

Lattice QCD on the Cell Processor

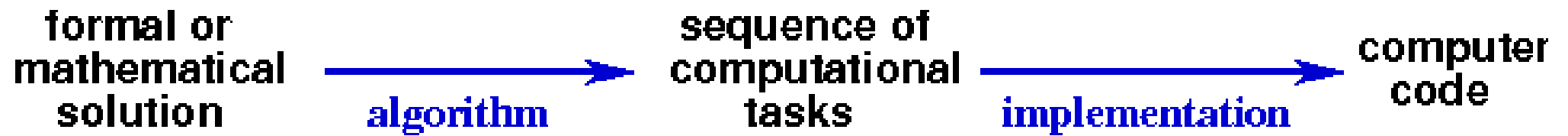
Hubert Simma

Università Milano Bicocca, DESY Zeuthen

Plan:

- Performance Modeling
- Micro-Benchmarks
- Application-Benchmarks
- Cell-based QCD Machine

Introduction



Performance modeling:

Estimate wall-clock time to solve a given task on a given architecture

$$\mathbf{T} = \mathbf{f} \left(\begin{array}{c} \text{algorithm} \\ \downarrow \\ \text{task} \end{array}, \begin{array}{c} \text{implementation} \\ \downarrow \\ \text{code} \end{array}, \text{machine} \right)$$

Motivation:

- predict/analyse machine performance
- gain info about hardware details
- guide optimisation strategies
- select/develop adequate algorithm

Naive Performance Model

Application Signature vs. Hardware Characteristics

$$\frac{\# \text{ theoretical FP operations}}{\# \text{ input} + \text{ output data}}$$

$$\approx 1.3 \text{ flop/B (LQCD 32-bit)}$$

$$\frac{\# \text{ remote data access}}{\# \text{ local data access}}$$

$$\frac{\text{peak flops}}{\text{memory bandwidth}}$$

$$8 \text{ flop/B (CBE 32-bit)}$$

$$\frac{\text{communication bandwidth}}{\text{memory bandwidth}}$$

Machine Elements

Hardware devices/units for:

- ❑ control (of data and program flow)

- ❑ storage of data (and code)
 - memory
 - cache(s)
 - registers
 - internal buffers, fifos, flip-flops, ...

- ❑ processing/transport of data
 - arithmetic pipelines
 - storage access (hopefully pipelined)
 - combinatorical logics
 - buses?

Machine Model

Structure:

graph with

- vertices = storage devices
- edges = data paths or transport devices

Parameters of storage devices:

σ_i = storage size

Data units: Byte, fword, cword, . . .

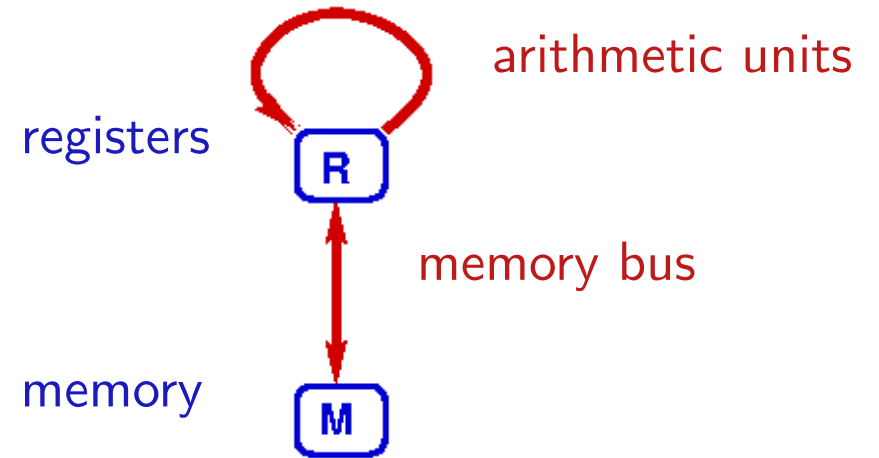
Parameters of transport devices:

ISA = instruction set architecture

β_i = bandwidth (data throughput/time)

λ_i = latency (delay between input and first output)

Time units: nsec, T_{clk} (clock cycle)



arbitration?

overlap?

Analysis of Implementations

Instruction Match

Mapping of tasks onto machine instructions (ISA) taking into account e.g.

- operand types (single vs. double, real vs. complex, . . .)
- granularity of operations (multiply/add vs. fused MulAdd)
- alignment constraints (vector or “SIMD” instructions)

Information Exchange

$$I_{xy}(N, \sigma_x) \equiv \text{data exchange for specific computational task of size } N \\ \text{between computer sub-systems } x \text{ and } y \text{ with storage } \sigma_x$$

where $x, y =$ registers (R), memory (M), cache (C), processors (P, P'), . . .

More explicit: For one or more implementations, i , compute separately

- $I_{xy}(N, i) \Rightarrow$ bandwidth model
- $S_x(N, i) \Rightarrow$ storage constraints: $S_x \leq \sigma_x$

N.B.: $I_{RR} \leftrightarrow$ number of arithmetic (FP) operations

Performance Estimate

Execution time of (micro-)tasks: (for a specific implementation)

$$T_{xy} \approx I_{xy}/\beta_{xy} + O(\lambda_{xy})$$

Total execution time:

$$\max T_{xy} \approx T_{exe}(i) \leq \sum T_{xy} + O(\lambda_{xy})$$

Assumptions: (w.r.t. task, implementation, and machine)

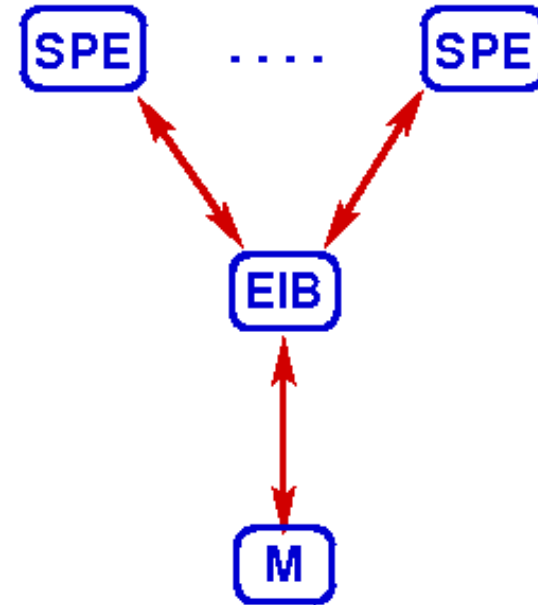
- tasks can be executed concurrently
- dependencies and latencies can be hidden (tends to increase S_x)

Limitations:

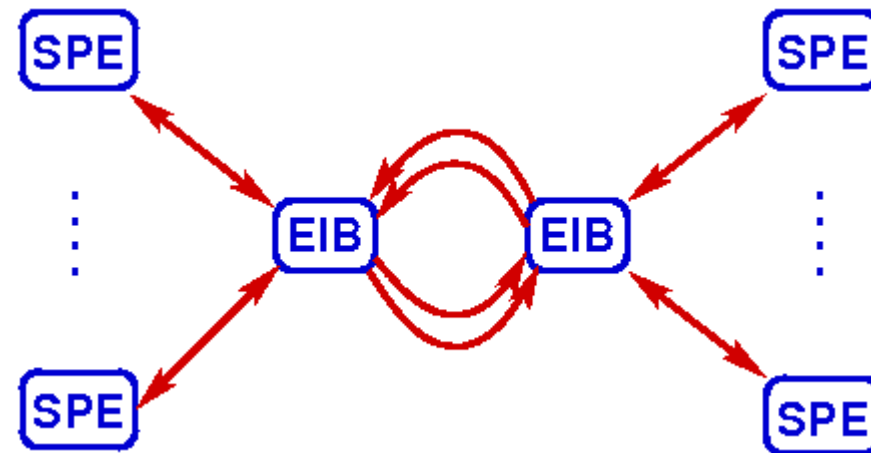
- context dependence of latencies
e.g. subsequent memory accesses: address dependence and partial overlap
- arbitration of transport devices
e.g. bi-directional or multiple bus

Different Models for EIB on the Cell

- Infinite bandwidth model:



- Finite bisection bandwidth model:

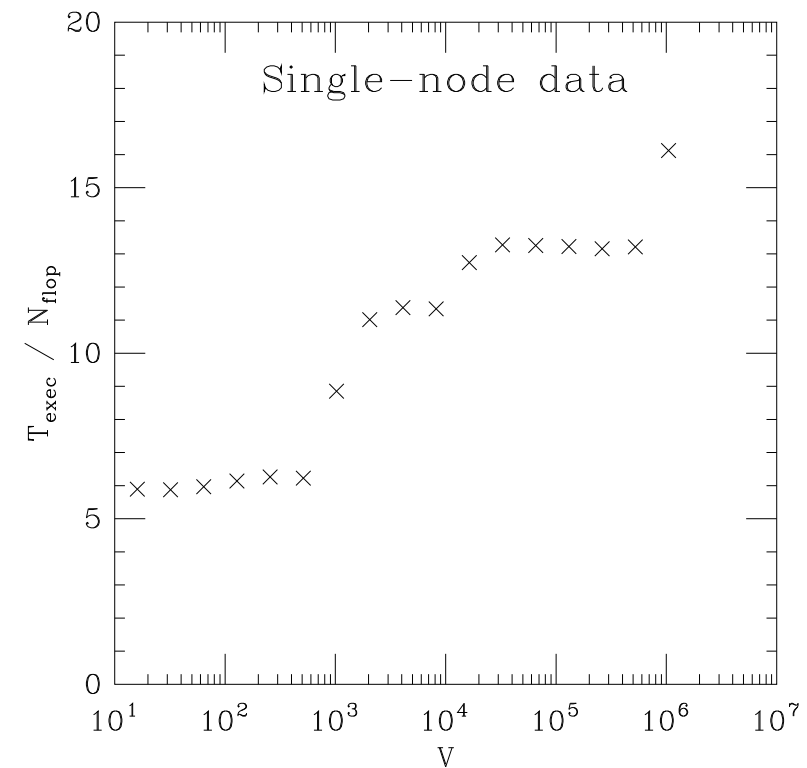
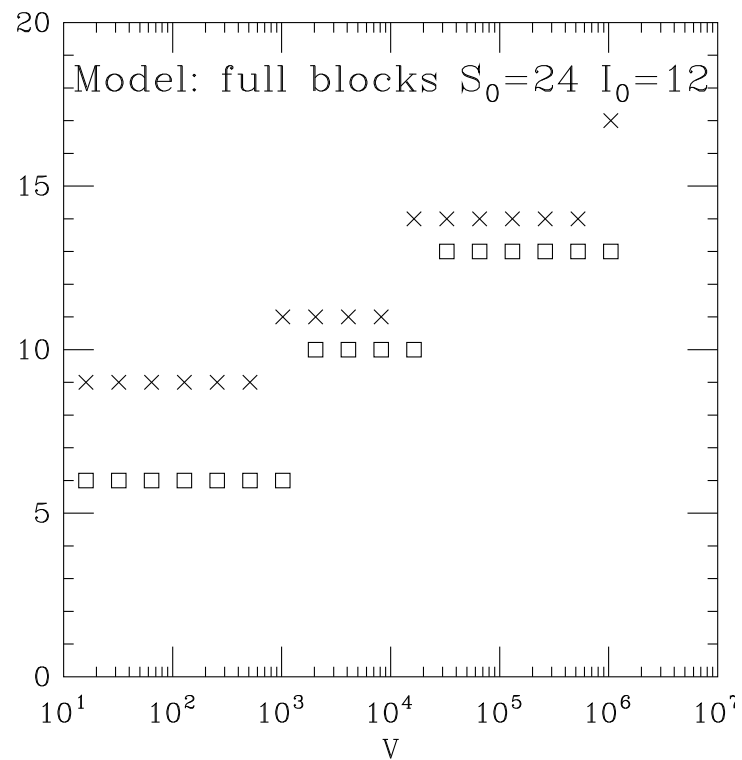


Application of Information Exchange

LQCD Dirac kernel (CHROMA code) on a single Opteron node

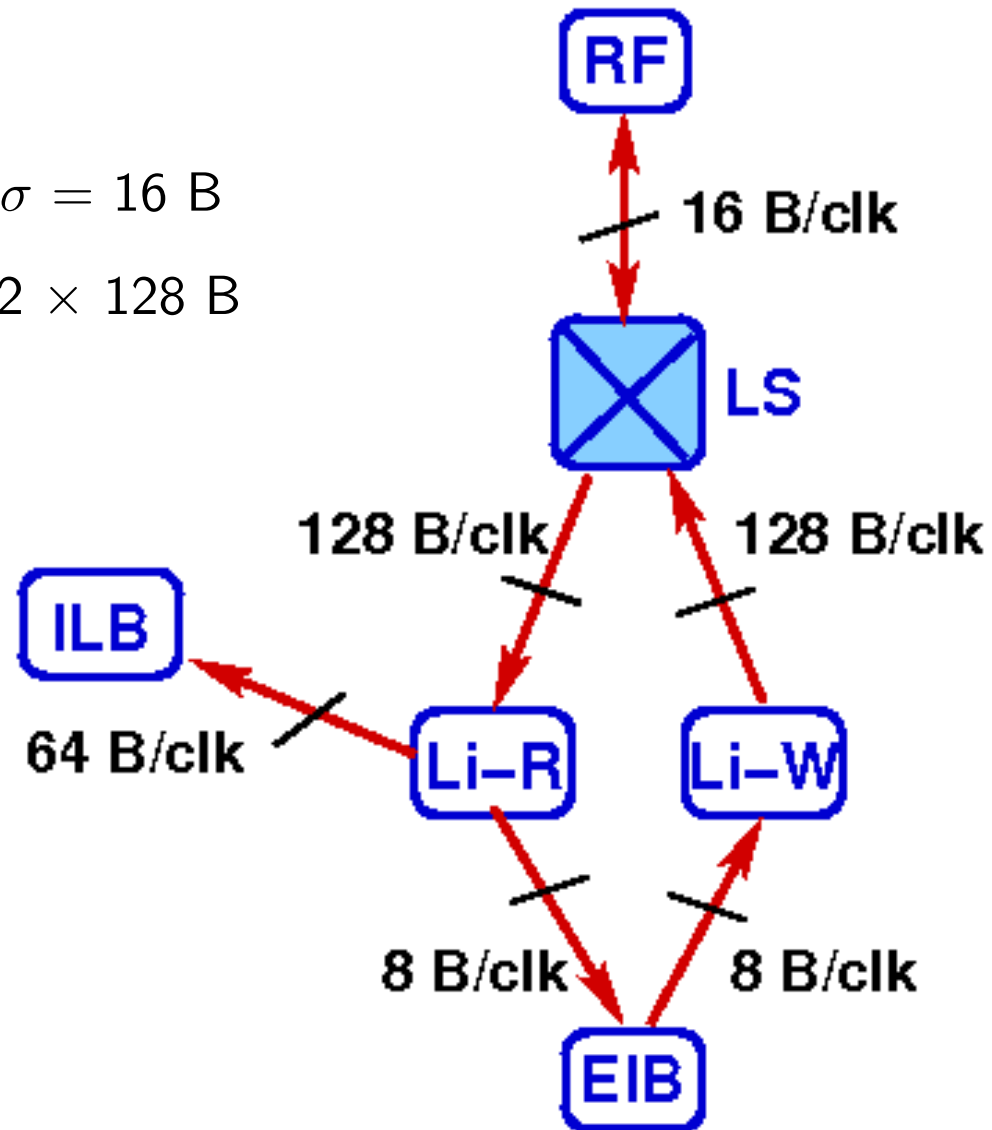
Model for information exchange
between cache and memory

Measured execution time
[J. Grieger]



Hardware Structure for LS Access on the Cell

- Quadword interface: $\sigma = 16 \text{ B}$
- Line interfaces: $\sigma = 2 \times 128 \text{ B}$



Theoretical Hardware Parameters of the Cell (from Data Sheets)

parameter	value	comment
β_{RR}^*	8 flop/clock	single precision throughput
β_{LS-R}^*	16 B/clock	quadword interface
β_{LS-ILB}^*	16 dinst/clock	double instructions (1 dinst = 2×4 B)
β_{LS-LI}^*	128 B/clock	line interface
β_{LI-EIB}^*	2×8 B/clock	data (concurrent R+W)
β_{1EIB}^*	8 B/clock	single transfer per ring
$\beta_{\Sigma EIB}^*$	64 B/clock	aggregate (limited by snooping)
β_{MM-EIB}^*	8 B/clock	XDR
σ_{LS}^*	256 KB	code size?
λ_{xy}^*	?	?

Test Hardware

- ❑ IBM blade QS20 (Jülich)
- ❑ Mercury Cell Accelerator Board (INFN Milano/Ferrara)
- ❑ Sony Playstation 3 (Uni Ferrara)

	QS20	CAB	PS3
f_{clk}	3.200 GHz	2.8 GHz	3.192 GHz
f_{tb}	14.3 MHz	14.3 MHz	79.8 MHz
# CBE	2	1	1
# SPE	2×8	8	6
XDR	2×512 MB	1 GB	256 MB
DDR2	—	4 GB	—
price [k]	$O(18)$	$O(7 + 3)$	$O(0.6)$
cent/Mflops	4.4	5.5	0.4

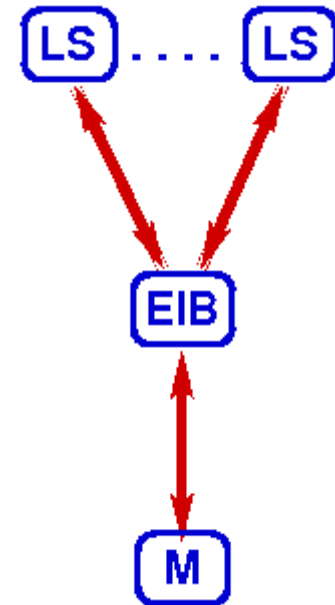
Real work: N. Eicker, N. Meyer, A. Nobile, D. Pleiter, F. Schifano, T. Wettig

Micro Benchmarks: LS – Memory

Benchmark: Single access to L bytes from N SPE:

```

for i=1, ... {
  get( $a_0, L$ )
  wait
}
    
```



Result:

$$T_{exe} = \lambda_0 + N \cdot \left\lceil \frac{L}{128} \right\rceil \cdot \frac{128}{\beta(N)}$$

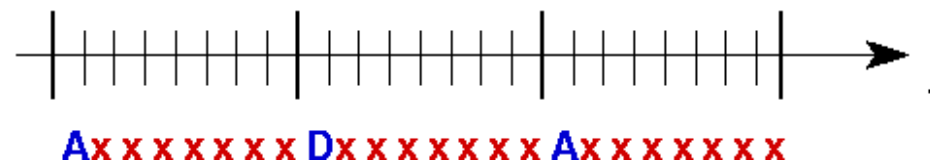
with

$$\lambda_0 \approx \begin{cases} O(10'000) \text{ cycles} & (\text{PTE miss}) \\ 950 \text{ cycles} & (\text{TLB miss}) \\ 400 \text{ cycles} & (\text{otherwise}) \end{cases}$$

and $\beta(N)$ in B/clock:

		$N = 1$	$N = 2$	$N = 8$
QS20	get	3.75 (47 %)	7.30 (91 %)	7.75 (97 %)
	put	3.91 (49 %)		
PS3	get	4.38 (55 %)	7.8 (97 %)	7.87 (98 %)
	put	6.88 (86 %)	7.8 (97 %)	7.87 (98 %)

Interpretation: each access occupies path LS-EIB for at least 2×16 cycles?



Micro Benchmarks: LS – Memory

Benchmark: Subsequent access from 1 SPE to 128 B at address distance Δa

```
get(a0,128)
get(a0+ $\Delta a$ ,128)
wait
```

Result on QS20 (T_{exe} in cycles):

$\Delta a/128$	SPU0	SPU1	...
0	1020	1022	
$k \cdot 16$	640	645	
$k \cdot 2$ ($k \neq 8$)	629	631	
$k \cdot 2 + 1$	622	624	

described by

$$T_{RR}(\Delta a) \approx \lambda_0 + \left[(\lambda_r \cdot \delta_{r_0, r_1} + \lambda_b) \cdot \delta_{b_0, b_1} + \lambda_x \right] \cdot \delta_{x_0, x_1} + \frac{128}{\beta_{XDR}^*}$$

with

$$\lambda_r \approx 400 \pm 1 \text{ cycles}$$

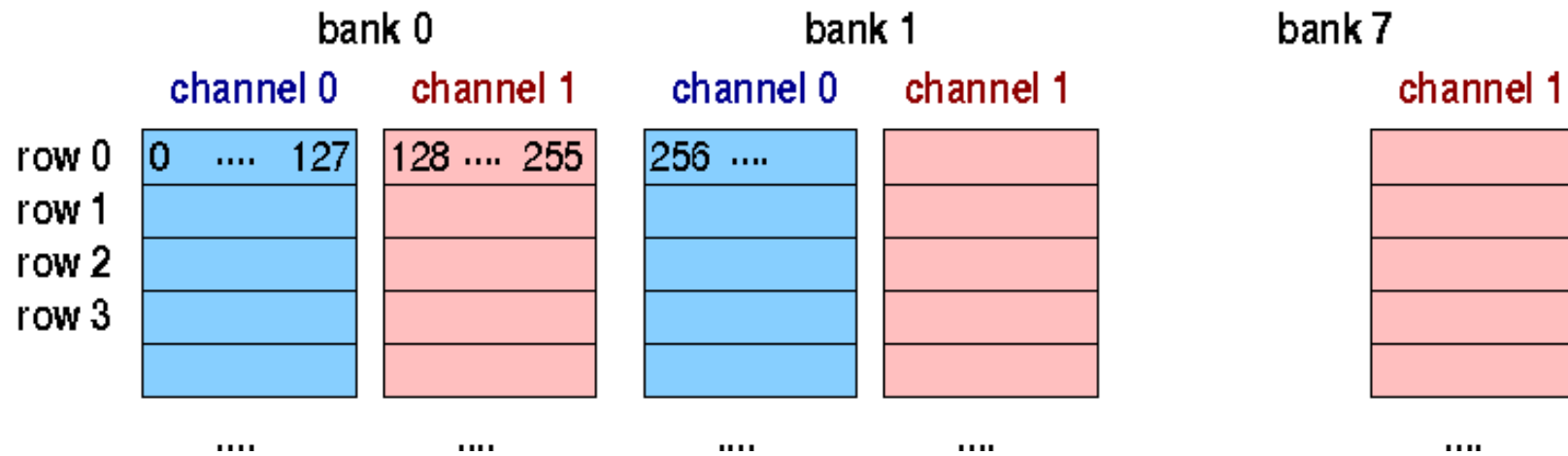
$$\lambda_b \approx 11 \dots 20 \text{ cycles}$$

$$\lambda_x \approx 1 \dots 7 \text{ cycles}$$

and r_i, b_i, x_i correspond to the row, bank, and channel index, respectively, of a_i

Micro Benchmarks: LS – Memory

Interpretation in terms of memory organisation:



Benchmark: Fragmented access from 1 SPE to $S = 128$ KB into **individual** DMA of L bytes

```

for i=1, ... S/L {
    get(a,L)
    a += L
}
wait
    
```

Result: aggregate bandwidth [B/cycle] reaches asymptotic value already at $L = 256$ B

L	get (QS20)	put (QS20)	get (PS3)	put (PS3)
128	3.2 (42 %)	3.5 (44 %)	4.0 (50 %)	6.0 (75 %)
16 K	3.7 (46 %)	3.9 (49 %)	4.8 (55 %)	7.0 (87 %)

but collapses for $L < 128$

Application Benchmark: Linear Algebra

Model: (single precision)

name	operation	N_{flop}	I_{RM} [float]	T_{peak} [cycles]	LS		MM	
					T_{exe} [cycles]	ϵ_{FP} [%]	T_{exe} [cycles]	ϵ_{FP} [%]
caxpy	$c \cdot \phi + \phi'$	12×8	24×3	12	18	66	288	4.1
dot	(ϕ', ϕ)	12×8	24×2	12	12	100	192	6.3
rdot	$Re(\phi', \phi)$	12×4	24×2	6	12	50	192	3.1
norm	$\ \phi\ ^2$	12×4	24×1	6	6	100	96	6.2

Benchmark: caxpy on 8 SPE

	T_{exe} [cycles]	ϵ_{FP} [%]
LS (C)	133	9
...
LS (asm)	26	46
MM/nop	370	3.2
MM	370	3.2

- LS: Performance limited by bandwidth of both SPU pipelines (critical scheduling)
- MM: Performance is heavily limited by memory bandwidth (no effect of dropping all FP operations)

Instruction Match for complex Arithmetics

Operands: $a \times \vec{b} + \vec{c}$

ar: rrrr (register)

ai: iiii (register)

b: riri (LS)

c: riri (LS)

Arithmetics:

$s = \text{shuf } b$

$t = \text{madd } ar \ b \ c$

$t' = \text{mul } ai \ s$

$r = \text{madd } sign \ t' \ t$

b_i	b_r
$a_r \cdot b_r + c_r$	$a_r \cdot b_i + c_i$
$a_i \cdot b_i$	$a_i \cdot b_r$

Cost:

caxpy

$$\beta_R = 1/3 \text{ saxpy}/\text{clk}$$

$$\beta_I = 1/4 \text{ saxpy}/\text{clk}$$

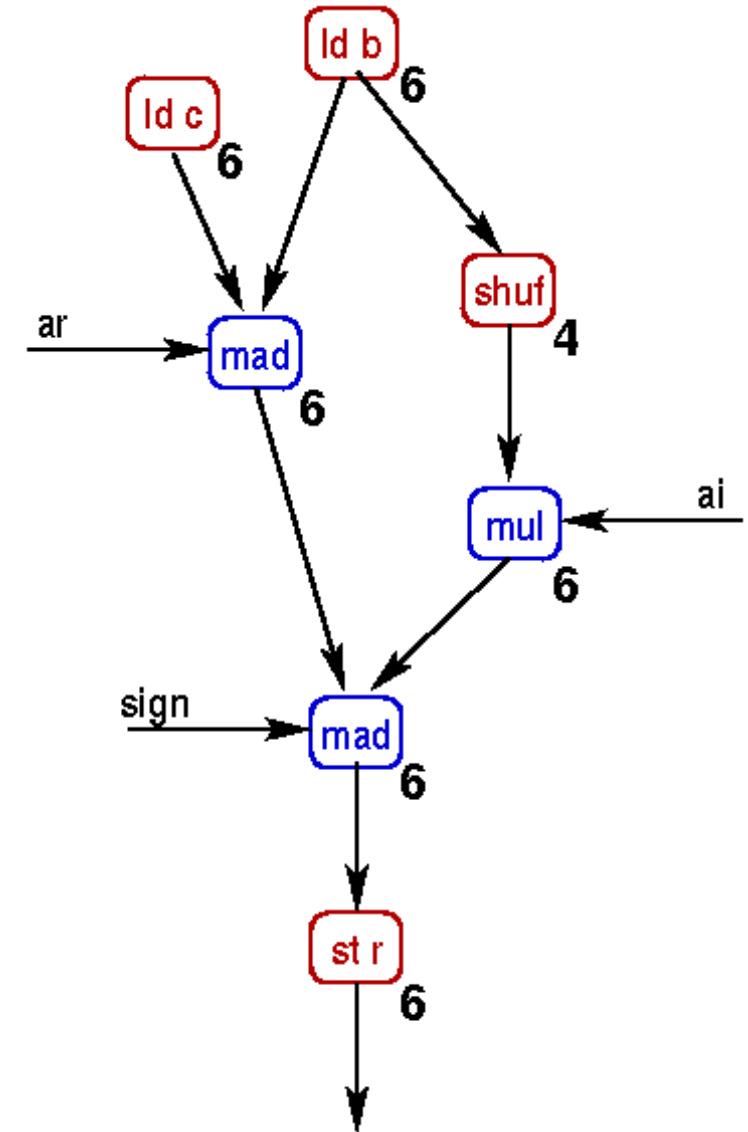
$$\lambda = 16 + \lambda_{I/O}$$

complex mad

$$\beta_R = 1/3 \text{ cmad}/\text{clk}$$

$$\beta_I = 1/7 \text{ cmad}/\text{clk}$$

$$\lambda = 16 + 2 + \lambda_{I/O}$$



N.B.: Arithmetics on SPU is only round to zero (truncation)

Lattice QCD Simulations

Data set: Fields on 4-d space-time lattice

- Quark field $\phi(x)$: 12 complex/site
- Gluon field $U(x, \mu)$: 9 complex/link

→ lattice sizes up to $50^3 \times 100$

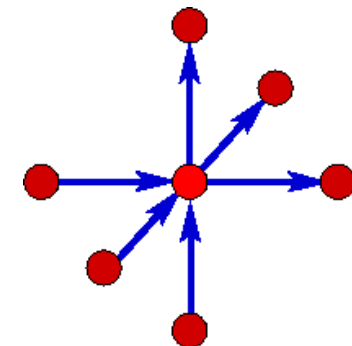
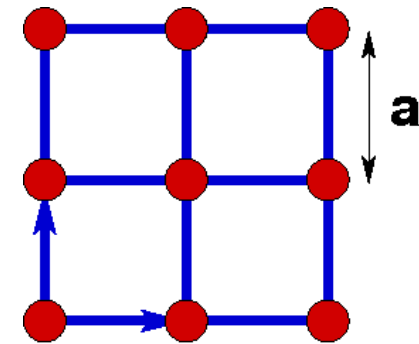
Main computational tasks:

- linear algebra on ϕ vectors
- matrix multiplication (Wilson-Dirac operator)

Hopping term of Wilson-Dirac operator:

$$[D\phi]_x \equiv \sum_{\mu=1}^4 \{U(x, \mu)(1 - \gamma_\mu)\phi(x + \hat{\mu}) + \dots\}$$

→ 1320 FP operations/site



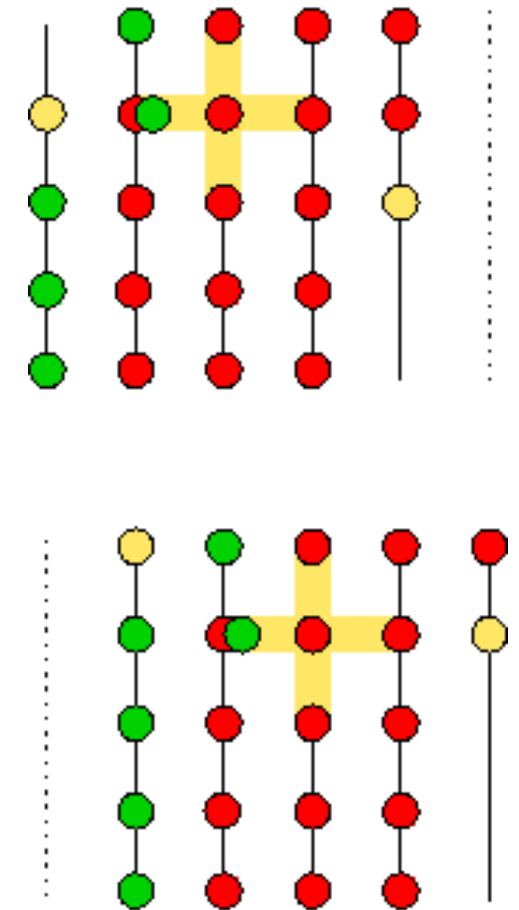
Application Benchmark: Dirac Operator

Implementation:

- Local $4^3 \times L_0$ lattice on each SPE ($L_0 = 16, 32, 64$)
- Computation of $D\phi$ on 3-dim slice involves 3 slices
- Keep 3+2 slices in LS moving along L_0
- All SPEs synchronised by signal after each slice
- Storage of fields in memory:

```
phi [8] [L0] [64]
u0 [...]          (with padding)
u1 [...]
...
```

	site	slice 4^3	SPE $4^3 \times 32$	Cell $8^3 \times 32$
ϕ	96 B	6144 B	192 KB	1.50 MB
U_0 (padded)	80 B	5120 B	160 KB	1.25 MB
...				
$I_{R(LS \leftarrow MM)}$	416 B	26 KB	832 KB	6.7 MB
$I_{RW(LS \leftrightarrow MM)}$	512 B	32 KB	1 MB	8.0 MB
$I_{R(LS \leftarrow LS)}$	144 B/dir	2.25 KB	36 KB	288 KB



Performance of Dirac Operator

Model:

	cycles
T_{FP}	$10.6 \cdot 10^3 \cdot L_0$
T_{LS-LS}	$2.3 \cdot 10^3 \cdot L_0$
T_{LS-MM}	$32.8 \cdot 10^3 \cdot L_0$

Benchmark: LS-MM access only

L_0	QS20			CAB		
	10^3 clk	ε_{MM}	ε_{FP}	10^3 clk	ε_{MM}	ε_{FP}
16	41.6	79 %	25 %	41.7	79 %	25 %
32	39.7	82 %	27 %	40.0	82 %	26 %
64	39.1	84 %	27 %	43.7	75 %	24 %

For $L_0 = 32$ and 4-KB pages, each SPE needs at least 208 TLB entries

A Cell-based QCD Machine

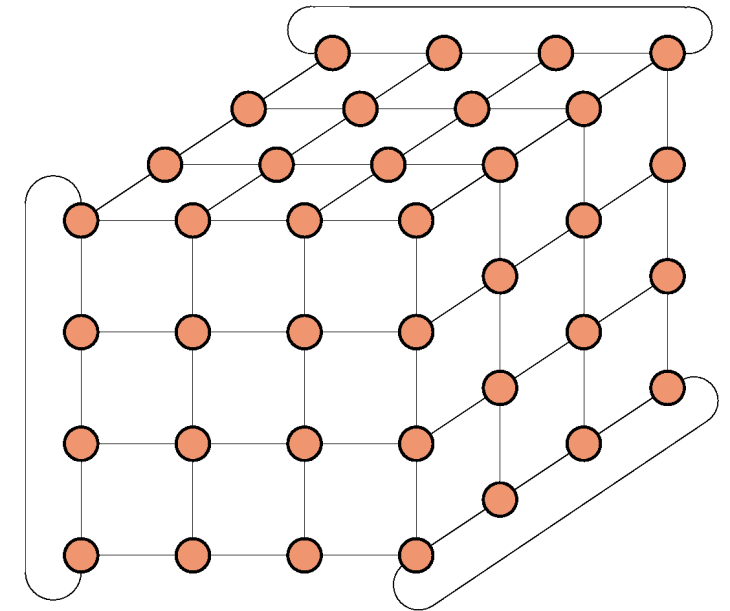
Goal: scalable Cell-based architecture

Parallelisation: spatial domain decomposition

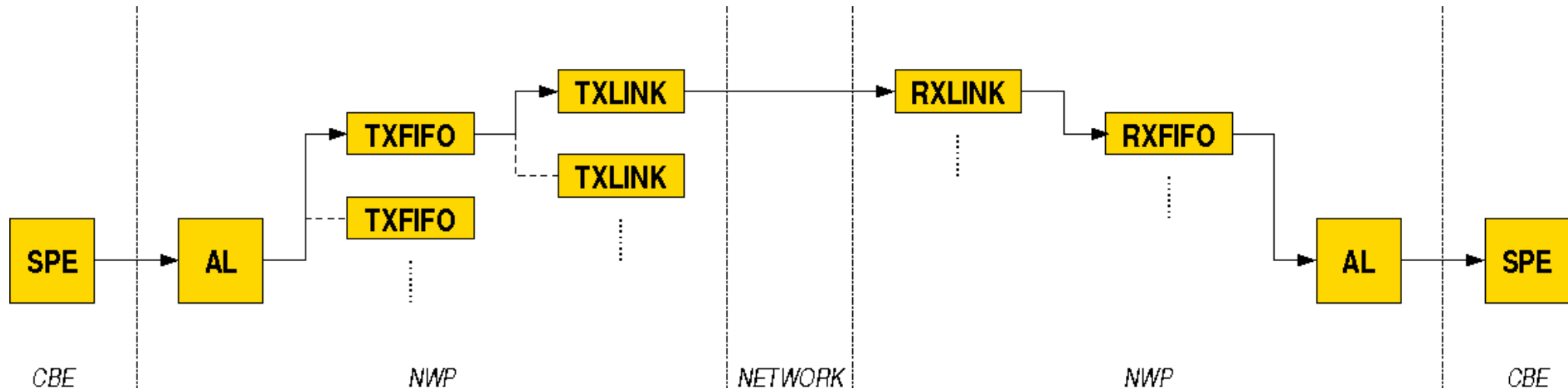
- nearest neighbour communication
- homogeneous communication patterns

Requirements:

- 3-dimensional torus topology
- high bandwidth: $\beta_{link} \geq 1$ GByte/link/direction
i.e. 6 GByte/s bidirectional between Cell and network
- low latency: $\lambda_{net} < 1$ μ sec from LS to LS over network



Application-optimised Network



Operations needed for communication:

- **Source: SPE performs DMA put to I/O device (=network processor)**
- Data is moved via send FIFO to sender
- Data is received and moved to receiver FIFO
- **Destination: DMA from I/O device to LS of SPE**

Summary and Conclusions

- ❑ Theoretical Performance Analysis based on
 - simplified machine model
 - information exchange functionallows to investigate algorithm-implementation space

- ❑ Micro-Benchmarks
 - determine effective machine parameters
 - improve understanding of hardware behaviour

- ❑ Preliminary application benchmarks confirm potential of a Cell-based QCD machine