

АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

РАЗДЕЛ 1 ТЕРМИНОЛОГИЯ КЛАССИФИКАЦИОННЫЕ ПРИНЦИПЫ

ВЫЧИСЛИТЕЛЬНАЯ МАШИНА (ВМ) - комплекс программных и аппаратных средств, предназначенный для автоматической обработки информации и содержащий один или несколько процессоров, взаимодействующих с общей памятью

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА (ВС, ВК) - множество программно и структурно специализированных ВМ, совместно выполняющих информационно-вычислительный процесс (ПРС)

МНОГОМАШИННАЯ АССОЦИАЦИЯ (ММА) - объединение нескольких ВМ для выполнения совместной обработки информации по согласованному протоколу вне зависимости от территориального размещения ВМ

ВИДЫ ММА:

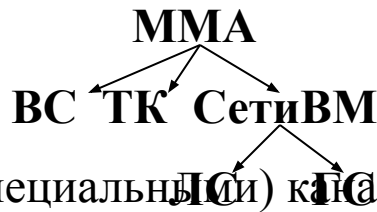
ВС - сосредоточенные ММА

ТК - рассредоточенные ММА

ЛС - рассредоточенные ММА,

объединенные выделенными (специальными) каналами (КНЛ) связи

ГС - рассредоточенные ММА, объединенные КНЛ связи общего применения



Достоинства ММА:

- объединение нескольких ВМ и ВС
- надежность решения задачи
- выравнивание нагрузки на ВМ
- объединение преимуществ различных ВМ членов ММА (информации, ПО, АС)
- перестройка, наращивание, разделение издержек и проблем эксплуатации
- огромный сервис для пользователя

АРХИТЕКТУРА – искусство строить сооружения, неразрывно сочетая решение практических и эстетических задач

АРХИТЕКТУРА ВС - комплекс оптимальных решений, принятых при проектировании ВС в :

- структурной и поведенческой организации Аппаратных средств (**АО**)
- системе программирования (**СПРГ**)
- операционной системе (**ОС**)

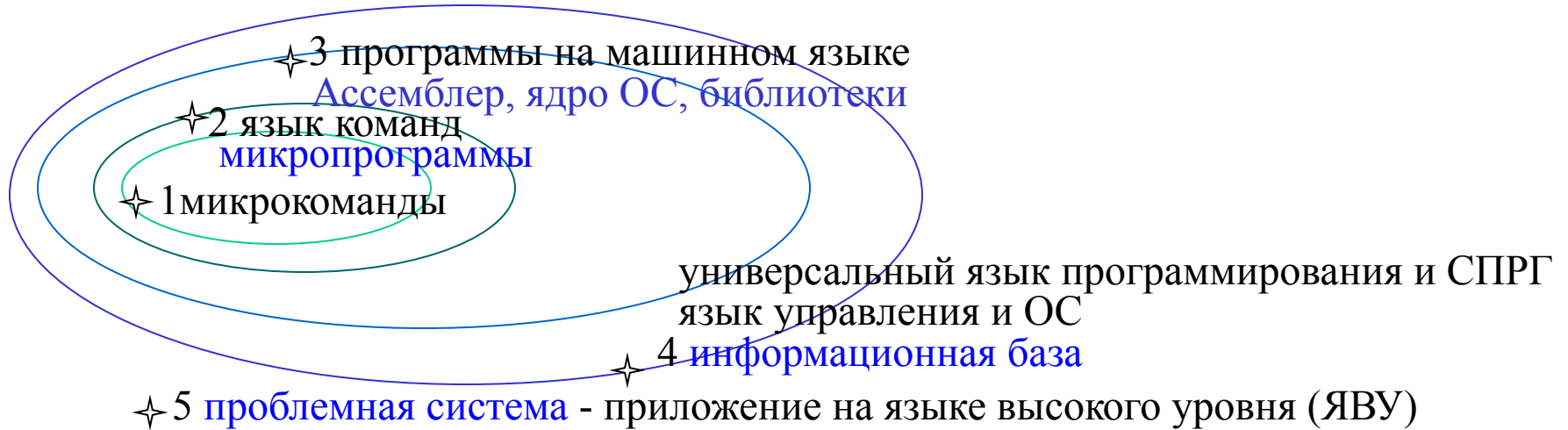
СПРГ – совокупность средств автоматизации разработки ПО: компиляторы, трансляторы, интерпретаторы, редакторы, загрузчики, отладчики, тестеры, документаторы, библиотеки

ОС – комплекс программ, обеспечивающий:

- Автоматизацию выполнения вычислительных процессов ВС в различных режимах Монопольном, Пакетной обработки, Разделения времени, Реального времени ...
- Автоматизацию распределения ресурсов ВС между вычислительными процессами :
время, память, периферия
- Автоматический контроль и защиту вычислительных процессов от взаимовлияний
- Автоматический диалог с пользователем
- Автоматический обмен с Окружением по фиксированном Интерфейсу и Протоколу

СТРУКТУРА – организация аппаратных средств

РАСПРЕДЕЛЕНИЕ ФУНКЦИЙ ВС ПО УРОВНЯМ

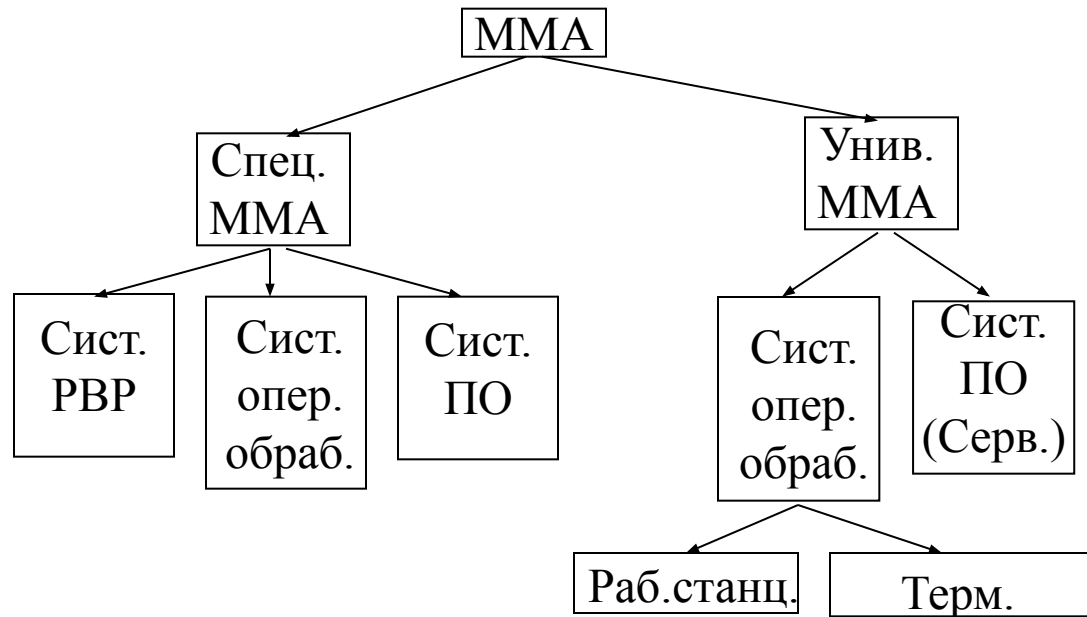


КЛАССИФИКАЦИЯ ВС ПО ПОКОЛЕНИЯМ

(принцип классификации по основным характеристикам в:)

| | I | II | III | IV |
|-------------------------------|-------------------------|---|---|---|
| 1. Элементная база | лампа 1 вент | полупроводн 1 триггер многоядерные СБИС | МИС, СИС, БИС $10^2 - 10^4$ | СБИС $10^5 - 10^7 \rightarrow$ |
| 2. Структура АО | фон-Нейман | фон-Нейман +ОШ | фон-Нейман + МОШ +конвейер + предвыб. + кэш | ММА: RISK, S/Scalar, Vector, WLIW |
| 3. Архитектура (платформа) | фон-Нейман +Ямаш+БСП | фон-Нейман +ЯВУ+монЗД | фон-Нейман+ЯВУ +многозадОС+БД | ММА+ЯВУ+БД/З +сетевОС+CASE |

КЛАССИФИКАЦИЯ ПО НАЗНАЧЕНИЮ



СИСТЕМЫ ЗАПРОС-ОТВЕТ - системы оперативного обслуживания (обработки), Для них характерно ограничение на среднее время ответа .

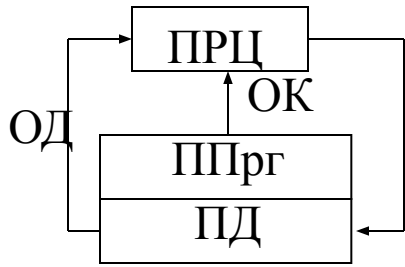
СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ (RT) - системы с абсолютным ограничением на время ответа

СИСТЕМЫ РАЗДЕЛЕНИЯ ВРЕМЕНИ (RVR) - системы коллективного пользования с преимущественным обслуживанием коротких заявок над длинными.

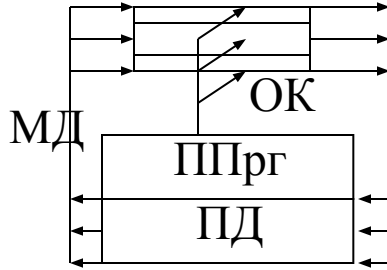
СИСТЕМЫ ПАКЕТНОЙ ОБРАБОТКИ (ПО) - системы обслуживания пакета заданий с высоким коэффициентом загрузки аппаратных средств и минимальным временем обслуживания пакета

КЛАССИФИКАЦІЯ ВС ПОТОКАМ ПК/ПД (Flynn)

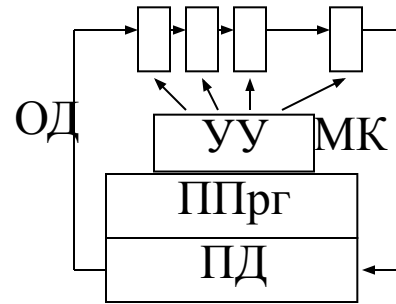
ОКОД/SISD



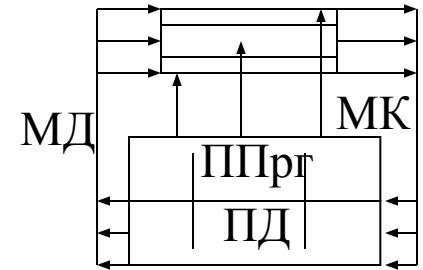
ОКМД/SIMD



МКОД/MISD



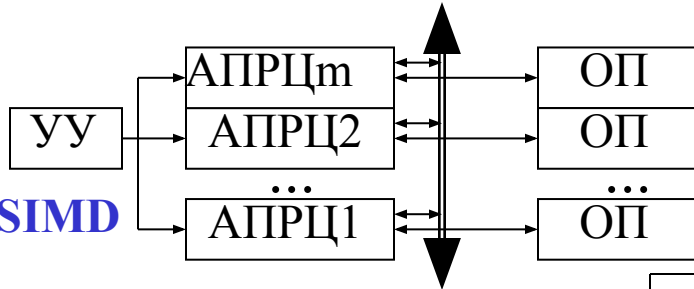
МКМД/MIMD



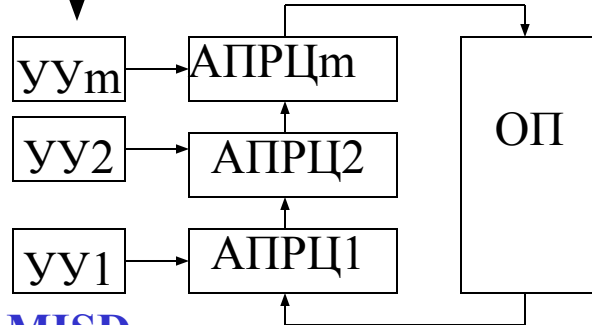
SISD



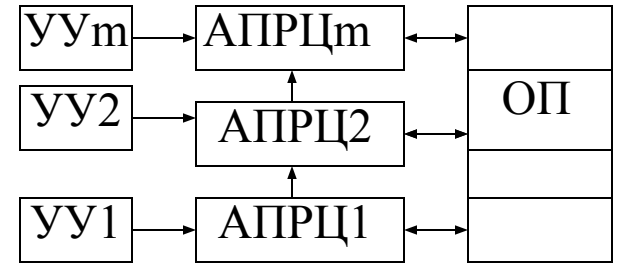
SIMD



MISD



MIMD



КЛАССЫ КЛАССИФИКАЦИИ ПО ПОТОКАМ

- **SISD** – фон-Неймановская архитектура: каждая команда арифметической обработки инициирует выполнение одной арифметической операции:
 - ▶ Единое вычислительное устройство из ПРЦ, ОП, последовательного УУ
 - ▶ Линейная структура адресации памяти (ОП – вектор слов)
 - ▶ Низкий уровень машинного языка – отсутствие типов слов
- **SIMD** – векторная архитектура:
 - ▶ Векторная команда обеспечивает одновременное выполнение операций в нескольких (m) арифметических ПРЦ
 - ▶ ОП должна быть либо в m раз скорее, либо расслоена (разделена) на m секций с независимым доступом
- **MISD** – конвейерная архитектура:
 - ▶ Обеспечивает одновременное выполнение множества операций одной формулы, связанных по промежуточным результатам
 - ▶ Одновременное выполнение множества операций не связанных промежуточными результатами и принадлежащих либо разным формулам, либо независимо вычисляемым фрагментам одной формулы
- **MIMD** – архитектура многопроцессорной матрицы

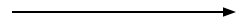
КЛАССИФИКАЦІЯ ВС ПОТОКАМ ПК/ПД (Flynn)

| SIMD | Load | Load | Mul | Sum | Sum | Sum |
|------|-----------------------|-----------------------|---------------------------|-----------------------------|-----------------------------|-----------------------------|
| ПРЦ1 | $a_1 \rightarrow r_1$ | $b_1 \rightarrow r_2$ | $r_2 * r_1 \rightarrow s$ | $s + s^{(2)} \rightarrow s$ | $s + s^{(3)} \rightarrow s$ | $s + s^{(4)} \rightarrow s$ |
| ПРЦ2 | $a_2 \rightarrow r_1$ | $b_2 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | | | |
| ПРЦ3 | $a_3 \rightarrow r_1$ | $b_3 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | | | |
| ПРЦ4 | $a_4 \rightarrow r_1$ | $b_4 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | | | |

$e (a_i * b_i)$
 $i=1..4$

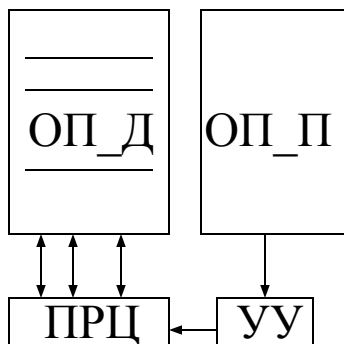
| MISD | 1 | 2 | 3 | 4 | 5 | 6 |
|------|-----------------------|-----------------------|-----------------------------|-------------------------|-----------------------------|-------------------------|
| Load | $a_1 \rightarrow r_1$ | $b_1 \rightarrow r_3$ | $a_2 \rightarrow r_1$ | $b_2 \rightarrow r_3$ | $a_3 \rightarrow r_1$ | $b_3 \rightarrow r_3$ |
| Mov | | $r_1 \rightarrow r_2$ | | $r_1 \rightarrow r_2$ | | $r_1 \rightarrow r_2$ |
| Mul | | | $r_3 * r_2 \rightarrow r_4$ | | $r_3 * r_2 \rightarrow r_4$ | |
| Sum | | | | $s + r_4 \rightarrow s$ | | $s + r_4 \rightarrow s$ |

| MIMD | Load | Load | Mul | Sum | Sum |
|------|-----------------------|-----------------------|---------------------------|-----------------------------|-----------------------------|
| ПРЦ1 | $a_1 \rightarrow r_1$ | $b_1 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | $s + s^{(2)} \rightarrow s$ | $s + s^{(3)} \rightarrow s$ |
| ПРЦ2 | $a_2 \rightarrow r_1$ | $b_2 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | | |
| ПРЦ3 | $a_3 \rightarrow r_1$ | $b_3 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | $s + s^{(4)} \rightarrow s$ | |
| ПРЦ4 | $a_4 \rightarrow r_1$ | $b_4 \rightarrow r_2$ | $r_1 * r_2 \rightarrow s$ | | |

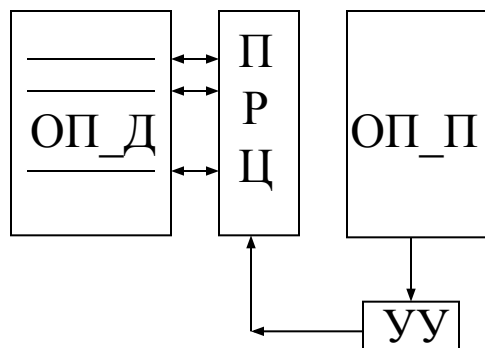
Time


КЛАССИФИКАЦИЯ ШОРА

I - SISD/W

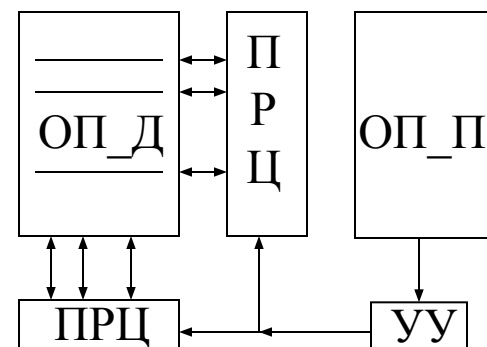


II - SIMD/b



АССОЦИАТИВНАЯ СИСТЕМА

III - SISD/W+SIMD/b

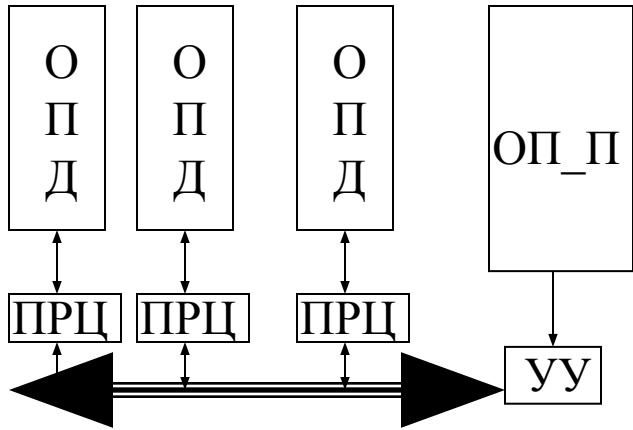


КЛАССЫ КЛАССИФИКАЦИИ ШОРА

- **I** – обычная ВМ с последовательной обработкой слов и параллельной обработкой разрядов слов (считывание данных – параллельная выборка всех разрядов слова)
- **II** – система с параллельной обработкой слов и последовательной обработкой разрядов в ассоциативном (вертикальном) ПРЦ за одну команду осуществляет параллельную обработку 1 разряда всех слов ОП (или разрядного среза). За счет этого поразрядно осуществляется параллельный поиск или обработка всех слов ОП одновременно. Адресация и выбор данных осуществляется по разрядам, выделенным маской и удовлетворяющим отношению из множества $\{ = \neq < \leq > \geq \min \max \}$
Достигается высокая скорость выполнения логических операций и их последовательностей, скорость выполнения арифметических операций ниже, чем в **I**.
- **III** - ортогональная система объединяет преимущества машин **I** и **II**. Обеспечивается эффективный поиск данных при обработке разрядных срезов в вертикальном ПРЦ и эффективная обработка найденных слов в горизонтальном ПРЦ

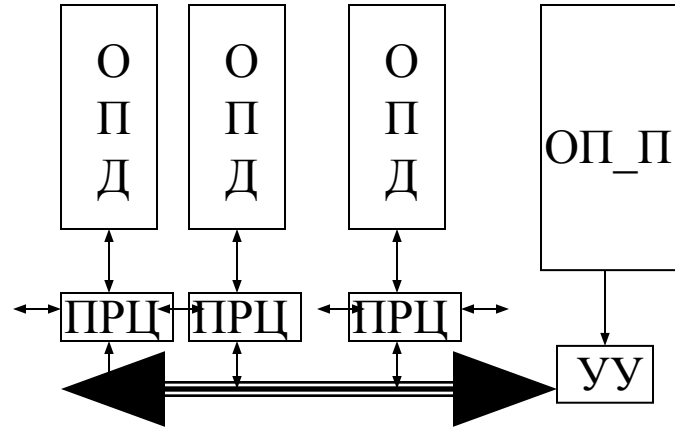
КЛАССИФИКАЦИЯ ШОРА

IV- SIMD/W/LC



АНСАМБЛЬ ПРОЦЕССОРОВ

V- SIMD/W/HC



МАТРИЧНАЯ СИСТЕМА

VI - SIMD/b + SIMD/W/HC

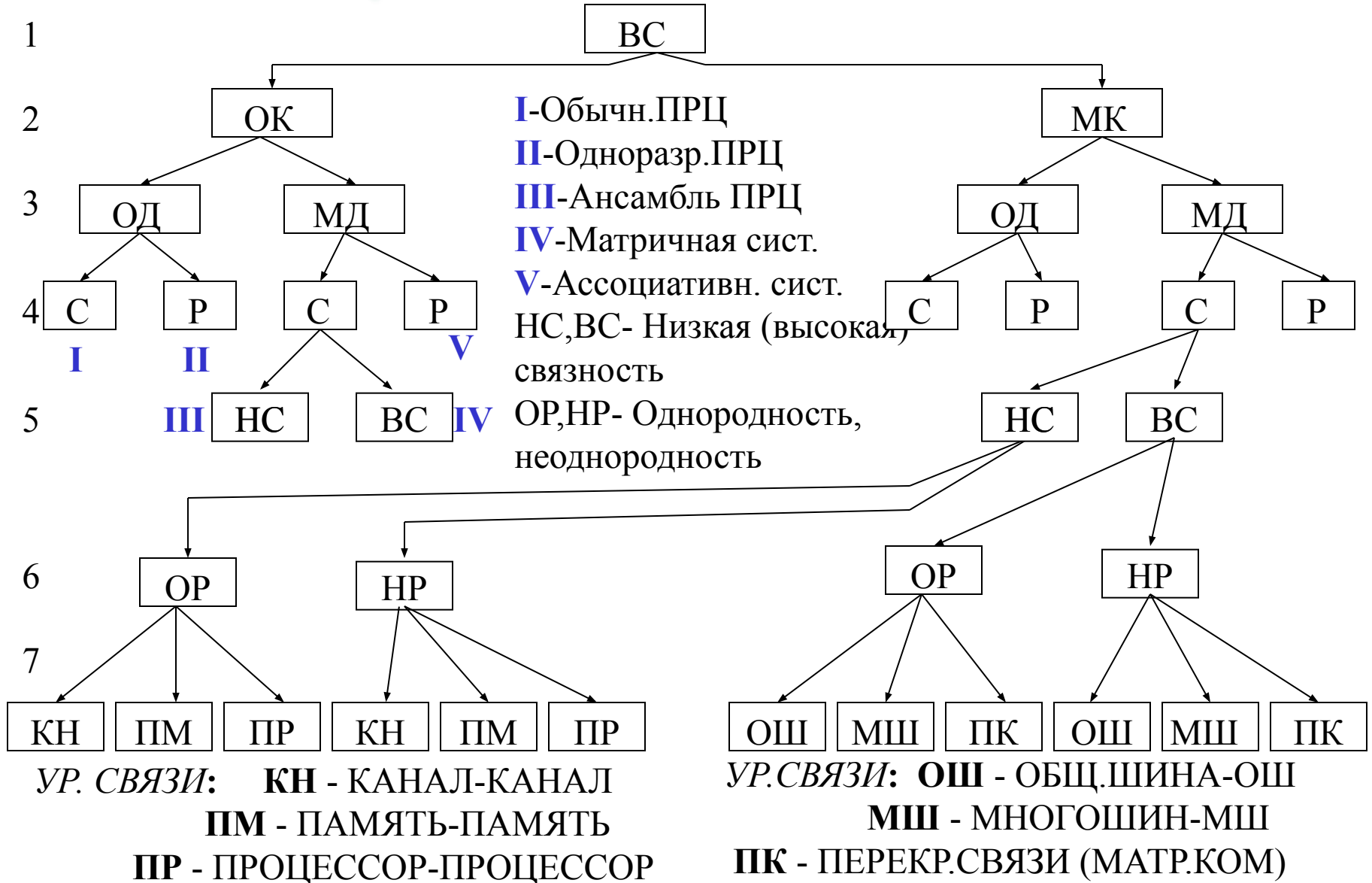


ЛОГИКА В ПАМЯТИ

КЛАССЫ КЛАССИФИКАЦИИ ШОРА

- **IV** – ансамбль ПРЦ получается путем интеграции модулей машины **I** в единую вычислительную структуру с общей шиной. Эта структура эффективна для обработки векторов, но по сравнению с машиной **I** скорость обработки $\ll \lg_2 M$
- **V** – матричная структура, получается введением наряду с ОШ прямых связей между соседними ПРЦ. Структура эффективна для обработки векторов и матриц, по сравнению с машиной **I** скорость обработки $\sim \lg_2 M$
- **VI** – объединяет логическую обработку с ассоциативным поиском прямо в ОП, поскольку в матричной ОП содержатся элементы логической обработки, которые осуществляют логические операции на проходе при доступе к ячейкам ОП

КЛАССИФИКАЦИЯ ПО СТЕПЕНИ ПАРАЛЛЕЛИЗМА ОБРАБОТКИ

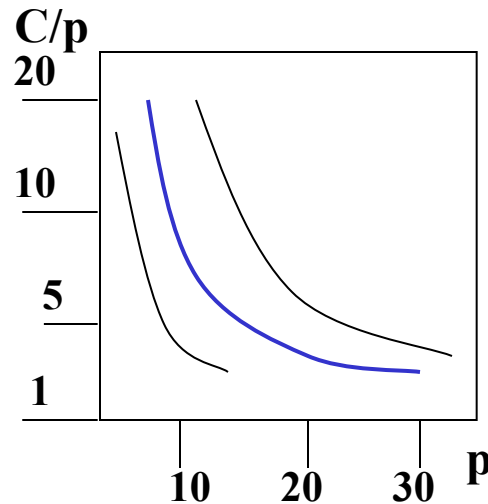
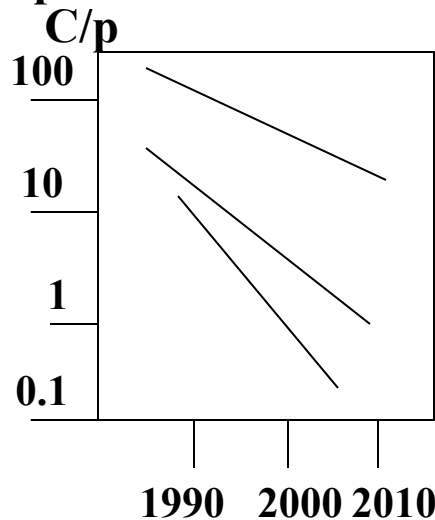


ЗАКОНЫ ПРОИЗВОДИТЕЛЬНОСТИ

Закон Мура – **Вычислительная мощность за данную цену удваивается каждые 18 месяцев**

Общая оценка ВС – Производительность/Стоимость
или **Стоимость единицы производительности**

Закон Гроша: Производительность и стоимость ВС связаны квадратичным законом: $p \sim C^2$



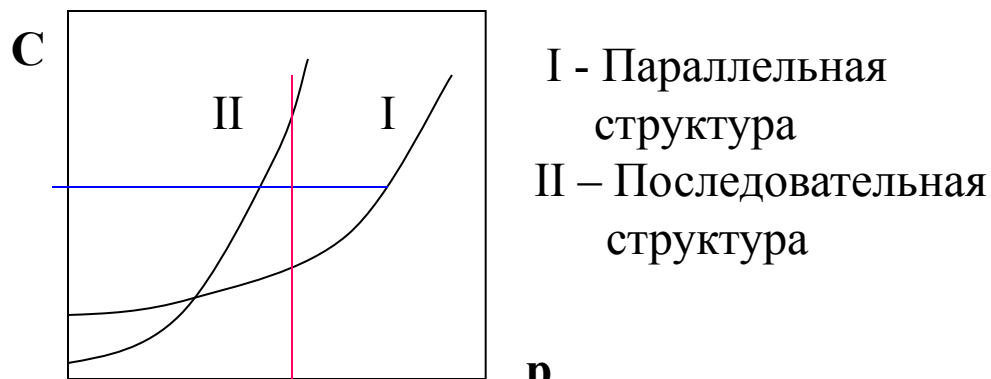
“Как бы ни старались разработчики HW, разработчики SW всегда сведут их усилия на нет. И это не предел.” Д.Платт

Стоимость единицы производительности со временем падает

Стоимость единицы производительности мощной ВМ всегда дешевле менее мощной для машин одного класса (сервера, рабочие станции, notebook)

ЗАКОНЫ ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРЫ

1. Параллельная ВС имеет более высокую производительность, чем последовательная при одинаковой стоимости.
2. Соотношение Стоимость/Производительность при одной и той же производительности всегда выше у последовательной ВС, чем у параллельной



I - Параллельная структура
II – Последовательная структура

3. Нарастиваемость и расширяемость P многомашинной архитектуры всегда выше за счет модульности и простого подключения дополнительных процессоров
4. Отказоустойчивость многомашинной архитектуры выше за счет рекофигурации и восстанавливаемости

Закон Джина Амдала – Любой поддающийся распараллеливанию процесс содержит часть, которая выполняется параллельно, и часть, которая выполняется последовательно. Если последовательная часть составляет долю X , то на бесконечном числе процессоров максимальное распараллеливание ограничено $1/X$. При доле 0.1 распараллеливание не больше, чем 10. Но на больших задачах, где доля последовательных компонент \ll параллельных, это ограничение влияет слабо.

ЗАКОНЫ ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРЫ

Закон Джина Амдала в более точной формулировке:

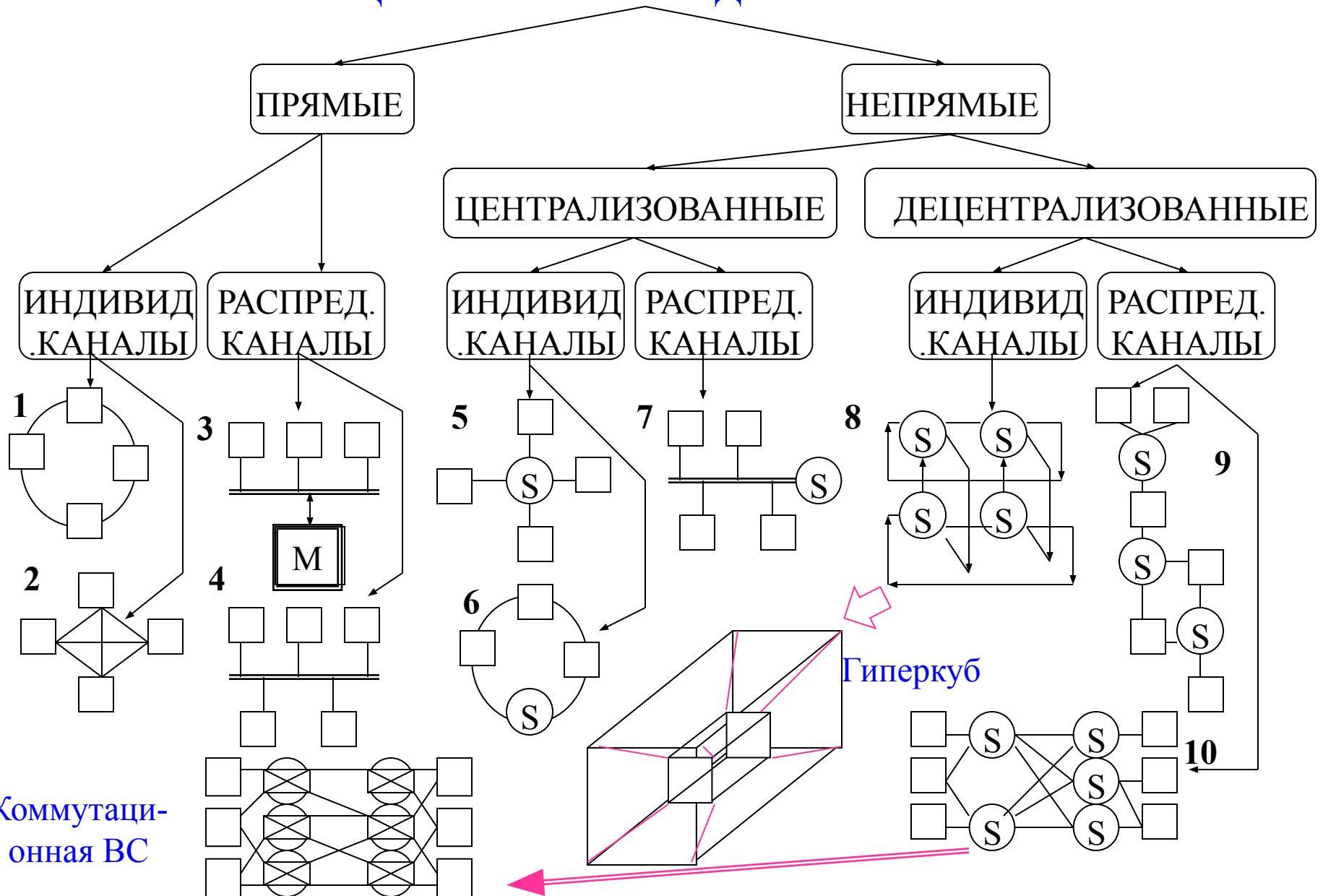
$$P = N / (X * N + 1 - X),$$

где X – последовательная часть вычислений,

N – число обработчиков (процессоров),

P – степень распараллеливания.

КЛАССИФИКАЦИЯ СВЯЗЕЙ МЕЖДУ ЭЛЕМЕНТАМИ ММА



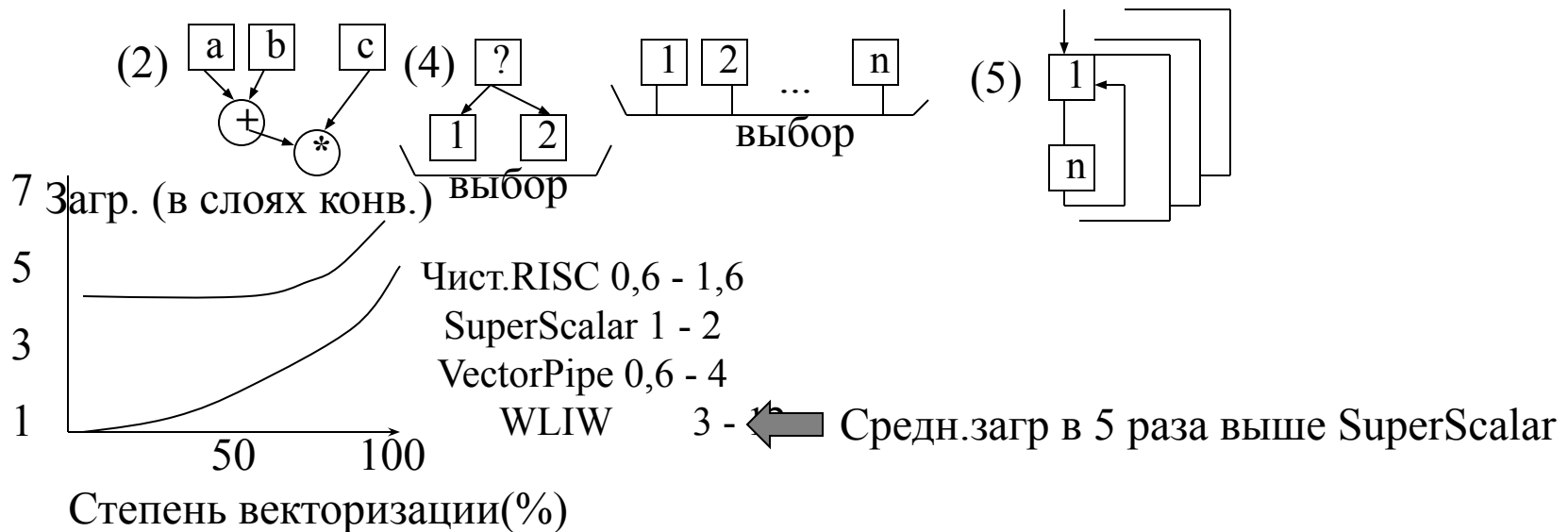
КЛАССИФИКАЦИЯ СВЯЗЕЙ МЕЖДУ ЭЛЕМЕНТАМИ ММА

| Имя структуры связей | Модуль-ность | Надежность | Скорость передачи |
|---|--------------|-------------------------------|-----------------------------|
| 1. Кольцевая без элементов коммутации | модуль-ная. | невысок, огр. ПРЦ транзитом | невысокая огр. ПРЦ трнз |
| 2. Полносвязная | немоду-льная | высокая | высокая |
| 3. Шина с разделением общей памяти | модуль-ная | невысок, огр. доступ. к ОП | средняя, огр. доступ. к ОП |
| 4. Общая шина без арбитра | модуль-ная | высокая | высокая огр. ОШ |
| 5. Звезда с центральным коммутатором | модуль-ная | высокая, огр. коммутатор. | высокая |
| 6. Кольцо с центральным коммутатором | модуль-ная | невысокая, огр. ПРЦ транзитом | невысокая, огр. ПРЦ трнз |
| 7. Общая шина с арбитром | модуль-ная | средняя, огр. арбитром | высокая, огр. ОШ и арбитр. |
| 8. Регулярная структура (Гиперкуб – представитель рег. структуры) | моду-льная | высокая | высокая |
| 9. Нерегулярная, специализированная структура | моду-льная | высокая, огр. элем. коммутац. | высокая, огр. элем. коммут. |
| 10. Структура с коммутацией каналов (Коммутационная машина) | моду-льная | высокая, огр. элем. коммутац. | высокая, огр. элем. коммут |

ЧТО ПОСЛЕ СУПЕРСКАЛЯРНОЙ АРХИТЕКТУРЫ

Суперскалярная архитектура обеспечивает параллелизм для традиционных последовательностей операций. Как ее улучшить:

1. Снять зависимость от избыточных обращений к памяти при промежуточных вычислениях Ю вести вычисления в регистрах (от 10 Pг Ю 100-1000 Pг)
2. Вычислять статически адреса a, b, c и сохранять их в стеке данных
3. Предсказывать переходы и заранее считывать программу и данные
4. Вычислять все ветки альтернатив и отбрасывать несостоявшиеся при вычислении условия(спекулятивные вычисления)
5. Подготовку всех статических вычислений поручить транслятору, а динамику вычислять аппаратно с использованием тегов.



ЧТО ПОСЛЕ СУПЕРСКАЛЯРНОЙ АРХИТЕКТУРЫ

- СУПЕРСКАЛЯРНУЮ АРХИТЕКТУРУ сменит Мультимашинная (MultyCore)