COMPUTER SYSTEMS LABORATORY

STANFORD UNIVERSITY · STANFORD, CA 943054055

A Survey of Concurrent Architectures

Victor W. K. Mak

Technical Report: CSL-TR-86-307

September 1986

The work described herein was supported by' NASA Ames Research Center under contracts NAG 2-248 and NAGW 419.

A Survey of Concurrent Architectures

by

Victor W. K. Mak

Technical Report: CSL-TR-86-307

September 1986

Computer Systems Laboratory

Department of Electrical Engineering

Stanford University

Stanford, California 94305

Abstract

A survey of 18 different concurrent architectures is presented in this report. Although this is by no means complete, it does cover a wide spectrum of both commercial and research architectures. A scheme is proposed to describe concurrent architectures using different dimensions: models of computation, interconnection network, processing element, memory system, and application areas.

Key Words and Phrases: Concurrent Architecture, Interconnection Network, Model of Computation, Parallel Processing, Survey, Taxonomy.

Contents

List of Figures

List of Tables

1 Introduction

In the search for faster computers, two directions have been followed: technology and architecture. Advances in semiconductor technology have been very successful in reducing the size of computers, and at the same time, increasing their speed. However, as the physical limits of semiconducting devices are approaching, people starts to turn to other ways. Current supercomputers make extensive use of pipeline and vector architectures. They are effective ways to improve the speed in problems that can be easily vectorized, but not for problems that involve a lot of data dependent computations like particle tracking. For general class of problems, concurrent architecture seems to be the most promising candidate.

Concurrent architecture is still in its infancy. A lot of problems related to its use are not fully understood yet. Design of such architecture is still in an ad hoc basis. One of the projects in the Center for Concurrency Studies at Stanford University is trying to design a testbed to project the performance of new concurrent architectures. The use of this testbed will certainly enhance our understanding of new architectures. Before the actual design of the testbed, we would like to know the range of architectures that the testbed will be designed for.

In this report, a survey of different concurrent architectures will be presented. Although not complete, care has been taken to include architectures of different flavors so as to cover the widest spectrum possible. A scheme is also used to describe architectures using different dimensions: models of computation, interconnection network, processing element, memory system, and application areas. Using this scheme, we can easily specify a new architecture and check to see if it is in the range of architectures designed for our testbed.

2 Taxonomy of Concurrent Architectures

There are literally hundreds of concurrent architectures proposed and designed in the last decade. These architectures are so different that the usual classifications as SIMD and MIMD [Fly 72] are not sufficient. New taxonomy schemes are developed by many researchers in order to categorize these different architectures into different classes. One particular taxonomy as reported by Haynes and others in [HLS 82] divided the wide spectrum of concurrent architectures into six classes:

1. Multiple special-purpose functional units.

2. Associative processors.

3. Array processors.

4. Data flow processors.

5. Functional programming language processors.

6. Multiple processors.

Architectures with multiple special-purpose functional units are usually designed to perform some specific tasks efficiently. One example is the systolic arrays which will be described in more detail in the next section. Computation intensive problems, in which the kernels are based on a number of basic mathematical operations, have found great success in these structures. Matrix multiplication, solution of linear systems, and FFT are some examples.

Associative processors are those architectures which utilize an associative memory. In associative memory, one bit of any memory word is available on one access, thus it is possible to search the whole memory simultaneously for specified contents by iteration on bit slices. This organization also allows memory words to be addressed by their contents instead of their addresses. One example of associative processors, STARAN, designed and built by Goodyear Aerospace Corporation, will be described in the next section.

Array processors are architectures with multiple arithmetic units operating in lockstep and performing the same operation on different data. This is the most common and popular type of concurrent machines available on the market. They are particularly suitable to problems that involve a large proportion of array data types. Five examples will be studied in the next section: Illiac IV, BSP, MPP, CHiP, and NON-VON.

Data flow computers are very different from the conventional von Neumann architectures in which a program counter is used to schedule the next instruction to be executed. An instruction in a data flow computer is ready for execution when its operands arrive. As a consequence of this data-activated property, a very high level of concurrency can be exploited. The Data Driven Signal Processor (DDSP) will be given as an example of this type of concurrent architecture.

Functional programming (FP) language processor or reduction machine has gained considerable interest recently. The main advantage of FP is that when algorithms are described in such applicative languages, much parallelism can occur automatically – with no analysis of program structure and without explicit programmer involvement with parallelism. SERFRE will be studied as an example in the next section.

Multiple processors belong to the class normally called MIMD. They are more flexible than the classes described above; however, their control is much more complex. The interconnection network, which connects the processors, usually forms a crucial part both in the design and operation of each architecture. Since the architectures in this class are so diverse that nine different machines in the next section will be studied: Cedar, FMP, S-l, Cm*, HEP, Empress, MP/C, Ultra, and TRAC.

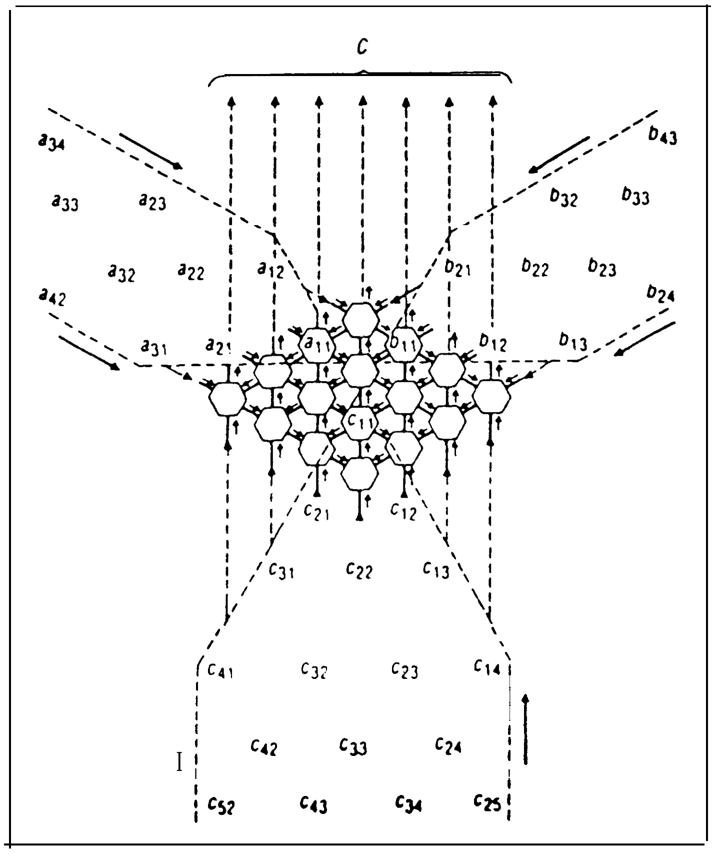


Figure 1: Hex-connected systolic array

3 Architectures Studied

In this section, 18 different architectures will be described. This is by no means a complete survey of all concurrent architectures, but an attempt to try to cover as wide a spectrum as possible. Using the taxonomy described in the previous section as a guideline, architectures that belong to these six different classes were chosen. Since array processors (SIMD) and multiple processors (MIMD) have received most attention, most architectures that were chosen belong to these two classes. In each subsection that follows, the most significant parts of each architecture will be described.

3.1 Systolic Array

The systolic architectural concept [Kun 82] was developed by Kung and associates at Carnegie-Melon University, and is a general methodology for mapping high-level computations into hardware structure. A systolic system consists of a set of interconnected cells, each capable of performing some simple operation. Information in a systolic system flows between cells in a pipelined fashion, and communication with the outside world occurs only at the boundary cells.

In Fig. 1, the hex-connected systolic array can be used to multiply two $N \times N$ band matrices of bandwidths W_1 and W_2, each of which performs the inner product operation $C \leftarrow C + A \times B$. The entire multiplication requires only $3N + \min(W_1, W_2)$ time units. As the matrices shift into the array, they always move in exactly the same direction and requires no control. Each cell performs one computation at each step, and input and output are overlapped with computation. For each I/O access, there are multiple computations performed on the data item, thus execution of compute-bound problems can be speeded up, without increasing the I/O requirements. This is a very significant improvement over the classical von Neumann architecture in which the memory access time is associated with each operation of the data item.

For specialized algorithms that can be implemented by the systolic array, they are fast, hardware efficient, and require no software control in communication and synchronization. The major problem with a systolic array is still in its I/O barrier. Implementation of the systolic array on a VLSI chip is limited by the number of pins, or I/O terminal, available on a single chip.

3.2 STARAN

STARAN [Bat 74] is the first bit-serial parallel processing system developed by Goodyear Aerospace Corporation in 1972. It consists of up to 32 associative array modules, each contains 256 processing elements, a 256-word 256-bit multidimensional access (MDA) memory, a flip network, and a selector. Each processing element operates serially bit by bit on the data in all MDA memory words. The MDA memory can be addressed in either bit-slice (one bit of all 256 words) or word-slice (all bits of one word). Thus, data can be input and output in the usual word by word fashion while processing can be done in bit-serial fashion. The flip network is used for data shifting or manipulation to enable parallel search, arithmetic or logical operations among words of the MDA memory.

STARAN has high-speed input-output capabilities and the ability to interface easily with conventional computers which handle the tasks that must be processed in a single sequential data stream. The main application areas of STARAN are in signal processing and database.

3.3 Illiac IV

Illiac IV [BDM 72] was developed at the University of Illinois in the 1960s and fabricated by the Burroughs Corporation in 1972. The original design had 4 quadrants of 64 mesh-connected processing elements under the supervision of 4 control units. Due to cost escalation and schedule delays, only 1 quadrant was ever built. The speed of the 64-PE

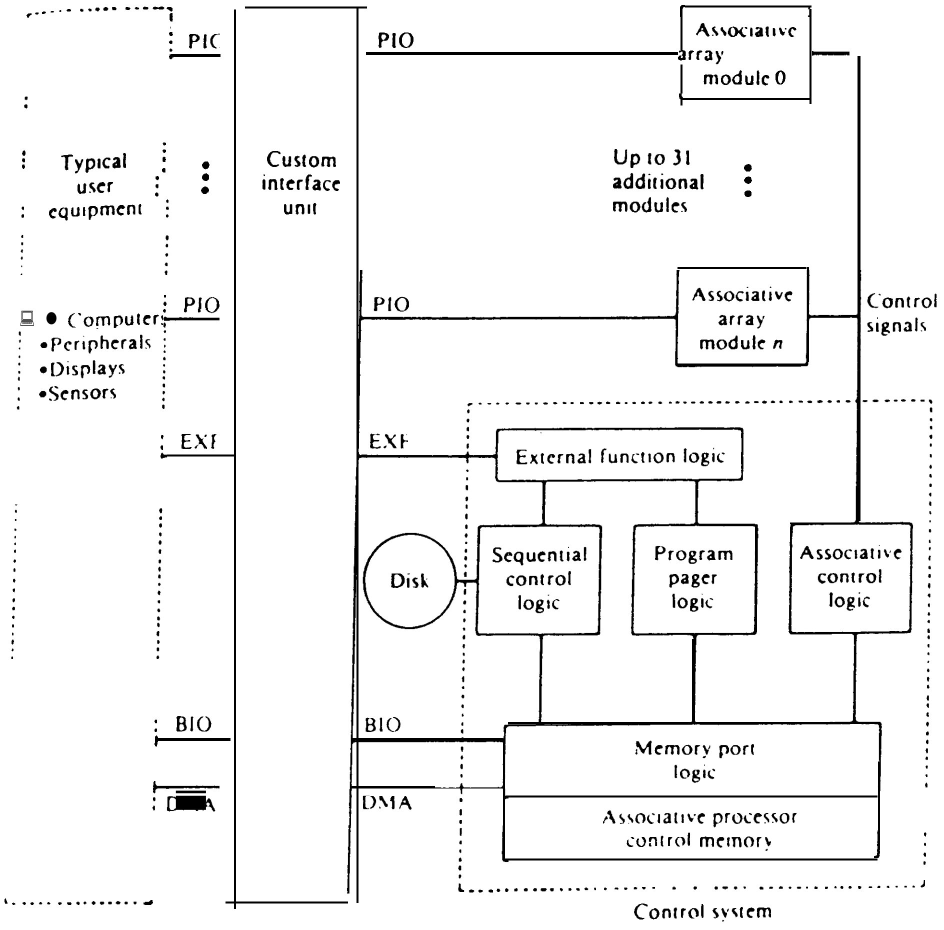


Figure 2: The STARAN System Architecture

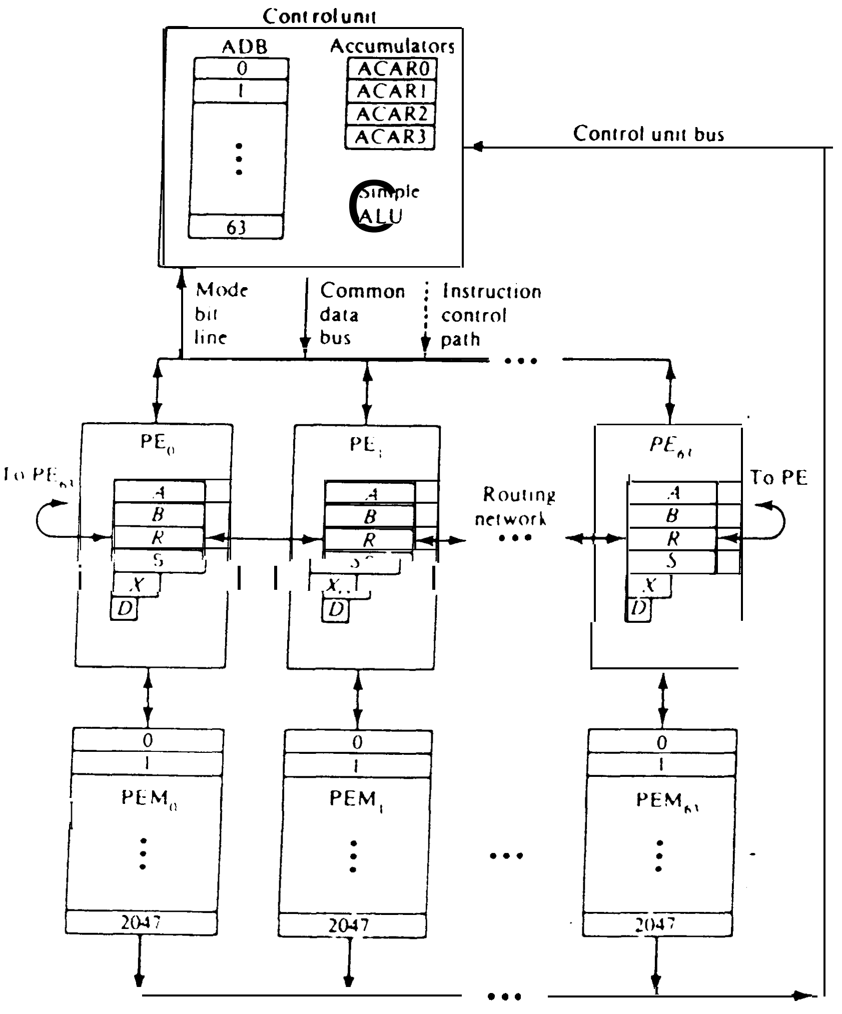
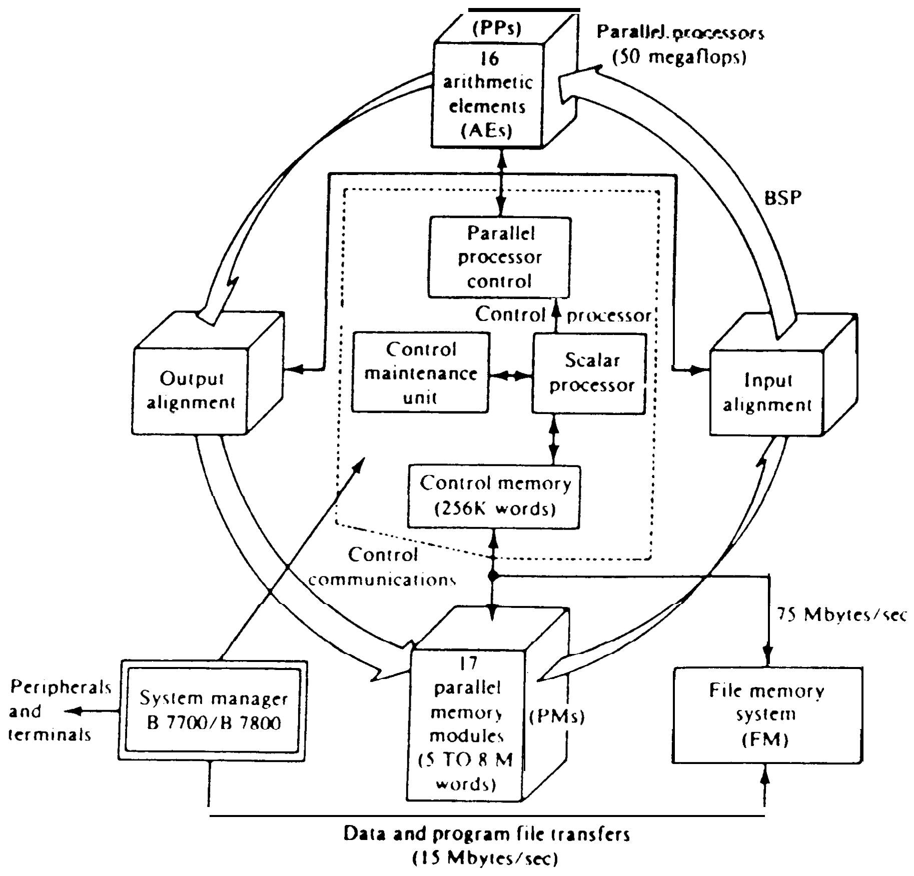


Figure 3: A 64-PE Illiac IV Array



(PPs)
16 arithmetic elements (AEs)
Parallel processors (50 megaflops)
BSP
Parallel processor control
Control processor
Control maintenance unit
Scalar processor
Output alignment
Input alignment
Control memory (256K words)
Control communications
75 Mbytes/sec
17 parallel memory modules (5 TO 8 M words)
(PMs)
File memory system (FM)
Peripherals and terminals
System manager B 7700/B 7800
Data and program file transfers (15 Mbytes/sec)

Figure 4: Functional Structure of BSP

quadrant is approximately 200 million operations per second. The control unit controls and decodes the instruction stream and broadcasts instructions and common data to all PEs. It is also a scalar processor by itself besides having the ability to control the PE-array. Each PE is a powerful computing unit, and has a 64-bit wide routing path to its four neighbors. The main application area is in scientific applications like numerical weather forecasting and nuclear engineering research.

3.4 BSP

The Burroughs Scientific Processor (BSP) [KS 82] was an attempt by Burroughs Corporation to improve on the Illiac IV design. It has 16 arithmetic elements and 17 (prime number) memory modules interconnected by two alignment networks: full crossbar switch with broadcasting and conflict resolving ability. This permits general-purpose interconnectivity between the arithmetic array and the memory-storage modules. It is the combined

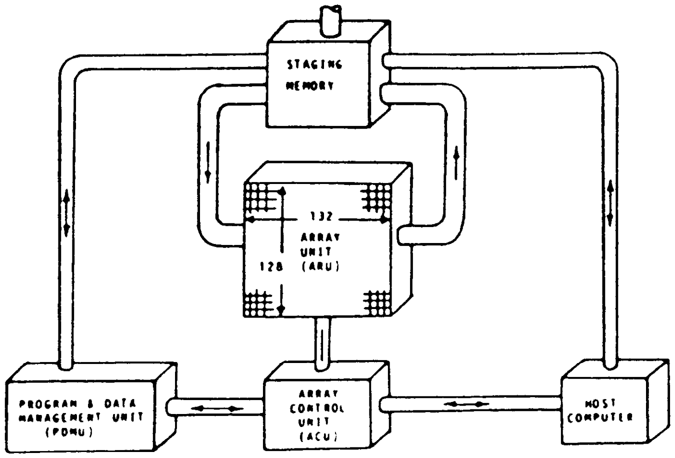


Figure 5: Block Diagram of the MPP

function of the memory-storage scheme and the alignment networks that supports the conflict-free capabilities of the parallel memory. The parallel processors perform vector computation with a clock period of 160 ns. The control processor provides the supervisory interface to the system manager in addition to controlling the parallel processor. The scalar processor processes all operating system and user-program instructions which are stored in the control memory. It executes some serial or scalar portions of user programs with a clock rate of 12 MHz and is able to perform up to 1.5 megaflops. The BSP is capable of executing up to 50 megaflops and is used mainly for scientific applications.

3.5 MPP

Like STARAN, Massively Parallel Processor (MPP) [Bat 82] was also designed and built by Goodyear Aerospace Corporation starting from 1979 to be a high speed satellite image processing system. The processor has 16,896 bit-serial processing elements (PE's) arranged in a 128-row by 132-row (4 redundant rows for fault tolerance) rectangular array with strictly nearest-neighbor connections. The edge connection is programmable so that the array may look like a plane, a cylinder, a torus, a spiral, or a linear string. On 32-bit floating-point data, addition occurs at 430 MOPS and multiplication at 216 MOPS. The staging memory in the input-output path of the array unit acts both as a buffer between the array unit and the outside world, and also to reformat data so both the array unit (bit-serial) and the outside world (word-serial) can transfer data in the optimum format. MPP is a SIMD machine and all PE's perform the same instruction on every machine clock cycle. Although built for satellite imagery processing, preliminary application studies indicate that MPP can also support general image processing, weather simulation, aerodynamic studies, radar processing, reactor diffusion analysis, and computer image generation.

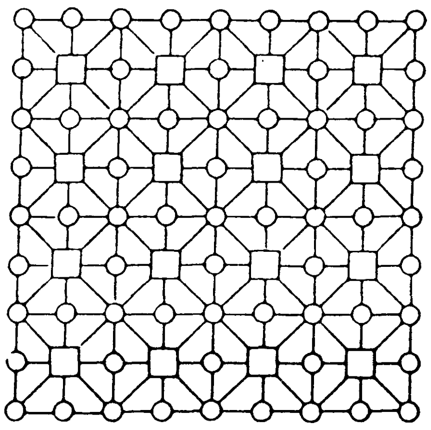


Figure 6: CHiP lattice. PEs shown as squares, switches as circles.

3.6 CHiP

The CHiP computer [Sny 82] is a family of architectures each constructed from three components: a collection of homogeneous microprocessors (28 to 216), a switch lattice, and a controller. The microprocessors are not directly connected to each other, but rather are connected at regular intervals to the switch lattice. Each switch in the lattice contains local memory capable of storing several configuration settings and thus be changed dynamically during program execution: mesh for dynamic programming; hexagonally connected mesh for LU decomposition; torus for transitive closure; tree for sorting; double tree for searching; etc.. The perimeter switches are connected to external storage devices. The controller is responsible for loading the switch memory. CHiP processing begins with the controller broadcasting a command to all switches to invoke a particular configuration setting. Individual microprocessors then synchronously execute the instructions stored in their local memory.

By integrating programmable switches with the processing elements, the CHiP computer achieves a polymorphism of interconnection structure that also preserves locality, thus allowing algorithms that exploit different interconnection patterns to be used in the same program. CHiP can be viewed as a configurable systolic array: it has all the advantages of the systolic array while it is still general enough to embed different interconnection patterns in its lattice.

3.7 NON-VON

The NON-VON architecture consists of two parts: primary processing subsystem and secondary processing subsystem. The primary processing subsystem is organized as a binary tree of small processing elements (SPE's) which have no stored program and can only execute instructions sent by its ancestor nodes. The SPE's in the first few levels of

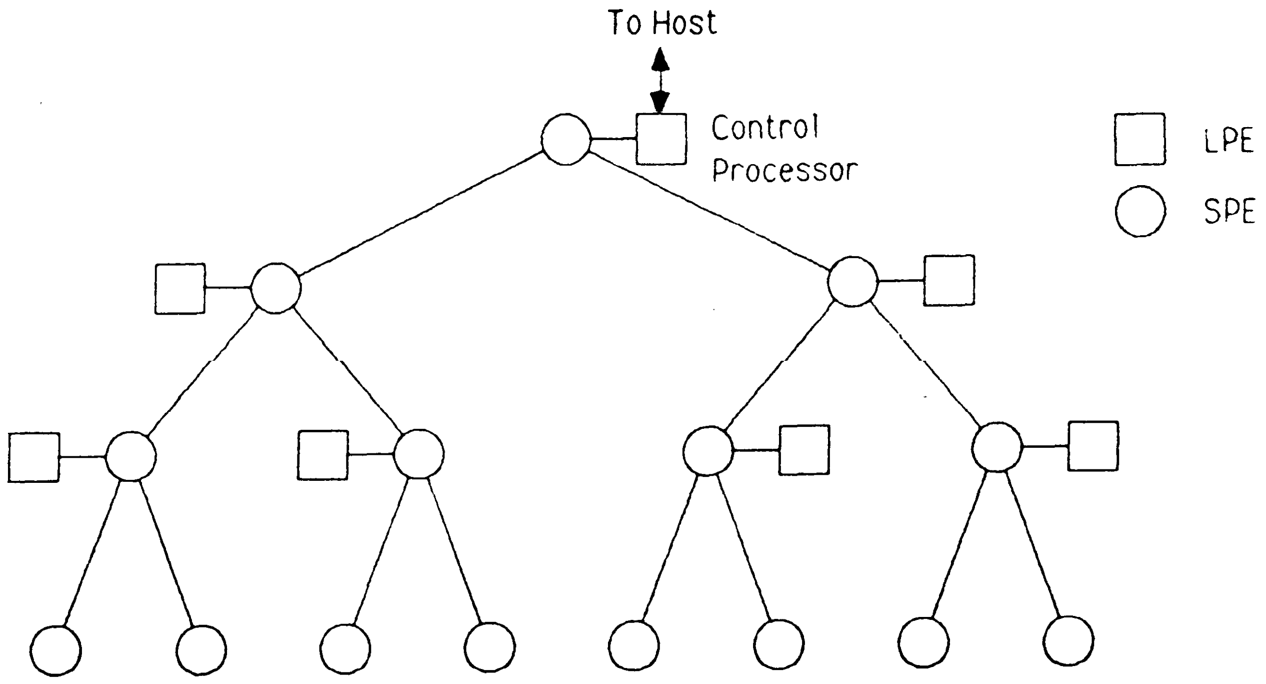


Figure 7: NON-VON Primary Processing Subsystem

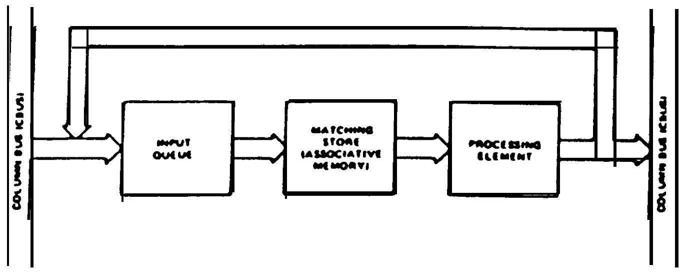


Figure 8: DDSP Processor Block Diagram

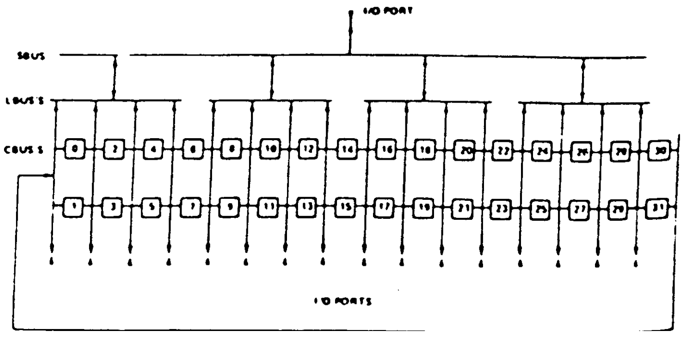


Figure 9: DDSP Interconnection Network

the tree are each connected to a large processing element (LPE) which has locally stored program and may operate independently. Thus, NON-VON can act as a single SIMD machine with the node at the root being the ancestor of all the nodes below it, or as a multiple SIMD machine with each subtree controlled by a node connected to an LPE. The LPE connected to the root is called the control processor and is also connected to the host processor.

The secondary processing subsystem consists of 64 to 256 disk-drives each connected via an intelligent head unit to an LPE. These intelligent head units perform certain computationally simple operations (e.g. selection) on the fly, thus added to the processing power of the whole system.

NON-VON is designed to be used mainly in the areas of relational database, sorting and vision.

3.8 DDSP

The Data Driven Signal Processor (DDSP) [HNI 82] is being developed by ESL Incorporated to be a programmable, modular, high-speed data flow computer primarily for signal processing applications. Its configuration ranges from one to 32 processors with a maximum performance of 71 MFLOPS. DDSP is designed with a high order language (Data

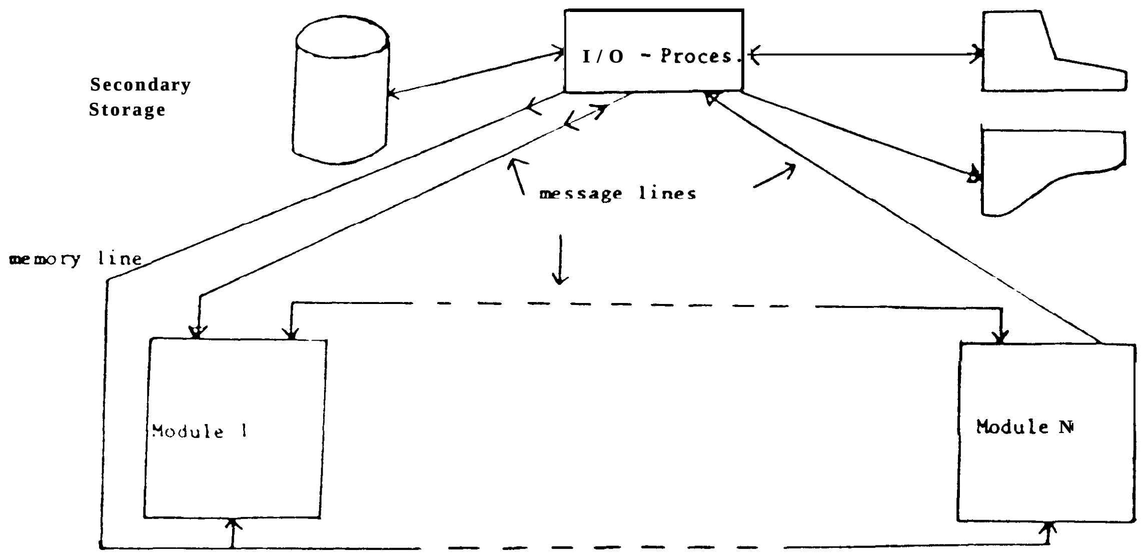


Figure 10: Architecture of the SERFRE

Driven Programming Language, or DDPL) capable of generating efficient machine code, and follows the single assignment rule. It implements a dynamic tagged data flow model where tokens are tagged with a label field determined at run-time. The processors in a DDSP system are closely coupled through an interconnection network. A processor consists of an input queue for temporarily saving tokens, a matching store (associative memory) for associating pairs of tokens, and a processing element for performing high speed integer and floating point computations (2.2 MFLOPS). Because of the nature of signal processing computations, the interconnection network is essentially a linear arrangement of processors with wrap-around from the last pair of processors to the first pair, and augmented by a three level tree used for long distance communication. Besides signal processing, DDSP can also be used in fields of sonar and image processing.

3.9 SERFRE

SERFRE [Vil 82] is a multi-processor command-driven (string reduction) machine and can directly execute a FP language, trying to have subprograms executed on different processors. It is a dynamic loosely-coupled system using direct communication with storage of messages. Fig. 10 describes the architecture of a possible, single-user implementation of SERFRE, and Fig. 11 the structure of a module. The I/O processor controls the memory system as well as the initiation of a program evaluation and returning of the result to the user. The C-processors have their own local memory to store data and function definitions. A C-processor consists of a register for the return address, a stack for the program, registers for the data, and a reduction engine. When asked to evaluate

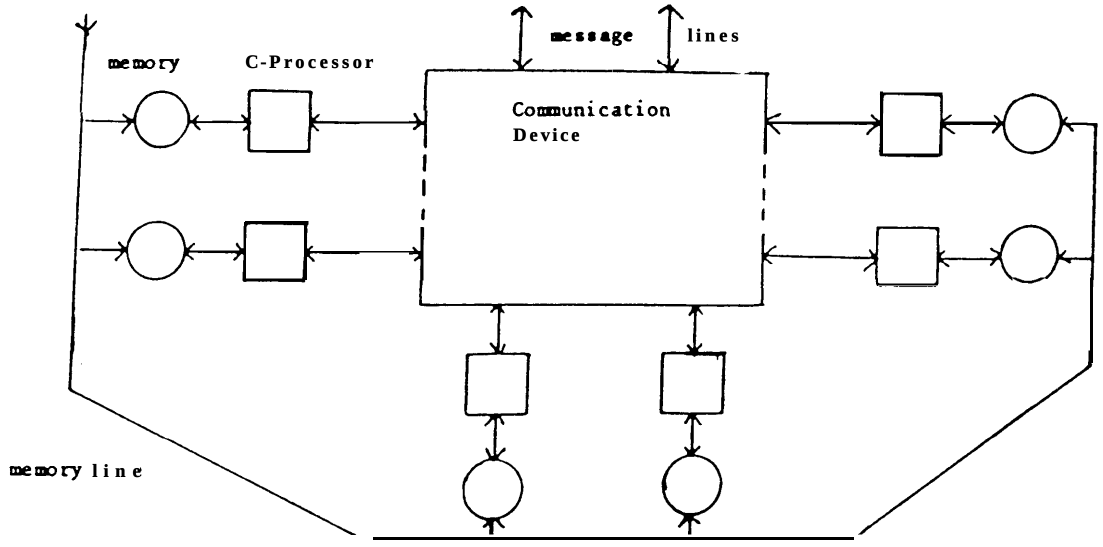


Figure 11: Structure of a Module in SERFRE

a function involving concurrency, it will try to call for other non-busy C-processors to execute the subprograms, if none is available, it will evaluate them sequentially.

3.10 Cedar

The objective of the Cedar project [GKL 83] at University of Illinois is to investigate ways to accommodate several thousands of high performance processors to deliver several gigaflops. It will make use of the VLSI technology to build powerful VLSI processors, for instance, 32-bit, 2.5 MFLOPS. The uniqueness of this architecture is the concept of Macro Dataflow which combines the control mechanisms of data flow architecture and storage management of the von Neumenn architecture. A program is viewed as a flow graph of nodes. Each node is either computational (CPF) or control (CTF). The Global Control Unit executes CTF while the processor clusters execute CPF. A processor cluster consists of a number of processors and local memory modules working cooperatively to execute a CPF. When a CPF is finished, the cluster control unit will signal the Global Control Unit so that other nodes depending on this CPF can be scheduled to be executed. Besides local memory, processor clusters can also access the global memory through the global network, an Omega network.

3.11 FMP

The Flow Model Processor (FMP) [Lun 85] was the result of a series of design studies conducted from 1975 through 1982, sponsored both by Burroughs Corporation and by

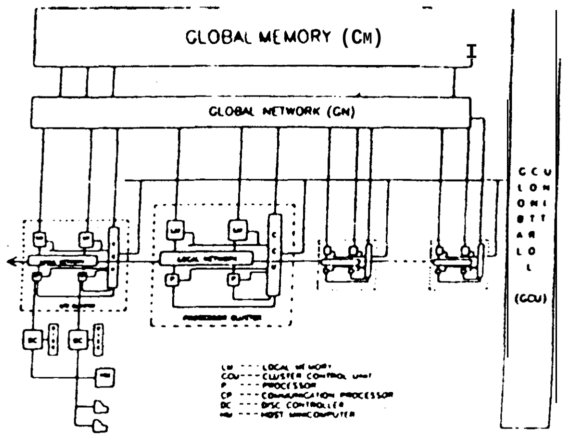


Figure 12: Structure of Cedar

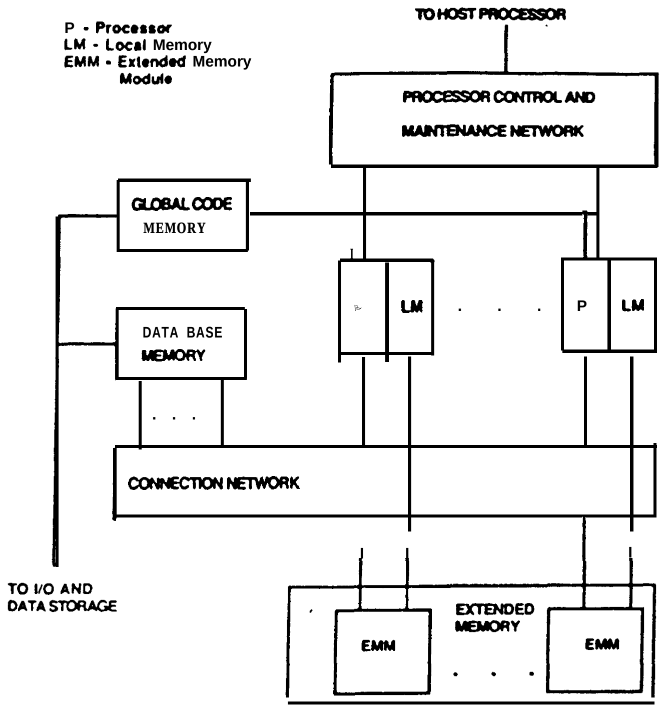


Figure 13: Flow Model Processor Conceptual Design

the NASA Ames Research Center. Its objective was to sustain throughput in excess of 1 GFLOP, and was intended to support large scientific problems especially modeling problems in computational aerodynamics. It was designed to support standard FORTRAN, with extended feature like DOALL, in which codes within the body of this construct is executed once for each value specified in the definition of the DOALL domain.

The conceptual design consists of 128 processor connected through a Connection Network (CN) to the Extended Memory. The Global Code Memory and the Data Base Memory can also be accessed through the CN. The CN, a form of Omega Network, is a circuit-switching network with decentralized control. The Processor Control and Maintainance Network acts as a tree of AND gates to be used to assist in the high speed synchronization at the end of the DOALL. Each processor in the FMP is a powerful computing unit. A scheme similar to that of the IBM 360/91 [Tom 67] was used to allow multiple functional units to be used efficiently.

3.12 S-l

The S-l project [WC 79] has as its general goal the development of advanced digital processing technology for potential application throughout the U.S. Navy. The S-l multiprocessor is designed to be at least 10 times the computing power of the Cray-1. Its architecture consists of 16 independent, identical uniprocessors sharing a main memory of 16 modules, each of 1 billion bytes of semiconductor memory. Each uniprocessor is a powerful computing unit with performance comparable to the Cray-1, and can execute instructions independent of others. A full Crossbar Switch is used as the interconnection network between the processors and the main memory. A maximum peak bandwidth of more than 10 billion bits per second can be achieved when all 16 channels of the Crossbar Switch are transferring data simultaneously. To further reduce the main memory access time, each member uniprocessor contains private cache memories (data and instructions). As many as eight peripheral processors can be attached to each uniprocessor to handle I/O. The synchronization box is a shared bus connected to each member uniprocessor; one of its major functions is to transmit interrupts and small data packets from one uniprocessor to any subset of other uniprocessors in order to coordinate processing streams.

3.13 Cm*

Cm* [SFS 77] is an experimental computer system designed and built at Carnegie-Mellon University. It is intended to be a testbed for exploring a number of research questions concerning multiprocessor systems. Cm* is a hierarchical and modular system, the basic building block is a processor memory pair called a computer module or Cm. Up to 14 Cm's are connected into a cluster. Each cluster has a shared address mapping and routing processor, Kmap, which allows communication with other clusters through the intercluster buses. Communication along the intercluster buses is done in packet switching mode to

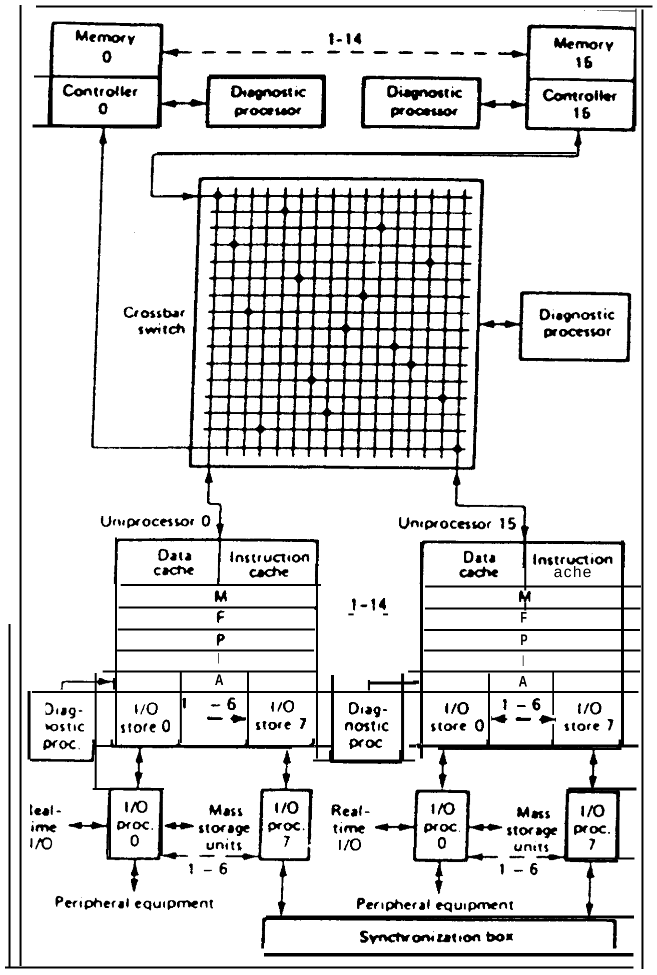


Figure 14: The S-l Multiprocessor

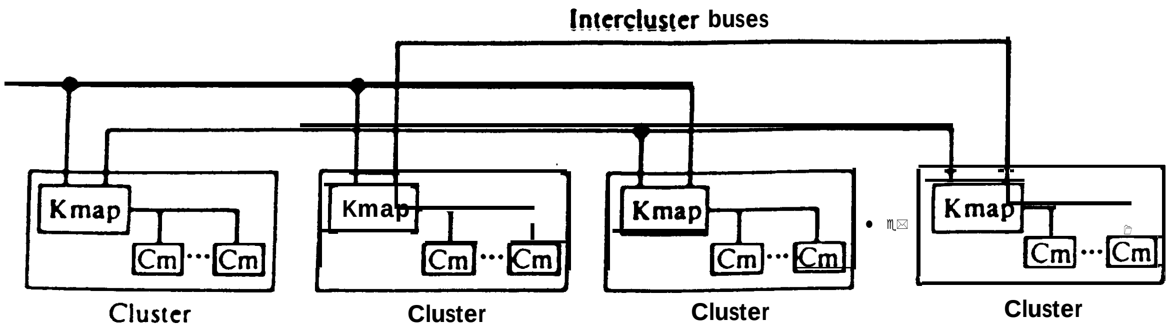


Figure 15: An Example Cm* System

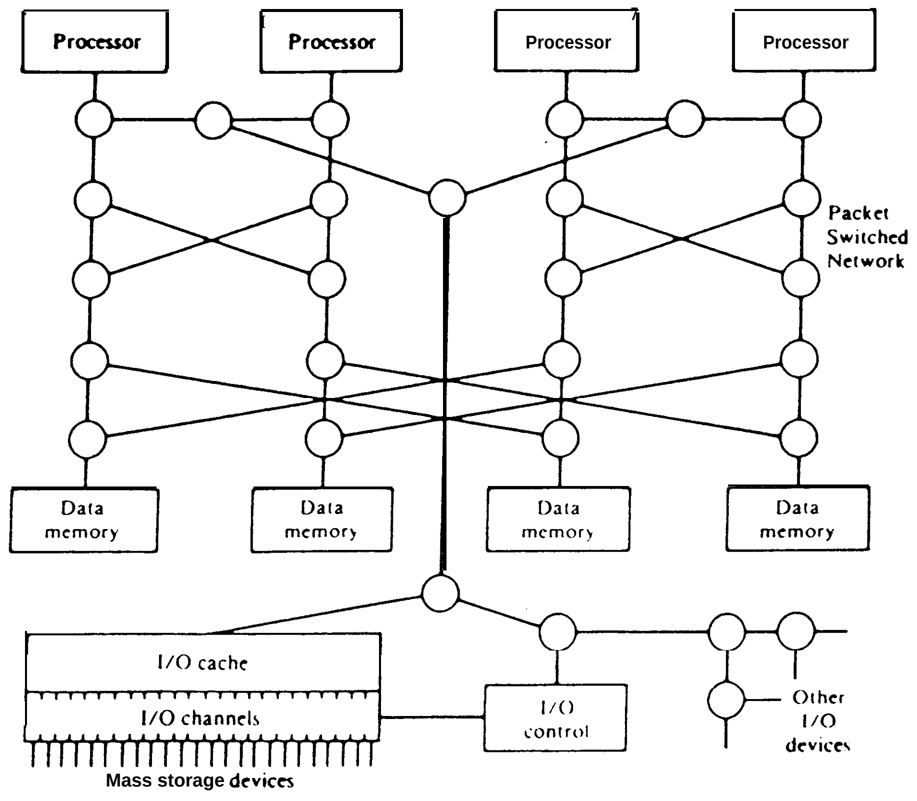


Figure 16: A Typical HEP System

avoid deadlock over bus allocation. The processor is a DEC LSI-11. All processors share a single segmented virtual memory address space of 228 byte. Each processor has a local memory of 64 Kbyte and is also part of the shared memory in the system. Efficient use of the system depends on ensuring that most of the code and the data references made by a processor are held locally to that processor. Inter-process communication is by message-passing and can be easily built on top of the Cm* architecture.

3.14 HEP

The HEP computer system [Smi 78] is an MIMD machine of the shared resource type. In this type of organization, skeleton processors compete for execution resources in either space or time. Two queues are used to time-multiplex the process states. One of these provides input to a pipelined instruction execution unit, which will decode and execute the instruction. For data memory access, the process state enters a second queue. This queue provides input to a pipelined switch which interconnects several data memory modules with several processors. Each processor of HEP can support up to 128 processes. Maximum throughput of 107 instructions per second per processor occurs when there are at least eight totally independent processes in each processor.

HEP instructions and data words are 64 bits wide. A domain of protection in HEP is called a task, and consists of a set of processes which are allowed to communicate with each other. Processes in different tasks or processors may only communicate via data memory if they have an overlapping allocation there. Any register or data memory location can be used to synchronize two processes in a producer-consumer fashion. Three states are provided: reserved, full, and empty. The execution of an instruction tests the states of locations and modifies them in an indivisible manner.

The interconnection switch consists of a number of nodes connected via ports. Messages are sent in packets and routed by the nodes according to their priorities.

3.15 Empress

The ETH-Multiprocessor Empress [BBB 82] was built in order to study the performance of MIMD architectures in general, and particularly in the field of simulation problems. Its architecture consists of a supervisor processor and a number of execute processors, all communicating through an Intercommunication System, Intercom. The supervisor computer is used to partition a problem into executable jobs which will be dispatched by the job control unit to the execute processors. If an execute processor (master) finds its job exhibit inherent parallelism, it can dynamically request more (slave) processors from the job control unit to form a cooperative group. All I/O, precompilation, optimization and code generation as well as the integration step control are done in the supervisor processor.

The Intercom consists of a quadratic organized memory-matrix in which each processor writes to all blocks in its row and can read blocks from its column. Data duplication within the intercom is only executed into the matrix elements of processors working on the same job. In this way, a result provided by any of the processor is made immediately available to all other cooperating processors. Different logical addressing methods are allowed in the Intercom so that cooperating processors may appear to be neighbors although they may be physically apart.

3.16 MP/C

The Multiprocessor/computer (MP/C) [AG 82], a dynamically partitioned system, has the shared memory aspect of tightly coupled multiprocessor systems and also the connection simplicity associated with message-connected, loosely-coupled multicomputer systems. It is proposed as a candidate for the effective execution of process-structured algorithms. Its architecture consists of a number of processor and memory modules, all connected to

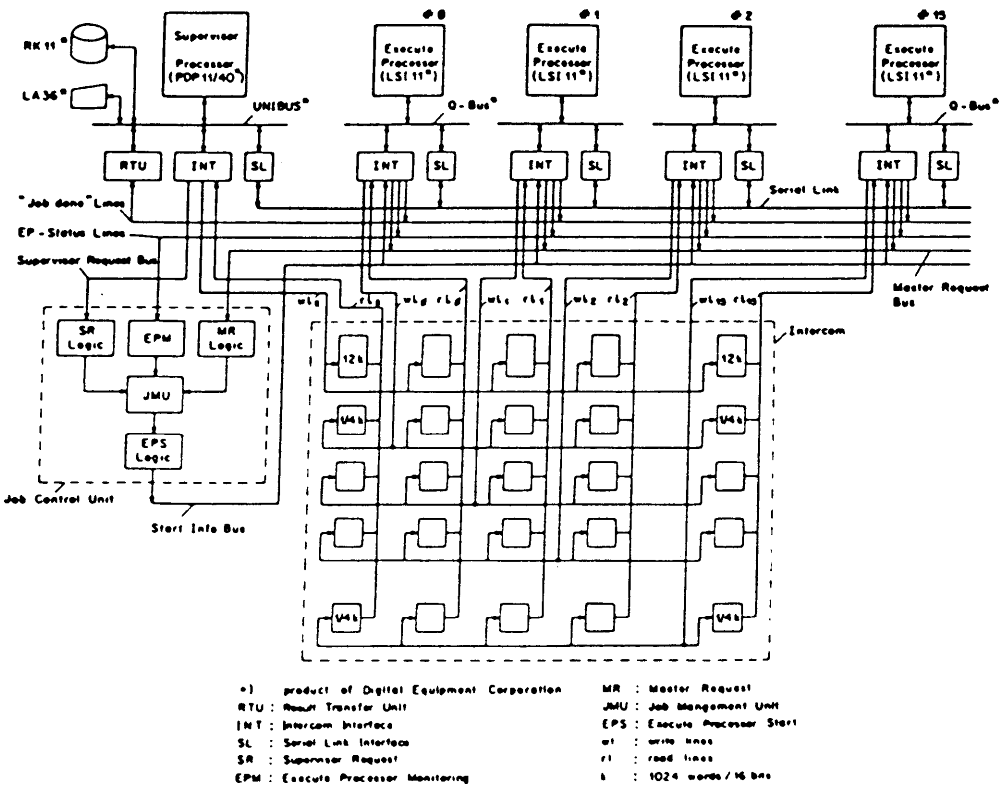


Figure 17: Hardware of Empress

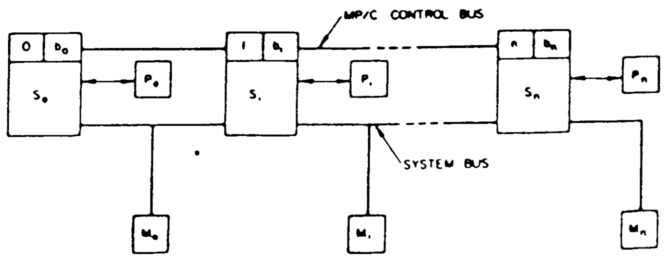


Figure 18: The Linear MP/C

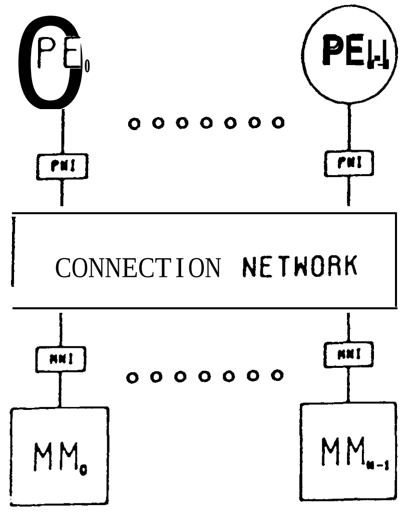


Figure 19: Block Diagram of the NYU Ultracomputer

a system bus. Process fork and join operations are implemented by bus switching as a means of partitioning and recombination of the address space. The bus can be opened between any two adjacent processor-memory pairs. Only the leftmost processor in each connected bus segment or partition is active, and can access all memory modules in that partition. All other processors in that partition are inactive. An active processor can activate an inactive processor by splitting the bus segment. Conversely, an active processor may deactivate itself by reconnecting its partition to the one on the left. This ability to partition and reconnect dynamically is best suited to execute tree algorithms, divide-and-conquer algorithms, and database functions. Multi-dimensional MP/C machines in which each row or column is a switchable bus, are also proposed.

3.17 Ultra

The NYU Ultracomputer [GGK 83] is a shared-memory MIMD parallel machine composed of thousands of autonomous processing elements (PE's). By the use of an enhanced message switching network with the geometry of an Omega-network, it can approximate the ideal behavior of Schwartz's paracomputer model of computation which permits every PE to read or write a shared memory cell in one cycle. The Omega-network also implements the fetch-and-add operation used as the synchronization primitive.

Its architecture consists of thousands of PE's connected through a connection network to thousands of memory modules. Each PE is a high-speed VLSI floating point processor. It can also support the fetch-and-add operation: a PE will continue execution of the instruction stream immediately after issuing a request to fetch a value from central memory, the target register would be marked *locked* until the requested value is returned from

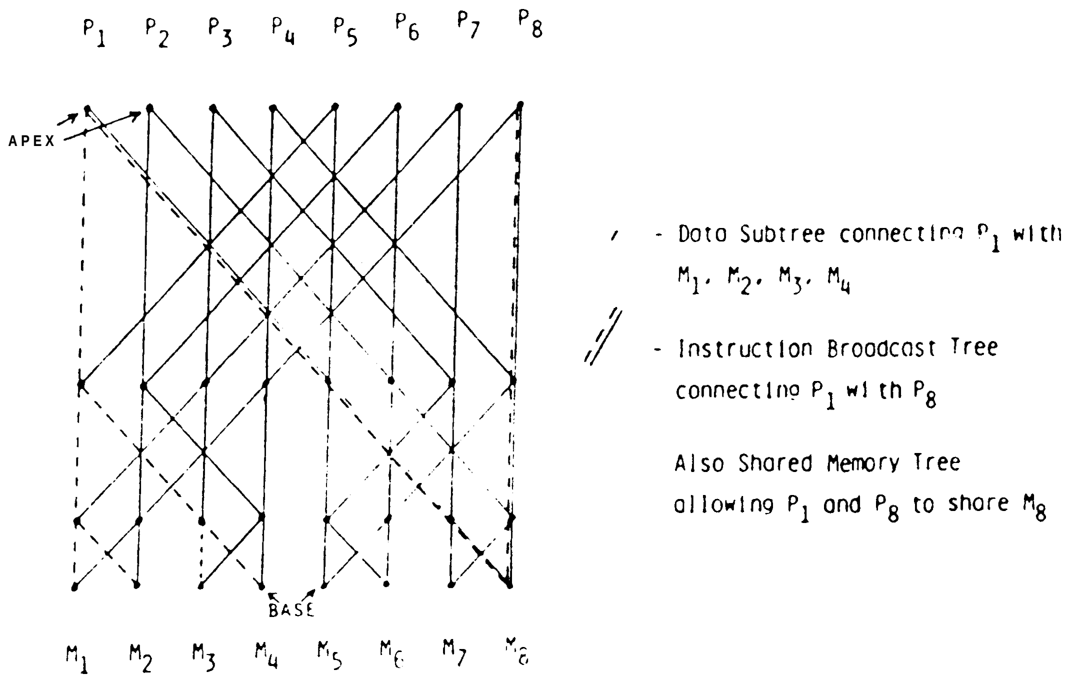

P1 P2 P3 P4 P5 P6 P7 P8
APEX
BASE
M1 M2 M3 M4 M5 M6 M7 M8
- Data Subtree connecting P1 with M1, M2, M3, M4
- Instruction Broadcast Tree connecting P1 with P8
Also Shared Memory Tree allowing P1 and P8 to share M8

Figure 20: TRAC's Banyan Network

memory; an attempt to use a locked register would suspend execution. The connection network is an enhanced message switching Omega-network. Each switch in the network has a queue and an internal adder to support the fetch-and-add operation. Simultaneous accesses to a common memory cell can be detected in the switch and are combined to a single fetch-and-add instruction. The memory unit also has an adder to implement the fetch-and-add instruction.

3.18 TRAC

The Texas Reconfigurable Array Computer (TRAC) [SUK 80] is an experimental computer system currently being built at the University of Texas at Austin. The uniqueness and the potential capabilities of TRAC arise from its interconnection network; a dynamically reconfigurable banyan network. The banyan network serves to partition and to configure the processor, memory and I/O resources of the system into different architectural organizations under software control. Within a partition, TRAC is varistructured in that regardless of the data structure requirements for the task, any data width or architecture may be used. Independent or interacting tasks can all be running simultaneously on the same computer. The architecture is also virtual in that user programs can be oblivious of the specific set of memory and processor modules used.

Inside the SW-banyan network, the nodes can be configured to form three types of subtree: data trees, instruction trees, and shared memory trees. Besides shared memory, another means of processor-processor communication is packet switching. The packet transmission occur as a background activity so that they do not interfere with other activity.

TRAC subsystems can be architectured to implement multiple models of computation: process forking and joining, task pipelining, data-flow, vector parallelism, and synchronous parallelism.

4 Different Dimensions of Concurrent Architectures

In the last section, 18 different concurrent architectures were described. They are so different in structure that it is hard to classify them in any single way. Different models of computation used by Haynes et. al. have already been described in section 2.0, there are four other dimensions that can be identified to describe these architectures. They are interconnection network, processing element, memory system, and application areas. In this section, these five dimensions will be used to classify the 18 architectures described in the previous section.

4.1 Models of Computation

The six different models of computation described by Haynes et. al. are multiple special-purpose functional units (or pipeline), associative processors, array processors, data flow computers, functional programming language processors, and multiple processors.

4.2 Interconnection Network

In the architectures that have been discussed, there are many types of interconnection network. Systolic arrays are connected in a pipelined fashion. STARAN has its own FLIP network. MPP and Illiac IV and MPP are mesh-connected. BSP and S-l use full crossbar. CHiP uses the switch lattice. NON-VON is tree structured. Cm* and MP/C are bus oriented. Cedar, FMP, and Ultra use the Omega network. HEP uses a pipelined switch. Empress has a quadratic memory matrix. TRAC has a 2-3 SW Banyan Network. Some of these interconnection networks can again be divided into either central or distributed control. Reconfigurability is a feature of some of the networks, which allow them to reconfigure the system resources dynamically to match the need of the problem. For multi-stage networks, three types of switching modes are possible: circuit, message, and packet. The purpose of the interconnection network is for the communication among processors (P-P), or processor to memory (P-M), or both.

Table 1: Models of Computation

	Pipeline	Associative Memory	SIMD	Dataflow	FP Language	Multiple Processor
Systolic	x					
STARAN		x				
Illiac IV			x			
BSP			x			
MPP			x			
CHiP			x			
NON-VON			x			
DDSP				x		
SERFRE					x	
Cedar						x
FMP						x
s-1						x
Cm*						x
HEP						x
Empress						x
MP/C						x
Ultra						x
TRAC	x		x	x		x

Table 2: Interconnection N twork

	Type	Control	Reconfig.	Switching	Communication
Systolic	Pipeline		N		P-P
STARAN	FLIP		N		P-P
Illiac IV	Mesh		N		P-P
BSP	Cross bar	Central	N	Circuit	P-M
MPP	Mesh		N		P-P
CHiP	Sw. lattice	Central	Y		P-P
NON-VON	Tree		Y		P-P
DDSP	Bus	Distributed	N		P-P
SERFRE	Bus		N		P-P
Cedar	Omega	Distributed	N	Circuit	P-M
FMP	Omega	Distributed	N	Circuit	P-M
S-l	Cross bar	Central	N	Circuit	P-M
Cm*	Bus	Distributed	N	Packet	P-M
HEP	Pipe. Switch	Distributed	N	Packet	P-M
Empress	Quad. Matrix	Central	Y		P-P
MP/C	Bus	Distributed	Y		P-M
Ultra	Omega	Distributed	N	Message	P-M
TRAC	2-3 Banyan	Central & Distributed	Y	Packet & Circuit	P-P, P-M

Table 3: Processing Element

	Number	Power	Size (bits)
Systolic	variable		
STARAN	32 x 256	simple	1
Illiac IV	64	3-5 MFLOPS	64
BSP	16	1-4 MFLOPS	48
MPP	16384	simple	1
CHiP	256 - 64K	microprocessor	
NON-VON	> 16K	simple	8
DDSP	l-32	2.2 MFLOPS	
SERFRE	variable		
Cedar	128 x 16	2.5 MFLOPS	32
FMP	128	10 MFLOPS	64
s-1	16	3 MFLOPS	64
Cm*	variable	LSI-11	16
HEP	variable	10 MIPS	64
Empress	variable	LSI-11	16
MP/C	variable		
Ultra	4096	VLSI, fast FP	
TRAC	16	microprocessor	8

4.3 Processing Element

The number of processing elements used in each architecture varies from 1 for the DDSP to 64K for the CHiP. Most architectures allow variable number of processing elements. Requirements on the processing elements also vary. STARAN, MPP, and NON-VON use very simple processors. CHiP, Cm*, Empress, and TRAC use off-the-shelf microprocessors or LSI-11. Others use powerful custom made processors. The word length of the architectures ranges from 1 bit for STARAN and MPP, to 64 bits for Illiac IV, Cedar, FMP, S-l, and HEP.

4.4 Memory System

Memory system can be either shared or local. In shared memory, different processors can access the same memory cell for communication and synchronization. Data stored in local memory can only be accessed by the processor attached to it. Some local memory are also used as instruction cache to reduce the traffic in the interconnection network. STARAN and DDSP have associative memory for content-addressable memory and matching store, respectively. Ultra has adders in the memory system to support the fetch-and-add operation. TRAC has index registers residing in the memory modules so that a shorter 8-bit macro-instruction can be sent by the processor instead of a longer 16-bit full address. References to locations in memory modules are made by specifying one of the

Table 4: Memory System

	Shared	Local	Associative	Extended Function
Systolic	x			
STARAN	x		x	
Illiac IV	x	x		
BSP	x			
MPP	x			
CHiP	x			
NON-VON	x			
DDSP		x	x	
SERFRE		x		
Cedar	x	x		
FMP	x	x		x
S-l	x	x		
Cm*	x	x		
HEP	x	x		
Empress		x		
MP/C	x	x		
Ultra	x	x		x
TRAC	x	x		x

index registers.

4.5 Application Areas

These 18 architectures are designed for different applications. In this section, a few important and representative areas are listed: general purpose, scientific, data base, image or signal processing, simulation, testbed, and divide-and-conquer.

5 Conclusions

In this report, a survey of concurrent architectures is presented. Although the survey is not complete, it does cover a wide spectrum of both commercial and research architectures. It is not the purpose of this report to give a detail summary of each architecture, but to give the reader a general idea of the current status of the research in concurrent architecture. Different architectures are specified using five dimensions: models of computation, interconnection network, processing element, memory system, and application areas. Other dimensions, especially software aspects, can also be used, for instance, languages, operating system, scheduling method, communication and synchronization method.

Table 5: Application Areas

	General Purpose	Scientific	Database	Image/Signal Processing	Simulation	Testbed	Divide & Conquer
Systolic		x					
STARAN				x	x		
Illiac IV		x					
BSP		x					
MPP				x			
CHiP		x					
NON-VON	x	x	x	x			x
DDSP				x			
SERFRE	x						
Cedar	x						
FMP		x			x		
s-1	x	x		x			
Cm*					x	x	
HEP	x	x					
Empress		x			x		
MP/C			x				x
Ultra	x						
TRAC		x	x		x	x	

References

[AG 82] B.W. Arden, and R. Ginosar, "MP/C: A Multiprocessor/Computer Architecture," *IEEE Trans. on Computers,* Vol. C-31, No. 5, May 1982, pp. 455-473.

[Bat 74] K.E. Batcher, "STARAN Parallel Processor System Hardware," *AFIPS Conf. Proc.,* Vol. 43, 1974 NCC, pp. 405-410.

[Bat 82] K.E. Batcher, "MPP: a supersystem for satellite image processing," *AFIPS Conf. Proc.,* Vol. 51, 1982 NCC, pp. 185-191.

[BBB 82] R.E. Buehrer, H.J. Brundiers, H. Benz, B. Bron, H. Friess, W. Haelg, H.J. Halin, A. Isacson, and M. Tadian, "The ETH-Multiprocessor EMPRESS: A Dynamically Configurable MIMD System," *Trans. on Computers,* Vol. C-31, No. 11, Nov 1982, pp. 1035-1044.

[BDM 72] W .J. Bouknight, S.A. Denenberg, D.E. McIntyre, J.M. Randall, A.H. Sameh, and D.L. Slotnick, "The Illiac IV System," *Proc. IEEE,* Vol. 60, NO. 4, Apr. 1972, pp. 369-379.

· [Fly 72] M.J. Flynn, "Some Computer Organizations and Their Effectiveness," *IEEE Trans. on Computer,* Vol. C-21, No. 9, Sept. 1972, pp. 948-960.

[GGK 83] A. Gottlieb, R. Grishman, C.P. Kruskal, K.P. McAuliffe, L. Rudolph, and M. Snir, "The NYU Ultracomputer - Designing an MIMD Shared Memory Parallel Computer," *IEEE Trans. on Computers,* Vol. C-32, No. 2, Feb. 1983, pp. 175-189.

[GKL 83] D.Gajski, D. Kuck, D. Lawrie, and A. Sameh, "Cedar - A Large Scale Multiprocessor," *Proc. of the 1983 International Conf. on Parallel Processing,* pp. 524-529.

[HLS 82] L.S. Haynes, R.L. Lau, D.P. Siewiorek, and D.W. Mizell, "A Survey of Highly Parallel Computing," *Computer,* Jan. 1982, pp. 9-24.

[HNI 82] E.B. Hogenauer, R.F. Newbold, and Y.J. Inn, "DDSP-A Data Flow Computer for Signal Processing," *Proc. of the 1982 International Conf. on Parallel Processing,* pp. 126 -133.

[KS 82] D.J. Kuck, and R.A. Stokes, "The Burroughs Scientific Processor (BSP)," *IEEE Trans. on Computers,* Vol. C-31, No. 5, May 1982, pp. 363-376.

[Kun 82] H.T. Kung, "Why Systolic Architectures?" *Computer,* Jan 1982, pp. 37-46.

[Lun 85] S.F. Lundstrom, "A Decentralized Control, Highly Concurrent Multiprocessor," *IEEE Proceedings of the 12th Annual International Symposium on Computer Architecture,* June 17-19, 1985, pp. 145-151.

[SFS 77] F.J. Swan, S.H. Fuller, and D.P. Siewiorek, "Cm*- A modular, multi-microprocessor," *AFIPS Conf. Proc.,* Vol. 46, 1977 NCC, pp. 637-644.

[Smi 78] B.J. Smith, "A Pipelined, Shared Resource MIMD Computer," *Proc. of the 1978 International Conf. on Parallel Processing, 1978,* pp. 6-8.

[Sny 82] L. Snyder, "Introduction to the Configurable Highly Parallel Computer," *Computer,* Jan. 1982, pp. 47-56.

[SUK 80] M.C. Sejnowski, E.T. Upchurch, R.N. Kapur, D.P.S. Charlu, and G.J. Lipovski, "An Overview of the Texas Reconfigurable Array Computer," *AFIPS Conf. Proc.,* Vol. 49, 1980 NCC, pp. 631-641.

[Tom 67] R.M. Tomasulo, "An Efficient Algorithm for Exploiting Multiple Arithmetic Units," *IBM Journal,* Vol. 11, Jan. 1967.

[Vil 82] F.Y. Villemin, "SERFRE : A General-Purpose Multi-processor Reduction Machine," *Proc. of the* 1982 *International Conf. on Parallel Processing,* pp. 140 -141.

[WC 79] L.C. Widdoes, Jr., and S. Correll, "The S-l Project: Developing High-Performance Digital Computers," *Energy and Technology* Review, Sept.8 1979.