RM-9000

Theory of Operation

Volume I

Original Text

503162-01

#67

DLR ASSOCIATES INC.
384 laurelwood road,
santa clara, calif. 95050

SIZE	PAPER WEIGHT	SHEET 1 OF 4		
8.5 x 11				
TYPE	PUNCH	REV DATE	DATE	NO COPIES
				250
BINDING INSTRUCTIONS		MANUAL		
GBC		RM-9000 Theory of Operation, Vol. I		

503162-00

1 4-3 = 2 sides
21 = 1 side
307 original

Negs FURN. 11 x 17 = 13
8.5 x 11 = 1

Title page	Notice	i	ii	iii	iv
v	vi	vii	viii	ix	x
1-1	1-2	1-3	1-4	11x17 Neg. Furn. 1-5	1-6
1-7	11x17 Neg. FURN. 1-8	1-9	1-10	1-11	1-12
1-13	1-14	1-15	1-16	1-17	1-18
1-19	1-20	11x17 Neg. FURN. 1-20A	X	11x17 Neg. FURN. 1-20B	X
1-21	1-22	2-1	X	11x17 Neg. FURN. 2-2	X
11x17 Neg. FURN. 2-3	X	2-4 → 2-21		11x17 Neg. FURN. 2-22	X

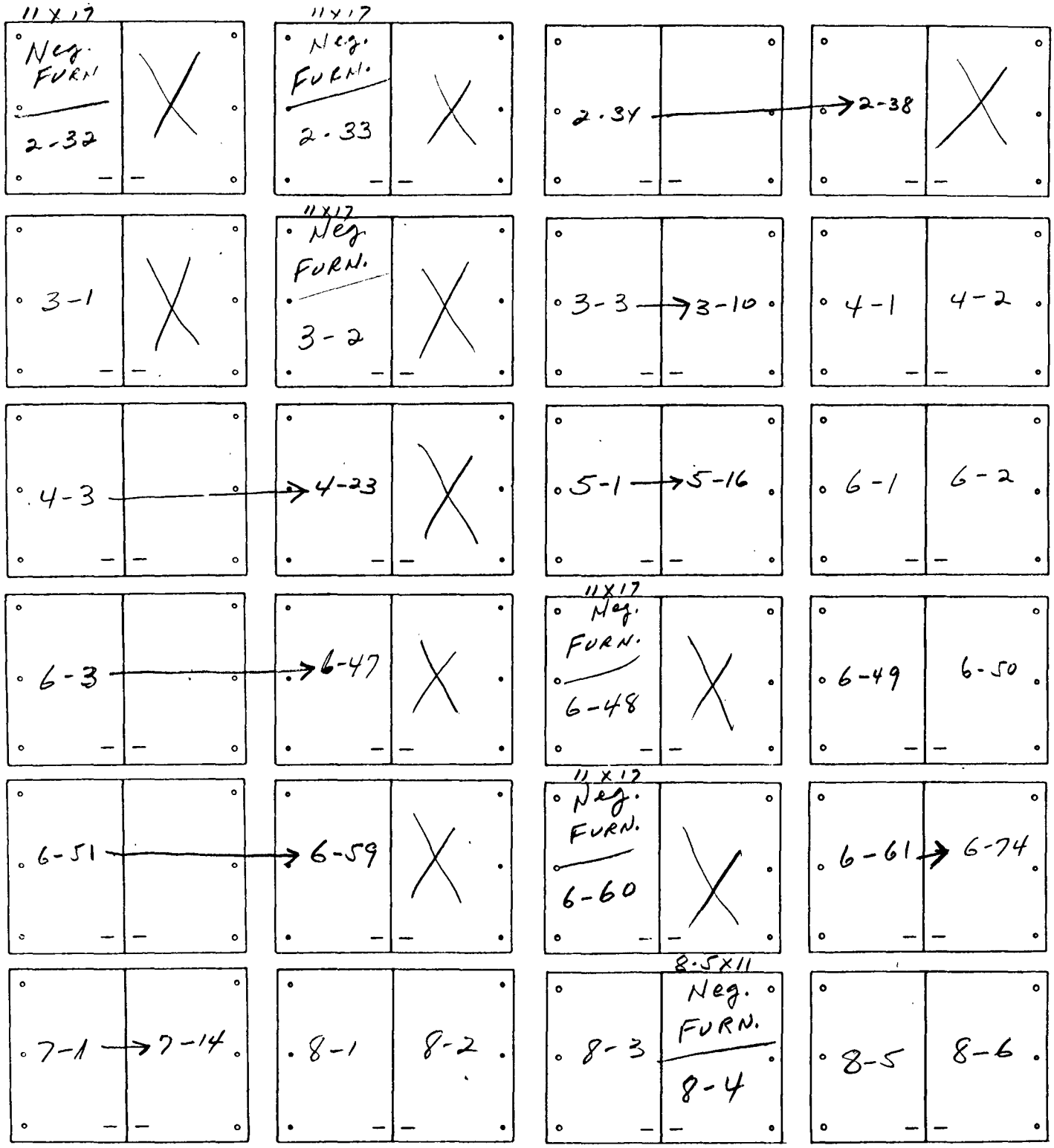
DLR ASSOCIATES INC.
384 laurelwood road,
santa clara, calif. 95050

SIZE	PAPER WEIGHT	SHEET 2 OF 4		
TYPE	PUNCH	REV DATE	DATE	NO COPIES
BINDING INSTRUCTIONS		MANUAL		

2-23	2-24	2-25	2-26	2-27	2-28	2-29	2-30
2-31	X						

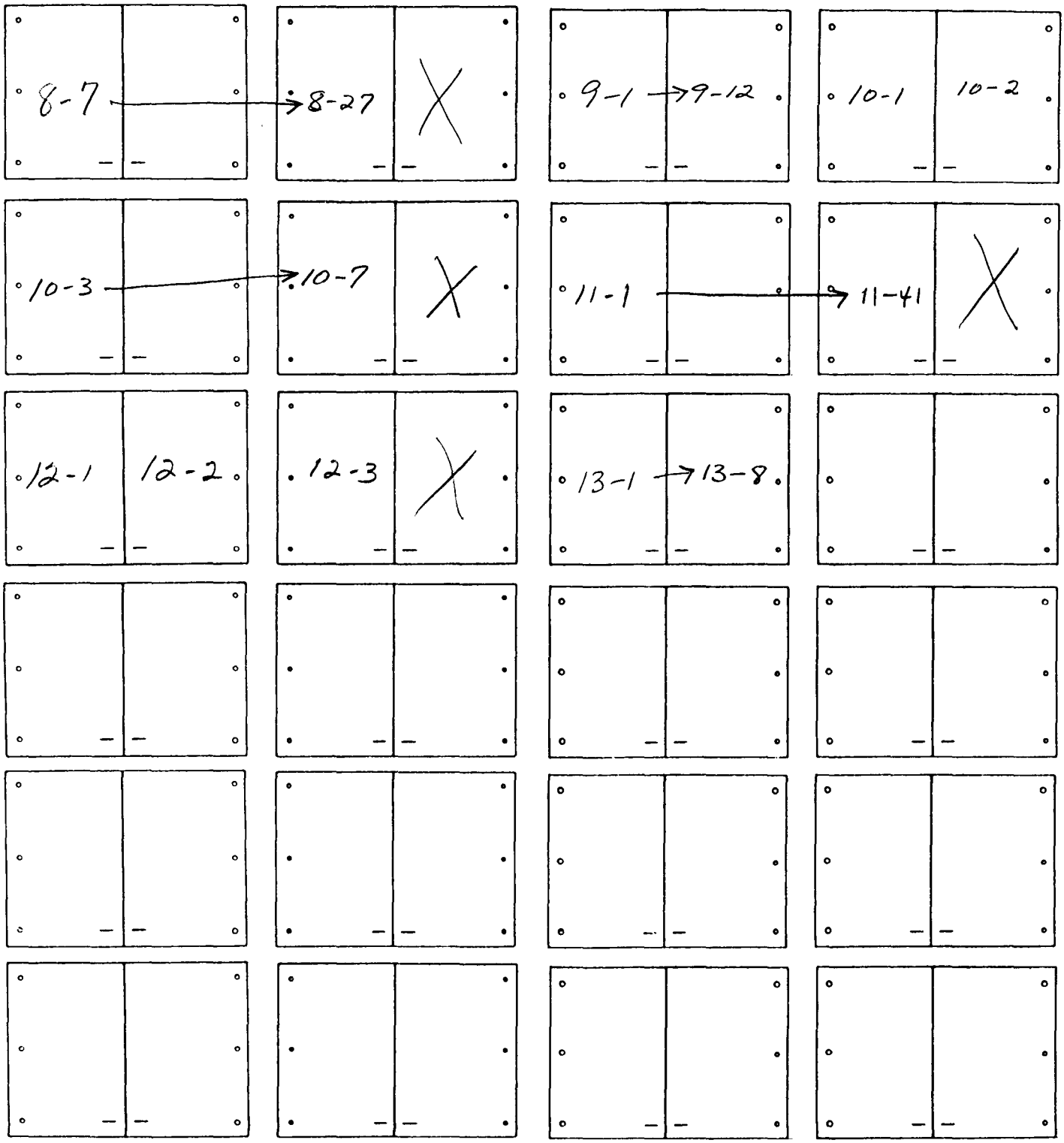
DLR ASSOCIATES INC.
384 laurelwood road,
santa clara, calif. 95050

SIZE	PAPER WEIGHT	SHEET 3 OF 4		
TYPE	PUNCH	REV DATE	DATE	NO COPIES
BINDING INSTRUCTIONS		MANUAL RM-9000, Vol. I		



DLR ASSOCIATES INC.
384 laurelwood road,
santa clara, calif. 95050

SIZE	PAPER WEIGHT	SHEET 4 OF 4		
TYPE	PUNCH	REV DATE	DATE	NO COPIES
BINDING INSTRUCTIONS		MANUAL RM-9000, Vol. I		



307 orig - 294 originals
143 sheets
13 11x17

140 8½x11 sheets
13 11x17 sheets

Plates:
293 x .75 =
13 x 2.00 =

Paper: 130 8½x11 x250 ÷500 x4.00 +20% =
13 x250 ÷500 x8.00 +10% =

Press time:
294 ÷12 x25.00 =
~~÷10 x 25.00~~ =
13 ÷6 x 25 = 50.00

Binding 4.00

Collated:
130 x250 x 3/1000 =

GBC 250 x.75

Revised from C to D

Part No. 5031620

Ramtek

RM-9000 Theory of Operation

Volume I

Part No. 503162-00D

Ramtek Corporation
2211 Lawson Lane
Santa Clara, California 95050

Ramtek

NOTICE

Ramtek Corporation has prepared this manual for use by Ramtek personnel, licensees and customers. The information contained herein is the property of Ramtek and shall neither be reproduced in whole nor in part without Ramtek's prior written approval.

Ramtek reserves the right to make changes without notice in the specifications and materials contained herein, and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including, but not limited to, typographical, arithmetic, or listing errors.

TABLE OF CONTENTS

		PAGE
1.0	INTRODUCTION TO THE 9000 SERIES	1-1
1.1	9000 Series System Architecture	1-3
1.2	Functional Specifications	1-7
1.3	Programming Specifications	1-14
1.4	Physical Specifications	1-20
1.5	Environmental Specifications	1-20
1.6	Power Requirements	1-22
2.0	INSTALLATION, MAINTENANCE AND OPERATION	2-1
2.1	Initial Inspection	2-1
2.2	Rack Mounting/Board Assignments	2-1
2.3	Cabling	2-7
2.4	Connector Reference Information	2-7
2.5	Computer I/O Cable	2-10
2.6	Video and Serial Link Cable	2-14
2.7	Distribution Panels	2-21
2.8	Power Adjustments and Modifications	2-31
2.9	Controls and Indicators	2-34
2.10	Acceptance Test Host-CPU Programs	2-36
2.11	Diagnostic Aids	2-36
2.12	Repacking for Shipment	2-37
3.0	THEORY OF OPERATION	3-1
3.1	General Sequencer Theory	3-5
4.0	INTERFACE BOARD AND INTERNAL DATA EXCHANGE BUS	4-1
4.1	General Purpose Simplified Block Diagram	4-2
4.2	Transfer Timing	4-6
4.3	GPIF Command and Status	4-7
4.4	Detailed Description - General Purpose Interface	4-11
4.5	Detailed Description - Internal Data Exchange Bus	4-16
5.0	SYNC TIMING BOARD	5-1
5.1	Functional Description	5-1
5.1.1	Waveform Timing Categories	5-1
5.1.2	Monitor Resolution Considerations	5-3
5.1.3	Monitor Format Timing Considerations	5-5
5.2	Theory of Operation	5-8
5.2.1	Block Diagram Drawing 502334 Sheet 1	5-8
5.2.2	Detailed Circuit Description	5-9

TABLE OF CONTENTS (Cont'd)

		PAGE
6.0	CONTROL BOARD	
6.1	Introduction	6-1
6.2	Display Generator and Other 8080 I/O Registers	6-5
6.2.1	Subchannel Mask Register (Addresses = 8000, 8001 or 8080, 8081)	6-5
6.2.2	Interface Interrupt Register (Addresses = 8002, 8003)	6-6
6.2.3	Readback Element Register (Addresses = 8004, 8005 or 8084, 8085)	6-6
6.2.4	Element Origin Address Register (Addresses = 8006, 8007)	6-7
6.2.5	Line Origin Address Register (Addresses = 8008, 8009)	6-8
6.2.6	DMA Source Register (Addresses = 800A, 800B)	6-8
6.2.7	DMA Destination Register (Addresses = 800C, 800D)	6-9
6.2.8	COP Control Register (Addresses = 800E, 800F or 808E, 808F)	6-10
6.2.9	Background Element Register (Addresses 8010, 8011 or 8090, 8091)	6-11
6.2.10	Foreground Element Register (Addresses 8012, 8013 or 8092, 8093)	6-11
6.2.11	Element Current Operating Point Address Register (Addresses = 8014, 8015 or 8094, 8095)	6-12
6.2.12	Line Current Operating Point Address Register (Addresses = 8016, 8017 or 8096, 8097)	6-13
6.2.13	Font Data Register (Addresses = 8018, 8019)	6-14
6.2.14	Host Data Register (Addresses = 801A, 801B)	6-15
6.2.15	System Status Register (Addresses = 801C, 801D)	6-16
6.2.16	DMA Word Count Register (Addresses = 801E, 801F)	6-17
6.2.17	DMA Address Register (Addresses = 8010, 8021)	6-17
6.2.18	Move COP and Write Register (Addresses = 8022, 8023 or 80A2, 80A3)	6-17
6.2.19	Move COP Register (Addresses = 8024, 8025 or 804A, 80A5)	6-18
6.2.20	Options Flag DIP Switch Register (Addresses = 8026, 8027)	6-19
6.3	Control Board Logic - Block Diagram - Sheet 1	6-19
6.4	Definition of the Internal Processor Bus	6-26
6.4.1	Internal Processor Bus Address Lines	6-26
6.4.2	Internal Processor Bus Data Lines	6-27
6.4.3	Internal Processor Bus Write Line	6-27
6.4.4	Internal Processor Bus Read Line	6-27
6.4.5	The Internal Processor Bus Ready Line	6-27

		PAGE
6.4.6	Internal Processor Bus Acknowledge Line	6-28
6.4.7	Internal Processor Bus Sync Line	6-31
6.4.8	Internal Processor Bus Hold Request Line	6-31
6.4.9	Internal Processor Bus Hold Acknowledge Line	6-31
6.4.10	Internal Processor Bus DMA Go Line	6-32
6.5	8080A-1 CPU and Associated Handshake Circuitry and Address-Data Driver-Receiver Network - Sheet 2	6-32
6.6	8080 Memory - Sheet 3	6-34
6.7	I/O Device Register Decode and DMA Source/Destination Registers - Sheet 4	6-35
6.8	Control Board Internal Data Bus/Foreground Register/ Background Register/Readback Register/Memory Plane Select Register/Refresh Memory Data Bus Drivers & Receivers - Sheet 5	6-38
6.9	Internal Processor Bus to Internal Data Exchange Bus Data Exchange Logic/Interrupt Register/System & 8080 Clocks/Acknowledge Handshake Control - Sheet 6	6-40
6.10	DIP Switch/Self Test LED/Ready Circuitry/CPU Clear Circuitry - Sheet 7	6-43
6.11	DMA Word Counter/Handshake Sequencer - Sheet 8	6-45
6.12	Display Generator Sequencer/Raster Data Register - Sheet 9	6-53
6.13	Refresh Memory Timing Generator - Sheet 10	6-59
6.14	Control Register/Update Equation Select/X-COP Register/ Y-COP Register/Refresh Memory Address Multiplexing - Sheet 11	6-67
6.15	X, Y Coordinate Multiply by Two/X Coord Address Map/Y Origin Register/X Origin Register/X Screen Refresh Address Counter/Y Screen Refresh Address Counter - Sheet 12	6-69
7.0	VIDEO TYPE I INTRODUCTION	7-1
7.1	Functional Description (Reference Drawing 502445)	7-1
7.1.1	Element and Field Selection (Sheets 3 and 4)	7-1
7.1.2	Overlays, Cursors & Blink Mixed with Data (Sheets 3 and 4)	7-3
7.1.3	The Inputs to the Digital to Analog Converters (Sheets 5, 6 and 7)	7-6
7.1.4	Cursor 10% Brightness Patch	7-9
7.2	Video Type I Logic Description	7-10
7.2.1	Video Type I Block Diagram	7-10
7.2.2	Timing (Sheet 2)	7-10
7.2.3	Sync, Blanking and Drive Signals (Sheet 2)	7-11
7.2.4	Generation of Element and Field Selection Signals (Sheet 2)	7-11
7.2.5	Input Reclock Registers (Sheets 3 and 4)	7-11

TABLE OF CONTENTS (Cont'd)

			PAGE
	7.2.6	Element and Field Selection (Sheets 3 and 4)	7-12
	7.2.7	Blanking Mixed with Data (Sheets 3 and 4)	7-12
	7.2.8	Overlays, Cursors and Blink Mixed with Data (Sheets 3 and 4)	7-12
	7.2.9	4 Bit DAC Outputs (Sheets 5, 6 and 7)	7-12
	7.2.10	"Straight Through" Video Outputs (Sheets 3 and 4)	7-13
	7.2.11	The Video Amplifiers (Sheets 8 and 9)	7-13
	7.2.12	Cursors (Sheet 10)	7-13
8.0	VIDEO 2A		8-1
	8.1	Functional Description Reference Drawing 503046	8-1
	8.1.1	Element and Field Selection (Sheets 2 and 3)	8-1
	8.1.2	LUT Address Selector (Sheets 2 and 3)	8-3
	8.1.3	Select Data for 8 Bit Digital to Analog Converter (Sheet 4)	8-6
	8.1.4	Mixing Cursors and Overlays with Data (Sheet 4)	8-8
	8.1.5	Inputs to the Digital to Analog Converters (Sheets 5, 6, 7, 8 and 9)	8-10
	8.1.6	Address and Data Bus Patching (Sheet 11)	8-12
	8.1.7	Instruction Formats	8-13
		8.1.7.1 LUT Address Counter (Sheet 11)	8-13
		8.1.7.2 Read/Write the LUT (PIØ)	8-14
		8.1.7.3 DMA Source Register	8-15
		8.1.7.4 DMA Destination Register	8-15
	8.2	Video 2A Logic Description	8-16
	8.2.1	Input Reclock Registers	8-16
	8.2.2	Element and Field Selection	8-16
	8.2.3	LUT Address Selector	8-16
	8.2.4	Lookup Table	8-17
	8.2.5	Cursors	8-17
	8.2.6	Select Data Before or After LUT	8-17
	8.2.7	Data Mixed with Overlays, Cursors and Blanking	8-17
	8.2.8	Color Outputs	8-17
	8.2.9	Grey Scale Output	8-18
	8.2.10	System Timing	8-18
	8.2.11	Sync Blanking and Drive Signals	8-18
	8.2.12	Data Receiver/Drivers, Address Counter, Address Receivers and Instruction Decoders	8-19
	8.2.13	System Reset	8-19
	8.2.14	Gating the Address Counter to the LUT	8-20
	8.2.15	DMA Control	8-20
	8.2.16	Read LUT Control (PIØ)	8-20

TABLE OF CONTENTS (Cont'd)

		PAGE
	8.2.17 Write LUT PIO	8-21
	8.2.18 Load the Source of Destination Flags	8-21
	8.2.19 Parallel Load of Address Counter	8-21
	8.2.20 DMA Transfer of Data to the LUT	8-21
	8.2.21 DMA Transfer of Data from the LUT	8-22
8.3	Turning the Video 2A 8 Bit Digital to Analog Converter (DAC) A Rev. and Above.	8-24
9.0	256 x 320 MEMORY BOARD	9-1
9.1	256 x 320 Memory Board Refer to Block Diagram-- Drawing 502331	9-2
9.2	Memory Chip Select Logic	9-6
9.3	Memory Data and Write Enable Select Logic	9-8
9.4	Memory Sections	9-10
10.0	MEMORY EXPANSION -- OPTIONAL	10-1
10.1	Control Logic -- Sheet 2	10-3
10.2	Address Logic -- Sheet 3	10-5
11.0	SERIAL LINK BOARD -- OPTIONAL	11-1
11.1	Activity Status Register	11-2
11.2	Peripheral Status Register	11-5
11.3	Receiver Registers	11-6
11.4	Transmit Registers	11-7
11.5	Cursor Element (X) Registers	11-8
11.6	Cursor Line (Y) Registers	11-9
11.7	Cursor Size Description	11-10
11.8	Operational Description - Hardware	11-11
	11.8.1 Serial Link Card/Address Set Selects	11-11
	11.8.2 Serial Link Input/Output Channels	11-11
	11.8.3 Baud Rate Selects	11-15
	11.8.4 Word Format Selects	11-17
	11.8.5 Line Driver/Receiver Selects	11-18
	11.8.6 Cursor Generators	11-21
11.9	Circuit Description	11-23
11.10	Block Diagram - Sheet 1	11-23
11.11	Detailed Circuit Description	11-26
	11.11.1 Cursor Line and Element Registers	11-26
	11.11.2 Cursor Video Generation	11-29
	11.11.3 Cursor Video Element Generation	11-29
	11.11.4 Cursor Video Line Generation	11-32
	11.11.5 Blink Clock and Blink Decode	11-35

		PAGE
11.11.6	Cursor Video	11-35
11.11.7	Processor Register Address Decode	11-35
11.11.8	Transmit/Receiver Registers	11-36
11.11.9	Cursor Readback Multiplexer	11-37
11.11.10	Sequence Controller	11-38
12.0	REPLACEABLE PARTS	12-1
12.1	Ordering Information	12-1
12.2	Recommended Spares	12-1
13.0	CIRCUIT DIAGRAMS	13-1

LIST OF FIGURES

NUMBER	TITLE	PAGE
1-1	RAMTEK 9000 SERIES FUNCTIONAL BLOCK DIAGRAM	1-5
1-2	9100, 9200 AND 9300 INSTRUCTION FORMATS	1-18
1-3	9100, 9200 CHASSIS BOARD ASSIGNMENTS	1-21
1-4	9300 CHASSIS BOARD ASSIGNMENTS	1-21
2-1	9100 AND 9200 REAR VIEW	2-4
2-2	9300 REAR VIEW	2-5
2-3	9100, 9200 CHASSIS BOARD ASSIGNMENTS	2-6
2-4	9300 CHASSIS BOARD ASSIGNMENTS	2-6
2-5	104 PIN CONNECTOR	2-8
2-6	SMALL DISTRIBUTION PANEL, TYPE I	2-22
2-7	LARGE DISTRIBUTION PANEL	2-22
2-8	POWER DISTRIBUTION 9100 - TRIO LABS PS	2-32
2-9	POWER DISTRIBUTION 9300	2-33
3-1	9000 SERIES BLOCK DIAGRAM	3-2
3-2	GENERAL SEQUENCER CONFIGURATION	3-6
3-3	TYPICAL SEQUENCER CONFIGURATION ON THE CONTROL BOARD	3-8
4-1	GPIF WITHIN A COMPUTER/9000 SYSTEM	4-1
4-2	GPIF SIMPLIFIED BLOCK DIAGRAM	4-3
4-3	I/O WRITE TIMING	4-5
4-4	I/O READ TIMING	4-5
4-5	TRANSFER TIMING	4-6
4-6	IDE BLOCK DIAGRAM	4-17
4-7	WRITE TIMING	4-21
4-8	READ TIMING	4-21
4-9	INTERRUPT TIMING	4-22
4-10	INTERRUPT NETWORK	4-23
6-1	INTERNAL PROCESSOR BUS WRITE TIMING	6-29
6-2	INTERNAL PROCESSOR BUS READ TIMING	6-30
6-3	HANDSHAKE SEQUENCER FLOW CHART	6-48
6-4	DISPLAY GENERATOR SEQUENCER FLOW CHART	6-60
7-1	SIMPLIFIED BLOCK DIAGRAM OF THE VIDEO I	7-2
7-2	OVERLAY, CURSOR AND BLINK PATCHING NETWORK	7-4

LIST OF FIGURES (Cont'd)

NUMBER	TITLE	PAGE
7-3	DAC PATCHING NETWORK	7-7
7-4	CURSOR 10% BRIGHTNESS PATCHING NETWORK	7-9
8-1	SIMPLIFIED BLOCK OF VIDEO 2A	8-2
8-2	LUT PARTITIONING PATCHING NETWORK	8-4
8-3	8 BIT DAC DATA SELECTION PATCHING NETWORK	8-6
8-4	OVERLAY SUBCHANNEL ALTERNATE SELECTION PATCHING	8-9
8-5	DAC CONVERSION PATCHING NETWORK	8-10
8-6	LUT ADDRESS PATCHING NETWORK	8-12
8-7	ELEMENT PULSE	8-24
8-8	VIDEO UP AND DOWN RAMPS	8-26
9-1	REFRESH MEMORY ADDRESS	9-3
11-1	SERIAL LINK SHORT FORM REGISTER LIST	11-3
11-2	SERIAL LINK FUNDAMENTAL BLOCK DIAGRAM	11-4
11-3	CURSOR SIZE AND SHAPE	11-11
11-4	REGISTER ADDRESS SET SWITCH SETTINGS	11-14
11-5	DEVICE SELECT DIAGRAM	11-21
11-6	ELEMENT COUNTER COUNT PATTERN	11-31
11-7	LINE COUNTER COUNT PATTERN	11-34
11-8	SERIAL LINK CONTROLLER MICROPROGRAM	11-39
13-1	SCHEMATIC NOTES	13-2
13-2	TYPICAL DEVICE LOCATION DESIGNATOR	13-3

LIST OF TABLES

NUMBER	TITLE	PAGE
1-1A	9000 SERIES RESOLUTION/LINE RATE LISTING	1-2
1-1	9000 SERIES RESOLUTION/LINE RATE SPECIFICATIONS	1-8
1-2	COMPUTER INTERFACE	1-15
1-3	9100, 9200 AND 9300 INSTRUCTION SUMMARY	1-16
1-4	COMPLEX FORMAT INSTRUCTION ARGUMENTS	1-19
2-1	PARTS LIST, 104 PIN CONNECTOR	2-9
2-2	INTERFACE TO CPU CONNECTOR, J1	2-11
2-3	SERIAL LINK AND VIDEO CABLE	2-15
2-4	VIDEO/SERIAL LINK CONFIGURATIONS	2-18
2-5	CURSOR CONTROLLER/KEYBOARD PIN ASSIGNMENTS	2-19
2-6	SMALL DISTRIBUTION PANEL WIRING	2-23
2-7	LARGE DISTRIBUTION PANEL WIRING	2-25
4-1	COMMAND REGISTER BIT ASSIGNMENTS	4-8
4-2	STATUS REGISTER BIT ASSIGNMENTS	4-10
6-1	9000 SERIES ADDRESS DOMAIN	6-2
6-2	I/O ADDRESS ASSIGNMENT	6-3
6-3	I/O DEVICE REGISTER DECODE OUTPUT TERMS	6-37
6-4	HANDSHAKE SEQUENCER FIRMWARE	6-50
6-5	HANDSHAKE SEQUENCER CONDITION SELECT CODES	6-51
6-6	DISPLAY GENERATOR SEQUENCER CONDITION SELECT CODES	6-57
6-7	DISPLAY GENERATOR SEQUENCER FIRMWARE	6-61
6-8	CONTROL BOARD SIGNAL TERM PINOUT	6-74
7-1	SIGNALS AT 'E' DESIGNATORS USED IN FIGURE 7-2	7-5
7-2	SIGNALS AT E DESIGNATORS USED IN FIGURE 7-3	7-8
7-3	SIGNALS AT DESIGNATORS USED IN FIGURE 7-4	7-10
7-4	VIDEO BOARD TYPE I SIGNAL TERM PINOUT	7-14
8-1	SIGNALS AT E DESIGNATORS FOR FIGURE 8-2	8-5
8-2	THE SIGNALS AT E DESIGNATORS FOR FIGURE 8-3	8-7
8-3	OVERLAY PATCHING FOR SIGNALS AT E DESIGNATORS FOR FIGURE 8-2	8-9
8-4	SIGNALS AT E DESIGNATORS FOR FIGURE 8-5	8-11
8-5	THE LUT ADDRESS	8-12
8-6	SIGNALS AT E DESIGNATORS FOR FIGURE 8-6	8-13
8-7	VIDEO BOARD TYPE 2A SIGNAL TERM PINOUT	8-23

LIST OF TABLES (Cont'd)

NUMBER	TITLE	PAGE
9-1	BACKPLANE CONTROL SIGNAL WIRING	9-10
9-2	256 x 320 MEMORY BOARD SIGNAL TERM PINOUT	9-12
10-1	MEMORY EXPANSION BOARD SIGNAL TERM PINOUT	10-7
11-1	ADDRESS SET ASSIGNMENTS	11-12
11-2	CHASSIS SLOT ASSIGNMENTS	11-13
11-3	BAUD RATE GENERATOR SWITCH SELECT SETTINGS	11-16
11-4	WORD FORMAT SELECT SWITCH SETTINGS	11-19
11-5	SERIAL LINE RECEIVERS	11-20
11-6	SERIAL LINE DRIVERS	11-20
11-7	SERIAL LINK BOARD SIGNAL TERM PINOUT	11-41
12-1	DEPOT LEVEL REPAIR EQUIPMENT	12-1
12-2	SPARE COMPONENT KIT	12-2
13-1	NUMBERICAL LISTINGS OF IC TYPES	13-4

1.0 INTRODUCTION TO THE 9000 SERIES

Each of the 9000 Series is a classic refresh memory type graphic display system in which enough memory is provided in order to correspond to every visible element on the video monitor. Furthermore, in the 9000 Series more than one memory bit is utilized per element to achieve levels of intensity (grey scale level) or color. The 9000 Series is a family of low cost raster scan graphic display systems which are optimized for cost sensitive imaging and graphics applications. All of the models within the 9000 Series utilize the same Printed Circuit (PC) boards. The various models represent low, medium and high resolution displays which are configured by various interconnections of the basic 256 line by 320 element memory plane building blocks. Variations within each of the three basic models are achieved by changing monitor synchronization timing signals. The 9000 Series variations are summarized in Table 1-1A on the following page. Each of the 9000 Series may contain up to 12 memory planes and are optimized for image data throughput. They contain a high speed 8080 microprocessor which also provides a powerful means of implementing graphic, cartesian, raster, alphanumeric, rectilinear, vector and conic modes of display. The descriptions shown in Table 1-1A apply to all models of the 9000 Series.

NOTE

It is assumed that the reader has a thorough working knowledge of the 8080 microprocessor. For additional information about the 8080, the user may wish to consult the publication *INTEL 8080 Micro Computer Systems User Manual*.

TABLE 1-1A 9000 SERIES RESOLUTION/LINE RATE LISTING

MODEL	CONFIGURATION: VISIBLE LINES X ELEMENT	FORMAT KEYWORD	FRAME REFRESH RATE (Hz)	FIELD REFRESH RATE (Hz)	RECOMMENDED POWER LINE FREQUENCY (Hz)	TOTAL LINES PER FRAME
RM9100 LOW RESOLUTION	240x320	R BROADCAST	60	60	60	262
	240x320	I BROADCAST	30	60	60	525
	256x320	R STANDARD	60	60	60	279
	256x320	I STANDARD	30	60	60	559
	256x320	R EUROPEAN	50	50	50	312
	256x320	I EUROPEAN	25	50	50	625
	256x320	R STANDARD	50	50	50	279
	256x320	I STANDARD	25	50	50	559
RM9200 MEDIUM RESOLUTION	240x640	R BROADCAST	60	60	60	262
	240x640	I BROADCAST	30	60	60	525
	256x640	R STANDARD	60	60	60	279
	256x640	I STANDARD	30	60	60	559
	256x640	R EUROPEAN	50	50	50	312
	256x640	I EUROPEAN	25	50	50	625
	256x640	R STANDARD	50	50	50	279
	256x640	I STANDARD	25	50	50	559
RM9300 HIGH RESOLUTION	480x640	I BROADCAST	30	60	60	525
	512x640	I STANDARD	30	60	60	559
	512x640	I EUROPEAN	25	50	50	625
	512x640	I STANDARD	25	50	50	559

LEGEND: R = REPEAT FIELD.[3] FIELD "A" IS IDENTICAL TO FIELD "B".

I = 2:1 INTERLACED[2] SCAN.

NOTES:

1. IN COLUMNS WHERE NOTE 1 IS INDICATED,

   60 HZ IS ACTUALLY 59.94005 HZ,
   50 HZ IS ACTUALLY 50.00000 HZ,
   30 HZ IS ACTUALLY 29.97003 HZ,
   25 HZ is actually 25.00000 HZ.

2. IN AN INTERLACE FIELD SYSTEM ONE FRAME = TWO FIELDS.

3. IN A REPEAT FIELD SYSTEM ONE FRAME = ONE FIELD. THAT IS, ONE FRAME-TIME = A FIELD TIME

1.1 9000 SERIES SYSTEM ARCHITECTURE

The Ramtek 9000 Series Systems are microprocessor based systems which combine an optimum and an efficient compromise between hardware and software implementation of all graphic display system tasks. As a result, there are three distinct levels of programming and, thus, three levels of instruction sets:

1. User Language Graphic Display Software

2. 8080 Interpretive Software

3. Hardware Microprogram Firmware

The user graphic display language is described in the *9000 Programming Manual*. This instruction repertoire provides the user with a graphics system language of user oriented instructions for drawing pictures on the screen. This user language is based on 16 bit length words which are common in the minicomputer systems used to drive the display systems. Thus, the user language is "data" to the minicomputer and "graphic instructions" to the display system.

The second level of programming is the 8080 interpretive software (not firmware) which resides in the display system. It is written in 8080 assembly language and, thus, is based on 8 bit bytes. This interpreter defines the action to be taken when each user language 16 bit word is received by the display. Thus, it is possible to tailor the 9000 Series Systems to any other user language by writing a new 8080 interpreter. However, most users will wish to stay with the standard user language since the process of generating a new interpreter requires an in-depth understanding of the display logic which surrounds the 8080.

The third level of programming is not software, but firmware. Several sections of the 9000 Series control logic are microprogrammed control

sections which contain firmware to control such operations as I/O handshake, DMA, and memory control. Firmware is used rather than 8080 software to control functions which require speeds greater than achievable by an 8080. The firmware is highly restricted to the available hardware functions and as such would never be modified by user.

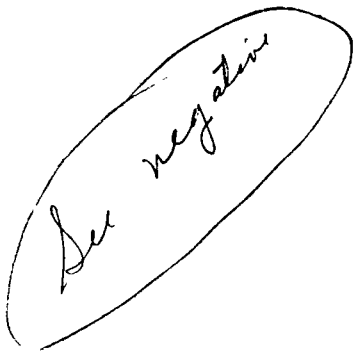
Figure 1-1 depicts a block diagram of a 9000 Series System. The 8080 microprocessor shown controls the internal processor (IP) bus and monitors the interface parallel link for the purpose of receiving the 16 bit user graphic display instructions from the host CPU. The host CPU communicates graphic display instructions through one of several Ramtek standard interface boards to the Internal Data Exchange (IDE) bus. These interface boards are identical in design as used for previous Ramtek products. When the 8080 microprocessor receives (16 bit) instructions from the host CPU, it will execute 8080 software which decodes the instructions and communicates the necessary data to the display generator which in turn causes RAM refresh memory planes to be loaded. Image data throughput is optimized by allowing direct transfer of parallel 16 bit data between the interface parallel link and the display generator. Up to 12 RAM refresh memory planes provide video data to the video generator. The video generator boards provide a flexible connection of the memory organization to many combinations of color and/or black and white CRT monitors. The number of independent CRT monitors is a function of the resolution and the color requirements of the CRT's to be driven.

The 9000 Series RAM memory planes use Metal Oxide Semiconductor (MOS) 4096 bit RAM for refreshing the monitor. The display generator I/O processor is only capable of writing data into the RAM refresh memories in image data mode. The 8080 microprocessor may receive data in many modes and convert it to image data mode before writing into the display generator. For instance, if the 8080 microprocessor receives alphanumeric data, it will consult its character generator PROM to equate the ASCII character code to font information which is driven to the display generator. Image data is communicated from the interface parallel link to the display generator in a high speed block transfer

Page 1-5

11 x 17

Artwork



1-5

fashion. The 12 least significant bits of each 16 bit image word are driven to the refresh memory planes under control of a memory plane select register contained in the display generator. Raster, cartesian and graphic mode data is always received by the 8080 microprocessor and converted to image mode data before being driven to the display generator. By expanding the 8080 microprocessor's PROM from 2K bytes upwards to 16K on the memory expansion board the 9000 Series Graphic Display System may contain complex graphic functions such as vectors and conics. It may also contain scaling software packages and arithmetic and logical capability. The 8080 microprocessor static RAM is utilized for the temporary storage of 8080 operands. The 8K byte dynamic RAM on the memory expansion board is utilized for storage of user defined graphic instructions, user defined programmable alphanumeric font and additional temporary storage for DMA transfers. The Memory Expansion board also contains DMA address counter for use in doing software scroll and scaling operations. It usually provides the address to the memory expansion RAM during DMA transfers between dynamic RAM and the display generator. Certain models of floppy disc controllers can be interfaced to the display system through the memory expansion board. Future expansion of the IP bus can allow the interconnection of specialized high speed graphic processors which optimize graphic instruction throughput.

A serial link board may also be connected to the IP bus in order to allow the user to connect keyboards and joystick cursor control mechanisms to the 9000 Series System. Each serial link board contains four Universal Asynchronous Receiver Transmitters (UART) which allow for interconnection of four keyboards or two keyboards and two cursor controllers. Two cursor generators drive cursor video to the video generator boards. As shown in Figure 1-1, the cursor generators are connected to the IP bus so that the 8080 can read or load cursor position and control bits independently of the cursor controllers.

The Voltage Controlled Oscillator (VCO) option shown in Figure 1-1 is available for certain of the interlaced variations of 9000 Series models. It allows the 9000 Series System to be synchronized or "locked" to another section of the user's system.

1.2 FUNCTIONAL SPECIFICATIONS

The following data functionally specify the Models 9100, 9200 and 9300 Graphic Display Systems:

REFRESH TECHNIQUE

The refresh technique employed is that of a classical raster scan display generator in that the generated image is stored in refresh memory which is continuously scanned at the television raster rate.

REFRESH MEMORY

The refresh memory stores up to 12 bits per picture element (pixel). Any combination of the 12 possible planes may be accessed and written simultaneously. The individual planes may be partitioned into independent display channels.

The refresh memory consists of 4K MOS dynamic RAMs. Each memory board stores up to six 9100 frames, three 9200 frames or three 9300 fields. Separate 9300 fields reside on separate memory boards. Access to any 12 bit cell of refresh memory is in 1.5 μs. Refresh memory access is interleaved such that write cycles do not interfere with refresh read cycles and thus screen refresh. Refresh memory is not parity checked or error corrected and as such should not be used for storing CPU software.

RESOLUTION AND LINE RATES

Six possible combinations of resolution are available. Table 1-1 lists the possible combinations and their associated television line rates. Note that a repeat field scan pattern (50 or 60 Hz refresh) is typically employed for the Models 9100 and 9200 Display Systems. The Model 9300 requires a 25 or 30 Hz refresh frequency and a 2:1 interlaced scan pattern.

DISPLAY PROCESSOR

The basic display processor consists of an 8080A-1 microprocessor with 2,048 bytes of program memory (PROM) and 512 bytes of scratch pad memory (RAM). Clock cycle time is approximately 326 ns.

MEMORY EXPANSION

The 8080 microprocessor program memory may be expanded to 16,384 bytes and its scratch pad memory to 8,704 bytes via the addition of a plug-in memory expansion board. This option also provides a microprocessor memory DMA address counter.

IMAGING CAPABILITY

The basic system processes and stores up to 12 bits of three dimensional, digitized image data. The image data is received in an industry compatible word-per-pixel and bit-per-plane format, and is normally written directly into the refresh memory at rates of up to 653,400 pixels per second. During the time that image data is being received into refresh memory, the microprocessor serves only to perform the raster margining function (start a new line when the previous line is completed). Bit plane selection and pixel-to-pixel addressing is achieved in hardware. The image data is written into a prescribed rectangular area (or page) of refresh memory. Any of eight possible scan sequences may be selected, for example, left-to-right/top-to-bottom. The scaling option provides for up/down scaling of image data. When downward scaling is specified, average pixel values are computed and stored versus storing each nth pixel of each nth line, where n represents the scaling ratio. When upward scaling is specified, each pixel is repeated n times on n lines. The Logical and Arithmetic Functions option (RM-LAF) provides a mechanism for correlating separate images. For example, two images might be compared by subtracting one from the other. This is achieved by merely specifying an appropriate logical/arithmetic function when writing the second image to the display, and is

implemented in 8080 software as a read-modify-write operation. As an example, the issued pixel might be subtracted from the corresponding stored pixel, and the difference would be stored versus the issued pixel itself. The various video generators provide for grey scale, color, pseudo color, automatic density slicing and contrast enhancement, and gamma correction of image data.

ALPHANUMERIC CAPABILITY

The basic system includes a character generator which decodes 64 USASCII character codes and generates the appropriate dot matrix symbols. The standard font size is five pixels by seven pixels within a seven pixel by nine pixel matrix. Row and column spacing is preset to seven pixels by nine pixels but is programmable. Sequential characters may be written from left-to-right, right-to-left, top-to-bottom or bottom-to-top, and may be rotated in 90 degree increments. Automatic margining is implemented such that a new line (or page) is started whenever a designated window boundry (margin) is crossed. A new line (or page) is also started whenever the USASCII carriage return symbol is decoded. A programmable font option provides for user definition (in real time) of up to 128 characters. For these characters, the matrix size is programmable up to a maximum size of eight pixels by twelve pixels. Characters may be up/down scaled by the scaling option.

GRAPHICS CAPABILITY

The optional graphics software draws end-point vectors, conics, plots and bar charts, and writes bit-per-element raster data into the refresh memory.

DYNAMIC CAPABILITY

Any rectangular area, even a single pixel, may be selectively erased and/or updated without affecting any other information in refresh memory. Graphic entities such as characters or vectors

may be erased by inverting the background and writing mode control bits and repeating the original command list.

READBACK CAPABILITY

The host computer may retrieve stored images from the refresh memory. Except for the direction of the data being transferred, the process duplicates that used for writing image data into the memory, except that planes are not maskable. Image data is retrieved from a specified rectangular area (or page) of refresh memory in a word-per-pixel and bit-per-plane format.

SCROLL CAPABILITY

There are two forms of scroll available. The first is a standard feature and allows the programmer to designate a coordinate value as the logical origin of the television raster. The image thus moves up or down and left or right simultaneously, within a 16.7 ms field time. The scroll is nondestructive in that data moving from the screen scrolls to the opposite edge of the screen. The second form of scroll is implemented as an 8080 software option and provides both windowing and partitioning. That is, only data within a prescribed rectangular area (or page) is scrolled, and only a prescribed combination of refresh memory planes are affected. Thus, multiple channels are supported because an associated set of memories can be scrolled without affecting the memory planes associated with other channels. This feature also provides for scrolling of images such as trend data behind a fixed grid overlay. This form of scroll is destructive in that data scrolling from the window is lost while the deserted area of the window is filled with the background color. Scroll is up, down, left or right and by a prescribed number of pixels. Scroll speed varies depending upon window size.

INTERACTIVE CAPABILITY

Up to eight keyboards or four keyboards and four joysticks can be interfaced to a 9000 Series System. Each serial link board can accept up to four keyboards or two keyboards and two joysticks. The baud rates for keyboards and joysticks are independently adjustable to any of the standard rates between 110 and 9600 baud.

RS-170 COMPATIBILITY

All video outputs conform to EIA Standard RS-170. All outputs are composite video.

B/W CAPABILITY

Each display controller may generate up to 12 independent B/W channels.

GREY SCALE CAPABILITY

Each display controller may generate up to 256 grey levels, or multiple independent channels of a lesser number of grey levels such as three independent channels producing 16 grey levels each.

COLOR CAPABILITY

Each display controller may generate up to 4,096 colors, or multiple independent channels having seven colors each.

CURSOR CAPABILITY

Each display controller is capable of generating up to four non-destructive, crosshair cursors which may be steered via a cursor controller (joystick or trackball) without host processor intervention. The cursor controller may interrupt the host processor either each time the cursor moves (TRACK mode) or only upon operator

request (ENTER function), and the host computer may read or write cursor location and status. The cursor target element is blanked in order to provide for accurate cursor positioning.

OVERLAY CAPABILITY

The Type I and Type II Video Generator boards provide for two independent overlay channels. Overlay channels are independent refresh memory plane outputs. These are mixed with data channel outputs in order to form a composite image output to a single television monitor. Thus, either may be modified without affecting the other.

CURSOR AND OVERLAY MIXING

Cursor, overlay and data channel outputs are mixed via PROM. Thus, individual cursor and overlay color or intensity can be specified by the user. Because data channel outputs are also subjected to the PROM, color priorities may be established. For example, red might be given priority over green.

PSEUDO COLOR AND GREY SCALE TRANSLATION

Pseudo color and grey scale translation is performed using the look-up table contained in the Type II Video Generator. The table is a 1,024 word x 12 bit RAM which is loaded via the host computer, and indexed by up to 10 bits of refresh memory. As each pixel is scanned from refresh memory, the corresponding word is retrieved from the function table and passed to the cursor/overlay mixers and digital-to-analog converters. For color translation, each 12 bit word is divided into three 4 bit binary fractions which, from left to right, describe the relative intensity of the primary colors (red, green and blue). For grey scale translation, the least significant 8 bits are treated as a single binary fraction which describes the output grey level intensity.

EXTERNAL SOURCE SYNCRHONIZER (VCO)

The crystal oscillator which is utilized to generate all of the video sync and blanking signals for the 9000 Series Systems may be replaced by a Phase Locked Loop (PLL) Voltage Controlled Oscillator (VCO). The VCO receives RS-170 vertical drive and RS-170 horizontal drive inputs from an external source provided by the user and synchronizes the screen refresh for the 9000 Series System to the user's system. The PLL/VCO has two modes of operation. In the absence of an external vertical or horizontal drive input, the VCO is connected automatically to a crystal which runs at the nominal frequency required for the specified resolution of the system. If both the vertical and horizontal drive inputs are connected, the VCO is automatically switches to the external inputs and the 9000 Series System becomes phase locked to the external source. The VCO requires approximately four frame times in order to achieve phase lock to the external RS-170 signals.

INTERFACING

Table 1-2 lists the computer interfaces which have been developed by Ramtek. In addition, certain computer mainframe manufacturers have developed Ramtek interfaces. Most are 16 bit parallel, bidirectional interfaces which typically use or implement direct memory access (DMA) in the host processor.

1.3 PROGRAMMING SPECIFICATIONS

Table 1-3 lists the user instruction repertoire of the Models 9100, 9200 and 9300 Graphic Display Systems. There are two distinctive categories of instructions and a variety of instruction formats. The machine instruction category provides for execution of machine-oriented functions such as directly loading or reading specific machine registers. *Their use requires an intimate familiarity of the display generator, its architecture registers, their format, and the specific hardware function associated with each.*

TABLE 1-2 COMPUTER INTERFACES

MODEL NO.	DESCRIPTION	PREREQUISITE	MAX CABLE LENGTH
RM9000-40	General purpose interface board	RM-9000 SERIES SYSTEM	CPU dependent
RM9000-51	DEC PDP-11 series bidirectional interface to DR11C.	DEC PDP-11 DR-11C I/O board.	10 ft.
RM9000-52	DEC PDP-11 series bidirectional interface to DR-11B.	DEC PDP-11 DR-11B I/O board	10 ft.
RM9000-56	Interdata 70/80 series bidirectional interface to programmed I/O multiplexer bus or SELH.	Interdata multiplexer bus or SELH.	15 ft.
RM9000-59	Univac AN/UYK-7 bidirectional interface to Navy Type B (NTDS FAST) digital data interface (OV, -3 V logic levels). Usually a DMA interface. See Mil-Std-1397 (ships). Note: actual cable not provided. Ramtek connectors provided.	Navy Type B NTDS FAST digital data interface (I/O controller).	150 ft.
RM9000-62	HP2000 series bidirectional interface to HP microcircuit boards.	2 each HP microcircuit I/F kits #12566B.	10 ft.
RM9000-63	Texas Instruments 980 series bidirectional interface to programmed I/O channel.	Ramtek logic is wired on a TI logic board (P/N 8EX112) which is plugged into any standard I/O bus connector.	15 ft.
RM9000-64	Varian 620 and 73 series bidirectional interface to buffered I/O contoller board.	Varian (BIOC) buffered I/O controller board (P/N E2832) and priority interrupt module (PIM) Model 620/i-16.	15 ft.
RM9000-65	Data General Nova and Eclipse bidirectional interface.	4192 internal cable in customer's CPU.	35 ft.
RM9000-12	Xerox Sigma series computer Model 7902 extended device subcontroller unidirectional interface to video system only.	Xerox Model 7902 extended device subchannel.	10 ft.

TABLE 1-3 9100, 9200 and 9300 INSTRUCTION SUMMARY

Instruction Name	Mnemonic	Description
LOAD REGISTER	LOAD	Directly loads a display generator machine register.
READ REGISTER	READ	Directly reads a display generator machine register.
WRITE AUXILLIARY MEMORY	WAM	Writes data to an auxilliary memory device.
READ AUXILLIARY MEMORY	RAM	Reads data from an auxilliary memory device.
RESET	RESET	Initializes the programmable stack and font, clears all serial ports and interrupts, and presets all complex format instruction arguments to known values.
INITIALIZE	INIT	Presets all complex format instruction arguments to known values.
NO-OPERATION	NOP	Performs no operation whatsoever.
SET PARAMETER	SET	Sets one or more complex format instruction arguments.
ERASE	ERS	Erases a rectangular area (page) of refresh memory.
WRITE IMAGE	WI	Writes three dimensional image data into a rectangular area (page) of the refresh memory.
READ IMAGE	RI	Retrieves three dimensional image data from a rectangular area (page) of the refresh memory.
WRITE TEXT	WT	Translates USASCII character codes to font data which is written into a rectangular area (page) of refresh memory.
WRITE RASTER	WR	Writes bit-per-element raster data into a rectangular area (page) of the refresh memory.
WRITE VECTOR	WV	Draws (or erases) a series of end-point vectors.
WRITE CONIC	WC	Draws (or erases) a prescribed conic.
WRITE PLOT	WP	Draws (or erases) a line plot, filled plot, or bar chart.
SCROLL X	SCRX	Performs a partitioned and windowed horizontal scroll.
SCROLL Y	SCRY	Performs a partitioned and windowed vertical scroll.
SAVE ENRIVONMENT	PUSH	Stores the current display system environment (including all 60 complex format instruction arguments) onto an internal processor stack.
RESTORE ENVIRONMENT	POP	Retrieves the last stored display system environment from the internal processor stack.
WRITE PROGRAMMABLE FONT	WPF	Writes bit-per-element font data into the character generators font buffer.
WRITE CURSOR STATE	WCS	Positions one of four cursors and establishes its state.
READ CURSOR STATUS	RCS	Retrieves the position and status of a particular cursor.
WRITE KEYBOARD	WKB	Writes a single character to a particular keyboard.
READ KEYBOARD	RKB	Reads the oldest buffered keystroke.
SENSE PERIPHERAL STATUS	SPS	Reads an internally maintained status word which indentifies the locally interfaced peripheral devices which are contending for the host processors attention.

The user instruction category provides for execution of user-oriented functions such as write image, text, vector and plot. It is anticipated that most users will employ only this second, high level instruction set. The design goals were simplicity, speed, power and flexibility. As a result, only a cursory knowledge of the 9000 Series System is required.

INSTRUCTION FORMATS

Figure 1-2 illustrates the standard instruction formats. The SHORT FORMAT instructions are a single 16 bit word in length and carry 8 bits of operand to the 9000 Series System. The IMMEDIATE FORMAT instructions are two 16 bit words in length and carry 8 bits of operand and 16 bits of data to the system. The COMPLEX FORMAT instructions are a variable number of 16 bit words in length and may carry both arguments and data to the system.

ARGUMENTS

Table 1-4 lists the 16 possible arguments to the COMPLEX FORMAT instructions. Except for NOP, any of these instructions may set any combination of the possible arguments. The arguments are issued in ascending sequence. The OPERAND FLAG WORD specifies the presence of each argument. Interpretation is from right to left, that is, bit 2^0 flags argument 01 (SUBCHANNELS), bit 2^1 flags argument 02 (FOREGROUND), ..., and bit 2^{15} flags argument 16 (START-POINT). Each argument is one or more 16 bit words in length and describes a parameter which typically affects the operation of the instruction being decoded.

DATA

Certain of the COMPLEX FORMAT instructions may carry data to the system, or retrieve data from the system. In either case, the DATA LENGTH WORD specifies the number of bytes to be transferred. Data format and interpretation varies from instruction to instruction. For image data, each 16 bit word (2 bytes) is interpreted as a

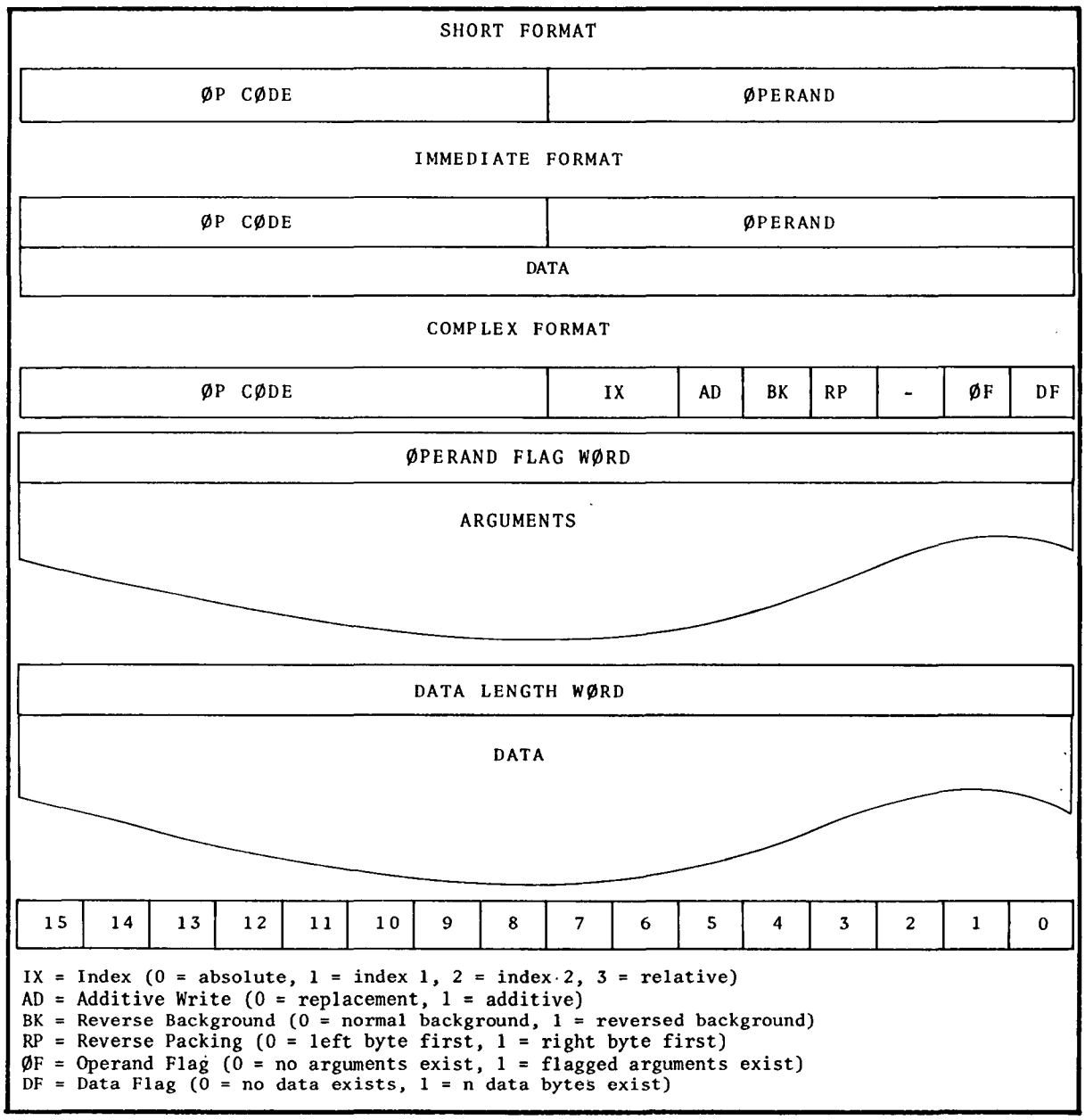


FIGURE 1-2 9100, 9200 AND 9300 INSTRUCTION FORMATS

TABLE 1-4 COMPLEX FORMAT INSTRUCTION ARGUMENTS

NO.	NAME	WORD LENGTH	DESCRIPTION
01	SUBCHANNELS	01	SELECTS REFRESH MEMORY BIT PLANES TO BE AFFECTED
02	FOREGROUND	01	DEFINES COLOR OR INTENSITY FOR "ONE" STATE TWO DIMENSIONAL DATA
03	BACKGROUND	01	DEFINES COLOR OR INTENSITY FOR "ZERO" STATE TWO DIMENSIONAL DATA
04	INDEX-1	02	DEFINES LOCAL REFRESH MEMORY ADDRESSING ORIGIN
05	INDEX-2	02	DEFINES LOCAL REFRESH MEMORY ADDRESSING ORIGIN
06	ORIGIN	02	IMPLEMENTS HARDWARE SCROLL BY ASSIGNING A PARTICULAR REFRESH MEMORY ADDRESS AS THE ORIGIN OF THE TV RASTER
07	WINDOW	04	DEFINES A RECTANGULAR AREA (OR LOGICAL PAGE) OF REFRESH MEMORY
08	SCAN	01	SELECTS ONE OF EIGHT REFRESH MEMORY SCAN SEQUENCES. ALSO SELECTS CHARACTER ORIENTATION AND PLOT DIRECTION
09	DIMENSION	02	SELECTS CHARACTER MATRIX SIZE AND PLOT/BAR SEGMENT WIDTH
10	SPACING	02	SELECTS CHARACTER AND PLOT/BAR SEGMENT SPACING
11	SCALE	01	DEFINES UP/DOWN SCALING RATIO OF UP TO 4:1 or 1:4
12	FUNCTION	01	SELECTS ONE OF EIGHT LOGICAL OR ARITHMETIC IMAGING FUNCTIONS (OR, XOR, AND, SUM, DIFFERENCE, GREATEST VALUE, LEAST VALUE, OR AVERAGE VALUE)
13	CONIC-EQUATION	12	DEFINES CONICAL FORM FOR CONICS GENERATOR IN TERMS OF THE EQUATION: $Ax^2+By^2+Cxy+Dx+Ey+F=0$
14	BASE-LINE	01	SELECTS BETWEEN LINE AND FILLED PLOTS. ALSO DEFINES FIXED POINT BASE LINE FOR FILLED PLOTS AND BAR CHARTS
15	SCROLL-COUNT	01	DEFINES SCROLL COUNT (LINES/ELEMENTS) AND DIRECTIONS
16	START-POINT	02	DEFINES VECTOR, CONIC, PLOT, BAR CHART, OR IRREGULAR IMAGE, TEXT OR RASTER DATA STARTING ADDRESS

single pixel description. For raster data, each 8 bit byte is interpreted as a single-bit-per-element description of eight consecutive pixels. For text data, each byte is interpreted as a USACII character code. For plot data, each 16 bit word (2 bytes) is interpreted as a plot point. Finally, for other graphics data, and each 16 bit word pair (4 bytes) is interpreted as an X, Y end-point coordinate.

1.4 PHYSICAL SPECIFICATIONS

The following data specify the physical characteristics of the Models 9100, 9200 and 9300 Graphic Display Systems:

DIMENSIONS

All systems are mountable in an EIA standard 19 inch wide rack. Adequate support with either rails or slides must be provided. The Models 9100 and 9200 require 8 3/4 inches of rack height and 26 1/4 inches of rack depth. The Model 9300 requires 17 1/4 inches of rack height and 28 inches of rack depth. See drawings which follow, 502853 and 502794, for the chassis outline dimensions.

CHASSIS LAYOUT

Figures 1-3 and 1-4 illustrate the chassis layout for the Models 9100, 9200 and 9300 Graphic Display Systems. Each chassis provides receptacles for a computer interface board, controller board, memory expansion board, two serial link/cursor board, a minimum of two video boards and two to eight memory boards (12 refresh planes). The Model 9300 chassis functionally differs from the Models 9100 and 9200 chassis in that up to four video boards can be accommodated.

1.5 ENVIRONMENTAL SPECIFICATIONS

The following data specify the environmental characteristics of the Models 9100, 9200 and 9300 Graphic Display Systems:

See keys. for

pages 1-20A and 1-20B

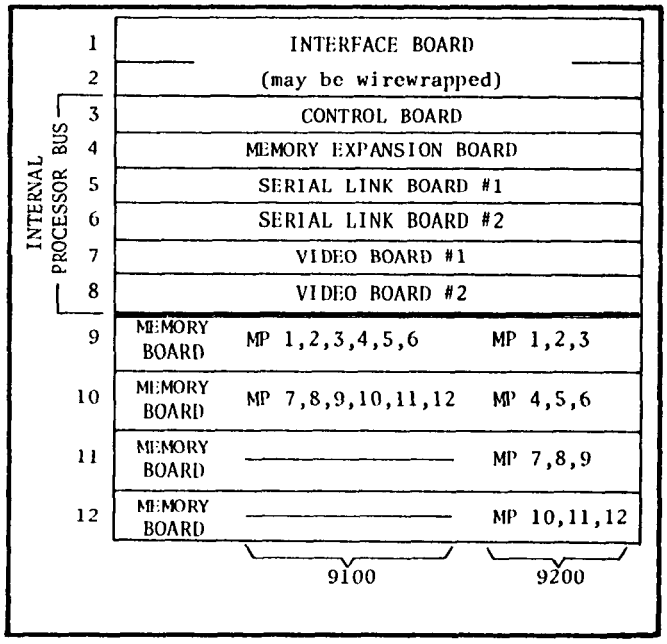


FIGURE 1-3 9100, 9200 CHASSIS BOARD ASSIGNMENTS

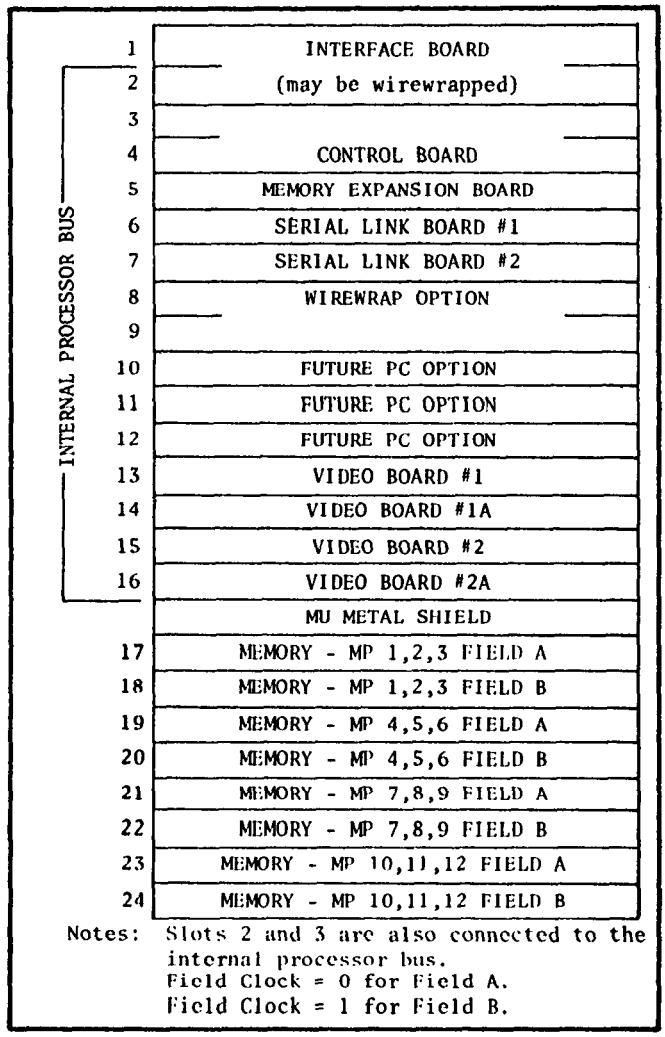


FIGURE 1-4 9300 CHASSIS BOARD ASSIGNMENTS

TEMPERATURE

Operating temperature range is 0-50° C.

HUMIDITY

Relative humidity range is 0-95 percent, noncondensing.

1.6 POWER REQUIREMENTS

The power requirements are as follows:

LINE VOLTAGE

117 or 230 volts AC ±10%

LINE FREQUENCY

50/60 Hz. ±5%

VOLT-AMPERES (FULL CHASSIS)

Model 9100 - 400 V-A
Model 9200 - 500 V-A
Model 9300 - 1000 V-A

2.0 INSTALLATION, MAINTENANCE AND OPERATION

2.1 INITIAL INSPECTION

Each 9000 Series Graphic Display System is carefully inspected both mechanically and electrically before shipment. It should be physically free of mars or scratches and in perfect operating order upon receipt. To confirm this, the system should be inspected for physical damage in transit. Check major component assemblies to determine if any assemblies or screws have been loosened by vibration. In particular, verify that all boards are seated deeply into their connectors and that they have not been "jarred" out of their guides. Tighten any loose screws or mounting hardware as required. Inspect input receptacles for foreign material which may impair electrical contact when cable connections are made. Also check for supplied accessories. If there is damage or deficiency, see the warranty contained in this manual.

2.2 RACK MOUNTING/BOARD ASSIGNMENTS

The controller chassis and other chassis supplied as a part of a particular system configuration should be mounted in a standard EIA 19 inch wide rack cabinet, such as the Ramtek GR-101 or GR-102, and located on a firm surface. Drawing 502819 for the models 9100, 9200 and Drawing 502821 for the model 9300 show the dimensional and installation information necessary to accomplish the physical installation. Never mount a 9000 Series Display System in a rack without using rails or slides to support the chassis weight. The display monitors and other associated equipment can be rack mounted or used as desk top units. Figure 2-1 shows the rear view of the 9100/9200 and Figure 2-2 shows a rear view of the 9300. These drawings show the positions of all rear panel connectors as well as the switching power supplies.

Figure 2-3 and 2-4 show the board slot assignments for standard 9000 boards. Note that special video boards usually reside in Slot 7 for the 9100/9200 and Slot 13 for the 9300. Consult the "Configuration Sheet" packed with the equipment for specific system configurations.

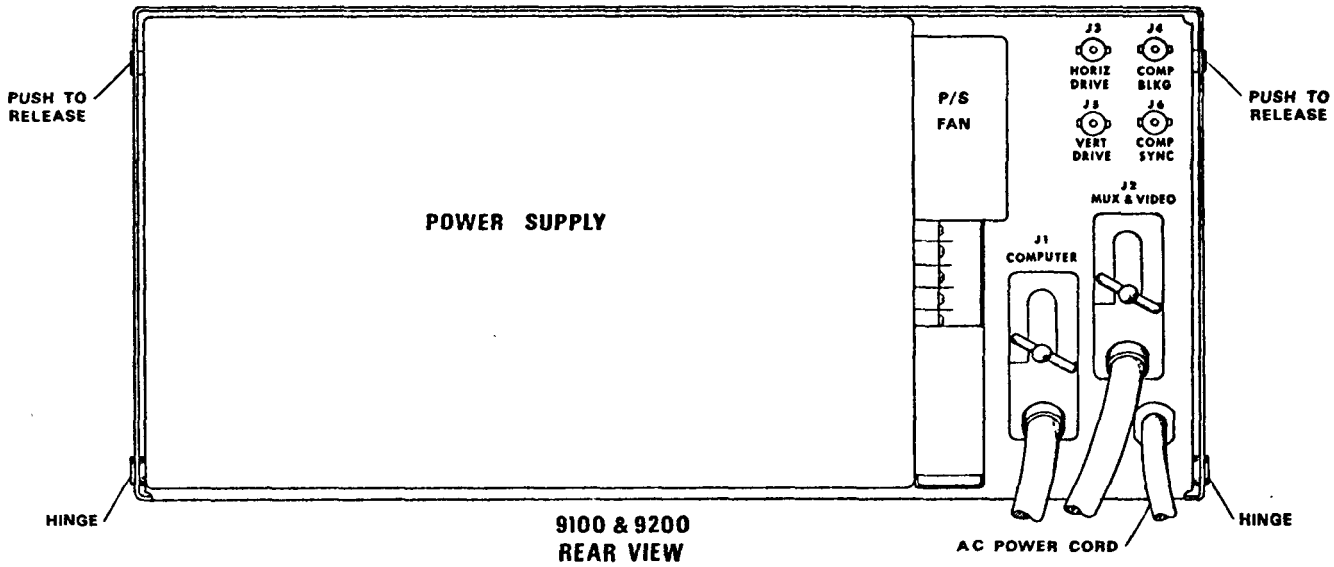


FIGURE 2-1

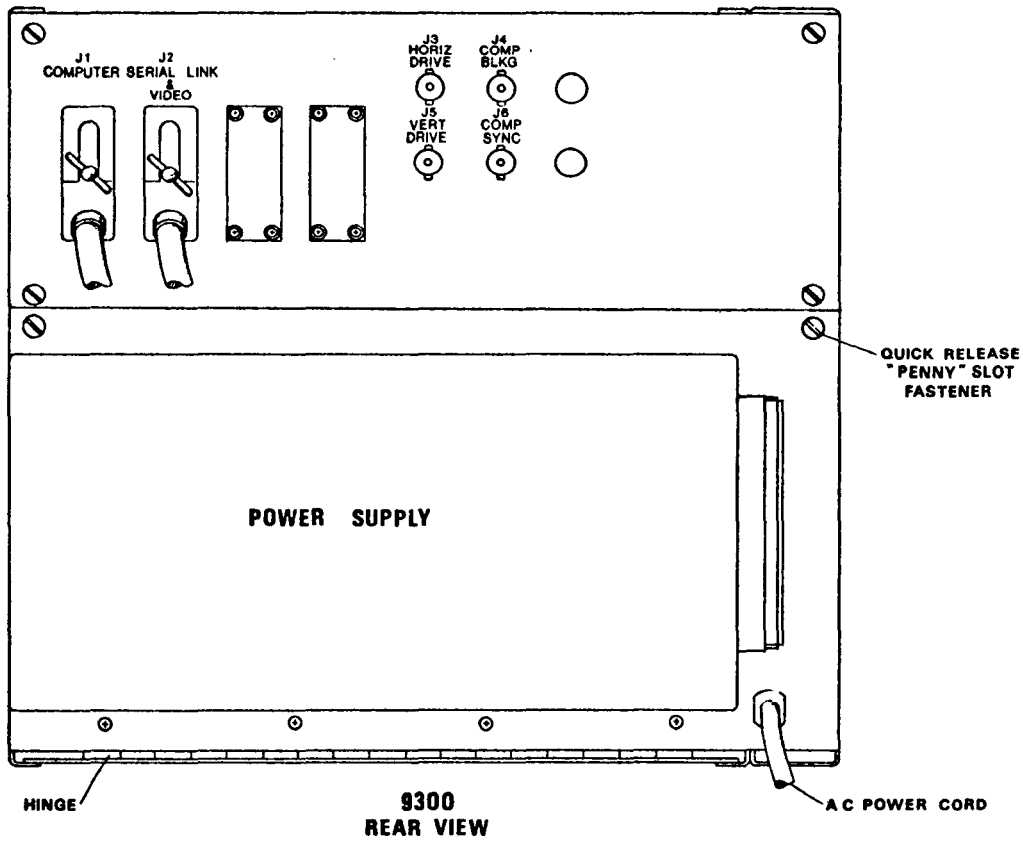


FIGURE 2-2

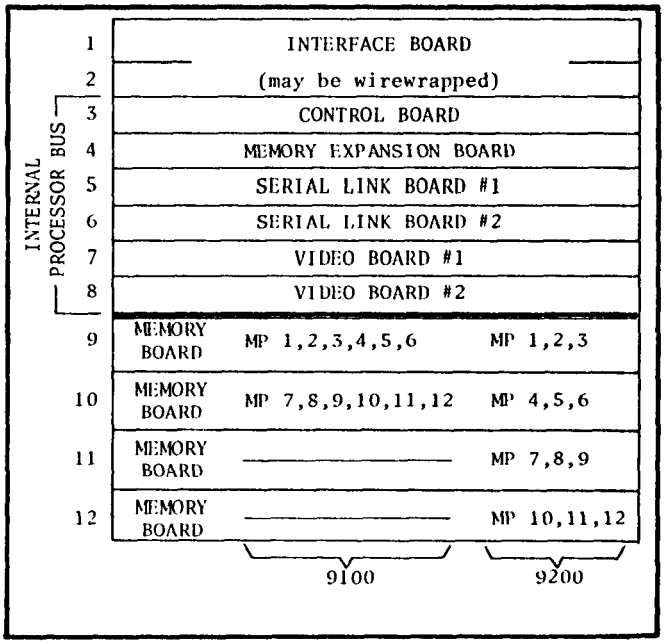


FIGURE 2-3 9100, 9200 CHASSIS BOARD ASSIGNMENTS

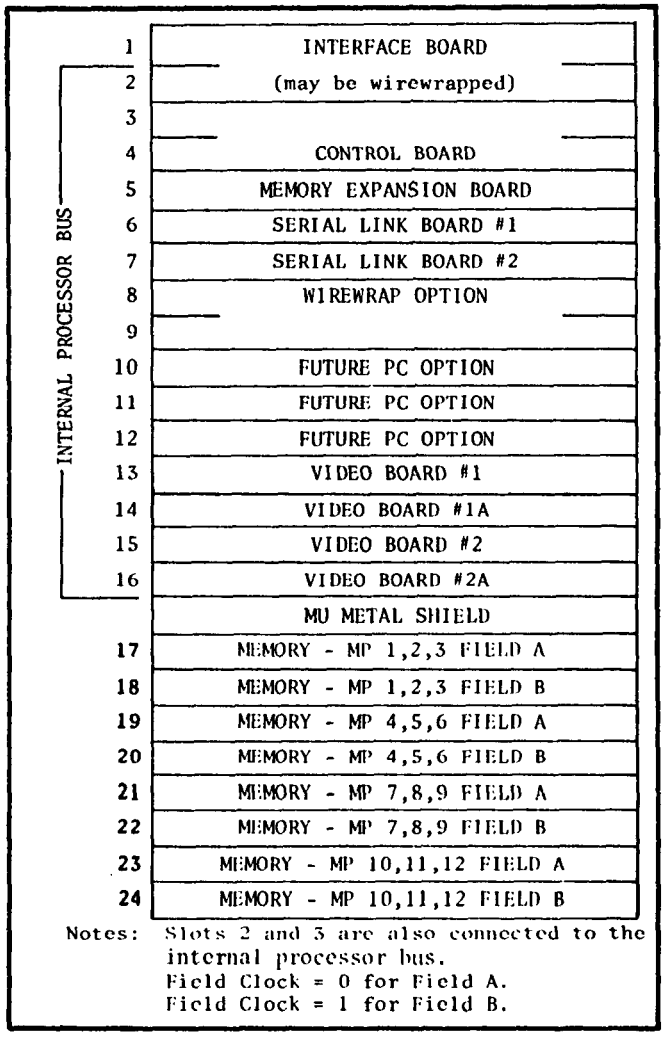


FIGURE 2-4 9300 CHASSIS BOARD ASSIGNMENTS

2.3 CABLING

To protect operating personnel, the National Electrical Manufacturers' Association (NEMA) recommends that the equipment panel and cabinet be grounded. The 9000 Series Systems are equipped with a three-conductor power cable, which, when plugged into an appropriate receptacle, grounds the equipment cabinet. The offset pin on the power cable three-prong connector is the ground wire. *Note that signal ground is isolated from earth ground.*

To preserve the protection feature when operating the 9000 Series Systems from a two-contact outlet, use a three-prong to two-prong adapter and connect the green pigtail on the adapter to ground.

The basic cabling configuration for a system usually includes those specific cables ordered with a particular system, which typically includes the following:

a. Computer I/O cable.
b. Distribution panel (video and serial link).
c. Video-Serial link octopus.
d. Video coax cables.

The 9000 Series Systems have two 104 pin connectors used as follows:

J1 - Computer I/O
J2 - Video outputs and serial link I/O

All three models use identical wiring.

2.4 CONNECTOR REFERENCE INFORMATION

The two connectors (J1, J2) previously mentioned are guide pin coded to prevent incorrect connection. Figure 2-5 shows the detailed pin layout of the 104 pin connectors. Note that power and ground functions are

usually assigned to lettered pins. Three digit pin numbers denote column (first digit) and number of pins down (second and third digit). The connector parts are listed in Table 2-1.

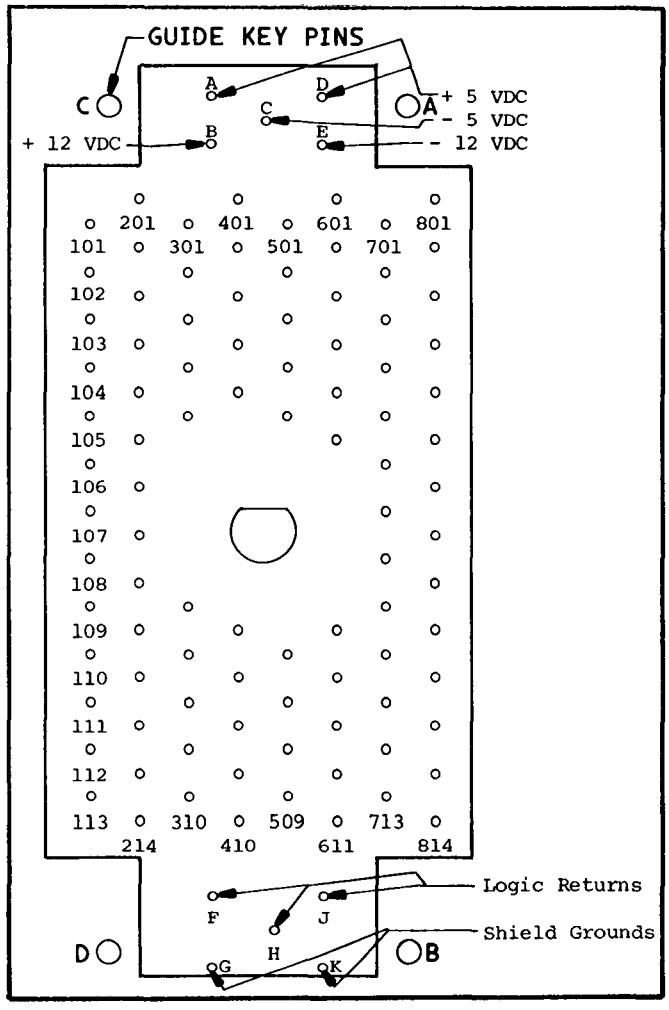


FIGURE 2-5 104 PIN CONNECTOR

The pin numbers in Figure 2-5 are as viewed looking at the connector socket on the rear of the 9000 Series Systems.

TABLE 2-1 PARTS LIST, 104 PIN CONNECTOR

DESCRIPTION	BURNDY	AMP
Plug	MSD104PM-410	201692-3 201692-4
Receptacle	MSD104RM-402	201532-2 201532-4
Hood, Cable Clamp Jackscrew	MSH104M-1	201110-1
Pins, 24-26 AWG	SM24M-1027 SM24M-1 SM24N-TK6	66306-4 66306-3 66306-1
Sockets, 24-26 AWG	SC24M-1027 SC24M-1 SC24M-1TK6	66308-4 66308-3 66308-1
Pins, Coax RG174/U	RMDX60-4D29	51562-1
Socket, Coax RG174/U	RCDX60-4D29	51564-1
Guide Pins, Female	MS104PM424P5	201047-4 201047-2
Guide Pins, Male	MS104PM424P6	201046-4 201046-2
Hand Crimp Tool	M8ND	
Crimp Die	N24RT-11	
Removal Tool	RX20-25V2	
Note: Burndy pins will not work with AMP blocks and visa versa. Completed non-coax assemblies will plug into each other.		

2.5 COMPUTER I/O CABLE

Table 2-2 lists all of the pin connections between the 104 pin computer I/O connector, J1 on the display system back panel and the "B" connector (BCC, BCW) on the computer interface board (Slot #1). This internal cable is constructed from flat ribbon conductor and thus consecutive pin numbers are physically next to each other in the cable. Table 2-2 may be used by the interface designer to assign signal-ground pairs. Note that the signal assignments are different for each interface. The user can consult the interface logic schematics for specific pin assignments.

The 9000 Series System may be remotely controlled thru the CPU cable. Pins 306 and 606 in the CPU connector J1 may be remotely connected to a contact closure or open collector transistor. Pins 606 and 306 should be cabled with a twisted pair with 606 used as return. The remote power control feature is operated in parallel with the front panel switch. If the front panel switch is turned on the system will be powered up regardless of the remote control. If the front panel switch is turned OFF, the system may be powered ON by the remote control.

Before turning on equipment power, check the fuse on the power supply on the rear panel of each chassis; it should be a 10 ampere slow-blow type.

Check that all circuit board assemblies are properly installed.

NOTE

> Do not attempt to verify D.C. voltages with circuit board assemblies removed, switching power supplies require a load to establish proper power on sequencing. Refer to paragraph 2.8 for power supply check out procedures.

Table 2-2 Interface to CPU Connector, J1

J1 PIN	INTERFACE CONNECTION	TERM NAME	FUNCTION
101	BCC 101		
102	BCC 103		
103	BCC 105		
104	BCC 107		
105	BCC 109		
106	BCC 111		
107	BCC 113		
108	BCC 115		
109	BCC 117		
110	BCC 119		
111	BCC 121		
112	BCC 123		
113	BCC 125		
201	BCC 102		
202	BCC 104		
203	BCC 106		
204	BCC 108		
205	BCC 110		
206	BCC 112		
207	BCC 114		
208	BCC 116		
209	BCC 118		
210	BCC 120		
211	BCC 122		
212	BCC 124		
213	BCC 126		
214	BCW 105		
301	BCC 127		
302	BCC 129		
303	BCC 131		
304	BCC 133		
305	BCC 135		
306	—	ZREMOTE	Remote Power Control
307	BCC 137		

Table 2-2 Interface to CPU Connector, J1 (cont'd)

J1 PIN	INTERFACE CONNECTION	TERM NAME	FUNCTION
308	BCC 139		
309	BCW 101		
310	BCW 103		
401	BCC 128		
402	BCC 130		
403	BCC 132		
404	BCC 134		
405	BCC 136		
406	BCC 138		
407	BCC 140		
408	BCW 102		
409	BCW 104		
410	BCW 106		
501	BCW 107		
502	BCW 109		
503	BCW 111		
504	BCW 113		
505	BCW 115		
506	BCW 117		
507	BCW 119		
508	BCW 121		
509	BCW 123		
601	BCW 108		
602	BCW 110		
603	BCW 112		
604	BCW 114		
605	BCW 116		
606	—	#REMOTE	Remote Power Control Twisted Pair Return
607	BCW 118		
608	BCW 120		
609	BCW 122		
610	BCW 124		
611	TCL 233		
701	BCW 125		

Table 2-2 Interface to CPU Connector, J1 (Cont'd)

J1 PIN	INTERFACE CONNECTION	TERM NAME	FUNCTION
702	BCW 127		
703	BCW 129		
704	BCW 131		
705	BCW 133		
706	BCW 135		
707	BCW 137		
708	BCW 139		
709	TCL 228		
710	TCL 229		
711	TCL 230		
712	TCL 231		
713	TCL 232		
801	BCW 126		
802	BCW 128		
803	BCW 130		
804	BCW 132		
805	BCW 134		
806	BCW 136		
807	BCW 138		
808	BCW 140		
809	TCU 328		
810	TCU 329		
811	TCU 330		
812	TCU 331		
813	TCU 332		
814	TCU 333		
A		ZVCC	+5VDC
B		ZP12	+12VDC
C		ZN5	-5VDC
D		ZVCC	+5VDC
E		ZN12	-12VDC
F		ZGND	Logic Return
G		ZSHLD	Shield Ground
H		ZGND	Logic Return
J		ZGND	Logic Return
K		ZSHLD	Shield Ground

2.6 VIDEO AND SERIAL LINK CABLE

The video and serial link cable serves a dual purpose as follows:

a. Interconnection of video monitors to the display system.
b. Interconnection of keyboards, joysticks and trackballs to serial link boards.

The video signals are interconnected via miniature RG-174 coaxial cable and the serial link signals utilize flat ribbon cable. Several versions of internal wiring are supplied depending on the number of serial link boards and the number or type of video boards supplied with a particular system. Table 2-3 shows the internal wiring and various configurations. Table 2-4 summarizes the requirements for the various configurations.

Table 2-5 shows the connector pin assignments for the GK-106 Joystick Cursor Controller and the GK-120 Keyboard. These connectors are on the rear of the units. They are 24 pin Burndy connectors as follows:

Keyboard/Joystick Connector	-	Burndy Female Connector P/N SMS 24R-1
Cable Mating Connector	-	Burndy Male Connector P/N SMS 24P-1
Cable Hood	-	P/N SMS 24H-1
Connector Pins for cable #24 wire	-	P/N SM 201 D27

All interfacing between the 9000 Series and the video display monitors is accomplished through connector J2. Interconnecting cable(s) are usually ordered with the system. Each individual monitor connected to a unique output line must always be terminated with 75 ohms to ground. Note that if more than one monitor is to be connected in series to the same output line, the last monitor in the chain must always be terminated with 75 ohms to ground. All other monitors in the chain should have high input impedance.

TABLE 2-3 SERIAL LINK & VIDEO CABLE

VIDEO	CONNECTION	J2 PIN	TERM NAME	FUNCTION
1	BCW-129	J2-802	VIDEØBW1	GREY SCALE SLOT 1
↑	BCW-130	J2-802	ZGNDBW1	
	BCW-127	J2-702	VIDEØB1	BLUE SLOT 1
	BCW-128	J2-702	ZGNDB1	
	BCW-125	J2-602	VIDEØG1	GREEN SLOT 1
	BCW-126	J2-602	ZGNDG1	
	BCW-123	J2-502	VIDEØR1	RED SLOT 1
	BCW-124	J2-502	ZGNDR1	
	BCW-121	J2-402	VIDEØ11	SUB CHAN 11
	BCW-122	J2-402	ZGND11	
	BCW-119	J2-302	VIDEØ10	SUB CHAN 10
	BCW-120	J2-302	ZGND10	
	BCW-117	J2-202	VIDEØ09	SUB CHAN 9
	BCW-118	J2-202	ZGND09	
	BCW-115	J2-10	VIDEØ8	SUB CHAN 8
	BCW-116	J2-102	ZGNDØ8	
	BCC-129	J2-801	VIDEØ07	SUB CHAN 7
	BCC-130	J2-801	ZGND07	
	BCC-127	J2-701	VIDEØ06	SUB CHAN 6
	BCC-128	J2-701	ZGND06	
	BCC-125	J2-601	VIDEØ05	SUB CHAN 5
	BCC-126	J2-601	ZGND05	
	BCC-123	J2-501	VIDEØ04	SUB CHAN 4
	BCC-124	J2-501	ZGND04	
	BCC-121	J2-401	VIDEØ03	SUB CHAN 3
	BCC-122	J2-401	ZGND03	
	BCC-119	J2-301	VIDEØ02	SUB CHAN 2
	BCC-120	J2-301	ZGND02	
	BCC-117	J2-201	VIDEØ01	SUB CHAN 1
	BCC-118	J2-201	ZGND01	
	BCC-115	J2-101	VIDEØ00	SUB CHAN 0
	BCC-116	J2-101	ZGND00	
	BCW-111	J2-E	9VCMSYE	COMPOSITE SYNC
	BCW-112	J2-E	#VCMSYE	
	BCW-113	J2-D	9VVERTE	VERTICAL DRIVE
	BCW-114	J2-D	#VVERTE	
	BCC-111	J2-B	9VCMPBLE	COMPOSITE BLANKING
	BCC-112	J2-B	#VCMPBLE	
	BCC-113	J2-A	9VHORZE	HORIZONTAL DRIVE
1	BCC-114	J2-A	#VHORZE	

TABLE 2-3 SERIAL LINK & VIDEO CABLE (cont'd)

SERIAL LINK	CONNECTION	J2 PIN	TERM NAME	FUNCTION
1	BCC-131	J2-808	#LLDR45	SERIAL IN CHAN 4 RETURN
	BCC-130	J2-707	9LLDR45	SERIAL IN CHAN 4
	BCC-129	J2-807	#LLDR35	SERIAL IN CHAN 3 RETURN
	BCC-128	J2-706	9LLDR35	SERIAL IN CHAN 3
	BCC-127	J2-806	#LLDR25	SERIAL IN CHAN 2 RETURN
	BCC-126	J2-805	9LLDR25	SERIAL IN CHAN 2
	BCC-121	J2-305	#LLDR15	SERIAL IN CHAN 1 RETURN
	BCC-120	J2-205	9LLDR15	SERIAL IN CHAN 1
	BCW-131	J2-208	#LLDT45	SERIAL OUT CHAN 4 RETURN
	BCW-130	J2-107	9LLDT45	SERIAL OUT CHAN 4
	BCW-129	J2-809	#LLDT35	SERIAL OUT CHAN 3 RETURN
	BCW-128	J2-708	9LLDT35	SERIAL OUT CHAN 3
	BCW-127	J2-209	#LLDT25	SERIAL OUT CHAN 2 RETURN
	BCW-126	J2-108	9LLDT25	SERIAL OUT CHAN 2
	BCW-121	J2-406	#LLDT15	SERIAL OUT CHAN 1 RETURN
1	BCW-120	J2-306	9LLDT15	SERIAL OUT CHAN 1
2	BCC-131	J2-705	#LLDR46	SERIAL IN CHAN 4 RETURN
	BCC-130	J2-605	9LLDR46	SERIAL IN CHAN 4
	BCC-129	J2-606	#LLDR36	SERIAL IN CHAN 3 RETURN
	BCC-128	J2-505	9LLDR36	SERIAL IN CHAN 3
	BCC-127	J2-810	#LLDR26	SERIAL IN CHAN 2 RETURN
	BCC-126	J2-709	9LLDR26	SERIAL IN CHAN 2
	BCC-121	J2-710	#LLDR16	SERIAL IN CHAN 1 RETURN
	BCC-120	J2-607	9LLDR16	SERIAL IN CHAN 1
	BCW-131	J2-608	#LLDT46	SERIAL OUT CHAN 4 RETURN
	BCW-130	J2-506	9LLDT46	SERIAL OUT CHAN 4
	BCW-129	J2-210	#LLDT36	SERIAL OUT CHAN 3 RETURN
	BCW-128	J2-109	9LLDT36	SERIAL OUT CHAN 3
	BCW-127	J2-207	#LLDT26	SERIAL OUT CHAN 2 RETURN
	BCW-126	J2-106	9LLDT26	SERIAL OUT CHAN 2
	BCW-121	J2-206	#LLDT16	SERIAL OUT CHAN 1 RETURN
2	BCW-120	J2-105	9LLDT16	SERIAL OUT CHAN 1
	BCW-122	J2-611	SL2+12	
	BCW-122	J2-405	SL1+12	
	CHASSIS	J2-G	SHLD GND	
	CHASSIS	J2-K	SHLD GND	

TABLE 2-3 SERIAL LINK & VIDEO CABLE (cont'd)

VIDEO	CONNECTION	J2 PIN	TERM NAME	FUNCTION
2	BCW-129	J2-804	VIDEØBW2	GREY SCALE SLOT 2
↕	BCW-130	J2-804	ZGNDBW2	
	BCW-127	J2-704	VIDEØB2	BLUE SLOT 2
	BCW-128	J2-704	ZGNDB2	
	BCW-125	J2-604	VIDEØG2	GREEN SLOT 2
	BCW-126	J2-604	ZGNDG2	
	BCW-123	J2-504	VIDEØR2	RED SLOT 2
2	BCW-124	J2-504	ZGNDR2	
3	BCW-129	J2-407	VIDEØBW3	GREY SCALE SLOT 3
↕	BCW-130	J2-407	ZGNDBW3	
	BCW-127	J2-307	VIDEØB3	BLUE SLOT 3
	BCW-128	J2-307	ZGNDB3	
	BCW-125	J2-211	VIDEØG3	GREEN SLOT 3
	BCW-126	J2-211	ZGNDG3	
	BCW-123	J2-110	VIDEØR3	RED SLOT 3
3	BCW-124	J2-110	ZGNDR3	
		(9300 ONLY)		
4	BCW-129	J2-811	VIDEØBW4	GREY SCALE SLOT 4
↕	BCW-130	J2-811	ZGNDBW4	
	BCW-127	J2-711	VIDEØB4	BLUE SLOT 4
	BCW-128	J2-711	ZGNDB4	
	BCW-125	J2-609	VIDEØ04	GREEN SLOT 4
	BCW-126	J2-609	ZGNDG4	
	BCW-123	J2-507	VIDEØR4	RED SLOT 4
4	BCW-124	J2-507	ZGNDR4	

TABLE 2-4 VIDEO/SERIAL LINK CONFIGURATIONS

CABLE ASSEMBLY NO.	APPLICABILITY
502337-01	9100 or 9200 basic configuration - One Type 2 Video or equivalent and one serial link.
502337-02	9100 or 9200 expanded configurations - One Type 1 Video and one Type 2 Video or equivalents, or two Type 2 Videos, and two serial link.
502789-01	9300 basic configuration - One Type 2 Video or equivalent and one serial link.
502789-02	9300 expanded configurations - One Type 1 Video and three Type 2 Videos or equivalents, or up to four video cards, and two serial links.

TABLE 2-5 CURSOR CONTROLLER/KEYBOARD PIN ASSIGNMENTS

PIN #	CURSOR CONTROLLER FUNCTION	KEYBOARD FUNCTION
1	Channel zero (0) driver non-inverting output-Y	Long-line differential driver non-inverting output-Y
2	Channel zero (0) driver inverting output-Z	Long-line differential driver inverting output-Z
3	Channel zero and one shield logic ground	Long-line differential driver shield - logic ground
4	Unused	'On Line' switch contact - NC contact closure when switch is off
5	Channel one (1) driver non-inverting output-Y	Long-line differential receiver non-inverting input-Y
6	Channel one (1) driver inverting output-Z	Long-line differential receiver inverting input-Z
7	Unused	Long-line differential receiver shield - logic ground
8	Unused	'On Line' switch contact - no contact closure when switch is 'ON'
9	Channel two (2) driver non-inverting output-Y	Current loop driver current current output
10	Channel two (2) driver inverting output-Z	Current loop driver Current return
11	Channel two and three shield - logic ground	Current loop driver shield - logic ground
12	Unused	'On Line' switch contact - common
13	Channel three (3) driver non-inverting output-Y	Current loop receiver current input
14	Channel three (3) driver inverting output-Z	Current loop receiver current return

TABLE 2-5 CURSOR CONTROLLER/KEYBOARD PIN ASSIGNMENTS (cont'd)

PIN #	CURSOR CONTROLLER FUNCTION	KEYBOARD FUNCTION
15	Unused	Current Loop receiver shield - logic ground
16	Unused	+5V DC Power
17	Unused	RS-232C compatible driver inverting output
18	Unused	RS-232C compatible driver return - logic ground
19	Logic Ground	RS-232C compatible driver shield - logic ground
20	Unused	Logic Ground
21	+5V DC power	RS-232C compatible receiver inverting input
22	+5V DC power	RS-232C compatible receiver return - logic ground
23	Logic Ground	RS-232C compatible receiver shield - logic ground
24	Chassis ground Chassis ground is isolated from logic ground within the cursor controller	Chassis ground Chassis ground is isolated from logic ground within the keyboard

Video channels are capable of driving up to 1000 feet of RG-59/U coaxial cable. The amplitude of a video channel will be 1.4 volts peak-to-peak when properly terminated with 75 ohms ±5%. The amplitude will be approximately double when not terminated. Note that some monitors are not supplied with termination connectors and some have built in terminations. This information should be considered when ordering equipment. All monitors supplied by Ramtek have 75 ohm terminations.

The four synchronization signals provided in J2 are horizontal drive, vertical drive, composite blanking and composite sync. The amplitude of these synchronization signals will be 4 volts peak-to-peak when properly terminated with 75 ohms ±5%. These signals are logically low ("1" = true = -4 VDC, "0" = false = 0 VDC). The synchronization outputs are useful for synchronizing other video equipment such as cameras, recorders and printers to the 9000 Series System.

Connector J2 also provides for two synchronization inputs. These are connected internally when the VCO option is added to the system. The input signals should be RS-170 compatible and are used by the optional VCO circuit which would be substituted for the crystal oscillator on the sync-timing assembly. These signals in conjunction with the VCO option allow the 9000 Series System to be synchronized to other video equipment.

2.7 DISTRIBUTION PANELS

Two types of distribution panels are supplied as optional equipment for allowing the user to quickly connect or disconnect individual monitors and/or serial I/O equipment to the 9000 Series System. Figures 2-6 and 2-7 show the two types of optional panels. Type I, the basic version may be used for imaging systems with one serial link and no VCO. Type II fills the need for a Type I Video Board or a VCO or two serial link boards. Tables 2-6 and 2-7 give the wiring for the two distribution panels.

TABLE 2-6 SMALL DISTRIBUTION PANEL WIRING

P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
502	J1	VIDEØ R	RED VIDEO OUTPUT
502	J1	ZGNDR	
602	J2	VIDEØ G	GREEN VIDEO OUTPUT
602	J2	ZGND G	
702	J3	VIDEØ B	BLUE VIDEO OUTPUT
702	J3	ZGND B	
802	J4	VIDEØ BW	GREY SCALE VIDEO OUTPUT
802	J4	ZGND BW	
C	J5	VIDEØ	DIAGNOSTIC VIDEO OUTPUT
C	J5	ZGND	
A	HORZ DRIVE	9VHORZE	HORIZONTAL DRIVE OUTPUT
A	HORZ DRIVE	#VHORZE	
B	COMP BLANK	9VCMPBLE	COMPOSITE BLANKING OUTPUT
B	COMP BLANK	#VCMPBLE	
D	VERT DRIVE	9VVERTE	VERTICAL DRIVE OUTPUT
D	VERT DRIVE	#VVERTE	
E	COMP SYNC	9VCMSYE	COMPOSIT SYNC OUTPUT
E	COMP SYNC	#VCMSYE	
306	T1-5	9LLDT15	SERIAL OUT CHANNEL 1
406	T1-4	#LLDT15	SERIAL OUT CHANNEL 1 RETURN
205	T1-2	9LLDR15	SERIAL IN CHANNEL 1
305	T1-1	#LLDR15	SERIAL IN CHANNEL 1 RETURN
108	T2-5	9LLDT25	SERIAL OUT CHANNEL 2
209	T2-4	#LLDT25	SERIAL OUT CHANNEL 2 RETURN
805	T2-2	9LLDR25	SERIAL IN CHANNEL 2
806	T2-1	#LLDR25	SERIAL IN CHANNEL 2 RETURN
708	T3-5	9LLDT35	SERIAL OUT CHANNEL 3
809	T3-4	#LLDT35	SERIAL OUT CHANNEL 3 RETURN
706	T3-2	9LLDR35	SERIAL CHANNEL 3
	T3-8		
	T4-8		

TABLE 2-6 SMALL DISTRIBUTION PANEL WIRING (cont'd)

P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
807	T3-1	#LLDR35	SERIAL IN CHANNEL 3 RETURN
	T3-7		
	T4-7		
107	T4-5	9LLDT45	SERIAL OUT CHANNEL 4
208	T4-4	#LLDT45	SERIAL OUT CHANNEL 4 RETURN
707	T4-2	9LLDR45	SERIAL IN CHANNEL 4
	T4-11		
	T3-11		
808	T4-1	#LLDR45	SERIAL IN CHANNEL 4 RETURN
	T4-10		
	T3-10		
405	T1-10	SL1+12	+12 VDC
	T2-10	SL1+12	+12 VDC
G	T1-6	SHIELD	
	T2-6		
	T3-6		
	T3-12		
	T4-6		
	T4-12		
K	T1-3	SHIELD	
	T2-3		
	T3-3		
	T3-9		
	T4-3		
	T4-9		

TABLE 2-7 LARGE DISTRIBUTION PANEL WIRING

VIDEO	P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
1	101	J17	VIDEØ00	SUB CHAN 0 VIDEO OUTPUT
	101	J17	ZGND00	
	201	J18	VIDEØ01	SUB CHAN 1 VIDEO OUTPUT
	201	J18	ZGND01	
	301	J19	VIDEØ02	SUB CHAN 2 VIDEO OUTPUT
	301	J19	ZGND01	
	401	J20	VIDEØ03	SUB CHAN 3 VIDEO OUTPUT
	401	J20	ZGND03	
	501	J21	VIDEØ04	SUB CHAN 4 VIDEO OUTPUT
	501	J21	ZGND04	
	601	J22	VIDEØ05	SUB CHAN 5 VIDEO OUTPUT
	601	J22	ZGND05	
	701	J23	VIDEØ06	SUB CHAN 6 VIDEO OUTPUT
	701	J23	ZGND06	
	801	J24	VIDEØ07	SUB CHAN 7 VIDEO OUTPUT
	801	J24	ZGND07	
	102	J25	VIDEØ08	SUB CHAN 8 VIDEO OUTPUT
	102	J25	ZGND08	
	202	J26	VIDEØ09	SUB CHAN 9 VIDEO OUTPUT
	202	J26	ZGND09	
	302	J27	VIDEØ10	SUB CHAN 10 VIDEO OUTPUT
	302	J27	ZGND10	
	402	J28	VIDEØ11	SUB CHAN 11 VIDEO OUTPUT
	402	J28	ZGND11	
	502	J1	VIDEØR1	RED VIDEO OUTPUT
	502	J1	ZGNDR1	
	602	J2	DVIDEØG1	GREEN VIDEO OUTPUT
	602	J2	ZGNDG1	
	702	J3	DVIDEØB1	BLUE VIDEO OUTPUT
1	702	J3	ZGNDB1	

TABLE 2-7 LARGE DISTRIBUTION PANEL WIRING (cont'd)

VIDEO	P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
1	802	J4	DVIDEØBW1	GREY SCALE VIDEO OUTPUT
	802	J4	ZGNDBW1	
	A	HORZ DRIVE	9VHORZE	HORIZONTAL DRIVE OUTPUT
	A	HORZ DRIVE	#VHORZE	
	B	COMP BLANK	9VCMPBLE	COMPOSITE BLANKING OUTPUT
	B	COMP BLANK	#VCMPBLE	
	D	VERT DRIVE	9VVERTE	VERTICAL DRIVE OUTPUT
	D	VERT DRIVE	#VVERTE	
	E	COMP SYNC	9VCMSYE	COMPOSITE SYNC OUTPUT
1	E	COMP SYNC	#VCMSYE	
2	504	J5	VIDEØR2	RED VIDEO OUTPUT
	504	J5	ZGNDR2	
	604	J6	DIVIDEØ62	GREEN VIDEO OUTPUT
	604	J6	ZGNDG2	
	704	J7	DIVIDEØB2	BLUE VIDEO OUTPUT
	704	J7	ZGNDB2	
	804	J8	DIVIDEØBW2	GREY SCALE VIDEO OUTPUT
2	804	J8	ZGNDBW2	
3	110	J9	VIDEØR3	RED VIDEO OUTPUT
	110	J9	ZGNDR3	
	211	J10	VIDEØG3	GREEN VIDEO OUTPUT
	211	J10	ZGNDG3	
	307	J11	VIDEØB3	BLUE VIDEO OUTPUT
	307	J11	ZGNDB3	
	407	J12	VIDEØBW3	GREY SCALE VIDEO OUTPUT
3	407	J12	ZGNDBW3	

TABLE 2-7 LARGE DISTRIBUTION PANEL WIRING (cont'd)

VIDEO	P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
4	507	J13	VIDEØR4	RED VIDEO OUTPUT
↕	507	J13	ZGNDR4	
	609	J14	VIDEØG4	GREEN VIDEO OUTPUT
	609	J14	ZGNDG4	
	711	J15	VIDEØB4	BLUE VIDEO OUTPUT
	711	J15	ZGNDB4	
	811	J16	VIDEØBW4	GREY SCALE VIDEO OUTPUT
4	811	J16	ZGNDBW4	
	410	VERT DRIVE	9XVERTE	VERTICAL DRIVE INPUT
	410	VERT DRIVE	#XVERTE	
	509	HORZ DRIVE	9XHORZE	HORIZONTAL DRIVE INPUT
	509	HORZ DRIVE	#XHORZE	

TABLE 2-7 LARGE DISTRIBUTION PANEL WIRING

SERIAL LINK	P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
1	306	T1-5	9LLDT15	SERIAL OUT CHANNEL 1
↑	406	T1-4	#LLDT15	SERIAL OUT CHANNEL 1 RTN
↑	205	T1-2	9LLDR15	SERIAL IN CHANNEL 1
↑	305	T1-1	#LLDR15	SERIAL IN CHANNEL 1 RETURN
↑	108	T2-5	9LLDT25	SERIAL OUT CHANNEL 2
↑	209	T2-4	#LLDT25	SERIAL OUT CHANNEL 2 RETURN
↑	805	T2-2	9LLDR25	SERIAL IN CHANNEL 2
↑	806	T2-1	#LLDR25	SERIAL IN CHANNEL 2 RETURN
↑	708	T3-5	9LLDT35	SERIAL OUT CHANNEL 2
↑	809	T3-4	#LLDT35	SERIAL OUT CHANNEL 2 RETURN
↑	706	T3-2	9LLDR35	SERIAL IN CHANNEL 3
↑		T9-2	9LLDR35	
↑	807	T3-1	#LLDR35	SERIAL IN CHANNEL 3 RETURN
↑		T9-1	#LLDR35	
↑	107	T4-5	9LLDT45	SERIAL OUT CHANNEL 3
↑	208	T4-4	#LLDT45	SERIAL OUT CHANNEL 3 RETURN
↑	707	T4-2	9LLDR45	SERIAL IN CHANNEL 4
↑		T9-5	9LLDR45	
↑		T10-5	9LLDR45	
↑	808	T4-1	#LLDR45	SERIAL IN CHANNEL 4 RETURN
↓		T9-4	#LLDR45	
1		T10-4	#LLDR45	
2	105	T5-5	9LLDT16	SERIAL OUT CHANNEL 1
↑	206	T5-4	#LLDT16	SERIAL OUT CHANNEL 1 RETURN
↑	607	T5-2	9LLDR16	SERIAL IN CHANNEL 1
↑	710	T5-1	#LLDR16	SERIAL IN CHANNEL 1 RETURN
↑	106	T6-5	9LLDT2L	SERIAL OUT CHANNEL 2
↑	207	T6-4	#LLDT26	SERIAL OUT CHANNEL 2 RETURN
↑	709	T6-2	9LLDR26	SERIAL IN CHANNEL 2
2	810	T6-1	#LLDR26	SERIAL IN CHANNEL 2 RETURN

TABLE 2-7 LARGE DISTRIBUTION PANEL WIRING (cont'd)

SERIAL LINK	P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
2	109	T7-5	9LLDT36	SERIAL OUT CHANNEL 3
	210	T7-4	#LLDT36	SERIAL OUT CHANNEL 3 RETURN
	505	T7-2	9LLDR36	SERIAL IN CHANNEL 3
		T9-8	9LLDR36	
		T11-8	9LLDR36	
	606	T7-1	#LLDR36	SERIAL IN CHANNEL 3 RETURN
		T9-7	#LLDR36	
		T11-7	#LLDR36	
	506	T8-5	9LLDT46	SERIAL OUT CHANNEL 4
	608	T8-4	#LLDT46	SERIAL OUT CHANNEL 4 RETURN
	605	T8-2	9LLDR46	SERIAL IN CHANNEL 4
		T9-11	9LLDR46	
		T12-11	9LLDR46	
	705	T8-1	#LLDR46	SERIAL IN CHANNEL 4 RETURN
		T9-10	#LLDR46	
2		T12-10	#LLDR46	
	405	T1-10	SL1+12	+12 VDC
		T2-10	SL1+12	
		T3-10	SL1+12	
		T4-10	SL1+12	
	611	T5-10	SL2+12	+12 VDC
		T6-10	SL2+12	
		T7-10	SL2+12	
		T8-10	SL2+12	
	K	T3-3	SHLD GND	
		T3-6	SHLD GND	
		T4-3	SHLD GND	
		T4-6	SHLD GND	

TABLE 2-7 LARGE DISTRIBUTION PANEL WIRING (cont'd)

SERIAL LINK	P2 PIN	PANEL CONNECTOR	TERM NAME	DESCRIPTION
		T8-6	SHLD GND	
		T8-3	SHLD GND	
		T7-6	SHLD GND	
		T7-3	SHLD GND	
		T11-9	SHLD GND	
		T12-12	SHLD GND	
	G	T1-3	SHLD GND	
		T1-6	SHLD GND	
		T2-3	SHLD GND	
		T2-6	SHLD GND	
		T6-6	SHLD GND	
		T6-3	SHLD GND	
		T5-6	SHLD GND	
		T5-3	SHLD GND	
		T9-3	SHLD GND	
		T9-6	SHLD GND	
		T9-9	SHLD GND	
		T9-12	SHLD GND	
		T10-6	SHLD GND	

2.8 POWER ADJUSTMENTS AND MODIFICATIONS

The 9100 and 9200 utilize a smaller power supply than the 9300. Figures 2-8 and 2-9 show the power wiring for the 9100/9200 and the 9300 respectively. Note that power is also supplied to the CPU connector for use by Ramtek with test aids. The power supplies may be connected for 110 or 220 volt operation. When operating as a 220 VAC system the fans are wired such that they always operate from 110 VAC. As shown in Figures 2-8 and 2-9, 110 VAC and 220 VAC operations require the two AC jumpers as follows:

110 VAC Operating:
Jumper 1 between TB1-1 and TB1-2

220 VAC Operation:
Jumper 1 between TB1-2 and TB1-3
Jumper 2 between TB1-4 and TB1-5

CAUTION

The fans and AC input wires must be wired exactly as shown or the AC jumpers will cause severe damage to the system.

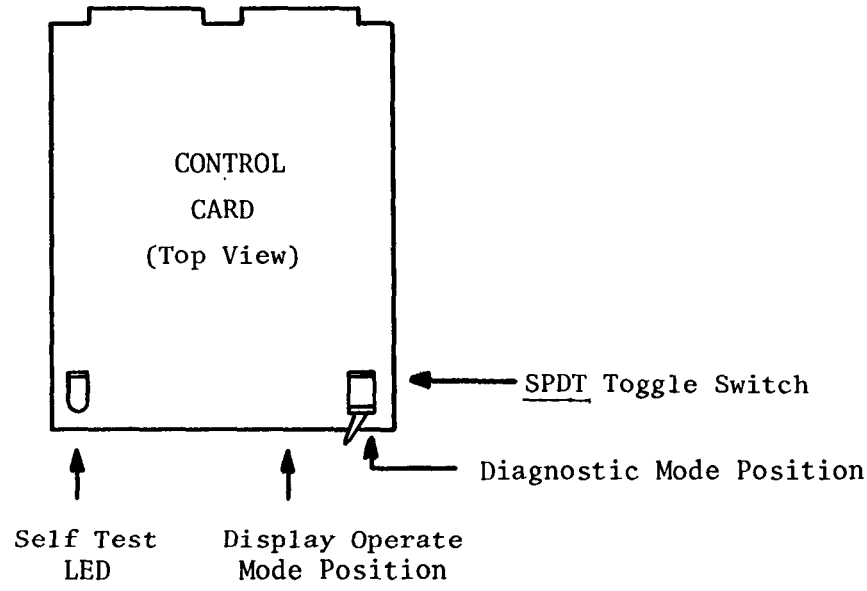
Power supply voltages should be within ±5% of nominal values shown on the power wiring diagrams. It is necessary to maintain a minimum load of two amperes on the +5 V for the 9100, 9200 power supplies when attempting to adjust or test power. It is necessary to maintain a minimum load of three amperes for +5 V and two amperes for +12 V when testing 9300 power. It should never be necessary to adjust power voltages, however, all voltages are adjustable for both supplies. When viewed from the rear, the voltage adjustments are under the plastic cover on the right end of the Trio Labs power supply for the 9100/9200 displays. They are on the left end of the Electro-Vector supplies used on the 9300.

2.9 CONTROLS AND INDICATORS

The 9000 Series Systems may be operated in one of two modes as follows:

a. Display operate mode.
b. Diagnostic mode.

In the display operate mode, the 9000 will erase the screen and operate as a graphic display system when power is applied. In the diagnostic mode, the 8080 will jump to software contained on the diagnostic test board (an optional test device), when power is turned ON or the reset switch is depressed. The switch which controls this mode selection is contained on the edge of the control board behind the front panel as shown below:



If all cables are properly connected and all boards plugged in properly, the system power switch may be turned ON. When turning on the system, the self test light will flash on and then go off as the screen is erased in the display operate mode. If the reset switch is depressed on the control panel the self test light will illuminate for as long as the reset switch is depressed and extinguish when the reset switch is released. The self test lamp going off indicates that the 8080 micorprocessor is operating properly and that all the PROMS which contain 8080 software contain good bit patterns. The self test does not indicate the state of the total system. For instance, self test does not indicate the state of display refresh memory or the display generator. It is indicative that the system is not "dead" and that the user may proceed. When the screen erases, the user can be more certain of the display generator and refresh memory.

If the system does not turn on as described above, the user may wish to recheck the CPU, CPU cabling, and make sure an interface board is inserted into Slot 1. If the interface board is not present the control card will not be "selected" and will not operate. The LED power indicator on the control panel is driven by +5 VDC and thus does not indicate the operating voltage levels, it simply indicates that power is present in the system.

If the system fails to operate properly but passes self test, the diagnostic board can be used to isolate the failure. If the display fails to pass self test, the 8080 may not be able to execute any software and it will most likely be necessary to exchange the control board, extract video, serial link, or memory expansion boards which may be forcing the control board to malfunction, or utilize a Ramtek model MM80 In-Circuit Emulator to isolate the problem. The MM80 In-Circuit Emulator requires knowledge of the control board but will always successfully find the problem even if the 8080 is totally inoperative for any reason. The MM80 would generally be employed at the factory or depot level, while board swapping or the diagnostic test board would be utilized in the field.

2.10 ACCEPTANCE TEST HOST-CPU PROGRAMS

The 9000 Series Display System is capable of being interfaced to virtually any computer currently available. Because all CPU are different in terms of instruction set, memory addressing, register definitions, etc., it is necessary to build a set of programs for each CPU and display system interface. These programs provide a mechanism for verifying that the communication lines between the CPU mainframe and the 9000 Series Display System are operational. The programs which must be built for each CPU can be divided into the following categories:

1. A program to read the instruction test data sets and transmit the tests to the 9000 Series display processor via the computer interface.

2. A program (or set of programs) to write, readback, and verify data between the CPU and the 9000 Series display processor.

3. A program (or set of programs) to check interface functions which are not merely data I/O functions (e.g. system reset, interrupt generation and/or disabling of interrupts, etc.).

Ramtek supplies the instruction test data sets which are CPU independent. These tests are essential in testing the total system for operability. They are also very useful for diagnosing a faulty system provided the interface can, at least, write to the RM-9000 and the 8080 is operating correctly. In most instances, the boards can be isolated and exchanged without the need for any other type of diagnostic aid.

2.11 DIAGNOSTIC AIDS

Two levels of diagnostic aids are available for trouble-shooting a faulty system as follows:

a. Diagnostic tester board.
b. Model MM80 In-Circuit Emulator

The diagnostic tester board is a PC board with small keyboard which inserts into any video slot. It contains a 12 position selector switch for selecting one "direct through" memory plane at a time for display on a monitor. This makes it easy to isolate faulty memory chips since "bad bits" appear as black and white, rather than grey scale. This board also contains diagnostic software PROMs and RAM which when used in conjunction with the control board 8080 circuitry can execute all the diagnostics. This board is a must for field service of 9000 Series Systems.

The MM80 is a complete software and hardware development system for the 8080 microprocessor. Its in-circuit emulator capability is essential for debugging problems that cause the 8080 to not operate as a CPU. Like the diagnostic board, it can test every part of the system execpt the interface board.

2.12 REPACKING FOR SHIPMENT

The following paragraphs contain a general guide for repacking of the 9000 Series System chassis for shipment. It is recommended that the original packing material be retained. Instructions are given if original or new packing is to be used. Before any material is returned to Ramtek a RMA (Returned Material Authorization) must be obtained from Ramtek Customer Service Department.

NOTE

If the equipment is to be shipped to Ramtek for service or repair, attach a tag to the chassis identifying the owner and indicate the service or repair to be accomplished. Include the RMA #, model number and full serial number of the equipment. In any correspondence, identify the equipment by RMA #, model number and serial number.

If the original container is to be used, proceed as follows:

a. Place chassis in original container as previously packed.
b. Ensure that container is well sealed with strong tape or metal bands.

If the original container is not available, proceed as follows:

a. Wrap chassis in heavy paper or plastic before placing in an inner container.
b. Place packing material around all sides of chassis and protect connectors and panel face with cardboard strips.
c. Place equipment and inner container in heavy carton or wood box and seal with strong tape or metal bands.
d. Mark shipping container with "DELICATE INSTRUMENT", "FRAGILE", etc.

3.0 THEORY OF OPERATION

The block diagram shown in Figure 3-1 summarizes the major sections which comprise the 9000 Series architecture. As shown, the host CPU communicates through a standard Ramtek interface. A standard bus structure called the Internal Data Exchange (IDE), has been defined for previous Ramtek display products. It allows the CPU to be interconnected to as many as eight major sections as defined by the eight select lines shown in Figure 3-1. The 9000 Series control board is considered one of the sections. Thus, one interface could potentially be wired to more than one 9000 Series controller. The Read, Write and Access lines cause the 16 bit transfer to occur over the 16 bit bidirectional data bus. These lines are shared by each section which communicate with the interface. The acknowledge line acts to synchronize handshake sequences during the data transfer. Six interrupt lines are available for interrupting the host CPU. The 9000 Series control board utilizes four of the six interrupts. It does not utilize the access line or interrupt acknowledge. CPU reset is used to reset the entire controller under software control or when depressing a reset switch on a host minicomputer. The 8080 microprocessor software can poll the write handshake line to determine when a 16 bit data word is available. When the 8080 reads or writes a 16 bit data value, the acknowledge line is automatically initiated. The interrupt lines are peripheral I/O bits to the 8080 microprocessor. Typically, the 8080 executes software from the 2K byte ROM, which causes it to poll the write handshake line as well as the serial peripheral device activity bits. If no activity has occurred, the 8080 simply continues to scan this status information. The interface parallel link portion of the control board shown in Figure 3-1 contains driver/receiver modules which interconnect the IDE bus with the Internal Processor (IP) bus. When the 8080 microprocessor receives a user level instruction from the host CPU, it will decode the instruction under software control and determine how to operate on the 16 bit words which follow. In the case of image data, the 8080 microprocessor would normally load the X current operating point and Y current operating point registers contained in the display generator. These registers determine where the writing will begin on the screen. After the 8080 microprocessor has initialized the necessary refresh memory

address registers, it will cause a DMA transfer to occur between the interface parallel link and the display generator refresh memory data registers. These DMA transfers can occur at a rate of one 16 bit word every 1.5 μsec. If the 8080 microprocessor receives a user command which causes alphanumeric characters to be displayed on the screen, it will receive the 8 bit ASCII character and utilize its character generator ROM to determine the related font. This font information is then loaded under software control into the raster data register contained in the display generator. The 8080 microprocessor may write up to 8 bits of font per transfer. Between transfers it will load the X and Y current operating point registers to move to the next line of font. The 8080 microprocessor may also utilize the static RAM or dynamic RAM on the memory expansion board as a lookup table for programmable font. When reading image data back to the host CPU, the 8080 microprocessor will set up a DMA transfer between the display generator readback register and the interface parallel link. These DMA transfers are internal to the control board and are not related to DMA transfers which may occur between the host CPU and the standard interface boards. When the 8080 microprocessor receives scroll instructions from the host CPU, it will convert the commands to the necessary codes which it loads into the X origin and Y origin registers contained in the display generator. The display generators memory plane select register determines which refresh memory boards are enabled during these writing processes. The screen refresh address register contained in the display generator is utilized to address the refresh memory boards for the purpose of driving data to the video generator boards. Thus, the addresses used for reading and writing refresh memory are independent of screen refresh address. Reading and writing may occur in any of eight possible directions as controlled by the control and update registers contained in the refresh memory control logic of the display generator. Note that the display generator logic on the control board consists of three major sections: the refresh memory control logic, the refresh memory data registers, and the refresh memory address registers. These three major sections drive the memory input bus shown in Figure 3-1. This bus consists of 16 memory address lines, 12 memory plane select lines, 12 I/O data lines and six memory handshake lines. One memory I/O data line is connected to each refresh memory plane. Each refresh memory

board can contain up to six 256 x 320 sections per board. Each model of the 9000 Series Display Systems can contain up to 12 planes of memory for its related resolution. The refresh memory boards drive a memory output bus as shown in Figure 3-1. This refresh memory data consists of 12 bits, 1 bit for each plane for the 9100; 24 bits, 2 bits for each plane for the 9200; and 48 bits, 4 bits for each plane for the 9300. The data rate on these lines is 6.5 MHz regardless of the resolution of the system. Up to two video boards may be contained in a 9100 or 9200 and up to four video boards may be contained in a 9300. All video boards receive all 12 planes of information. These video boards also receive video sync timing signals from the sync timing board contained in the back of each of the 9000 Series chassis. These video sync lines are mixed with the memory plane data in an appropriate fashion and result in composite video data which is in turn driven to the video monitors. This data is RS-170 compatible video. Note that the internal processor bus is also routed to all of the video generator boards. Certain video generator board designs contain video lookup tables which can be loaded under 8080 micorprocessor software control to map the refresh memory data to pseudo color or new grey scale values. The 9000 controllers may contain up to two serial link boards. Each serial link board contains up to two cursor generators which are driven to all of the video boards. As shown in Figure 3-1 each serial link board contains four Universal Synchronous Receiver Transmitter (UART) modules. Each of the UARTs may communicate over serial transmission lines to keyboards or joystick cursor control devices. Each serial link board may communicate with up to four keyboards, or two keyboards and two joysticks. UART1 and UART2 are clocked by separate baud rate generation circuitry from UART Number 3 and 4. The operation of the serial link board is controlled by the 8080 microprocessor software. When data transfers are not occurring within the system the 8080 polls the serial link boards on a continuous basis to determine if serial link activity has occurred. When serial link activity occurs, the 8080 microprocessor software processes the 8 bit values from the UARTs and stores or buffers the information in static RAM on the control board. Under software control, the 8080 interrupts the interface by loading peripheral device registers which drive one of four interrupt lines to the interface. The option software which allows the serial link

board to process these characters is contained on the memory expansion board in the option ROM. The option ROM may also contain other options for scaling and drawing vectors and conics.

3.1 GENERAL SEQUENCER THEORY

As previously discussed, there are three levels of programming that exist within a 9000 Series System. The third level as described is firmware that exists in PROMs that form part of a circuit called a sequencer. These sequencers are microprogrammed controllers that are used to perform hardware functions that require more speed than the 8080A-1 is capable of providing.

A generalized configuration block diagram of a sequencer as used in the 9000 Series Systems is shown in Figure 3-2. The sequencer is basically a simplified, but specialized computer that is limited to the functions of testing conditions to control the program flow and providing control signals that are used to control the operation of some specific hardware. Hence, a sequencer may be programmed to perform in some prescribed manner a specialized function(s). Thus, this sequencer is a microprogrammed decive.

In Figure 3-2 the block labled CONTROL PROGRAM STORE is the program memory containing the microprogram in PROM. The block labled ADDRESS REGISTER is analogous to the program counter of a computer. It contains the current address of the instruction that is being executed. The block labled TEST CONDITION SELECTOR is a selector that determines which of a given set of conditions is the one that is currently being tested. There are two methods of providing control to external circuitry as indicated in the block diagram, DIRECT CONTROL or DECODED CONTROL.

The "character" of a sequencer is determined by the microprogram that resides in its CONTROL PROGRAM STORE. The instruction format is as shown below.

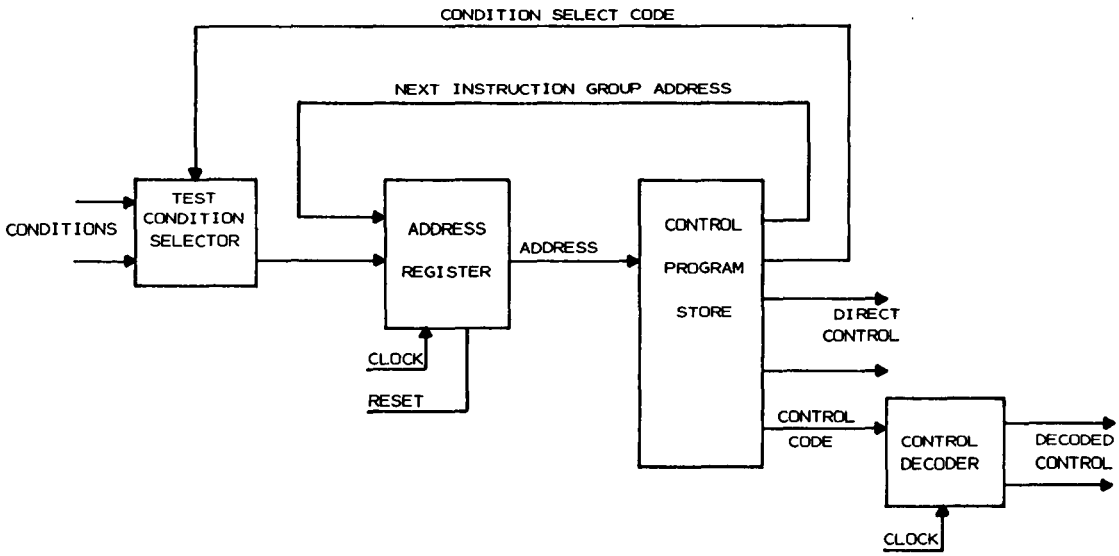


FIGURE 3-2
GENERAL SEQUENCER
CONFIGURATION

NEXT INSTRUCTION GROUP ADDRESS	CONDITION SELECT CODE	DIRECT CONTROL	DECODED CONTROL

The next instruction group address is a set of bits that determine the most significant bits of the address of the next instruction to be executed, thus specifying the group of instructions in which it resides. The condition select code specifies a condition or set of conditions that are used to determine the least significant bit or bits of the address of the next instruction to be executed. At this time, the address of the next instruction to be executed is totally specified dependent upon the state of the specified condition or set of conditions. Thus the sequencer has a condition jump capability to control the program flow. Unconditional jump capability is also present as a degenerate case of the conditional jump capability. Finally, the direct control and decoded control portions of the instruction provide the desired control to the external circuitry. Direct control uses bits out of the CONTROL PROGRAM STORE directly to perform control functions. Decoded control uses bits that are decoded in some manner to provide control functions. The major difference between these two methods of control are that direct control can provide multiple, simultaneous, control functions (as many as there are bits used) while decoded control, though more bit efficient, can only provide one control function at a time per decoder. Both methods may be used in combination.

As a micorprogram within the CONTROL PROGRAM STORE is executed, each instruction currently executing specifies the necessary information to determine the next instruction that will be executed and may or may not, as required, provide control information to the external circuitry.

A sequencer configuration that is more typical of the ones that are used in the 9000 Series Systems is shown in Figure 3-3. This sequencer has an instruction format that consists of 20 bits of information as shown below.

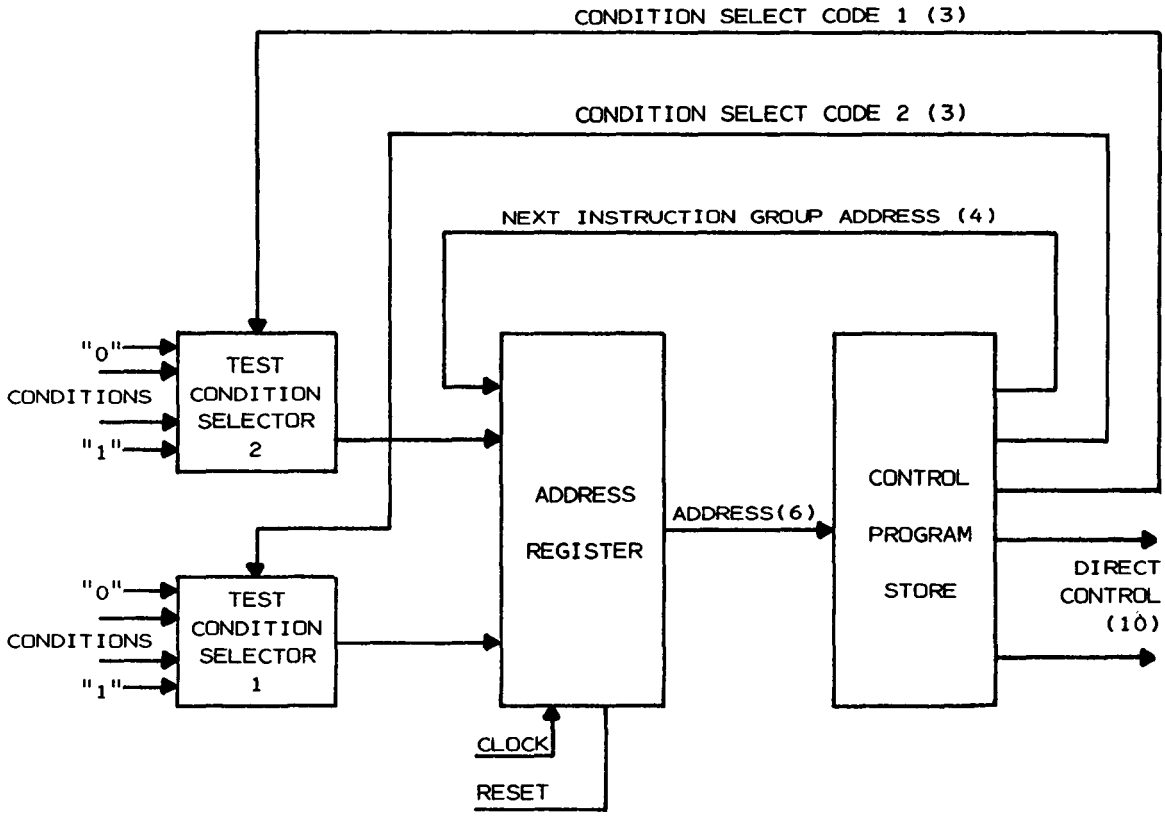
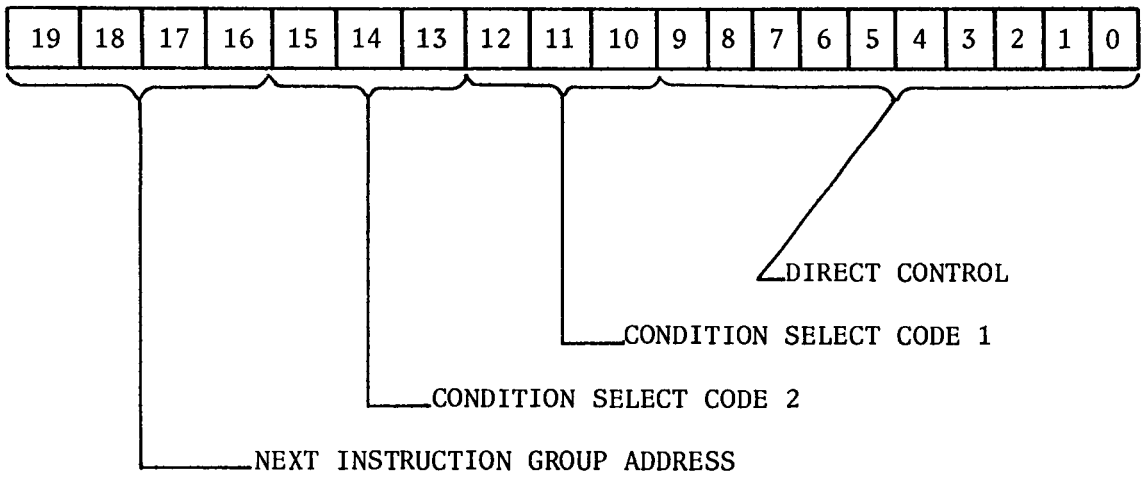


FIGURE 3-3
TYPICAL SEQUENCER
CONFIGURATION ON THE
CONTROL BOARD



The NEXT INSTRUCTION GROUP ADDRESS is 4 bits in width, and specifies a group of four instructions, one of which will be the next instruction to be executed. The CONDITION SELECT CODES 1 and 2 are applied to TEST CONDITION SELECTORS 1 and 2 respectively (each is a 3 bit code). The codes select a set of two conditions whose states are used directly to determine the least significant 2 bits of the next instruction address. Thus one of the four instructions specified by the NEXT INSTRUCTION GROUP ADDRESS is selected as the next instruction to be executed. Further, the next instruction address is 6 bits in width, defining the word length of the CONTROL PROGRAM STORE to be 64 words, or instructions.

Note also, that there are 16 possible conditions that are tested in pairs to control the flow of the microprogram. Four of the 16, two from each 1-of-8 selector, are given over to implement the unconditional jump capability by providing guaranteed states of "1" and "0" on each TEST CONDITION SELECTOR.

The next instruction address is applied to the input of the ADDRESS REGISTER and is loaded to become the current instruction address at the positive transition of the clock input of the ADDRESS REGISTER. The sequencers in the 9000 Series Systems operate on a 153 ns clock cycle time.

At power ON or CPU reset time a RESET signal is applied to the ADDRESS REGISTER to provide an initial starting place for the microprogram which is address 000000 binary.

Finally the sequencer of Figure 3-3 provides 10 bits of direct control. The pulse width or level of these control signals is a direct function of how the microprogram is written. In the 9000 Series Systems pulse widths will be in multiples of the 153 ns clock cycles.

Thus, the sequencer shown in Figure 3-3 can provide a very powerful, neat and orderly control circuit that can be implemented with only eight 16 pin MSI ICs.

4.0 INTERFACE BOARD AND INTERNAL DATA EXCHANGE BUS

The commonality between most interface assemblies used in the 9000 Series may be best illustrated by discussing the 9000 Series General Purpose Interface (GPIF) which provides a simple, generalized method of interfacing a variety of computers to the 9000 Series Systems. This includes computers manufactured by Texas Instruments, Honeywell, Data General and Interdata. The GPIF includes the necessary command/status and data registers, handshake control circuitry and interrupt control circuitry normally required in any computer interface to the 9000 Series. This minimizes the amount of logic design required to implement an interface to a particular CPU.

Physically, the GPIF consists of a single plug-in board assembly, mounted in a 9000 Series chassis, which communicates with the CPU through a single back-panel connector using 22 signal pairs.

Figure 4-1 depicts the relative position of the GPIF within a typical computer and a 9000 Series System configuration. User designed logic usually will reside in the CPU. In some cases where the computer configuration does not allow for this, custom designed logic will be placed on the GPIF. In this situation the user defines his own signal and cable requirements and merges his logic to the GPIF, by-passing the GPIF drivers and receivers.

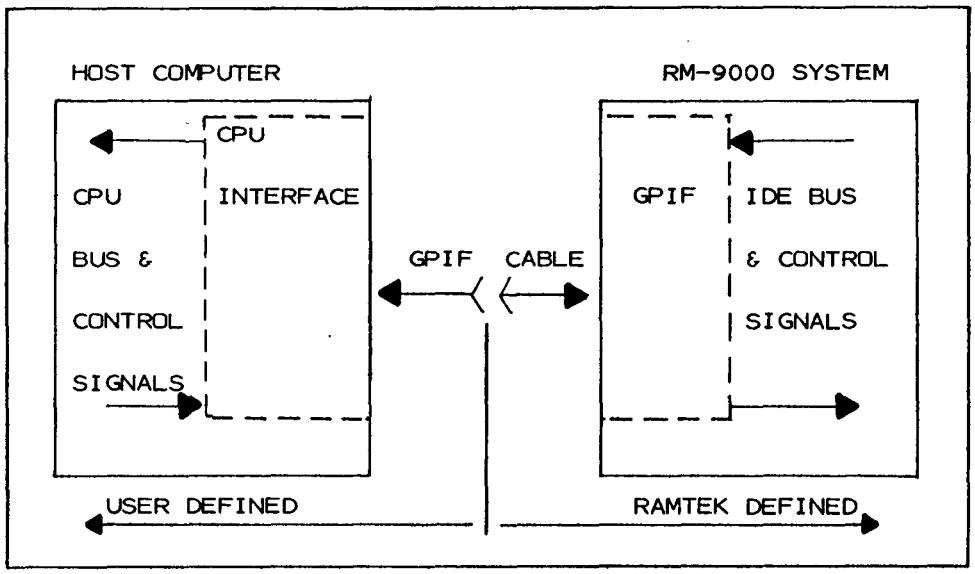


FIGURE 4-1 GPIF WITHIN A COMPUTER/9000 SYSTEM

4.1 GENERAL PURPOSE INTERFACE SIMPLIFIED BLOCK DIAGRAM

The simplified block diagram in Figure 4-2 shows 22 signal lines to/from the CPU. All are negative true (complement logic), the 16 data lines (IØBIT00-15) are bidirectional, the remaining six lines are undirectional control lines. A description of these lines and term names is as follows:

IØBIT00-IØBIT15. Bidirectional, data lines, IØBIT00 is the least significant bit (LSB).

IØWRITE. CPU to display, when true (0 VDC), causes the contents of IØBIT00-IØBIT15 to be loaded into the GPIF command register or output data register, as determined by IØCMD. IØWRITE should not be asserted simultaneously with IØREAD.

IØREAD. CPU to display, when true (0 VDCY) causes GPIF to place contents of the status register or input data register as determined by IØCMD onto IØBIT00-IØBIT15 for reading by the CPU logic. This signal should not be asserted simultaneously with IØWRITE.

IØCMD. CPU to display, during an IØWRITE or IØREAD operation, when true (0 VDC) indicates that the command or status register is selected. When false (+5 VDC) indicates that the output or input data register is selected.

IØREADY. Display to CPU, when false (+5 VDC), indicates that the GPIF is busy transferring data within the display system and no data transfers between the CPU and the GPIF should be initiated. (Exception: reading the status register is allowed.)

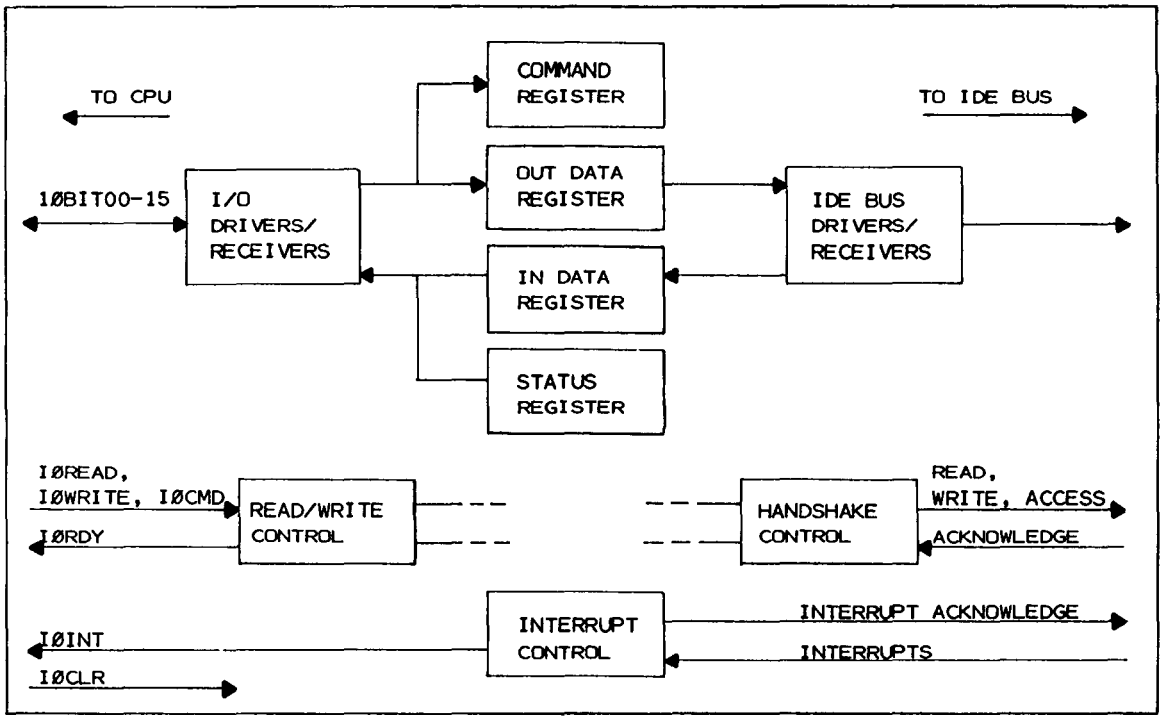


FIGURE 4-2 GPIF SIMPLIFIED BLOCK DIAGRAM

IØINT. Display to CPU, interrupt or service request to CPU. This signal is true (0 VDC) when any of the interrupt bits of the status register become set if the corresponding enable bit is set in the command register. IØINT remains true until the status register is read. (Exception: the DONE interrupt bit, set by the rising edge of IØREADY will also be reset automatically whenever a read or write data operation or an access cycle is initiated.)

IØCLR. CUP to display, master reset. When true (0 VDC) resets the command and data registers and issues a reset to the entire 9000 Series System.

DRIVERS AND RECEIVERS

The GPIF uses tri-state line driver/receiver ICs (typically Signetics type 8T26) for the 22 interface signals. All signals are terminated by 220 ohms to +5 volts and 330 ohms to ground. The cable to the CPU is usually comprised of twisted pairs. The same line driver/receivers and termination are usually used at the CPU. Cable length is normally in the 10-12 foot range. The maximum cable length is determined by the signal timing requirements of the CPU mounted interface.

I/O TIMING

The GPIF is relatively timing independent; leading and/or trailing edges of control signals are synchronized to an internal system clock before use and pulse widths are not critical.

The I/O write timing is shown in Figure 4-3. The IØBIT00-IØBIT15 and IØCMD should be stable $\geq$ 100 ns before IØWRITE is asserted, and should remain stable $\geq$ 300 ns after IØWRITE is asserted. IØWRITE should be $\geq$ 300 ns. It is not necessary for IØWRITE and IØBIT00 IØBIT15 to be coincident on the trailing edge, only the leading edge timing is important.

The I/O read timing is shown in Figure 4-4. The GPIF synchronizes IØREAD to an internal clock then gates the contents of the selected register (status or data) onto IØBIT00-IØBIT15. Allowing for this synchronization plus gate delays and settling time, the CPU logic should wait $\geq$ 250 ns after the assertion of IØREAD before accepting the data. There is no maximum pulse width. The data will be stable as long as IØREAD is true.

During the time IØREAD is true (low) and for a minimum of 250 ns after the release of IØREAD, IØCMD must remain true (low). (If the state of IØCMD is simply left unchanged until the start of another IØREAD or IØWRITE pulse, this latter timing requirement should pre-

sent no problem.) The minimum reset (IØCLR) timing pulse width is 500 ns.

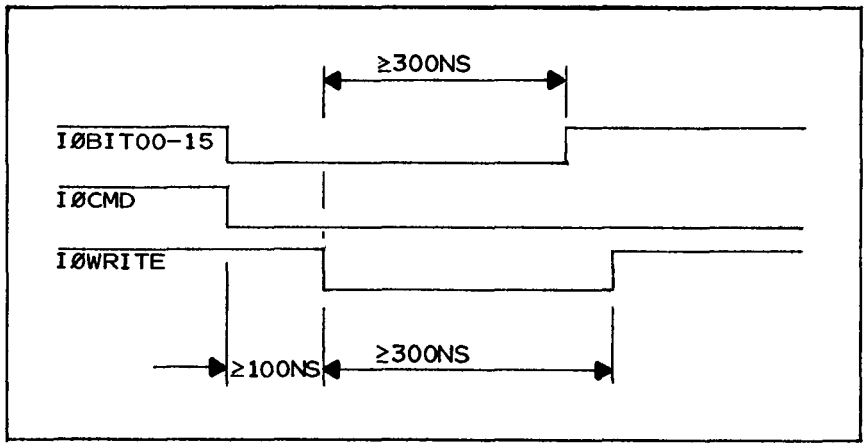


FIGURE 4-3 I/O WRITE TIMING

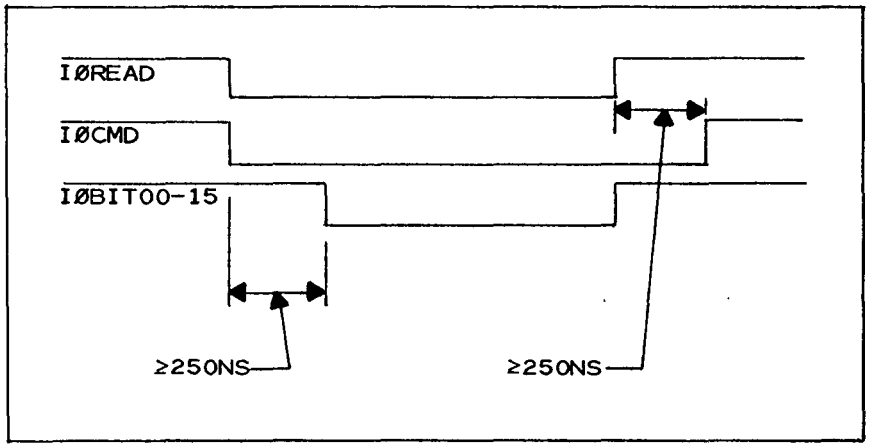


FIGURE 4-4 I/O READ TIMING

4.2 TRANSFER TIMING

To illustrate the minimum single-word data transfer time that can be achieved, refer to Figure 4-5. *Note that this example deals with a single transfer and should not be used to calculate maximum throughput rates.* The GPIF may have to wait for the IDE bus to be free before data can be transferred.

It is assumed that the IDE bus is not busy at the start of the operation and that the selected system block will accept data at its highest rate.

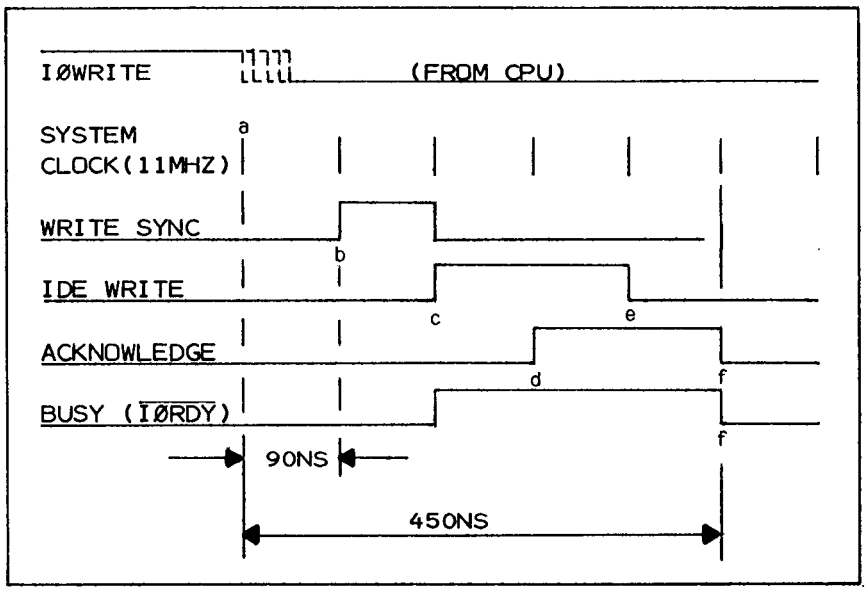


FIGURE 4-5 TRANSFER TIMING

The sequence illustrated in Figure 4-5 is as follows:

a. IØWRITE is enabled (asynchronous to system clock).
b. The GPIF recognizes IØWRITE, loads the output data register, and prepares to start an internal write operation.
c. A write signal is asserted to the 9000 Series System, IØRDY goes false.

d. The system acknowledges the write signal.

e. The GPIF releases the write signal.

f. The system releases the acknowledge signal. IØRDY goes true. The GPIF could now begin another cycle.

With an 11 MHz system clock, 5 x 90 or 450 nanoseconds have elapsed since the insertion of IØWRITE.

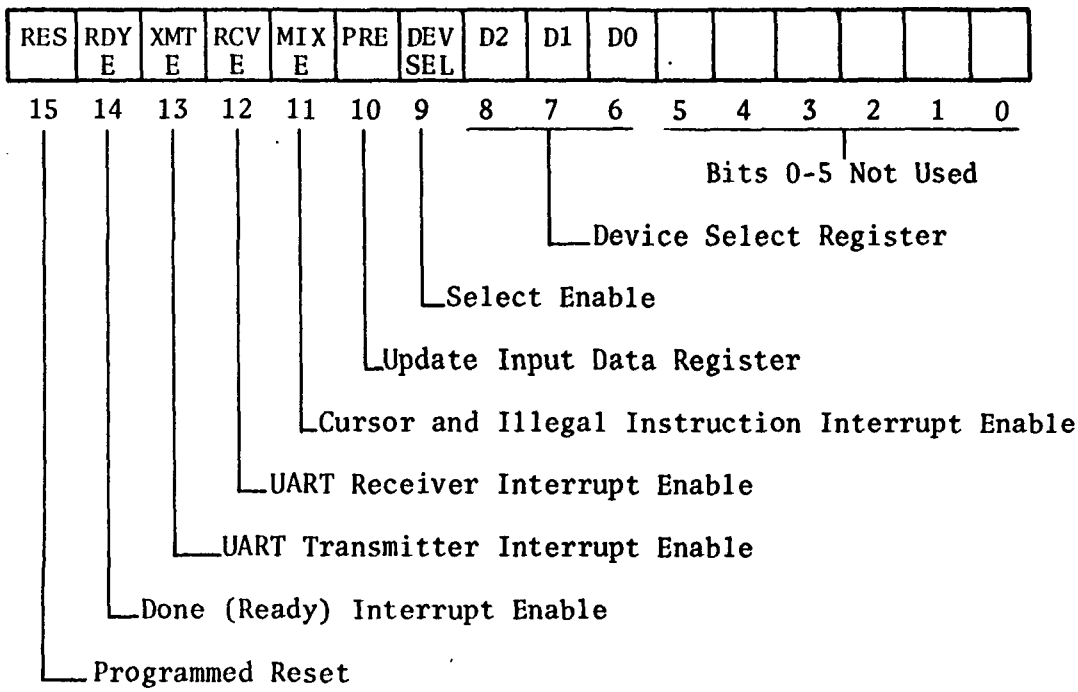
Figure 4-5 can also represent the minimum time for a read operation. The GPIF synchronizes to the trailing edge of IØREAD and using the same waveform diagram but replacing "IDE write" with "IDE read", again, a minimum of 450 nanoseconds elapses before the GPIF has updated the read data register and is ready for interrogation by the CPU.

4.3 GPIF COMMAND AND STATUS

The GPIF provides a command and a status register. The command register contains address, interrupt enables and command bits. The status register consists of certain command register bits (environment) as well as interrupt and status bits (status). The bit assignments are shown in Tables 4-1 and 4-2. *Note that several of the bits are used only in other Ramtek Display Products and are not used in the 9000 Series Systems.*

A detailed description of each bit in the command register, Table 4-1, is as follows:

TABLE 4-1 COMMAND REGISTER BIT ASSIGNMENTS



a. Bit 15 (RES). Programmed reset. Issues a reset signal to GPIF and display system. Ignores all other command register bits.

b. Bit 14 (RDY E). Done interrupt enable. Enables leading edge of READY to cause an interrupt, thus signalling that the GPIF has completed an operation. This may be considered to be an interrupt for the GPIF, as compared to XMT E, RCV E and MIX E below.

c. Bits 13-11 (XMT E, RCV E, MIX E). Interrupt enables. XMT E is UART transmitter interrupt enable, RCV E is UART receiver interrupt enable and MIX E is cursor and illegal instruction interrupt enable. "1" enables interrupts and "0" disables interrupts.

d. Bit (PRE). Update input data register. Usually used with DEV SEL (bit 9) before an anticipated read data operation. When PRE = "1", a read sequence is initiated (internal to the 9000 Series System) to the (newly) selected section so that the data register may be read with current data. Thereafter, after every read data operation, the interface will automatically fetch a new data word to the input data register. If a new section is selected and PRE = "0", a read data operation may still be performed but the contents of the first data word which is read will be the old contents of the read data register. The completion of a read data operation always causes a new internal read cycle to be initiated. When PRE is true it causes IØREADY to go false while the update operation is performed.

e. Bit 9 (DEV SEL). Select enable. When DEV SEL = "1", the contents of bits 8-6 are loaded into the device select register, thus selecting a new device. When DEV SEL = "0", bits 8-6 are ignored.

f. Bits 8-6 (D2, D1, D0). Device select register. D2, D1 and D0 select the various internal sections of the 9000 Series System. The code assignments are as follows:

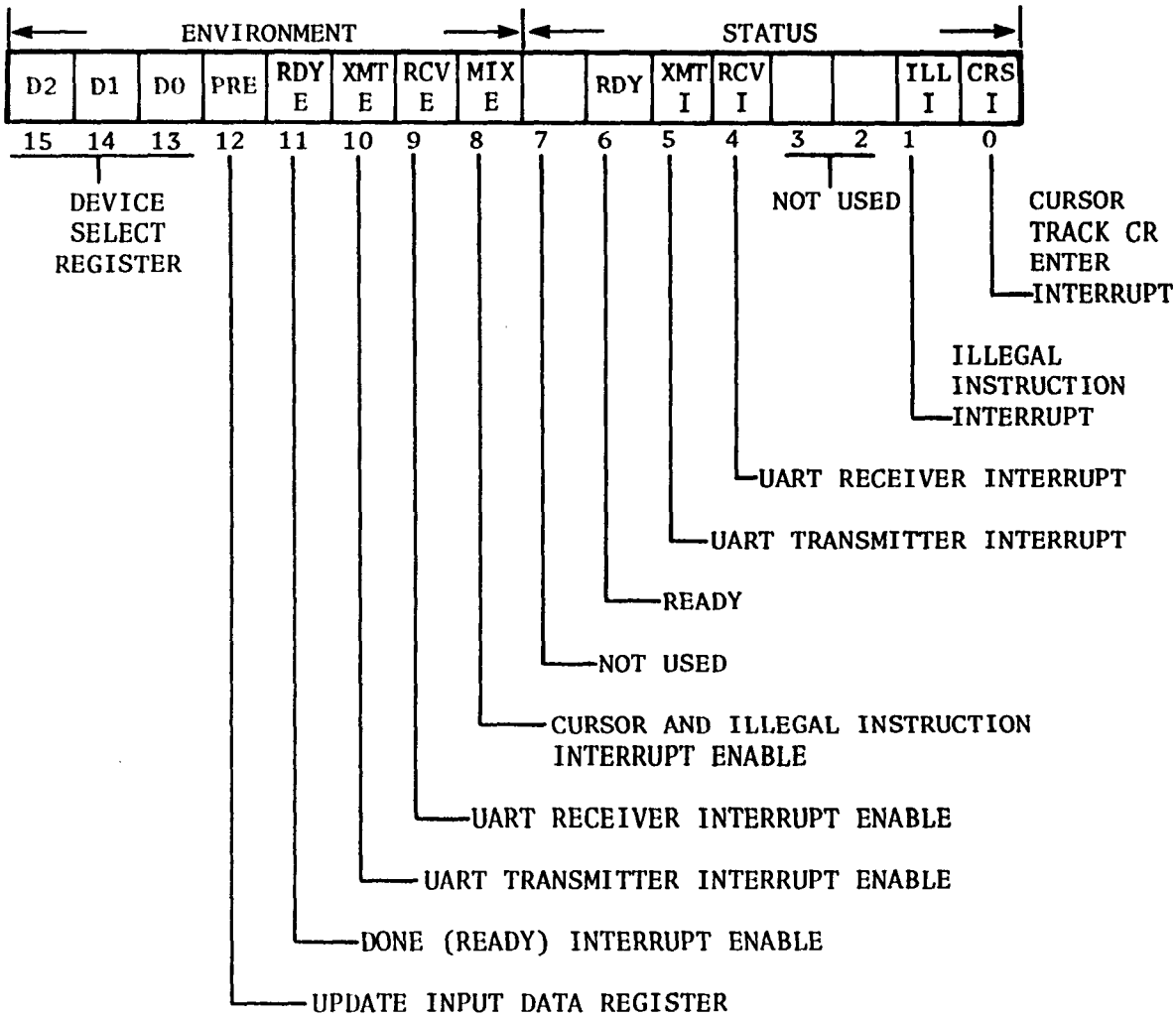
Bits

8	7	6		
0	0	0	-	9000 Controller
0	0	1	-	Unassigned
0	1	0	-	Unassigned
0	1	1	-	Unassigned
1	0	0	-	Unassigned
1	0	1	-	Unassigned
1	1	0	-	Unassigned
1	1	1	-	Unassigned

Bits 0-5 not used, but should always be set to logical "0".

A detailed description of each bit in the status register, Table 4-2, is as follows:

TABLE 4-2 STATUS REGISTER BIT ASSIGNMENTS



a. Bits 15-13 (D2, D1, D0). Device select register. Bits 8-6 from the command register.

b. Bit 12 (PRE). Update input data register. Bit 10 from the command register.

c. Bit 11 (RDY E). Done (ready) interrupt enable. Bit 14 from the command register.

d. Bits 10-8 (XMT E, RCV E, MIX E). Interrupt enables. Bit 13-11 from the command register.

e. Bit 7. Not used.

f. Bit 6 (RDY) Ready. Bit 6 = "0" indicates that the system is busy (not ready) with an internal operation and will not respond to data transfers. RDY will equal "0" for a period of time after each write data or read data operation, and after a write command operation in which PRE = "1". Bit 6 = "1" indicates that data may be transferred.

g. Bit 5 (XMT I). UART transmitter interrupt.

h. Bit 4 (RCV I). UART receiver interrupt.

i. Bits 2, 3. Not used.

j. Bit 1 (ILL I). Illegal instruction interrupt from control card.

k. Bit 0 (CRS I). Cursor track or enter interrupt.

4.4 DETAILED DESCRIPTION - GENERAL PURPOSE INTERFACE

Refer to Drawing 502029. Sheet 1 is a block diagram, sheet 2 is a timing diagram and sheets 3-7 are logic schematics. Sections of the block diagram are labled with sheet numbers indicating where the logic for a particular block is located.

The "I/Ø bus" (IØBIT00-*IØBIT15) refers to the bus between the interface and the CPU, while the "IDE" (9CDAT00-9CDAT15) refers to the Internal Data Exchange bus between the interface and the display system. Both busses are tri-state, negative-true and bi-directional.

Refer to drawing 502029-3. The I/Ø bus receivers are shown at locations 8G, 8H, 8J, 8K. The command register (locations 6G, 6J, 6K) is loaded with the I/Ø bus write data (WRBIT00-WRBIT15) by signal WRTCMD. WRTCMD

derives from signal *IØWRT from the CPU which, as signal 9IØWRT, is synchronized to the rest of the system by the 6.5 MHz clock (PØSCLK) producing signals *WRT2 (location 5L). Signal IØCMD from the CPU indicates whether an I/Ø transaction is a command/status or a data transaction. (IØCMD = logic "0" indicates command/status, IØCMD = logic "0" indicates data.) Algebraically this may be expressed as the following Boolean equation:

$$\text{WRTCMD then} = \text{IØCMD} \cdot \overline{\text{*WRT1}} \cdot \text{*WRT2}$$

Since *WRT2 is equivalent to *WRT1 delayed by one cycle of PØSCLK, WRTCMD (location 6M) is a positive pulse equal in duration to one 153 ns clock period (see timing diagram). WRTDATA is derived in a manner similar to WRTCMD except that IØCMD is a logic "0". The Boolean equation is as follows:

$$\text{WRTDATA} = \overline{\text{IØCMD}} \cdot \overline{\text{*WRT1}} \cdot \text{*WRT2}$$

Portions of the command register are selectively loaded depending on the state of certain bits in the command word. For instance, loading of the device select bits, (DSEL1-DSEL4, location 4J) is controlled by the state of WRBIT09, which is itself loaded by WRTCMD to generate the select enable signal (SELENA). The remainder of the command register is generated from WRBIT10-WRBIT15 loaded by WRTCMD (location 6K).

The device select bits are decoded by a three-to-eight decoder (location 6B) to produce signals *SEL0-*SEL7. These signals are synchronized to IDE bus transactions by signals *ACCESS, *IDEWRT and IDEREAD (location 3C). These signals generate the load signal (SELCLK) which loads two registers, the outputs of which are the actual system select lines (9CSEL0-9CSEL7). The logical AND of internal clear signal (INTCLR) and PØCLK insure that after a reset operation (especially a power-on-clear), multiple select lines are not asserted. The logical complement of INTCLR (*INCLR) resets the device selects bits to "0", causing *SEL0 to be a logic "0", which is then loaded by PØSCLK to cause only 9CSEL0 to be asserted.

If the command register is loaded with WRBIT15 set, a programmed clear operation is initiated. WRBIT15 and WRTCMD generate signal *DISCØN (location 7M), resetting a binary counter (Loc 6H) by means of its load input (loading all zeros). PØSCLK will increment the counter until a maximum count of 15 is reached, at which time the "max" signal (*PROGCLR) will go to a logic "1", disabling further counting. *PRØGCLR will then remain at a logic "1" until another *DISCØN signal is received. The "max" signal (*PRØGCLR) thus serves as the intended programmed clear signal. Note that during a power-on sequence, regardless of the state of the binary counter, it will reset itself in a few clock cycles, all the while generating an additional, and redundant, reset signal.

Refer to drawing 502029-4. The status register consists of bits from throughout the interface, including the command register and interrupt flags. These are selected for reading to the CPU by signal 91ØCMD (locations 4G, 4H, 4J, 4K). 91ØCMD = "0" selects the status word, 91ØCMD = "1" selects the input data register (discussed later). The output of the selectors is gated to the I/Ø bus by signal DRD1 (locations 8G, 8H, 8J, 8K). DRD1 derives from signal *IØREAD synchronized with the 6.5 MHz clock (locations 7A, 8A on drawing 501948-6) and will be discussed later.

Read data (RDAT00-RDAT15) from the IDE bus is loaded into the input data register (locations 4E, 4F, 5F) by signal *IDEREAD. This signal also will be discussed later.

The loaded read data, LRDAT00-LRDAT15, is available by reading to the CPU in a manner similar to the status register. 91ØCMD = "1" selects the data and DRD1 gates it to the I/Ø bus.

The 14 MHz system clock is received by ECL to TTL converters (location 8A). Neither of the converted signals 9N14CLK or 9P14CLK are used by the interface.

Refer to drawing 502029-5. The output data register (locations 5G, 5J, 5K) is loaded with the I/Ø bus write data (WRBIT00-WRBIT15) by signal

WRTDATA. Read data (RDAT00-RDAT15) is received from the IDE bus. Data written to the IDE bus consists of the output data word (locations 5G, 5J, 5K). The data word is selected (location 5E) for the write operation by signal IDEWRT (location 7G). DATENA, gates the selected data onto the IDE bus (locations 6E, 6F, 7E, 7F).

Refer to drawing 502029-6. Much of the operation of any of the handshake controls is similar. One of them will be discussed in detail and the others only generally to point out how they differ. The handshake control circuitry is located on the right half of drawing 501948-6. It will be helpful to also refer to the timing diagram drawing 501948-2 for the following discussion. The handshake control circuits can be thought of as D-type flip-flops (actually J-K flip-flops wired to act like D flip-flops) following the transition of the acknowledge line (9CACK) by one clock cycle if properly initialized. The write handshake is initiated by WRTDATA (location 4D) which loads the output data register. The asserted signal to the display system is 9CWRITE (location 6A).

The read (9CREAD) handshake cycle (location 6A) can be initiated two ways. First, by signal DØNEDAT. DØNEDAT is input to a flip-flop (location 5A) which signifies that a read-data operation has been completed to the CPU, and the IDE read operation that is to be performed is to fetch the next data word in anticipation of another read-to-CPU sequence. Second, by signal JINRD3 when input to the flip-flop (location 4C) is algebraically derived as follows:

$$JINRD3 = WRTCMD \cdot PREFTCH$$

Thus, a read cycle will be initiated immediately upon setting the pre-fetch bit.

The three flip-flops (locations 4C, 5A) which are set by the two methods discussed above are logically OR gated (location 3C) to generate signal SETREAD which serves to control the common read flip-flop (location 4D) generating signal IDEREAD. The asserted signal to the display system is 9CREAD.

At any time that an IDE bus cycle is in progress or about to be initiated the ready signal to the CPU is not asserted. This AND gated (location 4M) signal is IØREADY and is described by the Boolean equation:

$$IØREADY = 2CACK \cdot SETACC \cdot SETWRT \cdot SETREAD \cdot INTCLR$$

The asserted signal (*IØREADY) to the CPU is negative-true. The 11 MHz system clock is received and buffered by ECL to TTL converters (location 8A). Signal NEGCLK is not used by the interface and is provided for reference and trouble shooting if necessary.

The signals DØNEDAT and DØNESTS indicate that the CPU has completed reading data or status, respectively. These signals (locations 4M, 6M) are generated by a circuit similar to that used to generate WRTCMD and WRTDATA, discussed previously. Both of these are positive going signals equal in width to one clock period (shown in timing diagram, drawing 502029-2) occurring at the end of a read cycle and are described by the following Boolean equations:

$$DØNEDAT = \overline{IØCMD} \cdot \overline{DRD1} \cdot DRD2$$

$$DØNESTS = \overline{9IØCMD} \cdot \overline{DRD1} \cdot DRD2$$

DRD1 and DRD2 are derived at flip-flops (location 5L) from IØREAD synchronized to PØSCLK.

The interrupt control circuits (location 6D) detect the rising edge of any of the 9000 Series System interrupt lines arriving at the interface. The "rising edge" refers to the signal level after it is received (location 8D) and inverted by the interface; the interrupt lines are actually negative-true. One example will be given as typical for all the interrupts. Refer also to the timing diagram drawing 502029-2.

All the interrupt lines are synchronous to the system clock. As 9MINT7 (location 8D) is asserted (9MINT = logic "0", MINT7 = logic "1"), then SINT7 (location 6C) will be true for one clock period until LMINT7 is

set at the leading edge flip-flop (location 6D). This allows *MPXINT7 to be set at the interrupt flip-flop (location 5D). The bit 0-3 interrupts are OR gated (location 5H) to produce a common interrupt (MPXINT). Then if the interrupt is enabled by the gating circuit (location 6L), MPXIENA must = logic "1", the interrupt line (IØINT) will be asserted to the CPU. *MPXINT7 is reset at the completion of a read-status operation by *DØNEST at the interrupt flip-flop (location 5D). The interrupts are disabled from being set during a CPU read operation by signal EXTREAD, (the logical OR of DRD1 and DRD2) which disables the input to the interrupt flip-flops (locations 5B, 5C, 5D). EXTREAD also gates the system clock (PØSCLK) to the leading edge flip-flops (location 6D), preventing a leading edge transition in progress from being detected until after the read operation.

Refer to drawing 502029-7. The ready interrupt signal (RDYINT) differs from the others only in that it is generated by conditions on the interface rather than from system interrupt lines from the rest of the display system. It operates in an entirely analogous fashion generating an interrupt to the CPU with the circuitry consisting of two flip-flops and gates (locations 3E, 3L, 3M, 4L), every time the ready line IØREADY rises. It thus signals that the system is "done" with an IDE bus transaction. The interrupt is enabled by signal RDYIENA (location 6L on drawing 501948).

Note that if a valid acknowledge had occurred, the fall of SELCLK would reset ASSERT in the flip-flop (location 4L) and clear the binary counter (location 7K) by means of its "load" input, pin 11.

4.5 DETAILED DESCRIPTION - INTERNAL DATA EXCHANGE BUS

Figure 4-6 shows the Internal Data Exchange bus (IDE) block diagram. Communication with all portions of the system is accomplished with a common set of asynchronous TTL handshake lines and a 16 bit bidirectional tri-state TTL bus. All data and handshake lines which form the IDE operate in a complement logic fashion (active = "0" = 0 VDC).

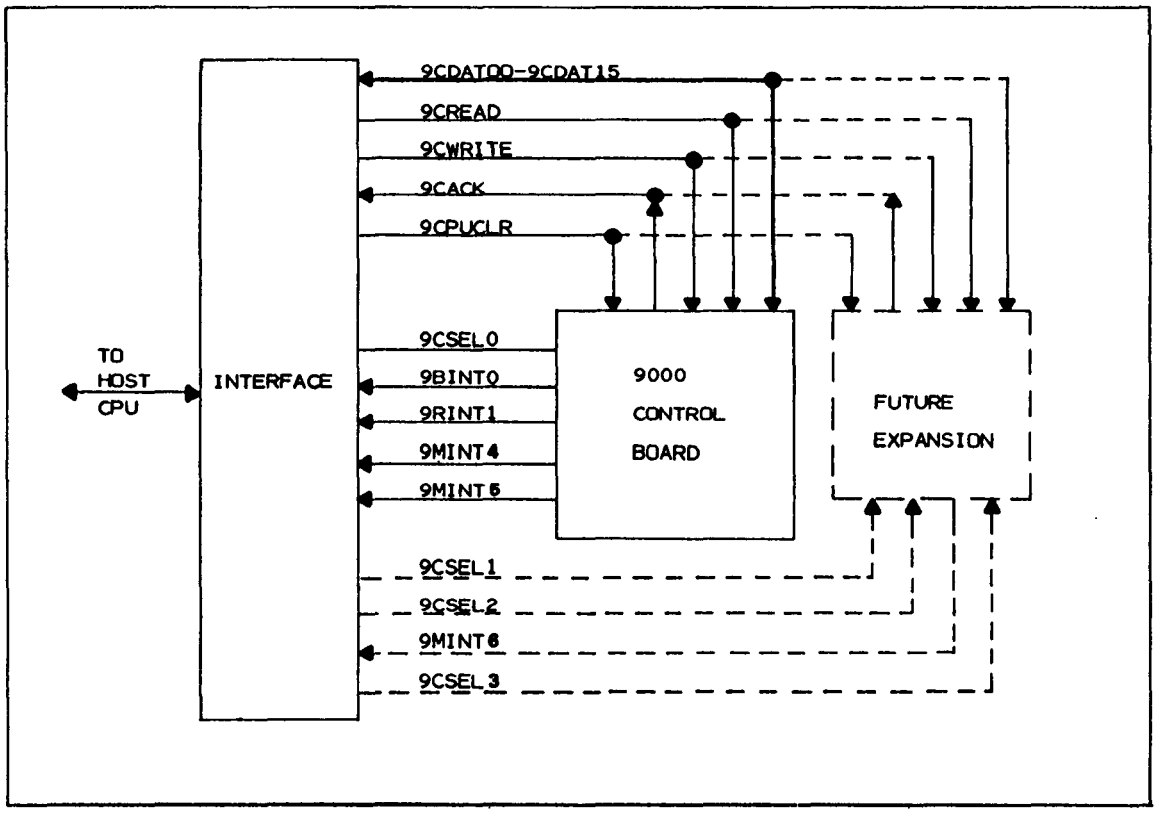


FIGURE 4-6 IDE BLOCK DIAGRAM

Each assembly connected to a unidirectional signal originating in the interface presents one standard TTL load to the signal. Unidirectional signals originating on other assemblies received by the interface are not loaded by the originating circuit(s). Therefore, the full drive capability of a standard TTL signal source is available for use by the interface.

The data exchange lines shown in Figure 4-6 are defined as follows:

DATA 0-15--9CDAT00-9CDAT15. Bidirectional, tristate, active low data bus bits 0-15 interconnect the interface and the 9000 Series System. Usually, 8T26 bidirectional driver/receiver modules are used when connecting to the IDE. The interface contains 470-ohm pullup resisters for line matching. Bit 0 is the LSB, Bit 15 is the MSB.

READ--9CREAD. Interface to the 9000 Series System, active low. The read handshake line (9CREAD = 0 VDC) indicates that the selected assembly of the display system should place a data word on the IDE. When the IDE contains valid data, the acknowledge line is driven (9CACK = 0). The data will be held until the read line is removed (9CREAD = +5 V). ACKNOWLEDGE will be removed (9CACK = +5 V) after READ is removed, when the assembly is no longer busy and is able to do other transfers. ACKNOWLEDGE will always last at least one clock time after READ.

WRITE--9CWRITE. Interface to the 9000 Series System, active low. The write line indicates (9CWRITE = 0 V) that the selected assembly should receive a data word from the IDE. Data must be valid when WRITE is activated and remain valid until ACKNOWLEDGE (9CACK = 0 V) is received from the receiving assembly. ACKNOWLEDGE will remain

(9CACK = 0 V) until WRITE is removed (9CWRITE = +5 V), and the receiving assembly is no longer busy. ACKNOWLEDGE will always last at least one clock time after WRITE is removed.

ACKNOWLEDGE--9CACK. 9000 Series System to the interface, active low. The acknowledge line, 9CACK, is used in conjunction with READ and WRITE, as described above. ACKNOWLEDGE always means busy. It will be driven by the portion of the system being selected. Thus, ACKNOWLEDGE may remain active for many milliseconds after a write operation, or when reading, it may not occur for milliseconds. A 470-ohm pullup resistor for 9CACK is contained on the interface.

CPU CLEAR--9CPUCLR. Interface to the 9000 Series System, active low. The CPU clear line, 9CPUCLR, is essentially an extension of the master reset switch on the CPU control panel. It is always desirable to use this line for master resetting interrupt circuitry so the programmer may clear any remaining interrupts when starting a program. This ensures the clearing of a program that did not complete an interrupt sequence. The CPUCLR line should reset (9CPUCLR = 0 V) all logic control sections so that no portion of the system remains busy. Some CPU's do not have a clear switch that is made available to the interface. In this case, use the "Master Clear" command and perform the same function under program control.

SELECT LINES--9CSEL0-9CSEL3. Interface to the 9000 Series System, active low. 9CSEL0 enables the 9000 System to use the IDE bus. The interface contains a "Command" register, which holds the 3-bit device selection code. Each section of the 9000 Series System must gate the READ and WRITE lines, as well as drive the control for the date lines with this select line.

INTERRUPT LINES--9BINT0, 9RINT1, 9MINT4-9MINT6. 9000 Series System to the interface, active low. The interrupt lines are not enabled by the select lines and thus are always active. When an interrupt is generated, the line remains at a 0-VDC level until status or data is read from the appropriate interrupting section. In the case of the 9000 Series System. (9BINT0), the interrupt will remain set until the interrupt acknowledge line (9CINTAK) is activated by the interface. The interface has a status register that is read by the CPU to determine which sections of the 9000 Series System are causing interrupts. In most cases, the leading edge of the interrupt line is detected and an appropriate bit is set in a status register. Upon reading the interface status register, all bits are reset. During the time that the CPU reads the status register it will not be changed by the interface.

VIDEO INTERRUPT ACKNOWLEDGE--9CINTAK. Although this line is not used in the 9000 Series Systems, it was used in an earlier version (GX100 Series) to reset the video interrupt line 9BINT0. This line is usually activated (9CINTAK = 0V) when the CPU reads from the interface.

INTERNAL DATA EXCHANGE TIMING

All lines received from other portions of the system will be synchronous to the system clock. The timing for the various types of transfers will vary, depending upon which section of the 9000 Series System is selected. Before beginning any transfers, the interface will always test the acknowledge line to determine the busy condition.

The timing for typical write and read cycles is shown in Figures 4-7 and 4-8. The write cycle may be a few hundred nanoseconds or several seconds. Transfers to the 9000 Series System usually last from 1.5 μs up to several seconds per data word, depending upon the 8080 software. When writing, the interface can get the next word as soon as

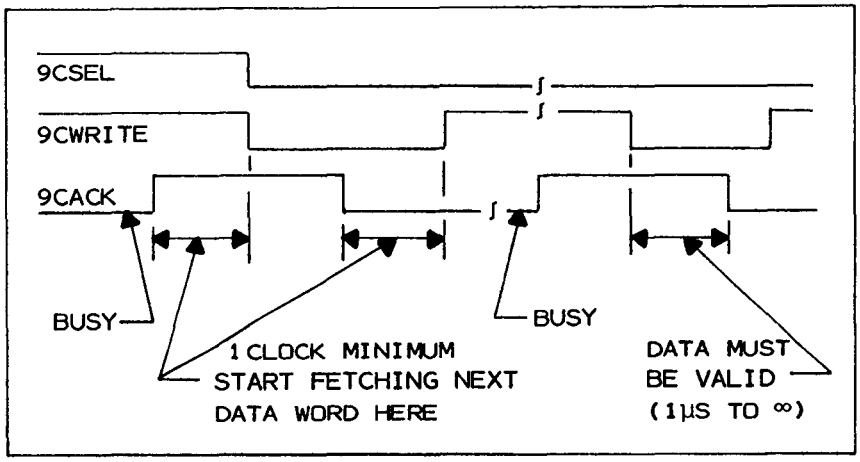


FIGURE 4-7 WRITE TIMING

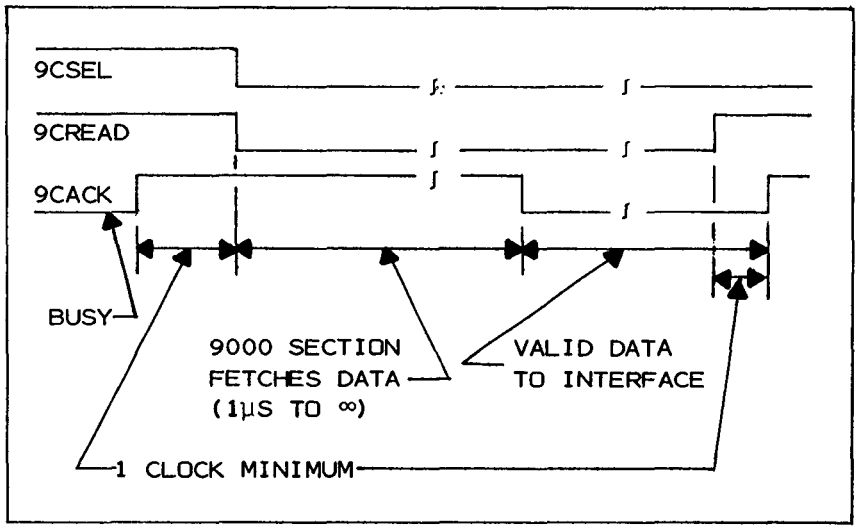


FIGURE 4-8 READ TIMING

ACKNOWLEDGE appears but it must not drive the write line until ACKNOWLEDGE is released from the last transfer. In general, the interface cannot start a write operation until ACKNOWLEDGE is inactive. An input data register is usually provided in the interface in order to prevent holding up the CPU.

The interrupt lines are always active and thus need not be selected. The interrupt lines all synchronously detect the leading edge of an interrupt line to the set the interrupt status flip-flop when enabled by bits in the command register. Typical timing is shown in Figure 4-9. A typical circuit is shown in Figure 4-10. Note that the interrupt flip-flop is inhibited from being set while being read by the CPU. It is reset after being read by the CPU and before enabling the interrupt leading edge circuit gated clock.

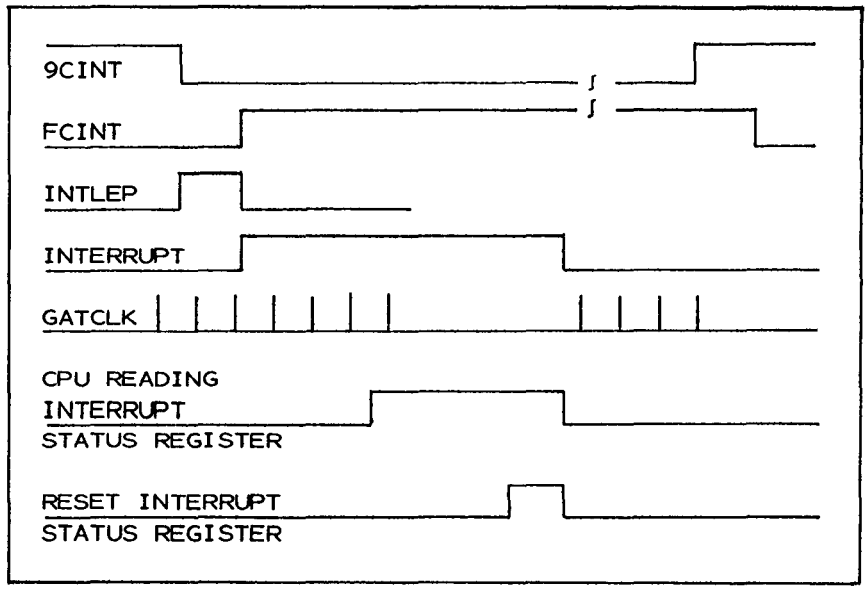


FIGURE 4-9 INTERRUPT TIMING

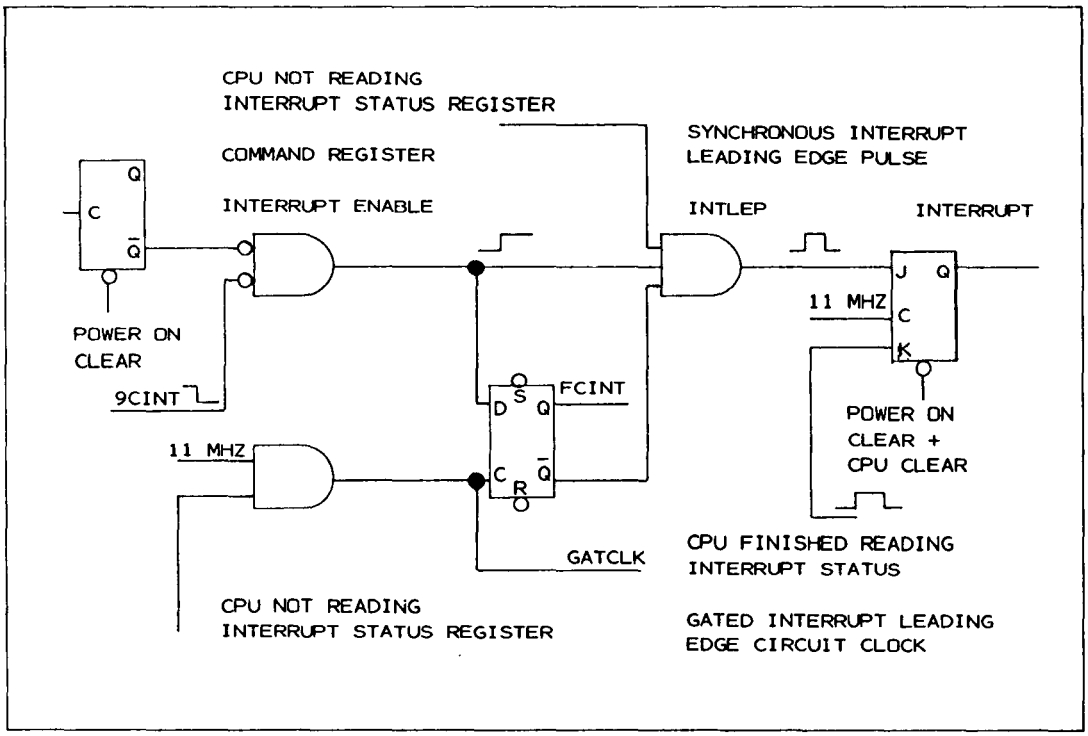


FIGURE 4-10 INTERRUPT NETWORK

5.0 SYNC TIMING BOARD

The Multi-Resolution Sync Timing Board (MRSTC) generates basic timing waveforms for the following RM-9000 Series Raster Scan Graphic Display Systems:

MODEL
RM-9100
RM-9200
RM-9300

5.1 FUNCTIONAL DESCRIPTION

5.1.1 WAVEFORM TIMING CATEGORIES

The following paragraphs discuss the five general categories of waveforms generated by the MRSTC.

Monitor Compatible Signals

The 9000 Series requires raster scan television monitor(s) for display. The MRSTC generates the synchronizing, blanking and driving waveforms required by the monitor(s). These signals are provided to the video boards which assemble the timing waveforms and the refresh memory stored image into a composite video picture signal. External sync-lock capability is available with the addition of the VCO option to the system.

TERM-NAME	DESCRIPTION
9THSSS	Horizontal Synchronization
9TVSSS	Vertical Synchronization
9TCSSS	Composite Synchronization
9THBBB	Horizontal Blanking
9TVBBB	Vertical Blanking
9TCBBB	Composite Blanking
9THDDD	Horizontal Drive
9TVDDD	Vertical Drive

Refresh Memory Synchronization

The 9000 Series Systems use a classical refresh-memory image-storage technique. The MRSTC provides memory synchronizing signals to align memory image storage with monitor synchronizing signals.

TERM NAME	DESCRIPTION
9THMMM	Horizontal Memory Synchronization
9TVMMM	Vertical Memory Synchronization

System Clocks

The MRSTC generates two clocks used by the TTL logic within the system. The two clocks are in phase, one clock being half the frequency of the other. The base-clock frequency is within the range of 12-13 MHz. The half-clock frequency is within 6-6.5 MHz.

TERM NAME	DESCRIPTION
PTBQ	Base Frequency Clock
NTBQ	(ECL Differential)
PTHQ	Half Frequency Clock
NTHQ	(ECL Differential)

Processor Clocks

The MRSTC generates two phase clocks used by the 8080A-1 microprocessors within the system. Phase clock specifications can be found in the 8080A-1 data sheet.

TERM NAME	DESCRIPTION
PTPQØNE	Processor (8080A-1) Phase One Clock
NTPQØNE	(ECL Differential)
PTPQTWØ	Processor (8080A-1) Phase Two Clock
NTPQTWØ	(ECL Differential)

Power-On Reset and Switch Clear

The MRSTC provides a power-on reset signal used to clear all logic within the system upon powering up. The system can be operator cleared at any time by momentarily depressing the reset switch located on the front panel.

TERM NAME	DESCRIPTION
9BPWCLR	Power-On Clear
9BSWCLR	Switch Clear (from reset switch)

5.1.2 MONITOR RESOLUTION CONSIDERATIONS

The term "Resolution" needs to be clear. In the sense used within this manual, resolution means picture information content differing by multiples of two.

Vertical Resolution

High resolution line systems have twice as many *lines of information* as low resolution systems. There are two vertical resolutions in three possible configurations as follows:

a. Low Resolution - Repeat field scan
b. Low Resolution - 2:1 interlaced scan. Information content is identical to a.
c. High Resolution - 2:1 interlaced scan. Information content doubles that of a.

Horizontal Resolution

High resolution element systems have twice as many *elements of information* as low resolution systems. There are two horizontal resolutions as follows:

a. Low Resolution - Dual wide pixels.
b. High Resolution - Single wide pixels. Twice as many pixels as in a.

Resolution Keywords

For purposes of brevity, the keyword terms - low resolution, medium resolution and high resolution - shall mean the following:

a. Low Resolution - Both low vertical resolution and low horizontal resolution.
b. Medium Resolution - Low vertical resolution high horizontal resolution.
c. High Resolution - Both high vertical resolution and high horizontal resolution.

The 9000 Series resolution configurations are summarized below:

MODEL	RESOLUTION KEYWORD
RM-9100	Low Resolution (2x3)
RM-9200	Medium Resolution (2x6)
RM-9300	High Resolution (5x6)

Note that specifying the RM-9000 Series model number automatically specifies the resolution.

Changing Vertical Resolution

Vertical resolution on the MRSTC is determined by the position of the interlace enable switch on the board.

For low vertical resolution - repeat field scan, the interlace switch is OFF.

For high vertical resolution - 2:1 interlaced scan, the interlace enable switch is ON.

If the 9000 Series chassis is wired for a CPU interlace select option, the interlace enable switch on the MRSTC *must be OFF* for proper oper-

ation. The option is wired if a twisted pair cable is attached to the MRSTC at the CPU interlace select option connector J2. The option is not wired and this note does not apply if the cable is not attached.

Changing Horizontal Resolution

Horizontal resolution is fixed by the particular 9000 Series model chassis.

5.1.3 MONITOR FORMAT TIMING CONSIDERATIONS

In the sense used within this manual, FORMAT refers to the total number of lines within a picture frame and the monitor refresh rate.

"NOTE"

Format does not change with resolution.

Format Names

Each 9000 Series model is available in three timing format options; broadcast, standard and European as follows:

MODEL	RESOLUTION KEYWORD	FORMAT KEYWORD	MRSTC DASH NUMBER
RM-9100	2x3 Low	BROADCAST	-01
		STANDARD	-02 or -04
		EUROPEAN	-03
RM-9200	2x6 Med	BROADCAST	-01
		STANDARD	-02 or -04
		EUROPEAN	-03
RM-9300	5x6 High	BROADCAST	-01
		STANDARD	-02 or -04
		EUROPEAN	-03

Format Information

The format specifies the raster scan timing format used to form the composite video picture and is described as follows:

a. Broadcast Format

The broadcast format option provides the following raster scan picture format:

240/480 visible lines
x 320/640 visible elements
within 525 total lines, 60 Hz refresh.

With the broadcast format option, raster scan timing most closely conforms to United States Monochrom Broadcast Standards defined in EIA Standard RS-170, Electrical Performance Standards - Monochrome Television Studio Facilities, Revision TR-135, November 1957.

b. Standard Format

The standard format option provides the following most frequently used domestic raster scan picture format:

256/512 visible lines
x 320/640 visible elements
within 559 total lines, 60 Hz refresh.

The monitor timing waveforms required to produce this format do not match any television broadcast standards. However, timing tolerances stay within industry-wide television monitor manufacturer tolerances for synchronizing, blanking and driving signals.

c. European Format

The European format option provides the following most frequently used raster scan picture format for the export market.

256/512 visible lines
x 320/640 visible lines
within 625 total lines, 50 Hz refresh

The European format option primarily allows the graphic display system monitor to run at a 50 Hz refresh rate. It also provides a video picture within a 625 total line picture frame.

The monitor timing waveforms most closely conform to European monochrome broadcast standard defined in the XIIIth Plenary Assembly of the C.C.I.R., Geneva, Report 624: Characteristics of Television Systems, 1974.

Format Detailed Descriptions

The raster scan information shown in Table 1-1 details the three format options in terms of refresh rates, picture lines, picture elements and crystal frequencies. The MRSTC dash number can be determined from this chart knowing the format chosen. The table also may be used in the reverse manner. Knowing the MRSTC dash number, the format type may be determined.

How the Format Changes the Circuit

a. The system clock crystal frequency (Y2)

-01 12.273 MHz
-02 13.068 MHz
-03 12.188 MHz
-04 10.910 MHz

b. The PROM program of location 2B

-01 502510
-02 502511
-03 502509
-04 502511

c. The PROM program of location 2D

-01 502507
-02 502508
-03 502509
-04 502508

To convert the MRSTC from one format to another necessitates changing the above three components.

5.2 THEORY OF OPERATION

This portion of the manual contains a description of the circuitry used in the MRSTC. Discussion begins with the block diagram shown in Volume II of this manual. Next, each circuit is described in detail using the block diagram and the schematics, drawing 502334.

Complete schematics are given in the diagrams section along with timing diagrams and information sheets.

5.2.1 BLOCK DIAGRAM DRAWING 502334 SHEET 1

System clocks are generated from a crystal-controlled square wave oscillator. For external sync-lock capability, the crystal is removed and a VCO is installed. The oscillator fundamental frequency from either the crystal or the VCO directly forms the system base frequency clock. A divide-by-two network is employed to form the system half-frequency clock. The system clocks are driven off the board at differential ECL levels.

One cycle of the system half-frequency clock represents one low resolution element. The element counter contains the screen refresh element address. The element PROMS decode the element counts to form all horizontal monitor

and memory synchronizing timing waveforms. The element PROMS also decode when to increment the line counter. The element PROM resets the elements to zero at the end of horizontal blanking.

The line counter stores the screen refresh line address. The line PROM decodes the line counts to generate all vertical monitor and memory synchronizing waveforms. The preset PROM determines the line count starting address going into vertical blanking and the reset address going into visible lines.

Horizontal and vertical waveforms are combined at the composite sync/ blanking combinatorial logic to form composite timing waveforms.

All horizontal, vertical, composite and memory synchronizing waveforms are reclocked with the half-frequency clock and are driven off the board with bus drivers.

Processor clocks are generated from a crystal-controlled square wave oscillator. The oscillator fundamental frequency is divided to form four time slots by the divide by four counter and the 1-of-8 decoder. The time slots are grouped to form phase one and phase two time relationships by the J-K pulse shaper and reclock network. The microprocessor phase clocks are driven off the board at differential ECL levels.

The power-on reset and switch clear circuitry counts 15 field times after ($\sim$ 250 ms) during a power up condition and 15 field times after either reset switch is depressed and released. Power-on reset pulse duration is minimally 250 ms.

5.2.2 DETAILED CIRCUIT DESCRIPTION

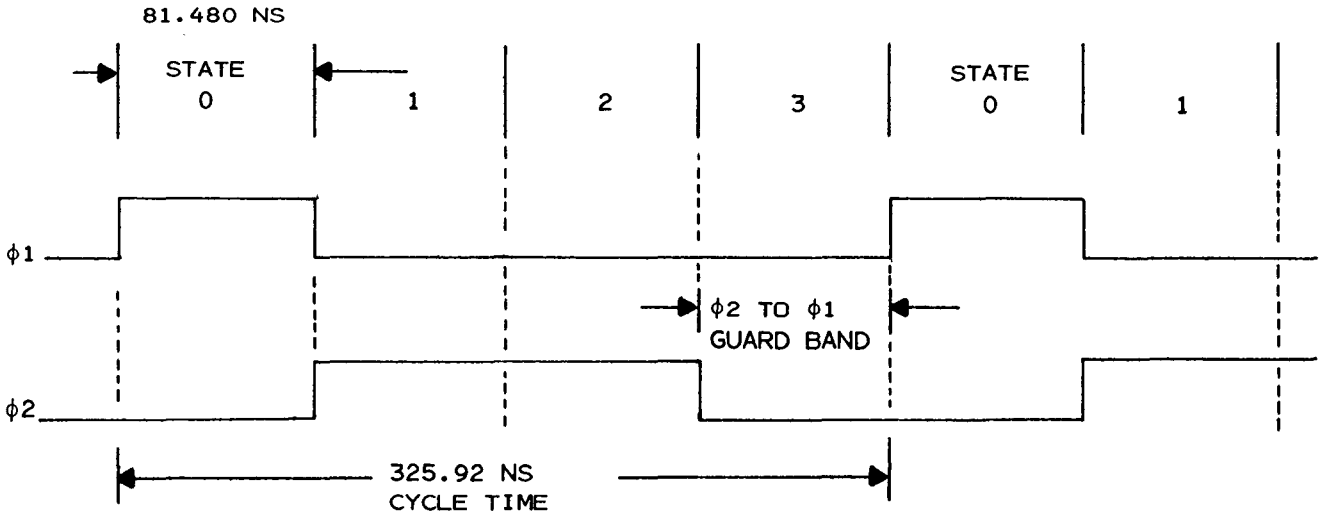
Processor Clock Generation

Crystal Y1 and the three supporting inverters of location 2X form a series resonant fundamental mode square wave oscillator. Crystal Y1 has a fixed

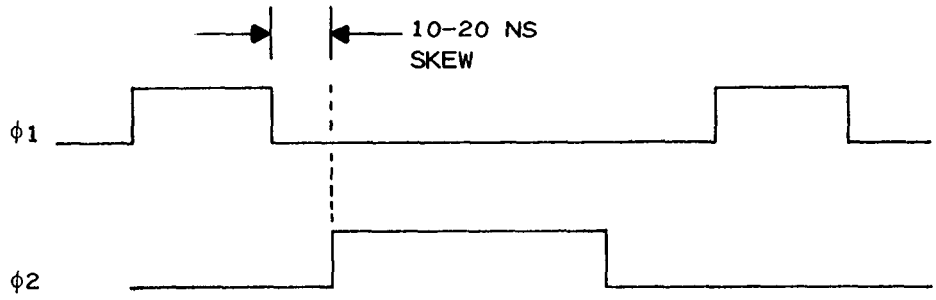
frequency of 12.273 MHz. The frequency value does not change with the MRSTC dash number.

The fundamental frequency drives a free running divide by four counter. The four counter states are decoded by the 1-of-8 decoder. Each decoded state has a duration of 81.480 nanoseconds. (The reciprocal of 12.273 MHz). One cycle through the four states is a cycle time of 325.92 nanoseconds (4 x 81.480 nanoseconds). The J-K pulse shaper and reclock circuit uses the decoded state pulses to set and reset J-K flip-flops according to 8080A-1 $\phi 1$ and $\phi 2$ timing relationships.

The phase one clock is formed from the first decoded state, while phase two is composed of the second two states. The guard band between $\phi 2$ and $\phi 1$ is the final state as shown below:



Two inverters skew the clock driving the $\phi 2$ J-K pulse shaper flip-flop by a minimum of 10 nanoseconds to a maximum of 20 nanoseconds. The result is that $\phi 2$ is skewed ten to twenty nanoseconds with respect to $\phi 1$ as shown below:



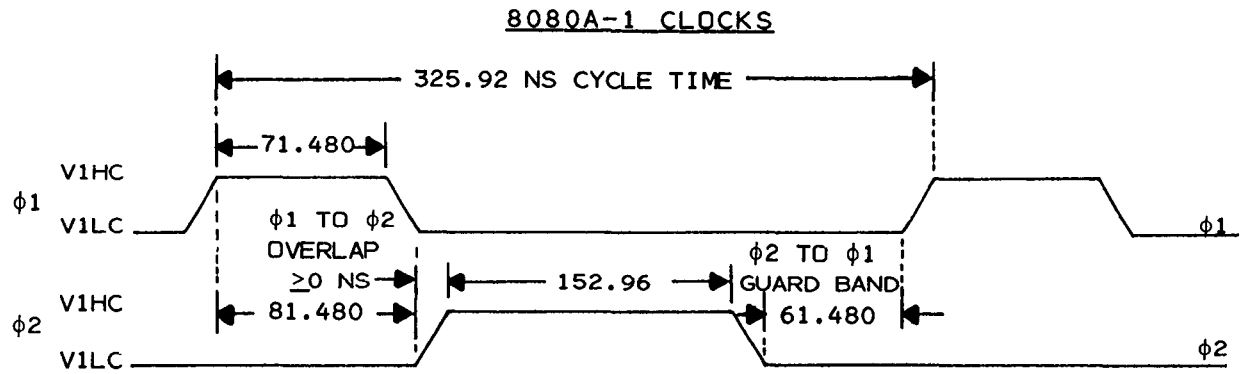
The phase one and phase two processor clocks are driven off the MRSTC by the differential ECL current drivers, location 2V. Resistor pack R11, location 2W forms an ECL line terminator.

Processor clock users receive the processor clocks with differential ECL receivers. Users then must convert the TTL level clocks obtained to MOS levels using MOS clock driver circuits.

The rise and fall times of the MOS level phase clocks must be under ten (10) nanoseconds. Greater rise and fall times will violate 8080A-1 $\phi 1$ to $\phi 2$ overlap specifications.

The purpose of the two inverter clock delay on the MRSTC is thus as follows follows: If the $\phi 1$ to $\phi 2$ overlap were zero (0) nanoseconds as the clocks left the MRSTC, no MOS clock-driver circuit could be devised to meet 8080A-1 clock specifications.

Typical MOS level phase clocks should appear as follows:



NOTES:
1. ALL TIMES SHOWN IN NANOSECONDS (NS).
2. ALL RISE AND FALL TIMES MUST BE 10 NS OR LESS.
3. MOS CLOCK DRIVER VOLTAGE LEVELS

	MIN	MAX
V1HC	9	13
V1LC	-1	0.8

System Clock Generation

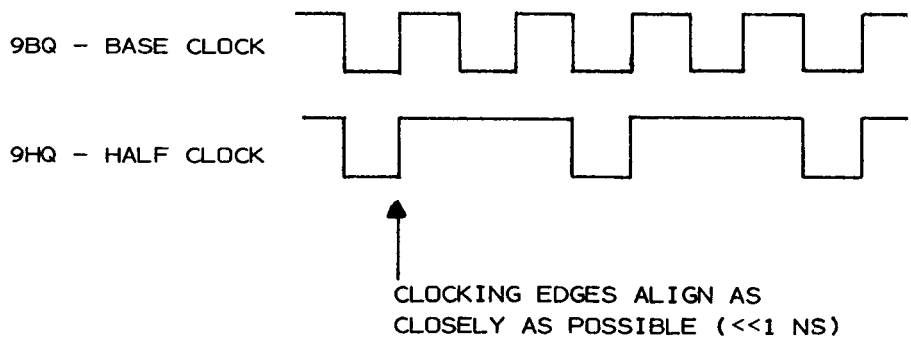
Crystal Y2 and the three supporting inverters of location 2X form a series resonant fundamental mode square wave oscillator. The frequency value of crystal Y2 changes with the MRSTC dash number as follows:

DASH	CRYSTAL FREQUENCY
-01	12.273 MHz
-02	13.068 MHz
-03	12.188 MHz
-04	10.910 MHz

External Sync Lock Option

For external sync-lock capability, an external VCO is attached to the MRSTC through connector J2. The VCO functionally replaces crystal Y2. Crystal Y2 is removed from the circuit upon installation of the VCO.

The fundamental frequency forms the base frequency square wave clock - BQ. The base frequency is divided by two by the J-K flip-flop of location 1X. This half frequency square wave is then combined with the base clock to form a 25% duty cycle half frequency clock - HQ.



The base frequency and half frequency system clocks are driven off the MRSTC by the differential ECL current drivers of location 2V. Resistor pack R11 of location 2W forms an ECL line terminator.

System clock users receive the system clocks with differential ECL receivers. A Schottky NAND gate delay is added to the clocks by the receiving board to allow gated clocks.

The MRSTC "receives" its own generated system clocks by an ECL receiver and two NAND gates. Treating the clocks in this manner insures that all logic on the MRSTC is running synchronously with all logic within the chassis. All timing signals generated by the MRSTC and used elsewhere within the system are aligned with the clock and are treated as data (not clock) signals.

Sync Timing Generator

The element counter consists of two binary counters at location 1A and 1B and a J-K flip-flop at location 2E. The element counter contains the screen refresh element address. One low resolution element is one cycle of the system half frequency clock - 7HQ.

Element counts are decoded by the element PROMS of locations 1C, 1D and 1E. The element PROM programs do not change with the MRSTC dash number. The element counts form the address to the PROMS. All horizontal timing pulses (blanking, sync, drive, memory sync equalizing) originate at the PROM outputs. Three signals originating from the element PROMS are reclocked before use and are "backed up" in the PROM decoding to compensate. The three element PROM outputs that are one clock-time ahead are 5TLCIV, 5TLCIB and 5THEQUAL.

All other element PROM outputs are correct in time. The element PROMS also decode the element counter reset *TECREST and the line counter increments TLCIV and TLCIB.

The element counter is reset to zero at the trailing edge of horizontal blanking. The line counter half-line count TLCHALF is incremented at the leading edge of horizontal blanking and again exactly one half line time away during the visible part of a line. The actual least significant bit of the line counts (TLC1) changes only once per line at the leading edge of horizontal blanking.

The line counter consists of a J-K flip-flop at location 2E and three binary counters at locations 2A, 2C and 2G. The line counter contains the screen refresh line address. Each complete cycle of the half-line count TLCHALF represents one video line.

The two most significant bits of the line counts form vertical blanking TVBLANK and the field flag TVFIELD. The field flag changes states once per field and indicates field A or field B.

The field flag changes state only in interlaced systems. The interlace enable switch of location 1H is ON in interlaced systems. In repeat field systems, the interlace enable switch is OFF and the field flag is inactive low.

The line counts are decoded by the line PROM of location 2D. The line counts form the line PROM address. Vertical timing pulses (sync, drive and memory sync) originate at the line PROM outputs. Note that the line PROM contains field and half-line information. This is significant in interlaced systems. Field A and Field B timing differ by one half-line.

The line PROM outputs align with the leading edge of horizontal blanking. (Remember: The line counts are increments at the leading edge of horizontal blanking.) Vertical sync 6TVSYNC and the vertical equalizing interval 6TVEQUAL are reclocked with the leading edge of horizontal sync by the D-Type flip-flops of location 2J.

The preset PROM determines the starting address of the line counter at the leading edge of vertical blanking and the starting address at the trailing edge of vertical blanking (i.e. the first visible line). The counter presets to a new starting address each time the line counter carries. The starting address depends upon the state of blanking and the field flag.

Composite sync and blanking is generated at the composite sync/blanking combinatorial logic of the sheet 4 of the schematics. (Drawing 502334) Horizontal and vertical timing is combined to form the composite wave forms.

The selector of location 1L brings together composite sync. Composite sync is composed of horizontal sync, vertical sync with vertical serrations and vertical equalizing pulses.

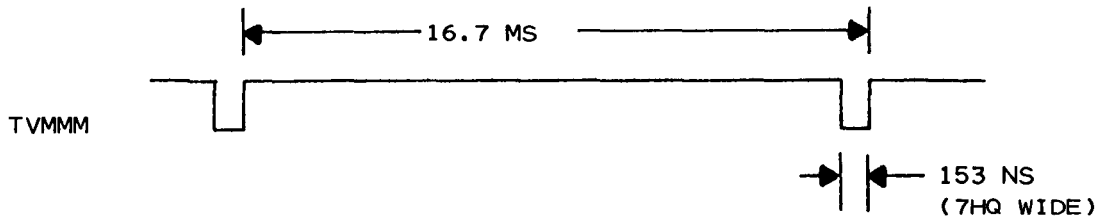
Composite blanking is simply an "OR" function of horizontal and vertical blanking.

All sync timing signals are reclocked by the D-flip-flops of locations of 1N and 2N and are driven off the MRSTC by the line driver buffers of locations 1P and 2P.

"CLOCK TO Q" delay for all sync timing signals is 39 nanoseconds maximum.

Power-On Reset and Switch Clear

Power-on reset duration is controlled by the binary counter of location 1Y. Upon application of power, the counter resets, counts 15 vertical field times and stops. One field time is 16.7 milliseconds. Signal TVMMM is a single 7HQ wide pulse occurring once per field as shown below:



Power-on reset duration is at least 250 ms (15 fields x 16.7 msec/field).

Either reset switch loads the counter with zeros (i.e. resets it) and activates the clear line. Clear remains active 250 ms after switch release.

6.0 CONTROL BOARD

6.1 INTRODUCTION

Section 3.0, Theory of Operation, describes the manner in which the control board is interconnected to all other portions of the 9000 Series System. The control board provides the basis for all memory and register address assignments in the 9000 Series System. Table 6-1 represents a list of the registers and memory addresses for the entire 9000 Series System. (Note that memory mapped I/O is utilized.) All addresses below hex address 8000 are utilized for memory and all addresses including 8000 and above are utilized for I/O address. Table 6-2 summarizes the address assignments for all the registers which may be thought of as I/O yevices to the 8080A-1 microprocessor. (note in Table 6-2 that the I/O addresses 8000 to 8026 and 80A6 are utilized internal to the control board.) I/O addresses 8036 through 805A are utilized for the two serial link boards which may be contained in a 9000 Series System. Addresses 8060 through 806E are utilized for up to four video boards.

Note that in Table 6-2 some of the registers on the control board have two separate addresses. For example, the foreground register addresses are 8012 and 8092. The difference between these two addresses is that in address 8012 bit 7 is a "0" and in address 8092 bit 7 is a "1". When an address for a register on the control board is used that has bit 7 as a "1", it is a signal to the display generator that a refresh memory operation should be performed after the register is loaded. Refresh memory operations consist of writing refresh memory and current operating point (COP) manipulation, reading memory and COP manipulation or only COP manipulation. Note also in Table 6-2 that not all the registers on the control board have addresses in the range 8080 to 80A6. Those registers that are not present should never be used with an address in this range. In one case in particular, the DMA word count register will cause a system hang condition to result.

Still referring to Table 6-2, in all cases except two, the register addresses shown for each register are the even byte addresses in memory.

All registers are 16 bit registers, therefore the implication is that each register really has 2 byte addresses associated with it. As an example, subchannel mask register byte addresses are 8000, 8001 or 8080, 8081. In all cases, the double byte addresses for each register are even-odd (low/high) byte pairs. The two exceptions noted above are the COP update control register and the host data register. For the COP update control register, only the address of the high byte is shown, as that is all that really need by used. For the host data register, the addresses of both the low byte and the high byte are shown.

TABLE 6-1

9000 SERIES ADDRESS DOMAIN

ØØØØ - Ø7FF	ROM(STD)
Ø8ØØ - 3FFF	ROM(ME)
4ØØØ - 5FFF	RAM(ME)
6ØØØ - 61FF	RAM(STD)
62ØØ - 63FF	CHARACTER ROM
64ØØ - 7CØØ	ROM(DIAG.)
7CØØ - 7FFF	RAM(DIAG.)
8ØØØ - FFFF	I/O ADDRESSES

TABLE 6-2

I/O ADDRESS ASSIGNMENT

SCMSKR	;8000	-	SUBCHANNEL MASK REGISTER
RUPTRG	;8002	-	INTERFACE INTERRUPT REGISTER
READR	;8004	-	READBACK ELEMENT REGISTER
XORGR	;8006	-	ELEMENT ORIGIN ADDRESS REGISTER
YORGR	;8008	-	LINE ORIGIN ADDRESS REGISTER
NPRSRC	;800A	-	DMA SOURCE REGISTER
NPRDST	;800C	-	DMA DESTINATION REGISTER
UPDTR	;800F	-	COP CONTROL REGISTER
BGR	;8010	-	BACKGROUND ELEMENT REGISTER
FGR	;8012	-	FOREGROUND ELEMENT REGISTER
XCOPR	;8014	-	ELEMENT COP ADDRESS REGISTER
YCOPR	;8016	-	LINE COP ADDRESS REGISTER
RASREG	;8018	-	FONT DATA REGISTER
HDIL	;801A	-	HOST DATA REGISTER (LO-BYTE)
HDIH	;801B	-	HOST DATA REGISTER (HI-BYTE)
SYSS	;801C	-	SYSTEM STATUS REGISTER
WRDCNT	;801E	-	DMA WORD COUNT REGISTER
DMAADR	;8020	-	DMA ADDRESS REGISTER
MOVWTR	;8022	-	MOVE COP AND WRITE REGISTER
MOVR	;8024	-	MOVE (W. NO WRITE) REGISTER
OPTFLG	;8026	-	OPTIONS FLAG DIP SWITCH REGISTER
	;8028		
	;802A		
	;802C		
	;802E		
	;8030		
	;8032		
	;8034		
DCKYB1	;8036	-	DIAGNOSTIC CARD KEYBOARD BYTE 1
DCKYB2	;8037	-	DIAGNOSTIC CARD KEYBOARD BYTE 2
DCKYB3	;8038	-	DIAGNOSTIC CARD KEYBOARD BYTE 3

TABLE 6-2 (Cont'd)

DCADR	;803A	-	DIAGNOSTIC CARD ADDRESS REGISTER
SPS1R	;803C	-	SERIAL STATUS REGISTER (BOARD 1)
SPS2R	;803E	-	SERIAL STATUS REGISTER (BOARD 2)
SPK11	;8040	-	KEYBOARD I/O REG #1 (BOARD 1)
SPK12	;8042	-	KEYBOARD I/O REG #2 (BOARD 1)
SPK13	;8044	-	KEYBOARD I/O REG #3 (BOARD 1)
SPK14	;8046	-	KEYBOARD I/O REG #4 (BOARD 1)
SPCX11	;8048	-	CURSOR #1 ELEMENT REG (BOARD 1)
SPCY11	;804A	-	CURSOR #1 LINE REG (BOARD 1)
SPCX12	;804C	-	CURSOR #2 ELEMENT REG (BOARD 1)
SPCY12	;804E	-	CURSOR #2 LINE REG (BOARD 1)
SPK21	;8050	-	KEYBOARD I/O REG #1 (BOARD 2)
SPK22	;8052	-	KEYBOARD I/O REG #2 (BOARD 2)
SPK23	;8054	-	KEYBOARD I/O REG #3 (BOARD 2)
SPK24	;8056	-	KEYBOARD I/O REG #4 (BOARD 2)
SPCX21	;8058	-	CURSOR #1 ELEMENT REG (BOARD 2)
SPCY21	;805A	-	CURSOR #1 LINE REG (BOARD 2)
SPCX22	;805C	-	CURSOR #2 ELEMENT REG (BOARD 2)
SPCY22	;805E	-	CURSOR #2 LINE REG (BOARD 2)
VLTAD1	;8060	-	TYPE 2 VLT #1 ADDRESS REGISTER
VLTD1	;8062	-	TYPE 2 VLT #1 DATA REGISTER
VLTAD2	;8064	-	TYPE 2 VLT #2 ADDRESS REGISTER
VLTD2	;8066	-	TYPE 2 VLT #2 DATA REGISTER
VLTAD3	;8068	-	TYPE 2 VLT #3 ADDRESS REGISTER
VLTD3	;806A	-	TYPE 2 VLT #3 DATA REGISTER
VLTAD4	;806C	-	TYPE 2 VLT #4 ADDRESS REGISTER
VLTD4	;806E	-	TYPE 2 VLT #4 DATA REGISTER
SCMSKR	;8080	-	SUBCHANNEL MASK REGISTER
READR	;8084	-	READBACK ELEMENT REGISTER
UPDTR	;808F	-	COP CONTROL REGISTER
BGR	;8090	-	BACKGROUND ELEMENT REGISTER
FGR	;8092	-	FOREGROUND ELEMENT REGISTER
XCOPR	;8094	-	ELEMENT COP ADDRESS REGISTER
YCOPR	;8096	-	LINE COP ADDRESS REGISTER
MOVWTR	;80A2	-	MOVE COP AND WRITE REGISTER
MOVR	;80A4	-	MOVE (W. NO WRITE) REGISTER

6.2 DISPLAY GENERATOR AND OTHER 8080 I/O REGISTERS

The following registers comprise all those registers which are contained on the 9000 Series System control board. Each of these registers is a separate I/O device address to the 8080A-1 microprocessor. Most registers are treated as 16 bit registers consisting of a most significant byte (Bits 8-15) and least significant byte (Bits 0-7). In the following descriptions of the registers, all undefined bit positions are treated as "don't care" bits. Table 6-2 shows the I/O Address assignment for each of these registers.

6.2.1 SUBCHANNEL MASK REGISTER (ADDRESSES = 8000, 8001 or 8080, 8081)

				MP11	MP10	MP9	MP8	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

In order to load the subchannel mask register, both low byte and the high byte of this register must be loaded, in that sequence and consecutively. If an attempt is made to only load the low byte of this register it will not be transferred to the register. It is also possible to cause a refresh memory right and update operation to occur when this register is loaded by using the addresses 8080, 8081. In this case, the data that currently resides in the foreground register will be written into refresh memory under the control of the memory plane select bits that were just loaded.

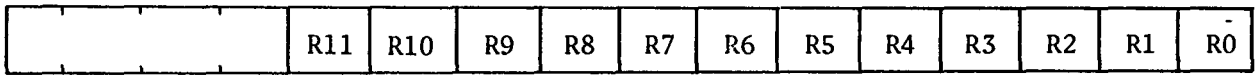
MP0 through MP11 (Bit 0 - Bit 11) represent RAM refresh memory planes 0 through 11. When set to a "1" state, the corresponding memory plane is enabled and will receive and store its corresponding data from the foreground or background register whenever a "write-element" function is generated. When set to a "0" state, the corresponding memory plane is disabled and will ignore all "write-element" functions. All memory planes are enabled upon powering the system up. The definition of enabled

memory planes is retained indefinitely after the memory planes register is reloaded.

6.2.2 INTERFACE INTERRUPT REGISTER (ADDRESSES = 8002, 8003)

To load the interface interrupt register, the 8080 must load both bytes of the register. An attempt to load only the low byte of the register will fail, as the data is only transferred to the register when the high byte is loaded. Therefore, this register must be treated as a 16 bit register. Bits 0-3 represent interrupts I0-I3 which are interrupt lines to the standard interface board. Upon loading this register with one or any combination of the interrupts, an interrupt is generated to the standard interface board. 8080 software may set any bit within the interrupt register and then reset the bit within a maximum of 10 microseconds. 8080 software may also choose to set an interrupt line and not reset it until such time as the host CPU reads the data value which generated the interrupt.

6.2.3 READBACK ELEMENT REGISTER (ADDRESSES = 8004, 8005 or 8084, 8085)

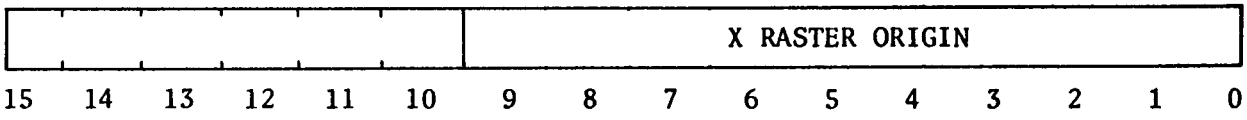


When the 8080A-1 microprocessor addresses the readback pixel register, the memory control logic causes an element value represented by the COP address to be fetched from RAM refresh memory and loaded into the readback element register. After the 8080A-1 Microprocessor reads the value from the readback element register the COP is then modified by one step in any of the eight directions specified by the +X, -X, +Y, -Y update

control bits of the COP/write control register, providing that the hold control bit (H) is a "0", if it is a "1" no update is performed. R0-R11 represent RAM refresh memory planes 0-11 in a manner identical to the subchannel mask register. All bits are read back regardless of the memory plane select register. This register is also used to pass image data under DMA readback. The value read back for unused planes is indeterminent.

In order for the 8080A-1 microprocessor to actually read back an element value from the RAM refresh memory, byte addresses 8084 and 8085 must be used. If the byte addresses 8004 and 8005 are used, the 8080 will only be able to read the data that currently resides in the readback elements register and no COP update will be performed. The reason for this is that the display generator circuitry was not flagged by address bit 7 to actually perform a RAM refresh memory operation.

6.2.4 ELEMENT ORIGIN ADDRESS REGISTER (ADDRESSES = 8006, 8007)



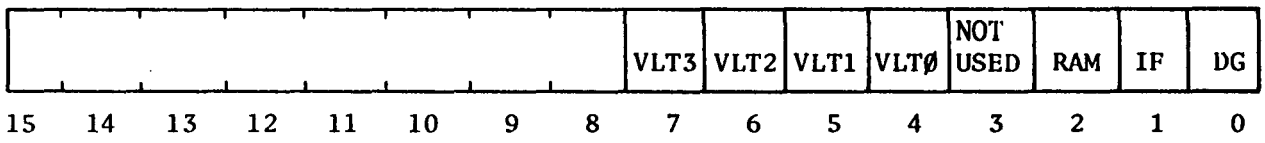
In order to load the element origin address register, both bytes (8006, 8007) must be transferred to the register. Bits 0-9 represent an X (element) address within the RAM refresh memory, which represents the raster origin. The X raster origin value is not transferred to the element origin address register until the high byte of the register is actually loaded. This register, under hardware control, may only be loaded once every twelve raster scan line times. The X raster origin value in this register specifies the first element to be written starting at the left hand side of the screen on every raster scan line.

6.2.5 LINE ORIGIN ADDRESS REGISTER (ADDRESSES = 8008, 8009)

To load the line origin address register, both bytes (8008, 8009) must be transferred. Bits 0-8 represent a Y (line) origin address within the refresh RAM memory which will be utilized to select the first line to be presented as the top line of the television raster scan field.

Thus, the values in the element origin address registers and the line origin address register specify the first element to be written into the upper left hand corner of the television raster, and the television raster is always generated relative to this specified origin.

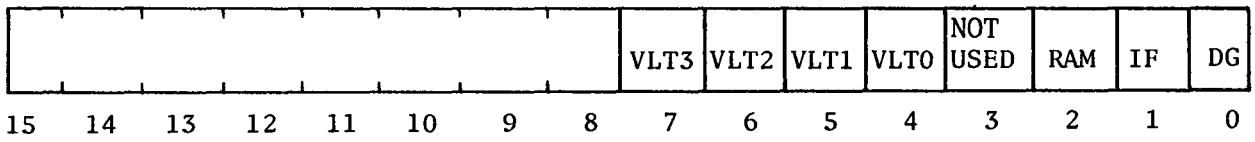
6.2.6 DMA SOURCE REGISTER (ADDRESSES = 800A, 800B)



To load the DMA source register, both bytes (800A, 800B) must be transferred. This register is a 16 bit register that contains all of DMA source flags for all devices that are capable of executing DMA operations. The device whose source flag is set will be the data source in a DMA operation. All defined bits in this register should be treated as mutually exclusive, that is, no 2 bits should ever be set to a "1" simultaneously (see FONT DATA Register). The definition of the DMA control bits are listed below:

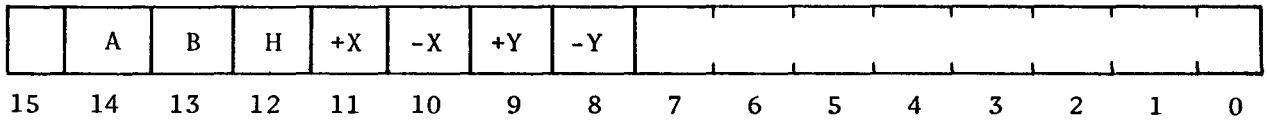
DG = DISPLAY GENERATOR
IF = INTERFACE
RAM = MEMORY EXPANSION BOARD RANDOM ACCESS MEMORY
VLTØ - VLT3 = VIDEO LOOKUP TABLES Ø THROUGH 3 RESPECTIVELY

6.2.7 DMA DESTINATION REGISTER (ADDRESSES = 800C, 800D)



To load the DMA destination register, both bytes (800C, 800D) must be transferred. This register is a 16 bit register that contains all of the DMA destination flags for all devices that are capable of executing DMA operations. The device whose destination flag is set will be the destination in a DMA operation. All defined bits in this register should be treated as mutually exclusive.

6.2.8 COP CONTROL REGISTER (ADDRESSES = 800E, 800F OR 808E, 808F)



To load the COP control register, it is only necessary to transfer the high byte (800F or 808F) because the low byte contains "don't care" information. If the byte address 800F is used to load the COP control register, the state of the bits in this register are loaded and nothing else occurs. If the byte address 808F is used to load the COP control register, after the register is loaded a memory write operation will be performed.and a COP update will occur based upon the information that was loaded in this register. In any case, the state of the bits in this register once loaded remain constant until loaded again. The COP CONTROL REGISTER specifies the mode to be used in writing memory and it establishes the RAM refresh memory COP update equation to be used for post memory operation (read or write) COP updates.

ADDITIVE WRITE BIT (A)

This bit is effective only when the font data register is in use.

A = 0
All data, "0" or "1" in the font data register will cause a write to the refresh memory.

A = 1
Suppression of "0" data in the font data register will occur; that is, only "1's" will cause a write to memory.

REVERSE BACKGROUND BIT (B)

This bit is effective in all modes of writing refresh memory, and under DMA writes to refresh memory (image data) it must be set to B = "0".

B = 0
Graphic write operations.
Causes selection of the data in the foreground element register.

B = 1
Graphic write operations.
Causes selection of the data in the background element register.

Font write operations.
Causes "0" font data to select the foreground element register and "1" font data to select the background pixel register.

HOLD BIT (H)

This bit is effective only when reading refresh memory for graphics operations. It may be used to accomplish a read-modify-write operation on refresh memory. This bit has no meaning (ignored) under DMA readback.

H = 0

The COP will be updated as specified after the execution of an element readback.

H = 1

The COP will *not* be updated after execution of an element readback.

The +X, -X, +Y, -Y control bits specify the direction of movement for the COP addresses which is modified while writing or reading element data. Eight combinations of the 4 bits denote eight possible directions of movement for the COP. These scan direction bits specify the direction of movement to be utilized when writing image data into the foreground element register from the 8080 microprocessor or the interface board. They also specify scan direction when reading the readback element register.

6.2.9 BACKGROUND ELEMENT REGISTER (ADDRESSES = 8010, 8011 or 8090, 8091)

				B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The background element register, bits B0-B11, represent RAM refresh memory planes 0-11 in a manner identical to the subchannel mask register. Hence, the 12 bit value contained in this register represents a background color or grey scale level of intensity as the "0" state for two dimensional graphics raster or font data.

6.2.10 FOREGROUND ELEMENT REGISTER (ADDRESSES = 8012, 8013 or 8092, 8093)

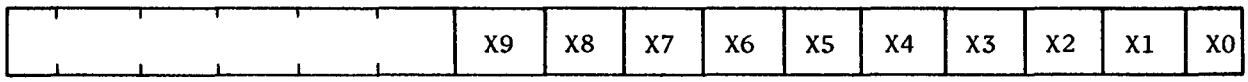
				F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The foreground element register bit F0-F11 represent RAM refresh memory planes 0-11 in a manner identical to the subchannel mask register. Hence,

the 12 bit value contained in this register may represent a foreground color or grey scale level of intensity as the "1" data for two dimensional graphics raster or font data.

Note, that by the appropriate selection of the addresses for either the background element register or the foreground element register, the data values that were loaded into these registers may be immediately written to the RAM refresh memory.

6.2.11 ELEMENT CURRENT OPERATING POINT ADDRESS REGISTER (ADDRESSES = 8014, 8015 OR 8094, 8095)



The element COP address register bits X0-X9 represent the X (element) address which is the XCOP.. The XCOP is a portion of the total address to RAM refresh memory. It specifies the horizontal displacement (in terms of visible elements on the monitor display) from the left edge of the raster, relative to the X origin, determined by the following equations:

D_X = (XORG + 10 + XCOP) MOD320 (low X resolution systems)

D_X = (XORG + 20 + XCOP) MOD640 (high X resolution systems)

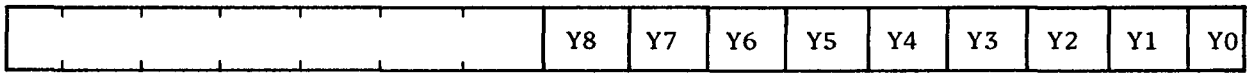
where,

D_X = displacement from the left edge of the raster

XORG = the value in the X origin register

XCOP = the value in the XCOP register

6.2.12 LINE CURRENT OPERATING POINT ADDRESS REGISTER (ADDRESSES = 8016, 8017 or 8096, 8097)



The Line COP address register bits Y0-Y8 represent the Y (line address that is the YCOP. The YCOP is a portion of the total address for RAM refresh memory. It specifies the vertical displacement (in terms of visible lines) from the top edge of the raster relative to the Y origin, determined by the following equations:

D_y = (YORG + 1 + YCOP) MOD 256 (low Y resolution systems)

D_y = (YORG + 2 + YCOP) MOD 512 (high Y resolution systems)

where,

D_y = displacement from the top edge of the raster

YORG = the value in the Y origin register

YCOP = the value

It may be seen from the above equations that to write an element at $D_x = 0$ and $D_y = 0$ on the raster, four registers must be loaded to the correct values. For the above, in a 256 x 320 system,

$D_x = D_y = 0$,

the four registers must contain the following values:

XCOP = 0 (0_{16})

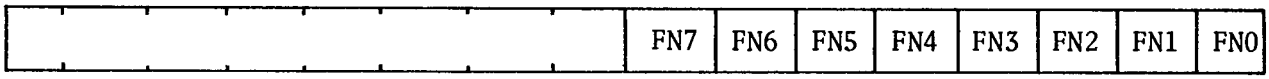
XORG = 310 (136_{16})

YCOP = 0 (0_{16})

YORG = 255 (FF_{16})

Combining the element COP address and the line COP address, any element on the raster is addressable to the granularity of a single element for which the size is a function of the horizontal and vertical resolution of the system. It is also possible with these registers by the appropriate selection of their byte addresses to cause the RAM refresh memory to be written with the data value residing in the foreground register at the RAM refresh memory address they specify.

6.2.13 FONT DATA REGISTER (ADDRESSES = 8018, 8019)



This register contains 8 bits of data which may be loaded and then may be used to control which of two element data values, that of the foreground element or background element registers, will be written to refresh memory, as influenced by the A and B bits of the control register. To use this register, the following steps must be executed.

1. Load the font data register with the desired data pattern.
2. Set both the source and destination flags of the display generator.
3. Set the control, XCOP and YCOP registers to the desired states to specify writing mode, COP update equation, and starting point.
4. Load the word count register with the desired number of write operations to be performed.

Steps 1, 2 and 3 may be performed in any order but 4 must be last as execution begins as soon as the word count is loaded.

The font data register is implemented in the hardware such that it is only necessary to load the low byte even though it is defined as a 16 bit register.

Using the font data register is a quasi-DMA operation as the DMA circuitry is used in execution.

There are two normal modes in which the font data register is used.

1. Writing alpha-numeric font data.
   The alpha-numeric font is loaded into the register and output serially, line by line. A series of 9, 7 element writes are executed to refresh memory to write an alpa-numeric character.

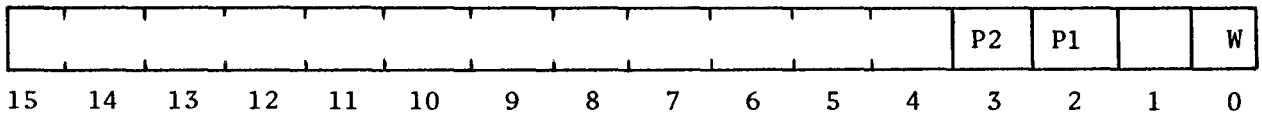
2. Writing raster data on the screen.
   The data pattern is loaded into the font data register and then output to the refresh memory. The register end-around-shifts its data, as data is written to RAM refresh memory, such that the pattern may be repeated a number of times depending upon the word count that is used.

6.2.14 HOST DATA REGISTER (ADDRESSES = 801A, 801B)

DE15	DE14	DE13	DE12	DE11	DE10	DE9	DE8	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The host data register bit DE0-DE15 represent a 16 bit data word to be received from or transferred to the host interface board. When the 8080A-1 outputs the most significant byte to the host data register, it is automatically transferred to the interface board and hence the host CPU. When the 8080A-1 Microprocessor inputs the most significant byte from this register, handshaking operations are completed between the 8080 and the interface board. There is no actual hardware implemented on the control board for this register. Rather, it is implemented with a technique of driving "not ready" to the 8080 such that it will hold the data on the bus while the hand shaking occurs.

6.2.15 SYSTEM STATUS REGISTER (ADDRESSES = 801C, 801D)



The system status register bit 0 represents the write handshake line from the host CPU interface board. This line is ANDed with the acknowledge handshake line before becoming a bit in the status register. Therefore, the 8080A-1 Microprocessor may simply test this bit for the "1" state and if it is a "1", then a data word transfer between the interface and the 8080A-1 should occur. Bits 2 and 3 of the system status register represent a flag which indicates that an event such as a joystick TRACK event, a joystick ENTER event, or a UART I/O operation has taken place on serial peripherals board #1 or #2 respectively. Bits 2 and 3 are cleared upon the act of reading the interface status register.

Typically, while the 8080 software is not executing user instructions or handling data transfers with a serial link board it will be executing a short loop that does nothing other than test the state of this register. When the interface write is detected, the word will be read and based upon this word the software can determine if this user instruction is a multi-word instruction. If it is found to be multi-word instruction, the additional transfers reads or writes that must be completed to execute the instruction will be handled within the 8080 software without returning to the idle loop. Since all user instructions to the 9000 System must begin by writing a word to the system from the host CPU and are handled in the manner described above, there is no need to test the state of the interface read line as the 8080 software will know which line should be active next as a result of the word it just read and will proceed without testing the status register.

Since the bits of interest in this status register are all in the low byte, only the low byte need be read.

6.2.16 DMA WORD COUNT REGISTER (ADDRESSES = 801E, 801F)

				WC11	WC10	WC9	WC8	WC7	WC6	WC5	WC4	WC3	WC2	WC1	WC0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 0-11 (WC0-WC11) represent a 12 bit word count which when loaded will initiate DMA operations between the source and destination previously specified. When the 8080A-1 loads the most significant byte of the DMA word count register, it is placed in the hold state at the beginning of the next instruction and DMA transfers will proceed. Before each 16 bit word is transferred between the source and destination, the DMA word count register is decremented by 1. The 8080 software must load the actual word count into the register when initiating a DMA transfer. When the word count register is decremented to zero, the 8080A-1 hold line will be released and allow the 8080A-1 to continue processing.

6.2.17 DMA ADDRESS REGISTER (ADDRESSES = 8010, 8021)

		STARTING ADDRESS													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

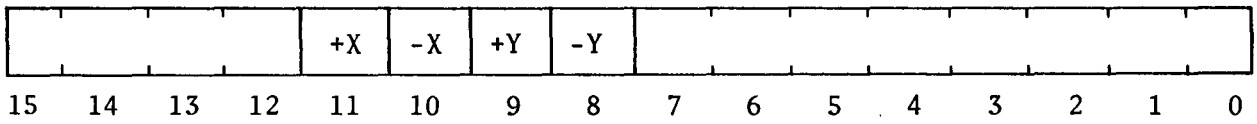
The DMA address register is loaded with the starting address for a DMA transfer to or from memory expansion RAM prior to loading the DMA word count register which starts such a DMA operation. This register is used only for DMA operations to and from memory option RAM only. The range of valid starting byte addresses extends from 4000 to 5FFF and must be even.

6.2.18 MOVE COP AND WRITE REGISTER (ADDRESSES = 8022, 8023 OR 80A2, 80A3)

				+Y	-X	+Y	-Y								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The move COP and write register, although defined as a 16 bit register, may be treated as a byte register transferring only the high byte. This register is not actually a register in the hardware, but is implemented using the ready line to the 8080A-1. Proper operation of this register requires that the byte address 80A3 must be used. It is required that bit 7 of the byte address be present in order to flag the display generator hardware to actually execute the operations specified by this register. If the byte address 8023 is used, nothing will occur. The +X, -X, +Y, -Y bits in the move COP and write register specify the direction of movement to be performed on the COPs before performing a RAM refresh memory write of the data in the foreground or background registers (depending upon the state of the B bit in the control register). This register which allows the movement of the COPs and then the writing of a single element is useful in generating vector data patterns on the monitor.

6.2.19 MOVE COP REGISTER (ADDRESSES = 8024, 8025 OR 80A4, 80A5)



The move COP register, although defined as a 16 bit register, may be treated as a byte register transferring only to the high byte. This register is not actually a register in the hardware, but is implemented using the ready line to the 8080A-1. Proper operation of this register requires that the byte address 80A5 must be used. It is required that bit 7 of the byte address be present to flag the display generator hardware to actually execute the operations specified by this register. If the byte address 8025 is used, nothing will occur. The +X, -X, +Y, -Y bits in the move COP register specify the direction of movement to be performed on the COP without performing any type of operation, read or write, on the RAM refresh memory. That is, this register simply moves the COP.

6.2.20 OPTIONS FLAG DIP SWITCH REGISTER (ADDRESSES = 8026, 8027)

								SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The options flags dip switch register is a register that may be read or written by the 8080 software. When reading this register it is only necessary to read the low byte. The configuration of a set of eight switches may be read by the 8080 software and used for software configuration control. If this register is written by the 8080 software, the only thing that will occur is to extinguish the self-test LED on the front panel. Therefore, the transfer need only be a byte transfer and the data is irrelevant.

6.3 CONTROL BOARD LOGIC - Block Diagram - Sheet 1

Sheet 1 of the logic schematics for the control board (Dwg. 502452) represent a detailed block diagram of the control board. As shown on the block diagram, the 8080A-1 CPU drives the internal processor (IP) bus through a set of address drivers and data drivers-receivers. Also shown is a block called 8080-IP Bus Handshake Interchange. This represent the circuitry that is necessary to synchronize the 8080 handshakes with the IP bus handshakes. The address drivers drive a 16 bit address (4BADDR00-4BADDR15) to all the circuits on the control board that require this address and onto the system backplane. The 8080A-1 CPU is an 8 bit (byte) CPU. The IP data bus is defined as a 16 bit data bus. Therefore, the circuitry consisting of the blocks Low Byte Register, 8080-IP Data Bus (high byte) DRVRS - RCVRS and 8080-IP Data Bus (low byte DRVRS-RCVRS make up the circuitry required to transform the 8 bit data bus of the 8080A-1 to the 16 bit data bus of the IP data bus. The conversion is achieved by the use of an 8 bit byte buffer packing register denoted low byte register to hold the low byte of the 16 bit word while the 8080A-1 fetches the high byte at which point both bytes are driven onto the IP bus in parallel to create a 16 bit word.

The memory that resides on the control board is shown in the lower left hand corner of the block diagram. The block labled 8080 Program Memory (addresses 000-07FF) is 2K of PROM memory that is used to contain the standard system software.

The block labled, 8080 Scratch Memory (addresses 6000-61FF) is 256 bytes of RAM memory that is used as a scratch area and program stack for the 8080 software. The block labled ASCII Character Memory (addresses 6200-63FF) is 512 bytes of ROM memory that contains the font data for the 64 upper case characters of the standard ASCII character set. The block labled Memory Decode and Control receives addresses off the address bus and decodes which memories are to be used. It also drives the 4BREABY line to cause the 8080A-1 to wait when reading the ASCII character memory as this memory is slower than an 8080 clock cycle. This block also controls the memory buffer drivers and receivers that buffer the memories onto and off of the high byte of the internal processor data bus.

At the top of the block diagram, there are several blocks that perform the transition between the interface bus and the IP bus. The block labeled IDE - IP Bus Data Drivers and Receivers buffers the IDE bus data onto the IP bus data. The block labeled IDE - IP Bus Handshake Interchange, handles transitions between the IDE bus handshake philosophy and the IP bus handshake philosophy with some control from the block called the Handshake Sequencer. The differences in handshake philosophy will be discussed later. The handshake sequencer also controls the initiation and completion of DMA transfers within the 9000 System. Just below the handshake sequencer block is the DMA word counter, which counts 16 bit words during DMA transfers. As can be seen, it is decremented by the handshake sequencer and the handshake sequencer detects when it goes to 0. The block labeled Interrupt Register is a register that can be set and reset by the 8080A-1 microprocessor. The register is used to drive four interrupts back to the interface board on the IDE bus.

The IP bus, as shown in the block diagram, provides the communication capability between the 8080A-1 microprocessor and all devices in the 9000 System. The IP bus provides capability for both PIO transfers and DMA transfers. PIO transfers can take place between the 8080A-1 micro-

processor and all device registers in a 9000 Series System. DMA transfers can take place between devices in a 9000 Series System excluding the 8080A-1 micro-processor. Therefore, this bus extends across the system backplane to all boards in the 9000 Series Systems except the interface board and the memory boards.

On the control board, to minimize the drive requirements on the IP bus data lines, the data lines are repeated through an inverting buffer found in the block labled Control Board Data Bus DRVRS-RCVRS. The drivers of this block provide the control board internal data bus PTD00 - PTD15 while the receivers of this block receive data from the readback register and drive it onto the IP bus data lines 9BDATA00-9BDATA15. The drivers are always enabled driving the control board internal data bus except when data is being transferred from the readback register onto the IP bus, as controlled by the signal 3RGREAD.

The block labeled I/O Device Decode receives all the necessary address and handshake control lines to properly decode and provide enables to all device registers on the control board. The I/O device decode provides a variety of signals for gating and clock load pulses as required by the hardware. The block labled Display Generator and Interface Source and Destination Flags contains a portion of the DMA source register flags and the DMA destination register flags. Particularly, the display generator flags and the interface flags. The other flags that are part of this register are contained on other boards as appropriate. The block labeled Memory Plane Select Register receives 12 bits from the IP data bus (9BDATA00-9BDATA11) and is loaded from the I/O device decode with the signal 7HQPLANE to provide the 12 memory plane select signals (9DWE00-9DWE11). The block labeled Foreground, Background and Readback Registers is a register file (RAM) that is used to implement these registers. These registers receive 12 bits of data from the control board internal data bus and provide 12 bits of data FB00-FB11). During RAM refresh memory write operations, the foreground or background register provides 12 bits of data (FB00-FB11) that are driven through the refresh memory data bus drivers contained in the block Refresh Memory Data Bus DRVRS - RCVRS to the backplane and hence the RAM refresh memories. During RAM refresh memory readback operation the readback register drives 12 bits of data (FB00-FB11) to the IP data bus through the control board data bus receivers as previously discussed.

The block labled F, B & R Registers Write and Select Control provides the necessary read and write control for the register file containing the foreground, background and readback registers as just discussed. The block labeled Refresh Memory Data Bus DRVRS - RCVRS contains the data from the foreground or background registers to the memories as discussed above, as well as receive data from the memories which is input to the readback register after being placed on the control board internal data bus.

The blocks not yet discussed on the block diagram collectively will be referred to as the display generator. The display generator contains all of the control and address functions necessary to read or write RAM refresh memory and provide current operation point updates. The block labeled Display Generator Sequencer is the heart of the display generator circuitry. This block contains a microprogrammed sequencer that provides all of the control for the necessary functions of the display generator. It monitors address and handshake lines from the IP bus, I/O decode lines from the I/O device decode, the hold bit and additive bit from the control register block, the I/O complete line (IOCPLT) from the memory timing sequencer block, the display generators DMA source destination flags (DMASDG, DMADDG) from the DMA source and destination registers and the data of the raster data registers block (ENDARAND) to provide the necessary sequencing. It drives IP bus handshaking signals, an opcode of 2 bits (OPCD1, OPCD2) to the memory timing sequencer block, two update signals (UPDTX, UPDTY) to the COP update control block a shift signal (5SHFT) to the raster data register block, and a DMA mode control signal (FKDMA) to the I/O device decode and the F, B & R registers right and select control block. The block labled Control Register contains the additive write bit, reverse background bit, hold bit and the +X, -X, +Y, -Y control bits that operate as described in paragraph 6.2.8. The control register is loaded by the signal 7HQUNDCL from the I/O device decode with data from the control board internal data bus (PTD08-PTD14). The Raster Data Register block is loaded by the signal 7WGFONT with data from the control board internal data bus (PTD00-PTD07). The data that is normally loaded into the raster data register is either alphanumeric font data or raster data to be written into RAM refresh memory. This register is an 8 bit end around shift register. The 8 bits of data that are loaded into this

register are used in conjunction with the background bit and the additive write bit to determine whether the contents of the foreground or background registers will be written to RAM refresh memory or if neither should be written to memory and only an update performed. To accomplish this, the display generator sequencer shifts the data in this register each time a RAM refresh memory operation is executed. The block labeled COP Update Control consists of a multiplexer and gating circuitry that provides the current update control signals to the YCOP register block and the XCOP register block based upon the current operation that is to be executed. Based upon the state of the control signal MOVE the +X, -X, +Y, -Y bits from the control register or the data bits PTD08-PTD11 are used to provide the update equation for execution of the current operation, thus when the display generator sequencer attempts to update both the YCOP register and the XCOP register, this circuitry in the COP update control gates the UPDTX and UPDTY update signals as determined by the current update equation.

The core of hardware of the display generator that actually performs the RAM refresh memory addressing is designed to operate in high resolution (512 x 640) only. Therefore, the block labeled X, Y Coordinate Multiply by two is included to provide the capability for the 9000 control board to operate at three different resolutions. These resolutions are for the 9100 System (low resolution) 256 lines by 320 elements, the 9200 system (medium resolution) 256 lines by 640 elements and the 9300 System (high resolution) 512 lines by 640 elements. The block X,Y Coordinate Multiplied by two, therefore, depending upon the state of the resolution keying signals (9KLWER, 9KLWLR) and the control signal 5XCXO, multiplies both the X and Y coordinates by a multiple of two for a low resolution system, the Y coordinate by a multiple of two for a medium resolution system and neither coordinate for a high resolution system. The multiplication is actually performed by a multiplexer which left shifts the value by one binary place. Thus, the address values used internally to the display generator are always high resolution values. The coordiante values are received from the control board internal data bus (PTD00-PTD11) and after multiplication by 2 if necessary provide the values as the signals COD00-COD09. The block labled X CO-ORD Address Map converts the binary

X-coordinate value to a mixed format coordinate value that is used as the XCOP address or X origin address. This coordinate address map consists of two PROMs that perform a straight mapping function to produce an X address value that is from the least significant bit to most significant bit, a 1 bit binary number, a 4 bit binary coded decimal number and 5 bit binary number. The reason for this addressing is to implement the optimal address format to address the RAM refresh memory. As can been seen, the two least significant bits are always correct and therefore do not actually go through the address map and the upper 8 bits go through the map to produce the 8 most significant bits of the mapped value (MXD02-MXD09).

There are actually two separate addresses that are supplied to the RAM refresh memory. One address consists of the values in the XCOP register and YCOP register that are used to read or write data from or to the RAM refresh memory. The other address is that used while refreshing the CRT monitor. This addressing is supplied by the Y origin register, X origin register, Y screen refresh counter and X screen refresh counter. The block labeled XCOP register receives a mapped coordinate value as discussed above and is loaded by the signal 7WGXCOP from the I/O device decode. When a RAM refresh memory operation is performed, it also receives control from the COP update control circuitry that controls the manner in which it updates the X current operating point. This register then supplies the necessary addressing signals to address the refresh memory in the X direction on the CRT monitor. The YCOP register receives a binary value that represents the Y portion of the RAM refresh memory on the CRT monitor. This register is loaded by the signal 7WGYCOP from the I/O device decode. It then supplies the necessary Y addressing signals. The block labeled X origin register receives a mapped X address value that is used to specify the X address within RAM refresh memory that is to be displayed as the 0 element on all lines of the CRT monitor. The value in this register is then loaded into the X screen refresh counter in horizontal blanking before each raster line is scanned to insure that the scan begins at the specified element in the X origin register. This register is loaded by the signal 7HQXORG from the I/O device decode. The Y origin register receives a binary value that represents the Y address value in RAM refresh memory and specifies the first line that is to be

presented as the 0 line of the raster scan on the CRT monitor. The Y origin is loaded by the signal 7HQYORG from the I/O device decode. The value from this register is loaded into the Y screen refresh counter in vertical blanking before each scan begins. Thus, it is assured that each scan begins with the value specified in the Y origin register.

The memory timing required for this system is derived from two sets of requirements. The first set of requirements is that necessary to accomplish the defined operation of the 9000 System. The second set of requirements are those necessary to provide proper operation for the dynamic RAM memories used to implement the RAM refresh memory. The block labeled Memory Timing Sequencer is a block that accomplishes the combining of these two sets of requirements to provide the memory timing. The memory timing sequencer is set up on a ten clock cycle timing sequence. The actual sequencing is generated with a set of PROMs that are addressed by the X screen refresh counter, in particular, the 4 bit BCD value in this counter. There are four basic types of cycles that the memory timing sequencer will execute. A memory timing cycle is defined in two parts. The first part of every cycle is allocated for screen refresh addressing. The four cycles that may be performed as determined by the state of the signals OPCD1, OPCD2 and 9THSSS. These four types of cycles are as follows:

a. No RAM Refresh Memory read or write - Screen Refresh read.
b. RAM Refresh Memory read - Screen Refresh Memory Read.
c. RAM Refresh Memory Write - Screen Refresh Memory Read.
d. No RAM Refresh Memory Read or Write - No Screen Refresh Memory Read.

The block labeled Screen Refresh Read/Write Address Selector is a 12 bit multiplexer that selects between the COP addresses or the addresses supplied by the X and Y screen refresh counters as controlled by the timing signal ADSL from the memory timing sequencer. The block labeled 6 Bit Address Multiplexer takes the 12 bit address from the screen refresh read/write address selector and under the control of the selection signal TTSMC from the memory timing sequencer, multiplexes this address into two 6 bit addresses called the row address and the column address which must be supplied to the 4K dynamic RAMs on the RAM refresh memory boards. The memory timing sequencer then provides strobe signals for the row address 9DSRRAS

(screen refresh) and 9DRWRAS (read/write) and the column address 9DCAS. A write pulse is provided (9DWRTP) which is used to write data into the memory. The other timing signals that are supplied to the RAM refresh memory are used for delineating addressing. These signals are 9DADSL, 9DSRA1, 9DRFLD, 9DRWFLD, 9DSRB1, 9DSRA8, 9DSRAG1, 9DRWG1, 9DRWA1-9DRWA8, 9DRWAG1.

6.4 DEFINITION OF THE INTERNAL PROCESSOR BUS

The IP bus as defined for usage in the 9000 System consists of a 16 bit byte address (15 bit word address) bus, a 16 bit data bus and the necessary handshake lines to implement both programmed I/O and DMA transfer capability. The handshaking philosophy used on the IP bus is basically derived from a combination of the handshaking philosophy used on the existing RAMTEK interface boards. This internal processor data bus originates on the control board of the 9000 System and is driven onto the system backplane to all other boards in the system, except the interface and the RAM refresh memory boards. The IP bus is defined to originate on the control board since it contains the hand shake sequencer which performs the function of the "bus traffic cop".

6.4.1 INTERNAL PROCESSOR BUS ADDRESS LINES

The 16 IP bus byte address lines (4BADDR00-4BADDR15) provide the capability to address 64K bytes of memory. These address lines are driven through high drive capability buffer devices and are pulled up to +5 V through 1 Kilohm resistors such that multiple wired ORed sources for this address bus may exist. Since the 9000 System makes use of memory mapped I/O this address bus also is involved in executing I/O transfers with other devices in the system. As indicated by the signal names chosen for these address lines, they are positive, true signals. Hence, the address appearing on these lines is the "true" address.

6.4.2 INTERNAL PROCESSOR BUS DATA LINES

The 16 bit IP data bus signal lines (9BDATA00-9BDATA15) are negative true signals, that are pulled up to +5 V through 1 Kohm resistors providing wired-ORed capability for multiple sources. These signal lines originating on the control board are driven by high drive capability buffer devices.

6.4.3 INTERNAL PROCESSOR BUS WRITE LINE

The IP bus write line (9BWRITE) is a negative true handshake signal. This line is implemented as a wired OR signal such that multiple sources may drive it in order to communicate on the internal processor bus. This line is used for communication in both the programmed I/O and the DMA transfer modes. The source that drives this signal does so to indicate that it has valid data on the bus to be transferred. This signal is low active only while the data on the bus is valid. Further, in the case of the 8080A-1 CPU performing PIO transfers, if the 4BREADY line does not become low inactive the transfer will proceed and the 9BWRITE line will be low active for one 8080A-1 clock cycle. If the 4BREADY line does become low inactive, the 9BWRITE line will remain low active, framing the valid data for a minimum of one 8080A-1 clock cycle period after the 4BREADY line has again become high active. In the case of a DMA transfer, the 9BWRITE line must remain low active until such time as the destination for the data transfer responds with the 9BACK signal becoming low active, at which point the 9BWRITE signal may become high inactive. Refer to Figure 6-1.

6.4.4 INTERNAL PROCESSOR BUS READ LINE

The IP bus read line (9BREAD) is a negative true handshake signal. This line is implemented as a wired OR signal such that multiple sources may drive this signal to communicate on the IP bus. This signal is only used for programmed I/O by the 8080A-1 CPU. There are two cases to consider when the 8080 is attempting to read data from the data bus. The first case is that when data is available immediately upon the 8080's request. In this case, the 9BREAD line will become low active for one 8080 clock

cycle during which it reads the data from the data bus. In the second case, the source of the data that the 8080 is attempting to read will not have the data as soon as the 8080 can read it. Therefore, the 4BREADY line will become low inactive indicating to the 8080 to wait until the data is present. In this case, the 9BREAD line will become low active and remain in that state until one 8080 clock cycle after the 4BREADY line becomes high active again. The data to be read need only be stable on the address bus during this final clock cycle that the 9BREAD line is low. Refer to Figure 6-2.

6.4.5 THE INTERNAL PROCESSOR BUS READY LINE

The IP bus ready line (4BREADY) is a positive true handshake signal. This line is implemented as a wired ORed signal such that multiple sources may drive this signal to communicate on the IP bus. This signal may be used by any device in the 9000 System as a not ready/ready signal. Devices that may not be able to respond to an 8080 request immediately can pull this line low inactive to cause the 8080 to enter a wait state until such time as the device releases this line to become high active again, allowing the 8080 to complete the communication with it.

6.4.6 INTERNAL PROCESSOR BUS ACKNOWLEDGE LINE

The IP bus acknowledge line is a negative true handshake signal. This line is implemented as a wired OR signal such that multiple sources may drive it to communicate on the IP bus. This signal is used only during DMA transfer operations. As was indicated in paragraph 6.4.3, it is used by a DMA destination to respond to the 9BWRITE line being driven by the DMA source. The transition of this signal from high inactive to low active indicates to the DMA source that the DMA destination has secured the data. As long as the DMA destination holds this line low active, it indicates a busy condition to the DMA source preventing it from driving the 9BWRITE line again until this line again becomes high inactive.

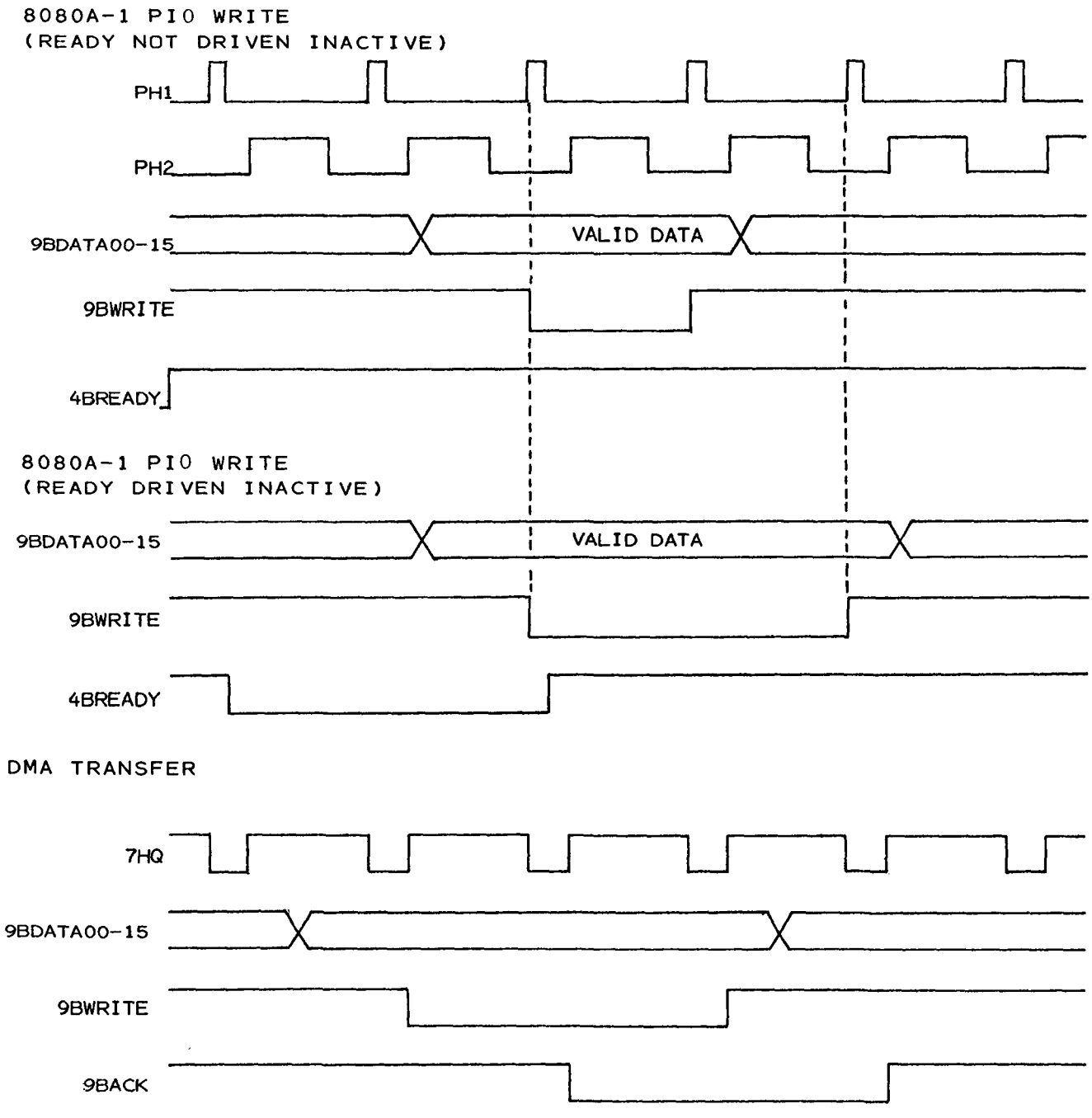
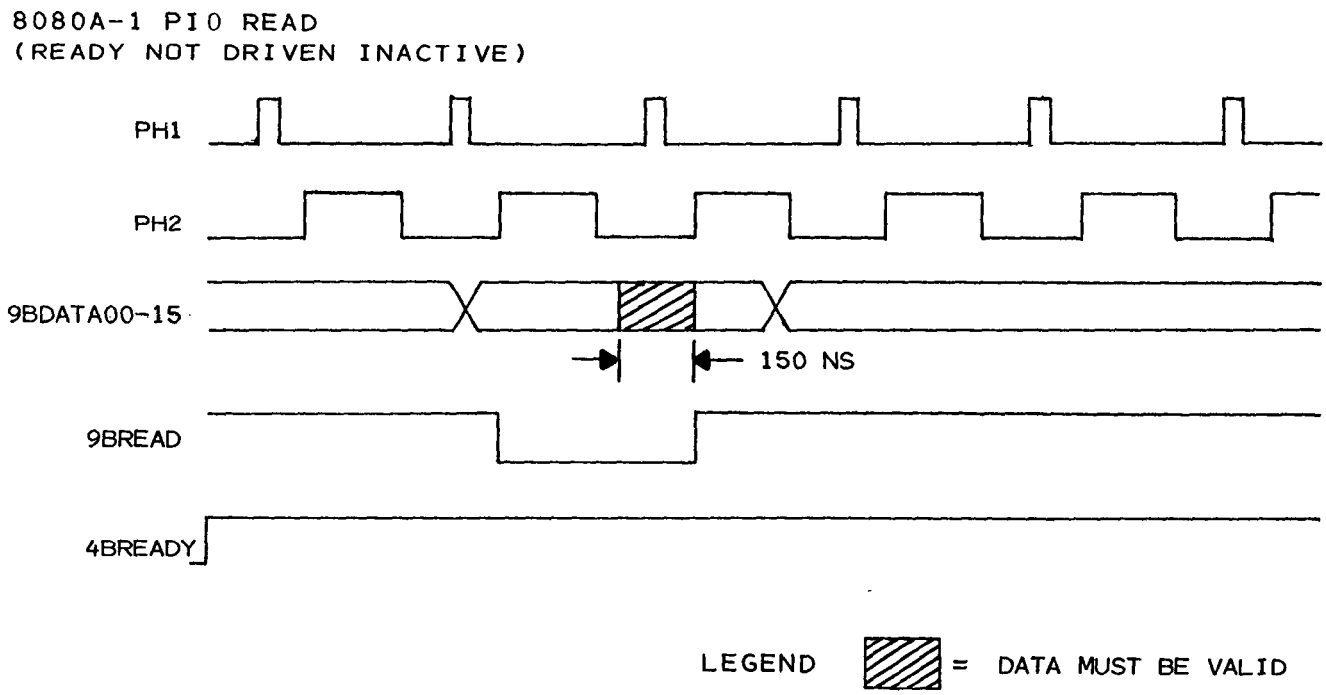


FIGURE 6-1
INTERNAL PROCESSOR BUS WRITE TIMING



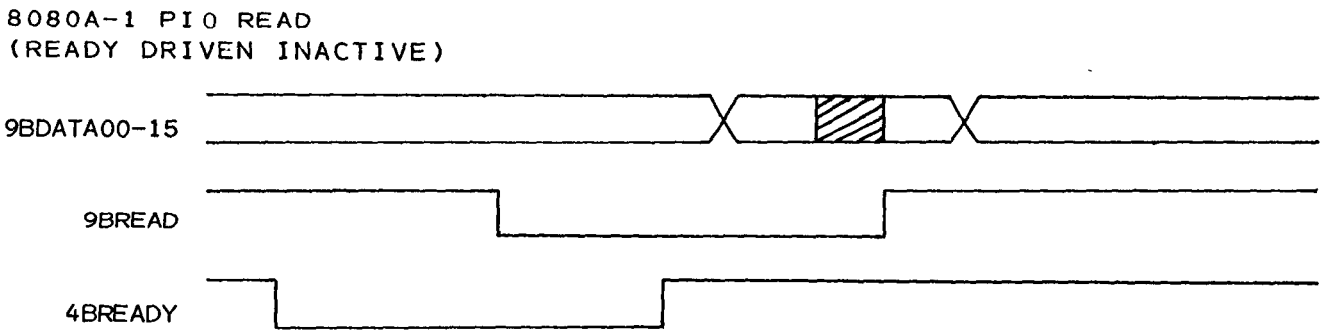


FIGURE 6-2
INTERNAL PROCESSOR BUS READ TIMING

6.4.7 INTERNAL PROCESSOR BUS SYNC LINE

The IP bus sync line (9BSYNC) is a negative true handshake signal. This line is implemented as a wired OR signal such that it may be driven from multiple sources that may wish to communicate on the IP bus. In particular, it is provided for additional 8080A-1 type sources to the IP bus. This signal is actually the sync line from the 8080A-1 microprocessor which is the marker for the beginning of every 8080A-1 machine cycle. This signal becomes low active for one 8080 clock cycle during every 8080 machine cycle at which time the 8080 data bus is presenting the current 8080 status word.

6.4.8 INTERNAL PROCESSOR BUS HOLD REQUEST LINE

The IP bus hold request line (9BDMAHR) is a negative true handshake signal. This line is implemented as a wired OR signal such that multiple sources may drive this line to communicate on the IP bus. This line is used to request the 8080 to enter the hold state as part of the sequence of signals that must occur to initiate a DMA transfer. This signal will become low active when the handshake sequencer is requesting the 8080 to enter the hold state. This signal is synchronized to the 8080 clock cycle before being driven to the 8080, and as long as it remains low active, the 8080 will continue in a hold state until such time as it becomes high inactive again.

6.4.9 INTERNAL PROCESSOR BUS HOLD ACKNOWLEDGE LINE

The IP bus hold acknowledge line (9BDMAHA) is a negative true handshake signal. This line is implemented as a wired OR signal such that multiple sources may drive the signals to communicate on the internal processor bus. In particular, this line is driven by the 8080 to indicate an acknowledgement of the hold request line and the fact that it has entered the hold state. This line will become low active and remain that way until such time that the hold request line (9BDMAHR) is released by the handshake sequencer, at which time this line will again become high inactive just before the 8080 begins program execution again.

6.4.10 INTERNAL PROCESSOR BUS DMA GO LINE

The IP bus DMA go line (9BDMAGO) is a negative true handshake signal. This line is driven by the handshake sequencer and will become low active when the handshake sequencer has determined that a DMA transfer is to be initiated and that the 8080 CPU is disconnected from the IP bus (in a HOLD state). This signal will remain low during the entire DMA transfer of whatever number of words were loaded into the DMA word counter. When the handshake sequencer has determined that the word counter has gone to zero, it will immediately release this line to the high inactive state, marking the end of a DMA transfer.

6.5 8080A-1 CPU AND ASSOCIATED HANDSHAKE CIRCUITRY AND ADDRESS-DATA DRIVER-RECEIVER NETWORK - SHEET 2

Refer to Sheet 2 of the logic schematics which shows the 8080A-1 CPU and the attendant circuitry required to interface the 8080A-1 CPU to the IP bus and hence the rest of the 9000 System. The 8080A-1 CPU (referred to as the 8080) drives its 16 bit byte address bus (A0-A15) through a set of drivers called the IP bus address drivers (locations 9W, 9V & 9U) to become the 16 bit IP byte address bus (4BADDR00-4BADDR15) whose signals are pulled up to +5 V through 1 Kohm resistors and driven off the board edge connector to the system backplane. The 8080 write line (*WR) is also driven through a driver (location 9U) to become the IP bus write line (9BWRITE) which is pulled up through a 220 ohm resistor and driven to the board edge connector. The 8080 read line (DBIN) is inverted (location 4S) and driven through a driver (location 9U) to become the IP bus read line (9BREAD) which is pulled up to +5 V through a 1 kohm resistor and driven off to the board edge connector. The enables for the IP bus address drivers (locations 9W, 9V & 9U) are controlled by signal HLDA*1 which is the 8080 hold acknowledge signal (HLDA) synchronized to the system half frequency clock (7HQ). Thus, when the 8080 is in a hold state, the address lines and the read and write lines are disconnected from the IP bus because the signal HLDA*1 will be high active disabling the address drivers. The 8080 8 bit bidirectional data bus (D0-D7) is interconnected to the IP bus 16 bit bidirectional data bus (9BDATA00-9BDATA15) via an 8 bit byte buffer register (locations 7R, 8R) and a set of 16 driver-

receivers (locations 7S, 8S, 7P & 8P). The interconnection is implemented such that two 8 bit bytes are packed to create a 16 bit word when the 8080 is writing device registers and 16 bit words are unpacked into two 8 bit bytes when the 8080 is reading a device register. When the 8080 is reading or writing its memory only the high byte portion of the IP data bus (9BDATA08-9BDATA15) is used. Since most of the device registers in the 9000 System are 16 bit registers (the 8080 usually will have to load both bytes of these registers) the 8080 executes a double byte transfer to a device register, the low byte is transfered to the byte packing buffer register first and loaded on the leading edge of the signal HQDATL, which corresponds to the leading edge of the 8080 write pulse (*WR), and only occurs when the 8080 is transferring low (even) bytes. The byte packing buffer register holds this low byte information and drives it onto the IP data bus.

Concurrently the 8080 is transferring the high byte which is driven directly on to the high byte of the IP data bus. In this manner, a 16 bit word is formed to be transferred on the IP data bus. When the 8080 transfers data in from a 16 bit I/O device register it will do so again with a double byte transfer. It will read the low byte off the IP data bus first and then it will read the high byte from the IP data bus. As can be seen for the read operation, no buffer registering is required as the device registers naturally perform this function. There are separate enables for both the drivers and the receivers of the internal processor bus data drivers and receivers. The drivers are enabled by the signal DBDAT which becomes high active to enable the driver and is the AND condition (Location 7T) of no hold acknowledge from the 8080 and an 8080 machine state of write memory. The 8080 machine state of write memory is loaded at sync time to a flip-flop at location 8E. The receivers are enabled by two separate signals RBDATH which enables the high byte and RBDATL which enables the low byte. There are two instances in which the high byte may be enabled. One is when the 8080 is attempting to read a 16 bit device register and the other is anytime that the 8080 is reading its memory. The low byte is enabled only when the 8080 is reading a 16 bit device register. The rest of the circuitry on this sheet is involved with interconnecting the 8080 to the IP bus handshake signals. At location 9M, the IP bus sync line (9BSYNC) is created as the AND condition of the 8080 sync

pulse and no hold acknowledge from the 8080. The IP bus hold acknowledge line (9BDMAHA) is created at location 9M as the AND condition of the 8080 hold acknowledge synchronized to the system half block and the device select signal from the interface board (PASEL). The reset signal for this board (RESET) and its inverse (6RESET) are created as the OR condition (location 9K) of the CPU clear signal (5CPCLR) and the power-on clear signal (9BPWCLR) which is synchronized to the system half-frequency clock to become the reset signal. The device select line from the interface (9ISEL0) is inverted at location 9H to become the signal PASEL and is ORed (Location 9K) with the IP bus hold request line (9BDMAHR) to become the signal HOLDR which is synchronized to the 8080 clocks through a flip-flop (Location 7J) to become the HOLD signal which will request the 8080 to enter a hold state. The IP bus ready line (4BREADY) is also synchronized to the 8080 clock through a flip-flop (Location 7J) to become the READY line to the 8080 which will, if inactive, request the 8080 to enter a wait state.

6.6 8080 MEMORY - SHEET 3

Sheet 3 of the logic diagram shows all of the 8080 addressable byte memory which resides on the control board. There are 2K bytes of PROM (addresses 0000-07FF hexidecimal) which are implemented with four 512 x 8 bipolar PROMs (Location 2W, 2V, 2T & 2S). There are 512 bytes of RAM (addresses 6000-61FF hexidecimal) implemented with four 256 x 4 bipolar RAMs (locations 5V, 5U, 4V & 4U). There is a 512 byte of ROM allocated to the character generator (location 2K) which is a single MOS device. As can be seen, this byte oriented memory resides on the high byte of the IP bus and is buffered onto and off of this bus by a set of buffer drivers implemented with four hex buffer drivers (Locations 5S, 5T, 5P & 5R) to create the memory data bus (9CDATA08-9CDATA15). The enables for these buffer drivers perform an AND function such that the IP bus write line can be used to control the directions that these memory buffer drivers are driving and an additional line (5MEM) from the 8080 memory enable decoders (Locations 3V, 3W) is used to enable them only when 8080 memory on this board is addressed. Upon inspection of the character generator ROM (location 2R) it will be seen that it only drives five of the eight data lines

when it is the device being accessed. Therefore, the other three data lines are pulled up to +5 V through 1 Kohm resistors as shown. The 8080 memory enable decoders are implemented with 256 x 4 bipolar PROMs that are used as decoders to decode the memory addresses and provide the appropriate enable to whichever of the memories is being addressed. The decoder of location 3W provides all four of the PROM enables (5ROM0-ROM3). The decoder of location 3V provides the RAM enables (5RAM0-5RAM1), the character ROM enable (5CHA0) and the memory buffer driver/receiver enable (5MEM). The printed circuit for this control board has been designed in such a manner as to allow several types of PROMs to be used. A table shown in the lower left hand corner lists the manufacturers and the part numbers for the different types of PROMs which can be accomodated. Also shown in the table are the modifications, trace cuts and jumpers, that are required to use each type of memory listed. The cuts are shown schematically with an "I" symbol drawn across the signal lines that must be cut. The jumpers are shown schematically by sets of E terminals across which the jumpers may be placed as required.

6.7 I/O DEVICE REGISTER DECODE AND DMA SOURCE/DESTINATION REGISTERS - SHEET 4

The I/O device register decode is implemented using three 256 x 4 bipolar PROMs (locations 3T, 3U & 4T) two 1-of-8 decoders (locations 2N, 2M) and some additional gating to finally generated all of the necessary enable gating and load pulses required for the device registers on the control board. The only address lines that are required to perform this decode function are 4BADDR01-4BADDR06 and 4BADDR15. These address lines are sufficient to cover the entire range of the addresses that have been assigned for the entire 9000 System. There are basically 3 types of decode signals provided by the I/O device decode. The first type is generated by the 256 x 4 PROMs, and are essentially pulses that will remain active as long as the I/O device address is true. The second type is generated by the 1-of-8 decoder at location 2M and the gates at location 3K. These are signals that are used as gating pulses to the registers with which they are associated. The last type of signal is generated by the 1-of-8 decoder at location 2N and the gate at location 4N. These signals are essentially system half-frequency clock width pulses that are used to

directly load the registers with which they are associated. The PROM (Location 4T) produces four decoded signals, two of which are not directly used to load the registers. These two signals are 5DENBL0 & 5DENBL1. The signal 5DENBL0 is used to enable the 1-of-8 decoder at location 2N and the signal 5DENBL1 is used to enable the 1-of-8 decoder at location 2M. It should be noted that all of these devices used in the I/O decode have the signal 9BDMAGO associated with them in order to insure that this decoder does not respond to any type of addressing that may appear on the IP address bus during DMA transfer operations. The leading edge detect circuit is implemented with two flip-flops from location 8K and the signal 9BWRITE as its input which produces a leading edge pulse that is synchronized to the system half-frequency clock. The signal name for this pulse is 5LEWRT. It is connected to the 1-of-8 decoder at location 2M to produce the actual pulse width for these gating pulses. Table 6-3 summarizes the decoded terms and their associated functions.

Sheet 4 also shows the DMA source and destination flags for the display generator and the CPU interface. These are two of the flags that exist in each of the DMA source and destination registers. In order to initiate a DMA transfer, one DMA source flag must be set and one DMA destination flag must be set. The source flag indicates the system device that will be driving data onto the IP bus. The destination flag designates the system device that will be receiving this data from the IP bus. If it is attempted to start a DMA transfer without both source and destination flags set, the DMA transfer will hang and never complete. Except for the case of writing font data to the RAM refresh memory there should never be a case where the source and destination flag for the same device are set. Also, there should never be two source flags set or two destination flags set simultaneously. The term DMASDG represents the display generators source flag and the term DMADDG represents the display generator destination flag. Similiarly, the term DMASIF represents the interface source flag and the term DMADIF represents the interface destination flag. In order to initiate a DMA, the 8080 must do the following:

a. Load the DMA source register with the flag of the device that will be sourcing data.

TABLE 6-3

I/O DEVICE REGISTER DECODE OUTPUT TERMS

FORB	FOREGROUND/BACKGROUND SELECT SIGNAL
READB	READBACK REGISTER SELECT
IFDATA	INTERFACE DATA REGISTER SELECT
IFSTAT	INTERFACE STATUS REGISTER SELECT
WORD	WORD COUNT REGISTER SELECT
UPWRT	MOVE AND WRITE REGISTER SELECT
MOVE	MOVE REGISTER SELECT
DIPLED	SELF TEST LED WRITE/SOFTWARE OPTION SWITCHES READ SELECT
XORG	X ORIGIN REGISTER SELECT
7HQUDCL	UPDATE AND CONTROL REGISTER LOAD CLOCK
7HQIF	DMA DESTINATION REGISTER LOAD CLOCK
7HQDG	DMA SOURCE REGISTER LOAD CLOCK
7HQYORG	Y ORIGIN REGISTER LOAD CLOCK
7HQPAINT	INTERRUPT REGISTER LOAD CLOCK
7HQPLANE	MEMORY PLANES SELECT REGISTER LOAD CLOCK
7HQXORG	X ORIGIN REGISTER LOAD CLOCK
7WGWORD	DMA WORD COUNTER LOAD GATING PULSE
7WGFONT	FONT DATA REGISTER LOAD GATING PULSE
7WGYCOP	Y CURRENT OPERATING POINT REGISTER LOAD GATING PULSE
7WGXCOP	X CURRENT OPERATING POINT REGISTER LOAD GATING PULSE
7WGFORE	FOREGROUND REGISTER LOAD GATING PULSE
7WGBACK	BACKGROUND REGISTER LOAD GATING PULSE
7RGIFDAT	READ INTERFACE DATA REGISTER GATING PULSE
7WGIFDAT	WRITE INTERFACE DATA REGISTER GATING PULSE

b. Load the DMA destination register with the flag of the device that will be receiving the data.

c. In some cases (as required) load the DMA address counter.

d. Always last, load the DMA word counter with the number of words to be transferred.

The reason the word counter is loaded last is because the act of loading this register actually initiates the DMA transfer.

6.8 CONTROL BOARD INTERNAL DATA BUS/FOREGROUND REGISTER/BACKGROUND REGISTER/READBACK REGISTER/MEMORY PLANE SELECT REGISTER/REFRESH MEMORY DATA BUS DRIVERS & RECEIVERS - SHEET 5

Sheet 5 of the logic schematics shows that the foreground register, background register and readback register are contained in three 4 bit by four word register files. The foreground register is word 00 binary, the background is word 01 binary, the readback register is word 11 binary and the fourth word, 10 binary, is not used. These register files have separate select codes for reading or writing. The signal 5RFWSG is the write pulse to the register file. It is a system half-frequency clock width pulse that is gated by the signal 6RFWSG which is the OR of four other gating signals. These signals are load gating pulses, two of which come from the I/O device decoder (7WGFORE, Load Foreground Register, 7WGBACK, Load Background Register), the signal 5RBRLD which comes from the memory timing sequencer which causes the readback register to be loaded and the signal 5LDFGR which comes from the display generators sequencer which causes the foreground register to be loaded while performing DMA transfers to RAM refresh memory. The high order bit of both the read and write select inputs is driven by the signal 3RGREAD which becomes high active whenever the readback register is to be interrogated either by the 8080 or during DMA transfers. The least significant bit of the read select signal to the 4 x 4 register files is driven by the signal RFORB, which is generated by the circuitry shown at the bottom of the sheet consisting of the gates at locations 3R, 4K, 4N, 3S and 4P. The signal RFORB will be a "1" if the readback register is to be interrogated (7RGREAD is low active) or the

data in the background register is to be written to the RAM refresh memory in any mode except font data write mode (FKDMA is low active and BCKGND is high active) or background data is to be written to RAM refresh memory in the font data write mode (FKDMA is high active and there is coincidence of the signals BCKGND & SRDAT). Otherwise, the signal RFORB will be low active and select the foreground register to be used in the various RAM refresh memory write modes. The least significant bit of the write selects to the 4 x 4 register files is driven by the signal SFORB generated by the gate at location 7F. This signal will be a "1" if the readback register is to be interrogated (3RGREAD is high active) or if the 8080 is attempting to write data to the background register (FORB is high active). Otherwise, the foreground register will be written when the signal SFORB is a "0". The 4 x 4 register files containing the foreground, background and readback register receive their data inputs from the control board data bus receivers (PTD01-PTD11) and drive their data (FB00-FB11) to the control board data bus drivers and also to the refresh memory data bus drivers. Hence, data from the readback register can be transferred to the IP data bus via the control board data bus drivers and the data in the foreground or background registers can be written into RAM refresh memory via the refresh memory data bus drivers. As just mentioned, the control board data bus drivers and receivers drive data from the readback register of the 4 x 4 register file onto the IP data bus (9BDAT00-9BDAT15), when the readback register is being interrogated, as indicated by the high active state of the signal 3RGREAD, which also controls the driver and receiver enables of these control board data bus drivers and receivers. When the readback register is not being interrogated the control board data bus drivers and receivers receive negative true data from the IP data bus and drive positive true data to the control board data bus (PTD00-PTD15). The refresh memory data bus drivers and receivers as mentioned above drive data from the foreground or background registers of the 4 x 4 register file to the RAM refresh memory via the refresh memory data bus (9DMDAT00-9DMDAT11). Alternately, when the readback register is being interrogated, as indicated by the low active state of the signal 7RGREAD, which controls the enables for the drivers and receivers, it receives data from the refresh memory data bus and drives it back on to the control board data bus, and hence, onto the input of the 4 x 4 register file to be loaded into the readback register. The memory plane select register, is a 12 bit register that

receives its inputs directly from the negative true data IP data bus and drives the memory plane select lines which are also negative true data on to the system backplane and on to the memory boards. This register is implemented with two hex-D flip-flop devices (locations 9R & 9S) which are loaded by the signal 7HQPLANE, driven from the I/O device decode. Also note that these registers are reset with the signal 6RESET which means that any occurance of the CPU clear signal or the system power-on clear signal will reset these registers to the state in which all memory planes are enabled.

6.9 INTERNAL PROCESSOR BUS TO INTERNAL DATA EXCHANGE BUS DATA EXCHANGE LOGIC/INTERRUPT REGISTER/SYSTEM & 8080 CLOCKS/ACKNOWLEDGE HANDSHAKE CONTROL - SHEET 6

Sheet 6 of the logic schematics shows the six hex buffer drivers (7H, 8H, 8G, 8F, 9G, 9F) that are used to perform the data exchange between the data bus of the IDE bus and the data bus of the IP data bus. The direction of the data exchange is controlled by controlling the enables of the hex buffer drivers with the signals 7WGIFDAT (write interface data register) and 7RGIFDAT (read interface data register). The interface write line 9IWRITE and the interface read line 9IREAD arrive on the control baord through pins ACC29 and ACC30 respectively. The interface write line is inverted at location 9A and ANDed with the signal 5PACK which is the acknowledge signal driven by the handshake sequencer and then driven through a hex buffer driver device to interface it to the IP data bus. The buffer driver device is enabled with the signal 7RGIFSTA which becomes low active when the 8080 addresses the interface status register. Also shown on Sheet 6 are the system and the 8080 clocks. The system clocks are received from the backplane by a set of ECL to TTL level translating receivers and developed into the clocks that are used on this board. The 8080 clocks also are received from the system backplane into a set of ECL to TTL level translating receivers and developed into the clocks used on this board. Two of the clocks 7PQONE and 7PQTWO are used to drive +12 V output clock drivers (Location 8T) which in turn drive the phase one (PH1) and phase two (PH2) clock to the 8080 Microprocessor. Shown in the upper right hand corner of this sheet is the interrupt register (location 8M) which receives the least significant

4 bits from the control board internal data bus and is loaded by the signal 7HQPAINT from the I/O device decoder. When the 8080 wishes to cause an interrupt to be sent (back) to the host CPU interface board, it may do so by loading "1" 's into the appropriate bits of this register. The four interrupt outputs from this register are inverted at Location 8L and driven onto the system backplane. The rest of the circuitry shown on this sheet is the acknowledge handshake control circuitry. Depending upon the type of operations that are being executed within the 9000 Series Systems this circuitry controls and drives the interface acknowledge (9IACK) and the IP bus acknowledge (9BACK). There are three separate cases that are handled by the control circuitry for the interface acknowledge line. These cases are treated as an OR situation as evidenced by the ORing of three inputs at Location 7L, the gate whose output is the signal IACK. These three cases are read or write PIO transfers between the 8080 and the host CPU interface, DMA transfers in which the host CPU interface is the destination and DMA transfers in which the host CPU interface is the source. Assume that a multiple word instruction will be written to the 9000 System that will cause a DMA transfer to be executed by the 9000 System. In this case, the 9000 System is waiting for an instruction, the 8080 is executing an idle loop that does nothing except test the contents of the interface status register. Looking at the acknowledge handshake control circuitry, the signal 7RGIFSTA is a signal that is driven to the NOR gate (Location 3S) from the I/O device decoder, which becomes low active each and every time the 8080 quiries the interface status register. Hence, a flip-flop (Location 1C) will be constantly set indicating the "no interface acknowledge" condition. Once the 8080 detects the presence of the write line in the status register, it will leave the idle loop and issue a read to the interface data register. The issuance of this read will cause the handshake sequencer to begin execution to control the transfer of the word from the interface data register. Once the handshake sequencer concludes handling the transfer, it will drive the signal 5PACK to its low active state resetting the flip-flop at 1C and causing the interface acknowledge line to become active. From this first word of a multiple word instruction transfer, the 8080 determines that it has further words to read and without returning to the idle loop will attempt to read these words by again interrogating the interface data register. The interrogation of the interface data register for the next and succeeding

words again causes the handshake sequencer to begin execution to handle these transfers. Before the handshake sequencer actually handles the transfer it drives the signal SEQR to the high active state which will cause the acknowledge flip-flop (Location 1C) to again be reset. It should be noted that the acknowledge flip-flop has been in the reset state, causing the interface acknowledge line to be active during the time during which the 8080 again interrogated the interface data register for the next word. Hence, in this situation, setting the interface acknowledge line active, in fact, acknowledges the transfer. Additionally, the period of time during which it remains active indicates a busy condition to the interface to prevent further transfers until the 8080 is ready. Again, once the 8080 has the second word of the multiple word instruction the handshake sequencer will drive the signal 5PACK to reset the acknowledge flip-flop, causing the second word to be acknowledged. This sequence of resetting and setting the acknowledge flip-flop will continue through all words of the multiple word instruction to be transfered. Finally after the last word of the instruction is received, the acknowledge flip-flop will be in a reset state driving the acknowledge line active indicating the 8080 is busy. Since this instruction will cause the 8080 to initiate a DMA transfer in which the interface is a participant, without returning to the idle loop, the acknowledge line flip-flop must (again) be set to indicate a not busy condition to the interface allowing the DMA transfer to proceed. This is accomplished by driving the signal 7WGWORD from the I/O device decoder (becomes low active when the 8080 loads the DMA word counter). This will cause the acknowledge flip-flop to be set, thus allowing a DMA transfer to proceed. In the second case, (interface is the source device for a DMA transfer) the interface will be driving the write line and the IP bus acknowledge line is gated to the interface acknowledge line by the gate at Location 7L. The additional conditions to allow the IP bus acknowledge line to be gated to the interface acknowledge line are that the DMA transfer has been initiated (indicated by 3BDMAGO being high active), and the signal GOIFW must be high active indicating that again the DMA transfer has been initiated and that the DMA source flag for the interface is set. In the third case, (the interface is the DMA destination device) the interface will be driving its read line when the DMA destination device drives its

write line. The circuitry (gate 4G and flip-flop 4H) will drive both the interface acknowledge line and the IP bus acknowledge line to acknowledge the transfer to both devices. The flip-flop at 4H that provides the acknowledge to both the interface and the IP bus is set when the following conditions are true on the gate at 4G. That a DMA transfer has been initiated (3BMDAGO is high), that the interface destination flag is set (DMAIF is high), that the interface is driving its read line (IFREAD is high) and that the DMA source device is driving the IP bus write line (WRITEC is high). Observing the open collector NAND gate at location 9L it will be seen that the bus is not actually acknowledged until the interface has reset its read line indicating it has received the interface acknowledge line at which time the IP bus write line will become inactive resetting the flip-flop at 4H and completing the handshake cycle for this particular word of a DMA transfer.

6.10 DIP SWITCH/SELF TEST LED/READY CIRCUITRY/CPU CLEAR CIRCUITRY - SHEET 7

The DIP switch shown on Sheet 7 may be read by the 8080 to determine what options are contained in the 9000 System and what resolution the system is. This configuration control switch is utilized as defined by the software. It is gated onto the IP bus by the drivers shown at Location 1R and 1P. One of the lines (DIP1) represents a toggle switch mounted on the edge of the board. This switch is utilized to tell the software either to run as a display system or to branch to the diagnostic tester board and execute the diagnostic test software. The signal DIP1 will be a logic "0" for the case of running as a display system. The same 8080 I/O address that is used to interrogate the option switches is also used to load a flip-flop (Location 1C) which drives an LED. This flip-flop is reset by the signals 6RESET at system power-on or whenever the reset switch is depressed. When the 8080 has executed its power-on software and determined that its PROMs are operational, it will reset this flip-flop turning the LED off.

The ready circuitry shown on this sheet consists of an inverter, two flip-flops and an open collector NAND gate (Locations 1B, 6J and 4B respectively) which receive the enable line for the character generator ROM

(5CHAO) from the 8080 memory enable decoders and drives the IP bus ready line (BREADY) to a low inactive (not ready) condition for two 8080 clock cycle periods. This causes the 8080 to enter a wait state for a minimum of two clock cycles each time it accesses the character generator ROM to allow for the 600 ns character access time of this device. Simply, this circuit is a leading edge detect circuit. The CPU clear circuitry shown on this sheet, consists of an inverter, a flip-flop and a 4 bit binary counter (Locations 8L, 8E and 4A respectively). It detects the leading edge of the CPU clear signal (9ICPCLR) and provides a pulse that is eight 8080 clock cycles wide in the form of the signal 5CPCLR which drives the reset flip-flop shown on Sheet 2 of the schematic. When the leading edge of 9ICPLCLR is detected the flip-flop 8E is reset on the first system half clock after the CPU clear line becomes active. This flip-flop in turn drives the load input of the binary counter (LCLR) to load an eight-count into the counter. The counter is clocked by phase 2 of the 8080 clocks. Once the eight-count is loaded, the high order bit of the counter (CPCLR) provides the enable to the counter to allow it to count. Through an inverter CPCLR provides the signal 5CPCLR which drives the reset flip-flop on Sheet 1 of the schematic as well as resetting the flip-flop 8E to remove the load signal from the input of the counter. The flip-flop 8E is held in hard reset state during the whole eight-cycle count. Once the counter counts from eight up through fifteen and then to zero, the signal CPCLR becomes a logic zero and hence, the clear signal 5CPCLR becomes high inactive again, reenabling the circuit to accept another CPU clear. If the CPU clear signal is as short as one system clock in width, it will produce a clean eight 8080 cycle clock wide pulse or if the CPU clear signal is several milliseconds in width, it will provide a whole series of eight 8080 clock cycle wide pulses continuing until the CPU clear signal no longer is present. The important function of this circuit is that it provides a clean, controlled width pulse to reset the 8080, thus preventing a false reset condition from occuring if the 8080 receives a reset pulse that is too short.

6.11 DMA WORD COUNTER/HANDSHAKE SEQUENCER - SHEET 8

Sheet 8 shows the handshake sequencer which is a microprogrammed device that controls the PIO handshake protocal between the IDE bus of the interface board and the IP bus of the 9000 Systems. The handshake sequencer also controls the IP bus signal sequences to initiate and terminate DMA transfers. During DMA transfers, the handshake sequencer monitors the IP bus write signal and bumps the word counter by one, for each cycle of the write signal until it detects that the word count is zero, at which time it terminates the DMA transfer. The handshake sequencer is typical of the sequencers that are used throughout the 9000 Systems. Refer to Section 3.1 of this manual for a tutorial discussion of the theory of sequencer operation. The handshake sequencer consists of three major sections, the Memory, Address Register and Condition Selects. The Memory is constructed of two 32 word by 8 bit bipolar PROMs that are connected to produce a micro-instruction word width of 16 bits. Thus, the total memory space consists of 32 (16 bit) words. The Address Register is a 5 bit register implemented with a hex D flip-flop chip (Location 3B) whose clock input is the system half clock (7HQA). Therefore, the instruction cycle time for this handshake sequencer is the cycle time of the system half clock. The reset line to the Address Register (6RESET) resets this register to zero, such that, upon system power up (or when the reset switch is depressed, or when a CPU reset occurs), the handshake sequencer will be initialized to Location 0 in its memory, at which point the idle loop resides. The two Condition Selects are implemented with one-of-eight decoders (Locations 2A and 2B), each of which provide a single output to the address register to control the state of the least significant 2 bits of the next executible instruction address. The state of each signal (CDN01 and CDN02) is controlled by the state of a selected input (one of eight possible). On each selector, six of the inputs (twelve total between the two condition selects) are used to read the state of six signals that the handshake sequencer is required to test. The other two inputs are used to force a "0" or "1", providing unconditional jump capability. Thus, the program flow of the handshake sequencer may be controlled based upon the inputs selected by the program. The program selects the input it requires via two sets of three signals direct from memory (C1S01, C1S02, C1S04, and C2S01, C2S02, C2S04). Note that some of

the conditions used as inputs to the condition selects are actually complex equations derived from the combinatorial logic (Zones D7 and D8) on this sheet. The following table summarizes the meaning of each input term.

TERM	FUNCTION
IØSTRT	IP Bus Read or Write IØSTRT = BREAD + BWRITE
9IWRITE	IDE Bus Write
5IFRØW	IDE Bus Read or Write 5IFRØW = $\overline{\text{IFRØW}}$ IFRØW = IFREAD + IFWRITE
XEQT	Interface data register addressed or the most significant byte of the word counter addressed and the IP Bus Read or Write. XEQT = (IFDATA + BADDR00•WORD) (BWRITE + BREAD)
5WCZERØ	Word Counter = 0
IDATAH	Interface data register addressed and the IP Bus Read or Write IFDATAH = IFDATA (BWRITE + BREAD)
VALIO	Valid PIO transfer VALIO = $\overline{\text{BWRITE}}$ • $\overline{\text{BADDR00}}$ + BWRITE • BADDR00 but, BWRITE = BREAD for the purposes of this test ∴ VALIO=BREAD•$\overline{\text{BADDR00}}$+BWRITE•BADDR00
IFWRITE	Interface Write

IFREAD	Interface Read
9BDMAHA	DMA Hold Acknowledge

Figure 6-3 is a flow chart which represents the sequence of operations which are executed by the handshake sequencer to perform its functions. There are basically two sequences within this flow chart. One sequence handles PIO transfers between the 8080 microprocessor and the interface data register, (hence, the interface) and the other sequence handles DMA transfers. Each instruction is represented by either a diamond, decision step, or a block. In the case of the decision step, the hexidecimal address of that instruction is represented with a two digit code at the top of the diamond. In the block, the address is represented by a two digit hexidecimal code at the left of the block. The instructions represented by blocks are basically executed for the purpose of providing control.

The handshake sequencer provides for the control to perform its functions through a set of six signals that come directly from the memory. These six signals and their functions are listed in the following table.

TERM	FUNCTION
SEQR	Reset the IDE Bus Acknowledge signal for Interface PIO transfers with the 8080 Microprocessor.
5ENDRDY	Driven active to force the IP Bus Ready signal (4BREADY) active causing the 8080 Microprocessor to leave the Wait state and complete execution of a PIO transfer with the Interface.
5PACK	Sets the IDE Bus Acknowledge Signal for an Interface PIO transfer with the 8080 Microprocessor.

4BDMAHR	Driven active to force the IP Bus Hold Request signal (9BDMAHR) active hence requesting the 8080 Microprocessor to enter the Hold state as preparation to begin a DMA transfer.
4BDMAGO	Driven active to force the IP bus DMA Go signal (9BDMAGØ) active, hence marking the beginning, and when again inactive, the end of a DMA transfer.
5BUMPWC	Driven active for an instruction cycle time to cause the Word Counter to decrement by one for every word transferred during DMA.

Table 6-4 is a listing of the handshake sequencer firmware in which the actual "1" and "0" patterns for the bipolar PROMs are shown. The patterns for the PROM at 3A is the column at the bottom of which is shown the Location, 3A, and the Ramtek part number. The same is true for the PROM at Location 3C. Table 6-5 is the handshake sequencer condition select codes in which are shown as 3 bit codes for each of the condition selects and the signal that each code selects.

It should be noted that the three most significant bits of the 5 bit next instruction address register are driven directly from memory. Each instruction directly specifies a group of four instructions from which the next instruction to be executed will be selected. Which of these four instructions is executed is dependent upon two select codes from memory thus, there are four possible cases to determine the next instruction address.

Case 1

The current instruction specifies select codes to both condition selects, 000_2 or 111_2, to force a "0" or a "1" respectively onto each of the two least significant address bits. Thus, totally specifying the next instruction address which may be anywhere in memory, and unconditional jump instruction.

	NXAD16	NXAD08	NXAD04	CDTN02	CDTN01	NEXT16	NEXT08	NEXT04	C2S04	C2S02	C2S01	C1S04	C1S02	C1S01			5ENDRDY	5PACK	5BDMAHR	5BDMAGØ	5BUMPWC	LABEL	C2 - C1	C RESULTS
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	1	IDLE	XEQT "0"	00-WAIT 10-TTCK
1	0	0	0	0	1																			
2	0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	0	1	1	0	0	1	TTCK	IFDATAH "1"	01-HWAIT 11-IFWAITE
3	0	0	0	1	1																			
4	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	GWAIT	IØSTRT "0"	00-WAIT 10-BUMP1
5	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	1	HWAIT	"0" 9BDMAHA	00-GWAIT 01-WAIT
6	0	0	1	1	0	1	0	1	0	0	0	1	1	1	0	0	1	1	1	1	0	BUMP1	"0" "1"	01-ROWZCK
7	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	0	0	1	IFWAITE	"1" VALIØ	10-JNE 11-JRØW
8	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	GØTØ1	"0" "1"	01-IFW1
9	0	1	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	1	IFWR	"0" IFDATAH	00-GØTØ1 01-WAIT
10	0	1	0	1	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	GØTØ2	"1" "1"	11-IRCPLT
11	0	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	0	0	0	1	IFRR	"1" IFREAD	10-GØTØ2 11-WAIT
12	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	RTN1	"0" "0"	00-IDLE
13	0	1	1	0	1	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	1	WWLDWN	"0" IFWRITE	00-RTN1 01-WAIT
14	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	RTN2	"0" "0"	00-IDLE
15	0	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	0	0	1	IRCPLT	"1" IFDATAH	10-RTN2 11-WAIT
16	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	WBRØW	"0" IØSTRT	00-WAIT 01-BUMP2
17	1	0	0	0	1	1	0	1	0	0	0	1	1	1	0	0	1	1	1	1	0	BUMP2	"0" "1"	01-RØWZCK
18	1	0	0	1	0	0	1	1	0	0	0	1	1	1	0	0	1	0	0	0	1	JWW	"0" "1"	01-WWLDWN
19	1	0	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	IFW2	"1" IFDATAH	10-JWW 11-WAIT
20	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	RTN3	"0" "0"	00-IDLE
21	1	0	1	0	1	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	RØWZCK	5WCZERØ IØSTRT	01,11-WAIT,00-RTN3,10-NXT
22	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	NXT	"0" "0"	00-WBRØW
23	1	0	1	1	1	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	(RØWZCK)	5WCZERØ IØSTRT	01,11-WAIT,00-RTN3,10-NXT
24	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	0	1	RØWCK	9IWRITE "1"	01-IFWR, 11-IFRR
25	1	1	0	0	1																			
26	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0	1	IFWAIT	5IFRØW "0"	10-WAIT, 00-RØWCK
27	1	1	0	1	1																			
28	1	1	1	0	0	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	IFW1	"0" IFDATAH	00-WAIT 01-JIFW2
29	1	1	1	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	1	JIFW2	"1" "1"	11-IFW2
30	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	JME	"1" "1"	11-IRCPLT
31	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	1	1	1	0	0	1	JRØW	"1" "0"	10-IFWAIT
						3A 502589								3C 502590										

TABLE 6-4

HANDSHAKE SEQUENCER FIRMWARE

TABLE 6-5

HANDSHAKE SEQUENCER

CONDITION SELECT CODES

C1 SELECT		C2 SELECT	
000	"0"	000	"0"
001	IØSTRT	001	IFDATAH
010	9BDMAHA	010	5WCZERØ
011	IFREAD	011	XEQT
100	IFWRITE	100	5IFRØW
101	VALIØ	101	9IWRITE
110	IFDATAH	110	IØSTRT
111	"1"	111	"1"

Case 2

The current instruction specifies select codes such that the LSB is forced to a "0" or a "1" and the state of the LSB+1 is dependent upon a selected signal, such that one of the two even, or odd, addressed instructions of the four instruction group will be selected based upon the state of the selected signal. This is a two way conditional branch instruction.

Case 3

The current instruction specifies select codes such that the LSB+1 is forced to a "0" or a "1" and the state of the LSB is dependent upon the select signal such that one of the two even-odd addressed instruction pairs of the four instruction group will be selected based upon the state of the selected signal. This is a two way conditional branch instruction.

Case 4

The current instruction specifies select codes such that both the LSB and the LSB+1 are dependent upon the state of two separate selected signals, such that one of the instructions in the four instruction group will be selected based upon the state of the selected signals. This is a four way conditional branch instruction.

In all four cases above, each instruction is additionally capable of providing any of the six control signals discussed previously. Upon inspection of the flow chart, Figure 6-3, it will be noted that one of the most important features of this handshake sequencer is its capability to execute a single instruction continuously while waiting for the state of a given signal to change, then to continue executing.

The last circuit to be discussed on this sheet is the 12 bit DMA word counter, which is implemented with three 4 bit binary counters (Locations 5L, 5N and 5M). The DMA word counter is loaded from the control board internal data bus when the load signal (7WGWORD) is received from the control board I/O device register decode. The act of loading the high byte of this register causes the handshake sequencer to initiate a DMA

transfer during which the handshake sequencer provides a decrement signal (5BUMPWC) to cause this counter to decrement. The handshake sequencer monitors the IP bus write signal during a DMA transfer and for each word transferred provides a system half clock cycle width decrement pulse that causes the word counter to decrement by one for each word transferred. When the DMA counter reaches the count of 0, the word count 0 signal (5WCZERO) becomes active and is detected by the handshake sequencer to terminate the DMA transfer.

6.12 DISPLAY GENERATOR SEQUENCER/RASTER DATA REGISTER - SHEET 9

Sheet 9 shows the DGS which is a micro-programmed device that controls the several modes of reading and writing refresh memory. These modes and their associated registers are listed below.

Mode 1. Cause the Refresh memory timing generator of Sheet 10 to generate a refresh memory read cycle which will place a 12 bit element value in the readback register and then update the COP registers.

Mode 2. Same as Mode 1, but do not update the COP registers.

Mode 3. Cause the refresh memory timing generator to generate a refresh memory write cycle which will place a 12 bit value into refresh memory under the influence of several registers and then update the COP registers.

Mode 4. Update the COP registers and then cause the refresh memory timing generator to generate a refresh memory write cycle which will place a 12 bit value into refresh memory.

Mode 5. Update the COP registers only.

NOTE: Modes 1-5 all require that the appropriate registers be loaded with both address bits 15 and 7 of their address set to a "1" to actually cause the display generator sequencer to execute any of these operations.

Mode 6. Enter a loop that during a DMA transfer tests the state of the additive write bit and the current bit from the raster data register, shift the raster data register left one position, and then based upon the results of the test either a) cause the refresh memory timing generator to generate a refresh memory write cycle which will write data from either the foreground or background register based upon the state of the current bit of the raster data register and the state of the background bit, and then update the current operating points, or b) update the current operating points.

Continue executing this loop once for each word of the DMA transfer until the word count goes to zero.

Mode 7. Enter a loop that during DMA transfer loads the foreground register with the data from the IP bus and then causes the refresh memory timing generator to generate a refresh memory write cycle will write the data from the foreground register into refresh memory and then update the current operating points. Continue executing this loop once for each word of the DMA transfer until the word count goes to zero.

Mode 8. Enter a loop that during a DMA transfer causes the refresh memory timing generator to generate a refresh memory read cycle which will place a 12 bit element value in the readback register and then update the current operating points. Continue executing this loop once for each word of the DMA transfer until the word count goes to zero.

The flow diagram, Figure 6-4, shows the instruction flow of the microprogram that executes the above listed functions of the display generator. This flow diagram may be read exactly the same as was Figure 6-3. There is one exception, however; that although the handshake sequencer has the capability to perform a four way conditional branch, that capability was not used in the microprogram.

The program memory is implemented with five 256 word by 4 bit bipolar PROMs (Location 6A, 6B, 6C, 6D and 6E). Thus, the program memory is a 256 location by 20 bit word width memory. The address register is

implemented as a 6 bit register with a hex D flip-flop chip. Therefore, the actual number of words of memory that may be addressed while executing the program are only 64 of the total 256 words available. The additional 2 bits of addressing to the program memory are provided with two resolution keying signals (9KLWER and 9KLWLR) which are used to select one of three programs that reside in program memory. One for each of the three 9000 Series System resolutions which this control board is capable of supporting. Sheet 12 of this schematic in zone D7 shows a small table that describes the states of these two signals for each of the system resolutions.

The condition selectors are implemented in exactly the same manner as the condition selectors of the handshake sequencer discussed in the previous paragraphs. The following table summarizes the function of each of the input terms to these condition selectors. Where combinatorial logic is used to generate one of the input signals to the selectors, the equation for that signal is also included in the table.

TERM	FUNCTION
WRITEC	IP Bus Write
IØSTRT	IP Bus Read or Write IØSTRT = BREAD + BWRITE
SDMA	Display Generator DMA operation in progress SDMA = BDMAGØ (DMADDG + DMASDG)
MVBK	Move and Write register addressed or the IP Bus Acknowledge MVBK = UPWRT + BACK
HLDRD	Hold bit from the control register; prevents updates during a PIO refresh memory read operation, not during DMA readback.

FKDMA	QUASI - DMA in progress FKDMA = BDMAGØ•DMADDG•DMASDG
IØCPLT	Signal from the Refresh Memory Timing Generator indicating the requested cycle has been executed.
UPDØ	During QUASI-DMA prevents a refresh memory write cycle from being requested but allows a COP update to occur.
MØVE	Move register addressed
VALIØ	Valid PIO transfer VALIP = $\overline{\text{BWRITE}}$ • $\overline{\text{BADDR00}}$ + BWRITE • BADDR00 but, $\overline{\text{BWRITE}}$ = BREAD for the purposes of this test ∴ VALIO = BREAD • $\overline{\text{BADDR00}}$ + BWRITE • BADDR00
DMASDG	Display Generator source flag for DMA transfer.
IØPNDG	Display Generator Sequencer has an operation to perform IØPNDG = BDMAGØ(DMASDG + DMADDG) + $\overline{\text{BDMAHR}}$ • BADDR15 • BADDR07(BWRITE + BREAD)

Table 6-6, the display generator sequencer condition select codes, lists the select codes for each of the condition selectors and the input signals that each code selects.

The display generators sequencer has nine control signals which it outputs to control the various functions required to execute its operations. The following table lists each of the signals and its functions.

TABLE 6-6

DISPLAY GENERATOR SEQUENCER

CONDITION SELECT CODES

	C2	C1
000	"0"	"0"
001	FKDMA	IØPNDG
010	HLDRD	DMASDG
011	UPWRT 4BACK	VALIØ
100	SDMA	MØVE
101	IØSTRT	UPDØ
110	WRITEC	IØCPLT
111	"1"	"1"

TERM	FUNCTION
5SHIFT	Programmed to provide a system half clock cycle width pulse that will cause the Raster Data Register to shift left one position for each pulse it receives.
UPDTX	Programmed to provide a system half clock cycle width pulse that will cause the XCOP register to be updated according to the current update equation.
UPDTY	Same as UPDTX above only for the YCOP register.
5LDFGR	Programmed to provide a system half clock cycle width pulse that will cause the foreground register to be loaded.
DGWRT	Drives the IP Bus Write signal during DMA transfer in which the Display Generator is the Source.
DGACK	Drives the IP Bus Acknowledge signal during DMA transfers in which the Display Generator is the destination.
EREADY	Driven active to force the IP Bus Ready signal (4BREADY) active causing the 8080 microprocessor to leave the Wait state and complete execution of a PIO transfer with the affected device register.
ØPCD1 & ØPCD2	A 2 bit opcode driven to the Refresh Memory Timing Generator to request Read or Write or NOP cycles.

Table 6-7 is a listing of the display generator sequencer firmware. This figure shows the "0" and "1" patterns contained in the bipolar memory. It should be noted that in a few instructions the "0" or "1" are missing. This listing is actually just a listing for one of the three programs contained in memory and therefore, these locations will be different in each program, dependent upon the system resolution. It may be determined what the proper state for these locations are by studying the flow chart, Figure 6-4, and noting that if an update must occur a "1" is required.

The raster data register, shown on this sheet, is loaded from the lower 8 bits of the control board internal data bus. When the load signal from the control board I/O device register decode (7WGFONT) occurs. The raster data register is an 8 bit shift register which is end-around shifted, to the left, by the display generator sequencer when it is executing its quasi-DMA routine.

The balance of the circuitry shown on this sheet is comprised of combinatorial logic that is used to generate the read line to the refresh memory (9DREAD). This signal becomes active only when the refresh memories are to have data read back from them. Otherwise, the signal is inactive and the refresh memories are left in the write condition.

6.13 REFRESH MEMORY TIMING GENERATOR - SHEET 10

Sheet 10 shows the refresh memory timing generator which generates a set of 10 timing signals necessary for the operation of the refresh memory. It is responsible for providing the sequences of pulses on the 10 timing signals that will cause the refresh memory to continually supply raster video data to the video board and upon demand from the display generator to perform read or write cycles that will ultimately drive data to or receive data from the IP bus. The timing signals sequences must be synchronous with the X screen refresh counter in order to implement the hardware scroll feature of the 9000 Series Systems. Thus, its addressing is partially derived from the X screen refresh address counter.

TABLE 6-7 DISPLAY GENERATOR SEQUENCER FIRMWARE

HEX	DEC	PAD32 PAD16 PAD08 PAD04 PAD02 PAD01	NXAD32 NXAD16 NXAD08 NXAD04	CS24 CS22 CS21 CS14	CS12 CS11 5SHFT UPDTX	UPDTY 5LDFGR DGWRT	DGACK EREADY ØPCD2 ØPCD1	LABEL	CTTN2	CTTN1	ØPCD	CONDITION RESULTS
0	0	0 0 0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 0	0 1 1 0	0 1 0 0	IDLE	"0"	IØPNDG	NØP	00-WAIT 01-DMACK
1	1	0 0 0 0 0 1	0 0 0 1	1 0 0 0	0 0 1 0	0 1 1 0	0 1 0 0	DMACK	SDMA	"0"	NØP	00-VALTØCK 10-FKDMACK
2	2	0 0 0 0 1 0	0 1 1 0	1 1 1 1	1 1 1 0	0 1 1 0	0 0 0 0	EXITE1	"1"	"1"	NØP	11-EXIT
3	3	0 0 0 0 1 1	0 0 1 0	0 0 0 0	0 0 1 0	0 1 1 0	0 1 0 0	IØSTRTE	"0"	"0"	NØP	00-IØSTRTW
4	4	0 0 0 1 0 0	0 0 0 0	1 1 1 0	1 1 1 0	0 1 1 0	0 1 0 0	VALIØCK	"0"	VALIO	NØP	00-EXITE1 01-IØSTRTE
5	5	0 0 0 1 0 1	0 1 1 1	0 0 0 0	1 0 1 0	0 1 1 0	0 1 0 0	DMASDGCK	"0"	DMASDG	NØP	00-DMAWE 01-DMARE
6	6	0 0 0 1 1 0	0 0 0 1	0 0 1 1	1 1 1 0	0 1 1 0	0 1 0 0	FKDMACK	FKDMA	"1"	NØP	01-MDASDGCK 11-FKDMA
7	7	0 0 0 1 1 1	0 1 1 1	1 1 1 1	0 1 0 0	0 1 1 1	0 1 0 0	FKDMA	"1"	UPDØ	NØP	10-GØW 11-JARND
8	8	0 0 1 0 0 0	0 0 1 0	1 0 1 0	0 0 1 0	0 1 1 0	0 1 0 0	IØSTRTW	IØSTRT	"0"	NØP	00-WAIT 10-WRTCK
9	9	0 0 1 0 0 1	0 0 1 1	1 1 1 0	0 0 1 0	0 1 1 0	0 1 0 0	READE	"1"	"0"	NØP	10-READ88
A	10	0 0 1 0 1 0	0 0 1 0	1 1 0 1	1 1 1 0	0 1 1 0	0 1 0 0	WRTCK	WRITEC	"1"	NØP	10-READ 11-MOVECK
B	11	0 0 1 0 1 1	0 0 1 1	0 0 0 1	0 0 1 0	0 1 1 0	0 1 0 0	MOVECK	"0"	MOVE	NØP	0 0-WRTU 01-UWRTE
C	12	0 0 1 1 0 0	0 0 1 1	0 0 0 1	1 0 1 0	0 1 1 0	0 1 1 0	WRTU	"0"	IØCPLT	WRITE	00-WAIT 01-UWRTE
D	13	0 0 1 1 0 1	0 1 0 0	0 0 0 1	1 1 1 0	0 1 1 0	0 1 0 0	UWRTE	"0"	"1"	NØP	01-WST2
E	14	0 0 1 1 1 0	0 0 1 1	1 1 1 1	1 0 1 0	0 1 1 0	0 1 0 1	READ88	"1"	IØCPLT	READ	10-WAIT 11-HOLDCK
F	15	0 0 1 1 1 1	0 1 0 0	0 1 0 0	0 0 1 0	0 1 1 0	0 1 0 0	HOLDCK	HLDRD	"0"	NØP	00WST1 10-EXITE2
10	16	0 1 0 0 0 0	0 1 0 0	0 0 0 1	1 1 1 0	0 1 1 0	0 1 0 0	WST1	"0"	"1"	NØP	01-WST2
11	17	0 1 0 0 0 1	0 1 0 0	1 1 1 1	1 1 1 0	0 1 1 0	0 1 0 0	WST2	"1"	"1"	NØP	11-UPWRTCK
12	18	0 1 0 0 1 0	0 1 1 0	1 1 1 1	1 1 1 0	0 1 1 0	0 0 0 0	EXITE2	"1"	"1"	NØP	11-EXIT
13	19	0 1 0 0 1 1	0 1 0 1	0 1 1 0	0 0 1	1 1 0	0 1 0 0	UPWRTCK	UPWRT	"0"	NØP	00-EXITE3 10-WRTAUE
14	20	0 1 0 1 0 0	0 1 1 0	1 1 1 1	1 1 1 1	1 1 1 0	0 0 0 0	EXITE3	"1"	"1"	NØP	11-EXIT
15	21	0 1 0 1 0 1	0 0 0 0	0 0 0 0	0 0 1 1	1 1 1 0	0 1 0 0	EXDMA3	"0"	"0"	NØP	00-IDLE
16	22	0 1 0 1 1 0	0 1 1 0	0 0 0 0	0 0 1 1	1 1 1 0	0 1 0 0	WRTAUE	"0"	"0"	NØP	00-WRTAU
17	23	0 1 0 1 1 1	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	0 1 0 0	RFNXW3	"1"	"1"	NØP	11-FKDMA
18	24	0 1 1 0 0 0	0 1 1 0	0 0 0 1	1 0 1 0	0 1 1 0	0 1 1 0	WRTAU	"0"	IØCPLT	WRITE	00-WAIT 01-EXITE4
19	25	0 1 1 0 0 1	0 1 1 0	1 1 1 1	1 1 1 0	0 1 1 0	0 0 0 0	EXITE4	"1"	"1"	NØP	11-EXIT
1A	26	0 1 1 0 1 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 1 0	0 1 0 0	RTN	"0"	"0"	NØP	00-IDLE
1B	27	0 1 1 0 1 1	0 1 1 0	1 1 1 0	0 1 1 0	0 1 1 0	0 0 0 0	EXIT	"1"	IØPNDG	NØP	11-WAIT 10-RTH
1C	28	0 1 1 1 0 0	1 0 0 1	0 0 0 0	0 0 1 0	0 1 1 0	0 1 0 0	DMAWE	"0"	"0"	NØP	00-DMAW
1D	29	0 1 1 1 0 1	1 1 0 0	1 1 1 0	0 0 1 0	0 1 1 0	0 1 0 0	DMARE	"0"	"0"	NØP	00-DMAR
1E	30	0 1 1 1 1 0	1 0 0 0	0 0 0 0	0 0 1 0	0 1 1 1	0 1 0 0	GØW	"0"	"0"	NØP	00-WRTSR
1F	31	0 1 1 1 1 1	1 0 0 0	0 0 0 1	1 1 1 0	0 1 1 1	0 1 0 0	JARND	"0"	"1"	NØP	01-WASTE1
			502571	502573	502572	502574	502575					

TABLE 6-7 DISPLAY GENERATOR SEQUENCER FIRMWARE (CONT'D)

HEX	DEC	PAD32 PAD16 PAD08 PAD04 PAD02 PAD01	NXAD32 NXAD16 NXAD08 NXAD04	CS24 CS22 CS21 CS14	CS12 CS11 5SHFT UPDTX	UPDTY 5LDFGR DGWRT	DGACK EREADY ØPCD2 ØPCD1	LABEL	CTTN2	CTTN1	ØPCD	CONDITION RESULTS
20	32	1 0 0 0 0 0	0 1 1 1	0 0 0 1	1 0 1 0	0 1 1 1	0 1 1 0	WRTSR	"0"	IØCPLT	WRITE	00-WAIT 01-WASTE1
21	33	1 0 0 0 0 1	1 0 0 0	1 1 1 0	0 0 1 0	0 1 1 0	1 1 0 0	WASTE1	"1"	"0"	NØP	10-WASTE2
22	34	1 0 0 0 1 0	1 0 0 0	1 1 1 1	1 1 1 0	0 1 1 0	1 1 0 0	WASTE2	"1"	"1"	NØP	11-FINISHCK3
23	35	1 0 0 0 1 1	0 1 0 1	1 0 0 1	1 1 1	1 1 0	0 1 0 0	FINISHCK3	SDMA	"1"	NØP	01-ESDMA3 11-RFNXW3
24	36	1 0 0 1 0 0	1 0 0 1	1 1 0 0	0 0 1 0	0 1 1 0	0 1 0 0	DMAW	WRITEC	"0"	NØP	00-WAIT 10-LAGØ
25	37	1 0 0 1 0 1	1 0 1 1	1 0 0 0	0 0 1	1 1 0	0 1 0 0	FINISHCK1	SDMA	"0"	NØP	00-EXDMA1 10-RFNXW1
26	38	1 0 0 1 1 0	1 0 1 0	0 0 0 0	0 0 1 0	0 0 1 0	0 1 0 0	LAGØ	"0"	"0"	NØP	00-WRTDMA
27	39	1 0 0 1 1 1	1 0 0 1	1 1 0 1	1 1 1 0	0 1 1 0	1 1 0 0	WDWNCK	WRITEC	"1"	NØP	11-WRITEC 01-FINISHCK1
28	40	1 0 1 0 0 0	1 0 1 0	0 0 0 1	1 0 1 0	0 1 1 0	1 1 1 0	WRTDMA	"0"	IØCPLT	WRITE	00-WAIT 01-GØØN1
29	41	1 0 1 0 0 1	1 0 0 1	1 1 1 1	1 1 1 0	0 1 1 0	1 1 0 0	GØØN1	"1"	"1"	NØP	11-WDWNCK
2A	42	1 0 1 0 1 0										
2B	43	1 0 1 0 1 1										
2C	44	1 0 1 1 0 0	0 0 0 0	0 0 0 0	0 0 1 1	1 1 1 0	0 1 0 0	EXDMA1	"0"	"0"	NØP	00-IDLE
2D	45	1 0 1 1 0 1										
2E	46	1 0 1 1 1 0	1 0 0 1	0 0 0 0	0 0 1 1	1 1 1 0	0 1 0 0	RFNXW1	"0"	"0"	NØP	00-DMAW
2F	47	1 0 1 1 1 1										
30	48	1 1 0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 0	0 1 1 0	0 1 0 0					
31	49	1 1 0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	GØØN2	"0"	"0"	NØP	00-4BACKUCK
32	50	1 1 0 0 1 0	1 1 0 0	1 1 1 1	1 0 1 0	0 1 1 0	0 1 0 1	DMAR	"1"	IØCPLT	READ	10-WAIT 11-4BACKDCK
33	51	1 1 0 0 1 1	1 1 0 1	0 0 0 0	0 0 1 0	0 1 1 0	0 1 0 0	4BACKDCK	4BACK	"1"	NØP	11-WAIT 01-GØØN2
34	52	1 1 0 1 0 0	1 1 0 1	0 1 1 0	0 0 1 0	0 1 1 1	0 1 0 0	4BACKVCK	4BACK	"0"	NØP	00-WAIT 10-GØØN3
35	53	1 1 0 1 0 1	1 1 0 0	1 0 0 0	0 0 1 1	1 1 1 0	0 1 0 0					
36	54	1 1 0 1 1 0	1 1 0 1	1 1 1 1	1 1 1	1 1 0	0 1 0 0	GØØN3	"1"	"1"	READ	11-FINISHCK2
37	55	1 1 0 1 1 1	1 1 0 1	0 1 1 1	1 1 1 0	0 1 1 0	0 1 0 0	FINISHCK2	SDMA	"0"	READ	00-EXDMA2 10-DMAR
38	56	1 1 1 0 0 0										
39	57	1 1 1 0 0 1										
3A	58	1 1 1 0 1 0										
3B	59	1 1 1 0 1 1										
3C	60	1 1 1 1 0 0										
3D	61	1 1 1 1 0 1										
3E	62	1 1 1 1 1 0										
3F	63	1 1 1 1 1 1										
			502571	502573	502572	502574	502575					

In understanding the operation of the refresh memory timing generator and also the refresh memory it is very important to understand the requirements for reading and writing the 4096 (16 pin) dynamic RAMs that are used to implement refresh memory. Since these 4096 dynamic RAMs are housed in a 16 pin package they require their 12 bit address to be multiplexed into two 6 bit addresses. These 16 pin dynamic RAMs may be read by maintaining the write enable line ($\overline{WE}$) high during $\overline{CAS}$.

The output pin of the selected device will unconditionally go to a high impedance state immediately following the leading edge of $\overline{CAS}$ and remain at this state until valid data appears at the output at access time. The selected output data is internally latched and will remain valid until a subsequent $\overline{CAS}$ is given to the device by a read/write or refresh cycle. A write cycle is performed by bringing the write cycle ($\overline{WE}$) low before or during $\overline{CAS}$. If $\overline{WE}$ goes low at or before $\overline{CAS}$ goes low the input data must be valid at or before the leading edge of $\overline{CAS}$. If $\overline{WE}$ goes low after $\overline{CAS}$, data must be valid at or before the leading edge of $\overline{WE}$. Data out goes to a high impedance state following the leading edge of $\overline{CAS}$. Each of the 64 rows internal to the dynamic RAM must be refreshed every 2 ms in order to maintain the data. Any data cycle read or write refreshes the entire selected row, which is defined by the low order row address bits. The refresh operation is independent of the chip select line. The chip is addressed with the use of two negative going TTL clock pulses called row address strobe ($\overline{RAS}$), and column address strobe ($\overline{CAS}$). These clocks are used to strobe the two sets of 6 bit addresses into the internal address buffer registers. The first clock $\overline{RAS}$, strobes in the six lower address bits (A0-A5) which selects 1-of-64 rows and begins the timing cycle which enables column sense amplifiers. The second clock, $\overline{CAS}$, strobes the six high order address bits (A6-A11) to select one of the 64 column sense amplifiers and chip select ($\overline{CE}$) which enables the data output buffer. A memory cycle always begins with addresses stable and a negative transition of the $\overline{RAS}$. The detail timing and operation of the 4096 dynamic RAM are described in the data sheets from the several manufacturers of these devices.

The outputs of the refresh memory timing generator which drives the timing signals to the memories are based on the operation of these dynamic RAMs described above. The following table summarizes each of the memory plane handshake control lines and its function.

TERM	FUNCTION
9DSRRAS	Screen Refresh RAS
9DRWRAS	Read/Write RAS
9DCAS	CAS
9DWRTP	Write Control Pulse
9DADSL	Read/Write Address/Screen Refresh Address Select
9DSRA1	Address Select Control
9DRWAG1	Address Select Control

The 10 timing signals described above are generated based on a timing cycle that is 10 system half clocks in length. Each of these 10 states within this cycle are defined by the state of the BCD portion of the X screen refresh counter of sheet 12. The four signals, SRA1, SRA2, SRA4 and SRA8 are generated at location 1E by a 4 bit BCD counter that is part of the X screen refresh counter and is used as a partial address to the three 256 word by 4 bit bipolar PROMs (Locations 7A, 7B and 8A) of the refresh memory timing generator. There are four basic types of cycles used, each of which is 10 states in length. They are as follows:

a. *Screen refresh only cycle.* The operation code for this cycle is 00. This cycle provides the timing necessary to only produce a screen refresh access.

b. *Read cycle.* The operation code for this cycle is 01. This cycle provides the timing necessary to produce a read access and then a screen refresh access.

c. *Write cycle.* The operation code is 10. This cycle provides the timing necessary to produce a write access and then a screen refresh access.

d. *NOP.* The operation code for this cycle is 11. This cycle provides the timing necessary to only allow additional cycles to continue, but provides no timing for any memory accesses.

Sheet 16 of the schematic shows a timing example of each of the cycles 1, 2 and 3. The following table lists the terms for the 10 signals produced by the bipolar PROMs and their function.

TERM	FUNCTION
LDNWR	Load the opcode for the next timing cycle.
IØCPLT	Indicate to the Display Generator Sequencer that the requested cycle is complete.
5RBRLD	Load the Readback register with the 12 bit value received from refresh memory.
TTSCMX	This signal is used as the source for the generation of the TTSMC signal which is used to multiplex the 12 bits address into two 6 bit portions known as the row address and column address.
ADSLX	This signal is used as the source for the generation of the 9DADSL signal which is used to select between the read/write address and the screen refresh address of the read and write cycles.

*DSRX1	This signal is used as the source for the generation of the 9DSRA1 signal which is used by the refresh memory to discern states 8 and 9 of the timing cycle.
*RA1	This signal is the source for the generation of the 9DSRRAS signal which is the $\overline{RAS}$ signal used to perform screen refresh accesses.
*RA2	This signal is the source for the generation of the 9DRWRAS signal which is the $\overline{RAS}$ signal used to perform read or write cycles.
*RA4	This signal is the source for the generation of the 9DCAS signal which is the $\overline{CAS}$ signal to refresh memory.
*RA8	This signal is the source for the generation of the 9DWRTP signal which is the write enable signal to refresh memory.

The signals in the above table that are indicated as being sources for other signals are driven through a section of logic called the combinatorial timing logic which modifies the pulse widths and shifts them in time before sending them off the control board onto the system backplane.

As mentioned previously, the four BDC state signals are a partial address to the timing sequence PROMs, and are the least significant 4 bits of the total address that is applied to these PROMs. The next two most significant bits (RADD1 and RADD2) are driven from an address register (Location 7C) that is enabled to load by the signal LDNWR from the timing sequence PROMs. The information that is loaded into the address register to provide these 2 bits of address are the 2 bits of operation code that are driven from the display generator sequencer, which requests the type of refresh memory timing cycle that is to be generated. When generating timing for the refresh memory, it must be guaranteed that the $\overline{RAS}$ and $\overline{CAS}$ signals that are

applied to the refresh memory RAMs must always be of the correct length. The hardware scroll feature of the 9000 Series System operates in such a fashion that it is possible in horizontal blanking to generate $\overline{RAS}$ and $\overline{CAS}$ signals that do not meet the necessary timing criteria. Therefore, the combinatorial logic that appears in front of the address register (zones C7, C8, D7 and D8) receives the horizontal sync pulse from the sync generator and uses this to start a timing cycle in horizontal blanking. This timing cycle will force an operation code of 11 into the address register such that there cannot be any access cycles during the period when the X screen refresh counter is being loaded from the X origin register.

Another RAM refresh memory timing requirement that could be violated by the scroll feature of the 9000 Series System is the 2 ms refresh requirement for the dynamic RAMs. The violation of this requirement can occur if the X origin register is changed too rapidly. Therefore, the circuitry shown (zones A7, A8, B7 and B8) which consists of a 4 bit binary counter and some combinatorial logic gates operates in such a fashion as to only allow the X origin register to be loaded once per 12 raster scan line times. Thus, the refresh memory always has had a complete memory refresh before the origin is changed. The reason that it is possible to violate the memory refresh requirements is because the design depends upon the continuing screen refresh operation to provide refresh for the RAM memories, and changing the origin too quickly could introduce the possibility of skipping some addresses for a period longer than 2 ms.

6.14 CONTROL REGISTER/UPDATE EQUATION SELECT/XCOP REGISTER/YCOP REGISTER/REFRESH MEMORY ADDRESS MULTIPLEXING - Sheet 11

Sheet 11 shows the control register (Location 6K) which is loaded from the control board internal data bus when the load signal (7HQUDCL) becomes low active. The load signal is driven from the control board I/O device decode. The Update Equations Select is implemented with a quad one-of-two multiplexer (location 5K) and selects either the update equation bits from the control register or the update equation bits from the IP Bus when the move or the move and write register are loaded. The select line is

driven by the MOVE signal from the I/O device decode. This signal is true when either the move register or the move and write register is loaded. The outputs from the update equation select are next driven through combinatorial logic that combines the selected update equation with update pulses (UPDTX and UPDTY) from the display generator sequencer. This combinatorial logic then supplies a direction control and count pulse to both the X current operating point register (XUP and XUDE) and the Y current operating point register (YUP and YUDE).

The X current operating point register is implemented with three 4 bit binary counters and one 4 bit BDC counter (Locations 2J, 2K, 2G and 2H). This register is actually an up/down counter that can be loaded with a value from the X coordinate address map of sheet 12 that maps the binary value of the X address into a value that can be directly used to address refresh memory. This value that is loaded into the X current operating point register is actually a mixed format value. Beginning with the least significant bit, this value is a 1 bit binary value, then 4 bits of a BDC value and then 5 bits of a binary value. The least significant bit of this value is ignored in all resolutions except the high resolution case (512 lines by 640 elements). This register is loaded with the value from the X coordinate address map with its load signal (7WGXCOP) from the I/O device register decode. When the display generator issues an update pulse to this register, it will either be incremented or decremented depending upon the state of the current update equation. The six least significant bits from this register are sent directly from the control board to the refresh memory boards where they are used to address individual elements on the screen. The four higher order bits from this register are included with additional bits from the Y current operating point register and are multiplexed into the two 6 bit addresses required by the dynamic RAMs.

The YCOP register is implemented with three 4 bit binary counters (Locations 3G, 3H and 3J). This register also is implemented as an up/down counter. It is loaded from the X-Y coordinate multiply-by-two when the load signal (7WGYCOP) is driven from the I/O device decoder. When the display generator sequencer drives the Y update signal, this counter is

either incremented or decremented as controlled by the Y update equation. The reason that both the XCOP register and YCOP register are loaded with values that are first driven through the X-Y coordinate multiply-by-two is that these registers always operate as though they were in a high resolution system. Therefore, in the low and medium resolutions systems, as appropriate the values are multiplied by two before these registers are loaded. All but one of the outputs of the YCOP register are driven to the refresh memory address multiplexing. The one output that is not driven there is the signal RWFLD, which is driven to the refresh memories where it is used as part of the memory addressing directly. The first stage in the refresh memory address multiplexing is driven by select signal ADSL from the refresh memory timing generator. This signal selects between the addresses supplied by the XCOP register and YCOP register and the addresses supplied by the X screen refresh counter and the Y screen refresh counter of sheet 12. Both sets of addresses that are delivered to this multiplexer are 12 bits in width. The next stage of multiplexing takes whichever 12 bit address is driven from the first stage and multiplexes it into two 6 bit addresses which are driven off the board and down the system backplane to the refresh memories.

These two 6 bit addresses are the row address and the column address that are used by the 4K dynamic RAMs. The select signal for this second stage of the address multiplexing also comes from the refresh memory timing generator in the form of two signals which are ANDed together, (TTSMCY and TTSMCZ).

6.15 X,Y COORDINATE MULTIPLY BY TWO/X COORD ADDRESS MAP/Y ORIGIN REGISTER/X ORIGIN REGISTER/X SCREEN REFRESH ADDRESS COUNTER/Y SCREEN REFRESH ADDRESS COUNTER - Sheet 12

Sheet 12 shows the X and Y coordinate multiply-by-two which is implemented with three quad, one-of-two multiplexers (Locations 1L, 1N and 1M). This circuit receives all X-Y coordinates that are destined to be loaded into the X origin register, Y origin register, X current operating point register or Y current operating point register and depending upon the 9000 System resolution, either passes the coordinate onto the register

into which it will be loaded or multiplies the coordinate by a factor of two, by simply shifting the value one position towards the MSB of the value. The control for the X-Y coordinate multiply-by-2 is derived from two keying signals (9KLWER and 9KLWLR) the state of which are controlled by jumpers on the system backplanes. Also included in this control circuit, is the signal 5XCXO from the I/O device decoder which indicates whether the value that is presently being passed through the multiply-by-2 is an X value or a Y value. The small table shown in the schematic just above the X-Y coordinate multiply-by-2 circuit indicates the state of the keying signals for each of the system resolutions. The intent of this circuit is that, in a 2x3 resolution system, both the X and the Y coordinate values will be multiplied by two. In a 2x6 system, only the Y coordinate value will be multiplied by two, and in a 5x6 resolution system, neither coordinate value will be multiplied by two. Again, the reason for this circuit is that the registers whose values it controls are all implemented as high resolution (5x6) registers.

The X coordinate address map receives all of the X values that will be loaded either to the X origin register or the X current operating point register and maps them into a mixed format value that is used to directly address refresh memory. The use of this mapping scheme is dictated by the need to provide the minimum wastage of memory on the refresh memory boards. This circuit is implemented with two, 256 word by 4 bit bipolar PROMs which receive as their address the binary value of the X coordinates and map them into the mixed format value. As can be seen, two of the bits of the binary value are always correct and are passed directly to the X origin register and the X current operating point register. The rest of the bits of the binary value drives the PROMs to produce the mapped value. The final mapped value is a 9 bit value, the least significant bit of which is a binary value, the next most significant 4 bits are a BCD value and the five most significant bits are a binary value. For a further discussion of how this value relates to refresh memory, see paragraph 6.14.

The Y origin register is loaded from the X-Y coordinate multiply-by-two with its binary value when the load line (7HQYORG) is driven from the I/O

device register decode. The Y origin register is implemented with two hex D flip-flop chips (Locations 2E and 2F). The output of the Y origin register is driven to the Y screen refresh address counter.

The X origin register receives its value, the mapped value discussed previously, and is loaded by the load line (7HQXORG) which is driven by the I/O device register decode. The X origin register is implemented with two hex D flip-flop chips (Locations 1H and 1G). The output of the X origin register is driven to the X screen refresh counter, with the exception of the least significant bit, XØR00 which is driven to the video board, the purpose of which is to indicate to the video board whether to begin each line with an even element or an odd element.

The 9000 System has a hardware scroll capability, i.e., any X,Y element in the screen refresh memory may be displayed on the screen as the X = 0, Y = 0 screen element. This is accomplished through the existence of the X origin register and the Y origin register. The value in the X origin register is loaded into the X screen refresh counter by a signal from the sync timing board (9THMMM) during every horizontal blanking period and the Y origin register is loaded into the Y screen refresh counter by another signal from the sync timing board (9TVMMM) every vertical blanking period. To place a specified X-Y element in refresh memory such that it appears at X = 0, Y = 0 on the screen, the equations described in Sections 6.2.11 and 6.2.12 must be used to compute the proper values to be placed in the origin registers such that the refresh memories are able to access the proper data before the visible portion of the raster. The X screen refresh counter is loaded in every horizontal period with the X value contained in the X origin register. Then it proceeds to count at the low X resolution element rate (system half clock rate) throughout the line scan period, providing the X refresh address that is used to access refresh memory. As previously mentioned, the four least significant bits of this counter provide the state timing to the refresh memory timing generator, while the next most significant bit (SRAG1) is sent directly to the refresh memory boards and is used in addressing the refresh memory. The four most significant bits of the counter as mentioned previously are driven to the refresh memory address multiplexing

to form part of the 12 bit address that is supplied to the dynamic RAMs. The X screen refresh address counter is implemented with two 4 bit binary counters and one 4 bit BCD counter (Locations 1D, 1F and 1E).

The Y screen refresh address counter is implemented with three 4 bit binary counters (Locations 3D, 3E and 3F). This counter is rather unusual counter in the respect that the most significant bit of this counter is loaded from the least significant bit of the Y origin register. This is because of the manner in which data is presented on a raster scan monitor in the interlaced mode, in which the data is presented in two fields such that all of the even lines of data are presented first in one field and then all the odd line data are presented in the second field. Therefore, this counter must count in an unusual manner to provide the proper addressing to refresh memory. In order to allow this to occur and also provide the capability for scrolling data in the Y direction also, some additional combinatorial logic is required to help control this counter (zones A4, A5 and B4, B5). When this control board is used in systems which are of a low line resolution type the combinatorial logic is disabled and the most significant bit of the counter is ignored by the rest of the system. Therefore, the least significant 8 bits of the counter count in what could be termed a normal manner. The counter is incremented in horizontal blanking by a signal from the sync board (9THMMM). When this control board is used in a system with a high line resolution, such that the refresh memory data must be presented in an interlaced manner, the most significant bit of the counter becomes active and is used by the rest of the system. To accomplish scrolling in the Y direction, the Y value in the origin register is loaded into the least significant 8 bits of this Y screen refresh address counter in every vertical blanking period. However, it is necessary to control the most significant bit of this counter in a different manner. It is necessary to know what will be the next field that the monitor will display. This information is provided by a signal from the sync timing board (9TVFFF). When it is known that the next field to be presented on the monitor is to be the even field, the signal 9TVFFF will be high inactive. This causes the signal *FLP from the gate at location 7G to load the most significant bit of the Y screen refresh address counter with the value of YØR00 from the Y origin register. Then before the odd

field is presented, the most significant bit of this counter must be incremented which is accomplished by the gate 7G whose output is *FCT. When the signal *FCT occurs, and if the value of YØROO that was loaded into the most significant bit of the counter is a "1", it will also be necessary to increment the least significant 8 bits of the counter in order to maintain a proper address to refresh memory. This is accomplished by supplying the signal *FCT to the control circuity in zones A4 and A5 which ultimately supplies an extra count pulse to the least significant 8 bits of the counter. The least significant 8 bits of the Y screen refresh counter are driven to the refresh memory address multiplexing on sheet 11 to form the rest of the 12 bit address for accessing refresh memory. The most significant bit of the Y screen refresh counter (RFLD) is driven directly off the control board to the refresh memory where it is used in addressing the dynamic RAMs.

	ACC	ACW		BCC	BCW
1	ZGND	ZGND	101	#BREAD	9BREAD
2	ZGND	ZGND	102	#BWRITE	9BWRITE
3	ZP05	ZP05	103	#BACK	9BACK
4	ZP05	ZP05	104	#BREADY	4BREADY
5	ZN12	ZN12	105	#BDMAHR	9BDMAHR
6	ZN12	ZN12	106	#BDMAHA	9BDMAHA
7	ZN05	ZN05	107	#BDMAGO	9BDMAGO
8	ZP12	ZP12	108	#DREAD	9DREAD
9			109	#DRFLD	9DRFLD
10	PTBQ	NTBQ	110	#DRWB1	9DRWB1
11	PTHQ	NTHQ	111	#DRWFLD	9DRWFLD
12	9ISELO	#ISELO	112	#BSYNC	9BSYNC
13		9DSRA8	113	9DRWA4	9DRWA8
14		#DSRA8	114	9DRWA2	9DRWA1
15	#DSRB1	9DSRB1	115	#DRWAG1	9DRWAG1
16		9DADSL	116	#DSRRAS	9DSRRAS
17		#DADSL	117	#DSRAG1	9DSRAG1
18		9DSRA1	118	9DWE09	9DWE03
19		#DSRA1	119	9DWE08	9DWE03
20	9IINT4	4STLED	120	9DWE11	9DWE05
21	9IINT5		121	9KLWER	
22			122	9DWE07	9DWE01
23	#DRWRAS	9DRWRAS	123	9DWE10	9DWE04
24	9ICPCLR	#ICPCLR	124	9DWE06	9DWE00
25	#DCAS	9DCAS	125	9DMDAT06	9DMDAT00
26	9IACK	#IACK	126	9DMDAT07	9DMDAT01
27	#DWRTP	9DWRTP	127	9DMDAT11	9DMDAT05
28	9BSWCLR	#BSWCLR	128	9DMDAT09	9DMDAT03
29	9IWRITE	#IWRITE	129	9KLWLR	
30	9IREAD	#IREAD	130	9DMDAT10	9DMDAT04
31	9IDATA01	9IDATA00	131	9DMDAT08	9DMDAT02
32	9IDATA03	9IDATA02	132	9BDATA01	9BDATA00
33	9IDATA05	9IDATA04	133	9BDATA03	9BDATA02
34	9IDATA07	9IDATA06	134	9BDATA05	9BDATA04
35	9IDATA09	9IDATA08	135	9BDATA07	9BDATA06
36	9IDATA11	9IDATA10	136	9BDATA09	9BDATA08
37	9IDATA13	9IDATA12	137	9BDATA11	9BDATA10
38	9IDATA15	9IDATA14	138	9BDATA13	9BDATA12
39	9BPWCLR	#BPWCLR	139	9BDATA15	9BDATA14
40	9THSSS	#THSSS	140	4BADDR01	4BADDR00
41	9IINT0		141	4BADDR03	4BADDR02
42	9IINT1	9DAD00	142	4BADDR05	4BADDR04
43	9TVFFF	#TVFFF	143	4BADDR07	4BADDR06
44	9DAD02	9DAD01	144	4BADDR09	4BADDR08
45	9DAD04	9DAD03	145	4BADDR11	4BADDR10
46		9DAD05	146	4BADDR13	4BADDR12
47	9THMMM	#THMMM	147	4BADDR15	4BADDR14
48	9TVMMM	#TVMMM	148	ZP05	ZP05
49	PTPQØNE	NTPQØNE	149	ZGND	ZGND
50	PTPQTWØ	NTPQTWØ	150	ZGND	ZGND

TABLE 6-8 CONTROL BOARD SIGNAL TERM PINOUT

7.0 VIDEO TYPE I INTRODUCTION

Refer to the block diagram shown in Figure 7-1.

The video board (Type 1) can provide (depending on the number of memory planes in the system) a maximum of 12 video output data channels at a resolution of 320 elements by 256 lines, or 640 elements by 256 lines or 640 elements by 512 lines. External outputs are provided for composite blanking, composite sync, vertical drive, and horizontal drive. Four independent cursors can be displayed on any of the video channels desired (the Serial Link option board must be included in the system). Two overlays may be mixed with any of the outputs desired.

Three 4 bit digital to analog converters (DAC) will provide 16 levels of intensity to three DAC video output channels (which are additional to the 12 output channels provided for above). If one DAC video channel of greater levels of intensity resolution is desired, two of the 4 bit DACs may be modified to produce one 5, 6, 7 or 8 bit DAC.

Reference drawings Logic Diagrams 502445, Assembly Drawing 502446, Internal Video Cable (9100 and 9200) 502337, Internal Video Cable (9300) 502789.

7.1 FUNCTIONAL DESCRIPTION (REFERENCE DRAWING (5024485)

7.1.1 ELEMENT AND FIELD SELECTION (SHEETS 3 AND 4)

In a high resolution system (640 elements by 512 lines) four memory data lines are necessary for one video data channel. As an example, the input memory data lines for channel 00 are as follows:

9MVA00E - A Raster Field, E = even element
9MVA00Ø - A Raster Field, O = odd element
9MVB00E - B Raster Field, E = even element
9MVB00Ø - B Raster Field, O = odd element

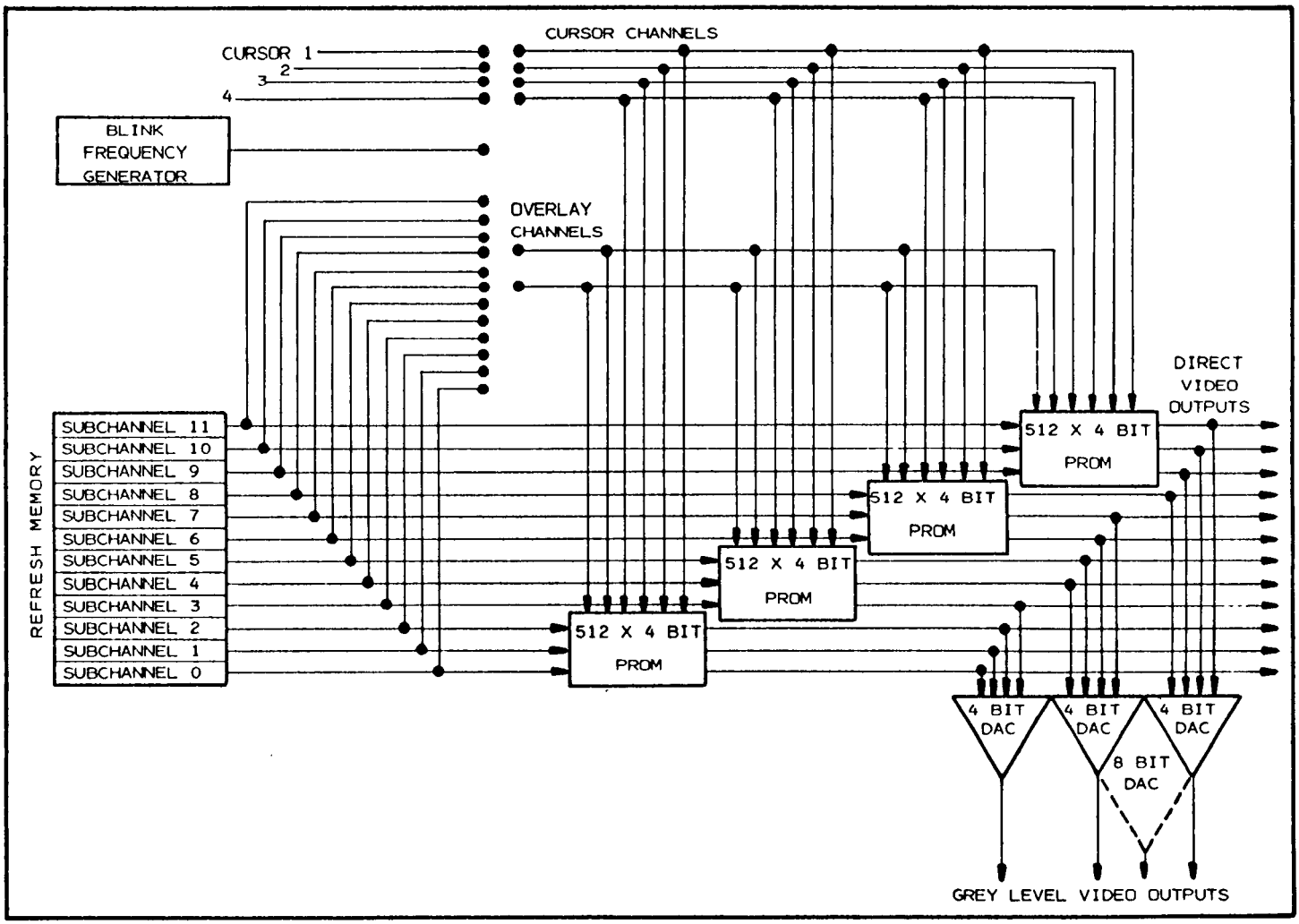


FIGURE 7-1 SIMPLIFIED BLOCK DIAGRAM OF THE VIDEO I

In a medium resolution system (640 elements by 256 lines) two memory data lines are necessary for one video output. As an example, the input memory data lines for channel 01 are as follows:

9MVA01E - A Raster Field, E = even element
9MVA01Ø - A Raster Field, O = odd element
9MVB01E - Not used in medium (or low) resolution
9MVB01Ø - Not used in medium (or low) resolution

In a low resolution system (320 elements by 256 lines) one memory data lines is necessary for one video data channel. As an example, the input memory data lines for channel 02 are as follows:

9MVA02E - A Raster Field, E = even element
9MVA02Ø - Not used in low resolution
9MVB02E - Not used in low (or medium) resolution
9MVB02Ø - Not used in low (or medium) resolution

The three examples above serve to illustrate the function of the 74S253 element and field selectors (locations 6A, 6B and 6C). These selector control signals 2ESEL will select even memory data when equal to 0 volts and 6FLDSEL will select 'A' raster field when equal to 0 volts.

Composite blanking (DCBBB) is also applied to the enable inputs of the 74S253 selectors. The purpose of DCBBB is to disable the outputs during retrace time.

The selected memory data becomes synchronized with the system full clock (7VFUL) through the reclock registers 6D and 5G. The data 6DAT00-6DAT11 is applied to the mixing PROMs and the overlay cursor patching network.

7.1.2 OVERLAYS, CURSORS & BLINK MIXED WITH DATA (SHEETS 3 AND 4)

Mixing of data occurs within the PROMs. As an example, refer to PROM 5J on sheet 4. The three data lines to be mixed with overlays and cursors are 6DAT06, 6DAT07 and 6DAT08. Three outputs out of the PROM are VID06,

VID07 and VID08. Notice that the data has been inverted in addition to being mixed with overlays and cursors. Four cursor lines are inputs to the PROM (7CURS1, 7CURS2, 7CURS31, and 7CURS41). Notice that 7CURS31 and 7CURS41 are patched in and may be removed to allow two more overlay inputs to the PROM. Two more lines that have been pulled up to +5 volts (6ØLAY1 and 6ØLAY2) are also inputs to the PROM and may be patched in to allow two additional overlays. The patching network is located at 4J and is shown in Figure 7-2 as seen from the component side. The signals at the network designators are shown in Table 7-1.

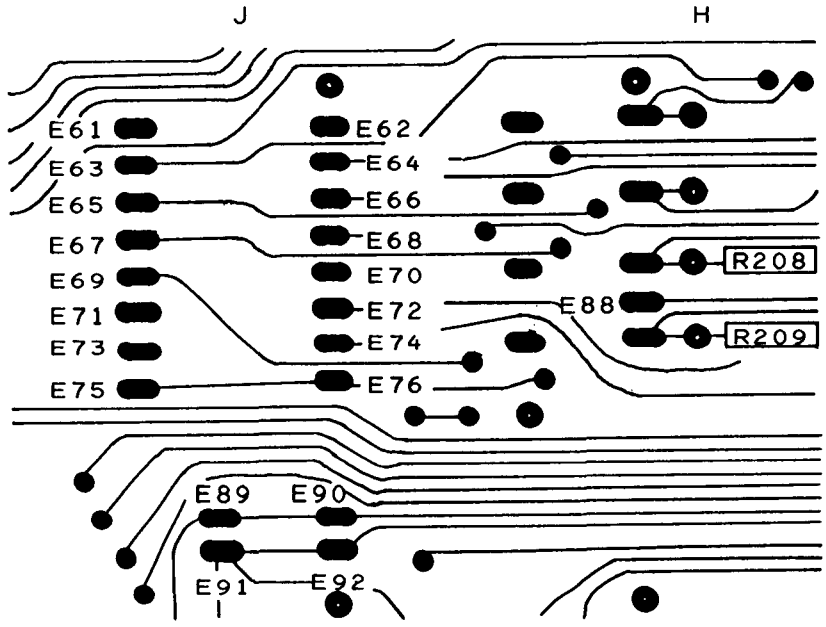


FIGURE 7-2

OVERLAY, CURSOR AND BLINK PATCHING NETWORK

TABLE 7-1 SIGNALS AT 'E' DESIGNATORS USED IN FIGURE 7-2

E DESIGNATOR	SIGNAL
E61	+5 V AFTER R205
E62	+5 V
E63	6DAT06
E64	6DAT05
E65	6DAT07
E66	6DAT04
E67	6DAT08
E68	6DAT03
E69	6DAT09
E70	6DAT02
E71	6DAT10
E72	6DAT01
E73	6DAT11
E74	6DAT00
E75	6ØLAY2 (PROM input)
E76	6ØLAY1 (PROM input)
E89	7CURS41 (PROM input)
E90	7CURS4
E91	7CURS31 (PROM input)
E92	7CURS3
E88	TIME

Described below are some examples of possible patching configurations:

a. Four cursors, no overlays and no blink:
No patching is necessary, the board is fabricated for this configuration. The determination of which cursor is mixed with which data subchannel is controlled by the coding entered into the PROM as is dictated by system requirements.

b. Four cursors, one overlay and no blink:
Cut trace between E75 and E76. Jumper E76 to the data subchannel dictated by the system requirements to be the overlay. Assume data subchannel 07 then E76 would be jumpered to E65.

c. Two cursors, two overlays (subchannels 09 and 10) and one blinking overlay (subchannel 11).

Cut trace between E75 and E76 (6ØLAY1)
Cut trace between E92 and E91 (7CURS31)
Cut trace between E90 and E89 (7CURS41)
Remove resistor R205 E75 (6ØLAY2)

Cursors 7CURS1 and 7CURS2 are provided to all PROMs, therefore, the first requirement is satisfied.

Jumper E76 to E69 (6ØLAY1 = Subchannel 09)
Jumper E75 to E71 (6ØLAY2 = Subchannel 10)
Jumper E91 to E73 (7CURS31 = Subchannel 11)
Jumper E89 to E88 (TIME = Blink OSC Output)

Now that the initial conditions have been met, (two cursors, two overlays and one blinking overlay) the input signals are present at the PROM. The coding of the PROMs is dependent on user requirements. This determines the color of the overlays and cursors and this also determines on which of the outputs the cursors and overlays will appear. The outputs from the PROMs at 5J, 5H, 6F and 6E on sheet 3 and sheet 4 provide the inputs to the DACs (on sheets 5, 6 and 7).

7.1.3 THE INPUTS TO THE DIGITAL TO ANALOG CONVERTERS (SHEETS 5, 6 AND 7)

The outputs from the mixing PROMs are reclocked and provide inputs to the "straight through" video summing junctions. Also, the mixing PROMs provide inputs to the DACs. On sheets 5, 6 and 7 the mixing PROM outputs pass through a patching network. The patching network (at 3K) makes

possible the conversion of the 4 bit DAC to a 3 of 2 bit DAC and this same patching arrangement allows the 8 bit DAC to be converted to a 7, 6 or 5 bit DAC. The patching network at 3K is shown in Figure 7-3. The signals at the network designators are shown in Table 7-2.

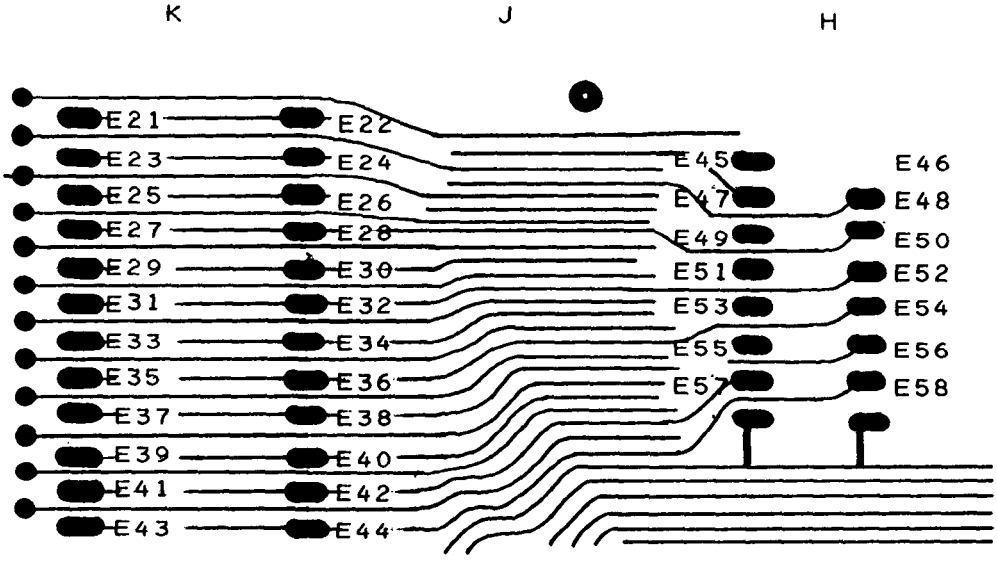


FIGURE 7-3

DAC PATCHING NETWORK

TABLE 7-2 SIGNALS AT E DESIGNATORS USED IN FIGURE 7-3

E DESIGNATOR	SIGNAL
E21	VID10 PROM Output
E23	VID11
E25	VID09
E27	VID08
E29	VID03
E31	VID04
E33	VID06
E35	VID05
E37	VID02
E39	VID07
E41	VID00
E42	VID01 PROM Output
E22, E45	VDAC43 DAC Input
E24, E46	VDAC83
E26, E47	VDAC23
E28, E48	VDAC13
E30, E49	VDAC82
E32, E50	VDAC11
E34, E51	VDAC41
E36, E52	VDAC21
E38, E53	VDAC42
E40, E54	VDAC81
E42, E55	VDAC12
E44, E56	VDAC22 DAC Input
E57	GROUND
E58	GROUND

Using the 4 bit DAC on sheet 6 as an example, and with the specified requirements as follows: four subchannels of memory with subchannel 03 to be overlay, and subchannels 02-00 to be DAC inputs with 02 as the MSB. To accomplish this patching network must be modified as follows:

a. Cut trace between E29 and E30

b. Cut trace between E37 and E38

c. Cut trace between E43 and E44

d. Cut trace between E41 and E42

e. Jumper E37 to E30

f. Jumper E43 to E38

g. Jumper E41 to E44

h. Jumper E55 to E57

7.1.4 CURSOR 10% BRIGHTNESS PATCH

The 10% brightness patching area (4H) provides for distributing the 10% brightness signals 5CURS1, 5CURS2, 5CURS3 and 5CURS4 to the video summing junctions. Shown in Figure 7-4 is the patching network at 4H. The signals at the network designators are shown in Table 7-3.

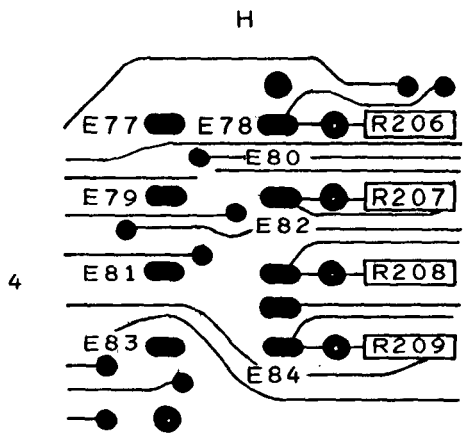


FIGURE 7-4

CURSOR 10% BRIGHTNESS
PATCHING NETWORK

TABLE 7-3 SIGNALS AT DESIGNATORS USED IN FIGURE 7-4

E DESIGNATORS	SIGNAL
E77	5CURS3
E79	5CURS4
E81	5CURS1
E83	5CURS2
E78	+5VDC
E80	+5VDC
E82	+5VDC
E84	+5VDC

7.2 VIDEO TYPE I LOGIC DESCRIPTION

7.2.1 VIDEO TYPE I BLOCK DIAGRAM

Refer to drawing 502445-1. The data inputs from the memory boards are brought into holding registers to resynchronize the data with the system clock on the video board. The data is transferred to the element and field selector, where signals 2ESEL and 6FLDSEL determine which memory board data is transferred to the following reclock register. The data is then mixed with cursors, overlays and blink options. The outputs are reclocked by flip-flops and output to the video amplifier. After being mixed with cursor and overlay outputs, the data is reclocked by high speed flip-flops leading into three 4 bit DACs. Each DAC output is sent to video amplifiers. All 15 video amplifiers also receive as inputs, cursor data (for 10% brightness), composite blanking and composite sync. Composite blanking, composite sync, vertical drive, and horizontal drive signals are provided to external output jacks.

7.2.2 TIMING (SHEET 2)

System clock (NTBQ & PTBQ) and system half clock (NTHQ & PTHQ) are brought into ECL to TTL converters at 8B. A timing diagram showing signal relation-

ships of 3VFUL, 7VFUL and 7VHAF, which are the outputs of a NAND gate at location 4C, is also shown.

7.2.3 SYNC, BLANKING AND DRIVE SIGNALS (SHEET 2)

Signals 9TCBBB (composite blanking), 9THDDD (horizontal drive), 9TCSSS (composite sync) and 9TVDDD (vertical drive) are reclocked on the board to align them with data at locations 5C, 5B, 5A and 3B. Signal DCBBB (3F-8) is mixed with data (at 6A, 6B, 6C on sheet 3 and 6G, 6H, 6J on sheet 4) and cursors (at 3G on sheet 10). Signals 7CBLK (4A-3), 7CBLK1 (4A-11), 7CYSN (3A-6) and 7CSYN1 (3A-8) are composite sync and composite blanking which are mixed with data at the video amplifier summing junction (sheets 8 and 9). NAND gate (4D) outputs 4CBBB, 4HDDD, 4CSSS and 4VDDD supply current to the 2N4258 transistors through current limiting resistors (3D). The MOS driver at 3E is used to convert the transistors current drive to approximately 4 volts P-P. The signals 9VCMPBLE (composite blanking), 9VHORZE (horizontal drive), 9VCMSYE (composite sync) and 9VVERTE (vertical drive) are brought out to external BNC connectors.

7.2.4 GENERATION OF ELEMENT AND FIELD SELECTION SIGNALS (SHEET 2)

Signal 2ESEL (5E-5) is the element select signal and is produced by keying signal 9ESEL (determines system element resolution) and 9MDRB3 (4B-9). Signal 9MDRB3 is a 7 MHz square wave and is generated from 9TCBBB.

Signal 6FLDSEL (5E-2) is the field select signal and is produced by keying signal 9FSEL (determines system field resolution) and 9DRFLD. Signal 9DRFLD is a 30 Hz square wave and is driven to the video board. From the control board.

7.2.5 INPUT RECLOCK REGISTERS (SHEETS 3 AND 4)

Refresh memory data 9MVXXXX enters the video board and is resynchronized with the system clock (7VHAF) at locations 7A, 7B, 7C, 7D, 7F, 7G, 7H and 7J. Each refresh data line is pulled up to +5 volts thru a 1K Ω resistor located at 8A, 8E, 8H and 8J.

7.2.6 ELEMENT AND FIELD SELECTION (SHEETS 3 AND 4)

Signals 2ESEL and 6FLDSEL are generated on sheet 2 and are discussed in paragraph 7.2.4. Element and field selection occur at 6A, 6B, 6C, 6G, 6H, and 6J. The outputs from the selectors are realigned with the system clock (7VFUL) at locations 6D and 5G.

7.2.7 BLANKING MIXED WITH DATA (SHEETS 3 AND 4)

Signal DCBBB (composite blanking) is generated on sheet and is applied to the enable inputs of the element and field selectors at 6A, 6B, 6C, 6G, 6H and 6J. When DCBBB is active, the selector is tri-stated and the outputs are pulled up to +5 volts by 220 Ω resistors.

7.2.8 OVERLAYS, CURSORS AND BLINK MIXED WITH DATA (SHEETS 3 AND 4)

The mixing of overlays, cursors and blink mixed with data is provided by PROMs (at locations 6F, 6E, 5J and 5H) and a patching network shown on sheet 4. Each PROM has three data inputs (6DAT_), two cursor inputs (7CURS1 and 7CURS2) and four inputs from the patching network. The four inputs from the patching network may be patched in any manner desired, refer to Functional Description (paragraph 7.1) for example. Signal TIME (2D-5) is the blank input to the patching network and is shown on sheet 4).

7.2.9 4 BIT DAC OUTPUTS (SHEETS 5, 6 AND 7)

After mixing occurs (in the PROMs) the data goes through a patching network to the inputs of the high speed J-K flip-flops (74S112) resynchronizing the data with the system clock (3VQFUL). The patching and the flip-flops are located at 1D, 1C, 1B, 1A, 1F and 1E. The data from the high speed flip-flop is used to trigger complementary pair transistor switches (four transistor switches per DAC). Each transistor switch consists of biasing resistors and a complementary transistor pair. The current output is summed by a R-2R resistive ladder network which drives

a current amplifier (1H, 1L and 1P). The analog data out of the current amplifiers drives the data input to the video amplifiers.

7.2.10 "STRAIGHT THROUGH" VIDEO OUTPUTS (SHEETS 3 AND 4)

After mixing occurs (in the PROMs) the data is reclocked to resynchronize the data with the system clock (7VFUL) at 6K on sheet 3 and at 5K on sheet 4. The data out of the flip-flops (6K and 5K) drives the data input to the video amplifiers.

7.2.11 THE VIDEO AMPLIFIERS (SHEETS 8 AND 9)

The video amplifiers provide the outputs to drive television monitors. Each video amplifier consists of a summing junction and a current amplifier. The summing junction adds to the data 10% cursor brightness (4CURS1 and 4CURS2), composite blanking (7CBLK) and composite sync (7CSYN). The transistor collector provides current drive to the current amplifier. The current amplifier output is approximately 1 volt P-P into a 75 Ω load.

7.2.12 CURSORS (SHEET 10)

The cursor 9CURS1-9CURS4 are reclocked (to align them with the data, and ANDed with blanking (DCBBB) at locations 5D, 3G, 5E, 3B and 4E. Signals 7CURS1-7CURS4 are to be mixed with data at the PROMs. Signals 4CURS1 and 4CURS2 are the 10% brightness inputs of the video amplifiers.

	ACC	ACW		BCC	BCW
1	ZGND	ZGND	101		
2	ZGND	ZGND	102		
3	ZP05	ZP05	103		
4	ZP05	ZP05	104		
5	ZN12	ZN12	105		
6	ZN12	ZN12	106		
7	ZN05	ZN05	107		
8	ZP12	ZP12	108		
9			109	#DRFLD	9DRFLD
10	PTBQ	NTBQ	110		
11	PTHQ	NTHQ	111	9VCMPBLE	9VCMSYE
12	9MVBOOE	9MVAOOE	112	#VCMPBLE	#VCMSYE
13	9MVBOOØ	9MVAOOØ	113	9VHØRZE	9VVERTE
14	9MVB01E	9MVA01E	114	#VHORZE	#VVERTE
15	#DSRB1	9DSRB1	115	VIDEØ00	VIDEØ08
16	9MVB01Ø	9MVA01Ø	116	ZGND00	ZGND08
17	9MVB02E	9MVA02E	117	VIDEØ01	VIDEØ09
18	9MVB02Ø	9MVA02Ø	118	ZGND01	ZGND09
19	#CURS1	9CURS1	119	VIDEØ02	VIDEØ10
20	9MVB03E	9MVA03E	120	ZGND02	ZGND10
21	9MVB03Ø	9MVA03Ø	121	VIDEØ03	VIDEØ11
22	9MVB04E	9MVA04E	122	ZGND03	ZGND11
23	9ICPCLR	#ICPCLR	123	VIDEØ04	CENTER
24	9MVB04Ø	9MVA04Ø	124	ZGND04	SHLD
25	9MVB05E	9MVA05E	125	VIDEØ05	DVIDEØ1
26	9MVB05Ø	9MVA05Ø	126	ZGND05	ZGNDD1
27	#CURS2	9CURS2	127	VIDEØ06	DVIDEØ2
28	9MVB06E	9MVA06E	128	ZGND06	ZGNDD2
29	9MVB06Ø	9MVA06Ø	129	VIDEØ07	DVIDEØ3
30	9MVB07E	9MVA07E	130	ZGND07	ZGNDD3
31	#CURS3	9CURS3	131	ZGND	ZGND
32	9MVB07Ø	9MVA07Ø	132		
33	9MVB08E	9MVA08E	133		
34	9MVB08Ø	9MVA08Ø	134		
35	#CURS4	9CURS4	135		
36	9MVB09E	9MVA09E	136		
37	9MVB09Ø	9MVA09Ø	137		
38	9MVB10E	9MVA10E	138		
39	9BPWCLR	#BPWCLR	139		
40	9MVB10Ø	9MVA10Ø	140		
41	9MVB11E	9MVA11E	141		
42	9MVB11Ø	9MVA11Ø	142		
43	9TCBBB	9TCSSS	143		
44	#TCBBB	#TCSSS	144		
45	9THDDD	9TVDDD	145		
46	#THDDD	#TVDDD	146		
47	#ESEL	9ESEL	147		
48	#FSEL	9FSEL	148	ZP05	ZP05
49	PTPQØNE	NTPQØNE	149	ZGND	ZGND
50	PTPQTWØ	NTPQTWØ	150	ZGND	ZGND

TABLE 7-4 VIDEO BOARD TYPE I SIGNAL TERM PINOUT

8.0 VIDEO 2A

Video Type 2A Introduction

Refer to the block diagram shown in Figure 8-1. The Video Type 2A board contains one 1024 x 12 bit LookUp Table (LUT) which is addressed by 10 or less memory planes. The Video 2A contains an 8080 interface which allows the 1024 x 12 LUT to be written to, or read from. The output of the LUT is driven through three 4 bit digital to analog convertors (DAC) which produce red, green and blue drives for a color monitor. Thus 4096 colors may be derived from the $(2^4)^3$ possible combinations of three (RGB) color DACs. Also contained on the Type 2A Video is one 8 bit DAC. The 8 bit DAC can be driven either before or after the LUT. This DAC can provide 256 shades of intensity to drive a black and white monitor. Composite blanking, composite sync, vertical drive and horizontal drive are also provided from this board to allow external use of these sync signals. Two cursors and two overlays can be mixed with the video outputs.

Reference drawings Logic Diagrams 503046, Assembly Drawings 503045, Internal Video Cable (9100 and 9200) 502337, Internal Video Cable (9300) 502789.

8.1 FUNCTIONAL DESCRIPTION REFERENCE DRAWING 503046

8.1.1 ELEMENT AND FIELD SELECTION (SHEETS 2 AND 3)

In a high resolution system (640 elements by 512 lines) four memory data lines are necessary for one video data channel. As an example, the memory input data lines for channel 00 are as follows:

9MVA00E - A Raster Field, E = even element
9MVA00Ø - A Raster Field, Ø = odd element
9MVB00E - B Raster Field, E = even element
9MVB00Ø - B Raster Field, Ø = odd element

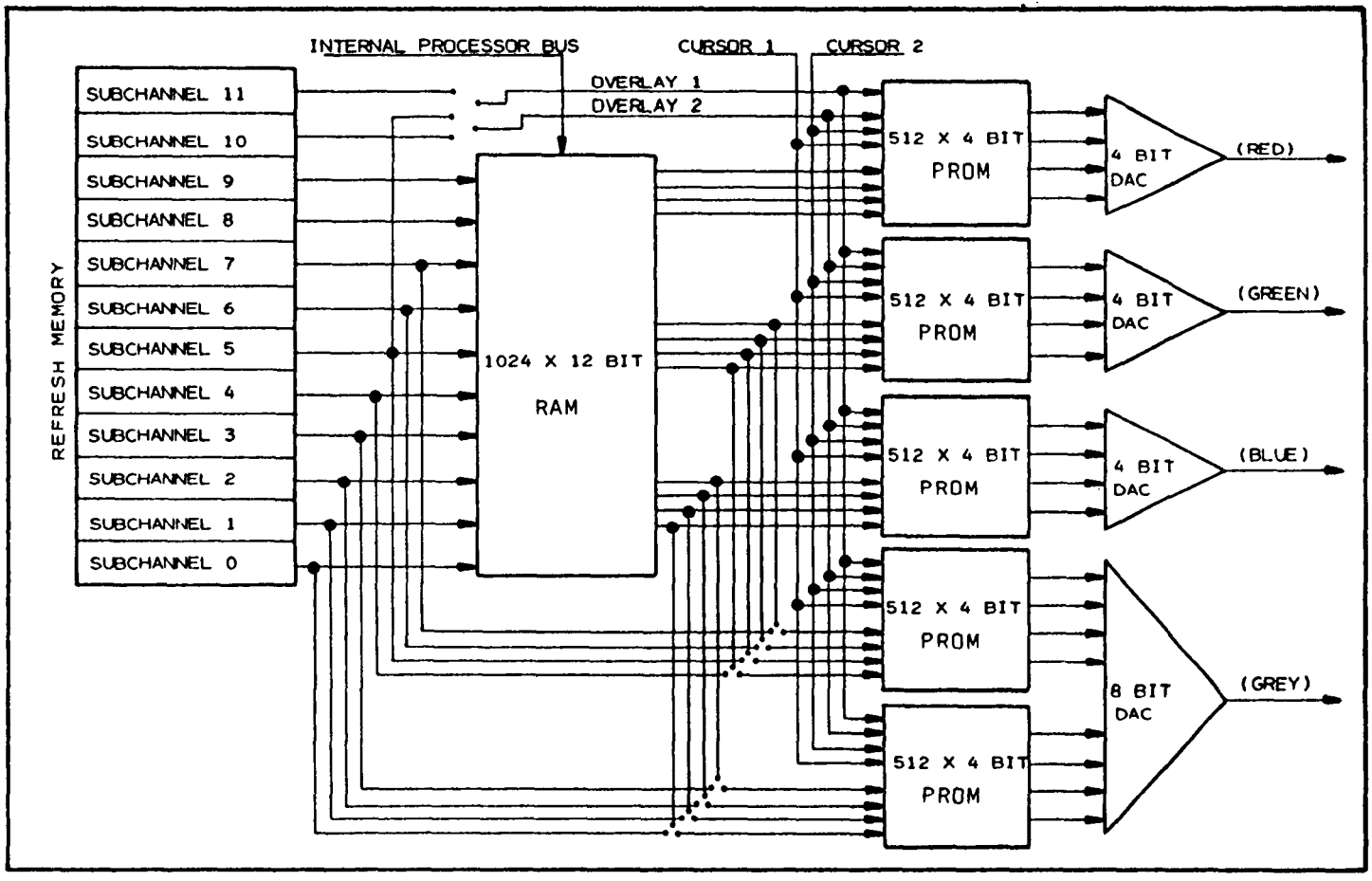


FIGURE 8-1 SIMPLIFIED BLOCK OF VIDEO 2A

In a medium resolution system (640 elements by 256 lines) two memory data lines are necessary for one video output. As an example, the input memory data lines for channel 01 are as follows:

9MVA01E - A Raster Field, E = even element
9MVA01Ø - A Raster Field, Ø = odd element
9MVB01E - Not used in medium (or low) resolution
9MVB01Ø - Not used in medium (or low) resolution

In a low resolution system (320 elements by 256 lines) one memory data line is necessary for one video data channel. As an example, the input memory data lines for channel 02 are as follows:

9MVA02E - A Raster Field, E = even element
9MVA02Ø - Not used in low resolution
9MVB02E - Not used in low (or medium) resolution
9MVB02Ø - Not used in low (or medium) resolution

The three examples above serve to illustrate the function of the element and field selectors (locations 7C, 7D, 7E, 7F, 6D and 6E). The selector control signals are 2ESEL (will select even memory data when equal to 0 volts) and 6FLDSEL (will select 'A' raster field when equal to 0 volts).

The selected memory data becomes synchronized with the system full clock (7VFUL) through reclock registers (6C and 6F, and the data 6DAT00-6DAT11 is applied to the LUT address selectors.

8.1.2 LUT ADDRESS SELECTOR (SHEETS 2 AND 3)

The purpose of the LUT address selector is to select the address counter when data is to be read from or written into the LUT, and to select the memory data at all other times.

If a system has less than 10 memory planes to address the LUT then the LUT may be partitioned. Partitioning means to convert the LUT from 1024 address locations to 512, 256, 128, 64, 32 or 16 address locations. At the same time, the effective number of LUTs increases from one up to a possible 64 LUTs of 16 addresses each.

Shown in Figure 8-2 is the patching network to accomplish this (see sheet 3 of logic). The signals at the network designators are shown in Table 8-1.

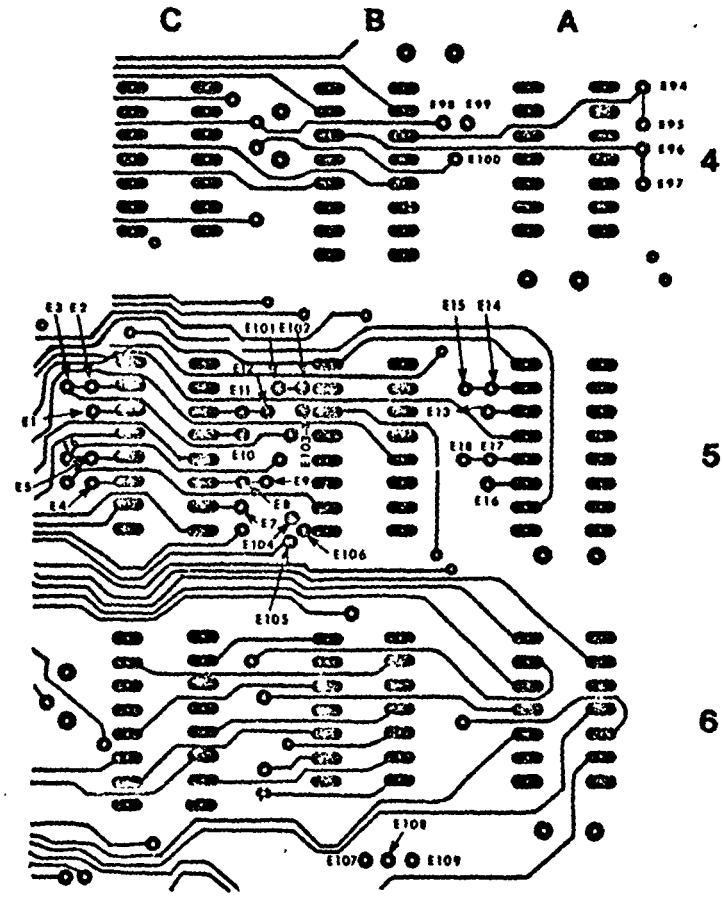


FIGURE 8-2 LUT PARTITIONING PATCHING NETWORK

TABLE 8-1 SIGNALS AT E DESIGNATORS FOR FIGURE 8-2

E DESIGNATORS	SIGNAL
E1	AD04
E2	ADAA
E3	DAT04
E4	AD05
E5	ADAB
E6	DAT05
E7	AD06
E8	ADAC
E9	DAT06
E10	AD07
E11	ADAD
E12	DAT07
E13	AD08
E14	ADAE
E15	DAT08
E16	AD09
E17	ADAF
E18	DAT09
E98	DAT03
E99	ADAG
E100	AD03
E101	DAT00
E102	ADAH
E103	AD00
E104	ADAI
E105	DAT01
E106	AD01
E107	DAT02
E108	ADAJ
E109	AD02

AD are the outputs from the address counter.
DAT are the outputs from the memory.
ADAX are the inputs to the selectors.

As an example assume the system contains eight memory planes. The LUT may be partitioned by making the following changes:

a. Cut trace between E14 and E15
b. Cut trace between E17 and E18
c. Jumper E14 to E13
d. Jumper E16 to E17

Now the LUT is partitioned into four 256 addresses by 12 bit LUTs. Each LUT is selected by loading the address counter bits AD08 and AD09.

8.1.3 SELECT DATA FOR 8 BIT DIGITAL TO ANALOG CONVERTER (SHEET 4)

A patching network is provided for selecting data before or after the LUT to drive an 8 bit DAC. The patching network is shown in Figure 8-3. The signals at the network designators are shown in Table 8-2.

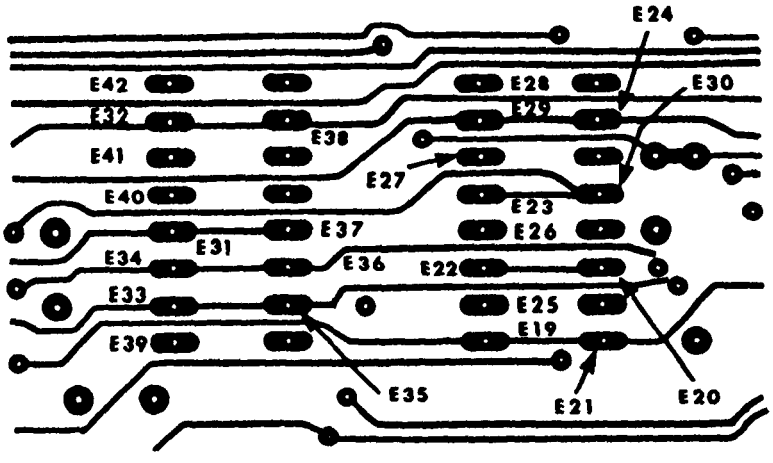


FIGURE 8-3
8 BIT DAC DATA SELECTION PATCHING NETWORK

TABLE 8-2 THE SIGNALS AT E DESIGNATORS FOR FIGURE 8-3

E DESIGNATOR	SIGNAL
E19	DL00
E20	DL01
E21	DAT00
E22	DAT01
E23	DAT02
E24	DAT03
E25	DLUT00
E26	DLUT01
E27	DLUT02
E28	DLUT03
E29	DL03
E30	DL02
E31	DL06
E32	DL07
E33	DL04
E34	DL05
E35	DAT04
E36	DAT05
E37	DAT06
E38	DAT07
E39	DLUT04
E40	DLUT05
E41	DLUT06
E42	DLUT07

NOTE:
DLXX is the data to the 8 Bit DAC

DATXX is the data before the LUT

DLUTXX is the data after the LUT

The patching network, as shown in Figure 8-3, is configured to select data before the LUT to drive the 8 bit DAC. To have the data after the LUT drive the 8 bit DAC the patching network must be modified as follows:

a. Cut trace between E24 and E29
b. Cut trace between E23 and E30
c. Cut trace between E22 and E20
d. Cut trace between E21 and E19
e. Cut trace between E38 and E32
f. Cut trace between E37 and E31
g. Cut trace between E36 and E34
h. Cut trace between E35 and E33
i. Jumper E32 to E42
j. Jumper E31 to E41
k. Jumper E34 to E40
l. Jumper E33 to E39
m. Jumper E29 to E28
n. Jumper E30 to E27
o. Jumper E20 to E26
p. Jumper E19 to E25

8.1.4 MIXING CURSORS AND OVERLAYS WITH DATA (SHEET 4)

The cursors and overlays are mixed with data in PROMs at locations 4H, 4G (grey scale), 4N, 4M and 4L (color). The coding of the PROMs will provide the following:

a. Location (where the cursor or overlay will appear on the color or black and white monitor).
b. The color or the shade of the cursor or overlay.

Overlays are normally provided by memories 10 (DØLAY1) and 11 (DØLAY2). If any of the memory subchannels zero through nine are not being used to address the LUT, then any two of these memories may replace memories 10 and 11. The network patching necessary is shown in Figure 8-2.

The signals at the network designators are shown in Table 8-3 and are on Sheet 3 of logic.

TABLE 8-3 OVERLAY PATCHING FOR SIGNALS AT E DESIGNATORS FOR FIGURE 8-2

E DESIGNATOR	SIGNAL
E94	1DOLAY2
E96	1DOLAY1
E101	DAT00
E105	DAT01
E107	DAT02
E98	DAT03
E3	DAT04
E6	DAT05
E9	DAT06
E12	DAT07
E15	DAT08
E18	DAT09
E97	DAT10
E95	DAT11

As an example the LUT is partitioned into two sections (8.1.2) this leaves DAT09 not used. Memory subchannel 09 is specified to be overlay 1. To make the change: Cut trace between E96 and E97 and add a jumper from E96 to E18.

8.1.5 INPUTS TO THE DIGITAL TO ANALOG CONVERTERS (SHEETS 5, 6, 7, 8 AND 9)

The mixing PROM outputs pass through a patching network. The patching networks (at 3C, 3N and 3J) make possible the conversion of the 4 bit DACs and the 8 bit DAC to smaller DACs. The patching networks are shown in Figure 8-5, and E designator signals are listed in Table 8-4.

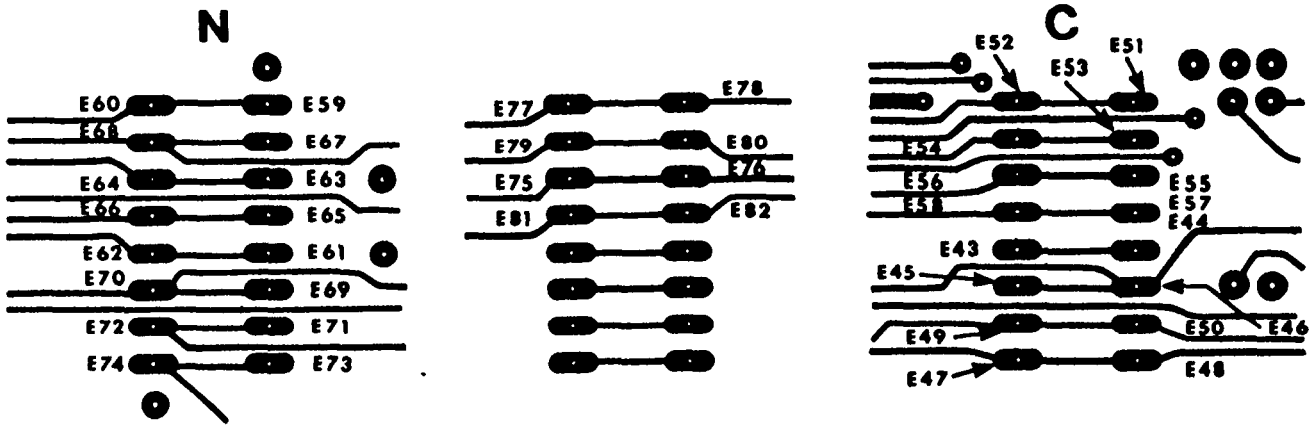


FIGURE 8-5 DAC CONVERSION PATCHING NETWORK

As an example, the conversion of an 8 bit DAC to a 6 bit DAC is shown as follows (see sheets 8 and 9):

a. Cut trace between E51 and E52
b. Cut trace between E55 and E56
c. Cut trace between E57 and E58
d. Cut trace between E53 and E54
e. Cut trace between E45 and E46
f. Cut trace between E43 and E44
g. Cut trace between E47 and E48
h. Cut trace between E49 and E50
i. Jumper E55 to E52
j. Jumper E43 to E56
k. Jumper E45 to E58
l. Jumper E47 to E44
m. Jumper E49 to E46
n. Jumper E57 to E54
o. Jumper E48 to GND
p. Jumper E50 to GND

When conversion is complete, the 6 bit DAC has DVID05 as the MSB and DVID00 as the LSB.

TABLE 8-4 SIGNALS AT E DESIGNATORS FOR FIGURE 8-5

3C-THE 8 BIT DAC

E DESIGNATOR	SIGNAL
E43	DVID03
E44	VDAC85
E45	DVID02
E46	VDAC45
E47	DVID01
E48	VDAC25
E49	DVID00
E50	VDAC15
E51	DVID07
E52	VDAC84
E53	DVID06
E54	VDAC44
E55	DVID05
E56	VDAC24
E57	DVID04
E58	VDAC14

3N-THE 'BLUE' 4 BIT DAC

E DESIGNATOR	SIGNAL
E59	VID03
E60	VDAC81
E61	VID02
E62	VDAC41
E63	VID01
E64	VDAC21
E65	VID00
E66	VDAC11

3J-THE 'RED' 4 BIT DAC

E DESIGNATOR	SIGNAL
E75	VID11
E76	VDAC83
E77	VID10
E78	VDAC43
E79	VID09
E80	VDAC23
E81	VID08
E82	VDAC13

3N-THE 'GREEN' 4 BIT DAC

E DESIGNATOR	SIGNAL
E67	VID07
E68	VDAC82
E69	VID06
E70	VDAC42
E71	VID05
E72	VDAC22
E73	VID04
E74	VDAC12

8.1.6 ADDRESS AND DATA BUS PATCHING (SHEET 11)

An RM-9000 System may contain up to four video cards. In order to allow four video 2A cards to operate independently in one system a patching network (7P on sheet 11) is used to change the LUT address. Table 8-5 briefly describes the address and data bus changes that are accomplished by the patching network. The network is shown in Figure 8-6 and the related signals in Table 8-6.

TABLE 8-5 THE LUT ADDRESS

LUT #	JUMPER CONFIGURATION	LUT ADDRESS	LUT DATA	S/D ADDRESS	DATA BIT
1	7P 16-12, 2-3, 5-6	8060-8061 (16)	8062-8063 (16)	Address Source 800A Address Dest. 800C	10 16
2	7P 16-11, 1-3, 5-6	8064-8065 (16)	8066-8067 (16)		20 16
3	7P 16-10, 2-3, 4-6	8068-8069 (16)	806A-806B (16)		40 16
4	7P 16-9, 1-3, 4-6	806C-806D (16)	806E-806F (16)		80 16

FIGURE 8-6 LUT ADDRESS PATCHING NETWORK

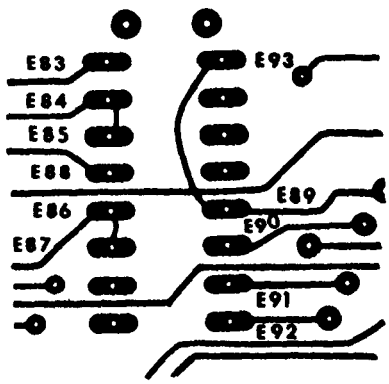


TABLE 8-6 SIGNALS AT E DESIGNATORS FOR FIGURE 8-6

E DESIGNATORS	SIGNAL
E83 (7P-1)	3BAD02
E84 (7P-2)	BAD02
E85 (7P-3)	BADS02
E88 (7P-4)	3BAD03
E86 (7P-5)	9BAD03
E87 (7P-6)	9BADS03
E92 (7P-9)	BDAT07
E91 (7P-10)	BDAT06
E90 (7P-11)	BDAT05
E89 (7P-12)	BDAT04
E93 (7P-16)	DSØD

The standard video card patching is:

Jumper 7P-16 to 7P-12

Jumper 7P-2 to 7P-3

Jumper 7P-5 to 7P-6

Modification for LUT 2, 3, or 4 is shown in Table 8-4.

8.1.7 INSTRUCTION FORMATS

The instruction formats are used by software to load the Address Counter, to read or write the LUT with data, and to set the source-destination flag registers. As a reference, turn to the table on sheet 12 of the schematic.

8.1.7.1 LUT ADDRESS COUNTER (SHEET 11)

Address Bus 4BADDR00-4BADDR15

1	0	0	0	0	0	0	0	0	1	1	0	X	X	X	Y
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 0 identifies the data (9BDAT00-9BDATA11) on the IP bus as a low byte when Bit 0 = 0 and a high byte Bit 0 = 1.

The decode for address counter of LUT #1

0	0	0	Y
3	2	1	0

8060-8061

The decode for address counter of LUT #2

0	1	0	Y
3	2	1	0

8064-8065

The decode for address counter of LUT #3

1	0	0	Y
3	2	1	0

8068-8069

The decode for address counter of LUT #4

1	1	0	Y
3	2	1	0

806C-806D

Data bus 9BDATA00-9BDATA11

X	X	-	-	-	-	-	-	-	-	-	-
11	10	9	8	7	6	5	4	3	2	1	0

X = Don't cares

The data (bits 0-9) is parallel loaded into the address counter by the address bus (bits 0-15) being decoded producing signal 7WLTAH (sheet 11).

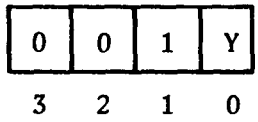
8.1.7.2 READ/WRITE THE LUT (PIØ)

Address Bus 4BADDR00-4BADDR15

1	0	0	0	0	0	0	0	0	1	1	0	X	X	X	Y
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

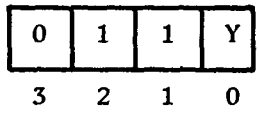
Bit 0 identifies the data as a low byte (bit 0 = 0) or as a high byte (Bit 0 = 1).

The decode for the address of LUT #1



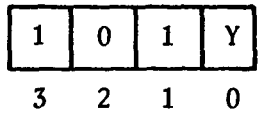
8062-8063

The decode for the address of LUT #2



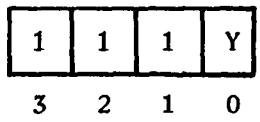
8066-8067

The decode for the address of LUT #3



806A-806B

The decode for the address of LUT #4



806E-806F

On the data bus (9BDATA00-9BDATA15), all the data bits are used in reading or writing the LUT.

8.1.7.3 DMA SOURCE REGISTER

The DMA source register contains four source flags, one each, for each of the four possible LUTs. The byte addresses for the DMA source register are 800A and 800B. Refer also to section 6.2.6 of this manual for a further description of this register. If one of the source flags is set for one of the LUTs it will be driving data onto the IP data bus during a DMA transfer.

8.1.7.4 DMA Destination Register

The DMA destination register contains four destination flags, one each, for each of the four possible LUTs. The byte addresses for the DMA destination register are 800C and 800D. Refer also to section 6.2.7 of this manual for a further description of this register. If one of the destination flags is set for one of the LUTs, it will be receiving data from the IP data bus during a DMA transfer.

8.2 VIDEO 2A LOGIC DESCRIPTION

Sheets 2 thru 13 (Drawing 503046).

8.2.1 INPUT RECLOCK REGISTERS

Refresh memory data 9MV enters the video board and is resynchronized with the system clock (7VHAF) at locations 8C, 9C, 8D, 9D, 8E, 9E, 8F and 9F (shown on sheets 2 and 3). Each refresh data line is pulled up to +5 V thru a 1K Ω resistor, location 9B, 8B, 8G and 9G (shown on Sheets 2 and 3).

8.2.2 ELEMENT AND FIELD SELECTION

Element (2ESEL) and field (6FLDSEL) select signals control which refresh memory is selected by the selectors at locations 7C, 7D, 6D, 7E, 6E and 7F (shown on sheets 2 and 3). Signal 2ESEL is developed by logic 6R, 4S and 4J (shown on sheet 2). Signal 6FLDSEL is developed by logic 6R, 4S, 7S and 7H (shown on sheet 2). Signals 9ESEL and 9FSEL are levels which determine the resolution of the system. The element and field selector outputs are resynchronized with the system clock (7VFUL) at locations 6C and 6F (shown on sheets 2 and 3).

8.2.3 LUT ADDRESS SELECTOR

The source of the LUT address is determined by the selectors at locations 5A, 5B and 5C (shown on sheets 2 and 3).

The jumper locations shown at the input to the address selector enables the LUT to be partitioned into more than one LUT. If less than 10 refresh memories are in the system, the address selector inputs that do not have memory data present, are tied such that the selector output is controlled only by the address register. See paragraph 8.1.2 for a detailed description of the jumper network.

Signal SEL1 is normally 0 volts which selects memory data lines to address the LUT. When the data stored in the LUT is to be modified or readback,

SEL1 will equal +5 volts selecting the address register to address the LUT. (See paragraph 8.2.14 for the development of SEL1).

8.2.4 LOOKUP TABLE

The LUT consists of 12 RAMs at locations 5E, 5F, 5G, 5H, 5J, 5K, 5L, 5M, 5N, 5P, 5R and 5S (sheet 4). Signal 6WRLUT when at 0 volts enables (BDAT_) data to be written to the LUT.

8.2.5 CURSORS

Two cursor signals (9CURS1 and 9CURS2) are reclocked to sychronize them with the data and blanking before being mixed data at locations 8H, 6H, 4B and 6G (sheet 4).

8.2.6 SELECT DATA BEFORE OR AFTER LUT

The 8 bit DAC may be patched to display data before or after LUT at locations 4D and 4E (sheet 4). For a detailed description of the patching network see paragraph 8.1.3.

8.2.7 DATA MIXED WITH OVERLAYS, CURSORS AND BLANKING

Mixing of overlays, cursors and blanking occurs at locations 4H, 4G, 4N, 4M and 4L (sheet 4).

8.2.8 COLOR OUTPUTS

After mixing occurs, the data is input through a patching network (paragraph 8.1.5) to the inputs of high speed J-K flip-flops, resynchronizing data with the system clock (3VQFUL). The patching and the flip-flops are located at 3N, 2P, 2R (sheet 5) 2M, 2L, 3N (on sheet 6) and 3J, 2J, 2G (on sheet 7). The complement data from the high speed flip-flops is used to trigger transistor switches (four transistor switches per DAC). The current output is summed by a R-2R resistive ladder network which drives

a current amplifier. The analog data out of the current amplifier drives the data input of the video amplifiers. The DAC data output is summed with cursors, blanking and sync data. The data is D-C coupled to the current amplifier and driven out to the back of the system chassis. The three DACs and video amplifiers are located on sheets 5, 6 and 7.

8.2.9 GREY SCALE OUTPUT

The patching and reclocking described in paragraph 8.1.5 is located at 3C, 3F, 3E, 3B and 3A (sheets 8 and 9). The DAC and video amplifier shown on sheets 8 and 9 are functionally identical to the DACs and video amplifiers discussed in paragraph 8.2.8. Additional variable resistors are for fine adjustment of the grey scale output, see 8.3.1 for tuning procedure.

8.2.10 SYSTEM TIMING

The system timing is received as an ECL signal by an ECL-TTL translator at location 9A (sheet 10). Signals 7VFUL, 7VHAF, 3VQFUL1 and 3VQFUL have high drive capability and are used throughout the board to reclock data, blanking and cursors. The J-K flip-flop 4J, NAND gate 7K and invertor 6J in zone A5 develop 7GCLK at either a half clock or full clock rate. The K input is controlled by 9DSRB1 which originates on the control board. With the K input of 4J low 7GCLK is at a full clock rate. When 9DSRB1 becomes active low 7GCLK will operate at a half clock rate. The half clock being applied to 8H will move the blanking signals to screen refresh element one time, therefore making the video out scroll one element to the left.

8.2.11 SYNC BLANKING AND DRIVE SIGNALS

Signals 9TCBBB (composite blnaking), 9THDDD (horizontal drive), 9TCSSS (composite sync), and 9TVDDD (vertical drive) are reclocked on the board to align them with the data at locations 9H, 8H, 7H and 7J (sheet 10). Signal 5CBLK is generated by 4CBLK (composite blanking) ORed with SEL1 (which is active when LUT is being written into or read from) and is used to blank the data. 5CBLK is generated at location 7M (sheet 10). 7CBLK2

generated at 7K (sheet 10) and 7CSYN1 are generated at 8J (sheet 10). These two signals are input to the video amplifier summing network.

Signals 9VCMPBLE, 9VHORZE, 9VCMSYE and 9VVERTE are the external drive sync and blanking signals provided to the connector J-2. These signals are generated by the logic located at 9L, 9K and 9J (sheet 10).

8.2.12 DATA RECEIVER/DRIVERS, ADDRESS COUNTER, ADDRESS RECEIVERS AND INSTRUCTION DECODERS

Data from the Internal Processor Bus (IPB) is shown (sheet 11) being applied to the tristate drivers/receivers are always held in the receive condition. To drive data back onto the IPB pin 15 must become high. The condition to activate this control is reading the LUT (RLDTH, RLTDL or RLTDMA). The terms BDAT00 through BDAT09 can be seen being applied directly to the address counter inputs and the data inputs of the LUT RAMs.

The address counter locations 8M, 8N and 8P (sheet 11) can be parallel-loaded by signal 7WLTAH or can be incremented to the next high address by signal 5INC. Signal 5INC is generated on sheet 12.

The address or instruction decoder 8R and 7R (sheet 11) decodes the bus address provided by the logic at locations 9R and 9S (sheet 11).

The signals decoded are 8LTAH (load address counter), 8LTDL (LUT data low byte), 8LTDH (LUT data high byte), 8DMAS (load DMA source register), and 8DMAD (load DMA destination register).

The patching network at location 7P (sheet 11) is detailed on sheet 12 and explained in the functional description paragraph 8.1.6.

8.2.13 SYSTEM RESET

Signal 9BPWCLR is the system reset signal and is clocked onto the video board by J-K flip-flop at 7G (sheet 12). The output from 7G-9 (5POR)

resets the source flag register, destination flag register (both at 6S) and the DMA: read control flip-flop at 7G (sheet 13).

8.2.14 GATING THE ADDRESS COUNTER TO THE LUT

Signal SEL1 controls the selection of address lines to the LUT, and is generated at 6L-3 (sheet 12). Signals 7DMSD (source at 8L-8) or 7LTD (source at 8S-11) will activate SEL1 prior to a LUT data transfer.

Signal 7DMSD is the response to the source or destination flags being set (7SØD generated at 8S-3 on sheet 12). Signal 7LTD generated at 8S-11 is the result of 8LTDL or 8LTDH being active. 8LTDL and 8LTDH are address bus decoder outputs (8R-14 and 8R-15 on sheet 11).

8.2.15 DMA CONTROL

Signal 9BDMAGØ is active during any DMA transfer. The video board will only respond to 9BDMAGØ being active if the source (7DMAS) or the destination (7DMAD) has been set. 9BDMAGØ is reclocked at location 9H sheet 13. The reclocked signal (8DMA) is ANDed with 7SOD, 7DMAD and 7DMAS at location 8L.

8.2.16 READ LUT CONTROL (PIØ)

Signal 9BREAD allows data stored in the LUT to read back to the 8080 on the control board. Signals RLTDL (read LUT low byte) and RLTDH (read LUT high byte) are generated at 6M (sheet 12) by ANDing 9BREAD with address bus decoder outputs 8LTDL and 8LTDH. RLTDL and RLTDH activate the data bus drivers (sheet 11). 9BREAD is reclocked and trailing edge detected by 8K, 4K and 7L (sheet 12). Signal 6TREAD at 7L-3 is the trailing edge detection of 9BREAD, which is ANDed with 8LTDH (LUT data high byte) producing INC2 at 7M-4 (sheet 12). INC2 produces 5INC (at 6M-13) which increments the address counter.

8.2.17 WRITE LUT PIO

Signal 9BWRITE allows data to be stored in the LUT under control of the 8080 on the Control board. Signal 9LEDWR is the leading edge detection of 9BWRITE and is generated at 6L-6 (sheet 12). Signal 7WLUTH (6P-11) is the result of ANDing 9LEDWR with 8LTDH (LUT data high byte decoder output) signal 7WLUTH (PIO WRITE) is ORed with 5WLUT (DMA WRITE) to produce the LUT write pulse 6WRLUT at 6N-8. After the LUT data has been written at the proper address, the address counter is incremented by 5INC 6M-13). Signal 5INC is activated by signal 7WLUTH at 6N-11 (sheet 12).

8.2.18 LOAD THE SOURCE OF DESTINATION FLAGS

Signal 9BWRITE under 8080 control (on the control board) is leading edge detected (9LEDWR at 6L-6) and ANDed with decoder outputs 8DMAS or 8DMAD to produce flag register clocks 9DMAS (6P-6) or 9DMAD (6P-8). Signal 9DMAS clocks the source flag register at 6S-11. Signal 9DMAD clocks the destination flag register at 6S-3 (sheet 12).

8.2.19 PARALLEL LOAD OF ADDRESS COUNTER

Signal 9BWRITE (under 8080 control on the control board) is leading edge detected (9LEDWR at 6L-6) and ANDed with decoder output 8LTAH to produce the parallel load enable signal 7WLTAH at 6P-3 (sheet 12).

8.2.20 DMA TRANSFER OF DATA TO THE LUT

A DMA transfer to the LUT is initialized by setting the destination flag register, parallel loading the address counter to the initial LUT address and signal 9BDMAGØ becoming active. Signal 7DMAD (source 6S-6 on sheet 12) is active as a result of setting the destination flag. This causes SLE1 to become active selecting the address counter outputs to address the LUT upon 9BDMAGØ becoming active. Signal 9BWRITE under control of the card with its source flag register set, is reclocked at 8K. Signal 7WRITE (8K-12) is ANDed with 7DDMA (8L-3) to produce 7WLTDMA (7L-6 on sheet 13). Signal 5WLUT (6L-8) is the leading edge detection of 7WLTDMA produced by ANDing

at 6L WLP and 9ACK, signal 5WLUT (DMA write pulse) generates 6WRLUT at 6N-8 on sheet 12.

Signal 9ACK generated by 7WLTDMA is gated thru 7L by 7DDMA to produce 9BACK (6K-6 on sheet 13) which is the acknowledge to the board that activated 9BWRITE. Signal 7INC1 (8L-11) is the result of the trailing edge detection of 7WRITE and developes 5INC (6M-13) which increments the address counter to the next LUT address. The transfers will proceed until 9DMAGØ becomes inactive.

8.2.21 DMA TRANSFER OF DATA FROM THE LUT

A DMA transfer from the LUT is initialized by setting the source flag register, parallel loading the address counter to the initial LUT address and signal 9BDMAGØ becoming active. Signal 7DMSD (source 8L-6 on sheet 13) is active as a result of setting the source flag. This causes SEL1 to become active, selecting the address counter outputs to address the LUT upon 9BDMAGO becoming active. Signal 9BDMAGØ active produces 8DMA at 9H-2. Signal 8DMA is ANDed with 7DMAS (the source flag) generating 7SDMA (at 8L-6 on sheet 13). Signal 7SDMA is ANDed with two other control signals 9ACK to produce 9SVWRITE (at 7L-8) and 1BACK to produce DREAD (at 7M-10). Signal DREAD active generates RLTDMA and 8RLTDMA on the system clock edge (at 7G-5 and 6 on sheet 13).

Signal RLTDMA gates data from the LUT on to the data bus and also is reclocked to produce 9ACK at 8K-5 on sheet 13. Signal 9ACK active causes 9SVWRITE to be active which drives the system write line 9BWRITE at location 6K-8. When 9BACK (the system acknowledge line) is received it is reclocked into the board to produce 9BACK1 at 9H-2 on sheet 12. Signal 9BACK1 is ANDed with 8RLTDMA (7G-6) to produce 8RDINC (7B-3) to increment the address counter. Also, signal 9BACK1 at 6J-1 causes DREAD to become inactive (7M-10). Signals RLTDMA and 8RLTDMA (7G-5 and 6) will go inactive on the next system half clock edge. As a result of RLTDMA going inactive 9ACK (8K-5) will go inactive gating 9BWRITE (6K-8) off. The next LUT data transfer will occur once the board acknowledging releases the 9BACK line.

	ACC	ACW		BCC	BCW
1	ZGND	ZGND	101	#BREAD	9BREAD
2	ZGND	ZGND	102	#BWRITE	9BWRITE
3	ZP05	ZP05	103	#BACK	9BACK
4	ZP05	ZP05	104	#BREADY	4BREADY
5	ZN12	ZN12	105	#BDMAHR	9BDMAHR
6	ZN12	ZN12	106	#BDMAHA	9BDMAHA
7	ZN05	ZN05	107	#BDMAGO	9BDMAGØ
8	ZP12	ZP12	108	#BSYNC	9BSYNC
9			109	#DRFLD	9DRFLD
10	PTBQ	NTBQ	110		
11	PTHQ	NTHQ	111	9VCMPBLE	9VCMSYE
12	9MVB00E	9MVA00E	112	#VCMPBLE	#VCMSYE
13	9MVB00Ø	9MVA00Ø	113	9VHØRZE	9VVERTE
14	9MVB01E	9MVA01E	114	#VHØRZE	#VVERTE
15	#DSRB1	9DSRB1	115		
16	9MVB01Ø	9MVA01Ø	116		
17	9MVB02E	9MVA02E	117		
18	9MVB02Ø	9MVA02Ø	118		
19	#CURS1	9CURS1	119		
20	9MVB03E	9MVA03E	120		
21	9MVB03Ø	9MVA03Ø	121		
22	9MVB04E	9MVA04E	122		
23	9ICPCLR		123		DVIDEØ1
24	9MVB04Ø	9MVA04Ø	124		ZGNDD1
25	9MVB05E	9MVA05E	125		DVIDEØ2
26	9MVB05Ø	9MVA05Ø	126		ZGNDD2
27	#CURS2	9CURS2	127		DVIDEØ3
28	9MVB06E	9MVA06E	128		ZGNDD3
29	9MVB06Ø	9MVA06Ø	129		DVIDEØ4
30	9MVB07E	9MVA07E	130		ZGNDD4
31	#CURS3	9CURS3	131	ZGND	ZGND
32	9MVB07Ø	9MVA07Ø	132	9BDATA01	9BDATA00
33	9MVB08E	9MVA08E	133	9BDATA03	9BDATA02
34	9MVB08Ø	9MVA08Ø	134	9BDATA05	9BDATA04
35	#CURS4	9CURS4	135	9BDATA07	9BDATA06
36	9MVB09E	9MVA09E	136	9BDATA09	9BDATA08
37	9MVB09Ø	9MVA09Ø	137	9BDATA11	9BDATA10
38	9MVB10E	9MVA10E	138	9BDATA13	9BDATA12
39	9BPWCLR	#BPWCLR	139	9BDATA15	9BDATA14
40	9MVB10Ø	9MVA10Ø	140	4BADDR01	4BADDR00
41	9MVB11E	9MVA11E	141	4BADDR03	4BADDR02
42	9MVB11Ø	9MVA11Ø	142	4BADDR05	4BADDR04
43	9TCBBB	9TCSSS	143	4BADDR07	4BADDR06
44	#TCBBB	#TCSSS	144	4BADDR09	4BADDR08
45	9THDDD	9TVDDD	145	4BADDR11	4BADDR10
46	#THDDD	#TVDDD	146	4BADDR13	4BADDR12
47	#ESEL	9ESEL	147	4BADDR15	4BADDR14
48	#FSEL	9FSEL	148	ZP05	ZP05
49	PTPQØNE	NTPQØNE	149	ZGND	ZGND
50	TPQTWØ	NTPQTWØ	150	ZGND	ZGND

TABLE 8-7 VIDEO BOARD TYPE 2A SIGNAL TERM PINOUT

8.3 TUNING THE VIDEO 2A 8 BIT DIGITAL TO ANALOG CONVERTER (DAC) A REV. AND ABOVE.

Equipment Needed:

- 9100 System
- MM80
- QQA Monitors
- Tektronix 7844 scope with differential comparator 7A13 and dual trace amplifier 7A26
- 9000 Extender Card
- Video II Logic Dwg. #503046
- Video II Assembly Dwg. #503045
- Video II Mixing PROM's #502767
- Memory Expansion Card (MOD) with Video II Test PROM's

Procedure:

(1) Execute the MOC routine at address 1400 (Set Parameters and Erase)

(2) Trigger scope on signal 9VHORZE at BCC 113 (probe #1 is connected to 7A26)

(3) Connect another probe (probe #2 is connected to 7A13) to the emitter of Q13 at R109.

(4) Adjust the scope to display the one element wide pulse that is at the beginning of every horizontal line. The pulse should look on the scope like Figure 8-7.

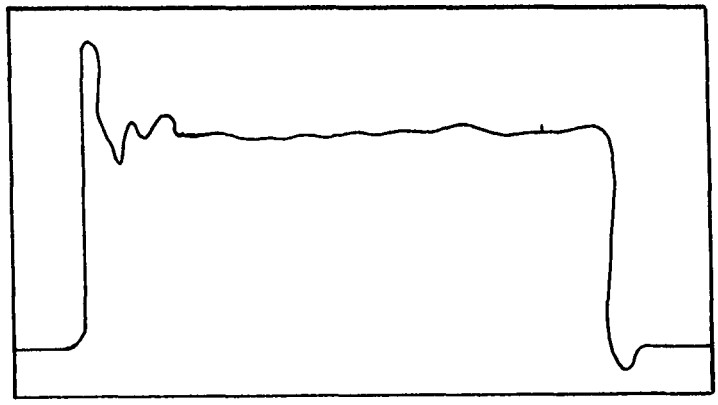


FIGURE 8-7 ELEMENT PULSE

(5) Adjust resistor R110 (the amplitude control pot) for maximum pulse amplitude, then back the amplitude down approximately ½ turn.

(6) Repeat step (5) for:
adjusting R115 move probe #2 to the emitter of Q14,
adjusting R120 move probe #2 to the emitter of Q15,
adjusting R125 move probe #2 to the emitter of Q16,
adjusting R145 move probe #2 to the emitter of Q17,
adjusting R150 move probe #2 to the emitter of Q18,
adjusting R155 move probe #2 to the emitter of Q19, and
adjusting R160 move probe #2 to the emitter of Q20.

(7) Move probe #2 to the emitter of Q13 and adjust pot R174 for minimum ringing on the rising pulse edge.

(8) Adjust R179 and R192 for minimum ringing on the rising pulse edge. Repeat step (7) and step (8) one more time.

(9) Move probe #2 to the emitter of Q14 and adjust R179 for minimum ringing.

(10) Adjust R192 for minimum ringing. Repeat steps (7), (8), (9), (10) one more time.

(11) Move probe #2 to the emitter of Q15 and adjust R192 for minimum ringing.

(12) Move probe #2 to the emitter of Q16 and adjust R197 for minimum ringing.

(13) Move probe #2 to the emitter of Q17 and adjust R202 for minimum ringing.

(14) Move probe #2 to the emitter of Q18 and adjust R207 for minimum ringing.

(15) Move probe #2 to the emitter of Q19 and adjust for minimum ringing.

(16) Move probe #2 to the emitter of Q20 and adjust R217 for minimum ringing.

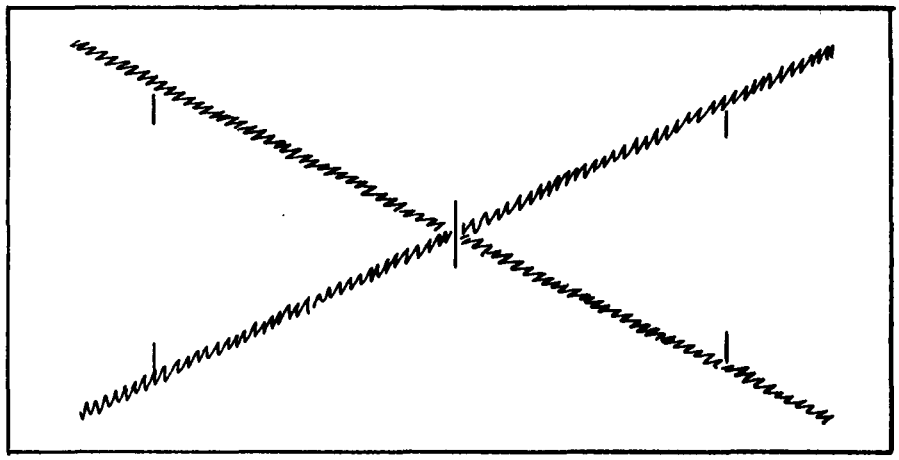
(17) Use the amplitude of the pulse at the emitter of Q13 as a guide. Adjust the pulse amplitudes at the emitters of the other DAC transistors to exactly match Q13's amplitude.

(18) Recheck the ringing at all DAC transistor emitters and adjust the appropriate pot.

Q13 adjust R174	Q17 adjust R202
Q14 adjust R179	Q18 adjust R207
Q15 adjust R192	Q19 adjust R212
Q16 adjust R197	Q20 adjust R217

(19) Move probe #2 to 1E-4 and adjust the scope to display the up and down ramp. Thie display should look like Figure 8-8.

FIGURE 8-8 VIDEO UP AND DOWN RAMPS



(20) First adjust the amplitude control pots by very small turns so that the ramp is linear.

(21) Looking at the scope adjust R197 by a very small turn look for a change in the "glitch" amplitude. If there is no change in the glitch amplitude return the pot R197 to its original position. If there is a change in the glitch amplitude adjust R197 for minimum glitch amplitude.

(22) Repeat step (21) with R192, R179, and R174 adjusting the pots in this order.

(23) Take the extender card out of the system and restart the test started in step (1).

(24) Using a coaxial cable that is plugged into the back of the system, look at the video output on the scope. Be sure that the scope side of the coaxial cable is terminated in 75 Ω resistor.

(25) The video signal must meet the following specifications to be considered good:

Signal Amplitude (overall)	=	1 volt P.P. ±0.15 volt
Video Data Amplitude	=	0.75 volt P.P. ±0.1 volt
Sync Amplitude	=	greater than 0.2 volts
Signal Rise/Fall Time	=	less than 35 nsec
Video Data Noise	=	less than 40 mvolts (glitch amplitude)

9.0 256 x 320 MEMORY BOARD

The 256 x 320 memory board is used in the 9000 Series System to provide modular refresh memory (RAM) as dictated by user requirements. A memory board may contain from one to six sections, each section containing sufficient dynamic RAM to supply 1 bit per element for 256 raster scan lines of 320 elements per line. The addressing scheme is implemented in such a fashion that each section may be used as a 256 x 320 building block to construct refresh memories that are optimized in size for user requirements. There are three basic system x, y resolution displays supported by the 256 x 320 memory board which are described below:

a. 2 x 3 resolution display (9100 System)
The 9100 System may contain up to 12 sections of memory (two full boards). Each section, then, represents one plane of refresh memory and each element may be described by 1 to 12 bits of memory.

b. 2 x 6 resolution display (9200 System)
The 9200 System may contain up to 24 sections of memory (four full boards). To implement a single plane of refresh memory requires two sections of memory. A system requiring one plane of refresh memory will contain one memory board with two sections of memory. One memory board can supply memory for one to three planes of refresh memory. Thus each element may be described by 1 to 12 bits of memory.

c. 5 x 6 resolution display (9300 System)
The 9300 System may contain up to 48 sections of memory (eight full boards). To implement a single plane of refresh memory requires four sections of memory. A system requiring one plane of refresh memory will contain two memory boards, each with two sections of memory. One board contains that portion of the memory used for field A while the other contains the memory for field B. Two memory boards can supply memory from one to three

refresh memory planes. Thus each element may be described by 1 to 12 bits of memory.

All of the necessary configuration control required to allow the 256 x 320 memory board to operate in any of the three systems is supplied by the system's backplane. However, when a board does not contain a full compliment of memory sections, the shift register for the missing section must also be left out and a jumper installed in its place (from pin 13 to pin 6 of the IC position).

9.1 256 x 320 MEMORY BOARD BLOCK DIAGRAM. REFER TO DRAWING 502331

Each 2 x 3 memory section is implemented by utilizing twenty 4K dynamic RAM chips. In order to supply data to the video board at the necessary data rate the memory must operate to allow two memory accesses, one to read out data for screen refresh and the other to allow random reads or writes of data from the Control Board. This requires that the 20 RAMs be arranged into two banks of 10 RAMs each, in which all ten chips are accessed simultaneously when reading video data and loaded into a 10 bit shift register. Then the banks (bank 0 and bank 1) are accessed alternately to load the 10 bit shift register. The alternate bank accessing allows the minimum hardware implementation and power consumption. The data loaded into the shift register is serially shifted to form a video data line. When data is read or written from the control board only the affected RAM is accessed and in the case of readback the proper output is selected by a one-of-ten multiplexer named the data bit multiplexer. A single 256 x 320 section contains 81,920 bits of memory and each 4K RAM contains 4096 bits. Therefore, the following three cases apply:

a. In a 2 x 3 resolution display, one RAM will supply each 20th bit, in the same places, along all the horizontal lines on the display monitor. Thus, inoperative RAM will give the appearance, (dependent on the failure mode), of vertical white/black lines spaced every 20 elements across a screen that has been erased to black/white.

b. In a 2 x 6 resolution display, it is the same as in a. above, except that wherever the number 20 appears substitute the number 40.

c. In a 5 x 6 resolution display, one RAM IC will supply each 40th bit, in the same places, along all even or odd lines on the display monitor, depending upon which field it is in. Thus, an inoperative RAM will give the appearance, dependent upon the failure mode, of dashed vertical white/black lines spaced every 40 elements across a screen that has been erased to black/white.

As previously mentioned in Section 6, a thorough understanding of the operation of the 4K, 16 pin dynamic RAM is necessary to understand the addressing scheme used to address refresh memory. Such understanding is assumed in the following discussion. Recalling the XCOP register and the YCOP register of paragraph 6.14, if these two registers are placed end to end as shown in Figure 9-1 and considered as the total refresh memory address, there are enough bits of address to address four sections of memory, arranged to provide a single plane of memory in a 5 x 6 resolution system.

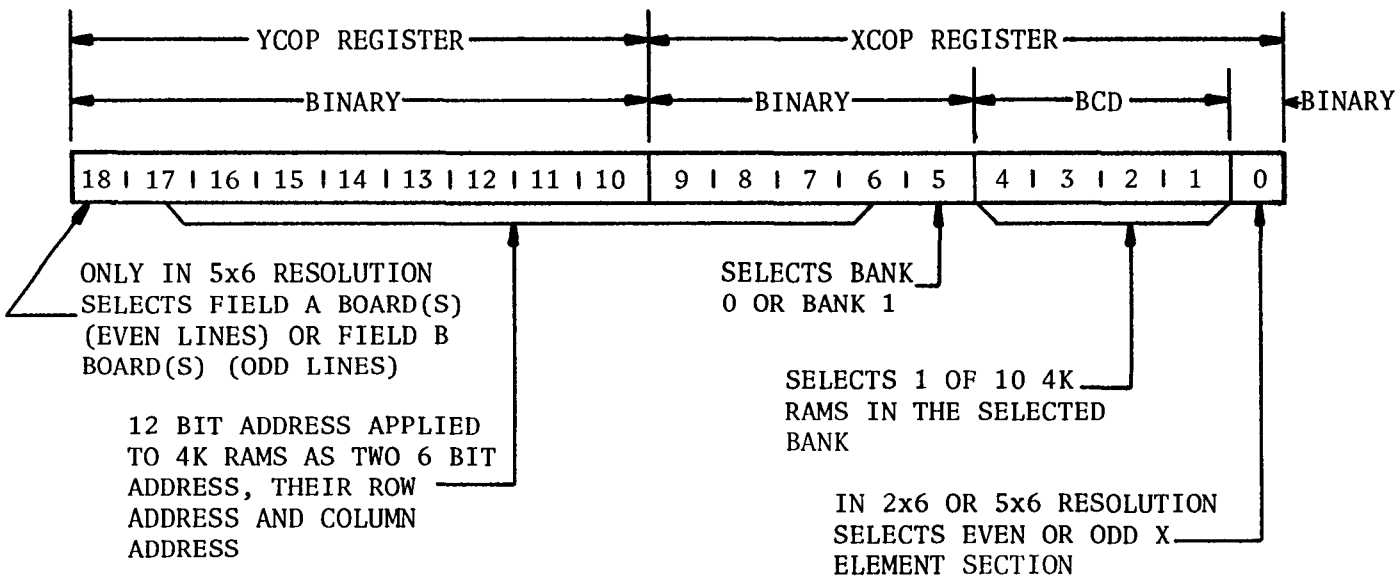


FIGURE 9-1 REFRESH MEMORY ADDRESS

As shown in Figure 9-1 the refresh memory addressing is provided by the COP registers for reading and writing refresh memory. The refresh memory addressing is exactly the same format for reading refresh memory for screen refreshing. Bit 0 is used only in systems that have a resolution of 640 elements in the X direction. In such systems this bit provides selection between the memory sections, on the same memory board, which contains the even numbered X elements and the odd numbered X elements. For screen refresh addressing this bit is sent to the video board(s) to control the multiplexing of the serial video data driven from the memory boards. For read/write addressing the memory plane select multiplexer two receives this bit. It is used as part of the condition to enable the write pulse to the memory section containing the addressed bit of refresh memory; the data out select multiplexer also receives this bit for selecting data output from the memory section containing the addressed bit of refresh memory.

Bits 1 thru 4 are a BCD value from 0 to 9. For screen refresh addressing to 10 RAMs of bank 0 or bank 1 are simultaneously accessed and the data is loaded into the 10 bit video data shift register while the BCD value is used as part of the addressing to the refresh memory timing generator, discussed in paragraph 6.13 to create the 10 state refresh memory access cycles. For read/write addressing, these bits (the BCD value) are driven to the memory select multiplexer 2 which will output them to the memory chip select decoder. This decoder decodes them, in conjunction with bit 5 of the address format above and selects one of the 20 RAMs, to which the $\overline{\text{RAS}}$ signal (5RCE00-5RCE19) will be gated, as the chip containing the addressed bit of refresh memory. When reading data back to the control board bits 1-4 also control the Data Bit Multiplexers.

Bit 5 is the bank select signal. For screen refresh addressing it selects the current bank of 10 RAMs to be simultaneously accessed. For read/write addressing it is used as described for bits 1 through 4 above.

Bits 6 through 17 are the 12 bit address that is applied to each RAM. This 12 bit address is multiplexed into two 6 bit addresses, on the control board (9DAD00-9DAD05). The first of the 6 bit addresses, the row address,

is loaded into the RAMs with the leading edge of $\overline{RAS}$. The second of the 6 bit addresses, the column address, is loaded into the RAMs with the leading edge of $\overline{CAS}$ (CAS*1).

Bit 18 is the field select line, used only in 5 x 6 resolution systems. For screen refresh addressing it is driven to both the memory and the video boards (9DRFLD). On the memory boards it is used to cause the board that is not currently supplying video data to enter a low power consumption dynamic RAM refresh mode. On the video boards it is used to multiplex the video data from the memory boards. For read/write addressing it selects the memory board that contains the addressed bit of refresh memory by forming another part of the gating conditions to enable the write pulse to the memory section containing the addressed bit of refresh memory.

The discussion of refresh memory addressing up to this point has been limited to addressing within a single plane of refresh memory. Any of the 9000 Systems may contain from one to 12 planes of memory which may be selected individually or in any multiple combination desired. This is accomplished with the 12 bit memory plane select register on the control board. The memory plane select signals (9DWE0-9DWE11) are driven to the appropriate memory boards, as determined by the backplane wiring, where they are received by the memory plane select multiplexer and used as the third (and final) condition to enable the write pulse to the memory section containing the addressed bit of refresh memory. Thus it may also be said that the memory plane select signal has no effect in regard to screen refresh.

From the discussion so far, it should be apparent that a memory board must be keyed to its logical position in refresh memory. This information is supplied via the backplane wiring of four keying signals (4KLLR, 4KRWFLD, 4KRWBI and 4KRFLD), three addressing signals (9DRLFD, 9DRWBI, and 9DRWFLD) and which memory plane select and data signals are connected to it.

The block labeled Memory Refresh Select Code Counter performs two functions by monitoring two screen refresh addressing signals received from the Con-

trol Board (9DSRA8 and 9DSRA1). This counter provides the load pulse (5SRLDP) to all memory section 10 Bit Video Data Shift Registers. In addition it provides a 4 bit select code used in gating the $\overline{RAS}$ (5RCE00-5RCE19) signals to the RAMs for dynamic RAM refresh in the 5 x 6 resolution systems.

The six MEMORY DATA DRVRS-RCVRS are bi-directional data drivers and receivers. In 2 x 3 resolution systems 6 of the 12 bits of data are wired to one board. In 2 x 6 or 5 x 6 resolution systems only 3 bits of the 12 bits of data are wired to any one board. The direction in which these drivers and receivers are transferring data is controlled by two signals (9DRWFLD and 9DREAD) from the Control Board.

There are six possible RAM data input signals (DIP0-DIP5) from the memory sections. Depending upon the system resolution (the keying signal 4KLLR and the address signal 9DRWBI) the Data Out Select Multiplexer connects the proper RAM data output signals to the Memory Data DRVRS--RCVRS.

9.2 MEMORY CHIP SELECT LOGIC

Refer to Sheet 2 which shows the logic that implements the memory chip select capability in three difference modes as follows:

a. One-of-twenty mode, which allows read or write of refresh memory from the Control Board.

b. Ten-of-twenty mode, which simultaneously will read data for screen refresh.

c. A sequence of 3-3-4-3-3-4 mode, which is used only in a 5 x 6 resolution system to implement dynamic RAM refresh.

The selection circuit is implemented with three 32 word x 8 bi-polar PROMs (locations 8D, 8F and 8N), programmed to drive their outputs in the selectable modes described above in response to their address inputs driven by the signals MAG1, MA8, MA4, MA2 and MA1. The decoded outputs of these PROMs enable the $\overline{RAS}$ signals (5RCE00 through 5RCE19) to the selected memory chips. The signals MA1 through MA8 are driven from the quad 1-of-2

selector (location 8K) which selects the read/write address inputs (9DRWA1 through 9DRWA8) or the dynamic RAM refresh address inputs (MRA1 through MRA4), when enabled, to supply the PROM address. When disabled, the 1-of-2 selectors outputs are forced to the logic "1" state which is the PROM address for a screen refresh data read. The signal MAG1 is driven by the quad, 1-of-2 selector (location 8J) which selects the read/write address input (9DRWAG1) and $\overline{RAS}$ pulse (5FRWS*1) or the screen refresh data read address input (9DSRAG1) and $\overline{RAS}$ pulse (9DSRRAS). The 5 bit addresses of the PROM are utilized as follows, in hexidecimal:

a. -00, 01, 02, 04, 06, 08, 09, 0A, 0C and 0E for bank 0 read/write 1 of 10 selection.
b. -03, 07 and 0B for bank 0 dynamic RAM refresh groups of 3, 3 and 4 selection.
c. -0F for bank 0 screen refresh all ten selected.
d. -10, 11, 12, 14, 16, 18, 19, 1A, 1C and 1E for bank 1 read/write 1 of 10 selection.
e. -13, 17, and 1B for bank 1 dynamic RAM refresh groups of 3, 3 and 4 selection.
f. -1F for bank 1 screen refresh all ten selected.

The hex D flip-flop (location 9D) is used as a reclock circuit for the memory timing pulses from the Control Board, which receives timing pulses in their correct timing relationship before they are used on this board. Refer to the timing diagram of sheet 10. Also shown on this sheet are the system full and half clocks represented by the signals 7BQ and 7HQ respectively.

The signal 9DADSL is used to select either a screen refresh address (if = "0") a read/write address (if = "1") by controlling the selector of location 8J and in conjunction with the signal RFLD the selector of location 8K. For 2 x 3 or 2 x 6 resolution systems RFLD is forced to a "1". For a 5 x 6 resolution system RFLD is a "1" when its memory board is supplying video data.

9.3 MEMORY DATA AND WRITE ENABLE SELECT LOGIC

Refer to sheet 3 which shows the bidirectional data (9BMD0-9BMD5) drivers and receivers (locations 10N and 10P) which normally are in a state to receive data from the Control Board. The exception occurs when the signal 9DREAD becomes active and the signal RWFLD is a "1"; then they switch to the state of driving data to the Control Board. If the memory board is in a 2 x 3 resolution system the backplane connects six data lines to 9BMD0-9BMD5. If the memory board is in a 2 x 6 or 5 x 6 resolution system the backplane connects only three data lines to 9BMD0-9DMD2. The system backplane also controls the state of a keying signal (4KLLR) which is a "0" for a 2 x 3 resolution system and a "1" for a 2 x 6 or 5 x 6 resolution system. This keying signal is used as the select line to two quad, 1-of-2 selectors (locations 9P and 9R) which control the routing of input data (DIP0-DIP5) to the six memory sections. The signal 9DRWB1 is "0" in a 2 x 3 resolution system, forced by the system backplane connection to ground. In a 2 x 6 or 5 x 6 resolution it is driven by the Control Board and will be a "0" if an even element is to be read or a "1" if an odd element is to be read. There are provisions to accept six memory plane select signals (9DWE0-9DWE5), all of which are connected in a 2 x 3 resolution system while only three (9DWE0-9DWE2) are connected in a 2 x 6 or 5 x 6 resolution system. The appropriate connections are made by the system backplane. The memory plane select signals are received and routed to the write enable gates (locations 9H, 9J and 10J) by two quad, 1-of-2 selectors whose select inputs are also controlled by the keying signal (4KLLR) described above.

The write enable gates just mentioned, gate the write enable pulse (WRTP*1) to a set of 8T26s (locations 3J and 6J), used as drivers to gain a high drive capability. These in turn drive the six write enable signals (5WRTN0-5WRTN5) to the six memory sections. The backplane wiring causes the following signals to be a logic "1"; CEE0, CEE1, CEE2, CEE3, CEE4, CEE5.

The following equations represent the enable conditions for the write enable gates:

WRTN0 = WRPT*1 · EENB · WE0
WRTN1 = WRTP*1 · ØENB · WE1
WRTN2 = WRTP*1 · EENB · WE2
WRTN3 = WRTP*1 · ØENB · WE3
WRTN4 = WRTP*1 · EENB · WE4
WRTN5 = WRTP*1 · ØENB · WE5

where EENB = ØENB = "1" for a 2 x 3 resolution system and,

WRTN0 = WRTP*1 · EENB · WE0
WRTN1 = WRTP*1 · ØENB · WE0
WRTN2 = WRTP*1 · EENB · WE1
WRTN3 = WRTP*1 · ØENB · WE1
WRTN4 = WRTP*1 · EENB · WE2
WRTN5 = WRTP*1 · ØENB · WE2

where EENB = RWFLD · $\overline{\text{DRWB1}}$ and ØENB = RWFLD · DRWB1 for 2 x 6 and 5 x 6 resolution systems.

The control necessary for a memory board to operate properly with respect to the resolution of the system in which it resides, is implemented through controlling the states of seven signals via the system backplane connections. The Table 9-1 below summarizes these connections.

TABLE 9-1 BACKPLANE CONTROL SIGNAL WIRING

SIGNAL	SYSTEM RESOLUTION			
	2x3	2x6	5x6 field A	5x6 field B
4KLLR	G	0	0	0
4KRWFLD	G	G	0	G
4KRWB1	G	0	0	0
4KRFLD	G	G	0	G
9DRFLD	G	G	CC	CC
9DRWB1	G	CC	CC	CC
9DRWFLD	G	G	CC	CC

where G = connected to gr-und
0 = open circuit
CC = driven from the Control Board

9.4 MEMORY SECTIONS

Refer to sheets 4 through 9 of the logic schematics which each contain one memory section, its associated 10 bit video data shift register and 1-of-10 readback data selector. The following discussion pertains to memory section 0, sheet 4, but is directly applicable to sheets 5 through 9 which will not be discussed. The 20 4K dynamic RAMs that comprise memory section 0 are shown surrounded by their peripheral circuitry. The $\overline{\text{CAS}}$ pulse (CAS*1) drives an 8T26 (location 5G) which in turn drives all 20 RAMs of memory section 0. The 6 bit multiplexed address from the Control Board (9DAD01-9DAD05) is received by a pair of 8T26s (locations 5G and 6G) which in turn drive all 20 RAMs of both memory sections 0 and 1 (a total of 40 devices). The 8T26s are used to obtain the high drive capability necessary to drive the capacitive load of a large number of dynamic RAMs. The 10 bit video data shift register is implemented using an 8 bit shift register (location 7B) which is loaded by the load signal (5SRLDP) derived on sheet 2 with an AND gate (location 10E), and part of a quad, 1-of-2 selector (location 7P).

The load signal 5SRLDP is state nine of the 10 state memory timing cycle as shown in the following equation:

$$5SRLDP = SRA1 \cdot SRA8$$

The 4K dynamic RAMs latch their data outputs. To save logic, bits 8 and 9 are multiplexed into the serial input of the 8 bit shift register by the signal SRA1. The corresponding RAMs in bank 0 and bank 1, such as 000 and 010, have their data outputs common, producing the 10 data out signals (D0P0-D9P0) which drive the inputs of the 10 bit shift register and the 1-of-10 data readback selector. The data readback selector is implemented with a 1-of-8 selector (location 7A) and one part of a quad, 1-of-2 selector (location 7N). The video data is shifted out at the system half clock rate through an 8T26 onto the system backplane and over to the video board(s).

	ACC	ACW		BCC	BCW
1	ZGND	ZGND	101	ZGND	ZGND
2	ZGND	ZGND	102	ZGND	ZGND
3	ZP05	ZP05	103	ZP12	ZP12
4	ZP05	ZP05	104	ZP12	ZP12
5	ZN12	ZN12	105	4KLLR	4KRFLD
6	ZN12	ZN12	106		4KRWB1
7	ZN05	ZN05	107		4KRWFLD
8	ZP12	ZP12	108	#DREAD	9DREAD
9	ZP12	ZP12	109	#DRFLD	9DRFLD
10	PTBQ	NTBQ	110	#DRWB1	9DRWB1
11	PTHQ	NTHQ	111	#DRWFLD	9DRWFLD
12	9MSDP0	9DSRA8	112	ZGND	ZGND
13	9MSDP1	#DSRA8	113	9DRWA4	9DRWA8
14	9MSDP2	9DADSL	114	9DRWA2	9DRWA1
15	ZGND	ZGND	115	#DRWAG1	9DRWAG1
16	9MSDP3	9DSRA1	116	#DSRRAS	9DSRRAS
17	9MSDP4	#DSRA1	117	#DSRAG1	9DSRAG1
18	9MSDP5		118		9DWE03
19	ZP12	ZP12	119		9DWE02
20	9MSDP0		120		9DWE05
21	9MSDP1		121	ZP12	ZP12
22	9MSDP2		122		9DWE01
23	ZGND	ZGND	123		9DWE04
24	9MSDP3	9DRWRAS	124		9DWE00
25	9MSDP4		125		9BMD0
26	9MSDP5		126		9BMD1
27	ZGND	ZGND	127		9BMD5
28	9MSDP0	9DCAS	128		9BMD3
29	9MSDP1		129	ZGND	ZGND
30	9MSDP2		130		9BMD4
31	ZGND	ZGND	131		9BMD2
32	9MSDP3	9DWRTP	132		
33	9MSDP4		133		9DCE0
34	9MSDP5		134	CEE1	CE0
35	ZP12	ZP12	135		9DCE1
36	9MSDP0		136	CEE0	CE1
37	9MSDP1		137		9DCE2
38	9MSDP2		138	ZP12	ZP12
39	ZGND	ZGND	139	CEE4	CE2
40	9MSDP3		140	CEE2	CE3
41	9MSDP4		141		9DCE3
42	9MSDP5	9DAD00	142	CEE3	CE4
43	ZGND	ZGND	143		9DCE4
44	9DAD02	9DAD01	144	CEE5	CE5
45	9DAD04	9DAD03	145		9DCE5
46		9DAD05	146		
47	ZP12	ZP12	147	ZN05	ZN05
48	ZP12	ZP12	148	ZP05	ZP05
49	ZGND	ZGND	149	ZGND	ZGND
50	ZGND	ZGND	150	ZGND	ZGND

TABLE 9-2 256 x 320 MEMORY BOARD SIGNAL TERM PINOUT

10.0 MEMORY EXPANSION CARD -- OPTIONAL

Sheet 1 of the logic schematics (drawing 502442) represents a detailed block diagram of the Memory Expansion Card. The IP bus, address and data lines, are buffered to form new 16 bit address and 16 bit data busses internal to the memory expansion card. As shown in the block diagram, the address bus is not only buffered but latched by address latches. The new address bus represented by the terms BADDR0-BADBR15 is utilized to drive the PROM memory sockets as well as the dynamic RAMs. Twenty-eight ROM sockets are available for use with any of six possible manufacturers of PROM memory. The PROM memory output data lines represented by the terms *RØM0-*RØM7 are driven by non-inverting buffer drivers to the high byte portion of the 16 bit data bus represented by the term 9BDAT8-9BDAT15. The memory expansion card contains 8194 bytes of dynamic RAM memory. When loading this RAM memory under DMA control, the 16 bit data bus is loaded as a parallel 16 bit word into the dynamic RAM. As a result, the RAM maybe thought of as a 4096 word by 16 bit memory consisting of a high byte and low byte per word. When operating in a programmed I/O mode the dynamic RAM memory may be thought of as 8194 sequential bytes represented by sequential addresses, or the DMA mode, it may be thought of as two 4096 byte sections where the lower section represents the lower portion of memory and the upper section represents the upper portion of memory. In the DMA mode, 8 bits, the low byte of a 16 bit word, is written into the lower portion of memory and the high byte, 8 bits, are written into the upper 4096 byte section of memory. As a result if a 16 bit word has been transferred into location zero of memory under DMA mode, when it is read under programmed I/O mode, the lower byte of the 16 bit word can be read from location zero, and the upper byte of the 16 bit word can be read from the first location of the upper 4096 bytes. When writing into the dynamic RAM memory under DMA control, the address bit zero, BADDR0, is ignored. When reading or writing under programmed I/O operation, address bit zero represents even and odd byte boundaries.

As shown in the block diagram, the RAM refresh counter is a 6 bit counter represented by the terms RCNT0-RCNT5. It is selected by the RAM Refresh/ 8080 Cycle Address Select Logic and routed to the Row/Column Select Logic

for the purpose of creating a 6 bit address that is used to refresh the 4096 bit dynamic RAM modules. When performing an 8080 or DMA cycle with the dynamic RAM, the Refresh/8080 Cycle Address Select Logic is used to select the 16 bit address from the IP bus. The Row Column Select Logic selects 6 bits at a time from the resulting 12 bit address and routes it to the dynamic RAM. As shown on the block diagram the dynamic RAM is divided into two sections, the high byte section and the low byte section. When the 8080 microprocessor reads or writes the dynamic RAM under programmed I/O mode, sequential 8 bit bytes represented by sequential addresses, are alternated between the low and high byte sections of the RAM. When DMA read or write operations are performed with the dynamic RAM, the 16 bit words are written into the low and high byte sections of the RAM simultaneously. Thus, during DMA operations, address bit zero, BADDR0, is ignored. During programmed I/O operations, address bit zero is utilized to select sequential bytes out of the low and high byte portions of the RAM. Physically, the low and high byte portions of the RAM are represented by two rows of 8 each RAM chips on the board. When performing an 8080 or a DMA cycle with the RAM memory, the output data bits are loaded into the RAM output data register which in turn drives the RAM output data high/low bytes select logic in order to drive the data to the internal processor bus. When doing DMA operations 16 bits of data is driven in parallel to the internal processor data bus. When performing programmed I/O operations, sequential 8 bit bytes are selected from the 16 bit output data register and routed to the most significant 8 bits of the internal processor data bus. When writing into the RAM under DMA mode the data bus is received by 8T26 receivers as represented by the terms BDAT0-BDAT15 and driven to the two sections of RAM memory. In programmed I/O mode the sections of RAM are sequentially chip selected during the writing process. When performing DMA write operations the 16 bit words are written simultaneously into the two byte sections of RAM. The received 16 bit data bus is also utilized for loading the DMA address counter. The output of the DMA address counter is driven through a non-inverting tri-state driver onto the Internal Processor Address Bus represented by the terms 4BADDR0-4BADDR15.

As shown in the block diagram, the RAM and DMA control logic receives the IP handshake signals and converts them into the necessary gating and clocking terms. The term 9BREAD represents an 8080 programmed I/O read operation. The term 9VWRITE represents a DMA or 8080 programmed I/O write operation. The term 9BSYNC represents the 8080 SYNC signal, which is used to load the address into the address latches and start an 8080 cycle with the dynamic RAM. The 9BDMAGO signal represents the enabling of a DMA block transfer. The terms 9BPWCLR and 9ICPCLR represent reset operations. The term 9BACK represents the acknowledge line which is used during DMA transfers to the interface. The term 9BREADY represents the 8080 ready signal which is used to synchronize the 8080 to the memory speeds of the PROM and the RAM. The resulting control signals generated by the RAM control logic provide WE which is a write enable signal to the RAM, RAS which represents the row address strob, CAS which represents the column address strob, and ROW/COL which represent the row/column select lines to the RAMs. The RAM control logic also determines if the cycle is a refresh or an 8080 cycle. The refresh logic assures that at least one column is refreshed each 30 microseconds. The control logic contains special networks which assure that priorities are established between 8080 cycle request and RAM refresh cycle requests.

10.1 CONTROL LOGIC -- SHEET 2

The ROM on the 9000 control board is represented by the hex addresses as 0000-07FF. The memory expansion card ROM continues at hex address 0800 to 3BFF. The standard static RAM contained on the 9000 control card begins at hex address 6000 to 61FF. The dynamic RAM contained on the memory expansion card begins at address 4000 and continues to address 5FFF. The 8080 microprocessor initiates a cycle with the RAM memory by supplying the sync signal, 9BSYNC, shown on Sheet 2. Along with the most significant 4 bits of the address which represent a 4 or 5. The resulting equation represented by the term SNRDY sets the not ready flip-flop, CY8080, and setting the RAG flip-flop (location 7L-10) which drives the memory cycle sequencer. The timing for a typical 8080 cycle with RAM is represented by the timing diagram on Sheet 6. Note that if a refresh cycle is presently in process when an 8080 cycle is initiated the not

ready flip-flop NRDY is set and when the refresh cycle is completed as represented by time state T4T4 the combination of T4T4 and NRDY form the signal 5T4NRDY which begins the 8080 cycle. Thus, a refresh cycle can hold off an 8080 cycle and vice versa, a 8080 cycle can hold off a refresh cycle until it completes. A refresh cycle is initiated when the 30 microsecond one-shot represented by the term 5ONST becomes high. The one-shot output is synchronized to the phase 1 clock by the flip-flop at location 6G represented by the term 5ONST*1. This flip-flop requests a refresh cycle as long as an 8080 cycle is not presently in process. The refresh cycle is initiated by setting the flip-flop RFSH (location 6G-6) which in turn sets the RAG flip-flop (location 7L-10). Approximately every 30 μsec a column in the RAM is refreshed and the refresh counter is incremented. The refresh counter shown on Sheet 3 is incremented if a refresh cycle is being performed and state T3T3 is active. As shown on Sheet 2, the not ready flip-flop is reset during state 2 of the memory cycle sequence. As shown on the timing diagram on Sheet 6, the 8080 actually enters the wait state for two full cycles during any 8080 programmed I/O operation with the RAM. Note also in the timing diagram that the row address strobe and column address strobe are set at particular edges of the Phase 1 and Phase 2 clocks.

The column address strobe, CAS, is represented by the timing state T3T3 which is triggered on the trailing edge of a Phase 2 clock. The row address strobe, RAS, is also triggered on the trailing edge of a Phase 2 clock represented by the signal 7 PQTW2. When performing a programmed I/O write operation to the RAM, the write enable signal is generated during the T3T3 time period. Sheet 2 also shows the DMA control logic which is used to write or read 16 bit words with the RAM memory. A DMA cycle may be initiated by first setting the source and destination flip-flops, shown on Sheet 2, under programmed I/O control. If the RAM memory is to be considered a source, it will fetch data and supply it to the internal processor bus. If the Memory Expansion RAM is to be considered a destination, it will wait for the necessary write control line before strobing the data from the internal processor bus into the RAM. After the source and destination flip-flops are appropriately set, the DMA operation begins when the DMA go signal is received from the IP bus.

The DMA go signal, 9BDMAGO, is combined with the term DMASOD which represents one of two cycle initiating conditions. The first condition is if the memory expansion card is the source for data during a DMA transfer. It is represented by the term DMAS*ACK which means that the source flip-flop is set and the acknowledge line (9BACK) is not active. The second condition is that the memory expansion card is being utilized as a destination for data and it is represented by the destination flip-flop (EDMAD) and the fact that the write handshake line (BWRITE) is active. These two conditions initiate a RAM cycle by driving the leading edge pulse detecting circuit shown on Sheet 2. The resulting leading edge of DMA read or write represented by the term 5LERWD is utilized to set the RAG flip-flop (7L-10). As shown on Sheet 2, a DMA read or write cycle may be delayed if a refresh cycle is in process. The term *DMAINH represents the inhibiting condition of refresh in process.

10.2 ADDRESS LOGIC -- SHEET 3

The IP address bus is latched by the address latches shown on Sheet 3 (locations 6S, 6R). These latches are clocked on the leading edge of Phase 1 during sync for a programmed I/O operation or upon the leading edge of read or write during a DMA operation. The latched addresses are represented by the terms BADDR00-BADDR15. BADDR00, the least significant address bit is utilized to drive the chip select logic shown on Sheet 2 in selecting the two sections of RAM. The address bits 1-12 as represented by the terms BADDR01-BADDR12 are utilized to drive the RAMs last cycle address select logic which selects either an 8080 address or the refresh counter. During a refresh cycle, the term RFSH is active high and causes the selection of the refresh counter. At all other times the 8080 address is selected. The selected address represented by the terms RIA1-RIA8 are driven to the address 6 bit half select logic. These 8 bits along with address bits 9-12 are selected 6 bits at a time and routed to the 8T26 address drivers (location 4L, 5L). The sixteen pin 4096 bit dynamic RAMs used on the memory expansion card require the 12 bit address to be fed to them 6 bits at a time. The first 6 bit section is defined as the row address and the second 6 bit section is defined

as the column address. The row address gating signal, RAG, is represented by the term 5RAG. The row address is gated to the RAMs when the term 5RAG (see locations 4S, 5S) is active low.

Sheet 3 also shows the key RAM clock lines, CAS, *RAS, *CSHI and *CSLOW. The DMA address counter shown on Sheet 3 is loaded by the 16 bit data bus which has been received through 8T26s shown on Sheet 4 (locations 4N, 4P, 5N and 6N). This address counter provides the address to the IP Address Bus during DMA operations. The address is gaited to the IP Bus by the tri-state non-inverting drivers shown on Sheet 3 (locations 8S, 8R, 8P).

The counter is incremented each time the acknowledge signal is returned to the IP Bus during a DMA operation. The DMA address counter is reset to 0 when power-on or when a reset is received. Sheet 3 also shows the 8080 Phase 1 and Phase 2 clocks as well as the 7 MHz system clock, 7HQ.

	ACC	ACW		BCC	BCW
1	ZGND	ZGND	101	#BREAD	9BREAD
2	ZGND	ZGND	102	#BWRITE	9BWRITE
3	ZP05	ZP05	103	#BACK	9BACK
4	ZP05	ZP05	104	#BREADY	4BREADY
5	ZN12	ZN12	105	#BDMAHR	9BDMAHR
6	ZN12	ZN12	106	#BDMAHA	9BDMAHA
7	ZN05	ZN05	107	#BDMAGØ	9BDMAGØ
8	ZP12	ZP12	108		
9			109		
10	PTBQ	NTBQ	110		
11	PTHQ	NTHQ	111		
12			112	#BSYNC	9BSYNC
13			113		
14			114		
15			115		
16			116		
17			117		
18			118		
19			119		
20			120		
21			121		
22			122		
23	9ICPCLR	#ICPCLR	123		
24			124		
25			125		
26			126		
27			127		
28			128		
29			129		
30			130		
31			131		
32			132	9BDATA01	9BDATA00
33			133	9BDATA03	9BDATA02
34			134	9BDATA05	9BDATA04
35			135	9BDATA07	9BDATA06
36			136	9BDATA09	9BDATA08
37			137	9BDATA11	9BDATA10
38			138	9BDATA13	9BDATA12
39	9BPWCLR	#BPWCLR	139	9BDATA15	9BDATA14
40			140	4BADDR01	4BADDR00
41			141	4BADDR03	4BADDR02
42			142	4BADDR05	4BADDR04
43	9TVFFF	#TVFFF	143	4BADDR07	4BADDR06
44			144	4BADDR09	4BADDR08
45			145	4BADDR11	4BADDR10
46			146	4BADDR13	4BADDR12
47	9THMMM	#THMMM	147	4BADDR15	4BADDR14
48	9TVMMM	#TVMMM	148	ZP05	ZP05
49	PTPQØNE	NTPQØNE	149	ZGND	ZGND
50	PTPQTWØ	NTPQTWØ	150	ZGND	ZGND

TABLE 10-1 MEMORY EXPANSION BOARD SIGNAL TERM PINOUT

11.0 SERIAL LINK BOARD -- OPTIONAL

Each serial link board provides any internal system processor access to maximum of four serial input links, four serial output links and two hardware generated cursors. The two cursors can also be interactively positioned independent of processor control using Ramtek's optional cursor controllers.

Two maximum configurations are available per serial link board within which quantities can be decreased but never increased without an additional serial link card.

The first configuration consists of:

1. Four unrestricted serial input links (keyboards)
2. Four unrestricted serial output links (printers)
3. Two processor controlled cursors

The second configuration consists of:

1. Two unrestricted serial input links (keyboards)
2. Four unrestricted serial output links (printers)
3. Two processor controlled cursors also under operator interactive control

The system can support a maximum of two serial link boards.

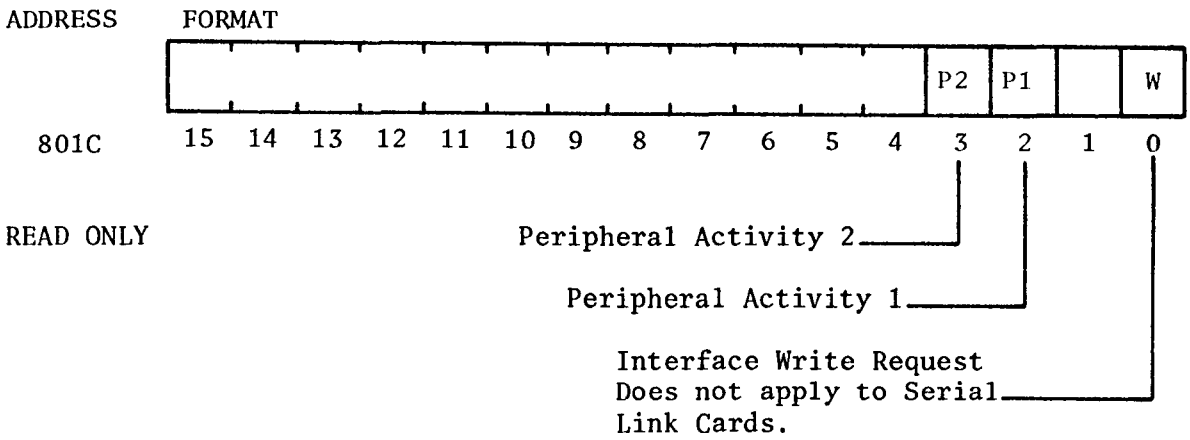
Access to the serial link input and output channels is provided through the serial link/video external connector or through an optional breakout panel. Processors interact with serial link boards through three general register sets:

1. Status Registers
2. Serial Link Registers
3. Cursor Coordinate Registers

A short form register list is shown in Figure 11-1 with corresponding addresses and bit descriptions. Figure 11-2 is a fundamental block diagram for the serial link board which shows how each register functions within the serial link architecture.

Below are the detailed register descriptions given in a typical order of access.

11.1 ACTIVITY STATUS REGISTER



Peripheral activities are recognized by the processor through the activity status register. A peripheral activity is defined as any of the following:

1. A serial word has been received and is available in any of four receiver registers.
2. The data word loaded into any of four transmit registers has completed serial transmission.
3. A cursor controller is being used to interactively position the cursor and enter has been pressed on the cursor controller or the track switch is on and the cursor position has changed.

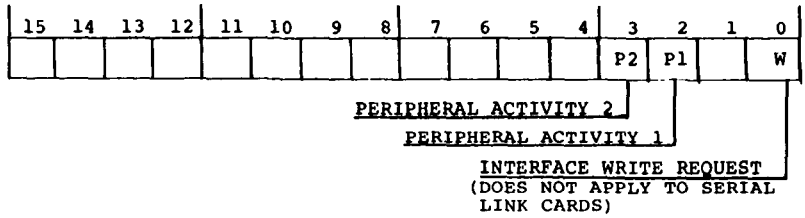
Any P bit set informs the processor that an activity has occurred on the corresponding serial link board. P1 is for serial link board/address set 1. P2 is for serial link board/address set 2. All peripheral activity bits are reset immediately after reading the activity status register.

STATUS REGISTERS

ACTIVITY STATUS REGISTER

801C

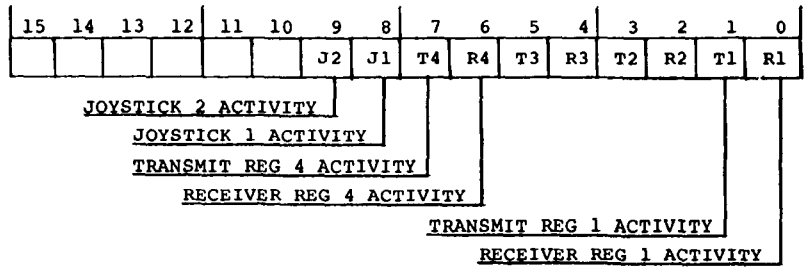
READ ONLY



PERIPHERAL STATUS REGISTER

ADDRESS SET 1	ADDRESS SET 2
803C	803E

READ ONLY

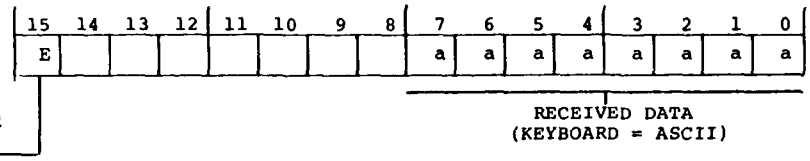


SERIAL LINK REGISTERS

RECEIVER REGISTERS (KEYBOARD)

	ADDRESS SET 1	ADDRESS SET 2
CH1	8040	8050
CH2	8042	8052
CH3	8044	8054
CH4	8046	8056

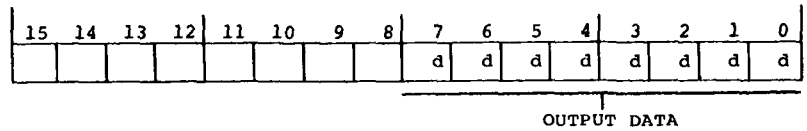
READ ONLY



TRANSMIT REGISTERS

	ADDRESS SET 1	ADDRESS SET 2
CH1	8040	8050
CH2	8042	8052
CH3	8044	8054
CH4	8046	8056

WRITE ONLY

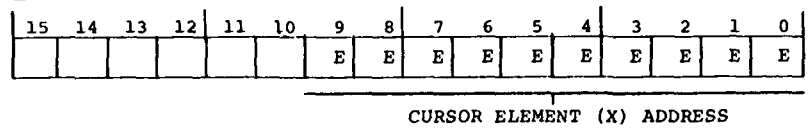


CURSOR COORDINATE REGISTERS

CURSOR ELEMENT (X) REGISTERS

	ADDRESS SET 1	ADDRESS SET 2
CH1	8048	8058
CH2	804C	805C

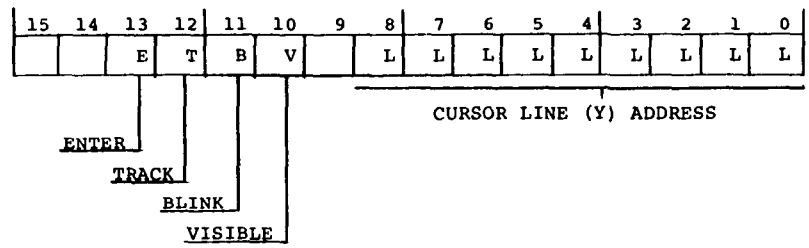
READ OR WRITE



CURSOR LINE (Y) REGISTERS

	ADDRESS SET 1	ADDRESS SET 2
CH1	804A	805A
CH2	804E	805E

READ OR WRITE



ACTIVITY STATUS REGISTER — Peripheral Activities are recognized by the processor through the Activity Status Register.

PERIPHERAL STATUS REGISTER — Peripheral Activities are sorted out by the processor through the Peripheral Status Register. Each possible peripheral event has a corresponding activity bit assigned to the event in the Peripheral Status Register.

RECEIVER REGISTERS — Receiver Registers store data words received from corresponding serial link input channels.

TRANSMIT REGISTERS — Transmit registers send data words across corresponding serial link output channels.

CURSOR ELEMENT (X) REGISTERS — Cursor Element Registers store the current cursor element address and control the cursor horizontal screen position.

CURSOR LINE (Y) REGISTERS — Cursor Line Registers store the current cursor line address and control the cursor vertical screen position and the visible/blink status.

FIGURE 11-1 SERIAL LINK SHORT FORM REGISTER LIST

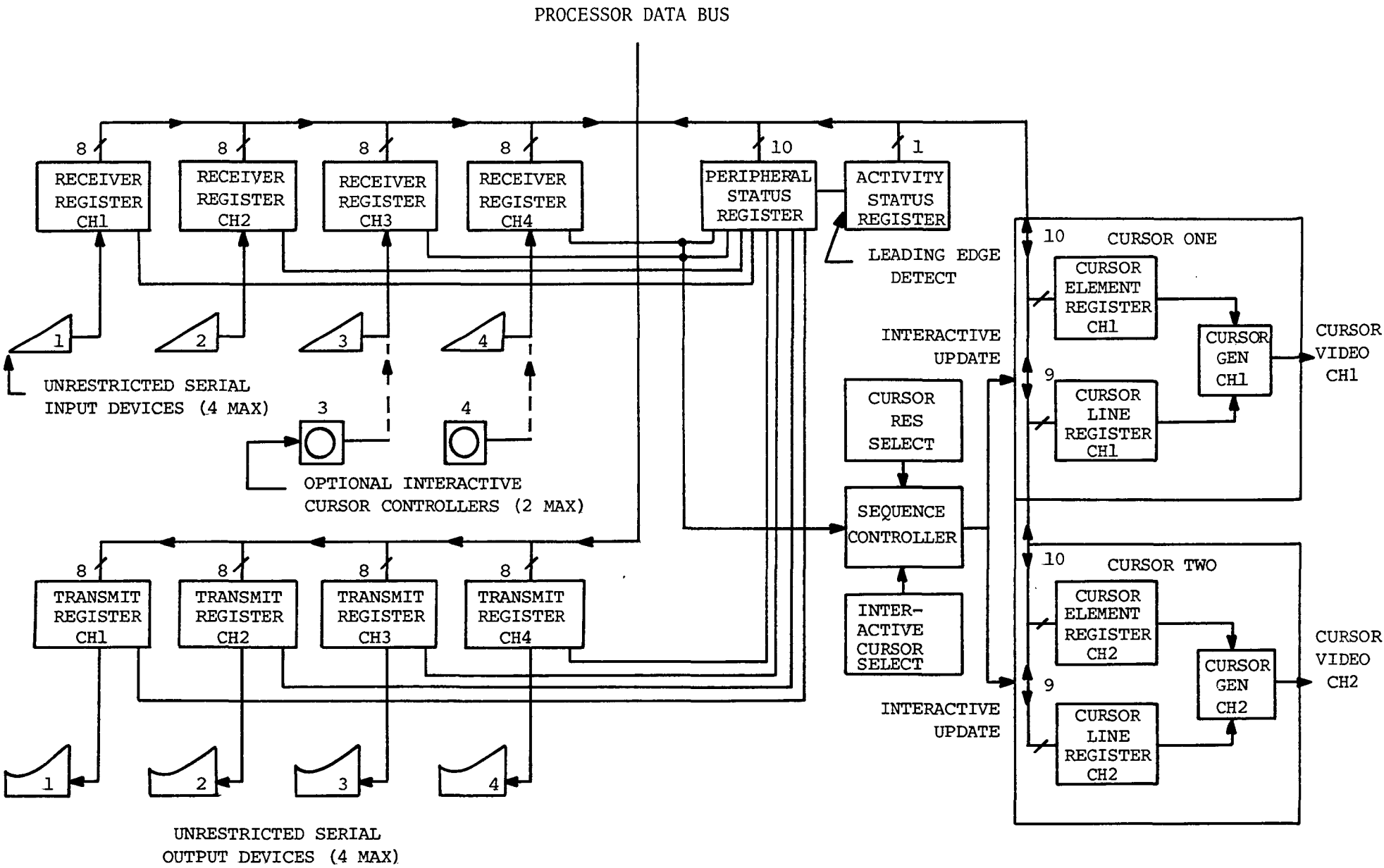


FIGURE 11-2 SERIAL LINK FUNDAMENTAL BLOCK DIAGRAM

NOTE

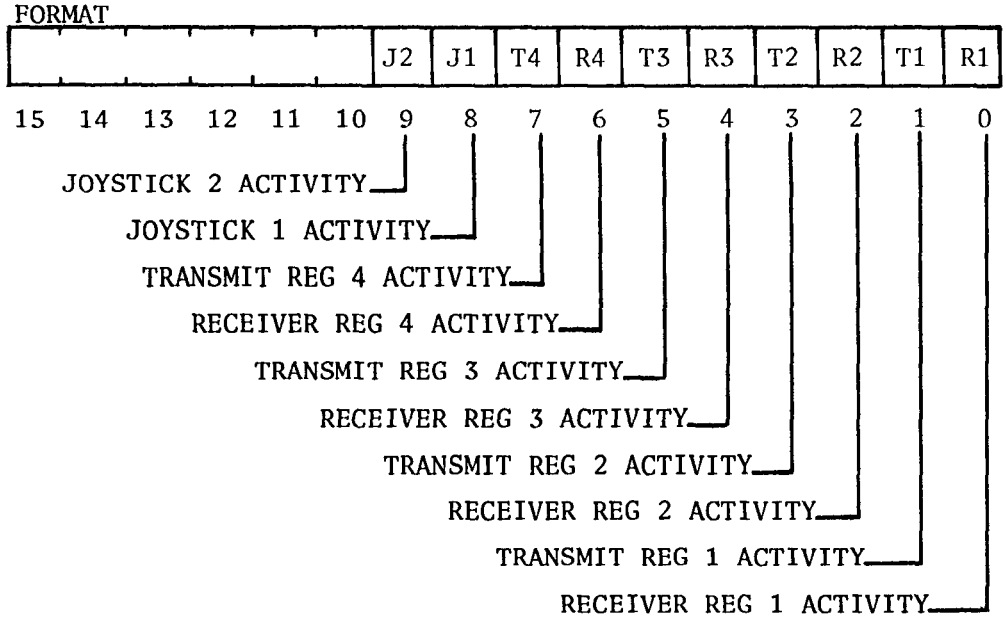
Peripheral activity bits always set upon power on. This indicates that the serial transmission channels are clear for use.

11.2 PERIPHERAL STATUS REGISTER

ADDRESS SET 1	ADDRESS SET 2
803C	803E

READ ONLY

FORMAT



Peripheral activities are sorted out by the processor through the peripheral status register. Each possible peripheral event has a corresponding activity bit assigned to the event in the peripheral status register.

An R bit set indicates that a serial word has been received and is available in the corresponding receiver register for that channel. An R bit resets immediately after reading the corresponding receiver register for that channel.

A T bit set indicates that the data word loaded into the corresponding transmit register for that channel has completed serial transmission. A T bit resets immediately upon loading the corresponding transmit register for that channel and remains reset until transmission is complete.

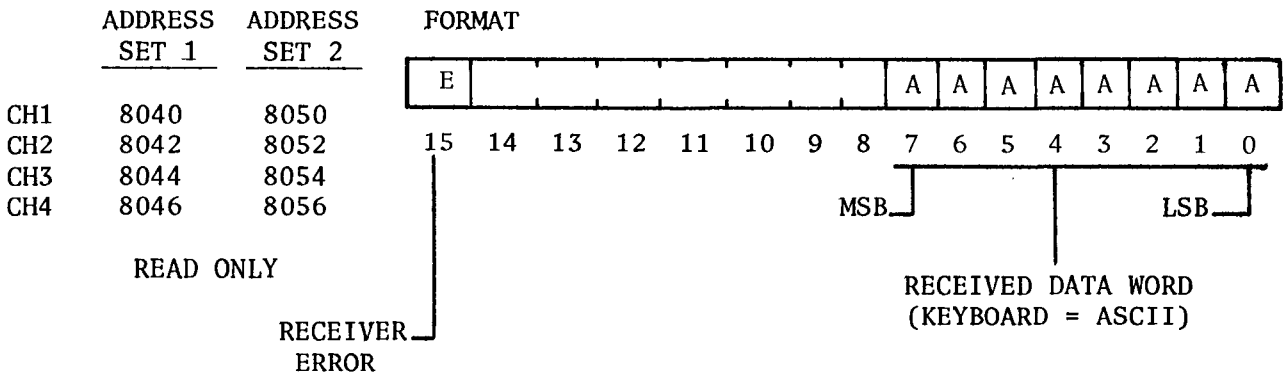
A J bit set indicates that a cursor controller is being used to interactively position the corresponding cursor for that channel and the enter switch has been pressed, OR, the track switch is on and the cursor position has changed.

A J bit resets immediately after reading the corresponding cursor line (Y) register.

NOTE

R3 and R4 are inactive when serial link input channels 3 and 4 are dedicated to cursor controllers.

11.3 RECEIVER REGISTERS



Receiver registers store data words received from corresponding serial link input channels. The A bits are the received data word, usually an ASCII character. The LSB is always in bit position zero. For data words of fewer than eight bits, received data is always right justified. The E bit set indicates that a serial transmission error has occurred and that data words have been lost or the data is invalid. Three transmission errors can occur that will set the E bit:

1. Overrun Error - Data has been lost or overwritten. Data is valid, but one or more characters have been lost. An overrun error occurs if a previously received data word has not been read by the processor before the next word is received.

2. Parity Error - Data parity does not check and the data is invalid.

3. Framing Error - The serial transmission path is faulty and the data is invalid. Framing errors occur if the serial word format or the baud rate do not match between the sending device and the receiving device on the serial link card. See sections 3.3 and 3.4 for information regarding selection of the serial word format and baud rate on the serial link card.

The E bit resets upon the serial word received if no further error occurs.

NOTE

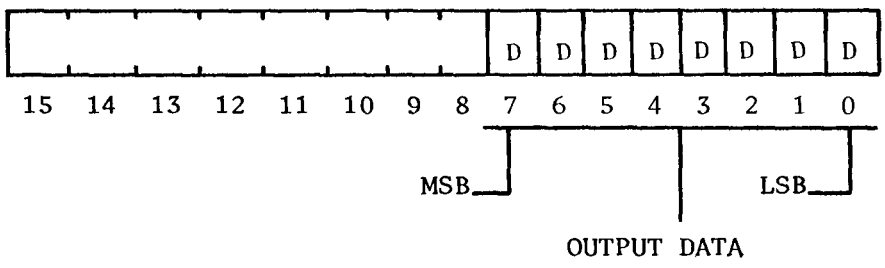
Receiver registers can only be read. Attempting to load a receiver register will result in transmitting a serial word out of the corresponding serial link output channel. Receiver registers and transmit registers share common addresses and differ only in the direction of data flow.

11.4 TRANSMIT REGISTERS

	ADDRESS SET 1	ADDRESS SET 2
CH1	8040	8050
CH2	8042	8052
CH3	8044	8054
CH4	8046	8056

WRITE ONLY

FORMAT



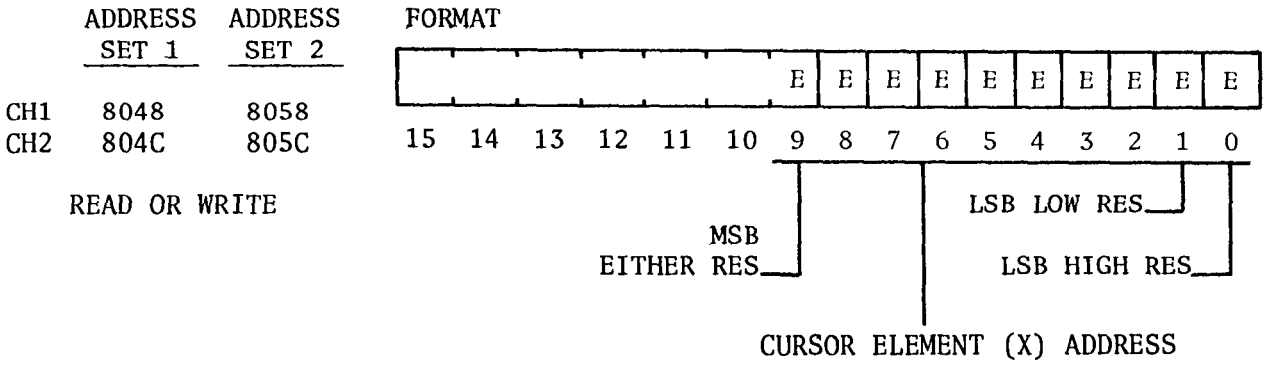
Transmit registers send data words across corresponding serial link output channels.

The D bits are the data bits in the word to be transmitted through the corresponding serial link output channel. Output data usually is an ASCII character but is not limited to ASCII data. The LSB is always in bit position zero. For data words of fewer than eight bits, data to transmit should always be right justified.

NOTE

Transmit registers can only be loaded. Attempting to read a transmit register will result in reading a received character from a receiver register. Transmit registers and receiver registers share common address and differ only in the direction of data flow.

11.5 CURSOR ELEMENT (X) REGISTERS



Cursor element registers store the current cursor element address and control the cursor horizontal screen position.

The E bits represent the corresponding cursor element address (i.e., its horizontal screen position). The element address is always left justified. That is, the MSB is always in bit position nine. The LSB is in bit position zero for high horizontal resolution cursor and in bit position one for low horizontal resolution cursors.

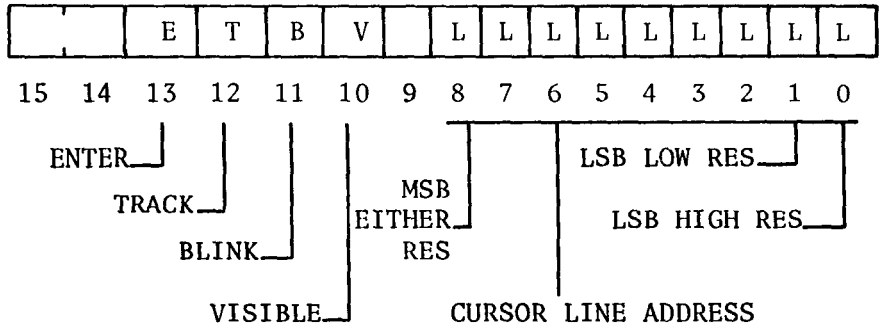
To position the cursor in the visible section of the screen, the element address must remain within the ranges as specified in the table below. Element addresses outside these ranges will place the cursor in horizontal blanking and will cause the cursor to disappear from the screen.

	Left Edge of Screen Minimum	Right Edge of Screen Maximum
Low Horizontal Resolution	0	319
High Horizontal Resolution	0	639

11.6 CURSOR LINE (Y) REGISTERS

	ADDRESS SET 1	ADDRESS SET 2
CH1	804A	805A
CH2	804E	805E

FORMAT



Cursor line registers store the current cursor line address and control the cursor vertical screen position and the visible/blink status.

The L bits represent the corresponding cursor line address (i.e. its vertical screen position). The line address is always left justified. That is, the MSB is always in bit position eight. The LSB is in bit position zero for high vertical resolution cursors and in bit position one for low vertical resolution cursors. For the standard and European monitor formats (i.e. systems with 256/512 visible lines), all line addresses place the cursor within the visible section of the screen. For the broadcast monitor format (i.e. systems with 240/480 visible lines), the cursor line address must remain within the ranges specified in the table below to remain visible. Line addresses outside these ranges will

place the cursor in vertical blanking and will cause the cursor to disappear from the screen.

Broadcast Format Only

	Top Edge of Screen Minimum	Bottom Edge of Screen Maximum
Low Vertical Resolution	0	239
High Vertical Resolution	0	479

The V (VISIBLE) bit determines the cursor visible status. When V is set, the cursor is visible upon the screen if the element and line addresses are valid. When V is reset, the cursor disappears from the screen regardless of cursor position.

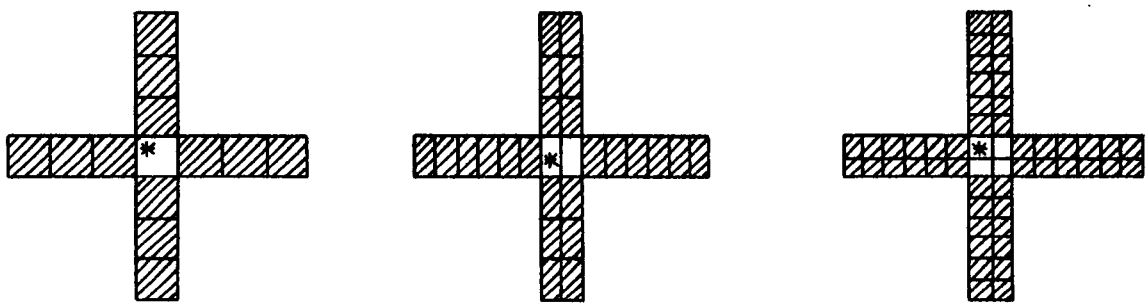
The B (BLINK) bit determines the cursor blink status. If the visible bit is set, setting the BLINK bit causes the cursor to blink. Resetting BLINK causes the cursor to remain steadily visible.

The E (ENTER) bit and the T (TRACK) bit have no effect if set or reset by the processor. The processor can determine the cursor controller switch positions by interrogating these bits if the cursor is being interactively controlled by a cursor controller.

11.7 CURSOR SIZE DESCRIPTION

The cursor appears as a cross (+) on the screen with the center element missing as in Figure 11-3 shown below. With identical monitors, the cursor retains the same size and shape for all RM-9000 series models below, regardless of format:

SERIES	MODEL	RESOLUTION	CURSOR LINES X ELEMENTS
RM9000	RM9100	LOW RES 2x3	7 LINES X 7 ELEMENTS
RM9000	RM9200	MED RES 2x6	7 LINES X 14 ELEMENTS
RM9000	RM9300	HIGH RES 5x6	14 LINES X 14 ELEMENTS



RM9100 RM9200 RM9300

7 LINES X 7 ELEMENTS 7LINES X 14 ELEMENTS 14 LINES X 14 ELEMENTS

*ADDRESSED COORDINATE

FIGURE 11-3 CURSOR SIZE AND SHAPE

11.8 OPERATIONAL DESCRIPTION - HARDWARE

11.8.1 SERIAL LINK CARD/ADDRESS SET SELECTS

Any RM9100, RM9200 or RM9300 chassis can support a maximum of two serial link cards. Each of the two cards is assigned a unique register address set and a unique peripheral activity bit position within the activity status register.

Table 1 and 2 below summarize the address set assignments for each card and specify which chassis slot to place each card into.

To select serial link card address sets, follow the switch setting drawings of Figure 4.

11.8.2 SERIAL LINK INPUT/OUTPUT CHANNELS

The serial link input and output channels can support any asynchronous serial device using long-line TTL differential or RS-232C line driver/ receivers.

The user can switch select any of 16 baud rates and any of 12 serial word formats. The baud rate determines the maximum serial word transmission

TABLE 11-1 ADDRESS SET ASSIGNMENTS

REGISTER	ADDRESS SET 1	ADDRESS SET 2
Activity Status	801C	801C
	Activity Bit in Position 2	Activity Bit in Position 3
Peripheral Status	803C	803E
Receiver Register	CH1 8040	8050
	CH2 8042	8052
	CH3 8044	8054
	CH4 8046	8056
Transmit Register	CH1 8040	8050
	CH2 8042	8052
	CH3 8044	8054
	CH4 8046	8056
Cursor Element (X)	CH1 8048	8058
	CH2 804C	805C
Cursor Line (Y)	CH1 804A	805A
	CH2 804E	805E

TABLE 11-2 CHASSIS SLOT ASSIGNMENTS

SLOT NUMBER MODEL	ADDRESS SET 1	ADDRESS SET 2
RM9100	5	6
RM9200	5	6
RM9300	6	7

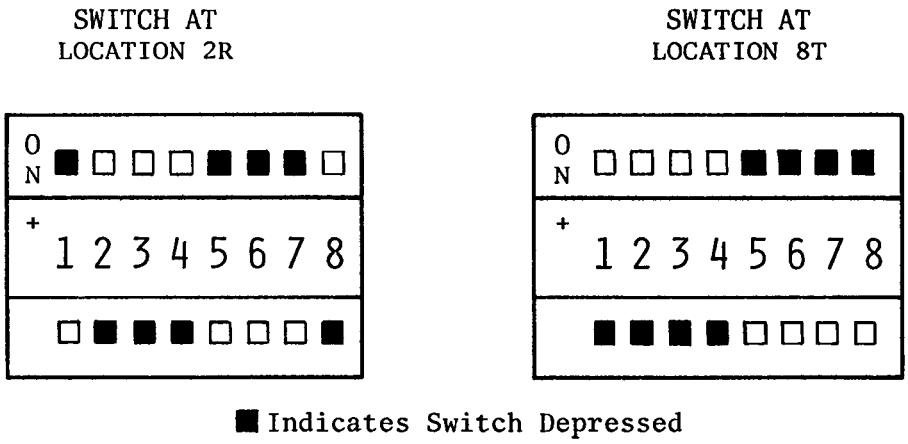
NOTE

Address set one is used in systems with only one serial link card.

REGISTER ADDRESS SET SWITCH SETTINGS

The drawings below detail the switch settings required to select serial link card register address sets.

ADDRESS SET ONE



ADDRESS SET TWO

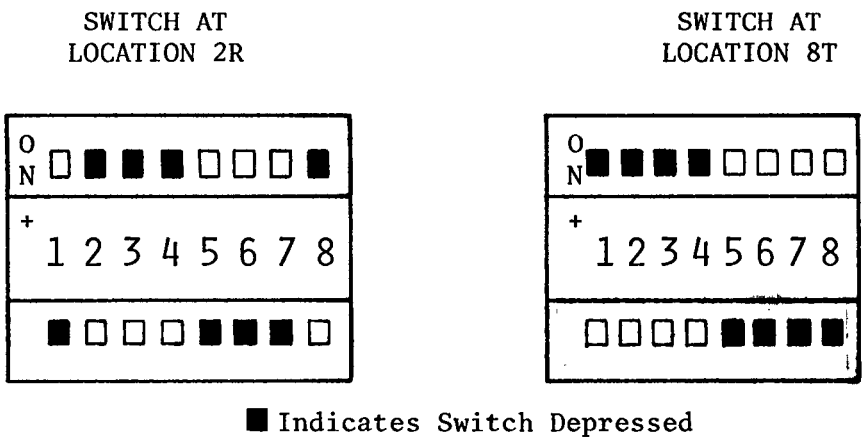


FIGURE 11-4 REGISTER ADDRESS SET SWITCH SETTINGS

rate. The serial word format determines the number of data bits, the parity sense and the number of stop bits within the serial word.

11.8.3 BAUD RATE SELECTS

The baud rate determines the maximum serial word transmission rate. Two independent baud rate generators configure the serial link input/output channels as follows:

SERIAL LINK CHANNELS

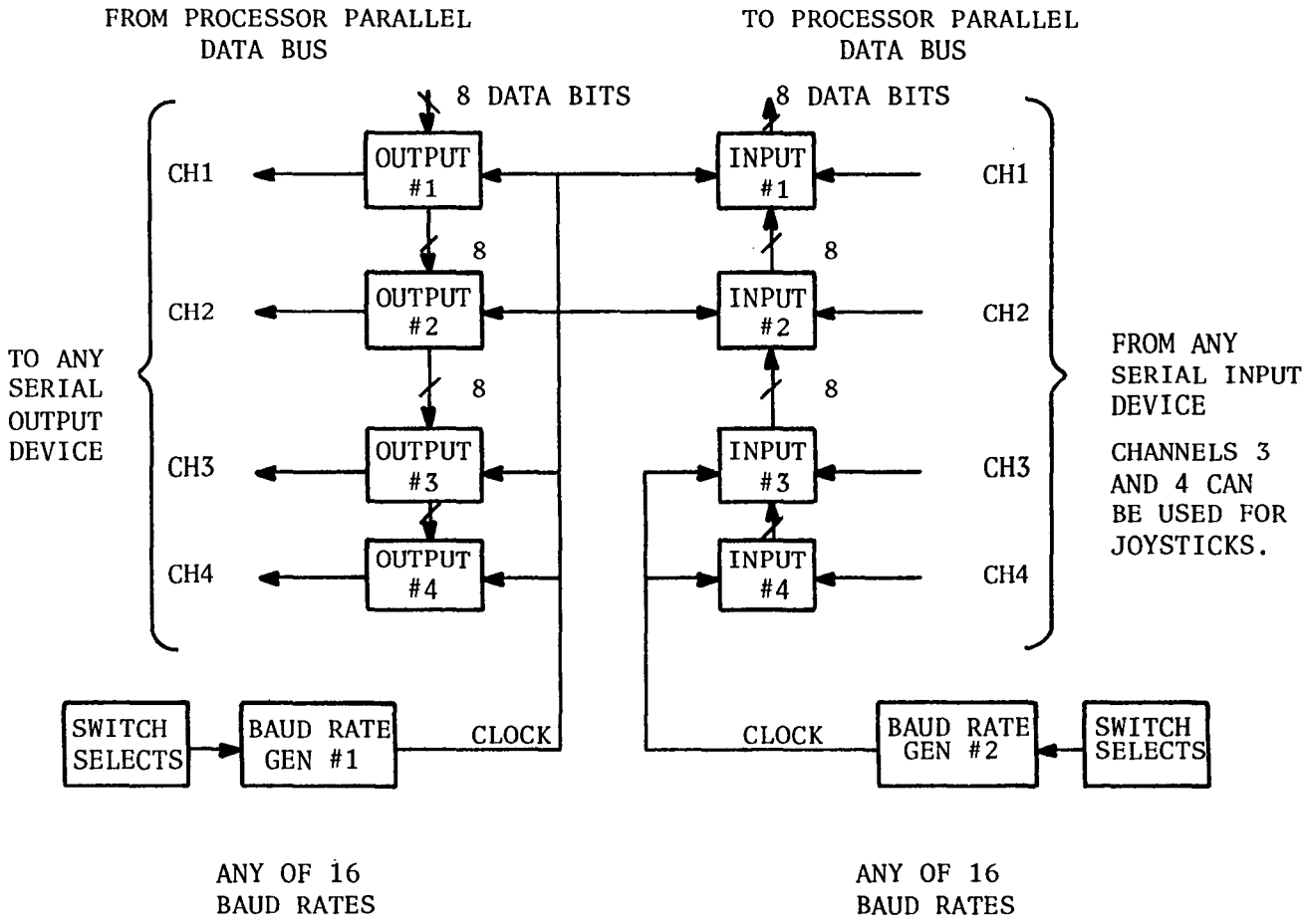


Table 11-3 specifies the baud rate generator switch select settings.

TABLE 11-3 BAUD RATE GENERATOR SWITCH SELECT SETTINGS

BAUD RATE SELECT SWITCHES - LOCATION 1N

"1" = SWITCH ON

"0" = SWITCH OFF

KEYBOARD - BAUD RATE GENERATOR #1	
SWITCH # 1 2 3 4	BAUD RATE
0 0 0 0	9600
0 0 0 1	7200
0 0 1 0	4800
0 0 1 1	3600
0 1 0 0	2400
0 1 0 1	1800
0 1 1 0	1200
0 1 1 1	900
1 0 0 0	600
1 0 0 1	300
1 0 1 0	150
1 0 1 1	134.5
1 1 0 0	110
1 1 0 1	75
1 1 1 0	50
1 1 1 1	---

JOYSTICK - BAUD RATE GENERATOR #2	
SWITCH # 5 6 7 8	BAUD RATE
0 0 0 0	9600
0 0 0 1	7200
0 0 1 0	4800
0 0 1 1	3600
0 1 0 0	2400
0 1 0 1	1800
0 1 1 0	~~900~~ 1200
1 0 0 0	600
1 0 0 1	300
1 0 1 0	150
1 0 1 1	134.5
1 1 0 0	110
1 1 0 1	75
1 1 1 0	50
1 1 1 1	---

0 1 1 1 → 900

TABLE 11-3 BAUD RATE GENERATOR SWITCH SELECT SETTINGS

BAUD RATE SELECT SWITCHES - LOCATION IN

"1" = SWITCH ON

"0" = SWITCH OFF

KEYBOARD - BAUD RATE GENERATOR #1	
SWITCH # 1 2 3 4	BAUD RATE
0 0 0 0	9600
0 0 0 1	7200
0 0 1 0	4800
0 0 1 1	3600
0 1 0 0	2400
0 1 0 1	1800
0 1 1 0	1200
0 1 1 1	900
1 0 0 0	600
1 0 0 1	300
1 0 1 0	150
1 0 1 1	134.5
1 1 0 0	110
1 1 0 1	75
1 1 1 0	50
1 1 1 1	---
JOYSTICK - BAUD RATE GENERATOR #2	
SWITCH # 5 6 7 8	BAUD RATE
0 0 0 0	9600
0 0 0 1	7200
0 0 1 0	4800
0 0 1 1	3600
0 1 0 0	2400
0 1 0 1	1800
0 1 1 0	1200
0 1 1 1	900
1 0 0 0	600
1 0 0 1	300
1 0 1 0	150
1 0 1 1	134.5
1 1 0 0	110
1 1 0 1	75
1 1 1 0	50
1 1 1 1	---

11.8.4 WORD FORMAT SELECTS

The serial word format determines the number of data bits, the parity sense and the number of stop bits within the serial word. Four serial word format selects configure the serial input/output channels as follows:

SERIAL LINK CHANNELS

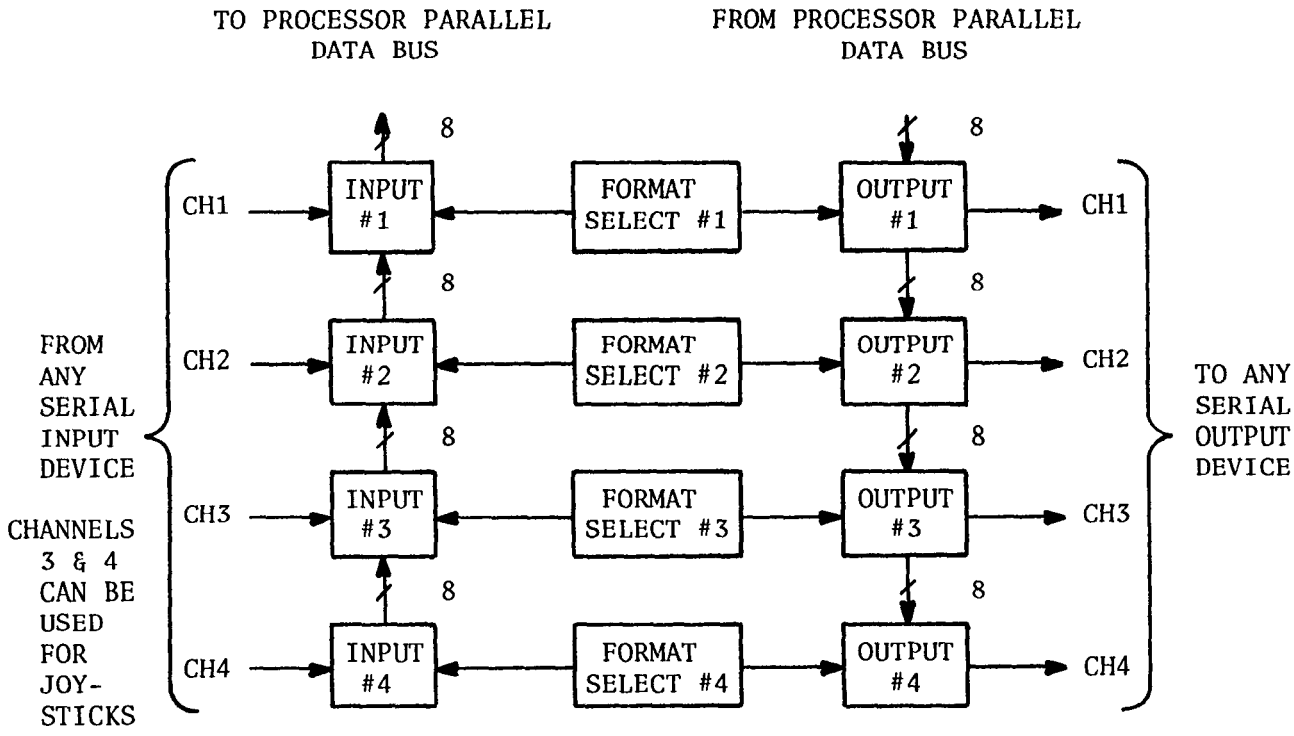
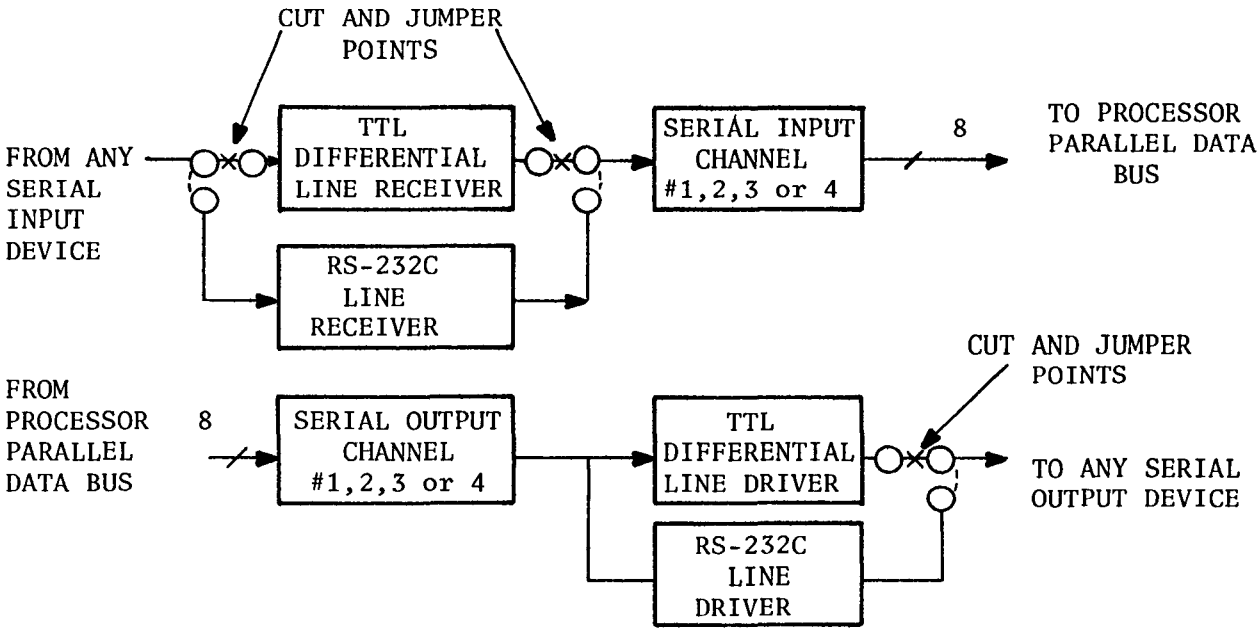


Table 11-4 specifies the word format select switch settings.

11.8.5 LINE DRIVER/RECEIVER SELECTS

Each of the four serial link input channels and each of the four serial link output channels can independently support long-line TTL differential or RS-232C line driver/receivers as follows:

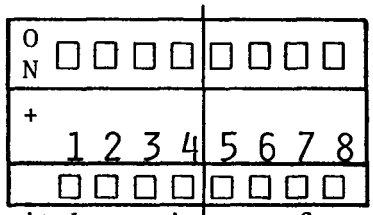


All serial link input/output channels are hardwired for long-line TTL differential line driver/receivers. Each channel can be individually configured for RS-232C by cutting appropriate P.C. traces and installing corresponding jumpers as indicated in the device select tables 5 and 6. Reconfiguration is accomplished by removing and installing jumpers as indicated in Tables 11-5 and 11-6.

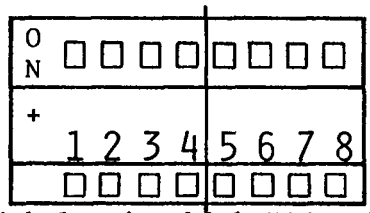
The corresponding device select diagram Figure 11-5 indicates the area on the serial link board within which jumpers are removed or installed and P.C. traces are cut.

WORD FORMAT SELECT SWITCHES

SWITCH AT LOCATION 1K



SWITCH AT LOCATION 1G



For switch settings, refer to Table 11-4 below in which "1" = Switch ON and "0" = Switch OFF.

TABLE 11-4 WORD FORMAT SELECT SWITCH SETTINGS

FORMAT SELECTS #1 AND #3

	WORD FORMAT		
SWITCH # 1 2 3 4	DATA BITS	PARITY	STOP BITS
1 1 1 0	7	EVEN	1
1 0 1 0	7	EVEN	2
1 1 1 1	7	ODD	1
1 0 1 1	7	ODD	2
0 1 1 0	7	NONE	1
0 0 1 0	7	NONE	2
1 1 0 0	8	EVEN	1
1 0 0 0	8	EVEN	2
1 1 0 1	8	ODD	1
1 0 0 1	8	ODD	2
0 1 0 0	8	NONE	1
0 0 0 0	8	NONE	2 *

FORMAT SELECTS #2 AND #4

	WORD FORMAT		
SWITCH # 5 6 7 8	DATA BITS	PARITY	STOP BITS
0 1 1 1	7	EVEN	1
0 1 0 1	7	EVEN	2
1 1 1 1	7	ODD	1
1 1 0 1	7	ODD	2
0 1 1 0	7	NONE	1
0 1 0 0	7	NONE	2
0 0 1 1	8	EVEN	1
0 0 0 1	8	EVEN	2
1 0 1 1	8	ODD	1
1 0 0 1	8	ODD	2
0 0 1 0	8	NONE	1
0 0 0 0	8	NONE	2 *

* Usual keyboard and joystick settings.

SERIAL LINE RECEIVERS

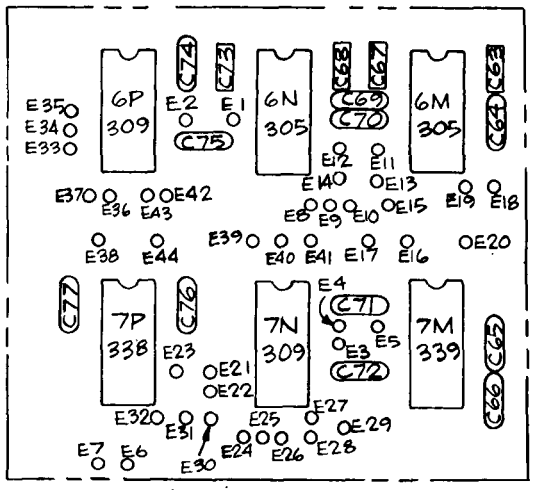
	TTL DIFFERENTIAL		RS-232C	
INPUT CHANNEL	REMOVE JUMPERS OR CUT TRACES	INSTALL JUMPERS	REMOVE JUMPERS OR CUT TRACES	INSTALL JUMPERS
1	E1 - E2 E4 - E5	E3 - E4	E3 - E4	E1 - E2 E4 - E5
2	E6 - E7 E9 - E10	E8 - E9	E8 - E9	E6 - E7 E9 - E10
3	E11 - E12 E14 - E15	E13 - E14	E13 - E14	E11 - E12 E14 - E15
4	E16 - E17 E19 - E20	E18 - E19	E18 - E19	E16 - E17 E19 - E20

TABLE 11-5

SERIAL LINE DRIVERS

	TTL DIFFERENTIAL		RS-232C	
OUTPUT CHANNEL	REMOVE JUMPERS OR CUT TRACES	INSTALL JUMPERS	REMOVE JUMPERS OR CUT TRACES	INSTALL JUMPERS
1	E22 - E23 E25 - E26	E21 - E22 E24 - E25	E21 - E22 E24 - E25	E22 - E23 E25 - E26
2	E28 - E29 E31 - E32	E27 - E28 E30 - E31	E27 - E28 E30 - E31	E28 - E29 E31 - E32
3	E34 - E35 E37 - E38	E33 - E34 E36 - E37	E33 - E34 E36 - E37	E34 - E35 E37 - E38
4	E40 - E41 E43 - E44	E39 - E40 E42 - E43	E39 - E40 E42 - E43	E40 - E41 E43 - E44

TABLE 11-6



COMPONENT SIDE VIEW

FIGURE 11-5 DEVICE SELECT DIAGRAM

The diagram above indicates the area on the component side of the serial link card within which jumpers are removed or installed and P.C. traces are cut. The jumper holes are noted by 'E' numbers E1 - E44. For installation or removal of jumpers or cutting of P.C. traces on a per channel basis. See the corresponding serial line driver/receiver device select tables, Tables 11-5 and 11-6.

11.8.6 CURSOR GENERATORS

Interactive Cursor

The serial link card provides the capability for interactive cursor positioning by dedicating two serial link input channels to Ramtek's optional cursor controllers. Each of the two cursors can be positioned independently of the other. Processor intervention is not required to update cursor position, but the processor is free to interrogate cursor position at will.

Keyboard/Joystick Keying Signal

The serial link card determines the use of input channels three and four by sensing the state of keying signal 4KEYBRD (ACC-42) originating from the back panel. Keying is done at the time of manufacture and normally requires no service. The state of 4KEYBRD is summarized in the table below.

	4KEYBRD STATE	
MODEL	LOW	HIGH
9100	Joysticks	Keyboards
9200	Joysticks	Keyboards
9300	Joysticks	Keyboards

4KEYBRD is pulled up on the serial link board with a 1K Ω resistor. A high level allows serial link input channels 3 and 4 to support any serial input device that requires processor interaction.

Grounding 4KEYBRD on the back panel by wirewrapping a small jumper between 4KEYBRD (ACC-42) and GROUND (ACW-42) dedicates serial link input channels 3 and 4 to cursor controllers.

The state of 4KEYBRD does not affect the operation of serial link output channels 3 and 4.

Cursor Resolution Keying Signals

The keying signals 4KHRESX (ACC-40) and 4KHRESY (ACC-41) determine cursor resolution by a factor of two in the horizontal and vertical directions respectively.

The RM-9000 model number determines the state of 4KHRESX and 4KHRESY as follows:

MODEL	RESOLUTION KEYWORD	4KHRESX	4KHRESY
9100	Low Resolution 2x3	Low	Low
9200	Med Resolution 2x6	High	Low
9300	High Resolution 5x6	High	High

Keying is automatic and normally requires no service. The PC backplanes for each RM-9000 series model are configured per the above table with PC traces, not wirewrap jumpers.

4KHRESX and 4KHRESY are pulled upon the serial board with a 1K Ω resistor. Grounding these signals is done on the backpanel as follows:

TO GROUND	PIN	IS CONNECTED TO
4KHRESX	ACC-40	ACW-40
4KHRESY	ACC-41	ACW-41

502643
This board is no longer avail.
Vol. II contains new dwg. 504416

11.9 CIRCUIT DESCRIPTION

This section of the manual contains a description of the circuitry used in the serial link card. Discussion begins with the block diagram shown on sheet 1 of the serial link logic diagrams in the diagrams section. Next, important circuits are described in detail using the block diagram and the schematics.

Complete schematics are given in the diagrams section along with flow charts and information sheets. Reference drawing number 504416.

11.10 BLOCK DIAGRAM - SHEET 1

The serial line driver/receiver circuits accomplish level translations between the voltages/currents of the serial line and a TTL compatible equivalent of the serial line. Particular line driver/receiver devices are selected by cuts and jumpers on the board.

Receiver registers are implemented with one half of a UART (Universal Asynchronous Receiver/Transmitters) which accomplishes serial to parallel conversion. The UART latches the received word, signals the peripheral status register that a word has been received and gates the word onto the processor data bus when the processor reads the associated receiver register.

Transmit registers are implemented with the other half of a UART. When the processor loads a transmit register, the parallel word is loaded into the UART transmit buffer and serial transmission begins. The UART shifts out the serial word and signals the peripheral status register when transmission is complete and the serial line is clear to send another character.

The serial word transmission rate is controlled by two switch selectable baud rate generators which clock the UART. The baud rate generators derive their timing pulses from a crystal controlled square wave oscillator.

The peripheral status register stores the receiver/transmit register activity bits. Receiver activity bits come from the UARTs. These bits tell when a serial word has been received by the UART, is latched in a UART receiver buffer and is available to the processor. Transmit activity bits come from the UARTs. These bits tell when a serial word that was loaded into a UART transmit buffer has completed transmission.

The peripheral status register activity bits are leading edge detected. The leading edge of any activity bit setting will set the peripheral activity bit in the activity status register.

The cursor generators translate cursor coordinates contained in the cursor line and element registers into a composite cursor video picture signal with the cursor center located at the stored cursor coordinate value.

The cursor line and element registers store static cursor coordinates and control the cursor screen position. The processor can load or read the cursor coordinates upon command. Upon loading, coordinate values are

taken from the processor data bus. Coordinates are multiplexed onto the processor data bus upon readback by the cursor readback multiplexer.

The cursor line and element registers are also automatically incremented or decremented without processor control when using an interactive cursor positioner.

When interactively positioning a cursor, the sequence controller interprets data words from cursor controllers attached to serial input channels 3 and 4 and issues appropriate increment/decrement commands to the associated cursor line and element registers.

The sequence controller is a microprogrammed control unit for the serial link board that has three main purposes: 1) update cursor coordinates per commands from interactive cursor positioners, 2) delay the processor for a minimum of 400 nsec when it attempts to read the receiver registers or load the transmit register, and 3) set the peripheral activity bit within the activity status register upon a leading edge setting of any activity even bit within the peripheral status register.

The sequence controller's microprogram resides in PROM. The program sequence is dictated by conditions that the program tests. The program can branch or loop upon the conditions tested. The sequence states are decoded and used as control pulses to load registers, increment/decrement counters, delay the processor, acknowledge peripheral events and set/reset register bits.

All processor 'Load Register' and 'Read Register' commands are decoded from the processor address bus by the register address decode logic. The address decodes load data into registers or, when reading, gate data onto the processor data bus. Some address decodes are examined by the sequence controller.

11.11 DETAILED CIRCUIT DESCRIPTION

11.11.1 CURSOR LINE AND ELEMENT REGISTERS

The cursor element registers are three-stage synchronous presetable binary up/down counters (CH1-6J, 6K, 6L; CH2-8J, 8K, 8L).

The processor loads the cursor element register with the cursor element address by doing a two byte register load command. The element address is taken from the IP data bus BDATA__ and loaded into the cursor element register on the high byte only. Signal 5WIØX__ is the processor element register load pulse which is a high byte register decode for the particular element register being loaded as tabulated below:

Cursor Element Register		Register Decodes	
Cursor Channel	Signal Term Name	Register Decode Addr. Set 1	Addr. Set 2
CH 1	5WIØX1	8048 - 8049	8058 - 8059
CH 2	5WIØX2	804C - 804D	805C - 805D

During processor loads, the least significant data bit is suppressed for horizontal resolution cursors by the horizontal resolution control gate (5C). Keying signal 4KHRESX is high for high horizontal resolution cursors and low for low horizontal resolution cursors.

The processor reads the cursor element register by doing a two byte register read command. The cursor element address XREG__ is multiplexed onto the processor data bus by the cursor readback multiplexer selectors of another page.

Interactive updates of the cursor element register by doing a two byte register read command. The cursor element address XREG__ is multiplexed onto the processor data bus by the cursor readback multiplexer selectors of another page.

Interactive updates of the cursor element register are controlled by the signals 5UPDTX__ and UBDOO. 5UPDTX__ is a '7HQ wide' update pulse issued by the sequence controller of another page.

Each update pulse causes the counter to count once. The direction of the count (i.e., whether to add one or subtract one from the current element address) is determined by the state of UBDOO. UBDOO is high for counting up and low for counting down. UBDOO is a tri-state signal from the UARTs and is valid only while the update pulse 5UPDTX__ is active.

For high horizontal resolution cursors, one update pulse (5UPDTX__) is issued for each update command received from a cursor controller. For low resolution cursors, two updates are issued in rapid succession for each word received.

The cursor line registers are three-stage synchronous presettable binary up/down counters (CH1-5J, 5K, 5L; CH2-7J, 7K, 7L). Also a part of a cursor line register is a Hex D flip-flop storage register (CH1-4E; CH2-4D) and a data selector (CH1-3E; CH2-3D).

The processor loads the cursor line register with the cursor line address by doing a two byte register load command. The line address is taken from the processor data bus (BDATA__) and loaded into the cursor line registers on a high byte only. Signal 5WIØY__ is the processor line register load pulse which is a high byte register decode for the particular line register being loaded as tabulated below:

Cursor Line Register		Register Decoder	
Cursor Channel	Signal Term Name	Register Decode Addr. Set 1	Addr. Set 2
CH 1	5WIØY1	804A - 804B	805A - 805B
CH 2	5WIØY2	804E - 804F	805E - 805F

Note that the processor load pulse (5WIØY__) loads both the counters with the cursor line address and the hex D flip-flop with enter, track, blink

and visible bits from the processor data bus. The data selector is switched to the processor data bus during processor loads.

During processor loads, the least significant data bit is suppressed for low vertical resolution cursors by the vertical resolution control gate (5C).

Keying signal 4KHRESY is high for high vertical resolution cursors and low for low vertical resolution cursors.

The processor reads the cursor line register by doing a two byte register read command. The cursor line address YREG__ and the status bits are multiplexed onto the processor data bus by the cursor readback multiplexer selectors of another page.

Interactive updates of the cursor line register (i.e., adding or subtracting a line from the current line address) are controlled by the signals 5UPDTY__ and UBDO2. 5UPDTY__ is a '7HQ wide' update pulse issued by the sequence controller of another page. Each update pulse causes the counter to count once. The direction of the count is determined by the state of UBDO2. UBDO2 is high for counting up and low for counting down. UBDO2 is a tri-state signal from the UARTs and is valid only while the update pulse 5UPDTY__ is active.

For high vertical resolution cursors, one update pulse (5UPDTY__) is issued for each update command received from a cursor controller. For low vertical resolution cursors, two updates are issued in rapid succession for each word received.

Interactive updates of the status bits (enter, track, blink, visible) within the cursor line register are controlled by the signal 5ETBV__. 5ETBV__ is a '7HQ wide' load pulse issued by the sequence controller of another page. Each load pulse temporarily switches the data selector to the UART data bus and loads the Hex D flip-flops register with enter, track, blink and visible data from the UARTs. This data indicates the position of the status switches on the cursor controller.

11.11.2 CURSOR VIDEO GENERATION

The cursor video generators transform cursor element and line addresses contained in the cursor line and element registers into a composite video cursor image.

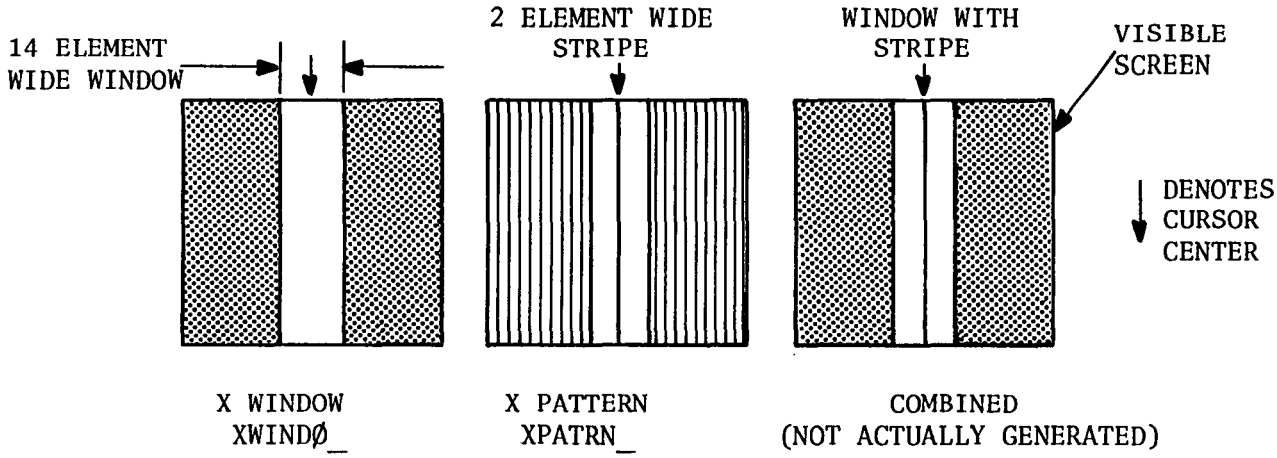
11.11.3 CURSOR VIDEO ELEMENT GENERATION

The cursor element counters are three-stage synchronous presettable binary up/down counters. (CH1 - 6H, 6G, 6F; CH2 - 8H, 8G, 8F). Since these are totally synchronous counters the clock must be present to count and it must be present during the load pulse.

The cursor element counters count high resolution elements during the visible portion of the screen. The counters are disabled during horizontal blanking. The counter enable/disable signal is THBBB, horizontal blanking. The counters are loaded with the true cursor element address directly from the cursor element register every line during horizontal blanking. The load signal is 5THMMM, horizontal memory synchronization. The 'count direction' flip-flop (CH1-5B; CH2-9F) is reset to count down when the cursor element counters are loaded. 5XDOWNX__ is low to count down. The counters hold the count until the end of blanking. Starting with the first visible element, the counters count down from the loaded value to zero. At zero, signal XSTØP__ halts the counter at zero for an extra count and sets the 'count direction' flip-flop to count up. 5XDOWNX__ is high to count up. The counter remains at zero for two high resolution elements. Zero represents the center of the cursor. Upon reversing count direction, XSTØP__ becomes false and allows the counter to begin counting up. The counter continues to count up across the remaining portion of the visible line. At blanking the counter is disabled and is ready to be reloaded with the true cursor element address.

The element decode logic consists of several gates. (CH1-6E, 6C, 6D, 6B; CH2-8E, 8C, 8D, 9C). The element decode logic examines the cursor element counter states and forms an X window XWINDØ__ and an X pattern XPATRN__ as shown in the sketches below. The sketches represent the

patterns that would appear on a monitor if the signals were jumpered to the cursor video output connector pins.



XWINDØ__ is formed by gates looking for all zeros in bit positions 3 through 9 and at least one zero in any bit position 0 through 2. XWINDØ__ becomes true for 14 consecutive high resolution elements every line centered around the cursor center.

XPATRN__ is formed by gates looking for all zeros in bit positions 0 through 2. XPATRN__ becomes true for one high resolution element many times across every line. Interest focuses upon the cursor center where XPATRN__ becomes true for 2 consecutive high resolution elements. This two element stripe is centered within the X window.

A typical count pattern of the element counters is shown in Figure 11-6 below with the associated waveforms generated by the element decode logic. The cursor is to be placed on element 16 in the high resolution sense. The element counters have been loaded in horizontal blanking with the value 16. The first entry on the left hand side of the diagram represents the first visible element on a line. The 1st right hand entry is not the last element on a line.

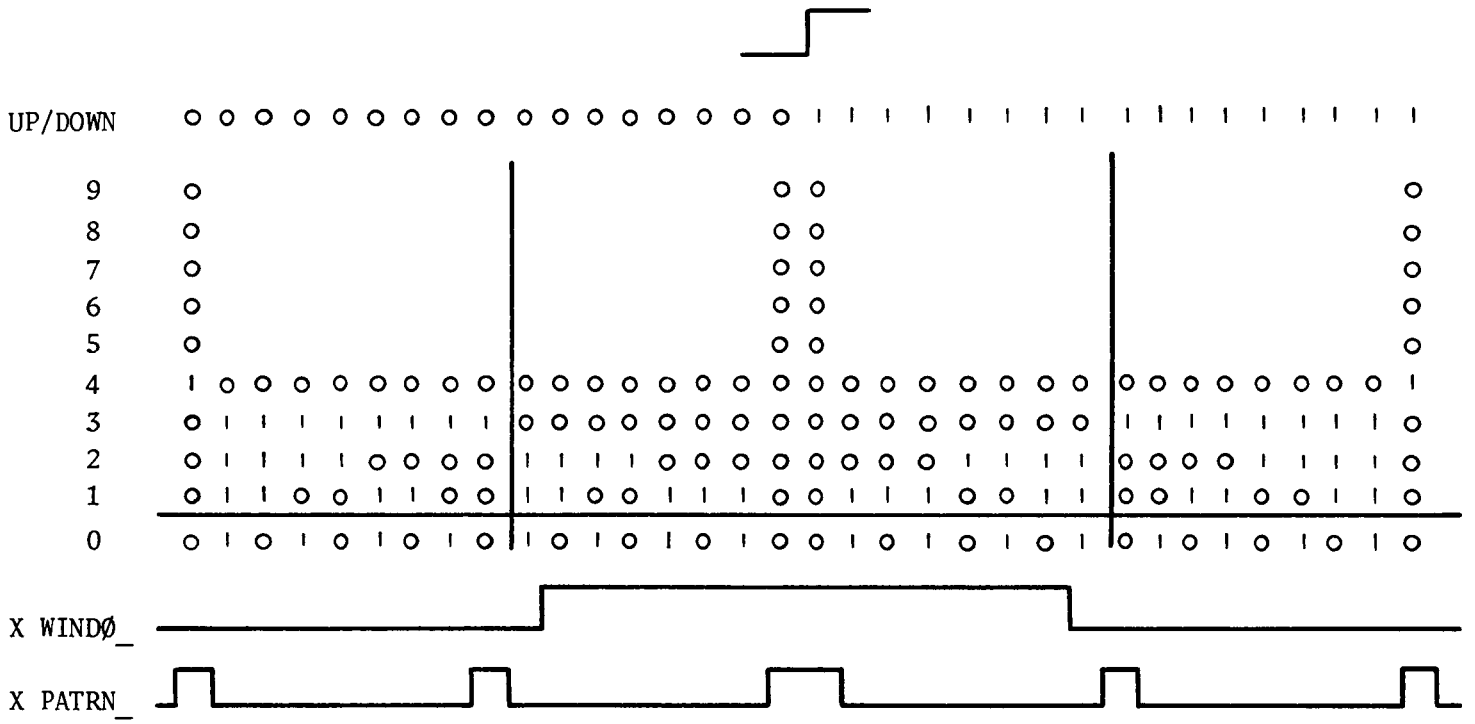


FIGURE 11-6 ELEMENT COUNTER COUNT PATTERN

11.11.4 CURSOR VIDEO LINE GENERATION

The cursor line counters are two-stage synchronous presettable binary up/down counters. (CH1-5H, 5G; CH2-7H, 7G). Since these are totally synchronous counters, the clock must be present to count and it must be present during the load pulse.

The cursor line counters count vertical lines during the visible part of a field for both fields. The counters are disabled during vertical blanking. The primary counter enable/disable signal is a reclocked version of vertical blanking 5TVBBB*1. The other enable signal INCR*__ combined with the NOR gate of location 7D is explained below. The counters are loaded with the true cursor line address directly from the cursor line register once every field. The cursor line address LSB is routed to an AND gate as explained below. The load pulse is vertical memory synchronization 5TVMMM, a '7HQ wide' pulse occurring only once per field on the last line in vertical blanking. 5TVMMM is coincident with 5THMMM. If the cursor line address contained in the cursor line register is an odd number (i.e. YREG00__ is high), then the true value loaded into the cursor line counters gets incremented by one in an odd field immediately after being loaded. The value is not incremented in an even field. The increment pulse INCR*__ is created by a four-input AND gate (8A) and a D flip-flop (6A). INCR*__ is a '7HQ wide' pulse that occurs only once per odd field for odd line addresses during vertical blanking immediately after the load pulse, 5TVMMM. For even line adding immediately after the load pulse, 5TVMMM. For even line addresses, INCR*__ is inactive. The 'count direction' flip-flop (7A) is reset to count down. The cursor line counters are loaded. 5YDOWNX__ is low to count down. The counters count until the end of blanking. Starting with the first visible line, the counters count down from the loaded value to zero. Zero represents the center of the cursor. At zero, signal YR4_ sets the 'count direction' flip-flop to count up. 5YDOWNY__ is high to count up. The counter now counts up and continues to count up for the remaining visible lines in the field.

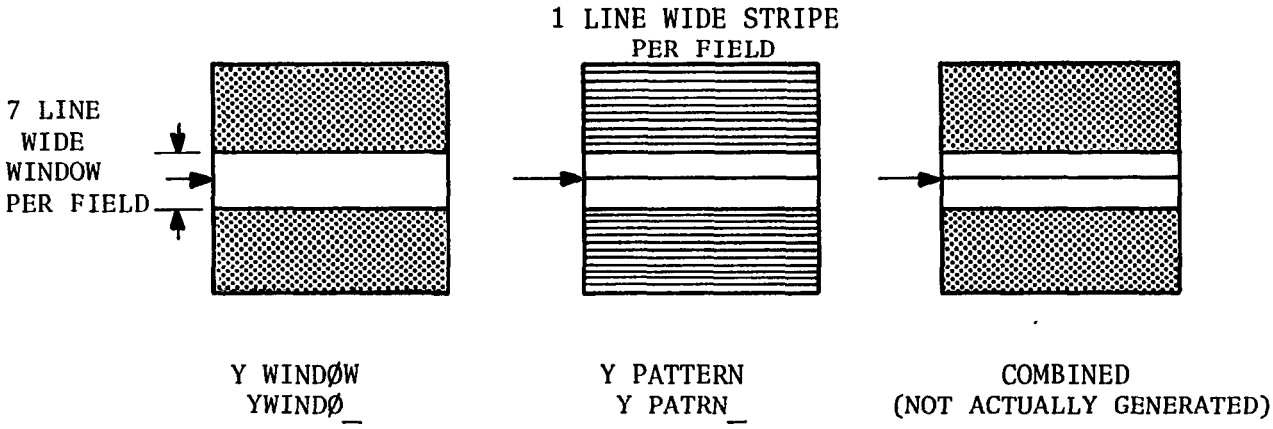
At vertical blanking, the counter is disabled and is ready to be reloaded with the true cursor line address.

The line decode logic consists of several gates.

CH1 - 5E, 5C, 7C, 4C, 5D

CH2 - 7E, 5C, 7C, 6B, 7D

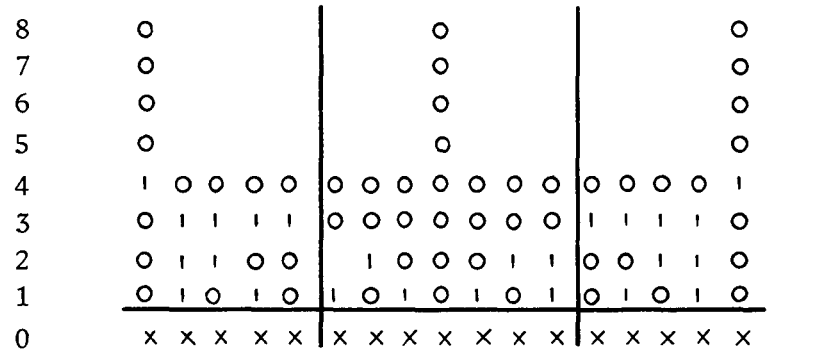
The line decode logic examines the cursor line counter states and forms a Y window YWONDØ__ and a Y pattern YPATRN__ as shown in the sketches below. The sketches represent the patterns that would appear on a monitor if the signals were jumpered to the cursor video output connector pins on the cursor video page.

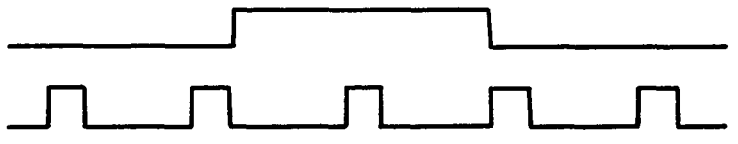


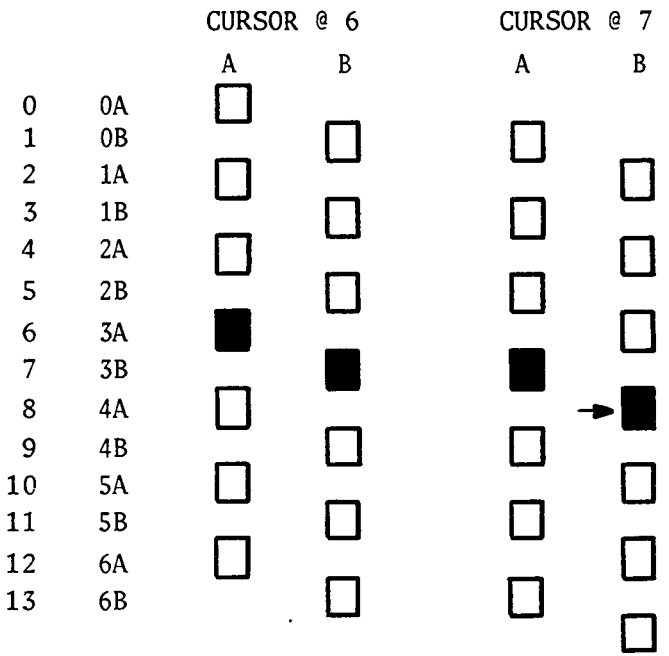
YWINDØ__ is formed by gates looking for all zeros in bit positions 3 through 8. YWINDØ__ becomes true for several consecutive lines per field centered around the cursor center.

YPATRN__ is formed by gates looking for all zeros in bit positions 1 and 2. YPATRN__ becomes true for one line every fourth line per field. Interest focuses upon the cursor center where YPATRN__ is true for one line. The one line stripe is centered within the Y window.

A typical count pattern of the line counters is shown in Figure 11-7 below with associated waveforms generated by the decode logic. The cursor is to be placed on line 8 in the low resolution sense. The line counters have been loaded in vertical blanking with the value 8. The first left-







HIGH RESOLUTION FIELD A/B LSB

	0	0	0 EVEN	0A	
LOW	0	0	1 ODD	0A	
RES	0	1	0 EVEN	0A	
	0	1	1 ODD	0A	
	1	0	0 EVEN	0A	
	1	0	1 ODD	0B	
	1	1	0 EVEN	0B	
	1	1	1 ODD	1A	ADD 1

FIGURE 11-7 LINE COUNTER COUNT PATTERN

hand entry of the diagram represents the first visible line in a field. The last right-hand entry is not the last line in the field.

11.11.5 BLINK CLOCK AND BLINK DECODE

The blink clock is a two stage counter (3B, 3A) that counts fields. Signal 5TVMMM vertical memory synchronization, is a '7HQ wide' pulse occurring once per field.

The blink decode logic (gates 4A, 4B, 4C) form a composite signal 5BLINC__ that allows the cursor to be turned on and off through the visible bit VISB__ from the cursor line register. The cursor is commanded to blink through the blink bit BLNK__ also from the cursor line register. When BLNK__ is set, the gate at 4B combines the blink clock outputs BNKC1 and BNKC2 to form a 75% duty cycle blink signal with a period of about one second.

11.11.6 CURSOR VIDEO

The cursor video logic (7B, 8B, 10E, 10D) combines the outputs of the cursor video element and line generators and the blink decode logic to form a composite cursor picture. The D flip-flop of location 10E causes the cursor video output to be one high resolution element off with respect to horizontal blanking.

11.11.7 PROCESSOR REGISTER ADDRESS DECODE

The processor register address decode logic decodes the IP address bus and constructs individual load pulses and read enables for each register contained on the serial link board.

The address set select switches are used to select one of two possible register address sets. There are two switch sets (8T and 2R). The switch settings required are detailed within this manual under the hardware operational description section - serial link card/address set selects. Depending upon the number of serial link boards within a chassis.

The cursor line and element register load decodes are active only for the high byte and also are '7HQ wide'. The write control line is leading edge detected and used to form these load pulses. All other decodes including load pulses and read enables, are active for both the low and high byte and remain active for the entire processor read or write command.

11.11.8 TRANSMIT/RECEIVER REGISTERS

Transmit and receiver registers are implemented with UARTs (3E, 3K, 3H, 3G). Detailed pin functions of a UART can be found in any UART specification sheet.

Basically the UART can be broken into three fundamental parts: A transmitter section, a receiver section and a common control section. The transmitter section converts up to 8-bit parallel data into serialized data. The 8-bit parallel data is taken off the processor data bus BDATA__ (pins 26-33) and loaded into the UART transmitter buffer with the signal 5WIØU__ (pin 23). 5WIØU__ is a register load pulse from the processor register address decode logic. The loaded data is shifted out SERØT__ (pin 25) at the earliest possible moment. TBEU__ (pin 22) is the transmit register flag bit. When high, the UART transmit buffer is empty and can be loaded with a character. When TBEU__ is low, the UART is busy transmitting. TBEU__ is a 'T' bit within the peripheral status register. The UART transmitter clock KEYCLK (pin 40) is driven by the baud rate generator.

The receiver section converts up to 8-bit serial data into parallel data. The serial data SERIN__ (pin 20) enters the UART and is latched internally. A receiver register flag bit DAU__ (pin 19) becomes active high upon the word being received. DAU__ is an 'R' bit within the peripheral status register. When the processor reads the receiver register, signal 5KRDAU__ (pin 4) enables the internal latched data onto the parallel data bus UBD__ (pins 5 - 12). Also, if any receiver errors occurred, they appear as an overrun error ØVR__ (pin 15), framing error FE__ (pin 14) or as a parity error PE__ (pin 13). Signal 5KRDAU__ (pin 18) also acknowledges to the

UART that the data has been read. This resets the data received flag (DAU__). The UART receiver clock (pin 17) is derived from one of two baud rate generators (KEYCLK or JØYCLK).

The common control section establishes the serial word format (pins 34-39). The serial word format select switch settings are described within this manual under the hardware operational description section.

The Serial Line Receiver circuits (6N, 6M, 7M) convert the serial line voltages into a TTL equivalent of the serial line usable by the UARTs. The serial line driver circuits (7N, 6P, 7P) convert the TTL signal from the UARTs into voltages appropriate for the serial lines.

The baud rate select switches (IN) determine the divide factor the baud rate generators (1P, 1R) use to formulate the baud rate clocks from the crystal oscillator output MEGACLK. The baud rate select switch settings are detailed within this manual under the hardware operational description section.

*UBD__ is a tristate bus. Data is valid only as long as the signal on pin 4 is active low.

11.11.9 CURSOR READBACK MULTIPLEXER

The selectors of the cursor readback multiplexer (10J, 10H, 9J, 9H, 9G, 10G) direct the data flow of the cursor line and element registers YRES__ and XRES__ when the processor performs a read. The bus drivers (8N, 9N, 10N) gate the selected data CD__ onto the processor data bus 9BDATA__.

The peripheral status leading edge detect circuits produce a '7HQ wide' signal STATREQ upon any leading edge set of the transmit (T) and receiver (R) register bits within the peripheral status register. The sequence controller controls the clocking to the second set of D flip-flops (7R, 7T) with the signal 5STØPCLK. 5STØPCLK is high only when the sequence controller is in the idle state, state zero. For all other states 5STØPCLK is low stopping all leading edge detects.

11.11.10 SEQUENCE CONTROLLER

1. To delay the processor a minimum of 500 ms when it reads a receiver register or loads a transmit register.

2. Set the activity status register bit upon a leading edge setting of any bit within the peripheral status register.

3. Update the cursor line and element registers during interactive control.

The sequence controller is a microprogrammed control unit. It is programmed to test conditions and issue instructions (operations) based upon the outcome of the conditions it tests. The sequencer's program resides in PROM (2E, 2F, 2H, 2G). The program directs the sequencer states through the signals NEXT__. The exact size to which the sequencer will go next depends upon the logic level of the current condition under test. Each condition is examined individually and selected by the program through the condition select signals CS__. At each state, the sequencer issues an operation (including NOP's) through the lines OPC__). These op codes are separated into individual control lines by the operation decode logic (2K, 2L, 2N).

The sequencer's operation is best described by the flow on sheets 15 and 16 of the serial link logic diagrams. Each sequencer state is given a symbol along with the current condition it is testing or the current operation it is giving. The first chart uses words to describe the decisions made and the operations issued. The second chart shows the sequencer states, and the term name of the operation issued.

The sequencer's microprogram contained in the 256 x 4 PROMs is shown in Figure 8 below.

FIGURE 11-8

SERIAL LINK CONTROLLER MICROPROGRAM

	NØW32 NØW16 NØW08 NØW04 NØW02 NØW01	NEXT32 NEXT16 NEXT08 NEXT04	NEXT02 CS08 CS04 CS02	CS01 ØPC8 ØPC4 ØPC2	ØPC1	5ENDRDY 5STØPCLK	LABEL	TEST	RESULTS	OP - TRUE/FALSE DESTINATION
0	0 0 0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	0	1 1	IDLE	DOIT		NØP/STOPCLK RWU1/*
1	0 0 0 0 0 1	0 0 0 0	1 0 0 1	0 0 0 0	0	1 0	RWU1	RØWIØU		RDAE1/REQJØY
2	0 0 0 0 1 0	0 0 0 1	0 0 1 1	0 0 0 0	0	1 0	REQJØY	JØYREQ		JOY3/RWS1
3	0 0 0 0 1 1	0 0 1 0	0 0 0 0	0 1 0 1	0	0 1	RDAE1	---		KRDA RDAE2
4	0 0 0 1 0 0	0 0 0 1	1 0 0 1	1 0 0 0	0	1 0	RWS1	5 IØSTAT		WUD1/SETST1
5	0 0 0 1 0 1	0 0 1 1	1 1 0 1	1 0 0 0	0	1 0	JØY3	5 JØY3		SS3/SS4
6	0 0 0 1 1 0	0 0 0 1	1 0 0 1	1 0 0 0	0	1 0	WUD1	5 IØSTAT		*/SETST1
7	0 0 0 1 1 1	0 0 0 0	0 0 0 0	0 1 0 1	0	1 0	SETST1	---		STAT IDLE
8	0 0 1 0 0 0	0 0 1 0	0 1 1 1	1 1 0 1	1	1 0	RDAE2	---		KRDA RDAE3
9	0 0 1 0 0 1	0 0 1 0	1 0 0 0	0 1 0 1	1	1 0	RDAE3	---		KRDA RDAE4
10	0 0 1 0 1 0	0 0 1 1	0 1 1 1	1 1 0 1	1	1 0	RDAE4	---		KRDA RDAE5
11	0 0 1 0 1 1	0 1 0 1	0 0 1 0	0 0 0 0	0	1 0	XMITE	5 ESP		RDA1A/XUPR1
12	0 0 1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	1 0	JIDLE	---		IDLE
13	0 0 1 1 0 1	0 0 1 1	0 0 0 1	0 1 0 1	1	0 0	RDAE5	RØWIØU		KRDA/ENDRDY */JIDLE
14	0 0 1 1 1 0	0 1 0 0	0 0 0 0	0 1 0 0	0	1 0	SS3	---		SETSEL3 WT1
15	0 0 1 1 1 1	0 1 0 0	0 0 0 0	0 1 0 0	1	1 0	SS4	---		SETSEL4 WT1
16	0 1 0 0 0 0	0 1 0 0	0 1 1 1	1 0 0 0	0	1 0	WT1	---		WT2
17	0 1 0 0 0 1	0 1 0 0	1 0 0 0	0 0 0 0	0	1 0	WT2	---		WT3
18	0 1 0 0 1 0	0 1 0 0	1 1 1 1	1 0 0 0	0	1 0	WT3	---		WT4
19	0 1 0 0 1 1	0 0 1 0	1 1 1 1	1 0 0 0	0	1 0	WT4	---		XMITE
20	0 1 0 1 0 0	1 0 1 0	0 0 0 0	0 0 1 0	0	1 0	RDA1A	---		RDAU RDA2
21	0 1 0 1 0 1	0 1 0 1	1 0 1 0	1 0 0 0	0	1 0	XUPR1	XUPREQ		RWX1/YUPR1
22	0 1 0 1 1 0	0 1 1 0	0 1 1 0	0 0 0 0	0	1 0	YUPR1	YUPREQ		RWY1/RWY2
23	0 1 0 1 1 1	1 0 1 1	0 0 1 1	1 0 0 0	0	1 0	RWX1	5 IØX		WUD2/UPX1
24	0 1 1 0 0 0	0 1 1 0	1 1 0 0	0 0 0 0	0	1 0	RWY2	5 IØY		WUD3/ETBV
25	0 1 1 0 0 1	1 1 0 0	1 1 0 0	0 0 0 0	0	1 0	RWY1	5 IØY		WUD4/UPY1
26	0 1 1 0 1 0	0 1 1 0	1 1 0 0	0 0 0 0	0	1 0	WUD3	5 IØY		*/ETBV
27	0 1 1 0 1 1	0 1 1 1	0 1 0 0	1 0 0 1	1	1 0	ETBV	UB04		ETBV RWS2/TRK
28	0 1 1 1 0 0	0 1 1 1	1 1 0 1	0 0 0 0	0	1 0	TRK	UB05		XUPR2/RDA1B
29	0 1 1 1 0 1	1 0 0 1	0 0 0 1	1 0 0 0	0	1 0	RWS2	5 IØSTAT		WUD5/SETST2
30	0 1 1 1 1 0	1 0 1 0	0 0 0 0	0 0 1 0	0	1 0	RDA1B	---		RDAU RDA2
31	0 1 1 1 1 1	1 0 0 0	0 0 1 0	1 0 0 0	0	1 0	XUPR2	XUPREQ		JRWS2A/YUPR2
		502795	502796	502798	502797					

FIGURE 11-8

SERIAL LINK CONTROLLER MICROPROGRAM (CONT'D)

	NØW32	NØW16	NØW08	NØW04	NØW02	NØW01/CDTN	NEXT32	NEXT16	NEXT08	NEXT04	NEXT02	CS08	CS04	CS02	CS01	ØPC8	ØPC4	ØPC2	ØPC1	5ENDRDY	5STØPCLK	LABEL	TEST	RESULTS	OP - TRUE/FALSE DESTINATION
32	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	YUPR2	YUPREQ		JRWS2B/RDA1C
33	1	0	0	0	0	1	0	1	1	1	0	1	1	1	1	0	0	0	0	1	0	JRWS2A	---		RWS2
34	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	RDA1C	---		RDAU RDA2
35	1	0	0	0	1	1	0	1	1	1	0	1	1	1	1	0	0	0	0	1	0	JRWS2B	---		RWS2
36	1	0	0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	1	0	WUD5	5 IØSTAT		*/SETST2
37	1	0	0	1	0	1	1	0	0	1	1	1	0	0	0	1	0	1	0	1	0	SETST2	5 IØY		STAT WUD6/RECV
38	1	0	0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	WUD6	5 IØY		*/RECV
39	1	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	1	0	RECV	---		RECV RDA1A
40	1	0	1	0	0	0	1	0	1	0	0	1	1	1	1	0	1	0	0	1	0	RDA2	---		RDAU RDA3
41	1	0	1	0	0	1	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	RDA3	---		RDAU RDA4
42	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	RDA4	---		RDAU IDLE
43	1	0	1	0	1	1																			
44	1	0	1	1	0	0	1	0	1	1	0	0	1	1	1	0	0	0	0	1	0	WUD2	5 IØX		*/UPX1
45	1	0	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	1	1	0	UPX1	4KHRESX		UPDTX JYUPR1/RWX2
46	1	0	1	1	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	0	RWX2	5 IØX		WUDT/UDX2
47	1	0	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	JYUPR1	---		YUPR1
48	1	1	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	WUD7	5 IØX		*/UDX2
49	1	1	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	1	0	UDX2	---		UPDTX YUPR1
50	1	1	0	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0	WUD4	5 IØY		*/UPY1
51	1	1	0	0	1	1	1	1	0	1	0	1	1	1	0	0	0	1	0	1	0	UPY1	4KHRESY		UPDTY JRWY2/RWY4
52	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	RWY4	5 IØY		WUD8/UDY2
53	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	JRWY2	---		RWY2
54	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	WUD8	5 IØY		*/UDY2
55	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	UDY2	---		UPDTY RWY2
56	1	1	1	0	0	0																			
57	1	1	1	0	0	1																			
58	1	1	1	0	1	0																			
59	1	1	1	0	1	1																			
60	1	1	1	1	0	0																			
61	1	1	1	1	0	1																			
62	1	1	1	1	1	0																			
63	1	1	1	1	1	1																			
							502795				502796				502798				502797						

	ACC	ACW		BCC	BCW
1	ZGND	ZGND	101	#BREAD	9BREAD
2	ZGND	ZGND	102	#BWRITE	9BWRITE
3	ZP05	ZP05	103	#BACK	9BACK
4	ZP05	ZP05	104	#BREADY	4BREADY
5	ZN12	ZN12	105	#BDMAHR	9BDMAHR
6	ZN12	ZN12	106	#BDMAHA	9BDMAHA
7	ZN05	ZN05	107	#BDMAGO	9BDMAGØ
8	ZP12	ZP12	108		
9			109		
10	PTBQ	NTBQ	110		
11	PTHQ	NTHQ	111		
12			112	#BSYNC	9BSYNC
13			113		
14			114		
15			115		
16			116	9RS232R1	9RS232T1
17			117	#RS232R1	#RS232T1
18			118	9TTYR1	9TTYT1
19	#CURS1	9CURS1	119	#TTYR1	#TTYT1
20			120	9LLDR1	9LLDT1
21			121	#LLDR1	#LLDT1
22			122	9RS232R2	9RS232T2
23	9ICPCLR	#ICPCLR	123	#RS232R2	#RS232T2
24			124	9TTYR2	9TTYT2
25			125	#TTYR2	#TTYT2
26			126	9LLDR2	9LLDT2
27	#CURS2	9CURS2	127	#LLDR2	#LLDT2
28			128	9LLDR3	9LLDT3
29			129	#LLDR3	#LLDT3
30			130	9LLDR4	9LLDT4
31	#CURS1	9CURS1	131	#LLDR4	#LLDT4
32			132	9BDATA01	9BDATA00
33			133	9BDATA03	9BDATA02
34			134	9BDATA05	9BDATA04
35	#CURS2	9CURS2	135	9BDATA07	9BDATA06
36			136	9BDATA09	9BDATA08
37			137	9BDATA11	9BDATA10
38			138	9BDATA13	9BDATA12
39	9BPWCLR	#BPWCLR	139	9BDATA15	9BDATA14
40	4KHRESX	#KHRESX	140	4BADDR01	4BADDR00
41	4KHRESY	#KHRESY	141	4BADDR03	4BADDR02
42	4KEYBRD	#KEYBRD	142	4BADDR05	4BADDR04
43	9TVFFF	#TVFFF	143	4BADDR07	4BADDR06
44	9TVBBB	#TVBBB	144	4BADDR09	4BADDR08
45	9THBBB	#THBBB	145	4BADDR11	4BADDR10
46			146	4BADDR13	4BADDR12
47	9THMMM	#THMMM	147	4BADDR15	4BADDR14
48	9TVMMM	#TVMMM	148	ZP05	ZP05
49	PTPQØNE	NTPQØNE	149	ZGND	ZGND
50	PTPQØNE	NTPQØNE	150	ZGND	ZGND

TABLE 11-7 SERIAL LINK BOARD SIGNAL TERM PINOUT

12.0 REPLACEABLE PARTS

This section contains information for ordering replacement parts. Lists of materials for all the major assemblies are included in Volume II of this manual. The lists of material includes the Ramtek part number, and description and quantity.

12.1 ORDERING INFORMATION

To obtain replacement parts, address order inquiry to Ramtek Corporation, 585 N. Mary Avenue, Sunnyvale, California 94086. Identify parts by their Ramtek part numbers listed in the column headed "Part Number." Include equipment model number and serial number.

12.2 RECOMMENDED SPARES

Table 12-1 lists the recommended equipment that may be purchased from Ramtek for depot level repair facilities.

TABLE 12-1 DEPOT LEVEL REPAIR EQUIPMENT

PART NUMBER	DESCRIPTION
MM80-100	In Circuit Emulator
MM80-851	Command Extension ROM
RM-DOC	Diagnostic Option Card
RM-EX	Logic Extender Board
RM-SCK	Spare Component Kit

In Table 12-1, the spare component kit contains a recommended set of space components for an RM-9000 Series System. Table 12-2 lists the components included in the spare component list.

TABLE 12-2 SPARE COMPONENT KIT

PART NUMBER	DESCRIPTION	QUANTITY
1301-		
030	74LS08	2
031	74LS08	2
033	74LS21	2
105	7438	2
108	7430	1
132	74S00	2
133	74LS00	2
134	74S37	2
135	74S20	2
136	74LS20	1
138	74S140	1
140	74H00	2
231	74S32	1
232	74LS32	2
303	7427	1
304	IM5003 (Intersil)	1
305	75182 (TI)	1
306	10124 (TI)	1
307	10125 (TI)	2
309	75183 (TI)	1
311	7414	1
330	74LS04	2
331	74S04	2
332	N8T26 (Signetics)	2
333	74S02	1
336	74LS27	1
337	74S260	2
338	75188 (TI)	1
339	75189 (TI)	1
348	TI-111 (TI)	1
365	74365	2
366	74366	2
404	74123	1
409	74164	2
426	SY2112A-2 (Synertek)	1
430	74S112	2
431	74S174	2
432	74LS174	2
433	74LS194	1
435	74LS109	2
436	74S194	1
444	74S374	1
445	74170	1
450	21044K (Intel)	5
507	74LS153	1
516	74157	2

TABLE 12-2 SPARE COMPONENT KIT (CONT'D)

PART NUMBER	DESCRIPTION	QUANTITY
517	74161	1
531	74S86	2
532	74S138	1
534	74S153	1
535	74S157	1
536	74S161	2
539	74S158	1
541	74LS51	1
543	74S169	2
544	74LS151	2
545	74LS157	2
546	74LS158	1
549	74LS138	2
552	74S168	1

13.0 CIRCUIT DIAGRAMS

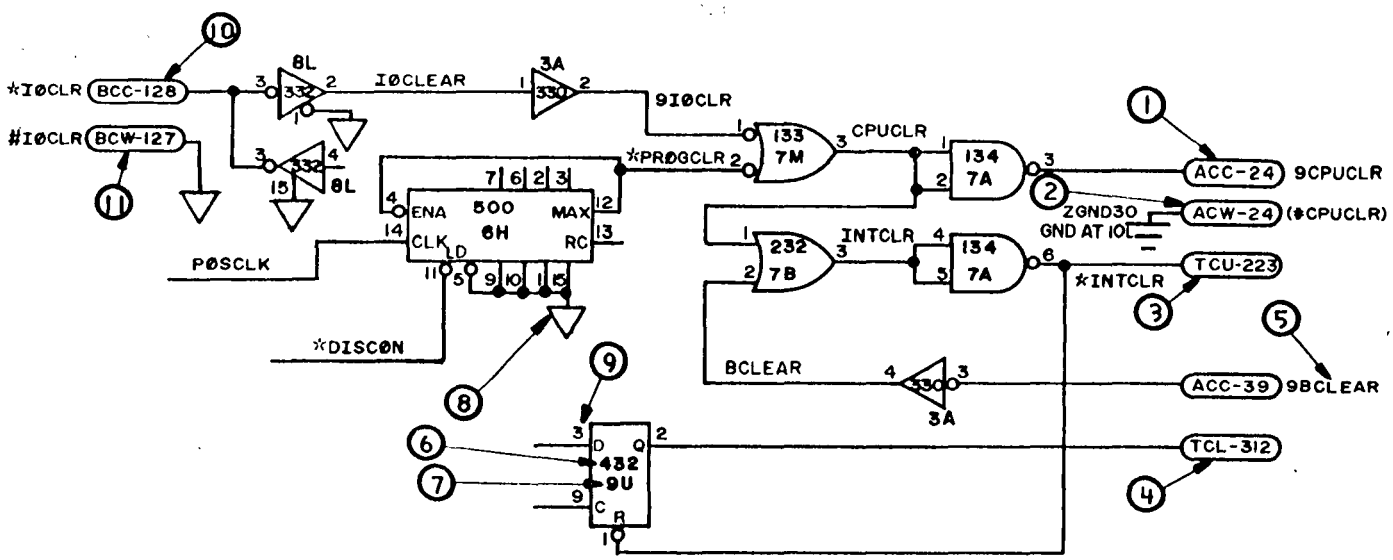
This section contains the circuit/logic diagrams to aid in operation and maintenance of the RM-9000 Series Systems. Detailed schematic diagrams, component location drawings and lists of material for each assembly are contained on the drawings following this section in Volume II. The drawings are bound in numerical order.

General schematic notes and an explanation of the terms and symbols which apply to all the schematic and assembly drawings are shown in Figure 13-1. In general the drawings conform to MIL-STD-806C and ANSI Y32.14. An example of module location designations as they appear on assembly drawings is shown in Figure 13-2.

Components such as resistors, capacitors, transistors and diodes are shown with R_, C_, Q_ and CR_ designators respectively. Logic elements are labeled with a three digit *type* number which are the last three digits of the Ramtek part number for each device. *Type* numbers, in general, have been assigned as follows:

000 - 099 AND gates
100 - 199 NAND gates
200 - 299 OR gates
300 - 399 NOR gates, inverters drivers and receivers
400 - 499 Flip-flops and memories
500 - 599 LSI and MSI components
600 - 699 Microprocessors
700 - 799 ECL components

A detailed description of each type number is shown in Table 13-1. If further information is desired, reference should be made to the specified manufacturer of a particular device.



LEGEND:

(1) "A" connector component side, pin number.

(2) "A" connector non-component side, pin number.

(3) Test connector upper, pin number.

(4) Test connector lower, pin number.

(5) Term name.

(6) Device type number, refer to Table 13-1.

(7) Device location, refer to Figure 13-2.

(8) Signal ground.

(9) Pin number on device.

(10) "B" connector component side, pin number.

(11) "B" connector non-component side, pin number.

FIGURE 13-1 SCHEMATIC NOTES

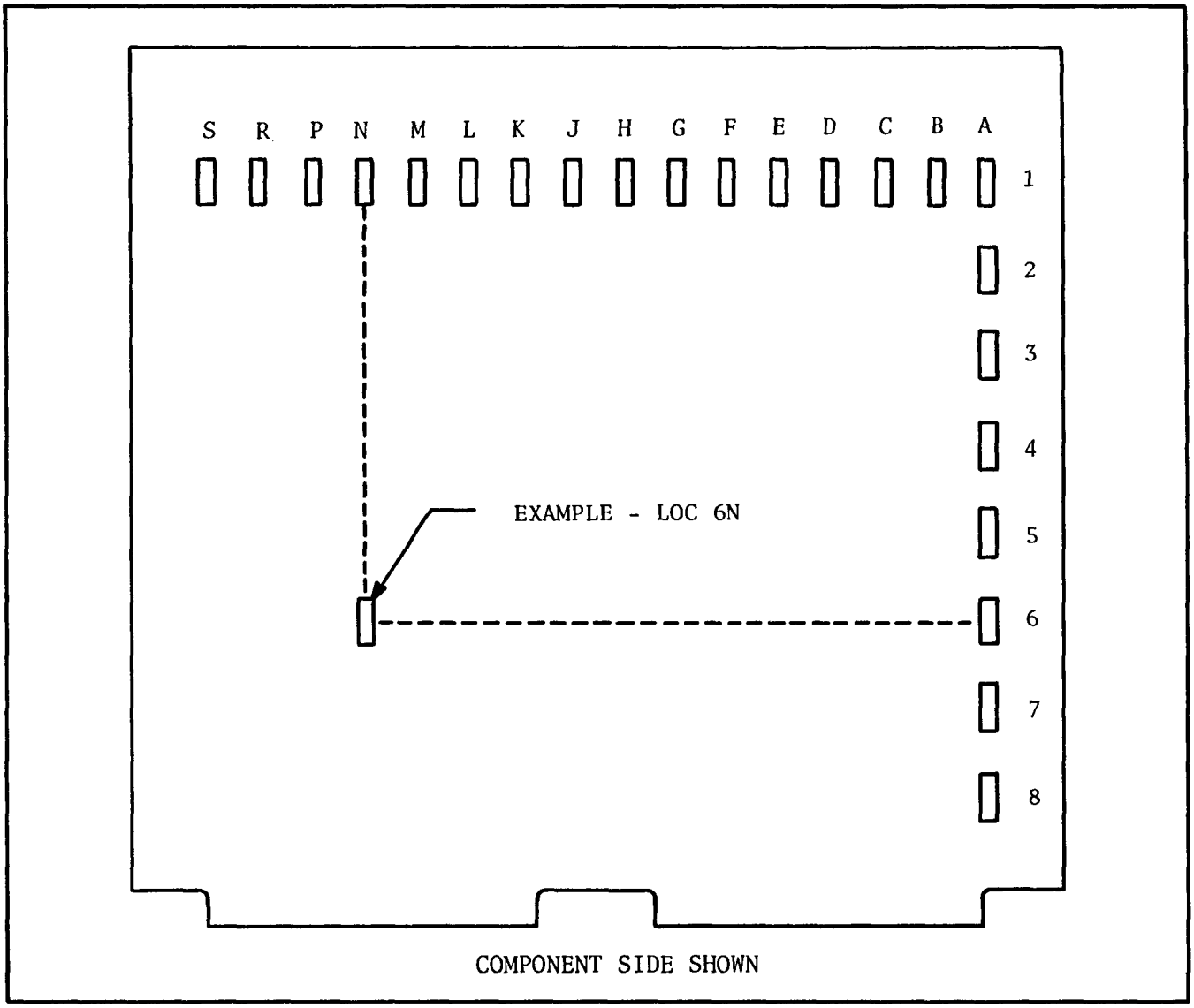


FIGURE 13-2 TYPICAL DEVICE LOCATION DESIGNATOR

TABLE 13-1 NUMERICAL LISTING OF IC TYPES

REFERENCE	RAMTEK PART NO.	DESCRIPTION	MFR.	MFR. PART NO.
001	1301001	IC: Dual 2 input AND gate open collector driver	01295	SN75451B
002	1301002	IC: Quad 2 input AND gate	01295	SN7408N
024	1301024	IC: Dual 4 input AND gate	01295	SN74H21N
030	1301030	IC: Quad 2 input AND gate	01295	SN74S08N
031	1301031	IC: Quad 2 input AND gate	01295	SN74LS08N
032	1301032	IC: Dual 4 input AND gate	01295	SN74S21N
033	1301033	IC: Dual 4 input AND gate	01295	SN74LS21N
034	1301034	IC: Quad 2 input AND gate open collector	01295	SN74LS09N
100	1301100	IC: Quad 2 input NAND gate open collector	01295	SN7401N
101	1301101	IC: Quad 2 input NAND gate driver	01295	SN7437N
102	1301102	IC: Quad 2 input NAND gate	01295	SN7400N
103	1301103	IC: Triple 3 input NAND gate	01295	SN7412N
104	1301104	IC: Dual 4 input NAND gate	01295	SN7420N
105	1301105	IC: Quad 2 input NAND gate driver (open collector)	01295	SN7438N
106	1301106	IC: Quad 2 input NAND gate driver	01295	SN7439N
107	1301107	IC: Dual peripheral positive NAND gate driver (open collector)	01295	SN75452B
108	1301108	IC: 8 input NAND gate	01295	SN7430N
112	1301112	IC: Quad 2 input NAND gate open collector	01295	SN7403N
132	1301132	IC: Quad 2 input NAND gate	01295	SN74S00N
133	1301133	IC: Quad 2 input NAND gate	01295	SN74LS00N
134	1301134	IC: Quad 2 input NAND gate driver	01295	SN74S37N
135	1301135	IC: Dual 4 input NAND gate	01295	SN74S20N
136	1301136	IC: Dual 4 input NAND gate	01295	SN74LS20N
137	1301137	IC: 12 input NAND gate with tristate control	01295	SN74S134N
138	1301138	IC: Dual 2 input 50 ohm NAND gate driver	01295	SN74S140N
139	1301139	IC: Triple 3 input NAND gate	01295	SN74H10N
140	1301140	IC: Quad 2 input NAND gate	01295	SN74H00N
141	1301141	IC: Dual 4 input NAND gate	01295	SN74H20N
142	1301142	IC: Quad 2 input NAND Schmitt triggers	01295	SN74LS132N
143	1301143	IC: Quad 2 input NAND (open collector)	01295	SN74LS03N
200	1301200	IC: Dual 2 input OR gate peripheral driver	01295	SN75453B
231	1301231	IC: Quad 2 input OR gate	01295	SN74S32N
232	1301232	IC: Quad 2 input OR gate	01295	SN74LS32N
300	1301300	IC: Hex inverter	01295	SN7404N
301	1301301	IC: Hex inverter	01295	SN74L04N
302	1301302	IC: Quad 2 input NOR gate	01295	SN7402N
303	1301303	IC: Triple 3 input NOR gate	01295	SN7427N
304	1301304	IC: Quad MOS clock driver	32293	IM5003
305	1301305	IC: Differential line receiver	01295	SN75182N
306	1301306	IC: TTL to ECL translator	01295	SN10124N
307	1301307	IC: ECL to TTL translator	01295	SN10125N
308	1301308	IC: Differential line receiver	01295	SN75107N
309	1301309	IC: Differential line driver	01295	SN75183N

TABLE 13-1 NUMERICAL LISTING OF IC TYPES (CONT'D)

REFERENCE	RAMTEK PART NO.	DESCRIPTION	MFR.	MFR. PART NO.
310	1301310	IC: Hex inverter open collector	01295	SN7405N
311	1301311	IC: Hex inverting Schmitt trigger	01295	SN7414N
312	1301312	IC: Dual line driver	18324	8T13
313	1301313	IC: Differential line driver	01295	SN75110N
314	1301314	IC: Hex inverter open collector driver	01295	SN7406N
315	1301315	IC: Hex inverter open collector, high voltage	01295	SN7416N
316	1301316	IC: Hex inverter	01295	SN74H04N
317	1301317	IC: Dual 2 input NOR gate peripheral driver	01295	SN75454B
318	1301318	IC: Video operational amplifier/driver	27014	LH0002C
319	1301319	IC: MOS clock driver TO-8 package	27014	MH0026CG
320	1301320	IC: Dual differential line driver	18324	DM8830
321	1301321	IC: Voltage follower 14 pin DIP	27014	LM110D
322	1301322	IC: Dual line driver	01295	SN75150N
323	1301323	IC: Operational amplifier	27014	LM301N
324	1301324	IC: Voltage regulator	27014	LM305N
325	1301325	IC: 5 volt regulator	27014	LM309N
326	1301326	IC: Voltage follower	27014	LM310N
327	1301327	IC: Audio power amplifier	27014	LM380N
328	1301328	IC: Quad amplifier	27014	LM3900N
329	1301329	IC: 5 volt regulator TO-0 package	27104	LM309H
330	1301330	IC: Hex inverter	01295	SN74LS04N
331	1301331	IC: Hex inverter	01295	SN74S04N
332	1301332	IC: Tristate line driver/receiver	18324	8T26
333	1301333	IC: Quad 2 input NOR gate	01295	SN74S02N
334	1301334	IC: Quad 2 input NOR gate	01295	SN74LS02N
335	1301335	IC: Triple 3 input NOR gate	01295	SN74S27N
336	1301336	IC: Triple 3 input NOR gate	01295	SN74LS27N
337	1301337	IC: Dual 5 input NOR gate	01295	SN74S26N
338	1301338	IC: RS232 line driver	01295	SN75188N
339	1301339	IC: RS232 line receiver	01295	SN75189N
340	1301340	IC: High speed hex inverter	18324	8H90A
341	1301341	IC: Noninverting hex buffer, high voltage (open collector)	01295	SN7417N
342	1301342	IC: Quad comparator	18324	LM139N
343	1301343	IC: Precision operational amplifier	27014	LM30
350	1301350	IC: Dual MOS clock driver	27014	DS0026CN
365	1301365	IC: Hex non-inverting bus drivers (tri-state)	01295	SN74365N
366	1301366	IC: Hex inverting bus drivers (tri-state)	01295	SN74366N
367	1301367	IC: Hex non-inverting bus drivers (tri-state)	01295	SN74367N
368	1301368	IC: Hex inverting bus drivers (tri-state)	01295	SN74368N
369	1301369	IC: Hex non-inverting bus drivers (tri-state)	01295	SN74LS365N
400	1301400	IC: Quad 80 bit shift register	34649	3409
401	1301401	IC: 256 x 1 bipolar RAM	01295	SN74S200N

TABLE 13-1 NUMERICAL LISTING OF IC TYPES (CONT'D)

REFERENCE	RAMTEK PART NO.	DESCRIPTION	MFR.	MFR. PART NO.
402	1301402	IC: Quad 256 bit MOS shift register	34335	2802DC
403	1301403	IC: 1k x 1 bit shift register	32293	7722
404	1301404	IC: Dual one shot	01295	SN74123N
405	1301405	IC: 4 bit latch	07263	9314
406	1301406	IC: 16 x 4 bit bipolar RAM	01295	SN7489N
407	1301407	IC: Dual 480 bit shift register	31471	51685
408	1301408	IC: 8 bit parallel in-serial out shift register	01295	SN74164N
409	1301409	IC: 8 bit serial in-parallel out shift register	01295	SN74166N
410	1301410	IC: Hex D flip-flop	01295	SN74174N
411	1301411	IC: 4 bit left-right shift register	01295	SN74194N
412	1301412	IC: 256 x 1 MOS RAM	18324	2501
413	1301413	IC: 1024 bit MOS shift register	18324	2525
414	1301414	IC: 1024 x 1 bit MOS dynamic RAM	50364	MM5260
415	1301415	IC: 256 bit bipolar RAM	32293	T5523
416	1301416	IC: 1024 bit dynamic shift register	18324	2512
417	1301417	IC: 4 bit shift register with JK inputs	01295	SN74195N
418	1301418	IC: Dual one shot	07263	9602
419	1301419	IC: Dual D flip-flop	01295	SN74H74N
424	1301424	IC: 64 x 4 RAM	01295	SN74S189N
426	1301426	IC: 256 x 4 static RAM	34335	AM91L12C
430	1301430	IC: Dual JK flip-flop	01295	SN74S112N
431	1301431	IC: Hex D flip-flop	01295	SN74S174N
432	1301432	IC: Hex D flip-flop	01295	SN74LS174N
433	1301433	IC: Left-right shift register	01295	SN74LS194N
434	1301434	IC: Dual D flip-flop	01295	SN74S74N
435	1301435	IC: Dual JK flip-flop	01295	SN74LS109N
436	1301436	IC: Left-right shift register	01295	SN74S194N
437	1301437	IC: Dual JK flip-flop	01295	SN74LS112N
438	1301438	IC: 8 bit serial in-parallel out shift	01295	SN74LS166N
439	1301439	IC: Dual JK flip-flop	01295	SN74S109N
440	1301440	IC: 4 bit D-type register (tri-state)	01295	SN74173N
441	1301441	IC: 1k x 1 static RAM	32293	IM55S18CJE
442	1301442	IC: 256 x 1 RAM	34335	AM27LS00PC
443	1301443	IC: 1k x 1 RAM	07263	93425AXC
444	1301444	IC: 8 bit D-type latch	01295	SN74S374
445	1301445	IC: 4 bit register file	01295	SN74170N
446	1301446	IC: 256 x 4 static RAM	34335	AM9112
447	1301447	IC: 256 x 4 bi-polar RAM (separate I/O)	07263	93L422
448	1301448	IC: Dual JK flip-flop	01295	SN74LS107N
449	1301449	IC: Mono stable multivibrator	01295	SN74LS123N
450	1301450	IC: 4k x 1 dynamic RAM	07263	40965DC
451	1301451	IC: Dual D type flip-flop	01295	SN74LS74N
452	1301452	IC: 256 x 4 static RAM	34649	8111-2
453	1301453	IC: 1k x 1 static RAM	34335	AM9102EPC
454	1301454	IC: 8 bit SI-PO shift register	01295	SN74LS164N
500	1301500	IC: 4 bit up-down counter	01295	SN74191N

TABLE 13-1 NUMERICAL LISTING OF IC TYPES (CONT'D)

REFERENCE	RAMTEK PART NO.	DESCRIPTION	MFR.	MFR. PART NO.
501	1301501	IC: 4 bit adder	01295	SN74283N
502	1301502	IC: 9 line to BCD encoder	01295	SN74147N
503	1301503	IC: Quad 2 input multiplexer with storage	01295	SN74298N
504	1301504	IC: UART	52840	TR1602A
505	1301505	IC: 4 to 10 line decoder	01295	SN7442N
506	1301506	IC: 4 bit binary up-down counter	01295	SN74LS193N
507	1301507	IC: 2 section - 4 input selector	01295	SN74LS153N
508	1301508	IC: 4 bit binary adder (not recommended for use dur to pwr, gnd arrangement)	01295	SN7483N
509	1301509	IC: 4 bit up-down binary counter	01295	SN74193N
510	1301510	IC: 4 bit up-down BCD counter	01295	SN74192N
511	1301511	IC: BCD to 7 segment decoder/driver	01295	SN7447N
513	1301513	IC: BCD counter	07263	93L16
514	1301514	IC: 8 to 1 line selector	01295	SN74151N
515	1301515	IC: 2 section - 4 input selector	01295	SN74153N
516	1301516	IC: Quad 2-line-to-1-line selector	01295	SN74157N
517	1301517	IC: 4 bit binary counter	07263	9316
518	1301518	IC: 4 bit comparator	01295	SN7485N
519	1301519	IC: Quad 2 input exclusive OR gate	01295	SN7486N
520	1301520	IC: 4 AND/NOR gate	01295	SN74H55N
521	1301521	IC: Keyboard encoder ROM	11711	AY-5-3600-XXX
522	1301522	IC: Baud rate clock generator, MOS	27014	MM5307AA/N
523	1301523	IC: Dual voltage controlled oscillators	01295	SN74S124N
524	1301524	IC: Phase frequency detector	04713	MC0444P
525	1301525	IC: Timer	18324	NE555Y
526	1301526	IC: Timer	18324	NE556A
527	1301527	IC: Timer	18324	NE565A
528	1301528	IC: 4 bit binary counter	01295	SN74LS93N
529	1301529	IC: Look ahead carry generator	01295	SN74S182N
530	1301530	IC: 4 bit comparator	01295	SN7485N
531	1301531	IC: Quad exclusive OR gate	01295	SN74S86N
532	1301532	IC: 3 to 8 decoder/demultiplexer	01295	SN74S138N
533	1301533	IC: 8 input selector	01295	SN74S15N
534	1301534	IC: 2 section - 4 input selector	01295	SN74S153N
535	1301535	IC: 4 section - 2 input selector	01295	SN74S157N
536	1301536	IC: 4 bit binary counter	01295	SN74S161N
537	1301537	IC: 3 to 8 decoder/demultiplexer	01295	SN74138N
538	1301538	IC: 5 x 7 font ASCII character generator	18324	2513N/CM2140
539	1301539	IC: 4 section - 2 input selector	01295	SN74S158N
540	1301540	IC: 4 AND NOR gate	01295	SN74S64N
541	1301541	IC: Dual 2 AND/NOR gate	01295	SN74LS51N
542	1301542	IC: Arithmetic logic unit/function generator	01295	SN74S181N
543	1301543	IC: Synchronous 4 bit up-down counter	01295	SN74S169N
544	1301544	IC: 8 input selector	01295	SN74LS151N
545	1301545	IC: 4 section - 2 input selector	01295	SN74LS157N
546	1301546	IC: 4 section - 2 input selector	01295	SN74LS158N
547	1301548	IC: 4 bit binary counter	01295	SN74LS161N

TABLE 13-1 NUMERICAL LISTING OF IC TYPES (CONT'D)

REFERENCE	RAMTEK PART NO.	DESCRIPTION	MFR.	MFR. PART NO.
548	1301548	IC: Quad 2 input multiplexer with D storage	01295	SN74LS298N
549	1301549	IC: 3-to-8 decoder/demultiplexer	01295	SN74LS138N
550	1301550	IC: Decode counter	18324	N7490
551	1301551	IC: BCD to 7 segment decoder/driver	01295	SN7448N
552	1301552	IC: Synchronous up/down decode counter	01295	SN74S168N
554	1301554	IC: Synchronous up/down binary counter	01295	SN74S169N
555	1301555	IC: Arithmetic logic unit	01295	SN74181N
556	1301556	IC: 1 of 16 decoder/demultiplexer	01295	SN74150N
558	1301558	IC: Synchronous 4 bit binary counter	01295	SN74163N
559	1301559	IC: Synchronous 4 bit decode counter	01295	SN74160N
560	1301560	IC: Quad 2 to 1 selector/multiplexer	01295	SN74S257N
561	1301561	IC: Dual 4 to 1 selector/multiplexer (open collector)	01295	SN74S253N
562	1301562	IC: Dual 2 to 4 decoder/demultiplexer	01295	SN74S139
536	1301563	IC: 8 bit I/O port	34649	8212
564	1301564	IC: Priority encoder	01295	SN74148N
565	1301565	IC: Programmable communication interface (USARI)	34649	8251
566	1301566	IC: Programmable bit rate generator	07263	F4702
600	1301600	IC: 8080 microprocessor (325 ns cycle time)	34335	9080A-1
602	1301602	IC: 8080 microprocessor (480 ns cycle time)	34335	8080A
603	1301603	IC: Z80 microprocessor (250 ns cycle time)		MK3880-44
608	1301608	IC: Bit slice bi-polar microprocessor	34335	AM2901
609	1301609	IC: Microprocessor sequencer	34335	AM2909
700	1301700	IC: ECL JK flip-flop	04713	MC10231P

RM-9000

Programming Manual

Part No. 503160

Masters

(Return to Publications Dept. Only)

#

97 2-sided } 194
16 1-sided } 16
210 Photo. direct

1-foil plate (Neg supplied)

WmH.MULLER PRINTING CO.

200 copies
GBC bind
collated

3550 Thomas Road • Santa Clara, California 95050 • Telephone (408) 246-8400

RM-9000 Programming Manual

FRONT	Cover	Title	i	iii	v	vii	ix	Blue Section I	1-1	1-3	1-5
BACK		Notice	ii	iv	vi	viii	X	X	1-2	1-4	1-6

FRONT	1-7	Section II	2-8	2-10	2-12	2-14	Blue Section III	3-15	3-17	3-19
BACK	X	X	2-9	2-11	2-13	X	X	3-16	3-18	3-20

FRONT	3-21	3-22A	3-24	26	28	30	32	34	36	38
BACK	3-22	3-23	3-25	27	29	31	33	35	37	39

FRONT	40	42	44	46	48	50	52	54	56	58
BACK	41	43	45	47	49	51	53	55	57	59 ✓

FRONT	60	62	64	66	68	70	72	74	76	78
BACK	61	63	65	67	69	71	73	75	77	79

FRONT	80	82	84	86	3-88	88B	89	91	93	95
BACK	81	83	85	87	88A	88C	90	92	94	96

FRONT	97	99	101	103	105	107	109	111	113	115
BACK	98	100	102	104	106	108	110	112	114	116

WmH.MULLER PRINTING CO.

3550 Thomas Road • Santa Clara, California 95050 • Telephone (408) 246-8400

FRONT	117	119	121	123	125	127	128A	130	132	134
BACK	118	120	122	124	126	128	129	131	133	135

FRONT	136	138	139A	SECTION IV	4-140	Blue SECTION V	5-142	144	146	148
BACK	137	139W	139B	X	141	X	143	145	147	149

FRONT	150	152	Blue APPENDIX A	A-1	A-3	Blue APPENDIX B	B-1	B-3	B-5	B-7
BACK	151	X	X	A-2	X	X	B-2	B-4	B-6	B-8

FRONT	B-9	B-11	B-13	Blue APPENDIX C	C-1	C-3	C-5	C-7	C-9	C-11
BACK	B-10	B-12	B-14	X	C-2	C-4	C-6	C-8	C-10	[redacted] C-12

FRONT	Neg. P. C-13	Blue APPENDIX D	D-1							
BACK	X	X	X							

FRONT										
BACK										

FRONT										
BACK										

ramtek

RM-9000 SERIES
GRAPHIC DISPLAY SYSTEM

PROGRAMMING MANUAL

Part No. 503160C

APRIL 1981

RAMTEK CORPORATION
2211 Lawson Lane
Santa Clara, California 95051

Ramtek

NOTICE

Ramtek Corporation has prepared this manual for use by Ramtek personnel, licensees and customers. The information contained herein is the property of Ramtek and shall neither be reproduced in whole or in part without Ramtek prior written approval.

Ramtek reserves the right to make changes without notice in the specifications and materials contained herein, and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

Ramtek

TABLE OF CONTENTS

PAGE

SECTION I - INTRODUCTION

1-0 INTRODUCTION TO DISPLAY CONCEPTS 1-1

1-1 SYSTEM OVERVIEW 1-2

1-2 SCREEN COORDINATE SYSTEM 1-3

1-3 DISPLAY DATA TYPES 1-4

1-4 Image Data 1-4
1-5 Text Data 1-5
1-6 Raster Data 1-5
1-7 Graphics Data 1-6

1-8 DISPLAY INTERACTION 1-6

1-9 Cursors/Joysticks/Trackballs 1-6
1-10 Keyboards 1-7

SECTION II - FUNCTIONAL DESCRIPTION

2-0 RM-9000 FUNCTIONAL DESCRIPTION 2-8

2-1 COMPUTER INTERFACE 2-9

2-2 CONTROL BOARD 2-9

2-3 8080 Microprocessor 2-10
2-4 Display Generator 2-10

2-5 REFRESH MEMORY 2-10

2-6 VIDEO GENERATOR 2-11

2-7 MEMORY EXPANSION BOARD (RM-MOC) 2-11

TABLE OF CONTENTS (Continued)

		PAGE
2-8	SERIAL LINK BOARD (RM-SLC)	2-14
	SECTION III - RM-9000 INSTRUCTION SET	
3-0	INTRODUCTION	3-15
3-1	INSTRUCTIONS	3-15
3-2	INSTRUCTION FORMAT	3-17
3-3	RM-9000 DATA MODES	3-17
3-4	NORMAL INSTRUCTION FORMAT	3-19
3-5	Parameter Byte	3-20
3-6	Data Flag (Bit 0)	3-20
3-7	Operand Flag (Bit 1)	3-20
3-8	Reverse Packing Flag (Bit 3)	3-20
3-9	Background Flag (Bit 4)	3-21
3-10	Additive Flag (Bit 5)	3-21
3-11	Addressing Mode Flags (Bit 6, 7)	3-21
3-12	Operand Flag Word	3-23
3-13	Parameter Operands	3-23
3-14	Subchannel Parameter (No. 0)	3-25
3-15	Foreground Parameter (No. 1)	3-25
3-16	Background Parameter (No. 2)	3-26
3-17	Index 1 Parameter (No. 3)	3-26
3-18	Index 2 Parameter (No. 4)	3-27
3-19	Origin Parameter (No. 5)	3-28
3-20	Window Parameter (No. 6)	3-29
3-21	Scan Parameter (No. 7)	3-30
3-22	Dimension Parameter (No. 8)	3-33
3-23	Spacing Parameter (No. 9)	3-34
3-24	Scale Parameter (No. 10)	3-36
3-25	Function Parameter (No. 11)	3-37
3-26	Conic Equation (Parameter (No. 12) . . .	3-43
3-27	Baseline Parameter (No. 13)	3-44

Framtek

TABLE OF CONTENTS (Continued)

		PAGE
3-28	Scroll-Count Parameter (No. 14)	3-44
3-29	Start-Point Argument (No. 15)	3-45
3-30	Data Length Word	3-46
3-31	Normal-Format Instruction Data	3-46
3-32	Normal-Format Standard Instructions	3-47
3-33	No-Operation Instruction (INOP)	3-48
3-34	Set Parameter Instruction (SET)	3-50
3-35	Erase Instruction (ERS)	3-52
3-36	Write Image Instruction (WI)	3-54
3-37	Read Image Instruction (RI)	3-60
3-38	Write Text Instruction (WT)	3-63
3-39	Special-Format Standard Instructions	3-67
3-40	Load Hard Register Instruction (LOAD)	3-68
3-41	Read Soft Register Instruction (READ)	3-74
3-42	Load Auxiliary Memory Instruction (LAM)	3-76
3-43	Read Auxiliary Memory Instruction (RAM)	3-78
3-44	Reset Instruction (RSET)	3-80
3-45	Initialize Instruction (INIT)	3-83
3-46	OPTIONS FIRMWARE INSTRUCTIONS	3-84
3-47	RM-GRA Graphics Option Instructions	3-85
3-48	Write Vector Instruction (WV)	3-86
3-49	Write Plot Instruction (WP)	3-89
3-50	Write Raster Instruction (WR)	3-93
3-51	RM-CON Conics Option Instruction	3-97
3-52	WRITE CONIC Instruction (WC)	3-98
3-53	RM-SCR Scroll Option Instructions	3-102
3-54	SCROLL X Instruction (SCRX)	3-103
3-55	SCROLL Y Instruction (SCRY)	3-106
3-56	RM-STA Status Management Option Instructions	3-109
3-57	SAVE ENVIRONMENT Instruction (PUSHE)	3-110
3-58	RESTORE ENVIRONMENT Instruction (POPE)	3-111
3-59	RM-FNT Programmable Font Option Instructions	3-112
3-60	LOAD PROGRAMMABLE FONT Instructions (LPF)	3-113
3-61	LOAD PROGRAMMABLE FONT WITH REVERSE PACKING Instructions (LPFRP)	3-115
3-62	RM-PER Interactive Peripherals Option Instructions	3-116

TABLE OF CONTENTS (Continued)

			PAGE
	3-63	Write Cursor State Instruction (WCS) . . .	3-117
	3-64	Read Cursor Status Instruction (RCS) . . .	3-119
	3-65	Write Keyboard Instruction (WKB)	3-121
	3-66	Read Keyboard Instruction (RKB)	3-123
	3-67	Sense Peripheral Status Instruction (SPS) . .	3-125
3-68		RM-USR User Subroutine Option Instructions . . .	3-127
	3-69	Load Control Memory Instruction (LCM) . .	3-129
	3-70	Load Control Memory With Reverse Packing Instruction (LCMRP)	3-131
	3-71	Read Control Memory Instruction (RCM) . .	3-133
	3-72	Read Control Memory With Reverse Packing Instruction (RCMRP)	3-135
	3-73	Call Control Memory Instruction (CCM) . .	3-137
	3-74	Execute Instruction Memory Instruction (XIM)	3-138
	3-75	RM-MAG Magnify Instruction (MAGNFY) . .	3-139

SECTION IV - INTERRUPT PROCESSING

4-0		INTERRUPT OPERATIONS	4-140
	4-1	ILLEGAL INSTRUCTION INTERRUPT	4-140
	4-2	CURSOR INTERRUPT	4-140
	4-3	RECEIVER INTERRUPT	4-140
	4-4	TRANSMITTER INTERRUPT	4-141

SECTION V - RM-9000 INSTRUCTION TIMING

5-0		RM-9000 INSTRUCTION TIMING	5-142
	5-1	Standard Instruction Set Execution Times . . .	5-142
	5-2	Parameter Operand Processing Time . . .	5-142
	5-3	INOP Instruction Time	5-142
	5-4	Set Instruction Timing	5-144
	5-5	Erase Instruction Timing	5-144

TABLE OF CONTENTS (Continued)

			PAGE
	5-6	Write Image Instruction Timing	5-144
	5-7	Read Image Instruction Timing	5-145
	5-8	Write Text Instruction Timing	5-145
	5-9	Load Hard Register Instruction Timing	5-145
	5-10	Read Soft Register Instruction Timing	5-145
	5-11	Load Auxiliary Memory Instruction Timing	5-146
	5-12	Read Auxiliary Memory Instruction Timing	5-146
	5-13	Reset Instruction Timing	5-146
	5-14	Initialize Instruction Timing	5-147
5-15	Options Instruction Set Execution Times		5-147
	5-16	Write Vector Instruction Timing	5-147
	5-17	Write Plot Instruction Timings	5-148
	5-18	Write Raster Instruction Timing	5-148
	5-19	Write Conic Instruction Timing	5-148
	5-20	Scroll X Instruction Timing	5-149
	5-21	Scroll Y Instruction Timing	5-149
	5-22	Save Environment Instruction Timing	5-150
	5-23	Restore Environment Instruction Timing	5-150
	5-24	Load Programmable Font Instruction Timing	5-150
	5-25	Load Programmable Font Reverse Packing Instruction Timing	5-150
	5-26	Write Cursor Instruction Timing	5-150
	5-27	Read Cursor Instruction Timing	5-150
	5-28	Write Keyboard Instruction Timing	5-151
	5-29	Read Keyboard Instruction Timing	5-151
	5-30	Sense Peripheral Status Instruction Timing	5-151
	5-31	Load Control Memory Instruction Timing	5-151
	5-32	Load Control Memory Reverse Instruction Timing	5-151
	5-33	Read Control Memory Instruction Timing	5-151
	5-34	Read Control Memory Reverse Packing Instruction Timing	5-151
	5-35	Call Control Memory Reverse Packing Instruction Timing	5-152
	5-36	Execute Instruction Memory Instruction Timing	5-152

TABLE OF CONTENTS (Continued)

APPENDIX A

PAGE

TYPE II VIDEO LOOK-UP TABLE
FUNCTIONAL DESCRIPTION

A-0 INTRODUCTION A-1

A-1 TYPE II VIDEO LOOK-UP TABLE A-1

A-2 TYPE II VIDEO CONFIGURATIONS A-2

A-3 TYPE II VLT PARTITIONING A-3

APPENDIX B

GENERATION OF CONICS

B-0 CONICS B-1

B-1 THE ELLIPSE B-4

B-2 Drawing an Ellipse B-5
B-3 Translating The Ellipse B-7
B-4 Rotating The Ellipse B-7
B-5 Rotating About the Center of the Ellipse B-9

B-6 THE HYPERBOLA B-9

B-7 Translating And Rotating The Hyperbola B-12

B-8 FINAL POINTS TO CONSIDER B-13

APPENDIX C

INTERACTIVE PERIPHERALS
OPERATING PROCEDURES

C-0 INTRODUCTION TO INTERACTIVE PERIPHERALS C-1

C-1 CURSOR DESCRIPTION C-1

TABLE OF CONTENTS (Continued)

		PAGE
C-2	Standard Cursor Pattern	C-1
C-3	Cursor Screen Addressing	C-3
C-4	JOYSTICK CURSOR CONTROLLER GC-106	C-3
C-5	JOYSTICK STATUS CONTROL SWITCHES	C-5
C-6	Joystick Cursor Selection Switches	C-5
C-7	TRACKBALL CURSOR CONTROLLER - GC-104	C-6
C-8	PROCESS CONTROL KEYBOARD - GK-120	C-6

APPENDIX D

RM-9000 STANDARD CHARACTER FONTS

TABLE OF CONTENTS (Continued)

	FIGURE TITLE	PAGE
1-1	Classical X-Y-Z Coordinate System	1-3
2-1	RM-9100, 9200 & 9300 Functional Block Diagram	2-8
2-2	RM-V2 Functional Block Diagram	2-12
2-3	RM-V1 Functional Block Diagram	2-13
3-1	Normal-Format Instruction Layout	3-16
3-2	Screen Coordinate Orientation	3-23
3-3	Operand Flag Word Format	3-24
3-4	Window Definition	3-30
3-5	Normal Spacing Values	3-35
3-6	Imaging Scan Example	3-59
3-7	RM-9000 Control Board	3-82
3-8	Programmable Font Example	3-114
A-1	Lam Instruction Format	A-1
C-1	Cursor Configuration	C-2
C-2	Joystick Model GC-106	C-13
C-3	Keyboard Model GK-120	C-13
D-1	Standard Text Character Fonts	D-1

TABLE OF CONTENTS (Continued)

	TABLE TITLE	PAGE
3-1	RM-9000 Instruction Repertoire	3-18
3-2	RM-9000 Instruction Set	3-19
3-3	Functional Relationship of Data Background Flag & Additive-Right Flag	3-22
3-3A	Attribute Summary of 4 Addressing Modes	3-22A
3-4	Systems Resolution Definitions & Origin Default Values	3-29
3-5	COP Placement After Window Setting	3-31
3-6	Image & Raster Mode Scan Directions	3-32
3-7	Write Text Scan Directions	3-32
3-8	Window Origin	3-33
3-9	Scaled Picture Ratio	3-37
3-10	Logical/Arithmetic Function Codes	3-39
3-11	I/O Device Register Address Map (RM-9000 Series)	3-70
3-12	Read Soft Register Addresses	3-75
3-13	Video Look-Up Table Addressing	3-77
3-14	Operand Parameter Default Values	3-81
3-15	Additive (AD), Background (BK) & Data Bit Relation	3-94
3-16	RM-9000 Internal Memory Map	3-128
3-17	Control Board (502450) Dip Switch Settings	3-128A
5-1	Normal-Format Parameter Operand Processing Times	5-143
C-1	Joystick/Cursor Movement	C-4
C-2	GK-120 TTY Mode Control	C-10

Framtek

SECTION I

INTRODUCTION

1-0 INTRODUCTION TO DISPLAY CONCEPTS

Some rudimentry knowledge of basic display principals is essential to understanding the RM-9000 display system. It must be understood that the image viewed upon a cathode ray tube is produced by a maneuverable, self-contained electron gun (or guns) which emits a beam of specified intensity in order to excite phosphors painted upon the inner face of the tube. Due to the short persistence of the phosphors, the image must be continually repeated (refreshed) in order to minimize phosphor decay therefore visible flicker. The electron beam either directly traces the image being produced or indirectly scans the entire screen while being turned ON and OFF at predetermined times. This second technique is known as raster scan and is employed by the RM-9000 as well as the broadcast television industry. Although somewhat limiting the resolution of the image, the raster scan technique does not constrain the image to a determined number of vectors as does the beam steering technique, nor does it preclude the implementation of the full color spectrum. Since the screen is reduced to a finite matrix of dots, phosphors representing the three primary colors (red, green and blue) can be painting into each dot area (triad) and separate electron guns can be energized to excite each of the phosphors. The relative excitation of the phosphors determines the emitted color.

Because the CRT cannot remember its image, the RM-9000 stores this information within its self-contained, solid-state refresh memory system. The image is stored in binary and is accessed by the video generator at the television raster rate. One or more bits of information describe each picture element (dot), i.e., three bits might be used to describe the discrete excitation of each of the primary phosphors, thus giving seven colors plus black. Additional colors (or grey-scale) are achieved by way of additional memory and digital-to-analog conversion of a prescribed number of bits per picture element, i.e., two bits provides for three intensity levels plus black. Pseudo color is achieved by way of a random access table which is indexed by the binary weighted value of each element as described within the refresh memory. This feature provides for the arbitrary equation of desired color (or grey scale level) and any particular refresh memory bit pattern.

It is the function of the RM-9000 display system to interpret data from the host computer in a specified binary format and compose an image in refresh memory. The internal structure of the RM-9000 allows the refresh memory to be loaded by the internal processor at the same time that it is being accessed by the video generator hardware. Since the refresh memory is truly random-access in nature, there is no internal delay caused by the image storage mechanism, as in previous technologies such as disk-based refresh memory storage.

1-1 SYSTEM OVERVIEW

The overall display system in the normal implementation consists of four major components: the host processor (computer), the computer interface, the RM-9000 display processor and refresh memory, and the display monitor or CRT. The interaction of these four system components produces the final image which can then be interpreted and analyzed by the human observer.

The responsibility of the host processor is to construct the RM-9000 format binary display instructions. The construction of a particular set of display instructions is usually initiated and defined by interaction with some external stimulus, perhaps a human being or another computer system.

The responsibility of the computer interface is to accept RM-9000 format binary instructions and transfer this information to the RM-9000 display processor hardware. This transfer is done on a 16-bit word-per-transfer basis. When an instruction passed to the RM-9000 dictates a change in direction of data transfer (i.e., from the RM-9000 display system to the host processor), the host processor can condition the computer interface to perform this function. In general, the computer interface actually consists of two interfaces: the interface within the computer mainframe which has access to memory within the host processor, and the RAMTEK-supplied module which is resident in the RM-9000 chassis, which converts the data transfer handshaking signals into their equivalent in the specific host-processor interface.

The responsibility of the RM-9000 display processor is to interpret the binary information transferred from the host processor, through the computer interface, and perform the display setup or generation of actual display data to refresh memory. The actual storage of information into refresh memory can either be performed directly by the display processor (as in the case of graphic or text data), or the display processor can setup and initiate a high-speed DMA transfer directly from the computer interface to the refresh memory. An integral part of the display processor is the video generator module. The video generator accesses refresh memory directly and, dependent on the type of video generator, converts the binary data stored there for each pixel to a color or grey scale and sends this information to the monitor.

The responsibility of the display monitor is to accept analog information from the video generator and excite the phosphor on the screen face, thereby producing an image which visually presents information to an observer. Standard raster scan refresh techniques are used, resulting in a refresh rate of 60 or 30 Hz dependent on system line resolution.

1-2 SCREEN COORDINATE SYSTEM

The RM-9000 refresh memory is organized as a three-dimensional coordinate system with its origin at the upper-left corner of the display screen. If one thinks in terms of the classical X-Y-Z coordinate system, the X dimension represents the element coordinate value, the Y dimension represents the line coordinate value, and Z dimension represents the binary value which is stored for each pixel (or picture element), i.e., which is stored for each unique X-Y coordinate pair (See Figure 1-1).

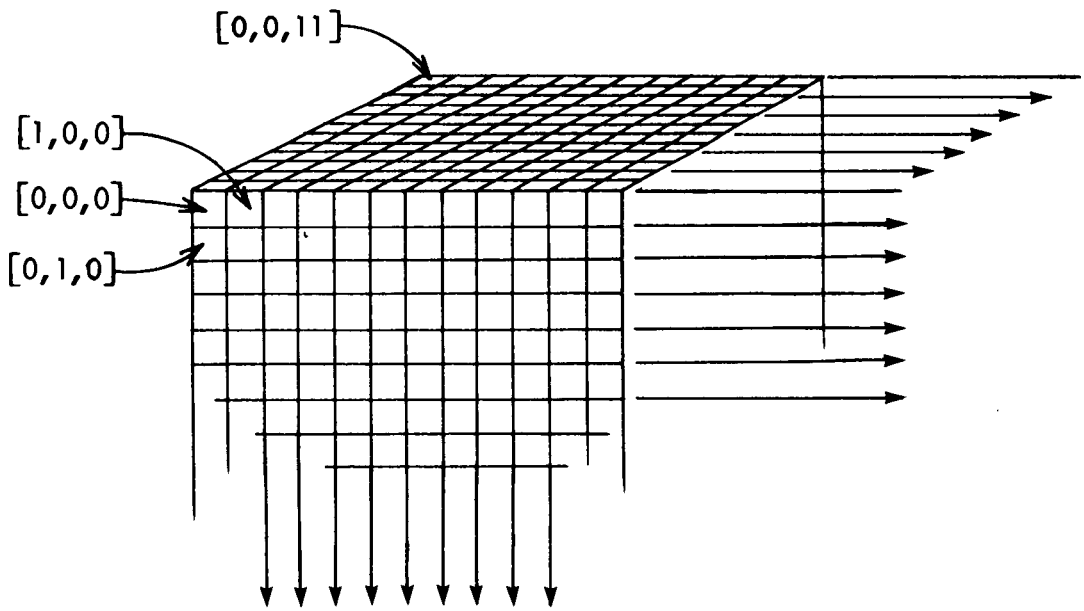


Figure 1-1 Classical X-Y-Z Coordinate System

Addressing in the Y-dimension is modulo 256_{10} or 512_{10} depending on system line resolution. Therefore, in the Y dimension, all coordinates wrap directly from the bottom of the screen back to the top of the screen.

Addressing in the X-dimension is modulo 320_{10} or 640_{10} depending upon system element resolution. Since the possible element resolution values (320_{10} and 640_{10}) are not powers of 2, if an element address is specified by the user in a display instruction is outside of the existing element address space, that element coordinate will be translated into pixels 318_{10} or 319_{10} dependent on the low-order bit of the non-existent address in a low-resolution element system, or into pixels 636_{10}, 637_{10}, 638_{10}, or 639_{10}, dependent on the low-order 2 bits of the non-existent address in a high-resolution element system.

1-3 DISPLAY DATA TYPES

Data which is transferred from the host processor to the RM-9000 display system can be classified according to the following functional types:

1) Image Data
2) Text Data
3) Raster Data
4) Graphics Data

These four (4) data types categorize all data which is transferred to the RM-9000 via a normal-format instruction and which either directly or indirectly is transferred into display refresh memory. Image data is transferred directly into refresh memory; text, raster and graphics data are transferred indirectly in the sense that the data is interpreted by the RM-9000 microprocessor firmware and data is generated to refresh memory as a result of this interpretation.

1-4 Image Data

Image data is loaded directly into refresh memory using the WRITE IMAGE normal-format instructions (See Section 3-36). The low-order 12 bits of each 16 bit image data word are loaded into the refresh memory associated with successive picture elements (or pixels) in the RM-9000 system, i.e., one image data word is written to one display pixel. Image data is always written into a rectangular region within display refresh memory. This region or "window" is defined by a parameter operand called WINDOW (See Section 3-20) which may be set in any normal-format instruction (See Section 3-4). The pixel-to-pixel updating direction for successive words of image data is defined by the parameter operand SCAN; (See Section 3-21). This parameter operand also defines the action to be performed at the window boundaries.

Image data in standard configurations defines the displayed image in one of two ways. In a Type I video standard configuration, (See Section 2-6) the image data is partitioned into three (3) 4-bit sections and used to drive three (3) 4-bit D/A converters directly to the color CRT monitor. Additionally, each subchannel (i.e., bit) of refresh memory can be displayed separately to a black and white CRT monitor. In a Type II video standard configuration (See Section 2-6), the low-order ten (10) subchannels are used as an address into a video look-up table (VLT) which contains the color definition information. Thus, any pixel which has the same data value in the low-order ten (10) subchannels will be represented by the color or greyscale value which is stored in the VLT at the address defined by the pixel value.

1-5 Text Data

Text data is transferred to the RM-9000 display system on a 2-bytes per word basis, using the WRITE TEXT normal-format instruction (See Section 3-38). Each byte of data is interpreted as an eight (8) bit ASCII code and the character font associated with the transmitted ASCII code is written into refresh memory. In a standard system (i.e., in a system without the RM-FNT option), the character font data is obtained internally from a standard PROM. The character font format is defined as a 5 pixel wide by 7 pixel high character within a 7 pixel by 9 pixel rectangle. The valid codes for a standard configuration are 20_{16} through $5F_{16}$.

Font definitions from the internal PROM are in the form of a 'ones/zeros' dot matrix. Two (2) RM-9000 internal hardware registers are used to define the color or greyscale intensity to be written into refresh memory for ones or zeros data. These are the foreground and the background registers. These values can be user-specified using the FOREGROUND and BACKGROUND parameter operands (See Sections 3-5 and 3-16). It is possible to reverse the interpretation of ones and zeros font data using the reverse-background flag in a normal-format instruction (See Section 3-9). Also, the additive write flag allows the user to specify that only ones data will be written to refresh memory (See Section 3-10).

Text data is written into refresh memory on a 'windowed' basis, i.e., characters will only be written into the rectangular region defined by the WINDOW parameter operand (See Section 3-20). The character-to-character update direction and the window margin update direction is defined by the parameter operand SCAN (See Section 3-21).

1-6 Raster Data

Raster data is transferred to the RM-9000 display system via the WRITE RASTER normal-format instruction which is part of the RM-GRA option package (See

Section 3-50). Raster data is interpreted as 'ones/zeros' data in the same manner as character font data is interpreted by the internal microprocessor. Color or grey-scale information for ones or zeros data is also defined by the foreground and background basis. Eight (8) pixels are written into by each byte of raster data, on a one (1) bit per pixel basis.

Raster data is written into refresh memory on a 'windowed' basis. The definition of this rectangular region is through the user-defined WINDOW parameter operand. Within this region, the SCAN parameter operand defines the pixel-to-pixel and the window margin update directions (See Section 3-21).

1-7 Graphics Data

Graphics data is transferred to the RM-9000 display system via the WRITE VECTOR or WRITE PLOT normal-format instructions which are part of the RM-GRA option firmware (See Sections 3-48 and 3-49). For both vector and plot generation, the graphic data specifies an endpoint coordinate which defines the vector or plot entity to be generated. For vectors, the endpoint of the previous vector is used with the current endpoint data to define the vector; for plots, the rectangular bar plot entity is defined by the previous plot entity coordinate and the current endpoint data. Windowing is not effective in graphics data mode.

1-8 DISPLAY INTERACTION

The RM-9000 display system can be configured such that interaction with the host processor through several devices is possible. Up to eight (8) keyboards or up to four (4) joysticks or trackballs are possible in a system.

1-9 Cursors/Joysticks/Trackballs

A joystick or trackball is a device which when connected to the RM-SLC serial link card can automatically change the position of a visible cursor on the display CRT monitor. This updating based on user interaction with the device is strictly a hardware function. It is possible to use the joystick or trackball to interrupt the host processor through depression of a momentary-action switch (labelled ENTER) on the device, or whenever the position of the display cursor is changed. This interrupt can be used by the host processor to signal some action to be taken perhaps based on the position of the cursor. This interpretation is completely flexible based on the needs of the host processor. Appendix 'C' defines the operational use of the joystick and trackball interactive devices.

1-10 Keyboards

The keyboard is used to transmit eight (8) bit ASCII codes from the RM-9000 display system to the host processor. The keyboards are attached to the RM-SLC serial link card and are handled by the RM-PER interactive peripheral option firmware. Each keyboard is buffered on input up to sixteen (16) characters. The entry of a character via the keyboard will generate an interrupt to the host processor if the interrupt has been enabled at the interface by the host processor. The keyboard allows user interaction via text input.

SECTION II

FUNCTIONAL DESCRIPTION

2-0 RM-9000 FUNCTIONAL DESCRIPTION

Figure 2-1 illustrates the major functional modules of the Model RM-9100, 9200 and 9300 display systems. The following is a brief description of each system component (for a detailed description of all hardware components consult the RM-9000 Theory of Operation - Volume I):

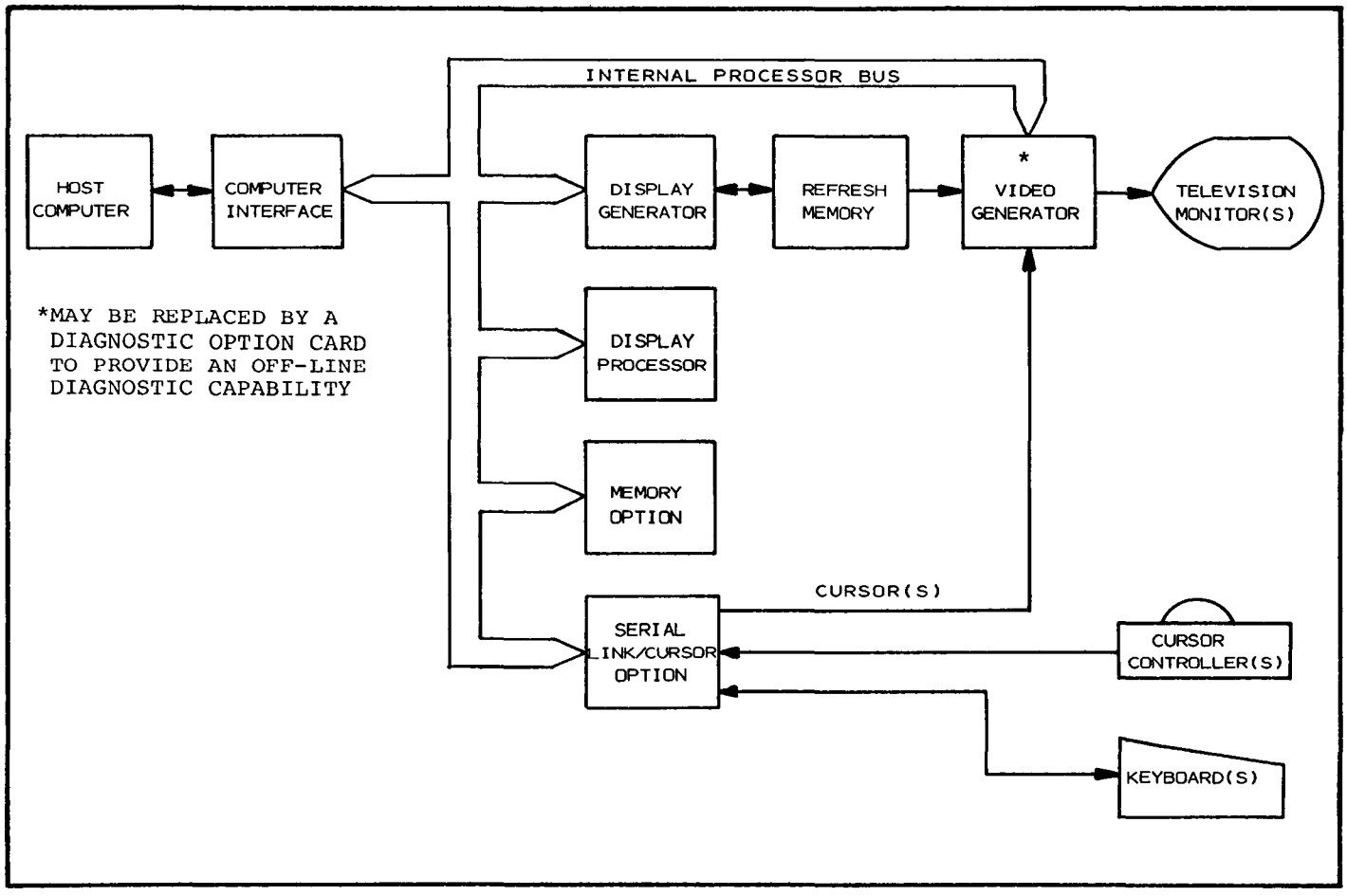


Figure 2-1 RM-9100, 9200 & 9300 Functional Block Diagram

2-1 COMPUTER INTERFACE

The RAMTEK computer interface is specific to the model of host processor used. It connects the host processor with the RM-9000 display system internal processor with the RM-9000 display system internal processor bus. A high-speed bidirectional, sixteen (16) bit parallel-bus communication path between host processor and the RM-9000 is provided. Four (4) external interrupt lines (illegal instruction, receiver, transmitter and cursor) can be generated by the RM-9000 display generator through the computer interface to the host processor. All interrupts have an enable/disable flag which is under host processor control.

Four (4) types of data transfers through the computer interface are possible:

1) Interface command word output to the RAMTEK interface
2) Interface status word input from the RM-9000 interface to the host processor
3) RM-9000 instruction/data output from the host processor to the RM-9000
4) RM-9000 data input from the RM-9000 to the host processor.

The interface command word allows the host processor to initiate such interface functions as enabling or disabling of interrupts and/or hard system resetting. This command word is strictly an interface interaction and has no RM-9000 interaction (except for hard system reset which causes an immediate reset sequence to be initiated). The interface status word defines to the host processor the status of the four (4) possible interrupts and their enable/disable functions as well as the I/O data transfer status. The instruction/data output mode is used to transfer instructions and data through the computer interface to the RM-9000 display system microprocessor. These instructions and data are in the binary formats described in Section 3 and perform the actual display generation functions. The RM-9000 data input mode is used to transfer data from the RM-9000 to the host processor after receiving one (1) of the readback initiating instructions from the host processor (RI, READ, RAM, RCS, RKB, RCM and RCMRP). This readback is always initiated by the host processor.

2-2 CONTROL BOARD

The RM-9000 control board is the component in every RM-9000 display system which contains the 8080 microprocessor and data bus, the standard RM-9000 operating firmware, as well as the display generator hardware. The internal data exchange (IDE) bus connects the control board (and thus the 8080 microprocessor data bus) with the computer interface.

2-3 8080 Microprocessor

The 8080 microprocessor contained on the control board is the principal device used to control the generation of display data to the refresh memory. The 2048 bytes of PROM resident on the control board contain 8080 microprocessor instructions used to interact with the computer interface and refresh memory using the display registers in the display generator hardware. The 8080 microprocessor 8-bit data bus is interfaced to the 16-bit RM-9000 internal processor bus through logic on the control board. It is this internal processor bus which connects all components within the RM-9000 system to the computer interface IDE bus.

2-4 Display Generator

The display generator is that portion of logic contained on the control board which provides the mechanisms for the optimized transfer of image, graphic, raster and text data to refresh memory and for the retrieval of refresh memory data in image mode on a pixel basis.

The display generator contains registers accessible to the 8080 microprocessor (in the address range 8000_{16} through $80FF_{16}$). The display generator allows 8080 firmware to perform such internal functions as refresh memory plane selection, raster/text data pattern definition, refresh memory addressing, initiation of such DMA operations as image input/output, video lookup table input/output, and internal generation of text data, and interrupt generation to the computer interface.

2-5 REFRESH MEMORY

The display refresh memory contains storage for 1 to 12 bits of data per picture element in the display system. These memories may be loaded or read directly from or to the host processor (in imaging mode via DMA across the computer interface), or they may be loaded on a pixel-by-pixel basis via the display generator in graphics, text or cartesian mode. Access time to these memories is 1.5 μ/pixel. The element/line resolution combinations for these memories is as follows:

- RM-9100 - 320 elements X 256 lines
- RM-9200 - 640 elements X 256 lines
- RM-9300 - 640 elements X 512 lines

It is these memories which supply color or intensity information for each pixel to the various video generator modules. The pixel information is constantly refreshed to the video generator(s) either at 60 Hz for the RM-9100 and RM-9200 Systems or 30 Hz for the RM-9300. This refresh process is totally under hardware control and no display processor intervention is necessary or possible.

2-6 VIDEO GENERATOR

The video generator combines information from the refresh memory and cursor generators and produces standard RS-170 compatible composite video signals to the television monitor(s). Each video generator combines and mixes this information according to different functional algorithms (in some cases, under host processor or display processor control). The following is a description of the two basic video boards:

(a) RM-VI Figure 2-2 functionally illustrates the RM-VI video board which provides for 12 B/W channels, four 7 color (RGB) channels, three 16 level grey scale channels, one 256 level grey scale channel or one 4,096 color (RGB) channel. Includes 12 direct outputs plus three 4 bit DACs (or one 8 bit and one 4 bit DAC), four cursor channels and two overlay channels.

(b) RM-V2 Figure 2-3 functionally illustrates the RM-V2 video board which provides for host programmable pseudo color or grey scale translation to any of 4,096 colors or 256 grey scale levels. Includes one 1,024 word x 12 bit programmable function memory plus three 4 bit DACs, one 8 bit DAC (assignable before or after the function memory), two cursor channels and two overlay channels.

2-7 MEMORY EXPANSION BOARD (RM-MOC)

The RM-MOC memory expansion board is an optional system component which allows the installation of the RAMTEK firmware options packages. It provides addressing for 14336_{10} bytes of PROM (addresses 0800_{16} through $3FFF_{16}$) and 8192_{10} bytes of RAM (addresses 4000_{16} through $5FFF_{16}$). Hardware logic is provided supporting DMA input and output between memory expansion RAM and interface, refresh memory, or video look-up table RAM.

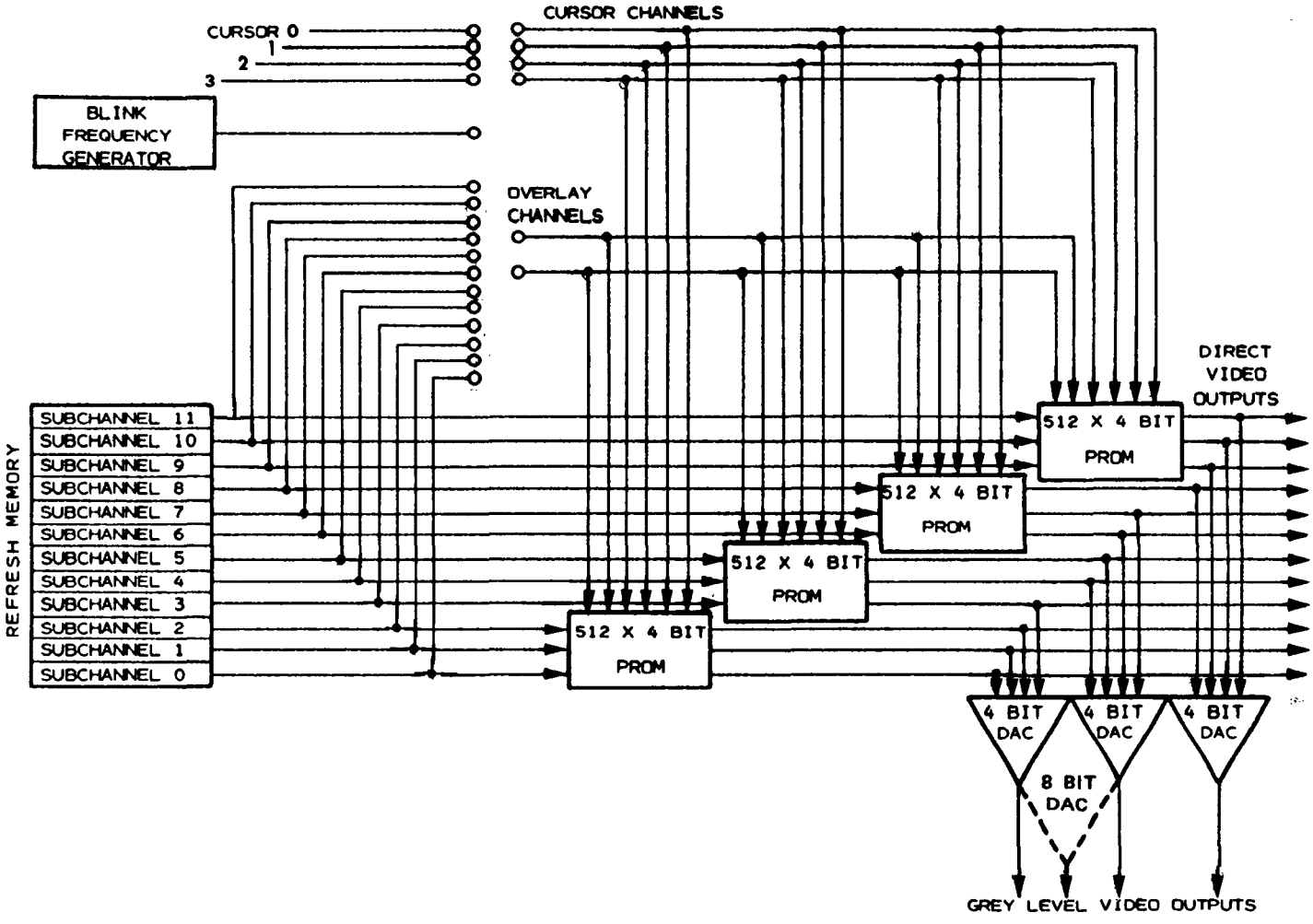


Figure 2-2 RM-VI Functional Block Diagram

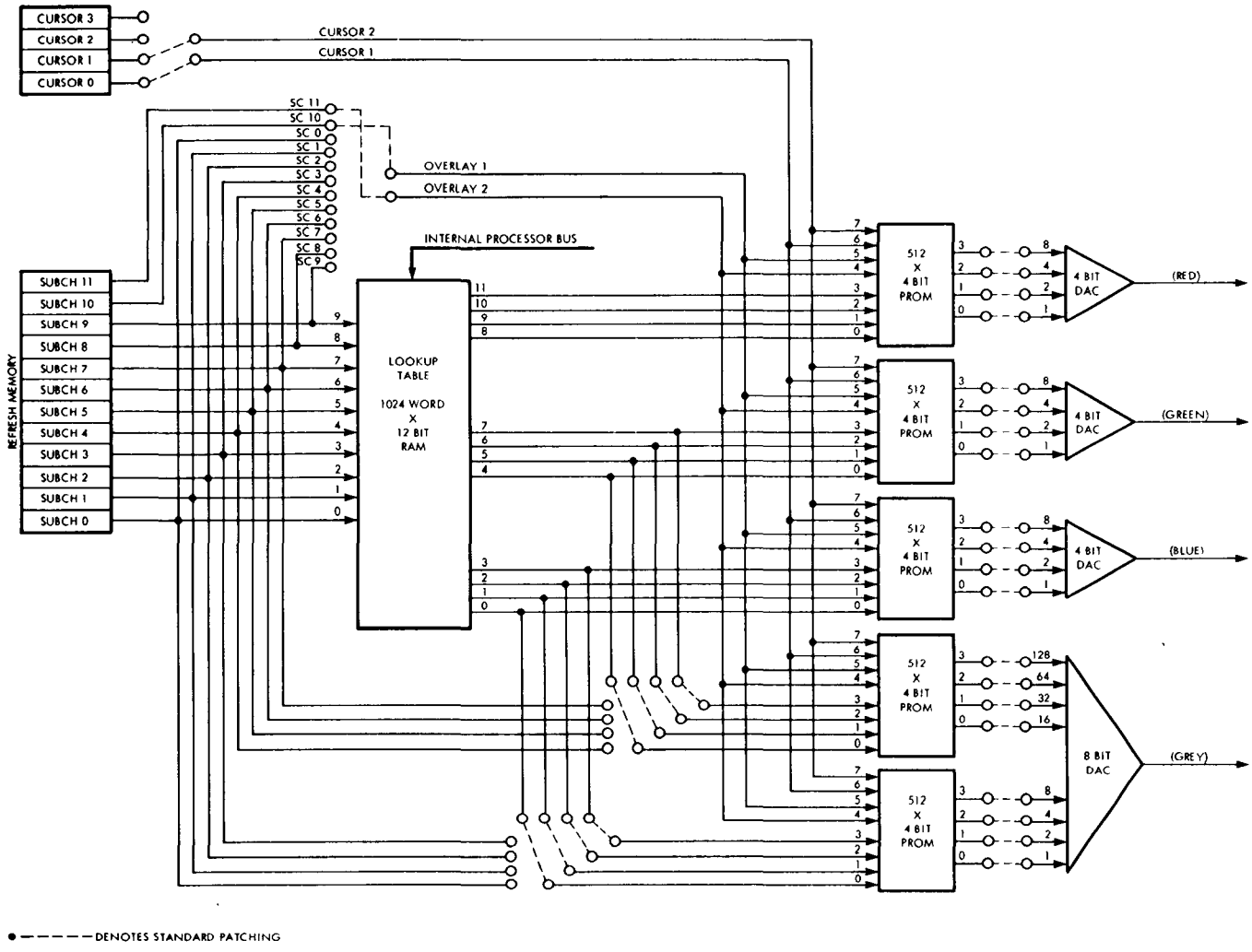


Figure 2-3 RM-V2 Functional Block Diagram

2-8 SERIAL LINK BOARD (RM-SLC)

The RM-SLC serial link board provides system communication with external interactive devices such as keyboards and joysticks or trackballs over serial communication ports. Each serial link card also can generate up to two video cursors. This cursor is mixed into the video generator card output for display. Each serial link card can accommodate either four (4) keyboard/transmitters or two (2) keyboard/transmitters and two (2) cursors. A maximum of 2 RM-SLC serial peripherals firmware packages provides the firmware mechanism for host processor communication with these interactive peripheral devices.

Ramtek

SECTION III

RM-9000 INSTRUCTION SET

SECTION III

RM-9000 INSTRUCTIONS

3-0 INTRODUCTION

The RM-9000 series is a microprocessor-controlled graphic display system. The standard firmware instruction set provides a high-level mechanism for the storage and retrieval of image data and associated image-generation information at high-data rates, as well as the generation of alphanumeric text information. This instruction frees the user from the complex tasks of communication directly with the hardware display registers. The standard firmware package provided by RAMTEK reduces significantly the amount of software display processing in the host computer.

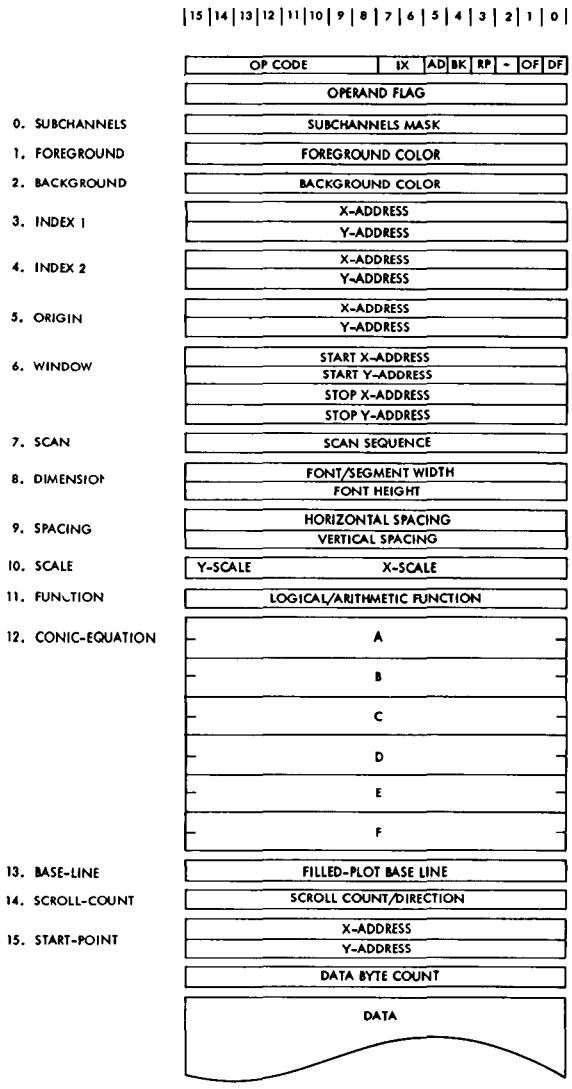
3-1 INSTRUCTIONS

The RM-9000 provides a set of standard instructions and a variety of optional instructions. This manual will cover all currently defined instructions. Since the RM-9000 is a very flexible system, the set of optional instructions will probably continue to increase. Addenda to this manual will be provided for additional instruction sets.

The RM-9000 instruction consists of two basic types of instructions, normal format and special format instructions. Normal format instructions constitute a set of high-level imaging, text and graphics functions, in a flexible, yet uniformly-defined format. The normal format instruction set reduces the user's programming effort with an attempt to optimize high-speed image and text generation. The special format instructions are used to perform functions which do not affect the display directly (i.e., do not write into or read from refresh memory). The format of the special instructions is different for each instruction.

The RM-9000 will generate an illegal instruction interrupt request whenever an optional instruction is referenced when the associated firmware option is not installed. The same interrupt request will also be generated when a truly undefined op code is referenced.

Currently defined optional instructions may be added at a later date. The standard software will sense the presence of the option PROM and reference it when



LEGEND

IX = ADDRESSING MODE (0 = ABSOLUTE, 1 = INDEX-1, 2 = INDEX-2, 3 = RELATIVE)

AD = ADDITIVE WRITE (0 = REPLACEMENT, 1 = ADDITIVE)

BK = REVERSE BACKGROUND (0 = NORMAL BACKGROUND, 1 = REVERSED BACKGROUND)

RP = REVERSE PACKING FLAG (0 = LEFT BYTE FIRST, 1 = RIGHT BYTE FIRST)

OF = OPERAND FLAG (0 = NO ARGUMENTS OR FLAG WORD EXISTS, 1 = FLAGGED ARGUMENTS EXIST)

DF = DATA FLAG (0 = NO DATA OR LENGTH WORD EXISTS, 1 = n DATA BYTES

Figure 3-1 Normal-Format Instruction Layout

it is installed. The installation of currently undefined option software will also require replacement of the standard option PROM's to enable the addition of the new op codes.

The instruction repertoire for the RM-9000 is listed in Table 3-1.

3-2 INSTRUCTION FORMAT

All instructions are 16-bit parallel and consist of one (1) or more 16-bit words. The initial 8-bits (most significant) of the initial word always describe the operation to be performed by the instruction. The remaining bits (and words) may be interpreted differently for each instruction.

The RM-9000 has a "normal" instruction format and a "special" instruction format. The normal format is specifically designed for the general requirements of imaging and graphics. The special format instructions are oriented to the control of entities not directly related to the refresh memory, such as serial devices, video lookup tables, etc. Each special format instruction is unique. Normal format instructions, on the other hand, share a common (variable length) form.

The special format may be used to reduce data flow across the interface or to handle unique data requirements. Each special format will be described with the instruction itself.

3-3 RM-9000 DATA MODES

Data is stored in the RM-9000 refresh memory in one of two data modes: image mode or raster mode. The RM-9000 display controller always writes a data value of up to twelve (12) data bits per pixel into refresh memory; the origin of this data determines the data mode. In image data mode, the low-order 12-bits of a 16-bit data word are stored in up to twelve (12) subchannels of refresh memory for a single pixel. This storage is only done in those refresh memory subchannels which are selected by the subchannel mask parameter. Therefore, in image mode, it is possible to store any 12-bit value in any pixel in refresh memory. Since the data to be loaded is externally generated, the FOREGROUND and BACKGROUND parameters which are crucial to raster data mode are not used. In raster data mode, only the FOREGROUND or BACKGROUND values are written into a pixel location. Each bit of raster data represents a separate pixel, and the value of each raster bit selects whether the FOREGROUND or BACKGROUND value will be written into the represented pixel. The reverse-background flag BK influences raster data mode in that when BK=1, the polarity of incoming raster data is reversed. The additive-write flag AD is also used in raster data mode to prevent the writing of raster data with a zero bit value. This is useful in writing text

Table 3-1 RM-9000 Instruction Repertoire

OP-CODE OCT	OP-CODE HEX	INSTRUCTION NAME	MNEMON	KEY F	KEY O	CONTROL FLAGS (BIT 2 IS UNUSED) IX 6,7	AD 5	BK 4	RP 3	OF 1	DF 0	PARAMETER FLAGS (NORMAL INSTRUCTIONS ONLY) COP 15	SCR 14	BAS 13	CON 12	LAF 11	SCL 10	SPC 9	DIM 8	SCN 7	WIN 6	ORG 5	IX2 4	IX1 3	BGD 2	FGD 1	MSK 0
000	00	(UNDEFINED)	–	–	–																						
001	01	LOAD HARD REGISTER	LOAD	S	–																						
002	02	READ SOFT REGISTER	READ	S	–																						
003	03	LOAD AUXILIARY MEMORY	LAM	S	–																						
004	04	READ AUXILIARY MEMORY	RAM	S	–																						
005	05	RESET	RSET	S	–							I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
006	06	INITIALIZE	INIT	S	–							I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
007	07	NO OPERATION	NOP	N	–	S	S	S	S	S	S	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
010	08	SET PARAMETER	SET	N	–	X	S	S	S	X	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
011	09	ERASE	ERS	N	–	X	X	X	S	X	S	S	S	S	S	S	S	S	S	S	X	X	X	X	X	X	X
012	0A	WRITE IMAGE	WI	N	–	X	S	S	S	X	X	X	S	S	S	X	X	S	S	X	X	X	X	X	S	S	X
013	0B	READ IMAGE	RI	N	–	X	S	S	S	X	X	X	S	S	S	S	S	S	S	X	X	X	X	X	S	S	S
014	0C	WRITE TEXT	WT	N	–	X	X	X	X	X	X	X	S	S	S	S	X	X	X	X	X	X	X	X	X	X	X
015	0D	WRITE RASTER	WR	N	G	X	X	X	X	X	X	X	S	S	S	S	X	S	S	X	X	X	X	X	X	X	X
016	0E	WRITE VECTOR	WV	N	G	X	S	X	S	X	X	X	S	S	S	S	S	X	S	S	X	X	X	X	X	X	X
017	0F	WRITE CONIC	WC	N	C	X	S	X	S	X	X	X	S	S	X	S	S	S	S	S	S	X	X	X	X	X	X
020	10	WRITE PLOT	WP	N	G	X	S	X	S	X	X	X	S	X	S	S	S	X	X	X	S	X	X	X	X	X	X
021	11	SCROLL X	SCRX	N	S	X	S	X	S	X	X	S	X	S	S	S	S	S	S	S	X	X	S	S	X	X	X
022	12	SCROLL Y	SCRY	N	S	X	S	X	S	X	X	S	X	S	S	S	S	S	S	S	X	X	S	S	X	X	X
023	13	SAVE ENVIRONMENT	PUSHE	S	M							↑	↑	↑	↑	?	?	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
024	14	RESTORE ENVIRONMENT	POPE	S	M							↓	↓	↓	↓	?	?	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
025	15	LOAD PROGRAMMABLE FONT	LPF	S	F																						
026	16	WRITE CURSOR STATE	WCS	S	P																						
027	17	READ CURSOR STATE	RCS	S	P																						
030	18	WRITE KEYBOARD	WKB	S	P																						
031	19	READ KEYBOARD	RKB	S	P																						
032	1A	SENSE PERIPHERAL STATUS	SPS	S	P																						
033	1B	LOAD CONTROL MEMORY	LCM	S	U																						
034	1C	READ CONTROL MEMORY	RCM	S	U																						
035	1D	CALL CONTROL MEMORY	CCM	S	U																						
036	1E	EXECUTE INSTRUCTION MEMORY	XIM	S	U																						
037	1F	LOAD CONTROL MEMORY (REV)	LCMRP	S	U																						
040	20	READ CONTROL MEMORY (REV)	RCMRP	S	U																						
041	21	LOAD PROGRAMMABLE FONT (REV)	LPFRP	S	F																						
042	22	MAGNIFY	MAGNFY	N	S	X	X	X	S	X	X	X	S	S	S	S	S	S	S	S	X	X	X	X	X	X	X
050	28	WRITE DASHED VECTOR	WDV	N	G	X	S	X	S	X	X	X	S	S	S	S	S	X	S	S	S	X	X	X	X	X	X

KEY LEGEND

FORMAT TYPES

N NORMAL

S SPECIAL

OPTION PRE-REQUISITES

P INTERACTIVE PERIPHERALS

G GRAPHICS

S SCROLL

M STATUS MANAGEMENT

F PROGRAMMABLE FONT

U USER-SUBROUTINE

C CONICS + GRAPHICS

S => SET/DEFINE FLAG/PARAMETER

X => USED IN EXECUTING THE INSTRUCTION

\- => NO EFFECT, NO OPERATION PERFORMED

↑ => PUSH DATA ON TO STACK

↓ => POP DATA OFF OF STACK

I => INITIALIZE TO DEFAULT VALUES

? => NO INTERNAL BUFFERS SAVED

on to an existing display; only the character (and not its font background) will be stored in refresh memory. Table 3-2 defines the mode used by the various complex instructions for writing data to refresh memory.

Note that these data modes define the mechanism by which data is stored in refresh memory. It is essential that the user understand the implications of each data storage mode in order to effectively use the RM-9000 instruction set.

Table 3-2 RM-9000 Instruction Set

INSTRUCTION	DATA MODE
ERASE	Raster
WRITE IMAGE	Image
READ IMAGE	Image
WRITE TEXT	Raster
WRITE RASTER	Raster
WRITE VECTOR	Raster
WRITE PLOT	Raster

3-4 NORMAL INSTRUCTION FORMAT

The normal instruction format can carry a variety of parameter and data information to the display system. The format also allows a variable amount of parameters and data information. The variable format allows the user to transmit only the information used in a particular operation. This means that a very complex function, such as window erase, can be performed via a single 16-bit word when the window parameters have been previously defined.

Figure 3-1 illustrates the normal instruction format. The first (most significant) byte of the first word defines the operation code as is done in all RM-9000 instructions.

The second (least significant) byte is called the parameter byte and defines the coordinate addressing mode, additive or replacement writing mode, reverse background mode, byte processing order and the presence of operand parameters and/or data.

3-5 Parameter Byte

The fields in the parameter byte are defined as follows:

NAME	BIT POSITION
Data Flag	0
Operand Flag	1
Undefined	2
Reverse Packing	3
Reverse Background	4
Additive Write	5
Addressing Mode	6, 7

3-6 Data Flag (Bit 0)

A Data Flag (DF) value of 1 indicates that a data length word (i.e., the numbers of bytes of data) and the specified number of data bytes will follow any complex parameters that might be set by the complex instruction. A (DF) value of 0 indicates the absence of a data length word or any data following any complex parameters. For a complete description of the data length word and the data format, see Sections 3-30 and 3-31.

3-7 Operand Flag (Bit 1)

An Operand Flag (OF) value of 1 indicates the presence of the operand flag word. An (OF) value of 0 indicates the absence of an operand flag word and correspondingly the absence of any parameter operands. The operand flag word follows the instruction word if present. (See Section 3-11 for a discussion of the operand flag word.)

3-8 Reverse Packing Flag (Bit 3)

The Reverse Packing Flag (RP) specifies the packing mode for byte oriented data. The (RP) flag effects only the data (not parameter) of byte-oriented instructions such as WI, RI, WT, and WR. A (RP) value of 0 indicates normal packing and a (RP) value of 1 indicates reversed packing.

The normal packing mode for 8-bit bytes in the 16-bit word specifies that bytes are unpacked and processed from left-to-right, i.e., the most significant data byte is processed first. Reverse packing means the order of unpacking is right-to-left, i.e., the least significant byte is processed first.

3-9 Background Flag (Bit 4)

The Background Flag (BK) selects between normal and reverse background for text, raster and graphic commands. Normal background is selected when the (BK) bit is zero and reverse background when (BK) is one.

The RM-9000 uses either the BACKGROUND parameter or the FOREGROUND parameter when generating raster, text and graphic data. The normal mode is for the FOREGROUND value to be selected when a data bit is one (1) and the BACKGROUND value is written to refresh memory when a data bit is zero (0). The reverse background mode reverses the selection of these two parameters just as if the parameters had been exchanged or as if the data bits were reversed. (See raster data section.)

3-10 Additive Flag (Bit 5)

The Additive Write Flag (AD) when set to a one will cause raster, text and cartesian data (i.e., data generated by the ERS instruction) to be written in refresh memory in an "additive" fashion, i.e., data bits with a zero value cause nothing to be written to memory and only data bits with a one (1) value cause data to be written.

An Additive Write Flag value of zero (0) causes "replacement writing of memory, i.e., both zero and one bits will cause memory contents to change.

The user may select combinations of the additive and background flags. Table 3-3 gives the various relationships.

3-11 Addressing Mode Flags (Bit 6, 7)

The normal instruction format permits selection of one of four addressing modes for each instruction. All coordinate information such as window position is modified by the addressing mode selected.

Table 3-3 Functional Relationship of Data, Background Flag & Additive-Right Flag

		AD 0	0	1	1
BK	0	B	F	X	F
	1	F	B	X	B
		0	1	0	1

DATA BIT

F WRITE FOREGROUND VALUE
B WRITE BACKGROUND VALUE
X NO WRITE

Additive (AD) - Background (BK) & Data Relation

The addressing mode flags (IX) are absolute (IX = 00), index using Index 1 (IX = 01), index using Index 2 (IX = 10), and relative (IX = 11). The RM-9000 maintains two (2) internal index registers, IX1 and IX2 which are set by the INDEX parameter (See Sections 3-17 & 3-18).

Absolute Addressing

The x, y values in the parameter or data list are used directly as screen coordinates.

Index Addressing

The parameter or data value referencing screen coordinates is added to the index selected to determine the coordinate desired.

Relative Addressing

The parameter or data value referencing screen coordinates is added to the current operating point (last screen coordinate read or written).

The relative addressing mode can be used to create a sequence of vectors, each new endpoint being relative to the termination of the previous endpoint. Table 3-3A summarizes the attributes of the 4 addressing modes.

Ramtek

Table 3-3A Attribute Summary of 4 Addressing Modes

IX	TYPE OF ADDRESSING	MNEMONIC	EQUATION
00	Absolute Addressing	AA^0BS	(REG)←⟨DATA⟩+ 0
01	Index 1 Addressing	AIXI	(REG)←⟨DATA⟩+ (INDEX 1)
10	Index 2 Addressing	AIX2	(REG)←⟨DATA⟩+ (INDEX 2)
11	Relative Addressing	AREL	(REG)←⟨DATA⟩+ (COP)

⟨DATA⟩ = 16-bit coordinate information arriving from the computer.

(REG) = content of a 16-bit data (or parameter) target register.

Notice that REG may be INDEX 1, INDEX 2, or COP as endpoint for vectors, plots, etc.

The index addressing modes allow the creation of a display independent of screen position. Note that the index register value used is the previous value defined for that index register, i.e., previous to the complex instruction which is using indexed addressing. Parameter values which refer to screen coordinates (such as WINDOW or START POINT) are computed by performing a 2's complement addition of the parameter value with the selected index register.

The absolute mode provides standard screen addressing capability.

The RM-9000 screen is addressed with a coordinate system having the upper left corner as (0, 0).

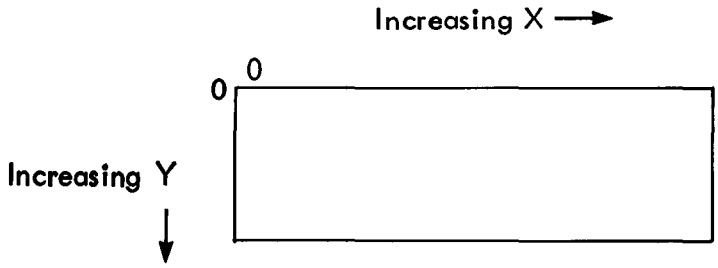


Figure 3-2 Screen Coordinate Orientation

3-12 Operand Flag Word

The Operand Flag Word indicates the presence or absence of the sixteen (16) possible parameters operands for normal instructions. Each parameter operand for a normal instruction must appear in a fixed sequence. Each bit in the Operand Flag Word corresponds to a parameter operand in the same sequence. A zero bit in the same position as a parameter operand indicates the operand is absent. Each bit is interpreted from right-to-left, i.e., Bit 0 corresponds to the first parameter operand and Bit 15 corresponds to the last parameter operand. Figure 3-3 shows the Operand Flag Word Format.

3-13 Parameter Operands

The purpose of the parameter operands is to define (load) the various internal parameter registers which define the operation and subsequent display for all normal instructions. The parameter operands may be set in any RM-9000 normal instruction and affect only the operation of normal instructions. Parameter operand values are non-volatile from normal instruction to normal instruction; once a parameter operand is set by a normal instruction, the parameter operand value remains the same until reset by the user (except COP). The presence of any set or subset of parameter operands

BIT	MNEMONIC	FUNCTION
0	MSK	SUBCHANNEL (MASK)
1	FGD	FOREGROUND
2	BGD	BACKGROUND
3	IX1	INDEX 1
4	IX2	INDEX 2
5	ORG	ORIGIN
6	WIN	WINDOW
7	SCN	SCAN
8	DIM	DIMENSION
9	SPC	SPACING
10	SCL	SCALE
11	LAF	FUNCTION (LOGICAL & ARITHMETIC)
12	CON	CONIC EQUATION
13	BAS	BASE LINE
14	SCR	SCROLL COUNT
15	COP	START POINT (CURRENT OPERATING POINT)

Figure 3-3 Operand Flag Word Format

is indicated by the state of the 16-bits in the operand flag word (See Section 3-12). The order of appearance of each of the parameter operands is defined by the position of its flag bit in the operand flag word. A parameter operand with a flag-bit position of m will appear before a parameter operand with a flag-bit position of (n), if (m) is less than (n). The number of words associated with each parameter operand is fixed, but this number varies from 1 up to 12_{10} words. The parameter operands involving coordinate data are sensitive to the addressing mode bits, but not to any of the other bits in the control byte.

3-14 Subchannel Parameter (No. 0)

				S_{11}	S_{10}	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The SUBCHANNEL parameter may be set via any normal instruction (except NOP). Its presence is flagged via Operand Flag Bit 0. The operand itself is a single 16-bit word and write-enables a prescribed combination of refresh memory bit planes (subchannels) for image generation purposes. There are twelve (12) possible subchannels and twelve (12) corresponding bits in the SUBCHANNEL argument, i.e., Bit 0 corresponds to Subchannel 0, Bit 1 to Subchannel 1, ..., and Bit 11 to Subchannel 11. When set to a "one" state, the corresponding subchannel is enabled, and visa-versa. Disabled subchannels are not affected by write operations (including Erase). The subchannel parameter has no effect during read instructions, e.g., Read Image "a" will respond with a value for all subchannels in the system. Although the RM-9000 is normally configured as a single channel system, the SUBCHANNEL parameter provides for multi-channel partitioning of the refresh memory. For example, the refresh memory might be partitioned into four (4) channels of three (3) subchannels each, with each channel producing a seven (7) color (plus black) image.

SUBCHANNEL default value = OFFF (Hex).

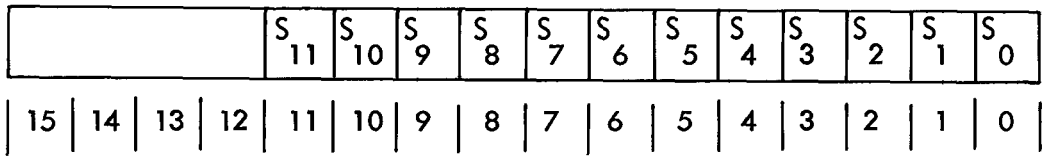
3-15 Foreground Parameter (No. 1)

				S_{11}	S_{10}	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The FOREGROUND parameter may be set via any complex format instruction (except NOP). It's presence is flagged via Operand Flag Bit 2^1. The operand itself is a single 16-bit word in length and is identical in format to the SUBCHANNELS argument. It establishes foreground color or intensity for normal foreground ("one" bits) font, raster and graphics data by assigning a "one" or "zero" for each of the twelve (12) possible subchannels. When writing reversed background character font and raster data, FOREGROUND establishes color or intensity for "zero" state data.

FOREGROUND default value = OFFF (Hex).

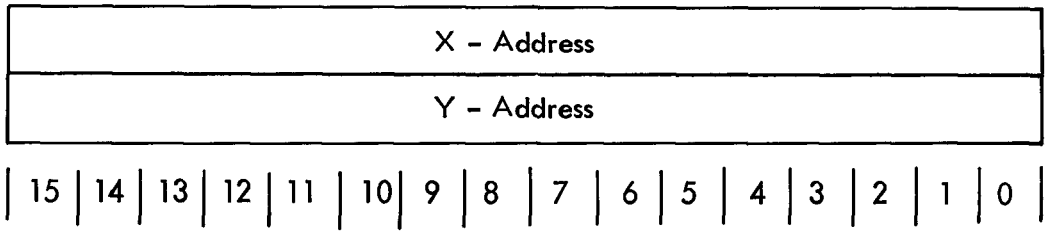
3-16 Background Parameter (No. 2)



The BACKGROUND parameter may be set via any complex format instruction (except NOP). Its presence is flagged via Operand Flag Bit 2. The operand itself is a single 16-bit word in length and is identical in format to the SUB-CHANNELS argument. Interpretation is identical to the FOREGROUND argument except that background color or intensity is specified, i.e., the color of normal background, "zero" state font or raster data. When background is reversed, data interpretation is likewise reversed. That is, BACKGROUND then specifies the color or intensity of "one" state character font, raster or graphics data.

BACKGROUND default value = 0000 (Hex).

3-17 Index 1 Parameter (No. 3)



The INDEX 1 parameter may be set via any complex format instruction (except NOP). Its presence is flagged via Operand Flag Bit 3. The operand itself is two 16-bit words in length and specifies a local addressing origin (or displacement). The first word specifies X address or horizontal displacement from element 0, whether positive or negative. The second word likewise specifies Y address or vertical displacement from line 0. Received coordinate values (X/Y) in subsequent normal instructions and in subsequent parameters in the current normal instruction are conditionally summed with the current INDEX 1 specified values in order to derive the effective (refresh memory) address, i.e., provided that Index Register 1 addressing was specified in the first word of the received command (IX = 01). If IX $\neq$ 0, the values loaded into the X and Y components of IX1 are the sum of current X and Y components of the specified registers and the X and Y

components of the operand. Therefore, IX = 01 causes IX1 to be the sum of the old contents and the new operand values.

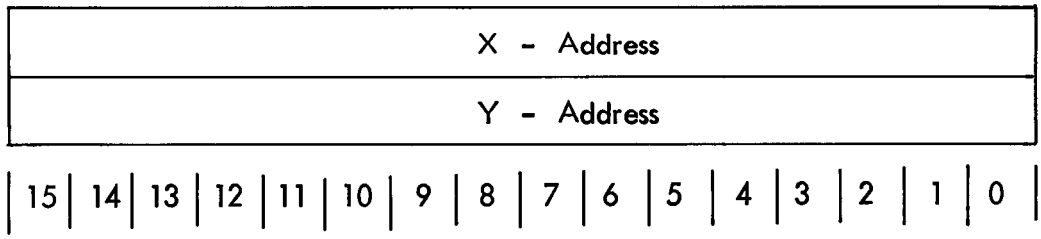
The actual values used for IX1 (and IX2) are computed as follows:

- If IX = 0 (absolute addressing mode), the actual received argument values for IX1 (both X and Y) are used as absolute addresses.
- If IX = 1 (Index 1 addressing mode), the received argument values for IX1 (both X and Y) are summed with the current values of IX1 to form a new set of IX1 values.
- If IX = 2 (Index 2 addressing mode), the actual received arguments values for IX1 (both X and Y) are summed with the current values of IX2 to form a new set of IX1 values.
- If IX = 3 (Relative addressing mode), the actual received argument values for IX1 (both X and Y) are summed with the current values of the XCOP and YCOP to form a new set of IX1 values.

  INDEX 1 X - address default value = 0000 (Hex).

  INDEX 1 Y - address default value = 0000 (Hex).

3-18 Index 2 Parameter (No. 4)



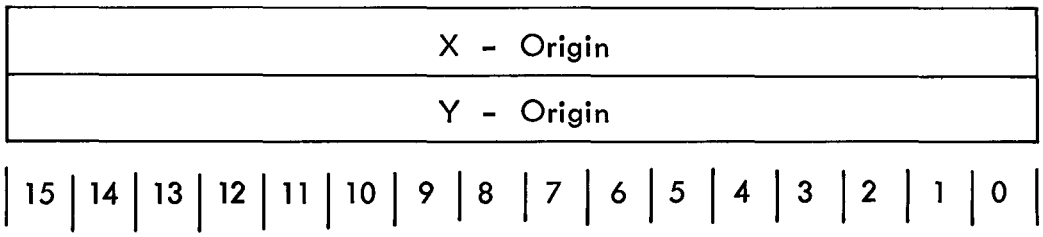
The INDEX 2 parameter may be set via any complex format instruction (except NOP). Its presence is flagged via Operand Flag Bit 2^4. The operand itself is two (2) 16-bit words in length and is identical to the INDEX 1 argument in both format and treatment. That is, received coordinate values (X/Y) in subsequent arguments are summed with the current INDEX 2 specified values in order to derive the effective (refresh memory) address, i.e., provided that Index Register 2

addressing was or is specified in the first word of the received command (IX = 10). Calculation of the absolute IX2 register values is computed in a manner identical to IX1.

INDEX 2 X - Address default value = 0000 (Hex).

INDEX 2 Y - Address default value = 0000 (Hex).

3-19 Origin Parameter (No. 5)



The ORIGIN parameter may be set via any normal instruction (except NOP). Its presence is flagged via Operand Flag Bit 5. The operand itself is two (2) 16-bit words and defines an address in refresh memory which becomes the origin for the video output to the CRT monitor (i.e., the upper, left-hand corner of the screen). Due to the RM-9000 memory system architecture, the ORIGIN values which place absolute refresh memory location (0,0) at the upper, left-hand corner of the video display are non-zero and different for each system type. These are the default value settings; Table 3-4 defines the default origin values for each RM-9000 series system type. The legal range of values for both the element and line origins is from zero to the element or line resolution value minus one. Therefore, for an RM-9300 display system, the legal range of element origin values is from 0 through 639_{10} and the legal range of line origin values is from 0 through 511. When altering the origin values, all data remains visible since wrapping will take place in both dimensions.

ORIGIN default values: See Table 3-4.

Table 3-4 Systems Resolution Definitions & Origin Default Values

SYSTEM	ELEMENT (X) RESOLUTION	LINE (Y) RESOLUTION	ELEMENT (X) DEFAULT ORIGIN	LINE (Y) DEFAULT ORIGIN
RM-9100	320_{10} 140_{16}	256_{10} 100_{16}	310_{10} 136_{16}	255_{10} FF_{16}
RM-9200	640_{10} 280_{16}	256_{10} 100_{16}	620_{10} $26C_{16}$	255_{10} FF_{16}
RM-9300	640_{10} 280_{16}	512_{10} 200_{16}	620_{10} $26C_{16}$	510_{10} $1FE_{16}$

3-20 Window Parameter (No. 6)

Start X - Address (X_L)
Start Y - Address (Y_T)
Stop X - Address (X_R)
Stop Y - Address (Y_B)

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

The WINDOW parameter may be set by any normal instruction (except NOP). Its presence is flagged by Operand Flag Bit 6. The operand itself is four (4) 16-bit words and specifies a rectangular region used in conjunction with the ERS, WI, RI, WT, WR, SCRX, SCRY, and MAG instructions. The WINDOW values are read in the following order: X_L, Y_T, X_R, Y_B (where these values correspond to coordinates in Figure 3-4 . It is necessary that the WINDOW parameters conform to the following conditions: $X_L \leq X_R$ and $Y_T \leq Y_B$. Whenever the WINDOW parameter is specified, the Current Operating Point (COP) is set to the coordinate defined by Table 3-5 determined by the value of the SCAN parameter prior to this instruction. The default values for WINDOW are such that the entire refresh memory is within the window.

WINDOW START X (X_L) default value = 000
WINDOW START Y (Y_T) default value = 000
WINDOW STOP X (X_R) default value = XRES-1
WINDOW STOP Y (Y_B) default value = YRES-1

where XRES = system element resolution

YRES = system line resolution

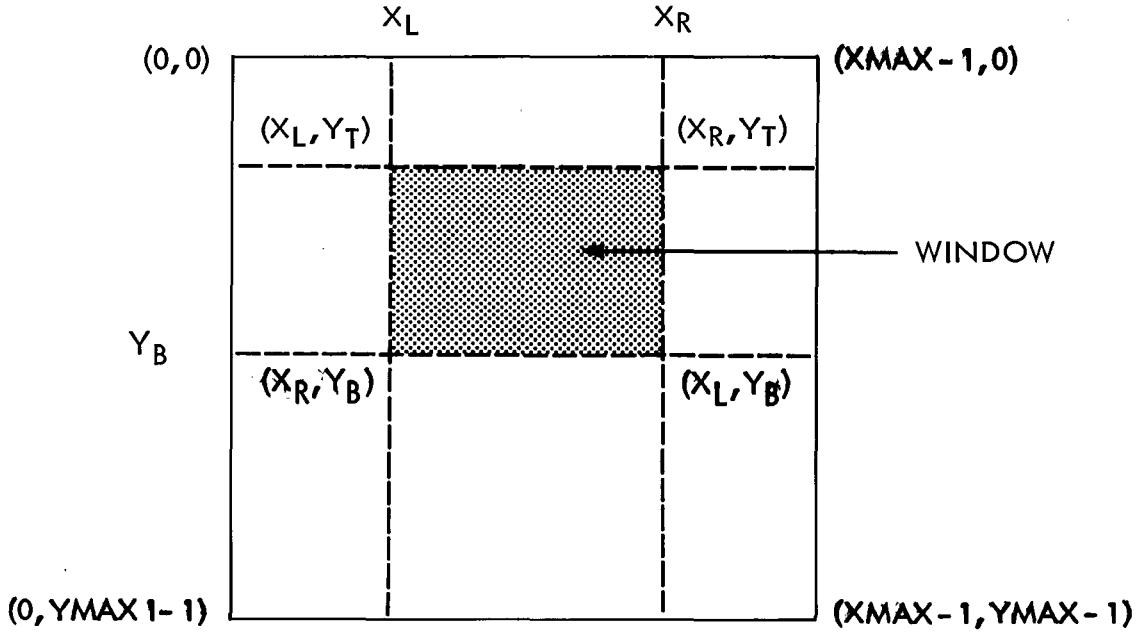


Figure 3-4 Window Definition

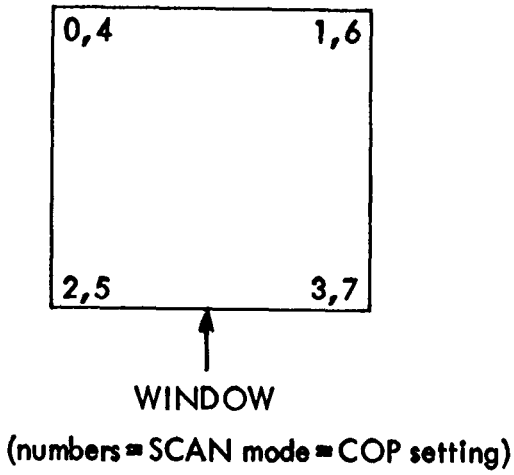
3-21 Scan Parameter (No. 7)

Scan Sequence															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The SCAN parameter may be set via any normal instruction (Except NOP). Its presence is flagged via Operand Flag Bit 7. The operand itself is a single 16-bit word in length and specifies one (1) of eight (8) possible scan sequences for the

Table 3-5 COP Placement After Window Setting

SCAN	XCOP	YCOP
0	X_L	Y_T
1	X_R	Y_T
2	X_L	Y_B
3	X_R	Y_B
4	X_L	Y_T
5	X_L	Y_B
6	X_R	Y_T
7	X_R	Y_B



WI, RI, WR, WT and WP instructions. For the WI, RI, WR instructions, SCAN is defined by Table 3-6.

Primary scan is the direction of consecutive pixels. Secondary scan is the wrap-around direction upon reaching a window boundary. That is, when the primary scan completes a line of pixels and is ready for wrap-around, the secondary scan will determine whether the second line of pixels is above, below, to the right or to the left of the first line.

For write text, scan direction is defined by both the SCAN and the SPACING parameters. The SCAN mode will determine the character orientation and whether the primary and secondary updates are horizontal or vertical. The SPACING parameter determines the direction of the update, i.e., to the left, to the right, up or down. The primary update is the update between successive characters. The secondary update is the update between successive character lines, i.e., the secondary update determines whether the second line of characters is above, below, to the right or to the left of the first line of characters. See Table 3-7.

A new line of characters is started whenever the last character reaches or passes the window boundary or a carriage return or line feed is encountered. The edge

Table 3-6 Image & Raster Mode Scan Directions

SCAN	DIRECTION OF WRITING PROCESS	
	PRIMARY	SECONDARY
0	Left-to-Right	Top-to-Bottom
1	Right-to-Left	Top-to-Bottom
2	Left-to-Right	Bottom-to-Top
3	Right-to-Left	Bottom-to-Top
4	Top-to-Bottom	Left-to-Right
5	Bottom-to-Top	Left-to-Right
6	Top-to-Bottom	Right-to-Left
7	Bottom-to-Top	Right-to-Left

of a character may exceed the window boundary. Wrap around is by complete characters only. A carriage return will start the next character line at the opposite window boundary.

Table 3-7 Write Text Scan Direction

SCAN	PRIMARY UPDATE	SECONDARY UPDATE	CHARACTER ORIENTATION
0	Horizontal	Vertical	A
1	Horizontal	Vertical	⋗
2	Horizontal	Vertical	∢
3	Horizontal	Vertical	∀
4	Vertical	Horizontal	A
5	Vertical	Horizontal	∢
6	Vertical	Horizontal	⋗
7	Vertical	Horizontal	∀

Table 3-8 Window Origin

SCAN	WINDOW ORIGIN
0	Upper Left-Hand Corner
1	Upper Right-Hand Corner
2	Lower Left-Hand Corner
3	Lower Right-Hand Corner
4	Upper Left-Hand Corner
5	Lower Left-Hand Corner
6	Upper Right-Hand Corner
7	Lower Right-Hand Corner

A line-feed will start a new character line at the current character position, i.e., no return to the opposite window boundary is made.

The positive direction for both the primary and secondary updates is to the right and down. If it is desired for either the primary or the secondary update to move to the left or up, then this update must be expressed as a two's complement negative number. See Figure 3-5 for the normal spacing values for each SCAN mode.

Normally the absolute values of the spacing parameters are equal to or greater than the character dimension parameters. If the spacing parameters are less than the dimension parameters, then the characters will overlap.

SCAN default value = 0

3-22 Dimension Parameter (No. 8)

WIDTH															
HEIGHT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

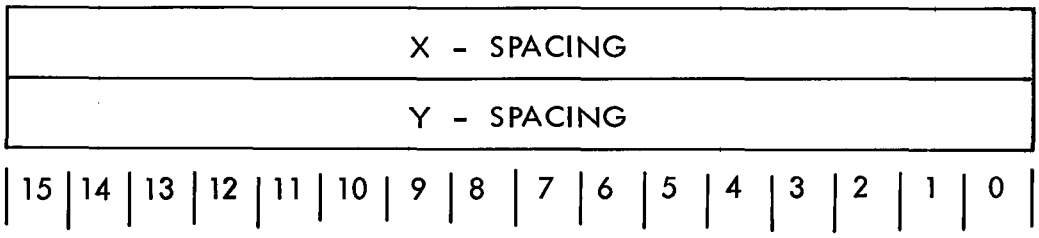
The DIMENSION parameter may be set via any normal instruction (except NOP). Its presence is flagged via Operand Flag Bit 8. The operand itself is two (2) 16-bit words in length and specifies the dimensions of the alphanumeric font in terms of height and width, and the height or width of individual plot segments in terms of lines or elements. The first word specifies character width, or plot segment width for horizontal plots, i.e., a curve being plotted from left-to-right, or visa-versa. The second word specifies character height, or plot segment height for vertical plots.

When this parameter is used to specify character width and height, it is independent of character orientation. It is possible to use the DIMENSION parameter with the programmable font option to generate character fonts of a smaller size than eight (8) elements by twelve (12) lines.

DIMENSION width default value = 7.

DIMENSION height default value = 9.

3-23 Spacing Parameter (No. 9)



The SPACING parameter may be set via any normal instruction (except NOP). Its presence is flagged via Operand Flag Bit 9. The operand itself is two (2) 16-bit words in length and negative spacing may be expressed in 2's complement form.

For write text, spacing determines the distance between successive characters and the distance between successive lines. The X-displacement is always in the horizontal direction and the Y-displacement is always in the vertical direction. The scan operand (Operand Parameter 8) will determine which displacement, X or Y, is between characters and which is between lines. See Figure 3-5.

The positive direction for the X-displacement is to the right and the positive direction for the Y-displacement is down. If it is desired for either displacement to be in the opposite direction, then that displacement must be expressed as a

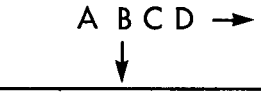
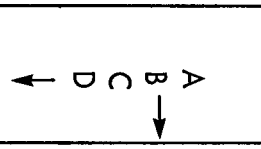
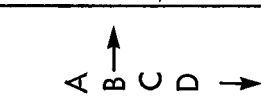
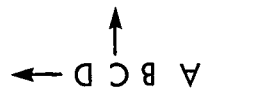
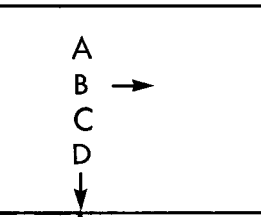
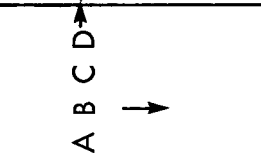
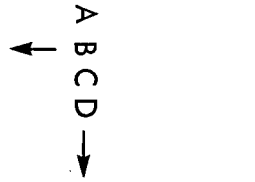
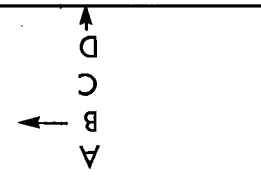
SCAN MODE	EXAMPLE	X SPACING	Y SPACING
0		Char Width	Char Height
1		- Char Height	Char Width
2		Char Height	- Char Width
3		- Char Width	- Char Height
4		Char Width	Char Height
5		Char Height	- Char Width
6		- Char Height	Char Width
7		- Char Width	- Char Height

Figure 3-5 Normal Spacing Values

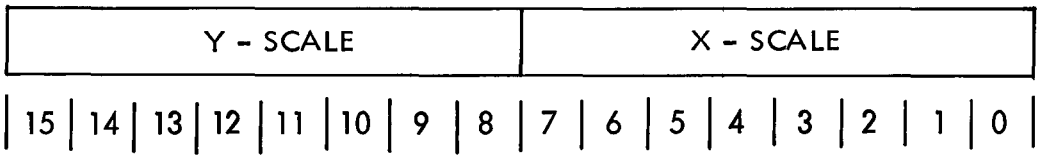
two's complement negative number. See Figure 3-5 for the normal spacing values for each scan mode.

For write plot, spacing defines the increment from plot entity to plot entity, along the plot axis. For horizontal plots, X-spacing is used to define the plot axis increment; for vertical plots, Y-spacing is used to define the plot axis increment.

X - SPACING default value = 7

Y - SPACING default value = 9

3-24 Scale Parameter (No. 10)



The SCALE parameter may be set via any normal instruction (Except NOP). Its presence is flagged via Operand Flag Bit 10. The operand itself is a single 16-bit word in length and specifies a scaling factor or ratio of received or generated picture elements to displayed picture elements for the WI, WR and WT instructions. When scaling text, X-Scale always refers to character width and Y-Scale always refers to character height before being subject to rotation via SCAN parameter. When writing a negatively scaled image where fewer picture elements will be displayed than will be received or generated, each displayed pixel will represent the arithmetic average of the corresponding received or generated pixels. When raster or text data is negatively scaled, the reduction process will just ignore N-1 of every N pixels where N is the scale factor.

The scale process is window oriented; the scaled results are stored in an internal buffer within the RM-9000 until a completely composed scan line(s) has been created. When the composed scan line is completed, it is written to refresh memory. Thus, if only a partial line of scaled image or raster data is output to the RM-9000, this data will be lost if:

- SCAN, WINDOW, or SCALE parameters are set in an ensuing normal-format instruction, or
- If a valid POPE instruction is issued.

Each scale factor is represented as an 8-bit 2's complement number. The received picture element ratio is defined in Table 3-9.

SCALE element ratio default value = 0.

SCALE line ratio default value = 0.

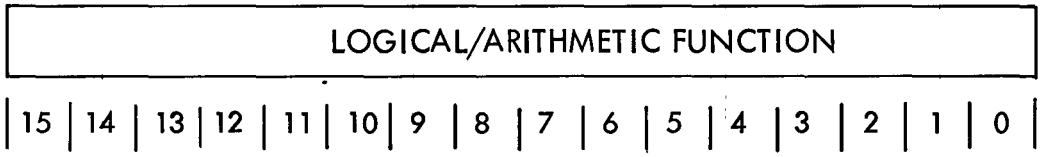
Table 3-9 Scaled Picture Ratio

SCALE	RATIO	PICTURE ELEMENTS	
		RECEIVED	DISPLAYED
-2	4:1	4	1
-1	2:1	2	1
0	1:1	1	1
1	1:2	1	2
2	1:4	1	4

NOTE

EITHER OR BOTH THE X AND Y AXIS MAY BE SCALED EITHER UP OR DOWN, HOWEVER, ONE (1) AXIS MAY NOT BE SCALED UP WHILE THE OPPOSITE AXIS IS SCALED DOWN.

3-25 Function Parameter (No. 11)



The FUNCTION parameter may be set by any normal format instruction (except NOP). Its presence is flagged by Operand Flag Bit 11. The parameter itself is a single 16-bit word in length and specifies a logical or arithmetic function (if any) to be performed in conjunction with the WRITE IMAGE instruction. By setting FUNCTION to a non-zero value, the WRITE IMAGE function is changed

from a write operation to a read-modify-write operation. Table 3-10 defines the legal function codes and their respective logical or arithmetic functions to be performed. All functions (except replacement) are performed using the old pixel value OP (i.e., the current value stored in refresh memory at the current pixel location) and the new pixel value NP (i.e., the pixel value loaded from the host processor). Before any function processing is performed, both the OP and NP are AND'ed with the current value of SUBCHANNEL; thus, only the subchannels of interest are subject to FUNCTION processing. The value resulting from FUNCTION processing RP is also AND'ed with SUBCHANNEL in the process of storing this pixel value in refresh memory.

The following is a description of each of the implemented logical/arithmetic processing functions. The following symbols are used in the ensuing section to describe the arithmetic/logical operations supported under the FUNCTION parameter:

RP	Pixel value resulting after processing specified by FUNCTION.
OP	Old pixel value (stored in refresh memory)
NP	New pixel value (input from the host computer via a WI instruction)
SC	Current value of the SUBCHANNEL parameter
$\leftarrow$	' is replaced by '
$\vee$	Logical inclusive OR
$\not\vee$	Logical Exclusive OR
$\wedge$	Logical AND
+	Addition
-	2's complement subtraction
>	' is greater than '
$\leq$	' is less than or equal to '
/	Division

Table 3-10 Logical/Arithmetic Function Codes

$CODE_{16}$	FUNCTION
0	NONE (DATA REPLACEMENT)
1	LOGICAL OR
2	LOGICAL XOR
3	LOGICAL AND
4	ARITHMETIC SUM
5	ARITHMETIC DIFFERENCE
6	GREATEST VALUE
7	LEAST VALUE
8	AVERAGE COMPUTATION
9	INVERSE ARITHMETIC DIFFERENCE
A	SIGN MAGNITUDE SUM
B	SIGN MAGNITUDE DIFFERENCE
C	SIGN MAGNITUDE INVERSE ARITHMETIC DIFFERENCE

FUNCTION 0 - REPLACEMENT

$RP \leftarrow NP \wedge SC$

The resulting pixel value is merely set equal to the new pixel value (i.e., normal WI processing is performed). The old pixel (OP) value is lost.

FUNCTION 1 - LOGICAL INCLUSIVE OR

$RP \leftarrow (NP \vee OP) \wedge SC$

The resulting pixel value is the logical inclusive OR of the old pixel (OP) value with the new pixel (NP) value.

FUNCTION 2 - LOGICAL EXCLUSIVE OR

$RP \leftarrow (NP \wedge SC) \vee (OP \wedge SC)$

The resulting pixel value is the logical exclusive OR of the old pixel (OP) value with the new pixel (NP) value.

FUNCTION 3 - LOGICAL AND

$RP \leftarrow (NP \wedge OP) \wedge SC$

The resulting value is the logical AND of OP and NP values.

FUNCTION 4 - ARITHMETIC SUM

$RP \leftarrow ((NP \wedge SC) + (OP \wedge SC)) \wedge SC$

The resulting pixel RP value is the arithmetic sum of NP and OP.

FUNCTION 5 - ARITHMETIC DIFFERENCE

$RP \leftarrow ((OP \wedge SC) - (NP \wedge SC)) \wedge SC$

The resulting pixel RP value is the 2's complement arithmetic difference of the old pixel OP value minus the new pixel NP value.

FUNCTION 6 - GREATEST VALUE

$RP \leftarrow OP \wedge SC$ if $(OP \wedge SC) - (NP \wedge SC) \wedge 0$

$NP \wedge SC$ if $(OP \wedge SC) - (NP \wedge SC) \wedge 0$

The resulting pixel RP value is set to either NP or OP whichever is greater. Since each value on input is masked by the subchannel mask value SUB-CHANNEL (default value of $OFFF_{16}$), the comparison becomes unsigned when Bit 15 of SC is zero.

FUNCTION 7 - LEAST VALUE

RP $OP \wedge SC$ if $(OP \wedge SC) - (NP \wedge SC) \leq 0$

$NP \wedge SC$ if $(OP \wedge SC) - (NP \wedge SC) \rightarrow 0$

The resulting pixel RP value is set to either NP or OP whichever is lesser in value. Since each value is masked by the subchannel mask value SUBCHANNEL (default value $OFFF_{16}$) on input, the comparison becomes unsigned when Bit 15 of SC is zero.

FUNCTION 8 - AVERAGE VALUE

$RP \leftarrow ((((OP \wedge SC) + (NP + SC))/2) \wedge SC)$

The resulting pixel RP value is the average value of OP and NP. The mechanism used to perform the division by 2 is such that rounding-up of the result does not occur, e.g., the average value of 0010_{16} and $000F_{16}$ is $000F_{16}$.

FUNCTION 9 - INVERSE ARITHMETIC DIFFERENCE

$RP \leftarrow ((NP \wedge SC) - (OP \wedge SC)) \wedge SC$

The resulting pixel RP value is the 2's complement arithmetic difference of the new pixel NP value minus the old pixel OP value.

In functions A_{16}, B_{16}, and C_{16}, sign-magnitude arithmetic is used. The sign-magnitude functions use the highest-order bit which is set to 1 in SUBCHANNEL (SC) as the sign bit. Negative numbers, therefore, have the same representation as positive numbers but with the sign bit set to 1. Since subchannels which do not exist read back as 1, it is crucial that the SUBCHANNEL parameter be set up to specifically include only those subchannels necessary for processing. In sign-magnitude arithmetic, if the sum of 2 positive numbers produces a carry into the sign bit, then the carry will be lost, i.e., the result will remain positive. If the sum of two (2) negative numbers produces a carry into the sign bit, then the result will remain negative.

EXAMPLE - Subchannel Mask = $0000001111110000 = 03F0_{16}$

$0020_{16} + 0210_{16} = 0010_{16}$ $\quad 2 + (-1) = 1$

$0220_{16} + 0210_{16} = 0230_{16}$ $\quad -2 + (-1) = 3$

$0020_{16} + 0240_{16} = 0220_{16}$ $\quad 2 + (-4) = 2$

Subchannel Mask = 0000000100001111 = $010F_{16}$

0002_{16} + 0101_{16} = 0001 2 + (-1) = 1

0102_{16} + 0101_{16} = 0103 -2 + (-1) = -3

0002_{16} + 0104_{16} = 0102 2 + (-4) = -2

FUNCTION A_{16} - ARITHMETIC SUM (SIGN MAGNITUDE)

RP ← ((OP ∧ SC) + (NP ∧ SC)) ∧ SC (Sign-Magnitude Sum)

The resulting pixel (RP) value is the sum of the old pixel (OP) value and the new pixel (NP) value. Both (NP) and (OP) are evaluated as sign-magnitude numbers using the SUBCHANNEL parameter to define the sign bit.

FUNCTION B_{16} - ARITHMETIC DIFFERENCE (SIGN MAGNITUDE)

RP ← ((OP ∧ SC) - (NP ∧ SC)) ∧ SC (Sign-Magnitude Difference)

The resulting pixel (RP value is the difference of the old pixel (OP) value minus the new pixel (NP) value. Both (NP) and (OP) are evaluated as sign-magnitude numbers using the SUBCHANNEL parameter to define the sign bit.

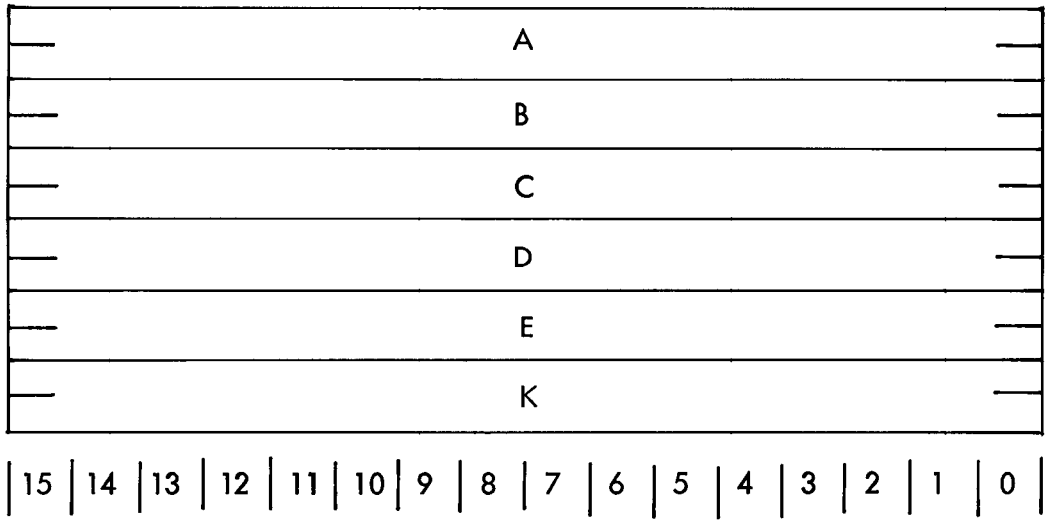
FUNCTION C_{16} - INVERSE ARITHMETIC DIFFERENCE (SIGN MAGNITUDE)

RP ← ((NP ∧ SC) - (OP ∧ SC)) ∧ SC (Sign-Magnitude Difference)

The resulting pixel (RP) value is the difference of the new pixel (NP) value minus the old pixel (OP) value. Both (NP) and (OP) are evaluated as sign-magnitude numbers using the SUBCHANNEL parameter to define the sign bit.

FUNCTION default value = 0 (Replacement Mode)

3-26 Conic Equation (Parameter (No. 12)



The CONIC EQUATION parameter operands may be set by any normal format instruction (except NOP). Its presence is flagged by Operand Flag Bit 12. This parameter consists of six double-word arguments A, B, C, D, E and K. The first word of each double-word pair is the high-order word. The second word is the low-order word. In the current implementation of RM-CON, the high-order word is not used, and was included to allow for possible future expansion to higher resolution display systems.

The arguments A, B, C, D and E are 16-bit 2's complement integers which are used as the generalized conic equation coefficients. The generalized form of the conic equation is:

$$Ax^2 + By^2 + Cxy + Dx + Ey = 0.$$

The argument K represents the total number of pixels to be generated for the conic described by A, B, C, D and E. For example, a circle of radius 100_{10} pixels would have a value of 400_{10} for K.

For a detailed description of the use of the CONIC-EQUATION parameter operand in conjunction with the WC instruction, see Section 3-51 and Appendix B.

CONIC-EQUATION default values:

A = 0	D = 0
B = 0	E = 0
C = 0	K = 1280_{10}

3-27 Baseline Parameter (No. 13)

FILLED PLOT BASELINE															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The BASELINE parameter may be set via any normal format instruction (except NOP). Its presence is flagged by Operand Flag Bit 13. The operand itself is a 16-bit word in length and specifies whether a filled plot or a line plot is to be drawn. When BASELINE is zero, a line plot (i.e., a plot in which each endpoint along the curve becomes the start point for the succeeding plot segment) is drawn. When BASELINE is non-zero, the BASELINE defines the start point for each plot segment. If SCAN is between 0 and 3, then BASELINE defines the horizontal axis to which the filled-plot segments will be drawn, i.e., BASELINE defines a Y-address. Similarly, if SCAN is between 4 and 7, then BASELINE defines the vertical axis to which the filled-plot segments will be drawn, i.e., BASELINE defines an X-address.

DEFAULT BASELINE VALUE : 0

3-28 Scroll-Count Parameter (No. 14)

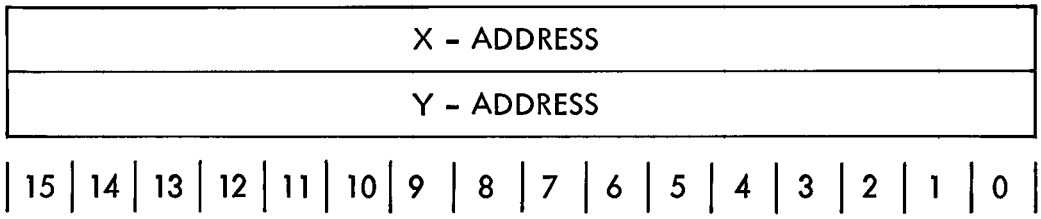
SCROLL COUNT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The SCROLL-COUNT parameter may be set by any normal-format instruction (except NOP). Its presence is flagged by Operand Flag Bit 14. The operand itself is a single 16-bit word in length and specifies scroll count and direction

i.e., left or right for SCRX, or up or down for SCRY. When set to a negative value, scroll will occur in the negative direction. That is, the image will be scrolled left for SCRX and up for SCRY. When set to a positive value, the image will be scrolled right for SCRX and down for SCRY. The scroll count is a 2's complement 16-bit number.

DEFAULT SCROLL-COUNT VALUE : 0

3-29 Start-Point Argument (COP) (No. 15)



The START-POINT argument may be set by any normal-format instruction (except NOP). Its presence is flagged by Operand Flag Bit 2^{15}. The operand itself is two (2) 16-bit words in length and specifies a start-point for the WI, RI, WR, WT, WV, WC, and WP instructions. For MAG, the COP determines the pixel to become the new center point of the window. Note that when WINDOW or SCAN are specified, the appropriate start-point is automatically calculated for the WI, RI, WR, and WT instructions and, therefore, need not be specified unless the writing process is to begin in other than the appropriate corner of the WINDOW. Table 3-8 defines the default values for START-POINT based on the setting of WINDOW and SCAN. Because WINDOW is not pertinent to the WV, WC, and WP instructions, START-POINT must be specified. Otherwise, the last end-point (current COP) will be used as the new start point. The START-POINT explicitly sets the Current Operating Point (COP) according to the addressing mode specified by the instruction containing the parameter. For window oriented commands, the value specified must be within the area specified by the WINDOW parameter for proper operation.

DEFAULT X START POINT VALUE : 0

DEFAULT Y START POINT VALUE : 0

3-30 Data Length Word

The Data Length Word defines the number of bytes of data for any normal format instruction. It defines either the number of bytes to be read by the RM-9000 for the NOP, SET, ERS, WI, WT, WR, WV, WC, WP, SCRX, SCRY instructions or the number of bytes to be read by the host computer for the RI instruction. The Data Length Word will be present immediately after the opcode word or any normal instruction parameters which may be present if Bit 0 of the Parameter Byte (i.e., the DATA FLAG) is set to one (1). The data length word may take on a maximum value of 65535_{10} Bytes represented by an unsigned 16-bit number.

3-31 Normal-Format Instruction Data

If the Data Flag (Bit 0 of the parameter byte) is 1, the n bytes of data as specified in the Data Length Word will be transferred immediately following the Data Length Word in any normal format instruction. The format of this data varies from instruction to instruction. Five data types are possible for the normal instruction set:

- Image data for the WI and RI instructions
- Text data for the WT instruction
- Plot data for the WP instruction
- Endpoint data for the WV and WC instructions
- Raster data for the WR instruction.

All other normal format instructions will read in the indicated number of bytes of data and subsequently discard them.

ramtek

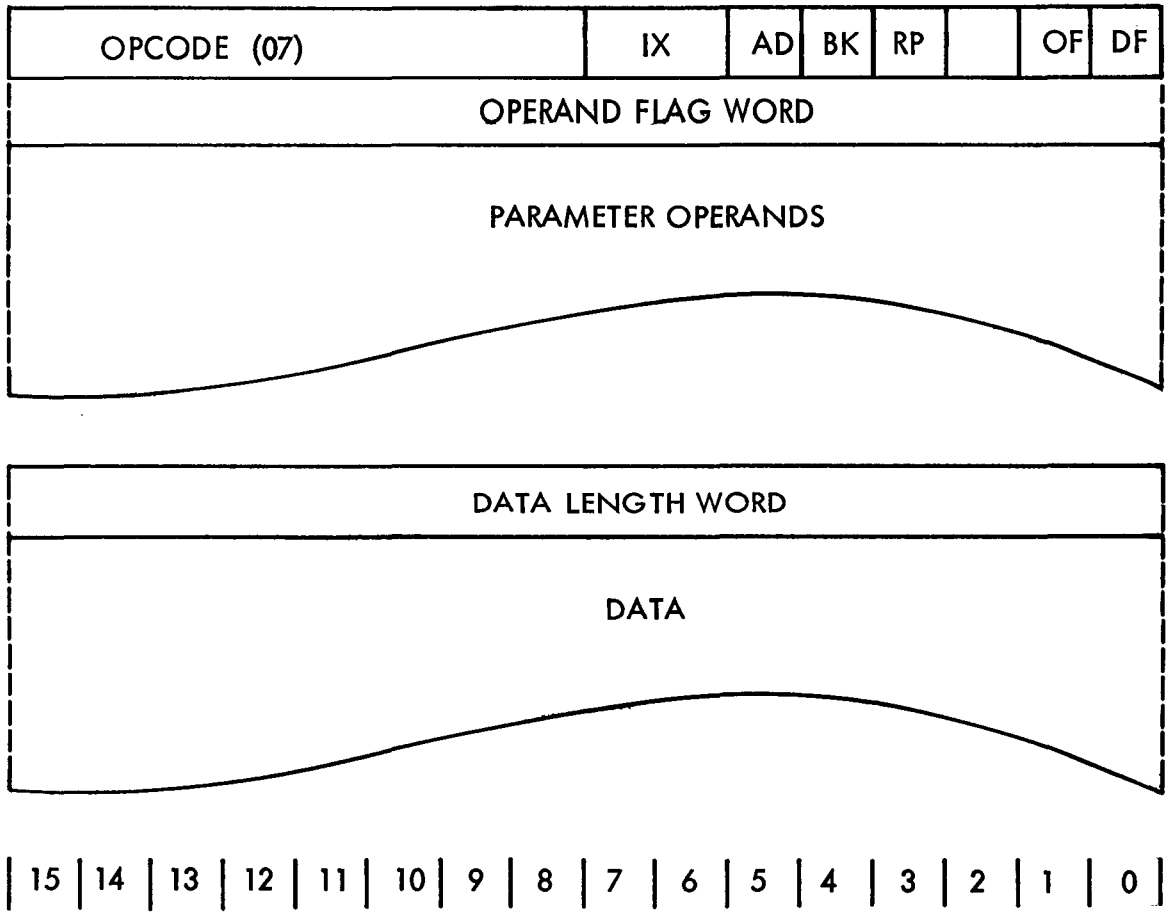
3-32 Normal-Format Standard Instructions

Sections 3-33 through 3-38 define the formats and functions of the normal-format instructions which are supported by the RM-9000 standard firmware package. The instructions in this set are:

INOP	-	No Operation Instruction
SET	-	Set Parameter Instruction
ERS	-	Erase Instruction
WI	-	Write Image Instruction
RI	-	Read Image Instruction
WT	-	Write Text Instruction

These instructions are standard in all RM-9000 systems.

3-33 No-Operation Instruction (INOP)



The INOP instruction is a normal-format instruction which is included in the RM-9000 standard firmware. The INOP instruction performs no internal functions whatsoever; any parameter operands or data which are present in the instruction stream are discarded. The INOP instruction is useful in facilitating the debugging of instruction streams passed from the host processor to the RM-9000. By simply changing the opcode byte of any normal-format instruction or of any single word special-format instruction, the integrity of the instruction stream can be maintained while selectively eliminating the effects of one or more instructions.

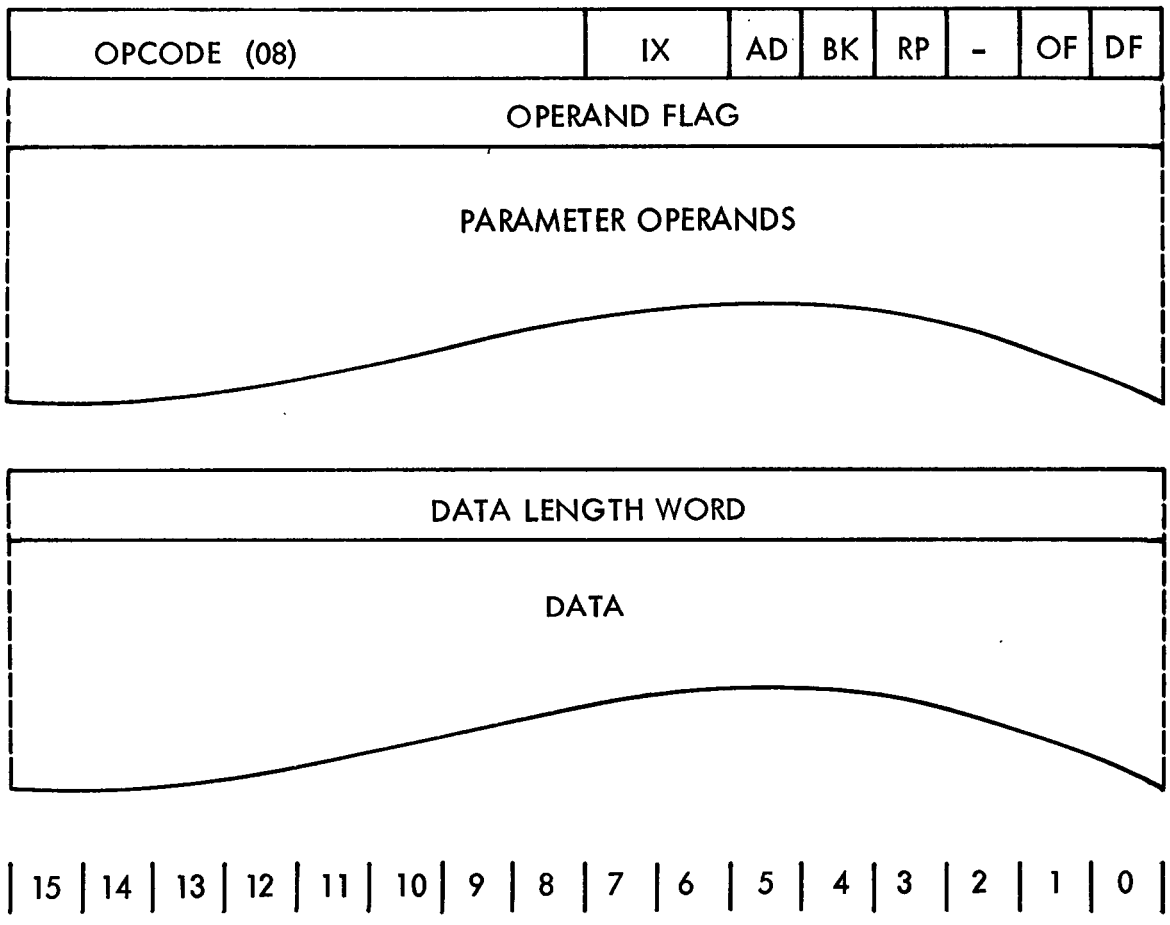
PERTINENT ARGUMENTS

None

POSSIBLE ERROR CONDITIONS

None

3-34 Set Parameter Instruction (SET)



The SET instruction is a normal-format instruction which is included in the RM-9000 standard firmware. The SET instruction allows all parameter operands to be defined (based in some cases on the addressing mode) as in any normal-format instruction, but any data which is present in the instruction stream will be ignored. This instruction is included in the standard firmware in order to facilitate debugging of a display instruction stream. It allows a user to perform parameter operand processing whose internal modifications could carry over to the subsequent instructions in the instruction stream, while ignoring the received data.

PERTINENT CONTROL BITS

IX Defines the address mode in which the INDEX 1, INDEX 2, ORIGIN, WINDOW, BASELINE and STARTPOINT will be evaluated.

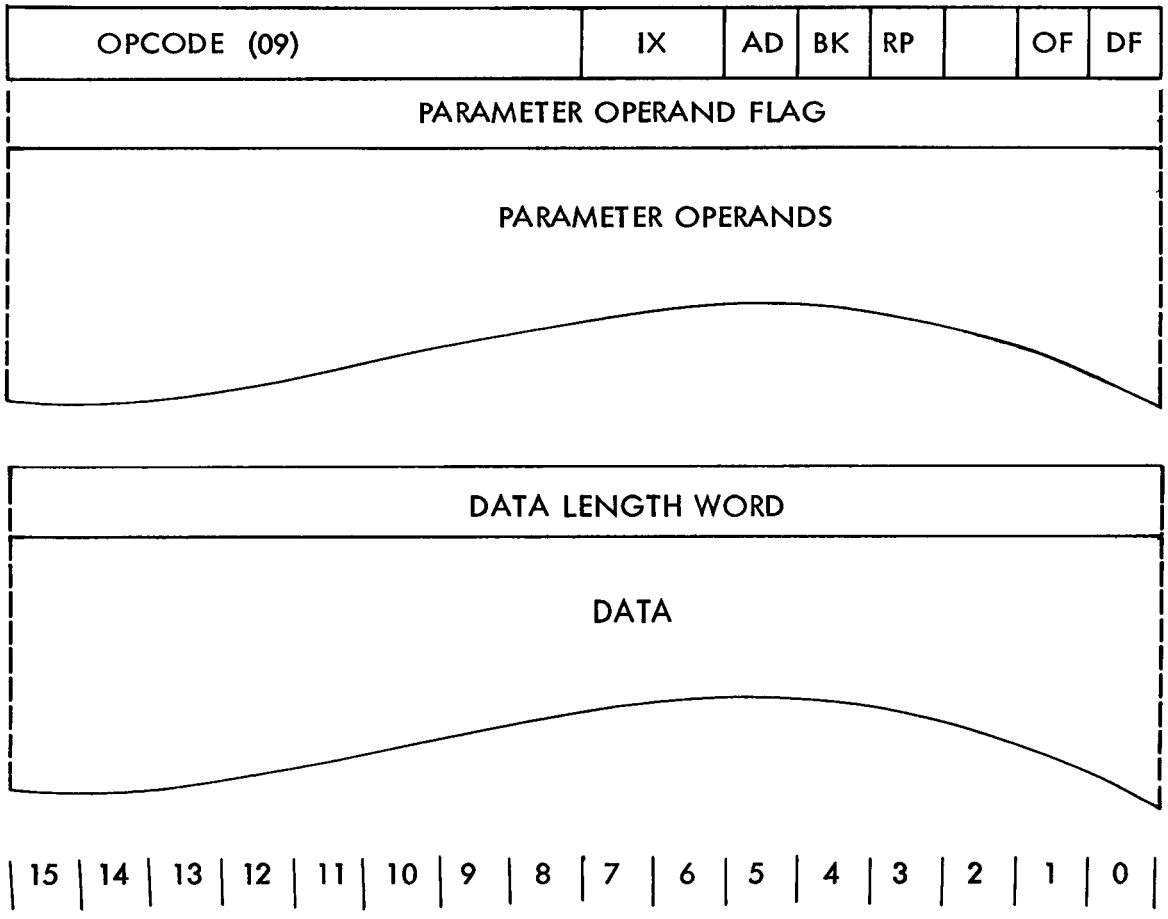
PERTINENT PARAMETER OPERANDS

All parameter operands are pertinent to the SET instruction in the sense that all parameter operands may be re-defined; however, since SET performs no display functions, none of the arguments are pertinent to the display of data in an immediate sense.

DATA FORMAT

Since all data is discarded, data format is irrelevant.

3-35 Erase Instruction (ERS)



The ERS instruction is a normal-format instruction which is included in the RM-9000 standard firmware. The ERS instruction sets the rectangular area in refresh memory defined by the WINDOW parameters to either the FOREGROUND or BACKGROUND value based on the value of BK. If BK = 0, the BACKGROUND value will be used; otherwise if BK = 1, the FOREGROUND value will be used. Regardless of FOREGROUND/BACKGROUND usage, the selected value will be write-masked by the SUBCHANNEL parameter operands. The Cartesian data generated by the ERS instruction uses the internal mode also used by the text and the raster processors, the AD flag can affect ERS processing. If AD = 1, then no Cartesian data will be written into refresh (See Table 3-3 of Section 3-10) with a data value of 0. All data associated with an ERS instruction will be discarded.

PERTINENT CONTROL BITS

IX Defines the address mode in which the WINDOW parameter operands will be evaluated, and as such, affects the rectangular area to be used.

AD Affects the generation of Cartesian data such that if AD = 1, no data will be written to refresh memory.

BK Defines the color or intensity value to be used, i.e., if BK = 0, the BACKGROUND value is used; otherwise, the FOREGROUND value is used.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL	Defines the subchannel write-enable mask; only those subchannels whose corresponding bit in the SUBCHANNEL mask will be written with data from the ERS instruction.
FOREGROUND	Defines the color or intensity value to be stored in refresh memory when BK = 1.
BACKGROUND	Defines the color or intensity value to be stored in refresh memory when BK = Ø.
INDEX 1	Displaces the WINDOW parameters when WINDOW is set in the ERS instruction and IX = 1.
INDEX 2	Displaces the WINDOW parameters when WINDOW is set in the ERS instruction and IX = 2.
WINDOW	Defines the rectangular area to be erased.

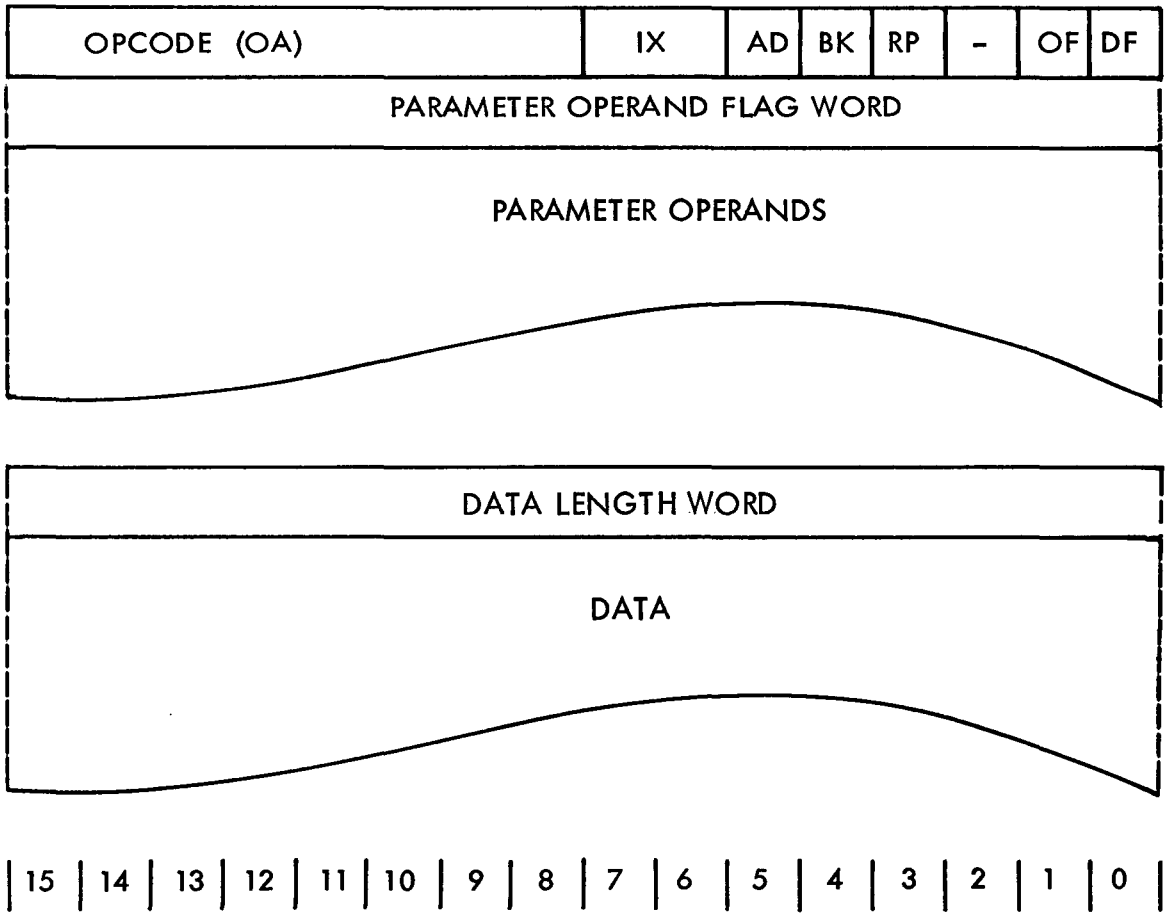
DATA FORMAT

Since all data is discarded, data format is irrelevant.

POSSIBLE ERROR CONDITIONS

If the WINDOW parameter values are outside of the system resolution, the resulting Cartesian data will be indeterminate.

3-36 Write Image Instruction (WI)



The WI instruction is a normal-format instruction which is included in the RM-9000 standard firmware. The WI instruction stores up to 32767_{10} words of data in refresh memory on a word-per-pixel basis within the rectangular area defined by WINDOW. The first 16-bit word of image data received from the host processor is stored in refresh memory at the pixel defined by the START-POINT parameter operand or by the current operating point which was the result of the previous instructions if no START-POINT value is specified in the WI instruction. The successive words of image data are stored in refresh memory based on the primary update mode associated with SCAN (See Section 3-21).

When a window boundary is encountered while storing image data words in successive pixels based on the primary update direction, the current operating point is set to the opposite window boundary and incremented in a direction

perpendicular to the primary scan direction (i.e., the secondary update direction defined by SCAN). When a window boundary is encountered in the process of incrementing in the secondary update direction, the current operating point is repositioned to the opposite window boundary.

Since the data to be written to the refresh memory is supplied directly from the host processor and not from the FOREGROUND or the BACKGROUND parameter operand, the AD and BK control flags are ineffective in image mode. Actually, setting BK=1 causes the value of the FOREGROUND register to be written **into memory** in place of the pixel value sent from the host CPU. Thus the proper use of WI **requires** that BK=0.

PERTINENT CONTROL BITS

BK	If BK=0, the BACKGROUND value is used; if BK=1, the FOREGROUND value is used.
IX	Defines the address mode in which the INDEX 1, INDEX 2, WINDOW and START-POINT parameter operands set by any given WI instruction, are evaluated.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL	Specifies the subchannels (i.e., bit planes in refresh memory) which are write-enabled and which will receive image data.
INDEX 1	Displaces the values to be used for WINDOW and START-POINT parameter operands set in the WI instruction when IX = 1.
INDEX 2	Displaces the values to be used for WINDOW and START-POINT parameter operands set in the WI instruction when IX = 2.
WINDOW	Defines the rectangular area into which image data will be written. In addition, if the START-POINT parameter is not explicitly set in the WI instruction, WINDOW along with SCAN defines the starting pixel coordinates (See Section 3-21).
SCAN	Defines the primary and secondary update directions as well as the starting pixel coordinates when START-POINT is not explicitly defined in a WI instruction.
SCALE	Defines the ratio of pixels written in refresh memory to the number of image data words received from the host processor (See Section 3-24).

FUNCTION — Defines one of thirteen (13) possible image processing modes to be applied as each pixel is stored in refresh memory. A value of Ø defines a write with no processing operation; while the other twelve (12) non-zero functions represent a read-modify-write operation per pixel (See Section 3-25).

START-POINT — Specifies the coordinates of the first pixel to be written with image data; if not defined, the current operating point which was the result of the previous instruction is used as the starting pixel coordinate.

DATA LENGTH WORD

The DATA LENGTH WORD represents the number of bytes of data to be transmitted from the host processor to the RM-9000 with a WI instruction. Since image data is defined on a word basis, the DATA LENGTH WORD should always reflect an even number of bytes. If the DATA LENGTH WORD is odd, the byte count used will be one less than the actual byte count stored, e.g., if the DATA LENGTH WORD had a byte count value of 33_{10} bytes, 32_{10} bytes or 16_{10} words would be expected by the RM-9000. The range of the DATA LENGTH WORD is from 0 through 65534_{10} bytes or 32767_{10} words.

DATA FORMAT

Data in image mode is interpreted on a word basis. Of the 16-bits of data per word, only the low-order twelve (12) are actually used when no logical/arithmetic FUNCTION processing is to be performed. Assuming that all subchannels have been write-enabled via SUBCHANNEL, each bit of the incoming data word is written to its corresponding subchannel, i.e., data Bit 0 to Subchannel 0, Data Bit 1 to Subchannel 1, ..., Data Bit 11 to Subchannel 11. When FUNCTION processing is to be performed, the incoming data is and'ed with SUBCHANNEL before use, therefore all 16-bits of image data could be significant if the value of SUBCHANNEL enabled bits in the upper four (4) positions (See Section 3-25 on FUNCTION processing).

COP MOVEMENT

The resulting current operating point after the completion of a WI instruction is the coordinates of the next pixel which would have been written if (N + 1) words of data had been passed in the instruction rather than N words. Since

the primary and secondary update directions are defined by SCAN, the resulting COP position is a function of SCAN and the number of words of image data written. For further discussion of the COP movement, see Example 3-1.

POSSIBLE ERRORS

If the DATA LENGTH WORD is odd, it is possible for an unsuspecting user to get out of synchrony with the RM-9000, i.e., interpret a word of data as an instruction opcode word.

EXAMPLE 3-1

This example demonstrates the use of the WI instruction and the operation of SCAN and WINDOW in conjunction with this instruction. The following instruction stream stores data in a rectangular window in refresh memory and the associated Figures 3-6 a through h, indicate the resulting displays. The window is defined by the coordinate corners (100,100) and (103,103). Sixteen (16) words of data (designated by the symbols D1 through D16) will be written into the rectangular area defined by the WINDOW parameter operand:

HEX	MNEMONIC	DESCRIPTION
0500	RESET	; Clear Screen
0A03	WI+OF+DF	; Write Image Instruction
00C0	SCN+WIN	; Operand Flag Word
XXXX	XXXX	; Scan Value from 0 through 7
0064	100	; Element Left Margin = 100_{10}
0064	100	; Line Top Margin = 100_{10}
0067	103	; Element Right Margin = 103_{10}
0067	103	; Line Bottom Margin = 103_{10}
0020	32	; Data Length Word = 32_{10}
	D1	;
	D2	; 32 bytes (16 words) of data
	⋮	;
	D16	;

Where XXXX defines the scan mode from 0 through 7.

Notice that the current operating point (COP) after all of the image transfers are complete is identical to the starting COP, since after data value D16 was transferred both primary and secondary updates are exercised.

If the DATA LENGTH WORD value was increased to 34_{10} and a seventeenth data value D17 added, the primary and secondary updates return the COP to the starting coordinates and over-write the pixel at (100,100) with D17.

100	101	102	103	
D1	D2	D3	D4	100
D5	D6	D7	D8	101
D9	D10	D11	D12	102
D13	D14	D15	D16	103

(a) SCAN MODE = 0
START X = 100
START Y = 100

100	101	102	103	
D4	D3	D2	D1	100
D8	D7	D6	D5	101
D12	D11	D10	D9	102
D16	D15	D14	D13	103

(b) SCAN MODE = 1
START X = 103
START Y = 100

100	101	102	103	
D13	D14	D15	D16	100
D9	D10	D11	D12	101
D5	D6	D7	D8	102
D1	D2	D3	D4	103

(c) SCAN MODE = 3
START X = 100
START Y = 103

100	101	102	103	
D16	D15	D14	D13	100
D12	D11	D10	D9	101
D8	D7	D6	D5	102
D4	D3	D2	D1	103

(d) SCAN MODE = 3
START X = 103
START Y = 103

100	101	102	103	
D1	D5	D9	D13	100
D2	D6	D10	D14	101
D3	D7	D11	D15	102
D4	D8	D12	D16	103

(e) SCAN MODE = 4
START X = 100
START Y = 100

100	101	102	103	
D4	D8	D12	D16	100
D3	D7	D11	D15	101
D2	D6	D10	D14	102
D1	D5	D9	D13	103

(f) SCAN MODE = 5
START X = 100
START Y = 103

100	101	102	103	
D13	D9	D5	D1	100
D14	D10	D6	D2	101
D15	D11	D7	D3	102
D16	D12	D8	D4	103

(g) SCAN MODE = 6
START X = 103
START Y = 100

100	101	102	103	
D16	D12	D8	D4	100
D15	D11	D7	D3	101
D14	D10	D6	D2	102
D13	D9	D5	D1	103

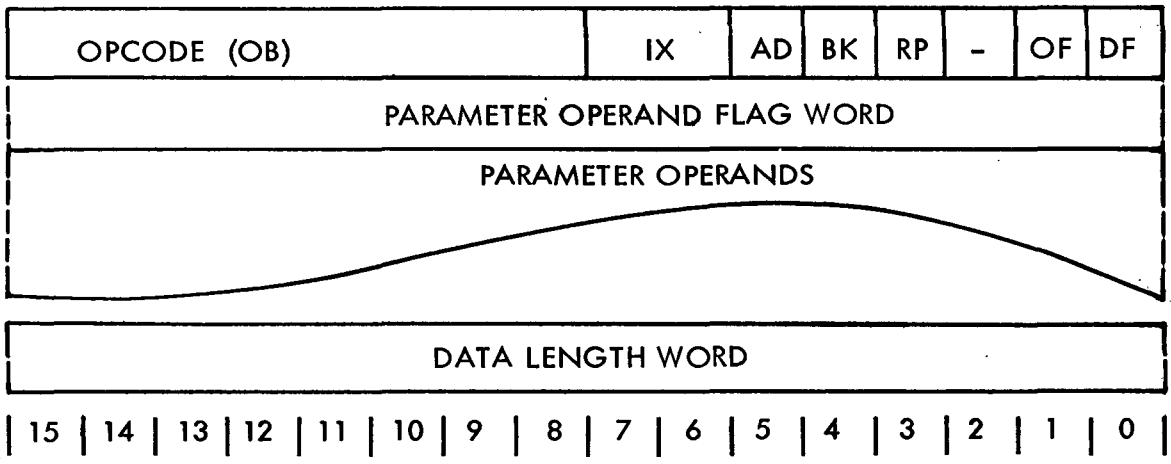
(h) SCAN MODE = 7
START X = 103
START Y = 103

100	101	102	103	
D17	D2	D3	D4	100
D5	D6	D7	D8	101
D9	D10	D11	D12	102
D13	D14	D15	D16	103

(i) SCAN MODE = 0
START X = 100
START Y = 100

Figure 3-6 Imaging Scan Example

3-37 Read Image Instruction (RI)



The RI instruction is a normal format instruction which is included in the RM-9000 standard firmware. The purpose of the Read Image instruction is to read back the contents of the refresh memory pixel-by-pixel into the host computer. Like all such "read-request" instructions, the RI instruction is executed in two parts:

1. output the RI instruction terminating with the data length word; and
2. input the image data, 1 pixel per 16-bit word, until the byte count specified by the data length word is satisfied.

Please note that the last word of the RI instruction MUST be the data length word specifying the number of bytes to be read back --- or no data is output. Likewise note that the RI instruction itself does not cause any data to be read back from the display system. Readback of data occurs only when the programmer initiates a data input transfer. The RI transfers up to 32767_{10} words of refresh memory data to the host processor on a word-per-pixel basis from the rectangular area defined by WINDOW. The first 16-bit word of image data transferred from refresh memory to the host is obtained from the pixel defined by the START-POINT parameter operand or by the current operating point which was the result of the previous instruction(s) if no START-POINT value is specified in the RI instruction. The successive words of image data are read back from refresh memory based on the primary update mode associated with SCAN (See Section 3-21).

When a window boundary is encountered while reading image data words from successive pixels based on the primary update direction, the current operating point is set to the opposite window boundary and incremented in a direction perpendicular to the primary scan direction. When a window boundary is encountered in the process of incrementing in the secondary update direction, the current operating point is repositioned to the opposite window boundary.

Since the data to be written to the host processor is supplied directly from the refresh memory and not from the FOREGROUND or the BACKGROUND parameter operand, the AD and BK control flags are ineffective in image mode (setting BK=1 will cause the value of the FOREGROUND register to be loaded into refresh memory in place of the incoming pixel value).

When data is read back from the refresh memory using the RI instruction, the SUBCHANNEL write-enable mask is not effective. All subchannels are passed back to the host processor. An RM-9000 can have any number of subchannels up to twelve (12) and the data which is read back from subchannels which are not present in the configuration is indeterminate. The high order 4 bits of the read back data value will read back to the host processor as 1. Thus, if refresh memory has been erased to zeros, all data words would read back as $F000_{16}$ in a 12-subchannel system.

PERTINENT CONTROL BITS

IX	Defines the address mode in which the INDEX 1, INDEX 2, WINDOW and START-POINT parameter operands set by any given RI instruction, are evaluated.

PERTINENT PARAMETER OPERANDS

INDEX 1	Displaces the values to be used for WINDOW and START-POINT parameter operands set in the RI instruction when IX = 1.
INDEX 2	Displaces the values to be used for WINDOW and START-POINT parameter operands set in the RI instruction when IX = 2.
WINDOW	Defines the rectangular area from which image data will be read back. In addition, if the START-POINT parameter is not explicitly set in the RI instruction, WINDOW along with SCAN defines the starting pixel coordinates (See Section 3-21).
SCAN	Defines the primary and secondary update directions as well as the starting pixel coordinates when START-POINT is not explicitly defined in the RI instruction.
START-POINT	Specifies the coordinates of the first pixel to be written with image data; if not defined, the current operating point which was the result of the previous instruction is used as the starting pixel coordinate.

DATA LENGTH WORD

The DATA LENGTH WORD represents the number of bytes of data to be transmitted from the RM-9000 to the host processor with a RI instruction. Since image data is defined on a word basis, the DATA LENGTH WORD should always reflect an even number of bytes. If the DATA LENGTH WORD is odd, the byte count used will be one less than the actual byte count stored, e.g., if the DATA LENGTH WORD had a byte count value of 33_{10} bytes, 32_{10} bytes or 16_{10} words would be expected from the RM-9000. The range of the DATA LENGTH WORD is from 0 through 65534_{10} bytes or 32767_{10} words.

DATA FORMAT

Data in image mode is transmitted on a word basis. Of the 16-bits of data per word, only the low-order 12 actually carry refresh memory data. The low-order 12-bits correspond to subchannels such that Bit 2^0 represents subchannel 0, Bit 2^1 represents Subchannel 1. Subchannels which do not exist read back indeterminately. Bit 2^{12} through 2^{15} of the read back data value will always be set to 1.

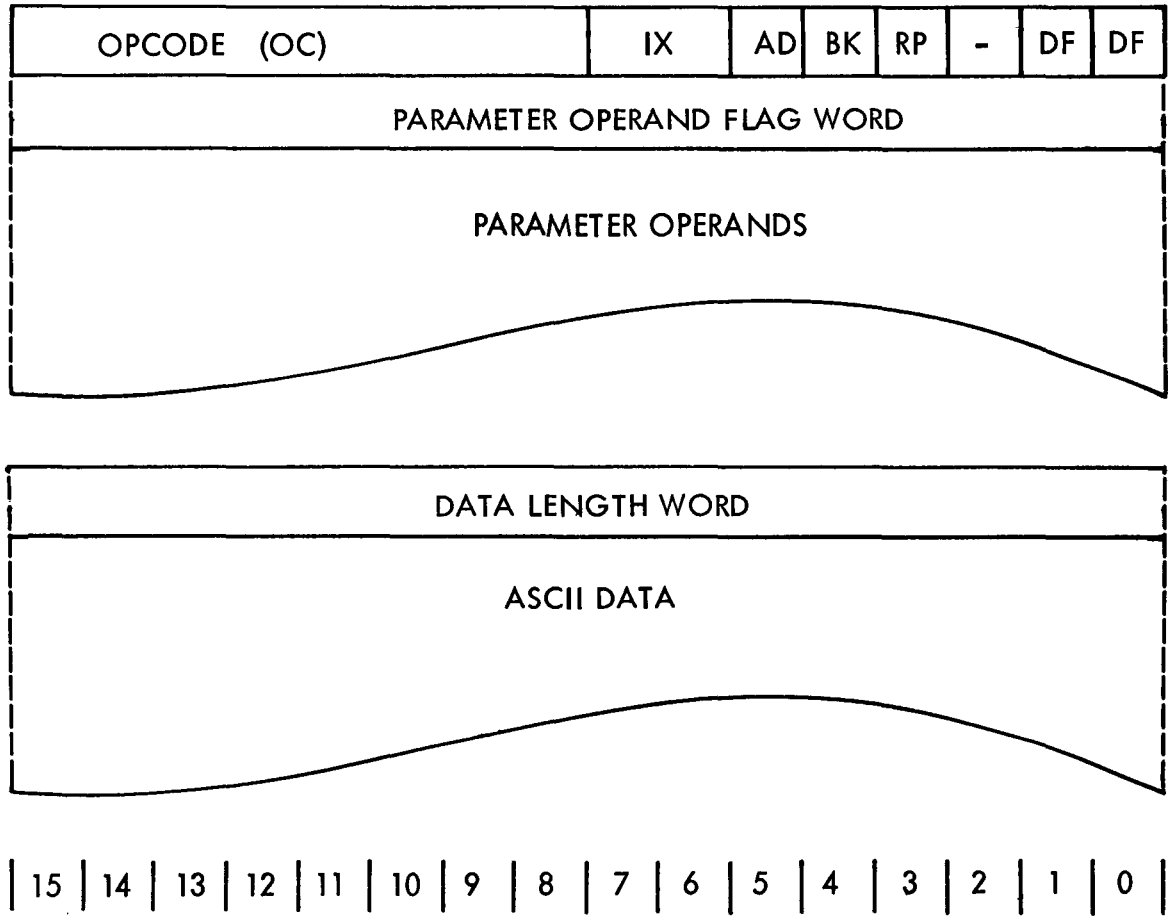
COP MOVEMENT

The resulting current operating point after the completion of an RI instruction is the coordinates of the next pixel which would have been read back if (N + 1) words of data had been passed in the instruction rather than N words. Since the primary and secondary update directions are defined by SCAN, the resulting COP position is a function of SCAN and the number of words of image data read back. For further discussion of the COP movement, see Example 3-1.

POSSIBLE ERRORS

If the DATA LENGTH WORD is odd, it is possible for the unsuspecting user to get out of synchrony with the RM-9000, i.e., both systems will be attempting to read from each other and a hard system reset will be necessary.

3-38 Write Text Instruction (WT)



The WT instruction is a normal format instruction included in the RM-9000 standard firmware. The WT instruction reads ASCII character codes from host processor and generates the corresponding text characters on to a rectangular area defined by the WINDOW parameter operand. The standard 64_{10} ASCII character codes (20_{16} through $5F_{16}$) are available in a system without the RM-FNT programmable font option (See Appendix B). If the RM-FNT programmable font option is installed, the character font descriptions for codes 20_{16} through $5F_{16}$ are copied to the programmable font option internal storage upon RESET (both hardware reset and the RM-9000 RESET instruction). Character codes 20_{16} through $9F_{16}$ are able to be loaded by the host processor via the LPF and LPFRP instructions (See Sections 3-60 & 3-61). Character codes 0 through $1F_{16}$ are illegal (except for carriage return OD_{16} and line feed OA_{16}) and will generate a character of undefined font.

The actual font size of a character is defined by the DIMENSION parameter operands. This is the size of the rectangle which will have text/font data stored in refresh memory. In a standard system without the RM-FNT option, the values of DIMENSION should never exceed its default values of 7 pixels wide and 9 pixels high. If DIMENSION is set to values less than 7 by 9, the right-most and bottommost portions of the 7 by 9 font will be lost, respectively. In a system with the RM-FNT option, the maximum values for the DIMENSION parameter operand are 8 pixels wide and 12_{10} pixels high. This is the font size which is able to be loaded by the host processor. The spacing which is performed between characters within the rectangular window is defined by the SPACING operand parameters. Since the SPACING operand parameter is defined in terms of X-spacing and Y-spacing, in order to get proper updating between characters it is necessary to change the SPACING parameter in the case where SCAN is changed (See Figure 3- 5) for the normal spacing values in text mode.

Although the WINDOW parameter operand defines the rectangular region for text generation, it is possible to generate part of a character outside of this area. If the first pixel of any character to be generated is within the WINDOW region, the entire character will be generated. Therefore, it is possible for up to (DIMENSION width - 1) pixels to be generated outside of the rectangular WINDOW region.

The carriage return character (Code OD_{16}) will cause an immediate return to the window boundary in the direction opposite to the primary update direction as well as an update in the secondary update direction as well as an update in the secondary update direction. The line feed character (Code OA_{16}) will cause only an immediate update in the secondary update direction. When the RM-FNT option is not installed, all characters in the ranges 00 through $1F_{16}$ and 60_{16} through FF_{16} are treated as legal characters in the sense that a character is generated into refresh memory, however, the font which is used for these characters is indeterminate. When the RM-FNT option is installed, all characters (other than carriage return or line feed) in the ranges 00 through $1F_{16}$ and AO_{16} through FF_{16} are generated in the same manner.

PERTINENT CONTROL BITS

IX — Defines the address mode in which the parameter operands INDEX 1, INDEX 2, ORIGIN, WINDOW, BASELINE and START-POINT will be evaluated and converted to absolute refresh memory addresses.

AD	Affects the generation of text data in conjunction with the BK flag. Characters are generated using a font consisting of one's and zeros. The AD flag inhibits the writing of pixels within the character font in a manner specified by Table 3-15 (See Section 3-10).
BK	Defines the selection of FOREGROUND and BACKGROUND colors based on the ones/zeros data bits within the character font and on the value of the AD flag (see above). If BK = 0, the FOREGROUND value is selected for 'ones' font data and the BACKGROUND value is selected for 'zeros' font data.
RP	Defines the byte-accessing order for ASCII characters within each 16-bit word. If RP = 0, the characters will be accessed high-byte first; if RP = 1, the low byte will be accessed before the high byte.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL	Specifies the subchannels (i.e., bit planes in refresh memory) which are write-enabled and which will receive text data.
FOREGROUND	Defines the foreground register color or intensity value to be written to refresh memory, in combination with the BACKGROUND operand and the AD and BK flags (See Table 3-3).
BACKGROUND	Defines the background register color or intensity value to be written to refresh memory, in combination with the FOREGROUND operand and the AD and BK flags (See Table 3-3).
INDEX 1	Displaces the values to be used for INDEX 2, WINDOW, ORIGIN and START-POINT parameter operands set in the WT instruction when IX = 01.
INDEX 2	Displaces the values to be used for WINDOW and START-POINT parameter operands set in the WT instruction when IX = 10.
WINDOW	Defines the rectangular area into which text will be written. If the first pixel of any character is within this region, the

entire character will be drawn, even though part of the character may exceed the WINDOW boundaries.

SCAN — Defines the primary and secondary text update directions as well as the pixel coordinate when START-POINT is not explicitly defined in the WT instruction (See Figure 3-5).

DIMENSION — Defines the character font size independent of SCAN direction or character orientation.

SPACING — Defines the character-to-character increments for the primary and secondary text updates. (See Figure 3-5) for legal SPACING values for the various text modes.

SCALE — Defines the received-pixel-to-displayed-pixel ratio when the RM-SCA option is installed (See Table 3-9).

START-POINT — Specifies the coordinates of the first pixel of the first text character to be written into refresh memory. START-POINT must be contained within the current WINDOW region.

DATA LENGTH WORD

The DATA LENGTH WORD represents the number of ASCII characters which will be transmitted from the host processor to the RM-9000 with a WT instruction on a one (1) byte per character basis. Since text data is byte-oriented, the DATA LENGTH WORD may take on any value from 0 to 65535_{10}.

DATA FORMAT

Data in text mode is interpreted on a byte basis, i.e., two (2) characters per word. The order that the bytes are referenced is determined by the RP flag. When the RM-FNT option is not installed, the legal character set consists of Codes 20_{16} through $5F_{16}$ and OA_{16} (line feed) and OD_{16} (carriage return). When the RM-FNT option is installed, the legal character set consists of Codes 20_{16} through $9F_{16}$ as well as OA_{16} and OD_{16}.

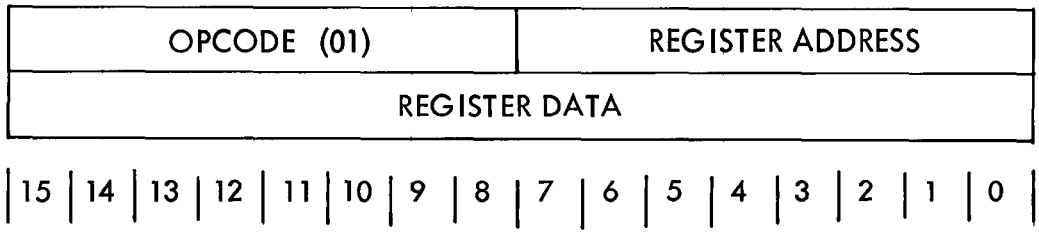
3-39 Special-Format Standard Instructions

Sections 3-40 through 3-45 define the formats and functions of the special-format instructions which are supported by the RM-9000 standard firmware package. The instructions in this set are:

LOAD	-	Load Hard Register Instruction
READ	-	Read Soft Register Instruction
LAM	-	Load Auxiliary Memory Instruction
RAM	-	Read Auxiliary Memory Instruction
RSET	-	Reset Instruction
INIT	-	Initialize Instruction

These instructions are standard in all RM-9000 systems.

3-40 Load Hard Register Instruction (LOAD)



The LOAD instruction is a special format instruction which is part of the RM-9000 standard firmware instruction set. The function of the LOAD instruction is to move the sixteen (16) bits of data, stored in the second word of the instruction, directly into the RM-9000 internal display register whose address is specified in the low-order byte of the opcode word. This instruction allows the user to bypass the overhead in execution time necessary to perform various display operations using the normal format instructions. The information which is written into the specified RM-9000 display register is not retrievable by the user. In addition, the direct storage of data into a display register using LOAD may cause the normal format instructions to malfunction. Since the normal format instruction software assumes that the state of all internal display registers can be modified only by a normal format instruction, the functioning of any normal format instruction after the execution of a LOAD can only be guaranteed if all normal format instruction parameters are reset within a normal format instruction. The use of the LOAD instruction requires extensive knowledge of the operation of the internal display registers, and as such its use is not encouraged. The function of the various RM-9000 display registers is defined in Volume I of the RAMTEK 9000 SERIES THEORY OF OPERATION.

PERTINENT OPERANDS

REGISTER ADDRESS — Defines the low-order 8 bits of the internal address of the desired display register. Table 3- defines the register address assignments.

REGISTER DATA — Defines the 16-bit value to be loaded into the display register defined by REGISTER ADDRESS. The format of the data word is specific and different for each register and may be found in VOLUME I of the RAMTEK 9000 SERIES THEORY OF OPERATION.

POSSIBLE ERRORS

If REGISTER ADDRESS is set to an undefined value (i.e., an address for which no display register exists), the display produced will be indeterminate. No register address error checking is performed.

Table 3-11 I/O Device Register Address Map (RM-9000 Series)

HEX ADDRESS	MNEMONIC	DESCRIPTION
00	SCMSKR	Memory Plane Select Register
02	ROPTRG	Interrupt Register
04	READR	Readback Register
06	XORGR	X Origin Register
08	YORGR	Y Origin Register
0A	NPRSRC	DMA Source Register
0C	NPRDST	DMA Destination Register
0F	UPDTR	COP/Write Control Register
10	BGR	Background Register
12	FGR	Foreground Register
14	XCOPR	Element COP Register
16	YCOPR	Line COP Register
18	RASREG	Raster Font Register
1A	ID	Interface Data Register
1C	SYSS	System/Interface Status Register
1E	WRDCNT	DMA Word Count Register
20	DMAADR	DMA Address Register
23	MOVWTR	Move and Write Register
25	MOVR	Move (W. No Write) Register

Table 3-11 (Continued)

HEX ADDRESS	MNEMONIC	DESCRIPTION
26	DPSW	Dip Switch Register
28		
2A		
2C		
2E		
30		
32		
34		
36		
38		
3A		
3C	SPS1	* Serial Peripherals Status Register (Board 1)
3E	SPS2	* Serial Peripherals Status Register (Board 2)
40	SPK11	* Serial Peripherals Keyboard #1 I/O Register (Board 1)
42	SPK12	* Serial Peripherals Keyboard #2 I/O Register (Board 1)
44	SPK13	* Serial Peripherals Keyboard #3 I/O Register (Board 1)
46	SPK14	* Serial Peripherals Keyboard #4 I/O Register (Board 1)

Table 3-11 (Continued)

HEX ADDRESS	MNEMONIC	DESCRIPTION
48	SPCX11	* Serial Peripherals Cursor #1 X-Position Register (Board 1)
4A	SPCY11	* Serial Peripherals Cursor #1 Y-Position Register (Board 1)
4C	SPCX2	* Serial Peripherals Cursor #2 X-Position Register (Board 1)
4E	SPCY2	* Serial Peripherals Cursor #2 Y-Position Register (Board 1)
50	SPK21	* Serial Peripherals Keyboard #1 I/O Register (Board 2)
52	SPK22	* Serial Peripherals Keyboard #2 I/O Register (Board 2)
54	SPK23	* Serial Peripherals Keyboard #3 I/O Register (Board 2)
56	SPK24	* Serial Peripherals Keyboard #4 I/O Register (Board 2)
58	SPCX21	* Serial Peripherals Cursor #1 X-Position Register (Board 2)
5A	SPCY21	* Serial Peripherals Cursor #1 Y-Position Register (Board 2)
5C	SPCX22	* Serial Peripherals Cursor #2 X-Position Register (Board 2)
5E	SPCY22	* Serial Peripherals Cursor #2 Y-Position Register (Board 2)

Table 3-11 (Continued)

HEX ADDRESS	MNEMONIC	DESCRIPTION
60	VLTAD0	* Lookup Table #0 Address Register
62	VLTD0	* Lookup Table #0 Programmed Data Register
64	VLTAD1	* Lookup Table #1 Address Register
66	VLTD1	* Lookup Table #1 Programmed Data Register
68	VLTAD2	* Lookup Table #2 Address Register
6A	VLTD2	* Lookup Table #2 Programmed Data Register
6C	VLTAD3	* Lookup Table #3 Address Register
6E	VLTD3	* Lookup Table #3 Programmed Data Register
		* May not exist, depending on system configuration.

3-41 Read Soft Register Instruction (READ)

OPCODE (02)	REGISTER ADDRESS

The READ instruction is a special-format instruction which is contained in the standard RM-9000 firmware package. The function of READ is to return the current value of one of seven internal "soft" registers used by the normal-format instruction firmware package. These soft registers are intended to reflect the state of the actual hardware registers as loaded and updated by the firmware. These soft registers are not affected when the associated hardware display register is loaded via the LOAD instruction.

When contents of the soft and hard registers aren't identical, unexpected results will be obtained when executing normal-format instructions. The data format of the soft register is identical to that of the associated hardware register which may be found in VOLUME 1 of the RAMTEK 9000 SERIES THEORY OF OPERATION Manual.

The READ instruction is, in fact, a read-request instruction requiring 2 steps. Following the output of the READ instruction, the user (i.e., host computer) must read back exactly one (1) 16-bit word from the computer interface; output of succeeding instructions may then proceed. If no read back is done and an attempt is made to output instructions, the RM-9000 interface and display system will hang and can only be restored with a hard system reset. Therefore, it is necessary to maintain synchrony with the RM-9000 in terms of the direction of data transfers.

PERTINENT OPERANDS

REGISTER ADDRESS — Defines the low-order byte of the internal display register address to be read back. Table 3-12 defines the valid register addresses which may be accessed.

POSSIBLE ERRORS

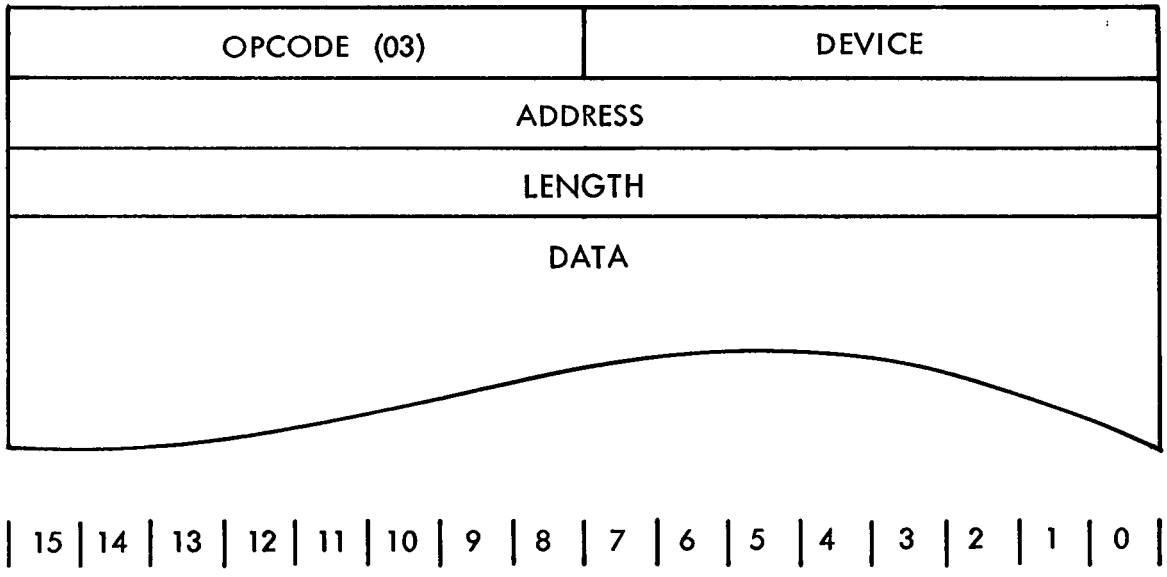
If the REGISTER ADDRESS specified in the low-order byte of the opcode word is a value not defined in Table 3-12, the read-back value will be meaningless.

If the host processor does not attempt to read back a word from the RM-9000, the system will hang and can only be restored by a hard system reset.

Table 3-12 Read Soft Register Addresses

REGISTER ADDRESS	REGISTER
00 (HEX)	SUBCHANNEL MASK
02	X (ELEMENT) COP
04	Y (LINE) COP
06	X (ELEMENT) ORIGIN
08	Y (LINE) ORIGIN
12	FOREGROUND
14	BACKGROUND

3-42 Load Auxiliary Memory Instruction (LAM)



The LAM instruction is a special-format instruction which is included in the RM-9000 standard firmware package. The function of the LAM instruction is to transfer data to auxiliary memory devices within the RM-9000 system other than refresh memory or internal 8080 memory from the host processor. In the current RM-9000 series implementation, the LAM instruction is used to write data to the video lookup tables provided on some of the video boards (e.g., V2, V8, V12). A description of the data format for the Type II video board is found in Appendix A. The value of DEVICE may range from 0 to 3.

PERTINENT OPERANDS

DEVICE — Defines the auxiliary memory device to which the data will be written.

ADDRESS — Specifies a 16-bit address within the auxiliary memory device specified by DEVICE. This address is device-dependent, and in the case of the Type II video cards, it represents the starting video look-up table address at which data will be written.

Table 3-13 Video Look-Up Table Addressing

DEVICE	LOOK-UP TABLE DEFINITION
0	Type II Video Look-up Table 0 (with internal addresses 60_{16} and 62_{16})
1	Type II Video Look-up Table 1 (with internal addresses 64_{16} and 66_{16})
2	Type II Video Look-up Table 2 (with internal addresses 68_{16} and $6A_{16}$)
3	Type II Video Look-up Table 2 (with internal addresses $6C_{16}$ and $6E_{16}$)

LENGTH Defines the number of bytes of data to be written to the auxiliary memory device.

DATA FORMAT

The format of the data associated with a LAM instruction is device dependent. Appendix A defines the format for the Type II video board.

3-43 Read Auxiliary Memory Instruction (RAM)

OPCODE (04)	DEVICE
ADDRESS	
LENGTH	

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

The RAM instruction is a special-format instruction which is included in the RM-9000 standard firmware package. The function of the RAM instruction is to transfer data from an auxiliary memory device within the RM-9000 system to the host processor. In the current RM-9000 series implementation, the RAM instruction is used to read the contents of the video lookup tables found on some video boards (e.g., V2, V8, V12) if one of these devices exists within a given system. A description of the data format to be read back by the host processor is found in Appendix A. The value of DEVICE may range from 0 through 3.

The RAM instruction is a read-request instruction requiring two steps. The first step is to output the three words of the RAM instruction. Following the output of the LENGTH word, it is necessary to change the sense of the computer interface board from a write mode to read mode. The number of bytes as specified by LENGTH must then be read back into the host computer. If this is not done, the handshaking sequence between the RM-9000 and the host processor will lose synchronization and it will be necessary to perform a hard system reset to restore communications.

PERTINENT OPERANDS

DEVICE	Defines the auxiliary memory device from which the data will be read back. (See Table 3-13)
ADDRESS	Specifies a 16-bit address within the auxiliary memory device specified by DEVICE. This address is device-dependent, and in the case of the Type II video cards, it represents the starting video look-up table address from which data will be read back.
LENGTH	Defines the number of bytes of data to be read back from the auxiliary memory device. This value must be an even number since data is transferred across the interface on a 16-bit basis. If LENGTH is an odd number, the value used will be LENGTH-1.

DATA FORMAT

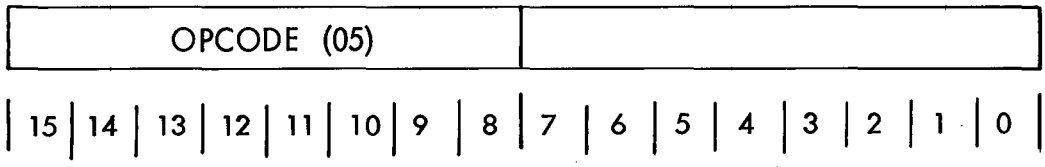
The format of the data associated with a RAM instruction is device dependent. Appendix A defines the format for the Type II video board.

POSSIBLE ERRORS

If the host processor does not read back the number of bytes defined in LENGTH, the system will hang and can only be restored by a hard system reset.

If the value in LENGTH is odd, the value used as a count will be LENGTH minus one (1) and an I/O direction conflict may arise (See above).

3-44 Reset Instruction (RSET)



The RSET instruction is a special-format instruction which is included in the RM-9000 standard firmware package. This instruction checksums all PROM-resident memory, clears all interrupt lines from the processor to the interface board, clears the RM-STA status management option stack, clears the RM-PER interactive peripherals option keyboard input buffers and resets the UART devices on the RM-9000 serial link board, sets all cursors to invisible at screen location (0,0), copies the standard ASCII character font from PROM to RAM if RM-FNT option is present (hex character Codes 20-5F), and sets all normal-format instruction parameter operands to their default values (See Table 3-14 for these values). If DIP SWITCH 3 is set to Ø (down on the '+' side of the switch), the entire refresh memory will be erased -- i.e. set to the BACKGROUND parameter operand default value, i.e., a value of 0; otherwise, no refresh memory erasing is performed. The RM-9000 system is then ready to accept the next instruction. If the toggle switch mounted on the side of the control board is in the DIAG position and a RESET instruction is received, the RM-DOC diagnostic tester firmware takes control (See Figure 3-7).

PERTINENT OPERANDS None

POSSIBLE ERROR CONDITIONS None

Table 3-14 Operand Parameter Default Values

OPERAND PARAMETER	DEFAULT VALUE	DESCRIPTION
SUBCHANNEL	$\emptyset FFF_{16}$	Write-Enable all subchannels
FOREGROUND	$\emptyset FFF_{16}$	1's Data to all subchannels
BACKGROUND	ØØØØ	Ø 's Data to all subchannels
INDEX 1	Ø, Ø	INDEX 1 X = 0, Y = 0
INDEX 2	Ø, Ø	INDEX 2 X = 0, Y = 0
ORIGIN	X_{OR}, Y_{OR}	ORIGIN X and Y values defined on a system-resolution basis
WINDOW	Ø, Ø, X_{RES}, Y_{RES}	Set WINDOW boundaries to full-screen size based on the system resolution
SCAN	Ø	Left-to-right, top-to-bottom
DIMENSION	7, 9	7 Elements wide, 9 lines high character font
SPACING	7, 9	7 Elements wide, 9 lines high spacing between characters or plot entities
SCALE	Ø	No scaling
FUNCTION	Ø	No logical/arithmetic processing
CONIC-EQUATION	6 Ø, Ø	Conic parameters A, B, C, D, E, K = Ø
BASELINE	Ø	Line plotting
SCROLLCOUNT	Ø	No Scroll
START POINT	Ø, Ø	XCOP = Ø, YCOP = Ø

Table 3-15 (Continued)

SYSTEM	X_{OR}	Y_{OR}	X_{RES}	Y_{RES}
9100	136_{16}	$ØFF_{16}$	$13F_{16}$	$ØFF_{16}$
9200	$26C_{16}$	$ØFF_{16}$	$27F_{16}$	$ØFF_{16}$
9300	$26C_{16}$	$1FE_{16}$	$27F_{16}$	$1FF_{16}$

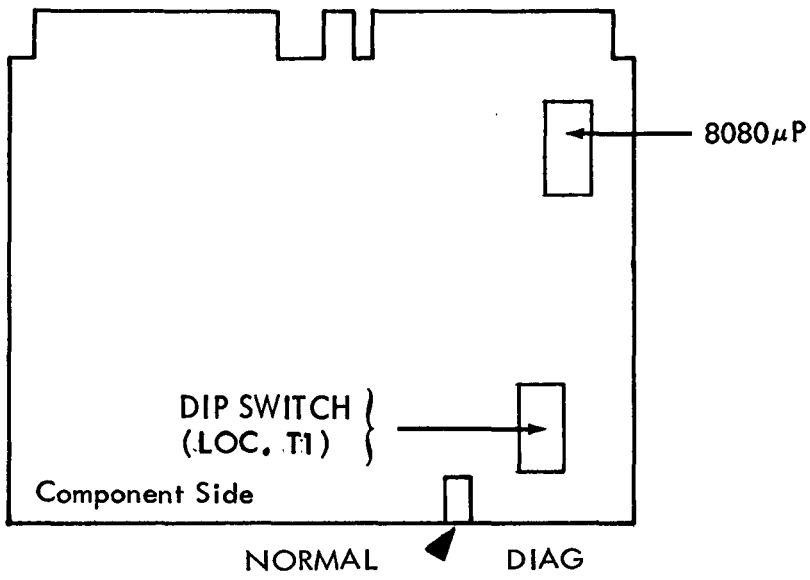
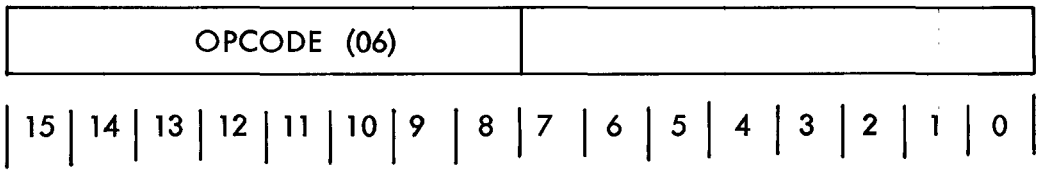


Figure 3-7 RM-9000 Control Board

3-45 Initialize Instruction (INIT)



The INIT instruction is a special-format instruction which is included in the RM-9000 standard firmware. The INIT instruction resets the normal-format instruction operand parameters to their default values (See Table 3-14). It is equivalent to issuing a SET instruction (See 3-42) with all parameter operand default values present in the instruction itself. The status management stack is not affected, thus, any previous PUSHE operations will still be available using the POPE instruction (See Sections 3-57 and 3-58).

PERTINENT OPERANDS

None

POSSIBLE ERROR CONDITIONS

Using the INIT instruction within an XIM instruction causes the XIM subroutine pointer to be reset. This effectively destroys that XIM instruction. Thus, issue the INIT instruction before the XIM instruction.

3-46 OPTIONS FIRMWARE INSTRUCTION

The options firmware instruction set provides the user with additional display and display support functions. Many of the functions, such as the drawing of vectors or conics, could be performed by the host processor; however, these operations would be prohibitive in terms of host processor software overhead. By using the firmware options instructions, it is possible to distribute the display generation process more evenly between the host processor and the display system.

All of the firmware supporting the firmware options instruction set is resident on the RM-MOC memory expansion card in the form of PROM. In order to enable any of the firmware options, it is necessary that Switch 2 of the DIP SWITCH REGISTER on the control board is set to the 'OFF' position. It is possible to disable all options' instruction processing by setting Switch 2 of the DIP SWITCH REGISTER to the 'ON' position, i.e., up on the '+' side of the switch. (See figure 3-7 and table 3-17, page 3-128A.) No removal of the RM-MOC card is necessary. If, however, Switch 2 is set to the 'OFF' position, it is necessary that the RM-MOC card as well as the RM-STD standard option support PROM be present in the system. If the RM-STD firmware is not present on the RM-MOC in the proper slot, the system will never leave self-test and will be hung until this condition is corrected and a hard system is issued.

The purpose of the RM-STD firmware is to provide a common interface between the RM-9000 standard and optional instruction set. The RM-STD firmware determines by inspection which options package(s) are actually installed in any given system. A section of this firmware is called during the firmware reset sequence to call any existing option power-on/reset routine to initialize all internal parameters. The RM-STD firmware allows the modularity of the various options' firmware packages. This package resides within the internal 8080 processor's address space at locations 0800_{16} through $OBFF_{16}$.

framtek

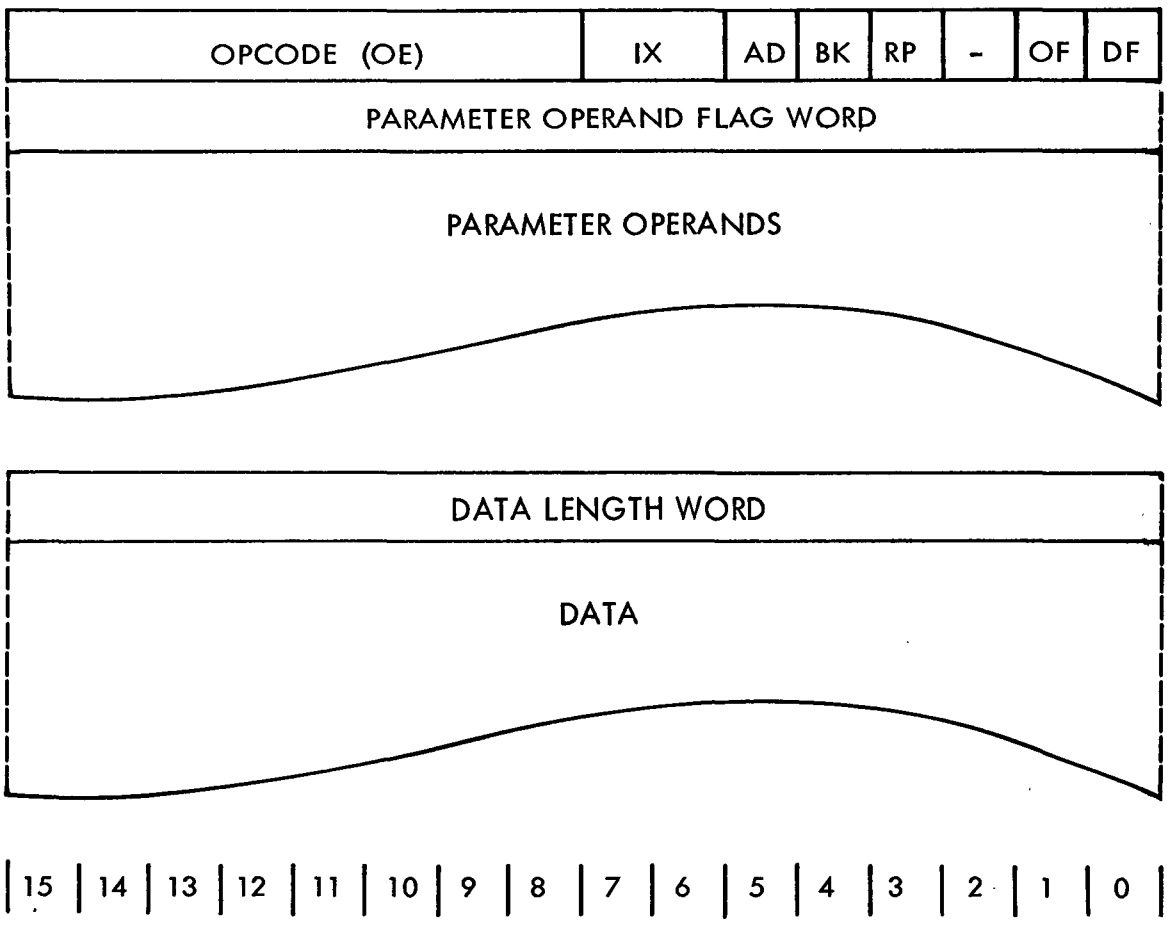
3-47 RM-GRA Graphics Option Instructions

The RM-GRA graphics option firmware implements the following instructions:

- WRITE VECTOR generating linked end-point vectors
- WRITE PLOT generating linked plots and filled bar plots with a fixed plot axis
- WRITE RASTER generating display data directly to refresh memory on a bit-per-pixel basis.

The RM-GRA option resides within the internal 8080 processor's address space at locations 2800_{16} through $2BFF_{16}$. The RM-GRA graphics option is a prerequisite for the RM-CON option (See Section 3-51).

3-48 Write Vector Instruction (WV)



The WV instruction is a normal format instruction which is part of the RM-GRA graphics firmware option. The WV (write vector) instruction draws a continuous, straight line in the color or intensity specified by Foreground between the point (X_S , Y_S) and the (X_1 , Y_1), where (X_S , Y_S) is either the START-POINT argument if actually specified in the instruction or the COP (Current Operating Point) resulting from the value resultant from the previous instruction, and (X_1 , Y_1) is the first end-point specified in the WV instruction. If more than one (1) vector end-point is specified, then continuous, straight lines will be drawn between point (X_{i-1} , Y_{i-1}) and point (X_i, Y_i) where i = 2 through N (N being equal to the number of endpoints specified in the WV instruction). Thus, it is not possible to draw more than one (1) fixed-point vector (i.e., vectors with the same starting point) in the same WV instruction; multiple WV instructions, specifying a common START-POINT and one (1) vector per WV instruction, must be constructed. The WV instruction is <u>not</u> affected by the window

parameters. The length value defines the positive number of bytes of vector endpoint information to be used by this instruction; therefore, length must be four (4) times the number of vectors to be drawn.

PERTINENT CONTROL BITS

IX	Defines the address mode in which the INDEX 1, INDEX 2, and START-POINT parameter operands (i.e., those operands directly affecting the absolute position of any vectors drawn) are evaluated (See Section 3-11).
BK	Causes the vector(s) to be drawn in the BACKGROUND color or intensity, rather than the FOREGROUND color or intensity (when BK = Ø).

PERTINENT ARGUMENTS

SUBCHANNELS	Defines which combination of memory planes will actually receive vector color or intensity data. The value actually written to refresh memory is the logical 'and' of SUBCHANNELS with foreground or background dependent on the state of BK.
FOREGROUND	Defines the vector color or intensity in conjunction with SUBCHANNELS when BK = Ø .
BACKGROUND	Defines the vector color or intensity in conjunction with SUBCHANNELS with BK = 1.
	INDEX 1 displaces all WV coordinate pairs as well as the INDEX 2 and START-POINT operands (when set in the same WV instruction) when IX = 01.
	INDEX 2 displaces all WV coordinate pairs as well as the START-POINT operand (when set in the same WV instruction) when IX = 10.
START-POINT	Defines the starting coordinate for the first vector to be drawn. If not specified, the starting coordinate used is the COP resulting from the previous display instruction.

DATA FORMAT

The format of the vector end-point value is as follows: Each successive pair of 16-bit words defines the X and Y coordinate for the next vector endpoint. The first 16-bit word contains the X-value, and the second 16-bit word contains the Y-value. The interpretation of this coordinate is defined by the value of IX in the opcode word (See Section 3-11 for the possible addressing modes).

POSSIBLE ERROR CONDITIONS

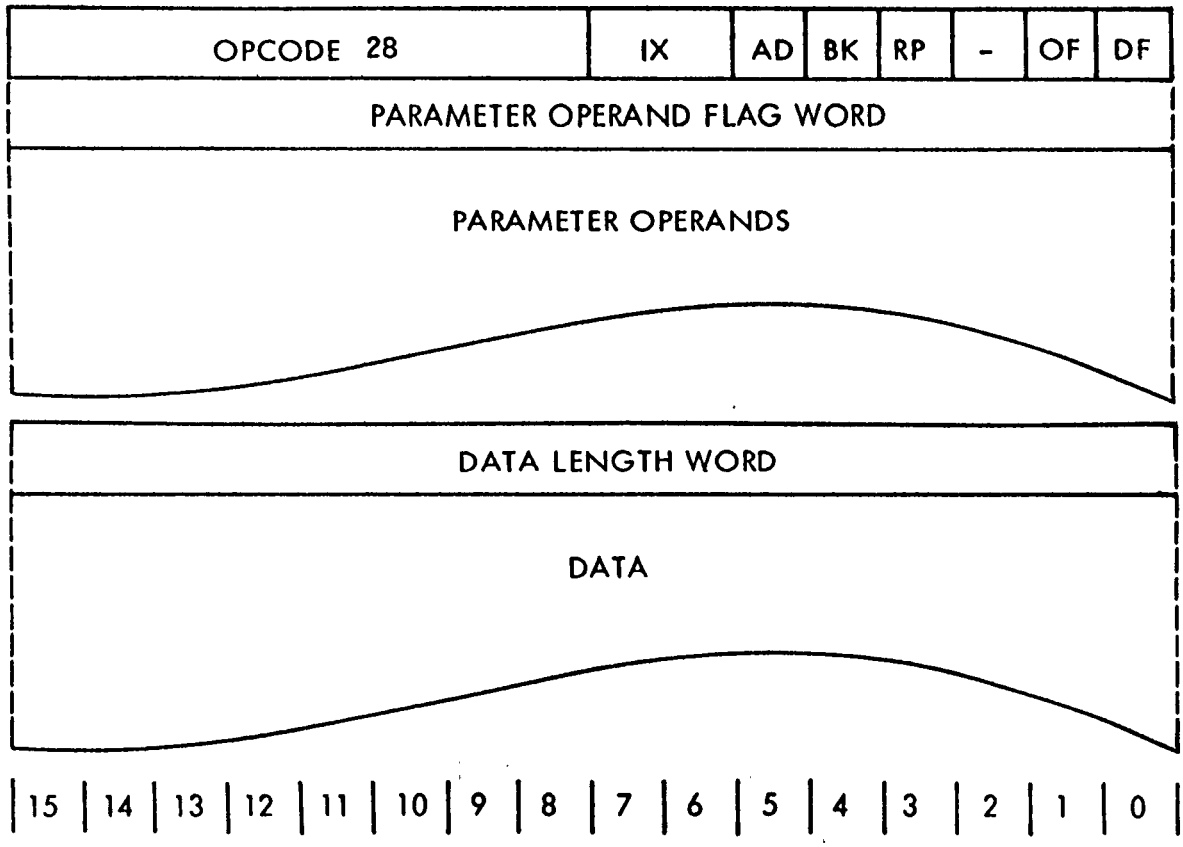
The following conditions are invalid and undefined, and will result in an unpredictable display if attempted for the WV instruction:

- If any of the specified endpoints exceeds the screen resolution, the generated vector(s) will be indeterminate.

COP MOVEMENT

After the successful execution of this instruction, the COP will be at point (X_n , Y_n) where N is the number of endpoints specified by the WV instruction.

3-48.5 Write Dashed Vector Instruction (WDV)



The WDV instruction is a normal format instruction which is part of the RM-GRA graphics firmware option. The WDV (Write Dashed Vector) instruction draws a dashed line in the color or intensity specified by foreground for dashes and background for spaces between the point (X_s, Y_s) and the (X_1, Y_1), where (X_s, Y_s) is either the start-point argument if actually specified in the instruction or the COP (current operating point) resulting from the value resultant from the previous instruction and (X_1, Y_1) is the first endpoint specified in the WDV instruction. If more than one (1) vector endpoint is specified, then linked, dashed lines will be drawn between point (X_{i-1}, Y_{i-1}) and point (X_i, Y_i) where i=2 through N (N being equal to the number of endpoints specified in the WDV instruction). Thus, it is not possible to draw more than one fixed-point vector (i.e., vectors with the same starting point) in the same WDV instruction; multiple WDV instructions, specifying a START-POINT and one (1) vector end point per WDV instruction, must be constructed. The WDV instruction is NOT affected by the window parameters. The data length value defines the positive number of bytes of vector endpoint information to be used by this instruction; therefore, data length must be four times the number of vectors to be drawn.

The dashed line values are provided via the SPACING parameter. The X-SPACING represents the on-count value which is the number of pixels in a dash while the Y-SPACING parameter represents the off-count value which is the number of pixels between dashes. The order of action is to start with on-count and toggle between it and off-count until the list of endpoints is exhausted. The off-count value may be zero (0) in which case this is effectively a WV instruction.

PERTINENT CONTROL BITS

IX	Defines the address mode in which the INDEX 1, INDEX 2, and START-POINT parameter operands (i.e., those operands directly affecting the absolute position of any vectors drawn) are evaluated (See Section 3-11).
BK	Causes the vector(s) to be drawn in the BACKGROUND color or intensity, rather than the FOREGROUND color or intensity (when BK = Ø).

PERTINENT ARGUMENTS

SUBCHANNELS	Defines which combination of memory planes will actually receive vector color or intensity data. The value actually written to refresh memory is the logical 'and' of SUBCHANNELS with foreground or background dependent on the state of BK.
FOREGROUND	Defines the vector color or intensity in conjunction with SUBCHANNELS when BK = Ø.
BACKGROUND	Defines the vector color or intensity in conjunction with SUBCHANNELS when BK = 1.
INDEX 1	INDEX 1 displaces all WV coordinate pairs as well as the INDEX 2 and START-POINT operands (when set in the same WV instruction) when IX = 01.
INDEX 2	INDEX 2 displaces all WV coordinate pairs as well as the START-POINT operand (when set in the same WV instruction) when IX = 10.

SPACING	X-SPACING defines the on-count value. Y-SPACING defines the off-count value.
START-POINT	Defines the starting coordinate for the first vector to be drawn. If not specified, the starting coordinate used is the COP resulting from the previous display instruction.

DATA FORMAT

The format of the vector end-point value is as follows: Each successive pair of 16-bit words defines the X and Y coordinate for the next vector endpoint. The first 16-bit word contains the X-value, and the second 16-bit word contains the Y-value. The interpretation of this coordinate is defined by the value of IX in the opcode word (See Section 3-11 for the possible addressing modes).

POSSIBLE ERROR CONDITIONS

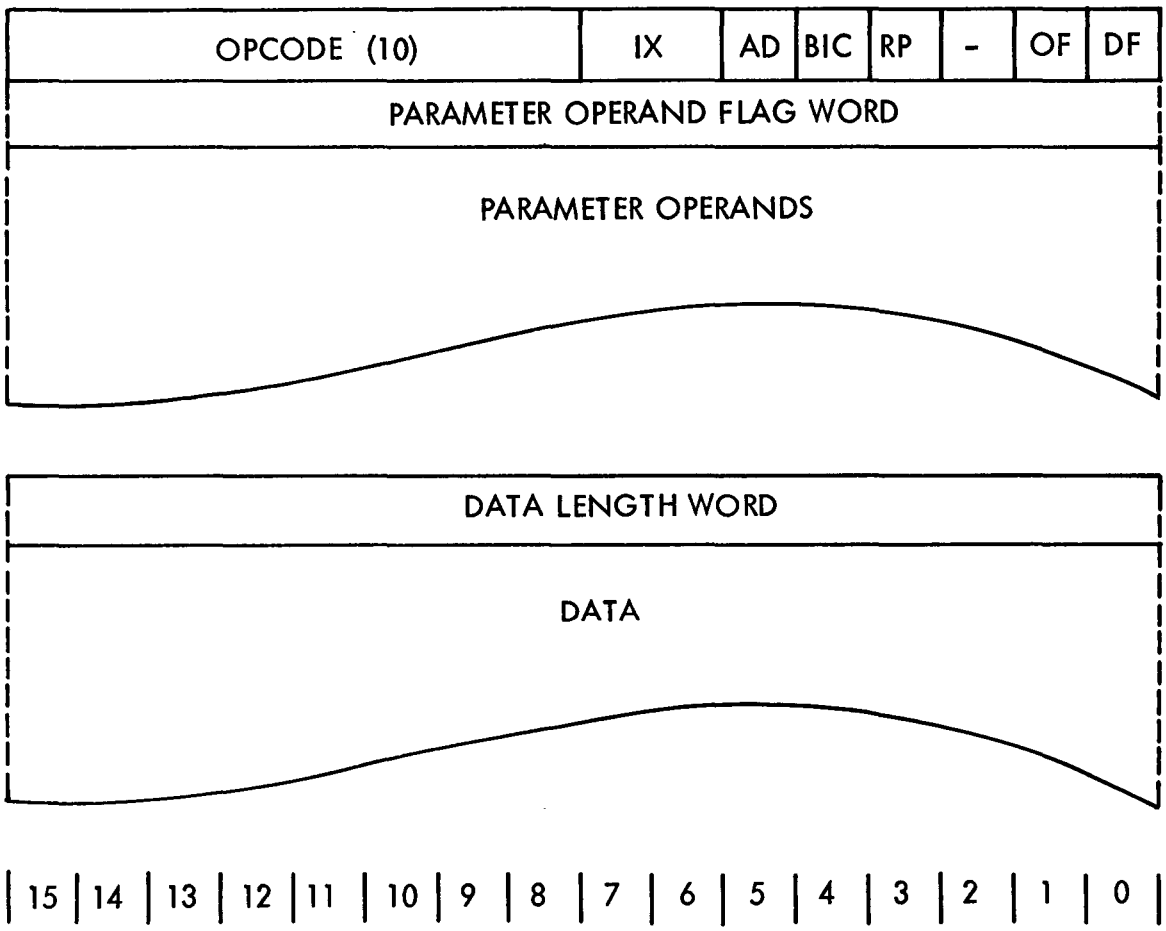
The following conditions are invalid and undefined, and will result in an unpredictable display if attempted for the WV instructions:

- If any of the specified endpoints exceeds the screen resolution, the generated vector(s) will be indeterminate.

COP MOVEMENT

After the successful execution of this instruction, the COP will be at point (X_n, Y_n) where N is the number of endpoints specified by the WV instruction.

3-49 Write Plot Instruction (WP)



The purpose of the Write Plot instruction is to plot curves (either filled or unfilled) and bar graphs. The WP instruction is a normal format instruction which is part of the RM-GRA graphics firmware option. The WP instruction will automatically update one (1) COP while receiving the other. This can reduce data transfers significantly. The WP instruction generates one (1) plot segment for each 16-bit word of data present in the WP instruction using a raster-data technique. The orientation of the plot segment (i.e., differentiating between X-axis plots and Y-axis plots) is defined by SCAN. A plot segment's height (i.e., dimension perpendicular to the plot axis as defined by SCAN) is determined by the current-operating point and a 16-bit data word from the instruction. If BASELINE is zero (i.e., line plot mode), then initially the COP is either set to the START-POINT value if specified in the WP instruction or retains the COP value from the previous instruction. If BASELINE is non-zero, then initially the plot-axis COP coordinate is set to the BASELINE value. For example, if SCAN is zero (i.e., Y-axis plot), then the line COP will be set to the BASELINE value. The width of a

plot segment (i.e., dimension parallel to the plot axis as defined by SCAN) is defined by the DIMENSION parameter operand parallel to the plot axis. After each plot segment is drawn, the COP coordinate parallel to the plot axis is updated by the SPACING parameter operand parallel to the plot axis. If BASELINE is zero, the COP coordinate perpendicular to the plot axis is unchanged (i.e., plot segments in linked mode will overlap by one (1) pixel in this axis). If BASELINE is non-zero, the COP coordinate perpendicular to the plot axis is set to BASELINE after each plot segment is drawn.

The color or intensity of the plot segments is defined by either FOREGROUND (when BK = 0) or by BACKGROUND (when BK = 1) and is masked by the write-enable parameter operand SUBCHANNEL. The WRITE PLOT instruction is not affected by the WINDOW parameter operands.

PERTINENT CONTROL BITS

IX	Defines the address mode in which the INDEX 1, INDEX 2, BASELINE and START-POINT parameter operands (i.e., those operands directly affecting the absolute position of the plot axis or the plot data) are evaluated (See Section 3-11).
BK	Defines the color or intensity used for the plot segments drawn. If BK = 0, the FOREGROUND value will be used; if BK = 1, the BACKGROUND value will be used.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL	Specifies the subchannels (i.e., refresh memory bit planes) which are write-enabled and which will receive plot data.
FOREGROUND	Defines the color or intensity value for all plot segments when BK = 0.
BACKGROUND	Defines the color or intensity value for all plot segments when BK = 1.
INDEX 1	Displaces all WP data values as well as INDEX 2, BASELINE and START-POINT operands (when set in the same WP instruction) when IX = 01.

INDEX 2	Displaces all WP data values as well as BASELINE and START-POINT operands (when set in the same WP instruction) when IX = 10.
SCAN	Defines the plot orientation. If SCAN = 0, the plot axis will be the Y-axis (i.e., horizontal plot); if SCAN = 4, the plot axis will be the X-axis (i.e., vertical plot).
DIMENSION	Defines the size of a plot segment parallel to the plot axis. If SCAN = 0, DIMENSION width is used; if SCAN = 4, DIMENSION height is used.
SPACING	Defines the distance between starting COP for successive plot segments along the plot axis. If SCAN = 0, X-spacing is used; if SCAN = 4, Y-spacing is used.
BASELINE	Defines the fixed-line plot axis if BASELINE is non-zero. If SCAN = 0, BASELINE represents the Y-coordinate of the plot axis; if SCAN = 4, BASELINE represents the X-coordinate of the plot axis.
START-POINT	Defines the initial COP for the first plot segment if BASELINE = 0; if BASELINE is non-zero, START-POINT defines the starting coordinate of the first plot segment along the plot axis.

DATA LENGTH WORD

The DATA LENGTH WORD represents the number of bytes of plot data to be transmitted from the host processor to the RM-9000. Since each plot segment is represented by a single 16-bit word, the DATA LENGTH WORD should always reflect an even number of bytes. If the DATA LENGTH WORD is odd, the byte count used will be one less than the actual DATA LENGTH WORD received.

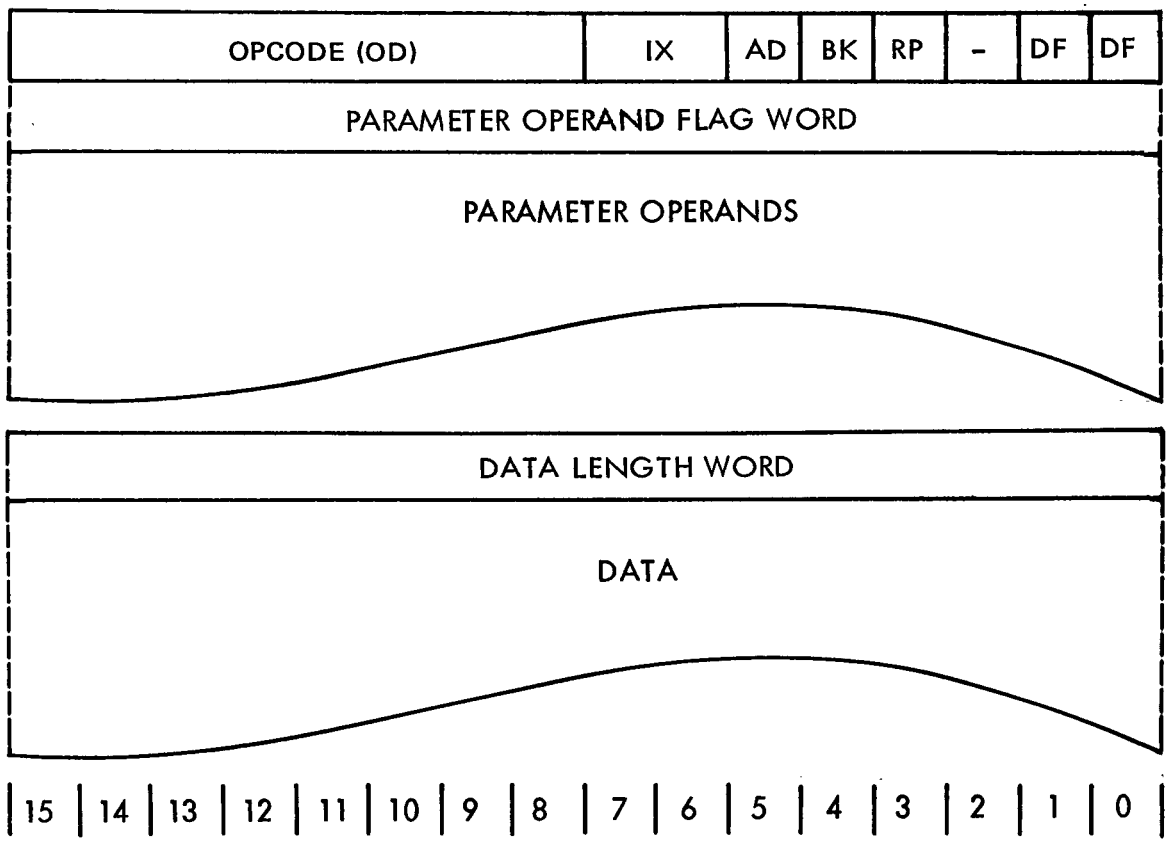
DATA FORMAT

Data in plot mode is interpreted on a word-per-plot segment basis. Each word represents the coordinate of a plot segment (as evaluated in light of the IX value) perpendicular to the plot axis.

COP MOVEMENT

After execution of the WP instruction the new COP reflects the starting coordinate of the next plot segment if one (1) additional data word has been specified in the WP instruction.

3-50 Write Raster Instruction (WR)



The purpose of the Write Raster instruction is to provide a method for writing into refresh memory data which is X-Y plane oriented as opposed to the WI instruction which is Z-axis oriented. A typical application of WR is in the generation of software-defined symbols or characters which the 8 x 12 dot-matrix used in the programmable font option.

The WR instruction is a normal format instruction included in the RM-GRA graphics option firmware. The WR instruction interprets data from the host processor and writes either the FOREGROUND or BACKGROUND value into refresh memory as a function of the incoming data bytes (on a bit-per-pixel basis) and the AD and BK flags. The WR instruction operates in a manner analogous to the WT instruction except that the data for the character generation is obtained from RM-9000 internal memory, whereas, the WR data is obtained directly from the host processor.

When a byte of data is received from the host processor by the RM-9000, each of the eight (8) bits represents a pixel in refresh memory. The value which is actually written to refresh memory is defined by Table 3-15. Within each byte of data, the bits are evaluated from high to low order (left-to-right).

Table 3-15 Additive (AD), Background (BK) & Data Bit Relation

BK \ AD	0	0	1	1
0	B	F	X	F
1	F	B	X	B
	0	1	0	1

F - Write Foreground Value
B - Write Background Value
X - No Write

The WR instruction operates within the rectangular region defined by the WINDOW parameter operands. The SCAN parameter defines the primary (pixel-to-pixel) update direction and the secondary (window boundary intersection) update direction in a manner identical to that of image mode (See Table 3-6). In the case of a zero data bit combined with AD = 1 (i.e., the 'no write' condition), pixel-to-pixel autoincrement by 1 is still performed even though no writing into refresh memory is performed. When the window boundary is intersected, any bits remaining in the data byte current at the intersection are discarded; therefore, after any secondary update, the first pixel value written is defined by the high-order bit of the next data byte.

When the RM-SCA scaling option firmware is installed and a scaling-down operation is specified by the SCALE parameter operand in either dimension, Bits 7, 5, 3 and 1 are used for a ratio of 2:1 and Bits 7 and 3 are used for a ratio of 4:1.

PERTINENT CONTROL BITS

IX — Defines the address mode in which the INDEX 1, INDEX 2, WINDOW and START-POINT parameter operands set in the WR instruction will be evaluated.

AD — Affects the generation of raster data in conjunction with the BK flag. Input data is inhibited in the manner specified by Table 3-15.

BK — Defines the selection of FOREGROUND and BACKGROUND colors based on the one's/zeros data bits within the raster byte and on the value of the AD flag (See Table 3-15). If BK = 0, the FOREGROUND is selected for 'ones' raster

data and the BACKGROUND value is selected for 'zeros' raster data (assuming AD = 0). If BK = 1, the FOREGROUND and BACKGROUND selection is reversed.

RP — Defines the byte-accessing order for each 16-bit word of raster data from the host processor. If RP = 0, the high-byte of the data word is processed before the low-byte; if RP = 1, the low-byte is processed first.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL — Specifies the subchannels (i.e., refresh memory bit planes) which are write-enabled and which will receive raster data.

FOREGROUND — Defines the foreground color or intensity value to be written to refresh memory when selected by the raster data bit value and the AD and BK flags (See Table 3-15).

BACKGROUND — Defines the background color or intensity value to be written to refresh memory when selected by the raster data bit value and the AD and BK flags (See Table 3-15).

INDEX 1 — Displaces the values to be used for INDEX 2, WINDOW and START-POINT parameter operands set in the WR instruction when IX = 01.

INDEX 2 — Displaces the values to be used for WINDOW and START-POINT parameter operands set in the WR instruction when IX = 10.

WINDOW — Defines the rectangular region into which the raster data will be written. No raster data will be written outside of the boundaries defined by this parameter operand.

SCAN — Defines the primary and secondary raster update directions as well as START-POINT when START-POINT is not specified in the WR instruction (See Table 3-6).

SCALE — Defines the received-bit-to-displayed-pixel ratio for raster data when the RM-SCA option is installed (See Section 3-24, Table 3-9).

START-POINT — Specifies the coordinates of the first pixel to be written with raster data START-POINT must be contained within the current WINDOW region.

DATA LENGTH WORD

The DATA LENGTH WORD represents the number of bytes of raster data to be transmitted from the host processor to the RM-9000 display system. Since raster data is byte oriented, the DATA LENGTH WORD may take on any value from 0 through 65535_{10}.

DATA FORMAT

Data in raster mode is interpreted on a byte basis, (i.e., two (2) bytes per data word transferred across the RM-9000 interface). The order that the bytes are referenced is determined by the RP flag. When RP = 0, the high byte will be used first; when RP = 1, the low byte will be used first. Within a data byte, the high order Bit 7 is used first, while Bit 0 will be referenced last. If a WINDOW boundary is encountered before all bits of a raster data byte are used, the remaining bits of that byte will be discarded.

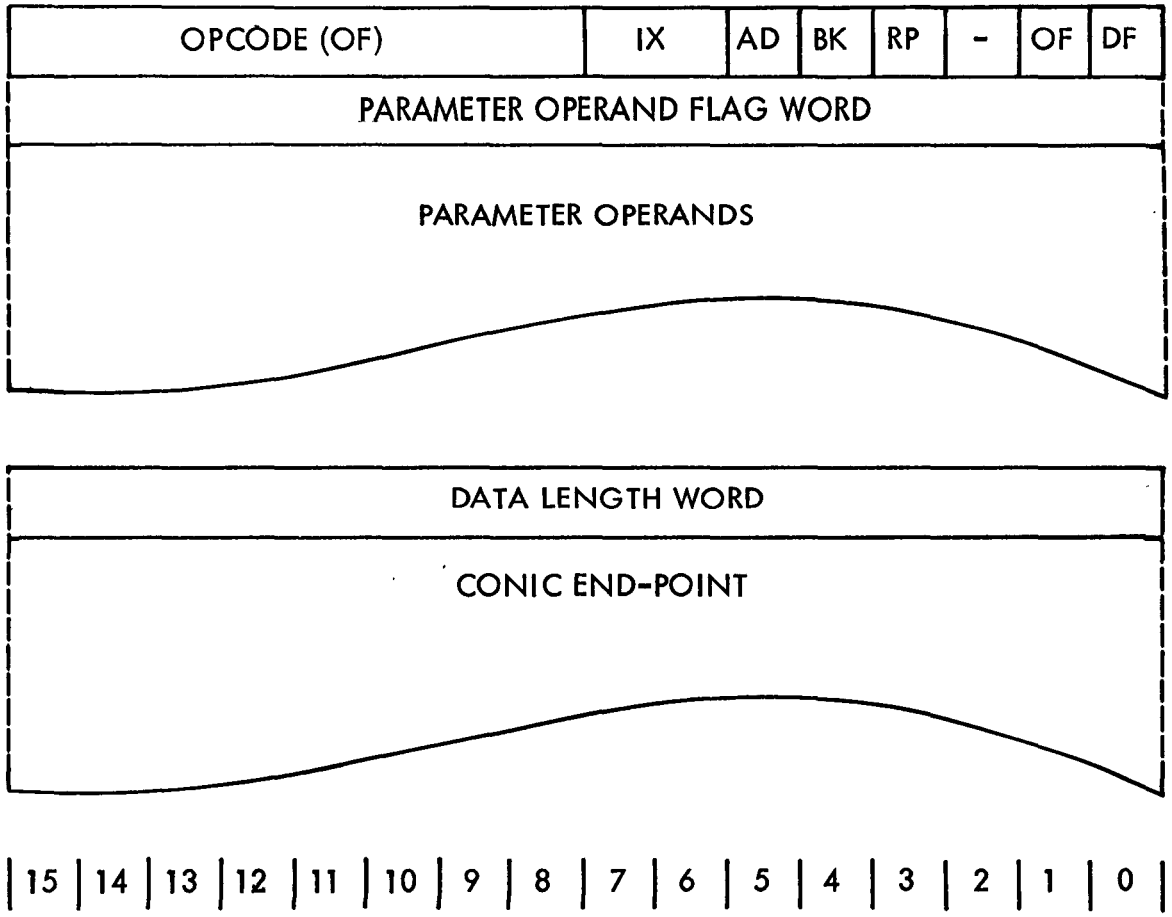
3-51 RM-CON Conics Option Instruction

The RM-CON conics option firmware implements the WRITE CONIC (WC) instruction for generation of parabolas, hyperbolas, ellipses, and circles. The generalized conic equation used in this implementation is

$$Ax^2 + By^2 + Cxy + Dx + Ey = 0.$$

The RM-GRA graphics option firmware is a prerequisite for installation of the RM-CON conics option firmware package. The RM-CON firmware resides within the internal 8080 processor's address space at locations $2C00_{16}$ through $33FF_{16}$.

3-52 WRITE CONIC Instruction (WC)



The WC instruction is a normal-format instruction which is part of the RM-CON conics firmware option. The WC instruction draws circles, parabolas, ellipses and hyperbolas or sections thereof, based on the CONIC-EQUATION parameter operands. The (reduced) generalized conic equation used is:

$$Ax^2 + By^2 + Cxy + Dx + Ey = 0$$

where A, B, C, D and E represent the first 5 CONIC-EQUATION parameter operands. The last CONIC-EQUATION parameter operand K represents the total number of pixels to be generated by the WC instruction for the conic specified. A default value of 1280_{10} is used for K.

The START-POINT parameter operand is used by the WC instruction to define the Cartesian origin of the conic or conic section to be drawn. All conics drawn will pass through the Cartesian origin defined by START-POINT. If a conic end-point is specified, the absolute coordinates of this pixel are compared to the coordinates of each pixel in the conic as it is being generated. If a pixel in the conic being drawn has the same coordinates as specified by the end-point, the conic will be terminated at that point.

The WC instruction is not affected by the WINDOW parameter operand.

Appendix B describes in detail the use of the WC instruction for generation of circles, ellipses, parabolas and hyperbolas.

PERTINENT CONTROL BITS

IX	defines the address mode in which the INDEX 1, INDEX 2 and START-POINT parameter operands set in the WC instruction are evaluated. Also, if a conic end-point is specified as data in the WC instruction, IX effects the address in which this coordinate will be evaluated.
BK	defines the selection of FOREGROUND or BACKGROUND value for the color or intensity value of the conic being drawn. If BK = 0, the conic will be generated with the FOREGROUND value. Otherwise, the BACKGROUND value will be used when BK = 1.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL	specifies the subchannels (i.e., refresh memory bit planes) which are write-enabled to receive conic data.
FOREGROUND	defines the color or intensity value of the conic when BK = 0.
BACKGROUND	defines the color or intensity value of the conid when BK = 1.
INDEX 1	displaces the values to be used for the INDEX 2 and START-POINT parameter operands, as well as the conic

	end-point if any of these are set in the WC instruction and if IX = 01.
INDEX 2	displaces the values to be used for the START-POINT parameter operand, as well as the conic end-point if any of these are set in the WC instruction and if IX = 10.
CONIC-EQUATION	specifies the conic-defining coefficients A, B, C, D and E and the number of pixels, K, to be drawn in the conic.
START-POINT	defines the origin of the conic to be generated. START-POINT is evaluated in the address mode defined by IX.

DATA LENGTH WORD

The DATA LENGTH WORD defines the total number of bytes contained in the WC following the DATA LENGTH WORD. Since only one conic can be generated per WC instruction, it is suggested that the DATA LENGTH WORD (if present due to DF = 1) always takes on a value of four (4), indicating that a single conic end-point is present in the WC instruction data stream. If more the four (4) bytes are indicated by the DATA LENGTH WORD, all additional bytes will be ignored. Since coordinates are composed of full-words, it is necessary that the DATA LENGTH WORD always reflects an even number of bytes.

DATA FORMAT

The first 16-bit word pair is interpreted as the conic end-point. All remaining data is discarded. The first 16-bit word is the element coordinate of the conic end-point. The second is the line coordinate. The conic end-point coordinate is interpreted in the address mode defined by IX.

POSSIBLE ERROR CONDITIONS

It is possible to generate a conic equation from the coefficients A, B, C, D and E which will not be drawn accurately. The conditions for this case are defined in Appendix B.

COP MOVEMENT

The current operating point after execution of the WC instruction is the coordinate of the last pixel of the conic generated.

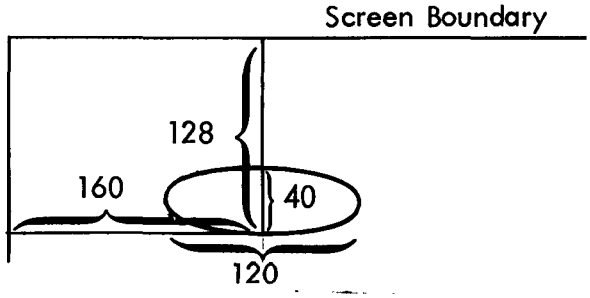
EXAMPLE

It should be noted that the conic arguments are each two words long. The first word of each argument is ignored by the RM-9000. Only the second word of each argument is taken to be meaningful. Negative values are taken to be 2's-complement, limiting the range of the arguments to integers between $7FFF_{16}$ and -8000_{16}.

With all of these things in mind, the entire complex instruction for the ellipse could be formatted as follows:

0F02	Write conic instruction
9000	Write conic and start point operand flags
0000	First null word
0006	Value of A
0000	Second null word
0029	Value of B
0000	Null word
0000	Value of C
0000	Null word
0000	Value of D
0000	Null word
FC01	Value of E
0000	Null word
0000	Value of K
00A0	Value of X start point
0080	Value of Y start point
0004	Data flag word indicating 4 bytes to follow
00A0	Value of X end point
0080	Value of Y end point

The image drawn on the screen would look similarly as follows:

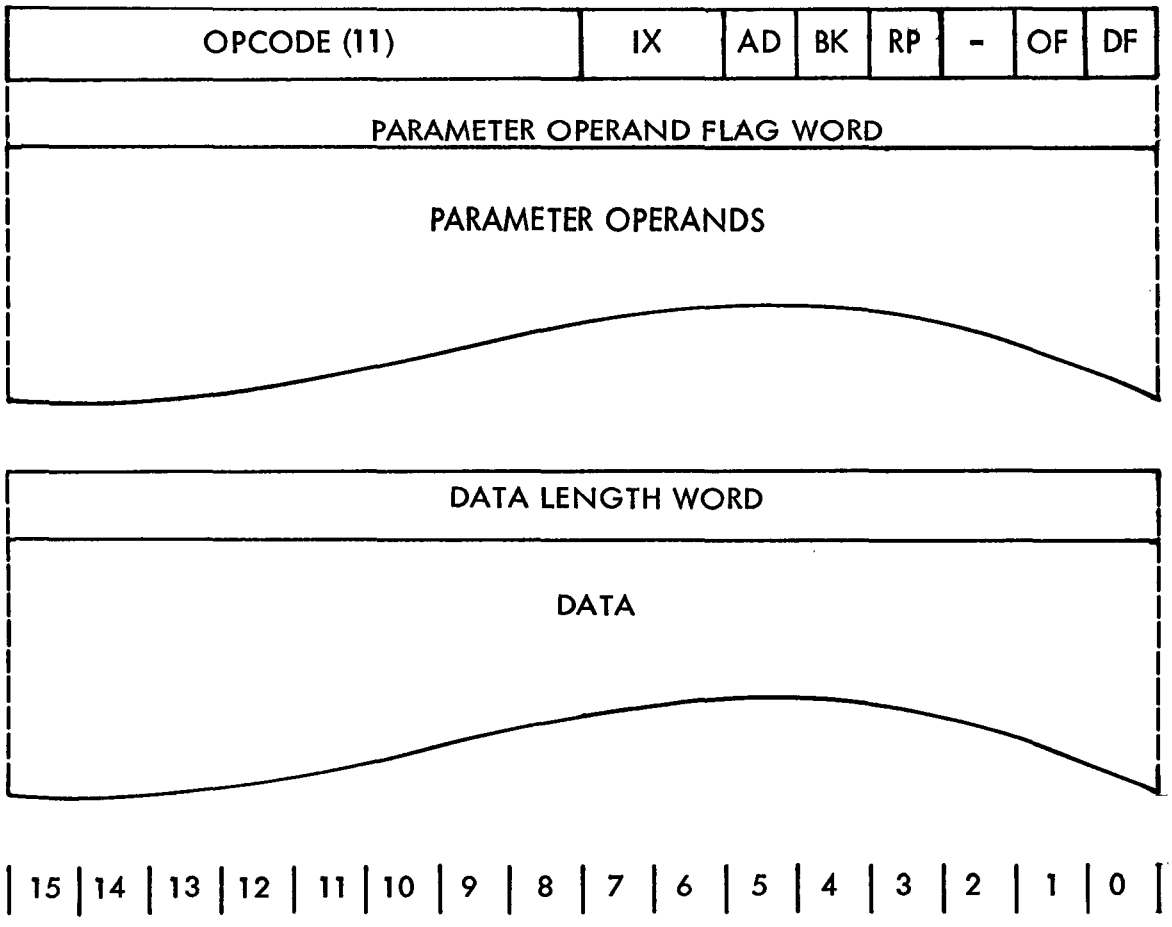


3-53 RM-SCR Scroll Option Instructions

The RM-SCR scroll option firmware implements the SCROLL X and the SCROLL Y normal-format instructions. This package allows the scrolling (or moving) of sub-channel-selectable image data within any rectangular region of refresh memory. This operation can be performed in either the X or the Y axis.

The RM-SCR option resides within the internal 8080 processor's address space at locations 1400_{16} through $17FF_{16}$.

3-54 SCROLL X Instruction (SCRX)



The SCRX instruction is a normal-format instruction which is part of the RM-SCR scroll firmware option. The SCRX performs a scroll of image data within the rectangular region defined by the WINDOW parameter operand along the X-axis. The sign of the SCROLL-COUNT parameter operand defines the direction of scroll within the WINDOW region along the X-axis. If SCROLL-COUNT is positive, scrolling will be to the right, i.e., towards increasing element coordinate values; if SCROLL-COUNT is negative, scrolling will be to the left, i.e., towards decreasing element coordinate values. The absolute value of SCROLL-COUNT defines the number of elements which the image data within WINDOW region will be moved. A SCROLL-COUNT of zero will result in no scrolling operation. Only those refresh memory planes selected by the SUB-CHANNELS will be scrolled.

Data which is scrolled out of the window is discarded. If BK=0, the data which is scrolled into the WINDOW region is taken from the BACKGROUND parameter operand; if BK=1, the fill data is taken from the FOREGROUND parameter operand. If the absolute value of SCROLL-COUNT is greater than the WINDOW width, the result will be erasure of the WINDOW region to either the FOREGROUND or BACKGROUND value dependent on the value of BK.

PERTINENT CONTROL BITS

IX	Defines the address mode in which the INDEX1, INDEX2 and WINDOW parameter operands are evaluated.
BK	Defines the color or intensity value to be used to fill the sub-region of the WINDOW region from which data is scrolled. If BK=0, the BACKGROUND value is used; otherwise, if BK=1, the FOREGROUND value is used.
AD	Allows the user to inhibit filling of the WINDOW region from which image data has been scrolled. If AD=1, this erasure will not be performed; otherwise, the erasure will take place if AD=0.

PERTINENT PARAMETER OPERANDS

SUBCHANNELS	Defines the refresh memory planes which will be scrolled and which will receive fill data (when AD=0).
FOREGROUND	Defines the color or intensity value to be used as fill data when BK=1 and AD=0.
BACKGROUND	Defines the color or intensity value to be used as fill data when BK=0 and AD=0.
INDEX1	Displaces the values to be used for the INDEX2 and WINDOW parameter operands when IX=01.
INDEX2	Displaces the values to be used for the WINDOW parameter operand when IX=10.

WINDOW	Defines the rectangular region inside of which image data will be horizontally scrolled.
SCROLL-COUNT	Defines the number of elements to scroll image data within the rectangular WINDOW region. SCROLL-COUNT is defined as a 2's complement number whose absolute value is used as the number of scrolling elements. If SCROLL-COUNT is positive, scrolling is to the right, and if SCROLL-COUNT is negative, scrolling is to the left. A SCROLL-COUNT of 0 results in no scrolling operation.

DATA LENGTH WORD

The DATA LENGTH WORD defines the number of bytes of data present in the SCRX instruction stream when DF=1.

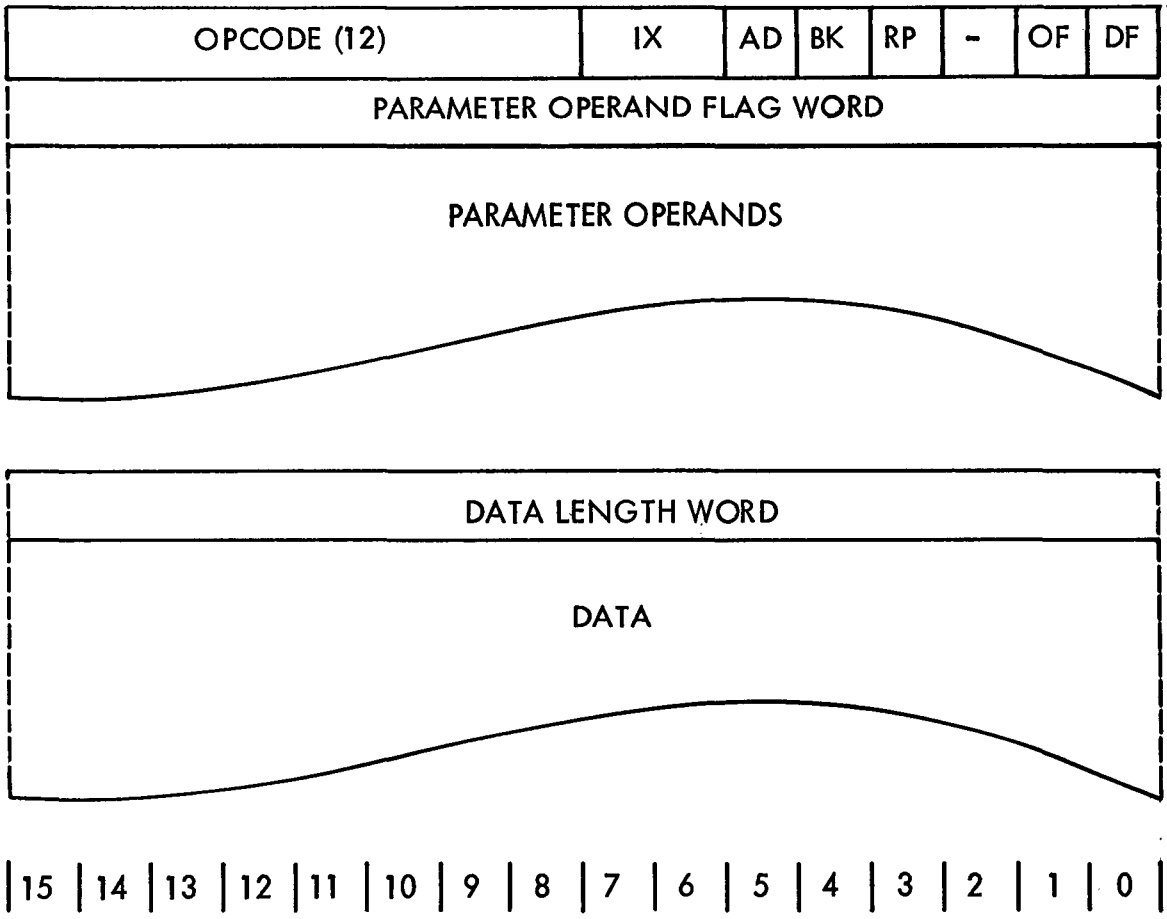
DATA FORMAT

All data present in the SCRX instruction is discarded.

COP MOVEMENT

The COP will remain unchanged during the execution of the SCRX instruction.

3-55 SCROLL Y Instruction (SCRY)



The SCRY instruction is a normal-format instruction which is part of the RM-SCR scroll firmware option. The SCRY performs a scroll of image data within the rectangular region defined by the WINDOW parameter operand along the Y-axis. The sign of the SCROLL-COUNT parameter operand defines the direction of scroll within the WINDOW region along the Y-axis. If SCROLL-COUNT is positive, scrolling will be towards the bottom of the screen, i.e., towards increasing line coordinate values. If SCROLL-COUNT is negative, scrolling will be towards the top of the screen, i.e., toward decreasing line coordinate values. The absolute value of SCROLL-COUNT defines the number of lines which the image data within WINDOW region will be moved. A SCROLL-COUNT of zero will result in no scrolling operation. Only those refresh memory planes selected by the SUBCHANNELS will be scrolled.

Data which is scrolled out of the window is discarded. If BK=0, the data which is scrolled into the WINDOW region is taken from the BACKGROUND parameter operand. If BK=1, the fill data is taken from the FOREGROUND parameter operand. If the absolute value of SCROLL-COUNT is greater than the WINDOW width, the result will be erasure of the WINDOW region to either the FOREGROUND or BACKGROUND value dependent on the value of BK.

PERTINENT CONTROL BITS

IX	Defines the address mode in which the INDEX1, INDEX2 and WINDOW parameter operands are evaluated.
BK	Defines the color or intensity value to be used to fill the sub-region of the WINDOW region from which data is scrolled. If BK=0, the BACKGROUND value is used. Otherwise, if BK=1, the FOREGROUND value is used.
AD	Allows the user to inhibit filling of the WINDOW region from which image data has been scrolled. If AD=1, this erasure will not be performed. Otherwise, the erasure will take place if AD=0.

PERTINENT PARAMETER OPERANDS

SUBCHANNELS	Defines the refresh memory but planes which will be scrolled and which will receive fill data (when AD=0).
FOREGROUND	Defines the color or intensity value to be used as fill data when BK=1 and AD=0.
BACKGROUND	Defines the color or intensity value to be used as fill data when BK=0 and AD=0.
INDEX1	Displaces the values to be used for the INDEX2 and WINDOW parameter operands when IX=01.
INDEX2	Displaces the values to be used for the WINDOW parameter operand when IX=10.
WINDOW	Defines the rectangular region inside of which image data will be vertically scrolled.

SCROLL-COUNT — Defines the number of lines to scroll image data within the rectangular WINDOW region. SCROLL-COUNT is defined as a 2's complement number whose absolute value is used as the number of scrolling lines. If SCROLL-COUNT is positive, scrolling is to the bottom of the screen, and if SCROLL-COUNT is negative, scrolling is to the top of the screen. A SCROLL-COUNT of 0 results in no scrolling operation.

DATA LENGTH WORD

The DATA LENGTH WORD defines the number of bytes of data present in the SCRY instruction stream when DF=1.

DATA FORMAT

All data present in the SCRY instruction is discarded.

COP MO VEMENT

The COP will remain unchanged during the execution of the SCRY instruction.

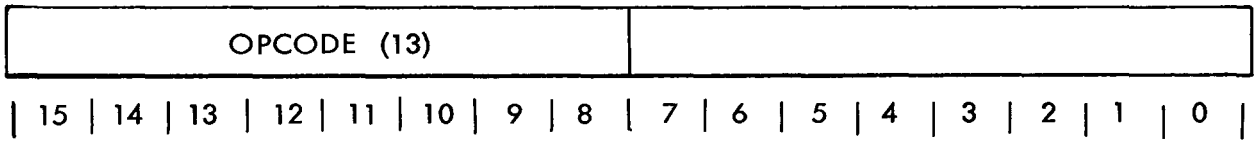
3-56 RM-STA Status Management Option Instructions

The RM-STA status management option firmware implements the SAVE ENVIRONMENT (PUSHE) and the RESTORE ENVIRONMENT (POPE) normal-format instructions. These two instructions provide for a multiple user (or level) environment by temporarily storing and subsequently retrieving display system status from an internal stack. For example, consider the case where Program A (PRGA) is writing an image to refresh memory. At a given point in time, Program B (PRGB) is initiated via keystroke input. PRGB is higher in priority than PRGA and performs the task of echoing keyboard input to the monitor. Therefore, PRGA is suspended (following completion of the current instruction) while PRGB operates. Because PRGB will write a character to a different point on the screen, display system status (environment) will be affected. Thus, PRGA's status must be stored and then restored before and after execution of PRGB. This can be accomplished with some difficulty in the host CPU, or very easily in the display system, using the status management instructions.

A constraint on the use of the status management option is that, since the use of the RM-SCA and RM-LAF options require large buffers to preserve data for each line of data being processed, partial lines of scaled or LAF-processed data will be lost when a given environment is saved via the PUSHE instruction and restored via the POPE instruction.

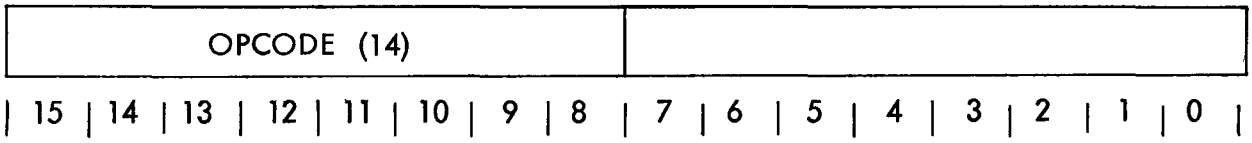
The RM-STA option resides within the internal 8080 processor's address space at locations 1000_{16} through $13FF_{16}$.

3-57 SAVE ENVIRONMENT Instruction (PUSHE)



The PUSHE instruction is a special-format instruction which is part of the RM-STA status management firmware option. The PUSHE instruction saves all parameter operands as well as associated internal parameters on an internal stack. This internal stack is a LIFO (last-in-first-out) structure. Up to 5 consecutive PUSHE instructions without intervening POPE instructions are allowed (i.e., up to 5 environments may be saved on the internal stack). If more than 5 are attempted by the user, an illegal instruction interrupt will be generated, and the current environment will not be saved on the internal stack.

3-58 RESTORE ENVIRONMENT Instruction (POPE)



The POPE instruction is a special-format instruction which is part of the RM-STA status management firmware option. The POPE instruction restores all parameter operands as well as associated internal parameters from an internal stack. This internal stack is a LIFO (last-in-first-out) structure. The display environment information which is restored from the internal stack will be the last set of parameter operands which was stored on the internal stack by a PUSHE instruction. If there is no display environment information stored on the stack, an illegal instruction interrupt will be generated and the current display environment will remain unchanged.

3-59 RM-FNT Programmable Font Option Instructions

The RM-FNT programmable font option firmware implements the LOAD PROGRAMMABLE FONT and LOAD PROGRAMMABLE FONT WITH REVERSE PACKING special-format instructions for 128_{10} characters. The RM-FNT package contains the firmware necessary to attach the programmable font character generation routines at power-on and to generate the characters from programmable font RAM in both unscaled and scaled modes.

The font matrix which can be stored using the RM-FNT option is 8 bits wide by 12 bits high. Symbols or characters larger than 8 by 12 can be generated by the user by combining one or more smaller (i.e., 8 by 12) characters, or by using the WR instruction.

At power-on or after receiving a RESET instruction, the character font from the PROM stored on the control board is copied to the RAM associated with the RM-FNT option for ASCII codes 20_{16} through $5F_{16}$. The font RAM associated with ASCII codes 60_{16} through $9F_{16}$ are not initialized and therefore contain either random data (after power-on) or contain previously loaded values. Since the DIMENSION and SPACING values remain 7 by 9, the use of RM-FNT is transparent to the user.

The RAM associated with the RM-FNT option is located from 4400_{16} through $47FF_{16}$ and use 12_{10} bytes per ASCII code. The RM-FNT option firmware resides within the internal 8080 processor's address space at locations 1800 through $1BFF_{16}$.

3-60 LOAD PROGRAMMABLE FONT Instruction (LPF)

OPCODE (15)	CHARACTER CODE
FONT LINE 1	FONT LINE 2
FONT LINE 3	FONT LINE 4
FONT LINE 5	FONT LINE 6
FONT LINE 7	FONT LINE 8
FONT LINE 9	FONT LINE 10
FONT LINE 11	FONT LINE 12

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

The LPF instruction is a special-format instruction which is part of the RM-FNT programmable font option firmware. The LPF instruction loads the 8 by 12 font which will be sent to display refresh memory when the byte value defined by CHARACTER CODE is transmitted as text data in a WT instruction. Although the standard ASCII codes (20_{16} through $5F_{16}$) are copied to the programmable font storage RAM, these codes may be overwritten at any time by the user.

ARGUMENT DEFINITION

CHARACTER CODE — Specifies any of 128_{10} character codes to be loaded with font data for subsequent character or symbol generation. The legal codes which may be loaded via the LPF instruction are 20_{16} through $9F_{16}$. Character codes from 0 through $1F_{16}$ and $A0_{16}$ through FF_{16} are not legal and the font data associated with these codes will be discarded.

DATA DEFINITION

Twelve bytes of font-definition data follow immediately after the opcode/character code word. Font line 1 defines the top line of the 8 by 12 font matrix and font line 12 defines the bottom line of the font matrix. The high-order bit

of each font byte represents the left margin of the font matrix and the low-order bit represents the right margin of the font matrix.

EXAMPLE 3-2

If the symbol described by Figure 3-8 were to be generated using the RM-FNT option at (0,0), the following sequence might be used for character code 9F.

```
0500        ; RESET
159F        ; LPF + CHAR. CODE
0010        ; FONT LINE 1 & 2
3854        ; FONT LINE 3 & 4
9210        ; FONT LINE 5 & 6
1010        ; FONT LINE 7 & 8
2800        ; FONT LINE 9 & 10
0000        ; FONT LINE 11 & 12
0C01        ; WRITE TEXT WITH DATA FLAG
0001        ; NUMBER OF BYTES = 1
9F00        ; CHARACTER CODE
```

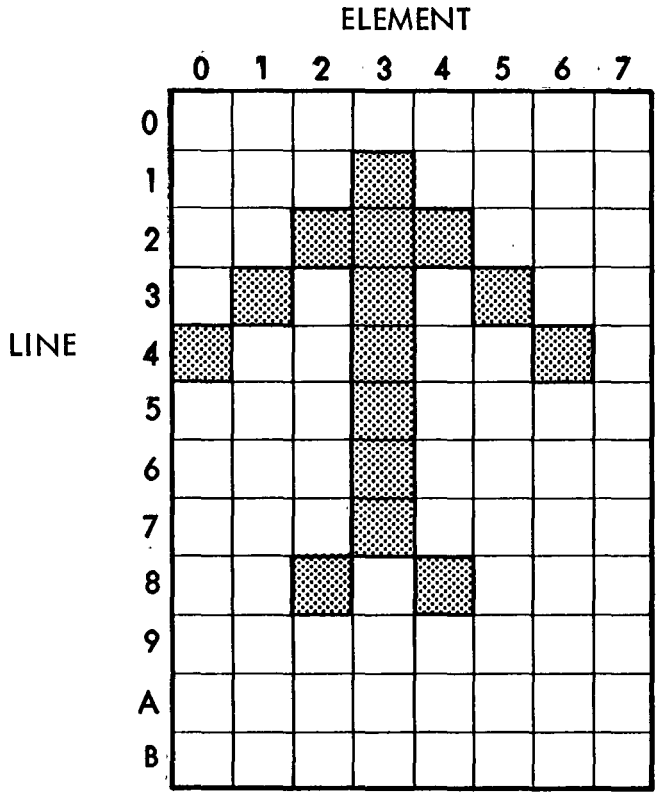


Figure 3-8 Programmable Font Example

3-61 LOAD PROGRAMMABLE FONT WITH REVERSE PACKING Instruction (LPFRP)

OPCODE (21)	CHARACTER CODE
FONT LINE 2	FONT LINE 1
FONT LINE 4	FONT LINE 3
FONT LINE 6	FONT LINE 5
FONT LINE 8	FONT LINE 7
FONT LINE 10	FONT LINE 9
FONT LINE 12	FONT LINE 11

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

The LPFRP instruction is a special-format instruction which is part of the RM-FNT programmable font option firmware. The operation of the LPFRP instruction is exactly the same as the LPF instruction (See Section 3-59) except that the font line description bytes are reversed within each word of data.

EXAMPLE 3-3

The symbol described in Figure 3-8 would be generated using the following sequence with the LPFRP instruction:

```
0500        ; RESET
059F        ; LPF + CHAR. CODE = 9F16
1000        ; FONT LINES 2 & 1
5438        ; FONT LINES 4 & 3
1092        ; FONT LINES 6 & 5
1010        ; FONT LINES 8 & 7
0028        ; FONT LINES 10 & 9
0000        ; FONT LINES 12 & 11
0C01        ; WRITE TEXT + DATA FLAG
0001        ; NUMBER OF BYTES = 1
9F00        ; CHARACTER CODE
```

3-62 RM-PER Interactive Peripherals Option Instructions

The RM-PER interactive peripherals option firmware implements the following RM-9000 instructions:

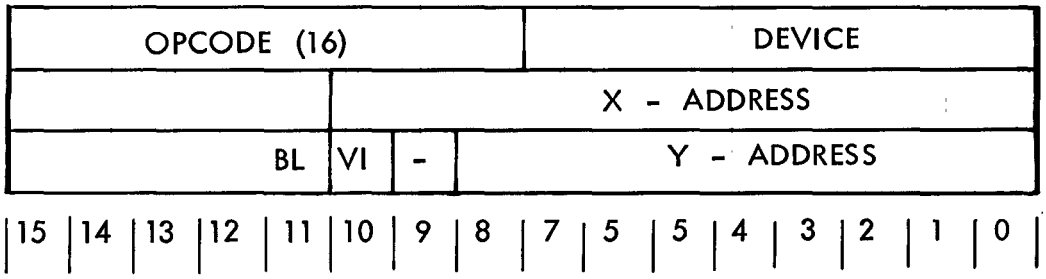
- WRITE CURSOR STATE controlling the position and visible/blink state of any cursors
- READ CURSOR STATUS transmitting the position and visible/blink state of any cursors to the host processor
- WRITE KEYBOARD outputting a byte to one (1) of eight (8) possible serial output devices
- READ KEYBOARD transmitting a received character from any of eight (8) possible serial input devices to the host processor
- SENSE PERIPHERAL STATUS transmitting activity status from either RM-SLC serial link card to the host processor.

The RM-PER option handles all interactive keyboard and cursor functions, generating the appropriate interrupts to the host processor. Buffering of serial input characters up to sixteen (16) characters per device is also handled by the RM-PER option.

The maximum number of RM-SLC serial link cards which can exist in an RM-9000 display system is two (2). A single RM-SLC serial link card may be configured for either four (4) serial I/O ports or for two (2) serial I/O ports and two (2) cursors. This indicates that several different interactive configurations are possible. It is necessary for the user to know the exact RM-SLC configuration of the particular RM-9000 display system and the device addressing scheme used by the RM-PER option. This will avoid software problems caused by improper device addressing.

The RM-PER option resides within the internal 8080 processor's address space at locations $0C00_{16}$ through $0FFF_{16}$.

3-63 Write Cursor State Instruction (WCS)



The WCS instruction is a special-format instruction which is part of the RM-PER interactive peripherals option firmware. The WCS instruction is used to define the position of any of four (4) cursors in the RM-9000 display as well as it's visible/invisible and blink/no-blink states. No errors are possible, since the attempted addressing of non-existent cursors will take place, but will have no effect upon the display system.

PERTINENT ARGUMENTS

DEVICE — Specifies the cursor whose state is to be defined by the instruction. The DEVICE assignments are as follows:

DEVICE	CURSOR
0	Cursor #1 of SLC Board #1
1	Cursor #2 of SLC Board #2
2	Cursor #1 of SLC Board #2
3	Cursor #2 of SLC Board #2

X-ADDRESS — Defines the element coordinate at which the cursor specified by device will be positioned. Note that due to mechanism used to superimpose the cursor onto the refresh memory video, there is a region of elements in which the cursor will be off of the visible screen area. These areas are as follows:

SYSTEM TYPE	REGION (IN ELEMENTS)
RM-9000	320_{10} through 511_{10}
RM-9200	640_{10} through 1023_{10}
RM-9300	640_{10} through 1023_{10}

Y-ADDRESS — Defines the line coordinate at which the cursor specified by DEVICE will be positioned. For low line resolution systems (i.e., the RM-9100 and RM-9200), cursor line addressing is module 256_{10}, i.e., line addresses 256_{10} through 511_{10} correspond to line address 0 through 225_{10}, respectively.

VI — Defines the visible or invisible state of the cursor specified by DEVICE. If VI = 0, the cursor will be made invisible; if VI = 1, the cursor will be made visible.

BL — Defines the blink state of the cursor specified by DEVICE. If BL = 0, the cursor will be made non-blinking; if BL = 1, the cursor will blink at a rate of one (1) Hz.

NOTE

THE CURSOR BLINK AND VISIBLE STATE CAN ALSO BE MANUALLY CONTROLLED BY A JOYSTICK OR TRACKBALL IF EITHER OF THESE DEVICES EXIST IN A GIVEN CONFIGURATION. THE VI AND BL STATE MAY BE MANUALLY CHANGED IN THIS CASE WITH NO INDICATION BEING MADE TO THE HOST PROCESSOR. (SEE APPENDIX C FOR OPERATIONAL CHARACTERISTICS OF THE JOYSTICK AND TRACKBALL IN THIS CONTEXT.

3-64 Read Cursor Status Instruction (RCS)

OPCODE (17)	DEVICE

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

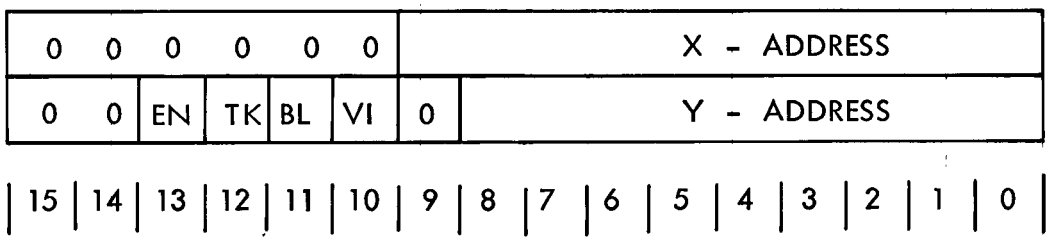
The RCS instruction is a special-format instruction which is part of the RM-PER interactive peripherals option firmware. The RCS instruction conditions the RM-9000 for the transfer of two (2) words of cursor status information to the host processor. After the RCS instruction has been transferred from the host processor to the RM-9000, it is the responsibility of the host processor to initiate a two (2) 16-bit word data transfer from the RM-9000. (See Addenda on the operation of specific CPU interfaces.) If these two (2) words are not read back properly by the host processor, it is possible that the inter-processor communications sequence may become hung. This condition can only be rectified using a hard system reset.

PERTINENT ARGUMENTS

DEVICE — Specifies the cursor whose status is to be read back to the host processor after the RCS instruction has been issued. The DEVICE assignment is as follows:

DEVICE	CURSOR
0	Cursor #1 of SLC Board #1
1	Cursor #2 of SLC Board #1
2	Cursor #1 of SLC Board #2
3	Cursor #2 of SLC Board #2

DATA FORMAT — The format of the two data words to be read back to the host processor is as follows:



X-ADDRESS — Specifies the current element coordinate of the cusor specified by DEVICE.

Y-ADDRESS — Specifies the current line coordinate of the cursor specified by DEVICE.

VI — Reflects the current visible/invisible state of the cursor selected by DEVICE. If VI = 0, the cursor is not visible; if VI = 1, the cursor is currently visible.

TK — Reflects the current state of the two-position TRACK switch on the joystick or trackball if present. If TK = 0, the TRACK function is not enabled; if TK = 1, the TRACK function is enabled.

BL — Reflects the blinking state of the cursor selected by DEVICE. If BL = 0, the cursor is non-blinking; if BL = 1, the cursor is blinking at a rate of approximately one (1) second.

EN — Reflects the state of momentary-action switch on the joystick or trackball if present. If EN = 0, the ENTER switch is not depressed; if EN = 1, the ENTER switch is currently depressed.

3-65 Write Keyboard Instruction (WKB)

OPCODE (18)	DEVICE
-	CHARACTER CODE

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

The WKB instruction is a special-format instruction which is part of the RM-PER interactive peripherals option firmware. The WKB instruction writes a single 8-bit character to one (1) of eight (8) possible serial output devices within the RM-9000 system. No buffering of output characters is performed; therefore, it is necessary to wait for the serial transmission of the character to complete. A transmitter interrupt request will be generated by the RM-9000 to the host processor if enabled. The serial peripheral status word should then be read by the host processor using the SENSE PERIPHERAL STATUS instruction to verify that the expected transmission was completed (See Section 3-66). It is then allowable to output another data byte to the serial output device specified by DEVICE. It is not necessary to wait for completion of a transmission before outputting to a different serial output device.

The RM-9000 keyboard is an I/O device. That is, not only can characters be input from it, they may also be output to it. Specifically, one may ring the bell and illuminate or extinguish any of the 16 LEDs along the top of the keyboard.

PERTINENT ARGUMENTS

DEVICE — Specifies the serial output device to which the CHARACTER CODE will be output. The DEVICE code assignment is as follows:

DEVICE	OUTPUT PORT
0	Serial Output Port #1 of SLC Board #1
1	Serial Output Port #2 of SLC Board #1
2	Serial Output Port #3 of SLC Board #1
3	Serial Output Port #4 of SLC Board #1
4	Serial Output Port #1 of SLC Board #2
5	Serial Output Port #2 of SLC Board #2
6	Serial Output Port #3 of SLC Board #2
7	Serial Output Port #4 of SLC Board #2

CHARACTER CODE

Defines the 8-bit byte of data which will be output to the serial output port specified by DEVICE. All values from 0 through FF_{16} are legal.

3-66 Read Keyboard Instruction (RKB)

OPCODE (19)	DEVICE

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

The RKB instruction is a special-format instruction which is part of the RM-PER interactive peripherals option firmware. The RKB instruction conditions the RM-9000 for the transfer of one word of serial data from the serial port input buffer specified by DEVICE. After the RKB instruction has been transferred from the host processor to the RM-9000, it is the responsibility of the host processor to read back one (1) 16-bit data word from the RM-9000. If the word of data is not read back properly by the host processor, it is possible that the interprocessor communications sequence may become hung. This condition can only be rectified using a hard system reset.

For each of the eight (8) possible serial input ports, there exists an internal buffer which can handle up to sixteen (16) bytes of serial input data, as well as corresponding status information. A receiver interrupt request will be generated to the host processor (if enabled) for each data byte received from a serial input port. After interrogation of the serial peripheral status words using the SPS instruction, the RKB instruction could be used to read the data byte received. Reading a receiver register clears the corresponding bit in the peripheral status register.

All input buffers are cleared upon the power-on or the hard reset condition. All characters received prior to this will be lost.

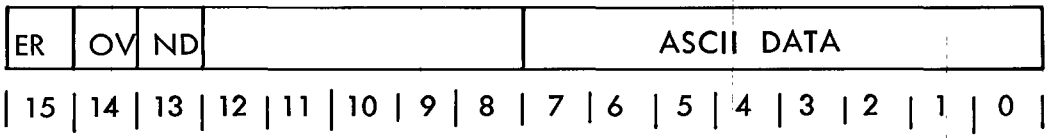
PERTINENT ARGUMENTS

DEVICE — Specifies the serial input port buffer from which data will be read along with the corresponding status bits. The DEVICE assignment is as follows:

DEVICE	INPUT PORT
0	Serial Input Port #1 of SLC Board #1
1	Serial Input Port #2 of SLC Board #1
2	Serial Input Port #3 of SLC Board #1
3	Serial Input Port #4 of SLC Board #1

4	Serial Input Port #1 of SLC Board #2
5	Serial Input Port #2 of SLC Board #2
6	Serial Input Port #3 of SLC Board #2
7	Serial Input Port #4 of SLC Board #2

DATA FORMAT — The format of the data word read back to the host processor is as follows:



DATA VALUE — Reflects the 8-bit data value input to the RM-9000 display processor from the external serial device

ND — Reflects a 'no data available' condition. If ND = 0 and ER = 0, the DATA VALUE represents a valid received data byte. If ND = 1, no data is present in the input buffer.

OV — Reflects a 'data buffer overflow' condition. If OV = 1, data was received from the serial input port but no room was available in the input buffer, i.e. data has been lost.

ER — Reflects a hardware error condition. If ER = 1, a hard error has been detected by the input UART in attempting to obtain a character from the external device.

3-67 Sense Peripheral Status Instruction (SPS)

OPCODE (1A)	DEVICE

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

The SPS instruction is a special-format instruction which is part of the RM-PER interactive peripherals option firmware. The SPS instruction conditions the RM-9000 for the transfer of one word of serial peripheral board status from the RM-SLC serial link board specified by DEVICE. After the SPS instruction has been transferred from the host processor to the RM-9000, it is the responsibility of the host processor to readback one (1) 16-bit word from the RM-9000. If the word of data is not readback properly by the host processor, it is possible that the interprocessor communication sequence may become hung. This condition can only be rectified using a hard system reset.

The data word readback from the RM-9000 reflects the status of the RM-SLC board selected by DEVICE. This status indicates which interactive device is currently requesting service or is available for output. Upon receipt of a transmitter or receiver interrupt, this data word defines the interrupting source.

PERTINENT ARGUMENTS

DEVICE — Specified the RM-SLC board whose status is to be interrogated. If DEVICE = 0, the RM-SLC Board 1 is to be interrogated. If DEVICE = 1, the RM-SLC Board 2 is to be interrogated.

DATA FORMAT

The format of the serial peripheral status word is as follows:

—	C1	C0	X_3	R_3	X_2	R_2	X_1	R_1	X_0	R_0	SLC BOARD 1
—	C3	C2	X_7	R_7	X_6	R_6	X_5	R_5	X_4	R_4	SLC BOARD 2

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

$C_m = 1$ Indicates that one of two events has occurred:

- The ENTER momentary action switch has been depressed with cursor m (where m = 0 through 3) selected on the joystick or trackball, or
- The coordinates of cursor m have changed with cursor m selected on the joystick or trackball and with the TRACK switch enabled. C will remain set to one (1) until the coordinates of cursor m have been read using the RCS instruction, at which time C will be reset to zero (0).

$X_n = 0$ Indicates that a serial transmission initiated by a WKB instruction has not yet completed at output port n (where n = 0 through 7). X_n will be set to one (1) when transmission is completed.

$R_n = 1$ Indicates that a data byte has been received by the serial input port n (where n = 0 through 7). R will be reset to zero (0) when the data byte along with its status has been readback after issuing an RKB instruction.

3-68 RM-USR User Subroutine Option Instructions

The RM-USR user subroutine option firmware implements the following six (6) instructions:

- LOAD CONTROL MEMORY and LOAD CONTROL MEMORY WITH REVERSE PACKING which store either executable 8080 processor instructions or RM-9000 display instructions into RM-MOC memory expansion RAM
- READ CONTROL MEMORY and READ CONTROL MEMORY WITH REVERSE PACKING which retrieve data from locations 000 through 7FFF of the RM-9000 display system internal 8080 address space
- CALL CONTROL MEMORY which executes any 8080 subroutine stored in the 8080 processor's address space
- EXECUTE INSTRUCTION MEMORY which executes one or more RM-9000 display instructions stored in the 8080 processor's address space.

The RM-USR option, in general, implement a writeable control store capability in that the host processor can load, retrieve and execute special-purpose subrountines and display instruction lists. The Call Control Memory instruction is executed within the RM-9000 display. It has direct access to all display registers and requires an intimate knowledge of the interaction between display registers for proper display generation. The Execute Instruction Memory provides local display list processing using the RM-9000 instruction. This does not require special knowledge of the RM-9000 hardware.

It is necessary to load either subroutine code or RM-9000 display lists into RAM within the RM-9000 display system. The memory map (Table 3-16) defines the areas of RAM and the options packages which use these specific areas. If a given option is not installed in a particular configuration, the RAM region assigned to this option is available for use with the RM-USR user subroutine option firmware. If an area used by an existing option is loaded via the RM-USR option, the operation of the option firmware will be adversely affected. No error checking is performed to restrict the use of areas of RAM.

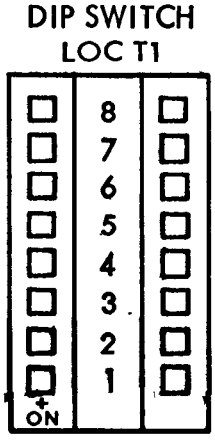
Table 3-16 RM-9000 Internal Memory Map

PROM ADDRESS	DESCRIPTION
0000 - 07FF	RM-IMG STANDARD IMAGER FIRMWARE
0800 - 0BFF	RM-STD STANDARD OPTION SUPPORT
0C00 - 0FFF	RM-PER INTERACTIVE PERIPHERALS OPTION
1000 - 13FF	RM-STA STATUS MANAGEMENT OPTION
1400 - 17FF	RM-SCR SCROLL OPTION
1800 - 1BFF	RM-FNT PROGRAMMABLE FONT OPTION
1C00 - 23FF	RM-SCA SCALING OPTION
2400 - 27FF	RM-LAF LOGICAL/ARITHMETIC FUNCTION OPTION
2800 - 2BFF	RM-GRA GRAPHICS OPTION
2C00 - 33FF	RM-CON CONICS OPTION
3400 - 37FF	RM-USR USER SUBROUTINE OPTION
3800 - 3BFF	RM-MAG MAGNIFY OPTION
6200 - 63FF	STANDARD CHARACTER ROM
6400 - 7BFF	RM-DIA DIAGNOSTIC OPTION

RAM ADDRESS	USAGE
4000 - 414F	RM-PER INPUT BUFFERS
4150 - 417F	RM-GRA INTERNAL VARIABLES
4180 - 41AF	RM-CON INTERNAL VARIABLES
41B0 - 41DF	RM-MAG INTERNAL VARIABLES
41E0 - 41FF	NOT USED
4200 - 438F	RM-STA INTERNAL STACK & VARIABLES
4390 - 439F	NOT USED
43A0 - 43AF	RM-PER INTERNAL VARIABLES
43B0 - 43FF	NOT USED
4400 - 487F	RM-FNT PROG FONT TABLE
4880 - 489F	RM-FNT INTERNAL VARIABLES
48A0 - 48FF	NOT USED
4900 - 491F	RM-SCR INTERNAL VARIABLES
4920 - 495F	NOT USED
4960 - 4AEF	RM-FNT PROG FONT TABLE
4AF0 - 55FF	NOT USED
5600 - 5FFF	RM-SCR, RM-LAF AND RM-MAG INTERNAL BUFFERS
6000 - 60FF	STACK AREA
6100 - 61FF	RM-IMG INTERNAL VARIABLES
7C00 - 7FFF	RM-DIAG DIAGNOSTIC CARD RAM

Ramtek

Table 3-17 Control Board (502450) Dip Switch Settings



NOTES:

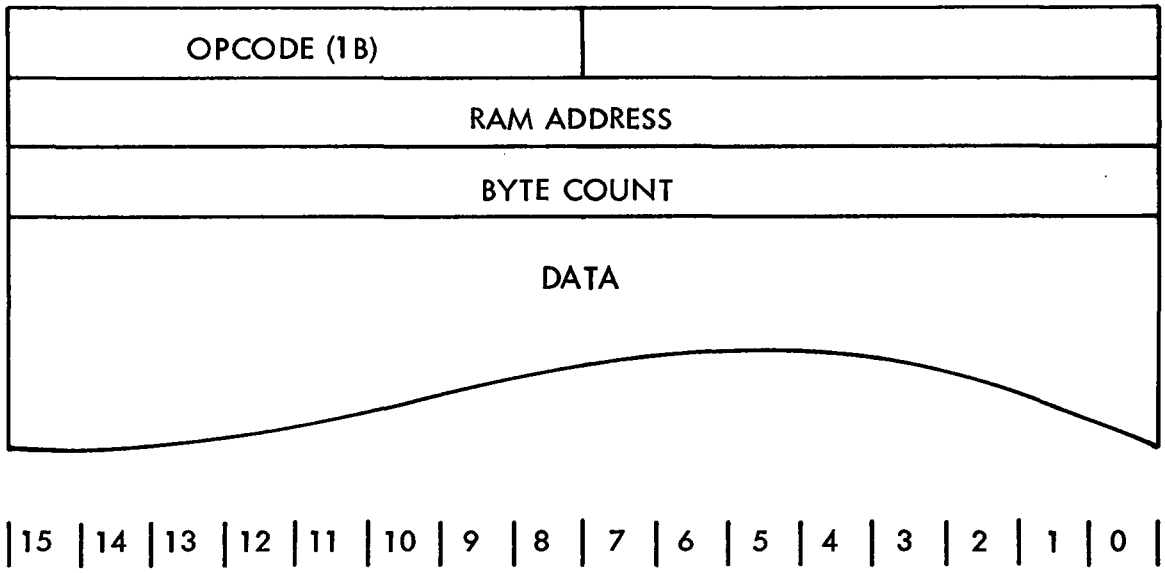
1. SWITCHES 1,4,&5 - NOT USED
2. SWITCH ON (+) = LOGICAL 0
3. SWITCH OFF = LOGICAL 1

SWITCH 2	FUNCTION
0	MEMORY OPTION CARD NOT PRESENT (OR DISABLED)
1	MEMORY OPTION CARD PRESENT (OR ENABLED)

SWITCH 3	FUNCTION
0	ERASE SCREEN ON RESET
1	DO NOT ERASE SCREEN ON RESET

SWITCHES 6	7	8	SYSTEM DEFINITION
0	0	0	RM-9100
1	0	0	RM-9200, RM-9202
1	0	1	RM9300

3-69 Load Control Memory Instruction (LCM)



The LCM instruction is a special-format instruction which is part of the RM-USR user subroutine option firmware. The LCM instruction loads either 8080 microprocessor instructions or RM-9000 display instructions into RAM memory on the RM-MOC memory expansion card. The LCM does not differentiate between 8080 instructions and RM-9000 display instruction. The use of either the CCM or the XIM instruction defines the functional use of any data loaded by the LCM instruction.

PERTINENT ARGUMENTS

RAM ADDRESS	Specifies the address in internal 8080 RAM at which subroutine or display instruction data will be stored. No address validity check is made that the address specified is legal.
BYTE COUNT	Defines the number of bytes of subroutine or display instruction data to be stored in internal 8080 RAM.

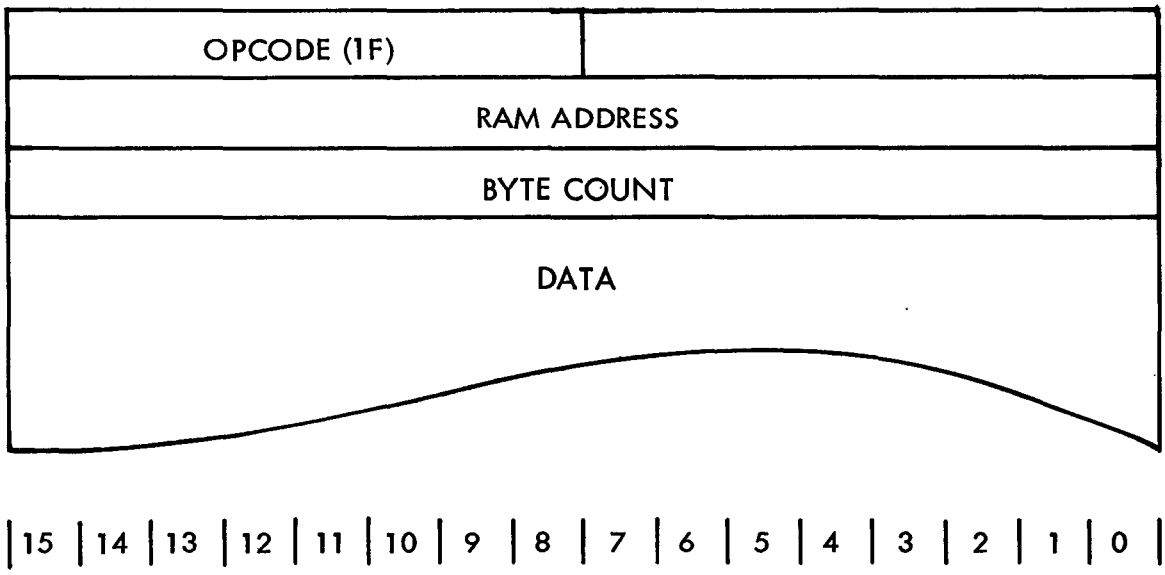
DATA FORMAT

The subroutine or display instruction data is stored into internal 8080 RAM high-order byte first, followed by the low-order byte. If the BYTE COUNT is odd, the low-order byte of the last data word will be discarded.

POSSIBLE ERRORS

If an area of RAM used by an option which exists in a given system is overwritten, the operation of that option cannot be guaranteed unless a reset sequence is initiated.

3-70 Load Control Memory With Reverse Packing Instruction (LCMRP)



The LCMRP instruction is a special-format instruction which is part of the RM-USR user subroutine option firmware. The LCMRP instruction loads either 8080 microprocessor instructions or RM-9000 display instructions into RAM memory on the RM-MOC memory expansion card. The LCMRP does not differentiate between 8080 instructions and RM-9000 display instruction the use of either the CCM or the XIM instruction defines the functional use of any data loaded by the LCMRP instruction.

PERTINENT ARGUMENTS

RAM ADDRESS	Specifies the address in internal 8080 RAM at which subroutine or display instruction data will be stored. No address validity check is made that the address specified is legal.
BYTE COUNT	Defines the number of bytes of subroutine or display instruction data to be stored in internal 8080 RAM.

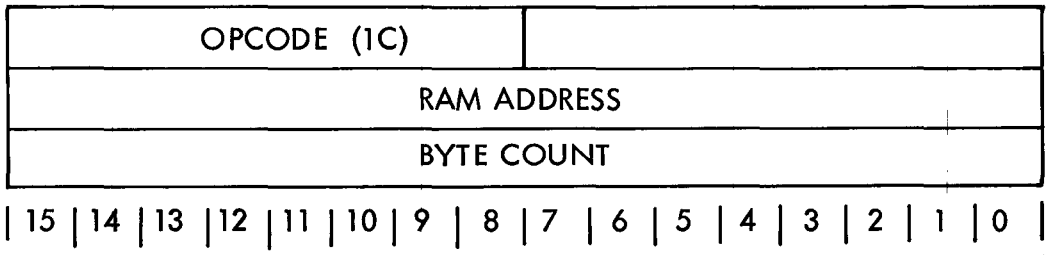
DATA FORMAT

The subroutine or display instruction data is stored into internal 8080 RAM low-order byte first, followed by the high-order byte. If the BYTE COUNT is odd, the high-order byte of the last data word will be discarded.

POSSIBLE ERRORS

If an area of RAM used by an option which exists in a given system is overwritten, the operation of that option cannot be guaranteed unless a reset sequence is initiated.

3-71 Read Control Memory Instruction (RCM)



The RCM instruction is a special-format instruction which is part of the RM-USR subroutine option firmware. The RCM instruction transfers up to 2000_{16} (8192_{10}) bytes of data from RAM on the RM-MOC memory expansion card to the host processor. This is the region within the internal 8080 microprocessor's address space which may be loaded via the LCM or LCMRP instruction. It is not valid to attempt to read back data from the region outside of the RM-MOC card RAM, and as a consequence, the system may be hung up in doing so.

PERTINENT ARGUMENTS

RAM ADDRESS — Specifies the address in internal 8080 RAM in the range 4000_{16} through $5FFF_{16}$. Addressing outside of this region is illegal and may cause the RM-9000 display system to hang up.

BYTE COUNT — Defines the number of bytes of data to be read back from RAM on the RM-MOC. Since readback to the host processor is done on a word basis, if the BYTE COUNT is odd, then one additional byte will actually be read back to the host processor. Since it is illegal to readback from outside of the RM-MOC RAM area, the sum of BYTE COUNT and RAM ADDRESS must not be greater than 6000_{16}.

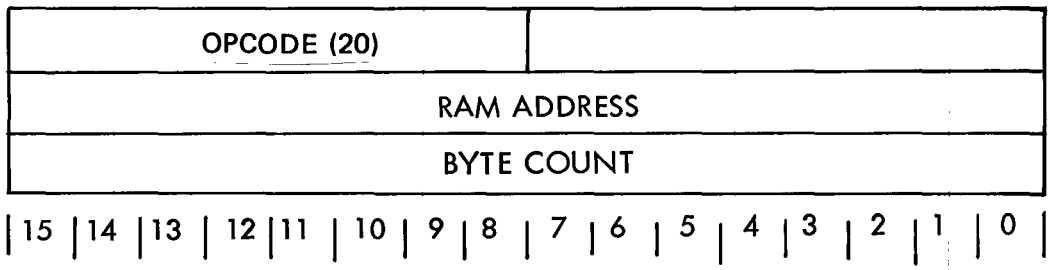
DATA FORMAT

The data which is read back to the host processor is stored in each data word as follows: The first data byte is stored in the high-order byte of the first readback word, the second data byte in the low-order byte of the first data word, the third data byte in the high-order byte of the second data byte, etc.

POSSIBLE ERRORS

If read back via the RCM is attempted outside of the address region 4000_{16} through $5FFF_{16}$, the system may hang up, and communication can only be reestablished after a hard system reset.

3-72 Read Control Memory With Reverse Packing Instruction (RCMRP)



The RCMRP instruction is a special-format instruction which is part of the RM-USR user subroutine option firmware. The RCMRP instruction transfers up to 2000 (8192_{10}) bytes of data from RAM on the RM-MOC memory expansion card to the host processor. This is the region within the internal 8080 microprocessor's address space which may be loaded via the LCM or LCMRP instructions. It is not valid to attempt to readback data from the region outside of the RM-MOC card RAM, and, as a consequence, the system may be hung up in doing so.

PERTINENT ARGUMENTS

RAM ADDRESS — Specifies the address in internal 8080 RAM in the range 4000_{16} through $5FFF_{16}$. Addressing outside of this region is illegal and may cause the RM-9000 display system to hang up.

BYTE COUNT — Defines the number of bytes of data to be read back from RAM on the RM-MOC. Since readback to the host processor is done on a word basis, if the BYTE COUNT is odd, then one additional byte will actually be read back to the host processor. Since it is illegal to read back from outside of the RM-MOC RAM area, the sum of BYTE COUNT and RAM ADDRESS must not be greater than 6000_{16}.

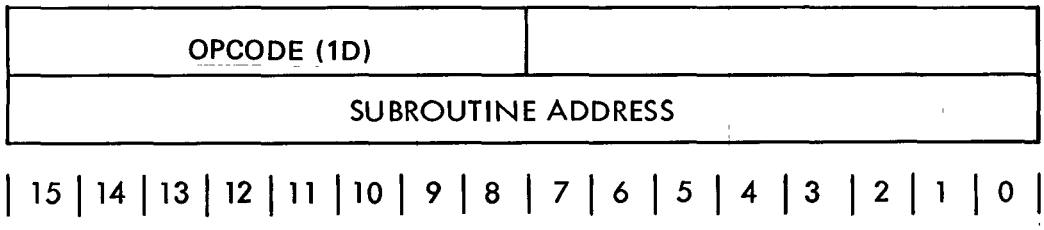
DATA FORMAT

The data which is read back to the host processor is stored in each data word as follows: The first data byte is stored in the low-order byte of the first readback word, the second data byte in the high-order byte of the first data word, the third data byte in the low-order byte of the second data byte, etc.

POSSIBLE ERRORS

If readback via the RCMRP is attempted outside of the address region 4000_{16} through $5FFF_{16}$, the system may hang up, and communication can only be reestablished after a hard system reset.

3-73 CALL CONTROL MEMORY INSTRUCTION (CCM)



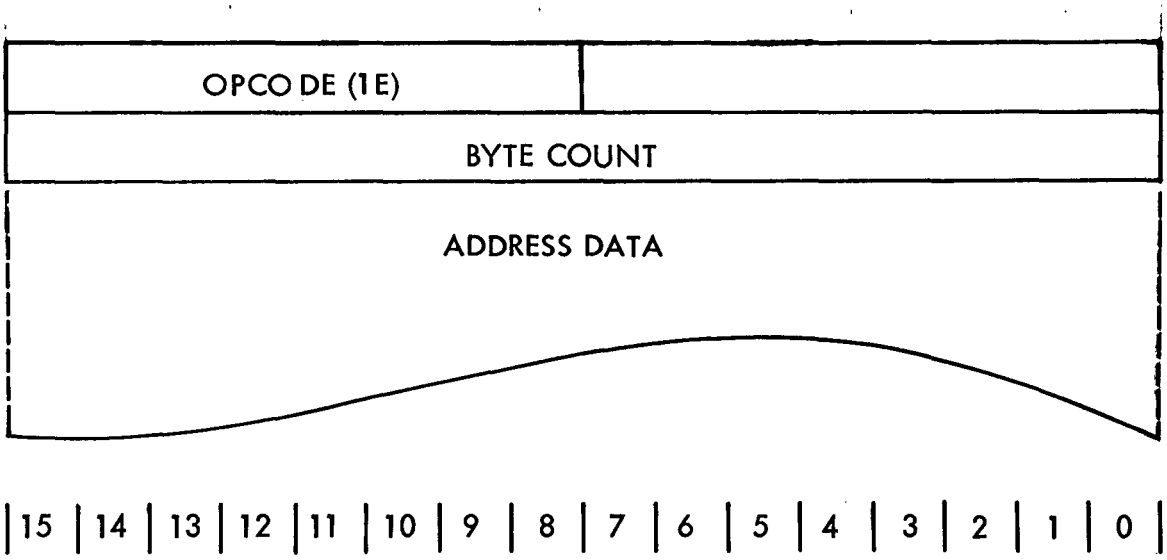
The CCM instruction is a special-format instruction which is part of the RM-USR user subroutine option package. The CCM instruction is used to transfer control from the RAMTEK-supplied firmware package to a user-written subroutine which is stored in the RAM on the RM-MOC memory expansion card via the LCM or LCMRP instructions. The user-subroutine must be executable 8080 microprocessor code with addressing compatible with the region into which the subroutine has been loaded. The user subroutine to be executed must return control to the RAMTEK-supplied firmware via the 8080 RET instruction. Since the user subroutine executes within the RM-9000 display system, it has complete access to all internal display and interfacing registers. As a result of this, it is possible to perform exotic display functions specific to a given user application. It is also possible to initiate actions which cause the system to hang up and which can only be resolved via a hard system reset. Therefore, great care should be taken in writing user subroutines, making careful reference to Volume I of the RM-9000 Theory of Operation Manual.

It is the responsibility of the user subroutine to maintain the integrity of the system stack pointer. When entry is made to the user subroutine, 127_{10} stack locations (words) are available for use. For each stack 'push' operation, there must be a corresponding stack 'pop' within the subroutine to maintain stack integrity. If this stack is not properly maintained, execution of the RET transferring control to the RAMTEK-supplied firmware system will fail and no further operations will be possible until a hard system reset is issued. All 8080 registers are available for use by the subroutine.

PERTINENT ARGUMENTS

SUBROUTINE ADDRESS Defines the internal 8080 microprocessor RAM address at which subroutine execution will begin.

3-74 Execute Instruction Memory Instruction (XIM)



The XIM instruction is a special-format instruction which is part of the RM-USR user subroutine firmware package. The XIM instruction executes one or more RM-9000 instructions from RAM memory on the RM-MOC memory expansion card. The 8080 address of each instruction to be executed via the XIM instruction is stored within the XIM instruction itself. It is possible to nest XIM instructions by pointing to an address of another XIM instruction. Up to 38_{10} such nesting operations are legal. More than 38_{10} levels of nesting will result in stack overflow and may result in the system hanging. The standard system stack is used to save the nesting information. No stack overflow check is performed. Also, there is no check made for recursion, i.e., an XIM instruction referencing the XIM which referenced it previously. This will result in a recursive loop which will always lead to a stack overflow condition.

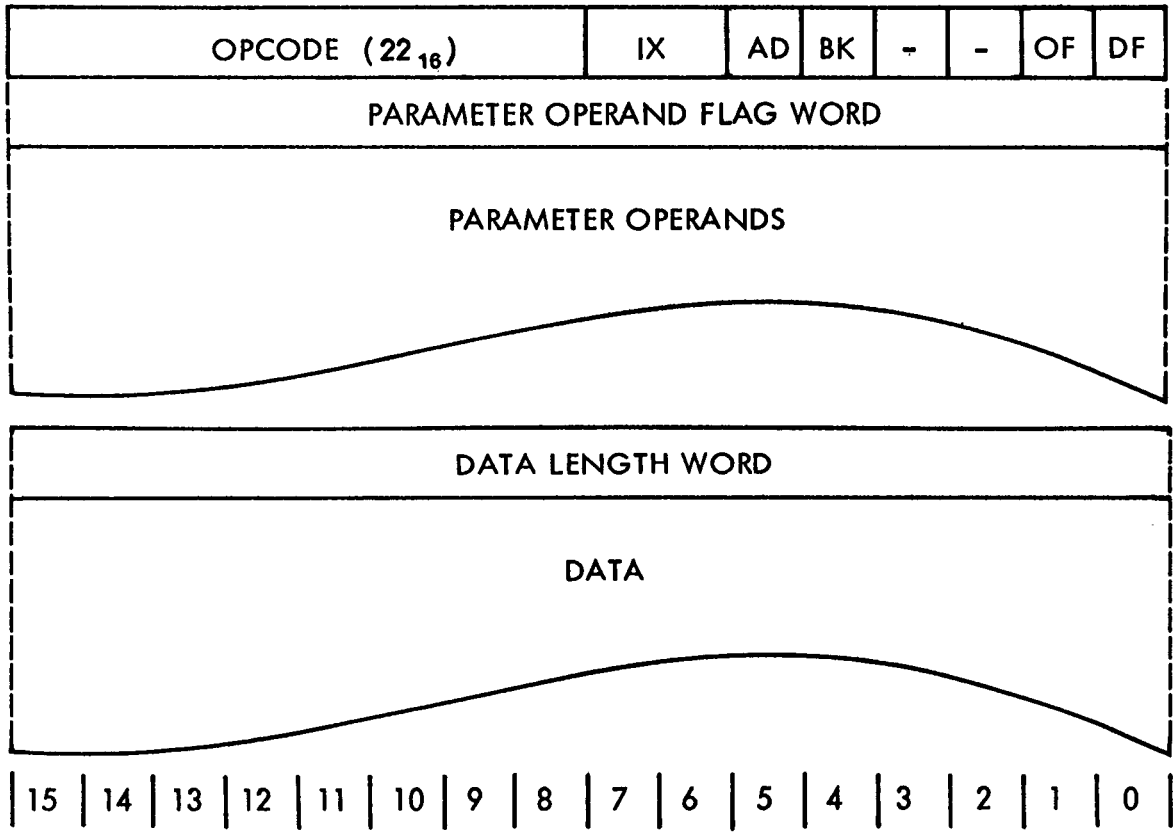
PERTINENT ARGUMENTS

BYTE COUNT — Specifies the number of bytes of address data which are present in the XIM instruction. Since the instruction addresses within the XIM instruction are 16-bit values, the BYTE COUNT must be an even value. If BYTE COUNT is odd, the value used will be one less than the stored value of BYTE COUNT.

DATA FORMAT

Each 16-bit word following the BYTE COUNT word defines the starting address of an RM-9000 display instruction of length n, where $1 \leq n \leq 32K$. These addresses are the absolute internal 8080 RAM addresses at which data was loaded via the LCM or LCMRP instructions.

3-75 Magnify Instruction (MAGNFY)



The MAGNFY instruction is a normal-format instruction which is implemented in the RM-MAG windowed magnification firmware option. The MAGNFY instruction performs a magnification by a factor of two in both the 'X' and the 'Y' (element and line) dimension with the rectangular region defined by the WINDOW parameter operand. This magnification algorithm is a simple pixel replication, such that each pixel in the window before magnification generates four pixels in the magnified image.

The START-POINT parameter operand defines the point which will become the new center of the rectangular area defines by WINDOW. The point defined by START-POINT must be defined within the WINDOW region or indeterminate actions will result. The START-POINT parameter operand effectively defines a rectangular sub-area which is exactly half the dimensions of the rectangular region defined by WINDOW. This is the region which will be magnified by a factor of two. However, if the perpendicular distance of this sub-area's center is within one-fourth of the WINDOW width in the appropriate dimension from the WINDOW margins, the sub-area will exceed the WINDOW region. Data will

not be taken from this region outside of the WINDOW region; instead, the BACKGROUND or FOREGROUND value as modified by the BK flag will be written into the final magnified region. If, however, the AD bit is set to one (1), the FOREGROUND or BACKGROUND value will not be substituted for the data outside of the current window. In effect, the region defined by the START-POINT parameter operand and the WINDOW boundary will be magnified about the center of the WINDOW, and any data which is not overwritten using data from inside the WINDOW will remain unchanged.

PERTINENT CONTROL BITS

IX	Defines the address mode in which INDEX1, INDEX2, WINDOW, and START-POINT parameter operands (i.e., those operands directly affecting the absolute position of the MAGNIFY window or centerpoint) are evaluated (see Section 3-11).
BK	Defines the color or intensity to be used for data which does not exist within the sub-area to be magnified (i.e., when the center point is sufficiently close to the WINDOW boundaries). IF BK = 0, the BACKGROUND value will be used; if BK = 1, the FOREGROUND value will be used.
AD	If AD = 1, specifies that no data will be written into WINDOW region areas where the FOREGROUND or BACKGROUND values would be written into the subregion for which no data existed due to the START-POINT being within one-fourth of the WINDOW dimension from any WINDOW boundary.

PERTINENT PARAMETER OPERANDS

SUBCHANNEL	Specifies the subchannels to be affected by the magnification process.
FOREGROUND	Specifies the data value to be used when BK = 1 and insufficient data is available for magnification.
BACKGROUND	Specifies the data value to be used when BK = 0 and insufficient data is available for magnification.
INDEX1	Displaces all WINDOW and START-POINT operands (when set in the same MAGNIFY instruction) when IX = 01.

INDEX2	Displaces all WINDOW and START-POINT operands (when set in the same MAGNIFY instruction) when IX = 02.
WINDOW	Defines the rectangular region which contains the sub-area to be magnified and which will contain the magnified result.
START-POINT	Defines the center of the sub-area to be magnified. The parameter operand must lie within the region defined by WINDOW.

DATA

Any data present in this normal-format instruction (i.e., when DR = 1 and the DATA LENGTH WORD value is non-zero) will be discarded.

Ramtek

SECTION IV

INTERFACE PROGRAMMING

4-0 INTERRUPT OPERATIONS

The RM-9000 is capable of generating interrupts to the host processor through the RAMTEK supplied interface associated with each display system. The four (4) possible interrupt types are:

- Illegal Instruction Interrupt
- Cursor Interrupt
- Receiver Interrupt
- Transmitter Interrupt

These interrupts are generated internally using the interface interrupt register (See Section 6 of the RM-9000 Theory of Operation Manual - Volume I).

All of the RAMTEK supplied interfaces generate an interrupt to the host processor based on a low-to-high transition of the interrupt lines from the RM-9000 control board to the computer interface. The standard firmware generates this transition by loading a zero in the interface interrupt register followed by the setting of the specific interrupt line(s) in this register. Reading of the computer interface status word clears the interrupt flags in this word.

4-1 ILLEGAL INSTRUCTION INTERRUPT

The illegal instruction interrupt is generated when a word of information from the interface to be interpreted as an instruction opcode has an opcode which does not exist for the specific display system configuration. The illegal instruction interrupt is then generated. The next word will then be interpreted as an opcode word.

4-2 CURSOR INTERRUPT

The cursor interrupt is generated when the RM-PER interactive peripherals option is installed and one of the following occurs: either the ENTER momentary action switch is pressed or the cursor position is changed with the TRACK switch in the ON position. These keys (ENTER and TRACK) are present on both the joystick and the trackball.

4-3 RECEIVER INTERRUPT

The receiver interrupt is generated when the RM-PER interactive peripherals option is installed and a key on the keyboard is depressed. The 8-bit code asso-

ciated with the depressed key is stored in the FIFO (first-in-first-out) buffer associated with that particular keyboard and may be read by the host processor using the READ KEYBOARD instruction.

If any characters are present in the keyboard buffers after execution of a READ KEYBOARD instruction, a receiver interrupt will be issued.

4-4 Transmitter Interrupt

The transmitter interrupt is generated when the RM-PER interactive peripherals option is installed and the completion of a serial output transmission using the WRITE KEYBOARD instruction is indicated to the firmware by the RM-SLC serial link card. This interrupt indicates that another WRITE KEYBOARD instruction to the interrupting output device is valid.

SECTION V

RM-9000 INSTRUCTION TIMING

5-0 RM-9000 INSTRUCTION TIMING

This section describes the execution times of the RM-9000 display instruction set. In several cases, the instruction times are approximate due to the variable nature of the parameters used as input to the instruction. In these cases (which are noted in the instruction's execution time description, the equation describing the execution speed becomes extremely complex and cumbersome, in which case a reasonable approximation to the equation has been made.

5-1 Standard Instruction Set Execution Times

The following sections describe the execution timing for the RM-9000 standard instruction set. For normal format instructions which are, by nature, of variable format, it is possible to set a variable number (up to 16) parameter operands as well as actually execute the display instruction itself. Section 5-2 defines the parameter operand processing time (POPT) for the normal format instructions.

5-2 Parameter Operand Processing Time

In any normal format instruction, it is possible to define the value of up to 16 parameter operands. The parameter operand processing time or POPT must be added to the execution time of the normal format instruction in which these parameter operands are set.

5-3 INOP Instruction Time

Execution Time = 135 µs + 65 µs * NP + 40 µs * NPW

+ 65 µs * NBD

where NP = number of parameter operands in NOP instruction to be ignored.

NPW = number of words of parameter operand values to be ignored in INOP instruction

NBD = number of bytes of data in INOP instruction to be ignored

NOTE

EVEN THOUGH INOP IS A NORMAL-FORMAT INSTRUCTION, ITS TIMING REPRESENTS A SPECIAL CASE, SINCE ALL PARAMETER OPERANDS AND DATA ARE DISCARDED.

Table 5-1 Normal-Format Parameter Operand Processing Times

i	PARAMETER OPERAND	PT_i
0	SUBCHANNEL	107 µs
1	FOREGROUND	107 µs
2	BACKGROUND	107 µs
3	INDEX 1	205 µs
4	INDEX 2	205 µs
5	ORIGIN	205 µs
6	WINDOW	1315 µs
7	SCAN	970 µs
8	DIMENSION	140 µs
9	SPACING	146 µs
10	SCALE	105 µs
11	FUNCTION	105 µs
12	CONIC-EQUATION	695 µs
13	BASE-LINE	105 µs
14	SCROLL-COUNT	105 µs
15	START-POINT	315 µs

$$POPT \cong \sum_{i=0}^{15} (PT_i)$$

where $PT_i = 0$ if the i^{th} parameter operand is not set in the instruction
otherwise the value of PT_i is the value from Table 5-1.

5-4 Set Instruction Timing

Execution Time = 125 µs + POPT + DFT +[57.5 µs * NBD]

where DFT = Ø if the data flag DF = 0

= 36 µs if the data flag DF = 1

NBD = number of bytes of data in the set instruction to be ignored

5-5 Erase Instruction Timing

Execution Time = 265 µs + POPT +[NL * [43.5 µs +[1.5 µs * NE]]

where NL = number of lines in WINDOW to be erased

NE = number of elements per line in WINDOW to be erased

For example, a full screen with no parameter operand processing in an RM-9200 system would be:

265 µs +[256 * [43.5 µs +[1.5 µs + 640]] = 257 ms

5-6 Write Image Instruction Timing

Execution Time = 490 µs + POPT

+ [[WW * DTR] + 113 µs][NL-1] + [RP * DTR]

where WW = window width in pixels (i.e., width parallel to the primary update direction)

DTR = interface data transfer rate in microseconds (minimum value of 1.5 µs due to refresh memory access rate)

NL = number of lines in image

RP = number of pixels in last line of image

For example, a full-screen image transfer on an RM-9100 system interfaced to a DR11B DMA interface (and assuming an infinitely fast image data source in the PDP-11 computer) and assuming no parameter operand processing, would be:

490 µs + [[320 * 1.5 µs] + 113 µs] * 255 + [320 * 1.5 µs] = 152 ms
using values of WW=320, DTR=1.5 µs, NL=256, RP=320

5-7 Read Image Instruction Timing

The execution time for the RI instruction is the same as that of the WI instruction (See Section 5-6).

5-8 Write Text Instruction Timing

Execution time = 480 µs + POPT
+ [263 µs + [64 µs*CH] + [1.5 µs*CH*CW]] *NPC
+ [162 µs *NCR] + [146 µs *NLF]

where CW - character width (defined by X DIMENSION)
CH = character height (defined by Y DIMENSION)
NCR = number of carriage returns or primary window boundary intersections
NLF = number of line feeds or secondary window boundary intersections
NPC = number of non-CR or non-LF characters

For example, a full-screen text write on an RM-9200 system (i.e., 91 by 28 characters of text) with no parameter processing, would be:

480 µs + [263 µs + [64 µs*9] + [1.5 µs *7*9]] *2548
+ [162 µs * 28] + [146 µs *1] = 2.38 seconds

5-9 Load Hard Register Instruction Timing

Execution time = 170 µs

5-10 Read Soft Register Instruction Timing

Execution Time = 150 µs

5-11 Load Auxiliary Memory Instruction Timing

Execution Time = 275 μs + [NW * DTR]
where NW = Number of words of auxiliary memory data to be transferred
DTR = Interface data transfer rate in microseconds (minimum value of 0.765 μs based on the Type II and V video lookup tables loading rate)

For example, to load a Type II video lookup table would take:

275 μs + [1024 * 0.765 μs] = 1.05 ms

5-12 Read Auxiliary Memory Instruction Timing

The execution time for the RAM instruction is the same as that of the LAM instruction (See Section 5-11).

5-13 Reset Instruction Timing

Execution Time = 29 ms + ERST + OPTT

where ERST = screen erase time (dependent on system resolution) when DIP SWITCH 3 = 0

STDT = 134 ms for RM-9100
= 257 ms for RM-9200
= 514 ms for RM-9300

OPTT = sum of power-on routine execution times for the options which exist in a particular configuration

$$= 63 \text{ ms} + \sum_{i=o}^{n} POT_i$$

where POT_i equals the power-on option time for the i^{th} option which exists and n is the number of existing options

RM-GRA	=	45 μs
RM-CON	=	0 μs
RM-SCR	=	0 μs
RM-SCA	=	0 μs
RM-STA	=	50 μs
RM-FNT	=	1493 μs
RM-PER	=	957 μs
RM-USR	=	0 μs

5-14 Initialize Instruction Timing

Execution Time = 4.97 ms

5-15 Options Instruction Set Execution Times

The following sections describe the execution timing for the set of RM-9000 options instructions. For normal-format options instructions, reference should be made to Section 5-2 for the parameter operand processing time POPT.

5-16 Write Vector Instruction Timing

Execution Time = 220 μs + POPT

$$+ \sum_{i=1}^{NV} \left[VST_i + \sum_{j=1}^{NPV_i} VPT_{ij} \right]$$

$$\cong 220\ \mu s + POPT + \sum_{i=1}^{NV} \left[480\ \mu s + \left[NPV_i * 36.3\ \mu s \right] \right]$$

where NV = Number of linked vectors to be generated in a given WV instruction

VST_i = Vector set-up time for the i^{th} vector (based on the slope of the vector) 480 μs

NPV_i = Number of pixels in the i^{th} vector

VPT_{ij} = Pixel calculation/write time for the j^{th} pixel of the i^{th} vector (dependent on the i^{th} vector's slope and quadrant); minimum value ~ 34.6 µs, maximum value ~ 37.9 µs, average value ~36.3 µs.

5-17 Write Plot Instruction Timings

Execution Time = 543 µs + POPT

$$+ \sum_{i=1}^{NP} \left[223\ \mu s + \left[103\ \mu s + \left[1.5\ \mu s * EDIM\right]\right] * EH_i\right]$$

where NP = Number of plot entities to be generated in a given WP instruction

EDIM = Width in pixels of plot entities parallel to the plot axis (defined either by the x-DIMENSION or y-DIMENSION parameter operand depending on the value of SCAN). See Section 3-49.

EH_i = Size in pixels of the i^{th} plot entity perpendicular to the plot axis.

5-18 Write Raster Instruction Timing

Execution Time = 420 µs + POPT

$$+ \left[NRB * 130\ \mu s\right] + \left[NRL * 62\ \mu s\right]$$

where NRB = Number of bytes of raster data.

NRL = Number of lines of raster data within the region specified by WINDOW (i.e., number of lines of raster data which are parallel to the primary scan direction).

5-19 Write Conic Instruction Timing

Execution Time = 887 µs + POPT + $\left[NCP * CPR\right]$

where NCP = Number of pixels to be generated per conic.

CPR = Conic pixel generation rate; minimum value ~ 250 μs, maximum value ~ 1180 μs, typical value ~ 300 μs.

NOTE

THE CONIC PIXEL GENERATION RATE IS EXTREMELY VARIABLE, BEING DEPENDENT ON SEVERAL NON-LINEAR PARAMETERS, AND THEREFORE THE VALUE OF CPR IS DIFFERENT FOR EACH PIXEL.

5-20 Scroll X Instruction Timing

Execution Time = 470 μs + POPT

$$+ \left[238\ \mu s * WW\right] + \left[3\ \mu s * WW * WH\right] - \left[122\ \mu s * COUNT\right]$$

$$- \left[1.5\ \mu s * COUNT * WH\right]$$

where WW = Width in elements of the rectangular region defined by WINDOW.

WH = Height in lines of the rectangular region defined by WINDOW.

COUNT = Absolute value of the SCROLL-COUNT parameter operand

5-21 Scroll Y Instruction Timing

Execution Time = 456 μs + POPT

$$+ \left[238\ \mu s * WH\right] + \left[3\ \mu s * WW * WH\right] - \left[122\ \mu s * COUNT\right]$$

$$- \left[1.5\ \mu s * COUNT * WW\right]$$

where WW = Width in elements of the rectangular region defined by WINDOW

WH = Height in lines of the rectangular region defined by WINDOW

COUNT = Absolute value of the SCROLL-COUNT parameter operand

5-22 Save Environment Instruction Timing

Execution Time = 627 μs

5-23 Restore Environment Instruction Timing

Execution Time = 5.05 μs

5-24 Load Programmable Font Instruction Timing

Execution Time = 385 μs

5-25 Load Programmable Font Reverse Packing Instruction Timing

Execution Time = 385 μs

5-26 Write Cursor Instruction Timing

Execution Time = 331 μs for RM-9100
= 299 μs for RM-9200
= 271 μs for RM-9300

5-27 Read Cursor Instruction Timing

Execution Time = 268 μs for RM-9100
= 242 μs for RM-9200
= 219 μs for RM-9300

5-28 Write Keyboard Instruction Timing

Execution Time = 246 μs

5-29 Read Keyboard Instruction Timing

Execution Time = 480 μs (Average)

5-30 Sense Peripheral Status Instruction Timing

Execution Timing = 163 μs

5-31 Load Control Memory Instruction Timing

Execution Time = 190 μs + 64 μs * NB

where NB = Number of bytes of data to be loaded into RAM

5-32 Load Control Memory Reverse Packing Instruction Timing

Execution Time = 195 μs + 67 μs * NB

where NB = Number of bytes of data to be loaded into RAM

5-33 Read Control Memory Instruction Timing

Execution Time = 212 μs + 12.5 μs * NB

where NB = Number of bytes of data to be read from RAM

5-34 Read Control Memory Reverse Packing Instruction Timing

Execution Time = 264 μs + 0.75 μs * NB

where NB = Number of bytes of data to be read from RAM

5-35 Call Control Memory Instruction Timing

Due to the variable nature of the user-written subroutine, the timing cannot be defined in this document.

5-36 Execute Instruction Memory Instruction Timing

Due to the variable number of instructions that can be executed by the XIM instruction, the execution time is variable and approximately equal to the sum of execution times for each display instruction.

Framtek

APPENDIX A

TYPE II VIDEO LOOK-UP TABLE

FUNCTIONAL DESCRIPTION

A-0 INTRODUCTION

The RM-9000 Type II Video Board contains a 1024_{10} by 12_{10} bit video look-up table which is loadable by the host processor and which defines the functional correspondence between all possible data values stored in refresh memory and the actual color or grey scale intensity generated for each pixel. Due to chassis size, the RM-9100 and RM-9200 systems may contain up to two (2) Type II Video Boards; the RM-9300 system may contain up to four (4) of these modules.

A-1 TYPE II VIDEO LOOK-UP TABLE

The video look-up table (or VLT) on a Type II video board is comprised of 1024_{10} addressable locations; each location in the VLT is 12 bits in length. The VLT is loaded from the host processor using the special-format LOAD AUXILIARY MEMORY instruction (See Section 3-42). Readback from the VLT to the host processor is implemented via the special-format READ AUXILIARY MEMORY instruction. In standard configurations, the first Type II VLT is addressed as Device 0 in the LAM and RAM, the second VLT as Device 1. It is possible for a user to specify the addressing configurations for access to all VLT's in the system.

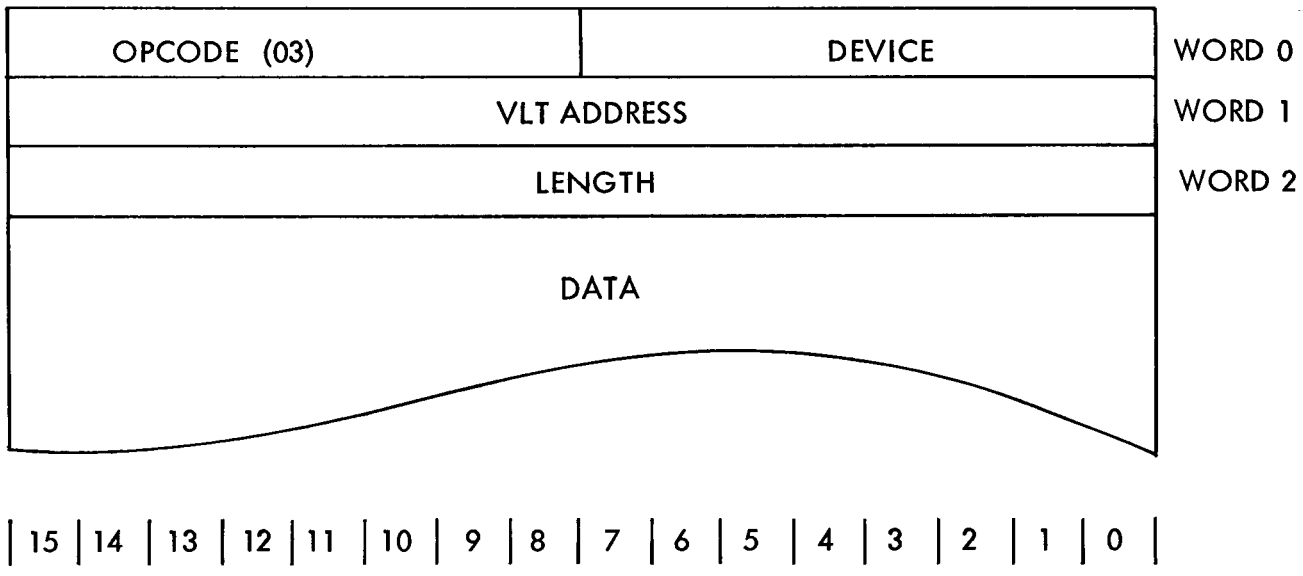


Figure A-1 Lam Instruction Format

The low-order byte of the word 0 of the LAM instruction defines the VLT to be loaded. Word 1 defines the 10_{10} - bit address at which VLT loading will begin. Bits 10-15 in the VLT ADDRESS word are ignored; only Bits 0-9 are effective. Internal addressing of the VLT is designed so that addresses greater than 1023_{10} wrap back to Address 0. The LENGTH word defines the number of bytes of VLT data to be loaded from the hostprocessor. Since loading is done on a full-word basis, the LENGTH value should always be even. The actual DATA to be loaded into the VLT RAM follows the LENGTH word. The low-order 12_{10} bits of each 16_{10}-bit word from the host processor are loaded into successive locations of the VLT. The functional use of the 12 bits of data loaded into the VLT is dependent on the Type II Video board configuration (discussed in Section A-2).

The RAM instruction is used to readback VLT data values from the Type II Video board. The definition of the DEVICE, VLT ADDRESS and LENGTH values is the same as those of the LAM instruction. The data from the VLT is stored in Bits 0-11 of the readback data word. Bits 12-15 of the readback data word are indeterminate.

Note that if a Type II VLT, which does not exist in an RM-9000, is accessed via the LAM or the RAM instructions, the RM-9000 system will hang and can only be cleared by a hardware reset.

A-2 TYPE II VIDEO CONFIGURATIONS

The function of the Type II Video board is to generate video signals to the CRT monitor based on the refresh memory data value for each pixel and the VLT RAM value stored for the corresponding location using the pixel value as an address into the VLT. For example, if pixel (0,0) contained a value of two (2) in refresh memory, the data stored in location two (2) of the VLT would define the color or grey scale intensity actually displayed for that pixel. All pixels with a refresh memory data value of two (2) would be displayed with the same color or grey scale intensity.

There are four (4) video outputs from each Type II Video board: three (3) outputs from three (3) 4-bit DAC's (digital-to-analog converters) and one (1) video output from an 8-bit DAC. The actual BNC connections for each of these four (4) video outputs are video slot-dependent and are available in Section 2 of the RM-9000 Theory of Operation Manual, Volume I.

The three (3) 4-bit DAC outputs designated as RED, GREEN and BLUE outputs correspond to the VLT RAM values defined by Bits 8-11, Bits 4-7 and Bits 0-3, respectively. The 8-bit DAC video output can be configured such that it repre-

sents either the low-order 8 bits of output from VLT or the binary value of Subchannel 0 thru 7 from refresh memory directly.

A-3 TYPE II VLT PARTITIONING

It is possible to configure any Type II VLT such that less than 10 subchannels from refresh memory are used to generate the video outputs. Since the full 1024_{10} x 12_{10} bit RAM is always present, the Type II VLT is "partitioned" or divided up into 2^m sections of equal length where m is the number of high-order subchannels not used as address input to the VLT for video generation. Thus for example, if subchannels 8 and 9 were not used for input to the VLT, then four (4) partitions of 256_{10} locations each would be available.

All partitions are identical to each other, varying only in the address by which they are loaded and read from the host processor. In the example where subchannels 8 and 9 were not used, four (4) partitions of 256_{10} locations each are available. They are addressed as VLT locations 0 through 255_{10}, 256_{10} through 511_{10}, 512_{10} through 767_{10} and 768_{10} through 1023_{10}. Only one (1) of these partitions is used to generate the actual video outputs at any given time. The last VLT location addressed defines the partition to be used. When using the LAM or RAM instructions, the last VLT location addressed is one greater than the last VLT location actually written with data. Therefore, a LAM instruction with a length of 256_{10} words starting at VLT Address 0 would select the second partition on a VLT which did not use Subchannels 8 and 9. Correspondingly, partitioning of the Type II VLT is not possible in a system with all 10 subchannels utilized.

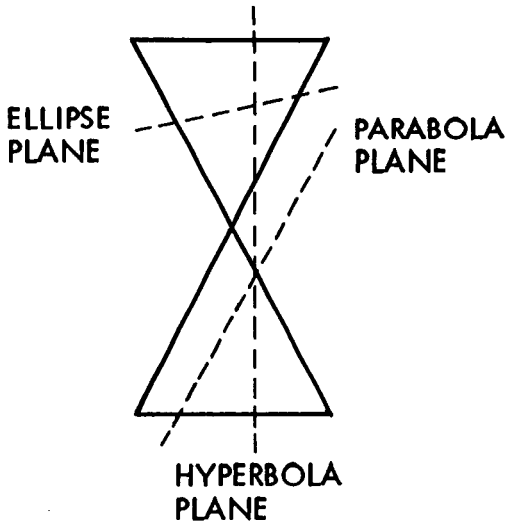
Framtek

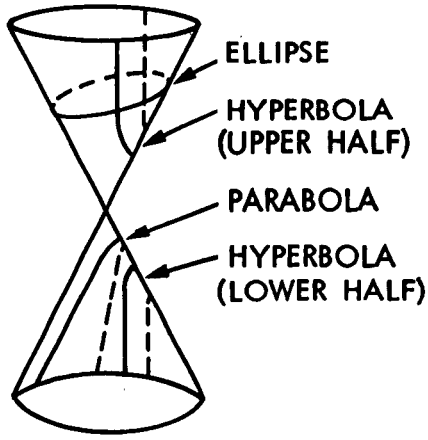
APPENDIX B

GENERATION OF CONICS

B-0 CONICS

A cone is the locus of a line rotated about an intersecting line by a constant angle. Conic sections, or more simply "conics", are defined as the intersection of the cone with some plane. Note that there are no planes in Euclidean space that do not intersect a given cone.





If a given point moves in a plane so that its distance from a fixed point (called the focus), divided by its distance from a fixed line (called the directrix), is a constant e (called the eccentricity), then the curve described by the point is a conic.

There are three basic families of conics, which are differentiated by their eccentricity, as follows:

- $e < 1$ = → ellipse
- $e = 1$ = → parabola
- $e > 1$ = → hyperbola

Cartesian representation of a conic rests on three basic tenets:

1. All points in space can be represented in terms of n variables, where n is the number of dimensions in the Cartesian space being considered. These variables correspond to reference tick marks along orthonormal axes.

2. Space is infinite in extent.

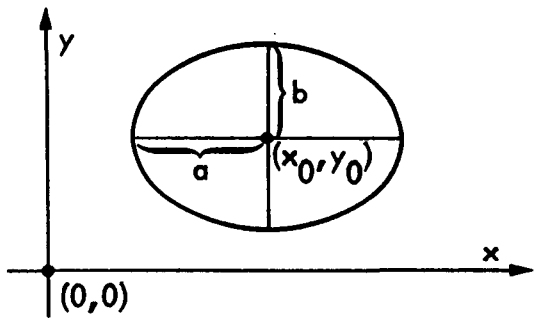
3. Space is infinitely divisible.

Using a Cartesian representation, a single point or set of points (called a locus) can be represented as a mathematical function of its relationship to the reference tick marks along the axes. If we consider two-dimensional Cartesian space, there are two ortho-normal reference axes, usually designated x and y. A specific point would be represented by the functional relationship $x = 3$ and $y = 7.4$. A specific line would be represented by the functional relationship $y = 4x + 3$. In this case, one of the variables becomes dependent on the other. In general, y is considered to be the dependent variable and x the independent variable.

Sets of loci, such as the set of straight lines, can be represented by $y = mx + b$, where m and b are considered constant for any particular member of the set. The constants can be given names that relate to how they affect the members of the set. In this case, m is called the "slope" of the line, and b is called its "y-intercept".

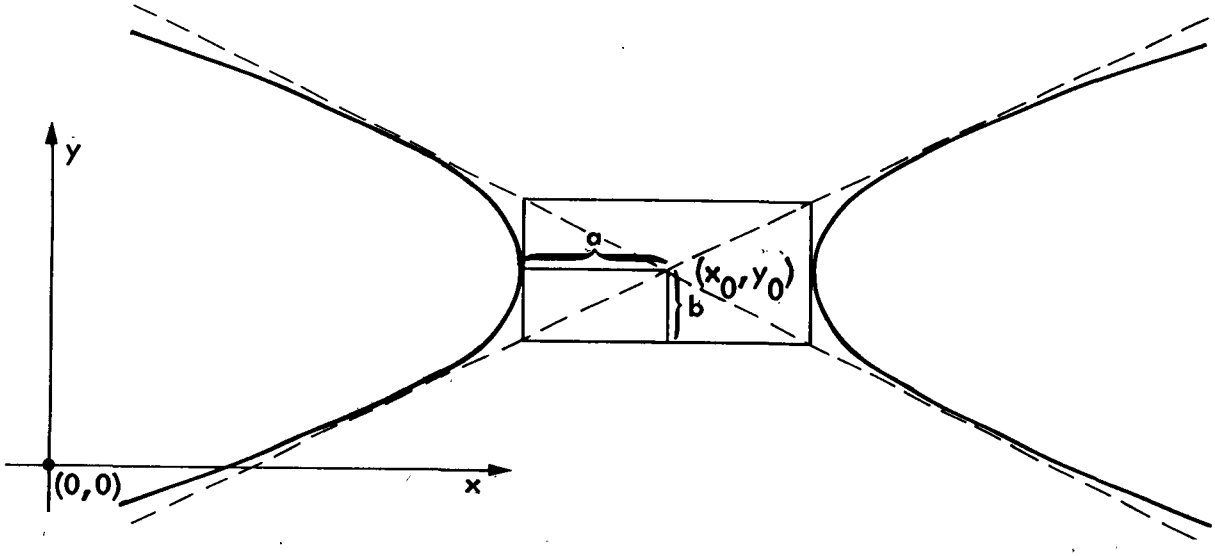
Consider again the families of conics. An ellipse can be specified in terms of its center point and semimajor and semiminor axes, as follows:

$$\left[\frac{(x - x_0)^2}{a^2}\right] + \left[\frac{(y - y_0)^2}{b^2}\right] = 1$$



A hyperbola can be described as:

$$\left[\frac{(x - x_0)^2}{a^2}\right] - \left[\frac{(y - y_0)^2}{b^2}\right] = 1$$



A parabola can be described as:

$$(y-y_0)^2 = 4a(x - x_0)$$

if the parabola opens to right, and

$$(y - y_0)^2 = -4a(x - x_0)$$

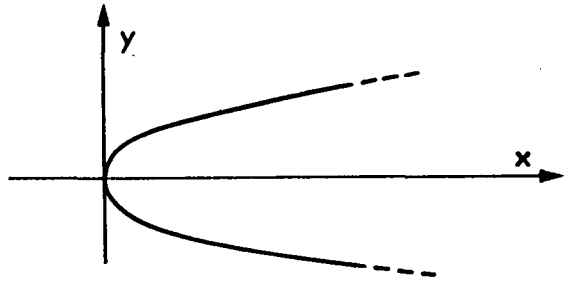
if the parabola opens to left

One of the convenient by-products of using Cartesian coordinates is that all conic sections can be described in their plane of intersection as a second-order polynomial, of the form:

$$\alpha x^2 + \beta y^2 + \gamma xy + \delta y + \varepsilon x + \zeta = 0$$

We can see, upon examination, that if $\alpha = \beta = \gamma = \delta = 0$, we have a straight line parallel to the y-axis. If $\varepsilon = 0$ instead of δ, the line is parallel to the x-axis; if δ and ε are nonzero, the equation is that of some line slanted with respect to the axis.

If α is now made nonzero, everything else going to zero, we have the equation of a parabola reflecting across the x-axis.



If α is made zero and β nonzero, the equation of the parabola reflects across the y-axis. If both α and β are nonzero and of the same sign, the equation will describe an ellipse whose axes are parallel to the coordinate axes. If α and β are both nonzero but of opposite signs, the equation describes a hyperbola whose axes are parallel to the coordinate axes. Making γ nonzero has the effect of slanting the conic section with respect to the axes.

The equation $\alpha x^2 + \beta y^2 + \gamma xy + \delta x + \epsilon y + \zeta = 0$ is the most general form of a second-order polynomial. Six constraints are necessary to describe all possible conic sections. However, the number of variables can be reduced from six to five, if we introduce the additional constraint that the conic section must always pass through the origin of our coordinate system. The form of the equation thus becomes:

$$Ax^2 + By^2 + Cxy + Dx + Ey = 0$$

We have reduced the number of variables from six to five, by introducing a non-variable constraint.

Although the constraint is nonvariable, it is certainly variable-related. The original equation represented all possible conics, as does the new one. It is possible, therefore, to map any conic representation in one scheme, to its representation in the other.

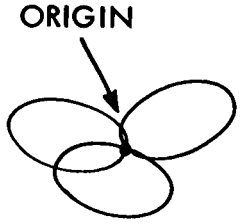
To go from the latter to the former is simple; adding a zero to the left-hand side of the equation puts it into its proper form without changing the conic section. Going from the older representation to the newer presents more of a challenge; however, it can be reduced to a few general transformations based upon whether the conic is a hyperbola, parabola, or ellipse.

B-1 THE ELLIPSE

It was seen earlier that an ellipse could be represented in terms of its semimajor and seminor axes and the offset of its center from the Cartesian origin:

$$\left(\frac{(x - x_0)^2}{a^2}\right) + \left(\frac{(y - y_0)^2}{b^2}\right) = 1$$

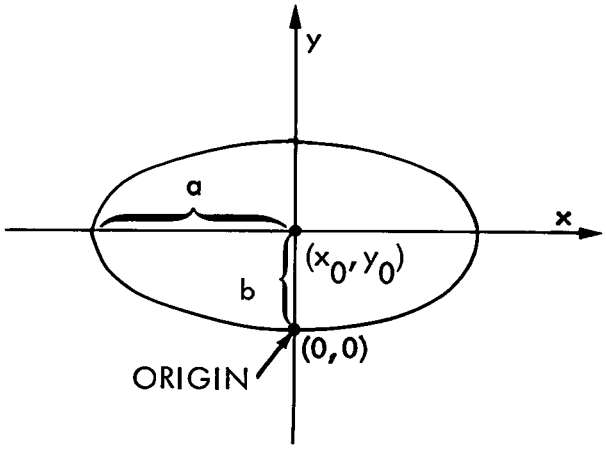
The new constraint that the Cartesian origin must be chosen to lie on the conic locus does not bind us to a single representation. There is still an infinite number of conics of the same eccentricity and orientation that pass through the origin; for example,



One way to think of this is that the origin is being moved along the conic section. The origin can be placed anywhere along the conic and the corresponding equation will fit our parameters. These new parameters can be referred to as Ramtek conic parameters, because this is the form accepted by Ramtek systems.

B-2 DRAWING AN ELLIPSE

For simplicity, let us take as an example the case where the Cartesian origin is placed at the bottom of the ellipse. In this case, $x_0 = 0$ and $y_0 = b$.



a = semimajor axis
b = semiminor axis
(x_0, y_0) = center point

Our equation thus becomes:

$$\frac{(x-0)^2}{a^2} + \frac{(y-b)^2}{b^2} = 1$$

Expanding, we see:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{2yb}{b^2} + \frac{b^2}{b^2} = 1$$

This fits into the equation:

$$Ax^2 + By^2 + Cxy + Dx + Ey = 0$$

where $A = 1/a^2$
$B = 1/b^2$
$C = 0$
$D = 0$
$E = -2/b$

Thus an ellipse with a semimajor axis length of 60 pixels and a semiminor axis of 20 pixels would have the following values:

$A = 1/60^2 \cong 0.000278$
$B = 1/20^2 = 0.0025$
$C = 0$
$D = 0$
$E = -2/20^2 = -0.1$

When the WRITE CONIC command is used with a start and end point specified, it will approximate a conic section by starting at the COP, which the system assumes to be the Cartesian origin, and drawing the conic in a clockwise fashion, until the prespecified end point is reached. The CONIC command takes as arguments the integer coefficients (A, B, C, D, and E) of the Ramtek conic equation and the x and y coordinates of the specified end point. To draw a complete ellipse, we wish to have the conic subroutine draw from the COP, around the ellipse, and back to where we started. Therefore, the end point will be the same as the starting point.

Note that the arguments must all be integers so that, to use the coefficients calculated, we must multiply the equation through by some number large enough to make all the arguments into integers. If the number we choose to multiply by is, say, 25,000, our coefficients turn out to be:

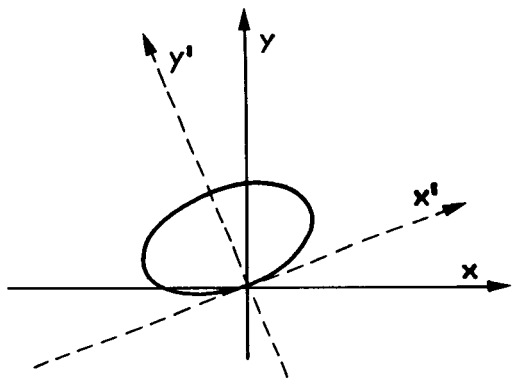
$A = 7$
$B = 63$
$C = 0$
$D = 0$
$E = -2500$

B-3 TRANSLATING THE ELLIPSE

Translating the ellipse proves to be a minor problem. Simply changing the starting point (COP), and the end point will translate the ellipse. Note that changing the relative position of the start point on the ellipse has the effect of translating the ellipse as well.

B-4 ROTATING THE ELLIPSE

Rotating the ellipse requires changing the coefficients of the equation. Rotating the ellipse is equivalent to rotating the coordinate axes and drawing the ellipse again, relative to the rotated axes.



$$x^1 = x\cos\alpha + y\sin\alpha$$
$$y^1 = y\cos\alpha - x\sin\alpha$$

Taking the old equation and putting it in terms of the new coordinates,

$$\frac{x^{1^2}}{a^2} + \frac{y^{1^2}}{b^2} - \frac{2y^1 b}{b^2} = 0$$

In terms of the old coordinates (the screen coordinates), we get

$$\left(\frac{(x\cos\alpha + y\sin\alpha)^2}{a^2}\right) + \left(\frac{(y\cos\alpha - x\sin\alpha)^2}{b^2}\right) - \left(\frac{2(y\cos\alpha - x\sin\alpha)b}{b^2}\right) = 0$$

We can expand this equation to:

$$\frac{x^2\cos^2\alpha}{a^2} + \frac{y^2\sin^2\alpha}{a^2} + \frac{2xy\cos\alpha\sin\alpha}{a^2} + \frac{y^2\cos^2\alpha}{b^2}$$

$$+ \frac{x^2\sin^2\alpha}{2} - \frac{2xy\cos\alpha\sin\alpha}{2} - \frac{2y\cos\alpha}{b} + \frac{2x\sin\alpha}{b} = 0$$

Regrouping in terms of our coefficients, we get

$$\left(\frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}\right)x^2 + \left(\frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}\right)y^2 = 2\cos\alpha\sin\alpha$$

$$\left(\frac{1}{a^2} - \frac{1}{b^2}\right)xy \qquad \left(\frac{2\sin\alpha}{b}\right)x - \left(\frac{2\cos\alpha}{b}\right)y = 0$$

That is, for the rotated ellipse, starting at the "bottom", we get the following coefficients:

$$A = \frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}$$

$$B = \frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}$$

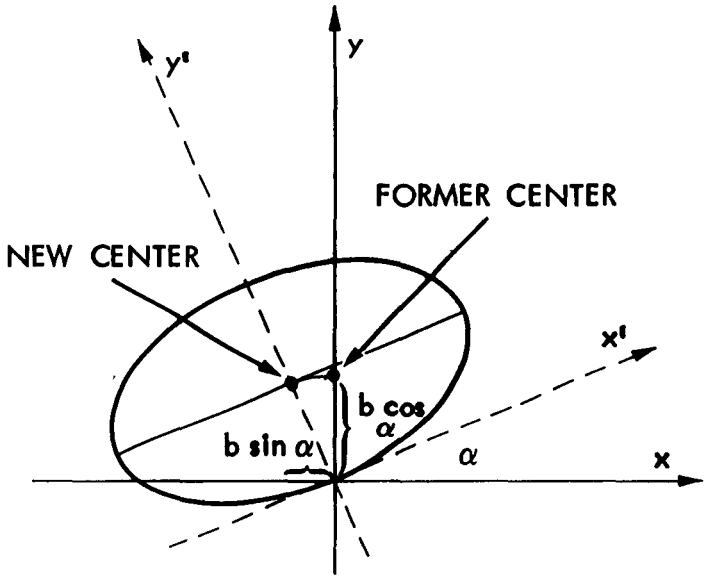
$$C = 2\cos\alpha\sin\alpha\left(\frac{1}{a^2} - \frac{1}{b^2}\right)$$

$$D = \frac{2\sin\alpha}{b}$$

$$E = \frac{-2\cos\alpha}{b}$$

B-5 ROTATING ABOUT THE CENTER OF THE ELLIPSE

Note that when the ellipse was rotated, we rotated it by transforming the axes. What was in effect being done was rotating the ellipse about the Cartesian origin. But remember that the Cartesian origin was a point on the ellipse. In the example we chose, it was at the bottom of the ellipse. If our intention is really to rotate the ellipse about its center, it will be necessary to translate the ellipse as well:

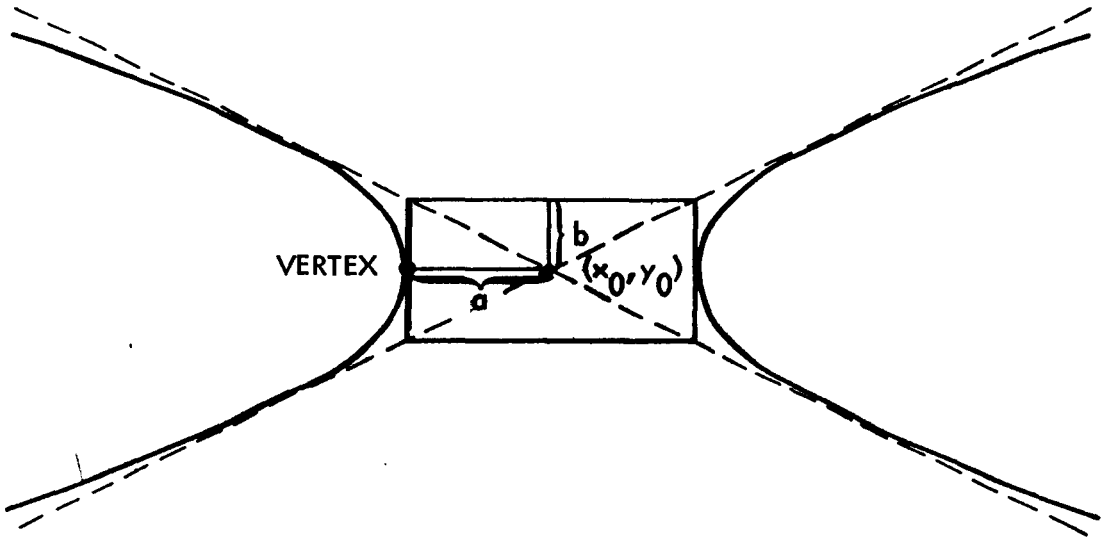


The center has been shifted by the last operation by b sin α in the x-direction and b (1 - cos α) in the y-direction. To rotate about the center, therefore, we must translate by this amount. The starting and ending point must be incremented by b sin α in the x-direction and decremented by b (1 - cos α) in the y-direction.

B-6 THE HYPERBOLA

The same kind of analysis performed on the ellipse can be performed on the hyperbola. Recall the following equation for a hyperbola in terms of its semimajor and semiminor axes:

$$\left(\frac{(x - x_0)^2}{a^2}\right) - \left(\frac{(y - y_0)^2}{b^2}\right) = 1$$



The equation may be expanded to

$$\frac{x^2 + x_0^2 - 2xx_0}{a^2} - \frac{(y^2 + y_0^2 - 2yy_0)}{b^2} = 1$$

To produce the hyperbola, it is still necessary to choose the origin as a point on the curve. For a first attempt, we might choose the origin to be a vertex, since at this point $y_0 = 0$ and $x_0 = a$. This makes it simple to write the equation for the hyperbola in terms of two parameters:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{2x}{a} = 0$$

$A = 1/a^2 = 1/60^2 = 0.000278$

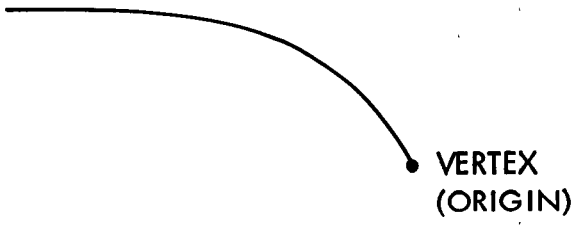
$B = -1/b^2 = -1/30^2 = -0.00111$

$C = 0$

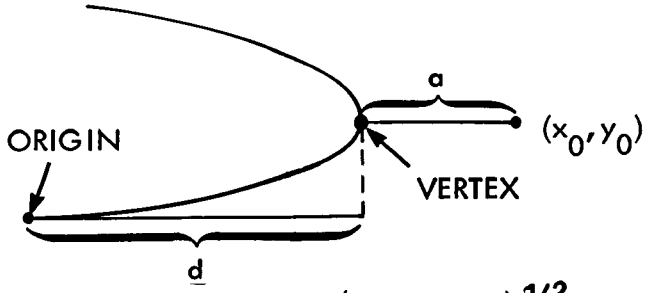
$D = -2/a = -2/60 = -0.0333$

$E = 0$

When we give, as arguments, integers proportional to these values, the following will be drawn:



If we wish to draw from asymptote to asymptote, the origin must be moved to a point along the asymptote.



At this point, $x_0 = a + d$ and $y_0 = b\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2}$

With this information plugged into the equation for the hyperbola, we get

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{2x(a+d)}{a^2} + \frac{2y}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2} = 0$$

which will give us our coefficients in terms of the semimajor and semiminor axes with the additional parameter of the distance along the axis from the origin to the vertex:

$$A = 1/a^2$$

$$B = -1/b^2$$

$$C = 0$$

$$D = \frac{-2(a+d)}{a^2}$$

$$E = \frac{2}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2}$$

B-7 TRANSLATING AND ROTATING THE HYPERBOLA

As for the ellipse, translating the hyperbola is simply a matter of where you choose to start drawing on the screen. Rotation involves transforming the coefficients in the same way as before, i.e., substituting

$$x' = x \cos \alpha + y \sin \alpha$$

$$y' = y \cos \alpha - x \sin \alpha$$

thus,

$$\frac{x'^2}{a^2} - \frac{y'^2}{b^2} - \frac{2x'(a=d)}{a^2} + \frac{2y'}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2} = 0$$

becomes

$$\frac{(x \cos \alpha - y \sin \alpha)^2}{a^2} - \frac{(y \cos \alpha - x \sin \alpha)^2}{b^2}$$

$$- \frac{2(x \cos \alpha - y \sin \alpha)(a+d)}{a^2} + \frac{2(y \cos \alpha - x \sin \alpha)}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2} = 0$$

We can see that this introduces the cross term usually associated with off-axis curves:

$$\frac{x^2 \cos^2 \alpha - 2xy \sin \alpha \cos \alpha + y^2 \sin^2 \alpha}{a^2} + \frac{(-y^2 \cos^2 \alpha - x^2 \sin^2 \alpha + 2xy \cos \alpha \sin \alpha)}{b^2}$$

$$- \frac{2x \cos \alpha (a+d)}{a^2} + \frac{2y \sin \alpha (a+d)}{a^2} + \frac{2y \cos \alpha}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2}$$

$$- \frac{2x \sin \alpha}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2} = 0$$

$$A = \frac{\cos^2\alpha}{a^2} - \frac{\sin^2\alpha}{b^2}$$

$$B = \frac{\sin^2\alpha}{a^2} - \frac{\cos^2\alpha}{b^2}$$

$$C = 2\sin\alpha\cos\alpha\,\frac{1}{b^2} - \frac{1}{a^2}$$

$$D = \frac{2\sin\alpha}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2} - \frac{2(a+d)\cos\alpha}{a^2}$$

$$E = \frac{2\cos\alpha}{b}\left(\frac{d^2}{a^2} + \frac{2d}{a}\right)^{1/2} + \frac{2(a=d)\sin\alpha}{a^2}$$

A similar analysis can be done on the parabola by placing the origin somewhere on the conic section and proceeding as we have before.

B-8 FINAL POINTS TO CONSIDER

Three assumptions were made in using Cartesian representations:

1. All points in space can be represented in terms of n variables, where n is the number of dimensions of the Cartesian space. These variables correspond to the tick marks along the orthonormal axes.

2. Space is infinite in extent.

3. Space is infinitely divisible.

Each of these three assumptions is violated on all computer graphics systems.

The first assumption, that we are dealing with an orthonormal coordinate system, is not accurate. We are not always dealing with a 1:1 screen ratio. This leads to severe problems when considering objects--circles can appear to grow and retreat as the angle increases. It is usually necessary to consider the screen aspect ratio when determining how the conic section is to be drawn.

The second assumption of our Cartesian representation, that space is infinite, is obviously limited. We can only draw on the screen. Although the internal software allows the conic to curve back onto the screen after once leaving, be aware that ending the conic off-screen will leave the COP off-screen, which is generally undesirable.

The third assumption, that any interval is infinitely divisible, is perhaps the most troublesome. We actually are dealing with a quantized space that simulates Cartesian space as closely as possible; sometimes this is not good enough. Our mathematical calculations may show that a certain point should be activated by issuing a certain conic command, but when the command is given, the actual pixel activated may be off by one. For this reason, a maximum number of pixels is always defaulted to in the WC command, in the event the desired endpoint is never activated. The drawing will automatically stop if the desired point was not reached in 1280 attempts.

Ramtek

APPENDIX C

INTERACTIVE PERIPHERALS OPERATING PROCEDURES

C-0 INTRODUCTION TO INTERACTIVE PERIPHERALS

There are three types of interactive functions which are available within the RM-9000 display system:

1) Cursor Interaction
2) Joystick/Trackball Interaction
3) Keyboard/Transmitter Interaction

In order to implement and access these interactive peripheral functions, a system must contain the RM-SLC serial link card and the RM-PER interactive peripherals option firmware package.

The purpose of these devices is to allow the human user to interact or communicate with the host processor application program in a machine-readable form, allowing visual feedback through the display system. The use of the cursor and joystick/trackball mechanism is particularly useful in this area.

C-1 CURSOR DESCRIPTION

The RM-SLC serial link card, in addition to support input and output from the keyboard and joystick/trackball devices, also supports the cursor controller/generator hardware. Cursors can be controlled by either a joystick or trackball or by the host processor using the WRITE CURSOR STATE and READ CURSOR STATUS instructions. Up to two cursors may be generated by each RM-SLC card; since up to two RM-SLC cards can be present in a given system, this means that a maximum of four cursors are possible.

C-2 Standard Cursor Pattern

The Display Controller is capable of generating up to four cursors. The cursor appears as a cross (+) on the screen with the center element missing in the configurations and sizes shown in Figure C-1. For the same size monitor, the cursor retains the same physical size regardless of the number of lines/elements or resolution of the system. Referring to Figure C-1, the cursor shown in block "A" is representative of systems with 240 or 256 lines and 320 elements. The "B" block cursor is configured in systems of 240 or 256 lines and 640 elements. The "C" block illustrates cursor for 480 or 512 lines and 640 elements.

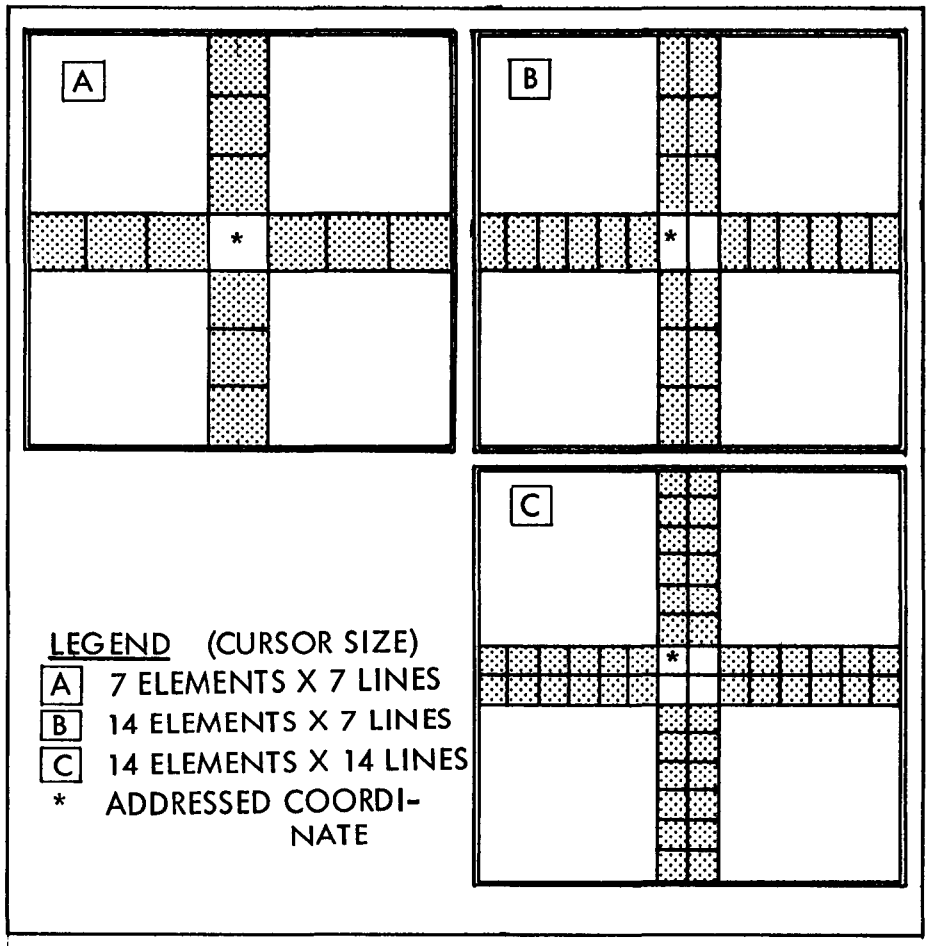


Figure C-1 Cursor Configuration

C-3 Cursor Screen Addressing

The cursor may be placed anywhere on the screen, including non-visible elements and lines in horizontal and vertical blanking. In other words, the cursor will not come to a stop at the edges of the visible portion of the screen, but will go beyond the edges and disappear into blanking. During CPU control of the cursor this is relatively unimportant as the CPU normally is programmed not to load data into non-visible elements. However, under joystick or trackball control, the operator may guide the cursor into blanking. If the same direction is maintained, "roll-over" will eventually occur and the cursor will reappear on the screen on the side opposite to that from which it disappeared, still maintaining the same direction.

C-4 JOYSTICK CURSOR CONTROLLER - GC-106

The joystick cursor controller is an interactive peripheral device used to position a cursor upon a video graphic display. The cursor controller consists of a joystick, four status switches (ENTER, TRACK, VISIBLE and BLINK), four channel select switches and a power switch. The controller interactively positions the cursor via the joystick, controls cursor status with the VISIBLE and BLINK status switches and informs the CPU of current coordinates and status by the ENTER and TRACK switches.

The cursor controller and trackball (See Section C-7) operate with the serial link option in an identical manner. Both use serial transmission lines to send data. The serial link option stores cursor coordinates and status while generating the cursor video image. The cursor controller does not store cursor coordinates, but issues increment/decrement commands to the serial link board which in turn update the cursor position on the screen. Since the amount of displacement of the joystick from the center position proportionately changes the rate of increment/decrement commands issued by the cursor controller, the further the joystick is displaced the faster the cursor moves on the screen. With a little practice, positioning of the cursor with the controller is simple, fast and more efficient than a trackball.

The cursor controller is a directional rate device and not a positional control device. That is, when the joystick is moved in any direction from the center (at rest) position, the cursor begins to move slowly in the direction the joystick was displaced. The further the joystick is displaced from center, the faster the cursor moves in that direction. When the joystick is held in a constant position, the cursor moves across the screen at a constant rate. Release of the joystick returns it to the spring loaded position and stops cursor movement.

The joystick may be displaced at any single angle even though it feels easier to move the stick directly up, down, right or left. When viewed from the front, the

position of the joystick corresponds exactly with the direction of the cursor movements as shown in Table C-1.

Table C-1 Joystick/Cursor Movement

Joystick Movement	Cursor Movement
Forward	Up
Backward	Down
Left	Left
Right	Right

The rate of cursor movement in any axis is infinitely variable from about 1 element/second when the stick is displaced $\simeq 5^{\circ}$ deflection, to a maximum of traversing the screen from one edge to the other in about 3 seconds (full deflection). This mode of operation is used to move the cursor quickly from point to point.

A second mode of joystick operation allows one element cursor movement in any direction to easily and accurately position the cursor on a single screen element. To move the cursor one element only, the joystick is slightly displaced or "bumped". This action causes the cursor to move one element or line in the direction of joystick displacement. The cursor will not move any more elements until the stick is released and "bumped" again, or displaced further to start cursor movement as defined in the above mode.

This unique feature of the cursor controller allows the operator to be assured of moving the cursor one and only one element in any direction for ease in accurate positioning. The joystick displacement versus rate of cursor movement curve is not linear but exponential.

There is a small null zone around the center position of the joystick so that minimal displacements do not cause cursor movement. This prevents the cursor from "creeping" on the screen when the joystick is centered. The null zone also allows the cursor controller to be used without requiring trim adjustments of compensation for drift effects.

C-5 JOYSTICK STATUS CONTROL SWITCHES

Four status switches determine the status of the cursor on the screen and control host processor interrupt generation. These switches are described as follows:

(a) VISIBLE - This alternate action switch turns the cursor ON or OFF. Cursor coordinates are not affected by the position of this switch.

(b) BLINK - The BLINK switch is an alternate action switch that, when ON, causes the cursor to blink at approximately a 1 Hz rate. When BLINK is OFF, the cursor remains steady on the screen. Cursor coordinates remain unaffected by the position of the BLINK switch.

(c) ENTER - ENTER is a momentary switch which causes a cursor interrupt (when enabled) to be sent to the CPU regardless of the position of any status switch or the position of the joystick. If the ENTER switch is held ON, the cursor controller ceases to function until the switch is released. As soon as the ENTER switch is released, the cursor controller resumes normal operation.

(d) TRACK - When ON, this alternate action switch causes every movement of the cursor to generate a host processor interrupt. Every movement of the cursor is defined to be a change in coordinates. When the TRACK switch is OFF, the cursor still moves on the screen, but the cursor interrupt not issued to the host processor.

C-6 Joystick Cursor Selection Switches

Using the four channel select switches, the operator can control up to four cursors simultaneously with one cursor controller unit. These alternate action switches cause the output of the controller to be distributed to the output channel(s) selected by the switches. When a switch for any channel is ON, the output of the controller appears on the serial output for that channel. When the switch is OFF, the serial output for that channel goes to an idle or no transmission mode. Any combination of switches can be ON simultaneously including all ON or all OFF.

NOTE

CHANNEL SELECT SWITCHES SHOULD NEVER BE CHANGED WHILE MOVING THE CURSOR WITH THE JOYSTICK OR WHILE SWITCHING THE STATUS SWITCHES. SINCE THE CONTROLLER OPERATES WITH A SERIAL OUTPUT LINE, CHANGING THE CHANNEL SELECT SWITCHES WHILE THE UNIT IS TRANSMITTING MAY CAUSE UNPREDICTABLE RESULTS OF CURSOR MOVEMENT OR STATUS.

As long as the joystick is centered and the status switches are stationary, the channel select switches can be changed with no effects. Power does not need to be OFF to change the channel select switches.

C-7 TRACKBALL CURSOR CONTROLLER - GC-104-2

The Model GC-104-2 trackball is a self-contained unit providing a serial output to the cursor controller for cursor updating. The trackball itself is a free-turning ball mounted in the GC-104-2 chassis. Cursor update is in the direction of rotation of the ball and the update rate is based upon the linear speed of rotation. The GC-104-2 trackball associated with the RM-9000 display system generates serial outputs at 2400 band.

A serial character is generated whenever any of the control switches changes and whenever the trackball is moved. The four changes of state, which are switch-controlled, are BLINK, VISIBLE, ENTER and TRACK. These have been described previously in the GC-106 joystick description (See Section C-5).

C-8 PROCESS CONTROL KEYBOARD - GK-120

The GK-120 Process Control Keyboard contains the following items as standard:

(a) A 61 key typewriter keyboard that generates all 128 USASC11 codes, including all alphanumeric, graphic and control characters. The keyboard features two key rollover/n key lockout operation. An auto-repeat feature of all keys is standard. Key arrangement is similar to the Model 37 teletype.

(b) Serial input and output options using a choice of EIA Standard RS-232C, TTY current loop, or short-line differential. Baud rates from 50 to 9600 baud; odd, even or no parity and one or two stop bits are selectable by the user.

(c) TTY mode operation allowing the keyboard to appear as a Model 33 teletype, generating a standard 93 character subset of USASCII. (Upper case alpha characters only).

(d) ON/OFF LINE switch allowing flexibility in the mode of operation of the keyboard.

(e) A 12 key cursor/function pad containing cursor up, down, right, left and home commands for computer controlled cursor. The remaining seven keys provide easily accessible function keys defined by the user.

(f) A convenient 12 key numeric pad containing 10 digits and 2 delimiter keys (. and ,). This pad can be modified to provide 12 additional function keys defined by the user.

Optional features include 16 special function keys defined by the user with corresponding CPU controlled status lights. Up to 40 function keys can be provided, each assigned with two codes per key. All function keys generate eight bit codes above ASCII (Octal 200-377). An attention signal activated by the reception of the USASCII defined "Bell" code.

All codes are 8 bit data. The code assignments are shown in Table C-2.

The alphanumeric section contains all functions including alphanumeric, control and graphic characters as defined by USASCII. Octal codes 000 through 177 are generated here.

If the numeric pad is selected, the codes issued are strictly the numeric and delimiter codes of the alphanumeric section. That is, when the keyboard is in the unshifted, uncontrolled mode, pressing the numeral "4" will issue the ASCII code for "4". When in the shifted mode, pressing "4" will also issue the ASCII code for "4".

If the function pad replaces the numeric pad, all functions are defined by the user. The codes issued are above ASCII, Octal 200 and higher.

The cursor pad contains the five cursor controls: cursor up, down, right, left and home. Each key has two codes associated with it providing the ability for slow and fast computer controlled cursor. The remaining 7 keys in the cursor pad are defined by the user. The codes issued are above ASCII, Octal 200 and higher.

If the 16 special function keys are included on the keyboard, all functions are defined by the user. Note that each key is assigned with two codes. The codes are above ASCII, Octal 200 and higher.

Due to MOS encoding techniques, note that all codes are preassigned by Ramtek and are unalterable. However, the user is free to assign function meaning and keytop legends to all keys.

The shift and control keys determine the coded output of all keys. For example, consider the key labeled "A" where "-" indicates key not pressed, "x" indicated key is pressed.

Shift	Control	Function	Octal Code Output
-	-	a	141
x	-	A	101
-	x	SOH	1
x	x	SOH	1

The control key allows only ASCII defined control characters (Octal 000-040 and 177) and all codes not defined by ASCII (Octal 200-377) to be serially outputted. All other codes are inhibited from appearing at the serial outputs when the control key is pressed. This is equivalent to a key being mechanically locked out on a teletype. The shift key does not lockout any codes. As an example of the control key locking out the above indicated codes consider the key labeled with the numeral "3".

Shift	Control	Function	Octal Code Output
-	-	3	63
x	-	3	43
-	x	3	Not Outputted (Lockout Operation)

Notice that both the shift and control keys determine the coded output of the keys in the alphanumeric section, but for all other sections (cursor pad, numeric/function pad and special functions), the shift key has no effect. Only the control key will determine the coded output for the keys in these sections. There are two shift keys on the keyboard with one alternate action shift lock. Only one control key is provided.

Placing the keyboard in TTY mode causes the logic to suppress lower case alpha characters. When in TTY mode, the keyboard generates a 93 character subset of ASCII. Only the keys in the alphanumeric section are affected. The codes for all other sections of the keyboard (cursor pad, numeric/function pad and special function) remain unaffected by TTY mode.

In TTY mode, the codes are altered using the following rules.

(a) ASCII

1) Octal codes 000 to 137: retain as is.
2) Octal codes 140 to 176: subtract Octal 40 to convert these codes to upper case characters. That is, convert 140 through 176 to 100 through 136.
3) Octal code 177 (delete): retain as is.

(b) NON-ASCII

1) Octal codes 200 to 377: retain as is.

This information is tabulated in Table C-2.

The following example shows how TTY mode affects keyboard behavior. In the unshift, uncontrol state, when the operator presses the key "A", normally the code for small letter "a" is issued (Octal 141). However, in TTY mode, the capital letter "A" is issued instead (Octal 101). This is shown as follows:

Shift	Control	Function	Octal Code Output
-	-	A	101
x	-	A	101
-	x	SOH	1
-	x	SOH	1

Tektronix

Table C-2 GK-120 TTY Mode Control

	0	1	2	3	4	5	6	7
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
01	BS	HT	LF	VT	FF	CR	SO	SI
02	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
03	CAN	EM	SUB	ESC	FS	GS	RS	US
04	SP	!	"	#	$	%	&	'
05	(	)	*	+	,	-	.	/
06	0	1	2	3	4	5	6	7
07	8	9	:	;	<	=	>	?
10	@	A	B	C	D	E	F	G
11	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]	^	_
14	`	a	b	c	d	e	f	g
15	h	i	j	k	l	m	n	o
16	p	q	r	s	t	u	v	w
17	x	y	z	{	¦	}	~	DEL

ASCII (octal)

Alphanumeric Section
Numeric Pad

UNCONTROL

CONTROL

	0	1	2	3	4	5	6	7
20	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
21	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
22	f_{17}	f_{18}	f_{19}	f_{20}	↑ f_{21}	f_{22}	← f_{23}	HOME f_{24}
23	← f_{25}	f_{26}	↓ f_{27}	f_{28}	f_{29}	f_{30}	f_{31}	f_{32}
24	f_{33}	f_{34}	f_{35}	f_{36}	f_{37}	f_{38}	f_{39}	f_{40}
25								
26								
27								
30	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
31	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
32	f_{17}	f_{18}	f_{19}	f_{20}	↑ f_{21}	← f_{22}	f_{23}	HOME f_{24}
33	← f_{25}	f_{26}	↓ f_{27}	f_{28}	f_{29}	f_{30}	f_{31}	f_{32}
34	f_{33}	f_{34}	f_{35}	f_{36}	f_{37}	f_{38}	f_{39}	f_{40}
35								
36								
37								

NON-ASCII (octal)

Cursor Pad
Function Pad
Special Functions
Status Lights

The auto repeat feature provides that any key, if held down continuously for longer than one second, will automatically repeat until the key is released. If more than one key is held down, only the first key struck will repeat. The output will not alternate between two keys. The time delay until any key begins to repeat is normally one second. The repeat rate is 10 characters per second for all keyboards, regardless of baud rate. The cursor keys have been defined with two codes per key allowing the ability to distinguish between slow and fast cursor commands. Both the time delay and repeat rate are hard-wired and are not under CPU control.

Two key rollover/n-key lockout is provided such that after a key closure is recognized by the keyboard logic and the appropriate code for that key is issued, all further key depressions are ignored. No further codes will be issued until the first key is released.

N-Key Lockout - The following characteristics are to be expected:

(a) LOCKOUT - If a key is depressed and not released, the code for that key is issued and the keyboard scan stops, locking out any recognition of further key depressions. Any other keys depressed and released will not be recognized. As soon as the original key is released, keyboard scan resumes until another key depression is found.

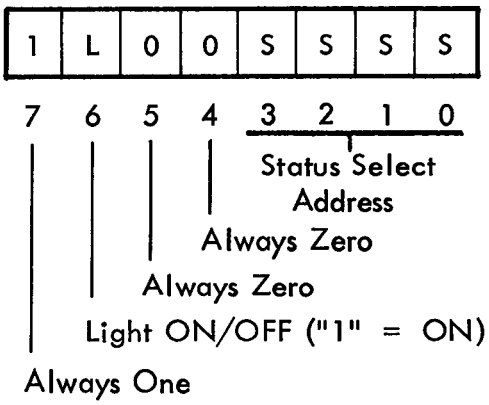
(b) 2-KEY ROLLOVER - A 2-key rollover can be experienced with n-key lockout. Depress the first key and its code is issued, stopping keyboard scan. Press the second key and no code is issued. While still holding the second key down, release the first key (starting keyboard scan) and the second key will now be recognized and its code will be issued. Keyboard scan now stops again until the second key is released. This chain action can be continued indefinitely.

(c) MULTIPLE DEPRESSIONS - If a key is depressed and held, its code will be issued. While continuing to hold the first key, if additional keys are depressed and held, no further codes will be issued until the first key is released and the next code issued will be unpredictable. It depends upon which key is encountered first by the scanning mechanism and only that one code will be issued.

(d) SIMULTANEOUS DEPRESSION - The first code issued is unpredictable. It depends upon the current position of the keyboard scan mechanism. Only one code will be issued.

The CPU can sound an attention signal by using the ASCII defined code for "bell" (BEL = Octal Code 007). The signal will remain active for approximately one second after the keyboard reception of the "bell" code. If a continuous signal is required, the CPU can retrigger the attention device as many times as required. The signal can be retriggered at any point in its cycle. However, the signal will only remain active for one second after the last "bell" code received. The attention signal is Mallory's SONALERT^{C} with a fixed frequency of 2,900 Hz.

Status lights, under host processor control, are provided by the LED's directly above the special function keys. The lights may be turned ON and OFF by the host processor using the WRITE KEYBOARD instruction. The format of the output data byte necessary to set the ON/OFF state of one of the 16 LED's on the GK-120 keyboard. The operator has no control over the operation of the status lights.



Facing the keyboard from the front, the status lights are addressed from 0 to 15 from left to right. Since the operation of the status lights is independent of the special function keys, the user may associate key functions with status lights or completely divorce the two.

Figure C-2 Joystick Model GC-106

Figure C-3 Keyboard Model GK-120

framtek

APPENDIX D

RM-9000 STANDARD CHARACTER FONTS

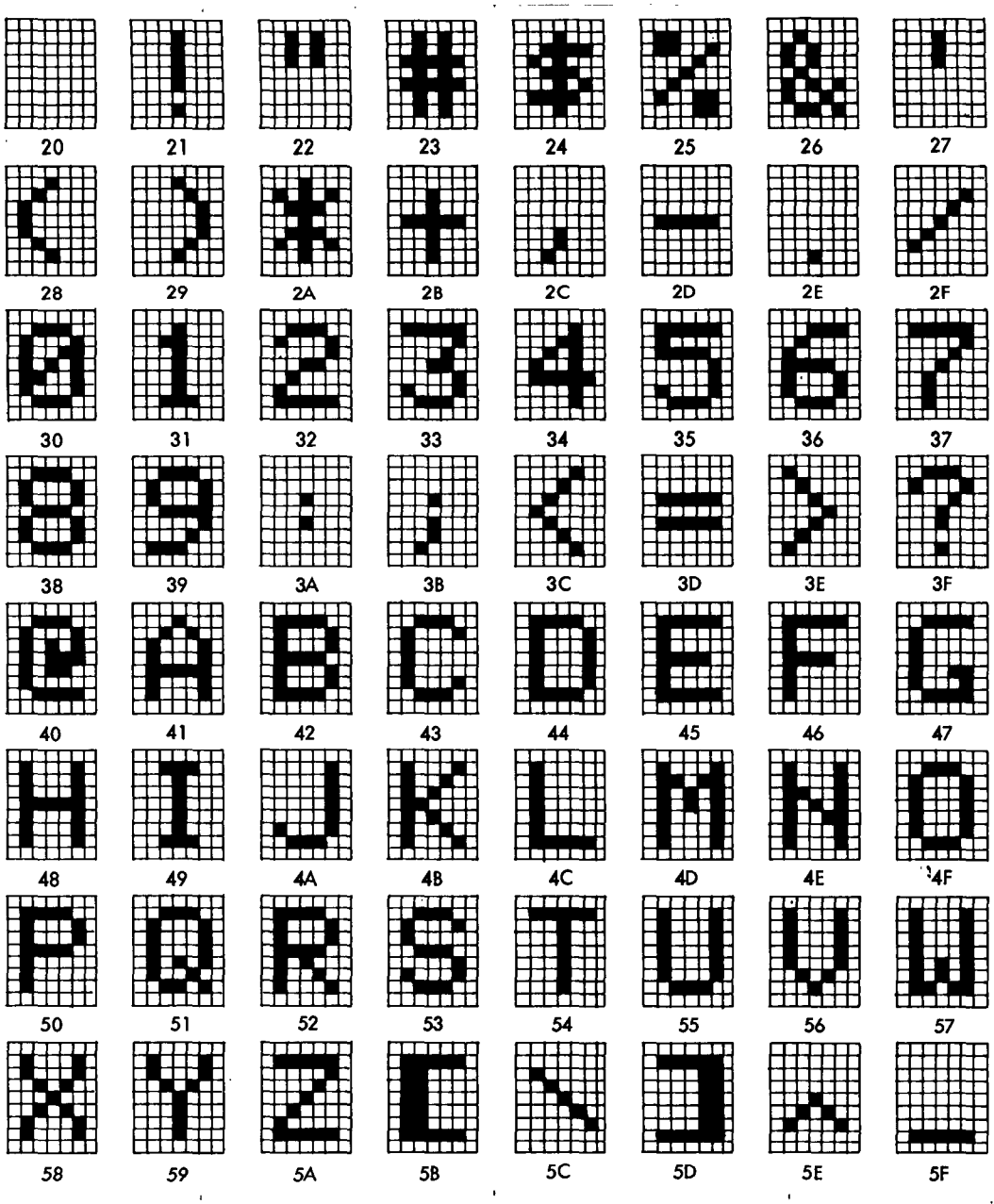


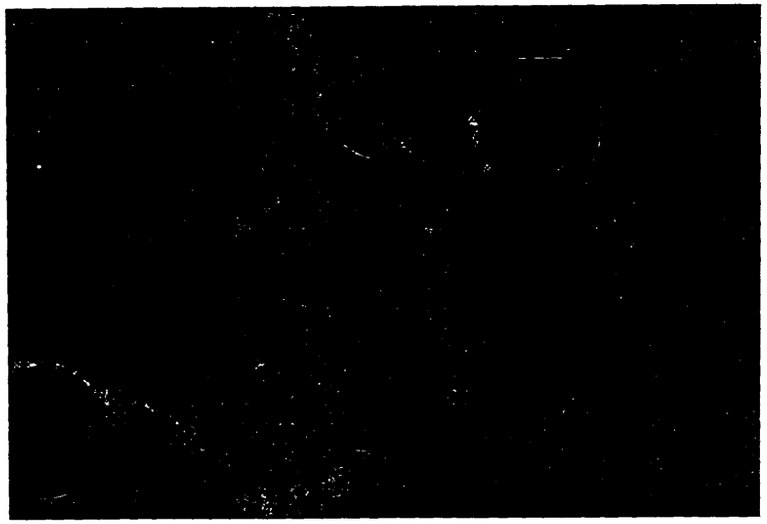
Figure D-1 Standard Text Character Fonts

PIV-000523

Ramtek

Theory of Operation Manual Vol. II

RM-9000 Graphic Display System



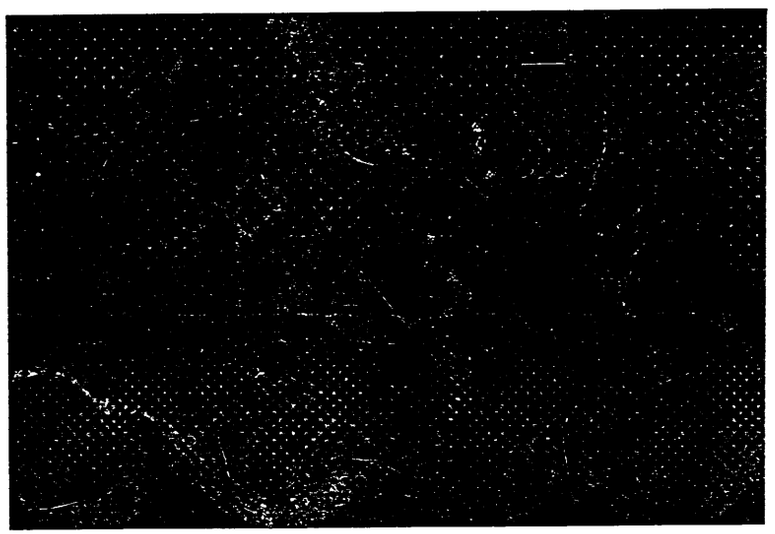
Printed #.

PIV-000524

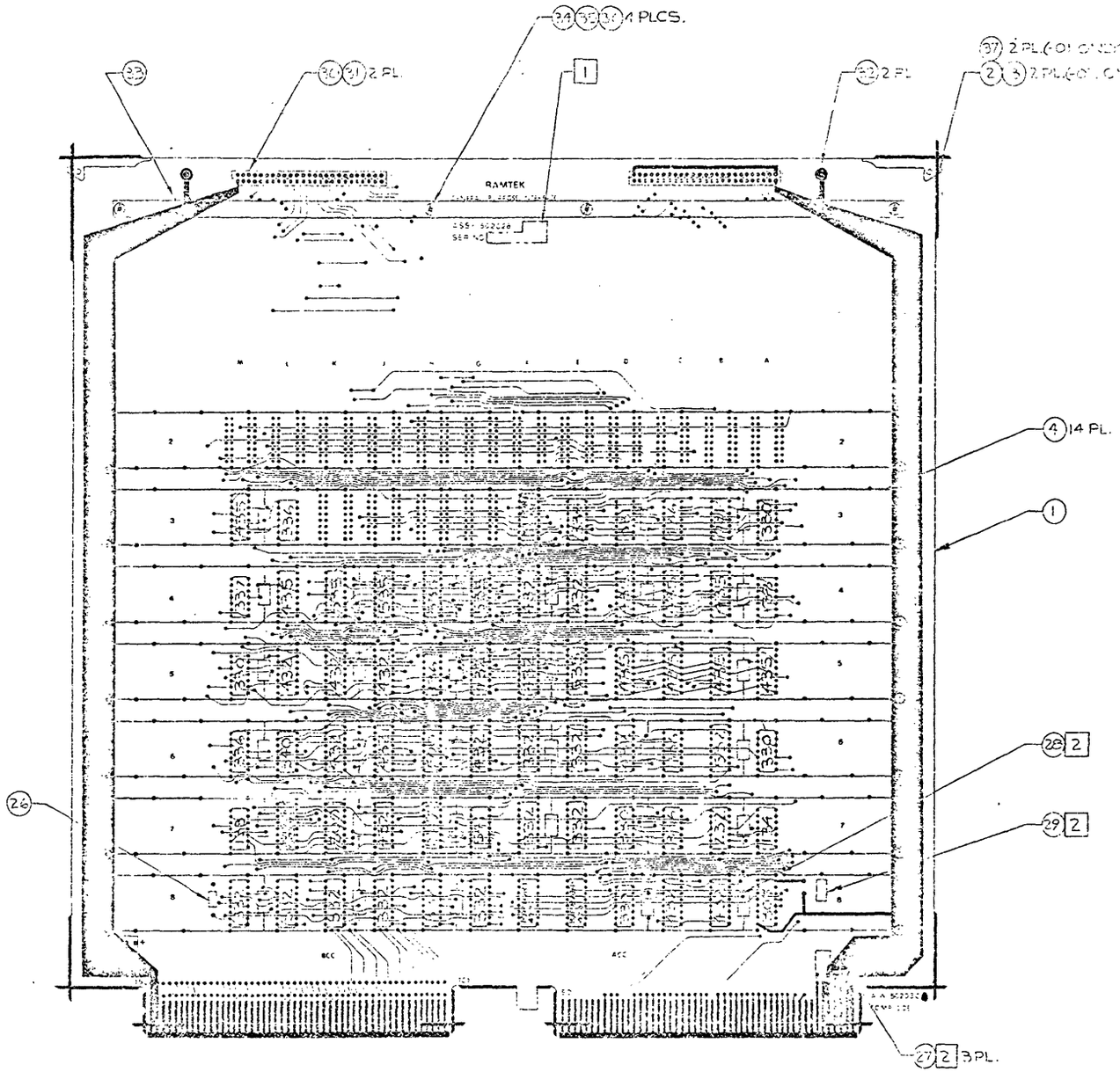
Ramtek

Theory of Operation Manual Vol. II

RM-9000 Graphic Display System

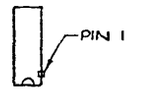


REVISIONS			
LTR	DESCRIPTION	APVD	DATE
-	PROTO RELEASE	[illegible]	1-17-75
A	PROD RTN PER ECO [illegible]	[illegible]	2-14-75
B	REV PER ECO [illegible]	[illegible]	3-7-75
C	REV PER ECO [illegible]	[illegible]	3-10-75
D	REV PER ECO [illegible]	[illegible]	3-19-75
E	REV PER ECO [illegible]	[illegible]	3-30-75

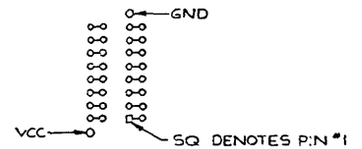


NOTES:

1. STAMP SERIAL & NO. ASSY. REV. LEVEL WITH DASH NO. WITHIN AREA INDICATED.
2. CAPACITORS TO BE INSTALLED LYING FLAT, NOT TO EXCEED .430 ABOVE BOARD SURFACE.
3. LOGIC DIAGRAM NO. 502029
4. FAB DWG. NO. 502030
5. LOGIC DEVICE TYPE NUMBERS ARE RAMTEK TTL STANDARD PART NUMBERS 1301XXX
6. ALL I.C.'S ARE ORIENTED AS SHOWN:



7. * SPARE I.C. LOCATIONS ARE ORIENTED AS SHOWN:



NOTE: (A) LOCATIONS ARE APPLICABLE FOR 14 & 16 PIN SOCKETS.
(B) PWR & GND ARE AVAILABLE AT BUSS BAR TERMINATIONS.
(C) PARALLEL PLATED-THRU HOLES PROVIDE WIRE TO PIN CONNECTIONS.
* THESE I.C. LOCATIONS ARE INTENDED FOR CUSTOMER OPTION USE.

8. BREAK OFF SECTIONS AS REQUIRED.

ITEM NO	PART NO	ITEM DESCRIPTION
		LIST OF MATERIALS

		TOLERANCES UNLESS OTHERWISE SPEC REMOVE ALL BURRS & SHARP EDGES .XX ±.01 .XXX ±.005	DO NOT SCALE DRAWING
-02	RM9000	MATL	SIGNATURES / DATE
-01	FS2000	FINISH	DRAWN [illegible] 1-20-75
DASH NO	NEXT ASSY		CHK'D [illegible] 1-20-75
			ENG [illegible] 1-20-75
			APPR [illegible] 1-20-75

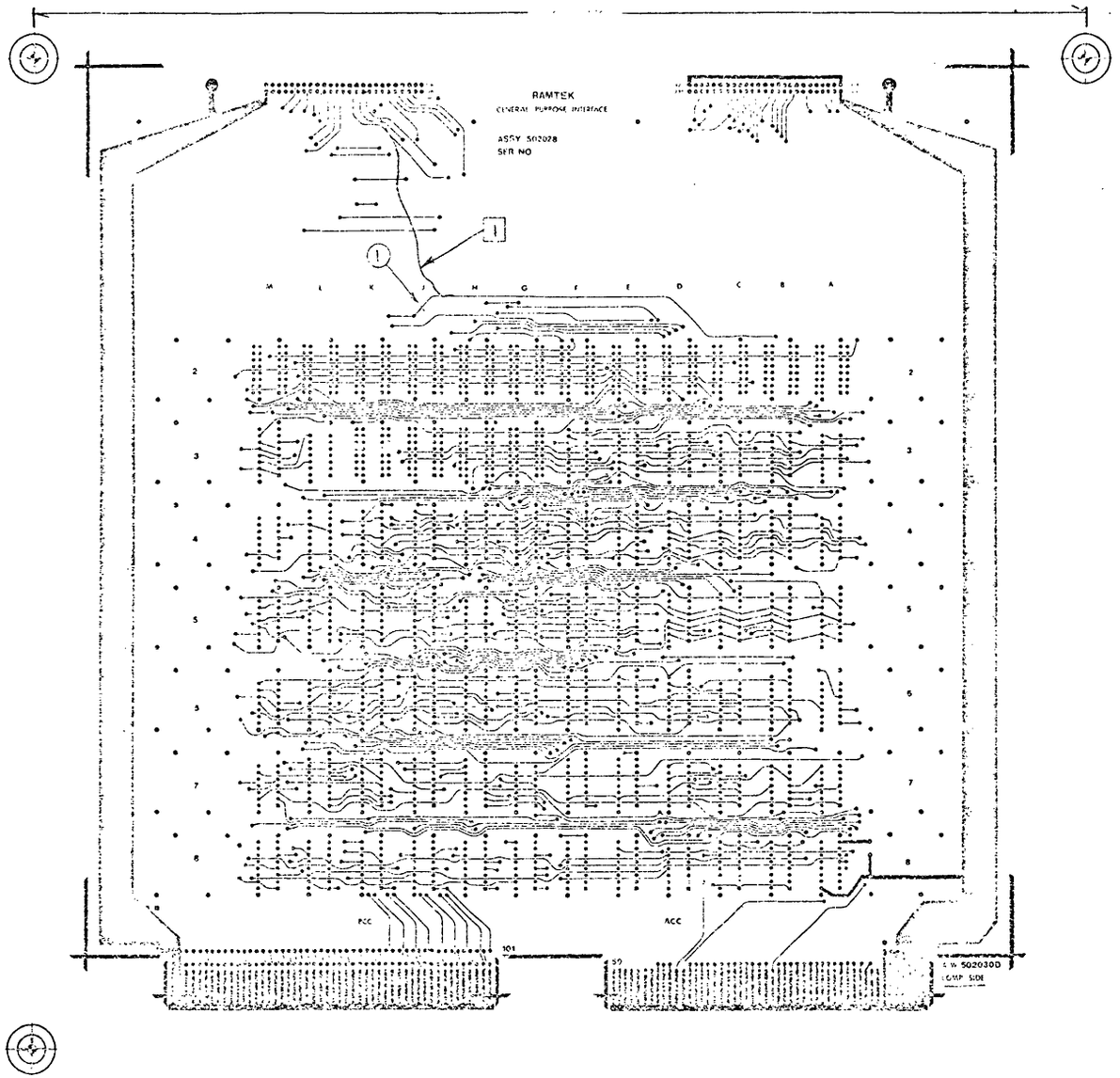
RAMTEK

P.C.B. ASSY, UTILITY GENERAL PURPOSE INTFC

SIZE D | SCALE 1/1 | DWG NO 502028 -01 -02 | REV E

PIV-000526

REVISIONS			
LTR	DESCRIPTION	APVD	DATE

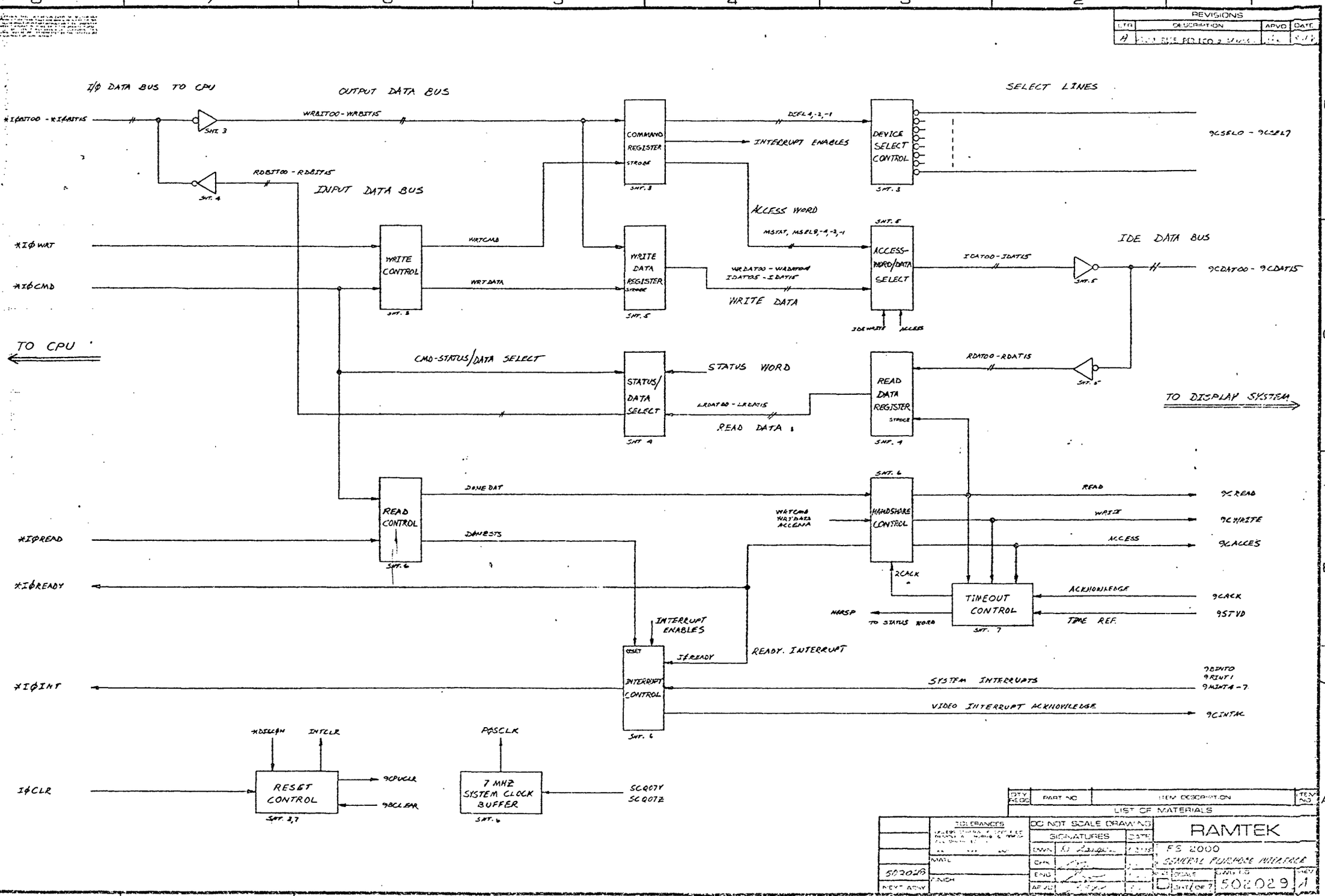


1. CUT TRACE AS SHOWN

1. ADD WIRE FROM TCU253 TO TRACE AS INDICATED.

DO NOT SCALE DRAWING			RAMTEK
	SIGNATURE	DATE	
DWN			
CHK			
ENG			
APVD			SIZE / SCALE / DWG NO / SHT OF

PIV-000527



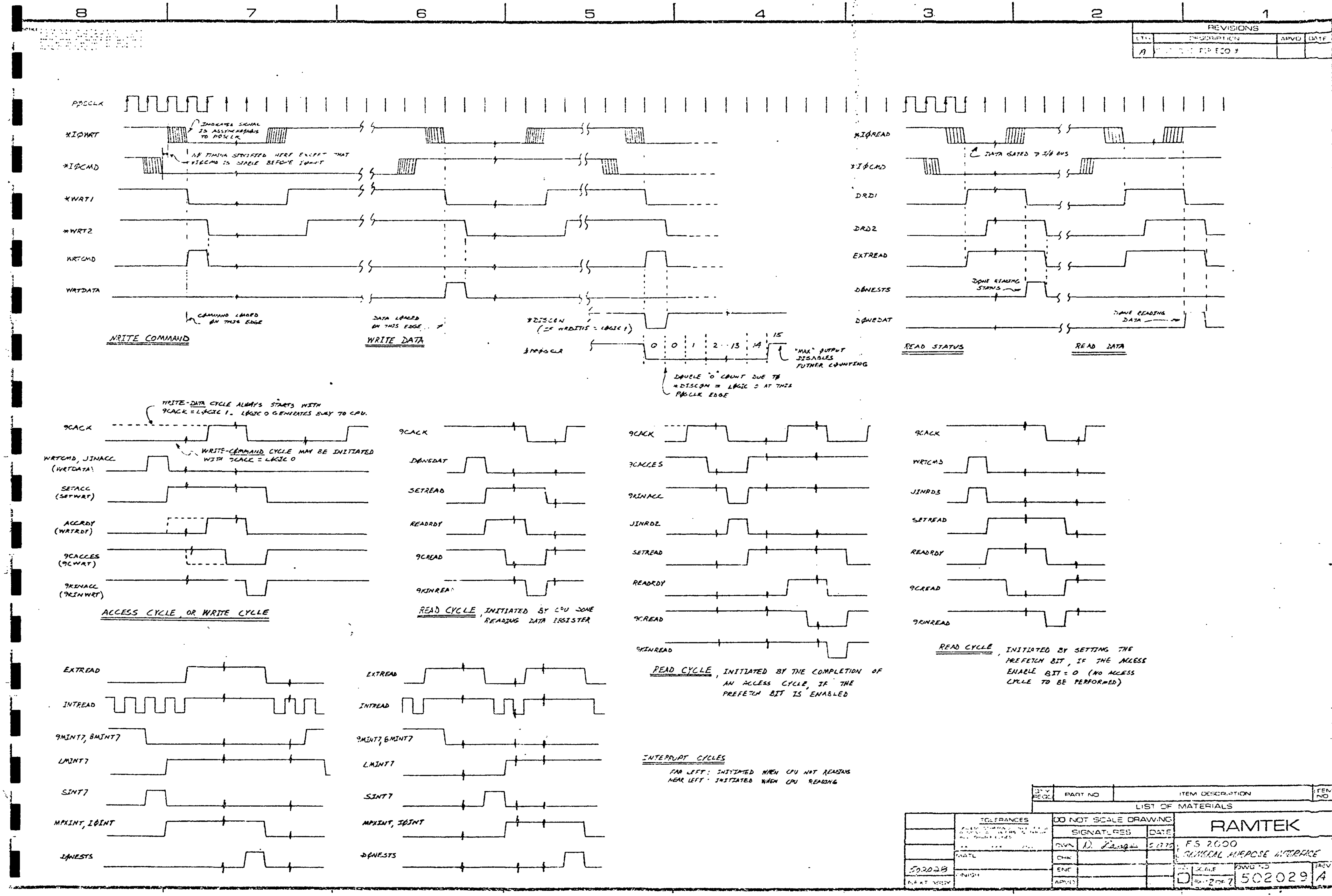
REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	[illegible]	[illegible]	[illegible]

QTY REQD	PART NO	ITEM DESCRIPTION	ITEM NO

LIST OF MATERIALS

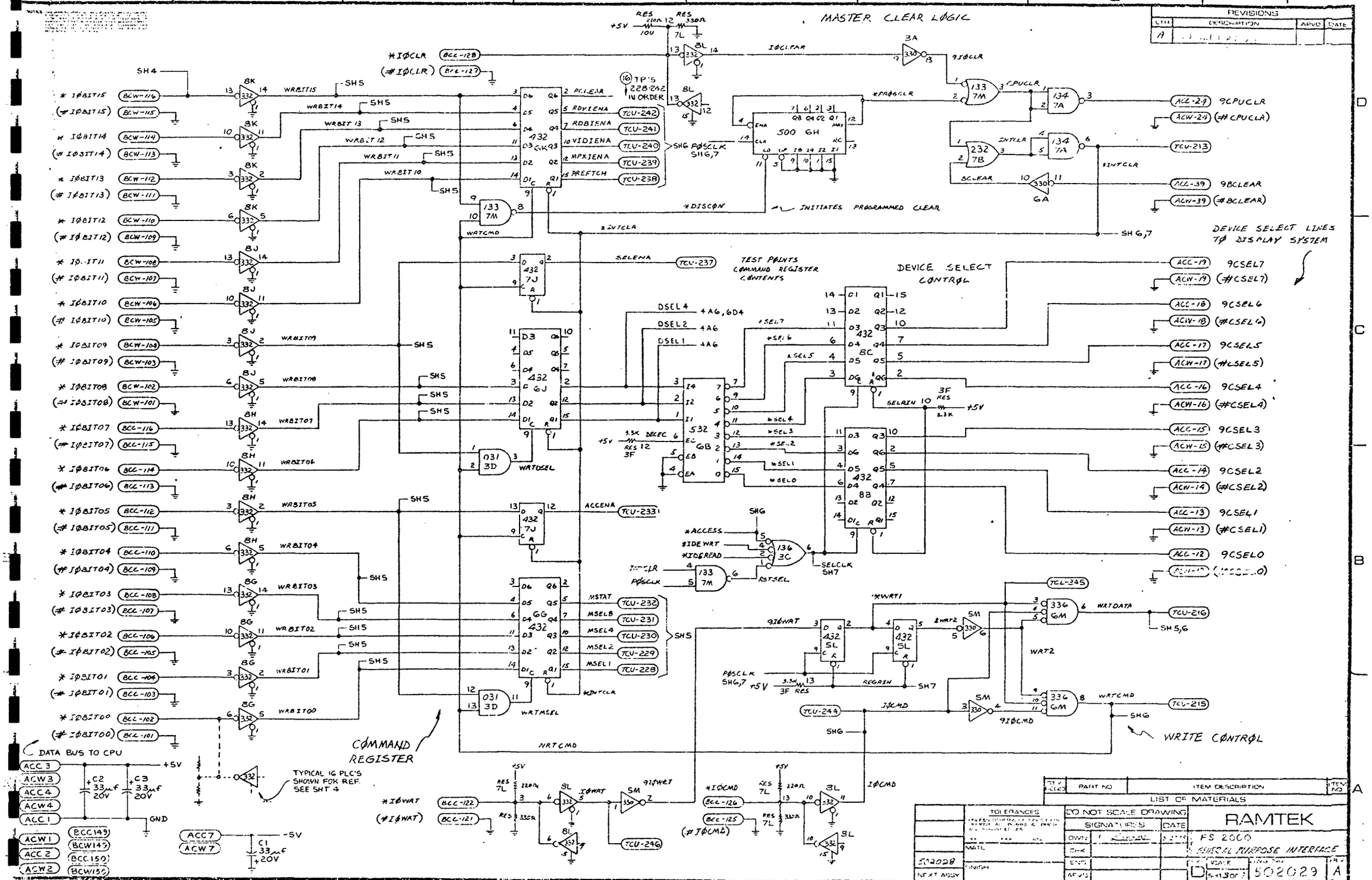
TOLERANCES	DO NOT SCALE DRAWING	RAMTEK	
	SIGNATURES / DATE	FS 2000	
MATL	DWN [illegible]	GENERAL PURPOSE INTERFACE	
502028	CHK		
NEXT ASSY	ENG	SCALE	DWG NO 502029 REV A
FINISH	APVD	SHT 1 OF 7	

PIV-000528

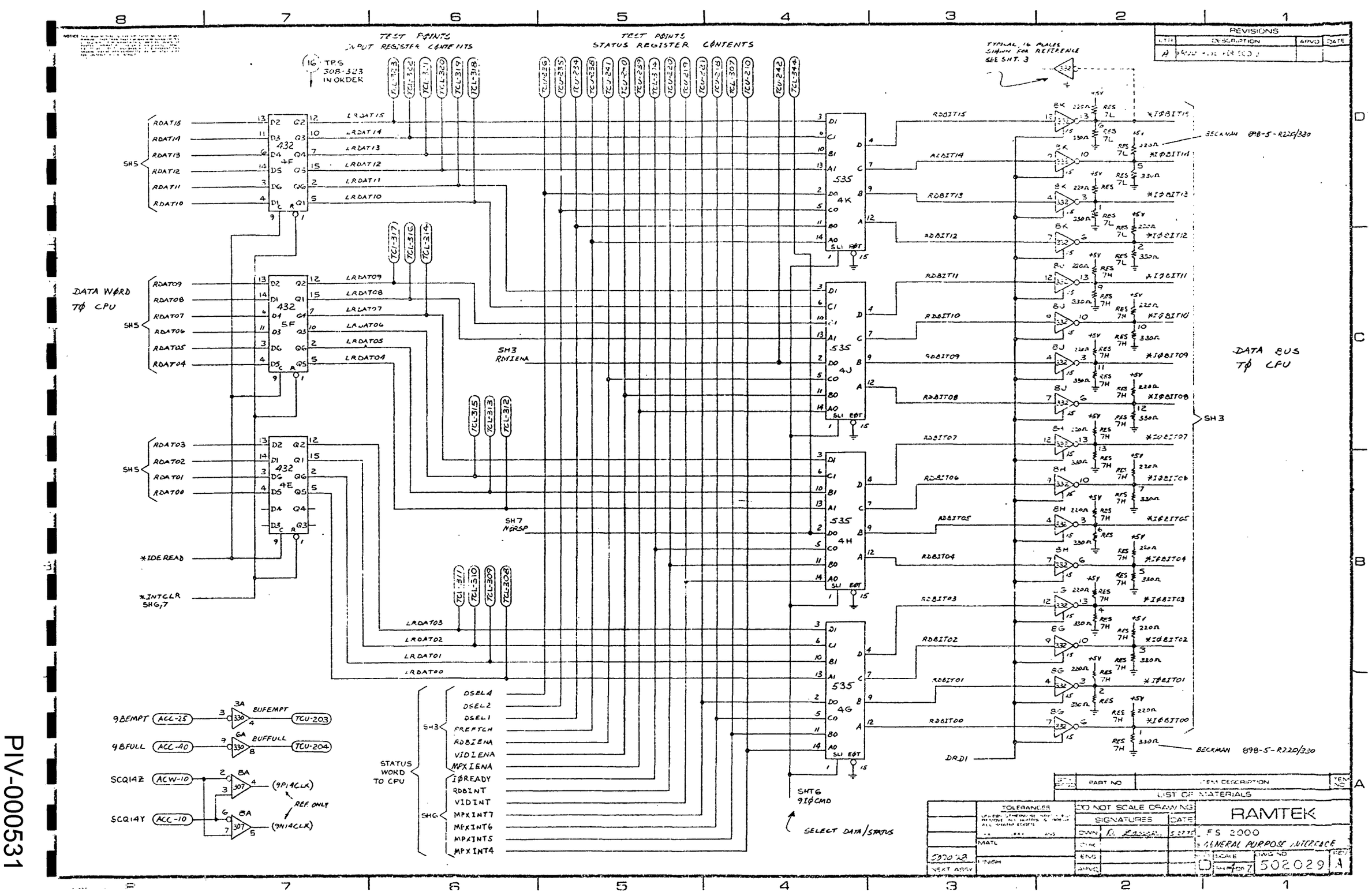


REVISIONS
WRITE COMMAND
WRITE DATA
READ STATUS
READ DATA
ACCESS CYCLE, OR WRITE CYCLE
READ CYCLE, INITIATED BY CPU DONE READING DATA REGISTER
READ CYCLE, INITIATED BY THE COMPLETION OF AN ACCESS CYCLE, IF THE PREFETCH BIT IS ENABLED
READ CYCLE, INITIATED BY SETTING THE PREFETCH BIT, IF THE ACCESS ENABLE BIT = 0 (NO ACCESS CYCLE TO BE PERFORMED)
INTERRUPT CYCLES
FAR LEFT: INITIATED WHEN CPU NOT READING
NEAR LEFT: INITIATED WHEN CPU READING
LIST OF MATERIALS
DO NOT SCALE DRAWING
RAMTEK
FS 2000
GENERAL PURPOSE INTERFACE
502028
502029
A

PIV-000529


MASTER CLEAR LOGIC
REVISIONS
LTR | DESCRIPTION | APVD | DATE
A
*IØCLR
(#IØCLR)
IØCLEAR
9IØCLR
*PRØGCLR
500 GH
INITIATES PROGRAMMED CLEAR
*DISCØN
*INTCLR
9CPUCLR
(#CPUCLR)
9BCLEAR
(#BCLEAR)
SH6,7
DEVICE SELECT LINES TØ DISPLAY SYSTEM
TEST POINTS COMMAND REGISTER CONTENTS
DEVICE SELECT CONTRØL
DSEL4 4A6,6D4
DSEL2 4A6
DSEL1 4A6
9CSEL7
9CSEL6
9CSEL5
9CSEL4
9CSEL3
9CSEL2
9CSEL1
9CSEL0
SELCLK SH7
RSTSEL
ACCENA
SELENA
WRTDATA
WRTCMD
WRTMSEL
WRTCSEL
9IØWAT
9IØCMD
WRITE CØNTRØL
CØMMAND REGISTER
DATA BUS TO CPU
TYPICAL 16 PLC'S SHOWN FOR REF. SEE SHT 4
PART NO | ITEM DESCRIPTION | ITEM NO
LIST OF MATERIALS
DO NOT SCALE DRAWING
SIGNATURES | DATE
RAMTEK
FS 2000
GENERAL PURPOSE INTERFACE
502028
502029
A
SHEET 13 OF 15

PIV-000530



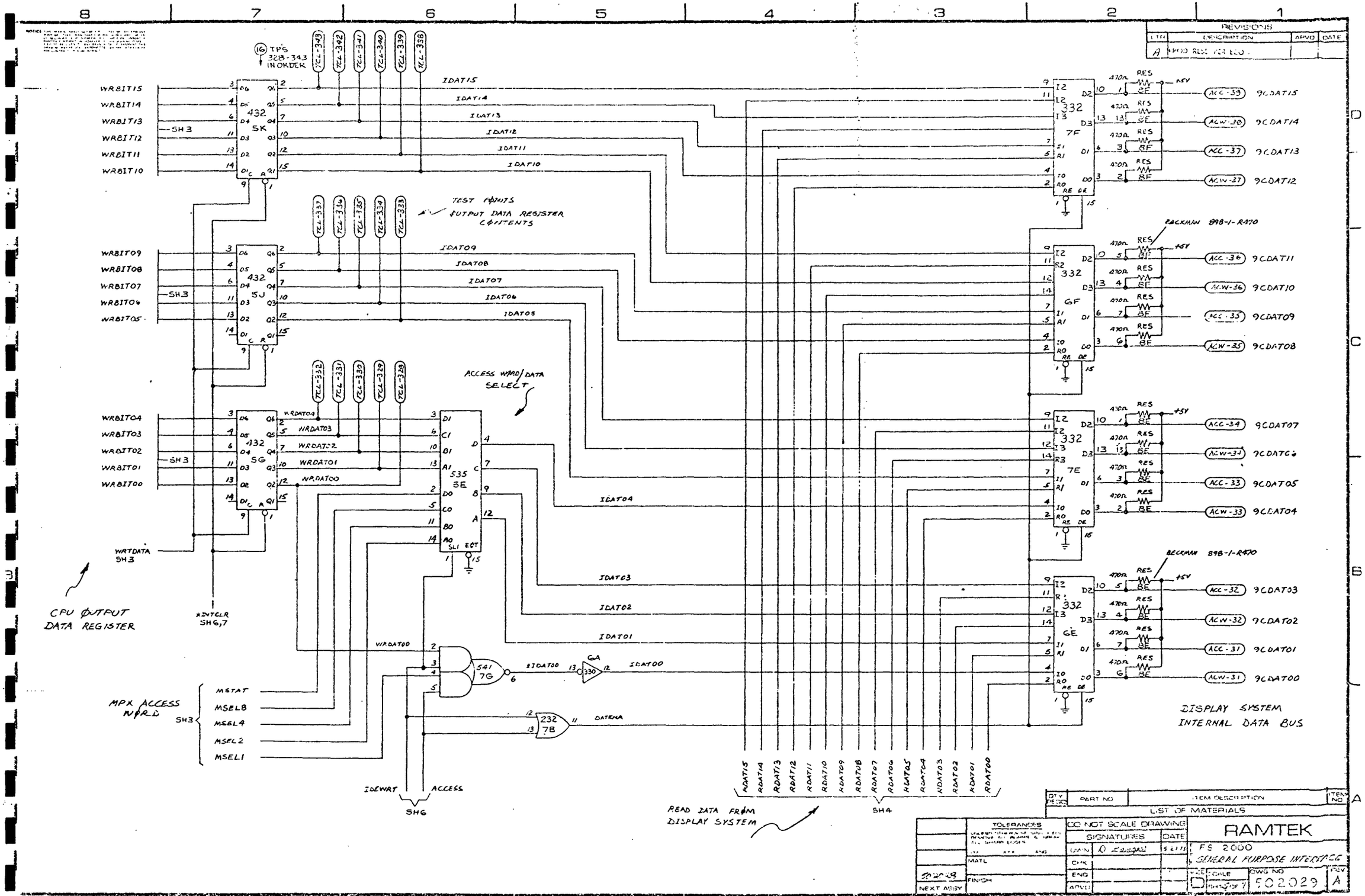
REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	[illegible]		

ITEM NO	PART NO	ITEM DESCRIPTION	ITEM NO

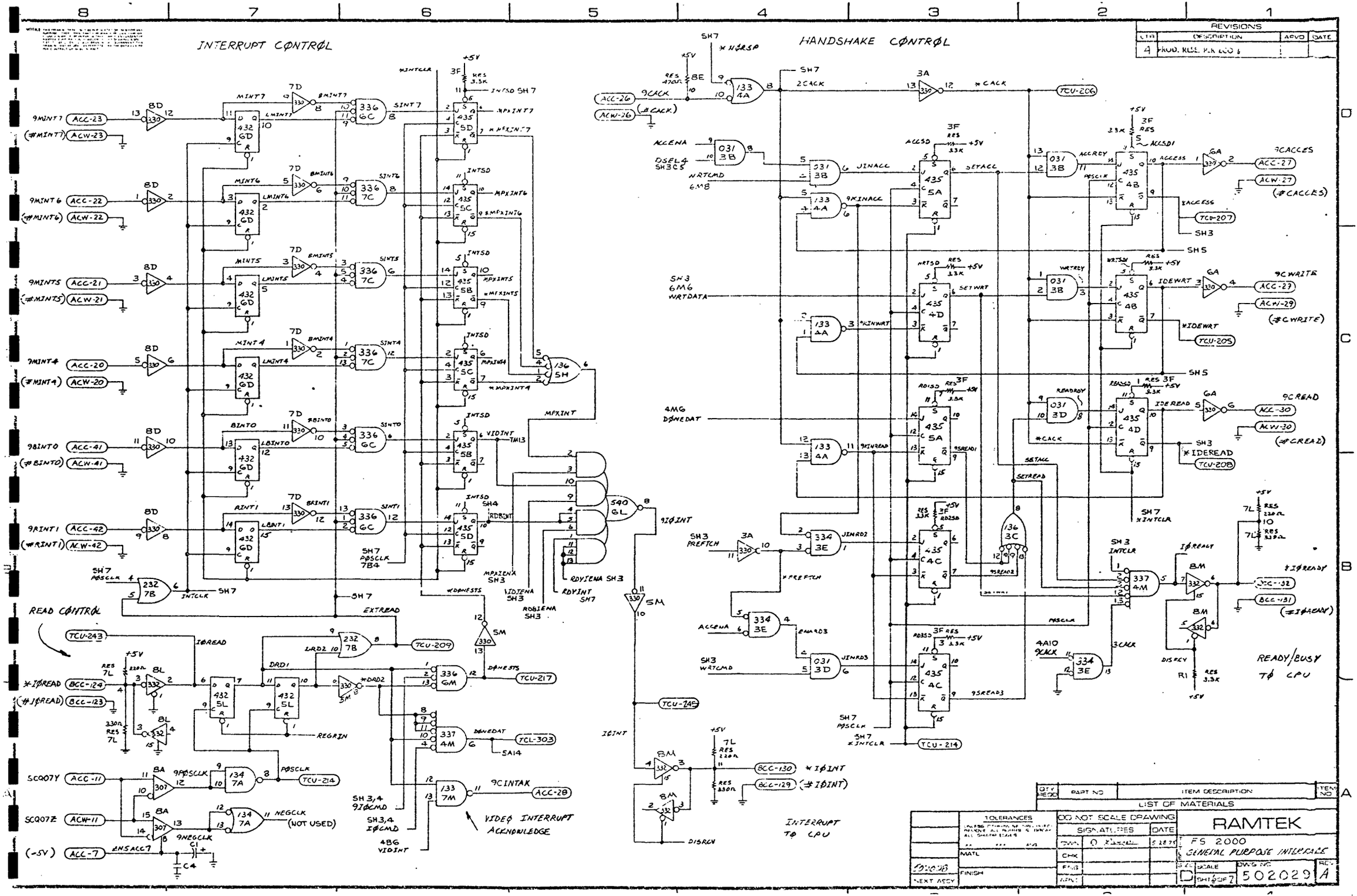
LIST OF MATERIALS

TOLERANCES	DO NOT SCALE DRAWING		RAMTEK
	SIGNATURES	DATE	
MATL	DWN C. Kaiser	5 22 75	FS 2000
FINISH	CHK		GENERAL PURPOSE INTERFACE
NEXT ASSY	ENG		SIZE D SCALE SHEET 7 DWG NO 502029 REV A
	APVD		

PIV-000531

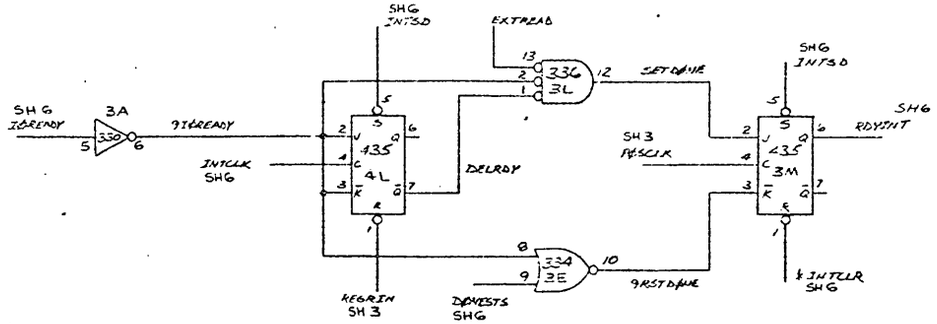

REVISIONS
LTR DESCRIPTION APVD DATE
A PROD RELEASE PER ECO
16 TP'S 328-343 IN ORDER
TCL-343 TCL-342 TCL-341 TCL-340 TCL-339 TCL-338
TCL-337 TCL-336 TCL-335 TCL-334 TCL-333
TCL-332 TCL-331 TCL-330 TCL-329 TCL-328
TEST POINTS
OUTPUT DATA REGISTER CONTENTS
ACCESS WORD/DATA SELECT
WRBIT15 WRBIT14 WRBIT13 WRBIT12 WRBIT11 WRBIT10
WRBIT09 WRBIT08 WRBIT07 WRBIT06 WRBIT05
WRBIT04 WRBIT03 WRBIT02 WRBIT01 WRBIT00
WRDATA SH3
CPU OUTPUT DATA REGISTER
XINTCLR SH6,7
IDAT15 IDAT14 IDAT13 IDAT12 IDAT11 IDAT10
IDAT09 IDAT08 IDAT07 IDAT06 IDAT05
IDAT04 IDAT03 IDAT02 IDAT01 IDAT00
WRDAT04 WRDAT03 WRDAT02 WRDAT01 WRDAT00
MPX ACCESS N#RD
MSTAT MSEL8 MSEL4 MSEL2 MSEL1
IDEWRT ACCESS
SH6
DATENA
RDAT15 RDAT14 RDAT13 RDAT12 RDAT11 RDAT10 RDAT09 RDAT08 RDAT07 RDAT06 RDAT05 RDAT04 RDAT03 RDAT02 RDAT01 RDAT00
READ DATA FROM DISPLAY SYSTEM
SH4
ACC-39 ACC-38 ACC-37 ACC-36 ACC-35 ACC-34 ACC-33 ACC-32 ACC-31
ACW-30 ACW-27 ACW-36 ACW-35 ACW-34 ACW-33 ACW-32 ACW-31
BECKMAN 898-1-R470
DISPLAY SYSTEM INTERNAL DATA BUS
LIST OF MATERIALS
QTY PER ASSY PART NO ITEM DESCRIPTION ITEM NO
DO NOT SCALE DRAWING
SIGNATURES DATE
TOLERANCES
MATL
FINISH
NEXT ASSY
RAMTEK
FS 2000
GENERAL PURPOSE INTERFACE
SIZE D SCALE DWG NO 502029 REV A
SHEET 7 OF 7

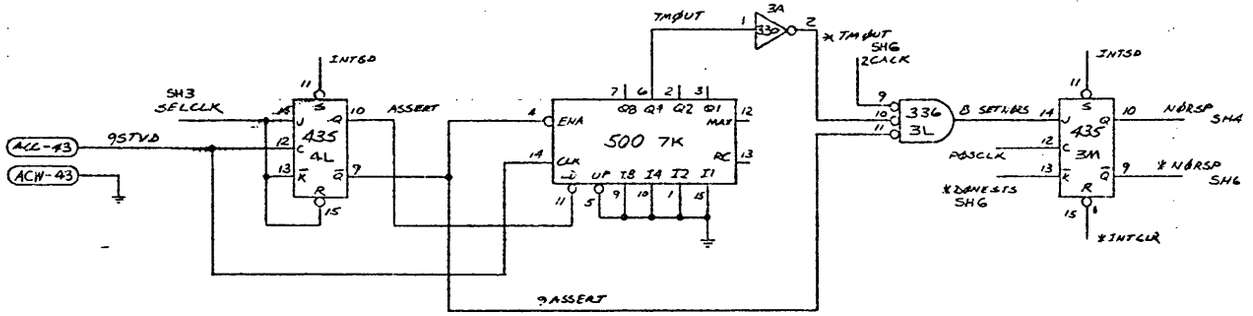
PIV-000532


INTERRUPT CONTROL
HANDSHAKE CONTROL
READ CONTROL
REVISIONS
LTR | DESCRIPTION | APVD | DATE
4 PROD. RLSE. PER ECO 6
LIST OF MATERIALS
QTY REQD | PART NO | ITEM DESCRIPTION | ITEM NO
RAMTEK
FS 2000
GENERAL PURPOSE INTERFACE
DWG NO 502029 REV A
SHT 6 OF 7
DO NOT SCALE DRAWING
TOLERANCES
MATL
FINISH
NEXT ASSY
READY/BUSY TO CPU
INTERRUPT TO CPU
VIDEO INTERRUPT ACKNOWLEDGE
(NOT USED)

PIV-000533

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	MINOR REDR. PER ECO [illegible]		



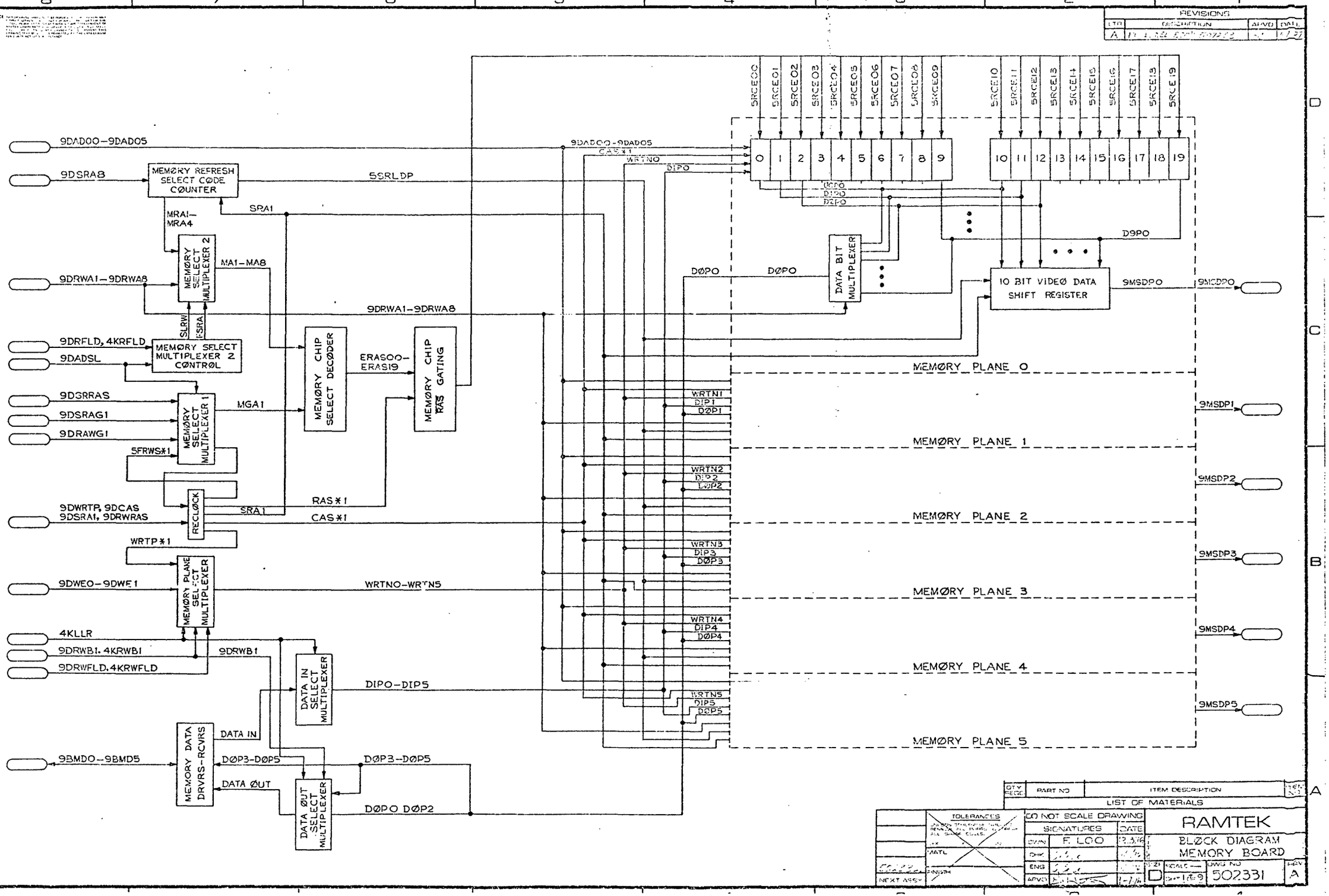


QTY REQD	PART NO	ITEM DESCRIPTION	ITEM NO

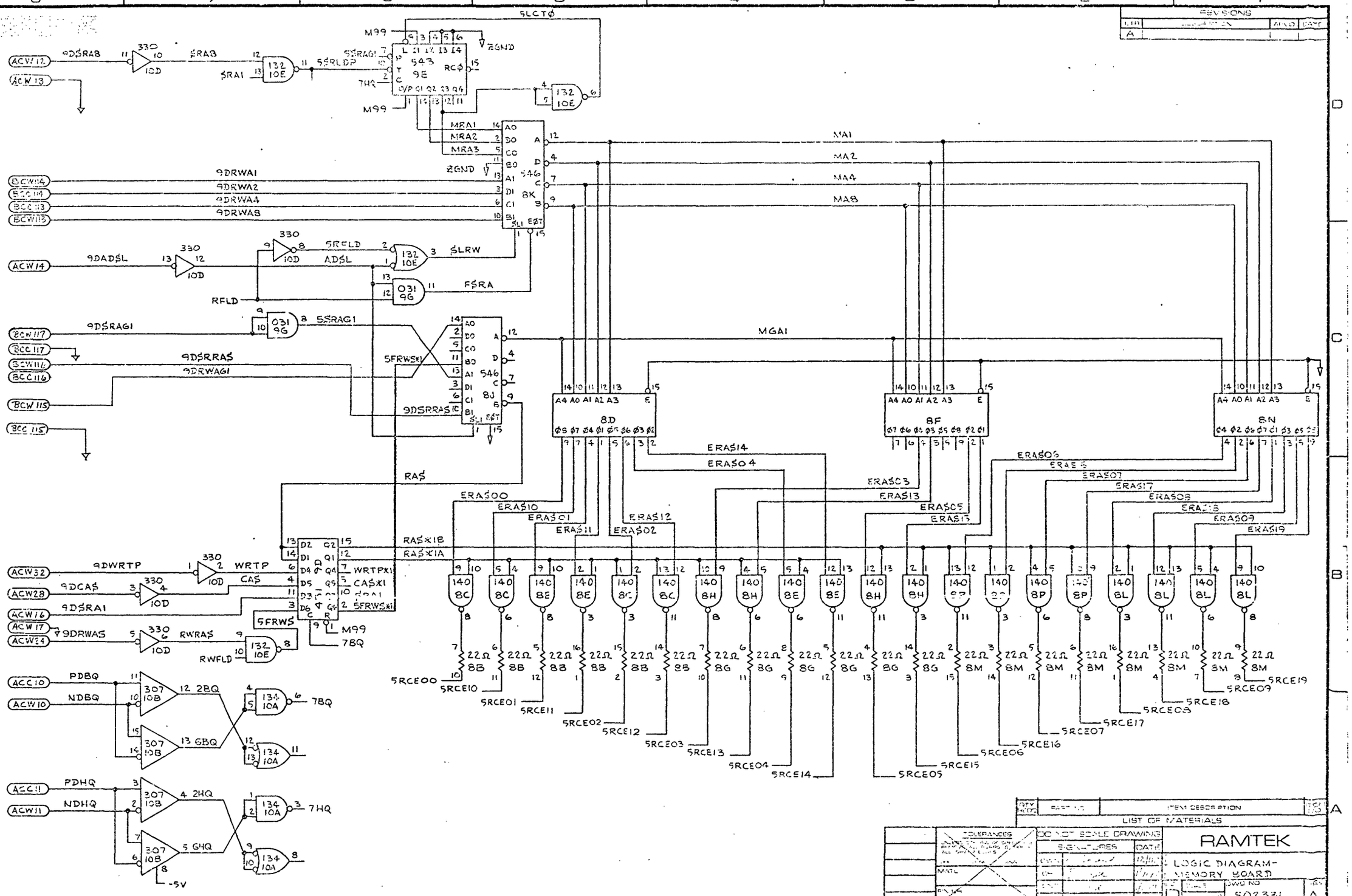
LIST OF MATERIALS

	TOLERANCES UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS IN INCHES	DO NOT SCALE DRAWING			RAMTEK
		SIGNATURES	DATE		PS 2000
	.XX .XXX ANG	DWN D. [illegible]	5-27-75		GENERAL PURPOSE INTERFACE
	MATL	CHK			
SOURCE	FINISH	ENG		SIZE D	SCALE	DWG NO 502029	REV A
NEXT ASSY		APPL			SHT 1 OF 7		

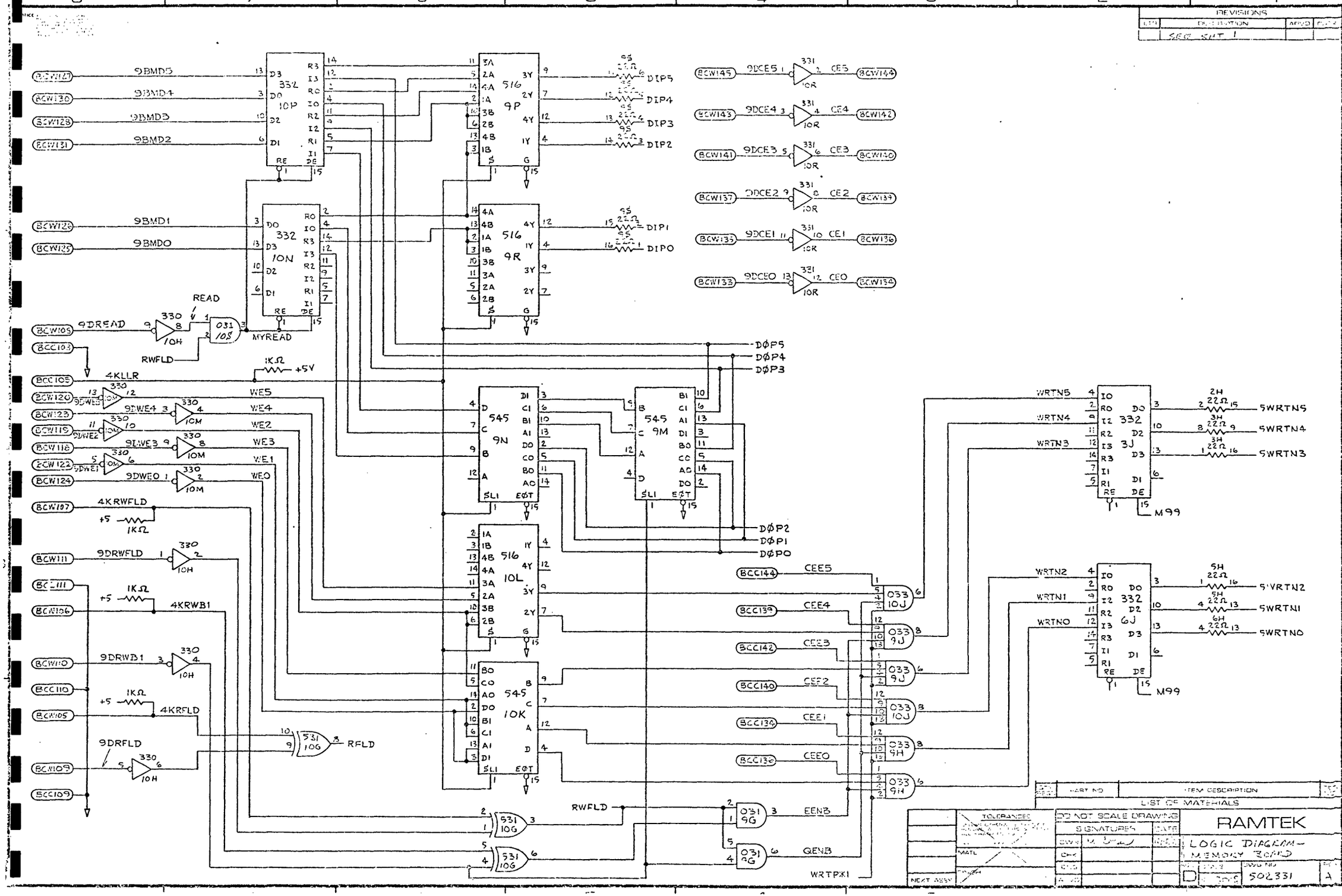
PIV-000534


REVISIONS
LTR DESCRIPTION APVD DATE
A
9DAD00-9DAD05
9DSRA8
MEMORY REFRESH SELECT CODE COUNTER
5SRLDP
SRA1
MRA1-MRA4
MEMORY SELECT MULTIPLEXER 2
MA1-MA8
9DRWA1-9DRWA8
SLRW
SRA
9DRFLD, 4KRFLD
9DADSL
MEMORY SELECT MULTIPLEXER 2 CONTROL
9DSRRAS
9DSRAG1
9DRAWG1
MEMORY SELECT MULTIPLEXER 1
MGA1
5FRWS*1
MEMORY CHIP SELECT DECODER
ERAS00-ERAS19
MEMORY CHIP RAS GATING
RECLOCK
9DWRTP, 9DCAS
9DSRA1, 9DRWRAS
SRA1
RAS*1
CAS*1
WRTP*1
MEMORY PLANE SELECT MULTIPLEXER
9DWE0-9DWE1
WRTN0-WRTN5
4KLLR
9DRWB1, 4KRWB1
9DRWFLD, 4KRWFLD
9DRWB1
DATA IN SELECT MULTIPLEXER
DIP0-DIP5
9BMD0-9BMD5
MEMORY DATA DRVRS-RCVRS
DATA IN
D0P3-D0P5
DATA OUT
DATA OUT SELECT MULTIPLEXER
D0P3-D0P5
D0P0 D0P2
9DAD00-9DAD05
CAS*1
WRTN0
DIP0
D1P0
D2P0
5RCE00 5RCE01 5RCE02 5RCE03 5RCE04 5RCE05 5RCE06 5RCE07 5RCE08 5RCE09
5RCE10 5RCE11 5RCE12 5RCE13 5RCE14 5RCE15 5RCE16 5RCE17 5RCE18 5RCE19
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
D9P0
D0P0
D0P0
DATA BIT MULTIPLEXER
10 BIT VIDEO DATA SHIFT REGISTER
9MSDP0
9MCDP0
MEMORY PLANE 0
WRTN1 DIP1 D0P1
9MSDP1
MEMORY PLANE 1
WRTN2 DIP2 D0P2
9MSDP2
MEMORY PLANE 2
WRTN3 DIP3 D0P3
9MSDP3
MEMORY PLANE 3
WRTN4 DIP4 D0P4
9MSDP4
MEMORY PLANE 4
WRTN5 DIP5 D0P5
9MSDP5
MEMORY PLANE 5
QTY PER ASSY
PART NO
ITEM DESCRIPTION
REV
LIST OF MATERIALS
TOLERANCES
DO NOT SCALE DRAWING
SIGNATURES
DATE
DWN F. LOO 12/3/76
CHK
ENG
APVD
MATL
FINISH
NEXT ASSY
RAMTEK
BLOCK DIAGRAM MEMORY BOARD
SIZE D
SCALE —
DWG NO 502331
REV A
SHT 1 OF 9

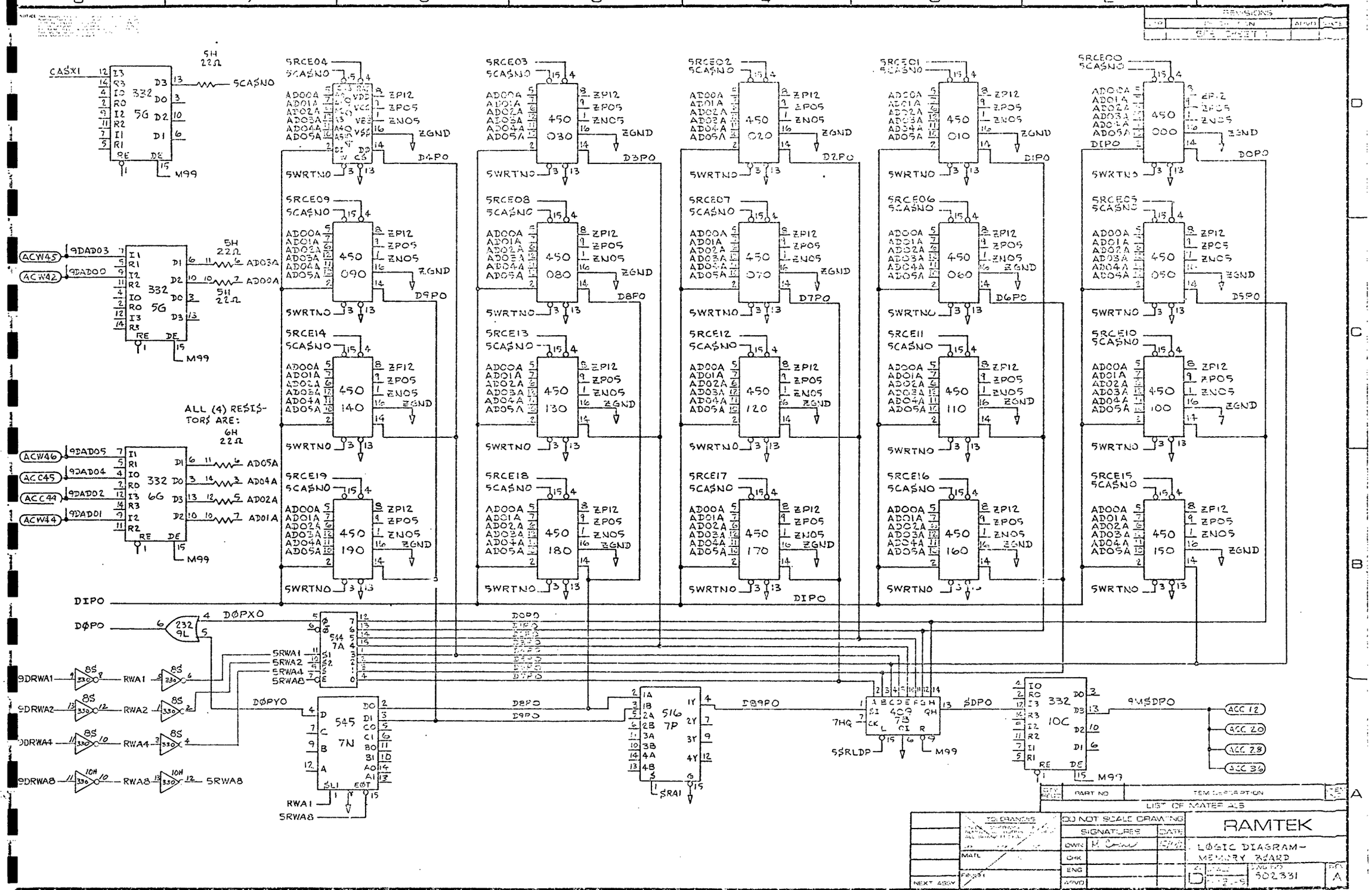
PIV-000535


REVISIONS
LTR DESCRIPTION APVD DATE
A
SLCT0
M99
RGND
543 9E
RC0
MRA1
MRA2
MRA3
546 8K
MA1
MA2
MA4
MA8
9DRWA1
9DRWA2
9DRWA4
9DRWA8
BCW114
BCC114
BCW113
BCC113
ACW14
9DAD0L
330 10D
132 10E
031 9G
SRELD
AD0L
0L0W
F0RA
RFLD
BCW117
BCC117
BCW116
BCC116
BCW115
BCC115
9D0RAGI
9D0RRA0
9DRWAGI
9D0RRA0
546 8J
5FRWSXI
MGAI
RA0
8D
8F
8N
ERA0I4
ERA0I0
ERA0I4
ERA0C3
ERA0I3
ERA0I2
ERA0I1
ERA0I5
ERA0I7
ERA0I8
ERA0I9
RA0X1B
RA0X1A
ACW32
9DWRTP
WRTP
CA0
WRTPXI
CA0XI
ACW28
9DCA0
ACW26
9D0RA1
ACW17
ACW24
9DRWA0
RWRA0
RWFLD
5FRW0
M99
7BQ
140 8C
140 8E
140 8H
140 8P
140 6L
140 8L
22Ω 8B
22Ω 8G
22Ω 8M
5RCE00
5RCE10
5RCE01
5RCE11
5RCE02
5RCE12
5RCE03
5RCE13
5RCE04
5RCE14
5RCE05
5RCE15
5RCE06
5RCE16
5RCE07
5RCE17
5RCE08
5RCE18
5RCE09
5RCE19
ACC10
PDBQ
ACW10
NDBQ
307 10B
12 2BQ
13 6BQ
134 10A
7BQ
ACC11
PDHQ
ACW11
NDHQ
4 2HQ
5 6HQ
7HQ
-5V
QTY PART NO ITEM DESCRIPTION ITEM
LIST OF MATERIALS
TOLERANCES
DO NOT SCALE DRAWING
SIGNATURES DATE
MATL
NEXT ASSY
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE DWG NO REV
D 502321 A

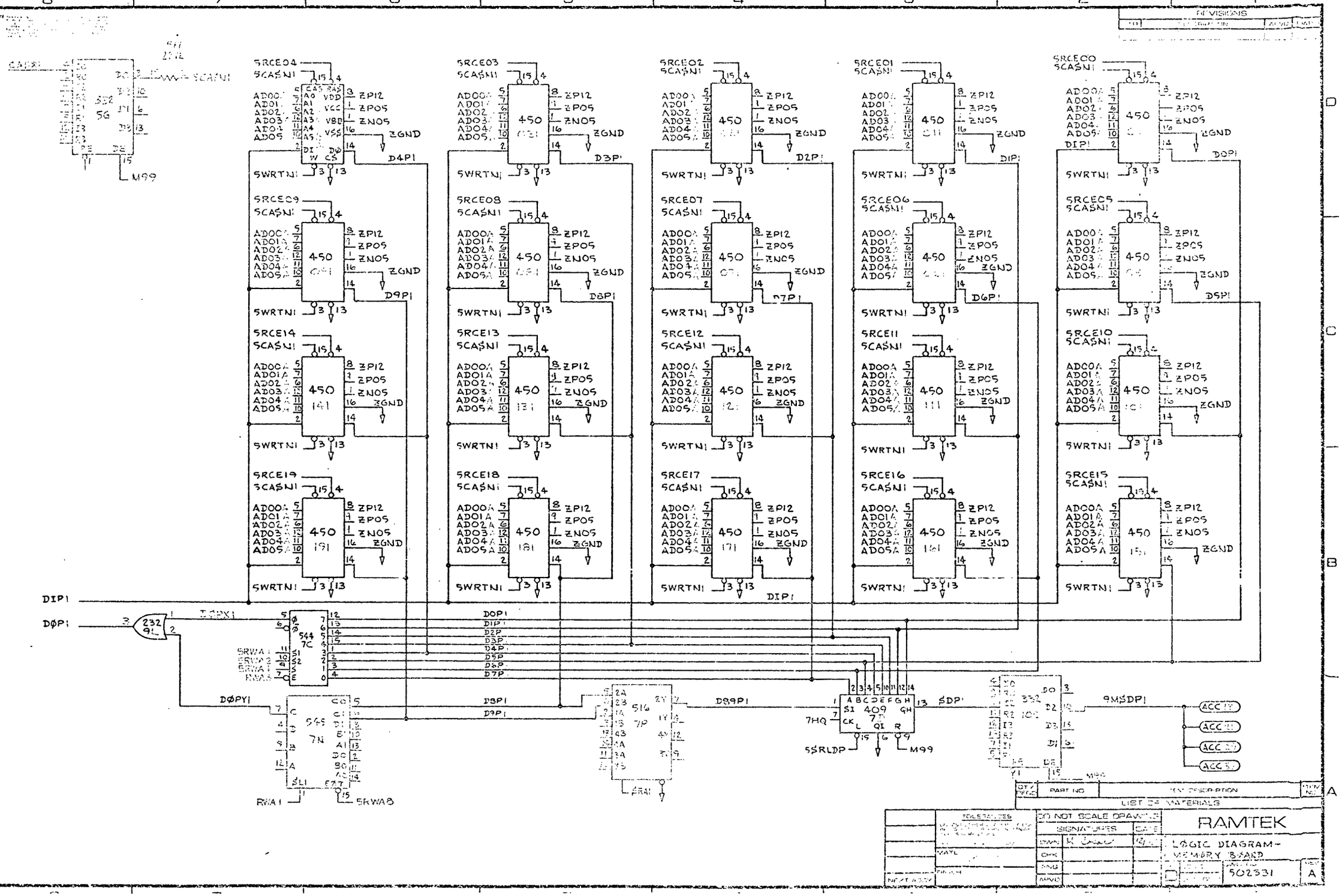
PIV-000536


REVISIONS
SEE SHT 1
9BMD5
9BMD4
9BMD3
9BMD2
9BMD1
9BMD0
332
10P
10N
516
9P
9R
DIP5
DIP4
DIP3
DIP2
DIP1
DIP0
9DCE5
9DCE4
9DCE3
9DCE2
9DCE1
9DCE0
331
10R
CE5
CE4
CE3
CE2
CE1
CE0
READ
9DREAD
330
10H
031
10S
MYREAD
RWFLD
4KLLR
1KΩ
+5V
DØP5
DØP4
DØP3
DØP2
DØP1
DØPO
545
9N
9M
WE5
WE4
WE2
WE3
WE1
WE0
9DWE5
9DWE4
9DWE2
9DWE3
9DWE1
9DWE0
10M
4KRWFLD
9DRWFLD
10L
4KRWB1
9DRWB1
10K
4KRFLD
9DRFLD
531
10G
RFLD
RWFLD
031
9G
EENB
QENB
WRTPXI
WRTN5
WRTN4
WRTN3
WRTN2
WRTN1
WRTN0
3J
6J
9H
9J
10J
033
M99
5WRTN5
5WRTN4
5WRTN3
5WRTN2
5WRTN1
5WRTN0
CEE5
CEE4
CEE3
CEE2
CEE1
CEEO
BCW145
BCW144
BCW143
BCW142
BCW141
BCW140
BCW137
BCW139
BCW135
BCW136
BCW133
BCW134
BCW105
BCC103
BCC105
BCW120
BCW123
BCW119
BCW118
BCW122
BCW124
BCW107
BCW111
BCC111
BCW106
BCW110
BCC110
BCW105
BCW109
BCC109
BCC144
BCC139
BCC142
BCC140
BCC134
BCC136
PART NO
ITEM DESCRIPTION
LIST OF MATERIALS
TOLERANCES
DO NOT SCALE DRAWING
SIGNATURES
DWN
CHK
MATL
FINISH
NEXT ASSY
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE
D
DWG NO
502331
REV
A

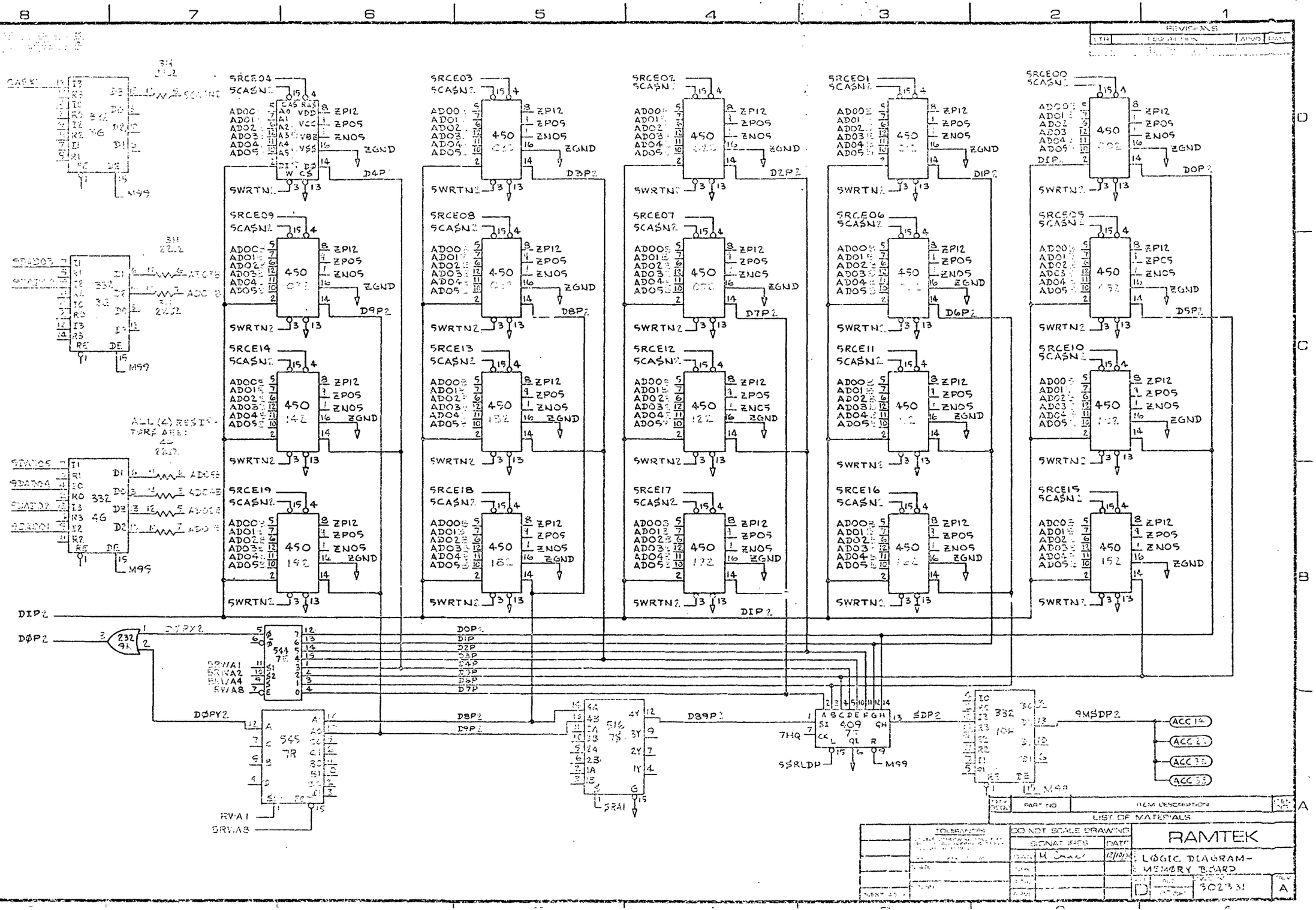
PIV-000537


REVISIONS
SHEET 1
CASX1
5H 22Ω
SCA$NO
M99
ACW45
ACW42
ACW46
ACC45
ACC44
ACW44
ALL (4) RESIS-TORS ARE: 6H 22Ω
SRCE04
SRCE09
SRCE14
SRCE19
SRCE03
SRCE08
SRCE13
SRCE18
SRCE02
SRCE07
SRCE12
SRCE17
SRCE01
SRCE06
SRCE11
SRCE16
SRCE00
SRCE05
SRCE10
SRCE15
ADD0A
ADD1A
ADD2A
ADD3A
ADD4A
ADD5A
ZPI2
ZPOS
ZNOS
ZGND
SWRTNO
D4PO
D9PO
D8PO
D3PO
D2PO
D7PO
D6PO
D1PO
D0PO
D5PO
DIPO
DØPO
DØPXO
DØPYO
D89PO
D8PO
D9PO
$DPO
9M$DPO
SRWA1
SRWA2
SRWA4
SRWA8
RWA1
RWA2
RWA4
RWA8
$RA1
$RLDP
7HQ
ACC 12
ACC 20
ACC 28
ACC 36
PART NO
DESCRIPTION
LIST OF MATERIALS
DO NOT SCALE DRAWING
SIGNATURES
DATE
DWN
CHK
ENG
APVD
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE
D
DWG NO
502331
REV
A
SCALE
SHEET
NEXT ASSY
MATL
FINISH

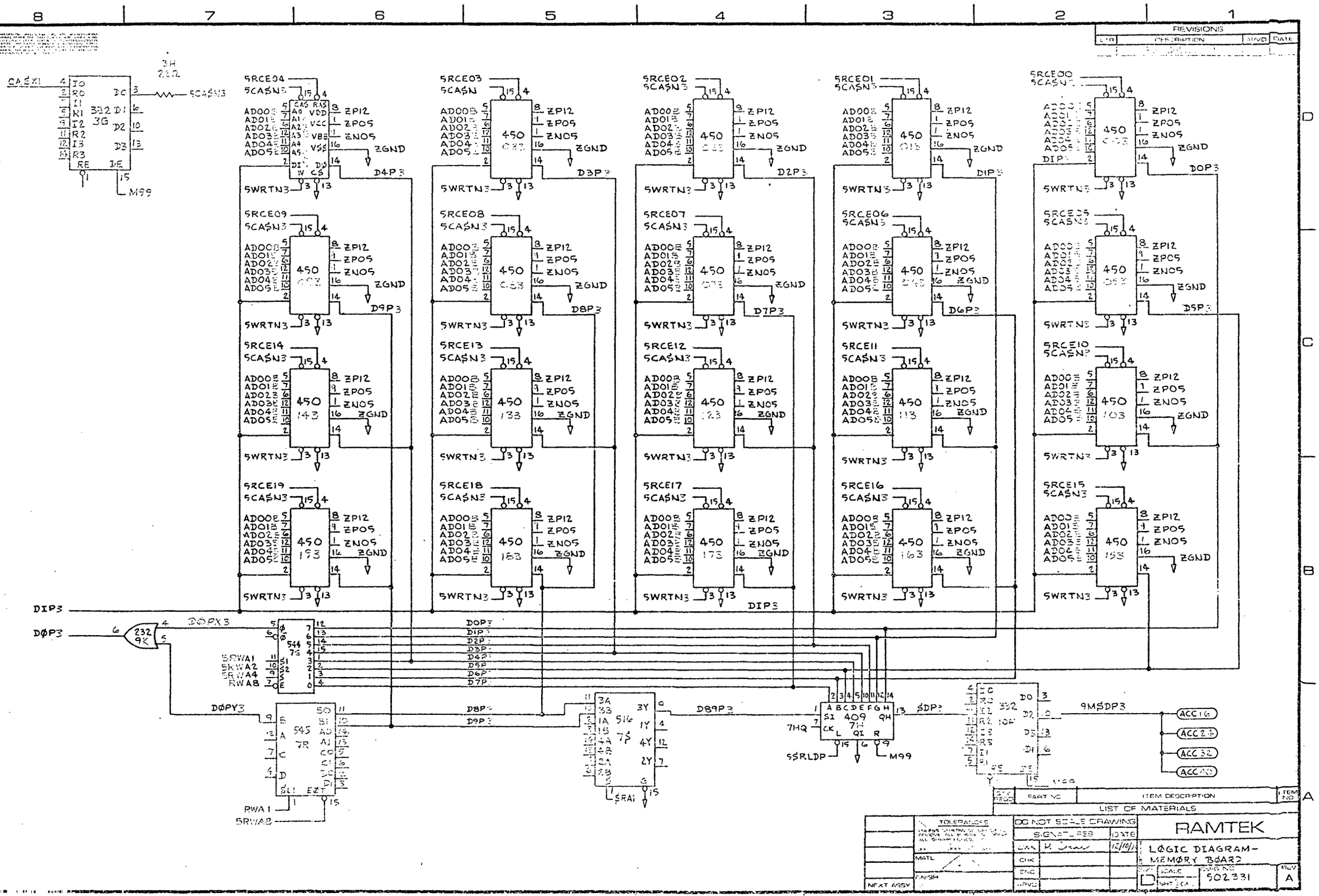
PIV-000538


REVISIONS
SRCE04
SRCE03
SRCE02
SRCE01
SRCE00
SRCE09
SRCE08
SRCE07
SRCE06
SRCE05
SRCE14
SRCE13
SRCE12
SRCE11
SRCE10
SRCE19
SRCE18
SRCE17
SRCE16
SRCE15
450
SWRTN!
ZPI2
ZPOS
ZNOS
ZGND
D4P!
D3P!
D2P!
D1P!
D0P!
D9P!
D8P!
D7P!
D6P!
D5P!
DIP!
DOP!
D2P
D3P
D4P
D5P
D6P
D7P
D8P!
D9P!
DØPX!
DØP!
DØPY!
SCA$N!
M99
5RWA!
5RWAB
RWA!
5$RLDP
$DP!
9M$DP!
ACC 17
ACC 16
ACC 15
ACC 14
LIST OF MATERIALS
QTY REQ'D
PART NO
ITEM DESCRIPTION
ITEM NO
DO NOT SCALE DRAWING
SIGNATURES
DATE
DWN
CHK
ENG
APVD
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
502331
A
SHEET 1 OF 1

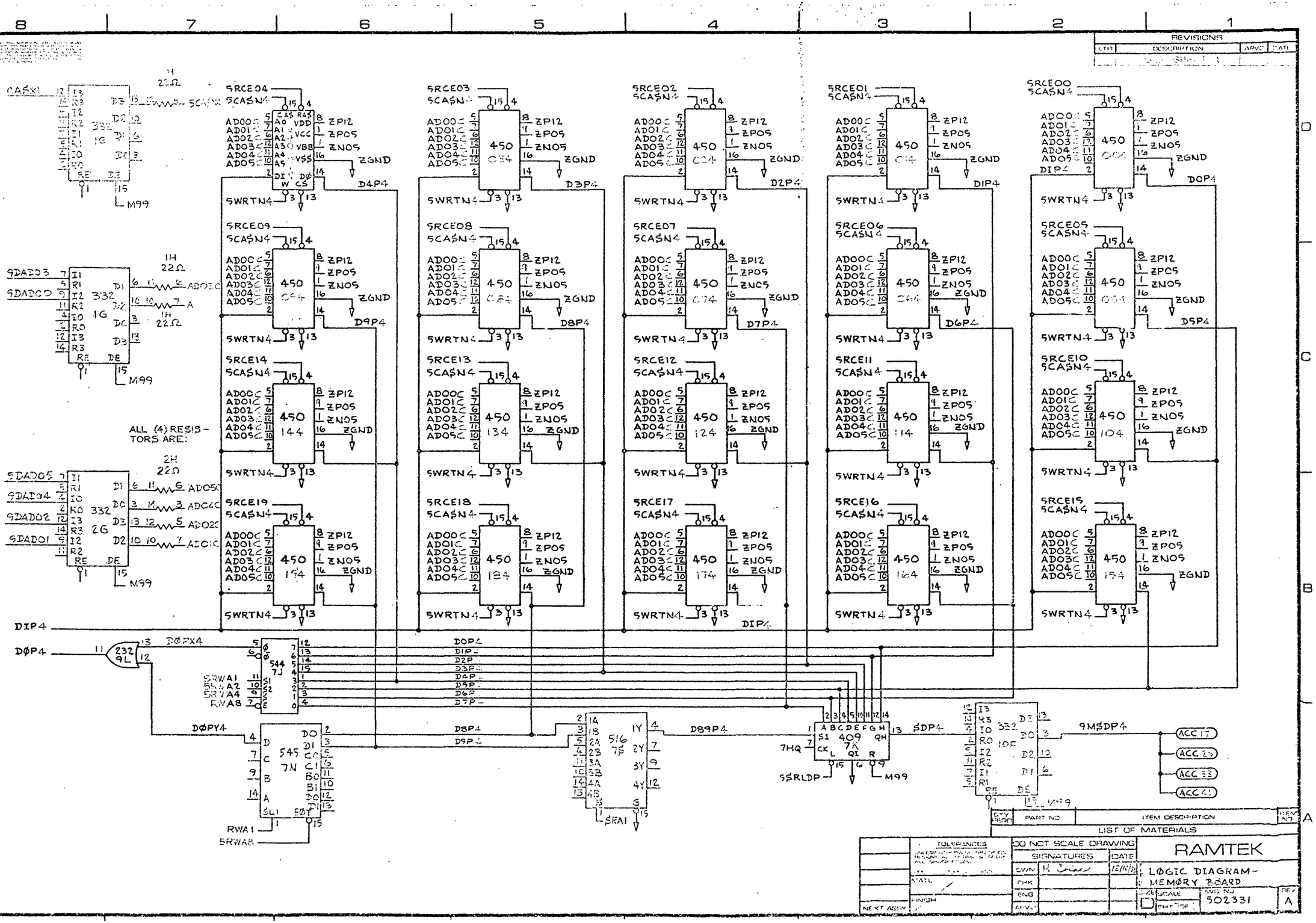
PIV-000539


REVISIONS
LTR
DESCRIPTION
APVD
DATE
SRCE04
SCA$N2
SRCE03
SRCE02
SRCE01
SRCE00
SRCE09
SRCE08
SRCE07
SRCE06
SRCE05
SRCE14
SRCE13
SRCE12
SRCE11
SRCE10
SRCE19
SRCE18
SRCE17
SRCE16
SRCE15
ADO0
ADO1
ADO2
ADO3
ADO4
ADO5
ZPI2
ZPOS
ZNOS
ZGND
SWRTN2
D4P2
D3P2
D2P2
D1P2
D9P2
D8P2
D7P2
D6P2
D5P2
D4P2
D3P2
D2P2
DIP2
DOP2
ALL (4) RESIS-
TORS ARE
4.7Ω
DIP2
D0P2
D0PY2
SRWA1
SRWA2
SRWA4
SRWA8
D8P2
D9P2
D89P2
7HQ
S$RLDP
M99
9MSDP2
ACC 14
ACC 2
ACC 3
ACC 5
RVA1
SRVA8
LIST OF MATERIALS
QTY REQ
PART NO
ITEM DESCRIPTION
REF NO
TOLERANCES
DO NOT SCALE DRAWING
SIGNATURES
DATE
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE
D
DWG NO
502331
REV
A
SHEET 1 OF 1

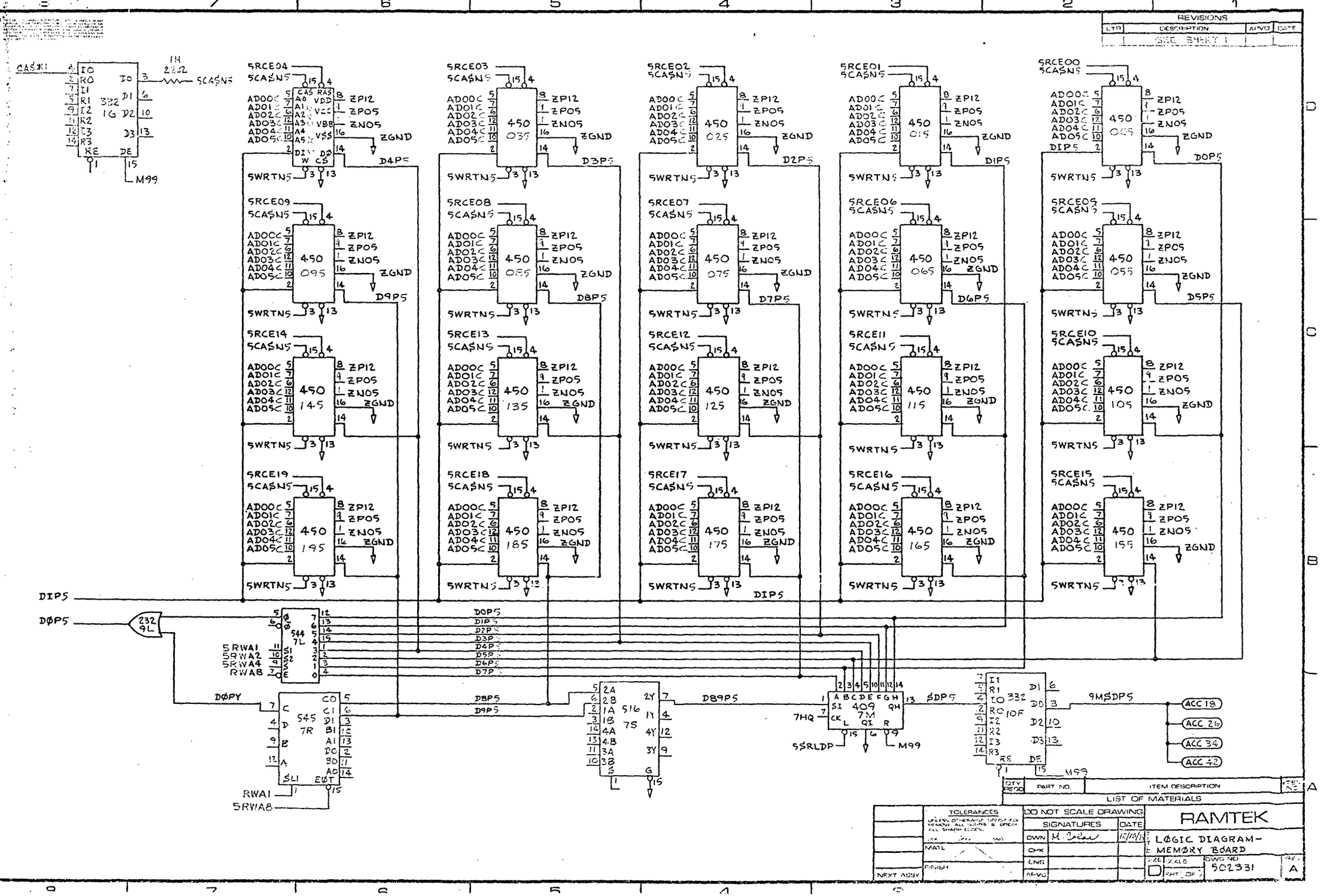
PIV-000540


REVISIONS
LTR DESCRIPTION APVD DATE
NOTICE
CA$X1
SCA$N3
M99
SRCE04
SRCE09
SRCE14
SRCE19
SRCE03
SRCE08
SRCE13
SRCE18
SRCE02
SRCE07
SRCE12
SRCE17
SRCE01
SRCE06
SRCE11
SRCE16
SRCE00
SRCE05
SRCE10
SRCE15
SCA$N
AD00B
AD01B
AD02
AD03
AD04
AD05
ZPI2
ZPOS
ZNOS
ZGND
SWRTN3
D4P3
D9P3
D3P3
D8P3
D2P3
D7P3
D1P3
D6P3
D0P3
D5P3
DIP3
D0P3
D0PX3
D0PY3
D1P3
D2P3
D3P3
D4P3
D5P3
D6P3
D7P3
D8P3
D9P3
SRWA1
SRWA2
SRWA4
RWA8
RWA1
5RWA8
$RAI
D89P3
$DP3
9M$DP3
7HQ
S$RLDP
ACC16
ACC24
ACC32
ACC40
ITEM NO
PART NO
ITEM DESCRIPTION
LIST OF MATERIALS
DO NOT SCALE DRAWING
TOLERANCES
SIGNATURES
DATE
DRAWN
CHK
ENG
APVD
MATL
FINISH
NEXT ASSY
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE
SCALE
DWG NO
502331
REV
A
SHT OF

PIV-000541


REVISIONS
LTR DESCRIPTION APVD DATE
NOTICE
CA$X1
SCA$N4
M99
SRCE04
SRCE03
SRCE02
SRCE01
SRCE00
SRCE09
SRCE08
SRCE07
SRCE06
SRCE05
SRCE14
SRCE13
SRCE12
SRCE11
SRCE10
SRCE19
SRCE18
SRCE17
SRCE16
SRCE15
450
ADO0C
ADO1C
ADO2C
ADO3C
ADO4C
ADO5C
ZPI2
ZPOS
ZNOS
ZGND
SWRTN4
D4P4
D3P4
D2P4
D1P4
D9P4
D8P4
D7P4
D6P4
D5P4
D14P4
D13P4
D12P4
D11P4
D10P4
D19P4
D18P4
D17P4
D16P4
D15P4
DIP4
DOP4
SDADO3
SDADO4
SDADO5
SDADO2
SDADO1
ADO3C
ADO4C
ADO5C
ADO2C
ADO1C
ALL (4) RESISTORS ARE:
1H 22Ω
2H 22Ω
332 1G
332 2G
D$PX4
D$PY4
544 7J
545 7N
516 7S
SRWA1
SRWA2
SRWA4
RWA8
RWA1
SRA1
D8P4
D9P4
D89P4
232 9L
409 7K
7HQ
$$RLDP
$DP4
9M$DP4
ACC17
ACC25
ACC33
ACC41
PART NO
ITEM DESCRIPTION
ITEM NO
LIST OF MATERIALS
TOLERANCES
DO NOT SCALE DRAWING
SIGNATURES
DATE
DWN
CHK
ENG
APVD
MATL
FINISH
NEXT ASSY
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE
SCALE
DWG NO
502331
REV
A

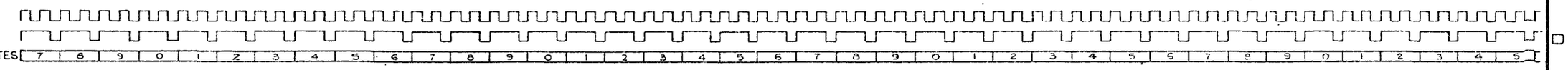
PIV-000542


REVISIONS
LTR DESCRIPTION APVD DATE
SEE SHEET 1
CAS*1
332 16
SCA$N5
SRCE04
SRCE09
SRCE14
SRCE19
SRCE03
SRCE08
SRCE13
SRCE18
SRCE02
SRCE07
SRCE12
SRCE17
SRCE01
SRCE06
SRCE11
SRCE16
SRCE00
SRCE05
SRCE10
SRCE15
450
ZPI2
ZPOS
ZNOS
ZGND
SWRTN5
D4P5
D9P5
D8P5
D3P5
D2P5
D7P5
D6P5
D1P5
D5P5
D0P5
DIP5
D0P5
232 9L
544 7L
5RWA1
5RWA2
5RWA4
RWA8
D0PY
545 7R
516 7S
D89P5
409 7M
7HQ
5$RLDP
$DP5
M99
9M$DP5
ACC 1B
ACC 2B
ACC 3B
ACC 4B
LIST OF MATERIALS
QTY REQD
PART NO.
ITEM DESCRIPTION
ITEM NO
TOLERANCES
DO NOT SCALE DRAWING
SIGNATURES
DATE
DWN
CHK
ENG
APVD
MATL
FINISH
NEXT ASSY
RAMTEK
LOGIC DIAGRAM-
MEMORY BOARD
SIZE
SCALE
DWG NO
502331
REV
A
SHT OF

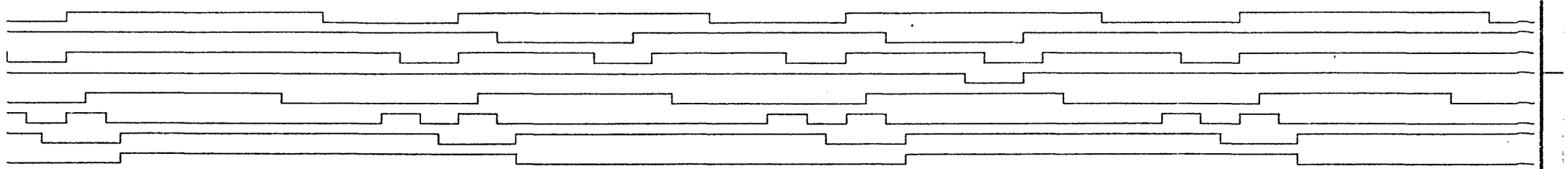
PIV-000543

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET 1.		

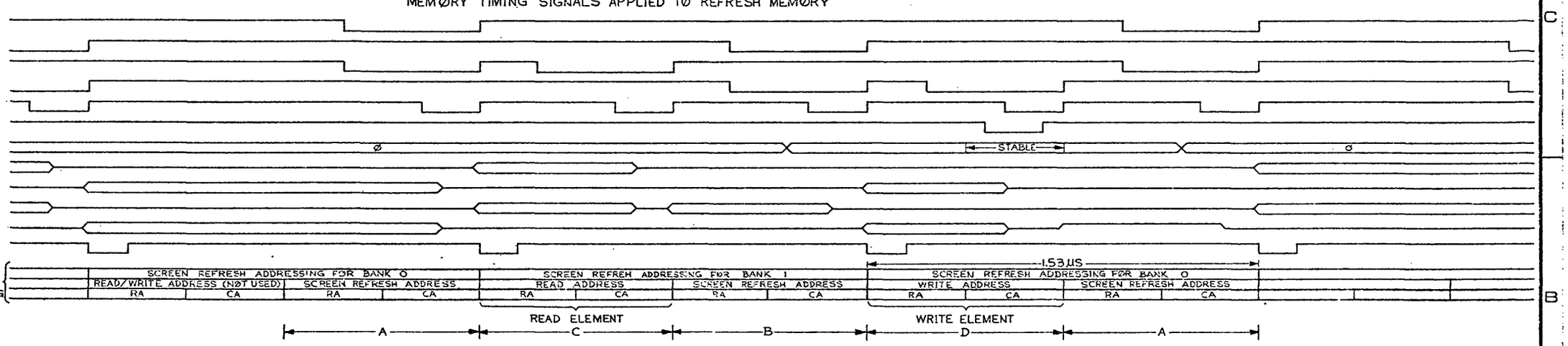
MEMØRY TIMING



MEMØRY TIMING SIGNALS AT THE CØNNECTØR



MEMØRY TIMING SIGNALS APPLIED TØ REFRESH MEMØRY



NØTES: 1. TIMING SHOWN FOR PLANE 0 ONLY.

2. FOUR PAIRS OF SIGNALS ARE SHOWN:
   - A. 5RCE06–D6PO
   - B. 5RCE13–D3PO
   - C. 5RCE02–D2PO
   - D. 5RCE15–D5PO

   EACH PAIR WERE SELECTED TO DEMONSTRATE THE FOLLOWING:
   - A. SCREEN REFRESH ONLY IN BANK 0 IN RELATION TO BANK 1.
   - B. SCREEN REFRESH ONLY IN BANK 1 IN RELATION TO BANK 0.
   - C. THE CONTROL CARD READING AN ELEMENT.
   - D. THE CONTROL CARD WRITING AN ELEMENT.

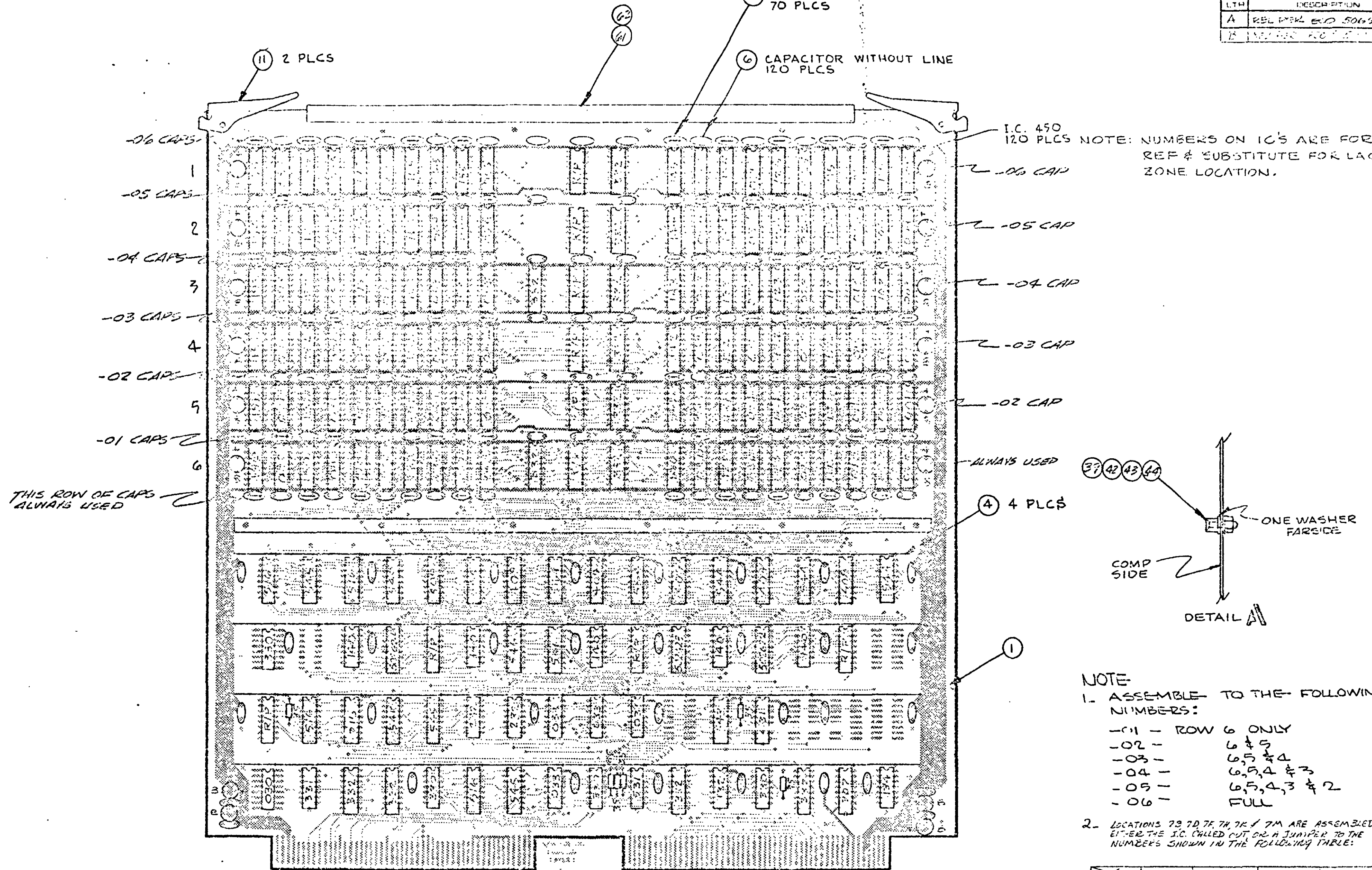
QTY REQD	PART NO	ITEM DESCRIPTION	ITEM NO

LIST OF MATERIALS

TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG	DO NOT SCALE DRAWING		RAMTEK
MATL	SIGNATURES	DATE	TIMING DIAGRAM MEMORY BOARD
FINISH	DWN F. LOO	12.4.73	
NEXT ASSY	CHK		SIZE D SCALE — SHT 9 OF 9 DWG NO 502331 REV A
	ENG		
	APVD		

PIV-000544

LTR	DESCRIPTION	APVD	DATE
A	REL PER ECO 506952	7-21-76	
B	[illegible]		



NOTE: NUMBERS ON IC'S ARE FOR MEMORY REF & SUBSTITUTE FOR LACK OF ZONE LOCATION.

NOTE:

1. ASSEMBLE TO THE FOLLOWING DASH NUMBERS:
   - -01 – ROW 6 ONLY
   - -02 – 6 & 5
   - -03 – 6,5 & 4
   - -04 – 6,5,4 & 3
   - -05 – 6,5,4,3 & 2
   - -06 – FULL
2. LOCATIONS 7B, 7D, 7F, 7H, 7K & 7M ARE ASSEMBLED WITH EITHER THE I.C. CALLED OUT, OR A JUMPER TO THE DASH NUMBERS SHOWN IN THE FOLLOWING TABLE:

NO. \ LOC.	7B	7D	7F	7H	7K	7M
-01	409	7D-13 TO 7D-06	7F-13 TO 7F-06	7H-13 TO 7H-06	7K-13 TO 7K-06	7M-13 TO 7M-06
-02	409	409	"	"	"	"
-03	409	409	409	"	"	"
-04	409	409	409	409	"	"
-05	409	409	409	409	409	"
-06	409	409	409	409	409	409

QTY REQD	PART NO	ITEM DESCRIPTION	ITEM NO

LIST OF MATERIALS

TOLERANCES: UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.

MATL

FINISH

502,344 NEXT ASSY

DO NOT SCALE DRAWING

SIGNATURES		DATE
DWN	M. Cohen	1-9-76
CHK		
ENG		
APVD		

RAMTEK

ASSEMBLY DRAWING - 320 x 256 MEMORY PC BD

SIZE D SCALE — DWG NO 502332 REV B

SHT 1 OF 1

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	RELEASED PER ECO 306867		5-11-76
B	REVISED PER ECO 306909		7-9-76
C	REVISED PER ECO 306985		8-4-76
D	REVISED PER ECO 307059	DWN	10/29/76
E	REV PER ECO # 307277	DWN	1-3-77
F	REV PER ECO # 307344	DWN	2-2-77

SHEET 2

SHEET 2

DASH #	CRYSTAL
BROADCAST -01	12.273 MHZ
STANDARD -02	13.068 MHZ
EUROPEAN -03	12.185 MHZ
STANDARD 50 HZ -04	10.910 MHZ

SYSTEM CLOCK GENERATION

[OPTIONAL VCO] → XTAL → BQ → ÷2 → MIX → HQ

BQ

PROCESSOR CLOCK GENERATION

12.273 MHZ → XTAL → TPBQ → ÷4 COUNTER → TPC_ → 1 OF 8 DECODE → TPD_ → JK PULSE SHAPER AND RECLOCK → TPQ0NE, TPQTWØ → ECL DRIVERS

ϕ_1 - 8080 PROCESSOR CLOCK
ϕ_2 - 8080 PROCESSOR CLOCK

POWER-ON RESET

RESET SWITCHES ON THE CONTROL CARD AND THE SYSTEM FRONT PANEL

98SWCLR → POWER-ON RESET & SWITCH CLEAR

98PWCLR
POWER-ON RESET

BQ, HQ → ECL DRIVERS

BQ - BASE FREQUENCY CLOCK
HQ - HALF FREQUENCY CLOCK

SHEET 2

SHEET 3

SYNC TIMING GENERATION

DASH #
BROADCAST -01
STANDARD -02
EUROPEAN -03
STANDARD 50 HZ -04

PRESET PROM (ADDRESS) ← VFIELD, VBLANK
INTERLACE ENABLE
OPTIONAL SPU INTERLACE ENABLE
TLL_

HQ

ZERO → ELEMENT COUNTER (DATA, LOAD, CLOCK) ← ELEMENT COUNT RESET
ELEMENT COUNTER: 9 BITS, COUNTS
EC_

1/2 LINE COUNT INCREMENT
TLCI

LINE COUNTER (DATA, LOAD, CLOCK, ENABLE, CARRY): 10 BITS, COUNTS ← LINE COUNT PRESET
TLC_ VBLANK VFIELD

ELEMENT PROMS (ADDRESS) ← EC_, DECODES ELEMENT COUNTS
HSYNC HBLANK HDRIVE HMEM HEQUAL HSERRA
RESET INCR

DASH #
BROADCAST -01
STANDARD -02
EUROPEAN -03
STANDARD 50 HZ -04

LINE PROM (ADDRESS) ← TLC_ VFIELD, DECODES LINE COUNTS
VSYNC VDRIVE VEQUAL VMEM
VBLANK VFIELD

VBLANK

SHEET 3

SHEET 4

HSYNC HBLANK HMEM HEQUAL HSERRA
VBLANK
VSYNC VEQUAL VMEM

HQ → COMPOSITE SYNC/BLANKING COMBINATORIAL (CLOCK) ← COMBINES HORIZONTAL AND VERTICAL TIMING
CSYNC CBLANK CMEM

HSYNC HBLANK HDRIVE HMEM
VSYNC VDRIVE
VBLANK VFIELD

RECLOCK AND DRIVE

HSYNC VSYNC CSYNC | HBLANK VBLANK CBLANK | H DRIVE V DRIVE | HMEM VMEM | VFIELD

SHEET 4

KEY TO TERM NAMES:

H ____	HORIZONTAL TIMING SIGNALS
V ____	VERTICAL TIMING SIGNALS
C ____	COMPOSITE TIMING SIGNALS
EC ___	ELEMENT [illegible]
LC ___	LINE COUNT

SYNC - SYNCHRONIZATION
BLANK - BLANKING
DRIVE - DRIVE
MEM - MEMORY SYNCHRONIZATION
FIELD - FIELD FLAG ("A" OR "B")
EQUAL - EQUALIZATION
SERRA - SERRATIONS

THE MULTI-RESOLUTION SYNC TIMING DASH NUMBER SPECIFIES:
1.) SYSTEM CLOCK CRYSTAL FREQUENCY
2.) PRESET PROM PROGRAM
3.) LINE PROM PROGRAM

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.
.XX ± .XXX ± ANG ±
MATL ~
FINISH ~
NEXT ASSY 502335

DO NOT SCALE DRAWING

SIGNATURES		DATE
DWN	[illegible]	10/29/76
CHK	[illegible]	11/1/76
ENG	[illegible]	11/1/76
APVD	[illegible]	11/1/76

RAMTEK

BLOCK DIAGRAM
MULTI-RESOLUTION SYNC TIMING

SIZE D | SCALE — | SHEET 1 OF 2 | DWG NO. 502334 | REV F

501003 REV: N/C

PIV-000546

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET ONE		

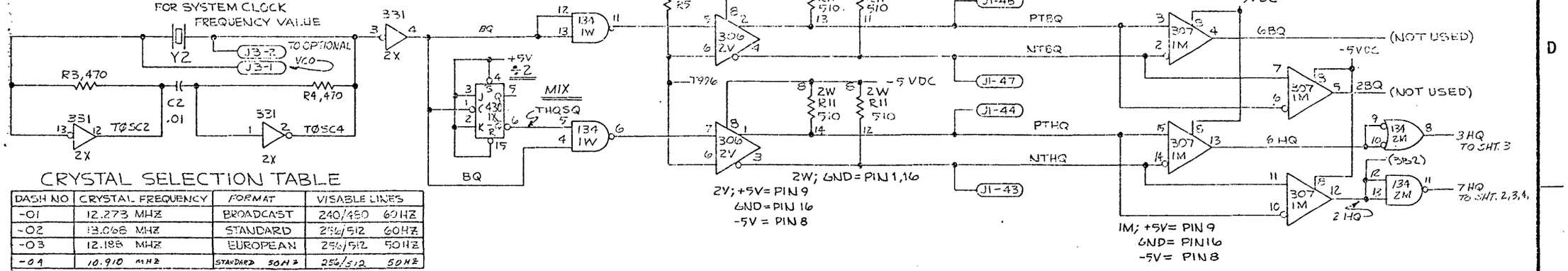
XTAL

SYSTEM CLOCK GENERATION

ECL DRIVERS

ECL RECEIVERS

SEE CRYSTAL SELECTION TABLE FOR SYSTEM CLOCK FREQUENCY VALUE



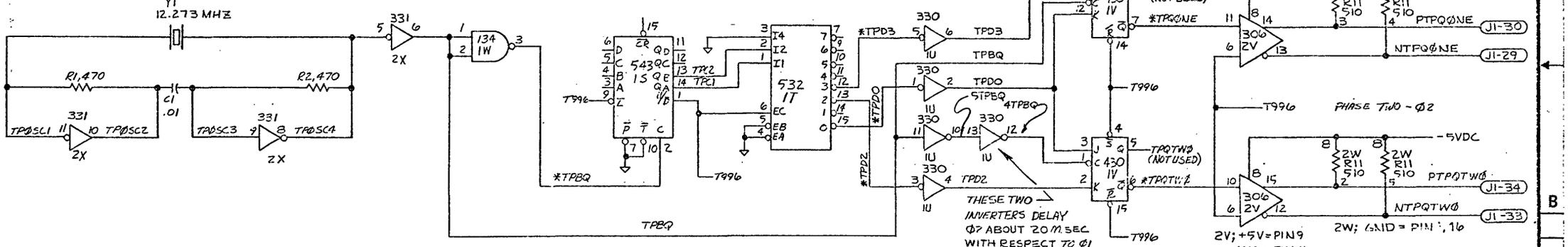
CRYSTAL SELECTION TABLE

DASH NO	CRYSTAL FREQUENCY	FORMAT	VISABLE LINES
-01	12.273 MHZ	BROADCAST	240/480 60HZ
-02	13.068 MHZ	STANDARD	256/512 60HZ
-03	12.188 MHZ	EUROPEAN	256/512 50HZ
-04	10.910 MHZ	STANDARD 50HZ	256/512 50HZ

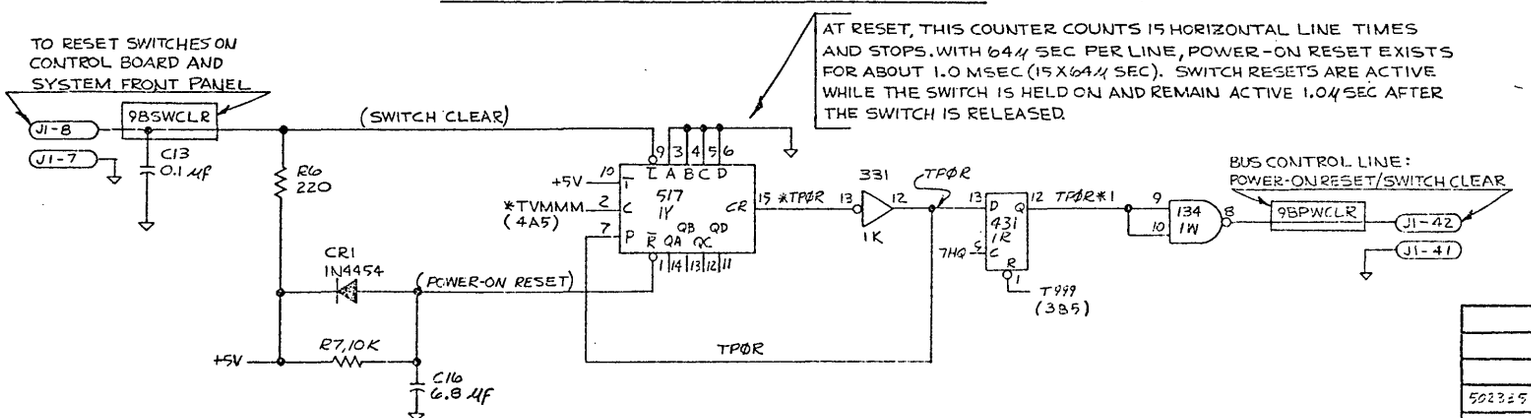
PROCESSOR CLOCK GENERATION

XTAL

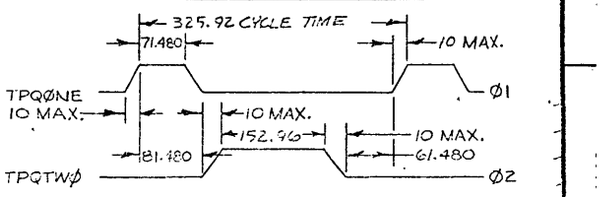
THE PROCESSOR CLOCK CRYSTAL FREQUENCY REMAINS THE SAME FOR ALL DASH NUMBERS.



POWER-ON RESET AND SWITCH CLEAR



8080A-1 CLOCKS



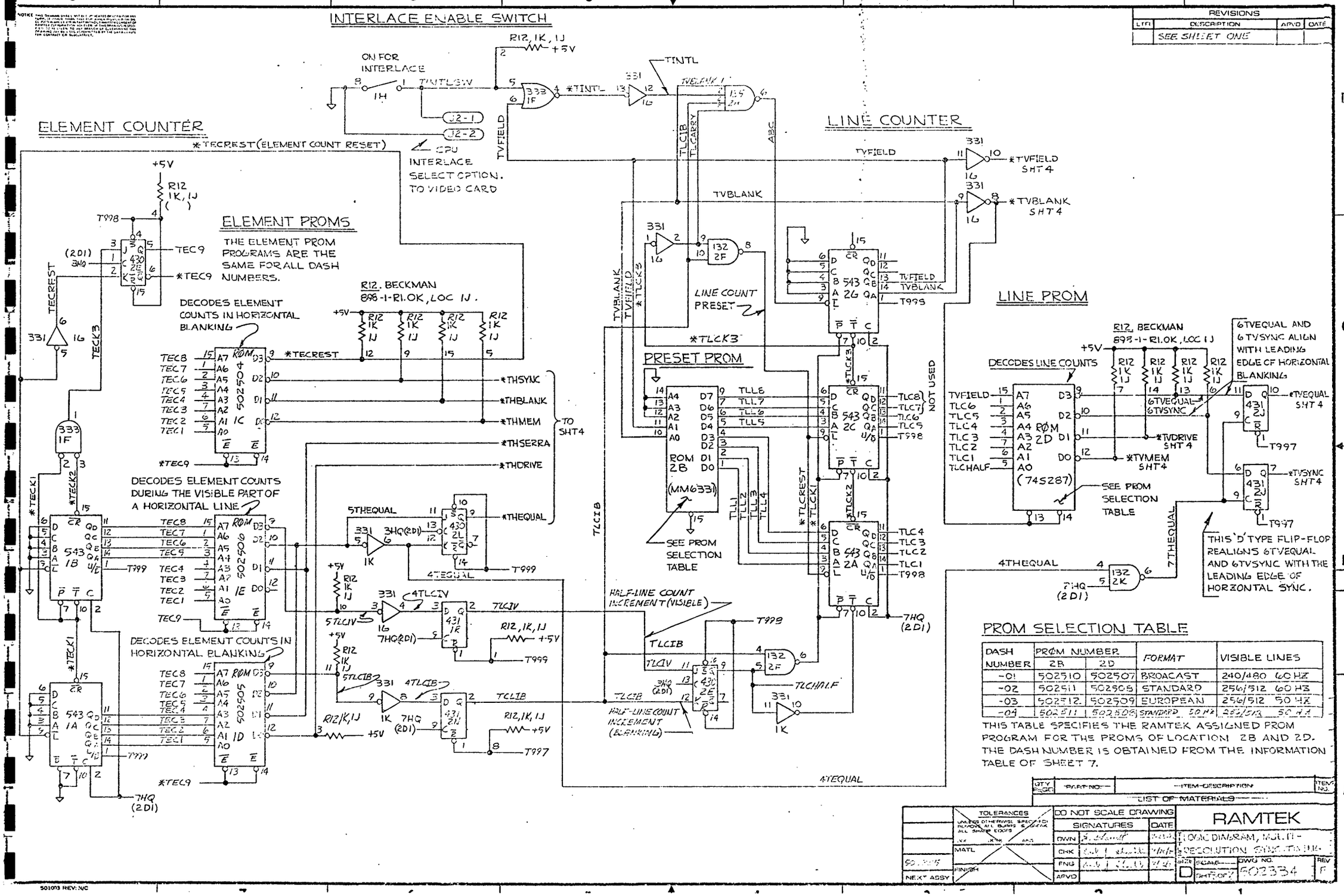
QTY PEGD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES	DO NOT SCALE DRAWING		RAMTEK
UNLESS OTHERWISE SPECIFIED ALL BURRS & BREAK ALL SHARP EDGES	SIGNATURES	DATE	
.XX .XXX ANG	DWN		LOGIC DIAGRAM, MULTI-RESOLUTION SYNC TIMING
MATL	CHK		
502335	ENG		SIZE D SCALE — DWG NO. 502334 REV F
FINISH	APVD		SHT 2 OF 7
NEXT ASSY			

501003 REV: N/C

PIV-000547


INTERLACE ENABLE SWITCH
ON FOR INTERLACE
1H
J2-1
J2-2
CPU INTERLACE SELECT OPTION. TO VIDEO CARD
R12, 1K, 1J
+5V
TINTL
*TINTL
TVELINK 1
TVFIELD
TLCIB
TLCARRY
ABS
ELEMENT COUNTER
*TECREST (ELEMENT COUNT RESET)
ELEMENT PROMS
THE ELEMENT PROM PROGRAMS ARE THE SAME FOR ALL DASH NUMBERS.
R12. BECKMAN 898-1-R1.0K, LOC 1J.
DECODES ELEMENT COUNTS IN HORIZONTAL BLANKING
DECODES ELEMENT COUNTS DURING THE VISIBLE PART OF A HORIZONTAL LINE
DECODES ELEMENT COUNTS IN HORIZONTAL BLANKING
TEC1
TEC2
TEC3
TEC4
TEC5
TEC6
TEC7
TEC8
TEC9
*TECREST
*THSYNC
*THBLANK
*THMEM
*THSERRA
*THDRIVE
*THEQUAL
TO SHT4
5THEQUAL
4THEQUAL
5TLCIV
4TLCIV
7HQ (2D1)
5TLCIB
4TLCIB
TLCIV
TLCIB
HALF-LINE COUNT INCREMENT (VISIBLE)
HALF-LINE COUNT INCREMENT (BLANKING)
TLCHALF
4TEQUAL
LINE COUNTER
TVFIELD
TVBLANK
*TVFIELD SHT4
*TVBLANK SHT4
LINE COUNT PRESET
*TLCK3
PRESET PROM
SEE PROM SELECTION TABLE
NOT USED
LINE PROM
DECODES LINE COUNTS
TLC1
TLC2
TLC3
TLC4
TLC5
TLC6
TLC7
TLC8
TLCHALF
6TVEQUAL
6TVSYNC
*TVDRIVE SHT4
*TVMEM SHT4
*TVEQUAL SHT4
*TVSYNC SHT4
7THEQUAL
THIS 'D' TYPE FLIP-FLOP REALIGNS 6TVEQUAL AND 6TVSYNC WITH THE LEADING EDGE OF HORIZONTAL SYNC.
6TVEQUAL AND 6TVSYNC ALIGN WITH LEADING EDGE OF HORIZONTAL BLANKING
PROM SELECTION TABLE
DASH NUMBER
PROM NUMBER
2B
2D
FORMAT
VISIBLE LINES
-01 502510 502507 BROADCAST 240/480 60 HZ
-02 502511 502508 STANDARD 256/512 60 HZ
-03 502512 502509 EUROPEAN 256/512 50 HZ
-04 502511 502508 STANDARD 256/512 50 HZ
THIS TABLE SPECIFIES THE RAMTEK ASSIGNED PROM PROGRAM FOR THE PROMS OF LOCATION 2B AND 2D. THE DASH NUMBER IS OBTAINED FROM THE INFORMATION TABLE OF SHEET 7.
REVISIONS
LTR DESCRIPTION APVD DATE
SEE SHEET ONE
LIST OF MATERIALS
QTY PER ASSY PART NO ITEM DESCRIPTION ITEM NO
DO NOT SCALE DRAWING
SIGNATURES DATE
DWN
CHK
ENG
APVD
RAMTEK
TIMING DIAGRAM, MULTI-RESOLUTION SYNC/TIMING
SIZE D SCALE DWG NO 502334 REV F
SHT 6 OF 7
TOLERANCES
MATL
FINISH
NEXT ASSY
501093 REV: N/C
PIV-000548

COMPOSITE SYNC/BLANKING COMBINATORIAL

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET ONE		

502335-02 STANDARD FORMAT TIMING SHOWN

RECLOCK

*THDRIVE (3A7) → 431 UNR → *THDDD
*TVDRIVE (3C2) → 431 UNR → *TVDDD
*THSYNC (3C6) → 431 UNR → *THSSS
*TVSYNC (3C1) → 431 UNR → *TVSSS
*TCSSS
*TVFIELD (3D2) → 431 UNR → *TVFFF
T997 (3A6)
7HQ (2D1)

DRIVERS

365 1P — HIGH CURRENT (NON-INVERTING) BUS DRIVERS

9THDDD	J1-22	HORIZONTAL DRIVE
#THDDD	J1-21	
9TVDDD	J1-36	VERTICAL DRIVE
#TVDDD	J1-37	
9THSSS	J1-24	HORIZONTAL SYNC
#THSSS	J1-23	
9TVSSS	J1-28	VERTICAL SYNC
#TVSSS	J1-27	
9TCSSS	J1-40	COMPOSITE SYNC
#TCSSS	J1-39	
9TVFFF	J1-18	FIELD FLAG
#TVFFF	J1-17	

59.7µS → 5.97µS
16.7MS → 597µS
59.7µS → 4.44µS
16.7MS → 179µS
16.7MS → COMPOSITE OF 9THSSS AND 9TVSSS WITH SERRATIONS AND EQUALIZATION PULSES
16.7MS → "1" IN REPEAT FIELD. TOGGLE IN INTERLACE: FIELD A = 1, FIELD B = 0

FOR HORIZONTAL SIGNALS, SYNC ON HORIZONTAL BLANKING.
FOR VERTICAL SIGNALS, SYNC ON VERTICAL BLANKING.

(3C6) *THSYNC
(3C5) T997
(3B5) *THEQUAL
(3B7) *THSERRA
(3C5) T997
(3C1) *TVEQUAL
(3C1) *TVSYNC
→ 534 1L → *RSYNC

COMBINES HORIZONTAL AND VERTICAL SYNC

(3C6) *THBLANK
(3D2) *TVBLANK
→ 132 2K → TCBLANK → 331 1K

COMBINES HORIZONTAL AND VERTICAL BLANKING

(3C5) *THMEM
(3C2) *TVMEM
→ 333 1F → TCMEM → 333 1F

COMBINES HORIZONTAL AND VERTICAL MEMORY SYNCHRONIZATION

*TCBLANK
*TCMEM

RECLOCK

*THBLANK (3C6) → 431 2N → *THBBB
*TVBLANK (3D3) → 431 2N → *TVBBB
*TCBBB
*THMEM (3C5) → 431 2N → *THMMM
*TVMEM → 431 2N → *TVMMM
T997 (3A6)
7HQ (2D1)

DRIVERS

365 2P — HIGH CURRENT (NON-INVERTING) BUS DRIVERS

9THBBB	J1-14	HORIZONTAL BLANKING
#THBBB	J1-13	
9TVBLB	J1-16	VERTICAL BLANKING
#TVBBB	J1-15	
9TCBBB	J1-20	COMPOSITE BLANKING
#TCBBB	J1-19	
9THMMM	J1-12	HORIZONTAL MEMORY SYNCHRONIZATION
#THMMM	J1-11	
9TVMMM	J1-10	VERTICAL MEMORY SYNCHRONIZATION
#TVMMM	J1-9	

TVMMM (2A6)

59.7µS → 10.7µS
16.7MS → 1.37MS FIELD 'A' 1.43 MS FIELD 'B'
16.7MS → COMPOSITE OF 9THBBB AND 9TVBBB
59.7µS → 0.153µS (WIDTH OF 7HQ)
16.7MS → 0.153µS (WIDTH OF 7HQ)

J1-1, J1-2, J1-3, J1-4 → C4 33µF, C5 THRU C12 .1µF → +5VDC
J1-31, J1-32, J1-35, J1-36, J1-45, J1-46, J1-49, J1-50
J1-25, J1-26 → C3 330µF, C13 .1µF, C14 .1µF → -5VDV

QTY REQD	PART-NO-	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES	DO NOT SCALE DRAWING		RAMTEK
UNLESS OTHERWISE SPECIFIED ROUND ALL HOLES & ARCS ALL SHARP EDGES. XX .XXX ANG	SIGNATURES	DATE	LOGIC DIAGRAM-MULTI-RESOLUTION SYNC TIMING
MATL	DWN		
FINISH	CHK		
NEXT ASSY	ENG		SIZE D SCALE SHEET 2 OF 2 DWG NO. 502334 REV F
	APVD		

PIV-000549

501003 REV: N/C

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET 1		

7BQ — SYSTEM BASE FREQUENCY CLOCK

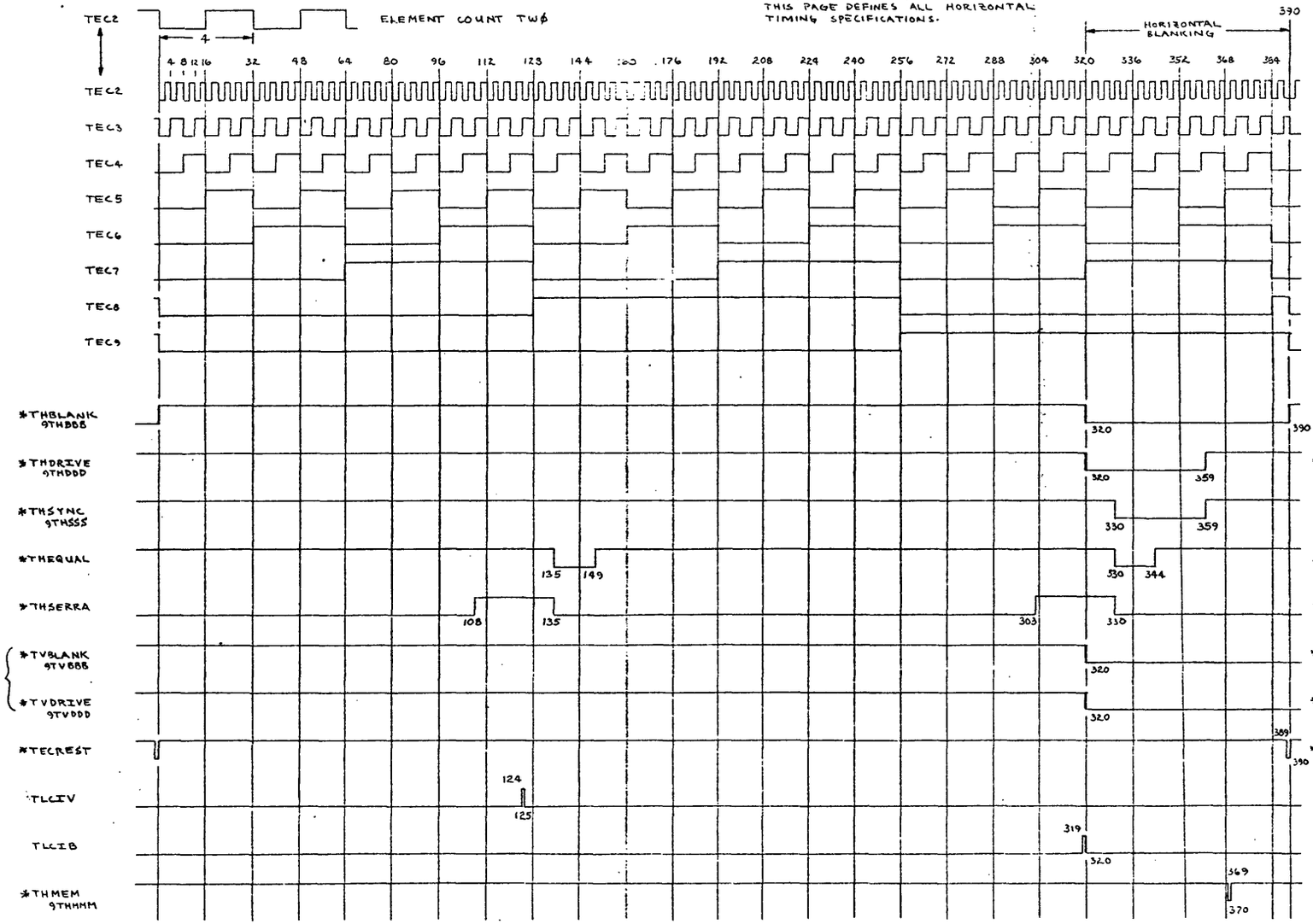
7HQ — SYSTEM HALF FREQUENCY CLOCK

TEC1 — ELEMENT COUNT ONE

TEC2 — ELEMENT COUNT TWO

NOTES:

1. ALL ELEMENT COUNTS SHOWN IN TERMS OF LOW RESOLUTION. ONE CYCLE OF THE SYSTEM HALF FREQUENCY CLOCK (7HQ) EQUALS ONE LOW RESOLUTION ELEMENT.
2. ALL 9100, 9200, AND 9300 SERIES IMAGER SYSTEMS ARE DEFINED WITH 320 VISIBLE ELEMENTS REGARDLESS OF THE NUMBER OF VISIBLE LINES. THEREFORE THIS PAGE DEFINES ALL HORIZONTAL TIMING SPECIFICATIONS.



VERTICAL BLANKING AND VERTICAL DRIVE ALIGN WITH THE LEADING EDGE OF HORIZONTAL BLANKING. COMPOSITE BLANKING ALWAYS ALIGNS WITH HORIZONTAL BLANKING.

ALL SIGNALS BELOW ARE PROM GENERATED FROM THE ELEMENT OR LINE COUNTERS.

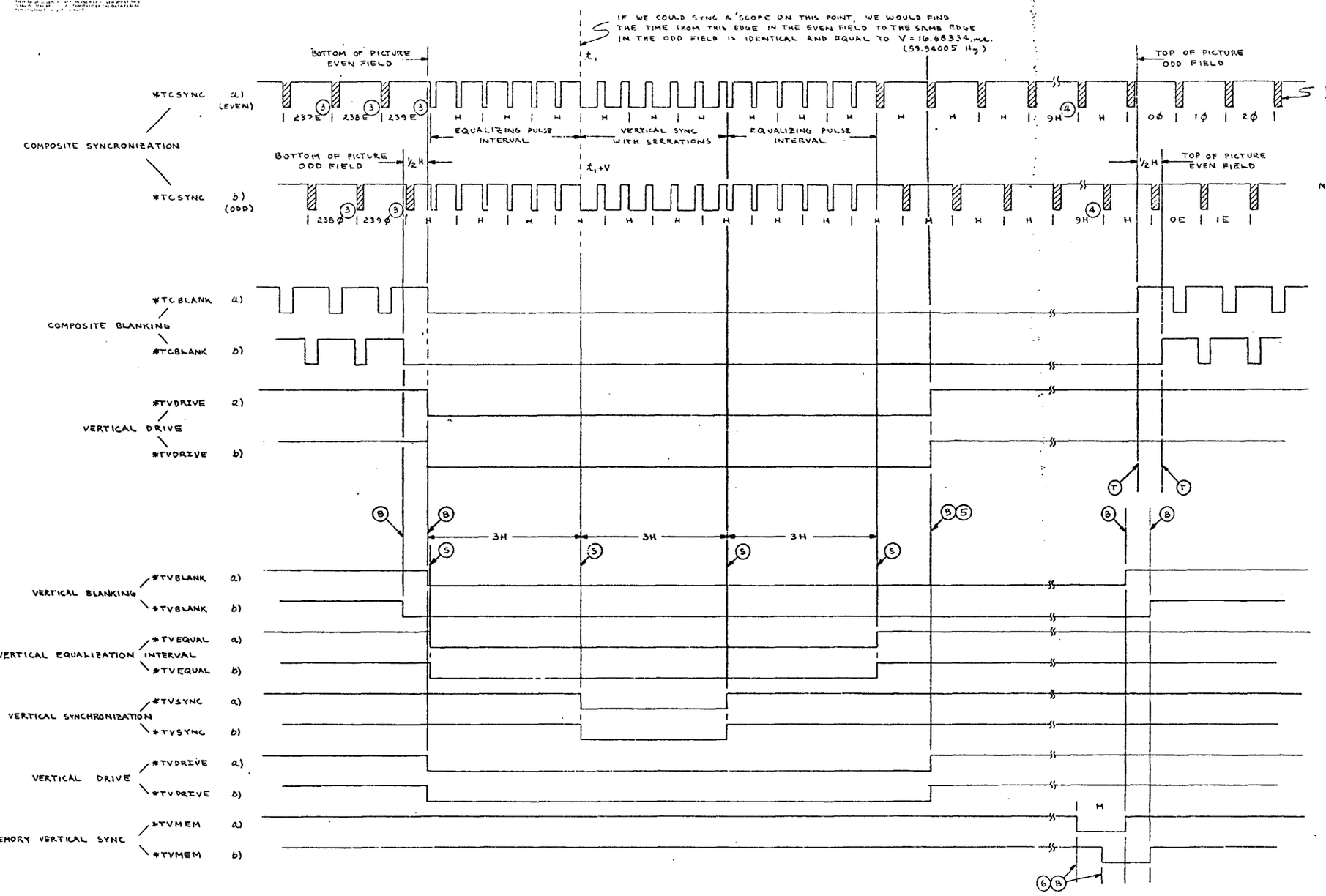
*THBLANK 9THBBB	HORIZONTAL BLANKING
*THDRIVE 9THDDD	HORIZONTAL DRIVE
*THSYNC 9THSSS	HORIZONTAL SYNCRONIZATION
*THEQUAL	HORIZONTAL EQUALIZATION
*THSERRA	HORIZONTAL SERRATION
*TVBLANK 9TVBBB	VERTICAL BLANKING
*TVDRIVE 9TVDDD	VERTICAL DRIVE
*TECREST	ELEMENT COUNTER RESET
TLCIV	LINE COUNTER INCREMENT (VISIBLE)
TLCIB	LINE COUNTER INCREMENT (BLANKING)
*THMEM 9THMMM	MEMORY HORIZONTAL SYNCRONIZATION

QTY PER	PART NO.	ITEM DESCRIPTION	ITEM NO.
LIST OF MATERIALS			

TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & SHARP ALL SHARP EDGES .XX ±.02 .XXX ±.005 ANG ±1°	DO NOT SCALE DRAWING		RAMTEK		
	SIGNATURES	DATE	HORIZONTAL TIMING — MULT-RESOLUTION SYNC TIMING		
MATL ~	DWN	1/9/76			
502335 FINISH ~	CHK	2/12/76			
NEXT ASSY	ENG	1/9/76	SIZE D	SCALE NONE	DWG NO. 502334 REV F
	APVD			SHEET 5 OF 7	

PIV-000550

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET 1		

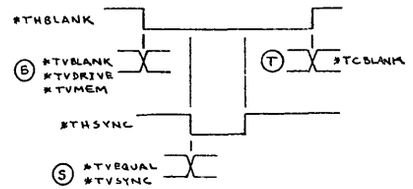


NOTES:

1. H = TIME FROM START OF ONE LINE TO START OF NEXT.
2. V = TIME FROM START OF ONE FIELD TO START OF NEXT.
3. THE ACTUAL LINE COUNT VARIES WITH THE TOTAL NUMBER OF VISIBLE LINES. SHOWN IS A SYSTEM CONFIGURED FOR 480 TOTAL VISIBLE LINES PER FRAME. (240 PER FIELD)
4. THE ACTUAL NUMBER OF LINES IN VERTICAL BLANKING VARIES WITH THE TOTAL NUMBER OF LINES PER FRAME. SHOWN IS A SYSTEM CONFIGURED FOR 480 VISIBLE LINES PER FRAME AND A TOTAL OF 525 LINES PER FRAME. SEE RASTER SCAN INFO CHART. (SHEET 7).

TOTAL LINES	= VISIBLE EVEN	+ BLANKED EVEN	+ VISIBLE ODD	+ BLANKED ODD
525	240	22	240	23
559	256	23	256	24
625	256	56	256	57

(B) SIGNAL ALIGNS WITH LEADING EDGE OF HORIZONTAL BLANKING. SEE DETAIL BELOW.
(S) SIGNAL ALIGNS WITH LEADING EDGE OF HORIZONTAL SYNCHRONIZATION. SEE BELOW.
(T) SIGNAL ALIGNS WITH TRAILING EDGE OF HORIZONTAL BLANKING. COMPOSITE BLANKING IS AN "OR" FUNCTION OF VERTICAL AND HORIZONTAL BLANKING.
(5) ACTUAL PULSE LENGTH VARIES WITH CONFIGURATION. REFER TO RASTER SCAN INFORMATION CHART. (SHEET 7).
(6) THIS PULSE OCCURS ON THE LAST LINE WITHIN VERTICAL BLANKING REGARDLESS OF CONFIGURATION OR FIELD.
7. FOR REPEAT FIELD OPERATION, ALL SIGNALS CORRESPOND TO THOSE SHOWN IN THE EVEN FIELD. (e.g. PULSE TRAINS a.)). ODD FIELD SIGNALS DO NOT EXIST.



525 LINE
2:1 INTERLACE SCAN

SEE NOTES ABOVE FOR REPEAT FIELD OPERATION AND FOR VARIOUS CONFIGURATIONS

QTY PER DEC	PART NO	ITEM DESCRIPTION	REV NO

LIST OF MATERIALS

	TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES	DO NOT SCALE DRAWING		RAMTEK	
		SIGNATURES	DATE		
	MATL N	DWN		1/10/1	VERTICAL TIMING -
		CHK		2/12/74	MULTI-RESOLUTION SYNC TIMING
502335	FINISH N	ENG		1/10/72	SIZE D SCALE NONE DWG NO 502334 REV F
NEXT ASSY		APVD			SHT 6 OF 7

PIV-000551

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET ONE		

MODEL	CONFIGURATION: VISIBLE LINES x ELEMENTS	FORMAT KEYWORD	FRAME REFRESH RATE * (Hz)	FIELD REFRESH RATE * (Hz)	RECOMMENDED POWER LINE FREQUENCY (Hz)	TOTAL LINES PER FRAME	TOTAL LINES FIELD "A"	TOTAL LINES FIELD "B"	VISIBLE LINES PER FRAME	VISIBLE LINES FIELD "A"	VISIBLE LINES FIELD "B"	BLANKED LINES PER FRAME	BLANKED LINES FIELD "A"	BLANKED LINES FIELD "B"	TOTAL ELEMENTS PER LINE	VISIBLE ELEMENTS PER LINE	BLANKED ELEMENTS PER LINE	CRYSTAL FREQUENCY (MHz)	PIXEL TIME (nSEC)	EXTERNAL SYNC-LOCK (VCO) AVAILABLE	MULTI-RESOLUTION SYNC TIMING CARD DASH NUMBER %
RM9100	240 x 320	R BROADCAST	60	60	60	262	262	-	240	240	-	22	22	-	390	320	70	12.273	162.96	-	502335 - 01
	240 x 320	I BROADCAST	30	60	60	525	262	263	480	240	240	45	22	23	390	320	70	12.273	162.96	YES	- 01
	256 x 320	R STANDARD	60	60	60	279	279	-	256	256	-	23	23	-	390	320	70	13.068	153.05	-	502335 - 02
	256 x 320	I STANDARD	30	60	60	559	279	280	512	256	256	47	23	24	390	320	70	13.068	153.05	-	- 02
	256 x 320	R EUROPEAN	50	50	50	312	312	-	256	256	-	56	56	-	390	320	70	12.188	164.10	-	502335 - 03
	256 x 320	I EUROPEAN	25	50	50	625	312	313	512	256	256	113	56	57	390	320	70	12.188	164.10	-	- 03
	256 x 320	R STANDARD 50HZ	50	50	50	279	279	—	256	256	-	23	23	-	390	320	70	10.910	183.32	-	502335 - 04
	256 x 320	I STANDARD 50HZ	25	50	50	559	279	280	512	256	256	47	23	24	390	320	70	10.910	183.32	—	- 04
RM9200	240 x 640	R BROADCAST	60	60	60	262	262	-	240	240	-	22	22	-	780	640	140	12.273	81.481	-	502335 - 01
	240 x 640	I BROADCAST	30	60	60	525	262	263	480	240	240	45	22	23	780	640	140	12.273	81.481	YES	- 01
	256 x 640	R STANDARD	60	60	60	279	279	-	256	256	-	23	23	-	780	640	140	13.068	76.526	-	502335 - 02
	256 x 640	I STANDARD	30	60	60	559	279	280	512	256	256	47	23	24	780	640	140	13.068	76.526	-	- 02
	256 x 640	R EUROPEAN	50	50	50	312	312	-	256	256	-	56	56	-	780	640	140	12.188	82.051	-	302335 - 03
	256 x 640	I EUROPEAN	25	50	50	625	312	313	512	256	256	113	56	57	780	640	140	12.188	82.051	-	- 03
	256 x 640	R STANDARD 50HZ	50	50	50	279	279	—	256	256	-	23	23	-	780	640	140	10.910	91.654		502335 - 04
	256 x 640	I STANDARD 50HZ	25	50	50	559	279	280	512	256	256	47	23	24	780	640	140	10.910	91.654		- 04
RM9300	480 x 640	I BROADCAST	30	60	60	525	262	263	480	240	240	45	22	23	780	640	140	12.273	81.481	YES	502335 - 01
	512 x 640	I STANDARD	30	60	60	559	279	280	512	256	256	47	23	24	780	640	140	13.068	76.526	-	502335 - 02
	512 x 640	I EUROPEAN	25	50	50	625	312	313	512	256	256	113	56	57	780	640	140	12.188	82.051	YES	502335 - 03
	512 x 640	I STANDARD 50HZ	25	50	50	559	279	280	512	256	256	47	23	24	780	640	140	10.910	91.654	—	502335 - 04

LEGEND: R = REPEAT FIELD. FIELD "A" IS IDENTICAL TO FIELD "B".
I = 2:1 INTERLACED SCAN.

% THE DASH NUMBER SPECIFIES:
1) SYSTEM CLOCK CRYSTAL FREQUENCY
2) TWO LINE COUNT PROM PROGRAMS

* AN INDICATED 60 Hz IN THIS COLUMN ACTUALLY IS 59.940 Hz
" " 50 Hz " " " 50.000 Hz
" " 30 Hz " " " 29.970 Hz
" " 25 Hz " " " 25.000 Hz

EXEPTIONS: STANDARD 50HZ FORMAT:
AN INDICATED 50 Hz IN THIS COLUMN ACTUALLY IS 50.044 Hz
AN INDICATED 25 Hz IN THIS COLUMN ACTUALLY IS 25.022 Hz

IMPORTANT NOTE:

IN AN INTERLACED SYSTEM ONE FRAME = TWO FIELDS.
THAT IS, THERE ARE TWO FIELDS PER FRAME.

IN A REPEAT FIELD SYSTEM ONE FRAME = ONE FIELD.
THAT IS, THERE IS ONE FIELD PER FRAME.
ONE FRAME-TIME = A FIELD TIME.

QTY PER	PART NO.	ITEM DESCRIPTION	ITEM NO

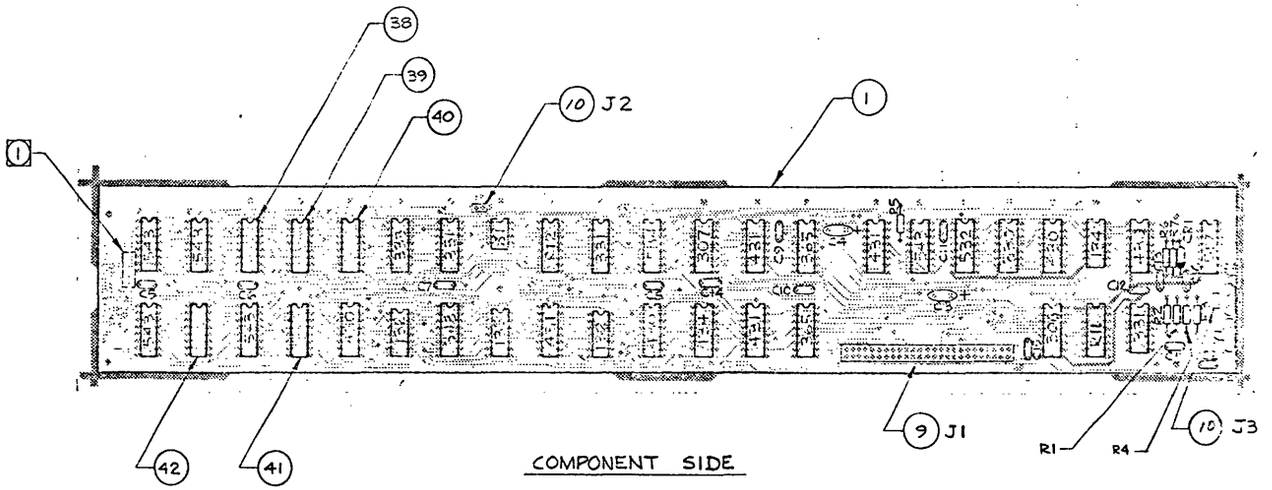
LIST OF MATERIALS

	TOLERANCES	DO NOT SCALE DRAWING	RAMTEK
	UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX ~ .XXX ~ ANG ~	SIGNATURES DATE	RASTER SCAN INFORMATION
	MATL ~	DWN	MULTI-RESOLUTION SYNC TIMING
502335	FINISH ~	CHK	
NEXT ASSY		ENG	SIZE D SCALE --- DWG NO. 502334 REV F
		APVD	SHT 2 OF 2

PIV-000552

501003 REV: N/C

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	RELEASED PER ECO 506565		
B	REVISED PER ECO 506709	[illegible]	7-7-76
C	REVISED PER ECO 506925	[illegible]	8/9/76
D	REVISED PER ECO 507059	DWG	10/5/76
E	REVISED PER ECO 507304	WTA	2-8-77



COMPONENT SIDE

-01, -02 & -03 SHOWN
& -04

NOTES: UNLESS OTHERWISE SPECIFIED

1. INK STAMP SERIAL NUMBER AND ASSY REVISION LEVEL IN AREA SHOWN.
2. REFERENCE DRAWINGS-
   LOGIC DIAGRAM : 502334
   FABRICATION DWG: 502333C
3. LOGIC DEVICE TYPE NUMBERS ARE RAMTEK STANDARD PART NUMBERS 1301XXX.

QTY PER ASSY	PART NO.	ITEM DESCRIPTION	ITEM NO.

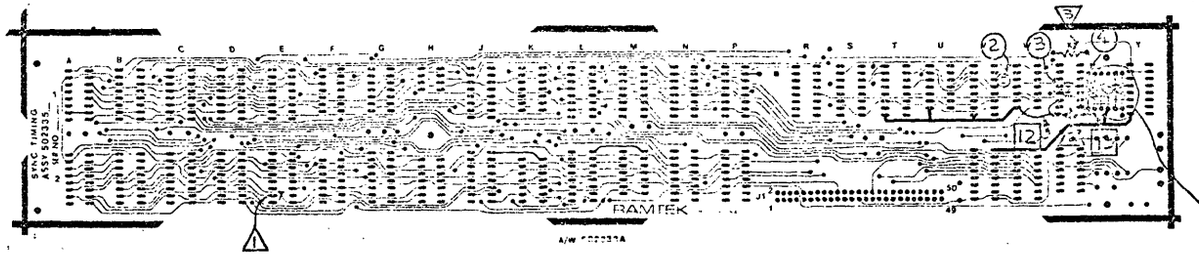
LIST OF MATERIALS

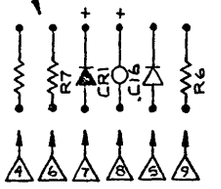
TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX = ±.010 .XXX = ±.005 ANG = ±1°	DO NOT SCALE DRAWING		
	SIGNATURES	DATE	RAMTEK
MATL	DWN	6/76	P.C. ASSEMBLY -01,-02&-03
	CHK	6/76	MULTIRESOLUTION SYNC TIMING
FINISH	ENG	6/76	SIZE D SCALE 1:1 DWG NO. 502335 REV E
NEXT ASSY	APVD		SHT 1 OF 3

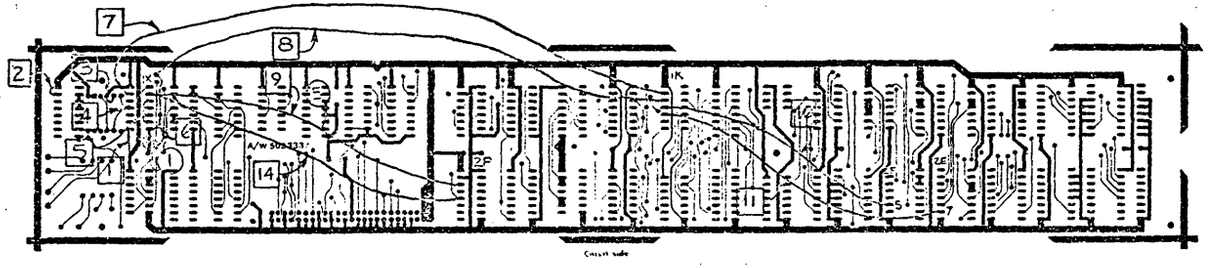
PIV-000553

501003 REV: N/C

REVISIONS			
LTR	DESCRIPTION	APVD	DATE







THIS CUT AND PATCH DRAWING APPLIES TO REVISION 'A' ARTWORK BOARDS ONLY

REVISION 'B'

REVISION 'B' IS FUNCTIONALLY EQUIVALENT TO REVISION 'C'

CUT P.C. TRACES AS INDICATED TO DELETE THE FOLLOWING CONNECTIONS:

(1) IX-07 TO FEED THRU
(2) IV-14 TO IY-03
(3) IX-10 TO FEED THRU
(4) IX-12 TO COMPONENT

ADD THE INDICATED WIRES TO MAKE THE FOLLOWING CONNECTIONS:

[1] IX-08 TO FEED THRU
[2] IY-15 TO IY-16
[3] COMPONENT TO COMPONENT
[4] IX-01 TO COMPONENT
[5] TIE TOGETHER IX-03, IX-04, IX-05, IX-06
[6] IX-07 TO FEEDTHRU
[7] IX-13 TO IK-13
[8] IK-12 TO FEEDTHRU
[9] IX-02 TO ZP-14
[10] IK-11 TO ZF-05
[11] IK-10 TO ZE-07
[12] TIE TOGETHER IW-07 TO BENT I.C. PIN AT IX AND TO IX-06
[13] BENT PIN AT IX TO COMPONENT HOLE

MAKE COMPONENT ADDS AND DELETES AND ADDITIONAL CHANGES AS NOTED:

△1 CUT PIN 7 OF THE I.C. AT LOCATION 2E. DO NOT CUT THE TRACE.
△2 REMOVE THE I.C. AT LOCATION IX. INSTALL I.C. 1301517 (SN74161) AT LOCATION IX. THIS REQUIRES BENDING PINS 8 AND 9 OF THE I.C. AS INDICATED.
△3 REMOVE RESISTOR
△4 REMOVE RESISTOR
△5 REMOVE DIODE
△6 REPLACE EXISTING COMPONENT WITH 10K, 1/4W, 5% RESISTOR. R7 (3109103)
△7 REPLACE EXISTING COMPONENT WITH IN4454 DIODE. CR1 (3202201) NOTE POLARITY.
△8 REPLACE EXISTING COMPONENT WITH 6.8 μF, 15V, 20% CAPACITOR. C16 (6706685) NOTE POLARITY
△9 REPLACE EXISTING COMPONENT WITH 220Ω, 1/4W, 5% RESISTOR. R6 (3109221)

REVISION 'C'

REVISION 'C' IS FUNCTIONALLY EQUIVALENT TO REVISION 'B' ABOVE

REVISION 'D'

REMOVE THE INDICATED WIRE TO DELETE THE FOLLOWING CONNECTION

(5) IX-02 TO ZP-14

ADD THE INDICATED WIRES TO MAKE THE FOLLOWING CONNECTION

[14] IX-02 TO ZP-12

QTY PER	PART NO	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

	TOLERANCES	DO NOT SCALE DRAWING			RAMTEK
	UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.	SIGNATURES		DATE	
	.XX .XXX ANG	DWN	Ernest J. Hamilton	10/1/76	P.C. ASSEMBLY-01-02403
	MATL	CHK	Ernest J. Hamilton	10/1/76	MULTI-RESOLUTION SYNC TIMING
		ENG	Ernest J. Hamilton	10/1/76	SCALE / DWG NO 502335 REV E
NEXT ASSY	FINISH	APVD	S. Cooper	10/1/76	D SHT 2 OF 3

PIV-000554

THIS CUT AND PATCH DIAGRAM APPLIES TO REVISION 'C' ARTWORK ONLY.

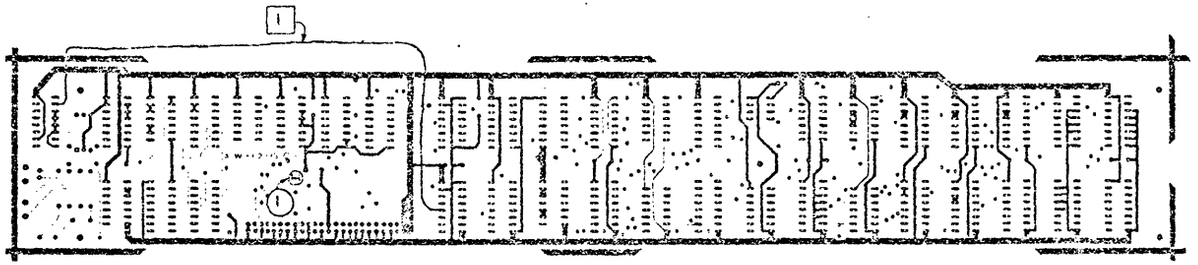
REVISION 'D'

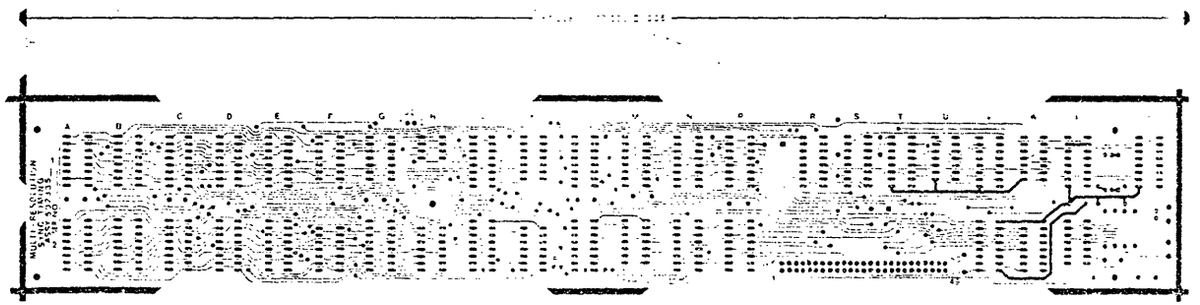
CUT P.C. TRACES AS INDICATED TO DELETE THE FOLLOWING CONNECTIONS:

(1.) 1Y-02 TO 2P-14

ADD THE INDICATED WIRES TO MAKE THE FOLLOWING CONNECTIONS:

[1.] 1Y-02 TO 2P-12

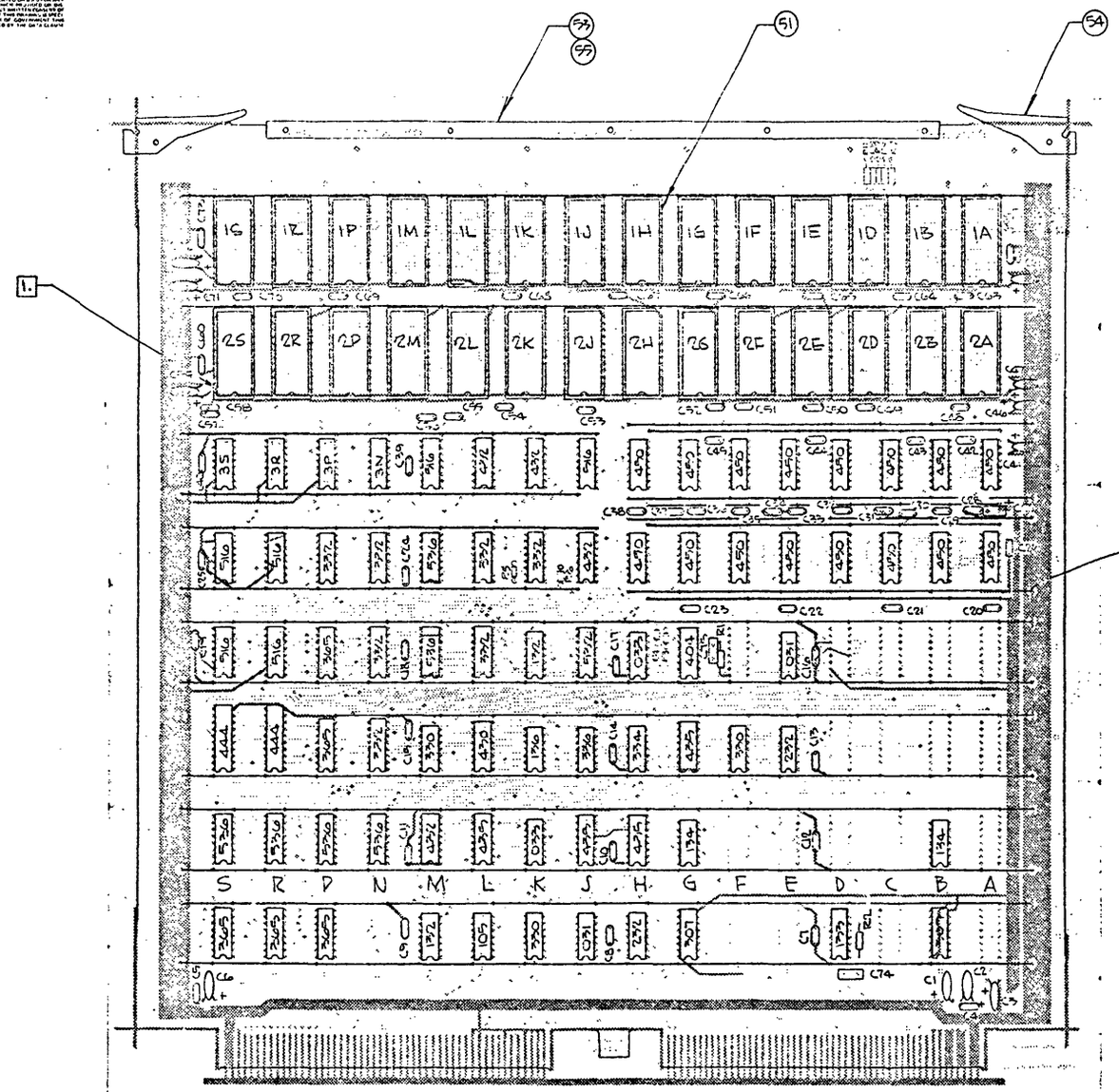




DO NOT SCALE DRAWING			RAMTEK
SIGNATURES		DATE	
DWN			PC ASSEMBLY HOI-0241-03
CHK			HI-RESOLUTION SYNC TIMING
ENG			SIZE SCALE DWG NO REV
APVD			D 502335 E

PIV-000555

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	RELEASED PER ECO 507013	DMS	9-7-76
B	ECO # 507052	DM	5/4/76
C	REV PER ECO # 507124	[illegible]	11-26-76
D	REV PER ECO # 507155	[illegible]	11-15-76
E	REV PER ECO # 507213	[illegible]	2-3-77
F	REV PER ECO # 507339	[illegible]	5/18/77



D	C	507155	NONE REQ'D
C	B	507124	CUT PADS AT 3R-16 & 3P-16 TO PREVENT SHORTS
B	B	507052	—
A	A	507013	—
ASSY REV	A/W REV	ECO #	WIRES & CUTS REQ'D

FAB 502441

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES
UNLESS OTHERWISE SPECIFIED: REMOVE ALL BURRS & BREAK ALL SHARP EDGES.
.XX .XXX ANG
MATL
FINISH
NEXT ASSY

DO NOT SCALE DRAWING

SIGNATURES		DATE
DWN	EIMICA	11-26-76
CHK	[illegible]	9/3/76
ENG	[illegible]	[illegible]
APVD	[illegible]	9-8-76

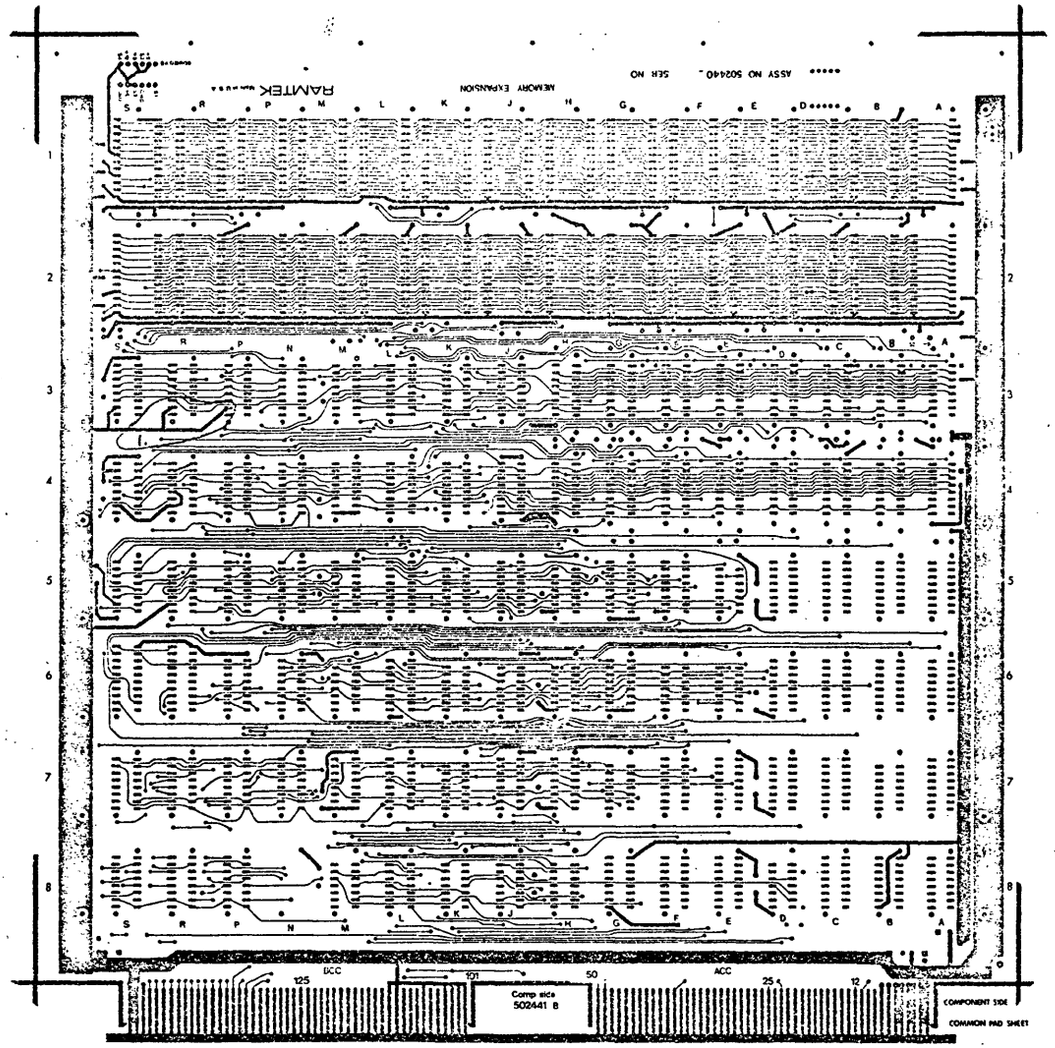
RAMTEK

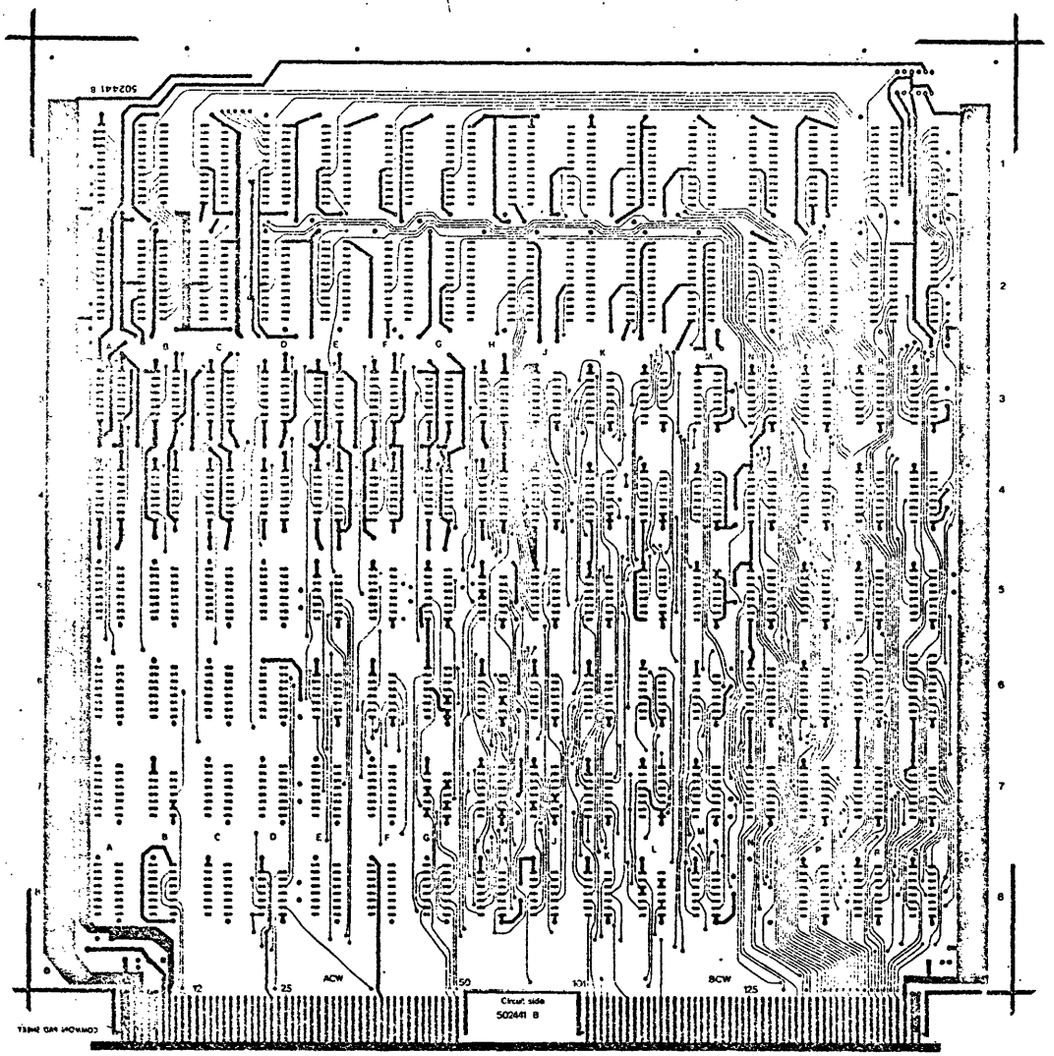
ASSEMBLY, MEMORY EXPANSION

SIZE D | SCALE 2/1 | DWG NO. 502440 | REV F | SHT 1 OF 2

501003 REV: N/C

PIV-000556





10.

1. SHAVE PADS AT 3R-16 & 3P-16 TO REV 'B' BD'S ONLY.

DO NOT SCALE DRAWING			RAMTEK
SIGNATURES		DATE	
DWN			ASSEMBLY, MEMORY EXPANSION
CHK			
ENG			SIZE D · SCALE 1/1 · SHT 2 OF 2 · DWG NO 502440 · REV F
APVD			

PIV-000557

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	ENG'S REL PROTOTYPES	DWG	9-9-76
B	ECO# P07052	WG	10/8/76
C	ECO# P07124	WG	11-26-76

ADDRESS BUS – 16 BITS — 4BADDRO–15

DATA BUS – 8 BITS — 9BDATAO–15

ADDRESS DECODE — ROM SELECT

OPTION PROM MEMORY 14K BYTES (28 ROM SOCKETS)

*RØMO–7

9BDAT8–15

ADDRESS LATCHES

ADDRESS–16 BITS

BADDRO–15

9BDAT8–15

9BREAD
9BWRITE
9BSYNC
9BDMAGO
9BPWCLR
9ICPCLR
9BACK
9BREADY
Ø1, Ø2
7HQ

RAM CONTROL LOGIC AND DMA CONTROL LOGIC

WE
RAS
CAS
ROW/COL
REFRESH
8080 CYCLE
DMA SOURCE
DMA DESTINATION

RAM REFRESH COUNTER 6 BITS.

RCNTO–5

REFRESH/ 8080 CYCLE ADDRESS SELECT

ROW/ COLUMN SELECT

ADRO–5

DATA IN — DYNAMIC RAM HIGH BYTE — DATA OUT

ADDR

RDAT8–15

DATA IN — DYNAMIC RAM LOW BYTE — DATA OUT

ADDR

RDATO–7

RAM OUTPUT DATA REGISTER

IRDAT8–15

IRDATO–7

RAM OUTPUT DATA HI/LOW BYTE SELECT

DT8–15

16 BITS DATA

BDATO–7

INCAC
Ø1

DMA ADDRESS COUNTER – 16 BITS

RESET

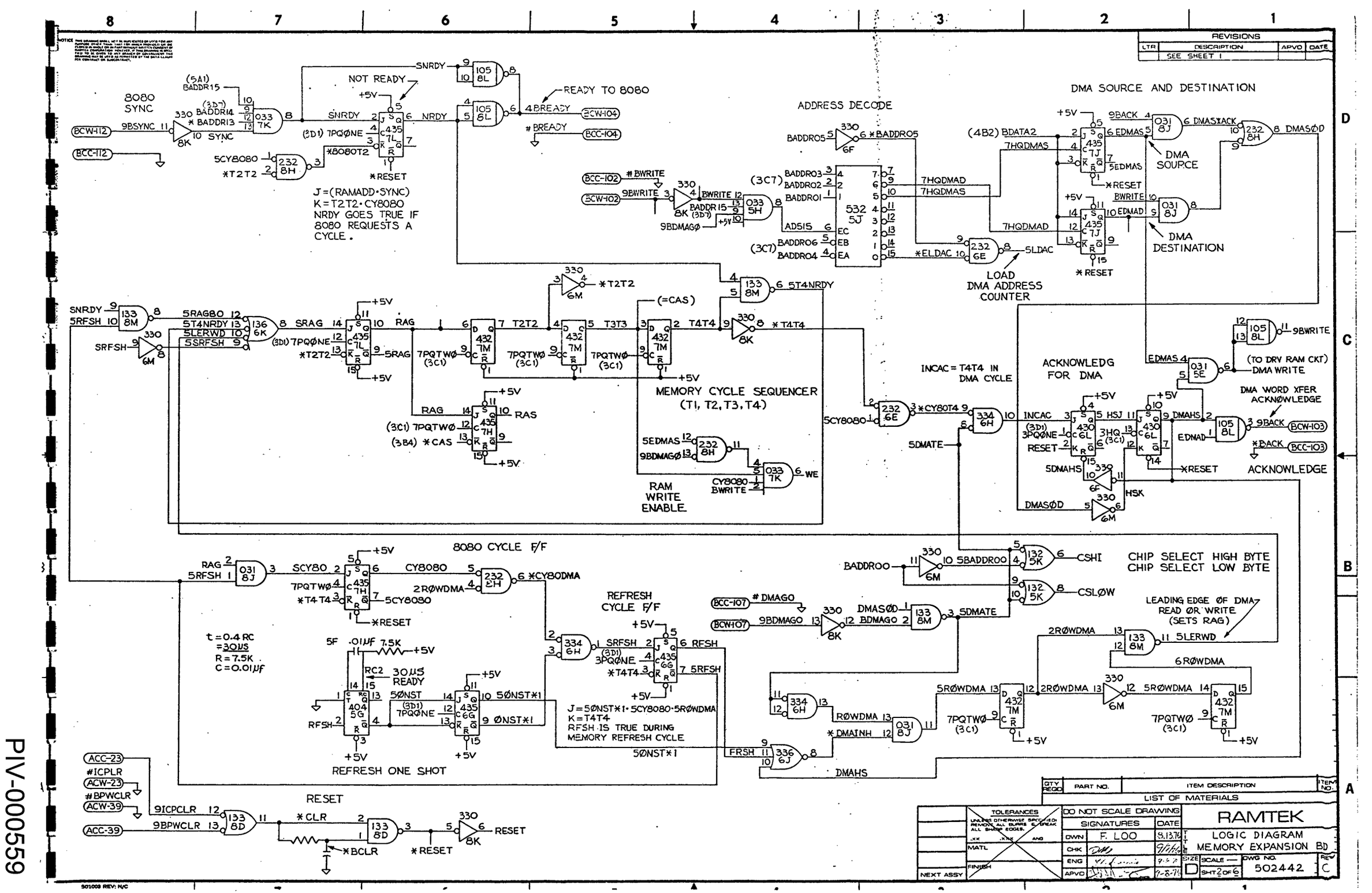
DMADO–15

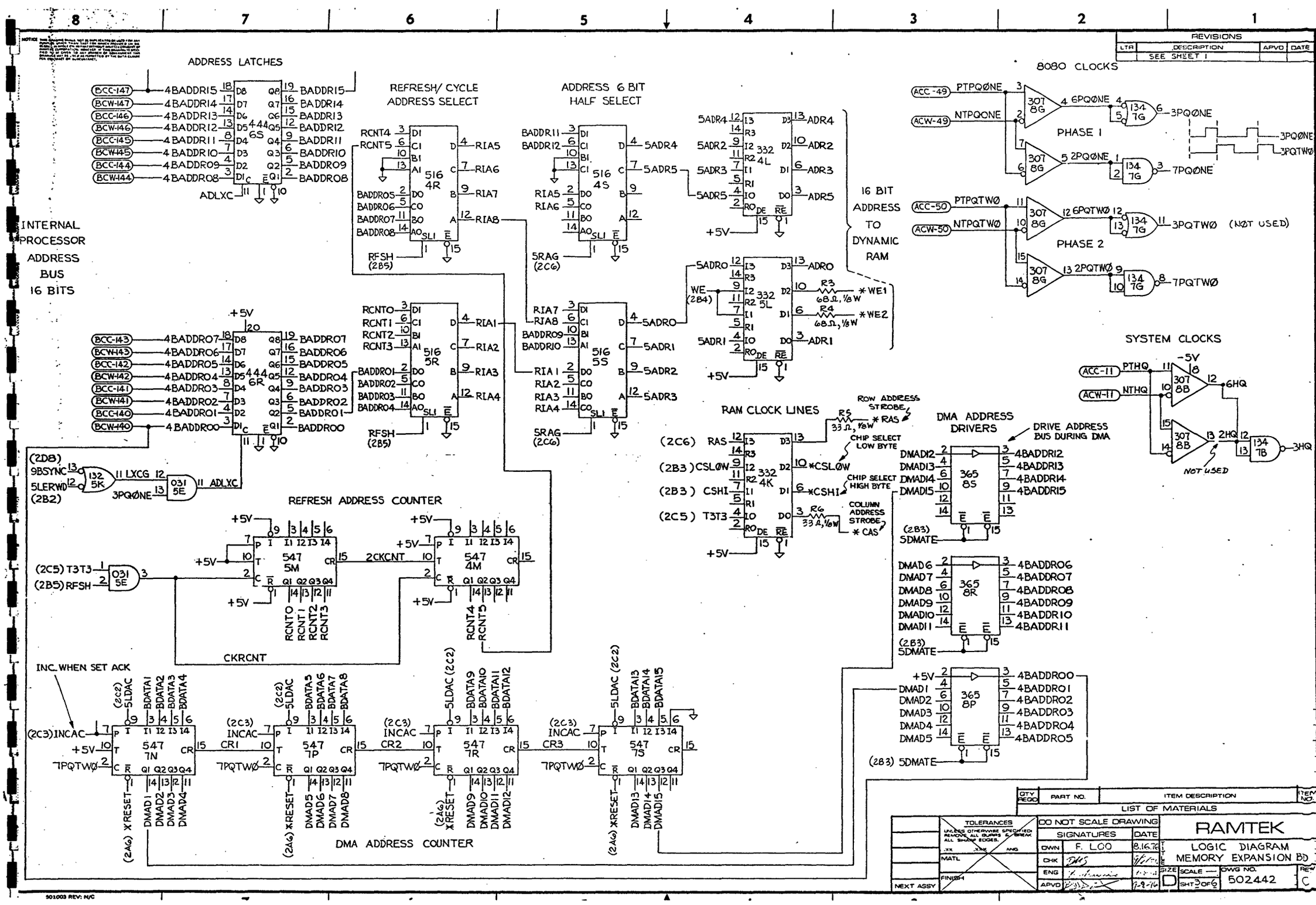
4BADDRO–15

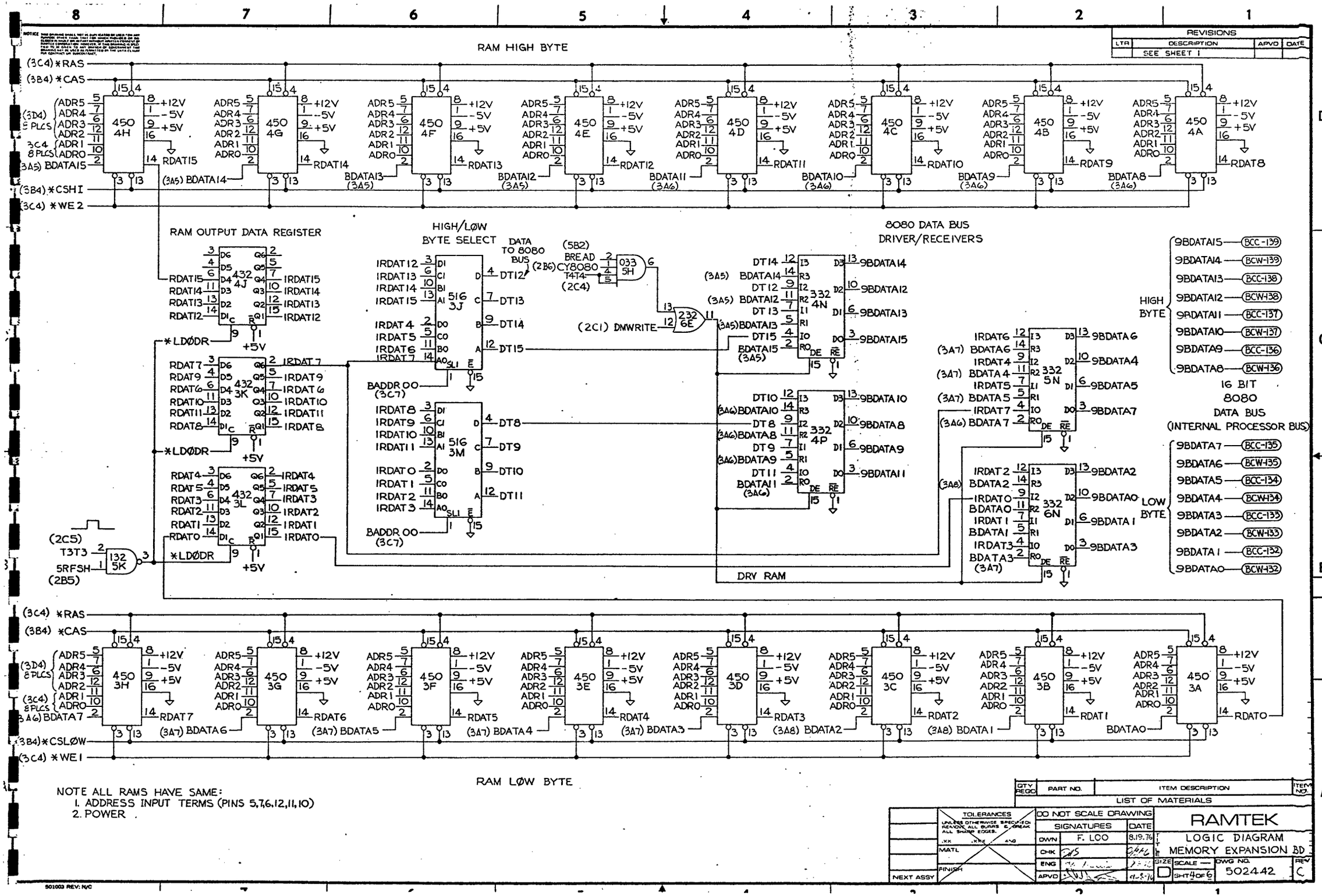
QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX ±.01 .XXX ±.005 ANG	DO NOT SCALE DRAWING		RAMTEK
MATL	SIGNATURES	DATE	MEMORY EXPANSION CARD BLOCK DIAGRAM
7020440 FINISH	DWN F. LOO	9-9-76	
NEXT ASSY	CHK		SIZE D SCALE — DWG NO. 502442 REV C
	ENG		
	APVD		SHT 1 OF 6

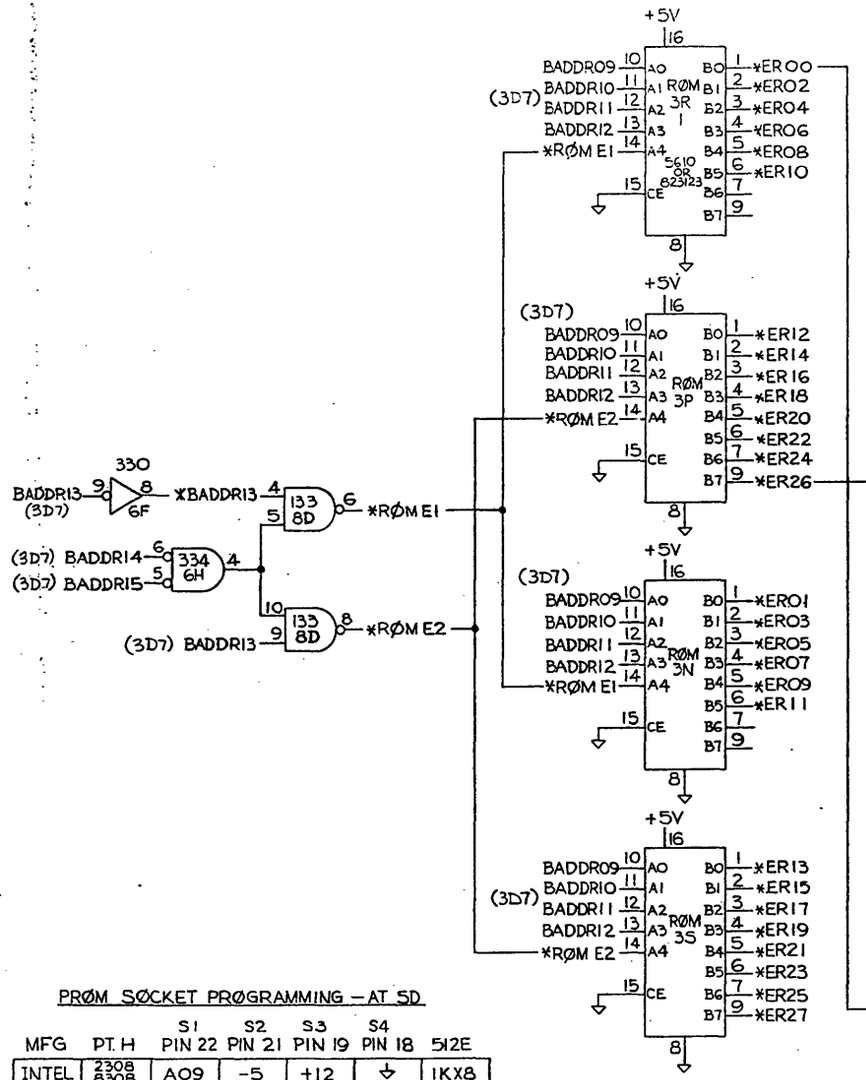

8080 SYNC
NOT READY
READY TO 8080
ADDRESS DECODE
DMA SOURCE AND DESTINATION
DMA SOURCE
DMA DESTINATION
LOAD DMA ADDRESS COUNTER
J=(RAMADD·SYNC)
K=T2T2·CY8080
NRDY GOES TRUE IF 8080 REQUESTS A CYCLE.
MEMORY CYCLE SEQUENCER (T1, T2, T3, T4)
RAM WRITE ENABLE
INCAC = T4T4 IN DMA CYCLE
ACKNOWLEDG FOR DMA
DMA WORD XFER ACKNOWLEDGE
ACKNOWLEDGE
(TO DRV RAM CKT) DMA WRITE
8080 CYCLE F/F
REFRESH CYCLE F/F
REFRESH ONE SHOT
t=0.4 RC =30µS
R=7.5K
C=0.01µF
30µS READY
J=5ØNST*1·5CY8080·5RØWDMA
K=T4T4
RFSH IS TRUE DURING MEMORY REFRESH CYCLE
CHIP SELECT HIGH BYTE
CHIP SELECT LOW BYTE
LEADING EDGE ØF DMA READ ØR WRITE (SETS RAG)
RESET
LIST OF MATERIALS
RAMTEK
LOGIC DIAGRAM MEMORY EXPANSION BD
DWG NO. 502442
REV C
SHT 2 OF 6
PIV-000559


ADDRESS LATCHES
REFRESH/ CYCLE ADDRESS SELECT
ADDRESS 6 BIT HALF SELECT
8080 CLOCKS
INTERNAL PROCESSOR ADDRESS BUS 16 BITS
16 BIT ADDRESS TO DYNAMIC RAM
PHASE 1
PHASE 2
SYSTEM CLOCKS
RAM CLOCK LINES
DMA ADDRESS DRIVERS
DRIVE ADDRESS BUS DURING DMA
ROW ADDRESS STROBE
CHIP SELECT LOW BYTE
CHIP SELECT HIGH BYTE
COLUMN ADDRESS STROBE
REFRESH ADDRESS COUNTER
DMA ADDRESS COUNTER
INC WHEN SET ACK
CKRCNT
NOT USED
REVISIONS
LTR DESCRIPTION APVD DATE
SEE SHEET 1
LIST OF MATERIALS
QTY REQD PART NO. ITEM DESCRIPTION ITEM NO.
DO NOT SCALE DRAWING
SIGNATURES DATE
DWN F. LOO 8.16.76
CHK
ENG
APVD 7-9-76
RAMTEK
LOGIC DIAGRAM
MEMORY EXPANSION BD
SIZE D SCALE — DWG NO. 502442 REV C
SHT 3 OF 6
PIV-000560


RAM HIGH BYTE
RAM LØW BYTE
RAM OUTPUT DATA REGISTER
HIGH/LØW BYTE SELECT
8080 DATA BUS DRIVER/RECEIVERS
DRV RAM
HIGH BYTE
LOW BYTE
16 BIT 8080 DATA BUS (INTERNAL PROCESSOR BUS)
*LDØDR
*CSHI
*CSLØW
*WE2
*WE1
*RAS
*CAS
DMWRITE
BADDR00
+12V
-5V
+5V
NOTE ALL RAMS HAVE SAME:
1. ADDRESS INPUT TERMS (PINS 5,7,6,12,11,10)
2. POWER
REVISIONS
LTR DESCRIPTION APVD DATE
SEE SHEET 1
RAMTEK
LOGIC DIAGRAM
MEMORY EXPANSION BD
DWG NO. 502442
SIZE D
SCALE —
SHT 4 OF 6
REV C
LIST OF MATERIALS
DO NOT SCALE DRAWING
SIGNATURES
DWN F. LOO 8.19.76
CHK
ENG
APVD
PIV-000561

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET 1		

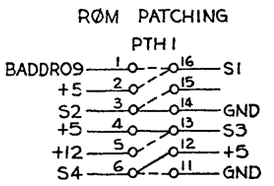
ROM SOCKET CHIP SELECT DECODING



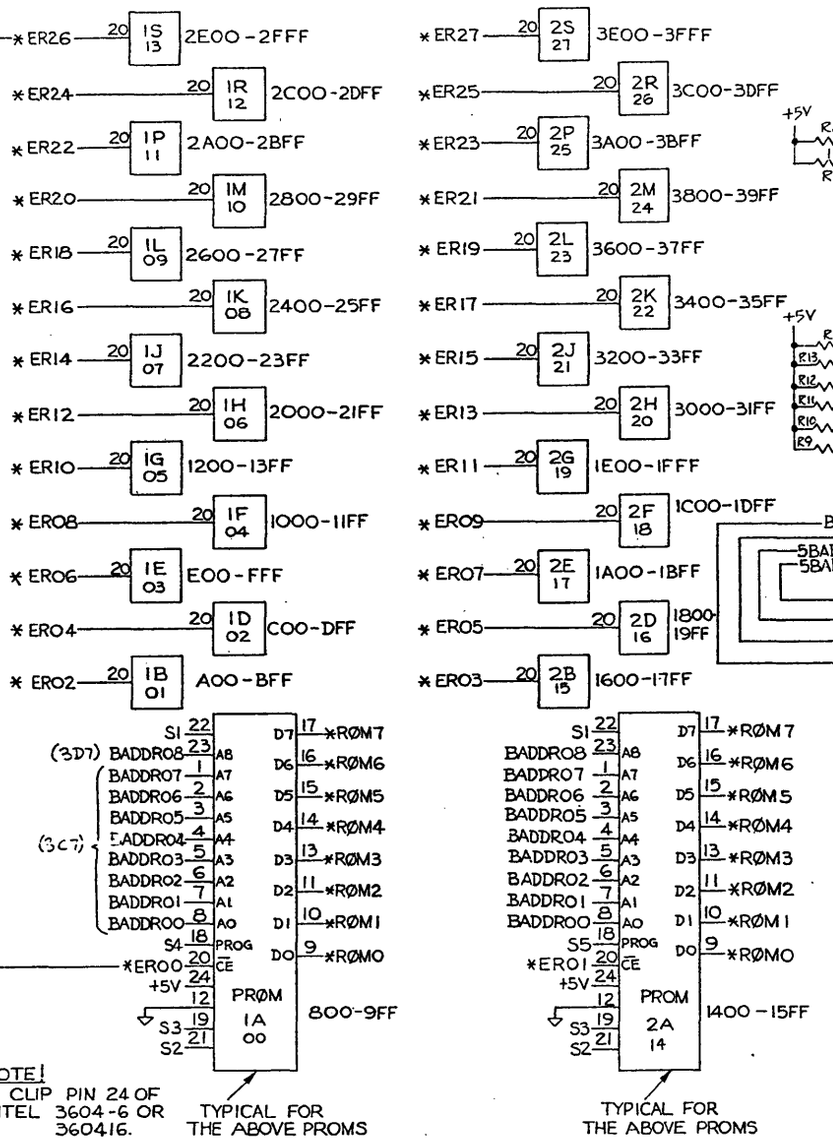
PRØM SØCKET PRØGRAMMING — AT 5D

MFG	PT. H	S1 PIN 22	S2 PIN 21	S3 PIN 19	S4 PIN 18	512E
INTEL	2308 8308	A09	-5	+12	⏚	1KX8
INTEL	2708	A09	-5	+12	⏚	1KX8
INTEL	3624 (-4)	+5	⏚	+5	+5	512X8
INTEL	3604 (-4)	+5	⏚	+5	+5	512X8
INTEL	3604-6	+5	⏚	+5	+5	512X8
AMD	9208	A09	X	+12	+5	1KX8
FCHD	93438 93448	X	⏚	+5	+5	512X8
INTRSR	IM5603 IM5625	X	⏚	+5	+5	512X8

RØM PATCHING



MEMORY EXPANSION ROM



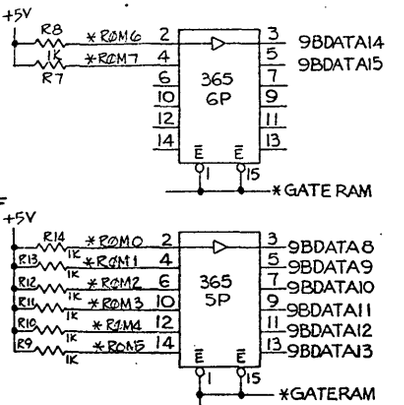
NOTE 1
CLIP PIN 24 OF INTEL 3604-6 OR 360416.

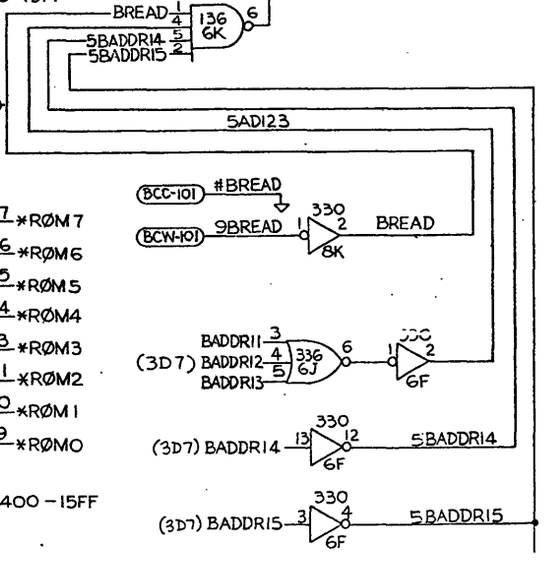
TYPICAL FOR THE ABOVE PROMS

NOTE 1
ALL PRØMS HAVE IDENTICAL TERMS EXCEPT PIN 20.

TYPICAL FOR THE ABOVE PROMS

DATA BUS DRIVERS





QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.
		LIST OF MATERIALS	

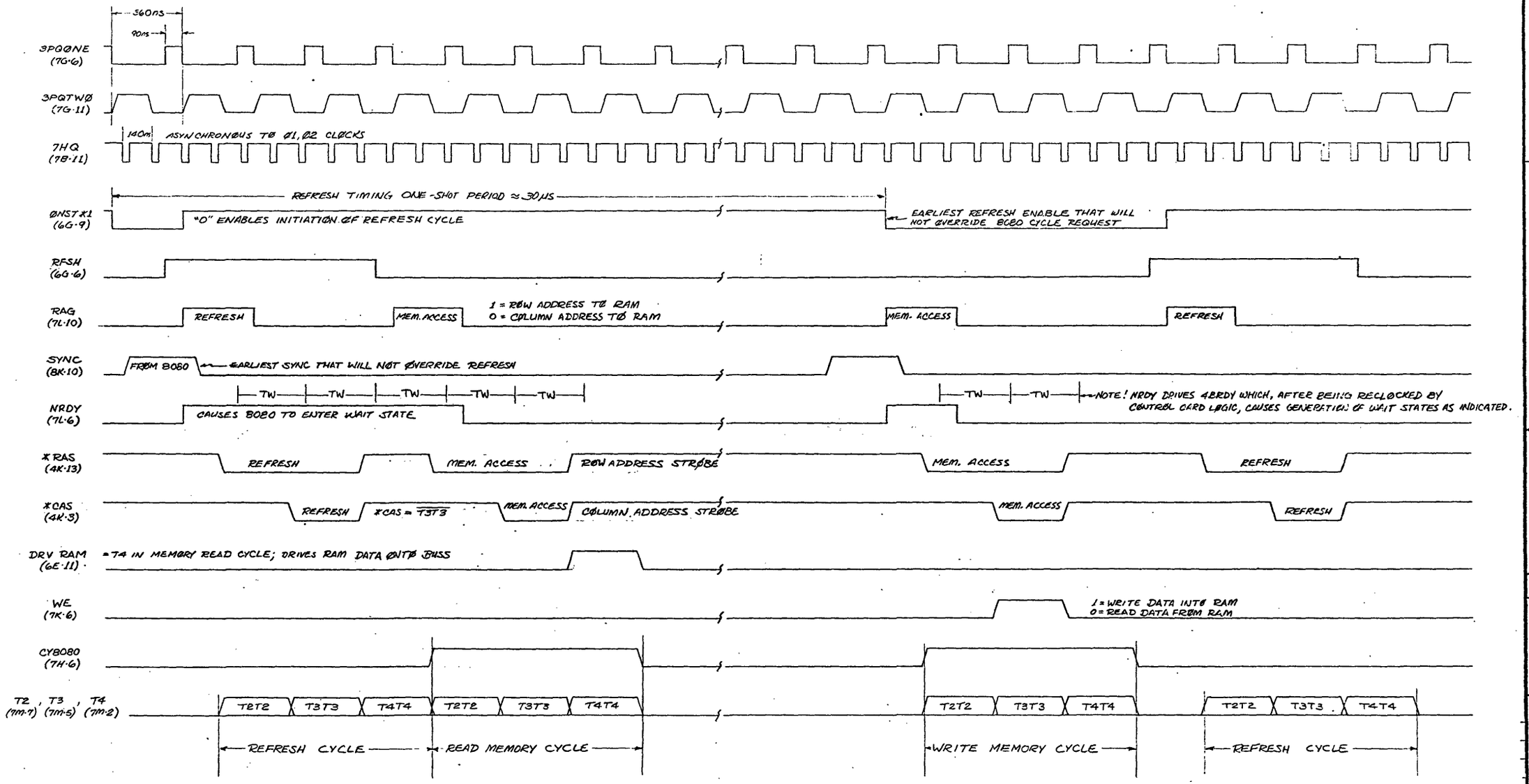
TOLERANCES	DO NOT SCALE DRAWING		
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG	SIGNATURES	DATE	RAMTEK
MATL	DWN F. LOO	6.20.76	LOGIC DIAGRAM
FINISH	CHK		MEMORY EXPANSION BD
NEXT ASSY	ENG		SIZE D SCALE — DWG NO. 502442 REV C
	APVD	9-8-76	SHT 5 OF 6

501003 REV: N/C

PIV-000562

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHEET 1		

MEMORY ACCESS / REFRESH TIMING



QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.
		LIST OF MATERIALS	

TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG	DO NOT SCALE DRAWING	RAMTEK	
MATL	SIGNATURES / DATE	MEMORY OPTION CARD	
FINISH	DWN H. Burris 6/7	SLM ACCESS/REFRESH TIMING	
NEXT ASSY	CHK DM 7/7	SIZE D	SCALE —
	ENG H. Burris 6-30-77	SHT 6 OF 6	DWG NO. 502442 REV C
	APVD [signature] 6-30-77		

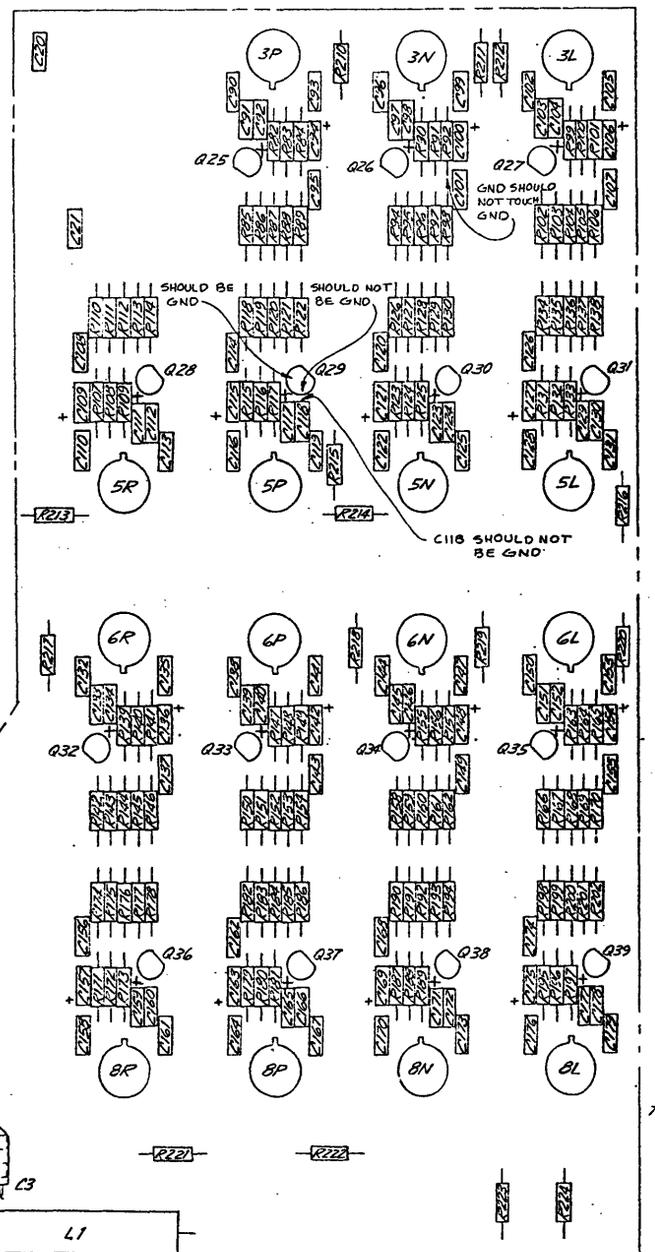
501003 REV: N/C

PIV-000563

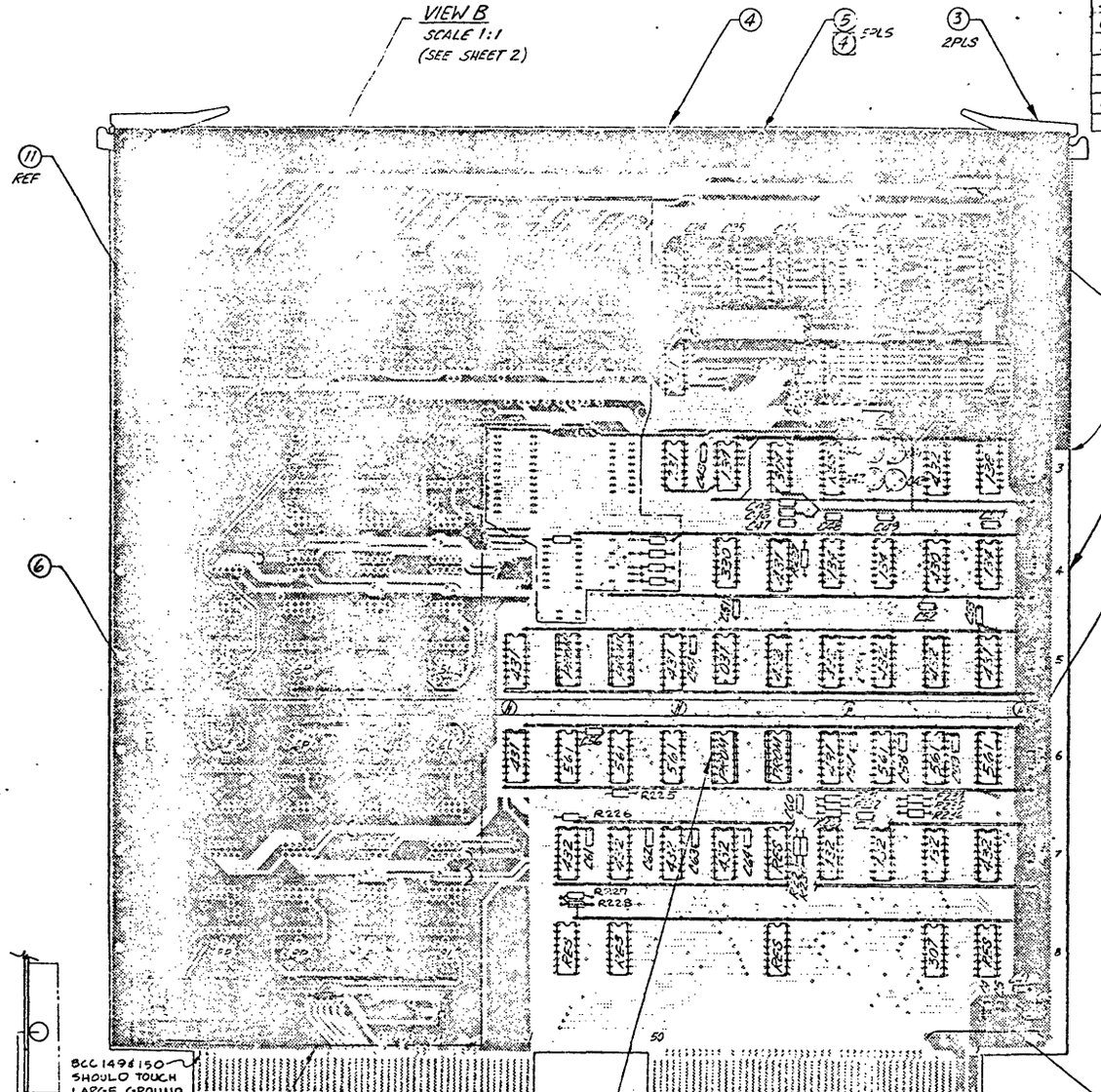
PIV-000564

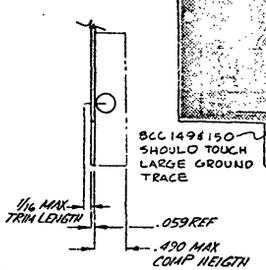
REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	PRODUCTION REL (ECO 506736)	DWS	8/3/76
B	REV PER ECO # 507156		
C	REV PER ECO # 507190	RCL	10/22/76
D	REV PER ECO # 507176	JHC	12-1-76
E	REV PER ECO # 507233	RCL	1-3-77
F	REV PER ECO # 507245	RCL	1-6-77
G	REV PER ECO # 507382	DHO	5/17/77
H	REV. PER ECO # 507437	TCI	6/14/77

VIEW A
SCALE 2:1



VIEW B
SCALE 1:1
(SEE SHEET 2)





VIEW A
SCALE 1:1

13 4PLS NOTE: PROMS ARE TO BE LOADED BY FILE TEST

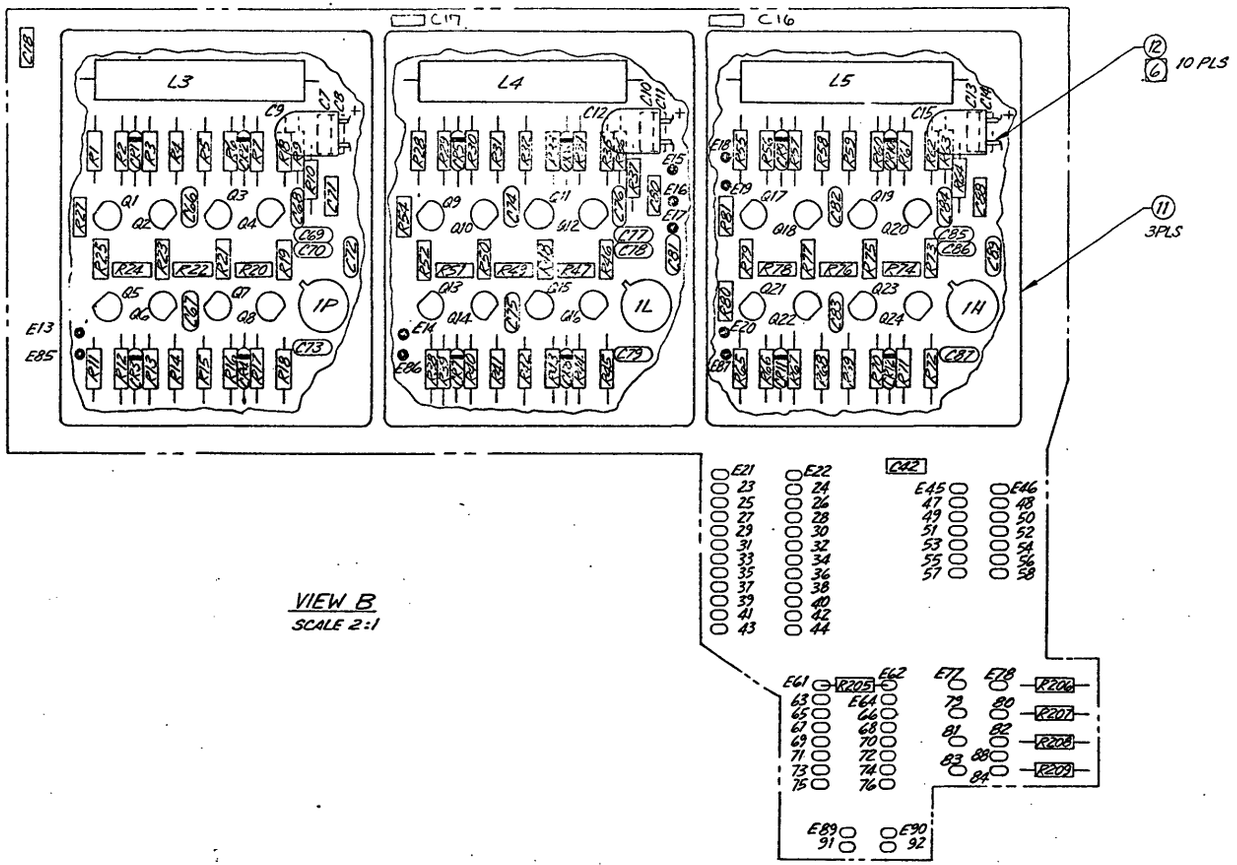
SEE SHEET 2 OF 2 FOR NOTES

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.
LIST OF MATERIALS			

TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. XX ± .XXX ± ANG ±	DO NOT SCALE DRAWING		RAMTEK
	SIGNATURES	DATE	
MATL ——	DWN		ASSEMELY DRAWING VIDEO TYPE I
FINISH ——	CHK		
NEXT ASSY	ENG		SIZE D SCALE 1:1 DWG NO. 502446 REV H
	APVD		SHT 1 OF 2

502446

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT 1		



VIEW B
SCALE 2:1

NOTES, UNLESS OTHERWISE SPECIFIED:

1. INK STAMP SERIAL NUMBER AND REVISION LEVEL IN AREA SHOWN
2. REFERENCE DRAWINGS:
   LOGIC DIAGRAM - 502448
   FABRICATION DWG - 502447
3. LOGIC DEVICE TYPE NUMBERS ARE RAMTEK STANDARD PART NUMBERS 1301XXX
4. INSERT ROLL PIN (ITEM 5)
5. CUT BUSS BAR (ITEM 10) TO LENGTH REQ'D
6. ADD SLEEVING (ITEM 12) TO COMPONENT LEAD

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.
.XX .XXX ANG
MATL —
FINISH —
NEXT ASSY

DO NOT SCALE DRAWING

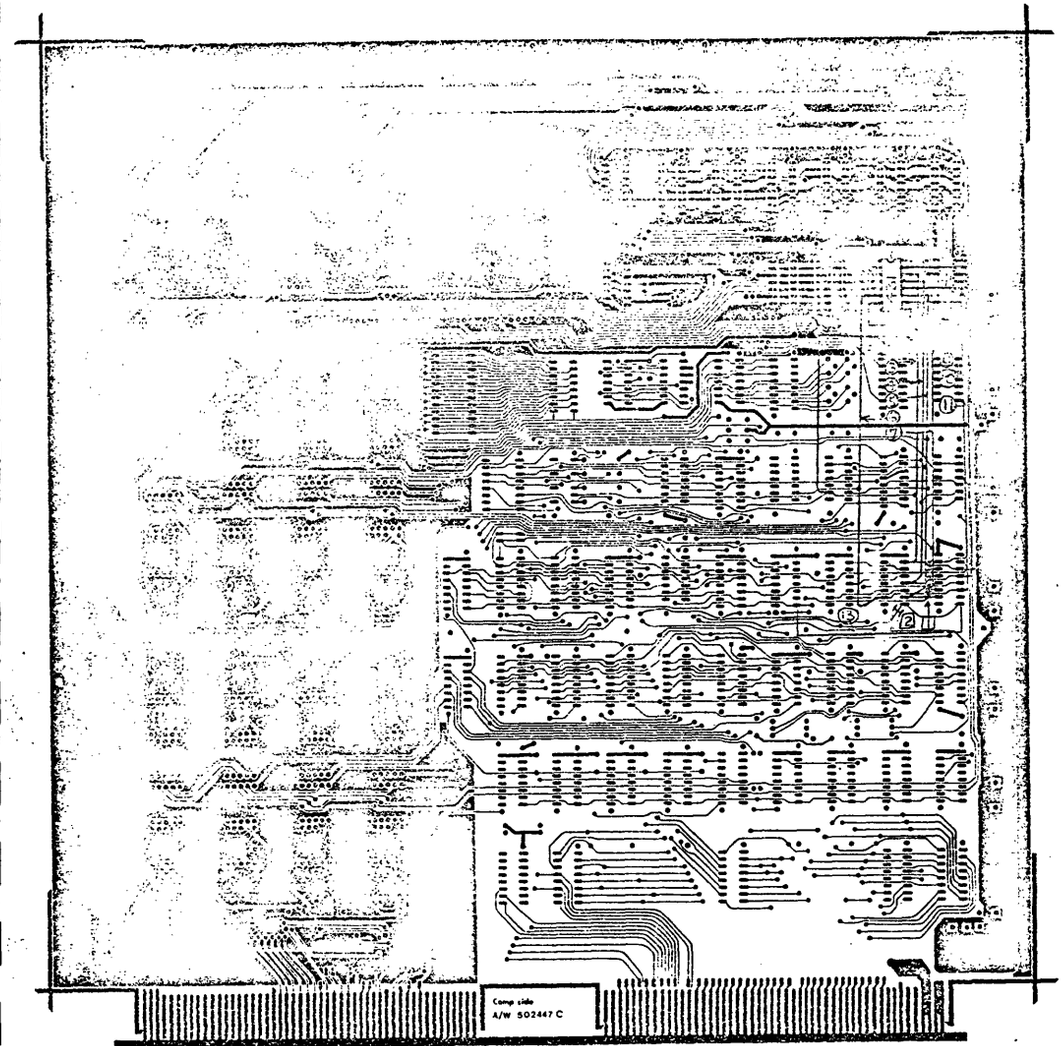
SIGNATURES		DATE
DWN	M.J	7-18-74
CHK	DMW	8/8/74
ENG	P.L.	7/16/74
APVD	S.L.	7/18/74

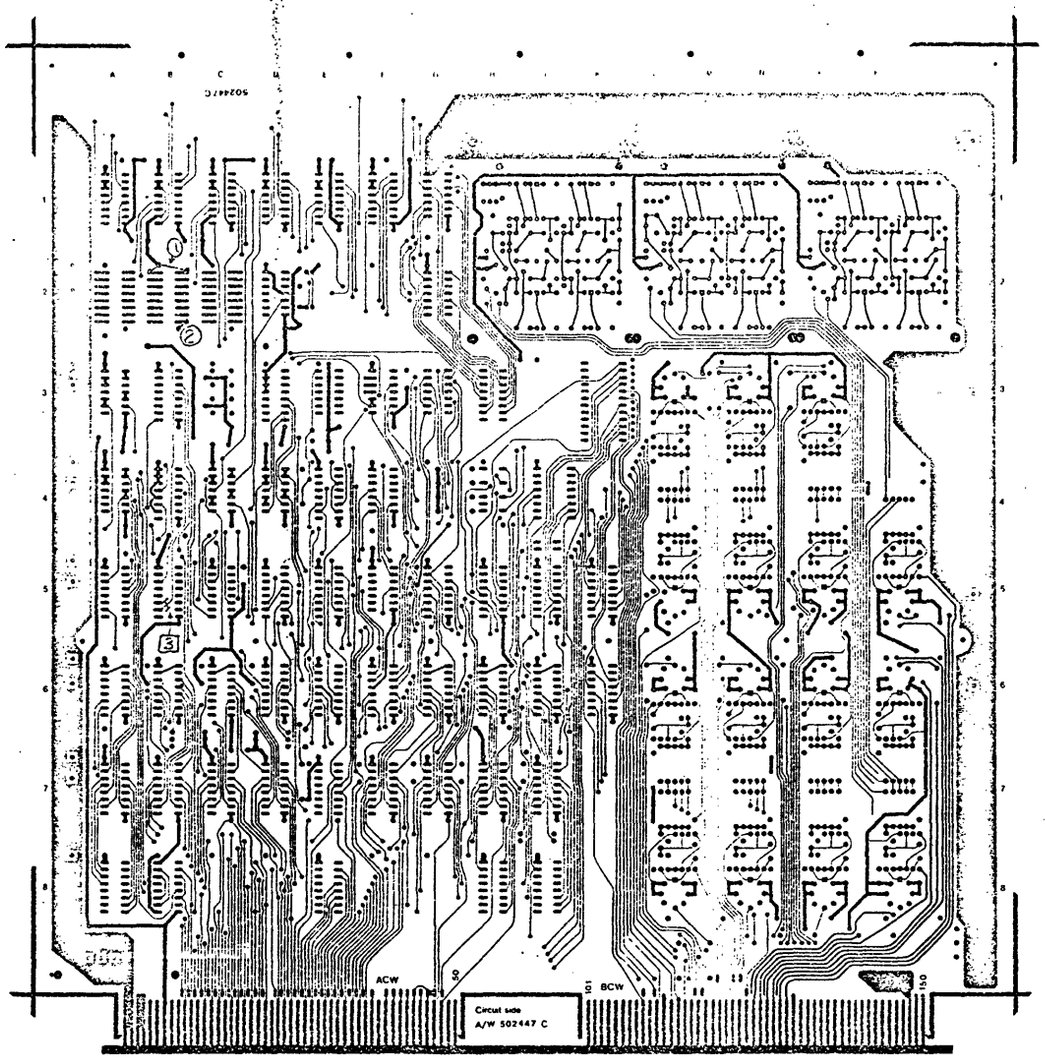
RAMTEK

TITLE: ASSEMBLY DRAWING
VIDEO TYPE I

SIZE D | SCALE 2:1 | DWG NO. 502446 | REV H | SHT 2 OF 3

502446





⚠ INSTALL A 745174 AT 2B.

RAMTEK	
ARTWORK, VIDEO TYPE 1	
SIZE	502447 C

DELETES

1. CUT TRACE AT 5B-15, 5B-12, 5B-10, & 5B-7 (comp side).
2. CUT TRACE AT 5B-9 AS SHOWN. (comp side).
3. CUT TRACE AT 5B-10 (CKT SIDE).

ADDS

1. 2B-1 TØ 2B-16 & THE FEED THRU SHOWN (CKT SIDE)
2. 2B-8 TØ FEEDTHRU SHOWN. (CKT SIDE)
3. 2B-14 TØ 5B-15 (comp side)
4. 2B-13 TØ 5B-12 (comp side)
5. 2B-11 TØ 5B-10 (comp side)
6. 2B-6 TØ 5B-7 (comp side)
7. 2B-7 TØ 5A-6 (comp side)
8. 2B-10 TØ 4F-5 (comp side) NOT SHOWN
9. 2B-10 TØ 5A-11 (comp side)
10. 2B-12 TØ 5A-13 (comp side)
11. 2B-15 TØ 5A-14 (comp side)
12. 2B-9 TØ 4C-6 (comp side)
13. 5A-9 TØ 5D-9 (comp side)

RAMTEK	
ARTWORK, VIDEO TYPE 1	
SIZE	502447 C

DO NOT SCALE DRAWING		RAMTEK
SIGNATURES	DATE	ASSY DWG VIDEO TYPE 1
DWN		
CHK		
ENG		SIZE D SCALE DWG NO 502446 REV H
APVD		SHT 1 OF 1

PIV-000566

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
A	[illegible] PER ECO # 507036		9-16-76
B	REV PER ECO 507066	TMS	[illegible]
C	REV PER ECO 507176	CC	12-8-76
D	REV PER ECO 507245	RCH	1-6-77
E	REV PER ECO 507382	HHO	3/3/77
F	REV. PER ECO # 507437	TCI	6/15/77

DATA FROM MEMORY

RECLOCK

BLANK DATA & ELEMENT & FIELD SELECT

RECLOCK

OVERLAYS & CURSORS MIX

RECLOCK

VIDEO AMP

+5VDC
1K
48 PLCS (4 BECKMAN 898-1-1.0K)
220 ← 12 PLCS

9MV_00_
9MV_01_
9MV_02_
9MV_03_
9MV_04_
9MV_05_
9MV_06_
9MV_07_
9MV_08_
9MV_09_
9MV_10_
9MV_11_
4 LINES

4 74LS174 SHT 3 CLK
3 74S253 SHT 3 SELECTOR (LSB) (MSB)
4 74LS174 SHT4
3 74S253 SHT 4 (LSB) (MSB)
7VHAF
2ESEL
6FLDSEL
7VFUL

7DAT00
7DAT01
7DAT02
7DAT03
7DAT04
7DAT05
7DAT06
7DAT07
7DAT08
7DAT09
7DAT10
7DAT11

1 74S174 SHT 3 CLK
1 74S174 SHT 4
6DAT00 – 01 – 02 – 03 – 04 – 05
6DAT06 – 07 – 08 – 09 – 10 – 11

2 512X4 PROM SHT 3
2 512X4 PROM SHT 4
7CURS1,2
VID00
VID01
VID02
VID03
VID04
VID05
VID06
VID07
VID08
VID09
VID10
VID11
TO DAC OPTION
OVERLAY 1,2

1 74S174 SHT 3
1 74S174 SHT 4

12-VIDEO AMP SHT 8&9
7CBLK,1
7CSYN,1
4CUR$1 10% BRIGHTNESS
4CUR$2 10%

VIDEO00
VIDEO01
VIDEO02
VIDEO03
VIDEO04
VIDEO05
VIDEO06
VIDEO07
VIDEO08
VIDEO09
VIDEO10
VIDEO11

COMP BLK
VERT DRIVE
HORIZ. DRIVE
COMP SYNC

9TCBBB
9TVDDD
9THDDD
9TCSSS

1 74LS174 SHT 2
1 74S174 SHT 2
1 74S174 SHT 2

7DCBBB
7DVDDD
7DHDDD
7DCSSS
4CBLK
CSYN

SHT2 4 LEVEL CONVERTERS & LINE DRIVERS

9VCMPBLE
9VVERTE
9VHORZE
9VCMSYE

COMP BLANKING
VERT. DRIVE
HORIZ. DRIVE
COMP. SYNC

(2) 7CBLK,1
(2) 7CSYN,1
TO VIDEO AMPS

VID00 THRU VID11

3VQFUL
CLK 6 74S112 SHTS 5,6&7

3-VIDEO AMPS SHT 8&9
7CBLK,1
7CSYN,1
4CUR$1
4CUR$2

SHT 5 4 BIT D TO A OR MSB 8 BIT DAC
SHT 6 4 BIT D TO A OR LSB 8 BIT DAC
SHT 7 4 BIT D TO A
4 LINES
DACVID1
DACVID2
DACVID3
DVIDEO1
DVIDEO2
DVIDEO3

CURSOR1,3
CURSOR2,4
ELEMENT SELECT
FIELD SELECT
RESOLUTION GATE1
RESOLUTION GATE2

9CURS1,3
9CURS2,4
9DSRB1
9DRFLD
9ESEL
9FLDSEL

1 74LS174 SHT 2&10 CLK
1 74S174 SHT 10 CLK
DATA ELEM. SELECT
2ESEL
6FLDSEL
DATA FIELD SELECT

7CURS1,3 TO MIX WITH DATA
7CURS2,4
4CUR$1 TO VIDEO AMP
4CUR$2 10% BRIGHTNESS

SYSTEM CLOCK 1/2 FREQUENCY
NTHQ
PTHQ
NTBQ
PTBQ
SYSTEM CLOCK
SHT 2
7VHAF
7VFUL
3VQFUL

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG

MATL

502446

FINISH

NEXT ASSY

DO NOT SCALE DRAWING

SIGNATURES		DATE
DWN	M. Cole	3/11/76
CHK	R. Hall	3/[illegible]
ENC	[illegible]	[illegible]
APVD	[illegible]	7/14/76

RAMTEK

BLOCK DIAGRAM-VIDEO TYPE 1

SIZE D SCALE — DWG NO. 502448 REV F

SHT 1 OF 11

PIV-000567

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT 1		

	9100	9200	9300
RESOLUTION	256L x 320E	256L x 640E	525L x 640E
9ESEL	OVOLTS	5VOLTS	5VOLTS
9FSEL	OVOLTS	OVOLTS	5VOLTS

MEMORY DATA ELEMENT SELECT

+5VDC, 1K, 7E, 16, 8

ACW 45 — 9ESEL

+5VDC, 3VQFUL1, 9MDRB3, 8ESEL, 7VFUL, 7ESEL, 2ESEL

ELEMENT SELECT IS OVOLTS FOR LOW RES. SYSTEMS & A 7MHZ SQUARE WAVE FOR MED & HIGH RES SYSTEMS. TO SHT 3&4

8DCBBB1, DCBBB — TO MIX WITH DATA & CURSORS SHT 3,4 & 10

4BCLK, CSYN

ACC 43 — 9TCBBB
ACC 45 — 9THDDD
ACW 43 — 9TCSSS
ACW 45 — 9TVDDD

8DCBBB, 8DHDDD, 8DCSSS, 8DVDDD → 7DCB, 7DHD, 7DS, 7DVD

7DCBBB, 7DHDDD, 7DCSSS, 7DVDDD → 6CBBB, 6HDDD, 6CSSS, 6VDDD, 3CBLK, 1CYSN

7GCLK

7GCLK - NOTE IN LOW RESOLUTION SYSTEMS (9100) 7GCLK WILL APPEAR AS A 7VFUL CLOCK. IN MEDIUM & HIGH RESOLUTION SYSTEMS (9200 & 9300) 7GCLK WILL APPEAR AS A 7VFUL IF THE FIRST ELEMENT TO BE DISPLAYED ON THE BEGINNING OF A LINE IS EVEN; IF AN ODD ELEMENT IS TO BE DISPLAYED FIRST 7GCLK WILL APPEAR AS A 7VHAF CLOCK.

7VHAF, 3QVFUL1, NEWCLK, 2VFUL

ACW 15 — 9DSRB1 — DSRB1

7DCB, 7DHD, 7DS, 7DVD → 7DCBBB, 7DHDDD, 7DCSSS, 7DVDDD

7V FUL

ACW 11 — NTHQ
ACC 11 — PTHQ
→ 6VHAF (NOT USED)
→ 2VHAF → 7VHAF

ACW 10 — NTBQ
ACC 10 — PTBQ
→ 2VFUL → 7VFUL
→ 6VFUL → 3VQFUL, 3VQFUL1

3VFUL, 7VFUL, 2VHAF, 7VHAF — CLOCKING EDGE

MEMORY DATA FIELD SELECT

+5VDC, 1K, 7E, 16, 12

ACW 48 — 9FSEL
BCW 109 — 9DRFLD → 8FSEL → FLDSEL → 7FLDSEL → 6FLDSEL

FIELD SELECT IS OVOLTS FOR LOW & MED RES. SYSTEMS & A 30 Hz SQUARE WAVE FOR HIGH RES. SYSTEMS. TO SHT 3&4

4CBBB → 2CBBB → Q41 2N4258 → 9VCMPBLE (BCC 111), #VCMPLB (BCC 112)
4HDDD → 2HDDD → Q43 2N4258 → 9VHORZE (BCC 113), #VHORZE (BCC 114)
4CSSS → 2CSSS → Q40 2N4258 → 9VCMSYE (BCW 111), #VCMSYE (BCW 112)
4VDDD → 2VDDD → Q42 2N4258 → 9VVERTE (BCW 113), #VVERTE (BCW 114)

C46 .01μf, -5VDC

7CBLK, 7CBLK1, 7CSYN, 7CSYN1 → TO VIDEO AMPS SHT 8 & 9

R237 470, 3VD1, 2VD1, C45 .01μf

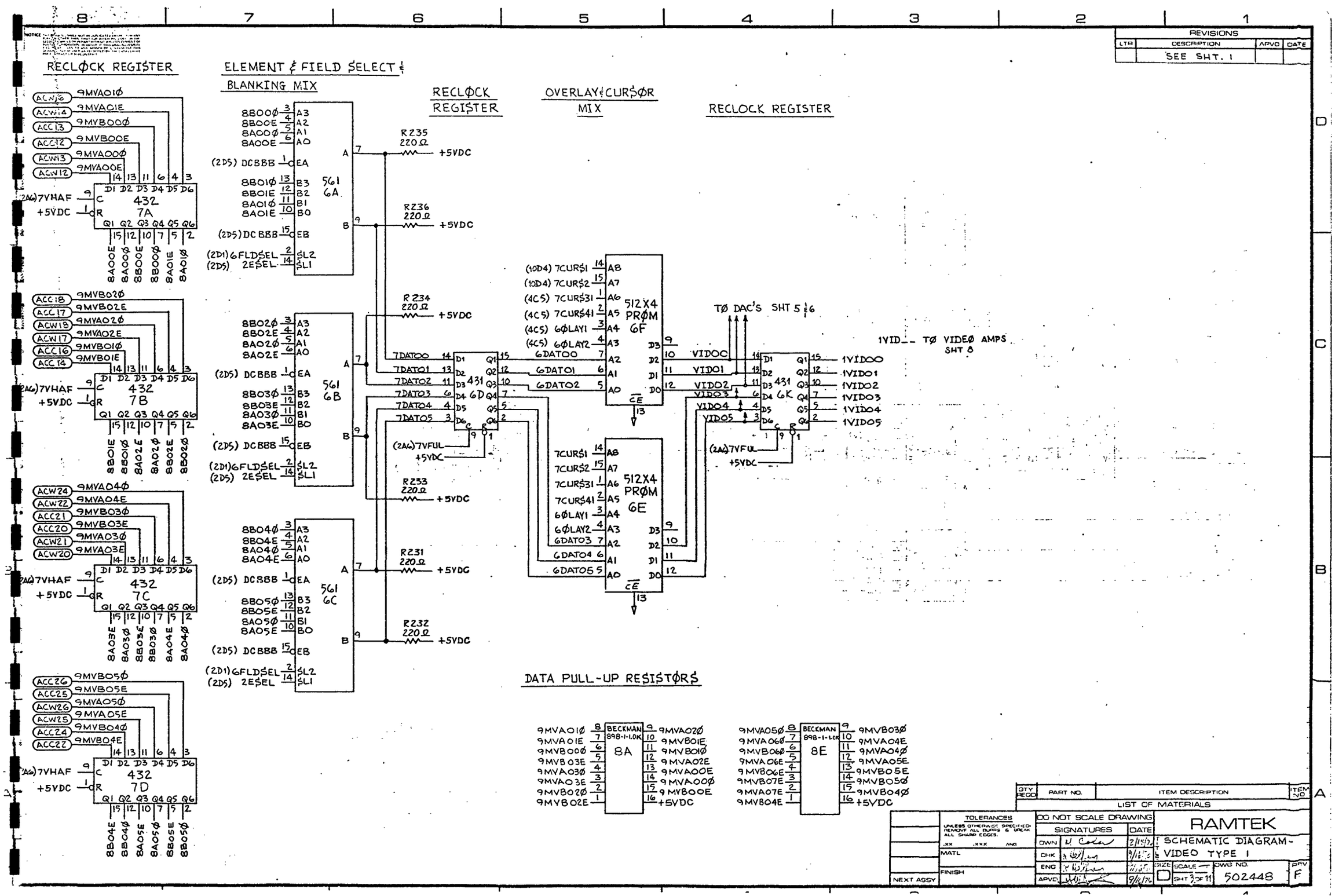
#CURS4 (ACC 35)
#CURS 3 (ACC 31)
#DSRB1 (ACC 15)
#CURS1 (ACC 19)
#FSEL (ACC 48)
#ESEL (ACC 47)
#CURS2 (ACC 27)
#TCBBB (ACC 44)
#THDDD (ACC 46)
#TCSSS (ACW 44)
#TVDDD (ACW 46)

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.
LIST OF MATERIALS			

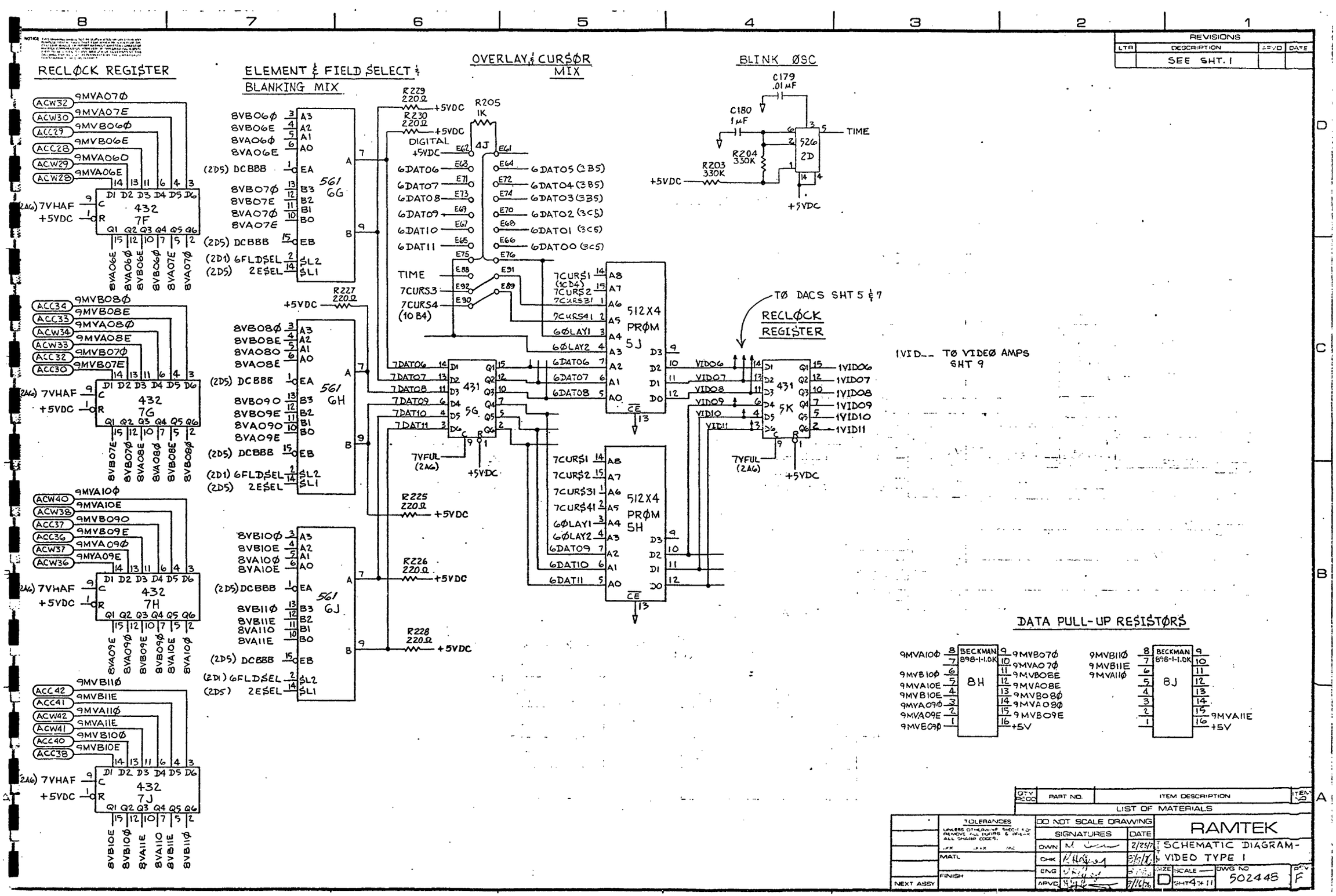
TOLERANCES	DO NOT SCALE DRAWING	SIGNATURES	DATE	RAMTEK
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG	DWN		9-2-76	VIDEO 1 LOGIC DIAGRAM
MATL	CHK			B (TIMING, SYNC, BLANKING)
FINISH	ENG			SIZE D — SCALE — DWG NO. 902448 — REV F
NEXT ASSY	APVD		9/4/76	SHT 2 OF 11

901003 REV: N/C

PIV-000568


RECLOCK REGISTER
ELEMENT & FIELD SELECT & BLANKING MIX
RECLOCK REGISTER
OVERLAY & CURSOR MIX
RECLOCK REGISTER
TO DAC'S SHT 5 & 6
1VID-- TO VIDEO AMPS SHT 8
DATA PULL-UP RESISTORS
REVISIONS
LTR DESCRIPTION APVD DATE
SEE SHT. 1
LIST OF MATERIALS
QTY REQD PART NO. ITEM DESCRIPTION ITEM NO
DO NOT SCALE DRAWING
TOLERANCES
SIGNATURES DATE
DWN
CHK
ENG
APVD
MATL
FINISH
NEXT ASSY
RAMTEK
SCHEMATIC DIAGRAM- VIDEO TYPE 1
SIZE D SCALE — DWG NO. 502448 REV F
SHT 3 OF 11

PIV-000569



REVISIONS

LTR	DESCRIPTION	APVD	DATE
	SEE SHT. 1		

LIST OF MATERIALS

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO

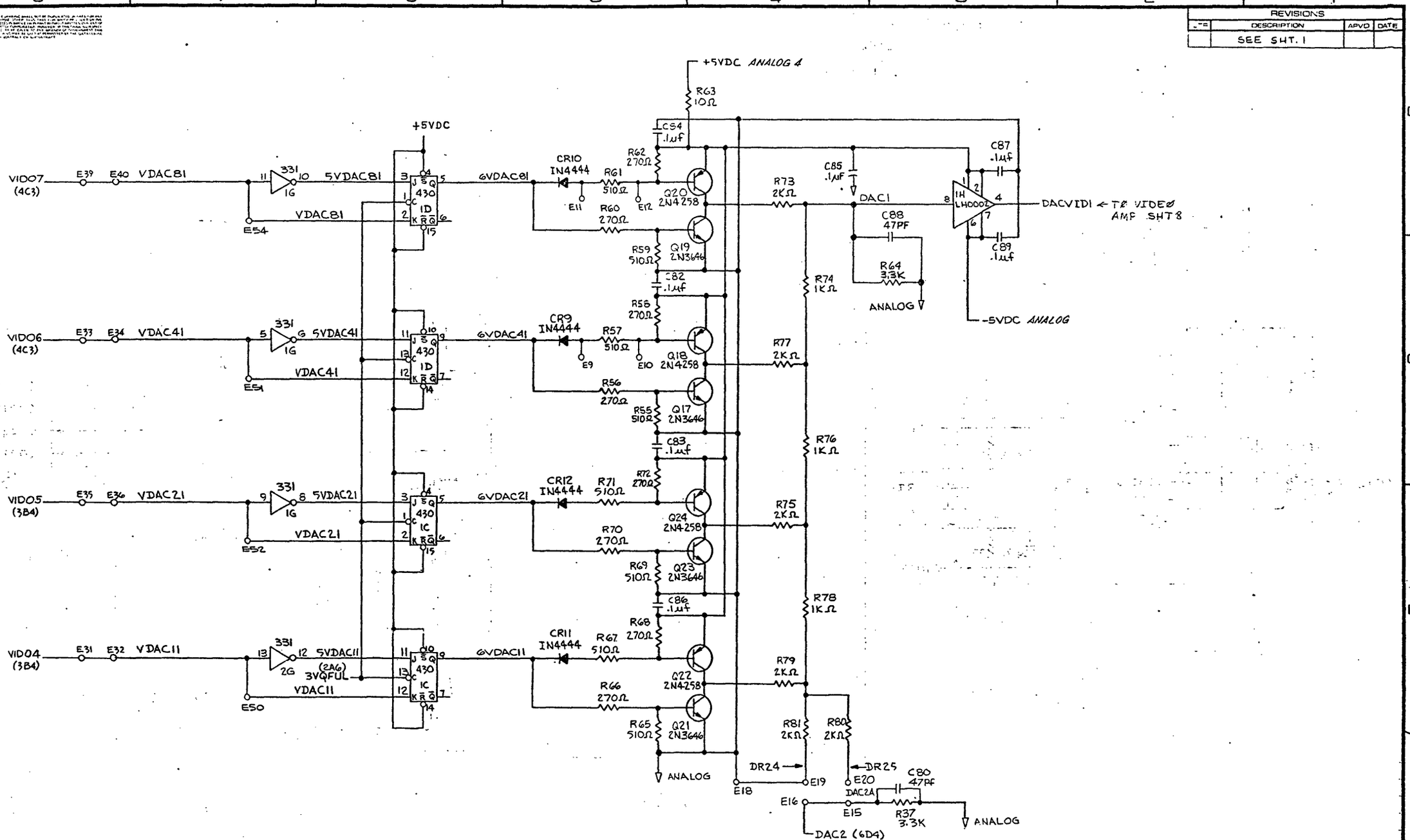
TOLERANCES	DO NOT SCALE DRAWING	SIGNATURES	DATE
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG	DWN		2/25/76
MATL	CHK		3/5/76
FINISH	ENG		3/16/76
NEXT ASSY	APVD		7/16/76

RAMTEK

SCHEMATIC DIAGRAM-
VIDEO TYPE 1

SIZE D | SCALE — | DWG NO 502448 | REV F | SHT 4 OF 11

PLV-000570



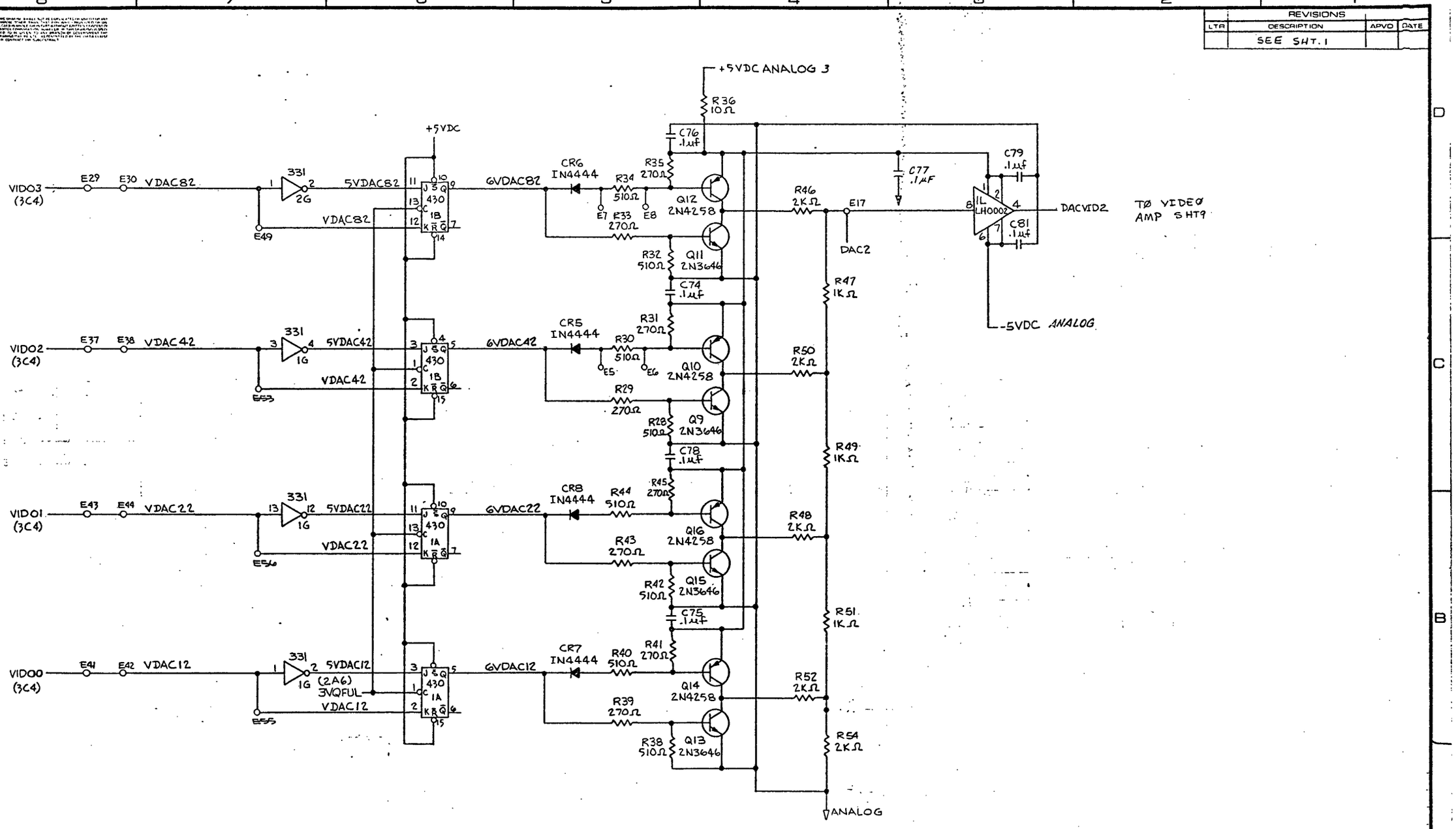
REVISIONS			
ZONE	DESCRIPTION	APVD	DATE
	SEE SHT. 1		

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

TOLERANCES	DO NOT SCALE DRAWING		RAMTEK
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.	SIGNATURES	DATE	
.XX .XXX ANG	DRN	2/24/76	LOGIC DIAGRAM VIDEO TYPE I
MATL	CHK	3/3/76	& MSB OF 8BIT DAC OR 4BIT DAC
FINISH	ENG	3/17/76	SIZE D SCALE — DWG NO 502448 REV F
NEXT ASSY	APVD	9/14/76	SHT 5 OF 11

PIV-000571



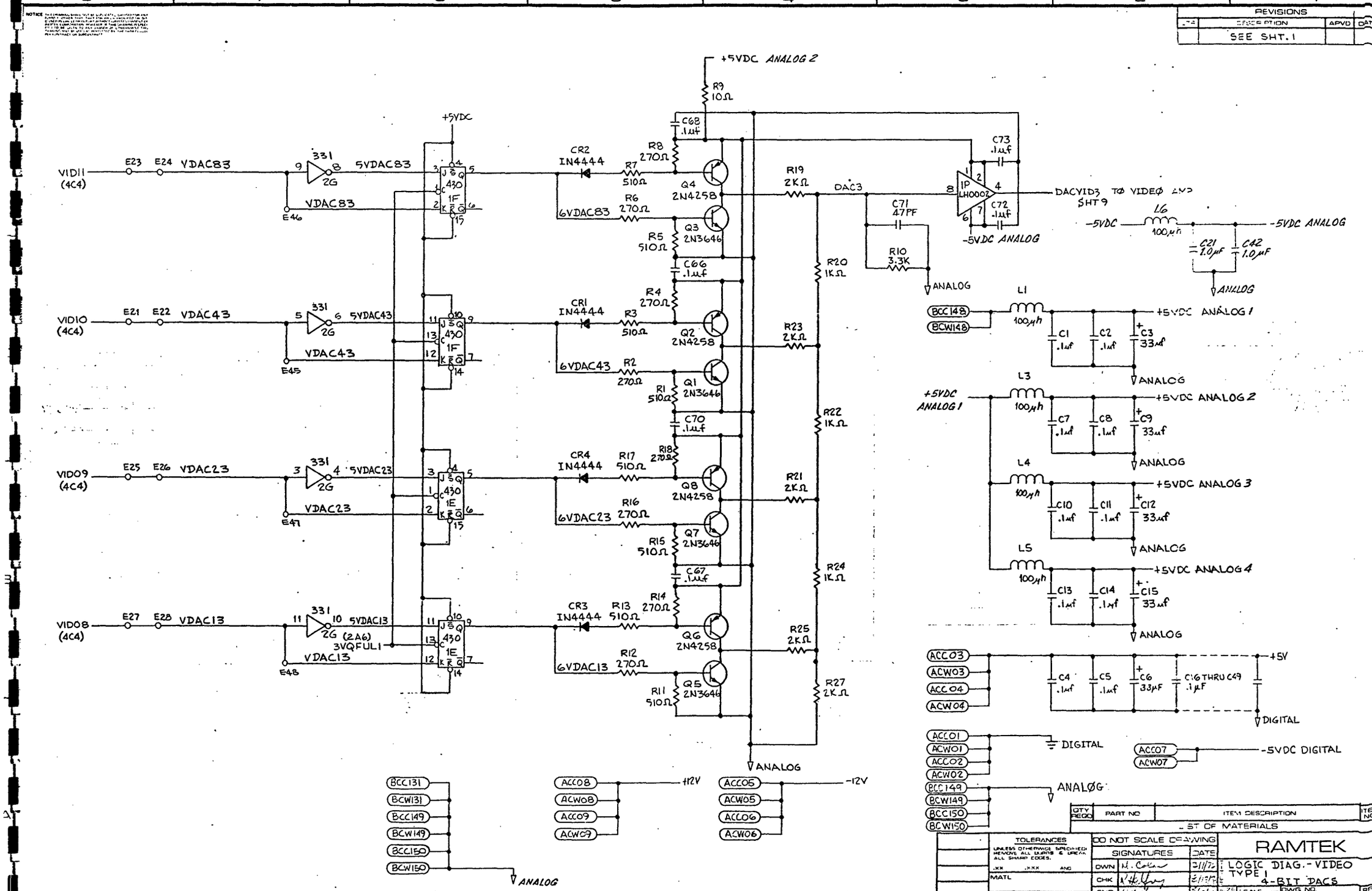
REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT. 1		

QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO

LIST OF MATERIALS

TOLERANCES	DO NOT SCALE DRAWING		
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG	SIGNATURES	DATE	RAMTEK
MATL	DWN	2/14/77	LOGIC DIAG. - VIDEO TYPE 1
FINISH	CHK	3/16/77	LSB OF 8BIT DAC OR 4BIT DAC
NEXT ASSY	ENG	3/16/77	SIZE D SCALE — SHT 6 OF 11 DWG NO. 502448 REV F
	APVD	4/16/77	

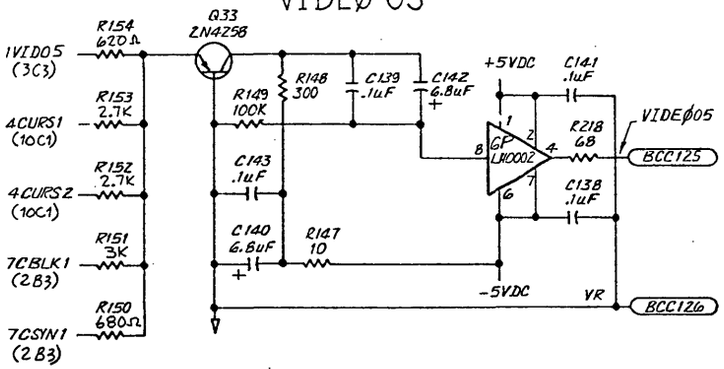
PIV-000572


REVISIONS
SEE SHT. 1
+5VDC ANALOG 2
R9 10Ω
+5VDC
VIDII (4C4)
E23 E24 VDAC83
331
5VDAC83
VDAC83
E46
CR2 IN4444
R8 270Ω
R7 510Ω
Q4 2N4258
6VDAC83
R6 270Ω
Q3 2N3646
R5 510Ω
C68 .1μf
C66 .1μf
R19 2KΩ
DAC3
C71 47PF
R10 3.3K
ANALOG
C73 .1μf
IP LH0002
C72 .1μf
-5VDC ANALOG
DACVID3 TO VIDEO AMP SHT 9
-5VDC
L6 100μh
C21 1.0μF
C42 1.0μF
-5VDC ANALOG
ANALOG
VIDIO (4C4)
E21 E22 VDAC43
5VDAC43
VDAC43
E45
CR1 IN4444
R4 270Ω
R3 510Ω
Q2 2N4258
6VDAC43
R2 270Ω
Q1 2N3646
R1 510Ω
C70 .1μf
R20 1KΩ
R23 2KΩ
L1
BCC148
BCW148
100μh
C1 .1μf
C2 .1μf
C3 33μf
+5VDC ANALOG 1
ANALOG
VID09 (4C4)
E25 E26 VDAC23
5VDAC23
VDAC23
E47
CR4 IN4444
R17 510Ω
R18 270Ω
Q8 2N4258
6VDAC23
R16 270Ω
Q7 2N3646
R15 510Ω
C67 .1μf
R22 1KΩ
R21 2KΩ
+5VDC ANALOG 1
L3
100μh
C7 .1μf
C8 .1μf
C9 33μf
+5VDC ANALOG 2
ANALOG
L4
100μh
C10 .1μf
C11 .1μf
C12 33μf
+5VDC ANALOG 3
ANALOG
VID08 (4C4)
E27 E28 VDAC13
5VDAC13
3VQFUL1
VDAC13
E48
CR3 IN4444
R13 510Ω
R14 270Ω
Q6 2N4258
6VDAC13
R12 270Ω
Q5 2N3646
R11 510Ω
C69 .1μf
R24 1KΩ
R25 2KΩ
R27 2KΩ
ANALOG
L5
100μh
C13 .1μf
C14 .1μf
C15 33μf
+5VDC ANALOG 4
ANALOG
ACC03
ACW03
ACC04
ACW04
C4 .1μf
C5 .1μf
C6 33μF
C16 THRU C49 .1μF
+5V
DIGITAL
ACC01
ACW01
ACC02
ACW02
DIGITAL
ACC07
ACW07
-5VDC DIGITAL
BCC149
BCW149
BCC150
BCW150
ANALOG
BCC131
BCW131
BCC149
BCW149
BCC150
BCW150
ANALOG
ACC08
ACW08
ACC09
ACW09
+12V
ACC05
ACW05
ACC06
ACW06
-12V
QTY REQD
PART NO
ITEM DESCRIPTION
ITEM NO.
LIST OF MATERIALS
TOLERANCES
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.
.XX .XXX ANG
MATL
FINISH
NEXT ASSY
DO NOT SCALE DRAWING
SIGNATURES
DATE
DWN M. Cohen 2/11/72
CHK 2/11/72
ENG 2/11/72
APVD 2/11/72
RAMTEK
LOGIC DIAG.-VIDEO TYPE I 4-BIT DACS
SIZE D
SCALE —
DWG NO. 502448
REV F
SHT 10 OF 11

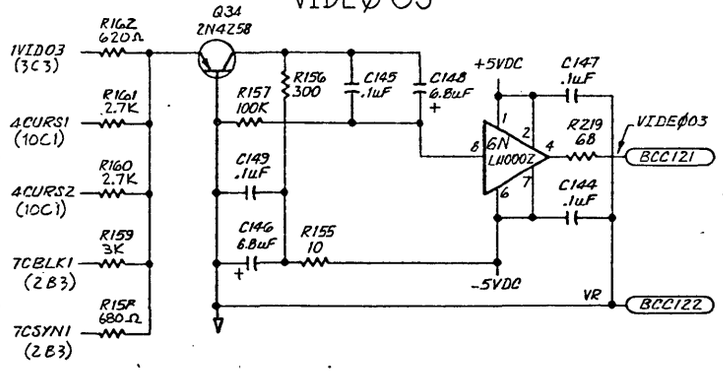
PIV-000573

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT 1		

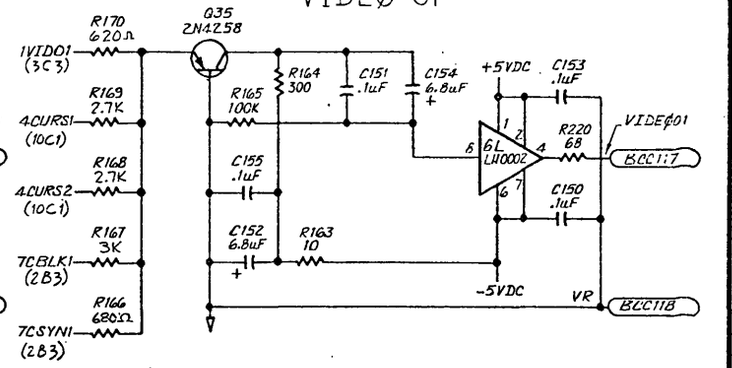
VIDEØ 05



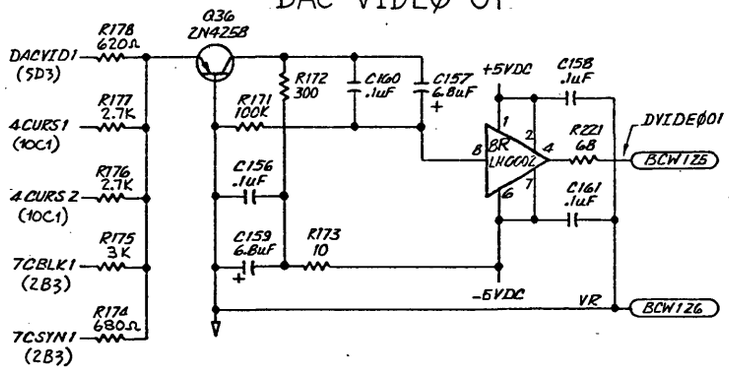
VIDEØ 03



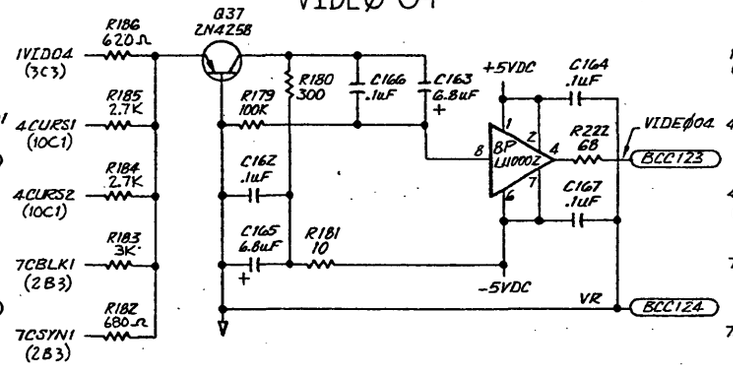
VIDEØ 01



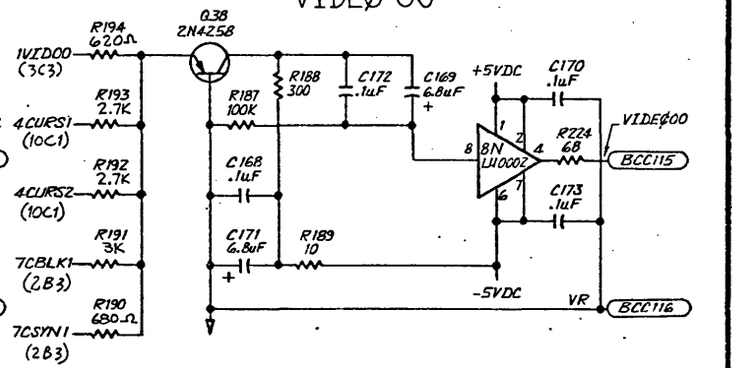
DAC VIDEØ 01



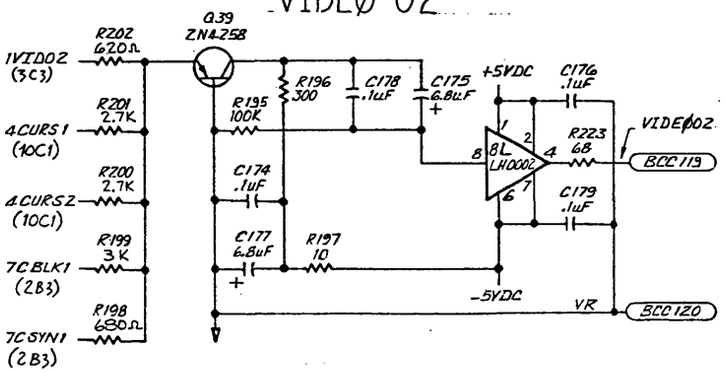
VIDEØ 04



VIDEØ 00



VIDEØ 02



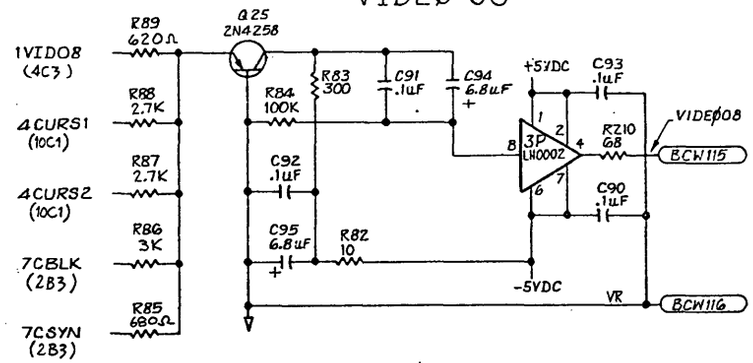
QTY PER	PART NO.	ITEM DESCRIPTION	ITEM NO

LIST OF MATERIALS

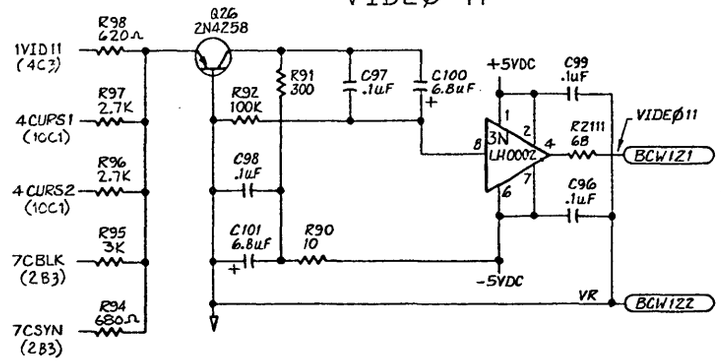
TOLERANCES	DO NOT SCALE DRAWING			RAMTEK
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.	SIGNATURES		DATE	
.XX .XXX ANG	DWN		2-9-76	TITLE: SCHEMATIC DIAGRAM
MATL —	CHK			VIDEO TYPE 1
FINISH —	ENG			SIZE: D SCALE — DWG NO. 502443 REV F
NEXT ASSY	APVD		7/6/76	SHT 2 OF 11

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT 1		

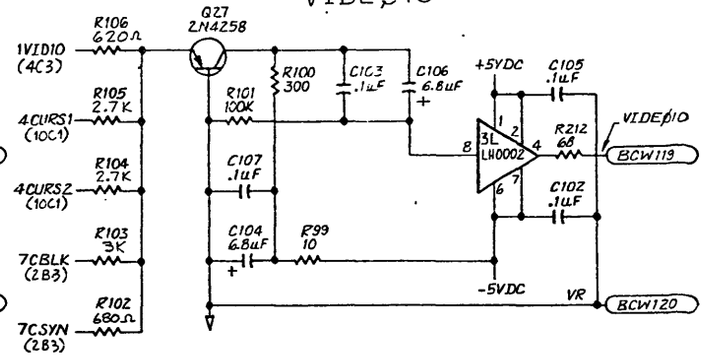
VIDEØ 08



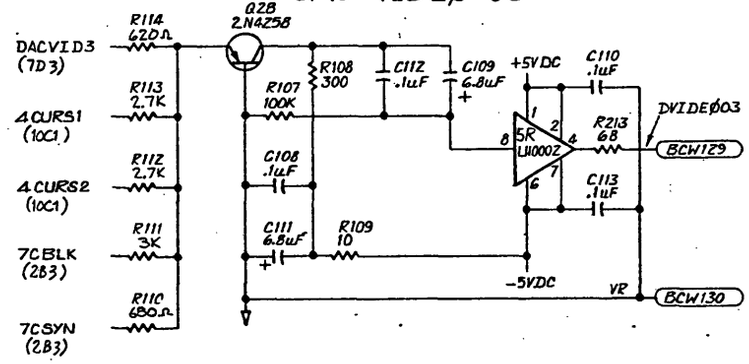
VIDEØ 11



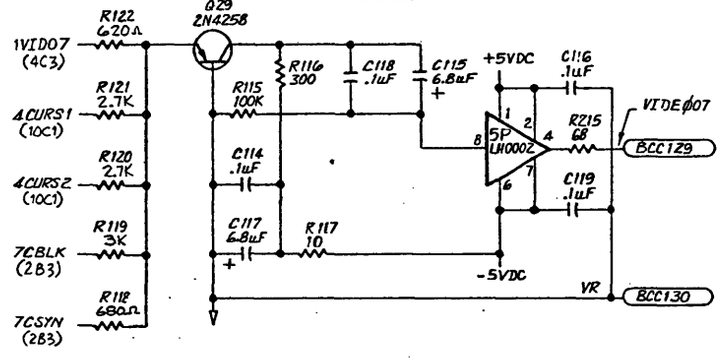
VIDEØ 10



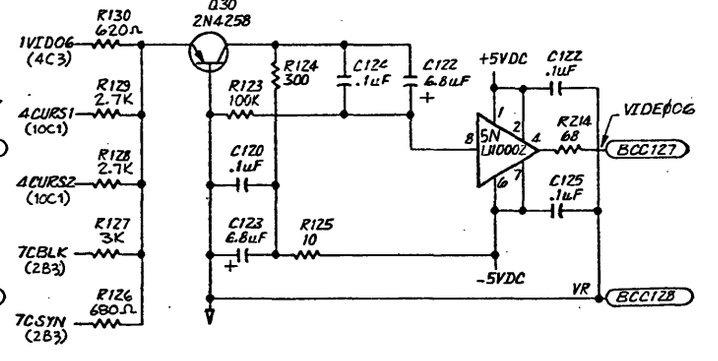
DAC VIDEØ 03



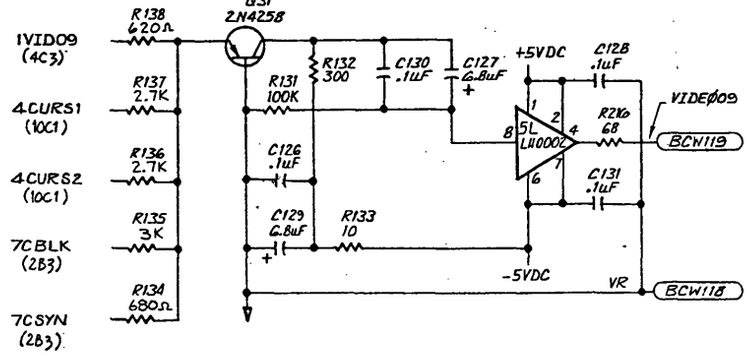
VIDEØ 07



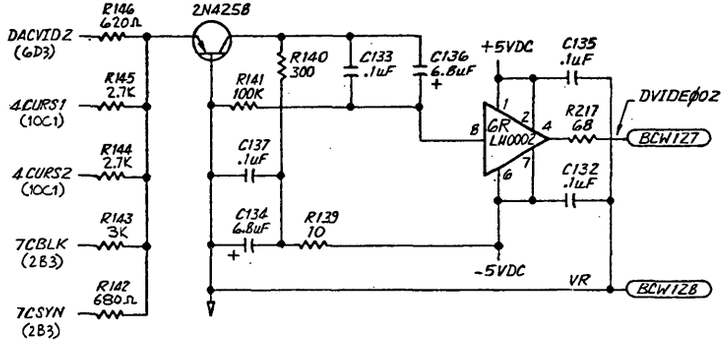
VIDEØ 06



VIDEØ 09



DAC VIDEØ 02



QTY REQD	PART NO	ITEM DESCRIPTION	ITEM NO

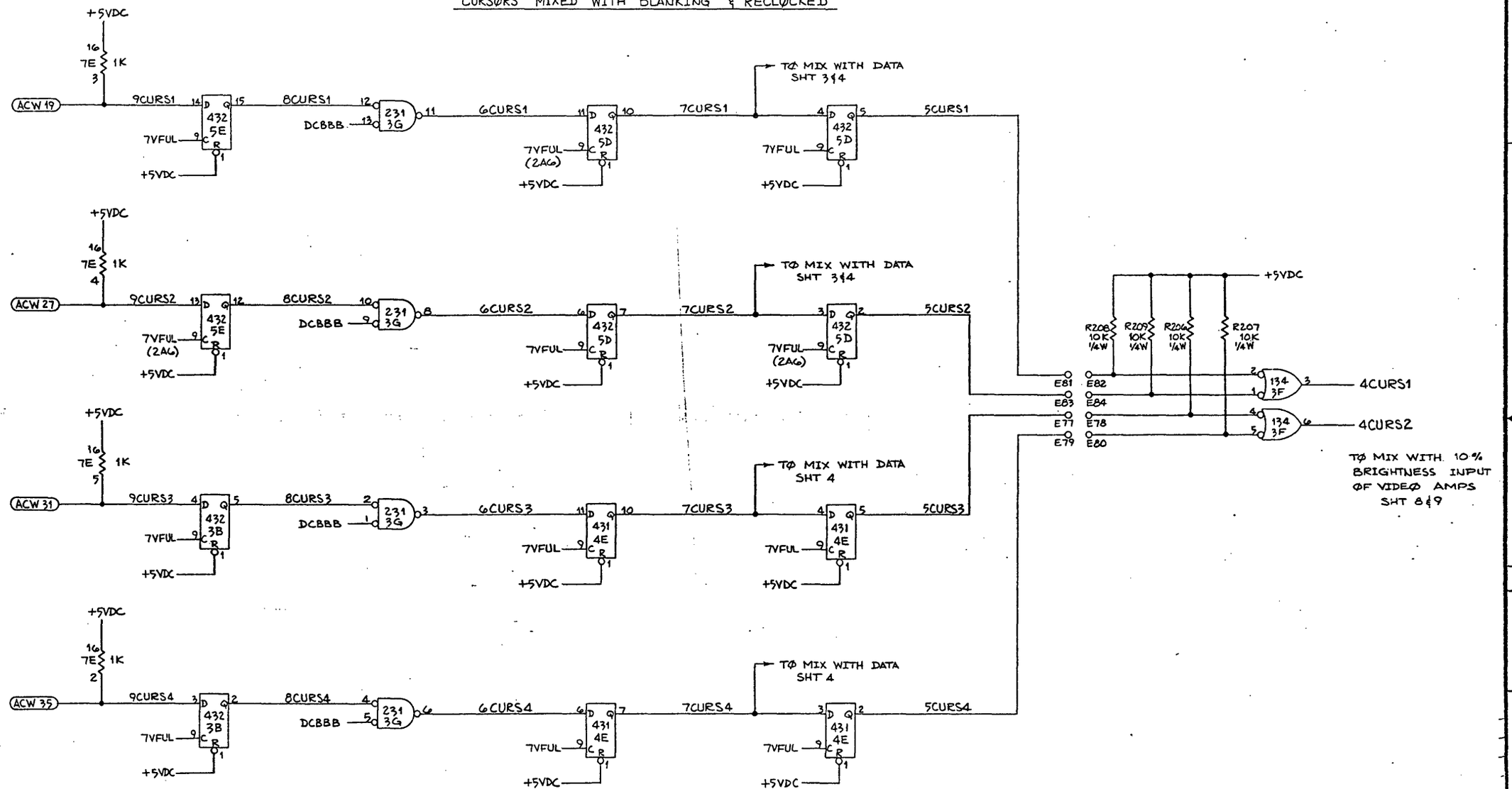
LIST OF MATERIALS

TOLERANCES	DO NOT SCALE DRAWING		
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.	SIGNATURES	DATE	RAMTEK
.XX .XXX ANG	DWN	7-9-76	TITLE: SCHEMATIC DIAGRAM VIDEO TYPE 1
MATL —	CHK		
FINISH —	ENG		SIZE D SCALE — DWG NO. 502448 REV F
NEXT ASSY	APVD	7/8/76	SHT 2 OF 11

PIV-000575

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT 1		

CURSØRS MIXED WITH BLANKING & RECLØCKED



QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.

LIST OF MATERIALS

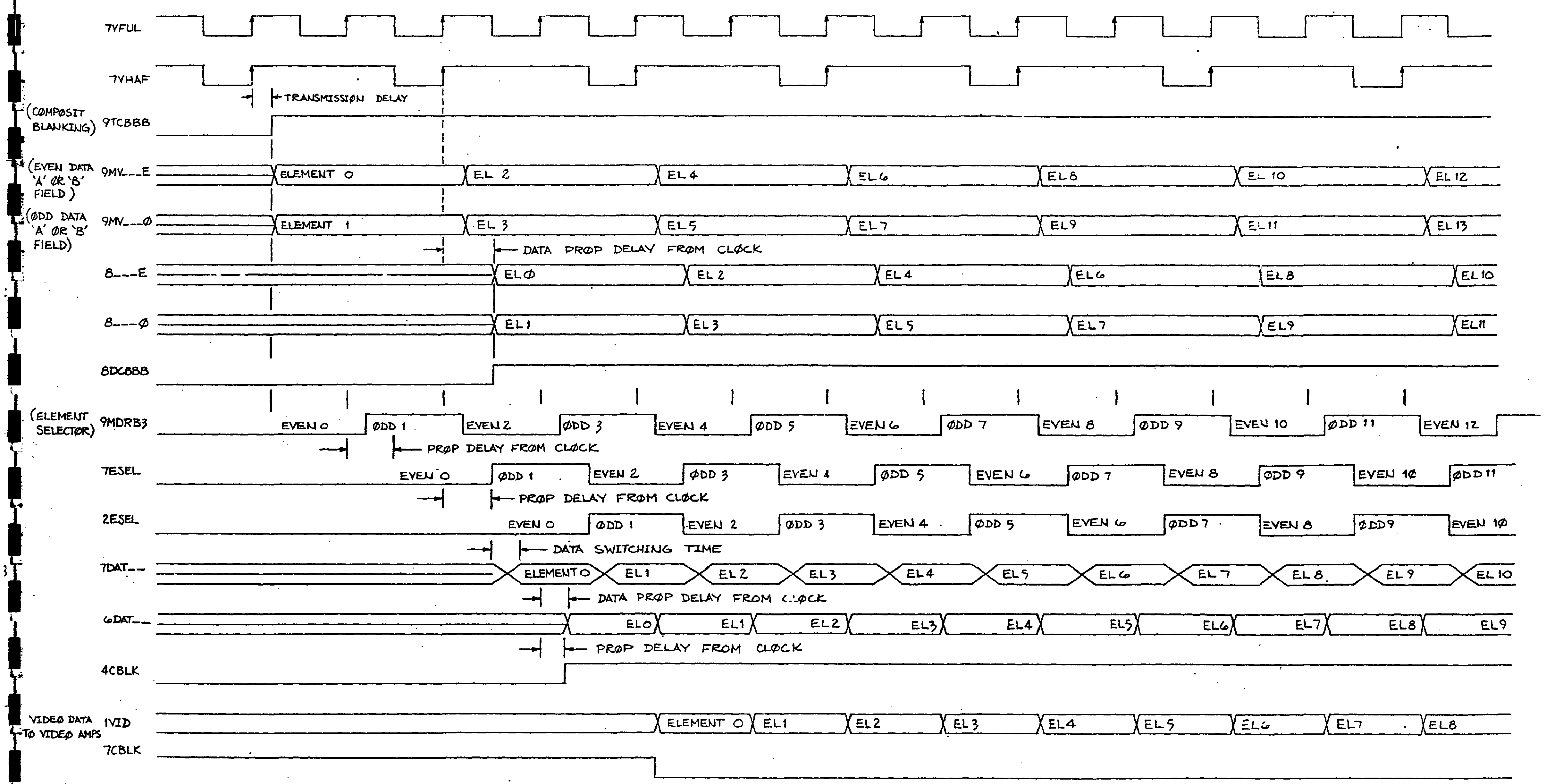
TOLERANCES	DO NOT SCALE DRAWING		RAMTEK	
UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES.	SIGNATURES	DATE		
.XX .XXX ANG	DWN	8-31-76	TITLE VIDEØ 1 LØGIC DIAGRAM	
MATL	CHK		CURSØRS	
FINISH	ENG		SIZE D	DWG NO. 502448 REV F
NEXT ASSY	APVD		SCALE	SHT 5 OF 11

501003 REV: N/C

PIV-000576

REVISIONS			
LTR	DESCRIPTION	APVD	DATE
	SEE SHT 1		

MEDIUM AND HIGH RESOLUTION



QTY REQD	PART NO.	ITEM DESCRIPTION	ITEM NO.
LIST OF MATERIALS			

TOLERANCES	UNLESS OTHERWISE SPECIFIED REMOVE ALL BURRS & BREAK ALL SHARP EDGES. .XX .XXX ANG
MATL	
FINISH	
NEXT ASSY	
DO NOT SCALE DRAWING	
SIGNATURES / DATE	DWN 9-5-76; CHK; ENG; APVD
	RAMTEK
TITLE	VIDEO TYPE 1 IMMAGER TIMING DIAGRAM
SIZE	D
SCALE	
SHT	11 OF 11
DWG NO.	502448
REV	

501003 REV: N/C

PIV-000577