

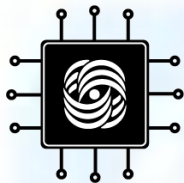
АРХИТЕКТУРА СОВРЕМЕННЫХ КОМПЬЮТЕРОВ

Лекция 7:

Параллельные компьютерные архитектуры

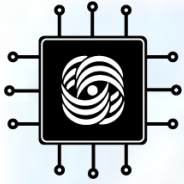
ВМК МГУ им. М.В. Ломоносова, Кафедра АСВК

Доцент, к.ф.-м.н. Волканов Д.Ю.



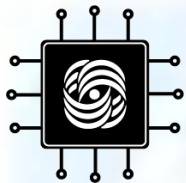
План лекции

- Архитектуры параллельных компьютеров
- Параллелизм на уровне команд
- Многопоточность
- Мультипроцессоры
- Мультикомпьютеры
- Классификация Флинна

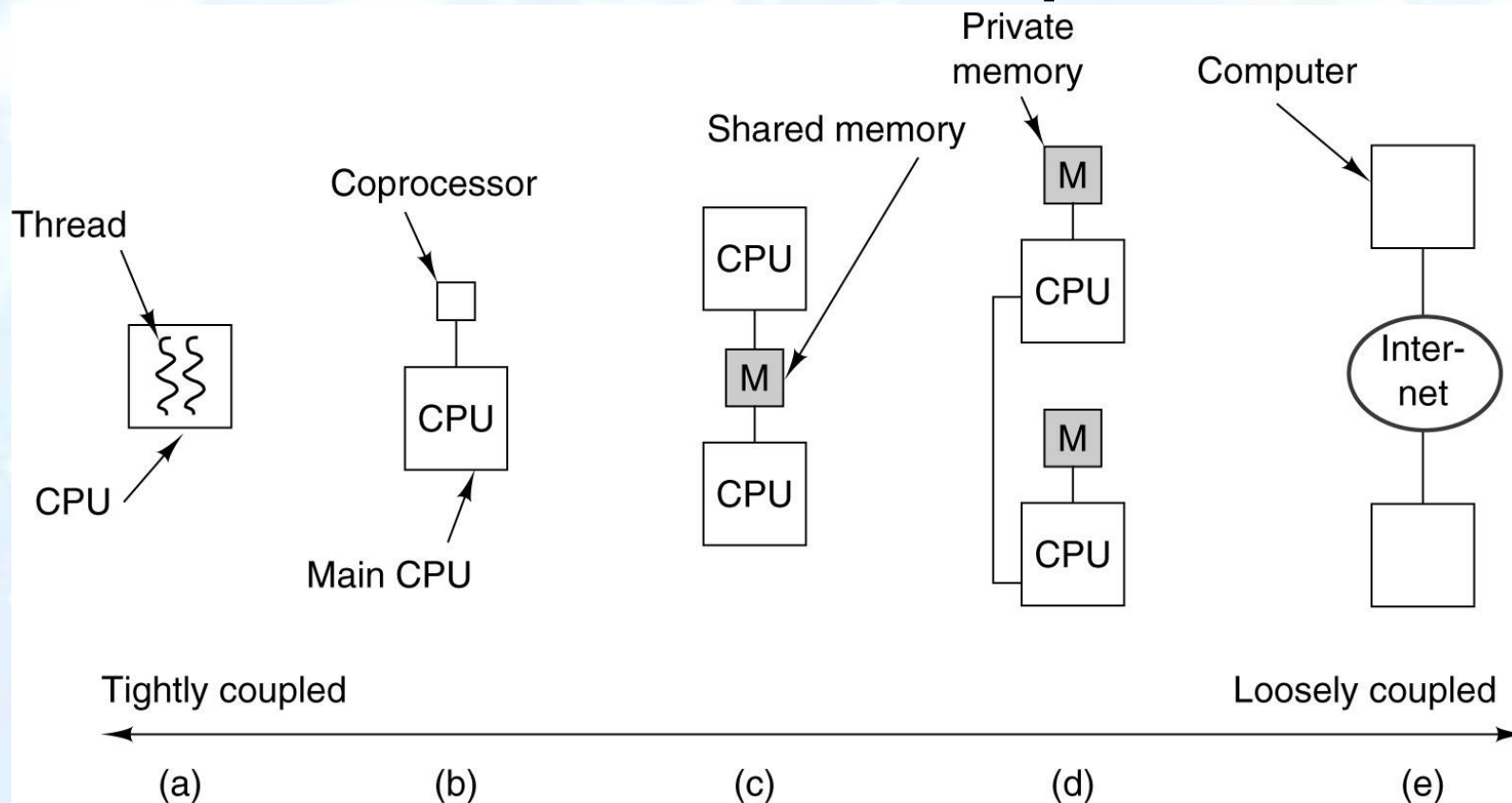


Уровни параллелизма

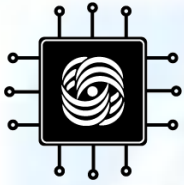
- В процессоре
 - Конвейризация
 - Суперскалярность
 - Удлинение длины команд
- Дополнительные специализированные процессоры
- Многоядерность
- Многопроцессорность (Сильносвязанный параллелизм)
- Кластеры (Слабосвязанный параллелизм)



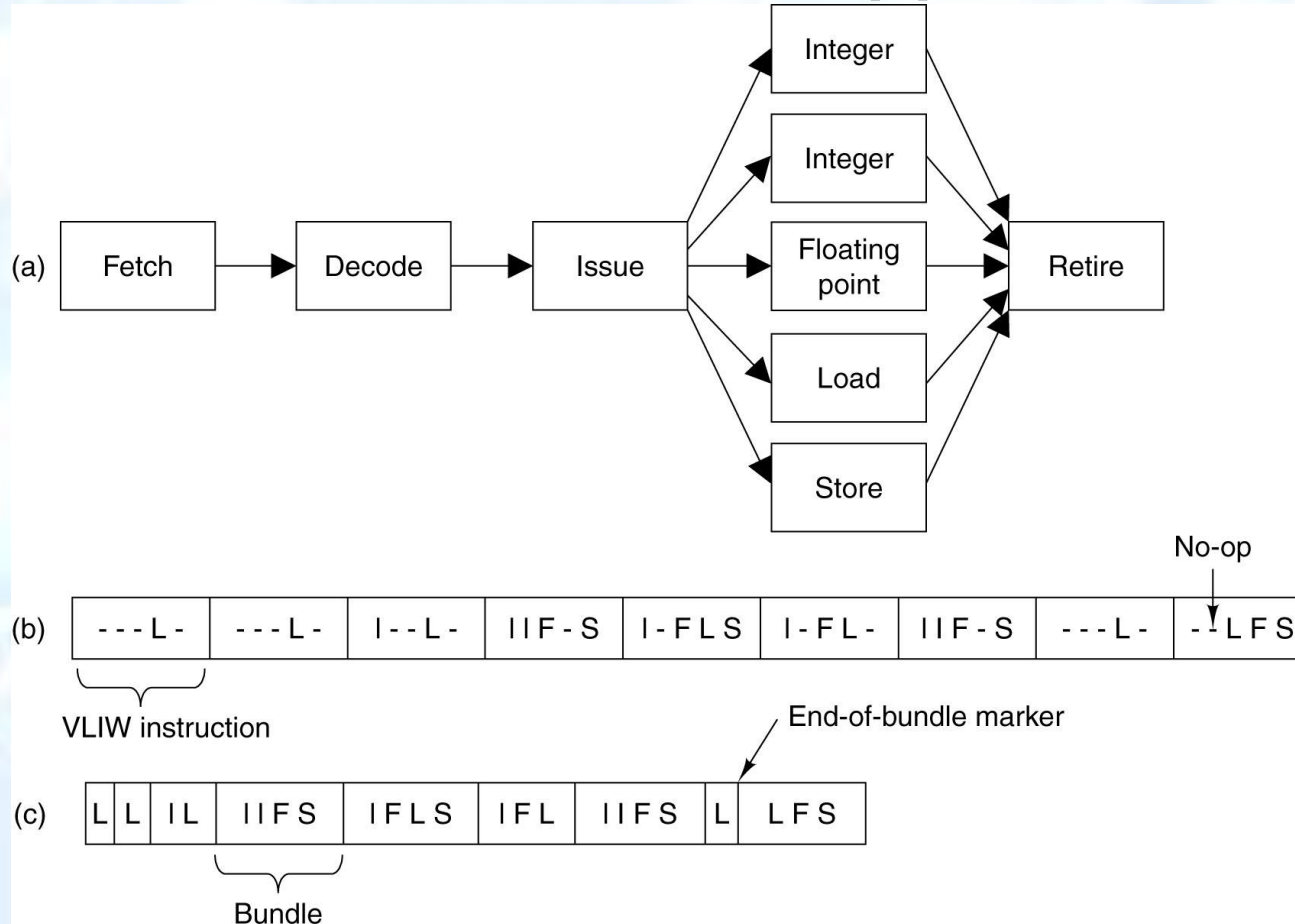
Архитектуры параллельных компьютеров



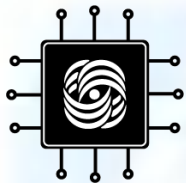
(a) На чипе (b) Сопроцессор (c) Мультипроцессор
(d) Мультикомпьютер (e) Грид



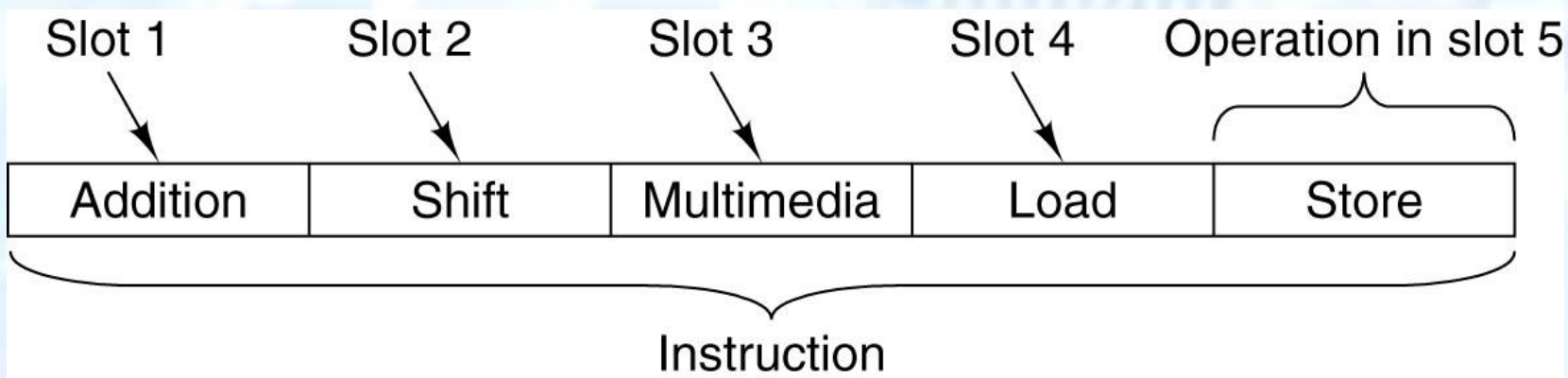
Параллелизм на уровне команд



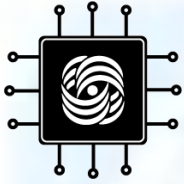
(a) Конвейер (b) Псл-ть VLIW инст-кций
(c) Поток команд с отмеченными связками⁵



TriMedia VLIW команда

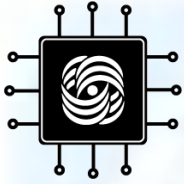


Типичная команда TriMedia.



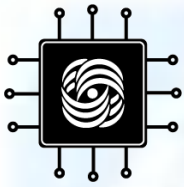
Функциональные блоки

- Операции с константами (5)
- АЛУ целочисленных операций (5)
- Сдвиги (2)
- Загрузка и сохранение из памяти (2)
- Умножение целых и вещественных чисел (2)
- АЛУ операций с плавающей точкой (2)
- Сравнение чисел с плавающей точкой (1)
- **Извлечение квадратного корня (1)**
- Ветвления (3)
- АЛУ ЦОС (2)
- Умножитель для ЦОС (2)

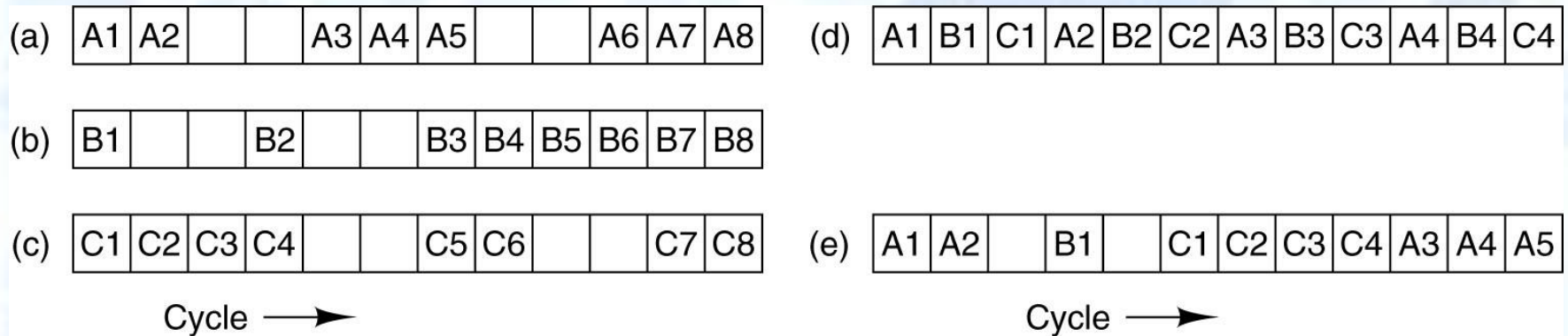


Особенности

- Арифметика с насыщением
- Команды 2-28 байт
- Отсутствует проверка на совместимость операций во время исполнения
- Прогнозирование операций
- Специализированные операции



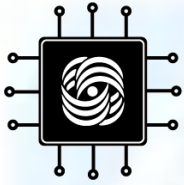
Внутрипроцессорная многopotочность (1)



(a) – (c) Три потока. Пустые квадраты означают простой в ожидании данных из памяти

многopotочность

(e) Крупномодульная многopotочность⁹



Внутрипроцессорная многопоточность (2)

A1	B1	C1	A3	B2	C3	A5	B3	C5	A6	B5	C7
A2		C2	A4		C4		B4	C6	A7	B6	C8

Cycle →

(a)

A1	B1	C1	C3	A3	A5	B2	C5	A6	A8	B3	B5
A2		C2	C4	A4			C6	A7		B4	B6

Cycle →

(b)

A1	B1	C2	C4	A4	B2	C6	A7	B3	B5	B7	C7
A2	C1	C3	A3	A5	C5	A6	A8	B4	B6	B8	C8

Cycle →

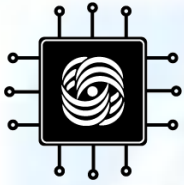
(c)

Многопоточность в сдвоенном процессоре

(a) Мелкомодульная многопоточность

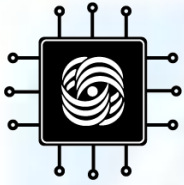
(b) Крупномодульная многопоточность

(c) Синхронная многопоточность



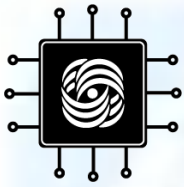
Варианты повышения производительности

- Повышение тактовой частоты
- Размещение на одной микросхеме двух процессоров
- Введение новых функциональных блоков
- Удлинение конвейера
- Использование многопоточности

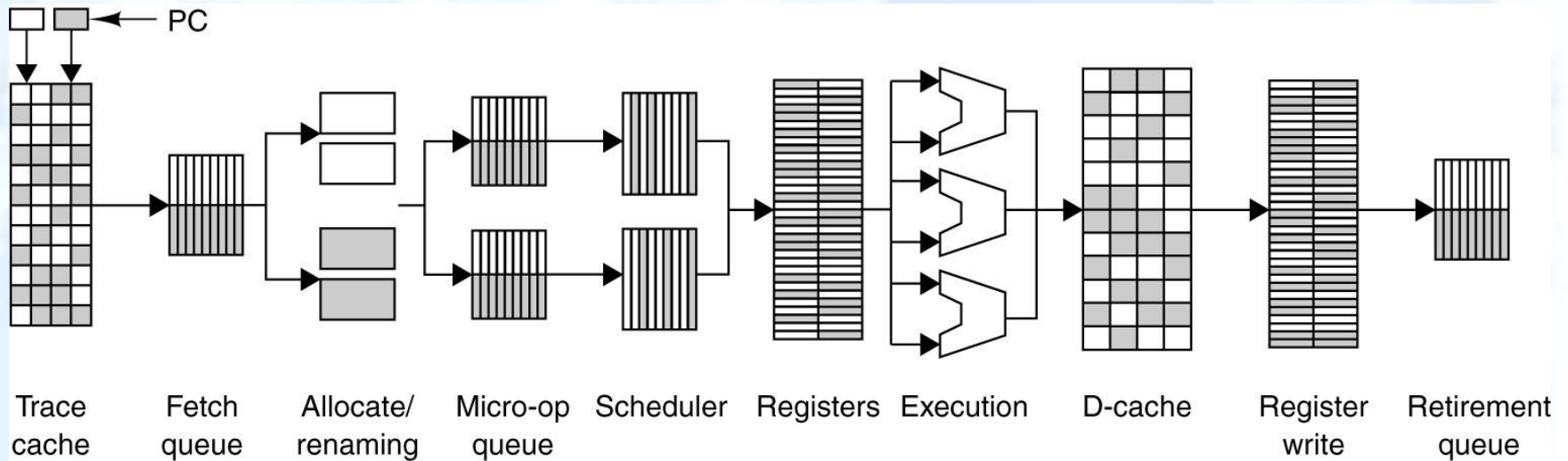


Стратегии управления совместным потреблением ресурсов

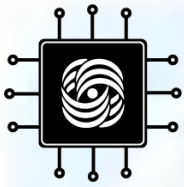
- Дублирование ресурсов
- Жёсткое разделение ресурсов
- Полное разделение ресурсов
- Пороговое разделение ресурсов



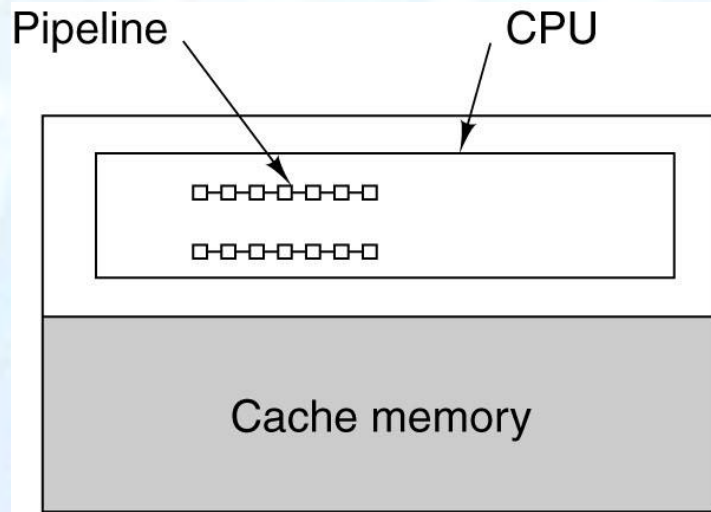
МНОГОПОТОЧНОСТЬ В Pentium 4



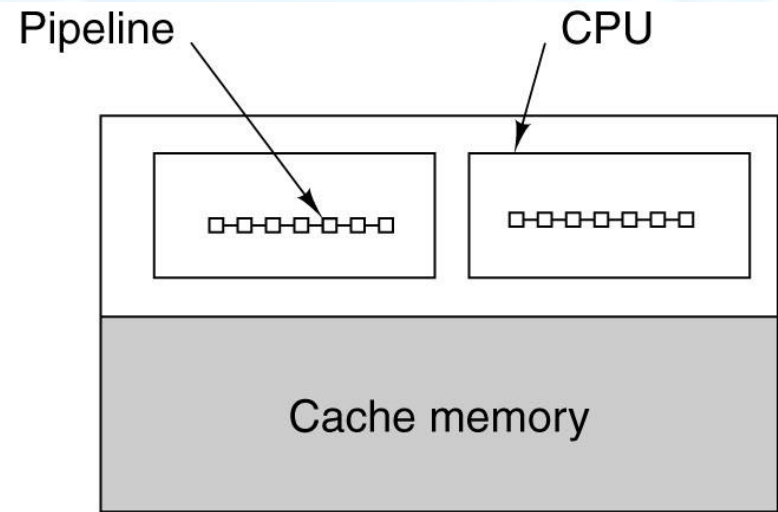
Разделение ресурсов между программными потоками
в микроархитектуре NetBurst Pentium 4.



Гомогенные однокристальные микропроцессоры



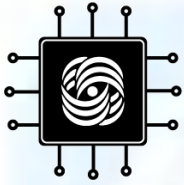
(a)



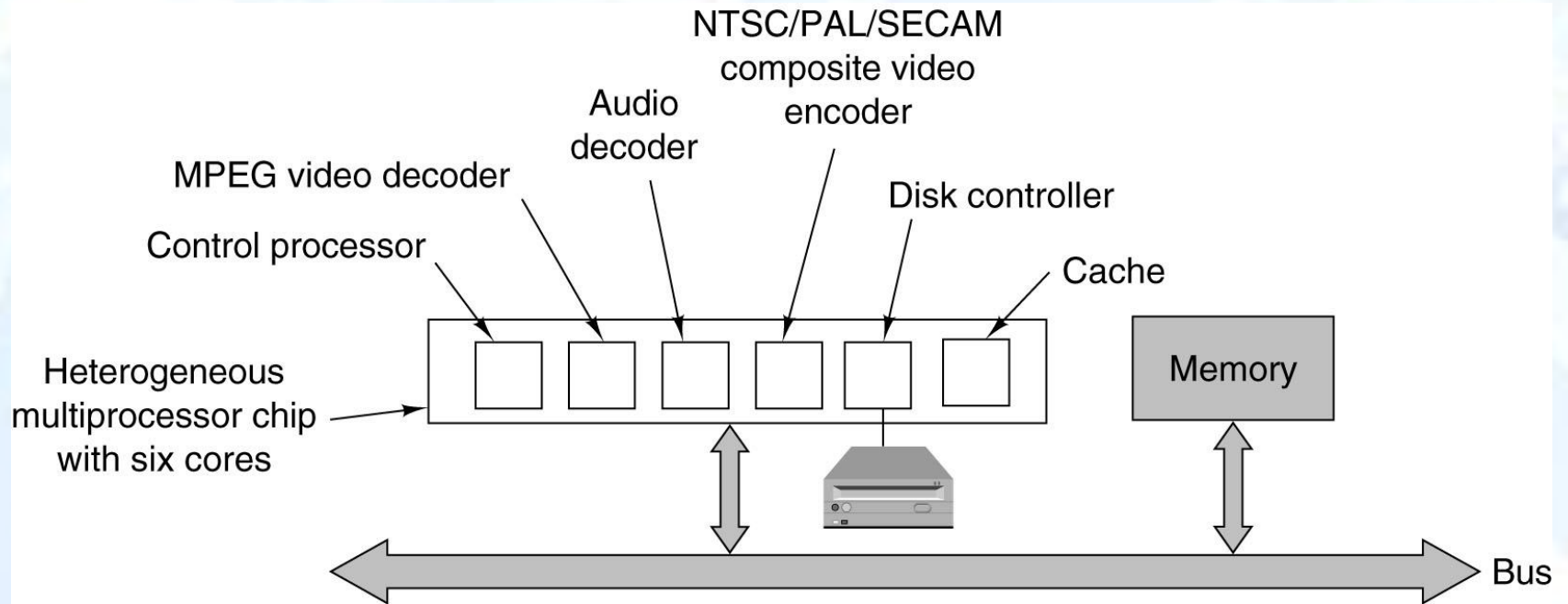
(b)

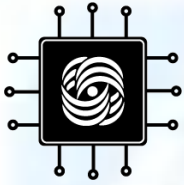
Однокристальные мультипроцессоры.

(a) Два конвейера (b) Два ядра

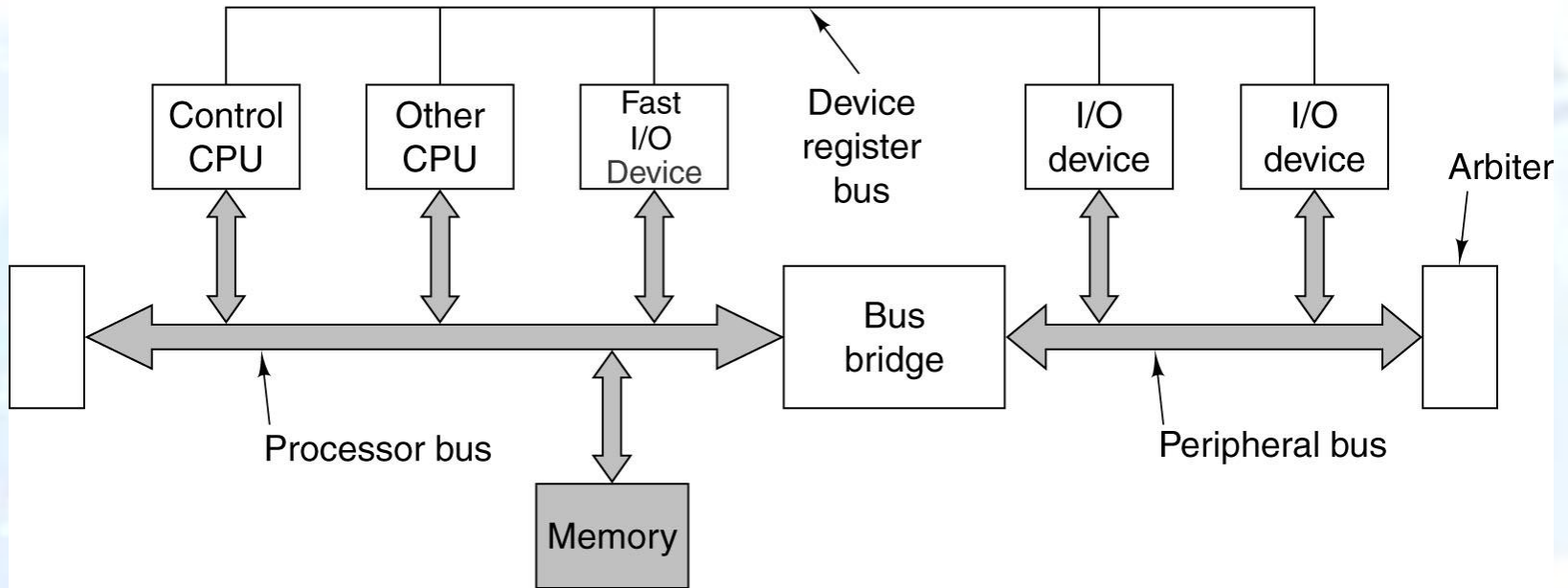


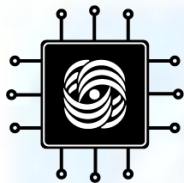
Гетерогенные однокристальные мультипроцессоры



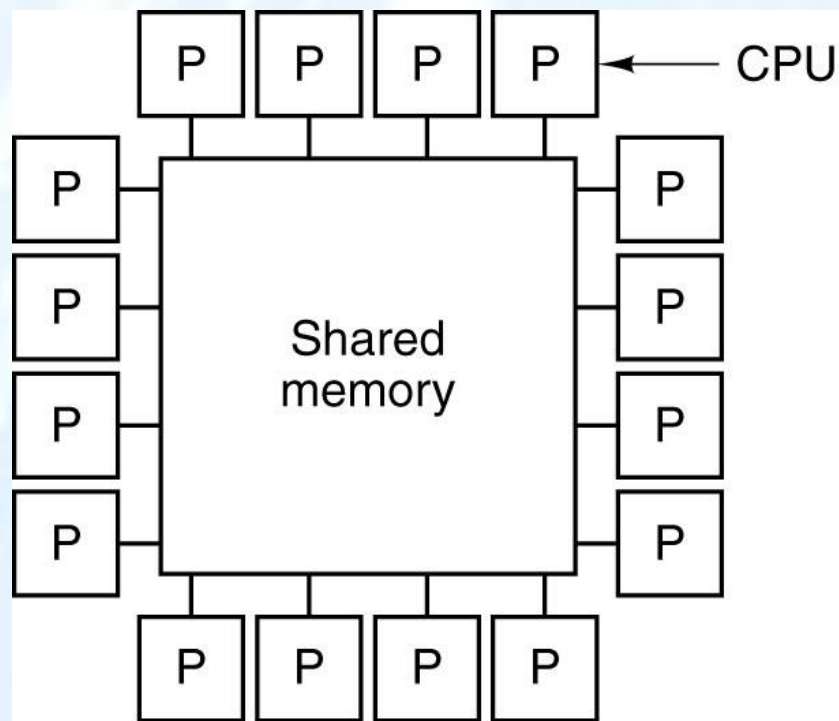


Архитектура CoreConnect

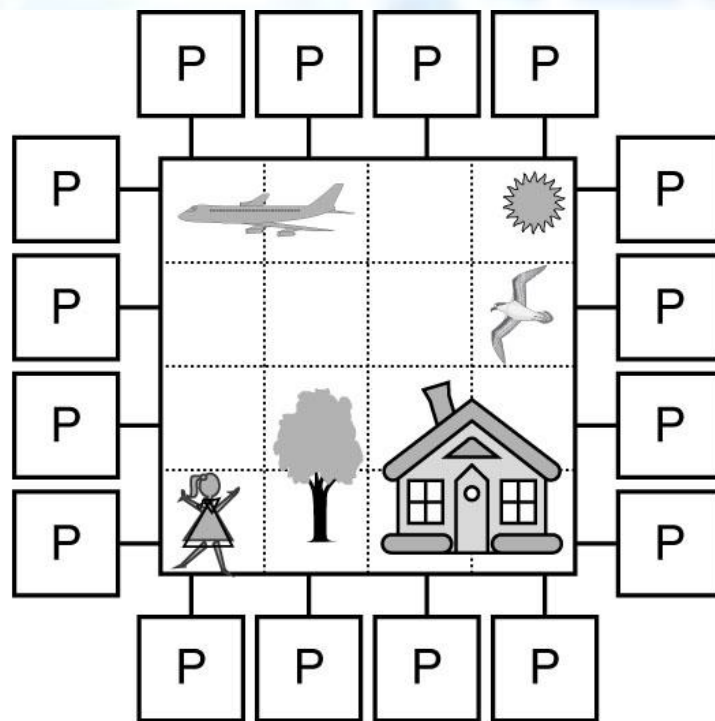




Мультимикропроцессоры

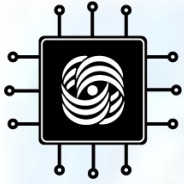


(a)

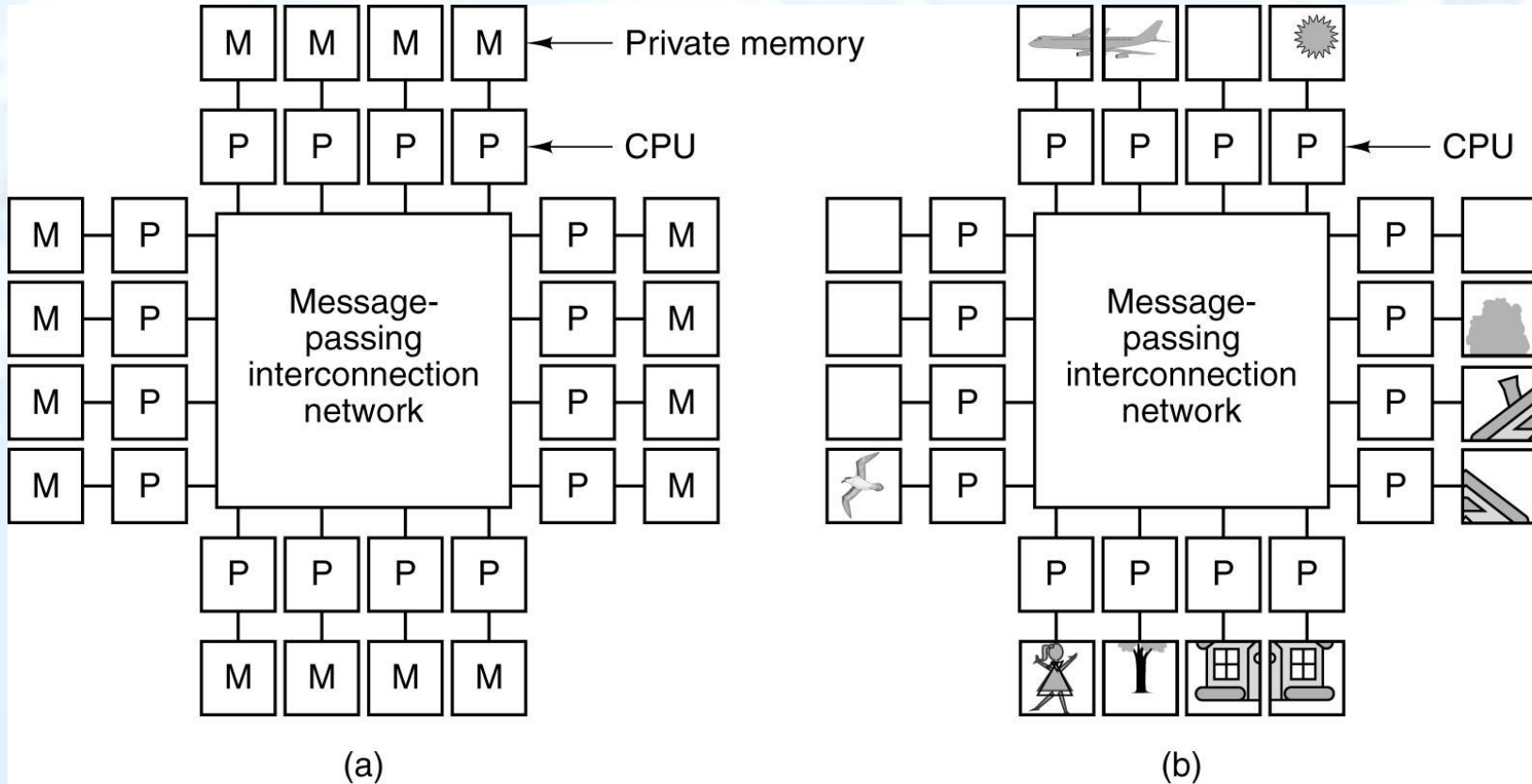


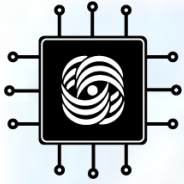
(b)

- (a) Мультимикропроцессор с 16 ЦПУ, разделяющих общую память
- (b) Изображение, разбивается на 16 частей, каждое обрабатывается своим ЦПУ

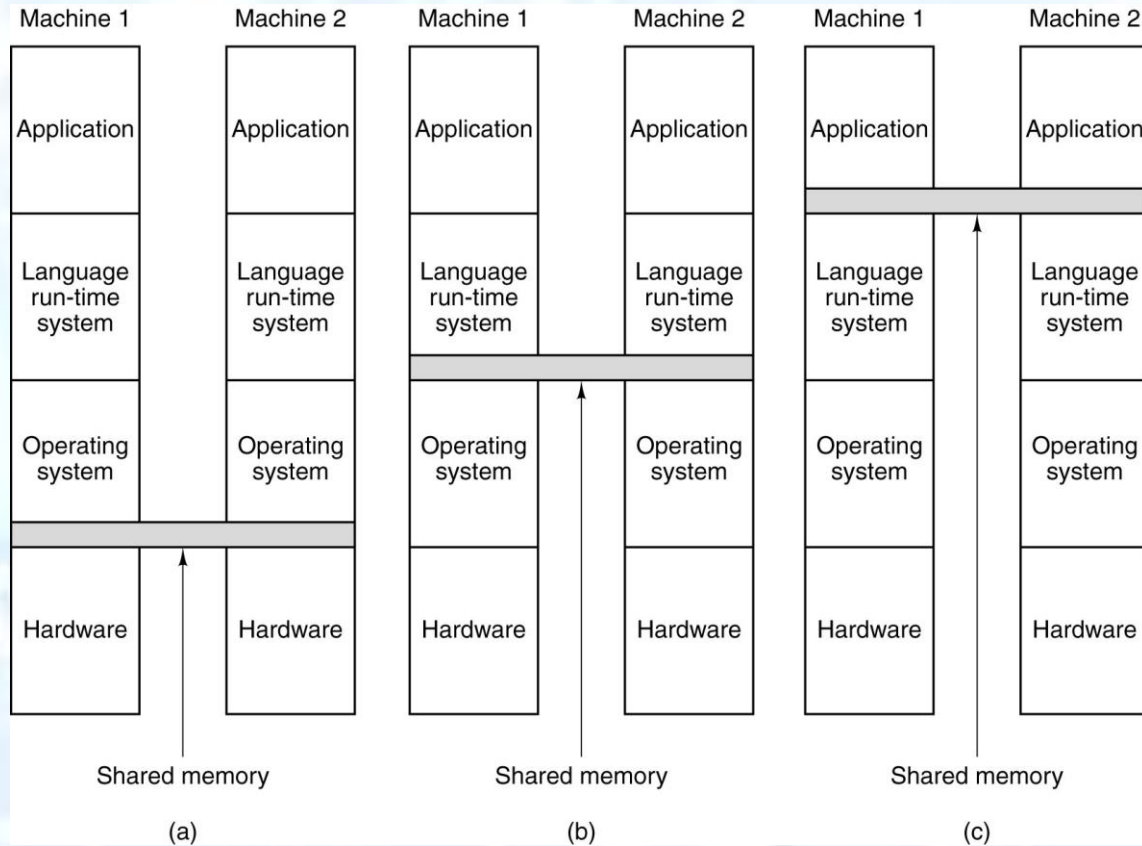


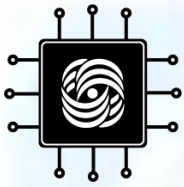
Мультикомпьютеры (1)





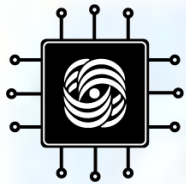
Мультикомпьютеры (2)



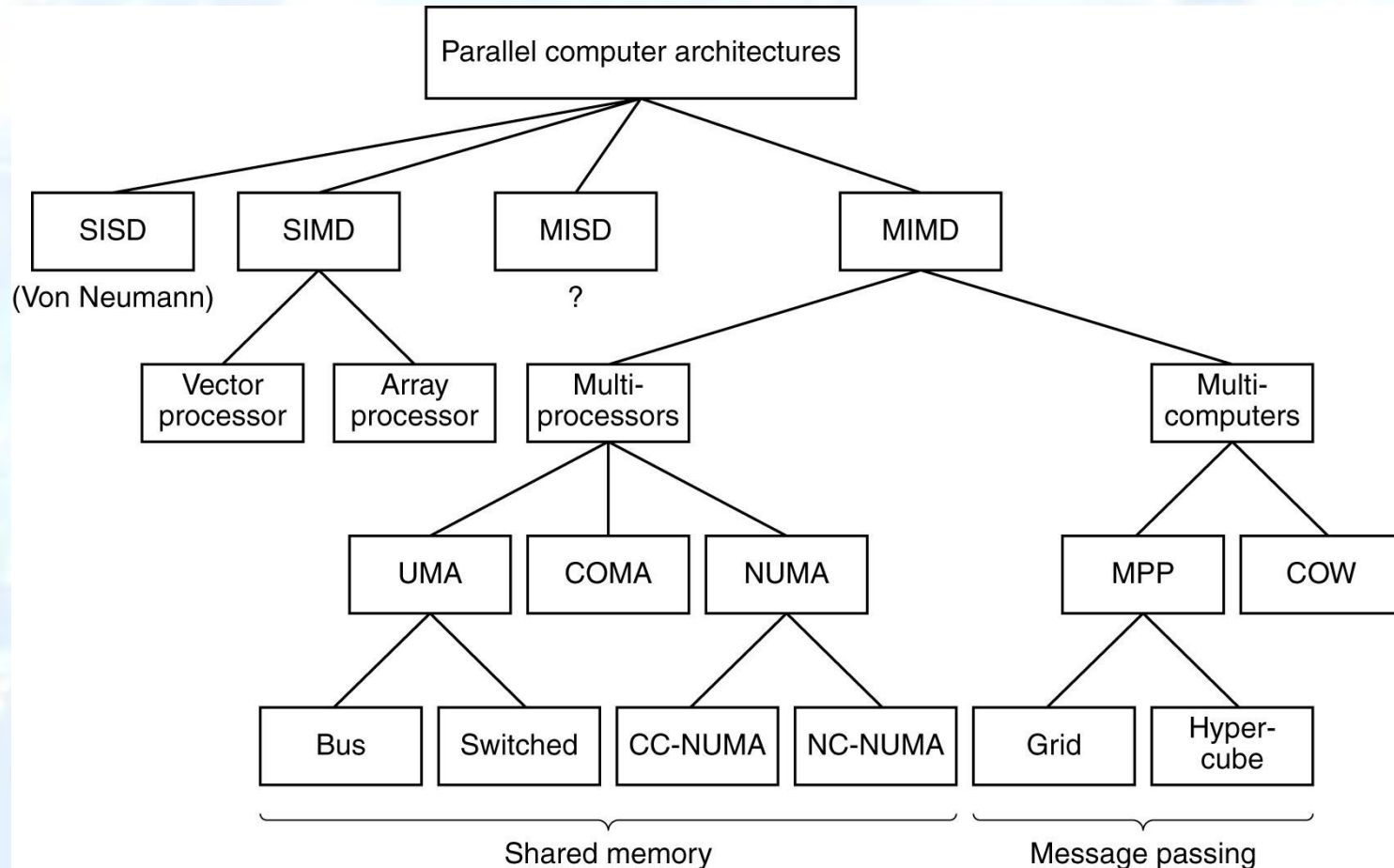


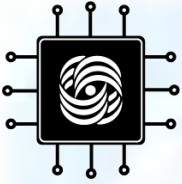
Классификация Флинна (1)

Instruction streams	Data streams	Name	Examples
1	1	SISD	Classical Von Neumann machine
1	Multiple	SIMD	Vector supercomputer, array processor
Multiple	1	MISD	Arguably none
Multiple	Multiple	MIMD	Multiprocessor, multicomputer



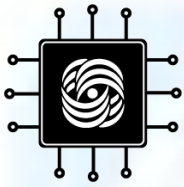
Классификация Флинна (2)





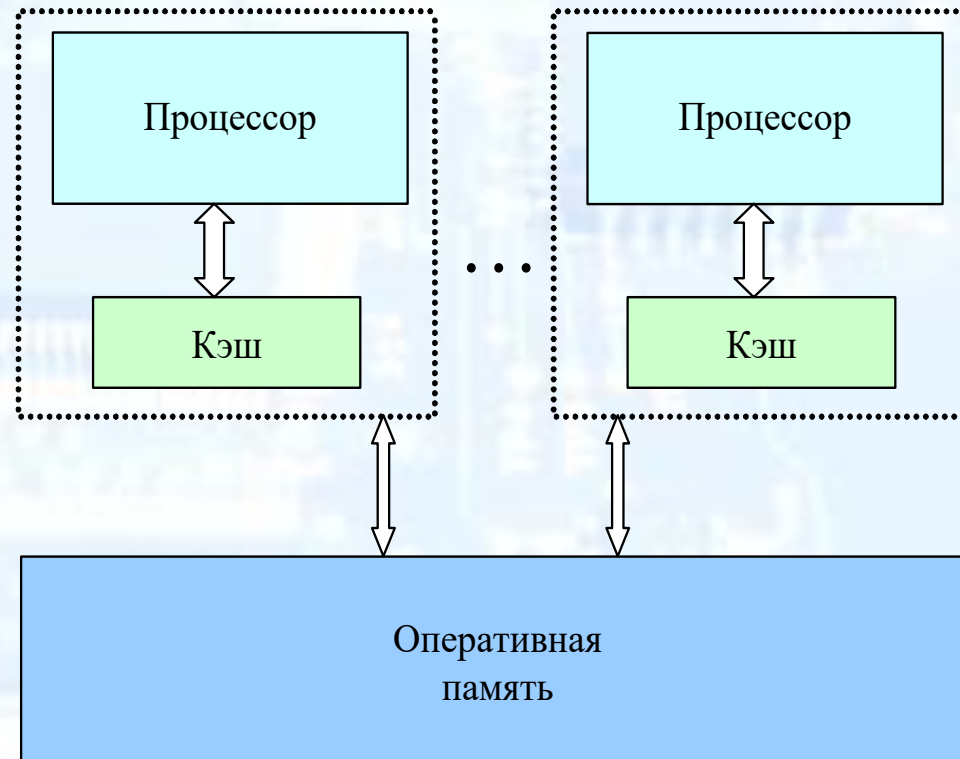
Классификация вычислительных систем

- **Мультипроцессоры** с использованием единой общей памяти (*shared memory*)...
 - Обеспечивается *однородный доступ к памяти (uniform memory access or UMA)*,
 - Являются основой для построения:
 - *векторных параллельных процессоров (parallel vector processor or PVP)*. Примеры: Cray T90,
 - *симметричных мультипроцессоров (symmetric multiprocessor or SMP)*. Примеры: IBM eServer, Sun StarFire, HP Superdome, SGI Origin.



Классификация вычислительных систем

- **Мультипроцессоры** с использованием единой общей памяти...



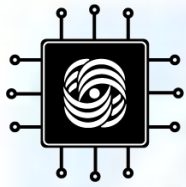


Классификация вычислительных систем

- **Мультипроцессоры** с использованием единой *общей памяти...*

Проблемы:

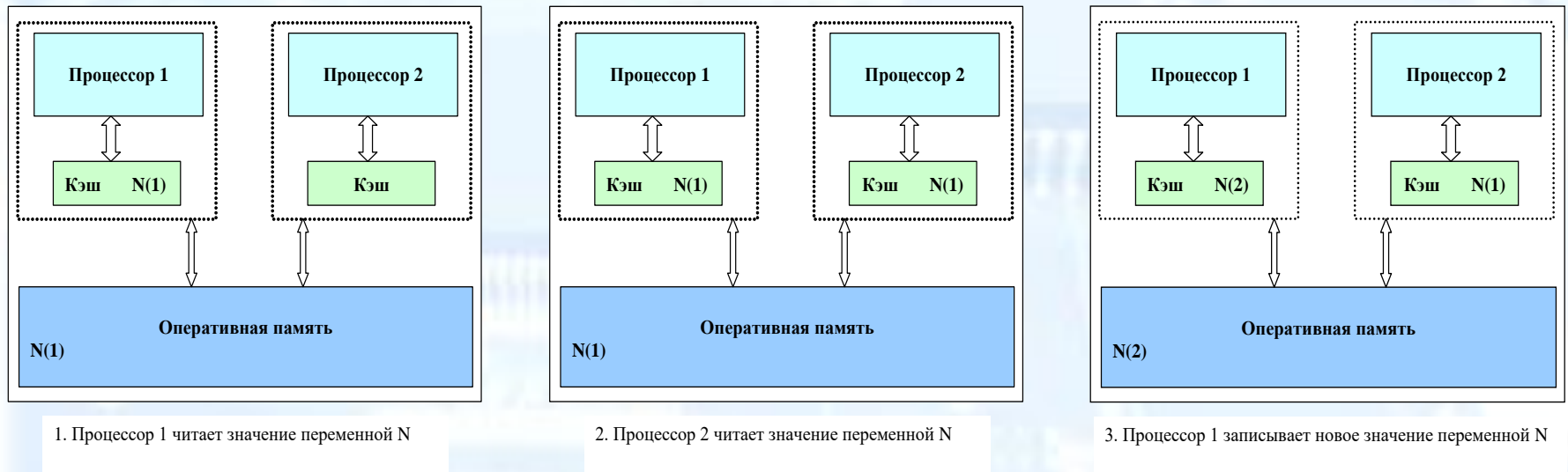
- Доступ с разных процессоров к общим данным и обеспечение, в этой связи, *однозначности (когерентности) содержимого разных кэшей (cache coherence problem),*
- Необходимость *синхронизации взаимодействия* одновременно выполняемых потоков команд



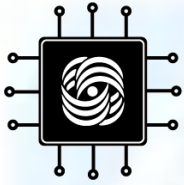
Классификация вычислительных систем

- **Мультипроцессоры** с использованием единой общей памяти...

Проблема: Обеспечение однозначности (когерентности) содержимого разных кэшей (*cache coherence problem*)



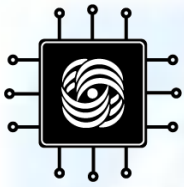
При изменении данных необходимо проверять наличие "старых" значений в кэш-памяти всех процессоров (обеспечивается на аппаратном уровне, но становится сложным при большом количестве процессоров)



Классификация вычислительных систем

- **Мультипроцессоры** с использованием единой *общей памяти...*

Проблема: *Необходимость синхронизации взаимодействия одновременно выполняемых потоков команд...*



Классификация вычислительных систем

Пример: Пусть процессоры выполняют последовательность команд

$N = N + 1$

Печать N

над общей переменной N (в скобках указывается значение этой переменной)

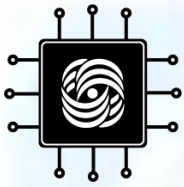
Вариант исполнения 1

Время	Процессор 1	Процессор 2
1	Чтение N (1)	
2		Чтение N (1)
3		Прибавление 1 (2)
4	Прибавление 1 (2)	
5	Запись N (2)	
6	Печать N (2)	
7		Запись N (2)
8		Печать N (2)

Вариант исполнения 2

Время	Процессор 1	Процессор 2
1	Чтение N (1)	
2	Прибавление 1 (2)	
3	Запись N (2)	
4	Печать N (2)	
5		Чтение N (2)
6		Прибавление 1 (3)
7		Запись N (3)
8		Печать N (3)

Временная последовательность команд может быть различной – необходима синхронизация при использовании общих переменных !



Классификация вычислительных систем

- **Мультипроцессоры** с использованием единой *общей памяти...*

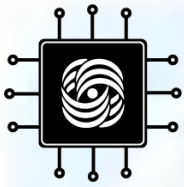
Проблема: *Необходимость синхронизации взаимодействия одновременно выполняемых потоков команд...*

- Рассмотренный пример может рассматриваться как проявление *общей проблемы использования разделяемых ресурсов* (общих данных, файлов, устройств и т.п.)



Классификация вычислительных систем

- Для **организации разделения ресурсов** между несколькими потоками команд необходимо иметь возможность:
 - *определения доступности* запрашиваемых ресурсов (ресурс свободен и может быть выделен для использования, ресурс уже занят одним из потоков и не может использоваться дополнительно каким-либо другим потоком);
 - *выделения свободного ресурса* одному из процессов, запросивших ресурс для использования;
 - *приостановки (блокировки) потоков*, выдавших запросы на ресурсы, занятые другими потоками.



Классификация вычислительных систем

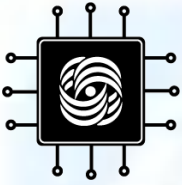
- **Мультипроцессоры** с использованием единой *общей памяти*

Проблема: *Необходимость синхронизации взаимодействия одновременно выполняемых потоков команд*

- Доступ к общей переменной в рассмотренном примере в самом общем виде должен быть организован следующим образом:

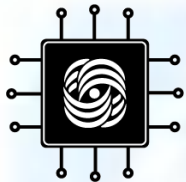
```
<Получить доступ>  
N = N + 1  
Печать N  
<Завершить доступ>
```

- Полное рассмотрение проблемы синхронизации будет выполнено позднее при изучении вопросов параллельного программирования для вычислительных систем с общей памятью



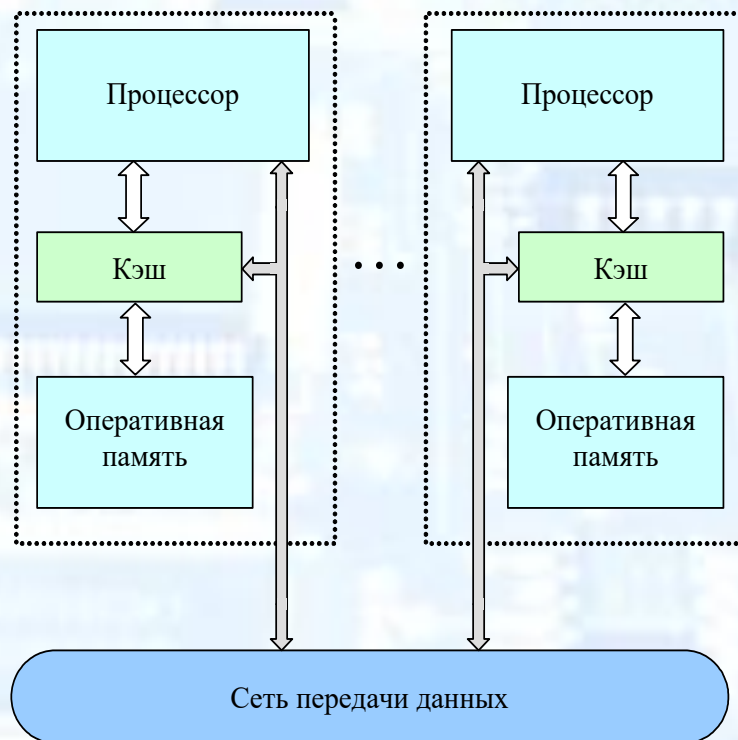
Классификация вычислительных систем

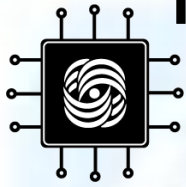
- **Мультипроцессоры** с использованием физически распределенной памяти (*distributed shared memory* or *DSM*):
 - Неоднородный доступ к памяти (*non-uniform memory access* or *NUMA*),
 - Среди систем такого типа выделяют:
 - *cache-only memory architecture* or *COMA* (системы KSR-1 и DDM),
 - *cache-coherent NUMA* or *CC-NUMA* (системы SGI Origin 2000, Sun HPC 10000, IBM/Sequent NUMA-Q 2000),
 - *non-cache coherent NUMA* or *NCC-NUMA* (система Cray T3E).



Классификация вычислительных систем

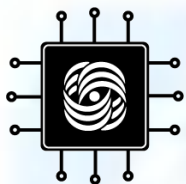
- **Мультимикропроцессоры** с использованием физически распределенной памяти...





Классификация вычислительных систем

- **Мультипроцессоры** с использованием физически распределенной памяти:
 - Упрощаются проблемы создания мультипроцессоров (известны примеры систем с несколькими тысячами процессоров)
 - Возникают проблемы эффективного использования распределенной памяти (время доступа к локальной и удаленной памяти может различаться на несколько порядков).

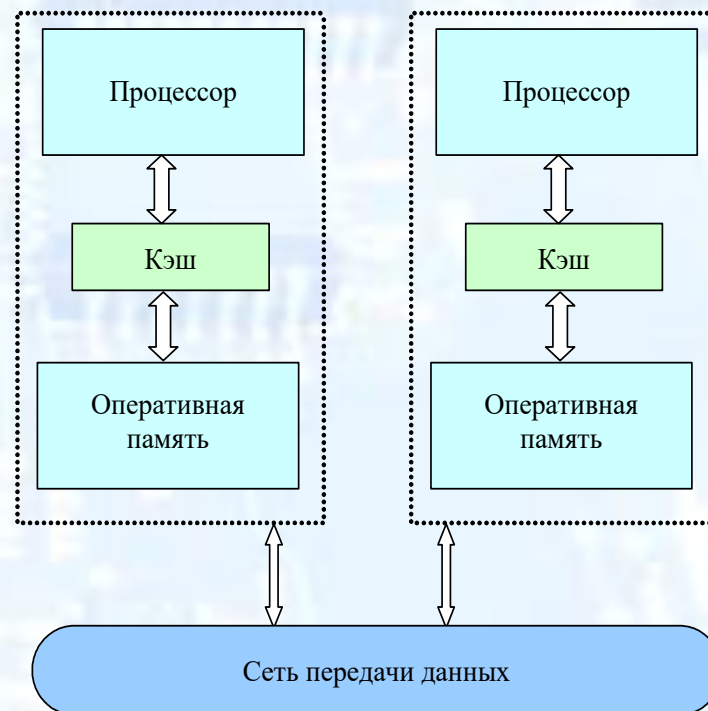


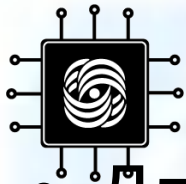
Классификация вычислительных систем

• Мультикомпьютеры

...

- Не обеспечивают общий доступ ко всей имеющейся в системах памяти (*no-remote memory access or NORMA*),
- Каждый процессор системы может использовать только свою локальную память





Мультикомпьютеры

- Для доступа к данным, располагаемым на других процессорах, необходимо явно выполнить *операции передачи сообщений (message passing operations)*,
- Основные операции передачи данных:
 - Отправить сообщение (*send*),
 - Получить сообщение (*receive*)

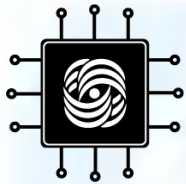
Пример:

Процессор 1

<Отправить сообщение>
<Продолжение вычислений>

Процессор 2

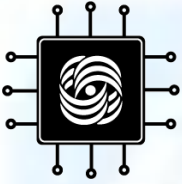
<Получить сообщение>
<Продолжение вычислений
с использованием данных
полученного сообщения>



Мультикомпьютеры

Данный подход используется при построении двух важных типов многопроцессорных вычислительных систем:

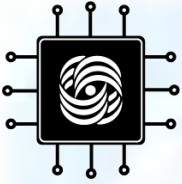
- *массивно-параллельных систем (massively parallel processor or MPP)*, например: IBM RS/6000 SP2, Intel PARAGON, ASCI Red, транспьютерные системы Parsytec,
- *кластеров (clusters)*, например: AC3 Velocity и NCSA NT Supercluster.



Кластеры

- **Преимущества:**

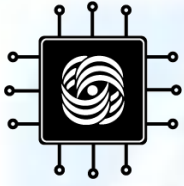
- Могут быть образованы на базе уже существующих у потребителей отдельных компьютеров, либо же сконструированы из типовых компьютерных элементов;
- Повышение вычислительной мощности отдельных процессоров позволяет строить кластеры из сравнительно небольшого количества отдельных компьютеров (*lowly parallel processing*),
- Для параллельного выполнения в алгоритмах достаточно выделять только крупные независимые части расчетов (*coarse granularity*).



Кластеры

- **Недостатки:**

- Организация взаимодействия вычислительных узлов кластера при помощи передачи сообщений обычно приводит к значительным временным задержкам
- Дополнительные ограничения на тип разрабатываемых параллельных алгоритмов и программ (*низкая интенсивность потоков передачи данных*)



Спасибо за внимание!