CERN-Data Handling Division
DD/85/25
October 1985

TWO RECENT SUPERCOMPUTERS, THE CRAY-2 AND THE SX-2

Tor Bloch*
Centre de Calcul Vectoriel pour la Recherche
Ecole Polytechnique, Palaiseau. France

Computing in High Energy Physics, June 25-28, Amsterdam
National Institute for Nuclear Physics, and
High Energy Physics, Section H (NIKHEF-H)
and
the Computer Science Dept. (FVI), University of Amsterdam

*on leave from Data Handling Division, CERN, Geneva, Switzerland

TWO RECENT SUPERCOMPUTERS, THE CRAY-2 AND THE SX-2

T. Bloch

Two of the most recent supercomputers, the CRAY-2 from Cray Research Inc. and the SX-2 from NEC Corporation are described and compared. Major trends seen are much bigger memories and multiple CPU's. The bigger memories will increase the comfort of the computational physicist and allow the use of more realistic models whereas the unavoidable MIMD architecture will cause a lot of new problems in development and debugging.

INTRODUCTION

We define a supercomputer as a state-of-the-art machine which is capable of solving problems in computational physics and simulation which cannot, in practice, be solved in any other way.

The total installed base of such computers is currently in the order of 150 machines world-wide if we consider that a CRAY-1 class computer (first delivered in early 1976) remains a supercomputer in the above definition. An "average" supercomputer costs about $ 10 million but the price of the fastest ones now approach $ 20 million. Computational physicists are expanding their usage as the latest improvements in memory sizes and speed allow important improvements in the quality of the modelling of physical phenomena. At the same time industrial usage is increasing as the cost-performance ratio of new machines improve to a point where computational modelling techniques become more and more economically advantageous: animation in the film industry, oil reservoir simulation, structural calculations, aerodynamics, electronic design, etc.

On the other hand supercomputers are little used in the area of data processing. In addition, the huge central memories of the latest supercomputers are, probably, much too large for these problems which deal with relatively little data at a time ('events' or pictures) but one should now ask the question whether the ability to have so much data resident in the central memory could open new horizons for this type of data processing.

We have chosen to describe two of the most recent supercomputers, the CRAY-2 and the NEC SX-2, in some detail in order to see if we can detect some recent trends. This choice is motivated by the fact that of the other supercomputers announced today the CRAY X-MP and the Fujitsu VP-200 are already described in the literature whereas the ETA-10 is a little too far from its first delivery for details of its design to be available.

Warning: the SX-2 has not been delivered and the description presented here is based on available publications only; it may contain some minor inexactitudes.

THE CRAY-2

The first production model of the CRAY-2 was delivered to the Magnetic Fusion Energy Computer Centre at Lawrence Livermore Laboratory in June 1985; NASA Ames Laboratory and the University of Minnesota are the next customers.

The CRAY-2 runs under the control of the AT & T UNIX system V operating system and accepts two types of peripherals at this moment: large disks and terminals (using the TCP/IP interface).

For all other types of input/output the CRAY-2 will use front-ends exactly like the CRAY-1 did when it was first delivered in 1976.

The most dominant traits of the CRAY-2 are:

- [illegible] central memory, 268 435 456 words
- four identical background processors
- a cycle time of 4.1 ns
- size : 1 meter tall and 1.20 meter in diameter (the liquid immersion cooling permits the dissipation of the 195 Kwatts developed in this space).

Detailed technical features

Memory

- 268 435 456 64-bit words organized in 128 banks
- 60 ns cycle time DRAM 256 Kbit chip
- 242 ns (scalar) to 262 ns (vector) access time (to registers)
- 1 Gwords/s total bandwidth
- Addressing in each processor offset by the contents of a fixed base address register and controlled by the contents of a limit address register.

CPU

- Four independent background processors capable also of working on the same application in MIMD (Multiple Instruction, Multiple Data) mode using eight semaphore flags available for coordination.

Per processor

- 4.1 ns clock period
- 512 64-bit vector registers, organized as 8 registers of 64 words each, 8 64-bit scalar registers and 8 32-bit registers for addresses
- 1 6-bit register for the vector length, 1 64-bit register for vector masks
- a real time clock
- a local memory of 16384 64 bit words with an access time of 45 ns (scalar) or 51.5 ns (vector) to registers
- 7 fully pipelined specialized functional units (plus two for 32-bit address calculations
- 122 MIPS (maximal instruction issue rate once every two clock periods)
- 488 MFLOPS maximum (one result from each of the two floating point pipelines per clock period)
- an instruction buffer of 8 'pages' of each 16 words (up to 4 instructions per word).

Instructions

127 different instructions including PASS and the only five systems control functions available: EXIT, SET SEMAPHORE, CLEAR SEMAPHORE, JUMP IF SEMAPHORE CLEAR AND SET, JUMP IF SEMAPHORE SET AND SET.
I/O, context switching and memory refresh are dealt with by a 32-bit 'foreground processor' and 'dead start' is performed by a standard microprocessor from a diskette.

Instructions are generally of the 3 address register-to-register type both for vector and scalars. A scalar register can be 'expanded' in almost all vector instructions to behave like a vector register with 64 identical operands. The Vector Length (VL) Register permits the handling of vectors shorter than 64 elements and the Mask Register (M) permits selective operations and mergers to take place between vector registers.

Loads and stores for the 8 vector registers (each 64 words long) and the 8 scalar registers are controlled by the addresses held in the 8 32-bit address registers both as far as the main memory and the local memory are concerned. Both SCATTER/GATHER and constant stride access is available on the main memory whereas the local memory access is limited to consecutive loads and stores (its purpose is to keep intermediate results).

Input/Output

4 channels of 500 Mbytes/s each communicate directly with the main memory.

Context switching

Each background processor is connected to one of the channels providing the vehicle by which it communicates with the other processors (setting and testing semaphores) and with the foreground processor. The foreground processor is connected to all four channels.

Functions sent by the foreground processor over a channel to the background processor permit the interruption, and restart of execution.

The background processor flags and status registers can be read and altered by the foreground processor over the appropriate channel (e.g. idle flag, P-counter, flag, range error interrupt flags, memory error interrupt flags, limit address, etc.).

The actual context switch (storing of register contents and reloading) is done by a system program initiated in the background processor by the foreground processor. This is, in general, the approach chosen by the operating system: the foreground processor acts as an interrupt handler but any real systems code to be executed is written for a background processor and executed in slave mode when needed.

Technology

- ECL gate array logic with 16 gates per chip and a switching time of 350 ps
- Up to 108 chips are mounted on each 10 cm x 20 cm printed circuit board which in turn are assembled 8 at a time to a module (2.5 cms thick)
- Connections within such a module are made in all three dimensions and the total system population is 320 modules.

The modules are stacked in a cabinet consisting of 14 columns arranged in a 300° arc. Inside this mainframe cabinet an inert fluor carbon liquid at room temperature circulates in direct contact with the integrated circuit packages. The heat is evacuated from the liquid by circulating it through an external chilled water cooled heat exchanger.

Comparison with the CRAY-1

The CRAY-2 background processor has a strong similarity with a CRAY-1 processor and it is interesting to make a short comparison of the two. This analysis should not, however, obscure the fundamental fact that the architecture of a CRAY-2 is by no means a simple collection of four CRAY-1's; the interrupt and I/O control systems are radically different quite apart from the fact that the size of the main memory dominates to an extent where the whole balance of the machine has changed completely.

A CRAY-2 processor can be viewed as a CRAY-1 processor with:

- fewer instructions overall: 127 instead of 180
- some new functionalities: SCATTER/GATHER, compressed iota (allowing quick access to operands identified by, for example, a threshold test), approximate square root
- replacement of the blocks of registers (B and T) with a comfortably sized local memory which, additionally, can receive intermediate vector results
- no chaining between vector operations (the code generators must aim at maximum overlap between load/stores and execution instead)
- peak vector rates improved by a factor of three per processor, but scalar rates not much different from a CRAY-1. This is because the basic speed-up of the clock period (4.1 ns instead of 12.5 ns) is obtained only partly (factor of 1.5) by a real speed up due to size and circuits. The rest is gained by having fewer gates per clock period. This penalizes scalar execution speed because more of each clock period ends up being used for the latching of intermediate results in the pipelines
- an access time to the main memory almost twice as long (but we think that this can be more than offset by good use of the local memory for the storing of intermediate results).

Models available

Only the one described; no options in terms of memory size or number of CPUs are available.

THE SX-2

The SX-2 was announced by the NEC Corporation in early 1983 and the first production model of SX-2 is to be delivered in Japan during the summer of 1985.

The SX-2 runs under the control of the ACOS operating system with a high level of IBM compatibility in its user interface: tape formats, floating point number formats, FORTRAN 77, etc. There is no need for a front-end computer.

The most dominant traits of the SX-2 are:

- a large central memory: up to 256 Mbytes with up to 2 Gbytes of secondary memory (DRAM)
- a scientific processing unit for the user programs and a control processor for the operating system both working out of the same central memory
- a cycle time of 6 ns
- quadruple floating point pipelines for each type of operation working in staggered mode (the first one deals with elements 1, 5, 9,..., the second with elements number 2, 6, 10,..., and so on).

Detailed technical features

Memory

- 33 554 432 (maximum) 64-bit words organized in 512 banks
- 40 ns cycle time SRAM 64 Kbit chip
- 1.375 Gwords (64 bits) per second total bandwidth
- Address translation based on 1 Mbyte pages
- Access time (memory to vector registers) probably slightly more than 200 ns.

Scientific processor

- One integrated processor for the execution of user code; rather clearly separated into a vector part and a scalar part
- 6 ns clock period
- 2048 64-bit registers for vectors, organized as 8 vector registers of 256 words each
- 1 256-bit register for vector control (masking, merging, etc.)
- 8 mask registers of 256 bits each
- A local memory of 8192 64-bit registers organized as 32, 64 or 128 vector registers of 256, 128 or 64 words each
- 128 64-bit scalar registers for operands and addresses
- 16 pipelines for vector operations organized in 4 sets of 4 synchronously cooperating pipelines for each of the following : Add, Multiply/Divide, Logicial, Shift
- Scalar operations are fully pipelined within the scalar processor
- Addresses and shift counts are handled by the scalar processor but otherwise the interface between the scalar processor and the vector operations seems to be limited to the broadcast of a scalar value to a vector register
- 167 MIPS (a maximum instruction issue rate of one per clock period)
- 1.3 GFLOPS maximum (one result from each of 4 collaborating pipelines (one operation) per cycle fully chained with another vector operation)
- A cache of 65 536 bytes for the scalar processor only (for operands and instructions)
- An instruction buffer of 2048 bytes with branch history logging permitting 'lookahead'; it is assumed that a conditional branch will behave as it did last time through the loop.

Instructions

167 different instructions (86 vector ones) of, generally, 3 address register-to-register type. The Vector Control Register and the Mask Registers permit selective operations and mergers to take place between vector registers.

Both SCATTER/GATHER and constant stride memory access are available.

All floating point arithmetic is done in 64-bit mode in the vector processor but 32-bit formats are available for data stored in the main memory by truncation/expansion. 128-bit format arithmetic is available in scalar mode only.

A first order iteration instruction is available to handle the following cases of recurrences: $X_i = A_i + X_{i-1}*B_i$, $X_i = (A_i + X_{i-1}).Bi'$ $X_i = A_i + X_{i-1}$, $X_i = A_i*X_{i-1}$.

Special instructions permit the halving, doubling and squaring of vector data.

Input/output

Handled by the control processor and its I/O processor over up to 32 channels, each capable of a transfer rate of up to 3 Mbytes/s.

The aggregate data transfer rate is limited to 50 Mbytes/s.

Technology

- CML bipolar VLSI logic with 1000 gates per chip and a switching time of 250 ps
- 1 K-bit bipolar memory chips with an access time of 3.5 ns are used for vector registers and the cache
- 36 chips using leadless chip carrier packaging are mounted on a 10 cm x 10 cm multilayer ceramic board which in turn are mounted, one at a time, in water cooled individual modules
- large multilayer printed circuit boards form the backplanes and are connected with coaxial cables.

Models available

A slower version of the SX-2, the SX-1, is available with a cycle time of 7 ns and only half as many registers and pipelines giving it a peak speed of 570 MFLOPS. The smallest SX-1 available is a 64 Mbytes version and the smallest SX-2 is 128 Mbytes.

COMPARISON BETWEEN THE CRAY-2 AND THE SX-2

Introduction

It is probably impossible to make a meaningful comparative benchmark of two supercomputers. The NEC SX-2 and the CRAY-2 each have a peak vector speed about two orders of magnitude faster than the worst case scalar speed. What this means in practice is that the selection of an algorithm -or even a problem formulation!- that vectorizes becomes much more important than the number of operations per second for any real performance measure of the time it takes to solve a specific computational physics problem. Furthermore, even for simple problems, the complexity involved in writing optimal code for multiple processor vector computers means that the quality of libraries and their optimal use becomes crucial; the unavailability of a vector random number generator or a mediocre Fast Fourier Transform routine can cripple major applications.

In any case the SX-2 as well as the CRAY-2 are both too recent for published FORTRAN benchmarks being available so the following remarks have to remain rather general.

Vector speed

The theoretical peak speeds (register-to-register operations) of an SX-2 and a CRAY-2 (all 4 processors working on the same problem) differ little; 1.3 GFLOPS for the SX-2 on chained operations versus, maybe, 1.9 GFLOPS for the CRAY-2 when the two floating point pipelines of each of the four processors work simultaneously on the same problem.

But when one compares the speed of complete vector operations the start-up time (the time it takes to get the first result out of a vector operation) and the memory access time are also of primary importance since many applications actually

work with relatively short vectors or with huge data structures which require frequent access to the main memory.

Start-up times are comparable between the two computers, probably in the order of 60 ns on the SX-2 and around 80 ns for the CRAY-2. Please notice that these start-up times will decrease peak performance on vector lengths of 256 from 1.9 and 1.3 GFLOPS respectively to 1.1 and 1 GFLOPS (for the CRAY-2 we assume, rather theoretically, that the 4 processors share the execution of vector operations of length 256).

The penalty incurred when main memory references have to be made is most probably a little over 200 ns in the case of the SX-2 and about 260 ns for the CRAY-2. Details about chaining and overlap are not available at the time of writing for the SX-2 but it would seem that on an isolated operation such as $A_i = B_i * C_i$ the execution for both computers rate would drop to around 500 MFLOPS really achieved, probably a little more on the SX-2 and a little less on the CRAY-2.

In reality, however, use of the local memories for intermediate results and proper overlapping of register loads and stores with floating point instructions should allow real peak rates on well suited operations such as matrix multiply to come within 20 or 25% of the respective peak speeds of the two machines.

Scalar speeds

The scalar speed of the SX-2 is expected to be at least a factor of two faster than the scalar speed of one CRAY-2 processor. In actual applications the scalar performance of either of the two machines will be dominated by the time to reference main memory if frequent memory references have to be made (in both cases the main memory is much slower than that of a CRAY-1). Memory access is speeded up automatically on the SX-2 through the use of a hardware cache while the CRAY-2 uses the local memory with explicit instructions for the storing of intermediate results and it is not possible at this stage to assert that either of these approaches results in a distinct advantage.

But the CRAY-2 has four processors! So the fundamental question becomes: 'Can typical scalar codes be split into 4 pieces at a reasonable cost or not?' On the answer to this question depends not only this comparison but also whether real future improvements can be made by going to 8 or 16 processors -speed improvements of a factor 2 or 4 which are most unlikely to be obtainable in any other way in the medium term.

Early experiments with the multiprocessor CRAY X-MP's seem to indicate that efficient partitioning of complete codes into 2 or 4 parts can be achieved in several fields such as meteorology, plasma physics and turbulence. But not much real experience is available yet.

A final point is that it is unclear to us if the very radical separation of the SX-2 into a vector part and a scalar part will cause a performance degradation in codes where intermixed scalar and vector calculations on the same data are required.

TRENDS

The most striking tendencies in the most recent commercially available supercomputers are:

- much bigger semiconductor memories
- continued peak speed improvements
- larger and larger gap between 'worst' case speed and peak speed
- not much development in the areas of I/O and software (FORTRAN).

We also believe that another significant speed increase of factor of five or ten necessitates the use of a MIMD approach -this in spite of the present balance between the SIMD approach (S-810/20, VP-200, SX-2) and the MIMD approach (X-MP, CRAY-2, ETA-10). If this prediction is true a lot of work with codes, algorithms and debugging techniques will have to be done by the users of this equipment over the next decade.

A few more comments on each of these general areas follow:

Memory and I/O

The I/O systems are not keeping up with the speed increases; neither in volume nor in speed. The fastest disks available today, the DD49 for the X-MP, are three times the transfer rate and half the access time of those installed on the CDC 7600 ten years ago. The disks available on the SX-2, the S-810/20 and the VP-200 are limited to a 3 MB/s transfer rate (due to the IBM channel...) -two thirds of that of the above mentioned disks from 1975.

To alleviate this problem bigger memories are needed thus avoiding the heavy input/output requirements which arise when the data does not fit in the main memory. In most cases today (S-810/20, SX-2, X-MP4) the main memory can be as big as 16 Mwords or 32 Mwords (64-bits) and a secondary DRAM memory ('electronic disk') option one order of magnitude bigger is available. In one case, the CRAY-2, the main memory is 256 Mwords.

We believe that this increase in size of the CRAY-2 of two orders of magnitude over the CRAY-1 is a step of capital importance: it brings within reach problems which could not be treated realistically before and it should allow much faster development of new application codes as the scientists are relieved of the burden of organizing the data on the disks (electronic or rotating), optimizing the transfers, etc, etc.

It is also clear that new problems will be posed with a main memory so big that it takes minutes to swap it to disks and requires 10 or 15 magnetic tapes for dumping it. Two are already clearly posed:

- some models cannot benefit from the whole memory because the computing time necessary to treat the finer resolution in a reasonable time would be excessive

- necessary intermediate outputs of fields developing in (simulated) time could easily become too voluminous for realistic handling afterwards. This will require new techniques for 'data reduction' one of which will probably be a much stronger reliance on graphics.

Architectural balance (Vector/Scalar)

As the execution speed of vector code increases more and more relative to that of scalar code one can rightly fear the paradoxical situation where the performance of the biggest vector computers becomes completely dominated by their scalar speed.

The answer may partly be that as models grow to fill the latest generation of supercomputers the vectorizable part increases much more than the scalar part (this is intuitively obvious when one imagines the expansion of a 2-dimensional model to a 3-dimensional one). It is also hoped that most scalar codes can be partitioned to run on several processors in parallel.

We believe that multiple processors collaborating on the same application programs represent the only way to reach higher and higher effective speeds; this is the

approach taken by the X-MP, the CRAY-2 and the future ETA-10. It would seem unlikely that today's machines can be speeded up by more than a factor of two or three by technology alone; physical size, basic circuit speeds and memory access times would be limiting factors. Thus more parallelism will be necessary and this has to impact today's scalar execution speed as well in order to avoid a significant change of the architectural balance. The only method known today to have a real hope of working on a wide range of applications is a (modest) number of cooperating processors (MIMD).

Software and FORTRAN

FORTRAN will remain the language because of its compatibility with the past; a new supercomputer needs to be amortized over at most 4 or 5 years. While waiting for a new FORTRAN standard with some vector statements incorporated the compilers will still need to 'vectorize' DO-loops and use complicated analysis techniques to try to find alternative ways of treating IF statements!

Vector statements, when they come, will not solve all problems neither: they will not choose the best numerical algorithms for the user and they will not lay out the data for the programmer. Now will they partition automatically a code in 4,8 or 16 independent pieces each executable on its own processor!

The debugging problem for multiprocessor codes must be emphasized here -it may not be too difficult to split a code satisfactorily in several independent pieces but it is certainly very, very difficult to find the cause of any errors discovered when using multiple processors in asynchronous execution.

In summary, the end user will have to invest significantly in the choice of algorithms as well as in vectorization and multiprocessing over the next few years if he wants to benefit from the higher and higher projected peak performance of supercomputers. A very positive element is that the advent of very large memories might relieve him of much of the organizational work for the input/output so typical of the development of large application codes today.

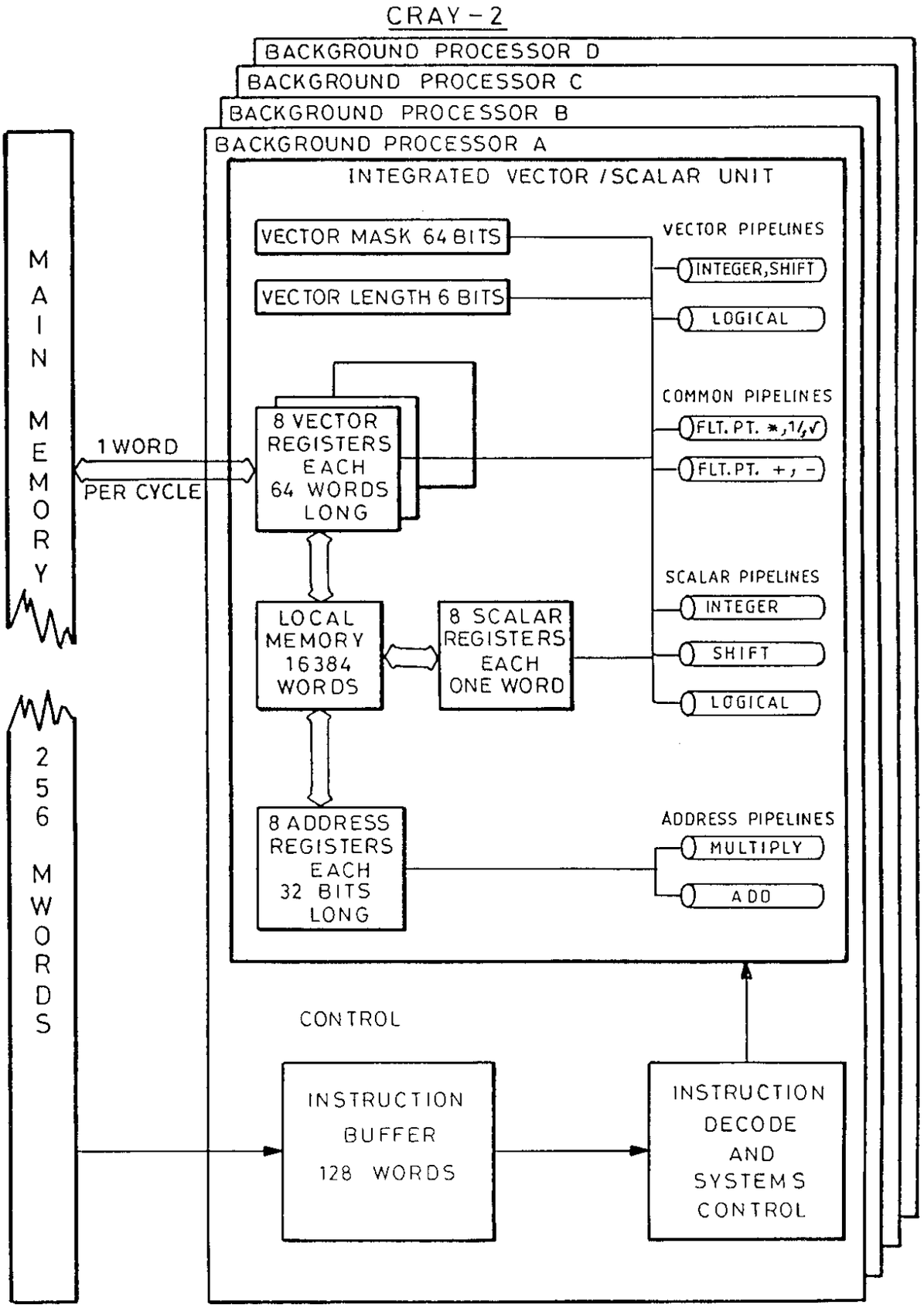
FINAL REMARKS

Each time a new generation of supercomputers offers increased computing power it becomes possible to attack new problems in computational physics. Although supercomputers such as the two described in this article will permit some numerical models to be extended by one dimension, mainly by virtue of their large memories, an improvement in the computing speed of another order of magnitude can already be seen to be necessary over the next few years.

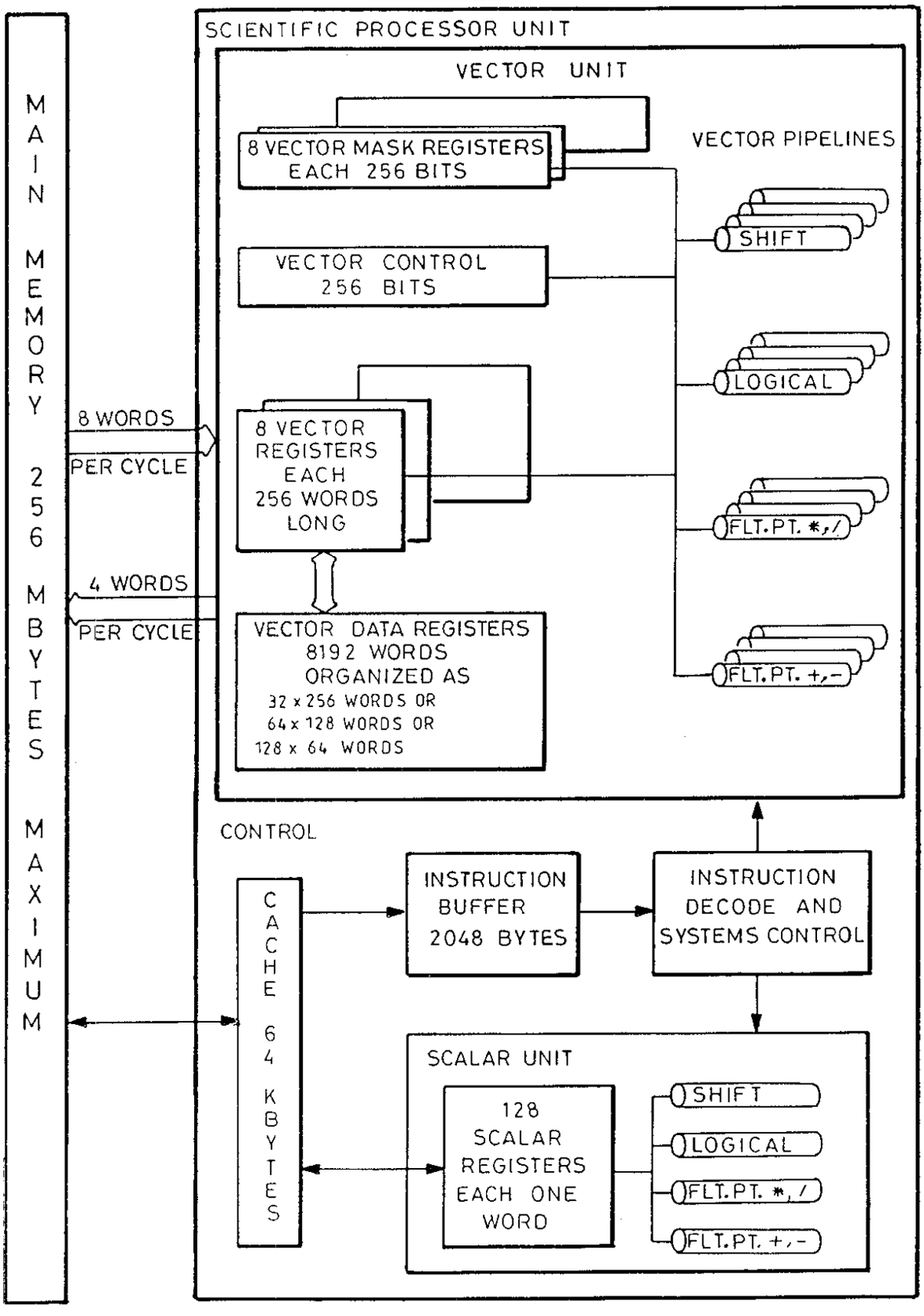
There also remains some areas of computational science where the gap to 'useful' speeds is still perceived as being several orders of magnitude (e.g. theoretical physics and molecular dynamics) and here many research groups have started projects to develop and construct supercomputers, often in collaboration with manufacturers.

Frequently these projects rely on architectures with a large number of processors with memory access through a sophisticated network but other new architectures such as data flow and cellular automata are also being studied by implementors.

Although most of these projects are still in too early a phase to draw any conclusions the experimentation with the architectures and applications which they imply will certainly, in the long term, have a significant impact on the field of scientific computing. It is to be regretted that compared to the United States and Japan, there seems to be relatively few such projects being actively pursued in Europe. The likely result will be to increase the already existing technology gap in conceiving and constructing large computers as well as possibly creating a new one in the usage which one is able to make of them in the future.


CRAY-2
BACKGROUND PROCESSOR D
BACKGROUND PROCESSOR C
BACKGROUND PROCESSOR B
BACKGROUND PROCESSOR A
INTEGRATED VECTOR/SCALAR UNIT
MAIN MEMORY
256 MWORDS
1 WORD
PER CYCLE
VECTOR MASK 64 BITS
VECTOR LENGTH 6 BITS
8 VECTOR REGISTERS EACH 64 WORDS LONG
LOCAL MEMORY 16384 WORDS
8 SCALAR REGISTERS EACH ONE WORD
8 ADDRESS REGISTERS EACH 32 BITS LONG
VECTOR PIPELINES
INTEGER, SHIFT
LOGICAL
COMMON PIPELINES
FLT. PT. *, 1/
FLT. PT. +, -
SCALAR PIPELINES
INTEGER
SHIFT
LOGICAL
ADDRESS PIPELINES
MULTIPLY
ADD
CONTROL
INSTRUCTION BUFFER 128 WORDS
INSTRUCTION DECODE AND SYSTEMS CONTROL

NEC SX-2


MAIN MEMORY 256 MBYTES MAXIMUM
8 WORDS
PER CYCLE
4 WORDS
PER CYCLE
SCIENTIFIC PROCESSOR UNIT
VECTOR UNIT
8 VECTOR MASK REGISTERS EACH 256 BITS
VECTOR CONTROL 256 BITS
8 VECTOR REGISTERS EACH 256 WORDS LONG
VECTOR DATA REGISTERS 8192 WORDS ORGANIZED AS
32 x 256 WORDS OR
64 x 128 WORDS OR
128 x 64 WORDS
VECTOR PIPELINES
SHIFT
LOGICAL
FLT.PT. *,/
FLT.PT. +,-
CONTROL
CACHE 64 KBYTES
INSTRUCTION BUFFER 2048 BYTES
INSTRUCTION DECODE AND SYSTEMS CONTROL
SCALAR UNIT
128 SCALAR REGISTERS EACH ONE WORD
SHIFT
LOGICAL
FLT.PT. *,/
FLT.PT. +,-