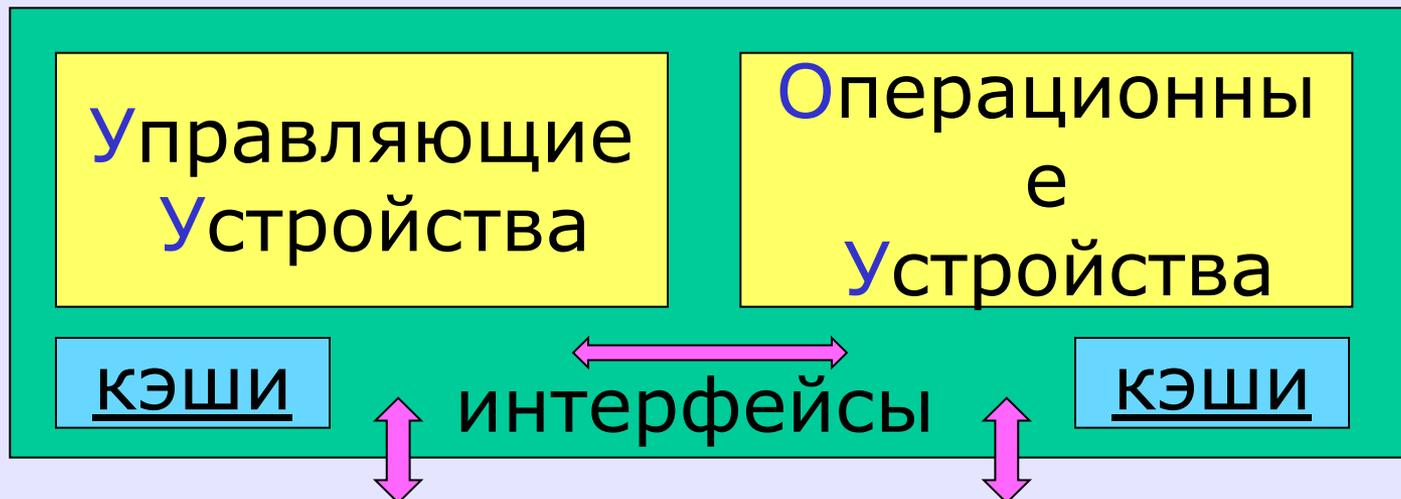
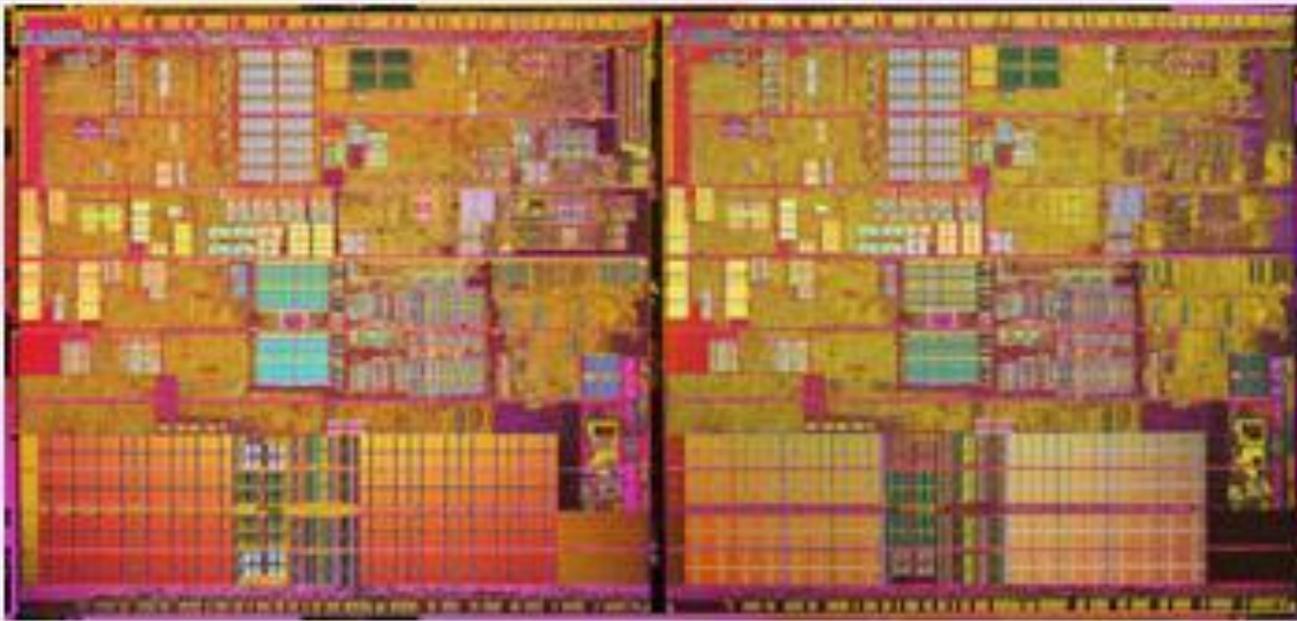


# Глава 3. Процессоры

## §1 Принципы работы

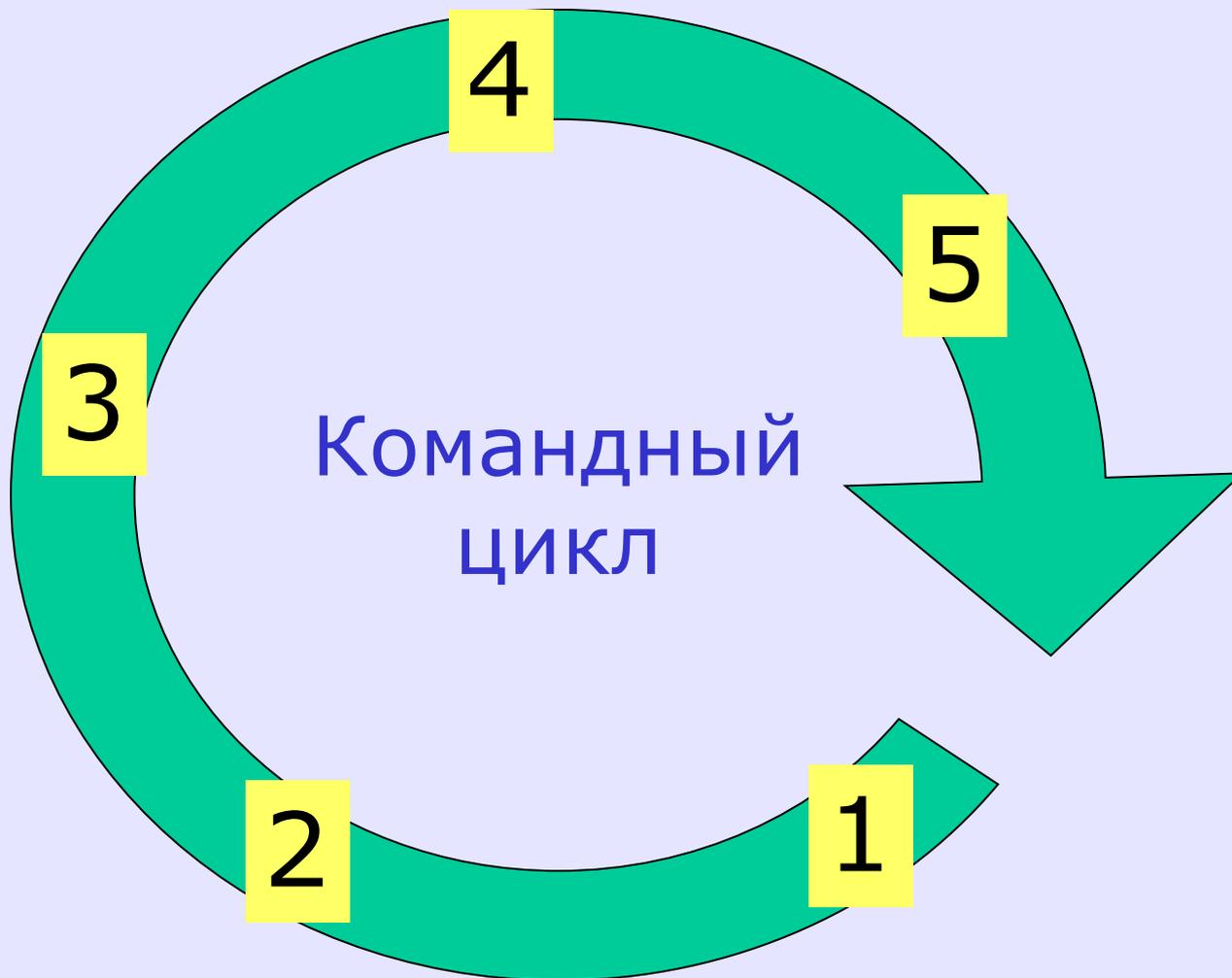
### 1. Процессор – устройство для обработки информации





- ОУ состоят из ОЭ, которые могут работать одновременно
- **Команда** (инструкция) – совокупность операций, н-р, сложение векторов

- Работа проц. циклична



- 1- извлечение команд из памяти,
- 2- их декодирование
- 3- извлечение данных,
- 4- выполнение,
- 5- запись результата

Исполняемая команда помещается в **регистр команд**

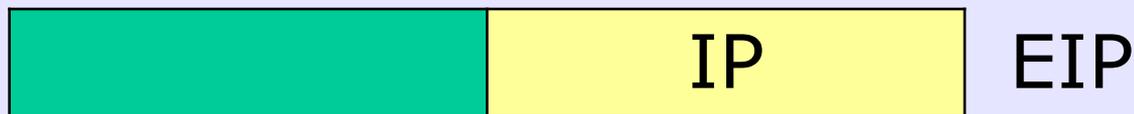
**Декодирование команды** – её разбиение на (микро) операции – раздача заданий для ОЭ

```
00    MAR := PC
01    MRd
02    CR := MDR
03 --> PC := PC+1
04    END_COMMAND
```

Адрес следующей команды  
хранится в регистре

«указатель инструкций»

(instruction pointer)



Как изменяется адрес:

- если текущая команда **линейная**, то прибавляется её длина (в байтах)
- если это команда **перехода** или **вызова** процедуры, то сама указывает новый адрес

Адреса данных и сами данные хранятся в регистрах общего назначения

Н-р, в архитектуре x86

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

## 2. Типы команд

- по типу ячеек (R-reg., M-mem.)
  - $R1, R2 \rightarrow R3$
  - $M1, M2 \rightarrow M3$
  - $R1 \rightarrow M1$
  - .....
- по их количеству
  - простые
  - сложные: несколько R и M

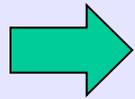
- по типу операндов
  - скалярные: число
  - векторные: массив чисел

Векторные команды дают экономию на 1,2 шагах цикла

### 3. «Одновременное» выполнение задач (программ)

Однопоточковый ЦП «создаёт  
иллюзию» одновременности

ОС вычисляет кванты  
времени для задачи согласно  
её **приоритету** (см. диспетчер задач)



ЦП по прерываниям от таймера периодически

- сохраняет в кэше содержимое регистров, доступных текущей задаче
- переключается на следующую задачу

# Многопоточковый процессор

имеет несколько

- декодеров команд
- регистров команд
- IP
- АЛУ



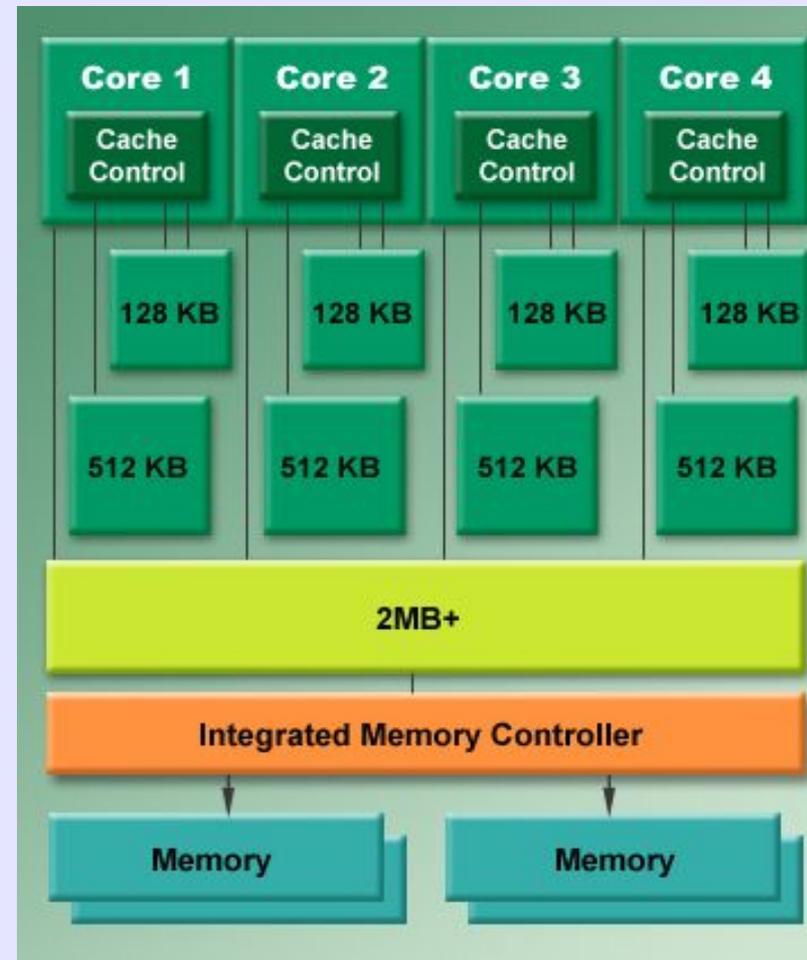
Много конвейеров  
и/или ядер



**Ядро** – часть процессора,  
выполняющая хотя бы 2-4  
этапы командного цикла

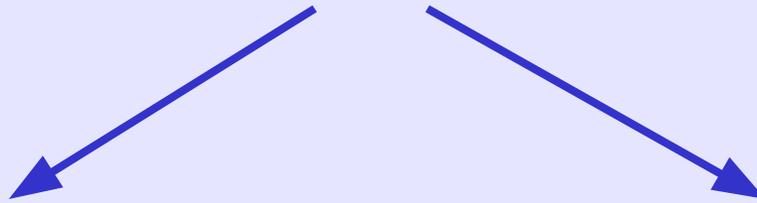
Общие

- кэш высокого уровня
- системная шина



Виртуальная машина – это  
программная среда,  
позволяющая запускать  
несколько ОС (одинаковых  
или разных)  
квазипараллельно

## 4. Пути повышения производительности

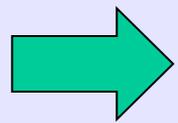
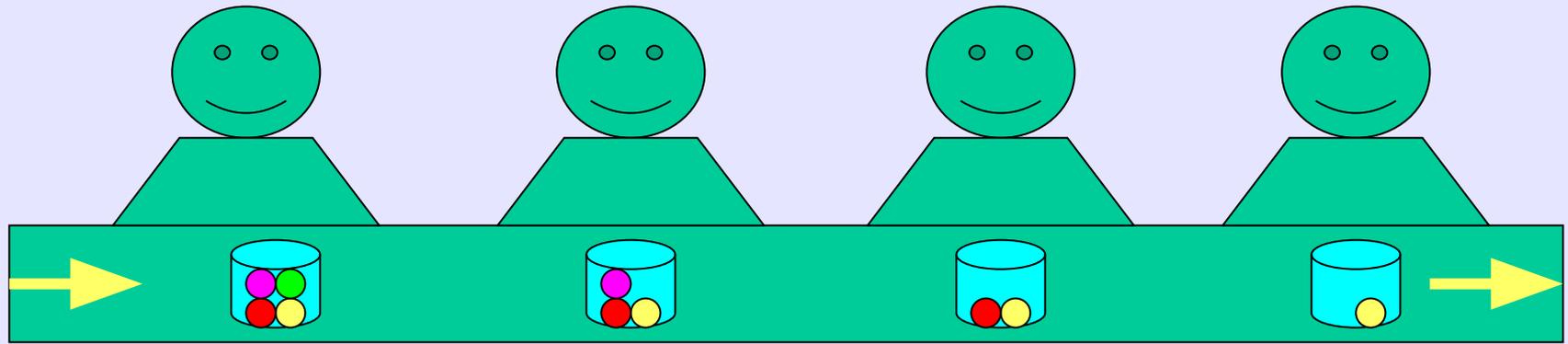


SpeedDaemon

– за счёт  
роста частоты

Brainiac –  
поумнение  
УУ, кэша,  
паралл.  
конвейеры

# • Конвейеризация (pipelining)



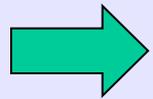
Конвейер выполняет  
одновременно несколько  
команд 

Процессорный цикл разбит на  
простые операции 

→ Можно увеличить частоту

Но!

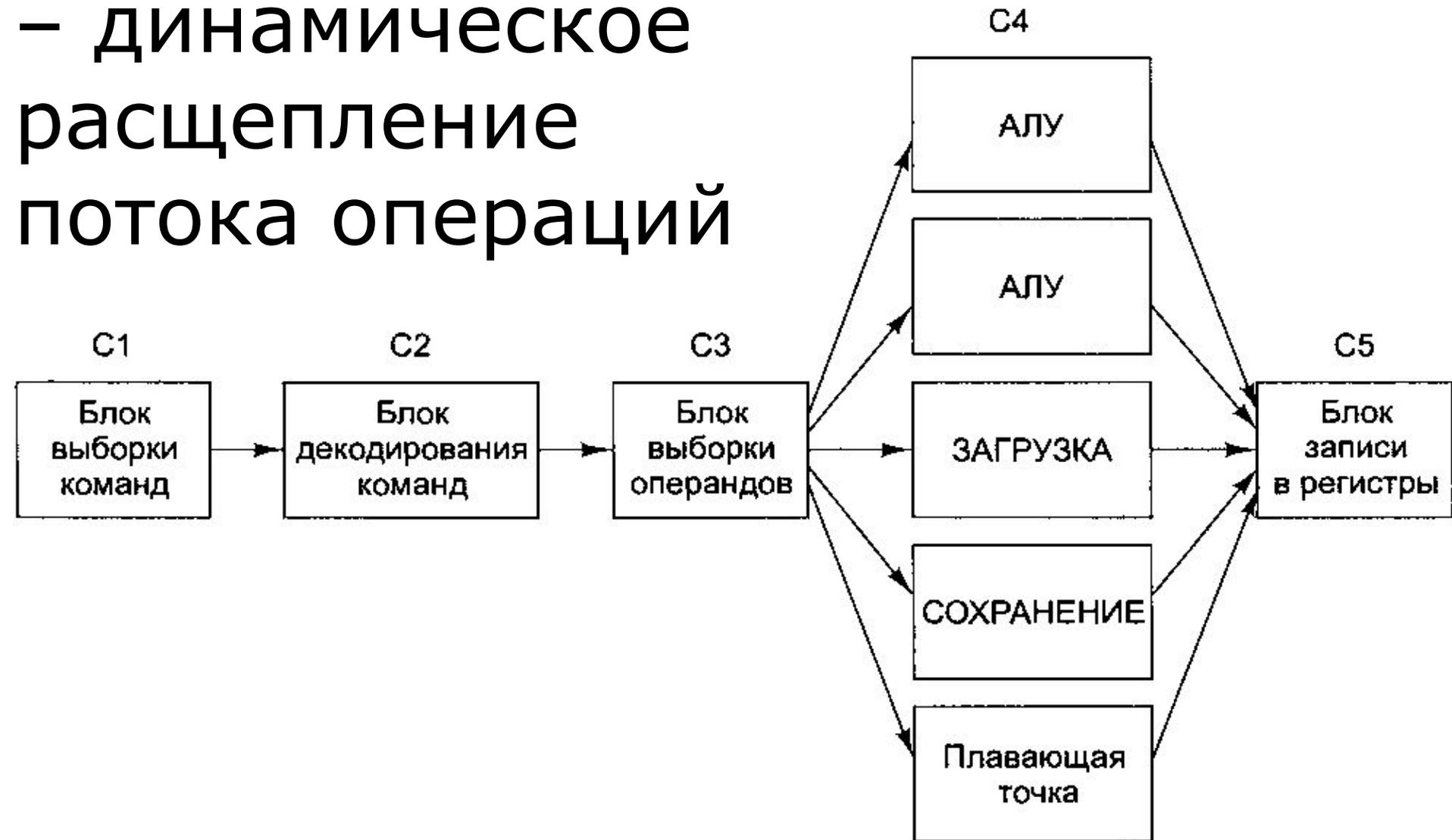
Если программа не  
оптимизирована под данный  
конвейер, то появляются  
холостые такты



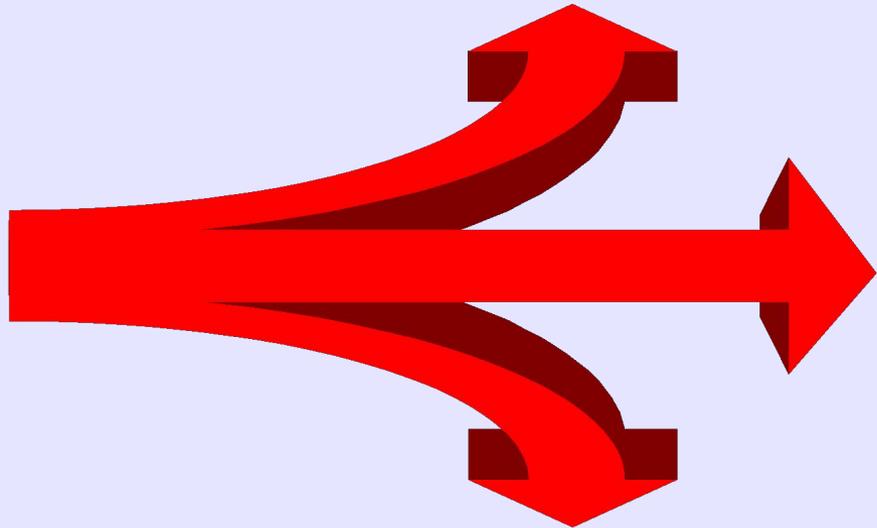
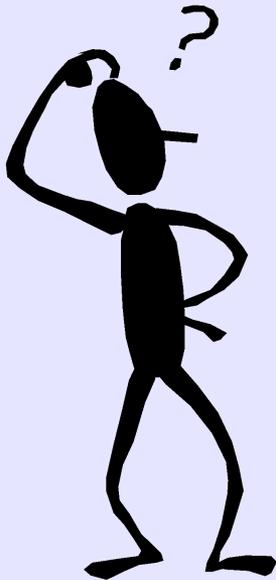
Эффективная длина  
конвейера – среднее  
число тактов в цикле

# Суперскалярность

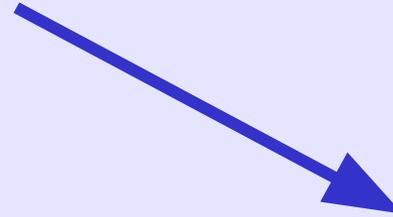
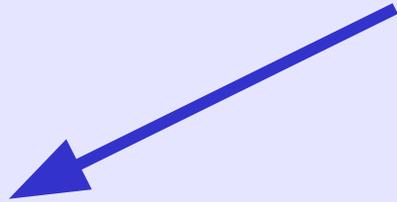
– динамическое  
расщепление  
потока операций



- Предсказание переходов –  
УУ прогнозирует направление  
условного перехода и, не  
дожидаясь его, выполняет 1  
и 2 этапы цикла



# Предсказание



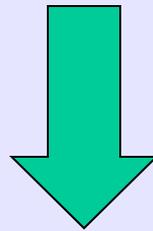
## Статическое

– на основе  
«опыта  
предков»,  
зашиито в УУ

## Динамическое

– на основе  
текущей  
статистики  
переходов

Если предсказание сбылось  
(98-99 % !), то ЦП не зря  
работал без «перерыва»



Экономия времени до 30 %

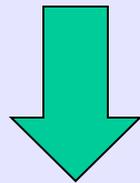
- **Исполнение по предположению** (speculative execution) – выполняются все этапы!

Но при неудаче конвейер очищается много тактов

- **Внеочередное исполнение** – порядок исполнения команд может меняться если их результаты независимы

Но в СШ результаты выдаются по порядку, т.е. программа «не замечает», что её «перекроили»

- **Переименование регистров** – если команды одновременно обращаются к одному регистру, его можно «размножить», используя запасные регистры



Логически он один, а физически – несколько

Н-р, схема  
работы одного  
из ядер Phenom

