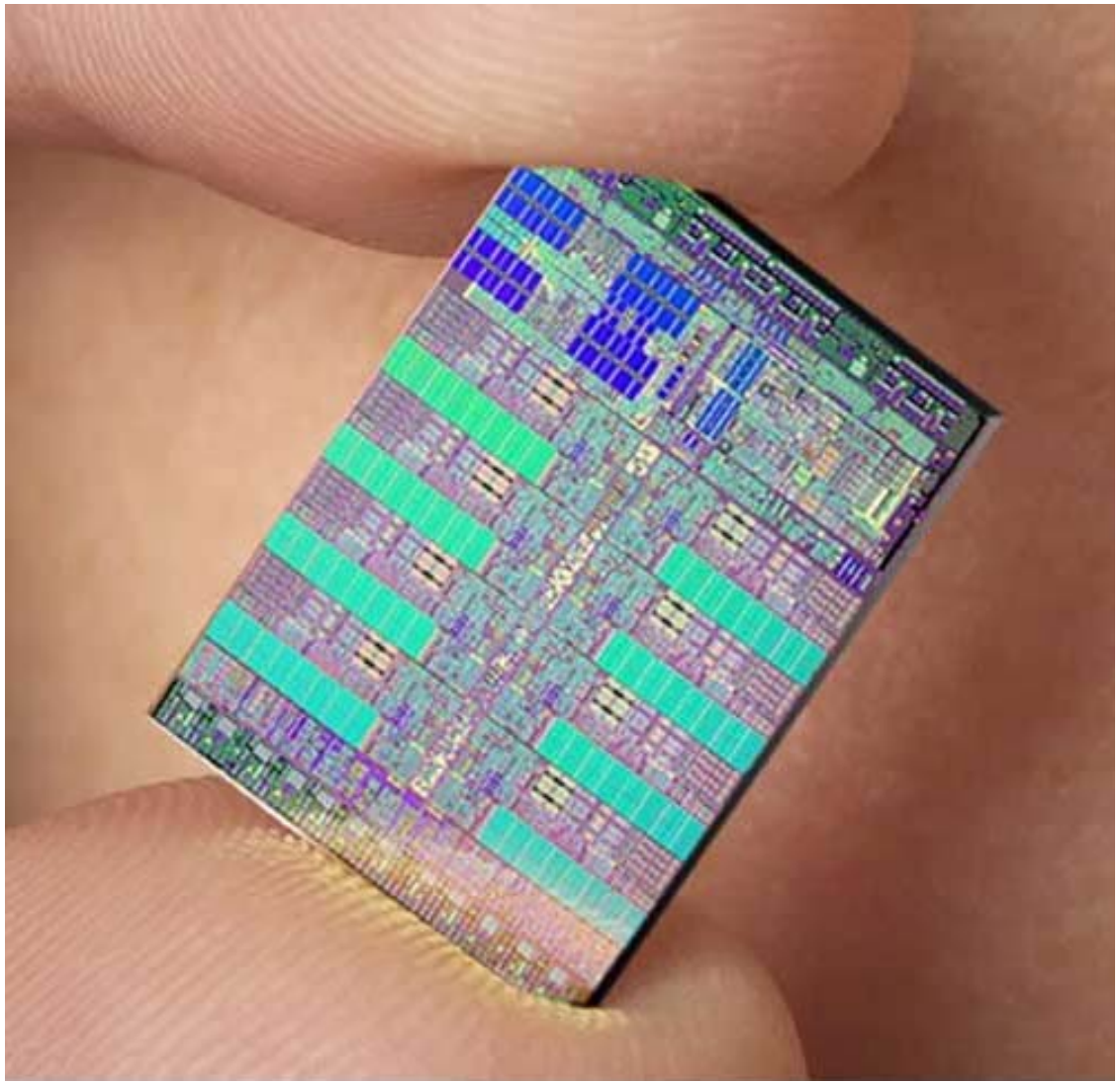
EE382V: Principles in Computer Architecture
Parallelism and Locality
Fall 2008

Lecture 20 – Sony (/Toshiba/IBM) Cell Broadband Engine

Mattan Erez

The University of Texas at Austin

Cell Broadband Engine



SONY
IBM
TOSHIBA

Outline

- Motivation
- Cell architecture
  - GPP Controller (PPE)
  - Compute PEs (SPEs)
  - Interconnect (EIB)
  - Memory and I/O
- Comparisons
  - Stream Processors
- Software (probably next time)

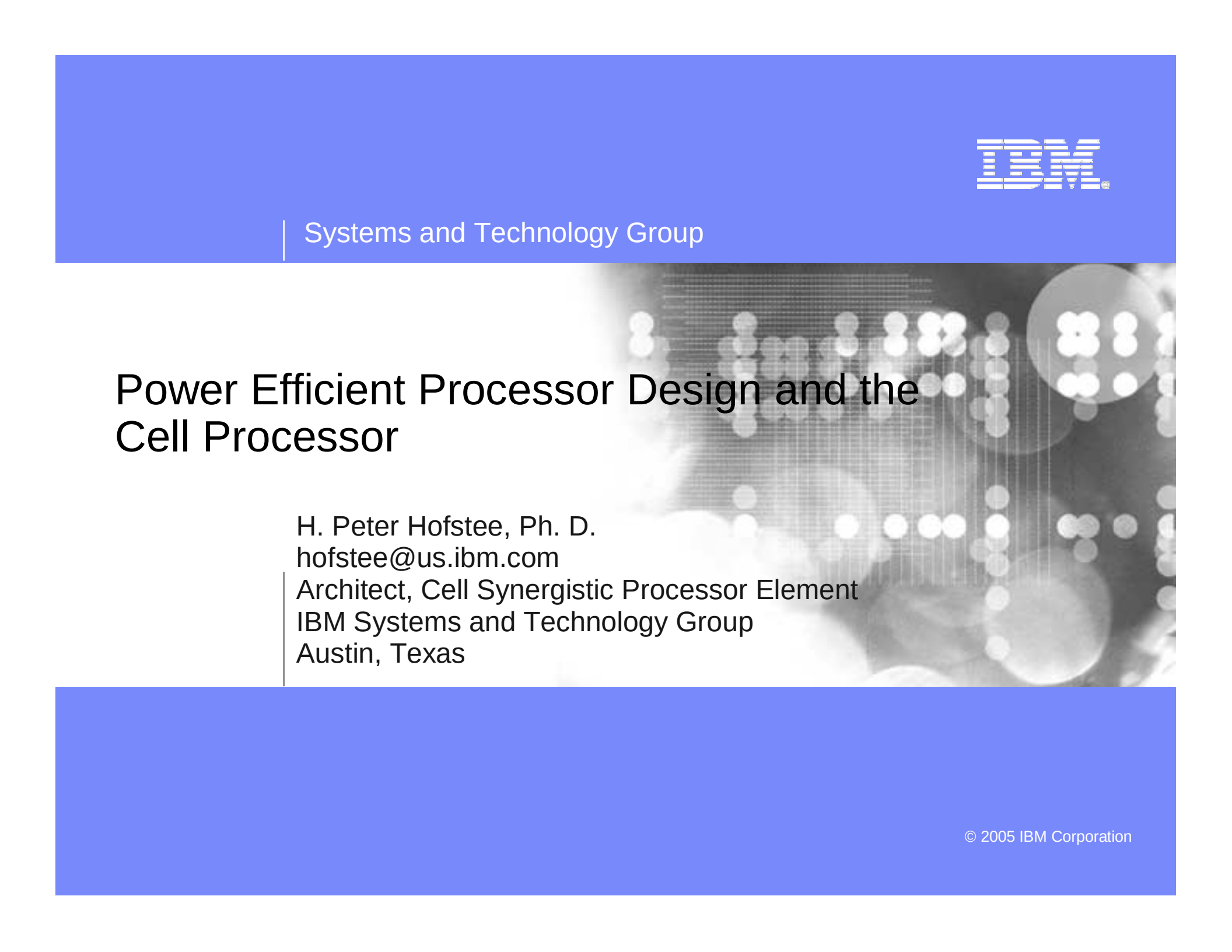
- All Cell related images and figures © Sony and IBM
- Cell Broadband Engine ™ Sony Corp.

Cell Motivation – Part I

- Performance demanding applications have different characterisitics
  - Parallelism
  - Locality
  - Realtime
- Games, graphics, multimedia …
- Requires redesign of HW and SW to provide efficient high performance
  - Power, memory, frequency walls
- Cell designed specifically for these applications
  - Requirements set by Sony and Toshiba
  - Main design and architecture at IBM

Move to IBM Slides

- Rest of motivation and architecture slides taken directly from talks by Peter Hofstee, IBM
  - Separate PDF file combined from:
    - http://www.hpcaconf.org/hpca11/slides/Cell_Public_Hofstee.pdf
    - http://www.cct.lsu.edu/~estrabd/LACSI2006/workshops/workshop3/Slides/01_Hofstee_Cell.pdf



Systems and Technology Group

Power Efficient Processor Design and the Cell Processor

H. Peter Hofstee, Ph. D.
hofstee@us.ibm.com
Architect, Cell Synergistic Processor Element
IBM Systems and Technology Group
Austin, Texas

Agenda

- **Power Efficient Processor Architecture**
- **System Trends**
- **Cell Processor Overview**

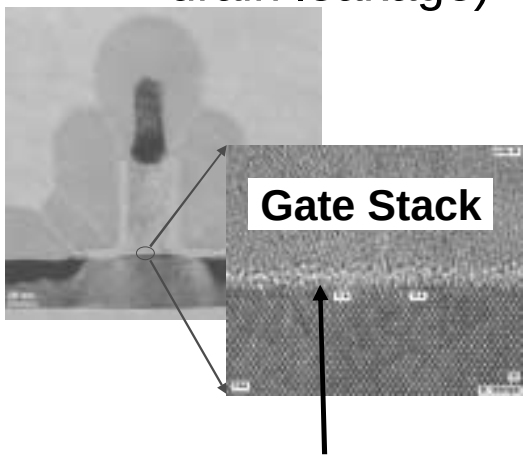
Power Efficient Architecture

Limiters to Processor Performance

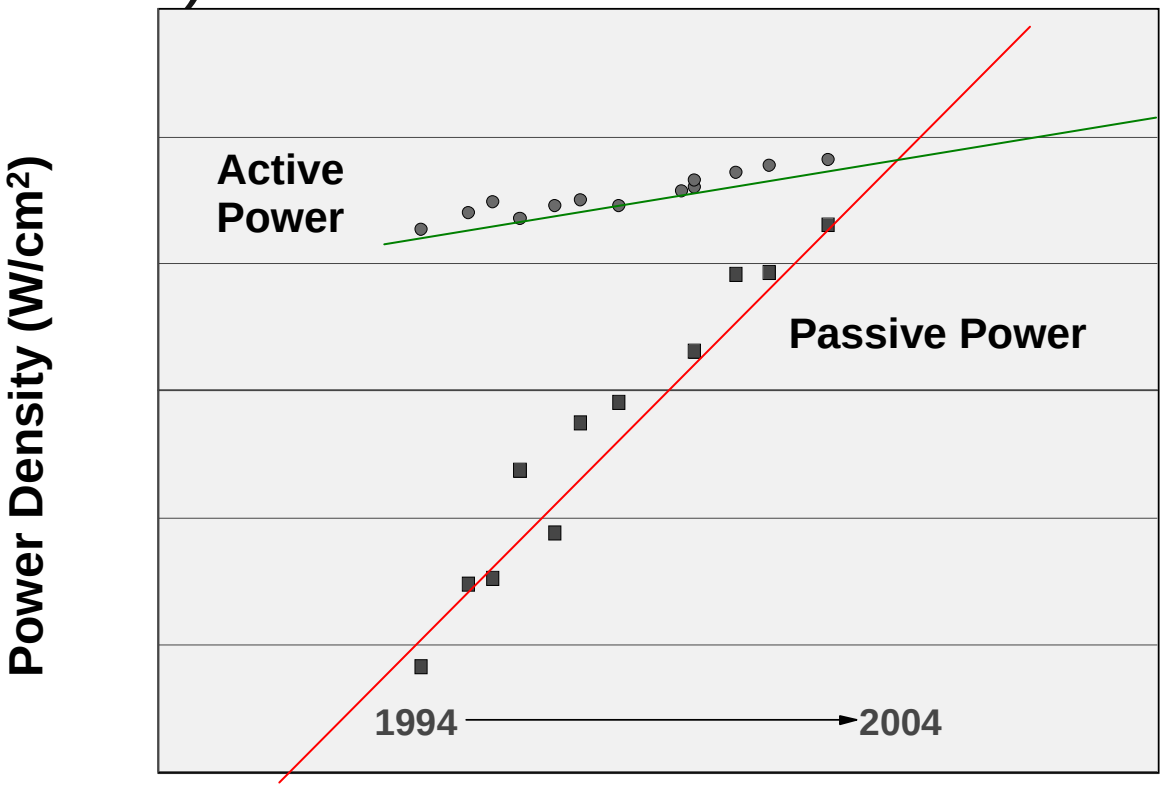
- **Power wall**
- **Memory wall**
- **Frequency wall**

Power Wall (Voltage Wall)

- **Power components:**
  - Active power
  - Passive power
    - Gate leakage
    - Sub-threshold leakage (source-drain leakage)



Gate dielectric approaching a fundamental limit (a few atomic layers)



NET: INCREASING PERFORMANCE

REQUIRES INCREASING EFFICIENCY

Memory wall

- **Main memory now nearly 1000 cycles from the processor**
  - Situation worse with (on-chip) SMP
- **Memory latency penalties drive inefficiency in the design**
  - Expensive and sophisticated hardware to try and deal with it
  - Programmers that try to gain control of cache content, but are hindered by the hardware mechanisms
- **Latency induced bandwidth limitations**
  - Much of the bandwidth to memory in systems can only be used speculatively
  - Diminishing returns from added bandwidth on traditional systems

Microprocessor Efficiency

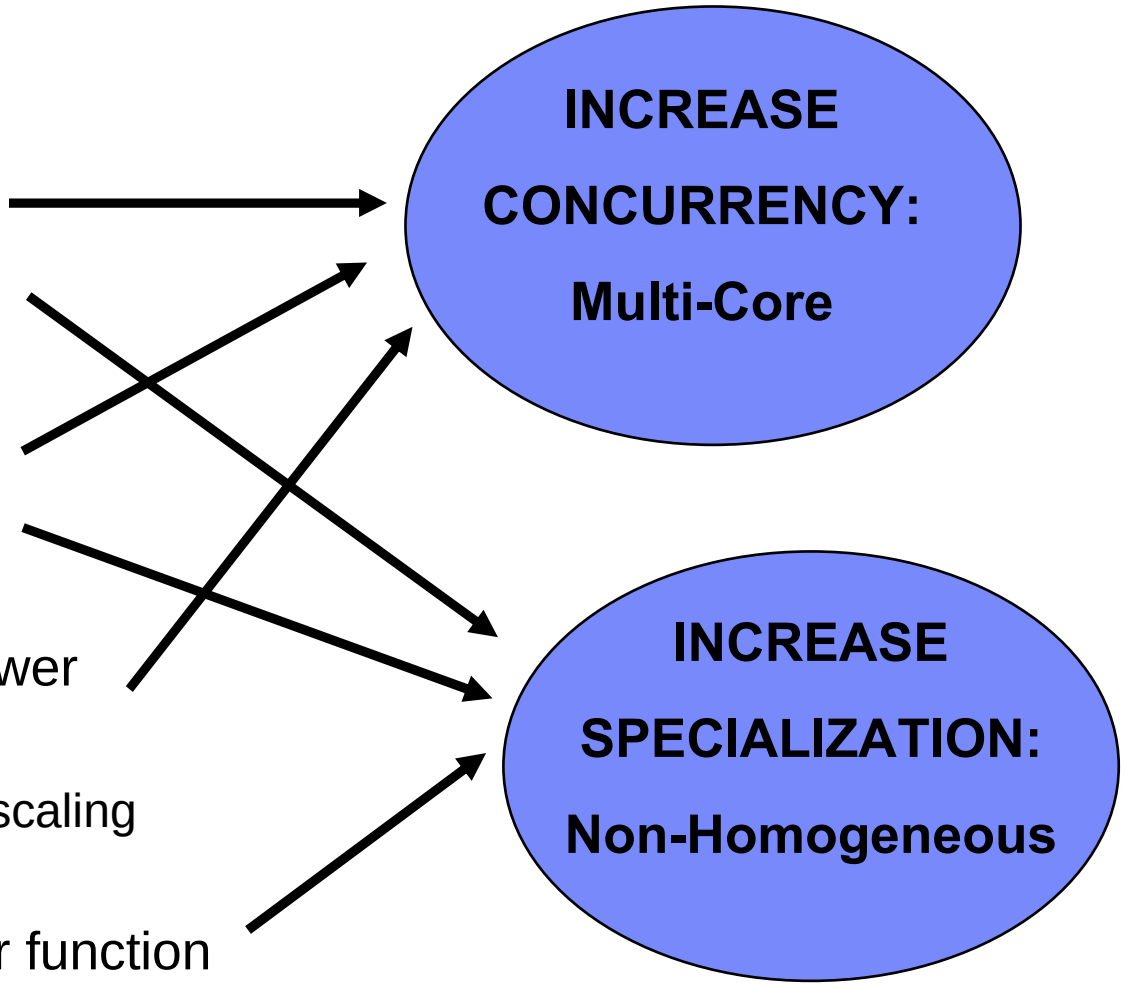
- **Recent History:**
  - Gelsinger's law
    - 1.4x more performance for 2x more transistors
  - Hofstee's corollary
    - 1/1.4x efficiency loss in every generation
    - Examples: Cache size, OoO, Superscalar, etc. etc.
- **Re-examine microarchitecture with performance per transistor as metric**
  - Pipelining is last clear win

Attacking the Performance Walls

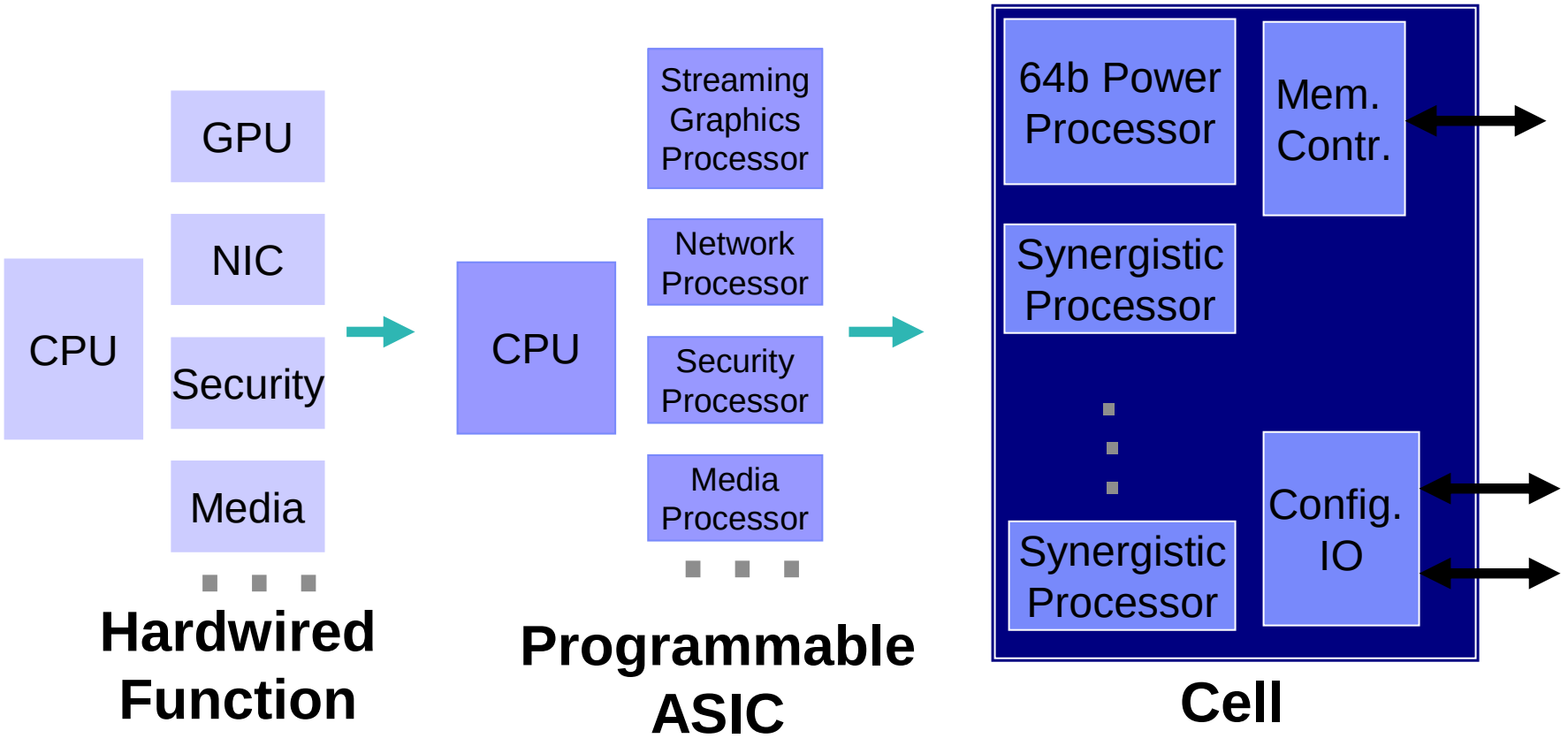
- **Multi-Core Non-Homogeneous Architecture**
  - Control Plane vs. Data Plane processors
  - Attacks **Power Wall**
- **3-level Model of Memory**
  - Main Memory, Local Store, Registers
  - Attacks **Memory Wall**
- **Large Shared Register File & SW Controlled Branching**
  - Allows deeper pipelines (11FO4 … helps power!)
  - Attacks **Frequency Wall**

Solutions

- Memory wall:
  - More slower threads
  - Asynchronous loads
- Efficiency wall:
  - More slower threads
  - Specialized function
- Power wall:
  - Reduce transistor power
    - operating voltage
    - limit oxide thickness scaling
    - limit channel length
  - Reduce switching per function



Next Generation Processors address Programming Complexity and Trend Towards Programmable Offload Engines with a Simpler System Alternative

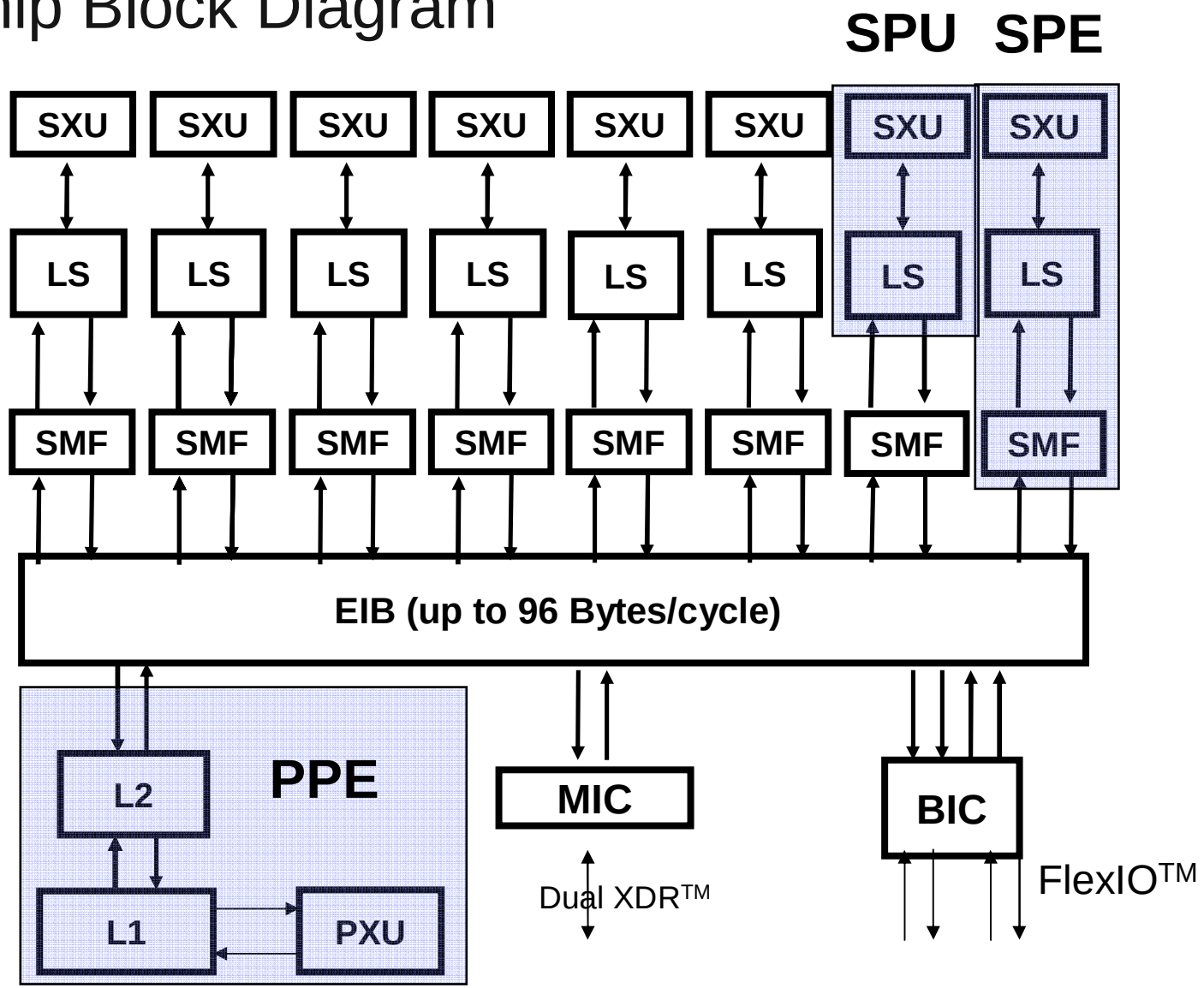


“Outward Facing” Aspects of Cell

- **Cell is designed to be responsive**
- **.. to human user**
  - Real-time response
  - Supports rich visual interfaces
- **.. to network**
  - Flexible, can support new standards
  - High-bandwidth
  - Content protection, privacy & security
- **Contrast to traditional processors which evolved from “batch processing” mentality (inward focused).**

Cell Overview

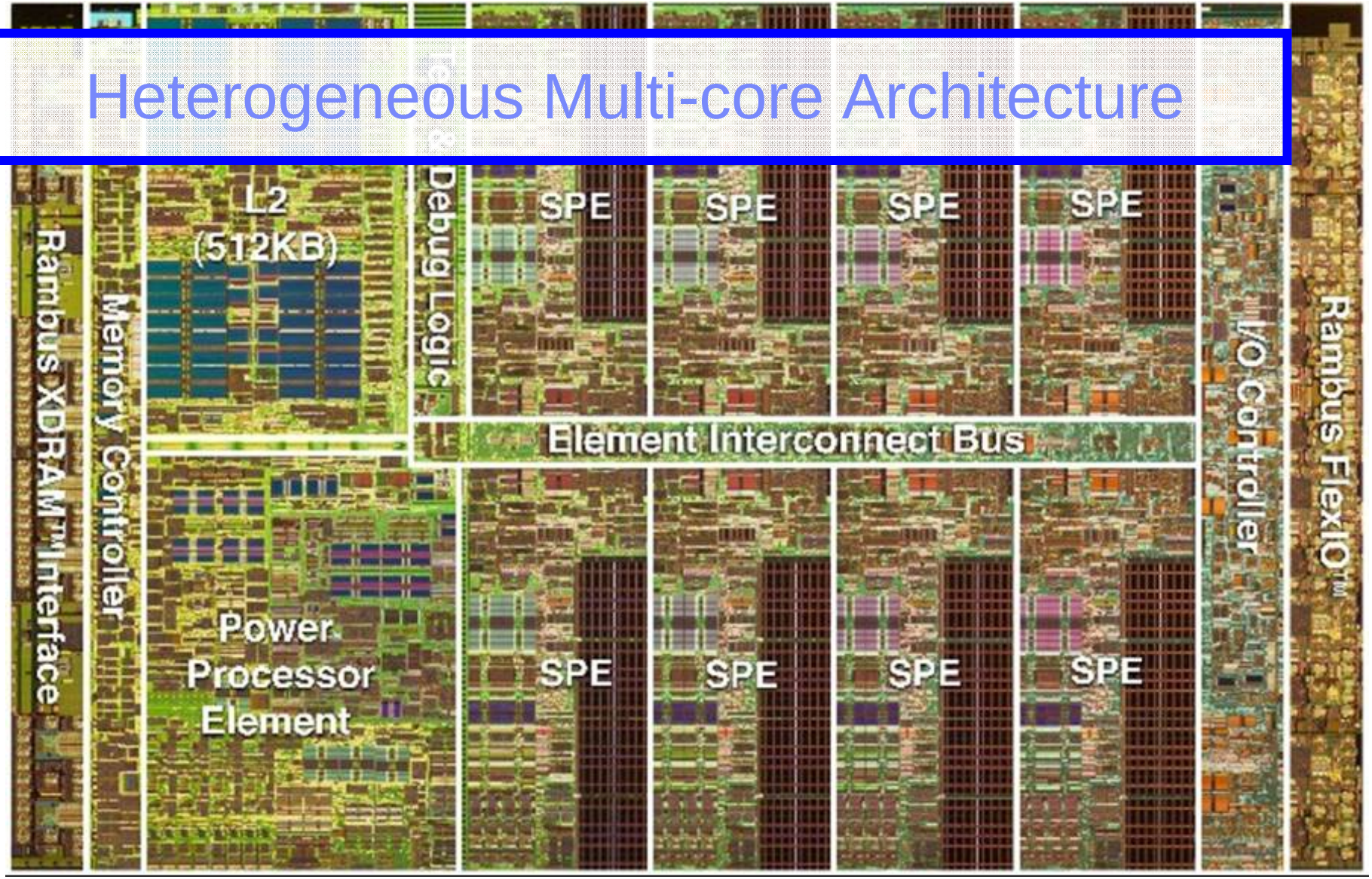
Cell Chip Block Diagram

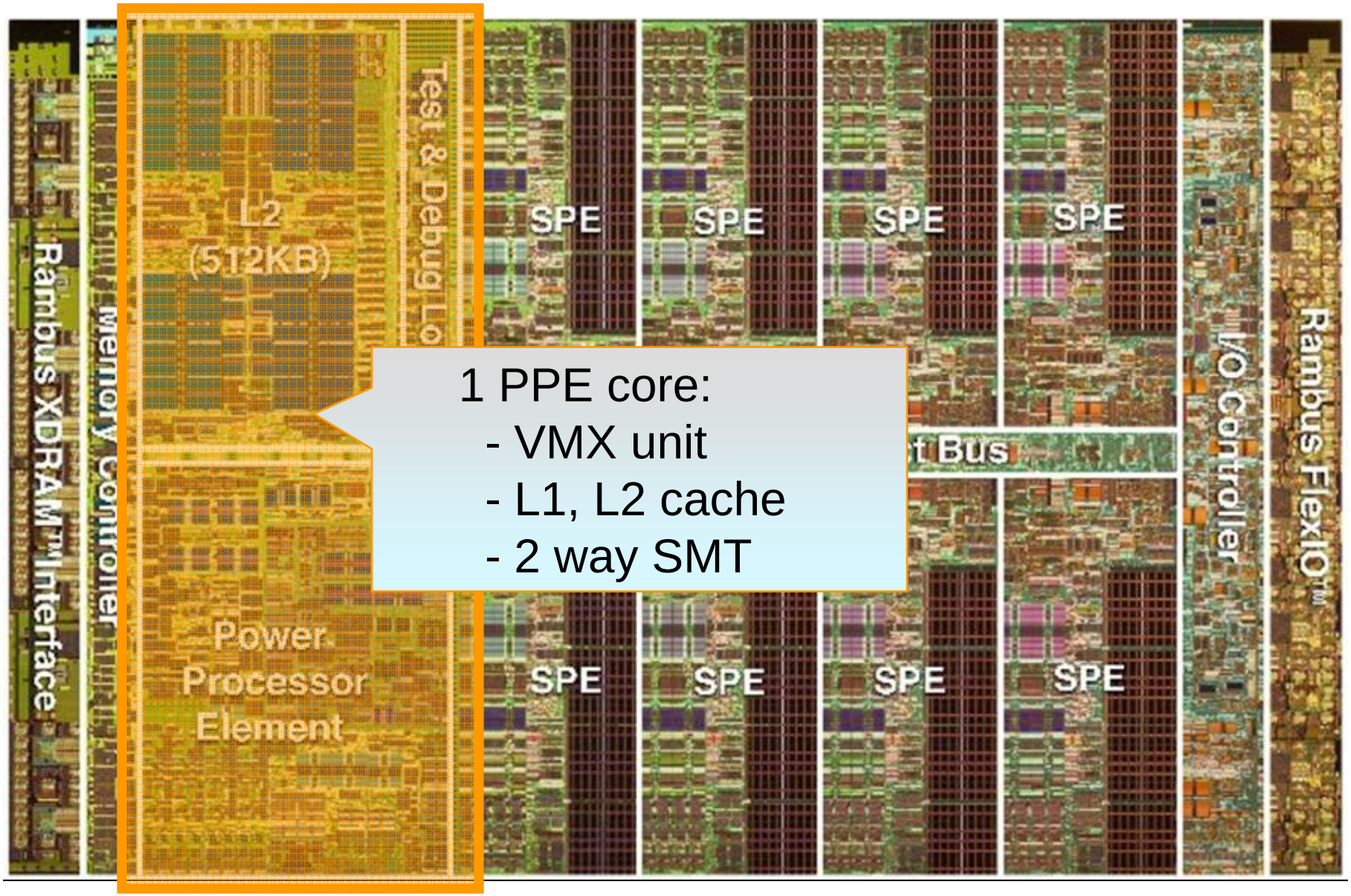


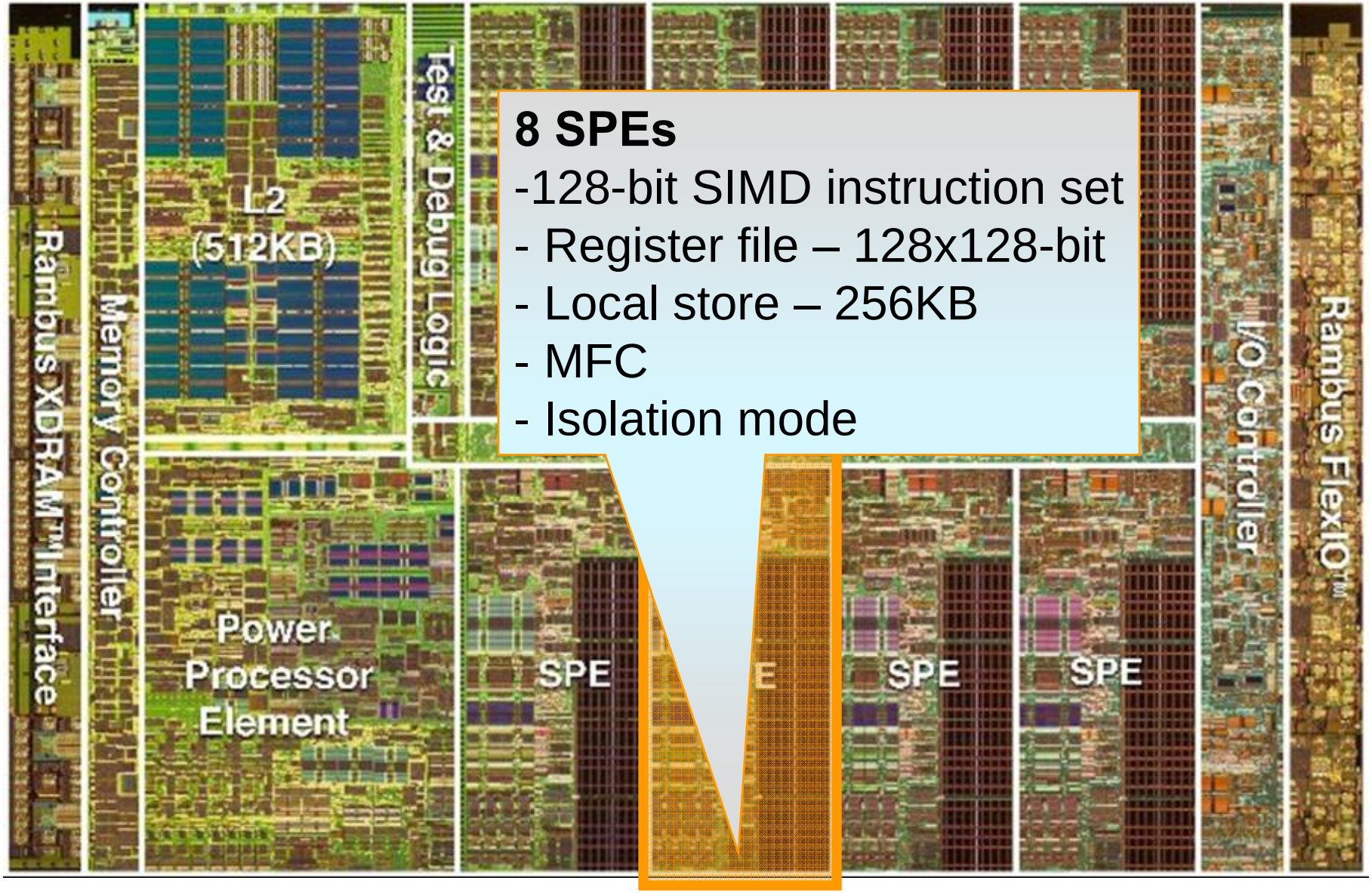
Cell Highlights

- Observed clock speed
  - **> 4 GHz**
- Peak performance (single precision)
  - **> 256 GFlops**
- Peak performance (double precision)
  - **>26 GFlops**
- Area 221 mm2
- Technology 90nm SOI
- Total # of transistors 234M

Heterogeneous Multi-core Architecture



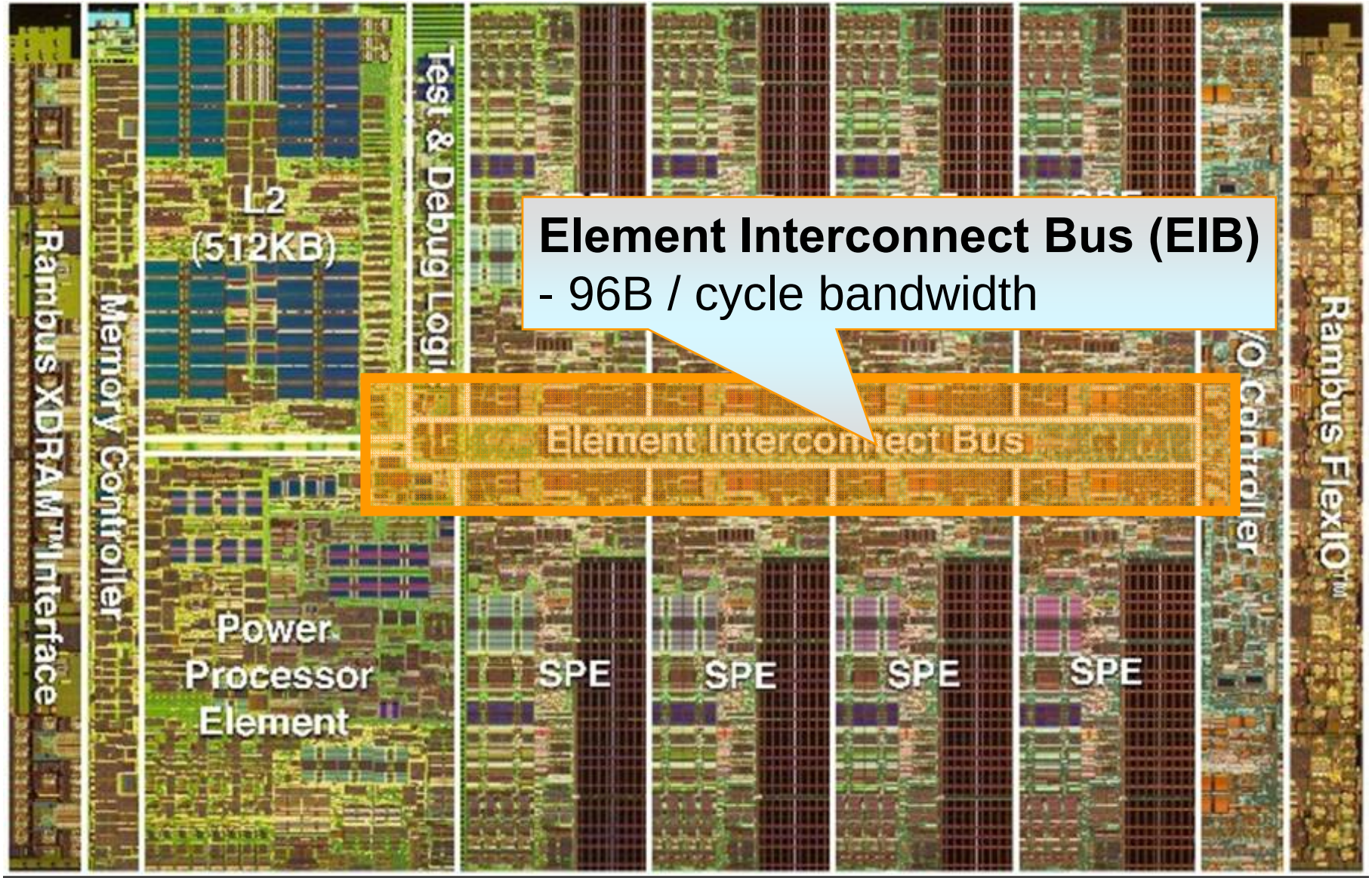


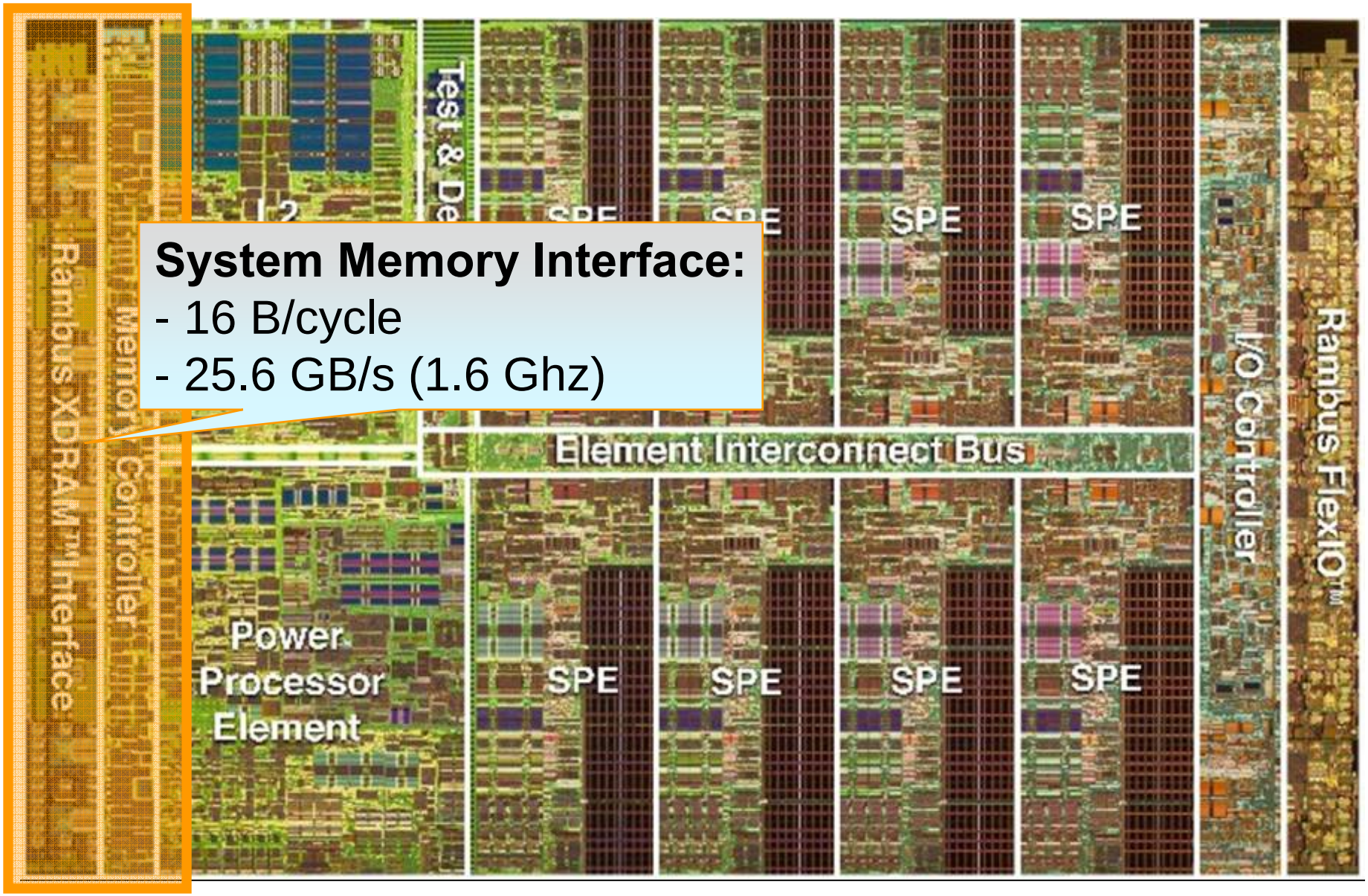


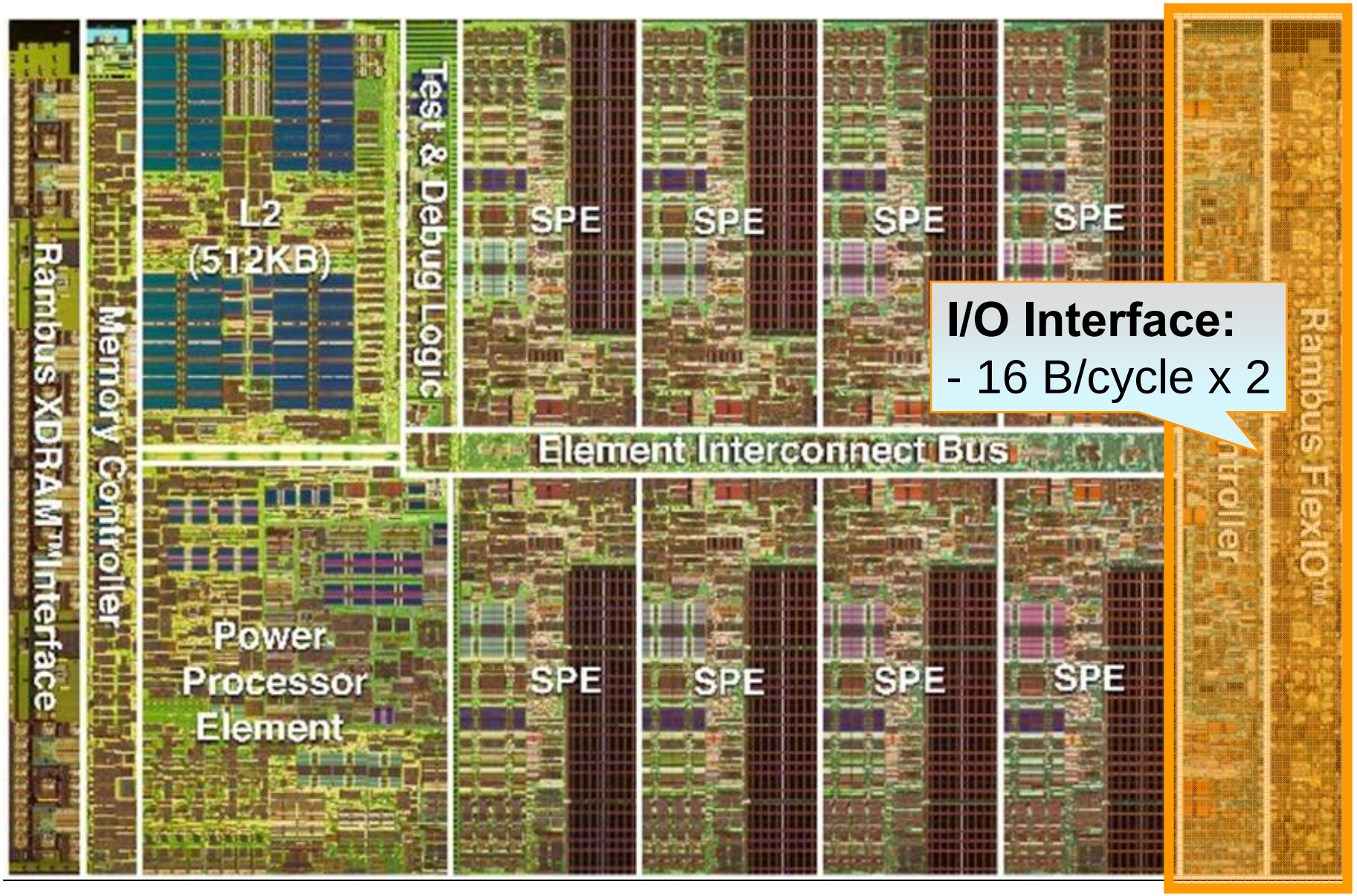
8 SPEs

-128-bit SIMD instruction set

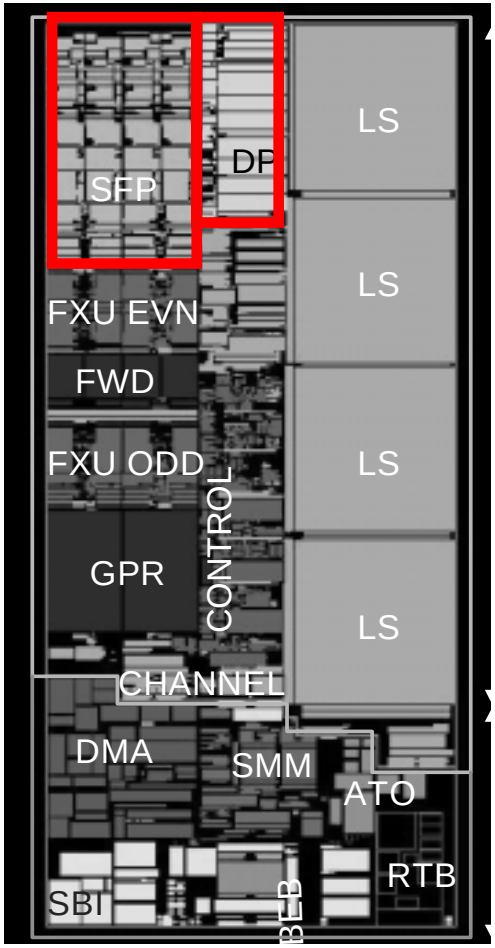
- Register file – 128x128-bit
- Local store – 256KB
- MFC
- Isolation mode







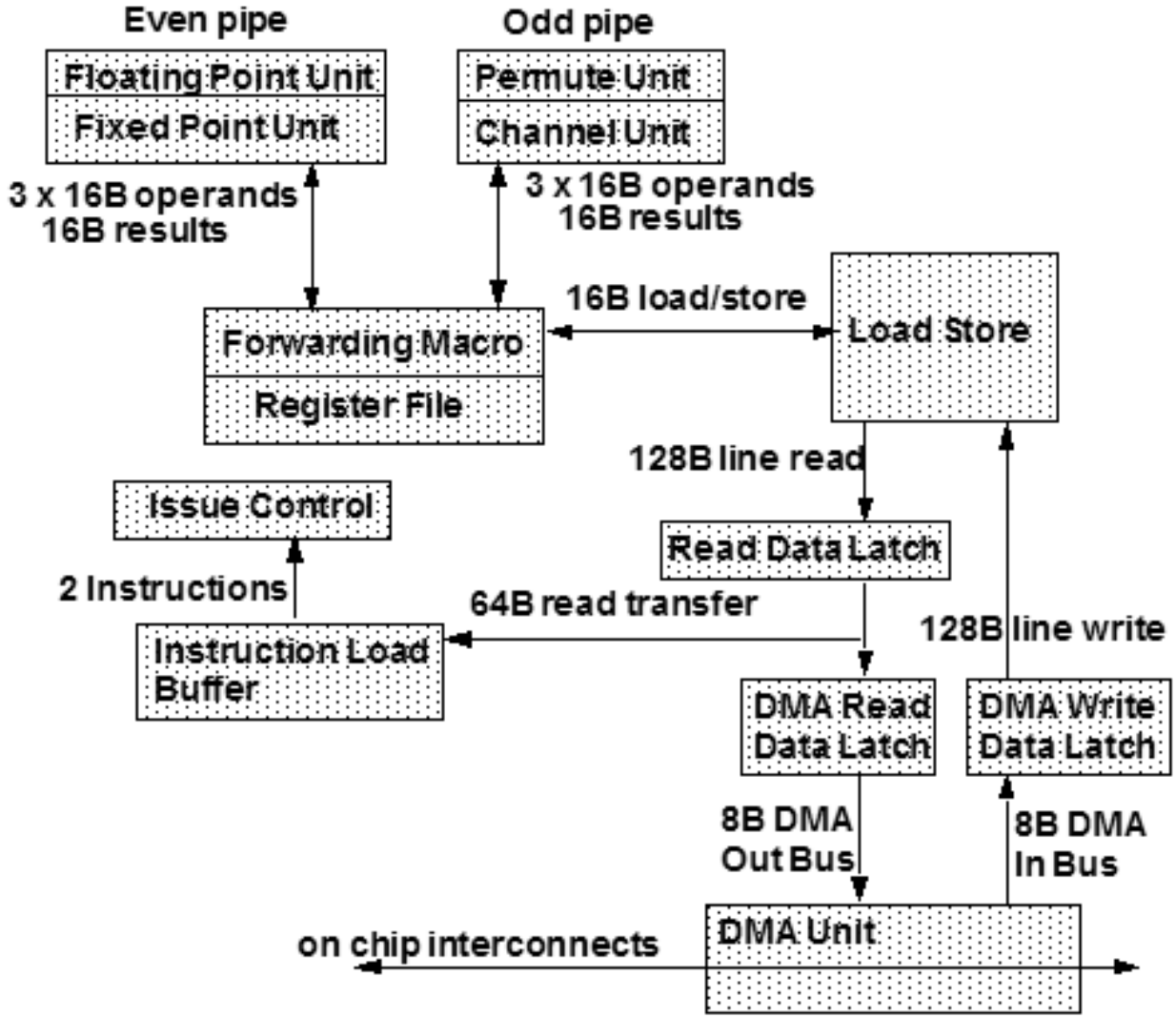
SPE Highlights



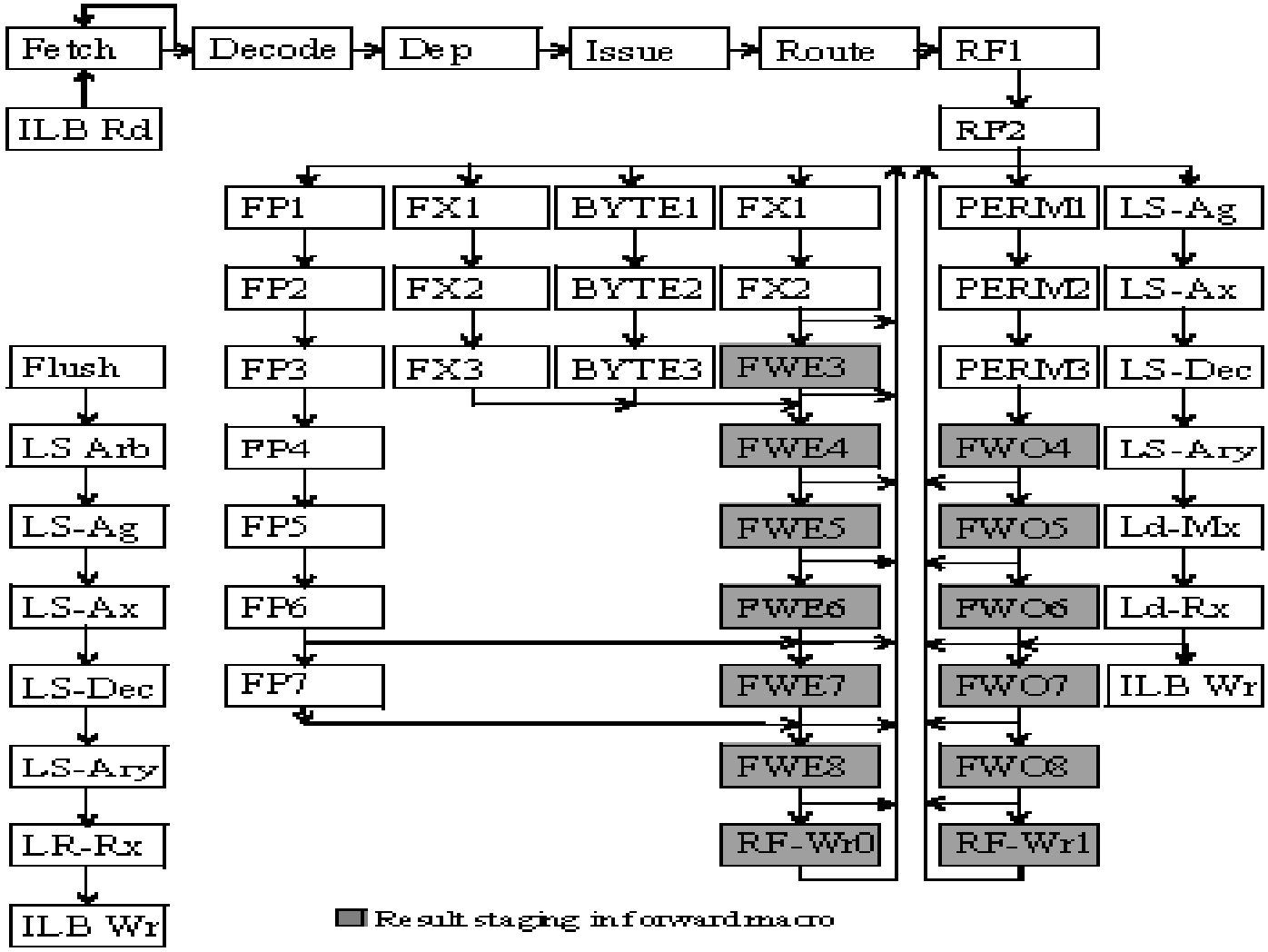
14.5mm2 (90nm SOI)

- **User-mode architecture**
  - No translation/protection within SPU
  - DMA is full Power Arch protect/x-late
- **Direct programmer control**
  - DMA/DMA-list
  - Branch hint
- **VMX-like SIMD dataflow**
  - Broad set of operations
  - Graphics SP-Float
  - IEEE DP-Float (BlueGene-like)
- **Unified register file**
  - 128 entry x 128 bit
- **256kB Local Store**
  - Combined I & D
  - 16B/cycle L/S bandwidth
  - 128B/cycle DMA bandwidth

SPE Organization (Flachs et al, ISSCC 2005)



SPE PIPELINE (Flachs et al, ISSCC 2005)



Cell Processor Example Application Areas

- Cell is a processor that excels at processing of rich media content in the context of broad connectivity
  - **Digital content creation (games and movies)**
  - **Game playing and game serving**
  - **Distribution of (dynamic, media rich) content**
  - **Imaging and image processing**
  - **Image analysis (e.g. video surveillance)**
  - **Next-generation physics-based visualization**
  - **Video conferencing (3D?)**
  - **Streaming applications (codecs etc.)**
  - **Physical simulation & science**

Summary

- **Cell ushers in a new era of leading edge processors optimized for digital media and entertainment**
- **Desire for realism is driving a convergence between supercomputing and entertainment**
- **New levels of performance and power efficiency beyond what is achieved by PC processors**
- **Responsiveness to the human user and the network are key drivers for Cell**
- **Cell will enable entirely new classes of applications, even beyond those we contemplate today**

Acknowledgements

- **Cell is the result of a deep partnership between SCEI/Sony, Toshiba, and IBM**
- **Cell represents the work of more than 400 people starting in 2001**
- **More detailed papers on the Cell implementation and the SPE micro-architecture can be found in the ISSCC 2005 proceedings**

Outline

- Motivation
- Cell architecture
  - GPP Controller (PPE)
  - Compute PEs (SPEs)
  - Interconnect (EIB)
  - Memory and I/O
- **Comparisons**
  - **Stream Processors**
- Software (probably next time)

- All Cell related images and figures © Sony and IBM
- Cell Broadband Engine ™ Sony Corp.

Hardware Efficiency → Greater Software Responsibility

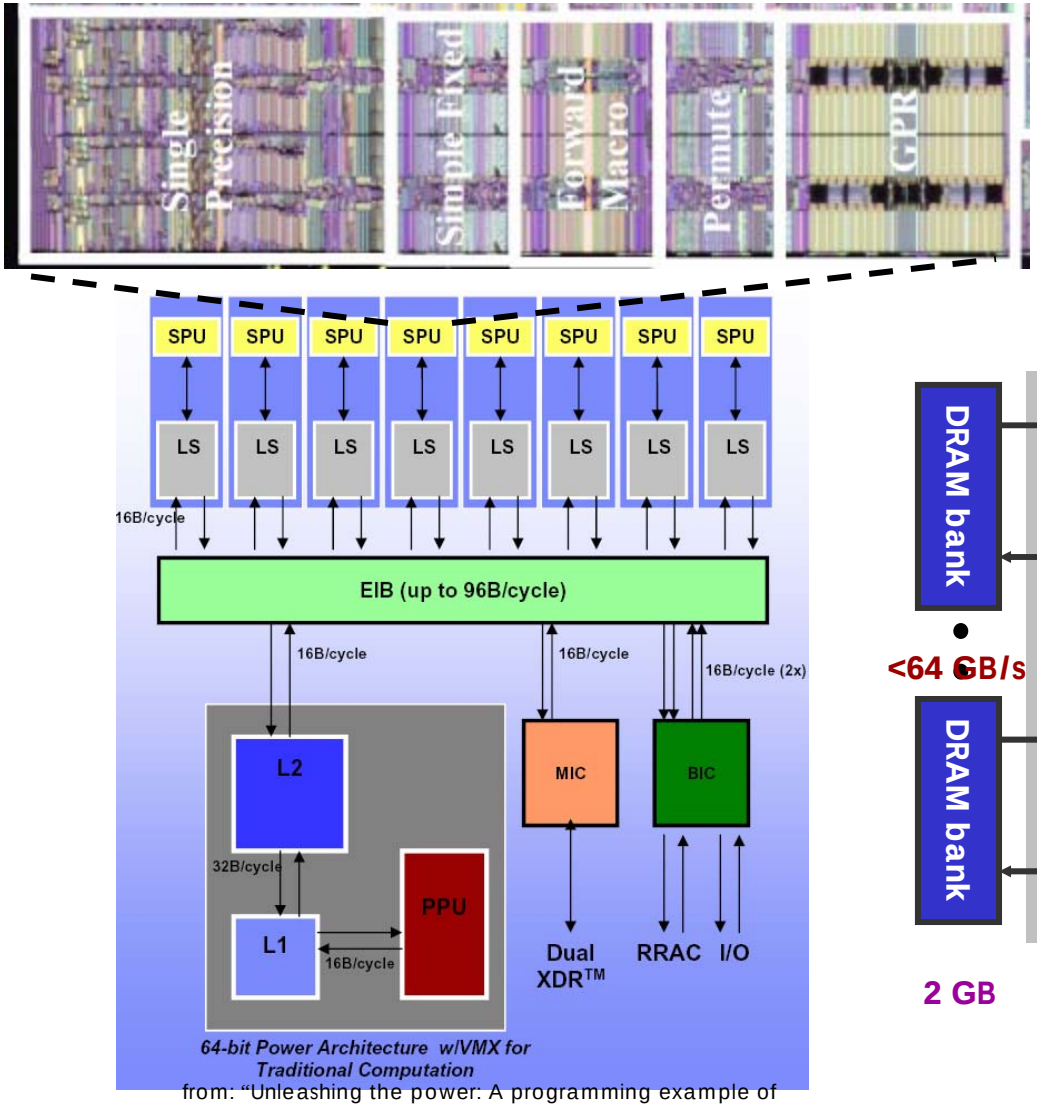
- Hardware matches VLSI strengths
  - Throughput-oriented design
  - Parallelism, locality, and partitioning
  - Hierarchical control to simplify instruction sequencing
  - Minimalistic HW scheduling and allocation

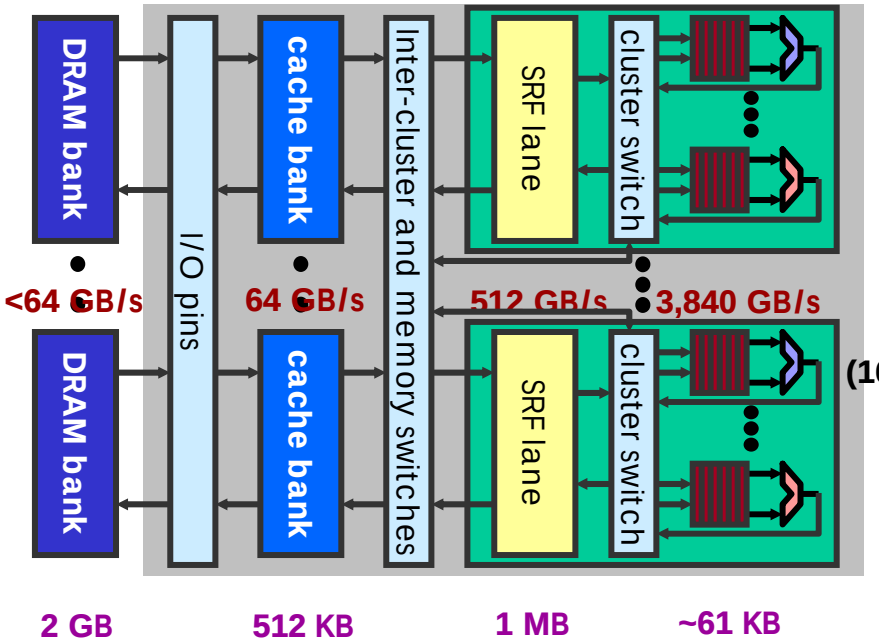
- Software **given** more explicit control
  - Explicit hierarchical scheduling and latency hiding (***schedule***)
  - Explicit parallelism (***parallelize***)
  - Explicit locality management (***localize***)

Must reduce HW "waste" but no free lunch

Locality

Storage/Bandwidth Hierarchy is Key to Efficient High Performance





from: "Unleashing the power: A programming example of large FFTs on Cell" given by Alex Chow at power.org on 6/9/2005

SRF/LS Comparison

- Serve as staging area for memory
- Capture locality as part of the storage hierarchy
- Single time multiplexed wide port
  - kernel access
  - DMA access
  - instruction access
- SPs uses word granularity vs. Cell's 4-word
- SP's SRF has efficient auto-increment access mode
- Cell uses one memory for both code and data
  - Why?

Parallelism

Three Types of Parallelism in Applications

- Instruction level parallelism (ILP)
  - multiple instructions from the same instruction basic-block (loop body) that can execute together
  - true ILP is usually quite limited (~5 - ~20 instructions)
- Task level Parallelism (TLP)
  - separate high-level tasks (different code) that can be run at the same time
  - True TLP very limited (only a few concurrent tasks)
- Data level parallelism (DLP)
  - multiple iterations of a “loop” that can execute concurrently
  - DLP is plentiful in scientific applications

Taking Advantage of ILP

- Multiple FUs (VLIW or superscalar)
  - Cell has limited superscalar (not for FP)
  - Merrimac has 4-wide VLIW FP ops
- Latency tolerance (pipeline parallelism)
  - Cell has 7 FP instructions in flight
  - Merrimac expected to have ~24 FP
  - Merrimac uses VLIW to avoid interlocks and bypass networks
  - Cell also emphasizes static scheduling
    - not clear to what extent dynamic variations are allowed

Taking Advantage of TLP

- Multiple FUs (MIMD)
  - Cell can run a different task (thread) on each SPE + asynchronous DMA on each SPE
    - DMA must be controlled by the SPE kernel
  - Merrimac can run a kernel and DMA concurrently
    - DMAs fully independent of the kernels
- Latency tolerance
  - concurrent execution of **different** kernels and their associated stream memory operations

Taking Advantage of DLP

- Multiple FUs
  - SIMD
    - very (most?) efficient way of utilizing parallelism
    - Cell has 4-wide SIMD
    - Merrimac 16-wide
  - MIMD
    - convert DLP to TLP and use MIMD for different "tasks"
  - VLIW
    - convert DLP to ILP and use VLIW (unrolling, SWP)
- Latency tolerance
  - Overlap memory operations and kernel execution (SWP and unrolling)
  - Take advantage of pipeline parallelism in memory

Memory System

High Bandwidth Asynchronous DMA

- Very high bandwidth memory system
  - need to keep FUs busy even with storage hierarchy
  - Cell has ~2 words/cycle (25.6GB/s)
  - Merrimac designed for 4 words/cycle
- Sophisticated DMA
  - stride (with records)
  - gather/scatter (with records)
- Differences in granularity of DMA control
  - Merrimac treats DMA as stream level operations
  - Cell treats DMA as kernel level operations