



# L'esperimento apeNEXT dell'INFN dall'architettura all'installazione

Davide Rossetti

I.N.F.N Roma - gruppo APE\*

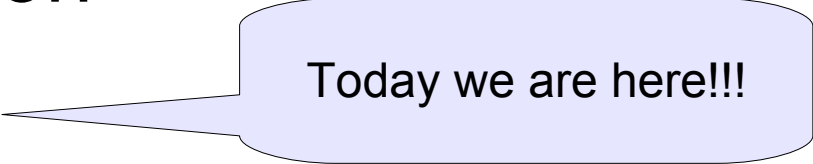
[davide.rossetti@roma1.infn.it](mailto:davide.rossetti@roma1.infn.it)

\*<http://apegate.roma1.infn.it/APE>



# Index

- ♦ The problem
- ♦ The solution(s)
- ♦ Our own solution: apeNEXT
  - Architecture
  - Implementation
  - Deployment



Today we are here!!!



# The Problem

Basically non-perturbative Lattice QCD ends up in solving (many times) the following equation:

$$(1 - \kappa D_{xy}[U])\varphi = \psi$$

where:

- $\varphi$  and  $\psi$  are (pseudo) quark fields ( $V \times T \times 4 \times 3$ )
- $U$  are gluon fields ( $V \times T \times 3 \times 3$ )
- $D$  is the Dirac operator matrix
- $\varphi$  is unknown  $\rightarrow$  inverter algorithm



# The solution(s)

Hard computational problem !!!  
Needs lots of calculations

→ Very Fast Computers = Parallel supercomputer

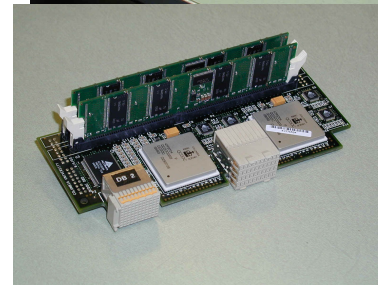
- Two options:
  - Great number of modest, cheap processors.
  - Moderate number of fast, fat processors.
- Connected by an efficient interconnection network.



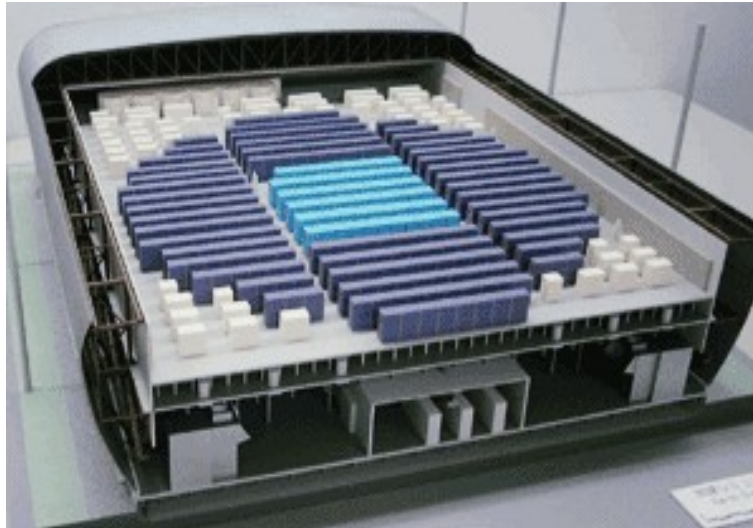
# Solutions by others: Earth Sim, QCDOC, BG/L



BG/L – LLNL  
(Oct 06)



Earth Simulator  
– Yokohama  
(May 02)



QCDOC – Riken  
– BNL (Mar 05)



# Our solution: apeNEXT

~ 1000's of simple, self designed, low power processors optimized for LQCD





# ApeNEXT HW

## Project HW milestones:

- Start *thinking* in 2000
- Cleared out main details during Fall 2001
- Proposal delivered Oct 2001
- HW Feature freeze in 2002
- Extensive VHDL tests up to Summer 2003



# ApeNEXT Design & Architecture

New challenges need new ideas (Spring 2000):

- Keep old good APE ideas.
- Design by benchmarking.
- New parallel paradigm.
- Allow for latency hiding.





# ApeNEXT Design

## Keep good old APE ideas

- No data cache
- Large, custom designed, multi-port register file.
- Very Long Instruction Set (VLIW), microcode optimized in software at compile time.
- Distributed memory with communications as memory access.
- Solid, low error rate, low latency network interconnect.
- First, second & third neighbor communication primitives.



# ApeNEXT Design

## Design by benchmarking

- A reconfigurable VLIW **code optimizer** was written and exercised with linear algebra and LQCD computational kernels.
- Some different architectures were **benchmarked**:
  - 1 or 2 independent FPUs ? Conservative, just 1.
  - How many Register File ports ? 6
  - # FP registers ? 256 complex
  - How many AGUs ? 1
  - # AGU registers ? 64, shared



# ApeNEXT Design & Architecture

## Parallel Paradigm

- APEmille taught us to watch for memory and network latencies hidden in the APE programming model.
  - Higher target clock than before ~200MHz.
  - Rethinking of classic SIMD APE architecture.
- Say goodbye to SIMD. Welcome MIMD.  
Processors are **loosely synchronized**.
- No Harvard architecture! Unified Prg & Data memory.  
Use on-chip **program memory buffer**.



# ApeNEXT Design

## I/O Data latency hiding

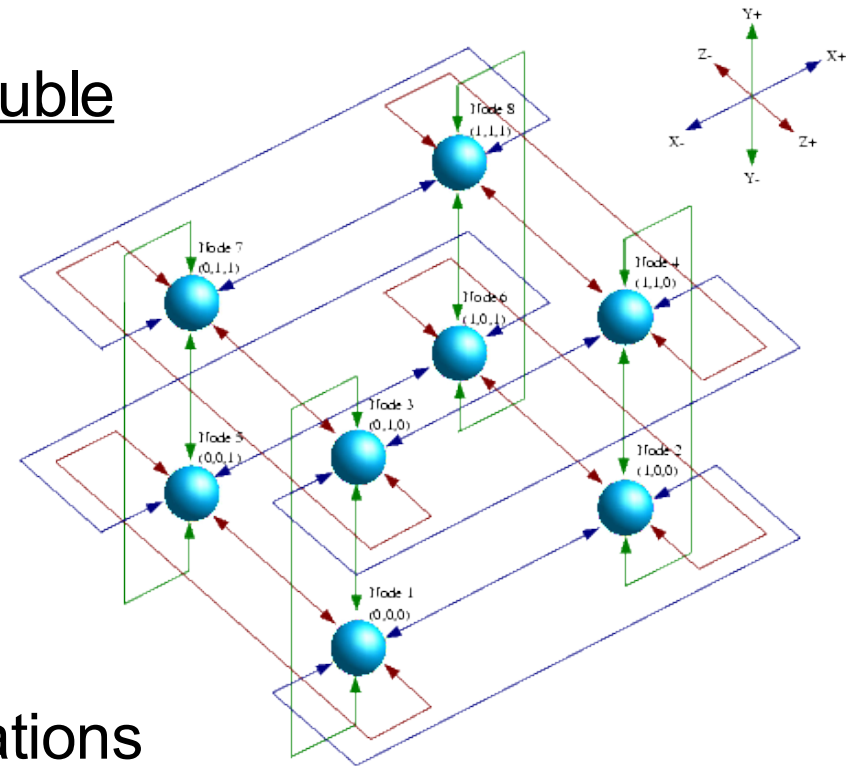
- DDR memory, high bandwidth but ~ 10 cycles latency.
  - Network link ~ 25 cycles latency.
- add on-chip memory buffers as fast scratch pads.
- add pre-fetching instructions.

Overlapping calculation with data I/O is needed for both memory and networking.



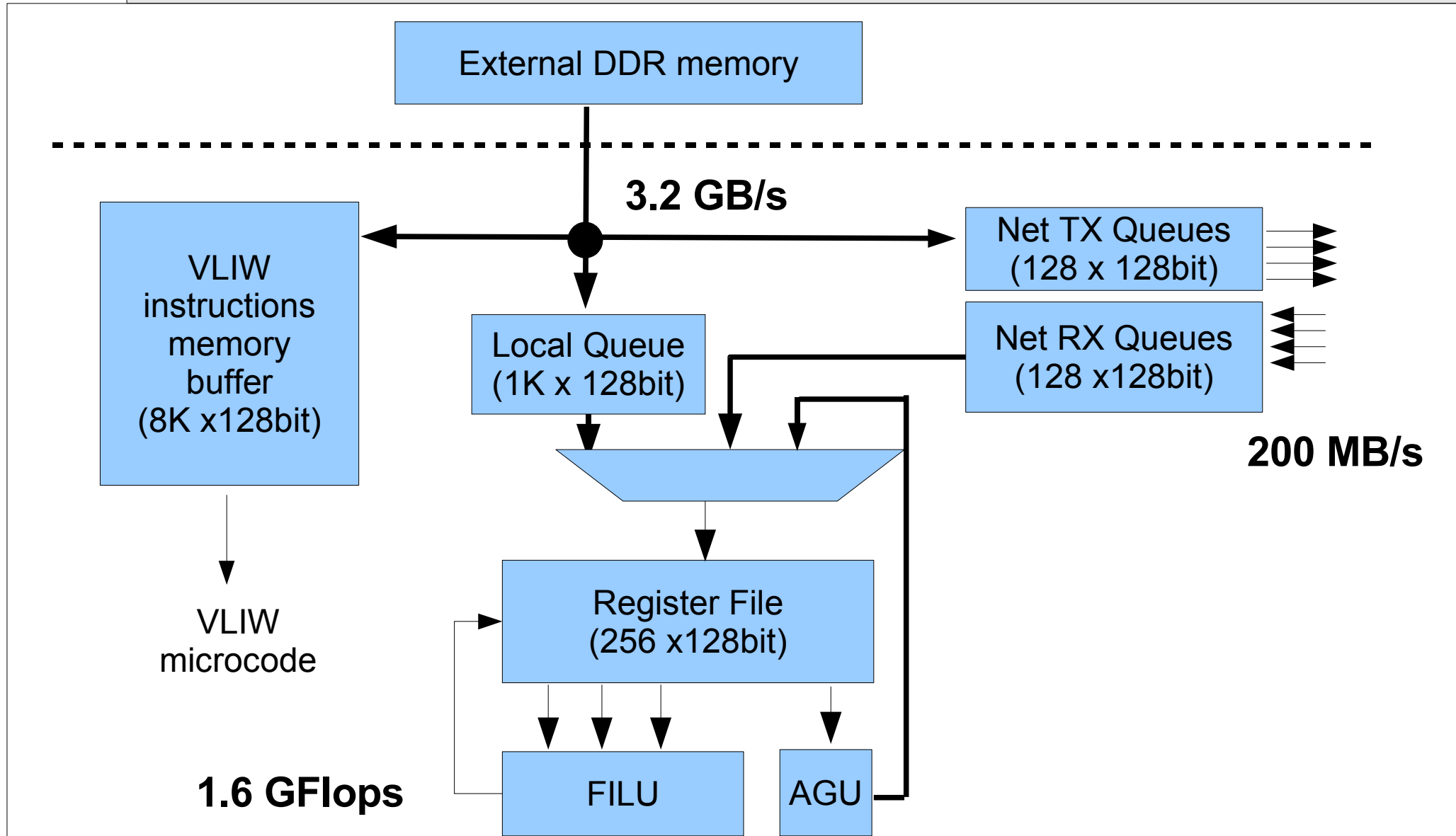
# ApeNEXT Architecture

- 3D mesh of computing nodes
- Vertexes are processors:
  - Each proc hosts its local memory
  - Each proc supports **64bit** complex, double and integer types
- Edges are 3D torus network channels
  - 6 bi-dir channels per proc
  - Basic comm primitive is first-neighbor send-recv
  - Processors synchronize on communications (send starts when recv is issued)





# ApeNEXT Architecture the J&T modules

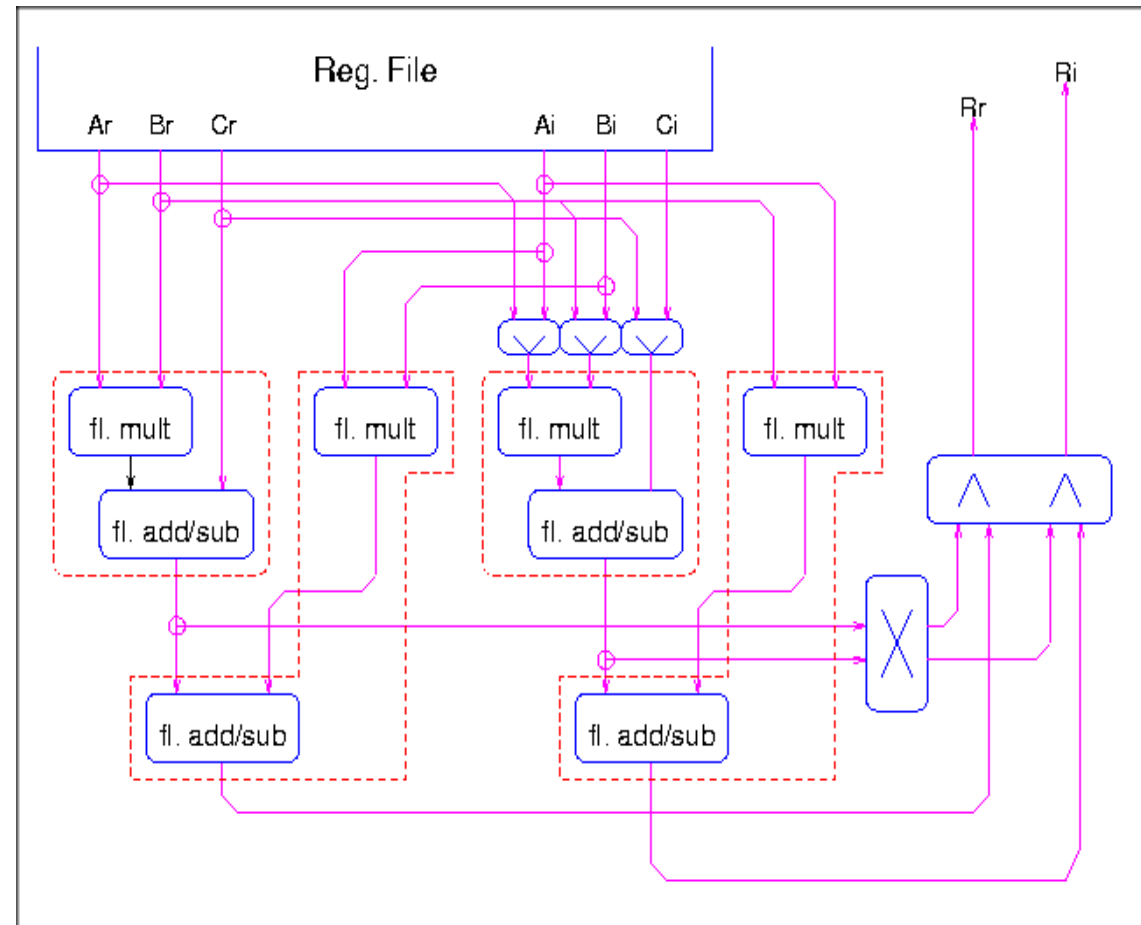




# ApeNEXT Architecture the J&T FILU

FILU is FP, Integer and Logical unit:

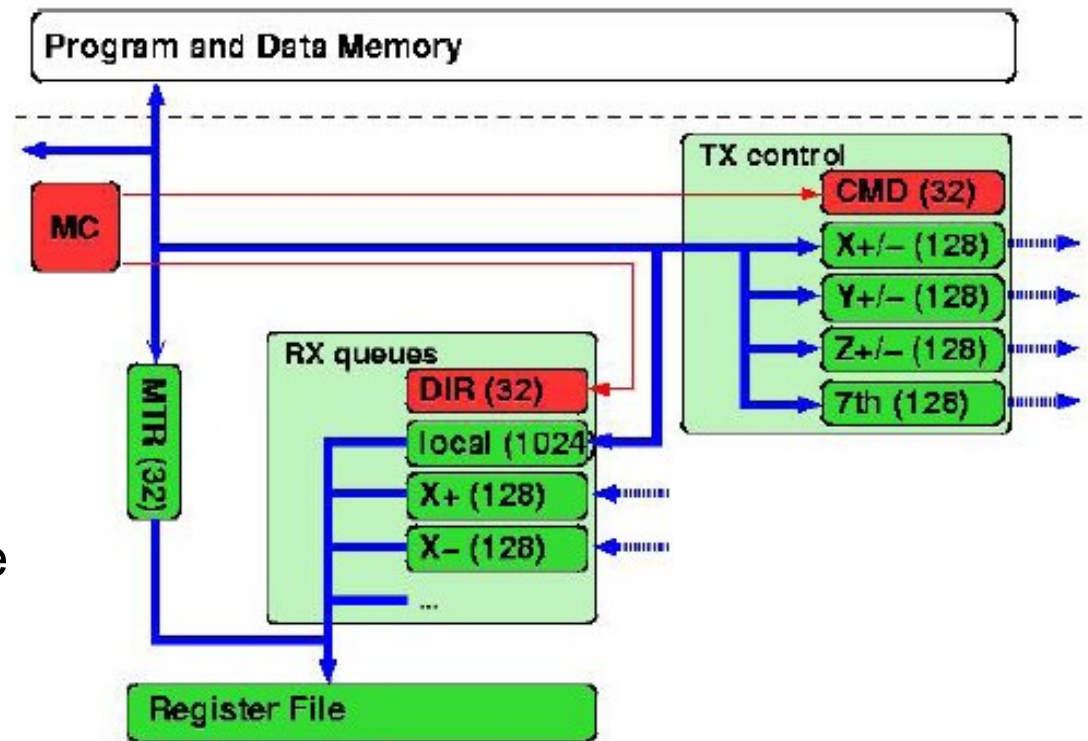
- MAC op:  **$A*B+C$**
- fully **pipelined**  
(1 result per cycle)
- ~ **12 cycles** latency
- synthesizes to **200MHz**
- 4 multipliers
- 4 adders
- **1.6GFlops** on complex MAC





# ApeNEXT Architecture the J&T network

- 6 3D channels: X+,X-,Y+,Y-,Z+,Z-
- 7<sup>th</sup> I/O channel
- LVDS 8+1 bit @200 MHz signaling
- Custom low-level protocol
  - 8bit header
  - 128bit data
  - 16bit CRC
  - ack/nack tokens
  - Time windows for acks
- 200 MB/s raw bandwidth
- 170 MB/s effective bandwidth
- 25 cycles activation latency
- Up to 6 channels concurrently active







# ApeNEXT Implementation

Atmel Silicon Fab for J&T production

APW for BP design/production

Eurotech/Exadron for all PCB design/production

J&T Chip Schedule:

- Design-out by APE on Jul 20<sup>th</sup> 2003
- Back-end by Atmel on Sept 10<sup>th</sup> 2003
- Sample J&T proto shipped by Dec 2003
- First tests on Jan 10<sup>th</sup> 2004
- First apps running end of Jan 2004



# ApeNEXT Implementation the J&T processor

Chip design flow:

- Over 55000 VHDL lines
- Synopsys synthesizer on CMOS
- Virage memory macro-cells
- Synopsys placement tool (Physical)
- Cadence for routing and back-annotation
- Hand-written, mostly functional test vectors
- Some tests via TetraMax

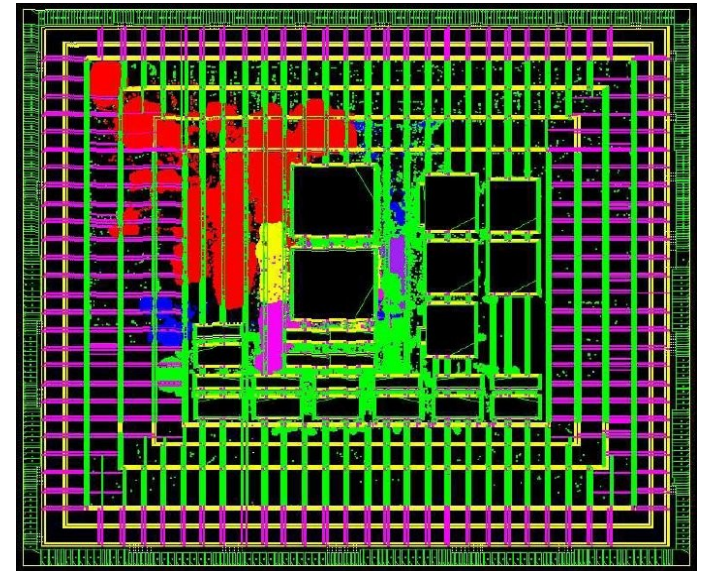




# ApeNEXT Implementation the J&T processor

J&T physical implementation:

- 0.18  $\mu$  / 5 layers CMOS by ATMEL
- 520K gates: 160K FPU, 240K Register File, 120K glue
- 450 used pins
- 600 pin BGA package
- LVDS pads for network channels
- SSTL pads for DDR memory
- 16mm<sup>2</sup> die area, **pad limited**
- Dissipates 5W at 200MHz





# ApeNEXT Implementation the J&T performance

comparing BG/L with apeNEXT

→ play the “performance sustained on algorithm” game

	N. Proc	Rpeak(TFs)	Rsust(TFs)	GF/proc(P)	GF/Proc(S)
Blue/Gene L	65536	367	73	5.6	1.12
apeNEXT	8192	13.2	8.0	1.6	0.97

on LQCD:

- BG/L runs 20% sust.perf. using hand-tuned assembly code + MPI (29% using low-level messaging)/
- ApeNEXT runs 60% sust.perf. using high-level language (TAO & C).



# ApeNEXT implementation

## The Board and the Module

Power: 340W

PCB on custom metal frame

Populated board: 5Kg

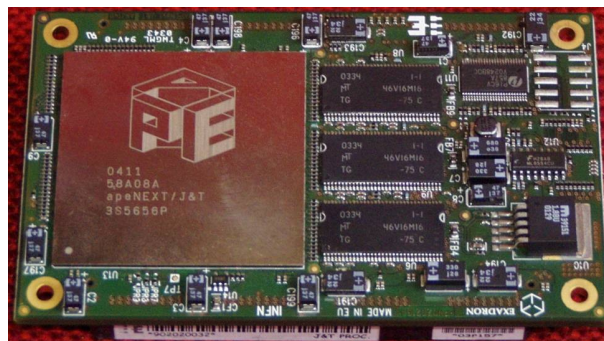
Connectors Insertion Force: 80-150Kg

Detailed Air Flow simulations

16 piggy back J&T modules

1 Altera APEX EB20K100

16 DC-DC converters 20W 48→2.5V

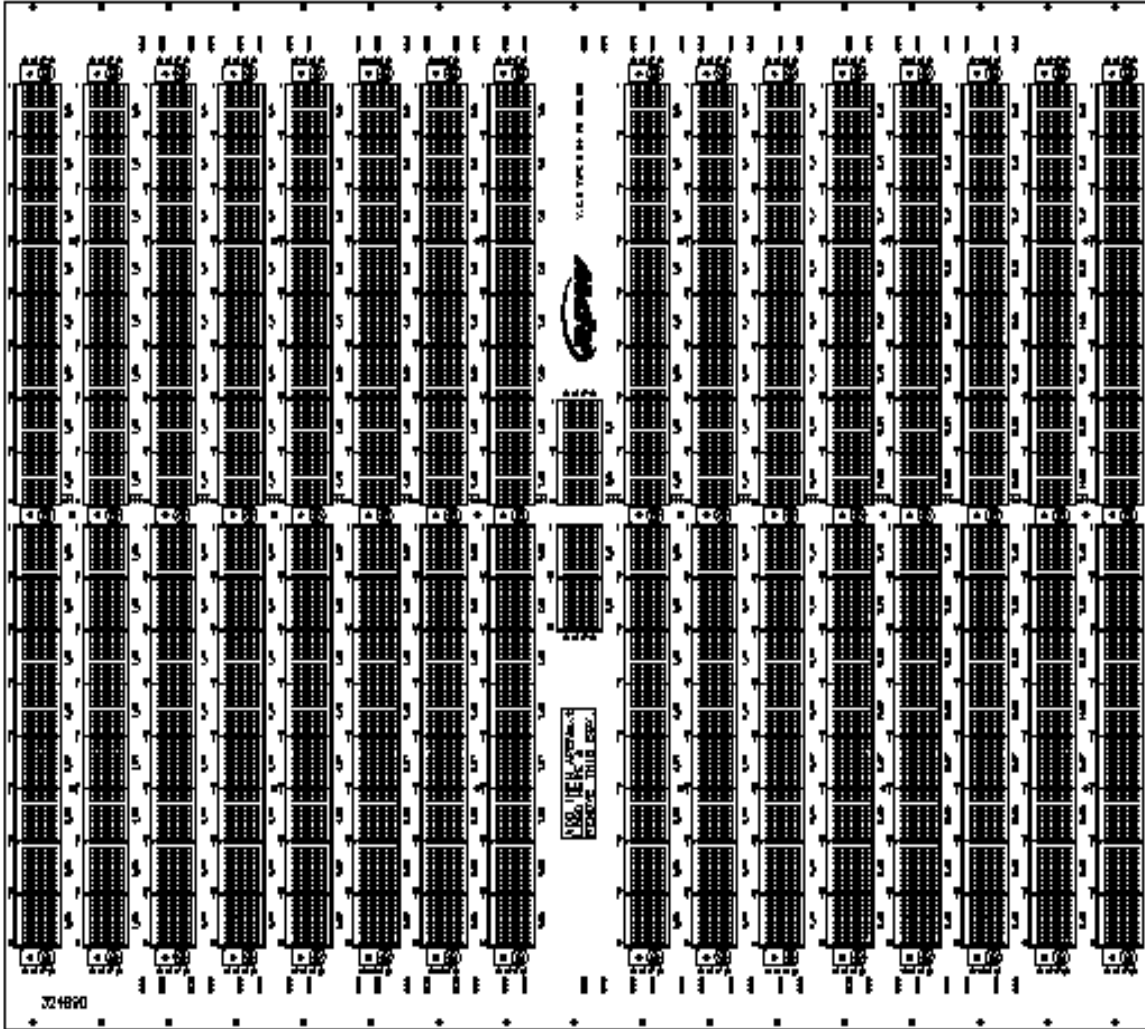






# ApeNEXT implementation

## The Backplane



Slots: **16+1**

Dim: **447x600mm<sup>2</sup>**

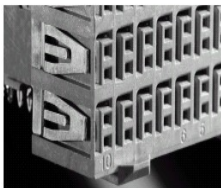
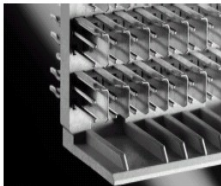
Wires: **4600** LVDS p2p

Layers: **16 + 16** controlled impedance

Speed: **600Mb/s**

Connectors kit: **~5KEur**

Insert.Force: **80-150Kg**

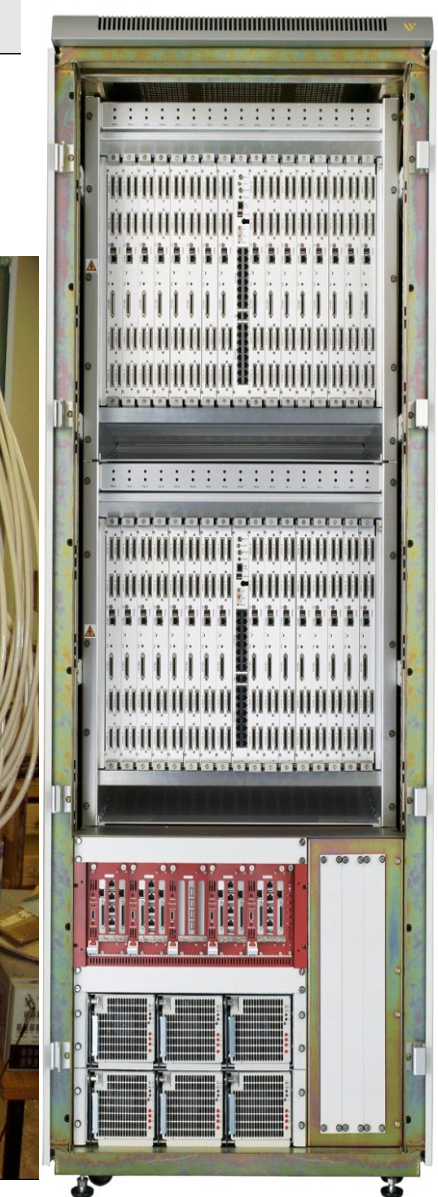




# ApeNEXT implementation

## The Rack

- 2 crates, 32 Board
- 42U rack system
- 2.1 m tall
- EMC tested
- 9 19" 1U slots
- Dual 48V150A hot-swap PSU
- Custom card cage & air system
- 8X8x8=512 Processors
- @160MHz = 655GFlops

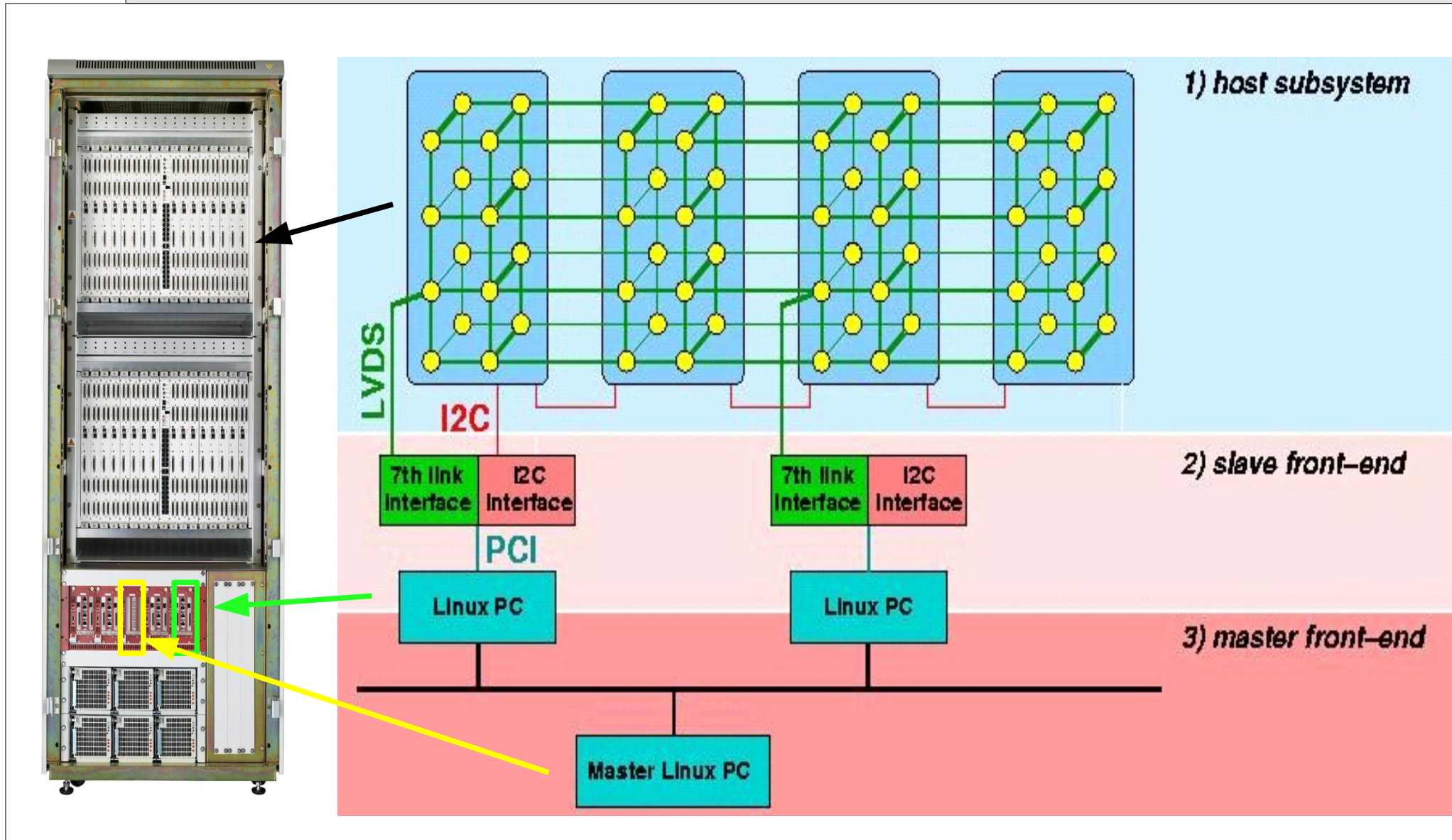






# ApeNEXT implementation

## The Rack







# ApeNEXT Deployment

- Feb 2005: first proto 4x8x8 crate running physics
- Jun 2005: start mass production
- Oct 2005: first 8x8x8 rack running physics
- Dec 2005: install site ready in Rome
- Jan 2006: 7 racks installed
- Jun 2006: 11 racks installed



**Integrated Peak Performance**  
**11 racks @ 160MHz ~ 7.2 Tflops**  
**(1TFlops =  $10^{12}$  floating point ops)**



# What's next apeNEXT ?

- World changed in 2004/2005 when Intel admitted “no *more* 4 GHz P4”... how could others ?
- Marketing refocused on *multi-cores* (Core Uno, Core Duo, ...).
- All CPUs are parallel processors today.
- Next big issue is how to program Multi-core CPUs easily!!!
- All programmers are expected to develop parallel programming skills !!!
- Meanwhile, silicon fabs keep on increasing transistor count.



# What's next apeNEXT ?

