Fermi National Accelerator Laboratory

FERMILAB-Conf-88/117

Computing Possibilities in the Mid 1990s*

Thomas Nash
Advanced Computer Program
Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

September 1988

*Talk given at Future Directions in Detector R&D for Experiments at pp Colliders, Snowmass, Colorado, July 5-7, 1988.

Operated by Universities Research Association Inc. under contract with the United States Department of Energy

COMPUTING POSSIBILITIES IN THE MID 1990S*

Thomas Nash

Advanced Computer Program
Fermi National Accelerator Laboratory
Batavia, IL 60510 USA

* Talk at Future Directions in Detector R & D for Experiments at pp Colliders, Snowmass, Colorado, July 5-7, 1988.

COMPUTING POSSIBILITIES IN THE MID 1990S*

Thomas Nash

Advanced Computer Program
Fermi National Accelerator Laboratory†
Batavia, IL 60510 USA

ABSTRACT

This paper describes the kind of computing resources it may be possible to make available for experiments in high energy physics in the mid and late 1990s. We outline some of the work going on today, particularly at Fermilab's Advanced Computer Program, that projects to the future. We attempt to define areas in which coordinated R&D efforts should prove fruitful to provide for on and off-line computing in the SSC era. Because of extraordinary components anticipated from industry, we can be optimistic even to the level of predicting million VAX equivalent on-line multiprocessor/data acquisition systems for SSC detectors. Managing this scale of computing will require a new approach to large hardware and software systems.

INTRODUCTION

In the mid 1990s, the complexity of high energy physics experiments will have reached the point that just dealing with the complexity itself will have become an important challenge.[1] Managing extremely complex systems will not just be an academic concern. The issue will be felt acutely by everyone involved with SSC detectors: physicists, lab and DOE managers, and even high tech aerospace corporations offering management solutions and tools.

The huge electronics, computer, and software requirements of SSC era experiments, both on and off line, are what concerns us here. There are several areas: detectors and digitizing readout; trigger physics algorithms; on-line data acquisition and processing systems; off line reconstruction software; and physics analysis. These are at the heart and soul of experimental activity. Yet, the scale of these systems, complex in the number, variety, and sophistication of their components, will not be amenable to conventional management techniques, certainly not the anarchistic, "we'll get to that later", approach traditional on HEP experiments.[2]

In the mid 90s we no longer will have the luxury to tolerate the extra complexity caused by the institutionally compartmentalized, bottom up, design which is now typical, encouraged, if not mandated, by the sociology, politics, and personality of our field. The fact that it may prove possible to operate large present day experiments, handicapped as they are by excess complexity, does not mean that this will continue to be possible for SSC/LHC era detectors. It will be essential to streamline everything, without compromising necessary physics flexibility. In the jargon of modern software engineering, this means *structured* yet flexible systems.

The answers are far from all in. More R&D work aimed at the systems strategy for dealing with large HEP data acquisition and computing efforts is certainly in order. Nonetheless, in the spirit of this meeting, one can suggest how to move in the right direction. We will need to develop a standard processing environment suitable for most off-line and on-line computing. We need to steer away from hard wired triggers and difficult to program, though fast, special devices often used for so called *second level* triggers. We need to use programmable and verifiable processors for all but the first level of triggers. This implies high level languages and something like structured analysis, structured design (SASD), which we must make work for both hardware and software. We will need to make extensive use of expert systems for real time diagnosis of problems in big data acquisition systems, large group software packages, and all other large subsystems of the experiment. And we need excellent human interfaces on workstations to allow effective development of software, debugging of hardware, and monitoring of systems.

* Talk at *Future Directions in Detector R&D for Experiments at pp Colliders*, Snowmass, Colorado, July 5-7, 1988.

† Fermilab is operated by Universities Research Assoc., Inc. under contract with the U.S. Departmentof Energy.

Reasonable successes by HEP groups working on computing problems, now and in the past, encourages some confidence about attaining these future goals. High energy physicists, like most scientists, have always wanted more computer power than they could afford to buy.[3] 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 5 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.

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.[4] 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 CPUs.[5] Such systems are now an acknowledged important component of computing in high energy physics. First generation ACP systems are now at over 30 installations in universities and laboratories worldwide, primarily, though not exclusively, for HEP applications. The first of these systems which provide CPU power at a cost of less than $2500 per VAX equivalent, were brought online two years ago. New work at the ACP is now mainly in two areas[6]: a second generation multiprocessor targeted at experiments, and a multi array processor "supercomputer" for the site oriented problems of theoretical physics, principally lattice gauge calculations.

In this paper I will first briefly describe the new ACP systems for experimental HEP, in order to introduce the kind of powerful yet structured computing capabilities that should be attainable by the middle of the next decade. Early in the next year, these 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.

The new RISC processors that will be used by the ACP are on such a rapid growth curve, that it is not unreasonable to anticipate access to processing systems online of a million VAX 11/780 power in easy to program form. This makes it possible to avoid extensive use of difficult to program, elaborate special purpose processors. This is key, as suggested earlier, to streamlining on-line systems. We will discuss some of the main hardware and software development issues surrounding these huge increases in processing power as they affect system and module design and user applications. The ACP multi array processor system for theory[7] will be briefly mentioned in the context of how its point to point switch architecture is relevant for future generations of experimental systems with extremely high performance processor nodes.

Industry is certainly providing us the opportunity, through the extraordinary array of new components being made available, to put together coherent systems that will meet the needs for SSC era computing *and* which will be manageable. At the end of this paper, there is a proposed list of projects that form a part of a coherent R&D program that should move us far in the direction we need to go for SSC experiment computing. The list is probably not complete, nor particularly well refined, but it is intended as a talking point so we can decide where to start, and get on with the job.

THE 2ND GENERATION ACP MULTIPROCESSOR

The continued saturation of computers (including ACP systems) by HEP experimenters motivates the development of a second generation of the ACP Multiprocessor system. This has been described in detail recently elsewhere.[6] A variety of new and increasingly powerful microprocessors are now available to incorporate into multiprocessors. Most of these are based on the Reduced Instruction Set Computer (RISC) philosophy. It was realized 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 will offer as much as a factor of twenty, more performance than the first generation ACP processors based on 68020s. Unlike earlier processors, some have usable FORTRAN and C compilers and UNIX operating systems even before the hardware appears. The broad availability of UNIX is particularly important. 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 are designed to be as transparent as possible to users of the original system.

As it has in the past, the ACP is encouraging competition in SBCs 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. The ultimate standard is how real physics code runs in a high level language, since it does not matter how many million instructions per second (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. Based on these measurements, we expect HEP code to run at 12-15 VAXes on 25 MHz R3000 boards and 16-20 VAXes on the 33 MHz versions we plan to use.

The new CPU module will provide high level language processing power with a cost effectiveness of well under $200/VAX 780 equivalent. 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.

The original ACP multiprocessor used a single (MicroVAX) host which was the master of large numbers of microprocessor nodes.[5] 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 (20 MBytes/sec) block transfers.[8] Improvements to the Branchbus system of interfaces allow higher performance and more complex interconnection schemes. Any VME master, in particular any node or smart I/O device controller in the system can now communicate with any other processor without host intervention, allowing the more elegant system architectures described later.

The new 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

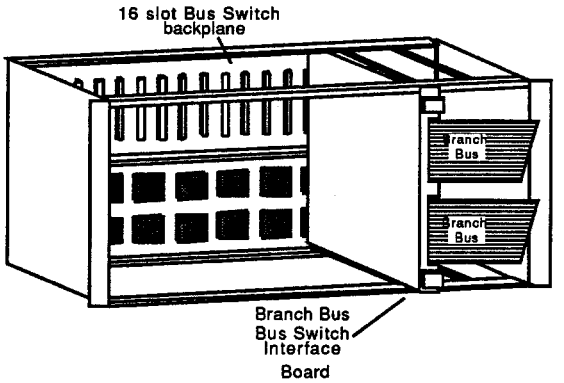

Figure 1. The ACP Branchbus Switch. The backplane uses single ended TTL Branchbus protocol.

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 8 X 20 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 16 X 16 X 4 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 Figure 1) as with VME. However, instead of the signals being connected to a bus, each slot in the crate is a crossbar switch point. The first two Switch Crates are now in operation.

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 (peak) device that is used for theoretical physics, primarily lattice gauge, calculations. The FPAP is described elsewhere in detail, as is the innovative "better than a hypercube" architecture that the Switch allows.[7] Important future applications of the versatile Switch are described in the next section.

It will continue to be possible to read and write tapes through a VAX or MicroVAX into an ACP system. High performance operation, however, will take advantage of the potentially higher bandwidth of high capacity mass storage I/O devices that interface directly to the multiprocessor VME crate bus. The low cost of these devices particularly encourages parallel I/O. This 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 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 magnetic tape. These are very appealing to HEP experiments anticipating huge amounts of data, such as one approved at Fermilab 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 the media is at least an order of magnitude more cost effective. Current versions allow bandwidths comparable to those achieved with standard 6250 bpi tapes, with improvements expected. It is clear that with devices like these, one can count on far better I/O performance than has been available.

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.[9] It will allow existing applications to run with minimal changes, yet will provide a variety of powerful new features so that users can 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 Figure 2. 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.

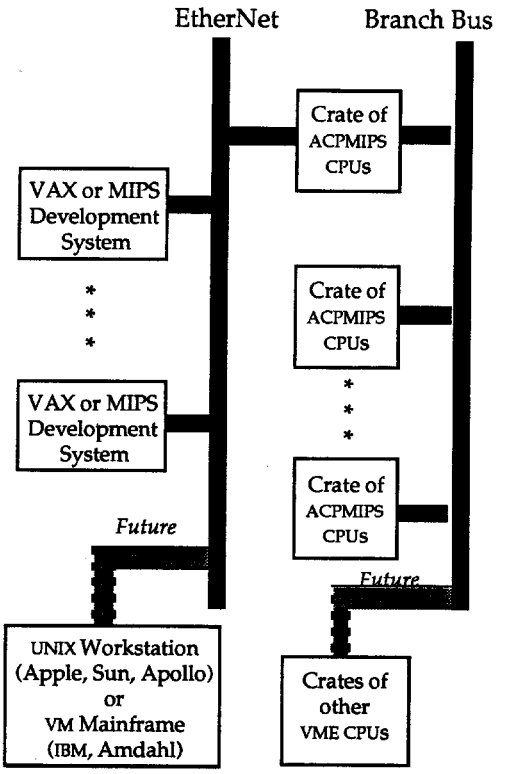

Figure 2. Second Generation ACP System: General hardware and network environment.

The ACP System Software is a tool to make it easy for physicists to bring programs to a high performance multiprocessor environment. An application is decomposed 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, including 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 processes 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

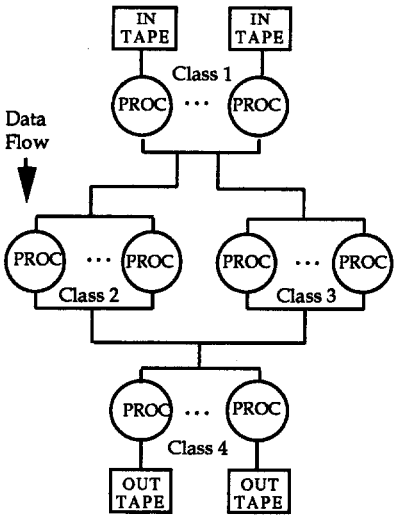

Figure 3. A multi rank configuration. Processes are indicated by circles.

shown in Figure 3. Note that this is a software configuration; the actual hardware connection of the nodes is over Branchbus including 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 as in Figure 3, and the second rank consists only of 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 analyzed in parallel in much less than an hour. Traditional means of passing hundreds of DSTs through a computer center for a physics analysis pass often take weeks.

THE NEW WORLD OF RISC AND ITS IMPLICATIONS

The big jump in microprocessor power available in the last year or two has been a pleasant surprise, allowing a 20X increase in per node ACP multiprocessor performance. This has been the result of the new technology of reduced instruction set computers (RISC) we described earlier. Even this impressive *rate* of increase is now projected to continue well into the 1990s. For example, one of the three suppliers of MIPS RISC processor silicon, Integrated Device Technology, Inc. (IDT), is promising a 160 VAX Mips version in 1991. MIPS, Inc.'s own projections are indicated by the chart in Figure 4.

Although by industry standards these projections are conservative, high energy physics reconstruction code tends to run at about 25% below these numbers.

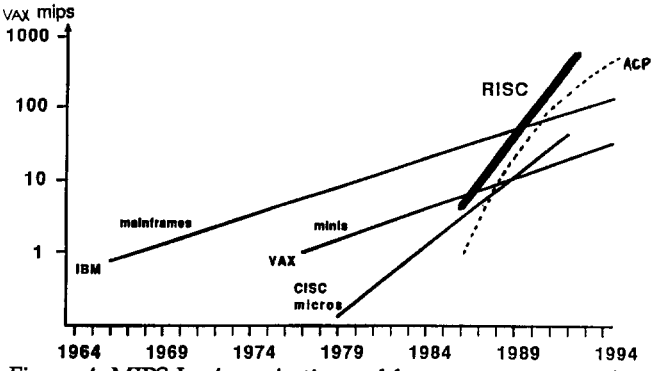

Figure 4. MIPS Inc.'s projections of future computer performance. Our projection of ACP style SBCs are added as a dashed line.

There is also an obvious delay from product announcement to when a group like the ACP can have usable processor modules available. Nonetheless, as indicated by the dashed curve we have added to Figure 4, ACP style SBC modules at a 500 VAX Mips level could be anticipated in 1995. It is important to emphasize that there are many companies playing in the RISC component field, including the semiconductor giant Motorola, and even IBM. The products from MIPS Inc. may well continue to be the most appropriate for our purposes, but this cannot be guaranteed. However, it seems likely, subject to the usual *caveat* about peering into crystal balls, that this technology, and the competition of industry, will produce RISC processors at and beyond 500 VAXes.

Where is all this performance coming from? We described earlier the philosophy behind RISC. The comparison with complex instruction set computers (CISC) is now well understood. Streamlining the instruction set can improve (reduce) the number of clocks per instruction by factors of 5 or more. Clock speeds can be maintained or even increased compared to CISC architectures. The cost of this simplification is an increase in the number of instructions to be executed, because complex instructions and functions are handled in software. RISC

wins because this increase in instruction count is small.

Over the past few years dramatic increases in memory size have been more common than in processor speed. Although memory improvements are continuing, it is becoming clear that this is now happening at a slower rate than CPU performance -- and slower than the growing appetite for memory of high energy physicists, now acculturated to assuming endless increases in memory availability. Present generation ACP CPUs are based on 256K DRAMs. The second generation will use 1 Mbit. The conventional wisdom in the trade press is that 4 Mbit chips will make their appearance in 1989 and become a significant part of the market in 1991. 16 Mbit parts will appear in 1991 and we may guess that they will be readily available for projected 500 VAX performance ACP type module in 1995. This would correspond to 128 Mbytes per node. Not bad. But the ratio to processing power will be reduced somewhat below the first and second generation ACP nodes.

The processor and memory technology environment is changed. Experimentalists will no longer have their memory demands easily satiated while they hunger desperately for computer cycles. It will be important to take advantage of incredible micro computer performance levels and cool it on memory profligacy.

The implications of this new environment for HEP computer designers will also be stimulating, to say the least. With 33 MHz RISC processors, we are already in a cache crisis. 20 nsec static memory is too slow and it is a struggle to find acceptably fast and large parts. Up to now lattice gauge processors[10,7] have avoided cache by matching DRAM speeds to the relatively slow floating point chip clocks. The next time around cache will be necessary, but theoretical problems make such regular accesses to memory that their cache miss rates are unacceptably high in standard cache.

Industry is sensitive to both aspects of the cache crisis and seems to be moving to multi level cache. If this is not satisfactory for theorists, it may be necessary to use *anticipatory* cache which requires hardware and software to request memory in advance. In principle, this is not hard to accomplish for theoretical problems. In practice, supporting this at the compiler level would be a big effort.

The second technical crisis resulting from the new processor environment is the obsolescence of computer busses for multi processor communication. Never mind the territorial religious wars about busses that have swept all HEP laboratories. No matter if we have crusaded for Fastbus, VME, Multibus, Nubus, or a homebrew, your or my favorite bus cannot handle the communication requirements of 100 VAX power processor nodes.

Even for event oriented reconstruction and trigger processing, just moving data in and out will saturate local crate busses.

The answer to the bus crisis is point to point communication between processors. Two variations on this solution are now being explored in the high energy physics community. There is considerable interest, strongest in Europe, in the INMOS Transputer architecture. These microprocessors incorporate several sophisticated communication channel ports on each chip that support direct links to neighboring Transputers. An example of how Transputers can be used in a point to point architecture is seen in the structure of the Global

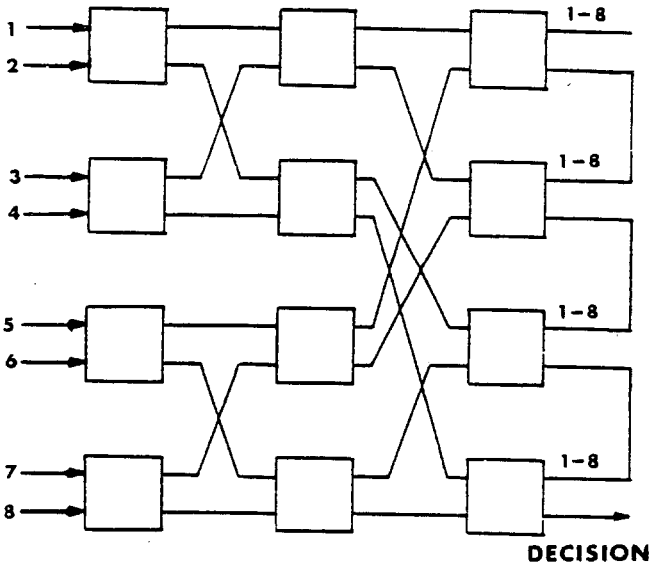

Figure 5. ZEUS 2nd level trigger box.

Second Level Trigger Box for ZEUS (Figure 5).[11] The architecture of the Transputer is certainly very appealing. However, no other semiconductor company besides INMOS has yet picked up on it. The processing performance of Transputers has lagged by over an order of magnitude behind the leading microprocessors at any given time, probably because design emphasis has been on the communication channels. Symptomatic is the fact that the FORTRAN compilers are still not very robust. This situation may change in the future as other manufacturers recognize the limitations of conventional communication mechanisms for their super fast processors and, possibly, pick up on the INMOS approach.

The ACP's Branchbus Switch Crate described earlier (Figure 1) represents an approach to point to point communication that does not depend on a particular processor's specialized communication ports. It thus allows a free choice of the highest performing processor at any given time. As already noted there are now two module types that plug into this crate, a Branchbus interface and the floating point array processor (FPAP) for theoretical calculations. It is very likely that the next version (1990) of an ACP processing node for experimentalists will also live in the Switch Crate so that there will be enough bandwidth to move data in and out.

The Switch Crate looks very much like a VME crate, using the same Eurocard hardware and having the same 6U height. It has all the advantages of a modular crate system and, with its cross bar switch back plane that allows up to 8 independent direct slot to slot connections, it does not have the bandwidth limitations of a bus. One essential characteristic of a commercial bus like VME is the wide spread availability in the standard of the latest in computer devices, such as I/O controllers and network interfaces. To make these available to the otherwise superior Branchbus Switch Crate environment, we expect to develop an interface to VME, using the extra depth of the Switch Crate. Any VME module could be plugged into the short interface card which in

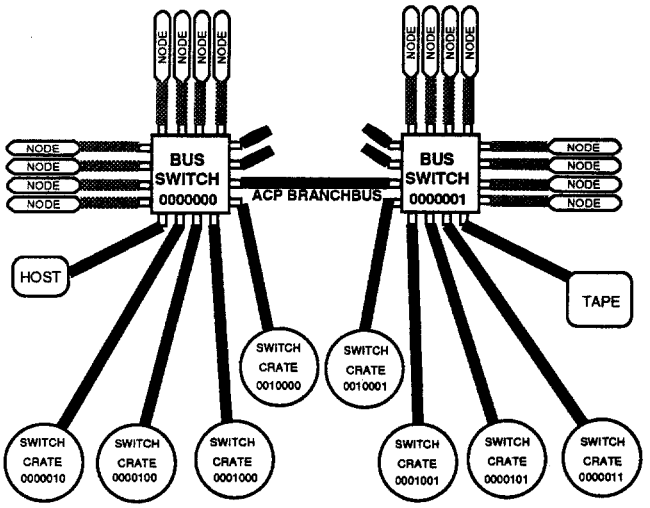

Figure 6. ACP Multi Array Processor System: 256 node configuration.

turn is plugged into the Switch Crate.

In order to display the versatility of this approach, I can't resist describing briefly the ACP Multi Array Processor architecture designed for theorists[7]. Figure 6 shows the 256 node configuration presently under construction. The individual single board FPAPs have peak performance of 20 MFlops. Performance of key kernels (SU(3) multiplies) have been measured on the prototypes to exceed 15 MFlops/node. The FPAPs are plugged into the Branchbus Switch Crates. The nodes can speak with each other in pairs at a full 20 MBytes/sec simultaneously. The architecture is a hypercube network of such crossbar Switch Crates each supporting 8-16 FPAPs. In a typical configuration 8 array processor nodes will be plugged into each switch crate along with up to 8 BSIB I/O modules (described earlier) that interconnect crates, via standard ACP Branchbus, in a hypercube (or better, if extra interconnects are desired).

Processing nodes do not participate in any communication activity other than their own. This is an important distinction from traditional hypercube implementations. The switches handle intra and intercrate routing automatically. The system therefore does not operate with all node programs (and/or communications) in lock step like an SIMD machine, as is the case in most of the other projects of this type[10]. It also does not strongly favor local communication (as existing hypercubes[12] do). It thus allows for any conceivable new lattice algorithm unconstrained by synchronous or local communication requirements. Despite its algorithmic flexibility the system ranks as the best (or nearly so, we won't argue) in terms of cost effectiveness of MFlops/$.

The first 16 node system is being assembled this summer. All components are working and tested. They have successfully run extensive physics code. Parts are being procured for a 256 node (5 GFlop for about $1 million) system which will be assembled at the end of the year. Maximum system size is 2048 nodes. The system is being designed in the ACP tradition to be commercialized and available to other institutions.

SSC TRIGGER AND DATA ACQUISITION

The key problems of SSC detector data acquisition and triggering are widely appreciated: the luminosity, the trigger reduction of 10^8, and the overlapping events in the detector. If the projections discussed earlier of 500 VAX equivalent processor nodes are realized in 1995, there is hope. A single module CPU, no matter what its capability, will always cost in the neighborhood of $2500. Given a willingness to spend $5 million (of the over $100 million cost of an SSC detector) for real time processing, a million VAX multiprocessor on-line is not only imaginable but likely.

The important thing about this megaVAX system is that it is all programmable in high level languages and, most important, it is accessible to modern software engineering tools. As was emphasized in the introduction, such tools, and a structured, homogeneous environment to work in, are nothing short of essential if we are to cope with the complex scale of SSC triggers and data acquisition. The same processors and programming environment would be available to off line computing.

Given the opportunity and advantages of such a coherent and powerful system, how much of the data acquisition and trigger reduction task can one imagine handling in this way? This question requires study, but it is not unreasonable to hypothesize that a large fraction of the trigger reduction and essentially all of the standard data formatting could be accommodated.

The vision is of on-detector first level triggers based primarily on customized VLSI components and taking on a trigger reduction of 10 - 1000. The remaining 10^{5-7} would be written in structured high level languages running on a multiprocessor, probably configured as a

tree structure. Each branch of the tree would handle the data formatting and second level trigger tasks of a particular sub detector. Unlike traditional second level triggers based on special purpose hardware and software, these would be easy to program and verify.

The opportunity to see structured, and manageable, first level triggers is also likely to be realized. This hope is based on the interesting development work now going on in several areas. At this workshop, Nygren and Shapiro described two different LBL projects that are moving rapidly toward pixel silicon detectors. These will ultimately incorporate on chip or hybrid (bump bonded) digital circuitry. Such circuits will take care of readout and data formatting, and will offer a fertile field for development of first level triggers.

There are two directions that such on-detector first level triggers may take (both for silicon and for other detectors). Perhaps futuristic, but certainly promising, is the hot new area of neural networks. Bruce Denby has described some early success in applying this to reconstruction problems.[13] Neural networks are patterned after the way the brain's system of neurons and synapses works by relaxing to a minimum energy state corresponding to recognition of a pattern. They are particularly appropriate for VLSI implementation and neural network chips are already being fabricated by Carver Mead at Cal Tech and L.D. Jackel at Bell Labs. Perhaps 1995 is a bit too optimistic, but one can easily imagine neural net readouts directly in terms of tracks and calorimeter clusters.

The other very strong candidate for implementation in VLSI and on detector triggering is the data driven processing system approach, now available as PC board modules in two HEP variations.[14] At this workshop Rudi Bock described his interest at CERN in identifying components of a similar kind from the signal processing world. These systems are programmed by how modules, each carrying out a specific computing operation, are interconnected. When one module completes a task it puts its result, and a ready signal, onto its output cable. As soon as a following module has all its incoming data, indicated by all ready signals being on, it carries out its operation. The operations range from simple arithmetic to complex track finding and table lookups. It will be interesting to see how such a system when available in VLSI modules can be coupled to neural network chips directly on a detector.

TOWARD A STRUCTURED ON-LINE ENVIRONMENT

One need only look at the data acquisition system of a present day large detector like CDF, and imagine it scaled up by an order of magnitude, to appreciate what we are getting into with SSC detectors. On-line systems coherence is far from being prevalent these days on experiments. Some of us have seen what happens when smaller experiments attempt to retrofit modern processors into existing electronics. Sergio Conetti at Trieste[15] described the extraordinary variety of ways in which ACP processors have been incorporated into experiments. Although widely used in standard configurations for off-line, there is no standard on-line implementation of these systems because there is nothing approaching a standard data acquisition system. Every experiment does its own thing.

In present day large experiments, the sociology of multi institution collaborations works against data acquisition system coherence. The typical scenario at design report time is to divide up responsibilities among the various groups allowing each to define the approach to be used in specific subsystems. A few years later the problem of interfacing these subsystems becomes a hot topic. The situation is, to some extent, due to the way approval committees operate. Expertise is compartmentalized by subdetector and institution. The overall system ends up being based on disparate subsystems, developed on the basis of correct, strongly held views, that don't mesh as a system.

Most experiments attempt to standardize at some level. Nonetheless, many still have systems that incorporate both Fastbus and VME, or otherwise mix standards. A similar situation exists with software standards. Much of this results from a plague of what might be called *standards evangelism* in our business. Individuals who have developed some degree of personal expertise in something like VME or Transputers or UNIX or VMS or Fastbus or ACP systems become strong advocates. At some level, this is an understandable tendency to protect an individual's intellectual investment in the expertise. However, the increasing complexity of detectors cries out for unbiased attitudes, a *secular humanist* approach, toward standards. We need to cool off the all too common electronics and computing standards religious wars which every laboratory has encountered.

The complexity of the behemoth detectors of the SSC/LHC era will require nothing short of a new and structured approach to on-line data acquisition and processor systems. Although there is much talk about code management for large off line programs, it is pleasing that the first extensive application of serious software engineering tools, like structured analysis structured design (SASD), are happening in the on-line data acquisition world. The ALEPH on-line electronics system is a prime example of this. At Fermilab, DO is committed to SASD tools for on-line software. This is also the case at ZEUS. At OPAL and HERA's H1 on-line system management is being addressed through advanced human interfacing techniques taking advantage of Macintosh Hypercard stacks.

The present state of commercially available software engineering tools leaves much to be desired, as anyone who has seen them in use can tell you. There are a variety of packages available. Each of them has some desirable features and lacks others. The ability to automatically generate code from SASD bubbles, even just data defining statements and common blocks, is apparently not yet available, at least not to physicists. One suspects that certain large software development organizations (IBM, DEC, AT&T) are holding some pretty powerful tools close to their corporate chests. Physicists will have to pioneer in this area, encouraging vendors, acting as beta test sites, and being willing to spend money for software and workstations that may be discarded on relatively short time cycles.

Structured design tools and methodologies are advocated because they lead to software that has a dramatically reduced density of errors and which is easy to read and understand so that changes can be made with some hope they will work. Such structured code is also more readily testable, an issue we will return to later. It should be possible to apply the same philosophy to large hardware system design. Jon Thaler at the University of Illinois advocates extending PC board CAE/CAD tools to systems of modules.

At the very least we need to structure and modularize our large systems in as homogeneous a way as possible. The ALEPH data acquisition system is a particularly good example of an existing well structured system. LEP experiments do not require large amounts of processing power for triggers, so ALEPH has not had to deal with integrating large numbers of high level language processors smoothly into a data acquisition system. For some reason, the traditional way to do this is through an event builder. Sub events come from the many (usually different) data acquisition electronics sub systems, each handling a sub detector, to a single one of these very elaborate boxes. It assembles the event and passes it on to the high level trigger.

The idea of assembling events directly into the memory of the processor seems so obvious. It has not been used because, in most cases, experiments treat the high level processor with trepidation and lack the confidence to incorporate it within the system, where it belongs. Trigger computers and data formatting and calibration processors should be integrated into a single system. Cutts and Zeller at DO have demonstrated that a system can be designed without an event builder. Eight data cables bring data from separate readout systems directly into special multi port memory that sits on the Q Bus of a MicroVAX in the high level trigger farm. White, Barsotti, *et al.* in the FUSE project proposal at Fermilab have a similar two dimensional architecture approach. The system is extendible in one direction to allow more sub detector read out controllers and in the other to allow a hierarchy of processing.

Using the Second Generation ACP software and hardware described earlier, one can picture a data acquisition system in which data is read out of VME digitizers and formatted by the same type processors as used in the high level trigger. Data is routed from the formatting processors to the trigger processors to data logging processors and even to monitoring work stations over Branchbus through one or more Branchbus Switch Crates at 20 MBytes/sec per channel. The software environment for all the computing activities is identical and modules are interchangeable throughout. The amount of intellectual effort required to understand and maintain such a homogeneous system is much less than has traditionally been required.

To us, this kind of system clearly points toward where we think we need to get for the far larger SSC data acquisition systems. What development areas should we be addressing? As noted earlier, future processors, and digitizers, should be in the Switch Crates themselves (or a successor technology) to handle the higher rates required by super fast processors. But how do we deal with the 2000 or more CPUs, and the associated digitizers, which probably will be arranged in a tree structure corresponding to the subdetector organization? How would one bring up such a monster when the different pieces of the hardware and software are in varying states of readiness, and robustness? We need a system skeleton that is easy to assemble in the sense that if a number of working subsystems are connected, the combination will work also, automatically. Or, equivalently, if a subsystem is removed, the remainder should continue to operate subject only to functional limitations caused by the missing piece.

The automatic routing ROM in the Branchbus Switch suggests a research direction that may well allow us to accomplish this very important requirement that a system skeleton be easy to assemble. One could imagine a smart switch router that looks at directives accompanying the data and replacing specific addresses. The directives would indicate the kind of processor to which the data should be routed (or from which data should be pulled). Such classifications are available now in software to users of ACP systems who need only request any free node, or any node that has completed a task, or all nodes in some class, or the same node as the last operation, etc. By handling this in hardware rather than software, it appears that a system would be transparently operable with or without all its subsystems. It would be easy to assemble in our way of speaking. This is not going to be an easy development, but it should not be thought impossible. Bruce Knapp has pointed out that this is similar to the data driven processor idea we described earlier for low level triggers. Only here, the modules are high powered, high level language programmed, CPUs.

In addition to the structuring issues described above, one should take advantage of higher speed data transmission and switching technologies, such as serial optical fibers, that are likely to prove very attractive in the near future. Here, certainly, is an important and interesting development opportunity: developing an appropriate skeleton for SSC data acquisition digitizing and processing.

AI AND HIGH ENERGY PHYSICS

Very little has been done with "AI" in high energy physics. That's not because the issues that Artificial Intelligence researchers work on are not relevant to us. In fact, one can argue that almost all high energy physics computing (the exception is theory) is of the non-numerical kind that AI is really all about.[1] The difficulty is these problems are so hard that progress is slow and practical applications limited. In a number of critical areas, HEP's needs for practical AI type solutions are so acute that we will have to get involved in the effort.

I have argued that the way in which to deal with large detector electronics systems and big trigger reductions is to move as much specialized activity as possible into high level language processors. This implies huge programs, and it is probably partially the case that people have avoided this approach because of fear of errors in the software. Rationally, for a given trigger and amount of effort there is no reason to expect less errors in a complex special trigger processor. Except, perhaps, because the expertise required limits access to a few, presumably very competent, individuals. This is in distinction to software where "everyone" can get their hands in. We can no longer afford such thinking. We will have to manage access along with everything else. But given the huge size that trigger programs will reach (CDF's offline code is now a million-lines), we will have to learn how to establish their correctness.

There are two aspects of determining correctness. The first is, in principle, straight forward. It is the question of whether the program does what it has been specified to do. This is where computer aided software engineering (CASE) tools, like SASD, become particularly important. It is likely they will form the platform for more sophisticated program verification tools. To take advantage of (and influence the development of) future CASE tools, we need to get heavily involved with what is presently available.

Though hardly solved, software verification is the easy part. The hard part is how to know whether the specification itself is correct. Everyone is familiar with how mistakes in understanding or definition or whatever, mistakes in the specification, cause major delays in developing off-line reconstruction and analysis code. This business of validating software specifications is a particularly advanced artificial intelligence issue because it involves the whole concept of a "common sense data base".

Both verification and validation are most acute in the real time situation (as many of us have pointed out in the SDI context). On experiments, it must be possible to change triggers as new information is learned. Even if such changes will no longer be tolerated on night shift, they will continue to happen on short time scales, and we have to know that they were both conceived and implemented correctly. While we wait for 21st century AI tools, we will have to discipline ourselves to maintain well managed, well structured programs and take advantage of whatever software engineering we can get.

AI is closer to providing practical solutions in some other areas we need, expert systems and human interfaces. Expert systems are already in use on CDF to help ordinary users maintain the Fastbus network during normal data acquisition. Carroll and Booth have provided a means for locating a faulty module without having to locate the university based expert at home 2000 miles away in the middle of the night. Such systems will become widely used. They are very quick at disgorging expert information, but very slow at absorbing it. High energy physics has particularly dynamic requirements on expert data bases, as systems change rapidly. Because of this somewhat unique requirement, we may have to get involved in improving the information input aspects of expert systems.

Physicists have taken to the Macintosh with amazing speed and unanimity despite the fact that these personal workstations don't really do very much specialized for us yet. This is a clear example of the importance of good human interfaces. When analysis processors soon are available that can turn around a large experiment's DST data base in an hour or less, it will no longer be acceptable for physicists to spend days preparing their next batch of histogram subroutine call statements. At CERN, René Brun has developed the Physics Analysis Workstation (PAW) graphic front end to CERNLIB histogram and plotting software on Apollo and other workstations.

The ACP also has underway a project to develop particularly efficient work station tools for doing analysis. Macintosh like human interfaces, adjusted to physicist needs and abilities, will be used on Apple or, perhaps, Sun or other workstations. Dennis Hall and colleagues at LBL have proposed an Accelerator Designer's Workbench. A Mac would front end a Cray or other super processor. On the Mac screen users would move magnet components around to prepare for a new run of a transport calculation. They are also proposing development of a "software bus" that all such front ends, as well as CAD and project management tools, could plug into.

Most important in these projects is a top down design. "Designer screens" must be effectively prepared by specialists. Physicists are not too good to have information communicated to them efficiently.

A FIRST CUT AT A COHERENT R&D PROGRAM

To conclude, I list the key definable R&D projects suggested by the discussions in the preceding sections. The list is not necessarily complete or properly partitioned but is meant as a starting point in defining a program of R&D for SSC experiment computing needs. Included are guesses at appropriate manpower and annual non salary costs (averaged over 5 years).

- Low level on-detector trigger system, based on VLSI, perhaps neural networks and/or data driven units. 10 persons, $300K.
- On-line data acquisition/multiprocessor system skeleton, pipelined, structured, and extensible (auto routing?). 7 persons, $200K.
- CPUs, multiprocessor software, and development environment commonly useful for on-line triggers and data formatting and off-line reconstruction and analysis. 10 persons, $200K.
- Computer aided software engineering tools for high energy physics. Standard, usable, across the board program development, verification, and validation. 5 persons interacting closely with commercial CASE development, $500K.
- Expert systems for data acquisition, hardware and software testing and diagnosis. 5 persons, $150K.
- Analysis workstation environment tools. 5 persons, $75K.
- Data management, code and data file service tools. 5 persons interacting closely with industry, $250K.

These projects are likely to be very rewarding. At the same time as they represent tremendous leaps in technology, one can enter into them with confidence that there are excellent chances for success. And that success is essential to the physics of the SSC future.

ACKNOWLEDGEMENT

The ideas and speculations expressed in this paper have benefited from many useful discussions and a long collaboration with my colleagues on the Advanced Computer Program: H.Areti, R.Atac, J.Biel, J.Deppe, M.Edel, M.Fischler, I.Gaines, and D.Husby. They are only responsible for the good ideas.

REFERENCES

1. Nash, T., *Conference ... Summary* Proc.Comp. in HEP, Asilomar, CA, Feb. 2-6, 1987, Comp.Phys.Comm. 45, 9-14 (1987).
2. Nash, Thomas, *Trieste Conference ..., Summary and Concluding Remarks,* Adriatico Conf...processers in Part. Phys., Trieste, March 28-30 ,1988, to be published. FERMILAB-Conf-88/110.
3. Gaines, I. and Nash, T., *Use of New Computer Technologies in Elem. Part.Phys.*, Ann. Rev. Nucl. Part. Sci. 37, 177-212 (1987)
4. Kunz, P. F., Nucl. Instrum. Methods 135 435-440 (1976); Kunz, P.F., Gravina, M., Oxoby, G., Trang, Q., Fucci, A., Jacobs, D., Martin, B., and Storr, K., *The 3081/E Processor* in Proc. Three Day In-Depth Rev. on the Impact of Specialized Processors in Elementary Part. Phys., Padova, 1983, 83-100 (1983).
5. Gaines, I., Areti, H., Atac., R., Biel, J., Cook, A., Fischler, M., Hance, R., Husby, D., Nash, T., and Zmuda, T., *The ACP Multipr. Sys. at Fermilab,* Comp. Phys. Comm. 45 323-329 (1987); Biel, J., Areti, H., Atac., R., Cook, A., Fischler, M., Gaines, I.,Hance, R., Husby, D., Nash, T., and Zmuda, T., *Software for the ACP Multipr. Sys.*,Comp. Phys. Comm. 45 331-3379 (1987).
6. Nash, T. ,Areti, H., Atac, R., Biel, J., Cook, A., Deppe, J., Edel, M., Fischler, M., Gaines, I., Hance, R., Husby, D., Isely, M., Miranda, E., Miranda, M., Pham, T., Zmuda, T., Eichten, E., Hockney, G., Mackenzie, P., Thacker, H.B., and Toussaint, D.,, *High Performance Parallel Computers for Science: New Developments at the Fermilab ACP* in Proc. of Workshop on *Computational Atomic & Nucl. Phys. at One Gigaflop,* Oak Ridge, Apr.14-16, 1988, to be published. FERMILAB-Conf-88/97.
7. Mackenzie, P., Eichten, E., Hockney, G., Thacker, H.B., Atac, R., Cook, A., Fischler, M., Gaines, I., Husby, D., Nash, T., *ACPMAPS: The Fermilab Lattice Supercomputer Project,* Proc. Int. Symp. *Field Theory on the Lattice,* Seillac, France, Sept. 28- Oct. 2, 1987, Nucl. Phys.B (Proc. Suppl.)4, pp 580-584 (1988)..; Nash, T., Areti, H., Atac, R., Biel, J., Cook, A., Deppe, J., Edel, M., Fischler, M., Gaines, I., Husby, D., Pham, T., Zmuda, T., Eichten, E., Hockney, G., Mackenzie, P., Thacker, H. B., and Toussaint, D., *The Fermilab ACP Multi-array Processor System (ACPMAPS) A Site Oriented Supercomputer For Theoretical Physics,* in Trieste Conf. *ibid,* to be published. FERMILAB-Conf-88/111.
8. Hance, R., Areti, H., Atac, R., Biel, J., Cook, A., Fischler, M., Gaines, I., Husby, D., Nash., T., and Zmuda, T., *The ACP Branchbus and Real Time Applications of the ACP Multiprocessor System,* IEEE Trans. Nucl. Sci, NS-34, 878-883 (1987).
9. *Second Generation ACP Multiprocessor System: System Specification Document;* July 25, 1988 and revisions;ACP unpubl. int. doc. Availability of ACP publications are listed at HEPNET location, `fnacp::acpdocs_root:[docs]doclist.doc`
10. Christ, N.H. and Terrano, A. E., IEEE Trans. Comp. C-33(4) 344 (1984); Beteem, J., Denneau, M. and Weingarten, D., J. Stat. Phys., 43 1171 (1986); Albanese, M., et al., The APE Computer, ROM2F/87/005 (1987).
11. ZEUS collab., *ZEUS Status Report 1987,* DESY, Hamburg (1987).
12. Fox, G.C., Otto, S.W. *Phys. Today* 37(5): 50-59 (1984); Fox,G. *The performance of the Caltech Hypercube in scientific calculations.* In Supercomputers - Algorithms, Architectures and Scientific Computation, ed. F.A. Masten, T. Tajima, Univ. Texas (1985); Seitz, C.L., Commun. ACM 28 22 (1985).
13. Denby,B.,*Neural Networks & Cellular Automation in Exp. High Energy Physics,* Comp. Phys. Comm. 49, 429-448 (1988).
14. Barsotti, E., Appel, J.A., Bracker, S., *et al.,* IEEE Trans. Nucl. Sci. 26(1),686-96(1979); Avilez, C., Borten, L., Christian, M., *et al,* Proc. Symp. on ... Computing ... for High Energy Physics, Guanauato, Mexico 1984, Fermilab, Batavia, 45-54 (1984).
15. Conetti, S. in Trieste Conf. *ibid* to be published.