The apeNEXT Project

INFN Ferrara, Rome

DESY Zeuthen

Université de Paris-Sud, Orsay

Outline:

❑ Introduction
❑ Architecture
❑ Hardware
❑ Software
❑ Status

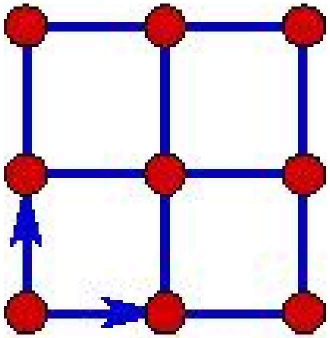
H. Simma, ACAT2005

Lattice QCD

Feynman path integral

$$\langle O \rangle \sim \int D[U]\, D[\psi]\, O(U,\psi) \cdot exp\{-S_g(U) - S_q(U,\psi)\}$$

Discretisation on a finite space-time lattice



$U(x,\mu)$: 9 complex/link

$\psi(x)$: 12 complex/site

e.g. lattice size $\mathbf{L^3 \times T = 32^3 \times 64}$ → $\mathbf{2 \cdot 10^6}$ sites

Monte-Carlo method

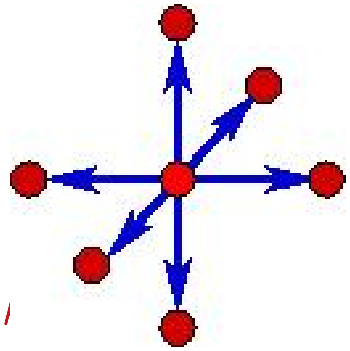
$$\langle O \rangle \;\rightarrow\; \frac{1}{\#U} \sum_{\{U\}} O(U)$$

with gauge configurations $\{U\}$ generated according to distribution

$$P(U) \;\sim\; e^{-S_g(U)} \cdot \int D[\psi]\, e^{-\bar{\psi} M(U) \psi}$$

Wilson-Dirac operator

$$\begin{aligned} [\mathbf{M}\psi]_x \;&=\; (D_\mu \gamma_\mu + m + a \cdots)\psi \\ &\sim\; \psi_x - \kappa \sum_{\mu=\pm 1}^{\pm 4} \mathbf{U}_{\mu,\mathbf{x}} \cdot (1-\gamma_\mu)\psi_{x+} \end{aligned}$$



→ 1320 floating-point operations per lattice site

apeNEXT Project

2000: ECFA ➠ O(10) Tflops in 2004

2001: MoU development of apeNEXT:

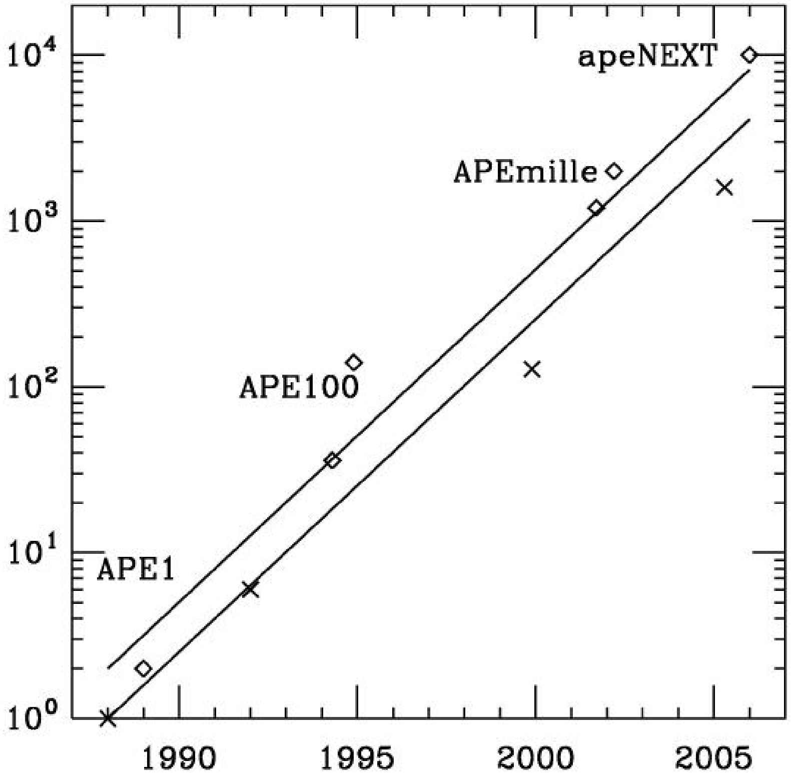
INFN	DESY	U. Paris-Sud
Pisa/Ferrara	Zeuthen	Orsay
Roma	Bielefeld	Rennes

$\approx$ 2 M€ for NRE + prototype HW

2003: chip sign-off

2004: 2×64 node prototype systems

2005: large installations



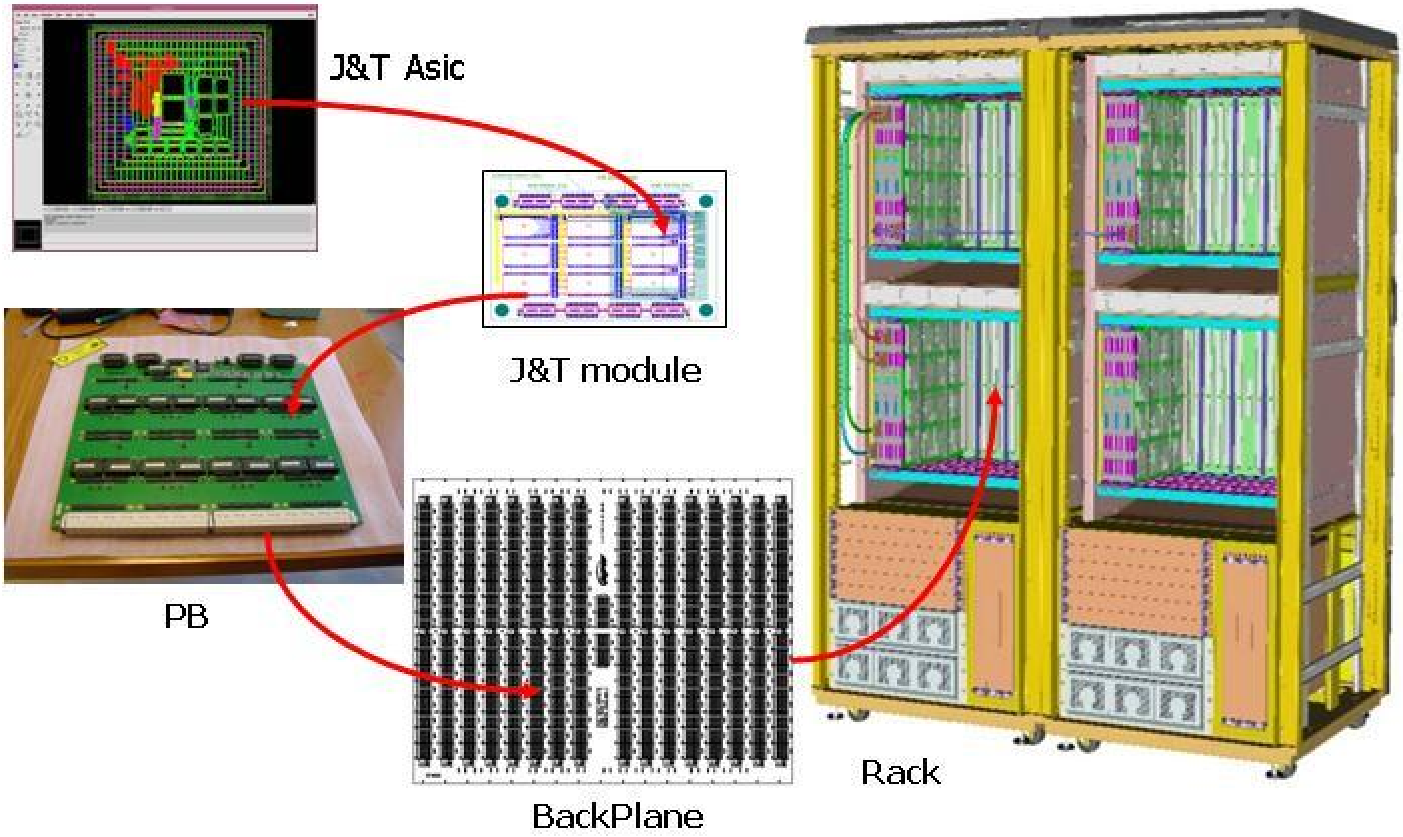
Other machine projects:

- ❑ QCDOC (Columbia + IBM)
- ❑ BlueGene/L (IBM)
- ❑ PC Clusters (PMS, Wuppertal, APEnet, etc.)

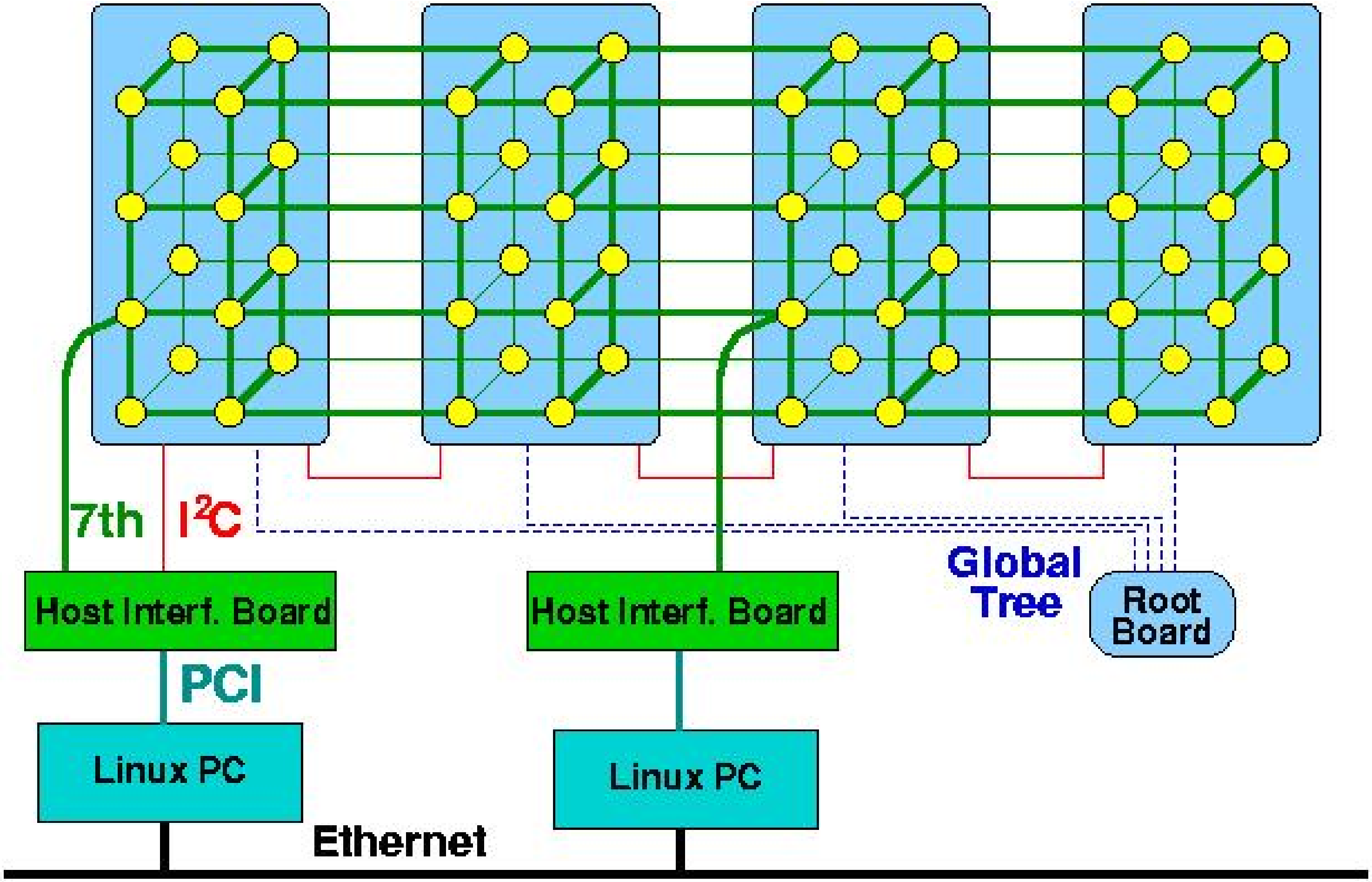
apeNEXT Design Aims:

- ☞ Scalable up to tens of Tflops
- ☞ Architecture mainly optimized for LQCD (i.e. $\approx 50\%$ sustained)
- ☞ All processor and network functionality on a single chip
- ☞ Support for C programming language

Hardware Overview



Global apeNEXT Architecture



Massively-Parallel System Architecture

APE100, APEmille, apeNEXT, . . . , QCDOC, BlueGene/L

☞ N-dim torus communication network

☛ Autonomous nodes with local on- and off-chip memory

☞ Integrated memory interface (with ECC)

☛ Integrated communication links (with sync and re-send)

☞ IO via separate nodes or hosts

☞ Single user process (minimal OS, no virtual addresses)

☛ Serial control network

☞ Global interrupt tree

☞ Global clock tree

☞ Low power consumption and high packing density

Optimized Node Architecture

❑ Processing Unit

- arithmetic operations and control instructions
- throughput ↔ N_{flop}
- data formats

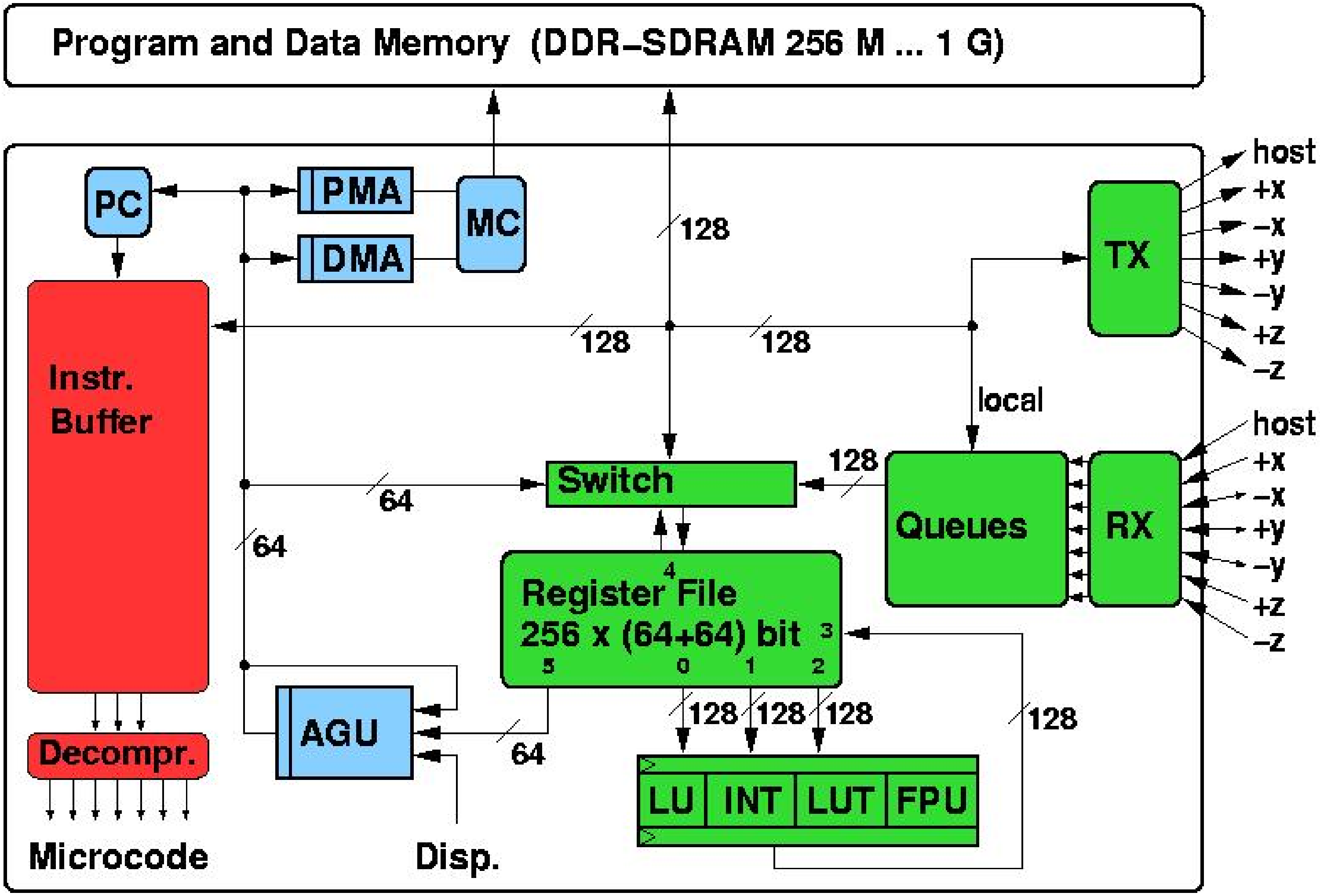
❑ Memory System

- size
- bandwidth ↔ N_{word}/N_{flop}
- latency
- access model and hierarchy

❑ Communication Network

- bandwidth ↔ $N_{com}(V_{loc})/N_{word}$
- latency
- connectivity

Processor Overview



Arithmetic Units

❑ floating point unit (FPU) performs one operation $a \times b + c$ per clock cycle, where a, b, c are complex numbers or pairs of float

➔ 8 Flops / cycle = 1.6 GFlops/sec

❑ 64-bit IEEE floating point format

❑ arithmetic unit provides also integer, logical and LUT operations on pairs of 64-bit operands

❑ address generation unit also usable for 64-bit integer operations

Memory Hierarchy

register file

- ❑ 512 64-bit registers

➔ 256 complex operands

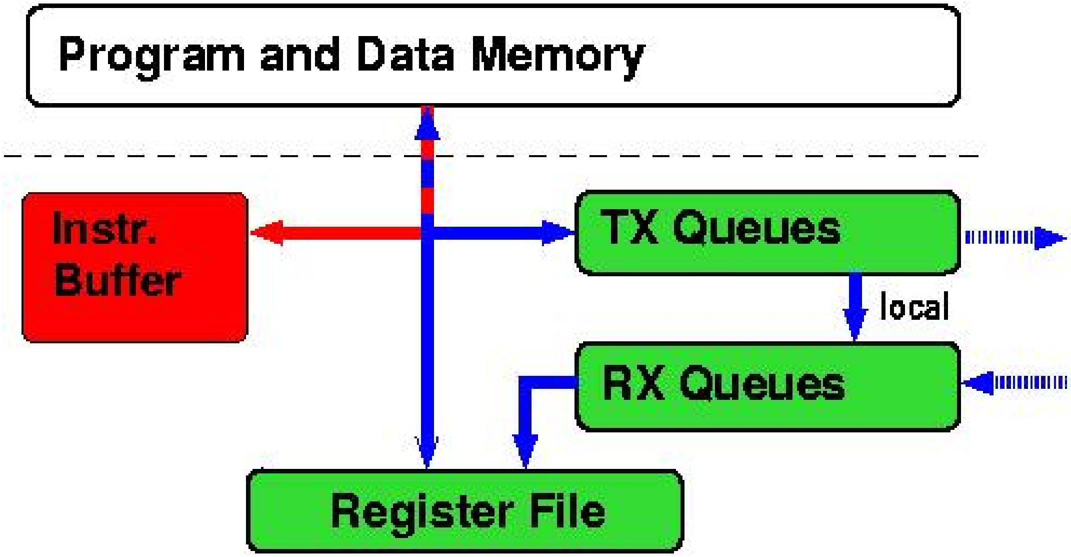
memory controller

- ❑ supports 256 MBytes upto 1 GBytes DDR-SDRAM (with ECC)
- ❑ maximum bandwidth: 1 complex word per clock cycle

➔ 2×64 bit/cycle $= 3.2$ GBytes/sec

- ❑ minimal latency: 16 cycles
- ❑ controls loading of data and program instructions

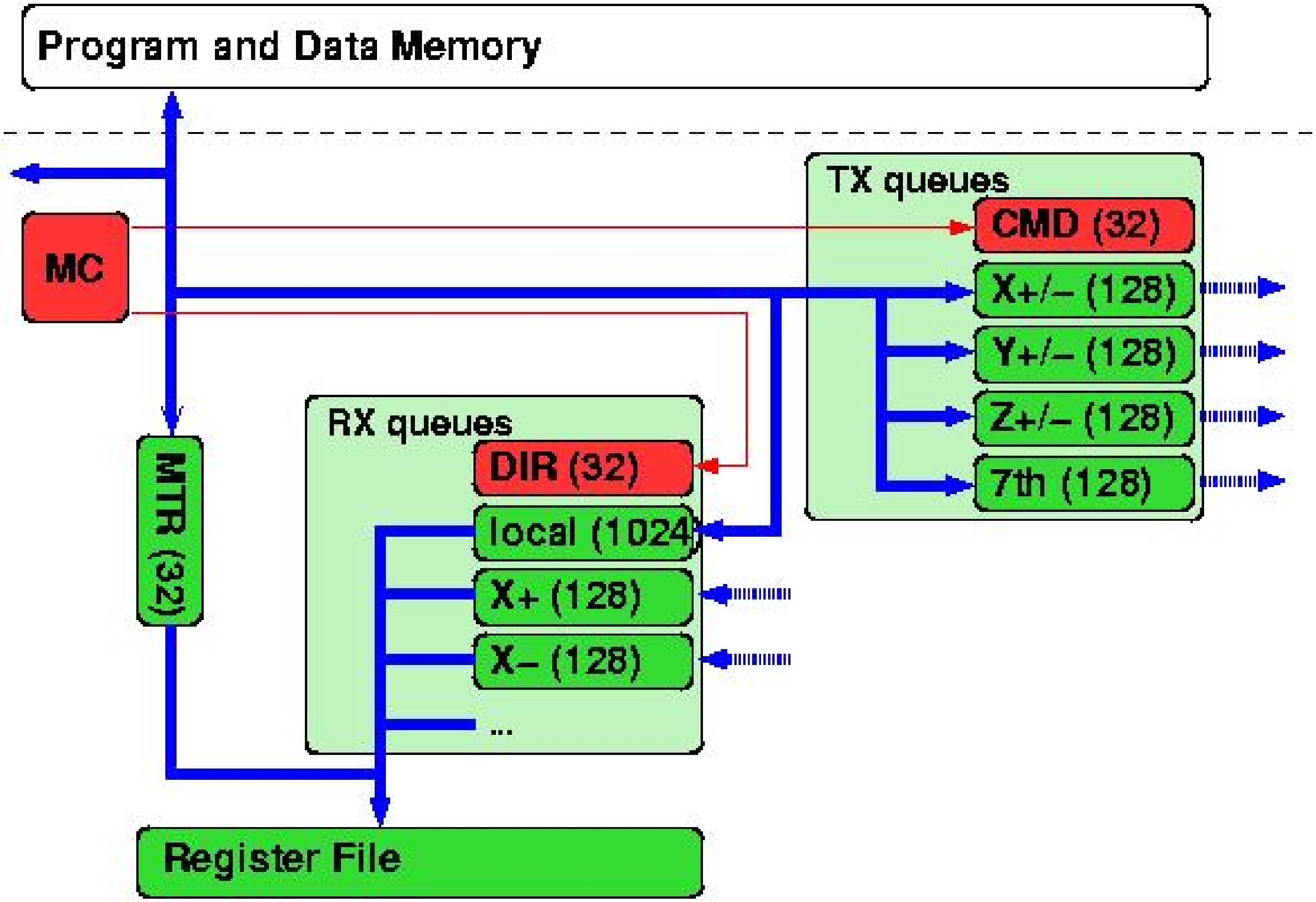
Memory Hierarchy (cont.)



instruction buffer

- ❑ allows storing 4k compressed, very long instructions words (VLIW)
- ❑ can be used as FIFO or dynamic/static cache

Prefetch Queues and Network Interface



Network

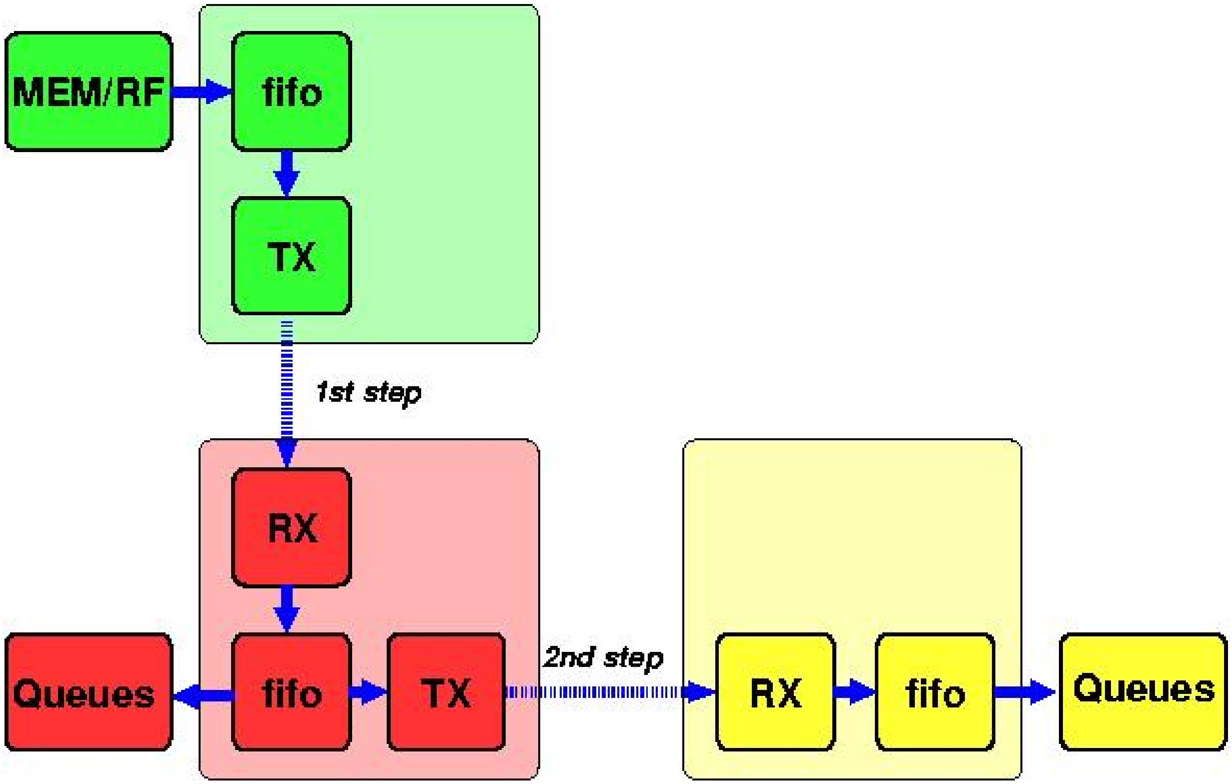
- ❑ 7 bi-directional LVDS links: $\pm x$, $\pm y$, $\pm z$, 7th
- ❑ gross bandwidth per link is one byte per clock cycle

➔ 8 bit/cycle = 200 MBytes/sec

- ❑ transmission by frames of 128 bit data + 16 bit CRC
  ➔ effective bandwidth $\leq$ 180 MBytes/sec
- ❑ very low startup latency: $\approx$ 25 cycles (125 ns)
- ❑ concurrent send and receive operations
- ❑ concurrent transfer along orthogonal directions
- ❑ support for non-homogeneous communications
- ❑ configurable direction mapping

Network (cont.)

HW supports synchronising 1-, 2-, and 3-step communications



→ direct access to all nodes on a $3 \times 3 \times 3$ cube

Processor Design

ASIC chip

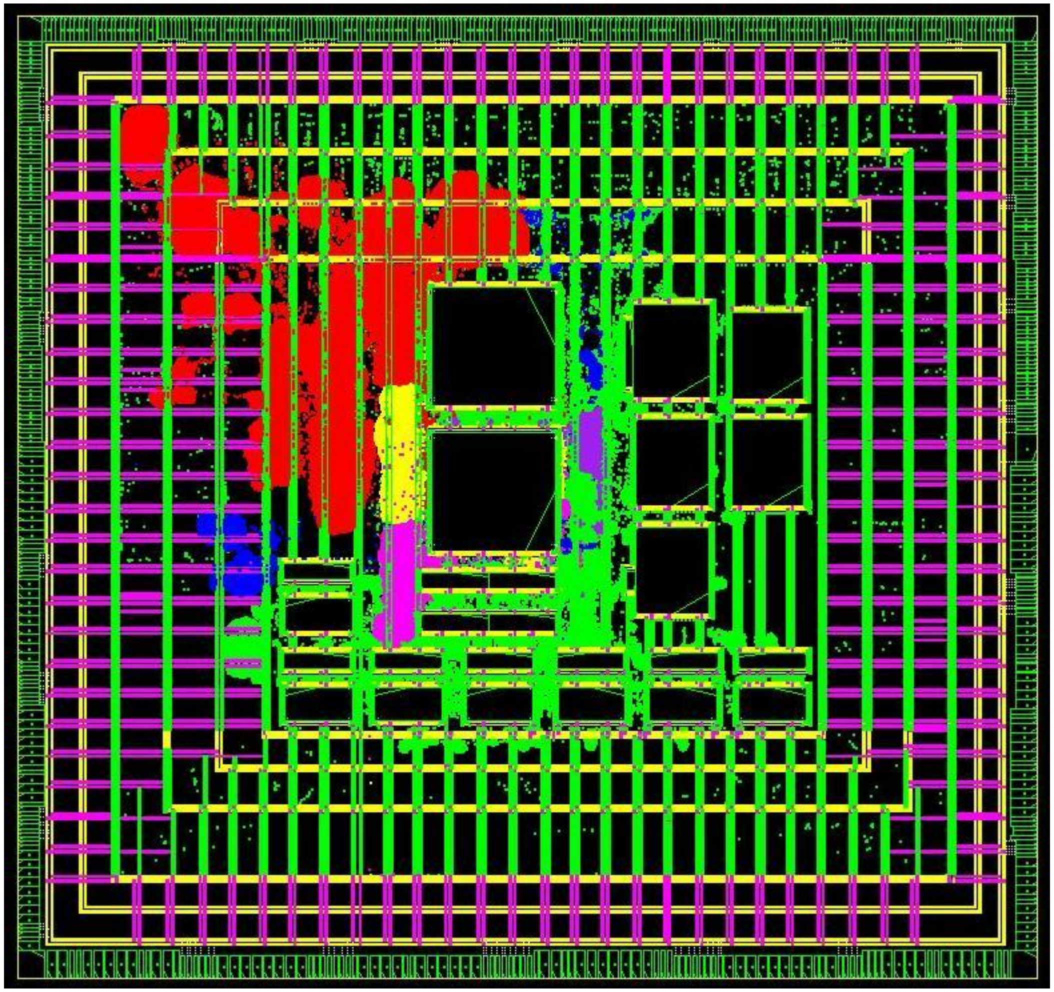
❑ 0.18μ CMOS

❑ $16 \times 16\ mm^2$

❑ 520 K gates

❑ 600 pins

❑ 200 MHz **??**



Hardware Details

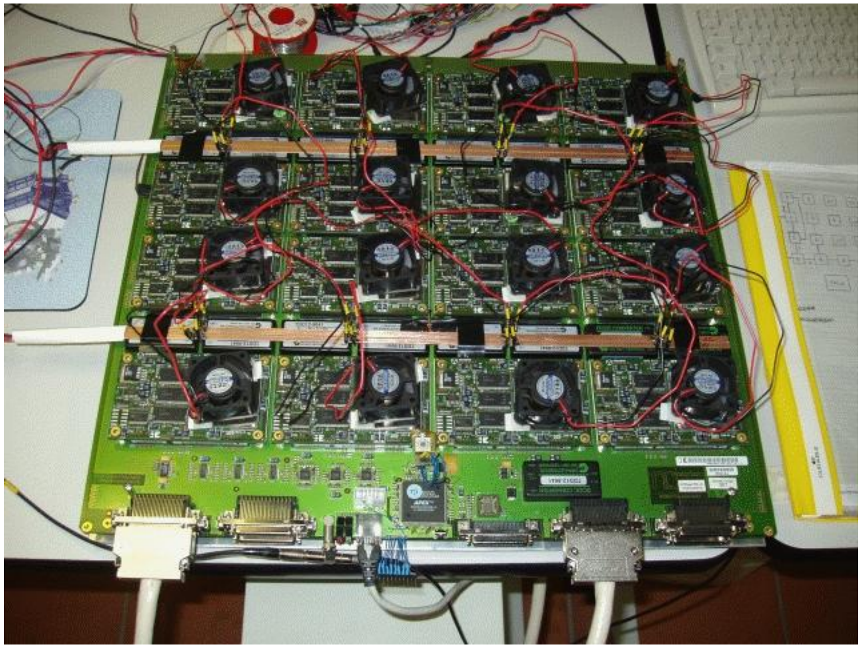
J&T Module (daughter board)

- ❑ processor chip
- ❑ 9 memory chips
- ❑ clock circuits
- ❑ power converters

Hardware Details (cont.)

Processing Board

- ❑ 16 daugther boards
- ❑ FPGA (global signals and I^2C)
- ❑ DC-DC converters (48 → 2.5 V)
- ❑ 1728 differential LVDS signals
- ❑ robust mechanical design
  (insertion force: 80-150 kg)



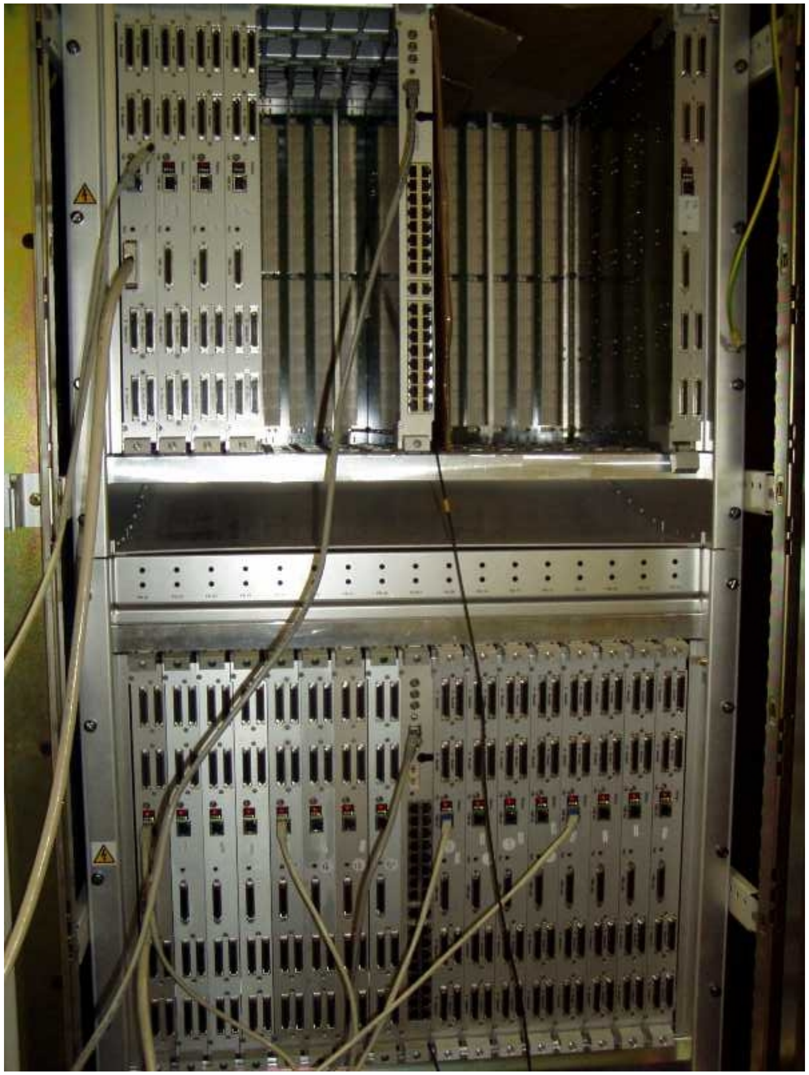
Hardware Details (cont.)

Backplane

- ❑ slots for 16 processing boards
- ❑ 4600 differential LVDS signals
- ❑ 16 PCB layers

Root Board

- ❑ global interrupt signals
  (500 ns round-trip for synch. barrier)
- ❑ clock distribution
- ❑ slow control

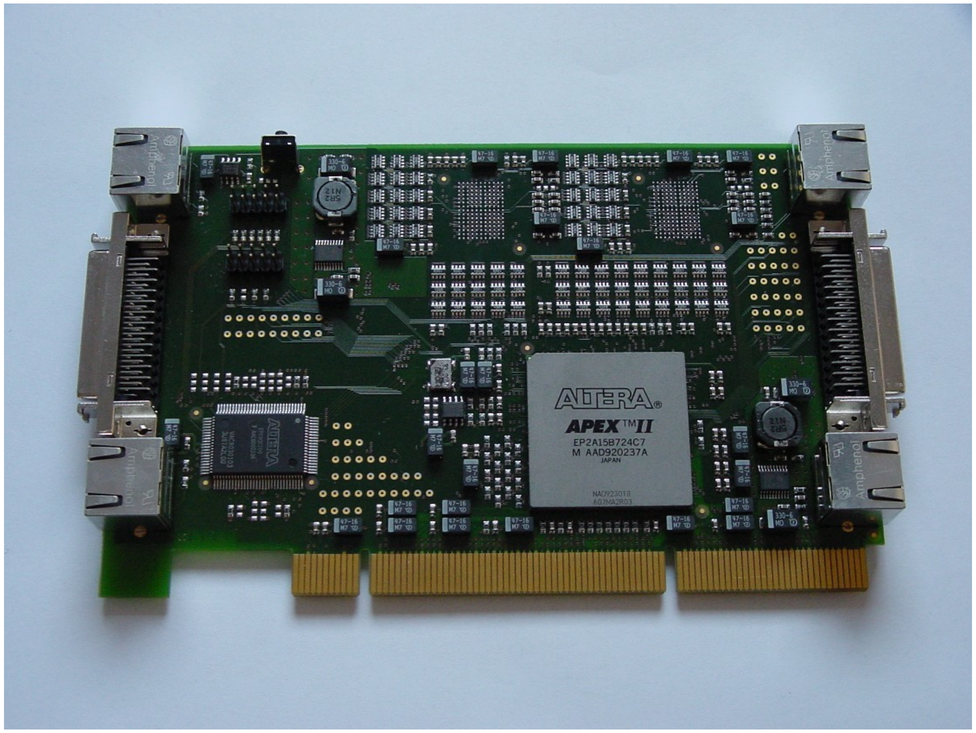


Hardware Details (cont.)

Rack

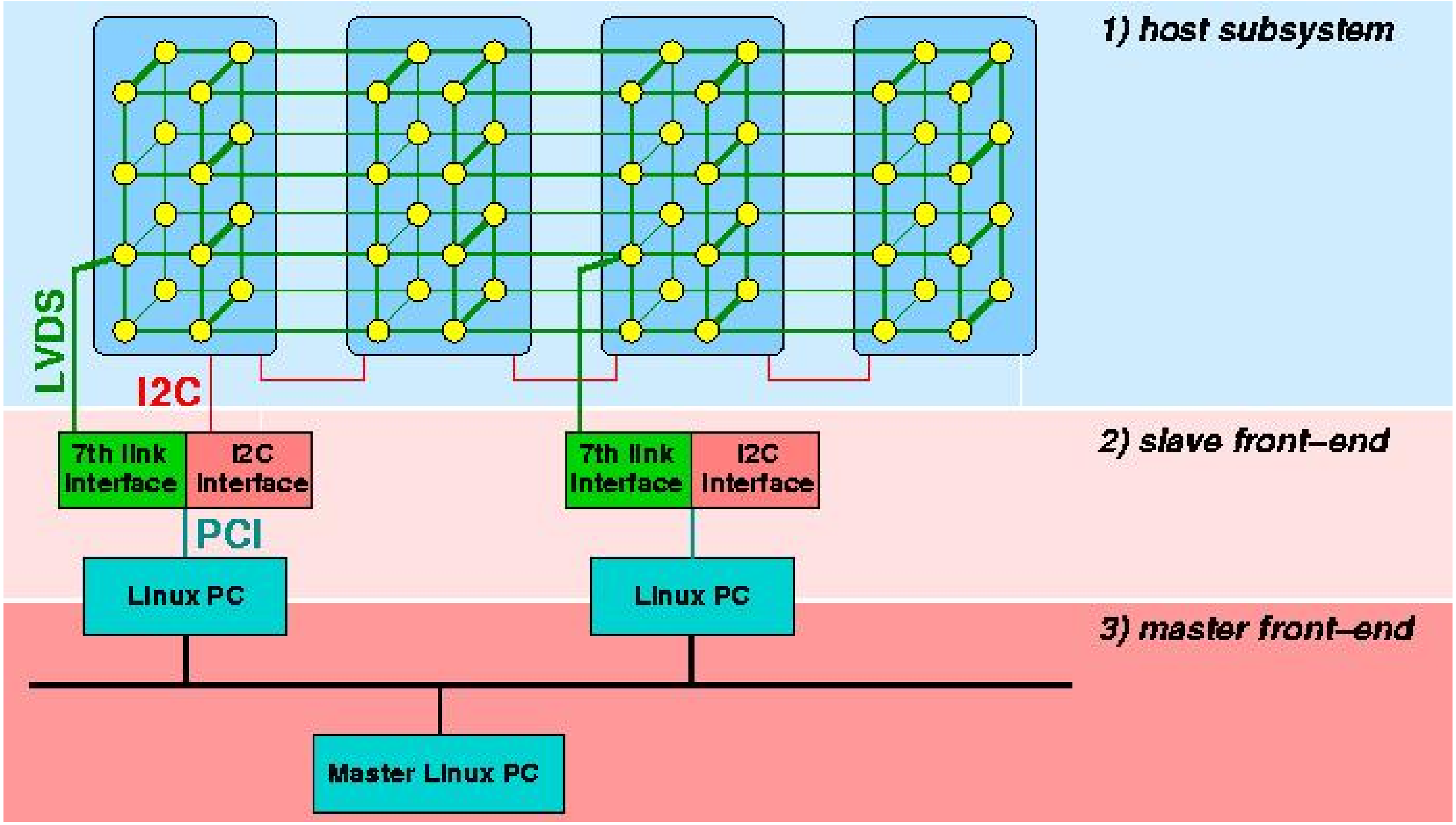
- ❑ slots for 2 backplanes
- ❑ footprint O(1 m^2)
- ❑ power consumption $\leq$ 8 kW
- ❑ air cooled
- ❑ hot-swap power supply

Host Interface Board



- Interface for 7th LVDS link (200 MByte/s)
- 4 interfaces for I^{2}C links
- PCI 64-bit / 66 MHz

Operating System



Operating System (cont.)

❑ Bootstrap, exception handling and debugging via I^2C

❑ Fast program loading and data IO via 7th link (BW ∼ #host PCs)

❑ Network toplologies (periodic communications):

- z $=$ 1, 2, 8
- y $=$ 1, 2, 8
- x $=$ 1, 2, 4, $4N_{crate}$

❑ Machine partitions (independent global control):

- node $1 \times 1 \times 1$
- cube $2 \times 2 \times 2$
- board $4 \times 2 \times 2$
- unit $4 \times 2 \times 8$
- crate $4 \times 8 \times 8$
- rack $8 \times 8 \times 8$
- . . .

Programming Languages

TAO

- ❑ FORTRAN-like programming language
- ❑ Dynamical grammar allows OO-style programming
- ❑ Needed for smooth transition from APEmille to apeNEXT

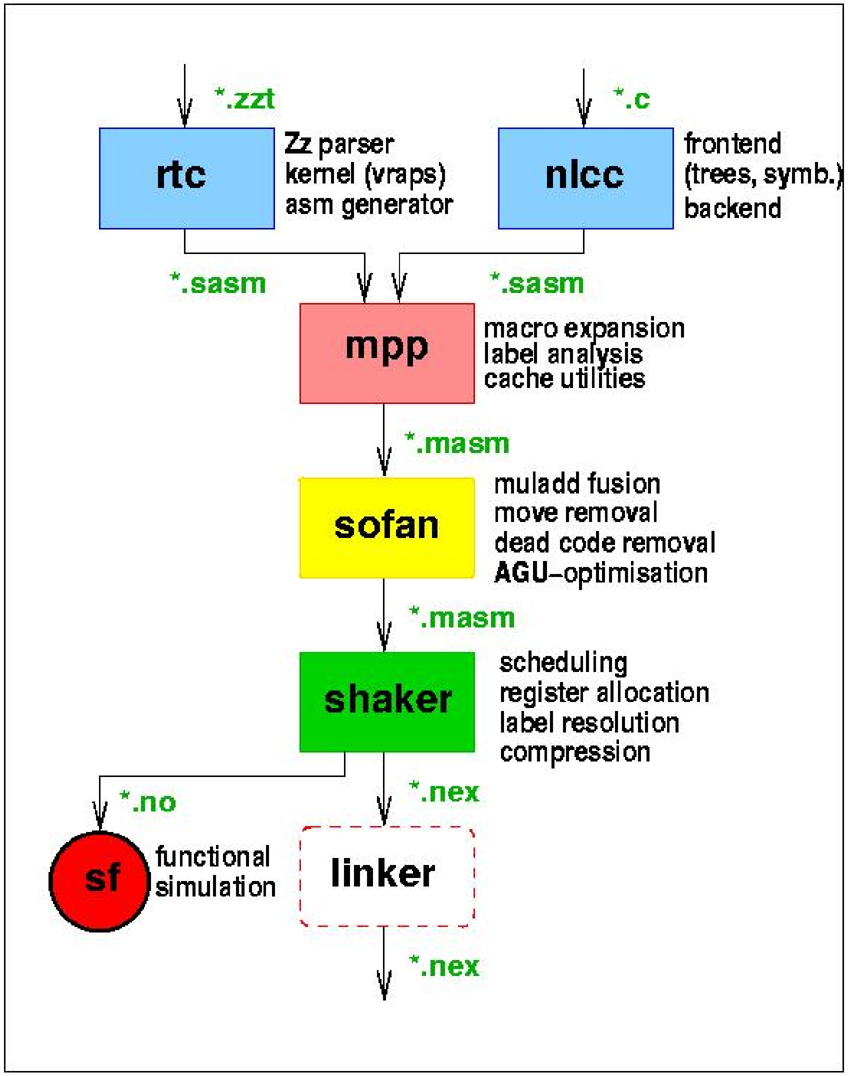
C

- ❑ Based on freely available lcc + custom implementation of libc
- ❑ Most of ISO C99 standard supported
- ❑ Few APE-specific language extensions

SASM

- ❑ High level assembly (e.g. for OS routines and C libraries)
- ❑ Aim: assembler programming by user not required

Compiler Overview



C-Compiler: Syntax Extensions

- ❑ New data types: `complex`, `vector`
- ❑ New operators: `~` (complex conjugation)
- ❑ New condition types: `where()`, `any()`, `all()`, `none()`
- ❑ `register struct` ➔ burst memory access
- ❑ `#pragma cache` ➔ enforce use of instruction buffer
- ❑ Inline functions and inline assembly

C-Compiler: Syntax Extensions (cont.)

- Magic offsets for remote communication:

```
complex  a[1],  b;

b = a[0+X_PLUS];        // read data from node in X+ direction
```

- Macros for data prefetching:

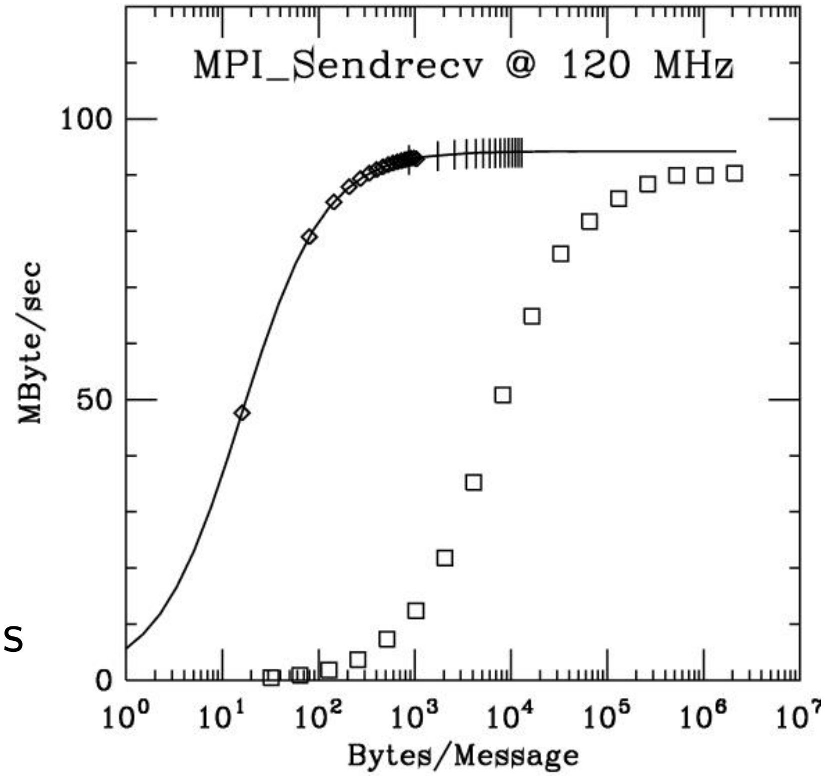
```
complex           a;
register complex  ra;

prefetch(a);            // memory → queue
fetch(ra);              // queue → register file
```

MPI on apeNEXT?

Restrictions:

- ❑ Only `MPI_COMM_WORLD`
- ❑ Only standard (buffered) mode
- ❑ Send always non-blocking
- ❑ Receive always blocking
- ❑ No request handles
- ❑ Only homogeneous communications beyond nearest neighbors

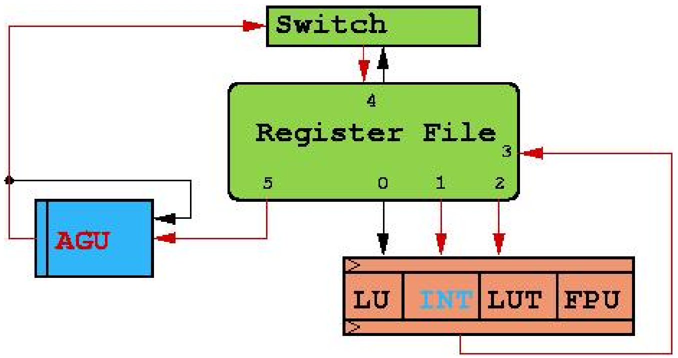


Extensions:

- ❑ `MPI_APE_Send`, `MPI_APE_Recv`

Assembly Optimizer: sofan

- Optimization operating on low-level assembly
- Based on optimization toolkit SALTO (IRISA, Rennes)
- Optimization steps:
  - merging APE-normal operations
  - removing dead code
  - eliminating register moves
  - optimizing address generation: →
  - code selection
  - instruction pre-scheduling
  - ...



Benchmarks: Linear Algebra

operation	N_{flop}	N_{word}	sustained performance				
			“max”	asm	C	C+Sofan	TAO+Sofan
`zdotc`	8	2	50%	41%	28%	40%	37%
`vnorm`	4	1	50%	37%	31%	34%	26%
`zaxpy`	8	3	33%	29%	27%	28%	28%

“max” sustained performance ⟵ ignoring latency of floating point pipeline and loop overhead

Optimization “tricks”:

➜ loop unrolling

➜ burst memory access

➜ instructions kept in cache

➜ Assembly not required

Performance limitations:

➜ start-up latency

➜ loop overhead

Benchmarks: Wilson-Dirac Operator

$$\Psi_x = D_{xy}[U]\,\Phi_y$$

Consider worst case: local lattice size 16×2^3

Measured sustained performance: 55%
Communication-wait cycles: 4% ➜ scalability

Optimization "tricks":

- ➜ keep gluon fields local
- ➜ pre-fetching 2 sites ahead
- ➜ orthogonal communication directions
- ➜ some unrolling

Status

- HW components:

processor	prototype tested
PB, backplane, rack	prototypes tested and frozen
host interface board	prototype tested and frozen

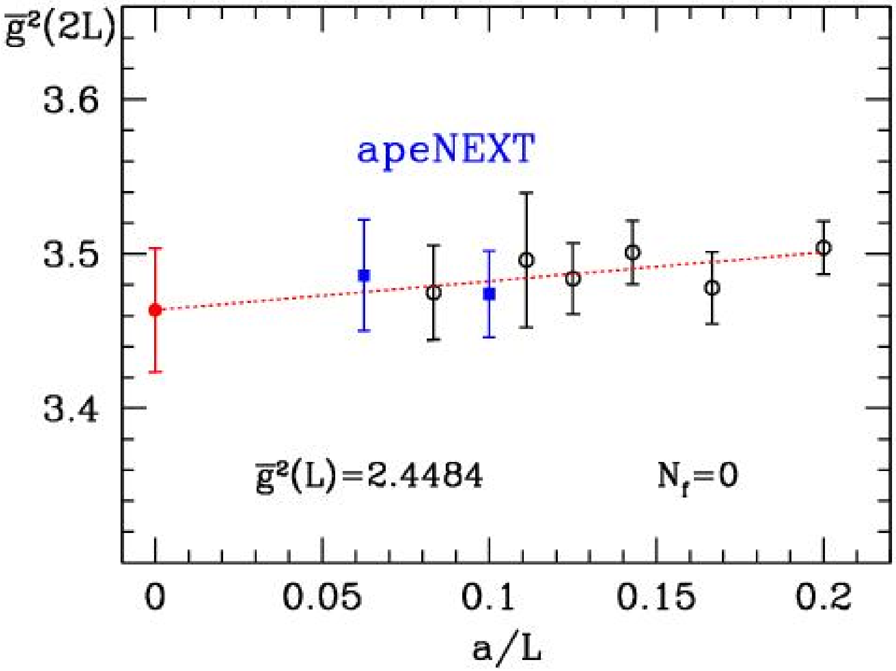
- SW elements:

TAO compiler	stable prototype
C compiler	prototype
assembly optimizer	developing
microcode generator	stable
linker	planned
operating system	developing

- 1st prototype rack with 512 nodes being tested
- 2nd prototype rack being assembled
- Successful test runs with physics codes

Physics Tests

ALPHA Collaboration



Continuum extrapolation of the step scaling function for the running coupling constant α_S with the Schrödinger functional for SU(3) pure gauge theory

Outlook

❑ On-going work:

- HW tuning to push speed and stability
- SW development to increase efficiency and usability
- Commissioning of “huge prototype” (1.6 Tflops)
- Qualification of revised chip production expected in July

❑ Large installations in 2005/2006:

- 5+5 Tflops @ INFN
- 3 Tflops @ DESY
- 5 Tflops @ Bielefeld
- Orsay?

❑ Exploit full potential of apeNEXT architecture

Architecture Comparison

	apeNEXT	QCDOC	BlueGene/L
Nodes	16 . . . ≥ 2 K	64 . . . 16 K	32 . . . 64 K
Topology	3d + 7th link	6d	3d + tree
Chip	CPU + FPU	CPU + FPU	2 CPU + 4 FPU
CPU core	custom VLIW	PowerPC 440	PowerPC 440
Clock	≤ 200 MHz	≤ 500 MHz	700 MHz
Peak/node	1.6 Gflops	1.0 Gflops	5.6 Gflops
Flop/clk	8 (C), 4 (V)	2	8
Mem. bandwidth	3.2 GB/s	2.6 GB/s	5.6 GB/s
Network	180 MB/s × 12	62.5 MB/s × 24 (16)	175 MB/s × 6
	150 ns	600 ns	1000 ns
	concurrent	concurrent	blocking
	1-,2-,3-step	1-step + store/forw.	cut-through
Efficiency	≈50%	≈50%	≈20%
Power/sust.	16 W/Gflops	16 W/Gflops	20 W/Gflops
Price/sust.	1 €/Mflops	1 $/Mflops	1 $/Mflops ???