Cell

Cell Broadband Engine Processor

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

- Outstanding performance, especially on game/multimedia applications.
  - Challenges: Power Wall, Frequency Wall, Memory Wall
- Real time responsiveness to the user and the network.
  - Challenges: Real-time in an SMP environment, Security
- Applicable to a wide range of platforms.
  - Challenge: Maintain programmability while increasing performance
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

Increase Concurrency

Increase Serialization

Micro38 Keynote
Cell Concept

- **Compatibility with 64b Power Architecture™**
  - Builds on and leverages IBM investment and community
- **Increased efficiency and performance**
  - Non Homogenous Coherent Chip Multiprocessor
    - Allows an attack on the “Frequency Wall”
  - Streaming DMA architecture attacks “Memory Wall”
  - High design frequency, low operating voltage attacks “Power Wall”
  - Highly optimized implementation
- **Interface between user and networked world**
  - Flexibility and security
  - Multi-OS support, including RTOS/non-RTOS
  - Architectural extensions for real-time management
Key Attributes

- High design frequency -> low voltage and low power
- Power architecture compatibility to utilize IBM software infrastructure & experiences
- SPE: SIMD architecture. Support media/game applications
- A power & area efficient PPE
Cell Architecture is ...

64b Power Architecture™

Memory

Power ISA

MMU/BIU

... 

Power ISA

MMU/BIU

COHERENT BUS

IO transl.

Incl. coherence/memory

compatible with 32/64b Power Arch. Applications and OS’s
Cell Architecture is … 64b Power Architecture™

Flow Control (MFC)

Memory

Plus

Memory

LS Alias

LS Alias

Power ISA +RMT

MMU/BIU +RMT

COHERENT BUS (+RAG)

MMU/DMA +RMT

Local Store Memory

MMU/DMA +RMT

Local Store Memory

IO transl.
Cell Architecture is … 64b Power Architecture™+ MFC

Plus

Synergistic

Processors

Memory

LS Alias

LS Alias

COHERENT BUS (+RAG)

IO transl.

Syn. Proc. ISA

MMU/DMA +RMT

Local Store Memory

Syn. Proc. ISA

MMU/DMA +RMT

Local Store Memory

Power ISA +RMT

MMU/BIU +RMT

Power ISA +RMT

MMU/BIU +RMT
Coherent Offload

- 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)
PPE Pipeline

PPE pipeline front end

Instruction cache and buffer

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

Microcode

MC1 → MC2 → MC3 → MC4 → ... → MC9 → MC10 → MC11

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
SPE Major Units

Floating-point unit
Fixed-point unit

Permute unit
Load/store unit
Branch unit
Channel unit

Result forwarding and staging
Register file

Instruction issue unit/instruction line buffer
On-chip coherent bus

Local store (256 KB)
Single-port SRAM

128R read
128R write

DMA unit

14.5mm² (90nm SOI)
SPE Pipeline

SPE pipeline front end
IF1 → IF2 → IF3 → IF4 → IF5 → IB1 → IB2 → ID1 → ID2 → ID3 → IS1 → IS2

SPE pipeline back end
RF1 → RF2
Branch instruction
Permute instruction
Load/store instruction
Fixed-point instruction
Floating-point instruction

IF  Instruction fetch
IB  Instruction buffer
ID  Instruction decode
IS  Instruction issue
RF  Register file access
EX  Execution
WB  Write back
• Please send your partner’s name

• Hw#4 demo and initial discuss for HW#5
  – Next Tuesday 4:00-5:30 at States Lab
  – Sing up a slot 15 min per each group

• Hw#5
  – Progress report: April 14th
  – Progress design review: April 14th afternoon
Admin 4/7/09

- HW#4&HW#5 meeting schedule
  - Hw#4: image size 6400x6400 (sight changes to make it multiple of your block size)
- Brining whatever your progress and F&Q section.

Initial design decisions

11:00-11:15  Steve, Rohit
4:15 – 4:30 : Bennet, Brandom
4:30 -4:45 Dan, Will
4:45 -5:00 Jacob, Dilan
5:00 -5:15 Jaewong, Vitaly
6:00 – 6:15 Tyler & Arias
Midterm-II

- AVG: 61.6/70
- STD: 5
- Median: 61/70
PPE (POWER PROCESSOR ELEMENT)
PPE Major Units

- L2 interface
- IU
  - Fetch control
  - Branch scan
- Pre-decode
  - L1 instruction cache
  - Thread A
  - Thread B
- SMT dispatch (queue)
- Threads alternate fetch and dispatch cycles
- Microcode
  - Decode
  - Dependency
  - Issue
  - Thread A
  - Thread B
  - Thread A
- VSU
  - VMX/FPU issue (queue)
  - VMX load/store/permute
  - VMX arith./logic unit
  - FPU arith./logic unit
  - FPU load/store
  - VMX completion
  - FPU completion
- XU
- Load/store unit
- Fixed-point unit
- Branch execution unit
- Completion/flush

(a)
PPE

- Pipeline depth: 23 stages
- Dual in-order issue
- 2way - SMT (issue 2 instructions from 2 threads)
- 1st level: 32KB 2nd level: 512KB
- Cache optimization:
  - Set locking, no write (reduce pollution) feature
PPE Pipeline

PPE pipeline front end

Instruction cache and buffer

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

Microcode

MC1 → MC2 → MC3 → MC4 → ... → MC9 → MC10 → MC11

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
PPE

• IU (Instruction unit): instruction fetch, decode, branch, issue and completion
  – Fetch 4 instructions per cycle per thread
  – 4KB branch predictor (global + local )
  – XU (Fixed point unit)

• VSU (A vector scalar unit): vector scalar and floating point
SPE (SYNERGISTIC PROCESSOR ELEMENTS)
SPE Major Units

- Floating-point unit
- Fixed-point unit
- Permute unit
- Load/store unit
- Branch unit
- Channel unit
- Result forwarding and staging
- Register file
- Instruction issue unit/instruction line buffer
- On-chip coherent bus
- Local store (256 KB)
- Single-port SRAM
- 128R read
- 128R write
- DMA unit

14.5mm² (90nm SOI)
Load and Store in SPE

- Local store is a private memory
- Load/store instruction to read or write
- DMW (Direct Memory Access) unit transfers data between local store and system memory
SPU Core

- SIMD RISC-style 32 bit fixed length instruction
- 2-issue core (static scheduling)
- 128 General purpose registers (both floating points, integers)
- Most instructions operates on 128bit wide data (2 x 64-bit, 4 x 32-bit, 8 x 16-bit, 1638-bit, and 128x1-bit)
- Operations: single precision floating point, integer arithmetic, logical, loads, stores, compares and branches
- 256KB of private memory
Even Pipe & Odd Pipe

Static scheduling:
- Fetch 2 instructions
- Check whether it can be done in parallel or not
- If not execute in-order
Memory Space

- No O/S on SPE
- Only user mode
- Fixed delay and without exception, greatly simplifying the core design
DMA Engine

• Transfers are divided into 128 Bytes packets for the on chip interconnect
• Typical 128B requires 16 processor cycles
• Instruction fetch 128B (reduce the pressure to DMA)
• DMA priority
  – Commands (high) $\rightarrow$ loads/stores $\rightarrow$ instruction (prefetch)
  – Special instruction to force instruction fetch
Branch

• Compiler/programmer hint
  – An upcoming branch address and branch target, prefetching at least 17 instructions

• 3-source bitwise selection instruction to eliminate branch (similar to predication)
On-chip network

- Rambus XDR
- 12.8 GB/s per 32-bit memory channel (x2)
- High bandwidth support between cell processors
- IOIF: Input–output interface; BIF: broadband interface
XBox360 vs. PS3

• 360 Hardware:
  2. All games supported at 16:9, 720p and 1080i, anti-aliasing
  3. Customizable face plates to change appearance
  4. 3 USB 2.0 ports
  5. Support for 4 wireless controllers
  6. Detachable drive
  7. Wi-Fi ready

• Custom IBM PowerPC-based CPU
  - 3 symmetrical cores at 3.2 GHz each
  - 2 hardware threads per core
  - 1 VMX-128 vector unit per core
  - 1 MB L2 cache

CPU Game Math Performance
- 9 billion dots per second

http://www.ps3vault.com/ps3-specifications/ps3-vs-xbox-360
XBox360 vs. PS3

- **Custom ATI Graphics Processor**
  - 500 MHz
  - 10 MB embedded DRAM
  - 48-way parallel floating-point shader pipelines
  - unified shader architecture

- **Memory**
  - 512 MB GDDR3 RAM
  - 700 MHz DDR

- **Memory Bandwidth**
  - 22.4 GB/s memory interface bus bandwidth
  - 256 GB/s memory bandwidth to EDRAM
  - 21.6 GB/s frontside bus

- **Audio**
  - Multichannel surround sound output
  - Supports 48khz 16-bit audio
  - 320 independent decompression channels
  - 32 bit processing
  - 256+ audio channels

- **Games:** Over 100 games available. Marquee games include Gears of War, Tom Clancy line of games, Call of Duty 3, and F.E.A.R.

http://www.ps3vault.com/ps3-specifications/ps3-vs-xbox-360
PS3 Specification

**CPU:** Cell Processor PowerPC-base Core @3.2GHz  
1 VMX vector unit per core  
512KB L2 cache  
7 x SPE @3.2GHz  
7 x 128b 128 SIMD GPRs  
7 x 256KB SRAM for SPE  
*1 of 8 SPEs reserved for redundancy  
Total floating point performance: 218 gigaflops

**GPU RSX** @ 550MHz  
1.8 TFLOPS floating point Performance  
Full HD (up to 1080p) x 2 channels  
Multi-way programmable parallel Floating point shader pipelines  
Sound Dolby 5.1ch, DTS, LPCM, etc. (Cell-based processing)

**Memory**  
256MB XDR Main RAM @3.2GHz  
256MB GDDR3 VRAM @700MHz  
System Bandwidth Main RAM– 25.6GB/s  
VRAM– 22.4GB/s  
RSX– 20GB/s (write) + 15GB/s (read)  
SB2.5GB/s (write) + 2.5GB/s (read)
XBox360 vs. PS3

- **SYSTEM FLOATING POINT PERFORMANCE:**
  2 teraflops

- **STORAGE**
  HDD Detachable 2.5” HDD slot x 1
  I/O–USB Front x 4, Rear x 2 (USB2.0)
  Memory Stickstandard/Duo, PRO x 1
  SD standard/mini x 1
  CompactFlash(Type I, II) x 1

- **COMMUNICATION**
  Ethernet (10BASE-T, 100BASE-TX, 1000BASE-T) x 3 (input x 1 + output x 2)
  Wi-Fi IEEE 802.11 b/g (60gig only)
  Bluetooth–Bluetooth 2.0 (EDR) ControllerBluetooth (up to 7)
  USB 2.0 (wired)
  Wi-Fi (PSP) Network (over IP)

- **AV OUTPUT**
  Screen size 480i, 480p, 720p, 1080i, 1080p , HDMI out x 2, AV multi out x 1, Digital out (optical) x 1

- **DISC MEDIA**
  CD, DVD…

- **Games**
  Currently, there are about 10 games released.

http://www.ps3vault.com/ps3-specifications/ps3-vs-xbox-360
• P#2) Minimum Number of threads to get a peak GFLOPS
  – #Peak Warp w/o bw = Mem_L/(#comp_inst X issue cycles)
  – (assumption (mem latency incudes memory instruction execution cost.)
  – # Bandwidth peak warp = Mem bandwidth /(BW per warp)
  – BW per warp = Freq x (load bytes per warp)/Mem L
  – Compare #Peak Warp w/o bw., # Bandwidth peak warp and lower is the answer
Mid-term Questions

• P#4)
• Assumption: each warp fetches one instruction
• 512 threads per block, 8 blocks per SM
• $\frac{512}{32} \times 8 \times 16 = 2048$ times (across all SMs)
• L-cache hit ratio: $50 \times 4B << 16KB$
• Only compulsory miss regardless of how many blocks are running
  – $100 - \left(\frac{50}{50 \times 16 \times 8}\right) = 99.99\%$
  – 50: static inst 16: # warps 8 : #blocks per SM
HW#4

Extra points receiving group

Less than 1 sec

SPEEDUP

1  2  3  4  5  6  7  8  9  10  11

0  1  2  3  4  5  6  7  8  9  10
Progress Report: (Brining during the meeting)
- Top design diagram
- pipeline stages, I/O unit, memory, plan
- Xbox360 core is similar to PPE (cell)

Progress Design Review (KACB 2344)
4:00 – 4:15 Steve, Rohit
4:15 – 4:30 Bennet, Brandom
4:30 -4:45 Dan, Will
4:45 -5:00 Jacob, Dilan
5:00 -5:15 Jaewong, Vitaly
6:00 – 6:15 Tyler & Arias