Interval arithmetic on the Cell processor

Stef Graillat Jean-Luc Lamotte Siegfried M. Rump
Svetoslav Markov

LIP6/PEQUAN, P. and M. Curie University, Paris

Institute for Reliable Computing, Hamburg University of Technology

Institute of Mathematics and Computer Science, Bulgarian. Academy of Sciences

13th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Verified Numerical Computations SCAN'08

El Paso, Texas, USA, September 29 - October 3, 2008

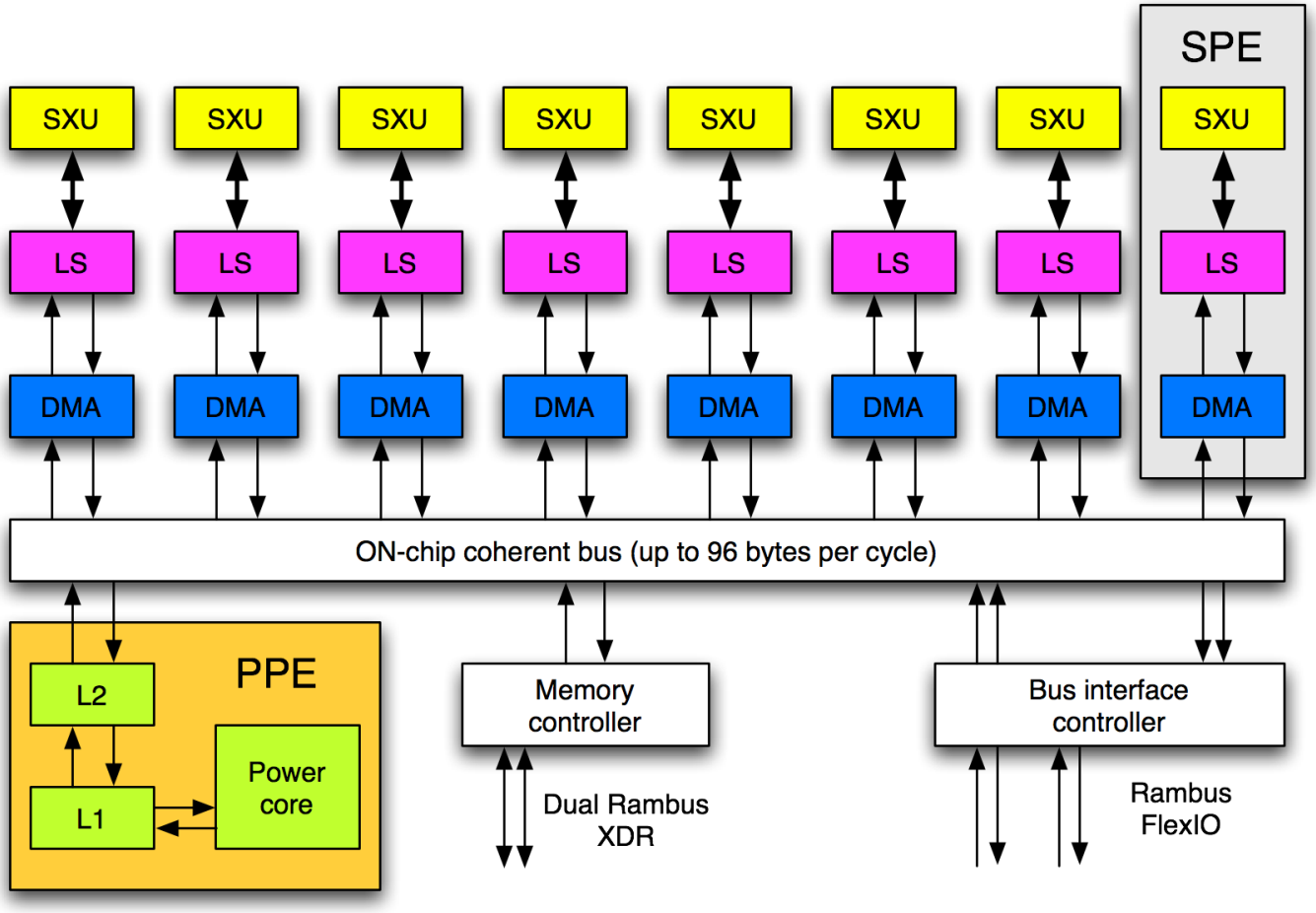
Overview

The Cell processor

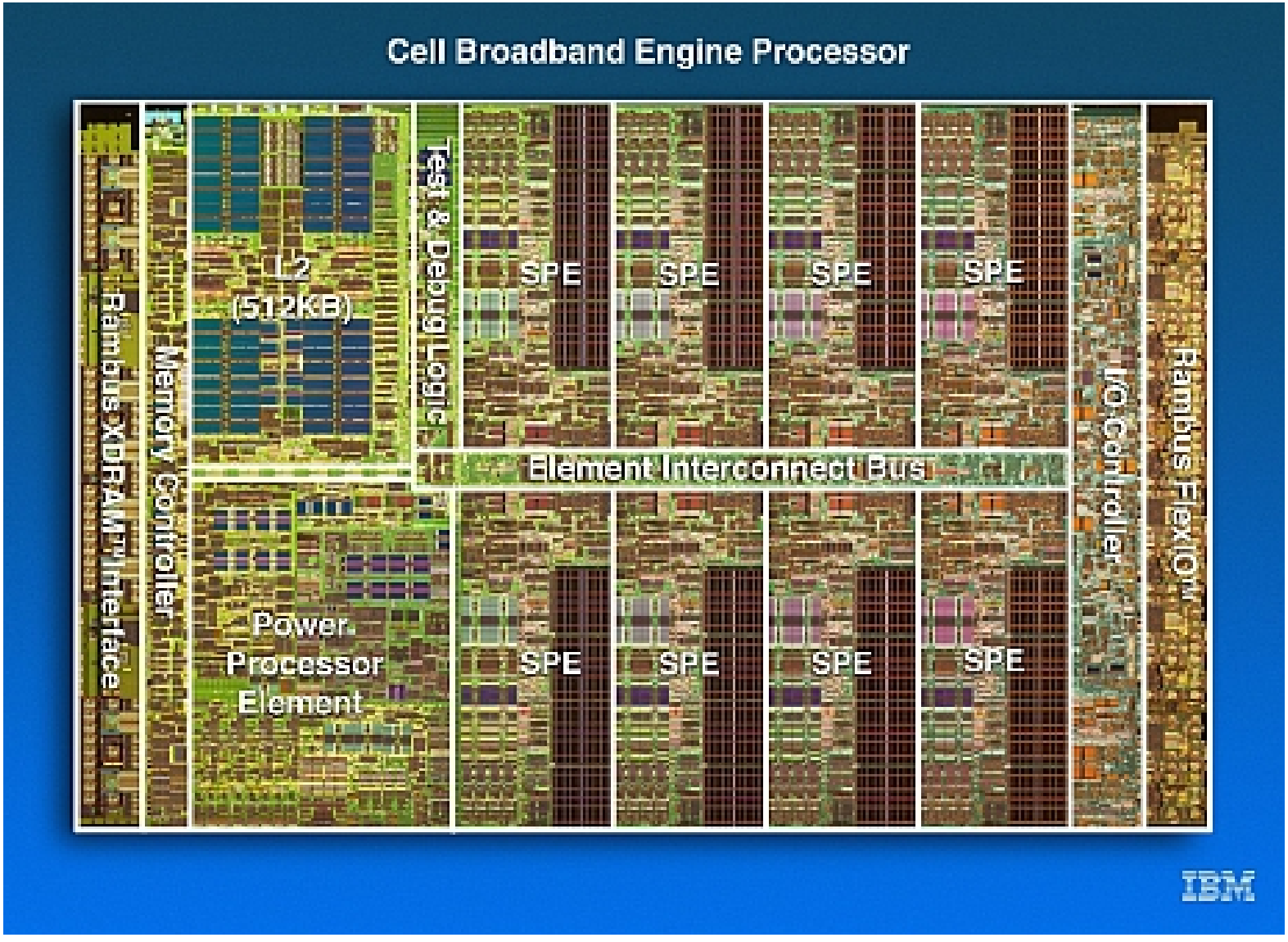
Interval Arithmetic on Cell processor

Conclusions

The Cell processor



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


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

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 :

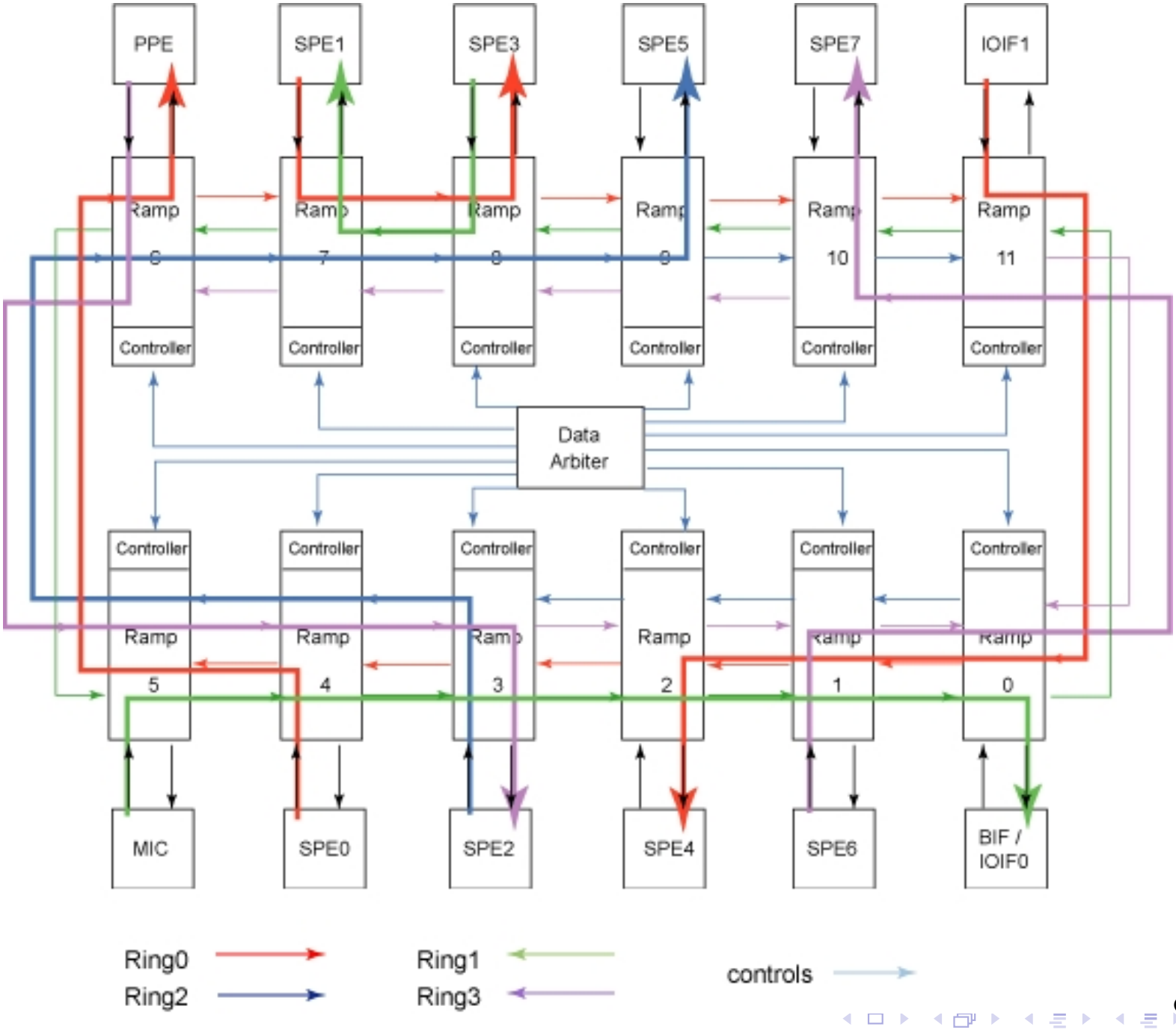
- 16 integers of 8 bit,
- 8 integers of 16 bit,
- 4 integers of 32 bit,
- 4 single precision floating point numbers,
- 2 double precision floating point numbers.

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 :

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


PPE
SPE1
SPE3
SPE5
SPE7
IOIF1
Ramp 6
Ramp 7
Ramp 8
Ramp 9
Ramp 10
Ramp 11
Controller
Data Arbiter
Ramp 5
Ramp 4
Ramp 3
Ramp 2
Ramp 1
Ramp 0
MIC
SPE0
SPE2
SPE4
SPE6
BIF / IOIF0
Ring0
Ring1
Ring2
Ring3
controls

Parallelism on Cell

3 levels of parallelism :

1. processes run on Cell processors, exchange with a MPI library,
2. Data distribution and communication on the 8 SPE,
   - ALF, Dacs
   - POSX thread, CELL thread,
   - mailing box, exchange through DMA
   - data need to be aligned on quadword
   - double buffering technique
3. inside a thread
   - only 256 KB
   - Altivec programming
   - code and data dependencies : not to break the SIMD pipeline

The performance price on SPE

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

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)

Reliable computing on Cell processor

- difficult to implement interval arithmetic.
- possible to "emulate" a rounding mode toward $+\infty$
  if $r \in \mathbb{R}$ non-negative, $\mathrm{fl}_0(r) \leq r \leq \mathrm{succ}(\mathrm{fl}_0(r))$
  and

$$\mathrm{succ}(f) = \max\{\mathrm{fl}_0((1+2\mathbf{u})f), \mathrm{fl}_0(f+\underline{\mathbf{u}})\}.$$

  where $\mathbf{u}$ is the relative rounding error and $\underline{\mathbf{u}}$ the underflow unit

On the Cell processor, no underflow

- $\mathrm{succ}(f) = \mathrm{fl}_0((1+2\mathbf{u})f)$ if $f > 0$
- $\frac{1}{2}\underline{\mathbf{u}}\mathbf{u}^{-1}$ if $f = 0$

Interval with a rounding mode toward zero

Three representations :

- endpoint
- center-radius
- leftpoint-diameter

Endpoint representation — addition

Let $A = [a_{inf}, a_{sup}]$, $B = [b_{inf}, b_{sup}]$ and $C = [c_{inf}, c_{sup}]$ be three intervals, $C = A + B$ is defined by :
Let $\oplus$ and $\otimes$ be the floating point addition and multiplication with rounding towoard zero.

$$c_{inf} = \begin{cases} -\operatorname{succ}(|a_{inf} \oplus b_{inf}|) & \text{if } (a_{inf} \oplus b_{inf}) < 0 \\ a_{inf} \oplus b_{inf} & \text{if } (a_{inf} \oplus b_{inf}) > 0 \\ -\frac{1}{2}\underline{\mathbf{u}}\mathbf{u}^{-1} & \text{if } (a_{inf} \oplus b_{inf}) = 0 \end{cases}$$

$$c_{sup} = \begin{cases} \operatorname{succ}(a_{sup} \oplus b_{sup}) & \text{if } (a_{sup} \oplus b_{sup}) > 0 \\ a_{inf} \oplus b_{inf} & \text{if } (a_{sup} \oplus b_{sup}) < 0 \\ \frac{1}{2}\underline{\mathbf{u}}\mathbf{u}^{-1} & \text{if } (a_{sup} \oplus b_{sup}) = 0 \end{cases}$$

Endpoint representation — multiplication

$$x = min(a_{inf} \otimes b_{inf}, a_{inf} \otimes b_{sup}, a_{sup} \otimes b_{inf}, a_{sup} \otimes b_{sup})$$
$$y = max(a_{inf} \otimes b_{inf}, a_{inf} \otimes b_{sup}, a_{sup} \otimes b_{inf}, a_{sup} \otimes b_{sup})$$

$C = A \times B$ is defined by :

$$c_{inf} = \begin{cases} -\,\mathrm{succ}(|x|) & \text{if } x < 0 \\ -\frac{1}{2}\underline{\mathbf{u}}\mathbf{u}^{-1} & \text{if } x = 0 \\ x & \text{else} \end{cases}$$

$$c_{sup} = \begin{cases} \mathrm{succ}(y) & \text{if } y > 0 \\ \frac{1}{2}\underline{\mathbf{u}}\mathbf{u}^{-1} & \text{if } y = 0 \\ y & \text{if } y < 0 \end{cases}$$

Center-radius — addition

Let $A = [a, \alpha]$, $B = [b, \beta]$ and $C = [c, \gamma]$ be three intervals.

Rump's algorithm :

$$\begin{aligned} c &= \square(a + b) \\ \gamma &= \triangle(2\mathbf{u} \cdot |c| + \alpha + \beta) \end{aligned}$$

$C = A + B$ is defined by :

- $c = a \oplus b$
- $\gamma = \mathrm{succ}(2u \otimes |c| \oplus \mathrm{succ}(\alpha \oplus \beta))$

Center-radius — multiplication

Rump's algorithm :

$$\begin{aligned} c &= \square(a \cdot b) \\ \gamma &= \triangle(\underline{u} + 2\mathbf{u} \cdot |c| + (|a| + \alpha)\beta + \alpha|b|)) \end{aligned}$$

$C = A \times B$ is defined by :

- $c = a \otimes b$
- $\gamma = \mathrm{succ}(\mathrm{succ}(2u \otimes |c| \oplus \mathrm{succ}(\mathrm{succ}(|a| \oplus \alpha) \otimes \beta)) \oplus \mathrm{succ}(\alpha \otimes |b|)$

Implementation of intervals vectors

```
typedef struct{
  float center;
  float radius;
} T_INTERVAL;

T_INTERVAL x[...]
```

c1	r1	c2	r2	c3	r3	c4	r4

```
typedef struct{
  int intervalnumber;
  float *center;
  float *radius;
} T_INTERVAL;
```

c1	c2	c3	c4
r1	r2	r3	r4

All the SIMD operations use vectors of four 32-bit floating point numbers.

Performances on 1 SPE

Operations	MFLOPs
Idle (function call)	477.6
Add [crcr]	273.5
Add [ccrr]	345.9
Mul [crcr]	244.3
Mul [ccrr]	255.1
Add inf-sup	285.7

Center-diameter representation

Seems to be useful with rounding toward zero !

$$A = (a, \alpha) = \begin{cases} \{x \in \mathbb{R} : a \leq x \leq a + \alpha\} & \text{if } a \geq 0 \\ \{x \in \mathbb{R} : a - \alpha \leq x \leq a\} & \text{if } a < 0 \end{cases}$$

But very difficult to implement !

Conclusions

- hard programming job
- necessity to develop complex algorithms to reach a high level of performance.
- to prepare the work for the new Cell :
  - fully pipelined double precision floating point number (100 DP GFlOPS)
  - up to now no information on the floating point quality

Rumours on the next generation

- IEEE compliant
- from 8 to 32 SPE
- over 1TFLOPS

Acknowledgment

Thanks to the CINES Center for providing IBM Cell Blade access.