PROGRAMME ON
INFORMATION &
COMMUNICATION
TECHNOLOGIES

Working Paper Series

23

From Symbolic to Numerical Computing

The Story of Thinking Machines

Eloina Pelaez

UNIVERSITY OF EDINBURGH

The Working Paper Series

The ESRC Programme on Information and Communication Technologies (PICT) has a strong commitment to the dissemination of research findings to a wide audience including academics, technologists, government officials, industrialists, trade unionists and community groups. As part of this commitment, Edinburgh PICT (see back cover) produces various internal publications which it distributes directly. These comprise Series of Research Reports, Working Papers and Student Papers.

The Working Paper Series aims to facilitate the fastest possible dissemination of results emanating from PICT research at Edinburgh. Specifically, the Working Papers cover three types of results

(i) preliminary results of research in progress

(ii) results about specific aspects of the Information and Communication Technologies (ICTs), representing little known information of value by way of 'background briefing'

(iii) conceptual or theoretical contributions to current policy or academic debate about technology in general or ICTs in particular

Not all of the papers in the Series originate directly from PICT projects but all reflect an association with Edinburgh PICT which has either benefited from or contributed to this research effort. The views expressed in individual papers are those of the authors and cannot be attributed to any part of the University or the ESRC.

A full list of other Working Papers in the series to date can be found at the back of this paper, along with an order form. Payments should be sent, with orders, to:

Research Centre for Social Sciences
University of Edinburgh
56 George Square
Edinburgh EH8 9JU

[redacted]

Edinburgh PICT
Working Paper No. 23

From Symbolic to Numerical Computing

The Story of Thinking Machines

Eloina Pelaez

1991

ISBN 1-872287-26-3

From Symbolic to Numerical Computing
The Story of Thinking Machines

Eloina Pelaez

The history of Thinking Machines and its computer, the Connection Machine, is a significant part of the general history of computing. Its history raises a whole number of issues central to the understanding of computing and its development. The aim of this paper is to look at some of these issues and thereby to throw light on the nature of technological development and transfer. The focus is on the transition of the Connection Machine from a university project to a commercial undertaking competing with the most advanced computers in the world. It is argued that such a move cannot take place without major changes in the technology involved, and that the shift from university to industry involves not a transfer but a transformation of technology.

1. Introduction

"Washington, Nov. 28, 1989 - Thinking Machines Corporation announced today the receipt of a $12 million contract from the Defense Advanced Research Projects Agency (DARPA) to accelerate its development of the world's most powerful supercomputer, with peak speeds above one trillion operations per second (1 TeraOps)."

The news release (Thinking Machines Corp. 1989) is a clear indicator of the success of a company founded in 1983 to develop massively parallel computers. Yet, once the mind recovers from the figure of one trillion operations per second - "the Holy Grail of supercomputing" (Supercomputing Review, Jan. 1990) - a number of questions arise: what is the relation between this enormous speed and 'thinking'? Is a 'thinking' machine simply a fast machine? What does 'powerful' mean when applied to a computer, and where does the software fit in? What is the role of DARPA in all this, and what is the relation between the projected computer and defence?

From the news release, it is clear that the history of Thinking Machines is a significant part of the general history of computing. It is an interesting object of study not just because of the company's success, but because its history raises a whole number of issues central to the understanding of computing and its development: the nature of the 'parallelism revolution'; the relation between symbolic and numerical computing; the role of commercial pressures in shaping the design of computers; the rise and relative fall of artificial intelligence; the relation between universities and industry; the role of the state in promoting technological development; the relation of software to hardware; the relation between the development of computing and the conceptualisation of science; and so on. The aim of this paper is to look at some of these issues in relation to the development of Thinking Machines and its computer, the Connection Machine, and thereby to throw light on the nature of technological development and transfer.

2. Early Days

"We would like to make a thinking machine. Someday, perhaps soon, we will build a machine that will be able to perform the functions of a human mind, a thinking machine". These are the opening words of the PhD thesis by Daniel Hillis on The Connection Machine, the machine that was to be developed by the Thinking Machines Corporation. As Hillis's words suggest, the project to build a Connection Machine was seen as an artificial intelligence project from the start. It originated in the artificial intelligence Laboratory at MIT, where Hillis was a graduate student, working with Marvin Minsky.

The artificial intelligence background is significant for the way that Hillis and his colleagues approached the question of computer architecture. From the late 1970s/early 1980s, there was an increasing drive behind criticisms of the traditional 'von Neumann' architecture, with its separation of memory and processor, which means that computations can essentially be processed only sequentially, one step at a time (Peláez 1990). It was felt by many that the von Neumann computer was reaching its limits in terms of performance: eventually, it would become more and more difficult to raise the speed of computers by increasing the numbers of processors on a chip, or by making chips run faster. Others felt that the von Neumann architecture imposed constraints on programming which contributed to the unreliability and costliness of software. With different motivations arising from the different perspectives on the limitations of the von Neumann architecture, a large number of projects were developing new computer architectures, and experimenting particularly in the development of parallel computers, in which a number of processors are harnessed to work in parallel. These projects differed in their approach. Some were modifications of the 'vector' computers developed in the late 1960s and early 1970s, modifications designed to increase the extent of concurrent operations in these mainstream supercomputers. Others were more novel, more fully parallel architectures, such as the 'hypercube', in which a large number of processors operate in parallel. Most of these projects came out of a straight computer science background and were mainly conceived as ways of increasing the speed of scientific computation; they made little reference to the notion of a 'thinking' machine.

The Connection Machine's origins in an artificial intelligence environment gave the project a different approach to the meaning of computation. Artificial intelligence has never been a clearly defined discipline, but in general terms it can be said that artificial intelligence approaches the development of computing through a comparison between computing and intelligence. By the mid 1960s, the dominant approach in the subject conceived of intelligence, seen in abstraction from the human organism, as the "ability to process symbols" (McCorduck 1988, 74). Artificial intelligence has therefore focussed on the use of computers to process symbols, that is, on the development of symbolic, non-numerical computing. In discussions of computer architecture, the artificial intelligence approach has been characterised by two things: a concern with symbolic, non-numerical processing and an analogy between computers and the brain - hence the notion of a 'thinking machine'.

The Connection Machine approach to parallel computing reflected both of these concerns. Hillis's account in his thesis started from a discussion of a typical example of symbol-processing: image recognition, the relative ability of humans and computers to recognise visual images. He pointed out how much faster humans are in the recognition of even the simplest images:

> "We might plausibly undertake to program a computer to generate one-sentence descriptions of simple pictures, but the process would be tedious, and the resulting program would be extremely slow. What the human mind does almost effortlessly would take the fastest existing computers many days. These electronic giants that so outmatch us in adding columns of numbers are equally outmatched by us in the processes of symbolic thought" (Hillis 1985, 3).

Hillis attributed the weakness of computers in symbolic as opposed to numerical computation to the architecture of traditional machines. While it is true that there are more neurons in the brain than there are transistors in even the largest computer, nevertheless a transistor can switch about a million times faster than can a neuron: the explanation for the relative slowness of the computer must lie, therefore, not in the number or the speed of the transistors, but in the way in which they are connected to one another. In the traditional 'von Neumann' architecture, with its division between processor and memory, there is a very inefficient use of the computer's resources:

> "In a large von Neumann computer almost none of its billion or so transistors do any useful processing at any given instant. Almost all the transistors are in the memory section of the machine, and only a few of those memory locations are accessed at any given time. The two-part architecture keeps the silicon devoted to processing wonderfully busy, but this is only 2 or 3 percent of the silicon area. The other 97 per cent sits idle... This inefficency remains no matter how fast we make the processor because the length of the computation becomes dominated by the time required to move data between processor and memory. This is called the von Neumann bottleneck. The bigger we build machines, the worse it gets" (Hillis 1985, 4-5).

The way to improve the performance of computers, therefore, was to abandon the von Neumann architecture, to combine memory and processor in small processing cells and to harness a large number of such cells to operate concurrently.

"It is not difficult to build a machine with hundreds of thousands or even millions of tiny processing cells which has a raw computational power that is many orders of magnitude greater than the fastest conventional machines. The problem lies in how to couple the raw power with the applications of interest, how to program the hardware for the job. How do we decompose our application into thousands of parts that can be executed concurrently? How do we coordinate the activities of a million processing elements to accomplish a single task? The Connection Machine architecture was designed as an answer to these questions" (Hillis 1985, 5).

The two elements of the artificial intelligence tradition, in Hillis's view, justified such a radical approach to parallelism: the analogy with the brain and the experience of symbolic processing.

"Why do we even believe that it is possible to perform these calculations with such a high degree of concurrency? There are two reasons. First, we have the existence proof of the human brain, which manages to achieve the performance we are after with a large number of slow switching components. Second, we have many specific examples in which particular computations can be achieved with high degrees of concurrency by arranging the processing elements to match the natural structure of the data" (1985,5).

In architectural terms, the Connection Machine response to the problems of the von Neumann computer was widely seen as far more radical than most of the other parallel projects. As Charles Seitz, leader of the Cosmic Cube project at the California Institute of Technology put it, comparing the Connection Machine with his own influential project:

"I think of the Connection Machine as being a much bolder experiment in the multicomputers world, a much more radical and adventurous idea. It's a machine that is further removed from conventional machines. If you look at the multicomputer, it is after all nothing but a collection of small computers connected by an unusual communication network, and programmed in rather an unusual way" (Seitz interview).

The radical nature of the Connection Machine solution lay most obviously in the massive scale of the parallelism. The comparison between processors and neurons led to the idea of combining not sixteen or sixty-four but 65,536 processors.

Developments in microchip technology made it economically feasible to think of using such a large number of processors. The difficulty lay in coordinating them in an effective way. At one level, this was a problem of wiring the machine in such a way that the connection between any two processors was not too distant. To have a direct connection between every pair of processors among the 65,536 would have required more than two billion wires, "obviously an impractical figure" (Hillis 1988, 27). The way in which it was eventually decided to connect the processors was through what is called a Boolean n-cube (or hypercube): indeed that configuration determined the number of processors used. A hypercube is the analogue, for an arbitrary number of dimensions, of the ordinary three-dimensional cube. The latter can be seen as formed by 8 nodes, each connected to its three nearest neighbours. In a four-dimensional hypercube, each of 16 nodes is connected to its four nearest neighbours. The configuration can easily be extended by analogy, to higher dimensions.

In the Connection Machine, the processors are grouped on 4,096 chips, each combining sixteen processors together with a device for routing communications. The routing devices are then wired together according to a twelve-cube pattern, representing a twelve-dimensional rather than the usual three-dimensional cube and having 2^{12} (or 4,096) rather than 2^3 (or 8) corners. This pattern ensures that no processor is more than twelve wires away from any other, so that communication between them is relatively direct.

For Hillis, more important than the precise topology of the machine is "the model you present to the programmer":

> "I think that the most fundamental notion was what's now called data parallelism. The notion was to really allocate one processor for every piece of data. So even if you imagine having millions of processors, which we could accomplish by having virtual processors, then if those processors could have arbitrary communication between them, then it was possible to have a single program working on all of the data at once... The important thing is the model you present to the programmer, and so the model you present to the programmer in the Connection Machine is that you have an infinite pool of processors, that it is possible to make a single program that controls the behaviour of all of those, that the communications primitives are synchronous and global, and that's the important part, and then the rest is just how to implement that" (Hillis interview).

The notion of data parallelism is sometimes explained by contrasting it with control parallelism. The more common approach to parallel computing is to think of parallelism in terms of breaking up a program into more or less independent parts or subsets of instructions and then allocating one subset to each processor within the parallel computer. Each processor then follows a different set of instructions. The difficulty with this approach, in the eyes of the proponents of data parallelism, is that it is difficult to break down a program into a very large number of sub-programs. Control parallelism thus implies the use of a relatively small number of processors and therefore limits the extent to which computing power can be increased through parallelism: such computers are said to be 'coarse-grained' (as opposed to the 'fine-grained' computers such as the Connection Machine which use many processors). Proponents of the latter argue that the synchronisation of the different sub-programs in control parallelism can cause particular programming difficulties (Boghosian 1990).

In data parallelism, on the other hand, it is not different parts of the program that are assigned to the different processors, but the data. The data are divided up so that each piece of data is allocated to one processor: in a galactic dynamics simulation, for example, each star might be associated with one processor. This does not, of course, mean that the computer contains as many processors as there are stars in the universe: the high-level applications programmer deals not with real processors but with virtual processors, while the low-level system software (invisible to the applications programmer) associates the requisite number of virtual processors with each physical processor. The (physical) processors then follow a single set of instructions, each of the processors repeating the instructions the requisite number of times for each of its virtual processors (Boghosian 1990; Hillis and Steele 1986). Such machines are often referred to as SIMD (single-instruction multiple-data), in contrast with the MIMD (multiple-instruction multiple-data) architectures of control parallelism. Hillis, however, emphasises that for him the essential point is the association of one piece of data with each processor rather than the character of the instructions:

> "I think people confuse that with SIMD versus MIMD, and to me SIMD versus MIMD is another engineering decision that is not very important. It depends on technology, it depends on how many pins you can get on a chip and what clock rates are, but in 1985 the best way to implement that was SIMD hypercube connection network, but maybe it wasn't the best way, it was the way I judged to be the best way. But all of those things are not so important in the sense that technology changes. The important thing is the model for the programmer" (Hillis interview).

The notion of data parallellism was not entirely new. James Bailey points out that a very similar idea had been proposed by Lewis F. Richardson for weather forecasting in his book on Weather Prediction by Numerical Process, published in 1922 (Bailey 1989, 13). Richardson considered a system of computation involving 64,000 computers, each one dealing with data from a different part of the globe - only the computers he had in mind were human computers! Bailey suggests that the reason the idea got lost was that early electronic computers were used for ballistic calculations rather than weather forecasting:

> "The thing that caused electronic computers to be born was ballistics, the need to compute range tables for artillery shells and that only requires one processor because there is only one piece of data which is one shell coming out of the cannon. So had the urgent problem of the 1940s been weather as opposed to guns, then we don't know..." (Bailey interview).

One does not, however, have to go back as far as the 1920s to find an example of data parallelism. In the late 1960s and early 1970s, Ken Batcher had developed a Massively Parallel Processor (MPP) for image processing at NASA. There had also been the earlier Distributed Array Processor (DAP), developed by ICL in Britain from 1972 and introduced commercially in 1979. Both of these machines were based on similar ideas, but the MPP was never developed commercially and the original version of the DAP was a commercial failure, with only six systems sold in the UK and none abroad (Johnson and Durham 1986, 416); a re-engineered DAP did, however, start to sell well in the late 1980s.

The challenge of the Connection Machine idea was to go beyond the research laboratory stage and develop it into a functioning machine, to take it out into the 'real world'. At this stage new considerations arise. In themselves, the differences about the attribution and relative importance of ideas are not surprising or problematic; they merely emphasise the collective nature of scientific and technological ideas. However, movement from the research laboratory into the commercial world introduces considerations of property and finance.

3. Commercialisation

Once the ideas for the Connection Machine were fairly clear, the next stage was to think about building the machine. Hillis felt that this could not be done in a university, not even at MIT, because "a university by its nature is focussed on very small teams of people" (Hillis interview) and does not provide the right environment for a large-scale project. It appears that the Artificial Intelligence Laboratory would, in any case, have been reluctant to commit itself to such a large project since it was felt by some, such as Minsky, that the lab had been 'wrecked' by the effort put into the development of another machine, the Lambda machine (Johnson and Durham 1986, 406).

The first idea was not to set up a new company but to get an established company to build the machine. Tom Knight, who was one of the principal designers of the first prototype and shared an office with Hillis as a postgraduate at MIT, recalls:

> "About this time, which I would guess was about '81 or '82, something in there, Danny and I both felt that it was time to go off and build this machine and we politicked a good deal trying to get some companies interested, notably IBM. IBM was reasonably receptive to the situation. They were on the verge of saying "we will help you build this machine". Now, it's quite clear that that would have been a strained relationship on both sides - physical distance is always a hard factor; also the people we were dealing with in IBM, some of them were certainly true believers, but the majority of the people at IBM were of the 'Gee, somebody told me I should do this so I guess I ought to be interested in this' variety; they didn't really have conviction that this was something that was going to be important" (Knight interview).

The relation with IBM never went beyond the exploratory stage because:

> "Around this time Sheryl Handler showed up. Sheryl Handler had previously been involved in ... the start-up of a company called Genetics Institute here in Cambridge... She had come out of there with some strong ideas about what it took to run a technical organisation. And she convinced Danny that the way to approach the building of this machine was to start a company. And the carrot there was that the company could be formed in such a way that the research oriented nature of the technical people would be preserved while at the same time being able to make money" (Knight interview).

The attraction of Handler's proposal was the promise that the Connection Machine could be commercialised in a way that did not subject the work to short-term profit considerations, that provided some sort of insulation, in other words, between the pressures of the market and the research work of the scientists. To Tom Knight, who had been involved in the start-up of another company which had grown out of work at the artificial intelligence Lab, Symbolics, this promise was "less than convincing", although he admits that "in retrospect I may have been wrong about that" (Knight interview).

The key to Handler's idea was the special nature of the financing to be arranged for the company.

> "She wanted to stay away from conventional venture capital because it has a very short term horizon typically. The venture capitalists typically make their money by a company going public and then it's very difficult for the company to continue to be innovative after they've locked in to a certain niche. So she felt it would be better to raise money from

private investors who might have a longer term perspective" (Hillis interview).

Hillis was convinced by Handler's plan; Knight was not, and decided to stay at MIT:

> "My view was that this idea was important enough, on the one hand, and unlikely enough to succeed that the place to be pursuing it was in the university environment. So there was a rather traumatic split between Danny and myself surrounding the Connection Machine development at about that phase. Danny took all of his marbles and went over to Thinking Machines and I stayed here" (Knight interview).

The setting up of the new company inevitably involved the definition of property rights, that is, the patenting of the ideas behind the Connection Machine. Both the origins of the idea of combining the n-cube with the array of processors within each chip and its significance for the Connection Machine were in dispute. Tom Knight recalls:

> "It became clear very early on that the process of getting messages from one processor to another in this array was going to be a major difficulty and was going to be a prime bottleneck in terms of the performance of the machine. I at least would be hard-pressed to tell you - I think Danny would have a strong opinion on this, he would say 'I invented it!' but I would claim not to know who thought of these ideas - I would say it is much more a case of cooperative rather than any unique individual contribution. But be that as it may, what evolved was a real shift in approach which was to think of this 2-dimensional mesh of processors augmented by a faster mechanism for communicating between the processors. And that faster mechanism was to conceive of these processors not as being laid out in a 2-dimensional mesh but rather at the corners of an n-dimensional hypercube, the great advantage of a hypercube being that because of its high dimension everything is close to everything else... That means that you can get data from one place in the processor array to another much more rapidly" (Knight interview).

In Knight's view, the combination of the 2-dimensional mesh with the n-cube was crucial for the development of the Connection Machine because for him it was the only really original feature of the machine. The Connection Machine was in other respects similar to the earlier Massively Parallel Processor (MPP). In Knight's view, the only significant innovation vis-a-vis the MPP was the introduction of the n-cube connections:

> "In retrospect you could turn around and say most of this machine had been conceived of previously. It had been conceived of in the context of image processing. And when you clear away all the dust, what was left in terms of innovation here was this key idea of combining together with this very simple processor in this 2-dimensional mesh this idea of taking short cuts, of getting messages from one place in this 2-dimensional array to another by going through these intermediate steps along the corners of a hypercube. Cutting through all of the mush and all of the hype, that's the key idea" (Knight interview).

Danny Hillis's view is quite different. He makes no claim to originality in the idea, but considers the matter to be relatively unimportant:

> "What was important to me was not the hypercube, but rather to make a network that was completely programmable, that was not visible. So to me the hypercube is not an important topology, it's just a better way of making a random connection network. My own interest into it came from looking at telephone switching literature and reading Benes's book, so I really thought of it as a Benes network more than a hypercube. I think people had talked

about making hypercube interconnection networks since the 1950s. There was a machine called the Holland machine which people talked about making with the hypercube network, so it was not a new idea, it was an idea that was floating around, and in fact the network that we have is partially a hypercube, but in fact it's a hybrid between crossbars at a low level and hypercube at a high level. But to me I don't really consider that such an important part of the machine, it's just a technology decision, and it can change from time to time, so it was the best way to build it when we built it, I think, but there's nothing fundamental about it" (Hillis interview).

What might otherwise have remained as an academic dispute entered a new domain when the question of patenting arose. Knight recalls:

"Danny originally had agreed that we should patent this machine and that all of the people - at that point it was a group of maybe ten people who had been involved in various aspects of this machine, thinking about the hardware and the software here in MIT, that all of the people would have their names associated with this patent. And so Danny and Sheryl had located a patent lawyer who sat down with Danny and myself for a good two weeks almost solid on a daily basis, really understanding this machine. Both Danny and I worked pretty hard in getting across the key ideas in the machine to this fellow. That was prior to the real knock-down, drag-out relationship. The thing that really put the nail in the coffin as far I'm concerned in the relations between Danny and Thinking Machines and myself was that after the fact what happened was that Danny and Sheryl decided that they wanted to have some patents that were Thinking Machines' own, exclusively, so what they did was that they split their patent in two pieces. One piece, which I would have to honestly say was a transparently invalid patent; if you had looked at the MPP architecture it only had ideas from the MPP. The other patent was a patent on the key idea to my mind, then and now, that what you wanted to do was to be able to do this routing from one point to another, i.e. this processor array. And the decision to do that after many of us had spent many hours teaching their patent lawyer the details of this really ticked me off. So that's the history of the patenting" (Knight interview).

The purpose of recording these disputes is not to reveal personal grievances, but to suggest that disputes of this type are inherent in the process of commercialisation. The commercial exploitation of ideas inevitably involves their definition as intellectual property. This is not a simple process but involves the demarcation of an idea as being separate from the general flow of intellectual thought which characterises academic life at its best: this inevitably generates conflict.

In spite of the conflict over patents, the new company was founded in 1983, with the name of Thinking Machines. The name was significant, expressing clearly that the company was to be an artificial intelligence company. This is reflected too in a saying of Hillis's, often quoted in the publicity of the company: "I want to make a machine that can be proud of us".

The name and the artificial intelligence orientation were significant for the success of the new company. Thinking Machines was founded just at the beginning of what has been called "the age of euphoria for artificial intelligence": "after decades languishing in the wings, seen merely as the arcane pursuit of idealistic eggheads, artificial intelligence has suddenly entered the limelight" (Jenkins 1986, 113). This limelight was largely the product of the Japanese Fifth Generation Project, the ambitious programme of computer development announced by the Japanese in September 1981.

The Fifth Generation Project, which aimed to build by 1992 intelligent machines with capabilities far beyond anything planned in the United States or elsewhere, did not at first stir up a significant reaction. Attention first focussed on Japan's other major project, the National Superspeed Computer Project, which was seen as a threat to US dominance in the field of conventional supercomputers, and which was the subject of a US government report (the Lax report) published in December 1982. It was only gradually that the Fifth Generation Project, with its heavy emphasis on artificial intelligence, came to be seen as a threat to the United States. A crucial role in identifying the Fifth Generation as a threat was played by a book published by Edward Feigenbaum, a leading figure in Artificial Intelligence, and journalist Pamela McCorduck in June 1983: <u>The Fifth Generation: artificial intelligence and Japan's Computer Challenge to the World</u>. Feigenbaum and McCorduck argued that the world was at the beginning of a second computer revolution, in which symbolic computing and artificial intelligence applications would become more important than numerical computation, and that it was essential for the future prosperity and position of the United States that it should fight off the Japanese challenge in this area.

The response to the Japanese programme throughout the world was for governments to pour new money into advanced computer development and particularly into artificial intelligence research. In the United States, the government was of course ideologically opposed to direct intervention in industry:

> "What the government has done instead is to throw money in the direction of anybody who has a reasonable idea about how the next generation of computers should work. The amount of money in the 1985 federal budget for all computer research doubled from the 1982 level, to $350 million. The director of the Defense Advanced Research Projects Agency, Robert S. Cooper, has announced that his agency alone plans to equal the total committed to the Japanese Fifth Generation project, roughly $600 million. The Defense Department's Strategic Computing Initiative is dedicated to building a new array of intelligent weapons and battle management systems that can help offset the larger conventional forces of the Warsaw Pact. NASA, meanwhile, is ready to spend $100 million on a network of supercomputers that would be capable of processing in parallel, yielding peak execution rates in the neighbourhood of several billion operations per second. The National Science Foundation also is a substantial patron of computer research, spending somewhere in the neighborhood of $70 million. And the Department of Energy, while not heavily involved in computer research, is the world's largest buyer of supercomputers" (Jenkins 1986, 25).

It was into this friendly environment that Thinking Machines was born. Parallel computing and artificial intelligence in particular had been identified as a key technology worthy of government support on a vast scale, and had thus also become an attractive commercial investment. Under these circumstances, Sheryl Handler was able to fulfill her promise of finding money from investors with a long-term perspective:

> "I think we were very lucky in funding. Our funding came from two sources, both of which were extremely farsighted, I think, and they were very good funding sources in that they didn't push us in directions. One of them was the initial set-up of private investors, and they were all people who

> really believed in long-term returns. If they didn't, they never would have invested in us because we told them we might not make a profit in five years. It was only people who would invest with that in mind... People like Bill Paley, Frank Stanton were very imporant in the beginning, Jim Cullen. They were people who said 'Go ahead, take the time that you need to get things right', and I think that's really unusual for investors. The other set of funding that we had came from DARPA" (Hillis interview).

The existence of the second source of funding, DARPA, was presumably crucial in providing an environment which was sufficiently secure to provide a basis for the patience of the private investors. DARPA, the Defense Advanced Research Projects Agency, "Lord DARPA", the "monarch ... with the largest coffers ever seen in the kingdom of the sciences" (Papert 1988, 3), was very important for the success of Thinking Machines:

> "DARPA had a vision that parallel computers should become possible now. And they really saw that before it was going to happen and said 'we think it's important that the US should have a lead in this area, and so we're going to get a big programme together and fund a lot of innovative projects - not only fund but also try to cross-fertilise information between the projects and provide a community for people who are working on this kind of thing'. And so they gave us the initial funding to build the very first machine; before the machine ever existed they agreed to purchase it from us; and that was very important because it gave us an initial customer... And they were in a sense a customer that was willing to stick their neck out and say 'we want parallel computers' before parallel computers existed" (Hillis interview).

DARPA, established as ARPA in 1958, had a tradition of supporting research in artificial intelligence: it had in fact helped to establish the Artificial Intelligence Laboratory at MIT in 1963, and the original work on the Connection Machine at MIT had been supported by a DARPA grant. As Jack Schwartz of New York University explains:

> "Partly because the various funding agencies were seeking well-defined roles for themselves, so they wouldn't step on each others' toes, DARPA decided to specialise in what they thought of as artificial intelligence machines or symbolic computation; which left the Department of Energy, that was the other big player, in a more comfortable position to specialise as they thought of it in numerical machines. So there was a felt difference ... between these two classes of machines. That whole conceptual picture was influenced by the fact that the Japanese had thought of their machine, their fifth generation machine, as some sort of new artificial intelligence machine and everything that they published emphasised that as an issue" (Schwartz interview).

In October 1983, in the context of the reaction to Japan's Fifth Generation project, DARPA launched the Strategic Computing Program, designed to achieve dramatic advances in the fields of microelectronics, computer architectures and artifical intelligence. The artificial intelligence technology was to be tested on three prototype military systems: an autonomous land vehicle, a pilot's associate and a battle management system (Miller and Walker 1987, 58). It was from this programme that Thinking Machines received funds to support the construction of the Connection Machine. DARPA support was not limited to a grant; as Hillis indicates, DARPA was also the purchaser of the first machine. Not only that, but the first commercial purchaser of a Connection Machine, the Perkin Elmer Corporation, bought a couple of machines for research in its optical laboratory with money provided by DARPA (Durham and Johnson, 1986, 407;

Tucker and Robertson 1988, 28); other Connection Machine purchases have also been subsidised by DARPA.

The role of DARPA in supporting the Connection Machine makes it clear that the move from the university to the outside world was not a move to a blind 'free' market. Technological development in advanced computing is very much influenced by state funding. The transition from a university project to a commercial venture depends for its success not simply on the invisible hand of the market, but also on winning the support of state agencies such as DARPA. In practice DARPA does not stand outside the computing community. It operates on the basis of a revolving door, with people constantly moving between government, industry and the universities. In this respect, the decision of Thinking Machines to establish a network of corporate fellows certainly helped to establish its credibility in this community. The corporate fellows are distinguished scientists who were brought in from the very beginning to be associated with the development of the new company, people such as Marvin Minsky of MIT, Richard Feynman, the Nobel Prize winning physicist, and Jack Schwartz, a leading computing scientist from New York University. Their association with Thinking Machines not only helped to enrich the new company in very practical ways (for example, in exploring the use of the Connection Machine for advanced scientific applications), but also helped to establish its credibility in the scientific community.

Thinking Machines' application to DARPA for funding came just at a time when a change in policy was taking place. DARPA had traditionally enjoyed great freedom in the way that it dealt with applications for funding, but Congress now insisted that any contract awarded by DARPA would have to be on the basis of competitive procurement. This meant that, on receiving an application, DARPA would now have to invite other bids in the same area. From the point of view of the applicant, this obviously meant a much slower, more bureaucratic procedure, especially when the scheme was first introduced and the procedures were unfamiliar. This was of significance in this case because, at about the same time as Thinking Machines made its application for funding, Tom Knight and the other people who had stayed at MIT also applied to DARPA for money to carry out a joint project with General Electric to construct a similar, though by no means identical, machine. Thinking Machines succeeded in having its application considered under the old rules:

> "The contract that Thinking Machines submitted was, I believe, if not the last, very close to the very last contract which was approved under the old rules. It was a non-competitive award, or if it was competitive, it was done in a very expeditious way. The MIT proposal was one of the very first which was submitted under the new set of rules. Now, DARPA in its entire history had never done a competitive procurement before and it took them literally, without any joke, two years to figure out how to do one... It was not a six-month event, that was a two-and-a-half or three year sequence which we went through, with them saying, on the one hand, 'This is a great proposal, we really want you to go ahead and do this', and on the other hand saying, 'No, there isn't money to support it'. Most of that was bureaucratic delay, some of it was not. I think that some of it had to do with the perception that Thinking Machines deserved a chance to get out there and establish a market" (Knight interview).

3. From CM1 to CM2:

On the face of it, Sheryl Handler had succeeded in fulfilling her seemingly impossible promise: to establish a commercial company insulated from commercial pressures. As one reporter put it:

> "Born of an uncommon union between business people and professors, Thinking Machines hovers delicately between the commercial and the academic. Profits and market share are not the common topic of conversation here. Indeed, broach such subjects too often and you're apt to draw raised eyebrows that suggest you're missing the whole point of what the company's about" (Porter 1987, 1).

The aims of the company were presented not in terms of sales and profits, but in terms of quality:

> "According to Handler, the company's goal is not to sell X amount of machines by a certain date. Rather, she says, 'My goal over the next two years is to have a customer base that believes we are the best company they've ever worked with. Period. That's my goal" (Porter 1987, 3).

The Connection Machine was seen very much as an artificial intelligence computer, "a vehicle for Artifical Intelligence applications" (Steele interview). This was reflected, for example, in the fact that there was no floating point capability incorporated into the hardware: floating point arithmetic, which is central to numerical computation, was done in software microcode. All the original programming was done in Lisp, the dominant language of the artificial intelligence community. It was expected that the machine would be used for artificial intelligence applications and that, consequently, the users would have little difficulty in using Lisp. The new computer was not presented by the company as a supercomputer, a competitor with the Cray machines, but as a computer of a new type: the flexibility resulting from its hypercube connections was seen as more important than the speed of the machine (Porter 1987, 3).

From the first version of the Connection Machine to the second version, the CM2, upon which work was begun in the spring of 1986, there were significant changes. The most obvious change was the addition of a floating point capability to the hardware: it was built into the chips used in the construction of the machine. This was just part of a general shift in the orientation of the machine away from the focus on artificial intelligence and symbolic computation to more general applications involving numerical computation. The shift in direction did not mean that Thinking Machines abandoned artificial intelligence applications. As Hillis points out, "many of the interesting applications of the Connection Machine do not involve numbers" (Hillis 1988, 29). Data parallelism has, for example, proved to be particularly suited to very rapid data retrieval systems, the most striking commercial venture being the Dow Jones document retrieval system, DowQuest (Waltz 1987; Dowline 1989). Nevertheless, the Connection Machine is now presented more as a general purpose machine, a supercomputer which can perform certain computations faster than the fastest Cray machine. In terms of programming languages, there has been a shift of emphasis away from Lisp to the mainstream languages of C and especially Fortran.

Why did this change in direction take place? How is the shift in the direction of computer development to be explained? The participants themselves present the change in a number of different ways.

The explanation that fits best with the image which the corporation projects of itself is that which emphasises the company's capacity to learn. This argument is expressed clearly by Hillis:

> "I think the important thing in this business is to be able to learn. Nobody ever comes up with the perfect idea, and I think maybe we've done better not because we had a better idea necessarily - though I think we had some good ideas - but I think we were very good about learning and understanding how people were using the machine and feeding that back into the design... In the beginning you had to program Lisp to use the machine, the initial users tended to be in some way related to artificial intelligence or visual understanding, the very first ones. But then we got some really inventive users who said 'we can use these machines to do what is done on conventional supercomputers, things like fluid flow, partial differential equations, big physics calculations, mechanical stress'. And really the customers drew us into that area. We had some very important lead customers that pulled us, people like United Technologies, NASA Ames, Los Alamos. They were very important in pulling us in that direction, Lockheed is another very important one. And they really had a different image of what they wanted to do with the machine, and in fact were putting in the creative effort to make it possible, and that helped us... That's an example of where they pulled us in a different direction... we would never have gone in that direction ouselves but once we realised they were succeeding we really did everything we could to support them and help them succeed" (Hillis interview).

A second way of presenting the reasons for the change in direction of the company is simply in terms of market forces: either in terms of the limited nature of the market - "artificial intelligence as marketing focus has turned out not to be a sufficiently large market to build a company on" (Palmer interview) - or in terms of a shift in market preferences: "I think the transition from emphasis on symbolic processing to numeric processing was just a matter of survival. The market place got fed up with the overselling of artificial intelligence and so companies like Thinking Machines, in order to survive, had to change directions" (Allen interview).

The two explanations are not necessarily incompatible. When asked whether the change in orientation had been due to commercial pressure rather than simply a process of learning, Hillis replied:

> "I think the two are very closely related because I think the company that will be most successful in this area will be the company that learns the best, because right now most of the interesting work is being done not by the manufacturers of computers but by the users of computers, so the company that can understand what the users are learning and incorporate that into the design will be much more successful than a company that tries to invent everything itself, so I think that in an area that's changing so rapidly there is a very strong commercial motivation to learn" (Hillis interview).

For Hillis, then, the market is educational: the successful producer is the one who learns about the needs of the users. In retrospect, he sees the move out of the university environment as having been beneficial precisely for that reason: that sort of learning is easier in a company than in the university (Hillis interview).

Hillis' account of the change as a cooperative process of learning reflects one aspect of the process, but it understates the pressure on Thinking Machines to learn in one particular direction. At a time when many artificial intelligence-oriented companies that had sprung up in the 'age of euphoria' were going out of business, long-term survival dictated that Thinking Machines transform both the presentation of its product, and indeed the product itself.

There is another issue here: the development of the technology of the Connection Machine suggests that market pressures tend to impose a certain conformism on technological development. One expert observer of the development of parallel computers, Paul Messina of Caltech, suggests that it is the capacity to perform rapid numerical computation that is decisive for the commercial success of computers:

> "The transputer [an Inmos chip designed specifically for parallel processing] is a very nice invention. One of the reasons it has not been so successful until now is because the first few years the transputer could do almost no numerical computations, that is, it was very, very slow for numerical computations and that's still such an important field for computers that when you bring out a new machine and it can't do that very well, it doesn't attract as many people as it could. The Connection Machine, by the way, is a good example of that, because the first version of the Connection Machine was not oriented towards numerical computations at all, and then people started realising... found it was very promising, and then these floating point chips were added on to it and made it a very powerful numerical machine, and now the whole company has moved away from emphasising the non-numerical applications to numerical ones. It's just an incredible change... They found that even though they'd produced this wonderfully powerful symbolic machine, it wasn't selling. There just aren't very many customers" (Messina interview).

The case of programming languages provides a specific illustration of the conformist forces at work. Fortran has long been criticised in academic circles as a language which does not promote rigour in programming (cf. Peláez 1988). It remains, however, by far the most widely used language in scientific programming practice, so there are strong commercial reasons for companies to provide software written in Fortran. The original software for the Connection Machine was all written in Lisp, but by the spring of 1987, Computing magazine reported that "some programs are now being written in Fortran, a surprising development for a machine that came out of the Lisp-dominated world of the MIT artificial intelligence laboratory" (Durham 1987, 21). Hillis sees the shift to Fortran as a good example of the learning process which the company has undergone, again emphasising the difference in perception between the university and the commercial company:

> "In the beginning I was very much against Fortran, because I was familiar with a set of problems for which Fortran was really not satisfactory. For the problems I was interested in, Fortran was not the solution, but I think once we started selling the machines and looking at the problems people were actually doing, two things became clear. One of them is that Fortran is very well organised for doing a certain class

> of problem, so there is a set of problems that Fortran is better for than Lisp. The other thing is that you look at the broader problem that a customer has, whether they're necessarily using the optimal language or not, that's not necessarily the important thing. The important thing is that they can capitalise on their basic personnel and use that resource. It's more important for them to optimise their use of personnel than it is for them to optimise their use of programming languages. And so in that sense, when you look at the bigger picture, Fortran makes a lot of sense for those, because they always have people trained in the use of Fortran and used to thinking about that... I see that very differently from the standpoint of a corporation than I saw that when I was at MIT as a graduate student, because at MIT as a graduate student I think I was focused on a much narrower set of issues. If you take the narrower issue, you come to the conclusion that the people who are using Fortran are being foolish. If you look at the broader issue, you realise in fact they're not being foolish, they're making good use of their resources" (Hillis interview).

Jack Schwartz, who, as a corporate fellow, had some influence on the shift to Fortran, comments:

> "Most of the people at the company came from the MIT background and they were used to Lisp... But I sat down one summer and used the Lisp and became clear enough that it was not doing very much for you in the context, that the thing would be just as clear or clearer in Fortran. And also it was clear that to the extent that the market would be a scientific market the customers would be more comfortable with Fortran... It is not that Fortran is better, but it's adequate and there is no reason to change things. It's the simple old proverb: 'if it's not broken don't fix it'. But that is just a matter of perceiving the market a bit more plainly." (Schwartz interview).

Guy Steele, who supervised the development of the software for the Connection Machine, and who is often seen as being very strongly identified with the Lisp language, tells a similar story:

> "Our first software was all written in Lisp and we did all of our development work on symbolic work stations. We still use symbolic work stations. Our microcode is still written in Lisp. There is no reason to take that outside the company. It's a very flexible development environment. On the other hand, we are now selling Connection Machines to a market that is used to working with Fortran and that's their tool of choice and most comfortable for them and we're willing to work with that so I guess we've had to be flexible" (Steele interview).

The experience of the Connection Machine in practice has even led Steele to suggest that Lisp may not be a very good language for parallel programming.

In general, the experience of the commercial development of the Connection Machine has led people in Thinking Machines to the conclusion that many of the features which were originally considered to be special features of the Connection Machine were not really so special after all. This is the case, for example, with the distinction between symbolic and numerical computing, which was previously considered to be of central importance. Thus, Hillis comments:

> "There was a feeling there was something special about computers for symbolic calculations as opposed to computers for floating point. I think in retrospect that turned out not to be true" (Hillis interview).

At about the same time, DARPA, ever present in significant shifts in computer technology, reached the same conclusion. Under the direction of Steven Squires, they abandoned the original emphasis on parallel artificial intelligence machines and started to promote a more general purpose approach: "without there ever having been any formal announcement of a technical shift, there really was a technical shift which I think made for a much sounder program" (Schwartz interview).

The shift away from the artificial intelligence emphasis in Thinking Machines and in DARPA coincided with a more general disillusionment with artificial intelligence. The 'age of euphoria', gave way to a more sober appreciation of the area:

> "We're seeing the reaction to the overselling of artificial intelligence five years ago. There's been a lot of people who predicted it and it has happened. A lot of wild promises were made. Expert systems were going to be the be-all and end-all and solve everyone's problems and they didn't and they can't and there were a lot of disappointments, so I think that the basic issue was just the overselling of artificial intelligence and the fact that now we're seeing the natural reaction in the market place to it. So, companies that were AI-orientated, I mean a lot of companies, have really had to scramble to stay alive" (Allen interview).

The artificial intelligence euphoria is seen now to have been a 'craze':

> "Of course it's not always the case in science that people are talking about reality. It's like everything else, science has its crazes, especially in new subjects like computer science. The physicists also have their crazes, it's a little bit more stable, but computer science as a new subject is exposed" (Schwartz interview).

What is not explained in this account, however, is how we know that the present trend to deny the importance of the distinction between symbolic and numerical computing is less of a 'craze' than the previous trend.

Once the Connection Machine began to be seen less as a specifically artificial intelligence machine, it became more acceptable to refer to it as a 'supercomputer'. In the earlier days, the term 'supercomputer' had been avoided, but as speed of performance in numerical computation became more central in defining the achievement of the Connection Machine, the company's publicity began to use the term more and more frequently. There is, however, still some ambivalence about the use of the term, which in the past has been associated with vector computers such as the Cray machines:

> "We've gone back and forth over that one, and there were times when we were very careful not to talk about ourselves as supercomputers, times when we do. Any time there's a technology shift, there's a question 'does the name go?' In one sense I think that what we are doing is a generation beyond vector supercomputers. The question is, should the name stick with them, or should it go forward into the next thing, and I don't have a strong opinion about that" (Hillis interview).

Whatever term is used, it is accepted in practice that the main competitors are the 'traditional' supercomputers:

> "It's not a word we are comfortable with but the definition of supercomputer is, whatever is the fastest computer in the world at a moment in time, and that's us so we can't say we're not a supercomputer but, so we accept that terminology. We think that people use our computers because they do the job, not because they're parallel and people do not buy our machines because they are parallel and try and figure out what to do with them, they buy them because they do the job and they happen to be parallel"

(Bailey interview).

The crucial question for the future of the Connection Machine (and for massive parallelism in general) is now seen not as being whether it can 'think' or 'be proud', but whether it can perform numerical computations faster than the Cray. In this sense, Thinking Machines proved that it had truly emerged from the university and into the real world when, in 1989, a Connection Machine out-performed a Cray in tests conducted at Los Alamos. For Paul Messina, this is a turning-point in the development of parallel computing:

> "I see it as a turning point in the sense that if five years or ten years from now we look back on 1989, we might be able to say, 'Yes, this was the year in which people first demonstrated without question that parallel computers could beat traditional supercomputers for real applications, not just for little toy demonstration programs, but for real applications that created new science', and that kind of a demonstration can make the difference in a field like this, because then the ambitious and aggressive scientists are going to say 'Aha, that's a better tool for my work'. They don't care whether it's a cute architecture, or interesting architecture, they want a better tool for their work, that's the way they should be. The people who do the biggest computations at Los Alamos have until this year been completely uninterested in parallel computers, and for a good reason, because the parallel computers had been very slow compared to the big Crays. For people at Los Alamos generally it doesn't matter that a Cray costs $20 million, because if it does the job, it's so much cheaper than anything else, and besides right now, we can't do atmospheric testing of nuclear weapons, so you have to do testing of weapons inside a computer. If you can do underground..., a typical test costs $50 million - one test - so if you buy a computer for 20 or 30 million dollars and you use it for a couple of years and you can do ten tests, it's much cheaper, so for those people, money didn't matter: capability mattered, and Crays had the capability, not as much as they wanted, but they could do something. None of the parallel machines could do it. Yet, when the Connection Machine was brought in, although they were sceptical, they said, 'well, let's try this Connection Machine, potentially it's more powerful than the Cray'. And this fellow, Harold Treece, was able to get the same performance as the biggest Cray available right now without doing very much work at all. When people see that, they say 'Ah, there's really something there. It isn't just a bunch of computer scientists and mathematicians who like to play with new architecture because it's fun'" (Messina interview).

The same feeling is echoed by Melvyn Ciment and Thomas Weber of the National Science Foundation:

> "Parallel machines are all of a sudden starting to become interesting. If you have a 16-node NCube or something like this, it's not very interesting, but when it gets up to being 512 or 1000 nodes, then they're competitive with the largest supercomputers, that's when they start getting interesting, that's when our scientists will say 'Oh gee, I could really do a problem on that machine I can't do on the Cray', and I think that's why we're just starting to get into parallelism in a big way, because the machines are just starting to become available that will really give us greater factors than you can get off a traditional supercomputer" (Interview with Ciment and Weber).

Even though it does not 'think', Thinking Machines' computer does have claims to be the fastest computer in the world in certain types of application.

4. Conclusion

This paper has not attempted to try to present a full history or appreciation of the Connection Machine. There are many issues that have been omitted: the implications of the projected construction of the TeraOps machine and the contribution of massive parallelism to changing conceptions of scientific research, the so-called 'new science', to mention just two.

The aim has been to focus rather on the transition of the Connection Machine from a university project to a commercial undertaking competing with the most advanced computers in the world. The movement from university to industry is often discussed in terms of 'technology transfer'. As this paper has shown, the original conception of the Connection Machine as a university project went through fundamental changes once it became a commercial venture. The term 'transfer' suggests that the technology is moved unchanged from one place to another. The experience of the Connection Machine suggests, however, that the move cannot take place without major changes in the technology involved, and that the shift from university to industry inevitably involves not a *transfer* but a *transformation* of technology.

Whether the technological transformation in this case was a positive one is not at issue, but the history of the Connection Machine suggests that the move from university to the 'real world' can often be a de-radicalising move. Confrontation with the 'real world' and particularly with the necessity of finding customers on the market, can force a modification of conceptions once felt to be of central importance. The question of programming is particularly important in this respect. For thirty years or more, academics have been criticising the use of Fortran in scientific programming, but attempts to introduce different languages, from Algol in the early 1960s to Lisp in the present case, have always been confronted by the same obstacle: existing programs are written predominantly in Fortran, and existing programmers are trained and experienced in the use of Fortran (Peláez 1988). This raises questions about the 'revolutionary' nature of the 'parallel revolution'. Although it is clear that many of the concepts in the architecture of parallel computers mark a radical break with the 'von Neumann' design, there are strong forces deriving from existing programs, existing programmers and existing patterns of use that impose a certain conformity and continuity on the development of computers.

From these considerations on the university and industry, it is clear that technological development cannot be considered in abstraction from the environment in which it takes place. The history of the development of massive parallelism by Thinking Machines makes it clear that the technological development can be understood only when seen as part of the broader social context of market relations and relations to the state.

Bibliography

Bailey, J. (1989), "The Ghosts of Computers Past: understanding data parallel architecture in the context of the human computing era", in Simon, H.D. *Scientific Applications of the Connection Machine*, World Scientific Publishers, Singapore, New Jersey.

Boghosian, B.M. (1990), "Computational Physics on the Connection Machine", *Computers in Physics*, Jan/Feb.

Dowline (1989), The Magazine of Dow Jones News/ Retrieval, Second Quarter.

Durham, T. (1987), "The Parallel Path to a Thinking Machine", *Computing*, 23 April.

Feigenbaum and McCorduck (1983), *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*, Addison-Wesley, Reading, MA.

Hillis, D. (1985), *The Connection Machine*, MIT Press, Cambridge, Ma.

Hillis, D. and Steele, G. (1986), "Data Parallel Algorithms", *Communications of the ACM*, Vol 29, no. 12, December.

Hillis, D. (1988), "The Connection Machine", *Scientific American*, Special Issue, Vol. 1, *Trends in Computing*.

Jenkins, R.A. (1986), *Supercomputers of Today and Tomorrow*, TAB Books, USA.

Johnson, T. and Durham, T. (1986), *Parallel Processing: The Challenge of the New Computer Architecture,* Ovum Ltd., London.

Lax P. D. (1982), *Report of the Panel on Large Scale Computing in Science and Engineering*, National Science Fundation, Washington, D.C., December.

McCorduck, P. (1988), "Artificial Intelligence: An Aperçu", in Graubard, S.R. (ed), *The Artificial Intelligence Debate*, MIT Press, Cambridge, Ma.

Miller, R. and Walker, T. (1987), *Parallel Processing: The Technology of the Fifth Generation Computers*, SEAI Technical Publication, USA.

Papert, S. (1988), "One AI or Many?", in Graubard, S.R. (ed), *The Artificial Intelligence Debate*, MIT Press, Cambridge, Ma.

Peláez, E. (1988), *A Gift from Pandora's Box: The Software Crisis*, PhD thesis, University of Edinburgh.

Peláez, E. (1990), "Parallelism and the Crisis of von Neumann Computing", Technology in Society, Vol. 12.

Porter, S. (1987), "Thinking Machines Redefines Computing", *Computer Graphics World.*

Richardson, L. (1922), *Weather Prediction by Numerical Process*, Dover Publications, New York,(reprint of 1922 original).

Supercomputing Review (1990), "Policies and Politics", January.

Thinking Machines Corporation (1989), *News Release*, Nov. 28.

Tucker, L. and Robertson, G. (1988), "Architecture and Applications of the Connection Machine", *Computer*, August

Waltz, D. (1987), "Applications of the Connection Machine", *Computer*, January.

Interviews:

A lot of this work was based on interviews, and I wish to thank the following people: Don Allen, James Bailey, Daniel Hillis, John Palmer and Guy Steele of Thinking Machines Corporation; Tom Knight of MIT AI Laboratory; Melvyn Ciment and Thomas Weber of the National

Science Foundation; Paul Messina and Charles Seitz of CalTech; and Jack Schwartz of New York University.

Edinburgh PICT Working Paper Series

1. *Edge, David,* The social shaping of technology.
2. *MacKenzie, Donald,* 'Micro' versus 'macro' sociologies of science and technology.
3. *Williams, Robin and Russell, Stewart,* Opening the black box and closing it behind you : on microsociology in the social analysis of technology.
4. *Fleck, James,* Innofusion or diffusation? The nature of technological development in robotics.
5. *Peláez, Eloína,* Parallelism and the crisis of von Neumann computing.
6. *MacKenzie, Donald,* Parallel computing : some issues for social research.
7. *Williams, Robin,* The development of models of technology and work organisation with information and communication technologies.
8. *Webster, Juliet,* New technology, old jobs : secretarial labour in automated offices.
9. *Fleck, James,* The development of information-integration : beyond CIM?
10. *Peláez, Eloína,* What shapes software development?
11. *Molina, Alfonso,* Information technology : anatomy of a successful European collaboration.
12. *Webster, Juliet,* Gender, paid work and information technology.
13. *MacKenzie, Donald,* The influence of the Los Alamos and Livermore National Laboratories on the development of supercomputing.
14. *Fleck, James, Webster, Juliet and Williams, Robin,* The dynamics of implementation : a reassessment of paradigms and trajectories of development.
15. *Rooney, Brendan,* The implementation of CIM in rubber manufacturing : a flexible response : case study of Avon Rubber.
16. *Fransman, Martin,* Cooperation, competition and international competitiveness : the case of Japanese central office switches.
17. *Webster, Juliet,* The shaping of software systems in manufacturing: issues in the generation and implementation of network technologies in British industries.

ORDER FORM OVERLEAF

18. *Oakley, Brian,* Look back in Alvey: why support for R&D is not enough.

19. *Molina, Alfonso,* The development of public switching systems in the UK and Sweden: the weight of history.

20. *Fransman, Martin,* Controlled competition in the Japanese telecommunications equipment industry: the case of central office switches.

21. *Webster, Juliet,* Automation in the social office: women's skills and new technology.

22. *Tierney, Margaret & Williams, Robin,* Issues in the blackboxing of information technologies.

23. *Peláez, Eloína,* From symbolic to numerical computing: the story of thinking machines

24. *Molina, Alfonso,* Building up a neural network sociotechnical constituency: a contribution to the formulation of a UK strategy.

25. *Tierney, Margaret,* The formation and fragmentation of computing as an occupation: a review.

26. *Howells, John & Hine, James,* Competitive strategy and the implementation of a new network technology: the case of EFTPoS.

27. *MacKenzie, Donald,* Economic and sociological explanations of technical change.

Please supply .... copy(ies) of **Edinburgh PICT Working Paper(s)** Nos.................

at a cost of £5 per copy inclusive of postage.

I enclose a cheque for ....

Please make cheques payable to '**The University of Edinburgh**' and send to
Mrs Barbara Silander, RCSS, The University of Edinburgh, 56 George Square,
Edinburgh, EH8 9JU

Edinburgh PICT Research Report Series

1. *Molina, Alfonso,* The transputer constituency : building up UK/European capabilities in information technology.

Edinburgh PICT Student Paper Series

1. *Hedges, David,* A review of value added and data services and their implications for the City of Edinburgh.

2. *Hosein, D.L. Abbass,* A critical evaluation of the industrial robot : use of a strategic perspective to enhance management ability to derive corporate benefit from its implementation.

3. *Harremoes, Steffen,* Consequences of the Single European Market on main exchange manufacturers : prospects for UK producers.

Please supply.......copy(ies) of **Edinburgh PICT Research Report(s)** Nos.................

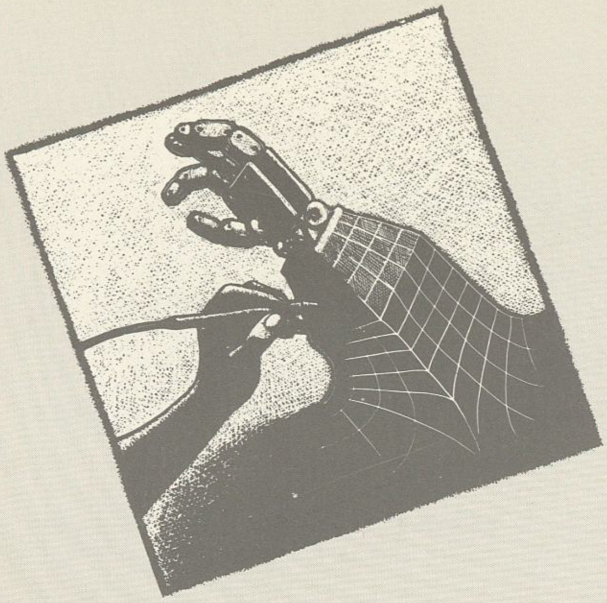
at a cost of £20 per copy inclusive of postage (reduced rate of £5 for academics).

Please supply.......copy(ies) of **Edinburgh Student Paper(s)** Nos.............................

at a cost of £5 per copy inclusive of postage.

I enclose a cheque for ....

Please make cheques payable to '**The University of Edinburgh**' and send to
Mrs Barbara Silander, RCSS, The University of Edinburgh, 56 George Square,
Edinburgh, EH8 9JU.



PICT at Edinburgh

The Programme on Information and Communication Technologies (PICT) is a major initative of the Economic and Social Research Council, which aims to explore social science perspectives on the rapidly evolving Information and Communication Technologies (ICTs) and inform policy debate in the field. The research is conducted by a network of six centres - Brunel University (CRICT); Polytechnic of Central London (CCIS); The University of Edinburgh (RCSS); UMIST (CROMTEC); University of Newcastle (CURDS); and University of Sussex (SPRU) - and coordinated from the University of Oxford.

Edinburgh PICT research is based at the Research Centre for Social Sciences and draws on expertise in the Departments of Business Studies, Economics and Sociology, as well as the Science Studies Unit. The group starts from the assumption that the development and implementation of new technologies cannot be wholly explained by technical considerations, but that complex social, political and economic factors are involved. The research effort therefore focuses on the 'social shaping' of ICTs , at the level of detailed technical design. It aims to elucidate the considerable scope which exists - for both producers and users of technology - to influence the direction and consequence of technological change. Much of the research involves building strong links with the policy community, in industry and in government.

Edinburgh PICT is part of a strong and growing base of socio-economic research on technology at the University, and runs a Doctoral Programme of Social and Economic Research on Technology within the Faculty of Social Sciences. Both teaching and research activities benefit from close links with departments in the School of Information Technology. In addition, members of the Edinburgh group collaborate with researchers in neighbouring Higher Education Institutions, and with other centres in the PICT national network.