New Developments at the Fermilab Advanced Computer Program (I.): A Second-Generation Multiprocessor for Experimental High-Energy Physics

T. Nash, H. Areti, R. Atac, J. Biel, A. Cook,
J. Deppe, M. Edel, M. Fischler, I. Gaines, R. Hance, D. Husby,
M. Isely, M. Miranda, E. Paiva, T. Pham, and T. Zmuda

Introduction

High-energy physicists, like most scientists, have always wanted more computer power than they could afford to buy. In the commercial marketplace, the emphasis is on software-backward compatibility, product differentiation, and isolation of a client "herd." On the scientist's computing agenda, raw processing power in a relatively easy-to-use form, not corporate profitability, is the dominant issue.

Despite the fact that industry is not motivated by the rest of the market to provide the extremely cost-effective computer systems demanded by much of science, it does provide an extraordinary array of components (chips, modules, peripherals, work stations, software, etc.) that can be assembled into what science requires. For over five years, the Advanced Computer Program (ACP) at Fermilab has drawn from industrial components to design and produce usable parallel computer systems of such cost effectiveness that high-energy physics (HEP) experiments are now being carried out that would otherwise be unthinkable.

In addition to its dependence on the latest offerings of industry, the ACP also takes advantage of developments coming from computer science. However, just as with industry, the motivations of natural and computer scientists have significant areas of divergence. When commissioning a new architecture prototype, the interest of computer science primarily extends to proof of principle. Although also concerned with the latest from computer science, the ACP has the goal-driven motivations of its scientific clientele. The ACP's activities, therefore, fall in the rather wide gap between the exploratory spirit of computer science and the

---

This article is based on a talk given by T. Nash at the "Workshop on Computational Atomic and Nuclear Physics at One Gigaflop" at Oak Ridge, Tennessee, April 14-16, 1988, and is also available as Fermilab preprint Conf-88/97.

production reliability and conservatism of industry. ACP systems must both take advantage of advanced concepts and components, and be robust enough to satisfy a large community of demanding users.

In common with computer science, the ACP is aiming toward reasonably general-purpose parallel computers. The difference is that the ACP cannot focus, as computer science usually does, on the often academic complications of the end goal. Rather, the ACP must take a step-wise approach, dealing with the easier, more obviously parallel, problems first. At each step, as more complex issues are addressed, there must be a fully working, commercializable machine that delivers the maximum cost effectiveness available at the time.

With their different perspective, natural scientists have brought to the computer-development community new philosophical and architectural approaches to investigate. The dynamic nature of scientific problems means that programs are frequently changed, often extensively. The scientific user generally has a deep understanding of the structure of the problem and the capability and willingness to map the algorithm to optimally cost-effective computer architectures. Issues of software compatibility can certainly become painfully acute at mundane levels like compiler syntax. However, when it comes to the global scale of a problem, scientists are virtually unique at this time in their receptiveness to explicit (user directed), as opposed to implicit (automatic), decomposition of programs onto parallel computers.

Fox and Seitz at Cal Tech, driven initially by the needs of theoretical physics, established the viability of hypercubes, the quintessential explicitly parallel, local memory architecture. These are now the subject of much attention in computer science, which earlier had focused heavily on global memory architectures and attempts at automatic parallelization of whole algorithms. Similarly, the natural parallelism inherent in the experimental HEP computing problem (running the identical reconstruction program on many millions of different events) suggests a simple parallel-processing solution. The pioneering emulator work by Kunz et al., at SLAC, demonstrated the feasibility of using multiprocessor systems to provide cost-effective computing for the reconstruction of experiment events.

Almost a decade after the first SLAC emulator, powerful 32-bit microprocessors allowed the ACP at Fermilab to develop even more convenient and cost-effective event-oriented parallel-processing systems using many more CPU's. Such systems are now an acknowledged important component of computing in high-energy physics. They have made crucial contributions to data analysis in a number of current experiments, and will be indispensable for the large experiments of the Superconducting Super Collider era. The first-generation ACP sys-

tems are rapidly becoming a standard. There are over 30 installations in universities and laboratories worldwide, primarily, though not exclusively, for HEP applications. The first of these ACP systems, which provide CPU power at a cost of less than $2500 per VAX 780 equivalent, were brought online two years ago. The initial 110 processor system, along with several newer installations, have been heavily used at Fermilab for physics event reconstruction.

In this first of two articles, we will cover the new ACP systems for experimental HEP, which, early in the next year, will provide well over an order of magnitude increase in cost effectiveness over the original systems. This second generation project will also allow much higher bandwidth for both I/O and interprocessor communication, and will have software tools allowing almost any UNIX-, VMS-, or (potentially) VM-based processor to be used equivalently as a node or "front end" in a multiprocessor ACP system. Perhaps most important is the way in which this new system allows for integration of powerful "back end" multiprocessors, now usable for both reconstruction and physics analysis, into a modern Ethernet-based workstation environment. In the September/October issue of *Fermilab Report*, we will discuss the multi-array processor system for theoretical physics.

The mandate for ACP development remains the computing needs of high-energy physics. However, the potential applicability of these new systems, particularly the one aimed at theoretical physics, is far broader than HEP alone.

The Second-Generation ACP Multiprocessor

The continued saturation of computers (including ACP systems) by HEP experimenters motivates the development of a second generation of the successful ACP Multiprocessor system. Industry is continuing to provide a stream of often surprisingly powerful technology that we can harness to meet the severe challenges of this science. Frankly, some of industry's new components are making our task particularly pleasurable and rewarding these days. A variety of new and increasingly powerful microprocessors are now available to incorporate into multiprocessors. Beyond the new versions of existing processors (like the 25-MHz 68030s replacing 16-MHz 68020s), there are entirely new families of chips. Most of these are based on the Reduced Instruction Set Computer (RISC) architectural philosophy. It was noted in the 1970s that many complex instructions in traditional machines with large instruction sets, like the IBM 360s and DEC VAXes, were rarely used. They effectively increased the cycle time for all instructions because they mandated extensive microcode. RISC machines are generally pipelined with no microcode and very fast instruction cycles. They defer complex instructions to software.

New processors that will offer at least a factor of 3, and in one case as much as a factor of 20, more performance than the first-generation ACP processors based on 68020s include: the MIPS R2000 and R3000 RISC chips; the Fairchild, now Intergraph, Clipper chip set; the AMD 29000 RISC chip; the Sun SPARC RISC chip; the INMOS T800 transputer; the Motorola 88000 RISC chip; and the National Semiconductor 32532 processor. Unlike earlier processors, many of these have usable FORTRAN and C COMPILERS and UNIX operating systems even before the hardware is available.

The broad availability of UNIX is particularly important. As we will describe below, the new ACP architecture will support any processor running UNIX (or VMS) that can be connected via VME or Ethernet. It is very difficult to predict with any certainty which processor or commercial single board (or single slot) computer (SBC) will be most cost effective in the future. The openness of the ACP system permits competitive purchase of processor nodes based on performance benchmarks and price.

The large increase in CPU power available for the second-generation ACP system requires a redesign of the multiprocessor hardware and software system architecture to remove bottlenecks in current systems that would be felt at the higher performance levels. The bottlenecks are I/O and interprocessor communication bandwidth, and the CPU power available for the host process. Continuing the successful strategy of attacking computing limitations with parallelism, the new architecture solves I/O and host limitations by supporting parallel I/O and parallel host processing. Moreover, to avoid mini-computer bus bandwidth restrictions, any node may take on host functions, including I/O, through controllers in its own local crate. Though significant, the changes in the new system hardware and software are designed to be as transparent as possible to users of the original system.

In the following, we describe the fast new ACP SBC, and then the new interconnection modules and topologies, the new (parallel) I/O options, and the new software system architecture, all designed to support the large increase in performance that will be available.

A Fast New ACP Processor Node

As it has in the past, the ACP is encouraging competition in SBC's targeted at parallel processing by developing an extremely cost-effective VME SBC. The choice of the R3000 RISC processor from MIPS Computer Systems, Inc., for this design was based on performance evaluations of the microprocessors referred to earlier. The ultimate standard is how real physics code runs in a high-level language, since it does not matter how many million instructions per sec-

ond (MIPS) the CPU can execute if the instructions are not useful in FORTRAN or C, and if the compilers fail to provide sufficient optimization. The ACP has performed benchmarks on several of the new chips using a suite of high-energy physics FORTRAN programs (a small Monte Carlo/track reconstruction package; an actual fixed-target tracking code running on experimental data; and a floating-point-intensive theoretical calculation). These benchmarks were run on a 16-MHz MIPS R2000 system. For the three benchmarks, performances were 7.9, 6.4, and 7.4 times that of a VAX 780, which is generally accepted as the standard 1-MIP normalization. The new ACP node will use an R3000, the new improved version of the R2000. Carefully taking into account processor, cache, as well as clock speed differences, leads to a projected HEP code performance of 12-15 VAXes for 25-MHz R3000 boards and 16-20 VAXes for the 33-MHz versions we expect to use.

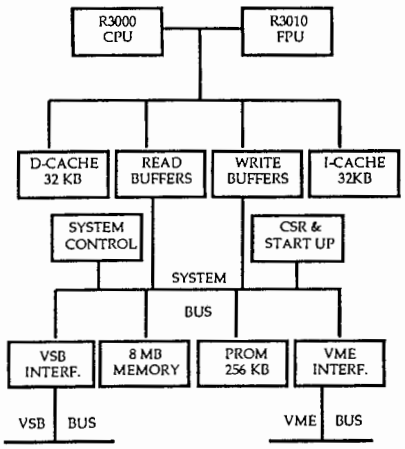

Fig. 1. The ACP MIPS processor, block diagram.

The ACP MIPS processor module (Fig. 1) consists of:

1) a 25- or 33-MHz MIPS R3000 CPU;

2) a 25- or 33-MHz MIPS R3010 floating point unit;

3) Four Write Buffers;

4) a 32-KByte instruction cache;

5) a 32-KByte data cache;

6) 8 MBytes of parity-checking main memory, made up of 1-Mbit (100-nsec nibble mode access) drams, expandable to 24 MBytes via VSB (VME System Bus);

7) interval timers that can interrupt the CPU;

8) a full-function VME Master Slave interface supporting 20 MBytes per sec block transfers;

9) a full-function VSB interface for peripherals and memory extensions.

The ACP MIPS processor module will be packaged as two connected boards that will fit in one standard VME slot. One board will contain the processors and cache; the other board will have main memory and the VME and VSB interfaces. Up to two extra memory boards may be plugged into other slots and connected via VSB to allow a total of 24 MBytes. The modular nature of this de-

sign means that as new components (denser memory or faster processors) become available, only one board at a time will need updating. With 4-Mbit DRAM expected in a year or so, 32 MBytes will be possible without an expansion board (or up to 96 MBytes with maximum expansion).

The new CPU module will provide high-level language processing power with a cost effectiveness of well under $200/VAX 780 equivalent, based on the physics benchmarks and features listed above. The FORTRAN compiler is the best we have encountered for a microprocessor. It supports VMS extensions and compares favorably with the VMS compiler in convenience and sophistication. Since the MIPS CPU chip has on-chip memory management, the board will be able to run the full UNIX operating system, booting either from a VME disk drive or using the Network File System (NFS) over the Branchbus. Full UNIX program development tools are available. This processor will form the cornerstone of the second-generation ACP systems.

New Interconnect Topologies

The original ACP multiprocessor used a single (MicroVAX) host which was the master of large numbers of microprocessor nodes. The ACP Branchbus was developed to link several high-performance commercial local bus crates (like VME) to a host and/or a data acquisition system. It is optimized for high-speed block transfers. No commercial alternative was available. The original Branchbus is a 32-bit bus connecting a single master (QBus, Unibus, or FASTBUS) to multiple crates (VME), with block transfers at up to 20 MBytes/sec. Improvements to the Branchbus system of interfaces allow higher performance and more complex interconnection schemes.

The new VME Branchbus Controller (VBBC) allows any VME master, in particular any node (or smart I/O device controller) in an ACP multiprocessor system to act as a Branchbus master and read or write to any Branchbus address. Any processor in the system can communicate with any other processor without host intervention, allowing the more elegant system architectures described below. Figure 2 shows a block diagram of a high-performance, offline ACP system using the VBBC. Here no specialized MicroVAX is required as a host; a standard node takes on that function, and an accompanying smart tape controller can transfer data directly from tape to any node in the system. The VBBC is a VME slave and Branchbus master. It is a shared resource in the VME crate, allocated on a first-come, first-served basis by a test-and-set bit in its control register. Once programmed with the Branchbus control words and address, the Branchbus cycles occur transparently to the VME master, which simply reads or writes data to the VBBC. First prototypes of the VBBC are now working.

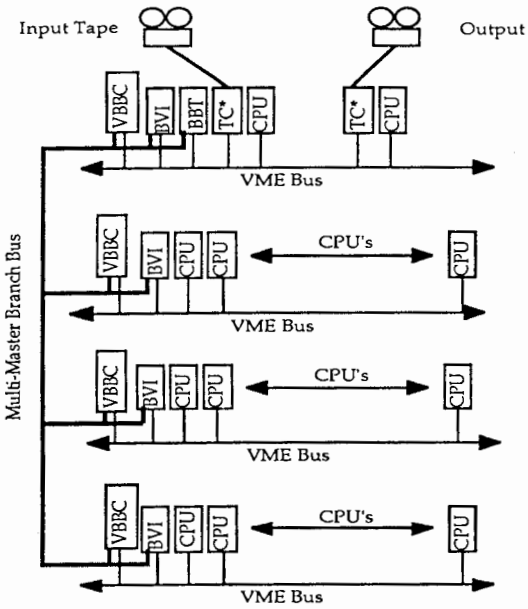

Fig. 2. A high-performance offline ACP system.

As implied above, the Branchbus now supports multiple masters. A distributed arbitration scheme similar to that used on the SCSI bus allows up to 16 masters to share the bus. Existing masters (Q/UBBC modules) can be used in multi-master systems by replacing their existing Branchbus Interface Daughter Board with a new Multi-Master Branchbus Interface Daughter Board which handles the arbitration transparently to the user.

The ACP Branchbus Switch allows full crossbar interconnection of up to 16 Branchbuses (or more using multiple switches). With this switch, any Branchbus master device can connect to any slave in the entire switch-connected system. All channels of the switch can be active simultaneously. For example, eight of the Branchbuses could be connected to the other eight, all transferring data simultaneously, giving an aggregate bandwidth of 8x20 MBytes/sec or 160 MBytes/sec (in addition to any local bus activity on any of the VME crates in the system). Thirteen Texas Instrument 16x16x4-bit crossbar chips (TI 74AS8840) are used for the main switching elements. The switch is a backplane in a 6U by 280 mm Eurocard crate. Modules may be plugged into the Switch Crate (see Fig. 3, page 10) much as with VME. However, instead of the signals being connected in a bus structure, each slot in the crate is a crossbar switch point. The first two Switch Crates are built and working.

Two modules now exist that plug into the switch crate. One is the Branchbus Switch Interface Board (BSIB) which converts the differential RS485 signals on the standard Branchbus cables to the single-ended TTL version of Branchbus used on the Switch backplane. The BSIB brings one standard Branchbus, with its VME crates, host, FASTBUS, etc., into a port of the Switch, and allows them to be switched to whatever else is plugged into the Switch Crate, such as other Branchbus circuits. Use of the Switch in this way will allow multiprocessor systems to obtain extremely high bandwidth for interprocessor communication. The second existing module that plugs into the Switch is the Floating Point Array Processor (FPAP), a 20-MFlop device that is used for theoretical physics (primarily lattice gauge) calculations.

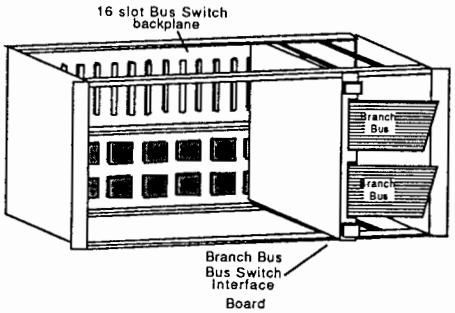

Fig. 3. The ACP Branchbus Switch. The backplane uses single-ended TTL Branchbus protocol.

Parallel Input/Output

Although it will continue to be possible to read and write tapes through a VAX or MicroVAX into an ACP system, high-performance operation will take advantage of I/O devices that interface directly to the multiprocessor system bus. Unlike Unibus and QBus tape drives, VME interfaces will give a bandwidth potential, depending on the I/O device, of up to the 20-30 MBytes/sec allowed by VME rather than the roughly 1 MByte/sec possible with minicomputer buses. One such VME tape interface, the Ciprico TM3000, has been used for data acquisition at Fermilab.

Parallel I/O is particularly encouraged by the availability of inexpensive, high-capacity, mass-storage devices which interface directly to VME. It is made possible by the availability of I/O directly in the node crates and by the new capability of a processor and/or its intelligent I/O controller to write through a VBBC from one crate to another. The multiprocessor system will have available to it many devices, all reading and writing simultaneously, allowing the total I/O bandwidth to be increased to whatever level is required.

New I/O devices are replacing standard 6250-bpi magnetic tape. These are very appealing to HEP experiments anticipating huge amounts of data, such as one approved Fermilab experiment which is planning to record tens of billions of events. At this time, video-tape cassettes appear most promising. The 8 mm format can pack well over 100 times the data in a given volume of shelf space as can conventional tapes, and they are at least an order of magnitude more cost effective. Current devices allow bandwidths comparable to those achieved with standard 6250-bpi tapes, with further improvements expected in the next year.

The ACP has tested the 8 mm video-tape device manufactured by Exabyte Corporation. Currently available devices store 2 GBytes (as much as 12 conventional tapes) on a standard $10 tape cassette. The drives cost less than $3000 and can deliver data at 250 KBytes/sec through a SCSI bus interface to VME or QBus. Both density and speed are expected to double in the next year.

The Exabyte drive package allows the design of a cheap and simple mechanical loader. Such a design is under way. It will incorporate a reader for optical bar code labels to insure mounting of the correct set of tapes from a large data sample.

Automating tape loads eliminates the time for human operators to mount tapes, which becomes significant when tapes are being processed at high speed.

The ACP plan at this writing is to support the Exabyte drives. However, it is important to emphasize that no standard has yet emerged in this rapidly developing field, and the long-term reliability of such devices has not been established. Nonetheless, it is clear that one can count on far better I/O performance than has been available.

Software System Architecture

The redesign of the system software for the Second-Generation ACP System will support the greatly improved performance and flexibility of the new processing, I/O, and communication hardware, while reducing the complexity encountered by both beginning and sophisticated users. It will allow existing applications to run with minimal changes, yet will provide a variety of powerful new features to allow users to realize the full potential of the new processors. Integration will be possible into a variety of computing situations, including large mainframe computer centers and the traditional VAX or MicroVAX host. Most important, in our view, is a distributed computing UNIX (or VMS) workstation environment in which the multiprocessor will function as a fully integrated back-end engine directly controlled from the workstations. The general hardware environment supported is shown in Fig. 4. Because of the emphasis on portable, nearly universal standards, it will be straightforward to incorporate any computer that runs UNIX and/or can communicate via TCP/IP over Ethernet. This makes the system open and receptive to future requirements and product options.

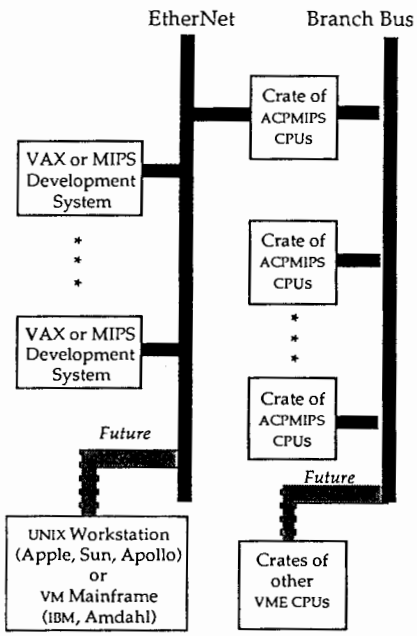

Fig. 4. *The Second-Generation ACP System: general hardware and network environment.*

As before, the ACP System Software is a tool to make it easy for physicists to bring their programs to a high-performance multiprocessor environment. An application for the second-generation system is decom-

posed into a set of cooperating processes. Support for running these sets of processes as a job, including interprocess communication and synchronization, startup, etc., is provided by the ACP System Software. The processes run on an ACP Multiprocessor System interconnected by Branchbus and on any associated UNIX or VMS (later, possibly, also VM) computers connected via Ethernet and TCP/IP. They may be distributed as appropriate over the available computers. In this way, a multi-process job may be tested first in a single machine and then with increasing numbers of nodes. Program development is done using the full set of UNIX (or VMS) tools, including compilers, linkers, and debuggers, of the computer on which the process will run.

Any node process can assume the functions previously exercised only by the VAX host processor. Any node process in the system can do I/O, reading or writing data tapes and accessing disk files. And any node process in the system can do "send" or "get" operations to or from an individual process (chosen by the system software from a class or rank of node processes) or set of processes in a given class or rank. As before, the system software will automatically find an available node process for the user. The ability to send and get to multiple nodes in a class allows "broadcast"- and "accumulate"-type operations.

Along with the traditional `send` and `get` type of ACP communication routines, there will be a variety of more primitive, yet powerful and easy to use, interprocess communication mechanisms. A process may send a block of data directly to a block of virtual address space in another process, or it may call a subroutine in another process (remote subroutine call), or it may send a small data packet (a message) to another process. Users of the new system will have direct access to process queues which they may define as they require or use in standard, traditional ACP-defined ways (like node process *ready* or *complete*). Synch points provide a way for all processes in a class to synchronize program execution.

There are many possible process configurations that the new system can support. An example for a reconstruction problem with multiple input tapes is shown in Fig. 5. Note that this is a software configuration; the actual hardware connection of the nodes is over Branchbus via VBBCS and Bus Switches (if necessary) and is transparent to the programmer. Nodes in the top rank read events from data tapes and pass them along to either class 2 or class 3 nodes, which process events of different trigger types. Nodes in the bottom rank collect events from any nodes in the middle rank, either class 2 or class 3, for output to tape.

Another important configuration has enormous implications for physics data analysis (as opposed to reconstruction). Here, the top rank of nodes is the same

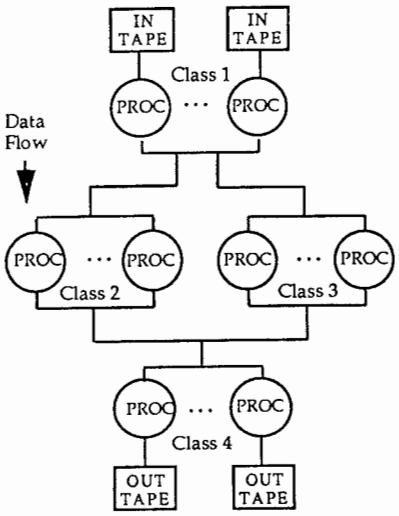

Fig. 5. A multi-rank configuration. Processes are indicated by circles.

as in Fig. 5, and the second rank consists only of one or more workstations and, perhaps, a single data-recording process. With such a configuration, a whole experiment's data base of data summary tapes (DSTS) can be read and analyzed in parallel in much less than an hour. The traditional means of passing hundreds of DSTS through a computer center for a physics analysis pass often takes weeks.

Conclusion

To summarize, the Second-Generation ACP System architecture is designed to support large increases in processing power and in I/O and communication bandwidth. It provides a set of hardware and software building blocks so that a system can be matched to the set of applications it will run. One can provide enough I/O devices and Branchbus interconnects so there are no bandwidth limitations. The number of basic CPU nodes is determined by processing power requirements, while special computer nodes, such as workstations or mainframes, can be incorporated into the Ethernet side of the system when needed and available. As each job is run on the system, nodes are assigned to run particular user processes (input, output, event processing, etc.) as appropriate. Not only the traditional compute-bound event reconstruction, but also more I/O-intensive data analysis will find a home on these systems.