

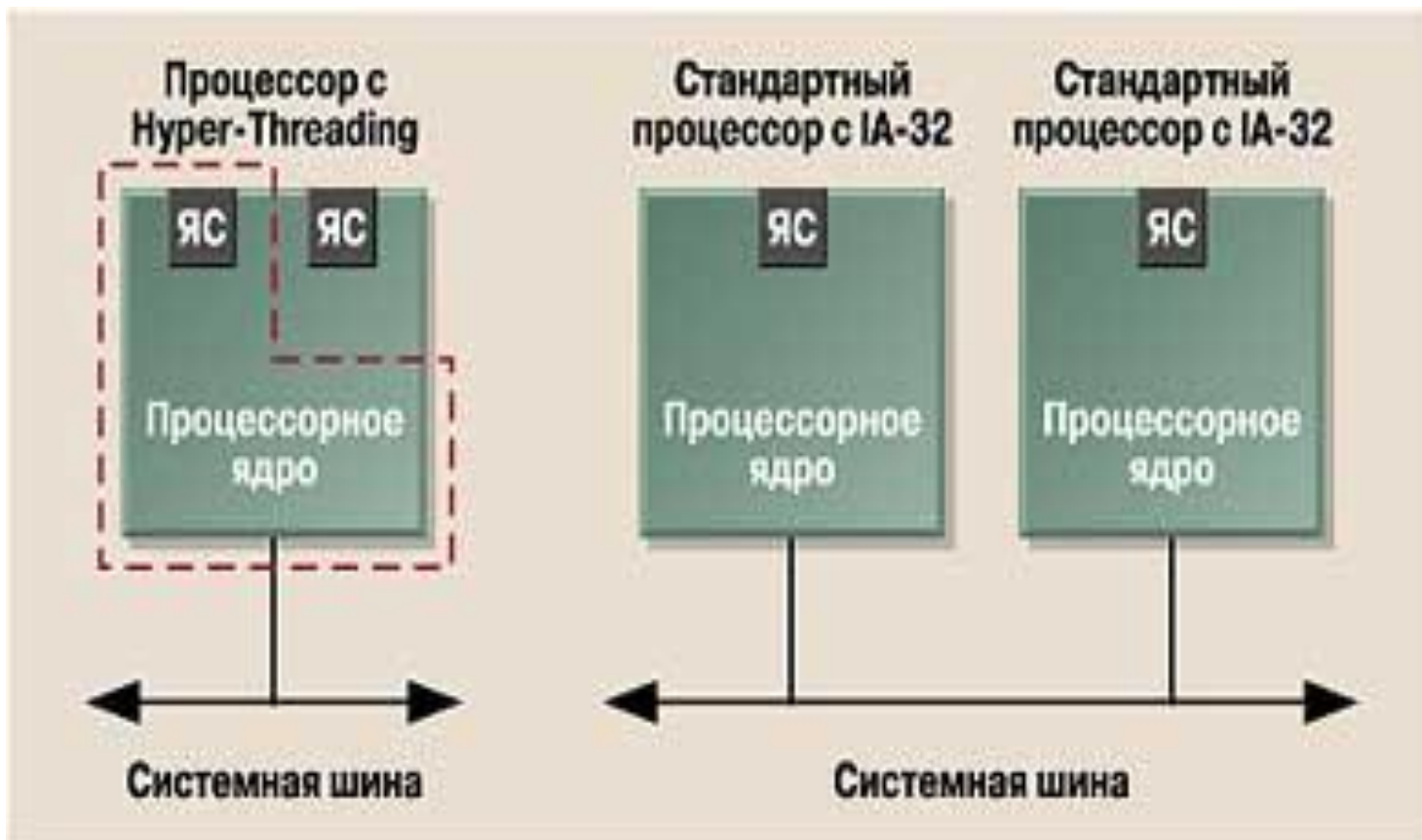
# Современные микропроцессоры

# Технология **Hyper-Threading**

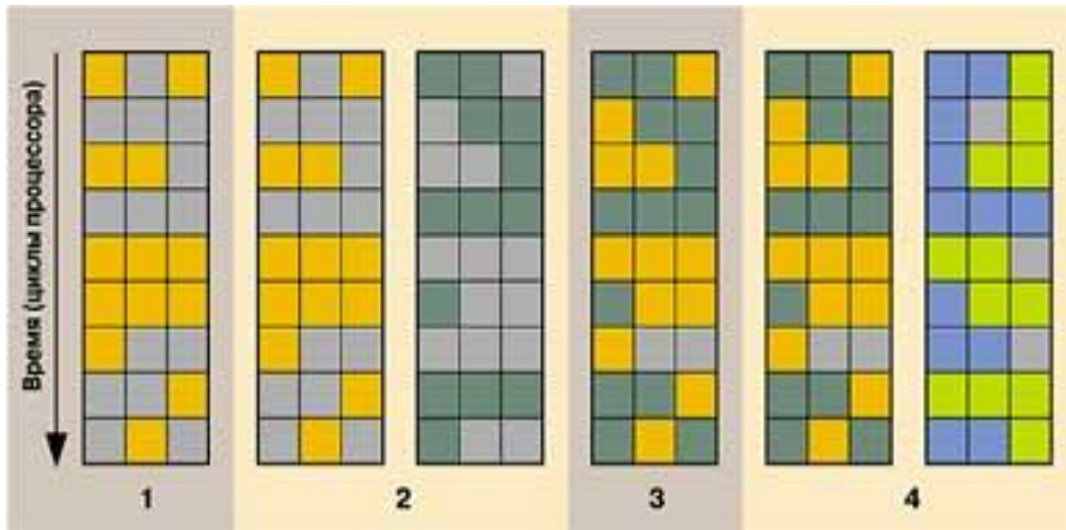
Главная цель применения **Hyper-Threading** — не выполнение двух (нескольких) задач одновременно, а **максимально возможная загрузка процессорных ресурсов.**

Процессоры, выполненные по технологии Hyper-Threading, одновременно обрабатывают **две (несколько) нитей** процессов, состоящие из потоков данных и команд двух (нескольких) разных приложений или различных частей одного.

# Система с двумя **IA-32** процессорами и ЦП, построенный по технологии **Hyper-Threading**



# Загрузка процессоров



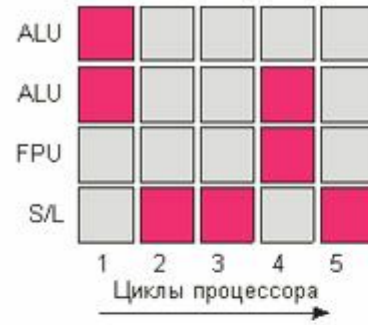
Оранжевые и зеленые блоки работают, серые простаивают.

- 1 — выполнение **1 нити** на обычном процессоре;
- 2 — выполнение **2 нитей** на **2** разных процессорах стандартной **2-процессорной** системой;
- 3 — одновременное выполнение **2 нитей** на **1** процессоре с технологией **Hyper-Threading**;
- 4 — выполнение **4 нитей** на **2** процессорах **2-процессорной** системы с технологией **Hyper-Threading**.

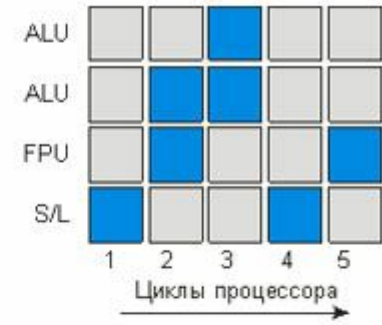
Исполнительные блоки



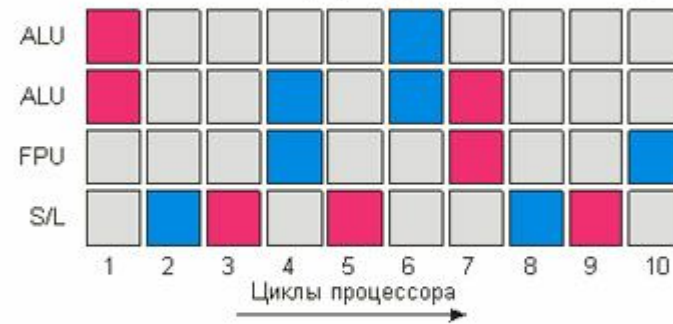
Поток А



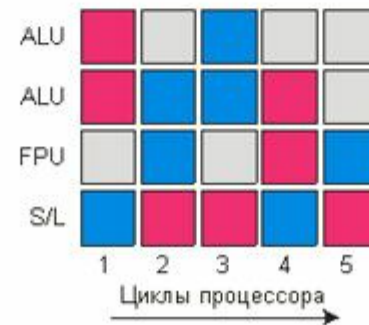
Поток В



Выполнение на процессоре без Hyper-Threading



Выполнение на процессоре с Hyper-Threading

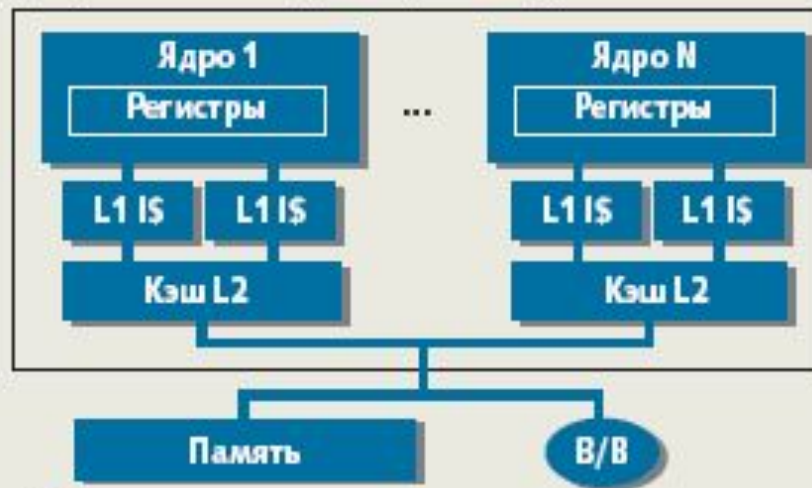


# Многоядерность

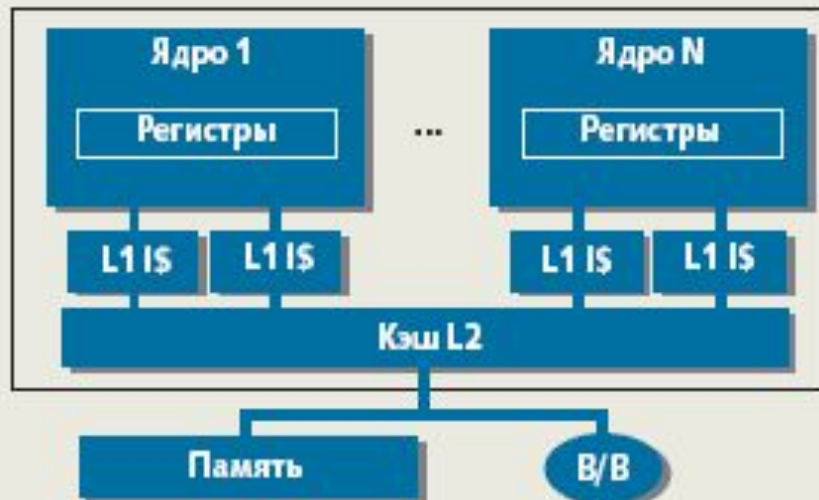
а) Традиционный процессор



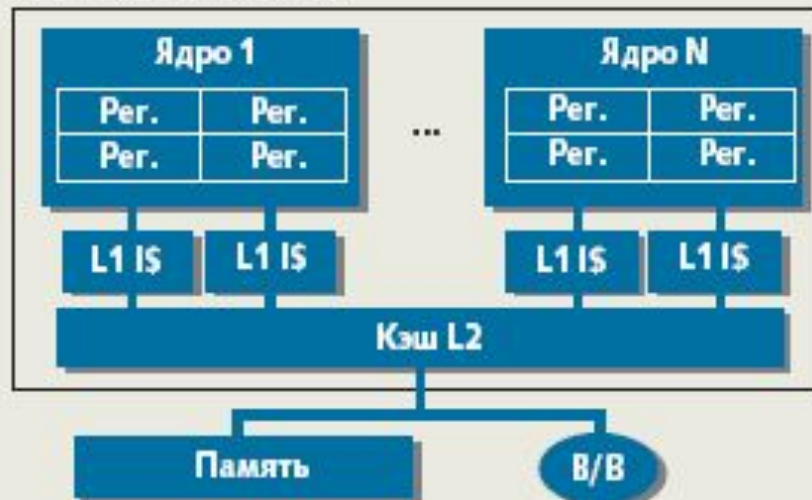
б) Простая многоядерная архитектура



в) Многоядерная архитектура с общей кэш-памятью



г) Многопоточная многоядерная архитектура с общей кэш-памятью



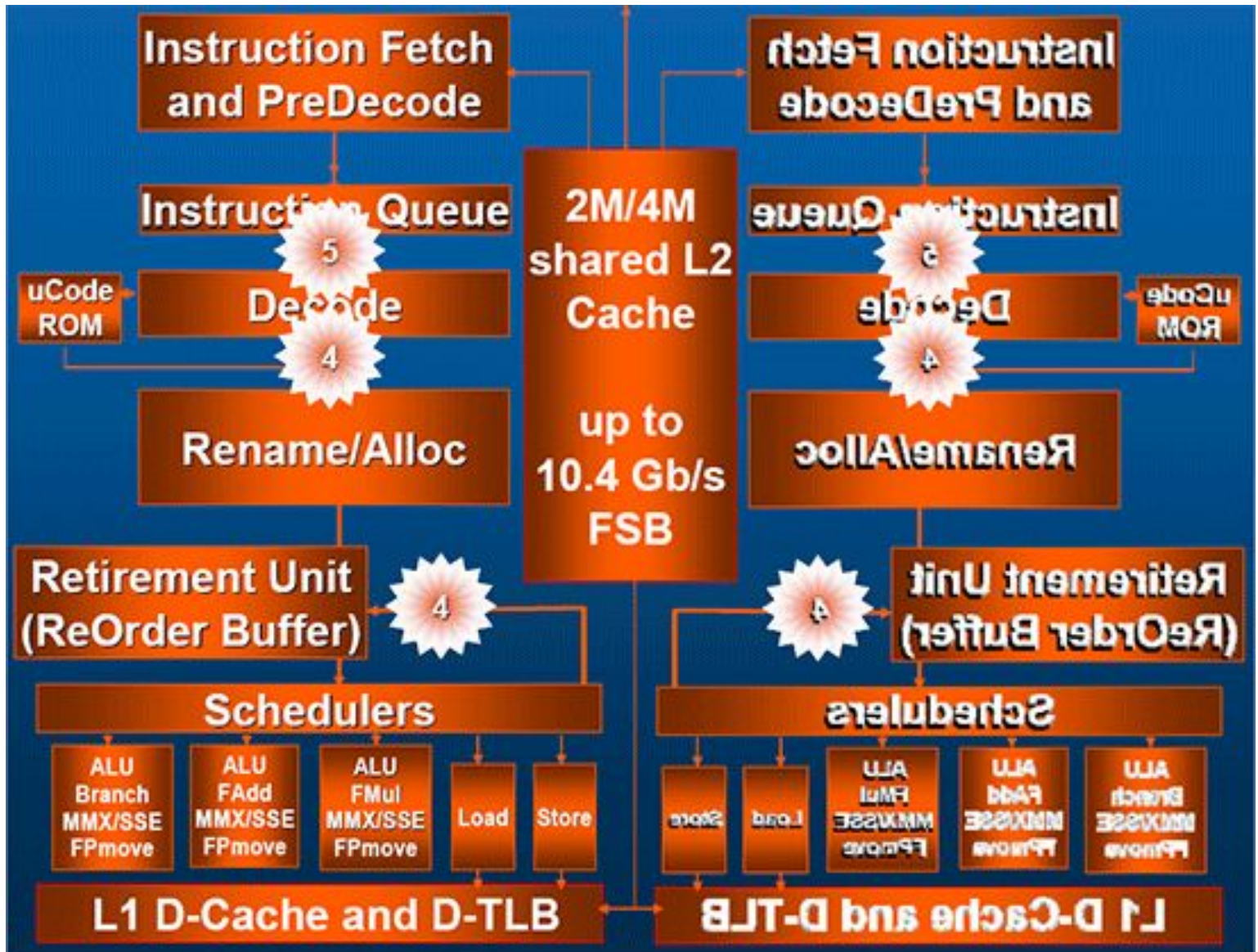
# Многоядерность

## Пути увеличения быстродействия:

- наращивание тактовых частот,
- увеличение числа инструкций, исполняемых за один такт,
- уменьшение числа операций, необходимых для обработки одних и тех же объёмов данных (SIMD инструкции).

$$\mathbf{Performance} = \mathbf{Frequency} * \mathbf{IPC}$$

$$\mathbf{Power} = \mathbf{Frequency} * \mathbf{U}^2 * \mathbf{C}_{dyn}$$





# Особенности **Core 2 Duo**

- **Intel Wide Dynamic Execution** (14 стадий конвейера, до 4х инструкций за такт в каждом ядре)
- **Intel Smart Memory Access** (Оптимизация доступа к памяти, в т.ч. Memory Disambiguation)
- **Intel Advanced Smart Cache** (Общий КЭШ 2го уровня, динамически распределяемый между ядрами)
- **Intel Advanced Digital Media Boost** (128-битный SSE, расширенный набор команд)
- **Intel Intelligent Power Capability**
- **Micro-ops fusion** и **macrofusion**

# Особенности **Core 2 Duo**

- **Intel Wide Dynamic Execution** — технология выполнения большего количества команд за каждый такт, повышающая эффективность выполнения приложений и сокращающая энергопотребление. Каждое ядро — технология выполнения большего количества команд за каждый такт, повышающая эффективность выполнения приложений и сокращающая энергопотребление. Каждое ядро может выполнять до 4-х инструкций одновременно с помощью 14-стадийного конвейера.
- **Intel Intelligent Power Capability** — технология, с помощью которой для исполнения задач активируется работа отдельных узлов чипа по мере необходимости, что значительно снижает энергопотребление системы в целом.
- **Intel Advanced Smart Cache** — технология использования общей для всех ядер кэш-памяти 2-го уровня, что снижает энергопотребление и повышает производительность, при этом, по мере необходимости, одно из ядер может использовать весь объём кэш-памяти при динамическом отключении другого ядра.
- **Intel Smart Memory Access** — технология оптимизации работы подсистемы памяти, сокращающая время отклика и повышающая пропускную способность подсистемы памяти.
- **Intel Advanced Digital Media Boost** — технология обработки 128-разрядных команд SSE — технология обработки 128-разрядных

# Intel® Wide Dynamic Execution

## EACH CORE

*EFFICIENT  
14 STAGE  
PIPELINE*

*DEEPER  
BUFFERS*

*4 WIDE -  
DECODE TO  
EXECUTE*

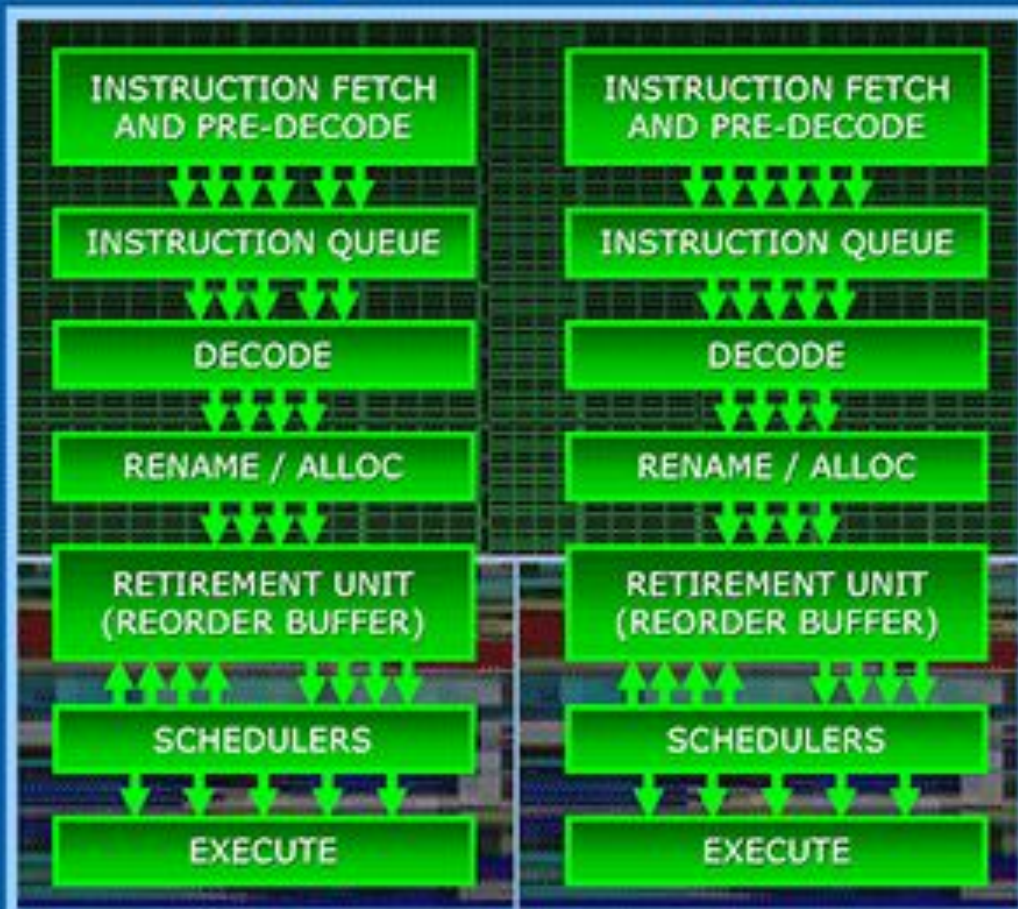
*4 WIDE -  
MICRO-OP  
EXECUTE*

*MICRO  
and  
MACRO  
FUSION*

*ENHANCED  
ALUs*

## CORE 1

## CORE 2



Perf ↑

Energy ↓

## ADVANTAGE

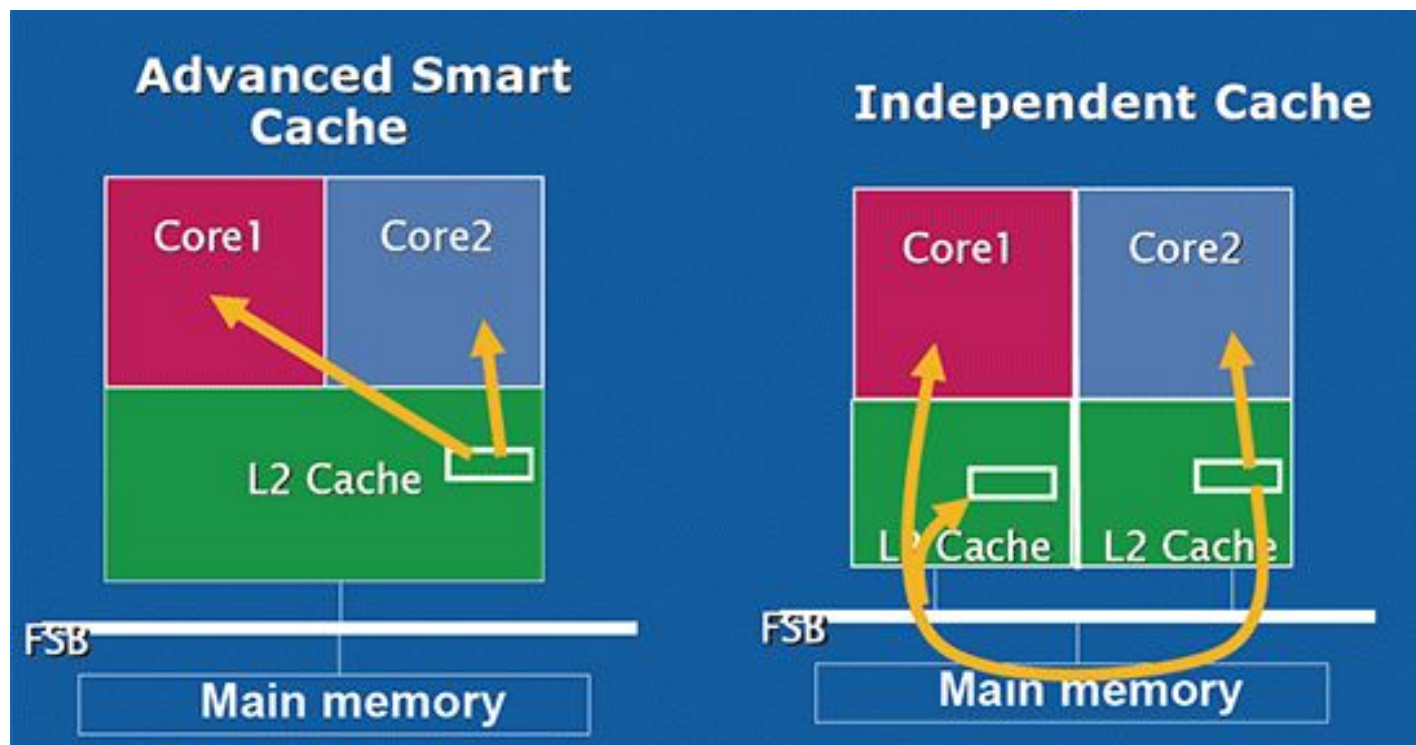
- 33% Wider Execution over Previous Gen
- Comprehensive Advancements
- Enabled In Each Core

# Intel Wide Dynamic Execution

- Каждое ядро выбирает из кода и исполняет до **4 x86 инструкций** одновременно.
- Имеет **4 декодера** (**1** для сложных инструкций и **3** – для простых).
- **6** портов запуска (**1** – Load, **2** – Store и **3** универсальных).
- Усовершенствованный блок предсказания переходов.
- Увеличены буферы команд, используемые на различных этапах анализа кода для оптимизации скорости исполнения,
- **Длина конвейера составляет 14 стадий.**  
Процессоры с микроархитектурой Core обладают поддержкой 64-битных расширений Enhanced Memory 64 Technology (EM64T).

# Intel Advanced Smart Cache

- Нет необходимости поддерживать когерентность.
- Динамически распределяется между ядрами.



# Intel Smart Memory Access

- **6** Блоков предвыборки (**2** для КЭШа 2го уровня, по **2** для КЭШей 1го уровня).
- **Memory Disambiguation** технология направлена на повышение эффективности работы алгоритмов внеочередного исполнения инструкций, осуществляющих чтение и запись данных в памяти. Она использует алгоритмы, позволяющие с высокой вероятностью устанавливать зависимость последовательных команд сохранения и загрузки данных, и даёт возможность, таким образом, применять внеочередное выполнение инструкций к этим командам.

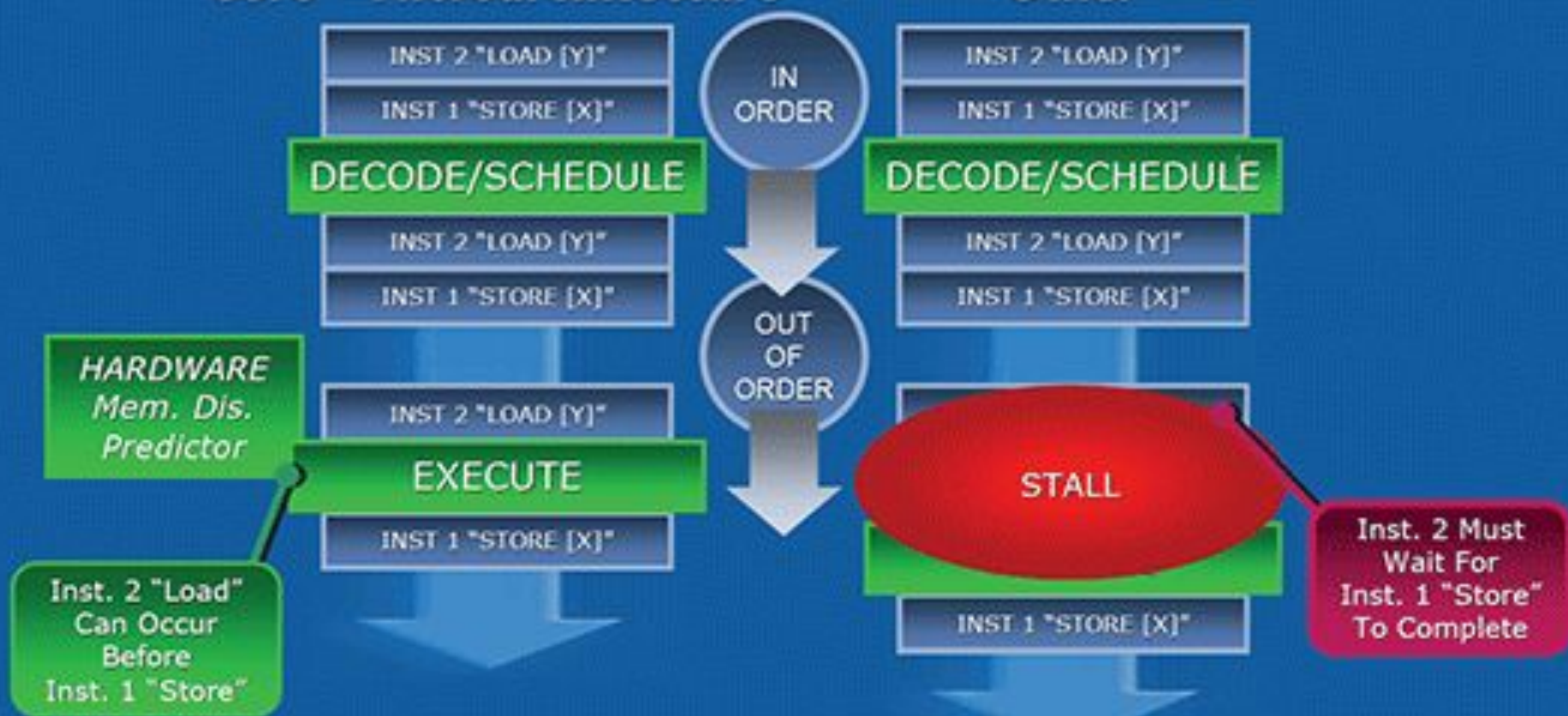
# Intel® Smart Memory Access

## Hardware-based Memory Disambiguation

Dmitry Bobrov - ALU

### Core™ Microarchitecture

### Other



Perf ↑

Energy ↓

### ADVANTAGE

- Higher Utilization of Pipeline
- Masks latency to data access
- Higher Performance

# **Micro-ops fusion** И **macrofusion** ТЕХНОЛОГИИ

Обе технологии увеличивают числа исполняемых команд за такт.

**1. Команда – это «связанные» декодером зависимые микро-инструкции, на которые распадается x86-команда.** Это позволяет избежать ненужных простоев процессора, если связанные микроинструкции оказываются оторванными друг от друга в результате работы алгоритмов внеочередного выполнения.

**2. Команда -- связанные между собой последовательных x86-команд, например, сравнение со следующим за ним условным переходом, представляются внутри процессора одной микроинструкцией.** Таким путём достигается как увеличение темпа исполнения кода, так и некоторая экономия энергии.



# Macro-fusion ТЕХНОЛОГИИ

## With Intel's New Macro-Fusion

Read five Instructions from Instruction Queue

Send fusable pair to single decoder

Single uop represents two instructions

Instruction Queue

inc ecx

store [mem3], ebx

jne targ

cmp eax, [mem2]

load eax, [mem1]

dec0

dec1

dec2

dec3

Cycle 1

inc ecx

store [mem3], ebx

cmpjne eax, [mem2], targ

load eax, [mem1]

Intel Developer  
FORUM



# Intel Advanced Digital Media Boost

Современное ПО позволяет работать со 128-битовыми операндами различного характера (векторами и целочисленными либо вещественными данными повышенной точности).

Этот факт заставил инженеров Intel задуматься об ускорении работы SSE блоков процессора, тем более что до настоящего времени процессоры Intel исполняли **одну SSE-инструкцию**, работающую с 128-битными операндами, **лишь за два такта**.

**Один такт** тратился на обработку **старших 64 бит**.

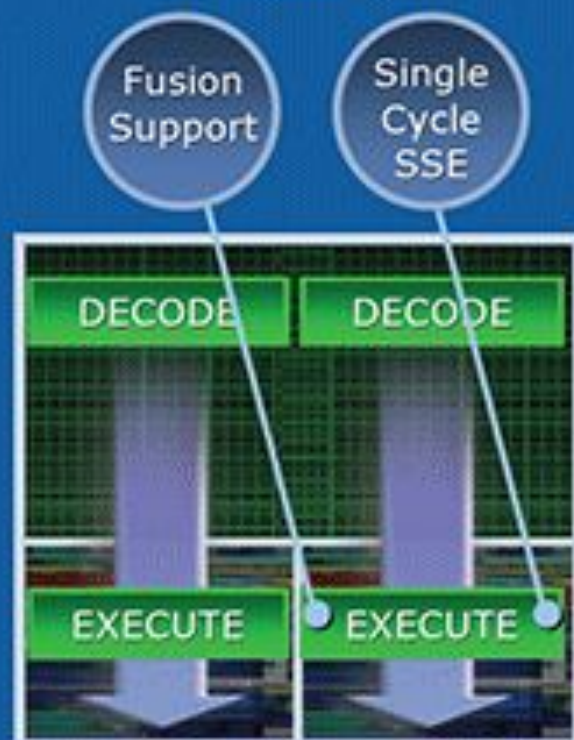
**Второй такт** – на обработку **младших 64 бит**.

**Микроархитектура Core** позволяет ускорить работу с SSE инструкциями в **два раза**.

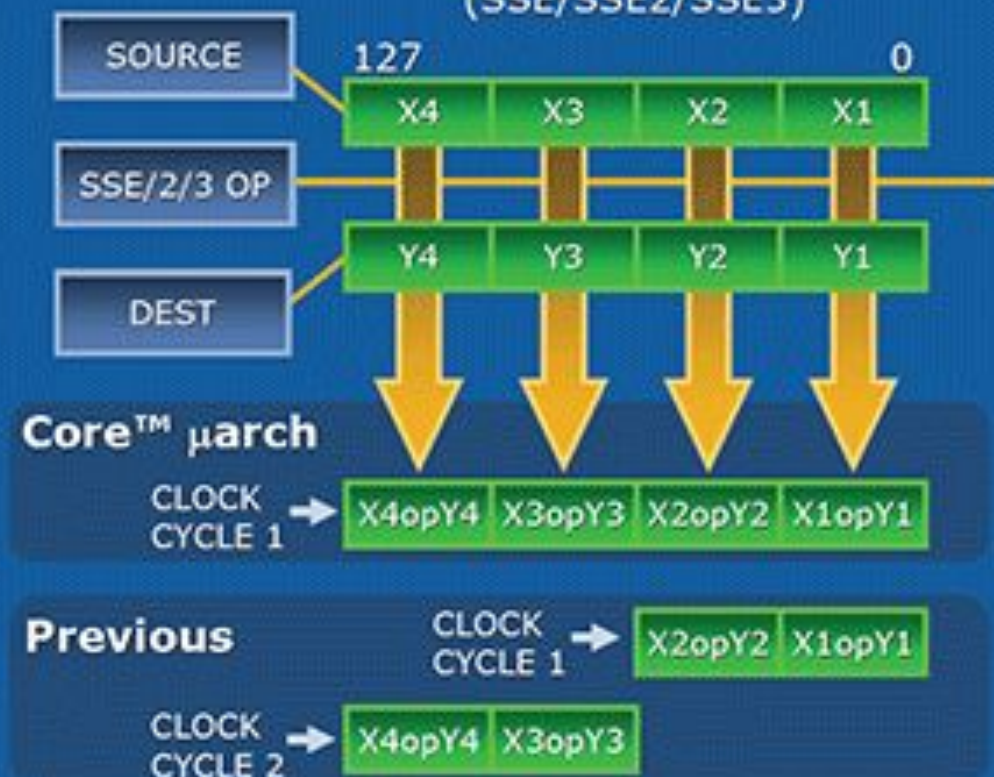
# Intel® Advanced Digital Media Boost

## Single Cycle SSE

### In Each Core



### SSE Operation (SSE/SSE2/SSE3)



Perf ↑

Energy ↓

### ADVANTAGE

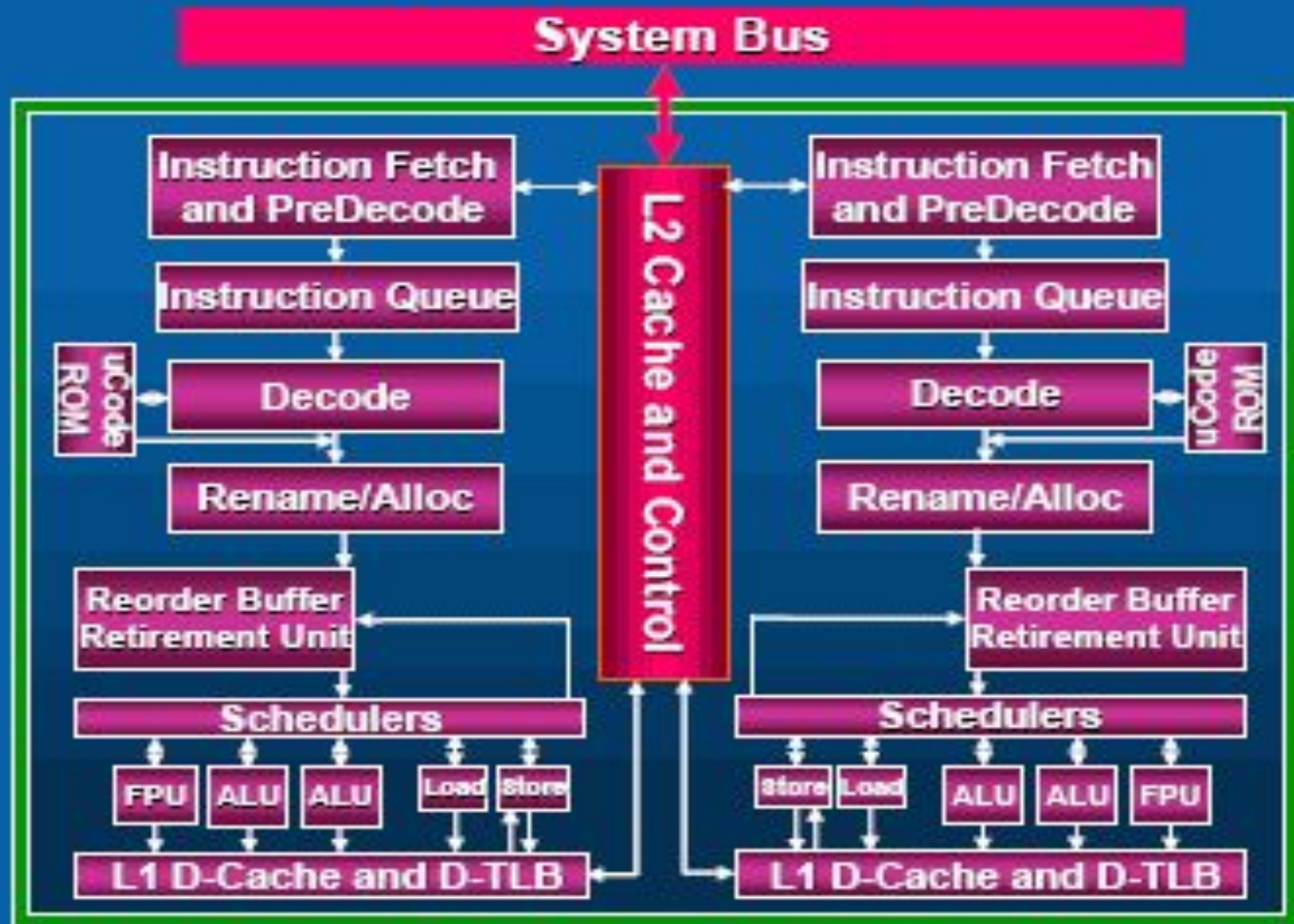
- Increased Performance
- 128 bit Single Cycle in each core
- Improved Energy Efficiency

# Технические характеристики **Core** **2 Duo**

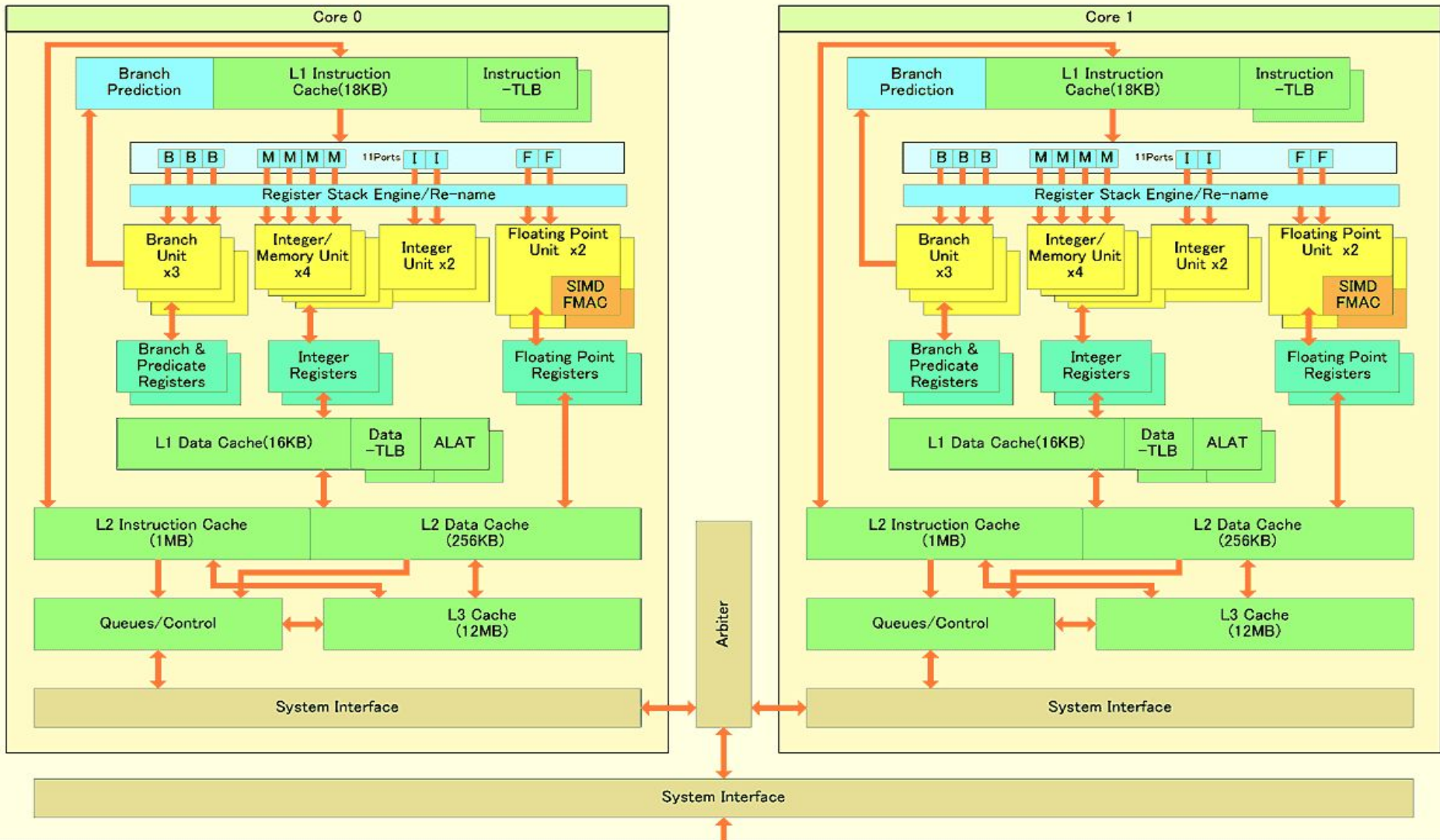
- **L1 DCache 32K 8-way**
- **L1 ICache 32K 8-way**
- **L2 Cache 4M / 2 Cores**
- **ITLB 128 ent**
- **DTLB 256 ent**
- **Устройства**
  - **5 Integer 3 ALU + 2 AGU**
  - **2 Load/Store (1 Load + 1 Store)**
  - **4 FP (FADD + FMUL + FLOAD + FSTORE)**
  - **3 SSE (128 bit)**

	Intel Core	AMD K8
<b>L1</b> кэш данных	<b>32</b> Кбайта	<b>64</b> Кбайта
<b>L1</b> кэш инструкций	<b>32</b> Кбайта	<b>64</b> Кбайта
Латентность кэша <b>L1</b>	<b>3</b> цикла	<b>3</b> цикла
Ассоциативность <b>L1</b> кэша	<b>8-way</b>	<b>2-way</b>
Размер <b>L1 TLB</b>	Инструкции – <b>128</b> вхождений	Инструкции – <b>32</b> вхождения
	Данные – <b>256</b> вхождений	Данные – <b>32</b> вхождения
Размер <b>L2</b> кэша	<b>4</b> Мбайта на два ядра	<b>1</b> Мбайт на каждое ядро
Латентность кэша <b>L2</b>	<b>14</b> циклов	<b>12</b> циклов
Ассоциативность <b>L2</b> кэша	<b>16-way</b>	<b>16-way</b>
Ширина шины <b>L2</b> кэша	<b>256</b> бит	<b>128</b> бит
Размер <b>L2 TLB</b>	-	<b>512</b> вхождений
Длина конвейера	<b>14</b> стадий	<b>12</b> стадий
Число <b>x86</b> декодеров	<b>1</b> сложный и <b>3</b> простых	<b>3</b> сложных
Целочисленные исполнительные устройства	<b>3 ALU + 2 AGU</b>	<b>3 ALU + 3AGU</b>
<b>Load/Store</b> устройства	<b>2 (1 Load + 1 Store)</b>	<b>2</b>
<b>FP</b> ИУ	<b>FADD + FMUL + FLOAD + FSTORE</b>	<b>FADD + FMUL + FSTORE</b>

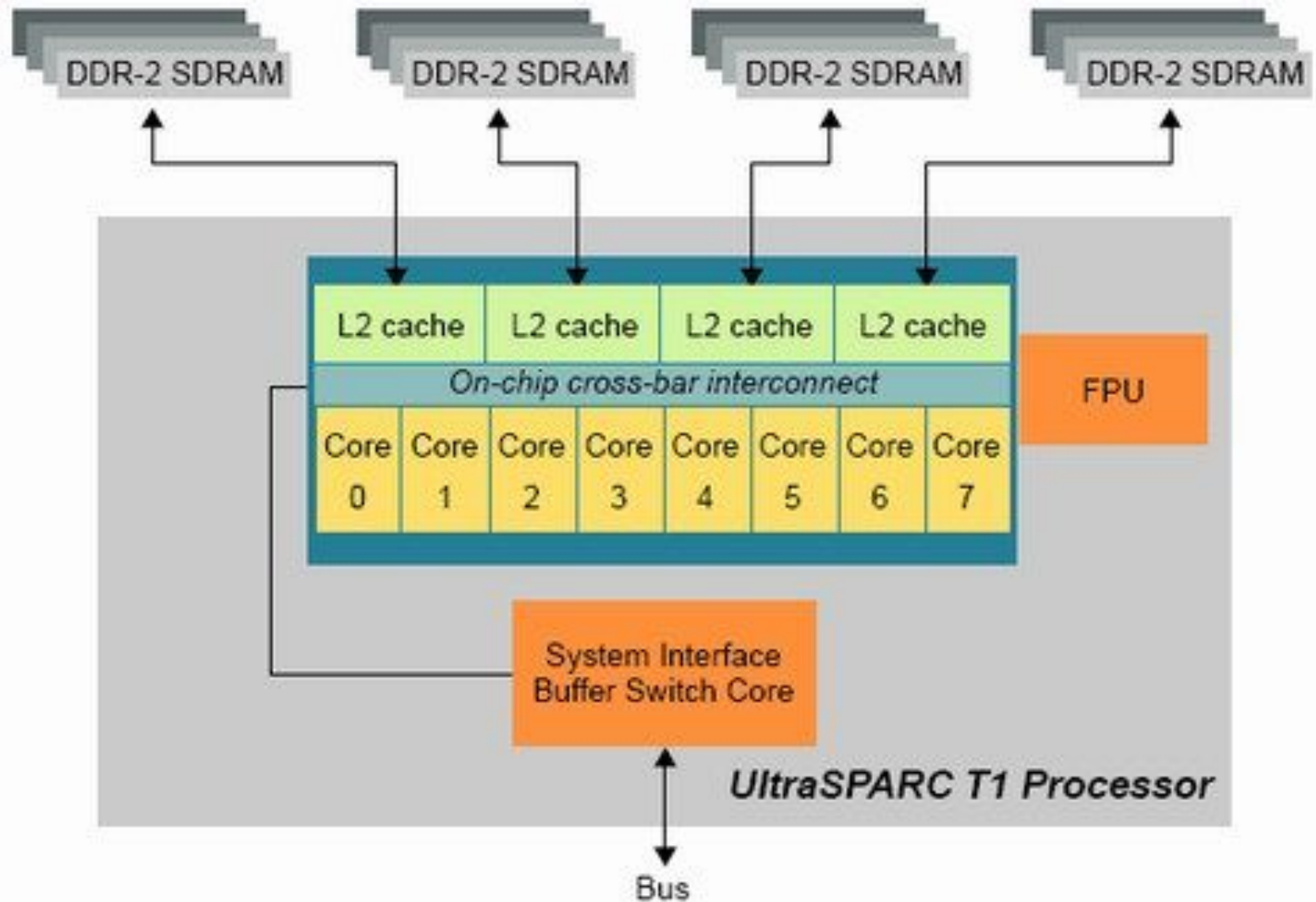
# Merom, Conroe and Woodcrest Block Diagram



# Itanium 2 (Montecito)



# Niagara

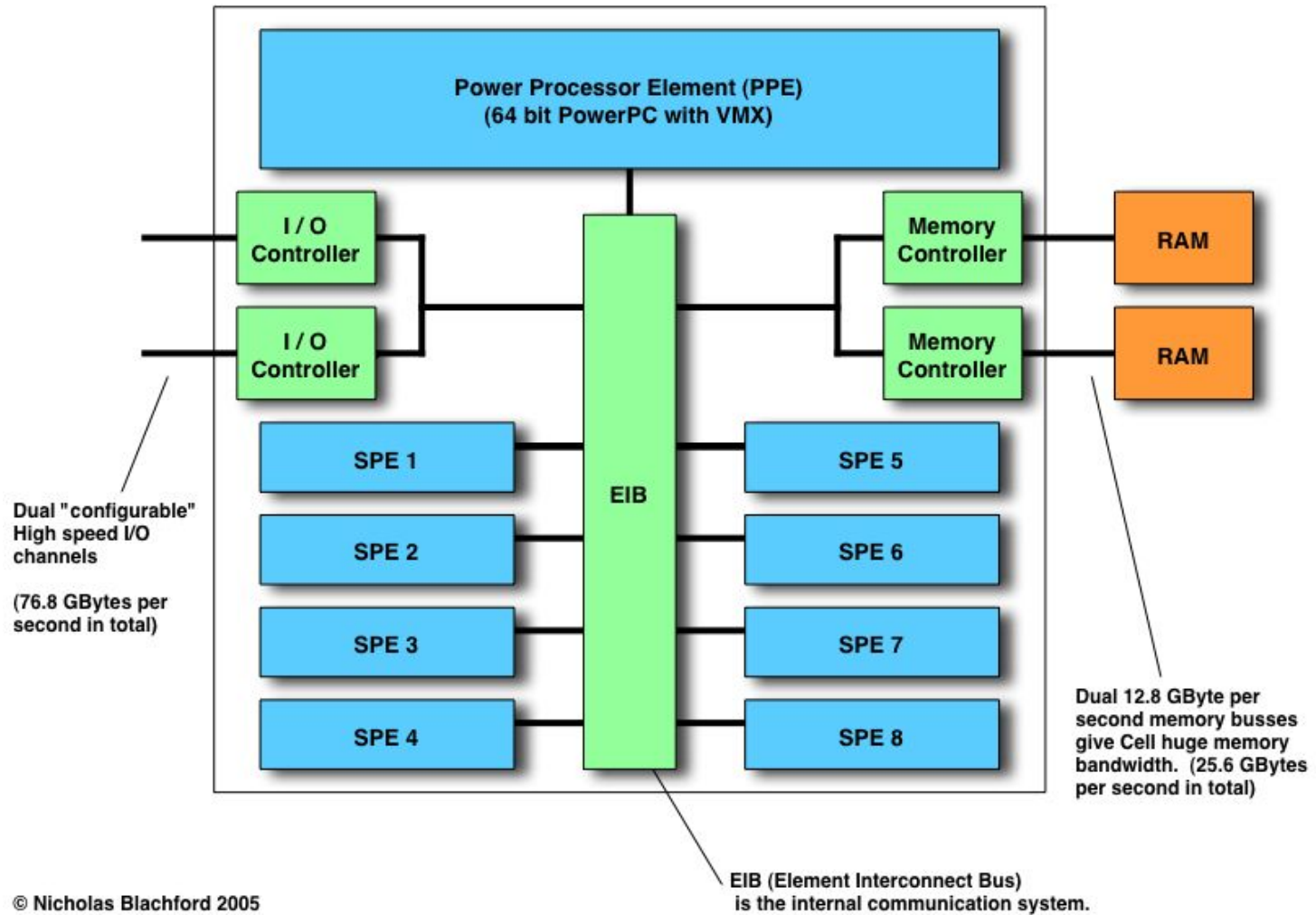




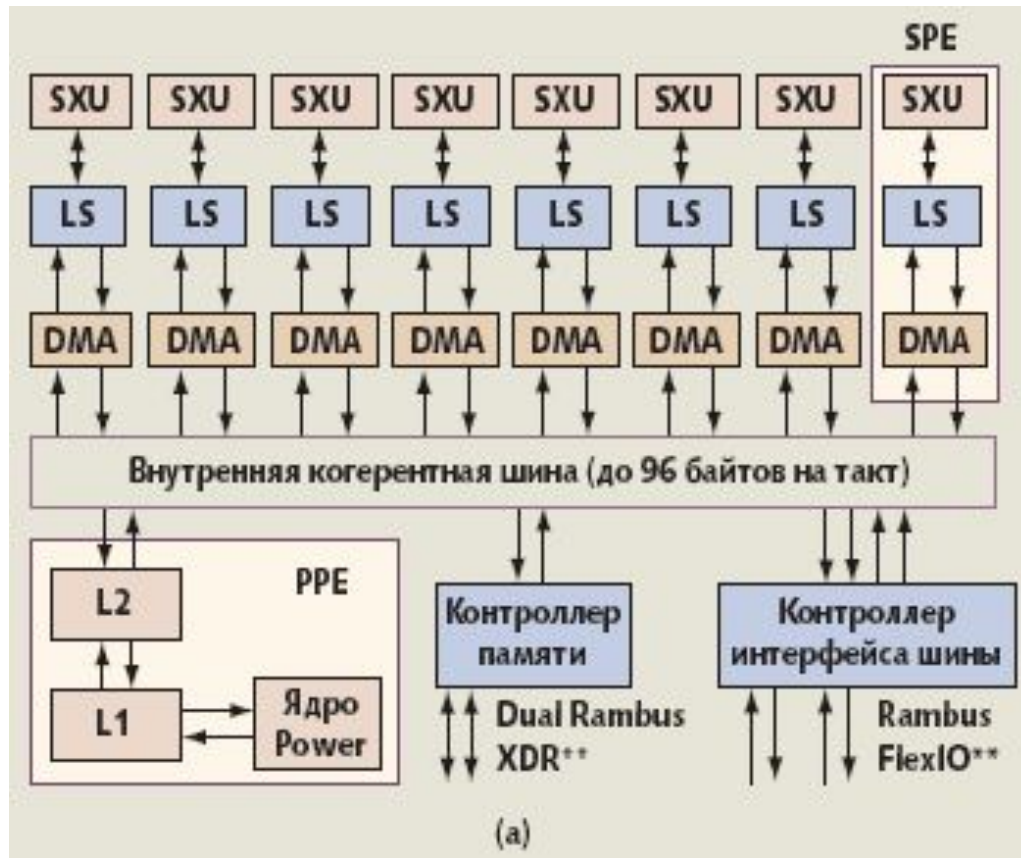
# Особенности **Niagara**

- 8 ядер
- 4 потока на ядро
- Общий FPU
- 79 Ватт при 1.2 ГГц
- 26.5 ГБ/сек

# Cell



# Архитектура Cell

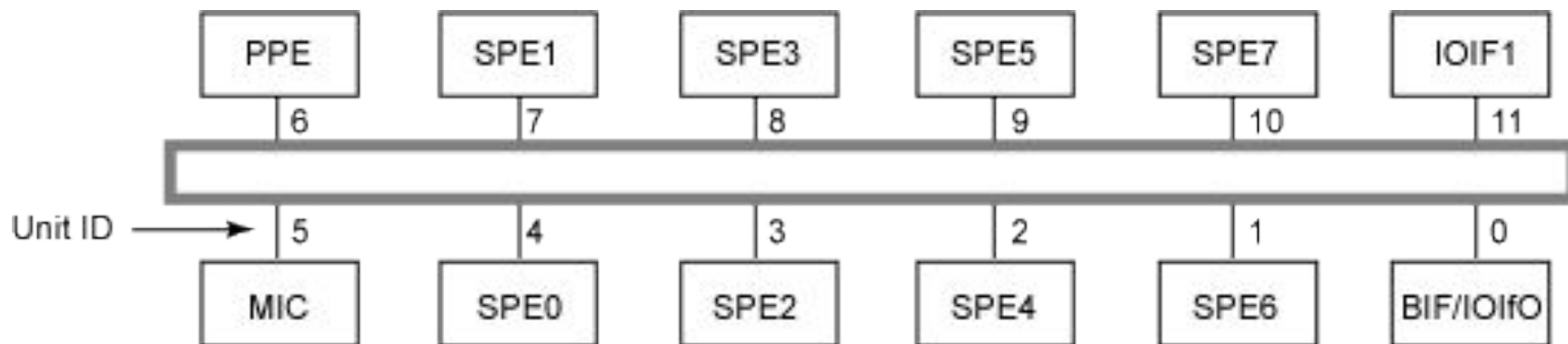


# Cell

- Главный процессорный элемент
  - Упорядоченное исполнение
  - Поддержка работы с двумя потоками
- **8** синергетических процессорных элементов
  - Ядро на основе 286 архитектуры
  - Поддержка векторных вычислений 128 бит
  - Отсутствие КЭШей
  - Локальная память 256 Кбайт с прямым доступом
- Шина ввода вывода
  - Пропускная способность 76,8 Гбайт/с

# Шина взаимосвязанных элементов

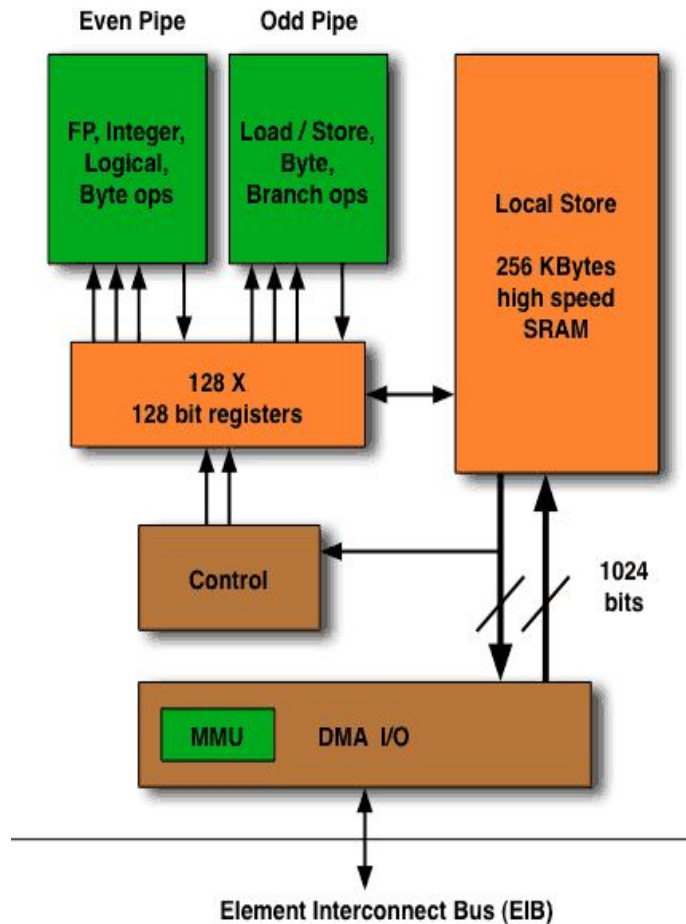
- Передает 96 байт/цикл
- Более 100 уникальных запросов



# Power Processor Element

- Два **64-битных** ядра на основе архитектуры **POWER**
- Упорядоченное исполнение команд
- Поддержка **SMT** (многопоточность)
- КЭШ
  - 1го уровня: 32+32 Кбайт
  - 2го уровня: 512 Кбайт

# Synergistic Processor Element



- **4** целочисленных векторных устройства
- **4** векторных устройства с плавающей запятой
- **128** регистров по **128** бит
- **256** Кбайт локальной памяти
- Динамическая защита доступа к памяти

# Производительность **Cell** (для **4GHz**)

- **256 GFLOPS** с плавающей запятой
- **256 GOPS** целочисленная арифметика
- **25 GFLOPS** с плавающей запятой  
двойной точности