

AD-A071 638

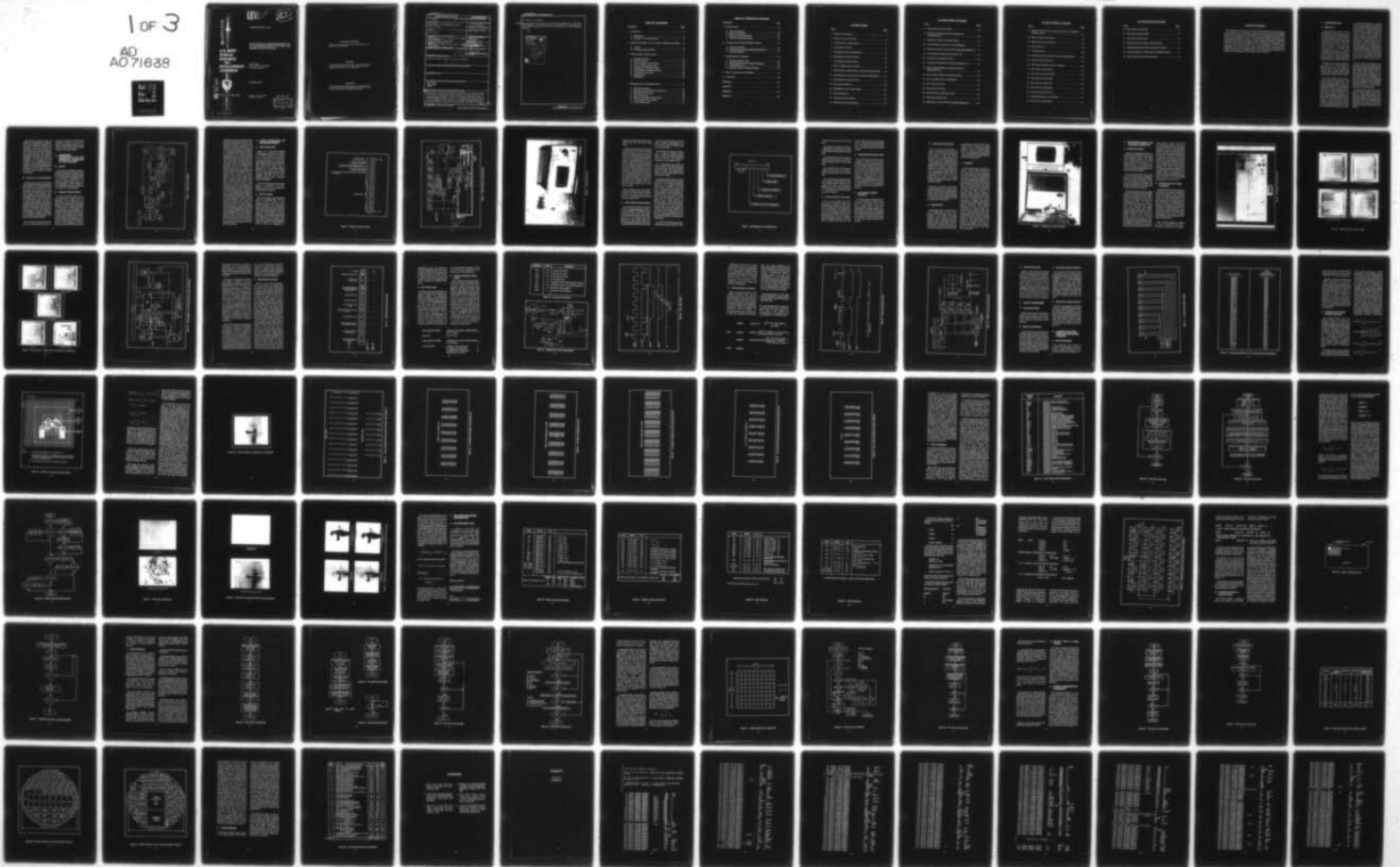
ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND REDSTO--ETC F/G 9/2
APPLICATION OF A MICROPROGRAMMED, BIT SLICE MICROPROCESSOR TO T--ETC(U)
SEP 78 L G MINOR
DRDMI-T-78-53

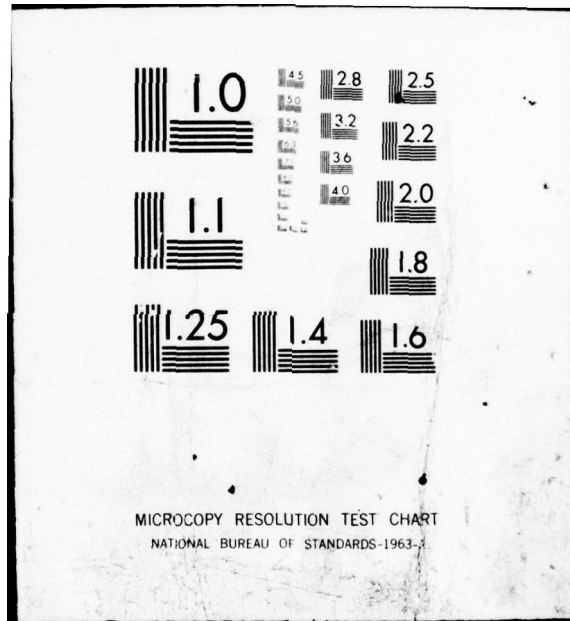
UNCLASSIFIED

NL

1 OF 3

AD
A071638





LEVEL IV

72

ADA 071 638



TECHNICAL REPORT T-78-53

APPLICATION OF A MICROPROGRAMMED, BIT SLICE MICROPROCESSOR TO THE REAL TIME IMAGING TRACKER PROBLEM

**U.S. ARMY
MISSILE
RESEARCH
AND
DEVELOPMENT
COMMAND**

Lewis G. Minor
Advanced Sensors Directorate
Technology Laboratory

30 September 1978

DDC FILE COPY



Redstone Arsenal, Alabama 35809

Approved for Public Release;
Distribution Unlimited.

DDC
RECEIVED
JUL 25 1979
REGULATED
D

DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL ENDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER T-78-53	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Application of a Microprogrammed, Bit Slice Microprocessor to the Real Time Imaging Tracker Problem.	5. TYPE OF REPORT & PERIOD COVERED Technical Report	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lewis G. Minor	8. CONTRACT OR GRANT NUMBER(s) 16 IT161101A91A	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander US Army Missile Research and Development Command ATTN: DRDMI-TE Redstone Arsenal, Alabama 35809	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA-PROJECT NO. 1T1.61101A91A AMCMS-611101.91A.0011	
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Missile Research and Development Command ATTN: DRDMI-TI Redstone Arsenal, Alabama 35809	12. REPORT DATE 30 September 1978	13. NUMBER OF PAGES 255
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release. Distribution Unlimited. 14 DRDMI-T-78-53		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) SOLID STATE IMAGE DEVICE TRACKER		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Following sections detail the design and construction of an imaging tracker built inhouse. The program was very successful and nearly all of the objectives in the original scope of work were achieved. The program's major accomplishments are listed as follows: (1) Designed, built, and tested the first Army imaging tracker capable of correlation and adaptive gate optical contrast tracking, (2) the feasibility of trackers using high speed microprocessors and CCD TV technology was demonstrated, (3) several correlation (over)		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393 427

next page

JOB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract (Continued)

cont. and template matching techniques were studied and implemented in real time microprocessors. Though the tracker was built primarily as a research tool, the hardware concept has the potential for achieving a very small packaging size.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	<input type="checkbox"/>
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

Description	Page
1. Introduction	9
A. Background	9
B. Summary of Accomplishments	10
2. Hardware Description of the Computer Interface to the Tracker ..	10
A. General	10
B. Computer Interface Details	10
3. Video Interface to Microprocessor	12
A. General Description	12
B. CCD TV Camera	12
C. Video Interface Control Latches	16
D. Row and Column Count Logic	18
E. Tracking Gate Generation Logic	18
F. Image Memory Address Generation	18
G. Conversion Control Logic	19
H. Image Memory	19
I. TV Monitor	19
4. Microprocessor and Tracker Hardware	21
A. Physical Description	21
B. Microprocessor and Tracker Architecture	21
C. Microinstruction Format	26
D. Bus Control Logic	28
E. Function Generation Circuit Details	28
F. Loop Counting Circuit Details	31
G. K Bus Encode Logic	34

TABLE OF CONTENTS (Continued)

Description	Page
5. Utility Software	34
A. General Description	34
B. Memory Test Software	34
C. Microcode Loading Procedure	34
D. Macrocode Loading Procedure	34
6. Correlation and Template Matching Tracker	34
A. General Description	34
B. Correlation and Template Matching Algorithms	37
C. Tracker Description	47
7. Detailed Software Description	56
A. EAI 640 Emulation Code	56
B. Addressing Technique for Image Processing	64
C. Tracker Microcode	67
D. Execution Times for Tracker Routines	76
8. Power Consumption and Packaging	76
9. Conclusions	82
References	85
Appendix A	87
Appendix B	101
Appendix C	189

ILLUSTRATIONS

Figure	Page
1. Memory block diagram	11
2. Function of bits in DF latch	13
3. Video interface to microprocessor	14
4. Fairchild 201 CCD TV	15
5. Bit assignments in the MODE latch	17
6. Photograph of tracker hardware	20
7. Photograph of operators panel	22
8. Macro and Micro memory boards	23
9. Microprocessor, image memory, and video front end boards	24
10. Block diagram of microprocessor and tracker architecture	25
11. Microinstruction format (48 bits)	27
12. Bus control mnemonics	29
13. Simplified bus control logic diagram	29
14. Bus control timing	30
15. Function generation timing	32
16. Details of loop counter hardware	33

ILLUSTRATIONS (Continued)

Figure	Page
17. K bus wiring for logic encoding.....	35
18. Bits patterns required for K bus constants and masks generation	36
19. Definition of image and reference arrays	38
20. Video IR image of the side view of an M48 tank	40
21. Image and reference arrays used for testing (See Equation 6.1) ..	41
22. Unnormalized correlation (See Equation 6.3).....	42
23. Correlation normalized by norms	43
24. Correlation normalized by means (See Equation 6.4)	44
25. Sum of absolute differences template matching (See Equation 6.5)	45
26. Sum of square differences template matching	46
27. List of variable names and definitions	48
28. Flow chart of scan loop	49
29. Flow chart of track loop	50
30. Iterative GAIN and BIAS flow chart	52
31. Plain paper tracking tests	53
32. Results from automatic GAIN and BIAS adjustments	54

ILLUSTRATIONS (Continued)

Figure	Page
33. Results of real time scan and track loop tests on a simulated M48 tank target	55
34. Memory reference instructions	57
35. Register operation instructions	58
36. Skip instructions	59
37. Other instructions	60
38. Conditional branches used to select microcode sequence	63
39. Subarray inside larger array	65
40. Addressing technique for image processing	66
41. Flow chart for routine WIN	68
42. Flow chart for routine CROSS	69
43. Flow chart for routine FRM	69
44. Flow chart for routine WAT	69
45. Flow chart for routine IMG	70
46. Flow chart for routine COR	71
47. Variable definition for routine MIN	73
48. Flow chart for routine MIN	74

ILLUSTRATIONS (Continued)

Figure	Page
49. Flow chart for routine REF	75
50. Flow chart for routine MNV	77
51. Flow chart for routine STD	78
52. Estimated execution times of tracker routines	79
53. Graphic illustration of DIP area needed for tracker	80
54. Graphic illustration of DIP chip area needed for tracker	81
55. Part inventory and power consumption	83

ACKNOWLEDGMENT

The development of the tracker contained within this report could not have been achieved without the contributions of Allen Davis and Roy Lee of the Redstone Arsenal Support Agency and Richard Sims of the Technology Laboratory. Allen Davis was responsible for much of the detailed logic design in the tracker and frequently offered significant improvements to designs by the author. Allen Davis and Roy Lee had total responsibility for the construction and debugging of all hardware. Their high motivation and truly outstanding work are largely responsible for the success of the tracker. It should also be acknowledged that Richard Sims wrote and debugged most of the utility software in Section 5, and assisted in the debugging of the emulation code in Section 7.

1. INTRODUCTION

A. Background

High speed microprocessors such as the Intel 3000 bit slice microprocessor (or AMD 2901) offer the highest speeds, the most flexibility, and the greatest computing power available (excluding perhaps the Motorola 10800 ECL microprocessor) in current microprocessor technology. The high speed microprocessors are directly TTL compatible and as such can interface to TTL LSI parts. The processors can achieve from 6-10 million operations per second and are certain to have wide ranging effects on military hardware.

Though slower speed microprocessors such as the Intel 8080 and Motorola 6800 have been applied in limited ways in Army seekers, little has been done to exploit the capability of high speed bipolar microprocessors.

This report details an application of a high speed microprocessor (Intel 3000) to imaging seekers. The work clearly demonstrates that software as opposed to hardware can be used as the basis for a real time imaging tracker. In particular, two dimensional correlation and template matching normally implemented in dedicated hardware can be accomplished in real time software. The bulk of this report contains a general hardware description and detailed description of correlator software. Software for an optical

contrast tracker appears in a separate report. Both trackers operate with the same hardware and the only difference between the two trackers is software changes. Thus, the software approach yields a tracker that is capable of radical changes in tracking mode. The changes can be effected merely by branching to different software. The implication is that trackers can be built that modify their tracking algorithms in response to countermeasures or battlefield conditions. Such changes could even occur after missile launch up to the time of missile and target impact. This capability has the effect of increased countermeasure immunity.

The bit slice, microprogrammed architecture of the Intel 3000 and other similar microprocessors appears to be well suited to many military applications. This follows because of the following reasons: 1) Architecture may be tailored to fit the application. 2) Microword structure can be configured to control complex hardware in parallel with microprocessor or other peripheral hardware functions. 3) The device operates at very high speeds.

In light of this report the development of future trackers will depend heavily on high speed software and hardware control. This follows since high speed software can replace much hardware. Thus, very complex tracking algorithms not practical in the past because of size and complexity are now feasible.

The following sections detail the design of a microprocessor applied to the image processing problem. The hardware structure and software techniques have general utility and can be used with any microprogrammed microprocessor. The tracking algorithms developed concentrate on correlation and template matching techniques, but the hardware with minimal or no modification can be applied to other tracking algorithms. Though the tracking algorithms were developed for the visible spectrum, they have direct application to other wavelengths.

B. Summary of Accomplishments

Following sections detail the design and construction of an imaging tracker built inhouse. The program was very successful and nearly all of the objectives in the original scope of work were achieved. The program's major accomplishments are listed as follows: 1) Designed, built, and tested the first Army imaging tracker capable of correlation and adaptive gate optical contrast tracking. 2) The feasibility of trackers using high speed microprocessors and CCD TV technology was demonstrated. 3) Several correlation and template matching techniques were studied and implemented in real time microprocessor software.

The work documented in this report is by no means complete and is

expected to continue into FY78. The emphasis, however, will shift to trackers using classical pattern recognition concepts involving feature extraction and classification.

2. HARDWARE DESCRIPTION OF THE COMPUTER INTERFACE TO THE TRACKER

A. General

This section is designed to present a general description of hardware used in the interface of the tracker computer to an existing lab minicomputer. The level of detail is sufficient for a complete understanding of the software presented in later sections. A more detailed description will be given in a report now in preparation.

B. Computer Interface Details

In order to develop software for a microprocessor a means must be generated to load software into its memories. By far the most convenient method is to load memory from cards or disks using an external computer. The hardware in this system was interfaced to an existing EAI 640 minicomputer. The interface is an important part of the system since it allows tracker software development to occur with minimum turn around. Figure 1 shows a block diagram of the microprocessor memory interface to the EAI

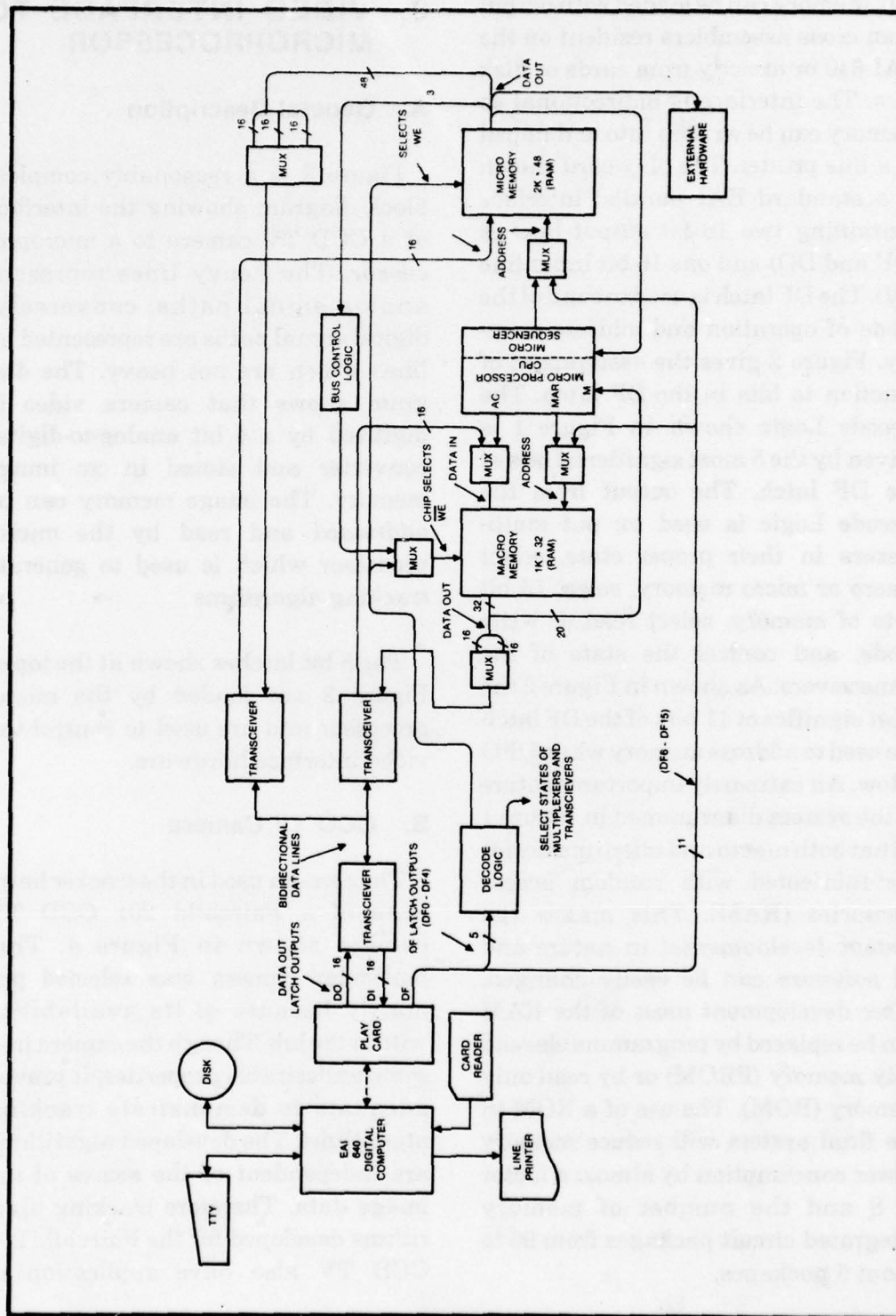


Figure 1. Memory block diagram.

640. Memory can be loaded with output from cross assemblers resident on the EAI 640 or directly from cards or disk files. The interface is bidirectional so memory can be written into or dumped to a line printer. The play card shown is a standard EAI parallel interface containing two 16 bit output latches (DF and DO) and one 16 bit input line (DI). The DF latch is used to control the mode of operation and address memory. Figure 2 gives the assignment of function to bits in the DF latch. The Decode Logic shown in Figure 1 is driven by the 5 most significant bits of the DF latch. The output from the Decode Logic is used to: put multiplexers in their proper state, select macro or micro memory, select 16 bit byte of memory, select read or write mode, and control the state of the transceivers. As shown in Figure 2 the least significant 11 bits of the DF latch are used to address memory when DFO is low. An extremely important feature of the system diagrammed in Figure 1 is that both macro and micro memories are fabricated with random access memories (RAM). This makes the system developmental in nature and all software can be easily changed. After development most of the RAM can be replaced by programmable read only memory (PROM) or by read only memory (ROM). The use of a ROM in the final system will reduce memory power consumption by almost a factor of 8 and the number of memory integrated circuit packages from 96 to about 6 packages.

3. VIDEO INTERFACE TO MICROPROCESSOR

A. General Description

Figure 3 is a reasonably complete block diagram showing the interface of a CCD TV camera to a microprocessor. The heavy lines represent analog signal paths; conversely, digital signal paths are represented by lines which are not heavy. The diagram shows that camera video is digitized by a 4 bit analog-to-digital converter and stored in an image memory. The image memory can be addressed and read by the microprocessor which is used to generate tracking algorithms.

The 8 bit latches shown at the top of Figure 3 are loaded by the microprocessor and are used to control the video interface hardware.

B. CCD TV Camera

The camera used in the tracker hardware is a Fairchild 201 CCD TV camera shown in Figure 4. This particular camera was selected primarily because of its availability within the lab. Though the camera has some undesirable properties, it proved adequate to demonstrate tracking algorithms. The developed algorithms are independent of the source of the image data. Therefore tracking algorithms developed for the Fairchild 201 CCD TV also have application to

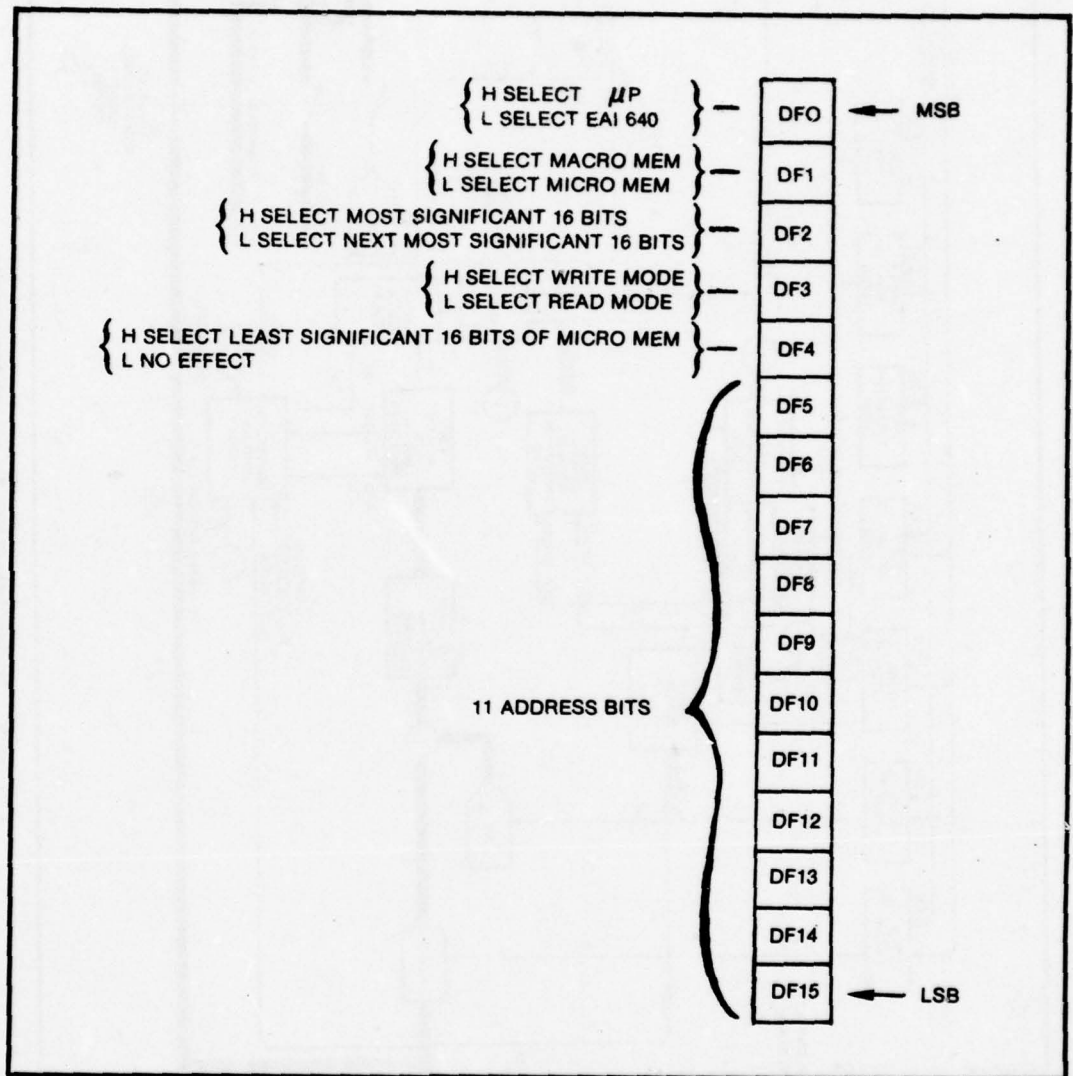


Figure 2. Function of bits in DF latch.

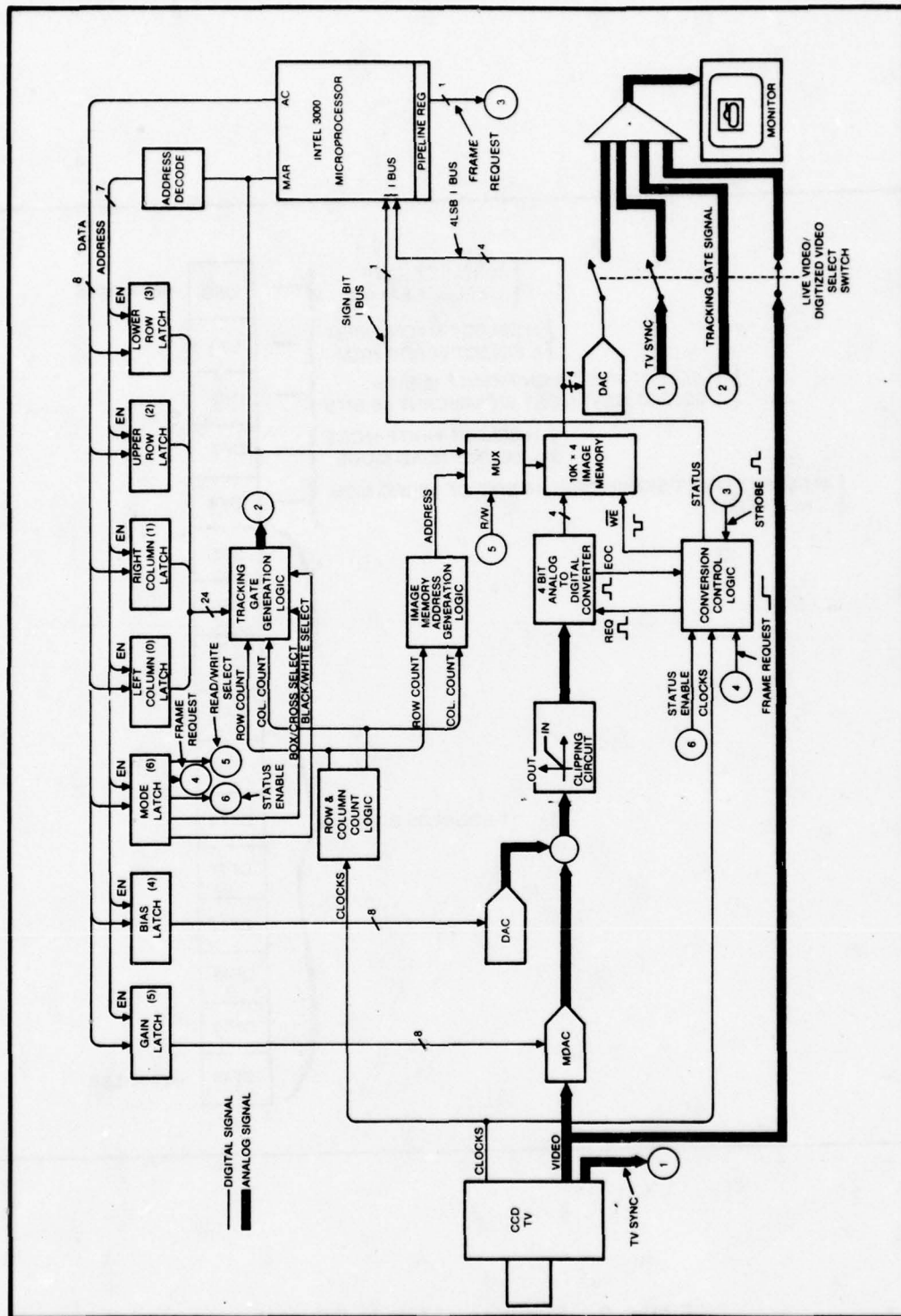


Figure 3. Video Interface to microprocessor.

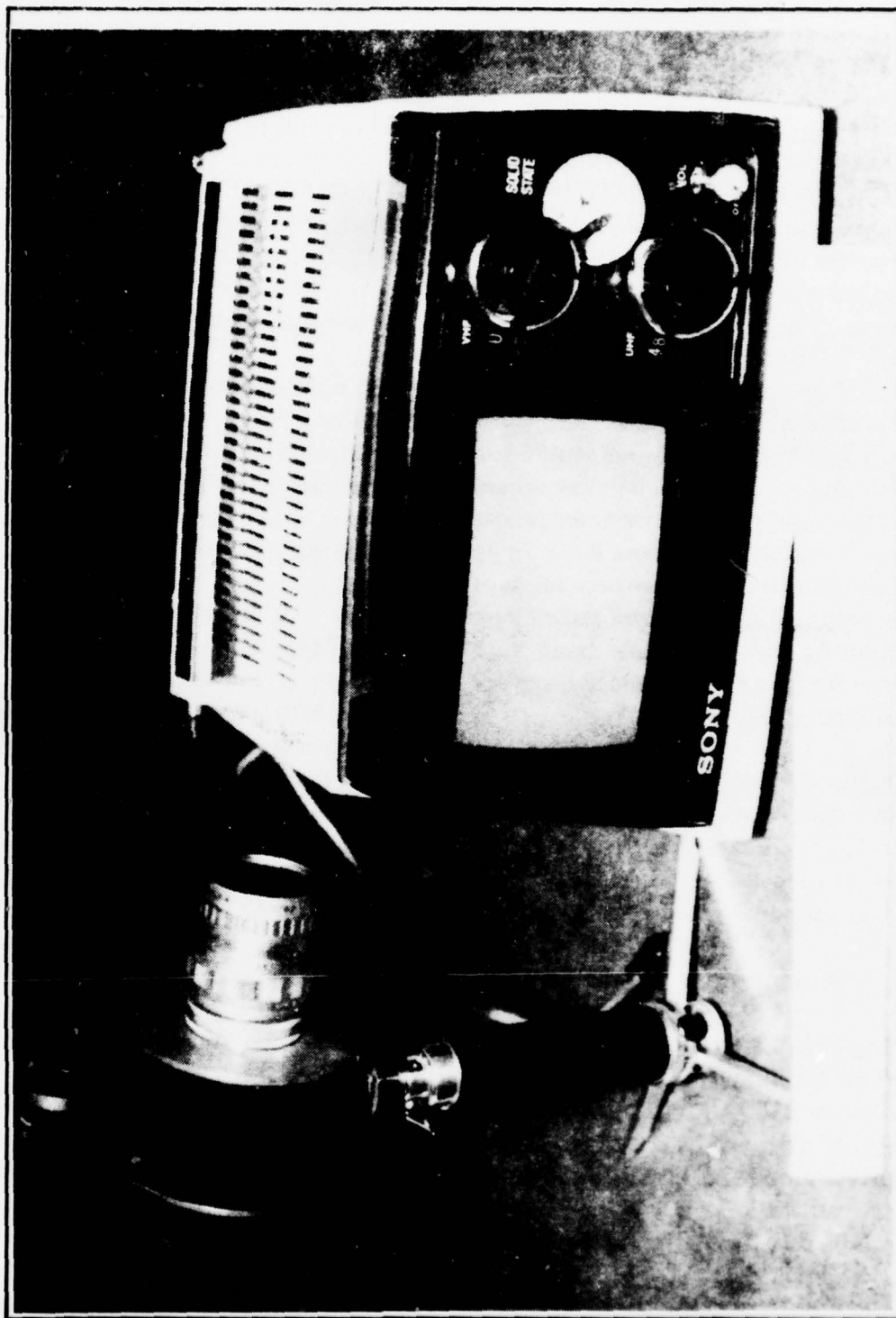


Figure 4. Fairchild 201 CCD TV.

imaging cameras operating in the 3-5 micro meter and 8-14 micro meter bands.

The Fairchild camera operates in the visible spectrum at a frame rate of 123 frames/sec with a resolution 100 x 100 elements. The camera uses 2:1 interlace and has non-standard composite video output. The non-standard video required a modified TV monitor to view the TV image, and made recording of the video difficult with standard video recorders. Video recordings could have been made with specialized recorders but the high cost of the units forced the use of available recorders. Recordings of marginal quality have been made of test results with an Ampex recorder by adjusting the recording head rate servo for a fairly stable lock on every 4th sync pulse.

Some problems were also experienced from: the tendency of the camera to bloom on bright objects, nonuniformity in response (pixel to pixel), and camera produced AGC.

C. Video Interface Control Latches

The GAIN latch, Figure 3, is used to control a multiplying digital to analog converter (MDAC). The MDAC in turn controls the gain of the video amplifier. The gain is selectable in the range from 0.0 to 5.0. Higher gains are possible but usable gains depend on the responsivity variation from pixel to pixel and fixed pattern noise. For the

CCD 201 solid state image sensor the photo response nonuniformity is on the order of +15% and higher gains were not indicated.

The BIAS latch, Figure 3, is used to drive a digital to analog converter (DAC). In this case the DAC output is used to "buck out " DC components appearing in the amplified video.

By manipulating the GAIN and BIAS latches it is possible to optimize the video going to the 4 bit analog to digital converter. The clipping circuit shown is used to protect the A/D from over range voltages.

The MODE latch, Figure 3, contains 5 bits of data which are used to control: the state of the address multiplexer (MUX shown in Figure 3), the gate color (black or white), the gate type (box or crosshairs), enable status, and frame requests. Figure 5 shows the bit assignments in the MODE latch. When the microprocessor is reading data from the image memory, the MUX is used to select the memory address register of the microprocessor as the source for image memory address. When status is enabled, the conversion control logic will force the sign bit of the I bus high after a new frame has been stored since the strobe in the pipeline register is enabled in the control logic.

The LEFT COLUMN latch, Figure 3, defines the column location (1 to 100)

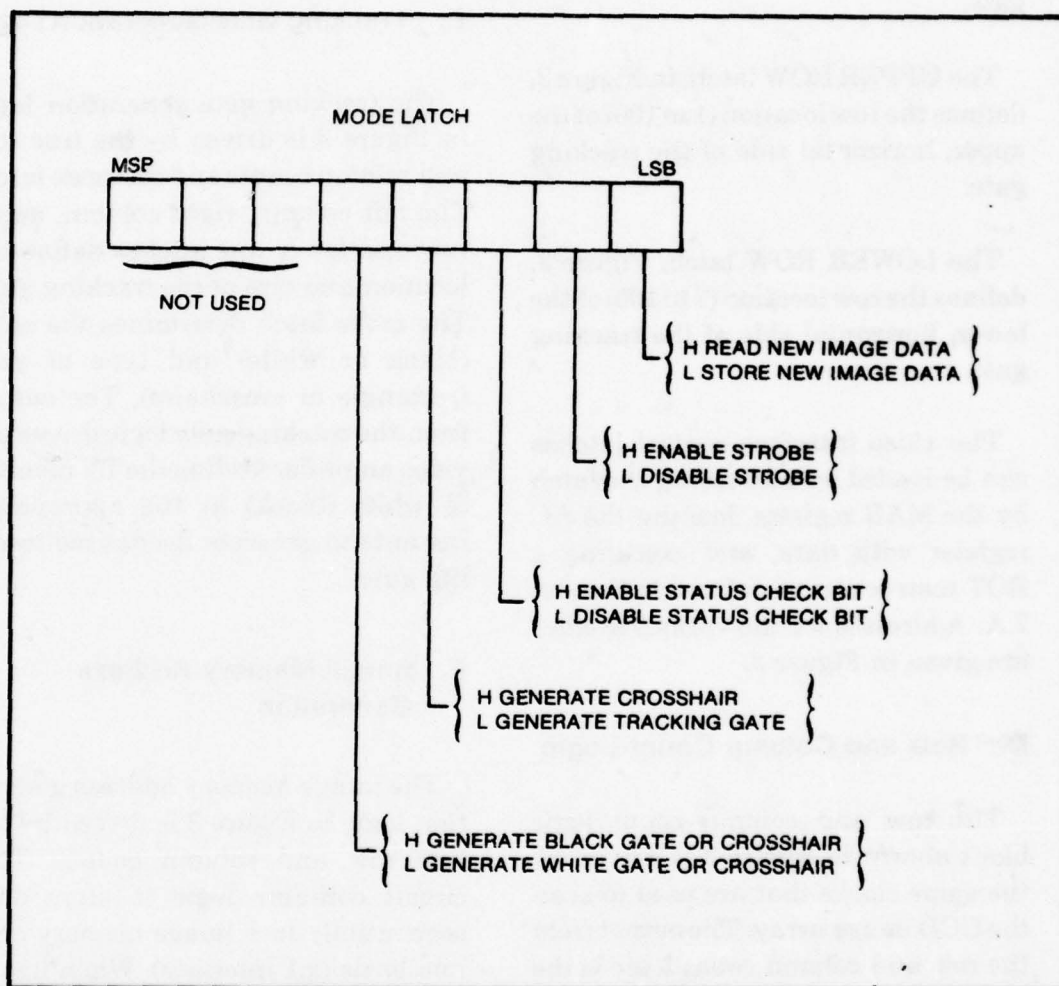


Figure 5. Bit assignments in the MODE latch.

of the left vertical side of the tracking gate.

The RIGHT COLUMN latch, Figure 3, defines the column location (1 to 100) of the right vertical side of the tracking gate.

The UPPER ROW latch, in Figure 3, defines the row location (1 to 100) of the upper, horizontal side of the tracking gate.

The LOWER ROW latch, Figure 3, defines the row location (1 to 100) of the lower, horizontal side of the tracking gate.

The video interface control latches can be loaded by addressing the latch by the MAR register, loading the AC register with data, and executing a ROT instruction as defined in Section 7.A. Addresses for the various latches are given in Figure 3.

D. Row and Column Count Logic

The row and column count logic block shown in Figure 3 is driven by the same clocks that are used to scan the CCD image array. The output from the row and column count logic is the true row and column location of the pixel being output from the TV camera. Since the camera uses 2:1 interlace, logic decoding and counters are necessary to produce the true row and

column count. The row count ranges from 1 to 100 starting in the uppermost row. The column count also ranges from 1 to 100 but starts in the left most column.

E. Tracking Gate Generation Logic

The tracking gate generation logic in Figure 3 is driven by the true row and column counts and the mode latch. The left column, right column, upper row and lower row latches define the location and size of the tracking gate. The mode latch determines the color (black or white) and type of gate (rectangle or crosshairs). The output from the tracking gate logic drives the video amplifier feeding the TV monitor to white (black) at the appropriate instants to generate the desired tracking gate.

F. Image Memory Address Generation

The image memory address generation logic in Figure 3 is driven by the true row and column counts. This circuit contains logic to store data sequentially into image memory on a line basis (2:1 interlace). When image data is stored in this manner addressing of image data by the microprocessor becomes an easier task and need not take into account the effect of 2:1 interlace.

G. Conversion Control Logic

The conversion control logic in Figure 3 is driven by the same clocks used to scan the CCD array. When the frame signal is high, any strobes generated by the microprocessor will result in a new TV frame being stored into image memory, and the conversion control logic issues convert requests (REQ) to the analog to digital converter. The converter in turn issues an end of conversion signal (EOC) for each request (REQ). When data has stabilized and an EOC has been issued, data is written into image memory by bringing the memory write enable low (WE).

Since the conversion control logic uses the clocks which are also used to scan the CCD array, conversions take place as pixel data is shifted from the CCD and the row and column address of each new pixel is known for each conversion that takes place.

H. Image Memory

The image memory can store $10K \times 4$ bits of image data (all 100×100 elements of the array). The large memory is not required for tracker operation but is convenient for test purposes. Most correlation trackers would require no more than $32 \times 32 = 1024$ memory locations. The output of the memory is placed on the I bus of the

microprocessor, but the data is meaningful only if addresses are being supplied from the microprocessor MAR bus. When a frame is being stored in image memory, the addresses are generated by the image memory address generation logic.

I. TV Monitor

The TV monitor shown in Figure 6 is used to monitor video and also the location and size of the tracking gate. In normal operation video is supplied directly to the monitor summed with the tracking gate signal. However, the monitor can also be used to display the stored image in image memory. When data is not being written into image memory, image data is available on the output of image memory. The output is used to drive a digital to analog converter (DAC). The output from the DAC is summed with TV sync and the tracking gate signal generating a composite video signal for the monitor. The image memory, address generation logic supplies addresses to image memory in synchronization with the clocks used scan to the CCD image array. The propagation delay from clocks to DAC output causes the video to be shifted relative to real time video, but still produces a very useful display. For example, the display can be used to determine the effects of preprocessing (gain and bias latches) on the video from the camera.

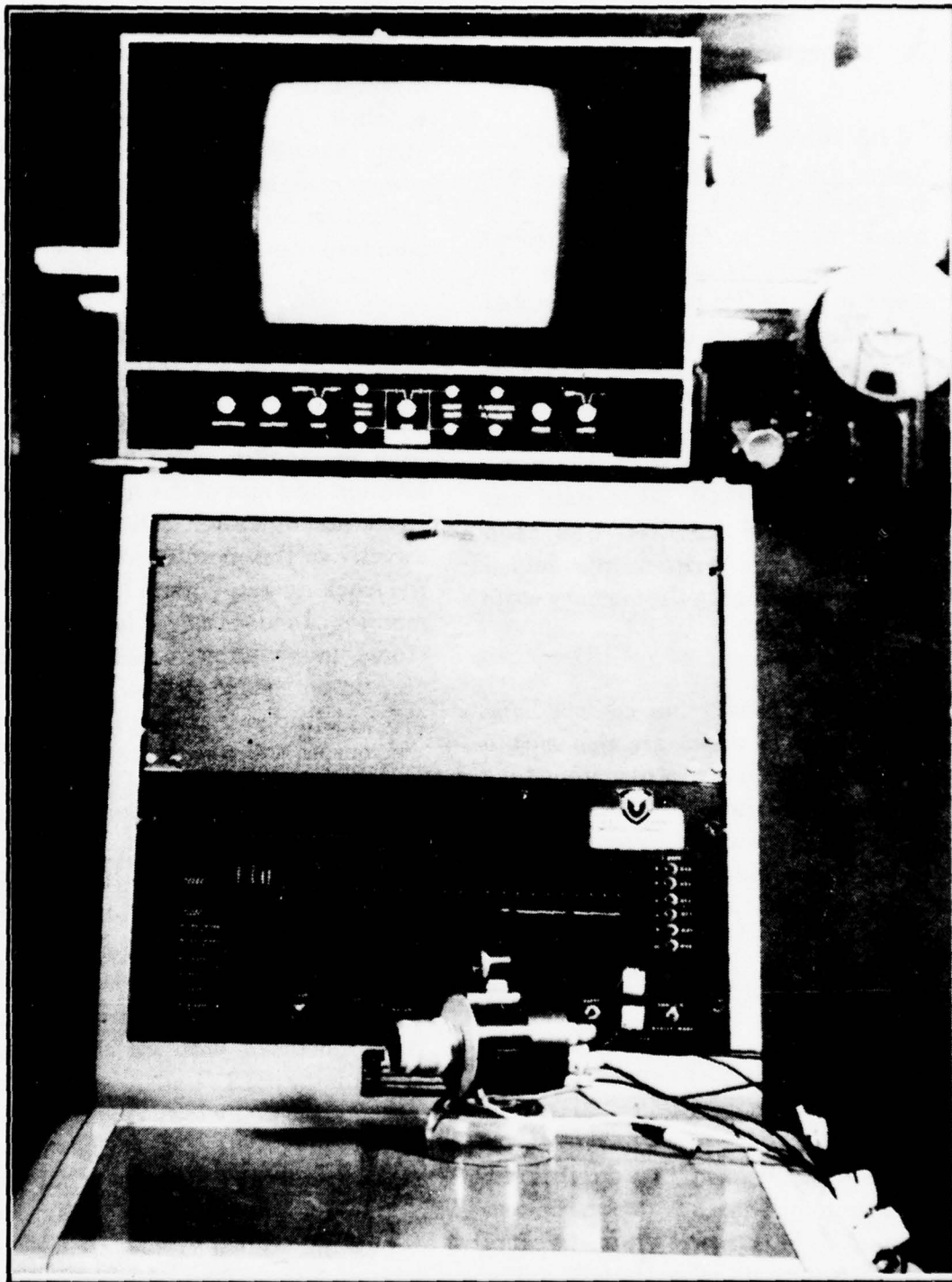


Figure 6. Photograph of tracker hardware.

4. MICROPROCESSOR AND TRACKER HARDWARE

A. Physical Description

Figure 6 is a photograph of the completed tracker and Figure 7 is a close up of the operator's panel. The thumb wheel switches shown in the figure drive address halt hardware. An address halt can be executed at any macroinstruction at any point within its microcode by setting addresses into the thumb wheel switches.

The selector switch shown at the left side of the panel drives multiplexer hardware which can display virtually any bus within the system. The column of switches to the right of the panel are software sense switches used for force branches in software.

The white switches to the lower right are the run (start the clock) and single switches (one clock pulse per depression). Since all memory can be read, modified and written into from the minicomputer there is no need to provide for these functions through front panel switches. The toggle switch located at the center, bottom portion of the panel selects one of two possible sources of video for the TV monitor. Video can be displayed directly from the camera or video generated from stored image data can be displayed. The toggle switch to the far right, bottom of the panel is used to take control of the data bus from the

minicomputer. Finally, the small momentary push button switch located to the left of the run and single switches is used to reset hardware. Figures 8 and 9 show all of the circuit boards used in the tracker. Multiplexer boards are not shown. Some of the components on the memory boards are taken up by minicomputer interface circuit components. It should be noted that no attempt was made to minimize package size. Many improvements could be made along these lines. As an example the 2K x 48 bit RAM micro memory could be implemented with 1024 x 8 PROMs reducing the package count from 96 to 6 and the power requirements by a factor of 8.

B. Microprocessor and Tracker Architecture

The microprocessor shown in Figure 3 is shown in more detail in Figure 10. The microprocessor is the Intel 3000 used in a standard pipelined configuration. The loop counters shown are used extensively in image processing. The counters can be addressed and loaded, or addressed and incremented and tested on a single microcycle. The counters are used primarily to keep track of loop counts in the nested loop computations that occur in image processing (e.g. convolution and correlation computations).

Since counting, loading or testing can occur in parallel with microprocessor functions the overhead

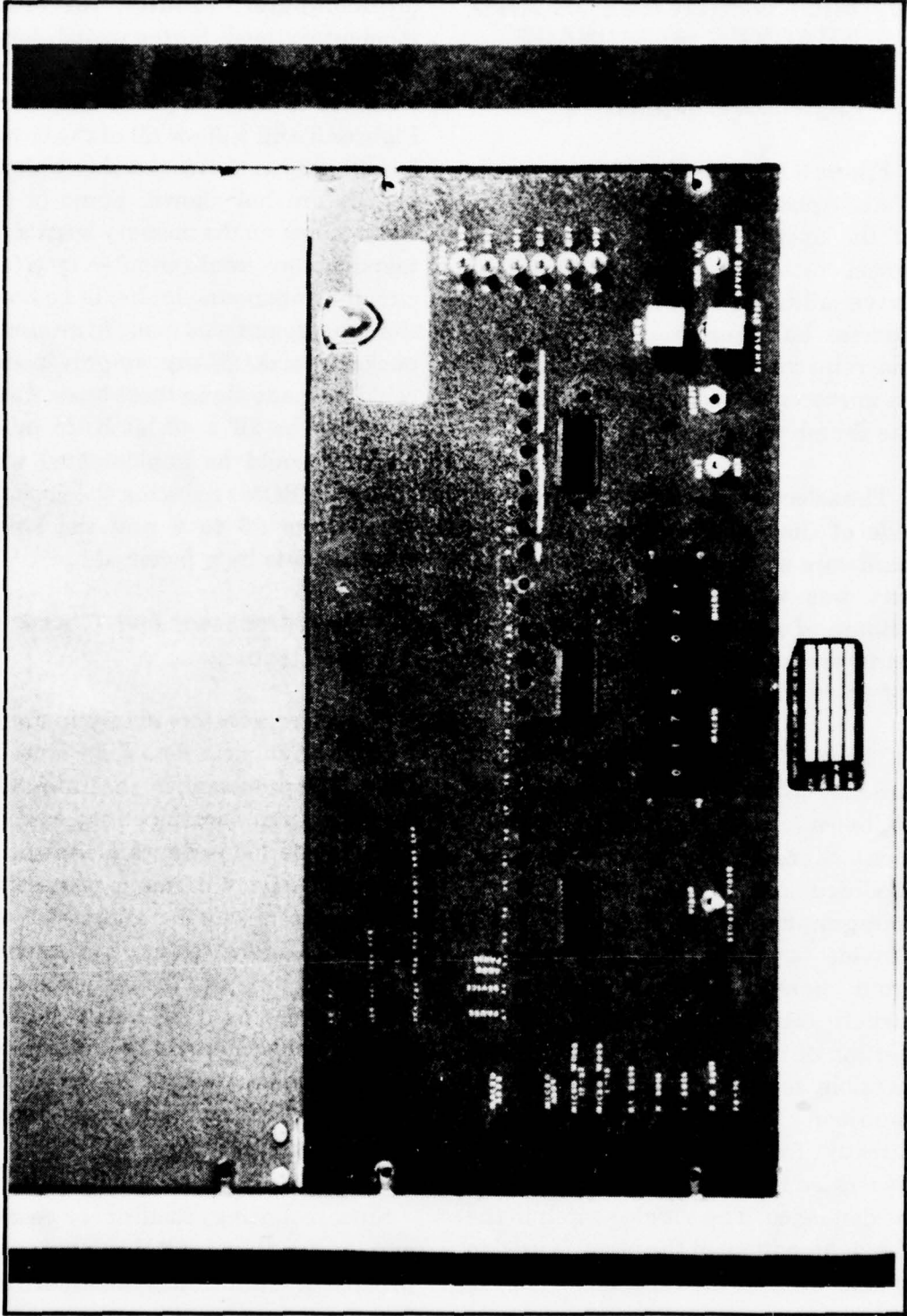
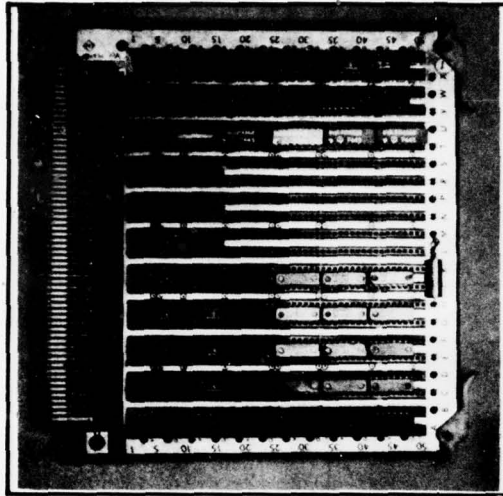
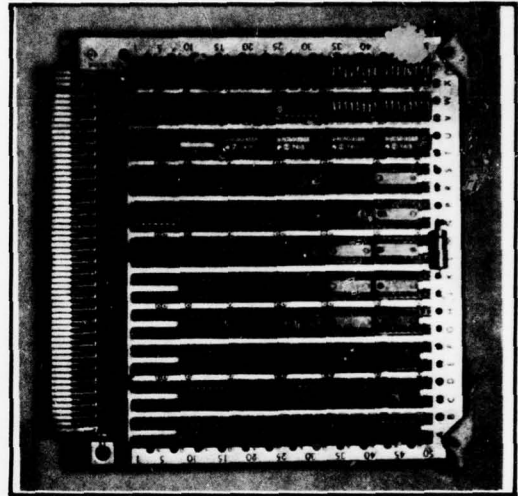


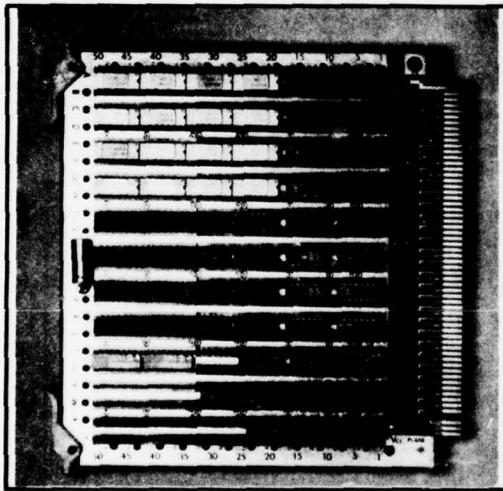
Figure 7. Photograph of operators panel.



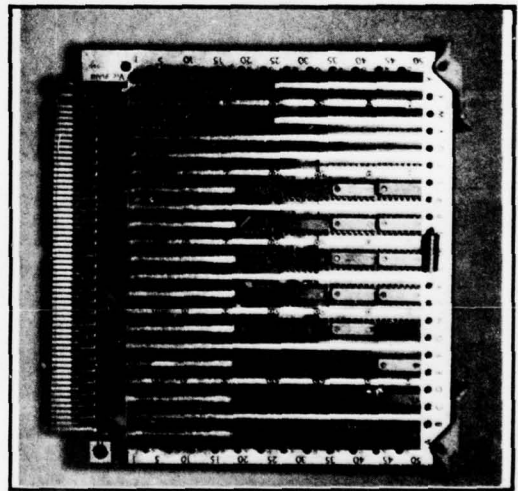
20 x 1K MACRO MEMORY



32 x 1K MICRO MEMORY

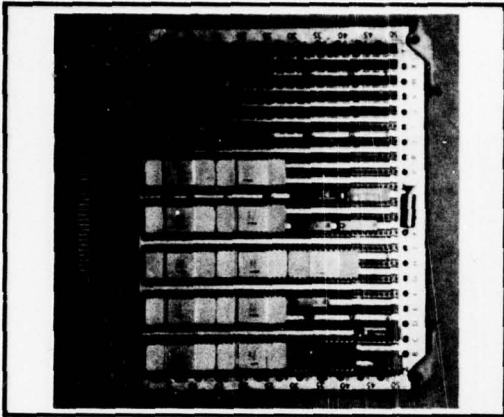


16 x 2K MACRO MEMORY

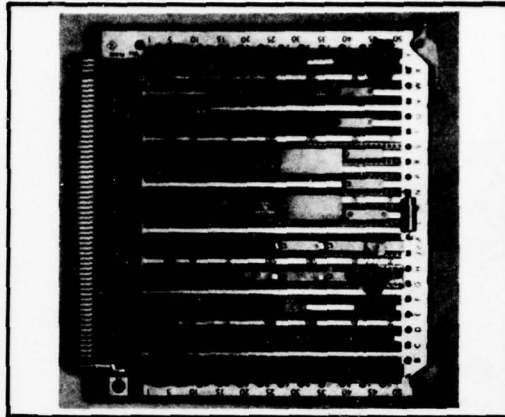


32 x 1K MICRO MEMORY

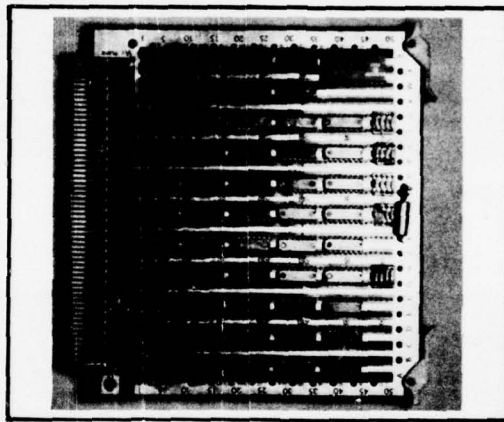
Figure 8. Macro and Micro memory boards.



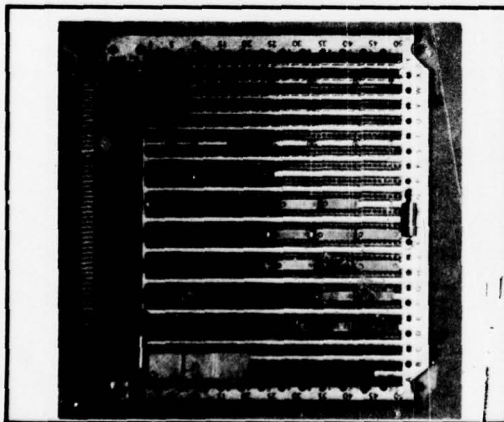
INTEL 3001 CPU



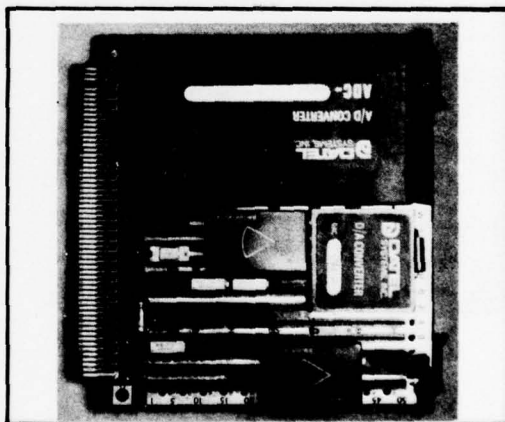
VIDEO CONTROL



10K x 4 IMAGE MEMORY



INTEL 3002



VIDEO FRONT END

Figure 9. Microprocessor, image memory, and video front end boards.

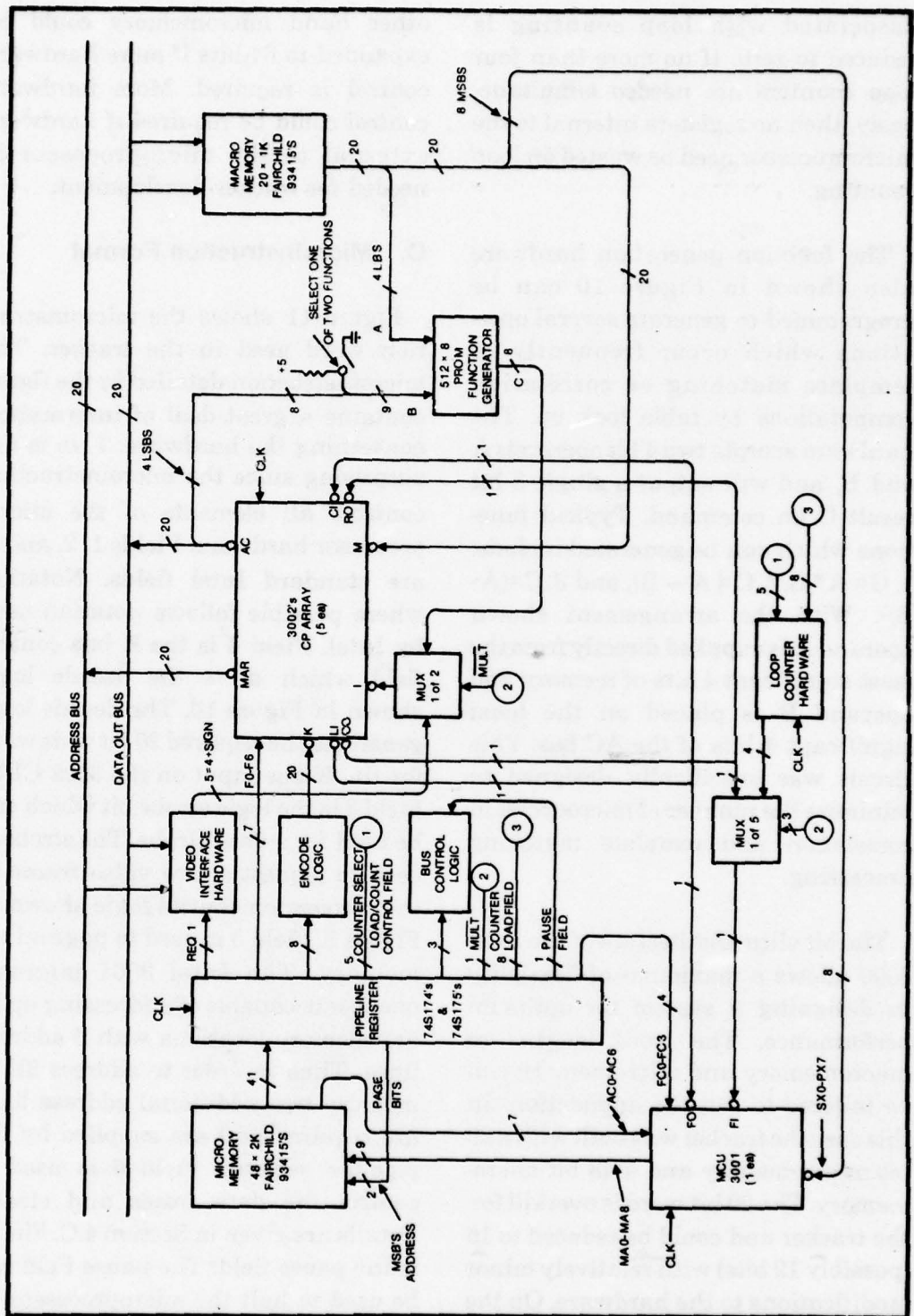


Figure 10. Block diagram of microprocessor and tracker architecture.

associated with loop counting is reduced to zero. If no more than four loop counters are needed simultaneously, then no registers internal to the microprocessor need be wasted on loop counting.

The function generation hardware also shown in Figure 10 can be programmed to generate several operations which occur frequently in template matching or correlation computations by table look up. The hardware accepts two 4 bit operands A and B, and will output a single 8 bit result C on command. Typical functions which can be generated include: 1. $C = A * B$, 2. $C = |A - B|$, and 3. $C = (A - B)^2$. With the arrangement shown operand A is supplied directly from the least significant 4 bits of memory and operand B is placed on the least significant 4 bits of the AC bus. This circuit was specifically designed to minimize the number of microcycles in correlation and template matching processing.

The bit slice architecture of the Intel 3000 allows a maximum of flexibility in designing a system for optimum performance. The word lengths of macromemory and micromemory can be tailored to suit the application. In this case the tracker was built with a 20 bit macromemory and a 48 bit micromemory. The 20 bit word is overkill for the tracker and could be reduced to 16 (possibly 12 bits) with relatively minor modifications to the hardware. On the

other hand micromemory could be expanded to 64 bits if more hardware control is required. More hardware control could be required if hardware external to the microprocessor is needed for tracker development.

C. Microinstruction Format

Figure 11 shows the microinstruction word used in the tracker. The microinstruction detailed in the figure contains a great deal of information concerning the hardware. This is not surprising since the microinstruction controls all elements of the microprocessor hardware. Fields 1, 2, and 3 are standard Intel fields. Notation where possible follows notation used by Intel. Field 4 is the K bus control field which drive the decode logic shown in Figure 10. The decode logic generates the required 20 bit wide word for the K bus input on the 3002 CPU. Field 8 is the logic strobe bit which can be used for a data strobe. The strobe is used to request a new video frame in the conversion control logic shown in Figure 3. Field 5 is used to page micromemory. The Intel 3001 microsequencer is capable of addressing up to 512 memory locations with 9 address lines. Thus in order to address 2K of memory two additional address lines are required and are supplied by the pipeline register. Field 6 is used to control the data buses and clocks. Details are given in Section 4.C. Field 7 is the pause field. The pause field can be used to halt the microprocessor by

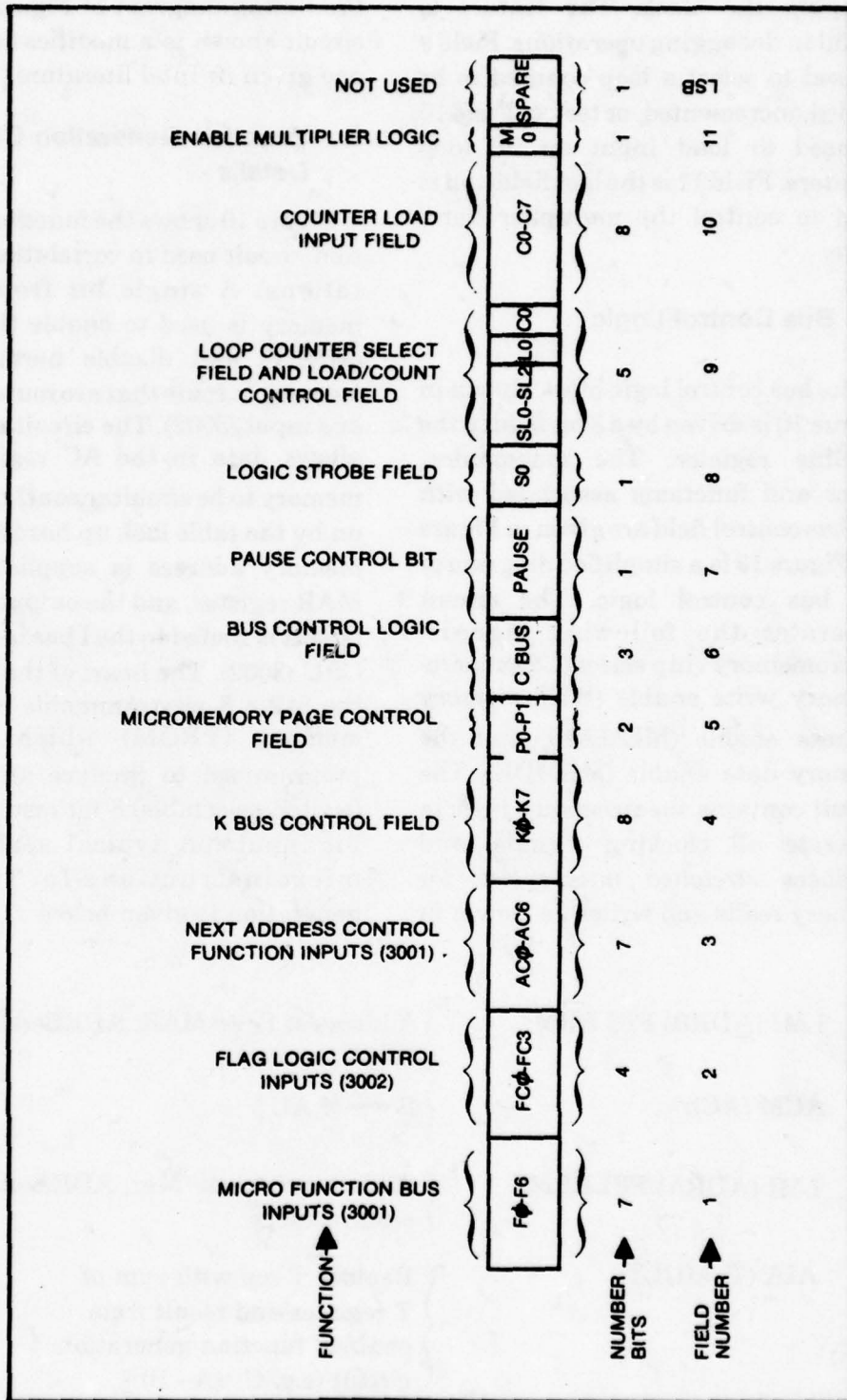


Figure 11. Microinstruction format (48 bits).

stopping the clock. The feature is useful in debugging operations. Field 9 is used to select a loop counted to be loaded, incremented, or tested. Field 10 is used to load input on all loop counters. Field 11 is the last field and is used to control the multiplier hardware.

D. Bus Control Logic

The bus control logic block shown in Figure 10 is driven by a 3 bit field in the pipeline register. The mnemonics, codes and functions associated with the bus control field are given in Figure 12. Figure 13 is a simplified diagram of the bus control logic. The circuit generates the following signals: macromemory chip select (CS), macromemory write enable (WE), memory address enable (MEMAE), and the memory data enable (MEMDE). The circuit contains the necessary logic to generate all clocking signals, and produces stretched microcycles for memory reads and writes as shown in

the timing diagram of Figure 14. The circuit shown is a modification of the one given in Intel literature [1].

E. Function Generation Circuit Details

Figure 10 shows the function generation circuit used in correlation computations. A single bit from micro-memory is used to enable the circuit (MULT) and disable normal video memory outputs that are routed to the I bus input (3002). The circuit as shown allows data in the AC register and memory to be simultaneously operated on by the table look up hardware. The memory address is supplied by the MAR register, and the output from the circuit is routed to the I bus input of the CPU (3002). The heart of the circuit is the 512×8 programmable read only memory (PROM) which can be programmed to produce any of two (switch selectable) 8 bit results from 4 bit inputs. A typical sequence of microinstructions for function generation is given below.

LMI (ADRB) FFI RRM;	{ Address of B \rightarrow MAR, ADRB=ADRB+1, { read - memory }
ACM (AC);	{ B \rightarrow AC }
LMI (ADRA) FFI RRM;	{ Address of A \rightarrow Mar, ADRA=ADRA+ { read memory }
AIA (T) MULT;	{ Replace T reg with sum of { T register and result from enabled function generation circuit (e.g. $C = (A - B)^2$ }

MNEMONIC	CODE	FUNCTION
NMO	000	NO BUS OPERATION
INH	001	INHIBIT CPU CLOCK
RMW	010	READ MODIFY WRITE
CNB	011	CPU NEEDS BUS
RIN	100	REQUEST INPUT FROM EXTERNAL DEVICE
ROT	101	REQUEST OUTPUT TO EXTERNAL DEVICE
RRM	110	REQUEST READ MEMORY
RWM	111	REQUEST WRITE MEMORY

Figure 12. Bus control mnemonics.

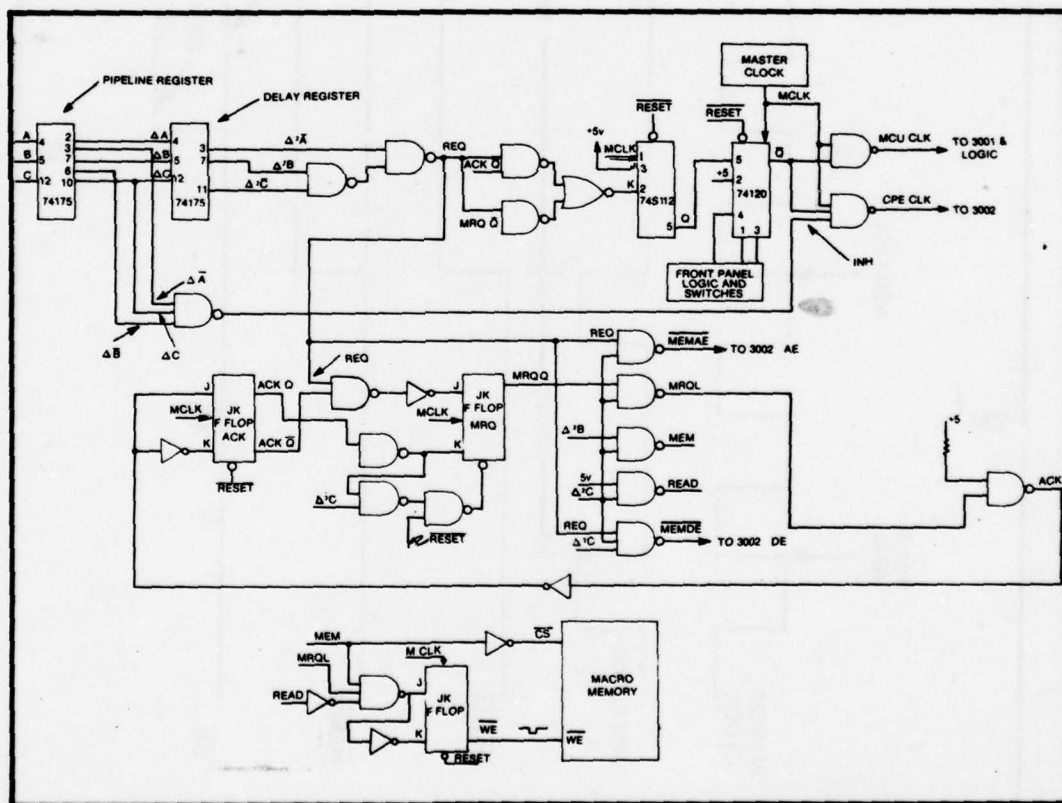


Figure 13. Simplified bus control logic diagram.

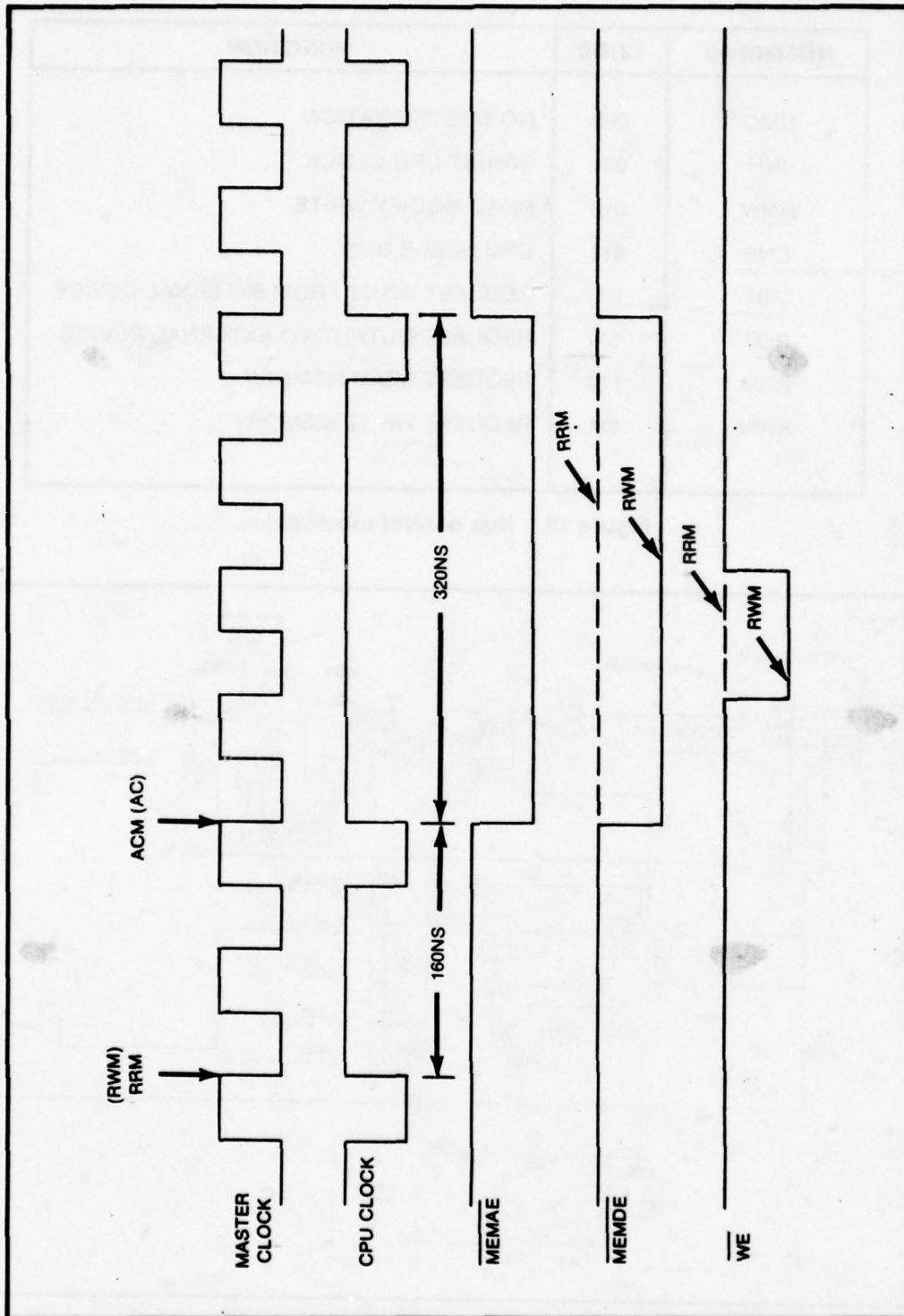


Figure 14. Bus control timing.

Hardware timing is shown in Figure 15 and relates directly to the example given above. The logic signal MEMAE is the macromemory address enable generated by the bus control logic, and the logic signal MULT is asserted (active low) by a bit in the microinstruction word (MULT). The stretched cycles for slow memory are evident in the diagram.

F. Loop Counting Circuit Details

Figure 16 shows the details of loop counting hardware. There are four counters shown, thus, loop counts for four computation loops may be maintained simultaneously. The counters have general utility and are especially useful when dealing with nested loops appearing in image processing. The counters may be loaded (or incremented and tested) simultaneously with CPU operations of the Intel

3002. The field mnemonics and associated hardware control word are shown in Figure 16. In practice a loop counter is initialized by loading a selected counter with the ones complemented of the desired loop count. The counter can be incremented and tested as required by testing for a carry from the counter. When counters are not being tested or loaded, the source of the Intel 3001 FI input is the carry out (CO) from the Intel 3002.

An example segment of code is given below showing how a counter is loaded and then tested by the JFL instruction of the Intel 3001.

The clock signals shown in Figure 16 are delayed $\frac{1}{4}$ of a microcycle from the MCU clock so that loading (or incrementation and testing) can be accomplished in a single microcycle.

```

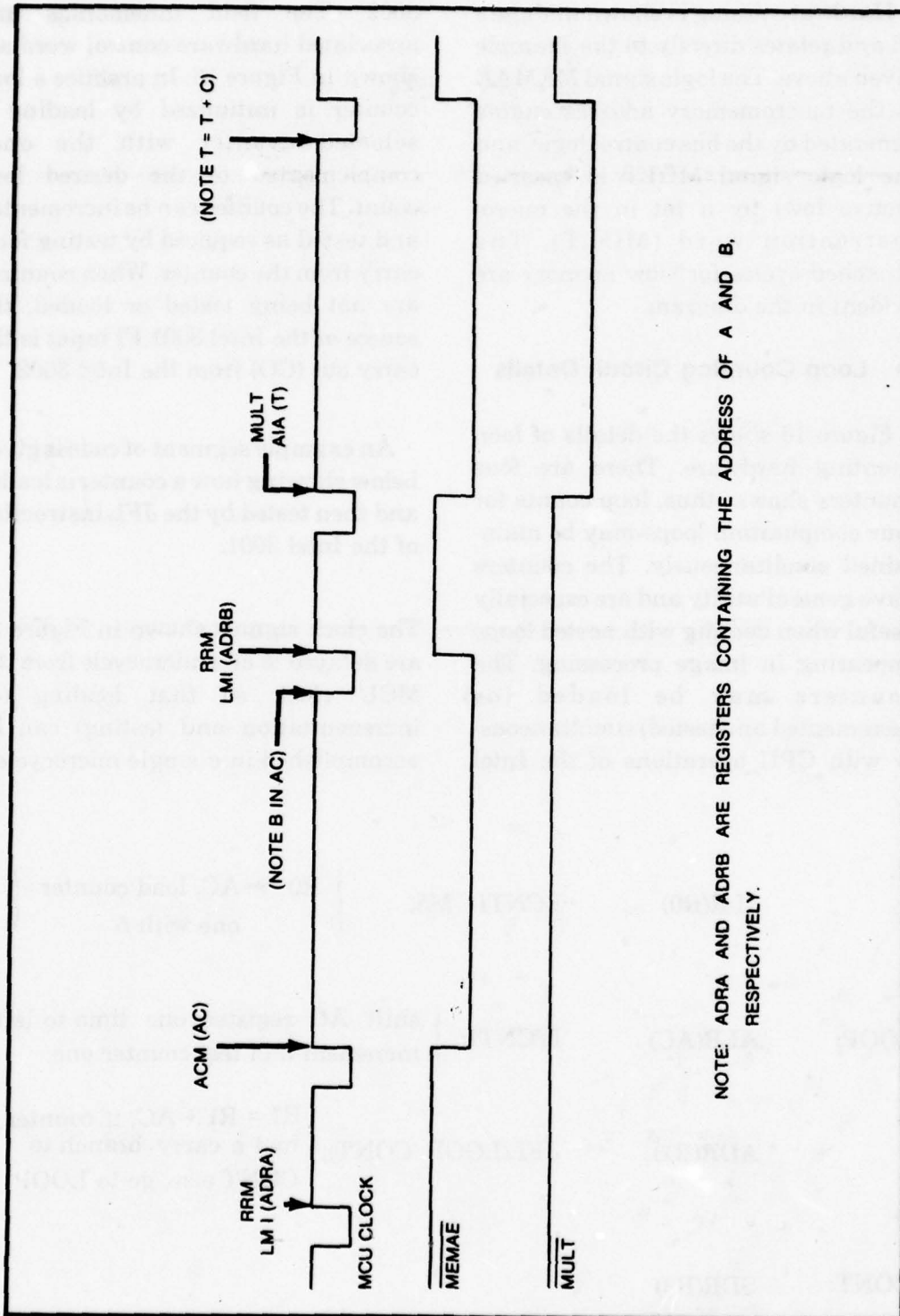
                ILR(R0)      LCNTI M5;      { R0 → AC, load counter
                                           one with 5 }

LOOP:          ALR(AC)      ITCNTI;      { shift AC register one time to left,
                                           increment and test counter one }

                ADR(R1)      JFL(LOOP, CONT); { R1 = R1 + AC, if counter
                                           had a carry branch to
                                           CONT else, go to LOOP }

CONT:          SDR(R3)

```



NOTE: ADRA AND ADRB ARE REGISTERS CONTAINING THE ADDRESS OF A AND B, RESPECTIVELY.

Figure 15. Function generation timing.

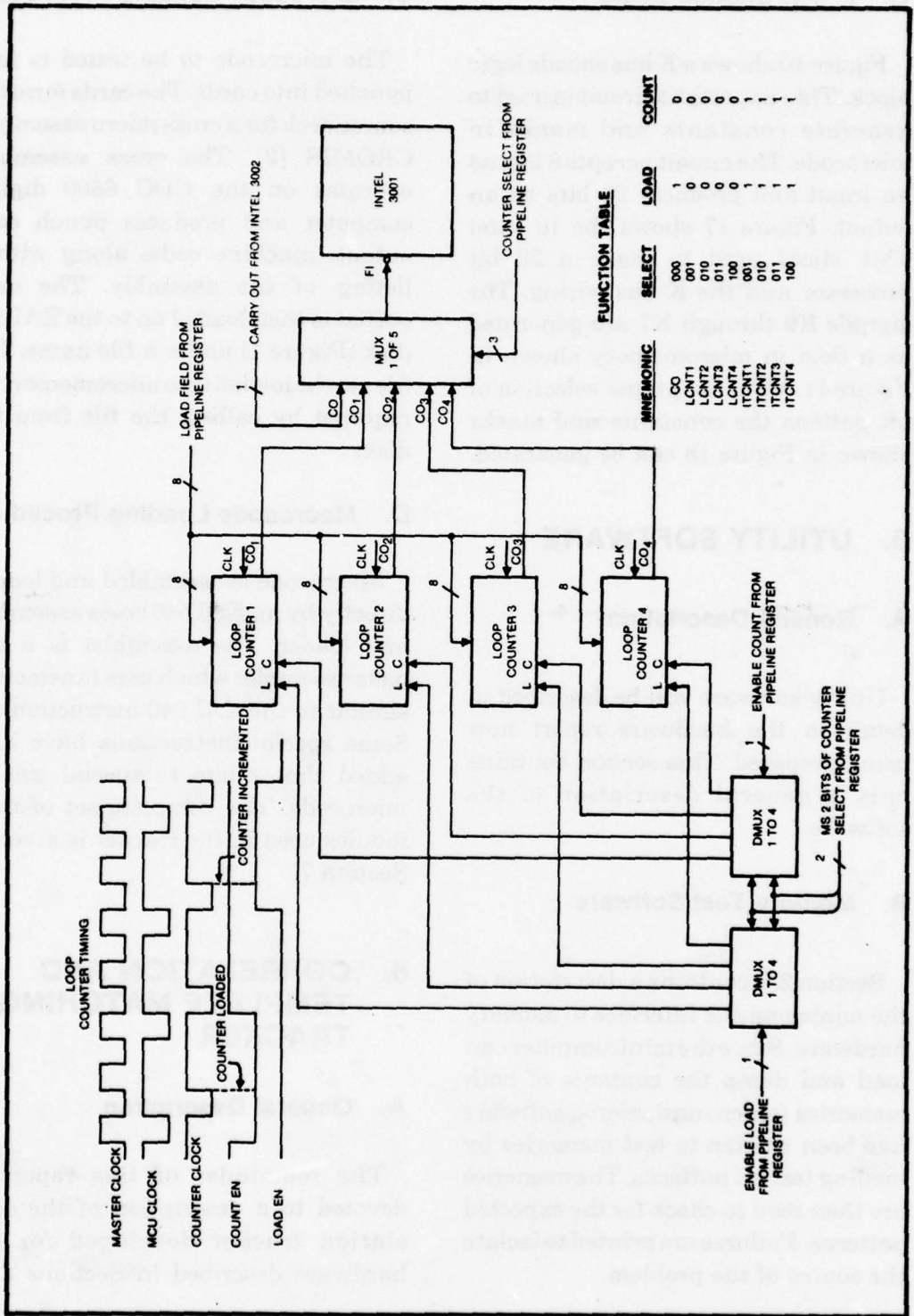


Figure 16. Details of loop counter hardware.

G. K Bus Encode Logic

Figure 10 shows a K bus encode logic block. This important circuit is used to generate constants and masks in microcode. The circuit accepts 8 bits as an input and produces 20 bits as an output. Figure 17 shows the 10 Intel 3001 slices used to make a 20 bit processor and the K bus wiring. The signals K0 through K7 are generated as a field in micromemory shown in Figure 11. With the proper selection of bit patterns the constants and masks shown in Figure 18 can be generated.

5. UTILITY SOFTWARE

A. General Description

Utility software will be described in detail in the hardware report now being prepared. This section contains only a general description of the software.

B. Memory Test Software

Section 2.B contains a description of the minicomputer interface to memory hardware. Since the minicomputer can load and dump the contents of both memories (macro and micro), software has been written to test memories by loading test bit patterns. The memories are then read to check for the expected patterns. Failures are printed to isolate the source of the problem.

C. Microcode Loading Procedure

The microcode to be tested is first punched into cards. The cards form the source deck for a cross micro assembler CROMIS [2]. The cross assembler operates on the CDC 6600 digital computer and produces punch card output (machine code) along with a listing of the assembly. The card output is then loaded on to the EAI 640 disk (Figure 1) under a file name. The file can be loaded into micromemory as required by calling the file from the disk.

D. Macrocode Loading Procedure

Macrocode is assembled and loaded directly by an EAI 640 cross assembler and loader. The assembler is a two pass assembler which uses mnemonics similar to the EAI 640 instruction set. Some special instructions have been added that relate to special tracker microcode. The complete set of mnemonics used in the tracker is given in Section 7.

6. CORRELATION AND TEMPLATE MATCHING TRACKER

A. General Description

The remainder of this report is devoted to a description of the correlation tracker developed for the hardware described in Sections 2, 3,

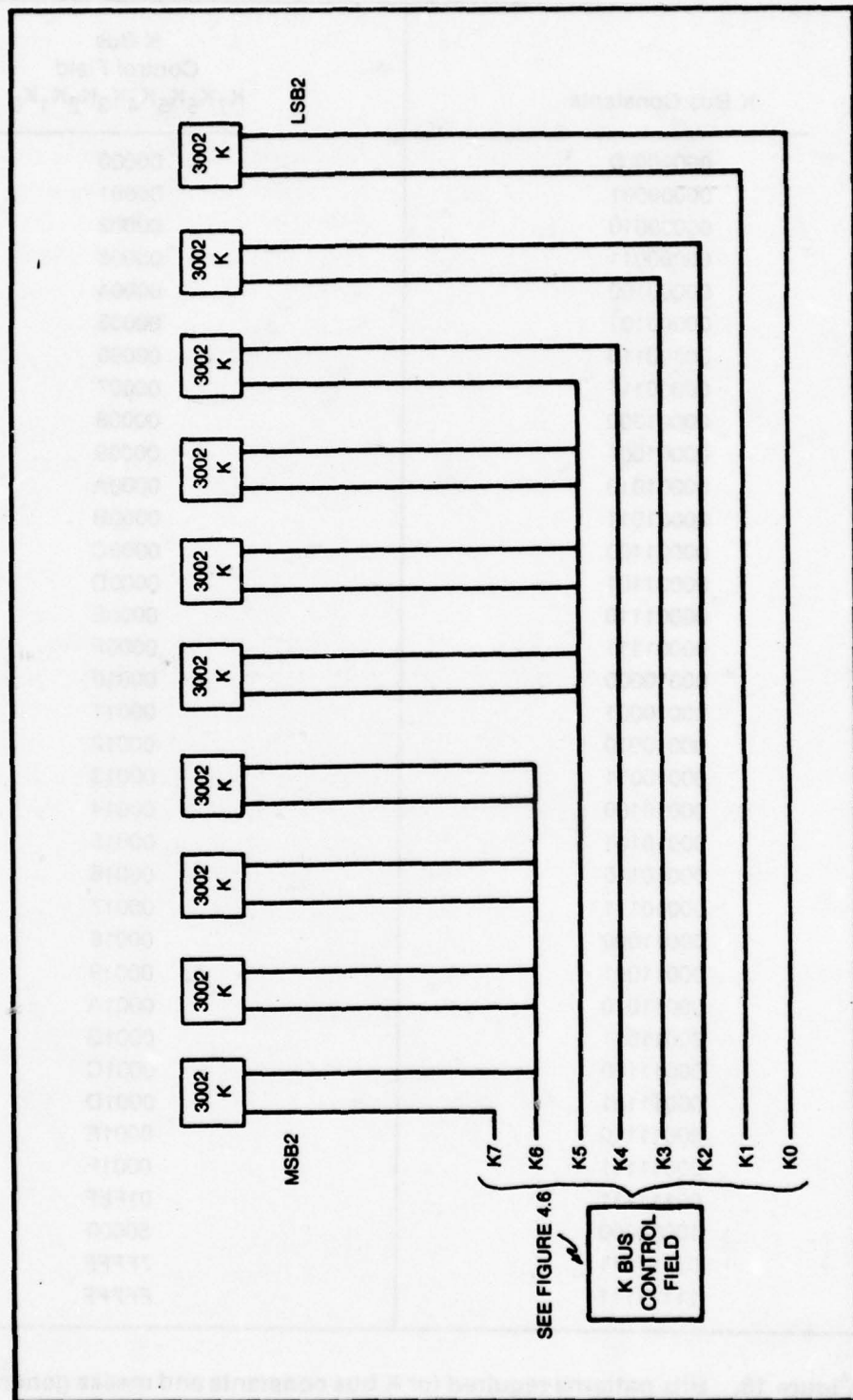


Figure 17. K bus wiring for logic encoding.

K Bus Constants	K Bus Control Field K ₇ K ₆ K ₅ K ₄ K ₃ K ₂ K ₁ K ₀
00000000	00000
00000001	00001
00000010	00002
00000011	00003
00000100	00004
00000101	00005
00000110	00006
00000111	00007
00001000	00008
00001001	00009
00001010	0000A
00001011	0000B
00001100	0000C
00001101	0000D
00001110	0000E
00001111	0000F
00010000	00010
00010001	00011
00010010	00012
00010011	00013
00010100	00014
00010101	00015
00010110	00016
00010111	00017
00011000	00018
00011001	00019
00011010	0001A
00011011	0001B
00011100	0001C
00011101	0001D
00011110	0001E
00011111	0001F
00111111	01FFF
10000000	80000
01111111	7FFFF
11111111	FFFFFF

Figure 18. Bits patterns required for K bus constants and masks generation.

and 4. The tracker is unique in that it operates in real time and uses almost 100% software to execute algorithms.

Appendix A contains a listing of the tracker macrocode. The macrocode contains EAI 640 instructions (emulation) along with some special instructions associated with the tracker. The EAI 640 instructions are given in Section 6.B. The special instructions, described in detail in Section 7, represent the bulk of the tracker software. The software described in these sections is written on the microcode level to achieve maximum efficiency. A complete listing can be found in Appendix B.

B. Correlation and Template Matching Algorithms

The complete tracker will be described in Section 6.C. This section identifies several correlation and template matching algorithms selected for study. Though several algorithms were tested, all operate on two image subarrays as defined in Figure 19. The 17×17 image array $I(I,J)$ is part of the 100×100 image produced by the camera. The smaller 9×9 reference array $R(I,J)$, bounded by the tracking gate, contains a high percentage of target pixels relative to background pixels.

The image array is always centered at the current track point (NX, NY) . The image array is read from the 10K

video memory 123 times a second during the vertical blanking interval. The reference array is obtained from the image array and stored separately when tracking is initiated. The reference array contains the "signature" of the target and is not changed or updated until correlation quality degrades. Correlation quality degrades typically due to range closure or target aspect changes. The correlation or template matching algorithm is used to locate the point of best match of the reference array and a 9×9 subarray of the image array. The point of best match is determined from the maximum or minimum of a function which in some way measures the quality of the match. Equations 6.1 through 6.3 define all the correlation algorithms or normalized correlation algorithms

$$\{C(M,\bar{N})\} = \sum_{i=1}^9 \sum_{j=1}^9 I(K,L) * R(I,J) \quad 6.1$$

$$\{C(M,N)\} = \frac{\sum_{I=1}^9 \sum_{J=1}^9 I(K,L) * R(I,J)}{\sum_{I=1}^9 \sum_{J=1}^9 I^2(K,L) \sum_{I=1}^9 \sum_{J=1}^9 R^2(I,J)} \quad 6.2$$

$$\{C(M,N)\} = \sum_{I=1}^9 \sum_{J=1}^9 (I(K,L) - \bar{I}) * (R(I,J) - \bar{R}) \quad 6.3$$

$$\{E(M,N)\} = \sum_{I=1}^9 \sum_{J=1}^9 |I(K,L) - R(I,J)| \quad 6.4$$

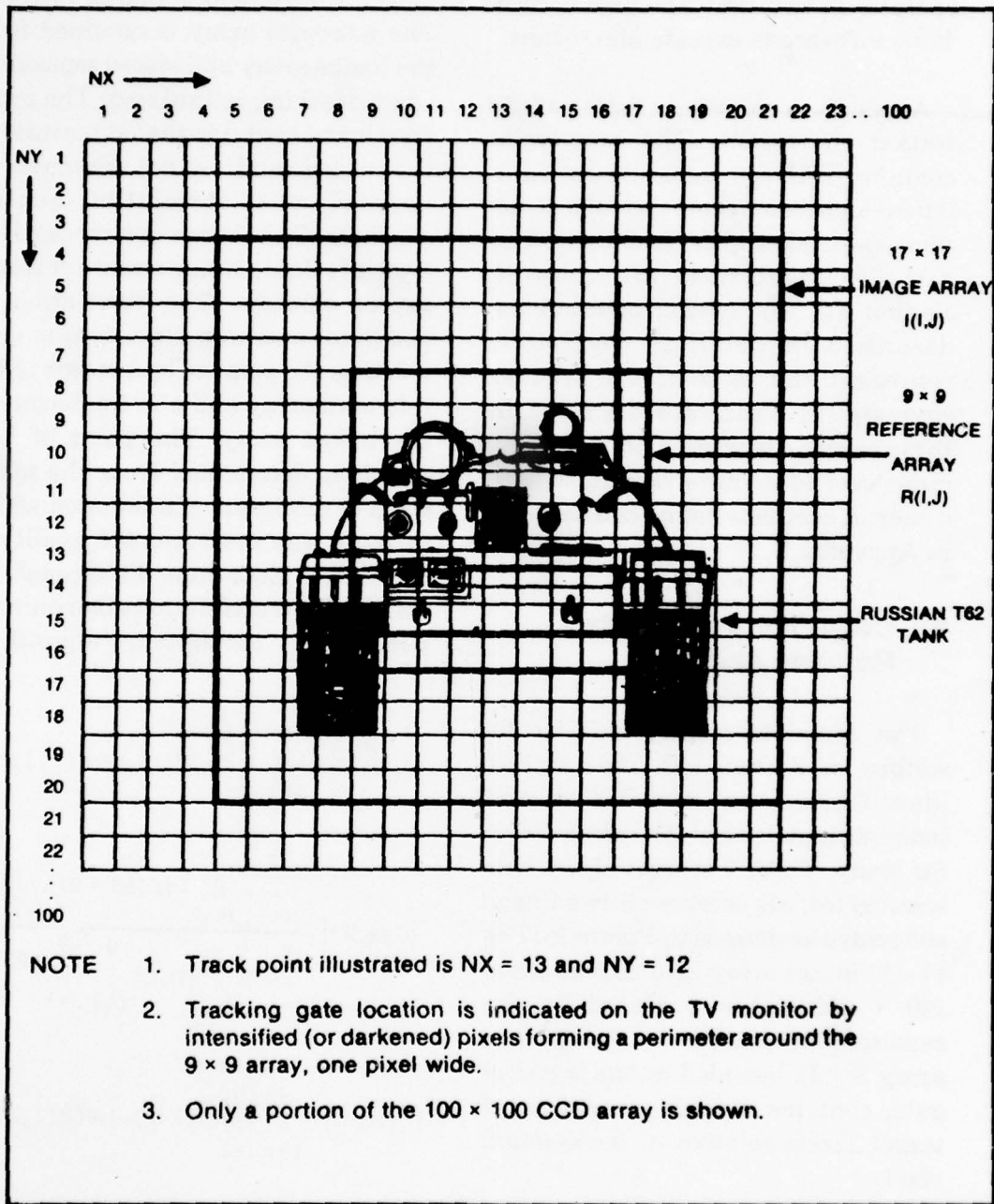


Figure 19. Definition of image and reference arrays.

$$\{E(M,N)\} = \sum_{I=1}^9 \sum_{J=1}^9 (I(K,L) - R(I,J))$$

where $K=I+M-1$, $L=J+N-1$, $1 \leq M \leq 9$,

$1 \leq N \leq 9$ and where

$$\bar{R} = \frac{1}{81} \sum_{I=1}^9 \sum_{J=1}^9 R(I,J)$$

$$\bar{I} = \frac{1}{81} \sum_{I=1}^9 \sum_{J=1}^9 I(I,J)$$

tested from imagery from the Fairchild 201 CCD TV. As a preliminary test, a 35MM slide representative of the IR signature of a target was projected on a screen. The image data obtained in this manner is approximately representative of the image data from a camera operating in the IR spectrum.

Figure 20 is a photograph of the TV monitor showing the projected image of such a slide. The bright areas correspond to the hottest areas in target, which in this case, are the tracks and bogey wheels.

The image and reference arrays shown in Figure 21 were obtained from the Fairchild 201 CCD TV and micro-processor hardware. The reference array is the center 9×9 of the 17×17

6.5 image array. The arrays were used to test the candidate tracking algorithms listed in Equations 6.1 through 6.5. Each algorithm results in a 9×9 array $C(M,N)$ or $E(M,N)$.

The correlation value and the side lobes of the match point give an indication of the robustness of the algorithm. Results from Equations 6.1 through 6.5 are presented in Figures 22 through 26 respectively. It is clear from the testing that scene brightness variation causes false peaks in the unnormalized correlation result. The results were surprising in view of the fact the shifting distance is so small. This test is representative of other results which also indicate the unnormalized correlation is not adequate as a tracking algorithm. Two normalization schemes were tried on the same data with improved results. In the case of normalization by norms the Cauchy Schwarz inequality guarantees a result less than one and less sensitivity to scene brightness variation. The result is given in Figure 23. Normalization by means as in Equation 6.3 gives similar results as shown in Figure 24. From an implementation point of view normalization by means is more difficult. As shown in Equation 6.3 the mean of the image 9×9 subarray and reference arrays must be known before the computation takes place. On the other hand, though a division is required, the normalization by norms for each correlation result can take place after the standard

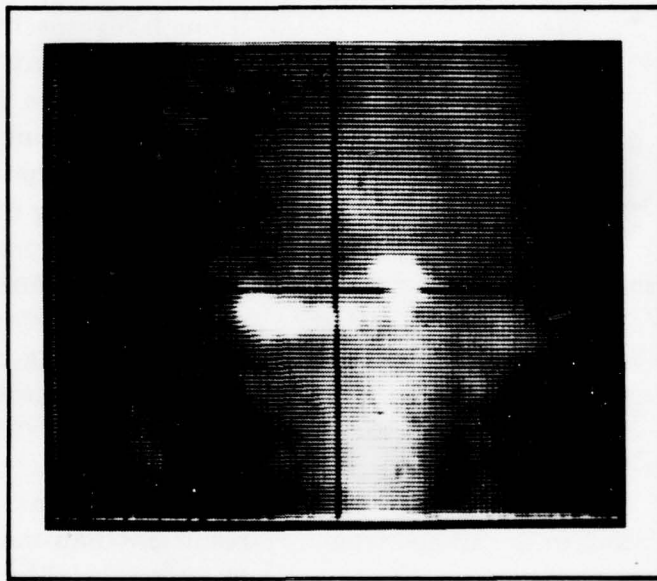


Figure 20. Video IR image of the side view of an M48 tank.

IMAGE ARRAY										REFERENCE ARRAY																			
3	3	3	3	4	5	5	7	8	9	10	11	12	13	14	15	15	15	15	15	11	12	12	13	13	14	15	15	15	15
2	3	4	4	4	5	4	6	7	9	9	9	12	13	14	15	15	15	15	15	12	14	15	15	15	15	15	15	15	15
4	4	3	4	4	6	5	6	7	8	7	9	12	13	14	15	15	15	15	15	12	13	13	13	14	15	15	15	15	15
3	3	3	4	3	5	5	5	5	7	7	9	12	13	13	14	15	15	15	15	12	13	13	13	14	15	15	15	15	15
3	3	3	4	3	5	5	6	6	8	7	9	12	14	15	15	15	15	15	15	12	14	15	15	15	15	15	15	15	15
3	4	3	3	4	5	6	6	6	8	6	7	8	12	14	15	15	15	15	15	12	14	15	15	15	15	15	15	15	15
2	3	4	4	3	6	6	6	6	6	6	7	8	10	11	11	11	11	11	11	10	11	11	11	11	11	11	11	11	11
2	4	4	3	3	5	6	6	6	7	8	8	10	10	12	11	9	7	7	10	10	12	11	10	9	7	7	7	7	
1	2	3	3	2	5	7	7	7	7	7	8	12	13	12	12	10	7	7	12	13	12	12	10	7	7	7	7	7	
3	3	0	5	7	7	7	7	7	7	8	8	13	13	15	13	12	8	8	13	13	15	13	12	8	8	8	8	8	
3	4	5	4	5	7	6	7	6	8	8	8	11	13	13	11	11	8	8	11	13	13	11	11	8	8	8	8	8	
4	7	4	6	7	9	8	7	6	7	8	8	10	10	12	11	10	8	8	10	10	12	11	10	8	8	8	8	8	
5	6	5	7	8	9	7	7	7	7	8	8	11	10	11	10	9	7	7	11	10	11	10	9	7	7	7	7	7	
6	7	8	10	10	11	9	8	7	8	9	9	12	12	11	10	11	8	8	12	12	11	10	11	8	8	8	8	8	

Figure 21. Image and reference arrays used for testing. (See Equation 6.1)

CORRELATION UNNORMALIZED

2834	2797	2854	2822	2870	2895	3054	3195	3464
3202	3130	3170	3116	3138	3170	3311	3410	3668
3555	3440	3454	3410	3419	3474	3589	3705	3948
4152	4012	4007	3995	4066	4088	4204	4331	4537
4917	4770	4738	4733	4837	4826	4877	4936	5074
5668	5505	5434	5408	5455	5463	5465	5437	5498
6273	6088	5981	5917	5916	5882	5850	5779	5745
6700	6483	6321	6190	6156	6095	6026	5914	5863
6660	6430	6226	6063	5997	5947	5884	5761	5708

Figure 22. Unnormalized correlation. (See Equation 6.3)

CORRELATION NORMALIZED BY NORMS

.97687	.96886	.97129	.96364	.95890	.95647	.94968	.93949	.92907
.97082	.96125	.96326	.95564	.95332	.95268	.94723	.93532	.92957
.97633	.96632	.96602	.96342	.96134	.96506	.95976	.95664	.94994
.97528	.96819	.96875	.97220	.98025	.97359	.97280	.97219	.97053
.98507	.97989	.97953	.98524	1.00000	.98627	.98097	.97792	.98027
.98026	.97490	.97294	.97858	.98452	.97798	.97121	.96638	.97101
.97716	.97017	.96792	.97093	.97375	.96714	.96245	.96050	.96371
.96379	.95476	.95165	.95164	.95525	.94816	.94251	.93991	.94517
.93266	.92343	.91831	.91785	.91971	.91585	.91224	.90977	.91674

Figure 23. Correlation normalized by norms.

CORRELATION NORMALIZED BY MEANS

270.00000	240.00000	241.00000	237.00000	250.00000	247.00000	252.00000	214.00000	224.00000
309.00000	258.00000	231.00000	230.00000	238.00000	217.00000	211.00000	170.00000	183.00000
350.00000	284.00000	263.00000	254.00000	256.00000	269.00000	258.00000	248.00000	235.00000
439.00000	376.00000	364.00000	380.00000	409.00000	389.00000	379.00000	380.00000	365.00000
598.00000	549.00000	531.00000	554.00000	616.00000	556.00000	509.00000	477.00000	475.00000
652.00000	601.00000	565.00000	574.00000	600.00000	566.00000	505.00000	449.00000	440.00000
665.00000	613.00000	569.00000	561.00000	567.00000	519.00000	466.00000	423.00000	396.00000
521.00000	465.00000	422.00000	421.00000	436.00000	382.00000	327.00000	285.00000	276.00000
250.00000	202.00000	159.00000	150.00000	179.00000	150.00000	115.00000	83.00000	93.00000

Figure 24. Correlation normalized by means. (See Equation 6.4)

SUM OF ABSOLUTE DIFFERENCES

231	224	228	223	207	202	195
186	190	199	187	187	185	180
150	158	169	166	156	158	158
112	124	126	108	116	116	117
84	92	70	0	87	100	96
149	144	123	112	149	159	149
207	195	179	170	197	191	192
284	270	252	227	252	248	236
351	334	322	311	316	309	294

Figure 25. Sum of absolute differences template matching. (See Equation 6.5)

METRIC	
909	783
602	720
460	512
200	201
154	220
413	469
011	694
1420	1066
2059	1436
966	838
769	765
577	520
363	270
196	232
419	507
002	763
1403	1194
2001	1605
914	867
736	741
572	550
360	290
190	193
410	453
769	775
1316	1236
1080	1670
911	911
706	706
560	560
306	306
135	135
362	362
720	720
1190	1190
1660	1660
949	949
001	001
614	614
262	262
0	0
274	274
636	636
1111	1111
1633	1633

Figure 26. Sum of square differences template matching.

correlation. The norm of the image array may be conveniently computed at the same time as the correlation result. Two template matching algorithms [3] shown in Equation 6.4 and 6.5 were also tested with good results. Template algorithms may be rigorously defined in terms of a distance metric [4]. The result from testing is given in Figure 25 and Figure 26 respectively. It should be noted that Equation 6.5 contains a correlation computation which is evident when the squared term is multiplied out. The algorithms in Equations 6.1, 6.4, and 6.5 have been implemented in real time. This report, however, contains only the software for the algorithms in Equations 6.4 and 6.5. The execution times for these algorithms as defined in the equations is on the order of 6.46 ms with a 160 nsec cycle time and a 320 nsec memory read cycle.

C. Tracker Description

This section describes the operation of the tracker macrossoftware given in Appendix A. An examination of the software mnemonics show that the code is basically typical minicomputer code as defined in Section 7.A. The exceptions are: WIN, COR, STD, CROSS, FRM, WAT, IMG, REF, MNV, and LAIBUS.

These instructions are covered here briefly and in more detail in Section 7.C. The variables and constants in cells 208 through 418 can be easily accessed in microcode by constants generated in the K bus logic. A

complete list of variable names and definitions are given in Figure 27.

The tracker code given in Appendix A contains both adaptive gate centroid code and template matching code. The following sections contain only a description of the template matching code which starts at label START2. The adaptive gate tracker code is documented in a separate report [5].

Tracker operation can be broken into two major computation loops, the scan loop and the track loop. These computation loops are flow charted in Figures 28 and 29 respectively. The hardware reset shown at the top of Figure 28 initializes the tracker hardware and software. When initialization is complete, a crosshair is displayed as an overlay on the TV monitor, centered at column 50, row 50 ($NX = NY = 50$) as shown in Figure 20. The crosshair is produced by routine CROSS. The operator then aims the camera at the target to be tracked. In roughly .5 seconds the image data is scanned and the GAIN and BIAS latches are set to optimum values. The iterative process of setting the GAIN and BIAS latches continues until the track switch is turned on. At this point the track loop is entered as shown in Figure 29. The AGC conditions existing at the time of track initiation are stored. Then continuing AGC action maintains the quality of the reference in spite of changes in scene brightness and local area responsivity in the CCD TV camera. The reference array is stored

VARIABLE NAME	DEFINITION
SSW7	CONSTANT, USED TO SELECT SSW7
ONEMC1	CONSTANT, USED IN FILTER EQUATION
FRMRQ	CONSTANT
ONE	CONSTANT
IP	CONSTANT, IMAGE POINTER
RP	CONSTANT, REFERENCE POINTER
CP	CONSTANT, CORRELATION POINTER
IP2	(NOT USED)
MEAN	VARIABLE, (MEAN VALUE)*81
STD	VARIABLE, (MEAN DEVIATION)*81
MAXI	VARIABLE, MAXIMUM OF CORRELATION ARRAY
MINI	VARIABLE, MINIMUM OF CORRELATION ARRAY
NX	VARIABLE, CURRENT COLUMN TRACK POINT
NY	VARIABLE, CURRENT ROW TRACK POINT
DELXX	VARIABLE, DELTA ROW TRACK CORRECTION
DELYY	VARIABLE, DELTA COLUMN TRACK CORRECTION
MODE	VARIABLE, CURRENT MODE LATCH
BWC	VARIABLE, CURRENT BLACK/WHITE CONTROL WORD
GAIN	VARIABLE, CURRENT GAIN LATCH VALUE
BIAS	VARIABLE, CURRENT BIAS LATCH VALUE
TP5	CONSTANT, (DEC .5)
TIP5	CONSTANT, (DEC 1.5)
N10	CONSTANT, LIMIT FOR TRACKING AREA
N90	CONSTANT, LIMIT FOR TRACKING AREA
TP5D	CONSTANT, (DEC .5)
TIP5D	CONSTANT, (DEC 1.5)
T7PO	CONSTANT, (DEC 7.0)
T8PO	CONSTANT, (DEC 8.0)
T14P5	CONSTANT, (DEC 14.5)
T13P5	CONSTANT, (DEC 13.5)
T31	CONSTANT, INITIAL VALUE FOR GAIN LATCH
DCBLK	CONSTANT, INITIAL VALUE FOR BIAS LATCH
T81	CONSTANT, (DEC 81)
T81PRM	CONSTANT, (DEC 81)
ONEONE	CONSTANT, BIAS ADJUSTMENT CONSTANT
ONEMC2	CONSTANT, FILTER EQUATION
C2	CONSTANT, FILTER EQUATION
GP933	CONSTANT, GAIN ADJUSTMENT FACTOR
FIFTY	CONSTANT, (DEC 50)
FULLG	CONSTANT, MAXIMUM GAIN LIMIT
FULLB	CONSTANT, MINIMUM BIAS LIMIT
ZERO	CONSTANT
MNSBPK	VARIABLE, BIAS ADJUSTMENT PARAMETER
SDSBPK	VARIABLE, AGC ADJUSTMENT PARAMETER
MNPK	VARIABLE, BIAS ADJUSTMENT PARAMETER
MNSPK	(NOT USED)
SDPK	VARIABLE, AGC ADJUSTMENT PARAMETER
TEMP	VARIABLE, WORKING
MEANO	VARIABLE, OLD VALUE OF MEAN
STDO	VARIABLE, OLD VALUE OF MEAN DEVIATION
C1	CONSTANT, FILTER EQUATION

Figure 27. List of variable names and definitions.

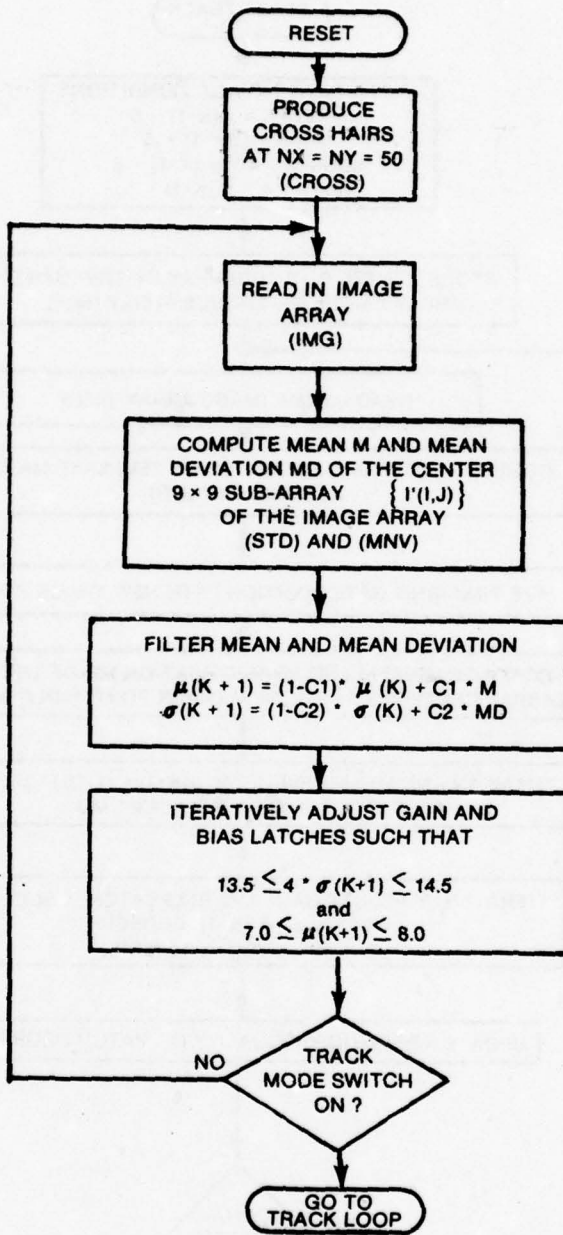


Figure 28. Flow chart of scan loop.

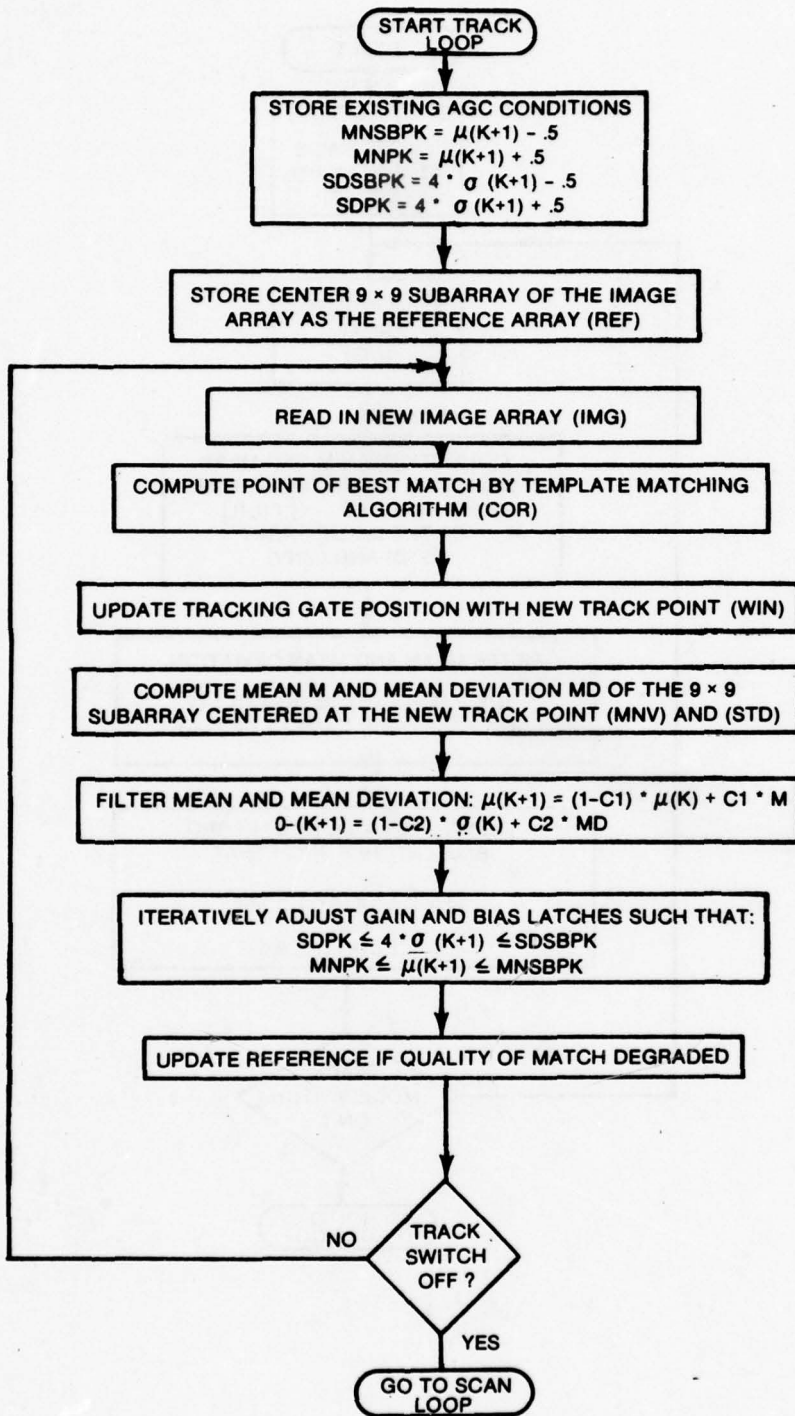


Figure 29. Flow chart of track loop.

and a new image array is read in while computation continues (in parallel). At this point the template matching algorithm finds the point of best match and the tracking gate positions are updated. New mean and mean deviation values are computed for the 9×9 subarray centered at the track point. The mean and mean deviation are then filtered and subsequently used by the iterative gain and bias adjustment software. The reference array is updated if the quality of match is degraded. This closes the track loop which starts by reading in a new image array. The iterative gain and bias computations appear in both the scan and track loops. The details of the computation are given in Figure 30. The computation starts after the filtered mean and mean deviation have been computed. The mean is computed by the equation.

$$M = \frac{1}{81} \sum_{I=1}^9 \sum_{J=1}^9 I'(I,J)$$

where $I'(I,J)$ is the 9×9 subarray centered at the current track point. Similarly the mean deviation is computed by

$$MD = \frac{1}{81} \sum_{I=1}^9 \sum_{J=1}^9 |I'(I,J) - M|$$

The filter equations are given in Figure 28. It should be noted that in the scan

loop the following constants are held to fixed values indicated below.

$$MNPK = 8$$

$$MNSBPK = 7$$

$$SDPK = 14.5$$

$$SDSBPK = 13.5$$

This choice of the constants has the effect of optimizing the gain and bias settings so that the full range of the analog-to-digital converter is utilized (0 to 15 counts). The effect of successive iterations is that the 9×9 subarray has a mean value of 7.5 counts and a dispersion ($4\sigma (K+1)$) of 14 counts. Since gain is limited to 5, the dispersion may not reach 14 counts for very low contrast targets, but the mean will always reach 7.5 (filtered value). This technique can cause extremely bright or dark objects to be clipped to 15 or 0 counts respectively, but the overall effect is to enhance the available contrast in the target. The software AGC performed quite well, and enabled the tracker to track the detail that exists in a piece of plain white bond typing paper in preliminary lab tests as shown in Figure 31. Figure 32 shows the enhancement of a very low contrast target which is barely visible on the monitor. Figure 33 shows results of a real time scan and track loop tests on a simulated M48 tank target.

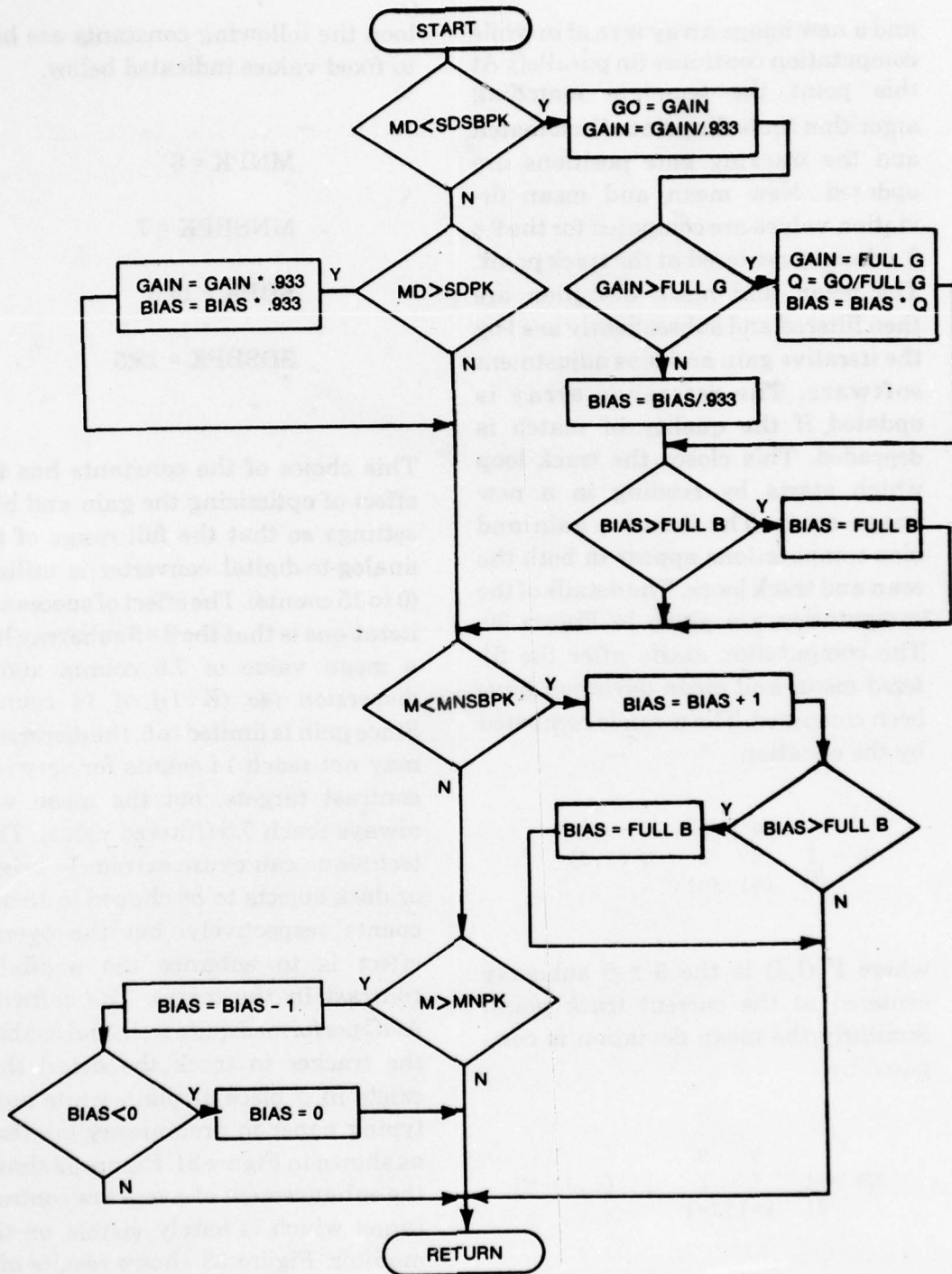
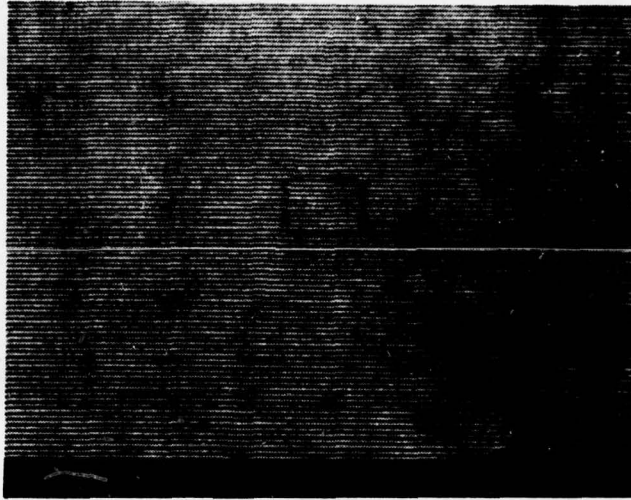
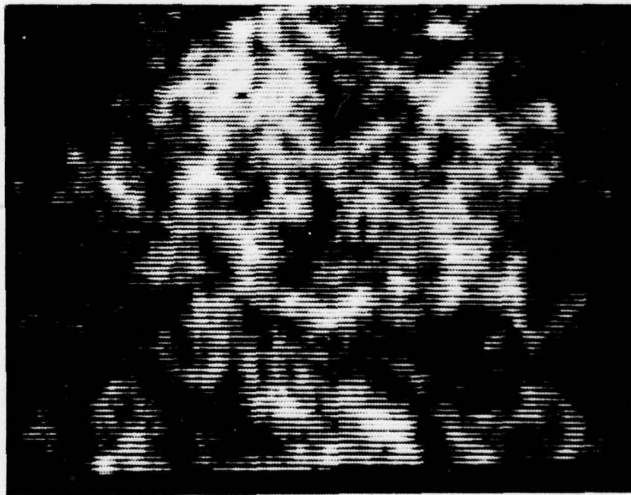


Figure 30. Iterative GAIN and BIAS flow chart.

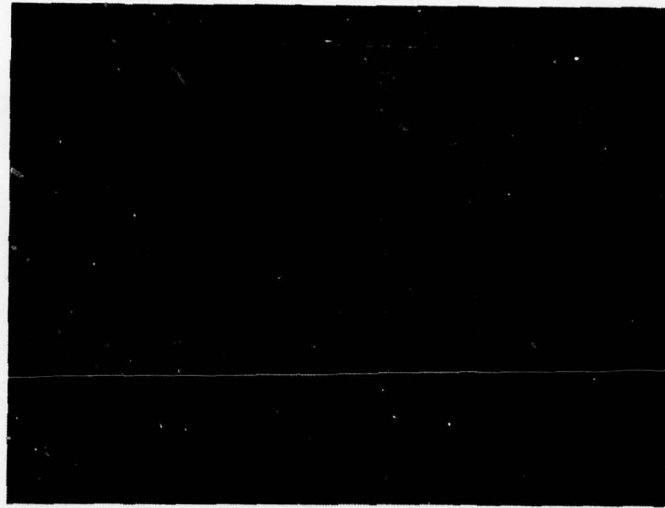


LIVE VIDEO

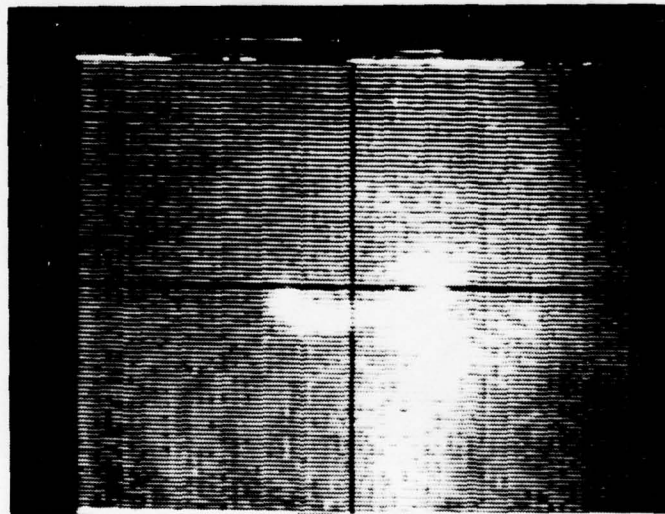


DIGITIZED AND ENHANCED VIDEO

Figure 31. Plain paper tracking tests.



LOW CONTRAST
CAMERA VIDEO



DIGITIZED AND ENHANCED VIDEO

Figure 32. Results from automatic GAIN and BIAS adjustments.



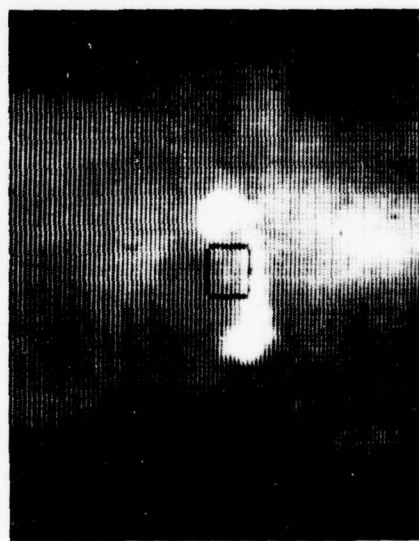
DIGITIZED AND ENHANCED VIDEO
(SCAN LOOP)



DIGITIZED AND ENHANCED VIDEO
(TRACK LOOP)



CAMERA VIDEO
(SCAN LOOP)



CAMERA VIDEO
(TRACK LOOP)

Figure 33. Results of real time scan and track loop tests on a simulated M48 tank target.

The reference update software is not given in this report though several methods for measuring the quality of the template match have been attempted. The methods generally involve the examination of the side lobes of the $E(M, N)$ array about the point of match. No suitable wide range zoom lens was available to make the necessary tests for developing the algorithms. It is hoped that a wide range zoom will be available in FY78 so that the algorithms can be developed. One approach for checking quality is to compute

$$Q = \left\{ E(M, N) \right\} \Big|_{\max} - \left\{ E(M, N) \right\} \Big|_{\min}$$

and to examine the side lobes about

$$\{m, n\} = \{(m, n) | E(m, n) \text{ minimum}\}$$

Define the set

$$\{D\} = \{E(m, n) - E(k, \ell) \mid |m-k|=1, \\ |n-\ell|=1\}$$

Then if $Q < T_q$ and $D|_{\min} < T_s$ (T_q and T_s are thresholds) the reference array is updated. A change in the color in the tracking gate has been used as a cue to the operator that a reference update has taken place.

7. DETAILED SOFTWARE DESCRIPTION

A. EAI 640 Emulation Code

A subset of the EAI 640 minicomputer instruction set has been emulated on the Intel 3000 microprocessor. A listing of the code is given in Appendix C. The code operates roughly 4 times faster than the EAI 640 and about 2 times faster than the Pacer 100, a newer model EAI digital computer.

The basic set of macroinstructions shown in Figures 34 through 37 is a modified EAI 640 instruction set. The symbols A, X, P, S, and W in the "operation performed" column refer to the microprogrammed machine's registers: accumulator, index register, program counter, stack pointer and status register respectively. The macroinstructions are all 20 bits wide and have two possible formats shown below.

Memory reference

A	B	6 bit OP Code	12 bit Displacement
└──────────┘			
2 bit address option			

Other

8 bit OP Code	12 bit Displacement
---------------	---------------------

CODE	BINARY	OCT	
LA	AB00 0000	000	C(E) → A
STA	AB00 0001	001	A → C(E)
LX	AB00 0010	002	C(E) → X
STX	AB00 0011	003	X → C(E)
A	AB01 0100	004	A+C(E) → A
S	AB00 0101	005	A-C(E) → A
M	AB00 0110	006	A*C(E) → A
D	AB00 0111	007	A/C(E) → A
AOM	AB00 1000	010	A+1 → A
OR	AB00 1001	011	AVC(E) → A
XOR	AB00 1010	012	A+C(E) → A
AND	AB00 1011	013	A C(E) → A
C	AB00 1100	014	SET FLAG FOR SKIP INSTRUCTIONS
NOT USED	AB00 1101	015	
NOT USED	AB00 1110	016	
NOT USED	AB00 1111	017	
J	AB01 0000	020	D → P (OR C(E) → P)
L	AB10 0000	040	P → S, S+1 → S, D → P (OR C(E) → P)

	ADDRESS OPTION	OP CODE	DISPLACE- MENT	AB	ADDRESS OPTION
MEMORY REFERENCE CODE	Z	OP	D	00	Direct
				01	Indirect
				10	Indexed
				11	Indirect, Indexed
	2 bits	6 bits	12 bits		

Figure 34. Memory reference instructions.

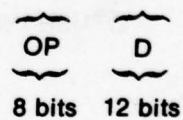
CODE	BINARY	OCT	
CLR	0011 0000	060	$0 \rightarrow A$
AOA	0011 0001	061	$A + 1 \rightarrow A$
CAO	0011 0010	062	$1 \rightarrow A$
TCA	0011 0011	063	$1 + A \rightarrow A$
ARS	0011 0100	064	ARITHMETIC RIGHT SHIFT (D TIMES)
ALS	0011 0101	065	ARITHMETIC LEFT SHIFT (D TIMES)
LRS	0011 0110	066	LOGICAL RIGHT SHIFT (D TIMES)
SSP	0011 0111	067	SET SIGN BIT OF A REG TO ZERO
SSN	0011 1000	070	SET SIGN BIT OF A REG TO ONE
EX	0011 1001	071	$X \rightarrow A$ and $A \rightarrow X$
EP	0011 1010	072	$A \rightarrow P$ and $P \rightarrow A$
ES	0011 1011	073	$A \rightarrow W$ and $W \rightarrow A$
ICX	0011 1100	074	$X + D \rightarrow X$, SKIP NEXT IF SIGN BIT CHANGES
DCX	0011 1101	075	$X - D \rightarrow X$, SKIP NEXT IF SIGN BIT CHANGES

INSTRUCTION FORMAT FOR REGISTER OPERATIONS	OP CODE	DISPLACEMENT
	OP	D
	8 bits	12 bits

Figure 35. Register operation instructions.

CODE	BINARY	OCT	
SE	0111 0000	160	SKIP IF C(E) = A
SG	0111 0001	161	SKIP IF A > C(E)
SL	0111 0010	162	SKIP IF A < C(E)
SNE	0111 0011	163	SKIP IF A ≠ C(E)
SGE	0111 0100	164	SKIP IF A ≥ C(E)
SLE	0111 0101	165	SKIP IF A ≤ C(E)
SKN	0111 1010	172	SKIP IF A < 0
SKP	0111 1011	173	SKIP IF A > 0
SO	0111 1100	174	SKIP IF PREVIOUS CAUSED OVERFLOW
SNO	0111 1101	175	SKIP IF PREVIOUS CAUSED NO OVERFLOW
SAE	0111 1110	176	SKIP IF A REG EVEN
NOT USED	0111 1111	177	—
ICX	0011 1100	074	SEE REGISTER OPERATIONS
DCX	0011 1101	075	SEE REGISTER OPERATIONS

INSTRUCTION FORMAT FOR SKIP INSTRUCTIONS



*Valid following compare instruction (C) only.

Figure 36. Skip instructions.

CODE	BINARY	OCT	
NOP	0011 1110	076	NO OPERATION
P	0011 1111	077	PAUSE
PUSA	0111 0110	166	PUSH ALL REGISTERS ON STACK, S + 12 → S
PUSX	0111 0111	167	PUSH A, X, P, W ON STACK, S + 4 → S
POPX	0111 1000	170	POP A, X, P, W FROM STACK, S - 4 → S
RTN	0111 1001	171	POP P FROM STACK, S - 1 → S
DI	1111 1100	374	DEVICE D → A
DO	1111 1101	375	A → DEVICE D
NOT USED	1111 1110	376	
NOT USED	1111 1111	377	

OTHER INSTRUCTION FORMAT--(SAME AS REGISTER OPERATIONS)

Figure 37. Other Instructions.

In the case of memory reference instructions, there are four addressing options.

	AB
Direct	00
Indirect	01
Indexed	10
Indirect, Indexed	11

In Figure 34, the symbol E is used to indicate an effective address formed in one of the four possible ways of. If D is the symbol for a 12 bit displacement in the instruction word, then

Direct $E = D$

Indirect $E = C(D)$

Indexed $E = X + D$ (sum of Index Register and D)

Indirect, Indexed $E = C(D) + X$

where the symbol C(D) represents the content of memory location D.

The register assignments in the Intel 3002 chip are made as follows:

Machine	Register
A	R0
X	R1
	R2 (Working)
P	R3

S	R4
	R5 (Working)
	R6 (Working)
	R7
	R8 (Working)
	R9 (Working)
	AC (Working)
W	T (Working)

Registers R2, R5, R6, R8, R9, AC, and T are used as working registers in microcode operations. Registers R0, R1, R3, R4, and R7 are maintained for the microprogrammed machining. The A and X registers are used as conventional accumulator and index registers respectively. The P register is the program counter and contains the address of the next instruction to be executed. The S register is the stack pointer and contains the next available address for the storage stack located in main memory. The W register is used as a status indicator and will be used in interrupt code.

When the front panel reset switch is activated, a logic high is placed on pin 36 of the Intel 3001 microprogram control unit. The logic high forces the instruction at location 0008 in micromemory to be executed. The initialization code for the microprocessor is placed at this location.

The last instruction in the initialization code is a JZR (FETCH). FETCH is a label for row zero column

15 and is the starting location for the instruction fetch code. The initialization code is given below and assumes that the initial value of the stack pointer and program counter are located in cells 0 and 1 of main memory respectively.

The instruction fetch code is located in row zero column 15 (OOF16), and is used to fetch the next instruction to be executed from macromemory. The code includes a mask so only the 12 bit displacement is loaded into the AC register of the Intel 3002. The AC

```

000H:   INIT:   CLR (A);           0→A
          CLR (X);           0→X
          CLR (W);           0→W
          CLR (T);           0→T
          LMI (T);          T→MAR

/*WRITE ZERO TO INTERRUPT STRUCTURE*/
          ILR (W)   ROT;      W AC
          LMI (T)   FF1 RRM;  0 MAR, T + T
          ACM (AC);  C (0) Ac AC

/* C (0) = INITIAL VALUE OF STACK POINTER*/
          SDR (S)   FF1;      AC→S
          LMI (T)   ff1 RRM;  1→MAR, T + 1→T
          ACM (AC);  C (1)→AC

/* C (1) =STARTING ADDRESS OF PROGRAM*/
          SDR (P) FF1          JZR (FETCH)

```

register is also copied into R9 for later usage. The last instruction in the fetch sequence contains a JPX instruction which causes a 16-way branch to the various "classes" of instructions. Figure 38 shows all conditional

branches used to determine the macro-instruction to be executed. Note that many unused instructions are shown in Figure 38 (FETCH), allowing for future growth to more complex macro-instructions. The fetch code also

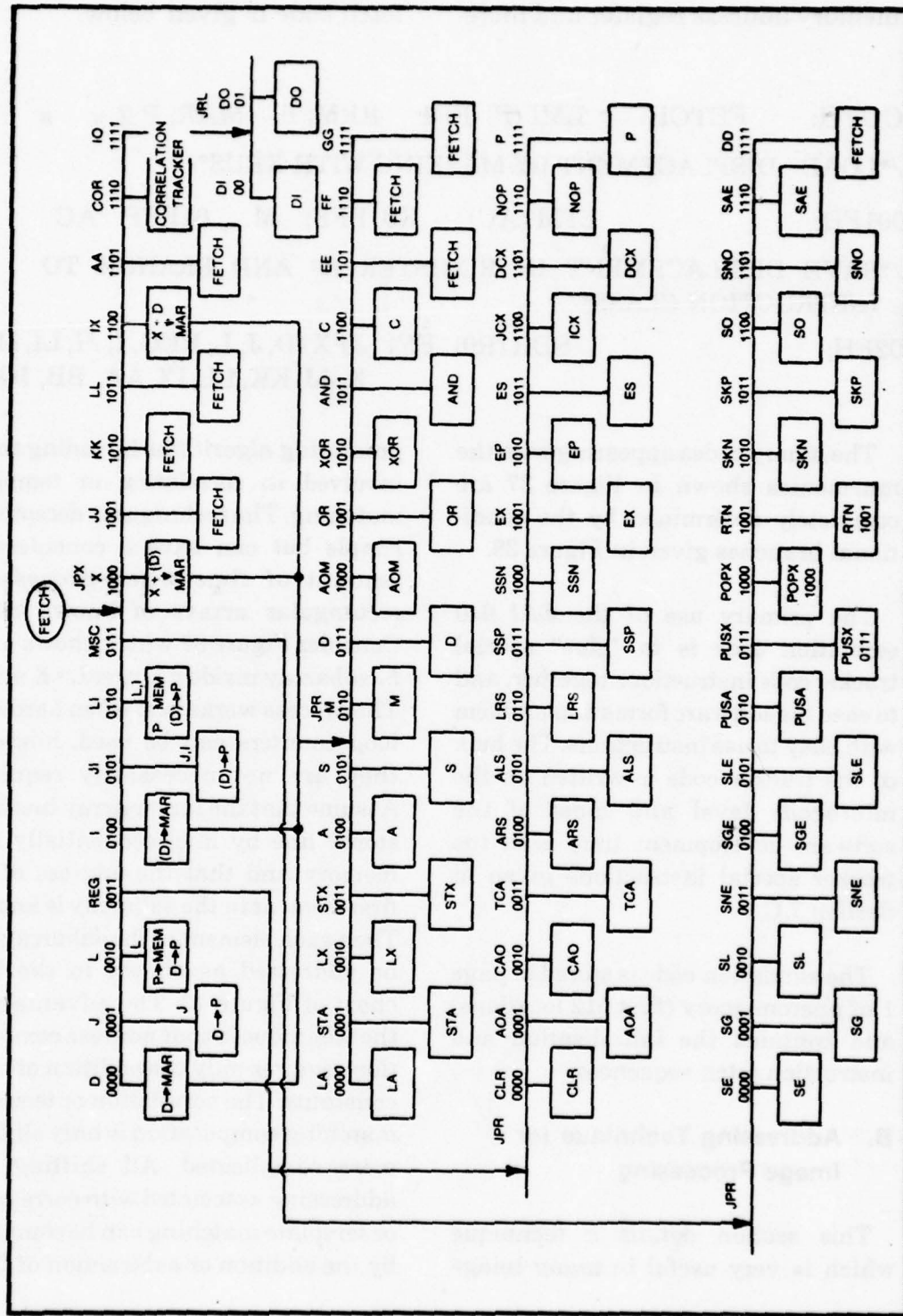


Figure 38. Conditional branches used to select microcode sequence.

forces the P register contents to the memory address register and incre-

ments the P counter by one. The fetch code is given below.

```
OOFH:    FETCH:    LMI (P) FF1   RRM; P   MAR, P Z  $\pi$   $\pi$ 
/*LOAD  DISPLACEMENT BY MASKING WITH KBUS*/
001FH;           LTM (AC)   K01FFF; M   01FFP AC
/*SAVE  DISPLACEMENT IN REGISTER R9 AND BRANCH TO
INSTRUCTION CLASS*/
02FH;           SDR (R9) FF1   JPX (D, J, L, REG, I, JI, LI, MSC,
X, JJ, KK, LL, IX, AA, BB, IO);
```

The binary codes appearing with the mnemonics shown in Figure 37 are completely determined by the conditional branches given in Figure 38.

The primary use of the EAI 640 emulation code is to "glue" special tracker code instructions together, and to ease the software formation problem with easy to use instructions. The bulk of the tracker code is written on the microcode level and most of the software development time is in the tracker special instructions given in Section 7.C.

The emulation code is stored in page 1 of micromemory (first 512 locations) and contains the initialization and instruction fetch sequences.

B. Addressing Technique for Image Processing

This section details a technique which is very useful in many image

processing algorithms including those involved in correlation or template matching. The technique is deceptively simple but can save a considerable amount of time when processing rectangular arrays of known sizes. Consider Figure 39 which shows a $N \times M$ subarray inside a larger $L \times K$ array. The process works best when hardware loop counters can be used, however, they are not necessarily required. Assume that the larger array has been stored line by line, sequentially into memory and that the address of the first element in the subarray is known. Then each element of the subarray can be addressed as shown in the flow chart of Figure 40. The advantage of the technique is that address computation requires only the addition of fixed constants. The correlation or template matching computation is only slightly more complicated. All shifting and addressing associated with correlation or template matching can be computed by the addition or subtraction of fixed

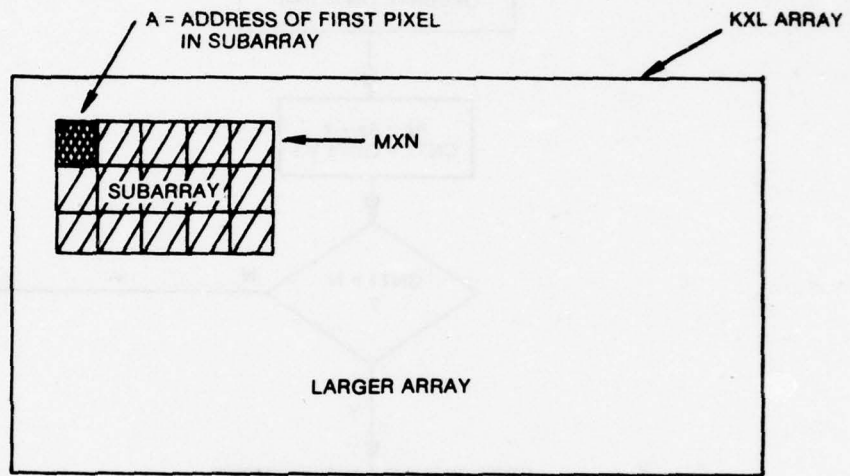


Figure 39. Subarray inside larger array.

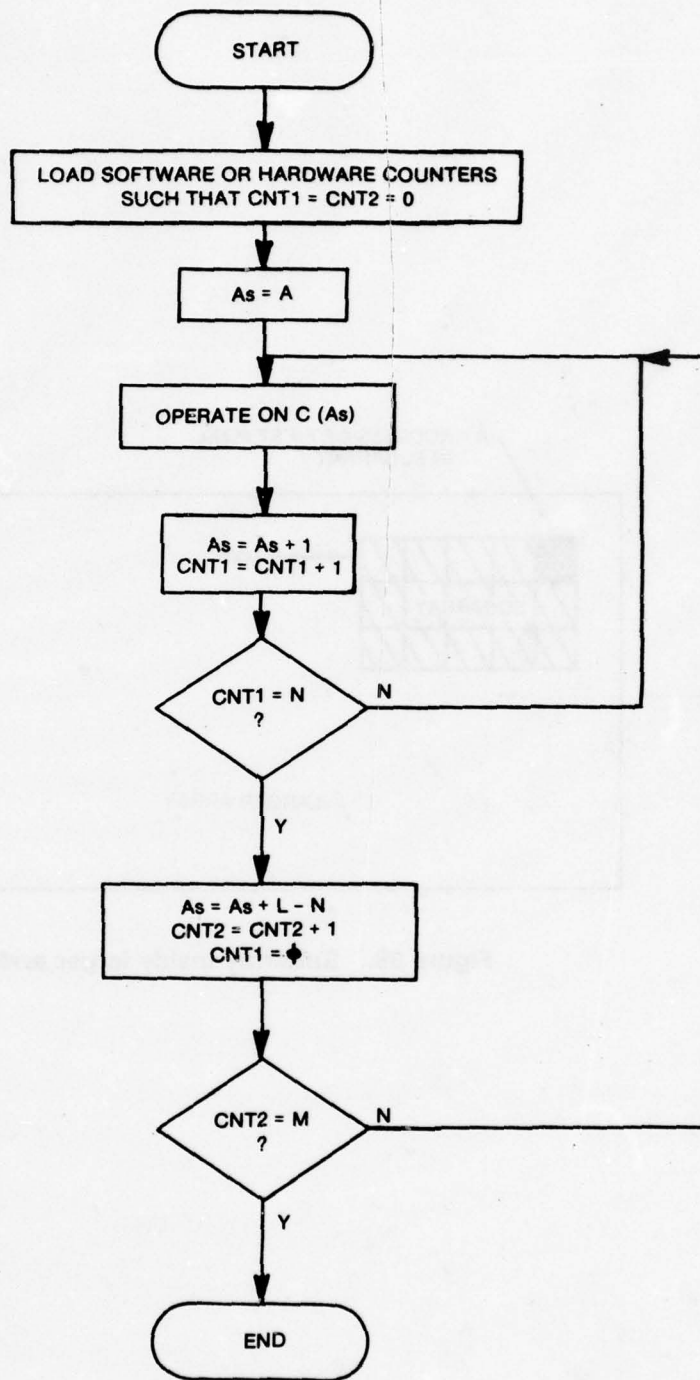


Figure 40. Addressing technique for image processing.

constants. Multiplications for address formation are eliminated. An example of template matching addressing can be found in the flow chart of Figure 46.

C. Tracker Microcode

(1) **Routine WIN.** Routine WIN is used to generate tracking gates to indicate track point on the TV monitor. The track point is defined by (NX, NY) the column and row locations of the tracking gate center. Figure 19 shows the 9×9 array which is inclosed by the tracking gate of fixed size, and Section 3.C gives a description of the latches which must be loaded to generate a tracking gate on the TV monitor.

The variables NX and NY are stored in macromemory, and routine WIN reads these values along with the black/white control word which defines the color of the tracking gate.

The mode latch is forced to the tracking gate mode with the requested tracking gate color. The mode latch status word is also updated to give the current mode latch status. Figure 41 is a flow chart of WIN and shows the addressing and loading of the tracking gate control latches with the 3002 MAR register.

(2) **Routine CROSS.** Routine CROSS is similar to WIN except that the mode latch is enabled to draw a crosshair. The right and left column latches are loaded with NX and the

upper and lower latches are loaded with NY. The crosshair color, as in routine WIN, is determined by the black/white control word in macromemory.

The flow chart for CROSS is shown in Figure 42.

(3) **Routine FRM.** Routine FRM is analogous to a shutter release on a camera. Execution of FRM causes a new frame to be loaded into image memory (10K memory).

The flow chart for FRM, shown in Figure 43, contains all operations performed in the routine.

(4) **Routine WAT.** Routine WAT is used only after FRM has been executed. Routine WAT tests the sign bit of the I bus to determine when a frame has been completely stored. If the frame is not ready, a wait loop is entered until the sign bit goes high (frame ready). The flow chart for WAT is given in Figure 44.

(5) **Routine IMG.** Routine IMG is used to read the image array $I(I,J)$, as shown in Figure 19, into macromemory stored starting at address IP (image pointer). The image pointer IP address is located at 20 octal in macromemory. The image is stored line by line, sequentially in memory. The routine is detailed in the flow chart shown in Figure 45. Since image data is read into macromemory during the

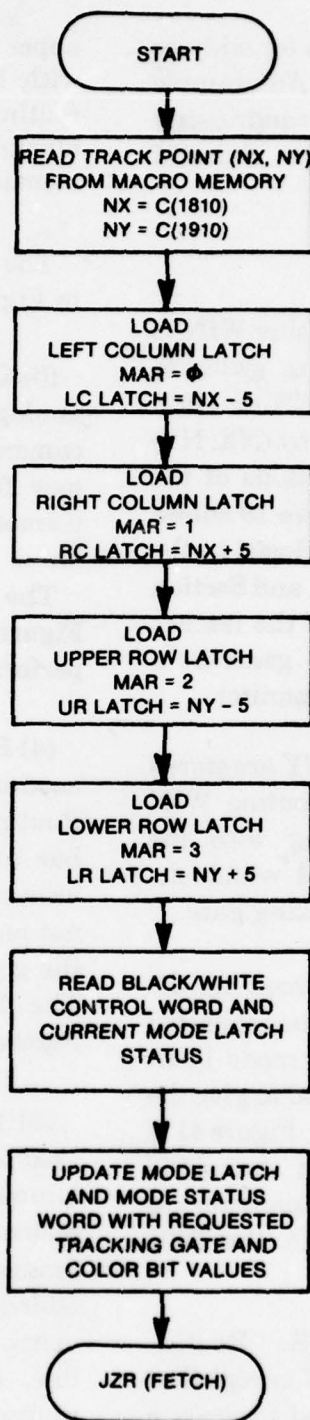


Figure 41. Flow chart for routine WIN.

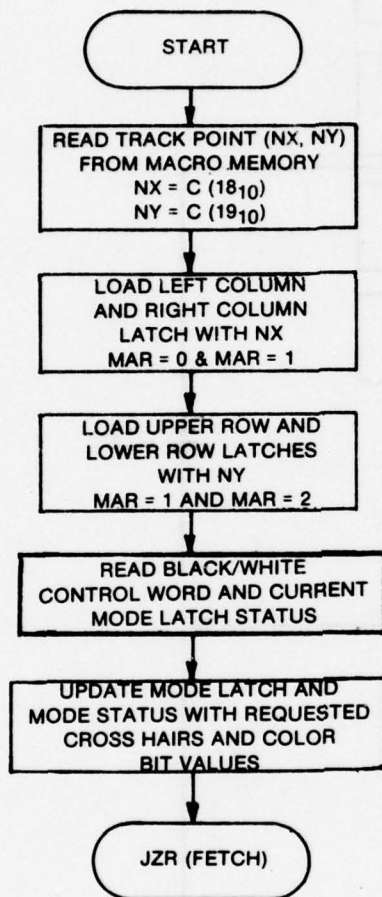


Figure 42. Flow chart for routine CROSS.

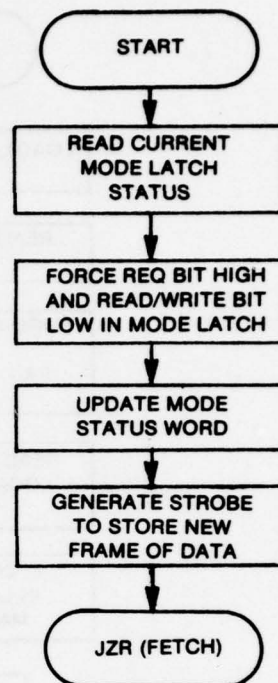


Figure 43. Flow chart for routine FRM.

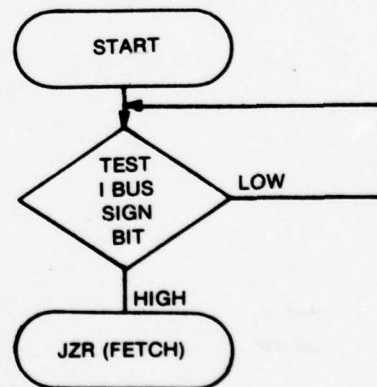


Figure 44. Flow chart for routine WAT.

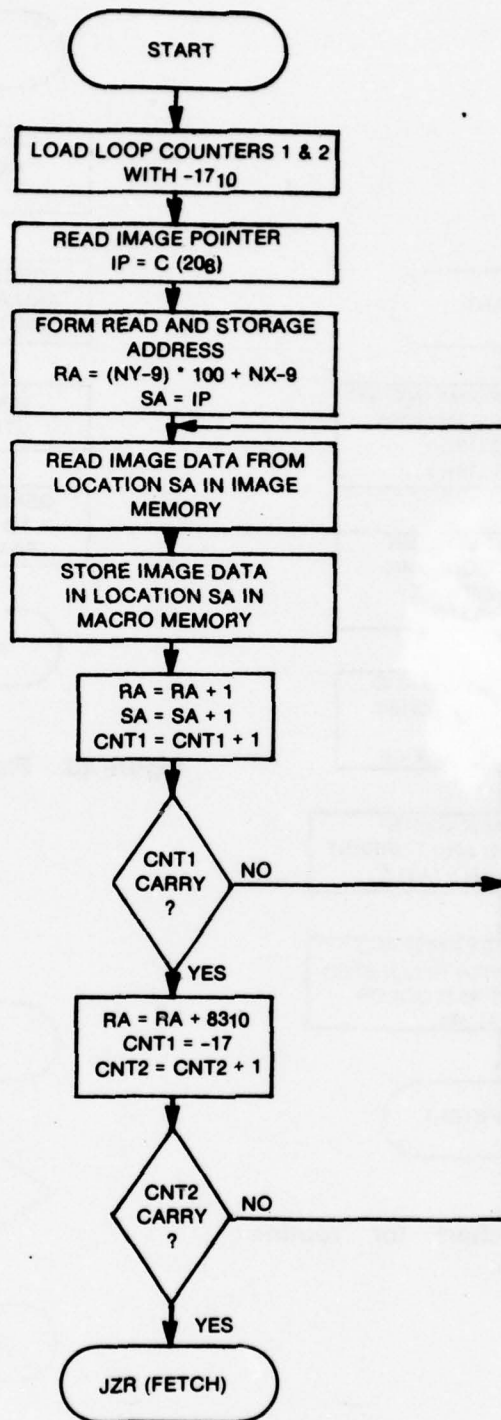
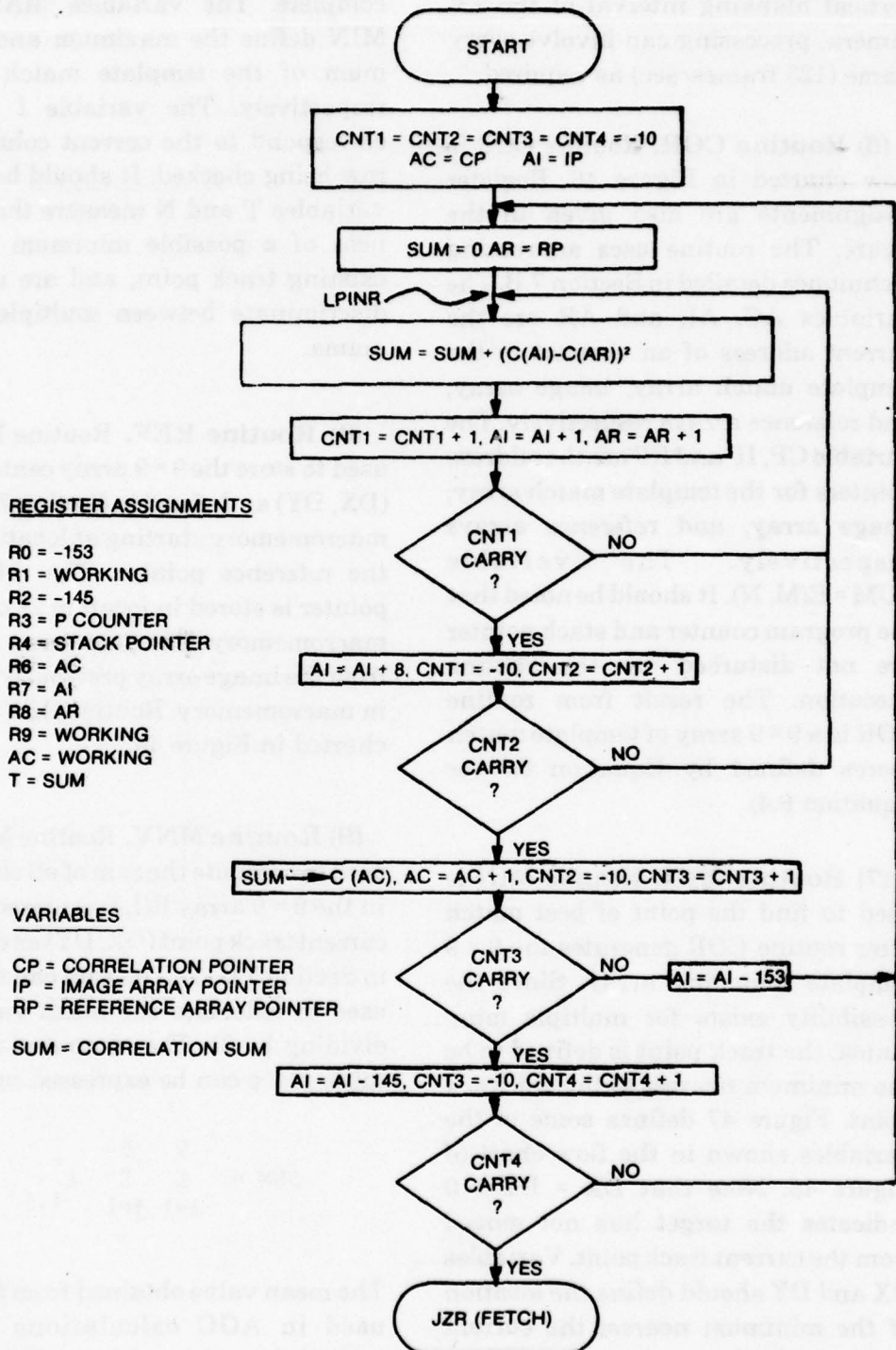


Figure 45. Flow chart for routine IMG.



REGISTER ASSIGNMENTS

- R0 = -153
- R1 = WORKING
- R2 = -145
- R3 = P COUNTER
- R4 = STACK POINTER
- R6 = AC
- R7 = AI
- R8 = AR
- R9 = WORKING
- AC = WORKING
- T = SUM

VARIABLES

- CP = CORRELATION POINTER
- IP = IMAGE ARRAY POINTER
- RP = REFERENCE ARRAY POINTER
- SUM = CORRELATION SUM

Figure 46. Flow chart for routine COR.

vertical blanking interval of the TV camera, processing can involve every frame (123 frames/sec) as required.

(6) Routine COR. Routine COR is flow charted in Figure 46. Register assignments are also given in the figure. The routine uses addressing techniques detailed in Section 7.B. The variables AC, AI, and AR are the current address of an element in the template match array, image array, and reference arrays respectively. The variable CP, IP and RP are the address pointers for the template match array, image array, and reference arrays respectively. The variable $SUM = E(M, N)$. It should be noted that the program counter and stack pointer are not disturbed by the routine execution. The result from routine COR is a 9×9 array of template match scores defined by Equation 6.5 (or Equation 6.4).

(7) Routine MIN. Routine MIN is used to find the point of best match after routine COR generates the 9×9 template matching array. Since the possibility exists for multiple minimums, the track point is defined to be the minimum nearest to the old track point. Figure 47 defines some of the variables shown in the flow chart of Figure 48. Note that $DX = DY = 0$ indicates the target has not moved from the current track point. Variables DX and DY should define the location of the minimum nearest the current track point when execution of MIN is

complete. The variables MAX and MIN define the maximum and minimum of the template match array respectively. The variable I and J correspond to the current column on row being checked. It should be noted variables T and N measure the closeness of a possible minimum to the existing track point, and are used to discriminate between multiple minimums.

(8) Routine REF. Routine REF is used to store the 9×9 array centered at (DX, DY) as defined in Section 7.C.(7) in macromemory starting at location RP, the reference pointer. The reference pointer is stored in location 21 octal in macromemory. The data array is read from the image array previously stored in macromemory. Routine REF is flow charted in Figure 49.

(9) Routine MNV. Routine MNV is used to compute the sum of all elements in the 9×9 array $R(I, J)$ centered in the current track point (DX, DY) as defined in Section 7.C.(7). The sum can then be used to calculate the mean value by dividing by 81. The computation that takes place can be expressed by

$$SUM = \sum_{i=1}^9 \sum_{j=1}^9 R_{i,j}$$

The mean value obtained from SUM is used in AGC calculations which control the gain and bias latches.

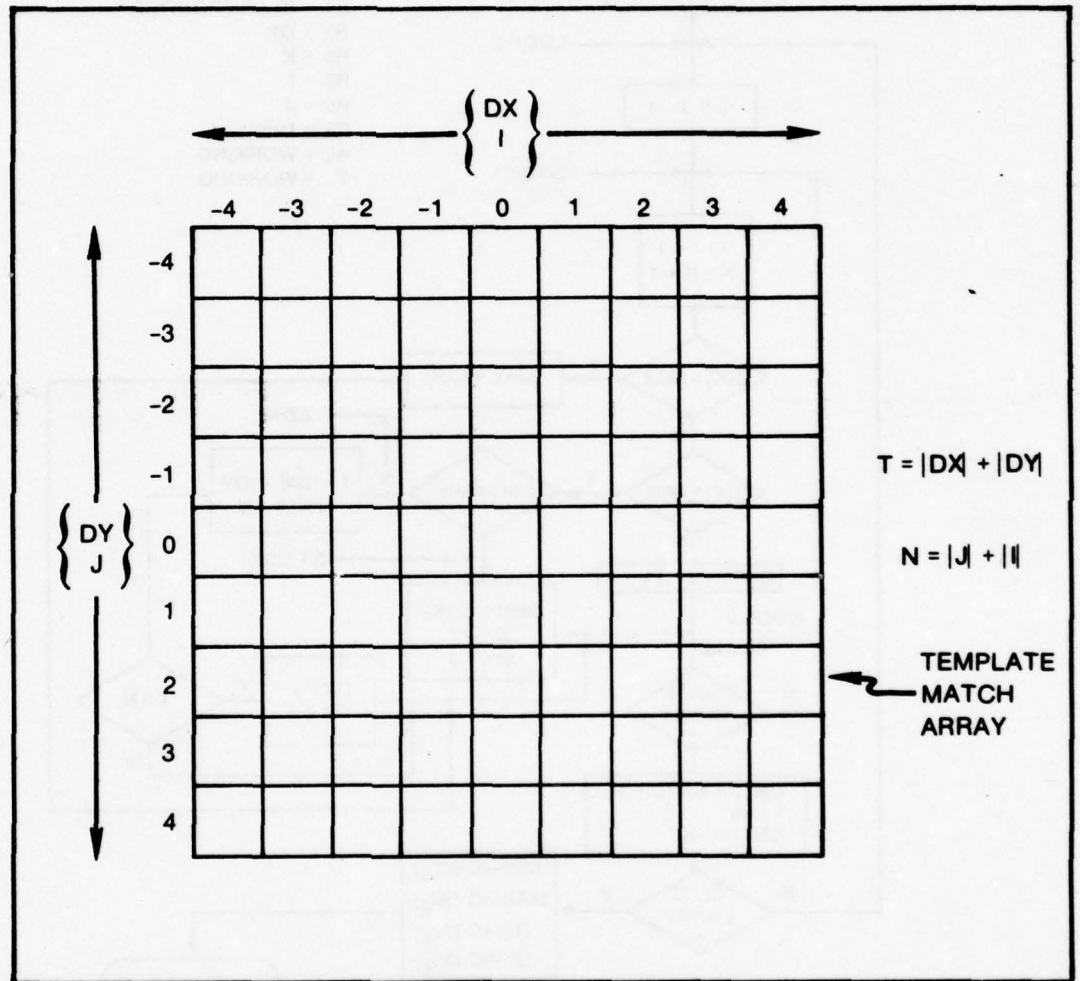


Figure 47. Variable definition for routine MIN.

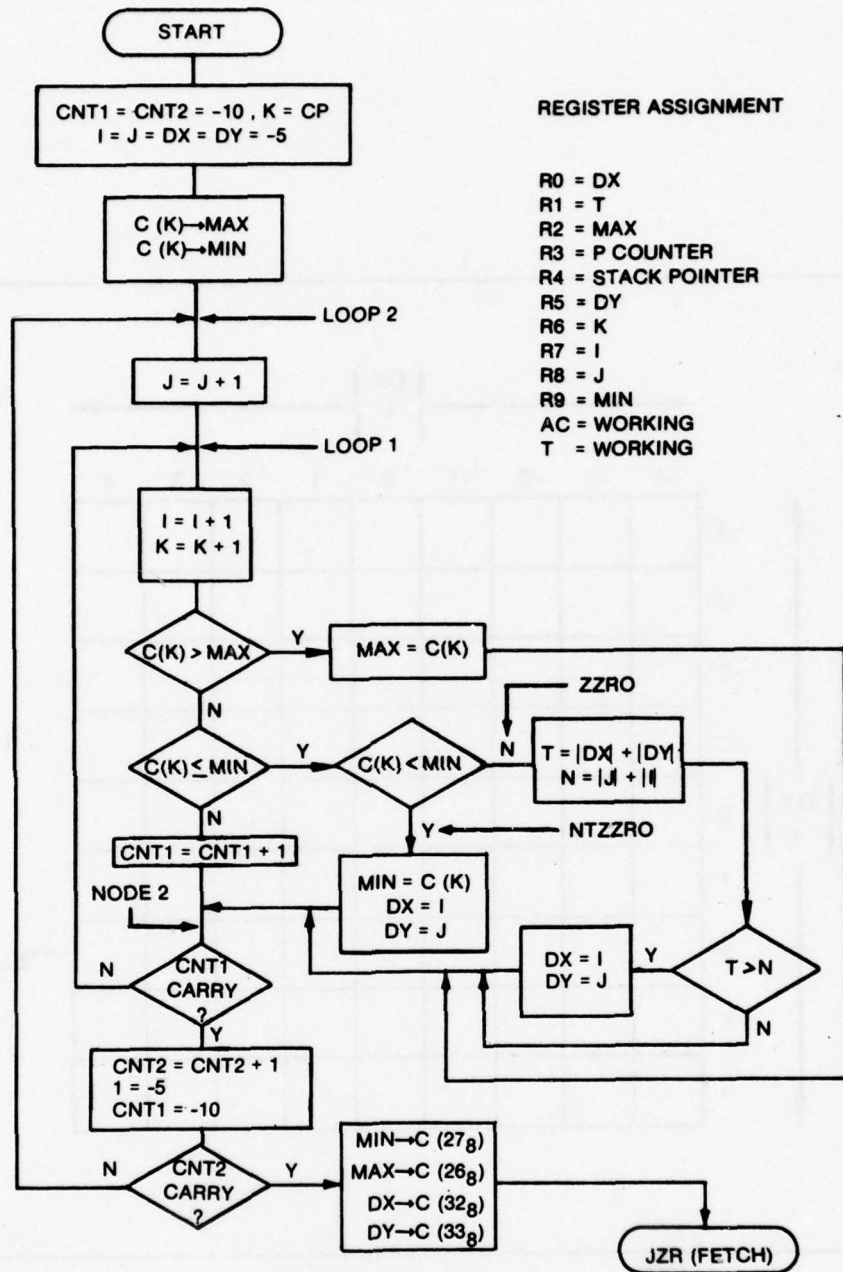


Figure 48. Flow chart for routine MIN.

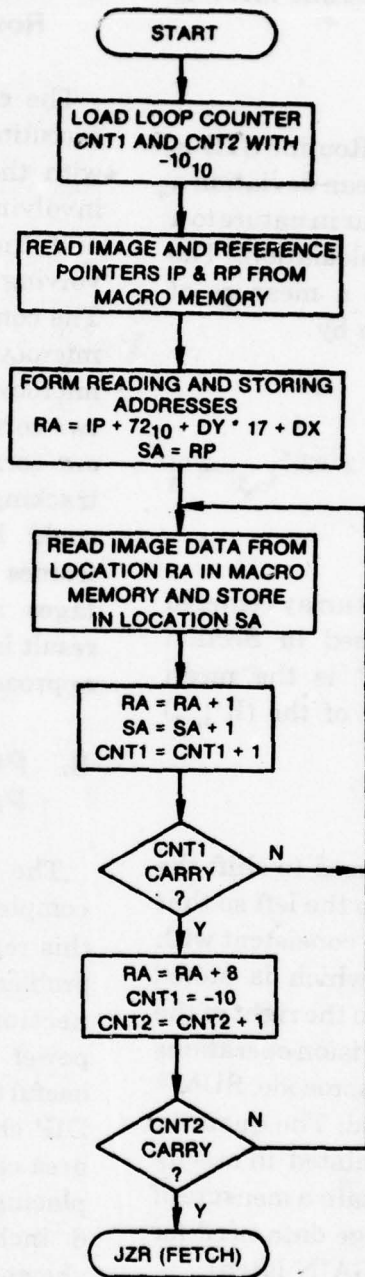


Figure 49. Flow chart for routine REF.

The flow chart for routine MNV is shown in Figure 50.

(10) Routine STD. Routine STD is used to calculate the mean deviation a quantity which is similar in nature to a standard deviation calculation. The quantity calculated is a measure of dispersion and is given by

$$\text{SUM} = 81 * \sigma_A = 4 \sum_{i=1}^9 \sum_{j=1}^9 | 2^5 * R'_{i,j} - \bar{R}' |$$

where $(R'_{i,j})$ is the 9×9 array centered at (DX, DY) as defined in Section 7.C.(7), and where \bar{R}' is the mean value of the elements of the $(R'_{i,j})$ array.

The constant 2^5 is used to shift the image data 5 places to the left so that the image data will be consistent with the mean value \bar{R}' which is stored with 5 binary places to the right of the binary point. Since division operations must take place in macrocode, $\text{SUM} = 81 * \sigma_A$ is calculated. The quantity $\sigma_A = \text{SUM}/81$ is calculated in macrocode and is used to obtain a measure of the dispersion of image data need for gain determination (GAIN latch).

Figure 51 shows a flow chart of the calculations performed in STD.

D. Execution Times for Tracker Routines

The chart in Figure 52 shows the execution time of each routine along with the number of microcycles not involving a read or write (μ cycles), and the number of microcycles involving a read or write ($R/W \mu$ cycles). The computations assume a 160 nsec microcycle and a 320 nsec read or write microcycle. It is of interest to note that two correlation tracking operations or one correlation and one adaptive gate tracking operation (simultaneously) could be accomplished at over 60 frames a second. Though the advantages are not clearly defined, this result is indicative of the power of the approach.

8. POWER CONSUMPTION PACKAGING

The packaging problem is a very complex one for the system given in this report, and it is clear that major problems will stem from interconnections of components and high power consumption. However, it is useful to consider chip area used by the DIP chips in the tracker. The needed area can be graphically illustrated by placing all tracker components on two 6 inch diameter circular areas as shown in Figures 53 and 54. The area analysis assumes the use of state of the art memory components available as of this writing and includes only those chips needed for tracker operation. No

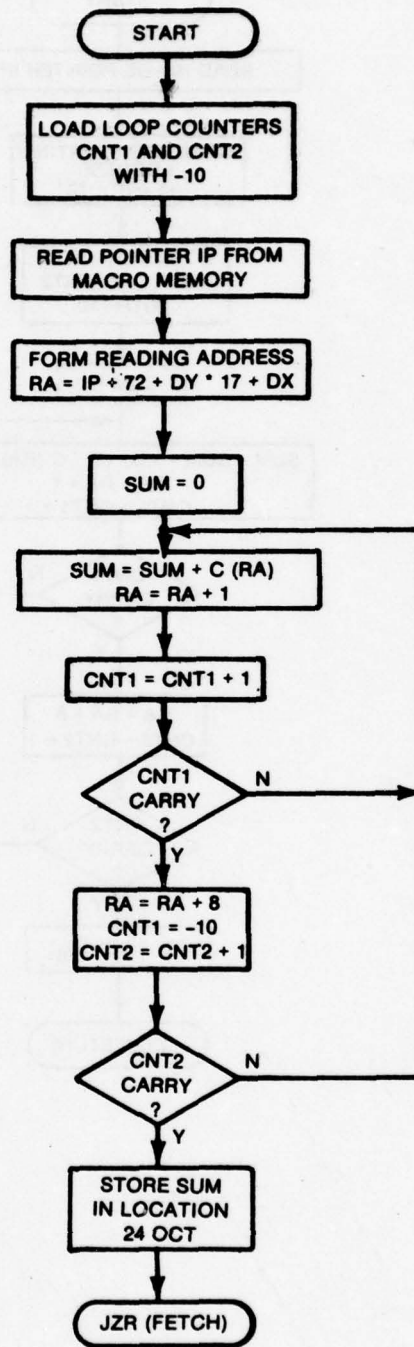


Figure 50. Flow chart for routine MNV.

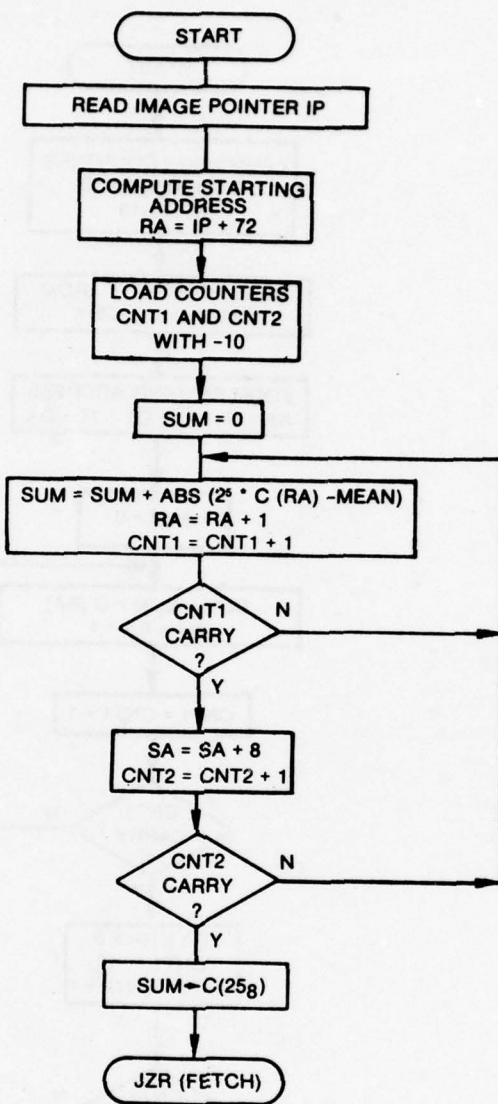


Figure 51. Flow chart for routine STD.

ROUTINE	CYCLE LESS R/W	READS (R)	WRITES (W)	EXECUTION TIME (m sec)
WIN	4	6	36	.0089
CROSS	4	6	21	.0066
FRM	1	2	13	.0030
IMG	85	86	291	.1013
COR	13,127	81	14,137	6.4884
MIN	245	324	3,399	.7259
REF	85	81	112	.0726
MNV	84	1	119	.0462
STD	85	1	1,168	.2144
WAT	0	0	5	.0000*
19,301	13,720	588	TOTAL	7.67

Figure 52. Estimated execution times of tracker routines.

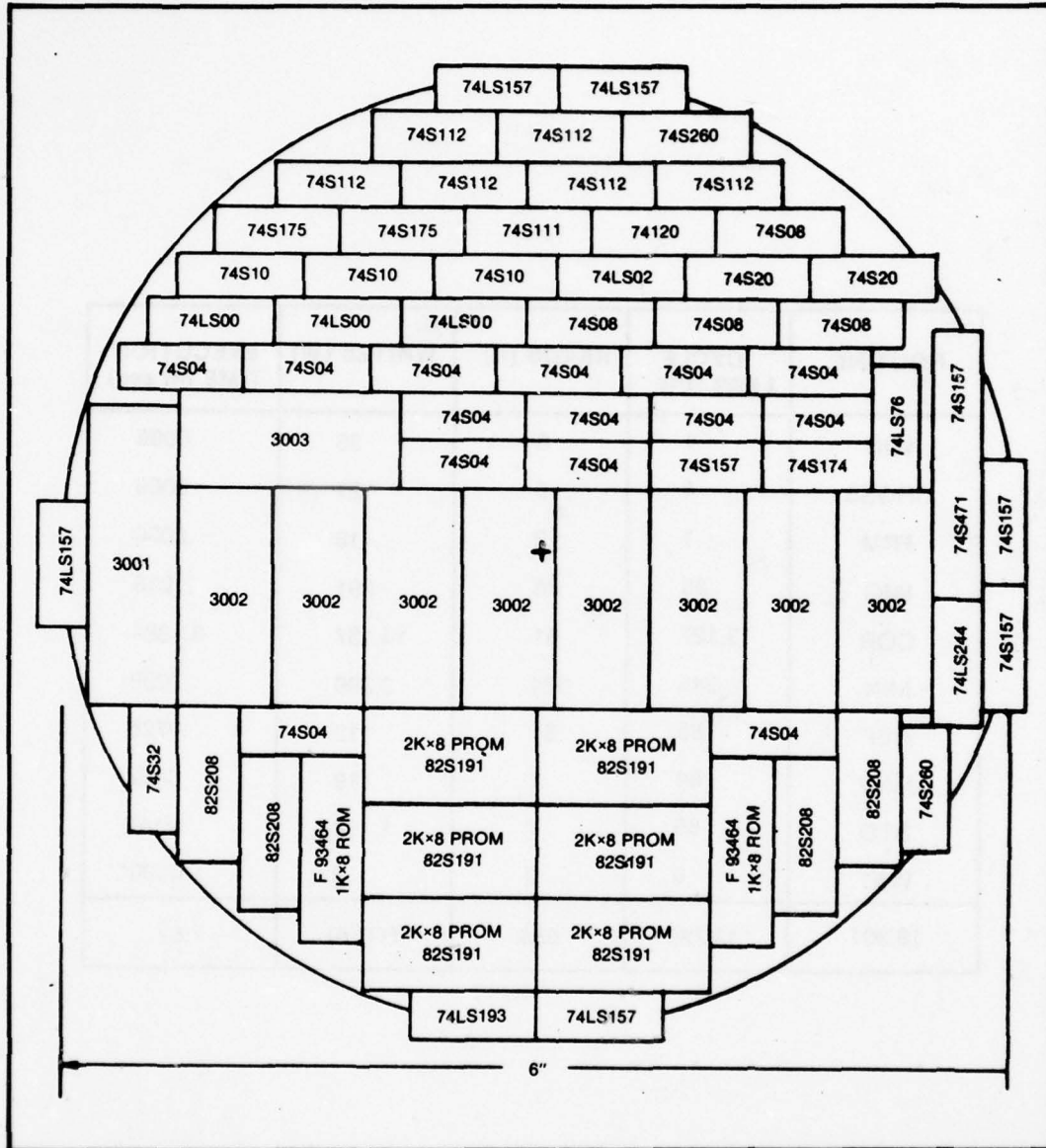


Figure 53. Graphic illustration of DIP area needed for tracker.

special attempts were made to reduce area, except in the case of memory, analog to digital conversion, video control latches, and operational amplifiers. The bulk of the chips shown actually exist in the current tracker hardware. It should be noted that an area reduction of 3:1 is possible using ceramic chip carrier packaging [6]. As an illustration of what this means consider the chip areas shown in Figures 53 and 54 would become 3.4 inch diameter circular areas. It appears that power consumption rather than package size is a more serious problem. The table shown in Figure 55 is an inventory of parts used in the tracker, along with power consumptions of individual parts. Only limited attempts were made to reduce power consumption. Shottky parts are used routinely throughout the tracker. It is felt that many paths and loading conditions would permit the use of more low power Shottky parts resulting in a considerable savings in power consumption. The power consumption of 39 watts typical, 56 watts maximum is high, but is not the "Hoover Dam Syndrome" expected.

9. CONCLUSIONS

This report contains a new concept for an imaging tracker using a high

speed microprocessor. The tracker, capable of adaptive gate centroid tracking and template matching tracking, achieved each type of tracking at frame rates up to 123 frames/second. Therefore, tracker concepts using simultaneous tracking of centroid and templates are possible at rates of 60 frames/second. It is felt that the concept of using a high speed microprocessor can eliminate much hardware associated with image processing trackers. Though power consumption is currently on the high side, the tracker hardware concept will easily fit in a 6 inch missile. The volume consumed could be less than a six (6) inch cylinder one inch in height, exclusive of power supply volume. It must be conceded however, that thermal conduction and interconnection problems could force a much larger package size.

The 1980's will usher in a new era with the advent of VLSI. The speed-power products of digital integrated circuits will plunge to new lows as the use of electron beam lithography becomes wide spread. Packaging will become more dense. All indications are that the tracker in this report could be built with smaller size, lower power consumption, and much higher performance in the 1980's.

NO CHIPS	PART	FUNCTION AND COMMENTS	TYP	MAX	NO PINS
8	3002	CENTRAL PROCESSING ELEMENT	5.800	7.600	28
1	3003	LOOK-AHEAD CARRY GENERATOR	.400	.650	28
1	3001	MICROPROGRAM CONTROL UNIT	.850	1.200	40
22	74S04	HEX INVERTER	3.3	5.940	14
5	74S174	HEX D FLIP-FLOP WITH CLEAR	2.250	3.600	16
1	74S11	DUAL JK FLIP-FLOP	.07	.1025	14
4	74S157	QUAD 2 TO 1 MULTIPLEXER	1.0	1.560	16
1	74LS244	OCTAL BUFFER WITH 3-STATE OUTPUTS	.135	.230	20
3	74LS00	POSITIVE NAND GATE	.036	.066	14
4	74S08	POSITIVE NAND GATE	.640	1.140	14
3	74S10	POSITIVE NAND GATE	.225	.405	14
1	74LS02	POSITIVE NOR GATE	.007	.013	14
2	74S20	POSITIVE NAND GATE	.100	.180	14
2	74S175	QUAD D FLIP-FLOP	.600	.960	16
1	74120	DUAL PULSE SYNCHRONIZER	.255	.450	16
6	74S112	DUAL JK FLIP-FLOP, CLEAR AND PRESENT	.450	.750	16
3	74S260	POSITIVE NOR GATE	.390	.675	14
5	74LS32	POSITIVE OR GATE	.122	.245	14
4	74LS76	DUAL JK FLIP-FLOP, CLEAR AND PRESET	.080	.160	16
1	74LS90	DECADE COUNTER	.045	.075	14
4	74LS93	BINARY COUNTER	.180	.780	14
1	8T26	TRANSCEIVER	.435	.435	16
2	74H21	POSITIVE NAND GATE	.200	.320	14
2	74123	RETRIGGERABLE MONOSTABLE	.460	.660	16
8	74LS85	4 bit MAGNITUDE COMPARITOR	.416	.800	16
2	74S11	POSITIVE AND GATE	.240	.420	14
7	74LS273	OCTAL D FLIP-FLOP WITH CLEAR	.595	.945	20
1	7442	4 TO 10 LINE DECODER	.140	.280	16
2	7430	POSITIVE NAND GATE	.030	.060	14
3	7432	POSITIVE OR GATE	.345	.570	14
3	74S471	256x8 PROM	1.65	2.325	20
4	82S208	256x8 RAM (SIGNETICS)	2.700	3.700	22
8	74LS163A	SYNCHRONOUS 4 bit COUNTERS	.760	1.280	16
2	74S138	DEMULTIPLEXER (3 TO 8)	.490	.740	16
6	82S191	2048x8 PROM (SIGNETICS) T _{ac} =80 NS	3.900	5.250	24
2	93464	1024x8 ROM FAIRCHILD T _{ac} =45 NS	1.100	1.500	24
1	DAC80	BURR BROWN 8 bit DAC	.900	.900	16
1	MDAC	COMPUTER LABS MULTIPLYING DAC 10 MHz	1.800	1.800	N/A
1	ADC-SH4B	ADC 4 bits WITH S/H	1.375	1.375	N/A
1	3554	OP AMP BURR BROWN	3.025	3.025	N/A
2	?	AMI 1024x4 VMOS RAM T _{AC} =55 NS	1.200	1.800	18
4	74LS193	SYNCHRONOUS 4 bit COUNTER WITH CLEAR	.380	.680	16
4	74LS157	QUAD 2 TO 1 MULTIPLEXER	.194	.320	16
1	74L154	4 TO 16 LINE DECODER	.085	.140	24
TOTALS	151		39.355	56.307	

Figure 55. Part inventory and power consumption.

REFERENCES

1. *Intel Series 3000 Reference Manual*, Publication 98, 221A, 1975 Intel Corp.
2. *Series 3000 Cross Microprogramming System CROMIS Reference Specification*. Intel Corp., 1975 Revision 4.
3. Duda, R. O. and Hart, P.E., *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973, New York.
4. Sebestyen, G. S. *Decision-Making Processes in Pattern Recognition*, Macmillan Company, New York, 1962.
5. Sims, S.R.F., *Optical Contrast Variable Gate Centroid Tracker for the Intel 3000 Microprocessor*, TE-77-13, June 1977.
6. Lyman, J., *Growing Pin Count Is Forcing LSI Package Changes*, Electronics, March 19, 1977, McGraw Hill, pages 81-90.

APPENDIX A

**Macrocode
(LISTING #1)**

*
 *
 * SSW 5 ON FOR CENTROID TRACKING
 *
 * SSW 10 ON FOR SWITCH TO CORRELATOR WHILE CENTROID TRACKER
 * IS ON
 *
 * SSW 10 OFF SWITCHES BACK TO CAGE MODE ON CENTROID TRACKER
 * IF SSW 5 IS OFF
 *
 * IF SENSE SWITCH 7 IS ON .. YOU WILL GO TO THE AGC SCAN
 * LOOP IN EITHER CENTROID OR CORRELATION
 *
 *
 *
 *

1:	00000	000002	100000		OCT	50
2:	00001	000004	000000		OCT	100
3:	00002	000001	000002	DELX	BSS	1
4:	00003	000001	000003	DELY	BSS	1
5:	00004	000001	000004	ASUM	BSS	1
6:	00005	000001	000005	MEAN2	BSS	1
7:	00006	000001	000006	XEEG	BSS	1
8:	00007	000001	000007	XAEG	BSS	1
9:	00010	000001	000010	XBLG2	BSS	1
10:	00011	000001	000011	XALG	BSS	1
11:	00012	000001	000012	YEEG	BSS	1
12:	00013	000001	000013	YAEG	BSS	1
13:	00014	000001	000014	YBLG2	BSS	1
14:	00015	000001	000015	YALG	BSS	1
15:	00016	000001	000016	XBAR	BSS	1
16:	00017	000001	000017	YBAR	BSS	1
17:	00020	000001	000020	BWF	BSS	1
					ABS	100
19:	00100	125400	000356	CAGE	START	INIT
20:	00101	000000	000367		LA	TEN
21:	00102	176400	060003		DO	6
22:	00103	124400	000000		WINDOW	
23:	00104	020000	000705		L	SAGC
24:	00105	000000	000752		LA	FIFTY
25:	00106	000400	000030		STA	NX
26:	00107	000400	000031		STA	NY
27:	00110	030000	000000		CLR	
28:	00111	000400	000032		STA	DELXX
29:	00112	000400	000033		STA	DELYY
30:	00113	000000	000401		LA	IPC
31:	00114	000400	000020		STA	IP
32:	00115	000000	000400		LA	C36
33:	00116	176400	060001		DO	6
34:	00117	020000	000771	AGCS	L	SCANN

35: 00120 077400 000000
 36: 00121 005400 000766
 37: 00122 006400 000000
 38: 00123 010000 000117
 39: 00124 000000 000037
 40: 00125 000400 000404
 41: 00126 000000 000036
 42: 00127 000400 000405
 43: 00130 125400 000356
 44: 00131 124400 000000
 45: 00132 077400 000000
 46: 00133 005400 000766
 47: 00134 006400 000000
 48: 00135 010000 000100
 49: 00136 020000 001000
 50: 00137 000000 000374
 51: 00140 000400 000346
 52: 00141 000400 000347
 53: 00142 000000 000400
 54: 00143 004400 000372
 55: 00144 176400 060000
 56: 00145 125000 000000
 57: 00146 000000 000377
 58: 00147 004400 000372
 59: 00150 176400 060000
 60: 00151 120400 000350
 61: 00152 174400 000000
 62: 00153 000000 000350
 63: 00154 032400 070000
 64: 00155 000400 000350
 65: 00156 030000 000000
 66: 00157 174400 000000
 67: 00160 003400 000350
 68: 00161 033000 140000
 69: 00162 006000 000366
 70: 00163 072000 000000
 71: 00164 010000 000171
 72: 00165 030000 000000
 73: 00166 000400 000372
 74: 00167 000400 000020
 75: 00170 010000 000175
 76: 00171 000000 000371
 77: 00172 000400 000020
 78: 00173 000000 000370
 79: 00174 000400 000372
 80: 00175 125400 000356
 81: 00176 000000 000364
 82: 00177 004400 000372
 83: 00200 176400 060000

CAGE1

BLACK

WHITE

FOR VIDEO
LATCH START

LAIBUS
 AND SSW7
 SZ
 J AGCS
 LA BIAS
 STA SBIAS
 LA GAIN
 STA SGAIN
 START INIT
 WINDOW
 LAIBUS
 AND SSW7
 SZ
 J CAGE
 L ZERR
 LA S62
 STA CX
 STA CY
 LA C36
 OR TARGET
 DO 6
 FRAMEW
 LA C33
 OR TARGET
 DO 6
 MEAN ECOUNT
 EQ
 LA ECOUNT
 ALS 7
 STA ECOUNT
 CLR
 EQ
 D ECOUNT
 LRS 14
 C SEVEN
 SGE
 J BLACK
 CLR
 STA TARGET
 STA BWF
 J WHITE
 LA OONE
 STA BWF
 LA TWENTY
 STA TARGET
 START INIT
 LA SIX
 OR TARGET
 DO 6

84:	00201	030000	000404	LA	SBIAS
85:	00202	176400	040000	DO	4
86:	00203	000000	000405	LA	SGAIN
87:	00204	176400	050000	DO	5
* 88:	00205	125000	000000	REQUEST FRAME AND PROCESS CORRELATOR AGC GAIN AND BIAS FRAMEW	
* 89:	00206	000000	000362	SET MODE LATCH FOR READ MODE	
90:	00207	004400	000372	LA	ONE
91:	00210	176400	060000	OR	TARGET
92:	00211	124400	000000	DO	6
93:	00212	077400	000000	WINDOW	
94:	00213	005400	000376	LAIBUS	
95:	00214	007000	000000	AND	SSW1
96:	00215	010000	000341	SNZ	
97:	00216	121000	000351	J	TT
98:	00217	120400	000350	SAVEIJ	I
99:	00220	174400	000000	MEAN	ECOUNT
100:	00221	000000	000350	EQ	
101:	00222	032400	070000	LA	ECOUNT
102:	00223	000400	000350	ALS	7
103:	00224	030000	000000	STA	ECOUNT
104:	00225	174400	000000	CLR	
105:	00226	003400	000350	EQ	
106:	00227	033000	140000	D	ECOUNT
107:	00230	000400	000005	LRS	14
108:	00231	121400	000351	STA	MEAN2
109:	00232	122000	000002	RESTIJ	I
110:	00233	000000	000344	EGATES	DELX
111:	00234	000000	000346	ADR	XG
112:	00235	000000	000347	ADR	CX
113:	00236	121000	000351	ADR	CY
114:	00237	120000	000007	SAVEIJ	I
115:	00240	000000	000004	CNTRD	XAEG
116:	00241	032400	070000	LA	ASUM
117:	00242	000400	000004	ALS	7
118:	00243	030000	000000	STA	ASUM
119:	00244	174400	000000	CLR	
120:	00245	000000	000016	EQ	
121:	00246	003400	000004	LA	XBAR
122:	00247	000400	001012	D	ASUM
123:	00250	033000	140000	STA	CXS
124:	00251	000400	000346	LRS	14
125:	00252	000000	000017	STA	CX
126:	00253	003400	000004	LA	YBAR
127:	00254	000400	001011	D	ASUM
128:	00255	033000	140000	STA	CYS
129:	00256	000400	000347	LRS	14
130:	00257	000000	001012	STA	CY
131:	00260	033000	110000	LA	CXS
				LRS	11

132: 00261 002400 001010
 133: 00262 020000 000406
 134: 00263 176400 070000
 135: 00264 000000 001011
 136: 00265 033000 110000
 137: 00266 002400 001010
 138: 00267 020000 000406
 139: 00270 176400 100000
 140: 00271 000000 000003
 141: 00272 003000 000344
 142: 00273 174400 000000
 143: 00274 000400 000353
 144: 00275 000000 000002
 145: 00276 003000 000345
 146: 00277 174400 000000
 147: 00300 000400 000354
 148: 00301 121400 000351
 149: 00302 122400 000353
 150: 00303 000000 000002
 151: 00304 174400 000000
 152: 00305 000000 000353
 153: 00306 032400 070000
 154: 00307 000400 000353
 155: 00310 030000 000000
 156: 00311 174400 000000
 157: 00312 003400 000353
 158: 00313 032000 140000
 159: 00314 123000 000002
 160: 00315 000000 000346
 161: 00316 123400 000354
 162: 00317 000000 000003
 163: 00320 174400 000000
 164: 00321 000000 000354
 165: 00322 032400 070000
 166: 00323 000400 000354
 167: 00324 030000 000000
 168: 00325 174400 000000
 169: 00326 003400 000354
 170: 00327 032000 140000
 171: 00330 124000 000003
 172: 00331 000000 000347
 173: 00332 077400 000000
 174: 00333 005400 000402
 175: 00334 007000 000000
 176: 00335 010000 000176
 177: 00336 000000 000401
 178: 00337 000400 000020
 179: 00340 010000 000430
 180: 00341 000000 000372
 181: 00342 004400 000367

S 0620
 L CONVET
 DO 7
 LA CYS
 LRS 11
 S 0620
 L CONVET
 DO 10
 LA DELY
 M XG
 EQ
 STA AXGH
 LA DELX
 M YG
 EQ
 STA AYGH
 RESTIJ I
 NGATEA AXGH
 ADR DELX
 EQ
 LA AXGH
 ALS 7
 STA AXGH
 CLR
 EQ
 D AXGH
 ARS 14
 NGATEB DELX
 ADR CX
 NGATEC AYGH
 ADR DELY
 EQ
 LA AYGH
 ALS 7
 STA AYGH
 CLR
 EQ
 D AYGH
 ARS 14
 NGATEL DELY
 ADR CY
 LAIBUS
 AND SSW10
 SNZ
 J START
 LA IPC
 STA IP
 J START2
 LA TARGET
 OR TEN

TT

182: 00343 010000 000131
 183: 00344 000001 000344
 184: 00345 000001 000345
 185: 00346 000001 000346
 186: 00347 000001 000347
 187: 00350 000001 000350
 188: 00351 000001 000351
 189: 00352 000001 000352
 190: 00353 000001 000353
 191: 00354 000001 000354
 192: 00355 000001 000355
 193: 00356 000003 010000
 194: 00357 000003 010000
 195: 00360 000003 030000
 196: 00361 000003 030000
 197: 00362 000000 010000
 198: 00363 000000 020000
 199: 00364 000000 060000
 200: 00365 000000 030000
 201: 00366 000000 070000
 202: 00367 000000 100000
 203: 00370 000001 000000
 204: 00371 177777 170000
 205: 00372 000001 000372
 206: 00373 060000 000000
 207: 00374 000003 020000
 208: 00375 077777 170000
 209: 00376 000001 000000
 210: 00377 000000 130000
 211: 00400 000000 160000
 212: 00401 000055 160000
 213: 00402 000040 000000
 214: 00403 000001 000403
 215: 00404 000001 000404
 216: 00405 000001 000405
 217: 00406 002000 000413
 218: 00407 031400 000000
 219: 00410 002400 000362
 220: 00411 074400 000000
 221: 00412 000003 020000
 222: 00413 000040 000000

J CAGE1
 XG BSS 1
 YG BSS 1
 CX BSS 1
 CY BSS 1
 ECOUNT BSS 1
 I BSS 1
 J BSS 1
 AXGH BSS 1
 AYGH BSS 1
 AG2 BSS 1
 INIT OCT 61
 OCT 61
 OCT 63
 OCT 63
 ONE OCT 1
 TWO OCT 2
 SIX OCT 6
 THREE OCT 3
 SEVEN OCT 7
 TEN OCT 10
 TWENTY OCT 20
 OONE OCT 377777
 TARGET BSS 1
 G OCT 1400000
 S62 OCT 62
 CONN OCT 1777777
 SSW1 OCT 20
 C33 OCT 13
 C36 OCT 16
 IPC OCT 1336
 SSW10 OCT 1000
 SBWF BSS 1
 SBIAS BSS 1
 SGAIN BSS 1
 CONVET A BIAX
 TCA
 S ONE
 RTN
 062 OCT 62
 BIAX OCT 1000

*
*
*
*
*

COMBINE BOTH TRACKERS.....

224: 00020 000055 160000 IP ABS 20 OCT 1336
 225: 00021 000050 140000 RP OCT 1214
 226: 00022 000043 120000 CP OCT 1072

227:	00023	000051	040000	IP2	OCT	1224
228:	00024	000001	000024	MEAN	BSS	1
229:	00025	000001	000025	STD	BSS	1
230:	00026	000001	000026	MAXI	BSS	1
231:	00027	000001	000027	MINI	BSS	1
232:	00030	000001	000030	NX	BSS	1
233:	00031	000001	000031	NY	BSS	1
234:	00032	000001	000032	DELXX	BSS	1
235:	00033	000001	000033	DELYY	BSS	1
236:	00034	000001	000034	MODE	BSS	1
237:	00035	000001	000035	BWC	BSS	1
238:	00036	000001	000036	GAIN	BSS	1
239:	00037	000001	000037	BIAS	BSS	1
240:	00040	177766	070000	M153	OCT	3777547
241:	00041	177766	170000	M145	OCT	3777557
					ABS	430
243:	00430	000000	000346	START2	LA	CX
244:	00431	000400	000031		STA	NY
245:	00432	000000	000347		LA	CY
246:	00433	000400	000030		STA	NX
247:	00434	000000	000755		LA	ZERO
248:	00435	000400	000032		STA	DELXX
249:	00436	000400	000033		STA	DELYY
250:	00437	000000	000362		LA	ONE
251:	00440	000400	000035		STA	BWC
*			CROSS			
252:	00441	162000	000000		OCT	3440000
*			BEGIN SCAN LOOP			
*			GO TO TRACK IF SSW7 OFF ELSE CONTINUE SCAN			
253:	00442	020000	000771	SCANNG	L	SCANN
254:	00443	077400	000000		LAIBUS	
255:	00444	005400	000766		AND	SSW7
256:	00445	006400	000000		SZ	
257:	00446	010000	000442		J	SCANNG
*			GO TO TRACK MODE			
*			START TRACKING, STORE AGC PARAMETERS			
258:	00447	166412	160000		REF	
259:	00450	000000	000763		LA MEANO	
260:	00451	002000	000727		A	TP5
261:	00452	000400	000760		STA	MNPK
262:	00453	002400	000730		S	T1P5
263:	00454	000400	000756		STA	MNSBPK
264:	00455	000000	000764		LA	STD0
265:	00456	002000	000734		A	TP5D
266:	00457	000400	000761		STA	SDPK
267:	00460	006000	000741		C	T13P5
268:	00461	072400	000000		SLE	
269:	00462	000000	000731		LA	T14P0
270:	00463	002400	000730		S	T1P5
271:	02464	000400	000757		STA	SDSBPK

272:	00465	010000	000473		J	OVER
273:	00466	164400	000000	LOOP	WAT	
274:	00467	077400	000000		LAIBUS	
275:	00470	005400	000402		AND	SSW10
276:	00471	007000	000000		SNZ	
277:	00472	010000	000100		J	CAGE
278:	00473	165372	120000	OVER	IMG	
279:	00474	000000	000037		LA	BIAS
280:	00475	176400	040000		DO	4
281:	00476	000000	000036		LA	GAIN
282:	00477	176400	050000		DO	5
283:	00500	164000	000000		FRM	
284:	00501	167000	000000		COR	
285:	00502	163024	160000		MIN	
* MOVE WINDOW TO NEW TRACKING POSITION						
286:	00503	000000	000032		LA	DELXX
287:	00504	002000	000030		A	NX
288:	00505	000400	000030		STA	NX
289:	00506	000000	000033		LA	DELYY
290:	00507	002000	000031		A	NY
291:	00510	000400	000031		STA	NY
* BOUND ON NX AND NY						
292:	00511	000000	000030		LA	NX
293:	00512	006000	000732		C	N10
294:	00513	072400	000000		SLE	
295:	00514	010000	000520		J	NCK
296:	00515	000000	000732		LA	N10
297:	00516	000400	000030		STA	NX
298:	00517	010000	000525		J	OUTH
299:	00520	006000	000733	NCK	C	N90
300:	00521	072000	000000		SGE	
301:	00522	010000	000525		J	OUTH
302:	00523	000000	000733		LA	N90
303:	00524	000400	000030		STA	NX
304:	00525	000000	000031	OUTH	LA	NY
305:	00526	006000	000732		C	N10
306:	00527	072400	000000		SLE	
307:	00530	010000	000534		J	NCK2
308:	00531	000000	000732		LA	N10
309:	00532	000400	000031		STA	NY
310:	00533	010000	000541		J	OUTH2
311:	00534	006000	000733	NCK2	C	N90
312:	00535	072000	000000		SGE	
313:	00536	010000	000541		J	OUTH2
314:	00537	000000	000733		LA	N90
315:	00540	000400	000031		STA	NY
316:	00541	163400	000000	OUTH2	WIN	
317:	00542	000000	000030		LA	NX
318:	00543	002400	000412		S	062
319:	00544	032400	030000		ALS	3

320: 00545 020000 000406
 321: 00546 176400 100000
 322: 00547 000000 000031
 323: 00550 022400 000412
 324: 00551 032400 030000
 325: 00552 020000 000406
 326: 00553 176400 070000
 327: 00554 020000 000564
 328: 00555 077400 000000
 329: 00556 005400 000766
 330: 00557 007000 000000
 331: 00560 010000 000563
 332: 00561 164400 000000
 333: 00562 010000 000430
 334: 00563 010000 000466

* SUBROUTINE AGC FOLLOWS
 * MNV

335: 00564 166000 000000
 336: 00565 000000 000024
 337: 00566 003400 000744
 338: 00567 032000 050000
 339: 00570 000400 000024
 340: 00571 161400 000000
 341: 00572 000000 000025
 342: 00573 003400 000745
 343: 00574 032000 020000
 344: 00575 000400 000025

* FILTER STANDARD DEVIATION

345: 00576 000000 000764
 346: 00577 003000 000750
 347: 00600 000400 000762
 348: 00601 000000 000025
 349: 00602 003000 000747
 350: 00603 002000 000762
 351: 00604 000400 000764
 352: 00605 006000 000757
 353: 00606 072000 000000
 354: 00607 010000 000622
 355: 00610 006000 000761
 356: 00611 072000 000000
 357: 00612 010000 000647
 358: 00613 000000 000036
 359: 00614 003000 000751
 360: 00615 000400 000036
 361: 00616 000000 000037
 362: 00617 003000 000751
 363: 00620 000400 000037
 364: 00621 010000 000647
 365: 00622 000000 000036
 366: 00623 006000 000753

L CONVET
 DO 10
 LA NY
 S 062
 ALS 3
 L CONVET
 DO 7
 L AGC
 LAIBUS
 AND SSW7
 SNZ
 J 000V
 WAT
 J START2
 J LOOP

000V

AGC

OCT 3540000
 LA MEAN
 D T81
 ARS 5
 STA MEAN
 OCT 3430000
 LA STD
 D T81PRM
 ARS 2
 STA STD

LA STDO
 M C2
 STA TEMP
 LA STD
 M ONEMC2
 A TEMP
 STA STDO
 C SDSBPK

SGE
 J INCRE
 C SDPK
 SGE
 J NOCHG
 LA GAIN
 M GP933
 STA GAIN
 LA BIAS
 M GP933
 STA BIAS
 J NOCHG
 LA GAIN
 C FULLG

INCRE

AD-A071 638

ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND REDSTO--ETC F/G 9/2
APPLICATION OF A MICROPROGRAMMED, BIT SLICE MICROPROCESSOR TO T--ETC(U)
SEP 78 L G MINOR
DRDMI-T-78-53

UNCLASSIFIED

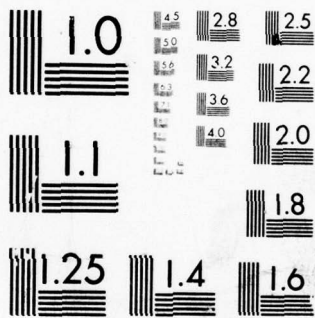
NL

2 of 3

AD
A071638



						V03E02E							
						V03E01E							
									V03E03E				



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

367: 00624 071000 000000
 368: 00625 010000 000647
 369: 00626 003400 000751
 370: 00627 006000 000753
 371: 00630 072400 000000
 372: 00631 010000 000637
 373: 00632 000400 000036
 374: 00633 000000 000037
 375: 00634 003400 000751
 376: 00635 000400 000037
 377: 00636 010000 000647
 378: 00637 000000 000036
 379: 00640 003400 000753
 380: 00641 000400 000762
 381: 00642 000000 000037
 382: 00643 003400 000762
 383: 00644 000400 000037
 384: 00645 000000 000753
 385: 00646 000400 000036
 386: 00647 037000 000000
 * FILTER MEAN VALUE
 387: 00650 000000 000763
 388: 00651 003000 000765
 389: 00652 000400 000762
 390: 00653 000000 000024
 391: 00654 003000 000767
 392: 00655 002000 000762
 393: 00656 000400 000763
 394: 00657 006000 000760
 395: 00660 072000 000000
 396: 00661 010000 000666
 397: 00662 000000 000037
 398: 00663 002000 000746
 399: 00664 000400 000037
 400: 00665 010000 000674
 401: 00666 006000 000756
 402: 00667 072400 000000
 403: 00670 010000 000674
 404: 00671 000000 000037
 405: 00672 002400 000746
 406: 00673 000400 000037
 407: 00674 000000 000037
 408: 00675 006000 000755
 409: 00676 072000 000000
 410: 00677 030000 000000
 411: 00700 006000 000754
 412: 00701 072400 000000
 413: 00702 000000 000754
 414: 00703 000400 000037
 415: 00704 074400 000000

SL
 J NOCHG
 D GP933
 C FULLG
 SLE
 J FULL
 STA GAIN
 LA BIAS
 D GP933
 STA BIAS
 J NOCHG
 LA GAIN
 FULL
 D FULLG
 STA TEMP
 LA BIAS
 D TEMP
 STA BIAS
 LA FULLG
 STA GAIN
 NOCHG
 NOP
 LA MEANO
 M C1
 STA TEMP
 LA MEAN
 M ONEMC1
 A TEMP
 STA MEANO
 C MNPk
 SGE
 J NEXT
 LA BIAS
 A ONEONE
 STA BIAS
 J NEXT2
 NEXT
 C MNSBPK
 SLE
 J NEXT2
 LA BIAS
 S ONEONE
 STA BIAS
 NEXT2
 LA BIAS
 C ZERO
 SGE
 CLR
 C FULLB
 SLE
 LA FULLB
 STA BIAS
 RTN

416:	00705	037000	000000	SAGC	NOP	
417:	00706	000000	000736		LA T7P0	
418:	00707	000400	000756		STA MNSBPK	
419:	00710	000400	000763		STA MEANO	
420:	00711	000000	000740		LA T14P5	
421:	00712	000400	000761		STA SDPK	
422:	00713	000400	000764		STA STDO	
423:	00714	000000	000737		LA T8P0	
424:	00715	000400	000760		STA MNPk	
425:	00716	000000	000741		LA T13P5	
426:	00717	000400	000757		STA SDSBPK	
* 427:	00720	000000	000742	INITIALIZE GAIN TO APPROXIMATELY UNITY	LA T31	
428:	00721	000400	000036		STA GAIN	
429:	00722	176400	050000		DO 5	
* 430:	00723	000000	000743	INITIALIZE BIAS TO APPROXIMATELY BLACK LEVEL	LA DCBLK	
431:	00724	000400	000037		STA BIAS	
432:	00725	176400	040000		DO 4	
433:	00726	074400	000000		RTN	
434:	00727	000001	000000	TP5	OCT	20
435:	00730	000003	000000	T1P5	OCT	60
436:	00731	001600	000000	T14P0	OCT	34000
437:	00732	000000	120000	N10	OCT	12
438:	00733	000005	120000	N90	OCT	132
439:	00734	000040	000000	TP5D	OCT	1000
440:	00735	000140	000000	T1P5D	OCT	3000
441:	00736	000016	000000	T7P0	OCT	340
442:	00737	000020	000000	T8P0	OCT	400
443:	00740	001640	000000	T14P5	OCT	35000
444:	00741	001540	000000	T13P5	OCT	33000
445:	00742	000001	110000	T31	OCT	31
446:	00743	000000	060000	DCBLK	OCT	6
447:	00744	005040	000000	T81	OCT	121000
448:	00745	050400	000000	T81PRM	OCT	1210000
449:	00746	000000	010000	ONEONE	OCT	1
450:	00747	040000	000000	ONEMC2	OCT	1000000
451:	00750	040000	000000	C2	OCT	1000000
452:	00751	075660	000000	GP933	OCT	1735400
453:	00752	000003	020000	FIFTY	OCT	62
454:	00753	000017	170000	FULLG	OCT	377
455:	00754	000017	170000	FULLB	OCT	377
456:	00755	000000	000000	ZERO	OCT	0
457:	00756	000016	000000	MNSBPK	OCT	340
458:	00757	001540	000000	SDSBPK	OCT	33000
459:	00760	000020	000000	MNPk	OCT	400
460:	00761	001640	000000	SDPK	OCT	35000
461:	00762	000001	000762	TEMP	BSS	1
462:	00763	000001	000763	MEANO	BSS	1
463:	00764	000001	000764	STDO	BSS	1

464:	00765	060000	000000	C1	OCT	1400000
465:	00766	000004	000000	SSK7	OCT	100
466:	00767	020000	000000	ONEMC1	OCT	400000
467:	00770	000001	160000	FRMRQ	OCT	36
468:	00771	164000	000000	SCANN	FRM	
469:	00772	164400	000000		WAT	
470:	00773	165006	170000		IMG	
471:	00774	020000	000564		L	AGC
472:	00775	000000	000037		LA	BIAS
473:	00776	176400	040000		DO	4
474:	00777	000000	000036		LA	GAIN
475:	01000	176400	050000		DO	5
476:	01001	020000	001003		L	ZERR
477:	01002	074400	000000		RTN	
478:	01003	000000	001007	ZERR	LA	C777
479:	01004	176400	070000		DO	7
480:	01005	176400	100000		DO	10
481:	01006	074400	000000		RTN	
482:	01007	000037	170000	C777	OCT	777
483:	01010	000031	000000	0620	OCT	620
484:	01011	000001	001011	CYS	BSS	1
485:	01012	000001	001012	CXS	BSS	1

APPENDIX B

Correlation Tracker Microcode (Page 2)

(LISTING #2)

Correlation Tracker Microcode (Page 3)

(LISTING #3)

XMAS VERS 2.0

ERRORS= 0 PAGE 1

SLISTFILE=2
 \$BITS
 \$CROSSRFF
 \$IMAGE
 \$WIDTH=132
 \$TITLE=CORRELATION MICROCODE
 \$DISPLAY(ALL)
 \$BITS=1
 \$LEFT=1
 \$RIGHT=72
 \$TITLE=CORRELATION MICROCODE
 \$WIDTH=132

FORMS=0
 IMAGE=1
 LINES=60
 PRINT=1
 SOURCEFILE=2
 MICROMEMORY=512

XMAS VERS 2.0 CORRELATION MICROCODE

ERRORS= 0 PAGE 2

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	PAUSEF	STRB	LPCNT	EMIT	MULTIPL	SPRF
1	6543210	10	10	6543210	76543210	10	210	0	43210	76543210	0	0
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												

MICRUPS(K00000=00H K00001=01H K00002=02H
 K00003=03H K00004=04H
 K00005=05H K00006=06H
 K00007=07H K00008=08H
 K00009=09H K0000A=0AH
 K0000B=0BH K0000C=0CH
 K0000D=0DH K0000E=0EH
 K0000F=0FH K00010=10H
 K00011=11H K00012=12H
 K00013=13H K00014=14H
 K00015=15H K00016=16H
 K00017=17H K00018=18H
 K00019=19H K0001A=1AH

```

16 K0001R=1BH K0001C=1CH
17 K0001D=1DH K0001E=1EH
18 K0001F=1FH K01FFF=1FH
19 K0000=00H K7FFFF=7FH
20 KFFFF=177Q KFFFFA=0FAH;
21
22 KBUSS
23 KRAUS;
24
25 FIELD LENGTH=2
26 MICROPS(PAGE1)=00R PAGE2=01R
27 PAGE3=10B PAGE4=11B;
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

```

K0001R=1BH K0001C=1CH
K0001D=1DH K0001E=1EH
K0001F=1FH K01FFF=1FH
K0000=00H K7FFFF=7FH
KFFFF=177Q KFFFFA=0FAH;

```

```

KBUSS
KRAUS;
FIELD LENGTH=2
MICROPS(PAGE1)=00R PAGE2=01R
PAGE3=10B PAGE4=11B;

```

```

CRUS
FIELD LENGTH=3
MICROPS(NR0)=00B INH=001R PMW=010B C'N3=011R
RTN=100B ROT=101R RRM=110R R'N4=111R);

```

```

PAUSEF
FIELD LENGTH=1
MICROPS(PAUSE=0);

```

```

STRR
FIELD LENGTH=1
MICROPS(STROBE=1R);
LPCNT
FIELD LENGTH=5
MICROPS(C0=00010R LCNT1=00100R LCNT2=01000R
LCNT3=01100B LCNT4=10000B ITCNT1=00111B
ITCNT2=01011R ITCNT3=01111R ITCNT4=10011B);

```

```

EMIT
FIELD LENGTH=8
MICROPS ( M1=0FFH M2=0FEH M3=0FDH M4=0FCH
M5=0FBH M6=0FAH M7=0F9H M8=0F84 M9=0F7H M10=0F6H
M11=0F5H M12=0F4H M13=0F3H M14=0F2H M15=0F1H
M16=0F0H M17=0EFH M18=0EEH);

```

```

MULTIPLY FIELD LENGTH=1
MICROPS (MULT=1R);

```

```

SPRF
FIELD LENGTH=1
MICROPS(SPARE=01H);

```

RECORD NUMBER	CPF	FI	FO	.JUMP	KRUSS	PAGEF	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
55	6543210	10	10	6543210	76543210	10	0	0	43210	76543210	0	0
56												
57	1101101	11	00	1100111	00000000	01	000	1	00010	00000000	0	0
	NOP(AC) .JPR(NOT1,NOT2,NOT3,STD,CROSS,NOT4,MIN,WTN,FRM, MAT,IMG,NOT12,MNV,REF,CORR,NOT16);											
58	1101101	11	00	0101111	00000000	00	000	1	00010	00000000	0	0
	NOT1: NOP(AC) PAGE1 JZR(FETCH);											
59	1101101	11	00	0101111	00000000	00	000	1	00010	00000000	0	0
	NOT2: NOP(AC) PAGE1 JZR(FETCH);											
60	1101101	11	00	0101111	00000000	00	000	1	00010	00000000	0	0
	NOT3: NOP(AC) PAGE1 JZR(FETCH);											
61	1101101	11	00	0101111	00000000	00	000	1	00010	00000000	0	0
	NOT4: NOP(AC) PAGE1 JZR(FETCH);											
62	/* ROUTINE WIN MIN MIN MIN MIN MIN MIN MIN MIN MIN */											
63												
64												
65												
66												
67												
68												
69												
70	1101101	11	00	0110110	00000000	10	000	1	00010	00000000	0	0
	WIN: NOP(AC) PAGE3 JCR(MIN);											

```

71 07RH1  N0T12I N0P(AC) PAGE1 JZR(FFTCH):
(007RH) 1101101 11 00 0101111 00000000 00 000 1 0 00010 00000000 0 0 0

72
73
74
75
76 /* ROUTINE CORR CORR CORR CORR CORR CORR CORR */
77
78
79 07EM1  CORR1 N0P(AC) PAGE3 JCR(CORR):
(007EH) 1101101 11 00 0111110 00000000 10 000 1 0 00010 00000000 0 0 0

80

81 07FM1  N0T16I N0P(AC) PAGE1 JZR(FFTCH):
(007FH) 1101101 11 00 0101111 00000000 00 000 1 0 00010 00000000 0 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSEFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	010	0	0	43210	76543210	0	0

82													
83													
84													
85													
86													
87													
88													
89													
90													
91													
92													
93													
94													

/* IMG IMG IMG IMG IMG IMG IMG IMG IMG IMG IMG IMG /*

/* R7=93*/

/*RR=SA /*

/*RR=STA /*

/* FORM STARTING ADDRESS /*

07AH! IMG! CLR(AC) !

(07AH) 1001101 11 00 0001000 00000000 01 000 1 0 00010 00000000 0 0

09AH! LMI(AC) K00019 RRM !

(09AH) 0011101 11 00 0111111 00011001 01 110 1 0 00010 00000000 0 0

09FH! ACM(T) ! /*T=NY /*

(09FH) 0001010 11 00 0001001 00000000 01 000 1 0 00010 00000000 0 0

09FH! CLR(AC) !

(09FH) 1001101 11 00 0001010 00000000 01 000 1 0 00010 00000000 0 0

0AFH! LMI(AC) K00009 !

(0AFH) 0011101 11 00 0001011 00011001 01 000 1 0 00010 00000000 0 0

99	08FH (008FH)	1111101	11 00 0001100	00000000	01 000	1	0	00010	00000000	0	0
				CMR(AC) :							
100	0CFH (00CFH)	0001100	11 11 0001101	11111111 01	000	1	0	00010	00000000	0	0
				ALR(T) FFI ; /*NY-9 TO T, AC */							
101	0DFH (00DFH)	1001100	11 00 0001110	00000000	01 000	1	0	00010	00000000	0	0
				CLR(T) :							
102	0EFH (00EFH)	0111101	11 00 0001111	11111111 01	000	1	0	00010	00000000	0	0
				ADR(AC) ; /*2(NY-9) TO AC */							
103	0FFH (00FFH)	0111101	11 00 0010000	11111111 01	000	1	0	00010	00000000	0	0
				ADR(AC) ; /* 4(NY-9) TO AC */							
104	10FH (010FH)	0111100	11 00 0010001	11111111 01	000	1	0	00010	00000000	0	0
				ADR(T) ; /*AC TO T */							

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
105	11FH (01FH)	0111101	11	00	0010010	11111111	01	000	1	0	00010	00000000	0
106	12FH (012FH)	0111101	11	00	0010011	11111111	01	000	1	0	00010	00000000	0
107	13FH (013FH)	0111101	11	00	0010100	11111111	01	000	1	0	00010	00000000	0
108	14FH (014FH)	0111100	11	00	0010101	11111111	01	000	1	0	00010	00000000	0
109	15FH (015FH)	0111101	11	00	0010110	11111111	01	000	1	0	00010	00000000	0
110	16FH (016FH)	0001100	11	00	0010111	11111111	01	000	1	0	00010	00000000	0
111	17FH (017FH)	1001100	11	00	0011000	00000000	01	000	1	0	00010	00000000	0
112	18FH (018FH)	0011100	11	00	0011001	00011000	01	110	1	0	00010	00000000	0

113	19FH (019FH)	0001011	AMA(AC) ; /*100(NY-9) *NX TO AC */ 11 00 0011010 1111111 01 000	1	0	00010	00000000	0	0
114	1AFH (01AFH)	1001100	CLR(T) ; 11 00 0000001 00000000 01 000	1	0	00010	00000000	0	0
115	01FH (001FH)	0011100	LMI(T) K0000 ; 11 00 0000010 00001001 01 000	1	0	00010	00000000	0	0
116	02FH (002FH)	1111100	CMR(T) ; 11 00 0000011 00000000 01 000	1	0	00010	00000000	0	0
117	03FH (003FH)	0001100	ALR(T) FFI ; /*100(NY-9) *NX-9 TO AC.T */ 11 11 0000100 1111111 01 000	1	0	00010	00000000	0	0
118	04FH (004FH)	0101000	SDR(RB) FFI ; /*SA=AC */ 11 11 0000101 1111111 01 000	1	0	00010	00000000	0	0

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CAUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
119	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
120	/*FORM CONSTANT P7=83 AND LOAD COUNTERS */												
	05FH!	11	00	0000110	00000000	01	000	1	0	00010	00000000	0	0
	(005FH)	11	00	0000110	00000000	01	000	1	0	00010	00000000	0	0
121	04FH!	11	00	0111110	00011111	01	000	1	0	00100	11101110	0	0
	(004FH)	11	00	0111110	00011111	01	000	1	0	00100	11101110	0	0
122	06FH!	11	00	0000001	00011111	01	000	1	0	01000	11101110	0	0
	(006FH)	11	00	0000001	00011111	01	000	1	0	01000	11101110	0	0
123	01FH!	11	00	0000010	00010101	01	000	1	0	00010	00000000	0	0
	(001FH)	11	00	0000010	00010101	01	000	1	0	00010	00000000	0	0
124	/* FETCH IMAGE POINTER */												
125	02FH!	11	00	0000011	00000000	01	000	1	0	00010	00000000	0	0
	(002FH)	11	00	0000011	00000000	01	000	1	0	00010	00000000	0	0
126	03FH!	11	00	0111101	00010000	01	110	1	0	00010	00000000	0	0
	(003FH)	11	00	0111101	00010000	01	110	1	0	00010	00000000	0	0
127	03DH!	11	00	0000001	00000000	01	000	1	0	00010	00000000	0	0
	(003DH)	11	00	0000001	00000000	01	000	1	0	00010	00000000	0	0

```

129 01DH! SDR(RS) FFI ! /*RQ=IP=STA */
(001DH) 0101001 11 11 0000010 11111111 01 000 1 0 00010 00000000 0 0 0
/* GO TO READ MODE */
130 /* UPDATE MODE STATUS WORD */
131 02DH! CLR(T) ;
(002DH) 1001100 11 00 0000100 00000000 01 000 1 0 00010 00000000 0 0 0
132 04DH! LMI(T) K0001C RRM ;
(004DH) 0011100 11 00 0000101 00011100 01 110 1 0 00010 00000000 0 0 0
133 05DH! ACM(AC) ;
(005DH) 0001011 11 00 0000110 00000000 01 000 1 0 00010 00000000 0 0 0
134 06DH! CLR(T) ;
(006DH) 1001100 11 00 0011111 00000000 01 000 1 0 00010 00000000 0 0 0
135 1FDH! INR(T) FFI !
(01FDH) 0111100 11 11 0001000 00000000 01 000 1 0 00010 00000000 0 0 0

```

XMAS VOPS 2.0 CORRELATION MICROCODE

RECORD NUMBER	OPF	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPDF
136	03DH (03EDH)	10	10	6543210	74543210	10	210	0	0	43210	74543210	0	0
		11	00	0001001	11111111	01	000	1	0	00010	00000000	0	0
		OPD(T) :											
137	09DH (09EDH)	11	00	0001110	00000000	01	000	1	0	00010	00000000	0	0
		ILR(T) :											
138	0EDH (09EDH)	11	00	0001111	00000000	01	000	1	0	00010	00000000	0	0
		CLR(T) :											
139	0FDH (09EDH)	11	00	0010001	00011100	01	111	1	0	00010	00000000	0	0
		LMI(T) K0001C RMV :											
140	11DH (011DH)	11	00	0010010	00000000	01	000	1	0	00010	00000000	0	0
		CLR(T) :											
141	12DH (012DH)	11	00	0111010	00000110	01	101	1	0	00010	00000000	0	0
		LMI(T) K00006 ROT :											
142	/* BEGIN LOOPS */												
143	12AH (012AH)	11	11	0010011	00000000	01	100	1	0	00010	00000000	0	0
		LPI: LMI(0A) FFI RIN :											
144	13AH (013AH)	11	11	0111100	00001111	01	000	1	0	00111	00000000	0	0
		LPIX: LMI(AC) K0000F ITCNTI FFI :											

```

145 13CH: LMI(R9) FFI RMN JFL(LP1.CT1) ;
      (013CH) 0011001 11 11 1000010 0000000 01 111 1 0 00010 00000000 0 0
146 12RH: ILP(R7) ITCNT2 ;
      (012RH) 0000111 11 00 0111100 00000000 01 000 1 0 01011 00000000 0 0
147 12CH: ADP(R8) LCNT1 M19 JFL(LP2.CT2) & /*SA=SA+R7 */
      (012CH) 0111000 11 00 1000101 1111111 01 000 1 0 00100 11101110 0 0
148 15AH: LP2: LMI(R8) FFI RIN MP(LP1X) ;
      (015AH) 0011000 11 11 0010011 00000000 01 100 1 0 00010 00000000 0 0
149 /* GO TO STORE MODE WITH NO RFO */
150 15RH: CT2: CLD(AC) ;
      (015RH) 1001101 11 00 0111100 00000000 01 000 1 0 00010 00000000 0 0

```

RECORD NUMBER	CPF	FT	FD	.JIMP	KRUSS	PAGEF	CAUS	PAUSFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
151	6543210 (15CH)	10	10	6543210	74543210	10	210	0	0	43210	74543210	0	0
	LMI(AC)	11	00	0010110	00011100	01	110	1	0	00010	00000000	0	0
152	15CH1 (15CH)	11	00	0010111	00000000	01	000	1	0	00010	00000000	0	0
	ACM(AC)	11	00	0010111	00000000	01	000	1	0	00010	00000000	0	0
153	17CH1 (17CH)	11	00	0011000	00000000	01	000	1	0	00010	00000000	0	0
	CLP(T)	11	00	0011000	00000000	01	000	1	0	00010	00000000	0	0
154	19CH1 (19CH)	11	00	0011001	00000011	01	000	1	0	00010	00000000	0	0
	LMI(T)	11	00	0011001	00000011	01	000	1	0	00010	00000000	0	0
155	19CH1 (19CH)	11	00	0011010	00000000	01	000	1	0	00010	00000000	0	0
	CMQ(T)	11	00	0011010	00000000	01	000	1	0	00010	00000000	0	0
156	1ACH1 (1ACH)	11	00	0011011	11111111	01	000	1	0	00010	00000000	0	0
	ANP(T)	11	00	0011011	11111111	01	000	1	0	00010	00000000	0	0
157	19CH1 (19CH)	11	00	0011100	00000000	01	000	1	0	00010	00000000	0	0
	ILP(T)	11	00	0011100	00000000	01	000	1	0	00010	00000000	0	0
158	1CCH1 (1CCH)	11	00	0011101	00000000	01	000	1	0	00010	00000000	0	0
	CLP(T)	11	00	0011101	00000000	01	000	1	0	00010	00000000	0	0

```

159 10RH: LMI(T) K0001C P44 : /MODE=MODE.013-KFFFFC */
      (010CH) 0011100 11 00 0111000 0001100 01 100 1 0 00010 00000000 0 0 0

160 10RH: CLP(T) :
      (010RH) 1001100 11 00 001110 0000000 01 100 1 0 00010 00000000 0 0 0

161 1FRH: LMI(T) K00006 ROT PAGE1 J2R(FETCH) 3 /MODFI AT=MODE *
      (01FRH) 0011100 11 00 0101111 00000110 00 100 1 0 00010 00000000 0 0 0

162
163
164
165
166 /* ROUTINE WIN WIN WIN WIN WIN WIN WIN WIN WIN WIN WIN WIN */
167
168
169
170 077H: WIN: CLP(T) :
      (0077H) 1001100 11 00 0001001 00000000 01 100 1 0 00010 00000000 0 0 0

```

RECORD NUMBER	CPF	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STR9	LPCNT	EMIT	MULTIPL	SPRF
171	0974H (00974H)	0011100	11	11	0111001	00011000	01	110	1	0	00010	00000000	0
					LMI(T) K0001R FFI RRM :								
172	0994H (00994H)	0001011	11	00	0001010	00000000	01	000	1	0	00010	00000000	0
					ACM(AC) :								
173	0A94H (00A94H)	0100101	11	11	0001011	11111111	01	000	1	0	00010	00000000	0
					SDR(95) FFI : /*95=C(118)=NX*/								
174	0B94H (00B94H)	0011100	11	00	0001100	00000000	01	110	1	0	00010	00000000	0
					LMI(T) RRM :								
175	0C94H (00C94H)	0001011	11	00	0001101	00000000	01	000	1	0	00010	00000000	0
					ACM(AC) :								
176	0D94H (00D94H)	0100110	11	11	0001110	11111111	01	000	1	0	00010	00000000	0
					SDR(96) FFI : /*96=C(119)=NY*/								
177	0E94H (00E94H)	1000111	11	00	0001111	00000000	01	000	1	0	00010	00000000	0
					*LOAD COLUMN LEFT LATCH */								
178	0F94H (00F94H)	0010111	11	00	0010000	00000101	01	000	1	0	00010	00000000	0
					LMI(R7) K00005 :								
179	0F94H (00F94H)	0010111	11	00	0010000	00000101	01	000	1	0	00010	00000000	0


```

180 109H!          CHE(R7) :
(0109H) 1110111 11 00 0010001 00000000 01 000 1 0 00010 00000000 0 0

181 119H!          ILP(R5) :
(0119H) 0000101 11 00 0010010 00000000 01 000 1 0 00010 00000000 0 0

182 129H!          ALP(R7) FFI :
(0129H) 0000111 11 11 0010011 11111111 01 000 1 0 00010 00000000 0 0

183 139H!          CLP(T) :
(0139H) 1001100 11 00 0010100 00000000 01 000 1 0 00010 00000000 0 0

184 149H!          LMI(T) FFI ROT : /*NX-5 TO COL L LATCH */
(0149H) 0011100 11 11 0010101 00000000 01 101 1 0 00010 00000000 0 0

185 /* LOAD COLUMN RIGHT LATCH */

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSEF	STRR	LPCNT	EMIT	MULTIPL	SPRF
184	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(1894)	0010101	11	00	0010110	00000101	01	000	1	0	00010	00000000	0	0
185													
(1894)	0000101	11	00	0010111	00000000	01	000	1	0	00010	00000000	0	0
186													
(1894)	0011100	11	11	0011000	/*NX*5 TO COL R LATCH */	01	101	1	0	00010	00000000	0	0
187													
(1894)	1000111	11	00	0011001	00000000	01	000	1	0	00010	00000000	0	0
188													
(1894)	0010111	11	00	0011010	00000101	01	000	1	0	00010	00000000	0	0
189													
(1894)	1110111	11	00	0011011	00000000	01	000	1	0	00010	00000000	0	0
190													
(1894)	0000110	11	00	0011100	00000000	01	000	1	0	00010	00000000	0	0
191													
(1894)	0000111	11	11	0011101	11111111	01	000	1	0	00010	00000000	0	0
192													
(1894)	0000111	11	11	0011101	11111111	01	000	1	0	00010	00000000	0	0
193													
(1894)	0000110	11	00	0011100	00000000	01	000	1	0	00010	00000000	0	0
194													
(1894)	0000111	11	11	0011101	11111111	01	000	1	0	00010	00000000	0	0

```

195 1094I LMT(T) FFL ROT ; /NY-5 TO ROW UP LATCH */
      (0109H) 0011100 11 11 0011110 0000000 01 101 1 0 00010 0000000 0 0

196 /* LOAD ROW DOWN LATCH */
197 1E9H! LMT(R6) K00005 ;
      (01E9H) 0010110 11 00 0011111 00000101 01 000 1 0 00010 0000000 0 0

198 1F9H! ILP(R6) ;
      (01F9H) 0000110 11 00 0111000 0000000 01 000 1 0 00010 0000000 0 0

199 1FAH! LMT(T) FFL ROT ; /* NY-5 TO ROW DOWN LATCH */
      (01FAH) 0011100 11 11 0001011 0000000 01 101 1 0 00010 0000000 0 0

200 /* READ BLACK/WHITE CONTROL WORD */
201 0A9H! COMMON! CLR(AC) ;
      (00A9H) 1001101 11 00 0001100 0000000 01 000 1 0 00010 0000000 0 0

```

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSEF	STRR	LPCNT	EMIT	MULTIPL	SPRF
202	0C34H (03C34)	001101	11 00	0001101 K00010 R34 :	0001101	01	110	1	0	00010	00000000	0	0
203	0D44H (00D44)	0001011	11 00	0101110 ACM(AC) :	00000000	01	000	1	0	00010	00000000	0	0
204	0E54H (03E54)	0100101	11 11	0001111 SDR(R5) FFI : /* R/W TO R5 */	11111111	01	000	1	0	00010	00000000	0	0
205	/* READ CURRENT MODE LATCH */ 0F64H (03F64)	1001100	11 00	0010000 CLD(T) :	00000000	01	000	1	0	00010	00000000	0	0
207	1074H (11074)	0011100	11 00	0010001 LMT(T) K0001C R34 :	00011100	01	110	1	0	00010	00000000	0	0
208	1184H (01184)	0001011	11 00	0011010 ACM(AC) :	00000000	01	000	1	0	00010	00000000	0	0
209	1A94H (01A94)	0100110	11 11	0010010 SDR(R6) FFI : /* MODE TO R6 */	11111111	01	000	1	0	00010	00000000	0	0

```

210 /* TEST FOR BLACK OR WHITE */
211 12AH 172(R5) FF0 ;
(012AH) 1010101 11 00 0110000 11111111 01 000 1 0 00010 00000000 0 0 0

212 120H CLR(AC) JFL(WHITE,BLACK) ;
(0120H) 1001101 11 00 1000100 00000000 01 000 1 0 00010 00000000 0 0 0

213 142H WHITE: LMT(AC) K0001R ;
(0142H) 0011101 11 00 0110001 00011000 01 000 1 0 00010 00000000 0 0 0

214 141H CLR(AC) ;
(0141H) 1111101 11 00 0010101 00000000 01 000 1 0 00010 00000000 0 0 0

215 151H ANR(R6) ; /* MODE=MODF.AND.KFFFE7 */
(0151H) 1000110 11 00 0010110 11111111 01 000 1 0 00010 00000000 0 0 0

216 141H ILR(R6) ;
(0141H) 0000110 11 00 0010111 00000000 01 000 1 0 00010 00000000 0 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
217	/* WRITE UP-ATED MODE INTO MODE CELL */												
218	171H	LMT(T) RWM :											
(0171H)	0011100	11	00	0011000	00000000	01	111	1	0	00010	00000000	0	0
219	141H	CLP(T) :											
(0141H)	1001100	11	00	0110010	00000000	01	000	1	0	00010	00000000	0	0
220	/* UPDATE MODE LATCH FOR WHITE BOX */												
221	1A2H	LMT(T) K00006 ROT PAGEF1 JZR(FETCH) :											
(01A2H)	0011100	11	00	0101111	00000110	00	101	1	0	00010	00000000	0	0
222	143H	BLACK! LMT(AC) K00010 :											
(0143H)	0011101	11	00	0110100	00010000	01	000	1	0	00010	00000000	0	0
223	144H	OPR(R6) :											
(0144H)	1100110	11	00	0110111	11111111	01	000	1	0	00010	00000000	0	0
224	147H	CLP(AC) :											
(0147H)	1001101	11	00	0010010	00000000	01	000	1	0	00010	00000000	0	0
225	127H	LMT(AC) K00008 :											
(0127H)	0011101	11	00	0110100	00001000	01	000	1	0	00010	00000000	0	0

```

226 124H: 1111101 11 00 0010001 00000000 01 000 1 0 00010 00000000 0 0
    (0124H)

227 114H: 1000110 11 00 0010101 11111111 01 000 1 0 00010 00000000 0 0
    (0114H)  /* MODE=MODF.AND.KFFFF7 */

228 154H: 0000110 11 00 0010110 00000000 01 000 1 0 00010 00000000 0 0
    (0154H)  ILR(06)

229 /* UPDATE MODE FOR PLACK BOX */
230 164H: LMI(T) RWM ;
    (0164H) 0011100 11 00 0010111 00000000 01 111 1 0 00010 00000000 0 0

231 /* UPDATE MODE LATCH FOR PLACK BOX */
232 174H: CLR(T) ;
    (0174H) 1001100 11 00 0011000 00000000 01 000 1 0 00010 00000000 0 0

233 184H: LMI(T) K00006 ROT PAGE1 JZR(FETCH) ;
    (0184H) 0011100 11 00 0101111 00000110 00 101 1 0 00010 00000000 0 0

```

ERRORS= 0 PAGE 13

KMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSEF	STRR	LPCNT	EMIT	MULTIPL	SPRF
234	6543210	10	10	6543210	74543210	10	210	0	0	43210	76543210	0	0
235													
236													
237													
238													
239													
240													
241													
242													
	/* ROUTINE FRD FRM FRM FRM FRD FRM FRM FRM FRM FRM FRM */												
	07AH			FRM: CLR(AC) :				1	0	00010	00000000	0	0
	(007AH)	1001101	11	00	0000001	01	000						
243													
	01AH			LMT(AC) K0001C RRM :				1	0	00010	00000000	0	0
	(001AH)	0011101	11	00	0000010	01	110						
244													
	02AH			ACM(AC) :				1	0	00010	00000000	0	0
	(002AH)	0001011	11	00	0000011	01	000						
245													
	03AH			SNR(R6) FF1 : /*MODE TO R6 */				1	0	00010	00000000	0	0
	(003AH)	0100110	11	11	0000100	01	000						
246													
				/* FORCE REQ BIT HIGH */									
	04AH			CLR(AC) :				1	0	00010	00000000	0	0
	(004AH)	1001101	11	00	0000101	01	000						
249													
	05AH			LMT(AC) K00002 :				1	0	00010	00000000	0	0
	(005AH)	0011101	11	00	0000110	01	000						


```

249 05AH1 1100110 000(R6) ; /* MODE=MODE.OR.00000 ? */
    (005AH) 11 00 0001000 1111111 01 000 1 0 00010 00000000 0 0 0
/* FORCE R/W HIT LOW */
250 05AH1 CLR(AC) ;
    (005AH) 1001101 11 00 0001001 00000000 01 000 1 0 00010 00000000 0 0 0
251 05AH1 LMI(AC) K00001 ;
    (005AH) 0011101 11 00 0001010 00000001 01 000 1 0 00010 00000000 0 0 0
252 05AH1 CLR(AC) ;
    (005AH) 1111101 11 00 0110000 00000000 01 000 1 0 00010 00000000 0 0 0
253 05AH1 ANR(R6) ; /* MODE=MODE.AND.FFFF ? */
    (005AH) 1000110 11 00 0001011 1111111 01 000 1 0 00010 00000000 0 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

REC030 NUMREP	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTPL	SPRF
255	0000110 (0000H)	10	10	6543210 ILR(R6) :	76543210 00000000	10	000	0	0	43210	76543210	0	0
256	0001100 (0000H)	11	00	0001101 CLP(T) :	00000000	01	000	1	0	00010	00000000	0	0
257	0001100 (0000H)	11	00	0001110 LMI(T) K00006 ROT :	00000110	01	101	1	0	00010	00000000	0	0
258	0001100 (0000H)	11	00	0001111 CLP(T) :	00000000	01	000	1	0	00010	00000000	0	0
259	0001100 (0000H)	11	00	0101111 LMI(T) K0001C PAGE1 RWM STORE JZR(FETCH) :	00011100	00	111	1	1	00010	00000000	0	0

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSEFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
281	0011001 (00C5H)	11	00	0001101 K0001F :	00011111 01	01	000	1	0	00010	00000000	0	0
282	0011001 (00D5H)	11	00	0110111 LMI(R9) K0001F : /*R9=SA=IP+62 */	00011111 01	01	000	1	0	00010	00000000	0	0
283	0011001 (00D7H)	11	00	0001100 LMI(R9) K0000A: /* R9=SA=IP+72 */	0001010 01	01	000	1	0	00010	00000000	0	0
284	1001000 (00C7H)	11	00	0110110 CLD(RR):	00000000 01	01	000	1	0	00010	00000000	0	0
285	0011000 (00C6H)	11	11	0001010 LMI(R9) K0001A FFI R0M:	00011010 01	01	110	1	0	00010	00000000	0	0
286	0001011 (00A6H)	11	00	0001001 ACM(AC): /* AC=DX */	00000000 01	01	000	1	0	00010	00000000	0	0
287	0001001 (0095H)	11	00	0001000 ALR(R9): /*R9=IP+72+0X */	11111111 01	01	000	1	0	00010	00000000	0	0
288	0011000 (00B6H)	11	00	0001111 LMI(R9) R0M:	00000000 01	01	110	1	0	00010	00000000	0	0

```

289 0F6H!          ACM(AC): /* AC=DY */
    (09F6H) 0001011 11 00 0110111 00000000 01 000 1 0 00010 00000000 0 0 0

290 0F7H!          AND(R9):
    (09F7H) 0111001 11 00 0010000 11111111 01 000 1 0 00010 00000000 0 0 0

291 107H!          ALR(AC): /* AC=DY*2 */
    (0107H) 0001101 11 00 0010001 11111111 01 000 1 0 00010 00000000 0 0 0

292 117H!          ALR(AC):
    (0117H) 0001101 11 00 0110110 11111111 01 000 1 0 00010 00000000 0 0 0

293 116H!          ALR(AC):
    (0116H) 0001101 11 00 0010000 11111111 01 000 1 0 00010 00000000 0 0 0

294 116H!          ALP(AC): /* AC=DY*16 */
    (0106H) 0001101 11 00 0001110 11111111 01 000 1 0 00010 00000000 0 0 0

```

XMAS VOPS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CAUS	PAUSFF	STPR	LPCNT	EMIT	MULTIPL	SPRF
295	0564H (00F6H)	10	10	6543210	74543210	10	010	0	0	43210	74543210	0	0
				AND(R9): /* R9=IP+72+DX+17*0; */									
296	0554H (00F5H)	11	00	0110101	11111111	01	000	1	0	00010	00000000	0	0
				/* FETCH MEAN VALUE */									
297	0554H (00F5H)	10	00	0010000	00000000	01	000	1	0	01000	11110110	0	0
				CLP(AC) LCNT2 M10;									
298	1054H (0105H)	11	00	0010001	00010100	01	010	1	0	00010	00000000	0	0
				LMT(AC) K00014 RRM;									
299	1154H (0115H)	11	00	0010010	00000000	01	000	1	0	00010	00000000	0	0
				ACM(AC);									
300	1254H (0125H)	11	00	0110110	00000000	01	000	1	0	00010	00000000	0	0
				CMP(AC);									
301	1264H (0126H)	11	11	0010011	11111111	01	000	1	0	00010	00000000	0	0
				SOR(R8) FFI; /* RR=COMP OF MEANS */									
302	1364H (0136H)	11	00	0110111	00000000	01	000	1	0	00010	00000000	0	0
				/* REGTN LOOPS */									
303	1364H (0136H)	11	00	0110111	00000000	01	000	1	0	00010	00000000	0	0
				ILP(R8);									

304	137H:	(0137H)	0100111	SOR(R7) FFI : /*MEAN TO R7 */	11 11 0110010 1111111	01	000	1	0	00010 00000000	0	0
305	132H:	(0132H)	LXPI LMI(R9) FFI PRM :	0011001 11 11 0010101 00000000	01	110	1	0	00010 00000000	0	0	0
306	152H:	(0152H)	LXPP2: ACM(AC) :	0001011 11 00 0110000 00000000	01	000	1	0	00010 00000000	0	0	0
307	150H:	(0150H)	ALR(AC) :	0001101 11 00 0010100 11111111	01	000	1	0	00010 00000000	0	0	0
308	140H:	(0140H)	ALR(AC) :	0001101 11 00 0010111 11111111	01	000	1	0	00010 00000000	0	0	0
309	170H:	(0170H)	ALP(AC) :	0001101 11 00 0011000 11111111	01	000	1	0	00010 00000000	0	0	0

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPF	FI	FO	JUMP	KBUSS	PAGEF	CBUS	PAUSEFF	STOR	LPCNT	EMIT	MULTIPL	SPQF
310	1404H (01A0H)	0001101	11 00	0011001	11111111	01	000	1	0	00010	00000000	0	0
				ALP(AC) :									
311	1304H (0100H)	0001101	11 00	0110010	11111111	01	000	1	0	00010	00000000	0	0
				ALP(AC) :									
312	1324H (0102H)	0000111	11 11	0110110	11111111	01	000	1	0	00010	00000000	0	0
				ALR(R7) FF1 :	/* (2**5*(SA)-MEANS) TO AC */								
				11 11	0110110	11111111	01	000	1	0	00010	00000000	0
313	1344H (0106H)	1101101	11 00	0010101	00000000	01	000	1	0	00010	00000000	0	0
				NOP(AC) :									
314	1544H (0156H)	1011101	11 00	0010110	10000000	01	001	1	0	00010	00000000	0	0
				/* TAKE ABSOLUTE VALUE */									
				I7P(AC) KR0000 IVM :									
315	1464H (0166H)	1101101	11 00	1000110	00000000	01	000	1	0	00010	00000000	0	0
				NOP(AC) JFL(APX*ANX) :									
317	1434H (0163H)	0011111	11 11	0110010	00000000	01	000	1	0	00010	00000000	0	0
				ANX1 C1A(AC) FF1 :									
318	1424H (0162H)	0001100	11 00	0111000	11111111	01	000	1	0	00010	00000000	0	0
				APX1 ALR(T) FF0 :	/* SUM=SUM*ARS(C1A(I))-MEAN) */								
				11 00	0111000	11111111	01	000	1	0	00010	00000000	0


```

319 1484I      ILR(R8) ITCNT1:      0001000 11 00 0110000 00000000 01 000 1 0 00111 00000000 0 0
    (01484H)

320 1494I      SDR(R7) FF1 JFL(LXXP-CONTX1): 0100111 11 11 100011 1111111 01 000 1 0 00010 00000000 0 0
    (01494H)

321 1334I      CONTX1: LMI(R9) K0000R ITCNT2: 0011001 11 00 0110100 00001000 01 000 1 0 01011 00000000 0 0
    (01334H)

322 1344I      LMI(Q9) FF1 RRM LCNT1 M10 JFL(LXXP2-CONTX2): 0011001 11 11 1000101 00000000 01 110 1 0 00100 11110110 0 0
    (01344H)

323 1534I      CONTX2:      ILR(T) :      0001100 11 00 0110111 00000000 01 000 1 0 00010 00000000 0 0
    (01534H)

324 1574I      ALR(AC) :

```

XMAS VERS 2.0 CORRELATION MICROCODE ERRORS= 0 PAGE 1A

RECORD NUMBER	PE	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
(0157H)	0001101	11	00	0010110	11111111	01	000	1	0	00010	00000000	0	0
325	167H	0001101	11	00	0010111	01	000	1	0	00010	00000000	0	0
(0167H)	ALR(AC) ; /*SUM=(SUM) */												
326	/*	ST.PF	STD	IN	MEMORY	*/							
327	177H	1001001	11	00	0011000	01	000	1	0	00010	00000000	0	0
(0177H)	CLR(R9) ;												
328	187H	0011001	11	00	0000115	00	000	1	0	00010	00000000	0	0
(0187H)	LMI(R9) K00015 RVM PAGE1 J72(FETC-) ;												
329	/*	ROUTINE	OFF	REF	REF	REF	REF	REF	REF	REF	REF	REF	REF
330	/*	RR=SA	*/										
331	/*	R9=SA	*/										
332	/*	F0PM	STARTING	ADDRESS	SA	*/							
333	070H	0011001	11	00	0010011	00	000	1	0	00010	00000000	0	0
(0070H)	OFF! CLR(AC) ;												

341	13DH (013DH)	0011101	LMT(AC) K00010 RRM :	11 00 0110101 00010000	01	110	1	0	00010	00000000	0	0
342	135H (0135H)	0001011	ACM(AC) : /*IP TO T */	11 00 0010100 00000000	01	000	1	0	00010	00000000	0	0
343	145H (0145H)	0011101	LMT(AC) K0001F :	11 00 0010101 00011111	01	000	1	0	00010	00000000	0	0
344	155H (0155H)	0011101	LMT(AC) K0001F :	11 00 0010110 00011111	01	000	1	0	00010	00000000	0	0
345	165H (0165H)	0011101	LMT(AC) K0000A :	11 00 0010111 00001010	01	000	1	0	00010	00000000	0	0
346	175H (0175H)	0101001	SDR(R9) FF1 : /*R9=IP.72=5A */	11 11 0110110 11111111	01	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUISS	PAGEF	CBUS	PAUSEF	STARR	LPCNT	EMIT	MULTIPL	SPRF
347	176H (0176H)	1001000	11 00	0011011	00000000	01	000	1	0	00010	00000000	0	0
				CLP(RR):									
348	186H (0186H)	0011000	11 11	0011100	00011010	01	110	1	0	00010	00000000	0	0
				LMT(RR) RRM FFI <0001A:									
349	1C6H (01C6H)	0001011	11 00	0011101	00000000	01	000	1	0	00010	00000000	0	0
				ACM(AC): /*AC=DX*/									
350	1D6H (01D6H)	0111001	11 00	0011110	11111111	01	000	1	0	00010	00000000	0	0
				ADR(R9) : /*R9=IP+SA+DX*/									
351	1E6H (01E6H)	0011000	11 00	0110111	00000000	01	110	1	0	00010	00000000	0	0
				LMT(RR) RRM:									
352	1F7H (01F7H)	0001011	11 00	0011101	00000000	01	000	1	0	00010	00000000	0	0
				ACM(AC): /*AC=DY*/									
353	1D7H (01D7H)	0111001	11 00	0011100	11111111	01	000	1	0	00010	00000000	0	0
				ADR(R9): /*R9=IP+DX+72+DY*/									
354	1C7H (01C7H)	0001101	11 00	0011011	11111111	01	000	1	0	00010	00000000	0	0
				ALP(AC): /*AC=DY*2*/									

```

355 1A7H! ALR(AC): /*AC=DY*4 */
    (01A7H) 0001101 11 00 0011010 11111111 01 000 1 0 00010 00000000 0 0

356 1A7H! ALR(AC): /*AC=DY*8 */
    (01A7H) 0001101 11 00 0110110 11111111 01 000 1 0 00010 00000000 0 0

357 1A6H! ALR(AC): /*AC=DY*16 */
    (01A6H) 0001101 11 00 0011000 11111111 01 000 1 0 00010 00000000 0 0

358 1A6H! ANP(R9): /*R9=IP+72+DX+DY*17 */
    (01A6H) 0111001 11 00 0110101 11111111 01 000 1 0 00010 00000000 0 0

359 /* FORM STORAGE ADDRESS STA
360 CLR(AC) LCNT1 M10;
    (01A5H) 1001101 11 00 0011001 00000000 01 000 1 0 00100 11110110 0 0

361 1A5H! LMT(AC) K00011 R2M ;

```

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	0	0	43210	76543210	0	0
(01954)	001101	11	00	0011010	00010001	01	1	0	00010	00000000	0	0
362	1A54I			ACM(AC) LCNT2 M10:								
(01A54)	0001011	11	00	0011011	00000000	01	1	0	01000	11110110	0	0
363	1A54I			SDR(R8) FFI :	/*RR=RR=STA #/							
(01A54)	0101000	11	11	0110100	11111111	01	1	0	00010	00000000	0	0
364	1A44I			CLP(T):								
(01A44)	1001100	11	00	0110010	00000000	01	1	0	00010	00000000	0	0
365	1A24I			/* STADT LOOPS #/								
(01A24)	0011001	11	11	0011100	00000000	01	1	0	00010	00000000	0	0
367	1C24I			LX2I ACM(AC):								
(01C24)	0001011	11	00	0110100	00000000	01	1	0	00010	00000000	0	0
368	1C44I			LMI(R8) FFI RWM JFL(LX1+CX1) :								
(01C44)	0011000	11	11	1001011	00000000	01	1	0	00010	00000000	0	0
369	1A34I			CX1I LMI(P9) K00008 ITCNT2 :								
(01A34)	0011001	11	00	0110001	00001000	01	1	0	01011	00000000	0	0

```

370 1914I LMI(R9) FF1 RRM LCNT1 M10 JFL(LX2,CX2):
    (0191H) 0011001 11 11 1001100 00000000 01 110 1 0 00100 11110110 0 0
371 1C34I CX2: NOP(AC) PAGE1 JZ3(FFTCB):
    (01C3H) 1101101 11 00 0101111 00000000 00 000 1 0 00010 00000000 0 0
372
373
374
375
376
377 /* ROUTINE WAT WAT WAT WAT WAT WAT WAT WAT WAT WAT */
378
379
380 0794I WAT: NOP(AC) RIN:
    (0079H) 1101101 11 00 0000001 00000000 01 100 1 0 00010 00000000 0 0
381 0194I LDI(AC) FF1:
    (0019H) 0101111 11 11 0000010 11111111 01 000 1 0 00010 00000000 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	OPER	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSEFF	STPR	LPCNT	FMIT	MULTIPL	SPRF
382	0294I (00204)	1111101	11 00	0000011	10000000	01	000	1	0	00010	00000000	0	0
				I70(AC) K40000 :									
383	0304I (00304)	1111101	11 00	1000011	00000000	01	000	1	0	00010	00000000	0	0
				N0P(AC) JEL(NRDY+00Y) :									
384	0344I (00354)	1111101	11 00	0111111	00000000	00	000	1	0	00010	00000000	0	0
				N0P(AC) J70(FETC+) PAGEF1 :									
385	0344I (00364)	1111101	11 00	0101001	00000000	01	000	1	0	00010	00000000	0	0
				N0P(AC) JZ0(XX2) :									
385	0094I (00094)	1111101	11 00	0000111	00000000	01	000	1	0	00010	00000000	0	0
				N0P(AC) JCC(WAT) :									


```

387
388
389
390
391
392
393
394
395
396
397
398
399
400
/* ROUTINE MNA MNV MNV MNA MNV MNV MNV MNV MNV MNV */
/* R9=SA */
/* T=SUM */
/* AC=WORKING */
/* ROUTINE TO CALCULATE MEAN OF CENTER 9X9 IMG */
07CH: MNA CLR(T) LCNT1 M10:
(007CH) 1001100 11 00 0000001 00000000 01 000 1 0 00100 11110110 0 0

/* P8=WORKING */
/* CALCULATE STARTING ADDRESS */
01CH: CLR(R9) LCNT2 M10:
(001CH) 1001001 11 00 0110100 00000000 01 000 1 0 01000 11110110 0 0

014H: LMI(P9) K00010 R24:
(0014H) 0011001 11 00 000010 00010000 01 010 1 0 00010 00000000 0 0

024H: ACY(AC):
(0024H) 0001011 11 00 0000000 00000000 01 000 1 0 00010 00000000 0 0

004H: SDD(R9) FF1: /* DD=IP */
(0004H) 0101001 11 11 0000011 11111111 01 000 1 0 00010 00000000 0 0

```

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
406 (0244)	6543210	10	10	6543210	76543210	10	110	0	0	43210	76543210	0	0
	1001000	11	00	0110110	00000000	01	000	1	0	00010	00000000	0	0
				CLP(R8):									
407 (0244)				LMT(R8) K0001A FFI RRM:									
	0011000	11	11	0000010	00011010	01	110	1	0	00010	00000000	0	0
408 (0244)				ACM(AC): /* AC=DX */									
	0001011	11	00	0110101	00000000	01	000	1	0	00010	00000000	0	0
409 (0254)				LMT(AC) K0001F:									
	0011101	11	00	0000011	00011111	01	000	1	0	00010	00000000	0	0
410 (0254)				LMT(AC) K0001F: /* AC=DX+62 */									
	0011101	11	00	0110111	00011111	01	000	1	0	00010	00000000	0	0
411 (0274)				LMT(AC) K0000A: /* AC=DX+72 */									
	0011101	11	00	0000001	00001010	01	000	1	0	00010	00000000	0	0
412 (0174)				ALP(R9): /*99.AC=TP+DX+72 */									
	0001001	11	00	0000010	11111111	01	000	1	0	00010	00000000	0	0
413 (0274)				LMT(R8) RRM:									
	0011000	11	00	0000100	00000000	01	110	1	0	00010	00000000	0	0

414	057H (0047H)	0001011	ACM(AC): /* AC=0Y */	01	000	1	0	00010	00000000	0	0
415	067H (0057H)	0111001	ADP(R9): /*R9=IP+0X+72+0Y */	01	000	1	0	00010	00000000	0	0
416	087H (0087H)	0001101	ALP(AC): /* AC=0X*2 */	01	000	1	0	00010	00000000	0	0
417	0A7H (00A7H)	0001101	ALP(AC): /* AC=0X*4 */	01	000	1	0	00010	00000000	0	0
418	0B7H (00B7H)	0001101	ALP(AC): /* AC=0X*8 */	01	000	1	0	00010	00000000	0	0
419	0D7H (00D7H)	0001101	ALP(AC): /* AC=0X*16 */	01	000	1	0	00010	00000000	0	0

RECORD NUMBER	CPF	FT	FO	JIMP	KRUSS	PAGEF	CRUS	PAUSFF	STPR	LPCNT	FMIT	MULTIPL	SPRF
420	054H (0054H)	0111001	11 00	0110001	11111111	01	000	1	0	00010	00000000	0	0
				ADD(R9): /* R9=SA=IP+72+DX+DY*17 */									
421	051H (0051H)	1001111	11 00	0110010	00000000	01	000	1	0	00010	00000000	0	0
				CLP(AC):									
422	052H (0052H)	1001101	11 11	0000100	00000000	01	110	1	0	00111	00000000	0	0
				L00P1: LMT(R9) FF1 RRM ITCNT1: /*SA=SA+1 */									
423	042H (0042H)	1001101	11 00	1000101	11111111	01	000	1	0	00010	00000000	0	0
				LXXX: AMA(AC) JFL(L00P1,CONT1):									
424	053H (0053H)	0011001	11 00	0110101	00001000	01	000	1	0	01011	00000000	0	0
				CONT1: LMT(R9) K00009 ITCNT2:									
425	055H (0055H)	0011001	11 11	1000110	00000000	01	000	1	0	00100	11110110	0	0
				LMT(R9) FF1 LCNT1 M10 JFL(L00P2,CONT2):									
426	042H (0062H)	1101101	11 00	0000100	00000000	01	110	1	0	00111	00000000	0	0
				L00P2: NOP(AC) ITCNT1 RRM JCC(LXXX):									
427	043H (0063H)	1001101	11 00	0110100	00000000	01	000	1	0	00010	00000000	0	0
				CONT2: CLP(R9):									

```

428 054H1 LMI(R9) K00014 ; /* ADR OF MEANS */
(0064H) 0011001 11 00 0110101 00010100 01 000 1 0 00010 00000000 0 0 0

429 055H1 NOP(AC) R44 PAGE1 J7R(FETCH);
(0055H) 1101101 11 00 0101111 00000000 00 111 1 0 00010 00000000 0 0 0

430
431
432
433
434
435 /*ROUTINE CROSS CROSS CROSS CROSS CROSS */
436
437
438 CROSS1 CLP(AC) ;
(0074H) 1001101 11 00 0001000 00000000 01 000 1 0 00010 00000000 0 0 0

439 084H1 LMI(AC) K00018 R24 ;
(0084H) 0011101 11 00 0001001 00011000 01 110 1 0 00010 00000000 0 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPF	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STPR	LPCNT	EMIT	MULTIPL	SPRF
440	034H (0304H)	11	00	0001010	00000000	01	000	1	0	00010	00000000	0	0
		ACM(AC) :											
441	0A4H (0A04H)	11	00	0001011	00000000	01	000	1	0	00010	00000000	0	0
		CLR(T) :											
442	034H (0304H)	11	11	0001100	/* 00=NX COL LL */	01	101	1	0	00010	00000000	0	0
		LMI(T) FFI ROT :											
443	0C4H (0C04H)	11	11	0001101	/* 01=NX COL PL */	01	101	1	0	00010	00000000	0	0
		LMI(T) FFI ROT :											
444	0D4H (0D04H)	11	00	0001110	00000000	01	000	1	0	00010	00000000	0	0
		CLR(AC) :											
445	0E4H (0E04H)	11	00	0001111	00011001	01	110	1	0	00010	00000000	0	0
		LMI(AC) K00019 R0W :											
446	0F4H (0F04H)	11	00	0110101	00000000	01	000	1	0	00010	00000000	0	0
		ACM(AC) :											
447	0F5H (0F05H)	11	11	0001011	/* D2=NY ROW UL */	01	101	1	0	00010	00000000	0	0
		LMI(T) FFI ROT :											

```

448 0R5H! 0011100 11 11 0011111 00000000 01 101 1 0 00010 00000000 0 0 0
(0R5H)

449 /* PFAD BLACK/WHITE CONTROL WORD */
450 1F5H! 1001101 11 00 0111100 00000000 01 000 1 0 00010 00000000 0 0 0
(01F5H)

451 1FCH! 0011101 11 00 0011110 00011101 01 110 1 0 00010 00000000 0 0 0
(01FCH)

452 1ECH! 0001011 11 00 0111101 00000000 01 000 1 0 00010 00000000 0 0 0
(01ECH)

453 1EDH! 0100101 11 11 0010111 11111111 01 000 1 0 00010 00000000 0 0 0
(01EDH)

```

```

454 /* READ CURRENT MODE */

```

XMAS VERS 2.0 CORRELATION MICROCODEF

RECORD NUMBER	CPF	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STR8	LPCNT	EMIT	MULTIPL	SPRF
455	17041 (01704)	10	10	0011000	00000000	01	000	1	0	00010	00000000	0	0
				CLR(T) :									
456	18041 (01804)	11	00	0011001	00011100	01	110	1	0	00010	00000000	0	0
				LMI(T) K0001C RRM :									
457	19041 (01904)	11	00	0011010	00000000	01	000	1	0	00010	00000000	0	0
				ACM(AC) :									
458	1A041 (01A04)	11	11	0011011	11111111	01	000	1	0	00010	00000000	0	0
				SOR(86) FFL : /* MODE TO R6 */									
459	1B041 (01B04)	11	00	0011100	11111111	01	000	1	0	00010	00000000	0	0
				/* TEST FOR BLACK OR WHITE */									
460	1C041 (01C04)	11	00	1001101	00000000	01	000	1	0	00010	00000000	0	0
				T7R(85) FFO :									
461	1D041 (01D04)	11	00	0111110	00010000	01	000	1	0	00010	00000000	0	0
				CLR(AC) JFL(WHITE2+BLACK2) :									
462	1E041 (01E04)	11	00	0111110	00010000	01	000	1	0	00010	00000000	0	0
				LMI(AC) K0001C :									
463	1F041 (01F04)	11	00	0010101	00000000	01	000	1	0	00010	00000000	0	0
				CMP(AC) :									

464	15EH (015EH)	1000110	ANP(R6) :	11 00 0010110	11111111 01	000	1	0	00010	00000000	0	0
465	16EH (016EH)	1001101	CLR(AC) :	11 00 0010111	00000000 01	000	1	0	00010	00000000	0	0
466	17EH (017EH)	0011101	LMI(AC) K000008 :	11 00 0010100	00001000 01	000	1	0	00010	00000000	0	0
467	14EH (014FH)	1100110	ORP(R6) :	11 00 0011000	11111111 01	000	1	0	00010	00000000	0	0
468	18EH (018EH)	0000110	ILR(R6) :	11 00 0011001	00000000 01	000	1	0	00010	00000000	0	0
469	19EH		LMI(T) RWM :									

RECORD NUMBER	OPF	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
	(119F4)	0011100	11	00	0011010	00000000	01	111	1	0	00010	00000000	0
470	1AFH1 (11AFH)	1001100	11	00	0011011	00000000	01	000	1	0	00010	00000000	0
471	1BFH1 (11BFH)	0011100	11	00	0101111	00000110	00	101	1	0	00010	00000000	0
472	1DFH1 (11DFH)	0011101	11	00	0111101	00011000	01	000	1	0	00010	00000000	0
473	1DFH1 (11DFH)	1100110	11	00	0111111	11111111	01	000	1	0	00010	00000000	0
474	1DFH1 (11DFH)	0000110	11	00	0011011	00000000	01	000	1	0	00010	00000000	0
475	1BFH1 (11BFH)	0011100	11	00	0011100	00000000	01	111	1	0	00010	00000000	0

```

474 ICFH! CLR(T) :
(01CFH) 1001100 11 00 0111110 00000000 01 000 1 0 00010 00000000 0 0 0

477 ICEH! LMI(T) K00004 ROT PAGE1 JZR(FETCH) :
(01CEH) 0011100 11 00 0101111 00000110 00 101 1 0 00010 00000000 0 0 0

478 00FH! FFTCH! NOP(AC) JZR(FETCH) :
(000FH) 1101101 11 00 0101111 00000000 01 000 1 0 00010 00000000 0 0 0

```

479 EOF

NO PROGRAM ERRORS
END OF PROGRAM

XMAS VERS 2.0 CORRELATION MICROCDEF

MICROPROGRAM MEMORY IMAGE

	04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
0004	JCC 00764	JCC 00794	J7R 000FM
	405	396	478 17
0014	JCC 00244	.	JCC 00274	JCC 00284	JCC 00294	JCC 00264	.	JCC 00164	JCC 00204	JCC 00254	JCC 00274	JCC 002FM
	403	.	412	243	244	391	.	402	128	123	115	
0024	JCC 00964	JCC 00754	JCC 00254	JCC 00474	JCC 00364	JCC 00304	.	.	JCC 00404	JCC 003FM	JCC 003FM	JCC 003FM
	404	409	408	413	246	392	.	.	131	125	114	
0034	JCC 00364	JCC 00734	JCC 00264	JCC 00174	JCC 00484	JCC 00344	J7R 00094	J7R 000FH	JCC 00104	JCC 00304	JCC 006FM	JCC 006FM
	405	410	407	411	245	393	395	394	127	126	117	
0044	.	JFL	.	.	JCC 00474	JCC 00504	.	JCC 00474	JCC 00504	JCC 00504	.	.	JCC 00504	JCC 005FM	JCC 005FM	
	.	423	.	.	414	267	.	414	267	.	.	.	132	119	114	
0054	JCC 00424	JCC 00454	JCC 00514	JCC 00424	JCC 00514	JCC 00424	JCC 00514	JCC 00424	JCC 00514	JCC 00424	JCC 00514	JCC 00424	JCC 00514	JCC 00424	JCC 00514	JCC 00424
	421	422	424	423	423	425	419	248	248	394	.	.	133	56	120	
0064	JCC 00424	JCC 00464	JCC 00454	JCC 00464	JCC 00454	JCC 00464	JCC 00454	JCC 00464	JCC 00454	JCC 00464	JCC 00454	JCC 00464	JCC 00454	JCC 00464	JCC 00454	JCC 00464
	426	427	423	429	429	415	249	415	249	.	.	.	134	122	121	
0074	J7R 000FM	JCC 00934	JCC 00884	JCC 00934	JCC 00884	JCC 00934	JCC 00884	JCC 00934	JCC 00884	JCC 00934	JCC 00884	JCC 00934	JCC 00884	JCC 00934	JCC 00884	JCC 00934
	59	60	275	434	434	431	170	242	242	390	94	71	400	40	41	

XMAS VERS 2.0 CORRELATION MICROCODE

MICROPROGRAM MEMORY IMAGE

	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	CH	DM	EM	FM
008H	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	0085H	0096H	0095H	00F6H	00A7H	009AH									
	277	439	278	28A	416	2E1					95		136		9A
	1	1	1	1	1	1					1		1		1
009H	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00A6H	00A5H	00A5H	00A6H	0099H	00A8H	00A9H								
	440	279	287	287	171	252	172						137		97
	1	1	1	1	1	1	1						1		1
00AH	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00B0H	00B4H	00C5H	0096H	00A7H	00A8H	00A9H								
	254	441	290	286	417	253	173								9A
	1	1	1	1	1	1	1								1
00BH	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00C0H	00C4H	01F5H												
	255	442	44A		41A	261	174								99
	1	1	1	1	1	1	1								1
00CH	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00D0H	00D4H	00F5H	00A6H	00C6H	00D4H	00D9H								
	256	443	281	285	284	2A2	175								100
	1	1	1	1	1	1	1								1
00EH	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00E0H	00E4H	00F7H												
	257	444	282		283	2A3	176								101
	1	1	1	1	1	1	1								1
00FH	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00F0H	00F4H	01A5H	00E5H											
	259	445	297	295		2A4	178								102
	1	1	1	1	1	1	1								1
00GH	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	00G0H	00G4H	0095H	00F7H	0107H	010AH	0109H								
	259	445	299	289	290	2A5	179								103
	1	1	1	1	1	1	1								1

MEMORY TRACE

	0-1	14	24	34	44	54	64	74	84	94	A4	R4	C4	D4	E4	F4
0104	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01154	00E64	01174	01194	01174	01194	01174	01194	01174	01194	01174	01194	01174	01194	01174	01194
	298	294	291	287	291	287	291	287	291	287	291	287	291	287	291	287
0114	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01564	01254	01164	01404	01164	01404	01164	01404	01164	01404	01164	01404	01164	01404	01164	01404
	227	209	202	208	202	208	202	208	202	208	202	208	202	208	202	208
0124	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01144	01264	01364	01264	01364	01264	01364	01264	01364	01264	01364	01264	01364	01264	01364	01264
	225	300	301	301	295	271	163	163	163	163	163	163	163	163	163	163
0134	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01524	01364	01374	01374	01374	01374	01374	01374	01374	01374	01374	01374	01374	01374	01374	01374
	305	321	303	303	304	304	143	144	144	143	144	144	143	144	144	143
0144	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01514	01414	01464	01474	01474	01474	01474	01474	01474	01474	01474	01474	01474	01474	01474	01474
	214	213	222	223	224	224	184	184	184	184	184	184	184	184	184	184
0154	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01614	01504	01574	01644	01654	01674	01664	01674	01664	01674	01664	01674	01664	01674	01664	01674
	215	306	323	229	344	315	324	324	186	148	150	151	151	151	151	151
0164	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01714	01684	01624	01744	01754	01624	01624	01774	01664	01794	01794	01794	01794	01794	01794	01794
	214	319	317	230	345	316	325	319	187	187	152	152	152	152	152	152
0174	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01814	01814	01844	01844	01844	01844	01844	01844	01844	01844	01844	01844	01844	01844	01844	01844
	219	218	232	232	346	347	327	327	189	189	153	153	153	153	153	153

XMAS VERS 2.0 CORRELATION MICROCODE

MICROPROGRAM MEMORY IMAGE

	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DM	EM	FM
0190H	JCC	JCR	JZR	JZR	JZ3	JCC	JCR	JZR	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	0190H	0182H	000FH	000F4	0105H	0185H	000FH	000FH	0199H	0199H	0199H	0199H	0199H	0199H	0199H	0199H
0191H	310	219	221	233	360	358	328	328	190	190	190	190	154	456	468	112
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0192H	JCR	JCR	JCR	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	0192H	0196H	0196H	01A5H	0154H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H
0193H	311	312	312	361	361	313	313	313	191	191	191	191	155	457	469	113
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01A0H	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01A0H	01A2H	01A2H	01A5H	01A6H	01A6H	01A6H	01A6H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H	01A9H
01A1H	357	357	356	356	357	357	356	356	192	192	192	192	156	458	470	114
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01A2H	JFL	JCC	JCC	JCR	JCR	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01C2H	01C2H	01C2H	01A2H	01A2H	01A2H	01A2H	01A2H	01C9H	01C9H	01C9H	01C9H	01C9H	01C9H	01C9H	01C9H
01A3H	370	366	369	364	363	348	355	355	193	193	193	193	157	460	471	475
	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01C0H	JCR	JCR	JZR	JFL	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JFL	JZR	JCR
	01C4H	000FH	000FH	01A2H	01D6H	01A7H	01A7H	01A7H	01D9H	01D9H	01D9H	01D9H	01D9H	01D9H	000FH	01CFH
01C1H	367	371	371	368	349	354	354	354	194	194	194	194	158	461	477	476
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01D0H	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01E6H	01E6H	01C7H	01E6H	01E6H	01C7H	01E6H	01E6H	01E9H	01E9H	01E9H	01E9H	01E9H	01E9H	01E9H	01E9H
01D1H	350	353	340	353	350	350	353	353	195	195	195	195	159	473	483	474
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01E0H	JCR	JCC	JZR	JCC	JCR	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01E7H	01D7H	000FH	01F9H	01E7H	01D7H	000FH	01F9H	01F9H	01F9H	01F9H	01F9H	01F9H	01F9H	01F9H	01F9H
01F0H	351	352	341	352	351	351	352	352	197	197	197	197	152	453	453	453
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01F1H	JCR	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC
	01FCH	01FCH	008FH	01FCH	01FCH	01FCH	008FH	01FCH	01FCH	01FCH	01FCH	01FCH	01FCH	01FCH	01FCH	01FCH
01F2H	450	450	450	450	450	450	450	450	198	198	198	198	135	135	135	135
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

XMAS VERS 2.0 CORRELATION MICROCODE

CROSS REFERENCE DIRECTORY

LABEL	REFERENCES
ANX	316.(317)
APX	316.(318)
BLACK	212.(222)
BLACK2	461.(472)
COMMON	(201)
CONT1	423.(424)
CONT2	425.(427)
CONTX2	322.(323)
CONTX	320.(321)
CORR	57.(401),40
CROSS	56.(438)
CTI	145.(146)
CTP	147.(150)
CXI	368.(369)
CX2	370.(371)
FETCH	58,59,60,61,71,81,161,221,233,259,328,371,384,420,471,477,(478),479
FRM	56.(242)
IMG	57.(94)
LOOP1	(422),427
LOOP2	425.(428)
LP1	(143),145
LP1X	(144),148
LP2	147.(148)
LX1	(366),368
LX2	(367),370
LXXP	(305),320
LXXP2	(306),322
LXXX	(423),424
VIN	56,(70),70
WNV	57.(400)
NOT1	56.(58)
NOT1P	57.(71)
NOT16	57.(81)
NOT2	56.(59)
NOT3	56.(60)
NOT6	56.(61)
NRDY	383.(385)
RDY	383.(384)
REF	57.(340)
STQ	56.(274)
WAT	57.(380),386
WHITE	212.(213)
WHITE2	461.(462)
WIN	56.(170)
XX2	385.(386)

(LISTING #3)

\$\$\$	\$\$\$\$	\$F\$	\$\$\$\$	\$:\$\$		
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$
\$	\$	\$	\$	\$	\$	\$

XMAS VERS 2.0

ERRORS= 0 PAGE 1

SLJSTFILF=2
8BITS
SCROSSOFF
STAGE

\$WIDTH=132
\$TITLE=CORRELATION MICROCODE
\$DISPLAY(ALL)

BITS=1 LEFT=1

RIGHT=72

TITLE=CORRELATION MICROCODE
WIDTH=132

.....

IMAGE=1

FJAMS=0

.....

SOURCEFILF=2

LINES=60

POINT=1

MICROMEMORY=512

XMAS VERS 2.0 CORRELATION MICROCODE

ERRORS= 0 PAGE 2

DECODE
VIMREQ

CPE FT FO JUMP KRUSS PAGEF CBUS PAUSFF STPB LPCNT EMIT MULTPL SPDF
6543210 10 10 6543210 76543210 10 10 210 0 43210 76543210 0 0

1 KRUSS

-TFLD LFNIGHT=8

DEFAULT=0

MICROPS(K00000=00H K00001=01H <00002=02H
K00003=03H <00004=04H
K00005=05H <00006=06H
K00007=07H <00008=08H
K00009=09H <0000A=0AH
K0000B=0BH <0000C=0CH
K0000D=0DH K0000E=0EH
K0000F=0FH <00010=10H

```

11          K00011=11H <00012=12H
12          K00013=13H <00014=14H
13          K00015=15H <00016=16H
14          K00017=17H <00018=18H
15          K00019=19H <0001A=1AH
16          K0001A=1AH <0001C=1CH
17          K0001D=1DH <0001F=1FH
18          K0001F=1FH <01FFF=3FH
19          KH0000=80H <7FFF=7FH
20          KFFFFFF=7770 KFFFFFFA=0FAH);
21  KRUSS      KRUS;
22
23  PAGFF      FIELD LENGTH=2          DEFAULT=10R
24            MICRUPS (PAGE1=00R PAGE2=01R
25              PAGE3=10R PAGE4=11R);
26
27  CUUS      FIELD LENGTH=3          DEFAULT=0
28            MICRUPS (NR0=000R  INH=001R  DMW=010R  CMB=011R
29              PTN=100R  RNT=101R  DRN=110R  O.M=111R);
30
31  PAUSEF     FIELD LENGTH=1          DEFAULT=1
32            MICRUPS (PAUSE=0);
33
34  STRR      FIELD LENGTH=1          DEFAULT=0
35            MICRUPS (STR09F=1R);
36  LPCNT     FIELD LENGTH=5          DEFAULT=00010R
37            MICRUPS (CO=00010R LCNT1=00100R LCNT2=01000R
38              LCNT3=01100R LCNT4=10000R ITCNT1=00111R
39              ITCNT2=01011R ITCNT3=01111R ITCNT4=10011R);
40
41  EMIT      FIELD LENGTH=8          DEFAULT=0
42            MICRUPS ( M1=0FFH M2=0FFH M3=0FDH M4=0FCH
43              M5=0FAH M6=0FAH M7=0F9H M8=0F8H M9=0F7H M10=0F6H
44              M11=0F5H M12=0F4H M13=0F3H M14=0F2H M15=0F1H
45              M16=0F0H M17=0FFH M18=0FEH);
46
47  MULTIPLY  FIELD LENGTH=1          DEFAULT=0
48            MICRUPS (MULT=1R);
49
50  SPQF      FIELD LENGTH=1          DEFAULT=0
51            MICRUPS (SPARE=01H);
52
53
54

```


74	0RFH! (00RFH)	0011100	LMI(T) RRM FF1: /* T=T+1 */ 11 11 0001100 00000000	10	110	1	0	00010 00000000	0	0
75	0CFH! (00CFH)	0001011	ACM(AC): 11 00 0001101 00000000	10	000	1	0	00010 00000000	0	0
76	0DEH! (00DEH)	0100101	SOR(RS) FF1: /* RS=RP */ 11 11 0001110 11111111	10	000	1	0	00010 00000000	0	0
77	0FFH! (00FFH)	0011100	LMI(T) FF1 RRM: 11 11 0111100 00000000	10	110	1	0	00010 00000000	0	0
78	0ECH! (00ECH)	0001011	ACM(AC): 11 00 0001111 00000000	10	000	1	0	00010 00000000	0	0
79	0FCH! (00FCH)	0100110	SOR(R6) FF1 /*R6=CP */ 11 11 0001101 11111111	10	000	1	0	00010 00000000	0	0

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	PAUSFF	STPB	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	0	0	43210	76543210	0	0
80	/* INITIAL17 LOOP COUNTERS AND LOAD REGISTERS */											
81	0NCH1	11	00	0010101	00000000	10	000	1	00100	11110110	0	0
	(0NDCH)	0000010	11	00	0010101	00000000	10	000	1	00100	11110110	0
82	15CH1	0100111	11	11	0010110	11111111	10	000	1	01000	11110110	0
	(15SCH)	0100111	11	11	0010110	11111111	10	000	1	01000	11110110	0
83	/* INITIAL17 CONSTANTS */											
84	16CH1	1001100	11	00	0010111	00000000	10	000	1	01100	11110110	0
	(16SCH)	1001100	11	00	0010111	00000000	10	000	1	01100	11110110	0
85	17CH1	0011100	11	11	0011000	00011111	10	000	1	10000	11110110	0
	(17SCH)	0011100	11	11	0011000	00011111	10	000	1	10000	11110110	0
86	18CH1	0011100	11	11	0011001	00000000	10	110	1	00010	00000000	0
	(18SCH)	0011100	11	11	0011001	00000000	10	110	1	00010	00000000	0
87	19CH1	0001011	11	00	0011100	00000000	10	000	1	00010	00000000	0
	(19SCH)	0001011	11	00	0011100	00000000	10	000	1	00010	00000000	0
88	1CCH1	0100000	11	11	0011101	11111111	10	000	1	00010	00000000	0
	(1CCCH)	0100000	11	11	0011101	11111111	10	000	1	00010	00000000	0
89	1DCH1	0011100	11	00	0011111	00000000	10	110	1	00010	00000000	0
	(1DCH)	0011100	11	00	0011111	00000000	10	110	1	00010	00000000	0

```

90 1FCHE 0001011 11 00 0011110 00000000 10 000 1 0 00010 00000000 0 0
   (01FECH)

91 1FCHE 0100010 11 11 0101110 11111111 /* R2=-145 */
   (01FECH) SDR(R2) FFI JZR(L00P); /* R2=-145 */

92 /* ENTRY POINT FOR NEW CORRELATION VALUE */
93 00FEH 1001100 11 00 0111101 00000000 10 000 1 0 00010 00000000 0 0
   (00FEH) LOOP! CLP(T) /* SUM=0 */

94 00DH 1000101 11 00 0000011 00000000 10 000 1 0 00010 00000000 0 0
   (000DH) LOOP! ILR(R5) /* RP TO AC */

95 03DH 0101000 11 11 0111010 11111111 10 000 1 0 00010 00000000 0 0
   (003DH) SDR(RA) FFI /* RA=RP */

```

RECORD NUMBER	CPE	FT	F0	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
96	03AH (003AH)	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
	IPINR:	LMI(R7)	FFI	RRM:									
	001011	11	11	0000100	00000000	10	110	1	0	00010	00000000	0	0
97	04AH (004AH)	11	00	0111001	00000000	10	200	1	0	00010	00000000	0	0
	IPINR2:	ACM(AC)	:	/* C(ΔI) TO AC */									
98	04RH (004RH)	11	11	0000011	00000000	10	110	1	0	00111	00000000	0	0
	LMI(RR)	FFI	RRM	ITCNT1:									
99	03RH (003RH)	11	00	1000011	00111111	10	200	1	0	00010	00000000	1	0
	A14(T)	MULT	K01FFF	JFL(LPINR,CONON):	/* T=T+(I-5)**2*/								
100	03RH (003RH)	11	00	0111110	00001000	10	200	1	0	01011	00000000	0	0
	LMI(R7)	K00008	ITCNT2:										
101	03FH (003FH)	11	11	1000100	00000000	10	110	1	0	00100	11110110	0	0
	LMI(R7)	FFI	RRM	LCNT1	M10	JFL(LPINC2,GOON):							
102	04RH (004RH)	11	00	0111100	00000000	10	200	1	0	00010	00000000	0	0
	GOON:	ILR(T)	:										
103	04CH (004CH)	11	11	1000101	00000000	10	111	1	0	01000	11110110	0	0
	LMI(R6)	FFI	RRM	LCNT2	M10	JFL(LOPOS,GOON2):							

104	05AH! (005AH)	LOOP! 000000	ILR(R0): 11 00 0111001	00000000	10	000	1	0	00010	00000000	0	0
105	059H! (0059H)		ALP(R7) J7R(LOOP): 11 00 0101110	11111111	10	000	1	0	00010	00000000	0	0
106	05RH! (005RH)	GOON2! 0000010	ILR(R2) ITCY4: 11 00 0111111	00000000	10	000	1	0	10011	00000000	0	0
107	05FH! (005FH)		ALR(R7) LCNT3 M10 JFL(GOLP.FND): 11 00 1000111	11111111	10	000	1	0	01100	11110110	0	0
108	07AH! (007AH)	GOLP! 1000001	CLR(R1) JZR(LOOP): 11 00 0101101	00000000	10	000	1	0	00010	00000000	0	0
109	07RH! (007RH)	FND! 1101101	NOP(AC) PAGE1 JZR(FETCH): 11 00 0101111	00000000	00	000	1	0	00010	00000000	0	0

RECORD NUMBER	CPE	FT	F0	JUMP	KRUSS	PAGEF	PBUS	PAUSFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
110	6543210	10	10	6543210	74543210	10	210	0	0	43210	74543210	0	0

```

111
112
113
114
115 /*MICROCODE TO LOCATE CORRELATION MIN*/
116
117
118
119 /* D1=T */
120 /* D2=MAX */
121 /* D0=JM-3 */
122 /* D5=JM-3 */
123 /* D6=COR+K */
124 /* D7=L-3 */
125 /* D8=J-3 */
126 /* D9=MIN */
074HI MJN: CLR(R6) PAGE3 LCNTI M10:
(0074H) 100110 11 00 001011 00000000 10 000 1 0 00100 11110110 0 0
    
```

127	174HI	LMI(R6)	K00012	RRM	LCNT2	M10:							
	(0174H)	0010110	11 00	0001000	00010010	10	110	1	0	01000	11110110	0	0
128	094HI	ACM(AC):											
	(0094H)	0001011	11 00	0001001	00000000	10	000	1	0	00010	00000000	0	0
129	094HI	SDR(R6)	FF1:										
	(0094H)	0100110	11 11	0001100	11111111	10	000	1	0	00010	00000000	0	0

```

I30 /* INITIALIZE REGISTERS */
I31 0C6H: CLR(AC) :
(00C6H) 1001101 11 00 0001101 00000000 10 000 1 0 00010 00000000 0 0

I32 0D6H: LMI(AC) K00004:
(00D6H) 0011101 11 00 0001110 00000100 10 000 1 0 00010 00000000 0 0

I33 0E6H: CMR(AC)/* -5 TO AC */
(00E6H) 1111101 11 00 0001111 00000000 10 000 1 0 00010 00000000 0 0

I34 0F6H: SDR(R0) FF1/*IM-5 TO R0 */
(00F6H) 0100000 11 11 0011101 11111111 10 000 1 0 00010 00000000 0 0

I35 106H: SDR(R5) FF1/*JM-5 TO R5 */
(0106H) 0100101 11 11 0011100 11111111 10 000 1 0 00010 00000000 0 0

I36 1C6H: SDR(R7) FF1/* I-5 TO R7 */

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPF	FT	FO	JJMP	KRUSS	PAGEF	CRUS	PAUSEFF	STRB	LPCNT	EMIT	MULTIPL	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(0106H)	0100111	11	11	0010110	11111111	10	000	1	0	00010	00000000	0	0
137	166H												
(0166H)	0101000	11	11	0010000	11111111	10	000	1	0	00010	00000000	0	0
138	106H												
(0106H)	0010110	11	00	0010001	00000000	10	110	1	0	00010	00000000	0	0
139	116H												
(0116H)	0001011	11	00	0010010	00000000	10	000	1	0	00010	00000000	0	0
140	126H												
(0126H)	0101001	11	11	0010011	11111111	10	000	1	0	00010	00000000	0	0
141	136H												
(0136H)	0100010	11	11	0010100	11111111	10	000	1	0	00010	00000000	0	0
142	146H												
(0146H)	1000001	11	00	0001010	00000000	10	000	1	0	00010	00000000	0	0
143	046H												
(0046H)	0010001	11	00	0000001	00001000	10	000	1	0	00010	00000000	0	0

```

144 /* BEGIN COMPUTATION LOOP*/
145 016H: L00P2I LMI(R8) FFI: /* J=J+1 */
      (0016H) 0011000 11 11 0000010 00000000 10 000 1 0 00010 00000000 0 0 0
146 026H: L00P1I LMI(R7) FFI: /* I=I+1 */
      (0026H) 0010111 11 11 0000011 00000000 10 000 1 0 00010 00000000 0 0 0
147 036H: L00P0I LMI(R6) FFI RRM:
      (0036H) 0010110 11 11 0000100 00000000 10 110 1 0 00010 00000000 0 0 0
148 /* TEST FOR MAXIMUM */
149 046H: LCM(T): /*COMP OF DATA T> T */
      (0046H) 1111010 11 00 0000101 00000000 10 000 1 0 00010 00000000 0 0 0
150 056H: ILR(R2):
      (0056H) 0000010 11 00 0000110 00000000 10 000 1 0 00010 00000000 0 0 0
151 066H: ADR(T) FFI:

```

XMAS VERS 2.0 CORRELATION MICROCODE ERRORS= 0 PAGE A

QFCORN NUMBER	CPE	FT	FO	HMP	KRUSS	PAGEF	CRUS	PAUSFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
(00664H)	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(00664H)	0111100	11	11	0110101	11111111	10	000	1	0	00010	00000000	0	0
152	0654H (00654H)	1101101	11	00	1000001	00000000	10	110	1	0	00010	00000000	0
153	0124H (00124H)	0001011	11	00	0110100	00000000	10	000	1	0	00010	00000000	0
154	0144H (00144H)	0100010	11	11	0101001	11111111	10	000	1	0	00111	00000000	0
155	0134H (00134H)	1111010	11	00	0110101	00000000	10	000	1	0	00010	00000000	0
156	0154H (00154H)	0001001	11	00	0000010	00000000	10	000	1	0	00010	00000000	0
157	0254H (00254H)	0111100	11	11	0000011	11111111	10	000	1	0	00010	00000000	0
159	0354H (00354H)	1101101	11	00	1000010	00000000	10	110	1	0	00010	00000000	0

159	023H!	CLEMN! T2P(T) FF0:	1011100 11 00 0110100 11111111 10 000	1	0	00010 00000000	0	0
	(0023H)							
160	024H!	NOP(AC) RRM JFL(ZZR0,NTZZR0) :	1101101 11 00 1000011 00000000 10 010	1	0	00010 00000000	0	0
	(0024H)							
161	022H!	DGT4N! NOP(R1) ITCNT1 JZR(NODF) :	1100001 11 00 0101001 00000000 10 000	1	0	00111 00000000	0	0
	(0022H)							
162	032H!	Z R0! NOP(T) :	1101100 11 00 0110100 00000000 10 000	1	0	00010 00000000	0	0
	(0032H)							
163	034H!	NOP(T) :	1101100 11 00 0000110 00000000 10 000	1	0	00010 00000000	0	0
	(0034H)							
164	064H!	ILR(R7) ! /* I-5 TO AC */	0000111 11 00 0001000 00000000 10 000	1	0	00010 00000000	0	0
	(0064H)							

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CAUS	PAUSEFF	STRB	LPCNT	EMIT	MULTIPL	SPRF	
165	6543210	10	10	6543210	74543210	10	010	0	0	43210	74543210	0	0	
166	/* TAKE ABSOLUTE VALUE */													
165	094H!	TZR(AC) KR0000 INH!												
	(094H)	1011101	11	00	0001001	10000000	10	001	1	0	00010	00000000	0	0
167	NOP(T) JFL(POSS*NEGG):													
167	094H!	NOP(T) JFL(POSS*NEGG):												
	(094H)	1101100	11	00	1001010	00000000	10	000	1	0	00010	00000000	0	0
168	POSS! NOP(AC) JCR(OVER):													
168	0A2H!	POSS! NOP(AC) JCR(OVER):												
	(0A2H)	1101101	11	00	0110101	00000000	10	000	1	0	00010	00000000	0	0
169	NEGG! CHR(AC):													
169	0A3H!	NEGG! CHR(AC):												
	(0A3H)	1111101	11	00	0110100	00000000	10	000	1	0	00010	00000000	0	0
170	INR(AC) FFI:													
170	0A4H!	INR(AC) FFI:												
	(0A4H)	0111101	11	11	0110101	00000000	10	000	1	0	00010	00000000	0	0
171	-VER! SDR(T) FFI: /*ARS(I-5) TO T */													
171	0A5H!	-VER! SDR(T) FFI: /*ARS(I-5) TO T */												
	(0A5H)	0101100	11	11	0001011	11111111	10	000	1	0	00010	00000000	0	0
172	ILR(RB)!/* J-5 TO AC */													
172	0A5H!	ILR(RB)!/* J-5 TO AC */												
	(0A5H)	0001000	11	00	0001101	00000000	10	000	1	0	00010	00000000	0	0
173	/* TAKE ABSOLUTE VALUE */													
174	0D5H!	TZR(AC) KR0000 INH!												
	(0D5H)	1011101	11	00	0001110	10000000	10	001	1	0	00010	00000000	0	0


```

175 0E5H1  NOP(AC) JFL(POS2,NEG2);
      (00E5H) 1101101 11 00 1001011 00000000 10 000 1 0 00010 00000000 0 0

176 0B2H1  POS2; ALP(T) JCR(V2); /* ABS(J-3)+ABS(I-3) TO AC.T*/
      (00B2H) 0001100 11 00 0110100 1111111 10 000 1 0 00010 00000000 0 0

177 0B3H1  NEG2; CMR(AC);
      (00B3H) 1111101 11 00 0110001 00000000 10 000 1 0 00010 00000000 0 0

178 0B1H1  ALP(T) FFI; /*ABS(J-5)+ABS(I-5) TO AC.T*/
      (00B1H) 0001100 11 11 0110100 1111111 10 000 1 0 00010 00000000 0 0

179 0B4H1  O.2; ILR(R1);
      (00B4H) 0000001 11 00 0001100 00000000 10 000 1 0 00010 00000000 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STPR	LPCNT	EMIT	MULTIPL	SPRF
180	0C4H! (00C4H)	1111101	11	00	0001101	00000000	10	000	1	0	00010	00000000	0
					CMP(AC):								
181	0D4H! (00D4H)	0001100	11	11	0001110	11111111	10	001	1	0	00010	00000000	0
					ALD(T) FFI INVH:								
182	0E4H! (00E4H)	1101101	11	00	1001000	00000000	10	000	1	0	00010	00000000	0
					NOP(AC) JFL(NLTO +NTGEO):								
183	0A2H! (00A2H)	0001100	11	00	0110000	00000000	10	000	1	0	00010	00000000	0
					NLTO: ILR(T):								
184	0A0H! (00A0H)	0100001	11	11	0110001	11111111	10	000	1	0	00010	00000000	0
					SDR(R1) FFI: /* T=V TO R1*/								
185	0A1H! (00A1H)	0000111	11	00	0000001	00000000	10	000	1	0	00010	00000000	0
					ILR(R7):								
186	011H! (0011H)	0100000	11	11	0000010	11111111	10	000	1	0	00010	00000000	0
					SDR(R0) FFI: /* I4-5 TO I-5*/								
187	021H! (0021H)	0001000	11	00	0000101	00000000	10	000	1	0	00010	00000000	0
					ILR(RA):								

```

188 0514I  SDR(R5) FFI ITCNT1 JZR(NODE); /* JMTJ */
(0051H) 0100101 11 11 0101001 11111111 10 000 1 0 00111 00000000 0 0

189 0834I  NTR001 NOP(AC) ITCNT1 JZR(NODE);
(0083H) 1101101 11 00 0101001 00000000 10 000 1 0 00111 00000000 0 0

190 0134I  NT77001 ACM(AC);
(0013H) 0001011 11 00 0110111 00000000 10 000 1 0 00010 00000000 0 0

191 0174I  SDR(R9) FFI;
(0017H) 0101001 11 11 0000001 11111111 10 000 1 0 00010 00000000 0 0

192 0174I  ILR(R7);
(0017H) 0000111 11 00 0000010 00000000 10 000 1 0 00010 00000000 0 0

193 0274I  SDR(R0) FFI; /*I4-5-I-5 */
(0027H) 0100000 11 11 0000100 11111111 10 000 1 0 00010 00000000 0 0

```

XMAS VERS 2.0 CORRELATION MICROCODE

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CAUS	PAUSFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
194	047H: (0047H)	0001000	11 00	0110011	00000000	10	000	1	0	00010	00000000	0	0
				ILP(RR):									
195	043H: (0043H)	0100101	11 11	0001100	11111111	10	000	1	0	00010	00000000	0	0
				SRR(R5) FFL:/* JM-5-J-5 */									
196	0C3H: (00C3H)	0000000	11 00	0111111	00000000	10	000	1	0	00010	00000000	0	0
				ILP(R0): /*IM-3 TO AC */									
197	0CFH: (00CFH)	1011101	11 00	0010000	10000000	10	001	1	0	00010	00000000	0	0
				I7R(AC) K0000 IVH:									
198	10FH: (010FH)	1101101	11 00	1000000	00000000	10	000	1	0	00010	00000000	0	0
				NOP(AC) JFL(POSS,NEGGS):									
199	10RH: (010RH)	1111101	11 00	0110101	00000000	10	000	1	0	00010	00000000	0	0
				NEGGS: CMP(AC):									
200	105H: (0105H)	0111101	11 11	0111010	00000000	10	000	1	0	00010	00000000	0	0
				INR(AC) FFL:									
201	10AH: (010AH)	0111100	11 11	0111110	11111111	10	000	1	0	00010	00000000	0	0
				POSS: SRR(T) FFL: /* ARS(IM-5) TO T */									

202	10FH: (010EH)	NOP(T): 1101100 11 00 0001111 00000000 10 100	1	0	00010 00000000	0	0
203	0FCH: (00FEH)	IUR(RS): /* JM-5 TO AC */ 0000101 11 00 0111011 00000000 10 100	1	0	00010 00000000	0	0
204	0FRH: (00FRH)	IZR(AC) K90000 IVH: 1011101 11 00 0110101 10000000 10 101	1	0	00010 00000000	0	0
205	0FSH: (00F5H)	NOP(AC) JFL(POS1+NEG1): 1101101 11 00 1001111 00000000 10 100	1	0	00010 00000000	0	0
206	0F7H: (00F7H)	NEG1 CMR(AC): 1111101 11 00 0110100 00000000 10 100	1	0	00010 00000000	0	0
207	0F4H: (00F4H)	INR(AC) FF1: 0111101 11 11 0110010 00000000 10 100	1	0	00010 00000000	0	0

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRR	LPCNT	EMIT	MULTIPL	SPRF
209	0F2H (00F2H)	001100	11 00	0110001	11111111	10 000	1	0	00010	00000000	0	0	0
/* TEST FOR I=9 */													
210	0F1H (00F1H)	010001	11 11	0101001	11111111	10 000	1	0	00111	00000000	0	0	0
SOP(R1) FF1 ITCNT1 JZR(NODE):													
211	099H (0099H)	1101101	11 00	1000110	00000000	10 000	1	0	00010	00000000	0	0	0
NOP(AC) JFL(TNE2,IFQ2):													
212	06AH (006AH)	1101100	11 00	0000010	00000000	10 000	1	0	00010	00000000	0	0	0
TNE2: NOP(T):													
213	02AH (002AH)	1101100	11 00	0110110	00000000	10 000	1	0	00010	00000000	0	0	0
NOP(T) JCP(LOOP1):													
214	06RH (006RH)	1000111	11 00	0110010	00000000	10 000	1	0	00100	11110110	0	0	0
IFQ2: CLP(R7) LCNT1 M10:													
215	062H (0062H)	0010111	11 00	0110011	00000100	10 000	1	0	00010	00000000	0	0	0
LMT(R7) K00004:													
216	063H (0063H)	1110111	11 00	0111100	00000000	10 000	1	0	01011	00000000	0	0	0
CMP(R7) ITCNT2: /* -5 TO 97 */													

```

217 /* TEST FOR J=9 */
218 04CH! NOP(AC) JFL(JNE2,JF02);
(006CH) 1101101 11 00 1000001 0000000 10 000 1 0 00010 0000000 0 0 0
219 01AH! JNE2! NOP(T) JCR(L00P2);
(001AH) 1101100 11 00 0110110 0000000 10 000 1 0 00010 0000000 0 0 0
220 01RH! JEQ2! NOP(T);
(001RH) 1101100 11 00 0111100 0000000 10 000 1 0 00010 0000000 0 0 0
221 /* WRITE RESULTS OF MACRO MEMORY */
222 01CH! CLR(AC);
(001CH) 1001101 11 00 0000010 0000000 10 000 1 0 00010 0000000 0 0 0
223 02CH! LMI(AC) K00017;
(002CH) 0011101 11 00 0111011 00010111 10 000 1 0 00010 0000000 0 0 0

```

RECORD NUMBER	CPF	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSEF	STRR	LPCNT	EMIT	MULTIPL	SPRF
224	029H! (0029H)	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
	TLR(R9) RWM: /* MIN TO AC*/						111	1	0	00010	00000000	0	0
225	020H! (0020H)	10	00	0000110	00000000	10	000	1	0	00010	00000000	0	0
	CLP(AC):												
226	060H! (0060H)	10	00	0001001	00010110	10	000	1	0	00010	00000000	0	0
	LMI(AC) K00016:												
227	090H! (0090H)	10	00	0001110	/*MAX TO AC*/	10	111	1	0	00010	00000000	0	0
	ILP(R2) RWM:												
228	0E0H! (00E0H)	10	00	0010011	00000000	10	000	1	0	00010	00000000	0	0
	CLR(AC):												
229	130H! (0130H)	10	00	0111110	K0001A	10	000	1	0	00010	00000000	0	0
	LMI(AC) K0001A :												
230	13EH! (013EH)	10	00	0111110	00000000	10	111	1	0	00010	00000000	0	0
	ILP(R0) RWM:												
231	13CH! (013CH)	10	00	0011010	00000000	10	000	1	0	00010	00000000	0	0
	CLP(AC):												

232	LACH! (01ACH)	0011101	LMI(AC) K0001B: 11 00 0111011	00011011	10	000	1	0	00010	00000000	0	0
233	IRCH! (01RCH)	0000101	ILR(RS) RWM PAGE1 J7R(FFTCH): 11 00 0111111	00000000	00	111	1	0	00010	00000000	0	0
234	00FH! (000FH)	1101101	FETCHI NOP(AC) J7R(FETCH): 11 00 0111111	00000000	10	000	1	0	00010	00000000	0	0

235 EOF

NO PROGRAM ERRORS
END OF PROGRAM

XMAS VERS 2.0 CORRELATION MICROCODE

MICROPROGRAM MEMORY IMAGE

	04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
0004										JFL				JCC	JCR	J7R
										006A4				003D4	000D4	000F4
										211				04	03	234
										5				2	2	3
0014	JCC	JCR	JCR	J7R	JCC	JCC	JCC	JCC			JCR	JCR	JCC			
	00214	00144	00154	00094	00254	00264	00274				00164	001C4	002C4			
	186	153	155	154	156	145	192				219	220	222			
	1	1	1	1	1	2	1				1	1	1			
0024	JCC	JCR	JCR	JFL	JCC	JCC	JCC				JCR	JCR	JCC			
	00514	00094	00244	00324	00354	00364	00474				00264	002D4	002E4	004D4		
	187	161	159	160	157	146	193				213	224	223	225		
	1	1	1	1	1	2	1				1	1	1	1		
0034	JCR	JCR	JCC	JCC	JFL	JCC	JCC				JCC	JCR	JFL	JCR	JFL	
	00364	00374	00664	00174	00224	00664	00174			JFL	00464	003F4		003A4	004A4	
	162	190	163	163	158	147	191			99	96	100		95	101	
	1	1	1	1	1	1	1			1	2	1		1	1	
0044	JCC	JCC	JCC	JCR	JCC	JCC	JCR			JCC	JCR	JCR	JFL			
	00C34					00564	00434			00394	00494	004C4	005A4			
	195	195	149	194	149	194	194			98	97	102	103			
	1	1	1	1	1	1	1			1	2	1	1			
0054	J7P	JCC	JCC	JCC	JCC	JCC	JCC			J7R	JCR	JCR	JCC			JFL
	00094					00664				000F4	00594	005F4				007A4
	188					150				105	104	106				107
	1					1				1	1	1				1
0064	JCR	JCR	JCC	JCC	JFL	JCC	JCC				JCC	JCR	JFL	JCC	JCC	
	00634	006C4	00864	00124	00554						002A4	00624	001A4	009D4		
	215	216	166	152	151						212	214	214	226		
	1	1	1	1	1						1	1	1	1		
0074	JCC	JCC	JCC	JCC	JCC	JCC	JCC				J7R	J7R	JCC	JCC	JCC	
	00764					01764					000D4	000F4			00AF4	
	126					126					108	109			70	
	0					0					1	1			0	

XMAS VERS 2.0 CORRELATION MICROCODE

MICROPROGRAM MEMORY IMAGE

0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH
JCR	JCC	JCR	JZR	JCC	JCC	JCC								JCC	
0081H	0011H	0080H	0009H	0094H		0096H								009FM	
184	185	183	189	166	128									71	
				JFL	JCC	JCC							JCC	JCC	
				0042H	00C6H								00EDH	00AEH	
				167	129								227	72	
		JCR	JCR	JCR	JCC	JCC								JCC	
		00A5H	00A4H	00A5H	0085H	0016H								00RFH	
		168	169	170	171	143								73	
	JCR	JCR	JCR	JCC	JCC									JCC	
	0084H	0084H	0091H	00C4H	0075H									00CEH	
	178	176	177	179	172									74	
			JCR	JCC	JCC	JCC								JCC	JCC
			00CFH	00D4H	00D6H									00DEM	010FH
			196	180	131									75	197
				JCC	JCC	JCC						JCC		JCC	
				00E4H	00F5H	00E6H						015CH		00EEM	
				181	174	132								76	
				JFL	JFL	JCC								JCC	JCR
				00A2H	00A2H	00F6H								00FCH	013DH
				182	175	133								77	
	JZR	JCR	JCR	JCR	JFL	JCC						JCR		JCR	
	0009H	00F1H	00F4H	00F2H	00F2H	0106H						00F5H	00DCH	00F8H	
	210	208	206	207	205	134						204	79	203	

XMAS VERS 2.0 CORRELATION MICROCODE

MICROPROGRAM MEMORY IMAGE

	04	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH	
010H	JCR 010AH	JCR 010AH	JCC 0116H	JCR 010AH	JCR 010AH	JCR 010AH	JCR 010AH	JCR 0105H	JCR 0105H	JCC 00FEH	JFL 010AH	JFL 010AH	
	200	13A	13A	201	199	202	19A	1	1	1	1	1	
011H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H	JCC 0126H
012H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H	JCC 0136H
013H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H	JCC 0146H
014H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H	JCC 00A6H
015H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H	JCC 0106H
016H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H	JCC 00R6H
017H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H	JCC 0177H

XMAS VERS 2.0 CORRELATION MICROCODE

MICROPROGRAM MEMORY IMAGE

0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH
018H												JCC 019CH			
												96 1			
019H												JCC 01CCH			
												97 1			
01AH												JCC 01RCH			
												232 1			
01BH												JZR 000FH			
												233 1			
01CH						JCC 0166H						JCC 01DCM			
						136 1						98 1			
01DH					JCC 01C6H							JCC 01FCM			
					175 1							99 1			
01EH												JZR 000EH			
												1e 1			
01FH												JCC 01ECH			
												90 1			

XMAS VERS 2.0 CORRELATION MICROCODE

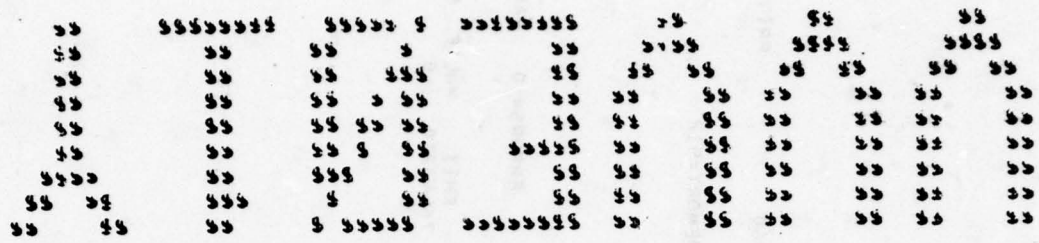
CROSS REFERENCE DIRECTORY

LABEL	REFERENCES
CONON	99.(100)
COR	(70)
GTMN	158.(161)
GLEMN	158.(159)
FND	107.(109)
FETCH	109,233,(234),234
GOLP	107.(108)
SOON	101.(102)
SOONP	103.(106)
TEQ2	211.(214)
JNE2	211.(212)
JEQ2	218.(220)
JNE2	218.(219)
LOOP	91.(93),105
LOOP1	(146),213
LOOP2	(145),210
LOOP3	(94),108
LOPOP	103.(104)
LPIR	(96),99
LPIR2	(97),101
MGED	152.(155)
MTN	(126)
MLTD	152.(153)
VEG1	205.(206)
VEG2	175.(177)
VEGG	167.(169)
VEGGG	198.(199)
MODE	154,161,188,189,210,(211)
VTGEN	182.(189)
VTLT0	182.(183)
VTZR0	160.(190)
AV2	176.(179)
OVER	168.(171)
POS1	205.(208)
POS2	175.(176)
POSS	167.(168)
POSSS	198.(201)
ZZR0	160.(162)

APPENDIX C

Pacer Emulation Microcode (Page 1)

(LISTING #4)



XMAS VERS 2.0

ERRORS= 0 PAGE 1

\$LISTFILF=2
 \$CROSSREF
 \$BITS
 \$IMAGE
 \$WIDTH=132
 \$TITLE=PACER EMULATOR MICROCODE FOR INTEL 3000
 \$DISPLAY(ALL)
 \$LEFT=1
 \$FORMS=0
 \$RIGHT=72
 \$TITLE=PACER EMULATOR MICROCODE FOR INTEL 3000
 \$LISTFILF=2
 \$CROSSREF=1
 \$WIDTH=132

IMAGE=1
 LINES=60
 PRINT=1
 SOURCEFILF=2
 MICROMEMORY=512

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 2

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRR	LPCNT	EMIT	MULTF	SPDF
1	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													

FIELD LENGTH=A
 MICROPS(K00000=00H K00001=01H <00002=02H
 K00003=03H K00004=04H
 K00005=05H <00006=06H
 K00007=07H <00008=08H
 K00009=09H <0000A=0AH
 K0000R=0BH <0000C=0CH
 K0000D=0DH <0000E=0EH
 K0000F=0FH <00010=10H
 K00011=11H <00012=12H
 K00013=13H <00014=14H
 K00015=15H <00016=16H
 K00017=17H <00018=18H

```

15                                     K0001F=1FH <01FFF=7FH
16                                     K00019=19H <0001A=1A4
17                                     K0001R=1RH <0001C=1CH
18                                     K0001D=1DH <0001E=1EH
19                                     K00000=00H <7FFFF=7FH
20                                     KFFFFF=7770 KFFFFA=0FAH);
21 KRUISS      KRUISS;
22
23 /* ALL MICROCODE IS ON PAGE 1 */
24 PAGEFF      FIELD LENGTH=2          DEFAULT=0
25                                     MICROPS(PAGE1=00R PAGE2=01R
26                                     PAGE3=10R PAGE4=11R);
27
28 CRUIS      FIELD LENGTH=3          DEFAULT=0
29 MICROPS(NR0=000R INH=001R  QMW=010R  CMB=011R
30          RTN=100R  QNT=101R  QPM=110R  QCM=111R);
31
32 /*      N00      NORUS OPERATION
33          INH      INHIBIT CPE ARRAY
34          R W      READ-MODIFY-WRITE
35          CMB      CPU NEEDS BUS
36          R M      REQUEST INPUT
37          R T      REQUEST OUTPUT
38          RDM      REQUEST READ MEMORY
39          RWM      REQUEST WRITE MEMORY
40
41
42 PAUSFF      FIELD LENGTH=1          DEFAULT=1R
43                                     MICROPS(SPO=0);
44
45 STRR      FIELD LENGTH=1          DEFAULT=0
46                                     MICROPS(STR0BF=1R);
47
48 LPCNT      FIELD LENGTH=5          DEFAULT=00010R
49 MICROPS(CO=00010R LCNT1=00100R LCNT2=01000R
50          LCNT3=01100R LCNT4=10000R ITCNT1=00111R
51          ITCNT2=01011R ITCNT3=01111R ITCNT4=10011R);
52
53 EMTT      FIELD LENGTH=8          DEFAULT=0
54 MICROPS (M1=0FFH M2=0FEH M3=0FDH M4=0FCH

```

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTFL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPQF
55	6543210	10	10	6543210	74543210	10	210	0	0	43210	76543210	0	0

M5=0F84 M6=0F44 M7=0F04 M8=0F84 M9=0F74 M10=0F64
M11=0F54 M12=0F44 M13=0F34 M14=0F24 M15=0F14
M16=0F04 M17=0FFH M18=0FEH

MULTIFIELD LENGTH=1 MICROPS(MULT=19); DEFAULT=08

SPRF FIELD LENGTH=1 MICROPS(SPARE=01H); DEFAULT=0

A STRING 'R0':
X STRING 'R1':
Q STRING 'R2':
P STRING 'R3':
S STRING 'R4':
R STRING 'R5':
E STRING 'R6':
W STRING 'R7':

73 00H! INIT! CLR(A);
74 (0000H) 1000000 11 00 0000001 00000000 00 000 1 0 00010 00000000 0 0

75 10H!
(0010H) 1000001 11 00 0000010 00000000 00 000 1 0 00010 00000000 0 0

76 20H!
(0020H) 1000111 11 00 0000100 00000000 00 000 1 0 00010 00000000 0 0

```

77 3041 0121 LMT(AC) RRM JPR(LA STA LX STX AI SF 4 0 ADM ORI XOR1
78 (00304) 0011101 11 00 1100111 00000000 00 000 1 0 00010 00000000 0 0 0

79 4041 INIT1 CLP(T) JCC(INIT2);
(00404) 1001100 11 00 0001001 00000000 00 000 1 0 00010 00000000 0 0 0

80 6041 SF1 NOP(A) JCF(NTEQL+EQL);
(00604) 1100000 11 00 1010010 00000000 00 000 1 0 00010 00000000 0 0 0

81 7041 LA1 ACM(AC);
(00704) 0001011 11 00 0001000 00000000 00 000 1 0 00010 00000000 0 0 0

82 8041 SDR(A) FF1 JZR(FETCH);
(00804) 0100000 11 11 0101111 11111111 00 000 1 0 00010 00000000 0 0 0

83 9041 INIT2 LMT(T);
(00904) 0011100 11 00 0001010 00000000 00 000 1 0 00010 00000000 0 0 0

```

AD-A071 638

ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND REDSTO--ETC F/G 9/2
APPLICATION OF A MICROPROGRAMMED, BIT SLICE MICROPROCESSOR TO T--ETC(U)
SEP 78 L G MINOR

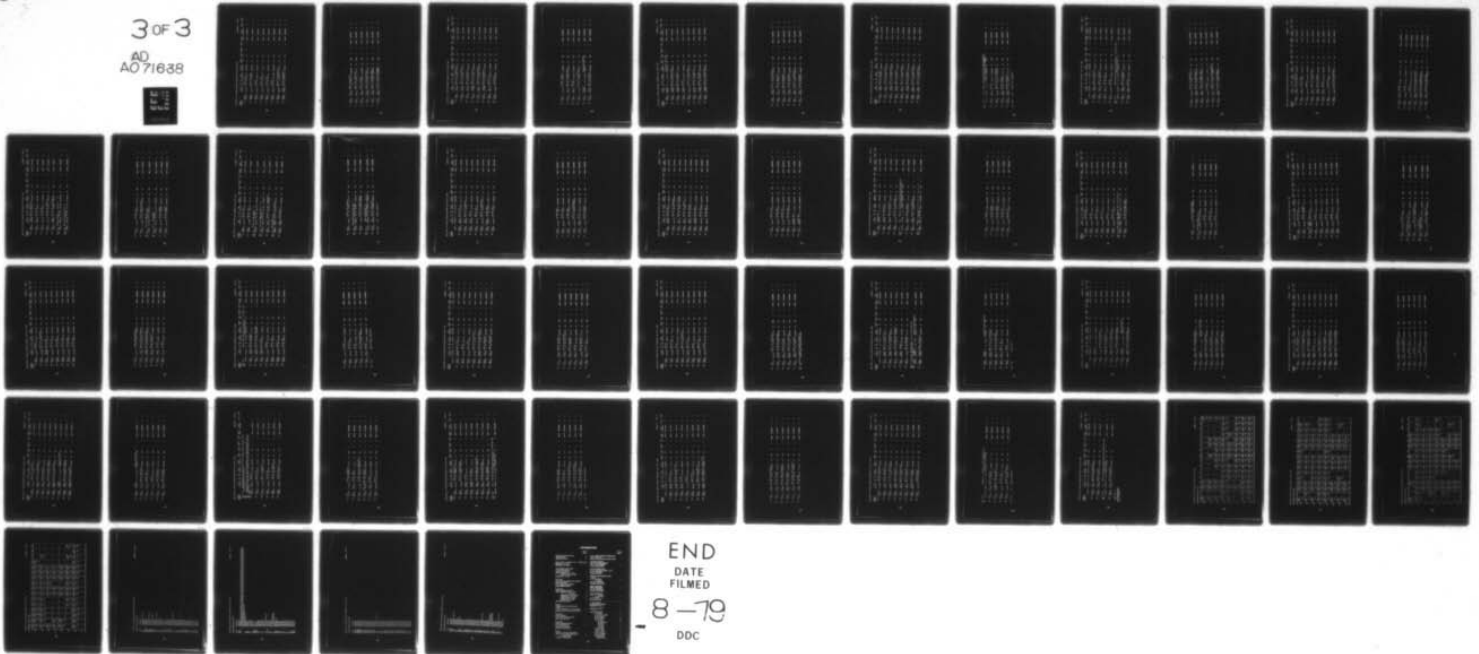
UNCLASSIFIED

DRDMI-T-78-53

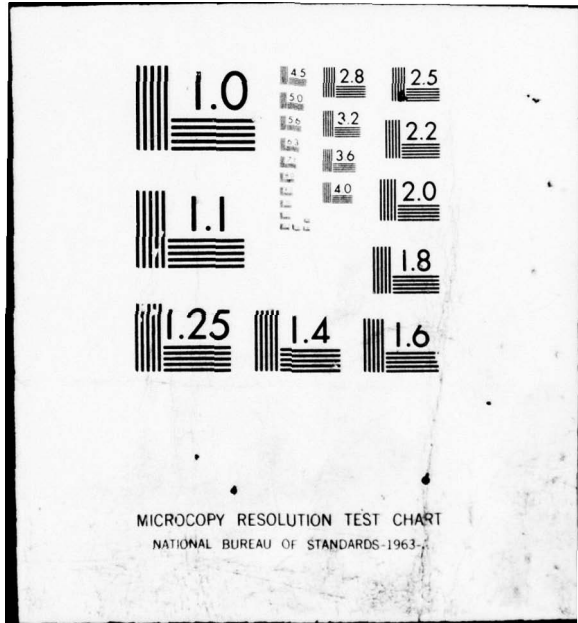
NL

3 OF 3

AD
A071638



END
DATE
FILMED
8-79
DDC



XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 4

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CUS	PAUSEF	STPR	LPCNT	EMIT	MULTF	SDRF
84	0A0H (00A0H)	0000111	11	00	0001011	00	101	1	0	00010	00000000	0	0
				ILR(W) ROT:									
85	0B0H (00B0H)	0011100	11	11	0001100	00	110	1	0	00010	00000000	0	0
				LMI(T) FFI RRM:									
86	0C0H (00C0H)	0001011	11	00	0001101	00	000	1	0	00010	00000000	0	0
				ACM(AC):									
87	0D0H (00D0H)	0100100	11	11	0001110	00	000	1	0	00010	00000000	0	0
				SDR(S) FFI:									
88	0E0H (00E0H)	0011100	11	00	0001111	00	110	1	0	00010	00000000	0	0
				LMI(T) RRM:									
89	0F0H (00F0H)	0001011	11	00	0010000	00	000	1	0	00010	00000000	0	0
				ACM(AC):									
90	100H (0100H)	0100011	11	11	0101111	00	000	1	0	00010	00000000	0	0
				SDR(P) FFI JZR(FETCH):									
91	110H (0110H)	1000000	11	00	0101111	00	000	1	0	00010	00000000	0	0
				CLR(I) CLR(A) JZR(FETCH):									

92	IF04I (01F04)	N1 110000	NOPI(A) JZR(FETCH): 11 00 0101111 0000000	00	000	1	0	00010	00000000	0	0
93	314I (00314)	J1 010011	SDR(P) FFI JZR(FETCH): 11 11 010111 1111111	00	000	1	0	00010	00000000	0	0
94	514I (00514)	SG1 110000	NOPI(A) JCC(SSC): 11 00 0001001 0000000	00	000	1	0	00010	00000000	0	0
95	714I (00714)	ST1 000000	ILR(A) RWM JZR(FETCH): 11 00 010111 0000000	00	011	1	0	00010	00000000	0	0
96	914I (00914)	SG1 110000	NOPI(A) JCF(NTFQL2*EQL2): 11 00 1010001 0000000	00	000	1	0	00010	00000000	0	0
97	1114I (01114)	A01 000000	ILR(A) FFI STC JZR(FETCH): 10 11 010111 0000000	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	F0	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	FMIT	MULTF	SPDF
98	131H (0171H)	6543210	10	10	6543210	76543210	10	0	0	43210	76543210	0	0
	Y!	SDR(A)	FF1	JZR(FETCH):			00	1	0	00010	00000000	0	0
		0100000	11	11	0101111	11111111	00	000	1	0	00010	00000000	0
99	151H (0151H)	0100000	11	11	0101111	11111111	00	000	1	0	00010	00000000	0
	W!	SDR(A)	FF1	JZR(FETCH):			00	1	0	00010	00000000	0	0
		0100000	11	11	0101111	11111111	00	000	1	0	00010	00000000	0
100	1E1H (01E1H)	N2!	NOP(A)	JZR(FETCH):			00	1	0	00010	00000000	0	0
		1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0
101	024 (0002H)	NTEQ3!	NOP(A)	JZF(LT,NTLT):			00	1	0	00010	00000000	0	0
		1100000	11	00	1011001	00000000	00	000	1	0	00010	00000000	0
102	12H (0012H)	LI!	ILR(P)	FF1	JZR(FETCH):		00	1	0	00010	00000000	0	0
		0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0
103	22H (0022H)	NTEQ1!	NOP(A)	JZR(FETCH):			00	1	0	00010	00000000	0	0
		1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0
104	32H (0032H)	LI	LMI(S)	FF1:			00	1	0	00010	00000000	0	0
		0010100	11	11	0000100	00000000	00	000	1	0	00010	00000000	0
105	42H (0042H)	ILR(P)	RWM:				00	1	1	0	00010	00000000	0
		0000011	11	00	0000101	00000000	00	111	1	0	00010	00000000	0

106	5241	ILR(P9) JCR(CL):	0001001	11 00 0110101	00000000	00	000	1	0	00010	00000000	0	0
	(00524)												
107	6241	SL: NOP(A) JCF(NTF23*EQ3):	1100000	11 00 1010000	00000000	00	000	1	0	00010	00000000	0	0
	(00624)												
108	7241	Lx: ACM(AC):	0001011	11 00 0001000	00000000	00	000	1	0	00010	00000000	0	0
	(00724)												
109	8241	SDR(X) FFI JZR(FETCH):	0100001	11 11 0101111	11111111	00	000	1	0	00010	00000000	0	0
	(00824)												
110	9241	NTFOL2: NOP(A) JZF(NTGT*GT):	1100000	11 00 1011010	00000000	00	000	1	0	00010	00000000	0	0
	(00924)												
111	0A241	NTGT: NOP(A) JZR(FETCH):	1100000	11 00 0101111	00000000	00	000	1	0	00010	00000000	0	0
	(00A24)												

XMAS VERS 2.0 PAGED EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	CIUS	PAUSEFF	STRR	LPCNT	FMIT	MULTF	SPRF
112	002H (0002H)	N FOL4! 0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0
113	0E2H (00E2H)	NTE05! NOP(A) JZF(NTGT5.ST5); 1100000	11	00	1011111	00000000	00	000	1	0	00010	00000000	0
114	0F2H (00F2H)	NTGT5! NOP(A) JZR(FETCH); 1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0
115	112H (0112H)	CA0! CLR(A); 1000000	11	00	0010010	00000000	00	000	1	0	00010	00000000	0
116	122H (0122H)	ILR(A) FFI JZR(FETCH); 0000000	11	11	0101111	00000000	00	000	1	0	00010	00000000	0
117	132H (0132H)	OUT2! ALR(AC) FFZ JCR(VV); 0001101	11	10	0110001	11111111	00	000	1	0	00010	00000000	0
118	142H (0142H)	OUT3! NOP(A) JZR(FETCH); 1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0
119	162H (0162H)	NTE06! NOP(A) JZF(NTGT6.ST6); 1100000	11	00	1011111	00000000	00	000	1	0	00010	00000000	0

120	1724: (01724)	NTGTA: ILR(P) FFI J7P(FETCH): 0000011 11 11 0101111 00000000	00	000	1	0	00010	00000000	0	0
121	1824: (01824)	N3: NOP(A) JZR(FETCH): 110000 11 00 0101111 00000000	00	000	1	0	00010	00000000	0	0
122	034: (00034)	E03: NOP(A) JZR(FETCH): 110000 11 00 0101111 00000000	00	000	1	0	00010	00000000	0	0
123	134: (00134)	NLT: NOP(A) JZR(FETCH): 110000 11 00 0101111 00000000	00	000	1	0	00010	00000000	0	0
124	234: (00234)	E01: ILR(P) FFI J7R(FETCH): 0000011 11 11 0101111 00000000	00	000	1	0	00010	00000000	0	0
125	334: (00334)	REG: ILR(A) JCC(REQ): 0000000 11 00 0010000 00000000	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 7

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STPR	LPCNT	EMIT	MULTF	SPRF
126	6543210	10	10	6543210	76543210	10	010	0	0	43210	76543210	0	0
(0063H)	SNE!	NOP(A)	JCC(SVFC):										
127	1100000	11	00	0001000	00000000	00	000	1	0	00010	00000000	0	0
(0073H)	STX!	ILR(X)	RWM	JZR(FETCH):									
128	0000001	11	00	0101111	00000000	00	111	1	0	00010	00000000	0	0
(0083H)	SNE!	NOP(A)	JCF(NTEQL4+EQL4):										
129	1100000	11	00	1010101	00000000	00	000	1	0	00010	00000000	0	0
(0093H)	EQL2!	NOP(A)	JZR(FETCH):										
130	0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0	0
(00A3H)	GI!	ILR(P)	FFI	JZR(FETCH):									
131	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
(00B3H)	EQL4!	NOP(A)	JZR(FETCH):										
132	0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0	0
(00C3H)	EQL5!	ILR(P)	FFI	JZR(FETCH):									
133	0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0	0
(00D3H)	GTS!	ILR(P)	FFI	JZR(FETCH):									

	P-GC	NOP(A)	JPR(CLRI ADA CAO TCA ARS ALS LMS SSP									
134	1134	1134	1134	1134	1134	1134	1134	1134	1134			
135												
			SSN EX EP ES ICX DCX NOMB PZ)	00	000	1	0	00010	00000000	0	0	
		TCA	CMP(A)	00	000	1	0	00010	00000000	0	0	
			ILP(A) FF1 STC JZR(FETCH)	00	000	1	0	00010	00000000	0	0	
		ACM2	DSM(R9) JCR(XX)	00	000	1	0	00010	00000000	0	0	
			ACM3	DSM(R9) JCR(BACK)	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 8

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
(0163H)	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
	0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0	0
141	173H	GT6!	NOP(A)	JZR(FETCH);									
(0173H)	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
142	1E3H	N4!	NOP(A)	JZR(FETCH);									
(01E3H)	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
143	34H	I!	LMI(AC)	RRM;									
(0034H)	0011101	11	00	0000100	00000000	00	110	1	0	00010	00000000	0	0
144	44H			ACM(AC);									
(0044H)	0001011	11	00	0000101	00000000	00	000	1	0	00010	00000000	0	0
145	54H			LMI(AC)	RRM	JPR(LA STA LX STX A1 S1	4 D	ADM	OR1	XOR1			
146	(0054H)	0011101	11	00	1100111	00000000	00	110	1	0	00010	00000000	0
						ANDD C SZ SNZ GG);							
147	64H	SGE!	NOP(A)	JCC(SGEC);									
(0064H)	1100000	11	00	0001110	00000000	00	000	1	0	00010	00000000	0	0
148	74H	A!		ACM(AC);									
(0074H)	0001011	11	00	0001000	00000000	00	000	1	0	00010	00000000	0	0

149	84H (0984H)	0000000	ALP(A) STC JZR(FETCH)!	00	000	1	0	00010	00000000	0	0
150	0E4H (00E4H)	1100000	SGEC! NOP(A) JCF(INTEUS.F05)!	00	000	1	0	00010	00000000	0	0
151	114H (0114H)	1010000	APS! TZP(A) STC K90000 INH!	00	001	1	0	00010	00000000	0	0
152	124H (0124H)	0011001	DSM(R9)!	00	000	1	0	00010	00000000	0	0
153	134H (0134H)	0001111	XX! SRA(AC) STZ FFC JFL(OUT2.AGN2)!	00	000	1	0	00010	00000000	0	0
154	1D4H (01D4H)	0011001	DWN! LMT(R9) FF1 JCR(PUSA4)!	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 9

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRR	LCNT	EMIT	MULTF	SPRF
155	1E4H (01F4H)	6543210	10	10	6543210	10	010	0	0	43210	76543210	0	0
		N5:	NOP(A)	JZR(FETCH):			00000000	1	0	00010	00000000	0	0
		1100000	11	00	0101111	00	00000000						
156	35H (0035H)	J1:	LMI(AC)	RRM:			00000000	1	0	00010	00000000	0	0
		0011101	11	00	0000100	00	00000000						
157	45H (0045H)	ACM(AC):					00000000	1	0	00010	00000000	0	0
		0001011	11	00	0000101	00	00000000						
158	55H (0055H)	CL1:	SDR(P)	FF1	JZR(FETCH):		00000000	1	0	00010	00000000	0	0
		0100011	11	11	0101111	00	00000000						
		0100011	11	11	0101111	11111111							
159	65H (0065H)	SLE1:	NOP(A)	JCC(SLFC):			00000000	1	0	00010	00000000	0	0
		1100000	11	00	0010101	00	00000000						
160	75H (0075H)	S1:	ACM(AC):				00000000	1	0	00010	00000000	0	0
		0001011	11	00	0001000	00	00000000						
161	85H (0085H)	CIA(AC)	FF1:				00000000	1	0	00010	00000000	0	0
		0011111	11	11	0001001	00	00000000						
162	95H (0095H)	ALR(A)	STC	JZR(FETCH):			00000000	1	0	00010	00000000	0	0
		0000000	10	00	0101111	00	00000000						
		0000000	10	00	0101111	11111111							

I63	115H (0115H)	ALS! 0011001	DSM(R9)! 11 00 0010010	11111111	00	000	1	0	00010	00000000	0	0	
I64	125H (0125H)	ILR(A)!	0000000	11 00 0010011	00000000	00	000	1	0	00010	00000000	0	0
I65	135H (0135H)	DSM(R9)!	0011001	11 00 0010100	11111111	00	000	1	0	00010	00000000	0	0
I66	145H (0145H)	RACK! 0000000	ALP(A) STC 10 00 1000100	JFL(OUT3,AGN3)!	11111111	00	000	1	0	00010	00000000	0	0
I67	155H (0155H)	SLECI 1100000	NOP(A) JCF 11 00 1010110	(NTEQ6,EQ6)!	00000000	00	000	1	0	00010	00000000	0	0
I68	105H (0105H)	OWN2! 0001000	ILP(R9) RWM 11 00 0110100	JCR(DWN1)!	00000000	00	111	1	0	00010	00000000	0	0

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSEFF	STRB	LPCNT	EMIT	MULTF	SPRF
169	1E5H! (01F5H)	T-AP!	NOP(A)	JZR(FETCH)!	00000000	00	000	1	0	00010	00000000	0	0
170	36H! (0036H)	LI!	LMI(S)	FFI!	0010100	11 11	0000100	00000000	00	00010	00000000	0	0
171	46H! (0046H)		ILR(P)	RVM!	0000011	11 00	0000101	00000000	00	00010	00000000	0	0
172	56H! (0056H)		LMI(R9)	RRM	JCR(CLI)!	0011001	11 00	0110111	00000000	00	00010	00000000	0
173	66H! (0066H)	PUSA!	LMI(R9)	FFI	JCC(PUSAC)!	0011001	11 11	0001011	00000000	00	00010	00000000	0
174	76H!	/*	LOAD MULTIPLIER	---	MLT---	MT	*/						
175		M!	ACM(T)!										
176	86H!	/*	SAVE STGN BIT	IN C FLAG	*/								
177			ILR(T)!										

```

178 8F4!          SDR(R9) FF1:
(009FH) 0101001 11 11 0001001 1111111 00 000 1 0 00010 00000000 0 0

179 9F4!          TZR(A) KR0000 INH STC:
(009FH) 1010000 10 00 0110110 1000000 00 001 1 0 00010 00000000 0 0

180 /*          WIPE OUT SIGN BIT */
181 9A4!          TZR(T) K7FFFF:
(0096H) 1011100 11 00 0001010 0111111 00 000 1 0 00010 00000000 0 0

182 /*          SET " LOOP COUNTER */
183 0A4H!          CLR(AC):
(00A6H) 1001101 11 00 0110101 0000000 00 000 1 0 00010 00000000 0 0

184 0A5H!          LMT(AC) K0000F:
(00A5H) 0011101 11 00 0110100 0000111 00 000 1 0 00010 00000000 0 0

185 0A4H!          LMT(AC) K00004:
(00A4H) 0011101 11 00 0001011 00000100 00 000 1 0 00010 00000000 0 0

```

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSEF	STRR	LPCNT	EMIT	MULTF	SQRF
186	084H (00R4H)	0011111	11	11	0110101	00000000	00	000	1	0	00010	00000000	0 0
				CIA(AC) FF1;							76543210	0	0
187	0A5H (00R5H)	0101000	11	11	0001100	11111111	00	000	1	0	00010	00000000	0 0
				SDR(RA) FF1;									
188	/*												
189	0C5H (00C5H)	1001101	11	00	0110100	00000000	00	000	1	0	00010	00000000	0 0
				CLEAR PARTIAL PRODUCT CLR(AC);									
190	/*												
191	0C4H (00C4H)	0001110	11	00	0110010	00000000	00	000	1	0	00010	00000000	0 0
				FETCH AND TEST MULTIPLIER SRA(T);									
192	0C2H (00C2H)	MUPI	LMI(RA)	FF1	STZ	JFL(MBZ,MB1);	00	000	1	0	00010	00000000	0 0
		0011000	01	11	1001011	00000000	00						
193	/*												
194	0B3H (00R3H)	0100111	11	11	0111111	11111111	00	000	1	0	00010	00000000	0 0
				ADD SEQUENCE MBI;									
				SDR(R7) FF1	JCR(OV1);								
195	0AFH (00R6H)	0000000	11	00	0111110	00000000	00	000	1	0	00010	00000000	0 0
				ILR(A) FF0;									

194	09E4I (00RF4)	0000111	11 00	0110010	11111111	00	000	1	0	00010	00000000	0	0
					ALR(R7) FFD JCR(MZ7):								
197	/*	SHIFT RIGHT FILL WITH ONES */											
198	0924I (00R24)	0001111	01 01	0110001	00000000	00	000	1	0	00010	00000000	0	0
					SRA(AC) FFC STZ:								
					MZ7:								
199	0914I (00R14)	0001110	11 10	1011100	00000000	00	000	1	0	00010	00000000	0	0
					SRA(T) FFZ JZF(MLP.MEX):								
200	/*	APPLY CORRECTION */											
201	0C34I (00C34)	1011001	11 00	0111001	10000000	00	001	1	0	00010	00000000	0	0
					MFX: TZR(R9) KR0900 INH JCR(XXX):								
202	0C94I (00C94)	1100000	11 00	1001100	00000000	00	000	1	0	00010	00000000	0	0
					XXX: NOP(A) JFL(POS.NEG):								
203	0C94I				NEG: CMR(A) JCR(OV2):								

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
	6543210	10	10	6543210	76543210	10	0	0	0	43210	76543210	0	0
	(00CRH)	11	00	0111101	00000000	00	000	1	0	00010	00000000	0	0
204	0CDH	OV2I	ALR(A)	FF1									
	(00CDH)	0000000	11	11	0111010	11111111	00	000	1	0	00010	00000000	0
205	0CAH	POS	SDR(A)	FF1	JCR(OV3)								
	(00CAH)	0100000	11	11	0111110	11111111	00	000	1	0	00010	00000000	0
206	0CEH	OV3I	ILR(T)										
	(00CEH)	0001100	11	00	0110001	00000000	00	000	1	0	00010	00000000	0
207	0CIH	SRA(AC)	FF0										
	(00CIH)	0001111	11	00	0111111	00000000	00	000	1	0	00010	00000000	0
208	0CFH	SOR(O)	FF1	JZR(FETCH)									
	(00CFH)	0100010	11	11	0101111	11111111	00	000	1	0	00010	00000000	0
209	0R6H	PUSC	ILR(R0)	RWM									
	(00R6H)	0000000	11	00	0001100	00000000	00	111	1	0	00010	00000000	0
210	0C6H	LMI(R9)	FF1										
	(00C6H)	0011001	11	11	0001101	00000000	00	000	1	0	00010	00000000	0

211	00641 (00064H)	0000001	ILR(R1) RWM:	11 00 0001110	00000000	00	111	1	0	00010	00000000	0	0
212	0E641 (00E64H)	0011001	LMT(R9) FFI:	11 11 0001111	00000000	00	000	1	0	00010	00000000	0	0
213	0F641 (00F64H)	0000010	ILR(R2) RWM:	11 00 0010000	00000000	00	111	1	0	00010	00000000	0	0
214	10641 (01064H)	0011001	LMT(R9) FFI JCC(FUSAP):	11 11 0010011	00000000	00	000	1	0	00010	00000000	0	0
215	11641 (01164H)	LPS: 0000000	ILR(A):	11 00 0010010	00000000	00	000	1	0	00010	00000000	0	0
216	12641 (01264H)	0011001	DSM(R9) JCC(ZZ):	11 00 0010101	11111111	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 13

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CSUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
217	136H (0136H)	0000011	11	00	0010100	00	111	1	0	00010	00000000	0	0
		PUSA2:	ILR(R3)	RWM:									
218	146H (0146H)	0011001	11	11	0010110	00	000	1	0	00010	00000000	0	0
		LMI(R9)	FFI	JCC(PUSA3):									
219	156H (0156H)	0001111	01	00	1000101	00	000	1	0	00010	00000000	0	0
		Z7:	SRA(AC)	STZ	JFL(OUT.AGN):								
220	152H (0152H)	0001101	11	10	0110001	00	000	1	0	00010	00000000	0	0
		OJT:	ALR(AC)	FFZ	JCR(WM):								
221	153H (0153H)	0011001	11	00	0110110	00	000	1	0	00010	00000000	0	0
		AGN:	DSM(R9)	JCR(Z7):									
222	166H (0166H)	0000100	11	00	0010111	00	111	1	0	00010	00000000	0	0
		PUSA3:	ILR(R4)	RWM:									
223	176H (0176H)	0011001	11	11	0011000	00	000	1	0	00010	00000000	0	0
		LMI(R9)	FFI:										
224	186H (0186H)	0000101	11	00	0011001	00	111	1	0	00010	00000000	0	0
		ILR(R5)	RWM:										

225	194HI (C1964)	0011001	LMI(R9) FFI: 11 11 0011010	00	000	1	0	00010	00000000	0	0
226	1A6HI (C1A64)	0000110	ILR(R6) RWM: 11 00 0011011	00	111	1	0	00010	00000000	0	0
227	1R6HI (C1R64)	0011001	LMI(R9) FFI: 11 11 0011100	00	000	1	0	00010	00000000	0	0
228	1C6HI (C1C64)	0000111	ILR(R7) RWM: 11 00 0000010	00	111	1	0	00010	00000000	0	0
229	024HI (A0244)	0011001	LMI(R9) FFI: 11 11 0110111	00	000	1	0	00010	00000000	0	0
230	274I (00274)	0001000	ILR(R8) RWM: 11 00 0111000	00	111	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 14

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	PAUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
231	29H (0029H)	6543210	10	10	6543210	76543210	10	0	0	43210	76543210	0	0
		LMI(R9)	FFI				000	1	0	00010	00000000	0	0
232	29H (0029H)	ILR(T)	RWM	JZR(FETCH)			000	1	0	00010	00000000	0	0
		0001100	11	00	0101111	00000000	00	1	0	00010	00000000	0	0
233	1E6H (01E6H)	N7	NOP(A)	JZR(FETCH)			000	1	0	00010	00000000	0	0
		1100000	11	00	0101111	00000000	00	1	0	00010	00000000	0	0
234	1D6H (01D6H)	LMI(R9)	FFI	JCR(DW2)			000	1	0	00010	00000000	0	0
		0011001	11	11	0110101	00000000	00	1	0	00010	00000000	0	0
235	37H (0037H)	M5C	NOP(A)	JPR(SF SG SL SNE SGE SLE PUSX POPX)			000	1	0	00010	00000000	0	0
		1100000	11	00	1100110	00000000	00	1	0	00010	00000000	0	0
237	57H (0057H)	CLI	LTM(AC)	JCR(CL)			000	1	0	00010	00000000	0	0
		1011011	11	00	0110101	11111111	00	1	0	00010	00000000	0	0
238	67H (0067H)	PUSX	LMI(S)	FFI	JCC(PUSX)		000	1	0	00010	00000000	0	0
		0010100	11	11	0001000	00000000	00	1	0	00010	00000000	0	0
239	/*	DIVIDE											
240	77H (0077H)	DI	CLR(R6)				000	1	0	00010	00000000	0	0
		1000110	11	00	0001111	00000000	00	1	0	00010	00000000	0	0

241	0F7H: (10F7H)	1011101	11 00	0010000	00000000	00	000	1	0	00010	00000000	0	0
				CLR(AC):									
242	117H: (1107H)	0011101	11 00	0010010	00001111	00	000	1	0	00010	00000000	0	0
				LMI(AC) K0000F:									
243	127H: (1127H)	0011101	11 00	0010011	00000101	00	000	1	0	00010	00000000	0	0
				LMI(AC) K00005:									
244	137H: (1137H)	0011111	11 11	0010100	00000000	00	000	1	0	00010	00000000	0	0
				CIA(AC) FF1:									
245	147H: (1147H)	0101000	11 11	0010101	11111111	00	000	1	0	00010	00000000	0	0
				SDR(R9) FF1:									

XMAS VERS 2.0 PAGED EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STRR	LPCNT	EMIT	MULTF	SPRF
246	157H (0157H)	0011001	11 00	0010110	00000000	00	110	1	0	00010	00000000	0	0
				LMI(R9) RRM:									
247	167H (0167H)	0001011	11 00	0010111	00000000	00	000	1	0	00010	00000000	0	0
				ACM(AC):									
248	177H (0177H)	0101001	11 11	0011000	11111111	00	000	1	0	00010	00000000	0	0
				SDR(R9) FFI:									
249	187H (0187H)	0101100	11 11	0110101	11111111	00	000	1	0	00010	00000000	0	0
				SDR(T) FFI:									
250	1A5H (01A5H)	0000000	11 00	0110100	00000000	00	000	1	0	00010	00000000	0	0
				ILP(A):									
251	1A4H (01A4H)	0100010	11 11	0110010	11111111	00	000	1	0	00010	00000000	0	0
				SDR(R2) FFI:									
252	/*	START MAIN LOOP, COMPARE SIGN Y WITH SIGN RI											
253	1A2H (01A2H)	1111001	11 00	0110001	11111111	00	000	1	0	00010	00000000	0	0
				DIV: XNR(R9):									
254	1A1H (01A1H)	1011001	11 00	0110000	10000000	00	001	1	0	00010	00000000	0	0
				TZR(R9) FFO KAR0000 INH:									

255	190H! (0190H)	0000110	11 00 1001001	00000000	00	000	1	0	00010	00000000	0	0
				ILR(R6) JFL(S7ER0.S0ME):								
256	/* DIFFERENT SIGNS. */											
257	192H! (0192H)	0000110	11 00 0110001	11111111	00	000	1	0	00010	00000000	0	0
				S7ER0! ALR(R6) FF0:								
258	191H! (0191H)	0000010	11 00 0110000	00000000	00	000	1	0	00010	00000000	0	0
				ILP(R2):								
259	190H! (0190H)	0000010	11 00 0011010	11111111	00	000	1	0	00010	00000000	0	0
				ALR(R2) FF0:								
260	1A0H! (01A0H)	0001100	11 00 0011011	00000000	00	000	1	0	00010	00000000	0	0
				ILP(T):								

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CAUS	PAUSFF	STR8	LPCNT	EMIT	MULTF	SPRF
261	180H (0190H)	0101001	11 11	0011100	11111111	00	000	1	0	00010	00000000	0	0
				SDR(R9) FF1:									
262	1C0H (01C0H)	0000010	11 00	0110001	11111111	00	000	1	0	00010	00000000	0	0
				ALR(R2) JCP(OVER):									
263	/*	SAME	STGNS	*/									
264	193H (0193H)	0000110	11 11	0110100	11111111	00	000	1	0	00010	00000000	0	0
				ALR(R6) FF1:									
265	194H (0194H)	0000010	11 00	0110101	00000000	00	000	1	0	00010	00000000	0	0
				ILR(R2):									
266	195H (0195H)	0000010	11 00	0110111	11111111	00	000	1	0	00010	00000000	0	0
				ALR(R2):									
267	197H (0197H)	0001100	11 00	0011010	00000000	00	000	1	0	00010	00000000	0	0
				ILR(T):									
268	1A7H (01A7H)	0101001	11 00	0011011	11111111	00	000	1	0	00010	00000000	0	0
				SDR(R9):									
269	1B7H (01B7H)	1111101	11 00	0110001	00000000	00	000	1	0	00010	00000000	0	0
				CMR(AC):									

270	1R1H:	0000010	11 11 0011100	11111111	00	000	1	0	00010	00000000	0	0
	(C1R1H)											
	/* CHECK FOR I=0 */											
271	1C1H:	0011000	11 11 0110111	00000000	00	000	1	0	00010	00000000	0	0
	(C1C1H)											
	/* APPLY CORRECTION */											
272	1C1H:	0011000	11 11 0110111	00000000	00	000	1	0	00010	00000000	0	0
	(C1C1H)											
273	1C7H:	1100000	11 00 1001000	00000000	00	000	1	0	00010	00000000	0	0
	(C1C7H)											
	/* APPLY CORRECTION */											
274	1R3H:	1001101	11 00 0111000	00000000	00	000	1	0	00010	00000000	0	0
	(C1R3H)											
274	1R9H:	0011101	11 11 0011001	10000000	00	000	1	0	00010	00000000	0	0
	(C1R9H)											

XMAS VERS 2.0 PACEP EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRR	LPCNT	EMIT	MULTF	SPRF	
277	199H!	0000110	11	00	0011010	11111111	00	000	1	0	00010	00000000	0	0
	(0199H)	ALR(R6):												
278	1A9H!	0100000	11	11	0101111	11111111	00	000	1	0	00010	00000000	0	0
	(01A9H)	SDR(A) FF1 JZR(FETCH):												
279	87H!	PUSX11	ILR(A)	RMM:										
	(0087H)	0000000	11	00	0001001	00000000	00	111	1	0	00010	00000000	0	0
280	97H!	LMI(S)	FF1:											
	(0097H)	0010100	11	11	0001010	00000000	00	000	1	0	00010	00000000	0	0
281	0A7H!	ILR(X)	RMM:											
	(00A7H)	0000001	11	00	0001011	00000000	00	111	1	0	00010	00000000	0	0
282	0B7H!	LMI(S)	FF1:											
	(00B7H)	0010100	11	11	0001100	00000000	00	000	1	0	00010	00000000	0	0
283	0C7H!	ILR(P)	RMM:											
	(00C7H)	0000011	11	00	0001101	00000000	00	111	1	0	00010	00000000	0	0
284	0D7H!	LMI(S)	FF1:											
	(00D7H)	0010100	11	11	0001110	00000000	00	000	1	0	00010	00000000	0	0

295	0F7H: (00F7H)	0000111	ILR(W) RWM JZR(FETCH): 11 00 0101111 00000000	00	111	1	0	00010	00000000	0	0
296	117H: (0117H)	1010000	SSP: TZR(A) K7FFFF JZR(FETCH): 11 00 0101111 01111111	00	100	1	0	00010	00000000	0	0
297	1D7H: (01D7H)	0001100	PUSAK: ILR(T) RWM JZR(FETCH): 11 00 0101111 00000000	00	111	1	0	00010	00000000	0	0
298	1F7H: (01F7H)	1100000	RVI: NOP(A) JZR(FETCH): 11 00 0101111 00000000	00	100	1	0	00010	00000000	0	0
299	3AH: (003AH)	0000001	YK: ILR(X): 11 00 0000001 00000000	00	100	1	0	00010	00000000	0	0
290	019H: (0019H)	0001001	ALP(P9): 11 00 0000100 11111111	00	100	1	0	00010	00000000	0	0

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CRUS	PAUSFF	STPR	LPCNT	EMIT	MULTF	SPRF
291	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
292	(004AH)	0011101	11	00	1100111	00000000	110	1	0	00010	00000000	0	0
		LMI(AC) RRM JPR(LA STA LX STX AI SI M D ADM ORI YORI AVDD C SZ SNZ GG);											
293	(0069H)	0010100	11	00	0001000	11111111	000	1	0	00010	00000000	0	0
		POPXI DSM(S) JCC(POPXI);											
294	(0078H)	0010111	11	11	0101111	00000000	111	1	0	00010	00000000	0	0
		ADM! ACM(AC) FFI RWM J7R(FETCH);											
295	(008AH)	0010100	11	00	0001001	00000000	110	1	0	00010	00000000	0	0
		POPXI LMI(S) RRM;											
296	(009AH)	0010111	11	00	0001010	00000000	000	1	0	00010	00000000	0	0
		ACM(AC);											
297	(00A8H)	0110111	11	11	0001011	11111111	000	1	0	00010	00000000	0	0
		SDR(W) FFI;											
298	(00BAH)	0010100	11	00	0001100	11111111	000	1	0	00010	00000000	0	0
		DSM(S);											
299	(00CAH)	0010100	11	00	0001101	00000000	110	1	0	00010	00000000	0	0
		LMI(S) RRM;											

300	009H! (0009H)	0001011	ACM(AC): 11 00 0001110	00000000	00	000	1	0	00010	00000000	0	0
301	0F9H! (00F9H)	0100011	SOR(P)FF1: 11 11 0001111	11111111	00	000	1	0	00010	00000000	0	0
302	0F9H! (00F9H)	0010100	DSM(S): 11 00 0010000	11111111	00	000	1	0	00010	00000000	0	0
303	109H! (0109H)	0010100	LMJ(S) RRM JCC(POPX2): 11 00 0010011	00000000	00	110	1	0	00010	00000000	0	0
304	119H! (0119H)	0101101	SSN! CSP(AC): 11 00 0010010	00000000	00	000	1	0	00010	00000000	0	0
305	129H!		ORR(A) K80000	JZR(FETCH):								

XMAS VERS 2.0 PACED EMULATOR MICROCODE FOR INTFL 3000

ERRORS= 0 PAGE 19

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPDF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(012AH)	1100000	11	00	0101111	10000000	00	000	1	0	00010	00000000	0	0
306	POPX?!	SDR(X)	FF1!										
(013AH)	0100001	11	11	0010100	11111111	00	000	1	0	00010	00000000	0	0
307	14AH!				DSM(S)!								
(014AH)	0010100	11	00	0010101	11111111	00	000	1	0	00010	00000000	0	0
308	15AH!				LMT(S) PRM!								
(015AH)	0010100	11	00	0010110	00000000	00	110	1	0	00010	00000000	0	0
309	16AH!				SDP(A) FF1 JZR(FETCH)!								
(016AH)	0100000	11	11	0101111	11111111	00	000	1	0	00010	00000000	0	0
310	1DAH!				PUSAS! LMT(R9) FF1 JCR(PUSAS)!								
(01DAH)	0011001	11	11	0110111	00000000	00	000	1	0	00010	00000000	0	0
311	1EAH!				SMT! NOP(A) JZR(FETCH)!								
(01EAH)	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
312	19AH!				JJI NOP(A) JZR(FETCH)!								
(0039AH)	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0

```

313 434I RTN2! ACM(AC) JCR(RTN3)!
(00434) 001011 11 00 0111010 00000000 00 000 1 0 00010 00000000 0 0 0

314 594I RTN1! LMI(S) RRM JCC(RTN2)!
(00594) 0010100 11 00 0000100 00000000 00 000 1 0 00010 00000000 0 0 0

315 694I RTN1! DSM(S) JCC(RTN1)!
(00694) 0010100 11 00 0000101 11111111 00 000 1 0 00010 00000000 0 0 0

316 794I 001! ACM(AC)!
(00794) 001011 11 00 0001000 00000000 00 000 1 0 00010 00000000 0 0 0

317 894I 000(A) JZR(FETCH)!
(00894) 1100000 11 00 0101111 11111111 00 000 1 0 00010 00000000 0 0 0

318 1194I EX! SDR(T) FFI!
(01194) 0101100 11 11 0010010 11111111 00 000 1 0 00010 00000000 0 0 0

```

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
319	129H! (0129H)	0000001	11	00	0010011	00	000	1	0	00010	00000000	0	0
		ILR(X):											
320	139H! (0139H)	0100000	11	11	0010100	00	000	1	0	00010	00000000	0	0
		SDR(A) FFI:											
321	149H! (0149H)	0001100	11	00	0010101	00	000	1	0	00010	00000000	0	0
		ILR(T):											
322	159H! (0159H)	0100001	11	11	0101111	00	000	1	0	00010	00000000	0	0
		SDR(X) FFI JTR(FETCH):											
323	169H! (0169H)	0101100	11	11	0011001	00	000	1	0	00010	00000000	0	0
		DMI: SDR(T) FFI:											
324	199H! (0199H)	0000010	11	00	0011010	00	000	1	0	00010	00000000	0	0
		ILR(O):											
325	1A9H! (01A9H)	0100000	11	11	0011011	00	000	1	0	00010	00000000	0	0
		SDR(A) FFI:											
326	1B9H! (01B9H)	0001100	11	00	0011100	00	000	1	0	00010	00000000	0	0
		ILR(T):											

327	1C3H1 (M1C9H)	0100010	11	11	0101111	1111111	00	000	1	0	00010	00000000	0	0	
					SDR(O) FFI IZR(FETCH):										
328	109H1 (M109H)	P15A41	ILR(R9)	RW4	JCR(PUSAS):			711	1	0	00010	00000000	0	0	
					0011001 11 00 0111000 00000000 00										
329	1F9H1 (M1F9H)	E0:	ILR(A)	JCC(OV1):			00	000	1	0	00010	00000000	0	0	
					0000000 11 00 0111000 00000000 00										
330	1A41 (M1A4H)	POS1:	ILR(P)	FF1	IZR(FETCH):			00	000	1	0	00010	00000000	0	0
					0000011 11 11 0101111 00000000 00										
331	24H1 (M24H)	POS5:	NOP(A)	JZR(FETCH):			00	000	1	0	00010	00000000	0	0	
					1100000 11 00 0101111 00000000 00										

332 /* ENTRY FOR PAGE 2 CENTROID TRACKER */
333 3A41 CTBACKI NOP(A) JZR(FETCH) PAGE3:

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 21

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	FMIT	MULTF	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(007AH)	110000	11	00	0101111	00000000	10	000	1	0	00010	00000000	0	0
334	4AH	RIN3!	SDR(P)	FF1	JZR(FETCH):								
(004AH)	0100011	11	11	0101111	11111111	00	000	1	0	00010	00000000	0	0
335	5AH	SKN1!	NOP(A)	JFL(POSS-NEGG):									
(005AH)	1100000	11	00	1000010	00000000	00	000	1	0	00010	00000000	0	0
336	6AH	SKN!	TZR(A)	KR0000	INH	JCC(SKNI):							
(006AH)	1010000	11	00	0000101	10000000	00	001	1	0	00010	00000000	0	0
337	7AH	XOR1!	ACM(AC):										
(007AH)	0001011	11	00	0001000	00000000	00	000	1	0	00010	00000000	0	0
338	8AH	XNR(A)	JCR(OWN5):										
(008AH)	1110000	11	00	0111101	11111111	00	000	1	0	00010	00000000	0	0
339	/*	IF A	.LT.	MEM	CF=0	ZF=0:	IF A	.GT.	MFM	CF=0	ZF=1		
340	IF A	.EQ.	MFM	CF=1	ZF=0	*/							
341	0AAH!	NTSAME!											
(00AAH)	1100000	11	00	1001111	00000000	00	000	1	0	00010	00000000	0	0
342	0BAH!	NOCRY!	NOP(A)	JZR(FETCH):									
(00BAH)	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0

343	0DAH (00DAH)	NQCRY21 1100000	10 11 0101111	NOP(A) FF1 STC JZP(FETCH): 00 000	1 0 00010	00000000	0 0
344	0FAH (00FAH)	APOS21 NOP(A) 1100000	01 11 0101111	FF1 STZ JZR(FETCH): 00 000	1 0 00010	00000000	0 0
345	10AH (010AH)	SP1 ILR(P) FF1 JZR(FETCH): 0000011	11 11 0101111	00 000	1 0 00010	00000000	0 0
346	11AH (011AH)	EP1 SDR(T) FF1: 0101100	11 11 0010010	1111111 00 000	1 0 00010	00000000	0 0
347	12AH (012AH)	ILR(P): 0000011	11 00 0010011	00000000 00 000	1 0 00010	00000000	0 0
348	13AH	SDR(A) FF1:					

XMAS VERS 2.0 PAGED EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
	(0174H)	010000	11	11	0010100	11111111	00	000	1	0	00010	00000000	0
343	14AH	ILR(T)!											
	(0144H)	0001100	11	00	0010101	00000000	00	000	1	0	00010	00000000	0
350	15AH	SDR(P) FFI JZR(FETCH)!											
	(0154H)	0100011	11	11	0101111	11111111	00	000	1	0	00010	00000000	0
351	16AH	MSKTP! NOP(A) JZR(FETCH)!											
	(0164H)	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0
352	17AH	EVFN! ILR(P) FFI JZR(FETCH)!											
	(0174H)	0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0
353	18AH	N-! NOP(A) JFL(NOVEFI+OVEFI)!											
	(0184H)	1100000	11	00	1001001	00000000	00	000	1	0	00010	00000000	0
354	19AH	NOVEFI!											
	(0194H)	1100000	11	00	0101111	NOP(A) JZR(FETCH)!	00	000	1	0	00010	00000000	0
355	1AAH	OVFF2! ILR(P) FFI JZR(FETCH)!											
	(01AAH)	0000011	11	11	0101111	00000000	00	000	1	0	00010	00000000	0

356	IRAH1 (01RAH)	NC21 110000	NOP(A) 11 00 1001100	JFL(NOVEF3,OVEF3): 00000000	00	000	1	0	00010	00000000	0	0
357	ICAH1 (01CAH)	NVEF31 0000011	11 11 0101111	ILR(P) FFI 00000000	00	000	1	0	00010	00000000	0	0
358	IDAH1 (01DAH)	0 EF41 0000011	ILR(P) FFI 11 11 0101111	JZR(FETCH): 00000000	00	000	1	0	00010	00000000	0	0
359	IEAH1 (01EAH)	N111 1100000	NOP(A) 11 00 0101111	JZR(FETCH): 00000000	00	000	1	0	00010	00000000	0	0
360	I9H1 (00I9H)	NFG11 1100000	NOP(A) 11 00 0101111	JZR(FETCH): 00000000	00	000	1	0	00010	00000000	0	0
361	2RH1 (002RH)	NFGG1 0000011	ILR(P) FFI 11 11 0101111	JZR(FETCH): 00000000	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACED EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 23

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	PAUS	PAUSFF	STRR	LPCNT	EMIT	MULTF	SPRF
362	6543210	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
(003RH)	LI!	NOP(A)	JZR(FETCH):										
363	1100000	11	00	1000001	00000000	00	000	1	0	00010	00000000	0	0
(005RH)	SKI!	NOP(A)	JFL(POS1,NEG1):										
364	1010000	11	00	0000101	10000000	00	001	1	0	00010	00000000	0	0
(006RH)	SKP!	TZR(A)	K8000 INH JCC(SKPI):										
365	0001011	11	00	0001000	00000000	00	000	1	0	00010	00000000	0	0
(007RH)	AND!	ACM(AC):											
366	1000000	11	00	0101111	11111111	00	000	1	0	00010	00000000	0	0
(008RH)	ANR(A)	JZR(FETCH):											
367	1111000	11	00	0111101	00000000	00	000	1	0	00010	00000000	0	0
(009RH)	SAME!	CMR(R8)	JCR(DWNT):										
368	1100000	11	00	1001101	00000000	00	000	1	0	00010	00000000	0	0
(008RH)	CRY!	NOP(A)	JFL(NOCRY2,CRY2):										
369	1100000	01	11	0101111	00000000	00	000	1	0	00010	00000000	0	0
(008RH)	COY2!	NOP(A)	FFI STZ JZR(FETCH):										

370	0F9H: (00F9H)	ANEG21 1100000	NOP(A) 11 00 0101111	JZR(FF1CH): 00000000	00	000	1	0	00010	00000000	0	0
371	109H: (0109H)	NOP1 1100000	NOP(A) 11 00 0101111	JZR(FETCH): 00000000	00	000	1	0	00010	00000000	0	0
372	119H: (0119H)	FS: 0101100	SDR(T) FFI: 11 11 0010010	11111111	00	000	1	0	00010	00000000	0	0
373	129H: (0129H)	ILR(W): 0000111	11 00 0010011	00000000	00	000	1	0	00010	00000000	0	0
374	139H: (0139H)	SDR(A) FFI: 0100000	11 11 0010100	11111111	00	000	1	0	00010	00000000	0	0
375	149H: (0149H)	ILR(T): 0001100	11 00 0010101	00000000	00	000	1	0	00010	00000000	0	0

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SDPF
376	1584! (015RH)	010011	11	11	010111	11111111	00	000	1	0	00010	00000000	0
					SDR(W) FF1 JZR(FETCH):								
377	168H! (016RH)	000011	11	11	010111	00000000	00	000	1	0	00010	00000000	0
					SKIP! ILP(P) FF1 JZR(FETCH):								
378	178H! (017RH)	110000	11	00	010111	00000000	00	000	1	0	00010	00000000	0
					ODD! NOP(A) JZR(FETCH):								
379	188H! (018RH)	110000	11	00	1001010	00000000	00	000	1	0	00010	00000000	0
					CR1 NOP(A) JFL(OVFF2,NOVFF2):								
380	198H! (019RH)	000011	11	11	010111	00000000	00	000	1	0	00010	00000000	0
					OVFF1! ILP(P) FF1 JZR(FETCH):								
381	1A8H! (01A8H)	110000	11	00	010111	00000000	00	000	1	0	00010	00000000	0
					NOVFF2! NOP(A) JZR(FETCH):								
382	1A8H! (01A8H)	110000	11	00	1001101	00000000	00	000	1	0	00010	00000000	0
					CR2! NOP(A) JFL(OVFF4,NOVFF4):								
383	1CBH! (01CBH)	110000	11	00	010111	00000000	00	000	1	0	00010	00000000	0
					OVFF3! NOP(A) JZR(FETCH):								

384	1DRH1 (01DRH)	NOVFF4: 110000	11	00	0101111	NOP(A) JZR(FETCH): 0000000	00	0000	1	0	00010	00000000	0	0
385	1FRH1 (01FRH)	NIP: 110000	11	00	0101111	NOP(A) JZR(FETCH): 0000000	00	0000	1	0	00010	00000000	0	0
386	3CH1 (003CH)	Iv: 0011001	11	00	0000100	LMI(R9) RRM: 0000000	00	110	1	0	00010	00000000	0	0
387	4CH1 (004CH)	ACM(AC): 0001011	11	00	0000101	0000000	00	000	1	0	00010	00000000	0	0
388	5CH1 (005CH)	ALP(X) JCR(CIX): 0000001	11	00	0111101	11111111	00	000	1	0	00010	00000000	0	0
389	6CH1 (006CH)	SN1 NOP(A) JCC(SO1): 110000	11	00	0011000	0000000	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PAGED EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 25

RECORD NUMBER	CPE	FT	FO	JUMP	KRUSS	PAGEF	PAUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SRPF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
390	/* REGISTER EIGHT IS SET TO STACK POINTER WHEN SIGNS OF MEMORY LOCAT AND ACCUMULATOR ARE DIFFERENT												
391													
392													
393	REGISTER EIGHT IS SET TO THE DIFFERENCE OF THE MEMORY LOCATION AND THE ACCUMULATOR WHEN THE SIGNS OF MEMORY LOCATION AND ACCUMULATOR ARE DIFF												
394													
395	ERENT												
396	7CH: C: ACM(AC):												
	(007CH)	0001011	11	00	0001000	00000000	00	000	1	0	00010	00000000	0 0
397	8CH: SDR(R8) FFI:												
	(008CH)	0101000	11	11	0001001	11111111	00	000	1	0	00010	00000000	0 0
398	9CH: SDR(T) FFI K80000:												
	(009CH)	0101100	11	11	0001010	10000000	00	000	1	0	00010	00000000	0 0
399	0ACH: ILP(A) STC:												
	(00ACH)	0000000	10	00	0001011	00000000	00	000	1	0	00010	00000000	0 0
400	0BCH: SDR(R5) FFI K80000:												
	(00BCH)	0100101	11	11	0001100	10000000	00	000	1	0	00010	00000000	0 0
401	0CCH: ILP(R5) FFO STZ:												
	(00CCH)	0000101	01	00	0001101	00000000	00	000	1	0	00010	00000000	0 0

402	00CH1 (100CH)	1111100	11 00	XNP(T) 0010000	JCC(CX2): 11111111	00	000	1	0	00010	00000000	0	0
403	0FCH1 (100FCH)	CX3: 1011100	11 00	T7P(T) 0001111	INH: 11111111	00	001	1	0	00010	00000000	0	0
404	0FCH1 (100FCH)	1010101	11 00	T7P(R5) 1001010	INH JFL(VTSAME.SAVE): 11111111	00	001	1	0	00010	00000000	0	0
405	10CH1 (110CH)	CX2: 0000000	11 00	ILR(A) 0001110	FF0 JCC(CX3): 00000000	00	000	1	0	00010	00000000	0	0
406	11CH1 (111CH)	ICX1: 0001001	11 00	ILP(R9): 0010010	00000000	00	000	1	0	00010	00000000	0	0
407	12CH1 (112CH)	0000001	11 00	ALP(X): 0010011	11111111	00	000	1	0	00010	00000000	0	0

XMAS VERS 2.0 PACER EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 24

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SPRF
408	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(017CH)	1011111	11	00	0010100	10000000	00	001	1	0	00010	00000000	0	0
	TZA(AC) K80000 INH;												
409	1100000	11	00	1000110	00000000	00	000	1	0	00010	00000000	0	0
(016CH)	NOP(A) JFL(NSKIP,SKTP);												
410	1911011	11	00	0010111	11111111	00	000	1	0	00010	00000000	0	0
(016CH)	DWN3; LTM(AC);												
411	0100000	11	11	0101111	11111111	00	000	1	0	00010	00000000	0	0
(017CH)	SDR(A) FF1 JZR(FETCH);												
412	1010000	11	00	1010000	10000000	00	001	1	0	00010	00000000	0	0
(018CH)	S01; TZR(A) K80000 INH JCF(INC.CR);												
413	0011001	11	00	0010110	00000000	00	100	1	0	00010	00000000	0	0
(01ECH)	D11 LMI(R9) RIN JCC(DWN3);												
414	1100000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
(003DH)	AA1 NOP(A) JZR(FETCH);												
415	0011101	11	00	1100111	00000000	00	110	1	0	00010	00000000	0	0
(005DH)	CTX1 LMI(AC) RRM JPR(LA STA LX STX A1 SI M D ADM ORI XORI												
	AND0 C S7 SNZ GG);												

```

417 504:  CNO:  NOP(A) JCC(CNO):
      (00604) 110000 11 00 0011001 00000000 00 000 1 0 00010 00000000 0 0 0

418 704:  S7:  T7P(A) INH:
      (00704) 101000 11 00 0001001 11111111 00 001 1 0 00010 00000000 0 0 0

419 904:  NOP(A) JFL(S1A.S2):
      (00904) 110000 11 00 1011001 00000000 00 000 1 0 00010 00000000 0 0 0

420 9A4:  S1A:  LMT(P) FFI JZR(FETCH):
      (009A4) 0010011 11 11 0101111 00000000 00 000 1 0 00010 00000000 0 0 0

421 9B4:  S2:  NOP(A) JZR(FETCH):
      (009B4) 110000 11 00 0101111 00000000 00 000 1 0 00010 00000000 0 0 0

422 804:  DUNS:  CMR(A) JZR(FETCH):

```

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	CAUS	PAUSFF	STRR	LPCNT	EMIT	MULTF	SPRF
	6543210	10	10	6543210	76543210	10	210	0	0	43210	76543210	0	0
(0090H)	1110000	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
423	0A0H:			DNW7: ALP(R8) FF1:									
(00A0H)	0001000	11	11	0001011	11111111	00	000	1	0	00010	00000000	0	0
424	0B0H:			C0XXH: TZR(R8) JFL(NOCRY.CRY):									
(00B0H)	1011000	11	00	1001011	11111111	00	000	1	0	00010	00000000	0	0
425	110H:			DCX: ILR(R9):									
(0110H)	0001001	11	00	0010010	00000000	00	000	1	0	00010	00000000	0	0
426	120H:			CMA(AC):									
(0120H)	1111111	11	00	0010011	00000000	00	000	1	0	00010	00000000	0	0
427	130H:			ALR(X) FF1:									
(0130H)	0000001	11	11	0010100	11111111	00	000	1	0	00010	00000000	0	0
428	140H:			NOP(A) JFL(SP.NSP):									
(0140H)	1100000	11	00	1000000	00000000	00	000	1	0	00010	00000000	0	0
429	190H:			SMO1: TZR(A) KRO000 INH JCF(NC? CR??):									
(0190H)	1010000	11	00	1010011	10000000	00	001	1	0	00010	00000000	0	0

430	1004: (1004)	044: 001001	LMI(99) 11 00 0101111	ROT 00	TZR(FETCH): 00000000	00	001	1	0	00010	00000000	0	0
431	1504: (1504)	00: 0000000	ILP(A) 11 00 0111101	JCC(044): 00000000	00	000	1	0	00010	00000000	0	0	0
432	3E4: (003E4)	00: 1100000	NOP(A) 11 00 0000101	PAGE2 JCC(DWNR): 00000000	01	000	1	0	00010	00000000	0	0	0
433	5E4: (005E4)	044: 1100000	NOP(A) 11 00 0000110	JCC(SAF): 00000000	00	000	1	0	00010	00000000	0	0	0
434	6E4: (006E4)	S4F: 1010000	TZR(A) 11 00 0010110	INH JCC(SAE1): 00000001	00	001	1	0	00010	00000000	0	0	0
435	7E4: (007E4)	SN7: 1010000	TZR(A) 11 00 0001110	INH: 11111111	00	001	1	0	00010	00000000	0	0	0

XMAS VERS 2.0 PAGED EMULATOR MICROCODE FOR INTEL 3000

ERRORS= 0 PAGE 29

RECORD NUMBER	CPF	FI	FO	JUMP	KRUSS	PAGEF	CBUS	PAUSFF	STRB	LPCNT	EMIT	MULTF	SDPF
436	0FFH (00FFH)	110000	11	00	1001110	00	000	1	0	00010	00000000	0	0
				NOP(A) JFL(S3.S4):									
437	0EAH (00FAH)	110000	11	00	0101111	00	000	1	0	00010	00000000	0	0
				NOP(A) JZR(FETCH):									
438	0E8H (00E8H)	0010011	11	11	0101111	00	000	1	0	00010	00000000	0	0
				LMT(P) FF1 JZR(FETCH):									
439	11EH (011EH)	110000	11	00	0101111	00	000	1	0	00010	00000000	0	0
				NOP(A) JZR(FETCH):									
440	16FH (016FH)	110000	11	00	1000111	00	000	1	0	00010	00000000	0	0
				S4F1: NOP(A) JFL(FVEN.030):									
441	1E8H (01E8H)	110000	11	00	0101111	00	000	1	0	00010	00000000	0	0
				MH: NOP(A) JZR(FETCH):									
442	0FH (000FH)	0010011	11	11	0000001	00	110	1	0	00010	00000000	0	0
				FETCH: LMT(P) FF1 RM:									
443	1FH (001FH)	1011011	11	00	0000010	00	000	1	0	00010	00000000	0	0
				LTM(AC) K01FFF:									

```

444 2FH: SDR(R9) FF1 JPX(DIP J L REG I JF UJ MSC XK J J
445 (002FH) 0101001 11 11 1111011 1111111 00 000 1 0 00010 00000000 0 0
      IO: NOP(A) JCC(I21):
446 3FH: 1100000 11 00 0010100 00000000 00 000 1 0 00010 00000000 0 0
      LATUS: LMI(R9) FF1 RRM:
447 6FH: 0011001 11 11 0000101 00000000 00 110 1 0 00010 00000000 0 0
      LMI(AC) FF1:
448 5FH: 0101111 11 11 0111000 11111111 00 000 1 0 00010 00000000 0 0
      SDR(A) FF1 J7R(FETCH):
449 59H: 0100000 11 11 0101111 11111111 00 000 1 0 00010 00000000 0 0
      GG: NOP(A) JZR(FETCH):
450 7FH:

```


XMAS VERS 2.0 PAGED EMULATOR MICROCODE FOR INTEL 3000

RECORD NUMBER	CPE	FI	FO	JUMP	KRUSS	PAGFF	CRUS	PAUSFF	STPR	LPCNT	EMIT	MULTF	SPRF
	6543210	10	10	6543210	74543210	10	210	0	0	43210	76543210	0	0
	(007FH)	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0
451	11FH			NOP(A) SPO JZR(FETCH):									
	(011FH)	11	00	0101111	00000000	00	000	0	0	00010	00000000	0	0
452	14FH			NOP(A) JCC(JK):									
	(014FH)	11	00	0011101	00000000	00	000	1	0	00010	00000000	0	0
453	1DFH			JPR(N1,N2,N3,N4,N5,TRAP,N7,PWI,SMI,EO,N11,N12									
454	(01DFH)	11	00	1100110	00000000	00	000	1	0	00010	00000000	0	0
				01 00 4H 11):									
455	1EFH			NOP(A) JZR(FETCH):									
	(01EFH)	11	00	0101111	00000000	00	000	1	0	00010	00000000	0	0

456 EOF

NO PROGRAM ERRORS
END OF PROGRAM

MICROPROGRAM MEMORY IMAGE

0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FM
JCC	JZR	JZF	JZR												JCC
0010H	000FH	0012H	000FH												001FH
74	1	101	122												442
0	1	1	1												00
JCC	JZR	JZF	JZR												JCC
0020H	000FH	0020H	000FH												002FH
75	1	102	123												443
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
0040H	000FH	0040H	000FH												0010H
74	1	103	124												444
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
0070H	000FH	0070H	000FH												005FH
77	33	104	125												445
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
00A0H	000FH	00A0H	000FH												005FH
73	1	105	126												446
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
00E0H	000FH	00E0H	000FH												005FH
73	1	106	127												447
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
0120H	000FH	0120H	000FH												005FH
73	1	107	128												448
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
0140H	000FH	0140H	000FH												005FH
73	1	108	129												449
1	1	1	1												1
JCC	JZR	JZF	JZR												JCC
0160H	000FH	0160H	000FH												005FH
73	1	109	130												450
1	1	1	1												1

MICROPROGRAM MEMORY IMAGE

	0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DH	EH	FH
000H	JZR	JCF	JCR	JCC	JCR	JCC	JCR	JCC	JCR	JCC	JCR	JCC	JCR	JCC	JCR	JCC
001H	000FH	0002H	0002H	000FH	000FH	0005H	000FH	0007H	0009H	000FH	0009H	000FH	000FH	000FH	000FH	000FH
002H	82	109	128	149	161	177	177	279	295	317	338	366	397	422	178	
003H	JCC	JCF	JZR	JZR	JZR	JCC	JCC	JCC	JCC	JZR	JZR	JZR	JCC	JFL	JCR	JCR
004H	00A0H	0092H	00A2H	000FH	000FH	00A6H	00A7H	00A9H					00ACH	009AH	0096H	
005H	83	96	110	129	162	181	181	280	296		420	421	398	419	179	
006H	JCC	JZR	JZR	JCC	JCR	JCR	JCC	JCC	JCC		JFL	JCR	JCC	JCC		
007H	00B0H	000FH	000FH	00A4H	00A4H	00A5H	00A5H	00A7H	00B9H		00FAH	00ADH	00RCH	009DH		
008H	84	111	170	185	184	183	183	281	297		341	367	399	423		
009H	JCC	JZF	JCR	JCR	JCC	JCC	JCC	JCC	JCC		JZR	JFL	JCC	JFL	JCR	JCR
00A0H	00C2H	00B1H	00B8H	00B5H	00C5H	00C6H	00C6H	00C7H	00C9H		00DFH	00DAH	00CCH	009AH	00B2H	00B4H
00A1H	85	199	198	194	186	187	209	282	298		342	368	400	424	196	195
00A2H	JCC	JCR	JCR	JCR	JCC	JCC	JCC	JCC	JCC		JFL	JCR	JCC	JCR	JCR	JCR
00A3H	00D0H	00CFH	00B2H	00C9H	00C2H	00C4H	00D6H	00D7H	00D9H		00CAH	00CEH	00C0H	00CCH	00C1H	00D0H
00A4H	86	207	192	201	191	189	210	283	299	202	205	203	401	204	206	208
00A5H	JCC	JZR	JZR	JZR	JCC	JCC	JCC	JCC	JCC		JZR	JZR	JCC			
00A6H	00E0H	000FH	000FH	000FH	00E6H	00E6H	00E6H	00E7H	00E9H		000FH	000FH	010CH			
00A7H	87	112	131	131	211	211	284	284	360		343	369	402			
00A8H	JCC	JZF	JZR	JCF	JCC	JCC	JCC	JZR	JCC		JZR	JZR	JCC			
00A9H	00F0H	00F2H	000FH	00E2H	00F6H	00F6H	00F6H	000FH	00FAH		000FH	000FH	00FCH			
00AAH	88	113	132	150	212	212	285	285	361		437	438	403		436	
00ABH	JCC	JZR	JZR	JZR	JCC	JCC	JCC	JCC	JCC		JZR	JZR	JFL			
00ACH	0100H	000FH	000FH	000FH	0106H	0107H	0108H	0107H	0109H		000FH	000FH	00A4H			
00ADH	89	114	133	133	213	213	241	241	362		344	370	404			
00AEH	JCC	JZR	JZR	JZR	JCC	JCC	JCC	JCC	JCC		JZR	JZR	JCC			
00AFH	00FH	00FH	00FH	00FH	00FH	00FH	00FH	00FH	00FH		00FH	00FH	00FH			

MICROPROGRAM MEMORY IMAGE

0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	CH	DM	EH	FH
J7R 000FH			JPR 0110H		JCC 0136H	JCC 0127H	JCC 0127H	JCC 0136H	JCC 0136H	J7R 000FH	J7R 000FH	JCC 00E4H			
010H			174		214	242	242	379		345	371	405			
J7R 000FH	JCC 0122H	JCC 0123H	JCC 0124H	JCC 0124H	JCC 0126H	JCC 000FH	JCC 000FH	JCC 0126H	JCC 0129H	JCC 012AH	JCC 012AH	JCC 012CH	JCC 012DH	J7R 000FH	J7R 000FH
011H	07	115	136	151	163	215	286	354	319	346	372	406	425	432	451
012H	J7R 000FH	J7R 000FH	J7R 000FH	JCC 0134H	JCC 0135H	JCC 0156H	JCC 0137H	J7R 000FH	JCC 0139H	JCC 013AH	JCC 013AH	JCC 013CH	JCC 013DH		
		116	137	152	164	216	243	345	319	347	373	407	424		
J7R 000FH	JCC 0131H	JCC 0132H	JCC 0133H	JCC 0134H	JCC 0135H	JCC 0146H	JCC 0147H	JCC 0148H	JCC 0149H	JCC 014AH	JCC 014AH	JCC 014CH	JCC 014DH		
013H	08	117	138	153	165	217	244	348	320	348	374	408	427		
	J7R 000FH	J7R 000FH	J7R 000FH	JCC 0142H	JCC 0143H	JCC 0164H	JCC 0157H	JCC 0158H	JCC 0159H	JCC 015AH	JCC 015AH	JCC 016AH	JCC 016AH	JCC 016AH	JCC 016FH
014H		118	139	154	166	218	245	347	321	349	375	409	428	452	
J7R 000FH	JCC 0151H	JCC 0152H	JCC 0153H	JCC 0154H	JCC 0152H	JCC 0152H	JCC 0157H	JCC 0158H	JCC 000FH	JCC 000FH	JCC 000FH	JCC 000FH	JCC 000FH	JCC 000FH	JCC 000FH
015H	09	220	221		167	219	246	358	322	350	376				
	J7R 000FH	J7R 000FH	J7R 000FH	JCC 0176H	JCC 0177H	JCC 0177H	JCC 0177H	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	JCC 017CH	JCC 017CH	JCC 017AH	JCC 017AH
016H		119	140		222	247	247	349		351	377	410	440		
	J7R 000FH	J7R 000FH	J7R 000FH	JCC 0184H	JCC 0184H	JCC 0184H	JCC 0184H	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH
017H		120	141		223	248	248	350		352	378	411			
	J7R 000FH	J7R 000FH	J7R 000FH	JCC 0192H	JCC 0192H	JCC 0192H	JCC 0192H	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH	J7R 000FH

MICROPROGRAM MEMORY IMAGE

0H	1H	2H	3H	4H	5H	6H	7H	8H	9H	AH	BH	C4	DM	EH	FM
JFL	JCR	JCR	JCR	JCR	JCR	JCC	JCR	JCC	JCC	JFL	JFL	JCF			
0192H	0180H	0181H	0188H	01R24	01R4H	0194H	01R5H	019AH	0199H	019AH	01AAH	018AH			
255	254	253	275	251	250	224	249	276	323	353	379	412			
JCC	JCR	JCR	JCR	JCR	JCR	JCC	JCC	JFC	JCC	JZR	JZR	JCF			
01A0H	0190H	0191H	0194H	01954	0197H	01A6H	01A7H	01AAH	01A9H	000FH	000FH	01AAH			
259	258	257	264	265	266	225	267	277	324	354	3A0	429			
JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCR	JCC	JZR	JZR	JCF			
01A0H	0180H	0180H	0180H	0180H	0186H	0197H	01A7H	000FH	01A9H	000FH	000FH	01AAH			
260	260	260	260	260	226	268	268	278	325	355	3A1	429			
JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JCC	JFL	JFL	JCF			
01C0H	01C1H	01C1H	01C1H	01C6H	01C6H	01R1H	01R1H	01C9H	01C9H	01CAH	01DAH	01DAH			
261	270	270	270	227	227	269	269	326	326	356	3A2	429			
JCR	JCR	JCR	JCR	JCR	JCC	JFL	JFL	JZR	JZR	JZR	JZR	JCF			
01C1H	01C7H	01C7H	01C7H	0026H	01R2H	01R2H	01R2H	000FH	000FH	000FH	000FH	01AAH			
262	272	272	272	228	273	273	273	327	327	357	3A3	429			
JCR	JCR	JCR	JCR	JCR	JCR	JCR	JCR	JCR	JCR	JZR	JZR	JCF			
01D0H	01D0H	01D0H	01D0H	01D94	01D4H	01D5H	000FH	01D7H	01D8H	000FH	000FH	01AAH			
154	154	168	234	154	168	234	287	310	328	358	3A4	429			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	01R9H	000FH	000FH	01AAH			
92	100	121	142	155	169	233	288	311	329	359	3A5	413			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
01EH	01EH	01EH	01EH	01EH	01EH	01EH	01EH	01EH	01EH	01EH	01EH	01EH			
431	431	431	431	431	431	431	431	431	431	431	431	431			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	01DDH	000FH	000FH	01AAH			
453	453	453	453	453	453	453	453	453	453	453	453	453			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01F0H	01AAH			
455	455	455	455	455	455	455	455	455	455	455	455	455			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	015CH	000FH	000FH	01AAH			
455	455	455	455	455	455	455	455	455	455	455	455	455			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	01DDH	000FH	000FH	01AAH			
455	455	455	455	455	455	455	455	455	455	455	455	455			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	015CH	000FH	000FH	01AAH			
455	455	455	455	455	455	455	455	455	455	455	455	455			
JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JZR	JCC	JZR	JZR	JCF			
000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	000FH	01DDH	000FH	000FH	01AAH			
455	455	455	455	455	455	455	455	455	455	455	455	455			

XMAS VERS 2.0 PACFP EMULATOR MICROCODE FOR INTEL 3000

CROSS REFERENCE DIRECTORY

LABEL	REFERENCES
AI	77.145.(148).291.415
AA	(414).44E
AGN	219.(221)
AGN2	(138).415
AGN3	(139).416
ALS	176.(143)
ANDD	78.146.202.(365).416
ANEGP	741.(370)
AOA	(97).174
ADM	77.145.201.(296).415
ADOSP	741.(344)
ARS	138.(151)
BACK	176.(145)
C	78.146.202.(396).415
CAD	(115).417
CLX	399.(415)
CL	106.(158).237
CLI	172.(237)
CLR1	(81).174
COR	(432).44E
COXX	(424)
CP	(379).412
CP22	(342).420
CPY	(368).424
CPY2	749.(369)
CTBACK	(333).44E
CTP	402.(405)
CTX	(403).40E
Y	77.145.(240).291.415
YCX	175.(425)
YI	(413).454
YI4	(77).444
YIV	(253).273
YVI	(323).320
YV	(431).45E
YV2	273.(275)
YONE	(154).160
YWI	(164).276
YWP	(410).417
YW3	(430).431
YW4	378.(422)
YW5	747.(423)
YW7	472.(433)
YW4	175.(346)
YD	(320).453
YD	107.(122)
YD3	(132).151
YD6	(147).147
YD1	91.(124)
YD2	94.(129)
YD4	129.(131)
YD	175.(172)

CROSS REFERENCE DIRECTORY

LABEL	REFERENCES
EVEN	(352), 440
EX	135, (310)
FETCH	82, 90, 91, 92, 93, 95, 97, 98, 99, 100, 102, 107, 109, 111, 112, 114, 116, 118, 120, 121, 122, 123, 124, 127, 129, 130, 131, 132, 133, 137, 140, 141, 142, 149, 155, 158, 162, 169, 208, 232, 233, 278, 285, 286, 287, 288, 294, 305, 309, 311, 312, 317, 322, 327, 330, 331, 333, 334, 342, 343, 344, 345, 350, 351, 352, 354, 355, 357, 358, 359, 360, 361, 362, 366, 367, 370, 371, 374, 377, 378, 380, 381, 383, 386, 385, 411, 418, 420, 421, 422, 430, 437, 438, 439, 441, (442), 449, 450, 451, 455
80	78, 146, 292, 416, (450)
8T	110, (130)
8T5	113, (133)
8T6	119, (141)
WH	(441), 454
I	(143), 444
ICX	135, (406)
II	454, (455)
INIT	(74)
INITI	76, (79)
INIT2	79, (83)
IO	445, (446)
IO1	446, (452)
IX	(386), 445
J	(93), 444
JI	(156), 444
JJ	(312), 444
JJ	452, (453)
JK	(104), 444
L	77, (81), 145, 291, 415
LA	236, (447)
LAI	(170), 444
LAIBUS	(362), 445
LL	134, (215)
LRS	101, (102)
LT	77, (108), 145, 291, 415
LX	77, 145, (175), 291, 415
4	192, (119)
481	192, 196, (194)
482	199, (201)
4EX	(192), 199
4LP	(235), 444
4SC	(92), 453
4I	(359), 457
4I1	(385), 457
4I2	(100), 457
42	(121), 457
43	(142), 457
44	(155), 457
45	(233), 457
47	(353), 412
4C	(356), 420
4C2	202, (203)
4EG	(460), 367
4E01	335, (361)
4E66	(362), 424
4OCRY	

XMAS VRS P.C PACFD EMULATOR MICROCOPY FOR INTFL 3000

CROSS REFERENCE DIRECTORY

LABEL	REFERENCES
VOCRY2	(343).369
VOPP	135.(439
VOVEF1	353.(354)
VOVEF2	379.(381)
VOVEF3	356.(357)
VOVEF4	342.(384)
VSKIP	(351).403
VSP	(371).428
VTE01	(101).107
VTE05	(113).156
VTE04	(110).167
VTE0L	97.(103)
VTE0L2	94.(111)
VTE0L4	(112).120
VTE1	110.(111)
VTE15	113.(114)
VTE14	119.(120)
VTLT	101.(123)
VTSAME	(361).404
VV	(378).444
VV1	77.145.201.(316).415
VV2	219.(220)
VV3	(117).152
VV4	(118).164
VV5	194.(195)
VV6	203.(204)
VV7	205.(206)
VV8	353.(354)
VV9	(355).373
VV10	356.(383)
VV11	(358).382
VV12	242.(272)
VV13	215.(233)
VV14	203.(235)
VV15	303.(316)
VV16	202.(285)
VV17	(330).343
VV18	(331).333
VV19	(173).212
VV20	214.(217)
VV21	218.(222)
VV22	154.(124)
VV23	(313).320
VV24	(297).310
VV25	173.(200)
VV26	235.(238)
VV27	238.(270)
VV28	175.(165)
VV29	(125).466
VV30	125.(176)
VV31	(288).457
VV32	236.(315)

CROSS REFERENCE DIRECTORY

LEVEL	REFERENCES
RTI	(314), 315
RTN2	(313), 314
RTN3	313, (334)
SI	77, 145, (160), 291, 415
SIA	419, (420)
S2	419, (421)
S3	436, (437)
S4	436, (438)
SAE	236, 433, (434)
SAE1	434, (440)
SAME	(367), 404
SE	(80), 235
S6	(94), 235
SGC	94, (96)
SGE	(147), 235
SGEC	147, (150)
SKIP	(377), 400
SKN	236, (336)
SKN1	(335), 334
SKP	236, (364)
SKP1	(363), 364
SL	(107), 235
SLE	(159), 235
SLEC	159, (167)
SMI	(311), 457
SNE	(126), 234
SNEC	126, (128)
SNO	236, (417)
SNQ1	417, (429)
SNZ	78, 146, 202, 416, (435)
SO	236, (389)
SOI	389, (412)
SOME	245, (264)
SP	(345), 424
SW	135, (304)
SPP	134, (286)
STA	77, (95), 145, 291, 415
STA	77, (127), 145, 291, 415
SZ	78, 146, 202, 416, (418)
SZER0	255, (257)
TCA	134, (134)
TRAP	(169), 457
W	(99), 220
W	(249), 444
XOR1	77, 145, 291, (337), 415
XX	134, (153)
XXX	201, (202)
YY	(98), 117
ZZ	216, (219), 221

DISTRIBUTION

	No. of Copies		No. of Copies
Defense Documentation Center Cameron Station Alexandria, VA 22314	12	US Army Materiel Systems Analysis Activity ATTN: DRXSY-MP Aberdeen Proving Ground, Maryland 21005	1
Deputy Chief of Research and Development Department of the Army Washington, D. C. 20310	1	University of California Dept. of Electrical Engineering ATTN: Dr. Jack Sklansky Irvine, California 92664	2
Office, Assistant Chief of Staff for Force Development Department of the Army ATTN: FORSD, MG W. A. Burke FORSD-FS, COL J. N. Hale Washington, D. C. 20310	1 1	US Army Research Office ATTN: Information Processing Office Box CM, Duke Station Durham, NC 27706	1
Commander US Army Combat Developments Command Field Artillery Agency ATTN: COCFA-WC, F. Shelton Fort Sill, Oklahoma 73503	1	Defense Advanced Research Project Agency ATTN: Maj. Karam Dr. Tegnella 1400 Wilson Boulevard Arlington, Virginia 22209	1 1
Commander US Army Weapons Command ATTN: AMSWE-RD, COL C. Alter J. Brinkman AMSWE-RDP, Dr. M. Amoruso AMSWE-RDF, G. Schlenker AMCPM-CAWS, H. Noble SWEER-R-ST, H. Garver Rock Island, Illinois 61201	1 1 1 1 1	Martin Marietta Corp. ATTN: A. C. Layton P. O. Box 5837 MP 251 Orlando, Florida 32805	1
Director Aberdeen Research and Development Center ATTN: AMXRD-BAD, Dr. R. Eichelberger Aberdeen Proving Ground, Maryland 21005	2	Texas Instruments, Inc. ATTN: L. J. Freeman Dr. Clay Shultz P. O. Box 6015 Dallas, Texas 75222	1 1
Commander Picatinny Arsenal ATTN: SMUPA-DWO, J. Brooks Dover, New Jersey 07801	1	Dr. Dave Casasent Carnegie-Mellon University Pittsburgh, PA 15213	1
Commander US Army Materiel Command ATTN: DRCRD-MT-MT 501 Eisenhower Avenue Alexandria, Virginia 22304	5	DRSMI-LP, Mr. Voigt -W DRDMI-X Commander -T, Dr. Kobler -T, Mr. James J. Fagan -TE, Mr. Lindberg -TE, Mr. Pittman -TEO, Dr. Minor Mr. Sitton Mr. Widenhofer Mr. Wylie Mr. Farmer Dr. Kulas -C, Maj. W. Ward Maj. W. Adams -TG, Mr. J. Huff -HF, Mr. N. Mangns -TD, Dr. McCorkle -TI, (Record Set) (Reference Set)	1 1
Director Night Vision Laboratory (US AECOM) ATTN: Dr. John Dehue DRESL-NV-VI Dr. George Jones Fort Belvoir, Virginia 22060	2		