Journal of Microcomputer Applications (1989) **12**, 213–232

Design of a hardware preprocessor for the 'DIAS' multilevel picture information system*

Nikolas G. Bourbakis and David F. Thurston†

George Mason University, Department of Electronic and Computer Engineering, Machine Vision Lab, Fairfax, VA 22030 and †Norfolk Southern Corporation, Research and Tests Lab, Alexandria, VA 22314 USA

DIAS is a multilevel-multiprocessor picture information system architecture. It receives many pictures simultaneously from its environment and 'decides' how to process them according to its built up experience.

This paper deals with the hardware design of the image preprocessor for the DIAS system. Several details concerning the image acquisition are discussed, including a unique photoarray camera and a hardware preprocessor which act to acquire and reduce the amount of data provided to the Master-Planner processor of the DIAS structure. In particular, the image preprocessor receives the image data in parallel and generates a number of critical image parameters, such as average intensities of various picture regions, geographic locations of the informative picture regions, number of pixels, etc., appropriate for fast decision making by the Master-Planner processor. The internal circuitry and functions of the image preprocessor are presented as well as a simple VLSI realization.

1. Introduction

Pattern recognition and image processing technologies have advanced very rapidly over the past few years[1–6]. A major thrust in this area has been the development of fast Von Neumann and non-Von Neumann multiprocessor architectures that act on entire arrays before decisions can be made[7–9]. Fast algorithms have also been developed that enable conventional processors to be used in image analysis systems[10–11]. Generally, these systems are used for simple task oriented functions such as assembly line manufacturing or quality control inspection of parts. In a general sense, these systems are not easily adaptable to the overall solution of multiple image analysis/processing in real-time.

A new system has been proposed as a real-time multiprocessor/multilevel picture information system "DIAS"[12]. The DIAS system consists of five main parts: (i) the decision-making and planning part, which uses a powerful processor as Master Planner; (ii) the service, orders distribution and control part consisted of a set of Service Distributor-Processors (SC); (iii) the multiprocessor execution part; (iv) the Backend part; and (v) the photosensitive input part.

Of primary importance in a machine vision system is the conversion of light (whether visible to the human eye or not) into a machine readable form. A vast majority of existing imaging cameras today rely on raster scan devices, such as vidicon tubes or CCD arrays[13–15]. In addition, raster scanning is not always the best method for image

*This work is a part of the DIAS project.

0745-7138/89/030213+20 $03.00/0

analysis, and scanning the entire scene to locate a small object in the lower left corner can be very time consuming[16]. A satisfactory solution is a photoarray with extra hardware features, such as capabilities of partial average intensities, capability for total average intensity, capabilities for a variety of scanning patterns, and parallel output of the received image data[13]. In order to support the above features in real-time, it is necessary to design a special hardware preprocessor. In this paper, we present the design of that preprocessor which is able to provide the above features (parameters) to the Master Planner Processor. In addition, a VLSI organization for this preprocessor is also provided.

This paper is organized in six sections. Section 2 presents the global configuration of the DIAS system in brief. Section 3 describes the structure and the operation of the image preprocessor. Section 4 discusses the circuit realization of the image preprocessor. Section 5 proposes a VLSI configuration of the image preprocessor and the last section summarizes the overall presentation.

2. DIAS picture information system

The overall organization of the DIAS multilevel picture information system[12] is illustrated in Figure 1. The DIAS system receives images either through a set of photoarrays (PA) from the environment or from a storage area. The picture information is conducted into a set of preprocessors (C) which can perform a few simple tasks on the image data extracting the critical image parameters, such as average intensities of various picture regions, number of pixels per region, locations of informative areas, etc. Since the preprocessors must function very quickly, they are implemented in hardware. The image parameters are carried over to a Master Planner (MP) processor for further evaluation and formulation of the processing plans. The Service Controller (SC) receives the abstract plans from the MP processor and schedules the synchronization and implementation of these plans on the multiprocessor array (IA). The image data are sent, on command by SC, to the IA processors. This multiprocessor array is composed of several special purpose processors in an efficient parallel/pipelined scheme to perform the tasks deemed too difficult for the preprocessors. Throughout the operation, the SC processor monitors the status of the IA array by interrupt-driven accesses to the interface buffer (B), and updates the MP processor accordingly. A unique feature of the DIAS structure is its Backend processors (BE). The BE processors perform the output operations and accumulate statistical and experiential data about the performed image analysis/processing tasks, which in turn, are supplied to the MP processor for adaptive planning based on experiential learning and making quick intelligent decisions.

3. Structure and operation of the image preprocessor

The overall I/O configuration of the image preprocessor is shown in Figure 2. In particular, each image preprocessor communicates with three system parts: (i) a 2-D photoarray with scanning facilities; (ii) the Master Planner Processor; and (iii) the Multiprocessor Array.

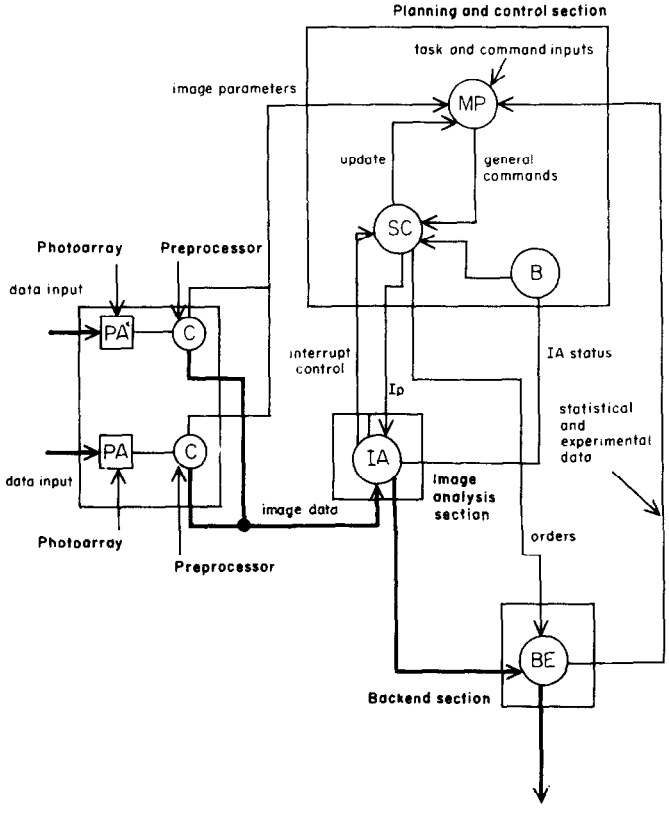

Figure 1. Organization of DIAS.

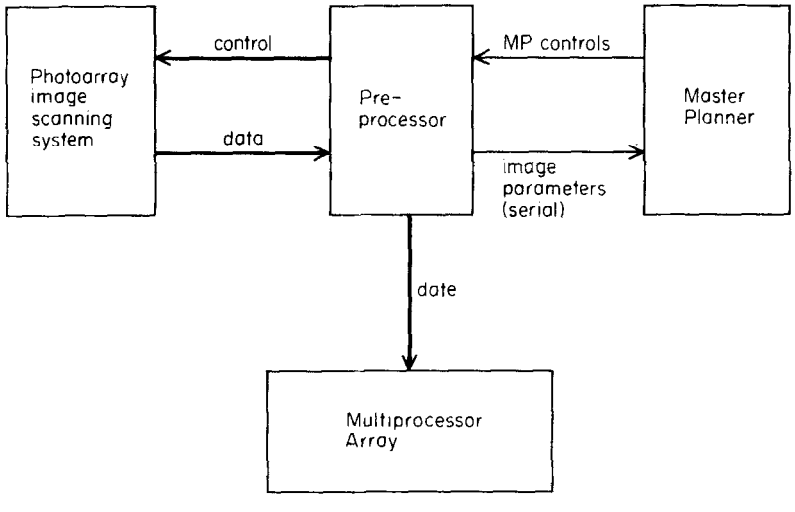

Figure 2. I/O schematic for preprocessor.

3.1 *Photoarray*

Each 2-D photoarray includes extra hardware circuitry facilities for scanning purposes, as shown in Figure 3. In particular, the hardware circuit for scanning [13] provides capabilities such as: (i) any sequential scanning of the photoarray cells with the same speed by using a programmable counter which increments one of a number of scanning decoders that places the appropriate pixel address on the counter buss. Each decoder is selected through a DEN_i address by the device that is controlling the array; (ii) scanning only portions of the photoarray using a starting number and a stop number on the programmable counter. This is a way to speed up processing when the location of interest in the 2-D array is known; (iii) producing digital grey scale data from pixels and regions within the 2-D array. Note that, whenever a scanning method is not chosen, the special Mean Value decoder is enabled directing one or more pixels to supply current to the Main Amplifier. The current is introduced to an A/D converter and the output is an n bit grey scale digital number. Since the amplifier switches deliver currents to the Main

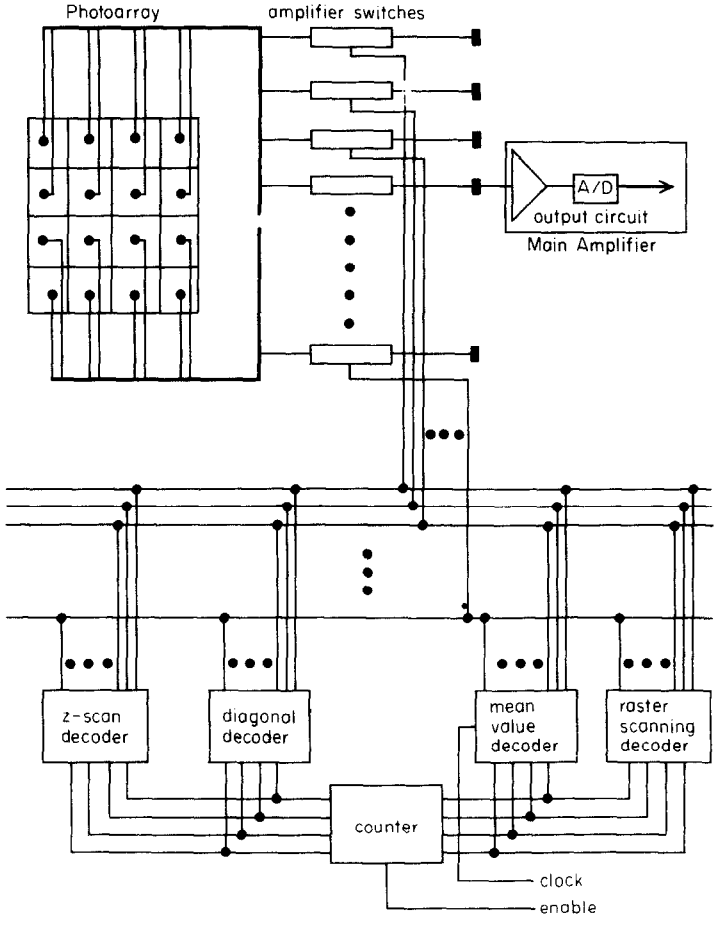

Figure 3. Photo diode array circuits.

Amplifier, a set of AGC (Automatic Gain Control) devices are used to provide the average intensity of the whole picture or part of it. In the case that a set of Main Amplifiers is used, the 2-D photoarray provides in parallel intermediate average intensity from various square parts of the array.

3.2 *Image preprocessor I/O scheme*

The I/O scheme of the image preprocessor includes five groups of information flow. Two parallel groups are used to communicate with the photoarray, two serial groups are used to communicate with the MP, and one parallel group is used to convey data to the Multiprocessor Array.

The two groups used with the Photo Diode Array consist of control bytes to load registers or programmable counters in the array, and data bytes of grey level information from the array. Control bytes consist of Counter start and stop bytes, the Mean Value Decoder Address bytes (the number of these is dependent on the number of A/D circuits in the array), and the DEN_i Scan Decoder Enable address byte (the length of this byte is dependent on the number of Scan Decoders in the array). In addition, the EN Counter Enable line is also sent to the array from the preprocessor. This line determines if the array is to clock through a scan of the array or to find an average value of a region in the array by selecting the Mean Value Decoder. Data bytes are grey level values of pixels or regions depending on the Mean Value Decoder address given to the array. The number of data bytes is also dependent on the number of A/D circuits in the array. These bytes are not multiplexed and have their own busses to use as required by the array.

The two groups used with the MP are high speed serial channels that convey controls from the MP to the preprocessor and image parameters that the preprocessor has determined from the array grey scale data to the Master Planner. Figure 4 shows the protocol to be used in these serial communications. As this illustrates, control information consists of scanning techniques, threshold levels for processes inside the preprocessor, image parameter requests, scan clock start and stop bytes, region of interest scanning, and grey level data path direction (from array to preprocessor or from array to MP). Image parameters are determined by the preprocessor and consist of the following:

(1) Binary picture—this bit is active when only two significant grey levels are detected.
(2) Homogeneous picture—this bit is active only when one significant grey level is detected.
(3) Total average—this eight bit byte is the average grey level of the entire picture.
(4) Number of pixels—this is the size of picture being scanned.
(5) Number of informative pixels—this is the number of pixels that are over a specified threshold.
(6) Number of grey levels—this is the number of grey levels in the area being scanned.
(7) Geographic locations of informative objects—this is a coded byte that has a unique address for regions in the picture that have a grey level average above a chosen threshold.
(8) Average intensity of these geographic locations—this is the average grey level for the informative objects.
(9) Size of the informative objects—this is the overall size of the informative geographic locations.

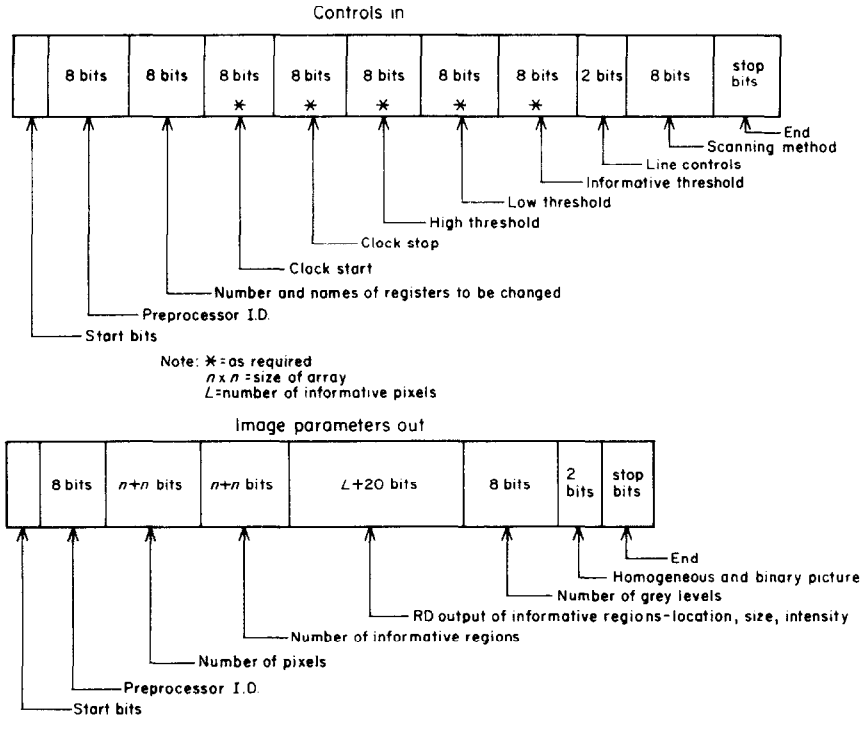

Figure 4. Preprocessor serial protocols.

The last group is the path for data from the preprocessor to the Multiprocessor processor array. A/D data from a scan of pixels in the array can be directed to the Multiprocessor Array on command of the MP for complex parallel processing. This generally occurs after the image parameters have been received by the MP and decisions about the image have been made.

4. Image preprocessor circuit realization

A simple internal block diagram of the preprocessor is shown in Figure 5. It consists of a System Clock, Switch, Communication Controller and Array Control Processor (ACP). Figure 5 also illustrates the internal connections as well as the external connections previously discussed. The Array control processor (ACP) is the heart of the preprocessor. The Clock, Switch and Communication Controller act to support the actions of the ACP.

The System Clock is a line supplied to the ACP and to the Communication Controller. It directly generates all of the timing pulses in the preprocessor. In the Communication Controller, the clock is used to generate the baud rates for the two high speed serial data lines that communicate to the MP. In the ACP, the clock is used to increment encoders that supply address lines for register latching, correctly time delay pulses used in image data processing circuits, and increment Mean Value Decoder addresses to the array.

The switch block diagram is shown in Figure 6. Controlled by the ACP, the switch

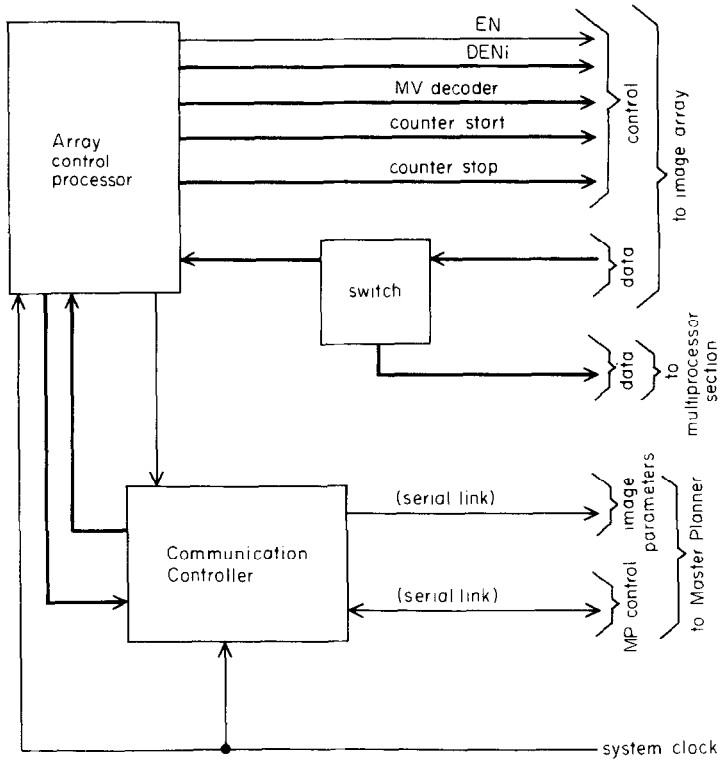

Figure 5. Internal structure of preprocessor.

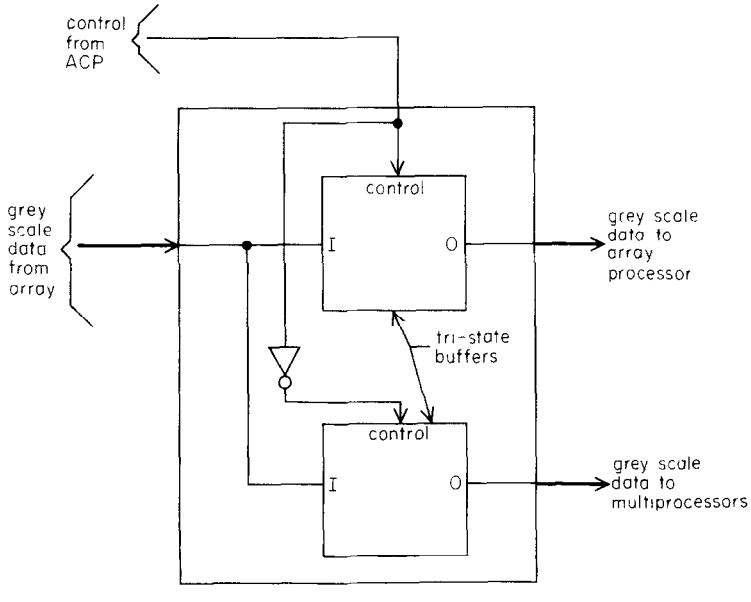

Figure 6. Switch operation.

directs data from the array to either the ACP itself, or to the Multiprocessor Array. In Figure 6, only one bit is shown, however, this is typical for all n bits in the switch. It should be noted that since data transferred to the Multiprocessor Array is a scanning of the array, only one A/D is required for this function.

The block diagram of the Communication Controller is shown in Figure 7. Through use of the System Clock as described before, the Communication Controller converts serial controls to parallel controls that are accepted by the ACP. These controls originate from the MP and are loaded into registers used in image parameter determination. Direct array controls such as Counter Start bytes and Decoder Enable bytes are also transferred through the Communication Controller in this fashion. Image parameter information comes to the Communication Controller in the form of bytes up to 20 bits long. As these bytes do not come to the Controller in a continuous stream, a FIFO

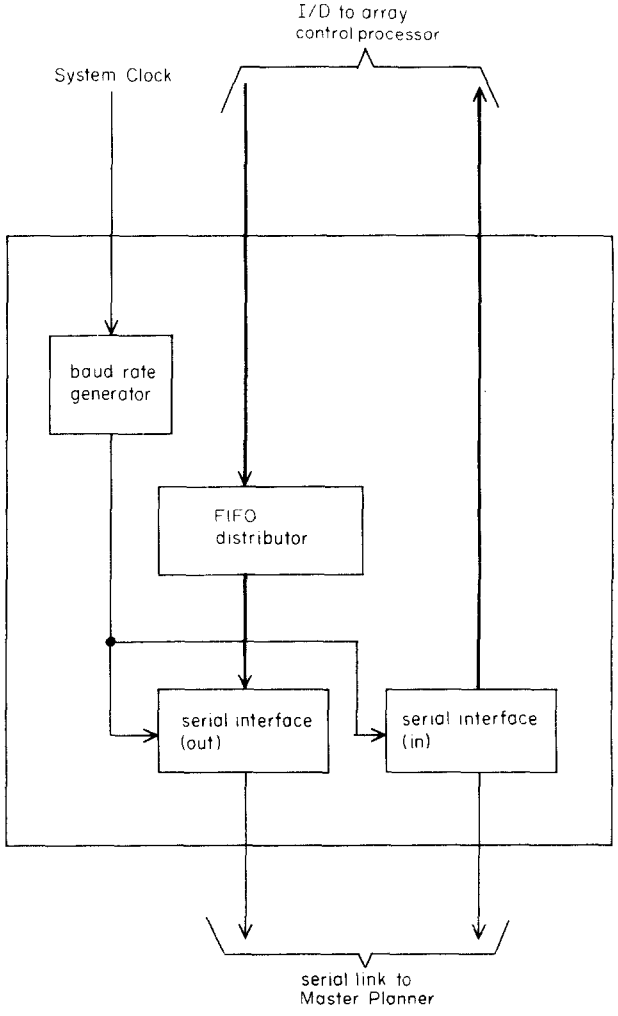

Figure 7. Communication Controller detail.

distributor stores the bytes as they do come in and passes them on to the serial interface when there are enough to ensure a continuous stream. This will prevent any disruption in data flow.

The ACP is broken down into Figures 8 and 11–16. Controls received by the Communication Controller and brought into the ACP are processed by the control decoders shown in Figure 8. The first few bytes are taken off the buss and decoded by the array address decoder. Here, the preprocessor address is checked against the ID in the decoder. If the ID does not match, the rest of the transmission is ignored. This action allows more than one preprocessor on each line from the MP. In addition to the local address, information about the control transmission is also obtained in the address decoder. For instance, the number and name of any registers contained in the ACP that are to be changed can be determined in the control decode operation. By transmitting only those registers that need to be changed, the serial transmissions from the MP can be kept as short as possible.

The detection of objects in the image is of prime importance to any general purpose vision system. Several methods can be used to extract this information, however, Regular Decomposition has proven to be very useful in determining many of the parameters required by the preprocessor. Figure 9 illustrates a quad tree data structure of an image. Each node represents the average grey level intensity of the four nodes below it. The top node is the average grey level intensity of the entire picture. There is an obvious correlation between this structure of image data and the Photo Diode Array. Since any pixel or region can be represented by an A/D converter, the Mean Value Decoder can select the A/D converters to match the pixels or regions represented by the nodes in the quad tree data structure. With this in mind, a Regular Decomposition (RD)

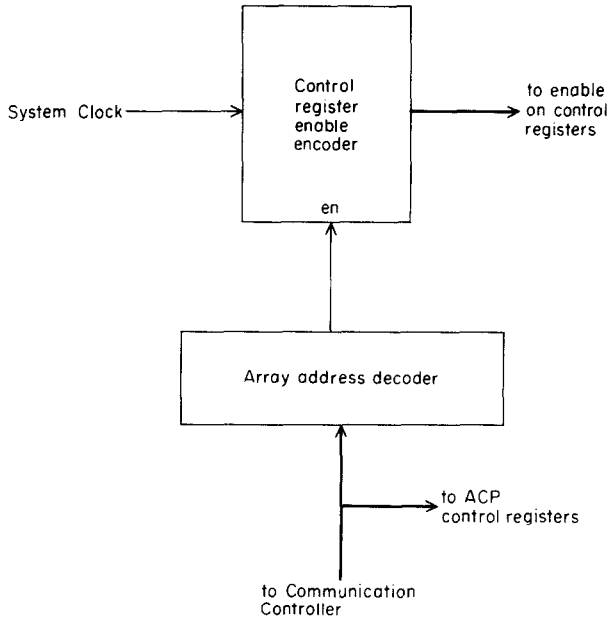

Figure 8. ACP control decode operation.

of the quad tree becomes trivial. The RD will threshold each node with given values to determine whether or not the node contains informative, non-sure (in which case the RD will continue to inspect nodes on that branch of the tree), or non-informative. The location of the informative node is known in the RD from the position of the A/D that is being used, and the output of the A/D is the intensity of the informative nodes. Here, however, is where some constraints come into play. With each level of the tree, the requirement for A/D increases dramatically. For instance, a common 256×256 array would require 87,381 n bit flash A/D converters, and 699,048 pins on the array as well as the preprocessor chip. This seems impractical using current technology. Although the complete complement of A/D converters will result in the fastest execution times for the RD, the high speed processing would be unnecessary. The serial data lines to the MP limit the amount of data that can be transferred out of the preprocessor. A more workable solution would be to limit the number of A/D converters and scan the array on blocks that the converters can address. The proper number of A/D converters is dependent on several factors. Hardware limitations include packaging and chip pinouts. Software limitations are restricted to storage memory in the MP and its ability to act on information received from the preprocessor. As stated previously, the preprocessor is to quickly establish certain parameters about the image. Therefore, the number of A/D converters is also dependent on the number of levels carried out in the RD. Figure 10 illustrates the amount of memory required for each level of the RD when all of the pixels are neither informative or non-informative[17]. As can be seen from the graph, the amount of memory required to store the information after the level 6 RD becomes

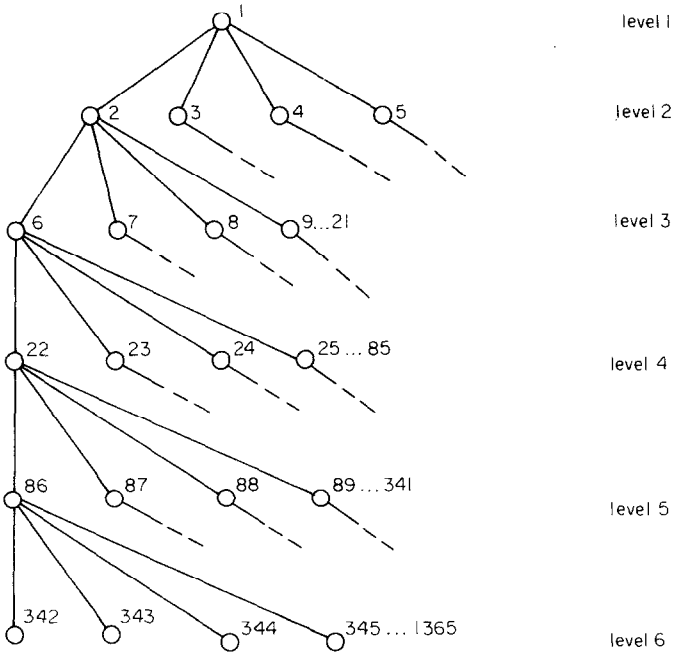

Figure 9. Quad tree pyramid (6 levels).

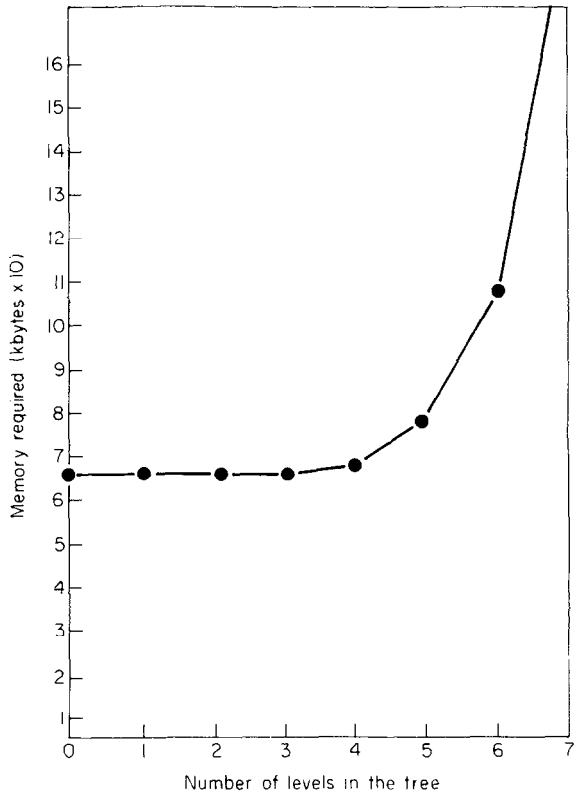

Figure 10. Typical memory requirements for a not sure image (256 × 256 array).

prohibitive. This would cause the serial transmission to the MP to increase to unacceptable durations, increase the MP memory requirements, and create hardware I/O problems (a large number of pins on the chip).

In an effort to achieve maximum utility of the converters in the array without sacrificing execution speed of the preprocessor, 21 A/D converters would be installed in the array. Twenty-one was chosen as it completes three levels of the RD in the first pass, and uses a manageable number of I/O lines. As explained previously, the RD will be executed to only the sixth level. This is equivalent to dividing the image into 1024 regions. A typical 256 × 256 image array would be divided into equal regions of 64 pixels each, and this represents the smallest region of interest for the preprocessor.

The Regular Decomposition carried out by the preprocessor uses A/D converter outputs from the array that are addressed to give the grey scale average of father and son nodes in the quad tree data structure (fathers are nodes that are above four others, and sons are the four that are below the father). Figure 11 shows the application of the first 21 A/D converters from the array. In addition to starting the RD, the first A/D is directed to the Total Average Register. This register is latched on the first read of the array, and read to the Communication Controller buss as one of the first bytes to be output by the serial line to the MP. Since it is the top of the quad tree, the grey level value

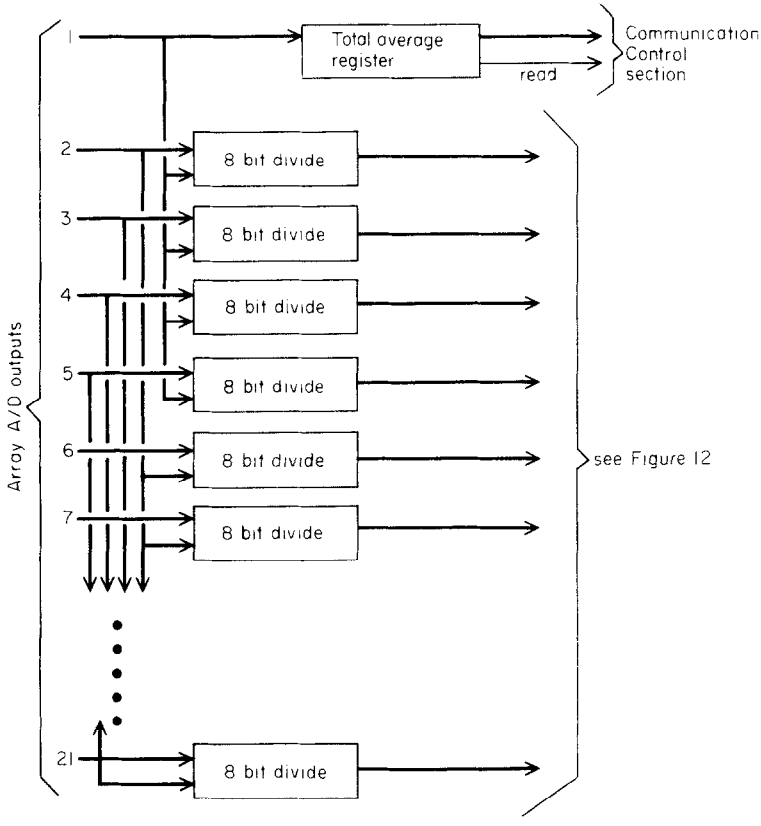

Figure 11. ACP RD circuits.

in this register is the average intensity of the entire array. The rest of the circuits in Figure 11 divide the sons by the father as a prelude to the RD. The scanning of the array will be discussed in the operation of the Mean Value Decoder Address Generator. The outputs from the n bit dividers are distributed to the comparators in Figure 12.

Note that there are two compares performed on each n bit divide output. Each compare also uses an input from one of two threshold registers that can be loaded on command from the MP. Experience and statistical information in the Back End Processor from Figure 1 will determine these values and if they are going to change. The upper compare (with the high threshold register) outputs an active signal when the divide output is of greater magnitude than the high threshold register value. This implies that the region (or son node being examined) is informative. The same divide output is compared to a low threshold byte from the low threshold register. This compare output goes active when the divide output is lower in magnitude than the value in the low threshold register. This implies that the region under consideration is non-informative. The informative signal lines are sent on to the circuits in Figures 13 and 14, while the non-informative signal lines are sent to the circuits in Figure 14. Figure 13 also shows the conclusion of the RD circuits. The informative lines are input to the 20 Region Address

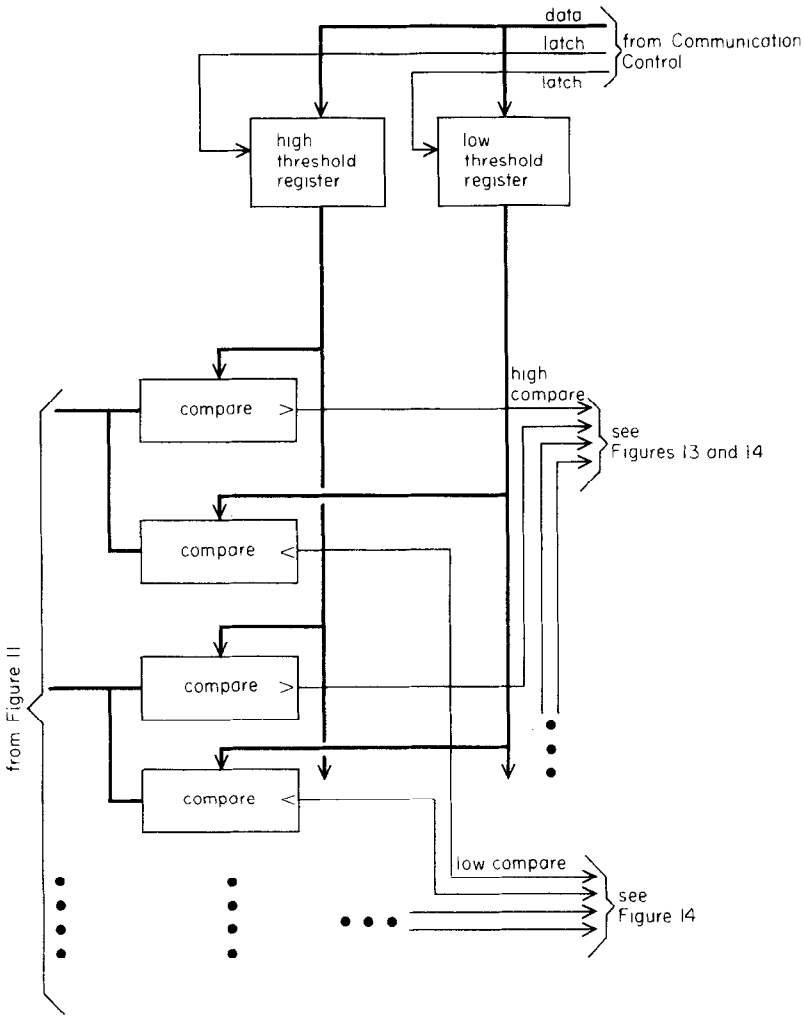

Figure 12. ACP RD circuits.

Encoders. If the region is determined to be informative, the current A/D converter grey level value and its associated Mean Value Decoder address are latched into the Region Address Encoder. In addition, these encoders receive a register clock input generated by a Mean Value Address Generator, and a daisy chain output control line. These lines allow the Region Address Encoder to output the intensity (grey level value), the position (mean value decoder address), and the size (level in the quad tree) of the informative regions. For example, when the register clock is active and the daisy chain input is high (because the encoder above this one finished its output, if any) the encoder puts the information on the Communication Controller Buss. Note that the encoder activates the Controller Buss only on input of an informative region.

The Mean Value Address Generator is shown in Figure 14. Its inputs are the informative and non-informative lines mentioned previously, and the system clock.

Based on information from a previous scan of the array as to whether regions are informative or not, the Mean Value Address Generator determines the next set of regions to be examined by the array's A/D converters. As the quad tree progresses, this circuit will skip those branches in the tree that have been determined to be informative or non-informative. In addition, this device generates the register clock to progress the flow of information to the Communication Controller, a write (WR) signal to the Communication Controller to latch information from its input buss, and an RD done line to another circuit in Figure 14 that count the number of informative regions. To accomplish this, the informative region lines are brought to a binary encoder that detects the number of lines that are active and converts it to a binary number. This binary number is added to a number that was previously the output of the adder through a latch that is delay triggered by the first register clock of each scan. The RD done line from the Mean Value Address Generator causes a buffer to accept the final latch output. This binary number is the total number of informative regions detected by the RD.

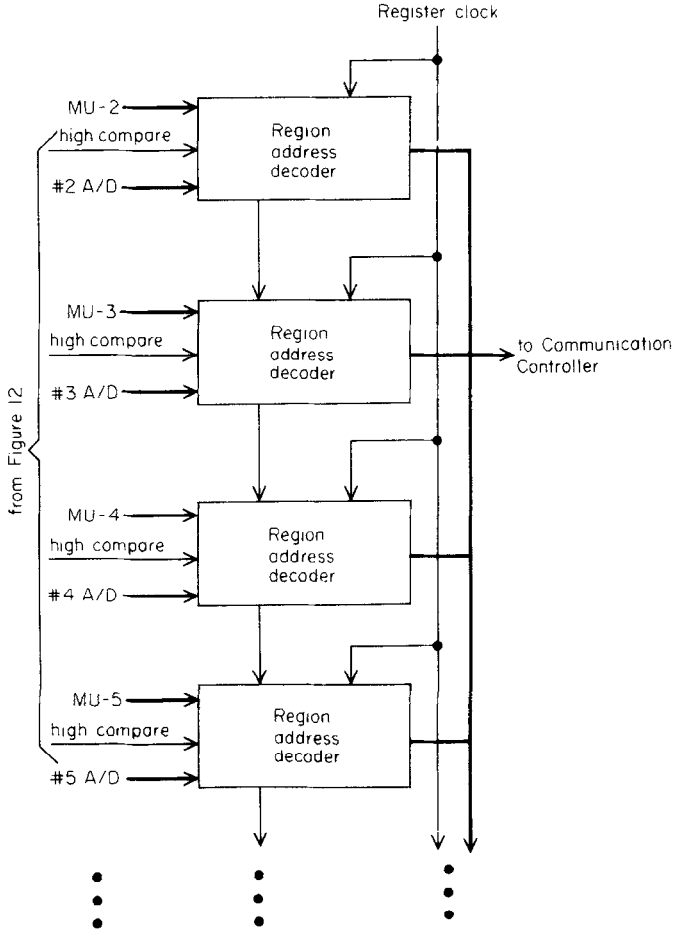

Figure 13. ACP RD circuits (cont).

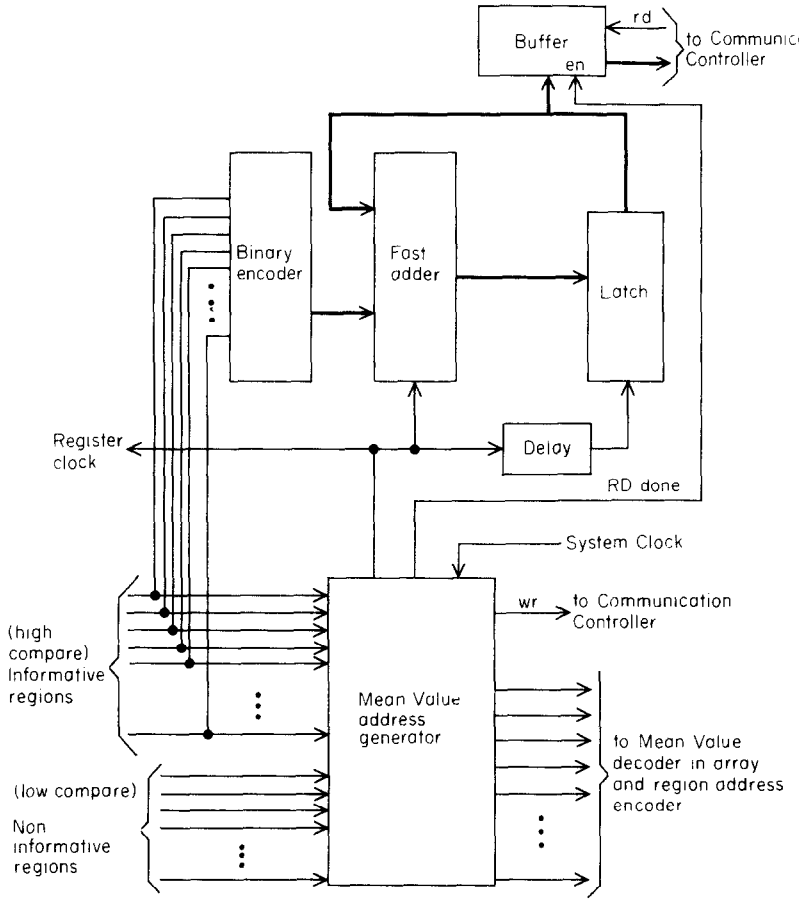

Figure 14. ACP mean value decoder.

The n bits that represent the grey level value of the informative regions in the Communication Controller buss are input to the FIFO buffer in Figure 15. This buffer is to assure that the circuits in Figure 15 do not get overwritten in a fast communication on the buss. The FIFO outputs bytes of region intensity to a grey level decode that functions as a one of 2^n decoder. These outputs are connected to two circuits. First, D flip-flops, incriminated by a delayed register clock, output an inverted Q to be ANDed to the original flip-flop input. The AND gate output is an active pulse that increments an n bit counter, whose output represents the number of informative grey levels. Note that the AND gates can only output one pulse per image. The output of this circuit is sent to the Communication Controller BUSs as number of grey level information.

The second circuit that the grey level decoder outputs are used consist of 2^n bit counters. Figure 16 is a continuation of this circuit. The counters accumulate like grey levels in a fashion similar to a Grey Level Histogram. The counter outputs are compared to a Significant Number of Grey Level Threshold. Again, this threshold is programm-

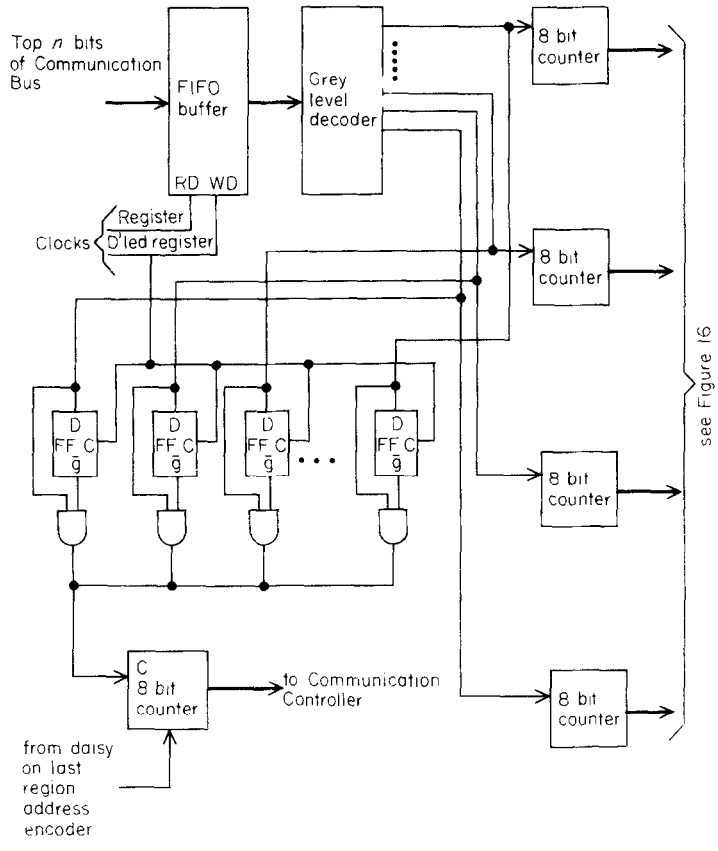

Figure 15. ACP grey level determination.

able by the MP through the control bytes in the serial line input to the preprocessor. The compare output goes active when the binary value from the counter is greater than the threshold level. These outputs are conveyed to a 2^n line to binary converter that determines the number of active lines and converts them to a binary number. The n bit output can determine certain parameters about the image. First, as shown in Figure 16, a simple NOR operation on all of the outputs implies that there were no informative regions in the image. This, of course, is a homogeneous picture. Second, if the least significant bit is ADDed to a circuit that NORed the remaining bits, the result would be the determination of a binary picture. These outputs are added to the Communication Controller Buss to be output as image parameters.

One other control function that the preprocessor performs is the programming of the array clock. Two bytes are required to load a start and stop value to the array. Figure 17 illustrates the circuitry involved in the operation. In addition to storing the start and stop bytes in registers similar to threshold registers, the operation of subtraction is performed on the two numbers. The result is the number of pixels to be scanned by the array. Since this is one of the image parameters requested by the MP, this value is output to the Communication Controller as data on the serial line.

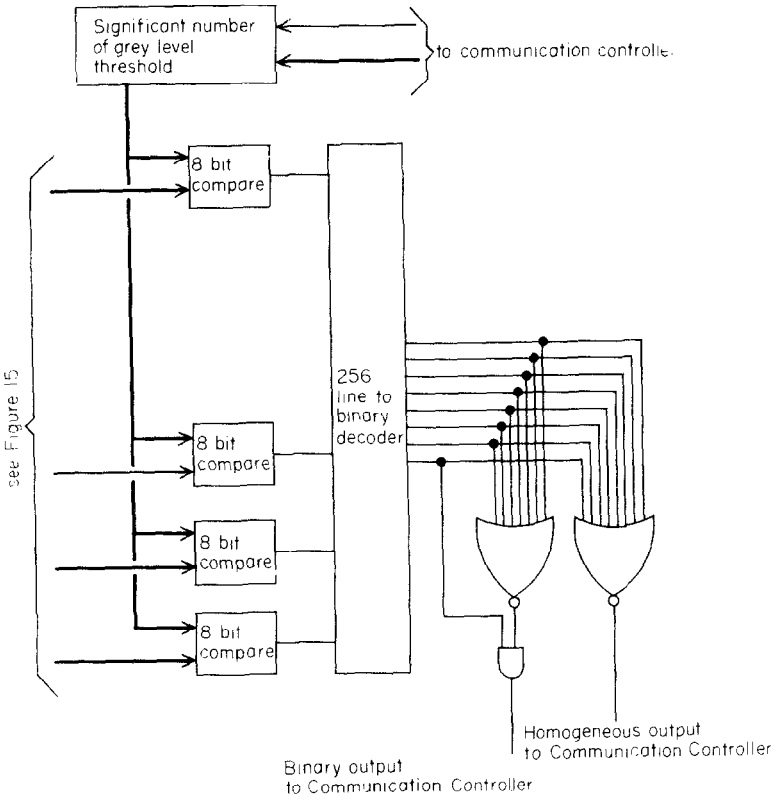

Figure 16. ACP binary picture circuitry.

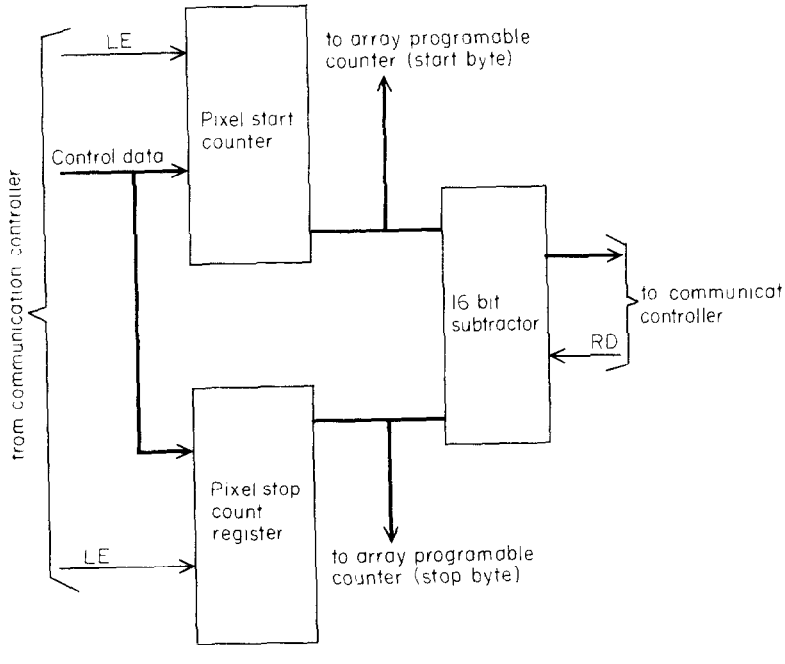

Figure 17. Array clock control.

5. Image preprocessor VLSI realization

The circuit design in the preprocessor has been limited to conventional technologies such as counters, flip-flops, etc. This provides for simple and reliable integration. The preprocessor is estimated to have approximately 250,000 circuit elements. This is certainly realizable with today's technology. The most difficult constraint is the pin out of the chip. As discussed previously, the A/D converters and the Mean Value Decoder Address lines will take the majority of the pins. Surface mount devices and pin grid arrays have been developed that contain large numbers of pins, and this is one solution to this problem. It is also possible to integrate the preprocessor with the array on the same wafer. Although this creates a very large chip, many of the I/O problems would be solved.

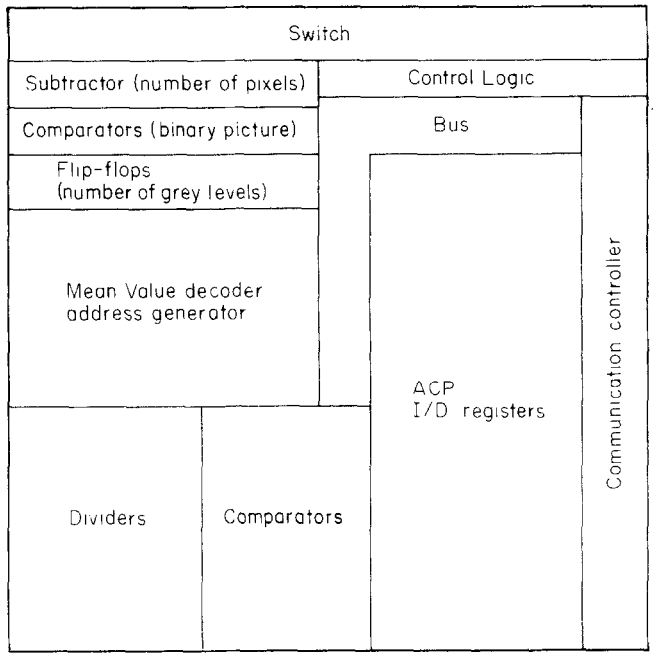

Figure 18. Preprocessor VLSI realization.

Figure 18 illustrates a VLSI realization of the preprocessor. Areas on the chip have been provided for each circuit that was discussed previously. These areas are in contact at a boundary whenever two or more areas require communication with each other. Larger chip surfaces are provided for areas that will need a great number of I/O lines or circuit components, such as the Mean Value Address Generator.

6. Conclusions

The discussions up to now have dealt with black and white grey levels. However, the circuits could easily be adapted to color grey levels. In black and white, the grey levels represent the variations between black and white. An 8-bit A/D converter will distinguish between 256 different levels of grey between black and white. In color, the three primary (or subtractive) colors are separated and represented by its own "grey level". This grey level is actually just shade or intensity of the color in the picture. By joining the color grey level data into one byte, the existing circuitry will be able to handle the switch to color. If the number of bits from the color grey level A/D converters exceeds the number of bits from the black and white A/D converters, then the circuit realization will grow dramatically.

As vision technology advances, the ability of machines to perceive images will be greatly enhanced. It is important to remember that conventional theories in machine vision may have to be reworked in order to find the optimum solution to the general problem. Different scanning techniques, as well as processing hardware, may change dramatically before design considerations are met.

References

1. T. Pavlidis 1982. *Algorithms for Computer Graphics and Image Processing*, Oxford: Computer Science Press.
2. K. S. Fu 1982. *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, NJ: Prentice Hall.
3. R. Offen 1985. *VLSI Image Processing*, New York: McGraw-Hill.
4. R. Gonzalez & P. Wintz 1987. *Digital Image Processing*. Reading, MA: Addison-Wesley Publishing.
5. A. Rosenfield 1987. Picture processing. *International Journal on CVGIP*.
6. N. Bourbakis 1985. Picture classification techniques. *Technical Report, GMU-ECE-TR-10-1-*, p. 102.
7. N. Bourbakis, *Knowledge Based Vision Architectures and Languages*, for publication, p. 347.
8. M. Duff 1983. *Computing Structures for Image Processing*. London: Academic Press.
9. E. Danielson & S. Leviald 1981. Computer architectures for pictorial information systems. *Computer*. November 1981.
10. Computer vision and pattern recognition: *Proceedings of IEEE Conference, Miami, Florida*. 1986.
11. Computer vision 1987. *Proceedings of IEEE Conference, London, 1987*.
12. N. Bourbakis & P. Ligomenides 1985. High performance architectures for real-time multilevel picture information systems. *Proceedings of IEEE Workshop on LFA, Spain, June 1985*.
13. D. Panagiotopoulos & N. Bourbakis 1984. The VLSI design of a 2-D image processing array. *International Journal on MM*, Vol. 14.
14. A. Behpour & N. Bourbakis 1987. A hardware implementation of a 2-D photoarray for array processors. *Proceedings of IEEE Conferences on JECON, November 1987*. Cambridge, MA.
15. N. Bourbakis, *et al.*, 1986. Image sensing devices: an overview, *Technical Report, GMU-ECE-TR-10-1986*, p. 62.
16. N. Bourbakis 1987. Image preprocessing algorithms. *Technical Report, GMU-ECE-TR-1-1-1987*, p. 115.
17. N. Bourbakis 1984. A real-time, hierarchical image reduction machine, *International Journal on MM*, Vol. 16, 1985.

G. Nicolaos Bourbakis received a B.Sc. degree in Mathematics in 1974 from the National University of Athens, and a Ph.D. degree in Computer Science and Computer Engineering in 1982 from the University of Patras. He is an Assistant Professor at George Mason University, Department of Electrical and Computer Engineering. His research interests are in knowledge-driven vision system architectures, algorithms and languages for computer vision, learning schemes and artificial intelligence, software environments for VLSI system design and languages evaluation, text processing, and neural-nets. He has published more than 70 articles in international journals and conference proceedings and he is the author of three books (in Greek). He is the General Chairman of the IEEE International Symposium on Architectures, Languages and Algorithms for Artificial Intelligence 1989, he is also Guest Editor for the international journals *Engineering Applications of Artificial Intelligence* and *Pattern Recognition and Artificial Intelligence*. He is listed in the Congressional Who's Who in Computers, and he is an IEEE senior member.

David F. Thurston received his B.Sc. in Electrical and Computer Engineering from Clemson University in Clemson, South Carolina, in 1977 and an M.Sc. degree from George Mason University in January 1989. He is a licensed Professional Engineer in the state of Pennsylvania.