

LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

RECEIVED

11/18/88

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

LA-UR--88-1970

DE88 014439

TITLE: THE BIRTH OF THE SECOND GENERATION: THE HITACHI S-820/80

AUTHOR(S) Christopher Eoying
Raul H. Mendez
Olaf M. Lubeck

SUBMITTED TO Supercomputing '88 Conference
Orlando, Florida, November 14-18, 1988

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UN

ZDX, f

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

MASTER

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545



The Birth of the Second Generation: The Hitachi S-820/80

Christopher Eoyang* Raul H. Mendez* Olaf M. Lubeck**

Abstract

The performance of the new Hitachi S-820/80 supercomputer was evaluated on a set of standard Fortran benchmark codes that range from simple kernels to fluid dynamics applications and compared with the performance of the NEC SX-2 and CRAY X-MP/48 supercomputers.

Keywords: Supercomputer Vector processor LFK loops LANL benchmark codes

1. Introduction

The entry of the Hitachi S-810/20 and Fujitsu VP-200 supercomputers in late 1983, followed by the NEC SX-2 in 1984, gave much credibility to the advanced technology of the Japanese manufacturers and had a profound impact on the area of supercomputing, not only in Japan, but throughout the world. All single-processor machines were capable of a peak speed in excess of 700 MFLOPS and boasted very advanced compilers and vectorizing tools. Although the performance of the Fujitsu VP series is, in general, slower than that of the SX-2 and faster than that of the S/810, the VP has taken the dominant share of the Japanese market (well over half), with about 50 systems currently installed.

A little more than four years later, all three manufacturers have introduced new machines, which could be said to constitute the "second generation" of Japanese supercomputers, even though the degree to which the new machines differ from their predecessors varies to a great extent. NEC and Fujitsu, having respectively introduced the SX-A and VP-E series supercomputers, have not departed significantly from the architecture and technology of their original machines; the new machines have additional

* Institute for Supercomputing Research, 2-11 Kachidoki, Chuo-ku, Tokyo, 104 JAPAN

** Los Alamos National Laboratory, Los Alamos, NM 87545

pipelines, higher capacities, and/or improved features, but the basic hardware remains essentially unchanged.

On the other hand, Hitachi has introduced a true second-generation machine, the S-820/80 (the other model, the S-820/60, has half the performance), which maintains the same architecture of its predecessor, while making tremendous technological improvements on the device level. This has resulted in a scalar clock speed of 8 ns (compare 28 ns on the S-810) and a vector clock of 4 ns (14 ns on the S-810), the fastest vector clock of any machine made today, closely edging that of the CRAY-2 (4.1 ns). The theoretical peak performance of the S-820/80 is 2 GFLOPS, making the S-820 the fastest single-processor vector machine in the world, for which scalar-vector speedups of 30 or more are not unusual.

In this article we shall evaluate the performance of the S-820 by using various benchmark codes not primarily to quote and compare the CPU times and MFLOPS rates, but to determine the strengths and weaknesses of this machine in comparison to other machines. We firmly believe that one cannot reduce supercomputer performance down to a single number on a single benchmark or set of benchmarks any more than one can generalize the characteristics of supercomputer application codes. Conclusions regarding the performance of a supercomputer are perhaps best made after running a wide range of codes and then trying to correlate the observed performance with the various attributes and peculiarities of the machine.

Vector machines are best suited for completely vectorized applications, and it is well known that because of Amdahl's Law, performance will drop off drastically from the theoretical peak if the vector ratio of an application is not well into the 90th percentile. It has also been established that the average vectorization ratio of the applications being run on current supercomputers is much closer to the 70th, rather than the 100th, percentile [1], and in many cases scalar speed is the determining factor in a machine's overall performance.

In Section 2 we will describe the basic features of the S-820, paying attention to the technology used on the device level. In Section 3 we will discuss the performance of the S-820 on the Livermore Fortran Kernel (LFK) loops, and Section 4 will cover the Los Alamos National Laboratory (LANL) benchmark set. Section 5 deals with the Mendez fluid dynamics codes.

The S-820/80 was benchmarked at Hitachi's Kanagawa Works in January and February 1988, using the FORT77/HAP (V21-0b) compiler.

2. Architecture of the S-820

The computational processor of the S-820 has the same basic architecture as its predecessor, the S-810, consisting of separate vector and scalar units, which can be run in parallel. The 8-ns scalar processor, based on the Hitachi M-series mainframes, is augmented with a very powerful vector processor with a cycle time of 4 ns. To increase the efficiency with which the scalar and vector units can be run in parallel, Hitachi has added "link" and "signal" functions to coordinate simultaneous operation between the units [2]. The basic architecture of the S-820 is shown in Appendix A.

The vector processor has 32 vector registers (each capable of containing 512 64-bit words) and 16 vector mask registers, supported by four vector load and four vector load/store pipes capable of concurrent operation (up to 8 load operations, or 4 loads and 4 store operations simultaneously). Computation takes place in the four arithmetic units (add/logical, multiply-add, divide, mask). Each load and load/store pipe can transfer 8 bytes (1 word) to/from memory every 4 ns, for a total bandwidth of 16 Gbyte/s (2 Gword/s). The add/logical and multiply-add units each consist of 4 fully segmented pipelines, with an execution speed of 1 GFLOPS for the add/logical unit and 2 GFLOPS for the multiply-add unit. The theoretical maximum computational performance of the S-820 is 3 GFLOPS when both units are running concurrently. This is limited, in many cases, by the 2-Gword main memory to vector register bandwidth. The divide and mask units both consist of one pipeline each.

The scalar processor is also much improved over the S-810, with add operations requiring 16 or 24 ns (2 or 3 clock periods), and multiplication taking 24 or 32 ns (the slower times for each operation occur only when one of the operands has been used in the previous instruction). Division times have also been improved from 588 ns to 168 ns.

Maximum main memory on the S-820 is 512 Mbytes, using bipolar CMOS (bi-CMOS) static RAMs with access times of 20 ns. Up to 12 Gbytes of extended storage (using 1-Mbyte 120-ns DRAMs) can be added. The bandwidth between main memory and extended memory is 2 Gbyte/s. The input/output processor of the S-820 is the same as on the M-series

mainframes and has a maximum capacity of 64 channels, with a total bandwidth of 288 Mbyte/s.

The FORT77/HAP (version V21-0b) compiler on the S-820 has been improved considerably and appears to be comparable to those of the other Japanese manufacturers. For example, the compiler is capable of vectorizing loops containing IF-statements, intrinsic functions, loops with out-of-loop GOTO statements; it is also capable of handling a variety of special case combination functions (inner product, first order linear recurrences, summation, first maximum/minimum, gather/scatter operations). For nested loops, the compiler can perform loop splitting, loop unrolling, and loop interchanging to maximize the vectorization ratio of a code.

3. The LFK Loops

The LFK loops are a widely used performance benchmark that serve as a general indication of a machine's maximum performance. The data in Figures 1 and 2 indicate the scalar and vector performance of the S-820 compared with the NEC SX-2 and CRAY X-MP (single processor) on the revised set of 24 LFK loops.

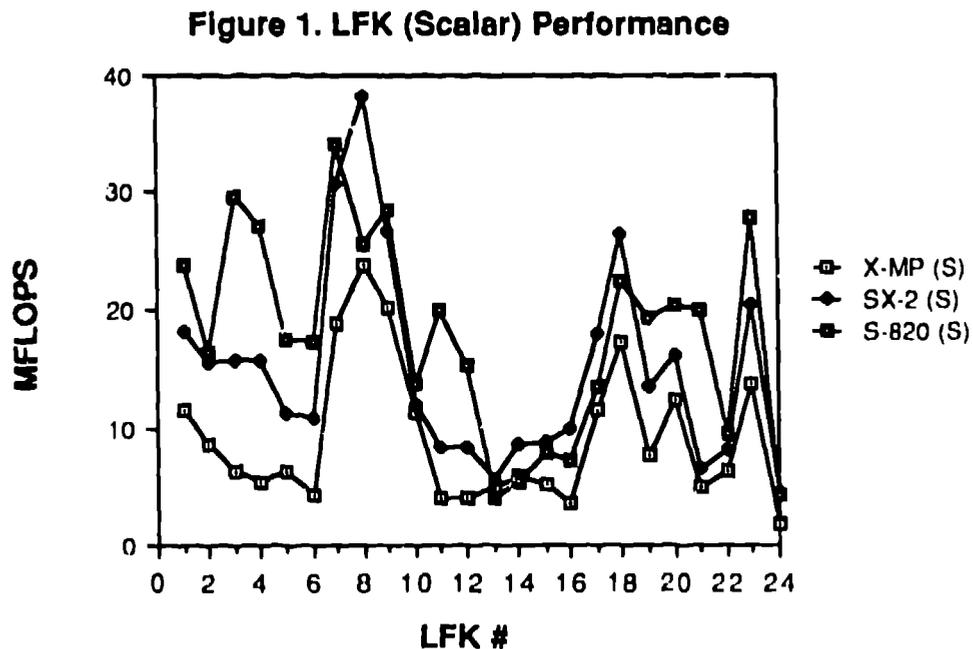
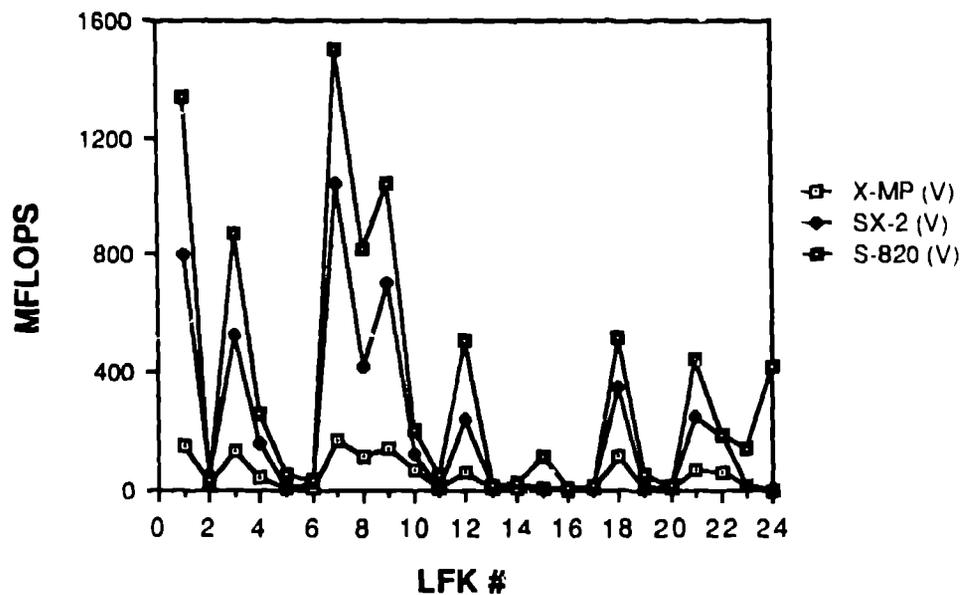


Figure 2. LFK (Vector) Performance



Of particular interest is the fact that kernels 5, 11, and 19 (all of which include first order linear recurrences that are not vectorized on any other machine) are vectorized by the Hitachi compiler, and all of these kernels showed speedups of about 2.7 in vector mode, whereas scalar mode was generally faster on the other machines. As indicated in the scalar performance data, the S-820 scalar mode is faster than the other machines in most, but not all, of the kernels (the NEC is faster in 6 of the 24 loops). Furthermore, the speedup over the NEC is only marginal. Since the scalar clock of the S-820 is 8 ns (double the vector clock of 4 ns), slower than the 6-ns NEC clock, these results suggest a very efficient scalar code generation on the S-820, which was also a feature of its predecessor on the S-810 [3, p. 22].

In vector mode, the performance of the S-820 is outstanding (Figure 2). On highly vectorizable code, it exhibits nearly twice the performance of the SX-2 and has about seven times the vector speed of a CRAY X-MP/1. In the extreme, the S-820 is more than 200 times faster than the Cray (kernel #24, first minimum) [4], because of a special vector "find minimum" instruction, which allows the code to be vectorized. This also accounts for the extraordinary factor of 100 speedup on kernel #24 over scalar mode on the S-820 (421.6 seconds in scalar mode, 4.2 seconds in vector mode).

4. The LANL Benchmark Set

The LANL benchmarks are a set of codes spanning a hierarchy of performance measurements including simple vector loops, basic routines representing building blocks of production codes, and stripped-down applications. Appendix B contains a short description of each code. The benchmark set has been executed on most major supercomputers and mini-supercomputers [1]. In this section, we will compare the results of the S-820 with another Japanese supercomputer, the NEC SX-2, and the CRAY X-MP/48 (single-processor results).

Table 1. NEC SX-2. Simple vector operation rates (MFLOPS) as a function of vector lengths.

Vector Length	10	50	100	200	1000
$A(I) = B(I) + S$	22	110	219	340	382
$A(I) = B(I) + S, I=1, N, 23$	22	108	136	449	153
$A(I) = B(I) + S, I=1, N, 8$	21	87	125	146	150
$A(I) = B(I) * C(I)$	20	97	181	265	275
$A(I) = B(I)*C(I) + D(I)*E(I)$	38	191	365	521	528
$A(I) = B(J(I)) + S$	11	33	44	50	52
$A(J(I)) = B(I) * C(I)$	13	38	47	53	54

Table 2. CRAY X-MP/416. Simple vector operation rates (MFLOPS) as a function of vector lengths.

Vector Length	10	50	100	200	1000
$A(I) = B(I) + S$	14	50	58	61	67
$A(I) = B(I) + S, I=1, N, 23$	10	35	47	52	64
$A(I) = B(I) + S, I=1, N, 8$	10	36	45	53	66
$A(I) = B(I) * C(I)$	14	48	50	59	61
$A(I) = B(I)*C(I) + D(I)*E(I)$	33	87	92	97	100
$A(I) = B(J(I)) + S$	13	32	37	38	42
$A(J(I)) = B(I) * C(I)$	12	29	31	32	36

Table 3. Hitachi S-820. Simple vector operation rates (MFLOPS) as a function of vector lengths.

Vector Length	10	50	100	200	1000
$A(I) = B(I) + S$	26	122	237	382	736
$A(I) = B(I) + S, I=1, N, 23$	26	113	185	270	419
$A(I) = B(I) + S, I=1, N, 8$	25	113	191	237	247
$A(I) = B(I) * C(I)$	27	116	186	276	418
$A(I) = B(I)*C(I) + D(I)*E(I)$	50	212	393	617	982
$A(I) = B(J(I)) + S$	16	64	111	185	311
$A(J(I)) = B(I)*C(I)$	15	57	83	105	137

Tables 1, 2, and 3 show the performance data from simple vector loops as a function of vector length on three supercomputers. Short vector performance on the S-820 has improved by a factor of 2.5 to 3 over the S-810 [1]. When we compare with X-MP and S-820 data, we see that the new Hitachi machine is significantly better than the Cray at short vectors. Comparison with the SX-2, which had been the best short vector machine, shows that the new Hitachi is 30-40% faster. At long vector lengths, the S-820's 4-ns cycle time and 4 sets of functional units are evident in the impressive rates achieved, clearly outperforming the single-processor X-MP and SX-2. In vector mode across all vector lengths, the S-820 is consistently faster than any other supercomputer that we have measured.

On strided vector operations, the Hitachi asymptotic rate is half of the contiguous vector performance and can degrade further with memory conflicts (stride 8, for example). However, the performance of the S-820 with strided vectors is still significantly better than the Cray or NEC machines.

Table 4. LANL Benchmark Results (CPU times).

	S-820/80	X-MP/48	SX-2
GAMTEB	4.2	5.2	3.8
SCALGAM	94.1	72.5	67.7
BMK21	1.8	2	1.6
PHOTON	116.1	120.3	
SIMPLE	4.4	5.8	2.4
FFT	2	3.9	3.7
LSS	5.5	6.1	3.7
MATRIX	25.6	34.9	24.7
INTMC	6.1	12.1	10.8

The first four codes in Table 4 show the scalar performance of the Hitachi S-820/80. These codes are scalar Monte Carlo simulations of neutral particle transport through a material. In two of the codes (PHOTON and BMK21) the S-820 is comparable to the single-processor X-MP and the SX-2. In the second code (SCALGAM), the Hitachi is 20-30% slower than both, and in the third code (GAMTEB), the S-820 is 20% faster than the X-MP. Overall, the S-820 has comparable scalar speed to both the X-MP and SX-2.

Of the remaining codes in Table 4, the S-820 compares favorably with both of the other supercomputers. Of note is its performance on fast Fourier transform (FFT) codes and the integer Monte Carlo (INTMC) code, where it is roughly a factor of 2 faster than either the SX-2 or the X-MP. On the hydrodynamics code SIMPLE, it is equivalent to the X-MP but is significantly slower than the SX-2.

5. The Mendez Codes

The Mendez suite of fluid dynamics codes have been used in earlier studies to characterize the performance of vector and parallel machines on a class of applications [1,5-6]. Although the characteristics of these codes are very different and cover a range of fluid dynamics applications, performance on these codes is by no means meant to be strictly representative of the aptitude of any given machine to handle fluid dynamics codes in general. Of these five codes, three are highly vectorizable (VORTEX, MHID2D, and BARO are all over 95% vectorizable),

and two have vectorization ratios of 73% and 89%. The codes are briefly described in Appendix B.

The results, shown below in Tables 5 and 6, are in line with what one would expect given the performance data on the LFK loops. In scalar mode, the S-820 is just a little slower than the SX-2 in four of the codes and faster in one (BARO). With the exception of MHD-2D, the machines divide into two groups, with the S-820 and SX-2 on the faster side, and the X-MP and VP-200 running about equal.

Table 5. Mendez codes: relative scalar performance.

	S-820	SX-2	X-MP/1
VORTEX	1.65	1.80	1.00
EULER	2.59	2.59	1.00
MHD2D	0.88	1.00	1.00
BARO	2.40	1.79	1.00
SHEAR3	1.96	2.62	1.00

Table 6. Mendez codes: relative vector performance.

	S-820	SX-2	X-MP/1
VORTEX	3.77	1.93	1.00
EULER	1.15	1.53	1.00
MHD2D	5.52	2.31	1.00
BARO	5.16	3.63	1.00
SHEAR3	1.70	1.31	1.00

In vector mode, the type of application is the crucial factor in determining the performance of the S-820. In particular, the three highly vectorized codes are all 3-5 times faster than the X-MP and quite a bit faster than the SX-2. In EULER, a scalar-dominated code where the memory accesses are powers of two, the Hitachi machine finishes second behind the SX-2. In SHEAR3, the S-820 is faster by a nose, but the differences between machines are minimal.

6. Conclusion

The Hitachi S-820 is a great deal faster in vector mode than any other supercomputer we have measured, with almost twice the performance on highly vectorized codes than the fastest machine we have seen up to now, the NEC SX-2. In scalar mode, however, the S-820 is roughly even with the X-MP and the SX-2, with a slight advantage going to the SX-2 in the applications we have tested.

We must emphasize that the applications we have tested are CPU intensive and not I/O bound and that results obtained on other benchmark sets may lead to different conclusions. In any case, the S-820 is substantially faster than its predecessor, the S-810. Although improvements have been made in both the hardware and software of the machine, our results indicate that the compiler technology is roughly on par with those of the other supercomputer manufacturers and that most of the major speedups have been realized through improvements in hardware and device technology.

Acknowledgments

We thank the engineers and staff at Hitachi for their cooperation and generous assistance.

References

[1] Olaf M. Lubeck, "Supercomputer Performance: The Theory, Practice and Results," Los Alamos National Laboratory report LA-11204-MS, January 1988.

[2] Shun Kawabe *et al.*, "The Single Processor S-820: Peak Speed 2 GFLOPS," Nikkei Electronics, No. 437, December 1988 (in Japanese).

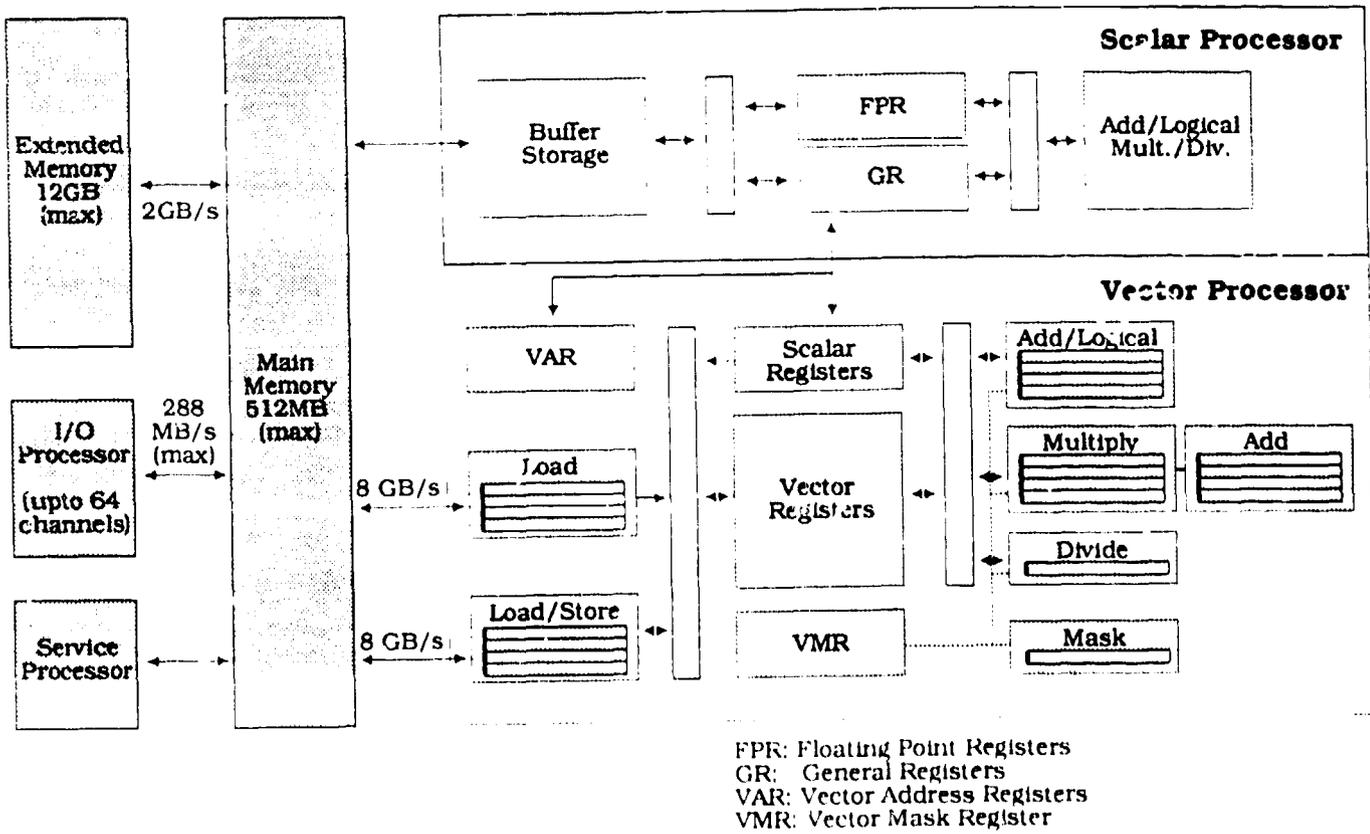
[3] J. M. van Kats, R. Llubra, and A. J. van der Steen, "A Close Look at the First Generation of Japanese Supercomputers," Technical Report TR-22, ACCU, Utrecht, 1986.

[4] Frank H. McMahon, "The Livermore Fortran Kernel: A Computer Test of the Numerical Performance Range," Lawrence Livermore National Laboratory document UCRL-53745, December 1986.

[5] Stephen C. Perrenod, "Automatic Parallel Processing for Science and Industry," Proceedings of the First Appi Workshop on Supercomputing., Appi, Japan, November 1987

[6] R. H. Mendez, "The Performance of the NEC SX-2 Supercomputer System Compared with that of the CRAY X-MP/4 and Fujitsu VP-200," to appear in "Parallel Computing."

Appendix A. S-820 Architecture



Appendix B: Description of Codes

Los Alamos National Laboratory Codes:

The Computing and Communications Division at Los Alamos National Laboratory maintains a set of portable benchmark programs representing characteristic tasks that a large supercomputer would be required to run at the Laboratory. This benchmark set has been run on a wide range of both scalar and vector machines. A database is maintained containing results of past runs of these programs on a variety of computers. The Los Alamos benchmark set consists of tests at the level of hardware demonstration programs, basic routines, and stripped down applications. A description of the codes follows. The programs described here are coded in ANSI Fortran for portability and can typically be run on a new machine with little or no change. Execution rates will be indicative of the potential initial usefulness of a new machine.

- INTMC:** An integer Monte Carlo code containing almost no floating point arithmetic. The random number generator requires at least 32-bit integer operations. There is no I/O, and all data are internally generated.
- FFT:** An FFT code that is highly vectorizable. This code measures the speed of single Fourier transformations. Because it executes many operations with short vector lengths, it is very sensitive to vector start-up times. FFT library routines supplied by all supercomputer manufacturers generally perform multiple FFTs at much higher execution rates than this benchmark code. No I/O is performed.
- VECOPS:** A code that tests rates of primitive vector calculations as a function of vector length. Vector operands and results are fetched from and stored to contiguous memory locations, except for four operations that involve gather/scatter. Typically one million floating point operations are timed.
- VECSKIP:** A code that performs the same operations as VECOPS. The vectors are accessed in noncontiguous memory locations with several values for the stride, which can be adjusted to test for performance during memory conflicts.
- MATRIX:** A code that performs basic matrix operations, including multiplication and transpose, on matrices of order 100. The code is highly vectorizable but not optimized for vector computers.
- GAMTEB:** A Monte Carlo photon transport code. This is a relatively small model code with a simple source and straightforward geometry. It is only slightly vectorizable.

PHOTON and

SCALGAM: Two very similar Monte Carlo photon transport codes that use the methods of GAMTEB, but with more complicated geometry, more materials, and more statistics gathered. Both codes require 64-bit arithmetic for its random number generator, as does GAMTEB, and neither vectorizes.

BMK21: A Monte Carlo neutron transport algorithm. The code is completely scalar and is similar to GAMTEB.

LSS: A linear system solver from LINPACK for systems of equations of order 100. It uses the method of Gaussian elimination. Although it is fully vectorizable, it is not optimized for supercomputers. Library routines supplied by supercomputer manufacturers will achieve considerably higher execution rates.

HYDRO: A two-dimensional Lagrangian hydrodynamics code based on an algorithm by W. D. Schultz. HYDRO is representative of a large class of codes in use at the Laboratory. The code is 100% vectorizable. A typical problem is run on a 100 x 100 mesh for 100 time steps.

SIMPLE: A two-dimensional Lagrangian hydrodynamics code with heat diffusion. The code is about 90-95% vectorizable, and it uses a 63 x 63 mesh.

Mendez Codes:

Five fluid dynamics applications codes gathered from different sources were used as testing instruments. The same five programs were used in an earlier comparison study of the Fujitsu VP-200 and CRAY X-MP systems [6]. These codes do not represent any given workload and are characteristic only of the types of fluid dynamics modeling used in these programs.

BARO: A two-dimensional shallow water model of the atmosphere that was developed on the CDC CYBER 205. The 61 loops of this code vectorized in all three systems amount to more than 99% of the total work. Memory accesses are contiguous, and vector lengths are moderately long at 300. Performance is dominated by vector speeds.

VORTEX: A particle code that simulates the dynamics of a one-dimensional vortex sheet by means of discrete vortices, developed on an IBM 3033 mainframe. In VORTEX as in BARO, memory accesses are contiguous, and the vector ratio is quite high (99% vector operation ratio).

EULER: A one-dimensional spectral code used to model the shock tube problem. This code was developed on Texas Instrument's ASC system. Because of the type of FFT used in this code, and because

it is a one-dimensional code, EULER is perhaps, within the benchmark set, least representative of the codes used in large-scale computing. The vectorization ratio is 73%.

MHD-2D and

SHEAR3: Two- and three-dimensional turbulence fluid dynamics simulation based on spectral techniques, which have been used extensively in turbulence simulations and were developed on Cray systems. The same FFT routine is used in both codes (different from the one in EULER) and accounts for most of the CPU time. The vector operation ratios of MHD2D and SHEAR3 are 99% and 89%, respectively.