HP 13255

GRAPHICS M-CONTROLLER MODULE

Manual Part No. 13255-91125

REVISED

MAY-04-78

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

NOTE: This document is part of the 264XX DATA TERMINAL product series Technical Information Package (HP 13255).

1.0 INTRODUCTION.

Graphics M-controller PCA together with the Graphics Display PCA display vectors on the terminal's screen when commanded so by the Processor PCA.
M-controller's major tasks are to read the image memory in normal or zoom mode, to refresh the image memory, to generate and store vectors or cursor in the image memory.
The vectors are specified by the Processor PCA in form of a startpoint, slope and length.
The M-controller interfaces with the Processor PCA via connector P1 and with the Graphics Display PCA through the connector P2.

2.0 OPERATING PARAMETERS.

A summary of operating parameters for the Graphics M-controller is contained in tables 1.0 through 5.4.

Table 1.0 Physical Parameters

Part Number	Nomenclature	Size (L x W x D) +/-0.100 Inches	Weight (Pounds)
02640-60125	Graphics M-controller	12.5 x 4.00 x 0.5	0.5
Number of Backplane slots Required: 1			

Table 2.0 Reliability and Environmental Information

Environmental: (X) HP Class B ( ) Other:
Restrictions: Type tested at product level
Failure Rate: 1.49 (percent per 1000 hours)

Table 3.0 Power Supply and Clock Requirements - Measured
(At +/-5% Unless Otherwise Specified)

+5 Volt Supply	+12 Volt Supply	-12 Volt Supply	-42 Volt Supply
@ 1.76 A	N/A	N/A	N/A

115 volts ac	220 volts ac
N/A	N/A

Clock Frequency: 10.5 MHz

Table 4.0 Connector Information Graphics M-controller PCA

Connector and Pin No.	Signal Name	Signal Description
P1, Pin 1	+5V	+5 Volt Power Supply
-2	GND	Ground Common Return (Power and Signal)
-3		Not used
-4		
-5	$\overline{\text{ADDR0}}$	Negative true, address bit 0
-6	$\overline{\text{ADDR1}}$	Negative true, address bit 1
-7	$\overline{\text{ADDR2}}$	Negative true, address bit 2
-8	$\overline{\text{ADDR3}}$	Negative true, address bit 3
-9	$\overline{\text{ADDR4}}$	Negative true, address bit 4
-10	$\overline{\text{ADDR5}}$	Negative true, address bit 5
-11	$\overline{\text{ADDR6}}$	Negative true, address bit 6
-12		Not used
-13	$\overline{\text{ADDR8}}$	Negative true, address bit 8
-14	$\overline{\text{ADDR9}}$	Negative true, address bit 9
-15	$\overline{\text{ADDR10}}$	Negative true, address bit 10
-16	$\overline{\text{ADDR11}}$	Negative true, address bit 11
-17		Not used
-18		
-19		
-20		
-21	$\overline{\text{I/O}}$	Negative true, Input/Output memory
-22	GND	Ground Common Return (Power and Signal)

Table 4.0 Connector Information Graphics M-controller PCA (Cont'd.)

Connector and Pin No.	Signal Name	Signal Description
P1, Pin A	GND	Ground Common Return (Power and Signal)
-B		Not Used
-C		
-D		
-E	$\overline{BUS0}$	Negative true, Data Bus Bit 0
-F	$\overline{BUS1}$	Negative true, Data Bus Bit 1
-H	$\overline{BUS2}$	Negative true, Data Bus Bit 2
-J	$\overline{BUS3}$	Negative true, Data Bus Bit 3
-K	$\overline{BUS4}$	Negative true, Data Bus Bit 4
-L	$\overline{BUS5}$	Negative true, Data Bus Bit 5
-M	$\overline{BUS6}$	Negative true, Data Bus Bit 6
-N	$\overline{BUS7}$	Negative true, Data Bus Bit 7
-P	$\overline{WRITE}$	Negative true, Write Strobe
-R		Not used
-S		
-T	PRIOR IN	Bus Controller Priority Out
-U	PRIOR OUT	Bus Controller Priority Out
-V		Not Used
-W		
-X		
-Y	$\overline{REQ}$	Negative true, Bus Data Valid
-Z		Not Used

Table 4.1 Connector Information Graphics M-controller PCA

Connector and Pin No.	Signal Name	Signal Description
P2,Pin 1	A0	Col or Row Address, Bit 0
-2	A1	Col or Row Address, Bit 1
-3	A2	Col or Row Address, Bit 2
-4	A3	Col or Row Address, Bit 3
-5	A4	Col or Row Address, Bit 4
-6	A5	Col or Row Address, Bit 5
-7	A6	Col or Row Address, Bit 6
-8	X0	Bit Address, Bit 0
-9	X1	Bit Address, Bit 1
-10	X2	Bit Address, Bit 2
-11	X3	Bit Address, Bit 3
-12	RAS	Row Address Strobe
-13	CAS	Column Address Strobe
-14	$\overline{W}$	Negative True, Write Enable
-15	$\overline{LOAD}$	Negative True, Load

Table 4.1 Connector Information Graphics M-controller PCA (cont'd.)

Connector and Pin No.	Signal Name	Signal Description
P2, Pin -A	CLK	10.5 MHZ Clock
-B	GND	Ground
-C	$\overline{\text{CLK}}$	Negative True, 10.5MHZ Clock
-D	$\overline{\text{103.D2}}$	Negative True, Col 103 and Dot 2
-E	DI	Data In
-F	A7	Address Bit 7
-H	n	Innibit Graphics Display
-J	$\overline{\text{STR3}}$	Negative True, Strobe 3
-K	$\overline{\text{STR4}}$	Negative True, Strobe 4
-L	$\overline{\text{STR5}}$	Negative True, Strobe 5
-M	VR	Vertical Retrace
-N	$\overline{\text{STR6}}$	Negative True , Strobe 6
-P		Not used
-R	SAMPLE	Sample Bit
-S	GND	Ground

Table 5.0 Module Bus Pin Assignments

Function Performed: Load registers, load B-buffer, set and clear flags as specified by A0,A5,A6

Poll Bit: Not Applicable

Module Address: (ADDR 4,11,10,9) = (1100)

Function Specifier:
Bits A0,A5,A6 determine which register or flag will be stroped.

Value	Bus Signal
1	ADDR 15
0	ADDR 14
0	ADDR 13
0	ADDR 12
1	ADDR 11
0	ADDR 10
0	ADDR 9
X	ADDR 8
X	ADDR 7
A6	ADDR 6
A5	ADDR 5
1	ADDR 4
X	ADDR 3
X	ADDR 2
X	ADDR 1
A0	ADDR 0
B7	BUS 7
B6	BUS 6
B5	BUS 5
B4	BUS 4
B3	BUS 3
B2	BUS 2
B1	BUS 1
B0	BUS 0

1=Logical 1=Bus Low
0=Logical 0=Bus High
X=Don't Care

A6	A5	A0	Signal	Function
1	1	1	CLR VR	Clear Vert. Retr. Latch
1	1	0	RESET	Clear Address Counter
1	0	1	STR5	Load Mode Reg
1	0	0	STR4	Load Pattern Reg
0	1	1	STR3	Load Prescaler Reg
0	1	0	STR2	Clock Flags
0	0	1	STR1	Load B-buffer bits 8-11
0	0	0	STR0	Load B-buffer bits 0-7

Data Bus Bit Interpretation: See Tables 5.1 through 5.4

Table 5.1 Module Bus Pin Assignments

Function Performed: Load B-buffer with vector drawing parameters	Value	Bus Signal
	1	ADDR 15
	0	ADDR 14
	0	ADDR 13
	0	ADDR 12
	1	ADDR 11
Poll Bit: Not Applicable	0	ADDR 10
	0	ADDR 9
	A8	ADDR 8
	X	ADDR 7
	0	ADDR 6
	0	ADDR 5
Module Address: (ADDR 4,11,10,9) = (1100)	1	ADDR 4
	A3	ADDR 3
Function Specifier:	A2	ADDR 2
Bits A1,A2,A3,A8 specify location in	A1	ADDR 1
the B-buffer that is loaded.	A0	ADDR 0
Bit A0 specifies lower or upper half		
of the B-buffer.	B7	BUS 7
	B6	BUS 6
	B5	BUS 5
Data Bus Interpretation: See Table below.	B4	BUS 4
See also Section 4.	B3	BUS 3
	B2	BUS 2
	B1	BUS 1
<--------A0=1----------> <-----A0=0---------	B0	BUS 0

	Upper 4 Bits				Lower 8 Bits									
A8	A3	A2	A1	B7	B6	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	X	X	X	X	F	X	X	X	X	X	X	X	X	X	X	Y/N	draw 1st dot
0	0	0	1	X	X	X	X	write Dot Count												
0	0	1	0	X	X	X	X	Display Control Byte												
0	0	1	1	X	X	X	X	F	X	X	X	X	X	X	X	X	X	X	Y/N	cont.test
0	1	0	0	X	X	X	X	F	X	X	X	X	X	X	X	X	X	X	Y/N	self-test
0	1	0	1	X	X	X	X	F	X	X	X	X	X	X	X	X	X	X	Y/N	new address
0	1	1	0	X	X	X	X	Upper 6 bits of vector start addr.												
0	1	1	1	X	X	X	X	Lower12 bits of vector start addr.												
1	0	0	0	X	X	X	X	Initial Slope Value												
1	0	0	1	X	X	X	X	Initial Vector Length												
1	0	1	0	X	X	X	X	Upper 6 bits of vector displ. M2												
1	0	1	1	X	X	X	X	Lower12 bits of vector displ. M2												
1	1	0	0	X	X	X	X	Upper 6 bits of vector displ. M1												
1	1	0	1	X	X	X	X	Lower12 bits of vector displ. M1												
1	1	1	0	X	X	X	X	Slope increment D2												
1	1	1	1	X	X	X	X	Slope increment D1												

Table 5.2 Module Bus Pin Assignments

Function Performed: Load B-buffer with zoom parameters

Poll Bit: Not Applicable

Module Address: (ADDR 4,11,10,9) = (1100)

Function Specifier:
Bits A1,A2,A3,A8 specify location in the B-buffer that is loaded.
Bit A0 specifies lower or upper half of the B-buffer.

Data Bus Interpretation: See table below.
See also Section 4.

Value	Bus Signal
1	ADDR 15
0	ADDR 14
0	ADDR 13
0	ADDR 12
1	ADDR 11
0	ADDR 10
0	ADDR 9
A3	ADDR 8
X	ADDR 7
0	ADDR 6
0	ADDR 5
1	ADDR 4
A3	ADDR 3
A2	ADDR 2
A1	ADDR 1
A0	ADDR 0
B7	BUS 7
B6	BUS 6
B5	BUS 5
B4	BUS 4
B3	BUS 3
B2	BUS 2
B1	BUS 1
B0	BUS 0

<-----A0=1-----> <-----A0=0-----

Upper 4 Bits				Lower 8 Bits													
A8	A3	A2	A1	B7	B6	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0
0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	0	0	1	X	X	X	X	write Dot Count											
0	0	1	0	X	X	X	X	Display Control Byte											
0	0	1	1	X	X	X	X	Word Count per Line											
0	1	0	0	X	X	X	X	Magnification-1											
0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	0	0	0	X	X	X	X	Upper 6 Bits of Zoom Start Addr.											
1	0	0	1	X	X	X	X	Lower 12 Bits of Zoom Start Addr.											
1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table 5.3 Module Bus Pin Assignments

Function Performed: Load B-buffer with cursor parameters	Value	Bus Signal
	1	ADDR 15
	0	ADDR 14
	0	ADDR 13
	0	ADDR 12
	1	ADDR 11
Poll Bit: Not Applicable	0	ADDR 10
	0	ADDR 9
	A8	ADDR 8
	X	ADDR 7
	0	ADDR 6
	0	ADDR 5
Module Address: (ADDR 4,11,10,9) = (1100)	1	ADDR 4
	A3	ADDR 3
Function Specifier:	A2	ADDR 2
Bits A1,A2,A3,A8 specify location in the B-buffer that is loaded.	A1	ADDR 1
Bit A0 specifies lower or upper half of the B-buffer.	A0	ADDR 0
	B7	BUS 7
	B6	BUS 6
	B5	BUS 5
Data Bus Interpretation: See Table below. See also Section 4.	B4	BUS 4
	B3	BUS 3
	B2	BUS 2
	B1	BUS 1
<--------A0=1----------> <-----A0=0--------|	B0	BUS 0

	Upper 4 Bits				Lower 8 Bits									
A8	A3	A2	A1	B7	B6	B5	B4	B3	B2	B1	B0	B7	B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
1	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
1	0	1	0	X	X	X	X	Start address +720 (upper 6 bits)												Vertical
1	0	1	1	X	X	X	X	Start address +720 (lower 12 bits)												vector
1	1	0	0	X	X	X	X	Start address - 1 (upper 6 bits)												Horizontal
1	1	0	1	X	X	X	X	Start address - 1 (lower 12 bits)												vector
1	1	1	0	X	X	X	X	Vertical Vector Length												
1	1	1	1	X	X	X	X	Horizontal Vector Length												

Table 5.4 Module Bus Pin Assignments

Function Perrormed: Set/Clear Flags

Poll Bit: Not Applicable

Module Address: (ADDR 4,11,10,9) = 1100

Data Bus Bit Interpretation:

B0	1/0 set/clear flag 1
B1	1/0 set/clear flag 3
B2	not used
B3	1/0 set/clear flag 4
B4	1/0 set/clear flag 5
B5	not used
B6	not used
B7	not used

Value	Bus Signal
1	ADDR 15
0	ADDR 14
0	ADDR 13
0	ADDR 12
1	ADDR 11
0	ADDR 10
0	ADDR 9
X	ADDR 8
X	ADDR 7
0	ADDR 6
1	ADDR 5
1	ADDR 4
X	ADDR 3
X	ADDR 2
X	ADDR 1
0	ADDR 0
B7	BUS 7
B6	BUS 6
B5	BUS 5
B4	BUS 4
B3	BUS 3
B2	BUS 2
B1	BUS 1
B0	BUS 0

1=Logical 1=Bus Low
0=Logical 0=Bus High
X=Don't Care

3.0 FUNCTIONAL DESCRIPTION. Refer to the block diagram (figure 1), schematic diagram (figure 2), component location diagram (figure 3), instruction format (figure 4), image memory address (figure 5), image memory bit displacement (figure 6), zoom example (figure 7), flowcharts A,B,C (figures 8,9,and 10), and parts list (02640-60125) located in the appendix. Refer also to the graphics microcode listing in section 6.0.

The Graphics M-Controller PCA consists of an address counter, ROM, ROM output register, instruction decoder, decoder enable flip-flop, condition selector, halt, A-buffer, hold register A, B-buffer, hold register B, address multiplexer, constant multiplexer, adder, carry flip-flop, sign flip-flop, S11 flip-flop, address register, bit register, RAS and CAS flip-flop, bus gates, bus decoder, flags, and test logic blocks.

To accomplish the tasks described by the algorithms outlined in section 4.0, the M-controller is designed as a microprogrammable machine. It is provided with a set of 8 instructions and 256 words of control storage, each 20 bits wide.

The hardware has basically two buffers A and B, both 16 words long and 12 bits wide. Each buffer can load its own hold register A,B. Both registers are addends of a 12-bit wide adder. The output of the adder is routed back to the buffer A.
The buffer B is loaded from the 2648 bus. Four load instructions can load the hold registers with a buffer location. Hold register B can be loaded also with a constant. STO instruction stores the result of the addition back in buffer A. The computation is done in 2's complement.

To branch within the microcode, a conditional jump with 8-bit target address is provided. To indicate certain states of the hardware, FLG instruction can set or clear seven hardware flags. In addition FLG instruction can halt the address counter. Three LSB's of the ROM field provide the timing signals for the image memory.

M-controller instruction set. (Refer to figure 4).

```
LAB     hold reg A:= buffer A(A);
        hold reg B:= buffer B(B);
        if Z=1 then hold reg A:=0;
        send bits (w,R,C) to image memory;

LAC     hold reg A:= buffer A(A);
        hold reg B(bit 0-7):= CONSTANT 1;
        hold reg B(bit 8-11):= 0;
        send bits (w,R,C) to image memory;

LDC     hold reg A:= 0;
        hold reg B:= CONSTANT 2;
        send bits (w,R,C) to image memory;

LDA     hold reg A:= buffer A(A);
        send bits (w,R,C) to image memory;
```

```
STO     if C=1 then sum S:= hold reg A + hold reg B + carry in
               else sum S:= hold reg A + hold reg B;
        buffer A(A):= sum S;
        carry ff:= carry out;
        if S=1 then sign ff:= S(11);
        if X=1 then bit reg XR:= S0-S3;
        if L=1 then if M=1 then address reg AR:= S0-S5,S11 ff;
                    if M=0 then addr reg AR:= S4-S10;
        send bits (W,R,C) to image memory;
JMP     if COND=TRUE then address counter AC:= TARGET;
        send bits (W,R,C) to image memory;

        Conditions:   0  unconditional        8  flag 6
                      1  carry ff             9  not used
                      2  not carry ff         10 sign ff
                      3  not flag 1           11 test not OK
                      4  not flag 2           12 flag 5
                      5  flag 3               13 not sign ff
                      6  not flag 4           14 not vertical retr.
                      7  vertical retr.       15 not used

FLG      if F= 0  then  nop;
         if F= 1  then set not flag 1;
         if F= 2  then clear not flag 2;
         if F= 3  then set not flag 2;
         if F= 4  then set not flag 4;
         if F= 5  then set flag 5;
         if F= 6  then set flag 6;
         if F= 7  then clear flag 6;
         if F= 8  then nop;
         if F= 9  then nop;
         if F=10  then enable self-test;
         if F=11  then disable self-test;
         if F=12  then set vertical retrace latch;
         if F=13  then nop;
         if F=14  then send STR6 to image memory;
         if F=15  then set H bit;
         if H1=1 then set halt 1;
         if H2=1 then set halt 2;
         send bits (W,R,C) to image memory;

NOP      send bits (W,R,C) to image memory;
```

Instruction execution speed.

An instruction cycle takes two clocks. One clock for fetch and one for execute. To speed up the code execution, the M-controller works in pipelined fashion. While one instruction is being executed, the next instruction is being fetched. Since the clock period is 95 nsec, the instruction execution rate is 10.5 MHz. This is true for all instructions exept the successful jump, where the execution takes two clocks (test and jam).
Since a new line of microcode is loaded into the ROM output register (ROR) on every clock, the image memory timing is controlled directly from ROR bits 0,1,2 (signals RAS,CAS,WRITE).

3.1 ADDRESS COUNTER (AC).

Address Counter (AC) provides the 8-bit address for the ROM. It is driven by 10.5 MHz clock. If signal $\overline{\text{TESTOK}}$ is asserted by the JMP condition selector, the address specified by ROR 5-12 is jammed into the Address Counter and branch in the code is accomplished. AC can be from advancing by Halt circuit. It can be initialized to zero by the $\overline{\text{RESET}}$ signal from the Bus Decoder.

3.2 READ ONLY MEMORY.

Read Only Memory provides storage for the microcode. It is 256 words long and 20 bits wide.
ROM is composed of 5 PROM's Harris 7611-5 or equivalent.

3.3 ROM OUTPUT REGISTER (ROR).

ROM Output Register (ROR) is 20 bits wide and holds the instruction being executed. New instruction is clocked in on every clock. The signal $\overline{\text{RESET}}$ clears opcode bits thus forcing NOP into the instruction decoder.

3.4 INSTRUCTION DECODER (IR).

The IR decodes the three most significant bits of the ROR. There is one instruction decoded per one clock period (95 nsec) as long as it is enabled by Decoder Enable flip-flop.

3.5 DECODER ENABLE FLIP-FLOP.

Decoder Enable FF disables the next instruction after JMP from being executed if the jump condition was found OK. It forces one dead clock when successful branch occurs.

3.6 CONDITION SELECTOR.

Condition Selector selects one of the 16 possible branching conditions. The condition is determined by ROR 13-16 bits. The selected condition and the decoded JMP signal determine the outcome of the $\overline{\text{TESTOK}}$ line. The possible jump conditions are:

0	unconditional	8	flag 6
1	carry ff	9	not used
2	not carry ff	10	sign ff
3	not flag 1	11	test not OK
4	not flag 2	12	flag 5
5	flag 3	13	not sign ff
6	not flag 4	14	not synch vert. retrace
7	synch vert. retrace	15	not used

3.7 HALT.

Halt circuit enables halting the Address Counter programatically by issuing the FLG instruction. The FLG instruction and bit ROR 11 on, halts the Address Counter until the Graphics Display asserts the synch pulse $\overline{103.D2}$ that re-enable the counter. FLG instruction and ROR bit 12 on, halts the Address Counter until the Graphics Display sends the next $\overline{\text{LOAD}}$ pulse. The halts are used to synchronize the M-controller and the image memory when reading and displaying a word.

3.8 A-BUFFER.

3.8.1 A-buffer is a 16-word RAM, 12 bits wide. It holds different variables as shown on the flowchart, figure 10. The microcode can write into or read any location of the A-buffer, but it is not accessible to the 8080 Processor. The STO instruction stores the result from the adder into a location of the A-buffer specified by bits ROR 13-16.
The instructions LAB,LAC or LDA loads a location specified by ROR 13-16 from the A-buffer into the Hold Register A.
Since the bit ROR 4 high disables the RAM, any LOAD instruction with ROR 4 high will load zero into the Hold Reg A.

3.8.2 A-buffer consists of three RAM's 74S189. The signal WE low strobes the data into the buffer. When the RAM's are disabled by ROR 4, the tri-state outputs are pulled high by external resistors. This loads effectively zero into the Hold Register A, since the register uses the complementary outputs.

3.9 HOLD REGISTER A.

3.9.1 Hold Register A holds one addend of the addition. It is loaded by the instruction LAB, LAC or LDA with the content of a location specified by ROR 13-16. Zero is loaded into the register by instruction LDC or LAB if ROR 4 is set. The other instructions do not affect the register.

3.9.2 The register consists of three 74LS175's. Complementary outputs are used to compensate for the inversion in RAM's. Since the bit ROR19 is set only in LOAD instructions, the trailing edge of signal ROR19.$\overline{CLK}$ is used as strobe.

3.10 B-BUFFER.

3.10.1 B-buffer is a 16-word RAM, 12 bits wide. It holds variables as shown in the flowchart, figure 10. The microcode can only read a location out of the B-buffer into the Hold Register B. The B-buffer is loaded by the Processor via the terminal bus.
The Address Multiplexer selects either the signals ROR 9-12 or $\overline{ADDR\ 1,2,3,8}$ as address for the B-buffer, depending whether the buffer is read out or written into. The 12 bit word is loaded into the buffer by the Processor in two bytes: the bus decoder signal $\overline{STR0}$ strobes bus data $\overline{BUS0-7}$ into the bit position 0-7 and $\overline{STR1}$ loads $\overline{BUS0-3}$ into the bit position 8-11 of a location specified by the address $\overline{ADDR1,2,3,8}$.

3.10.2 B-buffer consists of three RAM's 74S189. Since RAM inverts the input data, the negative true bus data appears as positive at the outputs. It should be noted, that since the bus address $\overline{ADDR1,2,3,8}$ is negative true the Processor has to complement the address when loading the B-Buffer.

3.11 HOLD REGISTER B.

3.11.1 This register holds one addend of the addition. If the instruction LAB loads the Hold Register B, the addend is a location from the B-buffer. If LAC is executed then the register bits 0-7 are loaded with ROR5-12 (constant 1) and bits 8-11 are zero-ed. Finally, the register is loaded with bits ROR5-16 if LDC is executed.

3.11.2 Hold register B is composed of three multiplexer registers 74LS298. The inputs are switched to the B-buffer, when the LAB signal is true, otherwise the signals ROR's are used as inputs. The trailing edge of the signal ROR19.DE.$\overline{LDA}$.CLK is used as a strobe.

3.12 ADDRESS MULTIPLEXER.

This multiplexer selects the address for the B-buffer. If either the flag F1 or F4 is set then ROR 9-12 address is used. Otherwise the bus address $\overline{ADDR}$ 1,2,3,8 is selected.

3.13 CONSTANT MULTIPLEXER.

The multiplexer is used for loading either CONSTANT 1 or CONSTANT 2 into Hold register B. When a Load instruction other than LDC is decoded the multiplexer supplies zero to the Hold register B. If LDC is decoded then the bits ROR 13-16 are used as the upper four bits of the constant.

3.14 ADDER.

The adder adds two numbers in Hold Registers A and B. The output of the Carry FF is added to the sum if enabled by the bit ROR6. The carry out and MSB of the adder are made available to the Carry and Sign FF. The adder outputs are valid as long as Hold Registers A,B are not loaded with new values or carry in is not disabled by ROR6. The sum is forwarded to the A-buffer, Address Register AR and Bit Register XR.

3.15 CARRY FLIP-FLOP.

It saves the carry out when the STO instruction is executed and makes it available to the adder in the next addition, provided ROR6 is on. Both outputs of the FF are used as jump conditions.

3.16 SIGN FLIP-FLOP.

It stores the MSB of the addition when the instruction STO is executed and the bit ROR5 is set.
The sign FF set to one means negative number. Both outputs of the FF are used as jump conditions.

3.17 S11 FLIP-FLOP.

Similarly, as with the Sign FF, the S11 FF saves MSB of the addition but data is clocked in by the positive transition of the signal ROR1.

3.18 ADDRESS REGISTER (AR).

3.18.1 The Address Register has the primary function of holding the address when M-controller accesses the image memory. Since the image memory is organized as 16K words times 1 bit RAM, it takes 18 bits to address a single bit. Four bits are needed to point out the bit within a word and 14 upper bits pick the word within the chip. 16K RAMs use address multiplexing. First 7 row address bits are sent followed by 7 column bits. (See figure 5).
Since adder and RAM buffers are only 12 bits wide, it requires two locations in A or B buffer to store the complete memory address.
It should be noted that to increment the image memory bit address means to add one to the LSB, while to increment the word address involves adding one in bit position 4 (add decimal 16).
Note that the 7 bit row address is contained fully in one location WALO or RALO (write address low or read address low); 6 bits of the 7 bit column address are in WAHI or RAHI (write addr.high or read addr.high) but one bit is in WALO (or RALO). Therefore, the LSB of the column address is stored in the S11 FF when WALO (or RALO) is being sent to the image memory.
The process of sending the memory address then lies in two steps:

1. Load WALO (or RALO) into the Hold Registers. Issue STO instruction with ROR7 high and ROR3 low. This loads the Adder bits S4-10 into the Address Register. Send ROR1 (RAS) to the memory, thus strobing the row address and saving the MSB in S11 FF.

2. Load WAHI (or RAHI) into the Hold Registers. Issue STO instruction with ROR3 and ROR7 high. This loads the Adder bits S0-5 and S11 FF into the Address Register. Then send ROR0 (CAS) to the memory, thus strobing the column address.
Besides sending the memory address through the Address register, the M-controller dispatches the Display Control Byte to the Graphics Display via the Address Register. The 8 upper bits are sent through the AR and 4 lower bits via the Bit Register (XR). (see description of Graphics Display PCA in module section 13255-91126).

3.18.2 Address Register consists of two multiplexer registers 74LS298. The select input is controlled programatically by ROR3. The data is strobed in the AR when the STO instruction is decoded and the bit ROR7 is on.

3.19 BIT REGISTER (XR).

This register holds the lowest four bits of the memory address or four lowest bits of the Control Display Byte. During self-test, in case of an error the four lowest bits of the failing address are saved here. Data from the adder are loaded in the register when the STO instruction is decoded and the bit ROR8 is set.

3.20 RAS & CAS FLIP-FLOP.

The flip-flops send the RAS and CAS signals to the image memory. Since they are clocked with $\overline{CLK}$, the signals RAS and CAS are shifted half a clock (47 nsec) behind the leading clock edge. This allows to propagate the address through the Address Register and drivers on the Graphics Display PCA before it is clocked in.

3.21 BUS GATES.

When enabled by the READ signal, the gates propagate the contents of the bit register XR, flag F5, signal $\overline{NF1.NF4}$, latched image memory data and the signal VRIN.

3.22 BUS DECODER.

Bus Decoder decodes signals sent by the Processor to the graphics hardware (module no. is 14B), and passes them in the form of strobes to the other inner blocks as follows:

$\overline{STR0}$ strobes bus data into B-buffer bit 0-7

$\overline{STR1}$ strobes bus data into B-buffer bit 8-11

$\overline{STR2}$ clocks flags F1,F3,F4 and F5

$\overline{STR3}$ loads prescaler on the Graphics Display PCA

$\overline{STR4}$ loads pattern reg on the Graphics Display PCA

$\overline{STR5}$ loads mode reg on the Graphics Display PCA

$\overline{STR6}$ is the signal $\overline{RESET}$ which initializes Address Counter to zero

$\overline{STR7}$ clears the Vertical Retrace latch

3.23 FLAGS.

3.23.1 The block called Flags is a set of flip-flops that indicates the internal states of the graphics hardware and firmware.

Flag F1 when set indicates the input B-buffer was loaded by the Processor with new data. It is set by the Processor and cleared by the M-controller's FLG instruction.

Flag F2 when set indicates the vector drawing is in progress. It is set and cleared by the FLG instruction.

Flag F3 when set indicates the zoom mode is on. It is cleared and set by the Processor.

Flag F4 when set indicates the input B-buffer was loaded with new zoom parameters. It is set by the Processor and cleared by the FLG instruction.

Flag F5 has two meanings:
First, if set by the M-controller during self-test, it signals to the Processor that the self-test failed. Before self-test it must be cleared by the Processor.
Second, if during vertical retrace both flags F1 and F5 are set by Processor, it makes the M-controller to draw cursor.

Flag F6 is used for reentry from WRITE subroutine in the graphics firmware. It is set and cleared by FLG instruction.

Flag T when reset enables self-test logic. It is set and cleared by the FLG instruction.

Flag H when set, forces the Graphics Display PCA to send blanks to the screen. It is set by the FLG instruction and reset by the synch signal $\overline{103.D2}$.

Vertical Retrace Latch when set by the FLG instruction indicates to the Processor that vertical retrace occurred. It is cleared by the Processor.

STR6 is a strobe that loads the Display Control Byte onto the Graphics Display PCA.It is asserted by the FLG instruction.

3.23.2 The Flag logic contains decoders, flip-flops and R-S latches. The two decoders 74LS138 decode the signals ROR 13-16 when enabled by the signal FLG. The decoder outputs manipulate the flags. The Flags F1,F3, F4,F5 are flip-flops that use bus data as inputs and are clocked by signal STR2 from bus decoder. Note that STR2 is NANDed with CLK for synchronization purposes.
When the Processor sets flag 1 or flag 4, it actually makes Q-outputs of the 74LS74 low since the bus data is negative true. That's why the signals are called NF1,NF4. In case of flag 3 and flag 5 the $\overline{Q}$-outputs are used, so that when these flip-flops are set by Processor the names F3,F5 corresponds to high level signals.

3.24 TEST LOGIC.

3.24.1 This block provides the self-test feature when enabled by the T-flag. The image memory data bit DI is latched and compared with the expected value SAMPLE loaded by the Processor into the Graphics Display PCA. Resulting signal OK is a condition for the JMP instruction. Latched memory data bit DATA is made available to the 8080 via Bus Gates.

3.24.2 Memory data bit DI is strobed by the leading edge of ROR2 (the WRITE signal to the memory) when read portion of the R-M-W cycle is finished. SAMPLE is a latched data bit sent by the Processor. When signal NT is low the compare logic is enabled and the signal NOK is determined by SAMPLE and DATA.

4.0 GENERAL.

This section outlines the image memory organization and in terms of algorithms describes the three tasks the graphics hardware performs: vector plotting, line displaying and zooming.

4.1 IMAGE MEMORY SIZE.

If the screen of 720 by 360 dots is viewed as two dimensional array, it takes 19 bits to address a dot and memory size of 2^{19} bits (0<=x<=719 ; 0<=y<=359). But if the memory is organized as a linear array where each screen dot is assigned a number, then the screen fits into the size of 2^{18} bits.
To meet this objective the memory array is organized as 16,200 words, 16 bits wide. Each 45 words cover one raster line (45*16=720). Given screen dot address x,y the corresponding memory address is:

```
bit addr   45*y*16 + x      (18 bits)
word addr  45*y + x/16      (14 bits)
```

4.2 VECTOR PLOTTING ALGORITHM.

Given a cartesian grid, a vector with a slope delta y/delta x less than 45 degrees can be plotted as follows:

step 1: Set d=-1/2+(delta y/delta x). Set X and Y to the x,y coordinates of the startpoint of the line.
step 2: Place a dot at coordinates X,Y.
step 3: If X equals x coordinate of the end of the line, stop. Otherwise go to step 4.
step 4: If d is negative set d=d+(delta y/delta x) and set X=X+1. Go step 2.
If d is positive set d=d+(delta y/delta x)-1 and set X=X+1, Y=Y+1. Go to step 2.

Now the different addressing scheme between screen and the linear memory must be considered. Given a dot on the screen, the adjacent dots have eight different displacements in the image memory. (see Figure 6) Therefore for the first octant, the step 4 of the algorithm is changed:
screen increment X=X+1 is substituted by memory increment M=M+1
and increment X=X+1,Y=Y+1 is substituted by M=M-719.
Similarly, for each octant there are two memory displacements for two adjacent points on the screen. Vector plotting algorithm then can be modified with regard to the linear memory arrangement:

Given two points on the screen P0(x0,y0), P1(x1,y1). P0 is the beginning.
step 1: Set delta x=x1-x0, delta y=y1-y0
Determine which octant the vector will be plotted in.
Assign the memory displacement M1,M2 according to the octant.
Calculate the vector start address WA=45*(359-y0)*16+x0
Set the initial value D=2*delta y - delta x
and increments D1=2*delta y, D2=2*delta y - 2*delta x.
Set the dot count (vector length) DC=delta x.
step 2: Write a dot at address WA
step 3: Decrement the dot count DC=DC-1. If DC=0 then done ; else go to step 4.
step 4: If D is positive then update D=D+D2 and address WA=WA+M2. Go to step 2.
If D is negative then update D=D+D1 and address WA=WA+M1. Go to step 2.

4.3 LINE DISPLAYING ALGORITHM.

Displaying a horizontal line requires reading consecutively 45 words each 16 bits wide, converting them to a serial stream and sending the bits to monitor. Whenever 45 words have been read, the beam is in horizontal retrace and new read operation must be synchronized with the raster. To display full frame requires to read 360 lines.

step 1: Initialize line count LC=0 and read address RA=0.
step 2: Initialize word count WC=0. wait for the raster to begin the line.
step 3: Read a word at address RA, convert 16 parallel bits to a serial stream and direct it to the monitor. Update address RA=RA+1 and word count WC=WC+1.
step 4: If WC=45 then go to step 5 else go to step 3.
step 5: Increment line count LC=LC+1,
if LC=360 then frame DONE else go to step 2.

4.4 ZOOM AND PAN ALGORITHM.

Zoom is a feature that displays a bit from memory, for a given magnification m in the form of (m-1)*(m-1) dots, followed by one blank row and one blank column. (see figure 7)
To stretch the line horizontally on the screen in the zoom mode, requires to send each dot (m-1) times to the monitor, followed by a blank dot, before shifting to the next dot.
This means to reduce the shifting frequency of P-S convertor to 21MHz/m. Vertical extention is achieved by reading the same line (m-1) times. Start point is specified by start zoom address ZASTR.

step 1: Initialize current zoom address ZA=ZASTR, line zoom address ZA0=ZASTR, word count per line K=45/m, line count LC=0.
step 2: Initialize repeat count RC=0.
step 3: Initialize word count WC=0, wait for raster to begin line.
step 4: Read a word at ZA, update ZA=ZA+1 and WC=WC+1.
step 5: Wait till serial conversion of the word is finished.
step 6: If WC<K then go to step 4,
if WC=K then proceed to step 7.
step 7: Increment LC=LC+1,
if LC=360 then frame DONE else proceed to step 8.
step 8: Update RC=RC+1.
if RC<m-1 then reinitialize ZA=ZA0, go to step 3.
if RC=m-1 then do one blank line and update ZA0=ZA0+45, ZA=ZA0 go to step 2.

4.5 TASK PARTITIONING BETWEEN GRAPHICS M-CONTROLLER AND PROCESSOR PCA.

The graphics M-controller serves the following functions:

a) It reads and refreshes the image memory either in normal or zoom mode. By changing zoom start address, panning through the image memory is accomplished.

b) During the horizontal retrace it generates and stores a vector into the image memory. Vector specification is recieved from the Processor in the form of an endpoint, vector length and vector slope.
The vector is generated by turning on the dots in the image memory that most closely approximate the straight line between the endpoints.

c) During vertical retrace it generates and stores graphics cursor into the image memory if desired by the Processor.

d) It reads any point on the screen and makes it known to the Processor, if so desired.

e) It can diagnose the whole image memory and identify RAM chip failure.

The Processor shares some of the tasks:
(Refer to sections 5.0 through 5.5, the flowcharts figures 8,9,and 10 and the graphic microcode listing in section 6.0).

a) Read and refresh is the sole function of the graphics hardware Read in the zoom mode is specified by the Processor into the B-buffer on the M-controller board in the form of magnification M, word count per line K, zoom start address ZASTRLO, ZASTRHI and display control byte.

b) Vector generation during horizontal retrace is done as outlined in section 4.2. Processor does the step 1 of the algoritnm and outputs the vector parameters into the B-buffer on the M-controller board. The parameters are: start memory address WALO,WAHI, dot count DC, initial slope D, slope increments D1, D2, image memory increments M1LO, M1HI, M2LO, M2HI. 8080 Processor indicates by setting MSB in the location B(10) whether the endpoint of the last vector is the beginning of the new one or if the new WA should be used. If self-test is requested, Processor sets MSB in B(11) or if test is to be continued after previous failure, MSB in B(12) is set. The number of dots generated per scan line is passed in the location B(14).

c) The graphics cursor is generated as one horizontal and one vertical vector. The Processor has to specify both vectors by start points WA1LO, WA1HI, WA2LO,WA2HI and by length DC1,DC2.

5.0 PROCESSOR FIRMWARE SPECIFICATION.

The Processor causes the graphics M-controller to perform the following tasks: vector drawing,self-test,zoom,cursor drawing and reading a raster bit.
It does so by loading the input B-buffer,scanning and setting flags on the M-controller as described below.
Since the bus address is negative true, the Processor has to complement the address before loading the B-buffer.
For example if the Processor specifies address 15, it actually loads buffer location B(0) etc. Then to load the buffer see Tables 5.0 through 5.4.

5.1 VECTOR DRAWING.

Processor action		Comment:
if Bus (0)=0 then		if F1,F4 reset then load B-buffer
Begin		
B (0):=delta D1	*	load slope increment D1
B (1):=delta D2	*	slope increment D2
B (2):=delta M1LO	*	lower 12 bits of mem. displacement M1
B (3):=delta M1HI	*	upper 6 bits of mem. displacement M1
B (4):=delta M2LO	*	lower 12 bits of mem. displacement M2
B (5):=delta M2HI	*	upper 6 bits of mem. displacement M2
B (6):=DC	**	intial vector length
B (7):=D		initial slope
B (8):=new WALO		lower 12 bits of vector start addr.
B (9):=new WAHI		upper 6 bits of vector start addr.
B(10):=4000B/0		use new start addr/use last addr.
B(11):=0		do not start self-test
B(12):=0		do not continue self-test
B(13):=Display Byte ***		Graphics Display Control Bits
B(14):=Write Dot Count**		4 dots in normal read,3 dots in zoom
B(15):=4000B/0		Draw 1st dot/Don't draw 1st dot
Load Mode register		Mode Reg on Graphics Display PCA
Set F1		set Flag 1
End;		

* only negative values loaded in 2's complement
**load always as negative 2's complement
***for meaning of Display Control Bits refer to the Graphics Display Module (13255-91126).

5.2 SELF-TEST.

Processor action:

A. Start self-test:

```
if Bus (0)=0 then
Begin
  load B-buffer as under vector drawing
  but B (11):=4000B
  load SAMPLE bit into Graphics Display PCA
  F1:=1
  F5:=0
End;
```

B. Evaluate self-test:

```
if F1=0 and F5=0 then test OK; go DONE;
if F1=0 and F5=1 then test failed;
```

C. Continue self-test:

```
Begin
  B (12):=4000B
  F1:=1
  F5:=0
  go B
End;
```

5.3 ZOOM.

A. Start Zooming.

Processor action:		Comment:
if Bus (0)=0 then		if F1,F4=0 then load B-buffer
Begin		
B(11):=M	*	load zoom magnification-1
B(12):=K	*	load no. of words per line
B (6):=ZASTRLO		lower 12 bits of zoom start addr.
B (7):=ZASTRHI		upper 6 bits of zoom start addr.
B(13):=Display Byte		Graphics display control bits
F3:=1		set flag F1
F4:=1		set flag F4
End;		

*load as negative 2's complement

B. End Zooming.

```
if Bus(0)=0 then F3:=0;
```

5.4 CURSOR.

```
  Processor action:                    Comment:

if Bus(0)=0 and Bus(5)=1 then      if F1,F4=0 and vertical latch
Begin                                is set then load B-buffer
  B (0):=DC1              *        length of horizontal vector
  B (1):=DC2              *        length of vertical vector
  B (2):=WA1LO                     start addr. of noriz.vector-1
  B (3):=WA1HI
  B (4):=WA2LO                     start addr. of vert. vector+720
  B (5):=WA2HI
  F1:=1                            set flag F1
  F5:=1                            set flag F5

                          * load as negative 2's complement
```

5.5 READ A RASTER BIT.

```
  Processor action:                    Comment:

if Bus (0)=0 then                  if F1,F4=0 then load B-buffer
Begin
  B (6):=7777B                     vector length=1 (2's complement)
  B (8):=WALO                      12 lower bits of raster addr.
  B (9):=WAHI                      6 upper bits of raster addr.
  B(10):=4000B                     use new addr.
  B(11):=0                         do not start self-test
  B(12):=0                         do not continue self-test
  B(15):=4000B                     draw 1st dot
  Mode Reg                         load Mode reg.
  F1:=1                            set flag F1
```

6.0 GRAPHICS MICROCODE LISTING.

The following microcode is described by flowcharts A, B and C (figures 8, 9, and 10). It is stored in M-controller's ROMs packs U14,U15,U16,U17,U26.

```
*************************************************
000 0420000 PON    FLG F=01                                  NF1 <= 0
001 0460000        FLG F=03                                  NF2 <= 0
002 0500000        FLG F=04                                  NF4 <= 0
003 0560000        FLG F=07                                  F6  <= 0
004 0660000        FLG F=11                                  DISABLE SELF TEST
005 1300340 A      JMP C=12 T=CURSR                          IF F5 = 1 DO CURSOR
006 1002300        JMP C=00 T=A0                               ELSE GO TO A0
007 1062300 CURSR  JMP C=03 T=A0                             IF NF1 GOTO A0
010 2000020        LAB A=00 B=00          Z                  COPY DC 1
011 1400000        STO A=00
012 2042020        LAB A=02 B=02          Z                  COPY WA1LO
013 1520000        STO A=05
014 2063020        LAB A=03 B=03          Z                  COPY WA1HI
015 1540000        STO A=06
016 2520040 CURS1  LAC A=05 C=+0001                           WA1LO <= WA1LO +1
017 1520600        STO A=05          XL
020 2540002        LAC A=06 C=+0000             R            WA1HI <= WA1HI + CY
021 1540312        STO A=06           LC     M R
022 2021033        LAB A=01 B=01          Z M RC             COPY DC2
023 1420013        STO A=01                 M RC
024 2400053        LAC A=00 C=+0001         M RC             DC1 <= DC1 + 1
025 1400013        STO A=00                 M RC
026 1040707        JMP C=02 T=CURS1          WRC             IF CURSOR 1 NOT DONE, LOOP
027 2104027        LAB A=04 B=04          Z  WRC             COPY WA2LO
030 1620000        STO A=09
031 2125020        LAB A=05 B=05          Z                  COPY WA2HI
032 1640000        STO A=10
033 3323020 CURS2  LDC C=-0720
034 3620000        LDA A=09                                  WA2LO <= WA2LO - 720
035 1620600        STO A=09          XL
036 2657742        LAC A=10 C=+7777B            R            WA2HI <= WA2HI + C
037 1640312        STO A=10           LC     M R
040 0000013        NOP                       M RC
041 0000013        NOP                       M RC
042 2420053        LAC A=01 C=+0001          M RC            DC2 <= DC2 + 1
043 1420013        STO A=01                  M RC
044 1041547        JMP C=02 T=CURS2           WRC            IF CURSOR 2 NOT DONE, LOOP
045 0420007        FLG F=01                   WRC            NF1 <= 0
046 1162740 A0     JMP C=07 T=A2                             IF VR THEN GO A2
047 3374020 A3     LDC C=-0064                               ELSE REFRESH 64 ROWS
050 1760000        STO A=15                                  WC = -64
051 2721000 REF    LAC A=13 C=+0016                          SEND ROW ADDRESS
052 1720200        STO A=13           L
053 2760042        LAC A=15 C=+0001             R            WC = WC + 1
054 1760002        STO A=15                     R
```

```
055 1042442        JMP C=02 T=REF              R       IF WC LESS THAN 0 THEN LOOP
056 1342342        JMP C=14 T=A3               R       IF NOT VR DO 64 MORE ROWS
057 0700000 A2     FLG F=12                            SET VERT RETRACE FLAG
060 1132340        JMP C=05 T=ZOOM                     IF F3 THEN GO ZOOM
061 1263240        JMP C=11 T=A1                       DONT SEND DISPL BITS IF NOK
062 3377760        LDC C=+7777B                        SEND DISPLAY CONTROL BITS
063 1760600        STO A=15          XL                TO THE MEMORY
064 0740000        FLG F=14
065 3351420 A1     LDC C=-0360                         LC<=-360
066 1400000        STO A=00
067 3375160        LDC C=-0045                         WC<=-45
070 1760000        STO A=15
071 3377020        LDC C=-0016                         RALO<=-16
072 1520000        STO A=05
073 3003760        LDC C=+0077B                        RAHI<=+77B
074 1540000        STO A=06
075 1163640 L1     JMP C=07 T=L1                       IF VR THEN L1
076 0000001        NOP                          C      CLOCK THE CLEAR MEMORY LATCH
077 2521000 L2     LAC A=05 C=+0016                    RALO<=RALO+16
100 1520200        STO A=05          L
101 2540002        LAC A=06 C=+0000            R       RAHI<=RAHI+ C
102 1540312        STO A=06          LC    M R
103 0404013        FLG F=00 HLT1             M RC      HALT1
104 0000013        NOP                       M RC
105 0000013        NOP                       M RC
* CONTROLLER HALTED UNTIL NEXT 103.D2 PULSE
106 2760040        LAC A=15 C=+0001                    WC<=WC+1
107 1760000        STO A=15
110 0000000        NOP
111 0000000        NOP
112 0000000        NOP
113 0000000        NOP
114 2521000 READ   LAC A=05 C=+0016                    READ LOOP  RALO<=RALO+16
115 1520200        STO A=05          L
116 2540002        LAC A=06 C=+0000            R                  RAHI<=RAHI+C
117 1540312        STO A=06          LC    M R
120 2760053        LAC A=15 C=+0001          M RC                 WC<=WC+1
121 1760013        STO A=15                  M RC
122 1044613        JMP C=02 T=READ           M RC                 IF WC<0 THEN GO
123 2400053        LAC A=00 C=+0001          M RC      LC = LC + 1
124 1400000        STO A=00
125 1020240        JMP C=01 T=A                        IF LC=0 THEN GO A
126 0540000        FLG F=06                            F6<=1
127 1005600        JMP C=00 T=WRITE                      GO WRITE
130 0560000 RET1   FLG F=07                            RE-ENTRY POINT 1 :F6 <= 0
131 3375160        LDC C=-0045                               WC<= -45
132 1760000        STO A=15
133 1003740        JMP C=00 T=L2                           GO READ AGAIN
134 1110100 WRITE  JMP C=04 T=LF1                      IF F2=0 THEN GO LF1
135 2376020        LAB A=15 B=14           Z           LOAD WORD COUNT
```

```
136 1760000        STO A=15
137 2420000        LAC A=01 C=+0000
140 1420040        STO A=01              S            SET S FROM SIGN OF D
141 1246640 w0     JMP C=10 T=w1                       IF D<0 THEN GO W1
142 2064000        LAB A=03 B=04                       WALO<= WALO+M2LO
143 1460600        STO A=03           XL
144 2105002        LAB A=04 B=05                 R     WAHI<= WAHI+M2HI+CARRY
145 1500312        STO A=04            LC      M R
146 2021013        LAB A=01 B=01               M RC    D<= D+ D2
147 1420053        STO A=01              S     M RC
150 2440053        LAC A=02 C=+0001            M RC    DC<= DC+1
151 1440013        STO A=02                    M RC
152 1027707        JMP C=01 T=DONE              WRC    IF DC=0 THEN GO DONE
153 1007307        JMP C=00 T=w2                WRC    ELSE GO w2
154 0000007        NOP                          WRC
155 2062000 w1     LAB A=03 B=02                       WALO<= WALO+MILO
156 1460600        STO A=03           XL
157 2103002        LAB A=04 B=03                 R     WAHI<= WAHI+MIHI+CARRY
160 1500312        STO A=04            LC      M R
161 2020013        LAB A=01 B=00               M RC    D<= D+ D1
162 1420053        STO A=01              S     M RC
163 2440053        LAC A=02 C=+0001            M RC    DC<=DC+1
164 1440013        STO A=02                    M RC
165 1027707        JMP C=01 T=DONE              WRC    IF DC=0 THEN GO DONE
166 1267547 w2     JMP C=11 T=BAD               WRC    IF DATA NOK THEN GO BAD
167 2760040        LAC A=15 C=+0001                    WC<= WC+1
170 1760000        STO A=15
171 1046040        JMP C=02 T=W0                       IF WC=0 THEN RETURN
172 1010000        JMP C=00 T=RETRN
173 2460000 BAD    LAC A=03 C=+0000
174 1460400        STO A=03           X                 BIT REG <= BAD ADDR 0-3
175 0520000        FLG F=05                            F5<= 1
176 0420000 DONE   FLG F=01                            F1<= 0
177 0460000        FLG F=03                            F2<= 0
200 1205400 RETRN  JMP C=08 T=RET1
201 1015440        JMP C=00 T=RET2
202 1070000 LF1    JMP C=03 T=RETRN                    IF NF1 GO RETURN
203 1310000        JMP C=12 T=RETRN                    RETURN IF CURSOR FLAG ON
204 2374020        LAB A=15 B=12           Z           IF B(12)=1 THEN
205 1760040        STO A=15              S              CONTINUE
206 0440000        FLG F=02                               SELF TEST.
207 1250000        JMP C=10 T=RETRN
210 2046020        LAB A=02 B=06           Z           A(2)<= INIT DC
211 1440000        STO A=02
212 2027020        LAB A=01 B=07           Z           A(1)<= INIT D
213 1420000        STO A=01
214 2372020        LAB A=15 B=10           Z           IF B(10)=1 THEN
215 1760040        STO A=15              S               FETCH NEW WA
216 1331140        JMP C=13 T=w4                       IF NOT SF THEN GO w4
217 2070020 w3     LAB A=03 B=08           Z           FETCHING
```

```
220 1460000       STO A=03                                    NEW  WALO
221 2111020       LAB A=04 B=09          Z                       AND
222 1500000       STO A=04                                        WAHI
223 2460000 W4    LAC A=03 C=+0000
224 1460600       STO A=03          XL                    PUT WALO IN AR
225 2500002       LAC A=04 C=+0000                  R
226 1500212       STO A=04           L        M R       PUT WAHI IN AR
227 2373033       LAB A=15 B=11            Z M RC       IF B(11)=1 THEN
230 1760053       STO A=15               S   M RC        ENABLE SELF TEST
231 0640013       FLG F=10                   M RC
232 1251613       JMP C=10 T=W5              M RC        ELSE
233 0660013       FLG F=11                   M RC        DISABLE SELF TEST
234 2377033 W5    LAB A=15 B=15            Z M RC        IF B(15) = 1
235 1760053       STO A=15               S   M RC       THEN
236 1332113       JMP C=13 T=W6              M RC       WRITE
237 0000003       NOP                          RC       THE FIRST
240 0000007       NOP                         WRC       DOT OF
241 0000007       NOP                         WRC       THE VECTOR
242 2440040 W6    LAC A=02 C=+0001                       DC<= DC+1
243 1440000       STO A=02
244 1267540       JMP C=11 T=BAD                         IF DATA NOK GO BAD
245 1050000       JMP C=02 T=RETRN                       IF DC<0 THEN NOT DONE
246 1007700       JMP C=00 T=DONE                       GO TO DONE
247 1153200 ZOOM  JMP C=06 T=Z0                          IF OLD ZOOM   GO Z0
250 2166020       LAB A=07 B=06          Z               ELSE A(7)<=ZASTRTLO
251 1560000       STO A=07
252 2207020       LAB A=08 B=07          Z                    A(8)<=ZASTRTHI
253 1600000       STO A=08
254 2353020       LAB A=14 B=11          Z                    A(14)<= -M
255 1740000       STO A=14
256 2274020       LAB A=11 B=12          Z                    A(11)<= -K
257 1660000       STO A=11
260 2375020       LAB A=15 B=13          Z               SEND DISPLAY CONTROL
261 1760600       STO A=15          XL                   TO THE MEMORY
262 0740000       FLG F=14                                 STROBE 6 SENT TO MEMORY
263 0500000       FLG F=04                                    NF4<= 0
264 3351420 Z0    LDC C=-0360                                 LC<= -360
265 1400000       STO A=00
266 2560000       LAC A=07 C=+0000                            ZALO<= ZASTRTLO
267 1520000       STO A=05
270 2600000       LAC A=08 C=+0000                            ZAHI<= ZASTRTHI
271 1540000       STO A=06
272 2560000       LAC A=07 C=+0000                            ZAOLO<= ZASTRLO
273 1620000       STO A=09
274 2600000       LAC A=08 C=+0000                            ZAOHI<= ZASTRHI
275 1640000       STO A=10
276 1173700 ZVR   JMP C=07 T=ZVR                         WAIT FOR END OF VERT RTRACE
277 2740001 B     LAC A=14 C=+0000             C              RC <= -M (CLOCK CLEAR
300 1700000       STO A=12
301 2660000 Z1    LAC A=11 C=+0000                            WC<= -K
```

```
302 1760000        STO A=15
303 2520000        LAC A=05 C=+0000                          READ 1ST WORD
304 1520200        STO A=05           L                        ZALO
305 2540002        LAC A=06 C=+0000             R
306 1540312        STO A=06           LC     M R                ZAHI
307 2760053        LAC A=15 C=+0001           M RC
310 1760013        STO A=15                   M RC               WC<=WC+1
311 0404013        FLG F=00 HLT1              M RC               HALT 1
312 0000013        NOP                        M RC
313 0000013        NOP                        M RC
*              CONTROLLER HALTED UNTIL  103.D2   PULSE
314 2521000 ZLOOP  LAC A=05 C=+0016                          ZOO LOOP
315 1520200        STO A=05           L                        ZALO
316 2540002        LAC A=06 C=+0000             R
317 1540312        STO A=06           LC     M R                ZAHI
320 0410013        FLG F=00 HLT2              M RC               HALT 2
321 2760053        LAC A=15 C=+0001           M RC                 WC<=WC+1
322 0000013        NOP                        M RC
*              CONTROLLER HALTED UNTIL NEXT LOAD PULSE
323 1760013        STO A=15                   M RC
324 1054613        JMP C=02 T=ZLOOP           M RC               IF WC<0 GO ZLOOP
325 2400040        LAC A=00 C=+0001                              LC<= LC+1
326 1400000        STO A=00
327 1020240        JMP C=01 T=A                          IF LC = 0 THEN GO A
330 1005600        JMP C=00 T=WRITE                                GO WRITE
331 2700040 RET2   LAC A=12 C=+0001                       RENTRY 2: RC<= RC+1
332 1700000        STO A=12
333 1036040        JMP C=01 T=Z2                         IF RC = 0 THEN GO Z2
334 2620000        LAC A=09 C=+0000                       ZALO<= ZAOLO
335 1520000        STO A=05
336 2640000        LAC A=10 C=+0000                       ZAHI<= ZAOHI
337 1540000        STO A=06
340 1014040        JMP C=00 T=Z1
341 3055020 Z2     LDC C=+0720                           ZA0 <= ZA0 + 720
342 3620000        LDA A=09
343 1520000        STO A=05
344 1620000        STO A=09
345 2640000        LAC A=10 C=+0000                      ZA <= ZA0
346 1540100        STO A=06             C
347 2540000        LAC A=06 C=+0000
350 1640000        STO A=10
351 3374020        LDC C=-0064                            WC <= -64
352 1760000        STO A=15
353 0404000        FLG F=00 HLT1                                 HALT 1
354 0000000        NOP
355 0000000        NOP
```

```
*                CONTROLLER HALTED UNTIL NEXT  103.D2  PULSE
356 0760000        FLG F=15                              TURN THE DISPLAY OFF
357 2721000 RLOOP  LAC A=13 C=+0016                      SEND
360 1720200        STO A=13           L                   ROW ADDRESS
361 2760042        LAC A=15 C=+0001           R          WC<= WC+1
362 1760002        STO A=15                   R
363 1056742        JMP C=02 T=RLOOP           R          IF WC<0 THEN LOOP
364 2400042        LAC A=00 C=+0001           R          LC<= LC+1
365 1400000        STO A=00
366 1020240        JMP C=01 T=A                          IF LC=0  GO A
367 1013740        JMP C=00 T=B                            ELSE GO  B.
370 0000000        NOP
END
```

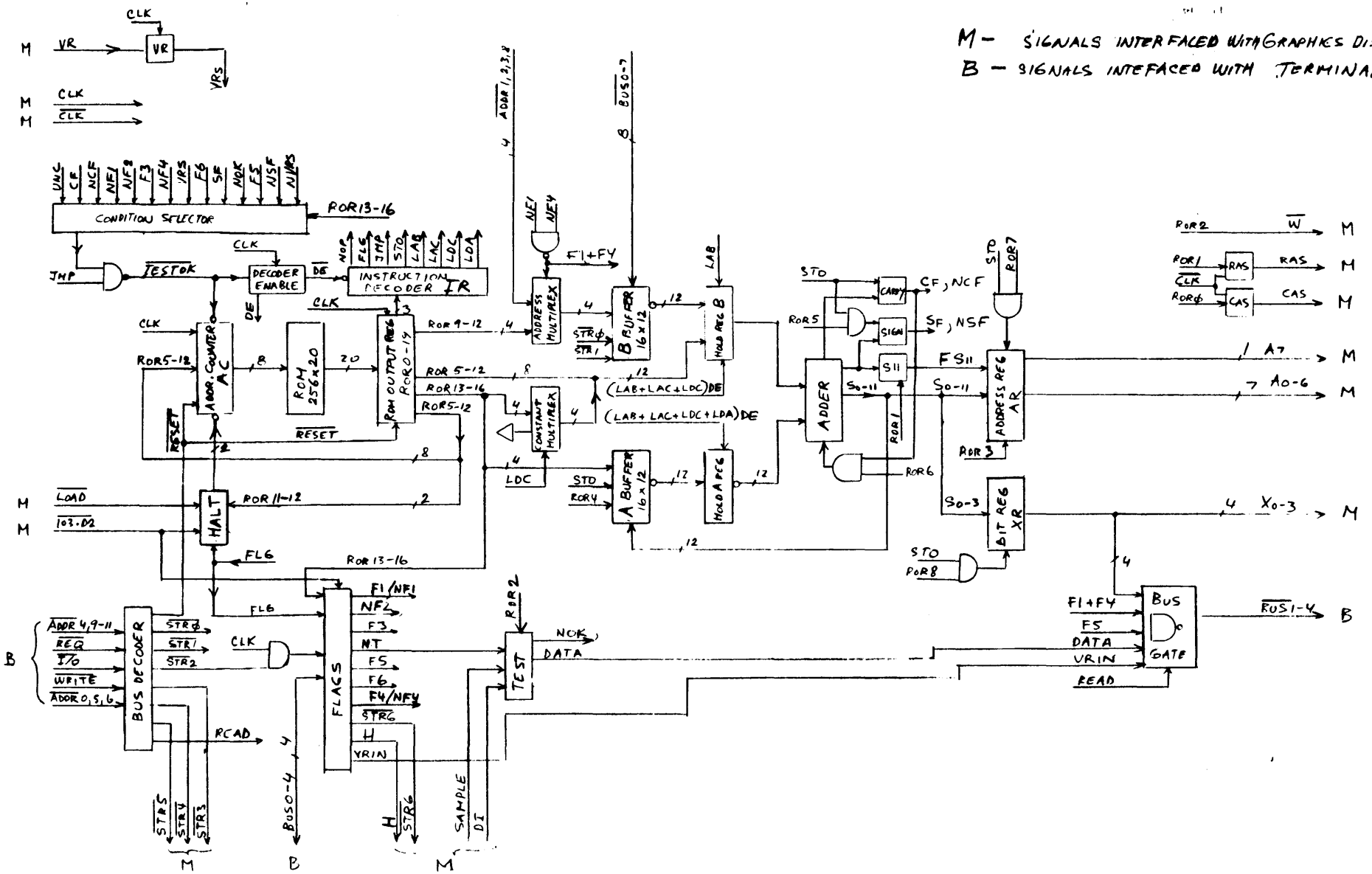

Figure 1
Graphics M-controller Block Diagram
MAY-04-78 13255-91125

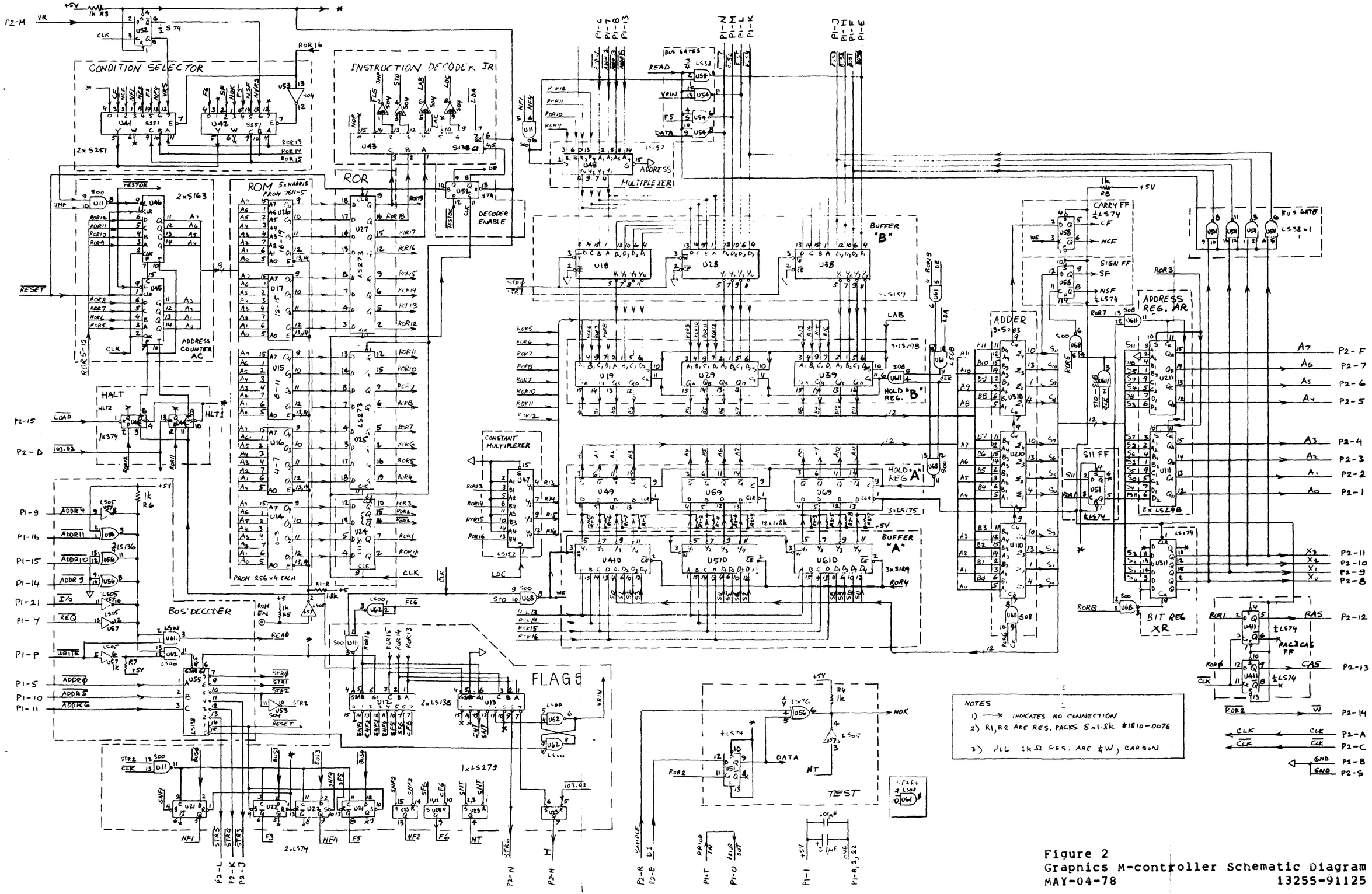

Figure 2
Graphics M-controller Schematic Diagram
MAY-04-78 13255-91125

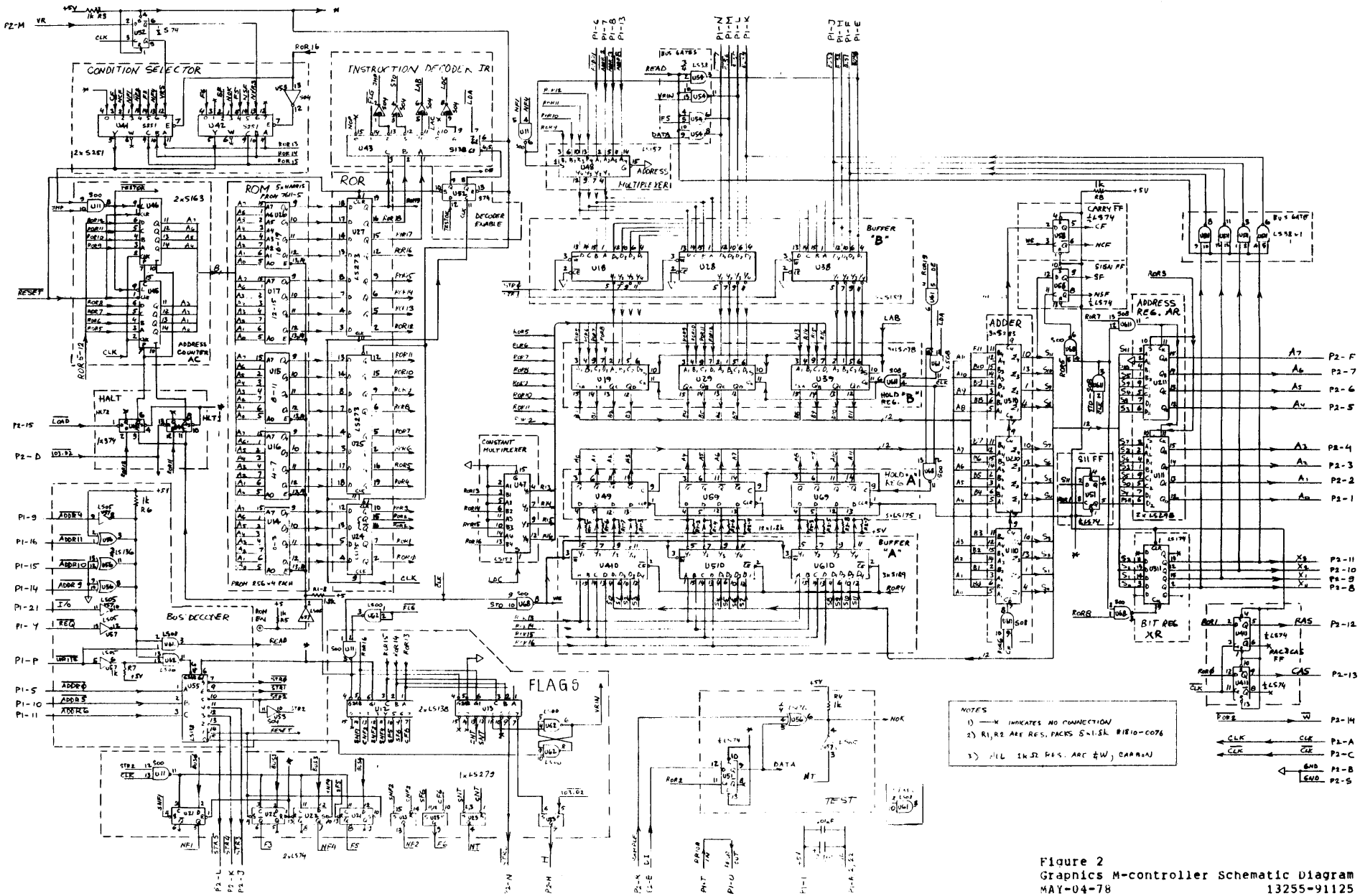

Figure 2
Graphics M-controller Schematic Diagram
MAY-04-78 13255-91125

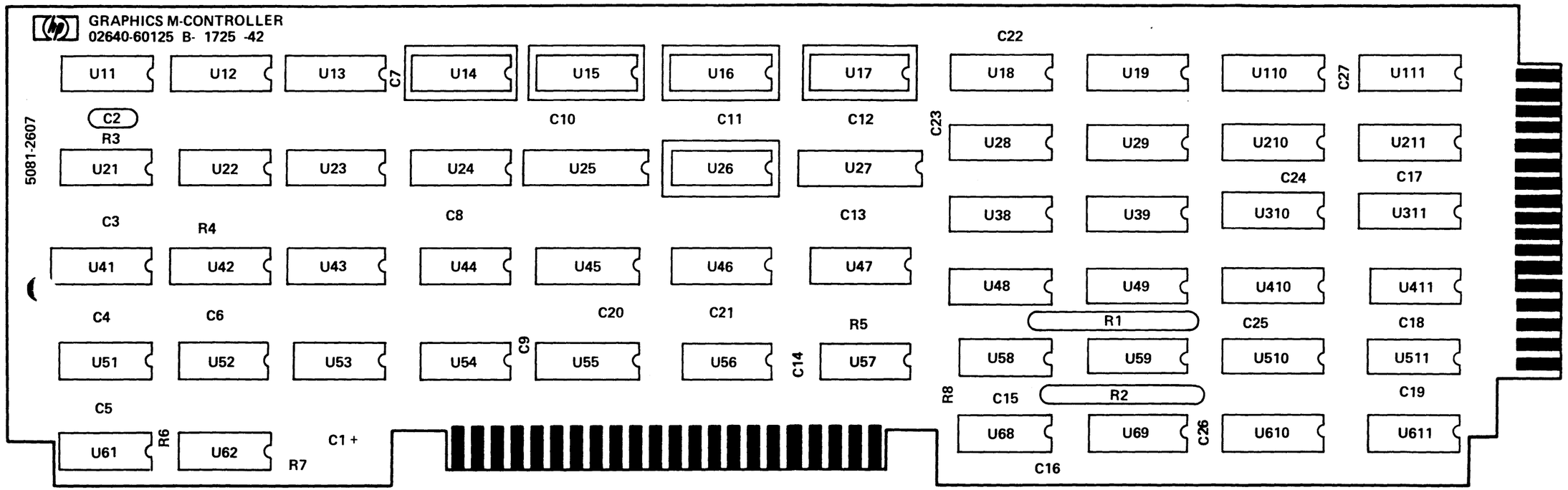

Figure 3
Graphics M-controller Component Location Diagram
MAY-04-78 13255-91125

INSTRUCTION FORMAT:

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAB	1	0	0	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0					Z	M	W	R	C
LAC	1	0	1	A_3	A_2	A_1	A_0	7	CONSTANT 1 6	5	4	3	2	1	0	0	M	W	R	C
LDA	1	1	1	A_3	A_2	A_1	A_0									0	M	W	R	C
LDC	1	1	0	11	10	9	8	7	CONSTANT 2 6	5	4	3	2	1	0	1	M	W	R	C
STO	0	1	1	A_3	A_2	A_1	A_0					X	L	C	S	0	M	W	R	C
JMP	0	1	0	COND. 3	2	1	0	7	TARGET 6	5	4	3	2	1	0	0	M	W	R	C
FLG	0	0	1	F_3	F_2	F_1	F_0	H_2	H_1							0	M	W	R	C
NOP	0	0	0													0	M	W	R	C

SHADOWED BITS ARE DON'T CARE

Figure 4
Instruction Format
MAY-04-78
13255-91125

IMAGE MEMORY ADDRESS BIT ARRANGEMENT

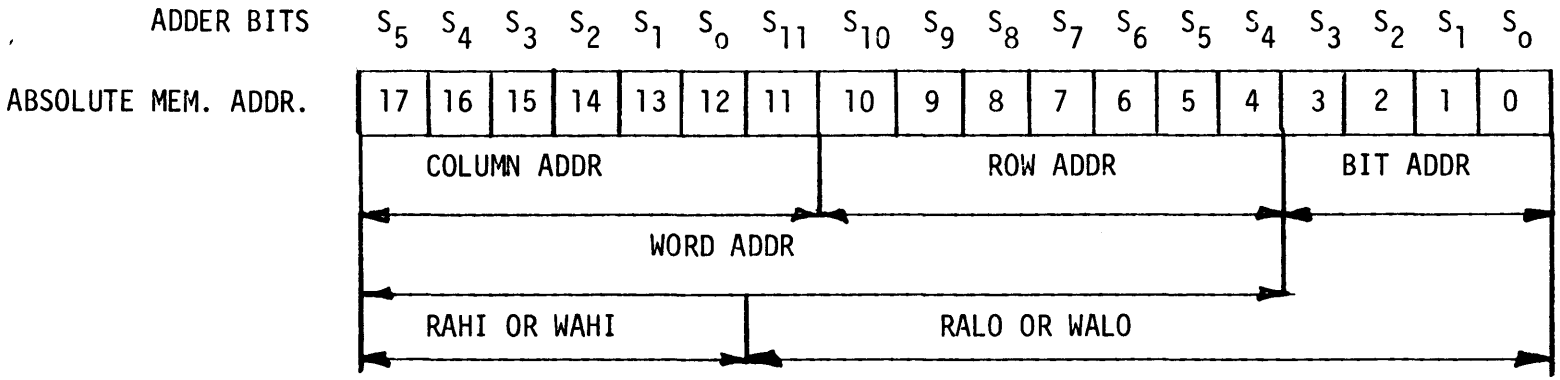

Figure 5
Image Memory Address
MAY-04-78
13255-91125

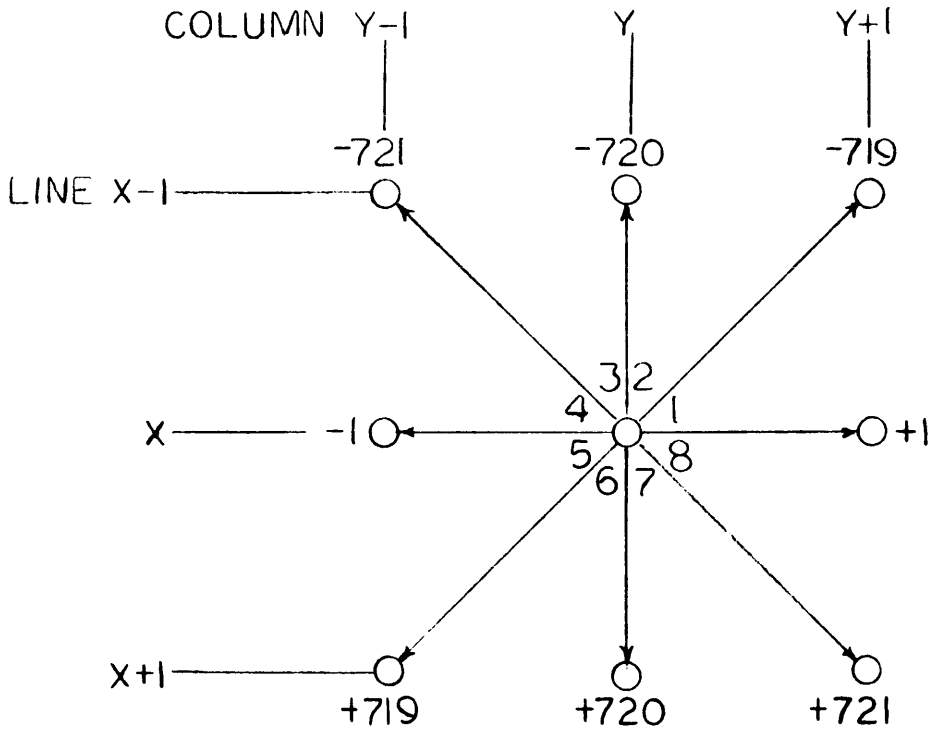

IMAGE MEMORY BIT DISPLACEMENTS.

Figure 6
Image Memory Bit Displacement
MAY-04-78 13255-91125

ZOOM EXAMPLE
MAGNIFICATION M=4

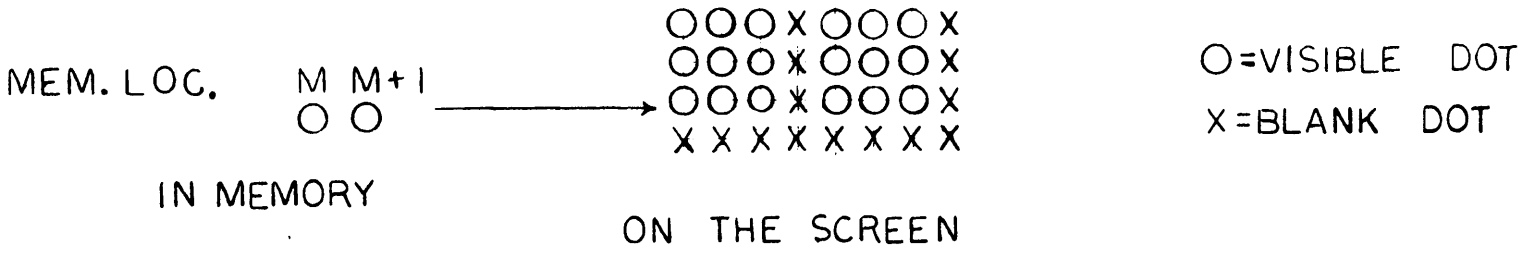

Figure 7 ZOOM Example
MAY-04-787 13255-91125

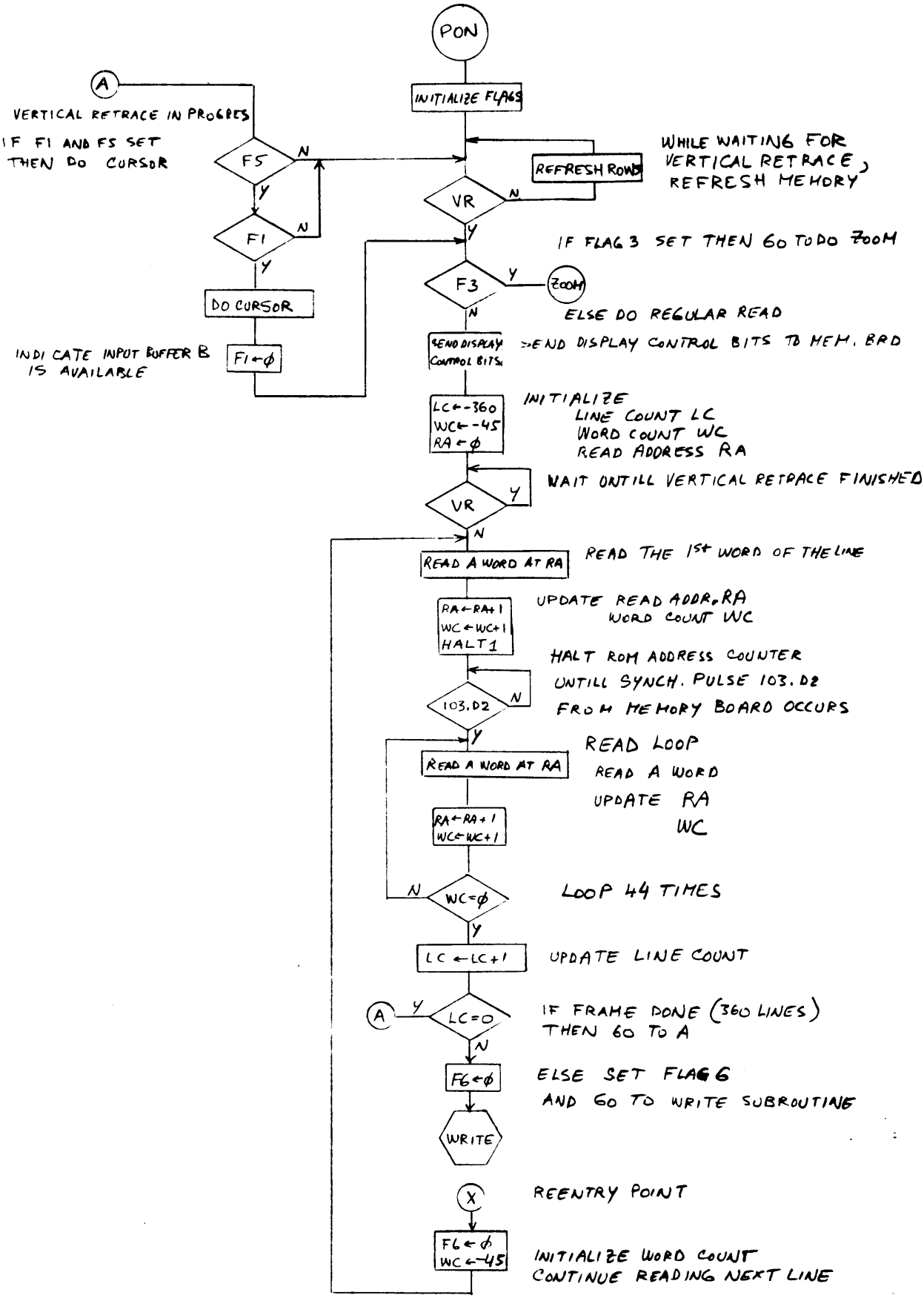

Figure 8 Flowchart A
MAY-04-78 13255-91125

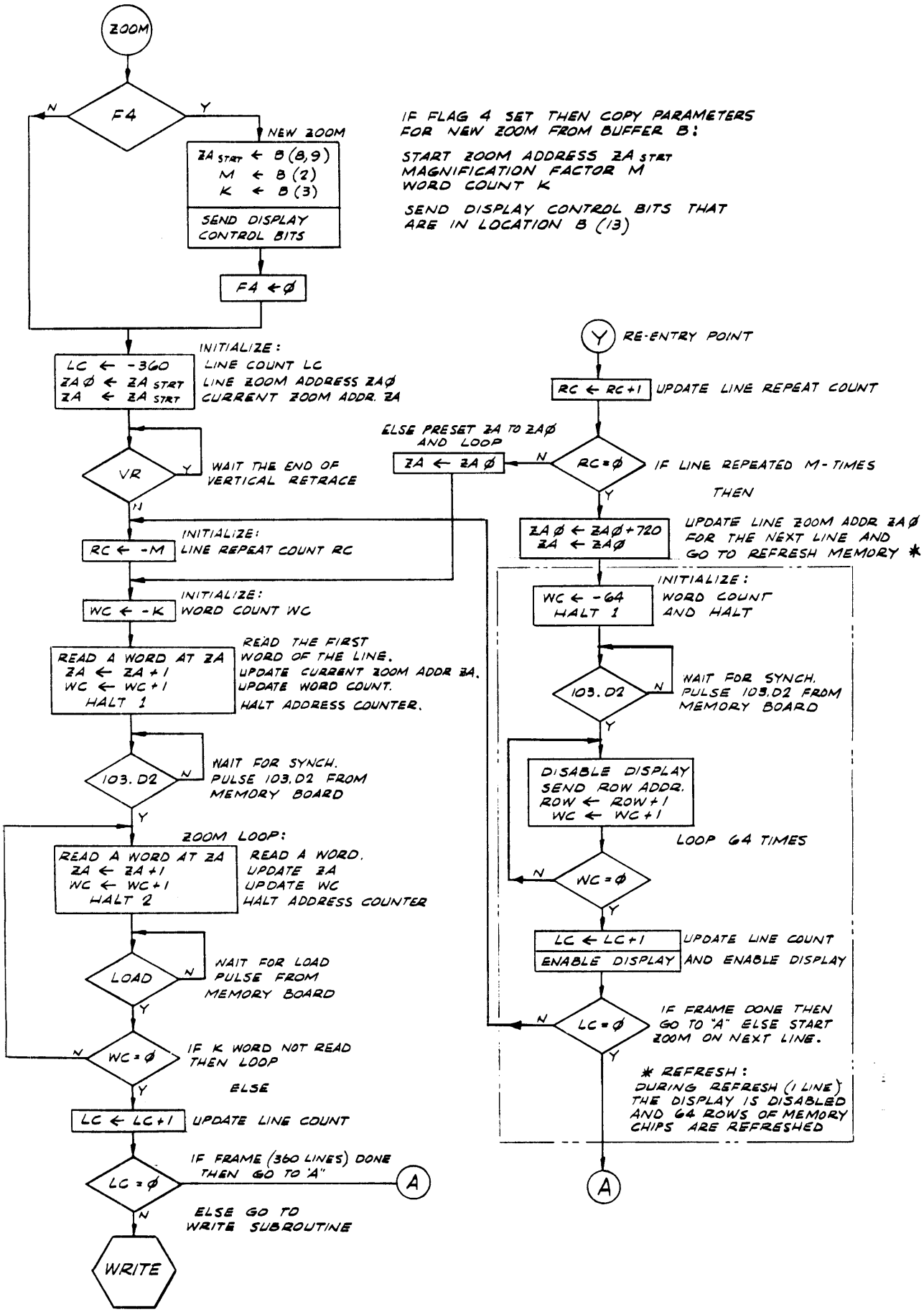

Figure 9 Flowchart B
MAY-04-78 13255-91125

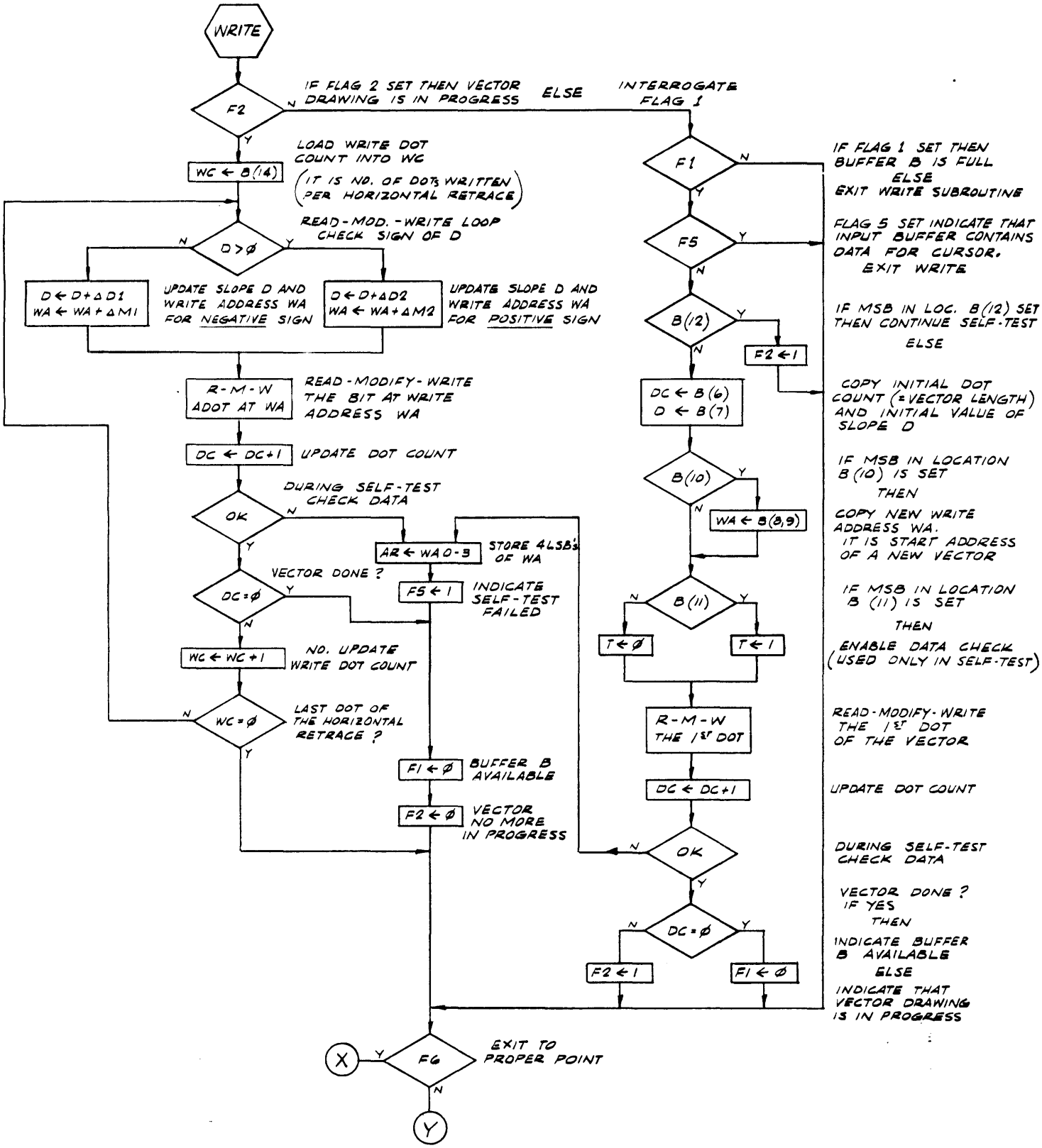

Figure 10
MAY-04-78

Flowchart C
13255-91125

Replaceable Parts

Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
	02640-60125	1	ASSEMBLY GRAPHICS M-CONTROLLER		
			DATE CODE: B-1725-42		
C1	0180-0393	1	CAPACITOR-39UF 10V		
C2 thru C19	0160-2055	18	CAPACITOR-.01UF		
C20 thru C27	0150-0121	8	CAPACITOR-0.1UF		
E1 thru E3	0360-0124	3	STUD SOLDER TERM		
R1 thru R2	1810-0076	2	NETWORK-RES SIP		
R3 thru R8	0683-1025	6	RESISTOR 1000 5% .25		
U14	1816-1121	1	IC MEMORY		
U15	1816-1123	1	IC MEMORY		
U16	1816-1122	1	IC MEMORY		
U17	1816-1124	1	IC MEMORY		
U18	1816-0724	1	IC SN74S189N		
U19	1820-1444	1	IC SN74LS298N		
U21	1820-1112	1	IC SN74LS74N		
U22	1820-1112	1	IC SN74LS74N		
U23	1820-1440	1	IC SN74LS279N		
U24	1820-1195	1	IC SN74LS175N		
U25	1820-1730	1	IC SN74LS273N		
U26	1816-1125	1	IC MEMORY		
U27	1820-1730	1	IC SN74LS273N		
U28	1816-0724	1	IC SN74S189N		
U29	1820-1444	1	IC SN74LS298N		
U30					
U38	1816-0724	1	IC SN74S189N		
U39	1820-1444	1	IC SN74LS298N		
U41	1820-1302	1	IC SN74S251N		
U42	1820-1302	1	IC SN74S251N		
U43	1820-1240	1	IC SN74S138		
U44	1820-0693	1	IC SN74S74N		
U45	1820-1453	1	IC SN74S163N		
U46	1820-1453	1	IC SN74S163N		
U47	1820-1470	1	IC SN74LS157N		
U48	1820-1470	1	IC SN74LS157N		
U49	1820-1195	1	IC SN74LS175N		
U51	1820-1112	1	IC SN74LS74N		
U52	1820-0693	1	IC SN74S74N		
U53	1820-0683	1	IC SN74S04N		
U54	1820-1209	1	IC SN74LS38N		
U55	1820-1216	1	IC SN74LS138N		
U56	1820-1215	1	IC SN74LS136N		
U57	1820-1200	1	IC SN74LS05N		
U58	1820-1112	1	IC SN74LS74N		
U59	1820-1195	1	IC SN74LS175N		
U61	1820-1201	1	IC SN74LS08N		
U62	1820-1197	1	IC SN74LS00N		
U68	1820-0681	1	IC SN74S00N		
U69	1820-1195	1	IC SN74LS175N		
U110	1820-1871	1	IC 74S283		
U111	1820-1444	1	IC SN74LS298N		
U210	1820-1871	1	IC 74S283		
U211	1820-1444	1	IC SN74LS298N		
U310	1820-1871	1	IC 74S283		
U311	1820-1196	1	IC SN74LS174N		
U410	1816-0724	1	IC SN74S189N		
U411	1820-1112	1	IC SN74LS74N		
U510	1816-0724	1	IC SN74S189N		
U511	1820-1209	1	IC SN74LS38N		
U610	1816-0724	1	IC SN74S189N		
U611	1820-1367	1	IC SN74S08N		
U11	1820-0681	1	IC SN74S00N		
U12	1820-1216	1	IC SN74LS138N		
U13	1820-1216	1	IC SN74LS138N		
XU14 thru XU17	1200-0607	4	SOCKET 16-DIP SLDR		
XU26	1200-0607	1	SOCKET 16-DIP SLDR		