apeNEXT

INFN Ferrara, Rome

DESY Zeuthen

Université de Paris-Sud, Orsay

Outline:

- Introduction
- Architecture
- Hardware
- Software and Usability
- Outlook

Introduction

2000: APEmille prototypes
ECFA ➠ O(10) Tflops in 2004

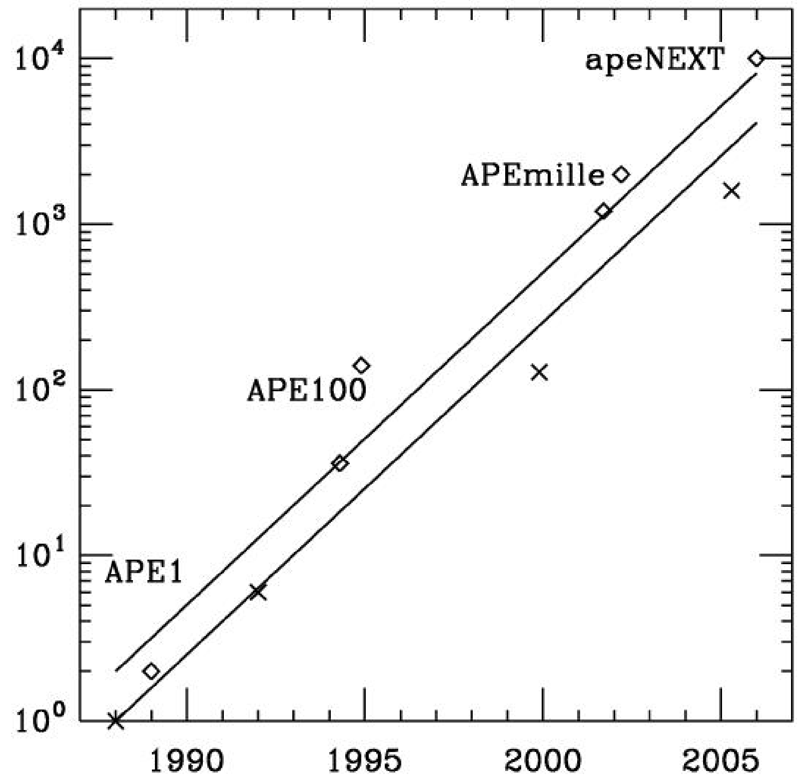
2001: MoU apeNEXT
(INFN, DESY, Orsay)

2002: APEmille installations completed

2003: apeNEXT chip sign-off

2004: Prototypes available

2005: Large apeNEXT installations



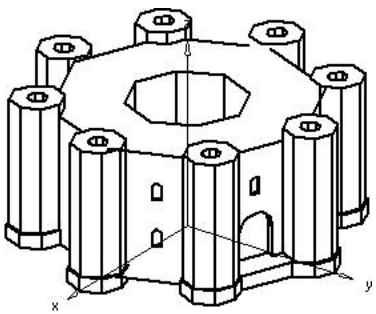
apeNEXT Aims

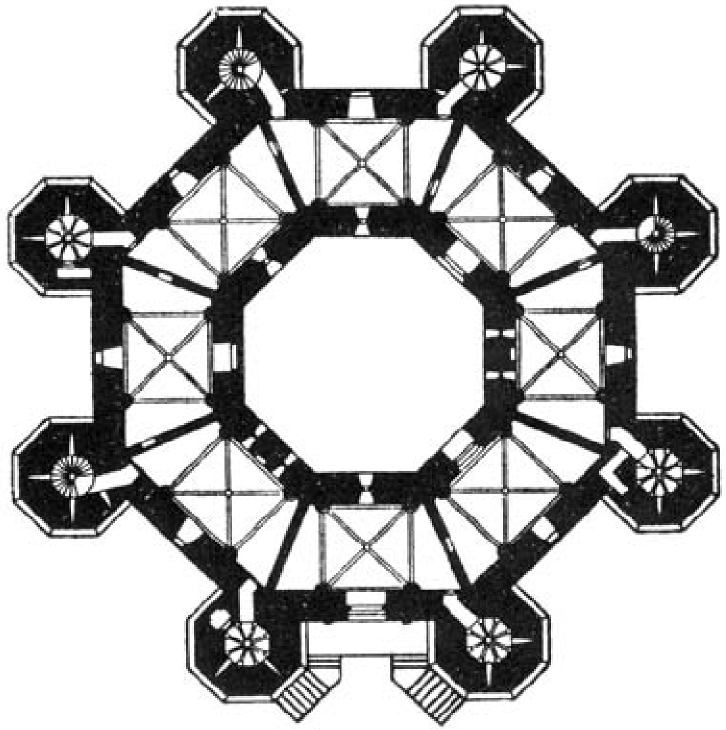
- ☞ Architecture mainly optimised for LQCD (i.e. $\approx 50\%$ sustained)
- ☞ All processor and network functionality on a single chip
- ☞ Support for C programming language
- ☞ Scalable up to tens of Tflops

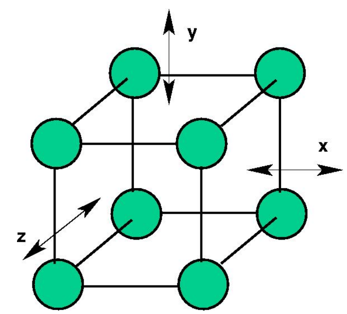
Other Machine Projects

- ❑ QCDOC (Columbia $+$ IBM)
- ❑ BlueGene ➠ BG/L (IBM)
- ❑ PC Clusters (PMS et al.)

Architecture







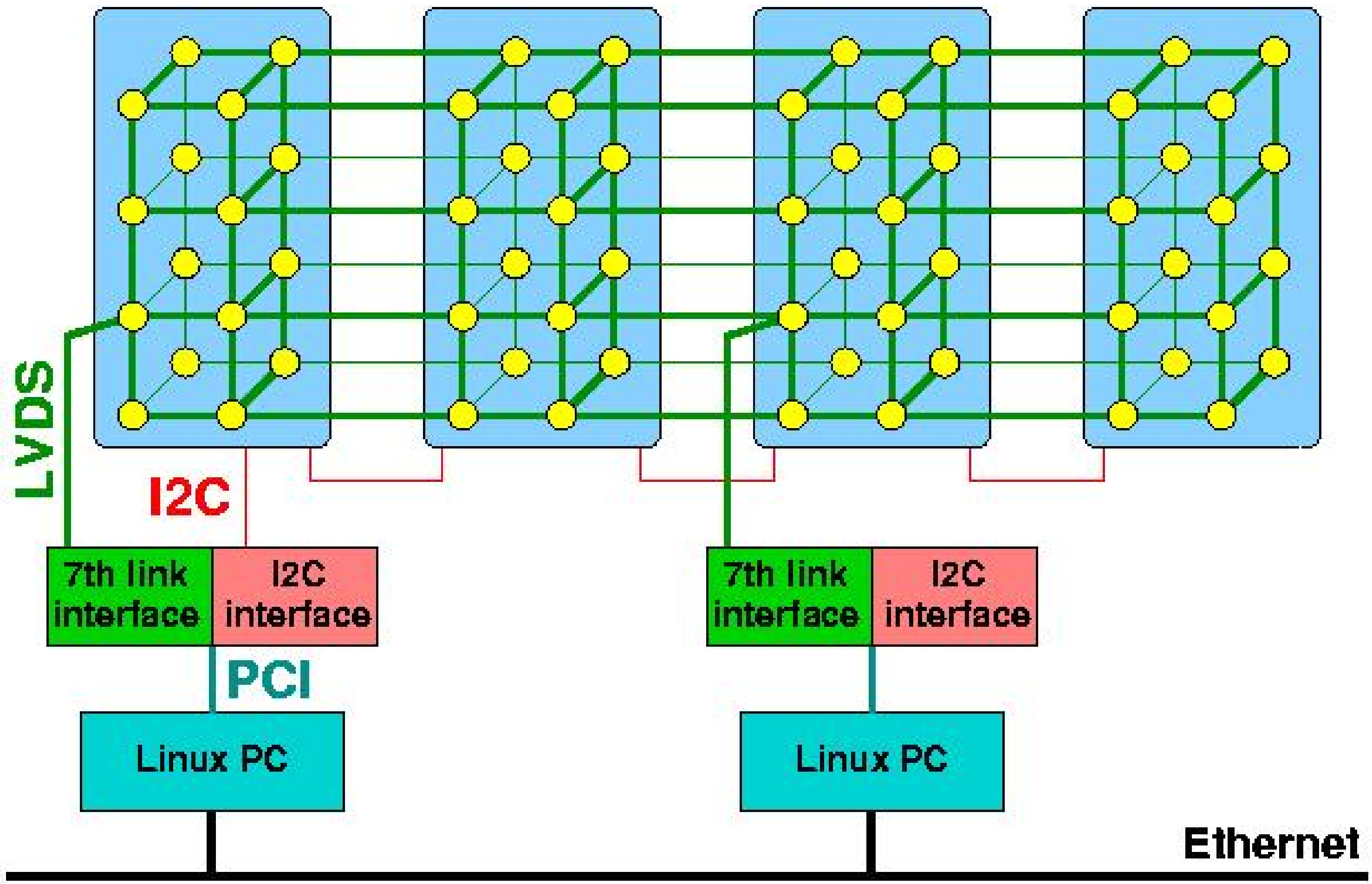
“Innovative” Architectures

(see APE1, APE100, APEmille, apeNEXT, . . . , QCDOC, BG/L)

- ☞ N-dim torus communication network
- ☞ Autonomous nodes with local on- and off-chip memory
- ☞ Integrated memory interface (with EDAC)
- ☞ Integrated communication links (with re-send)
- ☞ IO via separate nodes or hosts
- ☞ Global interrupt network
- ☞ Serial control network
- ☞ Low power consumption and high packing density
- ☞ Global clock tree
- ☞ Single user process (minimal OS, no virtual addresses)

	apeNEXT	QCDOC	BG/L
#nodes	16 . . . $\geq$ 2 k	64 . . . 16 k	32 . . . 64 k
Topology	3d	6d	3d + tree
Chip	CPU + FPU	CPU + FPU	2 CPU + 4 FPU
Clock	$\approx$ 200 MHz	$\approx$ 500 MHz	700 MHz
CPU core	custom VLIW	PowerPC 440	PowerPC 440
Peak/node	1.6 Gflops	1.0 Gflops	5.6 Gflops
Flop/clk	8 (C), 4 (V)	2	8
Network	180 MB/s $\times$ 12	62.5 MB/s $\times$ 24 (16)	175 MB/s $\times$ 6
	concurrent	concurrent	blocking
	1-,2-,3-step	1-step + store/forw	cut-through
IO/proc	$\leq$ 200/16 MB/s	100/64 MB/s	
	7th link + PC	ethernet+switch+PC	tree + IO nodes
Power/peak	6 W/Mflops		4 W/Mflops
Price/peak	0.5 €/Mflops	0.5 $/Mflops	(0.2. . . 0.3 €/Mflops)
Efficiency	$\approx$50%	$\approx$50%	$\approx$20% (LQCD)

Global apeNEXT Architecture



Arithmetic Units

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

➜ 8 Flops / cycle = 1.6 GFlops/sec

- IEEE, 64-bit floating point numbers

- 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

- ❑ 2×256 64-bit registers

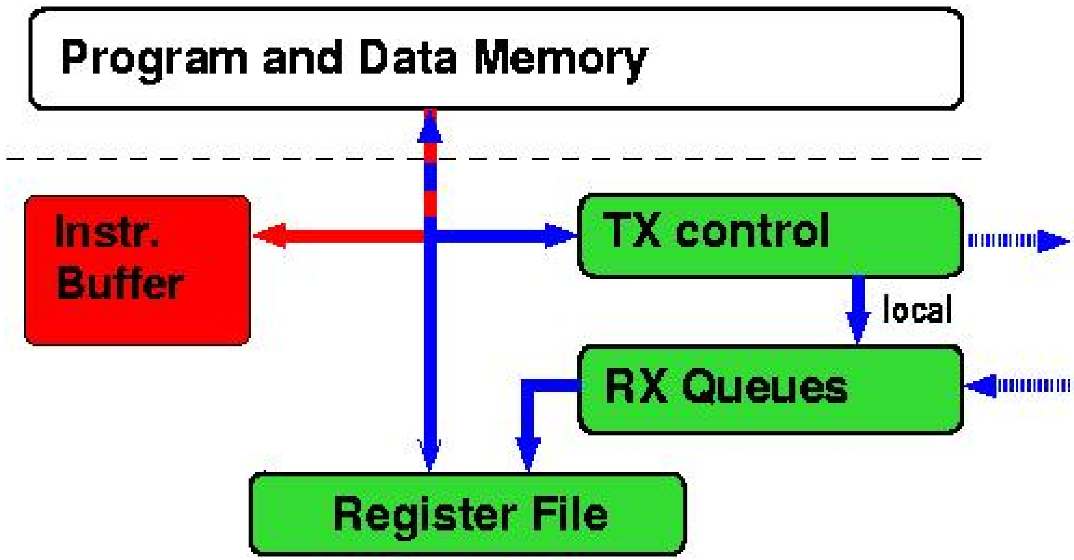
memory controller

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

➜ 2×64 **bit word/cycle** $=$ 3.2 **GBytes/sec**

- ❑ latency ≥ 16 cycles
- ❑ used for loading data and program instructions

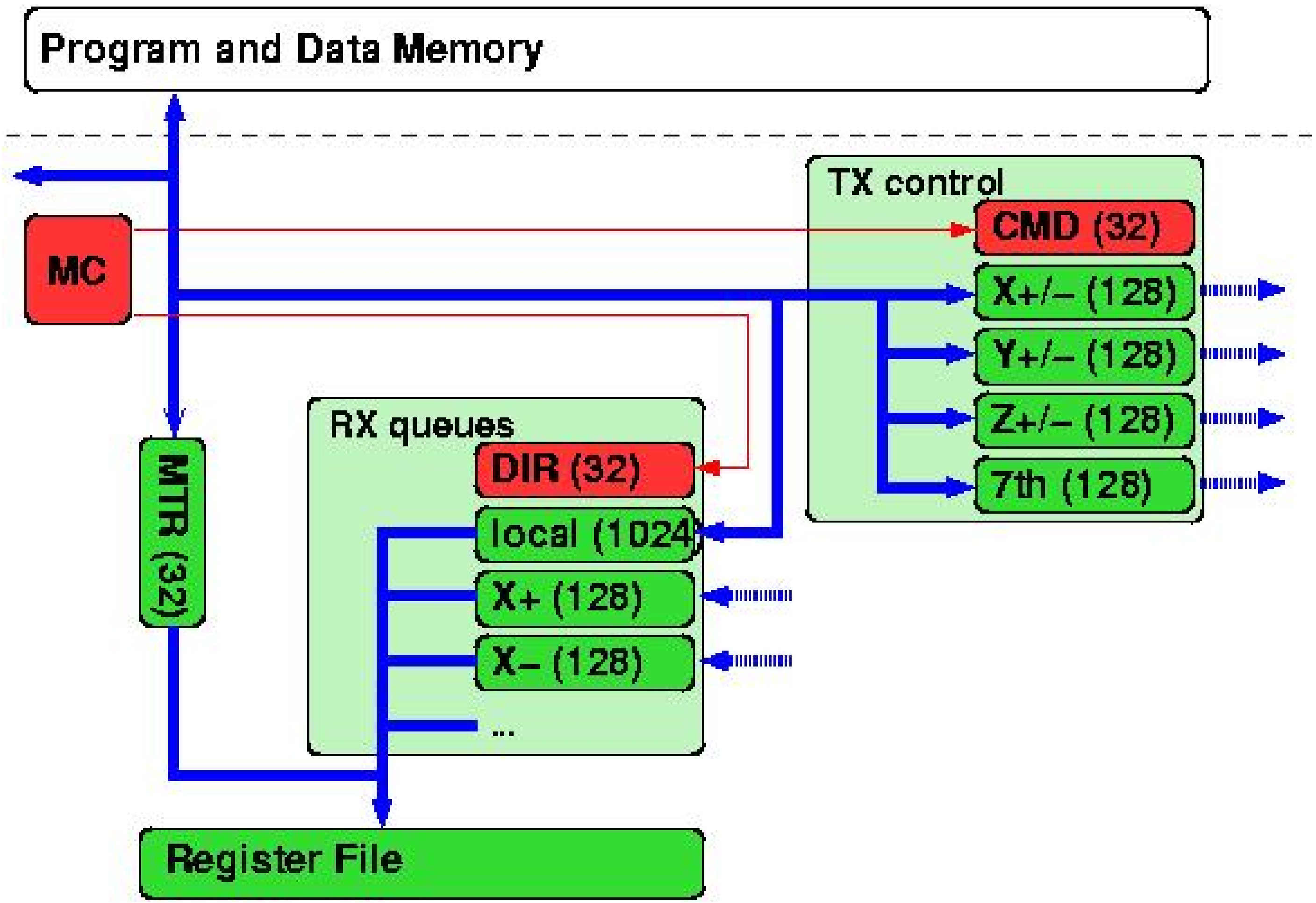
Memory Hierarchy (cont.)



instruction buffer

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

Prefetch Queues



Network

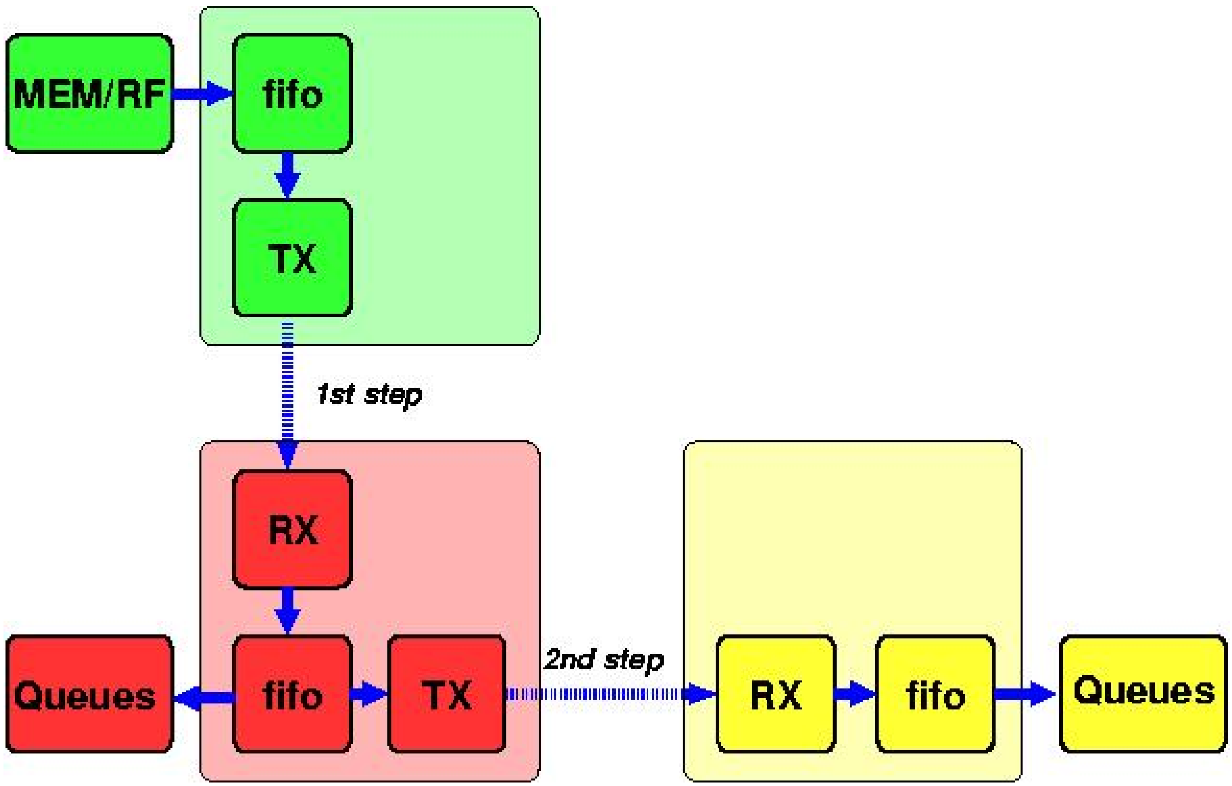
- 7 bi-directional LVDS links: $\pm x$, $\pm y$, $\pm z$, 7th
- gross bandwidth per link is one byte per clock cycle

➔ 8 **bit/cycle** $=$ 200 **MBytes/sec**

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

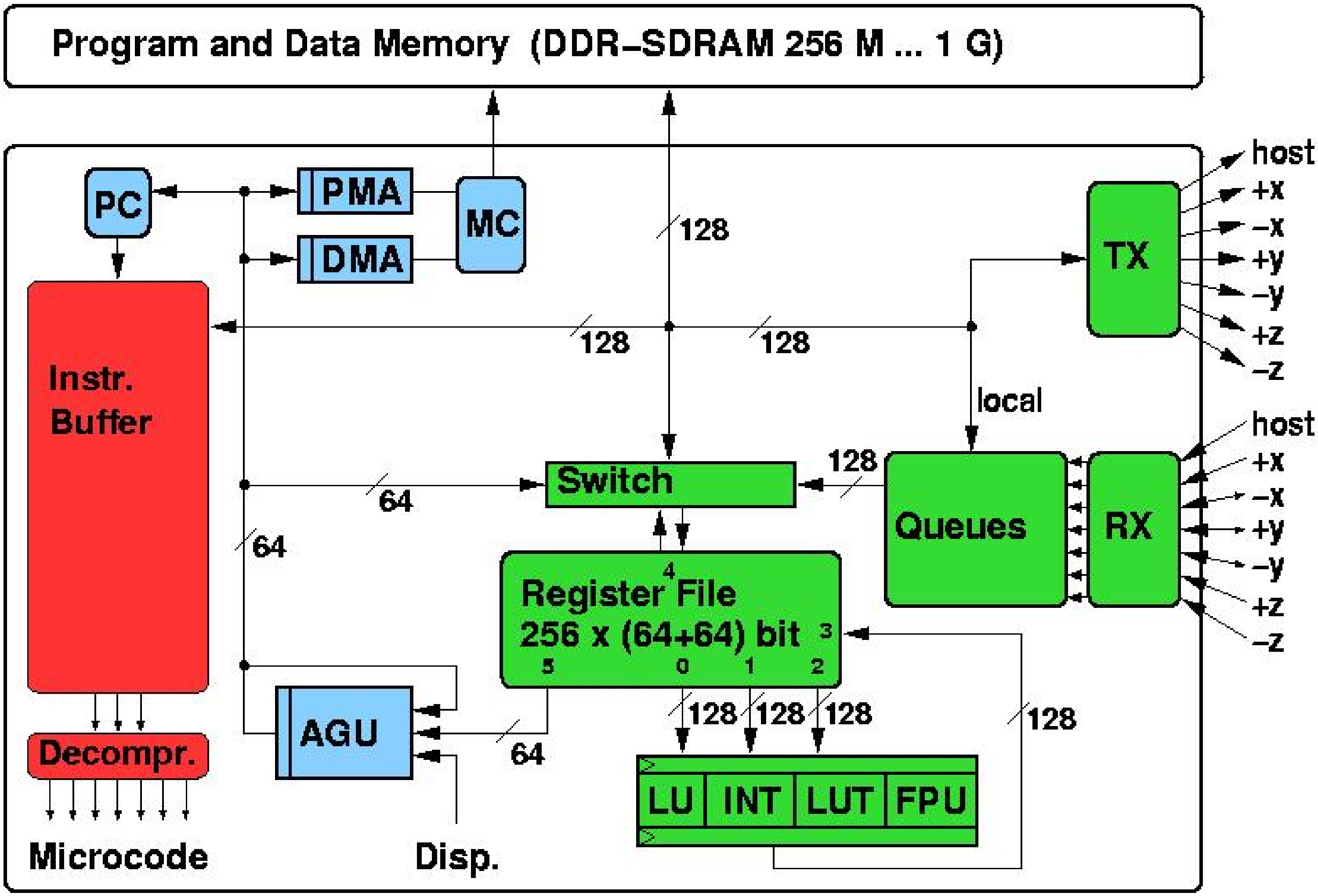
Network (cont.)

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



➔ direct access to all nodes on $3 \times 3 \times 3$ cube

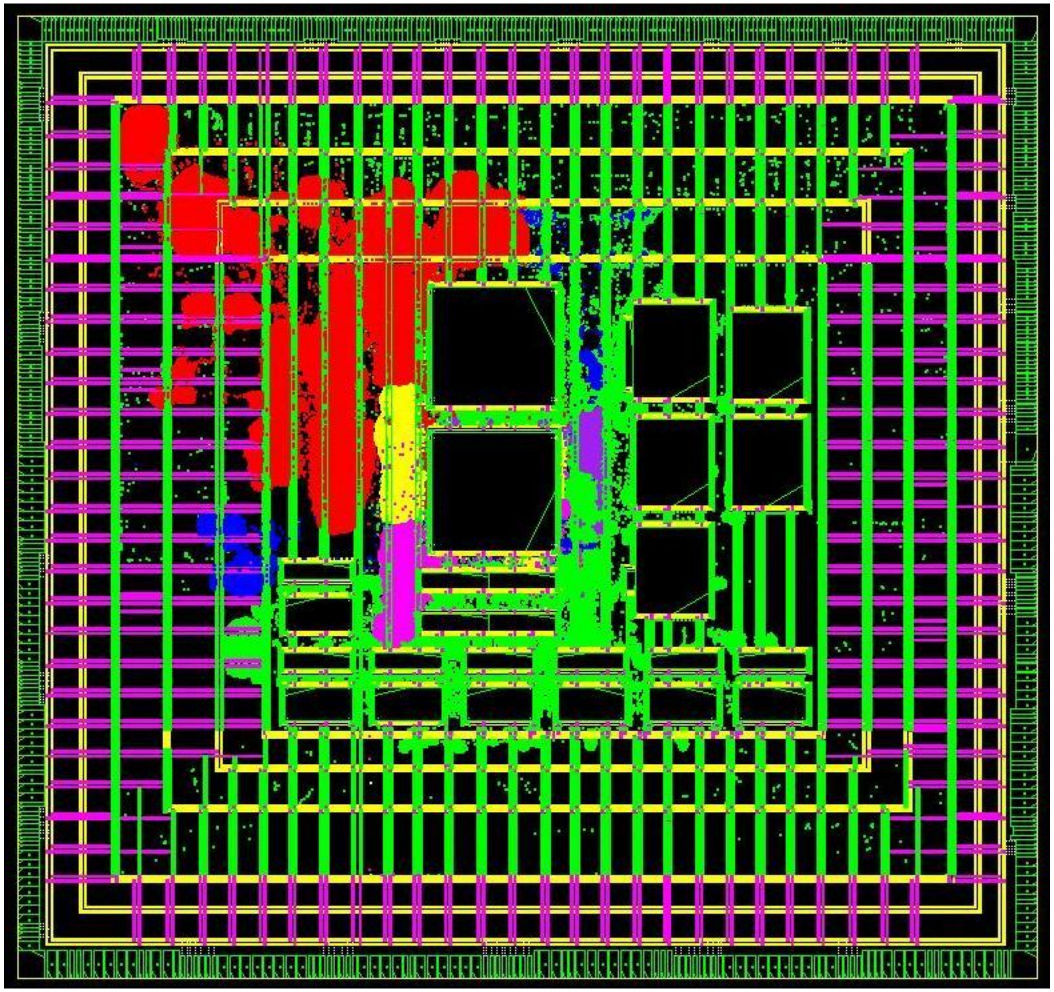
Processor Overview



Processor Design

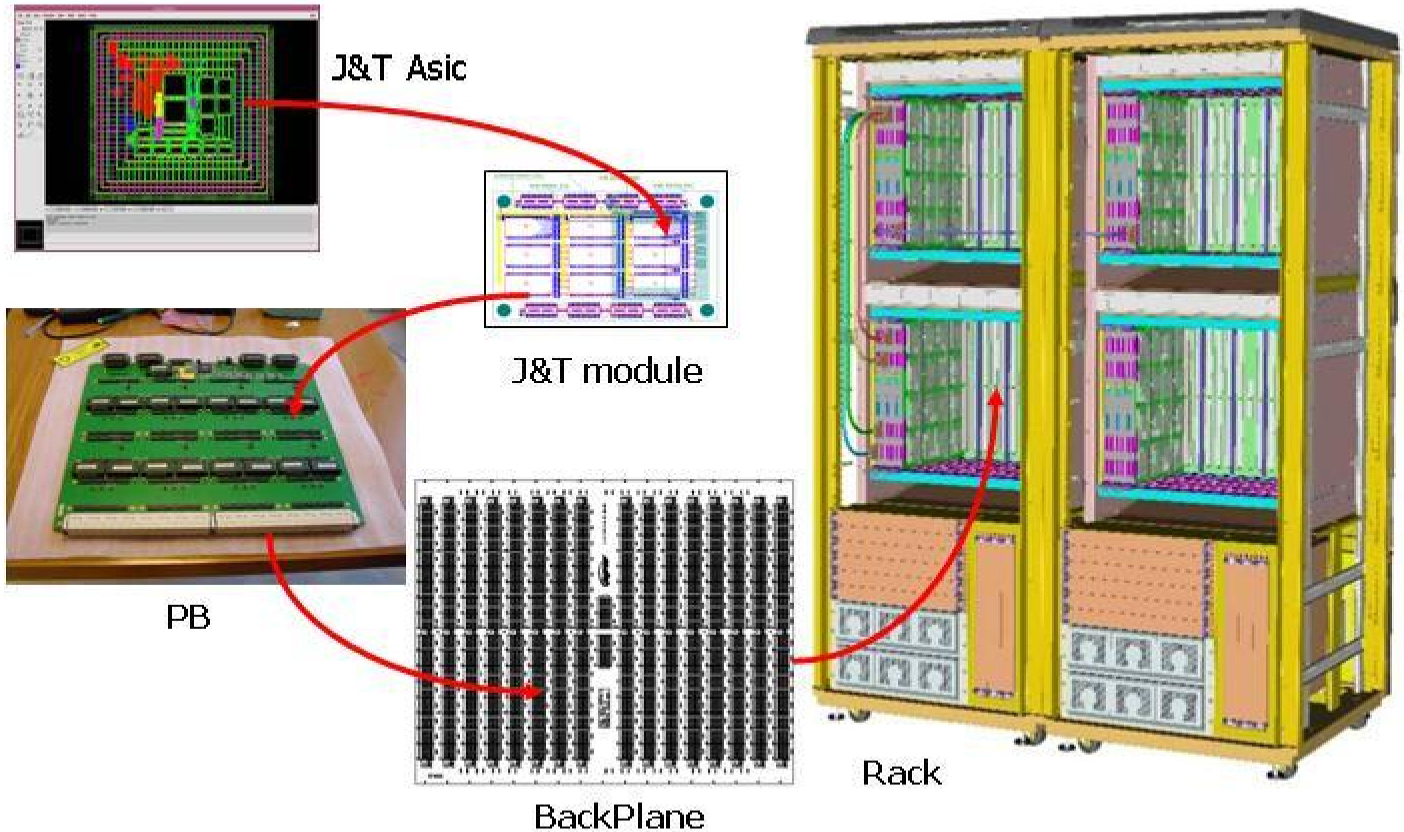
ASIC chip

- 0.18μ CMOS
- $16 \times 16\ mm^2$
- 520 k gates
- 600 pins



Hardware

Hardware Overview



Hardware Details

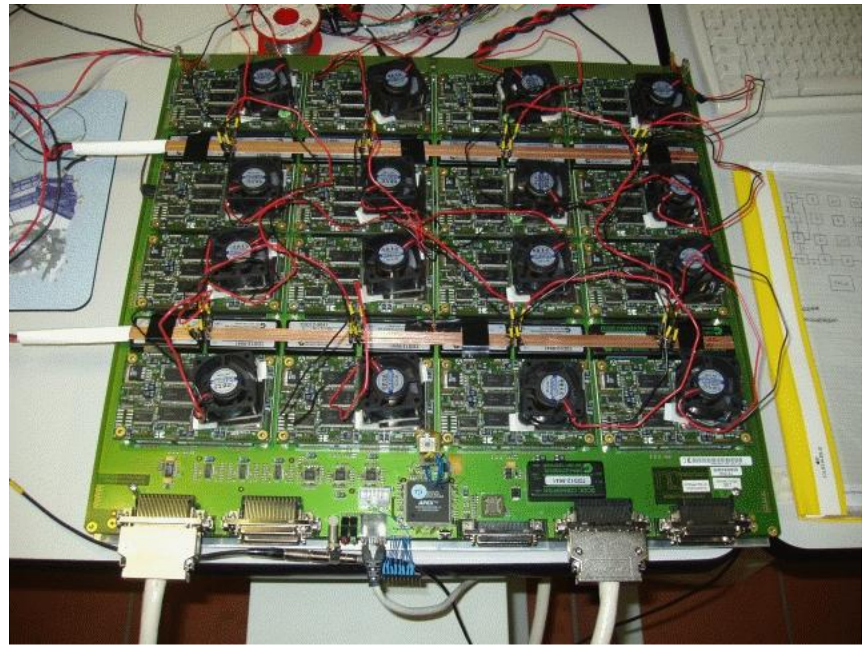
J&T Module (daughter board)

- processor chip
- 9 memory chips
- CLK 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

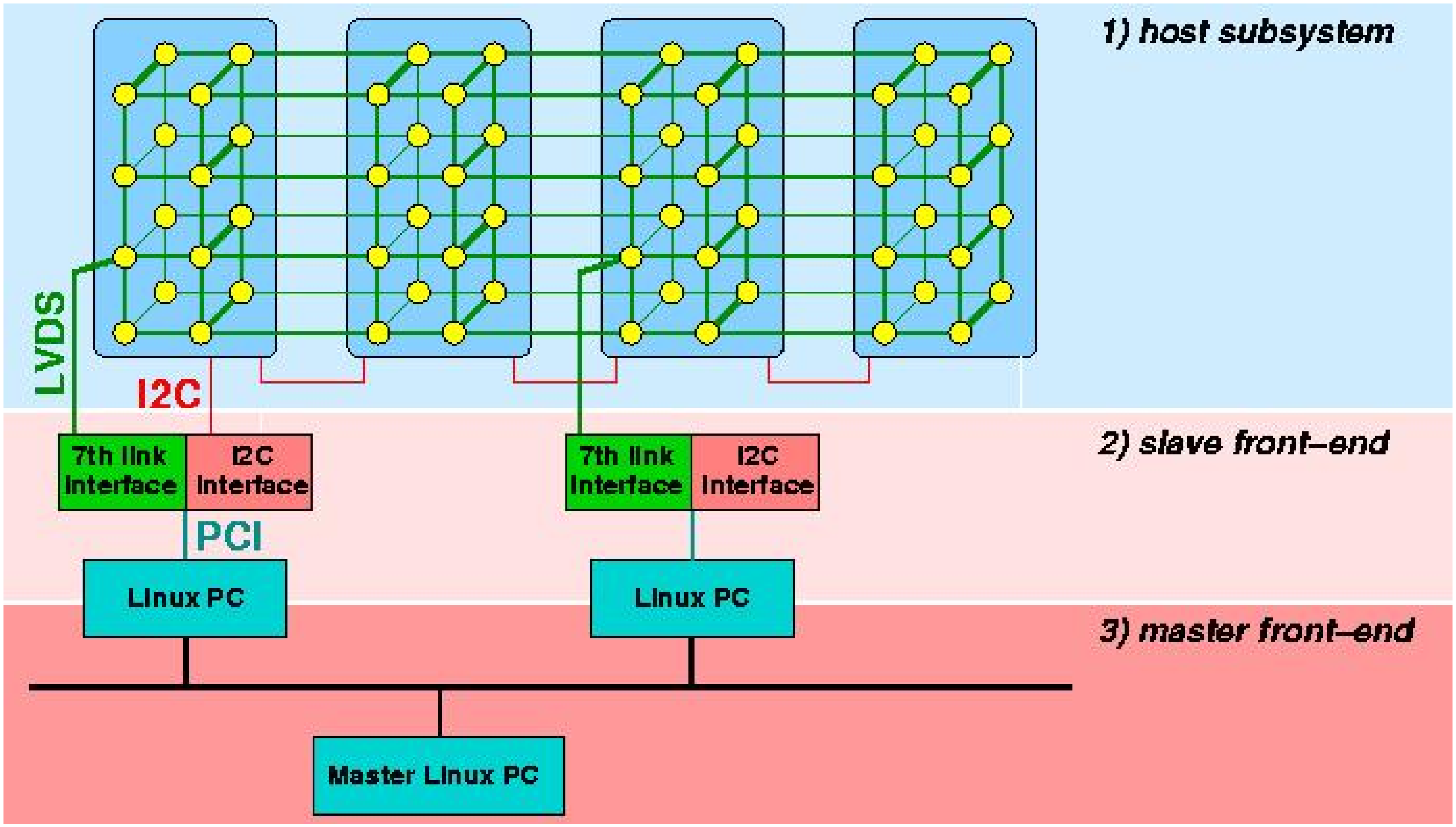
- slots for 16 processing boards
- 4600 differential LVDS signals
- 16 PCB layers

Rack

- slots for 2 backplanes
- footprint O(1 m^2)
- power consumption: 9 kW (estimated)
- air cooled
- hot-swap power supply

Software and Usability

Operating System



Operating System (cont.)

- Bootstrap, exception handling and debugging via I^2C
- Fast program loading and data IO via 7th link (BW $\sim$ #host PCs)
- 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 $8 \times 8 \times 8$
  - etc.
- Network toplologies (periodic communications):
  - x $=$ 1, 2, 4, max
  - y $=$ 1, 2, 8
  - z $=$ 1, 2, 8

Programming Languages

TAO

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

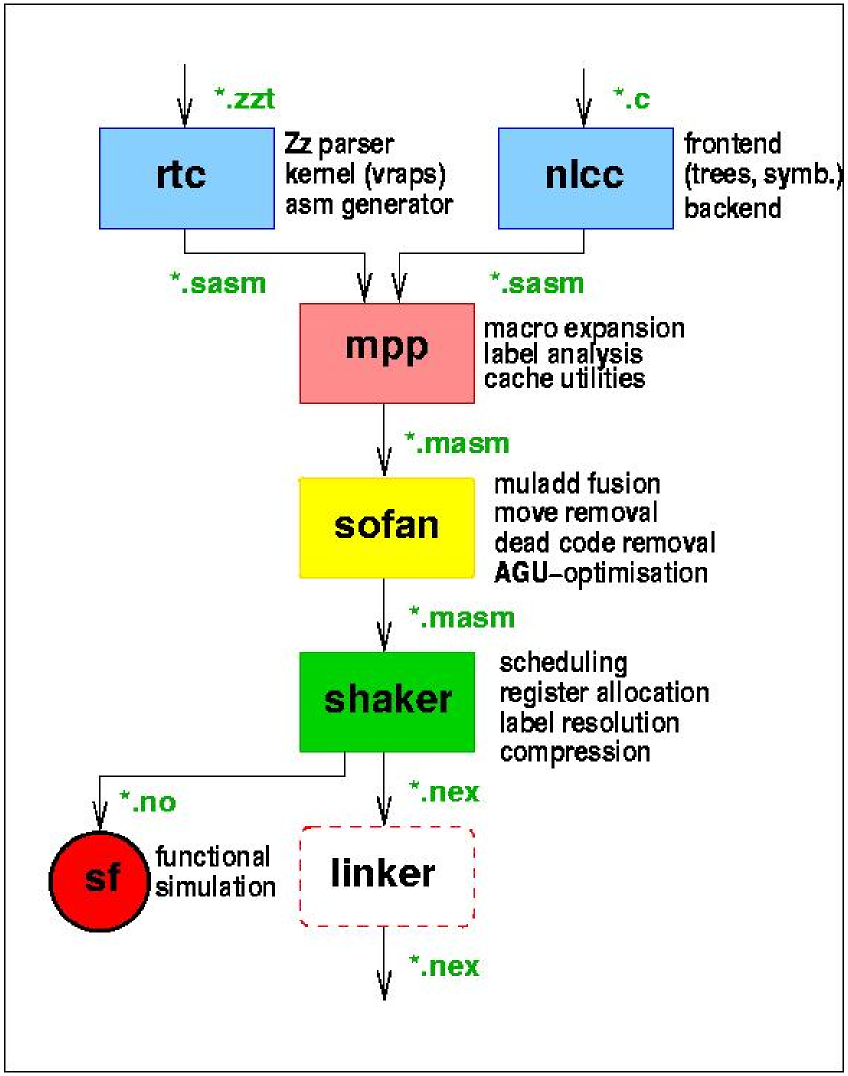
C

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

SASM

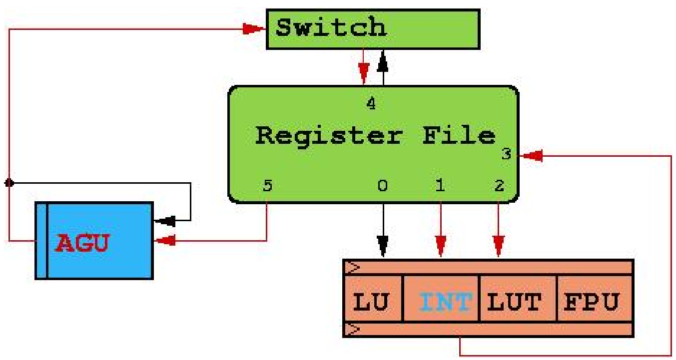
- High level assembly (e.g. for OS routines and C libraries)
- Aim: assembler programming by user not required

Compiler Overview



Assembler 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:
  - instruction pre-scheduling
  - ...



C-Compiler: Syntax Extensions

- New data types: `complex`, `vector`
- New operators: `~` (complex conjugation)
- `register struct` ➔ burst memory access
- New condition types: `where()`, `any()`, `all()`, `none()`
- `#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:

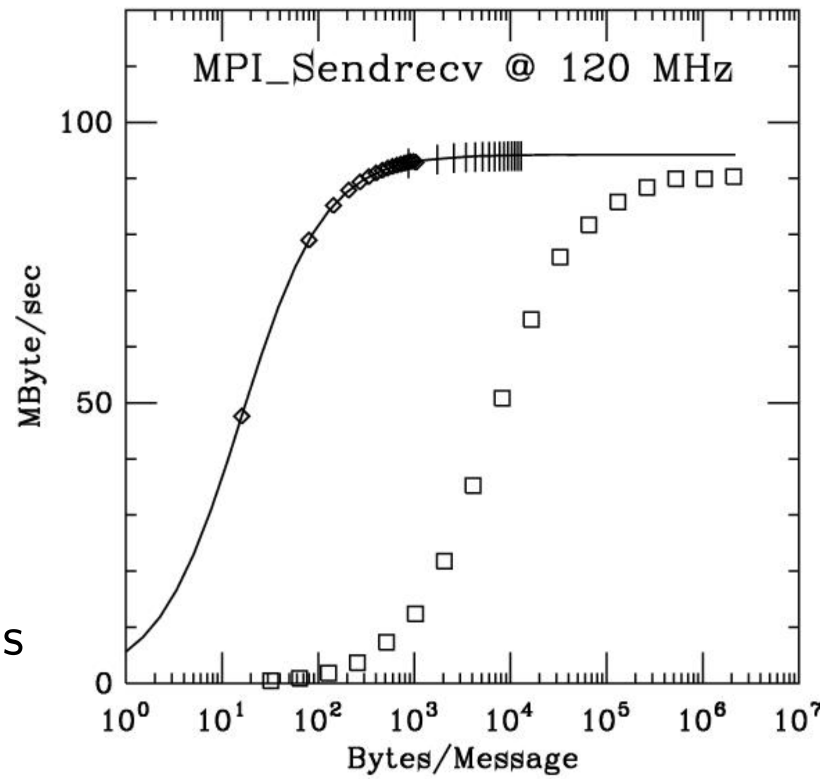
```
complex           a;
register complex  ra;

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`

C-Compiler: SPMD vs. SIMD

❑ Default: Enabled SPMD

```
int i = node_abs_id;                  // different values on each node

for( k=0; k<i; k++) { ... }           // SPMD: different flow on each node

if( i<8 ) {                           // SPMD: different flow on each node
   y = f(x);                          // SPMD: other control flow
   printf("%d",i);                    // ERROR: OS exception
   x = a[X_PLUS];i);                  // ERROR: non-homogeneous communication
}
```

Benchmarks: Linear Algebra

operation	IO-Op	Flop	sustained performance "maximum"	assembly	C	C+Sofan
`vnorm`	1	4	50%	37%	31%	34%
`zdotc`	2	8	50%	41%	28%	40%
`zaxpy`	3	8	33%	29%		
$U\ V$	27	202	92%	65%		

"maximum" 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

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

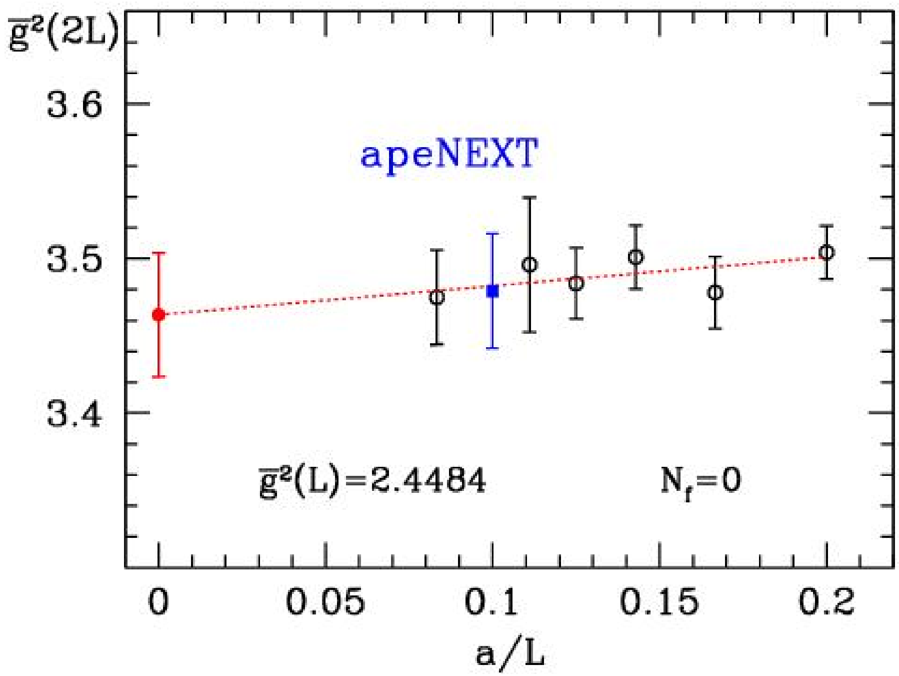
Consider worst case: local lattice size 16×2^3

Measured sustained performance:	**55%**
Measured number of stretch cycles:	**4%**

Optimization "tricks":

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

Physics Tests



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

Status

❑ HW components:

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

❑ SW elements:

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

❑ Two prototype systems with 64 nodes

❑ Successful test runs with physics codes

Installation Plans

- "Huge Prototype" (1.6 Tflops) expected in 3/2005
- Funding for 5 Tflops @ INFN approved
- Tender for large machines @ 0.5 €/Mflops completed in 9/2004
- Additional for 5 Tflops @ INFN planned
- Installations planned by Bielefeld, DESY and Orsay

Outlook

- tune prototype HW to push speed and stability
- complete remaining SW to push usability
- exploit full potential of apeNEXT architecture
- explore other (non-LQCD) applications?
- future dedicated machine developments?