

Классификация Флинна

данные

1

N

команды

1

SISD

Single **I**nstruction stream
Single **D**ata stream

SDMD

Single **I**nstruction stream
Multiple **D**ata stream

N

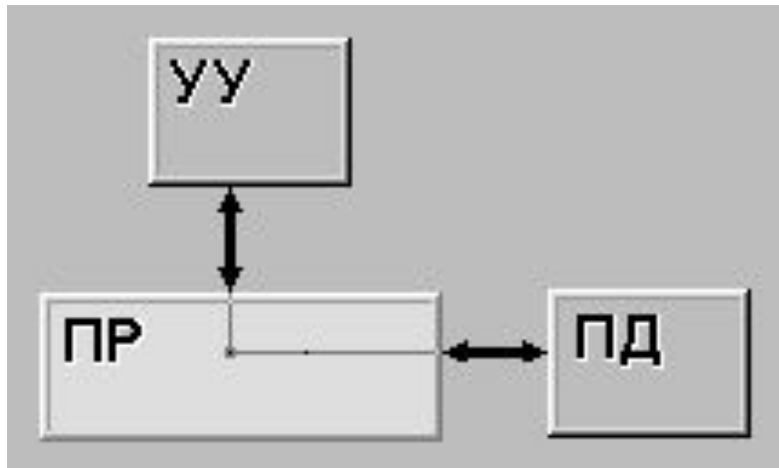
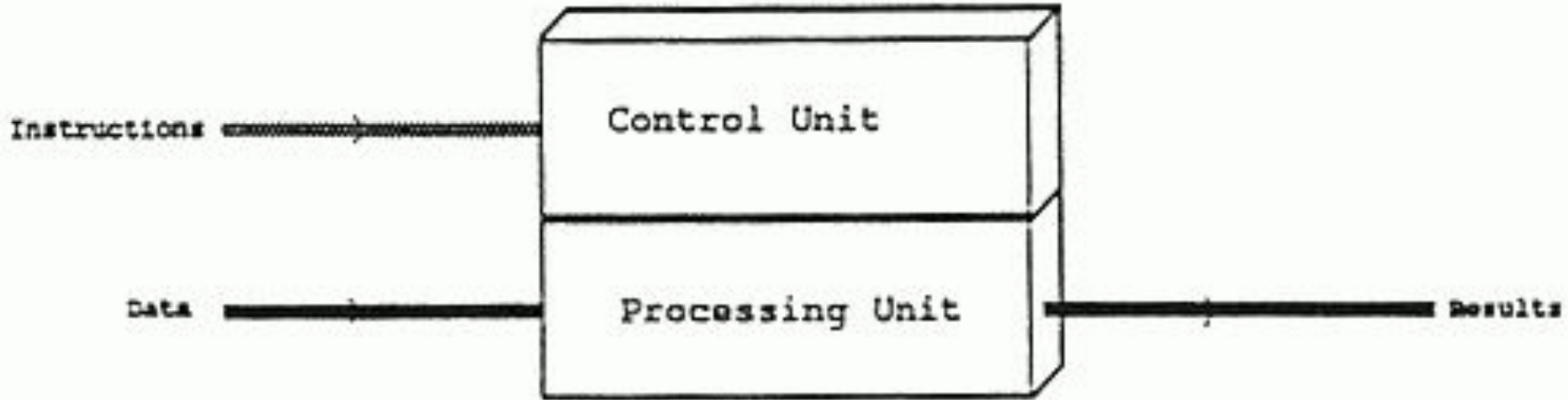
MISD

Multiple **I**nstruction stream
Single **D**ata stream

MMMD

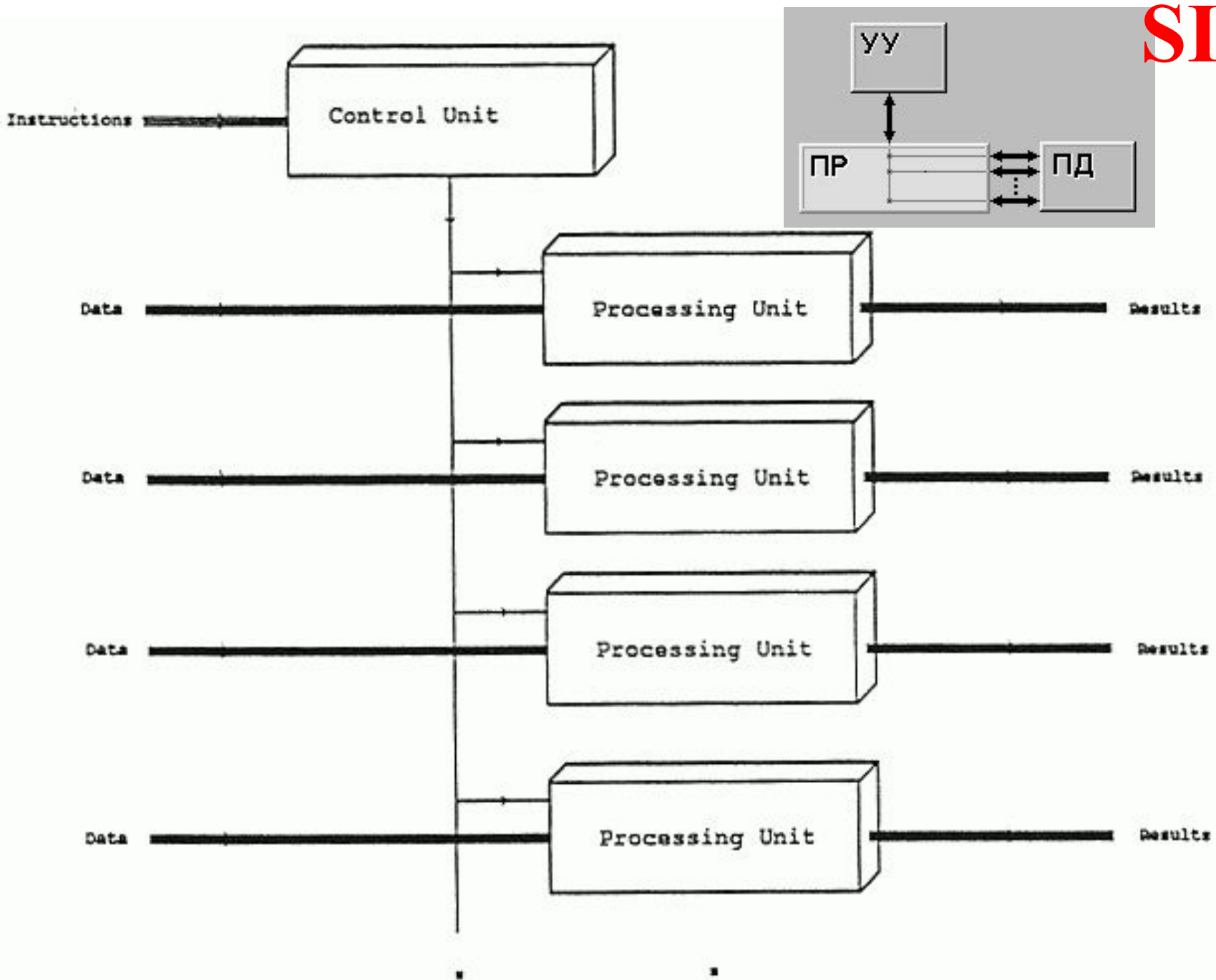
Multiple **I**nstruction stream
Multiple **D**ata stream

Классификация Флинна

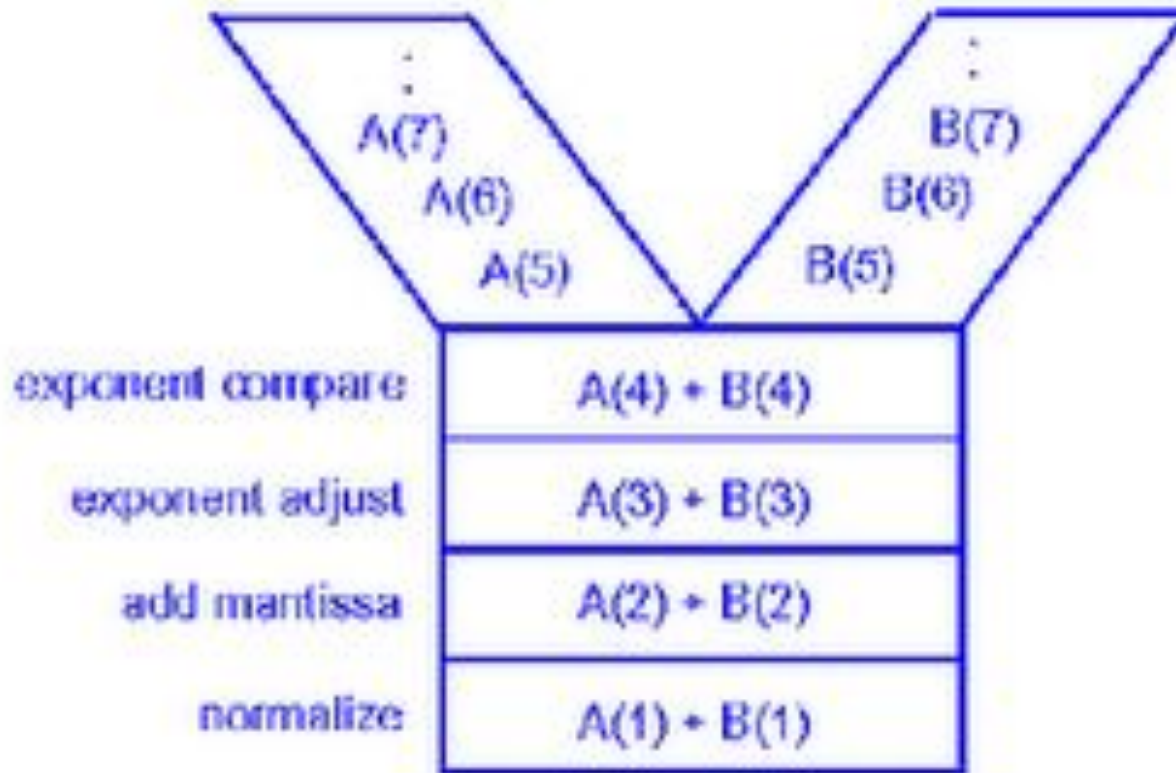


SISD

SIMD



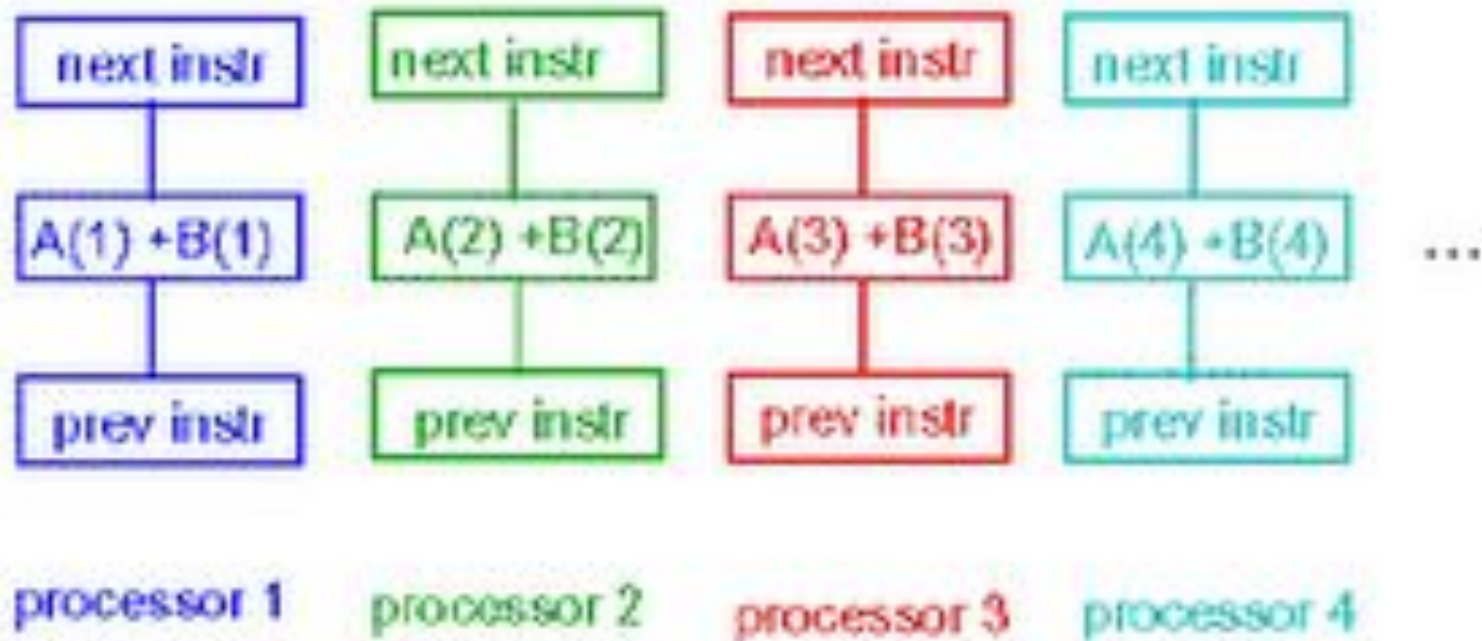
Vector SIMD Model



*Examples:

Cray 1, NEC SX-2
Fujitsu VP, Hitachi S820

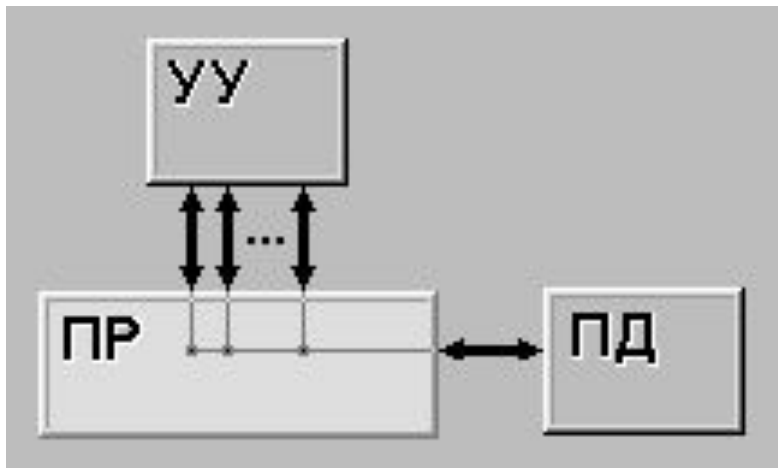
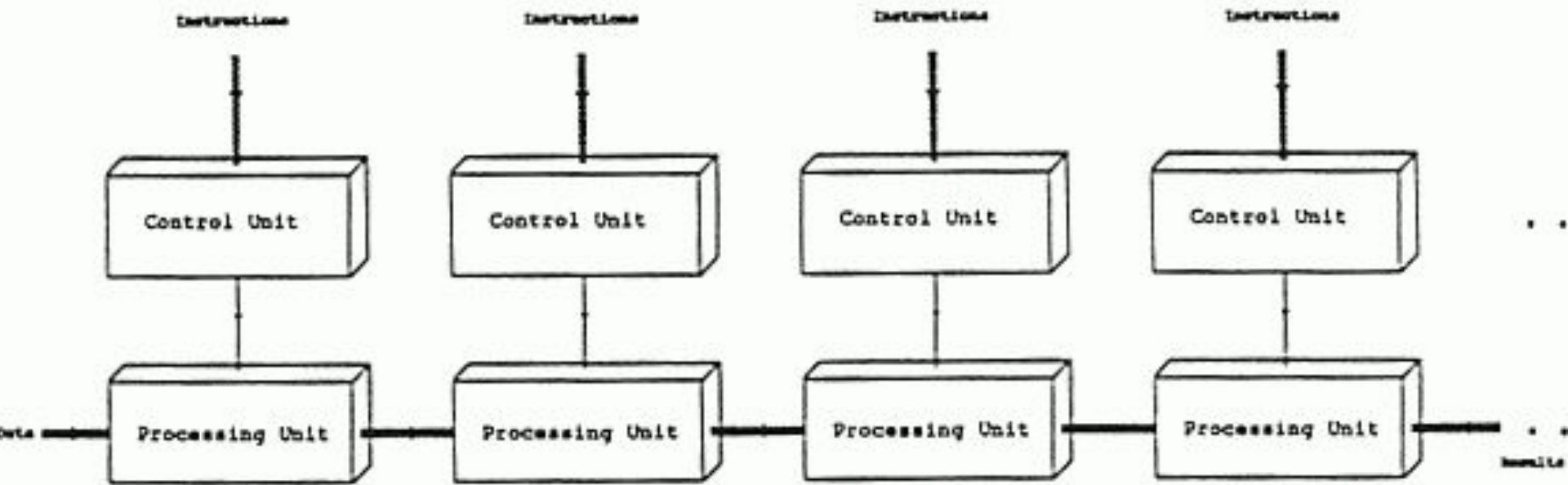
Parallel SIMD Model



* Examples:

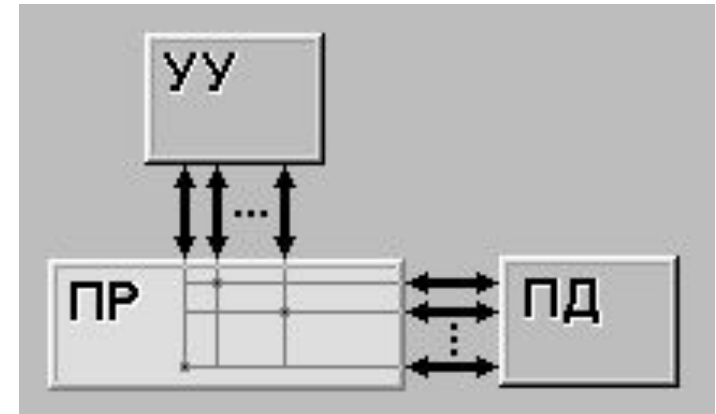
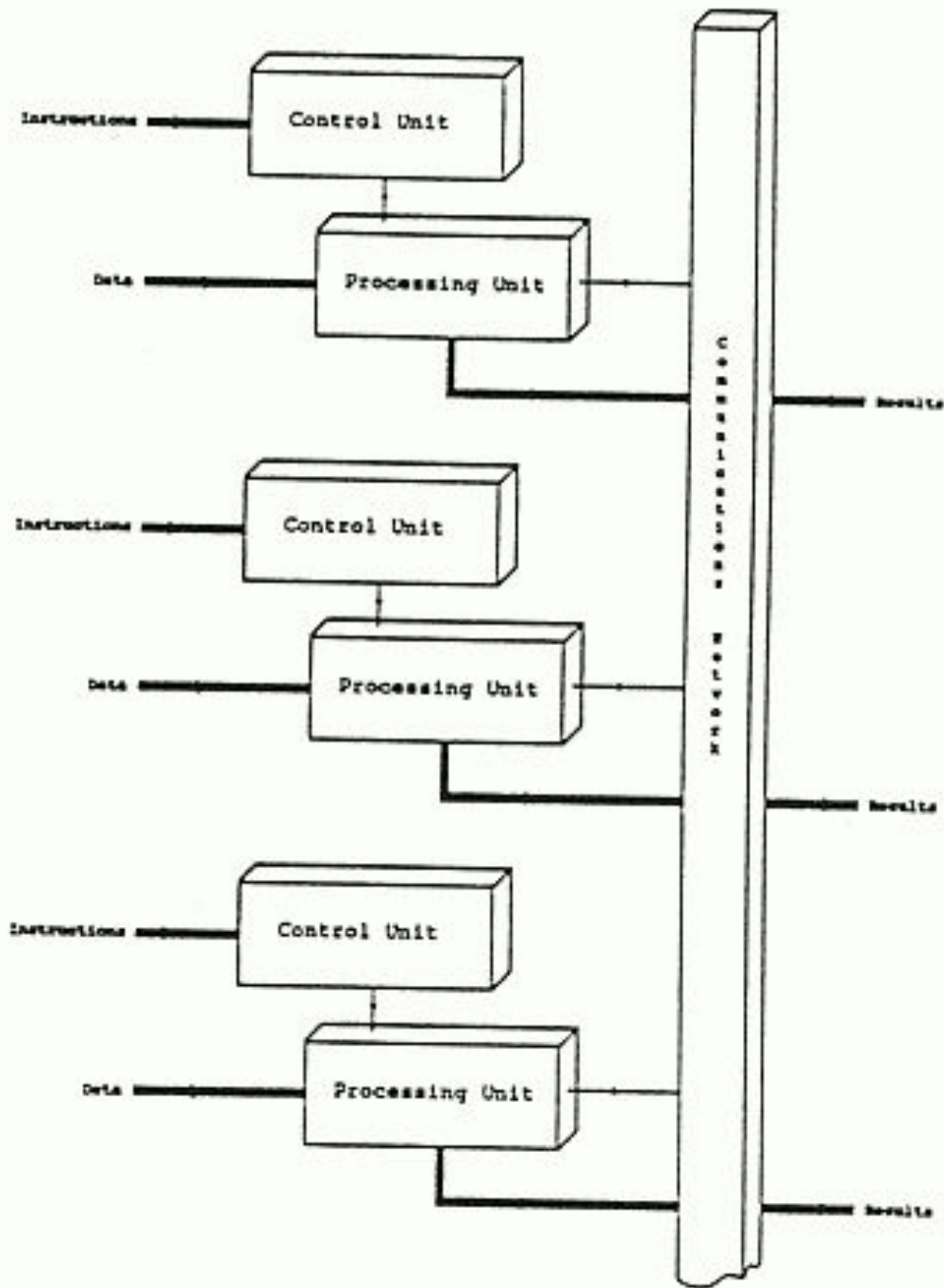
Connection Machine CM-2

Maspar MP-1, MP-2

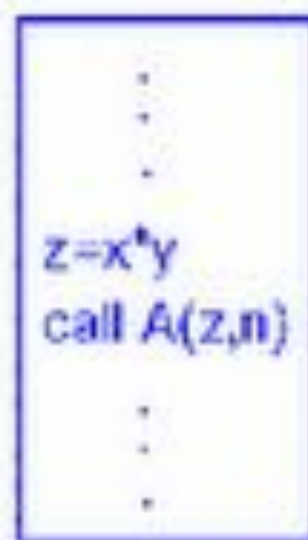


MISD

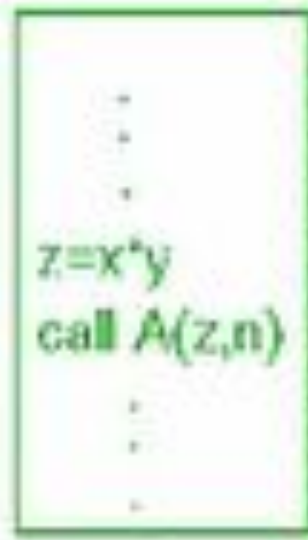
MIMD



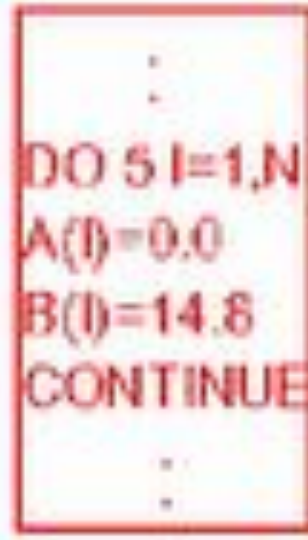
MIMD Model



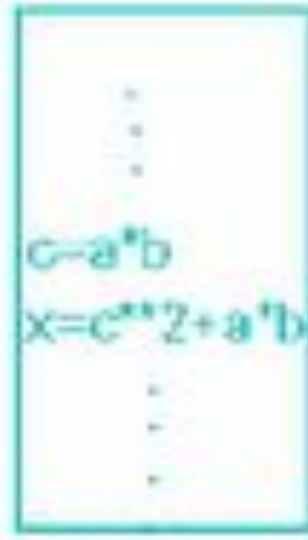
processor 1



processor 2



processor 3

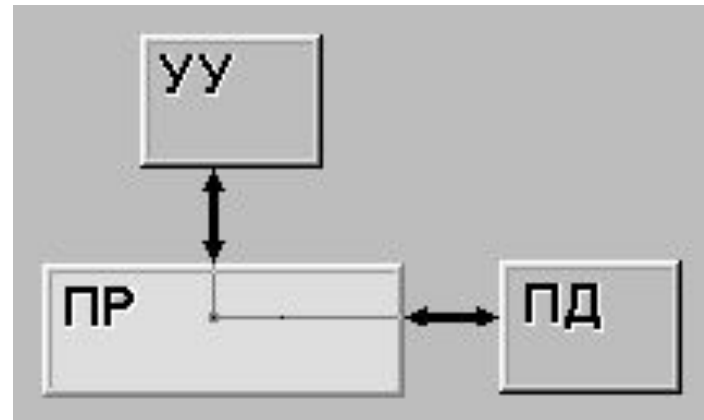


processor 4

Дополнения Ванга и Бриггса к классификации Флинна

Класс **SISD** разбивается на два подкласса:

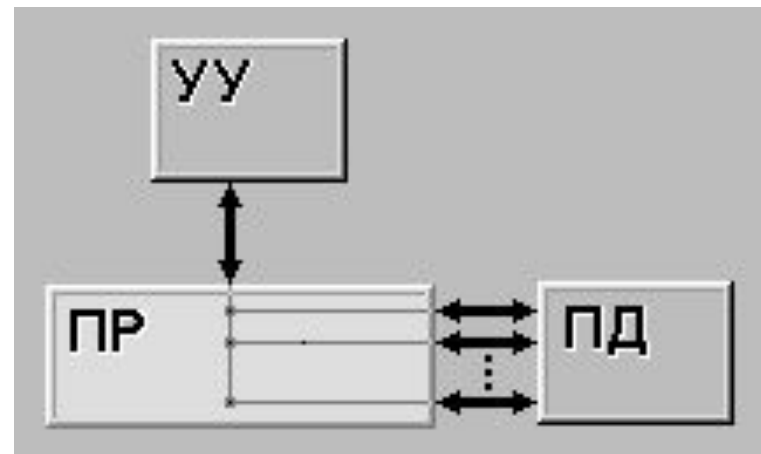
- архитектуры с единственным функциональным устройством, например, **PDP-11**;
- архитектуры, имеющие в своем составе несколько функциональных устройств - **CDC 6600, CRAY-1, FPS AP-120B, CDC Cyber 205, FACOM VP-200**.



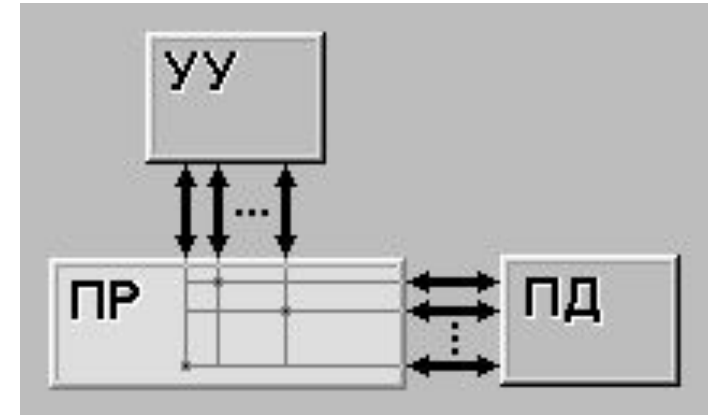
Дополнения Ванга и Бриггса к классификации Флинна

В класс **SIMD** также вводится два подкласса:

- архитектуры с пословно-последовательной обработкой информации - **ILLIAC IV, PEPE, BSP**;
- архитектуры с разрядно-последовательной обработкой - **STARAN, ICL DAP**.



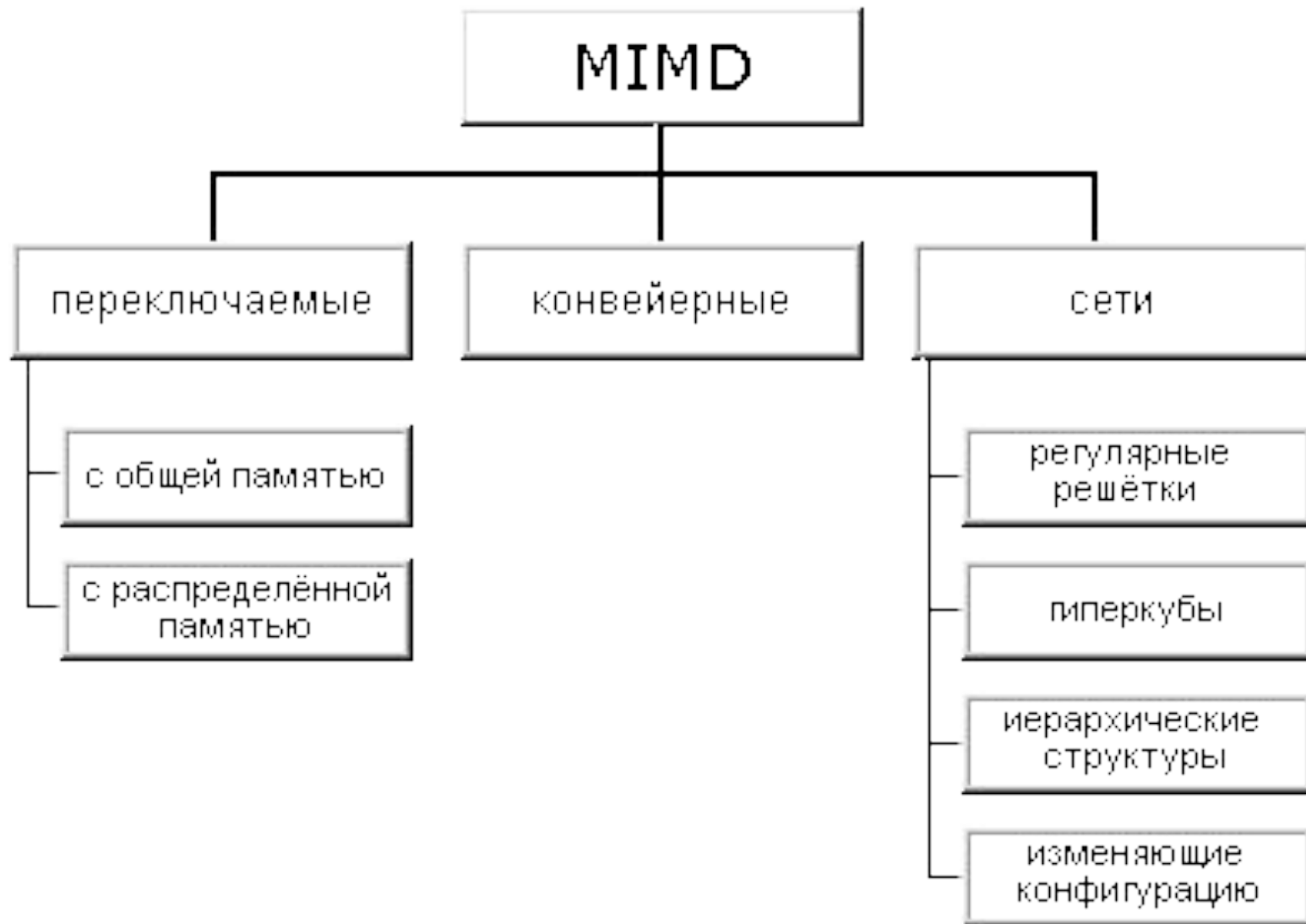
Дополнения Ванга и Бриггса к классификации Флинна



В классе **MIMD** авторы различают

- вычислительные системы со слабой связью между процессорами, к которым они относят все системы с распределенной памятью, например, **Cosmic Cube**,
- и вычислительные системы с сильной связью (системы с общей памятью), куда попадают такие компьютеры, как **S.mmp, VBN Butterfly, CRAY Y-MP, Denelcor HEP**.

Классификация Хокни



Классификация Хокни

Множественный поток команд может быть обработан двумя способами: либо одним конвейерным устройством обработки, работающем в режиме разделения времени для отдельных потоков, либо каждый поток обрабатывается своим собственным устройством. Первая возможность используется в MIMD компьютерах, которые автор называет конвейерными (например, процессорные модули в Denelcor NEP). Архитектуры, использующие вторую возможность, в свою очередь опять делятся на два класса:

- MIMD компьютеры, в которых возможна прямая связь каждого процессора с каждым, реализуемая с помощью переключателя;
- MIMD компьютеры, в которых прямая связь каждого процессора возможна только с ближайшими соседями по сети, а взаимодействие удаленных процессоров поддерживается специальной системой маршрутизации через процессоры-посредники.

Классификация Хокни

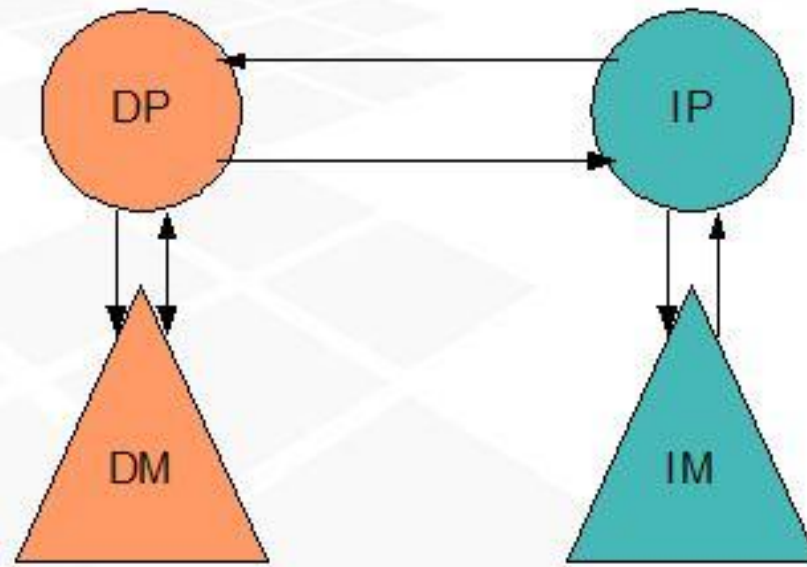
Далее, среди MIMD машин с переключателем Хокни выделяет те, в которых вся память распределена среди процессоров как их локальная память (например, PASM, PRINGLE). В этом случае общение самих процессоров реализуется с помощью очень сложного переключателя, составляющего значительную часть компьютера. Такие машины носят название MIMD машин с распределенной памятью. Если память это разделяемый ресурс, доступный всем процессорам через переключатель, то такие MIMD являются системами с общей памятью (CRAY X-MP, BBN Butterfly). В соответствии с типом переключателей можно проводить классификацию и далее: простой переключатель, многокаскадный переключатель, общая шина.

Многие современные вычислительные системы имеют как общую разделяемую память, так и распределенную локальную. Такие системы автор рассматривает как гибридные MIMD с переключателем.

Классификация Хокни

При рассмотрении MIMD машин с сетевой структурой считается, что все они имеют распределенную память, а дальнейшая классификация проводится в соответствии с топологией сети: звездообразная сеть (ICAP), регулярные решетки разной размерности (Intel Paragon, CRAY T3D), гиперкубы (NCube, Intel iPCS), сети с иерархической структурой, такой, как деревья, пирамиды, кластеры (C_m^* , CEDAR) и, наконец, сети, изменяющие свою конфигурацию.

Заметим, что если архитектура компьютера спроектирована с использованием нескольких сетей с различной топологией, то, по всей видимости, по аналогии с гибридными MIMD с переключателями, их стоит назвать гибридными сетевыми MIMD, а использующие идеи разных классов - просто гибридными MIMD. Типичным представителем последней группы, в частности, является компьютер Connection Machine 2, имеющим на внешнем уровне топологию гиперкуба, каждый узел которого является кластером процессоров с полной связью.



Классификация Скилликорна, 1989

DP - Data Processor

IP - Instruction Processor

DM - Data Memory

IM - Instruction Memory

Классификация Скилликорна, 1989

Предлагается рассматривать архитектуру любого компьютера, как абстрактную структуру, состоящую из четырех компонент:

- ***процессор команд*** (IP - Instruction Processor) - функциональное устройство, работающее, как интерпретатор команд; в системе, вообще говоря, может отсутствовать;
- ***процессор данных*** (DP - Data Processor) - функциональное устройство, работающее как преобразователь данных, в соответствии с арифметическими операциями;
- ***иерархия памяти*** (IM - Instruction Memory, DM - Data Memory) - запоминающее устройство, в котором хранятся данные и команды, пересылаемые между процессорами;
- ***переключатель*** - абстрактное устройство, обеспечивающее связь между процессорами и памятью.

Классификация Скилликорна, 1989

Функции **процессора команд** во многом схожи с функциями **устройств управления** последовательных машин и, согласно Д.Скилликорну, сводятся к следующим:

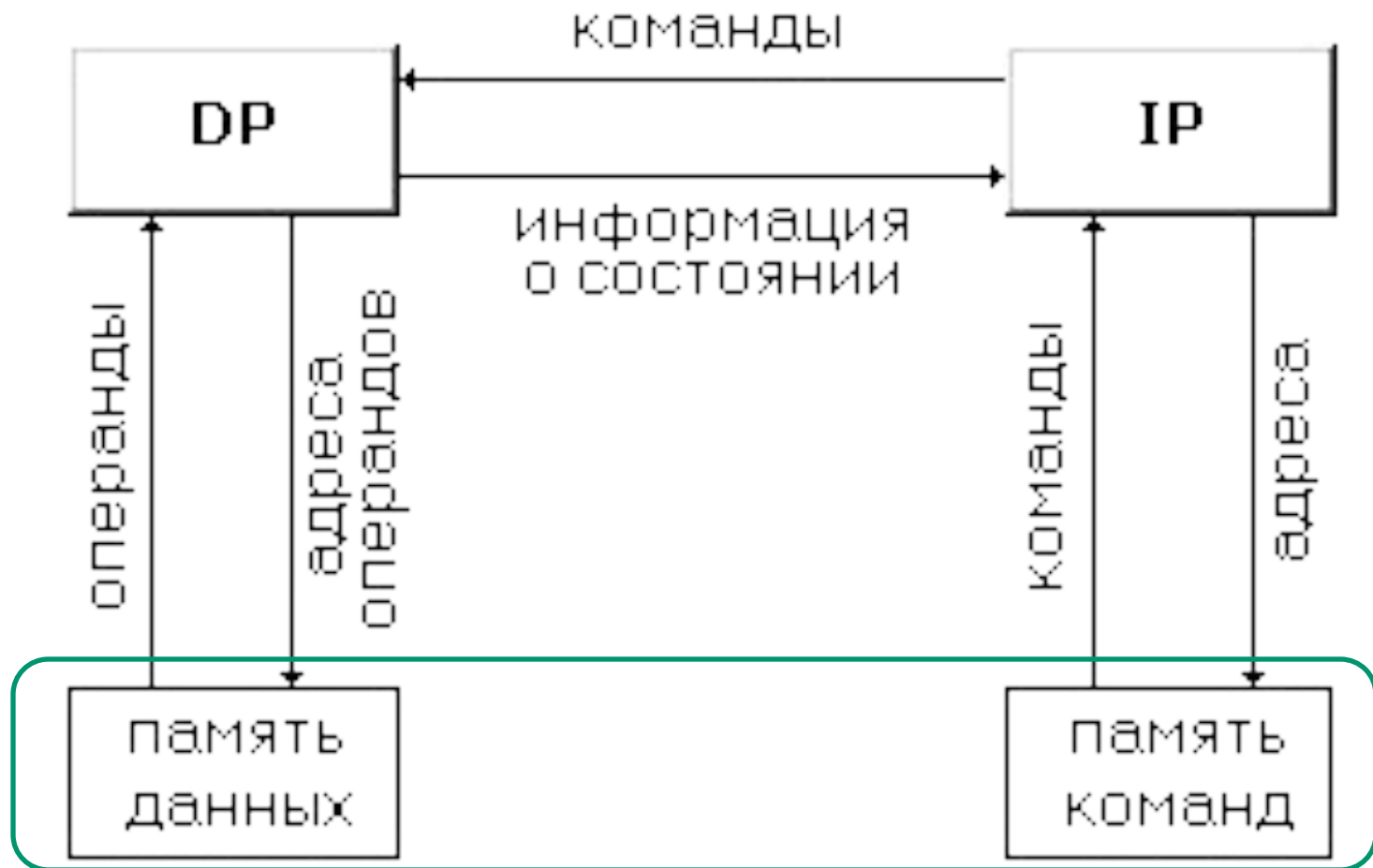
- на основе своего состояния и полученной от ДР информации IP определяет адрес команды, которая будет выполняться следующей;
- осуществляет доступ к IM для выборки команды;
- получает и декодирует выбранную команду;
- сообщает ДР команду, которую надо выполнить;
- определяет адреса операндов и посылает их в ДР;
- получает от ДР информацию о результате выполнения команды.

Классификация Скилликорна, 1989

Функции **процессора данных** делают его во многом похожим на **арифметическое устройство** традиционных процессоров:

- DP получает от IP команду, которую надо выполнить;
- получает от IP адреса операндов;
- выбирает операнды из DM;
- выполняет команду;
- запоминает результат в DM;
- возвращает в IP информацию о состоянии после выполнения команды.

В терминах таким образом определенных основных частей компьютера структуру традиционной фон-неймановской архитектуры можно представить в следующем виде:



Для описания параллельных вычислительных систем автор зафиксировал четыре типа переключателей, без какой-либо явной связи с типом устройств, которые они соединяют:

- **1-1** - переключатель такого типа связывает пару функциональных устройств;
- **n-n** - переключатель связывает i -е устройство из одного множества устройств с i -м устройством из другого множества, т.е. фиксирует попарную связь;
- **1-n** - переключатель соединяет одно выделенное устройство со всеми функциональными устройствами из некоторого набора;
- **$n \times n$** - каждое функциональное устройство одного множества может быть связано с любым устройством другого множества, и наоборот.

Примеров подобных переключателей можно привести очень много. Так, все матричные процессоры имеют переключатель типа **1-n** для связи единственного процессора команд со всеми процессорами данных. В компьютерах семейства Connection Machine каждый процессор данных имеет свою локальную память, следовательно, связь будет описываться как **n-n**. В тоже время, каждый процессор команд может связаться с любым другим процессором, поэтому данная связь будет описана как $n \times n$.



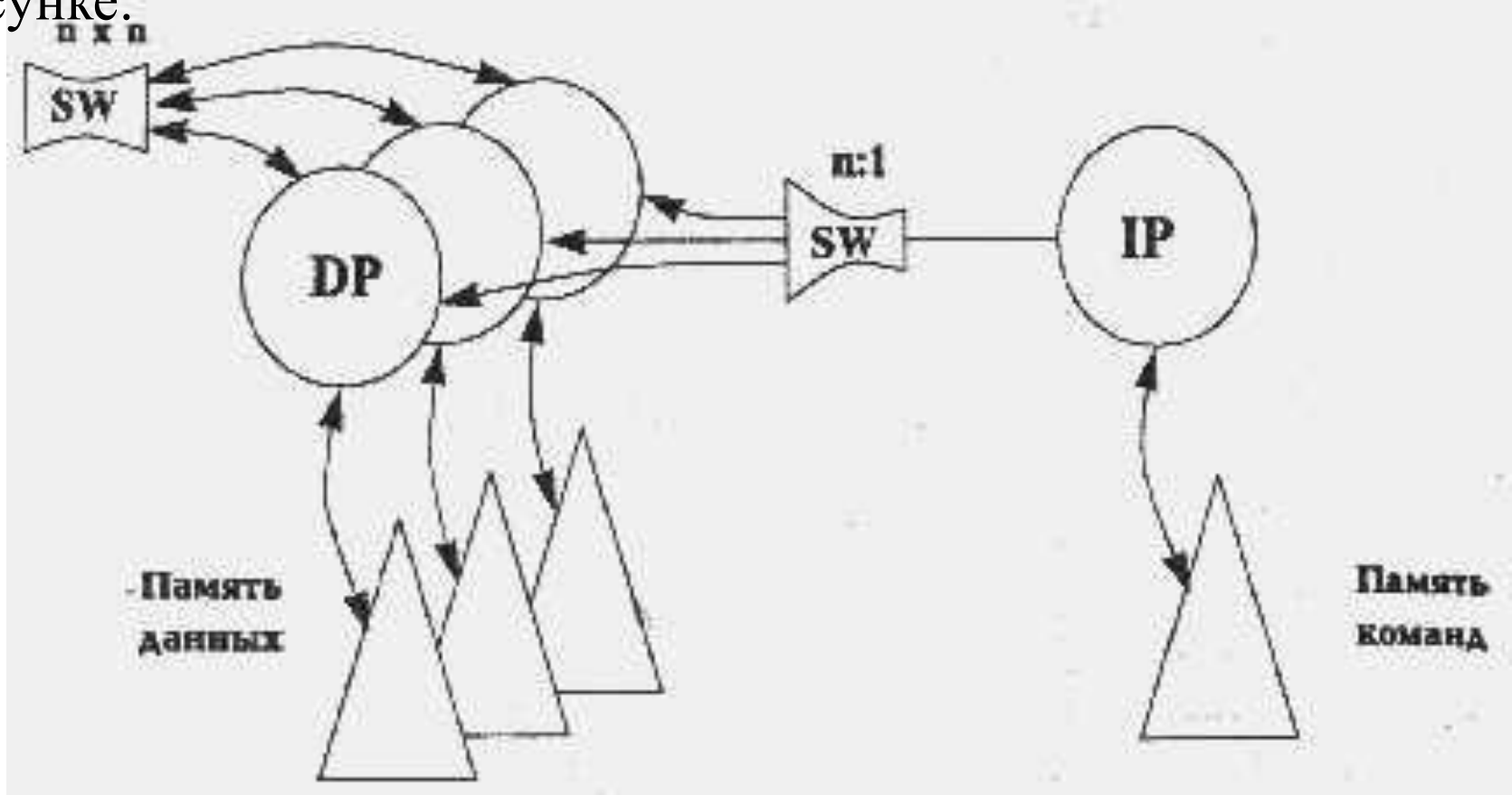
Классификация **Д.Скилликорна** состоит из двух уровней. На первом уровне она проводится на основе восьми характеристик:

1. количество процессоров команд (**IP**);
2. число запоминающих устройств (модулей памяти) команд (**IM**);
3. тип переключателя между **IP** и **IM**;
4. количество процессоров данных (**DP**);
5. число запоминающих устройств (модулей памяти) данных (**DM**);
6. тип переключателя между **DP** и **DM**;
7. тип переключателя между **IP** и **DP**;
8. тип переключателя между **DP** и **DP**.

Рассмотрим упомянутый выше компьютер Connection Machine 2. В терминах данных характеристик его можно описать:

IP IM DP DM IP-DP
 (1, 1, 1-1, n, n, n-n, 1-n, n×n),
IP-IM DP-DM DP-DP

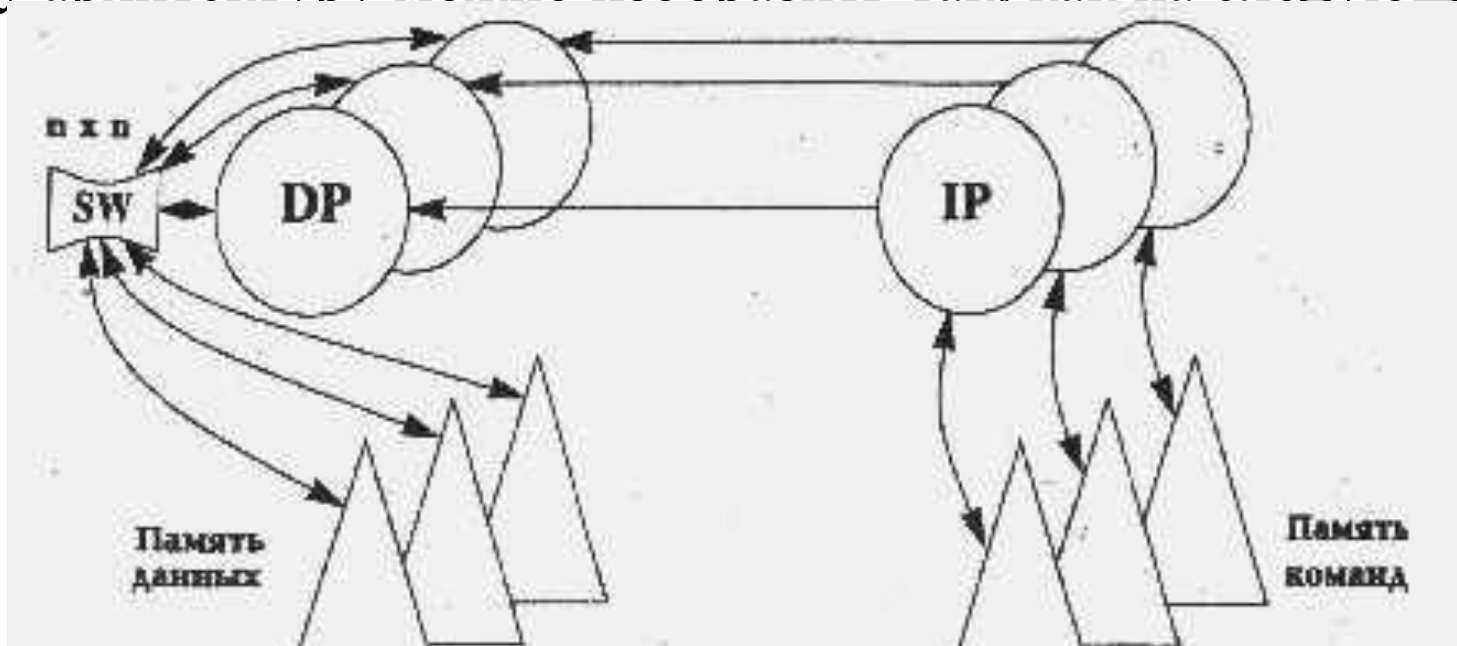
а условное изображение архитектуры приведено на следующем рисунке:



Для сильно связанных мультипроцессоров (BBN Butterfly, C.mmp) ситуация иная. Такие системы состоят из множества процессоров, соединенных с модулями памяти с помощью динамического переключателя. Задержка при доступе любого процессора к любому модулю памяти примерно одинакова. Связь и синхронизация между процессорами осуществляется через общие (разделяемые) переменные. Описание таких машин в рамках данной классификации выглядит так:

IP IM DP DM IP-DP
 (n, n, n-n, n, n, n×n, n-n, нет),
IP-IM DP-DM DP-DP

а саму архитектуру можно изобразить так, как на следующем рисунке:



Используя введенные характеристики и предполагая, что рассмотрение количественных характеристик можно ограничить только тремя возможными вариантами значений: 0, 1 и n (т.е. больше одного), можно получить **28** классов архитектур.

В классах **1-5** находятся компьютеры типа dataflow и reduction, не имеющие процессоров команд в обычном понимании этого слова. Класс **6** это классическая фон-неймановская последовательная машина. Все разновидности матричных процессоров содержатся в классах **7-10**. Классы **11** и **12** отвечают компьютерам типа MISD классификации Флинна и на настоящий момент, по мнению автора, пусты. Классы с **13**-го по **28**-й занимают всевозможные варианты мультипроцессоров, причем в **13-20** классах находятся машины с достаточно привычной архитектурой, в то время, как архитектура классов **21-28** пока выглядит экзотично.

На втором уровне классификации Д.Скилликорн просто уточняет описание, сделанное на первом уровне, добавляя возможность конвейерной обработки в процессорах команд и данных.

В конце данного описания имеет смысл привести сформулированные автором три цели, которым должна служить хорошо построенная классификация:

- облегчать понимание того, что достигнуто на сегодняшний день в области архитектур вычислительных систем, и какие архитектуры имеют лучшие перспективы в будущем;
- подсказывать новые пути организации архитектур - речь идет о тех классах, которые в настоящее время по разным причинам пусты;
- показывать, за счет каких структурных особенностей достигается увеличение производительности различных вычислительных систем; с этой точки зрения, классификация может служить моделью для анализа производительности.

Классификация Дункана

Из класса параллельных машин должны быть исключены те, в которых параллелизм заложен лишь на самом низком уровне, включая:

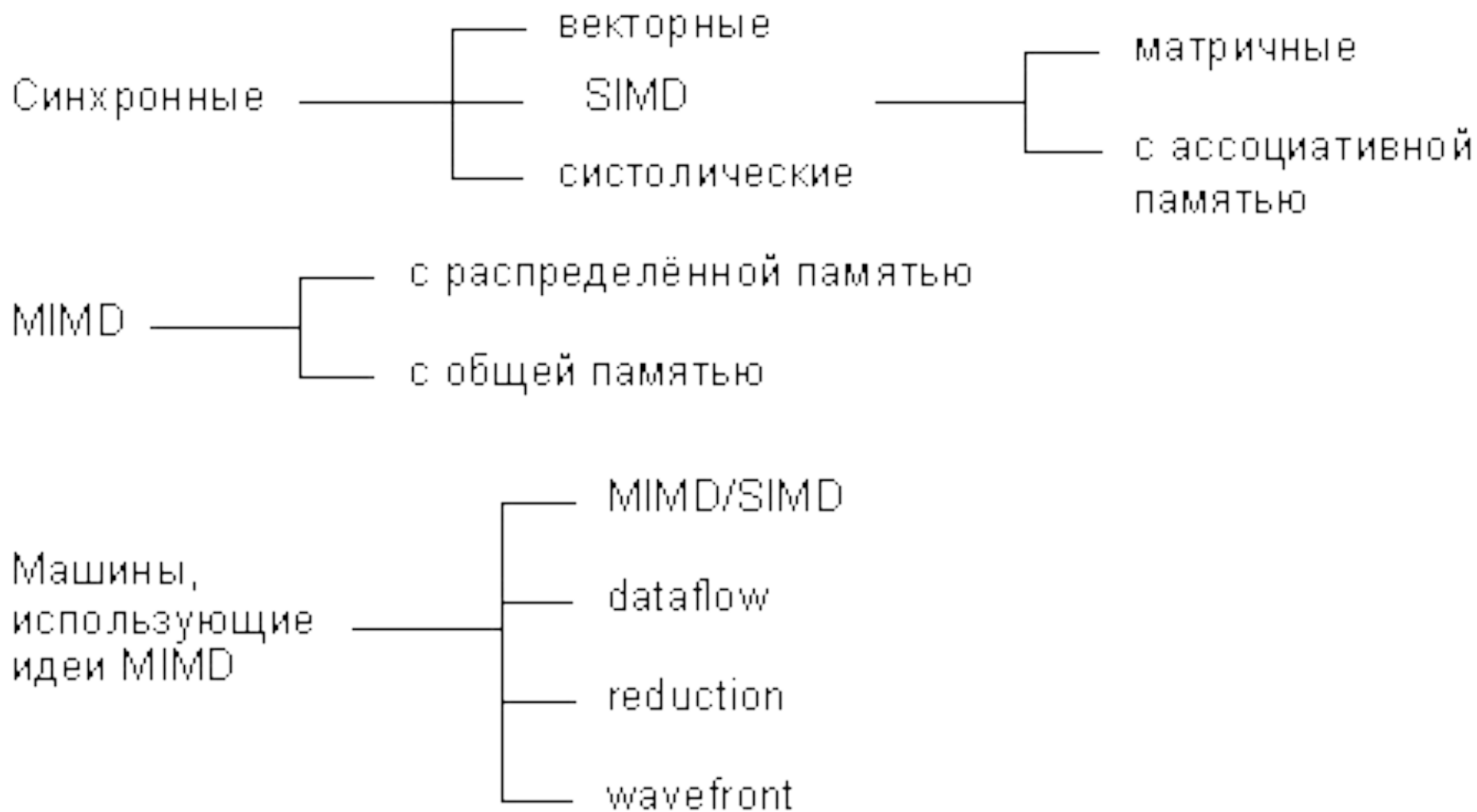
- конвейеризацию на этапе подготовки и выполнения команды (instruction pipelining), т.е. частичное перекрытие таких этапов, как дешифрация команды, вычисление адресов операндов, выборка операндов, выполнение команды и сохранение результата;
- наличие в архитектуре нескольких функциональных устройств, работающих независимо, в частности, возможность параллельного выполнения логических и арифметических операций;
- наличие отдельных процессоров ввода/вывода, работающих независимо и параллельно с основными процессорами.

Классификация Дункана

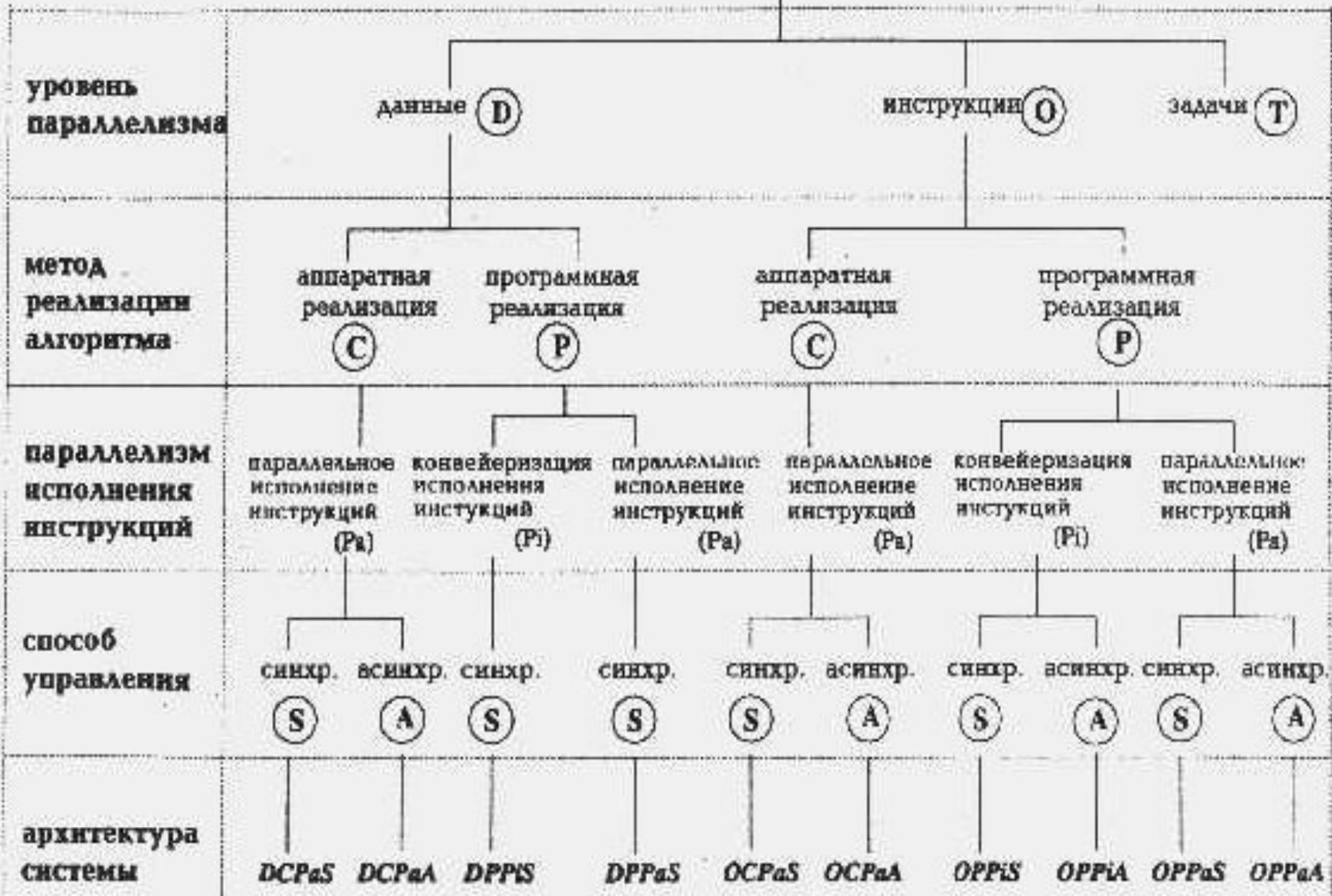
- Классификация должна быть согласованной с классификацией Флинна, показавшей правильность выбора идеи потоков команд и данных.
- Классификация должна описывать архитектуры, которые однозначно не укладываются в систематику Флинна, но, тем не менее, относятся к параллельным архитектурам (например, векторно-конвейерные).

параллельная архитектура - это такой способ организации вычислительной системы, при котором допускается, чтобы множество процессоров (простых или сложных) могло бы работать одновременно, взаимодействуя по мере надобности друг с другом.

Классификация Дункана



Классификация Базу



Классификация Базу

Конвейерные компьютеры, такие, как IBM 360/91, Amdahl 470/6 и многие современные *RISC* процессоры, разбивающие исполнение всех инструкций на несколько этапов, в данной классификации имеют обозначение OPP_iS . Более естественное применение конвейеризации происходит в векторных машинах, в которых одна команда применяется к вектору независимых данных, и за счет непрерывного использования арифметического конвейера достигается значительное ускорение. К таким компьютерам подходит обозначение DPP_iS . Матричные процессоры, в которых целое множество арифметических устройств работает одновременно в строго синхронном режиме, принадлежат к группе DPP_aS . Если вычислительная система подобно CDC 6600 имеет процессор с отдельными функциональными устройствами, управляемыми централизованно, то ее описание выглядит так: OPP_aS . Data-flow компьютеры, в зависимости от особенностей реализации, могут быть описаны либо как OPP_iA , либо OPP_aA .

Классификация Базу

Системы с несколькими процессорами, использующими параллелизм на уровне задач, не всегда можно корректно описать в рамках предложенного формализма. Если процессоры дополнительно не используют параллелизм на уровне операций или данных, то для описания можно использовать лишь букву T . В противном случае, Базу предлагает использовать знак '*' между символами, обозначающими уровни параллелизма, одновременно присутствующие в системе. Например, комбинация $T*D$ означает, что некоторая система может одновременно исполнять несколько задач, причем каждая из них может использовать векторные команды.

Классификация Базу

Очень часто в реальных системах присутствуют особенности, характерные для компьютеров из разных групп данной классификации. В этом случае для корректного описания автор использует знак '+'. Например, практически все векторные компьютеры имеют скалярную и векторную части, что можно описать как $OPP_iS + DPP_iS$ (пример - это TI ASC и CDC STAR-100). Если в системе есть возможность одновременного выполнения более одной векторной команды (как в CRAY-1) то для описания векторной части можно использовать запись $O * DPP_iS$, а полное описание данного компьютера выглядит так: $O * DPP_iS + OPP_iS$. Действуя по такому же принципу, можно найти описание и для систем CRAY X-MP и CRAY Y-MP. В самом деле, данные системы объединяют несколько процессоров, имеющих схожую с CRAY-1 структуру, и потому их описание имеет вид: $T * (O * DPP_iS + OPP_iS)$.

Классификация Хендлера

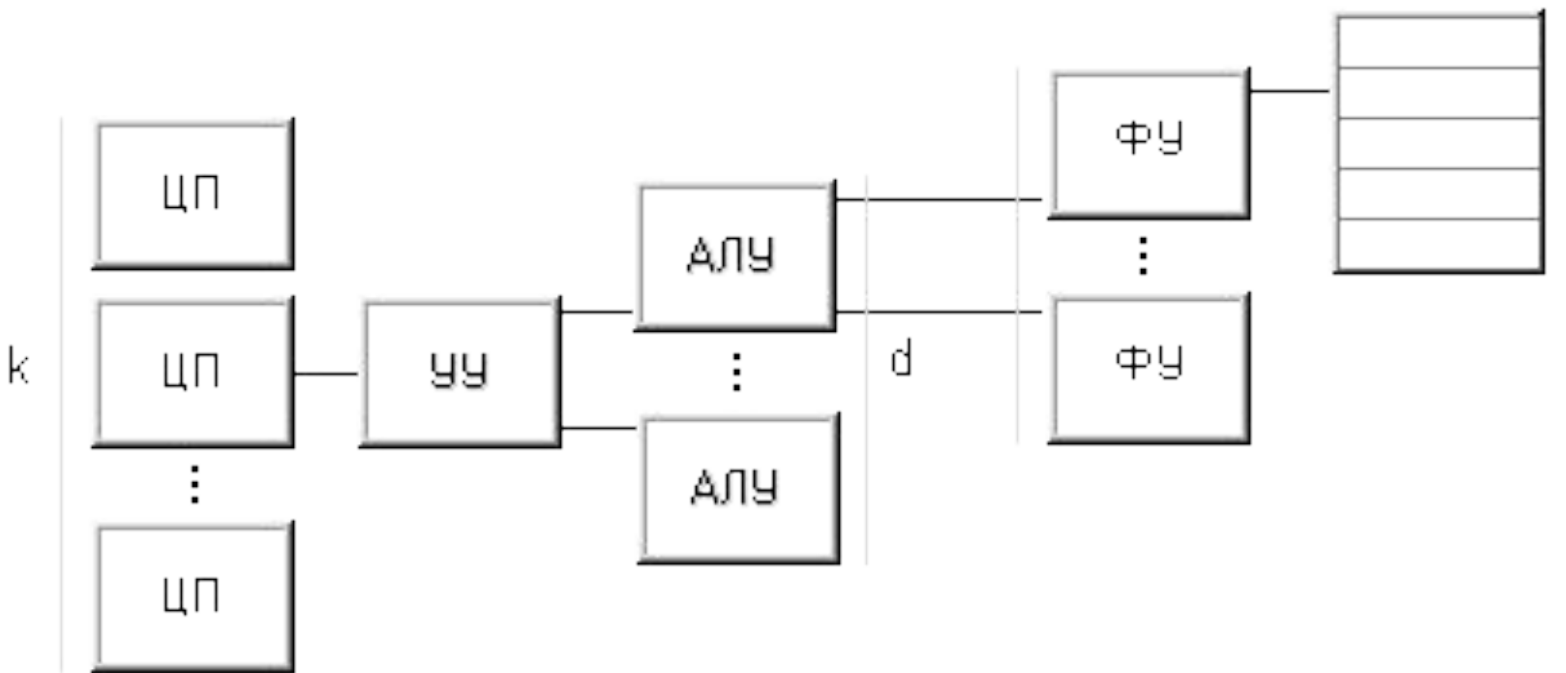
Предложенная классификация базируется на различии между тремя уровнями обработки данных в процессе выполнения программ:

- **уровень выполнения программы** - опираясь на счетчик команд и некоторые другие регистры, устройство управления (УУ) производит выборку и дешифрацию команд программы;
- **уровень выполнения команд** - арифметико-логическое устройство компьютера (АЛУ) исполняет команду, выданную ему устройством управления;
- **уровень битовой обработки** - все элементарные логические схемы процессора (ЭЛС) разбиваются на группы, необходимые для выполнения операций над одним двоичным разрядом.

Классификация Хендлера

$$t(C) = (k, d, w)$$

на каждой ступени
слово из w бит



Классификация Хендлера

$t(\text{MINIMA}) = (1,1,1);$

$t(\text{IBM 701}) = (1,1,36);$

$t(\text{SOLOMON}) = (1,1024,1);$

$t(\text{ILLIAC IV}) = (1,64,64);$

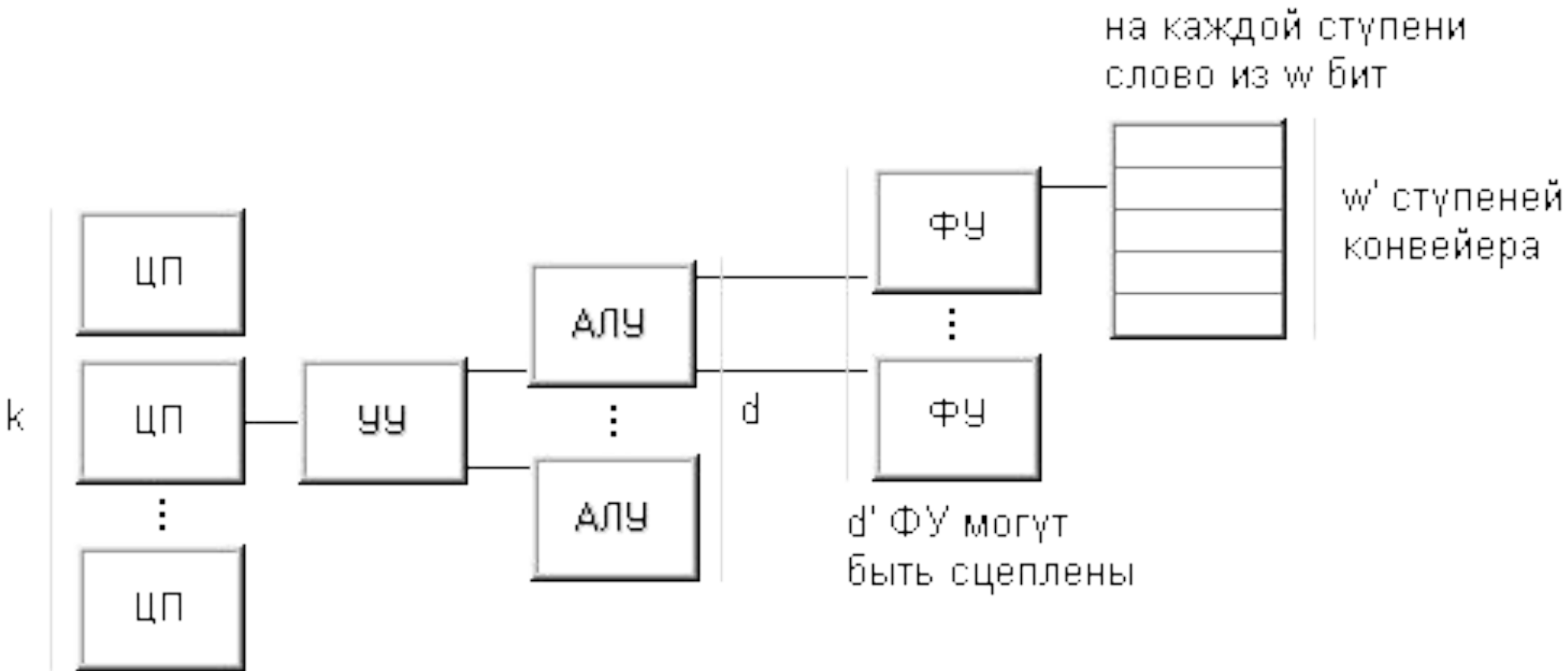
$t(\text{STARAN}) = (1,8192,1)$ - в полной конфигурации;

$t(\text{C.mmp}) = (16,1,16)$ - основной режим работы;

$t(\text{PRIME}) = (5,1,16);$

$t(\text{BBN Butterfly GP1000}) = (256, \sim 1, \sim 32).$

Классификация Хендлера



k' ЦП из k могут работать в макроконвейере

$$t = (k \times k', d \times d', w \times w')$$

k - число процессоров (каждый со своим УУ), работающих параллельно

k' - глубина макроконвейера из отдельных процессоров

d - число АЛУ в каждом процессоре, работающих параллельно

d' - число функциональных устройств АЛУ в цепочке

w - число разрядов в слове, обрабатываемых в АЛУ параллельно

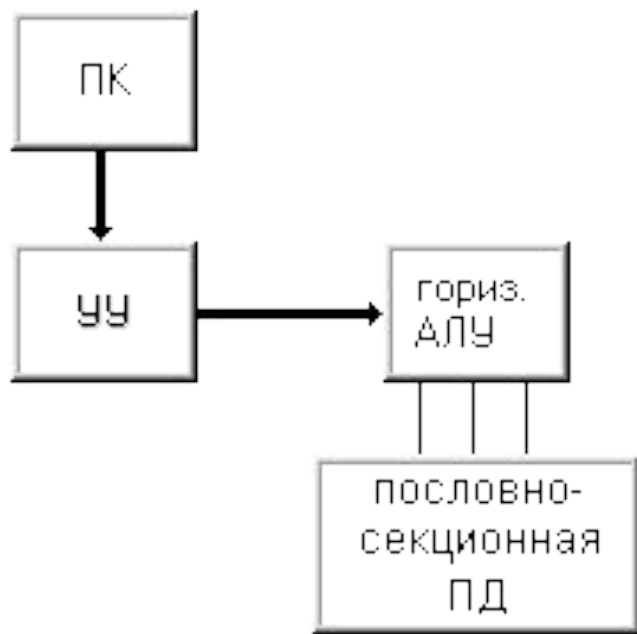
w' - число ступеней в конвейере функциональных устройств АЛУ

$$t(\text{TI ASC}) = (1,4,64 \times 8)$$

$$t(\text{PEPE}) = (1 \times 3,288,32)$$

Классификация Шора

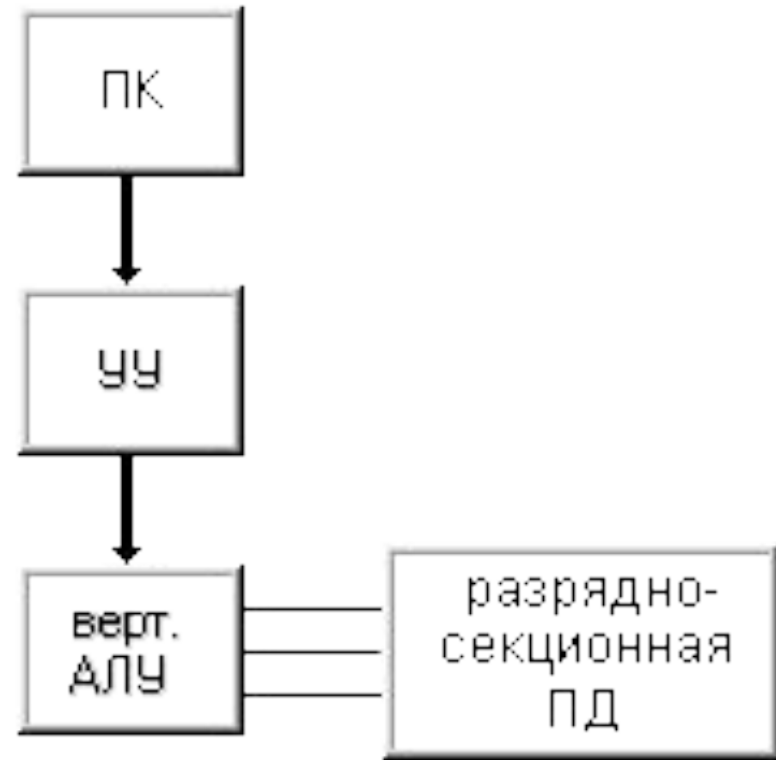
Машина I - это вычислительная система, которая содержит устройство управления, арифметико-логическое устройство, память команд и память данных с пословной выборкой. Считывание данных осуществляется выборкой всех разрядов некоторого слова для их параллельной обработки в арифметико-логическом устройстве. Состав АЛУ специально не оговаривается, что допускает наличие нескольких функциональных устройств, быть может конвейерного типа. По этим соображениям в данный класс попадают как классические последовательные машины (IBM 701, PDP-11, VAX 11/780), так и конвейерные скалярные (CDC 7600) и векторно-конвейерные (CRAY-1).



Классификация Шора

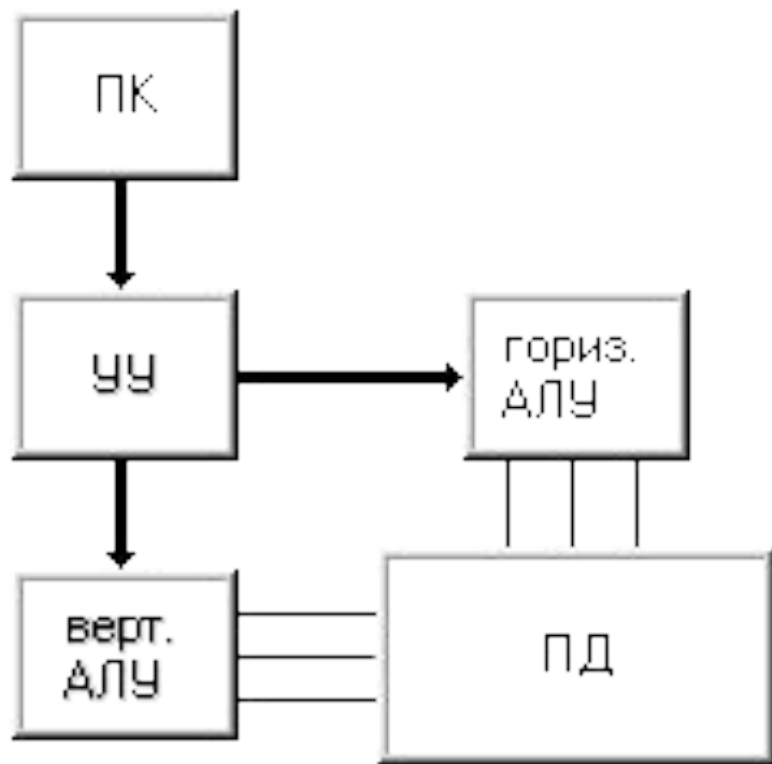
Если в машине I осуществлять выборку не по словам, а выборкой содержимого одного разряда из всех слов, то получим **машину II**. Слова в памяти данных по прежнему располагаются горизонтально, но доступ к ним осуществляется иначе. Если в машине I происходит последовательная обработка слов при параллельной обработке разрядов, то в машине II - последовательная обработка битовых слоев при параллельной обработке множества слов.

Другим примером служит матричная система ICL DAP, которая может одновременно обрабатывать по одному разряду из 4096 слов.



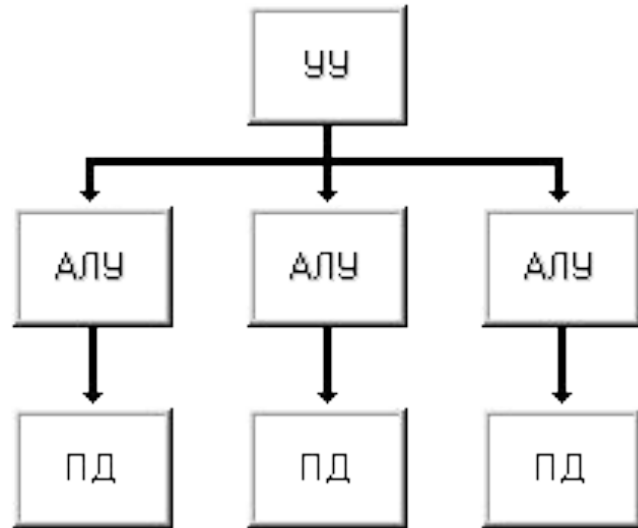
Классификация Шора

Если объединить принципы построения машин I и II, то получим **машину III**. Эта машина имеет два арифметико-логических устройства - горизонтальное и вертикальное, и модифицированную память данных, которая обеспечивает доступ как к словам, так и к битовым слоям. Впервые идею построения таких систем в 1960 году выдвинул У.Шуман, называвший их ортогональными (если память представлять как матрицу слов, то доступ к данным осуществляется в направлении, "ортогональном" традиционному - не по словам (строкам), а по битовым слоям (столбцам)).



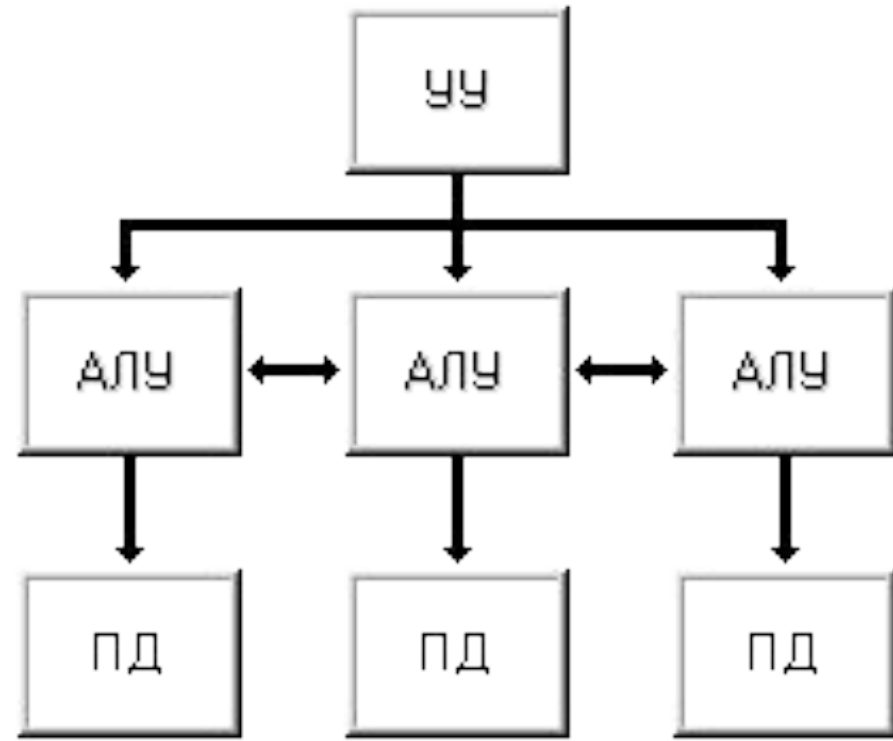
Классификация Шора

Если в машине I увеличить число пар арифметико-логическое устройство $\langle == \rangle$ память данных (иногда эту пару называют *процессорным элементом*) то получим **машину IV**. Единственное устройство управления выдает команду за командой сразу всем процессорным элементам. С одной стороны, отсутствие соединений между процессорными элементами делает дальнейшее наращивание их числа относительно простым, но с другой, сильно ограничивает применимость машин этого класса. Такую структуру имеет вычислительная система PERE, объединяющая 288 процессорных элементов.



Классификация Шора

Если ввести непосредственные линейные связи между соседними процессорными элементами машины IV, например в виде матричной конфигурации, то получим схему **машины V**. Любой процессорный элемент теперь может обращаться к данным как в своей памяти, так и в памяти непосредственных соседей. Подобная структура характерна, например, для классического матричного компьютера ILLIAC IV.



Классификация Шора



Заметим, что все машины с I-ой по V-ю придерживаются концепции разделения памяти данных и арифметико-логических устройств, предполагая наличие шины данных или какого-либо коммутирующего элемента между ними. **Машина VI**, названная *матрицей с функциональной памятью* (или памятью с встроенной логикой), представляет собой другой подход, предусматривающий распределение логики процессора по всему запоминающему устройству. Примерами могут служить как простые ассоциативные запоминающие устройства, так и сложные ассоциативные процессоры.

Классификация Шнайдера

В 1988 году Л.Шнайдер (L.Snyder) предложил новый подход [16] к описанию архитектур параллельных вычислительных систем, попадающих в класс **SIMD** систематики Флинна. Основная идея заключается в выделении этапов выборки и непосредственно исполнения в потоках команд и данных. Именно разделение потоков на адреса и их содержимое позволяет описать такие ранее "неудобные" для классификации архитектуры, как компьютеры с длинным командным словом, систолические массивы и целый ряд других.

Назовем *потоком ссылок* (reference stream) S некоторой вычислительной системы конечное множество бесконечных последовательностей пар:

$$S = \{ (a_1 \langle t_1 \rangle) (a_2 \langle t_2 \rangle) \dots, \\ (b_1 \langle u_1 \rangle) (b_2 \langle u_2 \rangle) \dots, \\ (c_1 \langle v_1 \rangle) (c_2 \langle v_2 \rangle) \dots \},$$

адрес значения поток команд I (поток данных D)

Классификация Шнайдера

Пусть S произвольный поток ссылок.

Последовательность адресов потока S , обозначаемая S_a , - это последовательность, чей i -й элемент - набор, сформированный из адресов i -х элементов каждой последовательности из S :

$$S_a = \langle a_1 \ b_1 \ \dots \ c_1 \rangle , \langle a_2 \ b_2 \ \dots \ c_2 \rangle , \dots$$

Последовательность адресов потока S , обозначаемая S_v , - это последовательность, чей i -й элемент - набор, образованный слиянием наборов значений i -х элементов каждой последовательности из S :

$$S_v = \langle t_1 \ u_1 \ \dots \ v_1 \rangle , \langle t_2 \ u_2 \ \dots \ v_2 \rangle , \dots$$

Если S_x - последовательность элементов, где каждый элемент - набор из n чисел, то для обозначения "ширины" последовательности будем пользоваться обозначением: $w(S_x) = n$.

Классификация Шнайдера

Каждую пару (I, D) с потоком команд I и потоком данных D будем называть *вычислительным шаблоном*, а все компьютеры будем разбивать на классы в зависимости от того, какой шаблон они могут исполнить. В самом деле, компьютер может исполнить шаблон

(I, D) , если он в состоянии:

- выдать $w(I_a)$ адресов команд для одновременной выборки из памяти;
- декодировать и проинтерпретировать одновременно $w(I_v)$ команд;
- выдать одновременно $w(D_a)$ адресов операндов и
- выполнить одновременно $w(D_v)$ операций над различными данными.

Если все эти условия выполнены, то компьютер может быть описан следующим образом:

$$I_{w(I_a)w(I_v)} D_{w(D_a)w(D_v)}$$

Классификация Шнайдера

Рассмотрим классическую последовательную машину. Согласно классификации Флинна, она попадает в класс SISD, следовательно $|\mathbf{I}| = |\mathbf{D}| = 1$.

Поэтому описание однопроцессорной машины с фон-неймановской архитектурой будет выглядеть так:

$\mathbf{I}_{1,1} \mathbf{D}_{1,1}$

Классификация Шнайдера

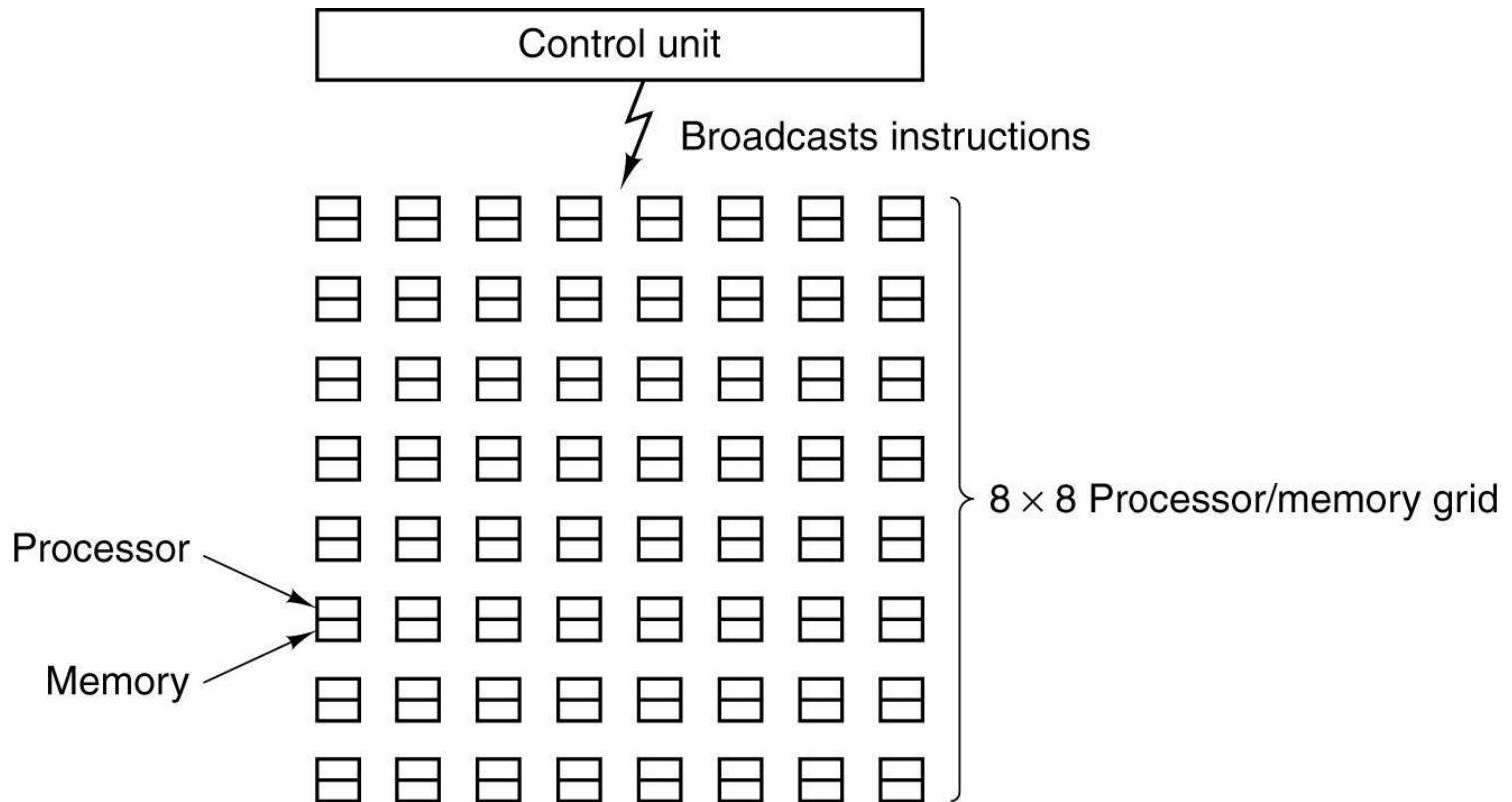
Теперь возьмем две машину из класса SIMD Goodyear Aerospace MPP. Единственный поток команд означает $|I| = 1$. По тем же соображениям, использованным только что для последовательной машины, для потока команд получаем равенство $w(I_a) = w(I_v) = 1$. Далее, вспомним, что для доступа к операндам устройство управления MPP рассылает один и тот же адрес всем процессорным элементам, поэтому в этой терминологии MPP имеет единственную последовательность в потоке данных, т.е. $|D| = 1$. Однако затем выборка данных из памяти и последующая обработка осуществляется в каждом процессорном элементе, поэтому $w(D_v) = 16384$, а вся система MPP может быть описана так:

$$I_{1,1} D_{1,16384}$$

Классификация Шнайдера

В ILLIAC IV устройство управления, так же, как и в MPP, рассылает один и тот же адрес всем процессорным элементам, однако каждый из них может получить свой уникальный адрес, добавляя содержимое локального индексного регистра. Это означает, что $|D| = 64$ и в системе присутствуют 64 потока адресов данных, определяющих одиночные потоки операндов, т.е. $w(D_a) = w(D_v) = 64$. Суммируя сказанное, приходим к описанию ILLIAC IV:

$I_{1,1}^D$ **$64,64$**



Классификация Шнайдера

Для более четкой классификации Шнайдер вводит три предиката для обозначения значений, которые могут принимать величины $w(I_a)$, $w(I_v)$, $w(D_a)$ и $w(D_v)$:

s - предикат "равен 1";

c - предикат "от 1 до некоторой (небольшой) константы";

m - предикат "от 1 до произвольно большого конечного числа".

Классификация Шнайдера

- $I_{ss} D_{ss}$ - фон-неймановские машины;
- $I_{ss} D_{sc}$ - фон-неймановские машины, в которых заложена возможность выбирать данные, расположенные с разным смещением относительно одного и того же адреса, над которыми будет выполнена одна и та же операция. Примером могут служить компьютеры, имеющие команды, типа одновременного выполнения двух операций сложения над данными в формате полуслова, расположенными по указанному адресу.
- $I_{ss} D_{sm}$ - SIMD компьютеры без возможности получения уникального адреса для данных в каждом процессорном элементе, включающие MPP, Connection Machine 1 так же, как и систолические массивы.
- $I_{ss} D_{cc}$ - многомерные SIMD машины - фон-неймановские машины, способные расщеплять поток данных на независимые потоки операндов;
- $I_{ss} D_{mm}$ - это SIMD компьютеры, имеющие возможность независимой модификации адресов операндов в каждом процессорном элементе, например, ILLIAC IV и Connection Machine 2.
- $I_{sc} D_{cc}$ - вычислительные системы, выбирающие и исполняющие одновременно несколько команд, для доступа к которым используется один адрес. Типичным примером являются компьютеры с длинным командным словом (VLIW).
- $I_{cc} D_{cc}$ - многомерные MIMD машины. Фон-неймановские машины, которые могут расщеплять свой цикл выборки/выполнения с целью обработки параллельно нескольких независимых команд.
- $I_{mm} D_{mm}$ - к этому классу относятся все компьютеры типа MIMD.