Accurate and High Performance Computing on the Cell processor

Stef Graillat

LIP6/PEQUAN - Université Pierre et Marie Curie (Paris 6)

Young Investigators Symposium, Oak Ridge National Laboratory, Tennessee, USA, October 13-15, 2008

Can you count up to 6 with your computer ?

$2-1$		1.000000000000000
$\left(\frac{1}{\cos(100\pi+\pi/4)}\right)^2$		2.000000000000011
$3\frac{\cos(\arccos(10000))}{10000}$		2.999999997414701
$\left(\left(\cdots\left(\sqrt{\sqrt{\cdots\sqrt{4}}}\right)^2\cdots\right)^2\right)^2$	(20 times)	4.000000000629434
$5\times\left\{\frac{(1+e^{-100})-1}{(1+e^{-100})-1}\right\}$		`NaN`
$\frac{\log(e^{6000})}{1000}$		`Inf`

Ariane 5 rocket failure

Conversion of the horizontal speed from a 64-bit floating point number to a 16-bit signed integer → overflow !

Other famous failures !

- Vancouver Stock Exchange : introduction in 1982 of a new index, with initial value of 1000.00 recomputed after each transaction and then truncated to 3 digits (e.g. 556.56 → 556)
  22 months later : 520 where the correct value was 1098
- German election in Schleswig-Holstein, April 5th, 1992, the print of the pourcentage of votes of the Green party with only one place after the decimal changed the result and gave the majority to SPD in the parlement (4.97% was printed as 5.0% after rounding).

Floating point number

Floating point system $\mathbb{F} \subset \mathbb{R}$:

$$x = \pm \underbrace{x_0.x_1 \dots x_{p-1}}_{mantissa} \times b^e, \quad 0 \le x_i \le b-1, \quad x_0 \neq 0$$

b : basis, p : precision, e : exponent range s.t. $e_{\min} \le e \le e_{\max}$

Machine epsilon $\epsilon = b^{1-p}$

Approximation of $\mathbb{R}$ by $\mathbb{F}$, rounding $\mathrm{fl} : \mathbb{R} \to \mathbb{F}$
Let $x \in \mathbb{R}$ then

$$\mathrm{fl}(x) = x(1+\delta), \quad |\delta| \le \mathbf{u}.$$

Unit roundoff $\mathbf{u} = \epsilon/2$ for round-to-nearest

Standard model of floating point arithmetic

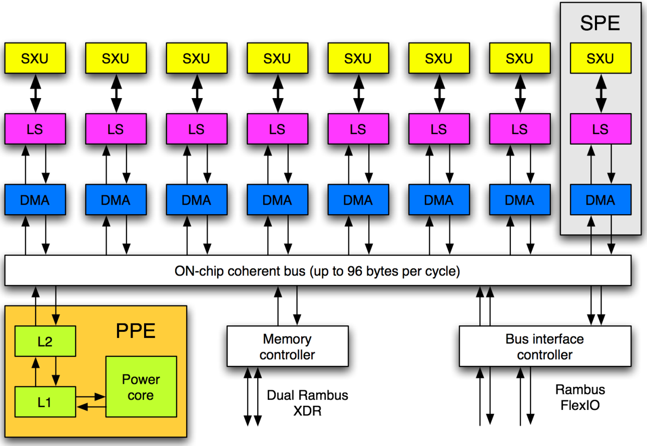
Let $x, y \in \mathbb{F}$,

$$\mathrm{fl}(x \circ y) = (x \circ y)(1+\delta), \quad |\delta| \le \mathbf{u}, \quad \circ \in \{+,-,\cdot,/\}$$

IEEE 754 standard (1985)

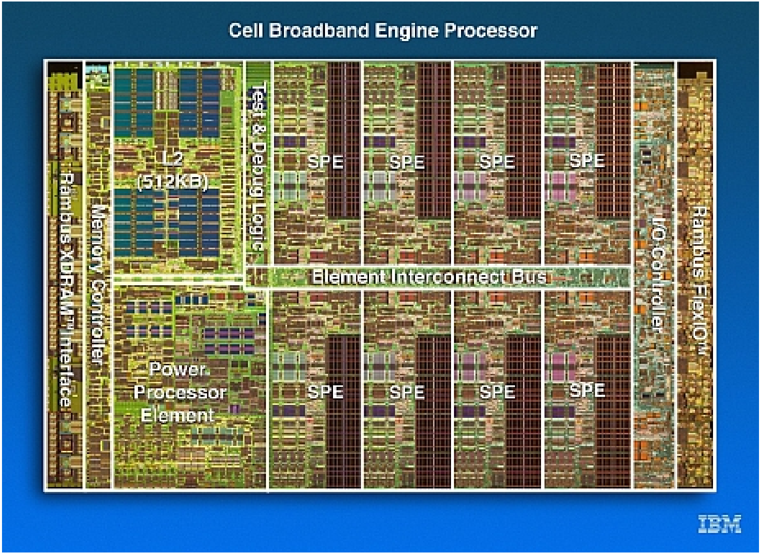
Type	Size	Mantissa	Exponent	Unit roundoff	Range
Single	32 bits	23+1 bits	8 bits	$\mathbf{u} = 2^{-24} \approx 5,96 \times 10^{-8}$	$\approx 10^{\pm 38}$
Double	64 bits	52+1 bits	11 bits	$\mathbf{u} = 2^{-53} \approx 1,11 \times 10^{-16}$	$\approx 10^{\pm 308}$

The Cell processor (1/2)



SP > 200 GFlops, DP=15 Gflops, 25GB/s memory BW, 300 GB/s EIB

The Cell processor (2/2)



Power Processor Element (PPE)

The PPE is based on the 2-way Power Architecture with :

- 32 KB of L1 cache for instructions
- 32 KB of L1 cache for data
- 512 KB of L2 cache

The PPE is fully pipelined for double precision computation and fully IEEE compliant.

Synergistic Processing Element SPE (1/2)

The SPE is a small processor with a vectorial unit.

- small memory (256 KB) for instructions and data, named "local store" (LS)
- 128 registers of 128 bits
- 1 SPU "Synergistic Processing Unit"
    - 4 units for single precision computation
    - 1 unit for double precision computation
- MFC "Memory Flow Controller" which manages memory access through DMA

Synergistic Processing Element SPE (2/2)

128-bit registers :

- 16 integers of 8-bits,
- 8 integers of 16-bits,
- 4 integers of 32-bits,
- 4 single precision floating point numbers,
- 2 double precision floating point numbers.

The SIMD processor is based on FMA and is fully pipelined in SP :

$$\text{Peak performance SP} : 4 \times 2 \times 3.2 = 25.6 GFLOPs$$

Not fully pipelined in double precision :

$$\text{Peak performance in DP} : 2 \times 2 \times 3.2/7 = 1.8 GFLOPs$$

The parallel programming

3 levels of parallelism

1. Processes run on Cell processors, exchange with a MPI library
2. Data distribution and communication between PPE and SPE :
    - ALF
    - mailing box
    - exchange through DMA
    - data need to be aligned on quadword
    - double buffering technique
3. on an SPE
    - only 256 KB
    - Altivec programming
    - code and data dependencies : not to break the SIMD pipeline

The performance price on the SPE (1/2)

No division
$1/x$ and $1/\sqrt{x}$: only the 12 first bits are exact.
SPU float arithmetic is not IEEE compliant :

- only rounding mode to zero (truncation).
- The highest exponent (128) is used not for Infinity or NaN, but is used to extend the range of the floating point.
- Inf and NaN are not recognized by arithmetic operations.
- Overflow results saturate to the largest representable positive or negative values, rather than producing +/-IEEE Infinity.
- No denormalized results : +0 instead.

The performance price on the SPE (2/2)

SPU double arithmetic is IEEE compliant except :

- FP trapping is not supported.
- Denormalized operands are treated as 0.
- NaN results are always the default QNaN (Quiet NaN)

High Precision Library on the Cell processor

Definition 1

An extended precision number of n is a non-evaluated sum of n floating point numbers $x = x_1 + x_2 + \cdots + x_n$

Precision used on Cell processor : single precision

- $n = 2$: single-single
- $n = 4$: quad-single

$\Rightarrow$ achieve a 64-bits and 128-bits precision while working with the fast single precision SIMD units.

Reliable Computing on the Cell processor

Intervals

$$\boldsymbol{x} = [\underline{x}; \overline{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \overline{x}\}.$$

Given 2 intervals $\boldsymbol{x}$, $\boldsymbol{y}$ and $\diamond \in \{+, -, \times, /\}$, one can define

$$\boldsymbol{x} \diamond \boldsymbol{y} = \{x \diamond y : x \in \boldsymbol{x}, y \in \boldsymbol{y}\}.$$

This can be computed

$$\begin{array}{rcl} \boldsymbol{x} + \boldsymbol{y} & = & [\underline{x} + \underline{y}; \overline{x} + \overline{y}], \\ \boldsymbol{x} \times \boldsymbol{y} & = & [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}; \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}]. \end{array}$$

In floating point arithematic, there are rounding errors !

$$\begin{array}{rcl} \boldsymbol{x} + \boldsymbol{y} & = & [\nabla(\underline{x} + \underline{y}), \Delta(\overline{x} + \overline{y})] \supseteq \{x + y | x \in X, y \in Y\} \\ \boldsymbol{x} - \boldsymbol{y} & = & [\nabla(\underline{x} - \overline{y}), \Delta(\overline{x} - \underline{y})] \supseteq \{x - y | x \in X, y \in Y\} \end{array}$$

where ∇ (resp. Δ) represents rounding toward $-\infty$ (resp. rounding toward $+\infty$).

Looking for collaboration !

We are looking for

- people needing accurate and reliable high performance computing for real-life applications
- people with high performance computing skills to improve the performance of our libraries

Thank you for your attention