PANEL DISCUSSION

THE NEW INTEREST IN PARALLEL COMPUTING.

Smith:

Many university researchers are suddenly interested in the architecture of parallel computers. What is the motivation?

Schwartz:

The basic motivations are two fold. First, there is a genuine technological opportunity, coming from the falling cost of the elementary processor. Second, everybody smells money and sees this as an opportunity to build up their own activities.

Smith:

Is the primary technological opportunity afforded by the reduced cost of hardware, the ease of designing the LSI, or more understanding of parallel computing?

Schwartz:

The falling cost.

Arvind:

My point of view is slightly different. I have been interested in parallel computing at least since 1975. I did not think in terms of building one then; the investigation was theoretical. I really believed that something was wrong in the way parallel computing was being attempted even though there were not very many parallel machines at that time. I always had doubts as to whether one could connect 16 commercially available processors together to get high performance out of the system.

Smith:

There are a few notably quiet members of the panel who are not saying what I expected them to say, namely, that we need to get our problems solved. Where are you guys when I need you?

Nash:

The point is that there is no more than another order of magnitude and a half, or so, in the conventional approach to enhancing computing and that's far away. There is just no other way of gaining large factors of computing power except by going parallel.

Wilson:

What excited me was when it became clear that by going to parallelism the jump in performance would be far more than the normal jump expected from commercial development. The factor of 100 per decade is going to be totally eclipsed by the gains that come from parallelism.

THE IMPORTANCE OF IMPLEMENTING ARCHITECTURAL IDEAS IN HARDWARE

Smith:

Under what circumstances is it important that university researchers have their architectural ideas implemented in real hardware? It's fairly clear that if you have a problem to solve and you are desperate you build real hardware. At what stage do people like Arvind at MIT or Schwartz at NYU need to build hardware?

Arvind:

I have had lots of discussions about this very question within MIT and with other colleagues outside. I believe it is important to go as far as possible with analysis, because building machines is very tedious. To build a machine which somebody else can use, is really a big task. It's not worth undertaking unless there is a good reason to believe that there is a bright architectural idea; the machine based on the idea cannot be analyzed any further but is worthy of further exploration. In other words, premature construction of new machines can be a colossal waste of time and money.

I strongly recommend that people should go as far as they can with the analysis and simulation of their architecture. They should talk to real users and program some applications before attempting to put together a machine which can be used. For example, there are two machines on Schwartz's chart, CM* and C. mmp, both at Carnegie Mellon University. The reliability of CM* was so poor and the total computing power so limited that few application programmers were interested in writing programs for the machine. The main claim the designers can make is that they put together 60 PDP - 11's and set some sort of a world record. I don't think it is worth building such machines because the learning is not commensurate with the money and the effort involved. Either the technology itself is to be tested by putting these machines together or the architectural concept is to be proven. The goals should be clear at the start.

Christ:

This is one aspect of the problem that has been changing. There was a time when building a computer was much more difficult than programming it. But as microprocessors and large scale integrated chips have become available that are really able to be interconnected by amateurs. At some point the possibility of building hardware for particular processes involves less work than programming. We may not have yet reached that stage, but in fact we might.

Schwartz:

This may be true for existing hardware. However building some new device like a Fast Fourier Transform chip is definitely several orders of magnitude harder than doing it in software.

Panel:

Perhaps our moderator could describe the Denelcor HEP approach to parallelism?

Smith:

The Denelcor HEP computer system is a dance hall machine in the sense of having all the boys on one side of the room and all the girls on the other with a switch in the middle. It has processors and memories and a connection network that ties them together, very much like the picture that Schwartz showed. It's not a data-flow machine according to some people but it is according to others. In fact, it's a hybrid lying somewhere between a data-flow machine and a multiprocessor.

It overcomes two of Arvind's objections to parallel architectures: it is possible to not wait for memory references and it is possible to avoid rewrite races in this architecture. It is still necessary to schedule programs using program counters as well as by the availability of data. In that sense it is a hybrid between the Von Neumann approach which schedules everything with program counters and the data-flow approach which schedules everything with the availability or the need for data. The HEP machine is pretty fast. It executes from 10 to 160 million instructions a second on 64 bit words. It's also pretty new. There are six HEP processors in the field in three systems. Each processor is a parallel processor though.

Wilson:

Bert Smith's HEP machine is the first machine for parallel processing which was designed the right way. The inter-communication network was designed first. The processors and the memory are designed to have all the features that the network requires including various signaling primitives, so that different processors can communicate with each other. The processors are designed so that they are not slowed down at all by the time-delay in the network. There are a number of really nice features on the HEP machine. The reason that it is not selling like hotcakes is that it is also expensive. It does not give the cost performance one would like to have these days.

However, we are leaning very hard on the agencies like the Department of Energy to point out that we have got to have HEPs available for experiments on parallel processing for which it is unique.

Smith:

The word unique is maybe a little strong in the light of what is going on in places like MIT and NYU. However, it is an available machine. Machines like the Ultra computer, the Tagged Token Data-flow machine that Arvind was describing, machines with static data-flow of which Jack Denis is an advocate, the Cedar machine at the University of Illinois, are being looked at, evaluated and in some cases designed at various universities. However, these machines are not yet available. Our machine is very much akin to those machines, that is, it has the same sort of applications and the same generic architecture.

Arvind:

I would like to make a simple point. Besides the fact that it is probably the most innovative machine which is out in the field today, I believe if people actually started using it we, the architects, would learn something from it. If there were eight or ten users of the machine, we would learn answers to questions such as "Does the machine stay up? Is it easy to program?"

STRATEGY OF DEVELOPMENT - HOW MANY GROUPS CAN BE INVOLVED?

Audience:

If you restrict attention to the people who have been working seriously on computing rather than on special purpose applications, that is, the people for whom the computing is more of an end in itself other than a means to a scientific end, then there is easily a score of very worthy projects. All of these are short of money and even more short of talent.

There is a difficult decision that has to be made because there is not enough money to go around. So Jack Schwartz raised the question about how do we narrow this down. Do we let one hundred flowers bloom? We certainly don't go as far as the Japanese in mandating from above. Perhaps there is a happy mid-ground.

Smith:

Underlying our concerns is the fact that in some cases it is necessary not just to build machines to avoid the problems of simulation but to build serious machines. In fact these must be quite serious machines. In Arvind's words these have to be machines that get some of the industrial people and some of the national laboratory people interested in actually writing programs and developing applications for the hardware. We need to develop computers with performance in the multimillion operation per second range and with concomitant tolerance invested in them. Are there comments?

Schwartz:

The right way to organize this would be to spot in a judiciously chosen set of projects across the spectrum of possibilities which tried all the major alternatives at a significant but neverless a relatively modest level of capability. Right now the universities are by and large trying to advance to the 64 processor level with relatively small processors. To go beyond that, one is starting to talk of major facilities and larger dollars. The right way to do that is to have some judiciously organized technical runoff with a smaller number of machines allowed to advance to the thousand and four thousand processor level. I am not sure that that degree of judicious judgment is being brought to bear.

COOPERATION WITH INDUSTRY

Smith:

What cooperation with industry have you had so far, and what cooperation would you like to have?

Wallace:

Within the UK scene we have had rather good cooperation with ICL. We would almost certainly not have had the machine but for a meeting between the regional computing center director and the director of ICL. It does appear to us that we have been able to show some of the things that this kind of machine can do. There is now a wider interest in other companies both within the United

Kingdom and the United States. We are optimistic that our future needs for computing engines can be met by early access to the manufacturers' prototypes that we really want.

Arvind:

It is not very difficult to convince a manufacturer to give you a computer if you say you will write programs for it. They will love you if you do that. We are made offers like this all the time. We have to turn them down because we cannot absorb too many different types of machines. The really difficult cooperation with industry is where we want industry to build a machine based on our ideas. This is the type of cooperation that is required to build new types of supercomputers.

Nash:

Over the last year we have had a seminar series on new computer concepts. We had a very difficult time getting people from industry to speak though we did have a couple of very good talks. What we are learning is that industry likes to talk on a one-on-one basis. That is, if I call up someone with whom I made a contact and start talking they will pretty quickly get me information on a non-disclosure basis. This does create a difficulty. With any large audience an industry person is apparently unlikely to say anything of great substance. On the other hand, if you try to talk to them individually there appears to be a lot of cooperation. As to what cooperation we are looking for, first, we are trying to get our hands on certain proprietary chips. Second, we need help with simulation software. Third, is an area where we are having difficulty which

I call crystal ball gazing. This is getting realistic estimates on what memories or processors will cost 18 or 24 months down the road. Industry is very good at that. They have in house crystal ball gazers who specialize in that. That sort of information could be very helpful to us.

Brenner:

I have a question for Wallace. After Illiac 4 the DAP is the first commercially available processor of its kind. There has been good cooperation between ICL and the universities and the National Research Council. Nevertheless it has been a commercial flop. What went wrong? What should one do differently? What can one do better to make it work next time around?

Wallace:

The one specific point that I already mentioned was that the manufacturer tied it very tightly into ICL main frames. These don't sell in any numbers abroad, but there are a number of centers in the United Kingdom with ICL main frames, so there is some kind of market for the DAP within Britain. Tying them to the mainframes was (at least with hindsight) a major mistake. If they had mounted it on a smaller machine like a VAX (as, one suspects, the designers proposed but were over ruled by the previous management) they would have had a viable machine three years ago costing $500,000 that would have approached Cray power for a wide range of problems. I think that would have been an interesting proposition.

Wilson:

Note that there was an ICL DAP that was bought commercially and then turned back to the company because the company couldn't make use of it. One of the problems with these machines is that someone has to learn how to use them. It's easier to learn in a university setting, partly because the problems people are trying to solve in a university setting are simpler than the problems that industry has to solve. I believe the logical progression is to learn how to use these machines starting in a university. Some expertise is established in what the machine is really good for and how to program it for that use. Then it is sent out to industry, but at same time with industry consulting with the universities to make sure they are buying it for something that is feasible to use it for.

This is not a Cray substitute. There are plenty of things that can be done very well on the Cray which could be done in principle on the DAP, but it would be a devil to program. This is especially true if the programs are already written for some other machine. Then it would be a disaster.

Wallace:

I don't think ICL thought very deeply about what the machine would be used for. I suspect if they thought about the possibility of having general image processing devices using the basic arrays out of which the DAP is built, then they could have built the DAP as a special machine based on these arrays. Possibly use of the basic arrays for image processing would have been commmercially feasible.

This is one of the points that encourages us to continue to work in this area because we can see that we are riding on the back of what could be a commercially successful architecture.

Christ:

At a somewhat lower level I could describe our interactions with the Intel Corporation. We are using their microprocessor in our device. We have gotten a fair amount of assistance from them. They have a program for interacting with universities that in our case essentially saved us 50% in the cost of items that we bought. We hope that perhaps our direction in using microprocessors might at some point produce a market for them.

Wilson:

Our interaction with Floating Point Systems was thrust on them. At first it was not something they particularly welcomed. It started when we bought one of their processors and asked people in the traditional FPS markets about Fortran. The general reation was "what?", because they were used to programming these machines in assembly language. That was the FPS market niche. I talked with the FPS people about Fortran and they said well maybe sometime. Then we started our Fortran project and a month later they hired a director of their Fortran project. We are not quite sure what the connection was. They actually bought half of the compiler we wrote. There were enormous difficulties that developed around that because of course they didn't understand very much about Fortran. They had various difficulties that they hadn't anticipated with our compiler. Actually it wasn't a very good compiler.

It took us a couple of years to get back their confidence after that.

We got a lot more respect from FPS after the manager of the project at Cornell started sending one paragraph summaries of every phone call he received with respect to the array processor. Somebody would call him up and a description of that phone call went off to FPS. One thing that became very clear is that it is necessary to identify a "friend" inside industry who can help. I think that everybody that deals with industry finds this. This is a particular person who is willing to fight on your behalf in company politics. The friend we had at FPS was the Vice President for Marketing. First, he had come from CDC and he was used to the role that universities have, because CDC had had some experience with that where as FPS did not. I remember one session when they were discussing the software for the FPS 164 and I was explaining what the universities would do. The FPS people were looking skeptical so they asked the Vice President for Marketing about the earlier experience of CDC. He described how important it was to have the universities running their equipment. Then someone pointed out the universities were running their own software, and didn't that hurt. The Vice President just laughed, because of course there was no CDC software worthy of the name at that time.

THE RELATION BETWEEN INDUSTRY, UNIVERSITIES AND NATIONAL LABORATORIES

Nash:

It's interesting to compare the roles and capabilities of industry, the weapons laboratories like Livermore and Los Alamos, the high-energy physics laboratories, and the universities to see where they are strong and where they are weak. There are a number of relevant factors like secrecy, hardware capability, and architectural creativity that one can see. There may be some hidden variables in industry that we don't know about such as the available dollars, the ability to work fast which is most important, the near versus far-sightedness, and the software capability. Recently for fun I put down all these factors; I scored them and added it all up. Surprisingly, their scores came out exactly the same to the last decimal point.

The point is, and it's an important point, in each area there are some great strengths and some weaknesses. For example, my personal view is that we at Fermilab are poor in software. Another example, is that the universities are creative and have a lot of foresightedness, but the ability to push something out the door is admittedly weak. (Even though this is one persons evaluation, Burt Smith looked at the chart and felt it in fact was pretty close to his own point of view.) The question is how can we get these four different, rather entrenched perspectives together. It's not easy, because they each have different motivations, different interests and fundamentally different perspectives. I don't know the answer but in some sense that is why I put this up here.

Smith:

George Michael of Livermore and Bill Buzbee of Los Alamos, with the cooperation of Don Austin of the Department of Energy, have been sponsoring a series of meetings pertaining to architecture, applications, algorithms, programming environments, new programming languages, and the like. These meetings get people from the national laboratories active in defense together with colleagues from industry and universities. The problem you have pointed out is well recognized within the Department of Energy.

Arvind:

I think this is one area where big bucks can make a big difference. If a national initiative is taken you can really bring together users as well as industries and universities. Somebody has to figure out all the details of how this cooperation is to be brought about.

Audience:

These are big institutional units, but isn't this really a matter of making something happen between two people?

Smith:

When I first started taking part in these meetings I found that what I was learning was language. What we have to do is teach each other our languages so that we can develop those one-to-one communications. I find now that I can go to a Monte-Carlo conference and understand about 50% of what's said.

I can talk to people about aspects of computational physics or other application areas much better by virtue of the fact that people have been talking to me about these subjects for some time. In part that's what this meeting is about.

Audience:

Some of these projects are very large. A lot of money is needed to develop these kinds of systems. The question is how can you work with industry? From my perspective I am not sure that there are that many products in industry that can support that kind of funding. I think that this is part of the problem. Is there some way where one can work with a university in a useful way without a product?

Nash:

That is what I was getting at earlier in terms of our needs at Fermilab. We can get funding at some reasonable level through our basic sources but from our perspective what we need are certain of the things that industry is very good at. The crystal ball gazers, the simulation, the proprietary chips, board level computers that they can produce in large quantities, and computer-aided design capabilities. In our case it is not funding that we are after.

Schwartz:

In this area the universities have started to function as fast-moving scouts wandering over the terrain and discovering many interesting possibilities. The ideal role for industry would be to be the large battalions that come marching behind them and do a good job of putting substantial equipment in place.

A crucial element that is missing now is that they are not marching, with the exception of a couple of small companies like DENELCOR.

The basic problem is that in this area industries are trying to decide whether they want to be involved at all. The smaller manufacturers have their product concerns. They have to have short-term product development goals to make money. In Japan the situation is different and industry is on the march. They may not have scouted the terrain very well but as a matter of fact we are doing that for them.

Wilson:

It is often difficult to find product support for university operations. Take lasers as an example. The time delay from when the university research was done to the present $100 billion revolution in laser communications was two decades. Twenty years ago a university researcher couldn't get access to any of that $100 billion. On the other hand, the computing situation is different. Floating Point System's next 32 bit product has to find a one billion dollar market otherwise Floating Point Systems goes down the tube. This product is something it has to produce in the next year. This is not long term. FPS's basic problem is having the right ideas as to what to build and to know that product will find the necessary market. Remember that's one billion dollars and Floating Point Systems is a company many people have never heard of, it's not IBM.

The bucks we are talking about in the computing business are incredible and they are incredible compared to other areas where one talks about technology transfer.

One serious problem is getting the universities into the computing game. The universities don't recognize the opportunity that is there. The second problem is working out how to pay the universities for their function once they got into the game.

Audience:

Would some kind of a clearing house for funding be useful where smaller industries not big enough to back entire projects could make contact with and assist university researchers?

Smith:

As I understand it, the panelists want other support than just financial support. There was some discussion of MCC earlier. That isn't such a clearing house. There are research foundations and other methods of channeling industrial funds into research activities. I believe the emphasis here was on something rather different, namely how do we get technical information from industry and how do we communicate what we find. How do we act as scouts, perhaps speaking a different language. I wonder how many of the battalions speak Arapaho or Cheyenne which seems to be what some of us are speaking at times. However this may be difficult to pay because instead of costing money it costs talent. Talent is as expensive in industry as it is in academia.

Wilson:

Of course there is a problem with money, but there is another problem. The number of computer scientists is absurdly small for the needs of the United States. The requirements for people are not calculated properly either, because people usually don't estimate how many computer scientists are needed to go in and start up companies. Manpower estimates are always in terms of the established market. The proper way to count is not just how many people Hewlett-Packard hires.

The money for computers and computer science is absurdly small in the university scene today even compared with other subjects like physics. This is because computer science started late and only really gathered steam in the seventies after the big funding crunch. Universities that have to think of terms of a twenty year time scale for a professor aren't eager to run up their funds rapidly. On the other hand the computer science students are doubling every year and the funding is not growing to match.

Part of the trouble is the big funding from ARPA only goes to a very few universities so NSF as usual is left holding the bag. Nobody is giving support to raising the funding for computer science at NSF including most of the people inside NSF. As a result an absurdly small sum of money is determining how many computer scientists we're going to have ten to twenty years from now.

I hope some of you will start complaining, at least to the government, that the ratio of funds for computer science has got to be changed to be more sensible.

IS THERE SUCH A THING AS A GENERAL PURPOSE PARALLEL COMPUTER?

Schwartz:

We believe the design we are proposing and the data flow machines are relatively general purpose. I see them as the parallel analogs of the IBM 3081. There will also be special devices, and the special devices will perform well.

Of course it's hard to define "general purpose" as precisely for parallel machines as it is for the 3081. The IBM 3081 is a general-purpose machine because it can be programmed for hundreds of applications. I believe the same will be true of parallel machines except that those will be exclusively for large applications.

Nash:

Maybe one should say that "general purpose" for a parallel machine means that the machine can be efficiently programmed for just about every problem, but not necessarily be the optimal processor for every problem.

Wilson:

It isn't reasonable to call a parallel processing machine general purpose because of the way typical users would understand that term. It will always be possible to find a problem which cannot be solved any faster on a parallel processor than it can be solved on one element of that processor. There are even mathematical theorems to that effect. What is fair to say is that there are going to be enormously cost effective ways to use parallel processors for problems for which they are suited. There will be a distortion of the computer market towards the problems which are suited to parallel processing because they will be enormously effective. Huge markets will develop around those applications in data bases and scientific computing. There will be other areas which will not develop because they are not suitable to parallel processing.

Smith:

Schwartz is saying regardless of whether there is one problem or perhaps half-a-dozen it doesn't make any difference because for economic reasons, people will want to buy parallel processors for what we now call general purpose computing.

Wilson:

I think we should talk about single purpose and multi-purpose parallel processing but I think it is dangerous to call it general purpose.

Smith:

But you are saying that in some future time we will be doing data base management and payroll checks in parallel?

Nash:

Why not?

Arvind:

I think it is a non-issue, you don't buy a truck to drive to your office, yet trucks are general purpose.

Schwartz:

Looking at those present I would come down a little differently. If one looks ahead to the availability of general-purpose parallel machines, Al Brenner in charge of a large computer center, will buy a general-purpose machine where Tom Nash is interested in a very special purpose machine for a particular experiment.

Nash:

Semantics are important here because planners are often not that cognizant of the details. One has to be careful that by using buzz words like "general purpose" or "artificial intelligence" the problem is stereotyped too much and one might end up funding and planning things in the wrong way.

ARTIFICIAL INTELLIGENCE AND SCIENTIFIC COMPUTING

Smith:

At a recent workshop, Duane Adams of DARPA walked up to Ken Wilson and me and asked if there were any possibility that the same sort of computer could be used to do scientific computing and artificial intelligence. Do we need one score or two of

DARPA machines and one score or two of scientific machines? Who knows what's next? Is some sort of synthesis possible?

Wilson:

It's a little hard to know what artificial intelligence is these days. It does seem to me that there is going to be an enormous need for high bandwidth data movement. That is as close as one comes to a general need. There is a limit to what can be done with one bus to move data around. What we are talking about in these parallel systems ultimately comes down to the form of the network on which the data movement takes place. The question is what is its total aggregate bandwidth? Artificial intelligence is going to need that just as much as the scientific processor.

Brenner:

The semantics question is serious. These days DARPA is also using the term supercomputer. This confuses the issue of overlap. What we in the scientific community mean by supercomputer and what they want in an artificial intelligence super machine are not necessarily the same. What we want is a number crunching super machine. One should be very careful about that. DARPA has done a disservice to society in making that confusion. We should all straighten that out whenever possible.

Arvind:

Even though today these machines are very different I don't agree with that. Both sides require the other. It is clear that there are AI applications which really require very fast floating point arithmetic.

One moves into robotics the more that is the case. Robotics applications cannot be done very effectively on many of these AI type of machines. All the proposed artificial intelligence machines will have fast floating point units on them, precisely for that reason. Similarily, in scientific computing it is very hard to graduate beyond machines designed to execute "inner loops" efficiently unless applications are examined in totality. Until now, designers of supercomputers have ignored the problems of managing large address spaces, and the I/O bandwith between primary and secondary storage. These problems have to be solved if data-base management and graphics are to be integrated in scientific computing. Designs of AI machines have a head start over designs of supercomputers in these areas.

Schwartz:

If I were asked what is the ideal computing machine for dealing with those equations that are going to replace quantum chromodynamics, I would have to reply that I don't know. I don't know what those equations are, hence I can't say what computing machine is ideal for dealing with them. I have the same sense of bewilderment about the ideal artificial intelligence machine. If you look at the field technically there is such a shifting mass of paradigms in use that it is hard to identify what artificial intelligence is.

Smith:

I agree. In particular, the differences between a machine designed for a language like LISP and a machine designed for a language like PROLOG can be quite large.

Nash:

The phrase "artificial intelligence" is an example of the problem with semantics. I've been trying to find out for at least six months what artificial intelligence is. Every time I encounter a computer scientist I ask him. The best answer I got was from David Kuck. He said "artificial intelligence is anything that you can't write a program for." That's a rather large category of problems. Perhaps the buzz words "artificial intelligence" mean what the defense people want them to mean, and what they want is their own computers. It gets to be a territorial question at a certain point. We have to avoid these territorial problems as much as possible.

Smith:

That's very interesting. It seems that the AI machines are machines that we can't write programs for.

Arvind:

Actually I find that comment very strange because some of the largest programs that have been written to date are all in AI.

Smith:

Kuck's statement is "to solve problems you can't write programs for," not that you can't write AI programs. I was jesting of course.

HOW FAST CAN WE EXPECT SUPERCOMPUTERS TO BE BY THE YEAR 2000?

Smith:

We would like to ask the panel to prognosticate a bit and answer the question, "How fast can we expect supercomputers to be by the year 2000?" I don't care what measure you choose, if you all choose different ones then you'll have the advantage of not being compared to your neighbor. Nevertheless, let's have something that we can get our hands on. Considering what's happened in the last seventeen years, what are the next seventeen years going to bring in machines that you can buy or issue a purchase order for in the year 2000.

Arvind:

I would like to know where we are today?

Smith:

So would we all. We don't know where we are today and that's why your metric can be in any scale that you like. You can just say how many Crays or how many IBM 3081s.

Wallace:

For us the simplest point of view is to consider the next generation of the kind of machine that we are working with, which is rather special purpose but still more general than might be thought at first sight. The next generation will be 30 times faster. That is still bit serial, so take 32-bit machines. Another factor has to be included for unanticipated hardware developments that may take place by the year 2000.

So I would suggest that a thousand-fold increase can be anticipated, with further factors from hardware and special algorithms which cannot be foreseen.

Schwartz:

I think I would like to ratify that estimate. That would suggest something around 100,000 million instructions per second (MIPS) for the general-purpose machines and maybe a factor of as much as 100 beyond that for special purpose machines.

Wilson:

Every time I have tried to predict what will happen in the next three years, let alone in the year 2000, anything I have said has been an underestimate. It has been impossible to make an overestimate. For instance the last case that wiped me out was when I said we needed a floating point chip, something with a one microsecond cycle time. I speculated that we might have it in three years. The next day the announcement of the Hewlett-Packard chip was in Electronics Magazine

Smith:

So you are underestimating industry production?

Wilson:

I have underestimated a lot of factors. I underestimated the importance of the ICL DAP. For parallel processors, I underestimated the importance of the Monte Carlo processor at Santa Barbara. These underestimates have forced me to become more radical in my thinking of how computing is developing. A second factor to consider is that for parallel processing "how fast" depends on how may processors are available.

The number of processors depends partly on just how much money someone is willing to spend.

Now I doubt that anybody here today with the possible exception of myself has a concept of how much money people will be spending on computers by the year 2000. Scientists have consistently underestimated the willingness of society to spend money on their science after it has been developed into something.

With these factors in mind I predict that by 1990, not 2000, there will be one billion dollars worth of scientific computing equipment at Cornell. These factors have to be folded into estimates of how fast computers will be because obviously with a billion dollars we can have a lot faster system than my present budget of ten thousand dollars permits. This is true because obviously industry is going to have to spend more money on research and development than it does today. More of the price of the goods will be the research and development cost. Second, industry will have to put a larger fraction of the total R&D budget in computing because computing costs are going down while the cost of everything else goes up. Unfortunately a typical university's management is ridiculous, so that the fact that the cost of computers goes down and the fact that everything else goes up means that they put their money in everything else instead of computers. But, I believe that by 1990 the universities will be important enough to the economy and to industry in particular, so that they will be forced into doing some sensible budgeting.

This will mean that the budget for computers in universities will be a larger fraction of their total budget than it is today. In addition, the importance of universities in the computer market will mean that by 1990 the amount of computing equipment that we have in the universities will be limited not by money but by the total production capacity of the computer industry. In those terms it is perfectly obvious that a billion dollars of computing equipment in one of the best universities of the United States is probably an underestimate rather than an over estimate.

Christ:

I don't see the difference between general purpose and special purpose as clearly as my colleagues here. I wouldn't be surprised if that difference was quite blurred by the year 2000. Scaling up the sort of numbers that one can easily talk about now, a gigaflop or a thousand million floating point operations per second, I wouldn't be at all surprised to see that increased by four orders of magnitude. The hardware may improve by a factor of 100 and the scale of the system that is considered feasible by a similar factor.

Nash.

I like the units megaflops per megadollars. By the year 2000 one will be able to get an enormous boost by using broadly flexible "catered" processors. It is simply a question of how many transistors will be packed on the head of a chip by industry and how effective computer-aided design is going to be for devising circuits to do special purpose co-processors in our concept.

With these one can probably get that factor of a million in cost effectiveness which is the real issue now. If you can get a factor of another million in dollars you have available then you're really doing well.

Arvind:

Today we can probably do twenty million floating point operations per second on a sustained basis. Based on that, I would say in the year 2000 we probably would be able to do 1-10 billion useful operations per second.

Smith:

In summary, we are getting numbers between a thousand and ten thousand with the exception of Wilson who suggests perhaps 10^{14}. My numbers are more-or-less 10^4 for both special purpose and general purpose machines. (These comments were followed by several moments of intense discussion among the panelists as they tried to arrange betting odds with each other on the possibilities.)

WHAT IS THE SINGLE MOST IMPORTANT UNSOLVED PROBLEM IN PARALLEL COMPUTING?

Schwartz:

Getting industry moving on the possibilities.

Wilson:

Learning how to program not for parallel processors but for ordinary sequential processors.

Arvind:

I think technically programming, otherwise cooperation with industry.

Christ:

Knowing how to use effectively parallel architecture, that is thinking of the problem in the right way and programming efficiently.

Nash:

Software!

Wallace:

High level software.

Smith:

Having posed the question, I had an advance look at it. I had a difficult time deciding whether industrial cooperation or software was more important. I came down in favor of industry cooperation primarily because the importance of high speed computing really depends on making it available for people. If we don't have manufacturers and industrial users of high speed computing cooperating in order to bring it into currency then we won't have to worry about how easy the art of programming is or anything else. We just won't be in the ball game. We will only get leverage in high speed computing to the extent that new devices are supplied by industry and used by people like you.