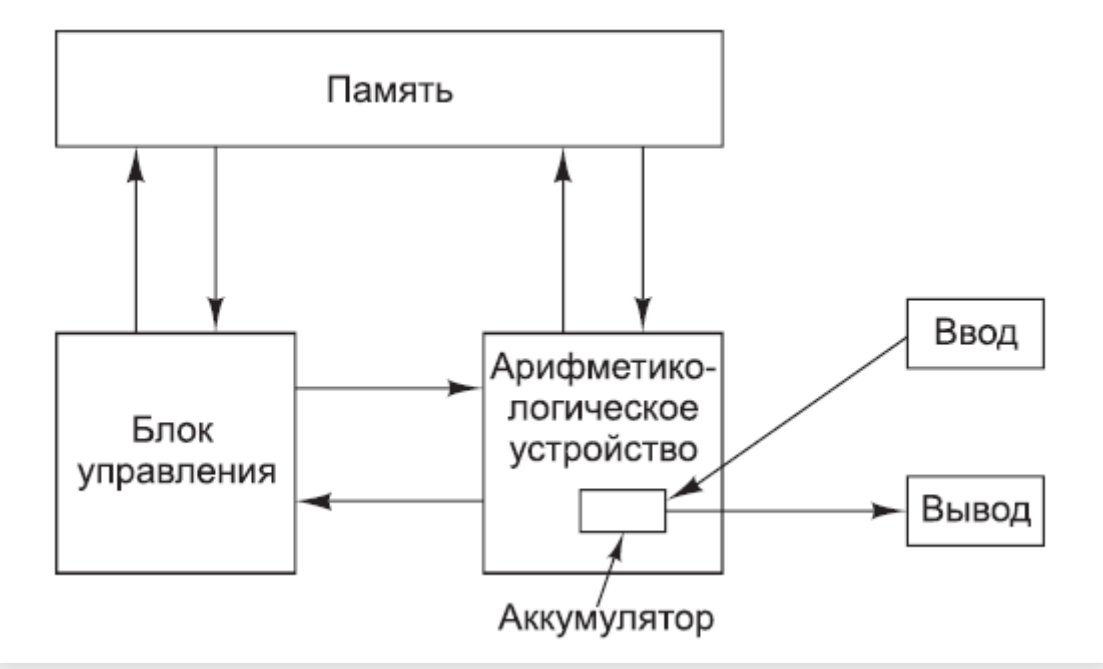
ПРОЦЕССОР

КЛАССИЧЕСКИЕ АРХИТЕКТУРЫ

АРХИТЕКТУРА ФОН-НЕЙМАНА



Speaker notes

[Во время работы над IAS] Фон Нейман заметил, что программирование компьютеров с большим количеством переключателей и кабелей — занятие медленное, утомительное и неудобное. Он пришел к мысли, что программа должна быть представлена в памяти компьютера в цифровой форме, вместе с данными. Он также отметил, что десятичная арифметика, используемая в машине ENIAC, где каждый разряд представлялся десятью электронными лампами (1 включена и 9 выключены), может быть заменена параллельной двоичной арифметикой.

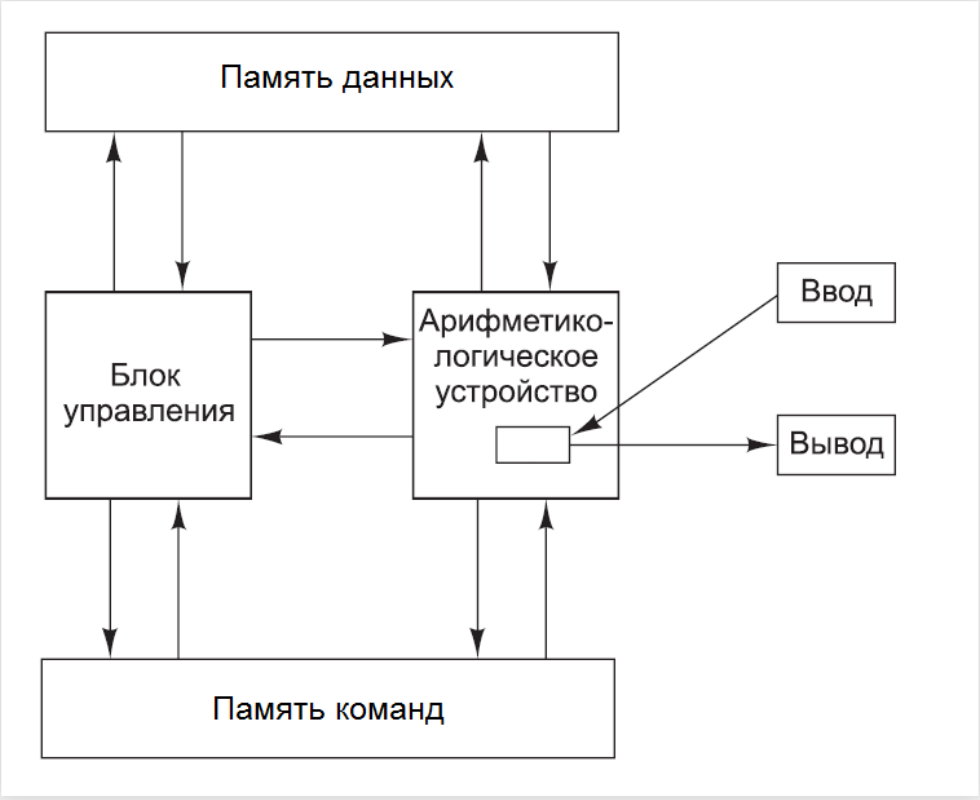
Арифметический блок и блок управления составляли «мозговой центр» компьютера. В современных машинах эти блоки сочетаются в одной микросхеме, называемой центральным процессором (ЦП). Внутри арифметико-логического устройства находился особый внутренний регистр на 40 бит, так называемый аккумулятор. Типичная команда прибавляла слово из памяти к аккумулятору или сохраняла содержимое аккумулятора в памяти.

В чистой архитектуре фон Неймана процессор в каждый момент времени может либо читать инструкцию, либо читать/записывать единицу данных из/в памяти. Оба действия одновременно происходить не могут, поскольку инструкции и данные используют один и тот же поток (шину).

Принципы:

- Однородность памяти: команды и данные хранятся в одной и той же памяти и внешне в памяти неразличимы.
- Адресность: структурно основная память состоит из пронумерованных ячеек, причем процессору в произвольный момент доступна любая ячейка.
- Программное управление: все вычисления, предусмотренные алгоритмом решения задачи, должны быть представлены в виде программы, состоящей из последовательности управляющих слов — команд.
- Двоичное кодирование: вся информация, как данные, так и команды, кодируются цифрами 1 и 0.

ГАРВАРДСКАЯ АРХИТЕКТУРА



Speaker notes

В компьютере с использованием гарвардской архитектуры процессор может считывать очередную команду и оперировать памятью данных одновременно и без использования кэш-памяти. Таким образом, компьютер с гарвардской архитектурой при определенной сложности схемы быстрее, чем компьютер с архитектурой фон Неймана, поскольку потоки команд и данных расположены на раздельных физически не связанных между собой аппаратных каналах.

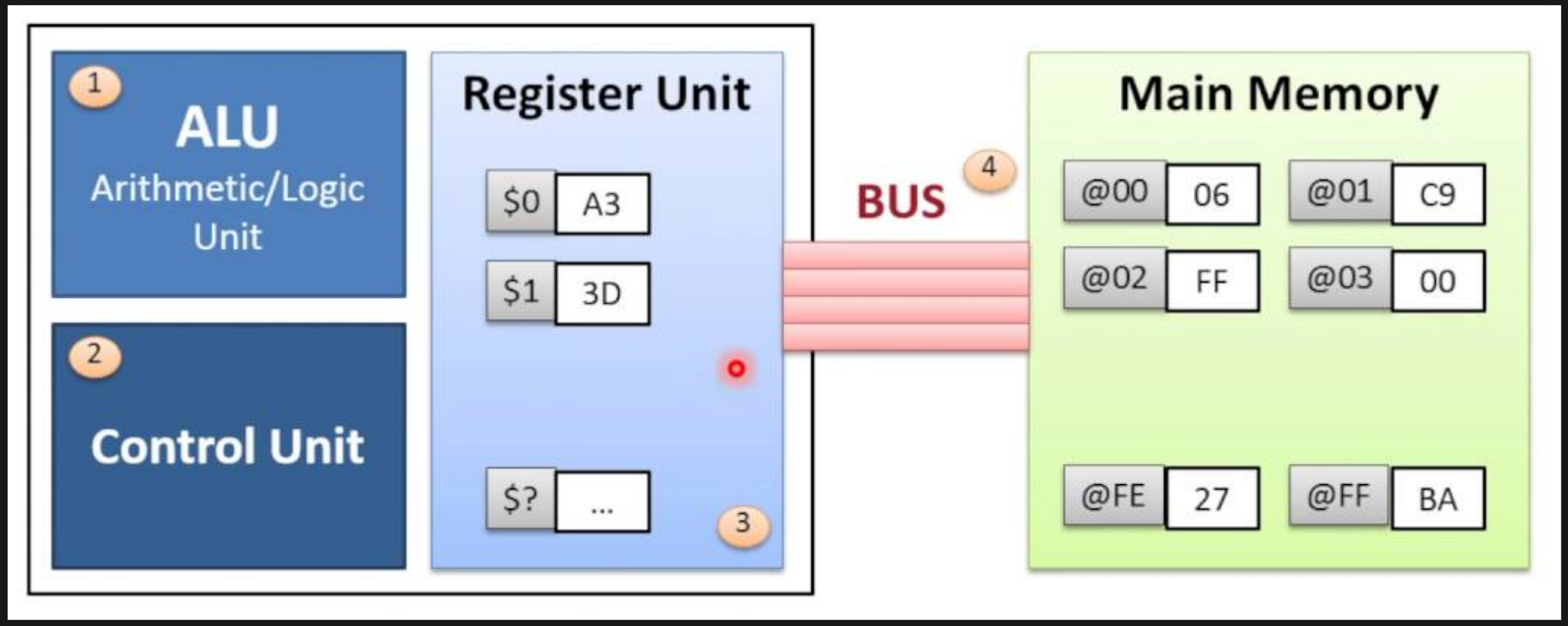
Принципы:

- Неоднородность памяти: команды и данные хранятся раздельно, с асинхронным доступом к ним.
- Адресность.
- Программное управление.
- Двоичное кодирование.

Speaker notes

При конвейерной организации обращения и к командам, и к данным (операндам) должны осуществляться одновременно. Разделенная кэш-память позволяет осуществлять параллельный доступ, а общая — нет. К тому же, поскольку команды обычно не меняются во время выполнения программы, содержание кэша команд не приходится записывать обратно в основную память.

КОМПОНЕНТЫ ПРОЦЕССОРА


1
ALU
Arithmetic/Logic Unit
2
Control Unit
Register Unit
$0
A3
$1
3D
$?
...
3
BUS
4
Main Memory
@00
06
@01
C9
@02
FF
@03
00
@FE
27
@FF
BA

- Устройства ввода-вывода (УВВ) — передают данные с/на шину.
- Устройство управления (УУ) — выбирает команды для выполнения с УВВ, передает арифметику и работу с данными на остальные компоненты.
- Арифметико-логическое устройство (АЛУ).
- Запоминающее устройство (ЗУ) — регистры и кэш-память.

АЛУ

Арифметико-логическое устройство (**arithmetic and logic unit, ALU**) — блок процессора для выполнения простых арифметических и логических преобразований над данными.

Входные данные — операнды.

АЛУ

- **FPU (floating point unit)** — блок выполнения операций с плавающей запятой.
- **AGU (address generation unit)** — блок вычисления адресов в памяти.

ПАМЯТЬ

ПАМЯТЬ

Регистр процессора — сверхбыстрая оперативная память внутри процессора.

Кэш-память — очень быстрая память небольшого размера, используется для уменьшения среднего времени доступа к оперативной памяти.

АРХИТЕКТУРЫ CISC И RISC

CISC

Complex Instruction Set Computer

- Много команд на все случаи жизни.
- Команды для сложных операций на несколько тактов.

Минусы:

- Большое количество сложных команд отрицательно влияет на производительность.

RISC

Reduced Instruction Set Computer

- Минимальный набор простых команд.
- Высокая скорость выполнения.

Минусы:

- Большое количество запускаемых команд (1 команда CISC ~= 4-5 команд RISC).

ПРИНЦИПЫ RISC

- Все команды выполняются непосредственно аппаратным обеспечением.
- Запуск как можно большего количества команд в секунду.
- Команды должны легко декодироваться.
- К памяти должны обращаться только команды загрузки и сохранения.
- Регистров должно быть много.

Speaker notes

- Все команды должны выполняться непосредственно аппаратным обеспечением.

То есть обычные команды выполняются напрямую, без интерпретации микрокомандами. Устранение уровня интерпретации повышает скорость выполнения большинства команд. В компьютерах типа CISC более сложные команды могут разбиваться на несколько шагов, которые затем выполняются как последовательность микрокоманд. Эта дополнительная операция снижает быстродействие машины, но может использоваться для редко применяемых команд.

- Запуск как можно большего количества команд в секунду.

При этом, система должна выдерживать выполнение этих команд после запуска. Это достигается параллелизмом.

- Команды должны легко декодироваться

Предел количества запускаемых в секунду команд зависит от темпа декодирования отдельных команд. Декодирование команд позволяет определить, какие ресурсы им необходимы и какие действия нужно выполнить. Все, что способствует упрощению этого процесса, полезно. Например, можно использовать единообразные команды с фиксированной длиной и с небольшим количеством полей. Чем меньше разных форматов команд, тем лучше.

- К памяти должны обращаться только команды загрузки и сохранения

Один из самых простых способов разбить операцию на отдельные шаги — сделать так, чтобы операнды большей части команд брались из регистров и возвращались туда же. Операция перемещения операндов из памяти в регистры и обратно может осуществляться в разных командах. Поскольку доступ к памяти занимает много времени, длительность которой невозможно спрогнозировать, выполнение этих команд могут взять на себя другие команды, единственное назначение которых — перемещение операндов между регистрами и памятью. То есть к памяти должны обращаться только команды загрузки и сохранения

- Регистров должно быть много

Так как запускается много команд на выполнение, их нужно где-то хранить.

ГИБРИДНАЯ АРХИТЕКТУРА.

“Поздние” x86 процессоры (Intel Pentium 4, D, AMD Athlon, Phenom) — CISC-совместимы, но являются процессорами с RISC-ядром.

CISC-инструкции преобразуются в набор внутренних RISC-команд.

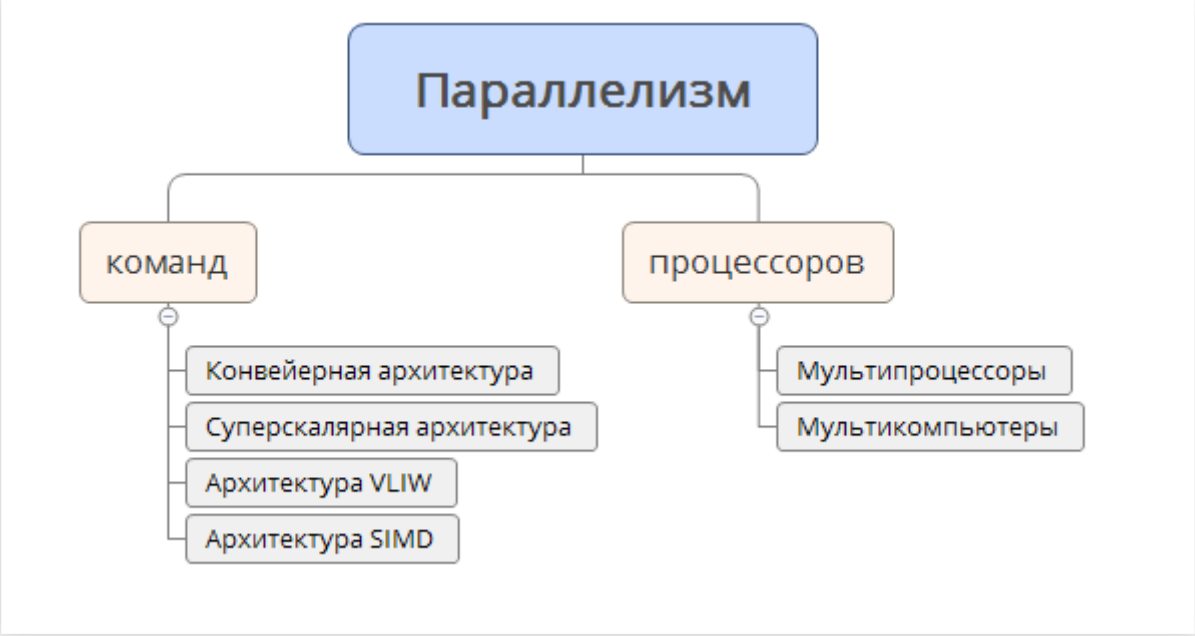
АЛГОРИТМ РАБОТЫ ПРОЦЕССОРА

АЛГОРИТМ РАБОТЫ ПРОЦЕССОРА

Speaker notes

1. Выборка команды из кэша или оперативной памяти в регистр команд
2. Дешифрация команды
3. Выборка операндов из кэша или оперативной памяти в регистры
4. Исполнение операции
5. Запись результата в регистры, кэш или оперативную память

ПАРАЛЛЕЛИЗМ



Speaker notes

Чтобы ускорить выполнение задач, можно делить их на фрагменты и выполнять параллельно.

Параллелизм на уровне команд происходит в одном процесоре, на уровне процессоров − в разных.

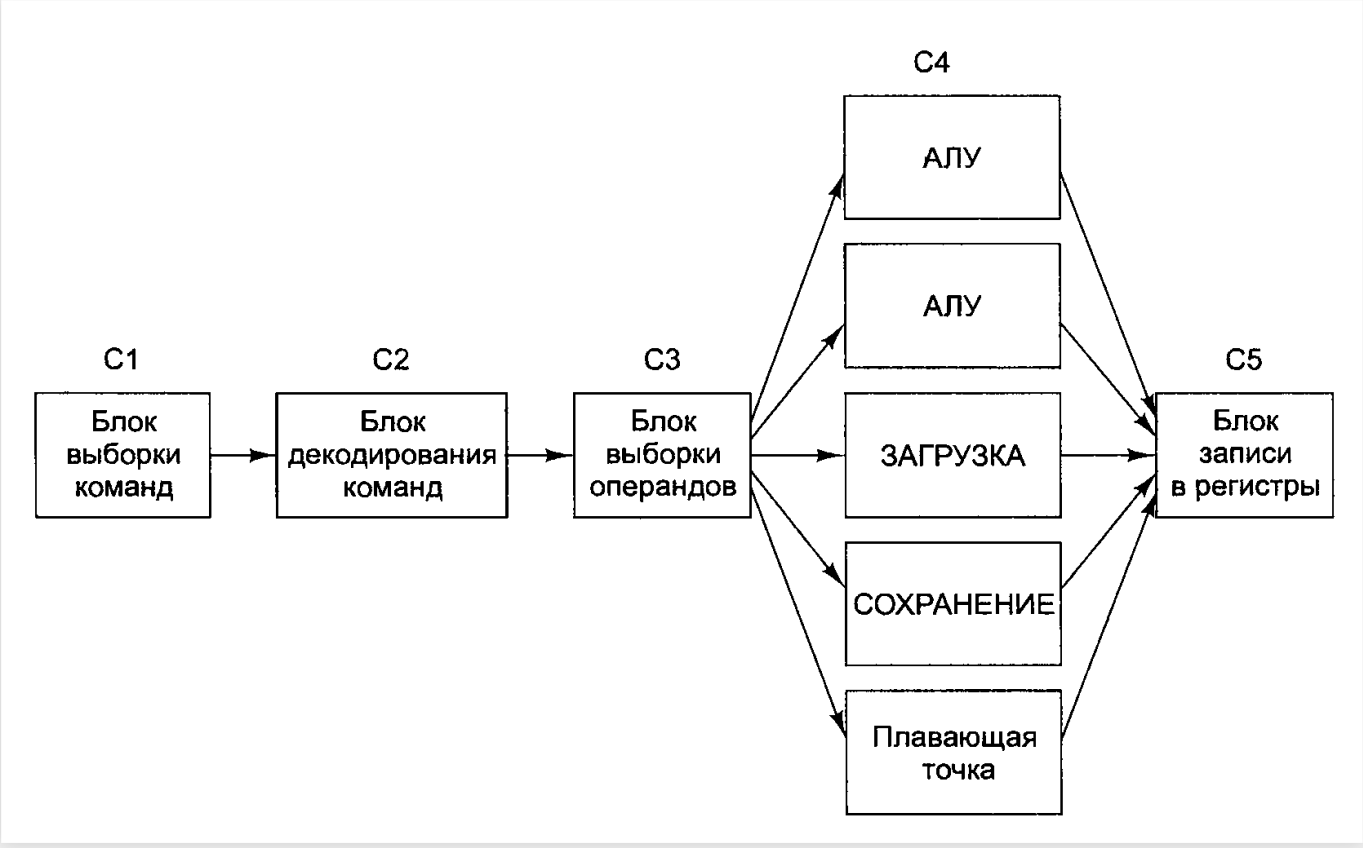
КОНВЕЙЕРНАЯ АРХИТЕКТУРА

Speaker notes

Управляющее устройство состоит из нескольких блоков/ступеней. Каждый блок занимается своим типом задач. Когда блок выполнил задачу, он передает результаты следующему. Таким образом, управляющее устройство за один такт может совершать несколько действий.

Первая ступень (блок C1) вызывает команду из памяти и помещает ее в буфер, где она хранится до тех пор, пока не потребуется. Вторая ступень (блок C2) декодирует эту команду, определяя ее тип и тип ее операндов. Третья ступень (блок C3) определяет местонахождение операндов и вызывает их из регистров или из памяти. Четвертая ступень (блок C4) выполняет команду, обычно проводя операнды через тракт данных. И наконец, блок C5 записывает результат обратно в нужный регистр.

СУПЕРСКАЛЯРНАЯ АРХИТЕКТУРА

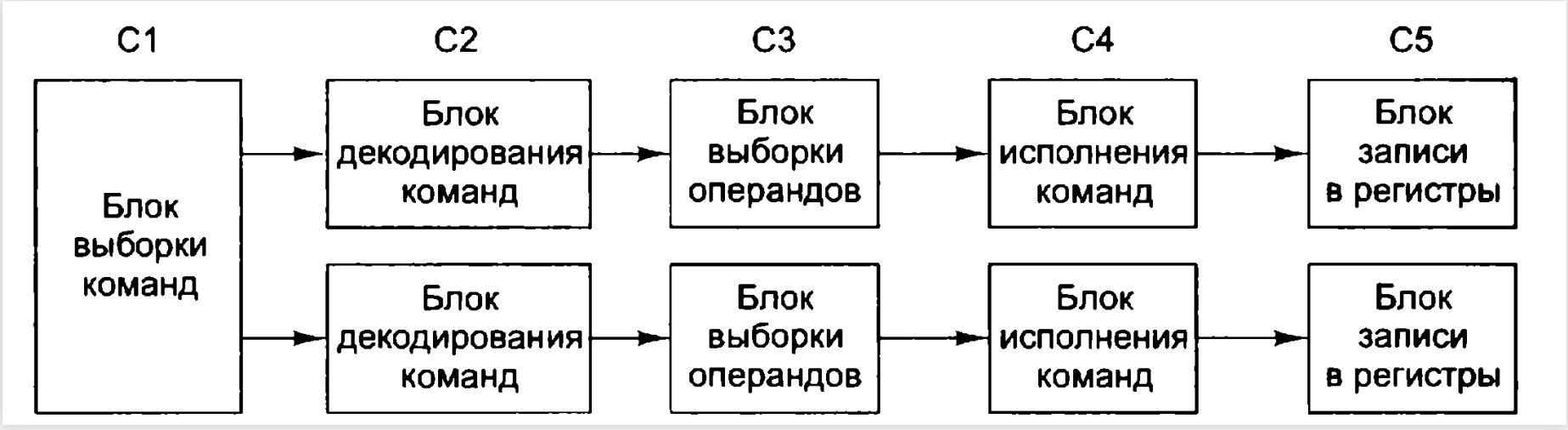


Speaker notes

Архитектура, в которой на одной ступени находятся несколько функциональных блоков.

Позволяет реализовать параллелизм на уровне инструкций — в течение одного такта может выполняться несколько инструкций.

2 КОНВЕЙЕРА


С1
С2
С3
С4
С5
Блок выборки команд
Блок декодирования команд
Блок выборки операндов
Блок исполнения команд
Блок записи в регистры
Блок декодирования команд
Блок выборки операндов
Блок исполнения команд
Блок записи в регистры

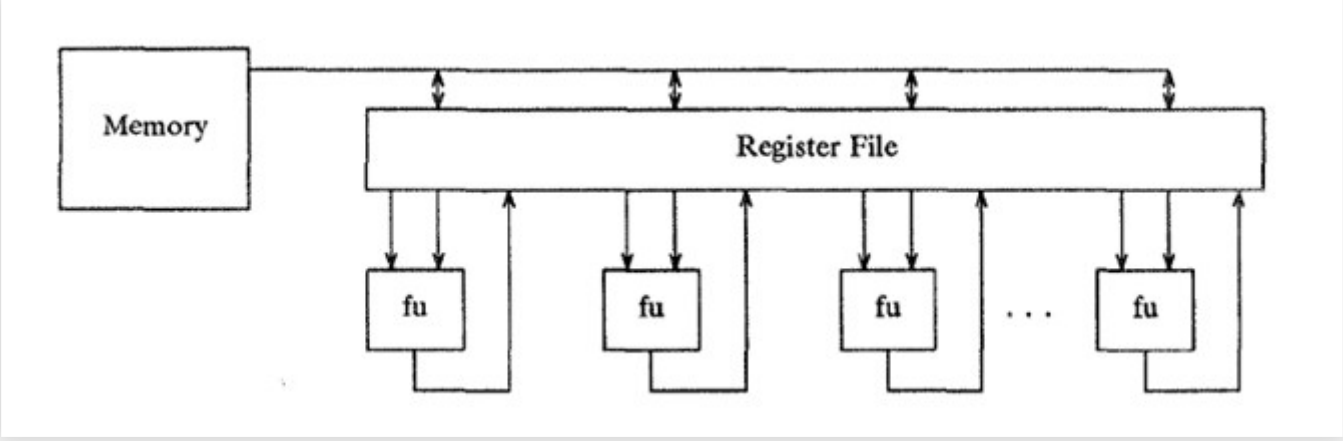
Speaker notes

Для дополнительного ускорения можно добавить еще один конвейер (Или ещё 30, как в Pentium 4 Cedar Mill). Конвейеры могут быть общего назначения или работать только с определенным типом команд, например − числа с плавающей точкой.

VLIW

VLIW (very long instruction word) — архитектура процессоров с несколькими АЛУ.

Одна команда содержит несколько операций, которые должны выполняться параллельно.



Speaker notes

В процессорах VLIW задача распределения решается во время компиляции и в инструкциях явно указано, какое АЛУ должно выполнять какую команду. Минусы – сложность разработки компилятора, большое количество кода, простой блоков процессора при блокировке одного из АЛУ

ТАКСОНОМИЯ ФЛИННА

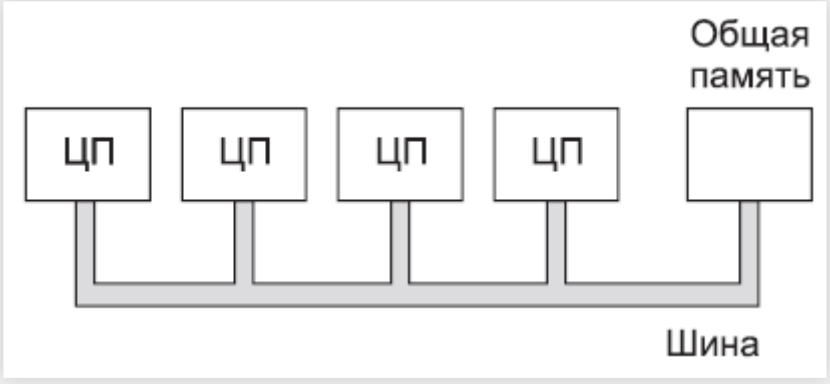
	Одиночный поток команд (Single Instruction)	Множество потоков команд (Multiple Instruction)
Одиночный поток данных (Single Data)	**SISD**	**MISD**
Множество потоков данных (Multiple Data)	**SIMD**	**MIMD**

ТАКСОНОМИЯ ФЛИННА

- **SISD** — обычная Фон-Неймановская машина без параллелизма.
- **SIMD** — одни команды на всех процессорах, разные данные.
- **MISD** — одни данные, разные команды для каждого процессора.
- **MIMD** — команды и данные для всех процессоров разные.

МУЛЬТИПРОЦЕССОРЫ

Система из нескольких параллельных процессоров, имеющих общую память. Такие процессоры называются сильно связанными

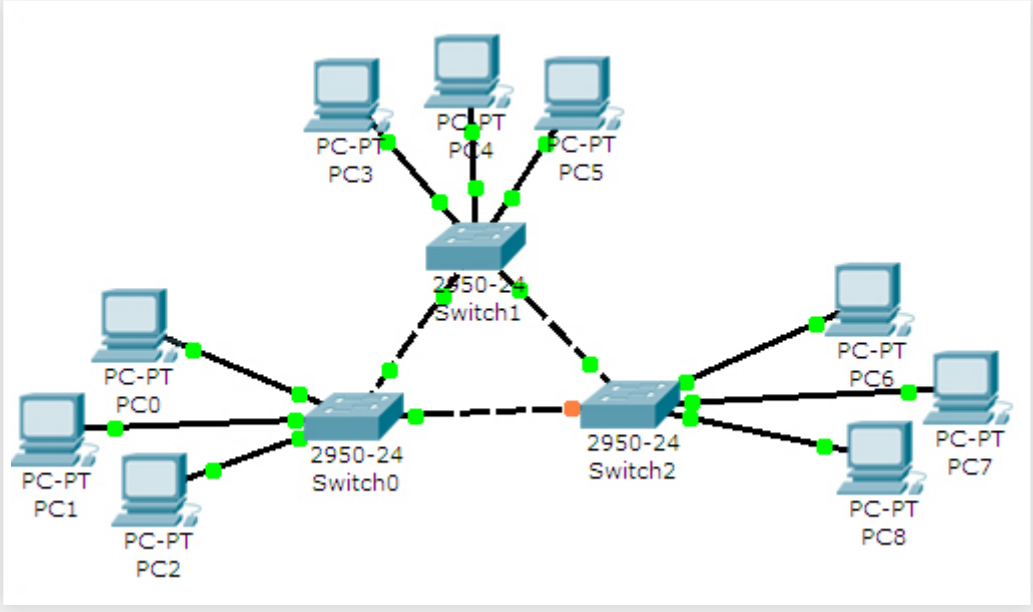


Speaker notes

Поскольку каждый процессор может записывать информацию в любую часть памяти и считывать информацию из любой части памяти, чтобы не допустить каких-либо нестыковок, их работа должна согласовываться программным обеспечением. Естественно, при наличии большого числа быстродействующих процессоров, которые постоянно пытаются получить доступ к памяти через одну и ту же шину, будут возникать конфликты. Чтобы разрешить эту проблему и повысить производительность компьютера, разработаны различные схемы. Например, каждый процессор имеет собственную локальную память, недоступную для других процессоров. Эта память используется для тех программ и данных, которые не нужно разделять между несколькими процессорами. При обращении к локальной памяти основная шина не используется, и, таким образом, объем передаваемой по ней информации становится меньше. Мультипроцессоры имеют преимущество перед другими видами параллельных компьютеров, поскольку с единой общей памятью легче и быстрее работать.

МУЛЬТИКОМПЬЮТЕРЫ (КЛАСТЕРЫ)

Системы из большого числа взаимосвязанных компьютеров, у каждого из которых есть собственная память.



Speaker notes

Связать больше 256 процессоров с общей памятью − затруднительно. Многие разработчики отказались от идеи разделения памяти и стали создавать системы без общей памяти, состоящие из большого числа взаимосвязанных компьютеров, у каждого из которых имеется собственная память. Такие системы называются мультикомпьютерами. В них процессоры являются слабо связанными, в противоположность сильно связанным процессорам в мультипроцессорных системах. Процессоры мультикомпьютера отправляют друг другу сообщения. Каждый компьютер не обязательно соединять со всеми другими, поэтому обычно в качестве топологий используются двух- и трехмерные решетки, а также деревья и кольца. Хотя на пути до места назначения сообщения проходят через один или несколько промежуточных компьютеров, время передачи занимает всего несколько микросекунд.

Процессоры в таких системах являются слабо связанными (Loosely coupled).

МНОГОПОТОЧНОСТЬ

Способность процессора выполнять одновременно несколько потоков выполнения.

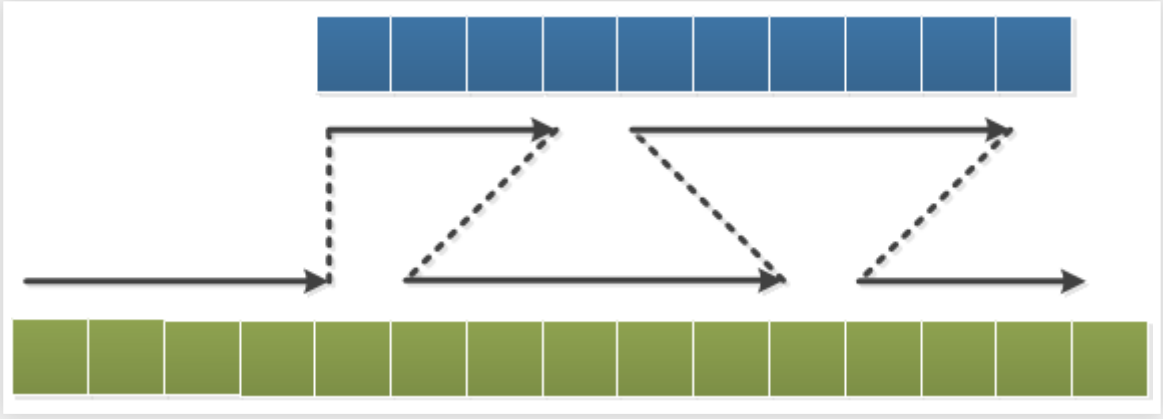
Поток выполнения — наименьший набор команд, который может выполняться (планироваться) независимо от других.

МНОГОПОТОЧНОСТЬ

- Временная
- Одновременная
- Кластерная

ВРЕМЕННАЯ МНОГОПОТОЧНОСТЬ

Крупнозернистая (Coarse-grained):

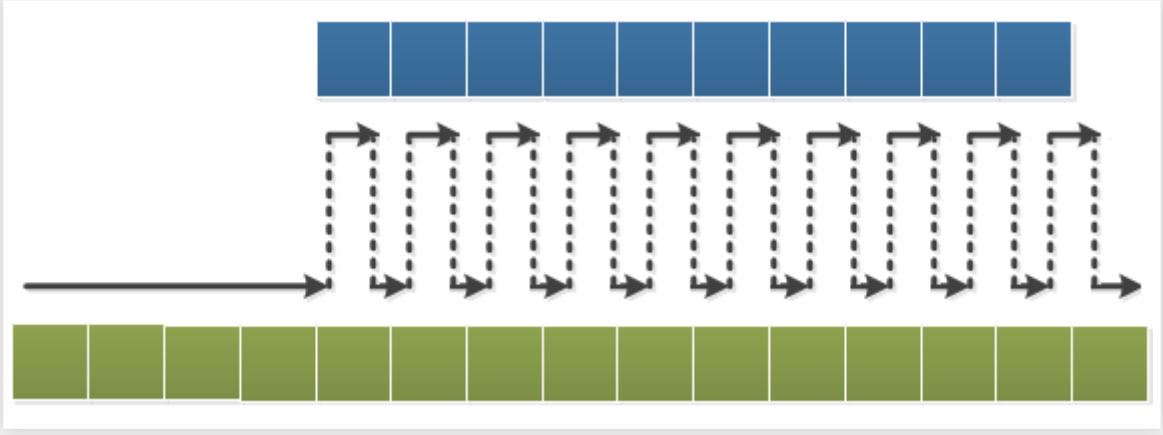


Speaker notes

В конвейере процессора исполняется только один поток некоторый промежуток времени. Переключение происходит если поток блокируется (конвейер встает) или кончается отведенное для потока время.

ВРЕМЕННАЯ МНОГОПОТОЧНОСТЬ

Мелкозернистая (Fine-grained):



Speaker notes

На каждом такте на вход конвейера подается команда из следующего потока. Таким образом, процессор гарантированно выполняет по одной команде N потоков каждые N циклов.

ОДНОВРЕМЕННАЯ МНОГОПОТОЧНОСТЬ

Используется в процессорах с суперскалярной архитектурой. На этапе выполнения команд могут выполняться команды относящиеся к разным потокам.