

# The STARAN architecture and its application to image processing and pattern recognition algorithms

by J. L. POTTER

Goodyear Aerospace Corporation  
Akron, Ohio

## INTRODUCTION

The STARAN E is a general purpose parallel computer. However, certain aspects of the STARAN E's architecture, specifically, a single instruction stream-multiple data stream organization, high speed I/O, a flip (permutation) network and a conditional operation capability on each parallel processing element, are particularly pertinent to the areas of image processing and pattern recognition.

Some of the characteristics of image processing and pattern recognition which provide a good fit to the STARAN's architecture are repetitiveness, spatial dependencies, complex parallel decision making and high I/O to computation ratios. These characteristics and their corresponding STARAN E architectural accommodations are described in detail. The other aspects of STARAN's architecture are described in only enough detail to provide a basis for discussion.\*

## BACKGROUND

The STARAN E consists of an associative processor (AP) control module and a number (1 to 32) of associative arrays. The AP module consists of the AP control circuitry itself and bulk core. Figure 1 shows the basic STARAN architecture.

The arrays can be thought of as consisting of high speed memory, low speed memory and a bank of processing elements. Each array is organized into 256 "words." Associated with each "word" is a processing element (PE). Each word is 9K bits long with 1024 bits of high speed memory and 8192 bits of lower speed memory. Figure 2 illustrates the conceptual array organization and a general purpose layout for a 512x512 8 bit/pixel image.

Arithmetic operations are performed in parallel on every (enabled) word of memory, one bit at a time. That is, the least significant bit of every word in a field (a bit column vector) is added (multiplied, etc.) to the corresponding least significant bit of every word in a second field and stored in

the least significant bits of the sum (product, etc.) field. The carry bits are saved and added into the second bit slices of the arguments. This process is repeated until the entire fields have been processed.

## ARCHITECTURE—ALGORITHM PARINGS

Certain characteristics of image processing and pattern recognition mesh perfectly with some of the architectural features of the STARAN E. In the following paragraphs,

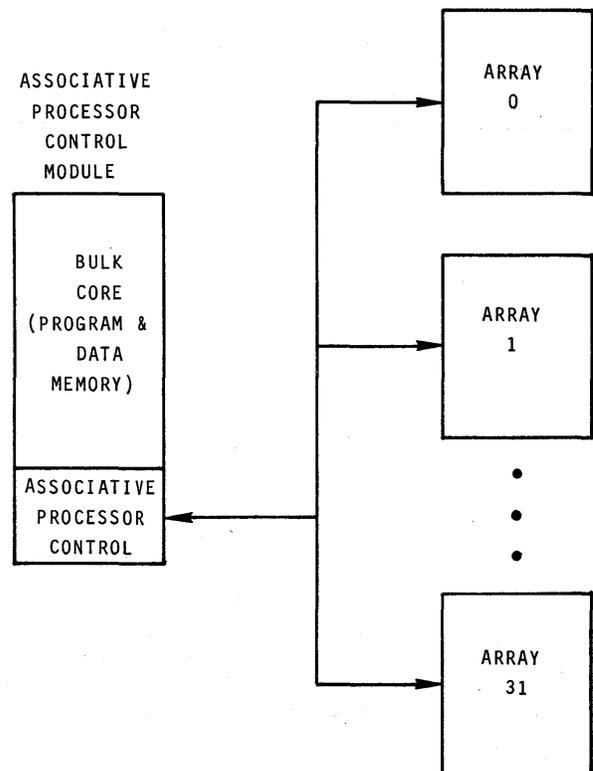


Figure 1—Basic STARAN architecture

\* For a detailed description of the STARAN line of computers see References 1, 2 and 3.

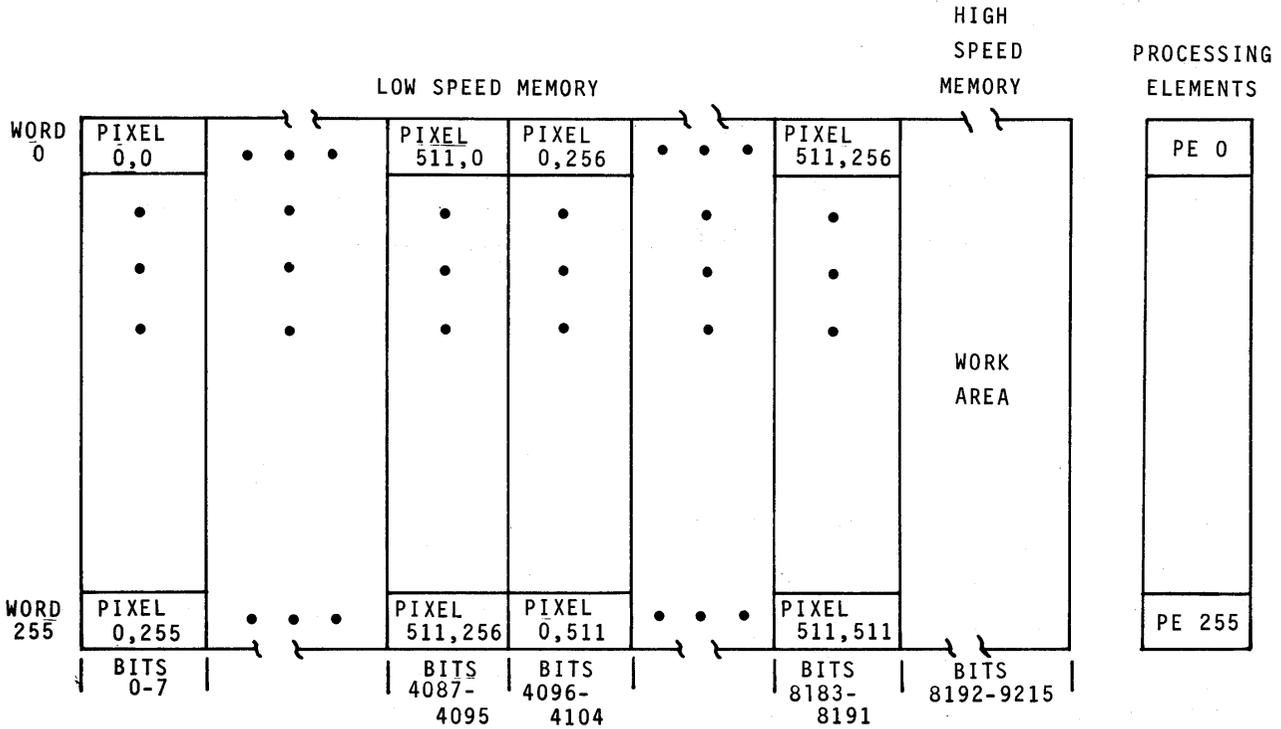


Figure 2—General purpose array organization

some characteristics of image processing and pattern recognition algorithms will be identified and described. Then the corresponding architectural feature of the STARAN E will be discussed and shown how it facilitates implementation of the algorithms.

**REPETITIVENESS**

Images consist of large volumes of data. A standard TV monitor size image contains over a quarter of a million pixels (picture elements). The images obtained from satellites may contain many millions of pixels. Almost all image processing and pattern recognition algorithms consist of performing the same sequence of operations for every pixel in an image. This aspect of image processing fits perfectly with the single instruction stream-multiple data stream organization of the STARAN.

The associative processor (AP) control portion of the STARAN computer provides a single sequential instruction stream to the associative arrays. Each associative array contains 256 processing elements (PE's) and a fully equipped STARAN E may have up to 32 arrays. Thus the single instruction stream can control from 256 to 8192 PE's resulting in a processing capability of from 11 to 356 million 32-bit adds per second (MIPS). Figure 1 illustrates the single instruction stream-multiple data stream organization of STARAN.

"Inherently serial" algorithms such as classical Maximum

Likelihood classification are easily implemented in parallel with the above organization. This is because the algorithm is still executed in serial for every pixel, but from 256 to 8192 pixels can be handled with one pass of the algorithm. With the large number of pixels that need to be processed in a typical image, parallel application of the algorithm is the only practical answer.

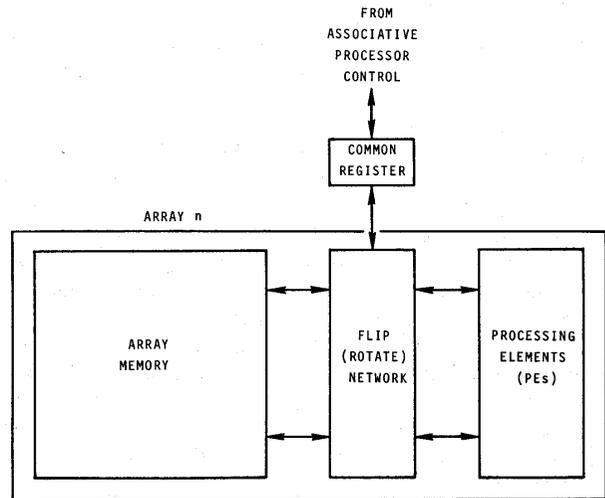


Figure 3—Flip network

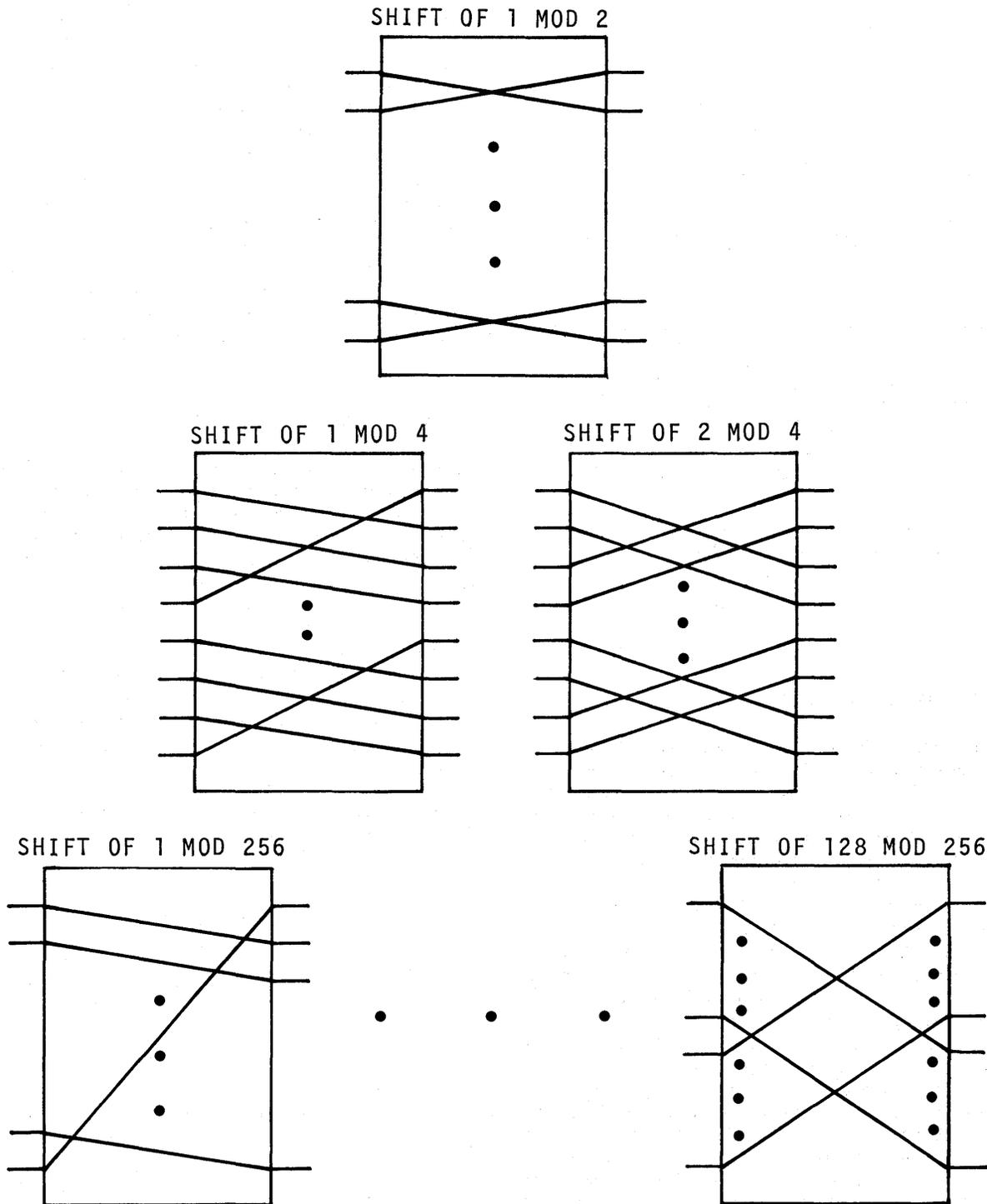


Figure 4—Flip network rotations

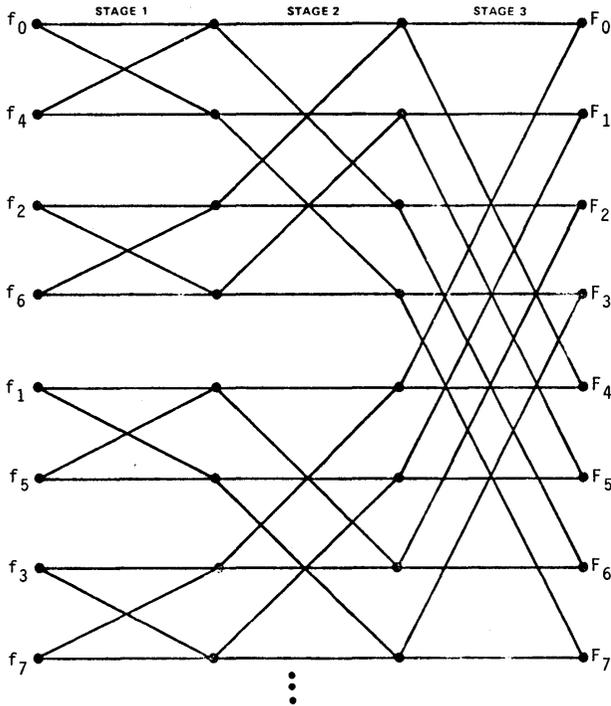


Figure 5—Butterfly diagram for FFT

**SPATIAL DEPENDENCIES**

Image data is inherently two dimensional and processing algorithms often must deal with this “built in” organization. In general, there are two different types of relationships. The neighborhood relationship requires that pixel neighbors in the two dimensional plane be readily accessible. The second type of relationship is normally spatially more extensive and is typified by the Fourier Transform where the pixel and its neighbors at power of two intervals for an entire row or column are related.

The STARAN associative arrays are well suited for accommodating both types of dependencies. Each array, as shown in Figure 3, has a flip network located between the memory and PE portions. This network performs as a specialized shift register and provides great flexibility in accessing pixels which are spatially related.

In particular, the flip network can accomplish power of two rotates within power of two sized fields with no time penalty. That is, a 1 bit rotate in 128 2 bit fields; 1 and 2 bit rotates in 64 4 bit fields; 1, 2 and 4 bit rotates in 32 8 bit fields; on up to 1, 2, 4, 8, 16, 32, 64 and 128 bit rotates in one 256 bit field. (See Figure 4) This means that inter-word operations at these intervals can be performed in parallel at the same rate as intra-word operations. Moreover, all interword operations can be performed with only a slight penalty.

The flip network then provides a very efficient method of implementing algorithms such as the Fast Fourier transform which utilize the spatial power of two interrelationships be-

tween pixels. This power of two spatial relationships is frequently expressed in the butterfly diagram shown in Figure 5. A detailed discussion of how the FFT can be implemented in the STARAN in log N steps of 1 add, 1 subtract, 2 real multiples and 2 exchanges has been published elsewhere.<sup>4</sup>

Template matching and spatial convolution are two algorithms which require the neighborhood type of pixel access. These processes can be implemented with little or no time penalty for the required spatial relationships. In particular, 2x2, 3x3, 5x5, 9x9 and 17x17 displacements require no time penalty. Other displacement amounts up to 16x16 require at most one extra shift except for 12x12 and 14x14 which require two shifts. In general, the required shifting for processing any window or template size within the 256 word array (i.e. 255x255 or less) is insignificant in overall algorithm time.

**PARALLEL DECISION MAKING**

An important aspect of image processing and pattern recognition is decision making. Many algorithms perform different operations as a function of the data. The complexity of the process is typified best perhaps by scene analysis techniques. In these situations it is essential to be able to record the exact state of each individual datum in the image.

Each STARAN array contains a special register which can be set as the result of searches (LE, GT, etc.) and arithmetic operations. This mask (or M) register can be used to select a subset of the words in an array to participate in subsequent operations. The results of tests, conditions and states can be stored, retrieved and operated on to achieve any desired logical combination of tests. Figure 6 indicates the physical location of this register.

Two examples of how the M register can be used are template matching and hierarchical structuring. To start a 3x3 template match, the M registers are set to all ones so

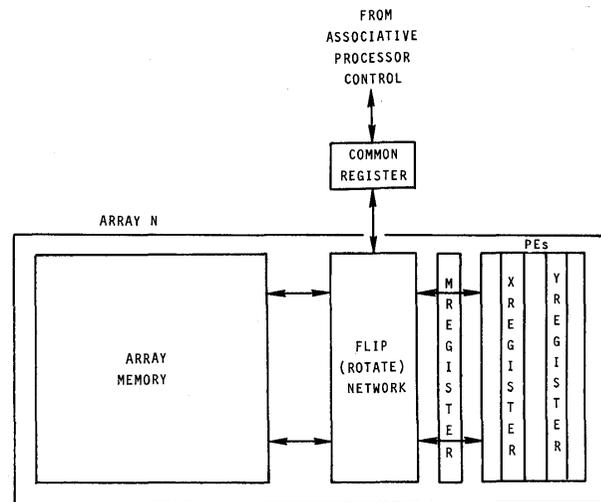


Figure 6—M register configuration

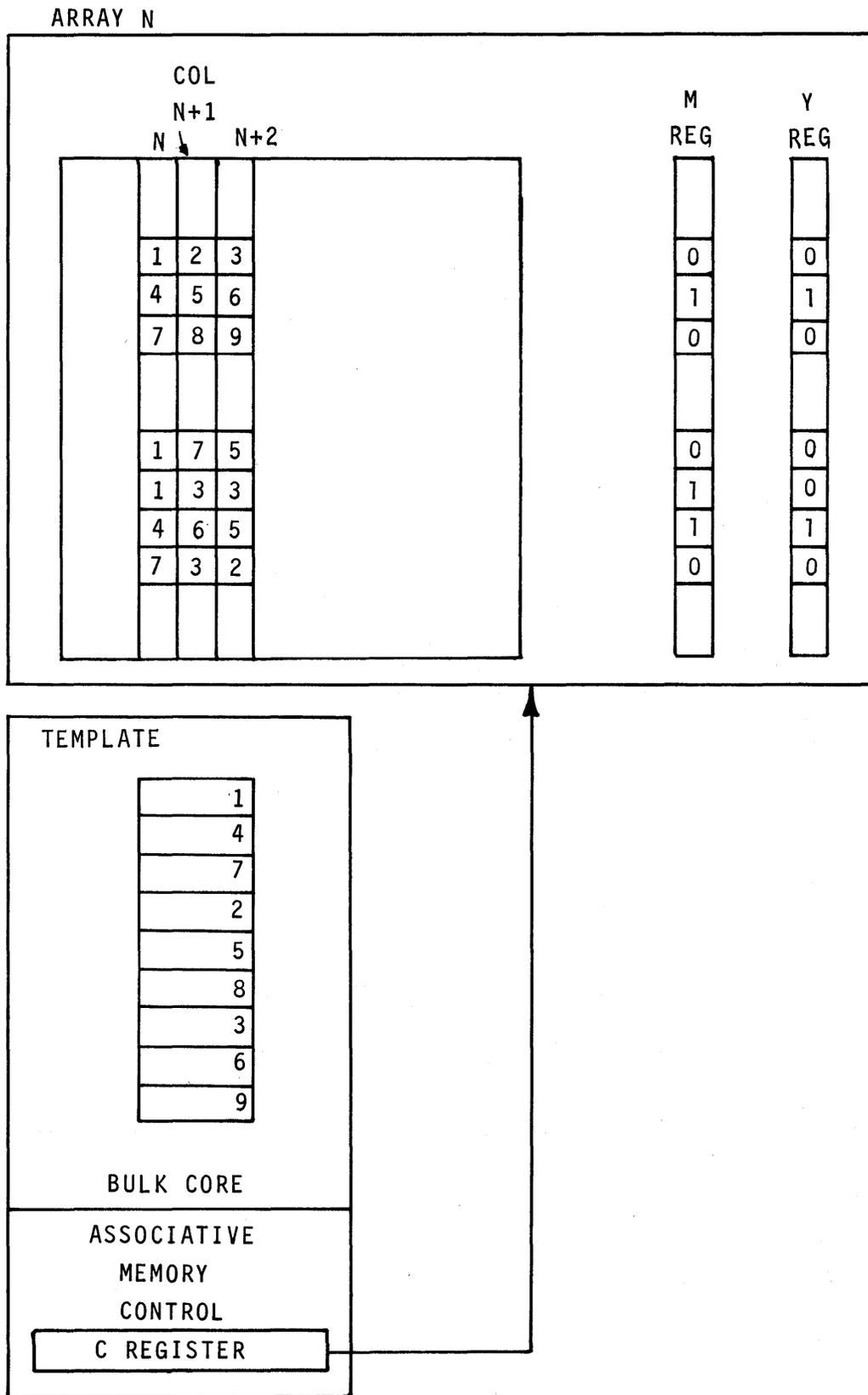


Figure 7—Template matching

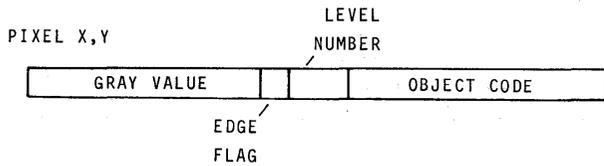


Figure 8—"Typical" scene analysis configuration

that every word participates in the search. The first value of the template is loaded into the common register and the first columns of all the arrays are matched against it. Every successful match is recorded in the Y register. The Y register is shifted down one and moved to the M register, the second template value is moved to the C register and the column is searched for a match of the second template value but on only those words with a corresponding one in the M register (i.e., only those which successfully matched the first value). Consequently at the end of the second search the Y register contains a one for every word and only those words which successfully matched both the first and second template values (the state shown in the M and Y registers in Figure 7). The procedure is repeated for each of the template values with an upward shift and column increment after searching for the third and sixth values. At the end of this process, the M register of all enabled arrays will contain a one (shifted down by two) corresponding to every successful template match covering columns 1, 2 and 3. The process is repeated for every set of columns to be searched.

Note that the columns need not be contiguous to be searched. Thus two situations can be easily handled. First, the columns can be organized in an order which facilitates another algorithm and/or input/output situations. Second, the template need not be contiguous but may cover essentially any size area in any desired manner. Both of these capabilities emphasize the flexibility of data processing in the STARAN arrays.

In a scene analysis application, each pixel might have a set of flags and auxiliary fields associated with it as shown in Figure 8. Then searches of the type "obtain the edges of object 5 on level 2" would be easily implemented in parallel by: first, searching all pixels in a column for object 5, then searching the matched words for level number 2 and then ANDing the edge flags with the Y register. The result is the answer to the search for the given pixel column. The process would then be repeated for each column under consideration. This example illustrates how easy it is to save the result of previous algorithms as flag vectors and codes in parallel and that this information is readily available for subsequent processing and analysis.

**INPUT/OUTPUT VERSUS COMPUTATION**

It has been established that in general, image processing involves large volumes of data. The algorithms vary quite markedly however in the degree of computation involved. Simple grayscale remapping is such a useful function that

special hardware circuitry is often contained within the display devices. In order to make such changes permanent, however, a computer must be used. Operations such as changing (remapping) the grayscale values of images require only one operation per pixel but must be performed on every pixel. These algorithms represent the high I/O-low computation end of the spectrum. At the other end are such algorithms as the domain transforms. Frequently these procedures require numerous arithmetic operations on a per pixel basis. For example, a two dimensional Fast Fourier Transform for a 512x512 pixel image requires 54 multiples and 90 adds per pixel (for a serial computer).

The STARAN has three I/O paths. The common register path (shown in Figure 1 and at the top in Figure 9) operates at between 12-15 million bits per second (MBPS). It is most useful for "broadcasting" data such as constants and parameters to all arrays in parallel. It is a 32 bit wide path.

The most useful path for data I/O is the 32 bit wide multiplexed I/O bus into and out of each array. This bus is capable of operating at between 80 and 640 MBPS. Thus an entire 512x512 8 bit per pixel image can be input or output in from 26.2 to 3.3 milliseconds. This data path is connected to a crossbar switch so that it can be used to transmit data between arrays as well as to peripheral storage or image display devices.

The fastest bus is the 256 bit parallel I/O bus which can operate from 512 to 2560 MBPS. The data transfer rate on this bus is such that special peripheral configurations are

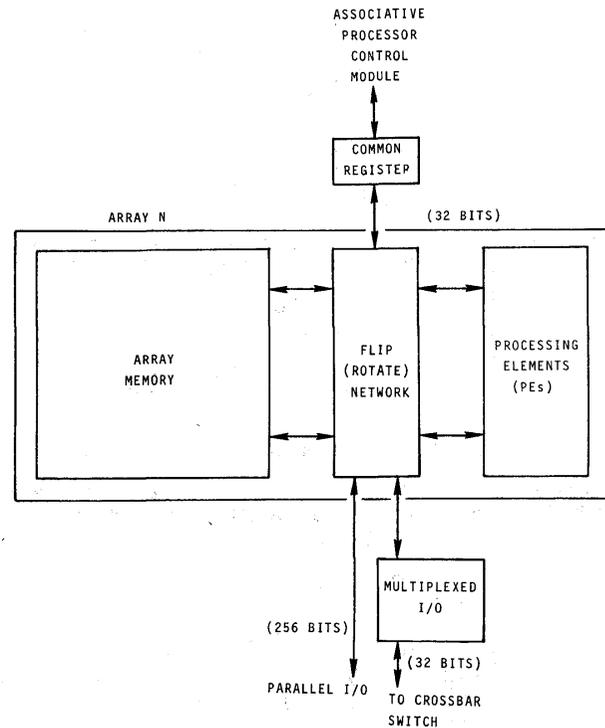


Figure 9—Array I/O paths

required. Consequently, it is best suited for special purpose applications.

Every array module is capable of being connected to either of two controllers. Thus in those applications where the equivalent of double buffering is desirable, some of the arrays can be switched to an I/O controller while the remainder are used for computation.

## SUMMARY

The multiple data stream parallelism of the STARAN allows it to perform a "serial" algorithm on up to 8192 data elements simultaneously. This is important in image processing where large volumes of data are processed. Images are inherently two dimensional, but the large array size of the STARAN E, readily allows an entire 512×512 8 bit/pixel image to be stored in one array with essentially its two dimensional topography intact. The array addressing structure and the array flip network provides easy access to every pixel and its neighbors in a simple efficient manner. The mask (M) register operation in an array enables complex decision processes to be made on any subset of pixels in an array. All of the above aspects of the STARAN's architecture would not be valuable if it could not be efficiently used. The three I/O paths into every array provide the capability to efficiently broadcast parameters and constants as well as loading and unloading image data in an expeditious manner. Thus it is apparent that many of the architectural features

TABLE 1.—Sample Algorithm Processing Times

Function	Description	Image Size	Speed*
Magnification	2.5× cubic convolution interpolation	512×512 8 bit/pixel	588 milliseconds
Convolution	3×3 window	512×512 8 bit/pixel	700 milliseconds
FFT	1 Dimension	512 16 bit/pixel	2.7 milliseconds

\* Measured times in the STARAN B machine exclusive of I/O. The STARAN E instruction execution time is approximately 20 percent faster.

of STARAN are ideally united for image processing and pattern recognition. This conclusion is confirmed by the execution times shown in Table I.

## REFERENCES

1. Batcher, K. E., "STARAN Parallel Processor System Hardware," 1974 *Computer Conference*, AFIPS Conf. Proc., Vol. 43, pp. 405-410.
2. Batcher, K. E., "STARAN Series E," 1977 *International Conference on Parallel Processing*.
3. Goodyear Aerospace Corporation, "The STARAN E System—An Overview," GAC Document Number AP-123226, 29 September 1976.
4. Goodyear Aerospace Corporation, "Application of STARAN to Fast Fourier Transform," GAC Document Number GER-16109, 31 May 1974.
5. Rohrbacher, D. and J. L. Potter, "Image Processing with the STARAN Parallel Computer," *Computer*, Vol. 10, No. 8, August, 1977, pp. 54-59.
6. Gambino, L. A. and B. L. Schrock, "An Experimental Digital Interactive Facility," *Computer*, Vol. 10, No. 8, August, 1977, pp. 22-28.

