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.

LA-UR-86-2890

B

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

TITLE BENCHMARK OF THE CONVEX C-1 MINI SUPERCOMPUTER

LA-UR--86-2890

DE88 009176

AUTHOR(S) Margaret L. Simmons
Olaf M. Lubeck

SUBMITTED TO Internal benchmark report available upon request.

DISCLAIMER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

BENCHMARK OF THE CONVEX C-1 MINI SUPERCOMPUTER

Margaret L. Simmons
Olaf M. Lubeck
Computing and Communications Division
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Introduction

In July 1985, we benchmarked the Convex C-1 computer at the Convex plant in Richardson, Texas. The machine is marketed as a mini-supercomputer executing a UNIX operating system. The architecture includes vector functional units, 16-million 64-bit words of physical memory and 64 kbytes of set-associative cache between main memory and the CPU. (See the next section for a more detailed description of the architecture.) The standard one-processor Los Alamos benchmarks were executed and timed in both single-precision (32-bit) and double-precision (64-bit) floating-point mode. Subsequent to the July benchmark, the machine architecture was changed to expand the cache bypass for vector memory accesses. The benchmarks were redone in October 1985 to include the significant architecture modification. The results in this paper are from the latest benchmark.

Convex C-1 Architecture

The Convex C-1 computer has a bus-based architecture that uses 64-bit integrated scalar and vector processing and multiple arithmetic units much like the Cray architecture. The machine employs virtual addressing but has a large physical address space (128 Mbytes).

The CPU consists of multiple asynchronous units capable of concurrent operation in a pipelined fashion. The major cycle time is 100 ns. These units are interconnected through 64-bit high-speed buses. The CPU is composed of four subsystems:

(1) The address translation unit (ATU) translates logical addresses to physical addresses before main memory references. It also provides some additional acceleration functions. The ATU contains a logical and an address cache to aid in speeding up accessing of memory addresses.

(2) The address and scalar unit (ASU) serves two functions. It generates logical addresses for those instructions that require it and it also performs scalar operations.

(3) The instruction processing unit (IPU) fetches and decodes instructions. The IPU contains a 1024-byte instruction cache. The size of this cache means that long series of instructions, including branches, can be executed without additional memory references.

(4) The vector processing subsystem (VPS) consists of a vector control unit and two vector processing units. In these vector processing units are three functional units: one for add, subtract, and logical; the second for load, store, and vector edit; and the third for multiply and divide. Each unit is pipelined and capable of chaining. The VPS contains a set of eight vector registers, each able to hold 128 64-bit elements.

Situated between the CPU units and main memory is a physical cache unit (PCU). It has a capacity of 64 Kbytes and is a direct mapped data store with an access time of 50 ns. The PCU is connected to both the main memory and the CPU units by high-speed (80-Mbyte/s) buses. Any load from main memory using the cache results in the four nearest words being loaded into cache. All scalar memory accesses use the cache. Contiguous vector loads and constant stride loads bypass the cache. Gather/scatter accesses are routed through it.

Input/output in the Convex is handled by an I/O subsystem. There are up to five I/O processors in this system capable of supporting up to 160 devices. There is also a service processor unit (SPU) that controls the operation of diagnostic programs and maintains a log of errors.

The Convex can contain up to 16 Mwords of CMOS memory, divided into up to eight memory modules (MAUs). Each memory module has two banks of 4-way interleaved memory. Each word has 64 bits of data and 8 SECDED bits (single error correction, double error detection). Main memory also contains two ports, one for the CPU and the other for the I/O.

Convex Compiler

The Convex compiler supports Fortran 77 syntax. The work that the company put into the compiler is evident in its excellent vectorization analysis. The benchmark codes were the same programs that we ran on the Fujitsu and Hitachi vector processors, and they vectorized at essentially the same levels as the Japanese compilers.

Results

Table I summarizes the Convex results. For comparison, CRAY-1 times and recent VAX-8600 times are also listed.

Code	Convex(32)	Convex(64)	8600(32)	CRAY-1(64)	Description
BMK1	89.1	90.1	106.6	63.9	Integer Monte Carlo
BMK4a	47.6	61.6	106.5	6.4	FFT
BMK5	254.3	410.0	149.9	28.8	Equation of State
BMK11a	77.2	103.3	171.11	3.0	Particle Pusher
BMK11b	74.8	94.5		8.1	Particle Pusher
BMK14	15.7	22.2	177.5	3.2	Matrix Calcuations
BMK18	10.2	18.2	617.0	2.4	Vector Operations
BMK21	25.2	33.1	15.3	5.0	Monte Carlo
BMK21a	49.1	70.1		8.9	Monte Carlo
BMK21b		446.4		159.6	Monte Carlo
BMK22	68.5	87.8	314.0	17.4	Linear Eqs. Solve
SIMPLE	23.8	31.6	40.28	2.9	Lagrangian Hydro (32x32)
HYDRO	447.4	509.5			Lagrangian Hydro (100x100)

TABLE I. Execution times (seconds) of the Los Alamos benchmark codes. Each column is denoted by the machine and precision (bits) of the floating-point operations.

A comparison of the 32-bit results from the Convex C-1 and VAX-8600 shows the following:

(1) The two purely scalar codes (BMK5 and BMK21) are 70% faster on the 8600. However, this margin is expected to decrease in 64-bit mode, which is supported in hardware in the C-1. The remaining times show the performance advantage of the vector processor in the C-1 over the 8600. These codes vary from 30% to 98% vectorization levels and execute from 1.12 to 5 times faster than the VAX.

(2) The 64-bit results comparing the C-1 and CRAY-1 indicate that the Convex is roughly one-fifth to one-eigth of a CRAY-1.

Tables II and III show the rate at which various vector loops execute on the C-1. Table IV is the timing of the Livermore loops.

Conclusions

Based on our benchmark results, it is interesting to compare price performance ratios of the three machines. The Convex has about the same price performance ratio as the CRAY-1, while the VAX-8600 is about a factor of 2 more for the same performance.

TABLE II. Times and rates for various vector loops on the C-1. (stored contiguously)

LABMX8A1

TIMES FOR 1 MILLION CONVEX C-1 VECTOR OPERATIONS
VECTORS ARE STORED IN CONSECUTIVE LOCATIONS
NSTEP=1

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	7.8134	4.3037	3.2279	2.7158	2.4099	2.3120
V=S*V	8.3970	4.3137	3.2288	2.6904	2.4304	2.2786
V=V+V	9.0673	5.4344	4.3047	3.7129	3.4673	3.3250
V=V*V	9.5764	5.4235	4.3037	3.6924	3.4683	3.3257
V=V+S*V	9.5947	5.5616	4.4021	3.7090	3.4700	3.3162
V=V*V+S	11.1107	6.0304	4.3940	3.7275	3.4902	3.3220
V=V*V+V	12.0941	7.3376	5.7747	4.8977	4.6589	4.4753
V=S*V+S*V	12.5623	6.8307	4.8884	3.9766	3.7776	3.5420
V=V*V+V*V	13.5804	8.4913	6.7571	5.8931	5.7112	5.5200
V=V(IND)+S	12.3354	7.0295	5.2346	4.6316	4.9063	5.2527
V(IND)=V*V	15.1321	9.3203	7.2373	7.6304	7.5554	7.6799
V(IND)=V(IND)+V*V	19.4594	12.0983	9.5581	11.2148	12.4556	14.1292
V=V+V*V(IND)	17.0355	10.4528	7.9807	7.0945	7.4578	7.5322
SUB CALLS	2.2085	0.8832	0.4417	0.2209	0.1206	0.0439

RATES (IN MFLOPS) ON CONVEX C-1
VECTORS ARE STORED IN CONSECUTIVE LOCATIONS
NSTEP=1

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	1.28	2.32	3.10	3.68	4.15	4.33
V=S*V	1.19	2.32	3.10	3.72	4.11	4.39
V=V+V	1.10	1.84	2.32	2.69	2.88	3.01
V=V*V	1.04	1.84	2.32	2.71	2.88	3.01
V=V+S*V	2.08	3.60	4.54	5.39	5.76	6.03
V=V*V+S	1.80	3.32	4.55	5.37	5.73	6.02
V=V*V+V	1.65	2.73	3.46	4.08	4.29	4.47
V=S*V+S*V	2.39	4.39	6.14	7.54	7.94	8.47
V=V*V+V*V	2.21	3.53	4.44	5.09	5.25	5.43
V=V(IND)+S	0.81	1.42	1.91	2.16	2.04	1.90
V(IND)=V*V	0.66	1.07	1.38	1.31	1.32	1.30
V(IND)=V(IND)+V*V	1.03	1.65	2.09	1.78	1.61	1.42
V=V+V*V(IND)	1.17	1.91	2.51	2.82	2.68	2.66
SUB CALLS	0.45	0.45	0.45	0.45	0.41	0.46

TABLE III. Rates for various vector loops on the C-1.

LABMK8A2

RATES (IN MFLOPS)

STEP = 1

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	1.06	2.02	2.90	3.67	3.74	4.15
V=S*V	1.00	2.02	2.90	3.67	3.89	4.14
V=V+V	0.93	1.67	2.11	2.65	2.77	2.82
V=V*V	0.89	1.65	2.11	2.58	2.77	2.91
V=V+S*V	1.81	3.23	4.29	5.11	5.50	5.77
V=V*V+S	1.56	3.11	4.38	5.12	5.48	5.73
V=V*V+V	1.46	2.49	3.32	4.01	4.07	4.25
V=S*V+S*V	2.11	3.98	5.67	7.31	7.40	7.85
V=V*V+V*V	1.85	3.29	4.14	4.97	4.84	5.20
V=V(IND)+S	0.66	1.23	1.63	2.01	1.96	1.85
V(IND)=V*V	0.59	0.94	1.26	1.32	1.27	1.14
V(IND)=V(IND)+V*V	0.97	1.36	1.61	1.68	1.51	1.19
V=V+V*V(IND)	1.01	1.80	2.41	2.87	2.77	2.56
SUB CALLS	0.45	0.45	0.45	0.45	0.45	0.46
S=S+V1(I)*V2(I)	1.16	2.36	3.71	5.05	5.28	5.30

STEP = 2

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	0.92	1.49	1.72	2.12	1.97	2.24
V=S*V	0.86	1.48	1.81	2.12	2.18	2.20
V=V+V	0.77	1.15	1.28	1.37	1.47	1.52
V=V*V	0.76	1.13	1.26	1.45	1.51	1.52
V=V+S*V	1.50	2.23	2.50	2.72	2.79	3.02
V=V*V+S	1.30	2.25	2.56	2.94	3.01	3.02
V=V*V+V	1.18	1.73	1.99	2.19	2.26	2.26
V=S*V+S*V	1.91	3.11	3.85	4.25	4.42	4.48
V=V*V+V*V	1.58	2.23	2.55	2.68	2.77	2.80
V=V(IND)+S	0.66	1.12	1.41	1.57	1.60	1.19
V(IND)=V*V	0.57	0.72	0.94	1.10	0.99	0.88
V(IND)=V(IND)+V*V	0.86	1.10	1.44	1.55	1.30	0.99
V=V+V*V(IND)	0.95	1.36	1.82	1.89	1.80	1.70
SUB CALLS	0.45	0.45	0.45	0.46	0.46	0.44
S=S+V1(I)*V2(I)	1.11	1.91	2.87	3.50	3.72	3.92

STEP = 4

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	0.66	0.90	1.05	1.11	1.16	1.17
V=S*V	0.66	0.88	1.07	1.12	1.12	1.14
V=V+V	0.55	0.66	0.74	0.78	0.79	0.79
V=V*V	0.55	0.63	0.75	0.73	0.78	0.78
V=V+S*V	1.07	1.30	1.48	1.52	1.56	1.55
V=V*V+S	1.00	1.30	1.49	1.52	1.54	1.56
V=V*V+V	0.80	1.02	1.14	1.17	1.17	1.18
V=S*V+S*V	1.39	1.90	2.18	2.29	2.31	2.34
V=V*V+V*V	1.06	1.23	1.34	1.41	1.41	1.44
V=V(IND)+S	0.51	0.84	1.04	1.16	1.08	0.89
V(IND)=V*V	0.41	0.62	0.71	0.75	0.75	0.58
V(IND)=V(IND)+V*V	0.65	0.96	1.10	1.09	1.01	0.74
V=V+V*V(IND)	0.69	1.03	1.13	1.12	1.15	1.13
SUB CALLS	0.41	0.45	0.37	0.45	0.45	0.46
S=S+V1(I)*V2(I)	0.88	1.42	1.79	2.06	2.09	2.18

TABLE III (cont'd)

RATES (IN MFLOPS)
STEP = 8

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	0.66	0.93	1.01	1.11	1.09	1.18
V=S*V	0.65	0.93	1.05	1.11	1.09	1.09
V=V+V	0.55	0.67	0.73	0.74	0.77	0.79
V=V*V	0.54	0.68	0.74	0.76	0.73	0.77
V=V+S*V	1.08	1.34	1.43	1.52	1.52	1.56
V=V*V+S	0.98	1.34	1.46	1.48	1.52	1.57
V=V*V+V	0.83	1.04	1.05	1.10	1.16	1.14
V=S*V+S*V	1.45	1.92	2.13	2.24	2.27	2.35
V=V*V+V*V	1.07	1.24	1.31	1.40	1.36	1.42
V=V(IND)+S	0.53	0.86	1.01	1.02	0.91	0.72
V(IND)=V*V	0.47	0.64	0.73	0.72	0.65	0.47
V(IND)=V(IND)+V*V	0.74	0.93	1.08	1.01	0.86	0.63
V=V+V*V(IND)	0.75	0.99	1.05	1.11	1.11	0.94
SUB CALLS	0.45	0.45	0.37	0.46	0.45	0.43
S=S+V1(I)*V2(I)	0.92	1.36	1.84	2.01	2.07	2.15

STEP = 23

OPERATION	VECTOR LENGTH					
	10	25	50	100	200	500
V=V+S	1.04	1.99	2.78	3.53	3.51	4.14
V=S*V	1.00	1.88	2.85	3.54	3.41	4.12
V=V+V	0.93	1.62	2.15	2.58	2.57	2.75
V=V*V	0.85	1.60	2.11	2.40	2.58	2.79
V=V+S*V	1.77	3.18	4.26	5.07	5.10	5.36
V=V*V+S	1.55	3.07	4.33	5.08	5.39	5.71
V=V*V+V	1.43	2.46	3.26	4.01	3.92	4.24
V=S*V+S*V	2.07	3.91	5.42	7.28	6.97	8.08
V=V*V+V*V	1.93	3.22	4.20	4.97	4.84	5.20
V=V(IND)+S	0.66	1.11	1.47	1.72	1.41	0.93
V(IND)=V*V	0.59	1.06	1.30	1.36	1.03	0.69
V(IND)=V(IND)+V*V	0.89	1.33	1.68	1.72	1.26	0.82
V=V+V*V(IND)	0.99	1.81	2.21	2.55	2.36	1.68
SUB CALLS	0.43	0.45	0.46	0.46	0.45	0.41
S=S+V1(I)*VSTOP:						

TABLE IV. Livermore Kernels.

CLOCK OVERHEAD = 0.76294E+02 USEC

LOOP	FLOPS	TIME	MFLOPS
1	200000	0.18977E+05	0.10539E+02
2	200000	0.36684E+05	0.54520E+01
3	200000	0.33513E+05	0.59679E+01
4	101700	0.19011E+05	0.53496E+01
5	199800	0.31879E+06	0.62674E+00
6	199800	0.29139E+06	0.68568E+00
7	192000	0.16649E+05	0.11532E+02
8	144000	0.14605E+05	0.98593E+01
9	170000	0.16684E+05	0.10190E+02
10	90000	0.37031E+05	0.24304E+01
11	99900	0.23933E+06	0.41741E+00
12	99900	0.43831E+05	0.22792E+01
13	89500	0.23124E+06	0.38747E+00
14	165000	0.28445E+06	0.58006E+00

AVERAGE MFLOPS= 0.47355E+01

SUM FLOPS / SUM TIME = 0.13430E+01

LOOP	RELATIVE DIFFERENCE BETWEEN EXPECTED AND CALCULATED VALUES
1	0.22E-13
2	0.81E-10
3	0.81E-10
4	0.15E+01
5	0.23E-14
6	0.22E-14
7	0.48E-12
8	0.00E+00
9	0.00E+00
10	0.00E+00
11	0.13E-11
12	0.00E+00
13	0.12E-13
14	0.57E-02