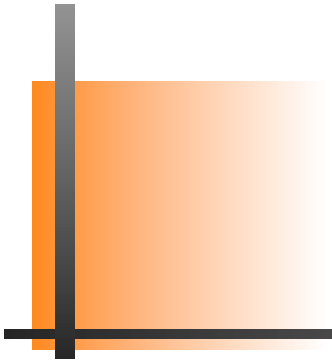
INF5063:
Programming heterogeneous multi-core processors

Introduction

September 13, 2010

Overview

- Course topic and scope
- Background for the use and parallel processing using heterogeneous multi-core processors
- Examples of heterogeneous architectures



INF5063: The Course

People

- Håvard Espeland
  email: haavares @ ifi

- Håkon Kvale Stensland
  email: haakonks @ ifi

- Carsten Griwodz
  email: griff @ ifi

- Pål Halvorsen
  email: paalh @ ifi

Time and place

- **Lectures**:
  Fridays 13.15 – 15.00
  Store Aud. ??? Veilabben???

- NB!
  The web page states that we will have
  group exercises on
  Thursdays 10.15 - 12.00, 3B.
  However, there will **NOT** be any weekly exercises, but this hour is assigned for your mandatory assignments (we will NOT be there).

About INF5063: Topic & Scope

- **Content:** The course gives …
  - … an overview of heterogeneous multi-core processors in general and three variants in particular and a modern general-purpose core (architectures and use)
  - … an *introduction* to working with heterogeneous multi-core processors
    - Intel IXP 2400 network processor card
    - SSE*x* for x86
    - nVIDIA's family of GPUs and the CUDA programming framework
    - The Cell Broadband Engine Architecture
  - … some ideas of how to use/program heterogeneous multi-core processors

About INF5063: Topic & Scope

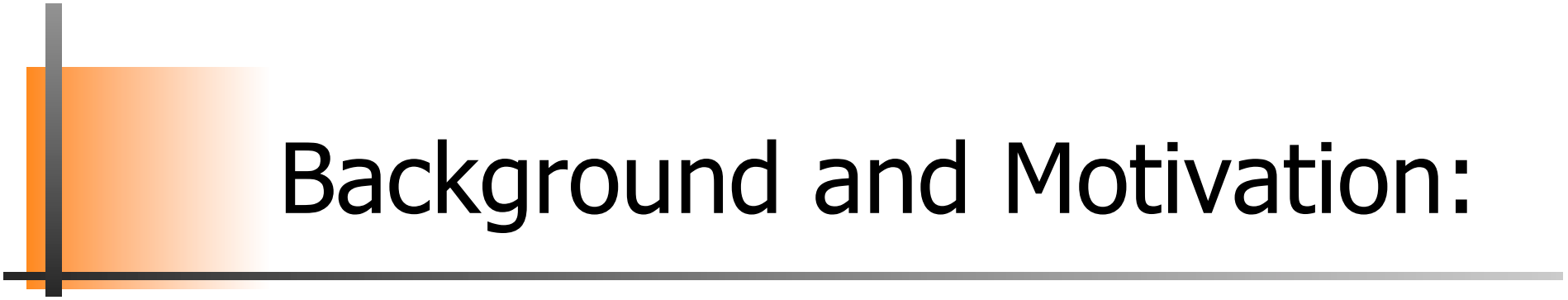
- **Tasks:**
  The important part of the course is lab-assignments where you program each of the three examples of heterogeneous multi-core processors

- 1 exercise (not graded) on **Intel IXP**
  - packet counter – download, run and extend ***wwpingbump***
- 3 graded home exams (counting 33% each):
  - Deliver code
  - Make a demonstration and explain your design and code to the class

  1. On the **x86**
     - Video encoding – Improve performance of *video compression* by using SSE instructions.
  2. On the **nVIDIA graphics cards**
     - Video encoding – Improve the performance of *video compression* by using the G80 architecture
  3. On the **Cell processor**
     - Video encoding – the same as above, but exploit the parallelity of the Cell processor's SBEs

Available Resources

- Resources will be placed at
  - http://www.ifi.uio.no/~griff/INF5063
  - Login: *inf5063*
  - Password: *ixp*
  - Manuals, papers, code example, …



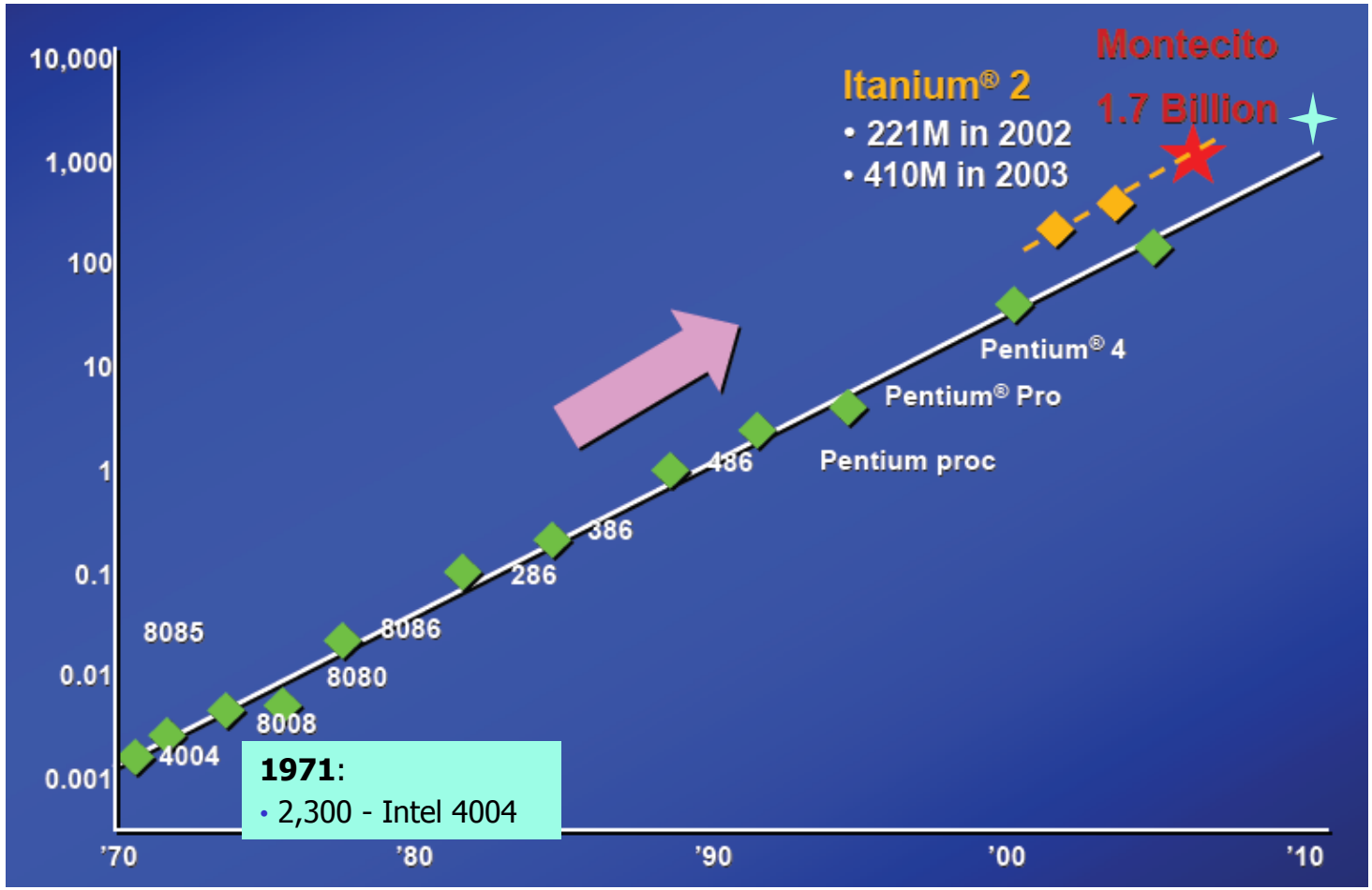
Background and Motivation:

Moore's Law

Motivation: Intel View

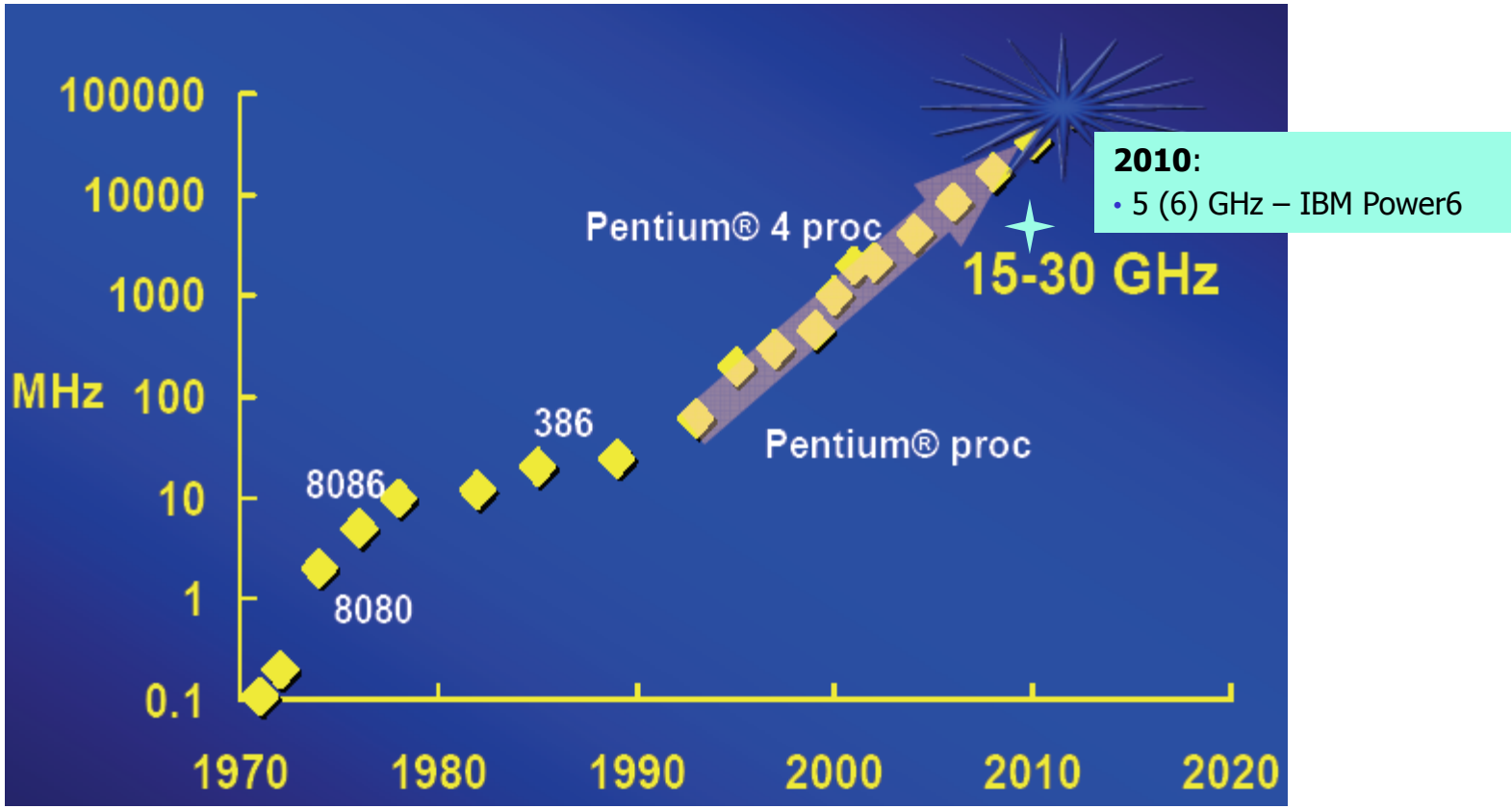
- >billion transistors integrated

2010:
- 2,3 billion - Intel 8-Core Xeon Nehalem-EX
- 3,0 billion - nVidia GF100 (Fermi)



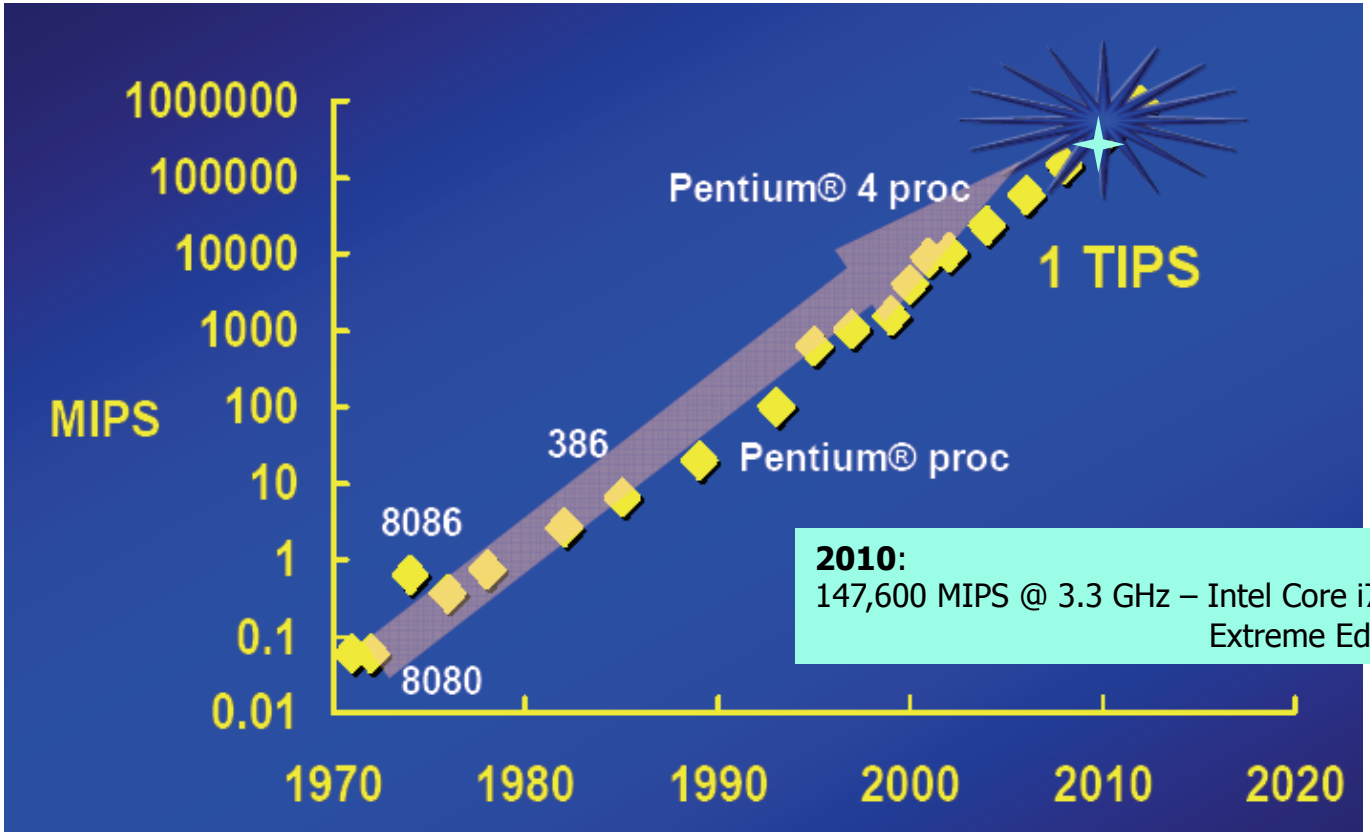
Motivation: Intel View

- >billion transistors integrated
- Clock frequency **can** still increase



Motivation: Intel View

- >billion transistors integrated
- Clock frequency **can** still increase
- Future applications will demand TIPS

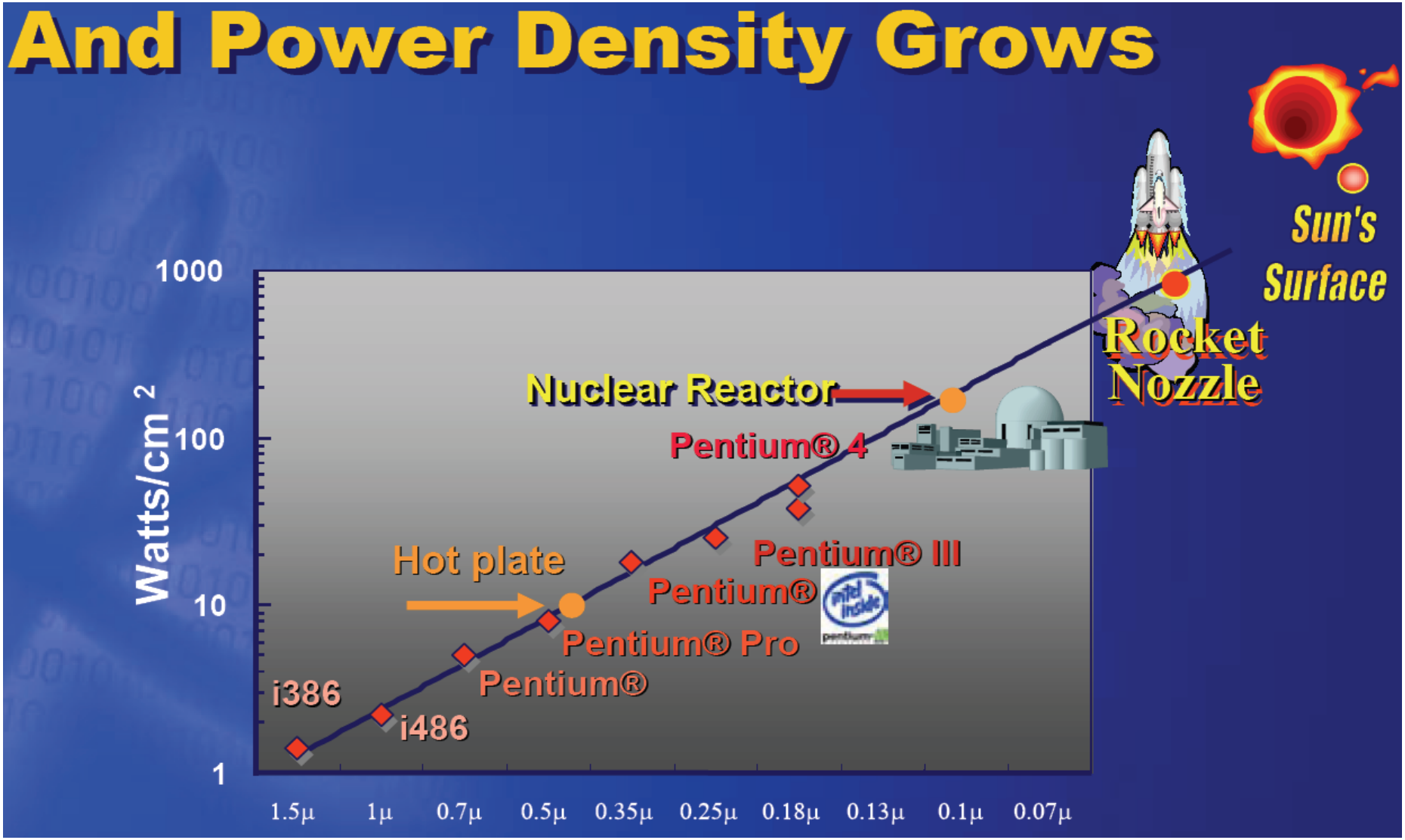


2010:
147,600 MIPS @ 3.3 GHz – Intel Core i7 Extreme Edition i980EE (6 cores)

Motivation: Intel View

- >billion transistors integrated
- Clock frequency **can** still increase
- Future applications will demand TIPS
- Power? Heat?

Motivation: Intel View



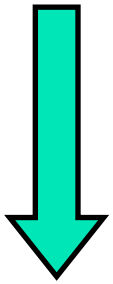
Motivation

"Future applications will demand TIPS"

"Think platform beyond a single processor"

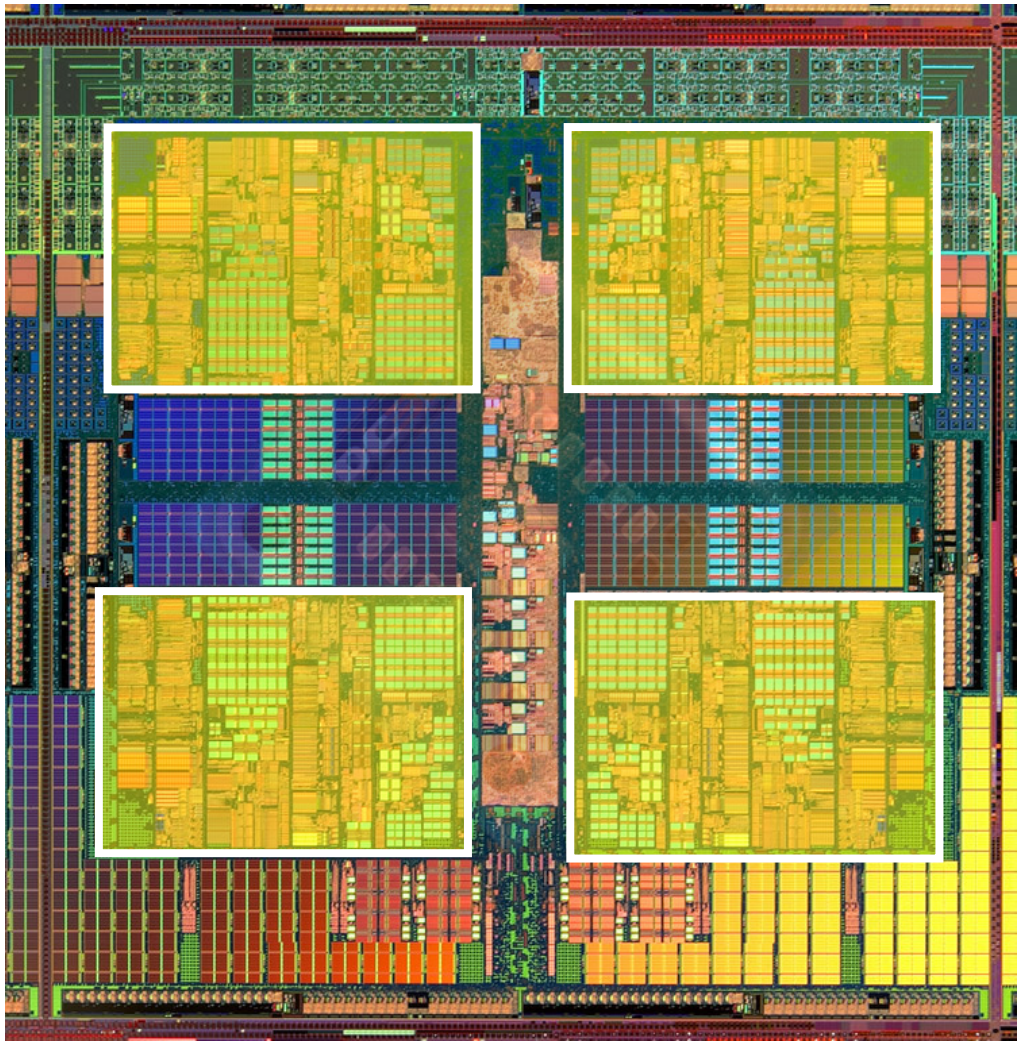
"Exploit concurrency at multiple levels"

"Power will be the limiter due to complexity and leakage"



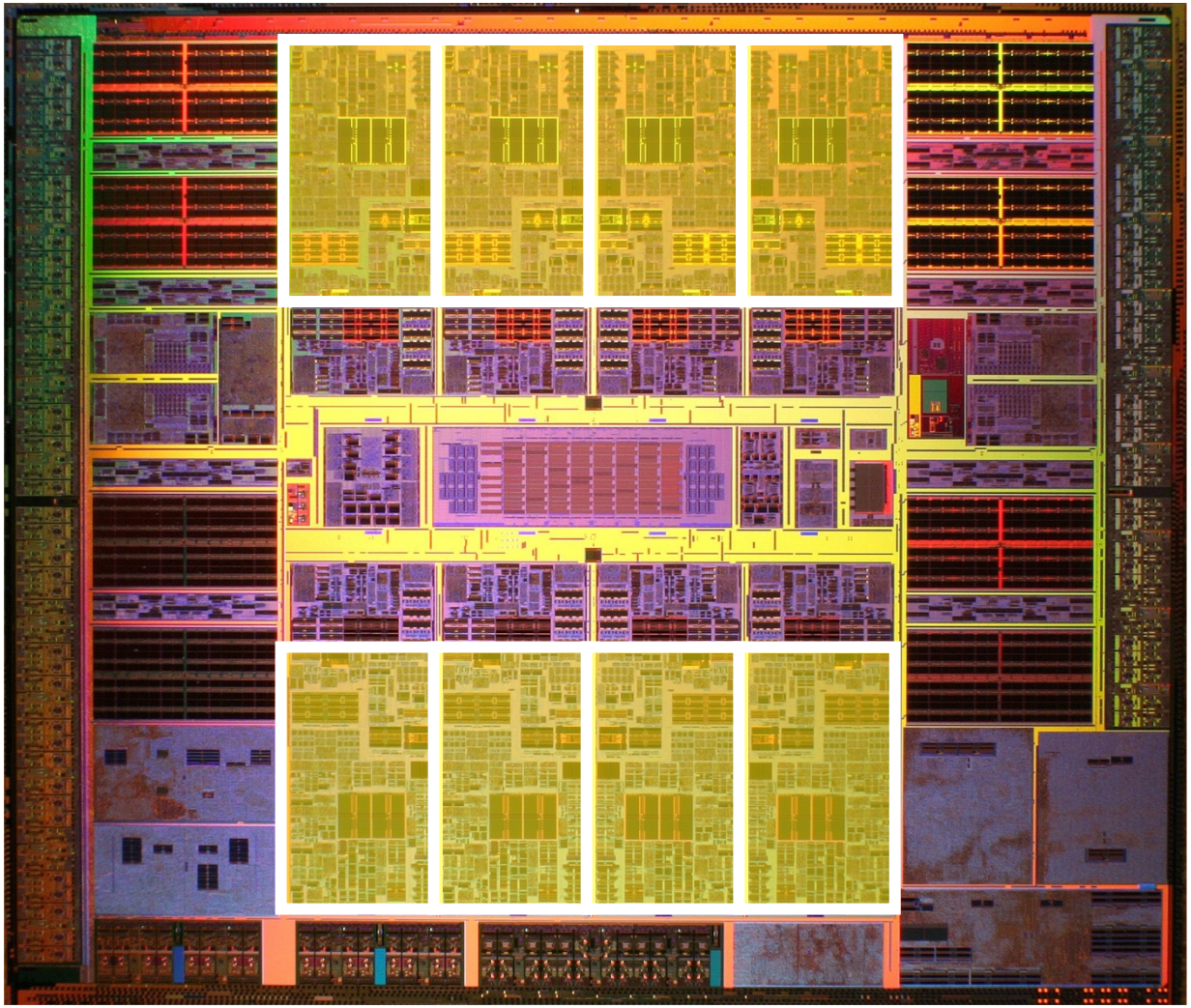
Distribute workload on multiple cores

Symmetric Multi-Core Processors



Phenom X4

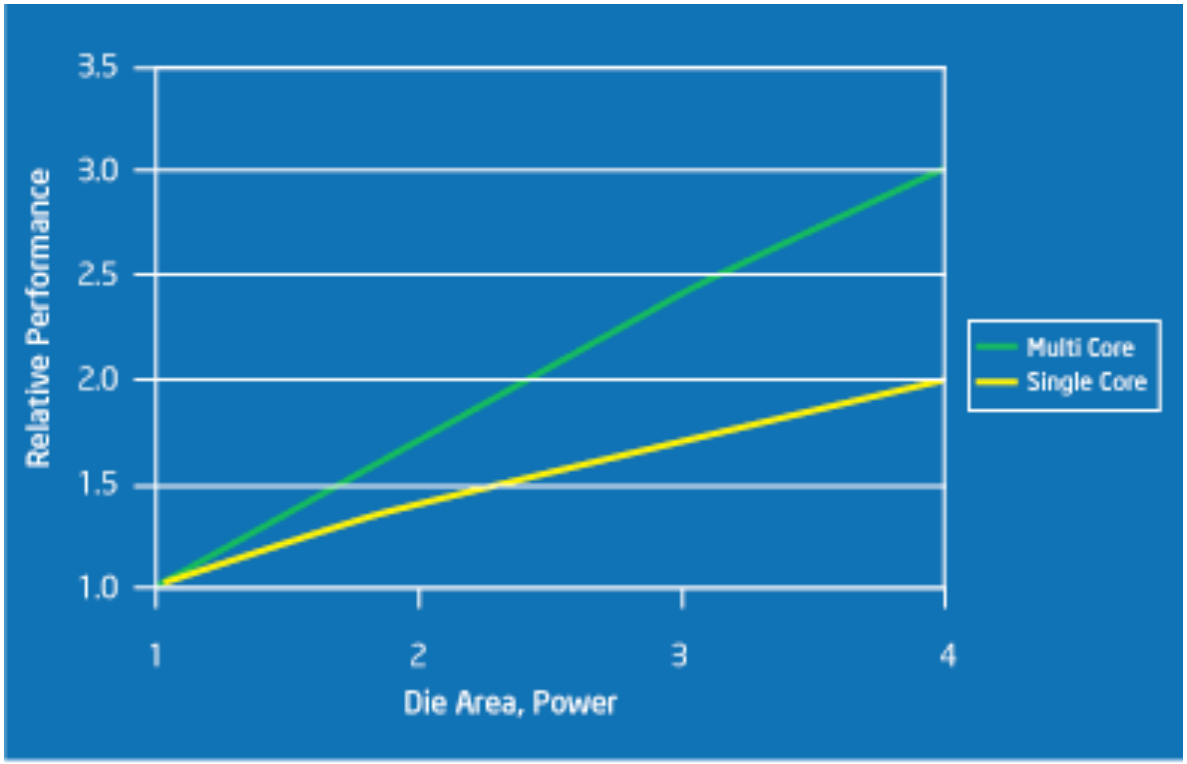
Symmetric Multi-Core Processors



UltraSparc

Intel Multi-Core Processors

- Symmetric multi-processors allow multi-threaded applications to achieve higher performance at less die area and power consumption than single-core processors



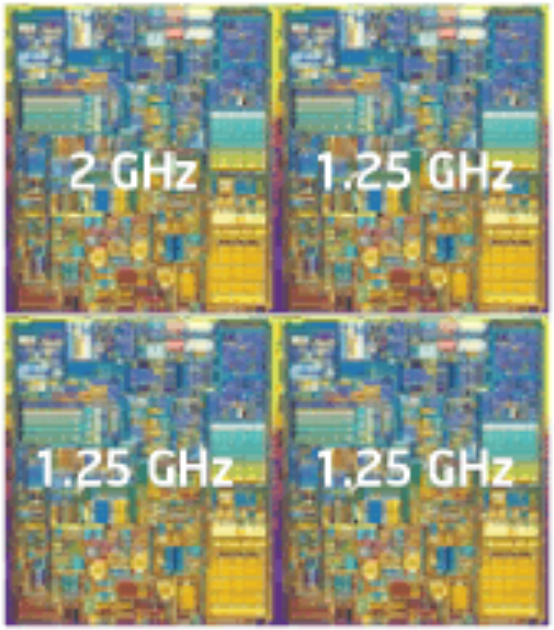
Symmetric Multi-Core Processors

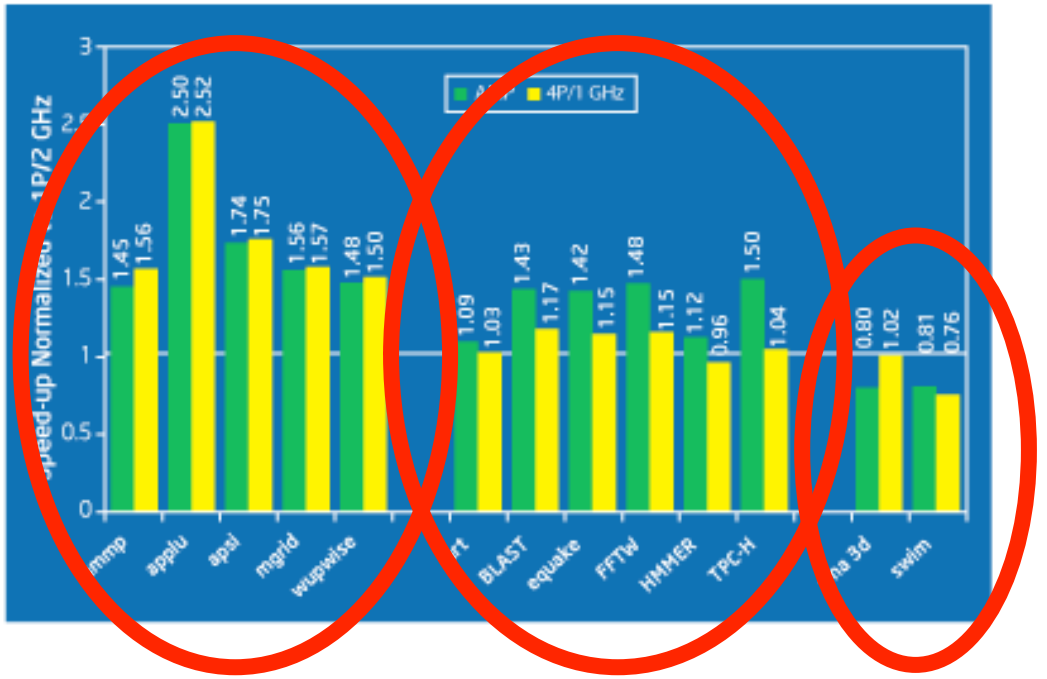
- Good
  - Growing computational power
- Problematic
  - Growing die sizes
  - Unused resources
    - Some cores used much more than others
    - Many core parts frequently unused
- Why not spread the load better?
  - Functions exist only once per core
  - Parallel programming is hard

⇒ Asymmetric multi-core processors

Asymmetric Multi-Core Processors

- Asymmetric multi-processors consume power and provide increased computational power only on demand





Highly parallel **Moderately parallel** **Sequential**

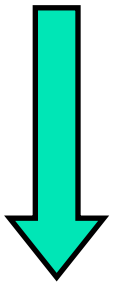
Motivation

"Future applications will demand TIPS"

"Think platform beyond a single processor"

"Exploit concurrency at multiple levels"

"Power will be the limiter due to complexity and leakage"



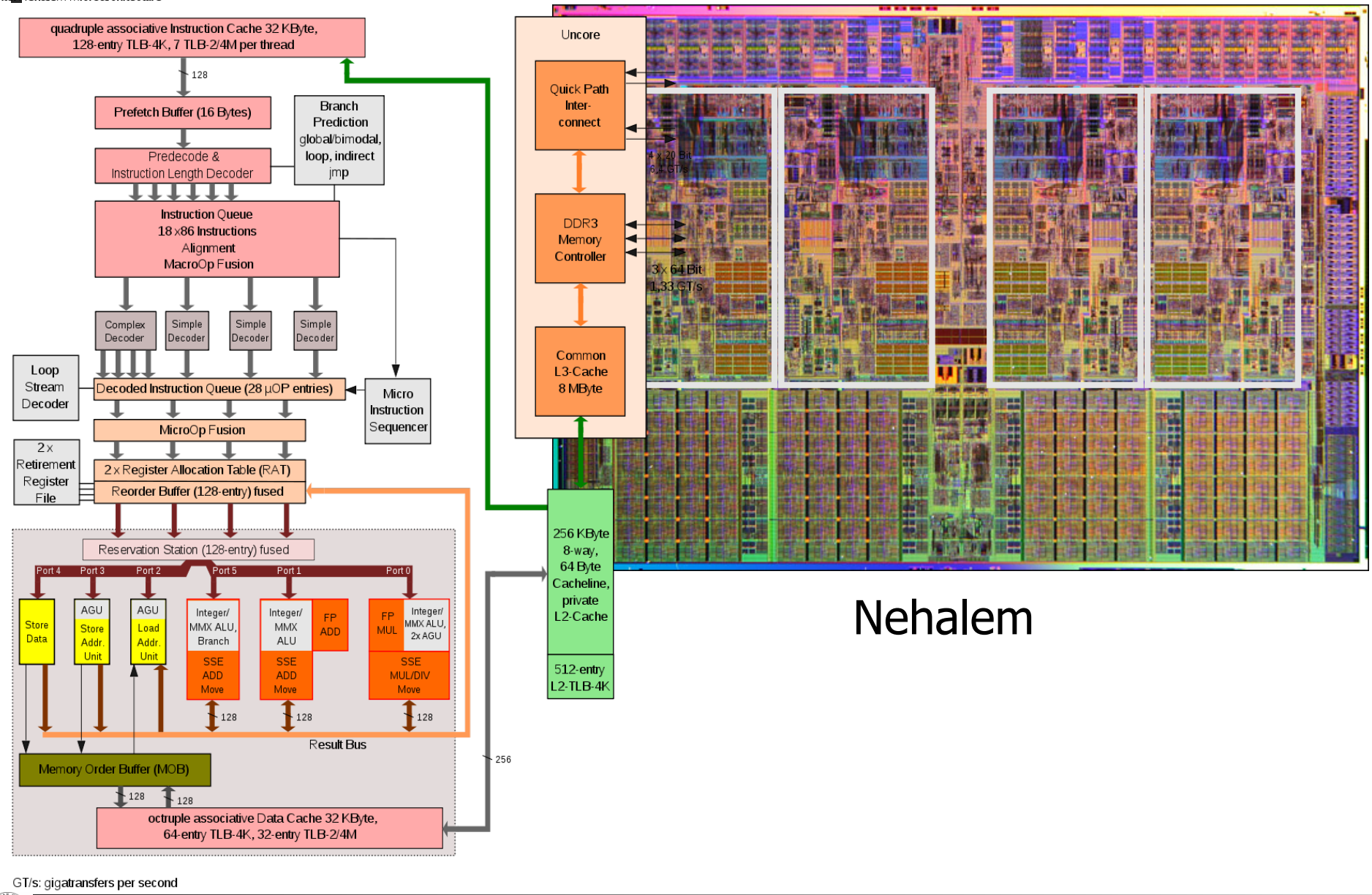
Distributed workload on multiple cores
+ simple processors are "easier" to program
+consume less energy

↳ **heterogeneous multi-core processors**

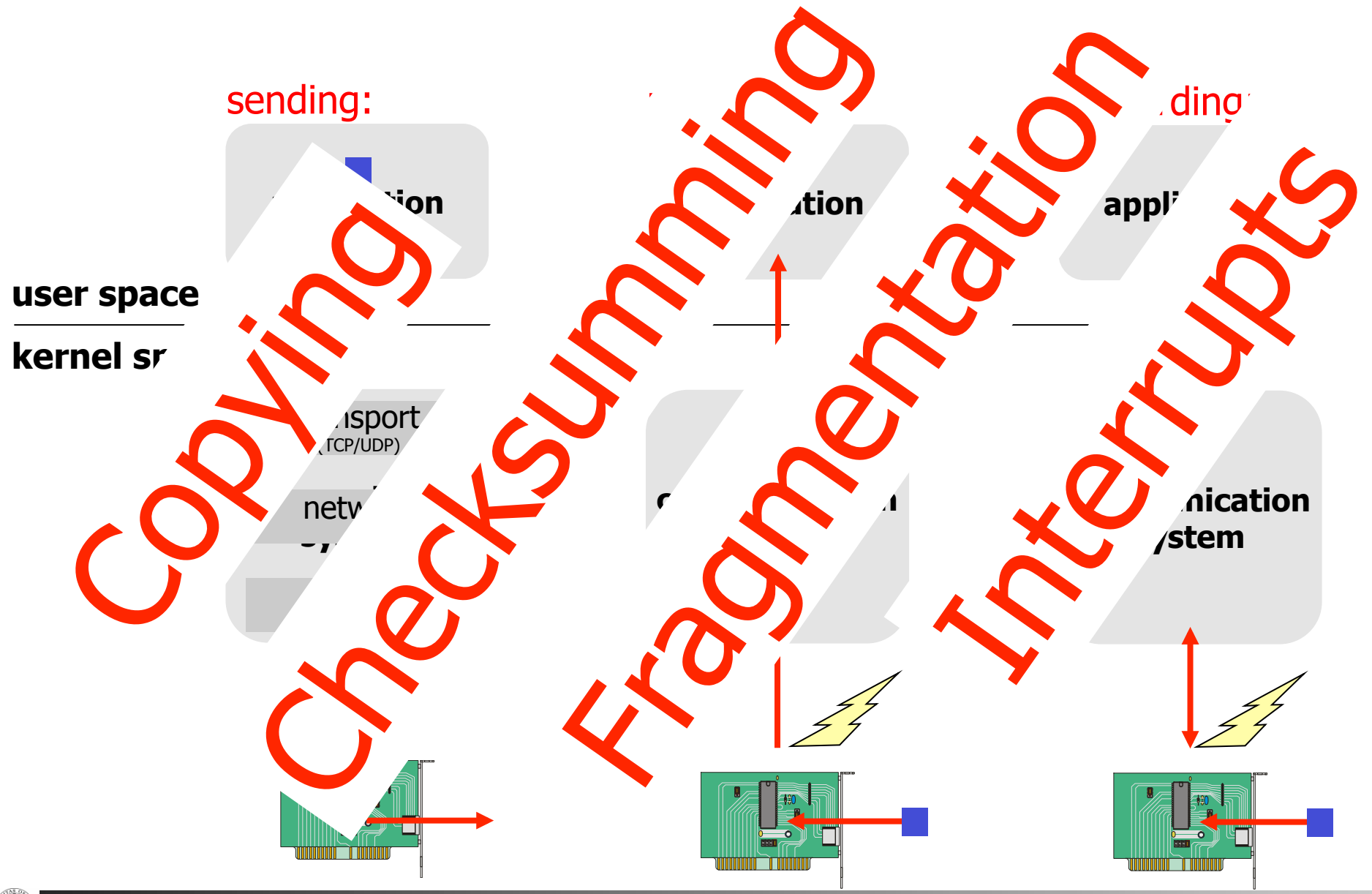
Co-Processors

- The original IBM PC included a socket for an Intel 8087 floating point co-processor (FPU)
  - 50-fold speed up of floating point operations
- Intel kept the co-processor up to i486
  - 486DX contained an optimized i487 block
  - Still separate pipeline (pipeline flush when starting and ending use)
  - Communication over an internal bus
- Commodore Amiga was one of the earlier machines that used multiple processors
  - Motorola 680x0 main processor
  - Blitter (block image transferrer - moving data, fill operations, line drawing, performing boolean operations)
  - Copper (Co-Processor - change address for video RAM on the fly)

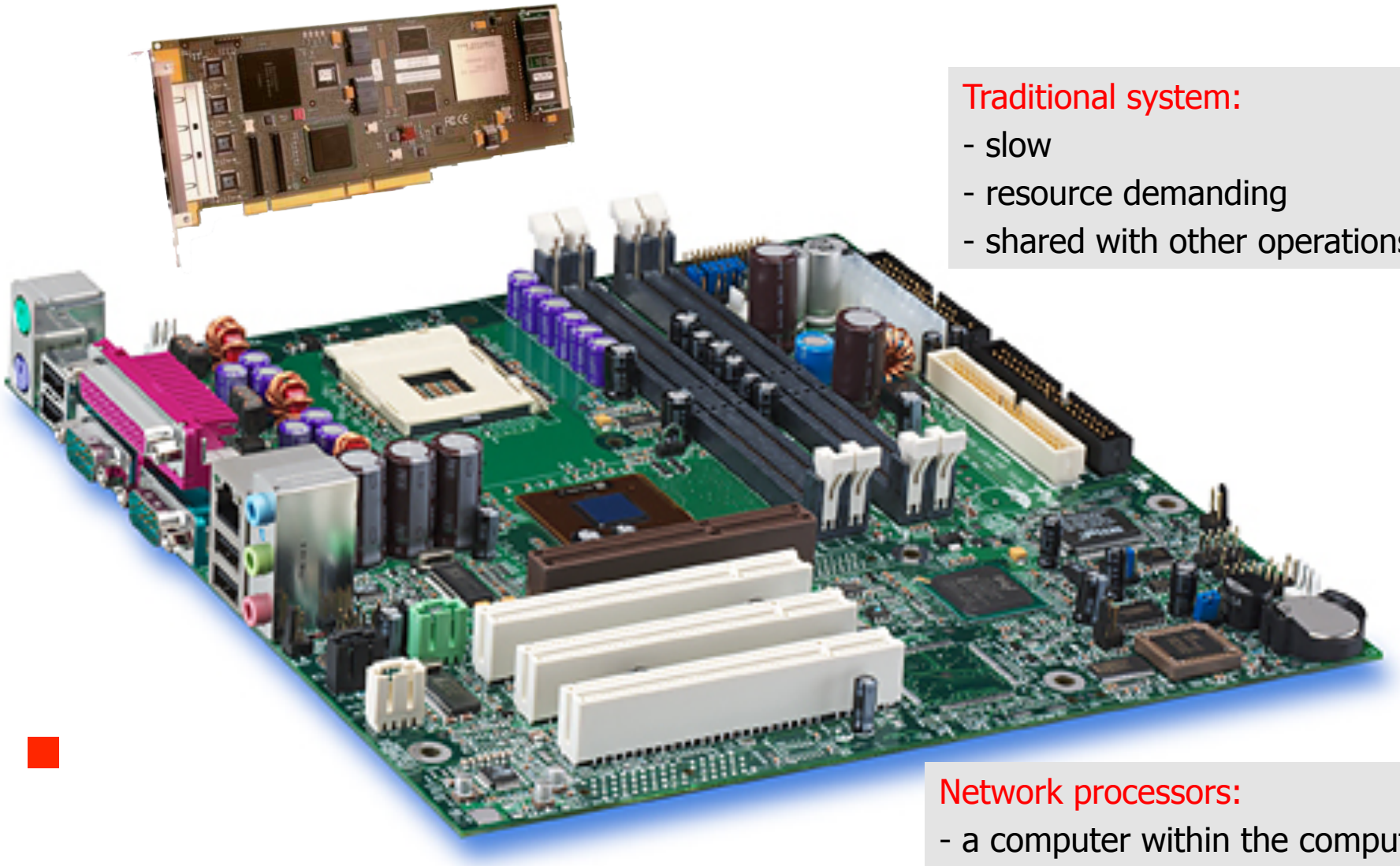
What now – are today's cores really "Symmetric"?



Review of General Data Path on Conventional Computer Hardware Architectures



Network Processors: Main Idea



Traditional system:
- slow
- resource demanding
- shared with other operations

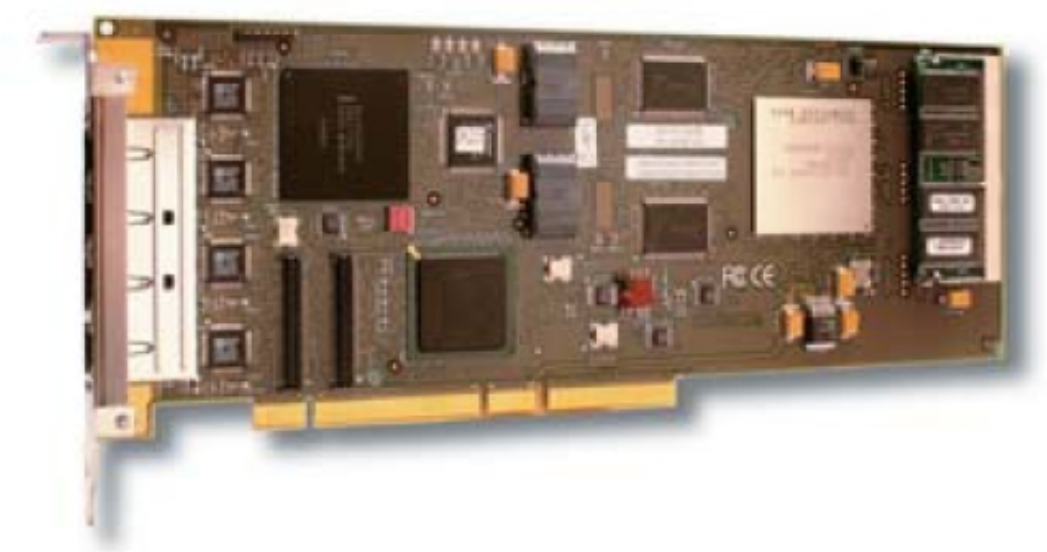
Network processors:
- a computer within the computer
- special, programmable hardware
- offloads host resources

IXA: Internet Exchange Architecture

- IXA
  - a broad term to describe the Intel network architecture
  - HW & SW, control- & data plane
- IXP: Internet Exchange Processor
  - processor that implements IXA
  - IXP1200 is the first IXP chip (4 versions)
  - IXP2xxx has now replaced the first version

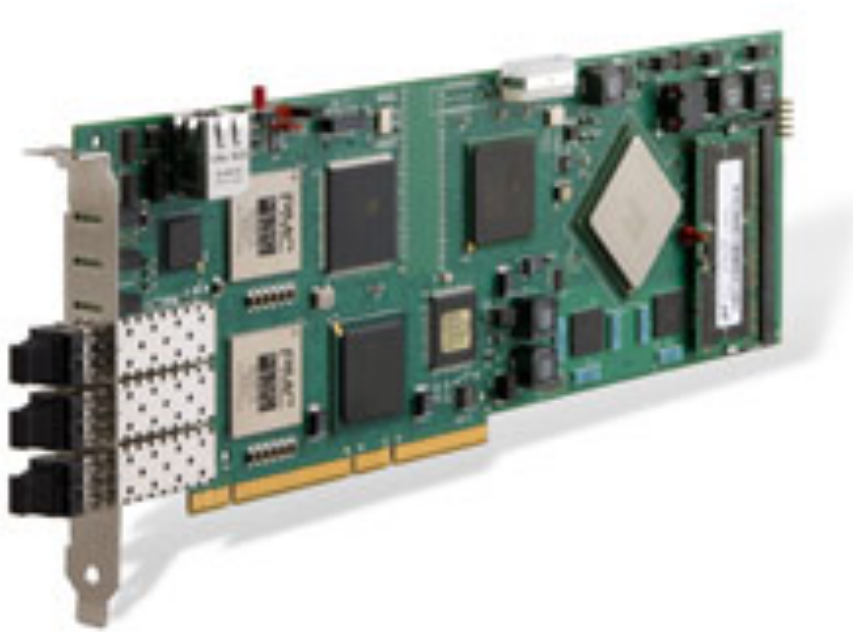
IXA: Internet Exchange Architecture

- **IXP1200** basic features
  - 1 embedded 232 MHz StrongARM
  - 6 packet 232 MHz *µengines*
  - onboard memory
  - 4 x 100 Mbps Ethernet ports
  - multiple, independent busses
  - low-speed serial interface
  - interfaces for external memory and I/O busses
  - ...

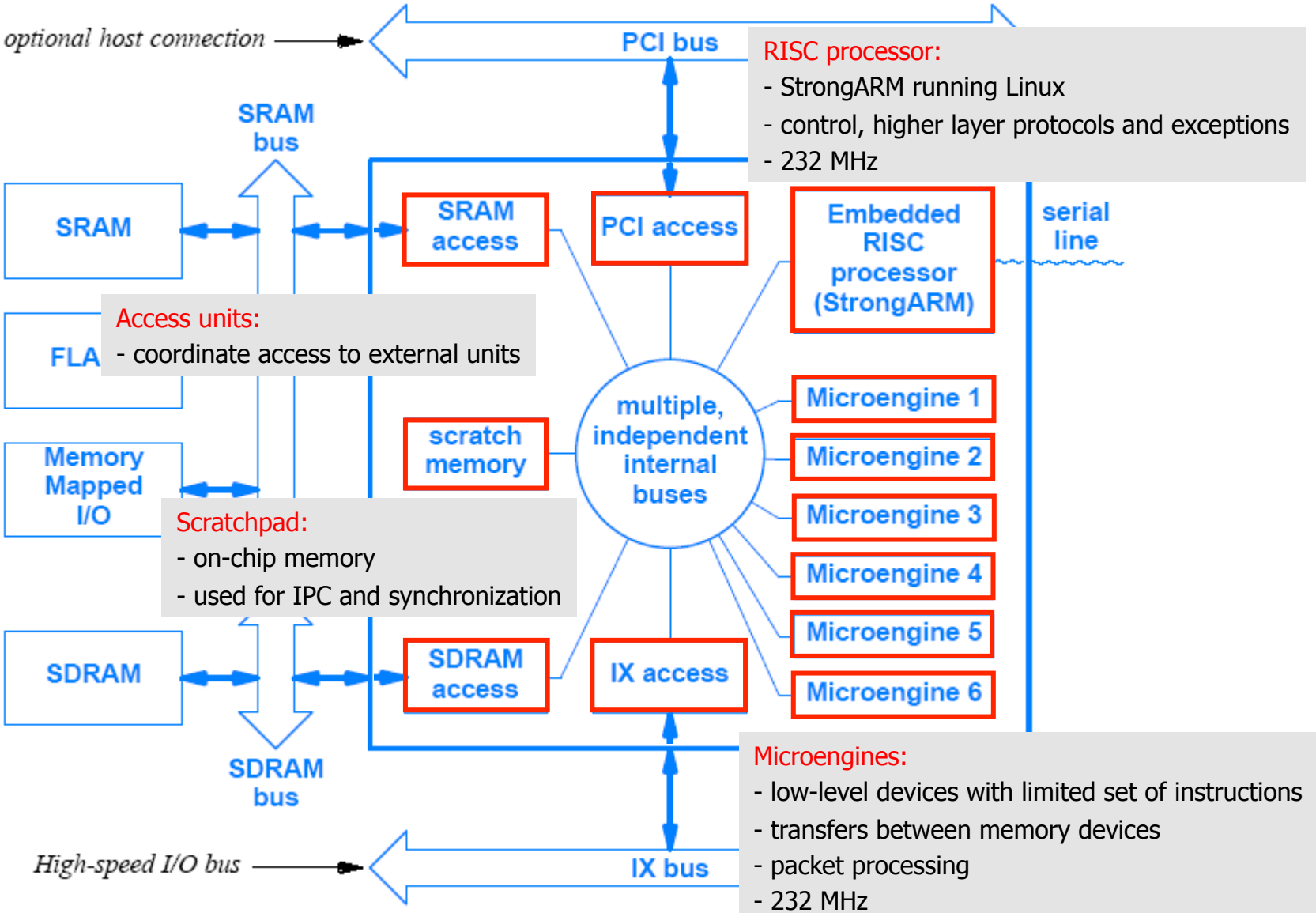


IXA: Internet Exchange Architecture

- **IXP2400** basic features
  - 1 embedded 600 MHz XScale
  - 8 packet 600 MHz *µengines*
  - onboard memory
  - 3 x 1 Gbps Ethernet ports
  - multiple, independent busses
  - low-speed serial interface
  - interfaces for external memory and I/O busses
  - …



IXP1200 Architecture



RISC processor:
- StrongARM running Linux
- control, higher layer protocols and exceptions
- 232 MHz

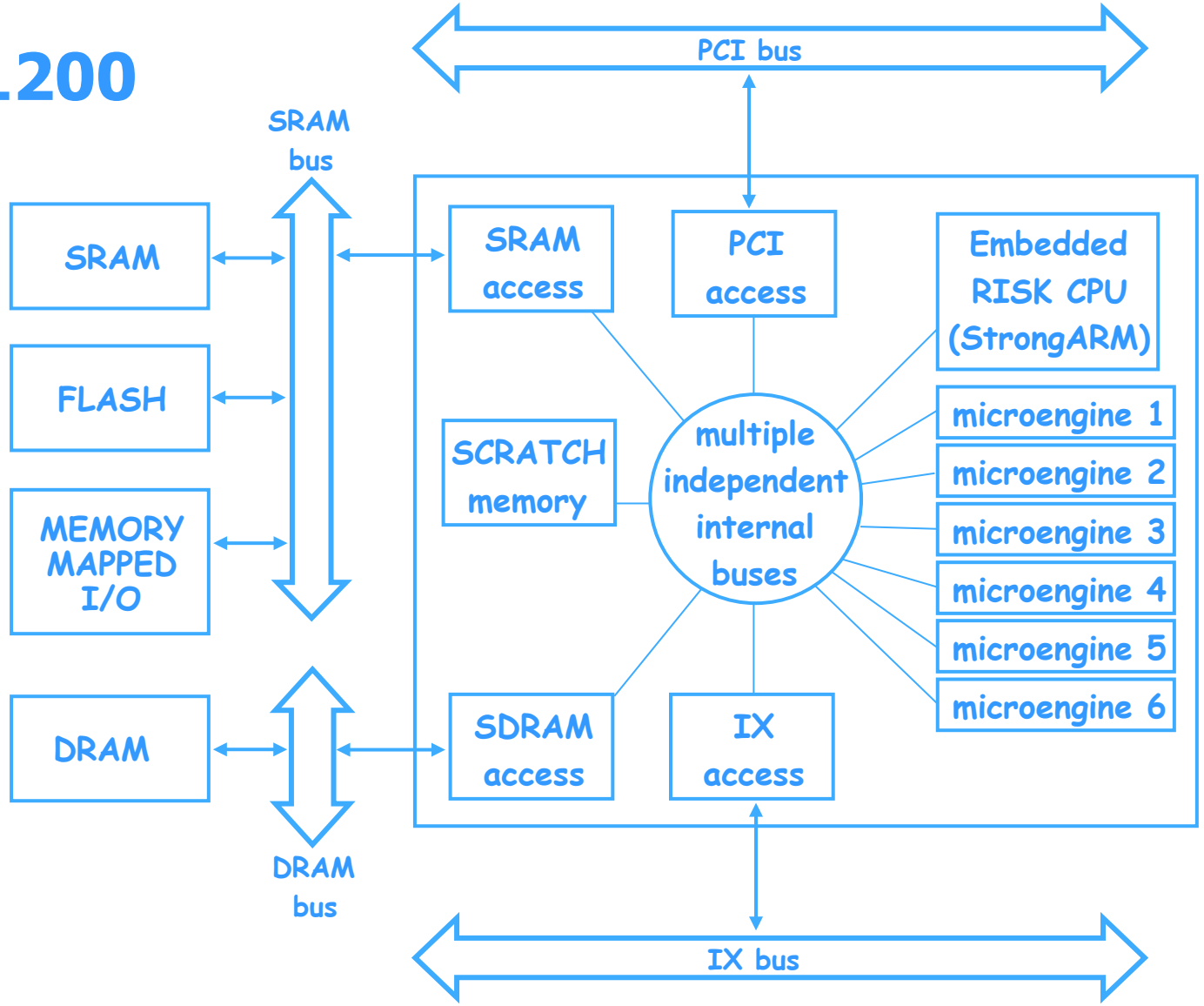
Access units:
- coordinate access to external units

Scratchpad:
- on-chip memory
- used for IPC and synchronization

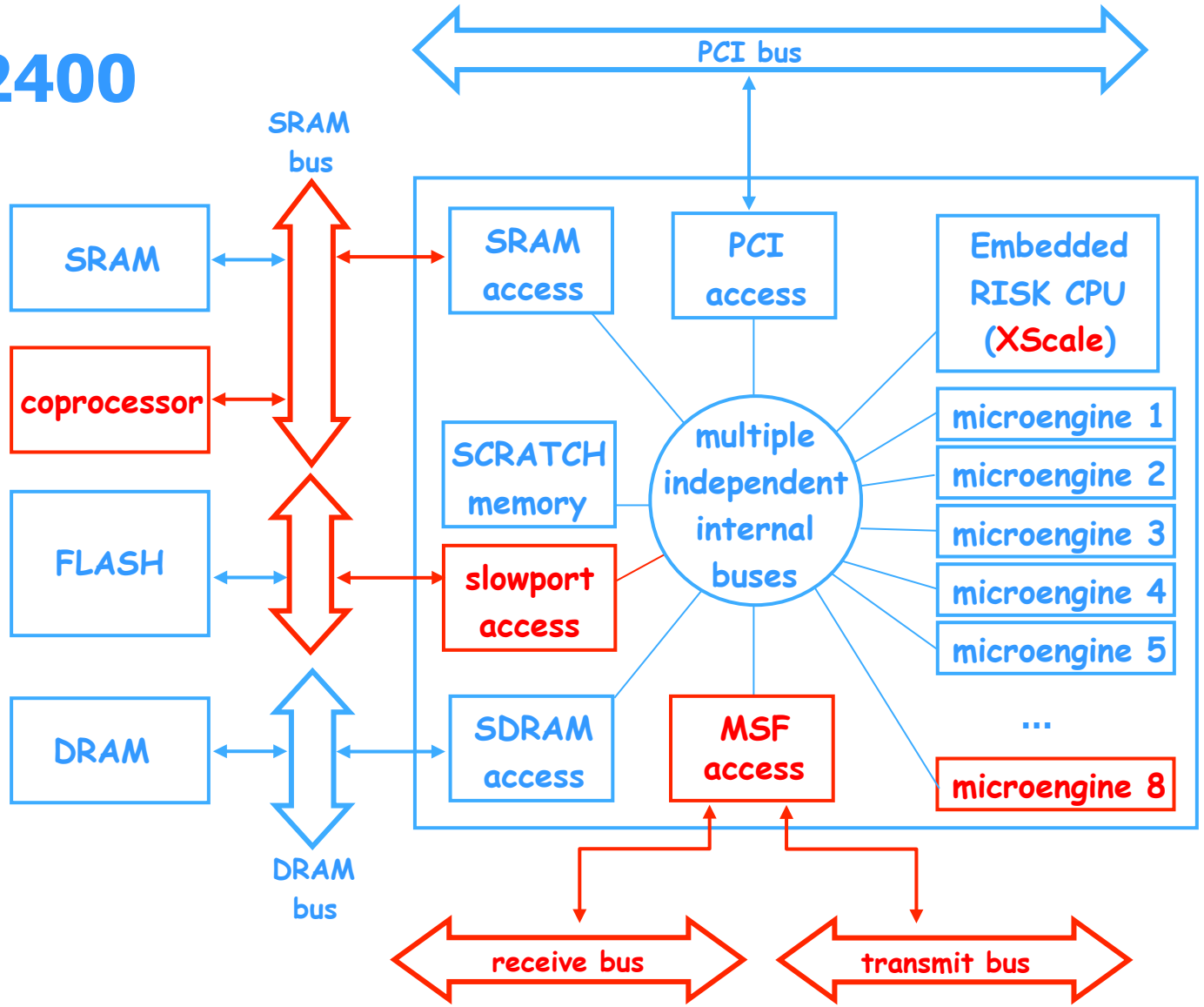
Microengines:
- low-level devices with limited set of instructions
- transfers between memory devices
- packet processing
- 232 MHz

IXP1200 → IXP2400

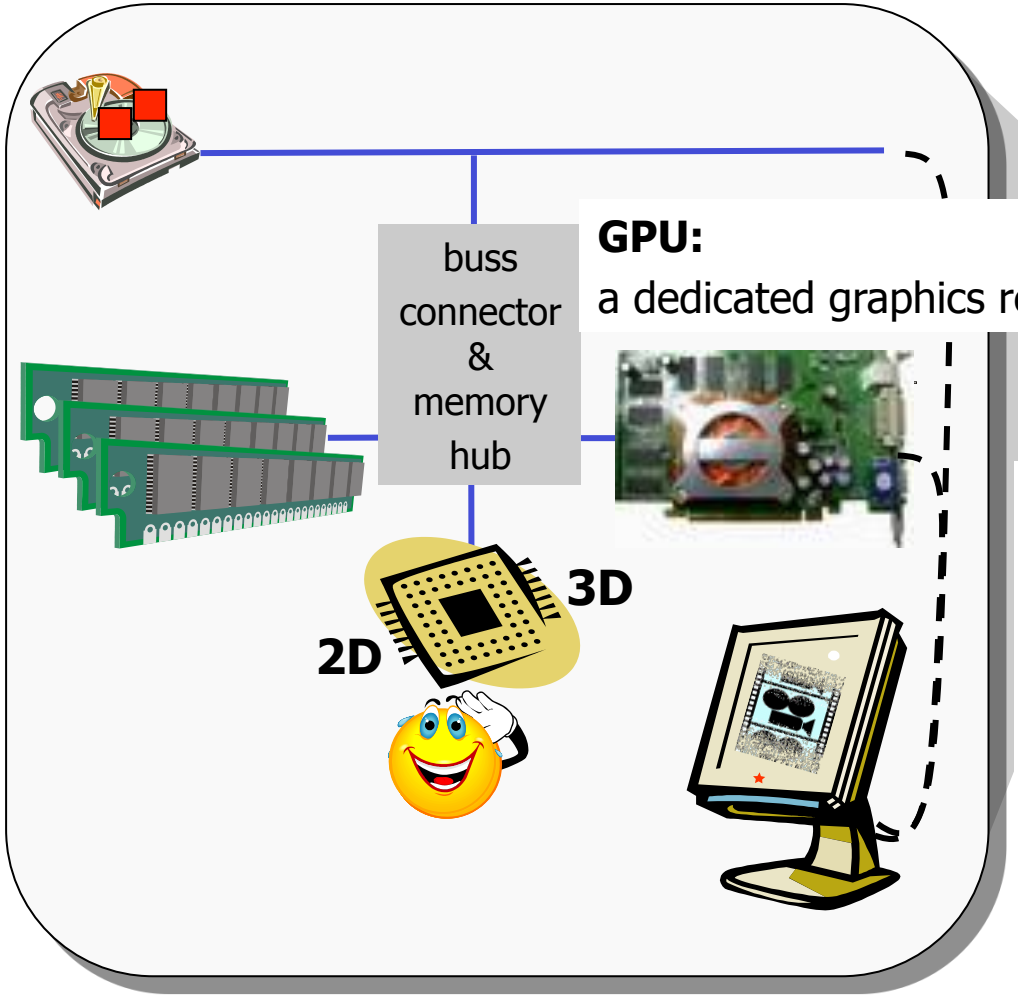
IXP1200



IXP2400 Architecture



Graphics Processing Units (GPUs)



GPU:
a dedicated graphics rendering device

New powerful GPUs, e.g.,:

- ✓ *Nvidia GeForce GX280*
  - ✓ 240 1476 MHz core
  - ✓ 1 GB memory
  - ✓ memory BW: 159 GB/sec
  - ✓ PCI Express 2.0
- ✓ similar to other manufacturers …

General Purpose Computing on GPU

- The
  - high arithmetic precision
  - extreme parallel nature
  - optimized, special-purpose instructions
  - available resources
  - …

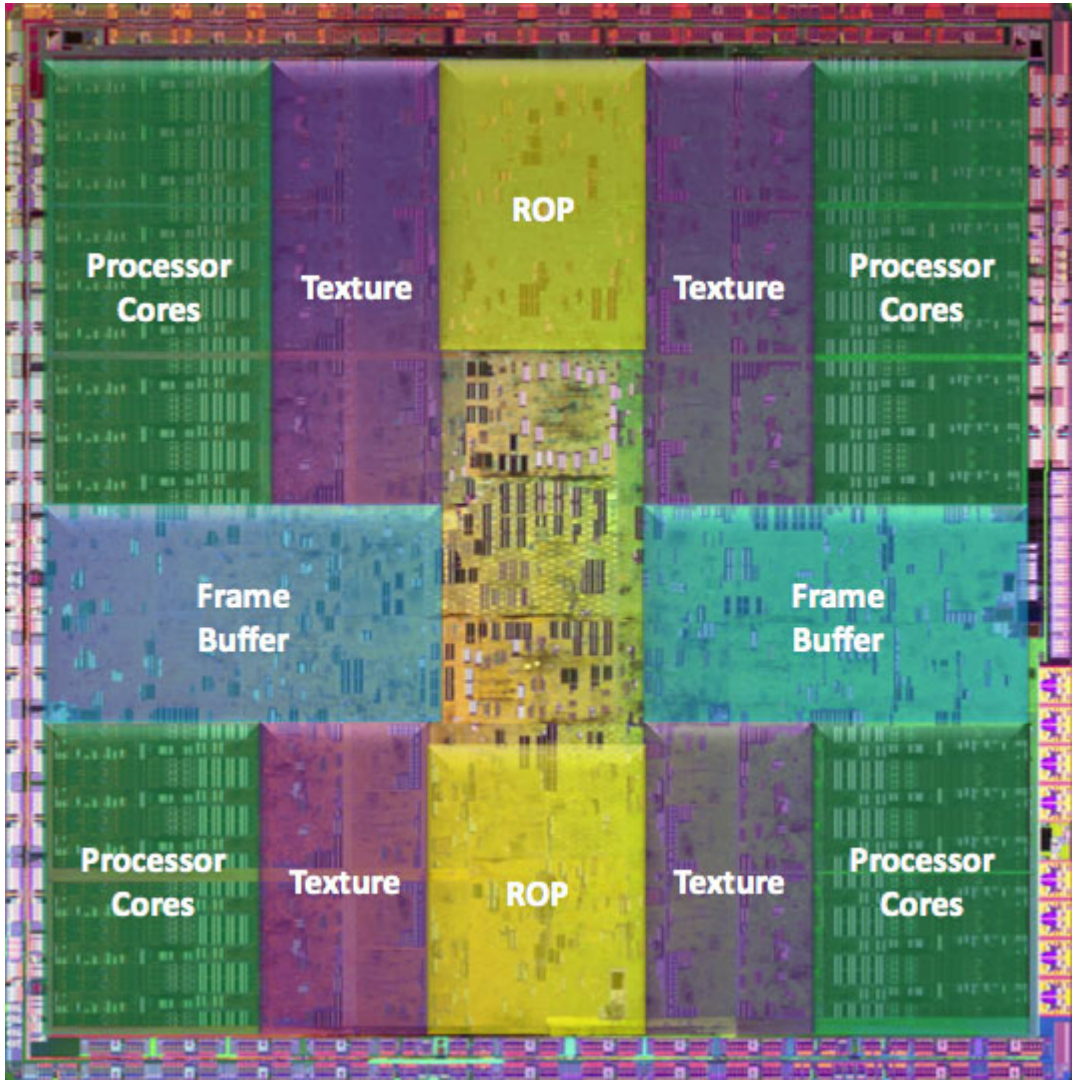
  … of the GPU allows for general, non-graphics related operations to be performed on the GPU

- Generic computing workload is off-loaded from CPU and to GPU

⇒ **More generically:**
Heterogeneous multi-core processing

nVIDIA G200 / GF100

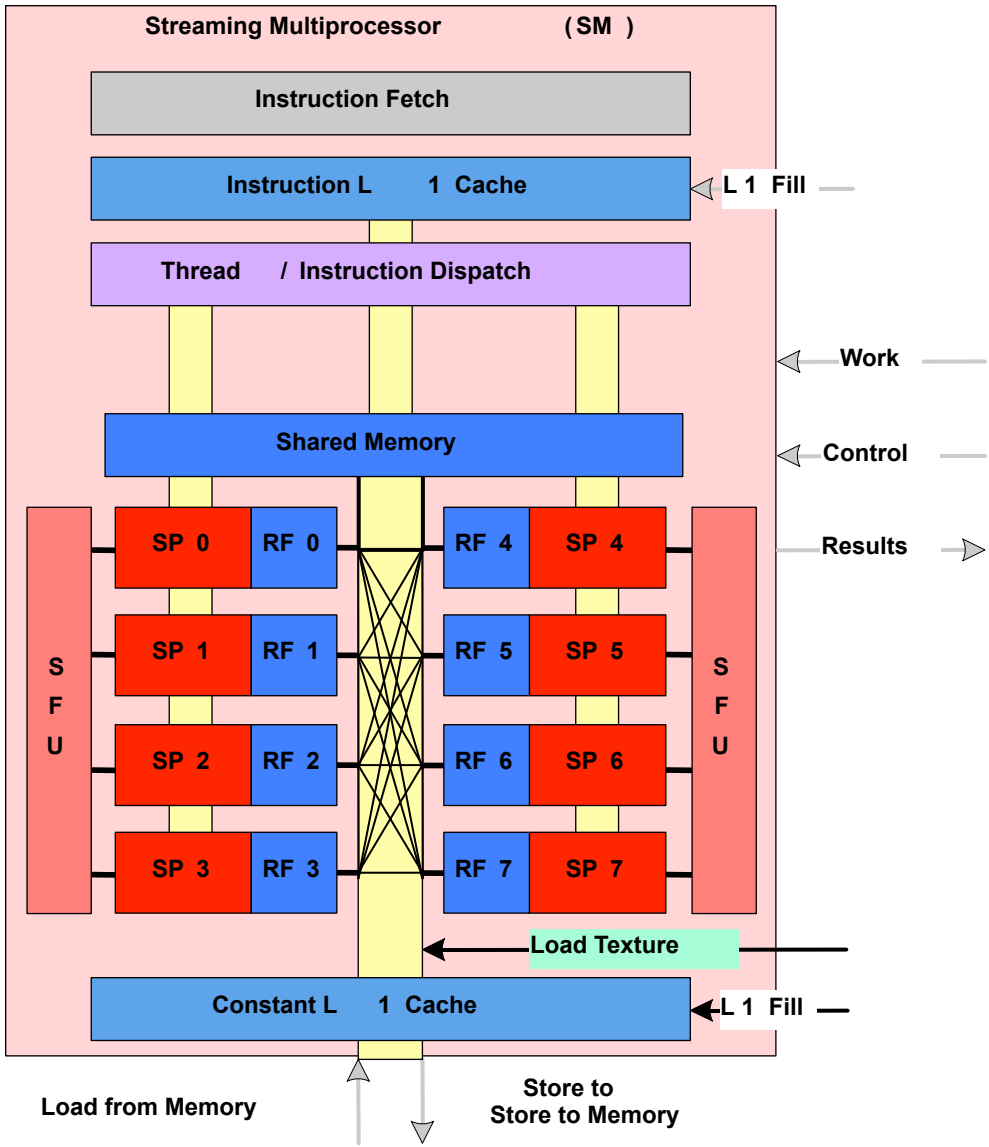
- 1.4 / 3 billion transistors
- 240 / 512 shaders
- 512 / 384 bit memory bus (GDDR3 / 5)
- 159 / 177 GB/sec memory bandwidth
- 933 / 1344 Gflops
- PCI Express 2.0



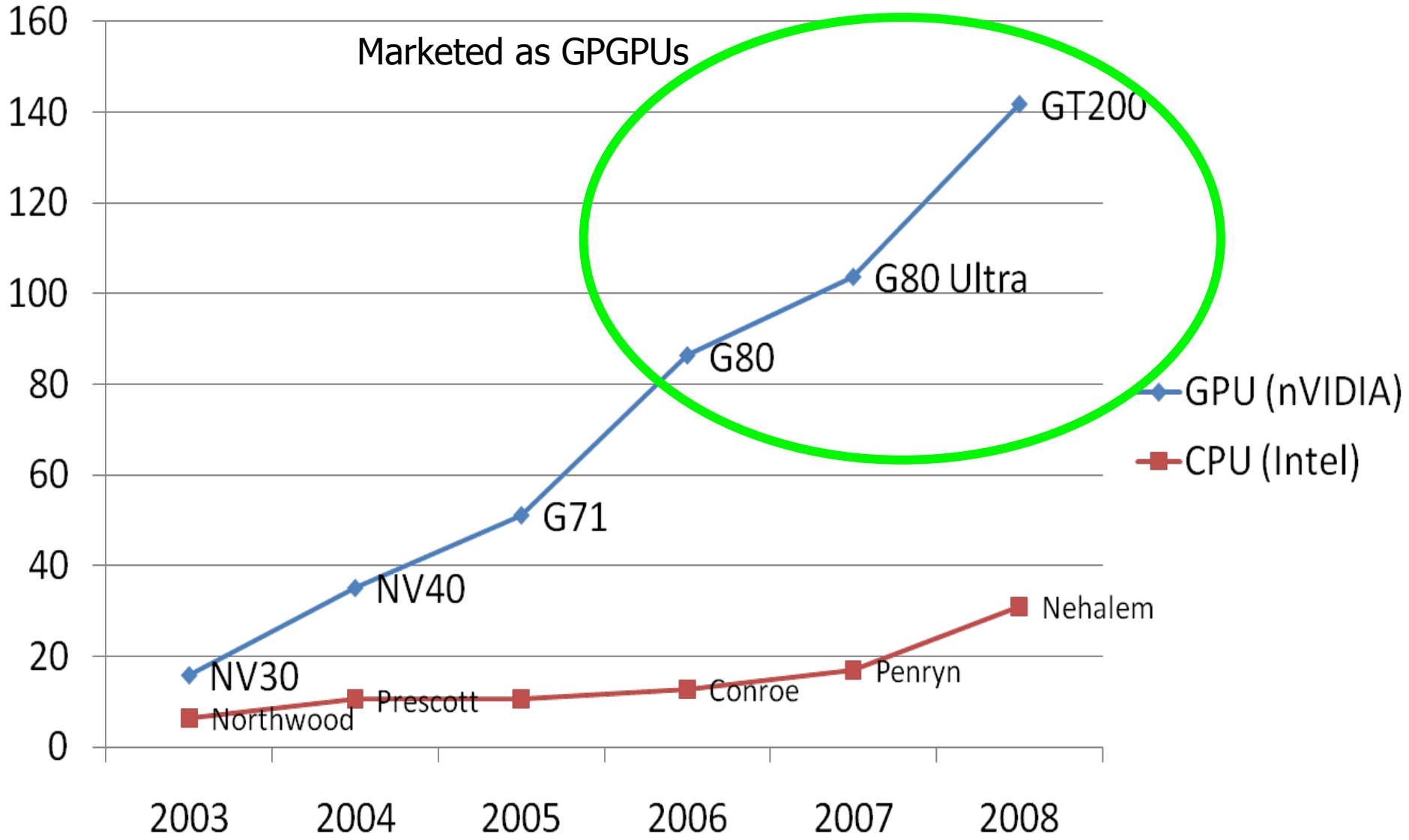
nVIDIA GT200

- Stream Multiprocessors (SMs)
  - fundamental thread block unit
  - 8 stream processors (SPs) (scalar ALU for threads)
  - 2 super function units (SFUs) (cos, sin, log, ...)
  - 8 32KB local register files (RFs)
  - 16 kB level 1 cache
  - 64 kB shared memory
  - 256 kB global level 2 cache

- Number of stream multiprocessors
  - 1 - Quadro NVS 130M
  - 16 - GeForce 8800 GTX
  - 30 - GeForce GTX 285
  - 4x30 - Tesla S1070

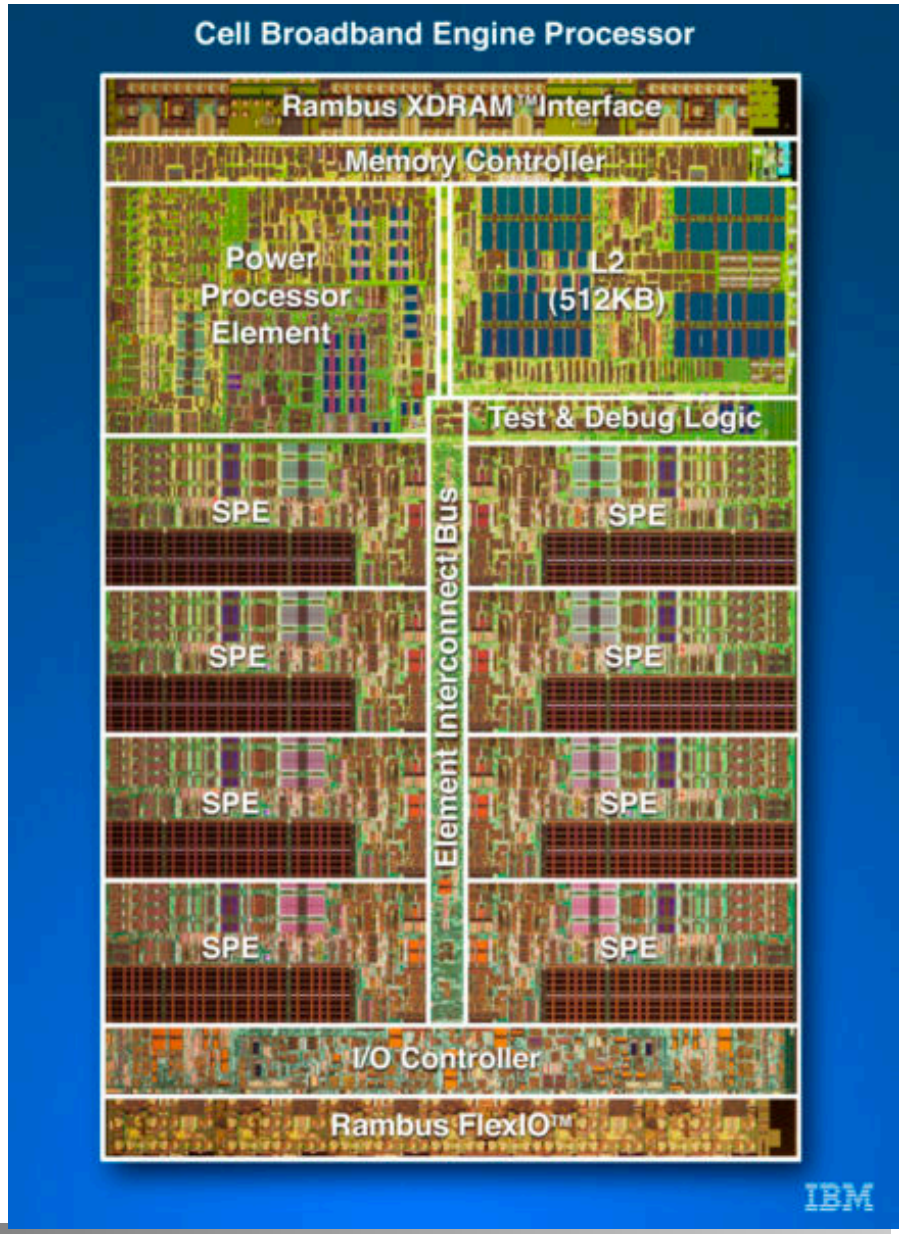


Memory Bandwidth for CPU and GPU



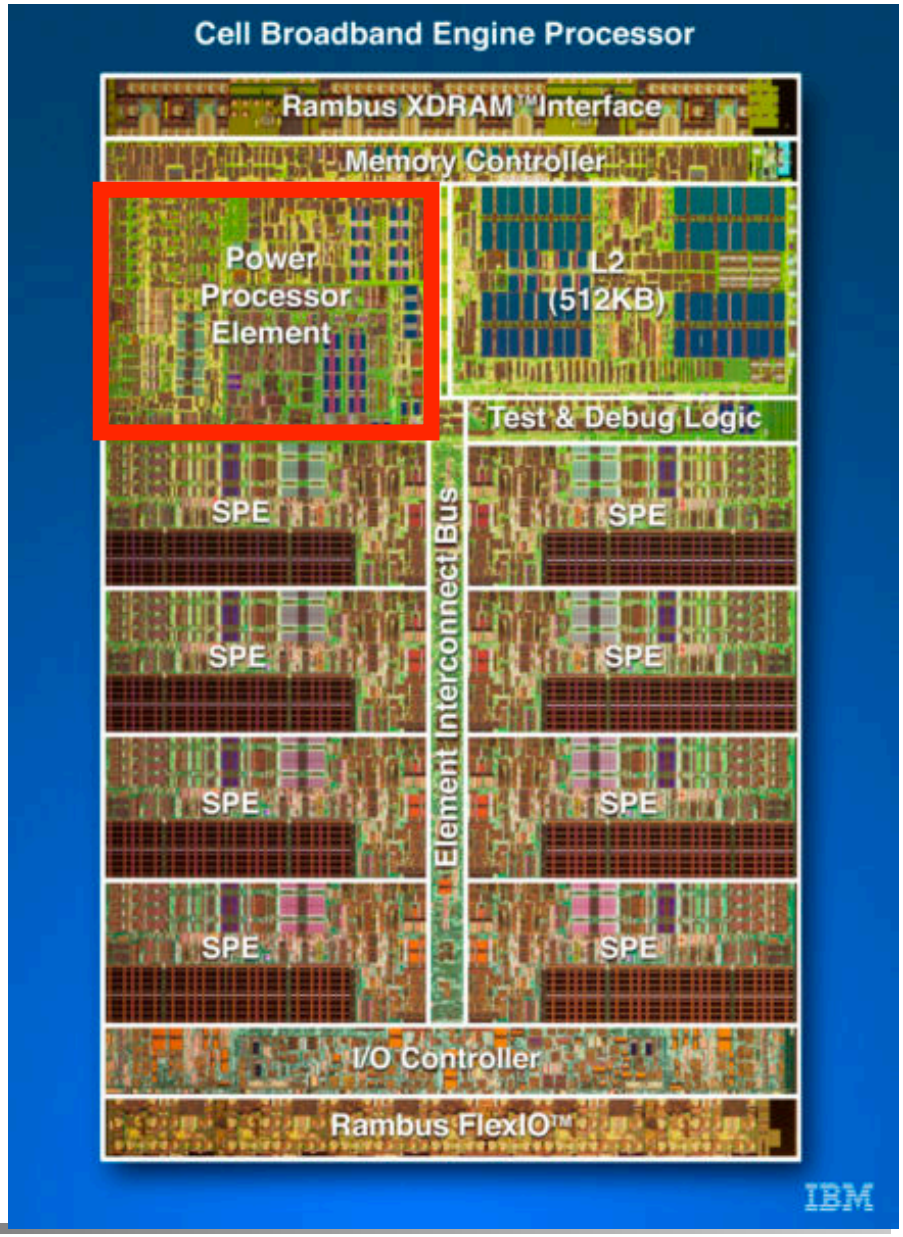
STI (Sony, Toshiba, IBM) Cell

- Motivation for the Cell
  - Cheap processor
  - Energy efficient
  - For games and media processing
  - Short time-to-market
- Conclusion
  - Use a multi-core chip
  - Design around an existing, power-efficient design
  - Add simple cores specific for game and media processing requirements



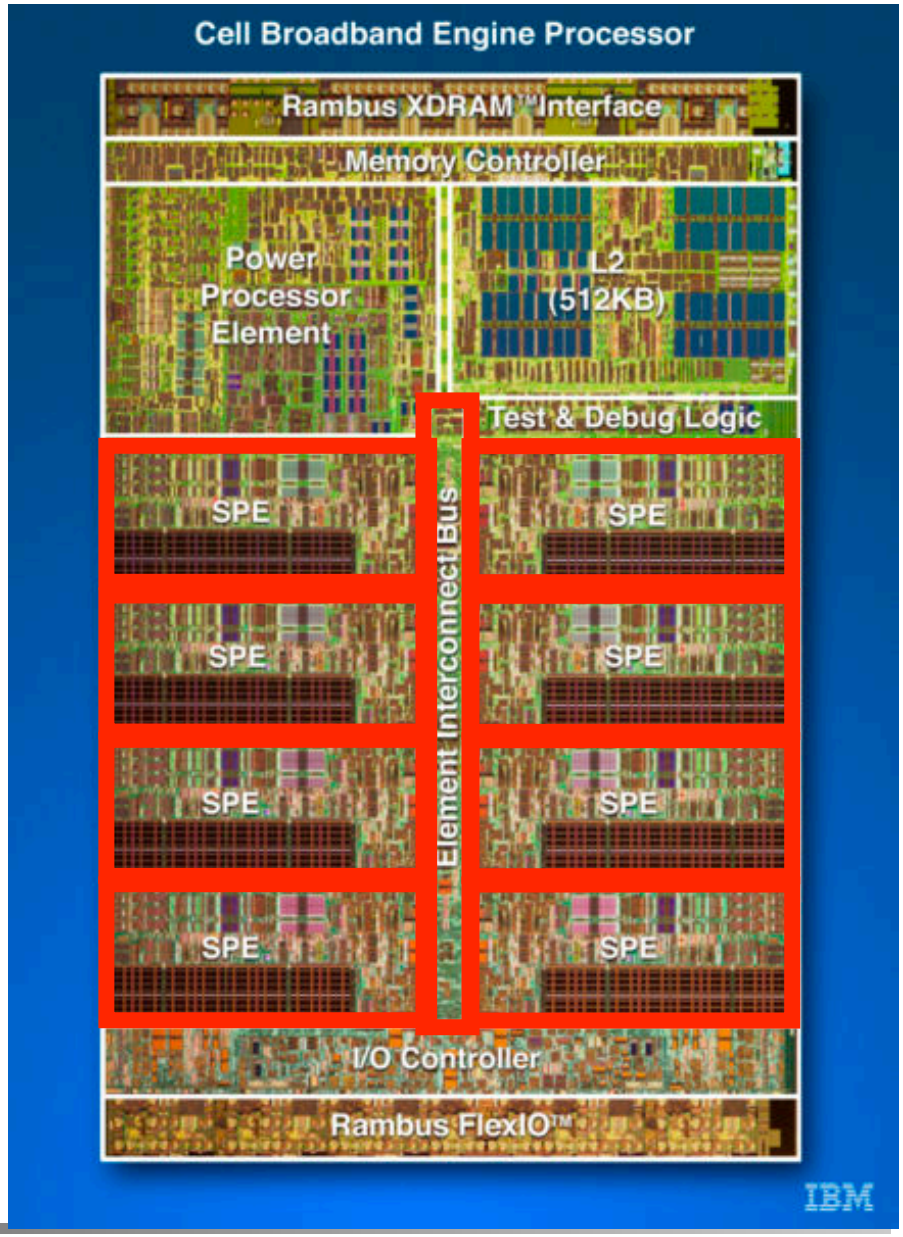
STI (Sony, Toshiba, IBM) Cell

- Cell is a 9-core processor
  - combining a light-weight general-purpose processor with multiple co-processors into a coordinated whole
  - *Power Processing Element* (PPE)
    - conventional Power processor
    - not supposed to perform all operations itself, acting like a controller
    - running conventional OSes
    - 16 KB instruction/data level 1 cache
    - 512 KB level 2 cache



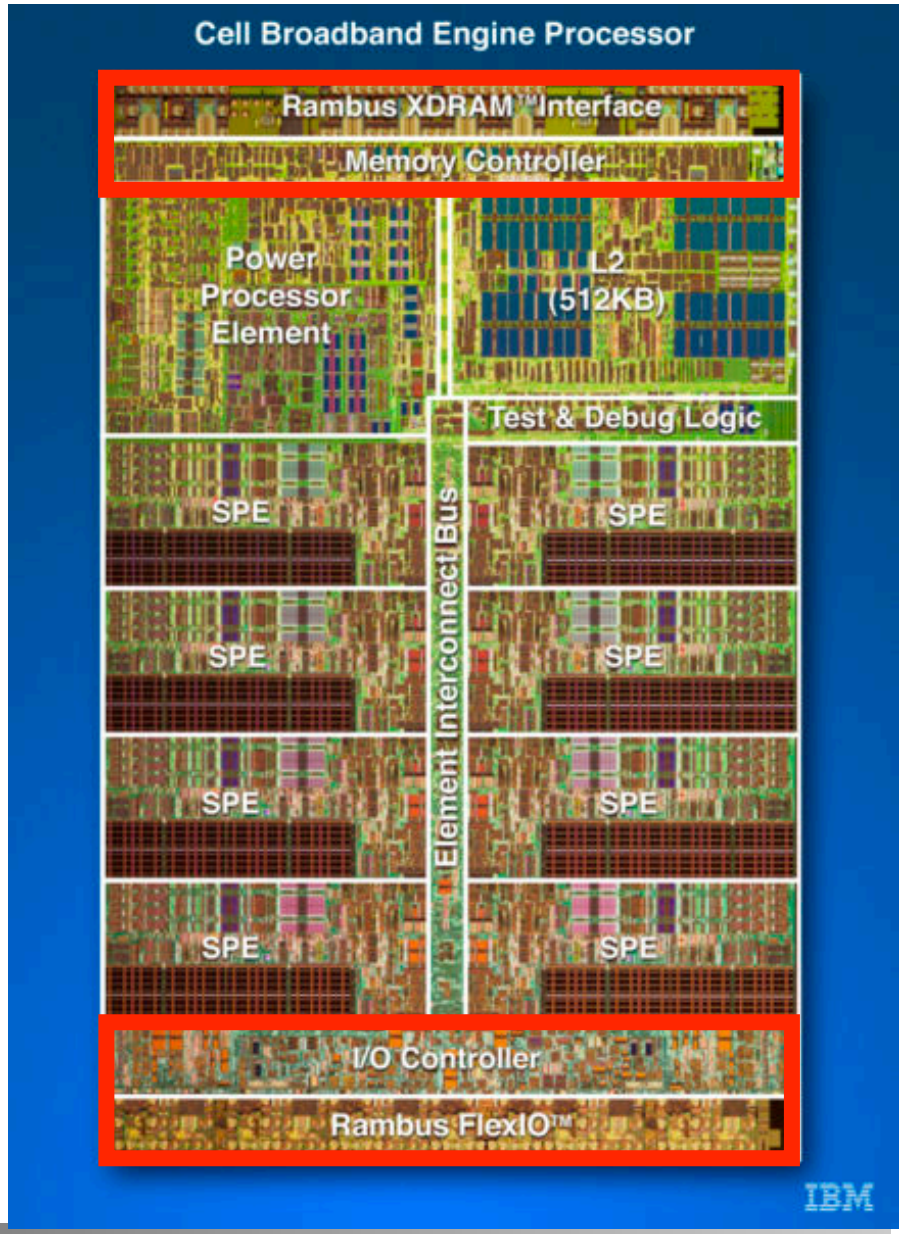
STI (Sony, Toshiba, IBM) Cell

- *Synergistic Processing Elements* (SPE)
  - specialized co-processors for specific types of code, i.e., very high performance vector processors
  - local stores
  - can do general purpose operations
  - the PPE can start, stop, interrupt and schedule processes running on an SPE
- Element Interconnect Bus (EIB)
  - internal communication bus
  - connects on-chip system elements:
    - PPE & SPEs
    - the memory controller (MIC)
    - two off-chip I/O interfaces
  - 25.6 GBps each way



STI (Sony, Toshiba, IBM) Cell

- memory controller
  - Rambus XDRAM interface to Rambus XDR memory
  - dual channels at 12.8 GBps → 25.6 GBps
- I/O controller
  - Rambus FlexIO interface which can be clocked independently
  - dual configurable channels
  - maximum ~ 76.8 GBps



STI (Sony, Toshiba, IBM) Cell

- Cell has in essence traded running everything at moderate speed for the ability to run certain types of code at high speed

- used for example in
  - Sony PlayStation 3:
    - 3.2 GHz clock
    - 6 SPEs for general operations
    - 1 SPE for security for the OS
  - Toshiba home cinema:
    - decoding of 48 HDTV MPEG streams
      → dozens of thumbnail videos simultaneously on screen
  - IBM blade centers:
    - 3.2 GHz clock
    - Linux ≥ 2.6.11

The End: Summary

- Heterogeneous multi-core processors are already everywhere

⇨**Challenge: programming**

- **Need to know the capabilities of the system**
- **Different abilities in different cores**
- **Memory bandwidth**
- **Memory sharing efficiency**
- **Need new methods to program the different components**

- Next time: how to start programming the **Intel IXP**