

HOMOGENEOUS MACHINE

TECHNICAL PLAN

COMPUTER SCIENCE
CALIFORNIA INSTITUTE OF TECHNOLOGY

No.
4705

ARPA CONTRACT REPORT

Orig ned.

HOMOGENEOUS MACHINE

TECHNICAL PLAN

9 January 1982

Erik P. DeBenedictis

Charles L. Seitz

Caltech Computer Science

Copyright (C) 1982 Caltech. Copying is allowed.

The research described in this document was sponsored by the Defense Advanced Research Projects Agency, ARPA Order number 3771, and monitored by the Office of Naval Research under contract number N00014-79-C-0597.

No.
4705

This is an internal working document of the Caltech Computer Science department. Some of the ideas expressed in this document may be only partially developed or erroneous. All of the materials included are the property of Caltech and its sponsors. Distribution of this document outside the immediate working community is discouraged; publication of this document is forbidden.

Table of Contents

1. Introduction	
1.1. The Proposed Machine	1
2. Architectural Innovations	2
2.1. Independence of Processors	3
2.2. Allocation of Memory	4
3. Potential Performance of the Hardware	6
3.1. A Model of Computation	6
3.2. Cost/Performance	6
3.3. Long Range Projections	8
4. An Overview of the Implementation	9
4.1. Main Array of Processors	9
4.2. Dedicated Host	9
4.3. Mainframe Hosts	10
5. Proposed Plan of Action	11
5.1. Current Status	11
5.2. Timetable for Future Work	11
5.3. Cost Estimates	13

1. Introduction

This homogeneous machine project is a product of more than five years research in concurrent processing by two departments at Caltech. Research in concurrent computation in the computer science department started with theses by Browning [Browning 80] and Locanthi [Locanthi 80], and continues with a thesis in preparation by Dick Lang, and work by Chuck Seitz [Seitz 81]. All this research is shared an emphasis toward implementation with higher and higher density integrated circuits. The computer science department has been planning to construct a machine of the genre for several years.

Our colleagues in High Energy Physics have been plagued by a lack of suitable computing technology to solve some fundamental physics problems. In the course of our collaboration it became evident that our research had studied architectures of the sort ideal for their physics problems. Our collaboration with High Energy Physics has caused us to select some particular versions of the architectures that we have been studying as the most likely to be useful. Given our theoretical interest in the architectures and the practical use sought by High Energy Physics we have decided to act now to construct a homogeneous machine.

This document describes the plan of the computer science department. The plan of High Energy Physics is described in [HEP 81].

1.1. The Proposed Machine

The homogeneous machine proposed here is a hypercube machine consisting of 64 identical microprocessors interconnected in a 1, 2, 3, and 6

dimensional array¹. Each of the processors consists of about 77 chips, including a 8086 microprocessor, a 8087 floating point chip, 1/8th of a megabyte of RAM, and six bidirectional interfaces to other processors. The processors will be constructed on 64 printed circuit boards mounted in a custom backplane. The hypercube machine will consist of this array of processors and a single dedicated host processor that will control the array.

The class of problems that can be solved by such a machine is limited. There is absolutely no intention of the machine ever being able to execute a conventional program. Certain very limited classes of problems can be solved efficiently, and many of these problems are so large and important that a special purpose architecture is justified.

This paper will discuss the architecture, physical design, and some applications for a machine of this architecture.

2. Architectural Innovations

The architecture of the hypercube machine is new. Previous multiprocessors were constructed to allow direct implementation of many conventional computer programming constructs. These multiprocessors typically included special hardware to allow each processor access to the memory of others. Using the shared memory semaphores could be implemented, but only with extra hardware. The hypercube machine discards many of the

¹

All four dimensionalities can be obtained simultaneously with the proper structure.

conventional programming constructs, and the hardware to implement them.

2.1. Independence of Processors

A popular architectural direction in multiprocessor architecture has been to make a single sequential process execute faster by putting more processors onto the same memory. The results may be communicating sequential processes, as in C.mmp and CM*, or a high speed execution of a single sequential process, as in the dataflow machines of Dennis.

Tightly coupled multiprocessor architectures have a range of problems: the hardware cost grows faster than linearly with the size of the machine, and the efficiency of the software decreases as machine grows. The hardware will invariably contain a large switching network to route memory accesses or commands between arbitrary processors. A large switching network has a large parts inventory, and will operate slowly due to long wires and complex switching.

Research at Caltech indicates that architectures communicating through message passing have a brighter future than those with tight couplings or shared memory. Consider the effect of decreasing feature size on the design of a hypercube machine processor. Figure 1 shows the progress of the design from the present size of 50-77 chips/system to a one or two chip implementation a decade from now.

<u>Feature Size</u>	<u>Chip Count</u>
3 microns	50-77
1 micron	5-8
0.5 micron	1-2

Figure 1: Effect of Decreasing Feature Size on Processor Chip Count

Consider the advantages of a 1-2 chip/processor hypercube machine: 1) the processors would be very fast because of very few off chip delays, 2) chips would be interconnected very regularly and with very high density, and 3) the parts inventory would be small, i.e. one or two.

The hypercube machine connects processors with high bandwidth, but very loose connections. Each of the processors is quite small and standard, allowing maximal use of LSI 'glue' components. The compactness of the the system and short buses allows for high clock rates. In a nearest neighbor² configuration all wires are short.

2.2. Allocation of Memory

The trend in multiprocessor research is to move away from the single large computer to more and more smaller and smaller computers. This trend is almost valid because present mainframes are very much larger than an optimal computer. The trend can be followed too vigorously, however produce computers that are too small to be cost effective.

Multiprocessors have been studied where the processors are too small. Two examples are the tree machine studied at Caltech [Browning 80] and the systolic array studied at CMU [Kung 80]. Both machines use processors that, in today's technology, would be less than one chip in size. The tree machine processors are programmable, containing about 1K bytes of RAM. The systolic array processors contain no program RAM, but are effectively programmed in their internal arithmetic layout.

²

One (but only one) of the proposed network configurations.

In the process of tree machine research at Caltech, Browning and the author programmed a number of useful algorithms for the tree machine and studied their performance. A pattern was noticed in the results: tree machine algorithms tend to require as much time to load the problem into the machine and to unload the answer as is required to solve the problem.³

Estimates of the necessary size of a tree machine required to solve a useful problem tend to be large. Tree machines too small to store an entire problem would have to solve a problem in parts, swapping the parts between a secondary storage and the tree machine. The effect of swapping is to degrade performance by orders of magnitude, making that an unreasonable alternative. The only solution is to make the machine large enough to store an entire problem. With only a fraction of their 1K byte storage available for data storage an unreasonably large number of processors are required.

The reason for this phenomenon is that the processors have so little memory that they cannot perform meaningful computation for very long. The solution to this problem for the hypercube machine is to reduce the number of processors and give each processor much more memory. The speed of the machine is reduced to a more reasonable level because the processors must multiplex their computations. Since a larger portion of the machine is low

3

For example, sorting N numbers requires N steps to load the problem and N to unload the answer. Sorting is accomplished during the loading process where $\log N$ processors cooperate to load a number into the proper processor while maintaining proper order. The average number of sort operations performed by each processor is $\log N$, whereas the number of steps to load and unload the problem is $2N$.

cost memory, the cost of a machine required to perform a useful problem is reduced.

3. Potential Performance of the Hardware

Let us consider the economics of processing with an array of microprocessors operating concurrently. Another paper examines the question of whether full concurrency can ever be achieved, and also whether a large enough body of problems exist to justify constructing such a machine[HEP 81].

3.1. A Model of Computation

In this preliminary analysis we will adapt a very simple view of computation: we will assume that a problem solution requires some amount of memory, and that some number of operations are performed. Those problems that will execute efficiently on the hypercube machine will have characteristic that they can be partitioned into a multiplicity of processors. In this partitioning, each processor will have a fraction of the total memory of the problem, and will perform the same fraction of the total operations performed in the problem. An array of n processors will be equivalent to a single conventional computer with n times the memory and n times the speed.

3.2. Cost/Performance

Let us consider compare the costs of such a machine and a conventional computer. The dominant cost in the hypercube machine is the cost of a single board that contains the basic processor. Let us examine the commercial viability of a hypercube machine by estimating the market cost

of a hypercube machine and comparing its performance with competitive products.

Since the single board of the hypercube machine would be produced in such large volume, its cost would follow the same economics as semiconductor RAM systems today. We will estimate the cost market of a hypercube machine by analogy to a large RAM system.

At today's market prices semiconductor RAM costs \$15,000 per megabyte⁴. Semiconductor RAM boards usually consist of boards populated approximately 75% with 16K RAM chips. One megabyte of RAM would consist of 512 RAM chips and 170 support chips by the above model. The cost per chip is therefore \$22.

The hypercube machine described in this paper consists of a processor with 77 chips. Each processor has a 0.5 MIP performance on normal instructions, a floating point speed of 20 uS, and 1/8 mB of memory. At \$22 per chip the cost is under \$1700 per processor.

A DEC 11/780 (VAX) has a floating point speed of about 1 uS, and could reasonably support 10 mB of memory. A machine in such a configuration would cost \$400,000. If the same \$400,000 were spent on hypercube machine processors at \$1700 each, 235 could be purchased. A hypercube machine of 235 processors would have an equivalent floating performance of 0.1 uS, and 30 mB of memory.

4

These prices include power supply and backplane.

A CRAY-1 has a floating point speed of 15 nS, and costs \$15 million with a large amount of memory. To obtain the equivalent floating point performance with hypercube machine processors at 20 uS per processor, 1,333 would be required. These 1,333 processors would additionally have 166 mB of memory, much more than the CRAY-1, and would cost \$2.2 million.

3.3. Long Range Projections

These prices are conservative. For example, approximately 2/3 of the chips (but less than 1% of the transistors) in the processor are SSI/MSI chips in the interprocessor interface section. Should a large effort be made to build such machines, these chips could be reduced to 1 or 2 LSI chips. Also, the processor used is the oldest 16 bit CPU and the floating point unit is the first constructed by the industry.

As discussed previously, improving technology will continue to decrease the number of chips per processor to a limit of one or two. Since processors are so amenable to IC implementation, their price/performance will increase much more rapidly than average.

In summary, the hypercube machine being constructed at present has a potential price/performance that is about 7 times better than products available today. Even given a substantial inefficiency in software, the hypercube machine will be noticeably better and either a VAX or CRAY-1. Future improvements in microprocessor technology will drastically improve this already good situation. More efficient CPUs and floating point units, and special interprocessor communication chips should reduce price/performance by an order of magnitude.

4. An Overview of the Implementation

The hypercube machine can be divided into three parts for convenience of explanation: 1) the array of microprocessors, called the main processors, 2) the dedicated host, which controls the array and interfaces to 3) the host (or hosts), which are mainframe machines that perform compiling of code for the machine. Figure 5 is an overview of these parts and their interconnections.

4.1. Main Array of Processors

The main array is essentially a multi-dimensional array of microprocessors. With one exception, these are all identical processors that connect only among themselves in a tightly connected network. One of the processors has one extra connection, however that connects the array to the rest of the world.

All of the main processors are connected by a control bus. This bus allows sharing of functions that are same for all processors, such as clock and memory refresh. The control bus also provides a flexible but low bandwidth global communications capability for use by diagnostics and as a network-wide software debugging aid.

The processors are interconnected by fully asynchronous bidirectional connections. The hardware supports a 64 bit interprocessor message by generating interrupts only when complete messages can be input or output.

4.2. Dedicated Host

The dedicated host is the interface between the general purpose host computers and the array. The dedicated host interfaces to the array

through one asynchronous connection and is the master of the control bus. The dedicated host also interfaces to the mainframe hosts and to console terminals.

In addition to serving as a hardware interface to the array, the dedicated host fulfills an important function in some algorithms, see [HEP 81]. For this reason, the dedicated host will have a substantial amount of RAM: 512Kb-1Mb.

An unsuspected function of the dedicated host is the running of diagnostics on the entire array. The dedicated host will have the ability to control the supply voltage and clock frequency as well as control the master reset and RAM refresh rate of the entire array. These abilities will aid diagnostic programs in locating faulty boards.

The present plans are to construct the dedicated host with the same architecture as the main processors, for reasons of software compatibility. Future plans may call for more than one dedicated host, or a processor that is faster than the main processors.

4.3. Mainframe Hosts

Since neither the array nor the dedicated host will have any secondary storage, they would be inappropriate for compilers. Compiling will occur on either the HEP VAX or the CS DEC-20 and the machine code will be downloaded to the dedicated host and then to the array. At present it appears that the CS DEC-20 will be used for assembly level system software development and the HEP VAX will be used for applications programming in C.

5. Proposed Plan of Action

A project to evaluate this architecture will consist of three phases:

1. Construction of a 64 processor prototype array and dedicated host.
2. Development of system software.
3. Application of the machine to different problems.
4. Construction of a 1024 processor hypercube machine.

This document will be concerned only with items 1,2, and 4. Caltech's High Energy Physics group is eager to apply such a machine to real physics problems [HEP 81].

5.1. Current Status

Work has already begun on constructing the hypercube machine. Funds were provided in anticipation in the computer science ARPA budget for work on building a concurrent machine. These funds, amounting to a non-renewable \$20,000, are being used at present. ARPA interest in the project is considerable, but in a general atmosphere of budget cutting, funds to construct a useful hypercube machine will be difficult to obtain. Additional funding is being pursued with ARPA as well as with others.

As of January 1982 approximately 50% of the engineering has been completed. Engineering is proceeding on the remainder of the machine and will be completed before any additional funding could have an effect.

5.2. Timetable for Future Work

The only part of the machine that has not yet been funded is the

construction of the actual array . A proposed timetable for future work is shown below:

Phase I - 64 processor machine

1 September 1981

Project to build 64 processor test model of the hypercube machine begins. Hardware design and prototyping begins immediately.

1 January 1982

Working model of the main processor. Software development begins now.

1 March 1982

Design of main processor is complete and the design is submitted to a contractor for PC layout and fabrication of 64 units. A complete software model of the hypercube machine is complete, including the dedicated host and at least two main processors.

1 July 1982

Primitive system software is completed. Boards to construct a 64 processor array are delivered by the contractor. Boards are now assembled into an array and tested.

1 October 1982

64 processor system is fully operational. Programs of the approximate complexity of Laplace's equation run. A technology evaluation is performed to determine if better chips are available for any part of the system. If necessary, redesign begins.

Phase II - 1024 processor machine

1 March 1983

A potentially redesigned main processor is submitted to a contractor for construction of 1024 units. Some physics research problems should be complete by this time.

1 October 1983

1024 unit hypercube machine becomes fully operational.

Other parts of the machine are used only in quantity one, and engineering prototypes will be used.

5.3. Cost Estimates

An estimate of the cost of constructing one hypercube machine processor and building it into a processor array is shown in figure 2.

It will be noted that in figure 2 the cost is largely influenced by the \$400.00 cost of the 8087 floating point chip. As of January 1982 the 8087 chips are scheduled for delivery to suppliers on a 4-6 week basis at a retail cost of \$400. It is expected that the price of these chips will drop very significantly in the following few months.

Besides the 8087, the price of other parts is dropping rapidly now. In particular, the 8086 processor and the 64K RAM chips should be available for less than the proposed price. Figure 3 is a prediction of the actual cost of producing the array on a per unit basis. These figures only are used in the later cost estimates.

Figure 4 is a schedule of the expenses that would be required to complete the project if funding were available.

Considerable graduate student and faculty interest has been expressed in the Computer Science Department in communications software for the hypercube machine. Funding of research in this topic can proceed after the machine is in operation. To get the machine into a basic operation it will be necessary to have a primitive network operating system and diagnostics. A manpower budget for this is included in figure 4.

<u>Item</u>	<u>Quantity</u>	<u>Cost</u>	<u>Extension</u>
Double sided PC board:	1	\$50.00	\$50.00
8087 floating point:	1	\$400.00	\$400.00
8086 microprocessor:	1	\$100.00	\$100.00
64K RAM (2164):	16	\$14.00	\$224.00
8529A interrupt:	2	\$18.00	\$36.00
74S225 fifo:	12	\$4.50	\$54.00
misc IC:	many	\$40.00	\$40.00
Share power supply (2A/5V)	1/10th	\$300.00	\$30.00
Share backplane	1/64th	\$4000.00	\$63.00
sub-total:			\$997.00
Assembly and testing:		\$250.00	\$250.00
total:			\$1247.00

Figure 2: Per Processor Costs

March 1982	\$1200
March 1983	\$800

Figure 3: Estimated Cost of the Main Processor on a Per Unit Basis

	<u>Period</u>		
	1-Mar-82- 30-Sep-82	1-Oct-82- 28-Feb-83	1-Mar-83- 1-Oct-83
64 processor prototype:	\$76,800		
(64 processors at \$1200 each)			
1023 processor array:	\$819,000		
(1024 processors at \$800 each)			
Staff (quantity):			
DeBenedictis	1	1	1
Graduate Student:			
Hardware work:	1	0	1
Software work:	3	4	3
Technical:			
Hardware work:	1	1	4
Software work:	1	0	0

Figure 4: Expense Schedule for Construction of the Hypercube Machine

HOMOGENEOUS MACHINE

16

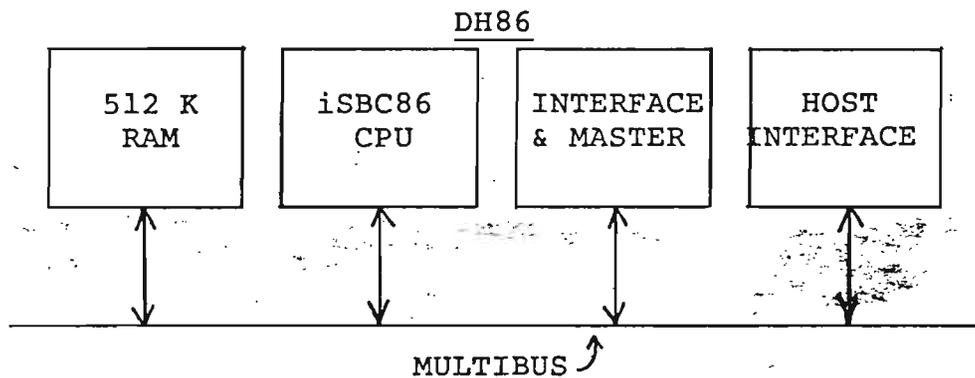
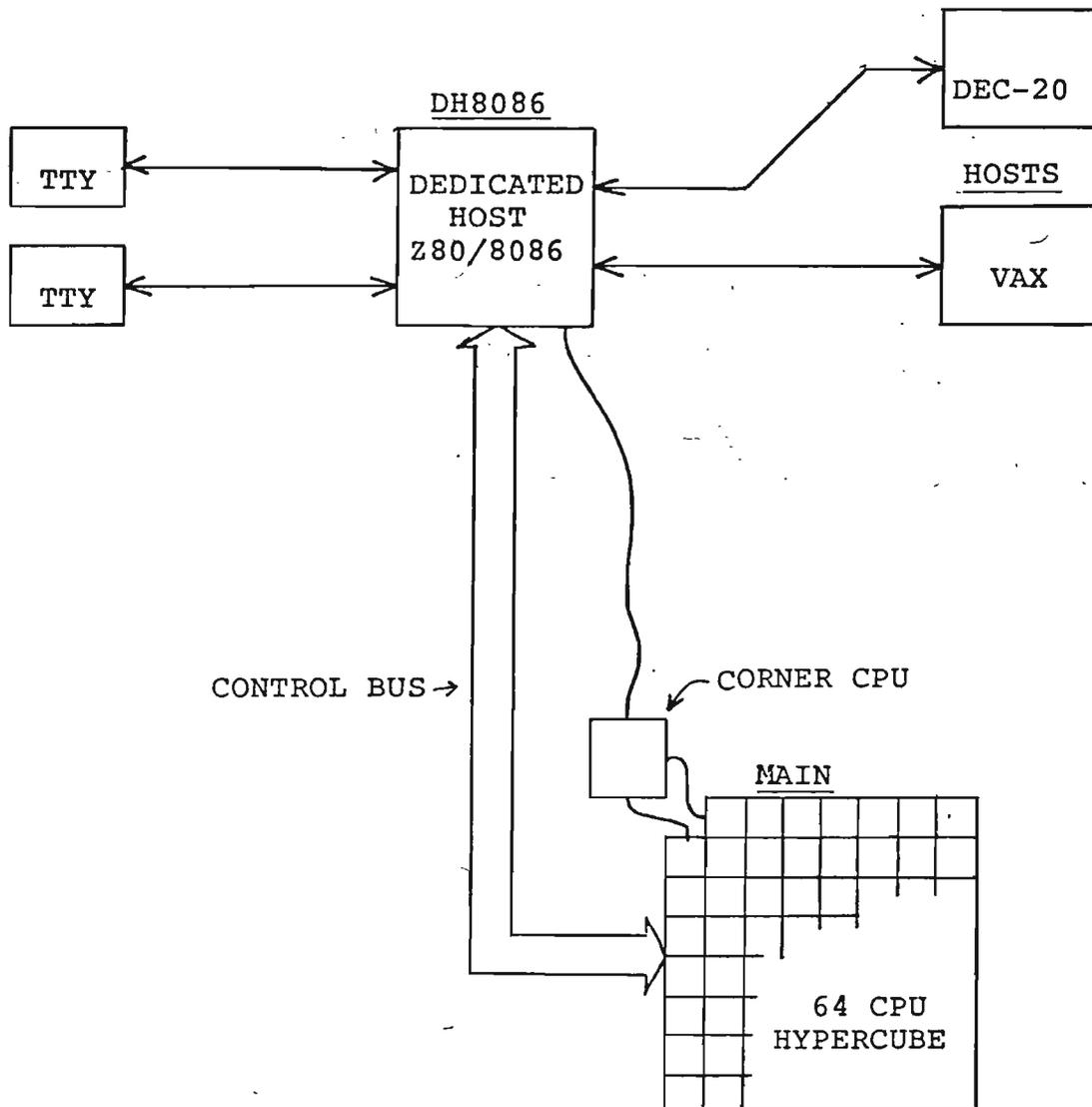


Figure 5: An Overview of the Hypercube Machine

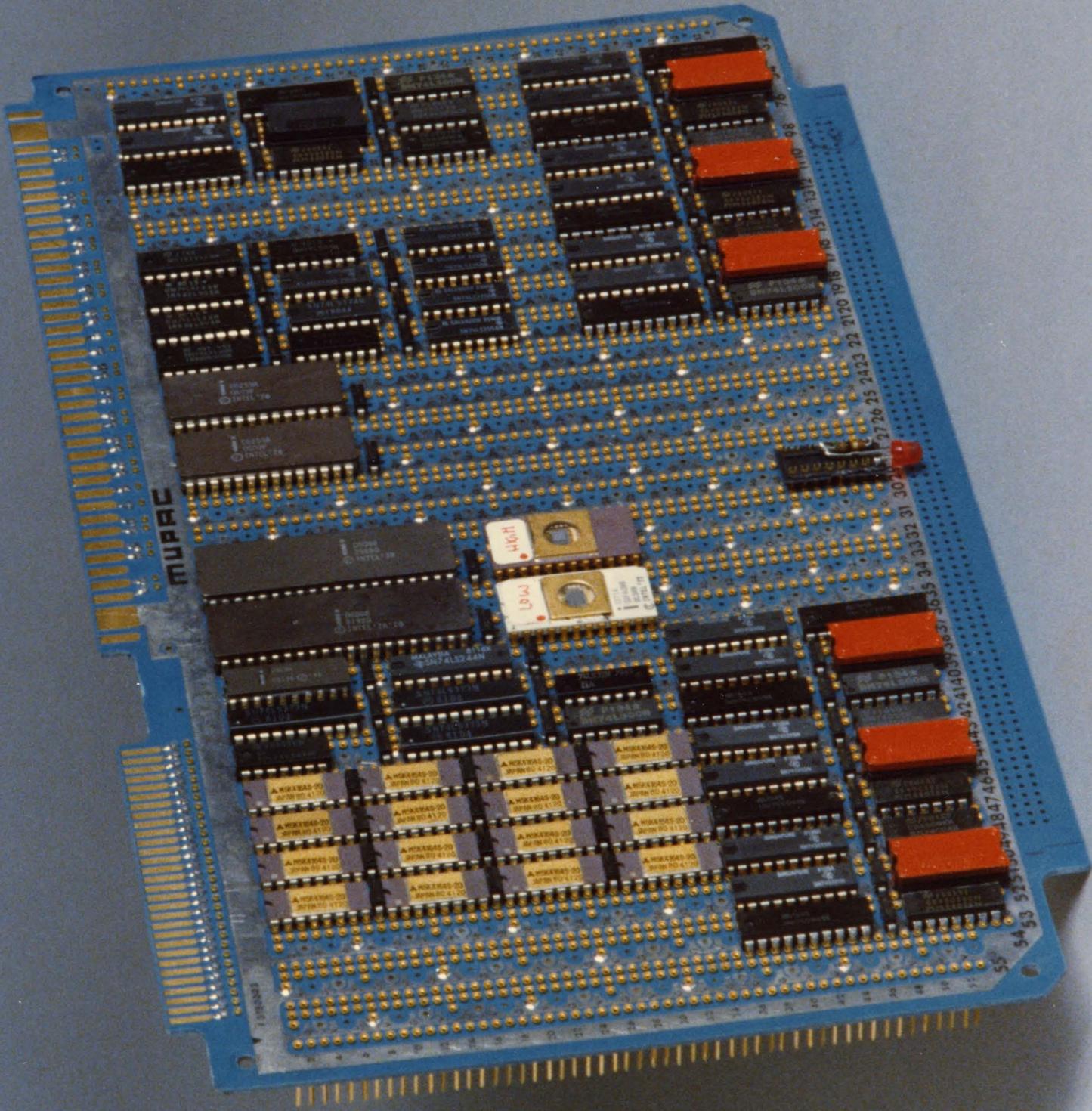


Figure 6: Engineering Prototype of the Main Processor

REFERENCES

- [Browning 80] Sally Browning, The Tree Machine: A Highly Concurrent Computing Environment, Ph.D. thesis, California Institute of Technology, 1980.
- [HEP 81] Brooks, E., Fox, G., Gupta, R., Martin, O., Otto, S., DeBenedictis, E., Nearest Neighbor Concurrent Processor, Caltech High Energy Physics, Technical Report, 1981.
- [Kung 80] Mead, C., Conway, L., Introduction to VLSI Systems, Addison-Weseley pp. 271-292, chapter 8.3, 1980. Section contributed by H. T. Kung and Charles E. Leiserson
- [Locanthi 80] Bart Locanthi, The Homogeneous Machine, Ph.D. thesis, California Institute of Technology, 1980.
- [Seitz 81] Chuck Seitz, Ensemble Architectures for VLSI -- A Survey and Taxonomy. Attached.

COMPUTER SCIENCE
CALIFORNIA INSTITUTE OF TECHNOLOGY