Overview of the QCDOC supercomputer

Tilo Wettig

Yale University and RIKEN-BNL Research Center

for the QCDOC collaboration

The QCDOC design team

Columbia:	N. Christ, C. Cristian, C. Kim, X. Liao, G. Liu, R. Mawhinney, A. Yamaguchi
UKQCD:	B. Joo, P. Boyle
RIKEN-BNL:	S. Ohta (KEK), T. Wettig (Yale)
SciDAC:	C. Jung, K. Petrov
IBM Yorktown:	A. Gara, M. Liu, B. Nathanson (also BlueGene/L manager/designers)

Introduction

- target application: lattice QCD (discretize space-time in 4 dimensions, compute Feynman path integral nonperturbatively)
- computational demands require massively parallel computers (global volume distributed over many processors)
- dominant operation: inversion of a large sparse matrix (dim$\sim 10^7 - 10^8$)
- done by conjugate gradient algorithm, two main ingredients:
  – matrix-vector product (requires n.n. communications)
  – global sum (requires global communications)
- predictable and regular communication patterns
  $\longrightarrow$ relatively easy to parallelize

- main design goal is scalability:
  - keep physical problem size fixed
  - increase number of processors (i.e. computing power)

  $\longrightarrow$ local volume decreases, communication effort increases

  How does overall sustained performance scale with local volume?

- communication performance determined by two main factors:
  - bandwidth
  - latency
- for small local volumes (large surface-to-volume ratio), latency is dominating factor

  $\longrightarrow$ try to hide the latency (overlap communication with computation)

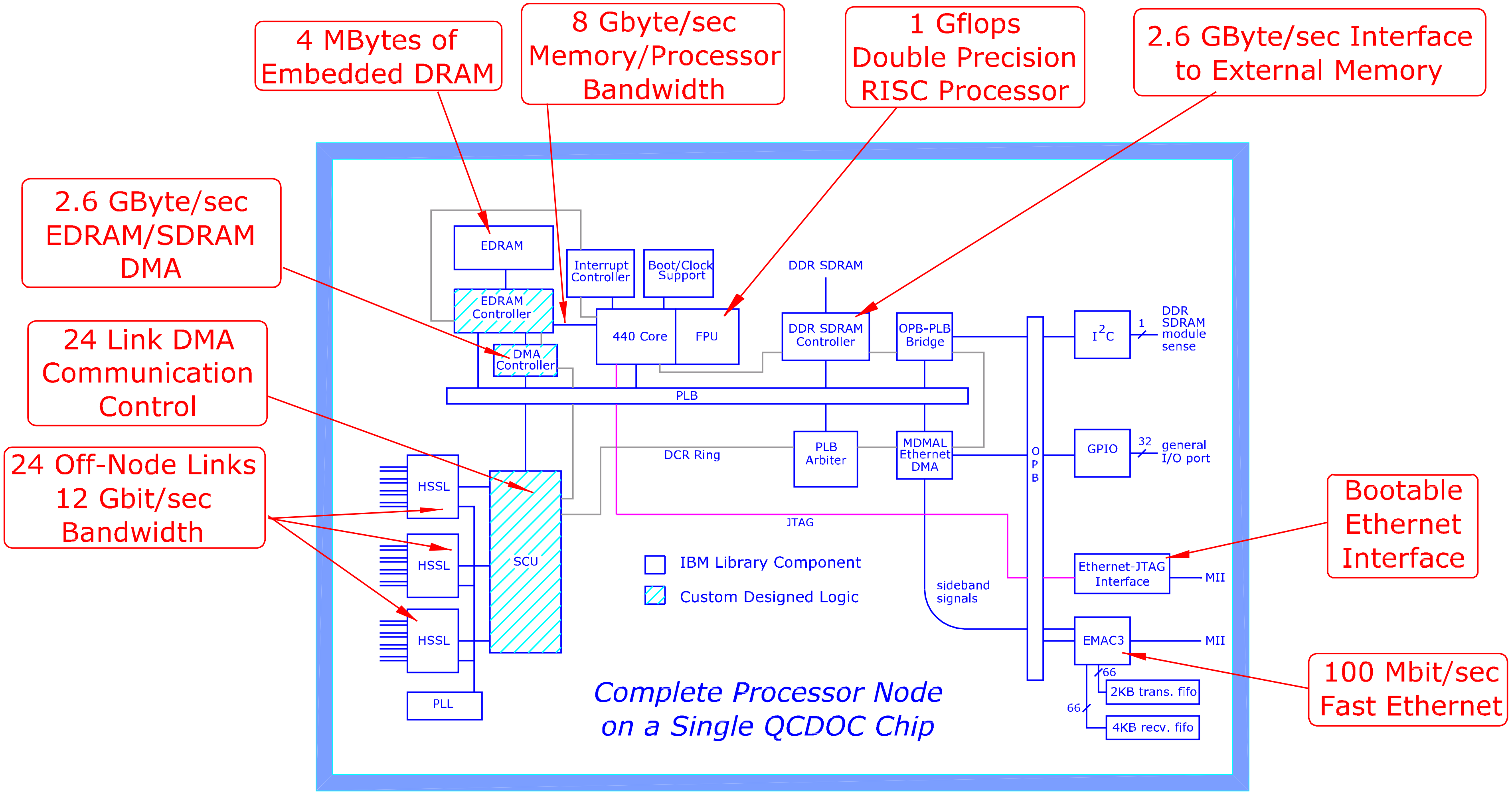
- QCDOC communications hardware is optimized for this case:
  - 50% sustained performance even for $V_{\mathrm{local}} = 2^4$
  - scalable to several 10 TFlops

Overview of the QCDOC architecture

- system-on-a-chip design (QCDOC = QCD on a chip)
- Technology: IBM SA27E = CMOS 7SF = $0.18\,\mu$ lithography process
- ASIC (Application-Specific Integrated Circuit) combines existing IBM components and QCD-specific, custom-designed logic:
  - 500 MHz PowerPC 440 processor core with 64-bit, 1 GFlops FPU
  - 4 MB on-chip memory (embedded DRAM), accessed through custom-designed prefetching eDRAM controller (PEC)
  - nearest-neighbor serial communications unit (SCU) with aggregate bandwidth of 12 Gbit/s
- separate networks for physics communications and booting/IO
- price/performance ratio = 1 US-$ per sustained MFlops

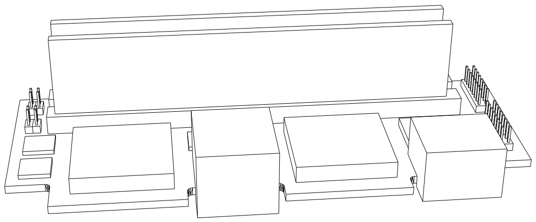
All this on a piece of silicon about 11 mm square, consuming $\sim$ 5 W

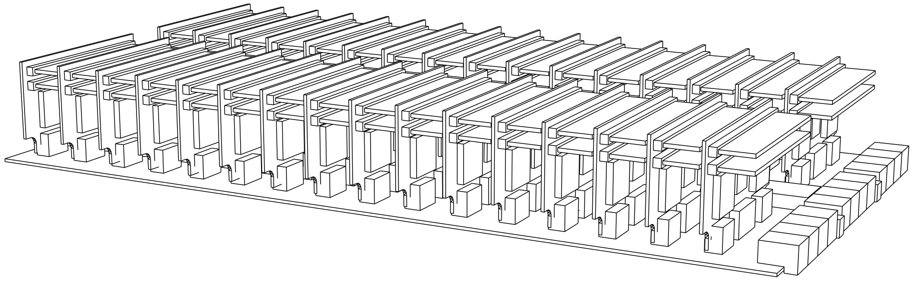
The QCDOC ASIC

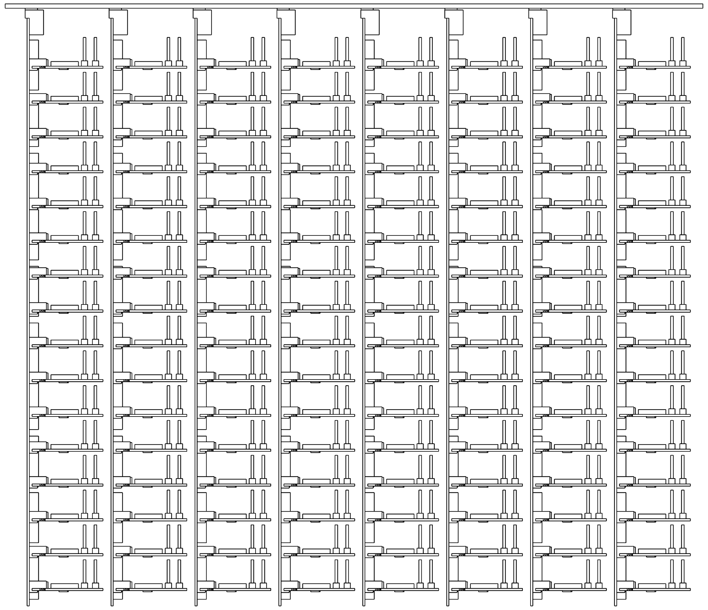


Mechanical design

- a daughterboard holds two QCDOC ASICs and two standard DDR SDRAM modules with ECC (one per ASIC, up to 2GB)
- 32 daughterboards mounted on a motherboard
- 8 motherboards mounted in a crate with a single backplane
- no air-conditioning necessary, only chilled water supply

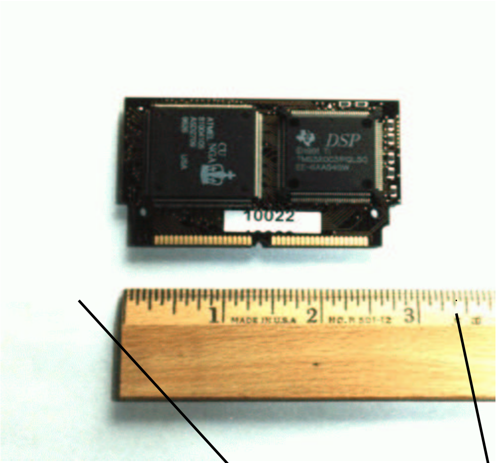






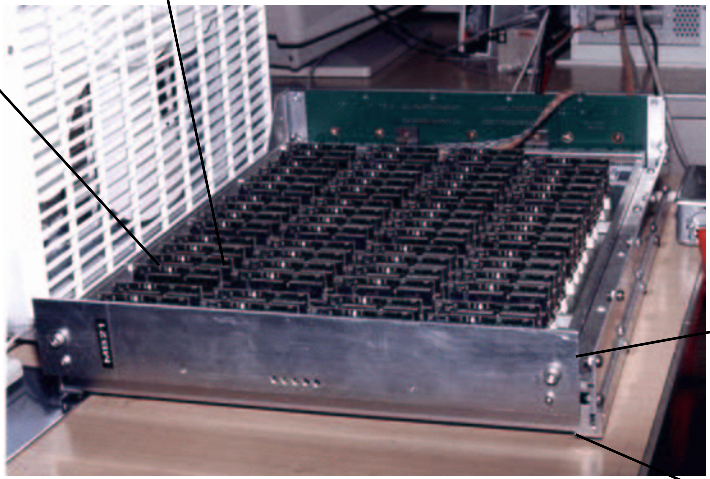
QCDSP

NGA CPU

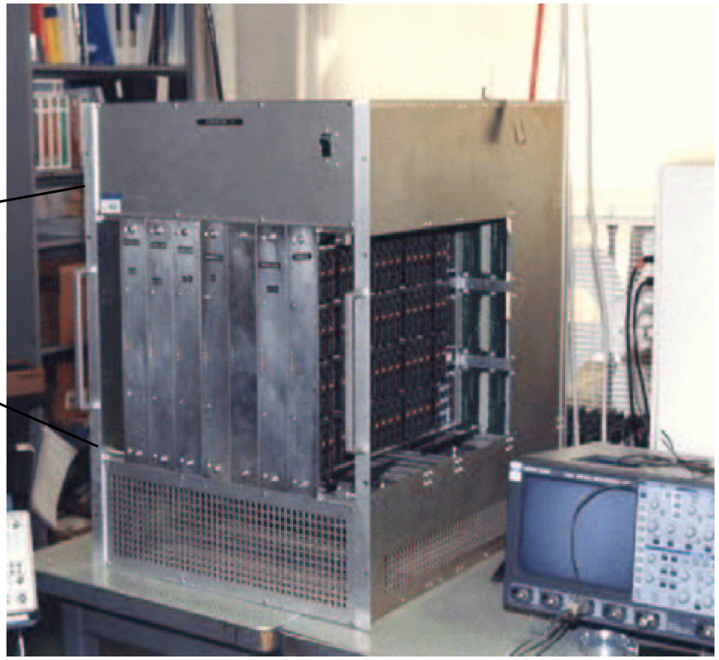


Daughterboard

Motherboard



Crate



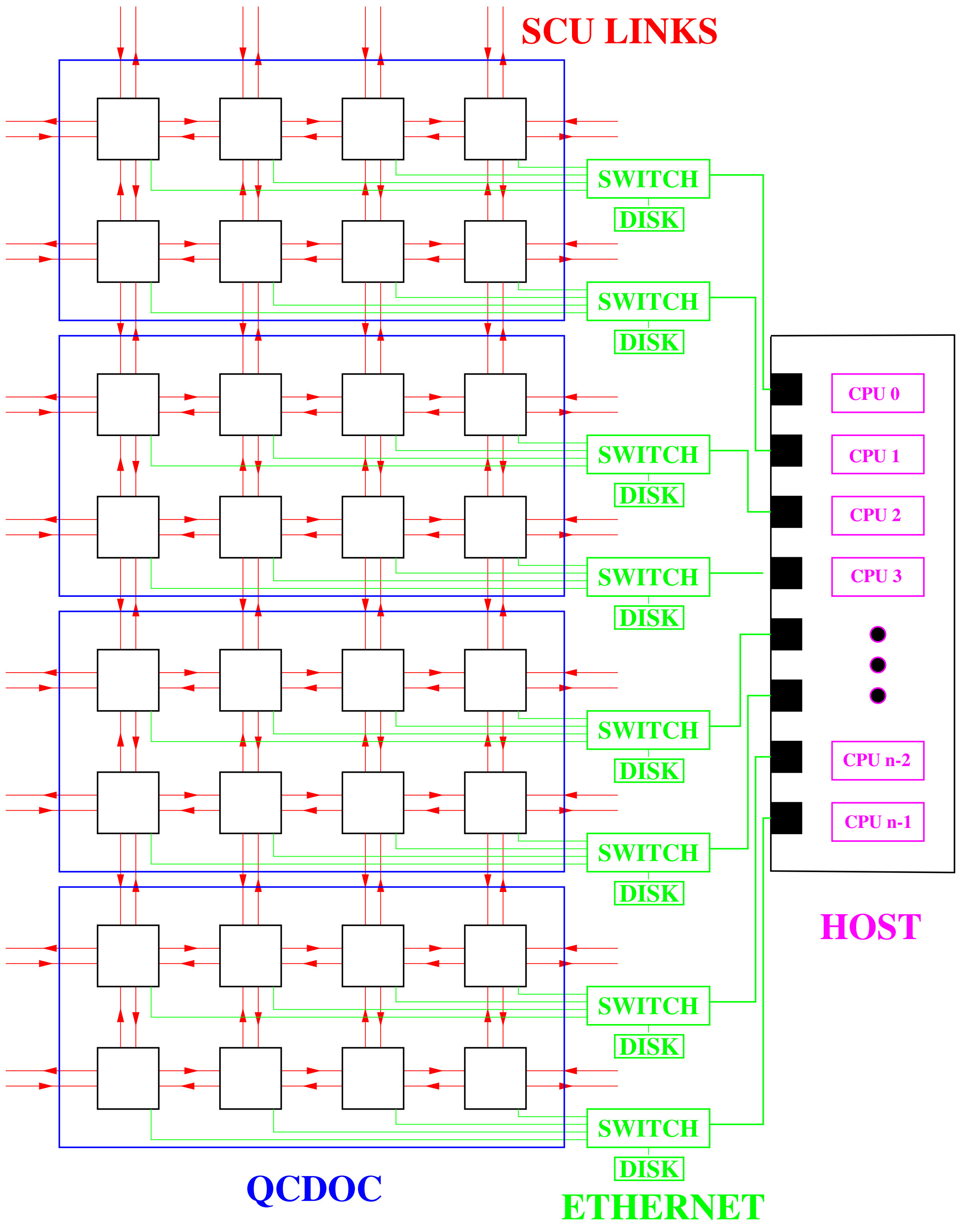

Gordon Bell
Prize
at SC ’98
The QCDSP
Supercomputer
The 400Gflops QCDSP Machine
Room 1106 Pupin Labs (11th Floor)
Columbia University, New York, NY

Physics network

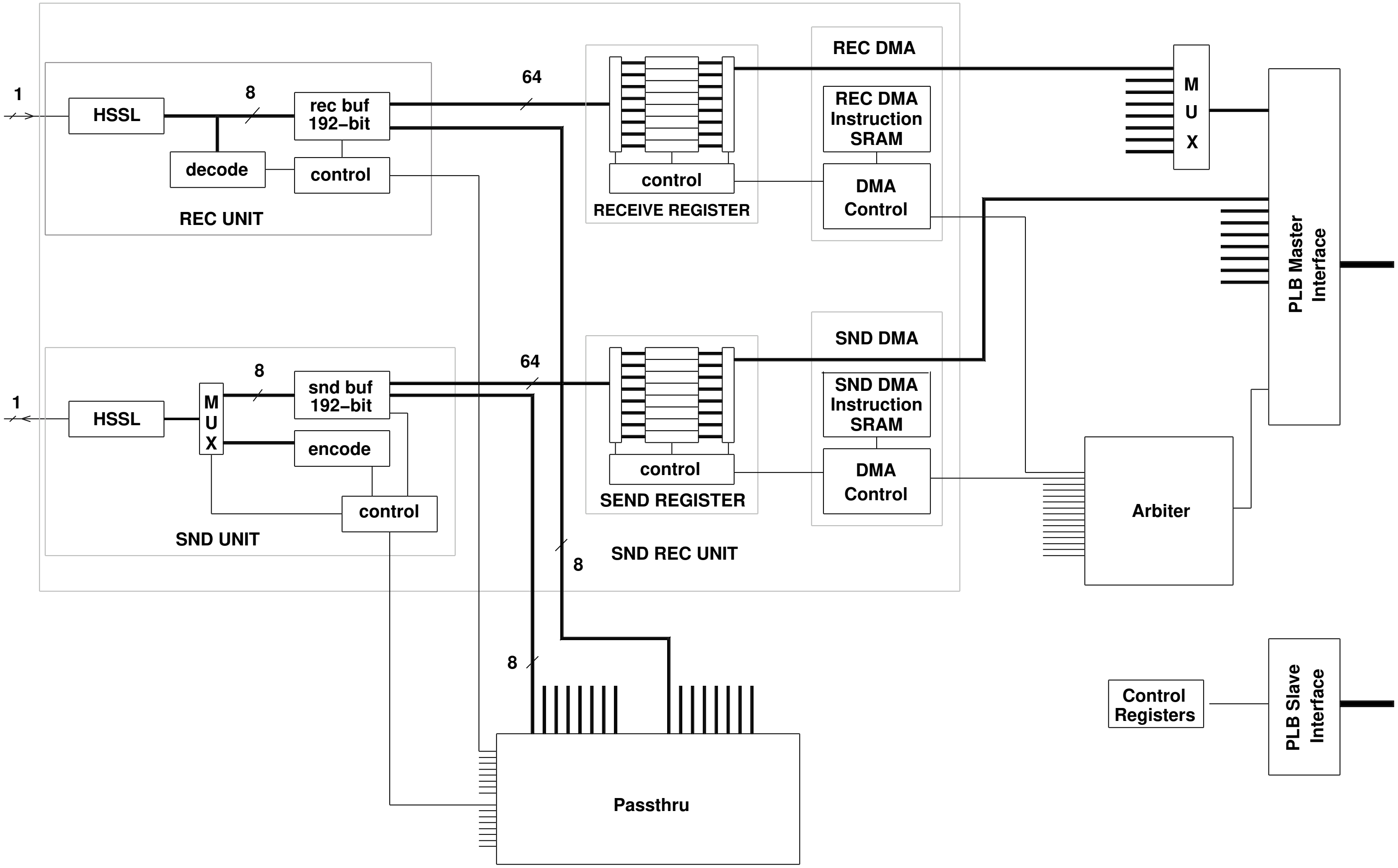
- 6-dimensional torus with nearest-neighbor connections (allows for machine partitioning in software)
  - 2^6 topology on motherboard
  - 3-d open, 3-d closed on motherboard (1-d closed on daugherboard)
- high-speed serial links with 2×500 Mbit/s bandwith per link
- custom-designed Serial Communications Unit (SCU) provides
  - direct memory access
  - single-bit error detection with automatic resend
  - low-latency passthrough mode for global operations
- global tree network for three independent interrupts

Auxiliary network

- Booting, diagnostics, and I/O over ethernet (controller on chip)
- 100 Mbit/s off daughterboards, hubs on DB/MB
- 800 Mbit/s off motherboards to commercial Gig-E switch tree, parallel disk system and host machine
- host is standard Unix SMP with multiple Gig-E cards

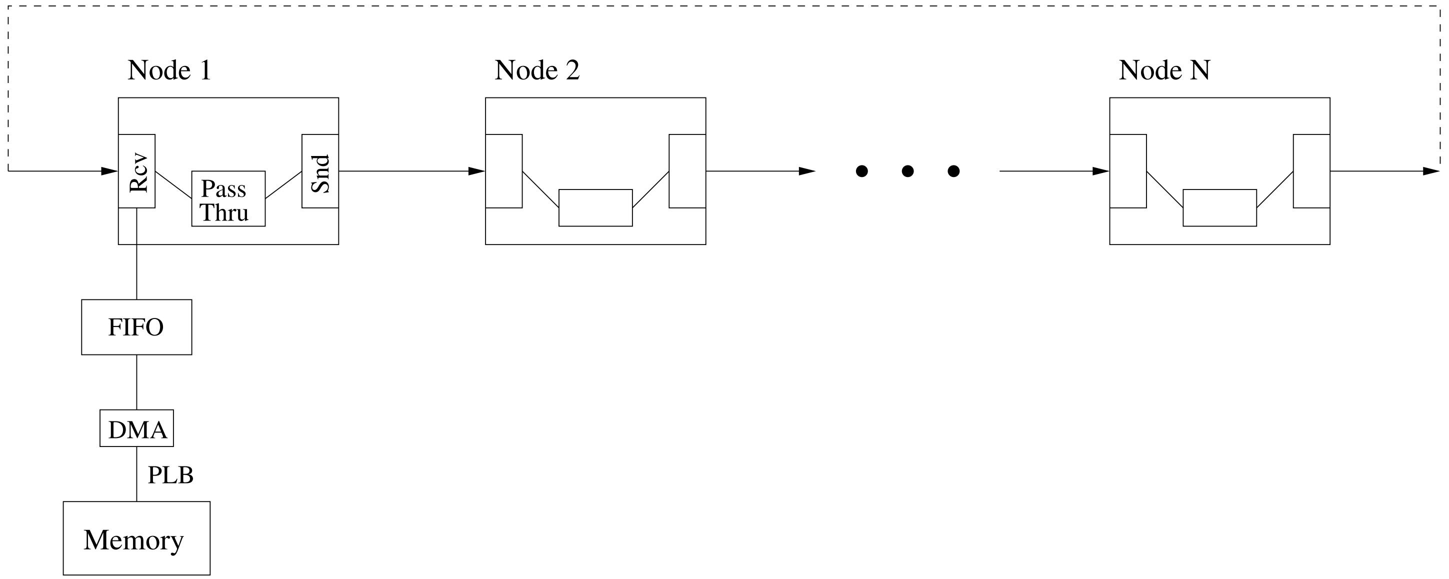

SCU LINKS
SWITCH
DISK
SWITCH
DISK
SWITCH
DISK
SWITCH
DISK
SWITCH
DISK
SWITCH
DISK
SWITCH
DISK
SWITCH
DISK
CPU 0
CPU 1
CPU 2
CPU 3
CPU n-2
CPU n-1
HOST
QCDOC
ETHERNET

Serial Communications Unit (SCU)



- nearest-neighbor communication through IBM high-speed serial links (HSSL) at 500 MHz, synchronized by IBM phase-locked loop (PLL)
- custom protocol: packets contain 64-bit data word and 8-bit header containing control and parity bits
- receive unit:
  - interprets headers, buffers bytes from HSSL and assembles 64-bit words, transfers words to receive register or passes them to send unit
  - receive buffer can store three 64-bit words so that send unit can send three words before acknowledgement is received
- send unit: inverse functionality of receive unit
- send and receive operations can proceed simultaneously
- DMA engines transfer data between memory and send/receive buffers (instructions stored in on-chip SRAM)
- hardware latency: send - 4cycles = 8 ns, receive - 36 cycles = 72 ns
- global sums: low-latency passthrough mode, hardware-enhanced shift-and-add by implementing store-and-forward in the SCU (software latency of $\sim$ 300 ns only paid once for store-and-forward per dimension)

Communications latency hiding for global sums (1-d example)



- receive unit gets number from neighbor in $-$ direction
  $\longrightarrow$ puts it in FIFO and also passes it on to neighbor in $+$ direction
- FIFO is flushed when PLB access is available
- after $N-1$ shifts, all numbers from this dimension are memory
- add done by CPU

$\longrightarrow$ software latency only paid once instead of N times
broadcast of final result not necessary

Status

- ASIC design finished, netlist handed over to IBM
- first hardware in May, large machines at the end of 2003
- funded machines: 10 TFlops each at Edinburgh and RIKEN-BNL
  5 TFlops at Columbia (approval pending)
- planned: $\geq$ 20 TFlops machine at BNL for US lattice community

Prospects for other applications

- two workshops at BNL in fall 2002 and spring 2003
- several potential "customers", e.g.
  - supernova core collapse simulations
  - molecular dynamics
- QCDOC can be used immediately for applications that
  - are based on local interactions
  - use linear algebra (LINPACK etc.)
- supervisor mode could be used to implement general routing schemes
- standard compilers available, OS not specific to QCD
- MPI is being ported to QCDOC

Summary of QCDOC advantages

1. highlights
   - price-performance ratio of $1 per sustained MFlops
   - scalable to many 10,000 nodes, still 50% efficiency
2. network:
   - 6-dimensional mesh
   - low-latency communications hardware
   - passthrough mode for store-and-forward
3. software:
   - standard PowerPC
   - standard compilers available (GNU gcc/g++/g77, IBM xlc/xlC/xlf)
   - high-performance software for lattice QCD (QCD API – SciDAC)
   - standard OS (linux) possible
   - MPI port by BNL
4. maintenance:
   - low power consumption
   - no air-conditioning necessary
   - easy debugging through JTAG