All About the Cell Processor

H. Peter Hofstee, Ph. D.
IBM Systems and Technology Group
SCEI/Sony Toshiba IBM Design Center
Austin, Texas

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**
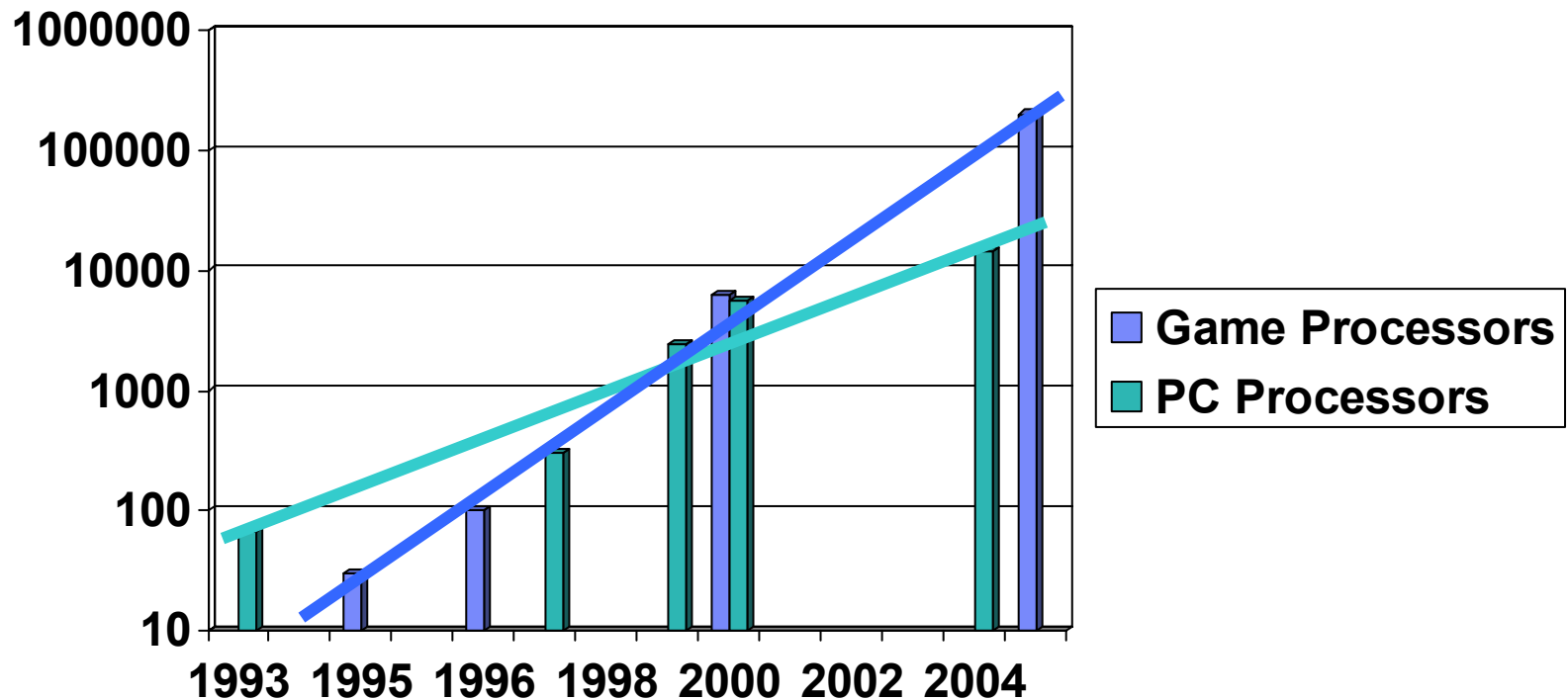
Agenda

- **Trends in Procesors and Systems**

- **Cell Processor Overview**

- **Aspects of the Implementation**

- **Power Efficient Architecture**

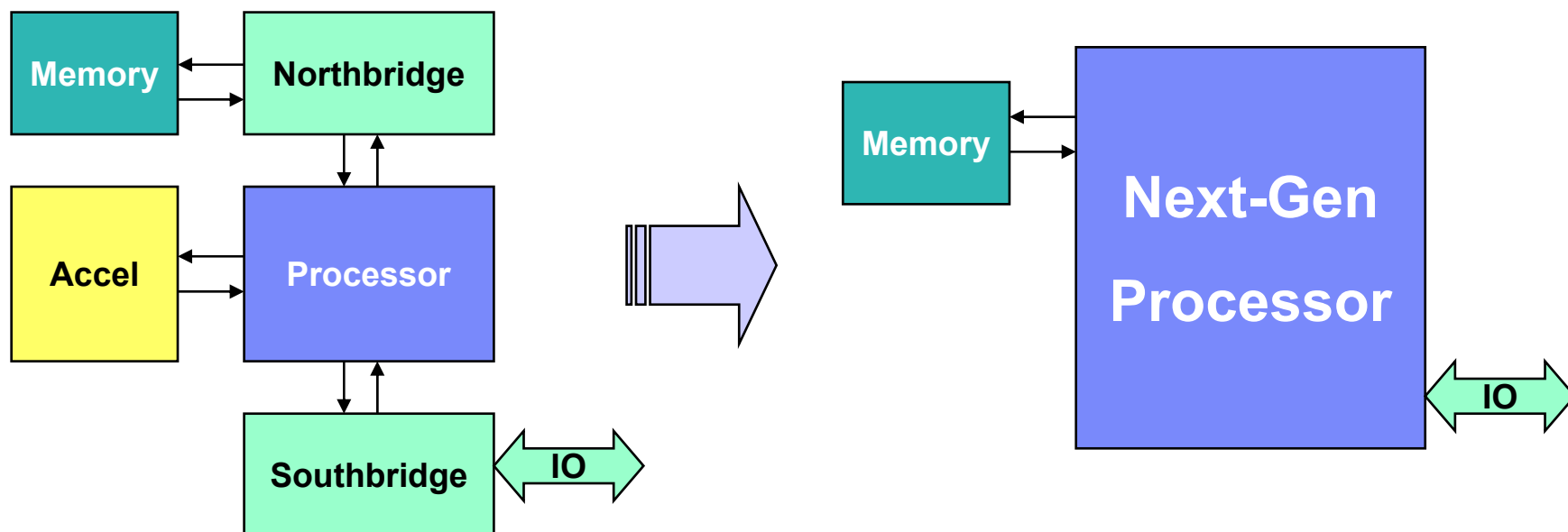# Trends in Microprocessors and Systems

Processor Performance over Time
(Game processors take the lead on media performance)

Flops (SP)

System Trends toward Integration

- **Implied loss of system configuration flexibility**

- **Must compensate with generality of acceleration function to maintain market size.**

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

Hardwired Function

Programmable ASIC

Cell
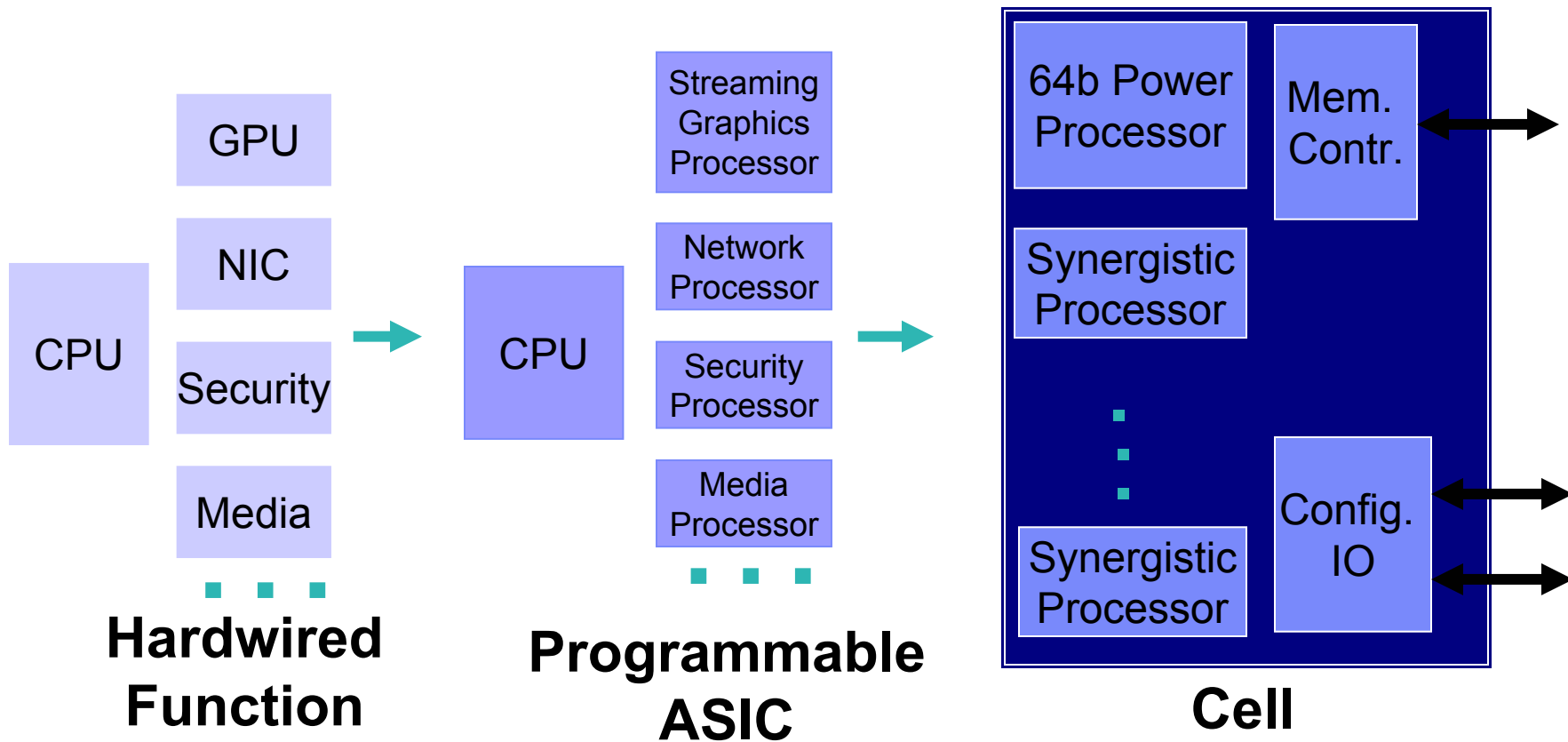
Performance Limiters in Conventional Microprocessors

- **Memory Wall**

  – Latency induced bandwidth limitations

- **Power Wall**

  – Must improve efficiency and performance equally

- **Frequency Wall**

  – Diminishing returns from deeper pipelines
 (can be negative if power is taken into account)

Cell Overview

Cell Goals

- **Outstanding performance, especially on game/multimedia applications.**

- **Real time responsiveness to the user and the network.**

- **Applicable to a wide range of platforms.**

- **Support introduction in systems in 2005.**

Cell Concept

- **Compatibility with 64b Power Architecture™**

  – Builds on and leverages IBM investment and community

- **Increased efficiency and performance**

  – Non Homogenous Coherent Chip Multiprocessor

  – High design frequency, low operating voltage

  – Streaming DMA architecture attacks "Memory Wall"

  – Highly optimized implementation

- **Interface between user and networked world**

  – Image rich information, virtual reality

  – Flexibility and security

- **Multi-OS support, including RTOS/non-RTOS**

  – Combine real-time and non-real time worlds

Cell Chip Block Diagram
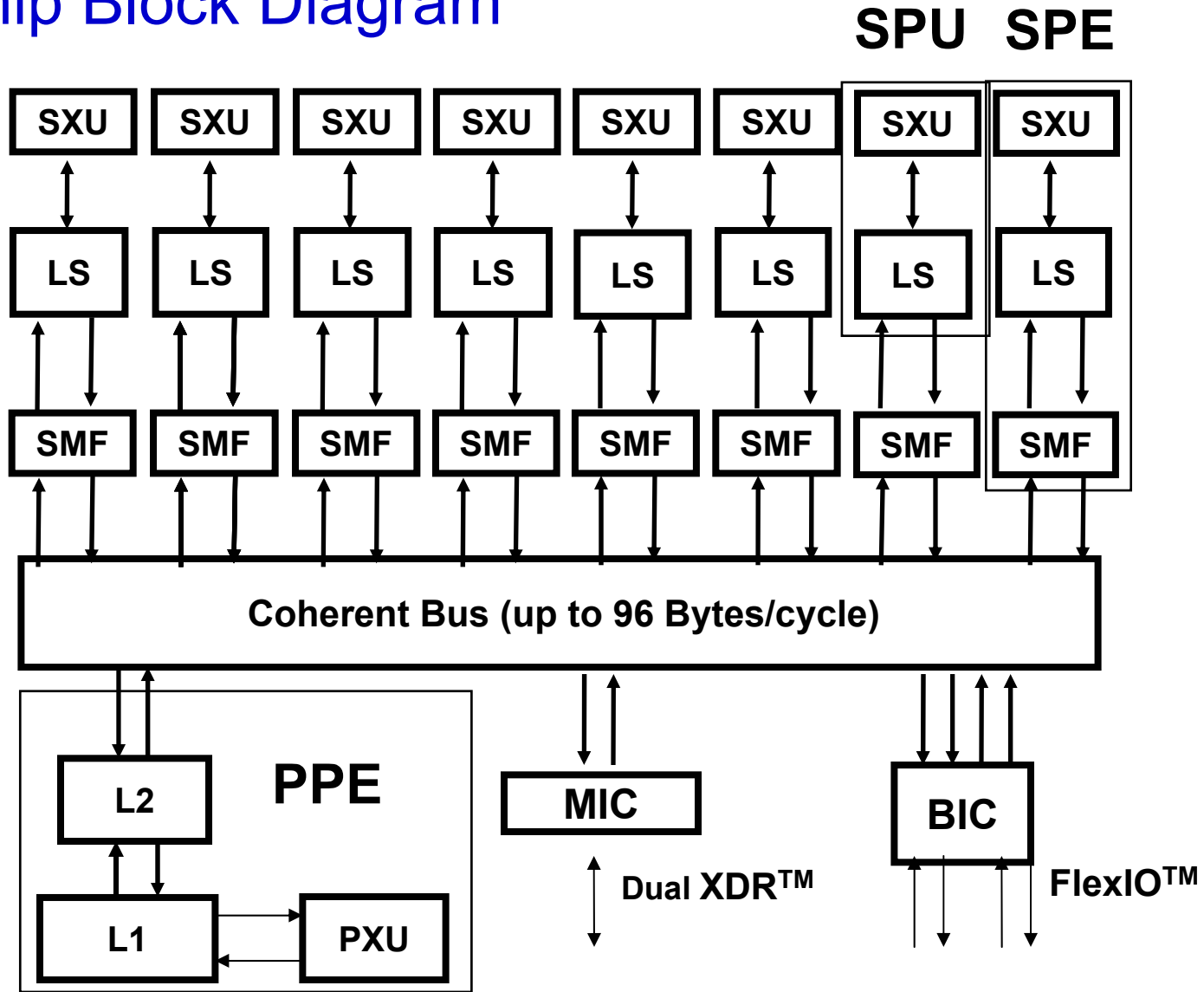
What is a Synergistic Processor?
(and why is it efficient?)

- **Local Store "is" large 2nd level register file / private instruction store instead of cache**

  - Asynchronous transfer (DMA) to shared memory

  - Frontal attack on the Memory Wall

- **Media Unit turned into a Processor**

  - Unified (large) Register File

  - 128 entry x 128 bit

- **Media & Compute optimized**

  - One context

  - SIMD architecture
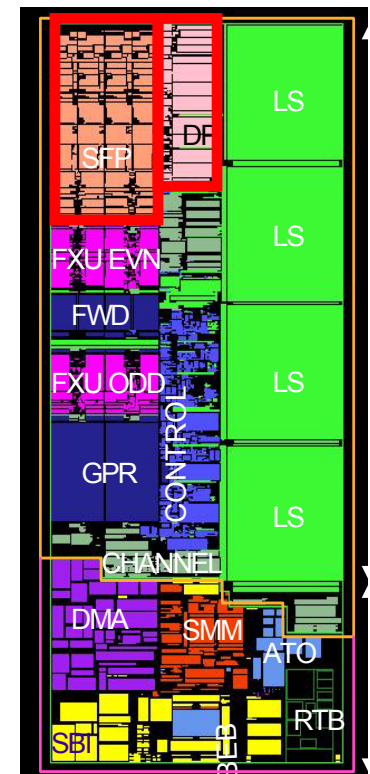
Coherent Offload Model

- **DMA into and out of Local Store equivalent to Power core loads & stores**

- **Governed by Power Architecture page and segment tables for translation and protection**

- **Shared memory model**

  - Power architecture compatible addressing

  - MMIO capabilities for SPEs

  - Local Store is mapped (alias) allowing LS to LS DMA transfers

  - DMA equivalents of locking loads & stores

  - OS management/virtualization of SPEs

    - Pre-emptive context switch is supported (but not efficient)

One approach to programming Cell

- **Single Source Compiler**

  - Auto parallelization (treat target Cell as an SMP)

  - Auto SIMD-ization (SIMD-vectorization)

  - Compiler management of Local Store as 2nd level register file / SW managed cache (I&D)

    - Most Cell unique piece
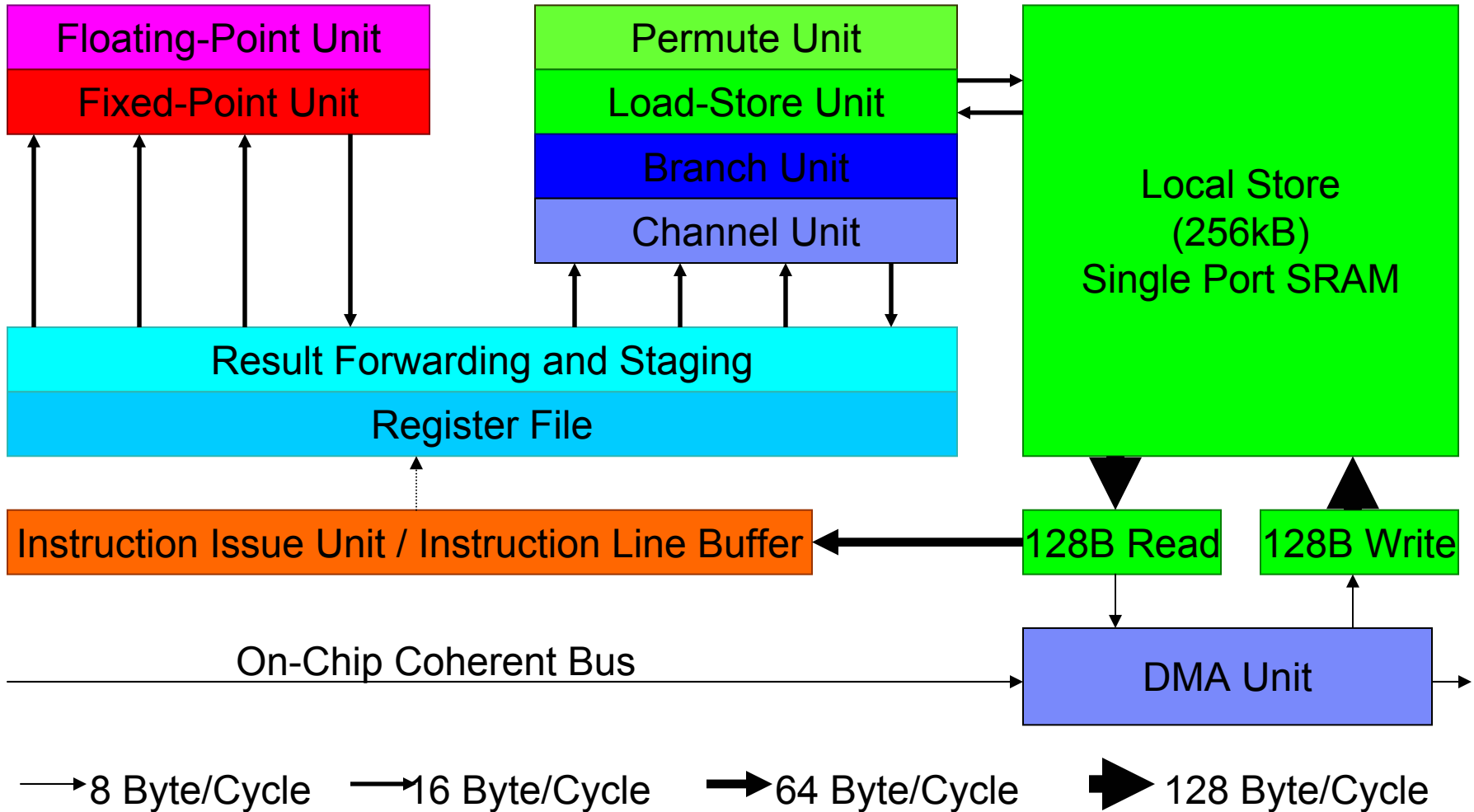
- **Optimization**

  - OpenMP pragmas

  - Vector.org SIMD intrinsics

  - Data/Code partitioning

  - Streaming / pre-specifying code/data use

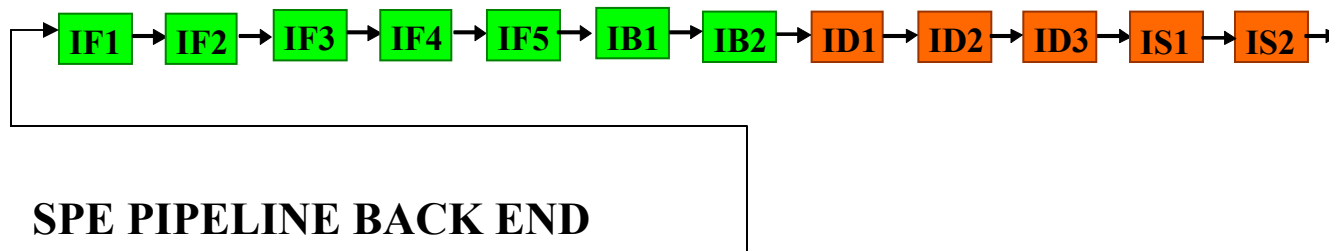- **Prototype Single Source Compiler Developed in IBM Research**
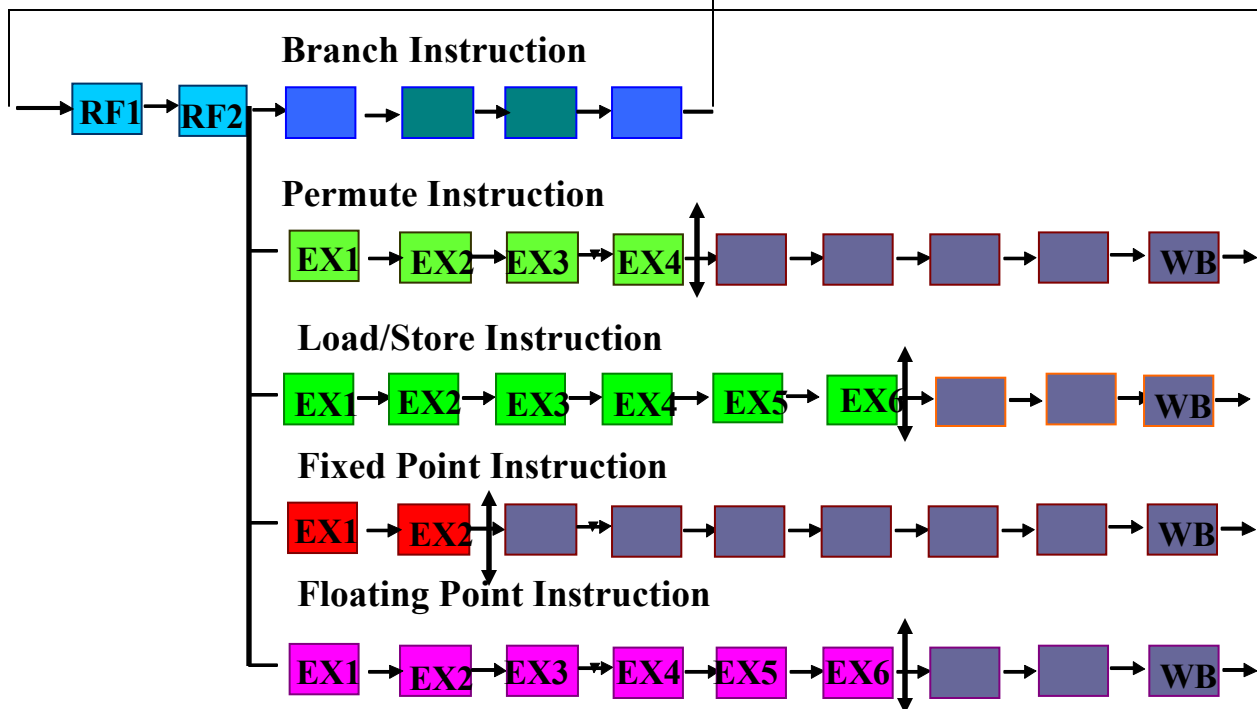
Cell Implementation Aspects

SPE BLOCK DIAGRAM

	Floating-Point Unit
	Fixed-Point Unit

	Permute Unit
	Load-Store Unit
	Branch Unit
	Channel Unit

Local Store
(256kB)
Single Port SRAM

Result Forwarding and Staging

Register File

Instruction Issue Unit / Instruction Line Buffer

128B Read

128B Write

On-Chip Coherent Bus

DMA Unit

→ 8 Byte/Cycle   → 16 Byte/Cycle   ➡ 64 Byte/Cycle   ▶ 128 Byte/Cycle

SPE PIPELINE FRONT END

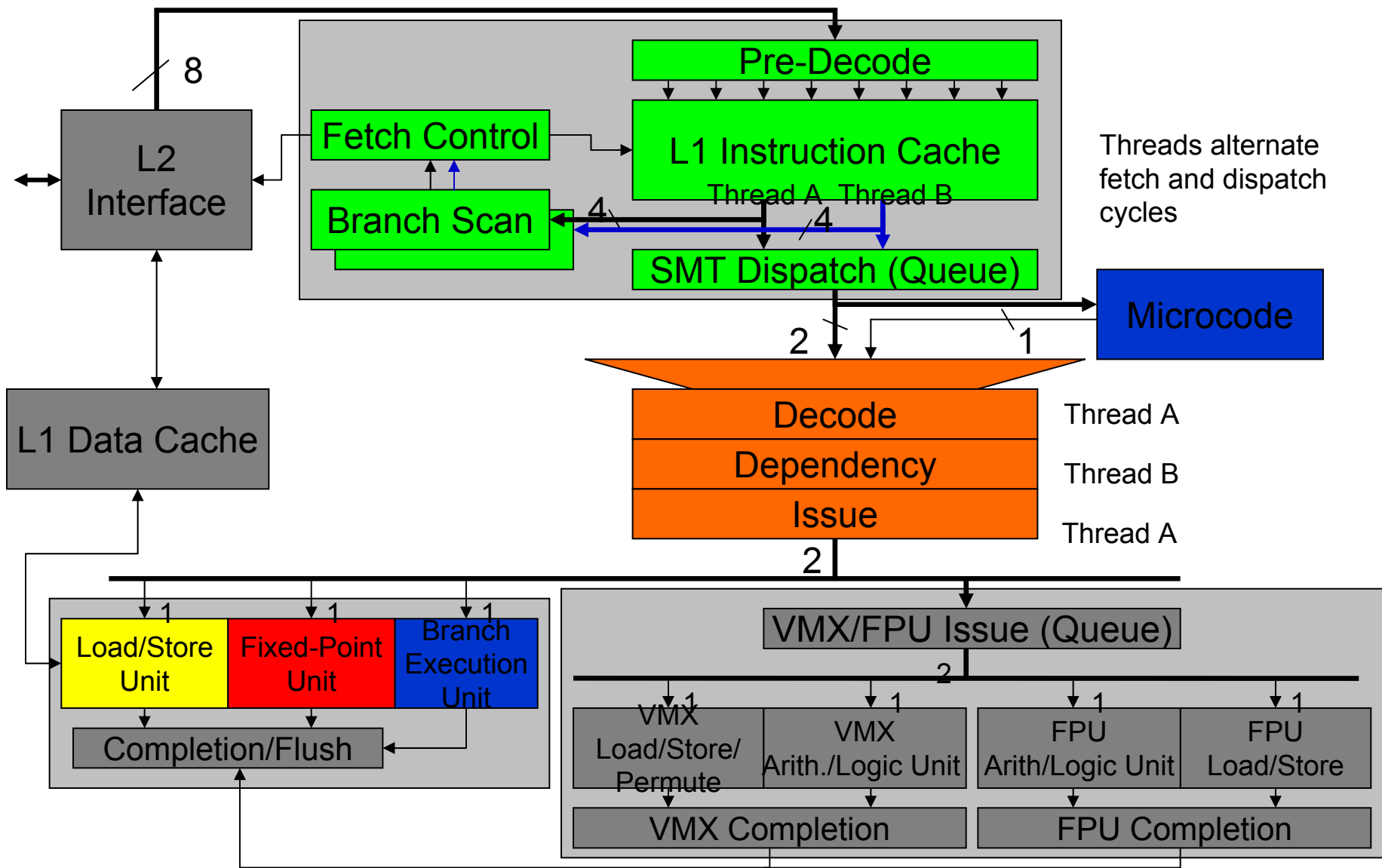**SPE PIPELINE BACK END**
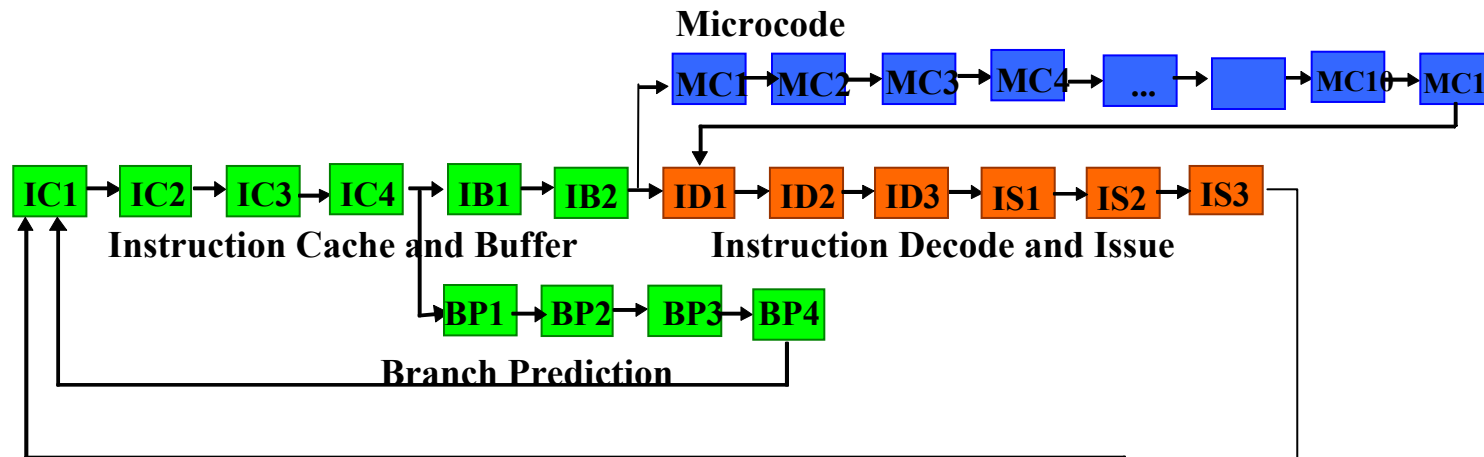
IF   Instruction Fetch
IB   Instruction Buffer
ID   Instruction Decode
IS   Instruction Issue
RF   Register File Access
EX   Execution
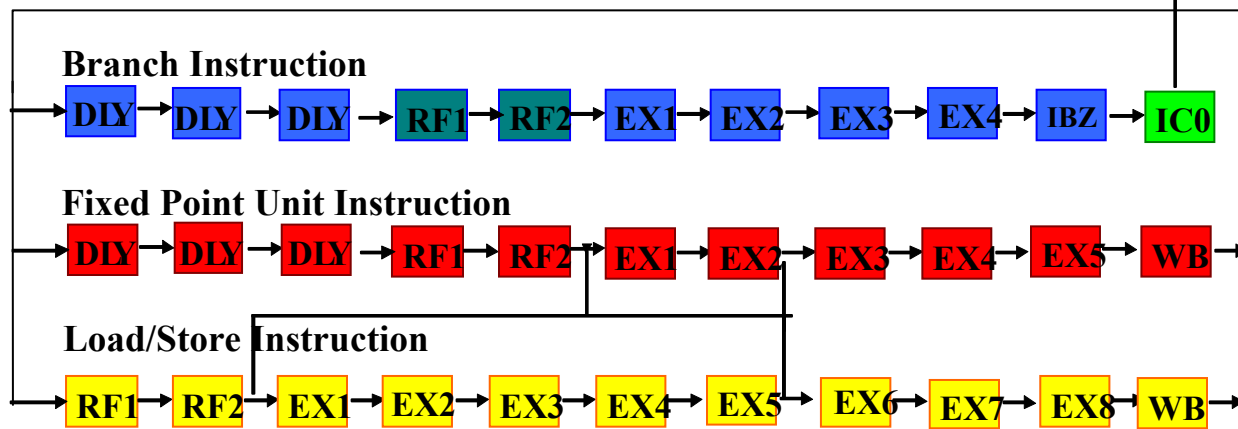WB   Write Back

PPE BLOCK DIAGRAM

PPE PIPELINE FRONT END

Microcode

MC1 → MC2 → MC3 → MC4 → ... → □ → MC10 → MC11

IC1 → IC2 → IC3 → IC4 → IB1 → IB2 → ID1 → ID2 → ID3 → IS1 → IS2 → IS3

Instruction Cache and Buffer   **Instruction Decode and Issue**

BP1 → BP2 → BP3 → BP4

Branch Prediction

PPE PIPELINE BACK END

Branch Instruction

DLY → DLY → DLY → RF1 → RF2 → EX1 → EX2 → EX3 → EX4 → IBZ → IC0

Fixed Point Unit Instruction

DLY → DLY → DLY → RF1 → RF2 → EX1 → EX2 → EX3 → EX4 → EX5 → WB

Load/Store Instruction

RF1 → RF2 → EX1 → EX2 → EX3 → EX4 → EX5 → EX6 → EX7 → EX8 → WB

IC   Instruction Cache
IB   Instruction Buffer
BP   Branch Prediction
MC   Microcode
ID   Instruction Decode
IS   Instruction Issue
DLY Delay Stage
RF   Register File Access
EX   Execution
WB   Write Back

Cell Processor
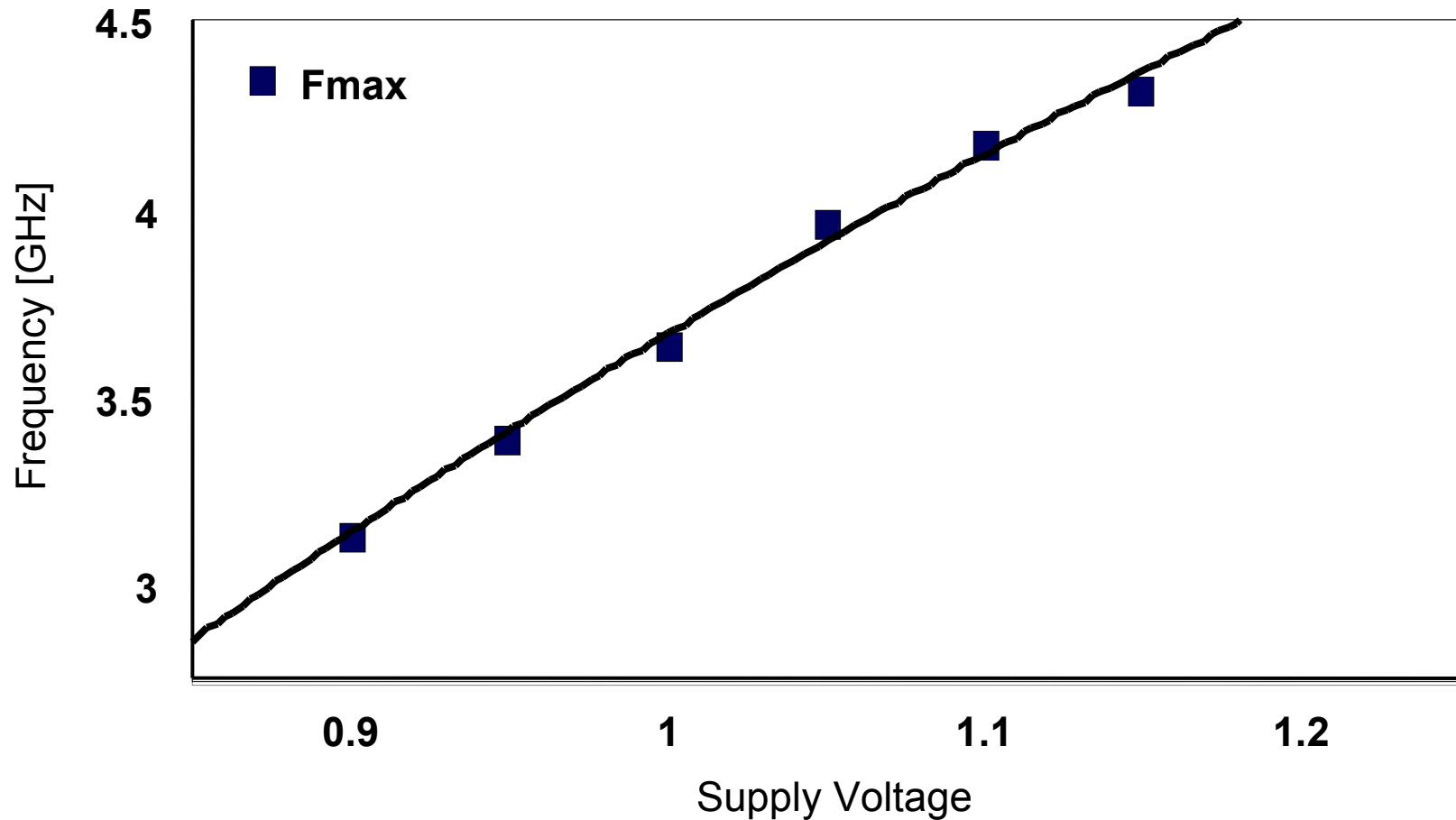
- **~250M transistors**

- **~235mm2**

- **Top frequency >4GHz**

- **9 cores, 10 threads**

- **> 256 GFlops (SP) @4GHz**

- **> 26 GFlops (DP) @4GHz**

- **Up to 25.6GB/s memory B/W**

- **Up to 75 GB/s I/O B/W**

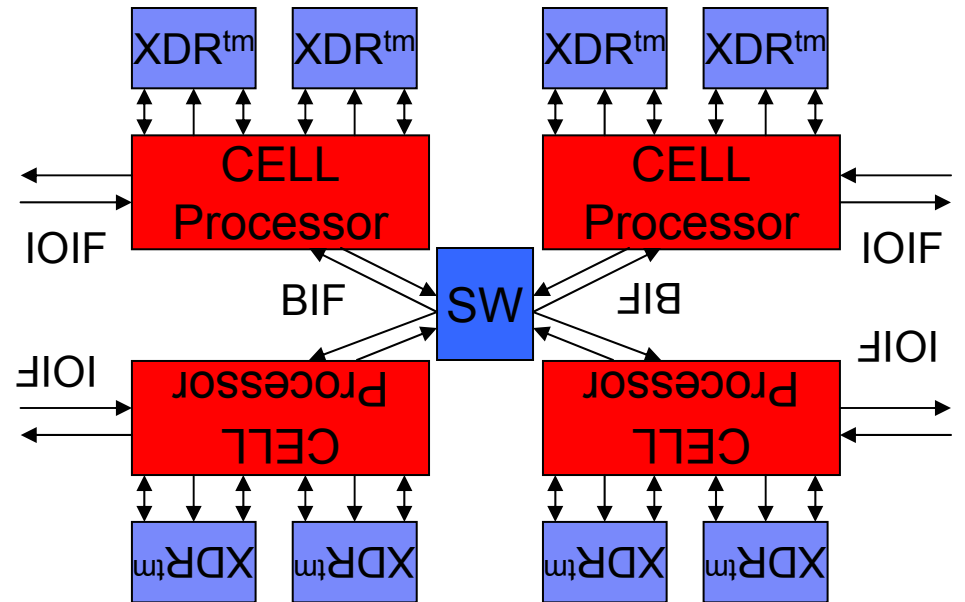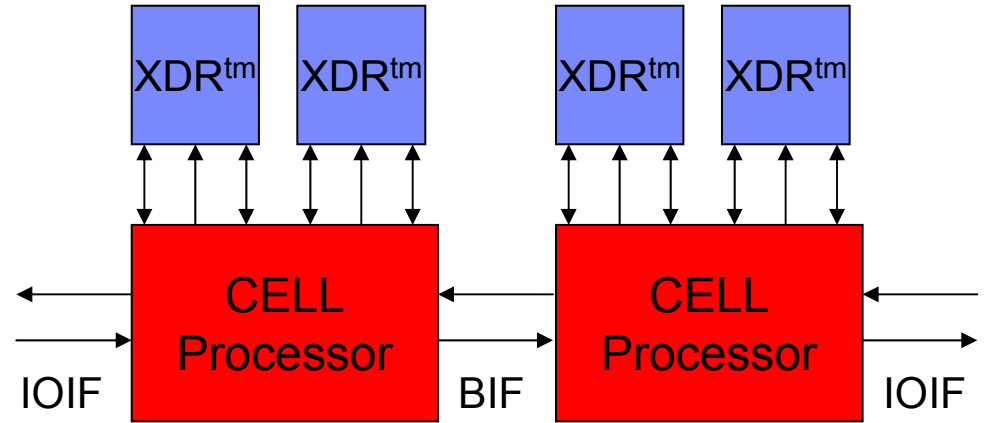- **~400M$(US) design investment**

Cell Broadband Engine Processor
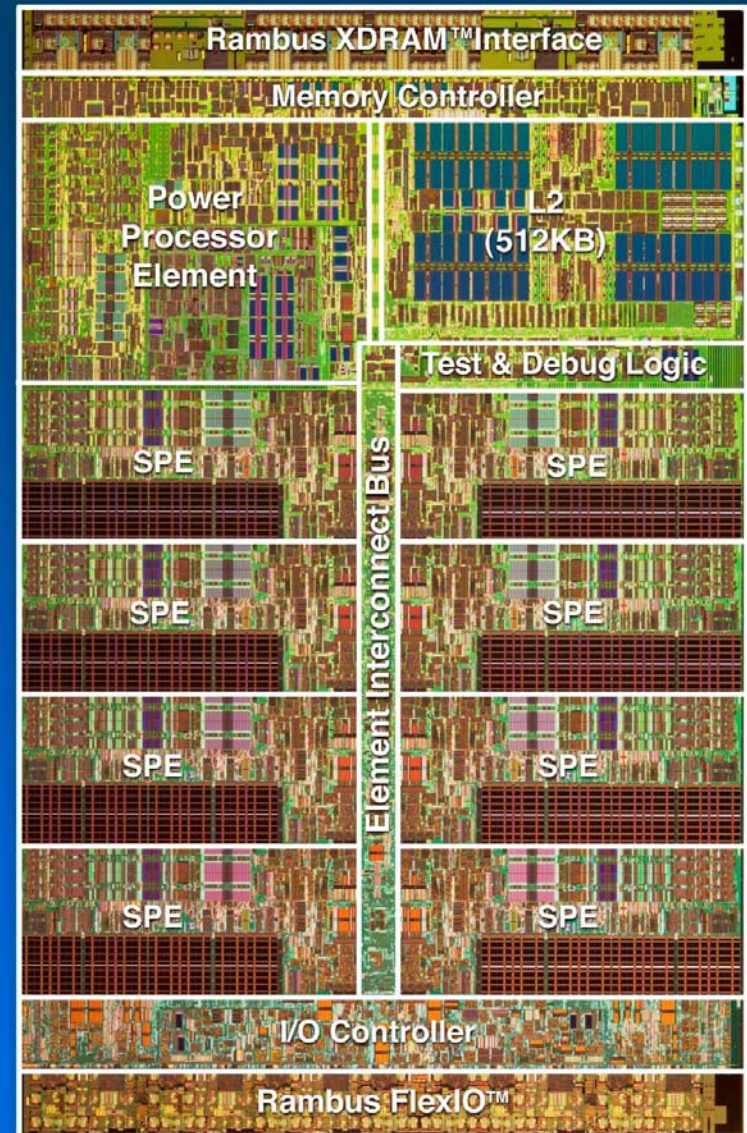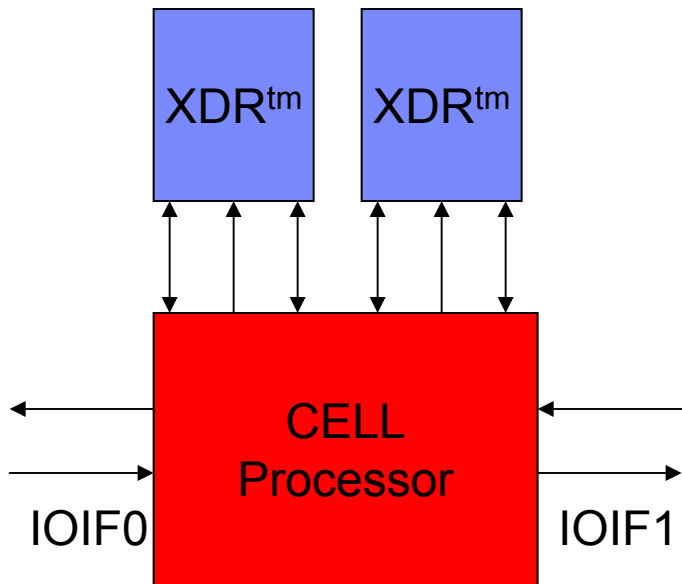
First pass hardware measurement in the Lab - Nominal Voltage = 1V

Hardware Performance Measurement (85°C)

Cell Processor Configurable I/O

- Direct Attach XDR
- Two I/O interfaces
  - Configurable number of Bytes
  - Coherent or I/O Protection

Power Efficient Architecture

Power Efficient Architecture and the BE

- **Non-Homogeneous Coherent Multi-Processor**

  – Data-plane/Control-plane specialization

  – More efficient than homogeneous SMP

- **3-level model of Memory**

  – Bandwidth without (inefficient) speculation

  – High-bandwidth .. Low power

Power Efficient Architecture and the SPE

- **Power Efficient ISA allows Simple Control**

  – Single mode architecture

  – No cache

  – Branch hint

  – Large unified register file

  – Channel Interface

- **Efficient Microarchitecture**

  – Single port local store

  – Extensive clock gating

- **Efficient implementation**

  – See Cool Chips paper by O. Takahashi et al. and T. Asano et al.

Cell BE Processor Example Application Areas

- Cell 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
    - Streaming applications (codecs etc.)
    - Physical simulation & science
- Cell is an excellent match for any applications that require:
    - Parallel processing
    - Real time processing
    - Graphics content creation or rendering
    - Pattern matching
    - High-performance SIMD capabilities

Cell Summary

- **Cell ushers in a new era of leading edge processors optimized for digital media and entertainment**

- **Responsiveness to the human user and the network are key drivers for Cell**

- **New levels of performance and power efficiency beyond what is achieved by PC processors**

- **Cell will enable entirely new classes of applications, even beyond those we contemplate today**