Cell Broadband Engine

21.March 2006

Benjamin Keck
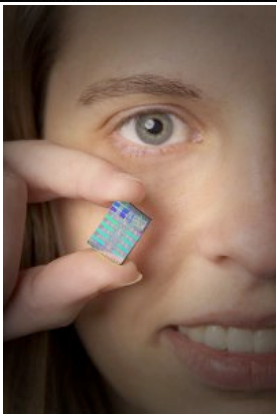
Outline
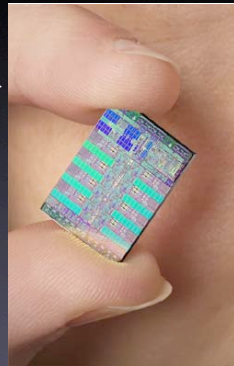
- Why Cell ?!?
- Application Areas
- Architectural Overview
- SPU Programming Model
- Programming on the PPE
- C/C++ Intrinsics

The Cell

- Supercomputer on a chip
- Multi-Core Microprocessor (9 cores)
- 10x perfomance for many applications
- > 4 GHz clock frequency
- Digital home to distributed computing

Application Areas

- Playstation 3
- Multiple parallel calculations
- High performance in multimedia applications
- Energy efficiency (GFLOPS/Watt)

Playstation 3 Example

Key Attributes

- **Cell is Multi-Core**
  - *Contains 64-Bit Power Architecture*
  - *Contains 8 Synergistic Processor Elements (SPE)*
- **Cell is a Flexible Architecture**
  - *Multi-OS support (including Linux) with Virtualization technology*
  - *Path for OS, legacy apps, and software development*
- **Cell is a Broadband Architecture**
  - *SPE is RISC architecture with SIMD organization and Local Store*
  - *128+ concurrent transactions to memory per processor*
- **Cell is a Real-Time Architecture**
  - *Resoure allocation (for Bandwidth Measurement)*
  - *Locking Caches (via Replacement Management Tables)*
- **Cell is a Security Enabled Architecture**
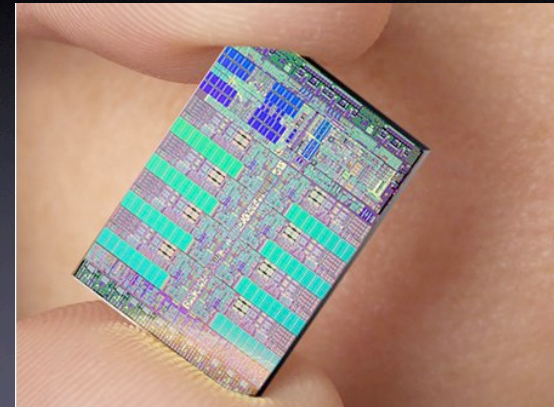 - *Isotalatable SPE for flexible security programming*

High Performance Computing

- Peak performance 256 GFlops

- Place 490 of the 500 Best Supercomputers has 854 GFlops using IBM xSeries Cluster with 256 Intel Xeon processors (2.8 GHz)

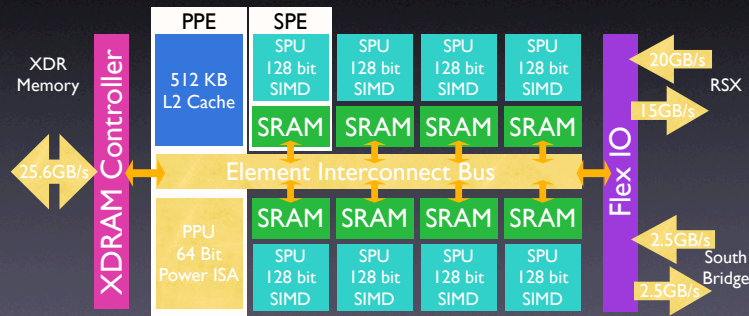- One Cell can decompress 48 MPEG2 Streams (DVD) in parallel

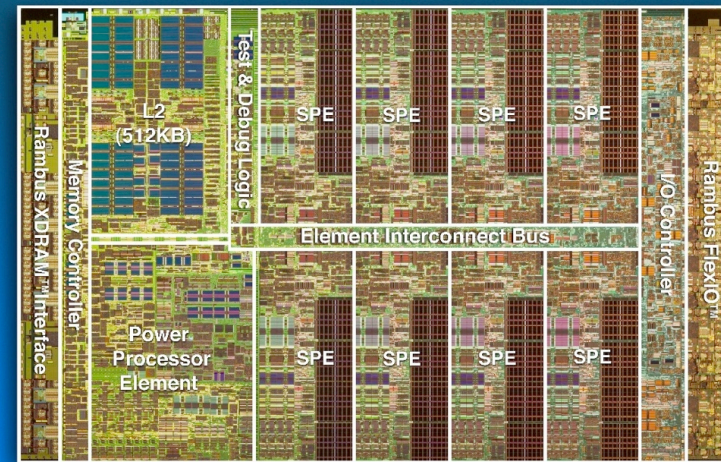## Cell Broadband Engine Architecture (CBEA)

Cell Broadband Engine Architecture (CBEA)

XDR Memory — XDRAM Controller — 25.6GB/s

PPE: 512 KB L2 Cache
PPU 64 Bit Power ISA

SPE:
SPU 128 bit SIMD | SPU 128 bit SIMD | SPU 128 bit SIMD | SPU 128 bit SIMD
SRAM | SRAM | SRAM | SRAM

Element Interconnect Bus

SRAM | SRAM | SRAM | SRAM
SPU 128 bit SIMD | SPU 128 bit SIMD | SPU 128 bit SIMD | SPU 128 bit SIMD

Flex IO
20GB/s — RSX
15GB/s
2.5GB/s — South Bridge
2.5GB/s

Cell Broadband Engine Processor

Rambus XDRAM™ Interface · Memory Controller · L2 (512KB) · Test & Debug Logic · SPE · SPE · SPE · SPE · Element Interconnect Bus · Power Processor Element · SPE · SPE · SPE · SPE · I/O Controller · Rambus FlexIO™

IBM

Power Processor Element (PPE)

- PPE handles operating system and control tasks

  - 64-bit Power Architecture with VMX

  - In-order, 2-way hardware Multi-threading

  - Cohorent Load/Store with 32 KB I & D L1 and 512 KB L2

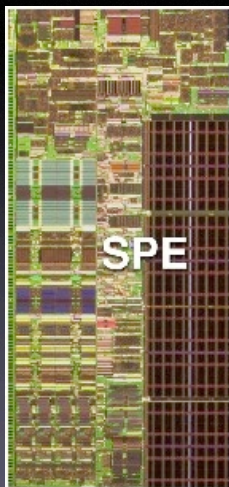

Power Processor Element
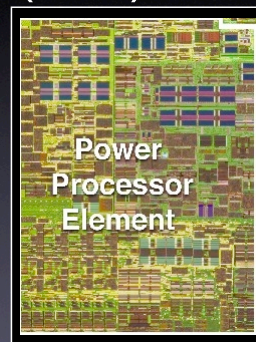
Synergistic Processor Element (SPE)

- **SPE provides computational performance**

  - Dual issue, up to 16-way 128-bit SIMD

  - Dedicated resources: 128 128-bit RF, 256 KB Local Store

  - Each can be dynamically configured to protect resources

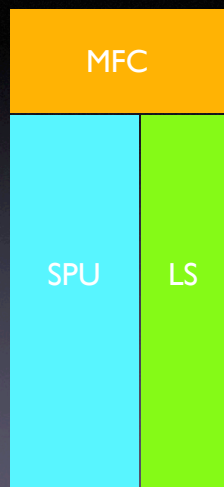 - Dedicated DMA engine: Up to 16 outstanding request



SPE
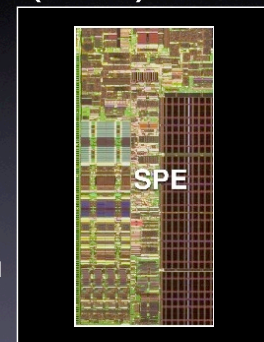
SPE

- Memory Flow Controller (MFC)
- Synergistic Processing Unit (SPU)
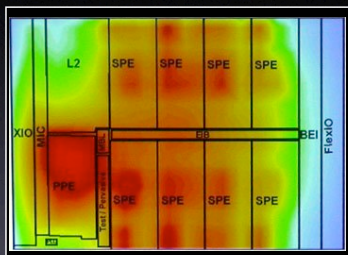- Local Store (LS) 256KB

MFC

SPU | LS

I/O and Memory Interfaces

- **I/O Provides wide bandwidth**
  - Dual XDR controller (25.6 GB/s @ 3.2Gbps
  - Two configurable interfaces (76.8 GB/s @ 6.4 bps)
  - Flexible Bandwidth between interfaces
  - Allows for multiple system configurations

Memory Controller
Rambus XDRAM™ Interface

Rambus FlexIO™
I/O Controller
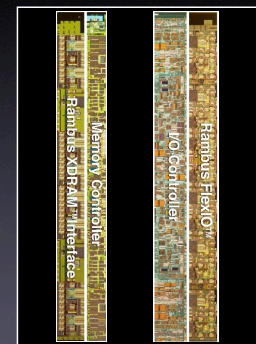
Power Management

- Dynamic Power Management (DPM)
- Five Power Management States
- One linear sensor
- Ten digital thermal sensors

I/O and Memory Interfaces

- **EIB data ring for internal communication**
  - Four 16 byte data rings, supporting multiple transfers
  - 96 B/cycle peak bandwidth
  - Over 100 outstanding requests
- **2 data rings are parallel in cycle**
  - each data ring can transfer max. 3 x 16 byte

Element Interconnect Bus
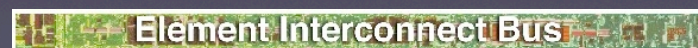
SPU Programming Model

- **Virtual File System to externalize the SPUs**
  - named spufs
  - mounted on /spu
- **Each spufs refers to a logical SPU**
  - mem: local store memory
  - run: starts execution of SPU code
  - mbox, ibox, wbox: abstractions for the userspace side
  - reg: register of the spu

Programming on the PPE

- Normal CPU programming
- AltiVec used as Single Instruction Multiple Data (SIMD) architectur
- AltiVec can calculate 8 elements per vector

C/C++ Intrinsics

```
void main(){
    reg regC;

    regC = vec_sums(regA, regB)

}
```

- saturating math
- sum accross
- log, power
- ceil / floor
- pixel vectors