Технологии повышения производительности МП

Технологии повышения производительности:
конвейеризация;
динамическое исполнение: предсказание ветвлений; внеочередное исполнение; ротация регистров; выполнение по предположению;
Многократное декодирование команд;
WLIV процессоры; технология Hyper-Treading

Термины

- Команда.
- Микрооперация или микрокоманда.
- Микропрограмма.
- Микропрограммный автомат.

Повышение производительности процессоров

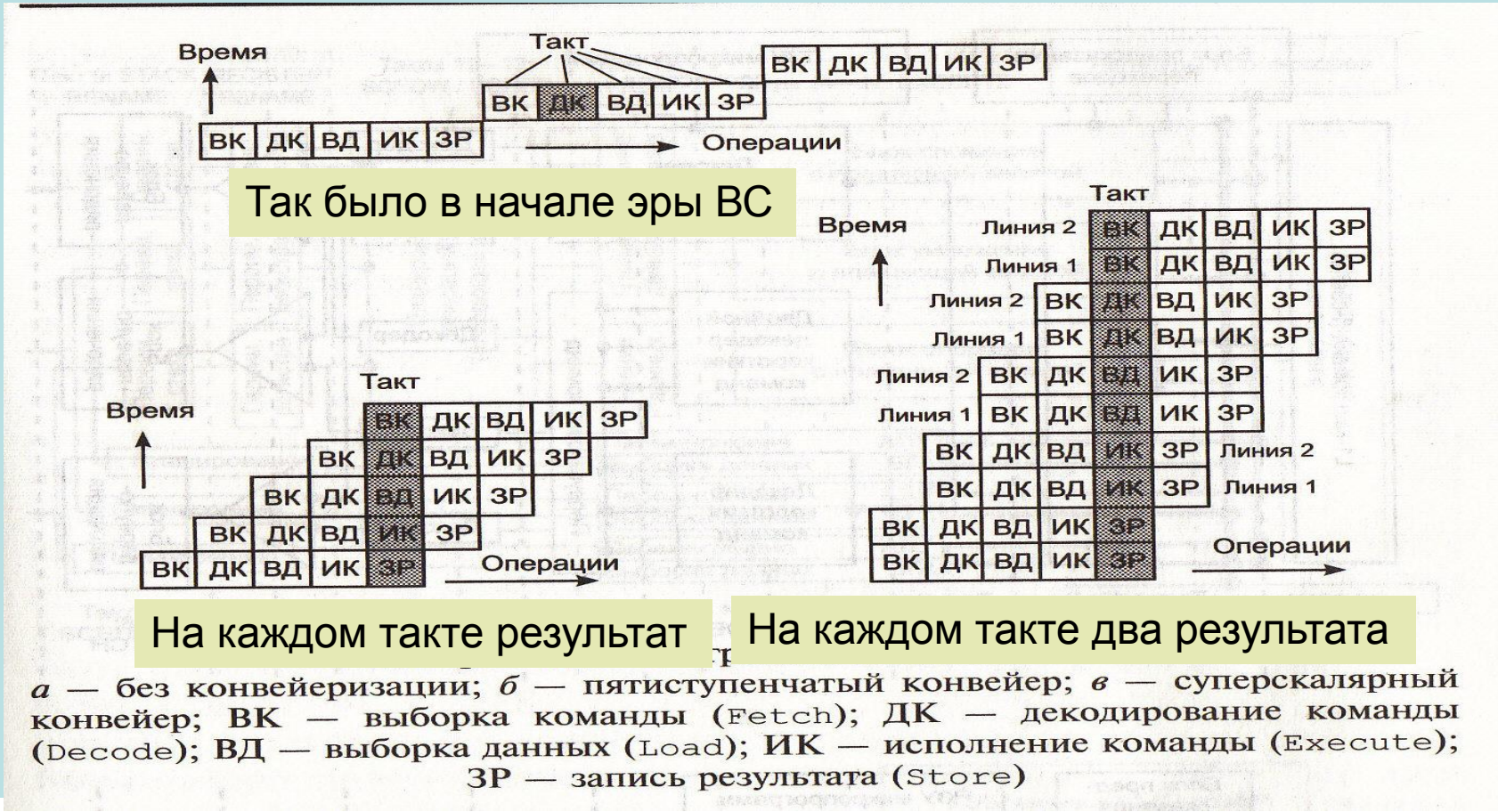
- Конвейеризация.
- Суперскаляризация.
- Увеличение количества исполнительных блоков.
- Введение принципа динамического исполнения команд.
- Гипертрейдинг.
- Параллелизм исполнения команд на уровне процессоров.

В 5-10 раз

более чем в 100 раз

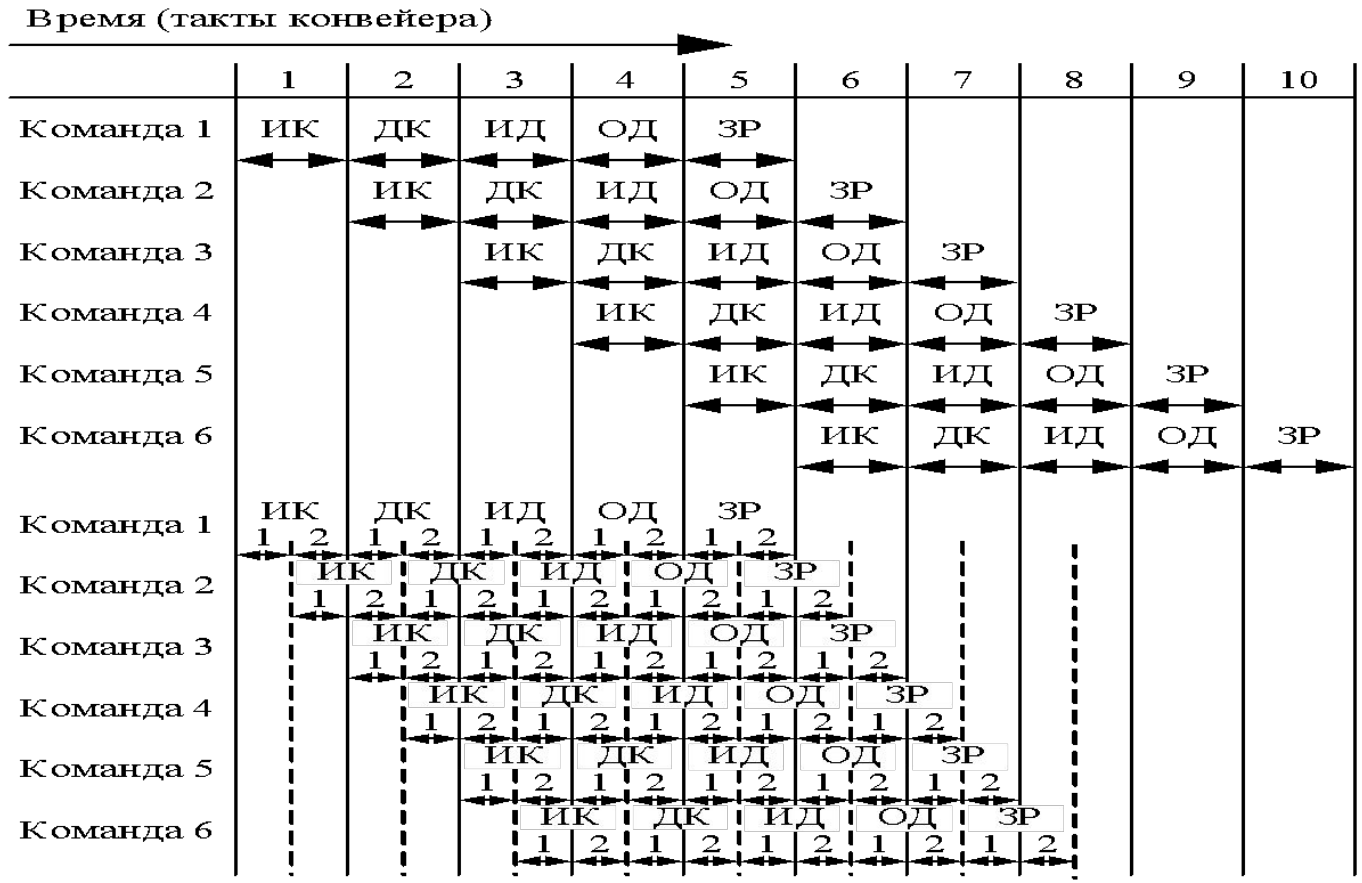
Архитектура процессоров с параллелизмом уровня команд

Как способ повышения производительности процессора



а — без конвейеризации; *б* — пятиступенчатый конвейер; *в* — суперскалярный конвейер; ВК — выборка команды (Fetch); ДК — декодирование команды (Decode); ВД — выборка данных (Load); ИК — исполнение команды (Execute); ЗР — запись результата (Store)

Суперконвейеризация



Каждый этап исполнения команды разбивается на меньшие этапы и повышается частота задающего генератора

Оценка производительности идеального конвейера

- **Предположим Твк=20; Тдк=15; Твд=20; Тик=25; Тзр=20;**
- **t = 5 – промежуточное время определяемое необходимостью записи промежуточных результатов.**
- **Тогда время такта**

T = max{Твк=20; Тдк=15; Твд=20; Тик=25; Тзр=20} + t = 30

При последовательной обработке время выполнения N команд:

T посл.= N*(Твк + Тдк + Твд + Тик + Тзр) = 100N

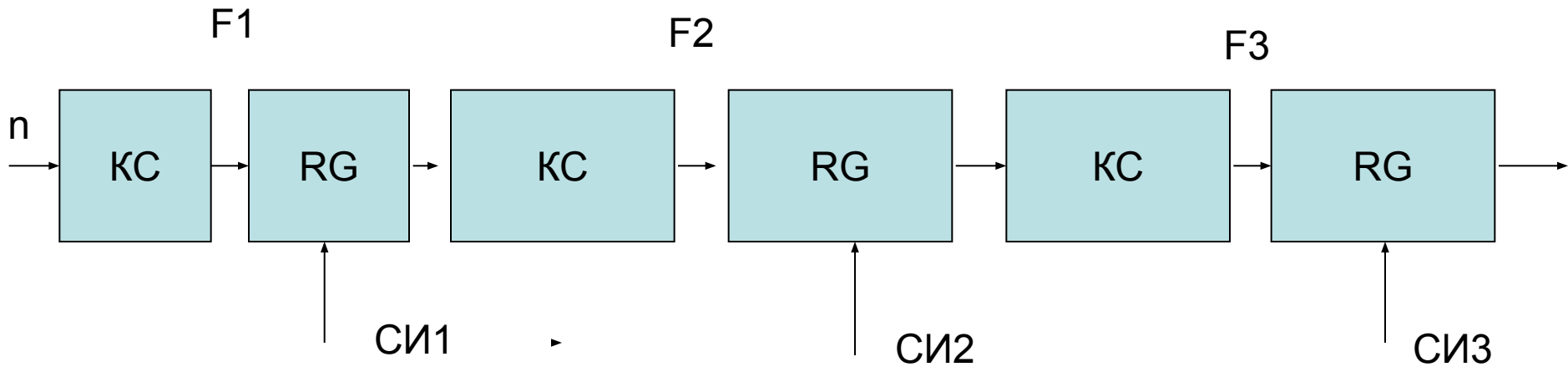
T конв.= 5*T+ (N-1) * T

N=10	**Тпосл.=1000**	**Тконв.=420**
N=100	**Тпосл.=10000**	**Тконв.=3120**

Идеальный – значит все команды выполняются последовательно за один такт и процесс состоит из пяти этапов одинаковой длинны.

Конвейер с точки зрения схемотехники

Чем проще схемотехника функционального узела, тем быстрее скорость вычисления.



Уменьшая функциональные блоки и увеличивая длину конвейера, можно увеличить тактовую частоту процессора.

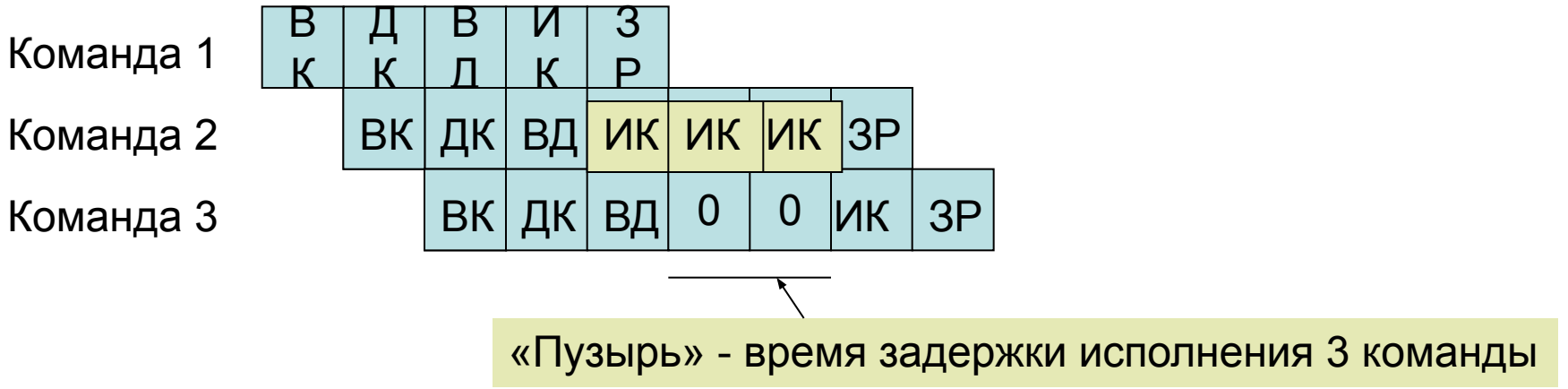
Конфликты в конвейере

- Конфликты – это ситуации при конвейерной обработке, которые препятствуют выполнению очередной команды.
- **Три группы конфликтов:**
  - **Структурные:**
    - **время выполнения команд разное;**
    - **конфликт обращений к ресурсам;**
  - **По управлению;**
    - **условные и безусловные переходы;**
  - **По данным.**
    - **команда исполняется, а данные не готовы**

Причины структурных конфликтов и способы минимизации их последствий

- **Скорость конвейера определяется скоростью исполнения самого медленного этапа исполнения команды.**

ПРИМЕР

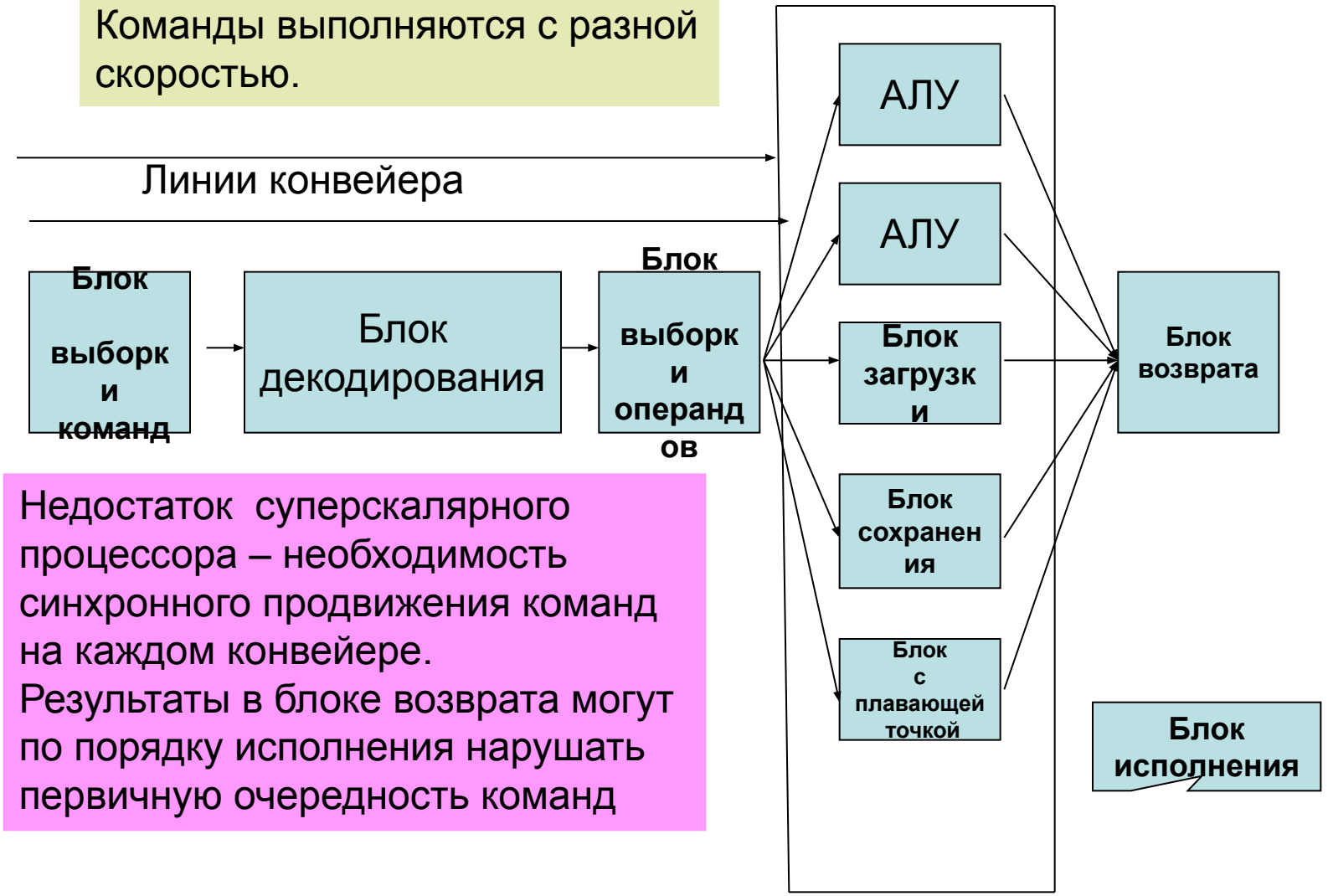


НЕ хватает исполнительных ресурсов для выполнения команды?

Можно ускорить выполнение команды за счет аппаратных решений в МП путем увеличения количества однотипных операционных блоков или за счет увеличения длительности такта исполнения команды.

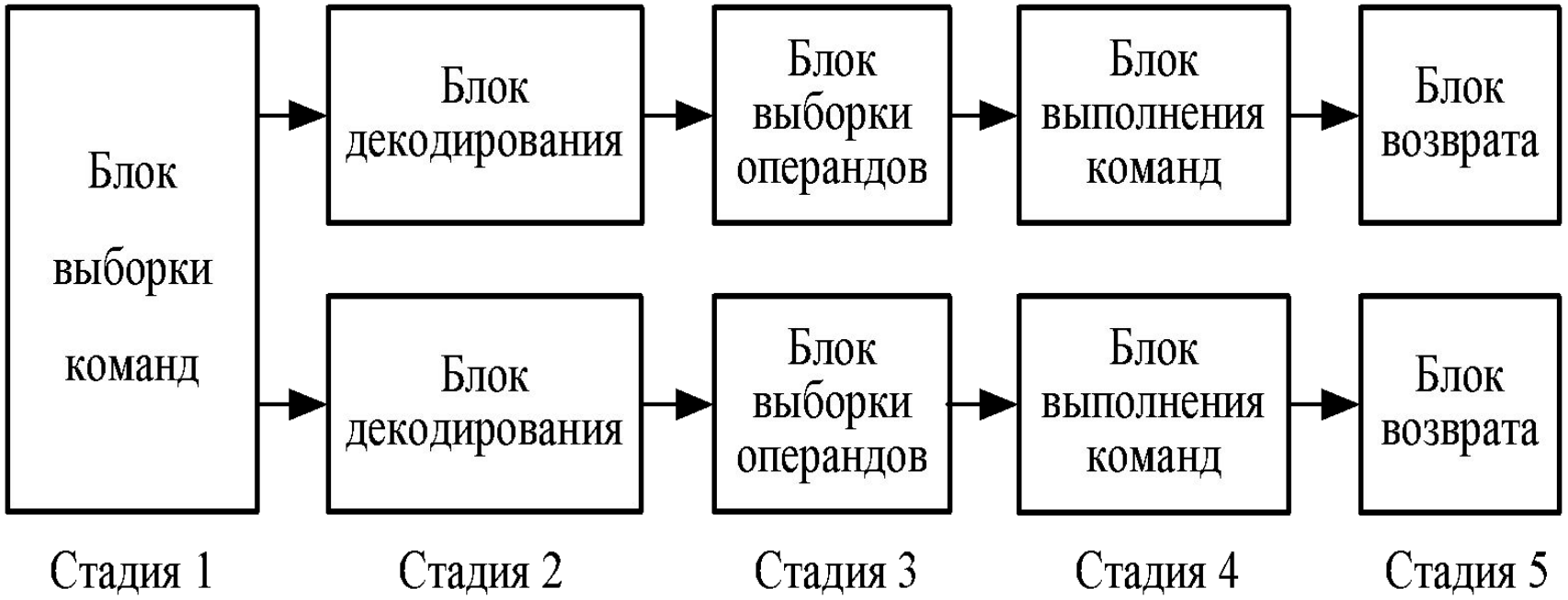
Конвейерная обработка команд суперскалярный процессор

Команды выполняются с разной скоростью.



Недостаток суперскалярного процессора – необходимость синхронного продвижения команд на каждом конвейере.
Результаты в блоке возврата могут по порядку исполнения нарушать первичную очередность команд

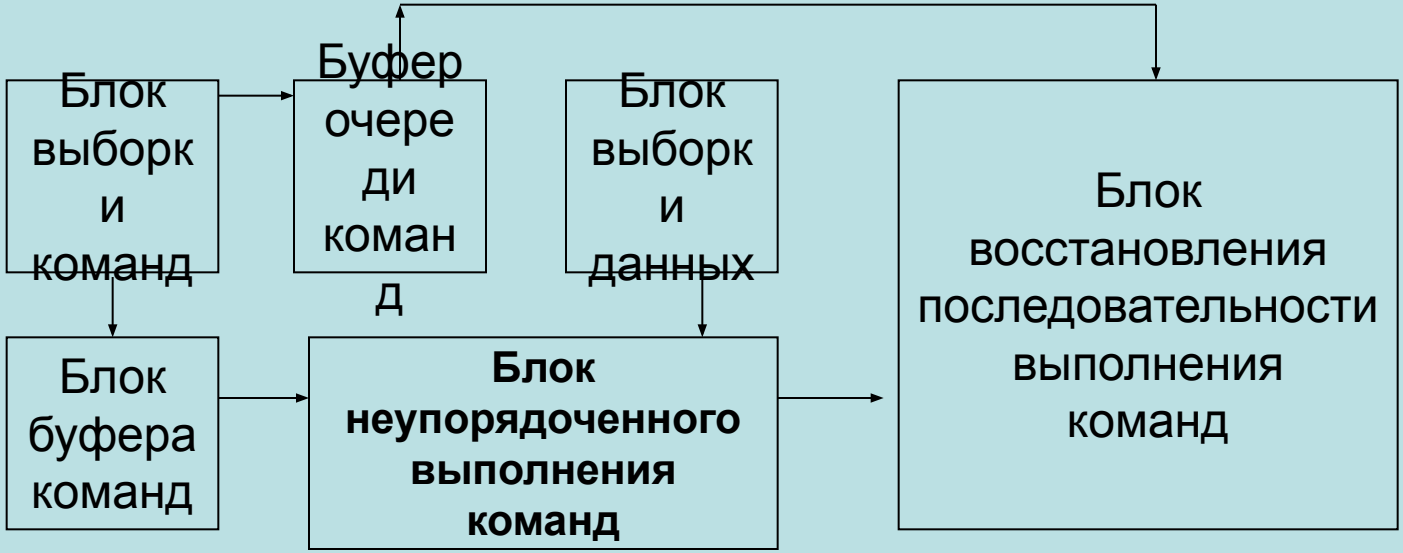
Суперскалярный процессор



В общем случае суперскалярный процессор может менять порядок выполнения машинных команд, заданный в исходном коде программы, не нарушая логики ее работы.

Конвейерное исполнение команд

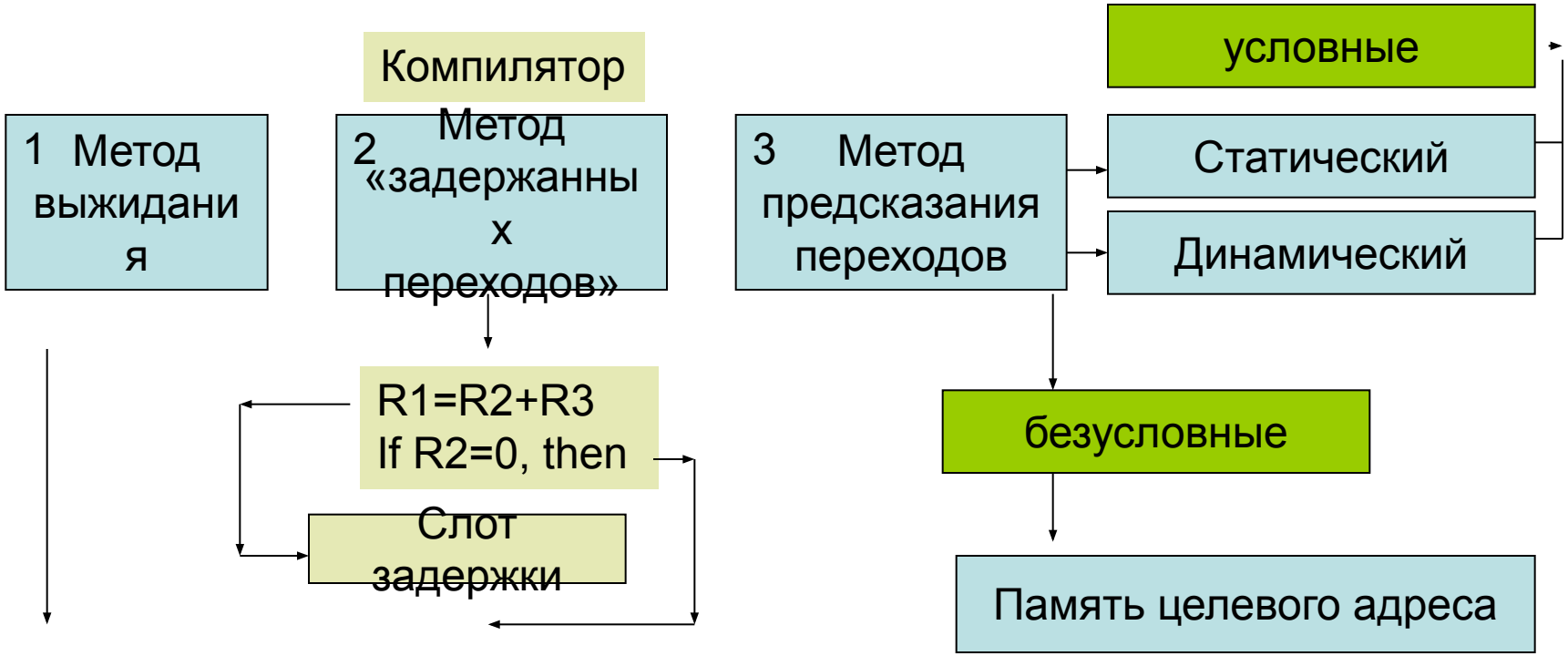
- Принцип неупорядоченного выполнения команд



Динамический принцип исполнения команд

Конфликты по управлению

- **Возникают при конвейеризации команд меняющих значение счетчика команд.**



Останавливается выполнение команд следующих за переходом, пока не станет известно направление перехода

Статические – используют компилятор.
Динамические – имеют сложную аппаратную часть.

Статическое предсказание переходов

- Осуществляется на основе некоторой априорной информации о подлежащей исполнению программе. Известны следующие стратегии:
- - переход происходит всегда;
- - переход не происходит никогда;
- - предсказание осуществляется по результатам профилирования;
- - предсказание определяется кодом операции команды перехода;
- - при первом выполнении команды переход происходит всегда. И.т.д.

Статическое прогнозирование переходов
ИСПОЛНЕНИЕ ПО ПРЕДПОЛОЖЕНИЮ

- **При компилировании программы можно создать граф предполагаемых ветвлений и задать вероятность направлений в процентах.**

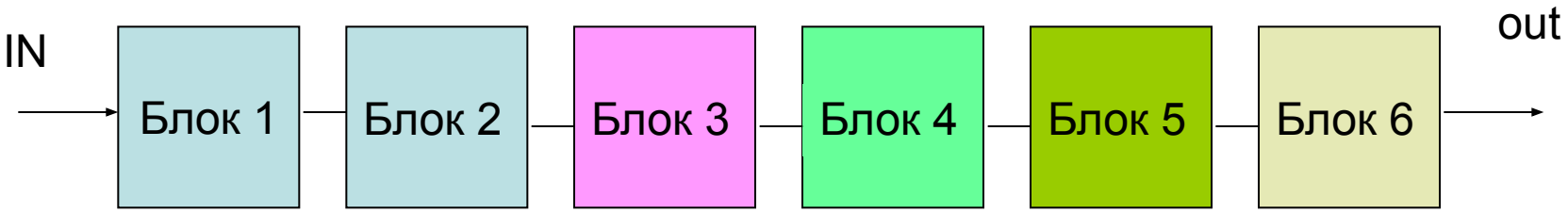
В результате - переход осуществляется на основании таблицы вероятности,
до вычисления реального условия перехода. Результаты исполнения накапливаются в специальном буфере.

НЕДОСТАТОК – возможно принятие неправильного решения

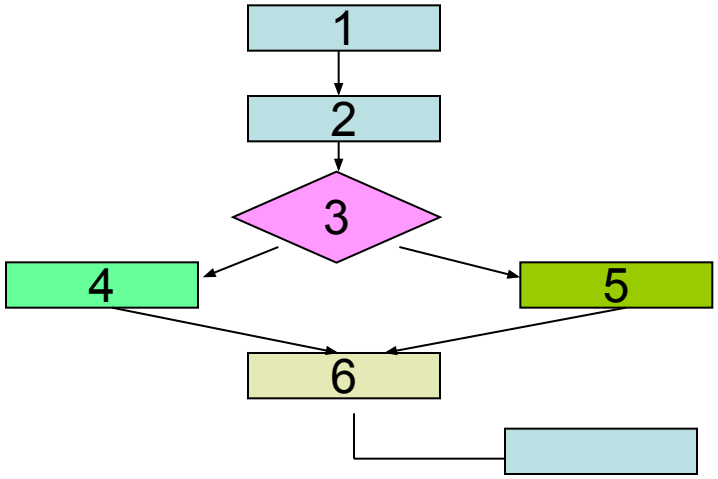
Предикативное выполнение

Предикация

Устраняет более половины ветвлений



Выполнение команды до того как станет известно нужна ли она.



Компилятор анализирует ветвления и помечает их предикатами (метками - ^)

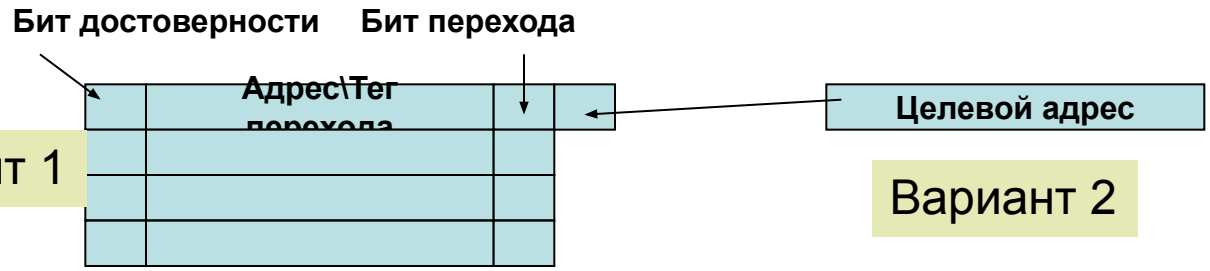
-123 - 34^6 - 35^6 -

Динамическое предсказание переходов

- Решение о наиболее вероятном исходе команды перехода принимается в ходе вычислений исходя из информации о предшествующих переходах.
- Динамические предсказания более точный инструмент

Динамическое прогнозирование ветвлений

- **Аппаратная реализация таблиц переходов**



Вариант 1

Вариант 2

BTB буфер

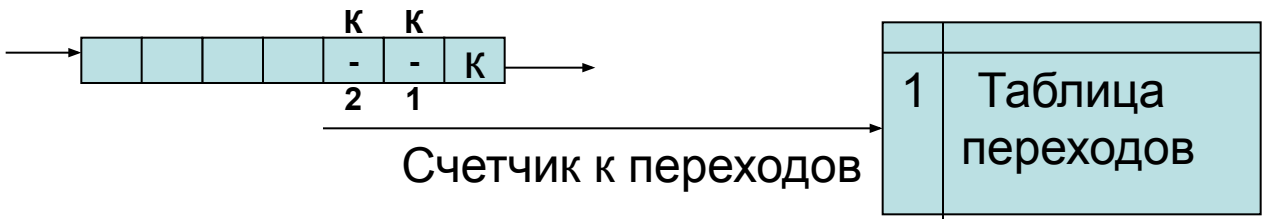
Прогноз – куда пойдет программа при выполнении перехода?

«1» - как раньше

«0» - по другому

Вариант 3

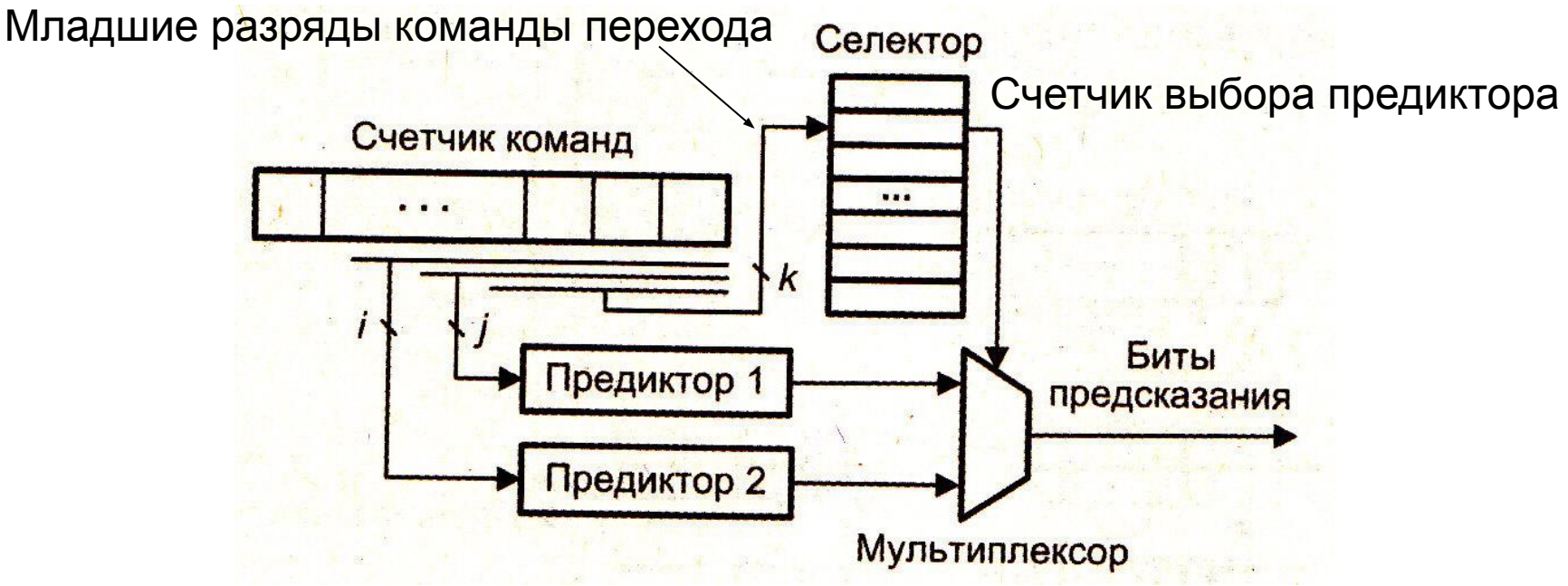
Сдвиговый регистр динамики переходов



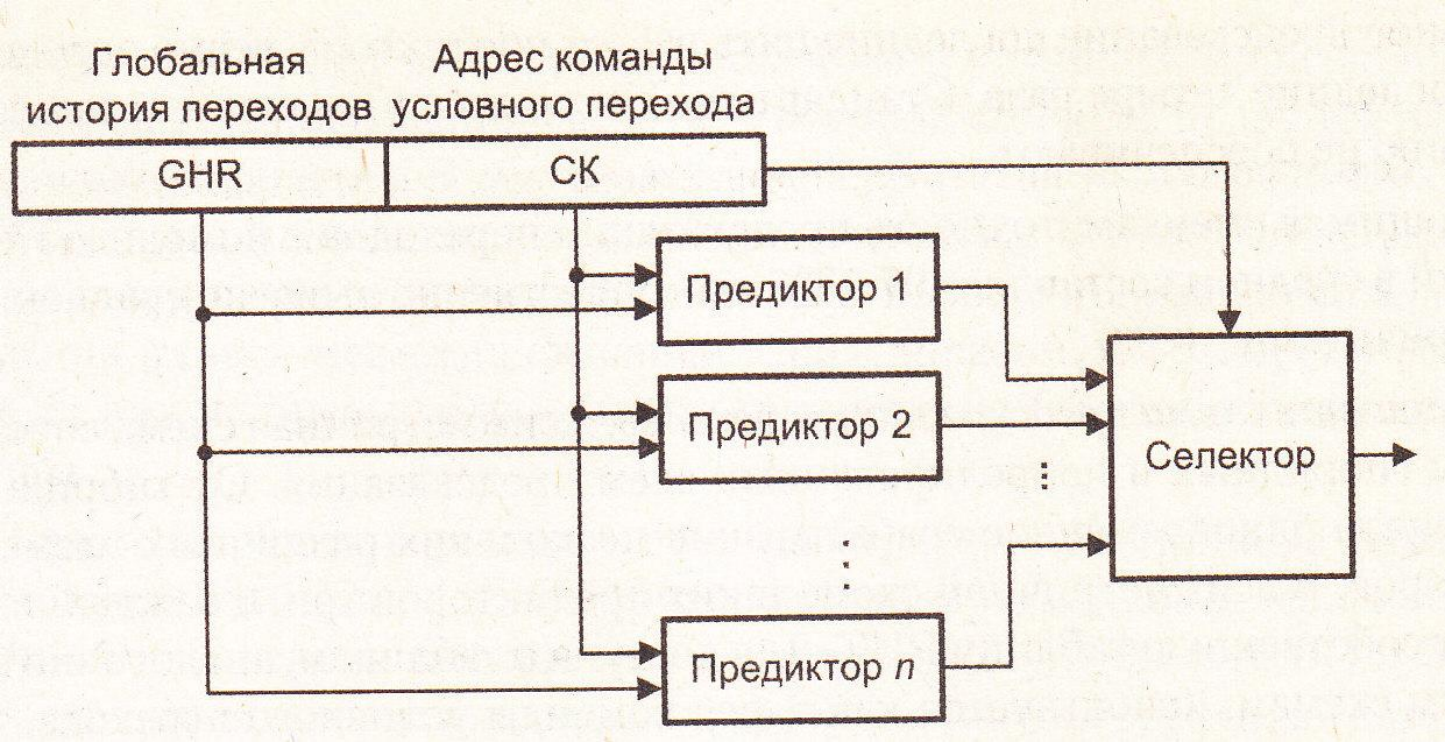
ЗАДАЧА – ускорить определение адреса команды, следующей за переходом.

Гибридный предиктор Макфарлинга

- Схема имеет два независимых предиктора, отличающиеся глубиной предыстории

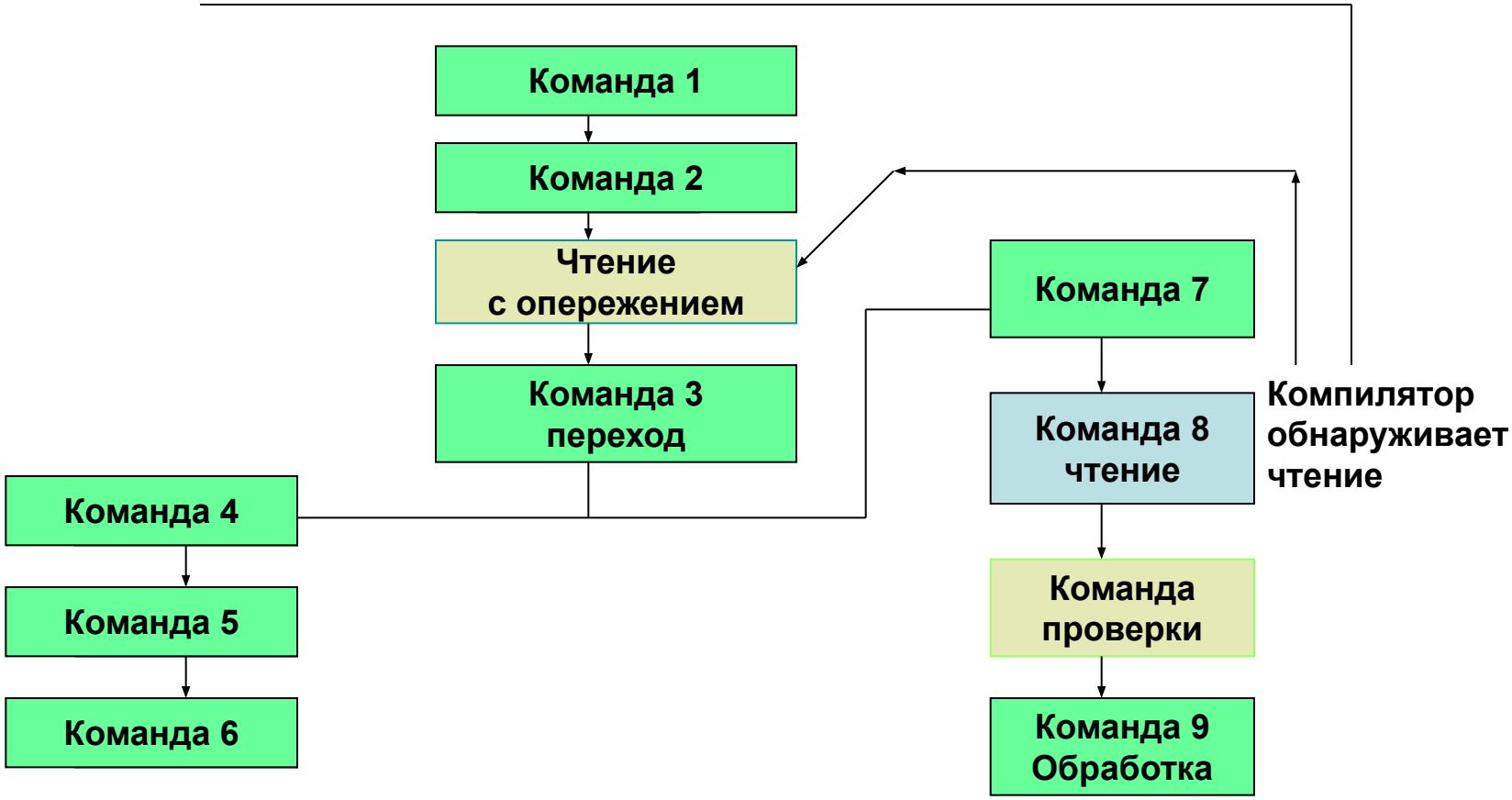


Общая схема гибридного предиктора



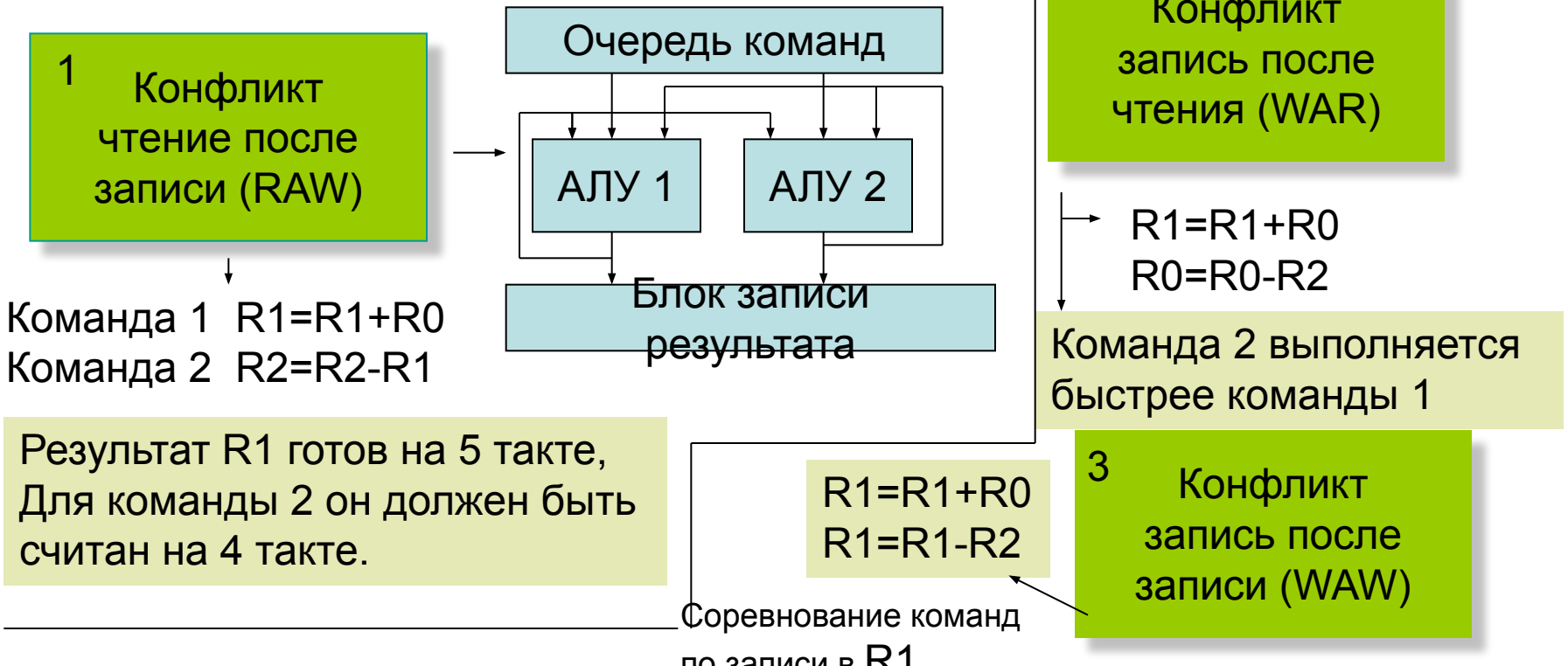
Спекулятивное выполнение команды

Опережающее чтение данных



Конфликты по данным и их решение

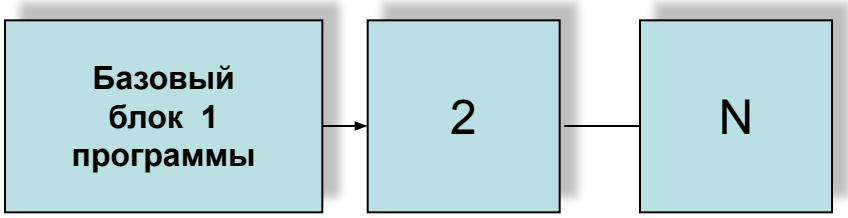
- **Выполнение одной команды зависит от результата выполнения другой**



Решение конфликтов по данным

- **Планирование загрузки конвейера.**

Компилятором выделяются участки программы без переходов



- **Переименование регистров МП.**

Таблица отображения регистров

Команда	Действие	Рабочий регистр
1	Запись в R0	R0 соответствует регистру PHY0
2	Чтение из R0	Чтение из PHY0
3	Запись в R0	R0 соответствует регистру PHY1
4	Чтение из R0	Чтение из PHY1

Технология динамического исполнения команд

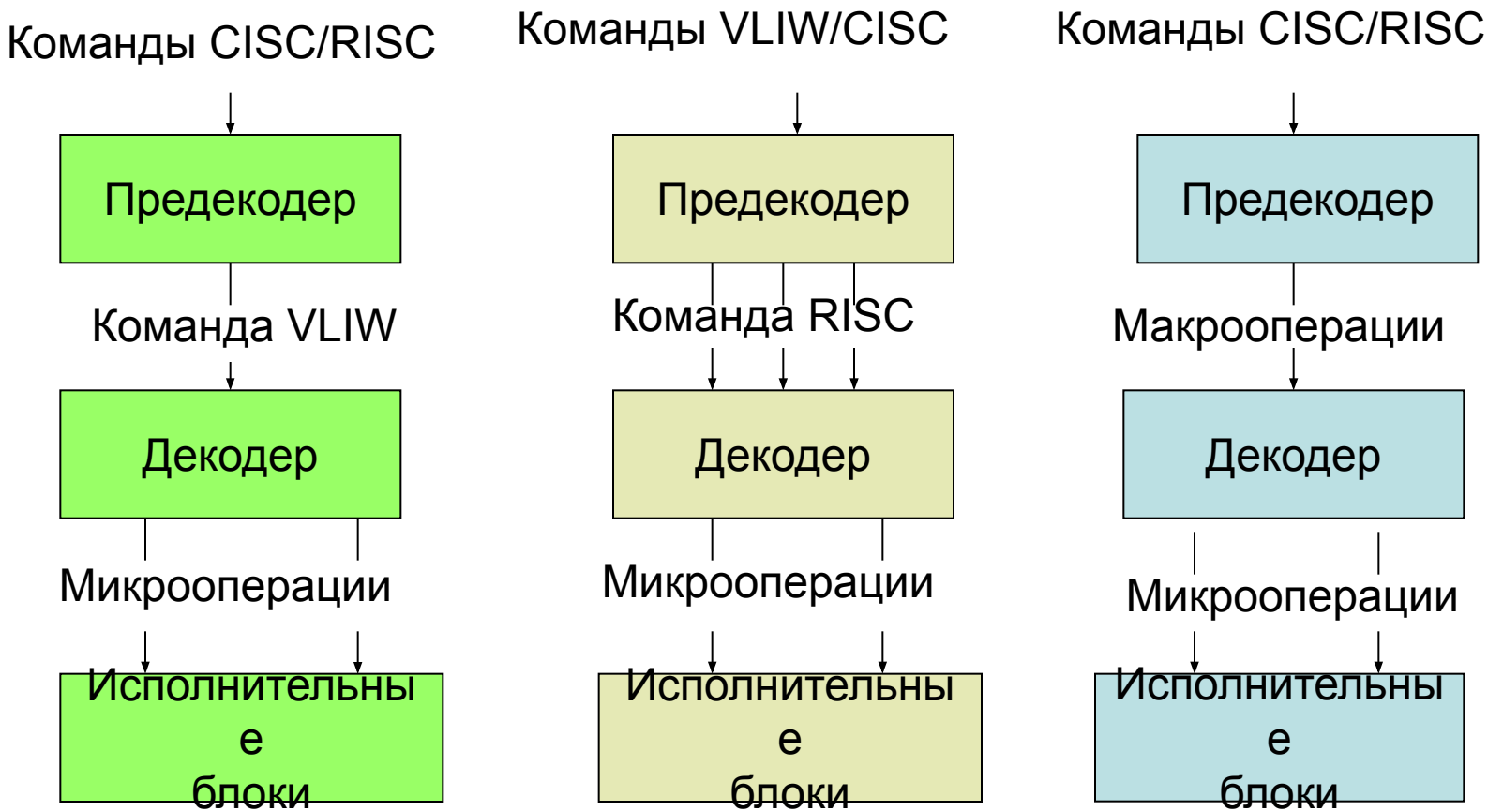
- Суперскалярность.
- Предсказание переходов.
- Неупорядочное исполнение команд.
- Предварительная загрузка данных.
- Переименование регистров.
- Предикативное исполнение.
- Резервирующая станция.
- Восстановление последовательности исполнения команд

Данная технология позволяет увеличить производительность процессора за счет оптимизации процесса исполнения команд

Характеристика конвейеров МП Intel и AMD

МП	i80486	Pentium	Pentium Pro	Pentium MMX	Pentium 2	Pentium 4	AMD Athlon
Число линий	1	2	3	2	3	3	3+3
Длина линий	5	5	14	6	14	20	17

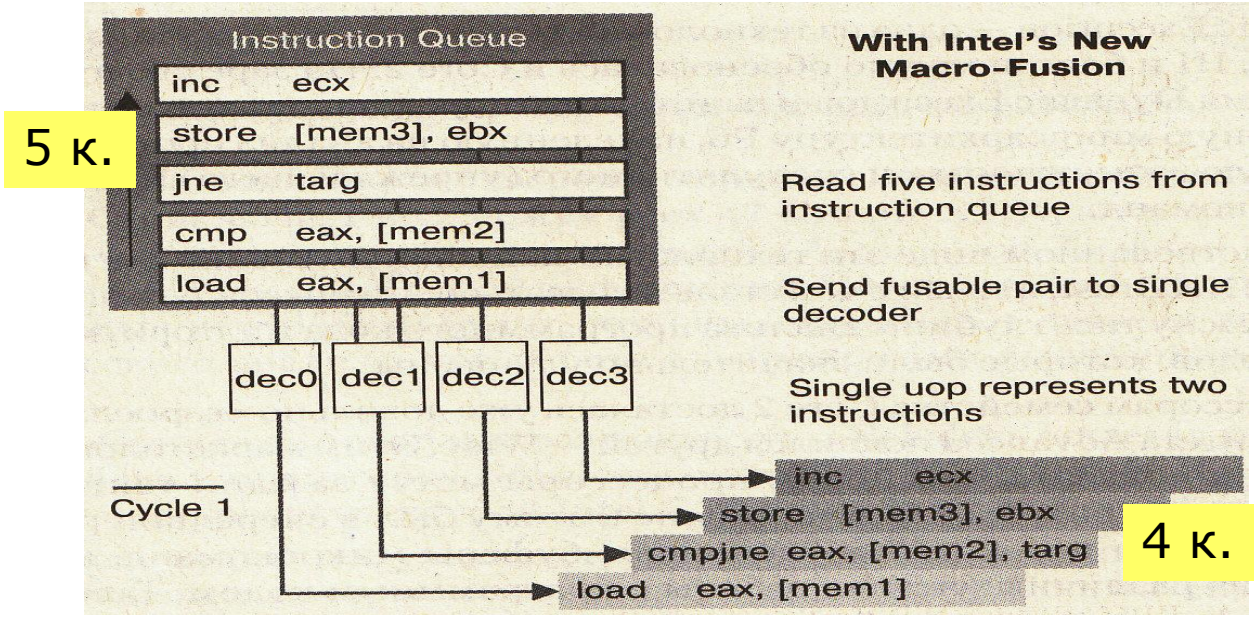
Технология многократного декодирование команд используя CMS



Code Morphing software (CMS) – программное обеспечение модификации кодов. Пример замены аппаратного декодирования программным.

Технология макрослияния (macrofusion)

- Макрослияние позволяет объединять типичные пары последовательных команд (например - сравнение и условный переход) в одну макрокоманду. И выполнять их в дальнейшем как одну.



Технология микрослияния (Micro-op fusion)

Команды при декодировании могут использовать одинаковые микрокоманды

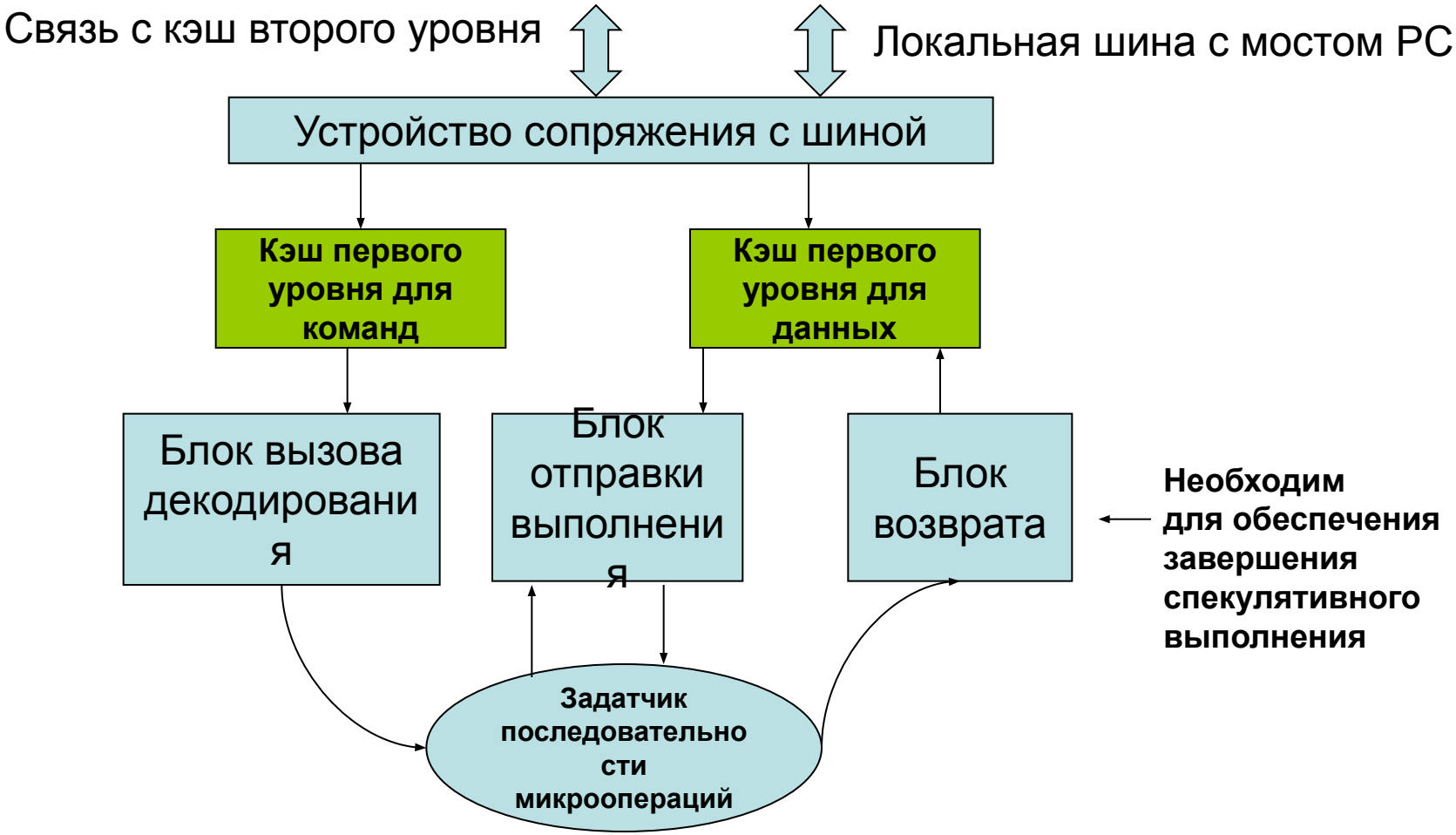
Технология предусматривает однократный вызов микрокоманды для разных команд. Технология позволяет уменьшить общее количество одновременно вызываемых микрокоманд до 10 %.

Технология применяется только в потоковых процессорах

Технология резервирующей станции

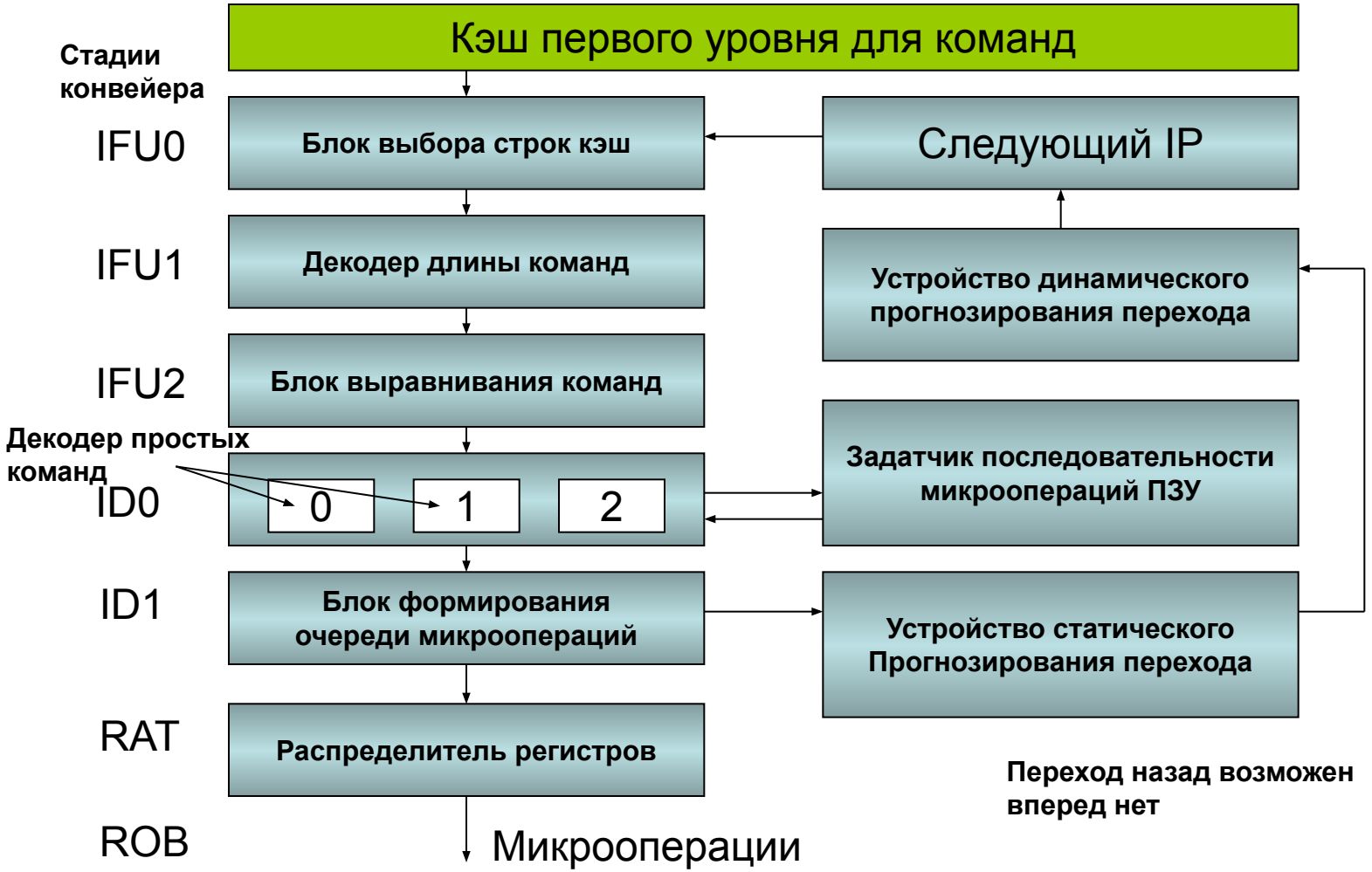
- Команды выполняются с разным быстродействием.
- Команды могут зависеть друг от друга.
- Командам могут требоваться одинаковые ресурсы для исполнения.
- Командам при выполнении необходимо обращение к памяти.

Микроархитектура Pentium2



3 линии конвейера по 14 ступеней

Микроархитектура Pentium2
блок вызова декодирования



Микроархитектура Pentium2
блок отправки\выполнения

Решает задачу выполнения очередности микрокоманд и разрешает конфликты ресурсов

Микроархитектура Pentium2 блок возврата

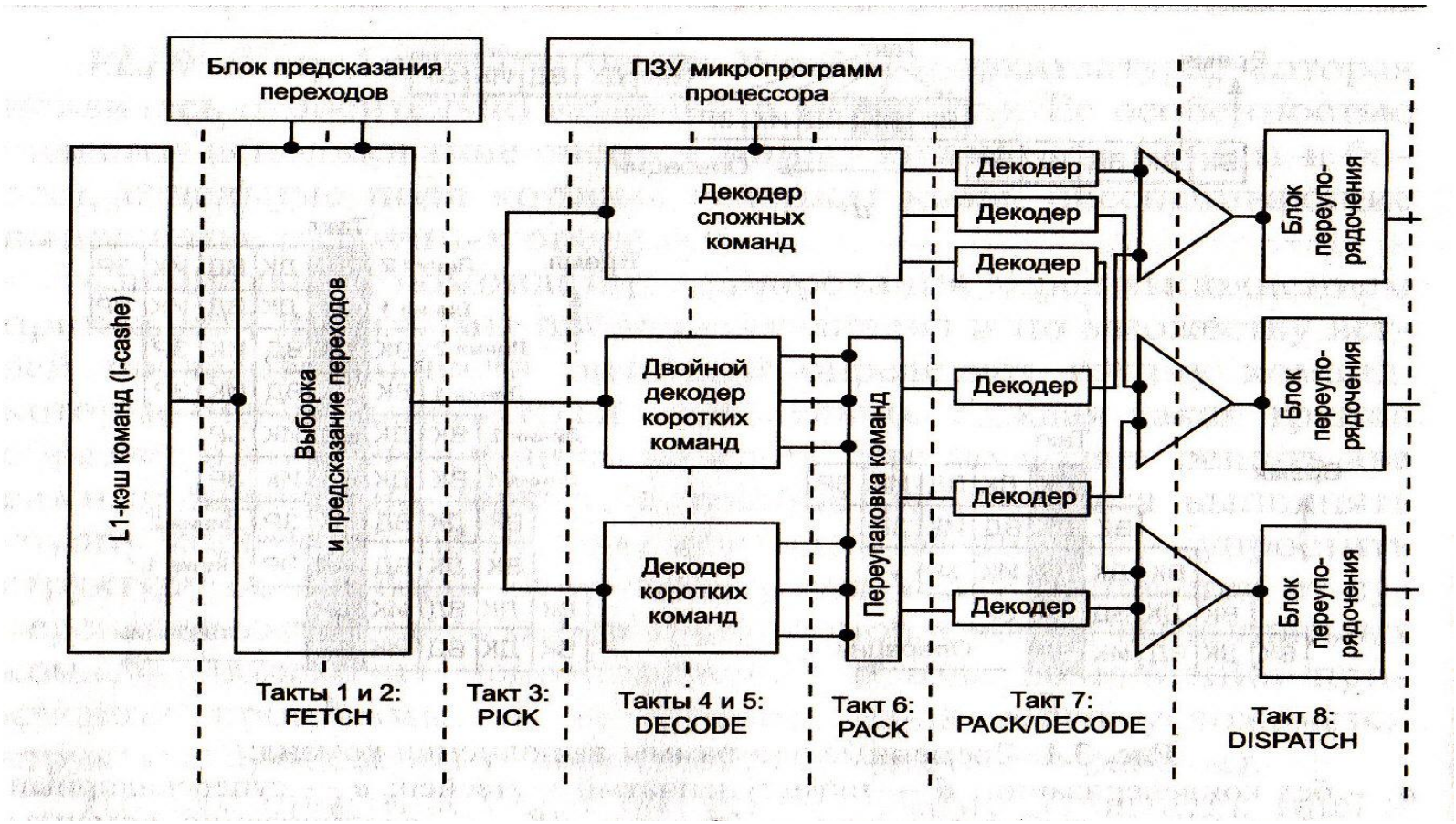
- Отвечает:

  - за отправку результатов в регистры или устройства, которым они требуются.

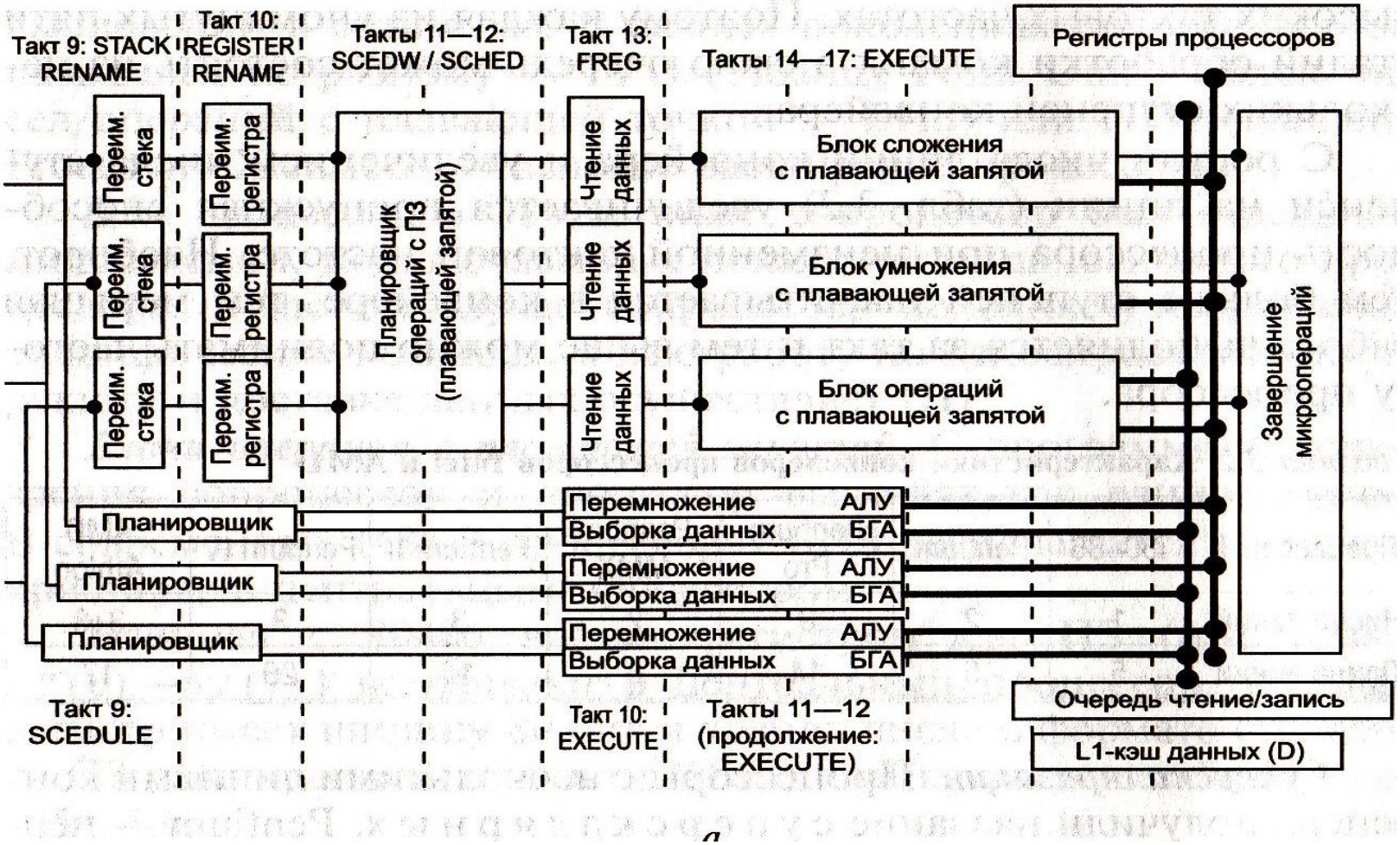
  - контроль возврата после спекулятивного исполнения (отбрасываются результаты микрокоманд, которые в дальнейшем не нужны).

  - временное хранение результатов исполнения микрокоманд.

Пример конвейера AMD K8 часть1



Пример конвейера AMD K8 часть 2



Проблемы суперскальных МП

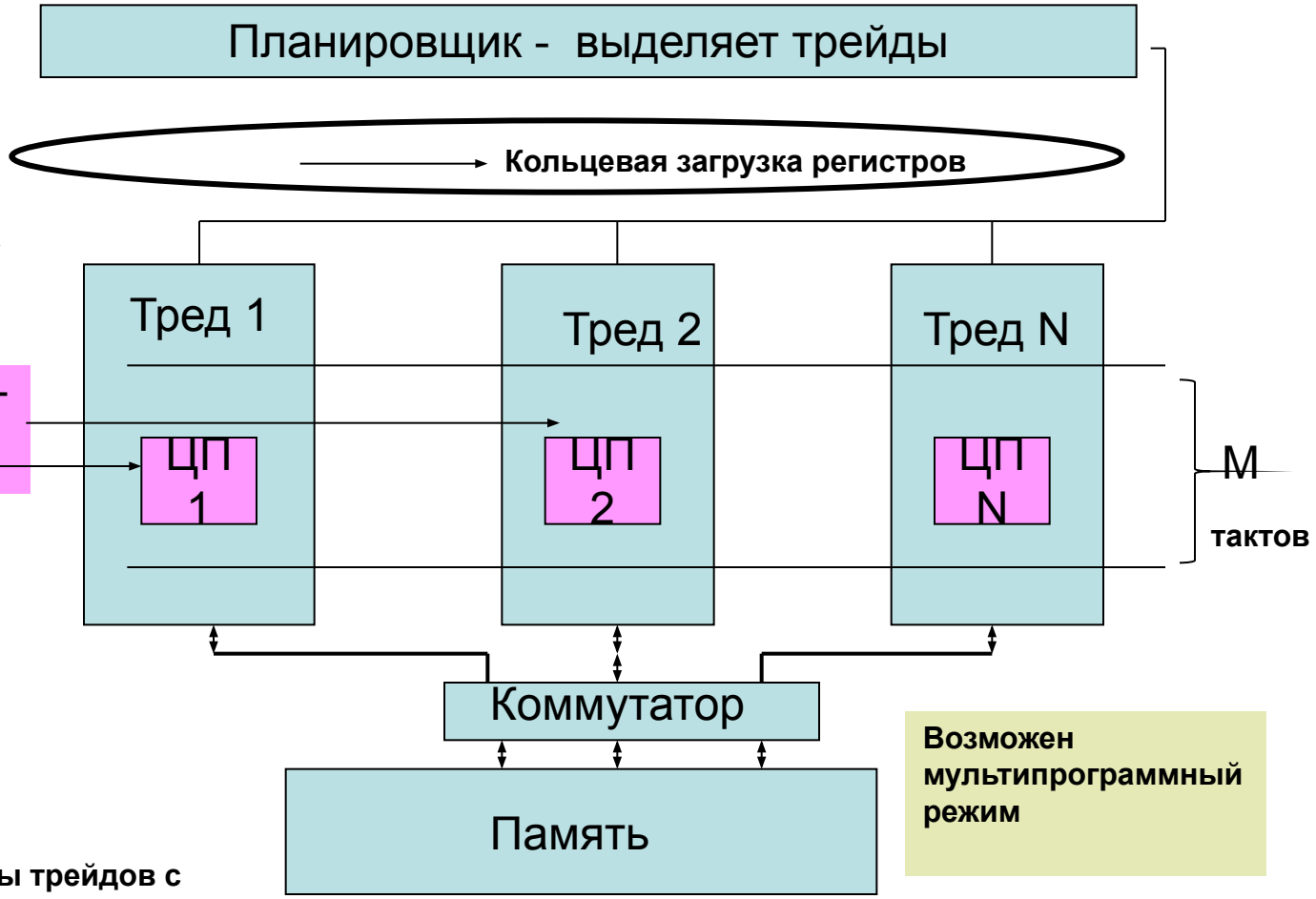
- **Простои конвейеров из-за нерегулярной загрузки функциональных устройств МП.**
- **Наличие одного счетчика команд.**
- **Ограничение на количество конвейеров и функциональных устройств МП – непропорциональное усложнение структуры.**
- **Сложные схемы декодирования команд и др. блоков.**
- **Одновременное исполнение нескольких программ только в режиме разделения времени.**

Мультитрейдовые микропроцессоры

- Тред – вычислительный процесс обслуживаемый отдельным набором регистров.
- Однотрейдовый микропроцессор – имеет один счетчик команд.
- Мультитрейдовый МП – выполняет одновременно несколько процессов и решает проблему простоя функциональных устройств из за невозможности выполнить следующую команду.
- Трейдом может быть как команда так и последовательность команд.

Принцип работы мультитрейдовой архитектуры

Переключение на следующий трейд происходит принудительно или при наступлении простоя процессора



2 второй вариант мультитрейда

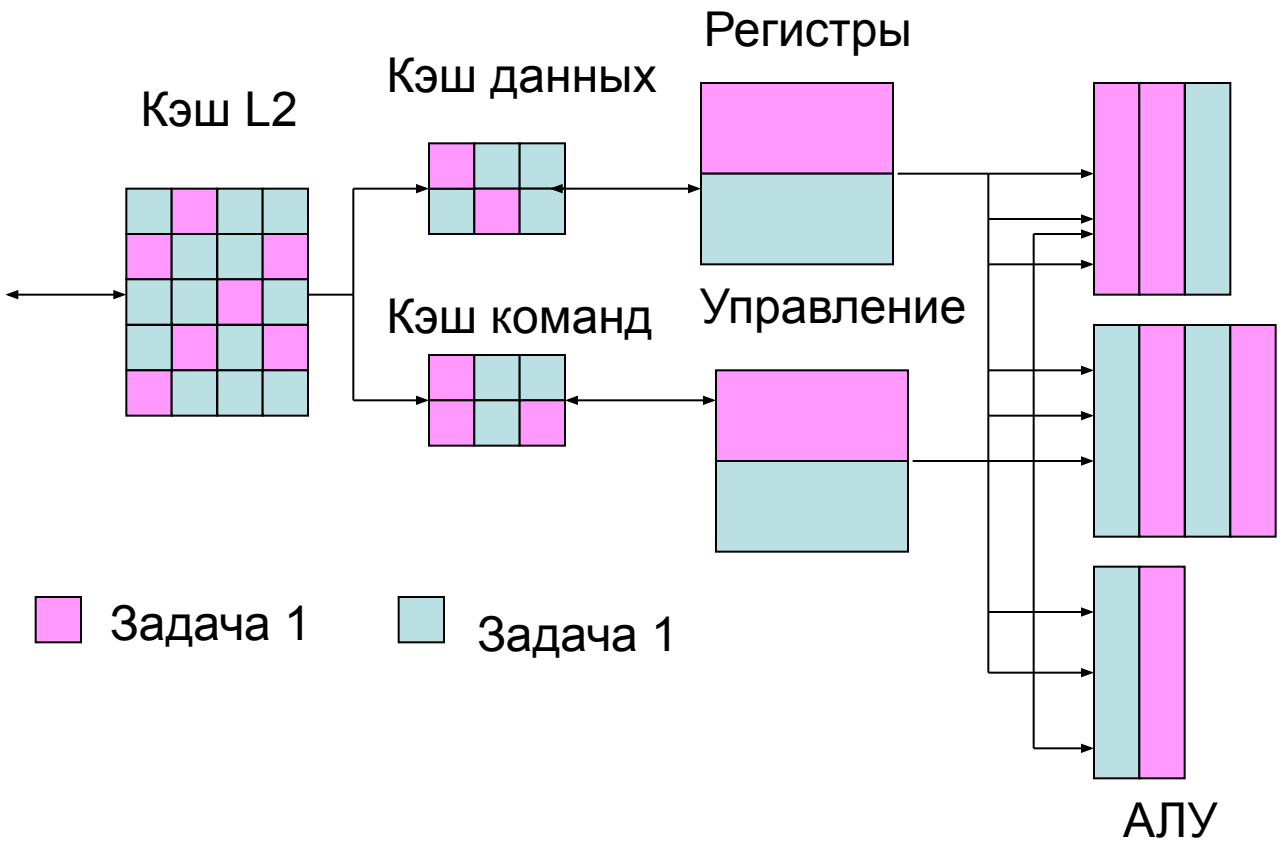
Xeon
Pentium 4
2 трейда

Возникают проблемы согласованности работы трейдов с памятью

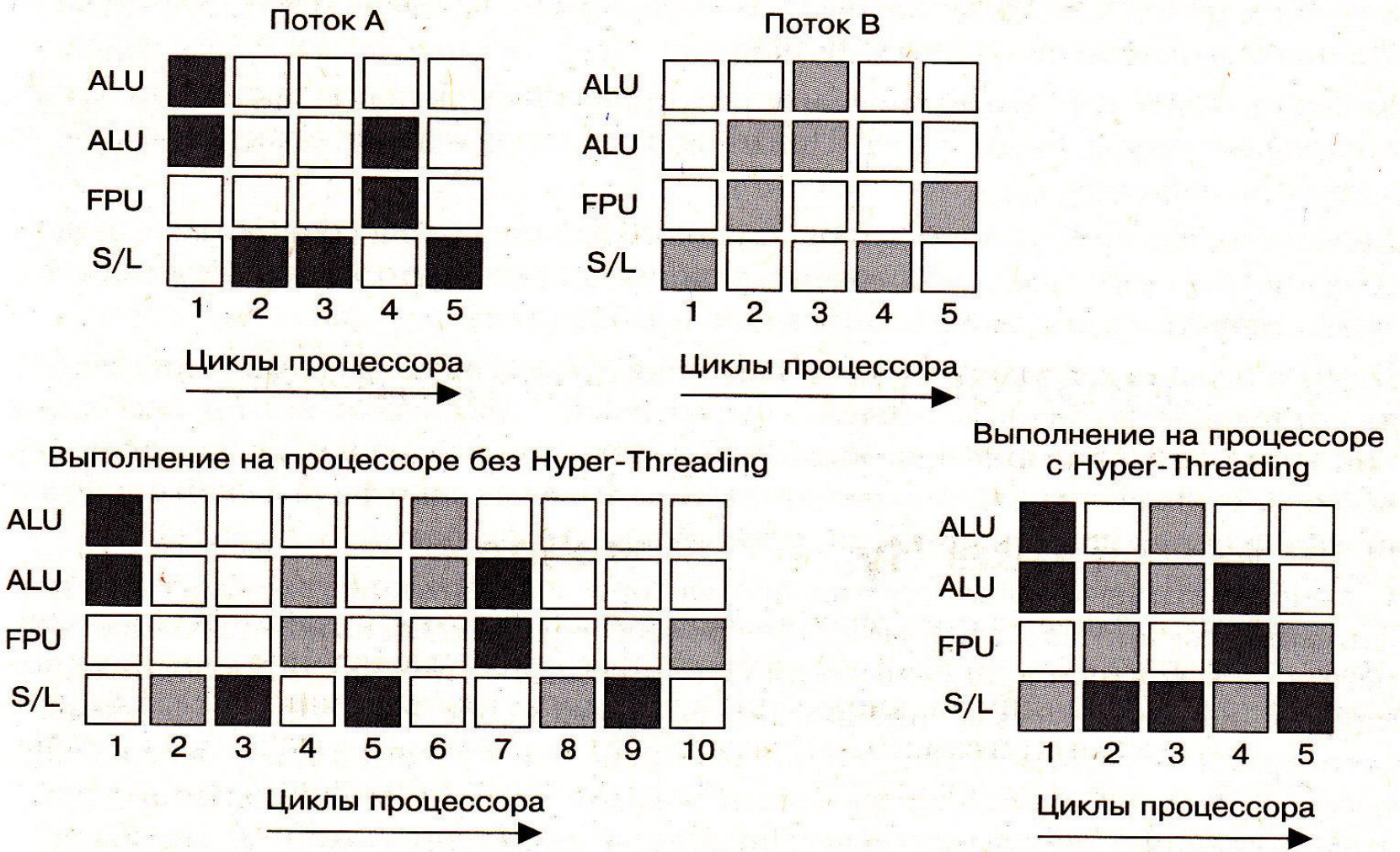
Возможен мультипрограммный режим

Технология Hyper-Threading

- **Реализуется идея разделения времени на аппаратном уровне**



Технология Hyper-Threading

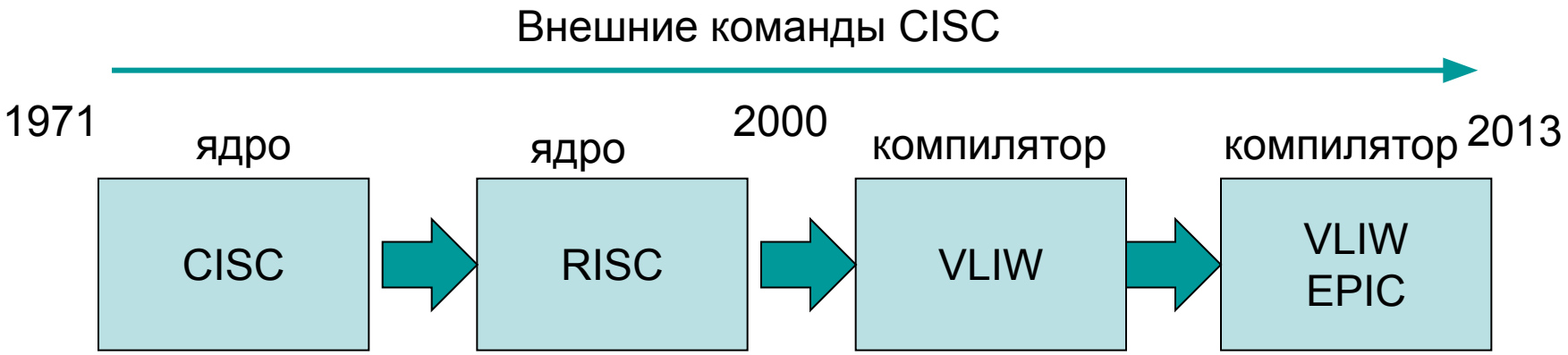


Основные отличия архитектур IA-32 и IA-64

IA-32	IA-64
Использование сложных команд переменной длины, обрабатываемых последовательно	**Использование простых команд одинаковой длины, сгруппированных в связки VLIW**
Переупорядочивание и оптимизация команд в процессе выполнения	**Переупорядочивание и оптимизация в процессе компиляции**
Попытки предсказания переходов аппаратно	**Параллельное выполнение последовательностей команд без предсказания переходов**
Загрузка данных по мере необходимости, первым проверяя кэш	**Загрузка данных прежде, чем они потребуются и кэш тоже**

IA-64 первый компромисс между CISC и RISC. 2 режима декодирования команд VLIW и CISC с автоматическим переключением.

Этапы развития структур МП по системам команд



Аппаратная оптимизация процесса

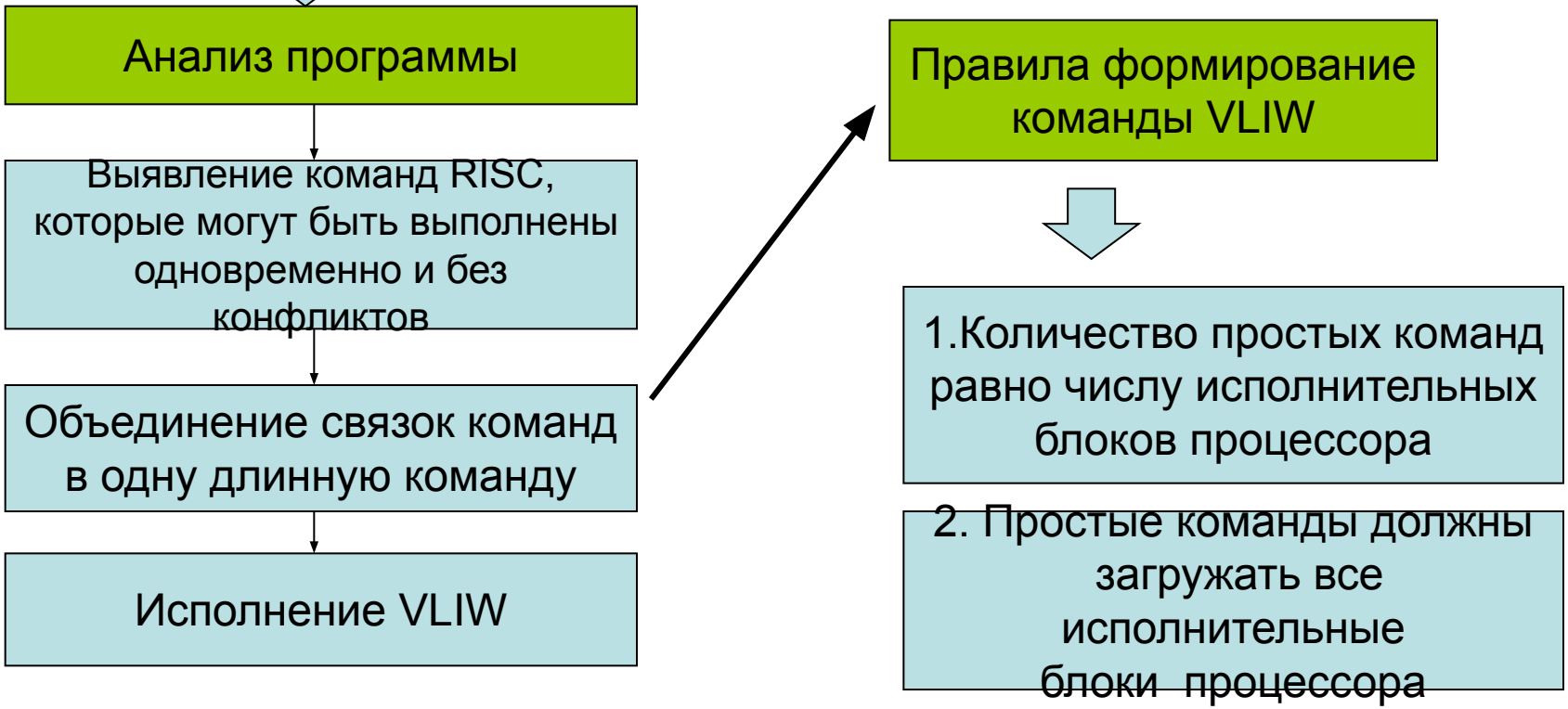
Программная и аппаратная оптимизация процесса

Причина перехода к многоядерности → Исчерпана возможность повышения производительности за счет повышения частоты? НЕТ – Явный параллелизм вычислений!!!

Общая шина - до 32 процессоров. При большем количестве ядер система снижает производительность

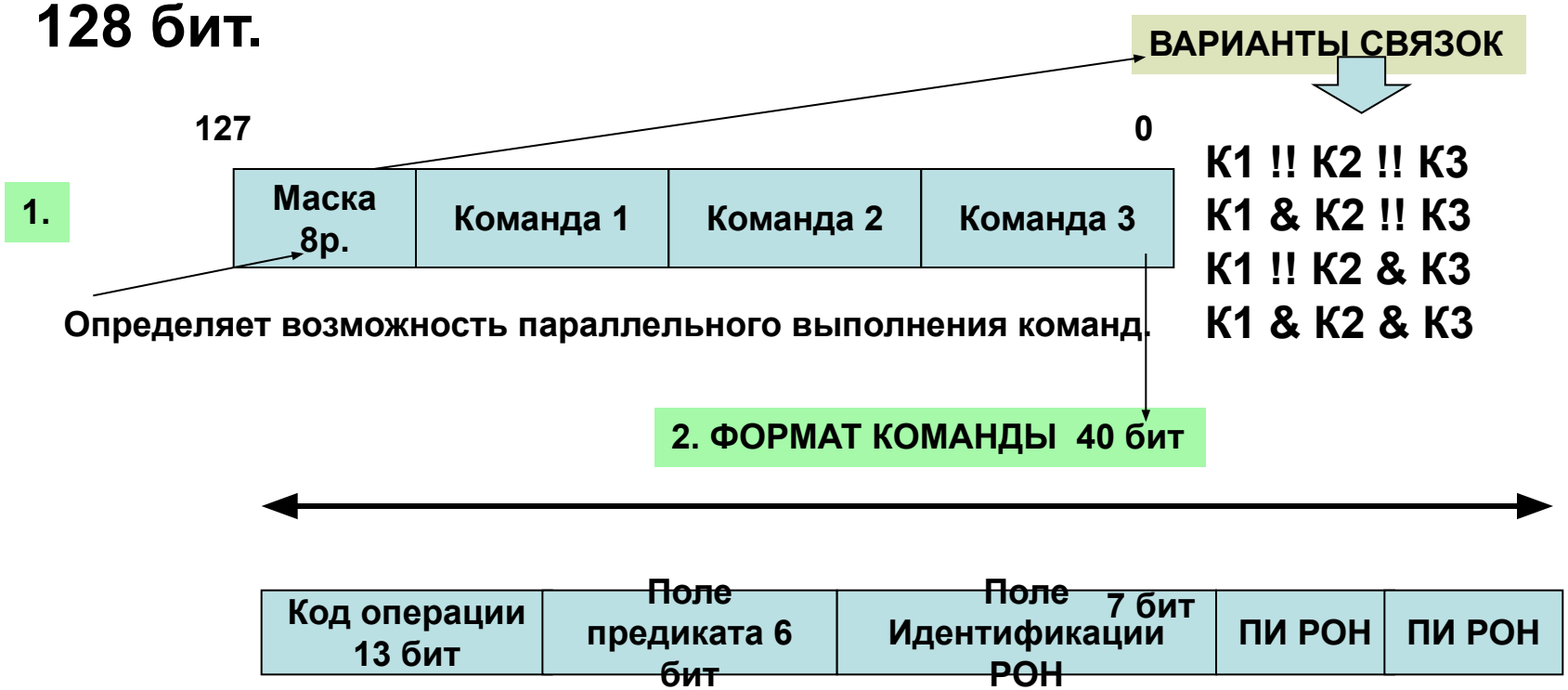
Синтез команд для процессоров VLIW

Задача эффективного планирования параллельных вычислений команд возлагается на «разумный» компилятор Code Morphing software (CMS).



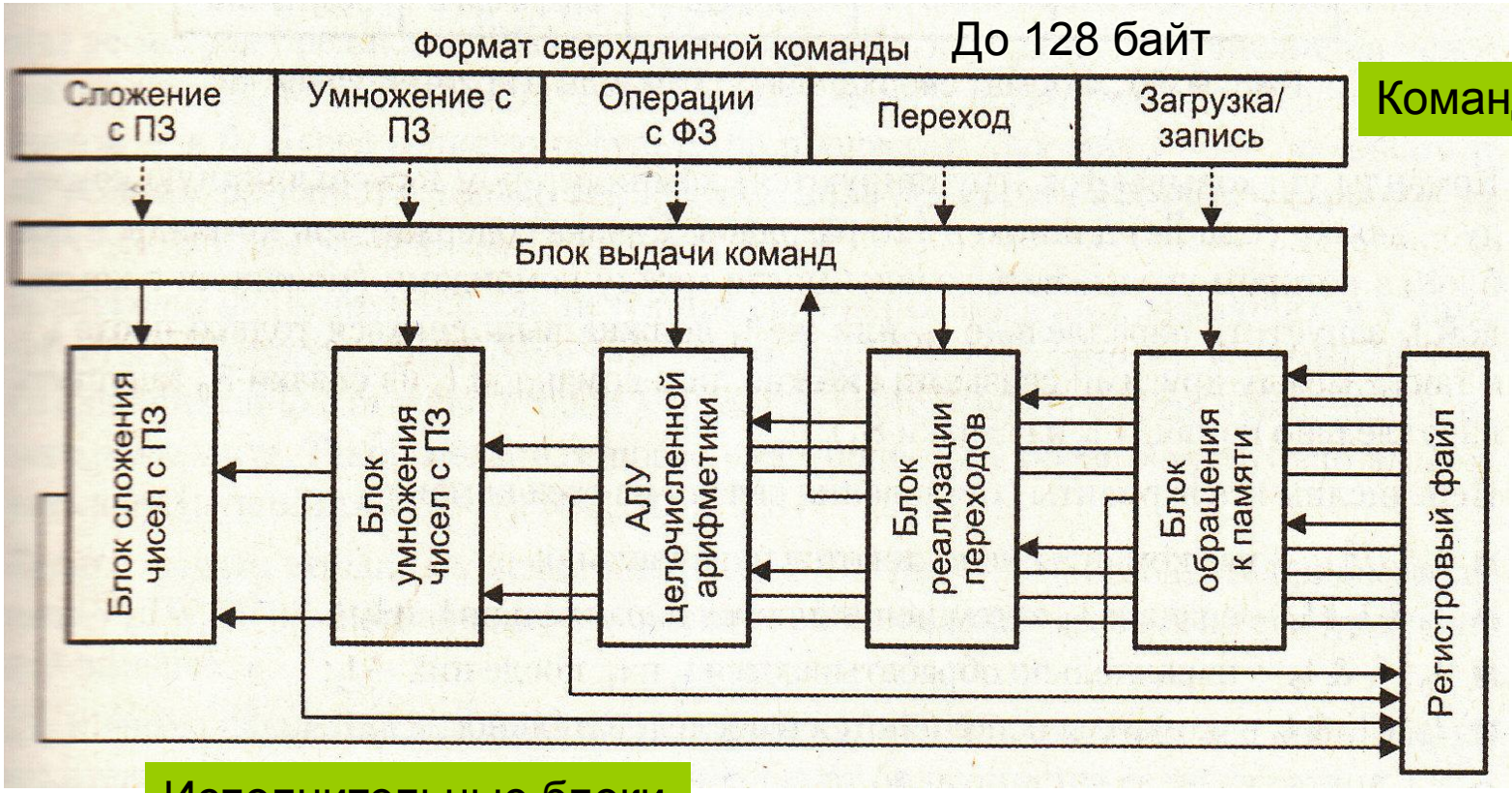
Формат связки команд Itanium

- **Компилятор формирует связки команд длинной 128 бит.**



Взаимосвязь полей команды VLIW с исполнительными блоками

Статическая суперскалярная архитектура – одно из названий VLIW процессоров



VLIW процессоры

- **Ядро МП использует команды длиной от 64 до 128 бит.**

Компания Transmeta TM5400 для мобильных приложений.

Процессоры семейства Crusoe

Характеристики	TM3120	TM5400
Тактовая частота, МГц	333–400	500–700
Кэш первого уровня, Кбайт	96	128
Кэш второго уровня, Кбайт	нет	256
Площадь кристалла, мм²	77	73
Напряжение питания, В	1,5	1,2–1,6
Технологический уровень, мкм	0,22	0,18
Тип корпуса	474-выводной BGA	474-выводной BGA

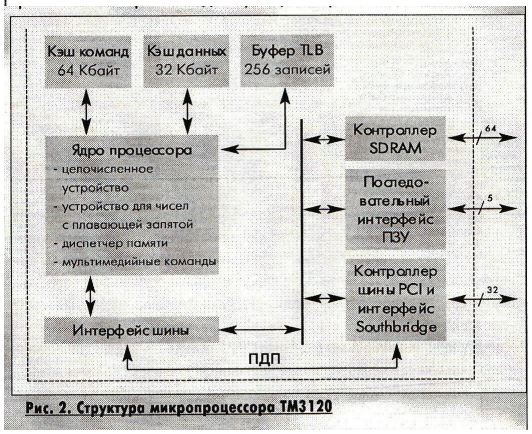


Рис. 2. Структура микропроцессора TM3120