Hardware Architecture of the Cell Broadband Engine Processor

Presented by Wei Wei, 04/20/2009

The CELL/B.E. processor

The *Cell Broadband Enginee* (Cell/B.E.) processor is the first implementation of a new multiprocessor family conforming to the *Cell Broadband Engine Architecture* (CBEA)

The CBEA and the Cell/B.E. processor are the result of a collaboration between Sony, Toshiba, and IBM known as STI, formally begun in early 2001

Although the Cell / B.E. processor is initially intended for applications in media-rich consumer-electronics devices such as game consoles and high-definition televisions, the architecture has been designed to enable fundamental advances in processor performance and supports a broad range of compute-intensive applications.

Cell/B.E. Basic Concepts

- **Compatibility with IBM 64b Power Architecture™**
  - Builds on and leverages IBM investment and community
- **Increased efficiency and performance, especially on media-rich applications**
  - Attacks on the “Power Wall”
    - Heterogeneous Multiprocessor
    - High design frequency @ a low operating voltage with advanced power management
  - Attacks on the “Memory Wall”
    - Streaming DMA architecture
    - 3-level Memory Model: System memory, Local Store, Register Files
  - Attacks on the “Frequency Wall”
    - Highly optimized implementation
    - Large shared register files and software controlled branching to allow deeper pipelines
- **Real time responsiveness to the user and the network**
  - Challenges: Real-time and security in a multiprocessor environment
- **Applicable to a wide range of platforms**
  - Multi-OS support, including RTOS / non-RTOS

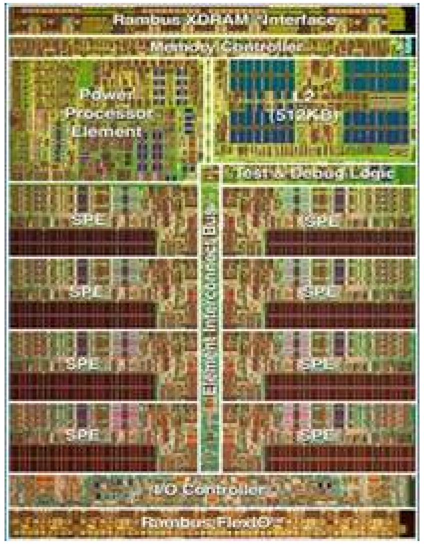
Comparison with traditional processors

Cell/B.E. vs traditional approaches

½ the space & power consumption & much higher performance

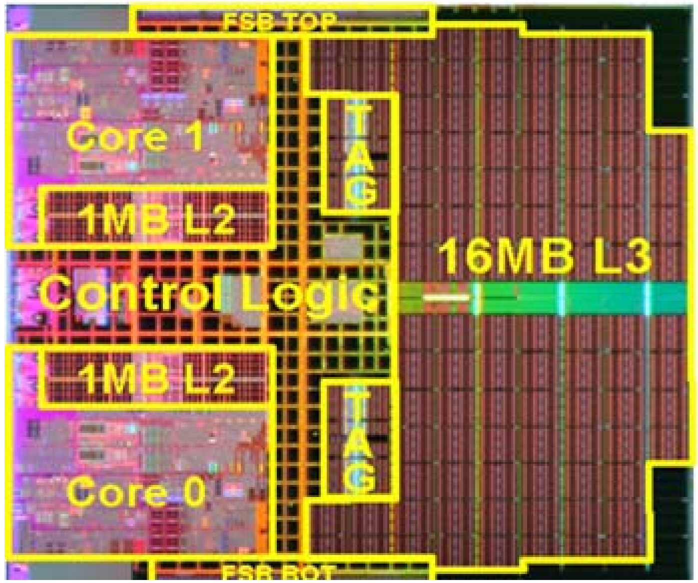
Cell/B.E.

175 mm², 3.2 GHz@60-80W
9 Cores, ~230 SP GFlops



Intel Tulsa (Xeon MP 7100 series)

424mm², 3.4 GHz@150W
2 Cores, ~54 SP GFlops

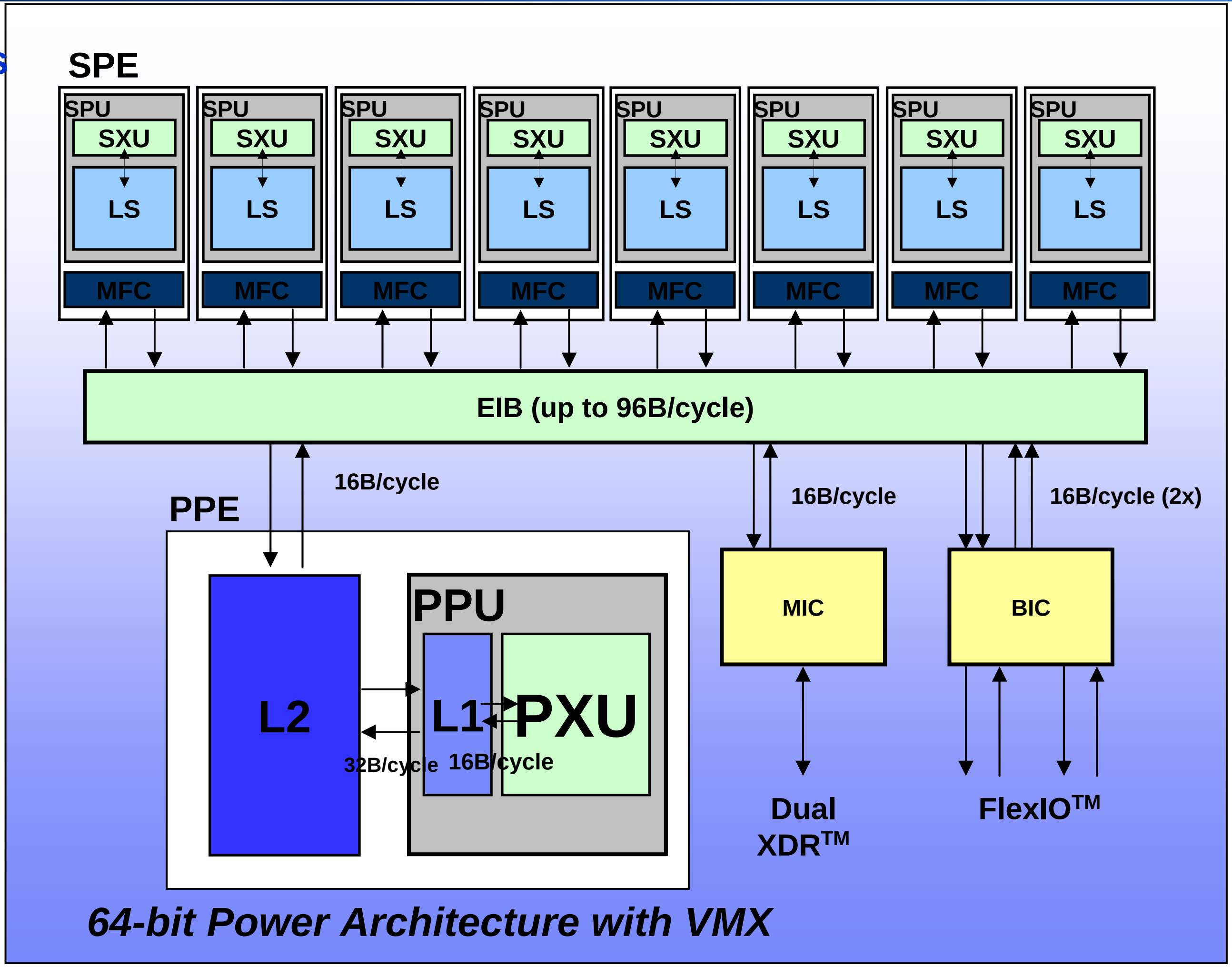


Please note, both processors use the 65nm process.

Overview of the CELL/B.E. processor

CELL/B.E. is a heterogeneous multiprocessor

- A Power Processor Element (PPE)
- 8 Synergistic Processor Elements (SPE)
- A high bandwidth Element Interconnect Bus (EIB)
- A Memory Interface Controller (MIC)
- A bus interface controller (BIC)



Why heterogeneous?

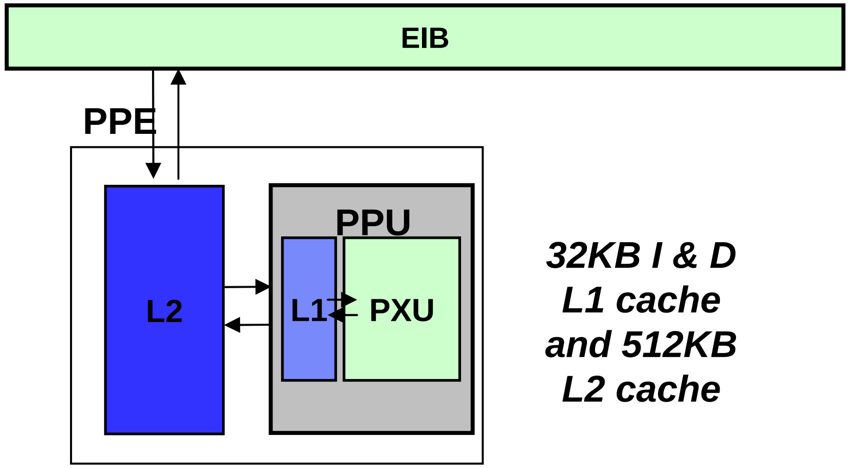
- **PPE: Control Plane**
  - The PPE is responsible for overall control of the chip, e.g., runing the operating system, managing system resources, and allocating tasks to the SPEs.

- **SPE: Data Plane**
  - The SPEs account for the computational power of the Cell/B.E. processor. They are designed to perform the compute-intensive, or ‘‘data plane,’’ processing.

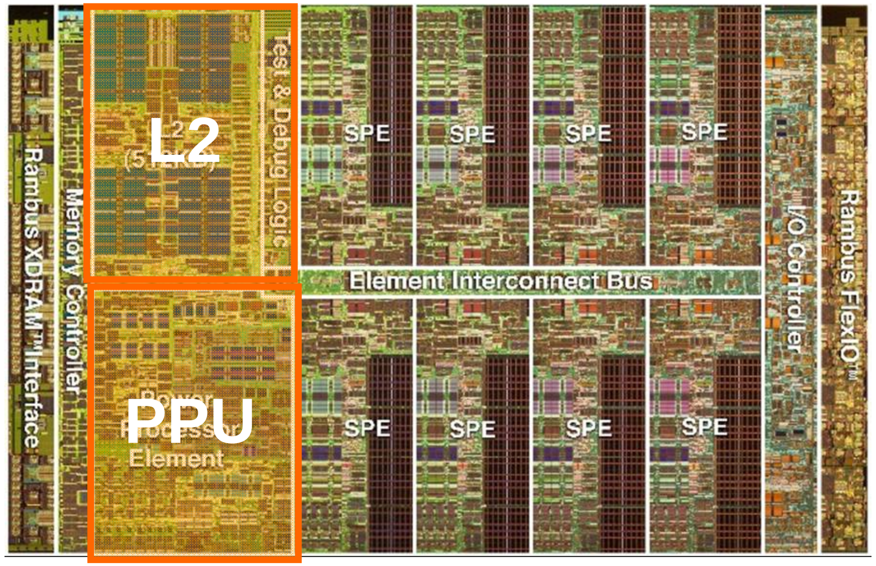
- **Decoupled data processing and control functions**
  - Architectures and implementations of the PPE and SPE can be optimized for their respective workloads and enables significant improvements in performance per transistor.

- **Benefits of Specialization**
  - Cell/B.E. can include nine cores in the same area as an industry-competitive general-purpose processor.
  - Is a significant factor in the substantial performance improvement achieved by CELL/B.E..

Power Processor Element



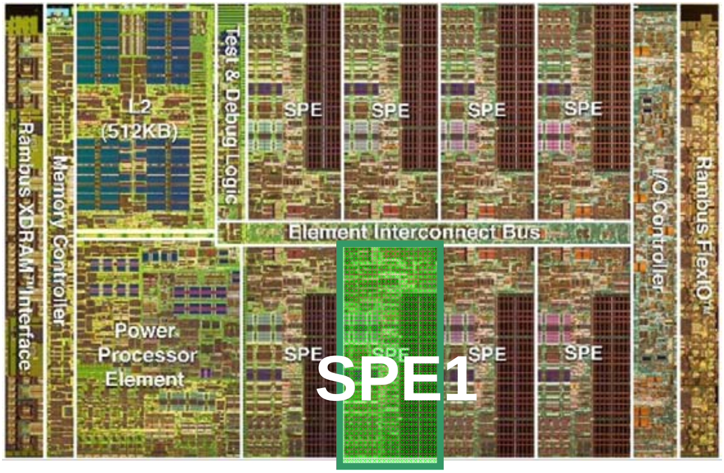
32KB I & D L1 cache and 512KB L2 cache

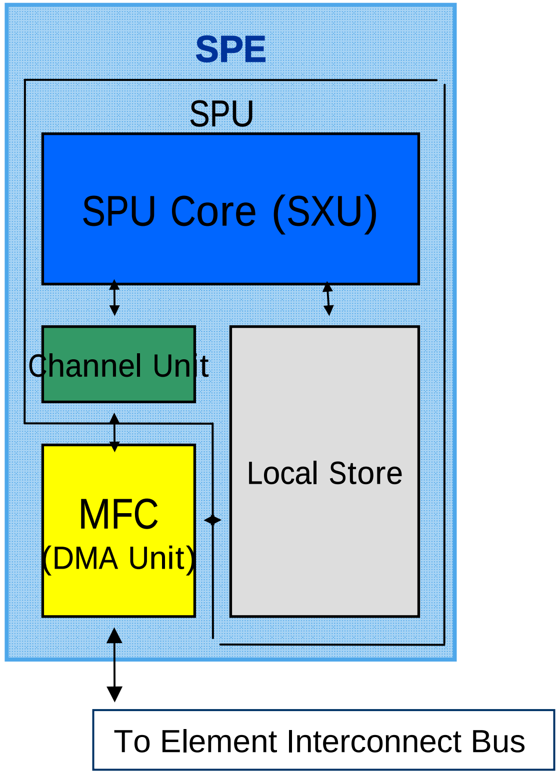


The PowerPC Processor Element (PPE) features:

- **A general-purpose 64-bit RISC processor, conforming to the PowerPC Architecture**
  - Leverage IBM investment
- **In-order, 2-way hardware simultaneous multi-threading (SMT)**
  - Less circuitry and lower energy consumption
- **With vector/SIMD multimedia extension (VMX)**
  - Makes it easier to develop and port applications to the SPE
  - Allows applications to be parallelized across the PPE and SPEs

Synergistic Processor Elements



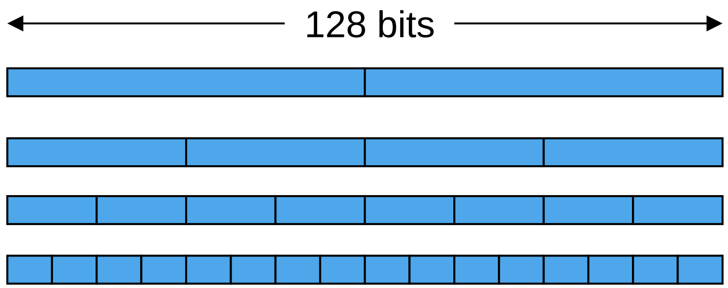


Each SPE:

- **Synergistic Processor Unit (SPU)**
  - A dual-issue, in-order, SIMD processor
  - Contains a 128-entry, 128-bit register file
  - 256KB of private memory (local store)
  - A channel interface to the MFC
- **Memory Flow Controller (MFC)**
  - Data movement to and from main memory, other SPEs’ local stores, or I/O devices

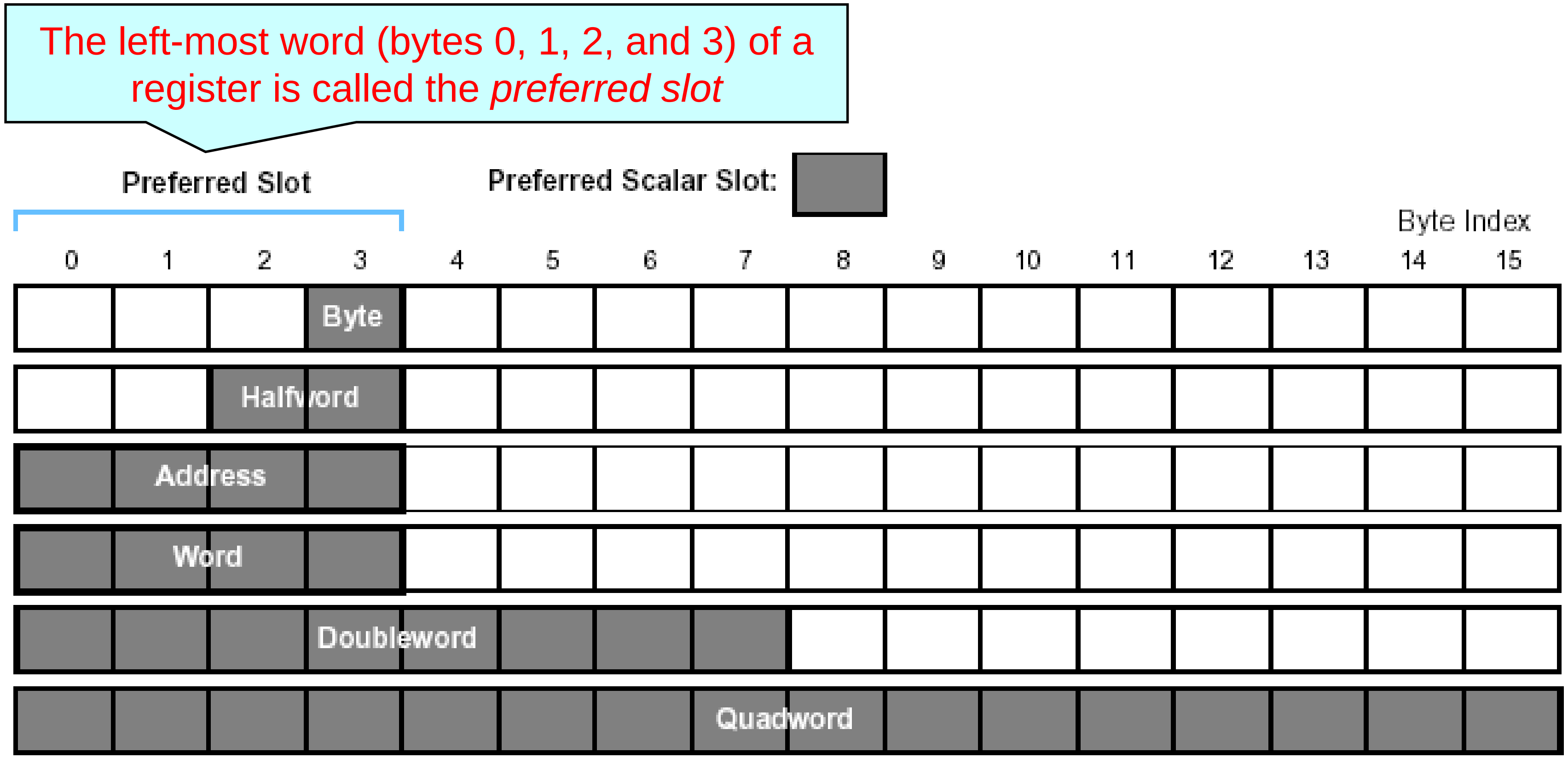
SIMD Architecture in Cell/B.E.

- **SIMD = "single-instruction multiple-data"**
- **SIMD exploits data-level parallelism**
  - a single instruction can apply the same operation to multiple data elements in parallel
- **SIMD units employ "vector registers"**
  - each register holds multiple data elements, e.g., SPE's large 128*128 register file.
- **SIMD is pervasive in Cell/B.E.**
  - PPE integrates SIMD multimedia extension of PowerPC architecture
  - SPE is a native SIMD architecture
    - A SIMD instruction set, SIMD functional units, vector registers
- **SIMD in SPE**
  - All SPE instructions are inherently SIMD
  - Processing 128-bit-wide data in one of four granules:
    - sixteen 8-bit integers
    - eight 16-bit integers
    - four 32-bit integers or SP FP numbers
    - two 64-bit DP FP numbers



Preferred Slot for Scalar Operations

When instructions use or produce scalar operands or addresses, the values are in the preferred scalar slot:



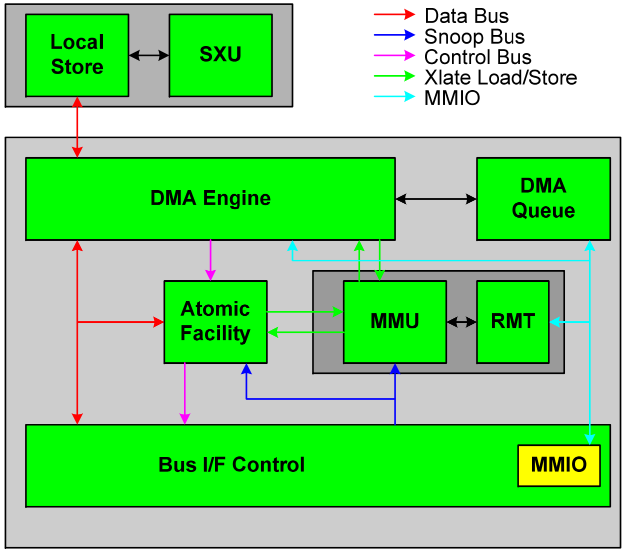
Local Store: CELL/B.E. Attacks the Memory Wall

- Traditional processor architecture
  - Program touches memory, processor checks the caches.
  - If necessary, data is brought in from main memory and left in the caches, hopefully to be reused.
  - Limited ability for the programmer to hint what is needed and what is not.

- CELL/B.E. SPE
  - 256-KB Local Store is a private memory, not a cache.
  - SPE has load/store & instruction-fetch access only to its local store.
  - No caching, tags, backing storage, etc. – fixed access time (6 cycles).
  - Access to main memory is entirely controlled by the programmer using DMA commands.
  - DMA transfers happen asynchronously; overlap processor computation with data movement.

This 3-level organization of memory (register file, LS, main memory) is a radical break from conventional architecture and programming models

DMA capability



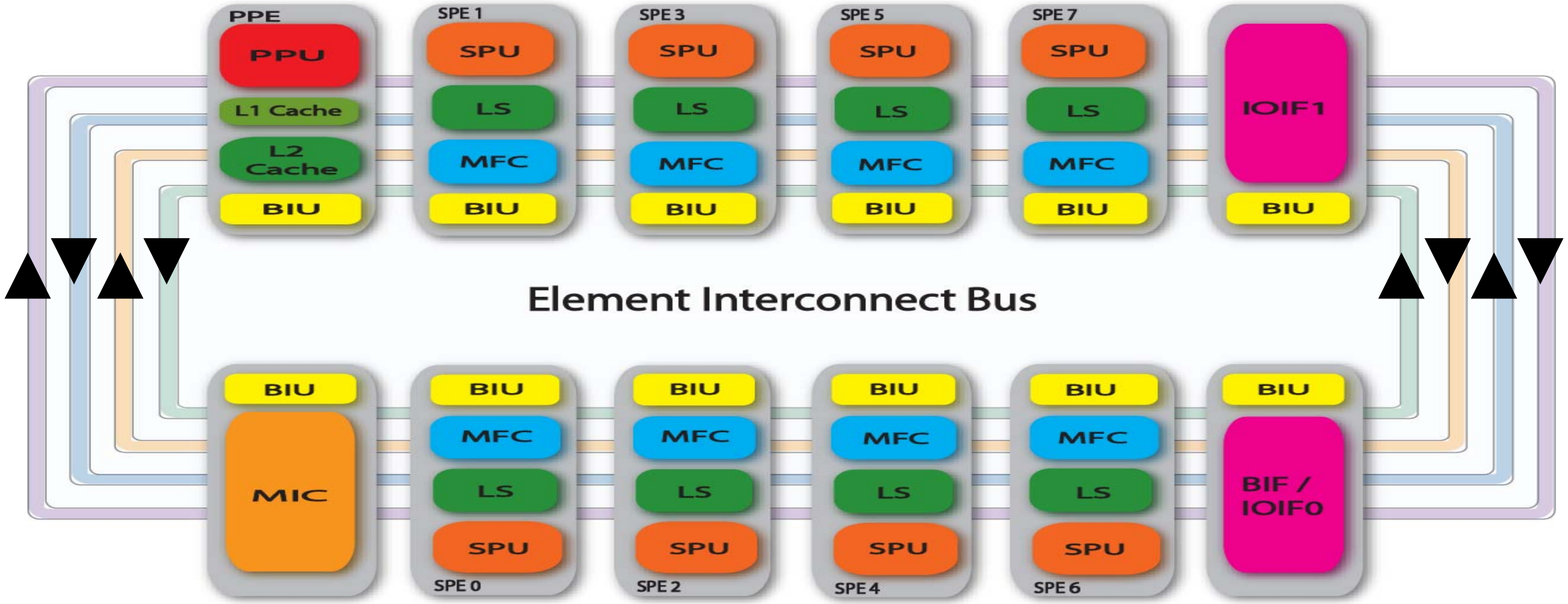
- The memory flow controller (MFC) delivers asynchronous DMA capability for data and instruction transfers between the local store and main memory.
  - DMA commands
- **DMA transfers**
  - DMA commands can be issued by either SPEs or PPE
  - Transfer sizes can be 1, 2, 4, 8, and n*16 bytes
  - Up to 16KB/command
- **DMA queues**
  - 16-element queue for DMA commands issued by the associated SPE
  - 8-element queue for DMA commands issued by external elements
- **DMA lists**
  - A single DMA list command can convey a list of DMA commands.
  - A list can contain up to 2K transfer requests
  - Amortize DMA latency (475 cycles for get)
  - Lists implement scatter-gather functions

PPE vs SPE

- PPE is designed for general-purpose tasks
- SPE is optimized for compute-intensive applications

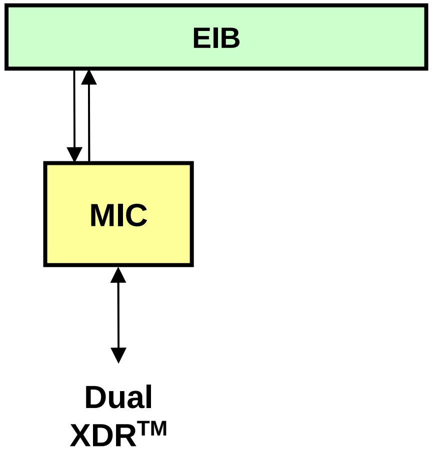
Feature	PPE	SPE
Number of SIMD registers	32 (128-bit)	128 (128-bit)
Organization of register file	separate fixed-point, FP, and vector multimedia registers	unified
Load latency	variable (cache)	fixed
Addressability	2^{64} bytes	256 KB local store 2^{64} bytes DMA
Instruction set	more orthogonal	optimized for compute-intensive applications
Single-precision	IEEE 754-1985	extended range 1.2E-38 to 6.8E38
Double-precision	no doubleword SIMD	2-way SIMD DP FP
Main memory access	load and store instructions moving data between main memory and private register file	DMA commands moving data and instructions between main memory and local store

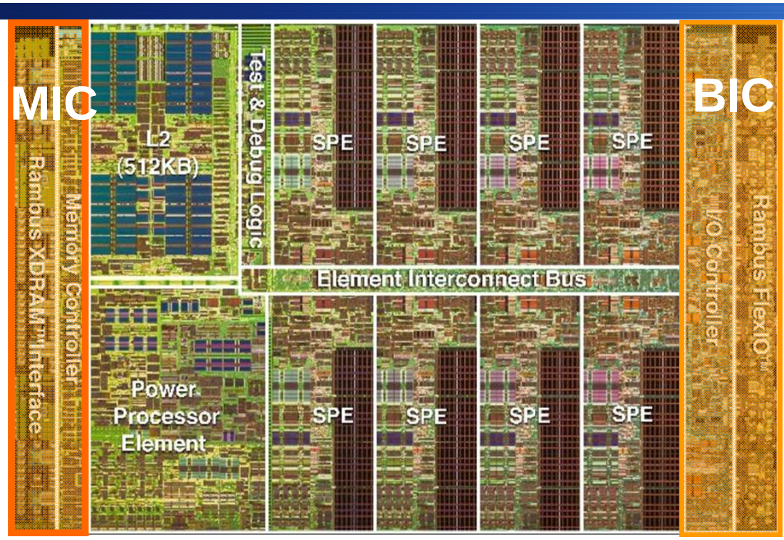
Element Interconnect Bus

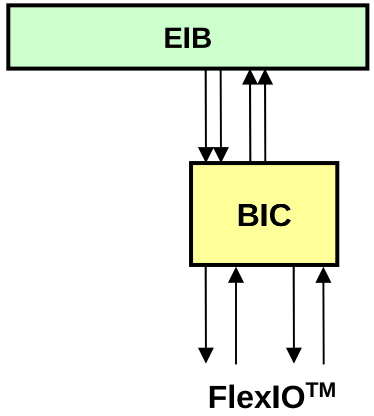


- Interconnects 12 elements
- Four 16-byte-wide unidirectional rings
- Each ring supports up to three simultaneous data transfers
- Transfers occur at half the frequency of the processor, i.e., 96 bytes/cycle theoretical peak bandwidth

Memory Interface Controller and Bus Interface Controller







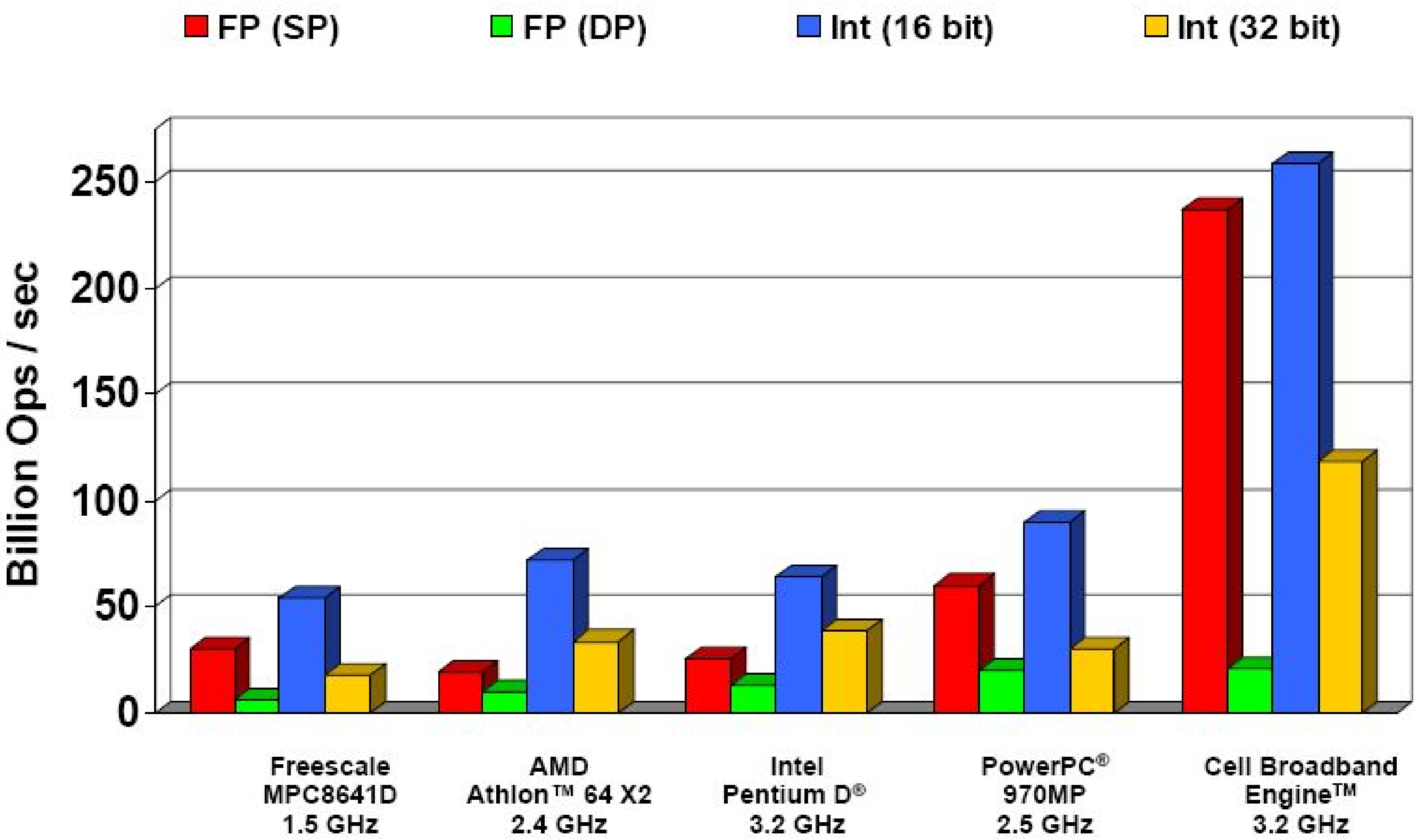
- Connected to the external Rambus DRAM through two XIO channels
- Each channel can have eight memory banks
- 32 read and 32 write queues for each channel
- 25.6 GB/s @ 3.2 GHz peak memory bandwidth

- 7 transmit and 5 receive Rambus FlexIO links configured as 2 logical interfaces
- 1-byte-wide each link @ 5GHz
- 35 GB/s outbound and 25GB/s inbound peak raw bandwidth

High bandwidth contributes to CELL/B.E.'s performance.

Cell/B.E. Performance

Theoretical Peak Performance



Cell/B.E. Performance

Cell BE Performance Summary

Type	Algorithm	3.2 GHz GPP	3.2 GHz Cell	Cell Perf Advantage
HPC	Matrix Multiplication (S.P.)	24 Gflops (w/SIMD)	200 GFlops* (8SPEs)	8x
	Linpack (S.P.)	16 GFlops (w/SIMD)	156 GFlops* (8SPEs)	9x
	Linpack (D.P.): 1kx1k matrix	7.2 GFlops (IA32/SSE3)	9.67 GFLops* (8SPEs)	1.3x
graphics	Transform-light	170 MVPS (G5/VMX)	256 MVPS** (per SPE)	12x
	TRE	1 fps (G5/VMX)	30 fps* (Cell)	30x
security	AES encryp. 128-bit key	1.03 Gbps	2.06Gbps** (per SPE)	16x
	AES decryp. 128-bit key	1.04 Gbps	1.5Gbps** (per SPE)	11x
	TDES	0.12 Gbps	0.16 Gbps** (per SPE)	10x
	DES	0.43 Gbps	0.49 Gbps** (per SPE)	9x
	SHA-1	0.85 Gbps	1.98 Gbps** (per SPE)	18x
video processing	mpeg2 decoder (CIF)	----	1267 fps* (per SPE)	--
	mpeg2 decoder (SDTV)	354 fps (IA32)	365 fps** (per SPE)	8x
	mpeg2 decoder (HDTV)	----	73 fps* (per SPE)	--

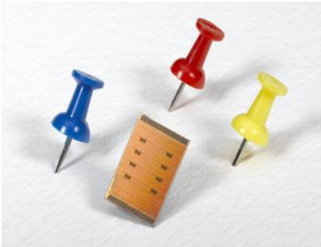
Notes: * Hardware measurement ** Simulation results

Source: Cell Broadband Engine Architecture and its first implementation – A performance view, http://www.ibm.com/developerworks/library/pa-cellperf/

Why is Cell/B.E. So Fast?

- ***The SPE is a fast lean core optimized for compute-intensive processing***
  - Each SPE (3.2 GHz) is up to 3 times faster than the Pentium core (3.6 GHz) when computing FFTs
  - That is 24X better performance chip to chip
- ***Parallel processing inside chip***
  - 8 SPEs run concurrently
- ***Specialization***
  - PPE: Control Plane
  - SPE: Data Plane
- ***High bandwidth***
  - 205 GB/s sustained ring bandwidth
  - 25.6 GB/s main memory bandwidth
  - 60 GB/s I/O bandwidth
- ***High performance DMA transfers***
  - DMA transfers can be fully overlapped with core computation
  - Software controlled DMA transfers can bring the right data into local store at the right time

Cell/B.E. Products



SCE PS3
(Cell/B.E. + GPU)

Mercury Cell/B.E. PCI Card
(Cell/B.E. + Network)

Sony Cell/B.E. Computing Unit

(Cell/B.E. + GPU + AV I/O)

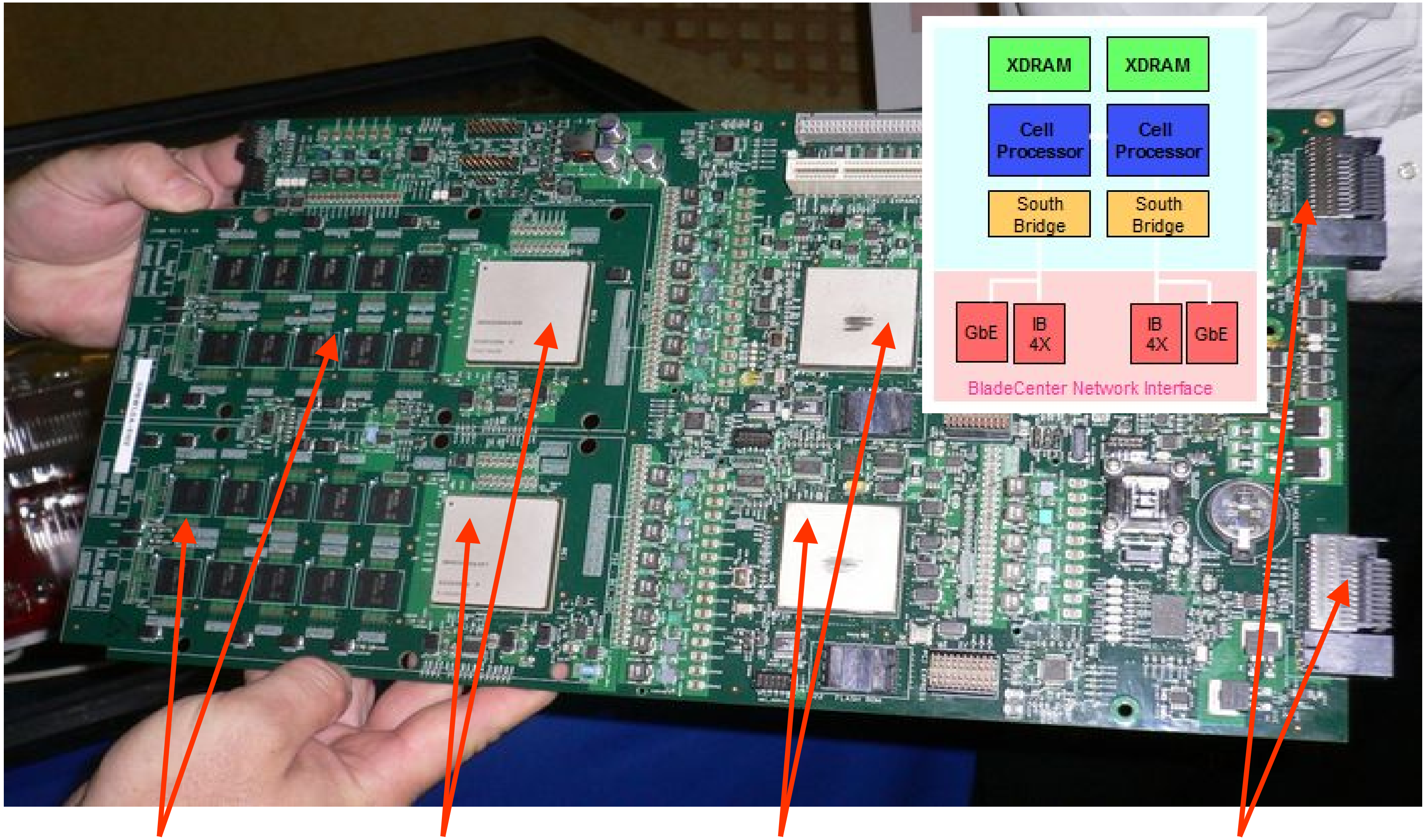
IBM Cell/B.E. Blade
(2 Cell/B.E.s)

IBM Roadrunner
(16,000 Cell/B.E.s + AMD)

Consumer ***Professional*** ***Business*** ***High Perf Computing***

Common Operating Systems, Infrastructure, Tools, Libraries, Code…

The First Generation Cell/B.E. Blade (QS20)



1GB XDR Memory | Cell Processors | IO Controllers | IBM Blade Center interface

IBM BladeCenter QS20 and beyond

September 2006

BladeCenter QS20
- 2 Cell/B.E. processors
- 1PPE + 8SPE
- SP: 460 GFLOPS per Cell blade
- DP: 42 GFLOPS per Cell blade
- 1 GB memory

SDK 1.1

Available July 2006

August 2007

BladeCenter QS21
- 2 Cell/B.E. processors
- 1PPE + 8SPE
- SP: 460 GFLOPS per Cell blade
- DP: 42 GFLOPS per Cell blade
- Next Generation I/O chip
- 2 GB memory

SDK 2.1

Available: March 07

SDK 3.0

Target release: September 07

May 2008

BladeCenter QS22
- 2 CBEA-compliant processors
- 1PPE + 8eDP SPE
- SP: 460 GFLOPS per blade
- DP: 217 GFLOPS per blade
- Up to 32 GB memory
- PCI Express™ x16 slots

SDK 4.0

Target release: March 08

Target availability: 1H10

BladeCenter QS2Z
- First CBEA teraflop processor
- 2PPE'+32 eSPE
- Power Architecture compliant
- ~2 TFLOPS SP per blade
- ~1 TFLOPS DP per blade
- Next generation memory technology

SDK 5.0

Target release: December 08

Committed
Concept

2006 | 2007 | 2008 | 2009-2010

Thank you!