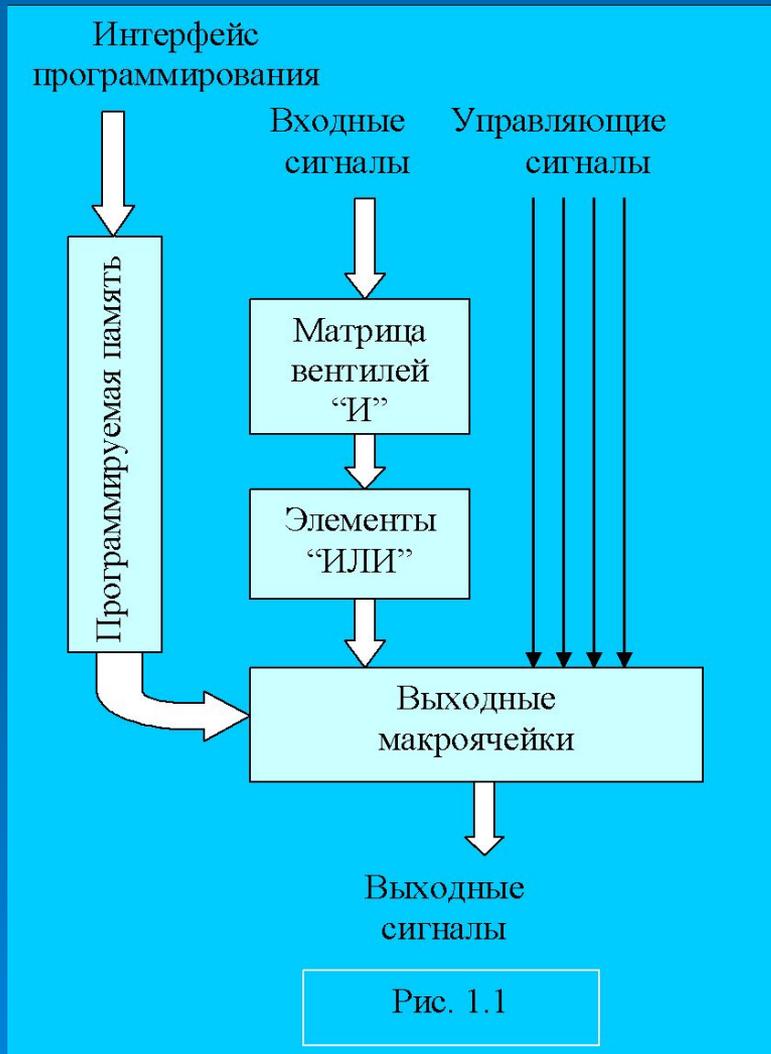


Обзор программируемых логических интегральных схем и интегральных схем гибкой логики



При этом учитывается тот факт, что любое логическое выражение может быть преобразовано в эквивалентную форму логической суммы произведений (минимальной дизъюнктивной нормальной формы МДНФ) [5] и таким образом реализовано на основе архитектуры матриц элементов "И" и "ИЛИ".

Интерфейс программирования и тестирования JTAG/IEEE 1149.1

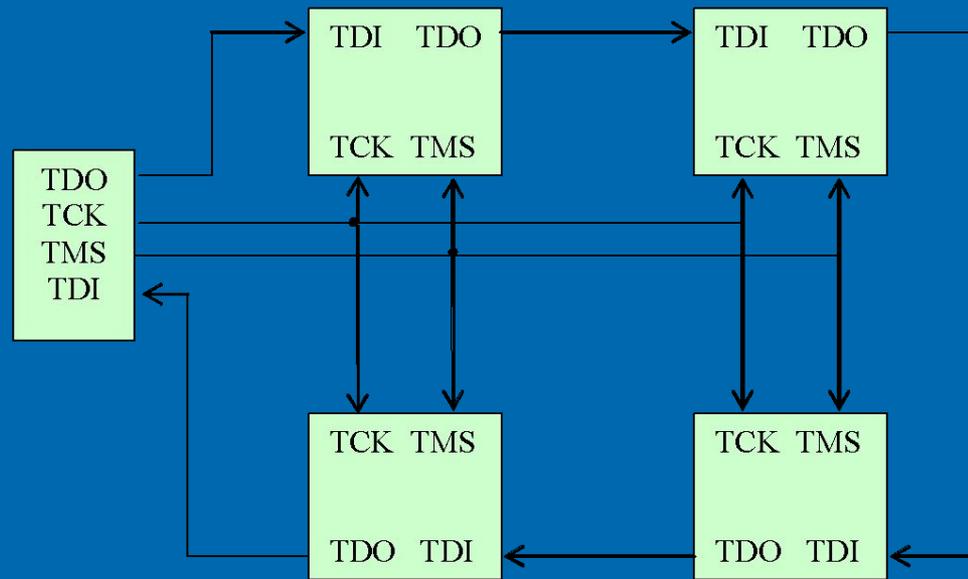


Рис. 1.2 Соединение 4 БИС интерфейсом тестирования.

При этом регистры интерфейса в БИС образуют соединяющими их информационными сигналами TDI и TDO единый последовательный однонаправленный сдвиговый регистр, а управляющий и сигнал синхронизации являются обычными параллельно подсоединяемыми к БИС сигналами.

Порт тестирования и программирования в каждой БИС управляется контроллером JTAG интерфейса. Структурная схема порта тестирования и программирования показана на рис. 1.3.

Интерфейс программирования и тестирования JTAG/IEEE 1149.1

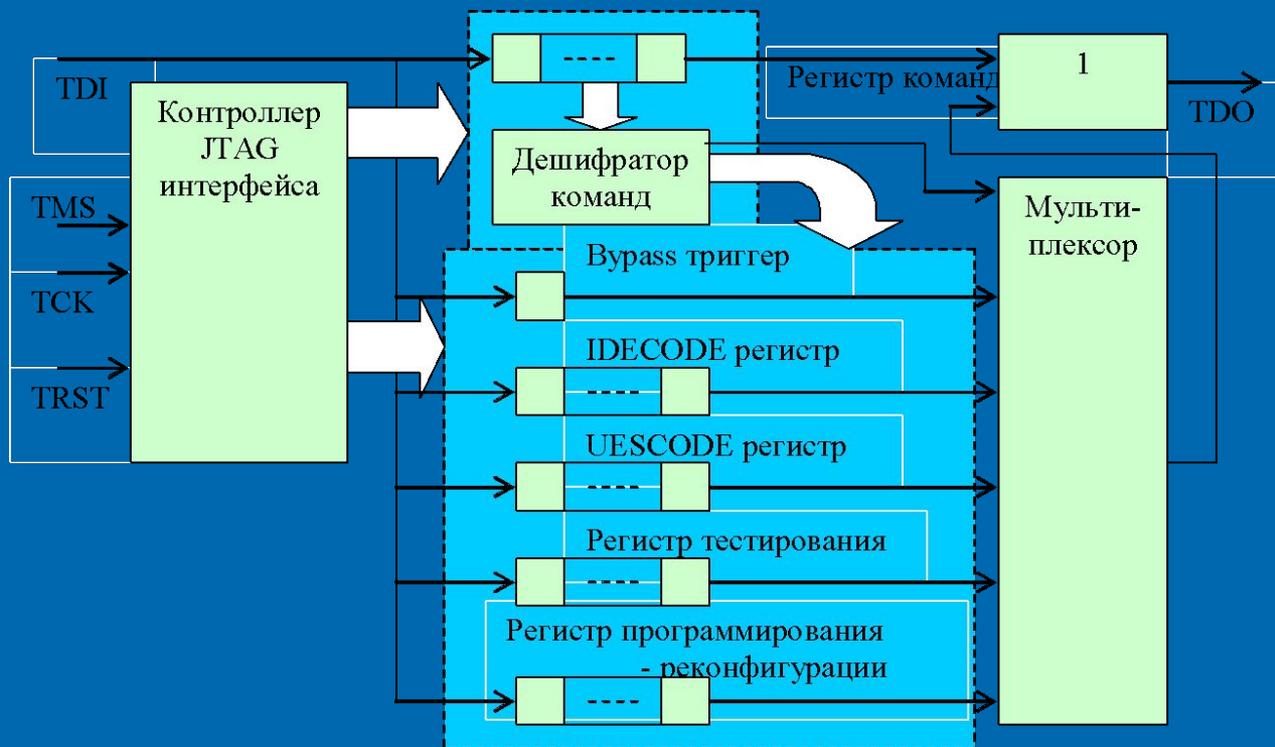


Рис. 1.3. Структурная схема порта тестирования и программирования.

Контроллер JTAG интерфейса сбрасывается в исходное состояние сигналом TRST. После этого он готов принимать команды по входу TDI последовательно в регистр команд. Контроллер JTAG интерфейса по переднему фронту сигнала TCK формирует все сигналы управления регистрами данных и команд. Стандартом не оговаривается длина регистра команд, в зависимости от сложности микросхемы он может содержать разное число бит. Признаком окончания приема команды служит установка в логическую единицу сигнала TMS. При этом контроллер JTAG интерфейса дешифрирует команду в дешифраторе и подсоединяет требуемый регистр данных на ввод или вывод. После сброса сигнала TMS в ноль контроллер готов выполнять команду и принимать или передавать данные по TDI и TDO.

Интерфейс программирования и тестирования JTAG/IEEE 1149.1

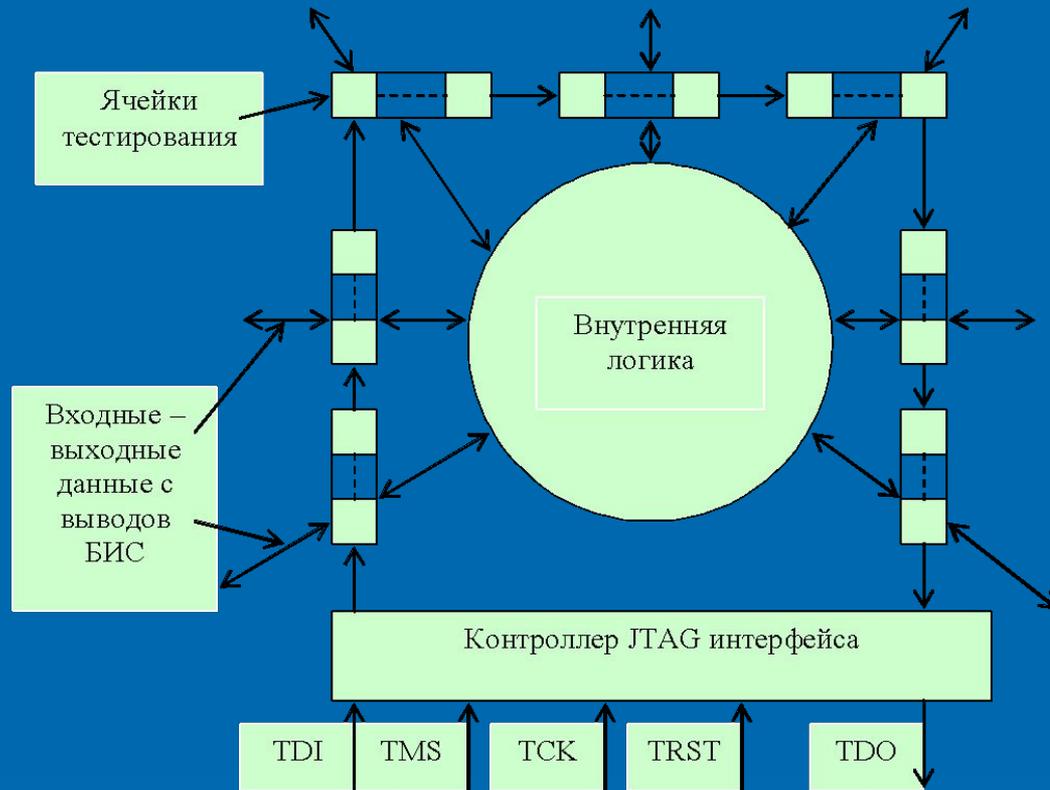


Рис. 1.4. Ячейки периферийного тестирования.

С точки зрения контроллера JTAG интерфейса они образуют один сдвиговый регистр для передачи данных от TDI к TDO. Для информационных данных с выводов БИС они являются запоминающими ячейками, причем запоминая данные, они их в то же время пропускают насквозь во (из) внутреннюю логику и регистры БИС, не влияя на общий алгоритм работы микросхемы.

КЛАССИФИКАЦИЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ЦИФРОВЫХ

ВЫЧИСЛИТЕЛЕЙ

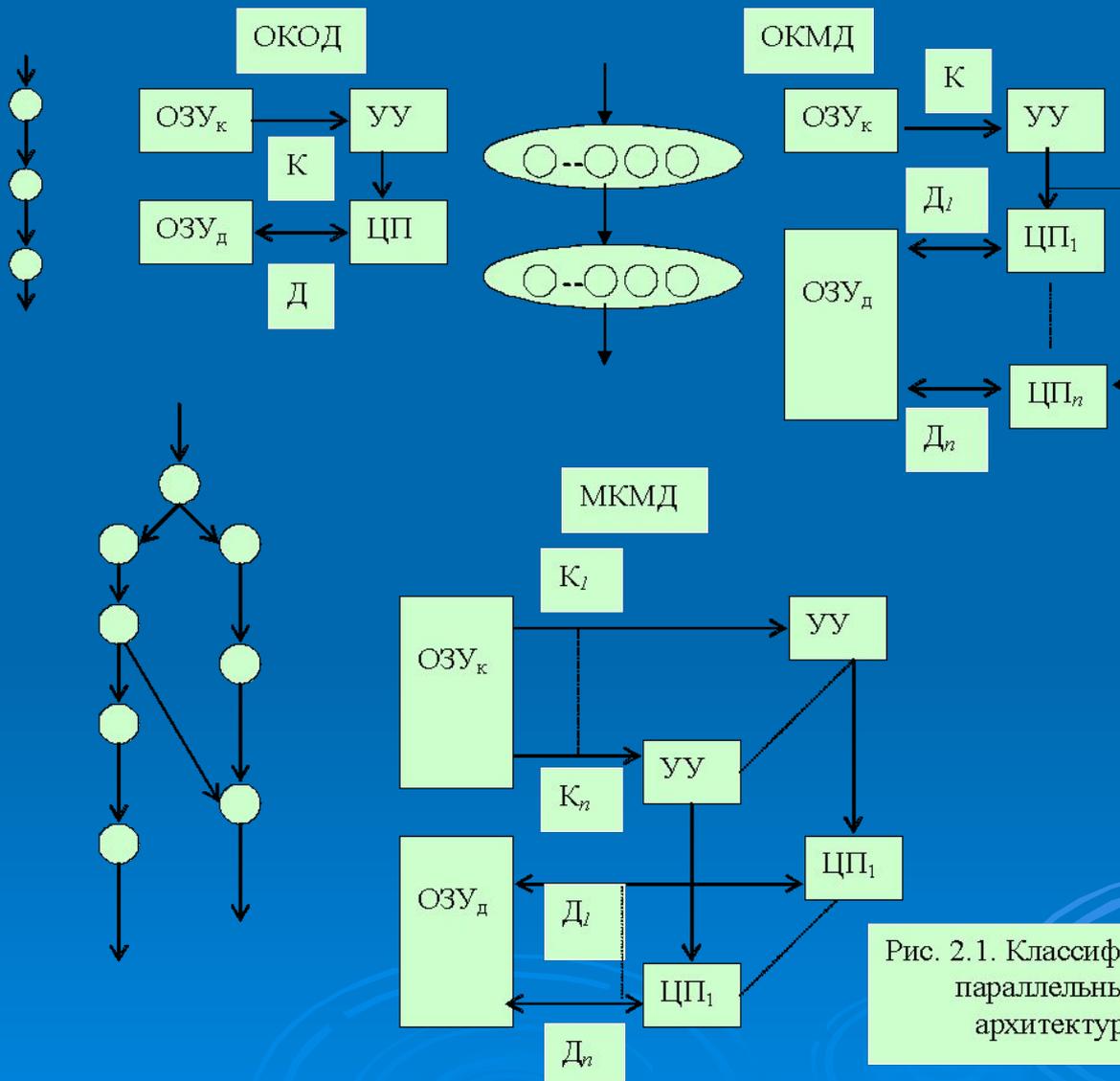


Рис. 2.1. Классификация параллельных архитектур

Векторные команды и матричная вычислительная система

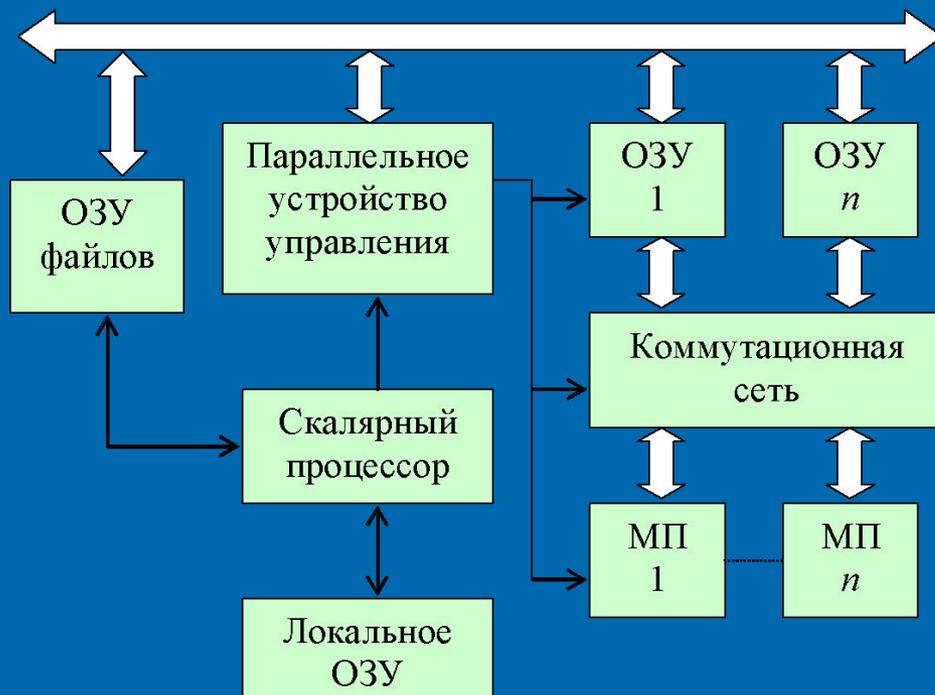


Рис. 3.1

Векторные команды задают операции над элементами одного или нескольких векторов. Исполнение векторной команды включает в матричной системе чтение из памяти элементов векторов, распределение их по массиву матричных процессоров, выполнение заданной операции и занесение результатов в память. Структурная схема матричной системы показана на рис.3.1. Матричные процессоры МП однотипны и выполняют общую для всех команду над различными элементами векторов. Управляет ими параллельное устройство управления. В такой системе для выполнения обычных скалярных команд (команд над одиночными данными) и подготовки векторных команд используется скалярный процессор - обычный производительный микропроцессор класса ОКОД. При использовании N матричных процессоров такая система в идеальном случае имеет производительность в N раз выше скалярного процессора.

Конвейеризация вычислений и организация векторных вычислителей класса ОКМД

таблица занятости конвейера

<i>CT1</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>	<i>D4</i>	<i>D5</i>
<i>CT2</i>	-	<i>D1</i>	<i>D2</i>	<i>D3</i>	<i>D4</i>
<i>CT3</i>	-	-	<i>D1</i>	<i>D2</i>	<i>D3</i>
<i>CT4</i>	-	-	-	<i>D1</i>	<i>D2</i>
	<i>t1</i>	<i>t2</i>	<i>t3</i>	<i>t4</i>	<i>t5</i>

Основы проектирования аппаратных средств конвейерных

вычислителей

При проектировании конвейерных устройств, следует обратить внимание на логическое проектирование логики и фиксаторов ступеней, учесть проблемы синхронизации быстродействующих конвейеров [3].

Особенности проектирования логики ступени конвейера проиллюстрированы на рис.

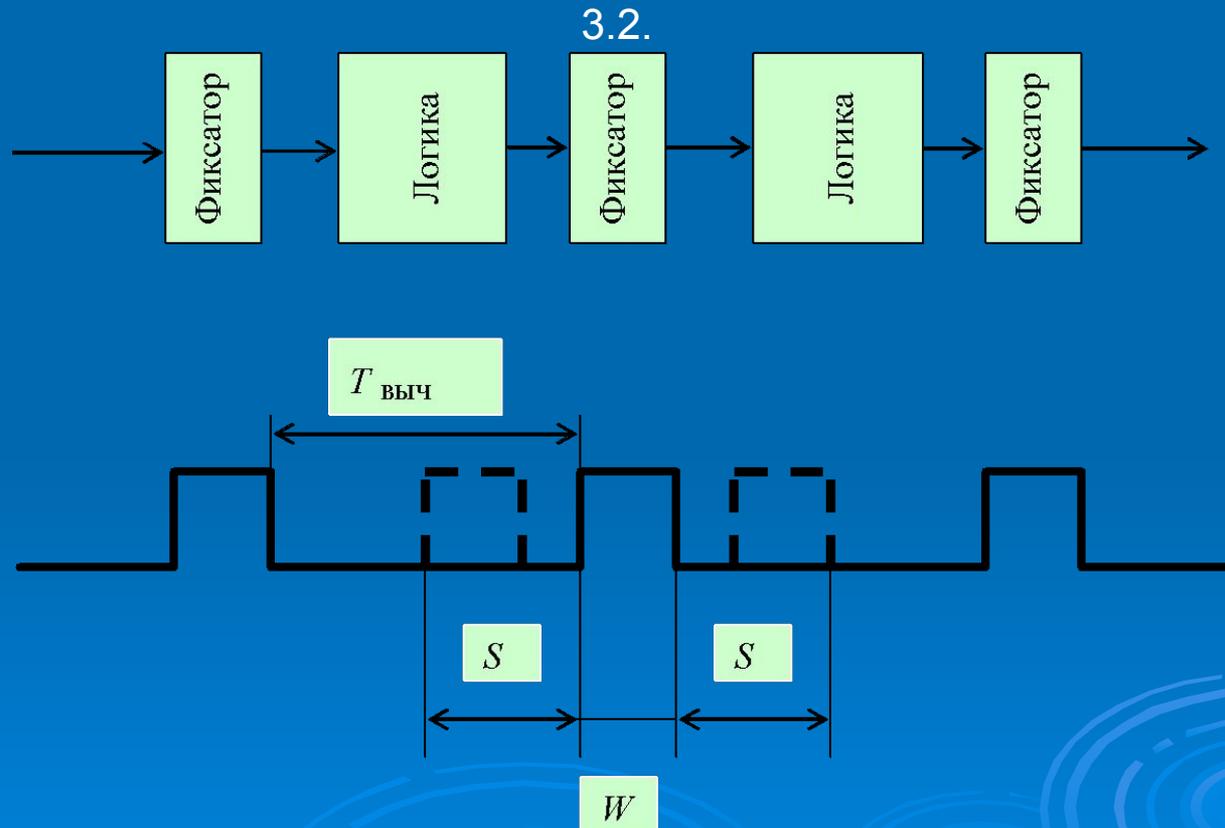


Рис. 3.2

Основы проектирования аппаратных средств конвейерных

вычислителей

Таким образом, условие правильного функционирования логики ступени конвейера:

$$T_{max} < T_{выч} - S, \quad T_{min} > W + S,$$

где T_{max} - время вычисления подфункций по максимальному логическому пути, T_{min} - время вычисления подфункций по минимальному логическому пути. Для быстродействующих конвейеров часто логику ступени объединяют с быстродействующим фиксатором ступени. В качестве быстродействующего фиксатора обычно используют логическую схему триггера Эрла (рис.3.3).

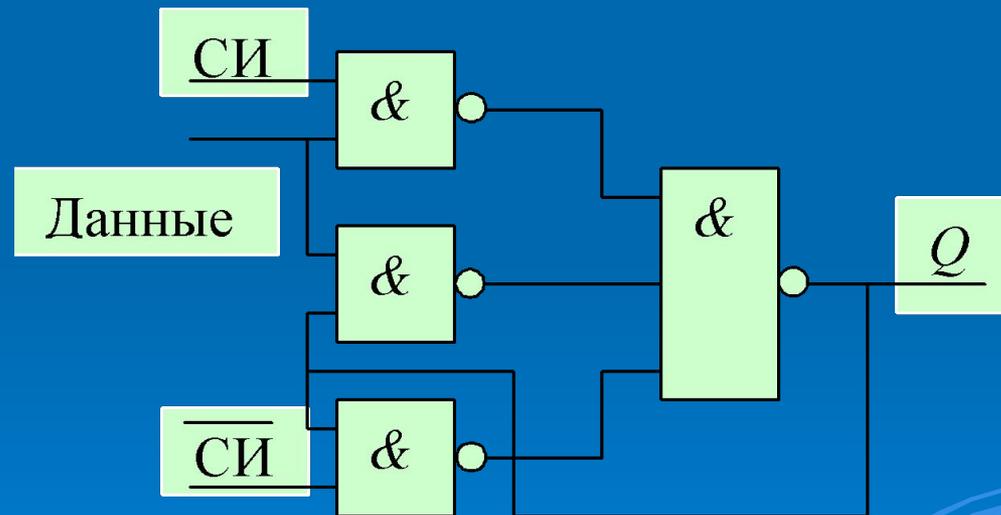


Рис. 3.3

Основы проектирования аппаратных средств конвейерных

вычислителей

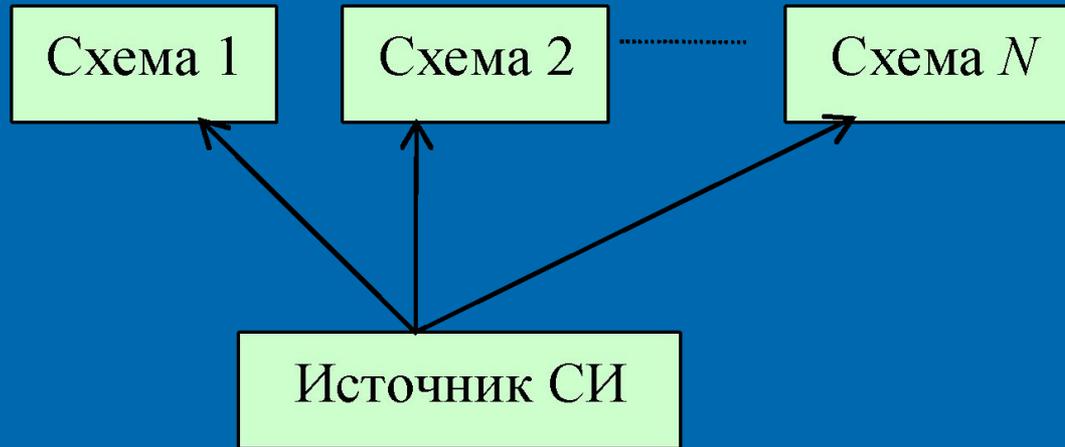


Рис. 3.4

Второй метод пакетирования СИ использует временную цепь передачи, которая включает в свой состав формирующие триггерные каскады передачи, устраняющие перекосы при распространении СИ (рис.3.5).

Описанные выше особенности логического проектирования специализированных конвейерных вычислителей необходимо знать разработчикам радиоэлектронной аппаратуры, поскольку развитие высокоскоростных ПЛИС позволяет реализовать их максимальное быстродействие только на основе таких параллельных архитектур.

Основы проектирования аппаратных средств конвейерных

вычислителей

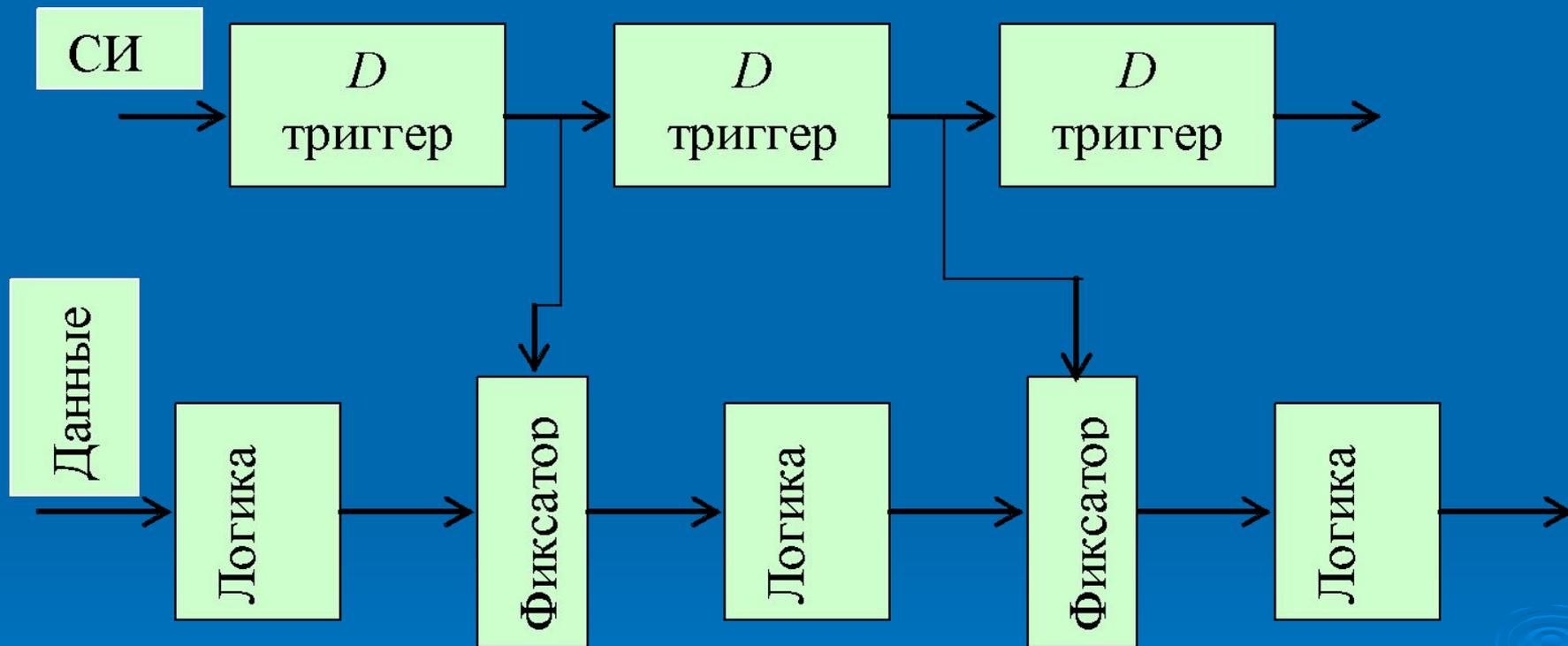


Рис. 3.5

Многопроцессорные комплексы

Вычислительные алгоритмы на основе ряда слабосвязанных потоков команд и данных реализуются в многопроцессорных комплексах класса МКМД. Основное внимание в таких комплексах уделяется синхронизации и взаимосвязи между параллельно функционирующими вычислительными процессами на ряде МП.

Применяются в основном два механизма взаимодействия процессов:

- использование общей многопортовой оперативной памяти для организации синхронизации и взаимодействия;
- работа по принципу обмена сообщениями процессов друг с другом через высокоскоростные линии связи. Структурные схемы двух таких параллельных архитектур показаны на рис. 4.1.

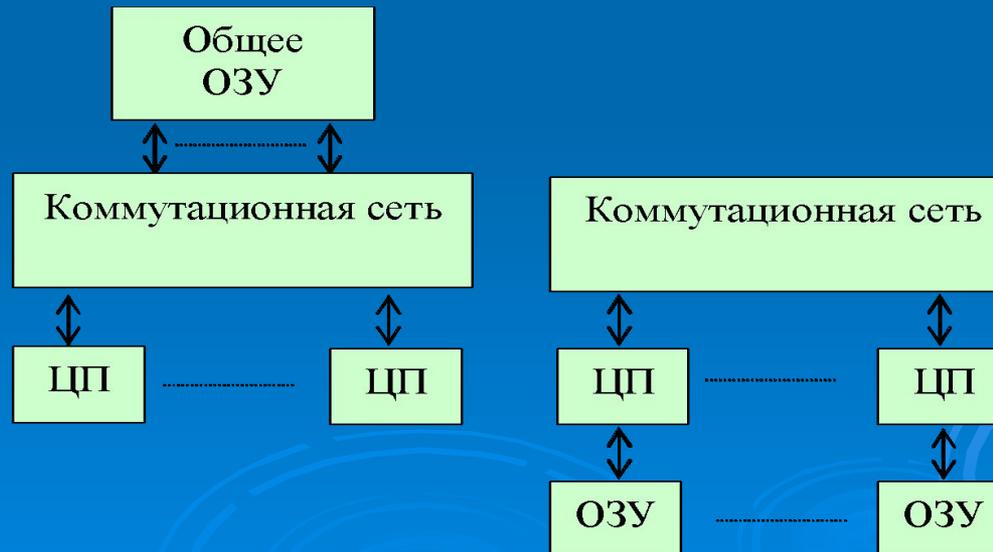


Рис. 4.1

Многопроцессорные комплексы

Для снижения требований по быстродействию к общему ОЗУ при реализации первого механизма, помимо общей, каждому процессору выделяется локальная память (рис. 4.2, а). Для уменьшения сложности коммутационной сети при реализации второго механизма несколько МП объединяются в группы - кластеры со своими локальными коммутационными сетями, переключение же таких групп осуществляется глобальной коммутационной сетью (рис. 4.2, б).



Многопроцессорные комплексы

Два фактора ограничивают производительность таких комплексов:

- организация связей между элементами комплекса, практически каждый элемент комплекса должен быть связан с другими, что часто не удается реализовать в полном объеме;
- сложность организации вычислительного процесса в многопроцессорном комплексе при параллельной работе нескольких процессоров.

Существуют три основных способа организации связей МП в многопроцессорных вычислительных комплексах:

- с общей разделяемой во времени шиной (рис. 4.3,а);
- с перекрестной коммутацией скоростных последовательных линий связи (рис. 4.3,б);
- с многопортовым общим ОЗУ (рис. 4.3,в).

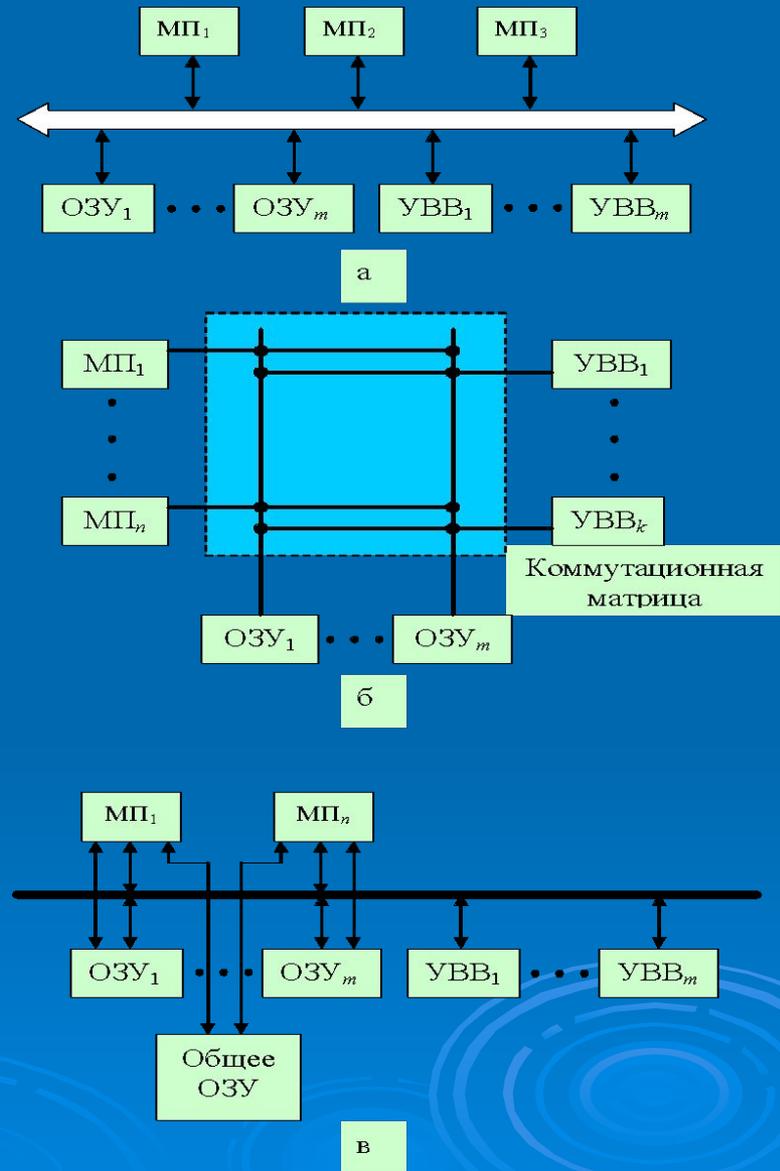


Рис. 4.3

Вычислители, управляемые потоками данных

К архитектуре МКМД вычислителей относятся потоковые параллельные вычислители, или устройства, управляемые потоками данных. Для организации процесса вычислений в таких архитектурах алгоритм вычислений строится в виде графа потоков данных (рис. 4.4).

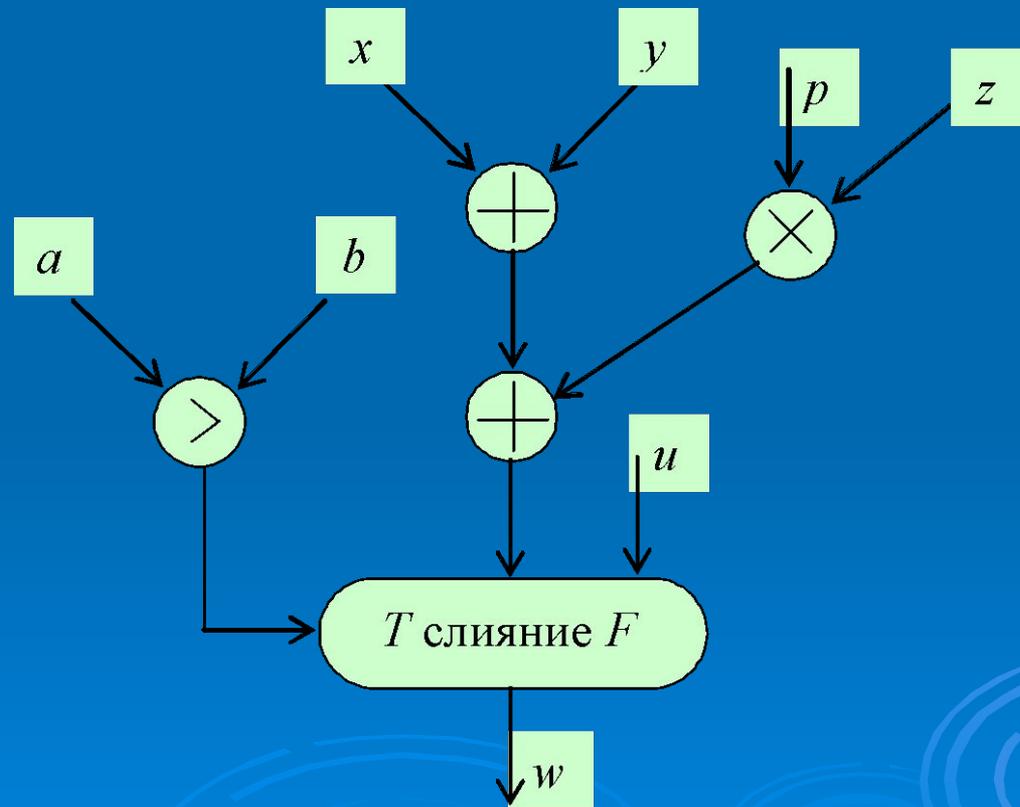


Рис. 4.4

ОРГАНИЗАЦИЯ СИСТЕМ ПАМЯТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ

ВЫЧИСЛИТЕЛЕЙ

Скоростные параллельные вычислители на несколько порядков опережают по быстродействию системы динамического ОЗУ для хранения больших массивов данных. Для обеспечения непрерывной загрузки высокопроизводительных конвейеров применяются иерархичные системы ОЗУ (рис. 5.1). В таких системах памяти блок основной динамической памяти имеет самое медленное время доступа.



Рис. 5.1

ОРГАНИЗАЦИЯ СИСТЕМ ПАМЯТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ

ВЫЧИСЛИТЕЛЕЙ

Другая причина усложнения систем ОЗУ в высокопроизводительных вычислителях - повышенные требования к защите данных при многозадачном режиме работы вычислителя. Последняя причина обусловила наличие виртуальной адресации памяти и соответствующей аппаратуры преобразования виртуальных адресов программы в физические адреса ОЗУ, как показано на рис. 5.2. Виртуальный адрес программы всегда начинается с нулевого адреса. Он суммируется с содержимым регистра базы, базовый адрес которого указывает, в какое место поместить программу в физической памяти, то есть операционная система управляет памятью.

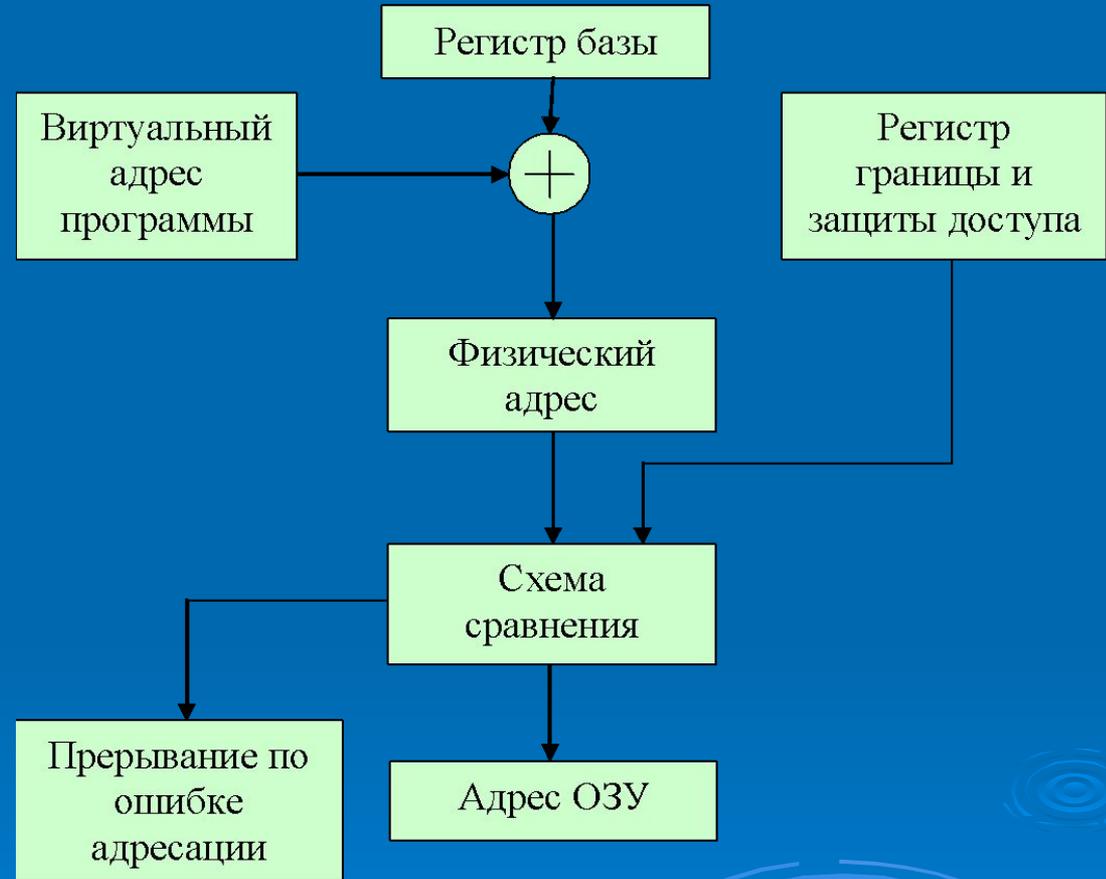


Рис. 5.2

ОРГАНИЗАЦИЯ СИСТЕМ ПАМЯТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ

ВЫЧИСЛИТЕЛЕЙ

При обращении вычислителя к КЭШ - памяти аппарата КЭШа сравнивает параллельно адреса в КЭШ - памяти с требуемым и при совпадении выдает данные по этому адресу.

Недостаток такой организации - громоздкая аппаратура параллельных компараторов сравнения адресов.

В КЭШ - памяти с прямым отображением каждый блок из основного ОЗУ занимает в КЭШе строго определенное место, тогда не требуется хранить и сравнивать адреса основной памяти в КЭШ - памяти (рис. 5.3).

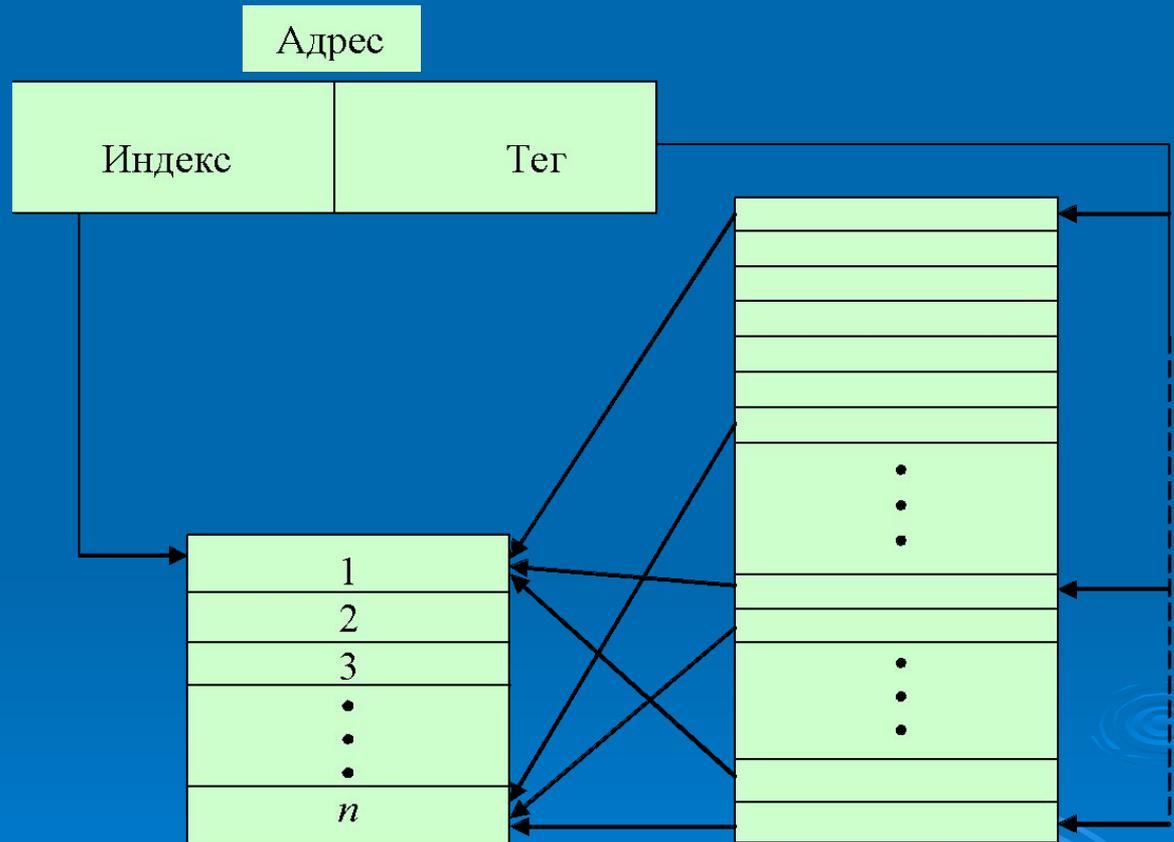


Рис. 5.3

ОРГАНИЗАЦИЯ СИСТЕМ ПАМЯТИ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЕЙ

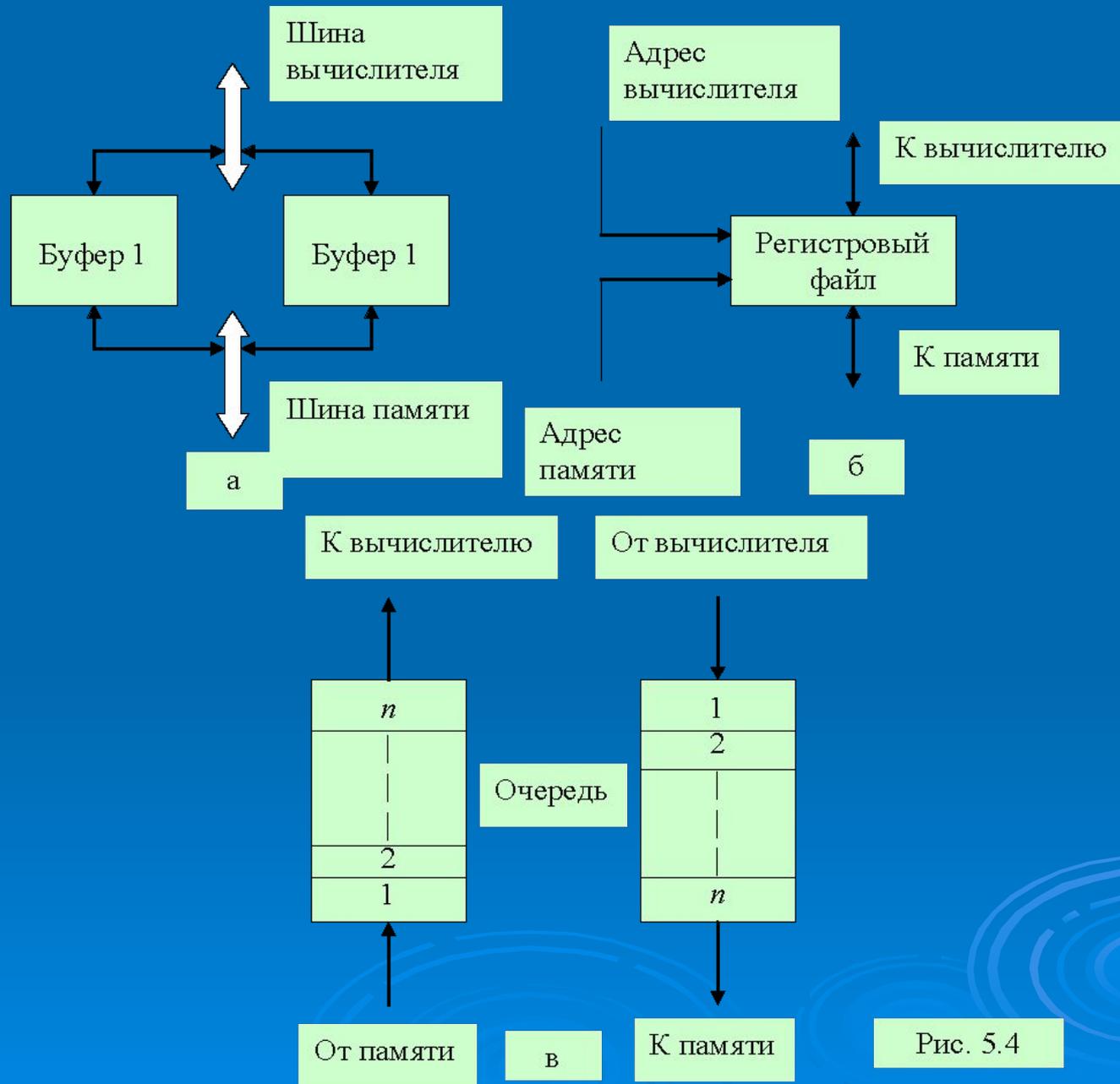
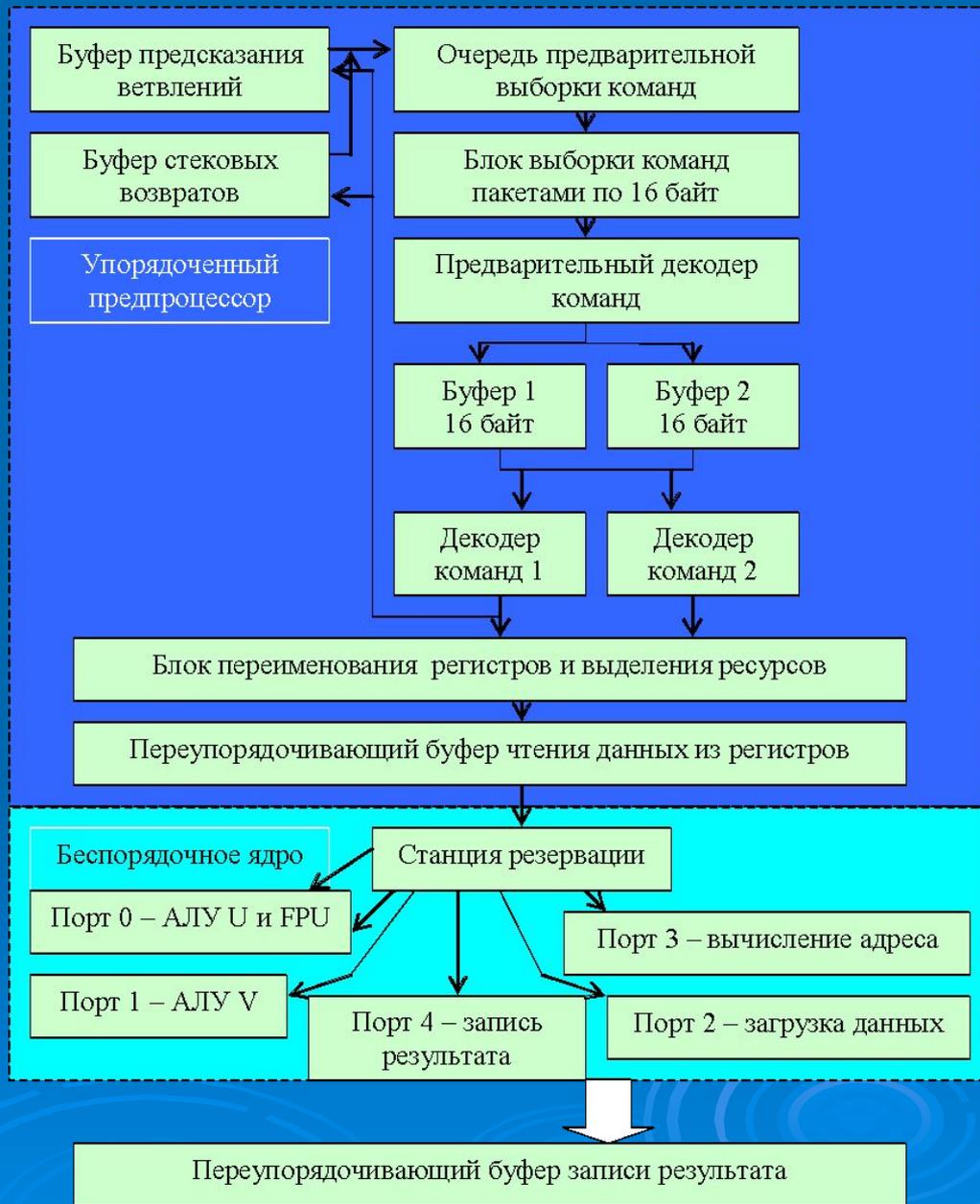


Рис. 5.4

Рис. 6.1 Структура исполнительных блоков Pentium III



Особенности архитектуры универсальных производительных микропроцессоров

Выигрыш в производительности при таком подходе происходит при выполнении условных циклов, так как разработчикам, пишущим программы на ассемблере, а также разрабатывающим языки программирования, рекомендуется команды условного выхода из цикла всегда ставить перед циклом. Выполнение команд цикла при этом происходит без потери производительности и только выход из цикла связан с потерей производительности. Достоинством такого статического предсказания ветвлений является минимальные аппаратные затраты, недостатком – наличие среднестатистической потери производительности в 40% команд ветвлений в программах. Он реализуется в относительно дешевых микропроцессорах, например, встраиваемых в управление измерительными приборами и обработку данных.

Вторая группа аппаратного предсказания ветвлений называется процессором ветвлений. На рис. 6.2 показана структура такого процессора. В этом случае для агрессивной предвыборки команд применяется две очереди. Обязательным условием является наличие в кристалле нескольких идентичных исполнительных блоков команд – арифметико-логических устройств (АЛУ1 и АЛУ2), работающих параллельно

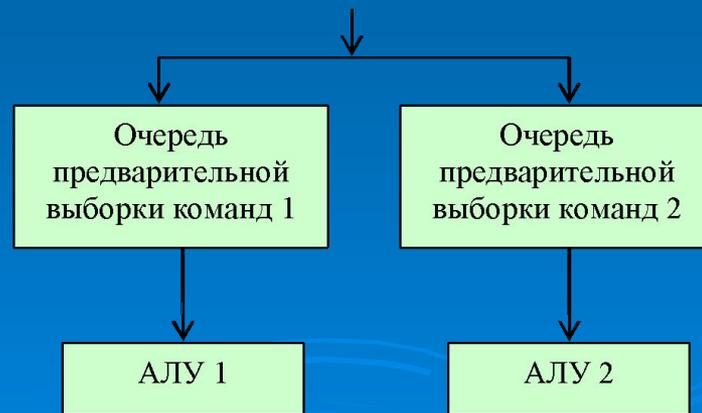


Рис. 6.2. Процессор ветвлений.

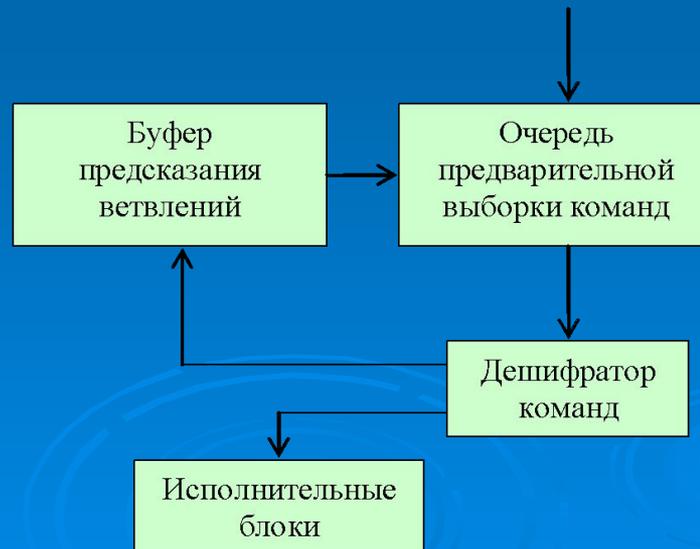
Особенности архитектуры универсальных производительных микропроцессоров

микропроцессоров

. Пока отсутствуют команды ветвлений, последовательность команд загружается в первую очередь. При появлении команды ветвления команды прямой ветки загружаются в первую очередь, а команды обратной ветки загружаются во вторую очередь. До вычисления ветвления в двух АЛУ выполняются команды прямой и обратной ветки параллельно. После вычисления ветвления принимается результат выполнения команд соответствующей ветки – той, куда нужно было перейти, а второй результат отбрасывается и далее выполняются команды из соответствующей очереди. Появление следующих команд ветвления вызывает использование двух очередей. Достоинством такого подхода является отсутствие потери производительности независимо от результата ветвления. Недостаток применения процессора ветвлений – существенные аппаратные затраты, поэтому такой подход используется только в дорогостоящих производительных процессорах, например, рабочих станциях фирмы SUN.

Третьим видом аппаратуры предсказания ветвлений является динамическое предсказание ветвлений (рис. 6.3).

Рис. 6.3.
Динамическое предсказание ветвлений.



Цифровые сигнальные процессоры.

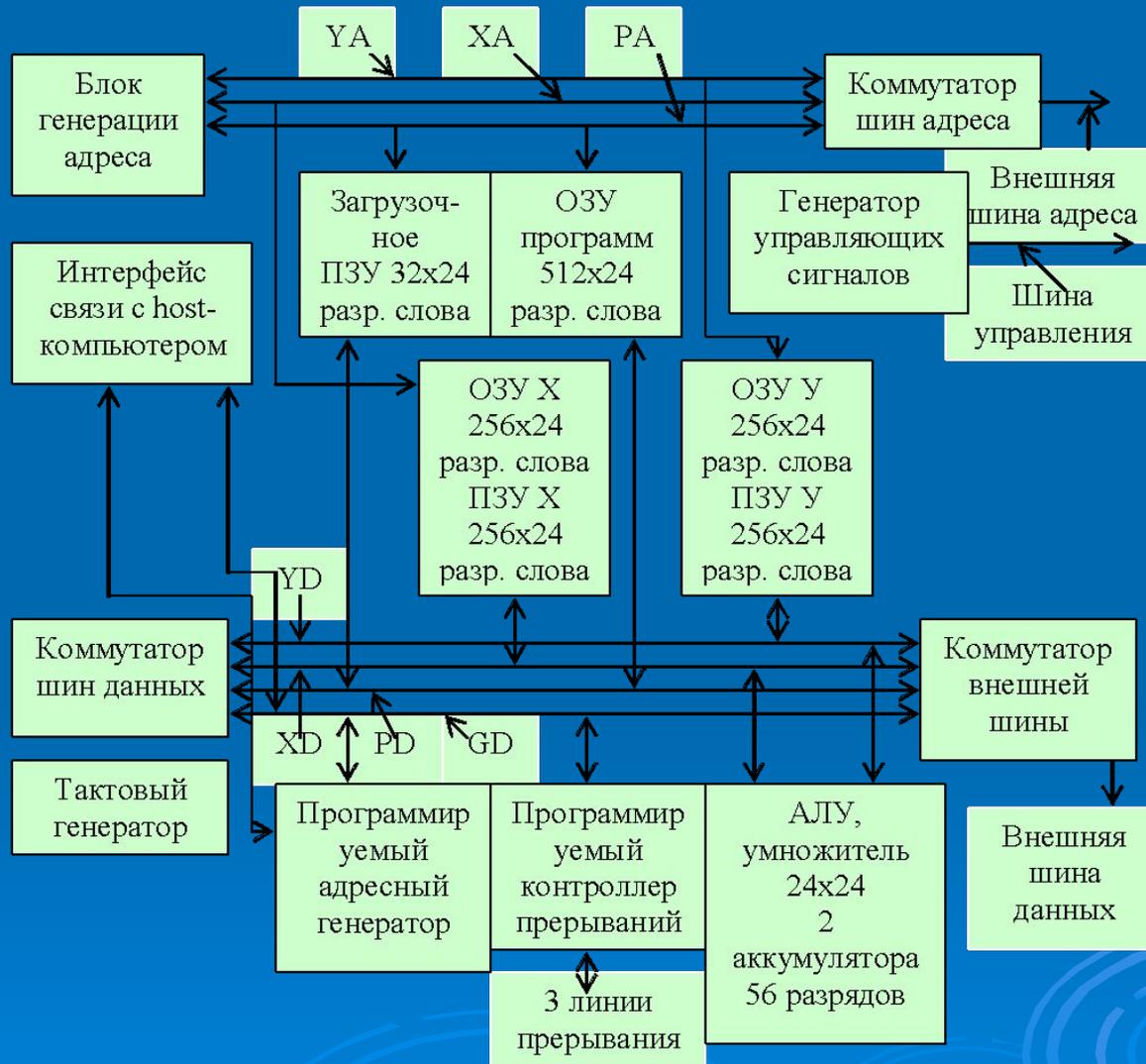


Рис. 7.1. Архитектура DSP56001

Цифровые сигнальные процессоры.

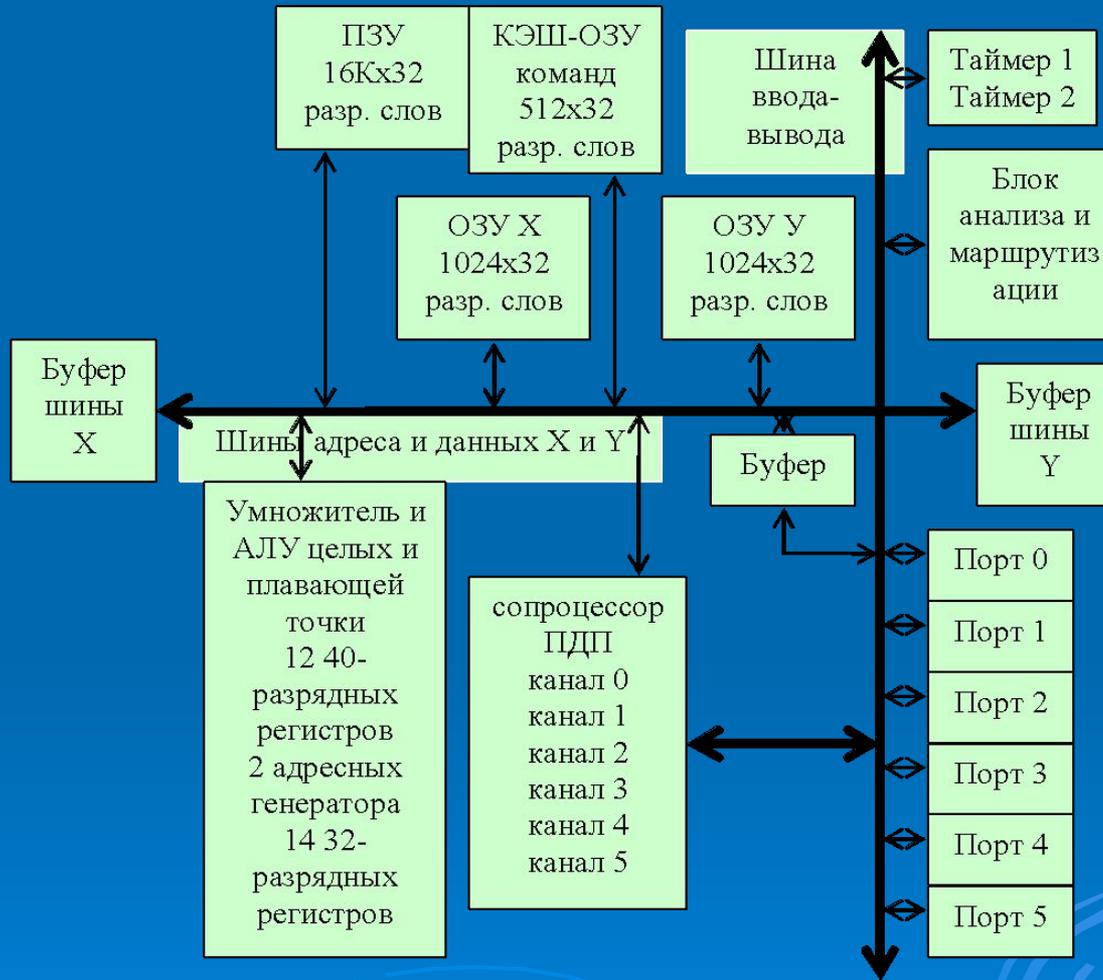


Рис. 7.2. Архитектура TMS320C40

Цифровые сигнальные процессоры.

На рис. 7.3 показан вариант топологии связей процессоров.

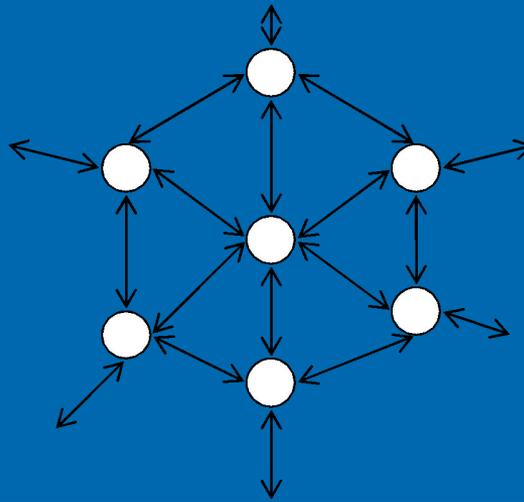


Рис. 7.3. Топология связей ЦСП

Цифровые сигнальные процессоры.

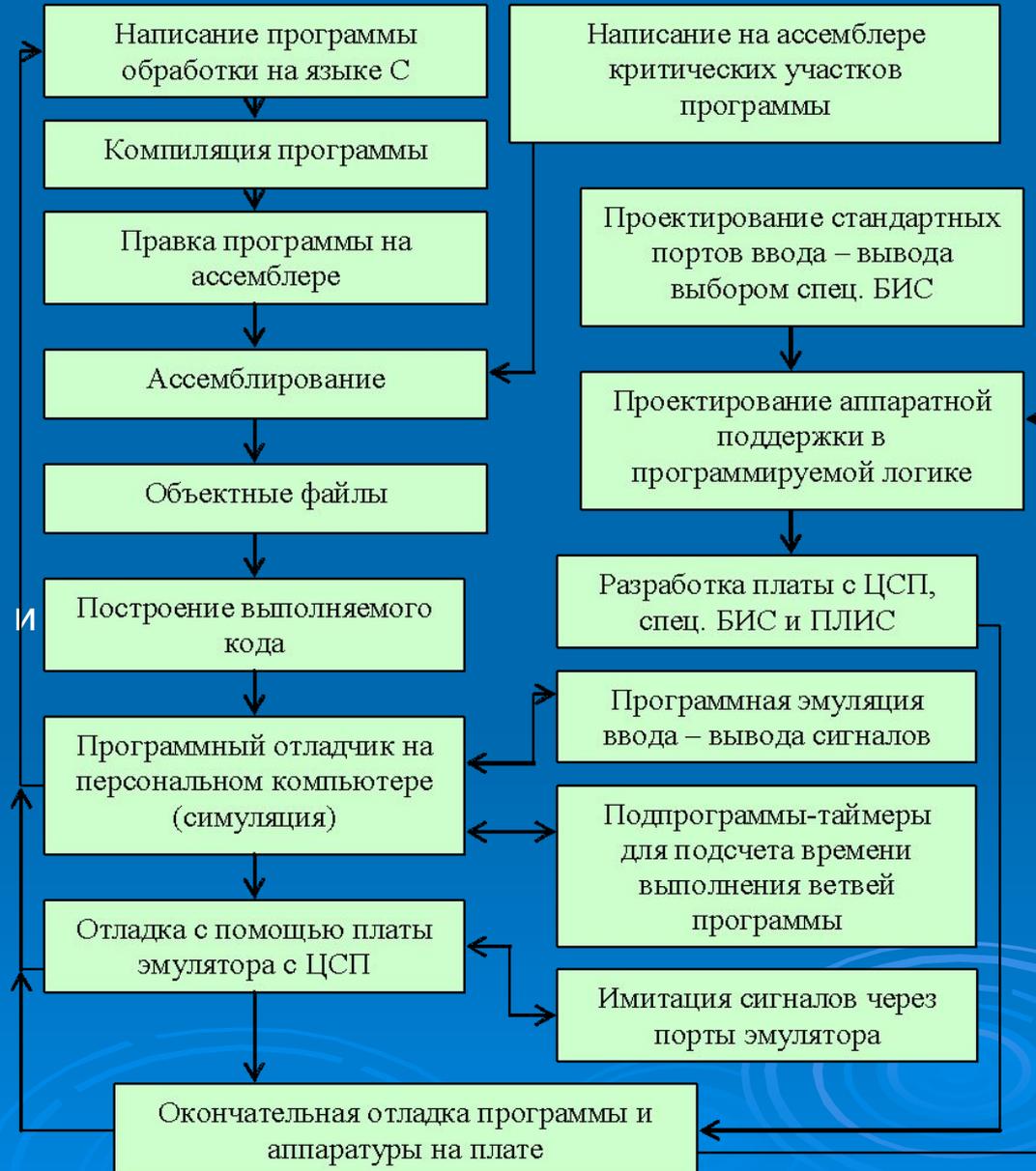


Рис. 7.4. Процесс разработки аппаратуры на ЦСП и ПЛИС