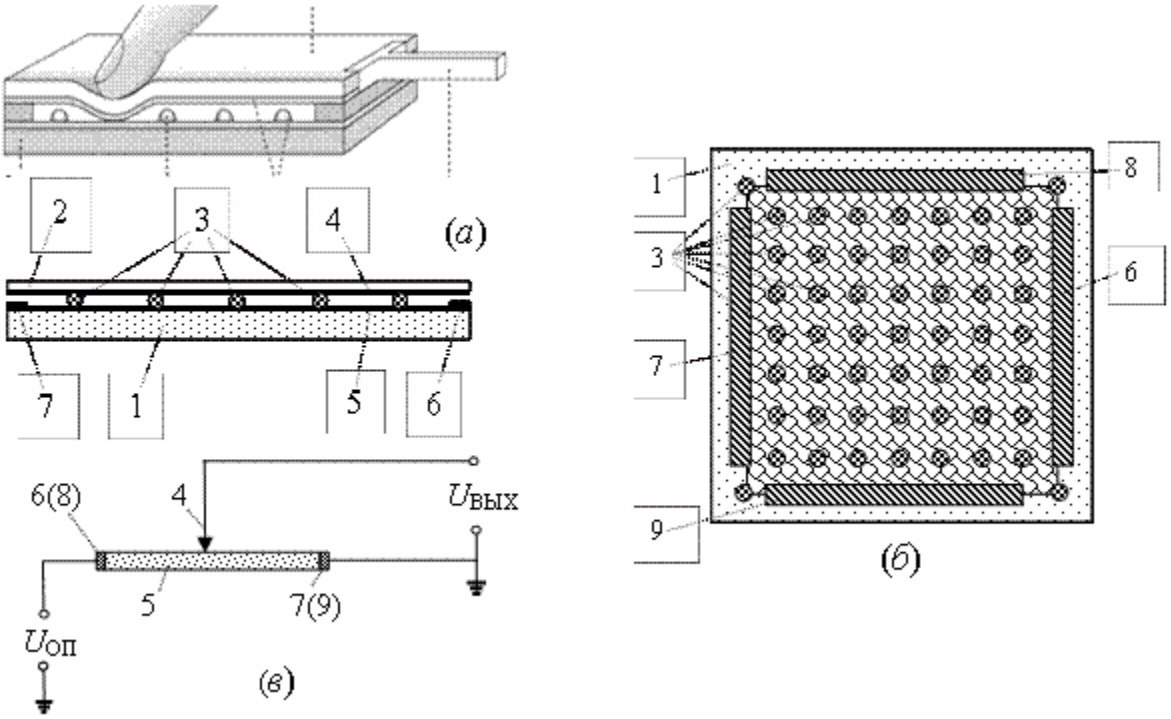


Рисунок 8.8 – Схема 4-проводного сенсорного экрана

В момент прикосновения к экрану плоскости войдут в контакт. Предположим, что вначале напряжение прикладывается к полюсам верхней (рис. 1) плоскости. Вторая плоскость с проводящим покрытием, металлизированные полоски-электроды которой подключены к входам АЦП, служит своеобразным щупом и может использоваться для снятия напряжения в точке контакта. Таким образом, вычисляется координата X. Затем источники напряжения и входы преобразователя переключаются, меняясь местами. Теперь напряжение прикладывается к металлическим полоскам на нижней плоскости, а потенциал точки соприкосновения снимается при помощи верхней плоскости. АЦП выдаст код пропорциональный координате Y.

Обычно полное сопротивление одной плоскости экрана колеблется в диапазоне от 100 до 900 Ом. При подаче на экран напряжения питания, допустим,

3 В, входные каскады контроллера должны будут в момент касания принимать ток больше 30 мА.

Рассмотренный тип сенсорных экранов наиболее часто применяется в конструкциях портативных приборов, медицинских прикроватных мониторах и специальном оборудовании, так как такие экраны дешевы, достаточно надежны, откликаются на прикосновение любым предметом. Они не предъявляют особых требований к окружающей среде и вполне могут применяться в полевых условиях. Недостатком сенсоров резистивного типа является высокая чувствительность к любым механическим воздействиям и повреждениям.

На рис.8.9 показана схема аналоговой части ИС контроллера 4-проводных экранов. В контроллер входят 12-разрядное АЦП последовательного приближения с простым и дифференциальными входами, входные ключи, главный и дополнительный аналоговый мультиплексоры. С помощью последнего производится выбор опорного источника напряжения для АЦП. Результат преобразования сохраняется в специальном регистре, который, наряду с регистрами управления, на схеме не показан. В процессе выполнения измерений АЦП может работать либо с внутренним, либо с внешним источником опорного напряжения. Во втором случае в качестве опорного может служить напряжение питания сенсорной панели.

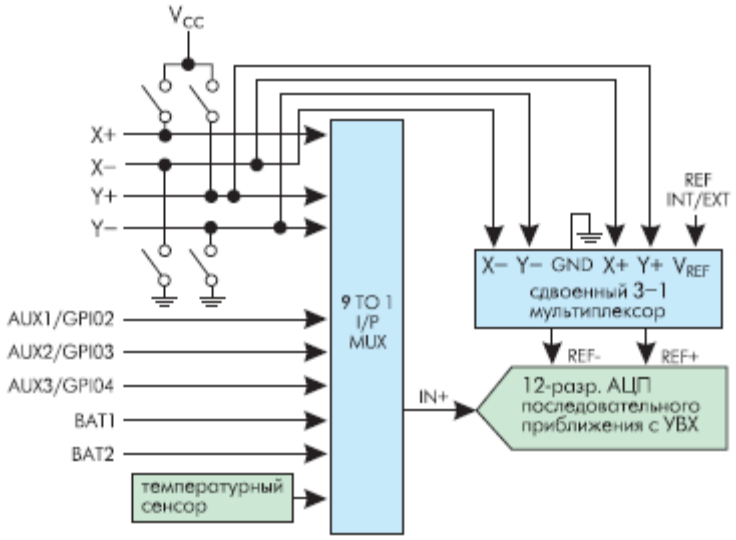


Рисунок 8.9 – Схема аналоговой части контроллера AD7877

В 5-проводной панели отсутствует одна из пленок, покрытых токопроводящим составом ITO. В данном случае он наносится на стекло. Таким образом, в 5-проводных экранах стекло не только придает конструкции необходимую жесткость, но и является сенсорной плоскостью.

На всех четырех сторонах прямоугольной стеклянной подложки делаются тонкие несоприкасающиеся металлизированные полоски-проводники, на которые поочередно и попарно подается напряжение. Одна пара электродов, размещенных на противоположных сторонах стеклянной подложки, необходима для измерений координаты по оси X, а другая — по оси Y. Вторая плоскость (май-

ларовая пленка) служит только лишь в качестве электрического щупа, подобного щупу вольтметра. Между двумя проводящими плоскостями, так же как и в 4-проводных конструкциях, имеются изолирующие упругие шарики, не позволяющие поверхностям соприкасаться при отсутствии внешнего давления.

Вся конструкция после сборки герметизируется и становится невосприимчивой к колебаниям влажности. Исключение из устройства одной гибкой поверхности позволило существенно улучшить качественные характеристики экрана и увеличить его надежность. Возросла не только механическая прочность изделия, но и упростилась, стала более надежной электрическая схема контроллера. Из нее исчез мультиплексор-коммутатор, переключавший в 4-проводных схемах проводники двух плоскостей, то подавая на них напряжение питания, то подключая их к входам АЦП. 5-проводные сенсорные экраны стали способны функционировать даже при небольших повреждениях проводящих поверхностей. С течением времени могут возникнуть небольшие неоднородности в проводящем покрытии пленки. Это происходит из-за постоянного выгибания пленки в моменты касания. Но поскольку, в отличие от 4- и 8-проводных панелей, эта плоскость в определении координат участвует очень мало, то и требования к ее однородности не столь велики и не сказываются на точности работы. От нее требуется лишь одно — передача напряжения из точки контакта на входы АЦП, а это вполне возможно при небольших повреждениях однородности покрытия.) Лучшие образцы способны выдерживать до 35 миллионов касаний. Это означает, что если к экрану будут прикасаться 10 раз в минуту (или 600 раз в час, сутки напролет и 7 дней в неделю), то его ресурса хватит на 6,5 лет непрерывной работы. В действительности же, конечно, срок службы устройства будет больше, так как столь частое и беспрерывное прикосновение к экрану вряд ли когда-либо потребуется.

8-проводные экраны по исполнению похожи на 4-проводные, но с целью компенсации деградации и технологических отклонений резистивных слоев панели, в конструкцию введены дополнительные проводники. 4 провода подпаяны к тем же самым четырем металлизированным полоскам по краям обоих плоскостей. Через них на проводящие поверхности подается напряжение смещения, компенсирующее уход параметров экрана после выполнения начальной калибровки экрана. В результате подводки опорного напряжения стабильность работы панели возрастает, однако в целом надежность ее не увеличивается. Также как и 4-проводные экраны, они способны выдерживать от 10^5 до 10^6 касаний.

Точность 4-, 5- и 8-проводных экранов примерно одинакова. Производители сенсорных панелей заявляют о том, что в их продукции стандартная плотность точек касания достигает 100 000 точек/дюйм (то есть различимый интервал между точками равен 0,00025 мм). Столь высокая точность позволяет использовать данную технологию в прецизионных системах ввода данных. Можно даже рисовать на экранах, вводить подписи под документами и т. п.

Резистивный экран требует периодической калибровки. Для ее выполнения пользователю предлагается под управлением специальной программы при-

коснуться к нескольким точкам на экране. Эти точки будут последовательно возникать на мониторе. Калибровка необходима по причине изменение с течением времени характеристик экрана В итоге меняется точность вычисления координат и позиционное положение некоторых точек экрана.

Емкостные сенсорные экраны. В емкостных сенсорах используется два набора электродов (горизонтальные и вертикальные), изолированных друг от друга стеклом (рис.8.10). Контроллер последовательно подает на каждый из электродов напряжение и измеряет амплитуду возникающего импульса тока. При прикасании пальцем к экрану емкость электродов, находящихся под ним, меняется, соответственно становится больше и импульс напряжения, который должен подать контроллер. Координата касания — это электрод, получивший приращение емкости.

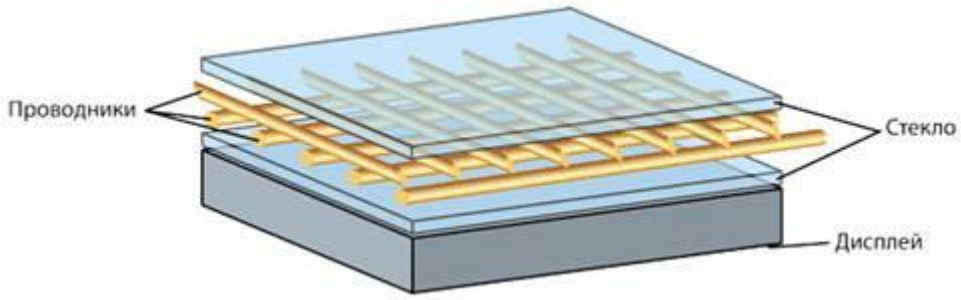


Рисунок 8.10 – Устройство емкостного сенсорного экрана

Важное преимущество ёмкостных сенсорных экранов — это способность их сохранять практически 90 % от изначальной яркости дисплея. Из-за этого изображения на ёмкостном экране смотрятся более чёткими, чем на экранах, имеющих резистивную конструкцию.

Матричные сенсорные экраны. На двух плоскостях, обращенных друг к другу проводящими поверхностями, нанесено прозрачное проводящее покрытие, но не сплошным равномерным слоем, как было в резистивных панелях, а полосками. Между полосками на каждой плоскости есть небольшой изолирующий промежуток. На одной из плоскостей они имеют горизонтальную ориентацию, а на другой — вертикальную. Вместе прозрачные проводники образуют прямоугольную координатную сетку. Плоскости разделены микроскопическими изолирующими шариками-спейсерами, подобно тому, как это делается в резистивных панелях. В момент прикосновения между двумя поверхностями в точке касания возникает электрический контакт. Контроллер периодически сканирует столбцы и строки сетки и, обнаружив контакт, сообщает управляющей программе координаты.

У экранов, выполненных по данной технологии, есть множество важных преимуществ перед всеми другими. Главные из них — независимость от изме-

нений температуры. Они не требуют настройки и калибровки, как резистивные. Они нечувствительны к пыли, и им не мешают посторонние источники света. Им не страшны колебания влажности. Им не мешают посторонние акустические шумы. Сканирующая цифровая схема контроллера много проще аналого-цифровой, дешевле и надежнее.

Однако у матричных экранов есть один существенный недостаток — сравнительно низкое разрешение. Очевидно, что точность определения координаты задается сеткой. Некоторые фирмы, разрабатывающие матричные сенсорные экраны утверждают, что достигают точности 0,1 мм. Однако лучшие промышленные экраны имеют разрешение 3,7 мм.

Сенсорные экраны на поверхностных акустических волнах. Сенсорные экраны на поверхностных акустических волнах (ПАВ) устроены следующим образом. В углах сенсорного экрана размещается специальный набор пьезоэлектрических элементов, на которые подается электрический сигнал частотой 5 МГц. Этот сигнал преобразуется в ультразвуковую акустическую волну, направляемую вдоль поверхности экрана, а сам экран представляется для программы управления сенсорными датчиками в виде цифровой матрицы, каждое значение которой соответствует определенной точке экранной поверхности. В ограничивающую экран рамку вмонтированы так называемые отражатели, распространяющие ультразвуковую волну таким образом, что она охватывает все рабочее пространство сенсорного экрана. Специальные рефлекторы фокусируют ультразвук и направляют его на приемный датчик, который снова преобразует полученное им акустическое колебание в электрический сигнал (рис. 8.11). Даже легкое касание экрана в любой его точке вызывает активное поглощение волн, благодаря чему картина распространения ультразвука по его поверхности несколько меняется. Управляющая программа сравнивает принятый от датчиков изменившийся сигнал с хранящейся в памяти компьютера цифровой матрицей - картой экрана, и вычисляет исходя из имеющихся данных координату касания, причем значение координаты рассчитывается независимо для вертикальной и горизонтальной оси.

У панелей, сделанных по этой технологии, точность определения координат высока, но при этом они чувствительны к качеству поверхности, наличию жира, грязи.

Известно два типа исполнения мониторов с экранами на ПАВ: а) излучатели и приемники устанавливаются на дополнительном закаленном стекле и б) они монтируются прямо на мониторе. У каждого подхода есть свои сильные стороны. Отсутствие каких-либо покрытий позволяет иметь яркие, сочные изображения на мониторе даже при обычной яркости и контрастности. С другой стороны, наличие дополнительного особо прочного стекла делает устройство более стойким к механическим воздействиям.

Сенсорный экран на ПАВ позволяет отслеживать не только координаты касаний, но и судить о силе нажатия на экран. Такое действительно возможно, так как при прочих равных условиях сила нажатия отражается на амплитуде

перемещающихся по экрану поверхностных волн. Большее усилие приводит к большему же затуханию колебаний.

Рисунок 8.11 – Определения координаты касания в экране на ПАВ

Средний ресурс экрана по данным компании GeneralTouch (Taiwan), составляет 50 млн касаний, что превышает ресурс 5-проводных резистивных панелей, но в несколько раз меньше, чем у емкостных экранов. Недостаток состоит в слабой защищенности от факторов окружающей среды, поэтому экраны с поверхностно-акустическими волнами нельзя применять на улице, а кроме того, такие экраны боятся любых загрязнений, блокирующих их работу. Применяются редко.

Инфракрасные сенсорные экраны. Инфракрасные сенсорные экраны имеют в своем составе элементы, способные реагировать на инфракрасное излучение. Две линейки светодиодов, размещенных вдоль горизонтальной и вертикальной сторон экрана, и две линейки фотодиодов с противоположных сторон контролируют поверхность панели. Появление любого предмета, касающегося экрана, изменяет поток излучения. Контроллер в каждый произвольный момент времени «знает» номер строки или столбца и «знает», следовательно, от какой именно пары светодиодов сигнал не пришел. Тем самым он определяет и координату касания.

Панели выполняются как отдельная рамка, которая не имеет никаких стекол или пленок над поверхностью монитора и поэтому не снижает яркость и контрастность наблюдаемой картинки.

Современные экраны на ИК-лучах имеют разрешение, достигающее 16 точек/см^2 , то есть излучатели и приемники размещаются в рамке с шагом 4 элемента на сантиметр. Применение специальных схемных методов и про-

граммных алгоритмов интерполяции позволяет определять координату с точностью, в 4 раза превышающей начальную — 64 точки/см^2

Приведенная величина много меньше, чем точность у резистивных панелей, но она выше, чем у матричных экранов. По этой причине экраны на ИК могут применяться в тех приложениях, где высокая точность не требуется, но и точности матричных панелей уже недостаточно. При выборе данной технологии учитывают также простоту монтажа экрана и отсутствие светопоглощения.

Индукционные сенсоры реагируют только на специальное перо. Применяются, когда требуется реакция именно на нажатия пером (а не рукой): художественные планшеты, некоторые модели планшетных ПК. Принцип работы основан на использовании резонансных катушек индуктивности, расположенных внутри сенсорной панели и катушки, помещенной внутрь стилуса. Внутри сенсорной панели размещается матрица индуктивных катушек, которые формируют на ее поверхности электромагнитное поле. Индукционная система в сенсорной панели производит возбуждение резонансного контура в наконечнике пера. Этот резонансный контур изменяет параметры катушек индуктивности под панелью. Специальный контроллер сканирует матрицу катушек и по изменению тока в катушке определяет координаты расположения контура пера относительно опорных точек возбуждающей индуктивной системы панели.

Схема резонансного контура, расположенного внутри стилуса, изображена на рис.8.12. Катушка индуктивности вместе с конденсаторами образует резонансный контур с изменяемой резонансной частотой.

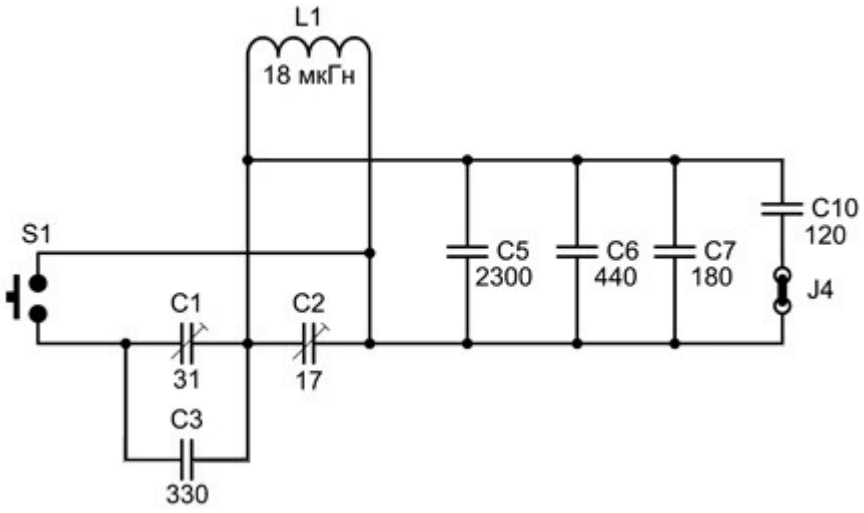


Рисунок 8.12 – Схема внутреннего устройства стилуса для индуктивного сенсорного экрана

Резонансную частоту можно изменить либо изменением емкости (дополнительный конденсатор подключается через кнопку, и соответственно, реагирует на ее нажатие), либо через изменение индуктивности — за счет изменения расстояния между двумя частями сердечника, на котором намотана катушка.

Такие сенсорные экраны используется в основном в ноутбуках именно из-за наличия возможности рисовать и писать от руки. В зависимости от силы

нажатия на стилус изменяется активность катушки индуктивности: чем выше давление, тем толще на экране получается линия.

С помощью стилуса можно легко эмулировать нажатие правой кнопки мыши. Для этого на стилусе предусмотрена кнопка, подключающая дополнительные конденсаторы (или витки катушки) к резонансному контуру. Электромагнитные поля, возникающие в процессе работы индукционной сенсорной панели, не нарушают работу схем управления ЖК-дисплеем и не влияют на характеристики самого дисплея. При свободной фронтальной поверхности дисплея обычно используются антибликовые фильтры, применение которых позволяет существенно повышают качество изображения ЖК-дисплея. Индуктивный экран не влияет на качество изображения, не реагирует на касание ладонью при письме или рисовании и широко применяется в мобильных устройствах, например в планшетных компьютерах, а также в образовательных учреждениях, дизайнерских студиях.

8.4. Оптическая мышь

Компьютерная мышь — механический манипулятор, преобразующий движение в управляющий сигнал. В частности, сигнал может быть использован для позиционирования курсора или прокрутки страниц. Получила широкое распространение в связи с появлением графического интерфейса пользователя на персональных компьютерах.

По принципу действия мышки подразделяются на механические устройства с шаровым приводом и оптические. В шаровом приводе движение мыши передается на выступающий из корпуса обрезиненный стальной шарик (его вес и резиновое покрытие обеспечивают хорошее сцепление с рабочей поверхностью). Два прижатых к шарику ролика снимают его движения по каждой из осей измерений и передают их на датчики угла поворота (инкрементальные энкодеры), преобразующие эти движения в электрические сигналы.

В настоящее время в компьютерных системах применяются преимущественно оптические мышки. Оптические мыши подразделяются на светодиодные (рисунок 8.13,а и лазерные (рисунок 8.13,б). Принцип работы обоих типов оптической мыши практически одинаков и состоит в следующем.

С помощью светодиода и системы фокусирующих его свет линз под мышью подсвечивается участок поверхности. Отраженный от этой поверхности свет, в свою очередь, собирается другой линзой и попадает на приемный сенсор микросхемы — процессора обработки изображений. Этот чип, в свою очередь, делает снимки поверхности под мышью с частотой несколько кГц. Причем микросхема (оптический сенсор) не только делает снимки, но сама же их и обрабатывает, так как содержит две ключевых части: систему получения изображения Image Acquisition System (IAS) и интегрированный DSP процессор обработки снимков.

а) б)

Рисунок 8.13 – Схема оптической системы компьютерной мышки

На основании анализа ряда последовательных снимков (представляющих собой квадратную матрицу из пикселей разной яркости), интегрированный DSP- процессор высчитывает результирующие показатели, свидетельствующие о направлении перемещения мыши вдоль осей X и Y, и передает результаты своей работы вовне по последовательному порту.

Структурная схема оптической мышки состоит из нескольких блоков (рисунок 8.14), а именно:

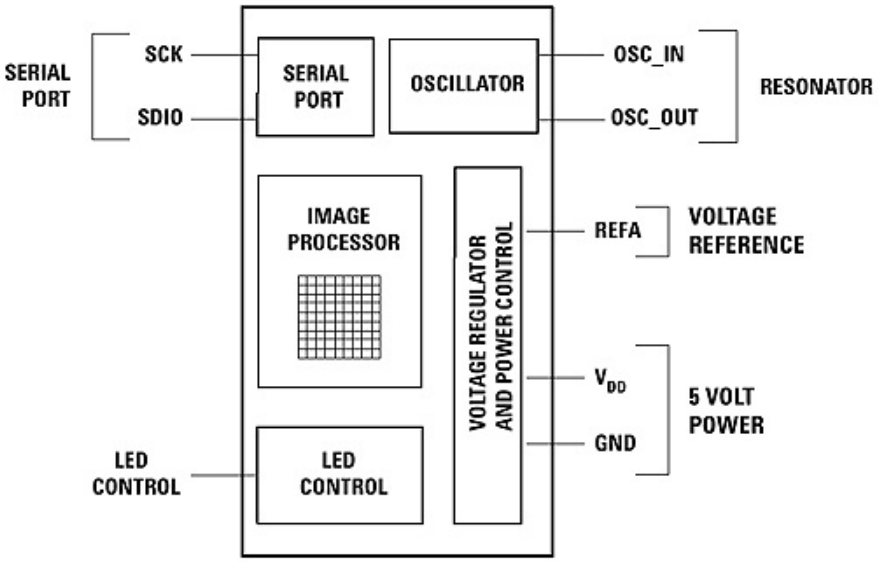


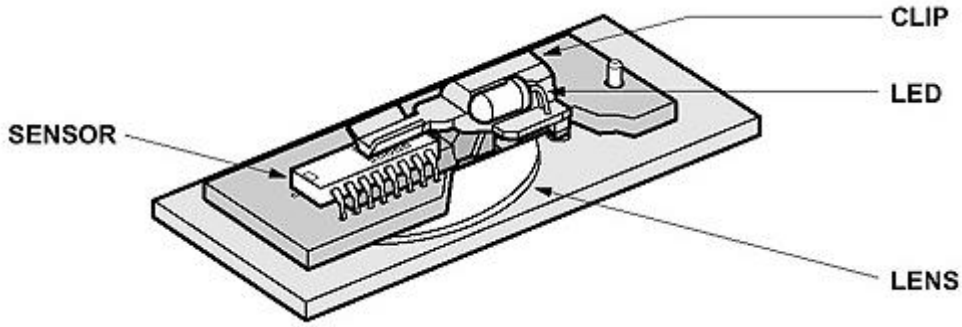
Рисунок 8.14 – Структурная схема оптической мышки

- **Image Processor** — процессор обработки изображений (DSP) со встроенным приемником светового сигнала (IAS);
- **Voltage Regulator And Power Control** — блок регулировки напряжения и контроля энергопотребления (в этот блок подается питание и к нему же подсоединен дополнительный внешний фильтр напряжения);
- **Oscillator** —задающий генератор тактовых импульсов, стабилизированный кварцевым резонатором, частота резонатора 2-24 МГц;

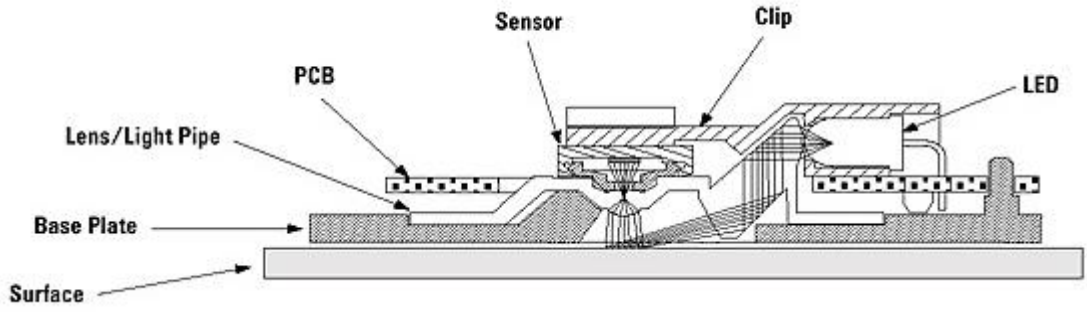
- **Led Control** —блок управления светодиодом, с помощью которого подсвечивается поверхность под мышью;
- **Serial Port** — блок передающий данные о направлении перемещения мыши вовне микросхемы.

Система оптического слежения мышек, помимо микросхемы-сенсора, включает еще несколько базовых элементов (рисунки 8.15,а и б). Конструкция включает держатель (Clip) в который устанавливаются светодиод (LED) и непосредственно сама микросхема сенсора (Sensor). Эта система элементов крепится на печатную плату (PCB), между которой и нижней поверхностью мыши (Base Plate) закрепляется пластиковый элемент (Lens), содержащий две линзы.

Информацию о перемещении мыши микросхема оптического сенсора передает через Serial Port не напрямую в компьютер. Данные поступают к еще одной микросхеме-контроллеру, установленной в мыши. Эта вторая «главная» микросхема в устройстве отвечает за реакцию на нажатие кнопок мыши, вращение колеса прокрутки и т.д. Данный чип, в том числе, уже непосредственно передает в ПК информацию о направлении перемещения мыши, конвертируя данные, поступающие с оптического сенсора, в передаваемые по интерфейсам PS/2 или USB сигналы..



а)



б)

Рисунок 8.15 – Конструктивное исполнение оптической мышки

Компьютер, используя драйвер мыши, на основании поступившей по этим интерфейсам информации, перемещает курсор-указатель по экрану монитора. В некоторых моделях оптических мышек сенсорная часть и контроллер кнопок и колеса прокрутки выполнены в виде одной микросхемы.

9. Технические средства вывода информации

4.1. Струйные принтеры

Струйная печать — это процесс получения изображения, при котором его элементы создаются капельками чернил, вылетающими из сопла со скоростью достаточной, чтобы преодолеть зазор между соплом и поверхностью, на которой формируется картинка. Струйные технологии разделяются на непрерывную и импульсную. Последняя, в свою очередь, включает печать твердыми чернилами, пьезоэлектрическую и пузырьковую.

В первых устройствах струйной печати (60-е годы XX ст.) использовался непрерывный процесс подачи капелек чернил. Такое название способ печати получил потому, печатающая головка непрерывно выстреливает капельками чернил в сторону бумаги. Схема принтера с непрерывной печатью изображена на рис. 9.1.

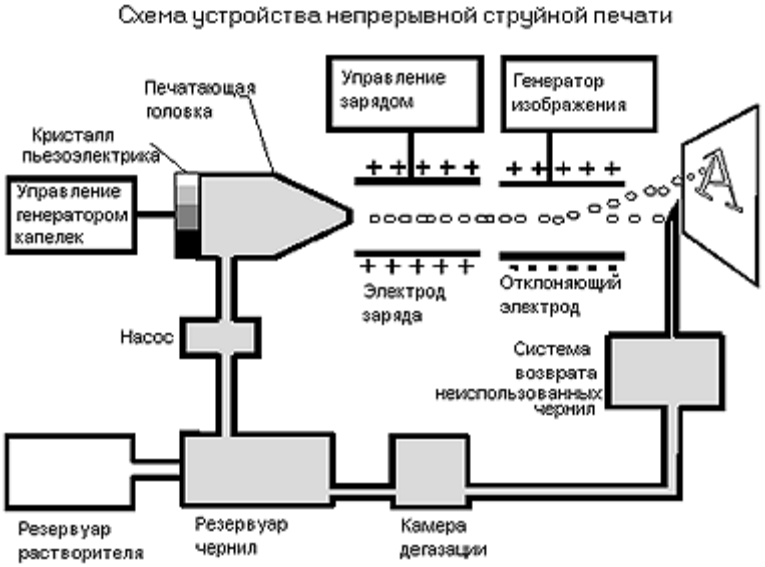


Рисунок 9.1 – Схема струйного принтера с непрерывной печатью

Поток чернил, поступающий из резервуара в печатающую головку, разбивается на капельки под воздействием вибрации сопла, вызываемой с помощью, например, пьезоэлектрического элемента. На бумагу должны попадать только те капельки, которые требуются для создания изображения, остальные — это отходы производства. С помощью цилиндрического охватывающего электрода вылетающие из сопла капельки приобретают электрический заряд. Далее они пролетают через отклоняющую систему, в которой действует электрическое поле высокого напряжения. Так как капельки получили заряд, то под действием электрического поля они изменяют свою траекторию. Таким образом, меняя напряжение электрического поля, можно управлять направлением полета капелек. Они либо попадают на бумагу в нужном месте, либо летят в уловитель, откуда поступают в резервуар для повторного использования. Устройства непрерывного действия имеют следующие особенности:

1) высокая производительность сопел — от 50000 до 150000 капелек в секунду на сопло;
2) возможность получать цветные изображения очень высокого качества.
3) относительно невысокая скорость печати;
4) требуются только электропроводные чернила;
5) имеют сложную систему рециркуляции чернил;
6) расстояние между соплом пишущей головки и поверхностью, на которой создается изображение, достаточно велико.

На изображения с непрерывным способом печати совершенно неразличимы глазом точки, из которых оно сформировано. За это приходится расплачиваться малой скоростью получения изображения, высокими эксплуатационными расходами (дорогие чернила и сложность в обслуживании). Однако отдельно установленные емкости для чернил позволяют контролировать их расход и осуществлять пополнение, что заметно снижает стоимость печати. Кроме того, для дозаправки не требуется извлечение картриджа из устройства.

В настоящее время преимущественно используются импульсные струйные технологии печати. Достигнутый в этой области прогресс, приведший к повышению качества печати, снижению ее себестоимости и удешевлению печатных устройств, способствовал тому, что струйные принтеры в большинстве случаев вытеснили матричные и успешно конкурируют с лазерными.

В устройствах печати с твердыми чернилами в печатающую головку закладываются четыре цветные твердые красящие палочки, соответствующие базовым цветам — голубая, пурпурная, желтая и черная. Нагреватели нагревают твердые чернила и при температуре 90 градусов они переходят в жидкое состояние, стекая в резервуары с подогревом, где чернила поддерживаются в жидкой фазе во время работы принтера. Для получения изображения печатающее устройство откачивает небольшое количество чернил из резервуара и затем дополнительно нагревает его. Механизм большинства таких принтеров устроен аналогично принтерам непрерывного действия. Схема печати твердыми чернилами показана на рис.9.2.

Бумага закреплена на вращающемся барабане, а печатающая головка формирует изображение за один проход печатающей головки над бумагой. Электронное устройство, обеспечивающее пульсацию чернил, выстреливает мельчайшими капельками только в те мгновения, когда это требуется. При контакте с бумагой чернила мгновенно переходят в твердую фазу, поэтому они не впитываются, а остаются на ее поверхности. С одной стороны, это очень хорошо, так как полностью отсутствует эффект расплывания чернил, присущий любым жидким чернилам, но, с другой стороны, так как капельки застывают мгновенно, поверхность изображения становится шершавой. Поэтому при финишной обработке лист бумаги прокатывают через валки, расплющивающие шероховатости твердых чернил, придавая изображению приятный глянцевый вид.

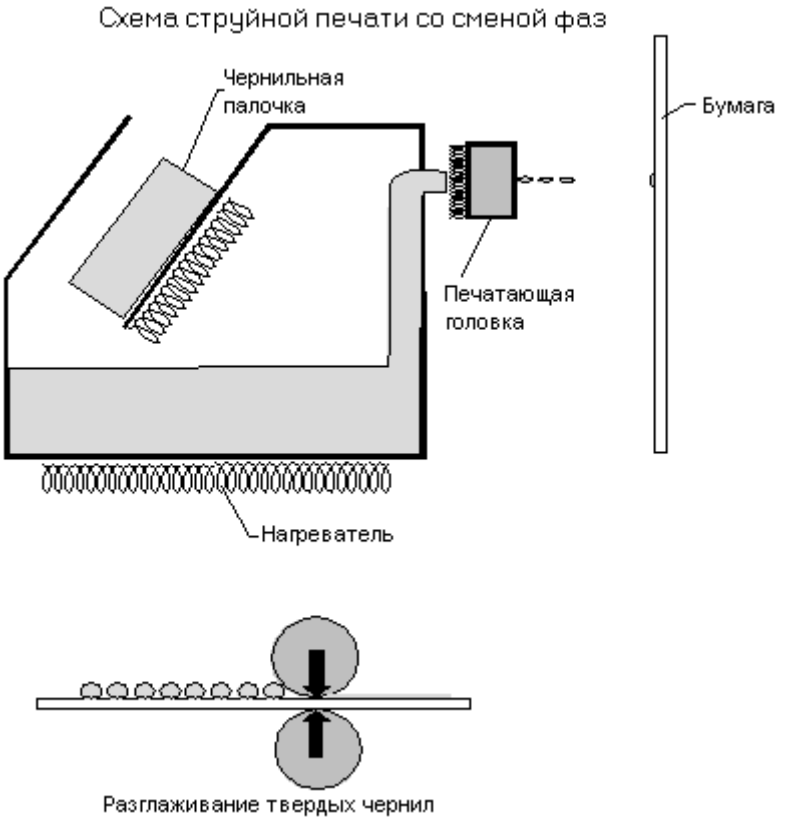


Рисунок 9.2 – Схема печати твердыми чернилами

Основное преимущество струйной печати со сменой фаз перед другими струйными технологиями в том, что чернила не впитываются в бумагу и удается достичь высокого качества печати. Недостаток один - высокая стоимость получаемых изображений. Для одноцветной печати это совершенно непрактичные устройства. Их стоит применять для вывода полноцветных изображений, когда требуется очень точная цветопередача и высокое качество печати.

В 70-х годах XX столетия для принтеров были разработаны пьезоэлектрические струйные головки. В таких принтерах избыточное давление в камере с чернилами создается с помощью диска из пьезоэлектрика, который изменяет свою форму (выгибается) при подведении к нему электрического напряжения (рис.9.3). Выгнувшись, диск, который служит одной из стенок камеры с чернилами, уменьшает ее объем. Под действием избыточного давления жидкие чернила вылетают из сопла в виде капли. Пионером пьезоэлектрической технологии явилась фирма Epson, однако она не смогла успешно соревноваться в объеме продаж со своими конкурентами Canon и Hewlett-Packard из-за сравнительно высокой технологической стоимости пьезоэлектрических печатающих головок. В принтерах Canon и Hewlett-Packard применены пузырьковые печатающие головки, которые заметно дешевле пьезоэлектрических.

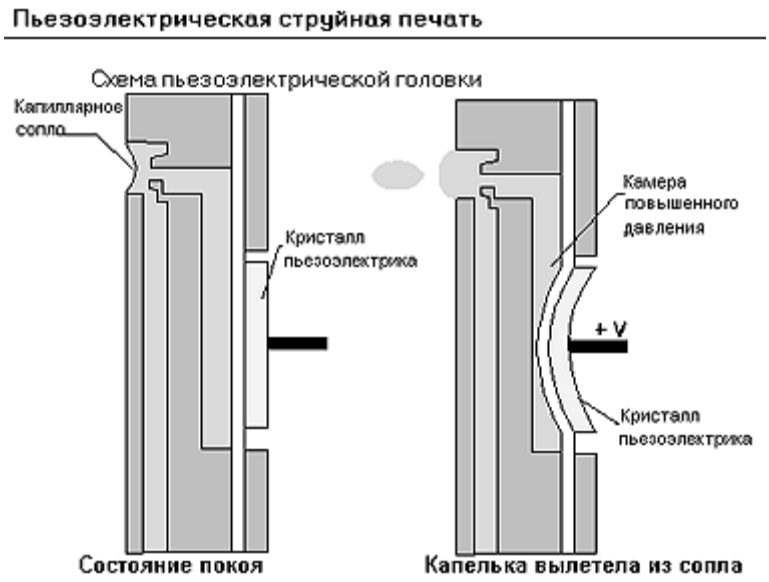


Рисунок 9.3 – Схема функционирования пьезоэлектрической печатающей головки

Фирма Hewlett-Packard создала первый струйный принтер с использованием пузырьковой технологии ThinkJet в 1985 году. В печатающих системах, использующих струйную пузырьковую технологию, текстовое и графическое изображение воспроизводятся при попадании на бумагу капелек чернил, вылетевших из очень тонких сопел. Функционирование принтера осуществляется следующим образом. В стенку сопла встроен нагревательный элемент. При подаче на него электрического импульса его температура резко возрастает. Поэтому практически все чернила, находящиеся в контакте с нагревательным элементом, мгновенно испаряются. Расширение пара вызывает ударную волну. Под действием избыточного давления капелька чернил "выстреливается" из сопла. После "выстрела" чернильный пар конденсируется, пузырек схлопывается и в сопле образуется зона пониженного давления, под действием которого всасывается новая порция чернил. Важной особенностью такого печатающего устройства является простота конструкции сопел. Конструкция печатающей головки позволяла легко достичь разрешения в 300 точек на дюйм. Причем, кроме низкой стоимости изготовления, такие устройства имеют ряд других преимуществ:

1) высокая надежность каждого сопла упрощает конструкцию и, следовательно, уменьшает размер печатающего узла, так как не надо обеспечивать возможность замены сопел;

2) сопла можно располагать гораздо ближе друг к другу, а это увеличивает разрешение печати;

3) бесшумная работа печатающей головки.

Сейчас Canon и Hewlett-Packard владеют большинством патентов на эту технологию, а путем обмена лицензиями им удалось захватить практически весь мировой рынок.

Полученные при помощи струйной печати изображения относятся к растровым, то есть состоят из точек, каждая из которых может иметь только один цвет — цвет используемых чернил. Например, для получения 256 оттенков серого на черно-белом устройстве вывода для каждой исходной точки требуется использовать миниатюрную матрицу из 16x16 элементов, каждый из которых может быть только белым или черным. Если элементы невелики, то для человеческого глаза они сливаются в одну точку. Закрашивая больше или меньше элементов, можно добиться ощущения появления оттенков серого на черно-белом изображении. Точно так же, смешивая точки различных цветов (голубой, пурпурный и желтый) в одной матрице, можно получить различные их оттенки. Естественно, растрирование снижает разрешающую способность принтера, поскольку каждая эффективная точка изображения состоит, к примеру, из 256 физических, то есть эффективное разрешение падает в 16 раз. Для того чтобы избежать этого, производителям приходится прибегать к различным достаточно сложным методам наложения растров.

9.2. Лазерные принтеры

В лазерных принтерах применяется принцип электрофотографии. Суть его состоит в первоначальном создании определенной полярности электростатического образа изображения на фоточувствительном барабане. Затем осуществляется контакт светочувствительного барабана с тонером, несущим заряд противоположного знака. При этом частицы тонера притягиваются к участкам поверхности фоточувствительного барабана, имеющим электрический заряд, образуя видимое изображение. После этого порошковое изображение переносится с барабана на бумагу и закрепляется на ней. Для закрепления тонера на бумаге используются термосиловой метод, при котором копия проходит между двумя разогретыми валиками, тонер расплавляется и фиксируется на поверхности бумаги, или термический - копия проходит под ИК-лампой. В этом случае тонер расплавляется и застывает без какого-либо механического воздействия. Упрощенная схема лазерного принтера изображена на рис.

Сердцем лазерного принтера является фотобарабан (рис.9.4), который также часто называют печатающим барабаном. С помощью барабана производится перенос изображения на бумагу. Он представляет собой металлический цилиндр, покрытый тонкой пленкой фотопроводящего полупроводника. Поверхности этого покрытия можно придать положительный или отрицательный заряд, который сохраняется на поверхности, но только до тех пор, пока барабан не освещен. Наиболее часто применяемый материал фотобарабана — фотоорганика. Фотоорганика требует использования отрицательного заряда, однако есть материалы (например, кремний), позволяющие использовать положительный заряд.

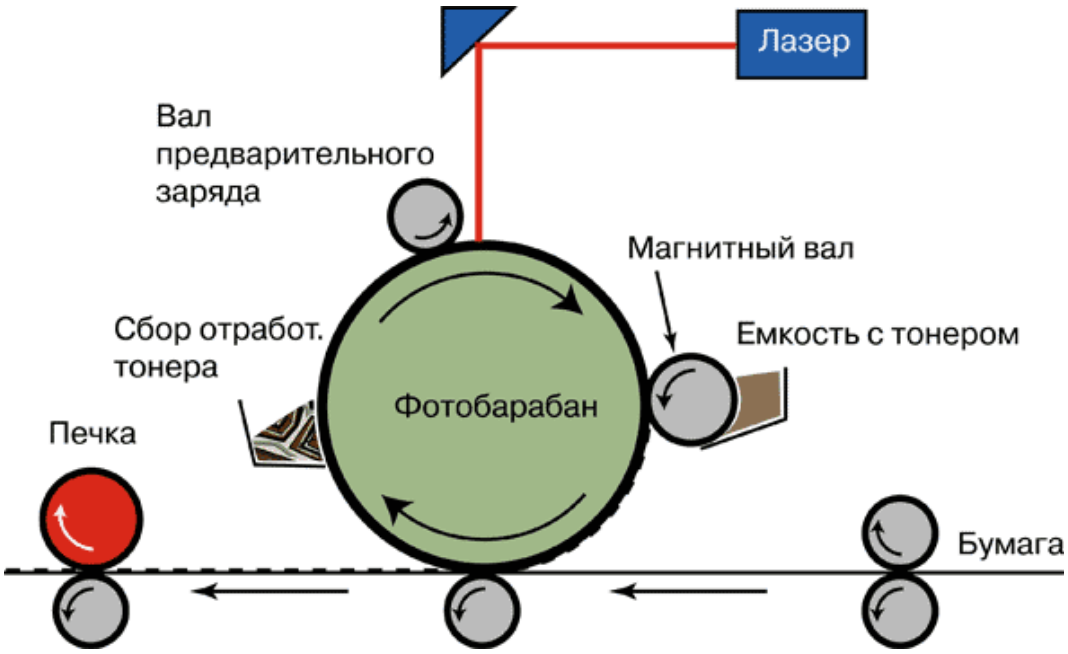


Рисунок 9.4 – Схема устройства лазерного принтера

Для нанесения на поверхность барабана заряда, применяется тонкая проволока, называемая "коронирующим проводом". На этот провод подается высокое напряжение, вызывающее возникновение светящейся ионизированной области вокруг него, которая и называется короной и придает барабану необходимый статический заряд. Между проводом и фотобарабаном обычно помещается металлическая сетка, служащая для выравнивания электрического поля. Позже стали применять зарядку с помощью вала предварительного заряда. Такая система позволила уменьшить напряжение и снизить проблему выделения озона в коронном разряде (преобразование молекул O_2 в O_3 под действием высокого напряжения), однако влечёт проблему прямого механического контакта и износа частей, а также чистки от загрязнений. Если фотобарабан не освещать, то заряд может сохраняться на его поверхности достаточно долгое время. При освещении какой-либо части поверхности барабана покрытие приобретает проводимость и заряд стечет с освещенного участка, образовав незаряженную зону. Роль источника света играет малогабаритный лазер, генерирующий тонкий световой луч, который направляется на шестигранное вращающееся зеркало. Отраженный от зеркала луч прочерчивает строку на поверхности заряженного фотобарабана и разряжает заряженную поверхность барабана. Чтобы получилось изображение, лазер включается и выключается управляющим микроконтроллером. Под действием луча лазера участки фоточувствительной поверхности фотобарабана, которые были засвечены лазером, становятся электропроводящими, и часть заряда на этих участках «стекает» на металлическую основу фотобарабана. Тем самым на поверхности фотобарабана создаётся строка электростатического изображения будущего отпечатка в виде «рисунка» из участков с менее отрицательным зарядом, чем общий фон.

После формирования строки изображения, специальный прецизионный шаговый двигатель поворачивает барабан так, чтобы можно было формировать

следующую строку. Это смещение равняется разрешающей способности принтера и обычно составляет 1/300 или 1/600 дюйма. В каждой строке на каждый дюйм приходится по 300 или 600 точек. Таким образом и получается "лазерное" разрешение в 300×300 (600×600) dpi. После нанесения электростатического изображения на всю рабочую поверхность барабана выполняется процесс «проявления» изображения, который происходит при контакте фоточувствительного барабана с тонером, несущим заряд противоположного знака. Электрофотографический тонер представляет собой порошок черной или цветной легкоплавкой смолы. При этом частицы тонера притягиваются к участкам поверхности фотобарабана несущим заряд, образуя видимое изображение. Для переноса тонера на фотобарабан служит еще один ролик — ролик подачи тонера. Чтобы частицы тонера расположились на нем тонким ровным слоем, используется дозирующее лезвие.

Следующим этапом является перенос тонера (а, значит, и изображения) на бумагу. Бумага вытягивается из подающего лотка и с помощью системы валиков перемещается к печатающему барабану. В месте контакта фотобарабана с бумагой, под бумагой находится ещё один ролик, называемый роликом переноса. На него подаётся положительный заряд, который он сообщает и бумаге, с которой контактирует. Частички тонера, войдя в соприкосновение с положительно заряженной бумагой, переносятся на неё и удерживаются на поверхности за счёт электростатики.

Перед самым барабаном бумаге сообщается статистический заряд с помощью еще одного коронирующего провода, подобного тому, что используется для подготовки барабана к экспонированию. Затем бумага прижимается к поверхности барабана. Заряды разной полярности, накопленные на поверхности бумаги и на поверхности барабана, вызывают перенос частиц тонера на бумагу и их надежное прилипание к ней. Остатки неперенесенного на бумагу тонера собираются с фотобарабана специальным скребком (ракелем) или чистящим валиком и затем возвращается в контейнер с тонером.

Для закрепления тонера на бумаге используются термосиловой метод, при котором копия проходит между двумя разогретыми валиками, тонер расплавляется и фиксируется на поверхности бумаги, или термический - копия проходит под инфракрасной лампой. За температурой термовала следит термодатчик (термистор). Печка представляет собой два соприкасающихся вала, между которыми проходит бумага. При нагреве бумаги (180-220 °C) тонер, притянутый к ней, расплавляется и в жидком виде вжимается в текстуру бумаги. Выйдя из печки, тонер быстро застывает, что создаёт постоянное изображение, устойчивое к внешним воздействиям. Чтобы бумага, на которую нанесён тонер, не прилипала к термовалу, на нём выполнены отделители бумаги.

Процесс лазерной печати схематично изображен на рис.9.5.

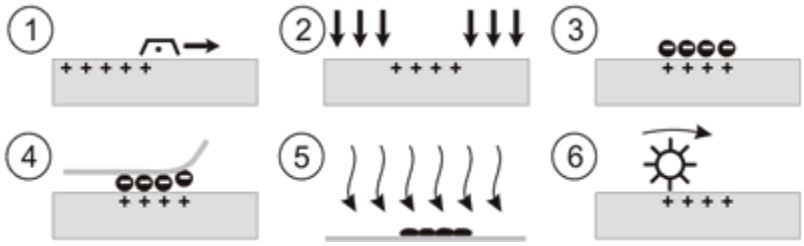


Рисунок 9.5 – Схема лазерной печати

Она осуществляется за 6 этапов:

1) зарядка светочувствительного барабана коронирующим устройством;
2) экспонирование светом (создание скрытого изображения);
3) проявление скрытого изображения тонером (частицы которого имеют заряд, противоположный заряду на светочувствительном барабане);
4) перенос изображения на бумагу;
5) термическое закрепление тонера на бумаге;
6) нейтрализация заряда на барабане и его очистка.

Трехмерное схематичное изображение лазерного принтера показано на рис.9.6.

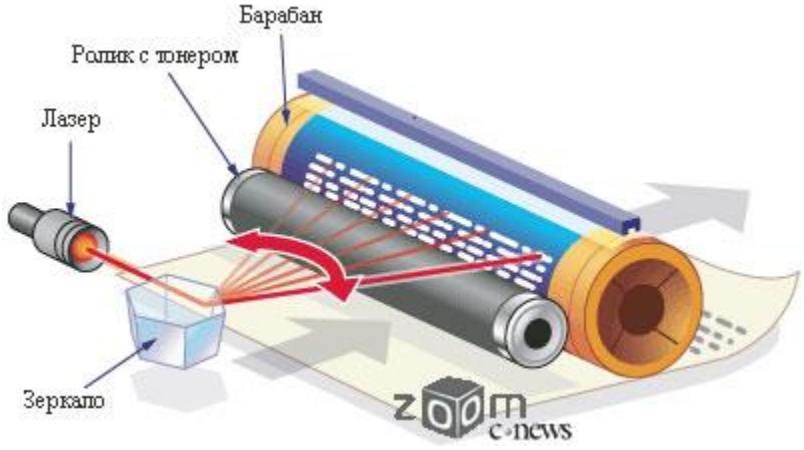


Рисунок 9.6 – Трехмерная схема лазерной печати

Для создания цветного изображения принтер должен сформировать на бумаге 4 накладывающихся друг на друга изображения, каждое из которых будет окрашено в свой цвет: голубой, пурпурный, жёлтый или чёрный. Это основные полиграфические цвета, участвующие в субтрактивной модели создания цветного изображения. Существуют 2 различных способа создания полноцветного изображения: многопроходная и однопроходная технология.

Многопроходная технология подразумевает наличие в принтере промежуточного носителя (т.н. ремня переноса изображения) на который на каждом из проходов попадает изображение своего цвета (рис. 4.8). После формирования всех четырёх изображений готовая полноцветная картинка переводится с

ремня переноса на бумагу. Такая технология очень хорошо отработана и на сегодня она используется в основном в самых младших моделях цветных лазерных принтеров, что позволяет делать их весьма дешёвыми.

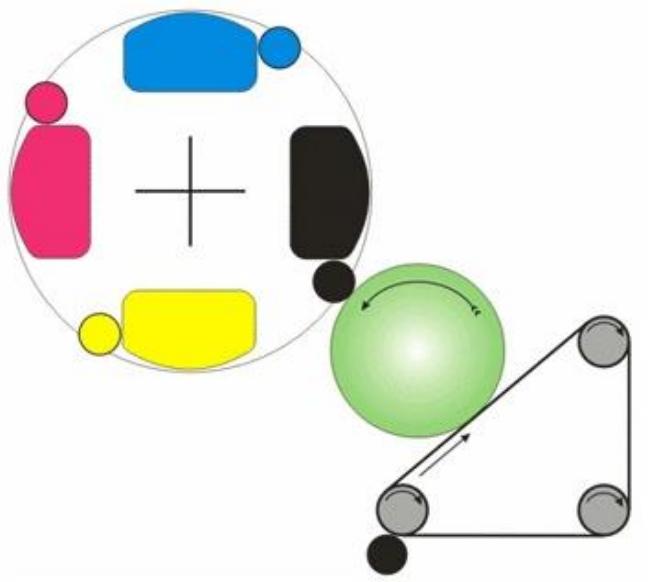


Рисунок 9.7 – Схема многопроходного цветного принтера

Одним из основных недостатков такой технологии считается сравнительно низкая скорость цветной печати. Так как для формирования полноцветного изображения механизм принтера вынужден совершить 4 рабочих хода. Кроме того, в силу достаточно большого количества подвижных элементов внутри принтера, при работе такого механизма создаётся много шума (особый вклад в это вносит вращающийся револьвер с тонер-картриджами). Скорость чёрно-белой печати таких принтеров обычно приближается к скорости печати хороших сетевых принтеров, а себестоимость чёрно-белой печати практически равна себестоимости печати на обычном чёрно-белом принтере.

Однопроходная печать (в наиболее характерной своей реализации, в том числе используемой и в цветных принтерах Оки) подразумевает наличие в принтере четырёх печатных механизмов, расположенных в ряд (тандемный тип) и создающих полноцветное изображение непосредственно на бумаге за один проход. Бумага движется на транспортном ремне через принтер и проходит последовательно под каждым из четырёх цветных фотобарабанов, с которых на неё переносится тонер, в результате чего за один проход создаётся полностью сформировавшееся цветное изображение. Такой способ формирования изображения позволяет достигать весьма высокой скорости цветной печати, в 3-4 раза превышающей скорость печати многопроходных принтеров.

Список рекомендованной литературы

Основная

1) Авдеев В.А. Периферийные устройства. Интерфейсы, схемотехника, программирование [Электронный ресурс] / Авдеев В.А.— Электрон. текстовые данные.— М.: ДМК Пресс, 2009.— 848 с.— Режим доступа: http://www.iprbookshop.ru/6929
2) Бройдо В.Л. Архитектура ЭВМ и систем: Учебник для вузов / В.Л. Бройдо, О.П. Ильина. — СПб.: Питер, 2006. – 718 с.
3) Крейгон Х. Архитектура компьютеров и ее реализация / Х.Крейгон. — М.: Мир, 2004. — 416 с.
4) Макуха В.К.. Микропроцессорные системы и персональные компьютеры: учебник для вузов/В.к. Макуха, В.А. Микерин. – М.: Изд-во Юрайт, 2019. – 156 с. https://biblio-online.ru/book/mikroprocessornye-sistemy-i-personalnye-kompyutery-438081
5) Новожилов О. П. Архитектура ЭВМ и систем в 2 Ч. Часть 1. Учебное пособие для академического бакалавриата / О. П. Новожилов. – М. : Юрайт, 2019. – 276 с.https://biblio-online.ru/book/arhitektura-evm-i-sistem-v-2-ch-chast-1-442223
6) Новожилов О. П. Архитектура ЭВМ и систем в 2 Ч. Часть 2. Учебное пособие для академического бакалавриата / О. П. Новожилов. – М. : Юрайт, 2019. – 246 с.
https://biblio-online.ru/book/arhitektura-evm-i-sistem-v-2-ch-chast-2-444138
7) Преобразователи АЦП [Электронный ресурс].— Режим доступа: http://studopedia.ru/ 5_135496_preobrazovateli.html
8) Таненбаум Э. Архитектура компьютера. 4-е изд / Э. Таненбаум. — СПб.: Питер, 2006. — 699 с.

Дополнительная

9) Гук М. Аппаратные средства IBM PC. Энциклопедия / М.Гук. — СПб: Питер, 2005.– 816 с.
10) Кучеров Д.П. Источники питания ПК и периферии / Д.П. Кучеров. – СПб.: Наука и техника, 2002. – 352 с
11) Лошаков С. Периферийные устройства вычислительной техники [Электронный ресурс]/ Лошаков С.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2013.— 272 с.— Режим доступа: http://www.iprbookshop.ru/16721
12) Миловзоров, О. В. Электроника : учебник для прикладного бакалавриата / О. В. Миловзоров, И. Г. Панков. — 6-е изд., перераб. и доп. — Москва : Юрайт, 2019. — 344 с. — (Бакалавр. Прикладной курс).. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/431928. — ISBN 978-5-534-00077-1

13) Павлов В. А. Периферийные устройства ЭВМ. Учебное пособие: Часть 1 / В.А. Павлов. — Саров: Изд-во СарФТИ , 2001. — 231 с.
14) Федотова Д.Э. Архитектура ЭВМ и систем [Электронный ресурс]: лабораторная работа. Учебное пособие/ Федотова Д.Э.— Электрон. текстовые данные.— М.: Российский новый университет, 2009.— 124 c.— Режим доступа: http://www.iprbookshop.ru/21263
15) Хамахер К. Организация ЭВМ / К. Хамахер, Э. Вранешич, С. Заки. — СПб.: Издательская группа BHV, 2003. – 848 с.