
Developing the Nanometer

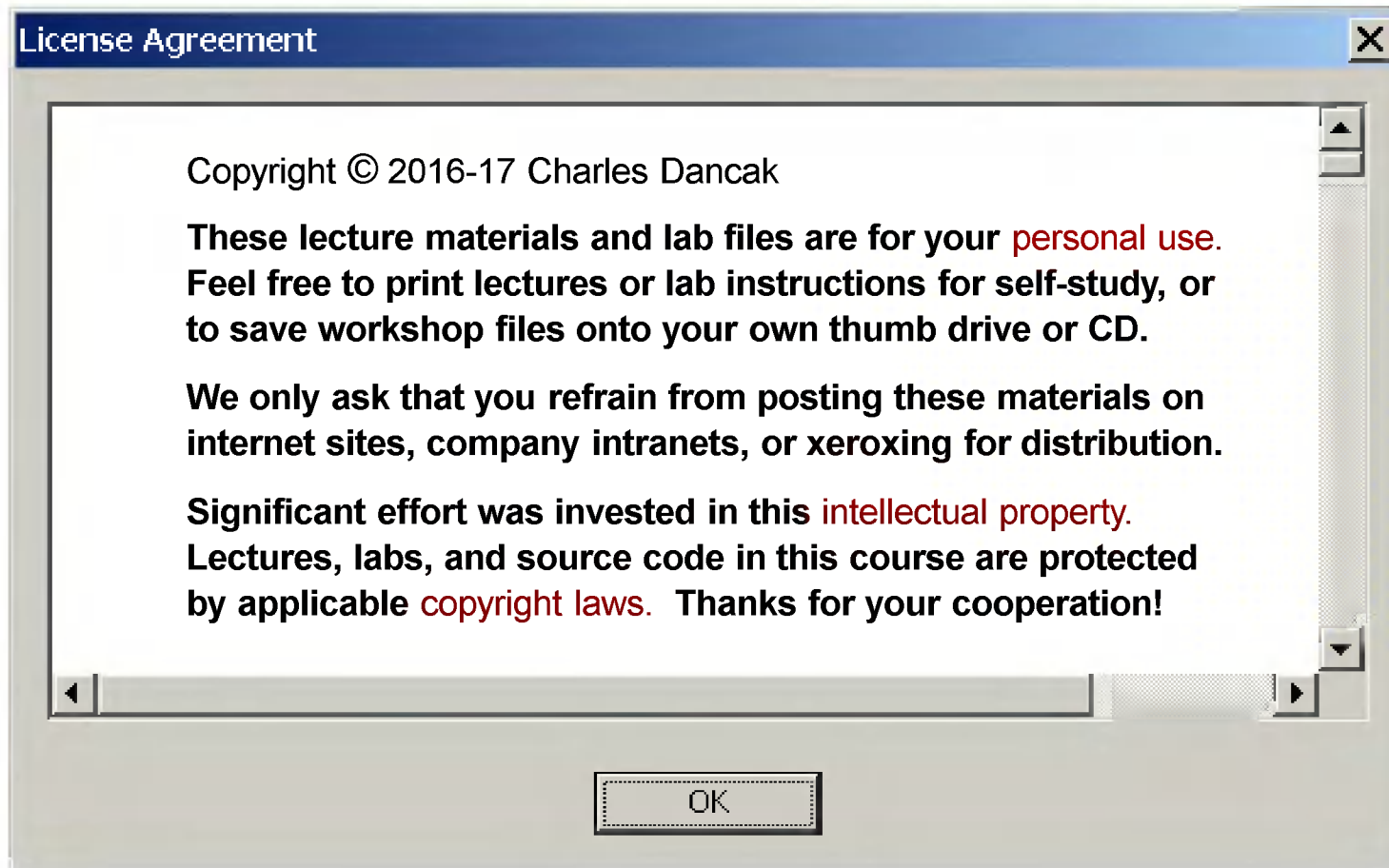
ASIC

From Specs to Silicon



Проектирование с использованием
блоков интеллектуальной
собственности (IP)





- Материалы предназначены только для **зарегистрированных студентов**.
- Не следует размещать их на корпоративных или любых других сайтах

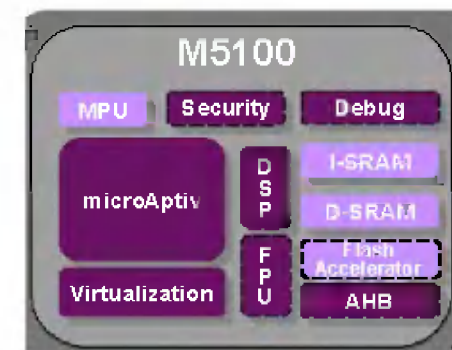
Благодарности

2-3

- Подтема 2.4 презентации подготовлена в сотрудничестве с Юрием Панчулом, старшим инженером по разработке аппаратуры компании Imagination Technologies, группа разработки микропроцессора MIPS I6400
- Перевод на русский язык выполнен Александром Барабановым, доцентом кафедры компьютерной инженерии факультета радиофизики, электроники и компьютерных систем Киевского национального университета имени Тараса Шевченко

Обзор IP-ядер

- Временная шкала интеграции IP
- Первые конвейерные архитектуры MIPS
- Современная эволюция ядер MIPS
- Компромисс площадь/время



Временная шкала интеграции 2-5 IP

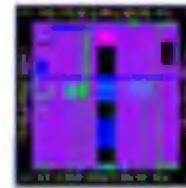
IP

Внешние IP-блоки



Окончание разработки RTL кода

ARM



Предупреждение:
График поставщиков IP-блоков должен быть согласован с контрольными точками проекта — такими, как окончание разработки RTL кода.

1. Спецификация

2. Проектирование

3. Верификация

4. Имплементация

5. Производство

6. Отбраковка

Цифровые IP-блоки:

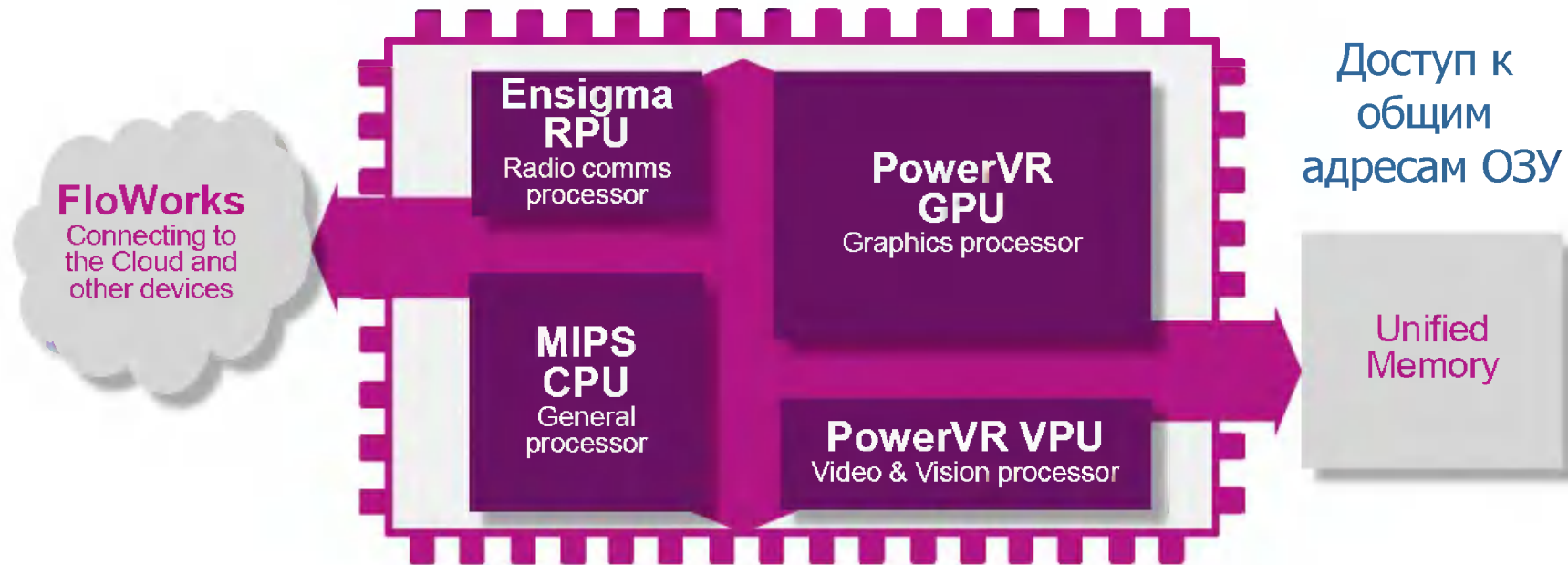
- Полученные извне
- Спроектированные внутри
- Модифицированные пользователем

Топологические макроячейки

- При проектировании на основе блоков интеллектуальной собственности (IP), поставщики системы на кристалле (SOC) повторно используют коммерческие IP-блоки.
- Или существующие IP-блоки из предыдущих проектов.
- Или новые функциональные блоки, которые созданы внутренними командами разработчиков.
- Поставщики IP-блоков, такие как ARM, тесно сотрудничают со своими основными клиентами.

Ядра Imagination MIPS

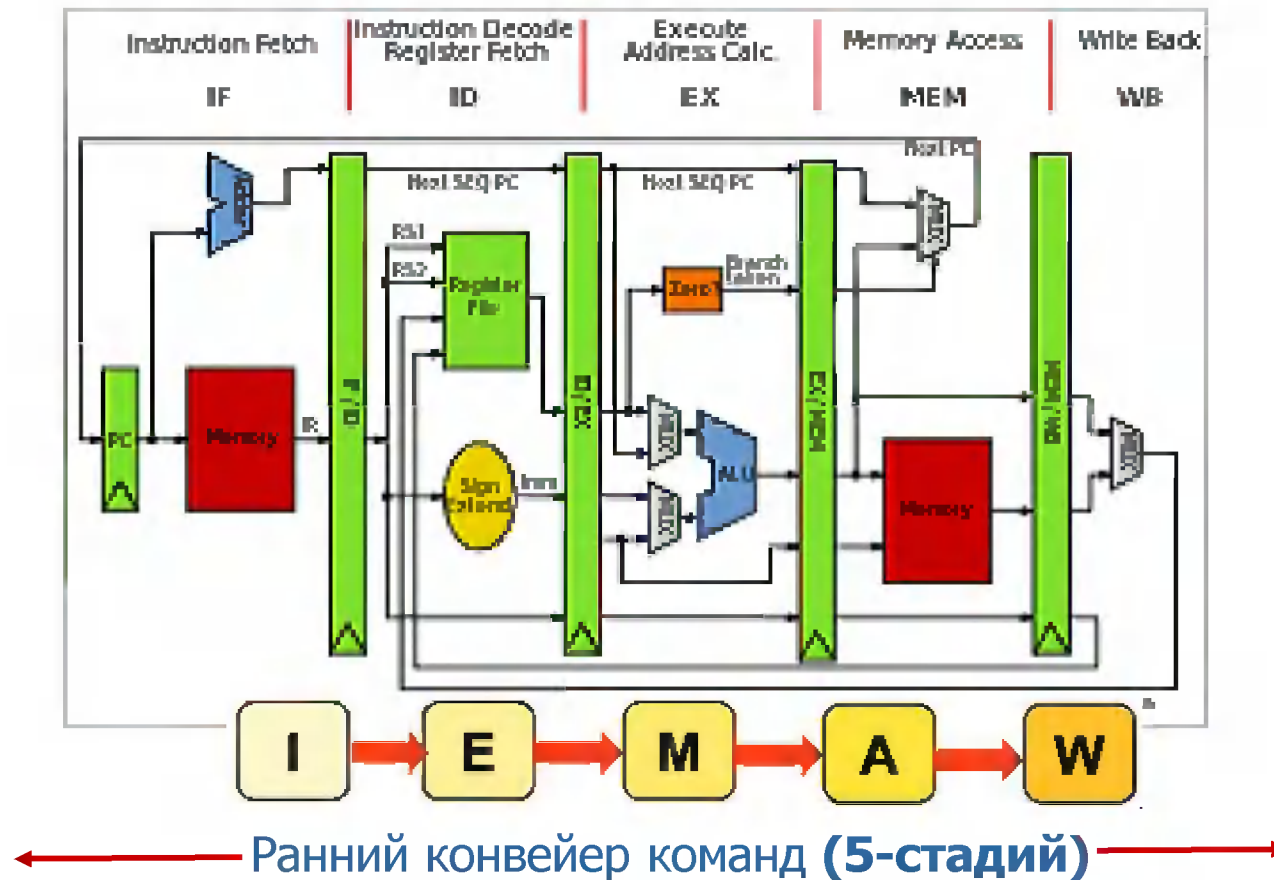
2-6



- Чем **различаются** различные проекты SOC, которые повторно используют IP-блоки?
- Команда разработчиков SOC должна сформировать правильное предложение продукта.
- Реализовать SOC с правильным **сочетанием**: усилий, цены, потребляемой мощности.
- Уделить внимание обеспечению эксплуатационных характеристик и поддержке заказчиков.

Первые конвейерные архитектуры MIPS 2-7

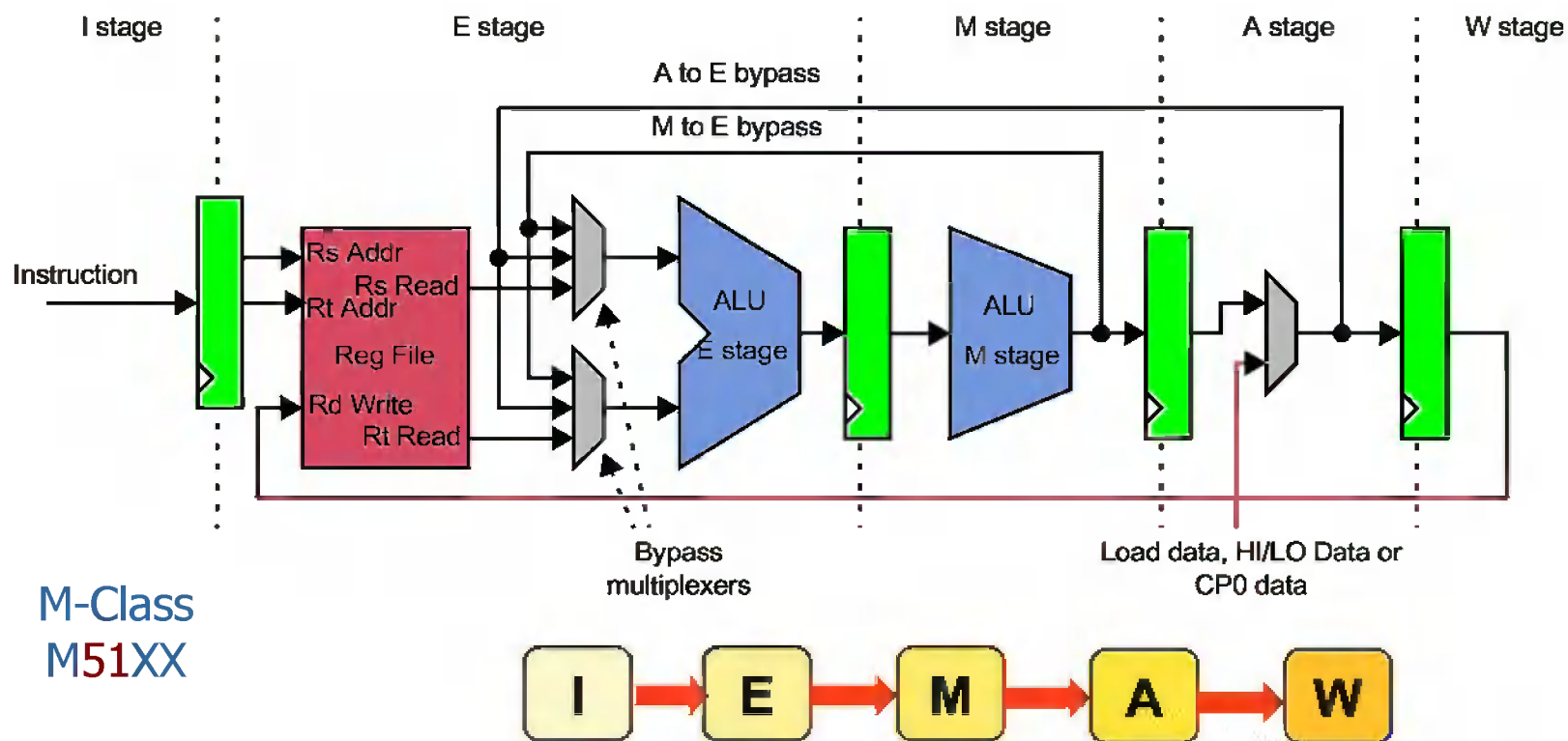
Устаревшая архитектура R2000



- Тонкая архитектура, т.е. **загрузка-сохранение** (регистр-регистр).
- Арифметические и логические операции выполняются между регистрами процессора.
- **Конвейер** постоянно развивался, становился глубже, было добавлено множество возможностей.

Пример младшего ядра M51XX

2-8

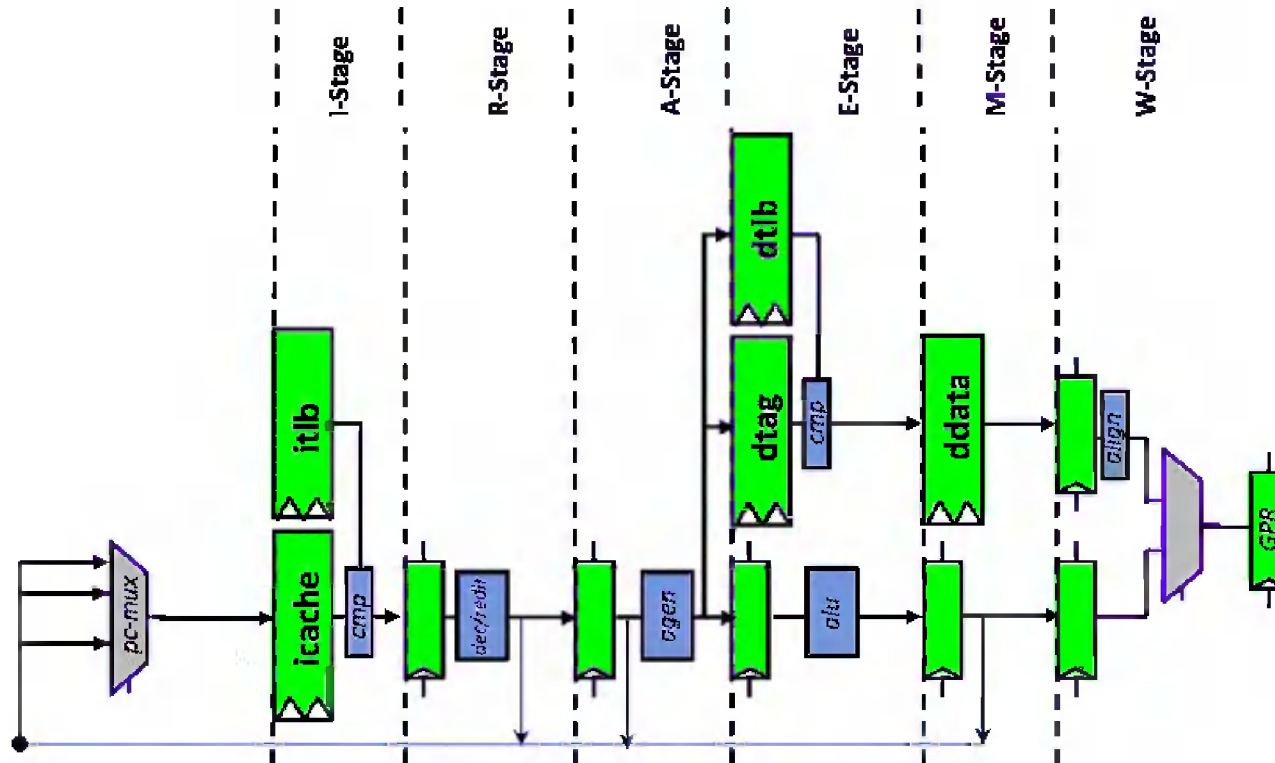


- Единственный конвейер с последовательным выполнением команд—глубина та же, но **задержки сбалансированы**.
- Варианты: Алгоритмы цифровой обработки с фиксированной точкой; векторная обработки и т.д.
- Ядро характеризуется минимальной площадью и энергопотреблением.

Пример младшего ядра M62XX

2-9

M-Class
M62XX

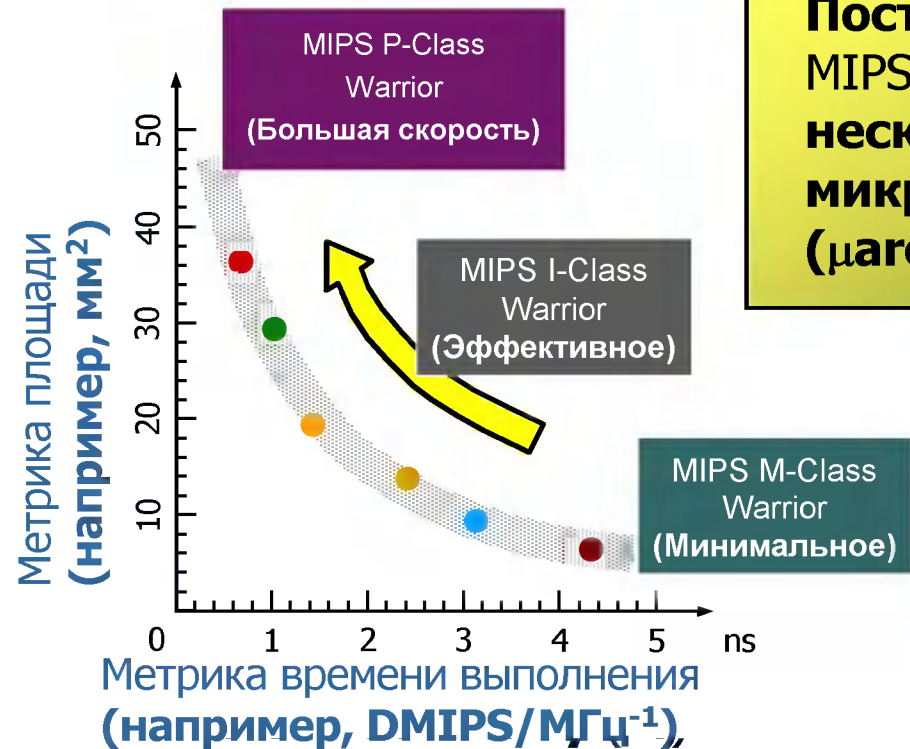


- Конвейер более **глубокий**, чем в M51XX—6 стадий, вместо 4 или 5.
- Компромисс: f_{max} больше в **1.3 раз**, но отношение DMIPS/МГц немного меньше.
- В целом, M62XX обеспечивает **более высокую производительность** чем M51XX.

Компромисс площадь/время 2-10

I-Class:

- Выполняется Linux
- Хабы для интернета вещей (IoT)
- Модемы LTE
- Помощник водителя автомобиля



← Ключевой принцип:
Поставщики IP—ARM, MIPS—предлагают несколько линий микроархитектуры (μarch).

M-Class:

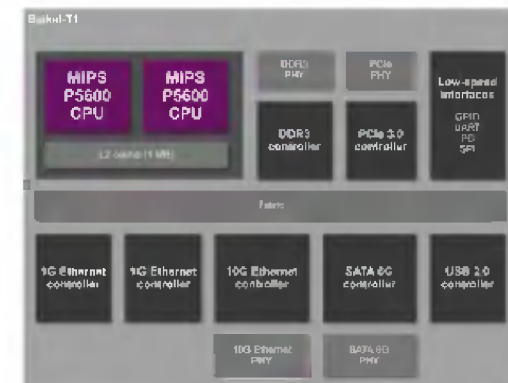
- Простые микроконтроллеры
- Встроенные системы
- Датчики (для интернета вещей)

- Младшие: оптимизированы - минимальная площадь, минимальная потребляемая мощность.
- Среднего уровня: достаточная производительность; меньшая потребляемая мощность. Эффективные!
- Высокого уровня: терминалы, планшеты, рынок минисерверов. Быстрые.

IP-блоки средне-высокого уровня:

- Примеры среднего уровня
- Аппаратная многопоточность
- Максимальная производительность
- Многоядерные кластеры
- Когерентность кэш-памяти
- Конфигурируемые μ Ps

SOC Байкал T1



Среднего уровня: interAptiv 2-12



Модемы

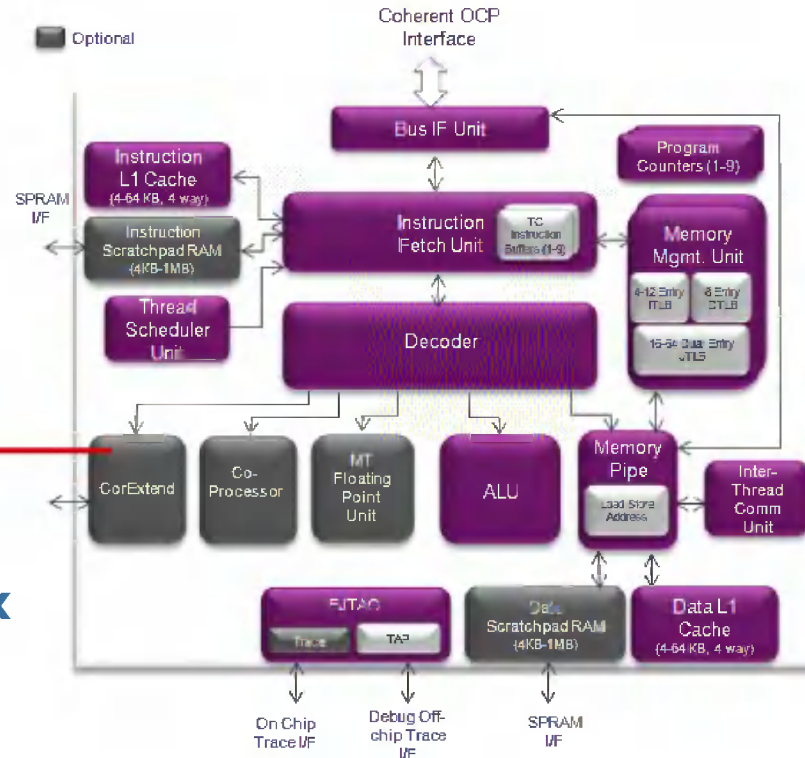


Камеры



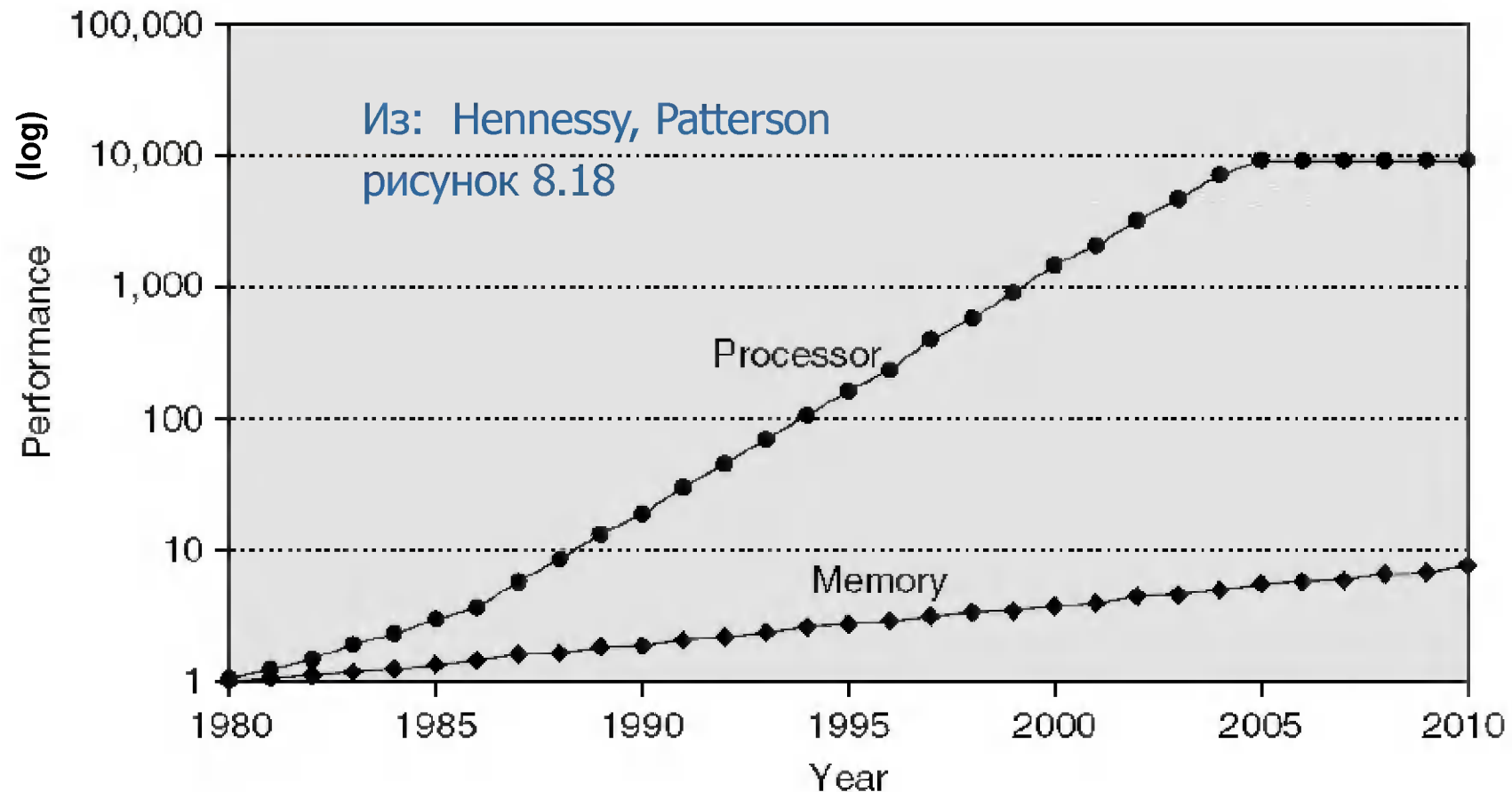
Устройства хранения данных

Технология конфигурируемых процессоров



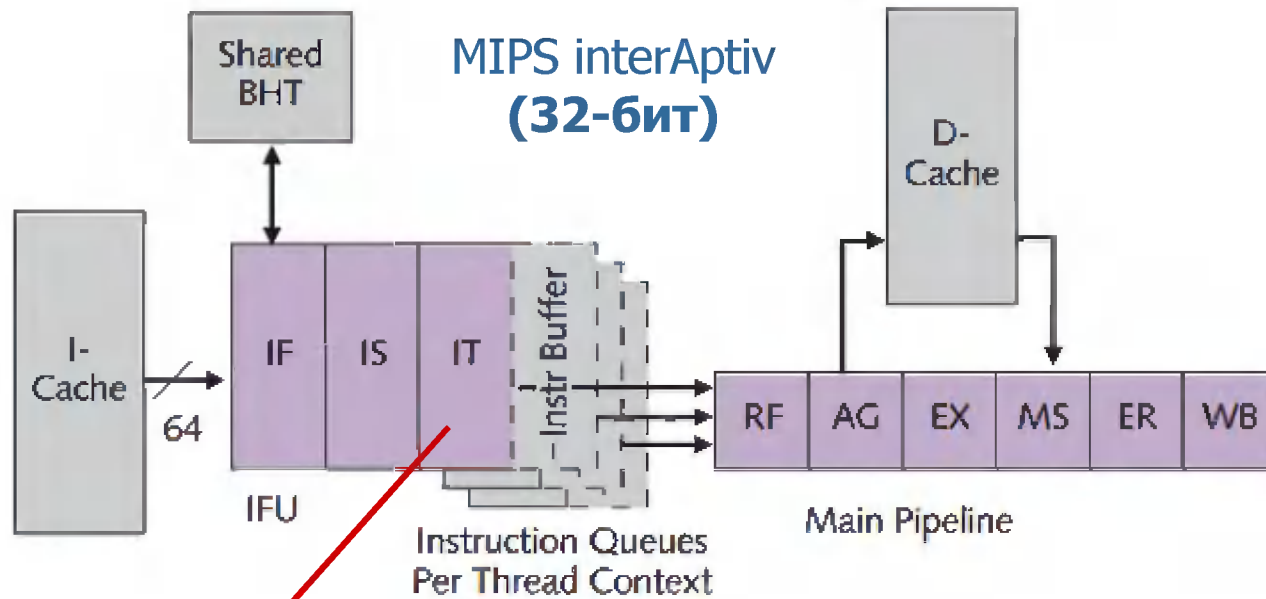
- Оптимальный выбор для многих микросхем, производительность достаточна для выполнения Linux.
- Потребляемая мощность достаточная мала для применения в мобильных и переносных устройствах и в интернете вещей.
- Потребляемая мощность уменьшается путем **остановки тактового сигнала**, например, для латентных запросов шины.
- Девятистадийный конвейер с последовательным выполнением команд и аппаратной многопоточностью.

Побочный эффект закона Мура 2-13



- Существует тенденция ускоренного роста скорости арифметико-логических устройств по сравнению с ростом скорости доступа к памяти.
- При **промахе** доступа к кэш-памяти процессор может простаивать 10^2 циклов.
- Процессор может потерять **50%** своего времени на такие паузы.

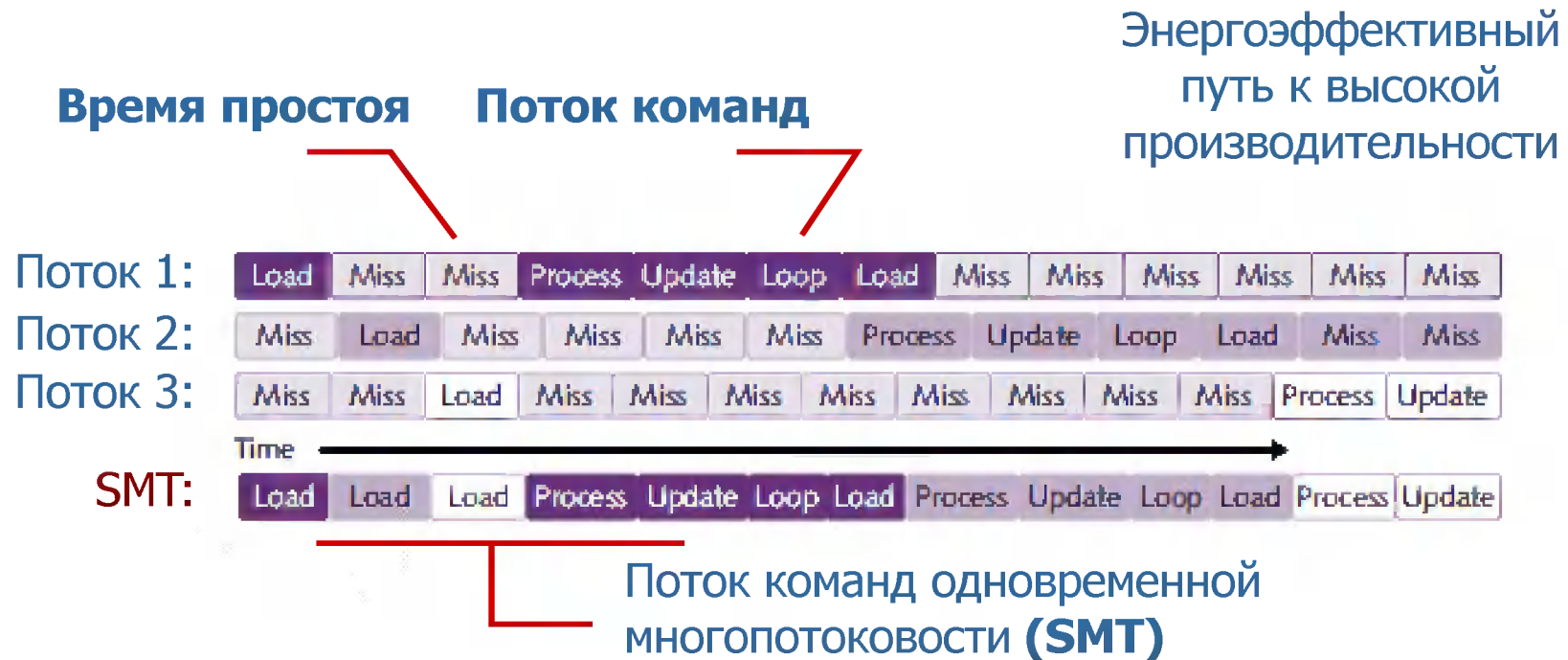
Аппаратная многопоточность (1/3) 2-14



Третья стадия выборки команды,
Instruction-Fetch Third (**Стадия IT**)

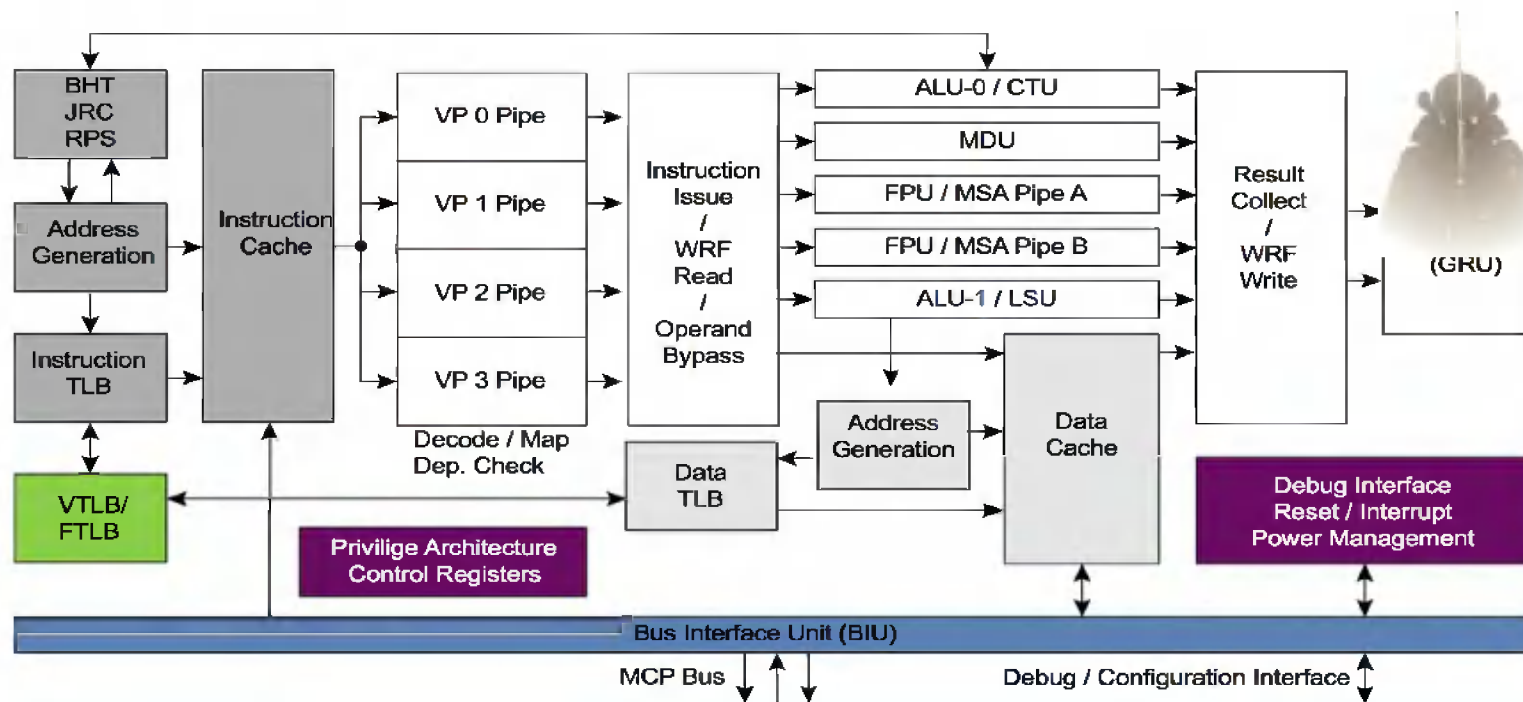
- Ядро interAptiv имеет чуть более глубокий конвейер для поддержки **SMT** (одновременной многопоточности).
- Дополнительная стадия IT назначает контекст для выбранной только что команды.
- Таким образом, отдельные потоки ссылаются на **корректные** регистры, состояния.

Аппаратная многопоточность (2/3) 2-15



- При промахе кэш-памяти время простоя используется для выполнения **другого потока**.
- Для аппаратной реализации многопоточности, каждый поток имеет собственный счетчик команд, регистры общего назначения.
- Возможность **мгновенного** (за один цикл) переключения контекста—основа SMT.
- В отличие от программной реализации, отсутствует обработка прерываний и сохранение контекста.

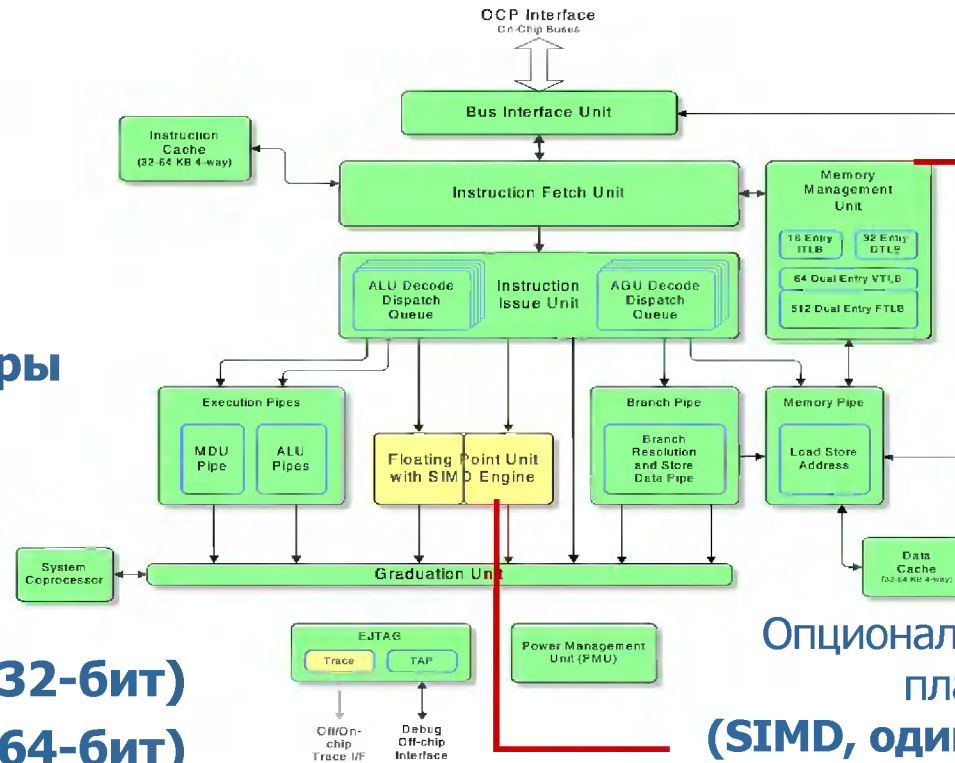
Аппаратная многопоточность (3/3) 2-16



- Как реализовать SMT без нарушения ограничений по потребляемой мощности?
- Поддерживаемое **количество потоков** (\leq глубина) является критическим выбором.
- 64-Разрядное ядро среднего уровня I-class I6400 **Samurai** использует четыре конвейера.
- Таким образом, обеспечивается одновременная многопоточность с **четырьмя** потоками.



Микросерверы



Модуль управления памятью/буфер ассоциативной трансляции (Поддержка виртуальной памяти ОС)

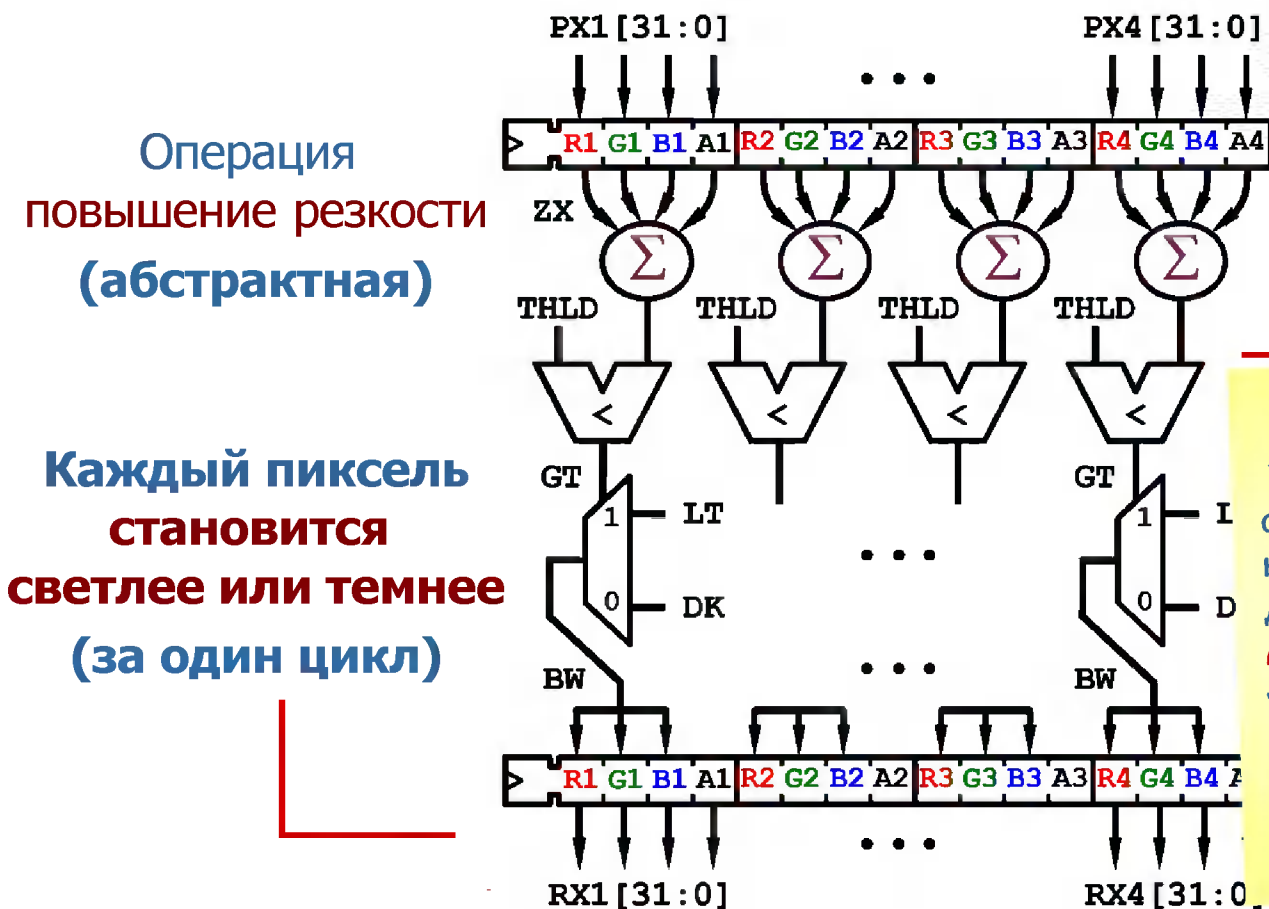
Опционально блок арифметики с плавающей точкой (SIMD, один поток команд - много потоков данных)

MIPS P5600 (32-бит)
MIPS P6600 (64-бит)

- **Наивысшая** однопоточковая производительность. Идеально подходит для приложений Java.
- Глубокий конвейер целочисленной арифметики; суперскалярная архитектура с внеочередным выполнением.
- Возможность внеочередного выполнения команд **минимизирует** приостановки.

Ускорение для мультимедийных приложений

2-18



Операция

повышение резкости
(абстрактная)

Каждый пиксель
становится
светлее или темнее
(за один цикл)

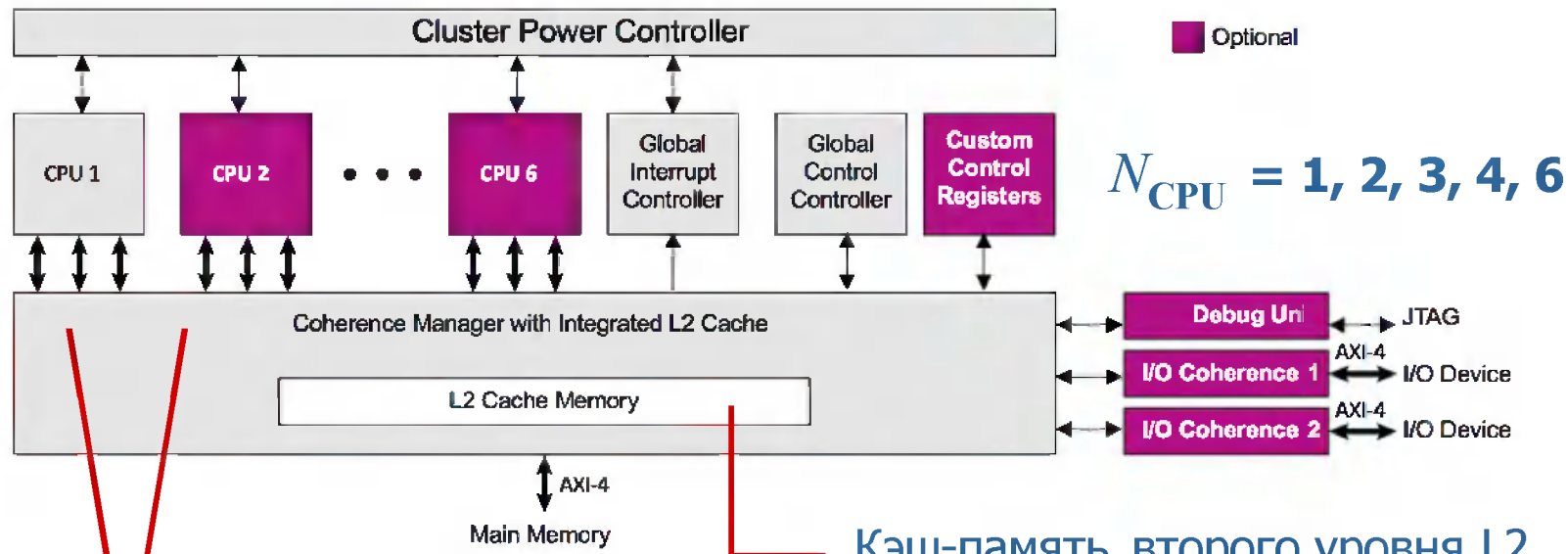
Одна команда
множество данных
(Четыре пикселя)

Другие применения архитектуры SIMD (один поток команд - много потоков данных):
Матричная математика ускоряется SIMD—помогает выполнению операций умножение-сложение при цифровой обработке сигналов.

- SIMD позволяет **параллельно** выполнять сложные арифметические операции.
- Каждая команда выполняется **синхронно** на всех параллельных блоках данных.

Многоядерные кластеры

2-19



Отдельные кэш-памяти
первого уровня L1

Интерфейс
шины AXI

Кэш-память второго уровня L2
(общая для всех)

- Включает N процессорных ядер, каждое со своей **отдельной** кэш-памятью первого уровня L1.
- Менеджер когерентности гарантирует отброс неактуальных данных.
- Поддерживает **вмешательство** в восстановление линии. Суперкластерные возможности.

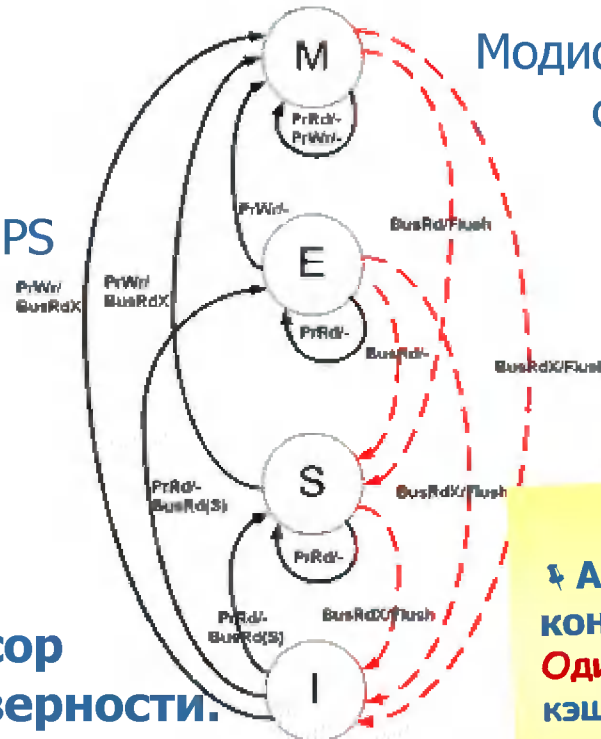
Когерентность кэш-памяти

2-20

Протокол **MESI**

Используется во всех
многоядерных кластерах MIPS
(средний-высокий
уровень)

Неактуальное состояние:
**Записывающий процессор
выдает сигнал недоверности.**



Модифицированное
состояние

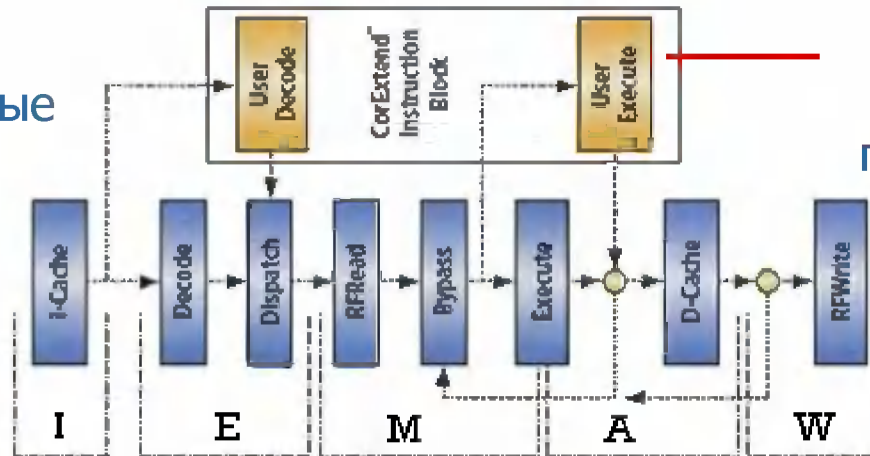
Разделяемое состояние:
**Чтение разрешено
многими ядрами.**

✦ Аппаратная реализация
конечного автомата:
Один конечный автомат на каждую
кэш-память.
Входные данные поступают с
собственного процессора или
шины.

- Области кэш-памяти, которые совместно используются N ядрами, должны быть **консистентными**.
- **Совместно** используемые блоки кэш-памяти на являются проблемой при **чтении**.
- Однако, **запись** слова требует более **исключительного** доступа.
- Встроенный конечный автомат **упрощает** задачу программиста.

Конфигурируемая обработка 2-21

Cor Extend
Команды, определенные
пользователем



Команды, определенные
пользователем
(UDI)

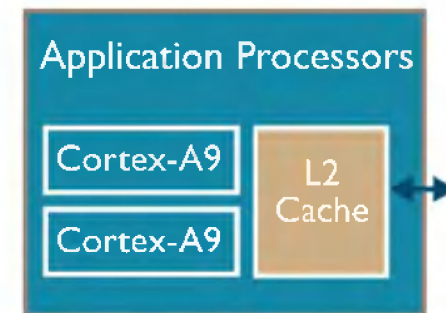


- Расширяет стандартный набор MIPS командами, **определенными пользователем**.
- Работают с **регистрами** общего назначения, промежуточными данными и т.д.
- Тесно связан с блоком исполнения команд четко определенными сигналами.
- Несколько специализированных команда могут **значительно повысить скорость**.

Характеристики IP-блоков:

- Программные/аппаратные IP-блоки
- IP-блоки в топологическом формате
- Топологические IP-ядра ARM
- Протоколы шины ARM

Типичные
IP-блоки



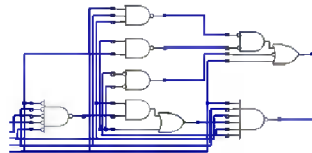
Форматы IP-ядер (1/2)

2-23

Программные
IP-блоки
(RTL код)

```
1 /* RTL Verilog code:
2  * Four-bit serializer,
3  * with jbbhex control.
4  */
5 module SERIAL4(
6  output wire TX, //serial.
7  input wire [3:0] MAC_DATA,
8  input wire LOAD_SHFT, HAZ, CLK
9  );
10 reg [3:0] FF; //Four flops.
11
12 always @(posedge CLK)
13   if (LOAD_SHFT)
14     FF <- MAC_DATA; //Load data.
15   else
16     FF <- {1'b1,FF[3:1]}; //shift.
17
18   assign TX = FF[0] HAZ; //ok.
19
20 endmodule
```

Аппаратные
IP-блоки
(Список
соединений)



Аспекты:

- **Модифицируемый пользователем RTL код**
- Синтезируются вместе с прочим RTL-кодом.

Аспекты:

- Можно подстраивать
- Могут синтезироваться по требованиям пользователя
- Может быть запутанным RTL— нечитабельным, защищенным!

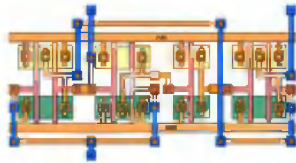
↳ Экономика IP:
Программные IP-блоки **можно модифицировать** — но поставщики защищают свои продукты.

- В целом, существует тенденция к наибольшему распространению **программных IP-блоков**.
- Передается **RTL код** (возможно зашифрованный или сгенерированный).
- Передача читаемого RTL более распространена для **младших ядер**.
- Последователи ядер контролируют процесс логического синтеза, ход физического проектирования.

Форматы IP-ядер (2/2)

2-24

Топологические
IP-блоки
(Топологический
чертеж)



Аспекты:

- Оптимизированы под одну технологию
- Пользователь не может изменить
- Физически спроектированы поставщиком.

Поставляется:

- Файл **GDSII** топологии.
- Список соединений Verilog:
- Файлы .lef, .lib.
- Список соединений LVS.
- Модели для моделирования.
- Тестовые векторы для автоматической тестирующей установки



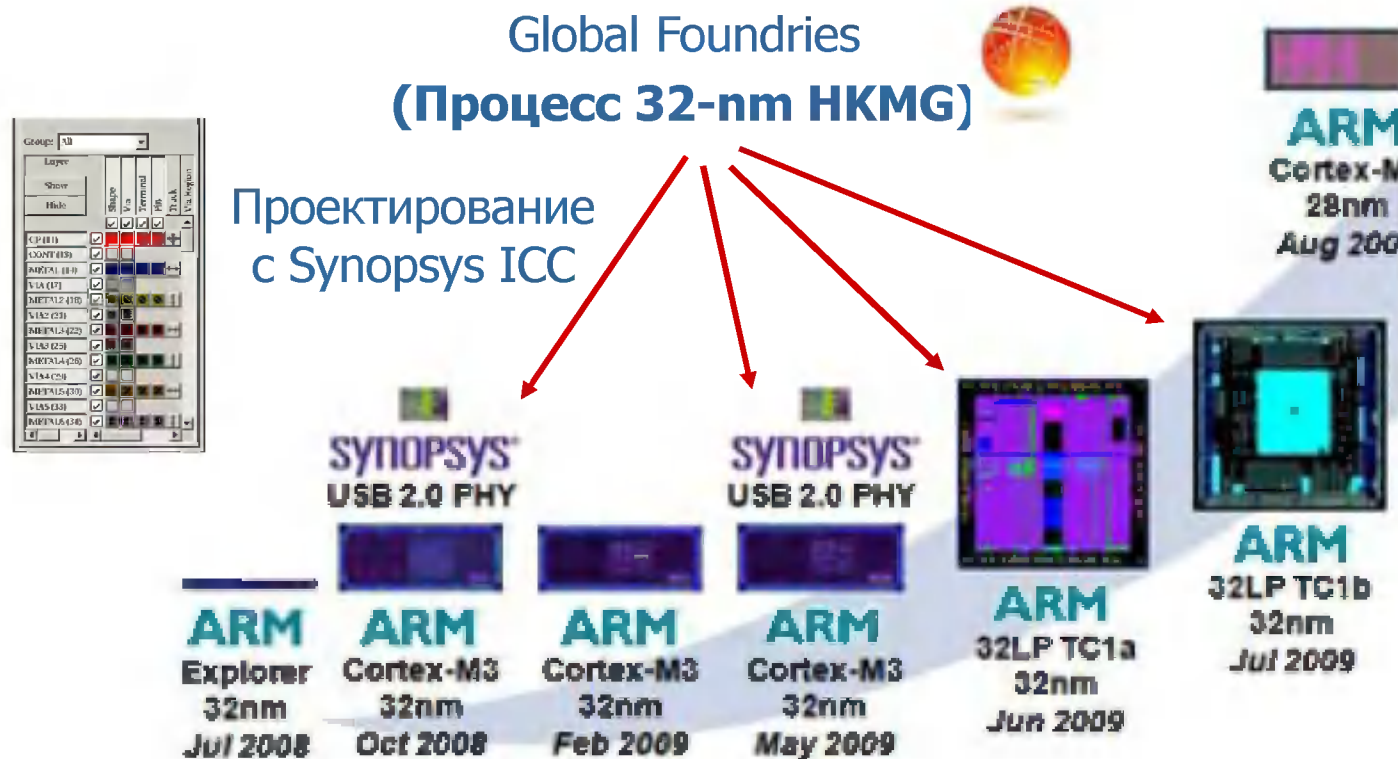
↳ Экономика IP:

Топологические IP-блоки могут быть в 10 дешевле из-за сложности изменения их пользователем.

- Топологические IP-блоки невозможно изменить, они, как правило, самые дешевые.
- Распространенный формат для **устаревших** ядер, поставляются на условия «как есть», без гарантий.
- Аналоговые IP-блоки или высокоскоростные цифровые интерфейсы всегда **ТОПОЛОГИЧЕСКИЕ**.

Пример ARM: Топологические IP-блоки

2-25



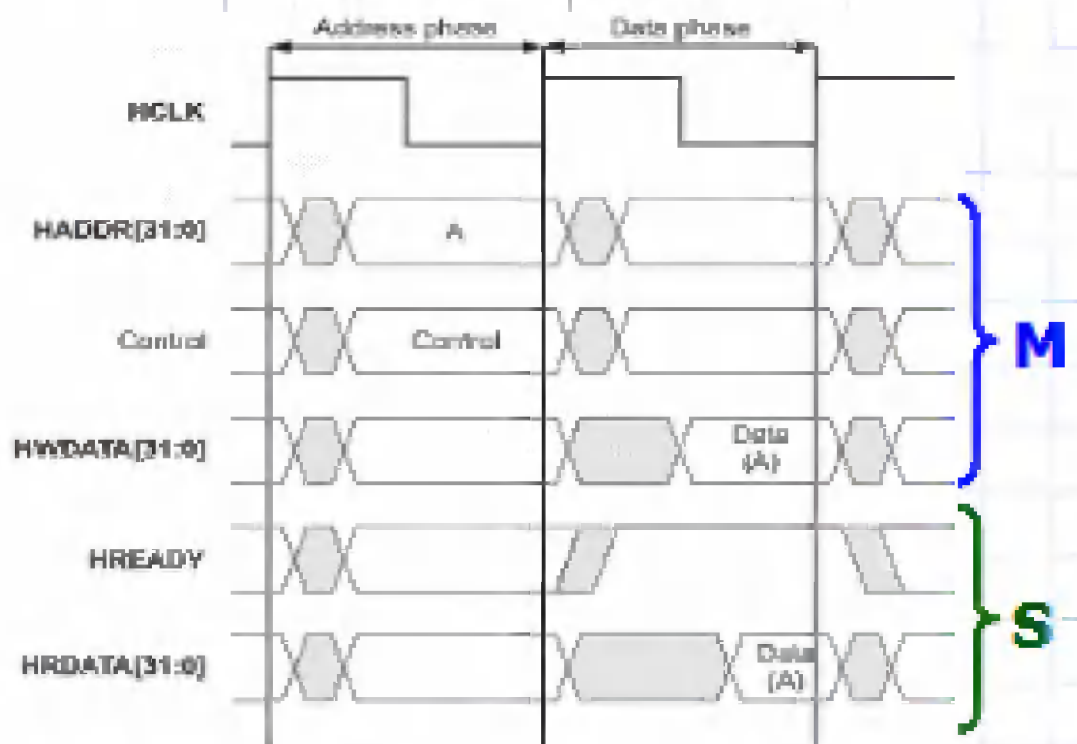
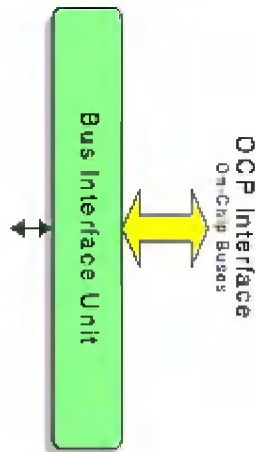
- Мягкий физический слой может каждый раз создавать различные файлы GDSII.
- Жесткий физический слой имеет согласованные задержки, меньшее дрожание фазы и ее разброс.
- Топологические IP-блоки могут разрабатываться одновременно с целевым технологическим процессом.

Пример: Протокол шины

2-26

Протокол АНВ

Современная шина
для микроконтроллеров
Архитектуры (АМВА)



- **AMBA** компании ARM является открытым стандартом для внутренних соединений на кристалле.
- Соединяет и управляет функциональными блоками в микросхемах систем на кристалле.
- Таким образом, АНВ, АХІ, и т.д. работают **как** с ядрами ARM, **так** и MIPS.
- Шины работают параллельно с **целочисленным конвейером** MIPS.

При приобретении и использовании IP-блоков могут возникнуть следующие вопросы:

- ✦ Интегрировать IP-блоки в кристалл при синтезе RTL (программные) или при разработке топологии (топологические блоки)?
- ✦ Оценить влияние блоков на **бюджет** всего кристалла: какая часть кристалла будет занята большими блоками IP?
- ✦ Интегрировать IP-блоки в общую стратегию **DFT** (проектирование для тестирования).
- ✦ Обеспечить раннюю доступность **моделей для моделирования** - особенно для топологических блоков (для которых RTL-код недоступен).
- ✦ Корпоративные сайты для проектировщиков для доступа к IP-блокам, отслеживания обновлений и исправления ошибок.