

Федеральное агентство по образованию

---

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Приоритетный национальный проект «Образование»

---

Инновационная образовательная программа  
Санкт-Петербургского государственного  
политехнического университета

**Бунтов В.Д., Макаров С.Б.**

**Микропроцессорные системы**  
**Часть II. Микропроцессоры**

Учебное пособие

Санкт-Петербург  
Издательство политехнического университета  
2008

Федеральное агентство по образованию

---

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Приоритетный национальный проект «Образование»

---

Инновационная образовательная программа  
Санкт-Петербургского государственного  
политехнического университета

**Бунтов В.Д., Макаров С.Б.**

**Микропроцессорные системы**  
**Часть II. Микропроцессоры**

Учебное пособие

Санкт-Петербург  
Издательство политехнического университета  
2008

УДК 621.3..049; 621.395.5; 681.327(075.8)

ББК 32.844-02: 32.973.26-02я73

**Бунтов В.Д., Макаров С.Б. Микропроцессорные системы  
Часть II. Микропроцессоры. Учебное пособие. – СПб.: Изд-во поли-  
технического университета, 2008. –199 с.**

В предлагаемом учебном пособии рассмотрены вопросы, связанные с построением структурных и электрических схем различных цифровых устройств современных радиотехнических систем. Книга посвящена применению микропроцессоров в функциональных узлах радиотехнических устройств. Описаны различные методы проектирования, в том числе с использованием микропроцессоров. Излагаемый материал иллюстрируется электрическими схемами различных функциональных узлов радиотехнических устройств.

Учебное пособие составлено преподавателями кафедры “Радиоэлектронные средства защиты информации” Санкт-Петербургского государственного политехнического университета и предназначено для студентов, обучающихся по направлениям подготовки 230100 – Информатика и вычислительная техника, 210300 – Радиотехника, 210400 – Телекоммуникации, а также соответствующим инженерным специальностям.

**Работа выполнена в рамках реализации Инновационной образовательной программы Санкт-Петербургского государственного политехнического университета «Развитие политехнической системы подготовки кадров в инновационной среде науки и высокотехнологичных производств Северо-Западного региона России».**

Печатается по решению редакционно-издательского совета Санкт-Петербургского государственного политехнического университета.

© Бунтов В.Д., Макаров С.Б., 2008

© Санкт-Петербургский государственный  
политехнический университет, 2008

## Содержание

Глава 1. Основные принципы построения микропроцессоров	5
1.1. Общие представления о микропроцессоре	5
1.2. Архитектура микропроцессоров	12
1.2.1. Структура микропроцессора с фиксированной разрядностью и аппаратной реализацией устройства управления	14
1.2.2. Структура микропроцессора с наращиваемой разрядностью и микропрограммным управлением	19
1.3. Интерфейс микропроцессорных систем	22
1.3.1. Магистралы	24
1.3.2. Порты и адаптеры	29
1.3.3. Прерывания и дисциплина обслуживания прерываний	33
1.3.4. Режим прямого доступа к памяти	38
1.3.5. Организация интерфейса с клавиатурой	40
1.4. Система команд и языки программирования микропроцессоров	42
1.4.1. Система команд	42
1.4.2. Языки программирования микропроцессоров	58
Глава 2. Структура и характеристики универсальных микропроцессоров	63
2.1. Шестнадцатиразрядные микропроцессоры	63
2.1.1. Микропроцессор K1810BM86	63
2.1.2. Математический сопроцессор K1810BM81	71
2.1.3. Сопроцессор ввода-вывода K1810BM89	76
2.1.4. Микропроцессор 80286	78
2.2. Тридцатидвухразрядные микропроцессоры	93
2.2.1. Микропроцессор 80386	93
2.2.2. Микропроцессор 80486	103
2.2.3. Микропроцессор Pentium	109
2.2.4. Микропроцессор Pentium II	117
2.2.5. Микропроцессор Pentium 4	118
2.3. Основные тенденции развития универсальных микропроцессоров.	126
Глава 3. Структура и характеристики специализированных микропроцессоров	133
3.1. Микроконтроллеры	133

3.1.1. Восьмиразрядные микроконтроллеры	133
3.1.2. Шестнадцати и тридцатидвухразрядные микроконтроллеры	149
3.2. Цифровые сигнальные процессоры	165
3.2.1. Цифровые сигнальные процессоры TMS320Cxx, ..., TMS320C6000	166
3.2.2. Цифровые сигнальные процессоры ADSP21xx, ADSP210xx, ADSP-TS2035	190
3.2.3. Цифровые сигнальные процессоры DSP56xxx, DSP9600x, MSC8101, MSC8102	200
3.2.4. Комбинированные высокопроизводительные микропроцессоры	209
Литература	213

# Глава 1. Основные принципы построения микропроцессоров

## 1.1 Общие представления о микропроцессоре

Появление в начале 70-х годов XX века больших интегральных схем (БИС) поставило новые задачи перед разработчиками цифровых устройств. С одной стороны, экономика производства требовала большого объема выпуска БИС при ограниченной их номенклатуре (числе типов), ибо только при этом их производство могло быть рентабельным. С повышением степени интеграции разработка микросхем усложнялась и, чтобы избежать ошибок, требовала автоматизированного проектирования. Процесс изготовления микросхем стал более длительным, занимающим иногда несколько суток, т.к. требовалась эпитаксия, легирование и прочее для многих слоёв полупроводника. Для этого необходимо было применение сложного и дорогостоящего оборудования (лазеров, электронной и рентгеновской литографии, ионной имплантации и т.д.). С другой стороны, каждый тип БИС, представлял собой «жесткую» неизменяемую структуру, состоящую из сотен и тысяч логических элементов. Такая БИС могла выполнять только одну определённую функцию или весьма ограниченный их набор, т.е. БИС проектировались узкоспециализированными. Поэтому число типов БИС, требуемых для реализации разнообразных цифровых устройств возрастало, а требуемый объём выпуска каждого из типов БИС оказывался небольшим.

Решение этих задач было достигнуто путем разработки БИС, функции которых задаются путём подачи на интегральную схему внешних электрических сигналов, изменяющихся по определённой программе. Такие программируемые (настраиваемые) БИС являются основой построения микропроцессоров. Качественным отличием микропроцессоров (МП) от других типов микросхем является возможность их функциональной перестройки с помощью изменения внешней программы.

Микропроцессор является программно-управляемым устройством, осуществляющим процесс обработки цифровых данных и управление им, реализованным на одной или нескольких БИС [13].

Сущность применения микропроцессоров заключается в том, что они заменяют цифровые ИС малой и средней степени интеграции и придают устройствам, в которых они используются, свойства «интеллектуальности».

Устройства и системы, построенные на основе микропроцессоров,

имеют два основных преимущества перед устройствами, реализованными аппаратным способом:

- обладают более высокой функциональной гибкостью, т.к. их перестройка для решения новой задачи требует только смены программы без изменений схемы (аппаратной части);
- требуют меньшего количества элементов, чем эквивалентные устройства с обычными логическими схемами.

Указанные преимущества проиллюстрируем следующим примером.

**Пример 1.** Рассмотрим реализацию алгоритма вычисления уравнения вида:

$$Y = (A \cdot X + B) \cdot X + C. \quad (1.1)$$

Это уравнение может быть вычислено с помощью следующих шагов:

$$A \cdot X = M;$$

$$M + B = N;$$

$$M \cdot X = K;$$

$$K + C = Y,$$

где  $A, B, C, X$  – исходные переменные,  $M, N, K, Y$  – переменные, присвоенные результатам выполнения соответствующих операций.

Структурная схема устройства, реализующая записанный выше алгоритм аппаратным способом, очевидно, состоит из двух перемножителей и двух сумматоров.

Программный способ заключается в описании алгоритма в форме, воспринимаемой вычислительным средством. Программа состоит из отдельных команд. Каждая команда предписывает определённое действие микропроцессору и указывает над какими переменными это действие производится. При реализации алгоритма (1.1) программным способом необходимы соответствующие операционные блоки – перемножитель, сумматор, а также дополнительное оборудование – запоминающие устройства для хранения чисел.

Структурная схема, реализующая алгоритм по программному способу, приведена на рис.1.1. На пересечениях вертикальных и горизонтальных шин находятся управляемые ключевые элементы, например, транзисторы, которые соединяют шины в точке пересечения. Каждый ключевой элемент имеет свой номер. Программа реализации алгоритма представляет собой совокупность команд, выполняемых последовательно во времени:

1-я команда – выбрать из запоминающего устройства (ЗУ) данных с указанными номерами числа  $A$  и  $X$ , перемножить эти числа, результат  $M$

занести в элемент памяти запоминающего устройства с указанным номером;

2-я команда – выбрать из элементов памяти ЗУ данных с указанными номерами числа  $M$  и  $B$ , сложить их и результат  $N$  занести в ЗУ данных;

3-я команда – выбрать числа  $N$  и  $X$ , перемножить их, результат  $K$  занести в элементы памяти ЗУ данных с указанными номерами;

4-я команда – выбрать числа  $K$  и  $C$ , сложить их, результат  $Y$  занести в ЗУ данных;

5-я команда – вывести результат  $Y$ .

Каждая команда записывается кодом, состоящим из кода операции (КОП), адресов  $A_1$ ,  $A_2$  чисел, над которыми выполняется операция, адреса  $A_3$ , куда надо занести результат.

Код операции определяется условным номером, которым обозначена та или иная операция, например: сложение – 01, умножение – 02, вывод результата – 03 и т.д. Номер элемента памяти ЗУ данных называется адресом числа, которое записано в него или будет записано. Если принять следующее распределение чисел по ЗУ данных:  $A - 1$ ,  $B - 2$ ,  $C - 3$ ,  $X - 4$ ,  $M - 5$ ,  $N - 6$ ,  $K - 1$  и  $Y - 8$ , то программа реализации алгоритма может быть представлена таблицей 1.1.



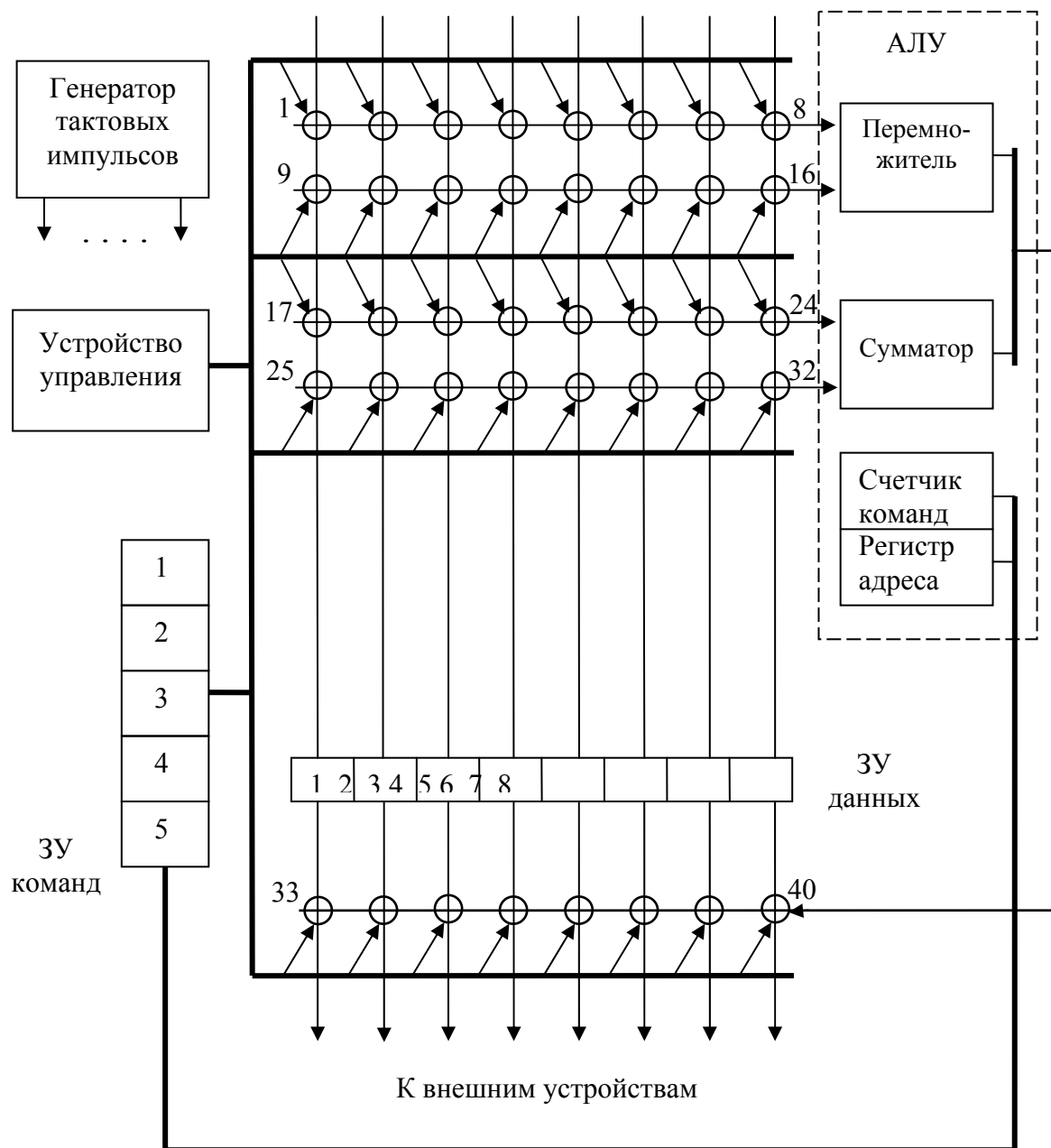


Рис. 1.1 Схема, иллюстрирующая программный способ вычисления уравнения (1.1)

Таблица 1.1

Номер команды	Код команды				Номер ключевого элемента
	КОП	A1	A2	A3	
1	02	1	4	5	1, 12, 31
2	01	5	2	6	21, 26, 38
3	02	6	4	1	6, 12, 39
4	01	1	3	8	23, 21, 40
5	03	8	0	0	40

При реализации программного способа реализации алгоритма необходимо иметь совокупность операционных блоков, называемую арифметико-логическим устройством (АЛУ), запоминающее устройство для хранения исходных чисел и результатов вычислений. Для управления процессом выполнения вычислений в состав устройства, наряду с АЛУ и запоминающим устройством, должно входить устройство управления (УУ). Для хранения команд используется ЗУ команд. Согласованность работы указанных функциональных узлов обеспечивается генератором тактовых импульсов (ГТИ).

В устройстве, схема которого представлена на рис.1.1, в ЗУ команд предварительно заносится программа вычислений, а в ЗУ данных – исходные данные, т.е. числа, над которыми должны быть выполнены определённые программой операции. Из ЗУ команд по сигналу устройства управления выбирается первая команда, дешифруется в УУ и превращается в систему сигналов, управляющих состоянием ключевых элементов (см. табл.1.1). Арифметико-логическое устройство выполняет над выбранными из ЗУ данных числами операцию, предписываемую сигналами устройства управления. Результат на выходе АЛУ записывается в элементы памяти ЗУ данных по адресу, указанному в команде. По окончании выполнения команды из ЗУ команд выбирается следующая команда по новому адресу, который формируется счётчиком команд путем прибавления единицы к предыдущему адресу.

Устройство управления учитывает результат выполненных вычислений по каждой команде по признакам результата: нулевой, единичный, переполнение и т.д. Признаки, представляемые нулем или единицей, передаются в устройство управления.

Программный способ реализации алгоритмов имеет по сравнению с аппаратным два основных преимущества: во-первых, с усложнением алгоритма объём оборудования увеличивается незначительно, главным образом за счёт объёма запоминающих устройств; во-вторых, путём изменения программы можно с помощью одного устройства решать различные задачи.

В самом общем виде структуру ЭВМ можно представить состоящей из 4-х блоков (рис.1.2). Совокупность арифметико-логического устройства и устройства управления, осуществляющих процесс вычислений и управления им, реализованная средствами интегральной технологии в одной или нескольких БИС, называется микропроцессором [8]. Программный способ

реализации алгоритмов устройств и систем имеет следующие особенности:

- одноптипные операции выполняются арифметико-логическим устройством в разное время;
- распределение переменных по входам и выходам блоков изменяется в процессе реализации алгоритма;
- порядок выполнения операций определяется программой.

Для составления программы вычислений используется микропрограммирование, где различают:

- микрооперацию – элементарное действие, для выполнения которого нужно подать один импульс в микропроцессор;
- микрокоманду – набор микроопераций, который одновременно выполняется в микропроцессоре;
- микропрограмму – список последовательно во времени реализуемых микрокоманд, представляющий собой алгоритм работы микропроцессора;
- команду – одну или несколько последовательно выполняемых микропрограмм.

В процессе выполнения программы происходит обмен информацией между микропроцессором, запоминающими устройствами данных и команд и устройствами ввода-вывода (УВВ). Эффективность решения задачи в значительной степени определяется организацией этого обмена и структурой связи между микропроцессором, запоминающими устройствами и устройствами ввода-вывода.

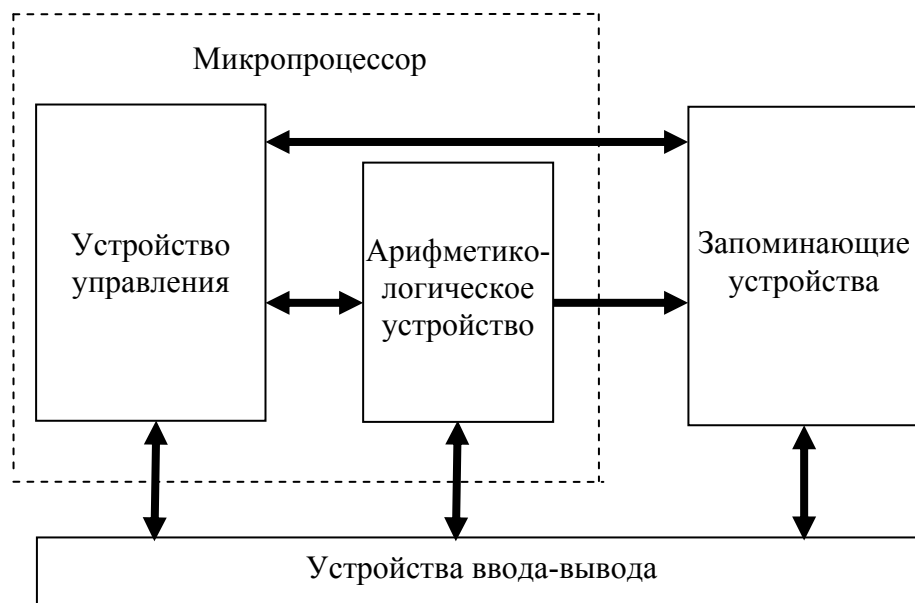


Рис. 1.2. Обобщенная структура ЭВМ

Организацию обмена между микропроцессором, запоминающими устройствами и УВВ осуществляет интерфейс. В функции интерфейса входят: дешифрация адреса устройств; синхронизация обмена информацией; согласование форматов слов; дешифрация кода команды, связанной с обращением к запоминающему устройству или УВВ; электрическое согласование сигналов, передаваемых по шинам и некоторые другие операции.

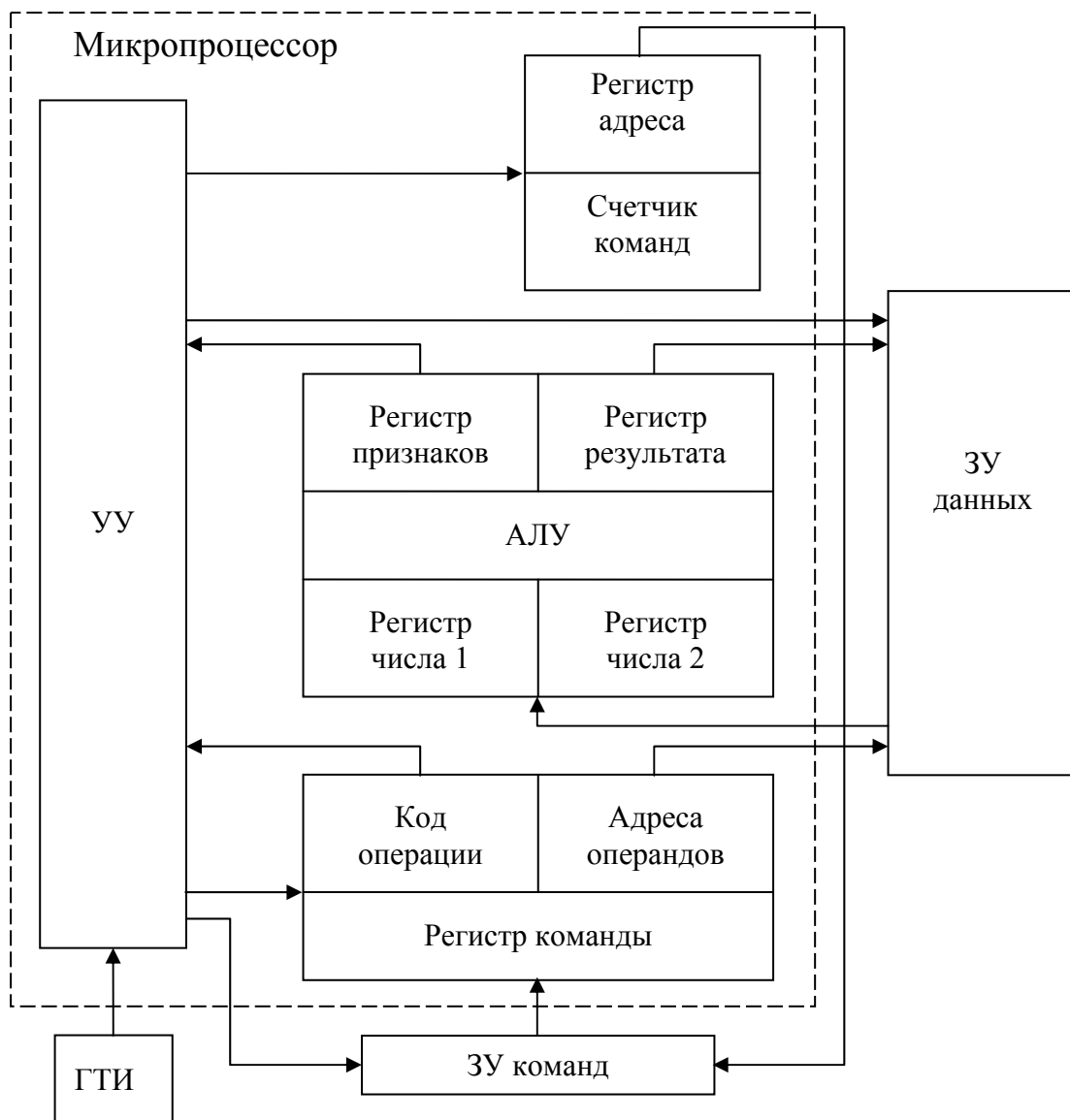


Рис. 1.3 Обобщенная структура микропроцессора

В отличие от универсальных ЭВМ, т.е. предназначенных для решения весьма широкого класса задач, существуют микроЭВМ, которые разрабатываются как специализированные вычислительные устройства для конкретных условий работы. Архитектура, подбор компонентов и программ-

ное обеспечение микроЭВМ создаётся для решения небольшого числа конкретных задач на данном микропроцессоре наиболее целесообразным способом.

Микропроцессор в общем случае (рис.1.3) состоит из трех основных функциональных узлов: арифметико-логического устройства, устройства управления и блока регистров [8]. Блок регистров включает в себя регистры общего назначения (РОН), регистры адреса и команды, регистры для кратковременного хранения чисел, участвующих в операции, а также для хранения результатов вычислений. Числа и команды представляются двоичным кодом, поэтому все показанные на рис.1.3 соединения являются многоуровневыми шинами. В общем случае информация по шинам подаётся в виде слов. Отдельные слова могут передаваться по одной линии последовательно во времени. В микропроцессоре шины являются средствами для передачи адресов, данных и управляющих сигналов.

Микропроцессор в общем случае (рис.1.3) состоит из трех основных функциональных узлов: арифметико-логического устройства, устройства управления и блока регистров [8]. Блок регистров включает в себя регистры общего назначения (РОН), регистры адреса и команды, регистры для кратковременного хранения чисел, участвующих в операции, а также для хранения результатов вычислений. Числа и команды представляются двоичным кодом, поэтому все показанные на рис.1.3 соединения являются многоуровневыми шинами. В общем случае информация по шинам подаётся в виде слов. Отдельные слова могут передаваться по одной линии последовательно во времени. В микропроцессоре шины являются средствами для передачи адресов, данных и управляющих сигналов.

Появление недорогих и надёжных микропроцессоров открыло возможность широкого применения их в сферах, где ранее средства вычислительной техники не применялись. Микропроцессоры оказывают сейчас влияние не только на традиционные области вычислительной техники, но и на такие важнейшие отрасли промышленности, как приборостроительная, радиотехническая, электротехническая, электронная, где создаются средства связи, системы управления, бытовая радиоэлектронная аппаратура [8].

## **1.2. Архитектура микропроцессоров**

Для описания свойств и характеристик микропроцессоров используется термин «архитектура», который является более широким, чем понятие «структура», отражающее состав, внутреннюю организацию и функции

только аппаратных средств вычислительного устройства. Архитектура включает в себя следующие компоненты: структуру, внутренний язык и программное обеспечение.

Внутренний язык представляет собой совокупность команд, реализуемых микропроцессором, структуру и форматы команд; структуру и форматы данных.

Программное обеспечение – набор программ для написания, проверки и отладки прикладных программ.

Таким образом, архитектура характеризуется совокупностью структурных и функциональных признаков и определяется логической структурой отдельных узлов микропроцессора, их взаимосвязью, системой команд и взаимодействием между аппаратурой и программой реализации алгоритмов работы [8].

С точки зрения архитектуры микропроцессоры делятся на два класса:

- с фиксированной разрядностью и аппаратной (схемной) реализацией устройства управления (УУ);
- с наращиваемой разрядностью (секционированные микропроцессоры) и микропрограммным управлением.

При построении микропроцессорных систем (МПС) используют общие принципы построения вычислительных устройств, но с учётом возможностей технологии изготовления БИС. При этом приходится учитывать ограничения на максимально достижимые размеры кристаллов, число выводов корпуса ИС и рассеиваемую на кристалле мощность.

Основными принципами построения МПС являются следующие.

- Микропроцессор реализует алгоритм работы, описание которого хранится во внешнем устройстве памяти.
- Каждая операция в микропроцессоре выполняется соответствующей микрокомандой в соответствии с принципом микропрограммного управления, обеспечивающим универсальность микропроцессора. Микрокомандный уровень управления АЛУ наиболее полно учитывает его структуру, в силу чего реализация операций с помощью микропрограмм является наиболее рациональной с точки зрения экономии объема памяти запоминающих устройств и повышения быстродействия. С другой стороны, поскольку микрокоманда (МК) представляет собой достаточно мелкую детализацию выполняемой операции, например, «обнулить регистр», «переслать содержимое регистра» и т.д., то для программирования сложных ал-

горитмов работы потребовалось бы составлять микропрограммы, содержащие сотни и тысячи микрокоманд. Для упрощения программирования вводится командный уровень управления, так что каждая команда реализуется как последовательность микрокоманд.

- Используется магистральный принцип, заключающийся в том, что все данные, передаваемые от одного блока к другому, проходят через внутреннюю магистраль. При обмене данными к магистрали подключаются только источник и приёмник информации.

- Устранение ошибок при выполнении операции, обусловленных переходными процессами в электрических цепях, обеспечивается внутренней и внешней синхронизацией микропроцессора.

- Разделение электрической схемы микропроцессорной системы на функционально завершённые модули. Модульный принцип построения МПС даёт возможность при проектировании выбирать только необходимые модули и наращивать функциональные возможности микропроцессорной системы. Выбор типа и числа модулей, учитывающих особенности реализуемого алгоритма работы, позволяет повысить эффективность применения микропроцессорной системы для решения конкретных задач.

### **1.2.1 Структура микропроцессора с фиксированной разрядностью и аппаратной реализацией устройства управления**

Микропроцессор с фиксированной разрядностью может быть реализован на одном кристалле, поэтому микропроцессоры этого типа часто называют однокристалльными. На рис.1.4 приведена упрощённая структурная схема такого микропроцессора, содержащего арифметико-логическое устройство, блок внутренних регистров, устройство управления и интерфейс.

Примером такой структуры является 8-разрядный однокристалльный микропроцессор КР580ВМ80, выполненный на основе 4140 элементов *n*-МОП структуры [1]. Микропроцессор содержит шесть 8-разрядных регистров общего назначения (В, С, D, E, H, L), 8-разрядный регистр-аккумулятор два 8-разрядных буферных регистра, два 8-разрядных регистра (W и Z), 8-разрядный регистр признаков и 8-разрядное арифметико-логическое устройство. Первый байт команд поступает через буфер данных в регистр команд и дешифрируется в дешифраторе команд и схеме управления машинным циклом. Микропроцессор имеет схему десятичной коррекции, позволяющую выполнять обработку данных, представленных в двоично-десятичной системе счисления. Указатель стека и регистр счётчика команд имеют 16 разрядов. Их содержимое поступает в 16-разрядную

шину адреса через буфер адреса. Для изменения содержимого регистров общего назначения, регистра счетчика команд и указателя стека имеется схема инкремента-декремента. Выбор нужного регистра общего назначения, регистра счетчика команд и указателя стека или регистров W и Z, осуществляется с помощью схемы выборки регистра (СВР) и мультиплексора.

Арифметико-логическое устройство представляет собой комбинационное устройство и предназначено для выполнения простейших операций: сложения и вычитания чисел, логических операций И, ИЛИ, сложения по модулю 2, сдвига, пересылок и т.д. Сложные операции, предусмотренные в некоторых командах, осуществляются обычно с помощью микропрограммы. Признаки операций, выполняемых АЛУ, а также состояние микропроцессора отражаются регистром признаков, состоящим из нескольких триггеров (флагов). Каждый флаг связан с определенной выполняемой функцией (флаг нуля, знака, переноса, чётности и т.д.).

Внутренние регистры микропроцессора состоят из специальных регистров и регистров общего назначения (РОН). К специальным относятся регистр адреса, регистр команд, накопительный регистр или регистр-аккумулятор, регистр счётчика команд, указатель стека, индексные регистры и др. Состав их может быть разным в различных микропроцессорах. Иногда функции специальных регистров выполняют РОН, либо ячейки ОЗУ. Регистры общего назначения реализуют функции сверхоперативного запоминающего устройства и их число лежит в пределах от 4 до 64. Использование РОН позволяет уменьшить число обращений к оперативному запоминающему устройству в 1,5 – 3 раза и тем самым повысить быстродействие микропроцессора.

В регистре-аккумуляторе фиксируется результат выполнения операции. Иногда результат операции записывается в один из регистров общего назначения. В этом случае микропроцессор называется многоаккумуляторным.

Для однокристалльных микропроцессоров характерно наличие указателя стека. Это объясняется тем, что подавляющее большинство микропроцессоров предназначено для работы в реальном масштабе времени, а специальный блок запоминающих устройств, реализующий стековую дисциплину доступа, позволяет быстро реагировать на запросы прерывания от внешних устройств и осуществлять переход к прерывающей программе и возврат к прерванной программе.



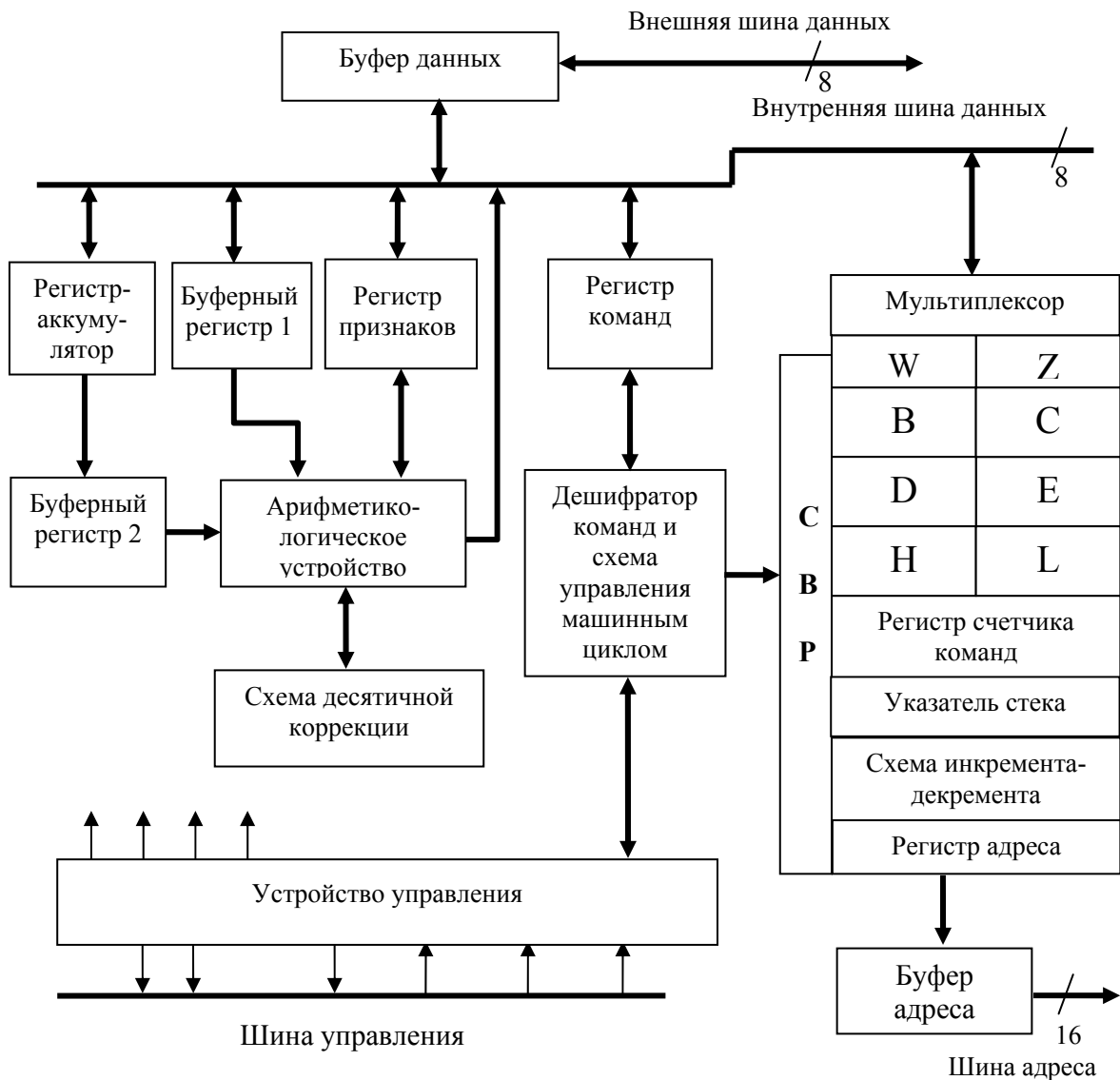


Рис. 1.4. Структура микропроцессора KP580BM80

Использование указателя стека оказывается также эффективным средством работы с микропрограммами, поскольку в этом случае достаточно применения простых аппаратных средств и однобайтовых команд для автоматического запоминания адресов возврата к основной программе. Некоторые микропроцессоры имеют внутренний встроенный стек с ограниченным объемом памяти, при использовании которого операции выполняются с высокой скоростью т.к. отпадает необходимость обращения к внешним запоминающим устройствам. Учитывая, что к стеку при реализации различных алгоритмов работы МП обращаются реже, чем к регистрам общего назначения, в ряде микропроцессоров используется регистр-указатель стека, а сам стек выполнен в виде некоторой зоны во внешнем оперативном запоминающем устройстве. Такая организация работы сопряжена со снижением быстродействия микропроцессора, однако общая

эффективность микропроцессорной системы значительно возрастает за счёт существенного увеличения объема памяти стека.

В простых микропроцессорах операции вычисления каждого следующего адреса программы могут выполняться арифметико-логическим устройством. Для освобождения АЛУ от таких элементарных операций и организации параллельного выполнения операций модификацию адреса может выполнять схема инкремента-декремента, которая увеличивает или уменьшает на единицу значение текущего адреса. Введение такой специальной схемы инкремента-декремента (индексного регистра) существенно расширяет возможности МП за счёт индексной адресации.

Устройство управления [8] строится на основе комбинационных и последовательностных устройств. Для этого широко используются программируемые логические матрицы (ПЛМ). В соответствии с кодом команды устройство управления формирует сигналы, необходимые для работы отдельных блоков МП. Это же устройство формирует сигналы, управляющие процессом считывания очередной команды и организацией переходов. При реализации устройства управления логические связи в ПЛМ закреплены таким образом, что каждому поступившему на вход коду команды соответствует своя система управляющих сигналов. Набор таких команд строго определён и их изменение, наращивание не допускается. При необходимости выполнения операции, для которых нет соответствующих команд в наборе (списке), требуется применение подпрограммы, составленной из группы разрешённых команд. Так как использование подпрограмм снижает скорость вычислений и, следовательно, эффективность применения микропроцессоров, то стремятся, чтобы микропроцессоры имели гибкую и развитую систему команд. При этом на микропрограммном уровне устройство управления координирует свою работу с содержимым регистра признаков (РП). Назначение регистра признаков заключается в идентификации состояния микропроцессора в каждый текущий момент времени выполнения команды. По окончании арифметических и логических операций в регистре признаков фиксируются наличие переполнения, нулевой результат, положительный или отрицательный знак и т.д. Во всех следующих операциях программным способом опрашиваются определённые разряды регистра признаков и производятся условные переходы в программе по значениям соответствующих признаков. Часть регистра признаков микропроцессора принадлежит устройству управления. В этой части регистра признаков фиксируются признаки, определяющие форматы команд и об-

рабатываемых слов, способ адресации, наличие запроса прерывания, разрешение или маскирование прерывания.

Интерфейс микропроцессора разделяется на внутренний и внешний. Внутренний интерфейс обеспечивает сопряжение между блоками микропроцессора, а внешний – между микропроцессором, запоминающими и периферийными устройствами.

Для структуры микропроцессора, представленной на рис.1.4, характерно наличие внутренней шины данных, связывающей все блоки микропроцессора [8]. Разрядность ее равна разрядности слов, с которыми оперирует микропроцессор, и, следовательно, разрядности внешней шины данных по которой передаются не только обрабатываемые слова, но и команды. Разрядность внешней шины данных ограничивает сложность и число команд микропроцессора. Как правило однокристалльные микропроцессоры имеют 4, 8, 16, 32 и 64-разрядные внешние шины данных. Внешняя шина данных используется в режиме двунаправленных передач, т.е. обмен данных между МП и внешними запоминающими устройствами осуществляется путём временного мультиплексирования. Для увеличения гибкости управления в некоторых микропроцессорах применяются отдельные внешние шины данных и адресов.

Большинство 8-разрядных однокристалльных микропроцессоров имеют 16-разрядную шину адреса (ША), которая позволяет непосредственно адресовать внешнее (по отношению к МП) запоминающее устройство. Формирование 16-разрядного адреса в регистре адреса осуществляется во внутренней шине данных полуадресами, так как внутренняя шина данных является 8-разрядной. Для структуры микропроцессора, показанной на рис.1.4, шина адреса и внешняя шина данных являются общими для внешних запоминающих устройств и периферийных устройств. Для прямой адресации внешних запоминающих устройств 8-разрядного кода команды недостаточно. В этой связи в микропроцессорах используются команды переменного формата (одно-, двух- или трёхбайтовые). Первый байт команды поступает только в регистр команд, а второй и третий байты – в регистры W и Z. В первом байте, представляющем код операции, присутствует признак длины команды.

Шину управления однокристалльного МП образуют линии, предназначенные для передачи управляющих сигналов, признаков состояния микропроцессора и периферийных устройств, число которых может быть от 6 до 12. В состав шины управления входят линии, предназначенные для пере-

дачи:

- синхронизирующих сигналов для сопровождения информации в обоих направлениях мультиплексируемой шины данных;
- сигналов, информирующих микропроцессор о состоянии внешних устройств;
- сигнала, указывающего обращение к запоминающему устройству (чтение или запись);
- сигналов запроса и разрешения прерывания и др.

Устройство управления принимает сигналы через дешифратор команд и схему управления машинным циклом, поступившие из запоминающего устройства команд. В соответствии с этими сигналами обеспечивается выполнение программы путем формирования необходимых управляющих сигналов для всех блоков микропроцессора.

Для выполнения одной команды требуется от 1 до 5 машинных циклов, каждый из которых требует от 3 до 5 тактов. В конце машинного цикла выполнения каждой команды анализируется наличие сигнала запроса на входе (ЗПРВ). Если этот сигнал присутствует и прерывания программы разрешены, то микропроцессор переходит в специальный цикл, во время которого содержимое регистра счетчика команд не изменяется и формируется признак начала обработки прерываний (INTA). В микропроцессор посылается код команды (RST) с адресом прерывающей программы.

В составе устройства управления имеется схема анализа прерываний, которая обеспечивает работу микропроцессора в реальном масштабе времени. В её состав входят программно управляемый триггер разрешения прерываний и триггер запроса прерываний. Схема анализа захвата шин, входящая в устройство управления, позволяет организовать режим прямого доступа к запоминающему устройству.

### **1.2.2. Структура микропроцессора с наращиваемой разрядностью и микропрограммным управлением**

Характерной особенностью таких микропроцессоров (рис. 1.5) является модульность структуры и распределение функциональных блоков по отдельным БИС. Структура микропроцессора с микропрограммным управлением (рис.1.5) включает исполнительный блок, содержащий дешифратор микрокоманд, арифметико-логическое устройство с регистром-накопителем (аккумулятором), регистры общего назначения, и блок микропрограммного управления, состоящий из запоминающего устройства микрокоманд, блока формирования адреса, регистра микрокоманд и реги-

стра команд [8]. Регистр микрокоманд предназначен для кратковременного запоминания той микрокоманды, которая подлежит исполнению.

В запоминающем устройстве команд записаны микрокоманды, по которым, как правило, выполняются простейшие операции: сложение, вычитание, сдвиг и т.п. Каждой микрокоманде соответствует одна или несколько операций, выполняемых за один такт. Расположенные в ЗУ команд в определённой последовательности микрокоманды составляют микропрограмму. Такое ЗУ микрокоманд обычно содержит несколько микропрограмм и его функции, как правило, выполняет ПЗУ.

Код микрокоманды имеет одноадресную структуру: [код операции (КОП); адрес числа]. В нём содержится адрес только того числа, которое будет выбрано из запоминающего устройства. Другое число, участвующее в операции, предварительно посылается в аккумулятор. В качестве оперативного запоминающего устройства чисел используются регистры общего назначения. Арифметико-логическое устройство может через свои мультиплексы получать числа из внешнего запоминающего устройства. Инструкция об адресах выбираемых чисел содержится в коде микрокоманды. После выполнения в АЛУ операции, заданной кодом операции, результат помещается в аккумулятор. Затем из запоминающего устройства микрокоманд выбирается следующая по порядку микрокоманда.

Адрес микрокоманды может быть указан в предыдущей микрокоманде, тогда код микрокоманды имеет структуру: [КОП; адрес числа; адрес микрокоманды]. Адрес следующей микрокоманды передаётся из регистра микрокоманд в блок формирования адреса.

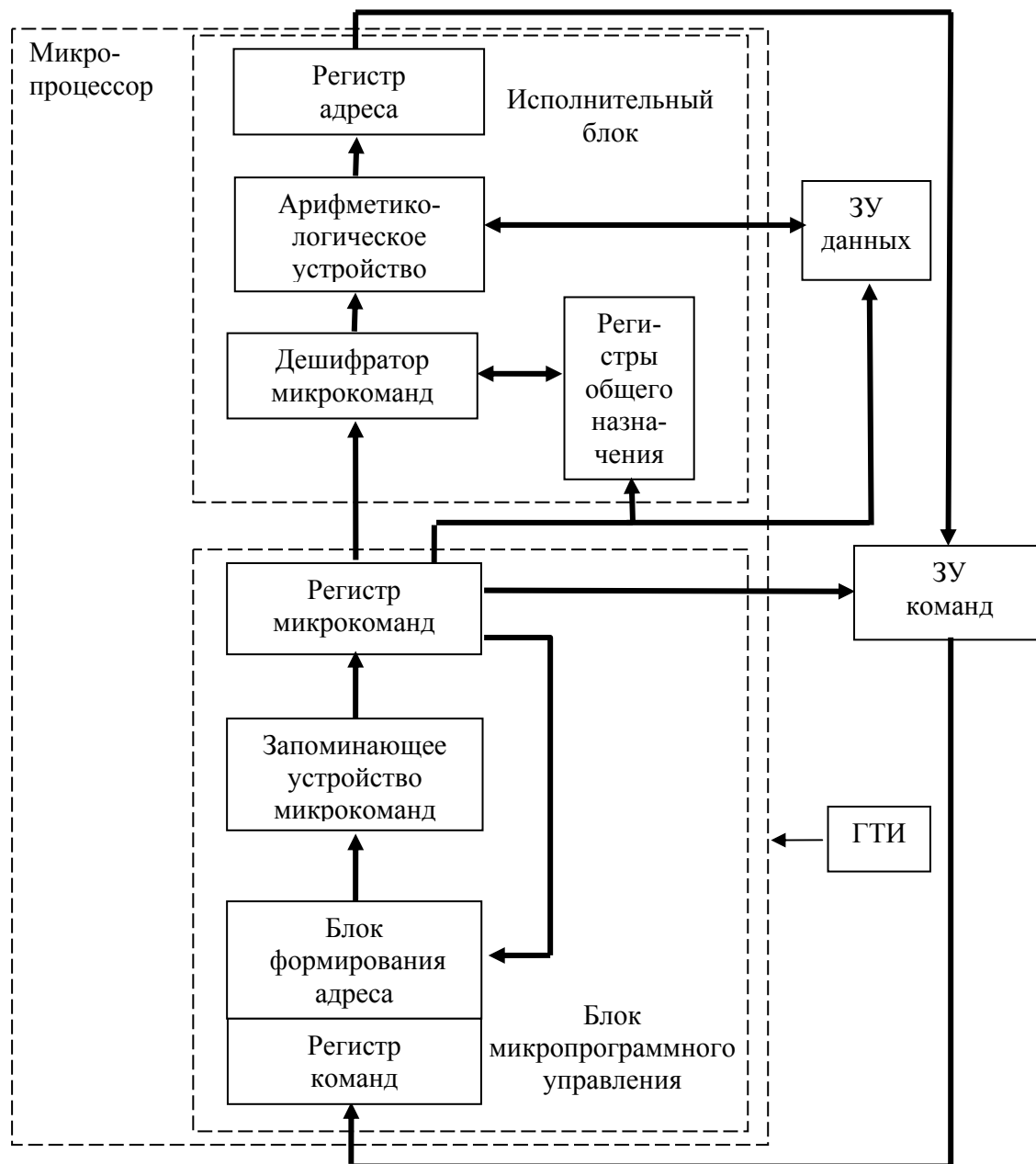


Рис. 1.5. Структура микропроцессора с микропрограммным управлением

Из-за громоздкости программ реализации алгоритма вычислений на уровне микрокоманд и трудностей по их составлению и контролю вводится второй уровень программного управления – командный. При этом используется система команд, каждой из которых соответствует своя микропрограмма. Возможно применение системы команд ЭВМ с развитым математическим обеспечением и приспособление её для данного микропроцессора. Для этого составляется микропрограмма для каждой команды ЭВМ. Такой способ использования микропроцессора называется эмулированием другой ЭВМ.

Микропрограммы различных операций хранятся в ПЗУ. Адрес ячейки ПЗУ, с которой начинается микропрограмма данной команды, соответ-

вует коду команды. Система команд, т.е. программа, заносится во внешнее запоминающее устройство. Процесс вычислений начинается с выборки первой команды из ЗУ команд. Она записывается в регистр команды и присутствует в течение выполнения одной команды программы. При выполнении микрокоманды реализуются следующие стадии, образующие цикл выполнения:

- адрес микрокоманды формируется в блоке формирования адреса и поступает в запоминающее устройство микрокоманд;
- в запоминающем устройстве микрокоманд из элемента памяти с номером, указанным адресом в коде, выбирается микрокоманда и пересылается в регистр микрокоманд;
- в ЗУ данных из ячейки, номер которой указан в адресной части микрокоманды, выбирается число;
- в исполнительном блоке микропроцессора выполняется операция, заданная кодом операции, и формируется результат, который заносится в аккумулятор.

В микропроцессорах с наращиваемой разрядностью чисел исполнительный блок образуется из ряда одинаковых модулей малой разрядности (2-, 4-, 8- и 16-разрядных), объединённых общей шиной микропрограммного управления. В состав каждого входит дешифратор микрокоманд, который преобразует код микрокоманды в совокупность сигналов, управляющих коммутацией узлов и блоков внутри модуля.

Важной особенностью микропроцессоров с модульной структурой является возможность смены БИС или набора БИС, образующих запоминающее устройство микропрограмм. В этой связи одну и ту же системную конфигурацию можно приспособлять для решения различных задач, формируя список команд для каждой из них на микропрограммном уровне наиболее эффективным способом.

### **1.3. Интерфейс микропроцессорных систем**

Выполнение любого алгоритма осуществляется с помощью команд и данных, хранящихся в запоминающих устройствах и микропроцессоре (рис.1.3). В запоминающем устройстве команд хранится совокупность команд (программ), которая извлекается из этого устройства в определенной последовательности, задаваемой микропроцессором. Микропроцессор определяет адреса элементов памяти запоминающего устройства, в котором хранятся данные. Данные передаются из ЗУ данных в микропроцессор и результаты вычислений снова отправляются в ЗУ данных. Для реализации

такой процедуры вычислений используется схема, показанная на рис.1.6. Кроме микропроцессора и запоминающих устройств на рисунке изображены три шины, по которым передаются сигналы. Все три шины в том виде, как они изображены на рис.1.6, должны соединяться с другими устройствами, поскольку данные и коды программ, циркулирующие по шинам между отдельными устройствами, не могут создаваться в этих устройствах. Действительно, перед началом выполнения алгоритма необходимо заполнить элементы памяти запоминающих устройств. В процессе выполнения алгоритма работы необходимо “подпитывать” запоминающие устройства дополнительной информацией, которая может поступать от разнообразных внешних устройств.

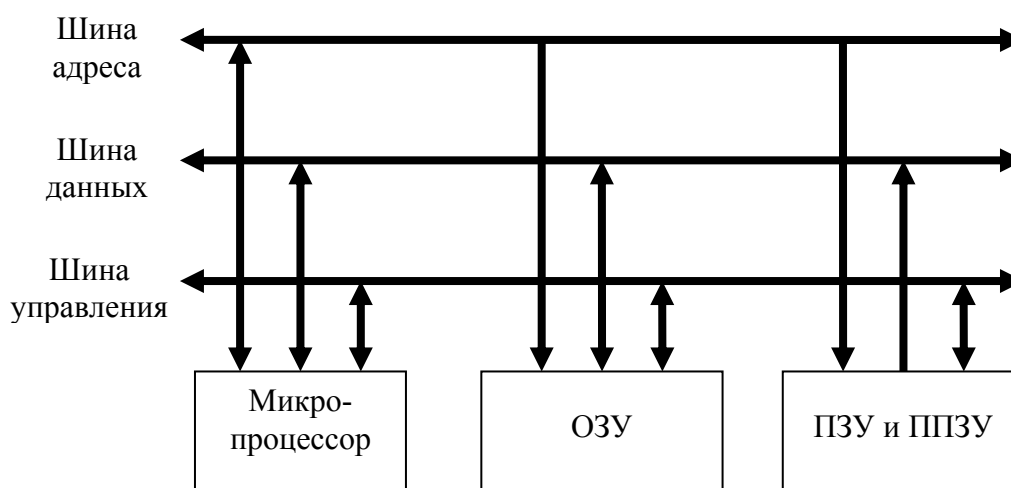


Рис. 1.6. Организация микропроцессорной системы

Микропроцессорная система, реализующая тот или иной алгоритм, может работать лишь постольку, поскольку она связана с внешней средой и получает от нее информацию. Прибегая к литературному образу, можно сказать, что микропроцессорная система, реализующая алгоритм, обращена лицом к внешней среде. На английском языке термин “обращена лицом” или, точнее, “лицом к лицу” звучит как “интерфейс” (interface). Этим термином обозначается весь комплекс принципов, правил и технических средств (иногда отдельных частей этого комплекса), регламентирующих и обеспечивающих обмен информацией между микропроцессором, запоминающими устройствами и внешней средой. Интерфейс представляет собой внешнюю среду, к которой подключаются все три шины (рис.1.6).

Таким образом, интерфейс является связующим звеном микропроцессорной системы и представляет собой совокупность аппаратных и про-



граммных средств, реализующих стандартный вид организации связей в магистрально-модульной системе [23]. Такая система обеспечивает функциональную, электрическую и конструктивную совместимость модулей.

При построении интерфейса микропроцессорных систем необходимо обеспечить:

- подключение модулей запоминающих устройств к шинам МП и обеспечение синхронизации и управления передачей данных по шинам, а также организацию своевременной передачи данных между микропроцессором и частями микропроцессорной системы;
- стыковку частей микропроцессорной системы с внешними устройствами ввода-вывода (УВВ), каналами передачи данных и т.п., включая преобразование внешних сигналов, которые могут быть как цифровыми, так и аналоговыми, в сигналы, совместимые с используемыми для передачи данных по шинам.

При организации обмена информацией могут возникнуть следующие случаи стыковки микропроцессорной системы с внешней средой:

- уровни электрических напряжений на входах и выходах внешнего устройства не соответствуют уровням электрических напряжений, принятым для шины данных;
- число двоичных символов (нулей и единиц) в словах, которыми оперируют внешние устройства, не совпадает с числом двоичных символов, передаваемых одновременно по шинам данных;
- моменты времени, когда внешнее устройство способно или должно осуществить обмен данными по шине данных, не совпадают с моментами времени, когда микропроцессор может или должен выработать управляющий сигнал, разрешающий обмен;
- данные, которыми оперируют внешние устройства, написаны не на том языке (с другим алфавитом и другой грамматикой), с которым оперирует микропроцессор.

Для увеличения степени универсальности микропроцессорной системы необходимо строить интерфейс с учетом перечисленных выше условий и случаев стыковки микропроцессора и внешних устройств.

### **1.3.1. Магистралы**

При организации микропроцессорной системы, состоящей из нескольких БИС (микропроцессоры, ОЗУ, ПЗУ и др.) требуется выполнение следующих основных условий.

Во-первых, различные БИС в микропроцессорной системе должны

соединяться между собой без каких-либо дополнительных устройств. Так, для БИС, построенных на основе транзисторно-транзисторной логики напряжение логического нуля соответствует значению от 0 до 0,5 В; напряжение логической единицы – от 2,8 до 4,5 В. Кроме того, по входным цепям таких БИС должны протекать токи до 0,2 мА. Для ТТЛ совместимых БИС, напряжение на входах и выходах должны иметь именно такие значения, а выходные элементы БИС должны обеспечить ток разветвление  $I_p$  не менее  $I_p = k \cdot 0,2$  мА, где  $k_p$  – коэффициент разветвления по выходу. Этот коэффициент показывает, сколько входов других БИС можно подключить к одному выходу данного БИС.

Во-вторых, мощность электрических сигналов должна быть достаточной для питания входов соединяемых между собой устройств, например, нескольких десятков дополнительных БИС.

Для выполнения этих требований применяют шинные усилители (рис.1.1).

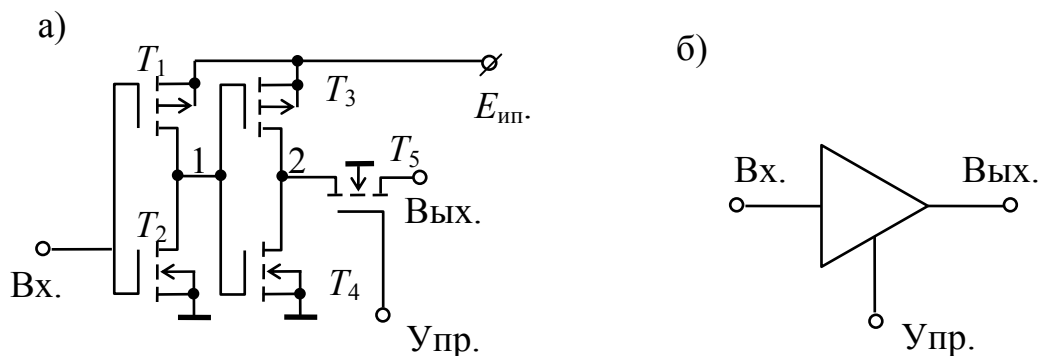


Рис. 1.7. Принципиальная схема шинного усилителя (а) и его условное обозначение (б)

Усилитель, схема которого представлена на рис.1.1,а, может находиться в трех состояниях. Предположим, что на входе "Упр" действует напряжение логической единицы. Тогда напряжение на выходе усилителя зависит от состояния транзисторов  $T_3$  и  $T_4$ . Пусть на входе усилителя присутствует логический нуль. Тогда транзистор  $T_2$  закрыт,  $T_1$  открыт. В точке 1 действует напряжение, практически равное напряжению питания  $E_{инп}$ . В этом случае транзистор  $T_4$  открыт, а транзистор  $T_3$  закрыт. В точке 2, а, следовательно, и на выходе усилителя присутствует напряжение, равное логическому нулю.

Пусть теперь на входе действует напряжение логической единицы. Как нетрудно убедиться на выходе усилителя будет также логическая единица. Шинный усилитель повторяет состояние, которое имеется на входе.

Третье состояние соответствует случаю, когда на входе "Упр" действует напряжение логического нуля. Транзистор  $T_5$  закрыт и выход усилителя отключён от других транзисторов.

Рассмотрим процессы, протекающие в шинном усилителе подробнее. Затворы транзисторов, в частности транзисторов  $T_1$  и  $T_2$ , практически изолированы от подложки, поэтому электрический ток в цепях затворов измеряется долями микроампера. При переключении транзистора из одного состояния в другое происходит заряд емкости затвора, составляющей единицы пикофарад. Следовательно, в динамическом режиме, т.е. при переключении, в цепи затвора протекает ток, в среднем равный 1 мкА. При напряжении  $E_{\text{пит}} = 5\text{В}$  потребление мощности на входе составляет 5 мкВт.

Когда на выходе усилителя имеется напряжение логической единицы, ток, протекающий в выходной цепи, определяется тремя факторами: напряжением питания, сопротивлением нагрузки и рассеиваемой транзистором  $T_4$  мощностью. Шинные усилители обеспечивают ток около 160 мА, при выходной мощности 500 мВт. Коэффициент усиления по мощности такого шинного усилителя равен 100 000.

В ряде случаев организация работы микропроцессорной системы (рис.1.6) предполагает передачу данных, например, из ОЗУ на вход микропроцессоров и обратно. При этом выходные сигналы ОЗУ и МП должны быть усилены так, чтобы их мощности хватило для работы элементов памяти ОЗУ и входных устройств микропроцессора.

В этих случаях используются комбинации шинных усилителей, схема которых представлена на рис.1.1,а, и шинных усилителей с инвертированным управлением (рис.1.8,а).

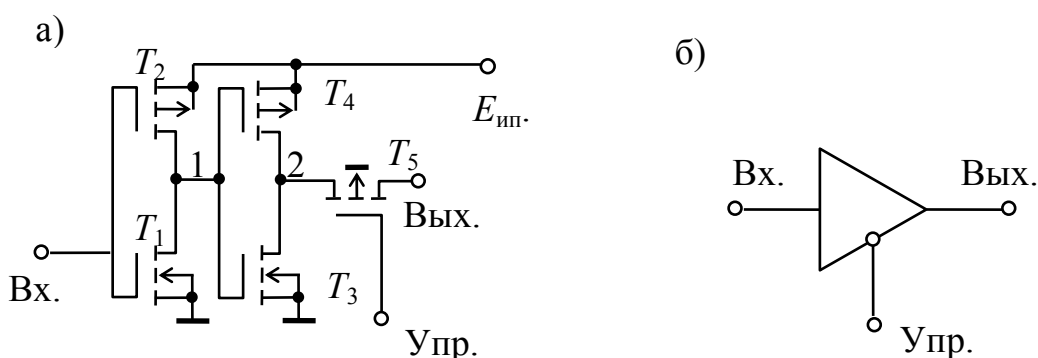


Рис. 1.8. Принципиальная схема шинного усилителя с инвертированным управлением (а) и ее условное обозначение (б)

Схема соединений транзисторов  $T_1$ - $T_4$  повторяет соответствующую часть схемы на рис. 1.1,а. Транзистор  $T_5$ , представляет собой транзистор с

индуцированным каналом  $p$  типа. Такой транзистор закрыт, когда напряжение на его затворе соответствует уровню логической единицы и открыт, когда напряжение на его затворе равно логическому нулю.

Рассмотрим работу комбинации шинных усилителей, принципиальная схема которой приведена на рис.1.9,*a*.

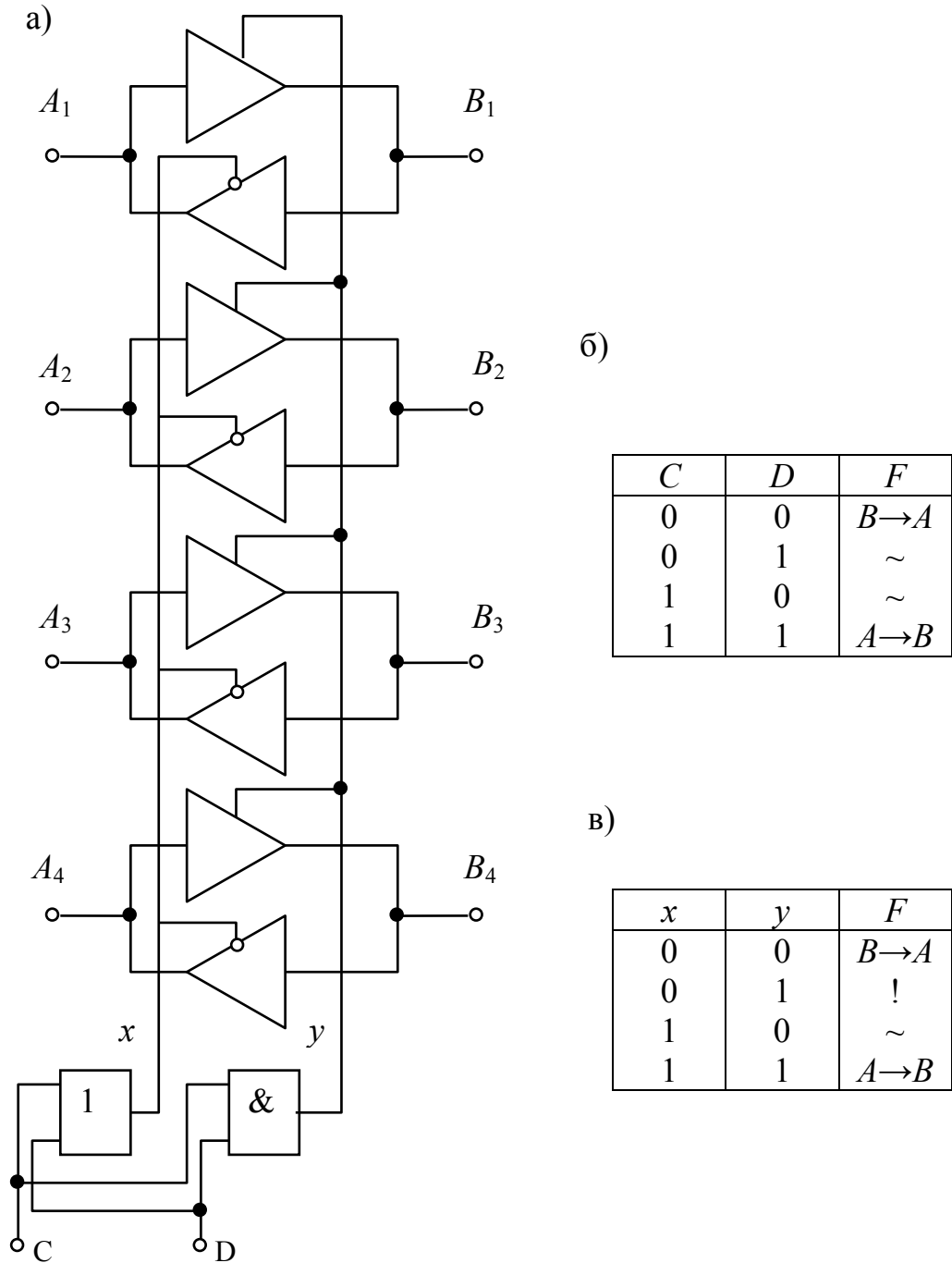


Рис. 1.9 Принципиальная схема комбинации шинных усилителей (а) и таблицы истинности (б), (в)

В комбинации имеется восемь шинных усилителей и два дополни-

тельных логических элемента И и ИЛИ [23]. В таблице истинности рис.1.9,б символами С и D обозначены логические значения сигналов, действующих на входах логических элементов И и ИЛИ, а символом F в этой таблице обозначена функция, выполняемая комбинацией шинных усилителей.

Если на входах С и D действуют сигналы, имеющие значение логического нуля, то комбинация шинных усилителей работает как четыре независимых усилителя, каждый из которых усиливает сигнал, поступающий соответственно от точек  $B_1-B_4$ , и передает его к одной из точек  $A_1-A_4$ . Этот режим работы символически отмечен в таблице истинности (рис.1.9,б) как  $B \rightarrow A$ .

Пусть теперь на входах С и D действуют сигналы, имеющие значение логической единицы. При этом на выходах элемента И и ИЛИ присутствует уровень логической единицы. Комбинация шинных усилителей передает сигнал от точек  $A_1 - A_4$  к точкам  $B_1 - B_4$ , что отмечено в таблице истинности (рис.1.9,б) как  $A \rightarrow B$ .

Наконец, рассмотрим случай, когда сигнал С имеет значение логического нуля, а сигнал D – значение логической единицы. На выходе логического элемента И будет уровень логического нуля, а на выходе элемента ИЛИ уровень логической единицы. Тогда сигнал от точек  $A_1-A_4$  не будет проходить в точки  $B_1-B_4$ . Аналогичная ситуация имеет место, когда сигнал С соответствует значению логической единицы, а сигнал D – значению логического нуля. В таблице на рис.1.9,б этот режим работы отмечен символом " $\sim$ ".

Наличие в схеме рис.1.9,а логических элементов И и ИЛИ защищает комбинацию шинных усилителей от возможного самовозбуждения устройства, которое может возникнуть, если подавать управляющие сигналы непосредственно на входы управления  $x$  и  $y$  усилителей. Действительно, как видно из таблицы истинности, представленной на рис.1.9,в, имеется ситуация, обозначенная символом !, когда на входы  $x$  поступает напряжение логического нуля, а на входы  $y$  – напряжение логической единицы. Все шинные усилители оказываются включенными и возникает самовозбуждение. Заметим, что введение дополнительных логических элементов в шинный усилитель представляет собой, по существу, резервирование необходимое для обеспечения надежности работы усилителя.

В микропроцессорных системах в общем случае может использоваться трехшинная (рис.1.6) и двухшинная конфигурация. При этом сохраняет-

ся полная совместимость параметров шин со входными и выходными параметрами дополнительных устройств.

В ряде микропроцессорных систем используется режим мультиплексирования, заключающийся в том, что по половине адресной шины передается половина адреса, а вторая половина адреса передается по очереди с передачей данных. При мультиплексировании требуется меньшее число проводников, а значит и выводов у корпусов БИС, но при этом усложняется функционирование микропроцессорной системы. У 8-разрядных МПС, как правило, шина данных содержит 8 проводников, шина адреса - 16 проводников и несколько (не больше десяти) проводников имеется у шины управления.

Магистралью микропроцессорной системы называется совокупность технических средств, включающая в себя проводники, шинные усилители и дополнительные устройства.

Магистраль обеспечивает единство уровней электрических сигналов, достаточную мощность этих сигналов, единство назначений каждого из проводников, а именно, если проводник предназначен для передачи, например, второго разряда адреса, то это условие должно выполняться независимо от того, какое дополнительное устройство подсоединяется к магистрали. Если на дополнительное устройство не должны поступать адреса, то оно не соединяется с этой магистралью.

В магистрали обеспечивается выполнение требований стандартизации параметров: напряжений, мощности, формы импульсов и т.п.

Существует три вида магистралей:

- адресная магистраль, содержащая шину адреса;
- магистраль данных, включающая в себя шину данных;
- сигнальная магистраль, содержащая шину управления.

В заключение следует отметить, что шинные усилители не могут быть размещены в БИС самого микропроцессора из-за возникающих при этом трудностей с отводом тепла, обусловленного рассеиваемой мощностью.

### **1.3.2. Порты и адаптеры**

Для подключения к магистралям микропроцессорной системы используются устройства, позволяющие принимать и передавать последовательности символов, состоящих из логических нулей, число которых равно числу проводников в магистрали.

На рис.1.10 показано подключение двух внешних устройств к магистралям, где  $D_1$  и  $D_2$  - дешифраторы,  $ШУ_1$  и  $ШУ_2$  – шинные усилители,  $ВУ_1$  и

$VY_2$  – внешние устройства. У каждого из дешифраторов имеется один выход, высокий уровень напряжения на котором появляется только в том случае, когда на входах имеется определенная комбинация логических нулей и единиц. Эта комбинация, как видно из рис. 1.10, поступает на входы дешифраторов из адресной магистрали. При этом используются не все разряды адресной магистрали, а только их часть, необходимая для выбора одного из двух внешних устройств. Работа такой микропроцессорной системы происходит следующим образом. Пусть, например, надо осуществить обмен данными с внешним устройством  $VY_2$  и этому внешнему устройству присвоен условный код (адрес) 0010. Эта комбинация логических нулей и единиц передается по четырем проводникам адресной магистрали, которые соединены с входами дешифраторов  $D_1$  и  $D_2$ . Дешифратор  $D_2$  формирует соответствующий уровень напряжения на одном из своих выходов. Выбор того или иного выхода дешифратора определяется следующими условиями: будет ли происходить передача информации из магистрали данных во внешнее устройство или наоборот, из внешнего устройства в магистраль данных, и в какой момент времени этот обмен произойдет. Для этого из сигнальной магистрали в соответствующий момент времени поступает комбинация символов, разрешающая передачу информации из магистрали данных во внешнее устройство, либо комбинация символов, разрешающая прием информации из внешнего устройства в магистраль данных. Это комбинация символов дешифрируется и шинный усилитель  $ШУ_2$  подключает внешнее устройство  $VY_2$  к магистрали данных, причем таким образом, чтобы передача данных происходила лишь в одном направлении. Таким образом дешифратор осуществляет дешифрирование кода внешнего устройства и дешифрирование комбинации символов, поступающих по сигнальной магистрали.

Комбинация из шинного усилителя и дешифратора (она обведена штриховой линией на рис. 1.10) называется порт [1, 8, 23].

Заметим, что в схеме микропроцессора также содержались элементы, способные подключать или отключать внутренние магистрали микропроцессора от внешних. Такие же элементы должны входить и в состав запоминающих устройств. В микропроцессоре, ОЗУ, ПЗУ и ППЗУ порты находятся внутри соответствующих устройств.

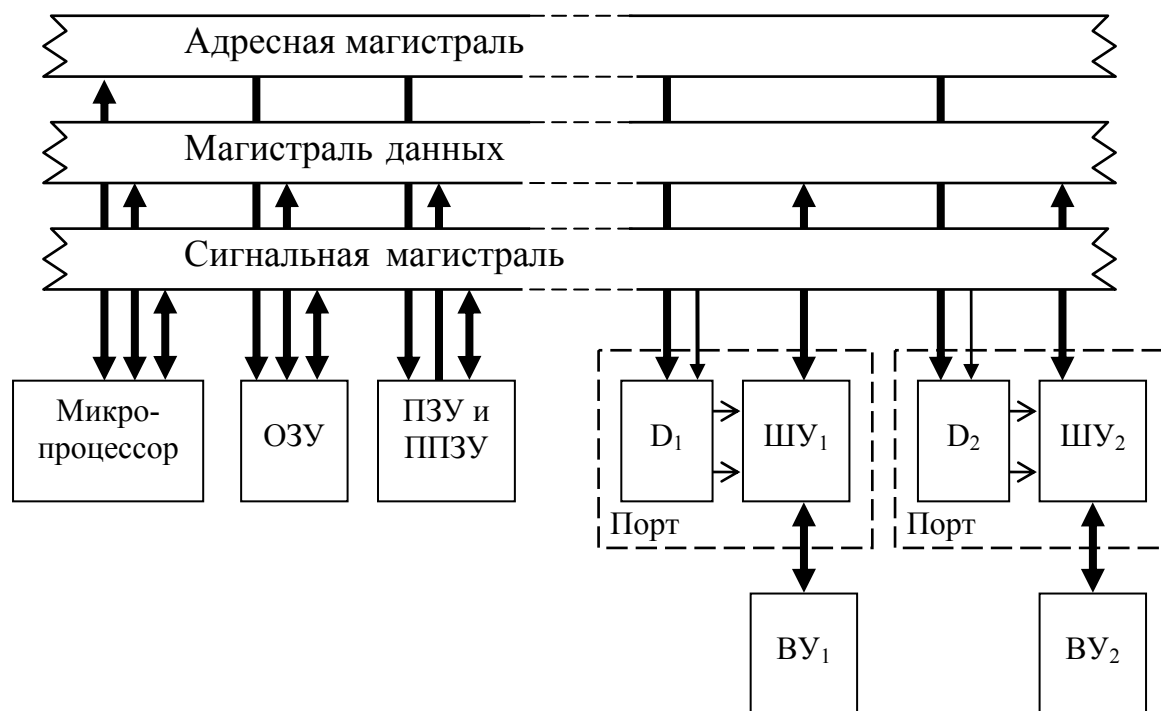


Рис. 1.10. Организация подключения дополнительных (внешних) устройств к магистралям

Рассмотрим случай подключения к магистралям внешних устройств при условии, что эти внешние устройства оперируют словами различной длины.

Для этого используется устройство преобразования формата данных - адаптер [1, 23], условное обозначение которого приведено на рис.1.11,а. Адаптер состоит в общем случае из сдвигового регистра, который осуществляет операции записи, сдвига и считывания данных. Пусть длина слова, состоящего из  $K$  разрядов, на входе адаптера меньше длины  $N$ -разрядного слова на его выходе ( $K < N$ ). В этом случае происходит запись первых  $K$  разрядов входного слова в сдвиговый регистр и сдвиг этих разрядов на  $K$  тактов, в результате чего первые  $K$  ячеек памяти сдвигового регистра освобождаются. Далее процедура сдвига разрядов повторяется и происходит до тех пор пока не заполнятся все  $N$  ячеек памяти сдвигового регистра. После заполнения сдвигового регистра происходит считывание  $N$ -разрядного слова на выход адаптера.

Если длина слова на входе адаптера больше длины слова на его выходе ( $K > N$ ), то сдвиговый регистр должен содержать  $N+K$  ячеек памяти. Входное  $K$ -разрядное слово записывается в первые  $K$  ячеек памяти сдвигового регистра. После  $N$  тактов сдвига, содержимое  $N$  последних ячеек памяти регистра считывается на выход адаптера. Далее осуществляются сле-



дующие  $N$  тактов сдвига и так происходит до тех пор, пока сдвиговой регистр не очистится полностью. После этого записывается следующее  $K$ -разрядное входное в первые  $K$  ячеек памяти сдвигового регистра и процедура считывания слова повторяется.

В общем случае устройство преобразования формата данных (адаптер) может содержать несколько входов и выходов, несколько портов и буферных регистров, сдвиговые регистры и группы вспомогательных устройств (рис.1.11,б, где  $A_i$  – адаптер,  $i = 1, 2, \dots, 2M$ ). Один из входов-выходов этого устройства через соответствующий порт соединяется с магистралью данных, а остальные — с внешними устройствами.

При передаче слова или группы слов из внешнего устройства в микропроцессор слово из внешнего устройства через соответствующий порт записывается в один из буферных регистров (группа слов записывается в группу буферных регистров). Таким образом освобождается внешнее устройство и его работа не связывается с работой микропроцессора. Слово (группа слов) преобразуется сдвиговыми регистрами так, как это было описано выше. После преобразования в сдвиговых регистрах слово переписывается в буферные регистры. По команде от микропроцессора это слово через порты в соответствующий момент времени передается в магистраль данных. Аналогично происходит передача данных из магистрали данных во внешние устройства. Описанное устройство (рис.1.11,б) называется программируемый адаптер ввода-вывода или программируемый периферийный адаптер.

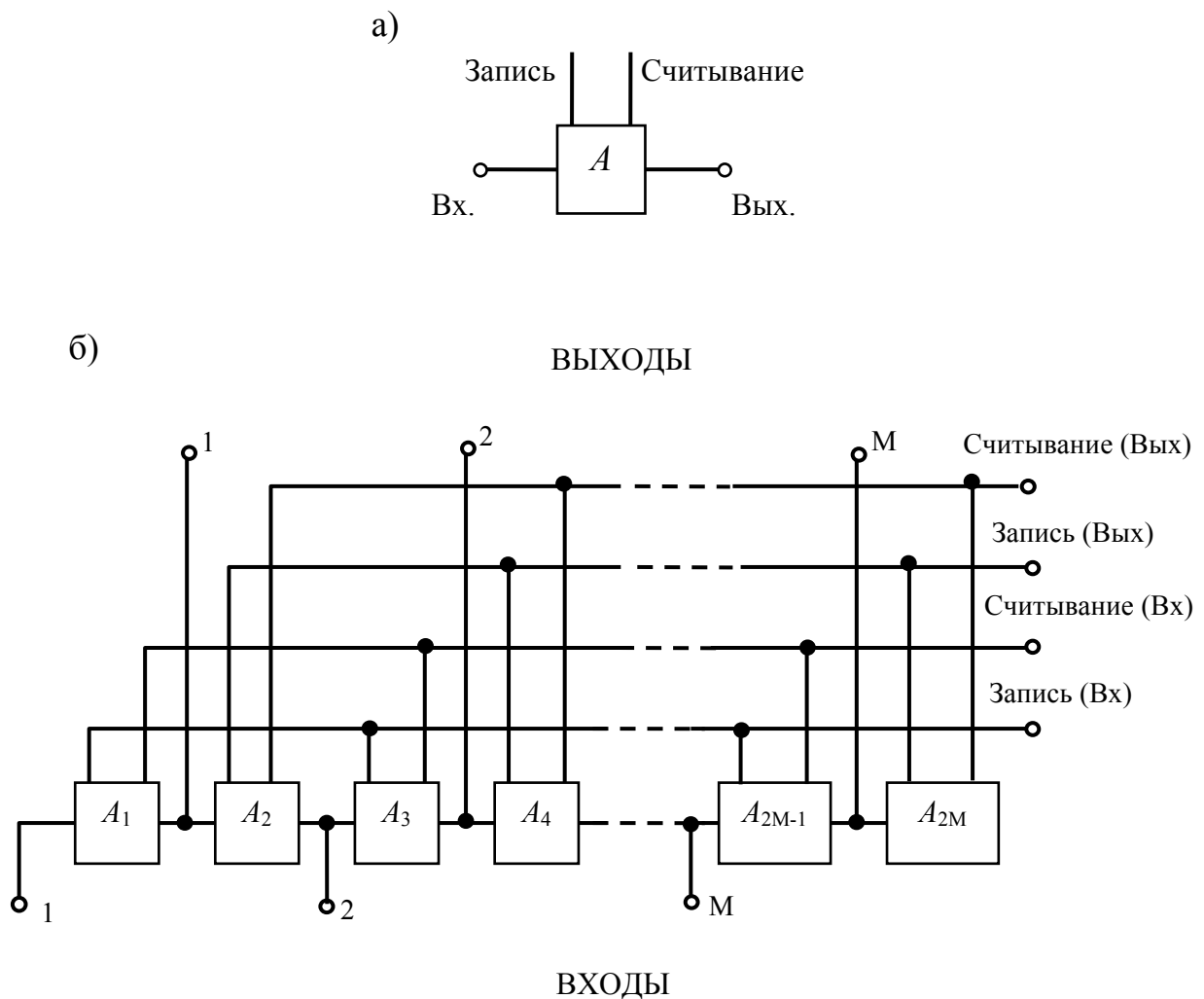


Рис. 1.11 Условное обозначение (а) и обобщенная схема адаптера (б)

Чтобы реализовать рассмотренную выше процедуру преобразования формата данных необходимо на входы “Запись” и “Считывание” каждого адаптера (рис.1.11,а) в соответствующие моменты времени подать необходимые микрокоманды, которые могут храниться в самом адаптере или в запоминающем устройстве. Часто в качестве программируемых периферийных адаптеров используются микропроцессоры.

### 1.3.3 Прерывания и дисциплина обслуживания прерываний

Передача слова из внешнего устройства в запоминающее устройство или в другое внешнее устройство требует использования по меньшей мере двух адресов. Один из них является адресом ячейки памяти запоминающего устройства, куда должно быть помещено (откуда должно быть прочитано) слово, а второй – адресом, позволяющим выбрать нужный порт или адаптер. Кроме того, необходимо иметь несколько управляющих сигналов, обеспечивающих выполнение команд записи (считывания) слова в запоминающее устройство и необходимых процедур в порту или адаптере. В течение операции ввода-вывода независимо от того, происходит передача одного слова (байта) или группы слов (байт), в других внешних устройст-

вах микропроцессорной системы не могут выполняться программы, поскольку занят основной источник команд — микропроцессор. Микропроцессорная система может находиться либо в состоянии, в котором она выполняет свою внутреннюю, системную программу, либо подпрограмму ввода-вывода. Переключение МПС из системной программы в состояние подпрограммы ввода-вывода называется прерыванием и выполняется с помощью специальных управляющих сигналов – сигналов прерывания, или, чаще, запросов на прерывание. По сигналу прерывания прекращается выполнение системной программы и начинается выполнение подпрограммы ввода-вывода.

Рассмотрим этот процесс подробнее. Пусть в некоторый момент времени в МПС возникла необходимость осуществить обмен данными с одним из внешних устройств. Она может возникнуть как в самой микропроцессорной системе, если исчерпана вся хранящаяся в запоминающем устройстве информация или завершена реализация алгоритма, так и во внешнем устройстве. В этих случаях формируется сигнал прерывания, который поступает в микропроцессор, имеющий для этого специальный вход. После поступления сигнала прерывания прекращается выполнение системной программы и принимаются меры к тому, чтобы прерванную программу можно было бы впоследствии продолжить без потери информации. Для этого содержимое регистра счетчика команд, а также всех или части регистров общего назначения (рис.1.11) передается в специальную область памяти оперативных запоминающих устройств - стек. После этого начинается выполнение подпрограммы ввода-вывода. Из памяти ОЗУ извлекаются начальный адрес подпрограммы ввода-вывода и дополнительные данные и помещаются в регистр счетчика команд и регистры общего назначения. Микропроцессорная система переходит в состояние ожидания.

Описанная последовательность действий получила название обслуживания сигнала прерывания. Обслуживание начинается с приема сигнала прерывания и заканчивается переходом микропроцессорной системы в состояние ожидания. В соответствующий момент времени формируется управляющий сигнал, по которому начинается процедура ввода-вывода. После завершения процедуры ввода-вывода формируется сигнал окончания и происходит обратный процесс. В регистр счетчика команд и регистры общего назначения возвращается информация, ранее переданная на хранение в запоминающее устройство и продолжается выполнение прерванной системной подпрограммы.

Перечисленная последовательность действий получила название рестарт. Обслуживание сигнала прерывания и рестарт, который тоже можно отнести к обслуживанию сигнала прерывания, представляет собой последовательность действий, которые совершаются под управлением подпрограммы.

Рассмотрим два случая обслуживания прерывания. В первом случае два и более устройств ввода-вывода одновременно формируют сигналы прерывания. Во втором – выполнение системной программы в микропроцессоре по тем или иным причинам не может быть прервано.

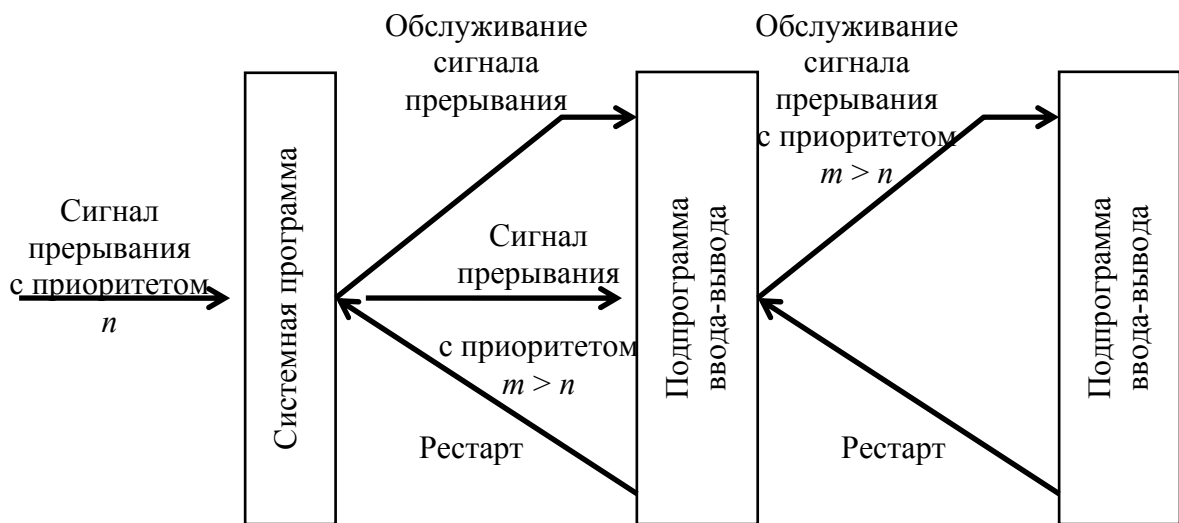


Рис. 1.12 Процесс обслуживания сигналов прерываний с разными приоритетами

В первом случае используется система приоритетов, а во втором применяется способ маскирования. Сущность системы приоритетов состоит в том, что каждому из устройств ввода-вывода или каждому из портов присваивается целое число (1,2,3,...), называемое приоритетом. При одновременном поступлении нескольких сигналов на прерывание обслуживается тот из них, который имеет старший приоритет. Если поступают два сигнала прерывания с одинаковым приоритетом, обслуживается любой из них. Не обслуженные сигналы прерываний, в зависимости от дисциплины обслуживания, либо теряются, т.е. соответствующее внешнее устройство повторяет сигнал прерывания до тех пор, пока оно не будет обслужено, либо запоминаются и ждут, когда будет обслужен сигнал прерывания со старшим приоритетом.

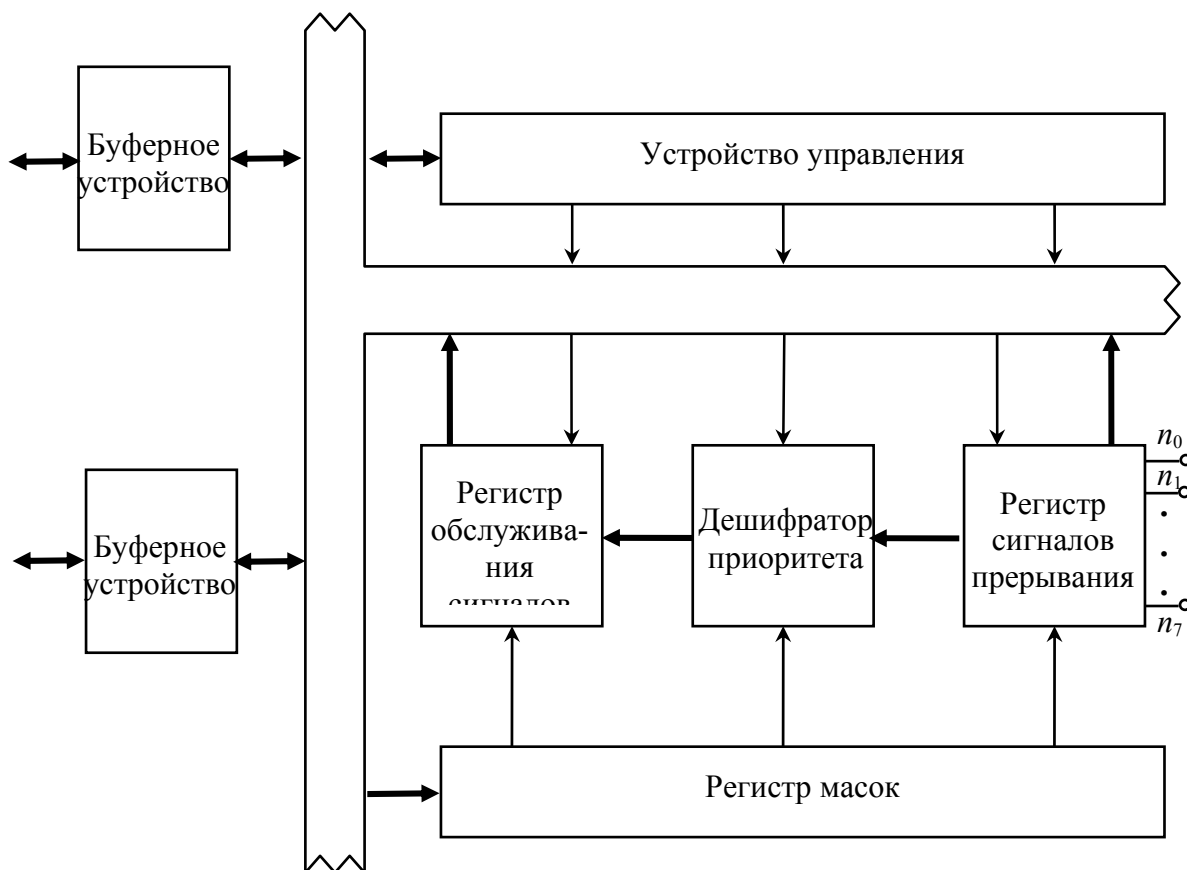


Рис. 1.13 Структурная схема программируемого контроллера прерываний

Возможна ситуация, когда сигнал прерывания с некоторым приоритетом поступает в момент выполнения подпрограммы ввода-вывода, которая в свою очередь была введена в действие по сигналу прерывания с низшим приоритетом. В этом случае выполняются те же действия по обслуживанию сигнала прерывания, как и при выполнении системной программы. В частности, содержимое регистра счетчика команд и регистров общего назначения передается в стек. Процесс обслуживания сигналов прерывания в этом случае схематически показан на рис. 1.12. После выполнения подпрограммы, начатой по сигналу прерывания с более высоким приоритетом, производится возврат к выполнению подпрограммы, начатой по сигналу прерывания с более низким приоритетом. После окончания выполнения этой подпрограммы будет осуществлен возврат к системной программе. Самый высокий приоритет присваивается наименее быстродействующим устройствам ввода-вывода.

В стеке может храниться несколько комплектов данных, относящихся к нескольким подпрограммам. Комплект данных, поступивший в него пер-

вым в ответ на первый сигнал прерывания, извлекается из указателя стека последним, что и требуется по сути выполняемых действий.

Если системная программа не может быть прервана, осуществляется запрет на прерывание. Для этого используется особое состояние микропроцессорной системы, называемое маскированием. Микропроцессорная система в состоянии маскирования не чувствительна к сигналам прерывания. Эти сигналы теряются, либо запоминаются и будут обслужены, когда микропроцессорная система выйдет из состояния маскирования. Замаскировать можно все сигналы прерывания или только сигналы, имеющие приоритет не выше заданного. В частности, при выполнении подпрограммы, начатой на основе сигнала прерывания с некоторым приоритетом, все сигналы прерывания, имеющие более низкий приоритет, маскируются автоматически.

Процедура обслуживания сигналов прерывания с различными приоритетами называется дисциплиной обслуживания прерывания. На рис.1.13 показана схема программируемого контроллера прерываний, реализующего дисциплину обслуживания прерывания [1, 11].

Запросы на прерывание поступают по восьми входам, обозначенным на рисунке как  $n_0-n_7$ , и запоминаются в регистре сигналов прерывания. Из этого регистра сигналы поступают в дешифратор приоритета, который связан с регистром масок. Если сигнал прерывания обладает самым высоким среди остальных приоритетом и не является замаскированным, он передается в регистр обслуживания сигналов прерывания. Как видно из рис.1.13, связь программируемого контроллера прерываний с другими устройствами микропроцессорной системы осуществляется через магистраль данных и через сигнальную магистраль, для чего имеются специальные буферные устройства.

Работа программируемого контроллера прерываний представляет собой выполнение подпрограммы с данными, хранящимися в регистре счетчика команд и регистрах общего назначения, в этом смысле все его функции могут быть выполнены микропроцессором. Иногда функции реализации дисциплины обслуживания прерываний возлагаются на программируемые адаптеры ввода-вывода. В состав многих микропроцессорных систем входят отдельные программируемые контроллеры прерываний.

Процедуры ввода-вывода совершаются в соответствии с подпрограммой ввода-вывода, которая формируется микропроцессором. Поэтому на все время выполнения подпрограммы ввода-вывода микропроцессор необ-

ходимо освободить от выполнения каких-либо других программ. При магистральной организации МПС по адресной магистрали и магистрали данных в каждый момент времени может передаваться лишь одно слово. Поэтому независимо от конфигурации микропроцессорной системы необходимо обеспечить дисциплину обслуживания прерываний.

Дисциплина обслуживания прерываний зависит от сложности самой микропроцессорной системы и может быть как простейшей без приоритетов, так и со сложными иерархиями приоритетов и маскирования.

#### **1.3.4. Режим прямого доступа к памяти**

При обмене данными между внешним устройством и запоминающим устройством в режиме прерывания данные сначала передаются от этих устройств в микропроцессор, а затем из микропроцессора в устройства. При необходимости передачи из внешнего устройства в запоминающее устройство (или в обратном направлении) больших массивов данных с высокой скоростью целесообразно организовать передачу данных непосредственно, минуя микропроцессор. Такой способ передачи данных называется прямым доступом к памяти (ПДП). В режиме прямого доступа к памяти микропроцессор отключается от шин адреса и данных и передает управление ими контроллеру прямого доступа к памяти (КПДП). Контроллер прямого доступа к памяти реализует аппаратно-управляемый (внепроцессорный) обмен данными между запоминающим устройством и внешними устройствами. Контроллер может управлять в режиме прямого доступа к памяти передачей как отдельных слов, так и передачей блоков данных.

С этой целью контроллер прямого доступа к памяти формирует и модифицирует адреса участвующих в передаче данных элементов памяти ОЗУ, задаёт размер блока данных, который подлежит передаче, ведет подсчет слов, передаваемых в ОЗУ или из ОЗУ, определяет момент завершения передачи, формирует управляющие сигналы для микропроцессора, ОЗУ и внешних устройств, обеспечивая их согласованную работу. Настраивается контроллер на тот или иной режим работы путем использования управляющих сигналов устройства управления.

В качестве примера рассмотрим контроллер прямого доступа к памяти К580ВТ57 (рис. 1.14). Он имеет 4 независимых канала прямого доступа к памяти. В состав каждого из которых входит 16-разрядный регистр адреса, 14-разрядный счетчик длины блока данных и 2-разрядный регистр режима канала [1, 11]. В каждом канале формируется выходной сигнал запроса прямого доступа в память (ЗПДП; DRQ - Direct Request) и выходной сиг-

нал подтверждения прямого доступа в память (ППДП; DACK – Direct Acknowledged).

Буфер шины данных обеспечивает связь контроллера прямого доступа к памяти с шиной данных микропроцессора. На этапе инициализации микропроцессор загружает через буферное устройство в контроллер начальные адреса областей элементов памяти ОЗУ, размер блоков данных и коды режимов. В процессе работы микропроцессор может считывать содержимое регистров и счетчиков, кроме регистров режима.

Устройство управления контроллера прямого доступа к памяти и схема управления вводом-выводом в зависимости от комбинации управляющих сигналов переводят контроллер в одно из двух возможных состояний: контроллер является ведомым на этапе инициализации, а также при считывании содержимого регистров по инициативе микропроцессора; ведущим – в режиме прямого доступа к памяти.

Схема управления приоритетами реализует фиксированный или циклический режимы распределения приоритетов между четырьмя каналами контроллера прямого доступа к памяти.



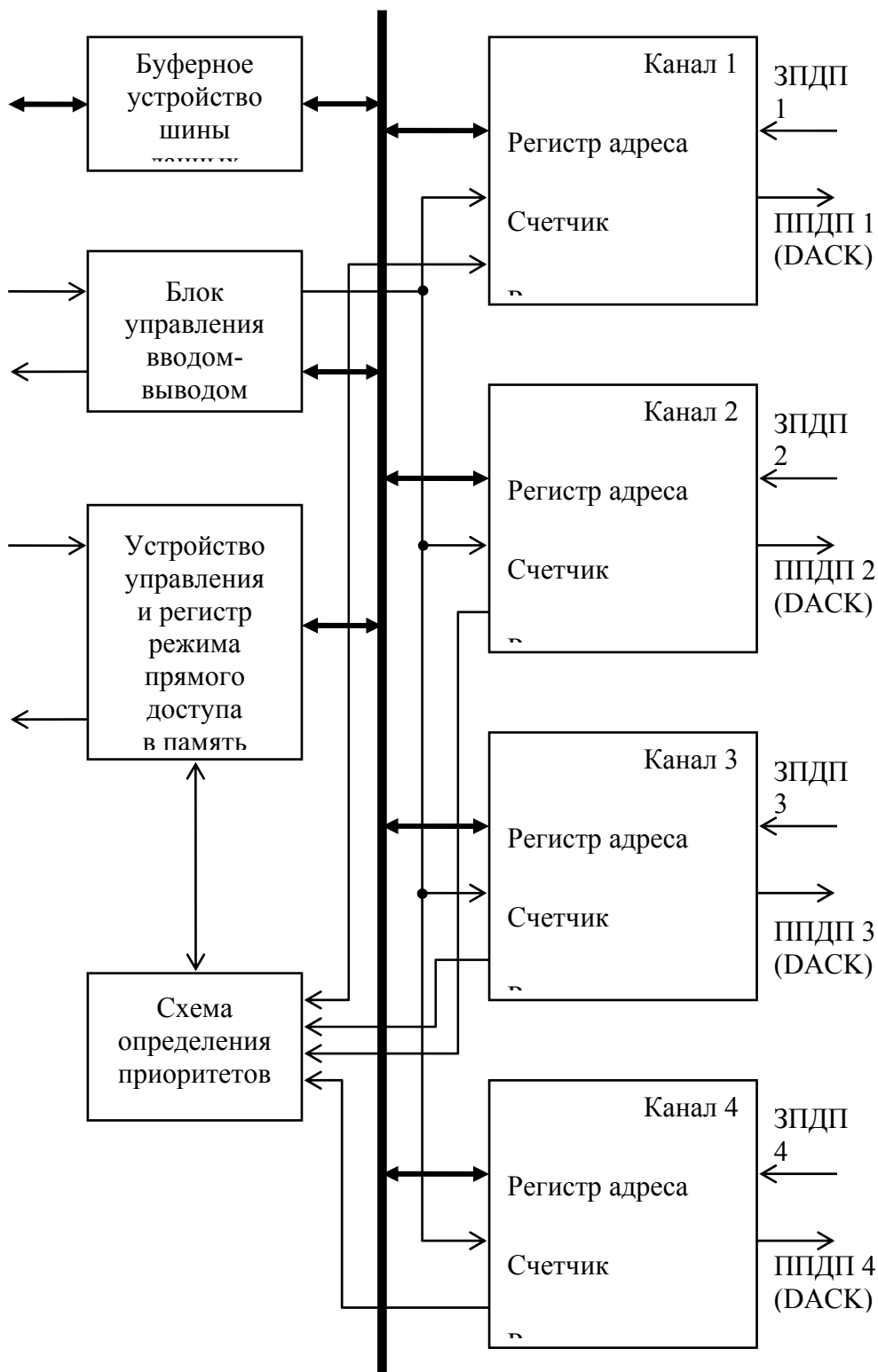


Рис. 1.14 Структурная схема контроллера прямого доступа к памяти

### 1.3.5. Организация интерфейса с клавиатурой

Основную часть клавиатуры (рис. 1.15) составляет матрица, имеющая четыре вертикальных и четыре горизонтальных шины. Каждая вертикальная шина соединена с каждой горизонтальной с помощью клавиш (сенсоров). Клавишей может быть обычная механическая клавиша, либо, напри-

мер, транзистор, касаясь затвора которого можно перевести его в проводящее состояние.

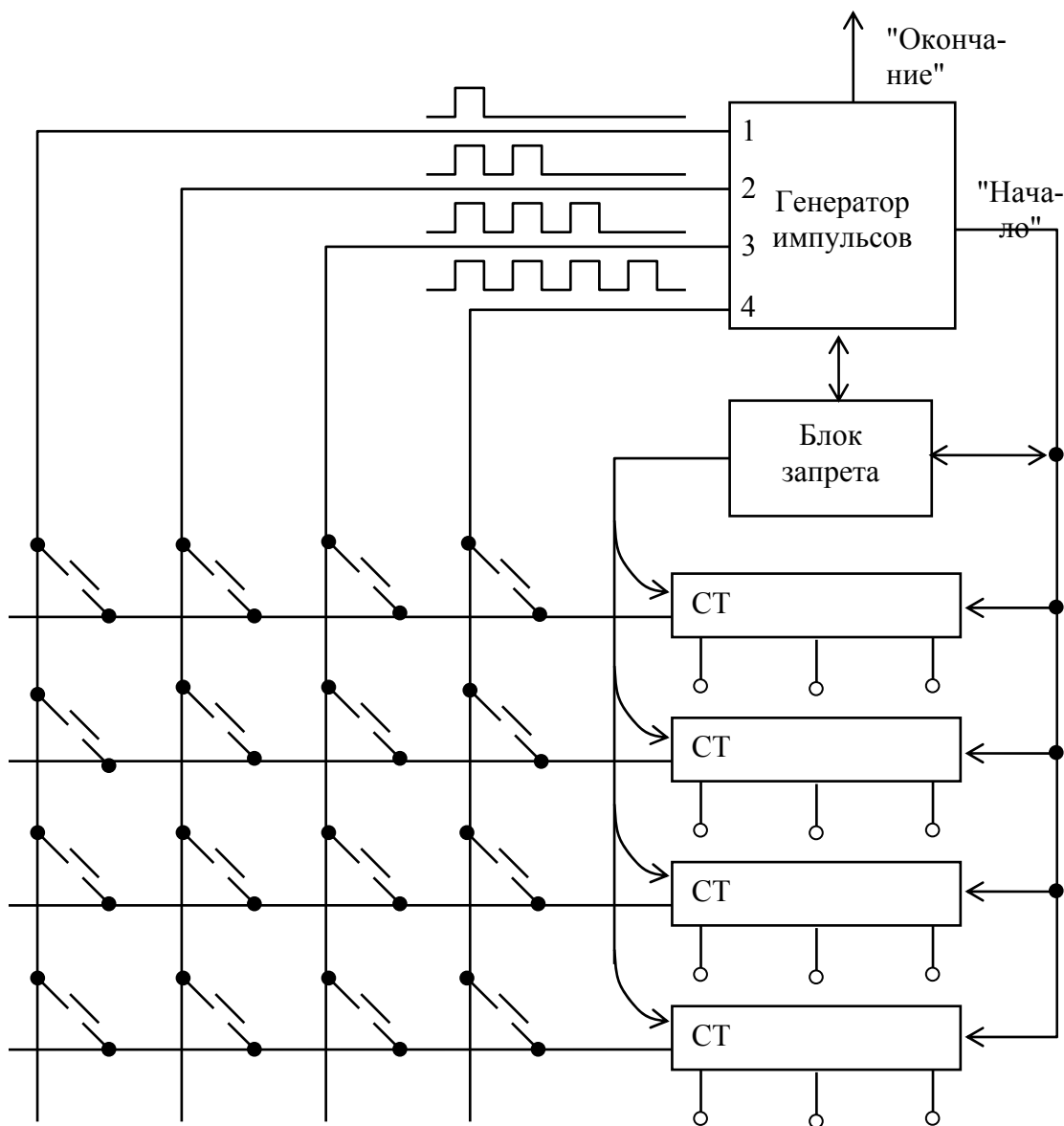


Рис. 1.15 Схема устройства организации интерфейса с клавиатурой

Устройство (рис.1.15) переводящее буквы алфавита в набор логических нулей и единиц, т.е. на язык, воспринимаемый микропроцессором, содержит генератор импульсов с шестью выходами. Принцип действия такого генератора следующий. В определенный момент времени на выходе “Начало” генератора формируется импульс, который подается на входы “Сброс” четырех двоичных счетчиков СТ и переводит их в такое состояние, когда на трех выходах присутствует комбинация логических нулей 000. Далее на первом выходе генератора формируется один импульс, на

втором - группа, состоящая из двух импульсов, на третьем – группа из трех и, наконец, на четвертом – группа из четырех импульсов. После завершения группы из четырех импульсов генератор импульсов вырабатывает сигнал “Окончание”.

Сигнал “Начало” можно рассматривать, как сигнал прерывания. Если нажата какая-либо из клавиш, то на вход соответствующего счетчика поступят один, два, три или четыре импульса, что приведет к появлению соответствующей комбинации логических нулей и единиц на его выходах. При этом допускается нажатие не более одной клавиши, так как дальнейшее поступление импульсов на вход того же счетчика не вызовет дополнительных изменений, что контролирует блок запрета. После того, как поступление импульсов на входы счетчиков прекратится (клавиша отпущена), блок запрета переведет устройство в исходное состояние. Сигнал “Окончание” свидетельствует о завершении процесса перевода языка клавиатуры в коды нулей и единиц, а также возможности передачи логических значений с выходов счетчиков в шину данных микропроцессора.

Двенадцать выходов четырех счетчиков дают возможность получить в общем случае 4096 возможных различных комбинаций логических значений нулей и единиц при числе клавиш равном 16.

В заключении раздела отметим, что в состав микропроцессора или МПС входит генератор тактовых импульсов или тактовый генератор. Он может быть размещен на кристалле микропроцессора. Этот генератор создает основу всей службы синхронизации микропроцессорной системы. Любой управляющий сигнал так или иначе соотносится во времени с частотой следования тактовых импульсов. Частота следования может быть от нескольких сотен килогерц до гигагерц. Эта частота определяет быстродействие микропроцессорной системы. Как правило, тактовые генераторы синхронизируются с помощью кварцевых резонаторов, что позволяет использовать их при необходимости для точного измерения или задания интервалов времени прерываний и дисциплины обслуживания прерываний.

## **1.4 Система команд и языки программирования микропроцессоров**

### **1.4.1 Система команд**

Изменение состояния микропроцессорной системы осуществляется с помощью микрокоманд (см.раздел 1.1), представляющих собой управляющие сигналы в виде комбинации логических значений нулей и единиц. В любой микропроцессорной системе возможное количество микрокоманд

значительно меньше возможного числа состояний МПС. Различие между числом состояний и числом микрокоманд объясняется тем, что каждый управляющий сигнал в микрокоманде может определять состояния одновременно нескольких логических элементов.

В запоминающих устройствах микропроцессорной системы микрокоманды хранятся в сокращенной форме, которая является командой (см. разд. 1.1). Язык команд представляет собой язык более высокого уровня по сравнению с языком микрокоманд. Он тоже использует комбинации логических значений нулей и единиц, но они уже не являются управляющими сигналами.

Для перехода с языка команд на язык микрокоманд используются, хранимые в ПЗУ словари, а так же программируемые логические матрицы (ПЛМ). Такие средства перехода могут входить в состав БИС микропроцессора или в состав отдельных БИС микропроцессорной системы. Так, микропроцессор КР580ВМ80А с фиксированным набором команд содержит в своем составе программируемые логические матрицы. В микропроцессоре КР588ВС2 набор команд можно изменять, заменяя соответствующую БИС.

Рассмотрим в качестве примера систему команд микропроцессора КР580ВМ80А, т. е. микропроцессора с фиксированным набором команд, который в наибольшей степени отражает особенности микропроцессорной системы. Команды, так же как и микрокоманды, имеют свой фиксированный формат. В частности, команды рассматриваемого набора по формату подразделяются на три основные группы: однобайтные, двухбайтные и трехбайтные. Следовательно, в зависимости от формата, команда может содержать 8, 16 или 24 двоичных символов.

Ячейка запоминающего устройства имеет емкость 8 двоичных символов или один байт. Хранение трехбайтной команды требует использование трех ячеек ЗУ с тремя различными адресами. Очевидно, что адреса всех трех ячеек ЗУ должны быть взаимосвязаны и в простейшем случае представлять собой три последовательно расположенных целых числа. Для извлечения или записи трехбайтной команды требуются три последовательных цикла обращения к запоминающему устройству. Первый байт размещается в регистре команд, а два остальных – в регистрах *W* и *Z* (рис. 1.4).



Рис. 1.16. Форматы команд микропроцессора КР580ВМ80А

Формат однобайтной команды показан на рис. 1.16,а. Однобайтная команда состоит из одного кода операции КО. Формат двухбайтной команды приведен на рис.1.16,б. Двухбайтная команда содержит код операции и 8-разрядный операнд, обозначенный на рисунке, как *D8*. Операндом называется слово, над которым непосредственно выполняется операция, предусмотренная кодом операции. Наконец, трехбайтная команда имеет формат, показанный на рис.1.16,в. Она состоит из двух частей, одна из которых является кодом операции (1 байт), а вторая может быть либо адресом (в этом случае она обозначается буквами *ADR*), либо 16-разрядным операндом *D16*.

Код операции, являясь 8-разрядным двоичным словом, может принимать 256 значений. Столько же различных команд входит в состав рассматриваемого микропроцессора. При описании наборов команд пользуются мнемоническими обозначениями или мнемоническими кодами. Мнемонический код представляет собой несколько букв алфавита, сочетание которых напоминает некоторое слово или фразу на русском, английском или другом языке. В микропроцессоре КР580ВМ80А принята мнемоника, основанная на английском языке.

По назначению команды делятся на пять групп:

- команды пересылок;
- арифметические и логические команды;
- команды ввода-вывода;
- команды управления;
- специальные команды.

### **Команды пересылок**

Они делятся на команды однобайтовых или двухбайтовых пересылок и команды обмена байтами. Типичным примером команды однобайтовой пересылки служит команда межрегистровой пересылки, мнемонический код которой имеет вид:

*MOV R1, R: R→R1*

В левой части (до двоеточия) записан мнемонический код команды, а в правой части записана ее реализация в символической форме. Буквы *MOV* взяты являются начальными буквами двух английских слов «*Move register*», что означает «Осуществить пересылку между двумя регистрами». Команда межрегистровой пересылки *MOV R1, R* представляет собой группу операций, отличающихся друг от друга конкретными регистрами, между которыми осуществляется пересылка. В частности, на место буквы *R* в мнемоническом коде подставляется имя конкретного регистра из которого осуществляется пересылка, а на место символов *R1* ставится имя регистра в который осуществляется пересылка слова. Эта процедура символически изображается в виде:

$R \rightarrow R1$

Знак " $\rightarrow$ " обозначает помещение слова в ячейку запоминающего устройства (регистр), код которой указан у острия стрелки. Передаваемое слово извлекается из ячейки ЗУ (регистра), имя которой указано слева от стрелки.

Для указанной группы команд межрегистровых пересылок, использующей регистровую адресацию, однобайтовая команда имеет следующий вид:

01 *DDD SSS*.

Два старших разряда образуют код операции пересылки. Три разряда, обозначенные буквами *DDD*, соответствуют коду регистра в который передается информация (*destination*). Три разряда, обозначенные буквами *SSS*, соответствуют коду регистра из которого передается информация (*source*). Коды регистров определяются таблицей 1.2. В системе команд имеется 63 команды типа *MOV*, которые представлены в табл. 1.3.

Таблица 1.2

Десятичный код регистра	Источник-приемник <i>SSS/DDD</i>	Регистр
0	000	<i>B</i>
1	001	<i>C</i>
2	010	<i>D</i>
3	011	<i>E</i>
4	100	<i>H</i>
5	101	<i>L</i>
6	110	<i>M</i>

7	111	A
---	-----	---

Таблица 1.3

Код	0000	0001	0010	0011	0100	0101	0110	0111
0100	<i>MOV</i> <i>B,B</i>	<i>MOV B</i> <i>,C</i>	<i>MOV</i> <i>B,D</i>	<i>MOV</i> <i>B,E</i>	<i>MOV</i> <i>B,H</i>	<i>MOV</i> <i>B,L</i>	<i>MOV</i> <i>B,M</i>	<i>MOV</i> <i>B,A</i>
0101	<i>MOV</i> <i>D,B</i>	<i>MOV</i> <i>D,C</i>	<i>MOV</i> <i>D,D</i>	<i>MOV</i> <i>D,E</i>	<i>MOV</i> <i>D,H</i>	<i>MOV</i> <i>D,L</i>	<i>MOV</i> <i>D,M</i>	<i>MOV</i> <i>D,A</i>
0110	<i>MOV</i> <i>H,B</i>	<i>MOV</i> <i>H,C</i>	<i>MOV</i> <i>H,D</i>	<i>MOV</i> <i>H,E</i>	<i>MOV</i> <i>H,H</i>	<i>MOV</i> <i>H,L</i>	<i>MOV</i> <i>H,M</i>	<i>MOV</i> <i>H,A</i>
0111	<i>MOV</i> <i>M,B</i>	<i>MOV</i> <i>M,C</i>	<i>MOV</i> <i>M,D</i>	<i>MOV</i> <i>M,E</i>	<i>MOV</i> <i>M,H</i>	<i>MOV</i> <i>M,L</i>	<i>HLT</i>	<i>MOV</i> <i>M,A</i>

Код	1000	1001	1010	1011	1100	1101	1110	1111
0100	<i>MOV</i> <i>C,B</i>	<i>MOV</i> <i>C,C</i>	<i>MOV</i> <i>C,D</i>	<i>MOV</i> <i>C,E</i>	<i>MOV</i> <i>C,H</i>	<i>MOV</i> <i>C,L</i>	<i>MOV</i> <i>C,M</i>	<i>MOV</i> <i>C,A</i>
0101	<i>MOV</i> <i>E,B</i>	<i>MOV</i> <i>E,C</i>	<i>MOV</i> <i>E,D</i>	<i>MOV</i> <i>E,E</i>	<i>MOV</i> <i>E,H</i>	<i>MOV</i> <i>E,L</i>	<i>MOV</i> <i>E,M</i>	<i>MOV</i> <i>E,A</i>
0110	<i>MOV</i> <i>L,B</i>	<i>MOV</i> <i>L,C</i>	<i>MOV</i> <i>L,D</i>	<i>MOV</i> <i>L,E</i>	<i>MOV</i> <i>L,H</i>	<i>MOV</i> <i>L,L</i>	<i>MOV</i> <i>L,M</i>	<i>MOV</i> <i>L,A</i>
0111	<i>MOV</i> <i>A,B</i>	<i>MOV</i> <i>A,C</i>	<i>MOV</i> <i>A,D</i>	<i>MOV</i> <i>A,E</i>	<i>MOV</i> <i>A,H</i>	<i>MOV</i> <i>A,L</i>	<i>MOV</i> <i>A,M</i>	<i>MOV</i> <i>A,A</i>

В табл. 1.3 в заголовке каждой строки даны комбинации из четырех старших двоичных символов кода операций, а в заголовке каждого столбца изображены комбинации из четырех младших двоичных символов кода операций. Например, первая строка таблицы имеет заголовок 0100, третий слева столбец - 0010. На пересечении первой строки и третьего столбца обозначено: *MOV B, D*. Следовательно, команда с кодом операции 0100 0010 вызывает пересылку содержимого регистра *D* в регистр *B* регистров общего назначения микропроцессора. Восемьразрядное двоичное слово из регистра *D* пересылается в регистр *B* и заменяет ранее находившееся там слово, причем содержимое регистра *D* сохраняется, что является важнейшим условием для всех пересылок.

Все перечисленные в табл. 6.2 комбинации символов содержат адреса. Однако при пересылках слов из регистра в регистр имеются особенности. Так, код операции 0100 0000 соответствует записи *MOV B,B*. В этом случае

из регистра  $B$  извлекается слово и помещается в тот же регистр  $B$  в инверсной форме. В операции участвует один регистр общего назначения. При этом предыдущее содержимое регистра уничтожается и вместо операции пересылки выполняется операция инверсии. В регистре  $B$  вместо кода 0100 0010 будет записан код 1011 1101.

Для кода операции 0100 0110 мнемонический код имеет вид: *MOV B,M*. Буквой  $M$  обозначается ячейка запоминающего устройства, адрес которой хранится в двух регистрах  $H$  и  $L$ . Код операции 0100 0110 требует выполнения следующей последовательности действий: извлечь содержимое двух регистров  $H$  и  $L$ , использовать его в качестве адреса ячейки ЗУ, содержимое которой поместить в регистр  $B$  микропроцессора. Содержимое ячейки ЗУ при этом остается неизменным.

Коду операции 0100 0111 соответствует мнемонический код *MOV B,A*. Буквой  $A$  обозначен регистр-аккумулятор. По этой команде содержимое регистра-аккумулятора передается в регистр  $B$  и сохраняется в самом регистре-аккумуляторе.

В четвертой сверху строке табл. 1.3 обозначена комбинация 0111. Первые шесть команд этой строки означают, что содержимое соответствующих регистров  $B$ ,  $C$ ,  $D$ ,  $E$ ,  $H$  или  $L$  передается в ячейку запоминающего устройства, адрес которой записан в двух регистрах  $H$  и  $L$ . Заметим, что имеется особенность команд с кодами 0111 0100 и 0111 0101. Каждая из них предусматривает, что в ячейку запоминающего устройства с некоторым адресом должна быть передана половина этого адреса. Команда с кодом операции 0111 0110 не относится к рассматриваемому классу. Это команда *HLT*, по которой прекращается выполнение операций.

Восемь команд последней строки в табл. 1.3 предусматривают передачу в регистр-аккумулятор содержимого любого из регистров общего назначения или ячейки запоминающего устройства, адрес которой хранится в двух регистрах  $H$  и  $L$ . Эти же команды предусматривают передачу содержимого регистра-аккумулятора в тот же регистр, но после инверсии.

Каждый код операции указывает ту последовательность действий, которая должна быть выполнена в микропроцессорной системе.

Кроме рассмотренной группы команд имеются шесть семейств операций однобайтовых пересылок.

О п е р а ц и и *MVI R, D: D8 → R*. Наличие символа  $D8$  означает, что команда имеет двухбайтовый формат. Смысл этого семейства операций заключается в том, что операнд  $D8$ , т. е. содержимое второго байта команды,



передается в регистр, имя которого стоит на месте символа  $R$ . В этом семействе операций восемь команд. Они позволяют записать операнд в регистр - аккумулятор, в любой регистр общего назначения и в любую ячейку запоминающего устройства, адрес которой хранится в регистровой паре  $H$  и  $L$ . Мнемонический код  $MVI$  происходит от английских слов «*Move immediate*», что означает «переслать слово, входящее непосредственно в состав команды».

Семейство операций  $STAX\ YZ: A \rightarrow M(YZ)$ . Символом  $YZ$  обозначается содержимое регистровой пары  $B$  и  $C$  или  $D$  и  $E$ . Ячейка запоминающего устройства с адресом, помещенным в скобки, обозначается символом  $M(YZ)$ . Адрес этой ячейки запоминающего устройства содержится в регистровой паре, имя которой помещено на место символов  $YZ$ . По командам этого семейства операций содержимое регистра - аккумулятора передается в соответствующую ячейку запоминающего устройства. Эти команды, соответствующие двум возможным различным регистровым парам, имеют однобайтовый формат. Регистровая пара именуется именем первого регистра. Мнемонические коды команд будут иметь вид:  $STAX\ B$ , либо  $STAX\ D$ .

Семейство команд  $LDAX\ YZ: M(YZ) \rightarrow A$ . По командам этого семейства происходит косвенная загрузка регистра - аккумулятора по аналогии с командами семейства операций  $STAX\ YZ$ .

Семейство команд  $STA\ ADR: A \rightarrow M(ADR)$ . Мнемонический код  $STA\ ADR$  происходит от английских слов. «*Store accumulator direct*», что означает «прямое размещение содержимого аккумулятора». Это семейство состоит из одной команды с трехбайтовым форматом, при выполнении которой содержимое регистра-аккумулятора передается в ячейку запоминающего устройства, адрес которой записан во втором и третьем байтах команды.

Команда  $LDA\ ADR: M(ADR) \rightarrow A$ . Эта команда по своему смыслу является обратной к предыдущей.

Команда  $SPhL: HL \rightarrow SP$ . По данной команде содержимое регистровой пары  $H$  и  $L$  передается в регистр - указатель стека, символическое обозначение  $SP$  которого образовано из начальных букв английских слов «*Stack pointer*», что и означает «указатель стека».

Следующую группу команд составляют команды двухбайтовых пересылок, отличающиеся от рассмотренной выше группы команд тем, что пересылаются не один, а два байта. Символические обозначения первых трех

команд двухбайтовых пересылок имеют следующий вид:

*LXI YZ, D16: D16* → *YZ*;

*SHLD ADR: HL* → *M (ADR), M (ADR + 1)*;

*LHLD ADR: M (ADR), M (ADR + 1)* → *HL*

Символом *YZ* обозначается содержимое регистровых пар *B* и *C*, *D* и *E*, *H* и *L* или регистра – указателя стека *SP*. По второй команде содержимое регистровой пары *H* и *L* передается в две ячейки запоминающего устройства с адресами *ADR* и *ADR+1*. Во всех операциях, выполняемых над словом длиной два байта, указывается адрес одной ячейки запоминающего устройства, а обращение производится к двум соседним ячейкам запоминающего устройства.

Третья команда по смыслу является противоположной предыдущей команде.

Символические обозначения двух оставшихся команд двухбайтовых пересылок имеют вид:

*PUSH YZ: YZ* → *M(SP-1), M(SP-2), SP-2* → *SP*

*POP YZ: M (SP), M (SP + 1)* → *YZ; SP + 2* → *SP*.

Символом *YZ* обозначается содержимое регистровых пар *B* и *C*, *D* и *E*, *H* и *L* или *PSW*. Команда *PUSH* предусматривает пересылку содержимого одной из перечисленных выше регистровых пар в две соседние ячейки ЗУ, адрес одной из которых записывается как *SP-1*. Содержимое регистра-указателя стека равно минус 1. Адрес второй ячейки ЗУ равен *SP-2*, так что содержимое регистра-указателя стека равно минус 2. При этом в регистр-указатель стека также засылается число *SP-2*.

Смысл стекового заполнения ячеек запоминающего устройства состоит в том, что к моменту начала любой операции в регистре - указателе стека хранится адрес последней занятой ячейки ЗУ.

По команде *PUSH* заполняются две следующие по порядку ячейки ЗУ, а в регистр-указатель стека вновь помещается число (в данном случае *SP-2*), представляющее собой адрес последней заполненной ячейки запоминающего устройства.

Последняя команда в рассматриваемой группе (команда *POP*), является противоположной предыдущей команде. По ней содержимое двух последовательных ячеек запоминающего устройства с адресами *SP* и *SP+1* передается в соответствующую регистровую пару. При этом число, хранящееся в регистре-указателе стека, увеличивается на 2.

К командам пересылок относятся также команды обмена байтами.

Символические описания этих команд следующие:

$XCHG: HL \leftrightarrow DE;$

$XTHL: H \leftrightarrow M(SP + 1), L \leftrightarrow M(SP).$

### **Команды арифметических и логических операций**

Эта группа команд делится на четыре подгруппы. К первой подгруппе относятся команды, выполняющие арифметические и логические операции с одним операндом. Первые три из них имеют следующие символические описания:

$CMC: \bar{C} \rightarrow C;$

$STC: 1 \rightarrow C;$

$CMA: \bar{A} \rightarrow A.$

По первой команде логическое значение признака переноса  $C$  инвертируется. Во второй команде логическое значение признака переноса устанавливается равным 1. Третья команда предполагает инвертирование содержимого регистра результата  $A$ . В этой подгруппе отдельно стоит команда десятичной коррекции  $DAA$ . Команда используется при выполнении арифметических операций сложения и вычитания над десятичными числами. Данную подгруппу команд с одним операндом завершают четыре команды, символические описания которых имеют вид:

$INR R: R+1 \rightarrow R$  (*increment register*);

$DCR R: R-1 \rightarrow R$  (*decrement register*);

$INX YZ: YZ+1 \rightarrow YZ$  (*increment register pair*);

$DCX YZ: YZ-1 \rightarrow YZ$  (*decrement register pair*);

Эти команды предусматривают увеличение или уменьшение на единицу соответствующего 8- или 16-разрядного числа. При выполнении команд  $INR R$ , а также  $DCR R$ , могут изменяться логические значения всех признаков, кроме признака переноса  $C$ .

Следующую подгруппу составляют арифметические и логические операции с двумя операндами. Символические описания первых четырех из них имеют вид:

$ADD R: A+R \rightarrow A$  (*add register*);

$ADC R: A+R+C \rightarrow A;$

$SUB R: A-R \rightarrow A$  (*subtract register*);

$SBB R: A-R-C \rightarrow A.$

Арифметическая операция всегда выполняется над двумя числами, одно из которых хранится в регистре результата, а второе – в одном из ре-

гистров общего назначения. Результат засылается в регистр результата. При выполнении команд  $ADC R$  и  $SBB R$  учитывается значение признака переноса  $C$ .

Символическое описание следующих трех команд рассматриваемой подгруппы записывается в форме:

$$ANA R: A \wedge R \rightarrow A;$$

$$ORA R: A \vee R \rightarrow A;$$

$$XPA R: A \oplus R \rightarrow A.$$

Первой из них является команда, предусматривающая выполнение логической операции И над восемью парами разрядов двух слов. Команда  $ORA R$  предусматривает выполнение логической операции ИЛИ над восемью парами разрядов двух слов. Выполнение над восемью парами разрядов двух слов логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ выполняется по команде  $XPA R$ .

Следующие четыре команды аналогичны командам  $ADD R$ ,  $ADC R$ ,  $SUB R$ ,  $SBB R$  рассматриваемой подгруппы с тем отличием, что в качестве второго операнда используется второй байт ( $D8$ ) двухбайтовой команды:

$$ADI D8: A + D8 \rightarrow A;$$

$$ACI D8: A + D8 + C \rightarrow A;$$

$$SUI D8: A - D8 \rightarrow A;$$

$$SBI D8: A - D8 - C \rightarrow A.$$

При выполнении следующих трех команд:

$$ANI D8: A \wedge D8 \rightarrow A;$$

$$ORI D8: A \vee D8 \rightarrow A;$$

$$XRI D8: A \oplus D8 \rightarrow A$$

в качестве второго операнда используется второй байт двухбайтовой команды.

Следующими двумя командами рассматриваемой подгруппы являются команды сравнения. Первая из них, имеющая мнемоническое описание  $CPI D8$ , предусматривает сравнение двух чисел, одно из которых хранится в регистре результата, а второе - представляет собой байт  $D8$  двухбайтовой команды. Если в результате выполнения операции оказывается, что числа равны, то признаку нуля  $Z$  присваивается значение 1. Если же число, хранимое в регистре результата, окажется меньше операнда  $D8$ , то логическое значение 1 присваивается признаку переноса  $C$ .

Последняя команда  $CMP R$  рассматриваемой подгруппы отличается от предыдущей тем, что в качестве второго операнда используется содержи-

мое одного из регистров общего назначения.

Следующая подгруппа команд арифметических и логических операций состоит из одной команды:

*DAD YZ: HL+YZ→HL,*

предусматривающей выполнение операции сложения над двумя 16-разрядными (двухбайтовыми) двоичными числами. Смысл этой команды раскрывается ее символическим описанием. Под *YZ* понимается содержимое регистровых пар *B* и *C*, *D* и *E*, *H* и *L* или регистра *SP*. В отличие от команд предыдущей подгруппы, при выполнении которых могут изменяться логические значения всех признаков, при выполнении данной команды может изменяться лишь значение признака переноса *C*.

Последнюю подгруппу составляют четыре команды сдвига:

*RAR (rotate accumulator right through carry);*

*RRC (rotate right);*

*RAL (rotate accumulator left through carry);*

*RLC (rotate accumulator left).*

Эти команды предусматривают выполнение операции сдвига содержимого регистра результатов. Порядок выполнения этих команд иллюстрируется на рис. 1.16.

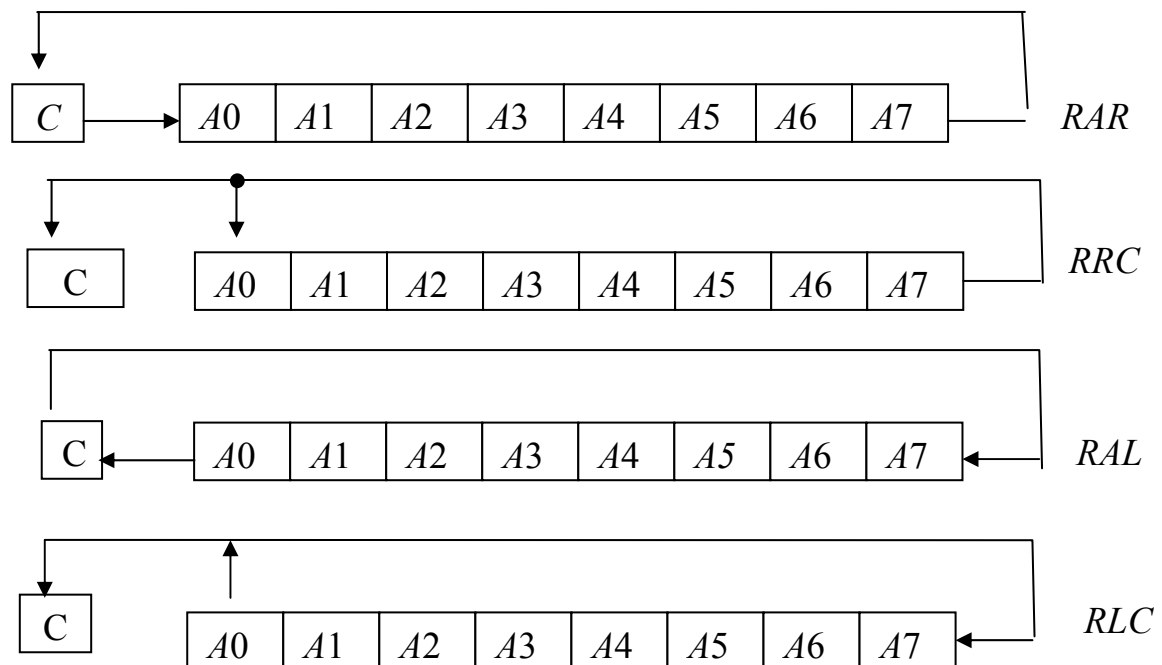


Рис. 1.17 Порядок выполнения операций сдвига

По каждой из этих команд логическое значение данного разряда заменяется логическим значением соседнего разряда. При командах *RAR* и *RRC* подобная замена производится слева направо (рис.1.17). Соответст-

вующие операции называются сдвиг вправо. При этом логическое значение разряда  $A_7$  присваивается признаку переноса  $C$ .

При выполнении команды  $RAR$  разряду  $A_0$  регистра присваивается логическое значение признака переноса  $C$ . По команде  $RRC$  логическое значение, которое до выполнения операции имел разряд  $A_7$ , присваивается разряду  $A_0$ . Подобная операция называется циклическим сдвигом. Из рис. 1.17 видно, что по командам  $RAL$  и  $RLC$  выполняются операции сдвига, аналогичные двум уже рассмотренным, за исключением того, что сдвиг производится не вправо, а влево. При выполнении операции сдвига значения всех признаков, кроме признака переноса  $C$ , остаются неизменными.

Микропроцессор  $KP580BM80A$  выполняет две арифметические операции - операции сложения и вычитания. Эти операции широко используются при реализации алгоритмов обработки сигналов в реальном масштабе времени, в частности для вычисления интеграла свертки в цифровых согласованных фильтрах, применяемых в демодуляторах сигналов телекоммуникационных и спутниковых систем связи.

### **Команды ввода-вывода**

Мнемонические коды и символическое описание таких команд имеют следующий вид:

$IN\ N: (N) \rightarrow A$

$OUT\ N: A \rightarrow (N)$ .

Символом  $N$  обозначается номер порта или программируемого периферийного адаптера. Число, хранящееся в регистре порта, символически обозначается в виде:  $(N)$ .

Особенность этих команд и, соответственно, особенность процессора  $KP580BM80A$  заключается в том, что по одной команде может быть введено или выведено только одно слово. Следовательно, при вводе или выводе группы слов, например, при обмене данными с магнитными дисками, соответствующая команда ввода или вывода должна быть повторена столько раз, сколько слов (байт) содержит группа. Перед каждым повторением операции ввода или вывода необходимо соответствующим образом изменить содержимое регистра результата  $A$ . При вводе группы слов передачи содержимого регистра  $A$  в запоминающее устройство команды должны выполняться несколько раз подряд. Следовательно, необходимо обеспечить формирование адресов ячеек памяти запоминающего устройства. Например, начальный адрес ячейки для каждой команды нужно увеличивать на единицу.

Таким образом, операция ввода-вывода группы слов, ввод или вывод отдельного байта требует не одну, а несколько последовательно выполняемых команд.

### Команды управления

Эта группа команд состоит из трех подгрупп. Первую подгруппу составляют команды передачи управления. Мнемонический код и символическое описание первой команды передачи управления имеет вид: *PCHL*:

*HL* → *PC*

По команде *PCHL* в счетчик адресов команд *PC* передается содержимое регистровой пары *H* и *L*. Это содержимое служит адресом ячейки памяти запоминающего устройства, из которой будет извлечена новая подлежащая выполнению, команда. Эта операция является операцией безусловного перехода по программе.

Вторая команда передачи управления записывается в форме:

*JMP ADR: ADR* → *PC*.

Это команда также предусматривает операцию безусловного перехода по программе. Ее отличие от предыдущей состоит в том, что в счетчик адресов команд передается содержимое двух байт трехбайтовой команды, обозначенное как *ADR*. После выполнения такой операции следующей выполняемой командой будет команда, код операции которой хранится в ячейке памяти запоминающего устройства с адресом *ADR*.

В состав арифметико-логического устройства микропроцессора КР580ВМ80А входит регистр признаков (регистр *F*), представляющий собой 8-разрядный регистр, каждый разряд которого имеет определенное назначение (рис. 1.18).

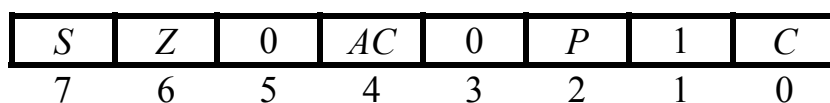


Рис. 1.18 Регистр признаков микропроцессора КР580ВМ80А

В нулевой разряд регистра записывается признак переноса *C*, которому присваивается логическое значение 1 в том случае, когда потребность в операции переноса возникает при сложении (вычитании) старших разрядов двух 8-разрядных двоичных чисел. Поскольку операции сложения и вычитания выполняются над 8-разрядными двоичными числами, то «единица в уме», полученная при сложении (вычитании) двух старших цифр записывается в нулевой разряд регистра признаков и учитывается при выполнении следующей операции, либо служит сигналом, указывающим на воз-

никшую ситуацию.

В следующем (на рис. 1.18 справа налево) разряде регистра признаков постоянно хранится логическая единица. В разряде  $P$  (*parity*) хранится признак четности. Этот признак принимает логическое значение 1, если число единиц в двоичном слове, хранящемся в регистре результата (регистре  $A$ ), четное. При нечетном числе единиц признак четности имеет значение 0.

В третьем и пятом разрядах регистра признаков постоянно хранятся логические нули. В четвертом разряде записывается признак  $AC$ , представляющий собой признак вспомогательного переноса. Этот признак принимает логическое значение 1 в том случае, когда перенос возникает при выполнении сложения (вычитания) над четвертыми разрядами двоичных чисел. Признак  $Z$  (*zero*), хранящийся в шестом разряде регистра признаков, принимает логическое значение 1, когда в результате выполнения команды образуется слово 0000 0000. В этом случае содержимое регистра  $A$  равно нулю. Признак знака  $S$  (*sign*) (седьмой разряд) принимает логическое значение 1, если в результате выполнения команды получается отрицательное число.

Результат выполнения команды всегда состоит из двух частей: двоичного слова (числа), поступившего в регистр результатов (регистр  $A$ ), и двоичного слова, хранящегося в регистре признаков. Эти два слова получили название: **с л о в о с о с т о я н и я п р о г р а м м ы**. Содержимое регистра результатов и регистра признаков достаточно подробно характеризует то состояние, к которому привело выполнение последовательности операций на данный момент времени. Символическое обозначение слова состояния программы имеет вид  $PSW$  (*Programmer state word*).

Во вторую подгруппу команд входит восемь команд условного перехода. Эти команды описываются одним мнемоническим кодом и одним символическим описанием:

$J[CON] ADR: ADR \rightarrow PC$  (код команды 11CCC 010, в котором биты CCC являются кодами условий).

При написании команд на место  $CON$  подставляется одно из следующих восьми условий:

$NZ$  – ненулевой результат; признак  $Z = 0$  ( $CCC = 000$ ).

$Z$  – нулевой результат;  $Z = 1$  ( $CCC = 001$ ).

$NC$  – отсутствие переноса;  $C = 0$  ( $CCC = 010$ ).

$C$  – перенос;  $C = 1$  ( $CCC = 011$ ).



*PO (parity omen)* – число единиц в регистре результата нечетное;  $P = 0$  ( $CCC = 100$ ).

*PE (parity even)* – число единиц в регистре результата четное;  $P = 1$  ( $CCC = 101$ ).

*P (plus)* – число неотрицательное;  $S = 0$  ( $CCC = 110$ ).

*M (minus)* – число отрицательное;  $S = 1$  ( $CCC = 111$ ).

Рассмотрим порядок выполнения команды *JNZ ADR*. Если результат, хранящийся в регистре результата является ненулевым, то признак *Z* равен нулю. В этом случае последовательность выполнения команд нарушается и следующей после данной будет исполняться команда, код операции которой хранится в ячейке памяти с адресом *ADR*. Если признак *Z* имеет значение логической 1, последовательность выполнения команд не нарушается и выполняется следующая по порядку команда. Отметим, что команда *JNZ ADR* имеет трехбайтный формат, поэтому, если код операции команды *JNZ ADR* хранится в ячейке памяти запоминающего устройства с адресом *k*, то код операции следующей по порядку команды должен храниться в ячейке памяти с адресом  $k+3$ . Следовательно, при выполнении команды *JNZ ADR*, если  $Z = 1$ , микропроцессор автоматически увеличивает содержимое счетчика адресов команд на 3.

Для остальных семи команд условного перехода порядок выполнения команд сохраняется тем же. Следует заметить, что приведенная процедура выполнения команд иллюстрирует роль регистра признаков. Условный переход по программе совершается или не совершается в зависимости от того, удовлетворяется или не удовлетворяется определенное условие, касающееся операндов.

Специальная подгруппа команд имеет своим назначением обращение к подпрограммам. Первая команда имеет вид:

*CALL ADR: PC* →  $M(SP-1)$ ;  $M(SP-2)$ ;  $SP-2$  → *SP*; *ADR* → *PC*.

Основное действие этой команды состоит в том, что величина *ADR* записывается в счетчик *PC* адресов команд. При этом имеется в виду, что *ADR* является адресом первой команды подпрограммы. Однако, в отличие от команд безусловного и условного переходов, в данном случае основному действию предшествует подготовка, состоящая в том, что предыдущее содержимое счетчика адресов команд передается в ячейки памяти запоминающего устройства с адресами  $SP-1$  и  $SP-2$ . Тем самым обеспечивается выход (возврат) из подпрограмм. Любая подпрограмма обязательно заканчивается командой безусловного возврата, символическое описание кото-

рой имеет вид:

$$RET: M(SP); M(SP+1) \rightarrow PC; SP+2 \rightarrow SP.$$

Как видно из этого описания, после выполнения подпрограммы в счетчик адресов команд передается содержимое тех ячеек памяти запоминающего устройства, куда перед началом выполнения подпрограммы было помещено содержимое счетчика адресов команд. Благодаря этому основная программа, прерванная обращением к подпрограмме, может продолжиться с того самого места, на котором она была прервана. Кроме того, при выполнении команды возврата содержимое указателя стека увеличивается на 2, т.е. восстанавливается его исходное значение.

При выполнении команды *CALL ADR* в стек засылается содержимое счетчика адресов команд, а не слово состояния программы. К моменту начала выполнения подпрограммы в регистр результата и регистр признаков в общем случае оказываются заполненными. Подпрограмма выполняется над операндами и первым из них оказывается является слово, хранящееся в регистре результата. Происходит автоматическая передача данных из программы в подпрограмму. В то же время можно представить ситуацию, когда подпрограмма искажает содержимое регистра результатов и регистра признаков и это сделает невозможным продолжение основной программы. Слово состояния программы (*PSW*) не сохраняется в стеке при выполнении команды обращения к подпрограмме. Если такое сохранение необходимо, то перед командой обращения к подпрограмме требуется выполнить команду *PUSH PSW*.

Кроме команд безусловного обращения к подпрограмме и безусловного возврата существуют их условные разновидности, символические описания которых имеют вид:

$$C[CON] ADR: PC \rightarrow M(SP-1); M(SP-2); SP-2 \rightarrow SP; ADR \rightarrow PC.$$
$$R[CON]: M(SP); M(SP+1) \rightarrow PC; SP+2 \rightarrow SP.$$

По своему действию обе эти команды эквивалентны рассмотренной выше команде обращения к подпрограмме. Различие состоит в том, что выполняются они при удовлетворении одного из восьми рассмотренных ранее условий. Наименование условия записывается на место *CON* так же, как это делалось для команд условного перехода.

Следующая разновидность команд обращения к подпрограммам имеет вид:

$$RSTX: PC \rightarrow M(SP-1); M(SP-2); SP-2 \rightarrow SP; ADR \rightarrow SP.$$

Число *ADR* зависит от конкретного значения числа *X*. Эта зависимость

задается табл.1.4. Команда *RST* предусматривает обращение к подпрограмме, первая команда которой хранится в одной из восьми фиксированных ячеек памяти запоминающего устройства.

Таблица 1.4

Код операции	<i>ADR</i>	Код операции	<i>ADR</i>
<i>RST 0</i>	0000	<i>RST 4</i>	0020
<i>RST 1</i>	0008	<i>RST 5</i>	0028
<i>RST 2</i>	0010	<i>RST 6</i>	0030
<i>RST 3</i>	0018	<i>RST 7</i>	0038

### Специальные команды

Эта группа состоит из четырех команд:

*EI* – разрешение прерывания;

*DI* – запрещение прерывания;

*HLT* – останов;

*NOP* – холостая операция.

Первые две команды реализуют механизм маскирования прерываний. По команде останов *HLT* прекращается выполнение операций, но сохраняется содержимое всех регистров. Наконец, по команде *NOP* операции не выполняются, но содержимое счетчиков адресов команд увеличивается на единицу и в следующем цикле выполняется следующая по порядку команда.

Заметим, что несмотря на то, что команды описываются мнемонической записью кода операций, на самом деле код операции представляет собой двоичное слово, состоящее из восьми двоичных символов.

#### 1.4.2. Языки программирования микропроцессоров

Рассмотрение каждой группы команд дает возможность сформулировать общие принципы построения языков программирования микропроцессоров.

Эти языки имеют свой алфавит. Из символов алфавита (как правило, букв латинского алфавита) составляются слова, а из слов – фразы. При составлении фраз пользуются правилами грамматики. Эти правила позволяют осуществлять контроль для выявления ошибок и помогают осуществить перевод на язык более низкого уровня, а именно на язык машинных команд.

Слова хранятся в ячейках памяти запоминающих устройств, и, соответственно, адрес ячейки памяти запоминающего устройства представляет собой имя слова и единственный признак, позволяющий отличать одно

слово от другого. Каждому слову ставится в соответствие идентификатор (буква, последовательность букв и цифр, начинающаяся с буквы), который служит постоянным именем данного слова от начала до конца программы.

Специальная программа, называемая транслятором, ставит в соответствие каждому идентификатору определённый адрес ячейки памяти, т.е. составляет таблицу соответствий. Эта же программа следит за тем, чтобы при перемещениях слов из одних ячеек памяти запоминающих устройств в другие в таблицу соответствий вносились необходимые коррективы.

Языком команд является язык ассемблера. Каждая команда в языке ассемблера состоит из мнемонического кода операции и идентификатора. Последний и служит основным отличием команды в языке ассемблера от команды в машинном языке. Например, если команда пересылки слова из ячейки памяти в регистр результата записывается на машинном языке, как *LDA ADR*, то на языке ассемблера она будет записана в форме *LDA FUN1*, где *FUN1* – идентификатор, представляющий собой имя слова, хранящегося в ячейке памяти с адресом *ADR*.

Прикладная программа микропроцессорной системы, написанная на языке ассемблера, содержит три типа высказываний:

- мнемокоды команд микропроцессора, которые транслируются в коды;
- директивы ассемблера, которые управляют трансляцией;
- комментарии, которые игнорируются в процессе трансляции и используются для документирования программы. Как правило, каждая строка программы является законченным высказыванием и имеет структуру: *< метка > : < операция > < операнд > ; < комментарий >*

- Поле комментария отделяется символом, например, точкой с запятой; поле метки - двоеточием. Различные ассемблеры накладывают разные ограничения на способ задания символьных имён. Так, например, для микропроцессора K580BM80 символьное имя должно начинаться с буквы и содержать не более 5 символов.

**Пример 2.** Рассмотреть реализацию программы временной задержки и программы сбора и формирования в ОЗУ данных от источника применительно к микропроцессору K580BM80.

Первая программа, реализует временную задержку (табл. 1.5 и рис. 1.19). Величина задержки определяется числом циклов (информация о числе циклов *X* заносится в регистр *B*) и временем выполнения команд цикла. В каждом цикле число *X* уменьшается на 1 и так продолжается до

тех пор, пока содержимое регистра *B* не станет равным 0, что интерпретируется программой как момент выхода из программного цикла.

Вторая программа (табл. 1.6 и рис. 1.20) представляет собой реализацию процедуры сбора и формирования в ОЗУ данных от одного источника - порта ввода с символическим адресом *IPOINT*. Здесь *BASE* представляет собой начальный адрес данных, регистр *C* используется в качестве счетчика данных, а регистровая пара *H* и *L* выполняет функцию указателя данных. Запись *ETX* представляет собой знак «конец текста».

Таблица 1.5

<i>TIME:</i>	<i>MVI</i>	<i>B, X</i>	; загрузка в регистр <i>B</i> числа <i>X</i>
<i>COUNT:</i>	<i>DCR</i>	<i>B</i>	; уменьшить на 1 состояние регистра <i>B</i>
	<i>JNZ</i>	<i>COUNT</i>	; цикл, если <i>B</i> ≠ 0
	<i>RET</i>		; возврат в основную программу
			; если <i>B</i> = 0

Таблица 1.6

	<i>LXI</i>	<i>H, BASE</i>	; указатель = <i>BASE</i>
	<i>MVI</i>	<i>C, 0</i>	; счетчик = 0
<i>SAVE:</i>	<i>IN</i>	<i>IPOINT</i>	; ввод данных
	<i>MOV</i>	<i>M, A</i>	; включение в массив
	<i>INX</i>	<i>H</i>	; указатель = указатель + 1
	<i>INR</i>	<i>C</i>	; счетчик = счетчик + 1
	<i>SUI</i>	<i>ETX</i>	; проверка терминатора
	<i>JNZ</i>	<i>SAVE</i>	; продолжение сбора
<i>DONE:</i>	...	...	

Директивы ассемблера или псевдокоманды не связаны с действиями над данными. Они сообщают сведения, используемые в процессе трансляции программы на язык машинных кодов. Для ассемблера микропроцессора K580BM80 имеется 6 директив:

- *ORG* сообщает с какого адреса элементов памяти запоминающих устройств следует размещать команды программы;
- *EQU* используется для задания значений переменных с определенными именами (например, запись *TIME: EQU 56* означает что переменной с именем *TIME* присваивается значение 56);

- *DB, DW, DS* задает резервирование для массива данных соответственно ячейки памяти (байта) ЗУ, двух ячеек памяти (два байта) ЗУ или нескольких ячеек памяти ЗУ;

- *END* означает конец программы.

Язык ассемблер позволяет относительно быстро создавать высокоэффективные программы, но затрудняет анализ программы. Кроме того, не удастся стандартизовать символы, так как каждый тип микропроцессора имеет свой набор мнемонических кодов команд и словарь директив.

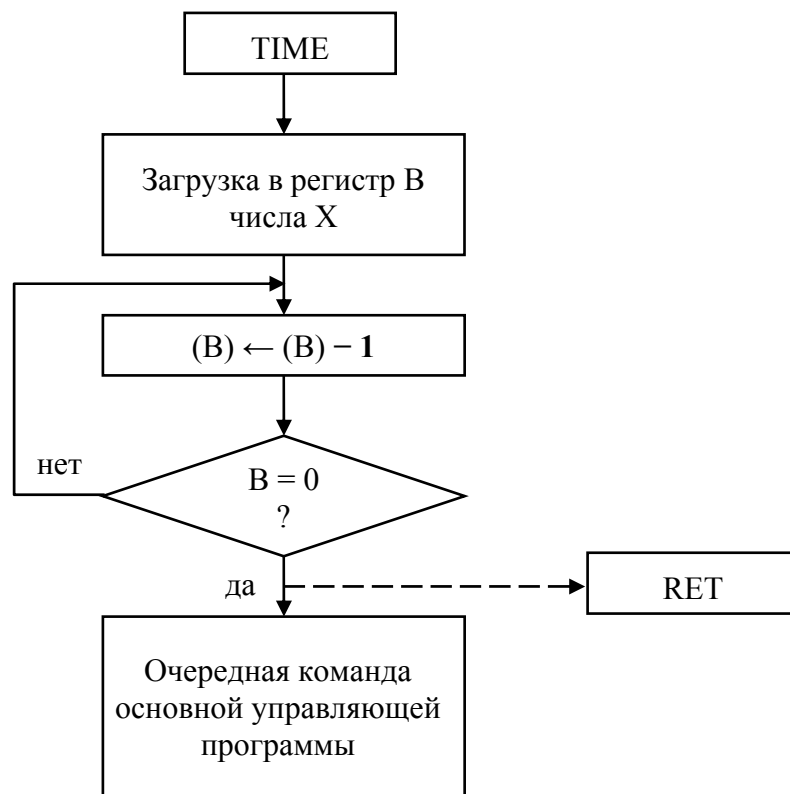


Рис. 1.19 Алгоритм временной задержки

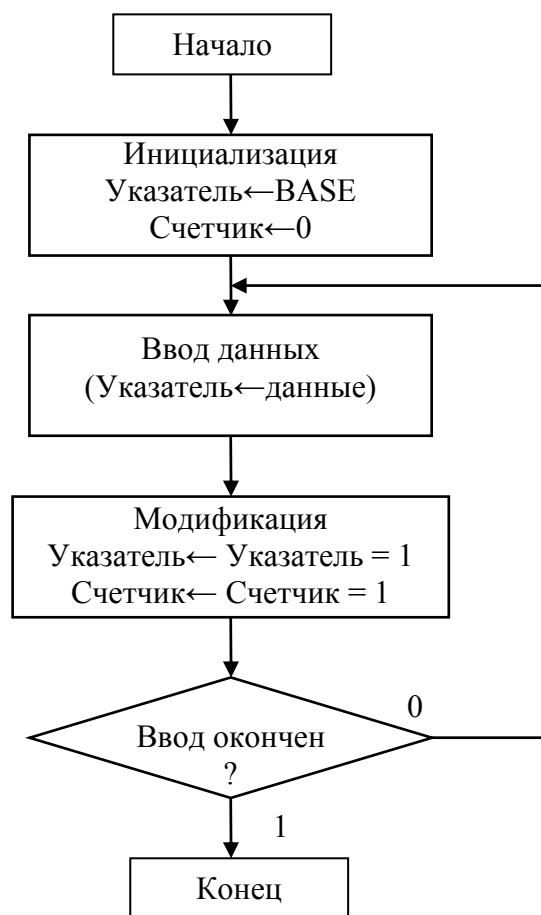


Рис. 1.20 Процедура сбора данных

В языках высокого уровня используются идентификаторы метки и операторы. Основными грамматическими конструкциями являются выражения, операторы, описания и блоки. Выражение эквивалентно некоторой последовательности команд, не содержащей команд ввода-вывода, управления и специальных команд. Выражения записываются в форме последовательностей идентификаторов, связанных между собой символами, обозначающими операции. В состав выражения обычно входит один из идентификаторов, которому присваивается результат выполненной операции.

Несомненным преимуществом языков высокого уровня является их независимость от типа используемого микропроцессора. Однако в этом кроются определенные трудности, поскольку программирование с использованием алгоритмических языков, не дает возможность учесть особенности конкретных алгоритмов. Программа становится малоэффективной, поскольку увеличивается объём памяти, занимаемой программой, полученной в результате трансляции с языка высокого уровня.

## **Глава 2. Структура и характеристики универсальных микропроцессоров**

В состав 16-разрядных и 32-разрядных микропроцессоров [8, 9, 11, 12] входят тактовые генераторы, таймеры, контроллеры прерываний и прямого доступа к запоминающим устройствам и др. Характерным для этих микропроцессоров является увеличенный набор операций, выполняемых аппаратным способом с большим числом формируемых признаков результата и расширенная емкость с соответственно увеличенной разрядностью адресной шины.

Повышение производительности микропроцессоров происходит за счет увеличения частоты работы тактовых генераторов и использования структурных решений, обеспечивающих совмещение во времени разных стадий выполнения команд, т.е. реализации конвейерного принципа обработки команд. Следует отметить, что в 32-разрядных микропроцессорах используются более совершенные формы реализации конвейерного метода выполнения команд, при которых несколько внутренних устройств работают параллельно, производя одновременно обработку нескольких последовательных команд программы.

### **2.1. Шестнадцатиразрядные микропроцессоры**

#### **2.1.1. Микропроцессор K1810BM86**

Структура 16-разрядного микропроцессора K1810BM86 [9, 11] представлена на рис. 2.1. Он является аналогом микропроцессора 8086, который, в свою очередь, оказался «прародителем» таких распространенных и широко применяемых в *IBM PC* 16- и 32-разрядных микропроцессоров, как 80286, 80386, 80486, *PENTIUM* и др. Микропроцессор 8086, выпущенный фирмой *Intel* в 1978 году, относится к первому поколению 16-битных микропроцессоров, который выполняет 8 и 16-битные логические и арифметические операции, включая умножение и деление, операции со строками и операции ввода-вывода. Микропроцессор имеет 20-разрядную шину адреса и 16-разрядную шину данных [9]. Микропроцессор поддерживает аппаратные и программные прерывания, допускает использование шин совместно с другими микропроцессорами или контроллерами (например, контроллером прямого доступа к памяти). Предусматривается совместная работа с математическим сопроцессором, существенно повышающим производительность вычислений.



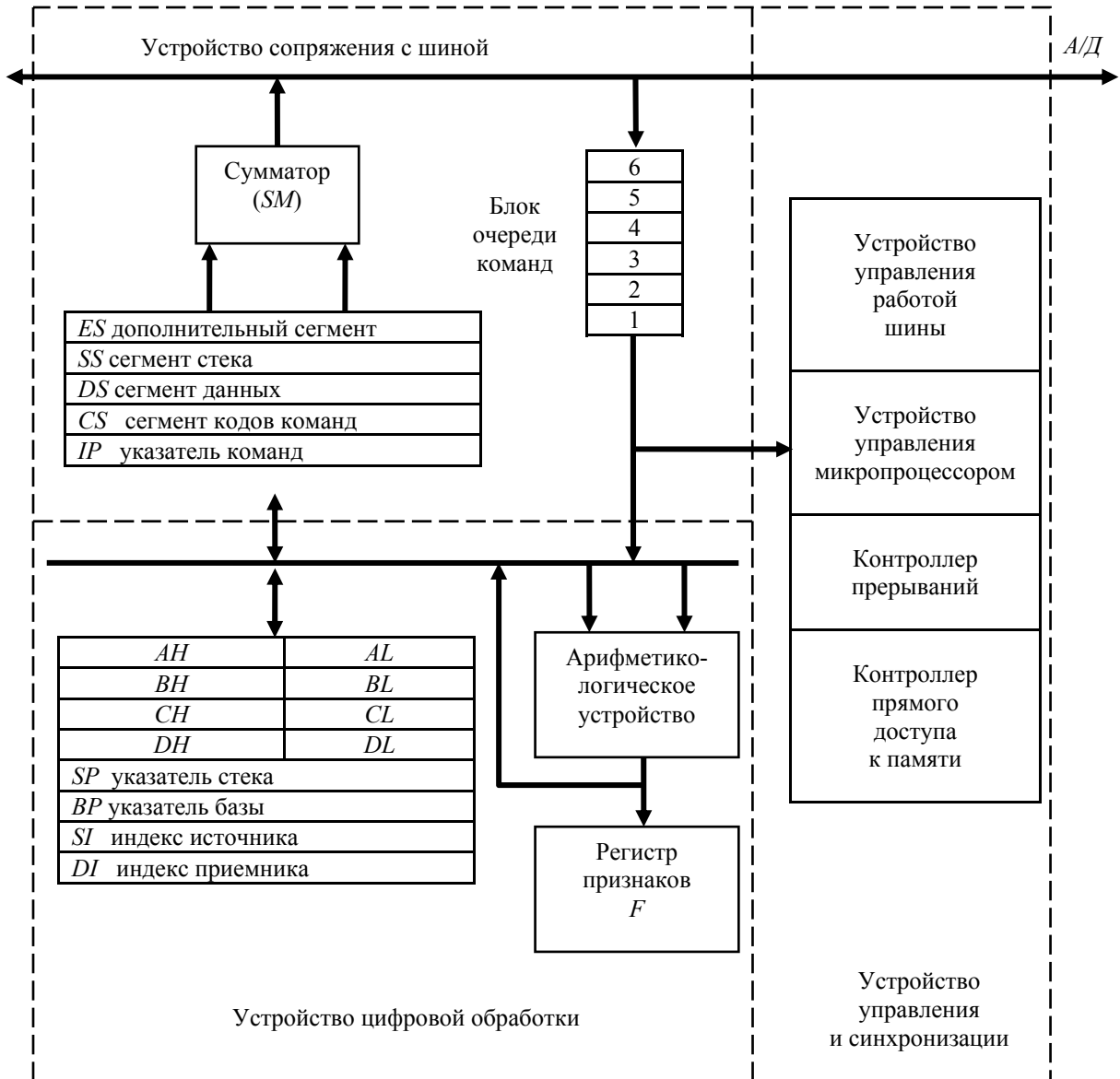


Рис. 2.1 Структура микропроцессора K1810VM86

В микропроцессоре K1810VM86 применена конвейерная архитектура, позволяющая выполнять выборку кодов программы из запоминающих устройств и их декодирование во время выполнения внутренних операций. Конвейер повышает производительность микропроцессора за счет сокращения времени простоя его операционных узлов. Конвейер микропроцессора имеет 6-байтную внутреннюю очередь команд. Блок предварительной выборки при наличии двух свободных байт в очереди старается ее заполнить в то время, когда внешняя шина микропроцессора не занята операциями обмена. Очередь обнуляется при выполнении любой команды передачи управления, даже при переходе на следующий адрес.

Микропроцессор имеет 14 разрядностью 16 бит. Операнды могут иметь разрядность 8 или 16 бит и представлять двоичные и двоично-десятичные числа. Система команд имеет 24 режима адресации операндов. Среднее время выполнения команды занимает 12 тактов синхронизации, а один цикл обмена по внешней шине занимает 4 такта (без тактов ожидания).

В структуре микропроцессора (рис.2.1) можно выделить три составные части: устройство цифровой обработки, выполняющее обработку данных; устройство сопряжения с шиной, с помощью которого обеспечивается адресация и выборка команд, адресация, прием и передача данных; устройство управления и синхронизации, формирующее на основе кодов команд и внешних сигналов сигналы управления.

В составе устройства цифровой обработки имеется 16-разрядное арифметико-логическое устройство, способное выполнять арифметические (включая деление и умножение) и логические операции, 16-разрядный регистр признаков  $F$  и блок регистров общего назначения.

Структура регистров микропроцессора приведена на рис.2.2. Блок регистров общего назначения содержит восемь 16-разрядных регистров. Четыре из них –  $AX$ ,  $BX$ ,  $CX$  и  $DX$ , используются для хранения старших  $H$  и младших  $L$  8-разрядных чисел. Регистр  $AX$  выполняет функцию аккумулятора. Регистры  $BX$ ,  $CX$ ,  $DX$  используются в качестве регистров общего назначения. При выполнении некоторых операций на них возлагаются специальные функции: регистр  $BX$  может быть источником базового адреса; регистр  $CX$  счетчиком; регистр  $DX$  – источником данных в командах умножения и деления или источником адреса в командах ввода-вывода.

Другая группа из четырех 16-разрядных регистров предназначена для хранения кодов, используемых в процессе адресации данных и команд при сегментной организации запоминающего устройства. В нее входят два регистра – указателя: указатель базы  $BP$  и указатель стека и два индексных регистра: источника  $SI$  и приемника –  $DI$ . Эти регистры участвуют при выполнении арифметических и логических операций.

Особое место занимают 16-разрядные регистры сегментов  $CS$ ,  $DS$ ,  $ES$  и  $SS$ . Содержимое этих регистров не может быть модифицировано (увеличено или уменьшено) командами, кроме команд их загрузки. Регистр  $CS$  предназначен для хранения начального адреса сегмента команд (программы). Регистр  $DS$  используется для хранения начального адреса сегмента данных, а регистр  $SS$  – начального адреса сегмента стека. Регистр  $ES$  пред-

назначен для хранения начального адреса сегмента промежуточных (вспомогательных) данных.

Адрес текущей команды хранится в 16-битном указателе команд *IP*. Регистр признаков (рис.2.2) хранит признаки результатов выполнения арифметических и логических операций и управляющие биты. Назначение ячеек регистра следующее:

- *CF (Carry Flag)* – перенос старшего бита в арифметических операциях;
- *PF (Parity Flag)* – паритет, который устанавливается при четном числе единиц в результате;
- *AF (Auxiliary Flag)* – дополнительный перенос в тетраде для десятичной арифметики;
- *ZF (Zero Flag)* – нулевой результат;
- *SF (Sign Flag)* – признак, который указывает на значение старшего бита результата;
- *TF (Trap Flag)* – трассировка (пошаговый режим) и при ее установке после выполнения каждой команды автоматически вызывается внутреннее прерывание (*INT 1*);
- *IF (Interrupt-enable Flag)* – управление прерываниями или признак прерывания, определяющий разрешение обработки запросов прерываний (маскирования) и при единичном значении разрешается выполнение маскируемых аппаратных прерываний;
- *DF (Direction Flag)* – управление направлением в строковых операциях, т.е. признак направления сканирования массива в операциях с цепочками и при единичном значении в этой ячейке индексные регистры, участвующие в строковых операциях, автоматически декрементируются на количество байт операнда, а при нулевом значении – инкрементируются;
- *OF (Overflow Flag)* – переполнение, т.е. признак переполнения разрядной сетки, отведенной для результата, который устанавливается, если результат арифметической операции не умещается в операнде назначения.

Младший байт регистра признаков полностью соответствует регистру признаков микропроцессора K580BM80. В этом микропроцессоре, в тех же позициях, записываются признаки переноса *CF*, признаки нулевого значения *ZF*, отрицательного значения результата *SF*, четности *PF* и вспомогательного переноса *AF*.

## Регистры данных

	7 0	7 0
<i>AX</i>	<i>AH</i>	<i>AL</i>
<i>BX</i>	<i>BH</i>	<i>BL</i>
<i>CX</i>	<i>CH</i>	<i>CL</i>
<i>DX</i>	<i>DH</i>	<i>DL</i>

## Регистры указателей и индексов

	15 0
<i>SP</i>	Указатель стека ( <i>Stack Pointer</i> )
<i>BP</i>	Указатель базы ( <i>Base Pointer</i> )
<i>SI</i>	Индекс источника ( <i>Source Index</i> )
<i>DI</i>	Индекс приемника ( <i>Destination Index</i> )

## Регистры сегментов

	15 0
<i>CS</i>	Сегмент кодов команд ( <i>Code segment</i> )
<i>DS</i>	Сегмент данных ( <i>Data segment</i> )
<i>SS</i>	Сегмент стека ( <i>Stack segment</i> )
<i>ES</i>	Дополнительный сегмент данных ( <i>Extra segment</i> )

## Указатель команд

	15	0
<i>IP</i>	Указатель команд ( <i>Instruction Pointer</i> )	

## Регистр признаков

<i>F</i>	<i>OF</i>	<i>DF</i>	<i>IF</i>	<i>TF</i>	<i>SF</i>	<i>ZF</i>	0	<i>AF</i>	0	<i>PF</i>	1	<i>CF</i>
----------	-----------	-----------	-----------	-----------	-----------	-----------	---	-----------	---	-----------	---	-----------

Рис. 2.2 Регистры микропроцессора K1810BM86

В составе устройства сопряжения с шиной (рис.2.1) имеются два блока: блок очереди команд и блок сегментных регистров с сумматором (*SM*). Блок очереди команд состоит из шести 8-разрядных регистров и обеспечивает хранение очередных байтов команд, выполняемых микропроцессором. Блок очереди команд представляет собой стек, в котором число, запи-

санное первым, извлекается так же первым. При считывании очередных двух байтов команд производится перемещение оставшихся байтов с автоматической выборкой из запоминающего устройства новых двух байтов.

В микропроцессоре реализуется адресация запоминающих устройств емкостью 1 Мбайт, что требует применения 20-разрядного адреса. При этом используется адресация, которая предусматривает применение в командах 16-разрядных кодов (логических адресов) с последующим формированием на их основе 20-разрядных физических (исполнительных) адресов. Это обеспечивается разделением адресного пространства на сегменты от 16 байт до 64 Кбайт каждый. Начальные адреса сегментов предусматривают нулевые значения 4 младших разрядов в 20-разрядном исполнительном адресе. Эти начальные адреса задаются с помощью 16-разрядного кода при наличии нулевых значений 4-х младших разрядов.

Память для микропроцессора K1810BM86 представляется в виде линейной последовательности байт. Для обращения к ЗУ микропроцессор формирует шинные сигналы MEMWR# (Memory write) и MEMRD# (Memory read) для операций записи и считывания соответственно. Физический адрес ячейки памяти, поступающий на шину адреса разрядностью 20 бит, состоит из двух 16-битных частей: адреса сегмента Seg, являющегося адресом, с которого начинается сегмент, и этот адрес называется базовым адресом сегмента; исполнительного адреса EA (Executive address), который формируется путем смещения относительно начала сегмента (рис.2.3). Сдвиг адреса сегмента на 4 бита эквивалентен его умножению на 16, следовательно, физический адрес PA равен сумме:  $16 \text{ Seg} + \text{EA}$ . Адрес сегмента Seg является содержимым одного из регистров: CS, DS, SS или ES. Исполнительный 16-разрядный адрес, также называемый эффективным адресом, может быть постоянной величиной, содержимым регистров, содержимым ячейки памяти или суммой нескольких величин (например, двух регистров и постоянной величины). Таким образом, физический адрес не переходит границу 64-килобайтного сегмента, на начало которого указывает используемый сегментный указатель. Это свойство обеспечивает защиту сегментов друг от друга. Однако, сегментация памяти является существенным неудобством для написания больших программных модулей. Такая организация адресного пространства, с одной стороны, является удобной с точки зрения использования 16-разрядных форматов сегментных регистров и адресов-смещений, а с другой стороны, открывает широкие возможности для использования разнообразных сегментов емкостью

от 16 байт до 64 Кбайт с динамическим перемещением сегментов, необходимым для многопрограммных режимов работы. Для этого требуется предварительное формирование определенных кодов сегментных регистров перед началом адресации ячеек соответствующих сегментов.

С сегментацией связаны понятия ближнего и дальнего адреса (вызова, перехода). При ближнем вызове доступ к требуемой ячейке запоминающего устройства осуществляется только указанием смещения, а адрес сегмента определяется текущим содержимым соответствующего регистра сегмента. Разновидностью ближнего перехода, по которому доступен любой адрес сегмента, является короткий переход, в котором смещение указывается не двумя, а одним байтом, что используется в командах условных переходов. Короткий переход позволяет достигать смещения от -128 байт до +127 байт относительно текущего адреса в пределах указанного сегмента. При дальнем обращении указывается полный адрес, содержащий 16-битное значение сегмента, загружаемое в соответствующий сегментный регистр и 16-битное смещение.

Процессор может обращаться как к одному байту запоминающего устройства, так и слову, состоящему из двух байт, или двойному слову (4 байта). При размещении слова в ЗУ байт с адресом, соответствующим адресу слова, содержит его младшую часть, а следующий байт содержит старшую часть. Слово может размещаться в ЗУ как по четному, так и по нечетному адресу. Двойное слово обычно используется для хранения полного адреса и в нем располагается сначала слово смещения (в порядке  $L$ ,  $H$ ), а затем сегмента в том же порядке.

Исполнительные адреса команд формируются путем сложения содержимого 16-разрядного регистра указателя команд  $IP$  с содержимым сегментного регистра  $CS$ . Регистр указателя команд  $IP$  соответствует счетчику команд микропроцессора K580BM80 и в нем формируется каждая следующая команда, которая будет выполняться после текущей. Устройство сопряжения с шиной записывает в регистр адреса команд из устройства цифровой обработки смещение следующей команды от начала текущего сегмента кода. При формировании исполнительных адресов данных, относящихся к источнику информации, в качестве смещения используется содержимое регистра  $SI$ , а относящихся к приемнику – содержимое регистра  $DI$ . Регистры  $SP$  и  $BP$  задают смещение при формировании адресов сегмента стека.

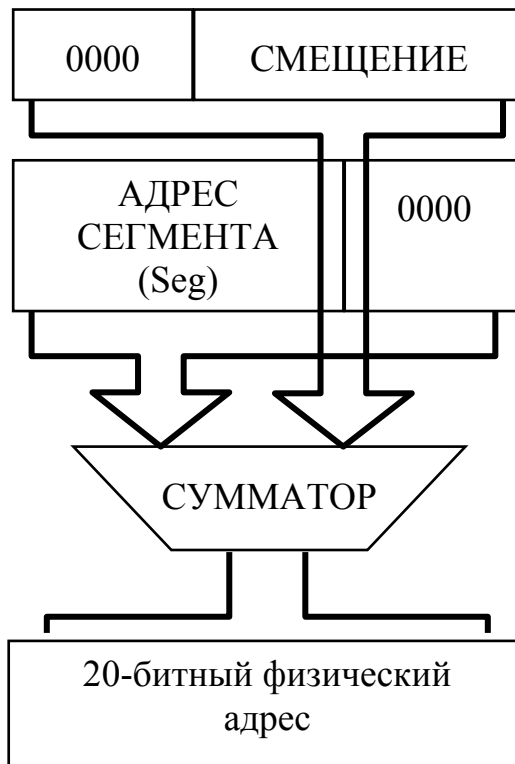


Рис. 2.3 Формирование физического адреса памяти микропроцессором K1810BM86

В устройстве управления и синхронизации формируются сигналы для организации взаимодействия между узлами микропроцессора, и сигналы управления для организации асинхронного обмена, прерываний, прямого доступа к памяти.

Для передачи адресов и данных используется 16-разрядная двунаправленная шина *А/Д* (рис.2.1), работающая в мультиплексном режиме. Четыре старших разряда исполнительного адреса (20-разрядного) передаются по дополнительным 4 линиям, входящим в шину управления. Эти линии могут использоваться для передачи содержимого регистра прерываний. Синхронизация работы микропроцессора осуществляется последовательностью синхроимпульсов с периодом следования импульсов, равным 200нС. Эта последовательность управляет выполнением машинных тактов. Каждая команда осуществляется за несколько машинных циклов *М*, состоящих из 4 машинных тактов *Т1 – Т4*. В такте *Т1* выдается информация об адресах данных на шину *А/Д*. Переключение направления передачи данных по шине *А/Д* происходит в такте *Т2*. В тактах *Т3* и *Т4* осуществляется сама передача данных. Кроме того, возможно наличие холостого такта *Т5* и такта ожидания *ТW*, который располагается между тактами *Т3* и *Т4*. Выполнение каждого машинного цикла сопровождается передачей состояния микропроцессора, задающего тип этого цикла.

### 2.1.2. Математический сопроцессор K1810BM87

Архитектурной особенностью микропроцессора K1810BM86, как уже отмечалось выше, является способность координировать взаимодействие нескольких микропроцессоров таким образом, что одни являются независимыми, которые выполняют команды, а другие являются вспомогательными (сопроцессорами). Сопроцессор просматривает команды, выбираемые центральным микропроцессором, отбирает и выполняет некоторые из них.

Сопроцессор K1810BM87, предназначен для расширения вычислительных возможностей центрального микропроцессора [9]. Его применение к системе команд микропроцессора K1810BM86 добавляет команды для выполнения арифметических, тригонометрических, экспоненциальных и логарифмических команд. Сопроцессор поддерживает семь типов данных: 16-, 32-, 64-битные целые числа; 32-, 64-, 80-битные числа с плавающей точкой и 18-разрядные числа в двоично-десятичном формате. Сопроцессор выполняет математические операции с производительностью, превосходящей их программную эмуляцию центральным микропроцессором более, чем в 100 раз. Пара микропроцессоров K1810BM86 и K1810BM87 выглядит как единое целое, где микропроцессор K1810BM86 осуществляет общее управление процессом вычислений и обработку команд основного набора, а микропроцессор K1810BM87 исполняет только свои специфические команды.

Сопроцессор, отслеживая сигналы состояния микропроцессора K1810BM86 ( $S[0:2]$  и  $QSO, QSI$ ), вместе с ним просматривает и декодирует команды. Математический сопроцессор имеет внутреннюю очередь команд, выполнение которой происходит по тем же правилам, что и выполнение очереди в микропроцессоре K1810BM86. Любая команда, предназначенная для сопроцессора, является двухбайтной. Первый байт имеет двоичный код 11011xxx, второй байт вместе с битами xxx уточняет команду, за которой может следовать 1 или 2 байта смещения. Эти команды в микропроцессорах K1810BM86 и K1810BM87 обрабатывают одновременно. Когда команда подразумевает обмен данными с запоминающим устройством, вычисляет их адрес и выполняет шинный цикл фиктивного чтения первого слова, но игнорирует считанные данные. Если должно выполняться считывание данных из ЗУ, эти данные считываются сопроцессором, который фиксирует во внутреннем регистре значение текущего адреса. В случае, когда команда подразумевает считывание более, чем одного слова (байта) данных, сопроцессор запрашивает управление локальной шиной и,



получив его, продолжает загрузку оставшейся части операнда, последовательно увеличивая адрес. При записи в запоминающее устройство, сопроцессор во время фиктивного чтения только перехватывает значение физического адреса ячейки памяти ЗУ, а реальную запись производит, начиная с этого адреса, получив управление шиной. Завершив все циклы передачи данных, сопроцессор переходит к выполнению следующей команды. После вычислительной команды, выполняемой сопроцессором, следует команда *WAIT* по которой сопроцессор дожидается уровня логического нуля на входе *TEST#* (микропроцессор K1810BM86) и на выходе *BUSY* (сoproцессор K1810BM87). Сигнал занятости *BUSY* вырабатывается сопроцессором на время выполнения вычислительной команды, чем и обеспечивается его синхронизация. Команда *WAIT* вводится непосредственно перед тем, как потребуются результаты вычислений. В этом случае микропроцессор и сопроцессор могут некоторое время работать параллельно.

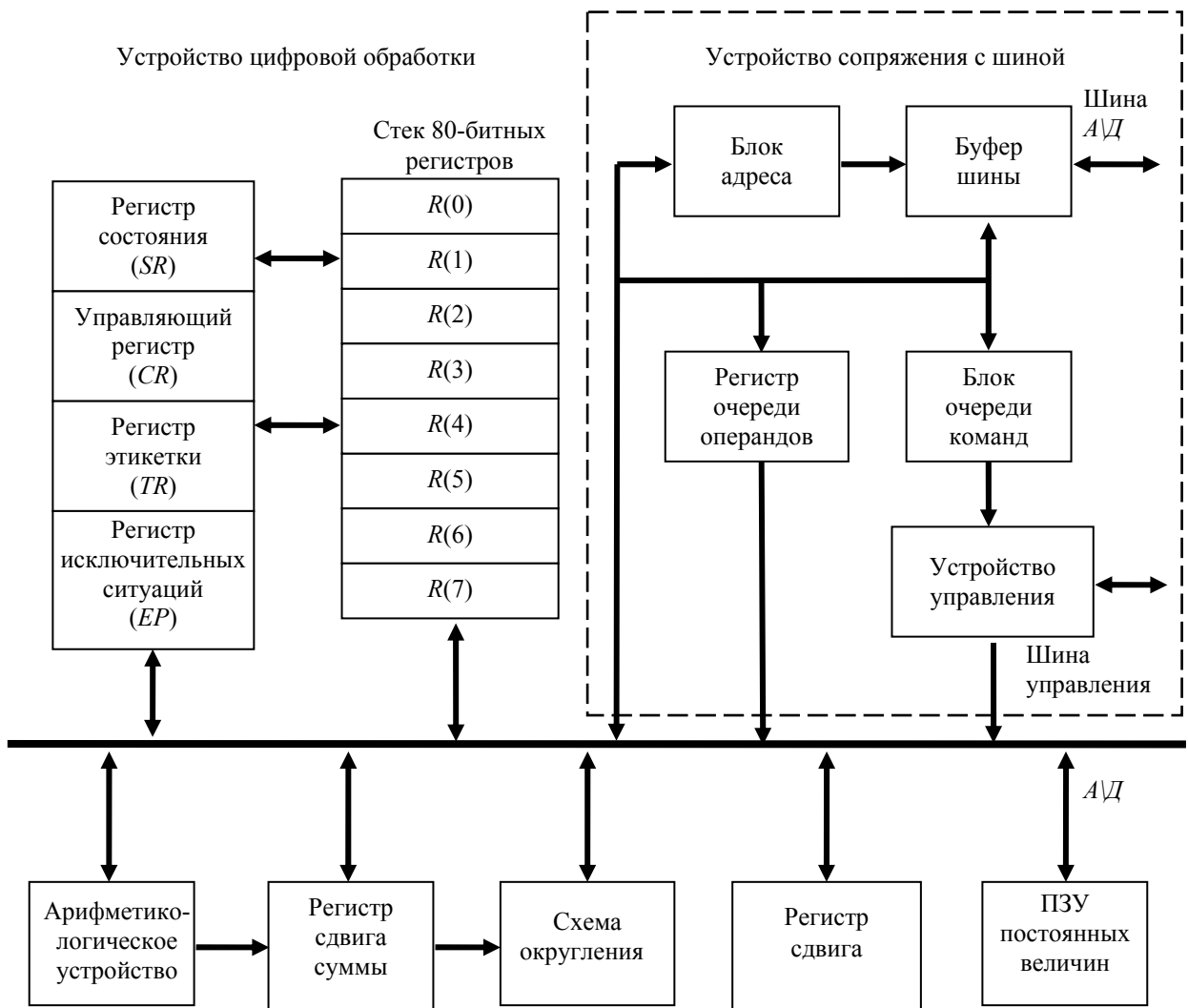


Рис. 2.4 Структура схема сопроцессора K1810BM87

Математический сопроцессор K1810BM87 (рис. 2.4) можно условно разделить на два независимо работающих устройства, позволяющих организовать конвейерную обработку принимаемых данных: устройство цифровой обработки (исполнительный блок) и устройство сопряжения с шиной. Команды и операнды по мультиплексированной шине *A/D* извлекаются из ячеек памяти запоминающего устройства и размещаются соответственно в блоке очереди команд и в регистре очереди операндов.

В устройстве цифровой обработки имеется восемь 80-разрядных регистров, выполняющих роль регистров общего назначения и организованных в стек. Формат представления данных в стеке следующий: 64 разряда – мантисса, знак числа и 15 разрядов – порядок чисел. В этом же формате производятся все внутренние вычисления. Трехразрядный указатель стека определяет один из восьми регистров. С каждым из регистров связано 2-разрядное поле признаков, которое предназначено для анализа состояния регистров. Комбинация бит 00 указывает на наличие операнда в регистре. Сочетание 01 предполагает его нулевое значение. Специальное назначение имеет комбинация 10, а сочетание 11 означает, что в регистре данные отсутствуют. Поля признаков объединены в один регистр этикетки *TR*. Регистр исключительных ситуаций (*EP*) используется для учета неиспользованных регистров стека и обозначения специальных величин (например, 0) при обработке особых ситуаций и оптимизации внутреннего микрокода.

Регистр состояния *SR* (рис.2.5) отображает выполняемые сопроцессором действия или результаты действия. Слово состояния включает в себя признак занятости, коды условий, указатель стека и признаки особых ситуаций. К этим признакам относятся: недействительная операция – *IE*; ненормализованный операнд – *DE*; деление на ноль – *ZE*; переполнение – *OE*; антипереполнение – *UE*; точность – *PE*. Единичное значение поля *B* указывает на занятость устройства цифровой обработки выполнением операции, на наличие запроса, который не был обслужен, либо на прерывание исключения.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<i>B</i>	<i>C3</i>	<i>TOP</i>			<i>C2</i>	<i>C1</i>	<i>C0</i>	<i>IR</i>	<i>X</i>	<i>PE</i>	<i>UE</i>	<i>OE</i>	<i>ZE</i>	<i>DE</i>	<i>IE</i>

Рис. 2.5. Регистр состояния сопроцессора K1810BM87

Поле *TOP* предназначено для указания вершины стека. Коды условия *C2*, *C1*, *C0* интерпретируются в зависимости от выполненной команды. Ес-

ли после выполнения команды *FSTSWAX*, пересылающей слово состояния сопроцессора в регистр данных *AX* на выполнение команды *SAHF*, коды условия переносятся в регистры признаков сопроцессора, возможно их использование в командах условных переходов. Поле *IR* используется при возникновении немаскированного исключения. Поля регистра состояний от 0 до 6 устанавливаются при возникновении соответствующих исключений (табл. 2.1).

### Стек 80-битных регистров

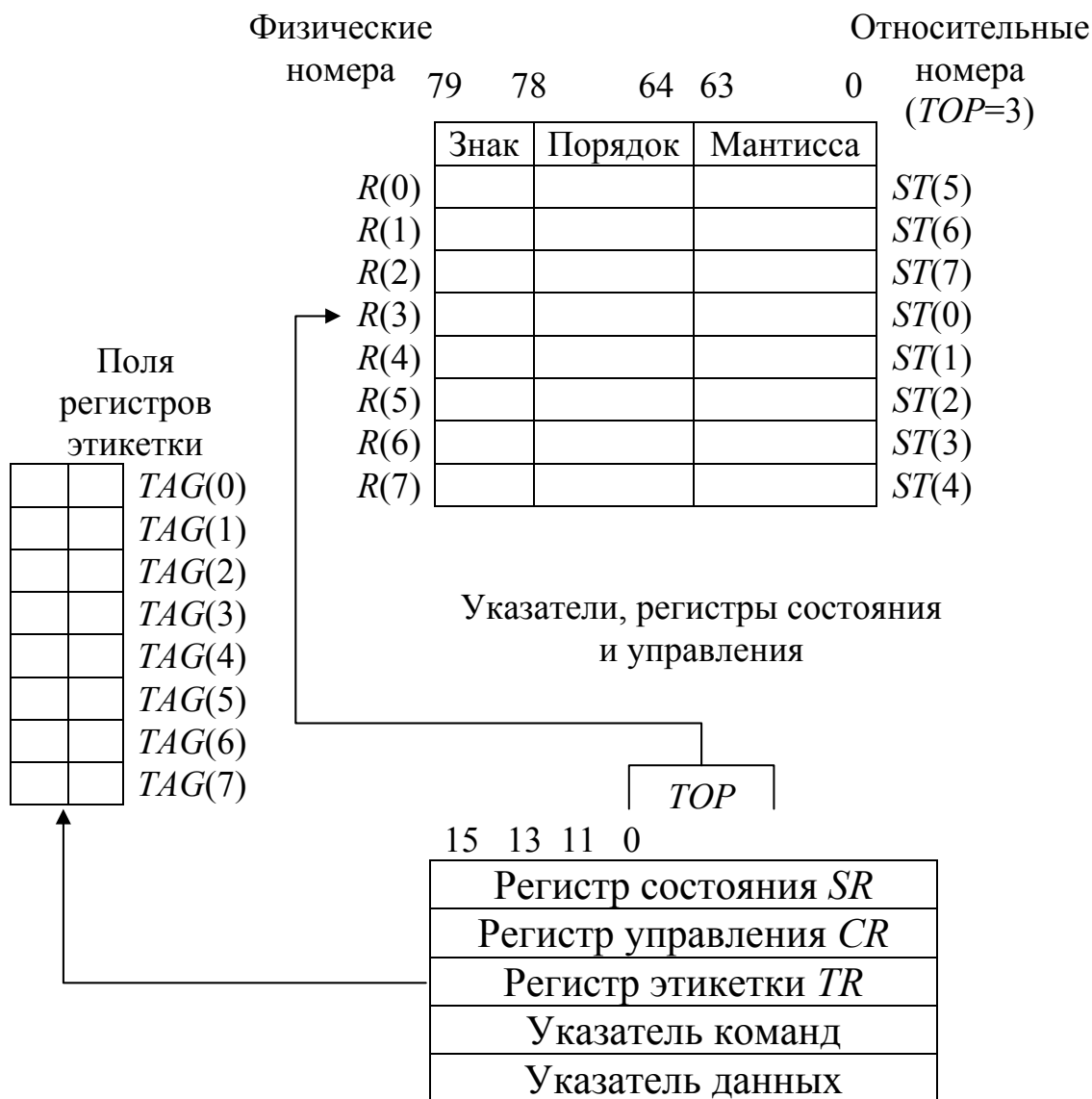


Рис. 2.6 Регистры сопроцессора K1810BM87

В стеке номер регистра, являющегося текущей вершиной стека, хранится в поле *TOP* регистра управления. Команда *PUSH* уменьшает значение поля *TOP* на 1 и помещает данные в регистр, являющийся новой вершиной стека. По команде *POP* происходит запись данных с вершины стека

в ячейку памяти запоминающего устройства и инкрементируется указатель стека. Команды адресуют регистры явно, либо неявно. Неявная адресация подразумевает операнд, находящийся в вершине стека. Явная адресация подразумевает адрес регистра относительно вершины стека.

На рис. 2.6 представлены регистры цифрового устройства обработки сопроцессора K1810BM8.

Регистр управления *CR* (рис. 2.7) служит для задания различных режимов работы сопроцессора. В регистре управления слово имеет разряды маскирования особых ситуаций и разряды управления, которые задают точность вычислений. Управление точностью происходит путем выбора одного из четырех способов округления.

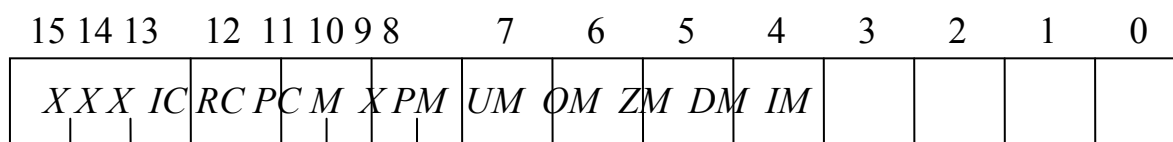


Рис. 2.7 Регистр управления сопроцессора K1810BM87

Единичные значения на полях *IM*, *DM*, *ZM*, *OM*, *UM*, *PM* маскируют (табл.2.1) отдельные исключения. Единичное значение на поле *M* маскирует прерывания. Поле *PC* (*Precision Control*) задает точность представление чисел. Код 00 соответствует одинарной точности представления чисел (24 бита); Двойная точность (53 бита) получается при значении кода 10. Код 11 обеспечивает расширенное представление чисел (64 бита) и сочетание кода 01 является резервным.

Поле *RC* (*Rounding Control*) определяет способ округления чисел. Указатели команд и данных служат для сохранения физического адреса кода выполняемой операции и физического адреса операнда.

Таблица 2.1.

Регистр состояния	Регистр управления	Исключение
<i>IE</i>	<i>IM</i>	Недействительная операция: извлечение данных из незаполненного регистра, неопределенный результат, извлечение квадратного корня из отрицательного числа, операция с нечисловыми переменными и т. п.

<i>DE</i>	<i>DM</i>	Хотя бы один из операндов является денормализованным.
<i>ZE</i>	<i>ZM</i>	Деление на нуль ненулевого операнда.
<i>OE</i>	<i>OM</i>	Переполнение.
<i>UE</i>	<i>UM</i>	Антипереполнение (исчезновение порядка).
<i>PE</i>	<i>PM</i>	Результат не может быть точно представлен в заданном формате. При этом выполняется округление.

Математический сопроцессор K1810BM87 поддерживает арифметические вычитания, умножения, деления, извлечения корня квадратного. Этот сопроцессор позволяет выполнить команды масштабирования, выделение остатка, целой части числа и пр. Кроме того, он реализует команды: передачи данных – загрузки, запоминание и обмена; управление процессом – загрузки и запоминание сигналов синхронизации и управления, установку и сброс регистра прерываний, сброс особых ситуаций; сравнения; трансцендентных операций; загрузки постоянных величин.

Сопроцессор определяет шесть различных условий исключения, возникающих в процессе выполнения команд (табл. 2.1). Любое из условий вызывает сигнал аппаратного прерывания *INT*, если оно не замаскировано и прерывание разрешено. При запрещении прерывания и возникновении исключения, продолжается выполнение команды, независимо от того, замаскирован данный класс исключений или нет. Когда прерывание разрешено возникновение замаскированного исключения только фиксируется в регистре состояния *SP*, не вызывая прерывания, а сопроцессор выполняет соответствующую данному типу внутреннюю процедуру обработки исключения для обеспечения возможности продолжения работы. Незамаскированное исключение вызывает прерывание и обрабатывается внешней командой.

### **2.1.3. Сопроцессор ввода-вывода K1810BM89**

Сопроцессор ввода-вывода K1810BM89, структурная схема которого представлена на рис.2.8, сочетает в себе свойства универсального контроллера прямого доступа к памяти и специализированного микропроцессора, осуществляющего различные преобразования данных во время операции пересылок.

Сопроцессор имеет два идентичных и независимых канала, обеспечивающих передачу данных в режиме прямого доступа к памяти со скоро-

стью 1,25 Мбайт/с. Одновременно с передачей осуществляются операции по преобразованию данных: маскированное сравнение, трансляция и преобразование кодов из 16-разрядных в 8-разрядные и наоборот. Обмен данными с системной памятью объемом до 1 Мбайта выполняется с помощью четырех 20-разрядных регистров-указателей.

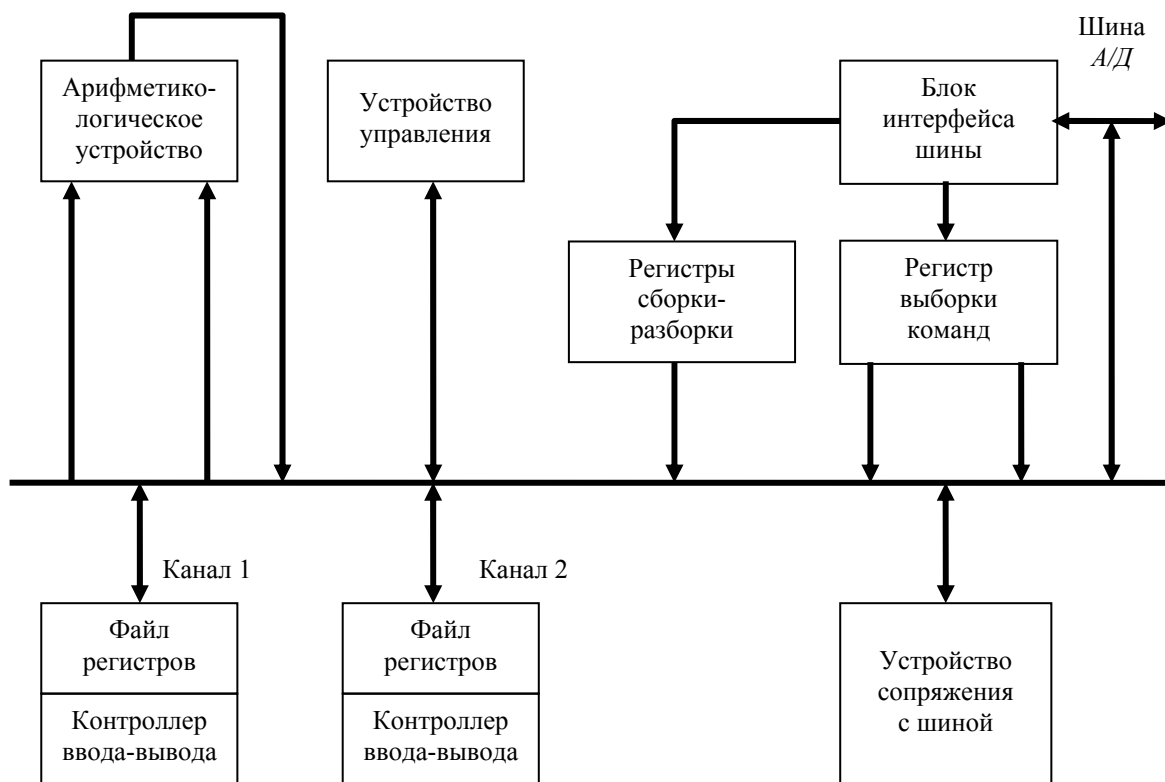


Рис. 2.8 Структурная схема сопроцессора K1810BM89

Каждый канал имеет набор (файл) регистров и контроллер ввода-вывода. Оба канала могут работать одновременно, выполняя каналные программы или осуществляя передачу с прямым доступом при мультиплексировании времени доступа и использовании внешней шины. Доступ к памяти во время одного цикла обращения к внешней шине имеет только один канал.

Устройство управления обеспечивает начальную инициализацию сопроцессора, определяя приоритеты каналов. Это устройство осуществляет синхронизацию циклов выполнения команд, пересылки по прямому доступу к памяти, ответы на запросы готовности канала.

Регистры сборки-разборки служат для выполнения пересылки данных за минимально возможное число циклов работы с шиной, когда данные пересылаются между шинами различной разрядности. Так при пересылке данных в режиме доступа к памяти от 8-разрядного устройства ввода-

вывода в 16-разрядную память, сопроцессор выполняет 2 цикла работы с шиной. При этом за каждый цикл осуществляется прием 8-ми разрядов, собирается 16-разрядное слово, а затем за один цикл это слово пересылается в запоминающее устройство. Регистр выборки команд служит для хранения команд, выбираемых из запоминающего устройства для работающего канала. Команды запоминаются в регистре в виде очереди из байтов команд.

При этом каждый канал имеет свою очередь, и работа одного канала не влияет на очередь другого канала. Устройство сопряжения с шиной управляет всеми циклами работы с шиной А/Д.

#### **2.1.4 Микропроцессор 80286**

Микропроцессор 80286 [9] имеет специальные средства для работы в многопроцессорных. Он содержит механизм управления адресацией запоминающих устройств, который обеспечивает четырехуровневую иерархическую систему привилегий и поддержку виртуальной памяти. В микропроцессоре 80286 имеются специальные средства, предназначенные для поддержки механизма переключения процедур решения задач. Он имеет расширенную систему команд, которая, кроме команд управления защитой, включает все команды микропроцессора K1810BM86 и несколько новых команд общего назначения. Микропроцессор может работать в двух режимах:

- Режим адресации, полностью совместимый с режимом работы микропроцессора K1810BM86. В этом режиме возможна адресация до 1 Мбайт физической памяти.
- Защищенный режим или режим виртуальной адресации. В этом режиме микропроцессор позволяет адресовать до 16 Мбайта физической памяти, через которую могут отображаться до 1 Гбайта виртуальной памяти для каждой задачи. Система команд в этом режиме также включает набор команд микропроцессора K1810BM86, расширенный для обеспечения аппаратной реализации функций супервизора многозадачной операционной системы (ОС) и виртуальной памяти. Переключение в защищенный режим осуществляется одной командой с предварительно подготовленными таблицами дескрипторов. Обратное переключение в реальный режим возможно только через аппаратный сброс микропроцессора, что требует значительных затрат времени.

Защищенный режим предназначен для обеспечения независимости выполнения нескольких задач, что подразумевает защиту ресурсов одной

задачи от возможного воздействия другой задачи. Основным защищаемым ресурсом является запоминающее устройство, в котором хранятся коды, данные и различные системные таблицы (например, таблица прерываний). Защищать требуется и совместно используемую аппаратуру, обращение к которой обычно происходит через операции ввода-вывода и прерывания.

Защита запоминающего устройства основана на использовании сегментации памяти, понимая под сегментом блок адресного пространства памяти определенного назначения. К элементам сегмента возможно обращение с помощью различных команд микропроцессора, использующих разные режимы адресации для формирования адреса в пределах сегмента. Сегменты памяти выделяются для задач операционной системой. В режиме реальной адресации для любой задачи можно переопределить значения сегментных регистров, задающих положение сегмента в пространстве памяти. В защищенном режиме сегменты также распределяются операционной системой, но прикладная программа использует только разрешенные для нее сегменты памяти, выбирая их с помощью селекторов из предварительно сформированных таблиц дескрипторов сегментов. Селекторы представляют собой 16-битные указатели, загружаемые в сегментные регистры микропроцессора.

Дескрипторы представляют собой структуры данных, используемые для определения свойств программных элементов (сегментов, вентилях и таблиц). Дескрипторы имеют 8-байтный формат. Их назначение определяется полями управления доступом. Дескрипторы 16- и 32-разрядных микропроцессоров отличаются разрядностью поля базового адреса (24 и 32 бит) и трактовкой поля размера сегмента, которое должно обеспечивать размер сегмента до 64 Кбит или 4 Гбит соответственно. Два старших байта у дескрипторов микропроцессора 80286 нулевые, что позволяет их отличать и корректно использовать, выполняя 16-битные приложения защищенного режима на 32-битных микропроцессорах.

Дескриптор определяет положение программного элемента в памяти, размер занимаемой им области, его назначение и характеристики защиты. Защита памяти с помощью сегментации не позволяет:

1. Использовать сегменты не по назначению (например, трактовать область данных как коды команд).
2. Нарушать права доступа (модифицировать сегмент, предназначенный для чтения, для обращения к сегменту).
3. Адресоваться к элементам, выходящим за границы сегмента.



4. Изменять содержимое таблиц дескрипторов (параметров сегментов) без определенных команд.

Защищенный режим предоставляет средства переключения задач. Состояние решения каждой задачи может быть сохранено в специальном сегменте состояния задачи (*TSS*), на который указывает селектор. При переключении задач достаточно загрузить новый селектор в соответствующий регистр и состояние решения предыдущей задачи автоматически сохранится в сегменте состояния задачи. В микропроцессор загрузится состояние новой задачи и начнется ее выполнение.

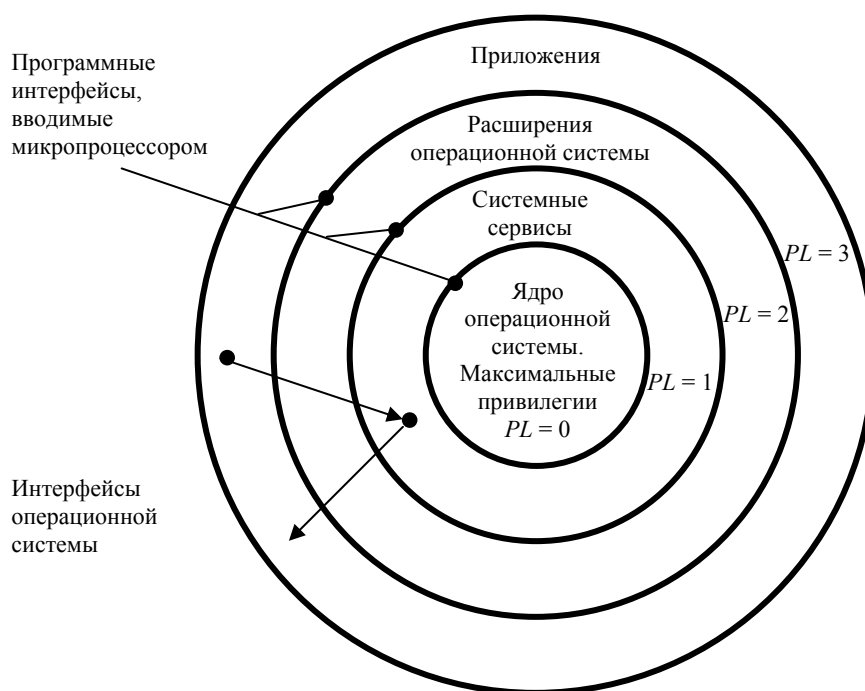


Рис. 2.9. Уровни привилегий

Четырехуровневая иерархическая система привилегий предназначена для управления использованием привилегированных команд и доступом к дескрипторам. Уровни привилегий нумеруются от  $PL = 0$  до  $PL = 3$ , нулевой уровень соответствует максимальным возможностям доступа и отводится для ядра операционной системы. Уровень  $PL = 3$  имеет ограниченные права и обычно предоставляется прикладным задачам. Система защиты изображается в виде концентрических колец, соответствующих уровням привилегий (рис.2.9), а сами уровни привилегий называются кольцами защиты. Сервисы, предоставляемые задачам, могут находиться в разных кольцах защиты. Передача управления между задачами контролируется вентилями, проверяющими правила использования уровней привилегий.

Через вентили задачи могут получить доступ только к разрешенным

им сервисам других сегментов.

Уровни привилегий относятся к дескрипторам, селекторам и задачам. Кроме того, в регистре признаков имеется поле привилегий ввода-вывода, с помощью которого обеспечивается управление доступом к командам ввода-вывода и управление признаком прерываний.

Дескрипторы и привилегии являются основой системы защиты. Дескрипторы определяют структуры элементов без которых невозможно их использование, а привилегии определяют возможность доступа к дескрипторам и выполнения привилегированных команд.

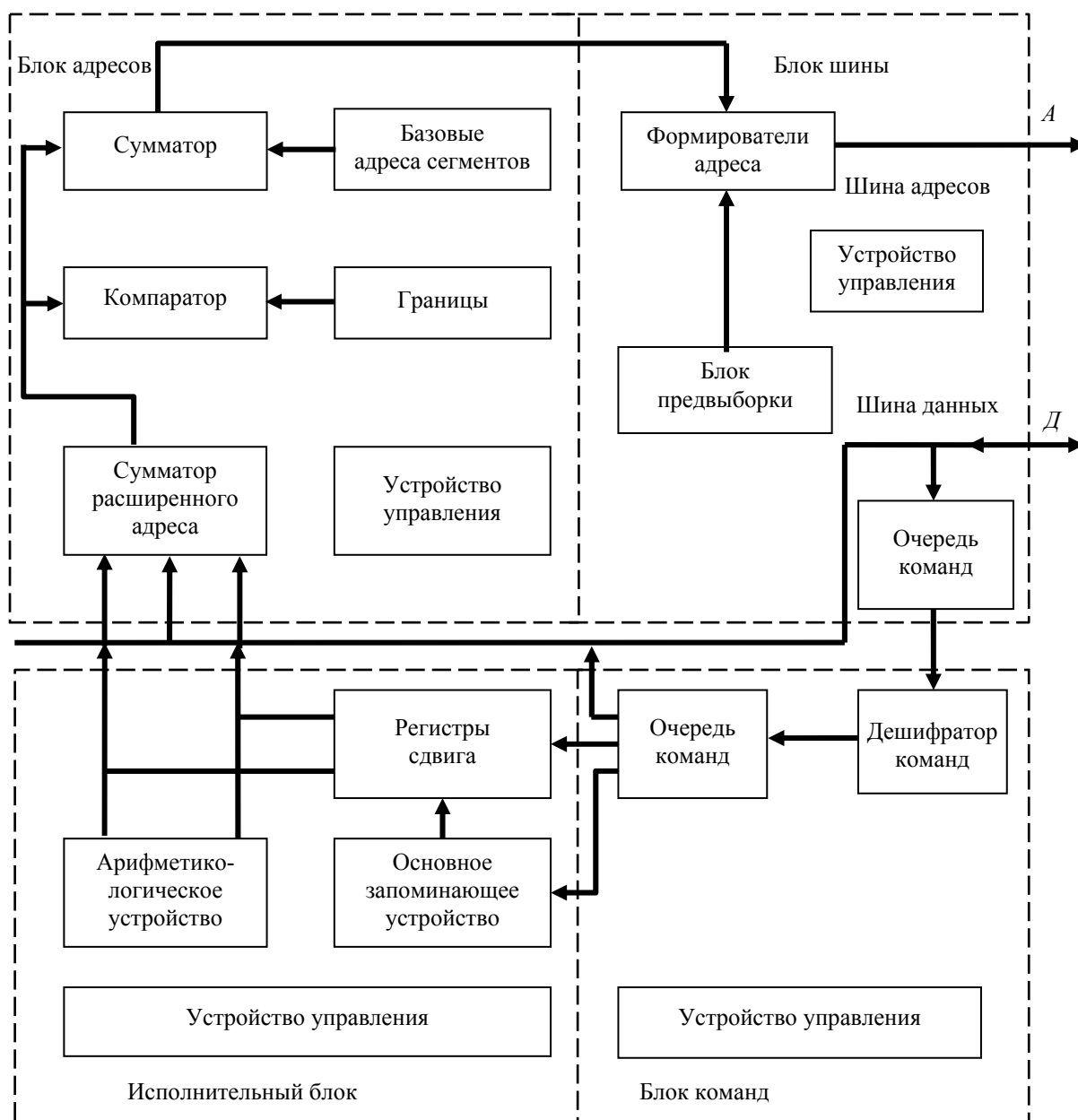


Рис. 2.10. Обобщенная структурная схема микропроцессора 80286

Механизм виртуальной памяти, используемый в защищенном режиме, позволяет для решения любой задачи использовать логическое адресное пространство. Для этого каждый сегмент (страница) в своем дескрипторе имеет специальное поле, в котором указывается присутствие данного сегмента в оперативной памяти в текущий момент времени. Неиспользуемый сегмент может быть выгружен из оперативной во внешнюю память, о чем делается пометка в его дескрипторе. На освободившееся место из внешнего запоминающего устройства может восстанавливаться содержимое другого сегмента и в его дескрипторе делается пометка о присутствии. При обращении задачи к отсутствующему сегменту микропроцессор вырабатывает соответствующее исключение, при обработке которого и происходит управление виртуальной памятью в операционной системе. После восстановления страницы выполнение задачи продолжается.

Микропроцессор предоставляет только необходимые аппаратные средства поддержки защиты и виртуальной памяти, а их реальное использование и устойчивость работы программ, а также самой операционной системы защищенного режима, зависят от корректности ее построения.

По составу и назначению в режиме реальной адресации регистры микропроцессора 80286 в основном совпадают с регистрами микропроцессора K1810BM86 (рис. 2.2). Изменения касаются назначения полей регистра признаков и использования регистров сегментов в защищенном режиме. Микропроцессор 80286 имеет 16-битную шину данных и 6 байтную очередь команд, а также возможность использования высокопроизводительного математического сопроцессора 80282.

В микропроцессоре 80286 (рис.2.10) используется конвейерный принцип организации выполнения команд с 4-мя уровнями конвейеризации, реализованными в 4 отдельных логических блоках: шины, адресов, команд и исполнительном. Эти блоки работают одновременно так, что циклы обращения к запоминающим устройствам, циклы вычисления адресов и контроля защиты, декодирования и выполнения команд могут совмещаться.

Блок шины передает информацию из своей очереди в блок команд. Блок команд декодирует и преобразует формат полных команд и помещает их в очередь команд, ожидающих выполнения.

Исполнительный блок содержит арифметико-логическое устройство, рабочие регистры сдвига и микропрограммное ПЗУ, которое определяет последовательность внутренних микрокоманд. Когда текущая команда близка к завершению, в микропрограммном ПЗУ формируется сигнал, по

которому исполнительный блок принимает следующий адрес из очереди команд. Это позволяет обеспечить постоянную загруженность исполнительного блока.

Блок адресов производит преобразование адресов и контролирует права доступа, обеспечивая защиту запоминающих устройств. В нем имеется быстродействующее статическое ОЗУ, содержащее как базовые адреса, так и предельные граничные значения и права доступа для всех сегментов виртуальной памяти, выбранных в данный момент для использования выполняющейся задачей. Наличие быстродействующего статического ОЗУ сводит к минимуму необходимость в считывании указанной информации из основного запоминающего устройства и позволяет блоку адресов выполнять свою функцию за один период тактовой частоты.

Раздельные шины адресов и данных в блоке шины (рис.2.10) способствуют обеспечению более высокой производительности микропроцессора.

По набору программно доступных регистров микропроцессор 80286 соответствует микропроцессору K1810BM86 (рис.2.11), но имеет дополнительные поля в регистре признаков и регистре состояния. Кроме того, микропроцессор имеет регистры глобальной и локальной таблиц дескрипторов (*GDTR* и *LDTR*), регистр дескрипторов таблицы прерываний (*IDTR*) и регистр задачи (*SR*). Эти регистры (кроме *IDTR*) функционально используются только в защищенном режиме.

#### Регистры общего назначения

	15 8	7 0	
<i>AX</i>	<i>AH</i>	<i>AL</i>	Аккумулятор
<i>BX</i>	<i>BH</i>	<i>BL</i>	База
<i>CX</i>	<i>CH</i>	<i>CL</i>	Счетчик
<i>DX</i>	<i>DH</i>	<i>DL</i>	Данные

#### Регистры указателей и индексов

	15 0
<i>SP</i>	Указатель стека
<i>BP</i>	Указатель базы
<i>SI</i>	Индекс источника
<i>DI</i>	Индекс приемника

#### Регистры сегментов

	15 0
<i>CS</i>	Коды команд

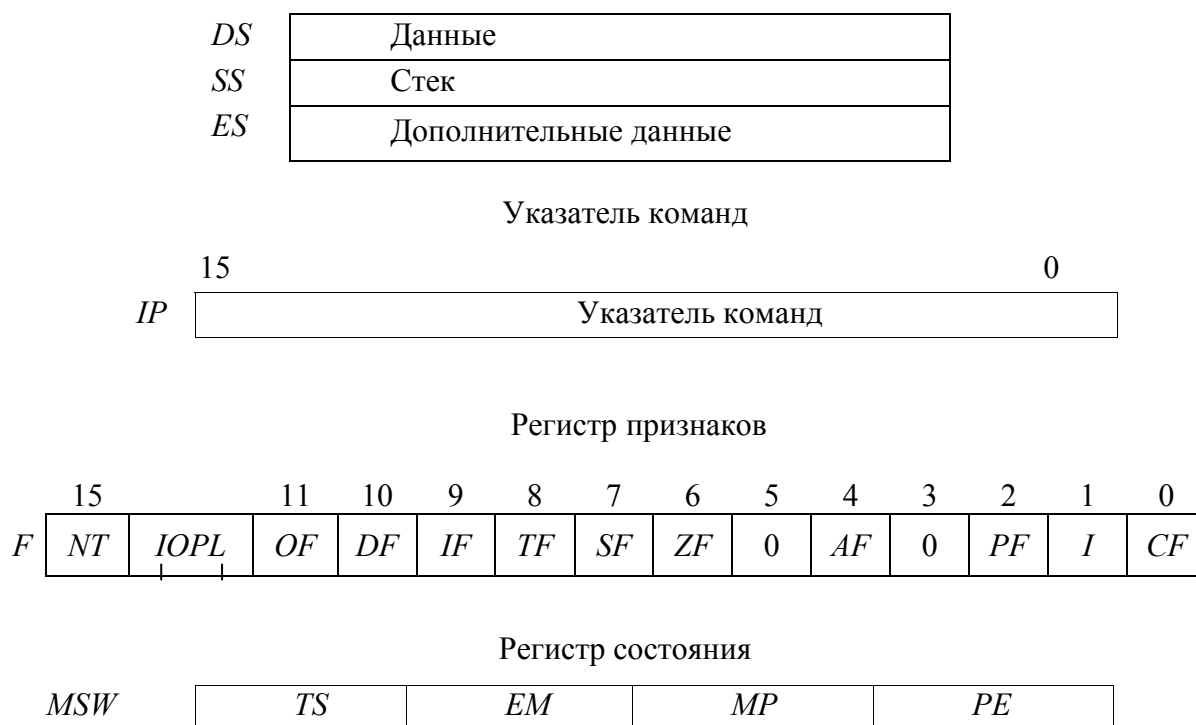


Рис. 2.11 Регистры микропроцессора 80286

Регистры общего назначения (рис. 2.11) предназначены для хранения промежуточных значений вычислений при выполнении задачи. Регистры указателей и индексов используются для локализации информации в определенных областях памяти, а регистры сегментов служат для задания этих областей запоминающего устройства.

Первые восемь регистров (регистры общего назначения и регистры указателей и индексов) предназначены для выполнения арифметических операций и формирования адресов и смещений.

В регистре признаков поля с 0 до 11 совпадают с аналогичными полями регистра признаков микропроцессора K1810BM86. Дополнительно поле *IOPL* (*Input/output Privilege Level*) используется для управления уровнем привилегий ввода-вывода. Поле *NT* (*Nested Task Flag*) применяется для записи признака вложенной задачи.

Режимом работы микропроцессора управляет регистр состояния *MSW*. Его содержимое может загружаться и сохраняться в памяти командами *LMSW* и *SMSW*.

Регистр *MSW* имеет следующие поля:

- *PE* (*Protection Enable*) — разрешение защиты. Установка этого состояния переводит микропроцессор в защищенный режим. Возврат в реальный режим (сброс состояния) возможен только по сигналу *RESET*.
- *MP* (*Monitor Processor Extension*) — мониторинг микропроцессора.

- *EM (Processor Extension Emulated)* — эмуляция микропроцессора. Установка этого состояния позволяет осуществлять его программную эмуляцию.

- *TS (Task Switch)* — переключение процедур выполнения задач. При установке этого состояния следующая команда, относящаяся к микропроцессору, позволяет программно определить, относится ли программа решений микропроцессора к текущей задаче.

Сочетание  $MP = 0$   $EM = 0$   $TS = 0$ , устанавливаемое по аппаратному сбросу, обеспечивает полную совместимость с микропроцессором K1810BM86. При значениях  $MP = 1$ ,  $EM = 0$  предусматривается работа с математическим сопроцессором 80287, а сочетание  $MP=0$ ,  $EM=1$  предполагает программную эмуляцию микропроцессора.

Четыре регистра сегментов (кодов команд данных, стека и дополнительных данных) являются 64-разрядными, причем каждый из регистров содержит 16-разрядный селектор и 48-разрядный дескриптор.

В виртуальном режиме 32-битный указатель называется виртуальным адресом. Он, как и реальный адрес, состоит из 16-битных смещения и селектора – индекса таблицы дескрипторов, хранящейся во внешнем основном запоминающем устройстве (рис. 2.12).

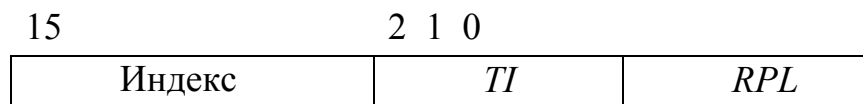


Рис. 2.12 Формат селектора виртуального адреса

Селектор определяет базовый адрес сегмента, к которому прибавляется смещение для получения физического адреса. Базовый адрес получается посредством обращения (индексирования) к таблице в памяти микропроцессора. Существуют три типа таблиц дескрипторов (рис.2.13): локальная таблица дескрипторов *LDT (Local Descriptor Table)*, глобальная таблица дескрипторов *GDT (Global Descriptor Table)* и таблица дескрипторов прерываний *IDT (Interrupt Descriptor Table)*. Размеры таблиц находятся в пределах от 8 байт до 64 Кбайт, что соответствует числу элементов в таблице от 1 Кбайта до 8 Кбайт.

С каждой из этих таблиц связан соответствующий регистр микропроцессора. Регистры *GDTR* и *IDTR* имеют программно-доступное 16-битное поле ограничений, задающее размер таблицы, а также 24-битное поле базового адреса, определяющее положение таблицы в пространстве физических адресов памяти запоминающих устройств. У регистра *LDTR* про-

граммно доступно только 16-битное поле селектора, по которому из глобальной таблицы дескрипторов автоматически загружаются программно-невидимые поля базового адреса и ограничений.

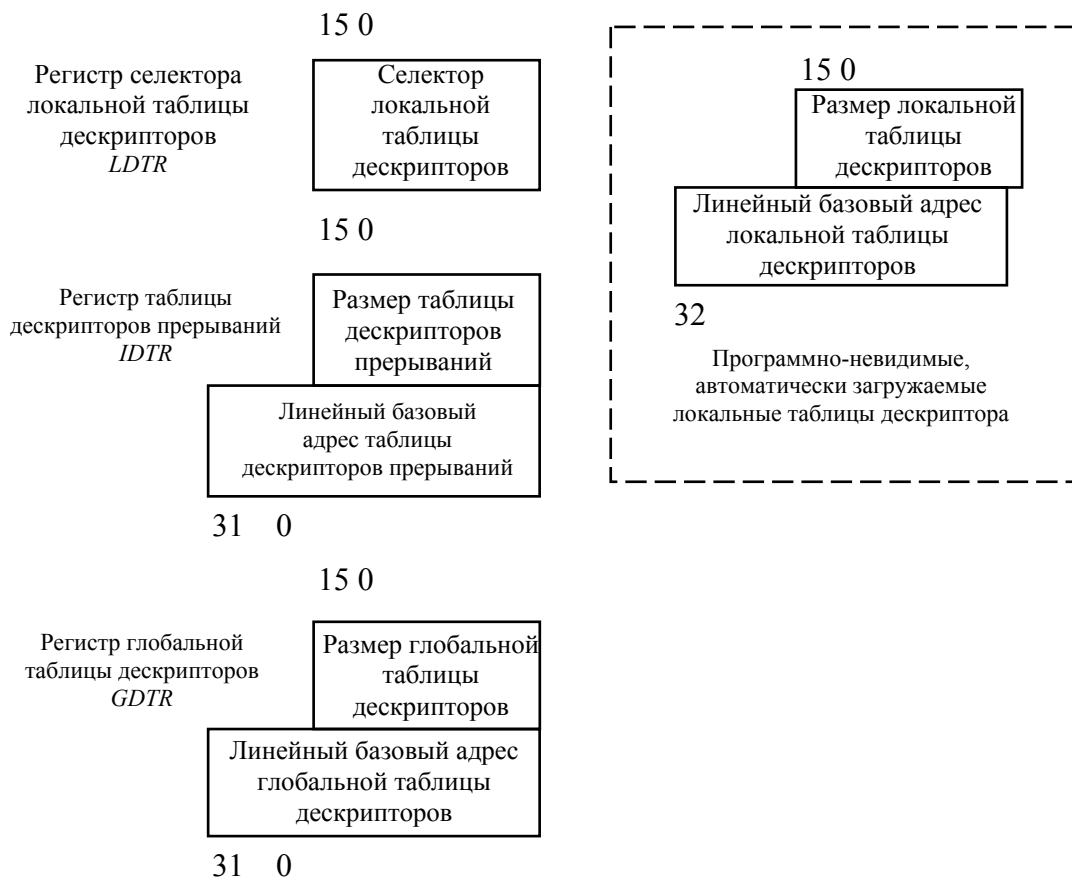


Рис. 2.13 Регистры дескрипторов таблиц

Команды *LGDT*, *LIDT* и *LLDT* загрузки регистров таблиц являются привилегированными. По командам *LGDT* и *LIDT* загружаются из запоминающего устройства 6-байтное поле, содержащее базовый адрес и ограничение локальной таблицы. По команде *LLDT* загружается только селектор, содержащий базовый адрес и ограничение локальной таблицы дескрипторов.

Глобальная таблица *GDT* содержит дескрипторы, доступные всем задачам. Она может содержать дескрипторы любых типов, кроме дескрипторов прерываний и ловушек. Локальная таблица *LDT* может быть собственной для каждой задачи. Она может содержать только дескрипторы сегментов, вентиляей задачи и вызовов.

Таблица дескрипторов прерываний, используемая в защищенном режиме, может содержать описания до 256 прерываний. Базовый адрес и ре-

зерв таблицы загружается привилегированной командой *LIDT*. Размер таблицы должен быть не менее 256 байт для того, чтобы в нее поместились все зарезервированные прерывания микропроцессора. Ссылка на элементы таблицы происходит по командам *INT*, аппаратным прерываниям и исключениям микропроцессора.

Селектор виртуального адреса (см. рис.2.12) содержит 3 поля: запрашиваемый уровень привилегий *RPL*, индикатор таблицы *TI* и индекс. Поле *RPL* используется операционной системой для решения проблем защиты памяти. Поле *TI* показывает какая из 2-х таблиц *GDTR* или *LDTR* привлекается для поиска базового адреса. Если *TI* = 0, то используется глобальная таблица дескрипторов *GDT*. При *TI* = 1 используется локальная таблица дескрипторов *LDT*, причем каждая задача имеет свою таблицу *LDT*. Поле индекса служит адресом сегмента в выбранной таблице. Каждый элемент таблицы является дескриптором, имеет длину 64 бита и содержит 24 битный базовый адрес сегмента. Полученный из выбранного дескриптора базовый адрес суммируется со смещением, в результате чего получается 24-битный физический адрес.

Дескриптор для каждого сегмента (формат дескриптора показан на рис.2.14) содержит базовый адрес, размер сегмента и поле прав доступа. Это поле определяет, правила использования информации данного сегмента.

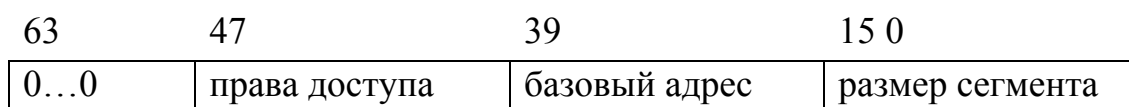
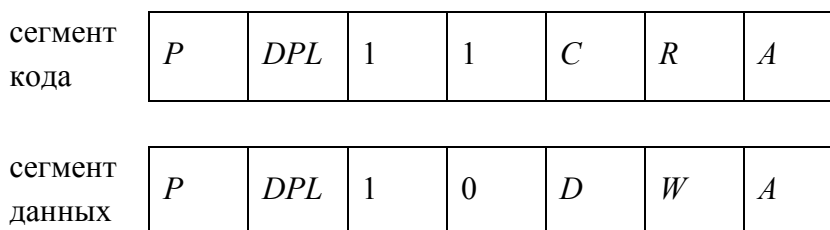


Рис. 2.14 Формат дескриптора сегмента

Например, сегменты кода защищаются по чтению, в то время как для сегментов данных чтение разрешается, но они могут быть защищены по записи. В дескрипторе сегмента поля базового адреса и размера сегмента определяют местонахождение и размер сегмента. В поле прав доступа имеются поля присутствия и уровня привилегированности дескриптора.





сегмент <i>LDTR</i>	<i>P</i>	<i>DPL</i>	0	0	0	1	0
------------------------	----------	------------	---	---	---	---	---

Рис. 2.15 Форматы поля права доступа сегментов

В поле права доступа (рис. 2.15) ячейка *ED* показывает, как интерпретировать поле размера сегмента. Ячейки *P* и *A* помогают работать виртуальной памяти, а ячейки *W* и *R* вводят ограничения на права доступа. Поля *DPL* и *S* осуществляют защиту от несанкционированных обращений.

Поле присутствия *P* (рис. 2.15) определяет, находится ли данный сегмент в основной физической памяти или он размещается во вторичной, внешней системе с виртуальной памятью.

Значение уровня привилегированности дескриптора сравнивается с текущим значением уровня привилегированности микропроцессора. Если текущий уровень привилегированности выше, то программе не разрешается доступ к указанному сегменту. Любая попытка доступа к подобному сегменту приводит к срабатыванию механизма защиты.

Основным механизмом защиты памяти является управляемый доступ к памяти, при котором каждой задаче предоставляется управляемый доступ к двум областям памяти: одной общей и одной частной в соответствии с содержимым глобальной и локальной таблиц дескрипторов.

Микропроцессор 80286 имеет средства контроля за переходом через границу сегмента, работающие и в режиме реальной адресации. При попытке адресации к слову, когда старший байт выходит за границу сегмента, или выполнения команды, все байты которой не умещаются в данном сегменте, микропроцессор вырабатывает прерывание. При попытке выполнения команды *ESCAPE* с операндом памяти, не умещающимся в сегменте, вырабатывается специальный код.

В защищенном режиме виртуальной адресации работают все режимы адресации, допустимые для микропроцессора K1810BM86 и режима реальной адресации микропроцессора 80286. Отличия касаются определения сегментов.

- В регистрах сегментов *CS*, *DS*, *SS* и *ES* хранятся не сами базовые адреса сегментов, а селекторы, по которым из таблицы, хранящейся в ОЗУ, извлекаются дескрипторы сегментов (рис.2.16).
- Дескриптор описывает базовый адрес, размер сегмента (1-64 Кбита) и его атрибуты.
- Базовый адрес сегмента имеет разрядность 24 бита, что обеспечивает адресацию 16 Мбит физической памяти.

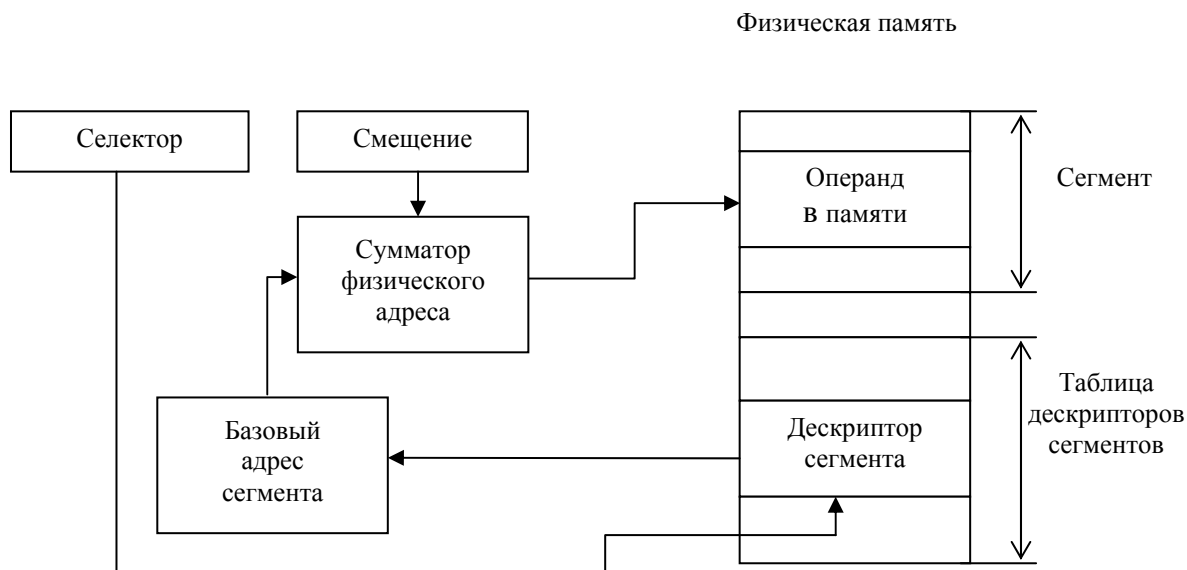


Рис. 2.16. Адресация памяти в защищенном режиме виртуальной адресации

Быстродействующее статическое ОЗУ небольшого объема, называемое кэш-памятью, предназначено для хранения текущей наиболее актуальной информации. Эта информация первоначально хранится в динамическом ОЗУ, быстродействие которого примерно в 10 раз ниже, чем статического. При считывании из динамического ОЗУ (ДОЗУ) информация попутно запоминается в быстродействующем статическом ОЗУ, причем запоминается информация из адресуемой и соседней с ней ячеек ДОЗУ. Так как программы обычно имеют циклический характер, а данные и команды размещены в соседних ячейках, в быстродействующем статическом ОЗУ постепенно накапливаются, а затем автоматически обновляются актуальные коды команд и данных, которыми микропроцессор пользовался.

В быстродействующем статическом ОЗУ накапливаются рабочие копии слов, хранимых в ДОЗУ, причем считывание этих копий производится в 10 раз быстрее, чем считывались бы слова-оригиналы. Если микропроцессор намерен получить информацию из некоторой ячейки ДОЗУ, а копия содержимого этой ячейки уже имеется в быстродействующем статическом ОЗУ, то вместо оригинала считывается копия. Если копии нет, то производится обращение к ДОЗУ со временем считывания 500нС. При записи информации в ячейку ДОЗУ, копия которой есть в быстродействующем статическом ОЗУ, обновляются оригинал и копия.

Если копии не было в быстродействующем статическом ОЗУ, то при обновлении содержимого ячейки ДОЗУ, копия ее в быстродействующем статическом ОЗУ не создается.

Таким образом, быстродействующее статическое ОЗУ ускоряет считывание команд и данных и тем самым позволяет повысить производительность микропроцессора.

Метод виртуальной памяти, используемой в микропроцессоре, обеспечивает иллюзию "безграничной" оперативной памяти при вполне ограниченной ее физической емкости. Идея метода заключается в том, что в случае отсутствия в ОЗУ нужной информации в него записывается необходимая страница или сегмент из внешнего запоминающего устройства, вытесняя одну из "старых" (давно не использовавшихся) страниц или сегментов. Механизм записи страниц поддерживается специальными аппаратными и программными средствами. Таким образом, оперативная память оказывается весьма значительной емкости, которая определяется разрядностью адресной шины.

## **2.2. Тридцатидвухразрядные микропроцессоры**

### **2.2.1 Микропроцессор 80386**

Микропроцессор 80386 является 32-разрядным быстродействующим микропроцессором. Он выполнен по технологии CHMOS III фирмы INTEL, в которой для повышения быстродействия используются технологии NMOS (МДП высокой плотности) и для малого потребления мощности - технологии CMOS (КМДП). Микропроцессор размещен на одном кремниевом кристалле, вставленном в корпус с матричным расположением 132 штырьковых выводов. Главной особенностью микропроцессора является аппаратная реализация программной среды, обеспечивающей совместную работу разнородных программ, ориентированных на разные операционные системы (UNIX, MSDOS и др.).

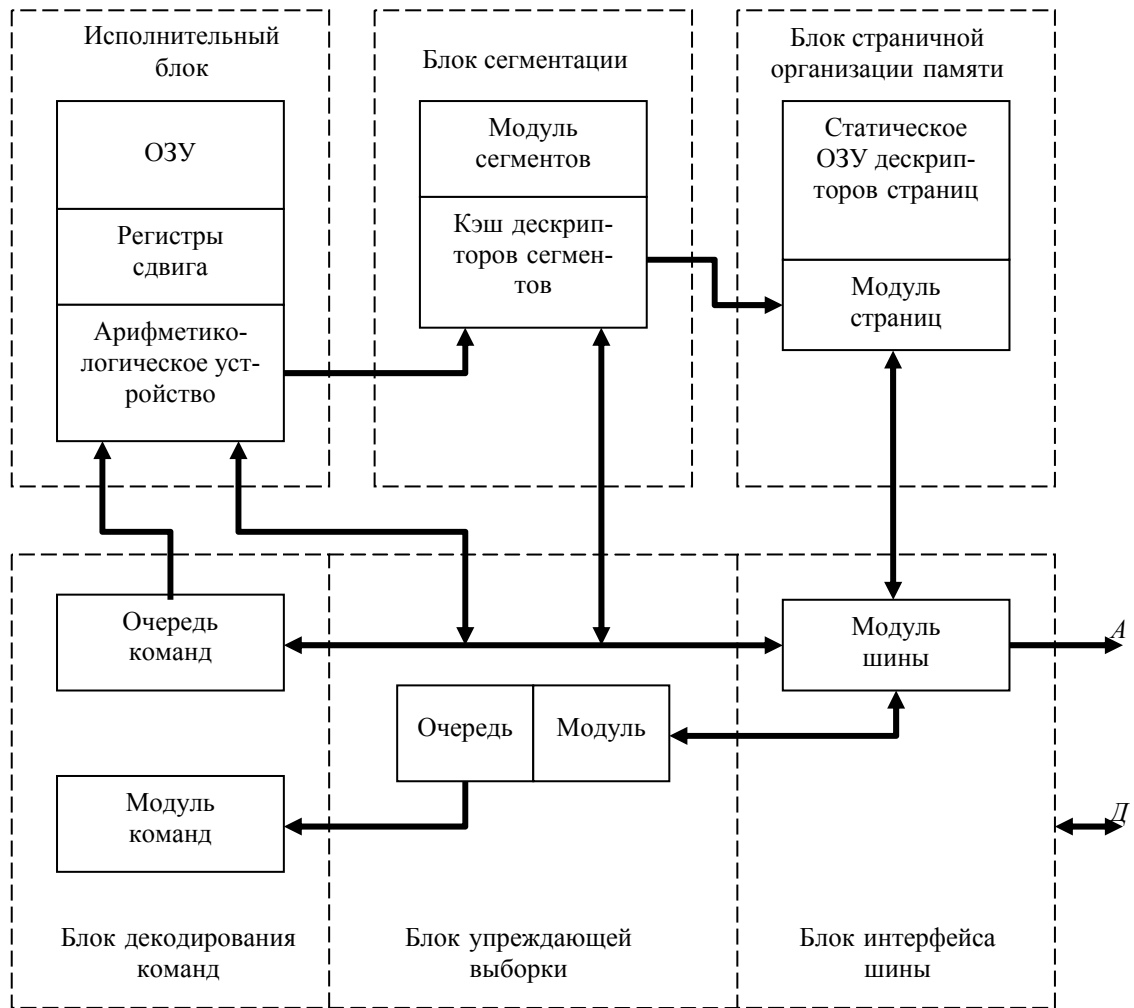


Рис. 2.17 Структурная схема микропроцессора 80386

Архитектурными особенностями микропроцессора 80386 являются следующие:

- Полная 32-разрядная архитектура с физическим адресным пространством 4 Гбайта.
- Наличие средств страничной организации памяти (организация виртуальной памяти емкостью 64 Тбайта).
- Развитые механизмы защиты программ и данных.
- Развитая система прерываний.
- Работа в мультипрограммном режиме.

Микропроцессор 80386 (рис. 2.17) содержит 6 блоков, реализующих управление выполнением команд, сегментацию и страничную организацию памяти, сопряжение с шинами, декодирование и упреждающую вы-

борку команд. Работа всех блоков осуществляется по конвейерному принципу. Во время выполнения одной команды производится декодирование второй и выборка из памяти третьей, занесение результата выполнения четвертой. Микропроцессор имеет следующие регистры:

- Регистры общего назначения, где хранятся адреса и данные.
- Указатель команд и регистр признаков.
- Регистры указателей и индексов.
- Регистры сегментов.
- Управляющие регистры.
- Регистры системных адресов.
- Регистры отладки.
- Регистры тестирования.

Набор регистров общего назначения (рис.2.18) включает соответствующие регистры микропроцессора 80286. Все регистры, кроме сегментных, имеют разрядность 32 бита, и к прежнему обозначению их имен добавилась приставка *E* (*Extended*). Отсутствие приставки в имени означает ссылку на младшие 16 бит расширенных регистров. Возможно независимое обращение к младшему и старшему байтам регистров *AX*, *BX*, *CX* и *DX*.

В шести регистрах сегментов хранятся значения селекторов, указывающих адресуемые в данный момент сегменты памяти: регистр *CS* указывает текущий сегмент программ, регистр *SS* - текущий стековый сегмент, четыре регистра *DS*, *ES*, *FS*, *GS* указывают текущие сегменты данных. Использование регистров сегментов определяется типом обращения к запоминающему устройству. Для многих типов обращений возможно применение альтернативных регистров сегментов, которое вводится префиксами команд *CS:*, *SS:*, *DS:*, *ES:*, *FS:* или *GS:*. С каждым из шести регистров сегментов связаны программно-недоступные регистры дескрипторов, автоматически загружаемые при наполнении соответствующих регистров сегментов. В защищенном режиме в регистры дескрипторов загружается 32-битный базовый адрес, 32-битное определение границ сегмента и атрибуты сегментов. В реальном режиме в качестве базового адреса заносится сдвинутое на 4 бита значение базового адреса сегмента, записанное в сегментном регистре.

### Регистры общего назначения

3	1			
		<i>A</i>	<i>A</i>	<i>E</i>
		<i>H</i>	<i>L</i>	<i>A</i>
				<i>X</i>
		<i>B</i>		<i>E</i>
		<i>X</i>		<i>B</i>
				<i>X</i>
		<i>C</i>		<i>E</i>
		<i>X</i>		<i>C</i>
				<i>X</i>
		<i>D</i>		<i>E</i>
		<i>X</i>		<i>D</i>

### Регистры сегментов

15	0
	<i>CS</i>
	<i>SS</i>
	<i>DS</i>
	<i>ES</i>
	<i>FS</i>
	<i>GS</i>

### Регистры указателей и индексов

31	16	15	7	0	
					<i>SI</i>
					<i>EDI</i>
					<i>EBP</i>
					<i>ESP</i>

### Указатель команд

31	16	15	0
			<i>IP</i>

### Регистр признаков

31	16	15	0
			<i>FLAGS</i>

Рис. 2.18 Основные регистры 32-разрядного микропроцессора 80386

Указатель команд *EIP* содержит смещение следующей исполняемой команды относительно базового адреса сегмента кодов. При 16-битной адресации используются только младшие 16 бит (*IP*).

Регистр признаков *EFLAGS* имеет объем 32 бит (рис.2.19). Биты, определенные для микропроцессора 80286, имеют прежнее назначение.

Назначение бит регистра признаков (рис.2.19) следующее:

- *VM* — в защищенном режиме включает режим виртуального микропроцессора K1810BM86;
- *NT* — признак вложенности задачи;
- *RF* — признак возобновления решения задачи;
- *IOPL* — уровень привилегий ввода-вывода;
- *OF* — признак переполнения, который устанавливается, если результат арифметической операции не умещается в операнде назначения;
- *DF* — признак управления направлением в строковых операциях, причем при значении единицы индексные регистры, участвующие в строковых операциях, автоматически декрементируются на количество байт операнда, а при значении нуля - инкрементируются.
- *IF* — признак управления прерываниями; при значении единицы разрешается выполнение маскируемых аппаратных прерываний;

- *TF* — признак трассировки (пошагового режима), по которому после выполнения каждой команды вызывается внутреннее прерывание типа *I (INT 1)*;
- *SF* — признак знака, который указывает на единичное значение старшего бита результата;
- *ZF* — признак нулевого результата;
- *AF* — признак дополнительного переноса в тетраде для десятичной арифметики;
- *PF* — признак паритета, который устанавливается при четном числе единиц в результате;
- *CF* — признак переноса старшего бита в арифметических операциях.

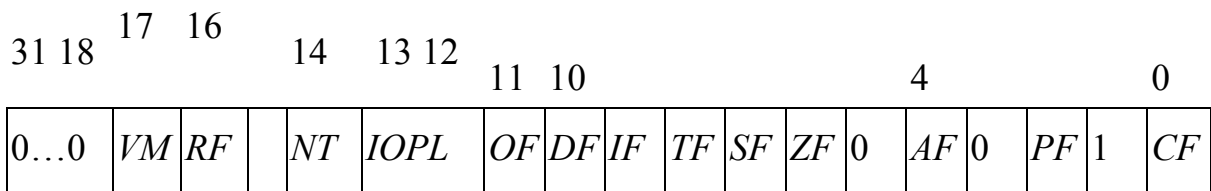


Рис. 2.19 Регистр признаков микропроцессора 80386

Регистры указателей и индексов те же, что у микропроцессора 80286 (см. рис. 2.11).

Структура управляющих регистров *CR0*, *CR1*, *CR2*, *CR3* представлена на рис.2.20.

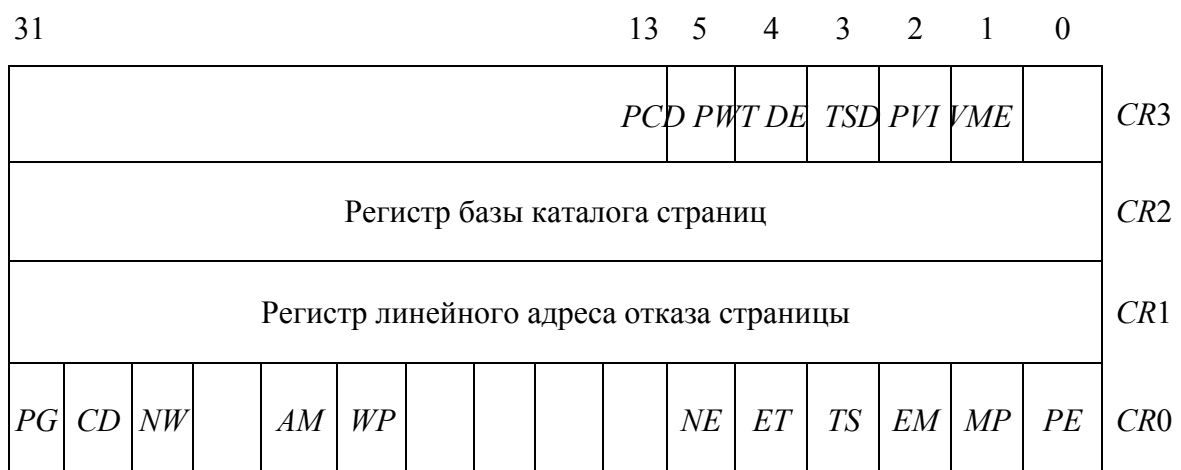


Рис. 2.20 Управляющие регистры

Регистр *CR0* включает в себя биты регистра состояния *MSW* микропроцессора 80286 (рис.2.11). Назначения остальных битов регистра следующие:

- *ET (Extension Type)* — индикатор поддержки команд математического сопроцессора.
- *NE (Numeric Error)* — разрешение стандартного механизма сообщения об ошибке в процессоре чисел с плавающей точкой *FPU* через генерацию исключения. При значении  $NE = 0$  и активном сигнале *IGNNE#*, ошибка *FPU* игнорируются. При  $NE = 0$  и пассивном сигнале *IGNNE#*, при возникновении ошибки *FPU* микропроцессор останавливается и ждет внешнего прерывания по сигналу на выходе *FERR#*. Таким образом, эмулируется обработка ошибки *FPU*.
- *WP (Write Protect)* — разрешение защиты от записи на уровне привилегий супервизора в страницы только для чтения.
- *AM (Alignment Mask)* — разрешение контроля выравнивания, который выполняется на уровне привилегий  $PL = 3$ .
- *NW (Not Writethrough)* — запрет сквозной записи в статическое ОЗУ при значении 1 на поле *AM*.
- *CD (Cache Disable)* — запрет заполнения статического ОЗУ.
- *PG (Paging Enable)* — включение механизма страничной переадресации запоминающего устройства.

Регистр *CR1* в микропроцессоре 80386 не используется. Регистр *CR2* применяется для хранения 32-битного линейного адреса, по которому был получен последний отказ страницы памяти. В старших 20 битах регистра *CR3* хранится физический базовый адрес таблицы каталога страниц. Остальные поля имеют следующее назначение:

- *PCD (Page-Level Cache Disable)* — запрет записи страницы в статическое ОЗУ (кэш-память).
- *PWT (Page-Level Writes Transparent)* — сквозная запись страницы в статическое ОЗУ.
- *VME (Virtual-8086 Mode Extensions)* — разрешение использования виртуального признака прерываний.
- *PVI (Protected-Mode Virtual Interrupts)* — разрешение использования виртуального признака прерываний в защищенном режиме.
- *TSD (Time Stamp Disable)* — превращение команды *RDTSC (Read From Time Stamp Counter)* в привилегированную.



- *DE (Debugging Extensions)* — разрешение точек останова по обращению к портам ввода-вывода.

Регистры системных адресов предназначены для ссылок на сегменты и таблицы в защищенном режиме (рис.2.21). Регистры системных адресов включают 4 специальных регистра для обращения к таблицам или сегментам, реализующим модель защиты:

- Глобальная таблица дескрипторов (*GDTR*);
- Таблица дескрипторов прерываний (*IDTR*);
- Локальные таблицы дескрипторов (*LDTR*);
- Сегмент состояния задачи (*TR*).

Регистры *GDTR (Global Descriptor Table Register)* и *IDTR (Interrupt Descriptor Table Register)* программно загружаются 6-байтными операндами, включающими 32-битный линейный базовый адрес и 16-битный размер глобальной таблицы дескрипторов и таблицы дескрипторов прерываний. В регистр задачи *TR (Task Register)* и регистр селектора локальной таблицы дескрипторов *LDTR (Local Descriptor Table Register)* загружаются 16-битные селекторы дескрипторов сегмента состояния задачи *TSS* и локальной таблицы дескрипторов *LDT*. Эта загрузка вызывает автоматическую загрузку самих дескрипторов, содержащих 32-битные поля линейного базового адреса и границ, а также полей атрибутов, в связанные с ними регистры дескрипторов.

Регистры отладки предназначены для задания и управления отладочными точками останова. Шесть программно-доступных регистров отладки реализуют поддержку процесса отладки программ. Четыре регистра предназначены для указания точек останова. Управляющий регистр используется для установки контрольных точек, а статусный регистр показывает текущее состояние точек останова. Эти регистры обеспечивают задание контрольных точек, останова по командам и данным, а также пошаговый режим выполнения программы. Регистры отладки имеют следующие назначения:

- *DR0...DR3 (Linear Breakpoint Address 0...3)* хранят 32-битные линейные адреса точек останова;
- *DR4, DR5* в микропроцессоре 80386 не используются;
- *DR6 (Breakpoint Status)* отражает состояние контрольной точки;
- *DR7 (Breakpoint Control)* управляет установкой контрольных точек.

- *TR6 (Test Control)* — управляющий регистр для теста записи страниц в статическое ОЗУ;
- *TR6 (Test Control)* — управляющий регистр для теста записи страниц в статическое ОЗУ.

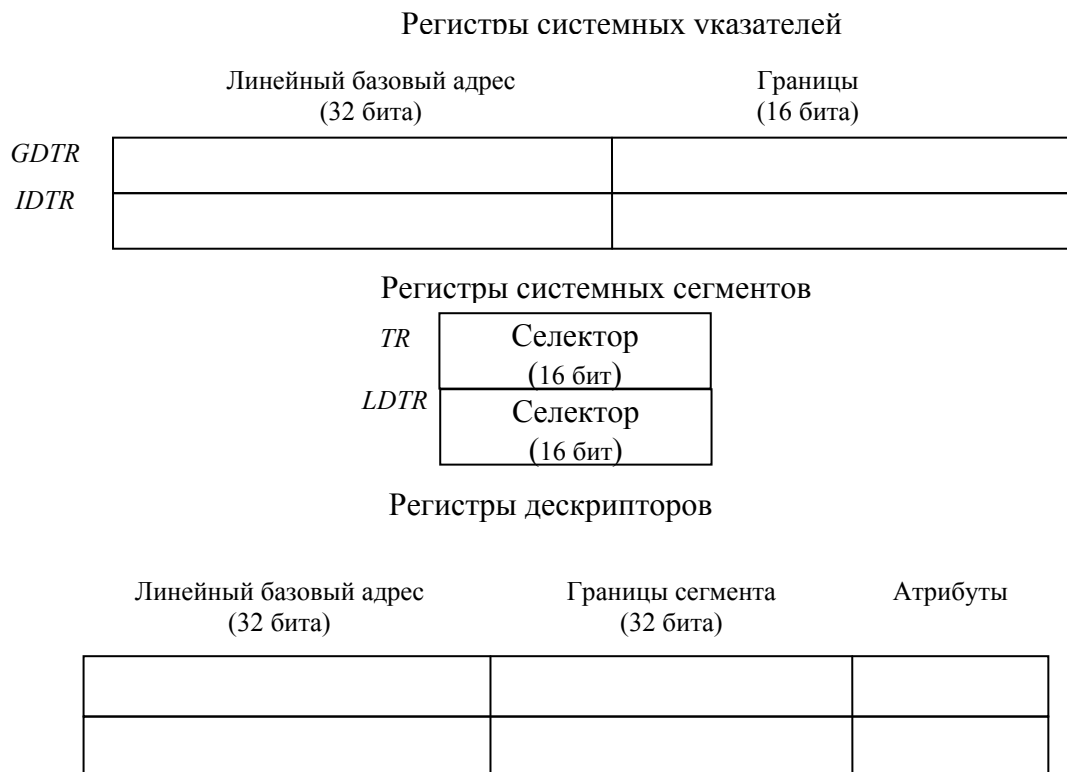


Рис. 2.21 Регистры системных адресов



Рис. 2.22 Дескриптор сегмента кода и данных

Регистры дескрипторов сегментов кода и данных (рис.2.22) содержат линейный базовый адрес сегмента, длину и дробность его, уровень защиты-привилегии по чтению, записи, выполнению, сведения о размере операндов, тип сегмента. Дескрипторы формируются системными программами (загрузчиком, редактором связей, компилятором).

База определяет начальный адрес сегмента и формируется из трех фрагментов дескриптора. Размер сегмента формируется из 2-х фрагментов. Если бит дробности  $G = 0$ , то размер интерпретируется в байтах. При значении  $G = 1$  размер определяется в единицах по 4 Кбайт, что соответствует одной странице.

Разряд сегмента  $S$  определяет принадлежность сегмента к системным ( $S = 0$ ) или прикладным программ.

Поле "Тип" содержит биты читаемости  $R$ , согласованности  $C$ , разрешения записи  $W$ , направления расширения  $E$ . Бит обращения  $A$  дескриптора служит для указания на доступ к сегменту с момента его очистки. При обмене страниц этот бит не используется.

Бит присутствия  $P$  сегмента определяет возможность использования дескриптора при трансляции адресов. При значении  $P = 0$  трансляция невозможна.

Уровень привилегий дескриптора определяется значением битов  $DPL$ . При  $DPL = 0$  максимальные привилегии даются ядру операционной системы (см. рис.2.9), для сервисных программ операционной системы  $DPL = 1$ . При  $DPL = 2$  привилегии даются управлению базами данных, а для  $DPL = 3$  обеспечивается привилегированный уровень прикладных программ.

Бит  $AVL$  определяет доступность для использования сегмента кода и данных. Бит  $D$  определяет разрядность относительного адреса и операнда (16 и 32). При операциях со стеком он определяет разрядность указателя стека.

В режиме реальной адресации только адрес базы обновляется непосредственным вычислением (сдвигом значения селектора на 4 бита), т.к. размер сегмента и атрибуты в этом режиме фиксированы. В режиме виртуальной адресации с защитой базовый адрес, размер и атрибуты обновляются согласно содержимому дескриптора сегмента, расположенного в памяти и индексированного в соответствующей таблице по селектору.

При обращении к памяти 32-разрядный базовый адрес сегмента участвует в вычислении линейного адреса, а 32-разрядный размер сегмента используется для проверки границ. Атрибуты проверяются на соответствие типу затребованного обращения к памяти.

31 16

15 0

Базовый адрес (биты 15-0)					Размер сегмента (биты 15-0)								
База (биты 31-24)	<i>G</i>	0	0	0	Размер (биты 19-16)	<i>P</i>	<i>DPL</i>	0	Тип <i>R C W E</i>			<i>A</i>	База (биты 23-16)

Рис. 2.23 Deskриптор системного сегмента

Три 32-разрядных управляющих регистра вместе с регистрами системных адресов содержат слова состояния микропроцессорной системы, влияющие на процедуры решения задач в системе.

Deskрипторы системных сегментов предназначены для хранения локальных таблиц deskрипторов *LDT* и таблиц состояния задач *TSS*. Их deskрипторы определяют базовый адрес, размер сегмента, права доступа (чтение, чтение/запись, только исполнение кода или исполнение/чтение) и присутствие сегмента в физической памяти.

Бит *P* определяет действительность ( $P = 1$ ) или недействительность ( $P = 0$ ) содержимого сегмента. Биты уровня привилегий *DPL* используются в deskрипторах сегментов состояния задач. Поскольку обращение к локальным deskрипторам возможно только по привилегированным командам, биты *DPL* для deskрипторов таблиц не используются. Поле "Тип" определяет тип сегмента.

Межсегментная передача управления непосредственно (командами *JMP*, *CALL*, *INT*, *RET* и *IRET*) возможна только к сегментам кода с тем же уровнем привилегий, либо к подчиненным сегментам, уровень привилегий которых выше. Для переходов с изменением уровня привилегий используются вентили, иногда называемые шлюзами. Для каждого способа косвенной межсегментной передачи управления имеются соответствующие вентили. Их использование позволяет микропроцессору автоматически выполнять контроль защиты. Вентили вызова используются для вызовов процедур со сменой уровня привилегий. Вентили задач используются для переключения задач, а вентили прерываний и ловушек определяют процедуры обслуживания прерываний. Вентили вызова позволяют автоматически копировать заданное число слов из старого в новый стек. Вентили прерываний отличаются от вентиля ловушек тем, что они запрещают прерывания.

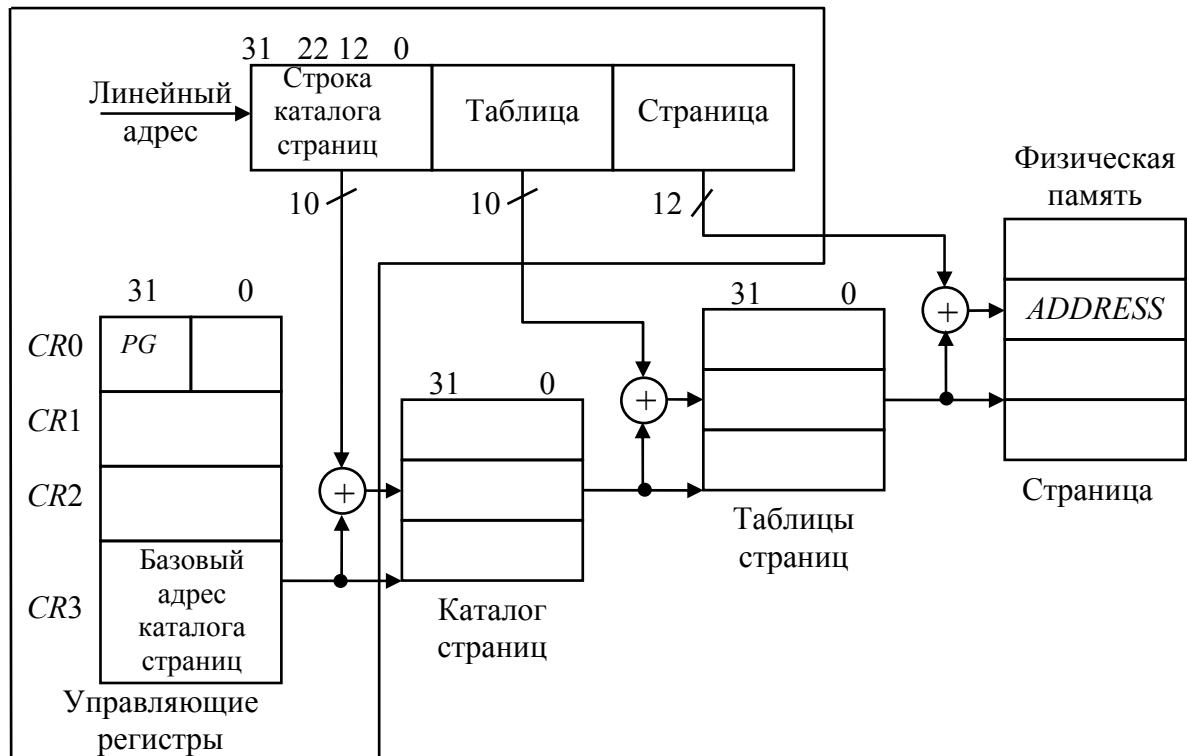


Рис. 2.24. Механизм страничной преадресации

Страничное управление является дополнительным средством организации виртуальной памяти. В отличие от сегментации, которая организует программы и данные в модули различного размера, страничная организация подразделяет модули на множество небольших страниц одинакового размера. Страницы не имеют прямой связи с логической структурой данных или программ. В то время как селекторы можно рассматривать как логические имена модулей кодов и данных, страницы представляют части этих модулей. Учитывая свойство локальности (близкого расположения требуемых ячеек памяти) кода и ссылок на данные, в оперативной памяти в каждый момент времени хранятся только небольшие области сегментов, необходимые активным процедурам решения задач. Такая организация виртуальной памяти позволяет увеличить допустимое число одновременно выполняемых задач при ограниченном объеме оперативной памяти.

Механизм страничного управления использует двухуровневую табличную трансляцию линейного адреса в физический (рис. 2.24).

Механизм имеет три части: каталог страниц, таблицы страниц и собственно страницы. Работа механизма начинается с установки – значение 1 на поле *PG* регистра *CR0*. Регистр *CR2* хранит линейный адрес отказа и адрес памяти, по которому был обнаружен последний отказ от загрузки

страницы. Регистр *CR3* хранит физический адрес каталога страниц. Младшие 12 бит разрядов этого регистра нулевые, поскольку каталог выравнивается по границе страницы.

Каталог страниц имеет 1024 строки. Каждая строка (рис.2.25) содержит 20 старших бит адреса таблицы следующего уровня и признаки (атрибуты) этой таблицы. Индексом поиска в каталоге страниц является 10 старших бит линейного адреса.

Таблица страниц также имеет 1024 строки аналогичного формата, но эти строки содержат базовый физический адрес и атрибуты самих страниц. Индексом поиска в таблице являются поля линейного адреса. Физический адрес собирается из адреса страницы, взятого из таблицы, и младших 12 бит линейного адреса. Строки каталога страниц и таблицы страниц имеют следующие биты (рис.2.25) атрибутов:

- *P (Present)* — бит присутствия. При  $P = 1$  имеется возможность использования данной строки для трансляции адреса.
- *A (Accesed)* — признак доступа, который устанавливается перед чтением или записью по адресу, в преобразовании которого участвует данная строка.
- *D (Dirty)* — признак, устанавливаемый перед операцией записи по адресу, в преобразовании которого участвует данная строка.

Биты *P*, *A*, *D* модифицируются микропроцессором аппаратным методом в заблокированных шинных циклах. При их программной модификации в многопроцессорных системах используется префикс *LOCK*, гарантирующий сохранение целостности данных.

Биты *OS Reserved* программно используются по усмотрению операционной системы и в них может храниться, например, информация о «возрасте» страницы, необходимая для реализации процедуры замещения наиболее давно не использовавшаяся страницы первой.

31	12	11	10	9	8	7	6	5	4	3	2	1	0
Таблицы адресов страниц (31...12)		<i>OS Reserved</i>		0	0	<i>D</i>	<i>A</i>	<i>PCD</i>	<i>PWT</i>	<i>U/S</i>	<i>R/W</i>	<i>P</i>	

Рис. 2.25 Строка каталога страниц

Бит *PWT (Page Write Through)* определяет политику записи в статическое ОЗУ. Бит *PCD (Page Cache Disable)* запрещает запись в статическое ОЗУ обслуживаемых страниц.

Механизм защиты страниц различает два уровня привилегий: пользователь и супервизор. Пользователю соответствует уровень привилегий  $PL = 3$ , супервизору — уровни  $PL = 0, 1$  и  $2$ . Строки таблиц имеют атрибуты защиты страниц: биты  $U/S$  (*User/Supervisor*) и  $R/W$  (*Read/Write*) (см. рис.2.25). Эти атрибуты в строке каталога страниц относятся ко всем страницам, на которые ссылается данная строка через таблицу уровня  $PL = 2$ . Атрибуты относятся к конкретной странице памяти, которую она обслуживает. Права доступа к странице логического адресного пространства приведены в табл.2.2.

Таблица 2.2.

$U/S$	$R/W$	Разрешено при $PL = 3$	Разрешено при $PL = 0,1,2$
0	0	Нет	Чтение/запись
0	1	Нет	Чтение/запись
1	0	Только чтение	Чтение/запись
1	1	Чтение/запись	Чтение/запись

При каждой операции доступа обращение к запоминающему устройству через две таблицы, существенно снижает производительность микропроцессора. Для повышения быстродействия в микропроцессор введено запоминающее устройство ассоциативной трансляции  $TLB$  (*Translation Lookaside Buffer*), предназначенное для хранения интенсивно используемых строк таблиц. Запоминающее устройство представляет собой четырехканальное наборно-ассоциативное статическое ОЗУ на 32 строки таблиц трансляции. Такой размер позволяет хранить информацию о трансляции 128 Кбит памяти.

Появление значения 1 бита  $PG$  (рис.2.24) разрешает выполнение страничного управления. Блок страничной переадресации получает 32-битный линейный адрес от блока сегментации. Старшие 20 бит адреса сравниваются со значениями из запоминающего устройства ассоциативной трансляции  $TLB$ . В случае совпадения физический адрес вычисляется с использованием значения начального адреса страницы, полученного из  $TLB$ , и выводится на шину адреса. Если соответствующей строки в запоминающем устройстве ассоциативной трансляции  $TLB$  нет, то производится чтение строки из страничного каталога. Если значение бита  $P$  строки каталога страниц (рис.2.25) 1, в этой строке устанавливается бит доступа  $A = 1$  и производится чтение ею указанной строки из таблицы второго уровня. Наличие в этой строке  $P = 1$ , приводит к обновлению в ней полей  $A$  и  $D$ , а

также вычислению физического адреса и обращения к нему. Если на этом пути встречается значение бита  $P = 0$ , вырабатывается сигнал отказа и принимаются меры по загрузке затребованной страницы в оперативную память. После успешной обработки сигнала отказа повторяется доступ к затребованной ячейке памяти.

Отказ может возникать и при нарушении прав доступа, определяемых уровнем привилегий и битами  $U/S$  и  $R/W$ . Для идентификации причины отказа в стек помещается 16-битный код ошибки, формат которого приведен на рис.2.26. Хотя названия битов совпадают с названиями битов атрибутов строк, их назначения отличаются. Бит  $U/S$  указывает на уровень привилегий, при котором случился отказ. Бит  $W/R$  указывает на операцию, при которой произошел отказ. Если  $W/R = 0$ , то операцией было чтение, а при  $W/R = 1$  – запись. Бит  $P$  указывает на причину отказа. При значении  $P = 1$  причиной отказа является отсутствие страницы, а при  $P = 0$  – нарушение защиты. Биты  $U$  не используются.

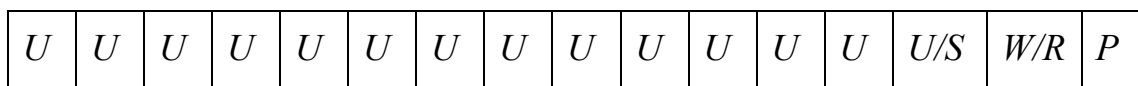


Рис. 2.26. Формат кода ошибки при отказе страницы

Операционная система формирует начальные таблицы трансляции и обрабатывает исключения отказов. В случае изменения таблиц она очищает запоминающее устройство ассоциативной трансляции  $TLB$  путем загрузки регистра  $CR3$  (см. рис.2.24). Очистка запоминающего устройства ассоциативной трансляции  $TLB$  производится и при изменении значений поля  $P$  в любых таблицах.

### 2.2.2 Микропроцессор 80486

Микропроцессор 80486 является быстродействующим 32-разрядным микропроцессором. Микросхема микропроцессора содержит в одном кристалле 32-разрядный процессорный блок, сопроцессор обработки чисел с плавающей точкой, диспетчер памяти и статическое ОЗУ емкостью 8 Кбайт. Основными особенностями микропроцессора 80486 являются:

- Полная 32-разрядная архитектура, включающая 32-разрядные регистры и шины.
- Поддержка программного обеспечения, работающего с операционными системами *MS-DOS*, *OS/2*, оболочкой *WINDOWS*, *OS UNIX* и др.



- Размещение на одном кристалле процессорного блока, сопроцессора обработки чисел с плавающей точкой (*FPU*), статического ОЗУ команд и данных объемом 8 Кбайт, диспетчера памяти.
- Тактовые частоты работы микропроцессора составляют 25/33/50 МГц, при скорости обмена 160 Мбайт/сек.
- Поддержка многопроцессорных систем.

В микропроцессоре наиболее часто используемые команды реализуются за один цикл, что обеспечивает его высокую производительность. Микропроцессор 80486 (рис.2.27) содержит устройство цифровой обработки, *FPU*, устройство управления (УУ), диспетчер памяти, блок предвыборки команд и данных, интерфейс магистрали. Устройство цифровой обработки включает арифметико-логическое устройство, восемь 32-разрядных регистров общего назначения (регистровый файл) и многоразрядное устройство сдвига, используемое при арифметических и циклических сдвигах, умножении и делении. Сопроцессор *FPU* по структуре и по программному обеспечению, соответствует сопроцессору 80382. Он имеет собственные регистры данных и управления, работает параллельно с устройством цифровой обработки, обеспечивая обработку данных по стандарту *IEEE 754-1985*.

Устройство управления представляет собой дешифратор команд и блок микропрограммного управления, содержащий ПЗУ микропрограмм.

Диспетчер памяти состоит из блока сегментации и блока страничной адресации. Он осуществляет двухступенчатое формирование физического адреса ячейки памяти сначала в пределах сегмента, а затем в пределах страницы. При этом используется четыре уровня защищенности.

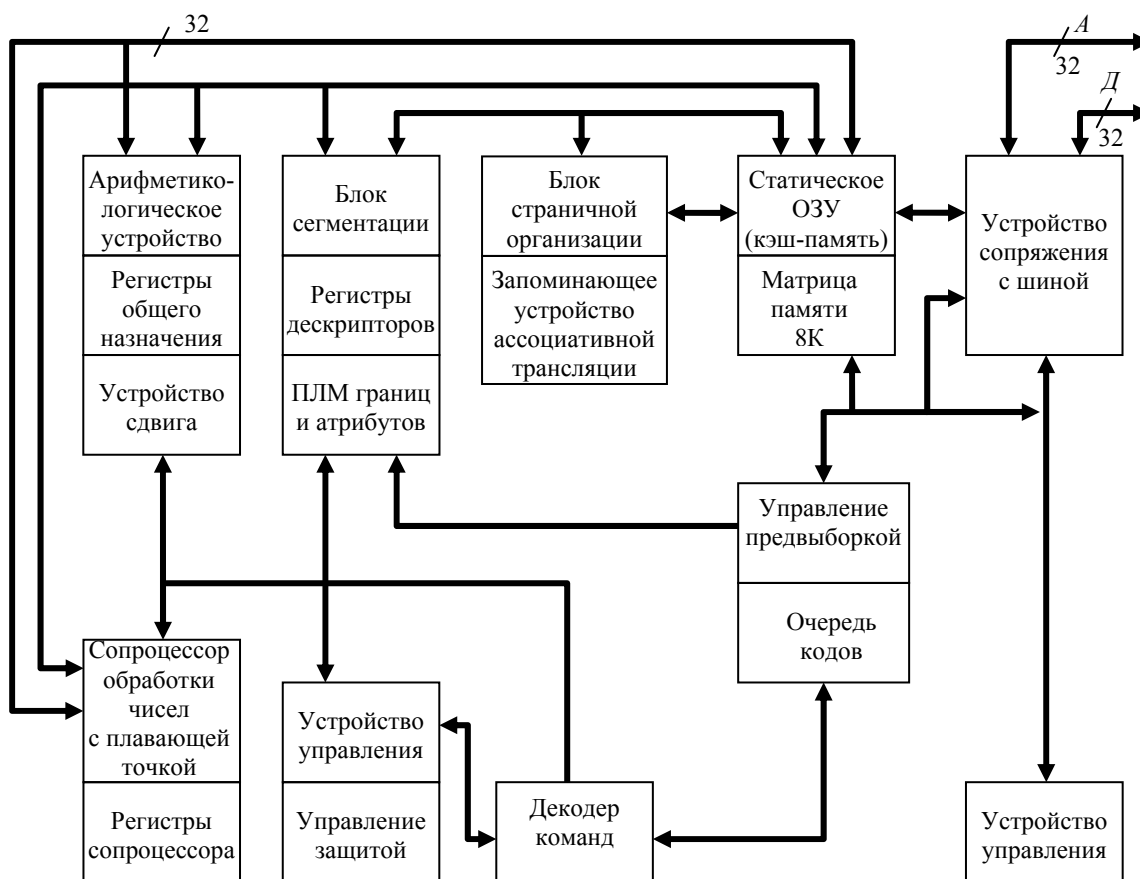


Рис. 2.27 Структурная схема микропроцессора 80486

Сегментированная память представляет собой набор блоков, характеризуемых расположением, размером, типом, характеристикой защиты. В микропроцессоре 80486 имеется 16384 сегментов до 4 Гбайт каждый. Виртуальная память достигает 64 Тбайт, и каждый сегмент делится на страницы по 4 Кбайта.

Блок предвыборки команд и данных включает устройство предвыборки команд и внутреннее статическое оперативное запоминающее устройство.

Интерфейс магистрали осуществляет обмен с помощью двунаправленной 32 разрядной шины данных, 32 разрядной шины адреса и 32 разрядной шины управления.

По шине данных может передаваться 8,16 или 32 бита. Поэтому шина адреса имеет 30 адресных линий и 4 линии выбора байтов. Сигналы выбора байтов определяют байты, участвующие в текущем цикле обмена, что позволяет без дополнительных аппаратных средств согласовать 32-разрядную шину данных с байтной организацией памяти микропроцессорной системы.

Циклы работы магистрали определяют одиннадцать бит шины управления, а 21 бит определяют состояние микропроцессора и его взаимодействие с другими активными внешними устройствами.

В микропроцессоре имеется 4 группы регистров:

- Основные регистры, аналогичные регистрам микропроцессора 80386.
- Регистры сопроцессора.
- Системные регистры.
- Регистры отладки и тестирования.

Регистры сегментов содержат значения селекторов сегментов, указывающих на текущие адресуемые сегменты памяти. С каждым из них связан программно недоступный регистр дескриптора сегмента. Регистр дескриптора содержит 32 битовый базовый адрес сегмента, 32 бита размера сегмента и другие необходимые атрибуты. При загрузке в регистр сегмента нового значения селектора, содержимое дескриптора автоматически корректируется. В реальном режиме базовый адрес сегмента получается путем сдвига значения селектора на 4 разряда, причем максимальный размер и атрибуты в этом режиме оказываются фиксированными. Указатель команд *EIP* (см. рис.2.18) определяет смещение при задании адреса следующей команды. Смещение задается относительно базового адреса сегмента *CS* в регистре сегментов.

В регистре признаков *EFLAGS* введен признак контроля выравнивания операндов *AC* при обращении к памяти.

В состав сопроцессора обработки чисел с плавающей точкой входят:

- регистры данных (*R0...R7*);
- регистры признаков;
- регистры управления и состояния;
- указатели команд и данных.

В состав системных регистров входят 3 регистра управления (*CR0..CR2*) и 4 регистра системных адресов и сегментов (*GDTR, LDTR, IDTR* и *TR*).

В регистре *CR0* введены 5 разрядов для управления встроенным статическим ОЗУ и устройством выполнения операций с плавающей точкой. Регистры управления служат для фиксации общего состояния микропроцессора.

В микропроцессоре применяется внутреннее статическое ОЗУ (внутренняя кэш-память), которое имеет наборно-ассоциативную архитектуру.

Работу внутреннего статического ОЗУ рассмотрим на примере четырехканальной наборно-ассоциативной кэш-памяти. Ее физическая структура приведена на рис.2.22.

Работу внутренней кэш-памяти характеризуют следующие процессы:

- Обслуживание запросов микропроцессора 80486 на обращение к памяти.
- Выделение и замещение строк для общения с кэш-памятью областей физической памяти.
- Обеспечение согласованности данных внутренней кэш-памяти и оперативной памяти.
- Управление работой кэш-памяти.

Внутренний запрос микропроцессора на обращение к памяти направляется во внутреннюю кэш-память. Этикетки (теги) четырех строк, которые обслуживают данный адрес, сравниваются со старшими битами запрошенного физического адреса. Если адресуемая область имеется в строке внутренней кэш-памяти, запрос на чтение обслуживается только ею, не выходя на внешнюю шину. Запрос на запись модифицирует данную строку и, в зависимости от программы записи, либо сразу выходит на внешнюю шину (при сквозной записи), либо несколько позже (при использовании алгоритма обратной записи).

В случае, когда адресуемая область отсутствует в строке внутренней кэш-памяти, запрос на запись направляется на внешнюю шину. Если этот запрос относится к области памяти, для которой разрешен обмен с кэш-памятью, выполняется цикл заполнения строки кэш-памяти. Все 16 байт считываются из оперативного запоминающего устройства и помещаются в одну из строк канала кэш-памяти, обслуживающего данный адрес. Если затребованные данные не укладываются в одной строке, заполняется соседняя строка.

Внутренний запрос микропроцессора на затребованные данные удовлетворяется после считывания их из ОЗУ. Дальнейшее заполнение строки происходит параллельно с обработкой полученных данных. Если в канале, который обслуживает данный адрес ячейки памяти, имеется свободная строка, она заполняется и для нее устанавливается бит достоверности. При отсутствии свободных строк в канале, замещается строка, к которой дольше всех не было обращений. Выбор строки для замещения выполняется на основе анализа бит *LRU (Least Recently Used)* (см. рис.2.28). Эти биты модифицируются при каждом обращении к строке данного канала.



Рис. 2.28 Структура кэш-памяти микропроцессора 80486

Таким образом, когда адресуемая область отсутствует в строке внутренней кэш-памяти, выделение и замещение строк, выполняется только для чтения. Если затребованная область памяти присутствует в строке внутренней кэш-памяти, то она обслужит этот запрос. Управление этим процессом происходит на этапе заполнения строк. Существует возможность их аннулирования и очистки всей внутренней кэш-памяти.

Очистка внутренней памяти при сквозной записи осуществляется внешним сигналом *FLUSH#*. Кроме того, имеются команды аннулирования *INVD* и *WBINVD*. Команда *INVD* аннулирует строки внутренней кэш-памяти без выгрузки модифицированных строк. Команда *WBINVD* предварительно выгружает модифицированные строки в основную память. При обратной записи очистка внутренней кэш-памяти подразумевает и выгрузку всех модифицированных строк в основную память.

В микропроцессорных системах, у которых в оперативное запоминающее устройство запись может производить не только микропроцессор, но и контроллеры шины — периферийные контроллеры, аннулирование строк выполняется внешними устройствами. В этом случае требуются специальные средства для поддержания согласованности данных в кэш-памяти первого и второго уровней и динамическим ОЗУ. Если внешний контроллер выполняет запись в память, микропроцессор получает сигнал *AHOLD*. По этому сигналу микропроцессор передает управление шиной адреса, на которой внешним контроллером устанавливается адрес ячейки памяти, сопровождаемый сигналом *EADS#*. Если адресованная память присутствует в первичной внутренней кэш-памяти, микропроцессор аннулирует строку. Аннулирование строки микропроцессор выполняет в любом состоянии.

Управление заполнением внутренней кэш-памяти возможно на аппаратном и на программном уровнях. Микропроцессор позволяет записать в кэш-память любую область физической памяти. Внешние устройства могут запрещать микропроцессору работу кэш-памяти с определенными областями памяти сигналом  $KEN\#$  во время циклов доступа к этим областям памяти. Этот сигнал управляет возможностью заполнения строк внутренней кэш-памяти из адресованной области памяти. Программным способом можно управлять общением кэш-памяти с каждой страницей памяти, устанавливая единичное значение на поле  $PCD$  (см. рис.2.25) в таблице или каталоге страниц. Бит  $PWT$  позволяет постранично управлять и алгоритмом записи. Общее программное управление работой кэш-памяти осуществляется посредством бит  $CD$  и  $NW$  управляющего регистра  $CR0$  (см. рис.2.20). Возможны следующие сочетания бит регистра.

- $CD = 1, NW = 1$ . После установки такого сочетания возможна очистка кэш-памяти и ее отключение. Если очистка не производилась, кэш-памяти превращается в «замороженную» область статической памяти.
- $CD = 1, NW = 0$ . Заполнение кэш-памяти запрещено, но сквозная запись разрешена. Этот режим может использоваться для временного отключения кэш-памяти, после которого возможно ее включение без очистки.
- $CD = 0, NW = 1$ . Запрещенная комбинация.
- $CD = 0, NW = 0$ . Нормальный режим работы со сквозной записью.

### 2.2.3 Микропроцессор Pentium

Микропроцессор *Pentium* (рис. 2.29) имеет быстродействие в 5 раз выше, чем микропроцессор 80486. Это качество обеспечивается следующим:

- Архитектурой, позволяющей проводить выполнение параллельно двух команд в одном цикле работы.
- Динамической предвыборкой команд программы, использующей специальную память ветвления ( $BTB$ ).
- Конвейеризованным блоком плавающей точки ( $FPU$ ), реализующим вычисления с 32, 64 и 80 разрядами.
- Раздельными внутренними кэш-памятью команд и 2-х портовой внутренней кэш-памятью данных, что позволяет параллельно выбирать команду и два операнда.
- Применением 64-разрядной шины данных и 32-разрядной шины адреса.

- Возможностью организации параллельной работы двух микропроцессоров.
- Поддержкой протокола обратной записи для внутренней кэш-памяти данных.
- Реализацией технологии со средствами управления энергопотреблением.

В микропроцессор введены дополнительные средства тестирования, диагностики, расширены средства отладки по адресам памяти и адресам портов адресного пространства ввода-вывода. Большая интегральная схема микропроцессора содержит 3.1 млн. транзисторов, размещается в корпусе с 296 выводами и работает с тактовыми частотами до 200 МГц.

Два конвейера микропроцессора в схеме на рис.2.29 реализуются с помощью арифметико-логических устройств  $U$  и  $V$ , генераторов адресов  $U$  и  $V$  и устройства управления, сигналы на которые подаются с управляющего ПЗУ.

Для выполнения операций оба конвейера используют регистры и устройство барабанного сдвига. Оба конвейера могут выполнять две команды одновременно. Однако они не являются независимыми, т.к. существуют правила, какие команды могут выполняться одновременно и при останове одного конвейера останавливается и другой.

Конвейерное устройство операций с плавающей точкой (рис.2.29) обладает более высокой производительностью, чем в микропроцессоре 80486. Оно содержит все необходимые узлы для реализации арифметических операций, файл регистров и устройство управления. В состав микропроцессора введены усовершенствованный контроллер прерывания (*APIC – Advanced Programmable Interrupt Controller*) и средства поддержки многопроцессорных систем (на рис.2.29 – двухпроцессорное логическое устройство).

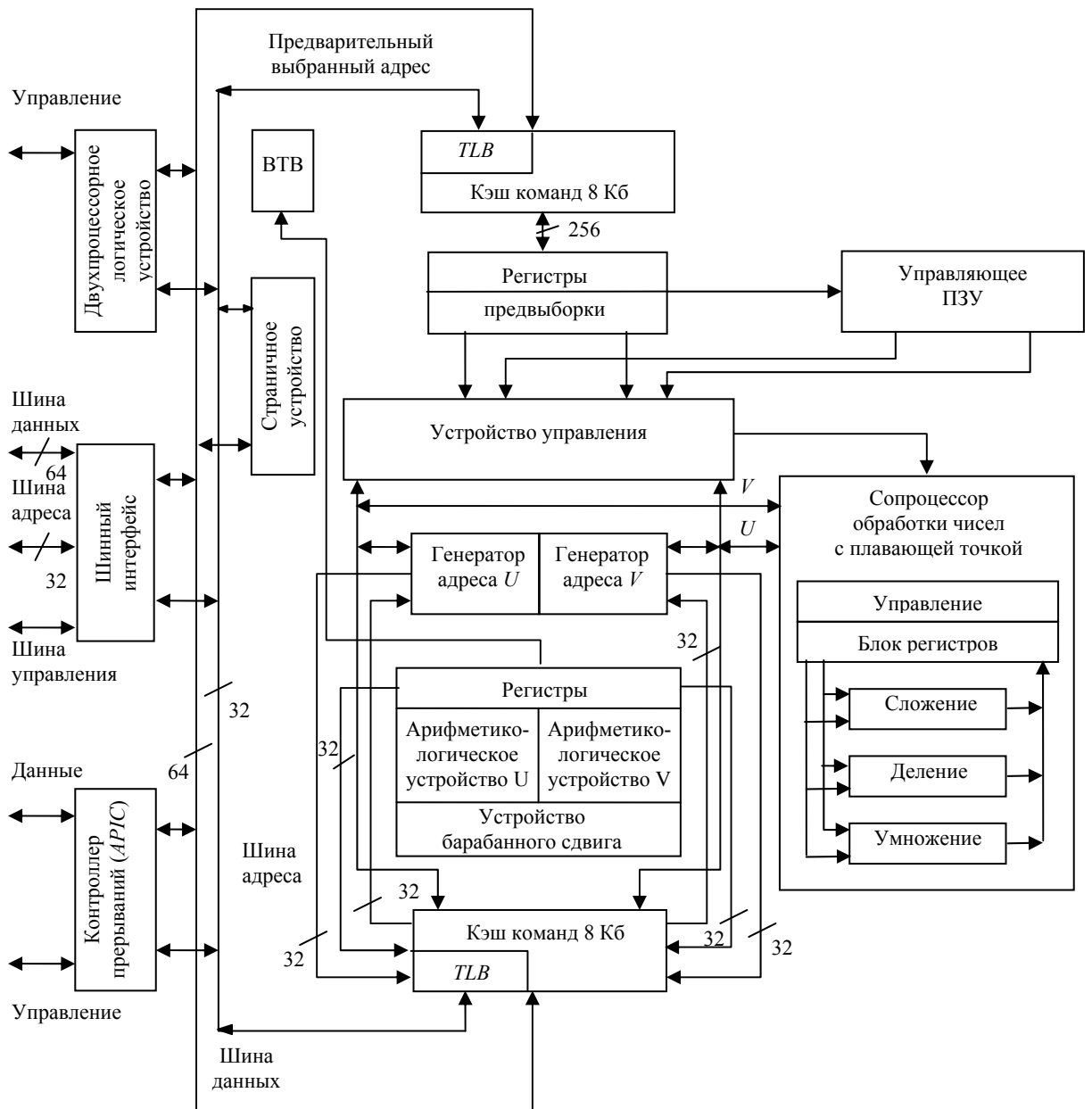


Рис.2.29 Структурная схема микропроцессора *PENTIUM*

Микропроцессор *PENTIUM* реализует упреждение переходов, используя память ВТВ (*Branch Target Buffer*) и два регистра предварительной выборки. Один регистр применяется для предварительной выборки команд, в предположении, что перехода нет, другой выполняет предвыборку команд в память, используя содержимое ВТВ.

Эффективность конвейера резко снижается из-за необходимости его перезагрузки при выполнении условных ветвлений. В этом случае требуется произвести очистку памяти на всех предыдущих этапах работы конвейера и выбрать команду из другой ветви программы. Чтобы сократить поте-



ри времени, связанные с перезагрузкой конвейера, используется блок предсказания ветвлений ВТВ. Его основной частью является специальная память ветвлений, в которой хранятся адреса ранее выполненных переходов. Кроме того, в этой памяти содержатся биты, хранящие информацию о предыстории ветвлений, которые указывают, выполнялся ли переход при предыдущих выборках данной команды. При поступлении очередной команды условного перехода указанный в ней адрес сравнивается с содержимым специальной памяти ветвлений. Если этот адрес не содержится в этой памяти, то предсказывается отсутствие ветвления. В этом случае продолжается выборка и декодирование команд, следующих за командой перехода. При совпадении указанного в команде адреса перехода с каким-либо из адресов, хранящихся в специальной памяти ветвлений, производится анализ информации о предыстории ветвлений. В процессе анализа определяется направление ветвления реализуемое наиболее часто, а также выявляются чередующиеся переходы. Если предсказывается выполнение ветвления, то выбирается и загружается в конвейер команда, размещённая по предсказанному адресу. В специальной памяти ветвлений хранится до 256 результатов переходов, что позволяет выполнять правильное предсказание с вероятностью, равной не менее 0.2.

В регистре признаков имеются дополнительные биты:

- *ID (Id Flag)* — бит доступности команды идентификации *CPUID*;
- *VIP (Virtual Interrupt Pending)* — виртуальный запрос прерывания;
- *VIF (Virtual Interrupt Flag)* — виртуальное прерывание;
- *AC (Alignment Check)* — бит контроля выравнивания.

При выполнении программ на уровне привилегий  $PL = 3$  в случае обращения к операнду, не выровненному по соответствующей границе, при установленном контроле выравнивания происходит исключение с нулевым кодом ошибки. На уровнях привилегий  $PL = 0, 1, 2$  контроль выравнивания не производится.

Микропроцессор *Pentium* имеет 12 тестовых регистров *TR1...TR12*. Они входят в группу модельно-специфических регистров *MSR (Model-Specific Register)*:

- *TR3* — регистр данных внутренней кэш-памяти;
- *TR4* — тестовый регистр состояния кэш-памяти;
- *TR5* — управляющий регистр тестирования кэш-памяти.

В микропроцессор введены средства энергосбережения, работающие на двух уровнях: на уровне микропроцессора и на уровне микропроцессорной системы. При выполнении задач, не требующих выполнения интенсивных вычислений, управление энергопотреблением предусматривает перевод микропроцессора в режим с более низкой тактовой частотой и пониженным напряжением питания. Возможна полная остановка микропроцессора - спящий (*SL*) режим. Режим управления системой *SMM* (*System Management Mode*) контролирует энергопотребление во всей микропроцессорной системе, включая периферийные устройства. Этот режим позволяет микропроцессору замедлять функционирование, замедлять или полностью прекращать функционирование отдельных компонентов системы, экономить потребление электроэнергии.

Модификацией микропроцессора *Pentium* является микропроцессор *Pentium Pro*, имеющий следующие отличия.

- Применено динамическое исполнение команд.
- Используется архитектура двойной независимой шины, повышающая суммарную пропускную способность шин.
- В микропроцессоре размещено внутреннее статическое ОЗУ (кэш-память) 2-го уровня большого объема.
- В систему команд введены команды условной пересылки данных, позволяющие сократить число условных переходов. При этом повышается предсказуемость кода и эффективность использования конвейера.
- Для повышения достоверности передачи данных шина данных поддерживает *ECC*-контроль (*Error checking and correcting*). Повышенная устойчивость к сбоям обеспечивается средствами анализа и восстановления после отказов. Двухпроцессорные системы допускают конфигурацию избыточного функционального контроля (*FRC*).
- Архитектура рассчитана на многопроцессорные системы. Интерфейс обеспечивает объединение до четырех микропроцессоров на одной шине с помощью внутренних схем арбитража в симметричную многопроцессорную систему.

Микропроцессор *Pentium Pro* имеет тактовую частоту 150/166/200 МГц при напряжении питания 3,1–3,3 В.

Динамическое выполнение команд позволяет одновременно разбивать составные длинные команды на множество наборов микрокоманд. В микропроцессоре обрабатывается основная часть команд, а остальные ставятся в очередь на выполнение до тех пор, пока не поступят необходимые дан-

ные. Используя предсказания ветвлений, анализ прохождения данных, исполнение команд с опережением и переупорядочение команд, микропроцессор выполняет команды, результат выполнения которых не зависит от результатов выполнения предыдущих команд. После этого происходит восстановление последовательности выдачи результатов в память и порты.

Применение динамического исполнения команд существенно повышает частоту запросов микропроцессора к шине за данными и командами, поскольку микропроцессор одновременно обрабатывает несколько команд. Микропроцессор использует архитектуру двойной независимой шины (*Dual Independent Bus*). Одна из этих шин используется только для связи с кристаллом вторичной кэш-памяти, расположенной в том же корпусе микропроцессора. Эта шина является локальной, что позволяет использовать ее на частоте работы микропроцессора. Значительный объем вторичной кэш-памяти позволяет удовлетворять большинство запросов к памяти глубоко локально. При этом коэффициент загрузки шины достигает 90%. Вторая шина процессорного кристалла выходит на внешние выводы микропроцессора, Она является системной шиной микропроцессора *Pentium Pro*. Эта шина работает на внешней частоте 66,66 МГц независимо от внутренней шины. Снижение нагрузки на внешнюю шину позволяет эффективно использовать многопроцессорную архитектуру.

Максимальная пропускная способность для микропроцессоров, использующих архитектуру двойной независимой шины, рассчитывается как сумма пропускной способности системной шины (533 Мбит/с) и пропускной способности шины вторичной кэш-памяти.

Можно выделить три составные части конвейера микропроцессора *Pentium Pro*:

- Упорядоченный препроцессор.
- Микропроцессор с неупорядоченным исполнением команд.
- Упорядочивающее устройство.

На рис. 2.30 показана схема конвейера, содержащая:

- Блоки предсказания ветвлений (*BTB0, BTB1*);
- Блок предвыборки команд (*IFU0, IFU1, IFU2*);
- Декодеры команд (*ID0, ID1*);
- Таблицу переименования (псевдонимов) регистров (*RAT*);
- Переупорядочивающий блок чтения (*ROB rd*);
- Резервирующая станция (*RS*);

- Порты микропроцессора с неупорядоченным исполнением команд Порт0...Порт4;
- Переупорядочивающий блок записи (*ROB wb*);
- Файл регистров выгрузки (*RRF*).

Микропроцессор *Pentium Pro* временно хранит каждый запрос на запись в память в блоке записи. Это позволяет продолжать исполнение команд, не дожидаясь завершения операций записи в память и (или) кэш-память.

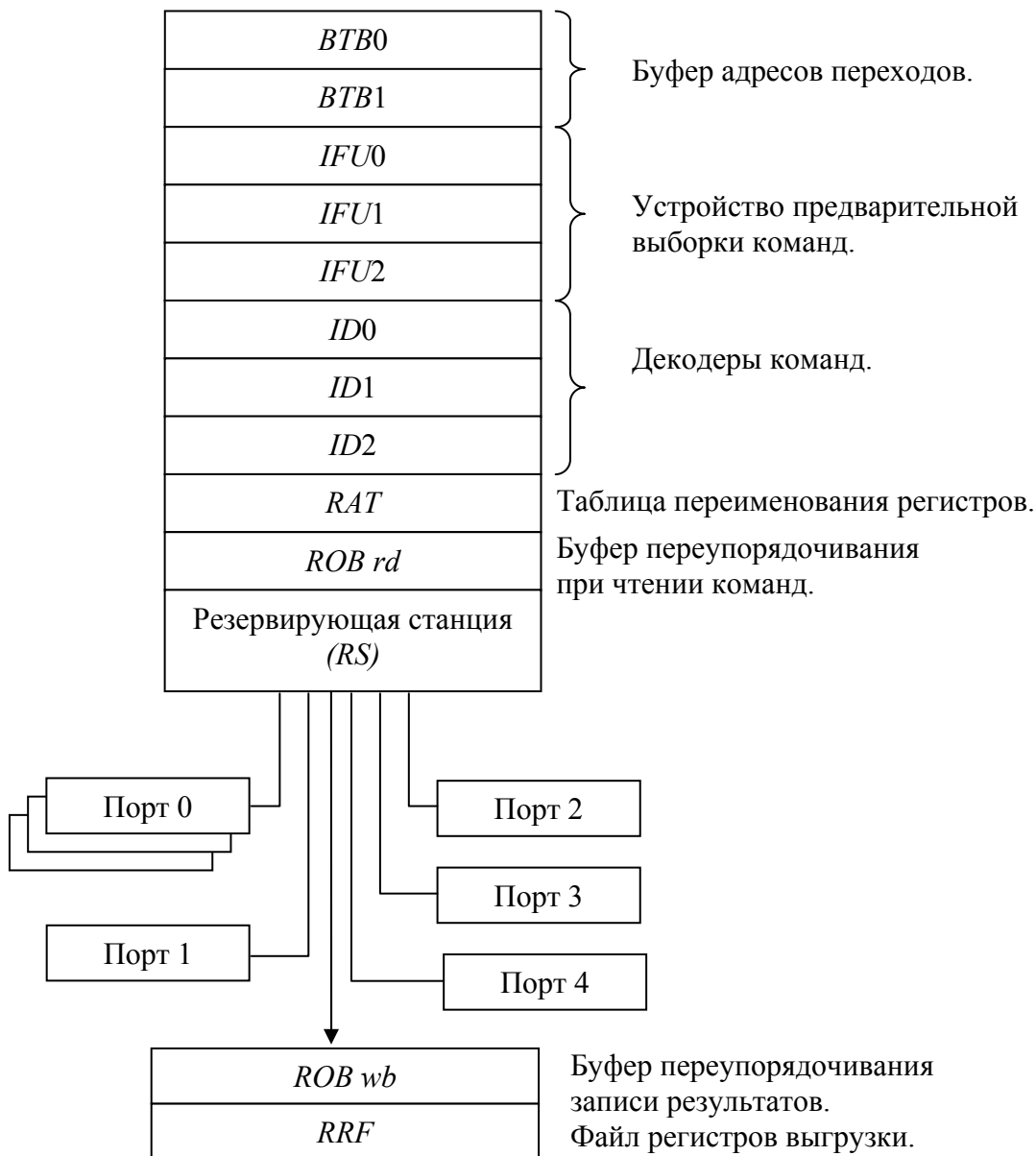


Рис.2.30 Конвейер микропроцессора *Pentium Pro*

Запись производится в память по правилам, предписанным программными кодами. Микропроцессор поддерживает логическое соответствие порядка физических операций чтения и записи памяти их порядку в программных кодах. Операции чтения могут пропускать буферизированные записи, а запись в память всегда идет в порядке, предписанном программой.

Микропроцессоры *Pentium MMX* используют технологию *MMX*, представляющую собой расширение набора команд с использованием техники *SIMD* (*Single Instruction, Multiple Data* — одна команда на множество данных) для ускорения мультимедийных и коммуникационных программ за счет параллельной обработки. Набор *MMX* - команд добавляет 57 новых кодов команд и новый тип 64-битных данных. Этот новый тип данных имеет упакованные целочисленные значения во время выполнения операций *MMX*. Дополнительно введены восемь новых 64-битных регистров с именами *MMX0-MMX7*.

Микропроцессоры *Pentium MMX* по сравнению с микропроцессорами *Pentium* имеют увеличенное число ступеней в конвейере. Их интеграция с целочисленным конвейером похожа на конвейер сопроцессора обработки чисел с плавающей точкой *FPU*. В целочисленный конвейер после ступени *PF* введена дополнительная ступень *F*, на которой производится синтаксический разбор команд.

Суперскалярная архитектура микропроцессоров *Pentium MMX* позволяет выполнять команды парами в пределах ограничений на число исполнительных устройств и их взаимосвязей.

- Арифметико-логическое устройство (АЛУ) выполняет арифметические (сложение и вычитание) и логические операции. Наличие двух АЛУ позволяет выполнять эти команды парами на обоих конвейерах.

- Перемножитель выполняет все операции умножения за три цикла. Он конвейеризирован, что позволяет получать результат очередного умножения в каждом такте. Микропроцессор имеет один перемножитель, так что операции умножения не могут выполняться парами. Однако они могут выполняться в паре с любыми другими командами. Перемножение может выполняться как на *U*-, так и на *V*-конвейерах.

- Сдвиговое устройство выполняет все операции сдвигов, упаковки и распаковки. Данные команды (на любом конвейере) могут выполняться в паре только с другими командами.

- Команды *MMX*, требующие доступа к памяти или обычным регистрам, могут исполняться только на *U*-конвейере и не могут выполняться в паре с другими командами.

#### **2.2.4 Микропроцессор Pentium II**

Микропроцессор *Pentium II* основан на архитектуре микропроцессора *Pentium Pro*, в которую добавлено несколько исполнительных устройств для использования технологии *MMX*. Он является сочетанием архитектуры микропроцессора *Pentium Pro* с технологией *MMX*. В структуре микропроцессора *Pentium II Port0* содержится арифметико-логическое устройство с технологией *MMX* и перемножитель *MMX*, а *Port1* представляет собой арифметико-логическое устройство с технологией *MMX* и устройство сдвигов *MMX*. Как и микропроцессор *Pentium Pro*, микропроцессор *Pentium II* использует архитектуру двойной независимой шины, повышающую пропускную способность и производительность.

Размер первичной кэш-памяти микропроцессоров с поддержкой технологии *MMX* увеличен до 256 Кбайт. Кэш-память данных также разбита на восемь чередующихся банков. Внутренняя кэш-память одновременно доступна для двух конвейеров, если они обращаются к разным банкам. Кэш-память данных допускает одновременное выполнение операций записи и считывания, адресованных к разным банкам. Если оба запроса (на чтение и запись) имеют один и тот же адрес, они выполняются за один цикл, минуя кэш-память.

Блок ветвлений *ВТВ* микропроцессоров *Pentium MMX* и *Pentium II* функционально идентичен блоку ветвлений *Pentium Pro*, но имеет характерную особенность. Если последние байты двух соседних команд попадают в одно выровненное двойное слово, возможно ошибочное предсказание ветвлений. Такая ситуация возможна при короткой второй команде ветвления. Этого можно избежать, например, применением 16-битного смещения во второй команде.

Снятие вторичной кэш-памяти с кристалла микропроцессора позволяет использовать стандартные наборы кристаллов внутренней памяти, памяти этикеток и контроллера кэш-памяти, применяя новые модели по мере их появления. Возможно использование вторичной кэш-памяти с *ЕСС*. В то же время сохраняется независимость шины вторичной внутренней памяти, которая работает на половинной частоте микропроцессора.

Микропроцессор предназначен для работы с 32-разрядными приложениями и операционными системами. Первичная кэш-память содержит 16

Кбайт данных и 16 Кбайт команд. Вторичная кэш-память имеет объем 512 Кбит.

Внутренняя тактовая частота составляет 233/266/300/450 МГц. Внешняя тактовая частота равна 66,66 МГц.

Системная шина *Pentium Pro* и *Pentium II* позволяет без дополнительных устройств объединять до четырех микропроцессоров.

Сигналы системной шины объединяются в группы запросов (*Request*) и ответов (*Response*). Каждый из микропроцессоров, подключенный к этой шине, до инициализации запроса через механизм арбитража должен получить право на использование шины. Запрос проходит за два смежных такта. В первом такте передается адрес, тип обращения (чтение-запись памяти или ввода-вывода). Во втором такте передается уникальный идентификатор транзакции, длина запроса, разрешенные байты шины и т. п. Через три такта после запроса проверяется состояние ошибки для защиты от ошибок передачи или нарушений протокола. Любая обнаруженная ошибка вызывает повторение запроса, а вторая ошибка для того же запроса вызывает исключение. На шине одновременно может присутствовать несколько запросов и ответов.

Микропроцессор *Pentium III* является совместимым с микропроцессором *Pentium II*. Он имеет более высокую тактовую частоту, улучшенную работу при параллельных вычислениях и более удлиненный конвейер. Слишком высокая цена практически не позволила ему приобрести наличие кэш-памяти второго уровня объемом 912 Кбайт. Микропроцессор *Pentium III* специально создан для портативных компьютеров. Он имеет пониженное напряжение питания, в некоторых режимах не превышающее 0,95 В и рассеиваемую мощность менее 0,5 Вт. Тактовая частота работы составляет 1 ГГц. Большая интегральная схема микропроцессора имеет 28 млн. транзисторов на площади кристалла 128 кв. мм.

#### **2.2.5 Микропроцессор Pentium 4**

Микропроцессор *Pentium 4* построен на основе архитектуры *Intel NetBurst*, в которой используются следующие технологии.

- Гиперконвейерная технология (*Hyper Pipelined Technology*).
- Микропроцессор быстрого выполнения (*Rapid Execution Engine*), повышающий производительность при работе с целочисленными данными за счет работы на удвоенной тактовой частоте по сравнению с частотой основного микропроцессора.

- Внутренняя кэш-память с отслеживанием выполнения команд (*Execution Trace Cache*), в которой хранятся уже декодированные команды, что устраняет задержку при анализе повторно исполняемых участков кода.

Микропроцессор *Pentium 4* содержит от 42 до 55 млн. транзисторов на кристалле, снабжен внутренней статической памятью объемом 256 Кбайт и имеет 144 новые команды, ускоряющие обработку блоков данных с плавающей точкой. Тактовая частота системной шины составляет 400 МГц при пропускной способности до 3,2 Гбайт/с.

Архитектура микропроцессора определяет реализацию его внутренней структуры, принцип выполнения поступающих команд, способы размещения и обработки данных. Она ориентирована на эффективную работу с Интернет-приложениями. Характерными чертами этой архитектуры являются:

- Структура с разделением потоков команд и данных.
- Архитектура, обеспечивающая одновременное выполнение нескольких команд в параллельно работающих исполнительных устройствах.
- Динамическое изменение последовательности команд (выполнение команд с опережением).
- Конвейерное исполнение команд.
- Предсказание направления ветвлений.

На рис.2.31 приведены обобщенная структурная схема микропроцессора *Pentium 4*.

Обмен по системной шине осуществляется с помощью 64-разрядной двунаправленной шины данных и 41-разрядной шины адреса (33 адресных линии и 8 линий выбора байтов), обеспечивающей адресацию до 64 Гбайт внешних запоминающих устройств.

Дешифратор команд совместно с кэш-памятью микрокоманд формирует последовательность микрокоманд, обеспечивающих выполнение поступивших команд. Декодированные команды загружаются в кэш-память микрокоманд. Кэш-память может хранить до 12000 микрокоманд. При поступлении очередной команды блок трассировки и предсказания ветвлений выбирает из этой кэш-памяти необходимые микрокоманды.

Если в потоке команд оказывается команда условного перехода (ветвления программы), то включается механизм предсказания ветвления, который формирует адрес следующей выбираемой команды до того, как будет определено условие выполнения перехода.



После формирования потоков микрокоманд производится выбор регистров, необходимых для выполнения декодированных команд. Эта процедура реализуется блоком распределения регистров, который выделяет для каждого указанного в команде логического регистра (регистра целочисленных операндов *EAX*, *ECX*, регистра операндов с плавающей точкой *ST0-ST7* или регистра блоков *MMX*, *SSE* рис.2.31) один из 128 физических регистров, входящих в состав блоков регистров замещения (БРЗ). Это позволяет выполнять команды, использующие одни и те же логические регистры, одновременно или с изменением их последовательности.

Выбранные микрокоманды размещаются в очереди микрокоманд. В ней содержатся микрокоманды, реализующие выполнение 126 поступивших и декодированных команд, которые затем направляются в исполнительные устройства по мере готовности операндов. Значительное увеличение числа команд, стоящих в очереди, позволяет более эффективно организовать поток их исполнения, изменяя последовательность выполнения команд и выделяя команды, которые могут выполняться параллельно. Эти функции реализует блок распределения микрокоманд путем выбора их из очереди не в порядке поступления, а по мере готовности соответствующих операндов и исполнительных устройств.

В микропроцессоре *Pentium 4* реализуется одновременное выполнение нескольких микрокоманд (команд) в параллельно работающих исполнительных устройствах. Таким образом естественный порядок следования команд нарушается, чтобы обеспечить более полную загрузку параллельно включенных исполнительных устройств и повысить производительность микропроцессора.

Суперскалярная архитектура реализуется путём организации исполнительного блока микропроцессора в виде ряда параллельно работающих блоков. Арифметико-логические устройства *ALU* производят обработку целочисленных операндов, которые поступают из заданных регистров БРЗ. В эти же регистры заносится и результат выполнения операции. При этом проверяются также условия ветвления для команд условных переходов и формируются сигналы перезагрузки конвейера команд в случае неправильно предсказанного ветвления. Исполнительный блок работает с повышенной скоростью выполнения операций. Например, микрокоманда сложения целочисленных операндов при тактовой частоте процессора 1,5 МГц выполняется всего за 0,36 нс.

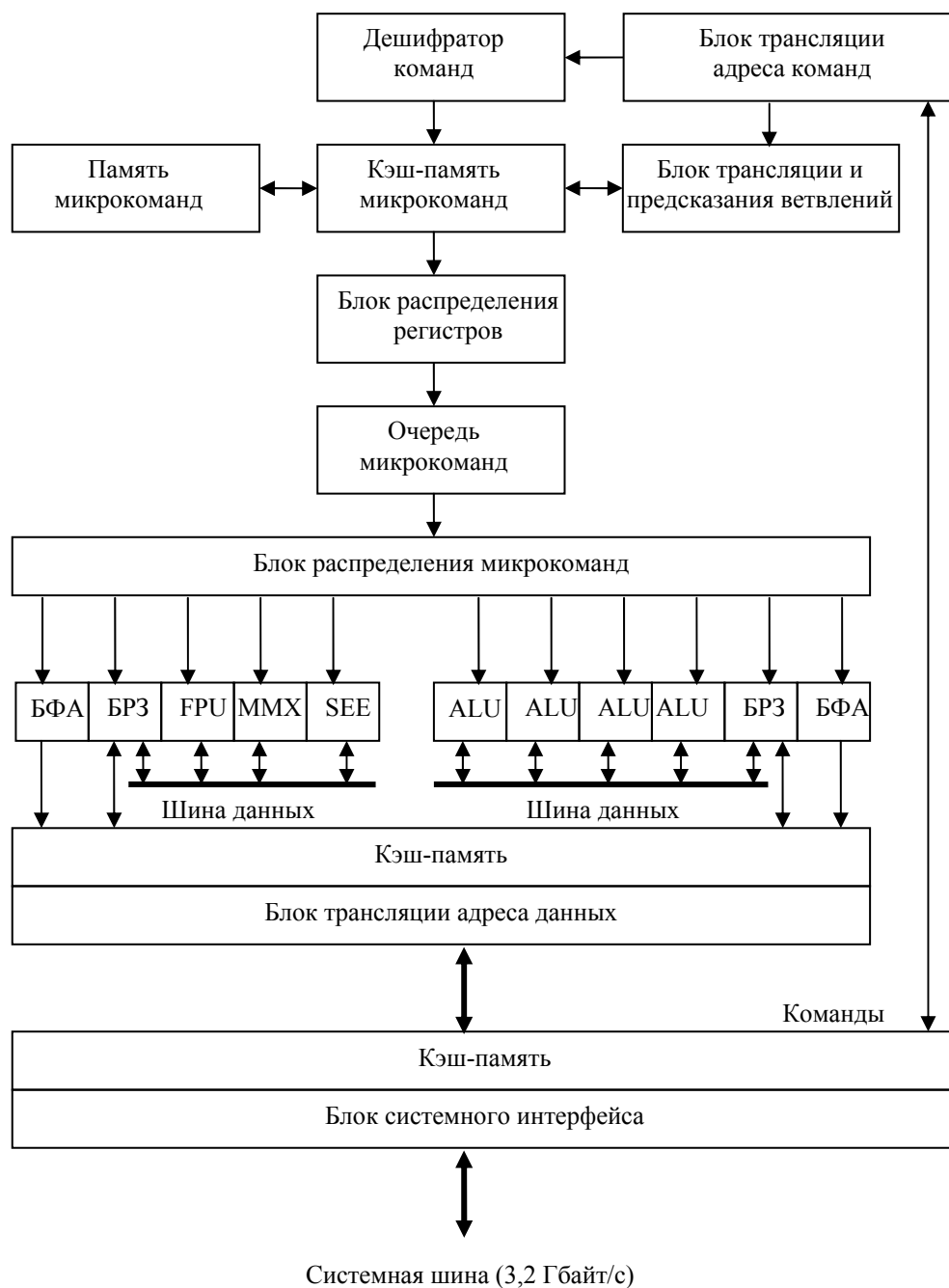


Рис. 2.31 Обобщенная структурная схема микропроцессора *Pentium 4*

Суперскалярная архитектура реализуется путём организации исполнительного блока микропроцессора в виде ряда параллельно работающих блоков. Арифметико-логические устройства *ALU* производят обработку целочисленных операндов, которые поступают из заданных регистров БРЗ. В эти же регистры заносится и результат выполнения операции. При этом проверяются также условия ветвления для команд условных переходов и формируются сигналы перезагрузки конвейера команд в случае непра-

вильно предсказанного ветвления. Исполнительный блок работает с повышенной скоростью выполнения операций. Например, микрокоманда сложения целочисленных операндов при тактовой частоте процессора 1,5 МГц выполняется всего за 0,36 нс.

Адреса операндов вычисляются блоком формирования адреса (БФА), который реализует интерфейс с кэш-памятью данных первого уровня ёмкостью 8 Кбайт. В соответствии с заданными в декодированных командах способами адресации формируются 48 адресов для загрузки операндов из памяти в регистр БРЗ и 24 адреса для записи из регистра в память, блок формирования адреса формирует адреса операндов для команд, которые ещё не поступили на выполнение. При обращении к памяти этот блок одновременно выдаёт адреса двух операндов: первый служит для загрузки операнда в заданный регистр БРЗ (рис.2.31), второй — для пересылки результата из БРЗ в память. Таким образом реализуется процедура предварительного чтения данных для последующей их обработки в исполнительных блоках.

Аналогичным образом организуется параллельная работа блоков *SSE*, *FPU*, *MMX*, которые используют отдельный набор регистров и блок формирования адресов операндов. При считывании операнда из памяти производится обращение к кэш-памяти данных первого уровня, которая имеет отдельные порты для чтения и записи. За один такт производится считывание операндов для двух команд. Время обращения кэш-памяти данных составляет 1,42 нс при тактовой частоте 1,5 ГГц.

При формировании адресов обеспечивается обращение к заданному сегменту запоминающего устройства. Каждый сегмент делится на страницы, размещаемые в различных местах адресного пространства. Блоки трансляции адреса обеспечивают формирование физических адресов команд и данных при использовании страничной организации памяти. Для сокращения времени трансляции используется внутренняя буферная память, которая хранит базовые адреса наиболее часто используемых страниц.

В микропроцессоре *Pentium 4* используется гиперконвейерная технология выполнения команд, при которой число ступеней конвейера достигает 20. Таким образом одновременно в процессе выполнения может находиться до 20 команд, находящихся на разных ступенях.

Для сокращения потерь времени, связанных с перезагрузкой конвейера, используется блок трассировки и предсказания ветвлений. Его основ-

ной частью является буфер адресов ветвлений, в котором хранятся 4092 адреса ранее выполненных переходов. В этом буфере содержатся биты, хранящие информацию о предыдущих ветвлениях, которые указывают, выполнялся ли переход при предыдущих выборках данной команды. При поступлении очередной команды условного перехода указанный в ней адрес сравнивается с содержимым в блоке трассировки и предсказания ветвлений. Если адрес не содержится в этом блоке, то предсказывается отсутствие ветвления. Тогда продолжается выборка и декодирование команд, следующих за командой перехода. При совпадении указанного в команде адреса перехода с каким-либо из адресов, хранящихся в блоке трассировки и предсказания ветвлений, производится анализ информации о предыдущих ветвлениях. В процессе анализа определяется чаще всего реализуемое направление ветвления, а также выявляются чередующиеся переходы. Если предсказывается выполнение ветвления, то выбирается и загружается в конвейер команда, размещённая по предсказанному адресу.

Реализованное в микропроцессоре *Pentium 4* значительное изменение микроархитектуры и повышение производительности потребовали введения дополнительных аппаратных средств. На кристалле микропроцессора располагаются 42 млн. транзисторов. Для изготовления большой интегральной схемы микропроцессора *Pentium 4* используется КМОП-технология с разрешающей способностью 0,18 мкм.

Практическая реализация потенциальных возможностей микропроцессора *Pentium 4* обеспечивается при использовании набора специализированных микросхем, необходимых для построения на его основе микропроцессорных систем различного назначения. Для реализации таких систем компания выпускается набор микросхем *Chipset 850*, в который входят:

- Контроллер-концентратор памяти *MCH (Memory Controller Hub)* типа *Intel 82850*;
- Контроллер-концентратор для устройств ввода/вывода *ICH2 (I/O Controller Hub)* типа *Intel 82801BA*;
- Контроллер микрокода *FWH (FirmWare Hub)* типа *Intel 82802AB*.

Основной особенностью таких систем является использование новой системной шины *FSB*, обеспечивающей обмен данными со скоростью 3,2 Гбайт/С. Такая скорость реализуется путём применения нового типа сверхбыстродействующей двухканальной памяти *RDRAM* и контроллера-

концентратора *MCH*, обеспечивающего 4 канала обмена с памятью этого типа.

Контроллер *MCH* выполняет обмен с оперативной памятью типа *Direct RAMBUS* ёмкостью от 128 Мбайт до 2 Гбайт с помощью сдвоенных каналов. Память реализуется на основе микросхем быстродействующей двухканальной *RDRAM*-памяти типа *PC800* или *PC600*, выпускаемых компанией *RAMBUS*. Таким образом общий доступ к оперативной памяти осуществляется с использованием четырёх каналов обмена.

Контроллер *ICH2* служит для подключения различных внешних устройств с использованием интерфейса *ULTRA ATA/66/100*. Этот интерфейс реализует обмен с жёстким диском со скоростью 66 или 100 Мбайт/с. *ICH2* также обеспечивает прямой доступ внешних устройств к памяти со скоростью 33 Мбайт/с при помощи интерфейса *ULTRA DMA/33*. Контроллер служит для подключения последовательных портов с шиной *USB*, связи с локальной сетью *Ethernet* и параллельного обмена по шине *PCI*.

Основой микропроцессора *Pentium 4* являются два параллельных 32-битных конвейера увеличенной длины. Это позволяет выполнять основные арифметические команды (типа *SUB*, *ADD*, *AND*, *OR*) за 0,5 такта. К этим "быстрым" арифметико-логическим устройствам добавлено одно арифметико-логическое устройство, исполняющее "медленные" команды. В микропроцессоре улучшена система динамического исполнения кода и расширенного предсказания условных переходов. Для ускорения очистки конвейеров от результатов неверно предсказанных переходов используется расширенная буферная память размером 4 Кбайт. Для хранения данных используется отдельное внутреннее статическое ОЗУ (кэш-память) объемом 8 Кбайт.

В микропроцессоре используется универсальная кэш-память объемом 256 Кбайт с ускоренной передачей сигналов на частоте работы микропроцессора, ассоциативная на 8 зон, что увеличивает процент попаданий в кэш-память при типичных операциях с данными. Эта кэш-память имеет встроенный код коррекции ошибок (ЕСС).

Модуль вычислений с плавающей запятой и потоковый модуль оптимизированы для работы с видео и аудио потоками данных, выполнения задач шифрования и специализированных вычислений.

Система динамического исполнения кода базируется на методике трансляции "внешних" команд микропроцессора в цепочки микрокоманд, которые складываются в специальную буферную память и затем исполняются.

ются параллельно несколькими модулями в порядке готовности данных. Результаты исполнения микрокоманд, в свою очередь, тоже оказываются в специальной буферной памяти, откуда их извлекает модуль-сборщик и ставит их в соответствие с результатами исполнения конкретных "внешних" команд. Недостатком этого алгоритма работы является наличие условных ветвлений, поскольку до вычисления нужных для ветвления данных имеется неопределенность, по какой ветви программы пойдут вычисления. Микропроцессор исполняет обе ветви программы, а после проверки условия холостая ветвь отбрасывается.

Путем оптимизации структуры микропроцессора *Pentium 4* удалось преобразовать универсальное устройство в специализированный вычислитель, достигающий высшей производительности на потоковых задачах таких, как кодирование, обработка видеоданных, но более медленно исполняющий офисные задачи.

Внутренняя кэш-память с улучшенной передачей данных (*Advanced Transfer Cache*) обеспечивает более скоростной канал данных между кэш-памятью и микропроцессором. Внутренняя кэш-память соединена с интерфейсом, передающим данные на каждый такт частоты работы микропроцессора. В результате микропроцессор *Pentium 4* с тактовой частотой 1,5 ГГц имеет скорость обмена с кэш-памятью, равную 48 Гбит/сек. Как следствие, многократно увеличивается производительность вычислительных циклов, данные которых укладываются в кэш-память.

Микропроцессор имеет 5 выводов, определяющих его напряжение питания в диапазоне от 1.1 до 1.85 В с шагом 0.025 В. Сигнальные выводы микропроцессора работают с уровнями сигнала *AGTL+*. Потребляемый микропроцессором ток (в основном по цепям питания ядра) составляет от 38 А при рассеиваемой мощности соответственно от 49 до 54,7 Ватт.

Дальнейшее развитие архитектуры микропроцессора *Pentium 4* нашло в микропроцессорах *Pentium M*, *Pentium 4HT* и *Pentium 4EE*.

**Микропроцессоры *Pentium M*** производятся с использованием 0,13 мкм технологии и содержат 77 млн. транзисторов. Микропроцессоры оснащены 400 МГц системной шиной, экономичной внутренней кэш-памятью объемом 1 Мбайт с функцией отключения неиспользуемых блоков, а также специальными функциями для снижения общего энергопотребления. Микропроцессор поддерживает усовершенствованную технологию *Intel SpeedStep* с возможностью выбора нескольких рабочих частот и напряжений питания, а также потоковые *SIMD*-расширения *SSE2*.

**Микропроцессор *Pentium 4HT*** изготавливается по 0,13 мкм технологии с длиной затвора транзистора 0,065 мкм. Этот микропроцессор имеет технологию *HyperThreading*, которая позволяет достичь заметного увеличения быстродействия при одновременной работе двух «тяжелых» приложений высокой загрузки операционной системы.

**Микропроцессор *Pentium 4EE (Extreme Edition)*** помимо технологии *HyperThreading*, системной шины 800 МГц и тактовой частоты 3,2 ГГц, имеет кэш-память третьего уровня объемом 2 Мбайт, расположенную на самом кристалле и работающую с тактовой частотой микропроцессора. Она сосуществует с внутренней кэш-памятью объемом 512 Кбайт, которая в этом случае позволяет считывать данные из кэш-памяти третьего уровня, а не из системной памяти. Микропроцессор *Pentium 4EE* с рабочей частотой 3,2 ГГц полностью совместим со всеми двухканальными микропроцессорами. Микропроцессор *Pentium 4EE* имеет 170 миллионов транзисторов на площади от 131 до 230 кв.мм. Тепловыделение микропроцессора составляет 93,9 Вт. Максимальный ток потребления равен 72,3 ампер. В данном случае применение стандартных системы охлаждения оказывается недостаточным и поэтому используется встроенная система терморегуляции, размещенная на самом кристалле.

### **2.3 Основные тенденции развития универсальных микропроцессоров**

Основные тенденции совершенствования универсальных микропроцессоров заключаются в увеличении их быстродействия, повышении производительности, расширении программных приложений, повышении степени универсальности.

Повышение тактовой частоты микропроцессора достигается за счет:

- использования более совершенных технологий изготовления интегральных микросхем, содержащих большое число слоев металлизации и плотную компоновку функциональных блоков;
- уменьшения размеров транзисторов и увеличения их числа в кристалле (миллионы и десятки миллионов). Это приводит к увеличению рассеиваемой мощности, что вынуждает снижать напряжение питания до 1.1 – 3.3 В.

Рост числа слоев металлизации позволяет уменьшить длину межсоединений на кристалле и тем самым повысить тактовую частоту работы микропроцессора. Тактовые частоты работы микропроцессоров возрастают до 2–3,2 ГГц. Увеличение объема и пропускной способности подсистем

мы памяти. Это достигается, как за счет увеличения разрядности шины адреса, так и использованием структурных решений: создание внутренних статических ОЗУ (кэш-памяти) нескольких уровней, объем каждого из которых, может наращиваться, введение дополнительных системных шин для связи микропроцессора с внутренним статическим ОЗУ.

Использование микропроцессоров с сокращенным набором команд (*RISC*-процессоров). Развитие архитектуры микропроцессоров с сокращенным набором команд базируется на совершенствовании технологии изготовления полупроводниковых БИС и на стремлении реализовать минимальными аппаратными средствами устройства с достаточно высокими вычислительными возможностями.

Системный подход с оптимизацией структурных решений при реализации технологических возможностей СБИС заложен при проектировании *RISC* – процессоров. Разбивая сложные команды на элементарные (загрузить, сравнить, сложить, ветвление и др.) и сокращая время их выполнения, можно добиться роста производительности микропроцессора. Основными принципами построения *RISC* – процессоров являются следующие:

- Небольшой набор простых команд.
- Выполнение всех команд за 1 машинный такт.
- Аппаратная реализация управления.
- Использование языков высокого уровня и их аппаратная поддержка.
- Интенсивное использование регистров общего назначения.
- Максимальная загрузка конвейера при выполнении команд.

Формат всех команд составляет 32 разряда при их общем числе равным 60. Все вычислительные операции выполняются над данными, находящимися в регистрах общего назначения. Набор команд декодируется дешифратором, что упрощает структуру устройства управления, а также позволяет увеличить тактовую частоту, разместить на кристалле внутреннее статическое ОЗУ команд и данных большой емкости и устройство управления памятью. Для реализации сложных команд используется дополнительный арифметический сопроцессор. Использование *RISC*-ядра позволяет перейти к микропроцессорам с многими системами команд, каждая из которых транслируется своим аппаратным транслятором в набор операций *RISC*-ядра микропроцессора.



**Увеличение количества параллельно работающих исполнительных устройств** позволяет существенно увеличить быстродействие микропроцессора, особенно работающего с большим числом внешних устройств.

**Построение многопроцессорных систем**. Одним из путей повышения производительности является построение многопроцессорных систем. При использовании для этого традиционных микропроцессоров имеются трудности, обуславливаемые проблемой организации совместной эффективной работы большого числа микропроцессорных узлов. Избежать их возможно путем создания СБИС, которую можно использовать в качестве стандартного элемента при построении сверхвысокопроизводительных многопроцессорных систем, содержащих большое (порядка 100...1000) число процессорных элементов.

Одной из таких СБИС является транспьютер (*transputer*). Его особенность заключается в реализации программ, написанных на специально разработанном языке параллельного программирования OCCAM. Функционирование вычислительной системы на этом языке описывается в виде совокупности параллельных процессов, взаимодействующих между собой с помощью обмена сообщениями.

Транспьютеры используют для связи между микропроцессорами последовательные каналы<sup>1</sup>, являющиеся аппаратной реализацией логических каналов в языке OCCAM. В семейство транспьютерных СБИС, кроме собственно транспьютера, входят транспьютер-связной адаптер, транспьютер-контроллер массовой памяти и набор внешних запоминающих устройств.

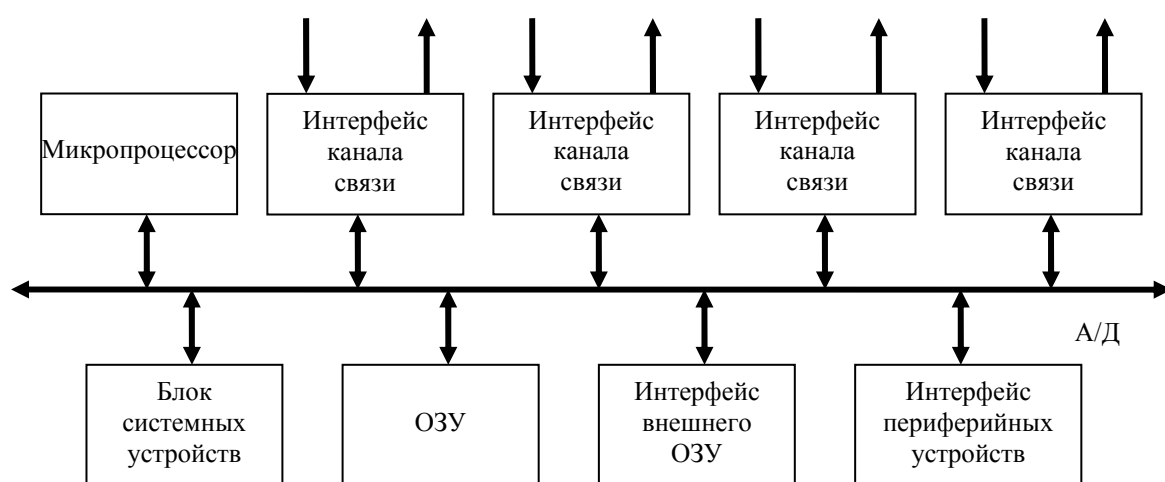


Рис. 2.32 Структурная схема транспьютера *IMS T424*

<sup>1</sup> Получившие в литературе название линки.

Транспьютер можно использовать как в качестве микропроцессора для построения высокопроизводительных параллельных многопроцессорных систем, так и в качестве автономной ЭВМ.

Структурная схема транспьютера *IMS T424* (рис.2.32) содержит 32-битовый микропроцессор, внутреннее ОЗУ емкостью 4 Кбайт, блок системных устройств, четыре блока интерфейса каналов связи, блок интерфейса внешнего ЗУ, блок интерфейса периферийных устройств.

В основу построения микропроцессора транспьютера положены принципы архитектуры с сокращенным набором команд (*RISC*-архитектура). Все команды имеют длину 1 байт и просты для декодирования, что обеспечивает высокую производительность. Микропроцессор осуществляет непосредственную реализацию модели параллельных вычислений и связи между процессами, принятой в языке *ОССАМ*. В нем имеется планировщик процессов, позволяющий совместно выполнить любое число параллельных процессов в режиме разделения времени. Микропроцессор обеспечивает 2 уровня приоритета. Процессы с высоким приоритетом используются для передачи сообщений и быстрой реакции на внешние события. Наличие таймера позволяет следить за временем, либо ждать определенного момента времени (события).

Внутреннее статическое ОЗУ имеет емкость 4 Кбайта. Последовательные каналы реализуются с помощью блока интерфейса каналов связи. Стандартная скорость передачи по последовательному каналу составляет 10 Мбит/сек. Интерфейсы каналов и микропроцессора работают параллельно. Интерфейс внешнего запоминающего устройства обеспечивает доступ к памяти с помощью 32-битовой шины *А/Д*. Для работы с различными типами запоминающих устройств он реализует различные режимы работы. В блок системных устройств входят тактовые генераторы, устройства инициализации и сброса, обеспечивающие функционирование транспьютера.

Объем адресуемой памяти составляет 4 Гбайта. Адресация внутренней и внешней памяти одинакова, а программы и данные могут располагаться в любой области памяти.

Обобщенная структурная схема транспьютера *T800* с встроенным устройством выполнения операций с плавающей точкой (*FPU*) приведена на рис.2.33. Она содержит центральный процессор, интерфейс внешней памяти, 4 двунаправленных канала для связи между транспьютерами в многопроцессорной системе (интерфейсы каналов связи), блок режима работы

интерфейсов канала связи, программируемый блок событий, таймер, внутреннее ОЗУ и блок системных функций.

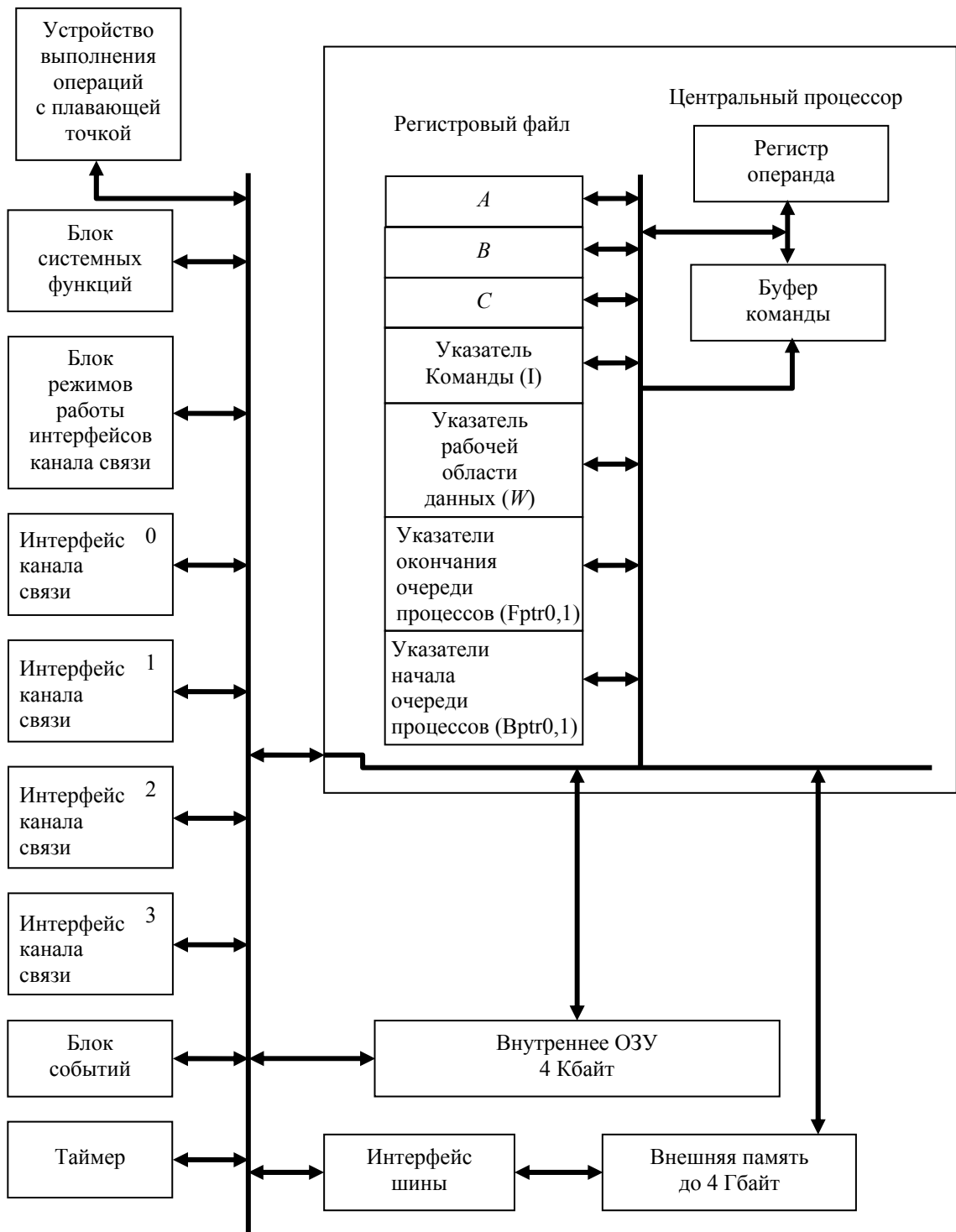


Рис. 2.33. Структурная схема транспьютера T800

Аппаратный центральный процессор организует в режиме разделения времени выполнение нескольких команд какой-либо программы или фрагмента программы. Каждому процессу в адресном пространстве транспьютера выделяется рабочая область памяти запоминающего устройства и ус-

танавливается приоритет. Адрес рабочей области и приоритет образуют дескриптор процесса. В центральном процессоре имеются 2 регистра - таймера для процессов с разными приоритетами.

Регистр *W* определяет рабочую область процесса, а в регистре *I* содержится адрес следующей выполняемой команды. Регистры общего назначения *A*, *B*, *C* образуют стек с дисциплиной *FIFO*.

Для организации очереди процессов на выполнение используются пары регистров *Bptr0*, *Fptr0* и *Bptr1*, *Fptr1*, указывающие начало и конец очередей с разными приоритетами.

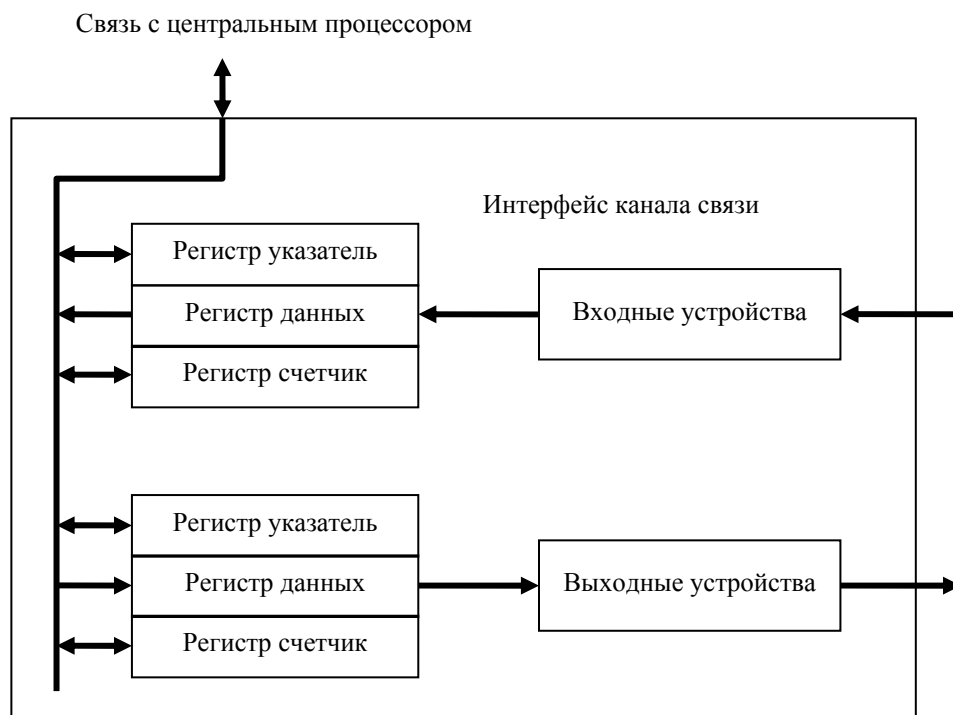


Рис. 2.34 Структурная схема интерфейса канала связи транспьютера T800

Взаимодействие между транспьютерами осуществляется посредством передачи сообщений по интерфейсам каналов связи. Байты передаются по одному проводу из пары проводов, а по другому передаются подтверждения приема каждого байта. Байт передается в обрамлении служебных битов. Сначала следует стартовый, затем управляющий, после него передаются 8 информационных и стоповый бит. Подтверждение состоит из стартового и управляющего битов. В транспьютерах, участвующих в обмене запрограммировано одно и то же число как передаваемых, так и принимаемых байт.

Если один из процессов, участвующих в обмене через интерфейс каналов связи, не готов к приему данных, байты накапливаются в регистре

данных интерфейса канала связи (рис.2.34). Скорость передачи по интерфейсам каналов связи задается блоком режимов работы интерфейса канала связи.

Дальнейшее совершенствование транспьютеров [12] привело к введению аппаратной поддержки механизма виртуальных интерфейсов каналов связи и аппаратного устройства группирования команд, позволяющего повысить загрузку параллельно работающих устройств многопроцессорной системы.

Механизм виртуальных интерфейсов каналов связи позволяет по одному физическому интерфейсу каналов связи вести обмен между произвольным числом пар процессов, протекающих в различных транспьютерах. Для управления процессами обмена используется встроенный процессор виртуального канала (*VCP*). Он делит сообщение на пакеты, содержащие заголовок пакета, 32 байта данных и окончание пакета. Используя информацию заголовка, встроенный процессор виртуального канала осуществляет маршрутизацию пакетов и сбор сообщений.

## Глава 3. Структура и характеристики специализированных микропроцессоров

### 3.1 Микроконтроллеры

Микроконтроллер содержит микропроцессор, оперативное и постоянное запоминающие устройства, параллельные и последовательные порты ввода-вывода, таймеры. Кроме того, в микроконтроллер могут входить блоки аналогового ввода-вывода (аналого-цифровые и цифроаналоговые преобразователи). Это делает микроконтроллеры функционально законченными устройствами с минимальными габаритами, высокой надежностью и эффективностью, которые могут быть использованы в устройствах и системах управления бытовой радиоэлектроники, аппаратуре связи и телекоммуникационных систем, системах охранной сигнализации, средствах контроля и диагностики, системах сбора данных и др.

#### 3.1.1 Восьмиразрядные микроконтроллеры

Такие микроконтроллеры используют архитектуру MCS – 51 [4, 11, 15, 19], либо *RISC* архитектуру (см. раздел 8.3, а также [8, 12, 15, 19, 26]). Рассмотрим подробнее микроконтроллер, использующий архитектуру MCS-51. Структурная схема представлена на рис.3.1. В микроконтроллер входят, описанные в разделе 7, узлы 8 - разрядного микропроцессора: арифметико-логическое устройство (АЛУ), набор регистров АЛУ (регистр-аккумулятор, буферные регистры БР1, БР2, регистр общего назначения *B*), регистр словосостояния программы (*PSW*), регистр команд и счетчик команд со схемой инкремента-декремента, указатель стека, регистр - указатель данных (*DPTR*). Устройство управления и синхронизации и генератор тактовых импульсов формируют последовательности сигналов, синхронизирующих процессы в узлах микроконтроллера и организацию обмена данными микроконтроллера с периферийными устройствами.

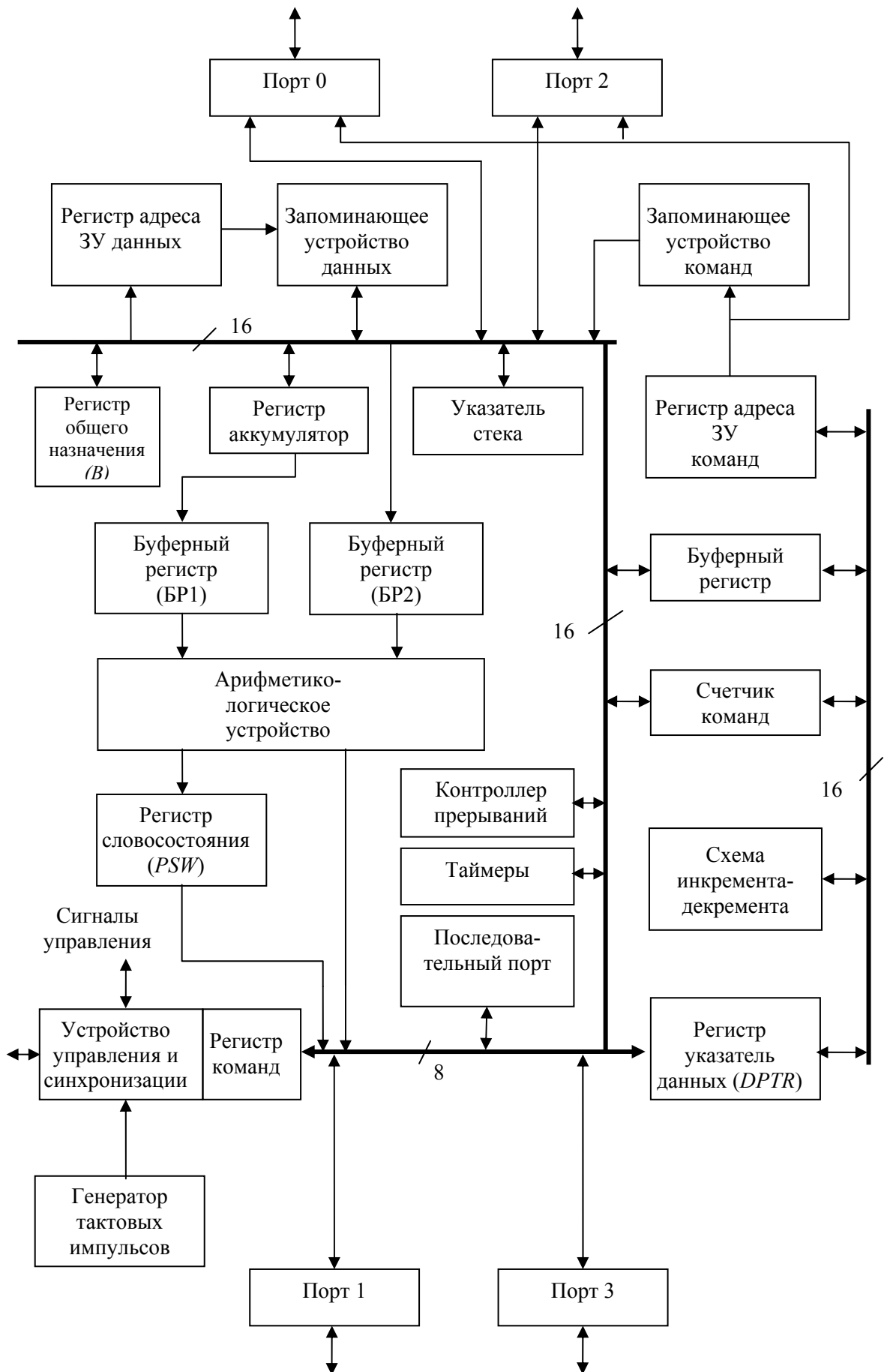


Рис. 3.1 Структурная схема микроконтроллера с архитектурой MCS-51

Кроме узлов микропроцессора в состав микроконтроллера входят:

- Запоминающее устройство команд с регистром адреса ЗУ команд. Запоминающее устройство может быть программируемым однократно, перепрограммируемым с ультрафиолетовым стиранием или флэш-память.
- Четыре двунаправленных восьмиразрядных порта (Порт 0-3) с возможностью побитной адресации устройств ввода-вывода. Они осуществляют обмен данными микроконтроллера с внешними устройствами. Контроллер прерываний (см. раздел 7.3.3 рис.7.13) реализует обработку прерываний от внешних и внутренних сигналов.
- Последовательный порт для передачи данных между микроконтроллером и внешними устройствами в последовательном коде.
- Таймеры, осуществляющие формирование временных интервалов, что позволяет реализовать управление в реальном масштабе времени.

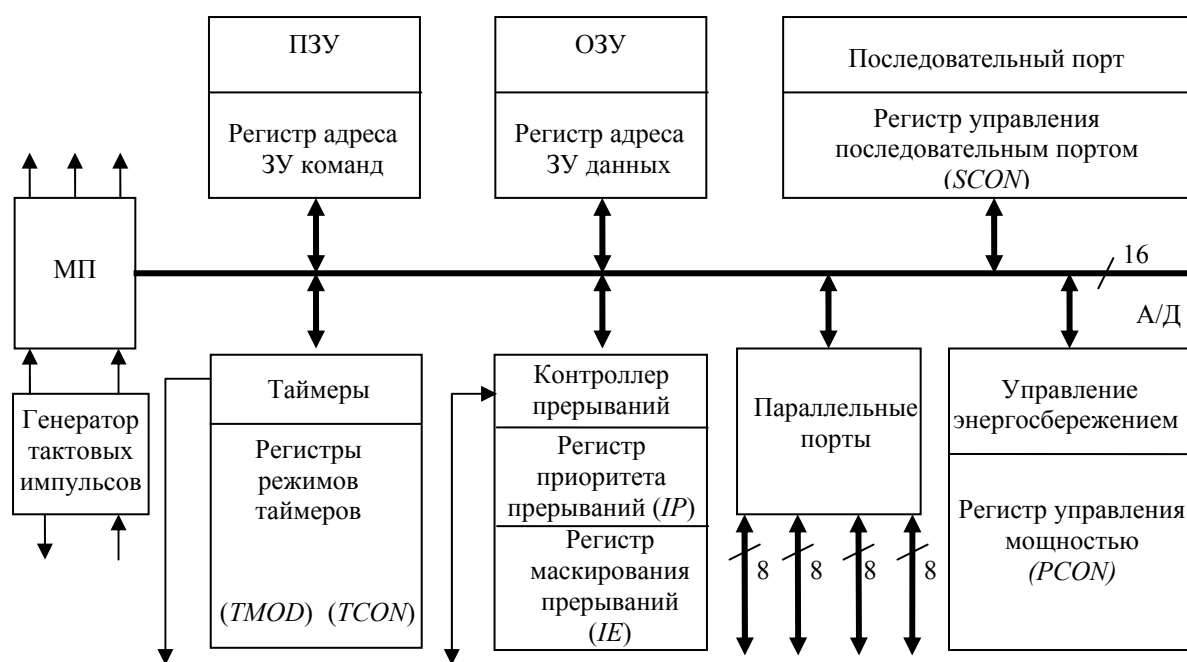


Рис. 3.2 Обобщенная структурная схема микроконтроллера KP1816BE51

Микропроцессор выбирает команды из запоминающего устройства команд и организует их выполнение в заданной последовательности. Машинные коды команды могут иметь форматы от 1 до 3 байт. Микроконтроллеры программируются с помощью программатора, либо с использованием персонального компьютера или другого микроконтроллера. Через



последовательный порт микроконтроллера передаются команды для записи их в ЗУ команд.

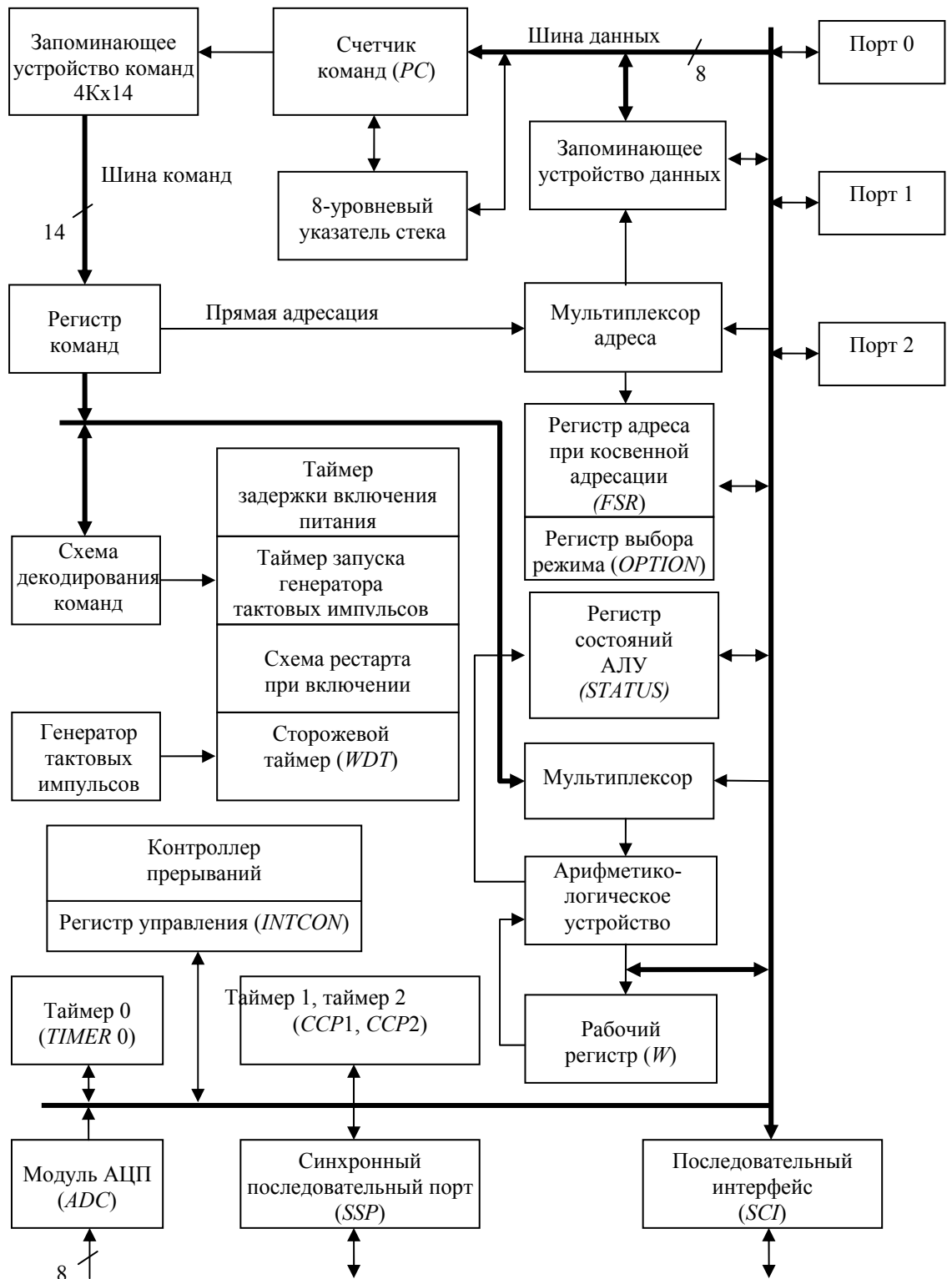


Рис. 3.3 Структурная схема микроконтроллера PIC16C74

В состав контроллера прерываний входят регистры *IE* и *IP* (рис.3.2). Информация, записанная в разряды регистра *IE*, определяет возможность

приема запросов прерывания контроллером прерывания. Биты регистра *IP* определяют приоритет запросов прерывания от внешних устройств, таймеров или последовательного порта. Различные серии микроконтроллеров, использующие архитектуру MCS-51, могут различаться уровнями напряжения питания, тактовыми частотами, числом таймеров и дисциплиной обслуживания запросов прерывания.

К микроконтроллерам, использующим архитектуру MCS-51, относится микроконтроллер КР1816ВЕ51 [11, 21], обобщенная структурная схема которого приведена на рис.3.2. Микропроцессор содержит дешифратор команд, арифметико-логическое устройство, битовый процессор (БП), блок программного управления, блок регистров специальных функций. В состав блока программного управления, который управляет последовательностью выполнения команд, хранящихся в ПЗУ, входит счетчик команд. Арифметико-логическое устройство выполняет операции сложения, вычитания, умножения, деления, сдвига и логические операции. Битовый процессор реализуют операции установки в единицу или ноль, переход по установленному биту, операции переноса.

Два таймера используются для измерения интервалов времени, формирования импульсов определенной длительности, счета и генерации периодических сигналов. Последовательный порт, таймеры и контролер прерываний имеют в своем составе управляющие регистры: регистр управления последовательным портом (*SCON*), регистр управления таймером (*TCON*), регистр режима таймера (*TMOD*), регистр управления энергосбережением (*PCON*), регистр приоритета прерываний (*IP*), регистр маскирования прерываний (*IE*).

Основные характеристики некоторых типов микроконтроллеров, использующих архитектуру MCS-51, приведены в табл.3.1 [19].

Таблица 3.1

Характеристики микроконтроллеров	Z86C11	I80C51	PIС17C42
Запоминающее устройство команд	4Кбайт x 8	4Кбайт x 8	2Кбайт x 16
Ввод-вывод: Порты	4x8 бит	4x8 бит	5 портов
Прерывания:			
Источники	8	5	11 внутренних и внешних
Приоритеты	48, про-	2, про-	Нет про-

Маскируемые	граммное задание 6	граммное задание 5	граммного задания 4
Внешние запоминающие устройства (ROM+RAM)	120Кбайт	120Кбайт	64Кбайт слов (только память команд)
Стек: Указатель стека Внутренний стек Внешний стек	16 бит Да Да	8 бит Да Нет	Аппаратный Да Нет
Таймеры/счетчики:	Два по 8 бит	Два по 16 бит или два по 8бит	Три по 16 бит
Последовательный интерфейс: Полный дуплекс Прерывания - при передаче - при приеме	Да Да Да	Да Одно в обоих случаях	Да Да Да
Быстродействие Время выполнения команды	1,0 мкс при 16 МГц	1,0 мкс при 16 МГц	0,25 мкс
Тактовая частота	12 и 16 МГц	12 и 16 МГц	16 МГц
Режим пониженного энергопотребления	Сохраняются состояния 256 регистров	Сохраняются состояния 128 регистров	Сохраняются состояния портов ввода/вывода

Примером микроконтроллера с *RISC* архитектурой является микросхема PIC16C74 (*Peripheral Interface Controller*) (рис.3.3). Микроконтроллеры серии PIC [3,16,19] имеют: программируемое ППЗУ, минимальное энергопотребление, высокую производительность, развитую *RISC* архитектуру, функциональную законченность и минимальные размеры. В них реализованы следующие принципы организации.

- Архитектура с физическим разделением памяти команд и данных с использованием двух отдельных шин доступа к памяти, по которым одновременно осуществляется информационный обмен как с памятью команд, так и с памятью данных.
- *RISC* архитектура с системой однословных команд (всего 40 команд).
- Использование концепции регистрового блока с возможностью адресации к любому регистру, включая счетчик команд. При этом все ре-

гистры специального назначения (регистры состояния таймеров, регистры управления портами ввода/вывода) являются частью регистрового блока и все ячейки запоминающего устройства данных являются адресуемыми регистрами.

- Система периферийных устройств, обеспечивающих возможность создания встроенных систем управления.
- Наличие системы энергосбережения (монитор питания), в которую входят задержка включения питания, таймер запуска генератора тактовых импульсов, схема рестарта при включении и сторожевой таймер с собственным *RC* - генератором для предотвращения сбоя программы.
- Порты параллельного ввода-вывода (Порт 0-2).
- Модуль АЦП (*ADC*), имеющий 8 аналоговых входов, сигналы с которых с помощью мультиплексора подаются на восьмиразрядный аналого-цифровой преобразователь.
- Таймеры для обработки входных сигналов (захват, сравнение) и генерации широтно-модулированных сигналов.
- Контроллер прерываний как от внутренних, так и внешних устройств.

Организация памяти микроконтроллера PIC16C74 имеет 13-битный счетчик команд и возможность адресации 14-битных слов памяти команд. Память команд реализована в виде двух страниц, переключение которых осуществляется посредством задания соответствующих битов в регистре адреса при косвенной адресации (рис.3.3).

Запоминающее устройство состоит из двух банков емкостью 128 байт каждый, которые содержат регистры общего назначения и регистры специального назначения. Переключение между банками данных осуществляется битом *RP0* и *RP1* в регистре *STATUS* при использовании прямой адресации. Основные регистры микроконтроллера доступны в обоих банках памяти данных. Их использование не требует переключения банка данных. Основными регистрами микроконтроллера являются:

*STATUS* - определяет состояние микропроцессора и управляет его работой с памятью;

*W* - рабочий регистр для выполнения операций;

*PC* - счетчик команд, содержит номер выполняемой команды;

*INTCON* - регистр управления системой прерываний;

*FSR* - регистр адреса при косвенной адресации памяти данных.

По своему основному назначению микропроцессор PIC16C74, предназначен для использования в составе систем управления, в которых необходимо обеспечить информационный обмен между составляющими такую систему частями. Эту задачу решают синхронный последовательный порт (*SSP*) и последовательный интерфейс (*SCI*).

Синхронный последовательный порт (*SSP*), позволяет осуществить один из двух режимов работы:

- *SPI* - синхронный последовательный интерфейс;
- *I<sup>2</sup>C* - двухпроводной интерфейс.

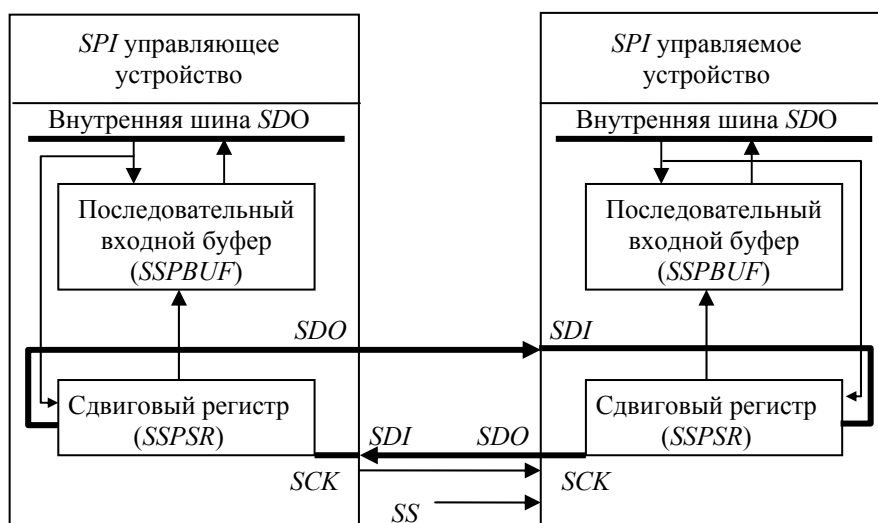


Рис. 3.4 Схема соединения устройств с использованием *SPI* интерфейса

Синхронный последовательный порт (*SSP*), как правило, используется для связи с компьютером в режиме *SPI*. В данном режиме можно передавать и принимать данные побайтно с генерацией прерывания при завершении обмена. Синхронный последовательный порт (рис.3.4) содержит приемопередающий сдвиговый регистр (*SSPSR*) и буфер обмена (*SSPBUF*). При обмене используются шины для передачи (*SDO*) и приема (*SDI*) последовательно организованных данных.

Выход *SDI* используются для синхронизации работы устройств. Для управления переводом в режим *SPI* управляемого устройства используется вход *SS*.

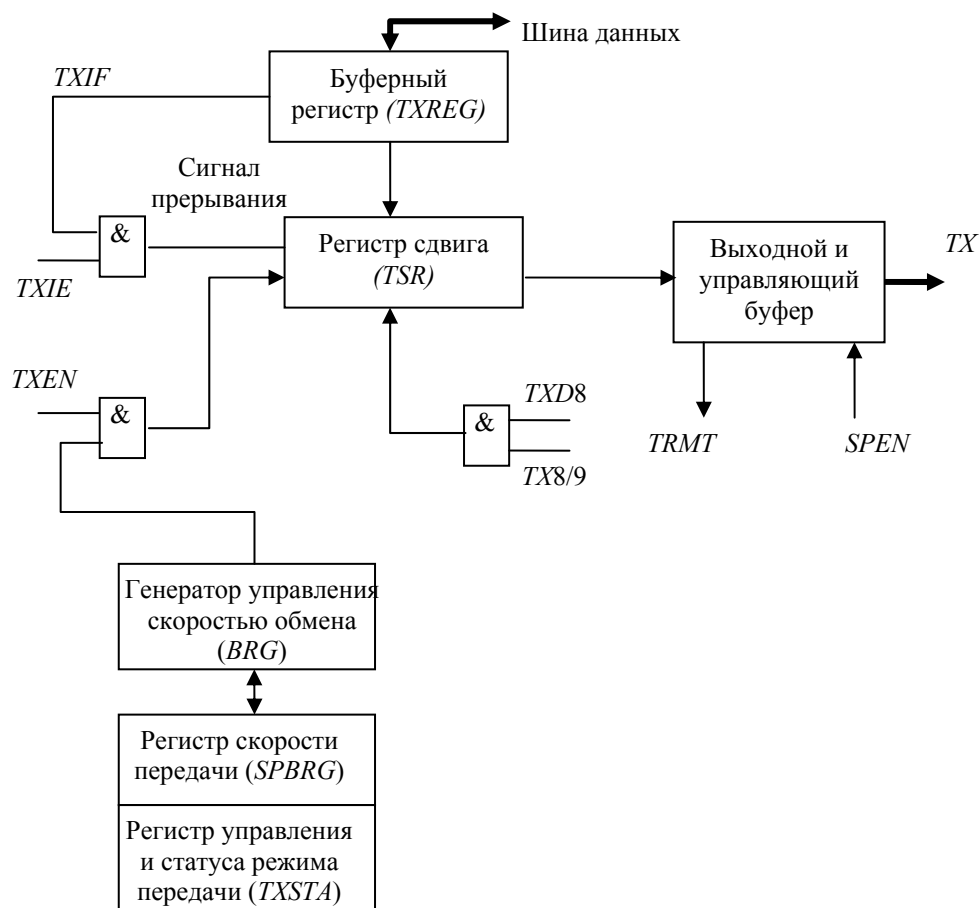


Рис. 3.5 Структурная схема асинхронного передающего устройства

Управление работой последовательного синхронного порта осуществляется посредством регистра *SSPCON*, а управление состоянием процесса передачи регистром *SSPSTAT* (на рис.3.4 – не показаны). Сдвиговые регистры передают данные из микропроцессора побитно и, соответственно, одновременно принимают данные из другого микропроцессора.

Механизм двойной буферизации обеспечивает возможность приема следующего байта до окончания предыдущего. Прием байта осуществляется в сдвиговом регистре, а промежуточное хранение – в регистре *SSPBUF*. Запись данных в регистр во время приема или передачи данных не допускается.

При ожидании приема очередного байта проводится считывание данных из регистра *SSPBUF* до момента приема очередного байта.

Бит *BF* в регистре *SSPSTAT* показывает, что в регистре *SSPBUF* содержится вновь полученный байт. После этого возможно чтение или запись в регистр *SSPBUF*.

Последовательный интерфейс *SCI* (рис.3.3) может работать в следующих режимах работы:

- асинхронный (полный дуплексный обмен);
- синхронный режим управляющего устройства;
- синхронный режим управляемого устройства.

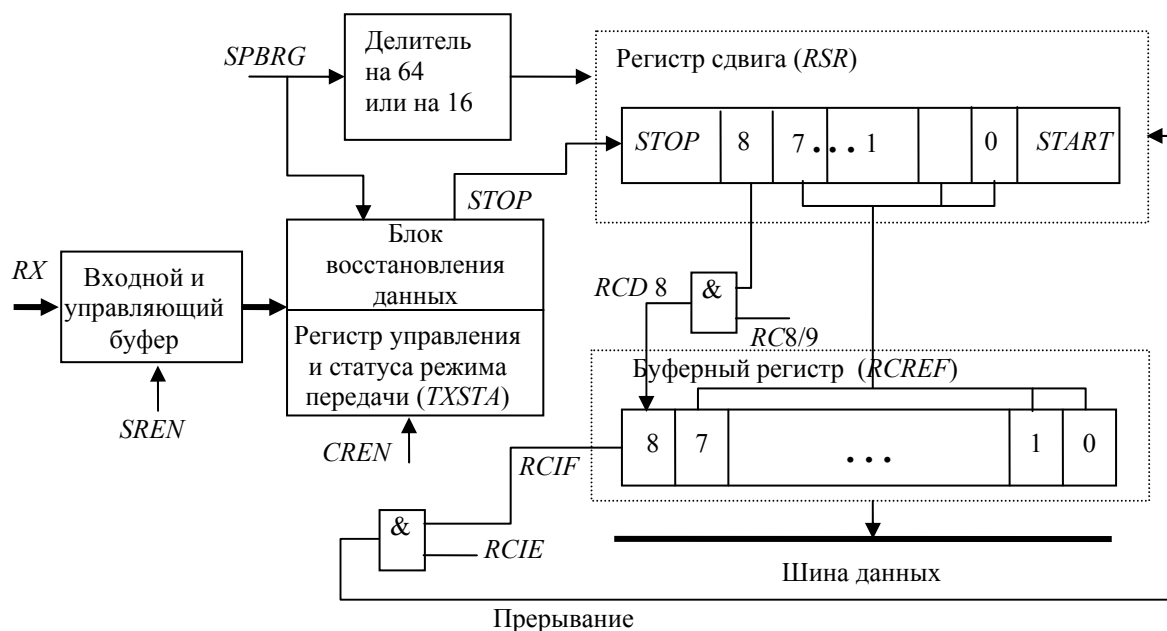


Рис. 3.6 Структурная схема асинхронного приемного устройства

Для организации информационного обмена, как в асинхронном, так и в синхронном режимах работы используются асинхронные передающее и приемное устройства (рис.3.5 и рис.3.6 соответственно). В асинхронном передающем устройстве (рис.3.5) применяется генератор управления скоростью обмена (*BRG*), который позволяет программным способом управлять скоростью обмена за счет использования регистра скорости передачи (*SPBRG*) и бита *BRGH* в регистре управления и статуса режима передачи (*TXSTA*).

Регистр сдвига (*TSR*) получает данные из буферного регистра (*TXREG*), куда они заносятся программным способом. Регистр сдвига не принимает новые данные до тех пор, пока не будет передан бит остановки для предыдущего обмена. При завершении передачи буферный регистр обнуляется и формируется сигнал прерывания *TXIF*. Состояние регистра сдвига отражается сигналом *TRMT* (бит индикации состояния регистра сдвига).

Информационный обмен начинается при установке единичного значения сигнала *TXEN* и при наличии сигнала разрешения *SPEN* на входе выходного и управляющего буфера (рис.3.5).

При необходимости использования режима прерываний сигнал *TXIE* устанавливается равным единице. При использовании 9-битовой посылки формируется уровень логической единицы *TX8/9* и с помощью сигнала *TXD8* в регистр сдвига загружается девятый бит.

В асинхронном приемном устройстве (рис.3.6) данные приходят на вход *RX* и через входной и управляющий буфер попадают в блок восстановления данных. Блок представляет собой высокоскоростной регистр сдвига, работающий со скоростью, превышающей в 16 раз скорость, задаваемую генератором управления скоростью обмена (рис.3.5). Основу асинхронного приемного устройства составляет регистр сдвига (*RSR*). После формирования сигнала *STOP*, принятые в регистр сдвига данные передаются в буферный регистр (*RCREG*). После завершения передачи устанавливается единичное значение сигнала *RCIE*. Сигнал с восьмого разряда буферного регистра считывается на вход логического элемента. В регистре управления и статуса режима приема (*RCSTA*) для установления непрерывного режима приема используется сигнал *CREN*. Сигнал *SREN* дает разрешение на побайтовый прием информации.

При завершении приема информации сигнал *RCIF* становится равным единице и запускается система прерываний. При этом значение сигнала *RCIE* также равно единице.

Для использования 9-битной посылки на вход логического элемента подается сигнал *RC8/9*, равный логической единицы.

Рассмотрим структуры основных регистров микроконтроллера PIC16C73. Структура регистра состояний арифметико-логического устройства (*STATUS*) приведена на рис.3.7. На этом рисунке символ *R/W* означает чтение/запись и указывает на возможность установки состояния бита (единица или ноль) независимо от режима работы микроконтроллера. Запись *R* означает процедуру чтения при условии записи бита только микроконтроллером.

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R</i>	<i>R</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
<i>IRP</i>	<i>RPI</i>	<i>RP0</i>	<i>TO</i>	<i>PD</i>	<i>Z</i>	<i>DC</i>	<i>C</i>

Рис. 3.7 Структура регистра состояний АЛУ (*STATUS*)

Назначение бит регистра состояний АЛУ следующее:

- *C* - является признаком переноса для команд сложения и вычитания;



- *DC* - признак переноса при сложении первых четырех разрядов при использовании двоично-десятичной системы счисления;
- *Z* - устанавливается в единицу, если результат арифметической или логической операции равен нулю;
- *PD* - устанавливается в единицу при включении питания и сбрасывается в ноль при его выключении;
- *TO* - устанавливается в единицу при включении питания и устанавливается в ноль при срабатывании сторожевого таймера (*WDT*);
- *RP0, RP1* – определяет выбор банка памяти области оперативного запоминающего устройства данных;
- *IRP*-не используется.

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
<i>RBPU</i>	<i>INTEDG</i>	<i>TOCS</i>	<i>TOSE</i>	<i>PSA</i>	<i>PS2</i>	<i>PS1</i>	<i>PS0</i>

Рис. 3.8/ Структура регистра выбора режима (*OPTION*)

На рис.3.8 приведена структура регистра выбора режима, назначение бит которого следующее:

- *PS0, PS1, PS2* - задание коэффициента деления тактовой частоты;
- *PSA* - структура делителя тактовой частоты;
- *TOSE* - установка фронта импульсов синхронизации таймера *TIMER0*;
- *TOCS* – бит, определяющий источник импульсов синхронизации для таймера *TIMER0*, равный единице, если регистр таймера использует внешнюю синхронизацию, и равный нулю при синхронизации регистра от внутреннего генератора;
- *INTEDG* - установка фронта импульсов внешнего прерывания;
- *RBPU* – бит установки внешних нагрузочных резисторов.

Регистр управления (*INTCON*) используется для чтения и записи информации. Он содержит биты признаков прерываний от таймера *TIMER0*, битов изменений сигнала на выводах порта и наличие внешнего прерывания.

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
<i>GIE</i>	<i>PEIE</i>	<i>TOIE</i>	<i>INTE</i>	<i>RBIE</i>	<i>TOIF</i>	<i>INTF</i>	<i>RBIF</i>

Рис. 3.9 Структура регистра управления (*INTCON*)

На рис.3.9 приведена структура регистра управления, где:

- *RBIF* - признак наличия изменений на выводах порта;
- *INTF* - наличие внешнего прерывания;
- *TOIF* - переполнение таймера (*TIMER0*);
- *RBIE* - разрешение прерывания по наличию изменения на выводах порта;
- *INTE* - разрешение внешнего прерывания;
- *TOIE* - разрешение прерывания по переполнению таймера (*TIMER0*);
- *PEIE* - разрешение прерывания с периферийных устройств; *GIE* - блокировка прерываний.

Источниками прерываний микросхемы PIC16C74 являются внешние прерывания, прерывания по переполнению таймера (*TIMER0*), прерывания по наличию изменений на выводах порта 1 (рис.3.3), сигнал прерывания с АЦП, прерывания по переполнению таймера 1 (рис.3.3), прерывания из модуля последовательного интерфейса (*SCI*) при приеме и передаче, прерывания от синхронного последовательного порта.

В таблице 3.2 [19] приведены основные характеристики микроконтроллеров фирмы Microchip, использующие *RISC* архитектуру.

Рассмотрим структуру микроконтроллеров MC68HC705C8 [26], представленную на рис.3.10. В микроконтроллер входят восьмиразрядный микропроцессор, внутренняя память общим объемом 8156 байт (ППЗУ с однократным программированием или с ультрафиолетовым стиранием и ОЗУ емкостью от 176 до 304 байт), 4 параллельных 8-разрядных порта, синхронный и асинхронный последовательные порты, 16-разрядный таймер и ряд обслуживающих блоков.

В микропроцессоре имеется 5 регистров: регистр-аккумулятор (*A*), индексный регистр (*X*), регистр условий (*CCR*), 13-разрядный указатель стека (*SP*), 13-разрядный счетчик команд (*PC*).

Таблица 3.2

Название	Объем запоминающих устройств команд	Тип запоминающих устройств команд	ОЗУ	Тактовая частота	Число линий ввода/вывода	Число таймеров	Последовательный интерфейс
PIC16C662	4096 бит x14	Однократно программируемые	176 бит	20 МГц	33	1, сторожевой таймер	-

PIC16C72	2048 бит x14	Однократно программируемые	128 бит	20 МГц	22	3, сторожевой таймер	I <sup>2</sup> C/SPI
PIC16C73 A	4096 бит x14	Однократно программируемые	192 бит	20 МГц	22	3, сторожевой таймер	асинхр.послед. интерфейс; I <sup>2</sup> C/SPI
PIC16C74 A	4096 бит x14	Однократно программируемые	192 бит	20 МГц	33	3, сторожевой таймер	асинхр.послед. интерфейс; I <sup>2</sup> C/SPI
PIC16C76	8192 бит x14	Однократно программируемые	368 бит	20 МГц	22	3, сторожевой таймер	асинхр.послед. интерфейс; I <sup>2</sup> C/SPI
PIC16F83	512 бит x14	Перепрограммируемые электрическим способом	36 бит 64 бит пере- про- грам- ми- руемое	10 МГц	13	1 сторожевой таймер	-
PIC17CR4 3	4096 бит x16	Масочное ПЗУ	454	33 МГц	33	4 сторожевой таймер	асинхр.послед. интерфейс;
PIC17C76 6	16384 бит x16	Однократно программируемые	902	33 МГц	66	4 сторожевой таймер	2 асинхр.послед. интерфейс; I <sup>2</sup> C/SPI
PIC16C17 8	16384 бит x16	Однократно программируемые	1536	40 МГц	22	3, сторожевой таймер	2 асинхр.послед. интерфейс; I <sup>2</sup> C/SPI

Микроконтроллер имеет три 8-разрядных параллельных порта ввода-вывода (Порт 0-2), каждый из которых содержит регистр данных  $PxDR$  и регистр направления пересылок  $PxDDR$  (символ  $x = 0, 1, 2$  указывает номер порта). Порт  $D$  является портом ввода. Информация с 7-ми линий этого порта поступает в микроконтроллер с помощью регистра порта  $PDIR$ . При включении асинхронного и синхронного последовательных портов (рис.3.10) соответствующие выходы порта используются для их обслуживания.

Асинхронный последовательный порт ( $SCI$ ) содержит буферный регистр данных ( $SCDR$ ), регистр состояния ( $SCSR$ ), два регистра управления

(*SCCR1*, *SCCR2*) и генератор синхронизирующих сигналов (*ГСС*), рабочая частота которого определяется содержимым регистра (*BRR*).

Синхронный последовательный порт (*SPI*) также имеет регистр данных (*SPDR*), регистр управления (*SPCR*) и регистр состояния (*SPSR*).

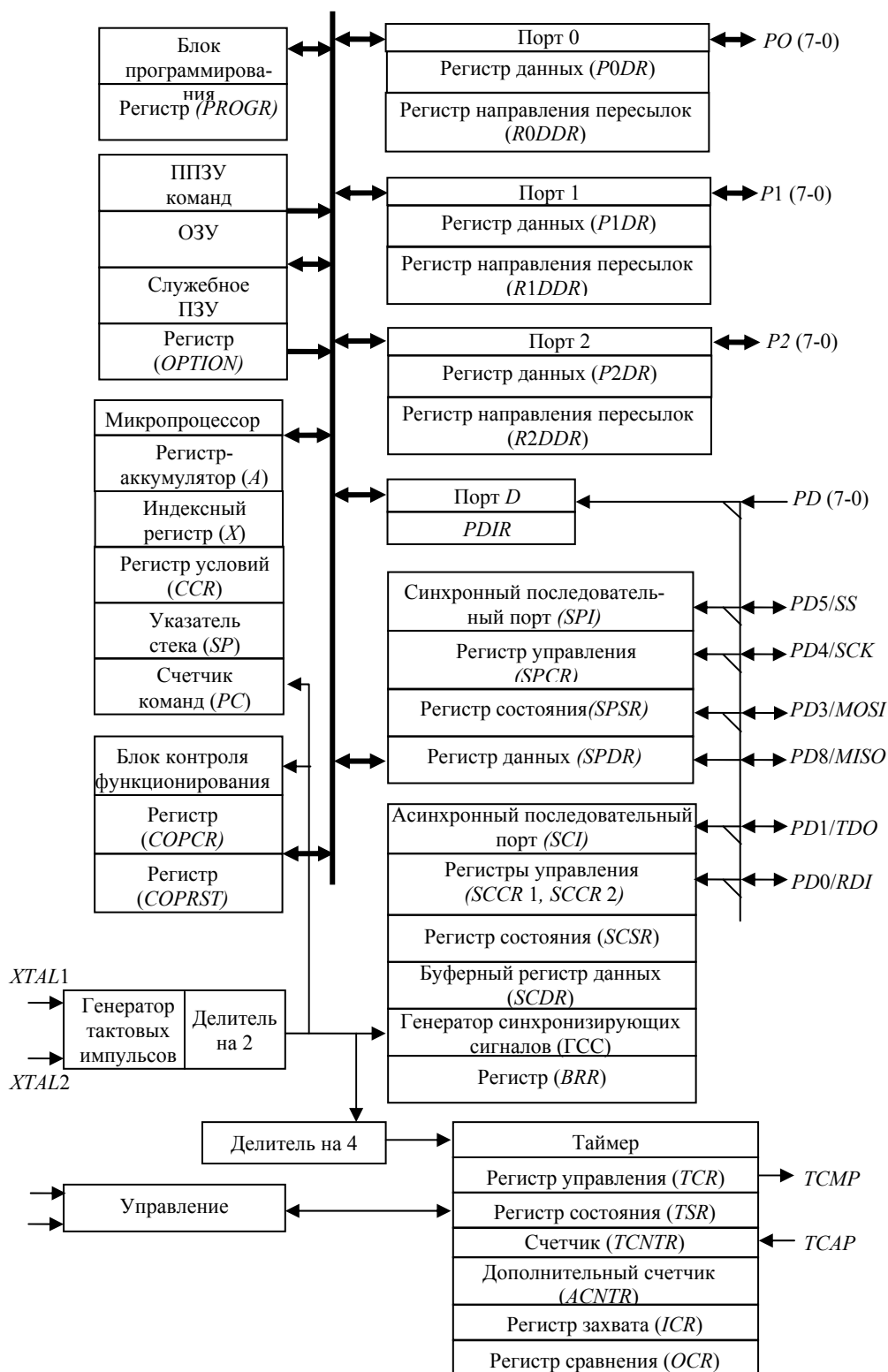


Рис. 3.10 Структура микроконтроллера MC68HC705C8

Режимы функционирования последовательных портов определяются управляющими кодами, записанными в их регистры управления.

Для организации приема и формирования управляющих сигналов в заданные моменты времени в микроконтроллере используется 16-разрядный таймер. В его состав входят 8-разрядные регистры управления (*TCR*) и состояния (*TSR*), а также 16-разрядные регистры: счетчик (*TCNTR*), дополнительный счетчик (*ACNTR*), регистр захвата (*ICR*) и регистр сравнения (*OCR*).

Блоками обслуживания в микроконтроллере являются блок программирования с регистром (*PROGR*) и блок контроля функционирования с регистрами (*COPCR*, *COPRST*), осуществляющий контроль выполнения программы и частоты тактовых импульсов.

Устройство управления энергосбережением реализует:

- режим ожидания, при котором прекращает работу микропроцессор, но продолжают работу таймеры и последовательные порты;
- режим остановки, при котором прекращается работа микропроцессора, таймера и последовательных портов.

Основные характеристики микроконтроллеров этой группы с однократно программируемым/стираемым ПЗУ, приведены в таблице 3.3 [19].

Таблица 3.3

Название	Запоминающее устройство команд	ОЗУ	ЗУ команд с программированием электрическим способом	Таймер	Число линий ввода/вывода	Последовательный интерфейс	АЦП
68HC705B5	6Кбайт	176 бит	-	16 бит	24	SCI/SPI	8 бит
68HC705B3 2	32Кбайт	528 бит	256 бит	16 бит	32	SCI/SPI	8 бит
68HC705G4	32Кбайт	1024 бит	-	16 бит	48	2SPI	8 бит
68HC705JP7	64Кбайт 64 бит программирование электрическим способом	224 бит	-	16 бит многофункциональный, прерывание в реальном времени	22	SPI	12 бит
68HC805K3	-	64 бит	920 бит 16Кбайт программирование электрическим	многофункциональный, прерывание в реальном времени	10	-	-

			ским способом				
68HC705T1 6	24Кбайт	320 бит	-	16 бит	40	I <sup>2</sup> C	5 бит
68HC05RC1 8	8Кбайт	350 бит	-	инфракрасный таймер	20	-	-
68HC05SC2 4	3Кбайт	128 бит	1Кбит	-	5	-	-

В заключение отметим, что расширение функциональных возможностей микроконтроллеров возможно путем объединения ПЛИС, статических ОЗУ, микропроцессора с *RISC* архитектурой и устройств связи с периферийными устройствами. Примером является микроконтроллер, содержащий ПЛИС типа AT40K FPGA, 36-Кбайтное ОЗУ, 8-битный микропроцессор с *RISC* архитектурой с аппаратным перемножителем, устройства связи (таймер, приемопередающее устройство, порты ввода-вывода) и контроллер конфигурации. Наличие ПЛИС позволяет реализовать операции цифровой обработки сигналов аппаратным способом.

### 3.1.2. Шестнадцати и тридцатидвухразрядные микроконтроллеры

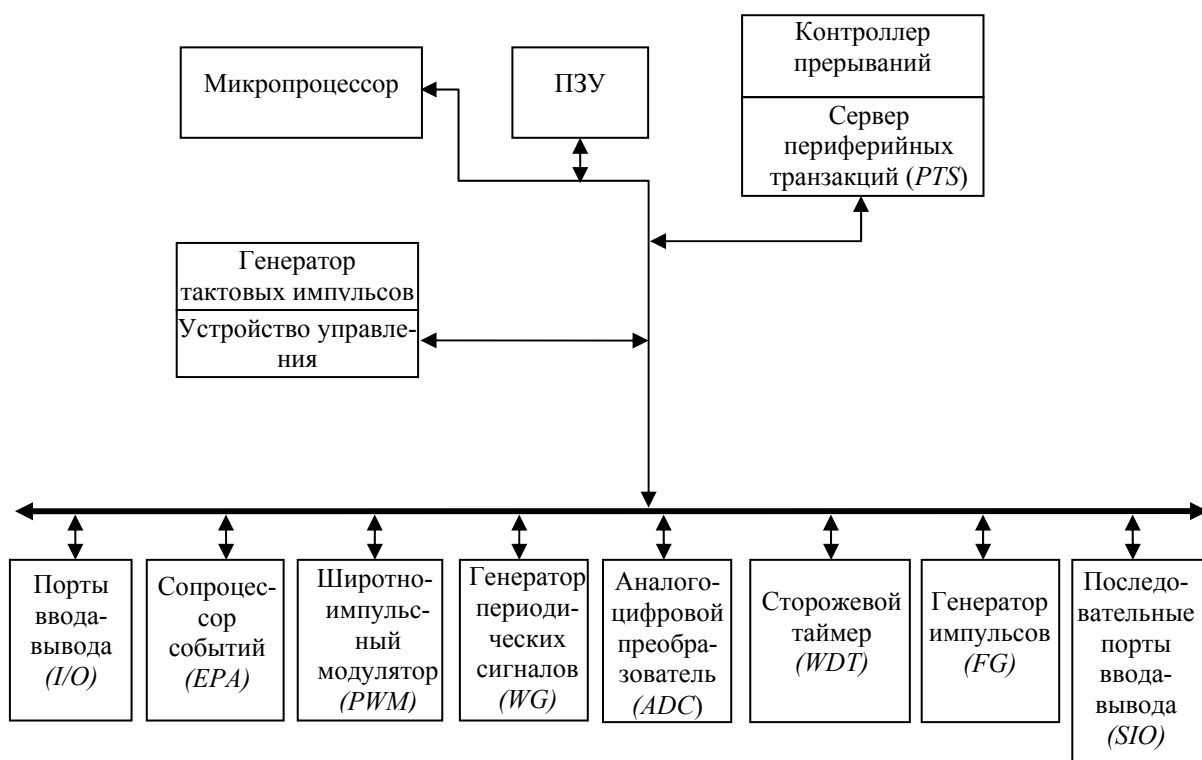


Рис. 3.11 Структурная схема микроконтроллера с архитектурой MCS196

Усложнение функций, выполняемых микроконтроллером, рост требований, предъявляемых к его производительности, а также увеличение разрядности АЦП и объемов обрабатываемых данных, привели к переходу на

микроконтроллеры с более высокой разрядностью – 16 и 32. Помимо увеличенной разрядности 16 и 32-разрядные микроконтроллеры обладают и более высокой тактовой частотой. Так, 16-разрядные микроконтроллеры имеют тактовую частоту 20,97 МГц; 32-разрядные до 40 МГц и выше.

Дополнительное повышение производительности осуществляется за счет введения в структуру микроконтроллера сопроцессоров различной функциональной ориентации: обмена данными; вычисления математических операций, ввода-вывода, цифровой обработки сигналов и др. Они позволяют разгрузить основной микропроцессор, берут на себя выполнение специфических функций и определяют ориентацию микроконтроллера на конкретную область использования. Рассмотрим особенности 16-разрядного однокристального микроконтроллера с архитектурой MCS-196, структурная схема которого приведена на рис.3.11.

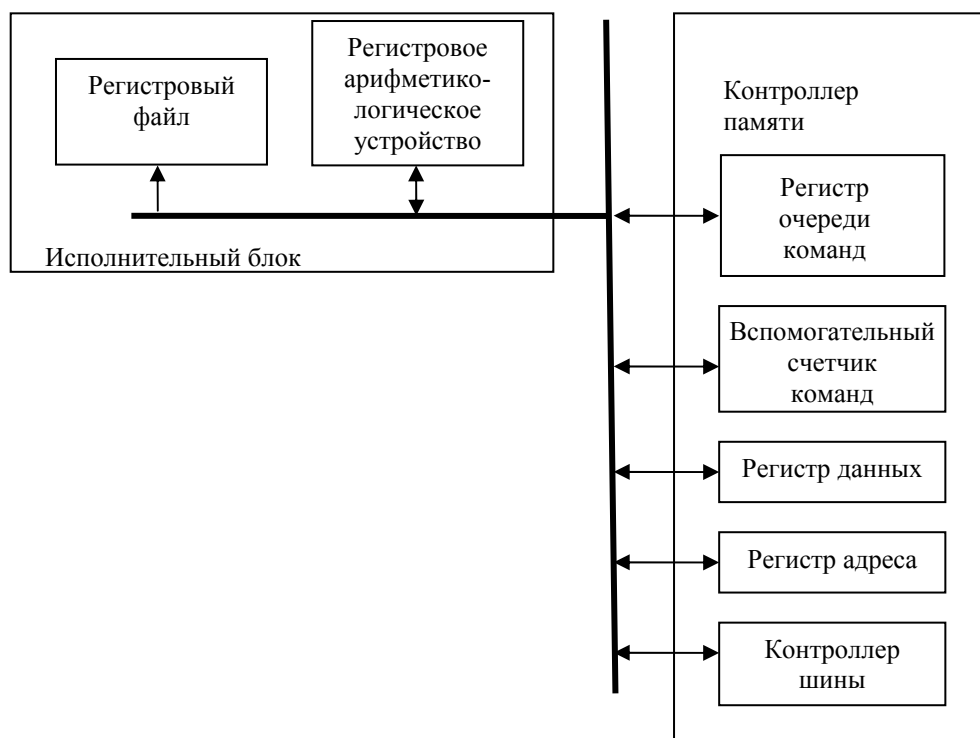


Рис. 3.12 Структурная схема микропроцессора, входящего в состав микроконтроллера с архитектурой MCS-196

В состав микроконтроллера входят микропроцессор, запоминающее устройство команд (ПЗУ), генератор тактовых импульсов и устройство управления энергосбережением, контроллер прерываний, сервер периферийных транзакций *PTS* (*Peripheral transaction server*). Последний представляет собой специализированный контроллер прерываний (сопроцессор ввода – вывода), обеспечивающий быстрый обмен данными между внеш-

ними устройствами и запоминающим устройством без участия микропроцессора. Набор встроенных периферийных устройств содержит:

- Порты ввода-вывода, в том числе высокоскоростные (*HSIO*).
- Сопроцессор событий (*EPA*), имеющий 10 каналов захвата/сравнения и 2 канала сравнения.
- Трехканальный широтно-импульсный модулятор (*PWM*).
- Аналого-цифровой преобразователь, который осуществляет преобразования сигналов, поступающих на 8 аналоговых входов (*ADC*).
- Многоканальный генератор периодических сигналов (*WG*).
- Сторожевой таймер (*WDT*).
- Генератор импульсов (*FG*).
- Последовательные порты ввода-вывода (*SIO*) с управляемыми программным способом генераторами скорости передачи или приема.

Генератор периодических сигналов (*WG*) предназначен для формирования трех синхронизированных сигналов с широтно-импульсной модуляцией. Эти сигналы используются для управления работой внешних устройств (инверторов, двигателей и др.). Генератор содержит цифровой двухканальный блок формирования импульсных последовательностей с заданной величиной периода следования импульсов, трехканальный блок цифрового компаратора, формирующего три пары взаимноинверсных независимых последовательностей сигналов с широтно-импульсной модуляцией. Кроме того, генератор периодических сигналов содержит блок управления выходами, который задает логические уровни выходных сигналов и временные интервалы их формирования. Сигналы с широтно-импульсной модуляцией имеют период следования импульсов от 0.25 мксек до 16 мксек.

Генератор импульсов (*FG*) предназначен для подключения удаленных периферийных устройств. Он формирует периодическую последовательность импульсов в диапазоне частот от 4 КГц до 1 МГц. Генератор содержит вычитающий счетчик, *D*-триггер и компаратор.

Трехканальный широтно-импульсный модулятор (*PWM*) предназначен для формирования импульсных сигналов с фиксированным периодом следования и регулируемой скважностью. Каждый канал модулятора имеет регистр скважности, компаратор и триггер, предназначенные для формирования сигналов с широтно-импульсной модуляцией. Трехканальный широтно-импульсный модулятор содержит программируемый делитель



тактовой частоты и регистр формирования периода следования импульсных сигналов.

Многоканальный аналого-цифровой преобразователь состоит из 8-канального аналогового мультиплексора, устройства выборки и хранения и 10-разрядного АЦП последовательных приближений с регулируемым временем преобразования в диапазоне от 10 мсек до 20 мсек.

Процессор событий (*EPA*) состоит из модуля таймеров/счетчиков, каналов захвата/сравнения и каналов сравнения. Программирование работы каналов осуществляется с помощью регистра управления.

В состав последовательных портов входят универсальные синхронные и асинхронные приемники и передатчики, программируемые генераторы скорости передачи и приема данных, буферные регистры приемников и передатчиков, блок управления последовательным портом.

Сторожевой таймер (*WDT*) осуществляет контроль выполнения программы работы контроллера. При этом выполняется принудительный сброс таймера в каждом цикле работы контроллера. Если сброс по каким-либо причинам не происходит (сбой в работе), то сторожевой таймер принудительно формирует сигнал сброса контроллера.

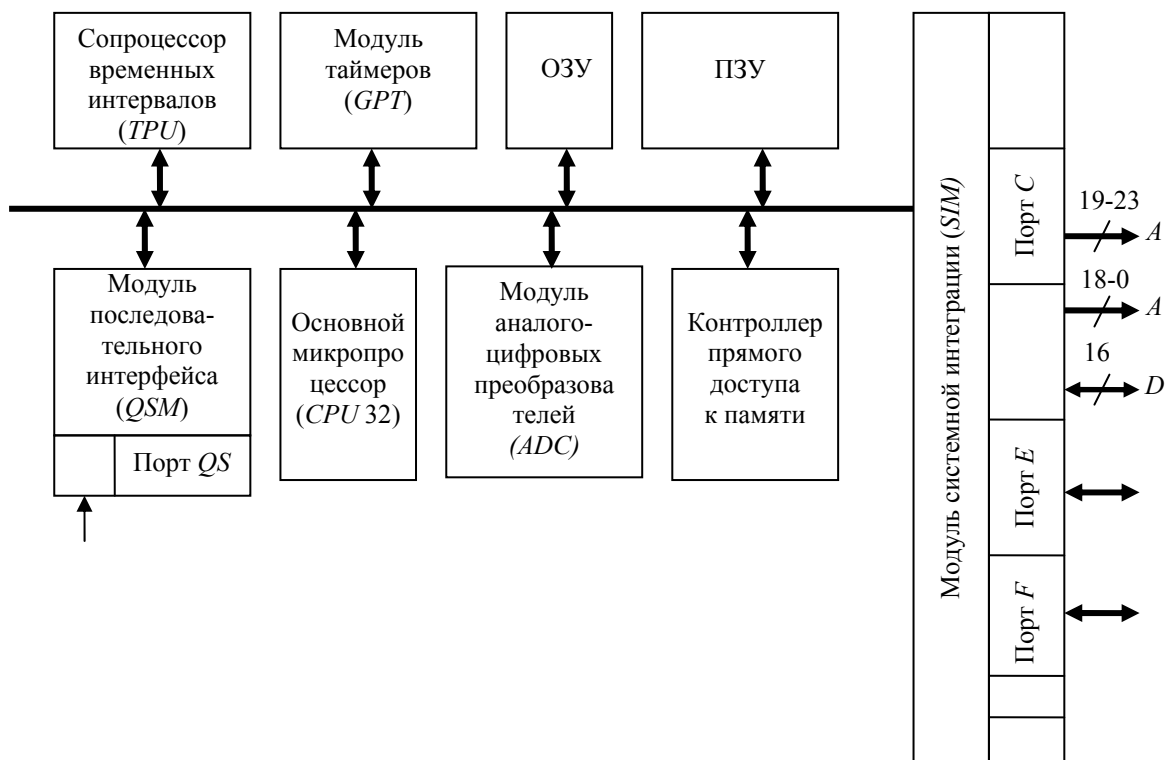


Рис. 3.13 Обобщенная структурная схема 32-разрядного микроконтроллера серии MC683xx

Микропроцессор (рис.3.12) содержит исполнительный блок, представляющий собой совокупность регистрового файла, регистрового арифметико-логического устройства и контроллера памяти.

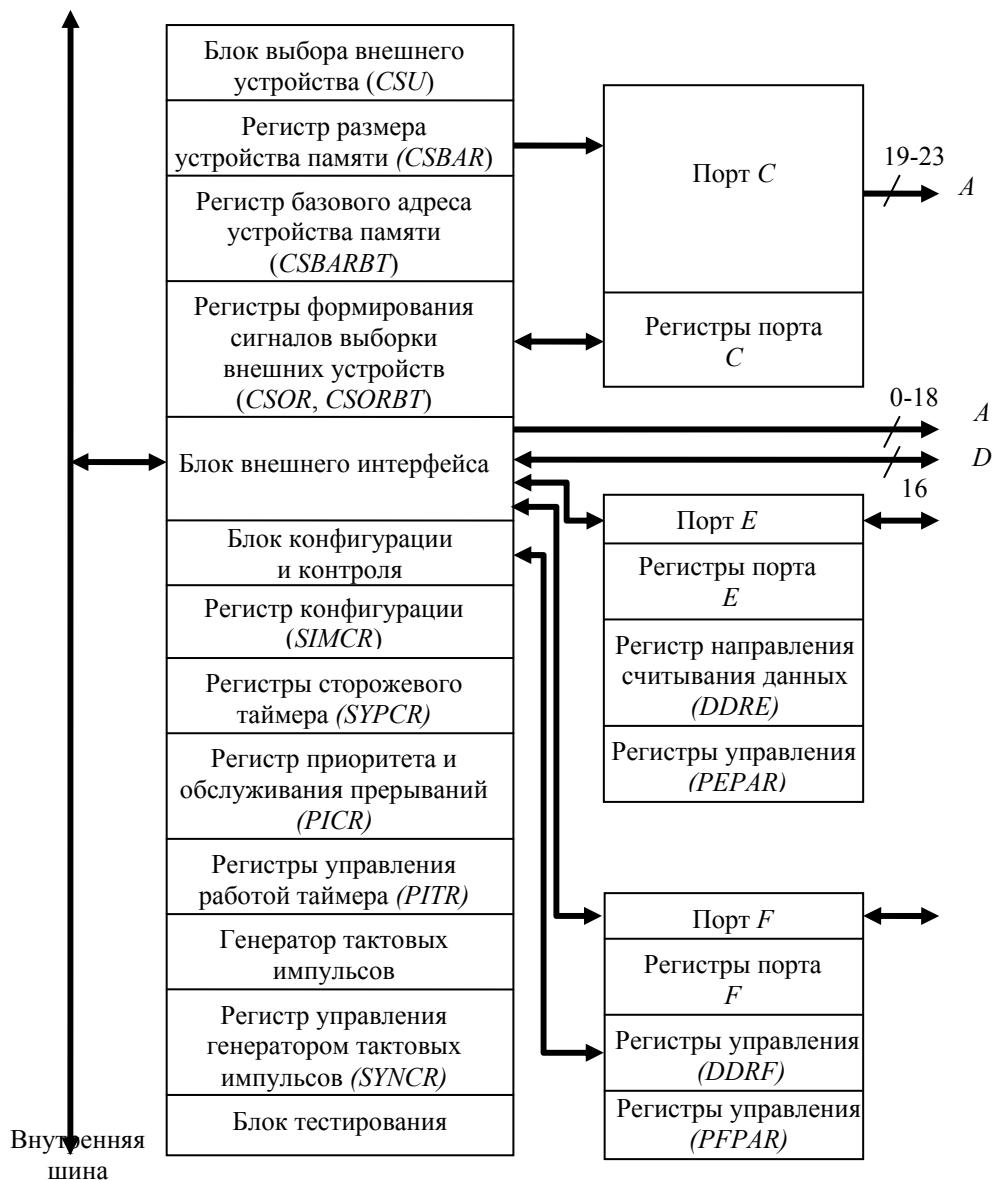


Рис. 3.14 Структура модуля системной интеграции (SIM) 32-разрядного микроконтроллера серии MC683xx

Шестнадцати и тридцатидвухразрядные микроконтроллеры серии MC683xx используют стандартную внутримодульную шину (IMB), позволяющую применять одни и те же периферийные устройства в совокупности с различными микропроцессорами. Это дает возможность компилировать программы, написанные для 16-разрядного микроконтроллера, в том числе и программы обмена с периферийными устройствами, для 32-разрядного микроконтроллера.

На рис.3.13 изображена обобщенная структурная схема микроконтроллера серии MC683xx, в котором могут использоваться микропроцессоры *CPU 32*, *CPU 32+*, *CPU 030*.

Кроме микропроцессоров в состав микроконтроллера входят сопроцессоры и периферийные устройства (рис.3.13).

Микропроцессор *CPU 32* содержит 32-разрядные регистры адреса и данных. Он содержит 16 Мбайт памяти с динамическим изменением разрядности шины (8 или 16). Микропроцессор *CPU 32* обеспечивает быструю реакцию на сигнал прерывания и поддерживает режимы пониженного энергопотребления. Развитием микропроцессора *CPU 32* являются микропроцессоры *CPU 32+* и *CPU 030*. Микропроцессор *CPU 32+* имеет разрядность внешней шины данных, равную 32, и повышенную производительность. Микропроцессор *CPU 030* объединяет микропроцессор *CPU 32+*, конфигурируемую кэш-память команд и блок управления запоминающим устройством.

Модуль системной интеграции (*SIM*) обеспечивает функционирование микропроцессора и его работу совместно с другими устройствами. Модуль *SIM* (рис.3.14) содержит генератор тактовых импульсов, блок внешнего интерфейса, организующий работу микроконтроллера с внешними устройствами, блок выбора внешнего устройства, блок конфигурации и контроля, который задает режимы работы других блоков и модулей микроконтроллера, блок тестирования, три порта *C*, *E*, *F*, и др. Блоки имеют регистры, определяющие функционирование модуля *SIM* и микропроцессора.

Модуль последовательного интерфейса (*QSM*) содержит последовательный периферийный синхронный интерфейс (*SPI*) с буферным ОЗУ очереди и последовательный коммуникационный интерфейс (*SCI*), обеспечивающий скорость передачи до 524 Кбод.

Сопроцессор временных интервалов (рис.3.15) предназначен для формирования и приема сигналов в реальном масштабе времени без участия основного микропроцессора. Сопроцессор, в отличие от обычного таймера, позволяет устанавливать взаимосвязь между каналами с помощью собственного микропроцессора, не обращаясь к основному микропроцессору. При этом для выполнения различных функций временных интервалов используется библиотека основных функций, находящаяся в ПЗУ микроконтроллера. Перечень функций, каждую из которых можно получить по любому из 16-ти независимым каналам временных интервалов (рис.3.15), приведен в табл.3.4 и табл.3.5

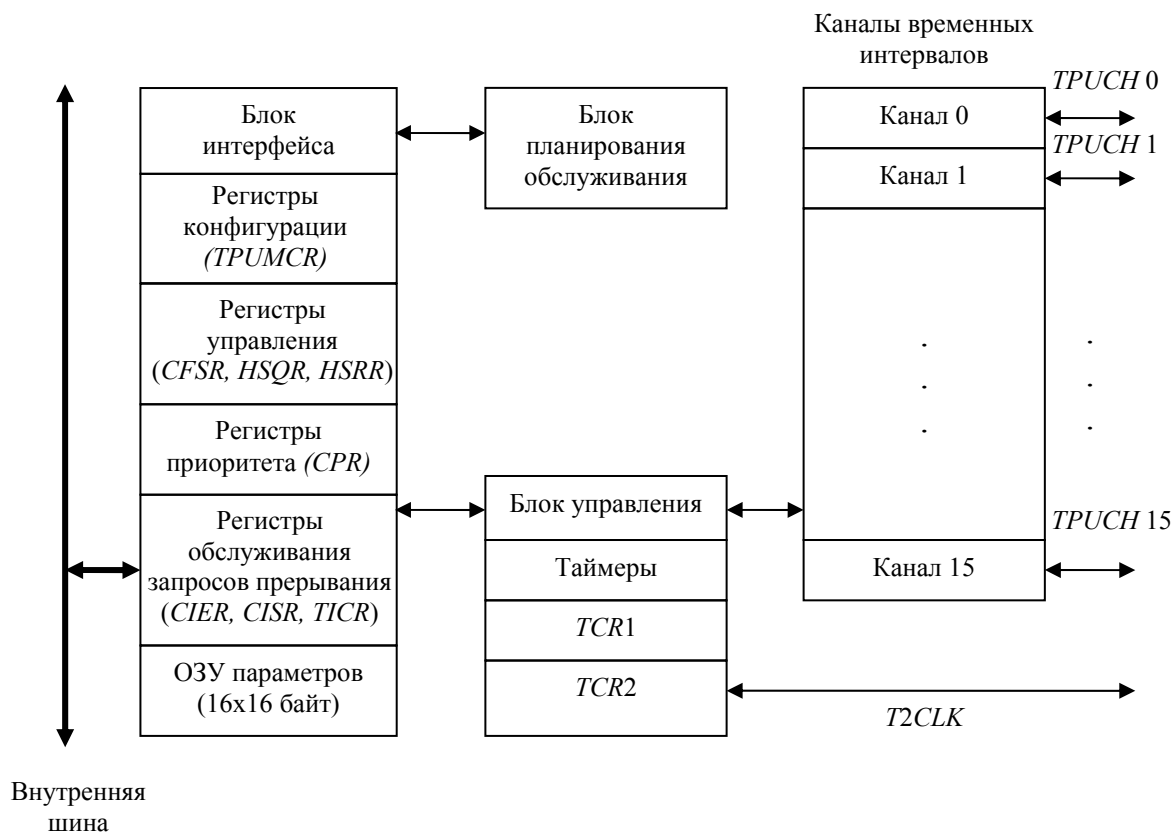


Рис. 3.15 Структура сопроцессора временных интервалов микроконтроллера серии MC683xx

Каждый из 16-ти двунаправленных каналов временных интервалов содержит регистр входной фиксации, регистр выходного сравнения и логические элементы защелок. Выбор и установка параметров функции (табл.3.4 и 3.5) осуществляется путем записи управляющих слов в соответствующие регистры управления, расположенные в ОЗУ параметров. Через ОЗУ параметров осуществляется также обмен параметрами между сопроцессором временных интервалов и основным микропроцессором. Сопроцессор выполняет обработку запросов на обслуживание, поступающих от каналов временных интервалов в соответствии с содержимым регистров приоритетов. Они реализуют алгоритмы обслуживания каналов временных интервалов в соответствии с микрокодом, считываемым либо из ПЗУ (микрокоды приведены в табл.3.4 и 3.5), либо из ОЗУ с эмуляцией сопроцессором временных интервалов. С помощью сигнала T2CLK (рис.3.15) осуществляется внешнее управление работой таймеров.

Таблица 3.4

Микрокод	Название	Описание
PWA	Period/Pulse-Width Accumula-	Изменение длительности импульсов (периодов) с накоплением в 16-или 24-разрядном регистре. Накопление

	tor	производится за программируемое число периодов (1-255) без прерывания. По завершении функции может устанавливаться связь с другим каналом временных интервалов либо формироваться сигнал прерывания
C	Output Compare	Формирование фронта импульсов в момент равенства содержимого регистра выходного сравнения и счетчика. Полярность импульса, периодичность, а также связь с другими каналами временных интервалов программируется
M	Stepper Motor	Функция управления шаговым двигателем поддерживает линейное ускорение и замедление с программируемой скоростью. Функция может выполняться 8-ью каналами временных интервалов.
SP	Position-Synchronized Pulse Generator	Формирование импульса программируемой длительности в определенный момент периода
PMA/PMM	Period Measurement with Additional or Missing Transition detect	Измерение периода импульсов. Обнаружение периода с отклонением, превышающим заданную величину.
ITC	Input Capture/Transition Counter	Фиксация значения одного из счетчиков в регистре входной фиксации в момент перепада сигнала на входе канала временных интервалов
WM	Pulse-Width Modulation	Формирование сигнала с широтно-импульсной модуляцией
IO	Discrete Input/Output	Использование каждого из 15 каналов временных интервалов для ввода/вывода
PWM	Synchronized PWM	Формирование сигнала широтно-импульсной модуляции с возможностью изменения длительности периода. Три возможных режима работы позволяют создавать сложные временные соотношения между каналами временных интервалов
DEC	Quadrature Decode	Управление по двум каналам временных интервалов регулированием физическим процессом

Модуль таймеров (*GPT*) представляет собой 11-канальный таймер и содержит счетчики с предделителями, каналы входной фиксации и выходного сравнения, широтно-импульсной модулятор и счетчик событий.

Модуль аналого-цифровых преобразователей (*ADC*) содержит восемь 10 и 8-разрядных каналов с программируемыми временами выборки/хранения. Модуль имеет несколько автоматических режимов преобра-

зования, 8 регистров результата и использует 3 формата представления данных.

Таблица 3.5

Микрокод	Название	Описание
TA	Programmable Time Accumulator	Накопление в 32-разрядном регистре параметров времени нахождения сигнала с высоким или низким потенциалом, а также периода сигнала за программируемое число периодов
OM	Queued Output Match	Формирование однократных, повторяющихся или непрерывной цепочки импульсов на основании таблицы в ОЗУ параметров.
SM	Table Stepper Motor	Управление ускорением и замедлением шагового двигателя с программируемым числом шаговых соотношений (до 58). Функция использует таблицу в ОЗУ параметров
QM	Frequency Measurement	Измерение частоты сигнала методом подсчета внешних импульсов за программируемый промежуток времени. Функция служит для измерения частоты высокочастотного сигнала
ART	Asynchronous Receiver /Transmitter	Обеспечение последовательного асинхронного обмена информацией по одному или двум каналам временных интервалов. Длина слова программируется от 1 до 14 бит. Поддерживается формирование и обнаружение бита четности. Максимальная скорость обмена составляет 100 Кбод. При скорости обмена 9600 бод сопроцессор временных интервалов может функционировать как 8 последовательных интерфейсов
ITC	New Input Transition Counter	Сопроцессор временных интервалов может фиксировать по одиночному событию или после заданного числа событий на внешнем входе не только содержимое выбранного счетчика, но и содержимое выбранной ячейки ОЗУ параметров
COMM	Multiphase Motor Commutation	Генерирование сигналов фазовой коммутации для различных бесколлекторных электродвигателей, включая трехфазные электродвигатели постоянного тока.
MCPWM	Multichannel PWM	Генерирование сигнала с широтно-импульсной модуляцией. Используется два канала временных интервалов и требуется один внешний логический элемент
HALLD	HALL Sensor Decode	Преобразование сигналов с датчиков Холла бесколлекторного двигателя, а также информации о направлении вращения, в число, необходимое для выполнения функции COMM. Функция ориентирована на работу с двумя или тремя датчиками

QD	Fast Quadrature Decode	Обеспечение обратной связи по положению, необходимой для управления двигателем.
----	------------------------	---

Некоторые микроконтроллеры серии MC683xx содержат АЦП с очередью преобразований (*QADC*). Такой аналого-цифровой преобразователь производит преобразование сигналов по 16 внутренним каналам и с внешним мультиплексором до 44 каналов, используя две независимые очереди и 32 регистра результата.

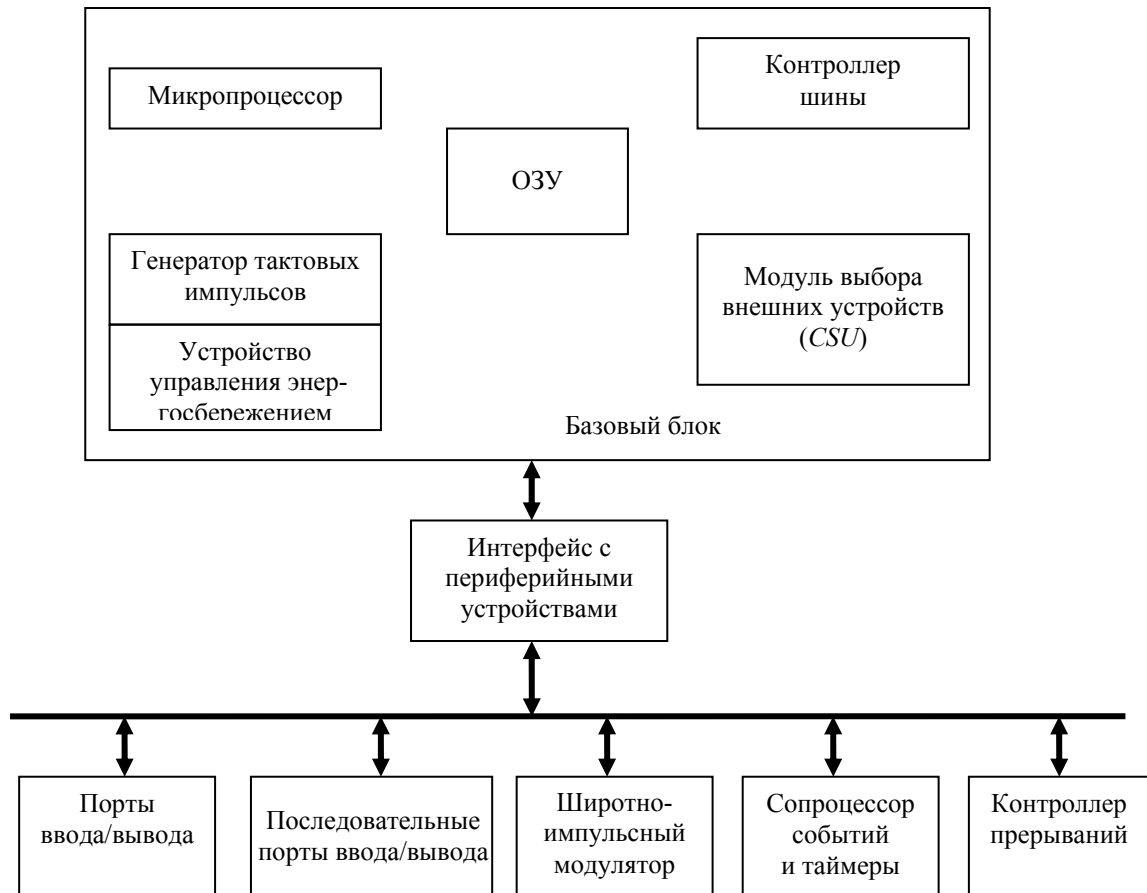


Рис. 3.16 Обобщенная структурная схема микроконтроллера 80C296SA

В устройствах управления и передачи сигналов, примерами которых являются устройства управления роботами и манипуляторами, устройства фильтрации в радиомодемах и радиотелефонах, аудио-, видео- и телевизионная техника, системы слежения за объектами в авиации и космосе и др., применяют микроконтроллеры, в которых реализуется обработка сигналов в реальном времени и функции цифровой фильтрации сигналов. Примером архитектуры таких микроконтроллеров является MCS-296. Они используются в устройствах приводов жестких дисков, сканерах, системах медицинской техники, радиомодемах телекоммуникационных систем, устрой-

ствах копировальной техники и т.д. Микроконтроллер с архитектурой MCS-296 представляет собой микроконтроллер со встроенными устройствами цифровой обработки сигналов. В этих микроконтроллерах используются следующие принципы организации.

- Увеличена производительность основного микропроцессора (*CPU*) за счет уменьшения времени выборки из запоминающих устройств команд и данных. Организовано единое адресное пространство для хранения команд и данных, при создании очереди команд из 10 байтов с приоритетом выборки для операндов.

- Использован конвейерный режим, позволяющий реализовать одновременно выполнение 4-х команд.

- Применено аппаратное перемножение и деление.

- В системе команд используются специальные команды цифровой обработки сигналов (умножение с накоплением, повторения, обновление данных в таблицах выборок и др.).

Обобщенная структурная схема микроконтроллера 80C296SA с архитектурой MCS-296 (рис.3.16) содержит базовый блок, в состав которого входят 32-разрядный микропроцессор, ОЗУ команд и данных, модуль выбора внешних устройств, контроллер шины, генератор тактовых импульсов, устройство управления энергосбережением и набор периферийных устройств. Периферийные устройства по принципу действия и функциональным возможностям аналогичны периферийным устройствам микроконтроллеров с архитектурой MCS-196 (см. рис.3.11). Следует отметить, что в микроконтроллере отсутствует аналого-цифровой преобразователь, что обусловлено возросшими требованиями к времени преобразования и сложностью размещения на кристалле быстродействующего АЦП.

Для пояснения особенностей структуры микроконтроллера со встроенными устройствами цифровой обработки сигналов на рис.3.17 приведена подробная структурная схема микроконтроллера 80C296SA. В качестве исполнительных устройств микроконтроллера используются 16-разрядное арифметико-логическое устройство, аппаратные перемножитель и делитель, 32-разрядный кольцевой регистр сдвига барабанного типа и 40-разрядный регистр-аккумулятор, 24-разрядный счетчик команд, регистр словосостояния программы (*PSW*) и счетчик числа циклов. Микроконтроллер реализует арифметические и логические операции и операции цифровой обработки сигналов (умножение, умножение с накоплением, сдвиг и др.). Кроме того, в состав микроконтроллера входят блок выравнивания



форматов команд с регистром очереди команд, блок упорядочивания команд и модуль интерфейса памяти.

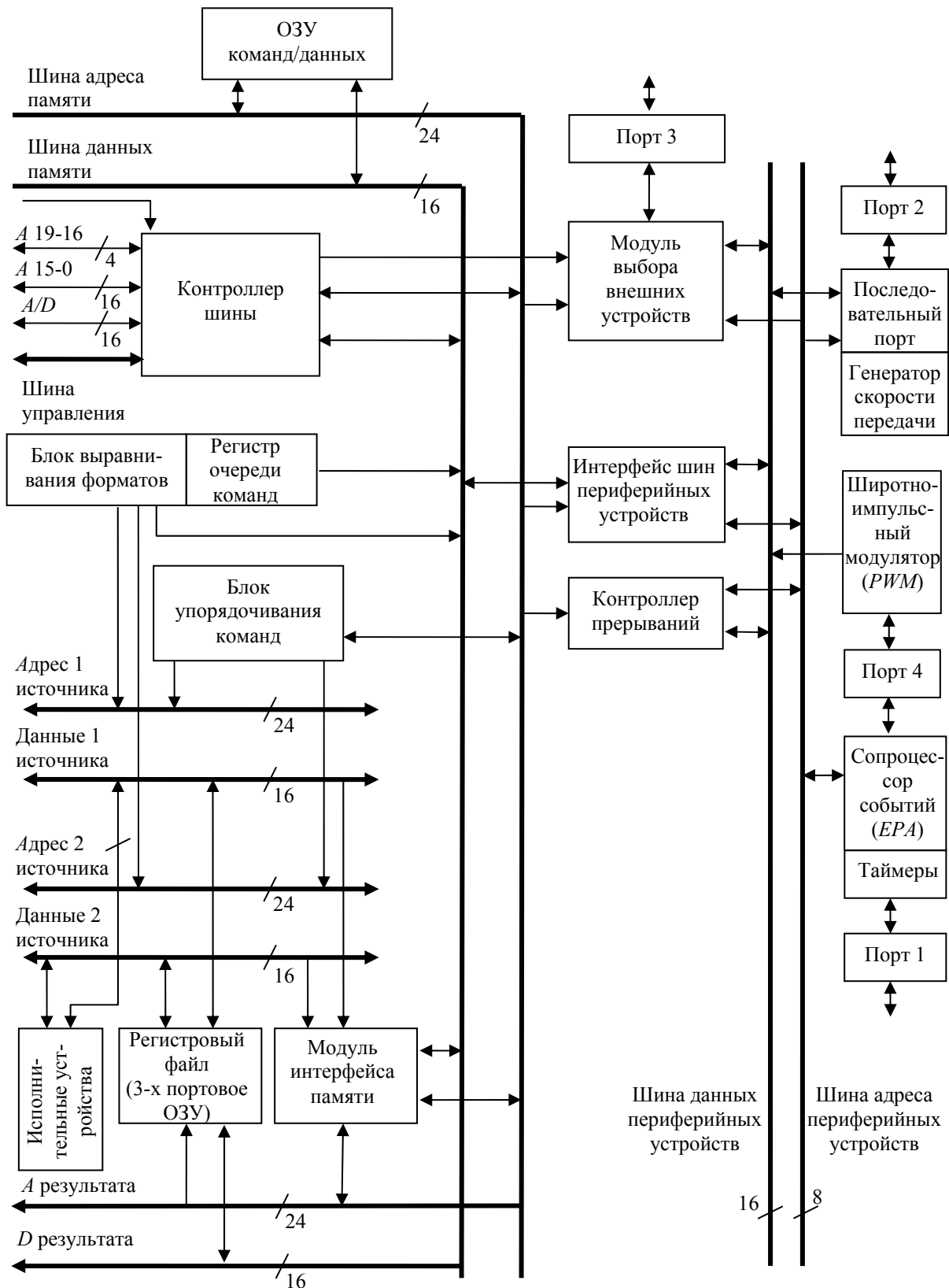


Рис. 3.17 Структурная схема микроконтроллера 80C296SA

Модуль интерфейса памяти обеспечивает предварительную выборку команд и размещение их в очереди на выполнение. Кроме того, этот модуль управляет считыванием операндов и записью в запоминающие устройства результатов обработки данных при работе конвейера команд. При этом модуль интерфейса памяти управляет работой внутренних 24-разрядной шины адреса памяти и 16-разрядной шиной данных. Управление работой внешних шин адреса и данных осуществляется от контроллера шины.

При вычислениях в микроконтроллере на входы основного микропроцессора одновременно подаются два операнда из ячеек запоминающего устройства с разными адресами. Для этого служат шины «Адрес 1, 2» и «Данные 1, 2» источника. Результат вычисления транспортируется по внутренним шинам  $A_{\text{результата}}$  и  $D_{\text{результата}}$  адреса и данных.

Все периферийные устройства, расположенные на кристалле, связаны между собой 8-разрядной шиной адреса и 16-разрядной шиной данных периферийных устройств.

Коммуникационный микроконтроллер представляет собой совокупность основного микропроцессора и скоростного коммуникационного сопроцессора (*СРМ*), содержащего микропроцессор, построенный на основе *RISC*-архитектуры. Коммуникационный сопроцессор организует обмен данными по нескольким каналам и поддерживает различные протоколы обмена. Коммуникационные микроконтроллеры применяются в цифровой телефонной связи, оборудовании сети *ISDN*, радиомодемах сотовой связи, аппаратуре обслуживания локальных сетей и других видах устройств систем связи. Структурная схема коммуникационного микроконтроллера МС 68360 приведена на рис.3.18. Микроконтроллер содержит основной микропроцессор (*CPU 32+*), осуществляющий общее управление, коммуникационный сопроцессор (*СРМ*), выполняющий прием и передачу потоков информации в соответствии с выбранным коммуникационным протоколом, и описанный выше модуль системной интеграции (*SIM*), реализующий подключение запоминающих и внешних устройств.

Коммуникационный сопроцессор функционирует под управлением входящего в его состав 32-разрядного микропроцессора с *RISC*-архитектурой. Кроме того, в структуру коммуникационного сопроцессора входят контроллеры прерываний и прямого доступа к памяти, ОЗУ и ПЗУ, таймеры и параллельные порты ввода/вывода. Контроллер прямого доступа к памяти содержит четырнадцать каналов последовательного прямого

доступа к памяти (*SDMA*) и два независимых канала общего назначения прямого доступа к памяти (*IDMA*).

В состав параллельных портов ввода/вывода входят четыре генератора частоты обмена.

Основной микропроцессор *CPU 32* имеет производительность 8.3 млн.операций/сек при тактовой частоте 33 МГц. Он содержит 32-разрядную шину адреса и 8/16/32-разрядную шину данных.

Модуль системной интеграции (*SIM*) имеет 32-разрядные шины адреса и данных, контроллер динамической памяти (*DRAM*), контроллер статической памяти (*SRAM*), четыре 16-разрядных таймера или два 32-разрядных таймера, включая сторожевой таймер, устройство тестирования и диагностики (*JTAG*).

Четыре последовательных коммуникационных контроллера (*SCC*) поддерживают наиболее распространенные в телекоммуникационных системах протоколы: *HDLC/SDLC*, *HDLC Bus*, *Ethernet*, *UART*, *Apple Talk*, *Transparent*, *ATM* и др.

Каждый из двух контроллеров управления обменом (*SMC*) работает со своим протоколом передачи данных, независимо от режимов работы каналов последовательного коммуникационного контроллера. Синхронный последовательный интерфейс (*SPI*) используется для передачи данных между микроконтроллерами.

Основные характеристики некоторых 16 и 32-разрядных микроконтроллеров приведены в табл.3.6.

Таблица 3.6

название	апоминающее устройство команд ЗУ	ЗУ	Таймеры	Число линий ввода/вывода	Последовательные порты	АЦП
MC68HC16 Y1	48 Кбайт	2Кбайт	Сопроцессор временных интервалов ( <i>TPU</i> ), Модуль таймеров ( <i>GPT</i> )	5	<i>SPI</i> <i>2SCI</i>	8 входов 10 бит
MC68HC91	48 Кбайт	4Кбайт	Сопроцессор вре-		<i>SPI</i>	8 вхо-

6Y1			менных интервалов ( <i>TPU</i> ), Модуль таймеров ( <i>GPT</i> )	5	2 <i>SCI</i>	дов 10 бит
MC68333	4 Кбайт	0,5Кбайт	Сопроцессор временных интервалов ( <i>TPU</i> ), ОЗУ 3Кбит	6	Модуль последовательного обмена ( <i>QSM</i> )	8 входов 10 бит
MC68335		8Кбайт	Сопроцессор временных интервалов ( <i>TPU</i> ), ОЗУ 2Кбит	7	Модуль последовательного обмена ( <i>QSM</i> )	
MC68336		4Кбайт	Сопроцессор временных интервалов ( <i>TPU</i> ), Модуль конфигурируемого таймера ( <i>CTM</i> ), ОЗУ 3,5Кбит	6	Модуль последовательного обмена ( <i>QSM</i> )	8 входов 10 бит
MC68360	4Кбайт Программируемое электрическим способом	2,5Кбайт;	Четыре таймера – счетчика ( <i>TC</i> )	02	Сдвоенный асинхронный приемопередатчик ( <i>DUAPT</i> ), 4 <i>SCC</i> , <i>SPI</i>	

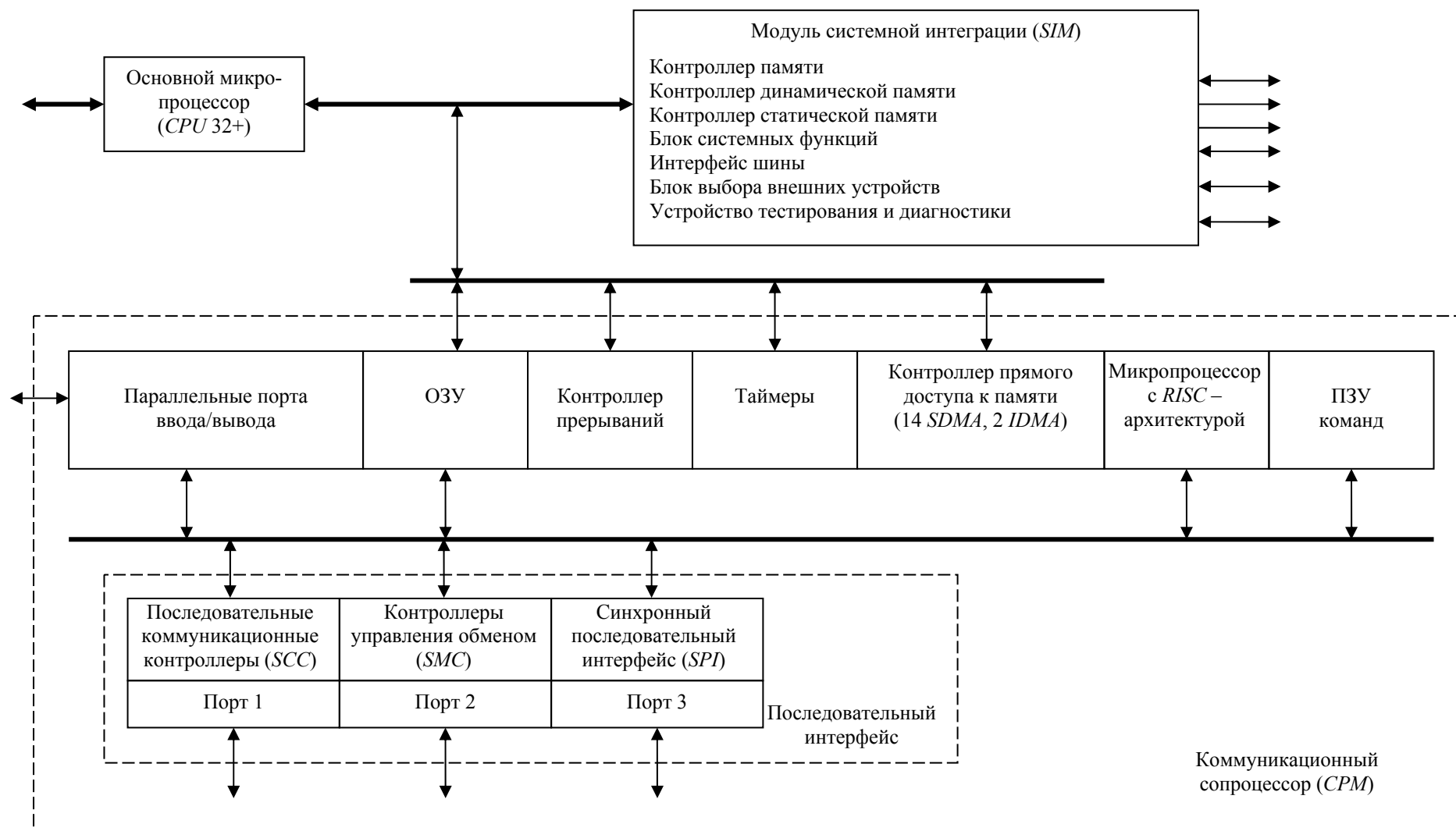


Рис. 3.18 Структурная схема коммуникационного микроконтроллера MC68360

В заключение отметим общие для шестнадцати и тридцатидвухразрядных микроконтроллеров особенности:

- высокая производительность, обеспечиваемая высокой тактовой частотой и наличием на кристалле сопроцессоров (коммуникационных, цифровой обработки сигналов, ввода-вывода, временных интервалов и др.);
- повышенная точность вычислений;
- унификация интерфейса для работы с запоминающими устройствами разных типов, с различными внешними устройствами, что достигается благодаря наличию в структуре микроконтроллеров модуля системной интеграции;
- возможность регулирования энергопотребления.

Наиболее перспективными являются микроконтроллеры, объединяющие на одном кристалле несколько типов микропроцессоров: цифровые сигнальные процессоры, коммуникационный процессор, поддерживающий различные протоколы обмена (*ISDN, HDLC, Ethernet, ATM* и др.), универсальный микропроцессор на основе *RISC* – архитектуры и сопроцессоры ввода-вывода.

### **3.2. Цифровые сигнальные процессоры**

Цифровые сигнальные процессоры применяются в радиомодемах телекоммуникационных систем, радиолокационных и гидролокационных станциях, радиоприемных устройствах спутниковых систем связи, в устройствах уплотнения данных и сжатия видеоизображения, для реализации процедур цифровой обработки сигналов в реальном масштабе времени и т.д. Отличительной особенностью реализации алгоритмов цифровой обработки сигналов является поточный характер поступления больших объемов данных в реальном масштабе времени. Это требует высокой производительности цифровых сигнальных процессоров при выполнении большого числа вычислительных арифметических операций и обеспечении возможности интенсивного обмена с внешними устройствами. Это достигается специфической архитектурой и проблемно-ориентированной системой команд цифровых сигнальных процессоров. Эволюция развития архитектуры, а именно, увеличение числа разрядов обрабатываемых данных, рост количества параллельно работающих арифметико-логических устройств, сокращение времени выполнения операций и переходов, наличие аналоговых входов-выходов и пр., позволит применить цифровые сигнальные процессоры в таких нетрадиционных областях радиотехники и связи, как построение мультимедийных систем, обработки

графической информации и изображений, ситуационный анализ, игровые задачи.

При построении цифровых сигнальных процессоров применяются следующие принципы.

- Модифицированная архитектура, использующая отдельные запоминающие устройства данных и команд и возможность обмена между ними.
- Сокращение длительности цикла выполнения команд.
- Конвейеризация работы узлов при реализации программы.
- Аппаратный перемножитель.
- Использование специальных команд, таких как умножение с накоплением, битовые операции, инверсия битов адреса для реализации быстрого преобразования Фурье и др.

Наиболее распространенными являются цифровые сигнальные процессоры серий TMS320Cxx, TMS320Cxxxx, ADSP21xx, ADSP210xx, i 960, 56xxx, 9600x. Архитектура различных серий цифровых сигнальных процессоров имеют много общего, но есть и отличия. Основные тенденции развития и совершенствования структур и систем команд проследим на примерах цифровых сигнальных процессоров указанных выше серий.

### **3.2.1 Цифровые сигнальные процессоры TMS320Cxx, ....., TMS320C6000**

Шестнадцатиразрядный цифровой сигнальный процессор TMS320C10 [12, 13] имеет производительность 5 млн. операций/сек, что обеспечивается эффективной системой команд и высокоразвитой конвейерной архитектурой. В нем все арифметические операции выполняются с фиксированной точкой в двоичном дополнительном коде.

Структурная схема TMS320C10 (рис.3.19) содержит арифметико-логическое устройство, регистр-аккумулятор, перемножитель и два устройства сдвига.

Арифметико-логическое устройство выполняет арифметические (сложение и вычитание) и логические операции с 32-разрядными словами. Один операнд находится в регистре-аккумуляторе, другой поступает из R-регистра перемножителя, либо по внутренней шине данных с возможностью сдвига на 0...15 разрядов.

В оперативном запоминающем устройстве размещается 256 шестнадцатиразрядных слов. Память данных разделена на страницы по 128 слов в каждой. При этом имеется возможность расширения памяти данных путем использования внешних ОЗУ.

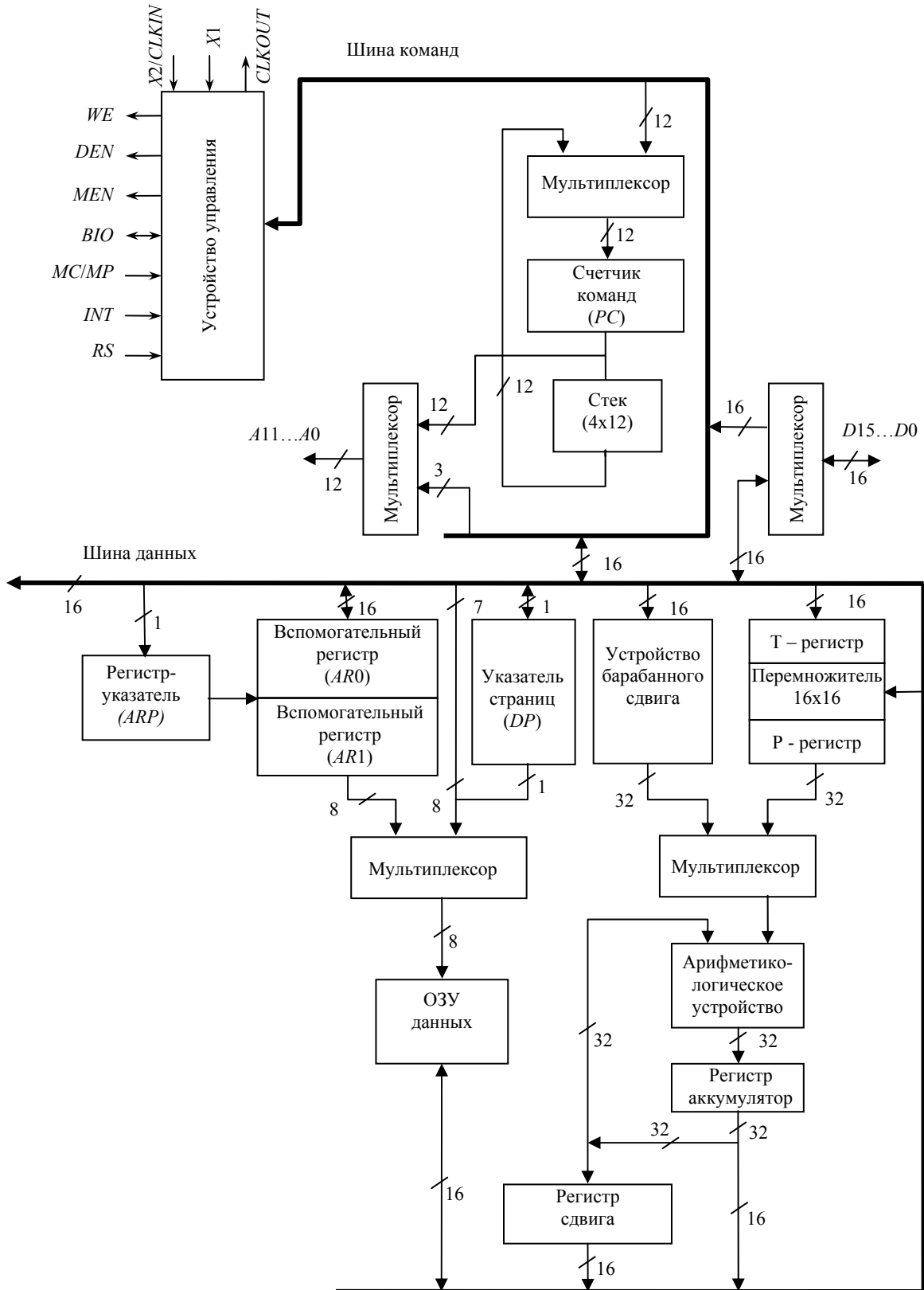


Рис. 3.19 Структурная схема цифрового сигнального процессора TMS320C10



Параллельный перемножитель шестнадцатиразрядных чисел состоит из трех узлов: входного Т-регистра, матричного перемножителя (16x16) и Р-регистра результата перемножения.

В цифровом сигнальном процессоре имеются два устройства сдвига: быстродействующее устройство барабанного сдвига, позволяющее получить сдвиг на 0...15 разрядов и регистр сдвига для пересылки данных из регистра-аккумулятора в ОЗУ данных со сдвигом на 0, 1 или 4 разряда.

При косвенной адресации используются вспомогательные регистры (AR0 и AR1). В этих регистрах 8 младших разрядов определяют адрес памяти данных. Выбор вспомогательного регистра осуществляется регистром-указателем (ARP). Указатель страницы (DP) используется для формирования адреса при прямой адресации памяти. Устройство управления принимает сигналы от внешних устройств, формирует для них управляющие сигналы и осуществляет синхронизацию их работы.

Рассмотрим подробнее отдельные узлы цифрового сигнального процессора TMS320C10. Основным блоком, который осуществляет синхронизацию и управление работой всех узлов цифрового сигнального процессора является устройство управления.

К входу X2/CLKIN подключается внешний генератор тактовых импульсов, частота которого лежит в интервале от 6.7 до 20МГц. При использовании встроенного генератора тактовых импульсов к входам X1 и X2/CLKIN подключается кварцевый резонатор. Выход CLKOUT устройства управления используется для контроля частоты тактового генератора. В устройстве управления формируется сигнал WE разрешения записи данных.

Сигнал готовности данных DEN указывает на то, что цифровой сигнальный процессор принимает данные с шины данных (Data Bus).

По сигналу разрешения выборки команды MEN происходит считывание команды по шине команд (Program Bus).

Кроме того, в устройство управления поступают сигналы сброса RS, внешнего прерывания INT и сигналы конфигурации памяти команд MC/MP. Тестируемые внешние сигналы BIO могут поступать от внешнего блока.

В цифровом сигнальном процессоре используется внешняя шина адреса A11-A0 и внешняя шина данных D15-D0.

Цифровой сигнальный процессор TMS320C10 имеет в своем составе 32-разрядное арифметико-логическое устройство и 32-разрядный регистр-аккумулятор, служащий для обеспечения вычислений с двойной точностью. Операции выполняются с 16-разрядными словами, принимаемыми из ОЗУ

данных или с 32-разрядными словами, принимаемыми из *P*-регистра. В дополнение к обычным арифметическим командам, арифметико-логическое устройство может выполнять логические операции, обеспечивая возможность управления перестановкой разрядов слов. Регистр-аккумулятор, разрядностью 32 бита, разделен на две части: старшее слово (биты 31-16) и младшее слово (биты 15-0). Команды предусматривают возможность сохранения, как старшего слова, так и младшего слова в ОЗУ данных.

Перемножитель выполняет операцию перемножения 16-разрядных слов, а результат выполнения этой операции формируется в виде 32-разрядного слова.

Множимое хранится в 16-разрядном *T*-регистре. Результат записывается в *P*-регистр. Значения множителя считываются из ОЗУ данных, либо содержатся в коде команды. Быстрый аппаратный перемножитель позволяет выполнять в реальном масштабе времени операции свертки, корреляции и фильтрации.

Два устройства сдвига (устройство барабанного сдвига и регистр сдвига) обеспечивают возможность управления размещением 16-разрядных данных в ОЗУ данных и в 32-разрядном регистре-аккумуляторе. Устройство барабанного сдвига обеспечивает возможность сдвига на 1 - 15 разрядов, что позволяет разместить произвольным образом 16-разрядное слово в ОЗУ данных при загрузке в 32-разрядный регистр-аккумулятор. Регистр сдвига обеспечивает возможность сдвига на 0, 1, или 4 разряда и используется при сохранении старшего или младшего слова регистра-аккумулятора в ОЗУ данных.

Оперативное запоминающее устройство цифрового сигнального процессора TMS320C10 разделено на два отдельных адресных пространства: память команд и память данных, как показано на рисунке 3.20. Конфигурация памяти команд зависит от состояния уровней сигнала *MC/MP* устройства управления. При *MC/MP*, равном логической единице, обеспечивается режим микроконтроллера, а при *MC/MP*, равном логическому нулю, - режим микропроцессора.

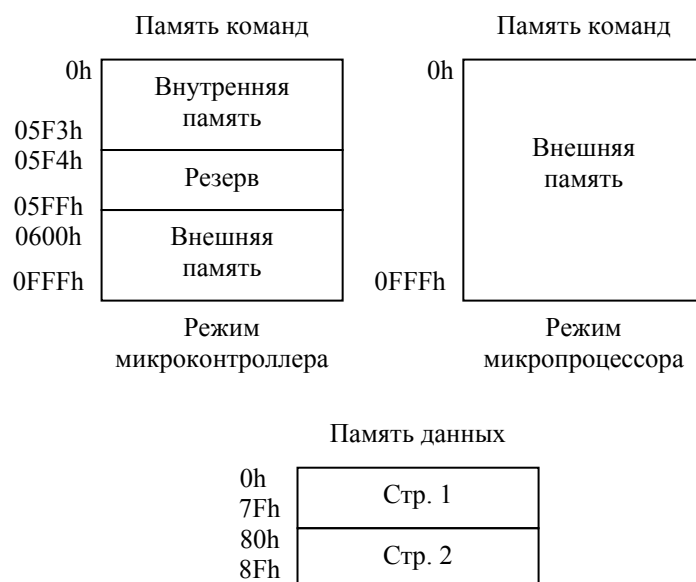


Рис. 3.20. Организация памяти цифрового сигнального процессора TMS320C10

В режиме микроконтроллера память команд разделена на три области: внутренняя память с адресами 0h-05F3h; резерв, который используется при необходимости загрузки дополнительных команд и внешняя память с адресами 0600h – 0FFFh. Адреса областей памяти записываются в шестнадцатичном коде. В режиме микропроцессора используется внешняя память с адресами 0h - 0FFFh. Память данных разделена на две равные части (Стр.1 и Стр.2) по 128 слов в каждой.

Для адресации внешних портов ввода - вывода используются три линии шины младших разрядов адреса. Шина данных осуществляет функцию ввода – вывода за два машинных цикла.

Два вспомогательных 16-разрядных регистра-указателя (*AR0*, *AR1*) используются для адресации памяти данных и организации циклов, а также, могут быть использованы для временного хранения данных. В регистре-указателе (*ARP*) находится бит состояния, который указывает на активный вспомогательный регистр. При значении этого бита, равного логической единице, активизируется вспомогательный регистр *AR1*, а при значении бита состояния, равного логическому нулю, – регистр *AR0*. . Содержание регистра содержит адрес команд. Указатель страниц ОЗУ данных содержит бит состояния, который указывает на адрес текущей страницы (Стр.1 и Стр.2) ОЗУ данных. Если бит состояния равен логическому нулю, то используются первые 128 слов, а при равенстве единице – вторые 128 слов. Счетчик команд (*PC*) представляет собой 12-разрядный регистр, используемый для адресации памяти команд. Счетчик команд содержит адрес команды, которая будет вы-

полняться следующей. Содержание регистра модифицируется после выполнения каждой команды, которая будет выполняться следующей.

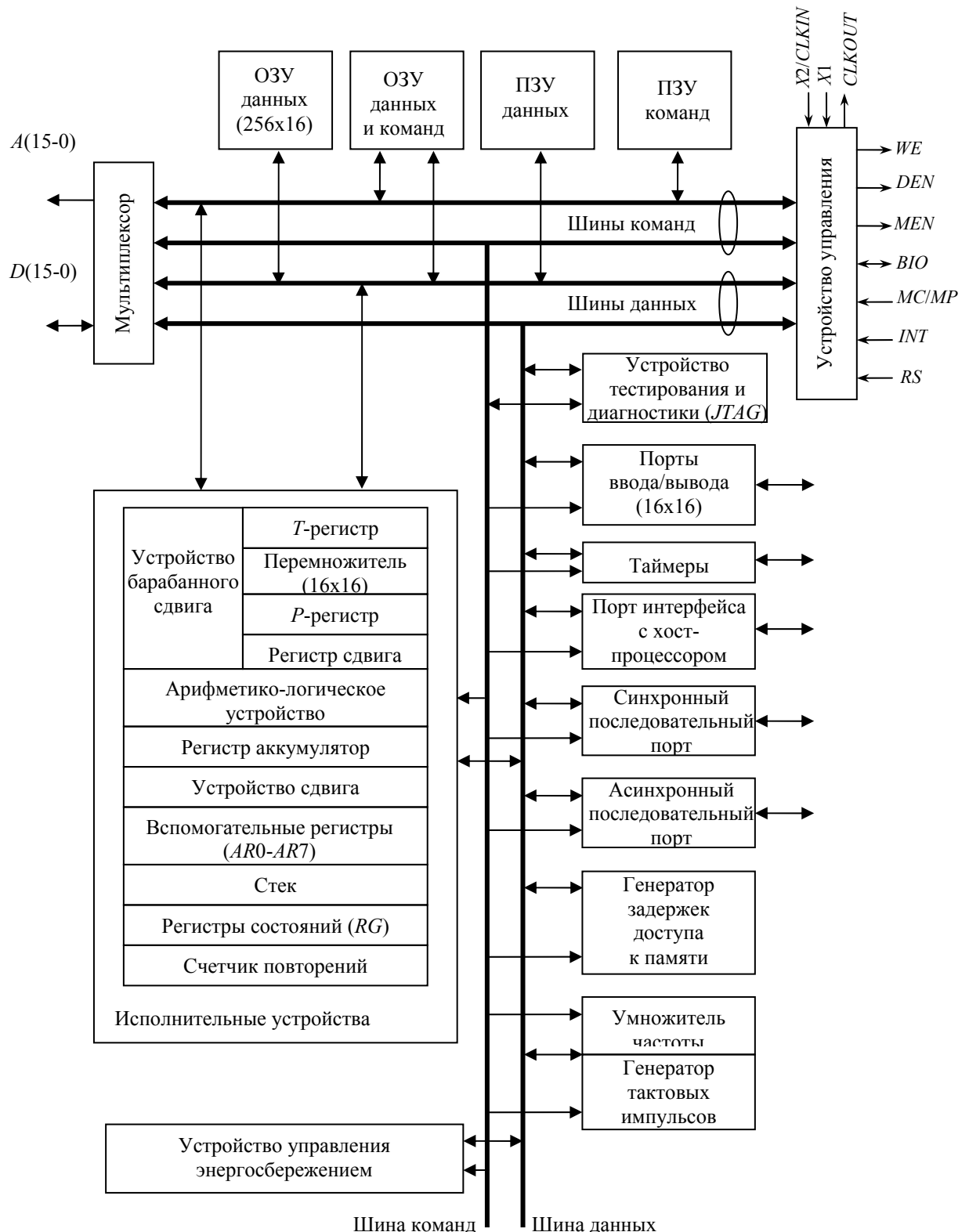


Рис. 3.21 Обобщенная структурная схема цифровых сигнальных процессоров серии TMS320C2x

Стек представляет собой запоминающее устройство, которое используется, чтобы сохранить содержимое счетчика команд при обслуживании пре-

рываний или при переходе на подпрограмму. Стек выполнен аппаратно и способен хранить до четырех 12-разрядных слов.

Аппаратное прерывание реализуется путем подачи сигнала логического нуля (*INT*) на вход устройства управления. Встроенная схема обработки прерывания начинает обслуживание прерывания путем перехода к ячейке памяти команд с определенным адресом. Адрес возврата (адрес следующей команды) сохраняется при этом в стеке.

Цифровые сигнальные процессоры серии TMS320C2x [12, 13], обобщенная структурная схема которых приведена на рис.3.21, характеризуются увеличением скорости вычислений и следующими нововведениями:

- набор команд поддерживает вычисления с плавающей точкой;
- увеличен объем внутреннего ПЗУ;
- возросло число таймеров и последовательных портов;
- увеличен объем стека и ОЗУ данных;
- реализована возможность прямого доступа к памяти;
- увеличено число портов ввода-вывода;
- введены команды побитной обработки данных и режим циклического повторения команды;
- для реализации процедуры запоминания при прерываниях в структуру введены теневые регистры;
- применено устройство тестирования и диагностики (*JTAG*);
- используется устройство управления энергосбережением;
- введен порт интерфейса с хост-процессором, позволяющий взаимодействовать с главным микропроцессором в многопроцессорной системе.

Устройство тестирования и диагностики представляет собой совокупность средств и операций, позволяющих проводить тестирование БИС без физического доступа к каждому ее контакту. Для реализации процедуры тестирования цифровой сигнальный процессор имеет набор специальных элементов периферийного сканирования и устройств управления их работой. При тестировании проверяется работа микросхемы путем считывания сигналов на контактах во время ее работы и происходит управление этими сигналами.

В цифровых сигнальных процессорах серии TMS320C2x содержатся ПЗУ команд объемом 4Кбайтх16 и три блока ОЗУ данных и команд (*B0*, *B1*, *B2*). Блок *B0*, объемом 256 битх16 используется либо как ОЗУ данных, либо как ОЗУ команд. Блоки *B1* и *B2* предназначены для хранения данных. Распределение памяти зависит от команд конфигурации памяти *CNFD* и *CNFP*

(рис.3.22). Команда *CNFD* (*Configure Data Memory*) определяет блок *B0*, как память данных (рис.3.21,а), а команда *CNFP* (*Configure Program Memory*) переводит блок *B0* в память команд (рис.3.22,б).

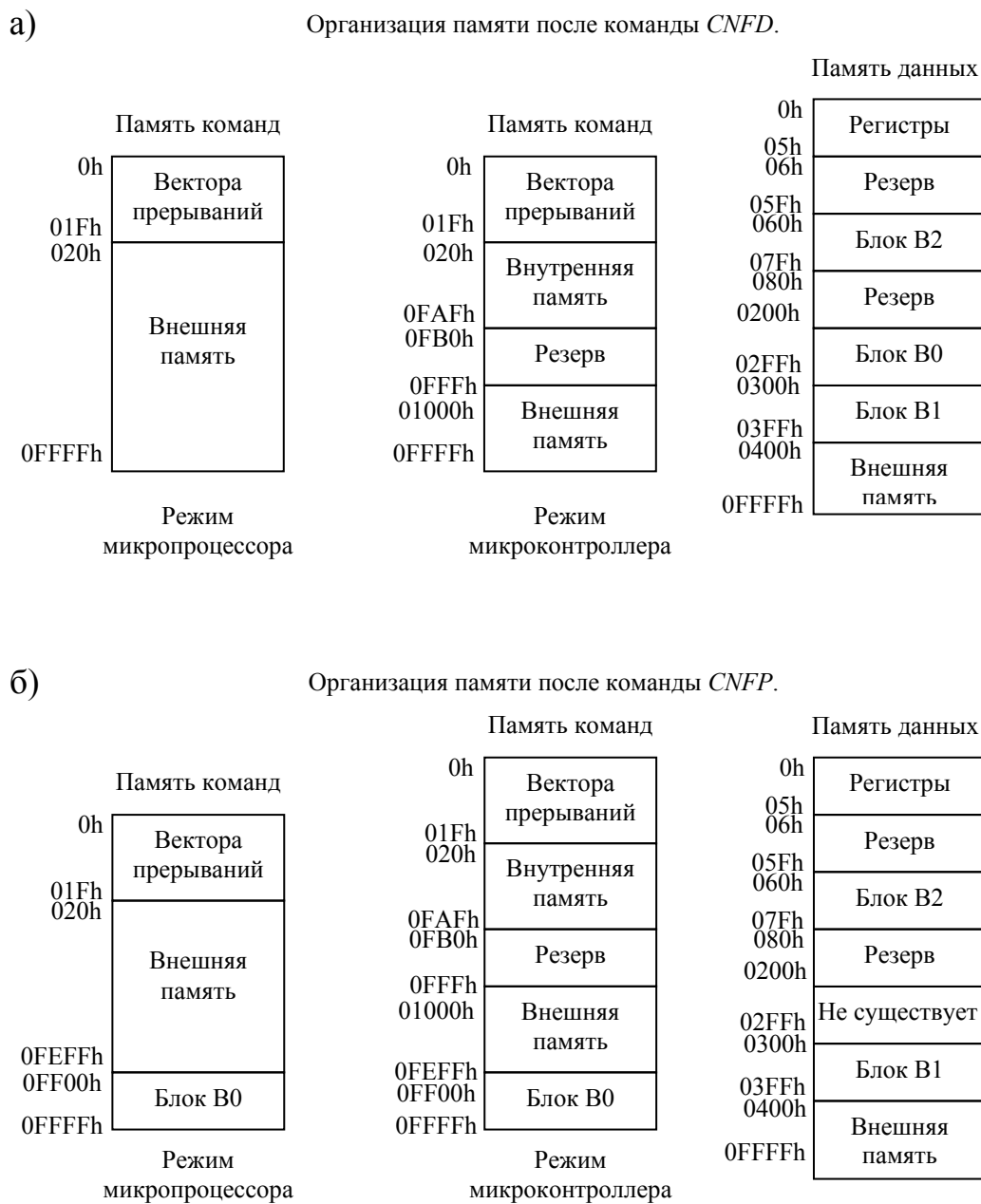


Рис. 9.22 Организация памяти цифровых сигнальных процессоров серии TMS320C2х

Использование внутренней памяти команд задается, как и в цифровом сигнальном процессоре TMS320C10 (см.рис.3.20), битом *MC/MP*, определяющим режим работы как микроконтроллера или микропроцессора. В первом режиме используется внутренняя память, а во втором – применяется

внешняя память команд. Следует отметить, что после подачи команды *CNFP* блок *B0* используется как память команд в обоих режимах работы.

В памяти данных расположено шесть регистров с адресами 0h – 05h:

- 16-разрядный приемный регистр последовательного порта, адресуемый как ячейка памяти данных с адресом 0;
- 16-разрядный передающий регистр последовательного порта, адресуемый как ячейка памяти данных с адресом 1;
- 16-разрядный регистр таймера, адресуемый как ячейка памяти данных с адресом 2;
- 16-разрядный регистр периода таймера, адресуемый как ячейка памяти данных с адресом 3;
- 6-разрядный регистр маски прерываний, адресуемый как ячейка памяти данных с адресом 4;
- 8-разрядный регистр глобальной памяти, адресуемый как ячейка памяти данных с адресом 5.

Полное адресное пространство цифровых сигнальных процессоров серии TMS320C2x состоит из памяти команд (64Кбайт) и памяти данных (64Кбайт). Эти процессоры имеют длительность цикла выполнения команд 80...200нС, объем внутреннего ОЗУ 544 бит x 16 или 1568 бит x 16, ПЗУ команд емкостью 4Кбит x 16. Постоянное запоминающее устройство команд может быть перепрограммируемым с ультрафиолетовым стиранием.

Особенностью этих цифровых сигнальных процессоров является 16 шестнадцатиразрядных параллельных портов ввода-вывода, таймер и последовательный порт для прямого подключения модема.

Цифровые сигнальные процессоры серии TMS320C3x, выполняют вычисления в формате с плавающей точкой (рис.3.23). Их отличает гибкая система команд, аппаратная поддержка операций с плавающей точкой, расширенная система адресации, увеличенное адресное пространство, поддержка языка высокого уровня. Архитектуру цифровых сигнальных процессоров серии TMS320C30 характеризует следующее.

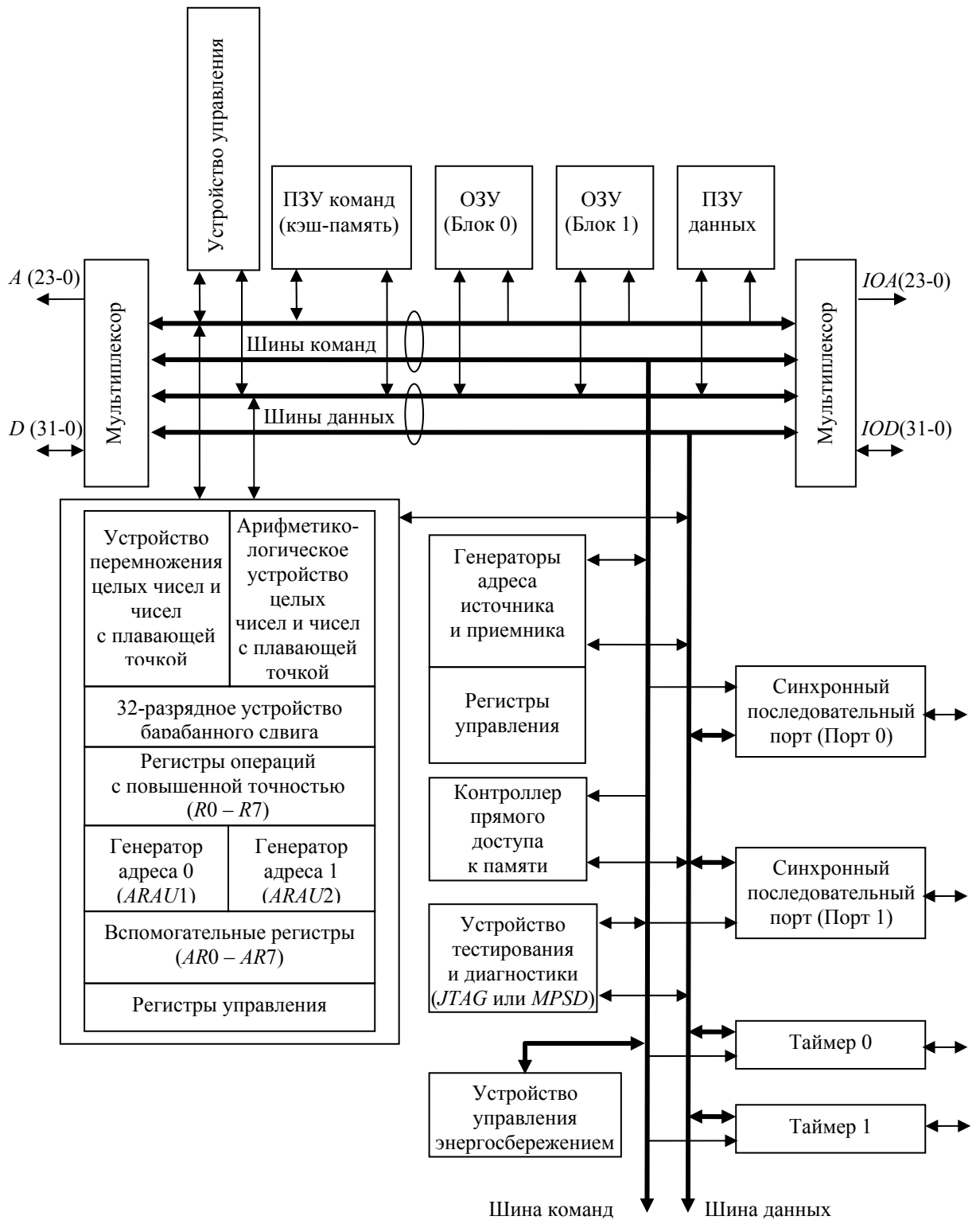


Рис. 3.23 Обобщенная структурная схема цифровых сигнальных процессоров серии TMS320C3x



Каждая команда выполняется за один такт длительностью 60 нс.

- Имеется два блока ОЗУ (Блок 0 и Блок 1) по 1 Кбайту 32-разрядных слов.
- ПЗУ данных емкостью 4Кбайт 32-разрядных слов.
- ПЗУ команд (кэш-память) содержат шестьдесят четыре 32 - разрядных слова.
- Используются 32-разрядные слова команд и данных, а также 24-разрядные адреса.
- Введено 32-разрядное устройство перемножения чисел с плавающей точкой и целых чисел.
- Применено 32-разрядное арифметико-логическое устройство, выполняющее операции над числами с плавающей точкой, операции с целыми числами и логические операции.
  - 32-разрядный кольцевой сдвиговый регистр.
  - Введены восемь регистров операций с повышенной точностью.
  - Используются два генератора адреса с восемью вспомогательными регистрами.
  - Контроллер прямого доступа к памяти используется для параллельного выполнения команд ввода-вывода.

При выполнении операций перемножения с плавающей точкой на входы устройства перемножения (рис.3.23) подаются 32-разрядные числа с плавающей точкой, а результатом является 40-разрядное число также с плавающей точкой. При перемножении целых чисел входные данные содержат 24 разряда, а результат - 32 разряда. Арифметико-логическое устройство работает с 32-разрядными целыми числами, 32-разрядными логическими переменными и 40-разрядными числами с плавающей точкой. Данные на выходе устройства перемножения и арифметико-логического устройства хранятся в 32-разрядном целочисленном формате или 40-разрядном формате с плавающей точкой. Цифровые сигнальные процессоры серии TMS320C3x обладают способностью выполнять параллельно за один цикл операции перемножения и сложения (вычитания) целых чисел или чисел с плавающей точкой. Блок регистров содержит 28 регистров, которые могут использоваться устройствами перемножения и арифметико-логического устройства. Восемь из них (R0 - R7) служат для операций с повышенной точностью, т.е. для операций над 40-разрядными числами с плавающей точкой и над 32-разрядными целыми числами. Вспомогательные регистры (AR0 - AR7) предназначены для

формирования адресов. Они могут использоваться как 32-разрядные регистры общего назначения. Два связанных с вспомогательными регистрами арифметических устройства (*ARAU1* и *ARAU2*) могут формировать по два адреса за один цикл. Эти устройства работают параллельно с устройством перемножения и арифметико-логическим устройством. Они обеспечивают режимы относительной адресации и индексной адресации. Регистры управления обслуживают адресацию, управление стеком, контроль состояния цифрового сигнального процессора, повторение блоков команд, прерывания.

Полный объем памяти цифровых сигнальных процессоров серии TMS320C30 составляет 16 Мбайт тридцатидвухразрядных слов. Машинное слово содержит 32 разряда, причем вся адресация выполняется по словам. В пределах одного цикла допускается выполнение двух выборок данных из каждого ОЗУ (Блок 0 и Блок 1) и ПЗУ. Например, в одном цикле возможна выборка из ПЗУ слова команды и слова данных. Раздельные шины команд, данных и прямого доступа к памяти допускают параллельное выполнение вызова команд, чтение и запись данных и операции с прямым доступом к памяти. Управление ресурсами памяти и шинами осуществляется контроллером прямого доступа к памяти. Так, стандартным является режим, при котором вызывается команда из ПЗУ команд и производится выборка двух данных из ОЗУ (Блок 0), а затем перенос данных в режиме прямого доступа к памяти в ОЗУ (Блок 1). ПЗУ команд, объемом 64 тридцатидвухразрядных слова, позволяет максимизировать быстродействие цифрового сигнального процессора. В ПЗУ команд хранятся часто повторяемые участки программы, которые могут вызываться из памяти, что значительно уменьшает число необходимых обращений к внешней памяти. Это дает возможность хранить всю программу во внешних запоминающих устройствах с малым быстродействием.

В цифровых сигнальных процессорах серии TMS320C3x предусмотрены две разновидности формата целых чисел со знаком: 16-разрядный формат, используемый для непосредственно заданных в команде целочисленных операндов; 32-разрядный формат целых чисел с обычной точностью.

Во всех форматах с плавающей точкой предполагается нормализованное представление чисел, что позволяет дополнительно высвободить один разряд на повышение точности. Первый формат представляет собой 16-разрядный короткий формат данных с плавающей точкой, используемый для непосредственно заданных в команде операндов, у которых 4 разряда отведено под порядок, один разряд под знак и одиннадцать разрядов под мантиссу. Второй формат является форматом с обычной точностью, имеющий 8-разрядный по-

рядок, один знаковый разряд и 23-разрядную мантиссу. Третий формат обеспечивает повышенную точность с 8-разрядным порядком, одним знаковым разрядом и 31-разрядной мантиссой.

Цифровых сигнальные процессоры серии TMS320C3x имеют контроллер прямого доступа к памяти и специализированные шины адреса и данных, что позволяет контроллеру читать и записывать информацию в память, не прерывая работу программы. Благодаря этому появляется возможность сопрягать процессоры серии TMS320C3x с внешними запоминающими устройствами и периферийными устройствами (аналого-цифровыми преобразователями, последовательными портами и т.д.) без снижения производительности вычислений. Контроллер прямого доступа к памяти содержит собственные генераторы адресов, входной и выходной регистры и счетчик пересылок данных.

Во внутреннем ПЗУ данных содержится начальный загрузчик. Он позволяет произвести загрузку исполняемого кода в ОЗУ данных и исполнение этого кода. Исполняемый код может быть загружен из внешнего 32- или 16-разрядного ПЗУ, хост-процессора или из последовательного порта.

В цифровом сигнальном процессоре имеются следующие периферийные устройства:

- Два 32-разрядных таймера/счетчика событий с возможностью работы, как от внутреннего, так и от внешнего тактового сигнала.
- Синхронные последовательные порты (Порт 0 и Порт 1) с возможностью обмена 8- 16- или 32-разрядными данными. Они могут использоваться для подключения ЦАП, АЦП, модемов и других внешних устройств, а также для объединения цифровых сигнальных процессоров в многопроцессорных системах. Порты имеют программируемую частоту генерации тактовых импульсов и синхронизации, а также программируемые режимы работы. Последовательные порты могут также использоваться как дополнительные таймеры.

Модификацией серии TMS320C3x являются цифровые сигнальные процессоры обработки сигналов с плавающей точкой серии TMS320C4x. Они сочетают высокую производительность TMS320C3x и высокопроизводительный контроллер прямого доступа к памяти, имеющий до 12 параллельных портов прямого доступа к памяти и обеспечивающий суммарную скорость ввода-вывода до 488Мбайт в секунду. Цифровые сигнальные процессоры этой серии имеют до 6 высокоскоростных синхронных последовательных

портов (линков), что позволяет отнести их к высокопроизводительным коммуникационным процессорам.

Цифровые сигнальные процессоры серии TMS320C54xx являются развитием архитектуры TMS320C2x, но обладают большим быстродействием, расширенным набором команд. Они более приспособлены для решения задач обработки сигналов в реальном масштабе времени.

К особенностям процессоров серии TMS320C54xx можно отнести следующие:

- программная совместимость с цифровыми сигнальными процессорами серии TMS320C10, TMS320C2x, и TMS320C20xx;
- уменьшенная длительность машинного цикла;
- увеличенный объем пространства памяти, так что полное адресное пространство составляет 192 Кбайт шестнадцатиразрядных слов;
- одиннадцать теневых регистров для сохранения предыдущих значений основных регистров при выполнении операций прерывания и выполнения подпрограмм;
- введение возможности многократного повторения блока команд;
- введение новых команд, выполняющих общие для цифровой обработки сигналов операции, такие, например, как обработка звена симметричного КИХ-фильтра.

На рис. 3.24 представлена обобщенная структурная схема цифровых сигнальных процессоров серии TMS320C54xx. В их составе имеется: арифметико-логическое устройство (*CALU*), параллельный логический модуль (*PLU*) и вспомогательное арифметическое устройство для работы с регистрами (*ARAU*).

Арифметико-логическое устройство предназначено для выполнения арифметических операций над целыми числами. Оно содержит 16-разрядный параллельный перемножитель, 32-разрядное арифметико-логическое устройство (*ALU*), 32-разрядный регистр-аккумулятор (*ACC*), 32-разрядный сумматор (*ACCB*), дополнительные регистры сдвига результатов суммирования и перемножения.

Четыре внутренние шины и два генератора адреса позволяют параллельно выполнять операции с несколькими операндами и снижают время доступа к памяти.

Сорокаразрядный сумматор и два сорокаразрядных регистра-аккумулятора позволяют производить параллельную обработку команд в одном цикле. Второй дополнительный сорокаразрядный регистр-аккумулятор

на выходе перемножителя позволяет выполнять операции, например, два суммирования и перемножения параллельно, которые не входят в конвейерные процедуры.

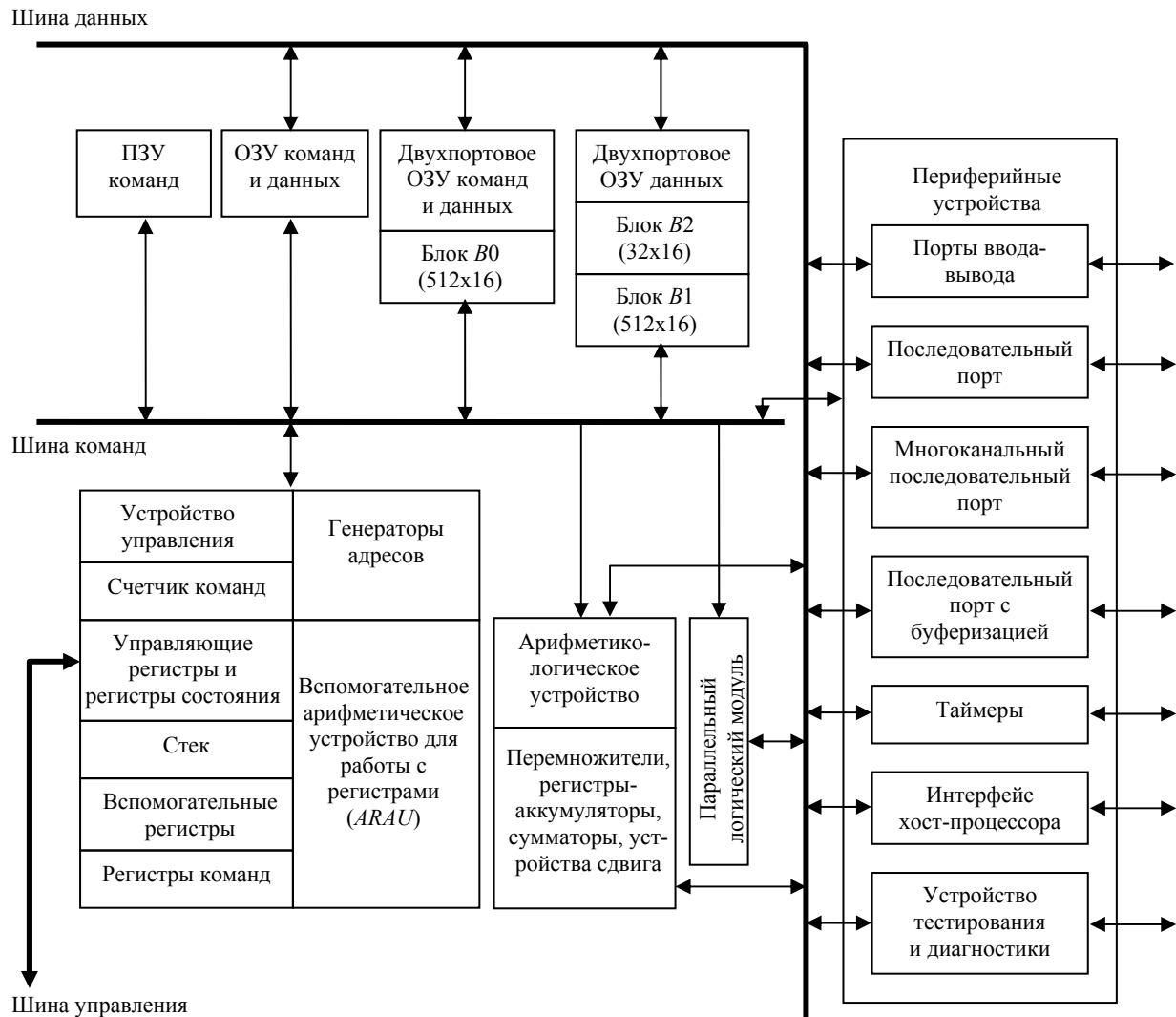


Рис. 9.24 Обобщенная структурная схема цифровых сигнальных процессоров серии TMS320C54xx

Перемножитель позволяет выполнять 16-разрядное перемножение чисел как со знаком, так и чисел без знака с округлением и контролем переполнения в одном командном цикле.

Параллельный логический модуль (PLU) работает параллельно с арифметико-логическим устройством. Он выполняет логические операции или операции манипуляции с отдельными битами. Параллельный логический модуль может устанавливать, очищать, проверять или изменять биты в регистре состояния, управляющих регистрах или в любой ячейке памяти данных.

Вспомогательные регистры ( $AR0-AR7$ ) связаны с вспомогательным арифметическим устройством для работы с регистрами ( $ARAU$ ), которое может автоматически индексировать текущий вспомогательный регистр. Это позволяет изменить адрес ячейки памяти данных после выполнения команды. В результате для изменения адреса операнда не требуется использовать арифметико-логическое устройство ( $CALU$ ).

Вспомогательный регистр  $AR0$ , в котором содержатся восемь младших битов кода команды, соединен с одним из входов вспомогательного арифметического устройства для работы с регистрами. На другой вход вспомогательного арифметического устройства подается адрес, указываемый текущим вспомогательным регистром  $AR1-AR7$ .

Вспомогательное арифметическое устройство для работы с регистрами выполняет следующие операции.

- Сложение содержимого текущего вспомогательного регистра с постоянной величиной, содержащейся в регистре  $AR0$ :

$$AR(ARP) + AR0 \rightarrow AR(ARP).$$

- Вычитание из содержимого текущего вспомогательного регистра постоянной величины, содержащейся в регистре  $AR0$ :

$$AR(ARP) - AR0 \rightarrow AR(ARP).$$

- Добавление единицы к содержимому вспомогательного регистра:  
 $AR$

$$(ARP) + 1 \rightarrow AR(ARP).$$

- Вычитание единицы из содержимого вспомогательного регистра:  
 $AR(ARP) - 1 \rightarrow AR(ARP)$ .

- Сложение содержимого текущего индексного регистра с 8-битной постоянной величины, находящейся в коде команды:

- $AR(ARP) + IR(7-0) \rightarrow AR(ARP)$ .

- Вычитание из содержимого текущего индексного регистра 8-битной постоянной величины, находящейся в коде команды:

- $AR(ARP) - IR(7-0) \rightarrow AR(ARP)$ .

В цифровых сигнальных процессорах серии TMS320C54xx имеются следующие периферийные устройства: таймеры, параллельные порты ввода - вывода, интерфейс хост-процессора ( $HPI$ ) для организации связи с центральным процессором в многопроцессорной системе, последовательный порт, последовательный порт с буферизацией, многоканальный последовательный порт с временным разделением каналов, устройство тестирования и диагностики.

Процессоры этой серии выполняют от 40 до 200 миллионов команд в секунду. Они применяются в устройствах обработки голоса и данных в сотовых телефонах, пейджерах, в системах передачи голоса в *IP*-телефонии и персональных информационных системах. Цифровые сигнальные процессоры серии TMS320C54xx содержат как процессоры со сверхнизким потреблением тока и напряжением питания 1.2В, так и высокопроизводительные процессоры с тактовой частотой работы до 160МГц. Это позволяет реализовать алгоритмы работы вокодера и эхоподавителя, многоканальных радиоприемных устройств.

Цифровые сигнальные процессоры серии TMS320C6xxx включает в себя две группы 32-разрядных процессоров: с фиксированной точкой (TMS320C62xx) и с плавающей точкой (TMS320C67xx). Высокая производительность этих процессоров достигается за счет использования архитектуры с длинным командным словом (*VLIW*), эффективных аппаратных решений и средств разработки программ. Производительность цифровых сигнальных процессоров серии TMS320C62xx составляют от 1200 до 2400 миллионов команд в секунду. Процессоры TMS320C67xx позволяют выполнять более одного миллиарда операций с плавающей точкой в секунду.

Цифровые сигнальные процессоры TMS320C67xx совместимы по командам и по контактам БИС с аналогичными процессорами TMS320C62xx. Это дает возможность реализовав программным способом алгоритмы обработки данных на цифровых сигнальных процессорах с плавающей точкой, перейти к процессорам с фиксированной точкой и наоборот. Такая концепция реализации алгоритмов, предполагает переход от аппаратно-ориентированной среды к программным моделям, что делает этот процесс более быстрым, дешевым и простым. Для поддержки этой концепции средства отладки программ выполнены так, чтобы дать возможность максимально использовать преимущества архитектуры цифровых сигнальных процессоров серии TMS320C6xxx. Средства отладки программ создают среду, которая дает возможность оптимизировать производительность устройств и минимизирует технические проблемы при разработке аппаратного и программного обеспечения. Средства отладки программ включают в себя *C*-компилятор, устройство оптимизации кода ассемблера и отладчик. С помощью этих средств отладки происходит перевод написанных на языке высокого уровня алгоритмов в программный код, оптимизированный под параллельную структуру цифровых сигнальных процессоров.

Средства отладки программ позволяют использовать моделирование алгоритмов в среде *Matlab*, с последующей трансляцией файлов через C-компилятор в код ассемблера. Цифровые сигнальные процессоры серии TMS320C6xxx (рис.3.25) содержат два функциональных модуля: Модуль 1 (регистры 1 и исполнительные блоки  $L1, S1, M1, D1$ ) и Модуль 2 задержки на каждом (регистры 2, исполнительные блоки  $L2, S2, M2, D2$ ). Каждый модуль состоит из перемножителя чисел, разрядностью  $16 \times 16$  ( $M$ ), три арифметико-логические устройства ( $S, L, D$ ) и блок буферных регистров. Арифметико-логические устройства  $S1, S2, L1, L2$  серии TMS320C6xxx выполняют арифметические, логические операции и операции перехода. При этом результаты выполнения этих операций формируются **без такте** работы цифрового сигнального процессора. Блок буферных регистров состоит из шестнадцати 32-разрядных регистров. Четыре исполнительных блока каждого модуля имеют произвольный доступ к блоку буферных регистров 1 и 2. Кроме того, модули 1 и 2 имеют шину, соединяющую блоки буферных регистров каждого модуля. Другой особенностью архитектуры цифровых сигнальных процессоров серии TMS320C6xxx является использование стратегии сохранения или загрузки. При этом два арифметико-логические устройства  $D1$  и  $D2$  используются для организации передачи данных между блоком буферных регистров ОЗУ данных и памятью. Эти же арифметико-логические устройства  $D1$  и  $D2$  управляют работой шины адресов, что позволяет использовать адрес, сформированный в одном блоке буферных регистров, для операций с данными в другом блоке буферных регистров.



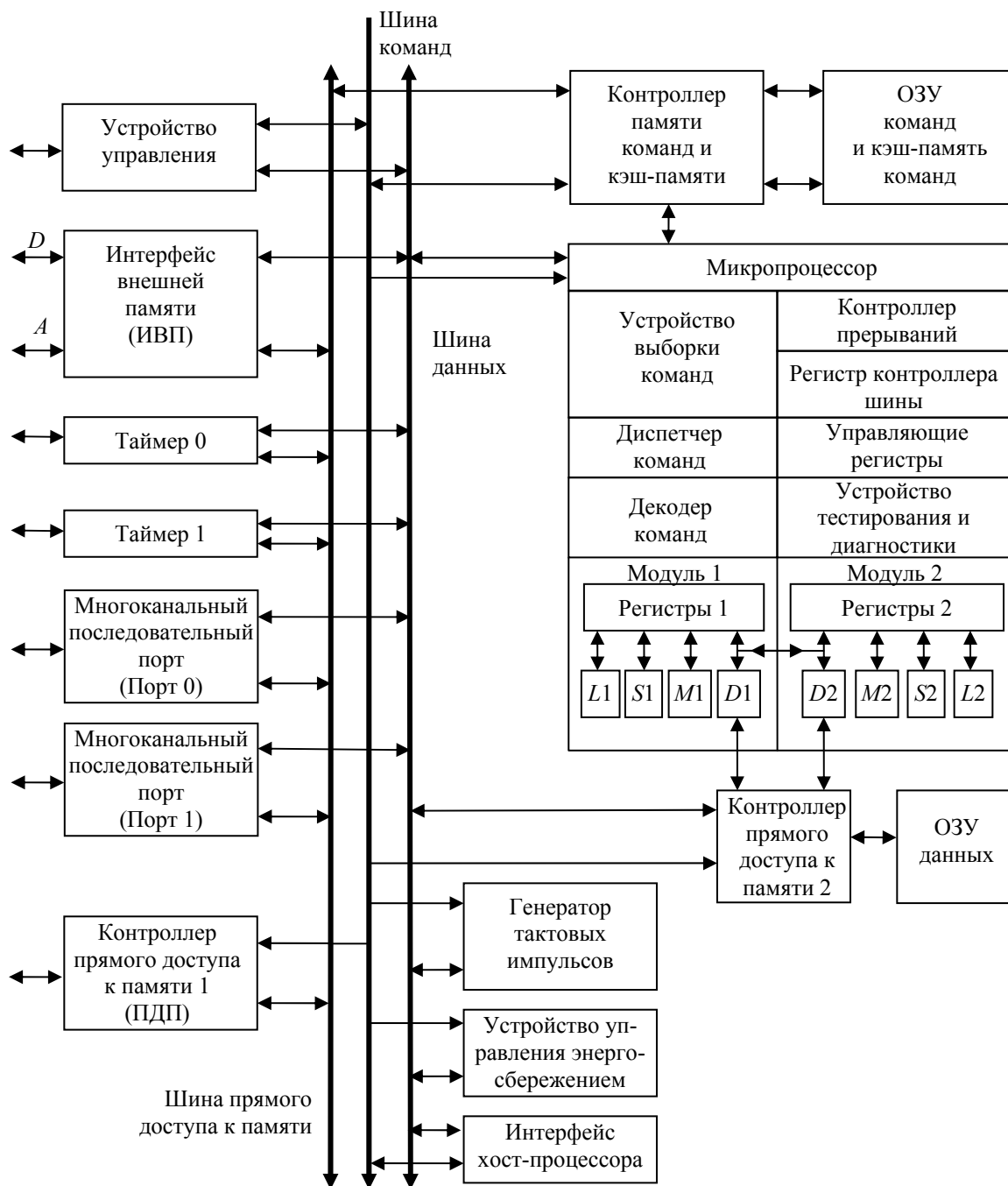


Рис. 3.25 Обобщенная структурная схема цифровых сигнальных процессоров серии TMS320C6xxx

На каждом такте работы цифрового сигнального процессора выбирается восемь 32-битных команд подобных командам микропроцессора с *RISC*- архитектурой. Предусмотренный в архитектуре цифровых сигнальных процессоров серии TMS320C6xxx формат команд позволяет выполнять эти восемь команд параллельно, последовательно или в смешанном режиме параллельно/последовательно, что существенно снижает размер кода и количество выборок команд. Возможность вычислений с плавающей точкой в двух моду-

лях, позволяет осуществлять преобразование цифрового сигнального процессора с фиксированной точкой в процессор с плавающей точкой. При этом новая система команд будет являться расширением системы команд процессора с фиксированной точкой и все коды, написанные для TMS320C62xx, будут выполняться на TMS320C67xx без модификаций самого кода.

Цифровые сигнальные процессоры серии TMS320C6xxx поддерживают широкий набор режимов косвенной адресации, включая линейный или кольцевой режимы адресации с 5- или 15-битным смещением. Все команды могут быть условными и большинство команд используют любой из 32 регистров. Некоторые регистры могут быть выделены для поддержки специфических режимов адресации или для хранения условий для условных команд. Процесс обработки команды начинается после выборки 256-битовой команды из внутренней памяти команд, которая также может быть сконфигурирована как кэш-память команд. Каждая из 32-битных команд распределяется на свой исполнительный блок. При этом у команд, выполняемых на разных блоках, осуществляется проверка младшего бита. Он устанавливается в единицу для всех команд, которые должны выполняться одновременно. Команды, которые собраны для одновременного выполнения (до 8 команд), образуют пакет выполнения. Ноль ставится в младшем бите команды, которая нарушает последовательность выполнения и откладывает команду на следующий пакет выполнения. В устройстве выборки команд может быть до 8 пакетов выполнения, образующих пакет выборки. Очередной пакет размещается для выполнения в исполнительных блоках в каждом такте. До окончания выполнения пакета выборки следующий пакет выборки из памяти не выбирается. Эта стратегия позволяет существенно экономить память команд и менять режим работы программы от одновременного параллельного выполнения 8 команд на двух функциональных модулях до последовательного выполнения команд в зависимости от требований алгоритма программы. Данные из исполнительных блоков помещаются в регистры 1 и 2, а затем по адресам, формируемым арифметико-логическими устройствами  $D1$  и  $D2$  идет их обмен с ОЗУ данных. При этом каждый из блоков буферных регистров 1 и 2 соединен 32-разрядными шинами с диспетчером команд. Диспетчер команд организует одновременную выборку данных из ОЗУ данных по 4 шинам. При этом по двум подаваемым адресам происходит передача 64 разрядов данных. Оперативное запоминающее устройство данных делится на несколько областей, что исключает конфликты при доступе к памяти. Такое решение обеспечивает доступ к памяти без задержек при параллельных потоках обращений и при

возможности отдельной адресации каждого байта памяти. Вся память цифровых сигнальных процессоров серии TMS320C6xxx организована как многопортовая и количество одновременно выбираемых данных может меняться.

Рассмотрим подробнее периферийные устройства.

Контроллер прямого доступа к памяти 1 осуществляет передачу данных без участия микропроцессора (рис.3.25). Он имеет четыре основных программируемых и пять дополнительных каналов. Кроме того, контроллер прямого доступа к памяти 1 используется при начальной загрузке программы в память процессора.

Интерфейс хост-процессора служит для синхронного и асинхронного обмена данными с управляющим контроллером. Он представляет собой 16-разрядный параллельный порт, который обеспечивает хост-процессору прямой доступ к памяти цифрового сигнального процессора. При этом хост-процессор является управляющим устройством для данного интерфейса, что существенно упрощает процедуру доступа. Хост-процессор и цифровой сигнальный процессор могут обмениваться информацией, как через внутреннюю, так и через внешнюю память. Кроме того, хост-процессор может иметь прямой доступ к большинству периферийных устройств, размещенных на кристалле.

Интерфейс внешней памяти предназначен для обмена данными с внешней памятью и быстродействующими внешними устройствами. Интерфейс внешней памяти принимает запросы на обмен с внешним запоминающим устройством от контроллера памяти данных, контроллера памяти команд и кэш-памяти и контроллера прямого доступа к памяти 2. Интерфейс внешней памяти формирует сигналы для непосредственного подключения быстродействующего динамического (*SDRAM*) или статического (*SBSRAM*) синхронного внешнего ОЗУ. Кроме того, к интерфейсу внешней памяти можно подключить статическое ОЗУ, ПЗУ и другие устройства.

Цифровые сигнальные процессоры TMS320C62xx и TMS320C67xx имеют несколько режимов начальной загрузки, которые определяют поведение процессора после сброса при подготовке к инициализации. В режимы загрузки включаются программы загрузки с внешнего ПЗУ через интерфейс внешней памяти или загрузку программы через интерфейс хост-процессора из внешнего устройства.

Многоканальные буферизованные последовательные порты (Порт 0 и Порт 1) представляют собой последовательные высокоскоростные порты, базой которых являются стандартные последовательные порты, применяемые в

цифровых сигнальных процессорах. С их помощью имеется возможность читать или записывать данные в память без участия микропроцессора (рис.3.25) через контроллер прямого доступа к памяти 1.

Цифровые сигнальные процессоры серии TMS320C6xxx имеют два 32-разрядных таймера, которые могут быть использованы для задания временных интервалов, реализации счетчиков, в качестве генераторов импульсов, устройств прерывания микропроцессора (рис.3.25), передачи импульсов синхронизации в контроллер прямого доступа к памяти.

Периферийные устройства могут иметь до 32-х источников прерываний. Контроллер прерываний, входящий в структуру микропроцессора (рис. 3.25), дает возможность выбора 12 прерываний, которые будут использоваться, а также обеспечивает возможность смены полярности внешних импульсов прерываний.

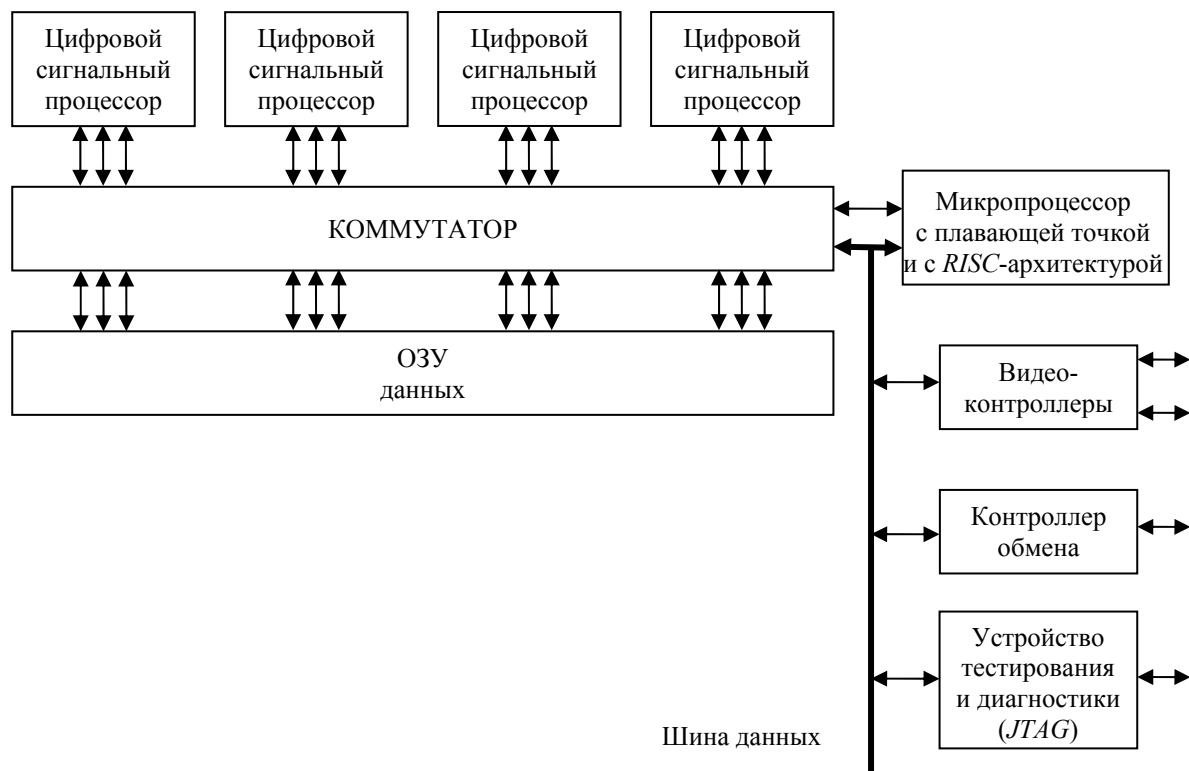


Рис. 9.26 Обобщенная структурная схема цифрового сигнального процессора TMS320C80

Для снижения энергопотребления цифрового сигнального процессора устройство управления энергосбережением имеет возможность прерывать работу генератора тактовых импульсов. При включении «спящих» режимов, у цифровых сигнальных процессоров прерывается работа генератора тактовых импульсов и отключается работа микропроцессора, затем периферийных уст-

роЙств, размещенных на кристалле, а далее прекращается работа всех устройств, в том числе и блока умножения частоты.

Дальнейшее совершенствование технологии производства БИС позволило разместить на одном кристалле несколько цифровых сигнальных процессора и управляющий микропроцессор, который организывает их работу. Такая архитектура применена в цифровом сигнальном процессоре TMS320C80, обобщенная структурная схема которого приведена на рис.3.26. В его состав входят четыре 32-разрядных цифровых сигнальных процессора и 32-разрядный микропроцессор с сопроцессором вычислений с плавающей точкой на основе *RISC*- архитектуры, коммутатор, осуществляющий связь цифровых сигнальных процессоров и микропроцессора с ПЗУ команд и данных объемом до 50 Кбайт. Кроме того, коммутатор обеспечивает связь двух видеоконтроллеров, контроллера обмена, который осуществляет обмен данными с внешней памятью в режиме прямого доступа к памяти, а также связь с устройством тестирования и диагностики (*JTAG*).

При частоте работы генератора тактовых импульсов 40...50 МГц цифровой сигнальный процессор обеспечивает производительность свыше 2 млрд. операций/сек. Большая интегральная схема содержит около 4 млн. транзисторов и имеет напряжение питания 3.3 В.

Архитектура TMS320C80 организована по принципу *MIMD* (*Multiply Instructions Multiply Data*), заключающемся в том, что командное слово содержит команды для нескольких цифровых сигнальных процессоров и данные для выполнения команд. Реализация указанного принципа оказалась возможной благодаря наличию нескольких параллельно работающих цифровых сигнальных процессоров (см. рис.3.26).

Микропроцессор с сопроцессором вычислений с плавающей точкой на основе *RISC*- архитектуры содержит целочисленное арифметико-логическое устройство, перемножитель с плавающей точкой, сумматор с плавающей точкой, устройство сдвига, 31 тридцатидвухразрядный регистр, четыре регистра-аккумулятора с плавающей точкой, таймер и управляющий регистр. Кроме того имеется контроллер памяти команд и кэш-памяти команд и данных, объемом по 4 Кбайт каждая, а также интерфейс с коммутатором цифровых сигнальных процессоров, входящих в структуру TMS320C80. Система команд микропроцессора с сопроцессором вычислений с плавающей точкой на основе *RISC*- архитектуры ориентирована на программирование на языке *C*.

Каждый цифровой сигнальный процессор, входящий в структуру TMS320C80, ориентирован на обработку видеоизображений и графических изображений. Он работает с 32-разрядными данными и 64-разрядными командами и имеет в своем составе 44 буферных регистра, 32-разрядное трехвходовое арифметико-логическое устройство, два генератора адресов, перемножитель и 32-разрядное устройство барабанного сдвига, блок условных операций для сокращения времени выполнения переходов, блок для работы с битами, блок и шины локальных и глобальных адресов и данных.

Цифровой сигнальный процессор TMS320C80 применяется в качестве устройств обработки сигналов в мультимедийных системах, системах видеоконференцсвязи, устройствах обработки изображений, высокоскоростных телекоммуникационных системах, устройствах сжатия видео и аудио данных.

В табл. 3.7 приведены основные технические характеристики некоторых цифровых сигнальных процессоров. Как видно из таблицы, увеличение производительности цифровых сигнальных процессоров происходит как за счет увеличения частоты работы генератора тактовых импульсов, так и благодаря совершенствованию технологии и организации параллельной работы большего числа исполнительных блоков.

Развитие архитектуры процессоров связано с увеличением объема ОЗУ и ПЗУ, расширением состава и числа периферийных устройств, появлением средств организации многопроцессорных устройств (интерфейс хост-процессора).

Таблица 3.7

Цифровые сигнальные процессоры	Частота МГц	Производительность млн. команд/сек/ (млн. операций/сек в формате с плавающей точкой)	Объем ОЗУ	Объем ПЗУ	Таймеры	Последовательные порты
TMS320C10	20	8.77	2.2 Кбит	24 Кбит		
TMS320C17	20	8.77	4 Кбит	64 Кбит		
TMS320C25	40	10	8.7 Кбит	64 Кбит	1	1
TMS320C30	40	20	64 Кбит	128 Кбит	1	2
TMS320F206	40	20	516Кбит	72 Кбит	1	2
TMS320C206	80	40	64 Кбит	72 Кбит	1	2
TMS320VC33-120	60	60	544Кбит	72 Кбит	2	1
TMS320C40	60	30/(60)	1.1Мбит	64 Кбит	2	6
TMS320C50	66	33	160Кбит	32 Кбит	1	2
TMS320VC549-80	80	80	160Кбит	768 Кбит	1	3
TMS320C6201-EP	200	1600	1Мбит		2	2

TMS320C6701-167	167	1600/(1200)	1Мбит		2	2
TMS320C80	40...50	/(2000)	400Кбит		5	

В заключение отметим, что основным направлением развития цифровых сигнальных процессоров рассмотренных выше серий является повышение степени интеграции и размещение на одном кристалле большого числа процессоров и управляющего микропроцессора, построенного на основе *RISC* – архитектуры.

### 3.2.2 Цифровые сигнальные процессоры ADSP21xx, ADSP210xx, ADSP-TS2035

В состав серии ADSP21xx входит четырнадцать однокристалльных 16-разрядных цифровых сигнальных процессоров, выполняющих вычисления с фиксированной точкой. Их архитектура ориентирована на реализацию алгоритмов цифровой обработки сигналов. Для цифровых сигнальных процессоров этой серии (обобщенная структурная схема приведена на рис.3.27) характерным является наличие 4-х внутренних шин:

- шина адреса памяти команд *PMA* (*Program Memory Address*),
- шина адреса памяти данных *DMA* (*Data Memory Address*),
- шина данных памяти команд *PMD* (*Program Memory Data*),
- шина данных памяти данных *DMD* (*Data Memory Data*).

Внутренние шины с помощью мультиплексоров подключаются к внешним шинам адресов и данных. Устройство обмена данными между шинами *DMD* и *PMD*, имеющими различную разрядность, позволяет согласовать формат передачи данных из одной шины в другую.

Два генератора адресов данных (*DAG1*, *DAG2*) и программный автомат, в состав которого входят: регистр команд, счетчик команд, генератор адресов команд и кэш-память, обеспечивают работу исполнительных блоков цифровых сигнальных процессоров: арифметико-логического устройства, перемножителя-накопителя и устройства барабанного сдвига. Два генератора адресов данных позволяют одновременно формировать адреса двух операндов. Исполнительные блоки обрабатывают 16-разрядные данные и поддерживают вычисления с повышенной точностью. Каждый исполнительный блок имеет входные и выходные регистры и соединяется с внутренней шиной результата (*R*). По этой шине передаются результаты вычислений с выхода одного исполнительного блока на вход другого.

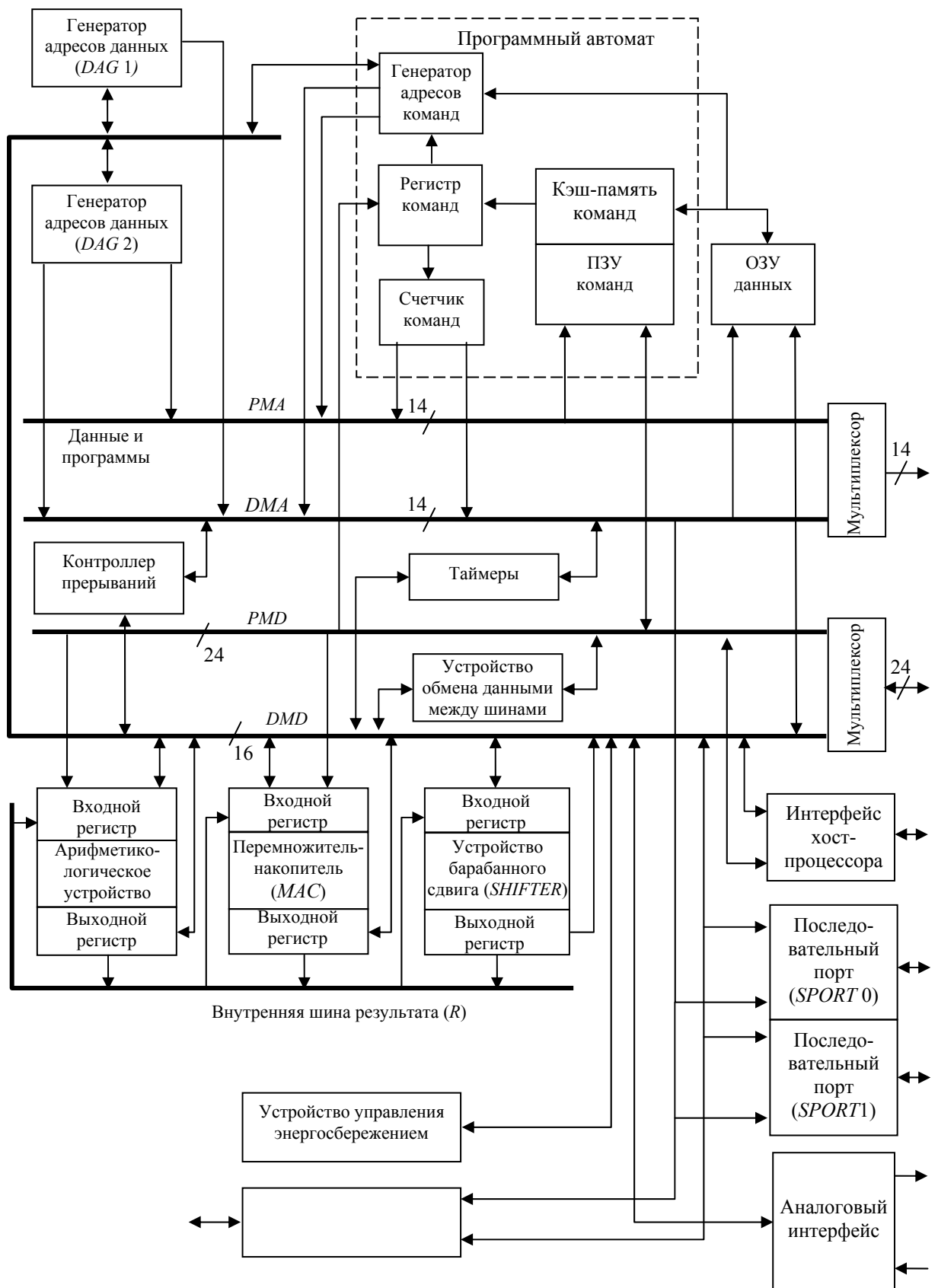


Рис. 3.27 Обобщенная структурная схема цифровых сигнальных процессоров серии ADSP21xx



Цифровые сигнальные процессоры серии ADSP21xx могут содержать следующие периферийные устройства:

- интерфейс хост-процессора,
- один или два последовательных порта,
- один или два порта прямого доступа к памяти,
- аналоговый интерфейс,
- таймеры,
- контроллер прерываний.

Интерфейс с хост-процессором (*HIP*) состоит из регистров, через которые цифровой сигнальный процессор и хост-процессор обмениваются данными и информацией о их состояниях.

В состав аналогового интерфейса цифровых сигнальных процессоров серии ADSP21xx входят входные усилители аналоговых сигналов, 16-разрядный аналого-цифровой преобразователь с регистром на выходе, а также цифроаналоговый преобразователь и встроенные фильтры, предназначенных для снижения уровня помех, вызванных наложением спектров сигналов. Аналого-цифровые и цифроаналоговые преобразователи построены с использованием принципов сигма-дельта модуляции.

Цифровые сигнальные процессоры этой серии содержат один или два двунаправленных последовательных порта *SPORT0*, *SPORT1*. Они имеют аппаратные средства сжатия и восстановления данных (устройства упаковки-распаковки) и приемные и передающие регистры (рис.3.28).

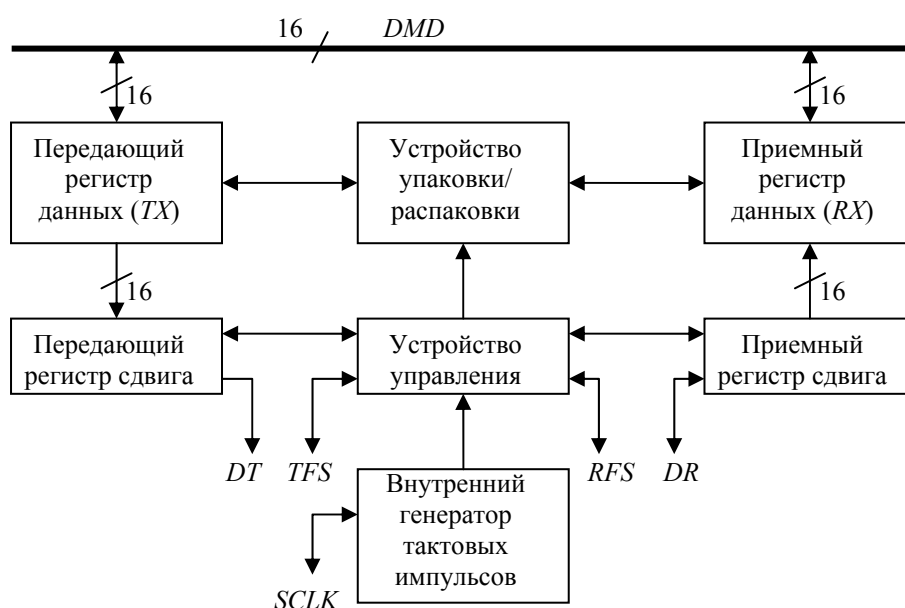


Рис. 3.28 Последовательный порт (*SPORT*)

Работа последовательных портов осуществляется следующим образом. После записи данных в передающий регистр  $TX$  последовательного порта, сигналом  $TFS$  инициализируется их передача. После начала передачи данных каждое число, записанное в передающий регистр  $TX$ , посылается в передающий регистр сдвига. Это число по битам считывается на выход  $DT$  передающего регистра сдвига.

При работе последовательного порта на прием данных, поступающие на вход  $DR$  приемного регистра сдвига биты, записываются в регистр. После окончания приема, записанное слово переписывается в приемный регистр  $RX$  и поступает в шину данных памяти данных. В последовательном порту генерируется сигнал прерывания приема. Последовательный порт может формировать с помощью внутреннего генератора тактовые импульсы или использовать тактовые импульсы от внешнего генератора.

Программируемый интервальный таймер (рис.3.29) цифровых сигнальных процессоров серии ADSP21xx обеспечивает периодическую генерацию сигналов прерывания. Для генерации этих сигналов используется регистр масштаба (шкалы) таймера  $TSCALE$ , регистр периода счета  $TPERIOD$ , регистр-счетчик  $TCOUNT$ , работающий в режиме вычитания.

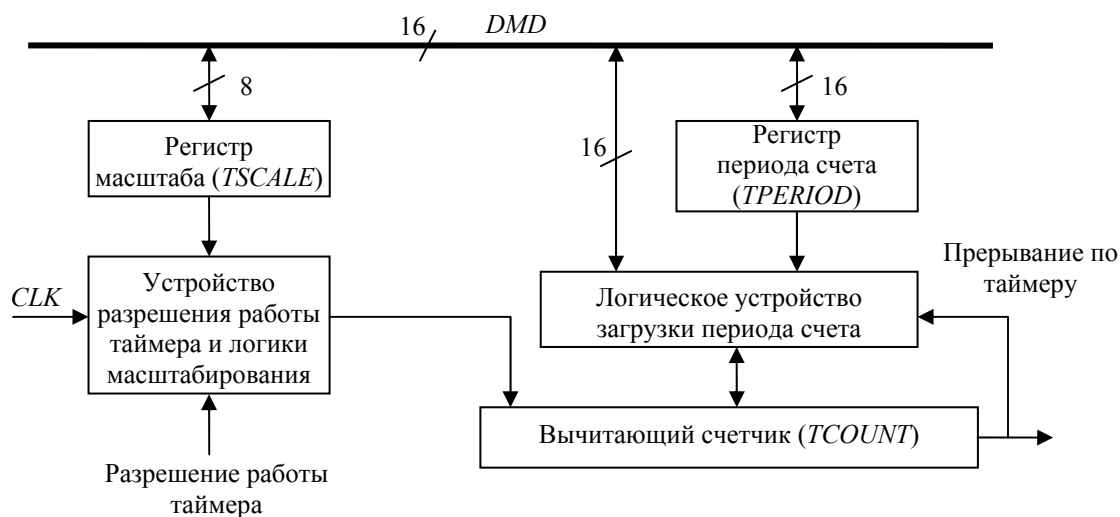


Рис. 3.29 Программируемый интервальный таймер

Программируемый интервальный таймер периодически генерирует сигналы прерывания через каждые  $(TPERIOD+1)$  периодов за  $(TSCALE+1)$  циклов, где  $TPERIOD$  – нормированное время периода счета,  $TSCALE$  - нормированное время масштаба таймера. Работа программируемого интервального таймера осуществляется следующим образом. После инициализации содержимое вычитающего счетчика  $TCOUNT$  декрементируется через каждые  $n$

циклов. При этом коэффициент масштабирования  $TSCALE$  равен  $n - 1$ . Когда код вычитающего счетчика станет равным нулю, генерируется сигнал прерывания по таймеру. После этого в вычитающий счетчик вновь загружается значение кода из регистра периода счета.

Порты прямого доступа к памяти, обеспечивают быстрый обмен с внутренней памятью, причем внутренний порт прямого доступа к памяти, ( $IDMA$ ) осуществляет асинхронный обмен с памятью команд, а байтовый порт прямого доступа к памяти ( $BDMA$ ) производит асинхронный обмен как с памятью команд, так и с памятью данных.

Контроллер прерываний обеспечивает переход на подпрограмму обслуживания прерываний по сигналам запросов прерываний от внешних источников ( $IRQ0, \dots, IRQ2$ ), таймера, последовательных портов, порта прямого доступа к памяти, интерфейса хост-процессора и устройства управления энергосбережением.

Структура контроллера прерываний содержит регистр управления прерываниями и регистр маскирования прерываний.

Цифровые сигнальные процессоры серии ADSP210xx (на рис.3.30 дана структурная схема одного из представителей этой серии ADSP21020) представляют собой 32-разрядные процессоры, реализующие вычисления с плавающей точкой.

В этих процессорах увеличена разрядность шин, реализована аппаратная поддержка обработки данных в формате с плавающей точкой и расширен набор периферийных устройств. Соответственно в структуре цифрового сигнального процессора (рис.3.30) появились: арифметико-логическое устройство вычислений с плавающей точкой, перемножитель чисел в формате с плавающей точкой, 48-, 40- и 32-разрядные шины и регистры общего назначения для обмена данными между исполнительными блоками и внутренней шиной данных, а также для запоминания промежуточных результатов.

Увеличение разрядов командного слова до 48 позволяет задавать в нем параллельное выполнение арифметических операций и обмен данными. За один такт работы возможно одновременно считать или записать два операнда в регистры общего назначения, загрузить два операнда в арифметико-логическое устройство, принять два операнда в перемножитель. Арифметико-логическое устройство и перемножитель могут получить два результата одновременно.

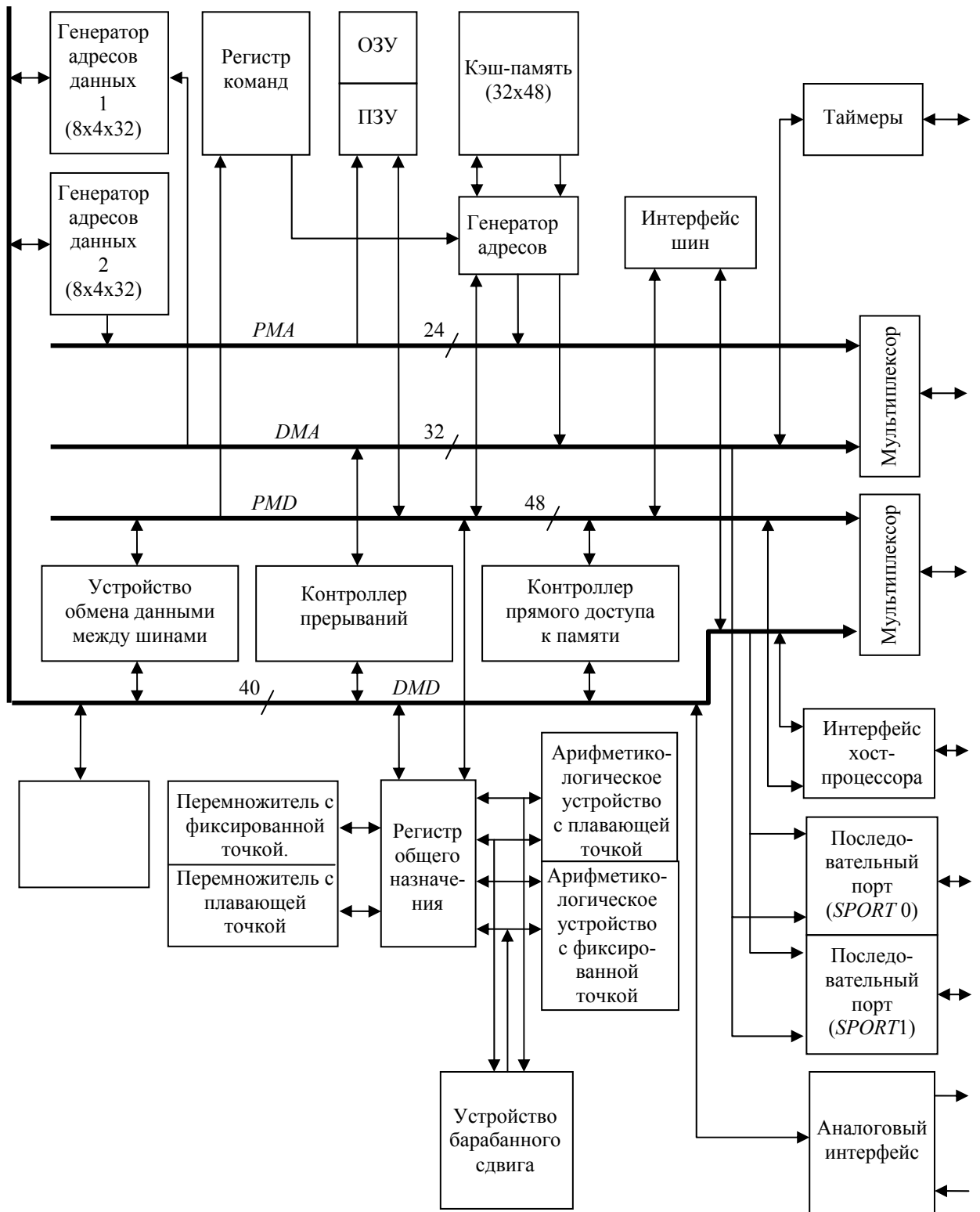


Рис. 3.30 Структура цифрового сигнального процессора ADSP21020

Производительность цифрового сигнального процессора ADSP-21020 составляет 66 млн. операций с плавающей точкой в секунду при тактовой частоте работы 33 МГц.

Цифровые сигнальные процессоры ADSP21060 и ADSP21062 имеют структуру (рис.3.31), ориентированную на расширение коммуникационных возможностей при построении многопроцессорных систем. Такая структура получила название *SHARC* (*Super Harvard Architecture Computer*). В нее входят:

- процессор, выполняющий обработку данных в формате с плавающей точкой;
- сопроцессор ввода-вывода и многоканальный контроллер прямого доступа к памяти;
- двухвходовое запоминающее устройство, объемом 4 Мбита;
- мультипроцессорный интерфейс, объединяющий интерфейс с хост-процессором и порты для связи с внешними устройствами; шинный коммутатор.

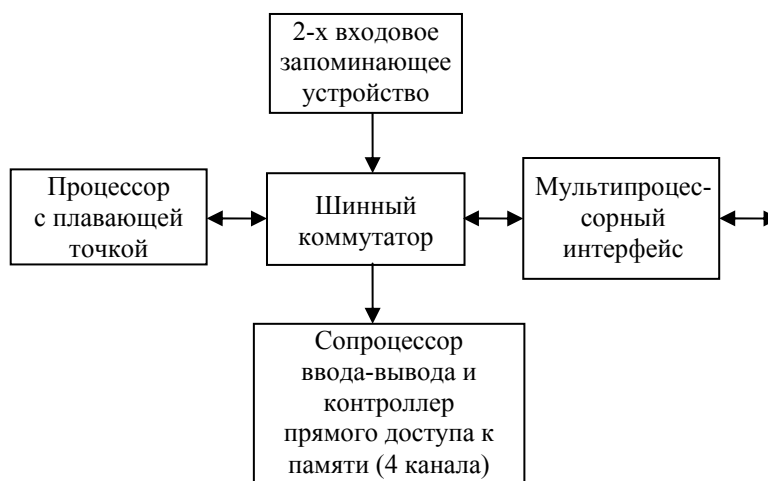


Рис. 3.31 Структурная схема цифровых сигнальных процессоров ADSP2106x

В цифровом сигнальном процессоре ADSP2116xx [12] введено два процессора с плавающей точкой, способных одновременно выполнять команду, каждый над своими данными (режим *SIMD*). В цифровом сигнальном процессоре ADSP2116xx увеличена разрядность шин, повышена до 100 МГц тактовая частота работы, снижено до 2,5 В напряжение питания.

Цифровой сигнальный процессор ADSP-TS203S имеют архитектуру, оптимизированную для решения сложных задач цифровой обработки сигналов в реальном масштабе времени и задач, связанных с обработкой высокоскоростных цифровых потоков данных.

В этом процессоре тактовая частота увеличена до 500 МГц, и время выполнения команды составляет 2.0 нсек. Объем внутреннего оперативного запоминающего устройства увеличен до 4Мбит. Цифровой сигнальный процессор (рис.3.32) содержит два параллельно работающих исполнительных

блока  $X$  и  $Y$ . Каждый из них содержит арифметико-логическое устройство, перемножитель, 64-битовое устройство барабанного сдвига и регистры общего назначения. Исполнительные блоки поддерживают операции с числами с плавающей точкой, имеющими разрядность 32 и 40 бит, а также с числами с фиксированной точкой, разрядностью 8, 16, 32, и 64 бит.

К особенностям архитектуры следует отнести память со 128-битовым кодом команд, что позволяет цифровому сигнальному процессору выполнять до четырех команд в течение каждого цикла. При этом реализуются двадцать четыре операции с 16-битовыми числами с фиксированной точкой или шесть операций с плавающей запятой. В длинном слове команды имеется информация обо всех операциях, которые должны выполняться одновременно.

Четыре независимых внутренних шины данных разрядностью 128 бит соединены с четырьмя блоками памяти, причем емкость каждого банка памяти составляет 1 Мбит. Цифровой сигнальный процессор может выполнять четыре миллиарда 40-битных операций перемножения с накоплением в секунду.

Цифровой сигнальный процессор ADSP-TS203S имеет следующие периферийные устройства:

- 10 канальный контроллер прямого доступа к памяти;
- интерфейс хост-процессора;
- два интерфейса канала связи;
- контроллер памяти;
- два таймера;
- устройство тестирования и диагностики (*JTAG*);
- устройство арбитража для организации мультипроцессорных систем;
- контроллер прерываний.

В состав цифрового сигнального процессора входит устройство формирования последовательности команд, содержащее устройство предвыборки, буфер предсказания (прогнозирования) команд ветвления (*BTB*), счетчик команд и блок очереди команд *IAB* (*Instruction Alignment Buffer*).

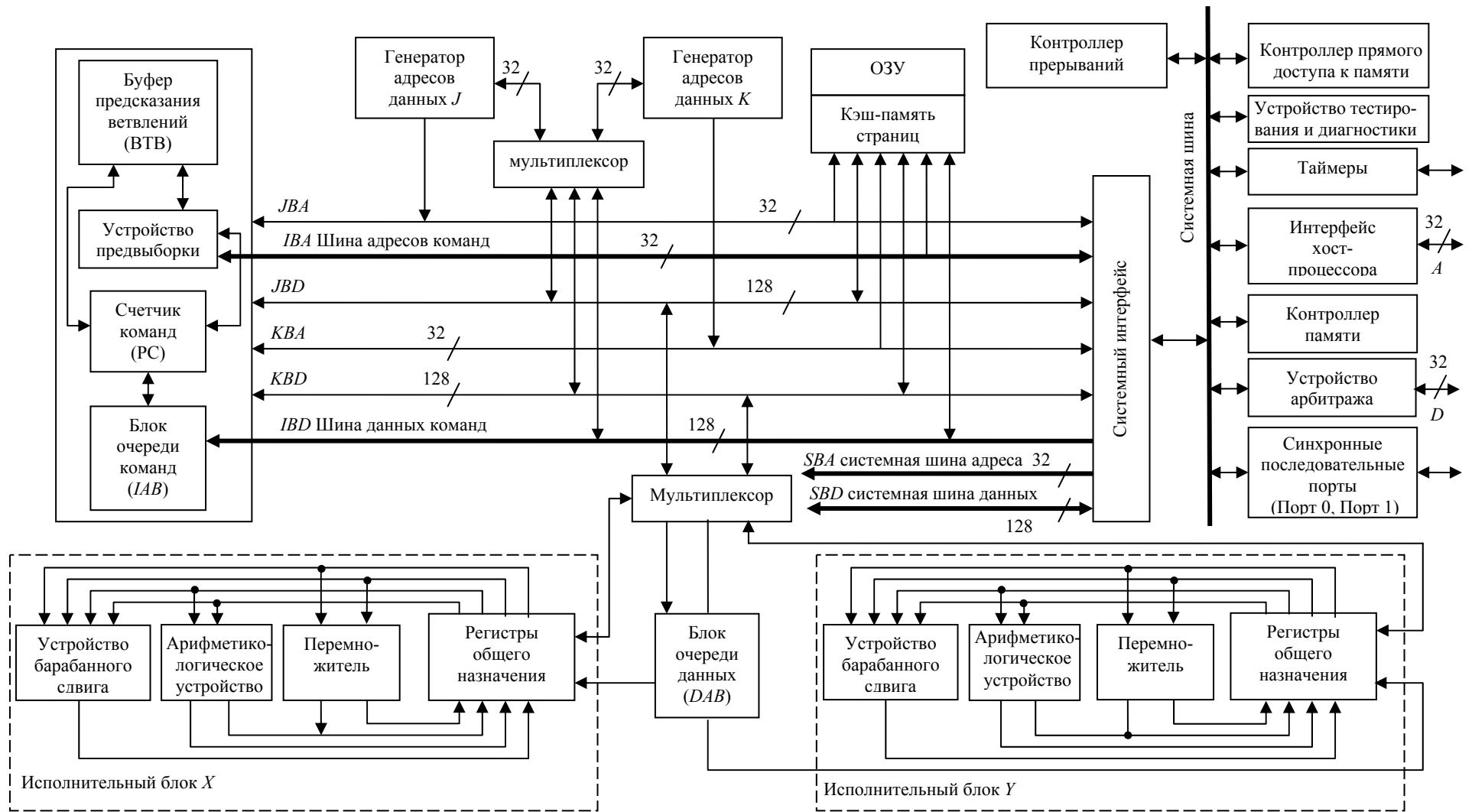


Рис. 3.32 Структурная схема цифрового сигнального процессора ADSP – TS203S

Внутренняя и внешняя память организована в объединенную карту памяти. Внутренняя память, объемом 4Мбита, разделена на четыре блока, каждый из которых содержит 32 тысячи 32-битных слова. Каждый блок (*M0*, *M2*, *M4* и *M6*) может содержать данные, команды или то и другое. Внутренняя архитектура шин обеспечивает пропускную способность 28 ГГбайт в секунду. Гибкая структура памяти цифрового сигнального процессора позволяет исполнительным блокам и устройствам ввода - вывода обращаться к различным блокам памяти в одном цикле. Процессор может обращаться к трем блокам памяти параллельно для выборки одной команда и двух операндов с помощью набора адресных шин *JBA*, *KBA*, *IBA* и шин данных *JBD*, *KBD* и *IBD* (см. рис.9.32).

Таблица 3.8

Цифровые сигнальные процессоры	Частота МГц	Производительность млн. операций в сек/ (млн.операций/сек в формате с плавающей точкой)	Объем ОЗУ (бит)	Объем ПЗУ (бит)	Контроллеры прямого доступа к памяти (число каналов)	Таймеры	Последовательные порты
ADSP2101	20	40	48К	128К		1	2
ADSP2181	20	40	512К	128К	1	1	2
ADSP2188N	80	80	896К	48К	1	1	2
ADSP-BF535	350	700	2,5М	16К	1	4	6
ADSP-21365	300	/(1800)	3М	4М	1(25)	3	6
ADSP-TS101S	300	/(1800)	6М		1	2	4
ADSP-TS203S	500	/(4000)	4М		1(10)	2	4

Цифровой сигнальный процессор имеет программируемое разделение памяти команд и памяти данных. Он обеспечивает простой и конфигурируемый интерфейс между его внешними шинами и ведущим процессором в многопроцессорной системе через внешний порт. Чтобы приспособить интерфейс к различным ведущим процессорам, интерфейс поддерживает быстрые и медленные протоколы для доступа хост-процессора. Каждый



протокол имеет программируемые параметры передачи, типы неактивных циклов и внутренних циклов ожидания.

Контроллер прямого доступа к памяти содержит 10 каналов и обеспечивает процесс передачи данных без вмешательства процессора, выполняя внешние поблочные пересылки между внутренней памятью и любой внешней памятью и обеспечивая связь по четырем каналам (два на передачу и два на прием) между портом связи и внутренней или внешней памятью.

В табл. 3.8 приведены основные технические характеристики некоторых цифровых сигнальных процессоров рассмотренных выше типов. Как следует из анализа этой таблицы, основным направлением развития подобных структур цифровых сигнальных процессоров является повышение производительности, увеличение быстродействия и объемов памяти. Кроме того, возрастают возможности работы с прямым доступом к памяти, что является характерным при построении различных радиотехнических устройств, требующих частого обращения к памяти.

### **3.2.3 Цифровые сигнальные процессоры DSP56xxx, DSP9600x, MSC8101, MSC8102**

Эти цифровые сигнальные процессоры [12, 13, 20, 21] представляют собой 16- или 24-разрядные микропроцессоры с фиксированной точкой, либо 32-разрядные микропроцессоры с обработкой данных в формате с плавающей точкой. Основные принципы архитектуры реализованы в цифровых сигнальных процессорах серии DSP56xxx. Их обобщенная структура представлена на рис.3.33 и содержит исполнительный блок и совокупность периферийных устройств и запоминающих устройств команд и данных.

В состав исполнительного блока входят:

- арифметико-логическое устройство, перемножитель-накопитель (*MAC*) и два 56-разрядных регистра-аккумулятора;
- устройство генерации адресов данных (*AGU*), которое обеспечивает вычисление требуемых адресов данных в памяти за один такт с помощью двух арифметических устройств;
- устройство управления программой (*PCU*), осуществляющее предварительную выборку команд, декодирование их, обработку команд циклических переходов и прерываний;
- двунаправленные 24-разрядные шины данных *XDB*, *YDB* памяти данных и шины памяти команд *PDB*.



правленной шине, являющейся выходом мультиплексора шин *XAB*, *YAB* и *PAB*.

В структуру цифровых сигнальных процессоров серии DSP56xxx входят интерфейсы хост-процессора, порты параллельного и последовательного ввода-вывода (*SCI* и *SSI*) и генератор тактовых импульсов с устройством фазовой автоподстройки частоты. Устройство тестирования и диагностики позволяет анализировать состояния регистров, запоминая устройств, и управлять процессом отладки программы по интерфейсу *JTAG*.

Устройство обработки битов расположено в блоке мультиплексора внутренней шины данных, что обеспечивает этому устройству доступ к любой области памяти и позволяет выполнять битовые операции над данными, записанными в памяти и регистрах, а также над содержимым адресных и управляющих регистров.

Генератор адресов команд обеспечивает вычисление требуемых адресов данных в памяти за один такт с помощью двух 16-разрядных арифметических устройств.

Конвейер выполнения команды осуществляется за 3 этапа: предвыборки, декодирования, выполнения. Работы конвейера организовывается устройством управления программой. Генератор тактовых импульсов имеет в своем составе устройства фазовой автоподстройки частоты и умножения частоты, что позволяет цифровому сигнальному процессору работать на повышенной внутренней тактовой частоте и обеспечивать синхронизацию внутренних и внешних тактовых импульсов. Кроме того, в энергосберегающем режиме осуществляется понижение тактовой частоты.

Для повышения скорости обработки данных, повышения точности вычислений и расширения функциональных возможностей, в цифровых сигнальных процессорах серии DSP9600x увеличиваются тактовые частоты работы до 100...300МГц, возрастает разрядность шин, регистров и обрабатываемых данных. Дополнительно, вводятся устройства обработки данных в формате с плавающей точкой и устройства сдвига, увеличиваются объемы ПЗУ и ОЗУ, используется кэш-память и контроллеры прямого доступа к памяти. В цифровых сигнальных процессорах серии DSP9600x пониженное напряжение питания и имеется устройство управления энергосбережением.

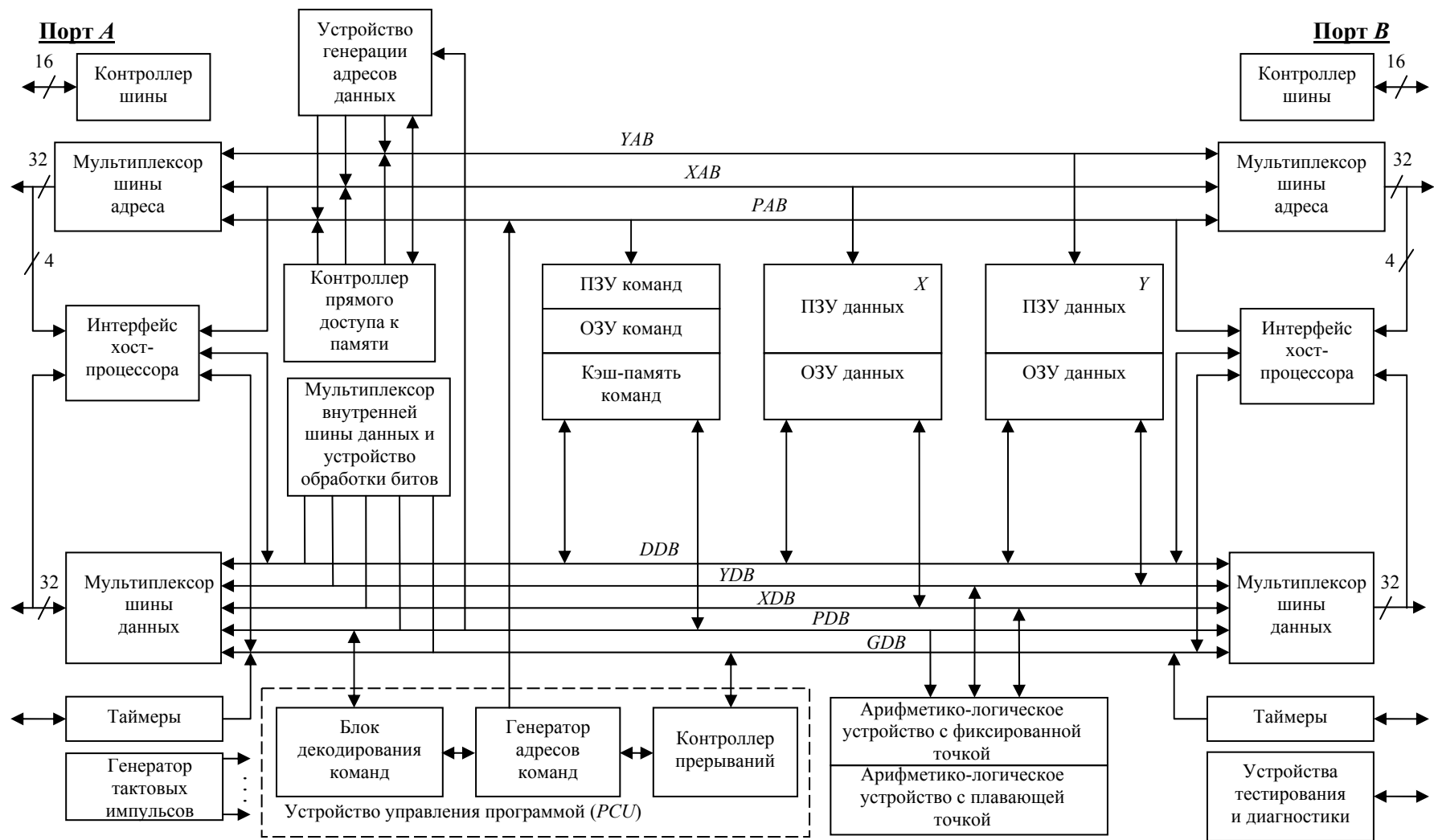


Рис. 3.34 Структурная схема цифровых сигнальных процессоров серии DSP9600x

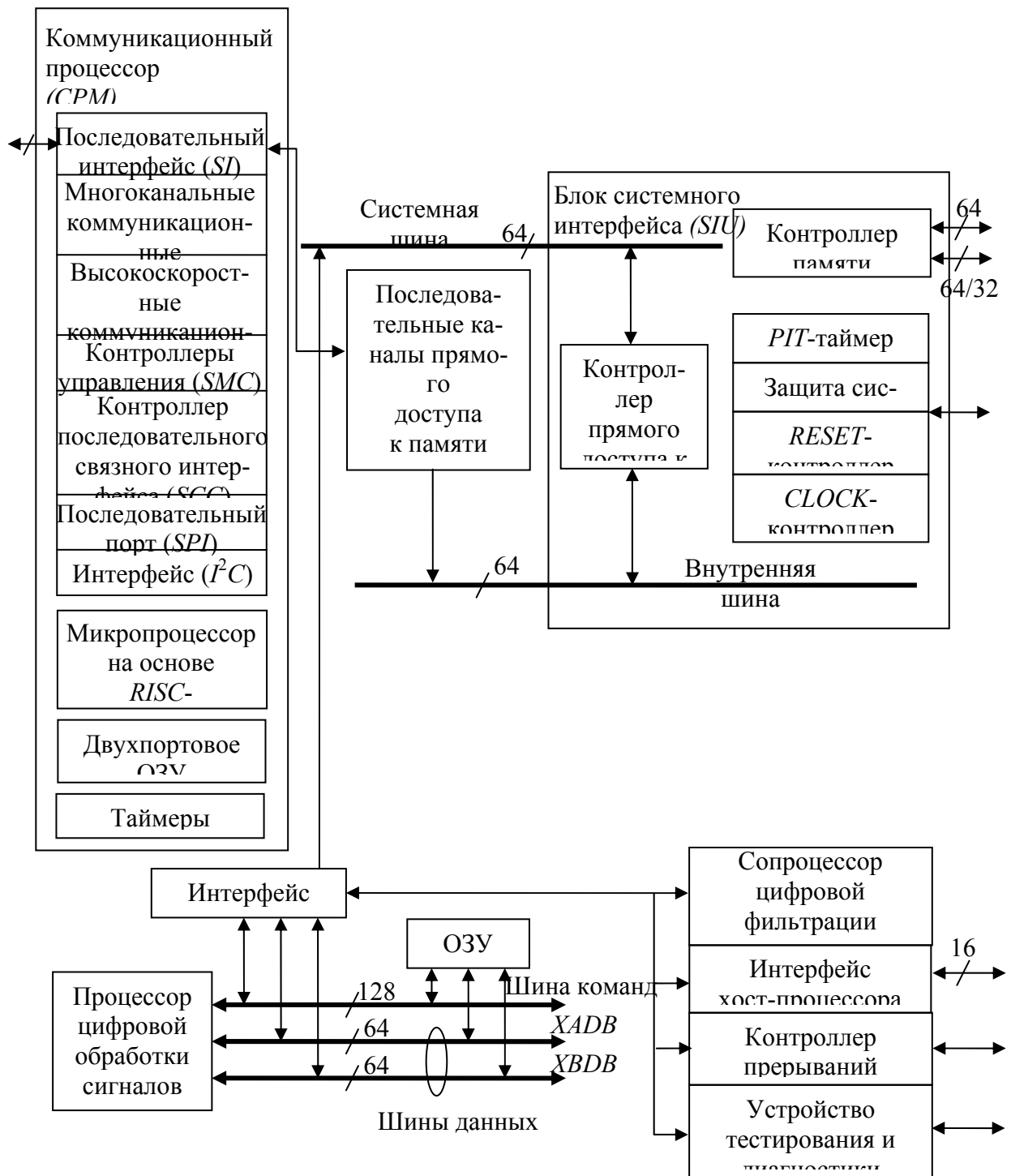


Рис. 3.35 Структура цифрового сигнального процессора MSC8101

Так, 32-разрядный цифровой сигнальный процессор серии DSP9600x (рис. 3.34) с устройством обработки данных в формате с плавающей точкой содержит 1Кбайт памяти команд, 1Кбайт памяти данных (X и Y блоки), кэш-память команд, два ППЗУ данных, двухканальный контроллер прямого дос-

тупа к памяти, встроенные устройства тестирования и диагностики, таймеры, интерфейс хост-процессора и мультиплексоры шин адреса и данных [18].

В состав цифровых сигнальных процессоров серии DSP9600x входят 3 параллельно функционирующих устройства: арифметико-логическое устройство, устройство генерации адреса данных и устройство управления программой.

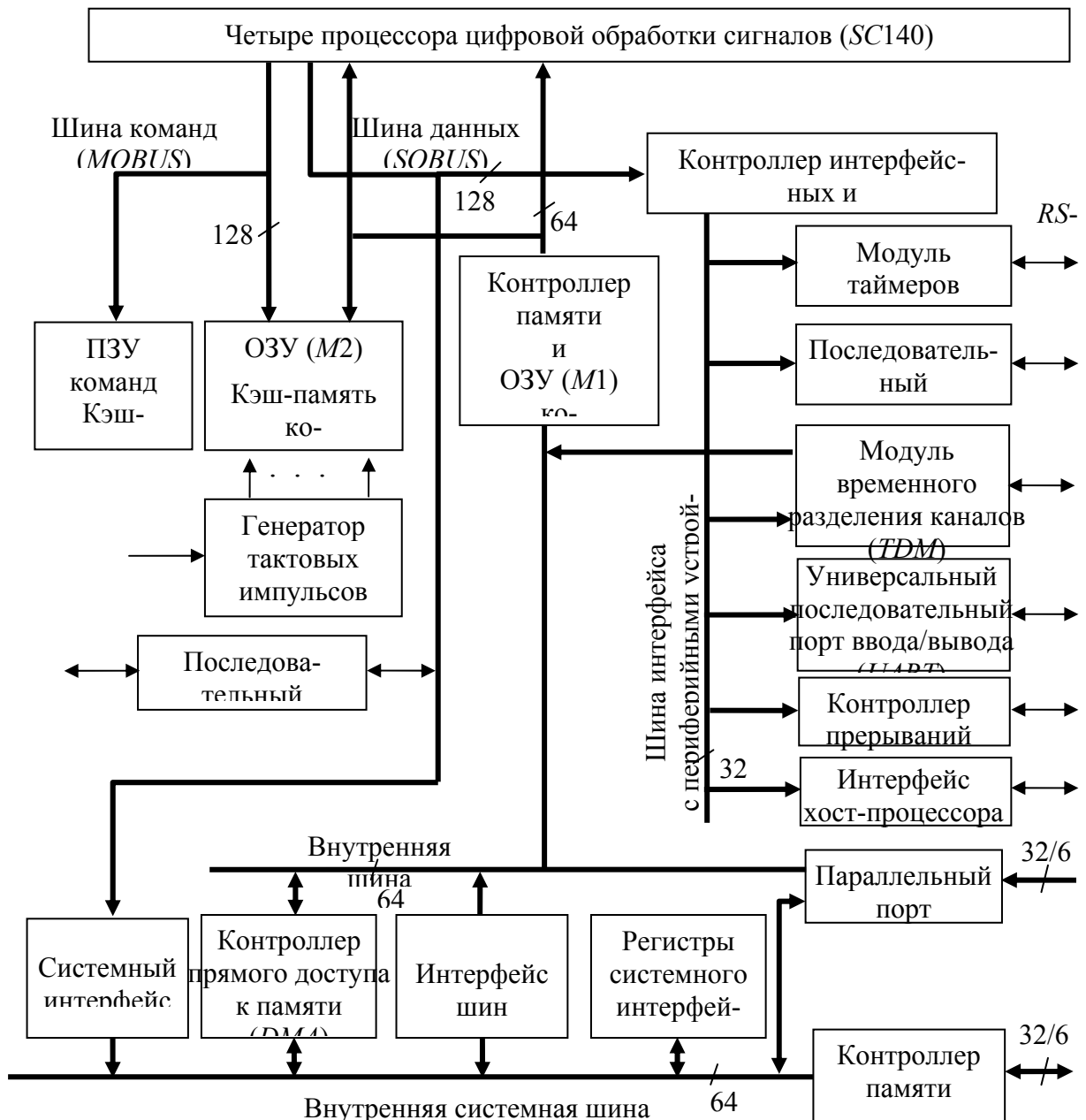


Рис. 3.36 Структура цифрового сигнального процессора MSC8102

Эти процессоры имеют пять 32-разрядных шин адреса ( $XAB$ ,  $YAB$ ,  $PAB$  и две внешние) и семь 32-разрядных шин данных ( $DDB$ ,  $YDB$ ,  $XDB$ ,  $PDB$ ,  $GDB$  и две внешние).

Производительность цифровых сигнальных процессоров этой серии на тактовой частоте 40 МГц составляет около 200 млн. операций в секунду. Для расширения функциональных возможностей в некоторых таких процессорах серии DSP9600x введены сопроцессоры реализации алгоритмов цифровой фильтрации. Цифровой сигнальный процессор MSC8101, структура которого приведена на рис.3.35, представляет собой устройство, которое объединяет цифровой сигнальный процессор и коммуникационный процессор, построенный на основе RISC-архитектуры. Он изготовлен по технологии 0.13 мкм, потребляет 500МВт при напряжении питания 1.5В и поддержке внешнего напряжения 3.3В. Цифровой сигнальный процессор MCS8101, обеспечивает производительность 1200 млн. операций в секунду. В состав MCS8101 устройства входят: цифровой сигнальный процессор с 4-мя арифметико-логическими устройствами, высокоскоростной коммуникационный процессор (СРМ) с тактовой частотой 150 МГц, блок системного интерфейса (SIU), ОЗУ объемом 4Мбайт, программируемый контроллер памяти, интерфейс шин, 16-канальный контроллер прямого доступа к памяти, сопроцессор цифровой фильтрации (EFCOP), устройство тестирования и диагностики (JTAG), интерфейс хост-процессора.

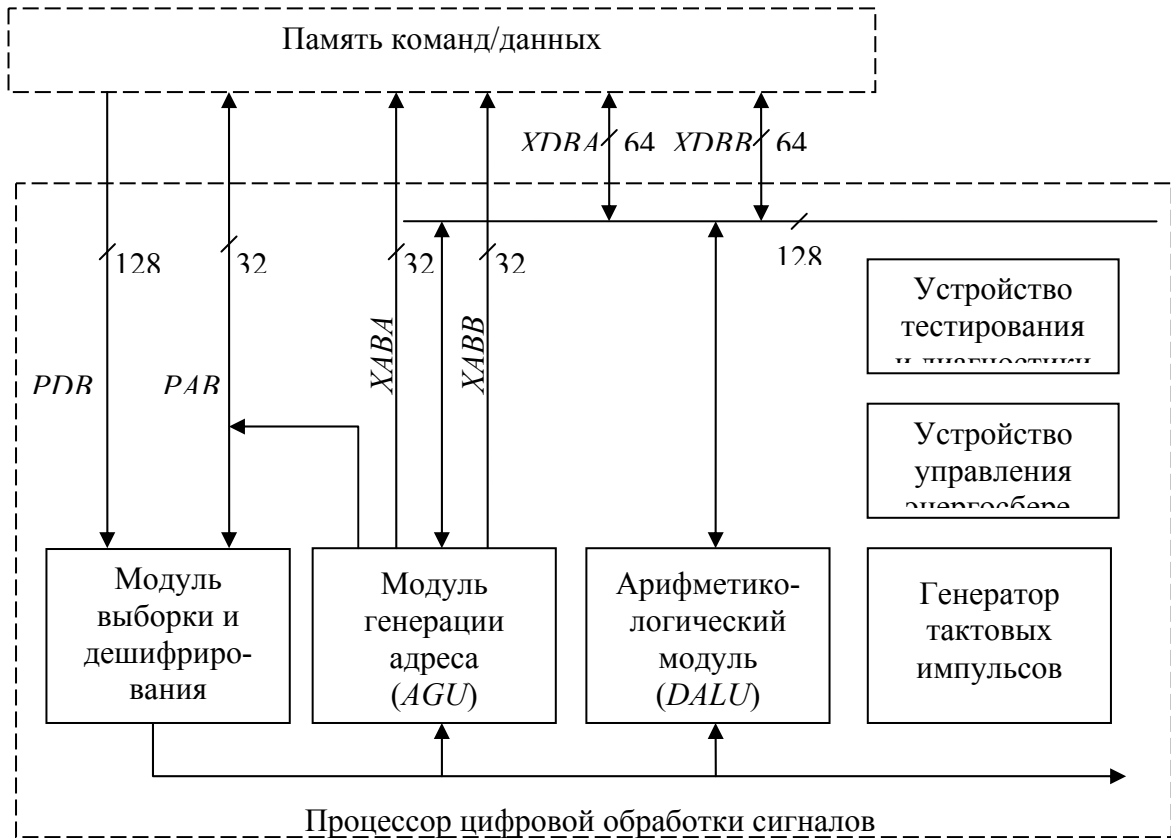


Рис. 3.37 Структура процессора цифровой обработки сигналов SC140

Коммуникационный процессор содержит микропроцессор на основе *RISC*-архитектуры, два последовательных интерфейса (*SI*), два многоканальных коммуникационных контроллера (*MCC*), два высокоскоростных коммуникационных контроллера (*FCC*), два контроллера управления (*SMC*), последовательный порт (*SPI*), интерфейс  $I^2C$ , восемь генераторов тактовой частоты (*BRG*), четыре таймера общего назначения.

Сопроцессор цифровой фильтрации (*EFCOP*) предоставляет возможность программным путем реализовать 32-разрядный цифровой фильтр с конечной импульсной характеристикой (*КИХ*-фильтр), цифровой фильтр с бесконечной импульсной характеристикой (*БИХ*-фильтр) или цифровой адаптивный фильтр.

Внутренняя 64-разрядная шина обеспечивает связь процессора цифровой обработки сигналов, коммуникационного процессора, блока системного интерфейса и внутренней памяти. Системная шина обеспечивает доступ устройств, входящих в структуру цифрового сигнального процессора *MSC8101* к внешней памяти и другим внешним устройствам.

Кроме того, имеются 128-разрядная шина команд (*P-bus*) и две 64-разрядные шины данных (*XADB* и *XBDB*), обеспечивающие связь процессора цифровой обработки сигналов с внутренней памятью и периферийными устройствами.

Развитием *MSC8101* является цифровой сигнальный процессор *MSC8102*, обобщенная структурная схема которого приведена на рис.3.36. Этот процессор содержит четыре отдельных процессора цифровой обработки сигналов *SC140*, четыре модуля последовательного интерфейса с временным разделением каналов (*TDM*) с аппаратной поддержкой процедур кодирования. Эти модули обеспечивают работу 1024 каналов ввода-вывода. Кроме того, цифровой сигнальный процессор *MSC8102* содержит универсальный последовательный порт (*UART*) с 16 каналами ввода-вывода, контроллер прямого доступа к памяти (*DMA*), внутреннее ОЗУ, емкостью 1436 Кбайт, устройство начальной загрузки, интерфейс хост-процессора. Каждый процессор цифровой обработки сигналов *SC140* имеет четыре исполнительных блока и выполняет 1100 миллионов операций умножения с накоплением в секунду, работая с тактовой частотой 275 МГц.

Процессор цифровой обработки сигналов *SC140* (рис.3.37) имеет следующие функциональные модули.

- Арифметико-логический модуль устройства обработки данных (*DALU*), содержащий четыре независимых арифметико-логических устройств



ва (*ALU*) и буферные регистры. В каждом арифметико-логическом устройстве имеется 40-битное устройство барабанного сдвига.

- Модуль генерации адреса (*AGU*) содержит два модуля вычисления адреса (*AAU*) и буферный регистр адреса.
- Модуль выборки и дешифрования команд (*PSEQ*).
- Генератор тактовых импульсов с устройством фазовой автоподстройки частоты (*PLL*), синхронизирующий работу всех устройств.
- Устройство управления энергосбережением.
- Устройство тестирования и диагностики.

Для обмена данными между арифметико-логическими устройствами и памятью используются две шины адреса памяти данных (*XABA*, *XABB*) и две шины данных памяти данных (*XDBA* и *XDBB*). Шина данных и шина адреса команд (*PDB* и *PAB*) служат для передачи команд из памяти в устройство дешифрации команд.

Процессор цифровой обработки сигналов SC140 может выполнять до шести команд за один такт. При этом данные передаются в цифровом сигнальном процессоре MSC8102 по системной шине, через параллельный порт или по многоканальному последовательному интерфейсу. Процессоры цифровой обработки сигналов SC140 осуществляют вычисления и данные передаются по одному из портов.

Таблица 3.9

Цифровые сигнальные процессоры	Частота МГц	Производительность млн. операций в сек / (млн. операций в сек в формате с плавающей точкой)	Объем ОЗУ (бит)	Объем ПЗУ (бит)	Контроллеры прямого доступа памяти (число каналов)	Таймеры	Последовательные порты
DSP56300	100	100	48К		6	3	3
DSPA5637	150	150	480К	960К	6	3	9
MC56F8322	60	60	256К	576К		2	4
DSP56F827	80	40	80К	1М		2	4
DSP9602	40	200	64К	34К	1	2	2
MSC8101	300	1500 (млн. операций умножения)	4М		1(16)		1
MSC8102	300	4400 (млн. операций умножения)	11.5М		1(16)		1

Архитектура цифрового сигнального процессора MSC8102 построена таким образом, чтобы эффективно использовать вычислительную мощность процессора, составляющую 4.4 миллиарда операций в секунду.

Основные технические характеристики некоторых цифровых сигнальных процессоров рассмотренных выше серий приведены в таблице 3.9. Как следует из анализа этой таблицы, основным направлением развития является повышение производительности, увеличение быстродействия и объемов памяти за счет перехода к модульному принципу построения процессоров. Возрастают возможности работы с прямым доступом к памяти, что является характерным при построении различных цифровых рекурсивных фильтров, применяемых в системах связи и телекоммуникационных системах.

### **3.2.4 Комбинированные высокопроизводительные микропроцессоры**

В комбинированных высокопроизводительных микропроцессорах используется сочетание архитектурных решений, характерных как для цифровых сигнальных процессоров, так и для универсальных микропроцессоров. Комбинированные микропроцессоры предназначены для обработки видеоизображений, графических изображений, аудиосигналов, а также для решения коммуникационных задач в мультимедиа системах (персональных компьютерах, бытовой технике и др.). В качестве примеров рассмотрим два комбинированных высокопроизводительных процессора TriMedia и MC9328MXL.

Микропроцессор TriMedia используется в качестве сопроцессора цифровой обработки сигналов, разгружающего основной процессор мультимедийной системы, а также применяется как основной процессор в мультимедийных устройствах (проигрывателях CD-дисков, игровых приставок к телевизорам и компьютерам и др.).

Структура комбинированного высокопроизводительного микропроцессора TriMedia (рис.3.38) содержит высокоскоростную шину со скоростью передачи 400 Мбит/с, связывающую цифровой сигнальный процессор, построенный на основе RISC-архитектуры, который обрабатывает 5 команд за один такт, сопроцессор обработки изображений, коммуникационный блок, декодер видеосигнала (MPEG-декодер), каналы прямого доступа к памяти.

Эффективная для мультимедийных приложений система команд TriMedia позволяет осуществлять одновременное декодирование аудио и видео-

данных при использовании 22% вычислительных ресурсов и 12% ресурсов памяти [12].

Дальнейшее развитие архитектуры построения комбинированных процессоров получило в микропроцессоре MC9328MXL.

Комбинированный высокопроизводительный микропроцессор MC9328MXL представляет собой интегрированную однокристалльную систему, содержащую несколько микропроцессоров. Он используется в сотовых телефонах со смарт-картами, системах Blue Tooth, в устройствах передачи видеоизображений. Микропроцессор MC9328MXL имеет тактовую частоту до 200 МГц. Характерной особенностью этого микропроцессора является низкое энергопотребление. Напряжение питания центрального микропроцессора составляет от 1.7 до 2.0В, а напряжение питания устройств ввода-вывода равно 1,7–3,3 В. Интегральная микросхема расположена в корпусе, имеющим 256 контактов.

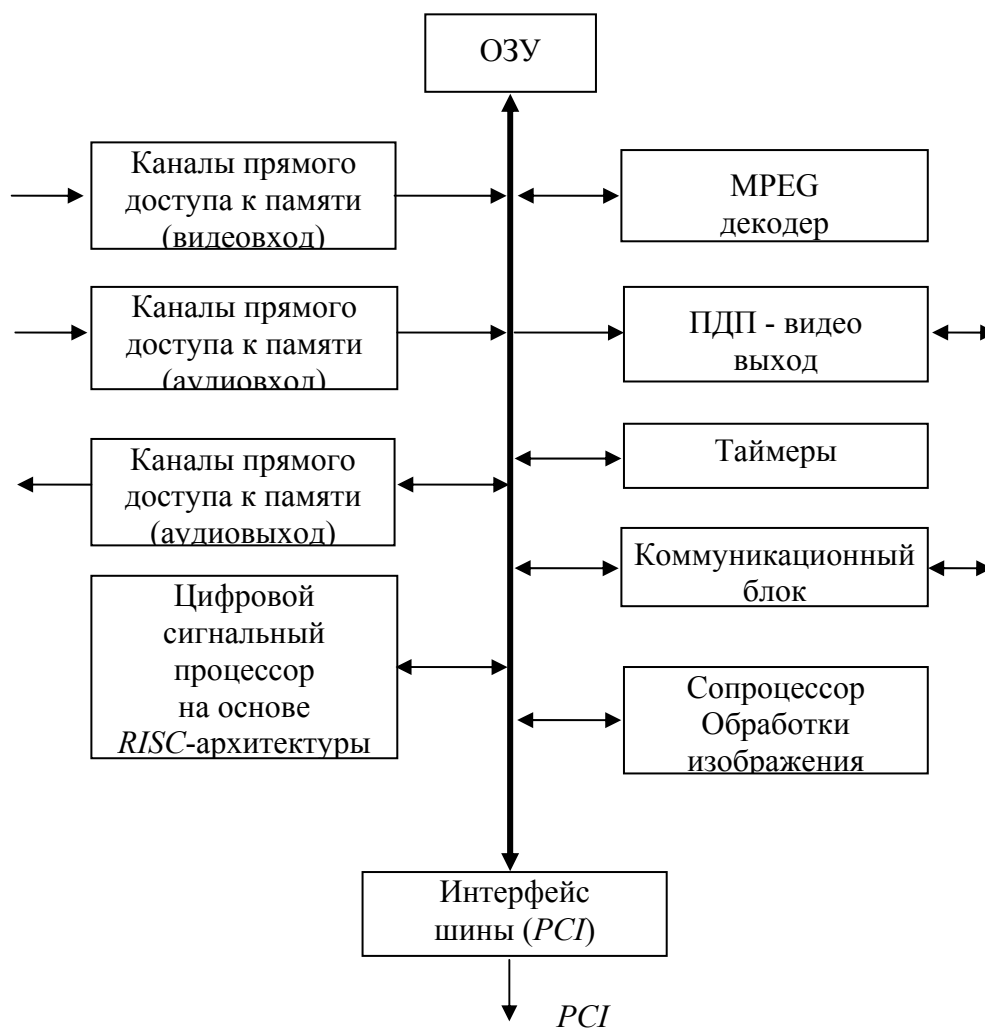


Рис. 3.38 Структура комбинированного высокопроизводительного микропроцессора TriMedia

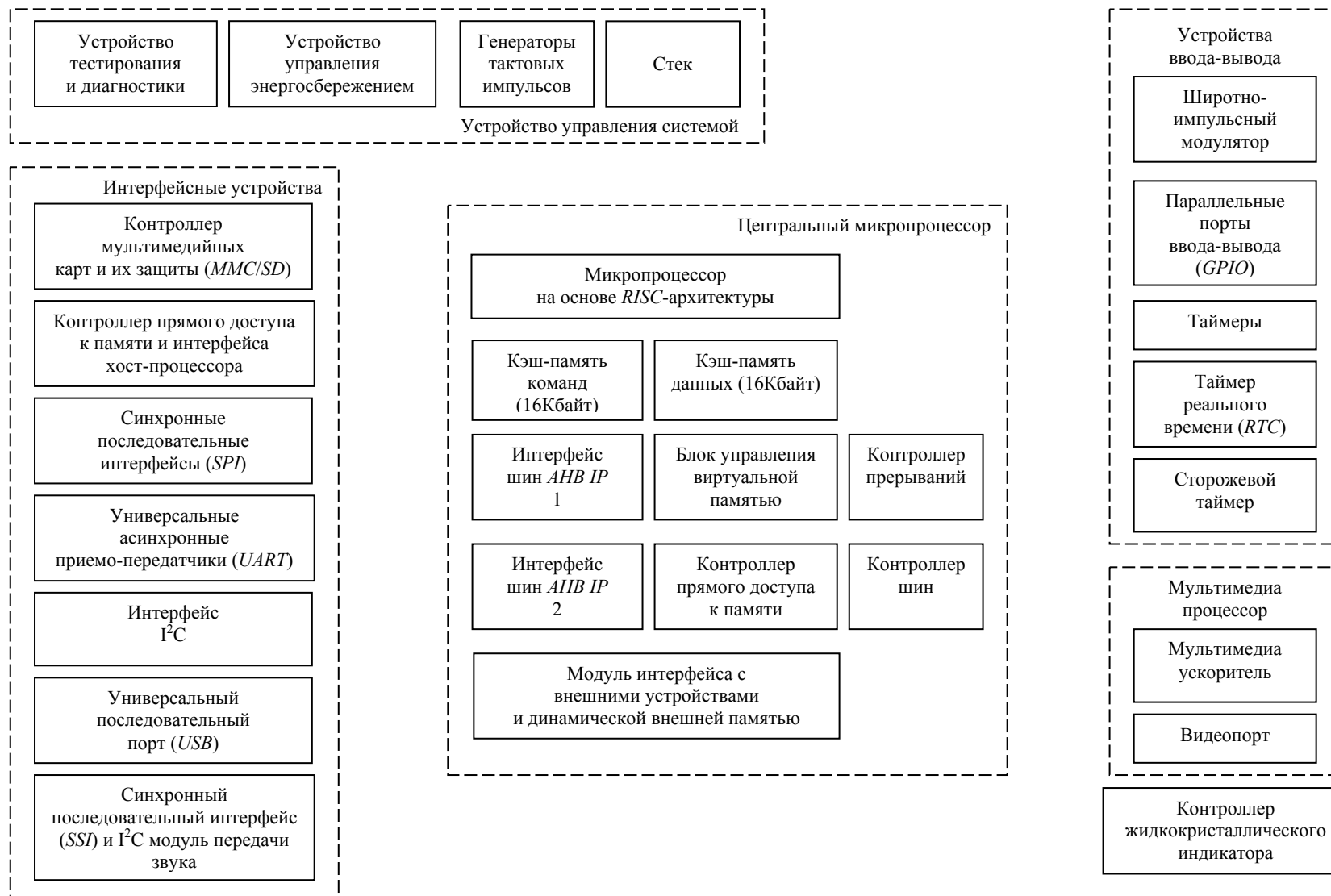


Рис. 3.39 Обобщенная структура комбинированного высокопроизводительного микропроцессора с мультимедиа расширениями MC9328MXL

Обобщенная структура комбинированного высокопроизводительного микропроцессора с мультимедиа расширениями MC9328MXL приведена на рис. 3.33.

В состав микропроцессора входят:

- Центральный микропроцессор, содержащий 32-разрядный микропроцессор на основе *RISC*-архитектуры, кэш-память команд и данных, блок управления виртуальной памятью (*VMMU – Virtual Memory Management Unit*), контроллеры прерываний и прямого доступа к памяти, контроллер шин, два интерфейса (*AIP1*) между высокоскоростной шиной (*AHB - Advanced High-performance Bus*) и низкоскоростной шиной периферийных устройств (*IP*) и модуль интерфейса с внешними устройствами и динамической внешней памятью.

- Интерфейсные устройства. Они состоят из 2 синхронных периферийных интерфейса (*SPI*), двух универсальных асинхронных приемопередатчика (*VART*), двухпроводного (*I<sup>2</sup>C*) интерфейса, синхронного последовательного интерфейса (*SSI*) с модулем передачи звука (*I<sup>2</sup>C*), контроллера прямого доступа к памяти и интерфейса хост-процессора, универсального последовательного порта (*VSB*) и контроллера мультимедийных карт и их защиты (*MMC/SD*).

- Устройство управления системой, состоящее из генераторов тактовых импульсов с системой цифровой фазовой автоподстройки частоты, стековой памяти, устройства управления энергосбережением и устройства тестирования и диагностики системы.

- Устройства ввода-вывода системы, в число которых входят параллельные порты ввода-вывода (*GPIO*), таймеры, таймер реального времени (*RTC*), сторожевой таймер, широтно-импульсный модулятор.

Кроме того, в состав комбинированного высокопроизводительного микропроцессора MC9328MXL входят мультимедиа процессор, состоящий из мультимедиа ускорителя и видеопорта, а также контроллер жидкокристаллического индикатора (ЖКИ на 640x512 пикселей).

Такая интегрированная однокристалльная система является эквивалентной структуре микропроцессора Pentium MMX, Pentium II (см. раздел 2) при их использовании во встраиваемых портативных вычислительных устройствах.

## Литература

1. Алексенко А.Г., Шагурин И.И. Микросхемотехника: учебное пособие для вузов. – М.: Радио и связь, 1990. – 496 с.
2. Астапкович А.М. Семинары ASK Lab 1997 /Под редакцией Сергеева М.Б. – СПб.: Политехника, 1998. – 134 с.
3. Бродин В.Б., Шагурин И.И. Микроконтроллеры: справочник – М.: ЭКОМ, 1999. – 395 с.
4. Вычислительные машины и системы: учебное пособие для вузов. / В.Д. Ефремов, В.Ф. Мелехин, К.П. Дурандин и др. /Под редакцией В.Д. Ефремова, В.Ф. Мелехина. – М.: Высшая школа, 1993. – 292 с.
5. Гук М. Процессоры Intel: от 8086 до Pentium II. – СПб.: Питер, 1997. – 224 с.
6. Каган Б.М., Сташин В.В. Основы проектирования микропроцессорных устройств автоматики. – М.: Энергоатомиздат, 1987. – 304 с.
7. Корнеев В.В., Киселев А.В. Современные микропроцессоры. – М.: НОЛИДЖ, 2000. – 320 с.
8. Марков С. Цифровые сигнальные процессоры – М.: Микроарт, 1996.
9. Предко М. Руководство по микроконтроллерам, т.1. – М.: Постмаркет, 2001, – 488 с.
10. Предко М. Руководство по микроконтроллерам, т.2. – М.: Постмаркет, 2001. – 416 с.
11. Современные микроконтроллеры: Архитектура, средства проектирования, примеры применения, ресурсы сети Интернет. /Под редакцией Коршуна И.В. – М.: АКИМ, 1998. – 272 с.
12. Сташин В.В., Урусов А.В., Малогонцева. Проектирование цифровых устройств на однокристальных микроконтроллерах.– М.: Энергоатомиздат, 1990..
13. Угрюмов Е.П. Цифровая схемотехника: учебное пособие. – СПб.: БХВ – Петербург, 2005. – 528 с.
14. Шагурин И.И. Микропроцессоры и микроконтроллеры фирмы Motorola: справочное пособие. – М.: Радио и связь, 1998. – 560 с.
15. Atmel Corporation 8051 Flash Microcontroller Data Book, 1997.
16. С.Б.Макаров, И.А.Цикин Передача дискретных сообщений по радиополосам с ограниченной полосой пропускания. - М.: Радио и связь, 1988. – 304 с.

17. Радиотехнические системы передачи информации: Учебное пособие для вузов/ В.А. Борисов, В.В. Калмыков, Я.М. Ковальчук и др.; Под ред. В.В. Калмыкова. – М.: Радио и связь, 1990. –304 с.

18. Г.А. Емельянов, В.О. Шварцман Передача дискретной информации.: Учебник для вузов. – М.: Радио и связь, 1982. –240 с.

19. В.А. Варгаузин Цифровая обработка сигналов. Минимаксные аппроксимации для задач цифровой фильтрации: Учебное пособие для вузов. – СПб.: изд. СПбГПУ, 2004 – 87 с.

20. Под ред. Д.В. Пузанкова. Микропроцессорные системы. – СПб.: Политехника, 2002.

21. В.И. Бойко и др. Схемотехника электронных схем. Микропроцессоры и микроконтроллеры. – СПб.: БХВ – Петербург, 2004. - 464 с.