DR. THOMAS NASH
Fermi National Accelerator Laboratory

The job of experimental high-energy physicists is twofold: for the cases where people like Norman Christ have successfully calculated predictions we have to check to see if they have done it correctly. For cases where they haven't calculated experimenters' results in advance, we provide them, in principle, with the intuition to understand how to get the right answers.

Experiments at Fermilab in the near future are somewhat typified by the apparatus shown in Fig. 1.

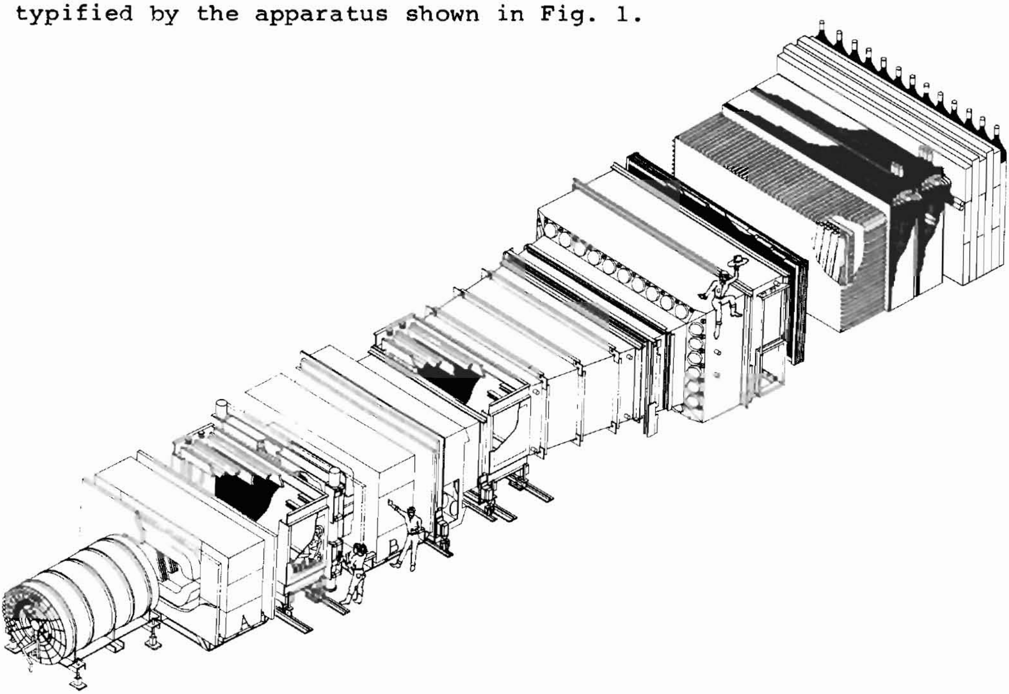

Fig. 1. Fermilab Tagged Photon Spectrometer. The apparatus is more than 25 meters long. The beam enters a target and recoil detector system on the left. The second figure is leaning on the second of two magnets. Drift chambers and an electromagnetic calorimeter are interlaced among the magnets. Large Cerenkov counters follow along with more drift chambers and calorimeters.

The scale of this experiment is indicated by the small figures. This was drawn by a Mexican artist and the sombrero is barely visible. The problem basically comes down to the following: there's a beam of particles of one kind or another that strike a target. A variety of secondary particles come off of the interaction of the beam with the target. This apparatus measures the angles and identities of all the secondary particles to study the physics of the interactions. This is done in a series of detectors that measure the point at which a particular projectile passed.

The analysis of this kind of experiment involves the reconstruction of all the data from these detectors. Just to indicate the scale of the problem for a recent experiment using this particular apparatus there were 1,000 6250 bpi tapes, containing about 25 million events with 1,500 words per event. Each event takes about a second on a Cyber 175 computer. This is pretty close to a Cyber year. This experiment is being analyzed on 20% of Fermilab's computer center, 30% of an IBM 3033, 3 VAXs, and 6 so-called 168E emulators, altogether equivalent to another 4 Cyber 175. Clearly there is a problem in getting this kind of data through.

We anticipate this problem will get worse with the Tevatron. We are trying to deal with this on two fronts, one is a 5 million dollar upgrade for the computer center. The other is the program that I'm involved with. This is the Advanced Computer R & D Program whose intention is to confront the computing-bound problems in high-energy physics by developing new approaches and thereby generally stimulating the computing atmosphere here at Fermilab. The interaction with industry and university computer science departments is one of our important mandates. This interaction has been quite fruitful up to now, and we hope it will remain so in the future.

The first project that we are concentrating on is an event reconstruction processor that focuses on the problems that were just outlined. The idea that we're pursuing is combining the power of specialized devices with more general purpose machines. We have some experience with the special purpose processor shown in fig. 2 which was developed here. It is incredibly powerful but rather inflexible. In one example, this processor, costing about $100,000, was able to do in 7 microseconds what a million and a half dollar Cyber 175 could do in about 40 milliseconds. Thus it is possible to do a lot with such devices, but they are not easy to program. That is why it's desirable to combine that power with the programmability of microprocessors that have Fortran compilers. The intention is to stay extremely modular in order to allow optimizing architechure for different classes of problems, and thereby maximize hardware utilization. We hope this will include the possibility of array interconnections for lattice gauge problems.

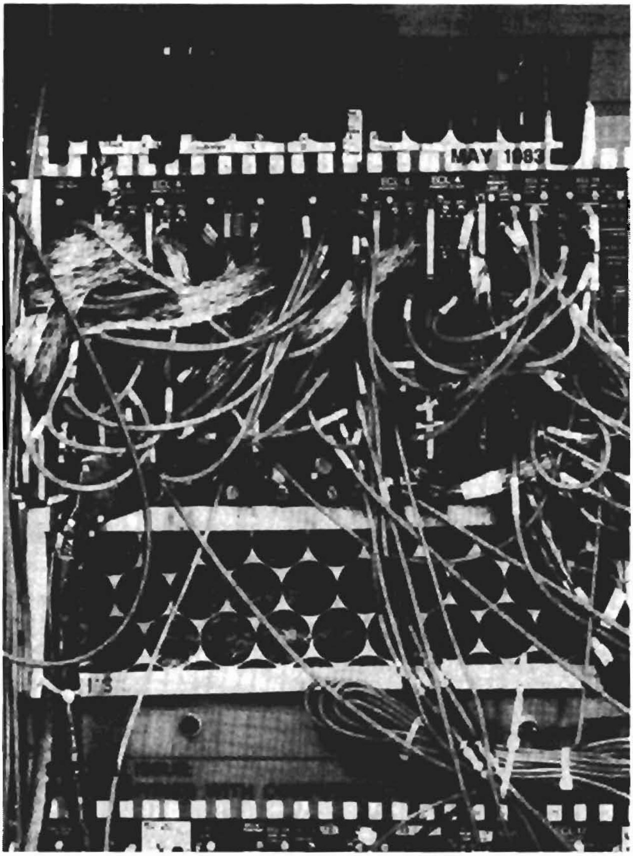

Fig. 2. Photograph of the powerful ECL Data Driven Processor developed at Fermilab. This system is configurable for different problems.

In the reconstruction problem multiprocessing is encouraged by several characteristics. The events are independent, the problem breaks easily into major vertical subroutines, within the events there is intrinsic parallelism, and most importantly there exists an instruction sequence that dominates the computing time.

What would a full-blown system naturally look like? Here one can divide the problem into a series of different subroutines each one of which takes a different amount of relative time, so that it is necessary to have the right number of processors to handle a particular level so that there aren't any traffic jams. Figure 3 illustrates the approach. The crucial idea that we're emphasizing, indicated by the circles, are the co-processors which are special purpose devices to do certain kernels of the algorithms extremely rapidly and effectively. However, as a first step in parallel with the co-processors, we are considering a simple system of microprocessors particularly appropriate to use in this kind of a system. The microprocessors must have good Fortran. We are now actively evaluating such processors and have a long list of candidates. We are discussing with various corporations the possibility of research agreements and arrangements that can help us solve our problem.

The co-processor concept is a generalization of the co-processors used as a commercially supplied adjunct to a microprocessor chip. The word is usually used in the context of the floating point co-processor. Here we mean it to be special purpose hardware to carry out at "blinding speed" the kind of algorithms that one needs, such as finding the line through 3 points in a set of wire chambers, using non Von Neumann techniques such as hit-arrays, memories, fast-cache, memory table look-ups and so forth.

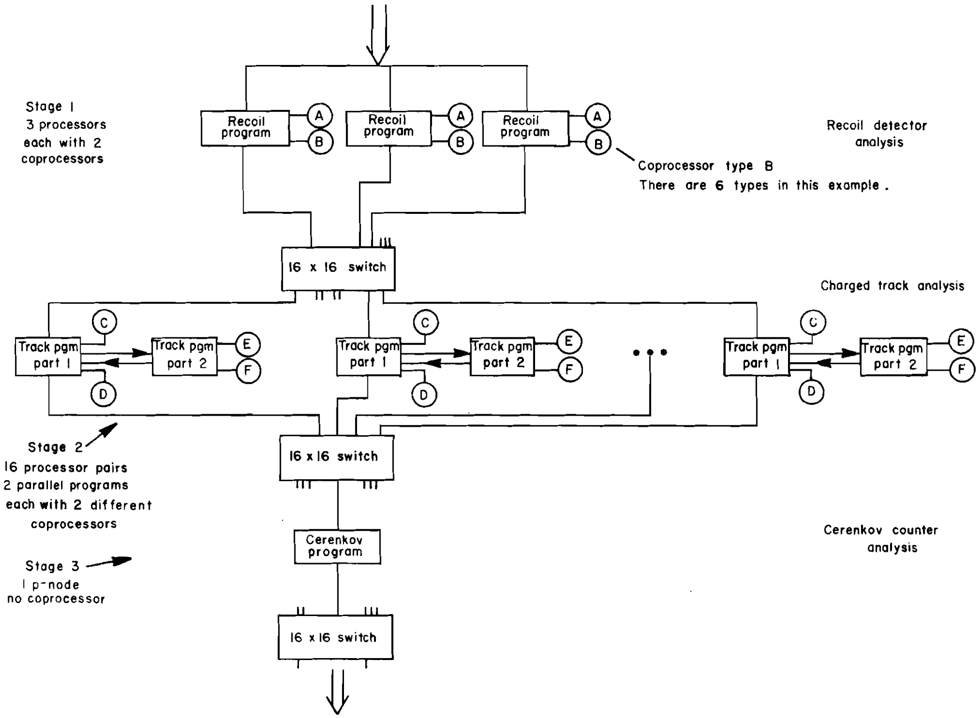

Fig. 3. Full blown co-processor system; the circles indicate co-processors.

The concept from a users standpoint is that the system looks very similar to a library. The user's program would be a series of subroutine calls to that library, well documented of course. Every time he calls a particular subroutine, the system would go into the hardware, execute the complex operation involving loops if necessary at high speed and return quickly. That kind of an approach has some broad applications. The speed up potential of co-processors (which is a critical issue), goes as 1/(1-f) where f is the fraction of the time spent in the co-processor algorithm when you are running without co-processors. For example, if 90% of the computing is inside the co-processor then there will be a maximum speed up of 10. On the other hand, if only one half is inside then there will be a speed up of no more than two. So the crucial issue is how much can be diverted into the co-processor in any particular problem. In order to answer that, a study has been made of the structure of our particular kind of problems. As Ken Wilson alluded to earlier, we find that they are dominated by lists and list manipulations and they turn out to be very similar operations to those used in relational data base problems which is clearly an application area far outside high energy physics. By a list we simply mean a series of columns of numbers that have identifiable attributes. In our problem what happens is that we have a series of disconnected lists that we start with as shown in Fig. 4.

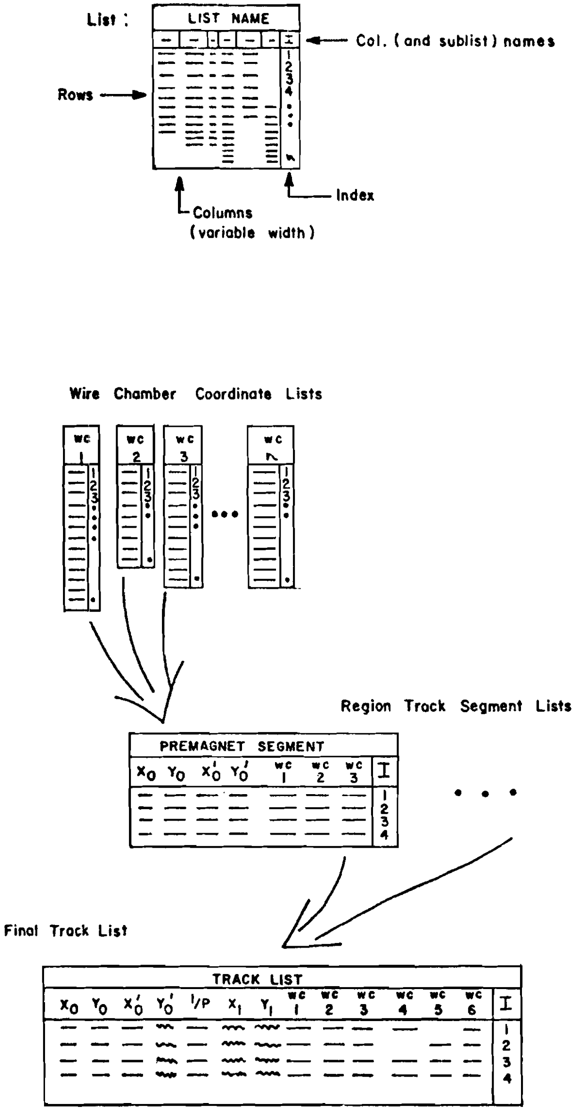

Fig. 4. List manipulation in a typical track recognition algorithm.

That is the raw data coming from the various wire chambers that have identified the particle track as it goes by. Through manipulation using what are equivalent to data base concepts these lists are related to make new lists which are the track segments in one section of the detector. Finally these segments are developed into a final track list. This is an over-simplified explanation but the operations involved are identifiable with those used elsewhere.

To summarize, at Fermilab the hardware subroutine-assisted multi microprocessor approach is natural for our three dominant computing problems which are track reconstruction, lattice gauge calculations, and beam orbit calculations. The latter are required to design the giant accelerators that are now being discussed.

In general the program is aimed at classes of computing problems which have some natural simple parallelism such as the event structure of high energy physics experimental data and that have a definable algorithm kernel that dominates the time. In addition the approach can take advantage of the structured vertical blocks in a program. This is not just for high-energy physics; it's really for problems that can run in a static configuration for days or weeks at a time, where the architechture can be reconfigured to optimize it for each problem. You can imagine that operators are not just plugging in tapes. They can also be plugging in modules for the programs working on that time scale.

There is a problem with semantics in the computing business. People think in terms of either fully general purpose computers or in terms of special purpose computers. But there is really a whole spectrum in between. I don't know what words to use and it's one that we are struggling with because sometimes semantics becomes important. The point here is that our kind of approach is not generally applicable. There are many problems for which it is totally inappropriate. But it is broadly applicable to many other problems. Perhaps the system should be called a flexible hardware assisted multiprocessor.