

## Reduced Instruction Set Computers – компьютер с сокращенным набором команд.

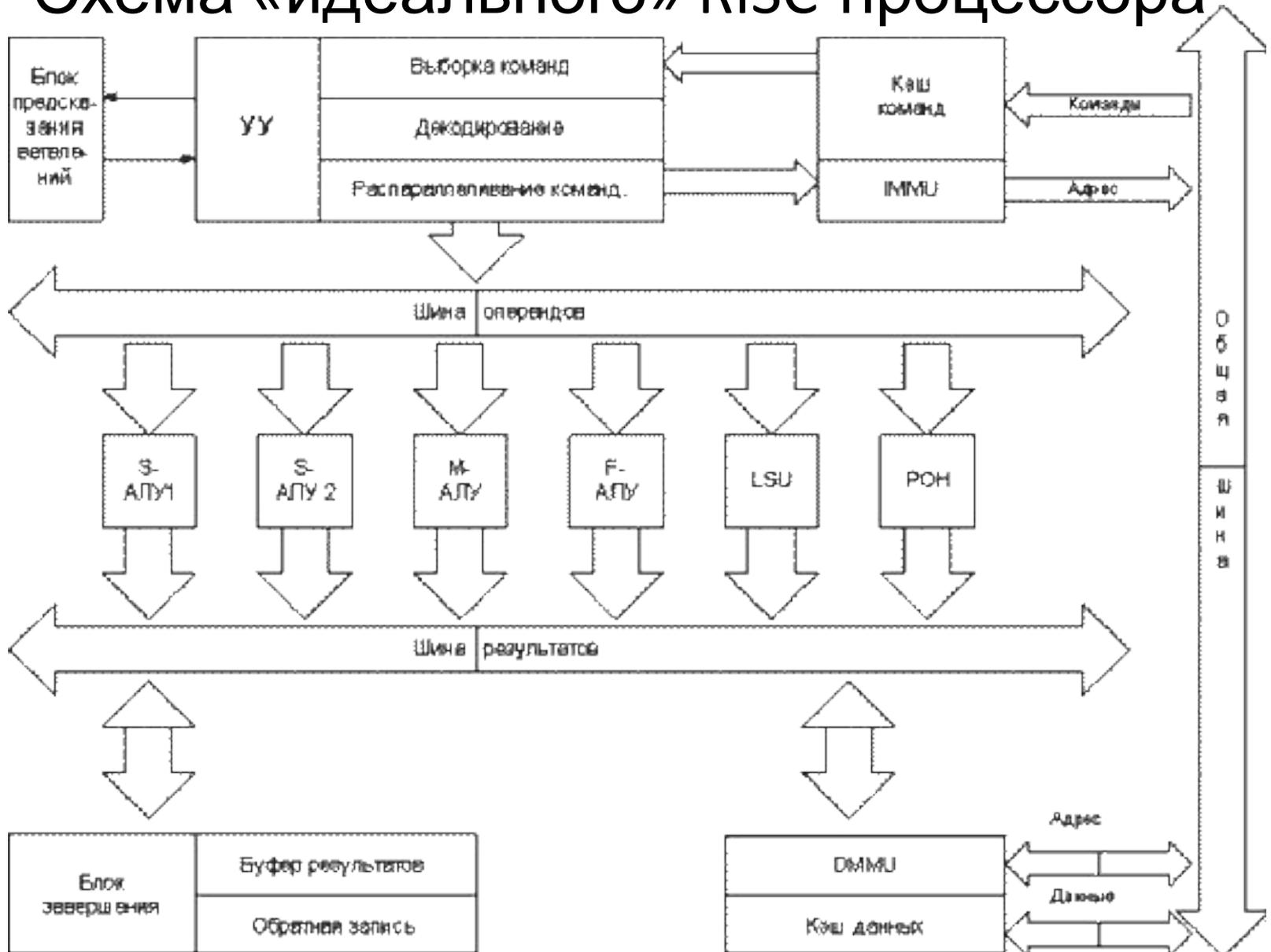
RISC-архитектура предполагает реализацию в ЭВМ сокращенного набора команд, что позволяет упростить аппаратуру процессора и повысить его производительность.

Набор команд включает простейшие и наиболее часто употребляемые команды, на выполнение которых затрачивается меньше времени. Выполнение более сложных, но редко встречающихся операций обеспечивают подпрограммы. Большинство команд в RISC-процессорах являются быстрыми командами типа «Регистр-регистр», которые выполняются без обращения к ОЗУ. Такие процессоры имеют не более 100 команд, 2-3 простых способа адресации, что ведет к упрощению устройства управления, которое строится по схеме с жесткой логикой. При этом структура процессора настолько упрощается, что становится возможной его реализация на одном кристалле, с оставлением места для увеличения до нескольких сотен числа общих и специализированных регистров.

Большое число регистров позволяет до предела сократить число обращений к ОЗУ, за счет:

- 1) сохранения в регистрах промежуточных результатов;
- 2) передачи через регистры операндов из одних программ в другие;
- 3) отказом от передач на сохранение в ОЗУ содержимого регистров при прерываниях программ.

# Схема «идеального» RISC-процессора



Принципиально новое, что отличает RISC- от CISC-процессоров – это:

отсутствие аппаратного стека – все операнды хранятся в регистрах общего назначения (РОН);

отсутствие регистра – счетчика команд;

наличие конвейера, позволяющего за один такт процессора осуществлять несколько вычислений;

четкое разделение потоков команд (инструкций) и данных;

полное равноправие РОН;

Кроме этого, в RISC-процессорах имеется:

шина операндов и шина результатов (вместо одной внутренней шины микропроцессора в CISC-процессорах 2-го поколения);

наличие специального исполнительного устройства, загружающего данные из памяти в РОН (LSU);

разделение АЛУ на одноктактные (длительность операции в котором занимает один такт процессора) и многотактные (длительность операций в котором превышает один такт);

наличие нескольких исполняющих устройств в микропроцессоре.

К ним относятся:

Одноктактные целочисленные АЛУ (S-АЛУ).

Многотактные целочисленные АЛУ (M-АЛУ).

Устройства (АЛУ) с плавающей точкой (F-АЛУ).

наличие отдельной кэш памяти для инструкций и данных, а также инструментов сегментного/страничного разбиения памяти;

общение микропроцессора с шиной только через кэш-память;

усложнение УУ, которое, кроме классической предвыборки и дешифрации команд, выполняет и распараллеливание вычислений и направление инструкций на несколько исполняющих модулей;

появление основного элемента конвейера – блока предсказания ветвлений (BPU), позволяющего осуществлять распараллеливание вычислений;

наличие блока завершения, который подготавливает результаты к записи в оперативную память.

## RISC-процессоры 3-го поколения

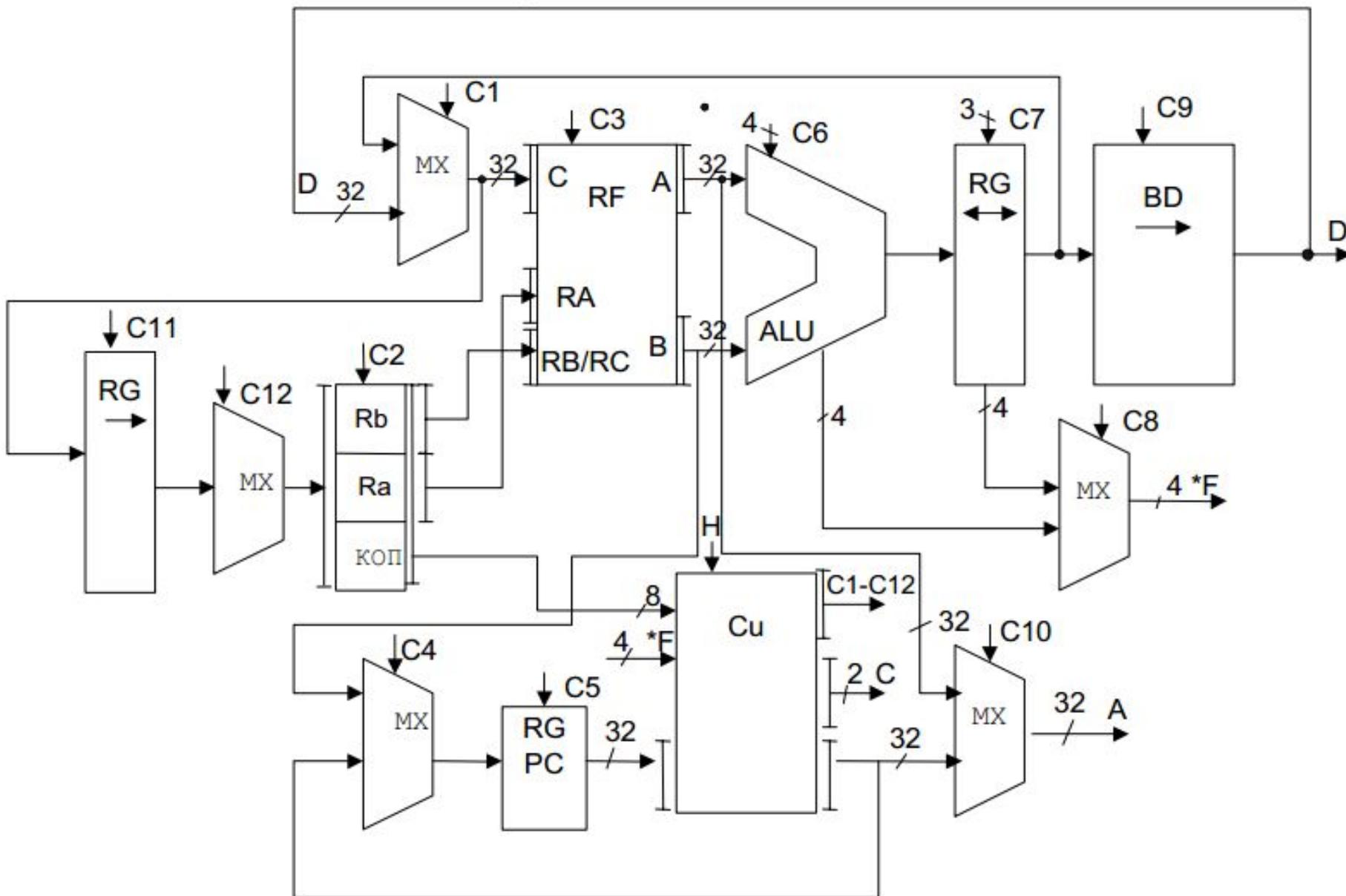
Самыми крупными разработчиками RISC-процессоров считаются Sun Microsystems (архитектура SPARC - Ultra SPARC), IBM (многокристальные процессоры Power, однокристальные PowerPC - PowerPC 620), Digital Equipment (Alpha - Alpha 21164), Mips Technologies (семейство Rxx00 -- R 10000), а также Hewlett-Packard (архитектура PA-RISC - PA-8000).

Все RISC-процессоры третьего поколения:

1. Являются 64-х разрядными и суперскалярными (запускаются не менее 4-х команд за такт);
2. Имеют встроенные конвейерные блоки арифметики с плавающей точкой;
3. Имеют многоуровневую кэш-память. Большинство RISC-процессоров кэшируют предварительно дешифрованные команды;
4. Изготавливаются по КМОП-технологии с 4 слоями металлизации.

# Функциональная схема RISC-процессоров

Функциональная схема



- RG является буферным сдвиговым регистром для ALU. Что означает буферный регистр? ALU является комбинационной схемой, поэтому результат, если будет снят операнд со входа, через некоторое время будет изменен, то есть не будет зафиксирован.
- Поэтому требуется его фиксировать в некотором регистре. мы обязательно должны зафиксировать этот результат в некотором буферном регистре, но попутно он еще выполняет роль сдвигателя. После этого мы подаем этот 32 разрядный результат опять в регистровый файл. То есть делаем так, что результат какой-то операции может быть записан только лишь в регистр. Но мы должны записывать в RF данные считываемые из внешней памяти. Для того чтобы из основной памяти переписать операнд хранящийся по какому-то адресу мы должны использовать тот же вход RF. Как следствие этого мы должны здесь поставить мультиплексор(Mx), который позволяет выбирать нам выход или с двигателя или же с основной памяти. Понятно что ALU формирует признаки результата и на вход его подается некие управляющие сигналы с блока управления. У нас 4 признака результата. Как правило ALU выполняет до 16 каких-то простых операций, поэтому в общем случае мы заведем 4 разряда которые будут определять ту операцию, которая будет выполняться ALU. Ясно, что регистр сдвига должен тоже управляться. Нам для этого потребуются 3 разряда. RF будет тоже управляться и на вход подаются два информационных сигнала: регистр

- Возможно формирование флагов в сдвиговом регистре, поэтому поставим мультиплексор который поможет получить нам как бы объединенные флаги. Получаем 4 разряда флагов, которые идут в УБ. Понятно, что этот мультиплексор управляется типом команды: командой сложения или сдвига. Если команда сложения, флаги идут с одного места, если команда сдвига, то с другого. На вход УБ с «жесткой» логикой работы подается КОП.
- Далее мы должны реализовать счетчик команд. Он будет реализован следующим образом: УБ будет модифицировать содержимое счетчика в зависимости от кода команды. Содержимое регистра счетчика команд – это РС считывается в УБ и он, в зависимости от того какая команда, или прибавляет 1 к РС или же заменяет его на содержимое регистра В. Поэтому впереди ставим мультиплексор на один из входов которого заводим выход регистрового файла В, а на другой вход выход устройства управления.
- Далее нам потребуются считывать текущую команду из памяти. Как только она появилась на выходе RF, мы должны ее подать на мультиплексор, образующий шину адреса процессора. Для того чтобы организовать двунаправленную передачу данных по одной и той же шине данных ставим шинный формирователь ВД и замкнем эти 2 шины. После этих дополнений шина данных получилась двунаправленная. В итоге получилась необходимость в наличии управляющих сигналов на каждый из перечисленных функциональных блоков. Эти все управляющие сигналы(С1-С10) вырабатываются устройством управления. Также устройством управления вырабатывается шина управления процессором, которая сообщает внешним запоминающим устройствам чтение или запись(один разряд на ЗУК а другой на ЗУД).

Современные RISC-процессоры характеризуются следующим:

- упрощенным набором команд;
- используются команды фиксированной длины и фиксированного формата,
- простые способы адресации, что позволяет упростить логику декодирования команд;
- большинство команд выполняются за один цикл процессора;
- логика выполнения команд с целью повышения производительности ориентирована на аппаратную, а не на микропрограммную реализацию, отсутствуют макрокоманды, усложняющие структуру процессора и уменьшающие скорость его работы;
- взаимодействие с оперативной памятью ограничивается операциями пересылки данных;
- для обработки, как правило, используются трехадресные команды, что помимо упрощения дешифрации дает возможность сохранять большее число переменных в регистрах без их последующей перезагрузки;
- создан конвейер команд, позволяющий обрабатывать несколько из них одновременно;
- наличие большого количества регистров;
- используется высокоскоростная память.

# Области применения RISC-процессоров

- Современные RISC-процессоры находят применение как:
- рабочие станции высшего ценового класса (12-15 тысяч долларов). Работают под ОС VMS, Unix;
- персональные рабочие станции (3-7 тыс. \$). ОС: Windows NT, Solaris;
- серверы;
- RISC ПК.