A Low Component Count Video Data Terminal Using the DP8350 CRT Controller and the 8080 CPU

INTRODUCTION

The DP8350 is an I²L—LS technology integrated circuit, designed to provide all control signals for a cathode ray tube (CRT) display system. This application note explains a system using the DP8350 and the 8080 microprocessor. The design philosophy shows how the DP8350 interfaces to the 8080, completing the function of a video data terminal with a minimum component count. After reading and understanding this application note the reader will realize the ease and flexibility of designing video terminals with the DP8350*. To thoroughly understand this application note the reader must be familiar with the DP8350 and the 8080 microprocessor. The video data terminal described is divided into the following sections, (*Figure 1*).

The DP8350 CRT controller (CRTC).

The 8080 μP system which includes ROM, RAM, interrupt instruction port, oscillator, and control support chips.

The character generator.

The communication element.

The keyboard and baud rate select ports.

THE CRTC

The DP8350 generates all the required control and timing signals for displaying video information on the video monitor. Here is a summary of the controller's functions:

Dot clock, control, and counter outputs for the character generator.

Bidirectional RAM address refresh counter for refreshing the video RAM and allowing microprocessor loading to the internal DP8350 registers.

Direct drive horizontal and vertical sync signal outputs.

Direct cursor address location output. The cursor is internally delayed or pipelined, allowing for the access time of video RAM and the character generator ROM, (*Figure 1*).

THE CPU

The microprocessor provides CRTC, operator, and external machine control for the system. When the CRT controller is not actively refreshing the video RAM, (i.e., during vertical retrace or blank scan lines), the microprocessor is enabled for system housekeeping, *(Figure 2)*. This method of multiplexing the RAM with the CPU and the CRTC eliminates the need for line buffers.

THE CHARACTER GENERATOR

The character generator consists of 3 elements: an address latch to hold the input address to the character ROM allow-*The DP8350 is equivalent to the INS8276. National Semiconductor Application Note 199 Al Brilliot July 1986



sing the DP8350 CRT

Low Component Count Video Data Termina

Controller and the 8080 CPU

ing for the access time of the ROM; the character ROM that stores the ASCII character in a form for parallel to serial conversion by the shift register; the shift register converts the character ROMs parallel output to serial form. The serial output from the shift register is the true video output, modulating the video monitors electron beam which writes characters on the screen.

THE COMMUNICATION ELEMENT

The INS8250 is the asynchronous communication element (ACE) for the data terminal. The ACE allows the CPU portion of the data terminal communication with peripherals or host computers at the correct baud rate, (*Figure 1*). The ACE is programmed by the CPU to send and receive serial data at the standard baud rates from 110 to 4800 baud. The ACE, in conjunction with the DS1488 and DS1489 line drivers and receivers, also provides full RS-232C synchronous communication if higher baud rates are desired. System communication speed must always be considered to insure the baud rate does not exceed the time required for the CPU to process a data byte. Asynchronous communication at baud rates higher than 4800 are possible by adding a line buffer.

SYSTEM INITIALIZATION

Application of the terminal's power supply resets the microprocessor, the communication element, and the CRT controller. Resetting the ACE is necessary to clear the interrupt. Resetting the CRTC is not absolutely necessary since the microprocessor loads the cursor and top of page registers in the initialization routine.

Following the reset all interrupts are disabled to avoid unwanted interrupts from the CRTC, ACE, or I/O ports. Refer to the initialization routine in the flowchart.

The stack pointer is loaded to the bottom of scratch pad RAM (3FFFH) for use as the register save pointer, *(Figure 3)*.

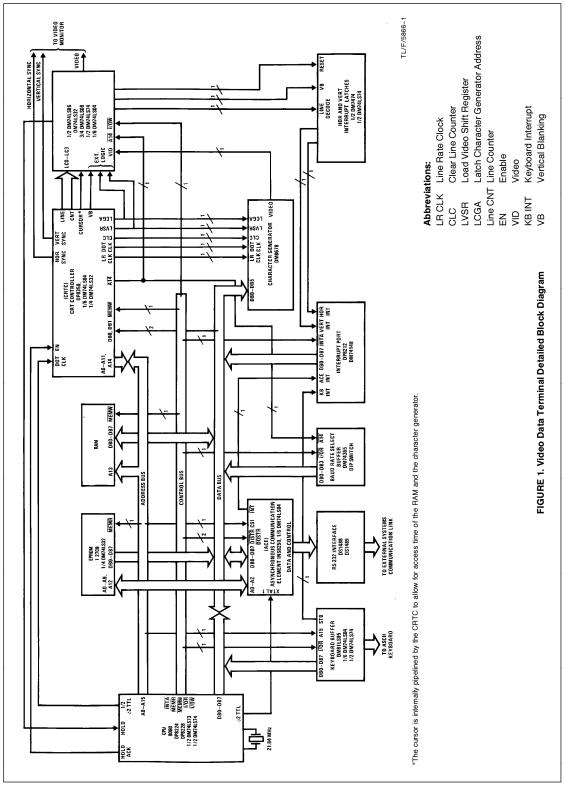
The entire RAM is written with ASCII spaces generating a cleared screen. After completion of the screen clear loop the CPU writes 000H to the cursor and the top of page registers in the DP8350 CRTC. The routine homes the cursor to the upper left corner of the screen. The top of the page register was loaded with 000H, therefore, the video RAM is refreshed by the CRTC from that starting address to the last address on the screen of video RAM (1920 characters).

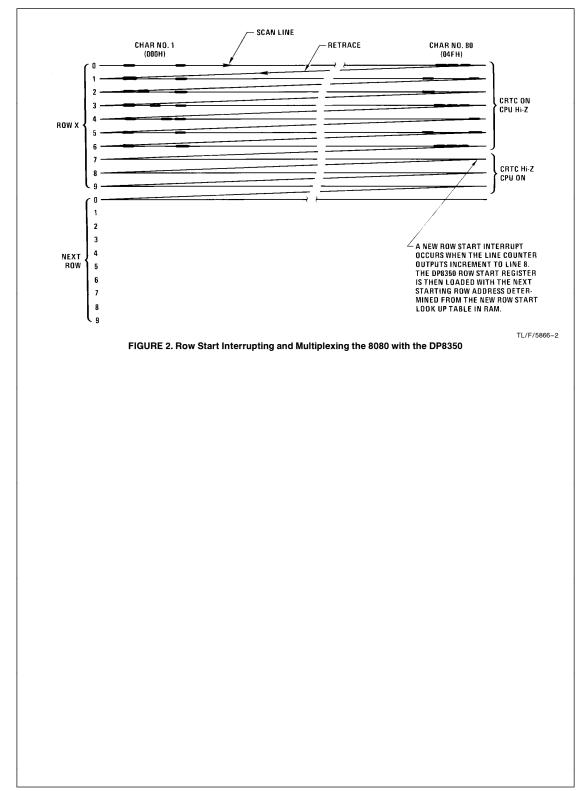
AN-19

TRI-STATE®is a registered trademark of National Semiconductor Corp.

© 1995 National Semiconductor Corporation TL/F/5866

RRD-B30M105/Printed in U. S. A





The CPU is ready to perform the communication element (ACE) load routine. First, the baud rate divisor for the ACE must be determined. The baud rate select switch is read providing a code which corresponds to the appropriate 16-bit divisor for the ACE. This divisor determines the baud rate at which the ACE will communicate. Any additional programming requirements needed for the ACE to communicate with host computer systems could also be done at this time. The software in this system does not contain any additional programming for the ACE. There are many programming modes related to the ACE. Details of these modes are beyond the scope of this application note.

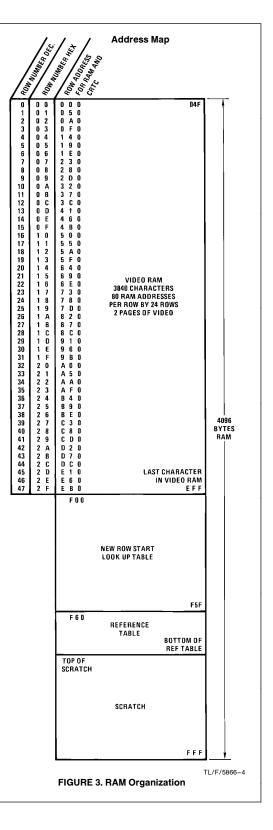
The row start look-up table, (*Figure 4*), is loaded up by a simple algorithm that loads and adds the data for referencing a row number to that row's starting address. The reference table, (*Figure 5*), is initialized next by direct loading. This table provides the CPU with top of page, bottom of page, next row load, cursor row, and scratch row numbers for system housekeeping.

Finally, the new row start and vertical interrupt latches are cleared, *(Figure 6).* The register pointers are loaded and the CPU is forced in a wait loop with interrupts enabled.

NON-SEQUENTIAL ADDRESSING

The data terminal described here was designed for non-sequential starting row addressing. In many systems sequential row addressing is used. If a character row consists of 10 scan lines the RAM is addressed 10 repetitive times from 000H through 04FH, (*Figure 2*). The next row is refreshed in the same manner from 050H to 09FH. The starting row address is sequential 000H, 050H, 0A0H–EB0H for row numbers 0H, 1H, 2H, -2FH, respectively. Non-sequential row addressing would be equivalent to 050H, 000H, 0A0H–EB0H for row numbers 1H, 0H, -2FH, respectively, (*Figure 3*).

In conjunction with the CPU, non-sequential row addressing is quite easily accomplished with the DP8350 since this is one of the features designed into the part. Accomplishing this task basically requires the following sequence of events. Assume the CRTC has finished writing a video row in the middle of the monitor's screen. This system has a 5 x 7 character font in a 7 x 10 field, (Figure 2). At the completion of the last video scan line 7 the CRTC line counters continue to count the last 3 lines. Video is not present since the character is only 7 scan lines high. The blank scan lines are 7, 8, and 9 permitting the CRTC address outputs to be at TRI-STATE®, allowing the CPU to run. When the line counter outputs increment to scan line 8 an interrupt signals the CPU. The interrupt occurring is the new row start interrupt. The interrupt routine fetches the next CRTC row number from the reference table (Figure 5). This number is converted to the new starting row address, explained later, and loaded to the CRTC row start register. The CPU finishes the routine by clearing the interrupt, readying itself for the next new row start interrupt. The entire routine takes 1 scan line of time, approximately 64 us. The CRTC continues to scan the video RAM from that new starting address on for the next 7 repetitive scan lines of the next row. Many advantages become apparent using the non-sequential addressing scheme. Scrolling up or down with the cursor always on the screen may be done faster and easier from a hardware/ software standpoint. Exchanging one row with another row is fast since it is not necessary to rewrite the video RAM. Row swapping is useful for higher end terminals requiring row editing functions.



B	ow			N	IRS	Hic	ıh				NRS	Lo	w		B	ow			N	IRS	Hig	h			1	NRS	Lov	N	-
Nu		er		Bow							w		mbe	er						w					Ro				
Dec	н	ex		Address Data				Address Data		Dec Hex		Address			Data		Address		Data										
	0		0	F	0	0	2	0	3	F	0	0		0				2	E	4	0	2	7	2	E	4	0	0	(
0	0	0 1	3	F F	0 0	0	3	0 0	3	F F	3 3	0 1	0	0 0	24 25	1	8 9	3	F F	1 1	8 9	3	7 7	3	F F	4 4	8 9	8 D	(
1 2		2	3	F	0	1 2	3	0	3	F	3	2	5	0	25	1	9 A	3	F	1		3	7 8	3	F		9 A		
	0																				A					4		2	
3	0	3	3	F	0	3	3	0	3	F	3	3	F	0	27	1	B	3	F	1	В	3	8	3	F	4	В	7	1
4	0	4	3	F	0	4	3	1	3	F	3	4	4	0	28	1	С	3	F	1	С	3	8	3	F	4	С	С	1
5	0	5	3	F	0	5	3	1	3	F	3	5	9	0	29	1	D	3	F	1	D	3	9	3	F	4	D	1	1
6	0	6	3	F	0	6	3	1	3	F	3	6	E	0	30	1	E	3	F	1	E	3	9	3	F	4	E	6	
7	0	7	3	F	0	7	3	2	3	F	3	7	3	0	31	1	F	3	F	1	F	3	9	3	F	4	F	В	
8	0	8	3	F	0	8	3	2	3	F	3	8	8	0	32	2	0	3	F	2	0	3	А	3	F	5	0	0	
9	0	9	3	F	0	9	3	2	3	F	3	9	D	0	33	2	1	3	F	2	1	3	А	3	F	5	1	5	
10	0	А	3	F	0	А	3	3	3	F	3	А	2	0	34	2	2	3	F	2	2	3	А	3	F	5	2	А	
11	0	В	3	F	0	В	3	3	3	F	3	В	7	0	35	2	3	3	F	2	3	3	А	3	F	5	3	F	
12	0	С	3	F	0	С	3	3	3	F	3	С	С	0	36	2	4	3	F	2	4	3	В	3	F	5	4	4	
13	0	D	3	F	0	D	3	4	3	F	3	D	1	0	37	2	5	3	F	2	5	3	В	3	F	5	5	9	
14	0	Е	3	F	0	Е	3	4	3	F	3	Е	6	0	38	2	6	3	F	2	6	3	В	3	F	5	6	Е	(
15	0	F	3	F	0	F	3	4	3	F	3	F	В	0	39	2	7	3	F	2	7	3	С	3	F	5	7	3	(
16	1	0	3	F	1	0	3	5	3	F	4	0	0	0	40	2	8	3	F	2	8	3	С	3	F	5	8	8	(
17	1	1	3	F	1	1	3	5	3	F	4	1	5	0	41	2	9	3	F	2	9	3	С	3	F	5	9	D	(
18	1	2	3	F	1	2	3	5	3	F	4	2	A	0	42	2	A	3	F	2	A	3	D	3	F	5	A	2	(
19	1	3	3	F	1	3	3	5	3	F	4	3	F	0	43	2	В	3	F	2	В	3	D	3	F	5	В	7	(
20	1	4	3	F	1	4	3	6	3	F	4	4	4	0	44	2	С	3	F	2	С	3	D	3	F	5	С	C	(
21	1	5	3	F	1	5	3	6	3	F	4	5	9	0	45	2	D	3	F	2	D	3	E	3	F	5	D	1	(
22	1	6	3	F	1	6	3	6	3	F	4	6	E	0	46	2	E	3	F	2	E	3	E	3	F	5	E	6	ļ
23	1	7	3	F	1	7	3	7	3	F	4	7	3	0	47	2	F	3	F	2	F	3	E	3	F	5	F	В	(
									D -4		lı	nitia	lize	d		Con	nma	nd						Fun	octic	on			
_		tion				dre			Dat				nta		OL	JT		40	0		Clea					tano	d ve	tica	l
Las						F60			XY				7					~	~		inte								
808						F61			XY 00					11			8			Rea		-							
Firs						F62			XY				0		11	N		40	0		Rea		aud	rate	sei	ect	SWIT	cn	_
Cha						F63			XY				0				I	FIG	URE	E 6.	Inpu	it/C	outp	ut S	Spac	e			
CR						F64			XY	1			0																
Rov			#			F65			XY				0																
Ten						F66			XY				0																
Ten	np. :	2			3	F67			XY			0	0																
			FI	GUF	RE 5	5. Re	efer	enc	е Та	able	•																		
			Device Address						Row				١	IRS	Hig	h			1	NRS	Lo	N							
			RO							OFF					Nu	mbe	er		Add	res	s		ow		Add	Ires	s	Ro	
			RA							3FF					Dec	H	ex					Da	ata					Da	Ita
				CRTC 5000 to 5FFF ACE 9000 to 9007					32	2	0	3	F	2	0	3	А	3	F	5	0	0	(
	Į														L	-			•	-	2	l Č	<u>∧</u> ↑	ı ٽ	•	U			↑
		*Dire						useo	to i	minin	nize 1	he			Row St		aares 1 20H												2
		syste FI					nt. Add	ree	sinc	Sn	are				3XXX S														
										, -P					5,0000	5.501	5.07												

ROW LOADING DETAILS

Obtaining the next starting row address for the CRT controller is accomplished by an addressing and adding scheme from the new row start look-up table. The same scheme is used to determine any needed address, given the row number.

Figure 8 shows a row number and address taken from the new row start look-up table.

The row number is loaded from the reference table in RAM to a register. The CPU determines the starting address from the row number and stores it in a 16-bit pointer register. The higher order 4 bits contain address for the RAM or the CRT controller, (*Figure 7*).

Here are the details of how this is accomplished. Refer to the new row start interrupt in the software listing the *Figure* 8.

The CPU D-E registers are loaded to point to a row number in the reference table. The number is put in the accumulator and moved into the E register. The D-E register in this example now contains 3F20 which points to NRS HIGH ROW DATA (3A). The addressed data is moved to the accumulator and then to the H register. If it was desired to point to the CRTC then 20H would have been added to it first. The D-E register still contains 3F20H. To obtain the NRS LOW ROW DATA the E register is moved to the accumulator and 30H is added to it. Now the D-E register contains 3F50H and points to NRS LOW ROW DATA (00H). The data is loaded to the accumulator and then to the L register. The H-L registers contain 3A00H which is the starting row address for row number 20H. The method just described is used throughout the terminals program to move the cursor, load the top of page, and load the new starting row address in the CRTC.

VERTICAL INTERRUPT

TABLES

FIRST ROW

VINEN

The vertical interrupt occurs when the CRTC has completed refreshing a video page (1920 characters) of information. Vertical blanking identifies that condition and interrupts the CPU forcing it to the vertical interrupt routine. Refer to the vertical interrupt in the flow chart. The routine moves the first row number to the CRTC row number, updating it so the next new row start load occurs with the top of the page address or the first row of the video screen.

4096 BYTES Of RAM

RAM

SCROLL UP

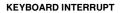
SCROLL DOWN

1920 CHARACTERS VIDEO SCREEN

> THE CPU KEEPS THE CURSOR

AI WAYS ON THE

IDEO SCREEN



The external keyboard requirements are ASCII outputs with a suitable strobe to interrupt the CPU for keyboard servicing. Refer to the keyboard interrupt in the flow chart. After the keyboard buffer is read the data byte is tested for a (CNTL E), new baud rate command. If the test fails the CPU writes the data byte to the ACE. Passing the test forces the CPU to read the baud switch and load the ACE with the new baud rate.

ACE INTERRUPT

As mentioned above, a data byte read from the keyboard that is not a baud rate command enters the accumulator. The CPU writes the data byte from the accumulator to the transmitter holding register in the ACE. The ACE proceeds to shift out the data byte, with the appropriate start and stop bits, serially from the (SOUT) output. The data is shifted to the serial input (SIN) of the ACE and loaded into the receiver holding register. When the register is full the ACE interrupts the CPU, initializating the ACE service routine. Refer to the ACE interrupt in the flow chart.

The CPU reads the receiver holding register in the ACE. Reading the ACE resets the interrupt. The data byte now resides in the accumulator. The CPU tests for a control or an escape function. The function is executed if test conditions are met. Refer to the keyboard interrupt routine in the software listing. The data byte is written to the video RAM at the cursor address which appears on the monitor screen. The cursor and character numbers are incremented as long as it is not at the end of a row. A character at the end of a row requires further testing to recognize the following situations. Is it the last row on the monitor's screen? Or is it on the maximum row of the video RAM? Essentially, the cursor is forced to stay visible on the video monitor's screen and video RAM is always kept out of scratch pad RAM, *(Figure 9)*.

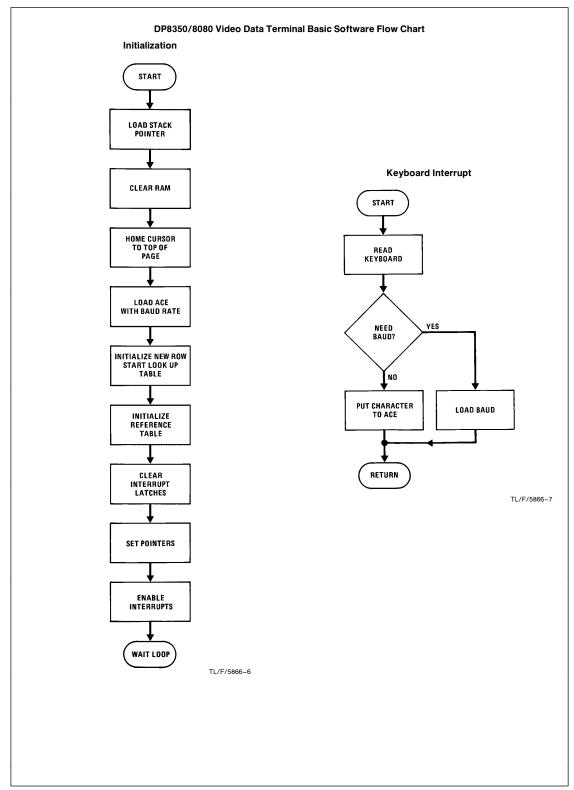
FULL/HALF DUPLEX OPERATION

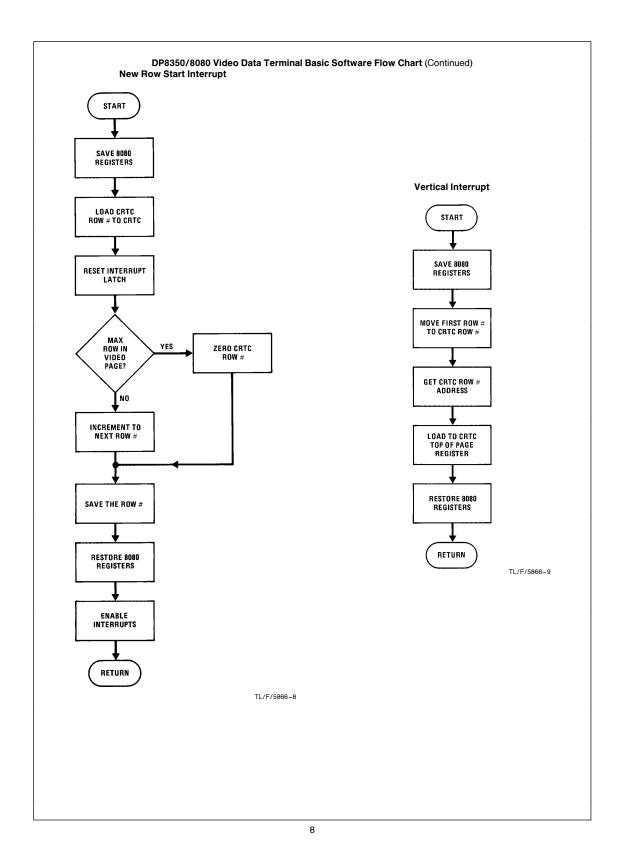
The data terminal and a host computer in the full duplex mode of operation would receive the serial information, process it, and send it back to the SIN input of ACE. Using the terminal in a stand-alone mode for testing, the serial out SOUT is tied to the serial in SIN of the ACE. In the half duplex mode a data byte is sent to the host computer at the same time it is sent to the terminal. When the data terminal is set up to communicate with a host computer the full duplex mode of operation is desirable.

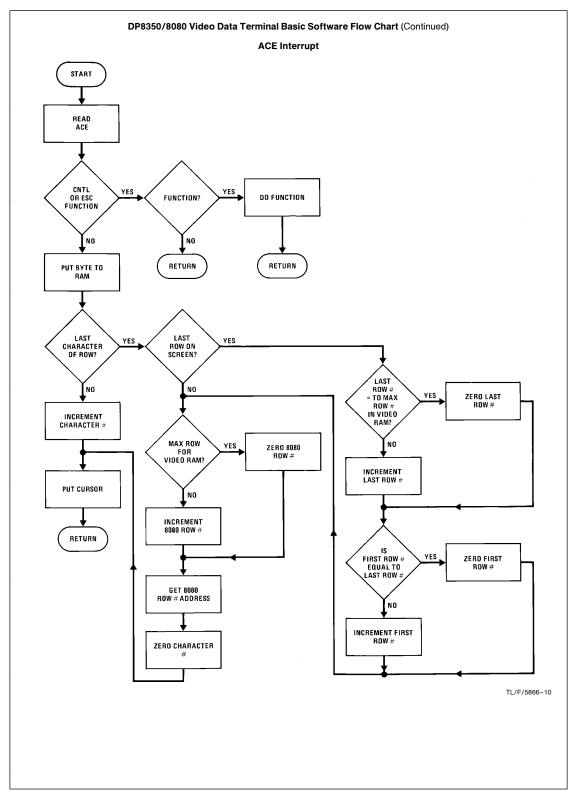
The video screen is allowed to scroll only through the video RAM (000H to EFFH). The CPU keeps the video screen within these bounds by loading the new row start register with that address range only (row 00H to 2FH).

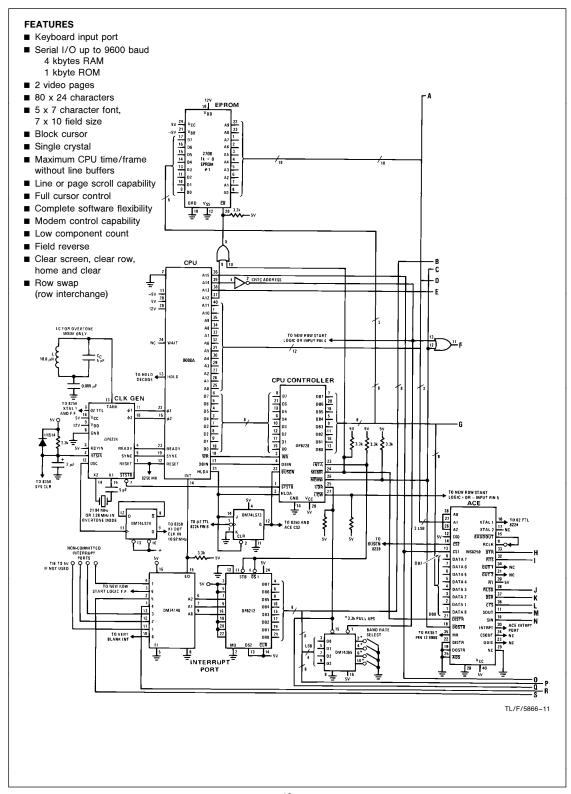
FIGURE 9. Drum Analogy for the RAM

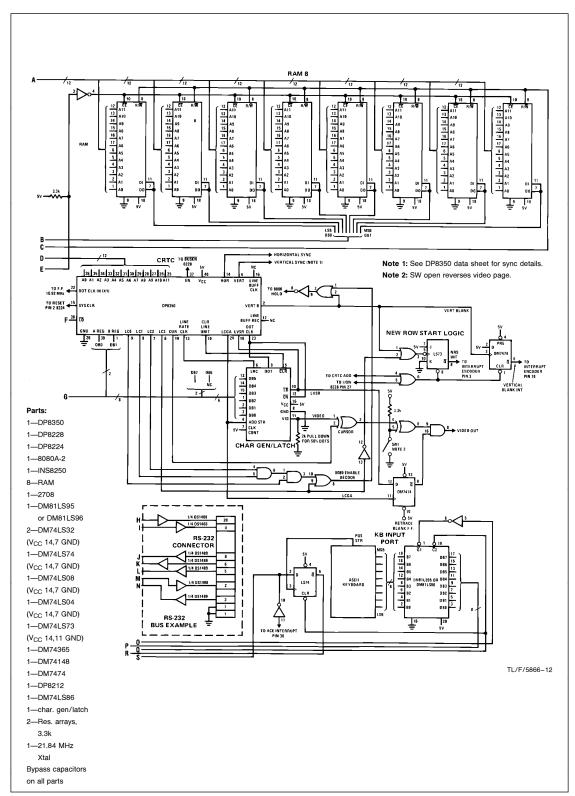












2 3 4				MICONDUCTOR 13 RAMMABLE CRT	CONTROLLER BOARD **	
5 6 7				IM TROUTNER		
8 9 10 11 12 13 14	0060 0061 0062 0063 0064 0065 0066 0066 0067 0068		LASTROW ROW8080 FIRSTRO CHARNUM CRTCROW ROWSAVE TEMP1 TEMP2 IMASK	= = = =	060 061 062 063 064 065 066 066 067 068	
18 0000 19 0000 20 0001	31FF3F	START:	DI LXI	SP, O3FFF	DISABLE INTERRUPTS LOAD STACK POINTER	
21 0004 22 0007 23 0008 24 000B			JMP . =0008 JMP . =0010	INIT NEWRO	;JUMP TO INITILIZE ROUTINE ;NEW ROW START INTERRUPT	
25 0010 26 0013			JMP . =0018	INTACE	ACE INTERUPT	
27 0018 28 001B 29 0038 30 003B 31 003E 32 0040 33 0042 34 0043 35 0044	C34F02 210030 0E20 3E3F 71 23	INIT: CLRAM:	JMP .=0038 JMP LXI MVI MVI MOV INX CMP	INTKB VERTI H.03000 C.020 A.03F M.C H H	;KEYBOARD INTERUPT ;VERTICAL INTERUPT ;IST RAM ADDRESS ;ASCII SPACE INTO C REG ;MAX RAM ADDRESS ;ASCII SPACE INTO MEM ;NEXT RAM ADDRESS ;MAX RAM ADDRESS?	
36 0045 37 0048 38 0040 39 0040 40 0040 41 004E 42 004F	C24200 0E00 3E40 71 23 BC C24C00	CLRAM1:	JNZ MVI MOV INX CMP JNZ	CLRAM C,000 A,040 M,C H H CLRAM1	;IF NO THEN NEXT ADD.	
43 0052 44 0055 45			CALL	HMCUR BAUD	;GO TO CUR HOME ROUTINE ;GO TO BAUD LOAD ROUTINE	
16 17 18 0058	21003E		; NEW RO	N START LOOK	UP TABLE GENERATION	
49 005B 50 005E 51 0061 52 0062 53 0063 54 0064 55 0066 56 0067 57 0068 58 0064 59 006B 50 006C 51 0062 53 0070	010030 70 79 12 C650 4F 78 CE00 47 2C 47 2C 1C 7B FE60	NRS:	LXI LXI MOV STAX ADI MOV ACI MOV ACI INR INR INR INR INR INR JNZ	D, 03F30 B, 03000 M, B A, C D C50 C, A A, B C, A A, B E E E LASTROW NRS	<pre>, R.S. LOW ADDRESS ; N.R.S. ADDRESS DATA ;STORE TO N.R.S. DATA TABLE '' ;N.R.S. DATA LOW TO ACC ;STORE TO N.R.S. DATA TABLE L ;ACC READY FOR NEXT LOAD ;ACC TO N.R.S. DATA HIGH ;N.R.S. DATA TO ACC ;ADD CARRY BIT TO DATA HIGH ;MOVE RESULT TO N.R.S. DATA H ;INCREMENT N.R.S. HIGH ADD ;INCREMENT N.R.S. LOW ADD ;N.R.S. ADD LOW TO ACC ;MAX TABLE ADDRESS ; IF FALSE JUMP</pre>	
5			; REFERE	NCE TABLE IN	TILIZE	
57 0073 58 0075 59			MVI STAX	A, 017 D	;LAST ROW NUMBER TO ACC. ;STORE TO REFERENCE TABLE	
70 71 72 0076	D340		CLEAR I	PERIPHERAL II	ITERRUPT FLOPS	
2 0078 73 0078 74 75 76			IN	080 POINTERS	KEYBOARD INTERRUPT CLEAR	
77 007A 78 007D 79 0080	210030		LXI LXI	D, 03F60 H, 03000 B, 09000	>POINT D-E TO REFERENCE TABLE >POINT H-L TO 1ST RAM LOCATI∩ >POINT B-C TO ACE	
31 82			;WAIT L	DOP FOR INTE	RUPTS	
83 0083 84 0084 85	FB C38300	BACK:	EI JMP	BACK	;ENABLE INTERRUPTS ;LOOP UNTIL INTERRUPTED	
36 37 38 0087 39 008A 90 008C 91 008D 92 008E 93 008F 74 0092	3E02 77 3C 77 210030	HMCUR:	,HOME U LXI MVI MOV INR MOV LXI RET	P CURSOR H, 05000 A, 002 M, A A M, A H, 03000	,POINT B-C TO CRTC ,T.O.P. REGISTER SELECT ,T.O.P. LOAD ,CURSOR REGISTER SELECT ;CURSOR LOADS TO T O.P. ;POINT H-L TO IST RAM ADD. ,RETURN	
č					Continued Next Page	TL/F/5866-13
					Continued Next 1 age	

97 98 0093 D5 BAUD: PUSH D 99 0094 DE40 IN 040 100 0094 DE40 IN 040 101 0098 FE00 CPI 000 102 0098 CAD400 JZ B110 103 0095 FE01 CPI 001	;SAVE D-E REGISTERS ;READ BAUD SELECT CODE ;ZERO THE HIGH ORDER 4 BITS
100 0096 E60F ANI 00F 101 0098 FE00 CFI 000 102 009A CAD400 JZ B110	
101 0098 FE00 CPI 000 102 009A CAD400 JZ B110	
	;110 BAUD ROUTINE
104 009F CADA00 JZ B150	;150 BAUD ROUTINE
105 00A2 FE02 CPI 002 106 00A4 CAE000 JZ B300	; 300 BAUD ROUTINE
107 00A7 FE03 CPI 003	, SOO EMEE ROOTINE
108 00A9 CAE600 JZ B600	,600 BAUD ROUTINE
109 00AC FE04 CPI 004 110 00AE CAECOO JZ B1200	1200 BAUD ROUTINE
111 00B1 FE05 CPI 005	
112 00B3 CAF200 JZ B1800 113 00B6 FE06 CFI 006	;1800 BAUD ROUTINE
114 00B8 CAF800 JZ B2000	;2000 BAUD ROUTINE
115 00BB FE07 CPI 007 116 00BD CAFE00 JZ B2400	2400 BAUD ROUTINE
117 00C0 FE08 CPI 008	
118 00C2 CA0401 JZ B3600 119 00C5 FE09 CPI 009	3600 BAUD ROUTINE
119 0005 FE09 CPI 009 120 0007 CA0A01 JZ B4800	4800 BAUD ROUTINE
121 00CA FEOA CPI 00A	
122 00CC CA1001 JZ B7200 123 00CF FE0B CPI 00B	7200 BAUD ROUTINE
124 00D1 CA1601 JZ B9600	9600 BAUD ROUTINE
125	
126 ; BAUD RATE SET UP ROU 127	JIINES .
128 00D4 116305 B110: LXI D,00563	,110 BAUD DIVISOR
129 00D7 031001 JMP ACELD 130 00DA 11F303 B150: LXI D,003F3	;50 TO ACE LOAD ROUTINE ;150 BAUD DIVISOR
130 00DA 11F303 B150: LXI D,003F3 131 00DD C31C01 JMP ACELD	1150 BHOD DIVISOR
132 00E0 11F901 B300: LXI D/001F9	;300 BAUD DIVISOR
133 00E3 031001 UMP ACELD 134 00E6 11F000 B600: LXI D,000F0	;600 BAUD DIVISOR
135 00E9 C31C01 JMP ACELD	
136 00EC 117E00 B1200: LXI D,0007E 137 00EF C31C01 JMP ACELD	,1200 BAUD DIVISOR
137 00EF C31C01 JMP ACELD 138 00F2 115400 B1800: LXI D,00054	; 1800 BAUB DIVISOR
139 00F5 C31C01 JMP ACELD	
140 00FS 114C00 B2000: LXI D,0004C 141 00FB C31C01 JMP ACELD	;2000 BAUD DIVISOR
142 OOFE 113FOO B2400: LXI D,0003F	;2400 BAUD BIVISOR
143 0101 C31C01 JMP ACELD	
144 0104 112A00 B3600: LXI B,0002A 145 0107 C31C01 JMP ACELB	; 3600 BAUD DIVISOR
146 010A 112000 B4800: LXI B,00020	;4800 BAUD DIVISOR
147 010D C31C01 JMP ACELD 148 0110 111500 B7200: LXI D,00015	7200 BAUD DIVISOR
148 0110 111500 B7200: LXI D,00015 149 0113 C31C01 JMP ACELD	,7200 BHOD DIVISOR
150 0116 111000 B9600: LXI D,00010	,9600 BAUD DIVISOR
151 0119 C31C01 JMP ACELD 152	
153 ACE LOAD ROUTINE	
154 155 0110 010390 ACELD: LXI B,09003	POINT B C TO ACE
156 011F 3E83 MVI A, 083	; INIT BAUD LOAD - 8 BITS
157 0121 02 STAX B	DO INIT BAUD LOAD
158 0122 0E01 MVI C,001 159 0124 7A MOV A,D	;POINT TO BAUD HIGH ;GET BAUD HIGH
160 0125 02 STAX B	STORE BAUD HIGH TO ACE
161 0126 0E00 MVI C,000	POINT ACE TO BAUD LOW
162 0128 7B MOV A, E 163 0129 02 STAX B	;GET BAUD LOW ;STORE BAUD LOW TO ACE
164 012A 0E03 MVI C,003	RESET DLAB TO ZERO
165 012C 79 MOV A, C	; INIT ACE T/R
166 012D 02 STAX B 167 012E 0E01 MVI C,001	FUT TO ACE INTERRUPT ENABLE REG
168 0130 79 MOV A, C	SELECT RECEIVED DATA INTERF"
169 0131 02 STAX B	LOAD IT
170 0132 0E00 MVI C,000 171 0134 D1 POP D	;RESTORE B-C ACE POINTER ;RESTORE D-E REGISTERS
172 0135 C9 RET	; RETURN
173 174	
174 ;KEYBOARD INTERRUPT F 175	NUU INC
176 0136 DBS0 INTKB: IN 080	READ KEYBOARD
177 0138 FB EI 178 0139 FE05 CPI 005	;ENABLE INTERRUPTS ;NEED BAUD RATE? (CNTL E)
179 0138 CA9300 JZ BAUD	IF YES GO TO BAUD ROUTINE
180 013E FE12 CPI 012	; INVERT NEXT CNTL R
181 0140 CA4803 JZ IVERTN 182 0143 FE13 CPI 013	; INVERT ROW CNTL S
183 0145 CA5403 JZ IVERTR	
184 0148 02 STAX B	STORE BYTE TO ACE
185 0149 C9 RET 186	; RETURN
	TL/F Continued Next Page

188 189 014A (A INTA	CE: LDAX	в	LOAD ACE DATA BYTE TO ACC.
190 014B F 191 014C F	в	EI CPI	07E	ENABLE INTERRUPTS
192 014E 0	A7001	JZ	FUNC	; TEST FOR ESC COMAMD
193 0151 F 194 0153 0		CPI JZ	07F	;TEST FOR DEL COMAND
195 0156 5	5F	MOV	FUNC E, A	SAVE CHAR IN REG.E
196 0157 E		ANI	060	; MASK OUT BITS FOR CNTL TEST
198 0150 3	3A683F	JZ LDA	FUNC 03F68	;IF ZERO JMP TO CNTL FUNC ;LOAD INVERT MASK
199 015F E 200 0160 7	33	ORA	E	;OR MASK AND CHAR
200 0180 /	· /	MOV	M, A	STORE DATA BYTE TO RAM
202 203		; ADVA	NCE CURSOR	
204 0161 1		R: MVI	E, CHARNUM	; POINT B-C TO CHAR #
205 0163 1 206 0164 2		LDAX INX	D H	;LOAD CHAR # TO ACC.
200 0164 2 207 0165 P		CPI	04F	<pre>>NEXT CHAR LOCATION >LAST CHAR OF ROW?</pre>
208 0167 0		JZ	NXRO	FIF TRUE JUMP TO NEXT ROW
209 016A 0 210 016C 1		ADI STAX	001 D	;INCREMENT CHAR # ;STORE CHAR # TO RAM REF.
211 016D 0 212	3B301	JMP	PCUR	PUT CURSOR
212		/ TEST	FOR FUNCTION	
214 215 0170 7	B FUNC			
216 0171 F	E01	CPI	A, E 001	HOME AND CLEAR CNTL A (SOH)
217 0173 0 218 0176 F		UZ CPI	START OOD	i
219 0178 0	A6E02	JZ	CR	CARRAGE RETURN
220 017B F	E11	CP I	011 SAVRO	;SAVE ROW # CNTL Q (DC1)
221 017D 0 222 0180 P		JZ CPI	SAVRO OOC	ADVANCE CURSOR ONTL L (FF)
223 0182 0	A6101	JZ CPI	ADCUR	
224 0185 F 225 0187 0	AA402	JZ	002 HOME	;HOME UP ONTL B (STX)
226 018A F	E1A	CP I JZ	01A SWAP	;SWAP CNTL Z (SUB)
227 018C 0 228 018F F	EOA	UZ CPI	OOA	LINEFEED
229 0191 0 230 0194 F	A8D02	JZ CPI	LF 008	
231 0196 0	CAE002	JZ	BS	;BACKSPACE ONTL H (BS)
232 0199 F 233 019B 0	EOB	CPI	OOB	;UP CURSOR ONTL K (VT)
234 019E F	E18	JZ CPI	UPCUR 018	CLEAR ROW CNTL X (CAN)
235 01A0 0 236 01A3 F	CA3003	UZ CPI	CLROW 007	
237 01A5 0	A4503	JZ	BELL	;RING BELL CNTL G (BEL)
238 01A8 F		CP I	012	(DC2) INVERT NEXT ONTL R (DC2)
239 01AA 0 240 01AD F		JZ CPI	IVERTN 013	/ INVERT ROW ONTL S (DC3)
241 01AF 0	A5403	JZ	IVERTR	
242 01B2 0 243	9	RET		; RETURN
244) STORI	E CURSOR TO CRT	C FROM H-L REGISTERS
245 246 01B3 7		: MOV	A, H	H REG TO ACC.
247 01B4 0	620	ADI	020	SET H-L REG TO CRTC ADD.
248 01B6 6 249 01B7 3	3603	MOV MVI	Н, А М, 003	;H IS CRTC ADD. ;CURSOR REGISTER SELECT
250 0189 7 251 01BA E	'C	MOV SUI	A, H 020	;H REG SET BACK TO VIDIO RAM ;ADDRESS
252 01BC 6	57	MOV	020 H, A	/ HUUREOD
253 01BD 0 254	9	RET		; RETURN
254 255				
256 257		/LAST	ROW ON SCREEN	
258 01BE C	DDC01 NXR0	CALL	NXR01	GO TO NEXT ROW SUBROUTINE
259 0101 0 260 0104 E	DF301	CALL	ZCHAR	ZERO CHARACTER
261 0105 1	E60	W3: PUSH MVI	H E,LASTROW	;SAVE H,L ;POINT D,E TO LASTROW
262 01C7 1 263 01C8 0	A	LDAX ADI	D	
264 01CA F	E30	CPI	001 030	;POINT AC TO FIRST ROW OFF SC ;CK IF LAST ROW IN RAM
265 01CC C	AD701	JZ	ROZERO	
266 01CF U 267 01D2 0	D8302 LOOF D3803	5: CALL CALL	LDHL1 CLROW2	;LOAD H,L WITH ADD. OF LASTRO
268 01D5 E	1	FOF	н	FRESTORE H.L
269 01D6 C 270	.7	REŤ		
271 0107 3		RO: MVI	A, 000	;LOAD ROW ZERO
272 01D9 C 273	30F01	JMP	LOOPS	
				TL/F/586
				Continued Next Page

279 DIED 23 UNX H H		/POINT D-E REG TO LAST ROW /PUT LAST ROW # TO ACC. /EXCHANGE H-L WITH D-E	E, LASTROW D	1: MVI LDAX XCHG	NXR01:	75 76 01DC 1E60 77 01DE 1A
1000000000000000000000000000000000000	₩ # TH	;H-L IS NOW AT 8080 ROW # ;COMPARE LAST ROW # WITH	м	INX CMP	02	79 01E0 23 30 01E1 BE
354 1100000000000000000000000000000000000	-					32 33
227 0182 CAFED1 JZ ZF04 ZERO FON 228 0182 GA IR IRONOCOL FOINT H-L TO CHAR * 230 0182 GA IR IRONOCOL FOINT H-L TO CHAR * 230 0182 GA IRI IRET IRETURN 231 0187 COSO CALL LDHL IRETURN 230 0183 GA JERO COMPACTER IRETURN 230 0187 GA STA OFF3 CASA IAD STORE 230 0187 GA JERO COMPACTER IRETURN IRETURN 230 0187 GA JERO BOBO ROW * IDT PUT CHAR * TO ZERO 230 0187 GA JERO BOBO ROW * IDT JERO BOBO ROW * 330 0187 JERO JERO BOBO ROW * IN R S ADDRESS HOW 331 0200 ZEO MUT L. 000 IN R S ADDRESS HOW 333 0202 CO RET INTERNA IN R S ADDRESS HOW 331 0200 ZEO MUT L. 000 IN R S ADDRESS HOW 332 0202 ZEO MUT L. 020 IN R S ADDRESS HOW 333 0203 ZEO MUT L. 020 IN R S ADDRESS HOW 332 0203 ZEO MUT L. 020 IN R S ADDRESS HOW 333 0203 ZEO MUT A. 027 IERURN 334 0203 ZEO <		TEST FOR MAX ROW AND	A, 02F	O: MVI	INCRO:	84 85 01E5 3E2F
289 01ED 1661 XCHG		; ZERO ROW	ZROW	JZ	01	7 01ES CAFB01
21 DIFF CB202 CALL LPHL 22 DIFZ CP .ZERO CHARACTER 23 DIFZ SEGO ZERAR: MVI A.000 .PUT CHAR * TO ZERO 23 DIFZ SEGO ZERAR: MVI A.000 .PUT CHAR * TO ZERO 230 DIFZ SEGO ZROHAR: MVI A.000 .PUT CHAR * TO ZERO 230 DIFZ SEGO ZROHAR: MVI A.000 .SOSO ROH * TO ZERO 230 DIFZ SEGO MVI L.000 .N R S ADDRESS HIGH ROUTINE 230 DIFZ SEGO MVI L.000 .N R S ADDRESS HIGH ROUTINE 230 DIFZ SEGO MVI L.000 .N R S ADDRESS HIGH ROUTINE 230 DIFZ SEGO MVI L.000 .N R S ADDRESS HIGH ROUTINE 230 DIFZ SEGO MVI L.000 .N R S ADDRESS HIGH ROUTINE 230 DIFZ SEGO RET .N R S ADDRESS HIGH ROUTINE 230 JERSCRESCH ROUTINE .N R S ADDRESS HIGH ROUTINE 230 SCROLL DIFT HIGH ROUTINE .N R S DATACH	W #	;INCREMENT THE SOBO ROW # ;POINT H-L TO CHAR #		XCHG		39 01EC EB
JUE NO.				CALL	02	1 01EF CD8202
250 01F3 3E00 2CHAR: 5TA NVI A.000 :PUT CHAR # TO ZER0 270 .ZER0 3030		, RETURN				3
207 01F5 32633F STA 03F63 .AND STORE 208 01F5 03501 .JPP PCUR .OD TO PUT CURSOR ROUTINE 209 01F5 03500 ZERO 8080 RON #		; PUT CHAR # TO ZERO			ZCHAR	25
300 , ZERO 8080 ROW # 301 DTEP 5400 ZROW MUI N. 000 , S080 ROW # TO ZERO 301 DTEP 5400 ZROW MUI L. 000 , N. R. S. ADDESS HIDH 303 DTEP 5400 MOV D. M N. R. S. ADDESS HIDH 304 D202 DEE MOV E. M N. R. S. ADDESS LOW 305 D202 DE MOV E. M N. R. S. ADDESS LOW 305 D202 DE MOV E. M N. R. S. ADDESS LOW 306 D202 DE MOV E. M N. R. S. ADDESS LOW 307 D203 DES SCROLL MUI A. 02F DEFORE SCRATCH TABLES. 301 D200 CALS NON SCROLL MUI A. 02F DEFORE SCRATCH TABLES. 301 D200 CALS ROW MUI L. FIRSTRO JOHNTH-L TO FIRST ROW# 302 D201 DALE NVI L. FIRSTRO JOHNTH-L TO SCRO ROW 302 D216 CALS NVI L. PIRSTRO JOHNTH-L TO SCRO ROW 302 D216<	INE	JAND STORE	03F63	STA	3F	7 01F5 32633F
020 01FF 36 000 ZFR0: MUI L. 000 .8060 ROW # T0 ZER0 030 01FF 26 MOV D.H N.R.S. ADDESS HIGH 030 01FF 26 MOV D.H N.R.S. DATA HIGH TO D.REG 030 01F5 26 MOV D.H N.R.S. DATA HIGH TO D.REG 030 0202 2E2 MOV E.H N.R.S. DATA HIGH TO D.REG 030 0203 01F5 26 MOV E.H N.R.S. DATA HIGH TO D.REG 030 010 (FR) 2004 02 RET .RETURN 031 0200 228 01 CKN SCROLL 031 010 (FR) 2006 3150 (FR) NR N.S. DATA HIGH TABLES 031 0200 28 020 264 (FR) MUI L.PIRSTRO 031 0200 214 (FR) NR H 031 0200 214 (FR) NVI H.P. TO FIRST ROWH 032 0200 2642 ROLD: MVI H.P. FIRSTRO 031 0200 214 (FR) NVI H.P. FIRSTRO 032 0201 2642 ROLD: MVI H.P. FIRSTRO 032 0202 2642 ROLD: MVI L.PIRSTRO 032 0210 CALEO JZ Z FRO 032 0210 CALEO ROLD: MVI L.PIRSTRO 032 0210 CALEO ROLD: MVI L.PIRSTRO 033 0212 COSOLD JP JP MR 033 0212 COSOLD JP MVI L.PIRSTRO 034 0212 SES JP MR 034 0213 CASOLD ZFRO						19 10
0204 01FF 56 MOV D.M. IN R.S. DATA HIGH TO D.REG 0206 0202 02 MOV E.M. IN R.S. DATA LOW TO D.REG 0207 0203 02 MOV E.M. IN R.S. DATA LOW TO D.REG 0207 0203 02 MOV E.M. IN R.S. DATA LOW TO D.REG 0207 0203 02 MOV F.M. IN R.S. DATA LOW TO D.REG 0207 0203 02 MOV F.M. IN R.S. DATA HIGH TO D.REG 0207 0203 02 F.M. IN R.S. DATA HIGH TO D.REG 0208 0207 26.2 SCROLL: DCX H POINT H-L TO LAST ROW# 0210 0200 26.42 ROLD: MVI L.FIRSTRO POINT H-L TO LAST ROW# 0210 0200 26.42 ROLD: MVI L.FIRSTRO POINT H-L TO RAT ROW 0210 0201 26.42 ROLD: MVI L.FIRSTRO POINT H-L TO SOBO ROW 0210 0210 26.42 ROLD: JMP INCREMENT ROW ROUTINE 0210 0210 26.50 J		, 8080 ROW # TO ZERO				2 01FB 3600
0306 0202 SE MOV E.M IN.R.S. DATA LOW TO E.REG 0307 0203 EB XCHG IRET IRETURN 0309 IRET IRETURN 0300 IRET IRETURN 0310 IRET IRETURN 0310 IRETURN POINT H-L TO LAST ROW# 0311 0205 SEZF SCROLL 0314 ORD SCROLL IRETURN 0315 0205 CA1902 JZ ZLRO JUMP TO ZERO LAST ROW# 0316 0200 CA1902 JZ ZLRO JUMP TO ZERO LAST ROW# 0310 0200 ZEA2 ROLD: MVI L.FIRSTRO POINT H-L TO FIRST ROW# 0320 0200 ZEA2 ROLD: MVI L.FIRSTRO JUMP TO ZERO LAST ROW# 0320 0200 ZEA2 ROLD: MVI L.FIRSTRO JUMP TO ZERO LAST ROW# 0320 0201 ZEA2 ROLD: MVI L.FIRSTRO JUMP TO ZERO LAST ROW# 0320 0216 ZEA2 ROLD: MVI L.FIRSTRO JUMP TO ZERO FIRST ROW# 0320 0216 ZEA2 ROLD: MVI H.RONO POINT H-L TO TOROW ROWTINE 0320 0216 ZEA2 ROLD: MVI M.RONO <t< td=""><td>REG</td><td>IN R.S. DATA HIGH TO D REG</td><td>D, M</td><td>MOV</td><td></td><td>04 01FF 56</td></t<>	REG	IN R.S. DATA HIGH TO D REG	D, M	MOV		04 01FF 56
308 0.00 , ROH SCROLL 310 , ROH SCROLL 311 0.005 28 SCROLL: 312 0.005 28 SCROLL: 313 0.005 28 SCROLL: 314 0.005 28 SCROLL: 315 0.005 28 SCROLL: 315 0.005 28 UP A.02F 315 0.005 28 SCROLL: NEXT FOR THE AST ROM 315 0.005 28 SCROLL: NEXT FOR THE AST ROM 315 0.005 28 ROLD: MUP N.UMP TO ZERO LAST ROM 316 0.000 28642 ROLD: MUP N.UMP TO ZERO LAST ROM 317 JUR M INCREMENT TO NEXT ROM 320 0.202 2862 ROLD: MVI L.FIRSTRO POINT H-L TO SERO ROM 321 0.214 2861 MVI L.ROM8080 POINT H-L TO NEXT ROM SCROMAN 322 0.216 28501 JMP INCRO POINT H-L TO SERO ROM SCROMAN 322 0.216 28501 JMP INCRO POINT H-L TO SERO ROM SCROMAN 323 0.218 C30D02		IN.R.S. DATA LOW TO E REG		MOV		06 0202 SE
310 , ROW SCROLL 311 312 0205 28 SCROLL DCX H , POINT H-L TO LAST ROWH 313 0206 3822 CMP M , A 02F BEFORE SCRATCH TABLES. 314 0208 BE CMP M , TEST FOR THE LAST ROW 314 0208 BE CMP M , TEST FOR THE LAST ROW 315 0207 CALO JUM M INCREMENT TO NEXT ROW 314 0207 DE CALO MVI L, FIRSTRO , POINT H-L TO FIRST ROW 312 0207 DE CALO MVI L, CHIRSTRO , POINT H-L TO SOB ROW 322 0210 OALEO2 JZ FRO , JUM PTO JEER FOR FIRST ROW 323 0212 SACO JUMP INCRO , FUT LAST ROWH TO ZERO ADOR ROW 324 0214 SACO JUM P INCRO , FUT LAST ROWH TO ZERO ADOR ROW 325 0219 SACO ZLRO MVI M. OOD , FUT FIRST ROWH TO ZERO 326 DE MVI M. OOD						07 0203 EB 08 0204 C9
312 0205 28 SCROLL: DCX H			CROLL	; ROW :		10
314 0208 BE CMP M , TEST FOR THE LAST ROW. 315 0202 CA1002 JZ ZLRO JUMP TO ZERO LAST ROW. 316 0200 CA1002 JZ ZLRO JUMP TO ZERO LAST ROW. 317 JANDARA M INCREMENT TO NEXT ROW. 318 0200 ZE62 ROLO: MVI L, FIRSTRO ; POINT H-L TO FIRST ROW. 320 0200 ZE62 ROLO: MVI L, FIRSTRO ; JUMP TO ZERO FIRST ROW. 321 020F BE CMP M INCREMENT TO NEXT ROW. 322 0210 CALEO2 JZ ZFRO ; JUMP TO ZERO FIRST ROW. 324 0214 ZE61 MVI L, ROMBOSO ; FOINT H-L TO SOSO ROW. 325 0216 CSED01 JMP INCRO ; PUT LAST ROW! TO ZERO 326 O218 CS0D02 JMP ROLO ; GO TO INCREMENT ROW ROUTINE 327 JAND INCRO ; PUT FIRST ROW! TO ZERO ; SAVE ACC AND FLAGS 330 0218 CS0D0 ZERO! MVI H, 000 ; PUT FIRST ROW! ; SAVE ACC AND FLAGS 331 JMP INCRO ; SAVE ACC AND FLAGS ; SAVE ACC AND FLAGS 331 G216 JMP INCRO					SCROLL	12 0205 2B
316 0.20C 34 INR H , INCREMENT TO NEXT ROW. 317 318 319 320 0.20D 26.2 ROLO: MVI L, FIRSTRO ; POINT H-L TO FIRST ROW. 321 0.20P 26.2 ROLO: MVI L, FIRSTRO ; JUMP TO ZERO FIRST ROW. 321 0.20P 26.2 ROLO: MVI L, ROWBOSO ; POINT H-L TO FIRST ROW. 322 0.21A 25.61 MVI L, ROWBOSO ; POINT H-L TO SOBO ROW. 324 0.21A C3E501 JMP INCRO ; GO TO ROUTINE FOR FIRST ROW. 326 0.21A C3E002 JMP ROLO ; GO TO ROUTINE FOR FIRST ROW. 325 0.21A C30D02 JMP ROLO ; GO TO ROUTINE FOR FIRST ROW. 330 0.21B C30D02 JMP ROLO ; GO TO INCREMENT ROW ROUTINE 333 0.21B C30D02 JMP ROLO ; GO TO ROUTINE FOR FIRST ROW. 331 0.222 C2E61 MVI L, ROWBOSO ; POINT H-L TO 3080 ROW. 334 0.220 C2F5 NEW ROW START INTERRUPT SAVE ACC AND FLAGS 336 0.222 C5F5 NEWROW ; SAVE ACC AND FLAGS		TEST FOR THE LAST ROW.	м	CMP	02	4 0208 BE
319 320 0200 26.2 ROLD: MVI L, FIRSTRO ; FOINT H-L TO FIRST ROWH 321 0200 DE CMP M ; IS FIRST LOW = TO LAST ROWH 321 0210 CALEC2 JZ ZFRO ; JUMP TO ZERO FIRST R 322 0213 34 MVI L, ROW8080 ; FOINT H-L TO SOBO ROW 324 0214 C2E501 JMP INCRO ; GO TO INCREMENT ROW ROUTINE 326 321 3400 ZLRO: MVI L, ROW8080 ; FOINT H-L TO SOBO ROW 327 328 0218 C30002 JMP ROLO ; GO TO ROUTINE FOR FIRST ROW 330 0218 C30002 JMP ROLO ; GO TO ROUTINE FOR FIRST ROW 331 0220 2661 MVI L, ROW8080 ; POINT H-L TO SOBO ROU 331 0221 2660 ZFRO MVI L, ROW8080 ; POINT H-L TO SOBO ROU 332 0218 C3600 ZERO ; GO TO INCREMENT ROW ROUTINE ; SAVE ACC AND FLAGS 334 0220 ZES1 JMP INCRO ;					~2	16 020C 34
320 0200 2£62 ROLD: MVI L,FIRSTRO FOINT H-L TO FIRST ROWH 321 0200 EE CMP M ISFIRST LOW = TO LAST ROWH 322 0210 CAIE02 JZ ZFRO JUMP TO ZERO FIRST ROWH 323 0213 34 INR M INCREMENT TO NEXT ROWH 324 0214 2E61 MVI L,ROWBOBO FOINT H-L TO SOBO ROW 325 0216 C3E501 JMP INCRO GO TO INCREMENT ROW ROUTINE 326 213 C3002 JMP ROLO GO TO ROUTINE FOR FIRST ROW 327 328 320 218 C30D02 JMP ROLO GO TO ROUTINE FOR FIRST ROW 330 0218 C30D02 JMP ROLO FUT FIRST ROWH TO ZERO 331 0212 C3E501 JMP ROLO FOINT H-L TO SOBO ROW 332 0212 C3E501 JMP INCRO FOINT H-L TO SOBO ROW 334 0222 C3E61 MVI L,ROWBOBO FOINT H-L TO SOBO ROW 335 0212 C3E501 JMP INCRO FOINT H-L TO SOBO ROW 336 0222 C3E61 MVI L,ROWBOBO FOINT H-L TO SOBO ROW 336 0222 C3E51 JMP INCRO FOINT H-L TO SOBO ROW 337 JMP						18
222 0210 CATEO2 JZ ZFRO ,JUMP TO ZERO FIRST R 232 0213 34 INR H INCREMENT TO NEXT ROW 234 0214 2E61 MVI L,ROWBOSO ;FOINT H-L TO SOBO ROW 235 0216 C3E501 JMP INCRO ;GO TO INCREMENT ROW ROUTINE 236 237 337 ;GO TO ROUTINE FOR FIRST ROW 237 238 ;GO TO ROUTINE FOR FIRST ROW ;GO TO ROUTINE FOR FIRST ROW 330 0218 C30D02 JMP ROLO ;GO TO ROUTINE FOR FIRST ROW 331 0218 C30D02 JMP ROLO ;GO TO ROUTINE FOR FIRST ROW 332 0218 C30D02 JMP ROLO ;GO TO ROUTINE FOR FIRST ROW 333 0218 C30D02 JMP ROLO ;GO TO ROUTINE FOR FIRST ROW 334 0220 2E61 MVI L,ROWBOSO ;FOINT H-L TO SOBO ROU 335 0222 C25501 JMP INCRO ;GO TO INCREMENT ROW ROUTINE 336 10220 C25 F5 NEWRO PSW ;SAVE ACC AND FLAGS 337 2228 116435 LXI D,03564 ;POINT H-E TO CRTCROW # 340 0225 F5 NEWRO EAA ;N.R.S. DATA ADD	₩# 37 6°0₩	; POINT H-L TO FIRST ROW# : IS FIRST FOW = TO FAST POW			ROLO:	20 020D 2E62
225 0214 C3E501 JMP INCRO ; G0 TO INCREMENT ROW ROUTINE 236 237 237 238 238 239 0219 3600 ZLRO: MVI M, 000 ; PUT LAST ROWH TO ZERO 330 0218 3600 ZFRO: MVI M, 000 ; PUT FIRST ROWH TO ZERO 331 0216 3600 ZFRO: MVI M, 000 ; PUT FIRST ROWH TO ZERO 334 0220 2661 MVI M, 000 ; G0 TO INCREMENT ROW ROUTINE 336 0222 C3E501 JMP INCRO ; G0 TO INCREMENT ROW ROUTINE 336 0222 C3E501 JMP INCRO ; G0 TO INCREMENT ROW ROUTINE 336 0222 C3E501 JMP INCRO ; G0 TO INCREMENT ROW ROUTINE 336 0222 C3E57 NEWR ROW START INTERRUPT SAVE H-L REG 337 JNEW ROW START INTERRUPT SAVE H-L REG 340 0225 FS NEWRO PUSH H ; SAVE H-L REG 341 0227 DS PUSH D ; LOADA ACC MITH CHC ROW # ; LOADA ACC MITH CHC ROW # 342 0228 IA LDAX D ; NEN		JUMP TO ZERO FIRST R	ZFRO	JZ	02	22 0210 CA1E02
323 329 329 0219 3600 ZLRO: MVI M.000 , PUT LAST ROWH TO ZERO 330 0218 C30002 JMP ROLO , GO TO ROUTINE FOR FIRST ROW 331 0218 C30002 JMP ROLO , GO TO ROUTINE FOR FIRST ROW 331 0220 2EA1 MVI L, ROWSOBO , FDINT H-L TO SOBO ROW 335 0222 CSE501 JMP INCRO , GO TO INCREMENT ROW ROUTINE 336 JNEW ROW START INTERRUPT SAVE ACC AND FLAGS JAVE H-L REG 337 JNEW ROW START INTERRUPT JAVE H-L REG JAVE H-L REG 337 JNEW ROW START INTERRUPT JAVE H-L REG JAVE H-L REG 338 0226 E5 PUSH H , SAVE H-L REG 334 0228 11643F LDAX D , COAD ACC MITH CRT CROW # JAVE ACC AND FLAGS 340 0228 16A LDAX D , ROW DATA ADD HIGH TO E JAVE ACC ACC AND FLAGS 344 0220 SF MOV			L, ROWSOSO	MVI	01	24 0214 2E61
229 0219 3600 ZLR0: MVI M,000 ; PUT LAST ROWH TO ZERO 330 0218 C30002 JMP ROLO ; GO TO ROUTINE FOR FIRST ROW 331 0321 ; GO TO ROUTINE FOR FIRST ROW ; GO TO ROUTINE FOR FIRST ROW 331 03220 2E61 MVI L, ROWSOB ; PUT FIRST ROWH TO ZERO 334 0220 2E61 MVI L, ROWSOB ; POINT H-L TO SOBO ROW 335 0222 CSES01 JMP INCRO ; GO TO INCREMENT ROW ROUTINE 335 335 (222 CSES01 JMP NEW ROW START INTERRUPT 336 JNEW ROW START INTERRUPT ; SAVE ACC AND FLAGS ; AVE H-L REG 334 0226 E5 PUSH H ; SAVE H-L REG 341 0227 E5 NEWRO E4 ; POINT D-E TO CRTCROW # 342 0228 116435 LXI D, 05664 ; POINT D-E TO CRTCROW # 344 0220 SF MOV E, A ; N.R. S. DATA ADD HIGH INTO H 345 0222 CSA ADI OZO ; ACC TO N.R.S. DA	OUT THE	, go to increment KOW KUUTIN	INCRO	One	~.	26
330 0218 C30002 JMP ROLO ; G0 TO ROUTINE FOR FIRST ROW 331 0212 C25002 JMP ROLO ; G0 TO ROUTINE FOR FIRST ROW 332 0220 2E61 MVI L, ROWS030 ; PUT FIRST ROWN TO ZERO 334 0220 2E61 MVI L, ROWS030 ; FOINT H-L TO S080 ROW 335 0222 CSE501 JMP INCRO ; G0 TO INCREMENT ROW ROUTINE 336 JMP NEW ROW START INTERRUPT SAVE ACC AND FLAGS 3000000000000000000000000000000000000	1	; PUT LAST ROW# TO ZERO	M, 000	: MVT	ZLRO	28
323 021 360 ZFR0* MVI M,000 ,PUT FIRST RGMM TO ZERO 334 0220 2EA1 MVI L,R0W8080 ,POINT H-L TO 8080 ROULINE 335 0222 CSE501 JMP INCR0 .GO TO INCREMENT ROW ROUTINE 336 JNEW ROW START INTERRUPT SAVE ACC AND FLAGS					02	0 021B C30D02
334 0220 2E61 MVI L,R0W8080 ,P0INT H-L_T0 8080 ROW 335 0222 C3E501 JMP INCR0 ,GO TO INCREMENT ROW ROWINE 336 ,NEW ROW START INTERRUPT 337 ,NEW ROW START INTERRUPT 338 ,NEW ROW START INTERRUPT 339 0226 E5 PUSH + ,SAVE ACC AND FLAGS 340 0226 E5 PUSH + ,SAVE H-L REG 341 0227 D5 PUSH D . 342 0228 11643F LXI D,03564 ;POINT D-E TO CRTCROW # 343 0228 12A LDAX D COAD ACC WITH CRTC ROW # 344 0220 SF MOV E,A .N.R.S. DATA ADD HIGH TO E 345 0221 C420 ADI 020 ROW DATA ADD HIGH INTO ACC 346 0224 C530 ADI 030 .ACC TO N.R.S. DATA ADD HIGH INTO H 348 0231 78 MOV A.E ROW DATA LOW LOT O ACC 350 0234 SF MOV A.A .N.R.S. DATA ADD LOW TO E REG 351 0235 1A LDAX D ROW DATA LOW TO ACC 355 0238 1654 MOV L.A .N.R.S. DATA ADD LOW TO L 355 0237 164 LDAX </td <td>0</td> <td>; PUT FIRST ROW# TO ZERO</td> <td>M-000</td> <td>MVI</td> <td>ZERO</td> <td>32 33 021E 3600</td>	0	; PUT FIRST ROW# TO ZERO	M-000	MVI	ZERO	32 33 021E 3600
377 , NEW ROW START INTERRUPT 388 389 0225 F5 NEWRO: PUSH PSW ; SAVE ACC AND FLAGS 389 0225 F5 PUSH H ; SAVE H-L REG 381 0227 D5 PUSH D ; SAVE H-L REG 381 0227 D5 PUSH D ; SAVE H-L REG 382 0228 11643F LDAX D ; LOAD ACC WITH CART COW # 384 0220 1A LDAX D ; LOAD ACC WITH CART COW # 385 0220 1A LDAX D ; ROW DATA HIGH INTO ACC 386 0230 67 MOV H,A ; N.R.S. DATA ADD HIGH INTO H 386 0232 C630 ADI 020 ; ACC TO N.R.S. DATA LOW 380 0234 57 MOV A.E ; ACC TO N.R.S. DATA ADD LIGH INTO H 380 0237 B MOV A.A ; N.R.S. DATA ADD LOW TO E REG 381 0235 1A LDAX ; ACC TO N.R.S. DATA ADD LOW ; ACC TO N.R.S. AND VERT INTER 381 0237 3601 MVI H, 6001 ; STOKE N.R.S. TO CRTC ; STOKE N.R.S. TO CRTC 385 0239 D340 OUT 040 ; STOKE N.R.S. AND VERT INTER	J	POINT H-L TO SOSO ROW	L, ROWSOSO	MVI		84 0220 2E61 85 0222 C3E501
339 0225 F5 NEWRO: PUSH PSW ;SAVE ACC AND FLAGS 340 0226 F5 PUSH D ;SAVE H-L REG 341 0227 D5 PUSH D ;SAVE H-L REG 341 0227 D5 PUSH D ;SAVE H-L REG 341 0228 114 LDAX D ;SAVE H-L REG 343 0228 114 LDAX D ;SAVE ACC WITH CRTC ROW # 343 0228 14 LDAX D ;NR S DATA ADD HIGH TO E 345 0220 1A LDAX D ,ROW DATA HIGH INTO ACC 345 346 0220 F MOV H,A ;N.R.S. DATA ADD HIGH INTO H 346 0230 67 MOV H,A ;N.R.S. DATA ADD LOW TO E REG 351 0231 FR MOV A,E ;ACC TO N.R.S. DATA ADD LOW TO E REG 351 0235 FA MOV L,A ;N.R.S. DATA ADD LOW TO E REG 352 0236 F MOV L,A ;N.R.S. DATA ADD LOW TO E REG		RUPT	OW START INTER	NEW (37
941 0227 D5 FUSH D 934 0228 11643F LIAX D 0.03F64 /FOINT D=E TO CRTCROW # 933 0228 11643F LIAX D .LOAD ACC WITH CRTC ROW # 943 0228 11643F HOW E.A N.R.S. DATA ADD HIGH TO E 943 0220 5F MOV E.A N.R.S. DATA ADD HIGH TO E 945 0220 1A LDAX D .ROW DATA HIGH INTO ACC 946 0220 C620 ADI 020 .ROW DATA HIGH INTO ACC 947 0230 67 MOV H.A .N.R.S. DATA ADD HIGH INTO H 948 0231 78 MOV A.E .ACC TO N.R.S. DATA ADD LOW TO E REG 935 0232 C630 ADI 030 .ACC TO N.R.S. DATA ADD LOW TO E REG 935 0235 1A LDAX D .ROW DATA ADD LOW TO L 935 0237 3601 MVI H.001 .STORE N.R.S. ADD VERT INTER* 935 0239 164 MUI 040 .RESET N.R.S. AND VERT INTER* 935 0230 1864 MVI LOAR .INCREMENT TO NEXT ROW 935 0230 1864 MVI LCRTCROW .STORE NEXT ROW AUMERT INTER* 935 0240 CA4A02 JZ ZCRTC .IFT TRUE ZERO ACC					NEWRO	39 0225 F5
943 0228 1A LDAX D LDAD ACC NITH CRTC ROW # 944 0222 SF MOV E, A IN R.S. DATA ADD HIGH TO E 945 0220 IA LDAX D , ROW DATA HIGH INTO ACC 946 0220 CADI 020 , ROW DATA HIGH INTO ACC 946 0230 CA MOV H.A , ROW DATA ADD HIGH INTO H 948 0231 78 MOV A.E , ACC TO N.R.S. DATA ADD LOW TO H 949 0232 CASO ADI 030 ; ACC TO N.R.S. DATA ADD LOW TO E 950 0234 SF MOV A. ; N.R.S. DATA ADD LOW TO L 951 0235 1A LDAX D , ROW DATA LOW TO ACC 952 0236 GAF MVI H.001 ; STORE N.R.S. TO CRTC 953 0237 3601 MVI E.CRTCROW ; 954 0239 D340 OUT 040 ; RESET N.R.S. ADD VERT INTER^M 950 <			D	PUSH	9E	1 0227 D5
945 0220 1A LDAX D , ROW DATA HIGH INTO ACC 946 0225 C620 ADI 020 ;N.R.S. DATA ADD HIGH INTO ACC 947 0230 67 MOV H.A ;N.R.S. DATA ADD HIGH INTO H 948 0231 78 MOV A.E ;ACC TO N.R.S. DATA ADD LOW TO H 949 0232 C630 ADI 030 ;ACC TO N.R.S. DATA ADD LOW TO H 950 0234 SF MOV E.A ;N.R.S. DATA ADD LOW TO E REG 951 0235 1A LDAX D ;N.R.S. DATA ADD LOW TO L 952 0236 6F MOV L.A ;N.R.S. DATA ADD LOW TO L 953 0237 3601 MVI H.001 ;STORE N.R.S. TO CRTC 954 0239 D340 OUT 040 ;RESET N.R.S. AND VERT INTER^ 956 0230 1A LDAX D ;STORE N.R.S. TO CRTC 957 0235 FE2F CPI 02F ;TEST FOR CRTC MAX ROW 950 0243 024 JA LDAX D ;STORE N.R.S. AND VERT INTER^ 950 0243 024 IA LDAX D ;STORE N.R.S. CON DAXE STORE N.S. 950 0243 024 INR A ;INCREMENT NO NEXT ROW STORE NEXT ROW STORE NEXT ROW)₩ #	;LOAD ACC WITH CRTC ROW #	Ð	LDAX	0.1	3 0228 1A
147 0230 67 MOV H.A ;N.R.S. DATA ADD HIGH INTO H 148 0231 78 MOV A.E ;ACC TO N.R.S. DATA ADD HIGH INTO H 149 0232 C630 ADI 030 ;ACC TO N.R.S. DATA ADD LOW TO E REG 151 0235 1A LDAX D ;ROW DATA LOW TO E REG 152 0236 6F MOV E.A ;N.R.S. DATA ADD LOW TO E REG 152 0237 2601 MVI H.OO1 ;STORE N.R.S. TO CRIC 154 0237 030 164 MVI H.OO1 ;STORE N.R.S. ADD LOW INTO L 155 0238 1644 MVI E.CRICROW ;RESET N.R.S. AND VERT INTER" 155 0238 1644 MVI E.CRICROW ;RESET N.R.S. AND VERT INTER" 156 0230 1A LDAX D ;STORE N.R.S. TO CRIC MAX ROW 157 0235 FC426 CPI 02F ;TEST FOR CRIC MAX ROW 158 0240 CA4402 JZ ZCRIC ;IF TRUE ZERO ACC 159 0243 3C INR A ;INCREMENT TO NEXT ROW 160 0241 12 LOOP ;STORE NEXT ROW NUMBER 360 0244 12 160 0245 D1 POP ;STORE NEXT ROW NUMBER 363 0247 F1 162 0246 E1 POP ;RESTORE ACC AND FLAGS	ж. С	ROW DATA HIGH INTO ACC	D	LDAX		15 022D 1A
349 0232 CASO ADI 030 ;ACC TO N.R.S. DATA LOW 350 0234 SF MOV E.A N.R.S. DATA ADD LOW TO E REG 351 0235 1A LDAX D ;ROW DATA LOW TO ACC 352 0236 6F MOV L.A ;N.R.S. DATA ADD LOW TO L 353 0237 3601 MU M.001 ;STORE N.R.S. TO CRTC 353 0237 3601 MU M.001 ;STORE N.R.S. ADD LOW INTO L 353 0237 3601 MU M.001 ;STORE N.R.S. AND VERT INTER^ 355 0238 16.4 MU E.CRTCROW 355 356 0230 1A LDAX D 357 357 0232 FE2F CF1 02F ;TEST FOR CRTC MAX ROW 358 0240 CAHA02 JZ ZCRTC ;IFTUE ZERO ACC 350 0241 12 00F STORE NEXT ROW NUMBER 361 361 0245 1NR	INTO H	IN R. S. DATA ADD HIGH INTO	H, A	MOV		17 0230 67
ST D235 IA LDAX D ROW DATA LOW TO ACC SS2 0236 6F MOV LA IN R STADE SS2 0236 6F MOV LA IN R STADE IN R SS2 0236 6F MOV LA IN R STADE STADE STADE STADE STADE IN R STADE	W 10 E REG	ACC TO N.R.S. DATA LOW AN R S. DATA ADD LOW TO F R	030	ADI		49 0232 C630
953 0237 3601 MVI M.001 -STORE N.R.S. TO CRTC 354 0239 0340 OUT 040 ;RESET N.R.S. AND VERT INTEF" 355 0238 164 MVI E.CRTCR0W ;RESET N.R.S. AND VERT INTEF" 355 0238 164 MVI E.CRTCR0W ; 357 023E FE2F CPI 02F ; 357 023E FE2F CPI 02F ; 357 023E FE2F CPI 02F ; 358 0240 CAAA02 JZ ZCRTC ; IF TRUE ZERO ACC 359 0243 3C INR A ; INCREMENT TO NEXT ROW 360 0244 12 LOOP STAX D ; STORE NEXT ROW NUMBER 361 0245 D1 FOP D ; RESTORE ACC AND FLAGS 364 0248 FB EI ; SESTORE ACC AND FLAGS 364	ITO L	ROW DATA LOW TO ACC N.R.S. DATA ADD LOW INTO L	Ð	LDAX		51 0235 1A
325 0228 1E64 MVI E.CRTCROW 325 0228 1E64 D 326 0230 1A LDAX D 327 0232 FE2F CPI 02F ; TEST FOR CRTC MAX ROW 326 0234 32 JZ ZCRTC ; JF TRUE ZERO ACC 328 0240 32 INR A ; INCREMENT TO NEXT ROW 340 0243 32 INR A ; INCREMENT TO NEXT ROW 360 0244 12 LOOP: STORE NEXT ROW NUMBER 36 361 0245 DI FOP FO 36 362 0247 F1 FOP F 38 363 0247 F1 FOP F 36 36 364 0248 FB EI 36 36 36 36 364 0249 C9 RET ; RETURN 36 36 364 0249 C9 <td></td> <td>STORE N.R.S. TO CRTC RESET N.R.S. AND VERT INTE</td> <td>M, 001</td> <td>MVI</td> <td></td> <td>53 0237 3601</td>		STORE N.R.S. TO CRTC RESET N.R.S. AND VERT INTE	M, 001	MVI		53 0237 3601
357 023E FE2F CPI 02F ; TEST FOR CRTC MAX ROW 358 0240 CA4A02 JZ ZCRTC ; IF TRUE ZERO ACC 359 0240 SCA4A02 JZ ZCRTC ; INCREMENT TO NEXT ROW 350 0245 3C INK A ; INCREMENT TO NEXT ROW 350 0245 3C INK A ; INCREMENT TO NEXT ROW 360 0244 12 LOOP: STORE NEXT ROW 361 0245 D1 POP D ; RESTORE NEXT ROW NUMBER 362 0247 F1 POP H ; RESTORE H-L REQ 363 0247 F1 POP FSW ; RESTORE ACC AND FLAGS 364 0248 FB EI 365 0249 C9 RET 364 367 367 XL/F/586			E, CRICROW	MVI		55 023B 1E64
359 0243 3C INK A ; INCREMENT TO NEXT ROW 350 0244 12 LOOP: STAX D ; STORE NEXT ROW 361 0245 D1 POP D ; STORE NEXT ROW NUMBER 361 0245 D1 POP D ; RESTORE NEC SO 362 0245 P1 POP H ; RESTORE ACC AND FLAGS 363 0247 F1 POP PSW ; RESTORE ACC AND FLAGS 364 0248 FB E1 365 0249 C9 RET ; RETURN 364 0249 C9 RET ; RETURN 367	J	; TEST FOR CRTC MAX ROW ; IF TRUE ZERO ACC	02F	CPI	02	57 023E FE2F
341 0245 DI POP D 362 0245 DI POP H ;RESTORE H-L REG 363 0247 FI POP PSW ;RESTORE ACC AND FLAGS 364 0248 FB EI 365 0249 C9 RET ;RETURN 366 367 367 TL/F/586		; INCREMENT TO NEXT ROW	A	INR		59 0243 3C
363 0247 F1 POP PSW ;RESTORE ACC AND FLAGS 384 0248 FB EI 365 0249 C9 RET ;RETURN 366 367 367			Ď	POP		51 0245 D1
365 0249 C9 RET ; RETURN 366 367 TL/F/586	ŝ	; RESTORE ACC AND FLAGS		POP		53 0247 F1
367 TL/F/580		; RETURN				5 02 4 9 C9
	TL/F/5866					
Continued Next Page		Continued Next F				
-						

368 # ZERO CRICROW 369 370 024A 3E00 371 024C C34402 MVI JMP ZCRTC A, 000 L00P ZERO ACC 371 024C C3440 372 373 374 375 024F F5 376 0250 E5 377 0251 D5 378 0252 1E62 370 0251 1E62 VERTICAL INTERRUPT PUSH PUSH PUSH VERTI ;SAVE ACC AND FLAGS ;SAVE H REG PSW Ð 377 0251 D5 378 0252 1E62 379 0254 1A 380 0255 1E64 381 0257 12 382 0258 E63F 383 0255 1E6 384 0255 1A 385 025C C620 386 0255 C620 387 025F 7B 388 0260 C630 389 0262 5F 390 0263 1A 391 0264 6F 392 0265 3602 393 0267 D340 394 0269 D1 395 0264 E1 396 0266 F1 396 0266 F1 397 026C FB 398 0260 C9 399 400 401 402 026E 1E63 403 0270 3E00 404 0272 12 405 0273 1E61 MVI E, FIRSTRO POINT D-E TO FIRST ROW # LDAX MVI ;LOAD 1ST ROW # INTO ACC ;POINT D-E TO CRTCROW # ;UPDATE CRTCROW # ;REMOVE MARKER E, CRTCROW STAX n 03F E, A ANT MOV LDAX ADI FOINT H L TO CRTC FIRST ROW D 020 H, A A, E 030 E, A MOV MOV ADI MOV LDAX Ð MOV MVI OUT POP Ľ, A M. 002 STORE TOP OF PAGE 040 D H POP POP POP EI RET PS₩ RESTORE ACC AND FLAGS RETURN ; CARRAGE RETURN MVI MVI STAX E, CHARNUM A, 000 ; POINT D-E TO CHAR # CR404 0272 12 405 0273 1E61 406 0275 CD8202 407 0278 C38301 408 409 410 411 027B 1E61 412 027D 1A 413 027E 1E65 414 0270 12 D E, ROW8080 MVI CALL Глн PCUR CURSOR TO THE BEGINNING OF R SAVE ROW ;POINT D-E TO SOBO ROW# ;PUT SOBO ROW # TO ACC ;POINT D-E TO ROW SAVE ;STORE ROW SAVE # IN REF TAE ;RETURN MVI LDAX SAVRO E. ROW8080 D MVI STAX RET E, ROWSAVE 413 027E 1E4 414 0280 12 415 0281 C9 416 417 418 D

 417
 ;

 418
 ;

 419
 0282
 1A
 LDHL:

 420
 0283
 5F
 LDHL:

 421
 0284
 1A
 420
 282

 422
 0285
 67
 423
 0284
 1A

 422
 0287
 C&30
 425
 0289
 5F
 426
 0284
 1A

 427
 0288
 6F
 424
 0280
 CD
 427
 0288
 6F

 430
 0280
 CDDC01
 LF:
 433
 0290
 CDC401
 434
 0292
 1E61
 435
 0298
 266.535
 ADDCH:
 437
 0298
 85
 438
 0290
 6F
 438
 0292
 6F
 439
 0291
 7C
 440
 0240
 67
 441
 0240
 67
 441
 0240
 67
 4420
 0241
 C38301
 5 H-L ROW DATA LOAD ROUTINE ;LOAD ACC WITH D-E DATA ;POINT D-E TO N.R.S. DATA HI' ;ROW # TO N.R.S. DATA HIGH ;ROW # TO H REG ;PUT IST ROW # TO ACC ;ACC TO N.R.S. ADD LOW ;POINT D-E TO N.R.S. DATA LOW ;ROW # TO N.R.S. DATA LOW ;ROW # TO N.R.S. DATA LOW ;ROW # TO N.R.S. LDAX MOV LDAX D E, A D H, A A, E 030 E, A D MOV MOV ADI MOV LDAX MOV RET Ĺ, A RETURN LINEFEED ; DO NEXT ROW SUBROUTINE ; OFF SCREEN CLEAR ROW ROUTINE ;MOVE REFERENCE ROW # TO H-L ;CHAR # TO ACC ;ADD THE CHAR # TO THE ;FIRST ROW ADDRESS ;IF A CARRY OCCURED ADD TO ;THE DATA HIGH ;H-L POINTS TO LINE FED ROW ;PUT CURSOR TO LINE FED ROW NXR01 CLROW3 E, ROW8080 LDHL 03F63 CALL CALL MVI CALL LDA ADD Ł L, A A, H 000 MOV MOV ACI 441 02A0 67 442 02A1 C3B301 443 444 445 446 02A4 1E62 447 02A6 1A 448 02A7 1E61 448 02A7 1E61 448 02A7 1E61 449 02A9 12 450 02AA CD8302 451 02AD 3E00 452 02AF 32633F 453 02B2 C3B301 454 MOV H, A JMP POUR PUT CURSOR TO LINE FED ROW HOME CURSOR TO T. O. P. HOME MVI E, FIRSTRO POINT D-E TO 1ST ROW LDAX MVI STORE FIRSTROW TO ROWSOSO D E, ROWSOSO STAX n ; MOVE REFERENCE ROW TO H-L ; PUT CHAR # BACK ; TO ZERQ ; PUT CURSOR HOME CALL MVI STA D LDHL1 A, 000 03F63 . IMP PCUR TL/F/5866-17 Continued Next Page

155			; SWAP	ROWS	
156 157 02B5 1 158 02B7 0	D8202	SWAP:	MVI CALL	E, ROWSAVE LDHL 03F66 E, ROWS080 LDHL E, ROWSAVE D E, A A, H D E, A A, E 030 E, A A, L D E, A A, L D A, E 03566 E, A A, H D C3F66 E, A A, E 030 E E, A A, E 030 E E A, C E 03F66 E E A C E C E C E C E C E C E C E C E C	;POINT D-E TO ROW SAVE # AND ;PUT IN H-L REG.
409 UZBA 2	2663F		SHLD	03F66	STORE ROW SAVE # TO TEMP 1 *
460 02BD 1 461 02BF 0			MVI	E, ROW8080	POINT D-E TO SOBO ROW # AND
62 02C2 1			MVI	E, ROWSAVE	;PUT ADDRESS IN H-L REG ;POINT D-E TO ROW SAVE # AND
¥63 02C4 1	A		LDAX	E, ROWSAVE D	PUT IN ACC
64 0205 5 65 0206 7			MOV	E, A	SOSO ROW # TO ADD HIGH
66 0207 1	.2		STAX	А, H D	;STORE SOSO ROW # TO N.R.S. ;DATA HIGH.
66 0207 1 67 0208 7	'B		MOV	A, E	
68 0209 0 69 0208 5	630		ADI	030	
70 02CC 7	10		MOV	Е, н А, L	;PUT SOSO ROW # TO ;N.R.S. DATA LOW
71 02CD 1	.2		STAX	D	\$8080 ROW # IS NOW IN ROW SA"
72 02CE 2	E61		MVT	03F66 E.R0W8080	;PUT ROW SAVE # BACK TO H-L ;COMENT SAME AS ABOVE
74 02D3 1	A		LDAX	D	CONERT SHILE NO ADOVE
175 02D4 5 176 02D5 7	SF VC		MOV	E, A	`
77 0206 1	2		STAX	D	
78 0207 7	'B		MOV	A, E	
179 02D8 0 180 02DA 5			ADI MOV	030 E.A	
81 02DB 7	םי		MOV STAX	A, L	
82 02DC 1			JIHA -	0	
183 02DD 0 184	39802		JMP	ADDCH	JUMP TO ADD CHAR.
			3 BACK	SPACE	
87 02E0 1	E63	BS:	MVI	SPACE D 000 UFROW A D H	POINT THE D-E REG TO CHAR #
88 02E2 1	A		LDAX	D	;AND PUT IN ACC
189 02E3 F 190 02E5 C	E00 AEE02		CPI JZ	ODD UPROW	;TEST FOR THE CHAR # = ;TO ZERO. JUMP IF TRUE ;DECEMENT CLOP #
91 02E8 3	D		DCR	A	DECREMENT CHAR #
92 02E9 1	2		STAX	D H	STORE DECREMENTED CHAR #
93 02EA 2	3B301		DCX JMP	PCUR	;DEC H-L FOR NEW CURSOR LOCA; ;PUT CURSOR IN DECREMENTED LO
195 196			NEXT		
197 198 02EE 3	E4F	UPROW:	MVI	A, 04F	;MOVE THE CHAR #
99 02F0 1			STAX	D	TO SOH AND STORE IT.
501			, MOVE	CURSOR UP	
502 000 0054 5					
503 02F1 E 504 02F2 2	E61	UPCOR:	MVI	L, ROW8080	;POINT H-L TO 8080 ROW AND D- ;TO NEW CURSOR LOCATION.
505 02F4 7			MOV	0. M	TEST IF NEXT UP CURSOR WILL
506 02F5 2			INX	H	BE ON THE FIRST ROW.
507 02F6 E 508 02F7 C				M UPSCL	/IF TRUE JUMP TO /UP SCROLL ROUTINE.
509 02FA 2			DCX	н	POINT H-L BACK TO SOBO ROW #
510 511 02FB F	FOO	BACK1-	CRI	000	
11 02FB F			JZ	R048	;IF 8080 ROW # IS EQUAL TO ;ZERO JUMP TO ROW 48 ROUTINE.
5 13 0 300 S			DCR	Μ	DECREMENT BOBO ROW #
514 515 0301 E	P	L00P1:	усно		POINT 1 TO NEW OURGOD LODA
16 0302 C	D8202		CALL	LDHL	;POINT H−L TO NEW CURSOR LOCA ;AND D−E TO 8080 ROW #. JUMP
17 0305 C	39802		JMP	ADDCH	TO ADD CHARACTER ROUTINE.
18 19 0308 7	E	UPSCL:	MOV	A, M	PUT FIRST ROW # INTO ACC.
20 0309 F	E00		CPI	000	/TEST IF FIRST ROW # IS = TO
621 030B C 622 030E 3			UZ DCR	FR048 M	ZERO, IF TRUE JUMP TO ROW
523					;48 ROUTINE.
i24 030F 2 i25 0311 7	E60	L00P2:	MVI MOV	L, LASTROW	
26 0312 F	E00		CPI	A, M 000	
27 0314 C	A2A03		JZ	LR048	
28 0317 3 29	5		DCR	м	
30 0318 2		L00P3:	MVI	L, ROW8080	FOINT H-L TO SOBO ROW #
31 031A 7 32 031B C			MOV	A, M RACKI	; AND LOAD TO ACC
33			JMP	BACK1	
34 031E 3		R048:	MVI	A, 02F	CHANGE 8080 ROW #
135 0320 7 136 0321 C			MOV JMP	M, A LOOP1	;TO 23D AND STORE ;JUMP TO POINTER EXCHANGE ROU
37					
i38 0324 3 i39 0326 7		FR048:	MVI MOV	A, 02F	;
39 0328 / 340 0327 C			JMP	M,A LOOP2	
i41 i42 032A 3		LR048:		A, 02F	PHIT THE 1ST BOU TO
42 032A 3 43 032C 7		ENU40.	MOV	M, A	; PUT THE 1ST ROW TO ; 17H.
i44 032D C	31803		JMP	LOOPS	JUMP TO SOSO ROW # STORE
545					
					TL/F/5

E 40				; CLEAR	ROW ROUT:	INE					
549		CD3603 C36E02		CALL JMP	CLROW1 CR						
552 553	0338 0338	1E61 CD8202 3E50 3620	CLROW1: CLROW2. LOOP4:	CALL	E, ROW808 LDHL A, 050 M, 020	30	; INTIL]	OW DATA I IZE LOOP ASCII SF	COUNTER	R.	
555 556 557	033F 0340 0341	3D C8 23		DCR RZ INX	A H LOOP4		⇒RETURN ⇒NEXT L	MENT LOOF N IF ZERC LOCATION) BIT IS		
559 560	0345	C33D03	BELL	UMP OUT	001		; CLEAR	NEXT LOC BELL	ATTON.		
562 563	0347 0348	AF	IVERTN		A						
565 566	034B 034C	17		MVI LDAX RAL	E, IMASK D			D, E TO M T 8 STATU			
568 569 570			RESET	JC MVI STAX RET	RESET A, 080 D		; INVER ; STORE	TBITS OUTNEW	MASK		
573		E5 1E61 CD8202	IVERTR	PUSH MVI CALL	H E, ROWSON LDHL	BÓ	: 1 ÂΔĤ	IST ADD.	0E 208	SECH TO	
575 576 577	035A 035C 035D	1E50 7E 17	L00P6:	MVI MOV RAL	E, 050 A, M		;SET CO ;GET CH	DUNTER			
579 580	0361	F680	BACK2:	JC RAR ORI MOV	RESET1 080 M,A			BIT S HIC MOD. CHA		EM	
582 583 584	0365 0366 0367	23 7B FE01		INX MOV CPI	H A, E 001		; POINT	TO NEXT	MEM		
586	0360	CA7603 1D C35C03		JZ DCR JMP	DONE E LOOP6			N IF COUN COUNTER	41 = ZEP	κυ	
590 591		1F E67F C36403	RESET1:	RAR AN I JMF	07F BACK2		RESET	BIT 8			
	0376 0377		DONE	POP RET END	H START						
373		0000		. E.ME	START						TL/F/5866-1
			0007 0000 00F2 0104	ACELD B110 B2000 B4800	00D4 00F8		0161 00EC 00FE 00E6	ADBCH B150 B300 B7200	0298 00DA 00E0 0110		
		B9600 Baud Charnu	0093	BACK BELL CLRAM		BACK1 BS CLRAM1	02E0 004C	BACK2 C CLROW	0001 0330		
		CLROW1 CRTCRO FIRSTR	0064 0062	CLROW2 D FRO48	0002 032 4	CLROW3 DONE FUNC	0376 0170	CR E H	026E 0003 0004		
		HMCUR INIT IVERTR	003B	HOME INTACE	0005	IMASK INTKB LASTRO LOOP	0136	INCRO IVERTN LDHL LOOP1	03 4 8 0282		
		LOOP2	030F 035C 0061	LF LOOP3 LR048 NXR0		LOOP4 M NXR01	033D 0006	LOOP1 LOOP5 NEWRO PCUR	01CF		
		PSW ROLO	0006 020D	RESET ROW808 SCROLL	0352 0061	RESET1 ROWSAV SP	0370	R048 ROZERO START	031E 01D7		
		SAVRO	027B		0066 *	TEMP2	0067 * 024F	UPCUR ZCHAR			
			0285 02EE	TEMP1 UPSCL ZFRO	0308	ZLRO	0219	ZROW	01FB		
		SAVRO SWAP UPROW ZCRTC NO ERF SOURCE OBJECT	0285 028E 024A COR LINES CHECKSUM CHECKSUM	UPSCL ZFRO = 403F = 0F51	0308	ZLRO	0219				

DEFINITIONS

ACE—Asynchronous communication element CRTC—Cathode ray tube controller Video Page —Visible screen data Video RAM—Entire portion of RAM used only for display First Row #—Address for top row of video page Last Row #—Address for bottom row of video page CRTC Row #—Address for next row load 8080 Row #—Address for cursor row Character #—Character location in a row

XXXH are hexidecimal numbers

REFERENCES

National Semiconductor Data Sheets:

DP8350 Series Programmable CRT Controllers INS8250 Asynchronous Communications Element

National Semiconductor Application Notes:

Simplify CRT Terminal Design with the DP8350, AN-198 Data Bus and Differential Line Drivers and Receivers, AN-83

Transmission Line Characteristics, AN-108

Hardware Reference Manual BLC 80/10 Board Level Computer. National Semiconductor Microcomputer Systems Chapter 6—System Interfacing.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

- Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
- A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

	National Semiconductor	National Semiconductor	National Semiconductor	National Semiconduct
V	Corporation	Europe	Hong Kong Ltd.	Japan Ltd.
	1111 West Bardin Road	Fax: (+49) 0-180-530 85 86	13th Floor, Straight Block,	Tel: 81-043-299-2309
	Arlington, TX 76017	Email: cnjwge@tevm2.nsc.com	Ocean Centre, 5 Canton Rd.	Fax: 81-043-299-2408
	Tel: 1(800) 272-9959	Deutsch Tel: (+49) 0-180-530 85 85	Tsimshatsui, Kowloon	
	Fax: 1(800) 737-7018	English Tel: (+49) 0-180-532 78 32	Hong Kong	
		Français Tel: (+49) 0-180-532 93 58	Tel: (852) 2737-1600	
		Italiano Tel: (+49) 0-180-534 16 80	Fax: (852) 2736-9960	

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.