Cell Processor and Playstation 3

Guillem Borrell i Nogueras

February 24, 2009

- Cell systems
- Bad news
- More bad news
- Good news
- Q&A

IBM Blades

- QS21
  - Cell BE based.
    - 8 SPE
  - 460 Gflops Float
  - 20 GFLops Double
- QS22
  - PowerXCell 8i based
    - 8 SPE*
  - 460 GFlops Float
  - 200 GFlops Double
  - Some of them already installed in BSC

IBM Blades

- QS21
  - Cell BE based.
    - 8 SPE
  - 460 Gflops Float
  - 20 GFLops Double
- QS22
  - PowerXCell 8i based
    - 8 SPE*
  - 460 GFlops Float
  - 200 GFlops Double
  - Some of them already installed in BSC
  - No SPU Double precision improvements expected from IBM

Playstation 3

- Cell BE based.
    - 6 SPE
- 460 Gflops Float
- 20 GFLops Double

Playstation 3

- Cell BE based.
  - 6 SPE
- 460 Gflops Float
- 20 GFLops Double
- 256 MB RAM

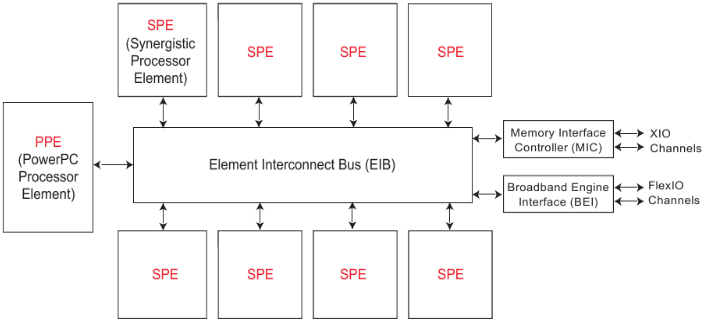
IBM Power 7

- 8 PowerPC cores
- 4 threads per core (32 Threads!)
- ? SPE

IBM Power 7

- 8 PowerPC cores
- 4 threads per core (32 Threads!)
- ? SPE
- 1 TFlop on a chip

Cell Broadband Engine



PPE	PowerPC Processor Element
PPU	PowerPC Processor Unit
EIB	Element Interconnect Bus
SPE	Synergistic Processor Element
SPU	Synergistic Processor Unit
MFC	Memory Flow Controller
DMA	Direct Memory Access
SIMD	Single Instruction Multiple Data

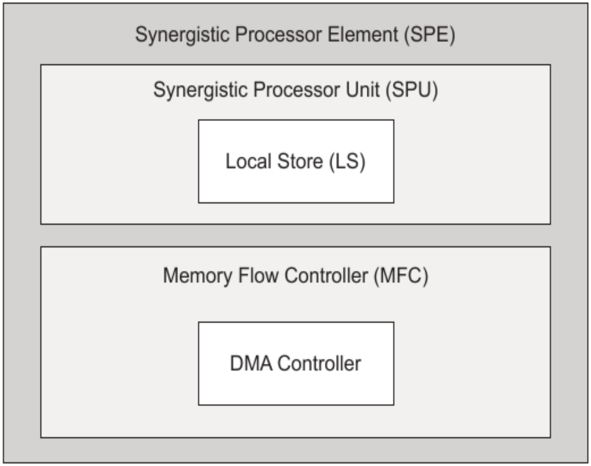
¿How does it compute?

- PPU starts a program
- PPU loads an SPU context on a thread
- SPU acquires the thread
- SPU executes context
- SPU ends the task and returns control to PPU

¿How does it compute?

- PPU starts a program
- PPU loads an SPU context on a thread
- SPU acquires the thread
- SPU executes context
- SPU ends the task and returns control to PPU
- I have said nothing about the data

Overview


Synergistic Processor Element (SPE)
Synergistic Processor Unit (SPU)
Local Store (LS)
Memory Flow Controller (MFC)
DMA Controller

Dumb vector unit

- General purpose vector unit
  - Designed to run compiled code. A context is a program.
  - Altivec unit on a box
  - IBM called that VMX

Dumb vector unit

- General purpose vector unit
  - Designed to run compiled code. A context is a program.
  - Altivec unit on a box
  - IBM called that VMX
- LS is regiser based
  - No type distinction
  - Data should be aligned by hand

Dumb vector unit

- General purpose vector unit
  - Designed to run compiled code. A context is a program.
  - Altivec unit on a box
  - IBM called that VMX
- LS is regiser based
  - No type distinction
  - Data should be aligned by hand
- MFC is a DMA controller
  - Data moved with DMA primitives.
  - No data scheduling
  - No data implicit copying.

SIMD Programming

- SPUs are programmed using SIMD primitives
- Like a vector unit

SIMD Programming

- SPUs are programmed using SIMD primitives
- Like a vector unit
- Almost coding in assembly
    - Access assembly instructions via libspu2
    - Add to that DMA instructions

SIMD Programming

- SPUs are programmed using SIMD primitives
- Like a vector unit
- Almost coding in assembly
    - Access assembly instructions via libspu2
    - Add to that DMA instructions
    - That can take us ages

PPU and SPU code

- PPE have L1 and L2 cache.
- SPE have LS (register based)

PPU and SPU code

- PPE have L1 and L2 cache.
- SPE have LS (register based)
- Their ssembly has nothing to do
  - They are compiled separately.
  - PPU code cannot be reused.

And now the good news.

PPU code libraries

- BLAS (Basic Linear Algebra Subroutines)
- LAPACK (Linear Algebra Package)
- FFTW (The Fastest Fourier Transform in the West)
- C and Fortran interfaces
- Fortran interface is not complete

PPU code libraries

- BLAS (Basic Linear Algebra Subroutines)
- LAPACK (Linear Algebra Package)
- FFTW (The Fastest Fourier Transform in the West)
- C and Fortran interfaces
- Fortran interface is not complete
- Almost all we need is in Cell SDK!

Thin ice

- PPU code means no SPU control.
- Data must be aligned too using `posix_memalign`.
- If SPU control is needed PPU code cannot be used at all
- Tells us what we can or cannot do
- BSC has been using those for about 2 years.

Optimizing compilers

- Cell Superscalar
  - Alpha state
  - OpenMP-like pragmas for SIMD
  - BSC
  - Free Software
- XL compilers for Multicore Acceleration
  - Alpha state
  - OpenMP support for SIMD
  - MASS (Mathematical Acceleration Subsystem)
  - Worth the money

Conclusions

- PPU code is possible

Conclusions

- PPU code is possible
- SPU code is not possible *for us*
- 2 options:
  - Cell SDK for Multicore Acceleration
  - Optimizing Compiler (WAIT)

Conclusions

- PPU code is possible
- SPU code is not possible *for us*
- 2 options:
  - Cell SDK for Multicore Acceleration
  - Optimizing Compiler (WAIT)
- Get a good C book.

Q&A