FSU-SCRI-99-15

QCDSP Machines: Design, Performance and Cost

Dong Chen
Dept. of Physics
Massachusetts Institute of Technology

chen@ctpa03.mit.edu

Ping Chen
Dept. of Physics
Columbia University

pchen@phys.columbia.edu

Norman H. Christ
Dept. of Physics
Columbia University

nhc@phys.columbia.edu

Robert G. Edwards
Supercomputer Computations Research Institute
Florida State University

edwards@scri.fsu.edu

George Fleming
Dept. of Physics
Columbia University

gfleming@phys.columbia.e du

Alan Gara
Nevis Labs
Columbia University Nevis Labs

gara@nevis.nevis.colu mbia.edu

Sten Hansen
Fermilab National Acceleration Lab

hansen@fnal.fnal.gov

CERN LIBRARIES, GENEVA
SCAN-0012008

Chulwoo Jung
Dept. of Physics
Columbia University

chulwoo@phys.columbia.edu

Adrian Kahler
Dept. of Physics
Columbia University

adrian@phys.columbia.edu

Stephen Kasow
Dept. of Physics
Columbia University

kasow@phys.columbia.edu

Anthony D. Kennedy
Supercomputer Computations Research Institute
Florida State University

adk@scri.fsu.edu

Greg Kilcup
Dept. of Physics
Ohio State University

kilcup@pacific.mp s.ohio-state.edu

Yubing Luo
Dept. of Physics
Columbia University

roy@phys.columbia.edu

Catalin Malureanu
Dept. of Physics
Columbia University

catalin@phys.columbia.edu

Robert D. Mawhinney

Dept. of Physics
Columbia University

rdm@phys.columbia.edu

John Parsons
Nevis Labs
Columbia University Nevis Labs

parsons@nevis.nevi s.columbia.edu

ChengZhong Sui
Dept. of Physics
Columbia University

sui@phys.columbia.edu

Pavlos Vranas
Dept. of Physics
Columbia University

vranas@phys.columbia.edu

Yuri Zhestkov
Dept. of Physics
Columbia University

zhestkov@phys.columbia.e du

Abstract:

The Quantum Chromodynamics on Digital Signal Processors (*QCDSP*) machines in operation at Columbia University and nearly complete at the RIKEN Brookhaven Research Center are MIMD machines with processing nodes based on the Texas Instruments TMS320C31-50 digital signal processor (DSP), interconnected as a four-dimensional torus. The Columbia machine contains 8,192 nodes and has a peak speed of 0.4T flops. The RIKEN/BNL machine has 12,288 nodes, a peak speed of 0.6 Tflops and a total cost of $1.85M. In order to establish a cost/performance figure for this architecture, we have run two standard lattice quantum chromodynamics (QCD) programs on portions of this hardware. The first program stochastically estimates the trace of the inverse of the Wilson Dirac operator, computed on a series of $16 \times 32 \times 64 \times 16$ lattice configurations. Running for 49 minutes on 1/6 of the Brookhaven machine, this code performs 6.8×10^{13} floating point operations for a

cost/performance of $13.2/Mflops. We also present the performance of a second production program which generates a Markov chain of $16^3 \times 64$ lattice

configurations distributed according to the statistical weight describing two species of light Wilson quarks interacting with the $SU(3)$ gauge degrees of freedom of QCD.

Running for 1334 minutes on a single cabinet at Columbia (equivalent to 1/12th of the Brookhaven machine), this program executes 9×10^{14} floating point operations

for a cost performance of $13.6/Mflops. Further information about these machines can be found at `http://www.phys.columbia.edu/~cqft`

Keywords:

parallel, supercomputer, digital signal processor, quantum chromodynamics

Introduction

Over that past five years a group in the Physics Department at Columbia University, together with collaborators from four other institutions, has designed and built a number of *QCDSP* machines[1][2][3][4] [5][6][7]. These are generally programmable, highly parallel computers targeted at large-scale numerical studies of the fundamental theory of the strong interactions, QCD. Each machine represents a different configuration of common, scalable, computer hardware. We presently have a 64-node machine installed at the University of Wuppertal in Germany, a 128-node machine at Ohio State University, and a 1024-node machine at Florida State University, as well as the 8,192- and 12,288-node machines at Columbia and RIKEN/Brookhaven respectively. A picture of the Columbia machine is shown in Figure 1.

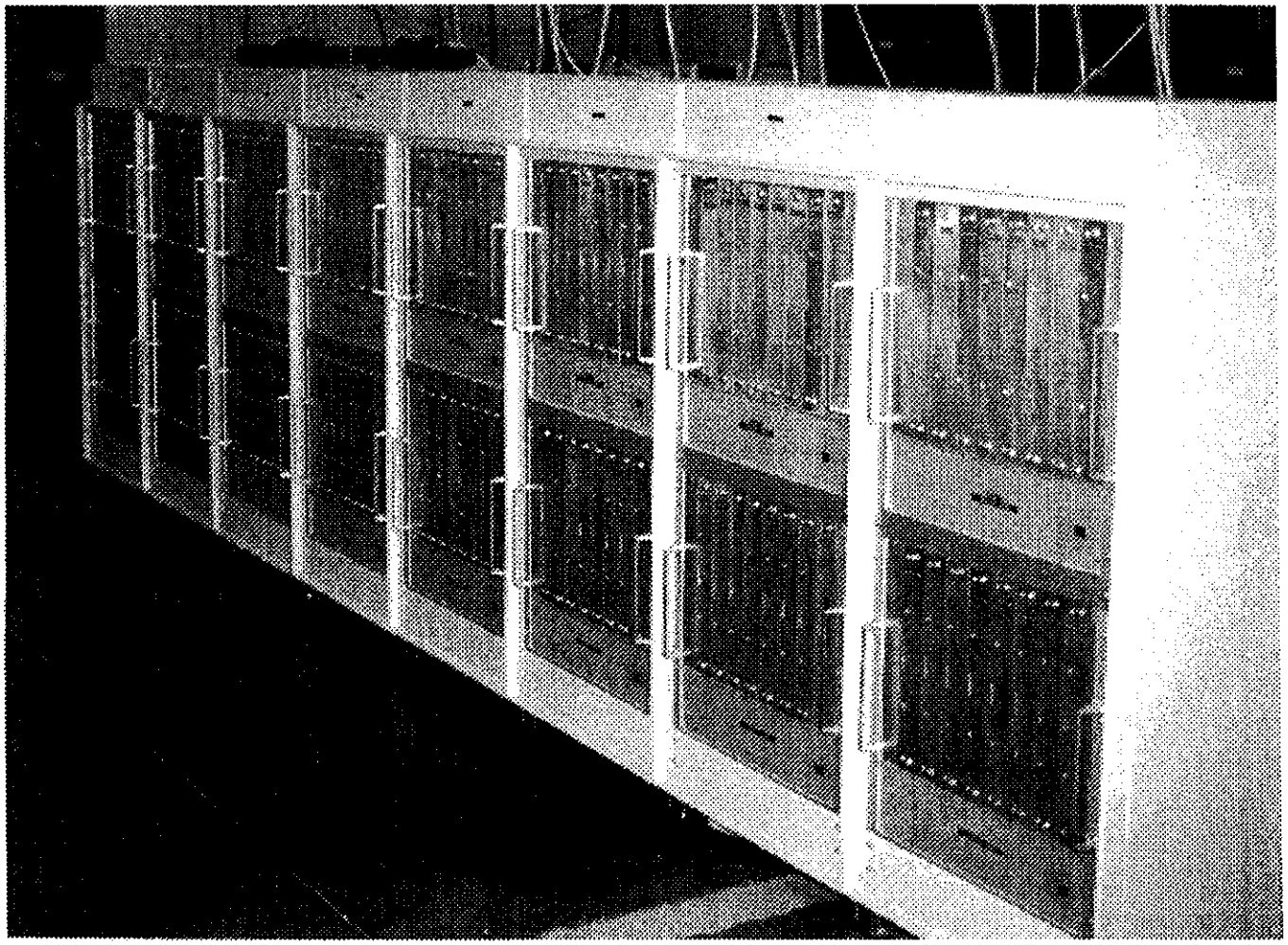

Figure 1: The 400 Gflops (peak) machine now operating at Columbia. The machine has 8,192 nodes and consumes about 50KW. Water cooling is provided to allow operation in a space without a raised floor.

The 0.4 Tflops (peak) Columbia machine was completed in April, 1998, but parts of this machine have been in intensive use since September, 1997. Since our initial physics program has been to reproduce earlier results obtained on smaller machines and to explore a new physics algorithm (domain wall fermions), we have cabled the machine at Columbia as eight separate 1024-node machines, each running an independent calculation. Therefore, in the benchmark presented from the Columbia machine, we have run on a single 1024-node cabinet. As our studies proceed over the next one-two months, we expect to join the Columbia machine into a smaller number of larger sections.

The 0.6 Tflops (peak) machine at Brookhaven is nearly complete but has had major components finished and operating since the middle of April, 1998. We therefore have run the RIKEN/Brookhaven machine benchmark on a 2048-node portion of the machine

that was complete.

The two benchmark programs used to establish the sustained speed of our machines were taken from our present suite of physics production code. They have been in use since March, 1998 and, for example, contributed the results presented by our group at the recent Workshop on Fermion Frontiers in Vector Lattice Gauge Theories, held at Brookhaven, May 6-9, 1998.

Hardware

Because of its regular, homogenous structure, lattice QCD lends itself easily to parallel computation. The problem is initially formulated on a regular, four-dimensional grid or lattice points representing a discretization of space time. The calculation can then be parallelized by dividing this large, space-time lattice into an array of smaller identical sub-volumes and assigning the field variables associated with each sub-volume to a separate processor. Since the fundamental interactions that must be modeled are local, or involve neighboring sites, only processors that correspond to contiguous sub-volumes need to communicate directly, for most parts of the calculation. Since the most interesting lattice QCD calculations are very computationally demanding, one is often limited to small lattice sizes even when using a powerful, many-processor, parallel machine. This implies a relatively large surface-to-volume ratio for the subvolume managed by a single processor and places increased demands on the inter-processor communications. As a rule of thumb, one needs one Mword/sec of off-node bandwidth per 10 Mflops of processor speed. There are similar demands for small communication latency, given the frequent, short communications required by this relatively small problem size.

Our architecture[6] is chosen to provide these characteristics at a reasonably low cost. The fundamental node of our machine is constructed from a commodity processor, a Texas Instruments DSP. This DSP executes 32-bit floating point arithmetic at a peak speed of 50 Mflops. The memory is standard 4 Mbit, 60ns DRAM (now a little dated). The only non-commodity component in the machine is the custom gate array which provides EDC and a 32-word programmable cache needed for the DSP to use DRAM effectively. This device also controls the 16 serial wires needed to provide bi-directional communications with the eight nearest neighbors in a four-dimensional mesh. We designed this ASIC using Viewlogic tools, relying heavily on VHDL. This 250K-transistor chip worked on our first try and is manufactured for us by the ATMEL corporation for under $20/unit. The entire processor node fits on a 1.8" × 2.7" PC board whose complete manufacture and test costs less than $80, see Figure 2.

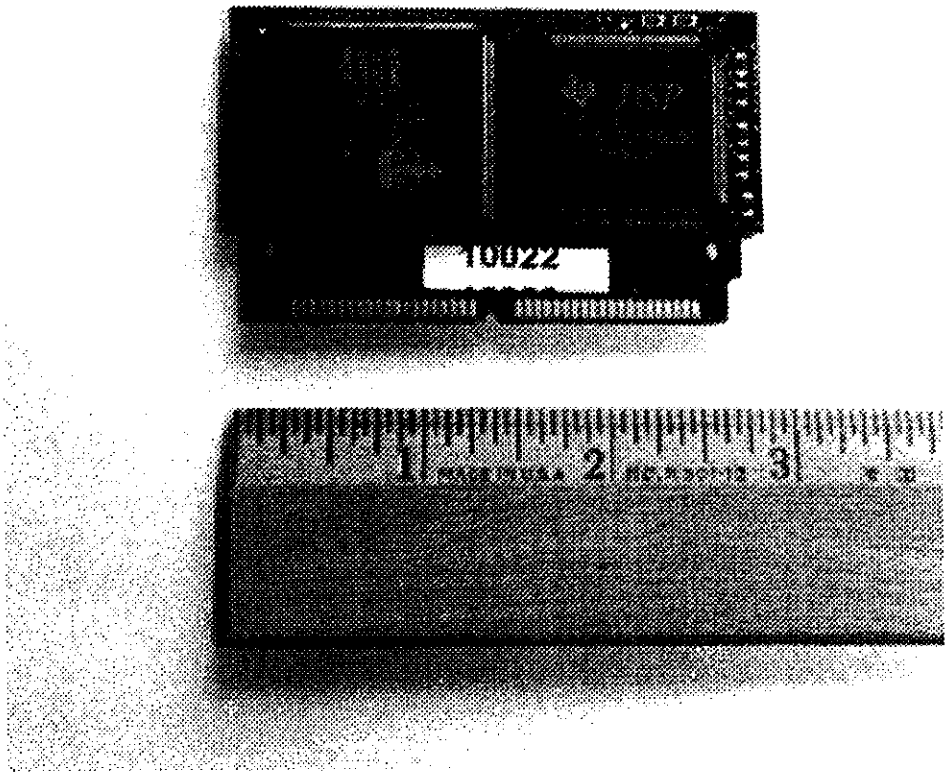

Figure 2: A picture of a single processor node. The DSP is on the right, the communications ASIC on the left and the five 4Mbit memory chips on the underside. This node now has a total manufacturing cost of $80, a peak computational speed of 50 Mflops and an off-node bandwidth of 40 Mbytes/sec.

Sixty three of these small cards mount in SIMM sockets on a 14.5" × 20" mother board which has a 64th node directly attached, with extra memory, PROM and SCSI access. One mother board is shown in Figure 3. The 64 nodes are interconnected on the mother board as a $4 \times 4 \times 2 \times 2$ array. Two of the eight faces of this hypercube are joined together on the mother board. The off-node serial wires corresponding to the remaining 6 faces are taken out to 6 separate cable connectors on the rear of the backplane, into which this board is inserted. These 6 connectors are then cabled to those of the other mother boards making up the machine to form the desired four-dimensional processor geometry.

A backplane holds eight mother boards and a large number of backplanes (at least 256) can be cabled together to form a large machine of various geometries. Each backplane has complete clock and reset circuitry so that no additional controller is required for the

machine. Each mother board has two independent SCSI ports. These are connected into a large tree with the UNIX host as its root. This SCSI tree is used to boot the machine, load code and extract results. Additional disks for checkpointing the calculation and data archiving can be joined to these SCSI connections as well. While all nodes in the machine can be joined together to form a single, large machine, it is also possible to cable the machine as a series of separate computers, each joined to a separate SCSI port on one or many UNIX host machines. In fact, we have even done production running on seven independent, two-mother board machines all attached to a single SCSI bus, connected to a single host.

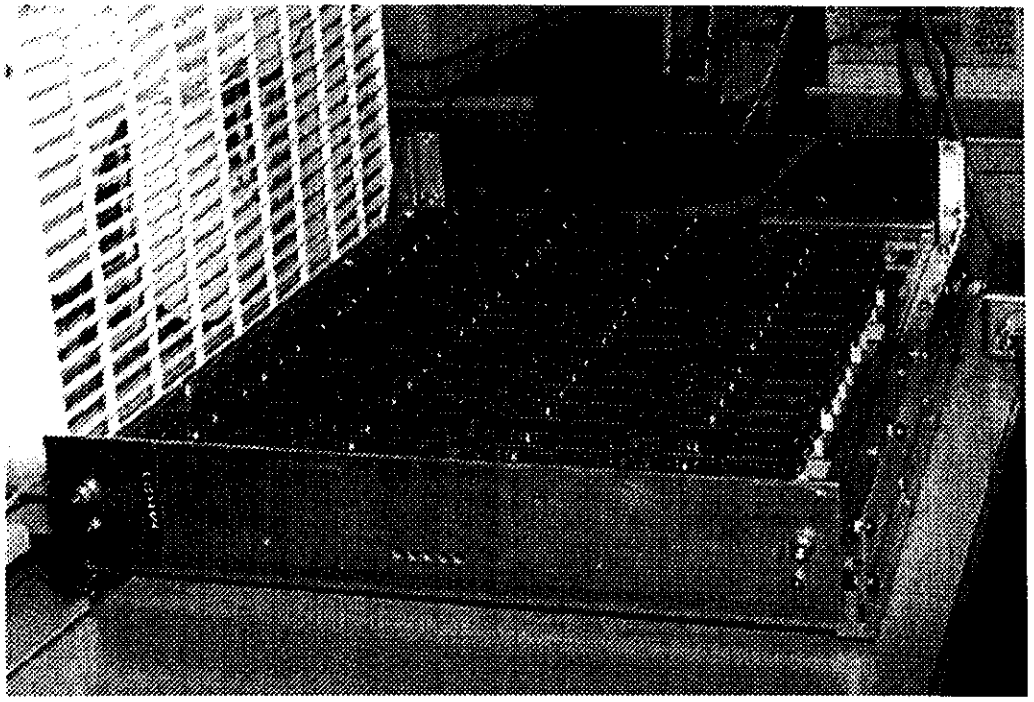

Figure 3: A picture of a single mother board mounted in our test fixture. This board holds 64 nodes, 63 socketed on daughter boards and the 64th soldered to the board with extra capability to access the PROM, to address the two SCSI controller chips, to drive a simple serial network linking it to the serial ports of the DSP's on other 63 nodes, and to use a larger 8 Mbytes of memory.

Let us complete this overview of the computer hardware by summarizing the characteristics of the inter-processor communication. Communications in each of the eight, off-node directions is controlled by a separate DMA engine within the ASIC. Each of these eight links can be programmed to send or receive a specified number of data blocks, of specified length, separated in memory by a fixed stride. The corresponding receive or send must also be programmed by the processor on the other end of the link. These two actions need not be synchronized: the hard-wired protocol does not permit data to be lost but rather stalls the DMA process until the transfer has been set up at both

ends of the link. The data transfer rate on a single link is ≈ 5Mbytes/sec, giving a net off-node bandwidth of 40Mbytes/sec. The DSP must only initiate such a transfer and later poll to determine that it has finished, thereby freeing the DSP to continue the calculation while the memory-to-memory transfer takes place in the background. The ASIC provides the arbitration needed so that both the DSP and this DMA communication can share access to the memory.

Since there is little overhead associated with initiating such a transfer, the communication latency is dominated by the time it takes for the actual data transfer and, given our ≈ 5Mbytes/sec bandwidth, little performance is lost because of the internode communications. One important exception to this statement comes from our need to perform global sums or global broadcasts over all nodes in the machine. Here the sum of the latencies associated with the required long sequence of serial transfers can potentially reduce the machine's performance by a factor of two. This problem is avoided by providing extra capability within the ASIC to perform what we term "pass-through" operations. By giving the ASIC on each node the capability to receive a serial data stream from one neighbor and then to rebroadcast that stream to as many as seven other neighbors with a four-cycle latency, we provide a mechanism for global broadcast that is nearly ten times faster than what would be achieved if that subsequent broadcast could only begin after the entire, incoming, 32-bit word had been received. Likewise, the ASIC contains circuitry to combine incoming serial data streams from up to seven neighboring nodes and to send to a single node the serial stream representing either the integer sum or the maximum of those incoming numbers. This again can be used to boost the performance of a integer (or floating point) global sum by more than a factor of 10.

Software

Code for the machine is written using commercial development tools for the DSP provided by Texas Instruments. These include C and C++ cross compilers as well as an assembler, all running on the host machine. The *QCDSP* machine is controlled from within a UNIX shell running on the host which, in addition to the normal c-shell commands, is augmented with further machine-specific instructions allowing the loading of code or data, the running of diagnostics, and the reading of data. Resident on each node is a small kernel which handles communication and provides the user with standard C programming support, such as the functions `printf()` and `fopen()`, so data can be written to the screen and files on the host can be directly accessed for reading or writing. In addition, there are specific "system" subroutines that can be called to initiate an inter-node data transfer.

With the present software environment, the actual programming of the machine is done from the perspective of a single node. Normally the lattice size on a single node will be

left as a run-time variable so a given piece of complied code can be run on a variety of machine topologies yielding results for a number of actual lattice volumes. While the most time critical inner loops may be written in assembler, the bulk of the application code for the machine is written in C^{++}. At present a large number of important physics programs have been completed, including code for computing hadron masses using staggered, Wilson fermions and domain wall fermions, as well as code for complete hybrid Monte Carlo sampling using each of these fermion formulations.

Our benchmarks for this submission use two of these production codes: the first computes the chiral condensate on a series of equilibrated, high-temperature lattices. The second carries out a hybrid Monte Carlo evolution including both the quark and gluon dynamics. The performance of both pieces of code relies on an efficient conjugate gradient inverter used to compute the inverse of the Wilson Dirac operator, a sparse $\approx 10^7 \times 10^7$ complex matrix. This inverter contains the usual algorithmic enhancements normally used to increase efficiency, including exploiting the spin projection structure of the **r=1** Wilson hopping terms and using a red-black preconditioning scheme.

Cost

The cost for such non-commercial hardware can be hard to determine. However, for this submission, we can use the actual cost of the machine being completed at the RIKEN Brookhaven Research Center at the Brookhaven National Laboratory. This machine is being constructed by the group at Columbia for an amount somewhat less than $1.85M. This figure has two components. The first is $1.8M in funding explicitly provided to Columbia for the procurement of the machine. Extensive paperwork is available at Columbia, detailing the cost of each component down to the last wire-lug and clock chip and each manufacturing/assembly contract. This sum has paid for complete, tested subassemblies: 206 fully populated mother boards, 16 spares, 12 fully assembled, water-cooled cabinets and two 8-slot, air-cooled crates.

The final $50K is an estimate of the labor costs required to configure and burn-in the entire system. This will take approximately four months and approximately one full-time Brookhaven technician and the 1/3-time supervision of one of us. We believe the $50K figure used is slightly larger than these actual personnel costs. We then compute the cost of the 2048-node machine on which the benchmark was run as a prorated $1/6^{th}$ of $1.85M or $308K.

Benchmark

We ran two separate benchmarks to establish the sustained speed of the machine. The

first is a series of measurements of the quark condensate, run on a 2048-node machine at Brookhaven configured as a $4 \times 8 \times 16 \times 4$ processor mesh machine with each node holding a 4^4 sub-volume. The code was set to generate a series configurations equilibrated at a coupling strength of $\beta = 6/g^2 = 6.0$ on which we computed an estimate of the trace of the inverse of the Wilson Dirac operator by averaging 50 random, diagonal elements. Estimates for four separate quark masses were performed on each configuration. Timed with the SUN processor clock, the code ran for 49 minutes and performed 46,514 conjugate gradient inversions. Since each such inversion requires 2,808 operations per site and the entire machine contains 524,288 sites, 6.8×10^{13} floating point operations were performed in that time, for a sustained rate of 23.3 Gflops and a cost performance of $13.2/Mflops.

The second benchmark was run on a 1024-node machine at Columbia. This was an actual complete Hybrid Monte Carlo run on a $16^3 \times 64$ lattice with a coupling strength $\beta = 5.3$ and a quark mass determined by the hopping parameter $\kappa = 0.1675$. In 1334 minutes this code performed 1,232,496 conjugate gradient inversions or 9×10^{14} floating point operations. This corresponds to a sustained rate of 11.3 Gflops for a hardware of one-half the $308K cost above, giving a cost performance of $13.6/Mflops.

Conclusion

In this paper we have described the design, performance and cost of a series of *QCDSP* machines that have come into operation over the last year. This represents a cost/effective and highly scalable architecture supporting computing platforms ranging from 64-node, 3.2 Gflops, single-mother board machines to the largest 0.6 Tflops machine under construction at the RIKEN Brookhaven Research Center. Because of its modular design the machine is constructed of only three assemblies: daughter boards, mother boards and backplanes. Thus, the manufacture of even a single large machines allows important efficiencies of scale. We have now begun to exploit these dedicated, Teraflops-scale resources to advance our knowledge of the properties and interactions of the quarks and gluons, the fundamental constitutients of the atomic nucleus.

References

1 Norman H. Christ, *A 0.5 Teraflops Machine Optimized for Lattice QCD*, Nucl. Phys. B (Proc. Suppl.) 34 (1994) 820.

2 Igor V. Arsenin, *Architectural Choices for the Columbia 0.8 Teraflops Machine*, Nucl. Phys. B (Proc. Suppl.) 42 (1995) 902, http://xxx.lanl.gov/abs/hep-lat/9412093.

3 Robert D. Mahwinney, *The Status of US Teraflops-scale Projects*, Nucl. Phys. B (Proc. Suppl.) 42 (1995) 140, http://xxx.lanl.gov/abs/hep-lat/9412068.

4 Igor V. Arsenin, *et al.*, *Status of the 0.8 Teraflops Supercomputer at Columbia*, Nucl. Phys. B (Proc. Suppl.) 47 (1996) 804, http://xxx.lanl.gov/abs/hep-lat/9509075.

5 Robert D. Mahwinney, *QCDSP: The First 64 Nodes*, Nucl. Phys. B (Proc. Suppl.) 53 (1997) 1010, http://xxx.lanl.gov/abs/hep-lat/9705028.

6 For a recent, thorough discussion of these QCDSP machines see: Dong Chen, *et al.*, *QCDSP: A Teraflop Scale Massively Parallel Supercomputer*, Proceedings, Supercomputing '97, http://www.supercomp.org/sc97/proceedings/TECH/CHRIST/INDEX.HTM.

7 Dong Chen, *et al.*, *QCDSP---A Status Report*, Nucl. Phys. B (Proc. Suppl.) 63A-C (1998) 997-999, http://xxx.lanl.gov/abs/hep-lat/hep-lat/9709135.