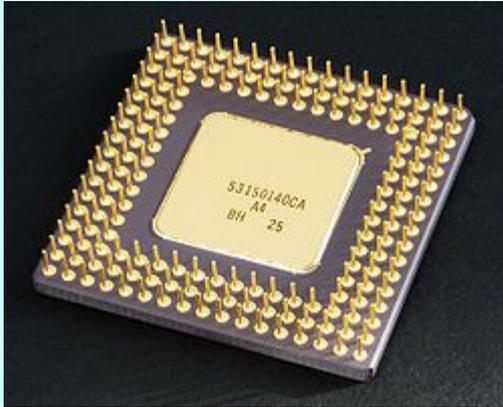


# Устройство памяти и процессора



# Память ЭВМ

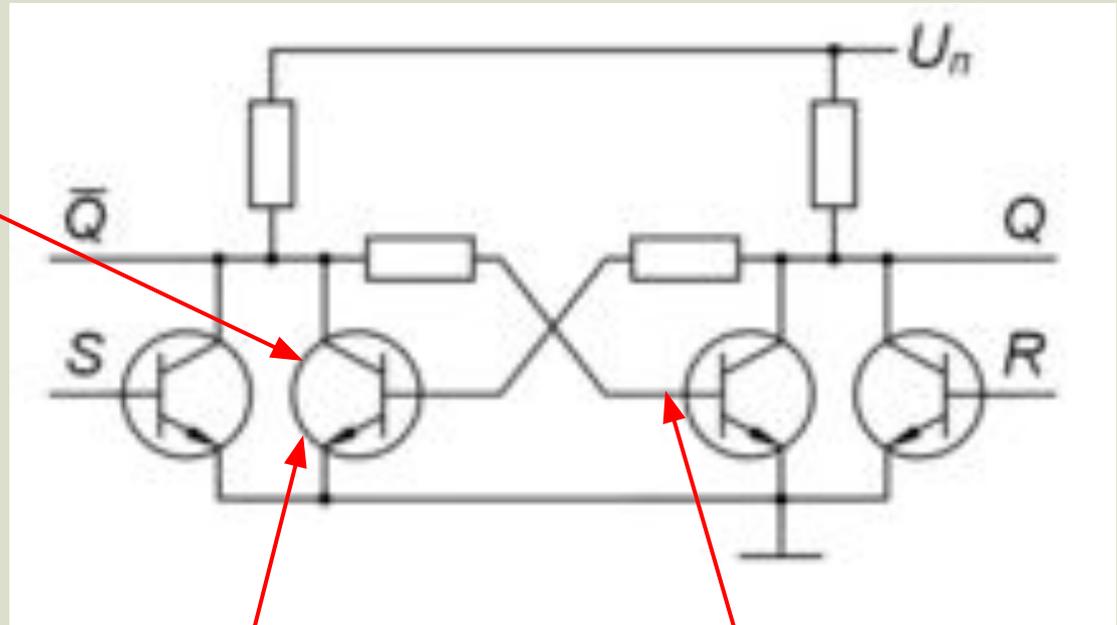
## Триггер

**Триггер** — устройство с двумя устойчивыми состояниями равновесия

Этот транзистор «открыт»

Подавая напряжение на входы  $S$  и  $R$  можно изменить состояние триггера

Переходные процессы в триггере происходят очень быстро



Через этот транзистор постоянно течет небольшой ток

Запирающий потенциал

# Память ЭВМ

## SRAM

**Статическая оперативная память с произвольным доступом**  
(*SRAM, static random access memory*)  
Выполнена из триггеров.



Быстрый доступ. Доступ к любой ячейке памяти в любой момент занимает одно и то же время.

Простая схемотехника — SRAM не требуются сложные контроллеры.



Высокое энергопотребление

Невысокая плотность записи (шесть элементов на бит).

# Память ЭВМ

## DRAM

### **DRAM (dynamic random access memory)**

DRAM-память представляет собой набор запоминающих ячеек, которые состоят из конденсаторов и транзисторов

Конденсаторы заряжают в случае, когда в ячейку записывается 1, и разряжают в случае, когда в ячейку необходимо записать 0.

Для поддержания необходимого напряжения на обкладках конденсаторов ячеек и сохранения их содержимого, их необходимо периодически подзаряжать



Высокая плотность записи

Небольшое энергопотребление



Относительно большое время доступа

Во время подзарядки память недоступна

# Устройство процессора

## Такт

**Такт** - промежуток времени, между последовательными сигналами синхронизации.

Величина такта выбирается такой, чтобы во время его прохождения в рассматриваемом объекте заканчивались все переходные процессы, вызванные изменением входных сигналов.

**Такт процессора** — промежуток между двумя импульсами тактового генератора, который синхронизирует выполнение всех операций процессора.

Выполнение различных элементарных операций может занимать от долей такта до многих тактов в зависимости от команды и процессора.



**Intel 80386**

**Тактовая частота**

12—40 МГц



**Intel Core i7**

**Тактовая частота**

2,66—3,46 ГГц

# Устройство процессора

## Регистры

### Регистры процессора

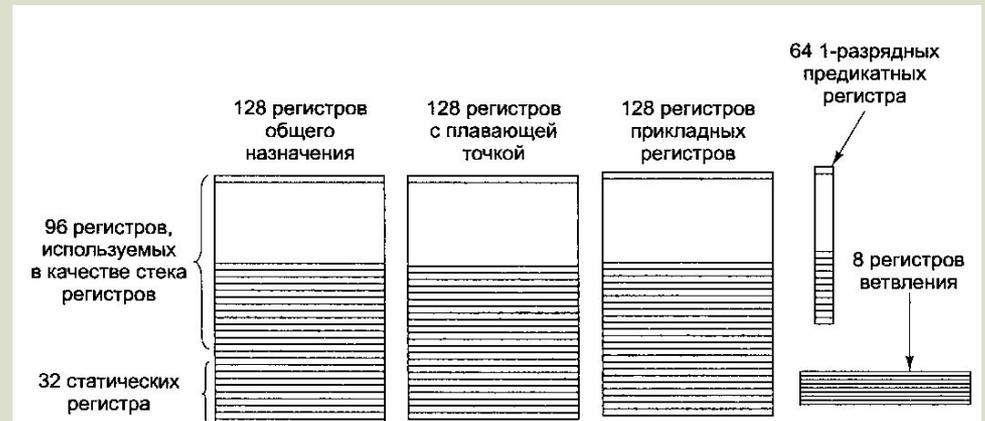
Регистры — ячейки памяти, «территориально» расположенные прямо в процессорном ядре.

Регистры собираются из триггеров.

### Регистры процессора Pentium

31	16	15	0	
		AH AX AL		EAX
		BH BX BL		EBX
		CH CX CL		ECX
		DH DX DL		EDX
		SI		ESI
		DI		EDI
		BP		EBP
		SP		ESP

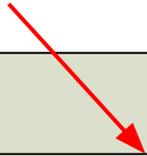
### Регистры процессора Itanium



# Устройство процессора

## Функциональные устройства

**ALU**- арифметико-логические устройства



- простые арифметические действия (сложение, вычитание, сравнение) с целыми числами
- логические операции («и», «или», «исключающее или» и «не»)
- копирование и простые преобразования чисел

**01110101+00000001= 01110110**

**FPU** - блоки вычислений с плавающей точкой

**SIMD (Single Instruction, Multiple Data )** - блоки векторной обработки

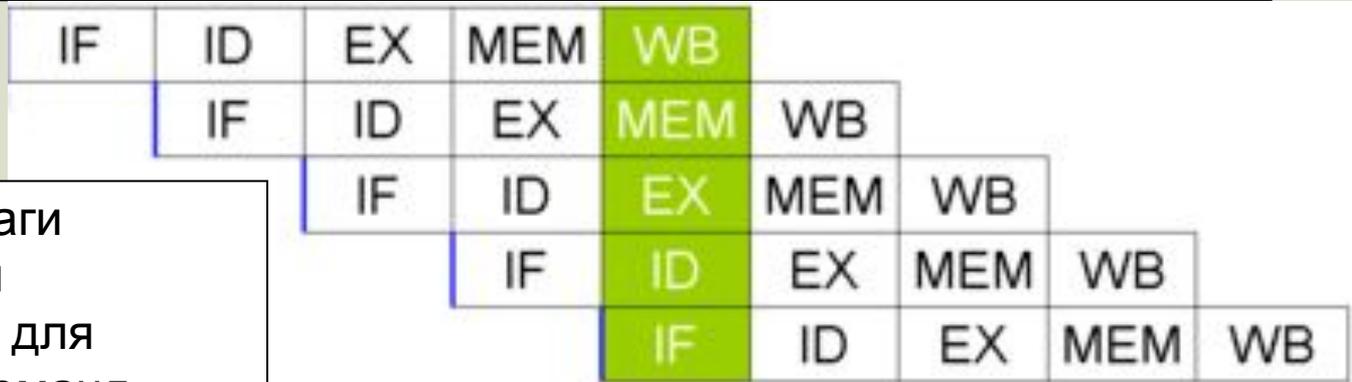
**Блоки обмена данных с памятью.**

# Устройство процессора

Конвейер

Обработка инструкции разделяется на последовательности независимых шагов, с сохранением результатов в конце каждого шага, например:

1. получение инструкции (*Instruction Fetch*)
2. декодирование инструкции (*Instruction Decode*) и чтение регистров
3. выполнение (*Execute*);
4. доступ к памяти (*Memory access*);
5. запись в регистр (*Register write back*)



Независимые шаги выполняются параллельно для нескольких команд

# Устройство процессора

Конвейер

## ПРОБЛЕМЫ:

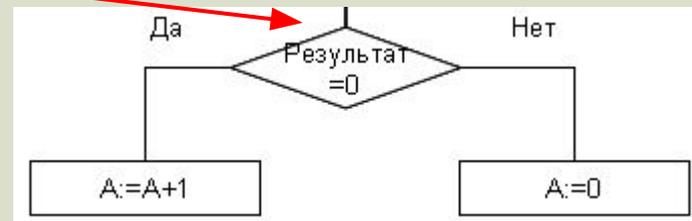
1. В процессорах используются инструкции самого разного рода, не всегда можно разделить инструкцию на независимые шаги

2. Инструкции могут зависеть друг от друга

1.  $A = B + C$   
2.  $K = A + M$

Нельзя выполнять инструкцию 2 пока не будет выполнена инструкция 1

3. При наличии условных переходов непонятно, какую инструкцию необходимо выполнять



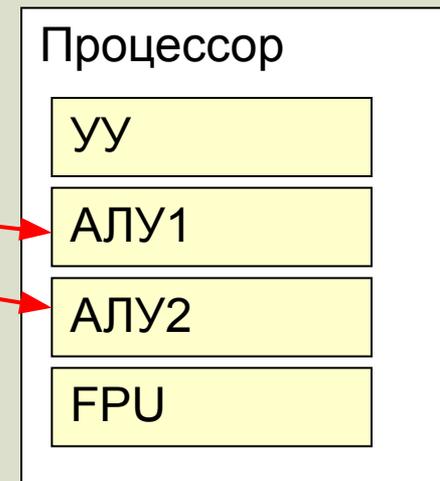
# Устройство процессора

## Суперскалярность

**Суперскалярность** — архитектура вычислительного ядра, использующая несколько декодеров команд, которые могут нагружать работой множество исполнительных блоков. Если в процессе работы команды не противоречат друг другу, и одна не зависит от результата другой, то такое устройство может осуществить параллельное выполнение команд.

(1) и (2) независимы и могут быть выполнены одновременно

1.  $A = B + C$
2.  $Z = X + Y$
3.  $K = A + Z$



# Устройство процессора

## Внеочередное исполнение

### Внеочередное исполнение

инструкции поступают в исполнительные модули не в порядке их следования, а по готовности к выполнению

1.  $A = B + C$
2.  $K = A + M$
3.  $Z = X + Y$

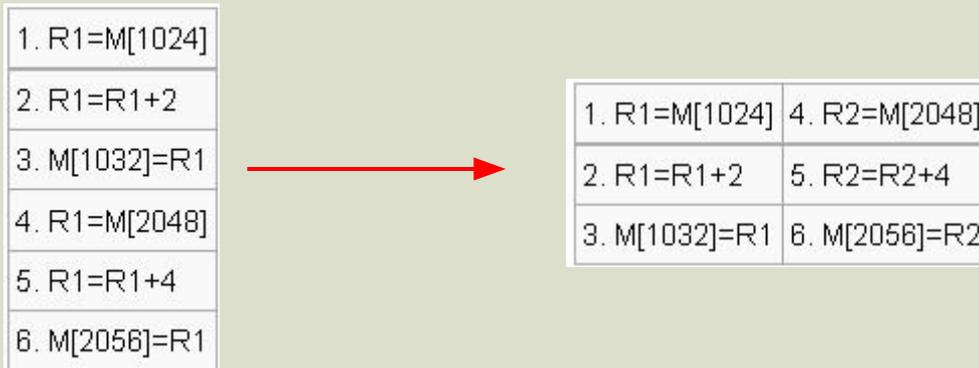
Последовательность выполнения будет изменена: сначала будут выполнены команды (1) и (3), а потом команда (2)

# Устройство процессора

## Переименование регистров

**Переименование регистров (*Register Renaming*)** — метод ослабления взаимозависимостей инструкций, применяемый в процессорах.

Переименование регистров представляет собой преобразование программных ссылок на архитектурные регистры в ссылки на физические регистры и позволяет ослабить влияние ложных взаимозависимостей за счёт использования большого количества физических регистров вместо ограниченного количества архитектурных



# Устройство процессора

## Предсказание переходов

### Предсказание переходов

**Статическое** - всегда выполнять или не выполнять определенные типы переходов.

**Динамическое** - анализируется таблица истории переходов, содержащая вероятность условного перехода: «скорее всего, будет выполнен», «возможно, будет выполнен», «возможно, не будет выполнен», «скорее всего, не будет выполнен» и обновляемая после каждого перехода.