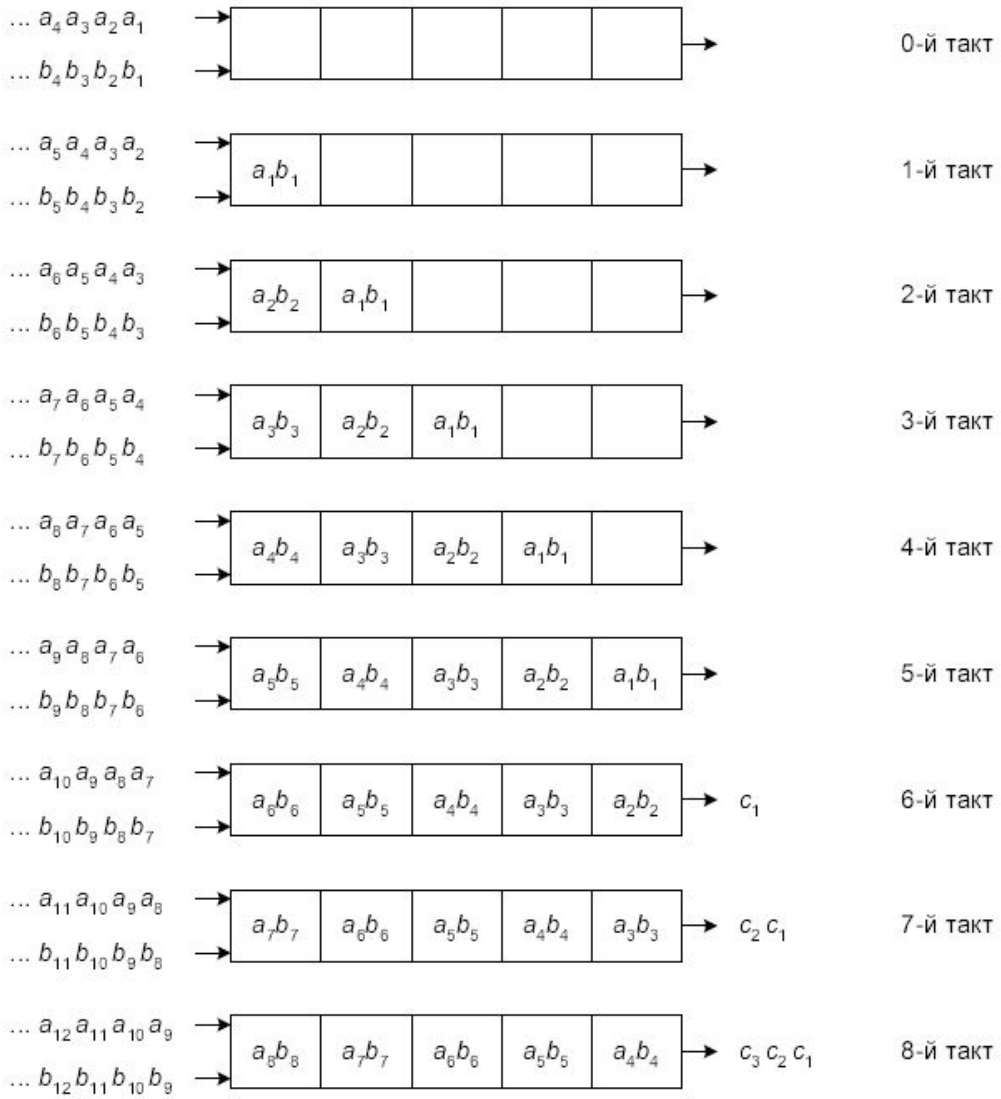
Лекция №4
по курсу
«Проектирование и архитектура вычислительных систем»

Москва, 2020

Конвейерная обработка



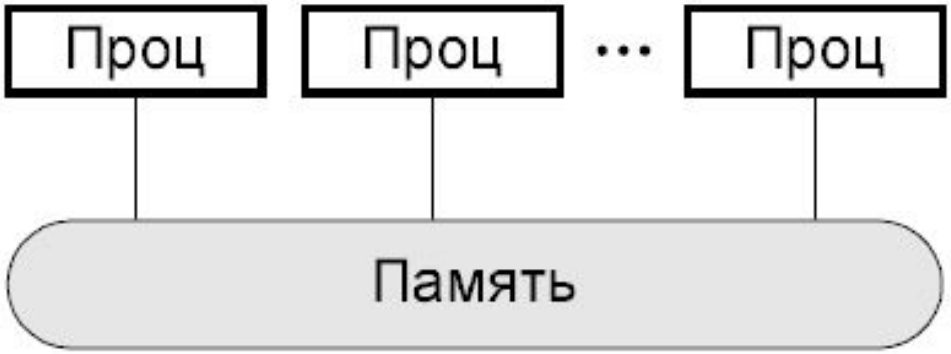
Разрядно-параллельная обработка

Как и прежде, будем считать, что конвейерное устройство состоит из l ступеней, срабатывающих за один такт. Два вектора из n элементов можно либо сложить одной векторной командой, либо выполнить подряд n скалярных команд сложения элементов этих векторов. Если n скалярных команд одна за другой исполняются на таком устройстве, то, согласно общему закону, они будут обработаны за $l + n - 1$ тактов.

- Поддержка параллелизма в аппаратно- программной среде вычислительной системы
- Спецпроцессоры для поддержки быстрого преобразования Фурье

Исмаилов Ш-М.А.

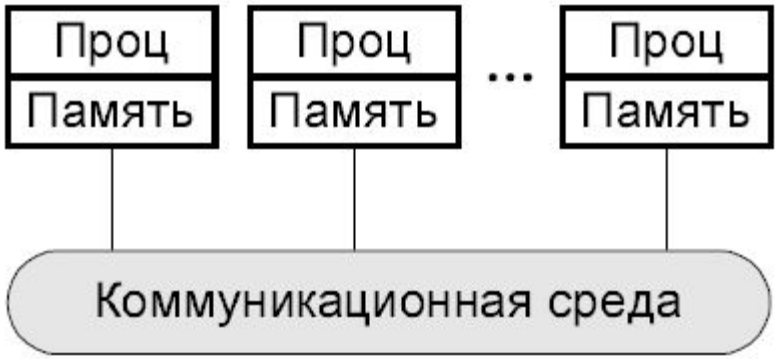
Параллельные компьютеры с общей памятью



Многопроцессорные системы
Symmetric multiprocessor

Каждый процессор может делать все что любой другой

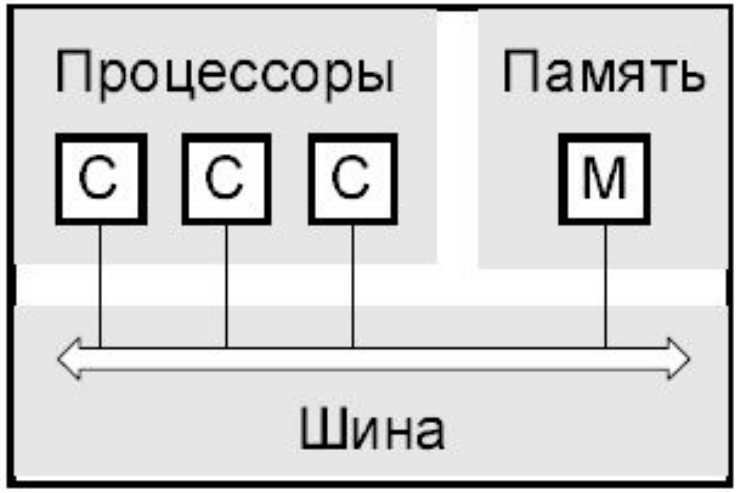
Параллельные компьютеры с распределенной памятью



Мультикомпьютерные системы

Пример:
Grid технология
Узлы сети блокчейн

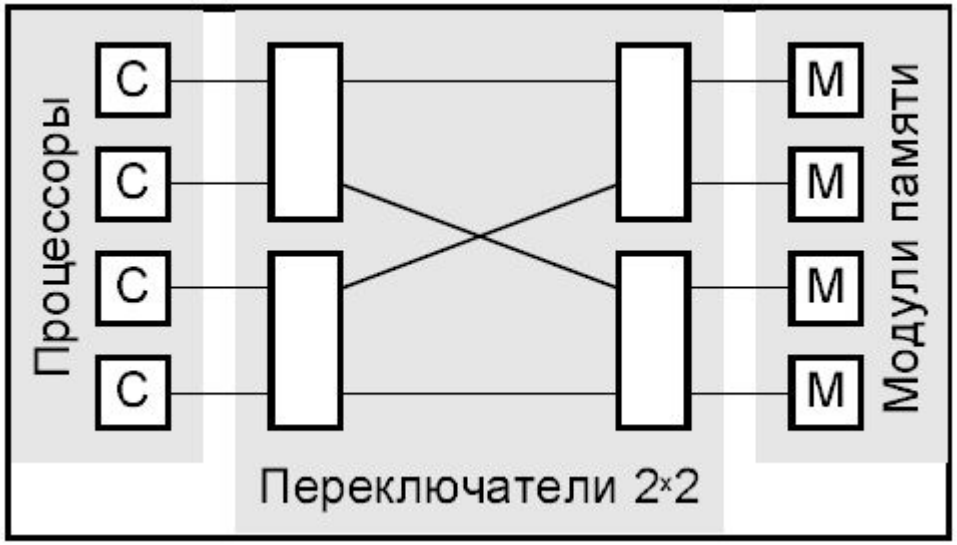
Мультипроцессорная система с общей шиной



4-5 устройств на шине приводит к потере производительности

Недостаток – большой объем необходимого оборудования – n^2 коммутаторов

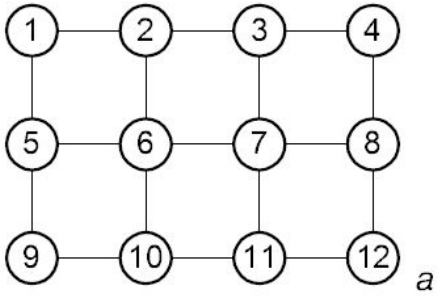
Мультипроцессорная система с омега сетью



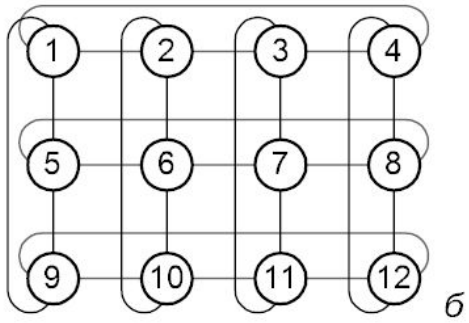
Каскадные переключатели.

Сеть из 4-х коммутаторов два на два

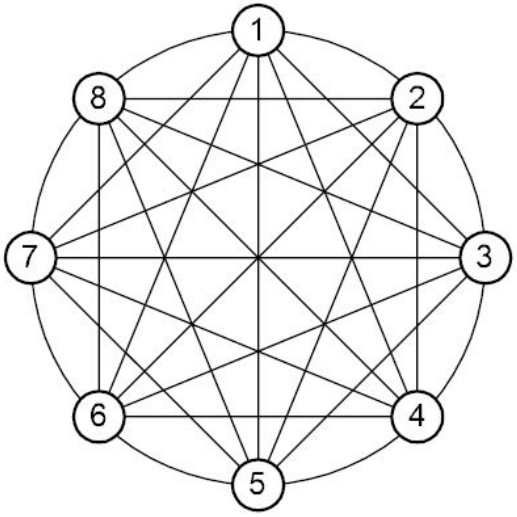
Варианты топологий связи процессоров



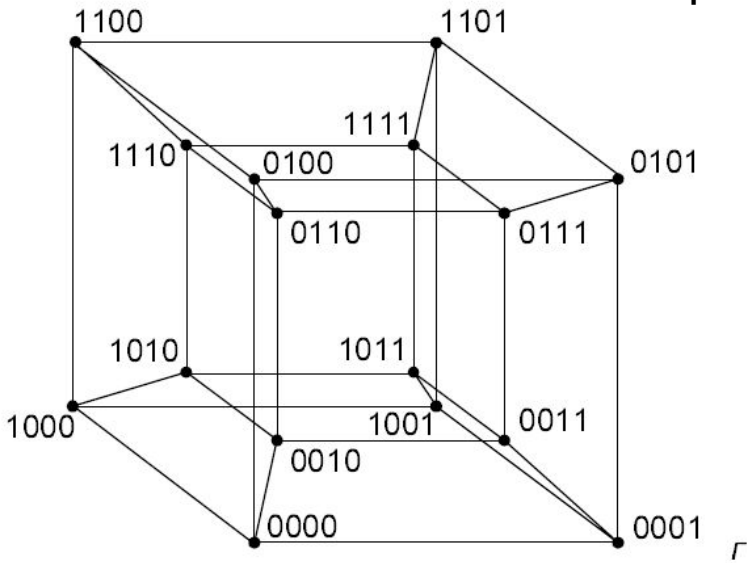
решетка



2-тор

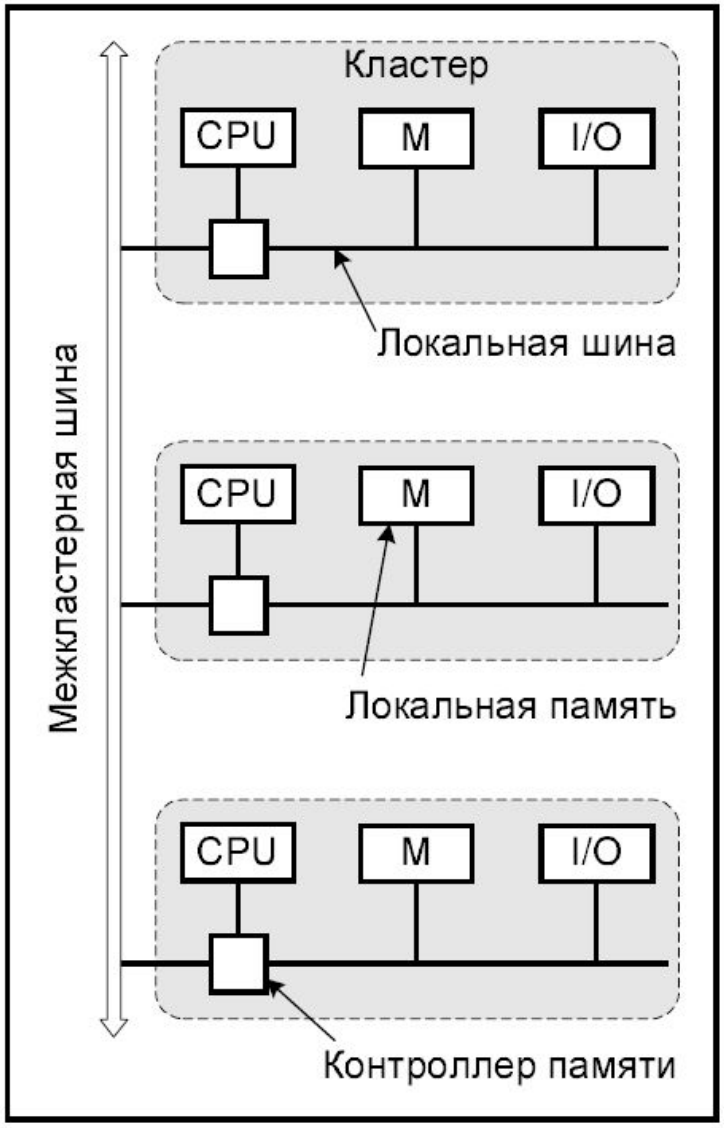


Полная связь



Гиперкуб

Кластер


Кластер
CPU
M
I/O
Локальная шина
CPU
M
I/O
Локальная память
CPU
M
I/O
Контроллер памяти
Межкластерная шина

беспокойства очень легко объяснить. Предположим, что процессор P_1 сохранил значение x в ячейке q, а затем процессор P_2 хочет прочитать содержимое той же ячейки q. Что получит процессор P_2? Конечно же, всем бы хотелось, чтобы он получил значение x, но как он его получит, если x попало в кэш процессора P_1? Эта проблема носит название проблемы согласования содержимого кэш-памяти *(cache coherence problem, проблема*

Кластер

Стоимость операции - время её реализации. Стоимость работы – сумма стоимостей всех выполненных операций.

Загруженность устройства = Отношение стоимости реально выполн. работы/ к максимально возможной

Реальной производительностью системы – кол-во операций реально выполненных за единицу времени

Пиковой производительностью - макс кол-во операций за единицу времени

Кластер

Ускорение (**speedup**), получаемое при использовании параллельного алгоритма для ***p*** процессоров, по сравнению с последовательным вариантом выполнения вычислений определяется величиной:

```
Sp(n)=T1(n)/Tp(n)
```

т.е. как отношение времени решения задач на скалярном процессоре(Оценка *T1* определяет время выполнения алгоритма при использовании одного процессора и представляет, тем самым, время выполнения последовательного варианта алгоритма решения задачи) к времени выполнения параллельного алгоритма (величина ***n*** применяется для параметризации вычислительной сложности решаемой задачи и может пониматься, например, как количество входных данных задачи).

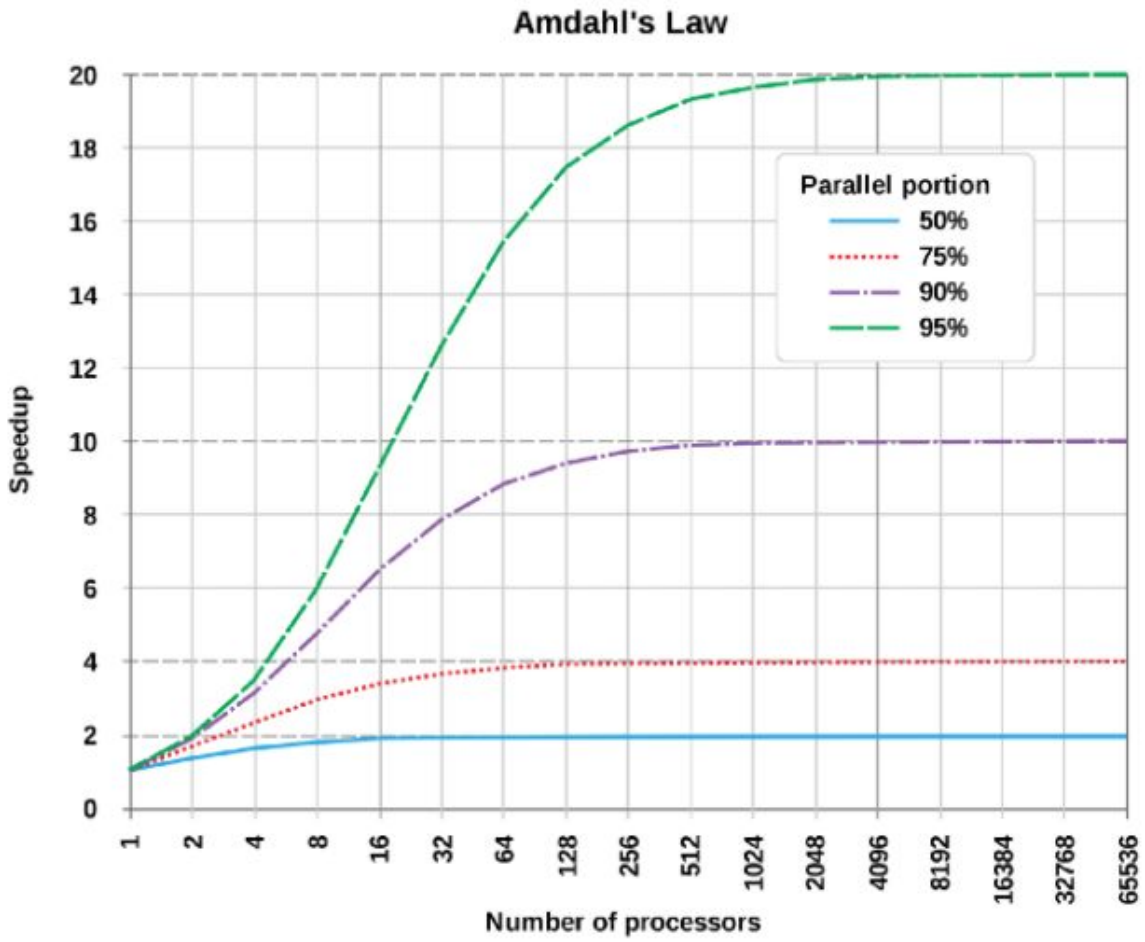
Кластер

Эффективность (**efficiency**) использования параллельным алгоритмом процессоров при решении задачи определяется соотношением:

```
Ep(n)=T1(n)/(pTp(n))=Sp(n)/p
```

Величина эффективности определяет среднюю долю времени выполнения алгоритма, в течение которой процессоры реально задействованы для решения задачи.

Закон Амдала



Закон Амдала

«В случае, когда задача разделяется на несколько частей, суммарное время её выполнения на параллельной системе не может быть меньше времени выполнения самого длинного фрагмента».
Согласно этому закону, ускорение выполнения программы за счёт распараллеливания её инструкций на множестве вычислителей ограничено временем, необходимым для выполнения её последовательных инструкций.

Загрузка процессоров

Закон Амдала

Пусть необходимо решить некоторую вычислительную задачу. Предположим, что её алгоритм таков, что доля a от общего объёма вычислений может быть получена только последовательными расчётами, а, соответственно, доля $1 - a$ может быть распараллелена идеально (то есть время вычисления будет обратно пропорционально числу задействованных узлов p). Тогда ускорение, которое может быть получено на вычислительной системе из p процессоров, по сравнению с однопроцессорным решением не будет превышать величины:

$$S_p = \frac{1}{\alpha + \frac{1-\alpha}{p}}$$

Таблица показывает, во сколько раз быстрее выполнится программа с долей последовательных вычислений a при использовании p процессоров.

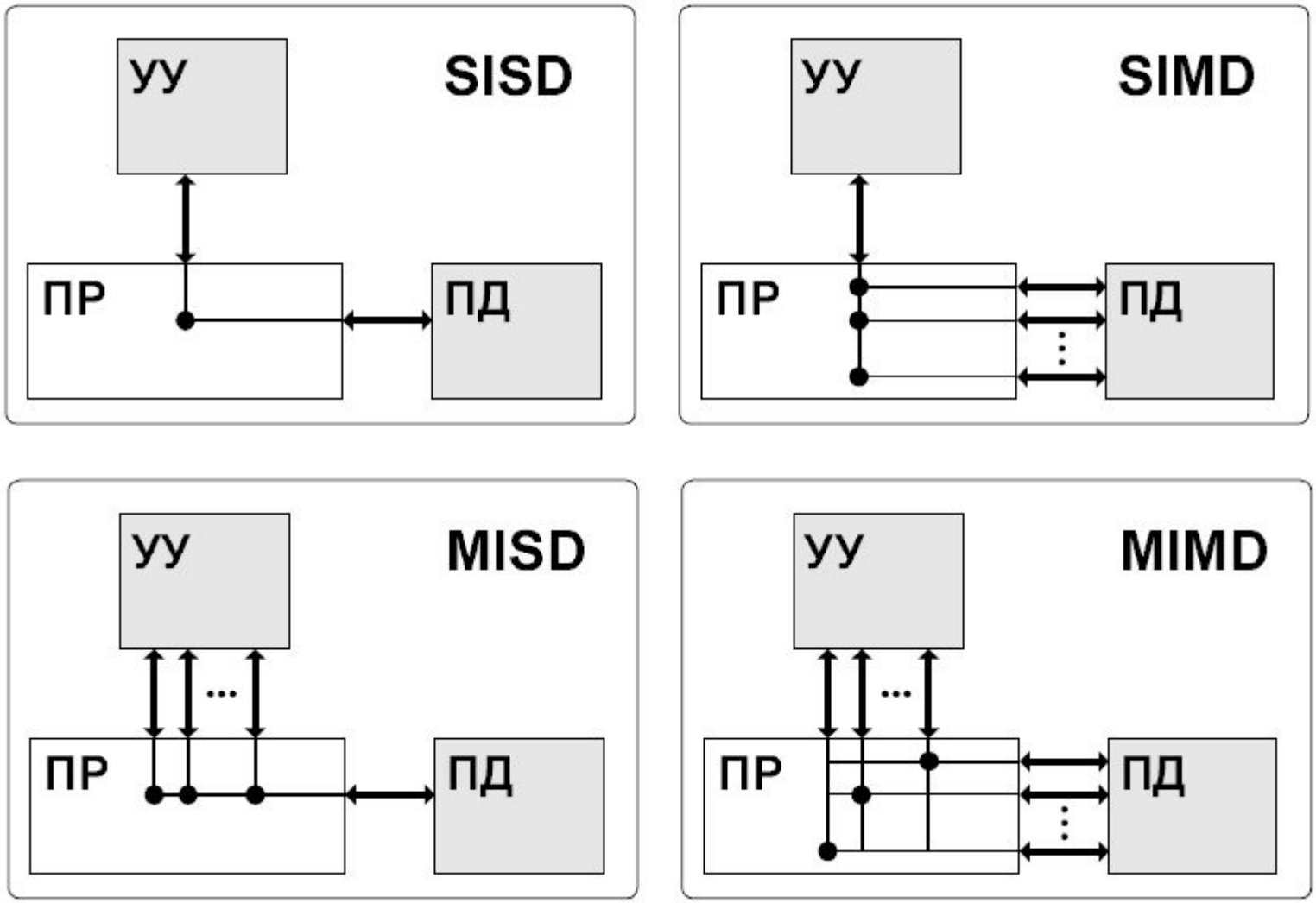
Закон Амдала

a/p	10	100	1000
0	10	100	1000
10%	5.263	9.174	9.910
25%	3.077	3.883	3.988
40%	2.174	2.463	2.496

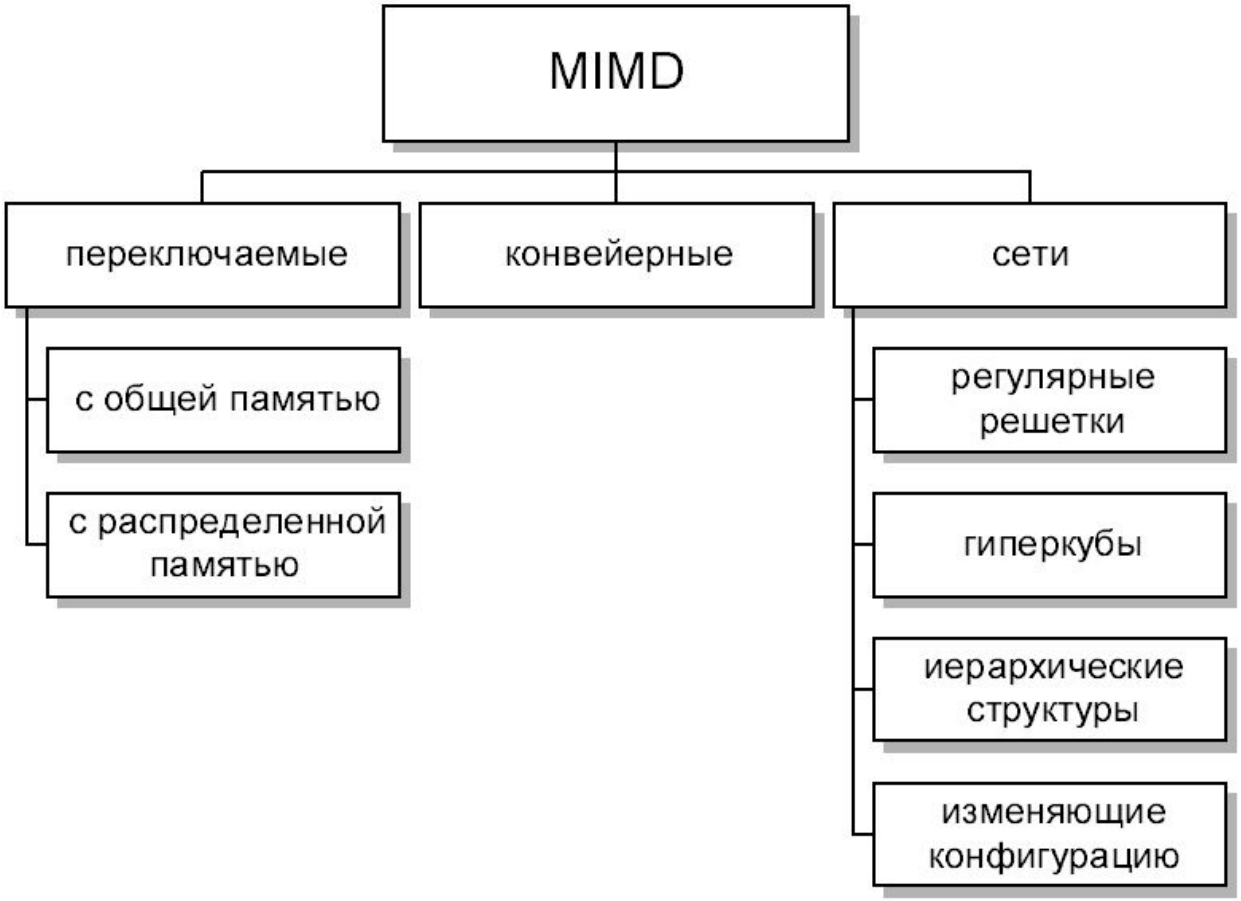
Из таблицы видно, что только алгоритм, вовсе не содержащий последовательных вычислений ($a = 0$), позволяет получить линейный прирост производительности с ростом количества вычислителей в системе. Если доля последовательных вычислений в алгоритме равна **25%**, то увеличение числа процессоров до 10 дает ускорение в **3,077** раза, а увеличение числа процессоров до 1000 даст ускорение в **3,988** раза.

Отсюда же очевидно, что при доле последовательных вычислений a общий прирост производительности не может превысить $1/a$. Так, если половина кода — последовательная, то общий прирост никогда не превысит двух.

Классификация Флина



Классификация Хокни



Дополнительная классификация Р. Хокни класса MIMD

Классификация Фенга

Каждую вычислительную систему C можно описать парой чисел (n, m). Произведение $P = n \times m$ определяет интегральную характеристику потенциала параллельности архитектуры, которую Фенг назвал *максимальной степенью параллелизма* вычислительной системы. По существу, данное значение есть не что иное, как пиковая производительность, выраженная в других едини-

Классификация Фенга

Разрядно-последовательные, пословно-последовательные ($n = m = 1$). В каждый момент времени такие компьютеры обрабатывают только один двоичный разряд. Представителем данного класса служит давняя система MINIMA с естественным описанием (1, 1).

Разрядно-параллельные, пословно-последовательные ($n > 1$, $m = 1$). Большинство классических последовательных компьютеров, так же как и многие вычислительные системы, используемые сейчас, принадлежат к данному классу: IBM 701 с описанием (36, 1), PDP-11 с описанием (16, 1), IBM 360/50 и VAX 11/780 — обе с описанием (32, 1).

Разрядно-последовательные, пословно-параллельные ($n = 1$, $m > 1$). Как правило вычислительные системы данного класса состоят из большого числа одноразрядных процессорных элементов, каждый из которых может независимо от остальных обрабатывать свои данные. Типичными примерами служат STARAN (1, 256) и MPP (1, 16384) фирмы Goodyear Aerospace, прототип известной системы ILLIAC IV компьютер SOLOMON (1, 1024) и ICL DAP (1, 4096).

Классификация Фенга

Разрядно-параллельные, пословно-параллельные ($n > 1$, $m > 1$). Подавляющее большинство параллельных вычислительных систем, обрабатывая одновременно $m \times n$ двоичных разрядов, принадлежит именно к этому классу: ILLIAC IV (64, 64), TI ASC (64, 32), C.mmp (16, 16), CDC 6600 (60, 10), BBN Butterfly GP1000 (32, 256).