

Associative Processor Architecture—A Survey

S.S. YAU

and

H.S. FUNG

Departments of Computer Science and Electrical Engineering, Northwestern University, Evanston, Illinois 60201

A survey of the architecture of various associative processors is presented with emphasis on their characteristics, categorization, and implementation, and especially on recent developments. Based on their architecture, associative processors are classified into four categories, namely fully parallel, bit-serial, word-serial and block-oriented. The fully parallel associative processors are divided into two classes, word-organized and distributed logic associative processors.

Keywords and Phrases: Associative processors, computer architecture, categorization, implementation, hardware, large-scale integration, fully parallel, bit-serial, word-serial, block-oriented, distributed logic

CR Categories: 3.80, 6.22, 6.30, 6.31, 6.32, 6.33, 6.34

INTRODUCTION

An *associative processor* can generally be described as a processor which has the following two properties: 1) Stored data items can be retrieved using their content or part of their content (instead of their addresses); and 2) data transformation operations, both arithmetic and logical, can be performed over many sets of arguments with a single instruction. Because of these parallel processing characteristics, associative processors have a much faster data processing rate than conventional sequential computers, and hence are more effective in handling many types of information processing problems such as information storage and retrieval of rapidly changing databases, fast search of a large database, arithmetic and logical operations on large sets of data, control and executive functions in large-scale computer systems, radar signal tracking and processing, and weather prediction computations. However, because of their relatively high implementation cost,

associative processors are usually used in conjunction with standard sequential computer systems so that many required high-speed parallel processing tasks which cannot be effectively executed by sequential processors are performed by associative processors. Because of the rapid development of large-scale integrated-circuit (LSI) technology, the implementation cost of associative processors will be greatly reduced and it is anticipated that associative processors will be used more extensively for enhancing the performance of many special-purpose and general-purpose computer systems. Although there have been several papers providing either tutorials or literature surveys on associative processors [1-5], a number of new developments have not been described in any of the previous survey papers. In this paper, we will present a survey of the architecture of various associative processors, with emphasis on their characteristics, categorization, and implementation, and especially recent developments.

Copyright © 1977, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

CONTENTS

INTRODUCTION

1 GENERAL DESCRIPTION

Associative Memories

Associative Processors

Hardware Implementation

2 FULLY PARALLEL ASSOCIATIVE PROCESSORS

Fully Parallel Word-Organized Associative Processors

Distributed Logic Associative Processors

Lee's Distributed Logic Associative Processor and Its Modifications

Parallel-Element Processing Ensemble (PEPE)

3 BIT-SERIAL ASSOCIATIVE PROCESSORS

The STARAN Processor

Other Bit-Serial Associative Processors

Associative Linear Array Processor (ALAP)

ECAM

Other Developments

Byte-Serial Word-Parallel Associative Processors

Data Manipulators

4 WORD-SERIAL ASSOCIATIVE PROCESSORS

5 BLOCK-ORIENTED ASSOCIATIVE PROCESSORS

SUMMARY

ACKNOWLEDGMENT

REFERENCES

1. GENERAL DESCRIPTION

In general, the architecture of an associative processor can be described as shown in Figure 1. It consists of an associative memory, arithmetic and logic unit (ALU), control system, instruction memory, and an input/output interface. The major difference between an associative processor as shown in Figure 1 and a standard sequential processor is the use of an associative memory or its equivalent instead of a location-addressed memory. Because of this difference, all the other blocks of an associative processor are also different from those of a standard sequential processor. Furthermore, the associative memory has a major impact on the architecture of an associative processor, and the architecture of an associative processor can be classified on the basis of the organization of its associative memory. Thus, we would first like to describe associative memories briefly.

Associative Memories

An *associative memory* [1-10] can be defined as a memory system with the property that stored data items can be retrieved by their content or part of their content (that is, by the first property of an associative processor). An associative memory has been also called *catalog memory* [11], *content-addressed memory* [12], *data-addressed memory* [13], *parallel search memory* [14], *search memory* [15-17], *search associative memory* [18], *content-addressable memory* [19], *distributed logic memory* [20], *associative push-down memory* [21], and *multi-access associative memory* [22].

From the hardware point of view, in order to retrieve stored data items by their content or part of their content, one must be able to access the memory words by matching their content or part of their content with the given search-key words, instead of by an address as in a location-addressed memory. The basic memory element of the associative memory is called the *bit-cell*. It has the property that one-bit information can be written in, read out, and compared to the interrogating information. The search operations, which consist of masking and comparison, are executed in a fashion that depends on the organization of the associative memory. The search-key word can be compared to all the words in the memory through the interrogating bit drives and the comparison logic circuitry. The possibility of matching multiple words to a search-key word requires that the associative memory have some method of tagging all the matched words. The tag function and matched-word indication are

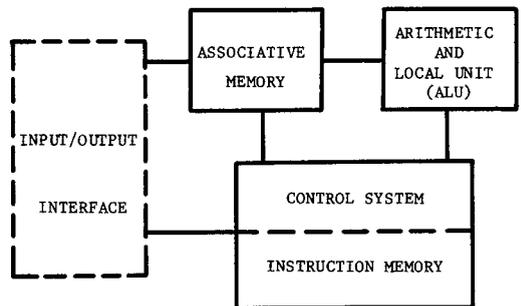


FIGURE 1. Block diagram of an associative processor.

performed by the so-called word-match tag networks. Those matched words can be accessed with a single instruction [1-3, 5, 23]. It is noted that a memory can have the associative property by performing either parallel-by-bit comparison (word-parallel or word-serial) or serial-by-bit comparison (bit-serial). For an associative memory, the following comparison operations can be performed:

equal	not equal
less than	greater than
not greater than	not less than
maximum value	minimum value
between limits	not between limits
next higher	next lower

The operation of an associative memory may be illustrated by the example of a personnel file search, as shown in Figure 2.

The query may require that data on all employees with a salary of more than \$1000 per month and less than or equal to \$1500 per month be searched. This will be accomplished by performing a *greater than* search and a *not greater than* search on the salary field of the file. Each of these two searches is performed in parallel. To set up the parallel search, a search-key word is loaded with the desired salary range information for comparison. A mask is included to mask the search-key word so that only the desired fields may be searched. A match-word indication is required to indicate the results of the search. The simplest indication consists of a bit for each of the words, indicating whether it is identical to the search-key word or not. A 1-bit indicates a match and a 0-bit a mismatch. Considering the

First Search Key Word	0 0 0 1000 0						
Second Search Key Word	0 0 0 1500 1						
Mask	00 00 0 00 00 11 11 1						
					Initial	After First Search	After Second Search
SMITH BILL W	1	PROGRAMMER	1200	0	1	1	
JONES BOB E	1	ENGINEER	1500	0	1	1	
FAIR MARIGOLD	0	CLERK	1000	0	0	0	
HALL TOM	1	ENGINEER	1470	0	1	1	
LAWRENCE LINDA	0	ACCOUNTANT	2000	0	1	0	
RALNER TED	1	ENGINEER	1800	0	1	0	
BLANAGAN ED	1	CONSULTANT	2100	0	1	0	
WHITE E MARY	0	OPERATOR	800	0	0	0	
ROGERS ALBERT	1	ENGINEER	1200	0	1	1	
ROPLEY HALEN	0	CLERK	800	0	0	0	
.	
.	
.	
MARTIN JOE K	1	PROGRAMMER	1100	0	1	1	
Name	Sex	Title	Salary	Indicator			

matched-
word
indication

FIGURE 2. Example of the operation of an associative memory.

above query, the first search-key word is loaded with the salary figure (1000) and the indicator (0) for comparison, and the operation is "greater than." Initially the indicator field of each word in the associative memory is 0. After the first search, the matched-word indication will signal the indicator field by setting the indicator-field to 1 to memorize the matched words. The second search-key word is loaded with the salary figure (1500) and the indicator (1) for comparison, and the operation is "not greater than." The result is shown in the matched-word indication. Data on all the employees with a salary of more than \$1000 per month and less than or equal to \$1500 per month can be printed out through the output circuit.

Associative Processors

From the operational point of view, an associative processor can perform other complicated data transformation operations in addition to the comparison operations that can be performed by its associative memory. For instance, the matched words in the associative memory can be retrieved serially to the ALU through the output circuit of the associative memory under the control of the control system. The ALU performs the specified data transformation operations and the result is then stored in the associative memory, if necessary.

From the point of view of architecture, associative processors belong to the general category called *SIMD* (*Single Instruction Stream Multiple Data Stream*) machines [5, 24]. An SIMD machine is a computer with a single instruction which instructs more than one processing element, all of which either execute or ignore the current instruction. An associative processor is an SIMD machine whose processing elements and data addressing satisfy the two properties mentioned at the beginning of this paper. There is another well-known class of SIMD machines called array processors, such as ILLIAC IV [25]. An array processor is an SIMD machine in which data are processed by its processing elements in parallel, but data are addressed by their addresses rather than by their content or part of their content. In this paper, our discussion

will focus on the architectural aspect of associative processors.

The architecture of associative processors can generally be classified into four categories according to the comparison process of their associative memories. The four categories are *fully parallel*, *bit-serial*, *word-serial*, and *block-oriented* associative processors. There are two types of fully parallel associative processors: *word-organized* and *distributed logic* types. In the former the comparison logic is associated with each bit-cell of every word, and the logical decision is available at the output of every word; in the latter, the comparison logic is associated with each character-cell (for a fixed number of bits) or with a group of character-cells. In a bit-serial associative processor, only one bit-column (also called bit-slice) of all the words is operated on at a time. (For this reason, a bit-serial associative processor is also called a *bit-serial word-parallel* associative processor.) The fully parallel and bit-serial associative processors are the two most important categories that have been so far developed. PEPE (Parallel Element Processing Ensemble) [26-31] and STARAN [32-35] are the best-known fully parallel and bit-serial associative processors, respectively. As we will see later, an important part of a STARAN-type bit-serial associative processor is the *permutation network* (also called *flip network* or *interconnection network*) which is the functional block for communication between the memory modules and the processing element modules. This permutation network is used not only in STARAN-type bit-serial associative processors, but also in array processors for preparing appropriate operands for execution. In many applications, such as air traffic control systems, matrix computation, weather forecasting, etc., preparing appropriate operands using a permutation network is very useful. Feng [36-40] proposed a generalized permutation network, the *data manipulator*. There are two types of data manipulators, the *line manipulator* and the *page manipulator*. The line manipulator is designed in a bit-serial fashion. The page manipulator operates in a fully parallel fashion; it is similar to the line manipulator except that there are n line manipulator circuits interconnected

together, where n is the word width. The implementation cost of a page manipulator is obviously greater than that of a line manipulator, but its use enables a bit-serial associative processor to operate like a fully parallel associative processor for a much lower cost than that of a fully parallel associative processor.

A *word-serial* associative processor essentially represents a hardware implementation of a simple program loop for search. The important factor contributing to the relative efficiency of this approach as compared to programmed search in a standard sequential processor is that the instruction decoding time is greatly reduced, since only a single instruction in the word-serial associative processor is required to perform the search operation. A *block-oriented* associative processor can be implemented by using a logic-per-track rotating memory which consists of a head-per-track disk with some logic associated with each track.

Before we discuss each of these categories in detail, we would like to discuss briefly the hardware implementation of associative memories and associative processors.

Hardware Implementation

The first associative memory was developed by Slade and McMahon [11] in 1956 using cryotrons. Since then, associative memories have been implemented using tunnel diodes [41, 42], evaporated organic diode arrays [7], magnetic cores [41, 43–48], plated wires [41, 49], semiconductors [50], transfluxors [51], biax cores [52], laminated ferrites [53], magnetic films [54], solenoid arrays [55], bicore thin-film sandwiches [41], multi-aperture logic elements [56], and integrated circuits [57]. The capacity of these associative memories is usually limited by factors such as half-select noise which limit the word length, and interrogation drive problems which limit the number of words. Because of these limitations and because of high implementation cost, most associative memories in early years had small capacity, say up to 1K words with length up to 100 bits [45].

The first associative processor was designed by Behnke and Rosenberger [58] in 1963 using cryotrons. Since then a number

of laboratory models of associative processors have been built using various types of associative memories. However, associative processors were not put to practical use until the development of PEPE [26–31] and STARAN [32–35]. In these systems, associative memories have become larger and more flexible due to the development of new architectural concepts and the use of LSI technology. For example, in PEPE there are a number of processing elements each of which contains a simple $1K \times 32$ -bit random-access memory, called the element memory, which is shared on a cycle-stealing basis by the arithmetic unit, correlation unit, and associative output unit in the processing element to perform associative processing. In each associative array module of STARAN, a so-called multidimensional access memory implemented by a 256×256 -bit random-access memory is used to accommodate both bit-slice accesses for associative processing and word-slice accesses for input/output. Recently Anderson and Kain [59] at Honeywell presented the Extended Content Addressed Memory, called ECAM, used in the high-speed database processing environment by the U.S. Air Force. This system can operate on databases up to 10^9 bits in size, and was originally based on the storage fabrication technique called the superchip technique, which is a device-fault-tolerant method of connecting and using a large number of individual memory arrays on a single chip. However, other LSI chips may also be used for constructing ECAM. Another associative processor using LSI technology is the Associative Linear Array Processor (ALAP) developed by Finnilla at Hughes Aircraft Co. [60], which will be discussed later. These appear to be promising approaches to making large-scale associative memories practical.

2. FULLY PARALLEL ASSOCIATIVE PROCESSORS

Fully Parallel Word-Organized Associative Processors

As mentioned before, the major characteristic of a fully parallel word-organized associative processor is that the comparison logic is associated with each bit-cell of every word of its associative memory. Thus its

comparison process is performed in a parallel-by-word and parallel-by-bit fashion. The general organization of a fully parallel word-organized associative processor is shown in Figure 3, in which each crosspoint represents a bit-cell of its associative memory. Although the operations of a fully parallel word-organized associative processor are simplest and fastest compared to other types of associative processors, its hardware is also the most complicated because each bit-cell has to contain the comparison logic. Because of its hardware complexity, this type of associative processor was developed only through the early stages. Many experimental models were developed and there were fully parallel word-organized associative memory systems using cryogenic components, [11, 13, 52, 61-66], magnetic cores [43, 44] and cutpoint cellular logic [67].

Distributed Logic Associative Processors

A distributed logic associative processor is a fully parallel character-oriented associative processor whose memory (usually called distributed logic memory [20]) has its comparison logic associated with each character-cell or each group of character-cells. A number of distributed logic associative processors have been developed.

The first associative processor of this type was proposed by Lee [68] and a number of its variations were presented later [69-75].

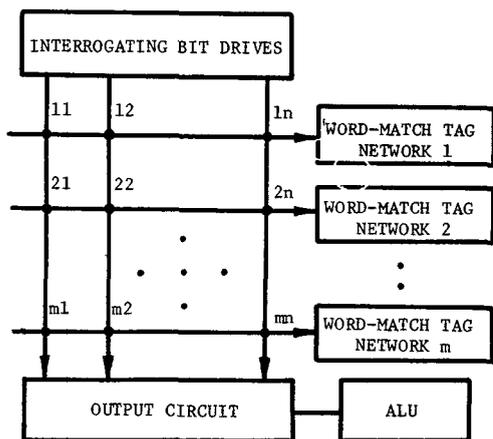


FIGURE 3. Fully parallel word-organized associative memory and ALU.

The best-known associative processor system of this type so far developed is the PEPE [26-31], developed by Bell Laboratories for the U.S. Army Advanced Ballistic Missile Defense Agency. These associative processors are now described.

Lee's Distributed Logic Associative Processor and Its Modifications

The distributed logic associative processor proposed by Lee [68] can be represented by the block diagram shown in Figure 4. Each character-cell has a single cell-state element (state part) S which may be either in an active state or in a quiescent state, and each character-cell also has a number of cell-symbol elements (symbol parts) X_1, \dots, X_n depending upon the size of the symbol alphabet. The cell-state element or cell-symbol element is a bistable device such as a flip-flop. Each character-cell stores one character symbol of information and can communicate with its two neighboring character-cells as well as with the control system. A string of information is therefore stored in a corresponding string of character-cells.

Each data block consists of a name string and an arbitrary number of parameter strings. Every name string is preceded by a tag α , and every parameter string is preceded by a tag β . When the input search key is a name string, the fully parallel distributed logic memory is expected to output all of the parameter strings associated with the name string. This is the so-called direct retrieval. On the other hand, when the input search key is a parameter string, the fully parallel distributed logic memory is expected to output all of the name strings associated with that parameter string. This is called cross-retrieval. In order to perform direct retrieval and cross-retrieval, each character-cell in the fully parallel distributed logic memory must have enough cell logic circuitry so that it can produce a *yes* or *no* answer to a simple question such as whether the symbol of the character-cell is A or not- A . If we want to retrieve all of the parameter strings whose name is AB , we will ask each character-cell whether its character symbol is A . If a cell answers

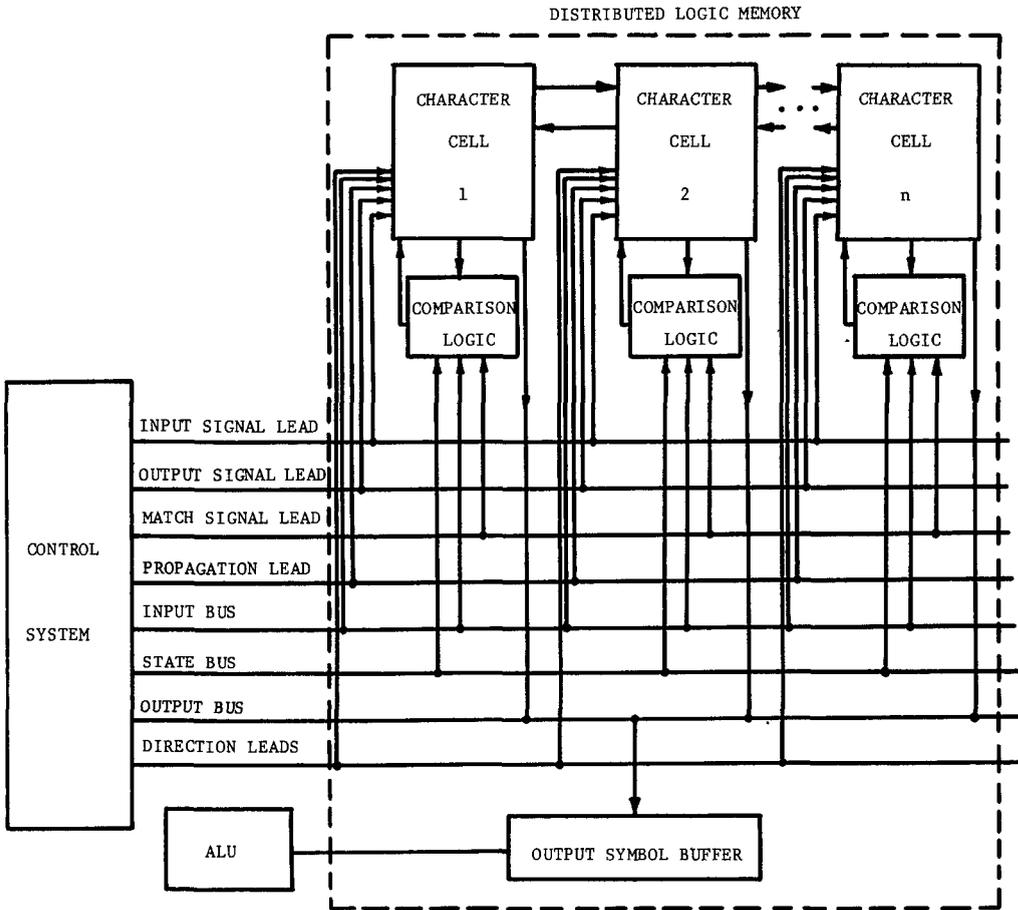


FIGURE 4. Fully parallel distributed logic associative processor.

yes, we also want each character-cell to have enough cell logic circuitry so that it can signal the next character-cell to be ready to determine whether the symbol of that character-cell is B . The character-cells which finally respond *yes* to the name string AB are now ready to signal all those character-cells storing all the parameter strings associated with the name string AB to output their contents.

Typical operations of a character-cell are changing state, transmitting state information to a neighboring character-cell, accepting data from the input bus, or putting its character symbol on the output bus. When a character-cell is in an active state and when the input signal lead is activated,

the symbol which is carried on the input bus is then stored in that character-cell. When a character-cell is in an active state, an output signal causes that character-cell to read out its symbol through the output bus and to store it in the output symbol buffer. Comparison operation is controlled by the match signal through the comparison logic of each character-cell. The store symbol of each character-cell is compared with the symbol carried on the input bus, and a signal from each matched character-cell is transmitted to one of its neighboring character-cells which then becomes active. The direction of transmission of the signals is controlled by the signal on the direction leads. All the character-cells evaluate and

act according to the input conditions (given by the input and state buses) independently and simultaneously.

Lee's system, consisting of 72 eight-bit character-cells has been built experimentally using cryogenics [75].

Several modifications of Lee's original system have been proposed. Lee and Paull [20] proposed a distributed logic memory using two cell-state elements instead of one for each character-cell, more control bus leads, and a threshold circuit. They defined the complex symbol of a character-cell which includes both the two cell-state elements and the cell-symbol elements of the character-cell. The matching process requires that an entire complex symbol be used for matching. They presented a more complicated design for a character-cell memory combined with an external control unit in order to have more capabilities to deal with problems such as cross-retrieval, erasing, gap closing, and preference, which appear in information retrieval.

In order to overcome the propagation timing problems, Gaines and Lee [72] proposed to redesign the logic circuitry using two different-purpose cell-state elements, called the match flip-flop and the control flip-flop, and adding a mark line to simultaneously activate all cells to the right of each active cell up to the first cell whose control flip-flop is set. Due to the control of the propagation of the marking signal, this memory system is capable of performing two new simultaneous operations, shifting and marking strings.

Crane and Githens [73] extended Lee's system to a two-dimensional distributed logic memory which can be used to perform highly parallel arithmetic operations through the use of a large number of identical processing units on many sets of data simultaneously, while retaining content-addressing capability to these data sets. Such an extension can be illustrated by the block diagram shown in Figure 5.

Edelberg and Schissler [76] recently proposed a distributed logic memory, called *intelligent memory*, which uses circulating serial storage loops and distributed processing logic. Each storage loop is a circulating shift register. All storage loops are shifted synchronously using a common clock. Processing logic is distributed between storage loops. In addition to the basic information storage operations, this memory performs associative searching, updating and retrieval. It is also capable of dynamically varying its loop size to accommodate varying data requirements.

Parallel Element Processing Ensemble (PEPE)

PEPE [26-31] is one of the two large-scale associative processors developed to date. Its basic concept was derived from Lee's distributed logic associative processor and was originally developed by Bell Laboratories for the U.S. Army Advanced Ballistic Missile Defense Agency [26-28]. A second model of PEPE with both architectural and

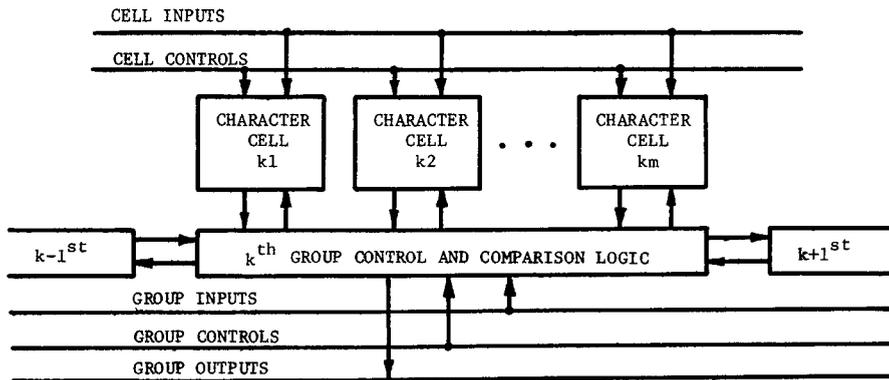


FIGURE 5. Fully parallel two-dimensional distributed logic associative memory.

circuit technology improvements is being developed by the Agency [29-31]. The description of PEPE presented here is primarily that of the current model.

PEPE is composed of the following functional subsystems: an output data control, an element memory control, an arithmetic control unit, a correlation control unit, an associative output control unit, a

control system, and a number of processing elements. Each processing element consists of an arithmetic unit, a correlation unit, an associative output unit and a 1024×32-bit element memory. In addition, there are primary power and signal distribution subsystems to convert and route power, and control and data signals between various functional subsystems. Note that the num-

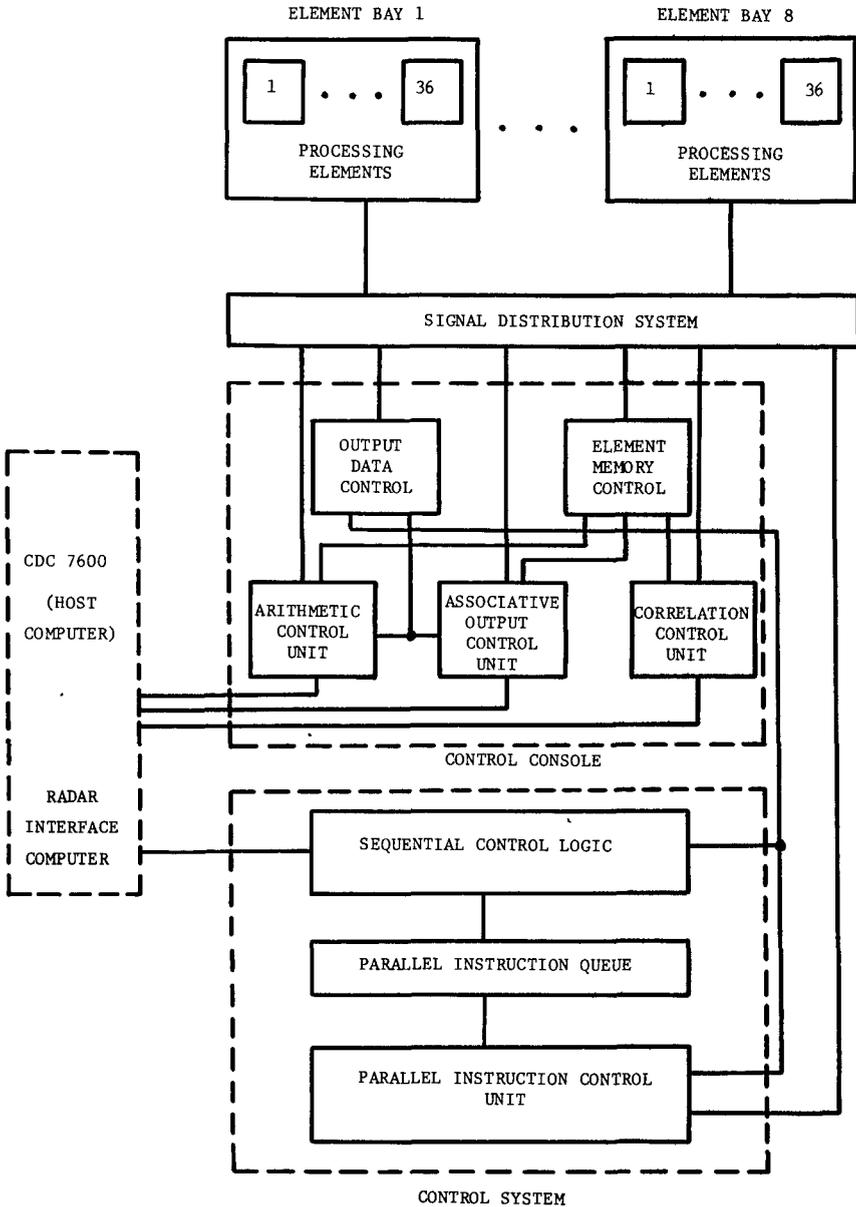


FIGURE 6. Parallel element processing ensemble (PEPE), a modified distributed logic associative processor.

ber of processing elements used in PEPE is variable and may be increased or decreased to meet the requirements of the application. This variability has no impact on PEPE system performance, except that enough processing elements must be available to accommodate the expected number of objects to be tracked. A PEPE with 288 processing elements organized into eight element bays was presented in [29].

The block diagrams of PEPE and its processing elements are shown in Figures 6 and 7, respectively. The processing elements are the main computational component of

PEPE. Selected portions of the data-processing load are loaded from the host computer (a CDC 7600) to the processing elements. The loading selection process is determined by the inherent parallelism of the task and by the ability of PEPE's unique architecture to manipulate the task more efficiently than the host computer. Each processing element is delegated the responsibility of an object under observation by the radar system, and each processing element maintains a data file for specific objects within its memory and uses its arithmetic capability to continually update its respective file.

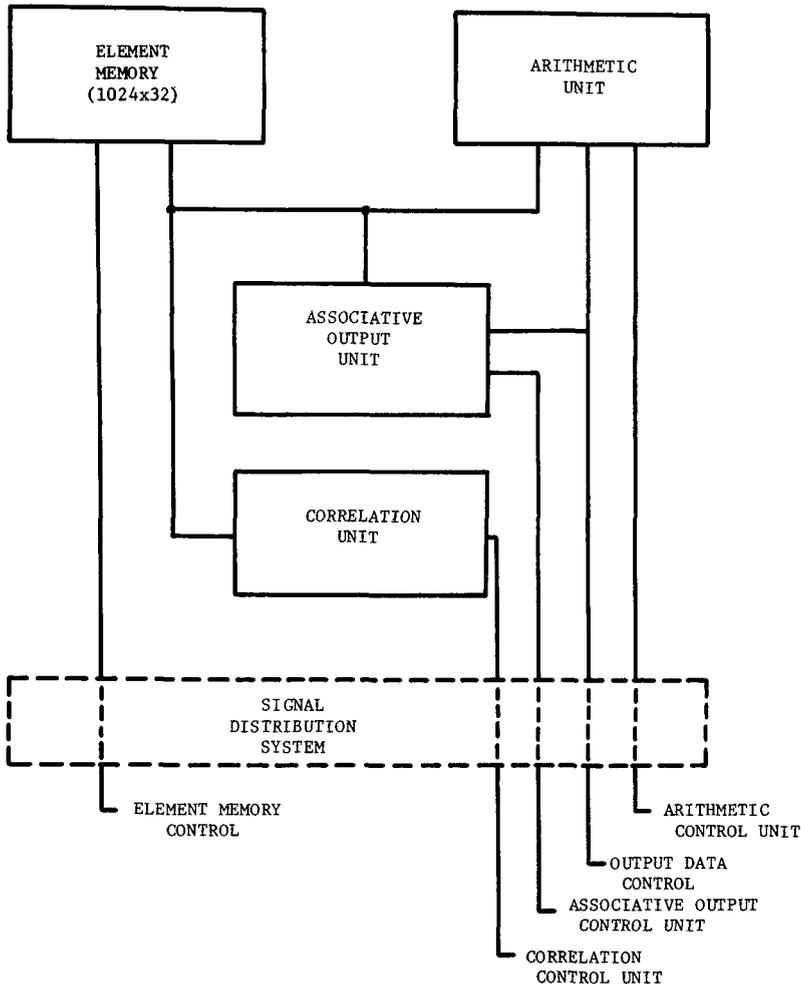


FIGURE 7. Processing element of the PEPE.

3. BIT-SERIAL ASSOCIATIVE PROCESSORS

Because of the expensive logic in each memory bit and the communication problems in fully parallel associative processors, the bit-serial word-parallel associative processor using the concept of parallel processing with vertical data (one bit-column of a large number of words is being processed at a time) was introduced by Shooman in 1960 [77]. His system is essentially a hypothetical vertical data processing computer (referred to as an orthogonal computer) which embodies both vertical data processing and conventional (referred to as horizontal data processing) techniques. Shooman also gave descriptions and algorithms for several vertical data processing instructions. Since the number of words to be processed is usually larger than the number of bits in each word, this approach represents a compromise between fully parallel and word-serial associative processing.

Since then, this concept has resulted in many proposals for associative processors. Kaplan [16] proposed a bit-serial associative memory which he called a search memory; this memory may be used as a subsystem for a general-purpose computer. The main memory may communicate via a memory register with the search memory subsystem, accumulator, arithmetic unit, control unit, and input/output unit. The match logic to execute search operations was placed in the search memory subsystem. Ewing and Davies [49] proposed the design logic of a bit-serial associative processor. The block diagram of a bit-serial associative memory with the ALU is shown in Figure 8. In this memory, storage for one bit is provided at each intersection of a word line and a bit line, and only one bit-column is operated on at a time. The particular bit-column is selected by the bit-column-select logic. A pulse on a bit line causes a signal to be emitted by each word line. The signals are transmitted through the word lines to the sense amplifiers. The word logic associated with each word line gives the ability to perform associative processing. This logic is identical for all words and consists of a sense amplifier, storage flip-flops, a write amplifier, and control logic. The storage

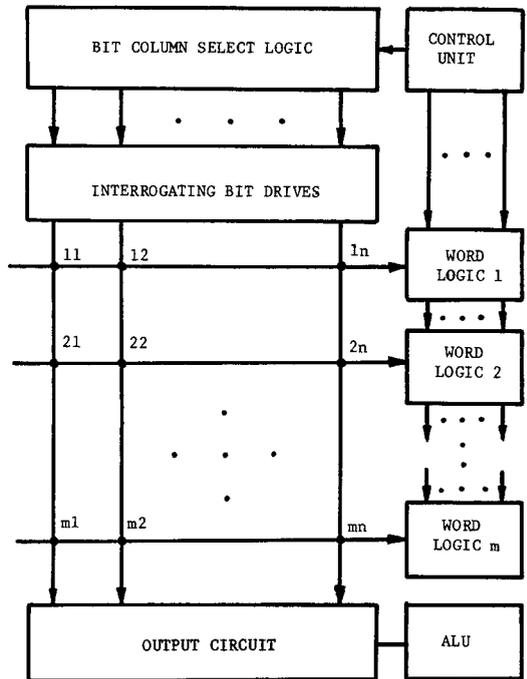


FIGURE 8. Bit-serial associative memory and ALU.

flip-flop remembers the match state from one interrogating bit to the next. The output of the sense amplifier determines the state of the storage flip-flops in various ways as determined by the control signals from the control unit. The capability of the storage flip-flops to act as shift registers provides the communication link between adjacent words. Such a bit-serial associative processor can be considered an external-logic associative processor, in contrast to a distributed logic associative processor.

Chu [8] proposed the implementation of a bit-serial associative memory which makes use of conventional destructive-readout magnetic memory elements. This memory has two-dimensional read/write capability, resulting in two word lengths: a short-word length which is the number of bits in a word, and a long-word length which is the number of words in a bit-column, since the number of bits in a word is usually smaller than the number of words. This memory can read or write in either the horizontal or the vertical direction of the array; the two are called the short-word mode and the long-

word mode, respectively. The short-word mode is the conventional memory organization. The long-word mode is equivalent to the bit-serial associative technique.

Bit-serial associative processing has been implemented through the use of $2\frac{1}{2}$ D core search memory [78-79]. Goodyear Aerospace Corporation [80-81] developed a modular plated-wire implementation of a bit-serial associative processor which uses the so-called processor modules as basic building blocks. Each processor module contains 256 plated-wire processing elements. Each plated-wire processing element consists of one plated wire, which is a memory device for one 256-bit word, and one response store, whose function is to signal the matching of the word stored in

the plated wire. The limit on the number of processor modules largely depends on the hardware's physical size (a single plated-wire module occupies about 0.5 ft³) and the processor's speed requirements.

The STARAN Processor

One of the two well-known large-scale associative processors developed to date is Goodyear Aerospace Corporation's STARAN [32-35]. The basic structure of STARAN Model B is shown in Figure 9. It consists of a control system and a number (up to 32) of associative array modules. Each associative array module contains a 256-word \times 256-bit multidimensional access memory, 256 simple processing ele-

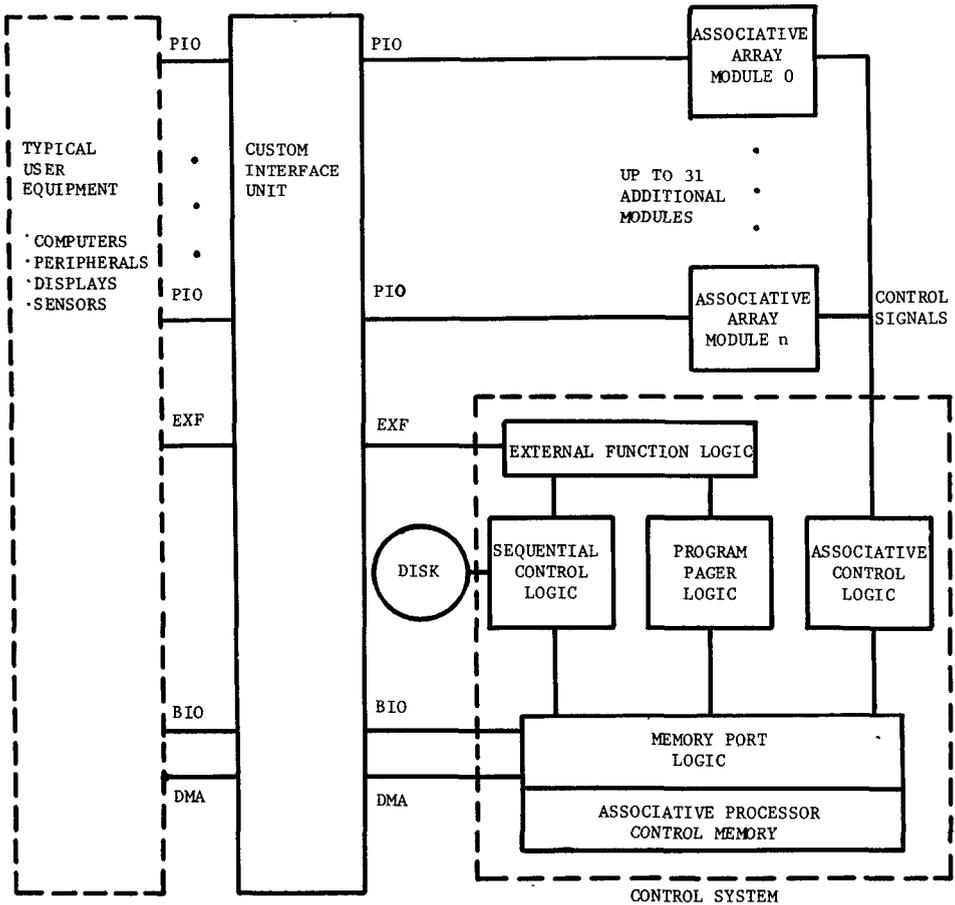


FIGURE 9. STARAN, a bit-serial associative processor.

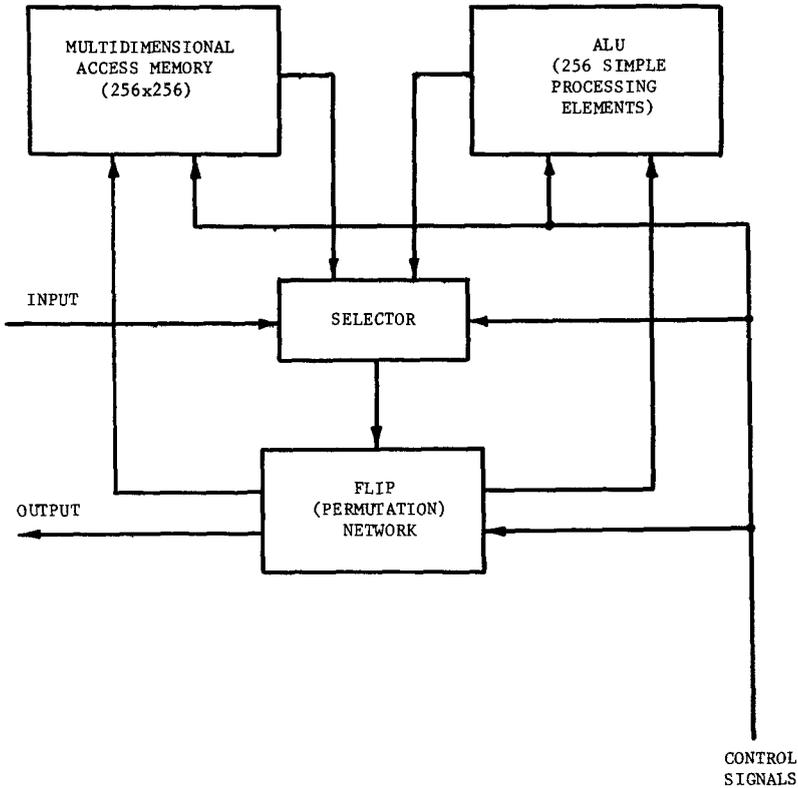


FIGURE 10. Associative array module of the STARAN.

ments, a permutation network (or so-called flip network), and a selector, as shown in Figure 10. There is a simple processing element for each of the 256 words of the memory, and each simple processing element operates serially by bit on the data in the memory word. This operational concept is shown in Figure 11. Using the permutation network, the data stored in the multidimensional access memory can be accessed through the input/output channel in the bit-slice direction, the word direction, or a combination of these. The permutation network is also used for shifting and rearranging of data in an associative array module so that parallel search, arithmetic or logical operations can be performed between words of the multidimensional access memory. By proper design of the permutation network, the multidimensional access memory can be implemented using random-access memory chips [34, 82].

against an input data item. All the words in the memories of all the modules that satisfy the search criterion are identified by a single instruction. The simple processing elements simultaneously execute operations

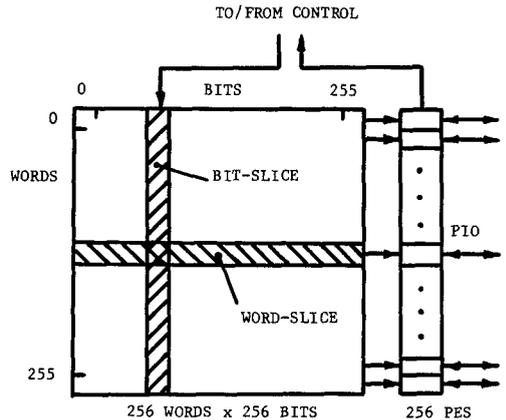


FIGURE 11. Operational concept of a STARAN associative array module.

To locate a particular data item, STARAN initiates a search by calling for a match

as specified by the associative control logic. Therefore, in one instruction execution, the data in all selected memories of all the modules are processed simultaneously by the simple processing element at each word.

The interface unit shown in Figure 9 involves interface with sensors, conventional computers, signal processors, interactive displays, and mass-storage devices. A variety of I/O options are implemented in the custom interface unit, including the direct memory access (DMA), buffered I/O (BIO) channels, external function (EXF) channels and parallel I/O (PIO). Each associative array module can have up to 256 inputs and 256 outputs into the custom interface unit. They can be used to increase speed of inter-array data communication, to allow STARAN to communicate with a high-bandwidth I/O device, and to allow any device to communicate directly with the associative array modules.

In many applications such as matrix computation, air-traffic control, sensor signal processing and data management systems, a hybrid system composed of an associative processor and a conventional sequential processor can increase the throughput rate, simplify the software complexity, and reduce the hardware cost. As mentioned before, STARAN has high-speed input/output capabilities and the ability to interface easily with conventional computers. In such a hybrid system, each associative array module performs the tasks best suited to its capabilities. STARAN handles the parallel processing tasks, and the conventional computer handles the tasks that must be processed in a single sequential data stream.

STARAN has been installed and operational at several locations. In 1973, an operational associative processor facility, called RADCAP, was installed at Rome Air Development Center [83-86]. This facility consists of a STARAN and various peripheral devices, all interfaced with a Honeywell Information Systems 645 sequential computer which runs under the Multics time-sharing operating system. The objective of the RADCAP facility is to explore various applications of the system to real-time problems. This facility is being expanded to include the STARAN, a QM-1 micro-

programmable sequential computer and a set of reconfigurable microprocessors for efficient emulation, application programming, and performance measurement of a wide variety of computer architectures [87]. In 1974 a STARAN was installed by the Defense Mapping Agency (DMA) and the U.S. Army Engineer Topographic Laboratories (USAETL) at the DMA/ETL facility in Fort Belvoir, Virginia [88]. The custom interface unit between STARAN and the host CDC 6400 computer consists of two parts. The first part is a command-channel interface unit that is capable of transferring data as well as command information. The second part is a data-channel interface unit that provides a path with extremely high bandwidth between the STARAN associative array modules and the CDC's extended core storage memory. Applications have been investigated to include automated cartography, digital image processing, stereophotogrammetry, and storage and retrieval. In 1975 a STARAN was installed at the NASA Johnson Space Center in Houston, Texas. It is used as a special-purpose processor in the Large Area Crop Inventory Experiment (LACIE). Goodyear Aerospace Corporation has recently introduced a new model, STARAN Model E. The organization of Model E is quite similar to that of Model B. However, the size of the multidimensional access memory has greatly increased. The current design has a size of 9216×256 per module. There are also improvements in processing speeds and I/O scheme.

Other Bit-Serial Associative Processors

Besides STARAN, several other important bit-serial associative processors have been developed. Among these are the OMEN computers [89] developed by Sanders Associates, the hybrid associative processor using an MOS shift-register bulk memory [90] developed by Hughes Aircraft Co., the Raytheon Associative/Array Processor (RAP) [57], the Associative Linear Array Processor (ALAP) [60], and the Extended Content Addressed Memory (ECAM) [59]. We briefly discuss the first three here, and consider ALAP and ECAM separately.

In the OMEN computer [89], a conventional serial processor such as the DEC PDP-11, and a bit-serial associative processor both address an orthogonal memory, which has a capacity of 64 words \times 16-bits. The associative processor contains 64 identical processors which form the vertical arithmetic unit that has bit-slice access to the orthogonal memory. These 64 processors perform the same operations at the same time under the control of masks.

The hybrid associative processor [90] developed by Hughes Aircraft Company contains 10-bit-serial associative memories and an MOS shift-register bulk memory. The bulk memory consists of a set of MOS shift registers, each having at least 16,000 bits. The purpose of this configuration is to achieve efficient operation of an associative memory when the data base is stored in a large inexpensive mass-storage device.

The Raytheon Associative/Array Processor (RAP) [57] contains a processing element array as well as a direct-array-access channel which facilitates bulk data transfer to and from the processing element array. The function of the processing element is to perform search, arithmetic, and logic operations on data stored in its own private memory. Each processing element can be thought of as a bit-serial microprocessor with associative capability.

Associative Linear Array Processor (ALAP)

ALAP [60] can be considered as a distributed-logic bit-serial associative processor. Its basic configuration is shown in Figure 12. The word cells form a line of processing-plus-memory elements. For demonstration purposes, an ALAP with 13-word cells has been implemented on an LSI wafer. A group of bit-serial buses that are common to all words is used for most data transfer and control communication. The extensive use of common buses is made practical by the multiuse chaining channel, which is the only bus not common to all words. This provides bit-serial communication between adjacent elements in the ALAP linear array. The common data and control block interprets the program to be executed in the ALAP memory array. The program is stored in a random-access program memory. The channels of data communication to the ALAP words are shown in Figure 13. In addition to the chaining channel there are three common data buses, namely the common input, the auxiliary input, and the common output, each of which has a connection to every word in ALAP memory. The common input channel may be used to supply common arguments to all words for matching or for arithmetic operations. If words are first cleared (set to 0), the

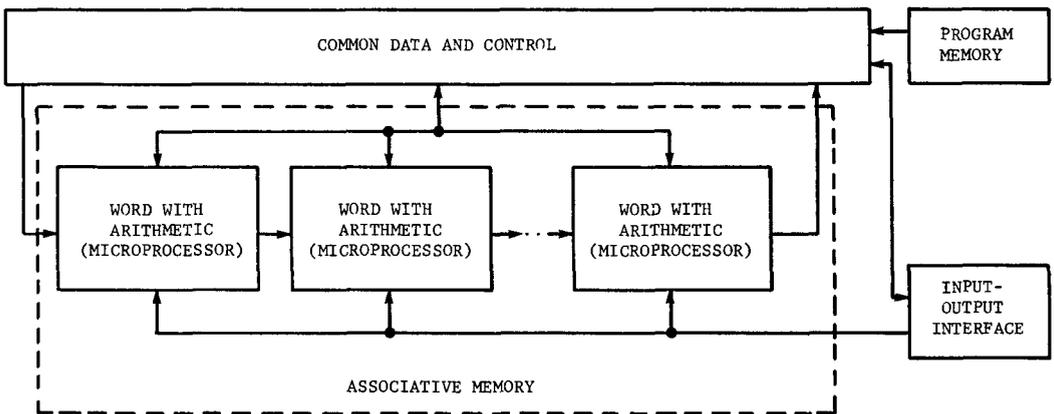


FIGURE 12. Basic Associative Linear Array Processor (ALAP) configuration.

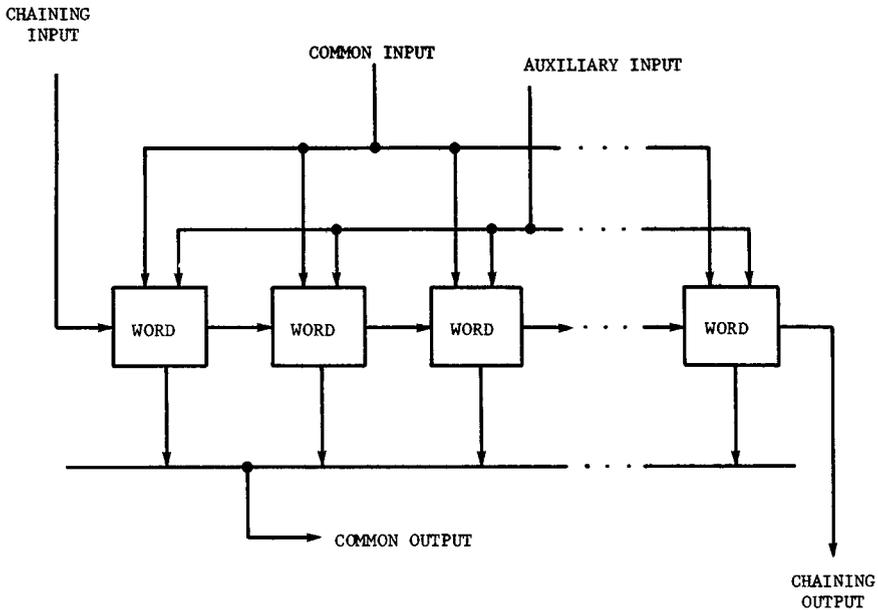


FIGURE 13. ALAP data communication channels.

common input channel can also be used to input new data. The auxiliary input channel, which is more often used for data input, is controlled by a common control line. The common output bus is arranged to supply the logical OR of all words trying to output, and its usual output is the contents of the word shift register.

In order to control the ALAP efficiently, three different overall global modes are time-shared. These are the flag-shift, word-cycle, and fault-isolation modes. Flag shift is the main setup mode for setting and transferring data among the flags in each word. This mode also includes logic operations that can be used to combine flag data in each word. Flag-shift operations are single-clock-time operations; a sequence of flag-shift operations can be executed rapidly. While flag-shift operations are in progress, none of the main data storage shift registers in words can be shifted. The word-cycle mode is the main processing mode for ALAP. All of these operations are word-cycle operations on the data fields stored in the shift registers, and controlled by global commands modified by local flag

data. Flexibility and the ability to do many different types of operation simultaneously are emphasized. In the fault-isolation mode, fault-isolation commands are executed. During fault isolation, shift-register and flag data are preserved.

ECAM

As mentioned before, Honeywell [59] has recently developed the Extended Content Addressed Memory (ECAM) based on their superchip technique for producing large associative memory arrays up to the billion-bit range for high-speed data management systems. The block diagram of the ECAM is shown in Figure 14 which is divided into two portions: the content-addressed memory (CAM) array, and the control unit.

The main unit in the control unit is the master control processor, which is a standard minicomputer. Its memory bus provides the basic structure of the control unit. The ECAM-host interface is designed to connect to the host computer as a standard high-speed peripheral such as a disk. It is controlled via the minicomputer's programmed I/O facility and transfers blocks

of information between the host computer and the master memory in a transparent fashion. Several host computers may be connected to a single ECAM by replicating the ECAM-host interface.

In addition to the master control processor, there is a slave control unit in which the interpreter and the iteration control are the two major subunits. The interpreter is a high-speed microprogrammed unit designed specifically for interpretively executing block-structured query language sequences from the master memory used to specify

ECAM operation sequences. The control store of the interpreter is writable, allowing easy changes to the query language. The output of the interpreter is a stream of array primitives which are passed to the iteration control unit via dedicated buffers. Iteration control is a hardwired subunit which generates control signal sequences to affect the array operations.

The CAM array consists of up to 250,000 words, each with 4096 bits (16 shift registers of 256 bits) randomly accessible to 256 bit registers. The word structure of the ECAM

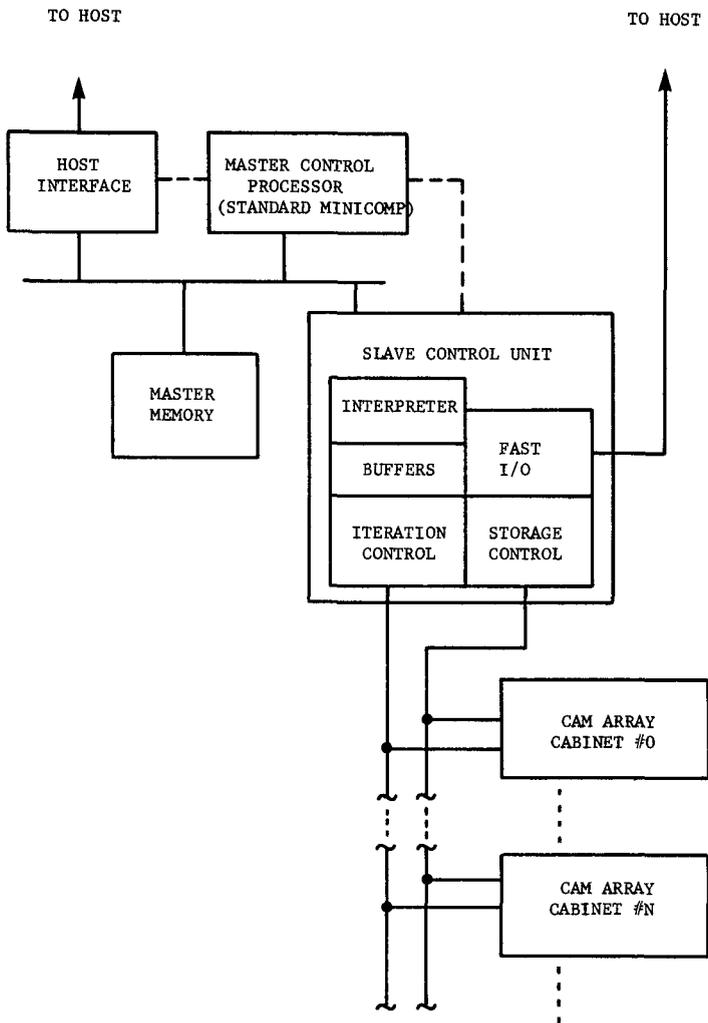


FIGURE 14. Extended Content Addressed Memory (ECAM) system block diagram.

is shown in Figure 15. The two major elements of the word logic are the match memory and the arithmetic-logic block. Word logic operations such as searches, arithmetic, etc. are performed by selecting one of 16 match bits from the memory and repeatedly executing the same sequence of combinational operations on each bit of a field within the memory word. The ECAM is provided with a high-speed I/O part which allows 10 words to be logically selected onto the I/O lines and to participate simultaneously during a single input or output operation.

Other Developments

Byte-Serial Word-Parallel Associative Processors

One of the related developments is the concept of a byte-serial associative processor, which is conceived to be between bit-serial and fully parallel associative processors. For reasons of efficiency at a reasonable cost, a byte-serial word-parallel associative

processor, called the Associative Processor Computer System (APCS), was proposed by Linde, Gates, and Peng [91] at System Development Corporation. APCS contains two associative processing units and a parallel input/output channel. The word logic consists of byte-operation logic rather than of bit-operation logic as in bit-serial associative processors.

Data Manipulators

For many applications such as air traffic control, matrix computation, and weather prediction, where the relations among data items are important, preparation of appropriate operands can greatly improve the efficiency of using associative processors (and array processors as well) in solving these problems, especially for bit-serial associative processors. In STARAN-type associative processors, the task of preparing appropriate operands is done in the permutation network. As mentioned before, Feng [36-40] proposed a generalized permutation network, called the data manipulator, for

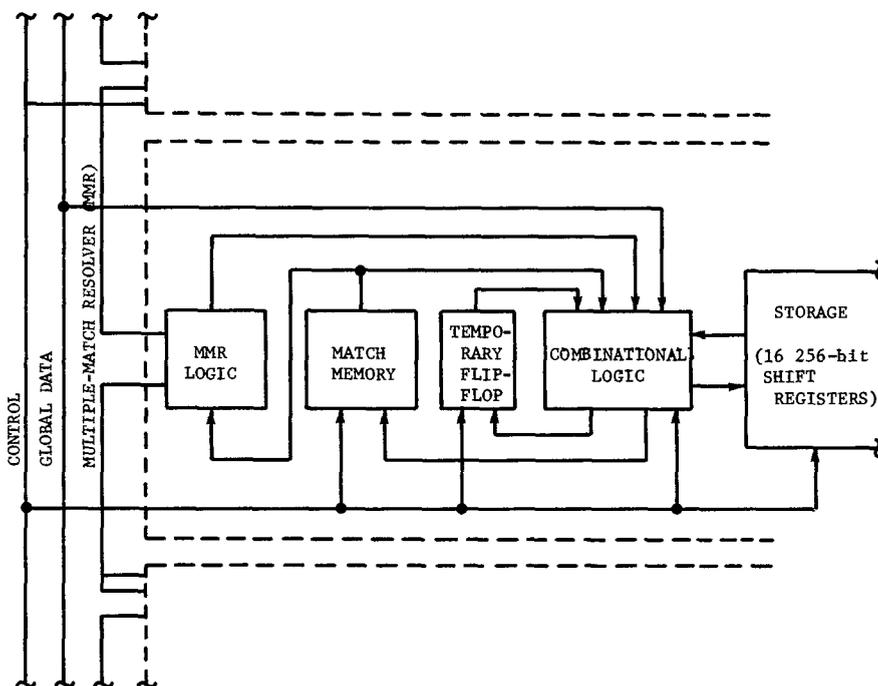


FIGURE 15. Word structure in CAM array of the ECAM.

this purpose. Because of its significance in improving the efficiency of bit-serial associative processors, we would like to discuss data manipulators in more detail.

In a conventional sequential computer, a machine instruction contains three operations: 1) instruction and operand fetching, 2) instruction decoding and address generation, and 3) execution. In an associative processor, because the operands are fetched by content, they usually require some manipulation before execution. Therefore, each instruction in an associative processor contains four operations: 1) instruction fetching, 2) instruction decoding, 3) operand fetching by content and data manipulation, and 4) execution. Examples of data manipulation are:

<i>Permuting</i>	<i>Replicating</i>
shift	multiply
flip	duplicate
shuffle	<i>Spacing</i>
transpose	spread
merge	compress
mix	transfer
bit-reverse sort	<i>Masking</i>
sort	<i>Complementing</i>

Conceptually, the data manipulator operates in a parallel processor as shown in Figure 16; the block diagram of the data manipulator proposed by Feng is shown in Figure 17 [38]. Because the number and the types of data manipulation required vary

with applications, the complexity of the data manipulator design also varies. All module broadcasting registers (MBR) in Figure 17 form a module of N words for $N \leq M$, or a number of modules for $N > M$, where M is the number of processing elements. The MBRs are in the control unit and the content of each MBR module may be manipulated by the module data manipulator.

As mentioned before, the data manipulator can be implemented as a line manipulator in bit-serial fashion or as a page manipulator in fully parallel fashion. Currently, Gaertner Research, Inc. is implementing a line manipulator [37] to operate in conjunction with the STARAN at Rome Air Development Center. For details of the implementation the reader is referred to [92].

It should be noted that the data manipulator is one of the many types of interconnection networks that are used in SIMD (Single Instruction Stream Multiple Data Stream) machines for providing communications among processing elements as well as memory modules. In addition to the data manipulator, various interconnection networks have been investigated by many researchers, for example Stone [93], Lawrie [94], Lang [95], and Siegel [96, 97]. Interested readers are referred to references [93-97].

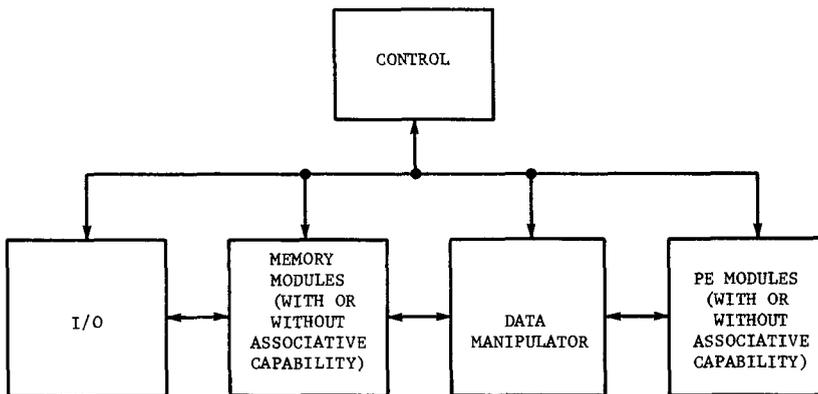


FIGURE 16. Operational concept of a data manipulator.

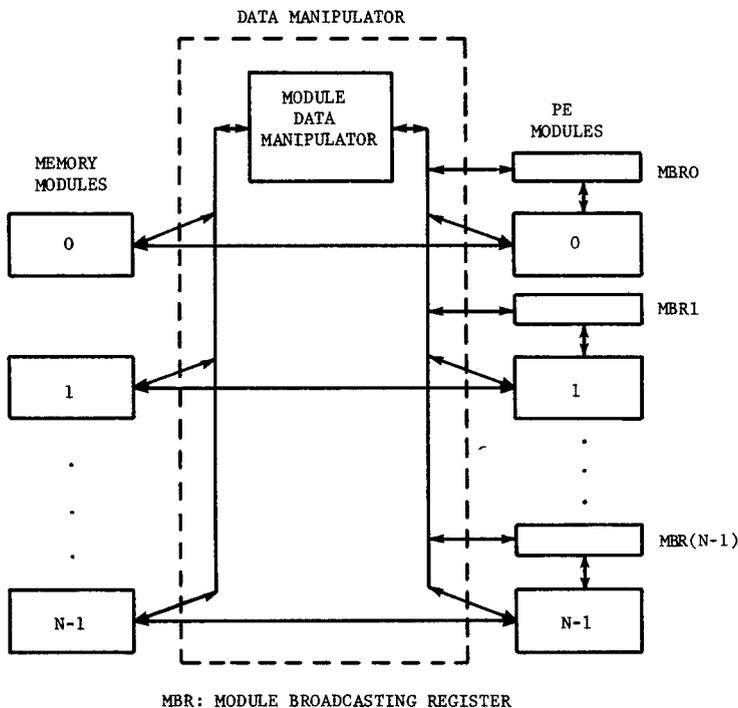


FIGURE 17. Block diagram of a data manipulator.

4. WORD-SERIAL ASSOCIATIVE PROCESSORS

As mentioned before, a word-serial associative processor essentially represents a hardware implementation of a simple program loop for search. The important factor contributing to the relative efficiency of this approach as compared to programmed search in a standard sequential processor is that the instruction decoding time is reduced, since only a single instruction in the word-serial associative processor is required to perform the search operation.

In 1962 Young [98] proposed to use circulating associative memories to allow many memory words to time-share a single set of content-addressing logic. In 1966, Crofut and Sottile [99] presented a word-serial associative processor based on a word-serial associative memory using n ultrasonic digital delay lines, where n is the number of bits of a word, operating at 100 MHz with 10 μ sec delay time. Each delay line stores one bit of the word, and all bits of the stored

word propagate down the delay lines synchronously. A stable oscillator (Stalo) was used to generate the synchronizing clock pulses for advancing the address counter. Individual words can be interrogated and updated when they appear at the output of delay lines. The rewrite control logic allows the delay-line system to select either recirculating information or new data inputs. The operational characteristics of such a memory resemble that of a drum or disk. Such a word-serial associative processor is shown in Figure 18. In 1969, Rux [100] presented a word-serial associative memory with 35 glass delay lines storing 2046 bits per line at 20.48 MHz, which was connected to a general-purpose medium-speed sequential computer called NEBULA [101, 102].

Because of the slow speed of word-serial associative memories, only experimental models of word-serial associative processors have been developed.

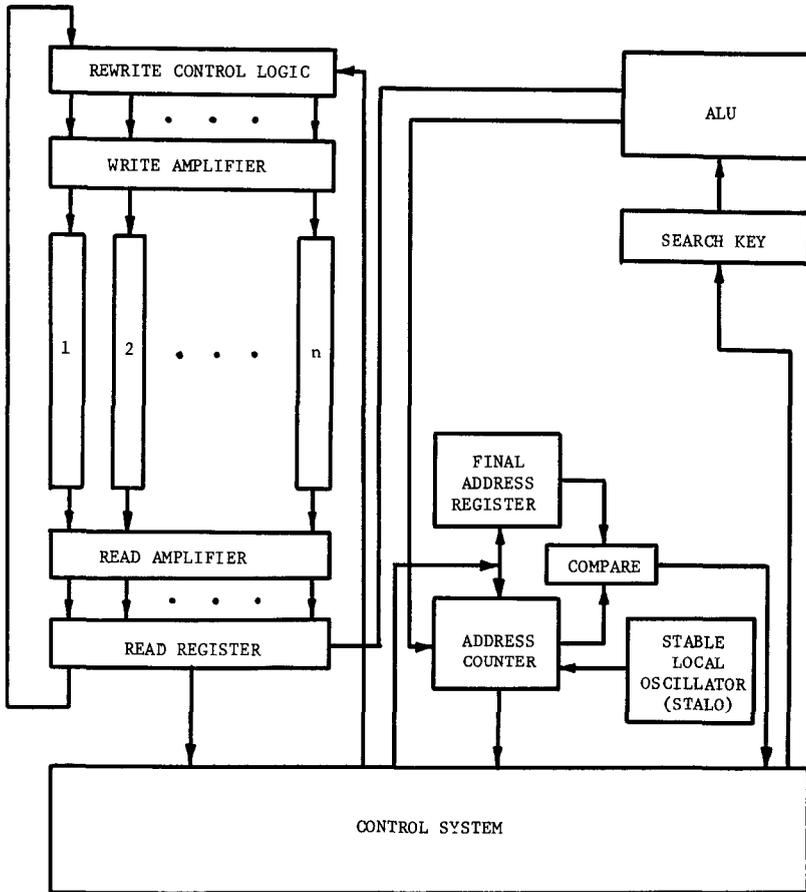


FIGURE 18. Word-serial associative processor.

5. BLOCK-ORIENTED ASSOCIATIVE PROCESSORS

The block-oriented associative processor [103–107] provides a compromise between the high cost of the bit-serial associative processor and the low speed of the word-serial associative processor. A block-oriented associative processor uses a mass rotating storage device such as a disk to provide a limited degree of associative capabilities. A number of block-oriented associative processors have been developed.

Slotnick [103] and Parker [104] presented the concept of logic-per-track devices which consist of a head-per-track disk memory having some logic associated with each track. Based on this concept and Lee's

distributed logic memory for information storage and retrieval applications, Parhami [105] presented a block-oriented associative processor, called RAPID (Rotating Associative Processor for Information Dissemination), which is shown in Figure 19. Since the data rates between head-per-track disks and distributed logic memory is high, the RAPID system is suitable for applications requiring a large storage capacity, which presently suffer from the high cost of random-access memories or from performance degradation due to the frequent transfers between primary and secondary memories.

Minsky [106] proposed associativity on rotating memories in the form of either drums or disks. He defined the term *partially associative memory* by specifying the

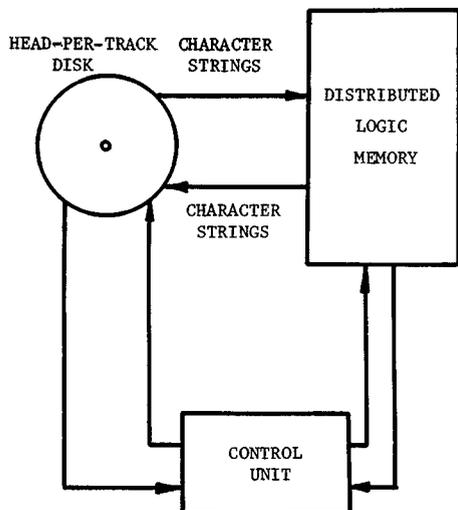


FIGURE 19. Block-oriented associative memory used in the RAPID.

primitive structure of information (name part and data part) to be stored on it as well as the operational characteristics (predicates and instructions). The activity of the memory is supervised by a special processor, called the controller. Instead of spending the time looking for a given address, he proposed to use the delay time in a search for content.

Another block-oriented associative processor has been proposed by Healy, Lipovski, and Doty [107]; it is based on storage and retrieval from a segmented sequential table data structure utilizing associative addressing.

SUMMARY

In this paper, we have reviewed the architecture of various associative processors and classified them into four major categories based on the organization of their associative memories. Among these associative processors, PEPE and STARAN are the two best-known large-scale associative processors that have been implemented and put into practical use. The performance of these two types of associative processors and others have been evaluated in a real-time environment by Lloyd and Merwin [108]. As we have described, several associative processors developed recently, such as

ALAP and ECAM, are based on using LSI technology for implementation and hence make large-scale associative processors in the billion-bit range economically feasible. From an architectural point of view, fully parallel and bit-serial associative processors are used for high-speed parallel data processing which cannot be carried out effectively in ordinary sequential computers. However, their implementation costs are higher. For the low-cost associative processing which is required in large information storage and retrieval systems, block-oriented associative processors offer a promising architecture.

ACKNOWLEDGMENT

The authors would like to thank Professors T. Y. Feng of Wayne State University, M. J. Gonzalez, Jr., of Northwestern University, and Dr. Richard E. Merwin of U. S. Army Ballistic Missile Defense Program Office for many helpful discussions. They also would like to thank Capt. A. R. Klayton of U. S. Air Force Rome Air Development Center, G. A. Anderson of Honeywell, and W. W. Gaertner of Gaertner Research Inc., for providing useful information on the associative processor systems recently developed at their organizations.

REFERENCES

- [1] HANLON, A. A. "Content-addressable and associative memory systems: a survey," *IEEE Trans. Computers* **EC-15** (August 1966), 509-521.
- [2] MINKER, J. "An overview of associative or content-addressable memory systems and a KWIC index to the literature," *Computing Reviews* **12**, 10 (Oct. 1971), 453-504.
- [3] PARHAMI, B. "Associative memories and processors: an overview and selected bibliography," *Proc. IEEE* **61** (June 1973), 722-730.
- [4] MINKER, J. "Association memories and processors: a description and appraisal," in *Encyclopedia of computer science and technology*, Vol. 2, Marcel Dekker, Inc., N.Y., 1975, pp. 283-324.
- [5] THURBER, K. J.; AND WALD, L. D. "Associative and parallel processors," *Computing Surveys* **7**, 4 (Dec. 1975), 215-255.
- [6] SEEBER, R. R. "Cryogenic associative memory," in *Proc. ACM National Conf.*, 1960, ACM, N.Y., 1960.
- [7] LEWIN, M. H.; BEELITZ, H. R.; AND RAJCHMAN, J. A. "Fixed associative memory using evaporated organic diode arrays," in *Proc. AFIPS 1963 Fall Jt. Computer Conf.*, Vol. 24, Spartan Books, Inc., Baltimore, Md., 1963, pp. 101-106.
- [8] CHU, Y. H. "A destructive-readout associative memory," *IEEE Trans. Computers* **EC-14** (August 1965), 600-605

- [9] WEINBERGER, A. "The hybrid associative memory concept," *Computer Design* (Jan. 1971), 77-85.
- [10] QUINONES, M. "An associative-capacitive ROM for reprogrammable logic applications," *Computer Design* (Jan. 1974), 98-101.
- [11] SLADE, A; AND MCMAHON, H. O. "A Cryotron catalog memory system," in *Proc. 1956 Eastern Jt. Computer Conf.*, American Inst. Electrical Engineers, N.Y., 1957, pp 115-120.
- [12] LEWIN, M H. "Retrieval of ordered lists from a content-addressed memory," *RCA Reviews* **23** (June 1962), 215-229.
- [13] NEWHOUSE, V. L; AND FRUIN, R. E. "A Cryogenic data addressed memory," in *Proc. AFIPS 1962 Spring Jt. Computer Conf.*, AFIPS Press, Montvale, N J, 1962, pp. 89-99.
- [14] FALKOFF, A. D. "Algorithms for parallel search memories," *J. ACM* **9**, 4 (Oct. 1962), 488-511.
- [15] JOSEPH, E. C., AND KAPLAN, A. "Target-track correlation with a search memory," in *Proc. IEEE National Conv on Military Electronics*, 1962, pp. 255-261
- [16] KAPLAN, A. "A search memory subsystem for a general-purpose computer," in *Proc. AFIPS 1963 Fall Jt. Computer Conf*, Vol 24, Spartan Books, Inc., Baltimore, Md, 1963, pp 193-200.
- [17] GALL, R. G. "A hardware-integrated GPC/search memory," in *Proc. AFIPS 1964 Fall Jt. Computer Conf*, Vol 26, Pt. I, Spartan Books, Inc, Baltimore, Md, 1964, pp. 159-173.
- [18] PICK, G G "A read-only multi-megabit parallel search associative memory," in *The 26th Annual Meeting of American Documentation*, 1963.
- [19] ESTRIN, A.; AND FULLER, R. "Algorithms for content-addressable memories," *Proc. IEEE Pacific Computer Conf.*, 1963, pp. 118-128.
- [20] LEE, C. Y., AND PAULL, M. C. "A content addressable distributed logic memory with applications to information retrieval," *Proc. IEEE*, **51** (June 1963), pp. 924-932
- [21] DERICKSON, R. B. "A proposed associative pushdown memory," *Computer Design* (March 1968), pp 60-66
- [22] NATARAJAN, N. K.; AND THOMAS, P A V. "A multiaccess associative memory," *IEEE Trans. Computers* **C-18** (May 1969), 424-428.
- [23] CHU, Y. H. *Computer organization and microprogramming*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1972.
- [24] FLYNN, M J "Some computer organizations and their effectiveness," *IEEE Trans. Computers* **C-21**, 9 (September 1972), 948-960.
- [25] BARNES, G H , BROWN, R. M.; KATO, M , KUCK, D. J., SLOTNICK, D. L.; AND STOKES, R. A. "The ILLIAC IV computer," *IEEE Trans. Computers* **C-47**, 8 (August 1968), 746-757.
- [26] CRANE, B A.; GILMARTIN, M. J., HUTTENHOFF, J. H.; RUX, P T., AND SHIVELY, R R "PEPE computer architecture," *IEEE COMPCON* (1972), 57-60.
- [27] WILSON, D E. "The PEPE support software system," *IEEE COMPCON* (1972), 61-64.
- [28] CORNELL, J A. "Parallel processing of ballistic missile defense radar data with PEPE," *IEEE COMPCON* (1972), 69-72.
- [29] EVENSEN, A J; AND TROY, J. L. "Introduction to the architecture of a 288-element PEPE," in *Proc. 1973 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1973, pp 162-169.
- [30] DINGELDINE, J R.; MARTIN, H R; AND PATTERSON, W. M "Operating system and support software for PEPE," in *Proc. 1973 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1973, pp. 170-178.
- [31] VICK, C. R., AND MERWIN, R. E. "An architecture description of a parallel processing element," in *Proc. 1973 Internatl. Workshop on Computer Architecture*.
- [32] RUDOLPH, J A "A production implementation of an associative array processor: STARAN," in *Proc. AFIPS 1972 Fall Jt. Computer Conf.*, Vol. 41, Pt. 1, AFIPS Press, Montvale, N.J., 1972, pp. 229-241.
- [33] BATCHER, K E "Flexible parallel processing and STARAN," *1972 WESCON Tech. Papers*, Session 1-Parallel Processing Systems, Sept 1972, pp. 1/5-1-1/5-3
- [34] BATCHER, K. E. "STARAN parallel processor system hardware," in *Proc. AFIPS 1974 National Computer Conf*, Vol 43, AFIPS Press, Montvale, N.J., 1974, pp. 405-410.
- [35] DAVIS, E W "STARAN parallel processor system software," in *Proc AFIPS 1974 National Computer Conf*, Vol 43, AFIPS Press, Montvale, N.J., 1974, pp 17-22.
- [36] FENG, T. "An overview of parallel processing systems," *1972 WESCON Tech. Papers*, Session 1-Parallel Processing Systems, Sept. 1972, 1-2
- [37] FENG, T. *The design of a versatile line manipulator*, Tech. Report RADC-TR-73-292, Sept. 1973.
- [38] FENG, T "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Computers* **C-23**, 3 (March 1974), 309-318.
- [39] FENG, T.; AND HSU, C. *Design and evaluation of an arithmetic unit*, Technical Report RADC-TR-73-86, March 1975.
- [40] FENG, T. "A configurable multiple-microprocessor organization," *EUROMICRO*, North-Holland Publ. Co., Amsterdam, The Netherlands 1975, pp. 207-217.
- [41] FULLER, R. H. "Content-addressable memory systems," Ph.D. Thesis, University of California, Los Angeles, June 1960, 549 pp.
- [42] NISSIM, S. "Organizing the Nanophile computers," *Electronic Design*, **11** (March 1963), 44-53
- [43] McDERMID, W. L; AND PETERSON, H. E "A magnetic associative memory system," *IBM J. R. and D* **5** (Jan. 1961), 59-62.

- [44] KISED, J. R.; PETERSON, H. E.; SEELBACK, W. C.; AND TEIG, M. "A magnetic associative memory," *IBM J. R. and D.* **5** (April 1961), 106-121.
- [45] HUNT, R. T.; SNIDER, D. L.; SUPRISE, J.; AND BOYD, H. N. *Study of elastic switching for associative memory systems*, U.S. Gov. Research Reports, Vol. 39, p. 188(A), AD 432041, May, 1964.
- [46] YOUNKER, E. L.; HECKLER, C. H.; MASHER, D. P.; AND YARBOURGH, J. M. "Design of an experimental multiple instantaneous reference file," in *Proc. AFIPS 1964 Spring Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1964, pp. 515-528.
- [47] APICELLA, A.; AND FRANKS, J. "BILOC—a high speed NDRO one core per bit associative element," in *Proc. 1965 IEEE Intermag. Conf.*, pp. 14.5-1 to 14.5-6.
- [48] GOODYEAR AEROSPACE CORP. *Handbook of operating and maintenance instructions for the associative memory*. GAC Tech. Report, GER-13738, Akron, Ohio March, 1968.
- [49] EWING, R. G.; AND DAVIES, P. M. "An associative processor" in *Proc. AFIPS 1964 Fall Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1964, pp. 147-158.
- [50] LEE, E. S. "Associative techniques with complementing flip-flops," in *Proc. AFIPS 1963 Spring Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1963, pp. 381-394.
- [51] LUSSIER, R. R.; AND SCHNEIDER, R. P. "All magnetic content addressed memory," *Electronic Indust.* **22** (March 1963), 92-98.
- [52] MCATEER, J. E.; CAPOBIANO, J. A., AND KOPPEL, R. L. "Associative memory system implementation and characteristics," in *Proc. AFIPS 1964 Fall Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1964, pp. 81-92.
- [53] WOLFF, M. F. "What's new in computer memories," *Electronics* (Nov. 1963), 35-39.
- [54] RAFFEL, J. I.; AND CROWTHER, T. S. "A proposal for an associative memory using magnetic films," *IEEE Trans. Computers* **EC-13** (Oct. 1964), p. 611, (short notes).
- [55] PICK, G. G. "A semipermanent memory utilizing correlation addressing," in *Proc. AFIPS 1964 Fall Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1964, pp. 107-121.
- [56] TUTTLE, G. T. "How to quiz a whole memory at once," *Electronics* (Nov. 1963), 43-46.
- [57] COURANZ, G. R.; GERHARDT, M. S.; AND YOUNG, C. J. "Programmable radar signal processing using the RAP," in *Proc. Sagamore Computer Conference on Parallel Processing*, Springer-Verlag, N.Y., 1974, pp. 37-52.
- [58] BEHNKE, E. A.; AND ROSENBERGER, G. B. *Cryogenic associative processor*, IBM Final Report, Sept. 1963.
- [59] ANDERSON, G. A.; AND KAIN, R. Y. "A content-addressed memory design for data base applications," in *Proc. 1976 Internatl. Conf. on Parallel Processing*, IEEE, New York, 1976, pp. 191-195.
- [60] FINNILA, C. A. "The associative linear array processor," to appear in *IEEE Trans. Computers*, February 1977.
- [61] SLADE, A. E.; AND SMALLMAN, C. R. "Thin film Cryotron catalog memory," *Automatic Control* **13** (August 1960), 48-50.
- [62] ROSIN, R. F. "An organization of an associative Cryogenic computer," in *Proc. AFIPS 1962 Spring Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1962, pp. 203-212.
- [63] DAVIES, R. M. "A superconductive associative memory," in *Proc. AFIPS 1962 Spring Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1962, pp. 79-88.
- [64] AHRONS, R. W. "Superconductive associative memories," *RCA Reviews* **24** (Sept. 1962), 325-354.
- [65] BARNARD, J. D.; BEHNKE, F. A.; LINDQUIST, A. B.; AND SEEBER, R. R. "Structure of a Cryogenic associative processor," *Proc. IEEE* **52** (Oct. 1964), 1182-1190.
- [66] YAU, S. S.; AND YANG, C. C. "A Cryogenic associative memory system for information retrieval," in *Proc. National Electronic Conf.*, 1966, Vol. 22, pp. 764-769.
- [67] YANG, C. C.; AND YAU, S. S. "A Cutpoint cellular associative memory," *IEEE Trans. Computers* **EC-15** (August 1966), 522-528.
- [68] LEE, C. Y. "Intercommunicating cells, basis for a distributed logic computer," in *Proc. AFIPS 1962 Fall Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1962, pp. 130-136.
- [69] EDWARDS, R. P. "Content-addressable distributed-logic memories," *Proc. IEEE* **52** (Jan. 1964), 83-84.
- [70] SPIEGELTHAL, E. S. "A Content addressable distributed logic memory with applications to information retrieval," *Proc. IEEE* **52** (Jan. 1964), 74.
- [71] LEE, C. Y. "Content-addressable and distributed logic memories," *Applied automata theory*, J. T. Tou, (Ed.), Academic Press, N.Y., 1968.
- [72] GAINS, R. S.; AND LEE, C. Y. "An improved cell memory," *IEEE Trans. Computers* (Feb. 1965), 72-75.
- [73] CRANE, B. A.; AND GRITHENS, J. A. "Bulk processing in distributed logic memory," *IEEE Trans. Computers* **EC-14** (April 1965), 186-196.
- [74] LIPOVSKI, G. J. "The architecture of a large associative processor," in *Proc. AFIPS 1970 Spring Jt. Computer Conf.*, AFIPS Press, Montvale, N.J., 1970, pp. 385-396.
- [75] CRANE, B. A.; AND LAANE, R. R. "A Cryoelectronic distributed logic memory," in *Proc. AFIPS 1967 Spring Jt. Computer Conf.*, Spartan Books Inc., Baltimore, Md., 1967, pp. 517-524.
- [76] EDELBERG, M.; AND SCHISSLER, L. R. "Intelligent memory," in *Proc. AFIPS 1976 National Computer Conf.*, AFIPS Press, Montvale, N.J., 1976, pp. 393-400.
- [77] SHOOMAN, W. "Parallel computing with vertical data," in *Proc. 1960 Eastern Jt. Computer Conf.*, Eastern Jt. Computer Conf., N.Y., 1960, pp. 111-115.
- [78] HARDING, P. A.; AND ROLUND, M. W. "A

- 2- $\frac{1}{2}$ D core search memory," in *Proc. AFIPS 1968 Fall Jt. Computer Conf.*, Thompson Book Co., Washington, D.C., 1968, pp. 1213-1218.
- [79] STONE, H. S. "Associative processing for general purpose computers through the use of modified memories," in *Proc. AFIPS 1968 Fall Jt. Computer Conf.*, Thompson Book Co., Washington, D.C., pp. 949-955.
- [80] FULMER, L. C.; AND MEILANDER, W. C. "A modular plated wire associative processor," in *Proc. 1970 IEEE Internatl. Computer Group Conf.* IEEE, N.Y., 1970, pp. 325-335.
- [81] RUDOLPH, J. A.; FULMER, L. C.; AND MEILANDER, W. C. "The coming of age of the associative processor," *Electronics* (Feb. 1971), 91-96.
- [82] BATCHER, K. E. "The multi-dimensional access memory in STARAN," in *Proc. 1975 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1975, pp. 167.
- [83] FELDMAN, J. D.; AND FULMER, L. C. "RADCAP—an operational parallel processing facility," in *Proc. AFIPS 1974 National Computer Conf.*, AFIPS Press, Montvale, N.J., 1974, pp. 7-15.
- [84] FELDMAN, J. D.; AND REIMAN, O. A. "RADCAP: an operational parallel processing facility," in *Proc. 1973 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1973, pp. 140-146.
- [85] BATCHER, K. E. "STARAN/RADCAP Hardware Architecture," in *Proc. 1973 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1973, pp. 147-152.
- [86] DAVIS, E. W. "STARAN/RADCAP system software," in *Proc. 1973 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1973, pp. 153-159.
- [87] KLAYTON, A. R. "Concept for a computer architecture research facility," in *Proc. 1976 Internatl. Conf. on Parallel Processing*, IEEE, New York, 1976, 189-190.
- [88] GAMBINO, L. A.; AND BOULIS, R. L. "STARAN complex, defense mapping agency, U. S. Army Engineer Topographic laboratories," in *Proc. 1975 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1975, pp. 132-141.
- [89] HIGBIE, L. C. "The OMEN computers: associative array processors," *IEEE COMPCON* (1972), 287-290.
- [90] LOVE, H. H., JR., "An efficient associative processor using bulk storage," in *Proc. 1973 Sagamore Computer Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1973, pp. 103-112.
- [91] LINDE, R. R.; GATES, R., AND PENG, T. F. "Associative processor applications to real-time data management," *Proc. AFIPS 1973 National Computer Conf.*, AFIPS Press, Montvale, N.J., 1973, pp. 187-195.
- [92] GAERTAER, W. W.; PATEL, M. P.; AND RETLER, C. T. "Construction of a versatile data manipulator for parallel/associative processors," in *Proc. 1976 Internatl. Conf. on Parallel Processing*, IEEE, New York, 1976, p. 72.
- [93] STONE, H. S. "Parallel processing with the perfect shuffle," *IEEE Trans. Computers*, (Feb. 1971), 153-161.
- [94] LAWRIE, D. "Access and alignment of data in an array processor," *IEEE Trans. Computers*, (Dec. 1975), 1145-1155.
- [95] LANG, T. "Interconnections between processors and memory modules using the shuffle-exchange network," *IEEE Trans. Computers*, (May, 1976), 496-503.
- [96] SIEGEL, H. J. "Analysis techniques for SIMD machine interconnection networks and the effects of processor address masks," in *Proc. 1975 Sagamore Conf. on Parallel Processing*, Springer-Verlag, N.Y., 1975, pp. 106-109.
- [97] SIEGEL, H. J. "Single instruction stream—multiple data stream machine interconnection network design," in *Proc. 1976 Internatl. Conf. on Parallel Processing*, IEEE, New York, 1976, 273-282.
- [98] YOUNG, F. H. *Circulating associative memories*, Dept. of Mathematics Report, Oregon State Univ., 1962.
- [99] CROFUT, W. A.; AND SOTTILE, M. R. "Design techniques of a delay-line content-addressed memory," *IEEE Trans. Computers* (August 1966), 529-534.
- [100] RUX, P. T. "A glass delay line content-addressable memory system," *IEEE Trans. Computers* (June 1969), 512-520.
- [101] BOLES, J. A.; RUX, R. T.; AND WEINGARTEN, W., JR. "NEBULA: a digital computer using a 20 mc glass delay line memory," *Comm. ACM*, 9, 7 (July 1966), 503-508.
- [102] WEINGARTEN, W. *On an associative memory for the NEBULA computer*, Dept. of Mathematics Report, Oregon State Univ., 1964.
- [103] SLOTNICK, D. L. "Logic per track devices," *Advances in computers*, Vol. 10, Academic Press, N.Y., 1970, pp. 291-296.
- [104] PARKER, J. L. "A logic per track retrieval system," in *Proc. IFIP 1971 Congress*, Vol. 1, North-Holland Publ. Co., Amsterdam, The Netherlands, 1971, pp. 711-716.
- [105] PARHAMI, B. "A highly parallel computing system for information retrieval," in *Proc. AFIPS 1972 Fall J. Computer Conf.*, AFIPS Press, Montvale, N.J., 1972, pp. 681-690.
- [106] MINSKY, N. "Rotating storage devices as partially associative memories," in *Proc. AFIPS 1972 Fall Jt. Computer Conf.*, AFIPS Press, Montvale, N.J., 1972, pp. 587-595.
- [107] HEALY, L. D.; LIPOVSKI, G. J.; AND DOTY, K. L. "The architecture of a context addressed segment-sequential storage," in *Proc. AFIPS 1972 Fall Jt. Computer Conf.*, AFIPS Press, Montvale, N.J., 1972, pp. 691-701.
- [108] LLOYD, G. R.; AND MERWIN, R. E. "Evaluation of performance of parallel processors in a real-time environment," in *Proc. AFIPS 1973 National Computer Conf.*, AFIPS Press, Montvale, N.J., 1973, pp. 101-108.