

# Вычислительные системы.

## Параллельная обработка

### информации

1. Предпосылки возникновения и принципы построения ВС
2. Архитектурные свойства ВС
3. Назначение, область применения, классификация ВС
4. Уровни комплексирования в ВС
5. Параллельная обработка информации
6. Классификация систем параллельной обработки (Флинна)

# 1. Предпосылки возникновения и принципы построения ВС

## Закономерности развития ЭВМ:

- доминирующая роль классической структуры ЭВМ;
- неуклонный рост производительности (быстродействия) и интеллектуальности вычислительных средств;
- комплексное совершенствование ЭВМ (элементно-конструкторская база, структурно-аппаратурные решения, системно-программный и пользовательский, алгоритмический уровни);
- кризис классической структуры ЭВМ.

- Развитие элементарно-конструкторской базы, основанное на физических принципах электроники, всегда отстает от требований конкретно решаемых задач. Быстродействия одного компьютера недостаточно для, например, массового обслуживания группы перехватчиков, стартующих против налета баллистических целей.
- Своевременно пришло понимание того, что **только структурными методами** можно добиться необходимой производительности вычислительных средств. Таким основным структурным методом является **распараллеливание вычислений**.

Термин **ВС** появился в 60-х годах  
(ЭВМ III поколения):

- разделение процессов обработки информации и ее ввода-вывода;
- множественный доступ и коллективное использование вычислительных ресурсов в пространстве и во времени;
- сложные режимы работы ЭВМ - многопользовательская и многопрограммная обработка.

- Под вычислительной системой (ВС) понимают совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенную для подготовки и решения задач пользователей.
- Отличие ВС от ЭВМ - наличие в них нескольких вычислителей, реализующих параллельную обработку.

# Предпосылки возникновения ВС

## 1. Экономические

- Закон Гроша для ЭВМ

$$C_{\text{ЭВМ}} = K_1 * \Pi_{\text{ЭВМ}}^2$$

Стоимость, производительность

- Закон Гроша для ВС

$$C_{\text{ВС}} = K_2 * \sum \Pi_i$$

$K_1, K_2$  - коэф. пропорциональности

# Предпосылки возникновения ВС

## 2. Технические

- ↑ надежности,
- ↑ достоверности результатов обработки,
- возможность резервирования,  
централизации хранения и обработки  
данных,
- децентрализации управления и т.д.

# Основные принципы построения ВС

- возможность работы в разных режимах;
- **модульность структуры** технических и программных средств;
- **близкодействие;**
- унификация и стандартизация технических и программных решений;
- иерархия в организации управления процессами;
- обеспечение необходимым сервисом пользователей.

**Модульность** – принцип, определяющий формирование ВС из унифицированных элементов (модулей), которые функционально и конструктивно закончены, имеют средства сопряжения с другими элементами и разнообразие которых составляет полный набор.

Модуль должен иметь средства автономного управления, располагать арифметико-логическим устройством и памятью и содержать локальный коммутатор схему для связи с другими модулями (элементарный процессор (ЭП) или элементарная машина (ЭМ)).

**Близкодействие** – принцип построения ВС, обуславливающий такую организацию информационных взаимодействий между модулями–вычислителями, при которой каждый из них может непосредственно (без “посредников”) обмениваться информацией с весьма ограниченной частью модулей-вычислителей.

Принцип близкодействия допускает реализацию механизма управления ВС, не зависящий от числа составляющих ее вычислителей, т.к. поведение каждого вычислителя зависит от поведения только ограниченного множества других вычислителей системы (своих соседей).

ВС, основанные на принципах  
модульности и близкодействия,  
удовлетворяют также требованиям

- асинхронности,
- децентрализованности,
- распределенности.

**Асинхронность** функционирования ВС обеспечивается, если порядок срабатывания ее модулей определяется не с помощью вырабатываемых тем или иным образом отметок времени, а достижением заданных значений определенных (как правило, логических) функций.

Использование асинхронных схем позволяет достичь в системе алгоритмически предельного быстродействия: модули ВС срабатывают немедленно после достижения соответствующего условия.

Применение асинхронных схем обмена информацией между вычислителями позволяет не учитывать разброс в их тактовых частотах и колебания времени задержки сигналов в линиях связи.

**Децентрализованность** управления ВС - в системе нет выделенного модуля, который функционирует как единый для всей системы центр управления.

Децентрализованное управление системой позволяет:

- достичь живучести ВС, т. е. ее способности продолжать работу при отказах модулей;
- избежать очередей при обслуживании “заявок” на управление.

Под ресурсами ВС понимаются все объекты, которые запрашиваются, используются и освобождаются в ходе выполнения вычислений (процессоры модули ОП, ВУ, линии межмодульных связей, шины, файлы данных, компоненты программного обеспечения).

***Распределенная ВС*** - система, в которой каждый ресурс рассматривается как общий, доступный любому потребителю

## 2. Архитектурные свойства ВС

1. Масштабируемость
2. Универсальность
3. Производительность
4. Реконфигурируемость (статическая и динамическая)
5. Надёжность и живучесть (***способность реализации параллельных программ, допускающих варьирование числа ветвей в известных пределах***).
6. Самоконтроль и самодиагностика
7. Технико-экономическая эффективность

### 3. Назначение, область применения, классификация ВС

Использование МВС:

Традиционное -

для  
вычислительных  
задач

В настоящее время -

1. обработка транзакций в режиме реального времени (OLTP)
2. хранилищ данных для организации систем поддержки принятия решений (Data Mining) и Т.Д.

# «Grand challenges»- круг фундаментальных и прикладных проблем, решаемых с помощью ВС

- предсказания погоды, климата и глобальных изменений в атмосфере;
- науки о материалах;
- построение полупроводниковых приборов;
- сверхпроводимость;
- структурная биология;
- разработка фармацевтических препаратов;
- генетика;
- квантовая хромодинамика;
- астрономия;

«Grand challenges»- круг фундаментальных и прикладных проблем, решаемых с помощью ВС

- транспортные задачи;
- гидро- и газодинамика;
- управляемый термоядерный синтез;
- эффективность систем сгорания топлива;
- геоинформационные системы;
- разведка недр;
- наука о мировом океане;
- распознавание и синтез речи;
- распознавание изображений.

# Наиболее распространенные типы МВС:

- СИСТЕМЫ ВЫСОКОЙ НАДЕЖНОСТИ;
- СИСТЕМЫ ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ;
- МНОГОПОТОЧНЫЕ СИСТЕМЫ.

# Классификация ВС

- **По назначению:**
  - универсальные
  - специализированные
- **По типу ЭВМ или процессоров**
  - однородные
  - неоднородные
- **По степени территориальной разобщенности**
  - сосредоточенные
  - распределенные

# Классификация ВС

- По методам управления элементами ВС
  - централизованные
  - децентрализованные
  - со смешанным управлением
- По принципу закрепления вычислительных функций за отдельными модулями
  - с жестким
  - плавающим закреплением функций
- По режиму работы ВС
  - работающие в оперативном режиме
  - работающие в неоперативном временном режиме

## 4. Уровни комплексирования в ВС

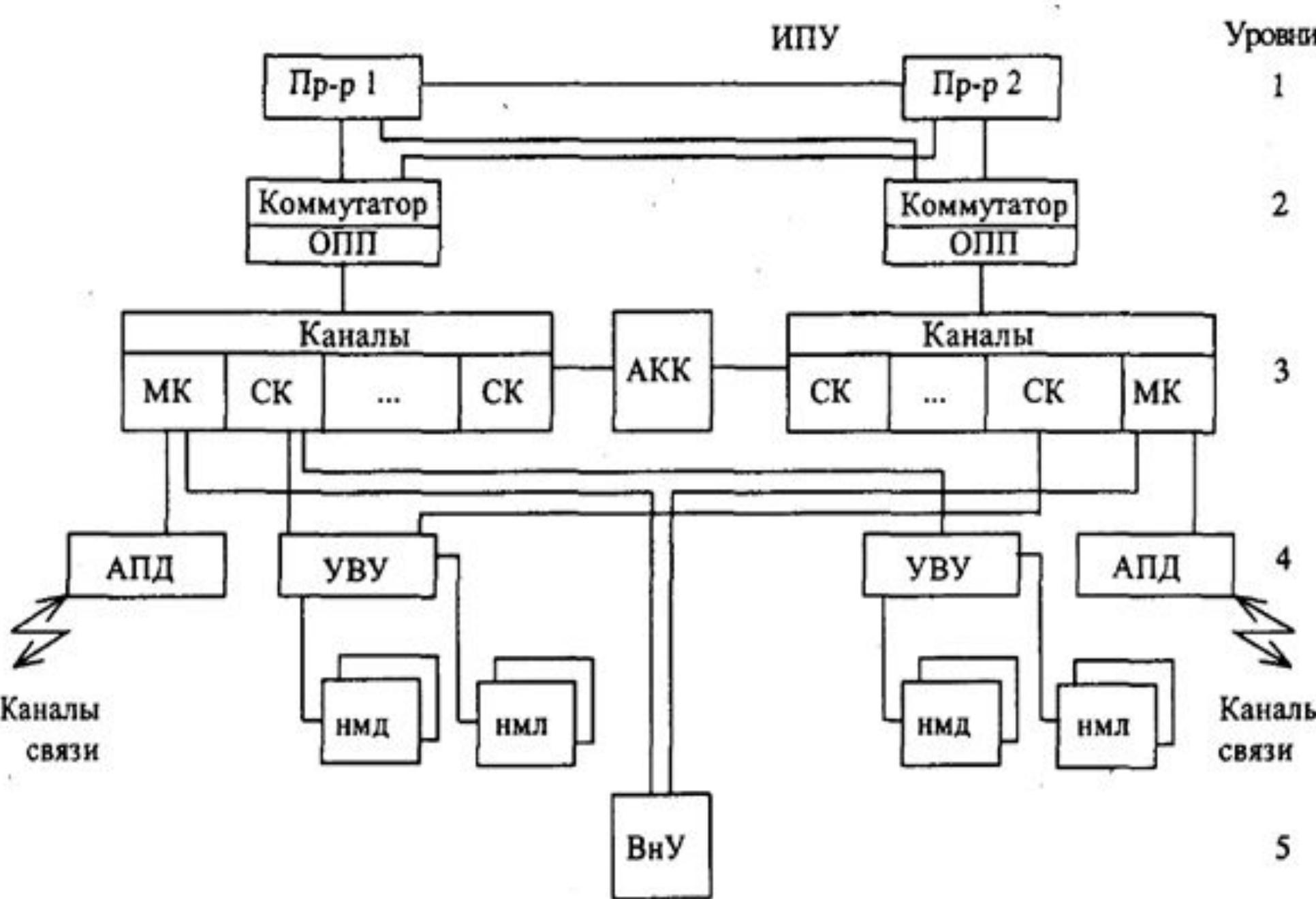
Совместимость комплекслируемых в ВС элементов

предполагает:

- Аппаратную или техническую (унифицированные средства соединения + параметры электрических сигналов + алгоритмы взаимодействия)
- Программную (передаваемые программы должны быть правильно поняты и выполнены другим устройством);
- Информационную (информационные массивы будут одинаково интерпретироваться)

совместимость.

- В создаваемых ВС стараются обеспечить несколько путей передачи данных, что позволяет достичь необходимой надежности функционирования, гибкости и адаптируемости к конкретным условиям работы.
- Сочетание различных уровней и методов обмена данными между модулями ВС в наиболее полной форме нашло свое выражение в универсальных суперЭВМ и больших ЭВМ, в которых сбалансировано использовались все методы достижения высокой производительности.



## Уровни мультиплексирования:

- прямого управления (процессор — процессор);
- общей оперативной памяти;
- комплексируемых каналов ввода-вывода;
- устройств управления внешними устройствами (УВУ);
- общих внешних устройств.

## Уровень прямого управления (передача коротких однобайтовых приказов-сообщений)

1. Процессор-инициатор обмена по интерфейсу прямого управления (ИПУ) передает в блок прямого управления байт-сообщение и подает команду «Прямая запись».
2. У другого процессора эта команда вызывает внешнее прерывание, вырабатывается команда «Прямое чтение» и передаваемый байт записывается в его память.
3. Принятая информация расшифровывается и по ней принимается решение.
4. Прерывания снимаются, процессоры продолжают работу.

Уровень общей оперативной памяти (ООП) (для оперативного взаимодействия процессоров)

- эффективно работает только при небольшом числе обслуживаемых абонентов;
- широко используется в многопроцессорных серверах ВС.

## Уровень комплексизируемых каналов ввода-вывода

- предназначается для передачи больших объектов информации между блоками оперативной памяти, сопрягаемых ЭВМ;
- обмен данными между ЭВМ осуществляется с помощью адаптера «канал-канал» (АКК) и команд «Чтение» и «Запись».

Уровень устройств управления вн. устройствами

- предполагает использование встроенного в УВУ двухканального переключателя и команд «Зарезервировать» и «Освободить»;
- Двухканальный переключатель позволяет подключать УВУ одной машины к селекторным каналам различных ЭВМ;
- по команде «Зарезервировать» канал-инициатор обмена имеет доступ через УВУ к любым ВУ.

## Уровни комплексирования позволяют создавать различные структуры ВС

- 2-й + 1-й → многопроцессорные ВС
- 1, 3, 4, 5 → обеспечивают построение разнообразных машинных комплексов
- 3 + 4 → особенно часто используется
- 5-й → используется в редких специальных случаях (уникальное оборудование)

## 5. Параллельная обработка информации

Способы ↑ производительности ЭВМ:

- а) совершенствование алгоритмов решения задач;
- б) оптимизация программ;
- в) ↑ быстродействия элементной базы ЭВМ;
- г) модификация структуры процессора;
- д) конвейерно-параллельная обработка информации.

Компьютер	Время такта(с)	Скорость (оп/с)
EDSAC (1949 г.)	$2 \cdot 10^{-6}$ с	$10^2$ оп/с
CRAY C90 (1991 г.)	$4 \cdot 10^{-9}$ с	$10^9$ оп/с

Производительность  $\uparrow$  в 10 млн раз:

- Уменьшение времени такта (с 2мкс до 4нс) дает вклад лишь в 500 раз.
- Остальное – за счет новых решений архитектуры (*принцип параллельной обработки данных*).

## Разновидности параллельной обработки данных:

- **Параллельная обработка:**
- устройство выполняет одну операцию за единицу времени, то тысячу операций оно выполнит за тысячу единиц;
- система из  $N$  устройств ту же работу выполнит за  $1000/N$  единиц времени.

## • Конвейерная обработка

операция = 5 микроопераций, выполняемых за одну единицу времени;

одно неделимое устройство 100 пар аргументов оно обрабатывает за 500 единиц;

Если каждую микрооперацию выделить ступень конвейерного устройства, то на пятой единице времени на разной стадии обработки будут находиться первые пять пар аргументов, а все данные будут обработаны за  $5+99=104$  единицы времени - ускорение по сравнению с последовательным устройством почти в пять раз (по числу ступеней конвейера).

- **Конвейерная обработка**

Идея - выделение отдельных этапов выполнения общей операции, причем каждый этап, выполнив свою работу, передает результат следующему, одновременно принимая новую порцию входных данных.

***Почему не заменить конвейер обычным параллелизмом?***

**Параллелизм** – основа высокопроизводительной работы всех подсистем вычислительных машин:

- организация памяти любого уровня иерархии,
  - организация системного ввода/вывода,
  - организация мультиплексирования шин и т.д.
- базируются на принципах параллельной обработки запросов.
- Современные ОС являются многозадачными и многопользовательскими, имитируя параллельное исполнение программ посредством механизма прерываний.

- конвейер Intel Pentium 4 состоял из 20 ступеней;
- в 80-е годы - параллелизм уровня команд посредством размещения в CPU сразу нескольких конвейеров.

# Уровни параллелизма по степени гранулярности (отношение объёма вычислений к объёму коммуникаций)



Во всем многообразии способов организации параллельной обработки можно выделить направления:

- совмещение во времени различных этапов разных задач (мультипрограммная обработка);
- одновременное решение различных задач или частей одной задачи;
- конвейерная обработка.

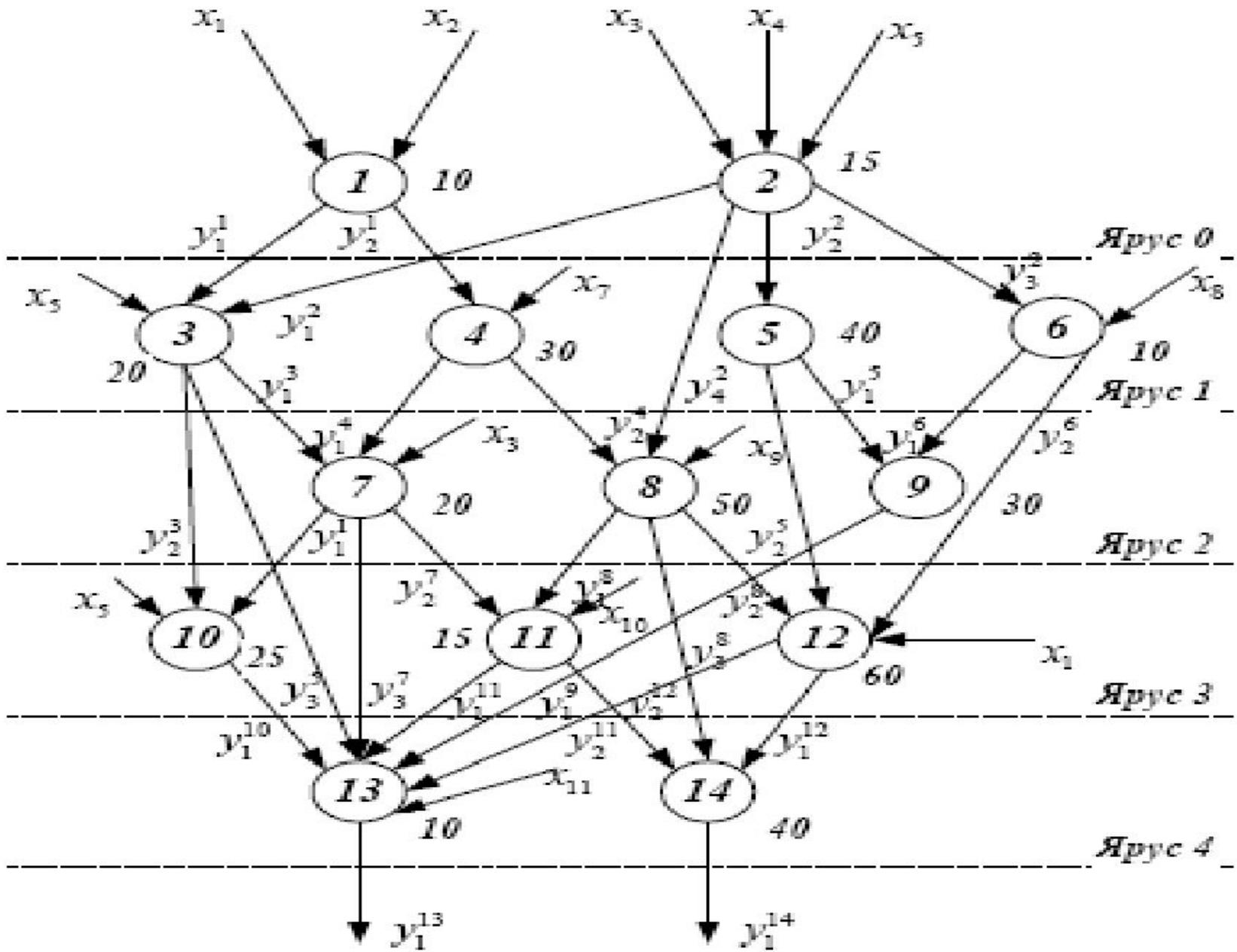
# Виды параллелизма

- Естественный параллелизм независимых задач
- Параллелизм независимых ветвей
- Параллелизм объектов или данных

Двумя независимыми ветвями программы считают такие части задачи, при выполнении которых выполняются следующие условия:

- ни одна из входных для ветви программы величин не является выходной величиной другой программы (отсутствие функциональных связей);
- для обеих ветвей программы не должна производиться запись в одни и те же ячейки памяти (отсутствие связи по использованию одних и тех же полей ОП);
- условия выполнения одной ветви не зависят от результатов или признаков, полученных при выполнении другой ветви (независимость по управлению);
- обе ветви должны выполняться по разным блокам программы (программная независимость).

# Параллелизм независимых ветвей



Если для исполнения  $i$ -ой ветви требуется  $t_i$  единиц времени, то для исполнения всей программы потребуется

$$T = \sum_{i=1}^{N=14} t_i = 435$$

единиц времени

## Варианты исполнения 2-мя процессорами

1. Процессор 1 - ветви 1-4-5-9-10-13,  
процессор 2 – ветви 2-6-3-7-8-11-12-14.
2. Процессор 1 ветви 1-4-5-9-10-11-13,  
процессор 2 - ветви 2-6-3-7-8-12-14.
3. Процессор 1 ветви 1-4-8-12-11-13,  
процессор 2 – ветви 2-5-6-3-7-9-10-14

Во всех случаях время, через которое двухпроцессорная система выдает результаты, существенно сокращается: вместо 435 единиц времени результаты выдаются:

- в 1 варианте через 260,
- во 2 – через 245
- в 3 – через 235 единиц времени, т. е. в последнем случае время решения задачи уменьшается в 1,85 раза

## Параллелизм объектов или данных

- По одной и той же программе должна обрабатываться некоторая совокупность данных, поступающих в систему одновременно;
- Векторизация или распараллеливание выполняется уже на этапе компиляции;
- Обработкой данных управляет одна программа;
- Пространство имен является глобальным;
- Параллельные операции над элементами массива выполняются одновременно на всех доступных данной программе процессорах.

# Конвейерная обработка

эффективна при выполнении следующих условий:

- система выполняет повторяющуюся операцию;
- эта операция может быть разделена на независимые части, степень перекрытия которых невелика;
- трудоемкость подопераций примерно одинакова.

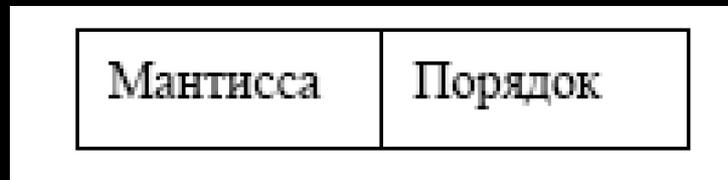
Количество подопераций называют *глубиной конвейера*.

Важным условием нормальной работы конвейера является отсутствие конфликтов, то есть данные, подаваемые в конвейер, должны быть независимыми.

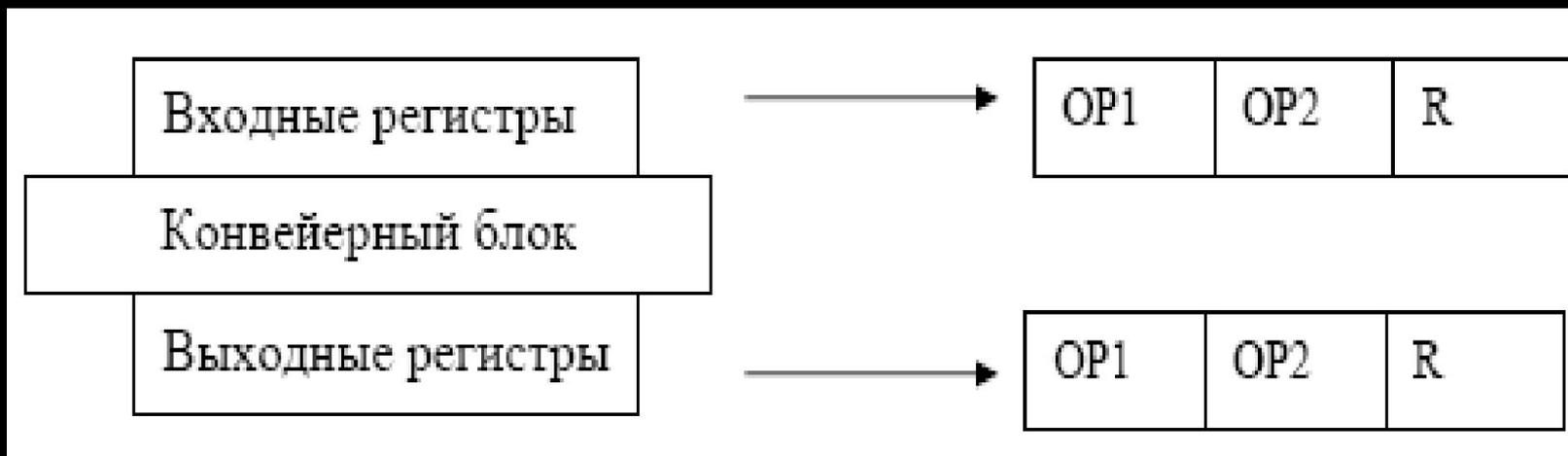
В том случае, когда очередной операнд зависит от результата предыдущей операции, возникают такие периоды работы конвейера ("пузыри"), когда он пуст.

# Конвейерная обработка

## Команда умножения чисел с ПЗ



# Структура сегмента конвейера данных



$T$  – время прохождения одной пары операндов  
 $n$ -звенный конвейер

$N$  – число пар операндов

на обработку  $N$  операций потребуется время

$$(n + N) \cdot (T/n)$$

На одну операцию потребуется время

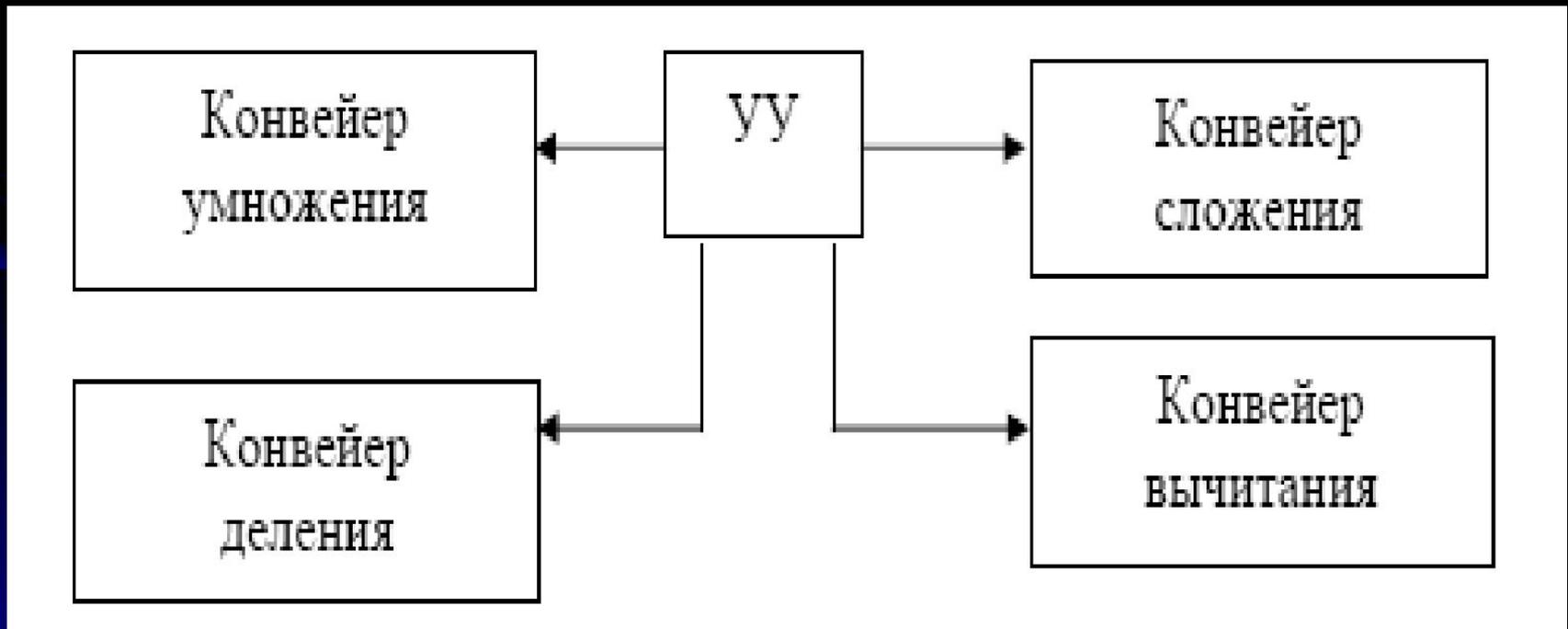
$$((n + N) \cdot T) / (n \cdot N)$$

# Конвейер команд

<i>Этап</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
<b>ВК</b>	К1	К2	К3	К4	К5	К6	К7	К8
<b>ДК</b>		К1	К2	К3	К4	К5	К6	К7
<b>ВО<sub>1</sub></b>			К1	К2	К3	К4	К5	К6
<b>ВО<sub>2</sub></b>				К1	К2	К3	К4	К5

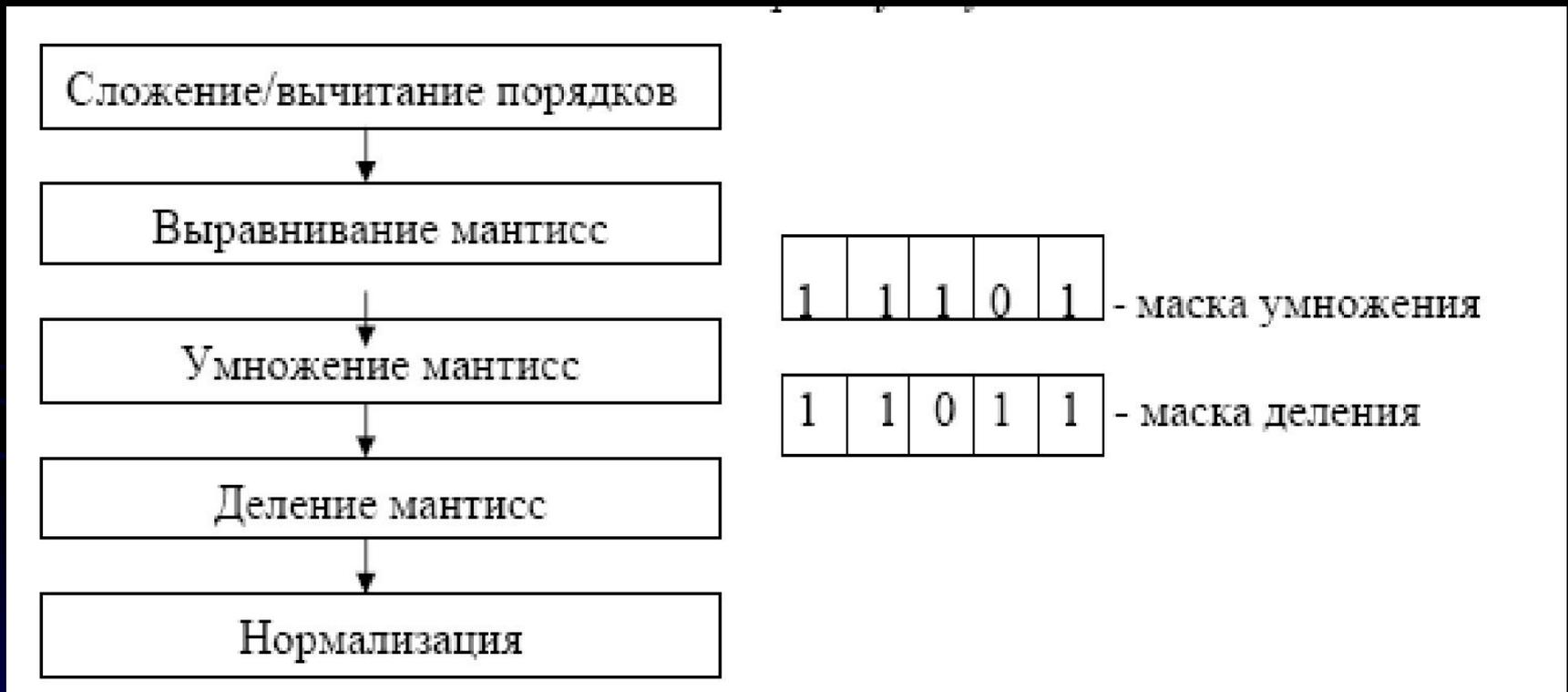
# Многооперационные конвейеры

- Конвейер в ширину (аппаратная реализация каждой операции в виде набора конвейерных сегментов)



# Многооперационные конвейеры

- Конвейер в глубину



## Проблемы конвейерных процессоров:

- Стоимость аппаратных средств
- Обработка особых ситуаций
- Синхронизация (УУ должно выполнять синхронизацию конвейерных блоков, конвейера команд и конвейера данных).

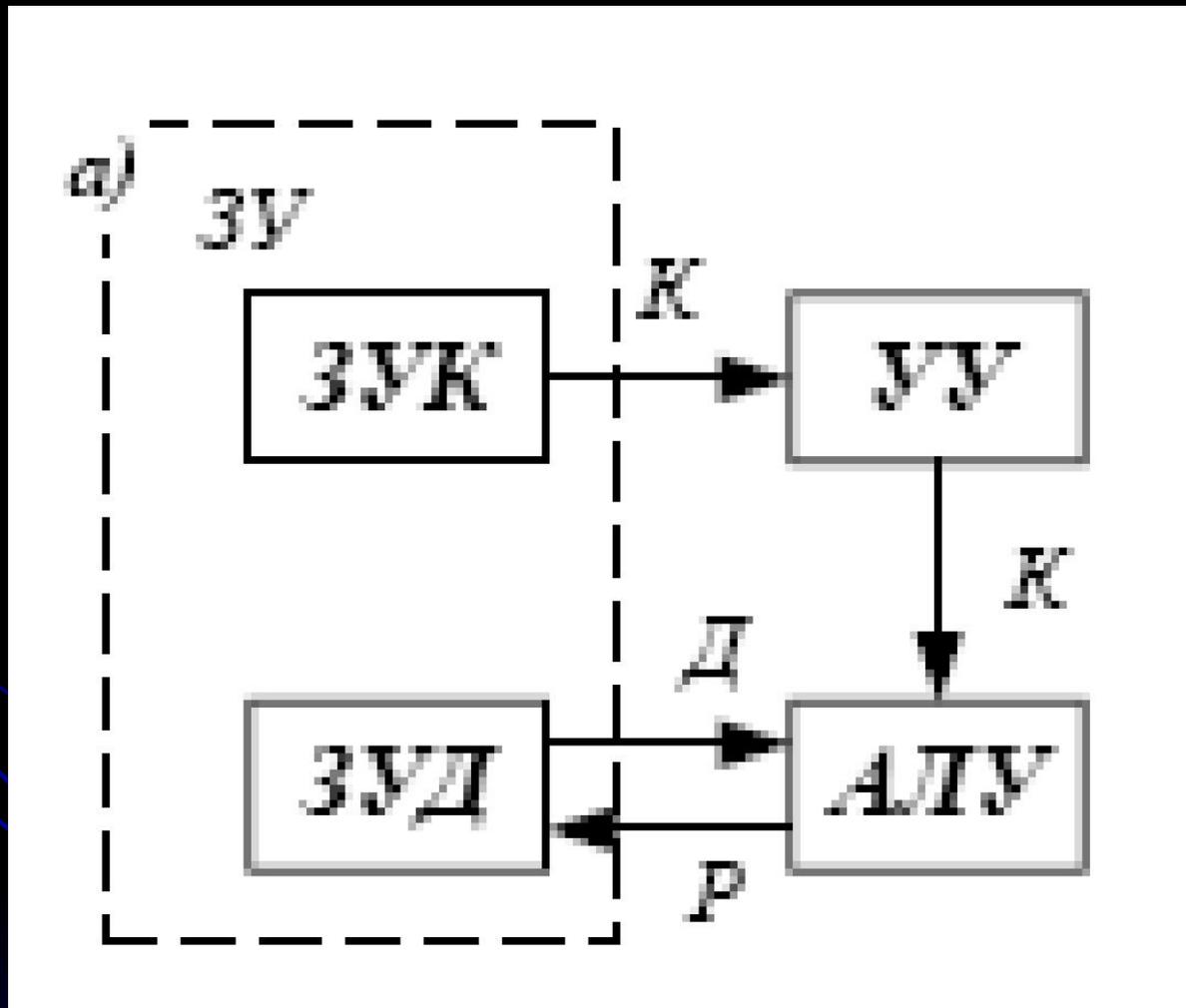
## 6. Классификация систем параллельной обработки

- Под архитектурой ВС можно понимать и способ параллельной обработки данных, используемый в системе, и организацию памяти, и топологию связи между процессорами, и способ исполнения системой арифметических операций.
- 1966 М. Флинн - понятие потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором.

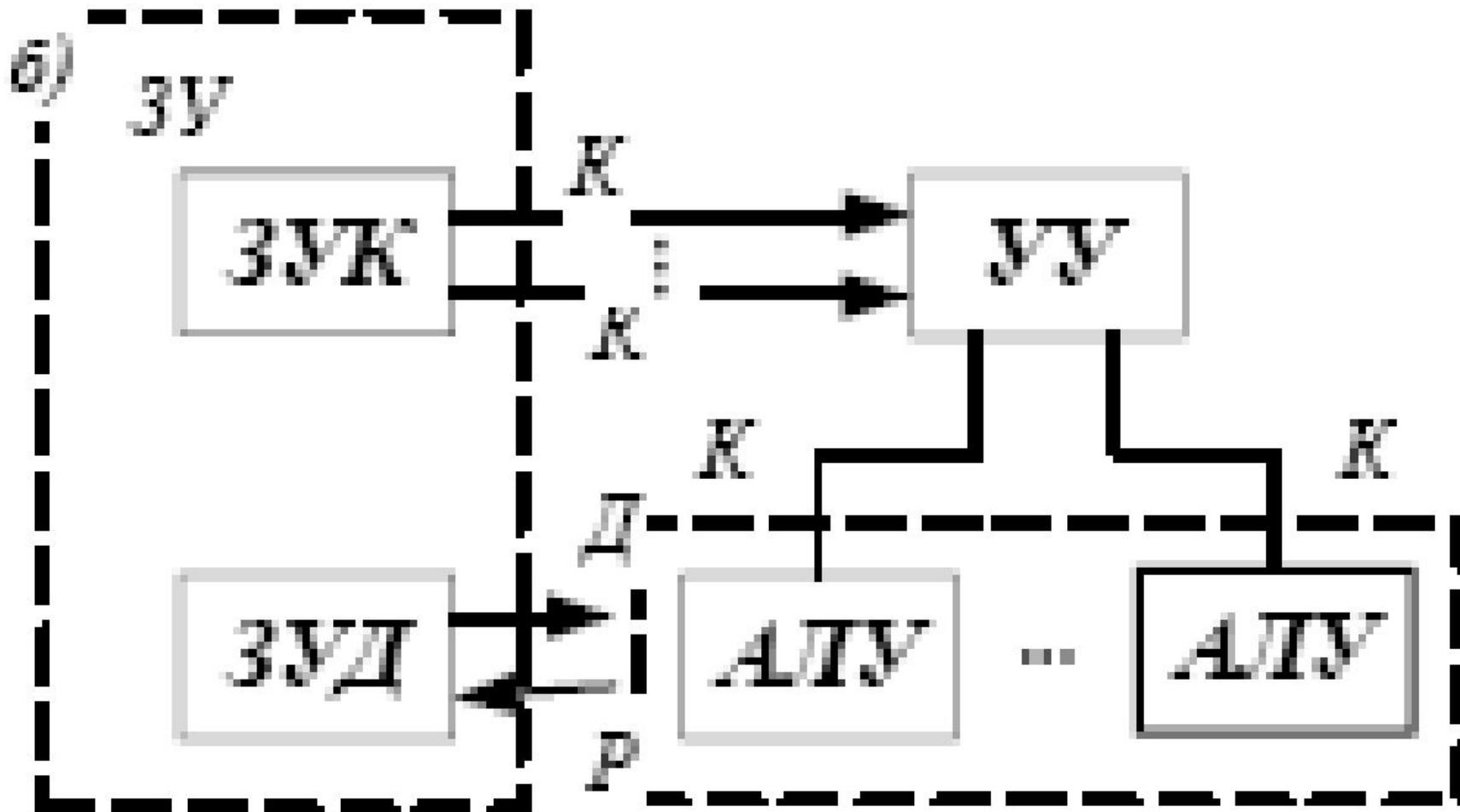
# Классификация Флинна

- ОКОД (SISD = Single Instruction Single Data)
- МКОД (MISD = Multiple Instruction Single Data)
- ОКМД (SIMD = Single Instruction Multiple Data)
- МКМД (MIMD=Multiple Instruction Multiple Data)

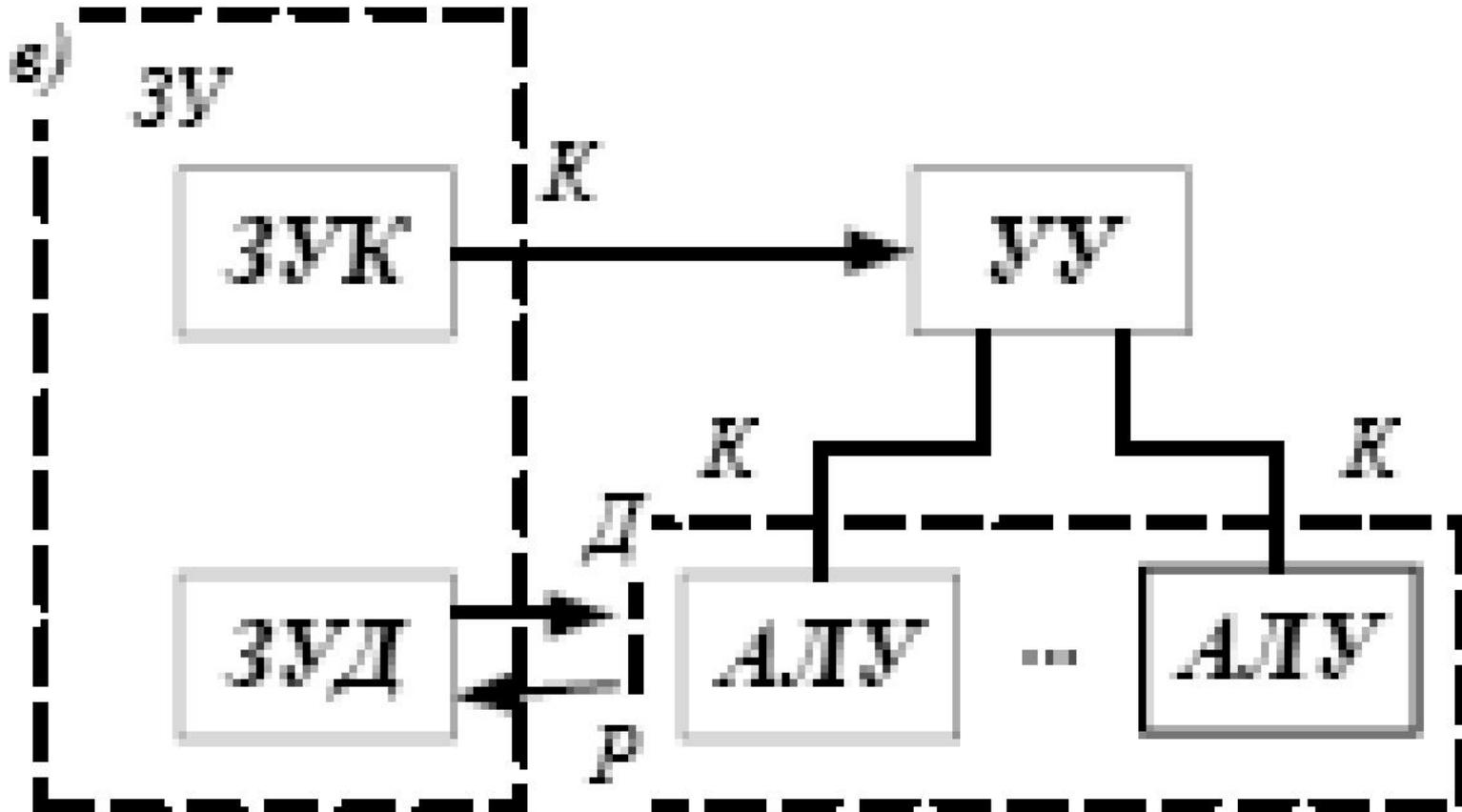
# Системы класса SISD



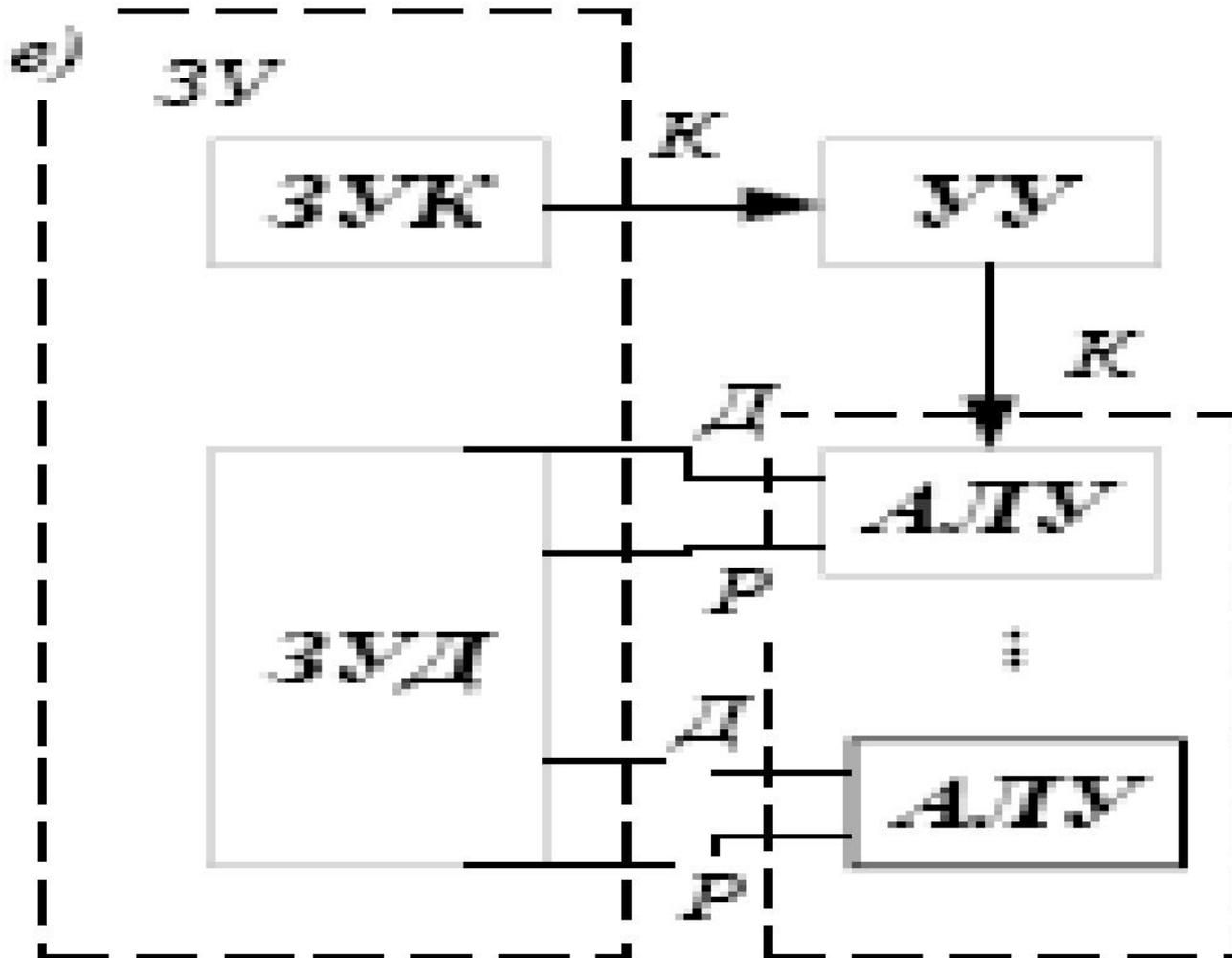
# Системы класса MISD



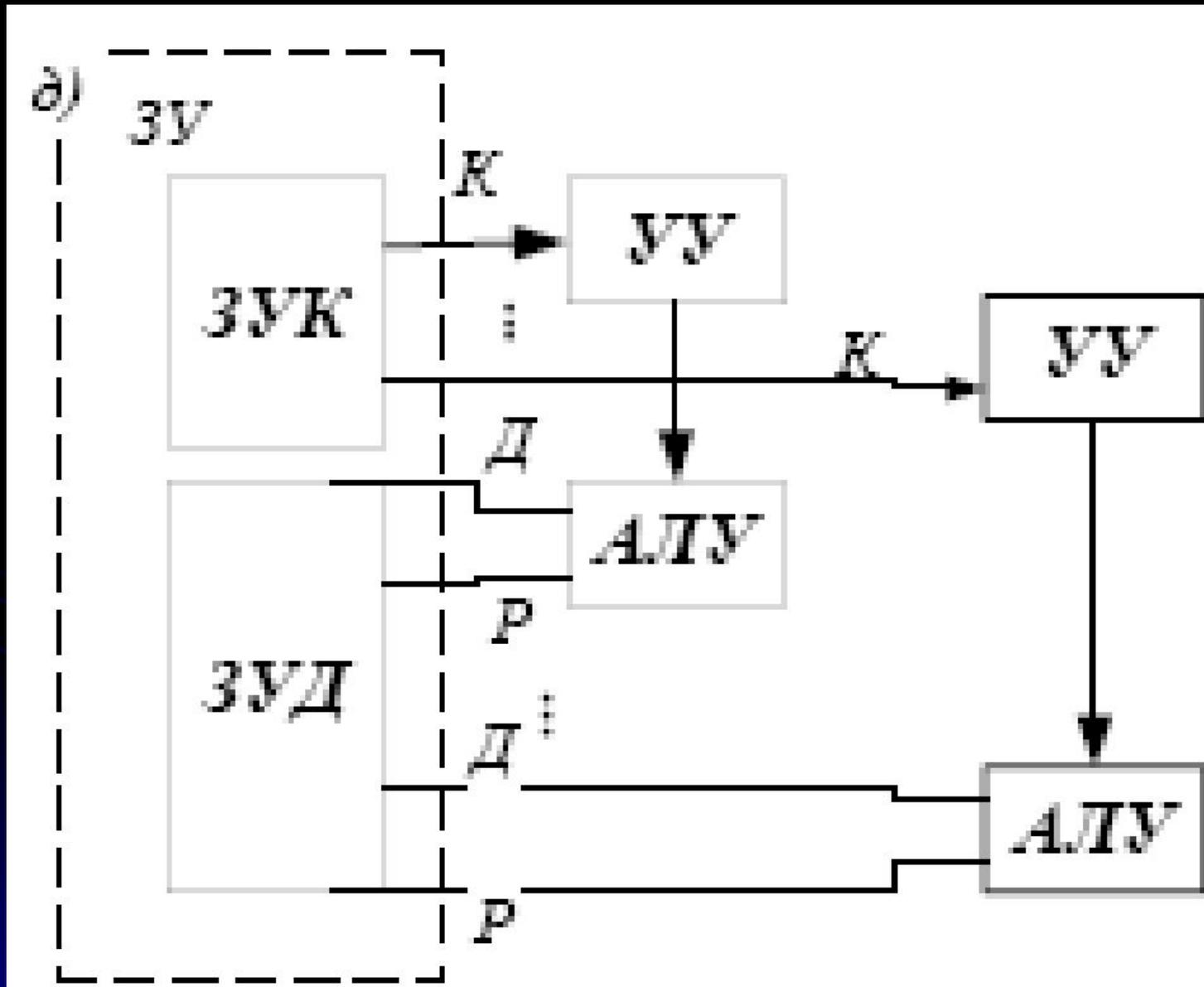
# Системы класса SIMD



# Системы класса SIMD



# Системы класса MIMD



# Дополнения Ванга и Бриггса

Класс SISD разбивается на два подкласса:

- архитектуры с единственным функциональным у-вом
- архитектуры с несколькими функциональными у-вами

В класс SIMD также вводится два подкласса:

- Арх-ры с пословно-послед. обработкой информации
- арх-ры с разрядно-последовательной обработкой

В классе MIMD различают

- ВС со слабой связью между процессорами (с распредел. памятью)
- ВС с сильной связью (системы с общей памятью)