1986

Multi-Flex Machines Preliminary Report

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
86-612

MULTI-FLEX MACHINES
PRELIMINARY REPORT

John R. Rice

CSD-TR-612
June 1986

MULTI-FLEX MACHINES

PRELIMINARY REPORT

John R. Rice*

Computer Science Department

Purdue University

CSD-TR 612

June 25, 1986

Abstract

The multi-FLEX machines use FLEX/32 multi-computer modules for larger machines. These machines provide wide flexibility and scalability, the innovative ingredients of the design are: (a) use of locally shared memory in addition to local and global memory, (b) very high internal and external bandwidth, (c) no use of network protocols for communication, (d) distribution of external I/O throughout the machine. The 64 node FLEX-cube would have 1500 processors and 700 megawords of memory, operate at 2.5 gigaflops and have external I/O bandwidth of 4 Gigabytes/second. It is conjectured that it could service 40,000 terminals or service one job to do weather forecasts 1000 times faster than real time (covering the entire northern hemisphere with about 500 million unknowns.)

* Supported in part by Air Force Office of Scientific Research grant AFOSR-84-0385 and Army Research Office contract xxxx

I. RATIONALE FOR THE MULTI-FLEX MACHINES

This main rationale for the class of machines described here is to provide wide flexibility and scalability in applications. The FLEX-cube described can, it is conjectured, either do a weather forecast 1000 times faster than real time for the northern hemisphere or service 40,000 terminals (it is a 1500 processor, 2.5 Gigaflops machine). The innovative ingredients of this design are

a) Memory heirarchies of local, locally shared and global (there might be more than one level of locally shared memory)

b) Very high internal and intermodule communication bandwidth.

c) No use of network protocols for communication.

d) External I/O distributed throughout the machine.

The design is based on the existing commercial product, the FLEX/32 of Flexible Computer Corp. Other desirable properties of this design are a high level of modularity, fault tolerance and the use of ordinary packaging.

At this point this is entirely a "paper" machine. However, real machines of this type can be constructed quickly (given the money) using existing machines. The main barrier to the use of the multi-FLEX machines is in the software systems. However, the situation here is no better nor worse than for other designs.

II. THE FLEX/32 MODULE

The multi-FLEX machines are built with the existing FLEX/32 machines as modules. We refer to [FLEX 86] for details of this machine, its characteristics are summarized here. Figure 1 shows a block diagram of the machine. For concreteness, we

assume the following characteristics of the boards and bases:

PE #1:	4 MIPS processor (e.g. 68020) 4 Mbytes of local memory (1.5 μsec access)
PE #2→20:	4 MIPS processor 1 Mbytes of local memory (7 μsec access)
Common #1→10:	500 Kbytes of locally shared memory with hardware support for concurrent access control.
Common bus:	40 Mbytes/sec
Local bus #1→10:	20 Mbytes/sec

The FLEX/32 module may have characteristics of a standalone computer or part of it may operate in this mode. In this case, PE #1 assumes the function of control processor. The ten local busses all connect outside the module, in the multi-FLEX these will be divided into 3 groups:

(a) **Global Module Access:** Provide access to the global memory (or a higher level of locally shared memory) and to the control module (one bus).

(b) **Local External I/O:** Provides access to all types of peripherals and network facilities (disks, terminal, printers, ether nets, etc.), (three busses)

(c) **Intermodule Communication:** Provides communication with other modules of the multi-FLEX (six busses).

SCHEMATIC OF FLEX/32 ARCHITECTURE

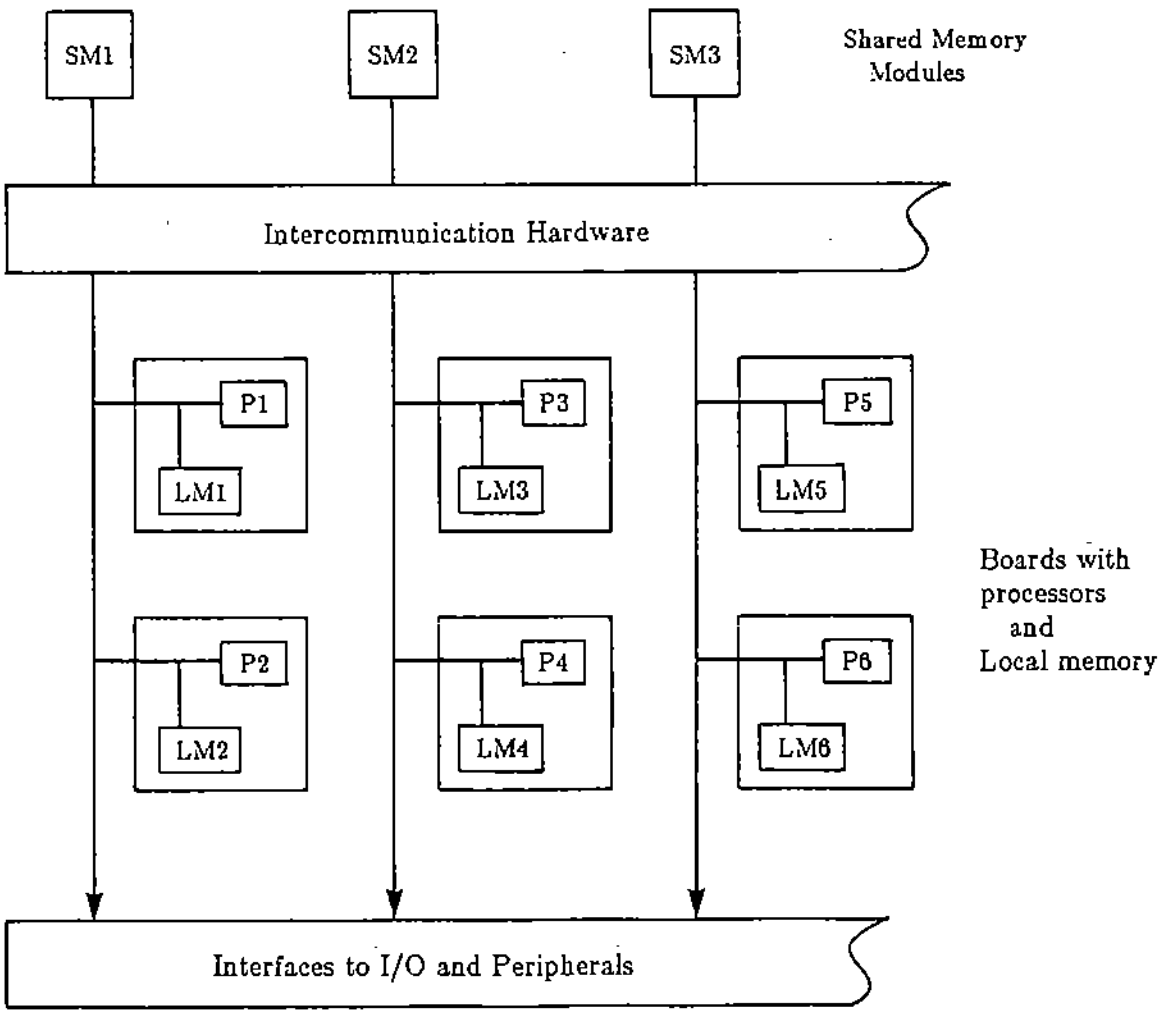

Figure 1. Block diagram of the FLEX/32 from the Flexible Computer Corp. P1, etc., denote processors; LM1, etc., denote local memories; SM1, etc., denote boards with shared (common) memory. There are ten local busses for up to twenty processor boards.

The existing FLEX/32 has two operating systems. There is an ordinary UNIX (System V with some Berkely 4.2 enhancements) that runs on one processor. Its purpose is to support software development and general use. There is also a parallel-processing system, MMOS, which supports complete parallelism within the FLEX/32 but provides minimal system services, only Fortran and C are provided in MMOS.

In order to achieve broad flexibility of the multi-FLEX machines, we assume that the following extensions of the FLEX/32 software system are made:

(a) **Multi-processor UNIX.** This is a UNIX system which uses multiple processors by assigning UNIX tasks to processors using a simple-minded allocation process (e.g. as the Sequent, Encore machines operate).

(b) **Partitioning.** The system allows a FLEX/32 module to be partitioned into two pieces, one which runs the Multiprocessor UNIX and one which runs MMOS.

III. THE MODULE INTERFACE PROCESSOR

Most of the existing multiprocessors with local memory have very slow interprocessor communications. They use message passing protocols which require one millisecond or more to initiate communications. The FLEX/32 does not have this bottleneck, but connecting many FLEX/32's together as modules can have this problem. It would be less severe than in the current hypercube topology machines because the computation power of a FLEX/32 module is much higher than a typical hypercube node. The ratio of computing power to communication delay is the more critical measure of balance in a multi-processor machine.

However, to avoid potential problems, we hypothesize that a special purpose module interface processor (MI processor) can be built inexpensively with the following function. It contains a list of address correspondences, triplets (a_1, a_2, l) where:

a_1 is a virtual address used by the sending module

a_2 is an actual address used by the receiving module (note that each FLEX/32 has all its memory in a single address space)

l is the length of the memory block associated with a_1 and a_2.

As a block of data comes into the MI processor its destination is identified, the data is buffered and then sent on to the next module. It is assumed that MI processor transit has an initiation time of perhaps 20 micro seconds. That is, communication time is

$$r + s^*(\text{data block length})$$

where r and s are about 20 and 7 microseconds, respectively.

IV. THE GLOBAL MODULE

Each multi-FLEX has a single *global module* whose principal functions are to: (a) provide some global memory for the multi-FLEX, and (b) provide overall control of the system. The global module is also a FLEX/32 but it is configured with more memory, less computing power and different communications than the modules described in Section II. The typical configuration is shown in Figure 2. The components are:

PE #1: 4 MIPS processor plus 4 Mbytes of local memory This is the processor that has overall control of the system.

PE #2: 4 MIPS processor plus 4 Mbytes of local memory. This serves as a high speed secondary memory and slave for PE #1.

PE #3→6: 4 MIPS processor plus 1 Mbyte of local memory These PEs assist PE #1 in control and in maintaining the global memory. They also serve as external I/O interfaces. Note that most system external I/O is through the other modules.

PE #7,8,...,20: 4 MIPS processor plus 4 Mbytes of local memory These memories, along with this module's locally shared memory, form the global memory of the system. The PE's assist in memory management.

The common boards and the busses are the same as with the basic FLEX/32 module. The memory in this module is 61 Mbytes of global memory (divided into two types), 8 Mbytes for the control processor and 4 Mbytes for global external I/O interfaces.

V. THE FULLY CONNECTED MULTI-FLEX

One of the strengths of the multi-FLEX machines is the high communication bandwidth available. It is well known that full interconnection is a multiprocessor limits the number of processors to a small number. Using the FLEX/32 as modules naturally leads to a machine with a global module and seven or eight other modules. We discuss the seven other module cases as illustrated in Figure 3. All eight modules are connected together directly and each has two busses for local external I/O. The characteristics of this machine are briefly summarized in the following table.

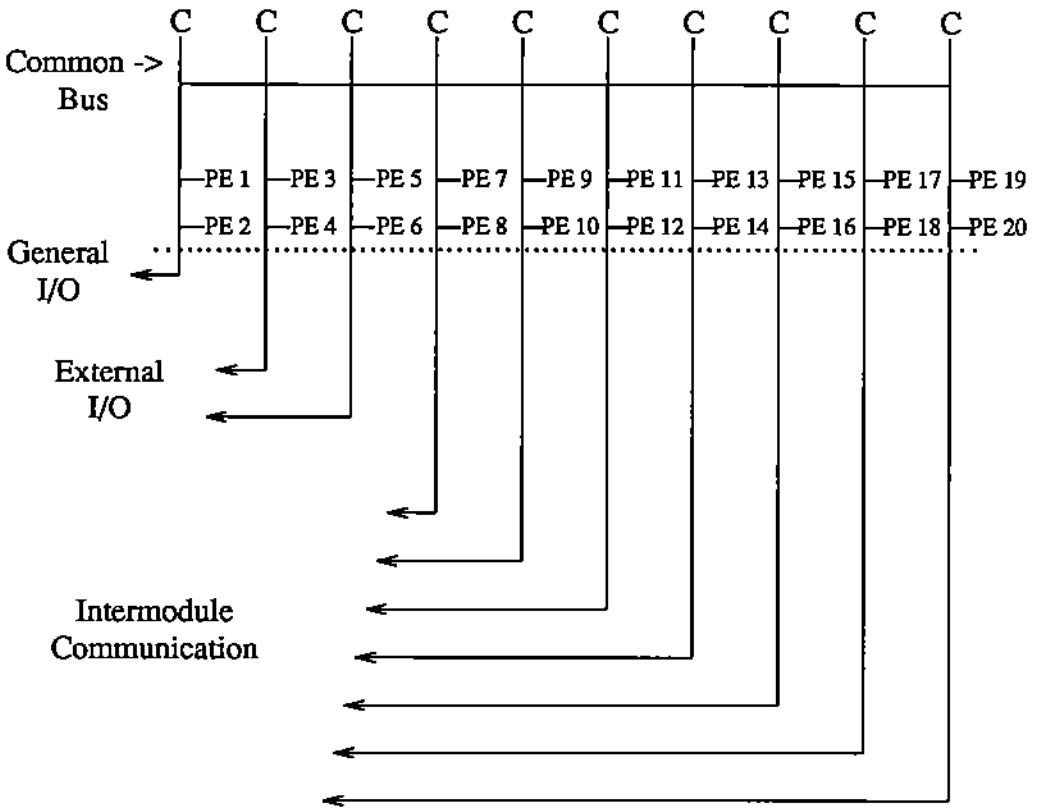

Figure 2. Diagram of the FLEX/32 configured for the Global Module of a multi-FLEX. PE denotes a board with a processor and local memory, and C denotes a board with shared (common) memory.

Computing Power	*Processors*	*MIPS*
Control	2	8
Global I/O	4	16
Internal Communication	14	56
Processing	140	560
Total	160	640

Memory	*Megabytes*
Global	61
Shared local	35
Local	560
Total	596

Communication	*Busses*	*Mbytes/sec.*
External: Global	3	60
Local	21	420
Internal: Global memory	7	140
Inter-module	21	240
Intra-module	8	320
Total	60	1080

The number of busses and inter-module communication is less than one might expect because we count two busses joined by a MI processor as a single bus with 20 Mbytes/sec gross capacity.

The characteristics of this machine may be restated in scientific computing terms as having about 200 Mflops, 75 million words (64 bits) of memory, 87 Mwords/sec of internal communications, and 60 Mwords/sec of external communications. It is very rich in communications capacity, probably unbalanced in that respect just as most existing machines are unbalanced by having too little communication capacity. For comparison, a fully configured Sequent 2200 with 20 processors has about 10 Mflops, 4 million words of memory, 125 Kwords/sec of external communications, and 500 Kwords/sec of internal communications. The ratios for these four characteristics are

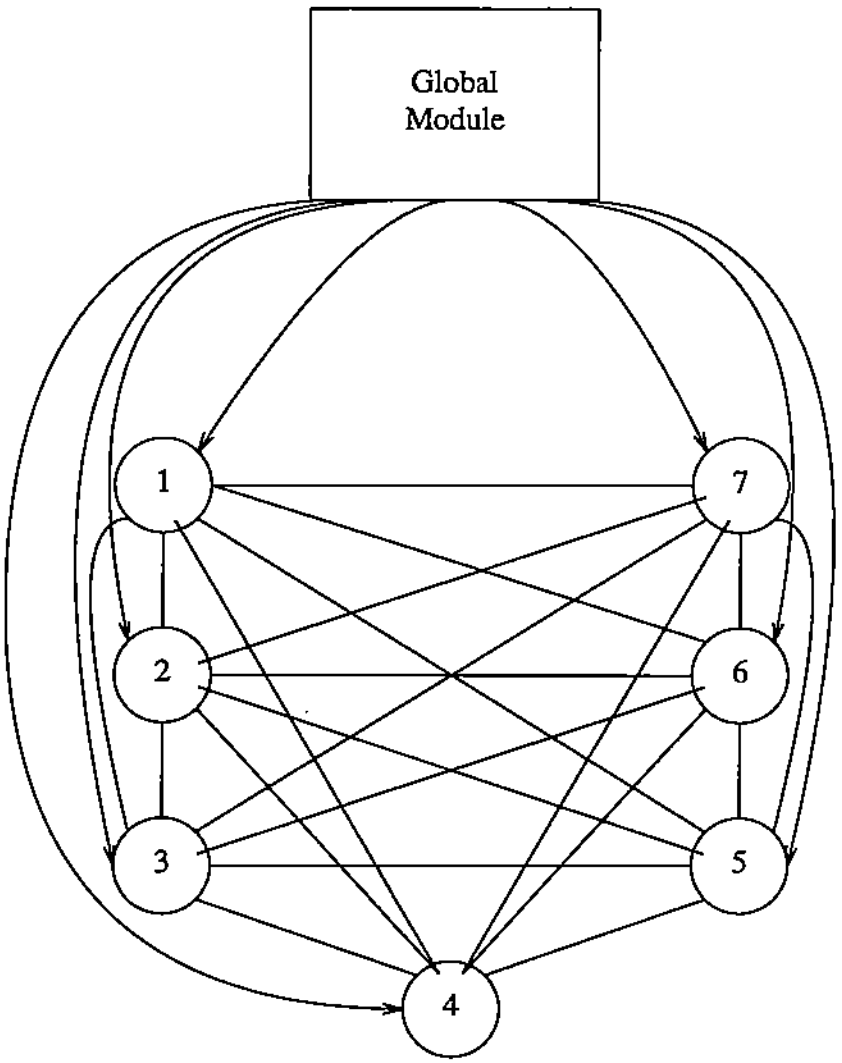

Figure 3. Block diagram of the Fully Connected Multi-FLEX.

	Mflops:	Mwords:	Mwords/sec:	Mwords/sec
Fully Connected Multi-FLEX	1:	0.38:	0.43:	0.3
Sequent 2200	1:	0.4:	0.012:	0.05
Cyber 205 (2 pipes)	1:	0.02:	0.016:	--

It is widely believed that the Cyber 205 does not have enough memory and that the Sequent has both internal and external communications bottlenecks. The internal communication ratio on the Cyber 205 is left off as it only has local memory.

VI. THE FLEX-CUBE

Once one cannot have full interconnection between modules in a multi-FLEX, many possibilities open up. We choose to consider a 64 module hypercube multi-FLEX (a FLEX-cube) with one global module. We must have the 64 modules (hypercube nodes) connect to the 7 internal communication busses of the global module. This will require a special switch to be built, we make the following assumptions about it:

* It will cost a small percentage of the total cost of the FLEX-cube
* It will cause negligible extra delay in the access to the global module beyond that inherent in a fan-in of 64 to 7.

The FLEX-cube is shown schematically in Figure 4.

The characteristics of this machine are briefly summarized in the following table.

Computing Power	*Processors*	*MIPS*
Control	2	8
Global I/O	4	16
Internal Communication	14	56
Processing	1280	5120
TOTAL	1300	5200

Memory	*Megabytes*
Global	61
Shared local	320
Local	5120
TOTAL	5501

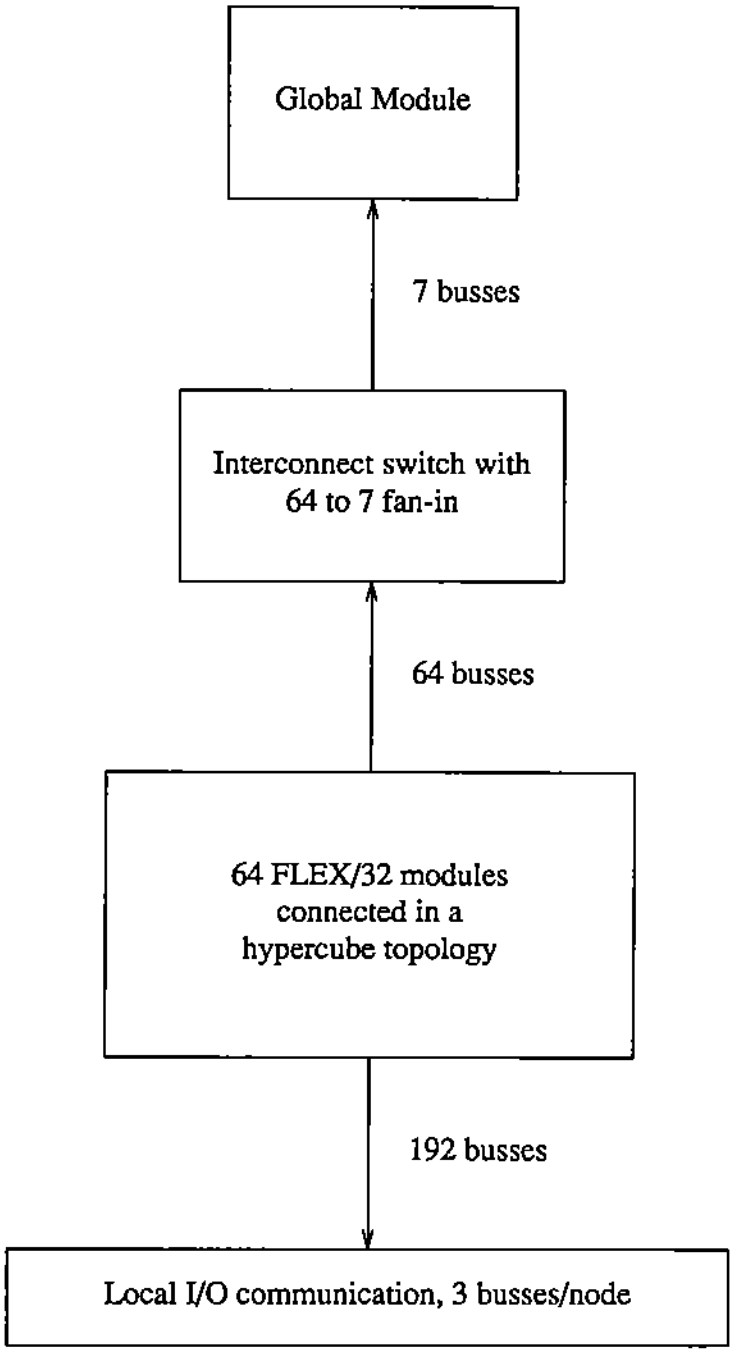

Figure 4. Block diagram of the FLEX-cube with 64 nodes.

Communication		*Busses*	*Mbytes/sec.*
External:	Global	3	60
	Local	192	3840
Internal:	Global memory	7	140
	Inter-module	192	3840
	Intra-module	64	2560
Total		458	10440

This is, of course, a very large and powerful machine. It would, however, fit in existing machine rooms of large scientific installations without major modification in power or cooling and without a major effort at repackaging. In terms of scientific applications, the FLEX-cube is a 2.5 Gflops machine with 650 Megawords of memory, 450 Mwords/sec of external communication, and 800 Mwords/sec of internal communication. Its characteristic ratios are 1:0.25:0.18:0.32.

The breadth of applicability that such a machine would have is illustrated by the following two applications it could handle (we conjecture):

Application 1: Dedicated to one problem: Weather Forecasting for the northern hemisphere with

* effective horizontal grid spacing of 1/4 to 1/2 mile (2 miles of uniform)
* effective vertical spacing of 2000 feet
* current conditions every 2 seconds from 50 million locations.

Weather forecasts could be made at approximately 1000 times faster than real time; new 3 hour forecasts could be made every 5 seconds. The accuracy of short term forecasts would be very accurate except for very local disturbances.

Application 2: Large numbers of low intensity users: Terminal users doing typical UNIX oriented tasks now done on VAX 11/780's such as:

* editing of small/medium files
* compilation/execution of small/medium jobs
* processing of mail, short documents

* manipulation, examination, etc. of small/medium files.

Adequate response times and service could be provided for 30-40,000 terminals.

This machine probably has too little global memory, the balance in this respect could be improved by reducing the size of the cube to 32 or 16. These FLEX-cubes would still be very powerful and flexible. Another approach would be to further exploit the locally shared memory idea and extend it to two levels. The 64-cube would be divided into eight 8-cubes, each with its own "global module" providing shared memory between the modules in the 8-cube. Then, in turn, these eight module would connect to a single global module that provided global control and memory for the whole machine. The number of busses used for global module access would be changed to 8 instead of 7 as described in Section IV. This approach would increase the cost of the machine by perhaps 15%, probably a worthwhile investment. The main deterent is not cost but software. The use of locally shared memories and the resulting higher level of control is discussed in Section VIII and is seen to present significant research issues. It would be risky to go to two such levels of control and memory before a reasonable understanding was reached on how to use one effectively.

VII. EFFECTS OF TECHNOLOGY ADVANCES

The multi-FLEX machines discussed earlier are based on current products. Since such machines will not be built for several years, we should consider the effects of potential technological advances. The FLEX/32 architecture allows fairly easy enhancement of the 30 boards. Shortly, we expect to see:

* faster processors (say 10 MIPS at current prices)
* denser memories (say 4 Mbytes instead of 1)
* specialized processor boards (pipelined vector processors in the 10-20 Mflops range, FFT boards, ...)

Thus we can expect the speed to increase by 2½, the memory by 4 simply because of technology advances. Since the multi-FLEX is rich in communication capacity, this will make the machines better balanced. Before very long memory chips with 16 times the current density will be available and they should be considered for the global modules in order to provide more global memory. For example, we could foresee a FLEX-cube with 6 Gflops, 2.8 Gwords of memory (including 120 Mwords of global and 160 Mwords of locally shared) along with 650 and 450 Mwords of internal and external, respectively, communication. Its characteristic ratios are 1:0.45:0.11:0.07.

A serious effort to produce such machines (note their costs will be in the $4-20 million range) would probably include repackaging to reduce the size somewhat. This would be worthwhile, but size is not a critical problem even now.

VIII. MACHINE PARTITIONING

The multi-FLEX machines are quite powerful, more so than any currently available machines and comparable for machines to be delivered in the 1987-89 time frame. As the two potential applications presented in Section VI suggest, we believe one of the major strengths of such machines is their flexibility in handling jobs of all sizes. We think that the FLEX/32 module is too coarse a level for partitioning the machine so we propose that a module be partitionable into two parts, one that runs a true parallel processing system (like MMOS) and one that runs a multi-processor UNIX system (like the Sequent or Encore). Of course, a job that has 11 processors allocated to it may use them in whatever way it chooses, including running many sub-jobs. The key is to provide the capability to effectively use the computational power of the multi-FLEX on almost the full spectrum of applications that arise. Note also that partitioning provides a good deal

of fault tolerance in the multi-FLEX.

IX. OPERATING SYSTEMS

We have already mentioned that each FLEX module will have three operating systems available:

1) Normal uniprocessor UNIX running on PE #1.

2) Multiprocessor, task allocating UNIX running on PE #1 through PE #k ($2 \leq k \leq 20$). This is a system such as those on the Sequent, Encore and similar multiprocessors.

3) MMOS, a parallel multiprocessor system running on PE #(k+1) through PE #20 ($0 \leq k \leq 19$)

The normal UNIX and MMOS systems already exist though they do not yet permit concurrent operation on a partitioned FLEX/32. The development of a multiprocessor, task allocating UNIX for the FLEX/32 is moderate project (as operating systems development goes). The MMOS system is quite rudimentary in the services it provides. This is the norm for operating systems for parallel machines, their developers all seem to be content to provide a very low level of service, even for the most expensive machines.

The technical challenge here is to provide the operating system for the global module. This system must operate at a level considerably above that of current operating systems. The routine chores of task scheduling, file manipulation, I/O scheduling, etc. are handled by the lower level operating systems running on each module. The primary responsibilities here are load balancing, job priority arbitration, meeting deadlines and general management of the multi-FLEX. This is an "intelligent operating system", one

that includes an expert system for its management responsibilities. It has substantial computing power dedicated to it so it is not restricted to using simple-minded, trivial to compute, rules. The lower level operating systems use conventional scheduling procedures, the machine is controlled by the global system assigning users and jobs to modules or module parts and then by setting (and adjusting) their priorities. These priorities are passed to the lower level operating systems in a tabular form.

The development of such an operating system is a major R&D task. More details of the concepts involved and approaches to be used are given in research proposals unrelated to the multi-FLEX, see [CoRi86].

X. SUMMARY: ARCHITECTURE CHARACTERISTICS AND COSTS

In this section we gather together the basic facts about the various multi-FLEX machines discussed in this report. Cost estimates are not based on current prices of Flexible Computers but rather on the following:

* A PE with local memory costs $15,000
* A board with only memory costs $10,000. These include the boards with locally shared memory (common cards in FLEX/32 terminology)
* A FLEX/32 cabinet (with no boards) costs $150,000
* Software and peripherals adds 30% to the total cost.
* Putting the modules together (including MI processors) adds 10% to the total cost.

These estimates assume a substantial volume of production of identical units.

Table 1 gives a summary of the architecture characteristics of various multi-FLEX machines and their components. The suffix AT stands for Advanced Technology and indicates the use of the technology described in Section VII. The 64-cube-2level uses two levels of locally shared memory as described in Section VI. All these modules have 240 Mbytes/sec of internal communication capacity. Note that the same busses are used

for intermodule, external and for internal communication, so that maximum communication capacity of all kinds for a module is 240 Mbytes/sec.

Table 1. Architecture characteristics of FLEX and multi-FLEX machines. AT means Advanced Technology (see Section VII) and 2level means use of two levels of locally shared memory (see Section VI). Memories are in Megabytes and communication is in Megabytes/second.

		Memory			Communication	
Machine	Processors	Local	Locally shared	Global	Intermodule	External
FLEX/32	20	80	5	--	--	200
FLEX module	20	80	5	--	140	60
Global module	20	12	--	61	140	60
MULTI-FLEX						
Fully connected	160	168	35	61	600	480
32-cube	660	2560	160	61	3340	1980
64-cube	1300	5120	320	61	6540	3900
64-Cube-2level	1460	5248	320/552	61	7180	2140
Fully connected AT	160	672	140	244	600	480
32-Cube AT	660	10240	640	976	3340	1980
64-Cube AT	1300	20480	1280	976	6540	3900

Table 2 summarizes these machines in terms of performance, ratios of performance and estimated cost.

Table 2. Performance and estimated cost characteristics of FLEX and multi-FLEX machines. The notations AT and 2level of Table 1 are used, memory is in 64 bit megawords and operations are 64 bit, cost is dollars (K=1000, M=1,000,000). The ratio entry is as described in Section V.

			Communication			
Machine	Mflops	Memory	Intermodule	External	Ratios	Cost
FLEX/32	40	11	--	25	1:0.27:--:0.63	680K
FLEX module	40	11	18	8	1:0.27:0.44:0.19	680K
Global module	40	9	18	8	1:0.24:0.44:0.19	680K
MULTI-FLEX						
Fully connected	200	75	87	60	1:0.38:0.43:0.30	6M
32-cube	1250	325	225	400	1:0.38:0.43:0.30	25M
64-cube	2500	650	450	800	1:0.25:0.18:0.30	48M
64-cube-2level	2800	730	500	860	1:0.26:0:18:0.30	54M
Fully connected	500	300	87	60	1:0.60:0.17:0.12	6M
32-cube AT	3200	1400	225	400	1:0.44:0.07:0.12	25M
64-cube AT	6300	2700	500	860	1:0.43:0.08:0.14	48M

REFERENCES

[FLEX86] FLEX/32 Reference Manual, Flexible Computer Corp., Dallas, Texas.

[CoRi86] D. Comer, J. Rice, Intelligent Operating Systems (working papers), Computer Science Department, March, 1986.