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

National Semiconductor
Application Note 199
Al Brilliot
July 1986

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

INTRODUCTION

The DP8350 is an I^2L—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 allowing 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 DP8350 is equivalent to the INS8276.

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

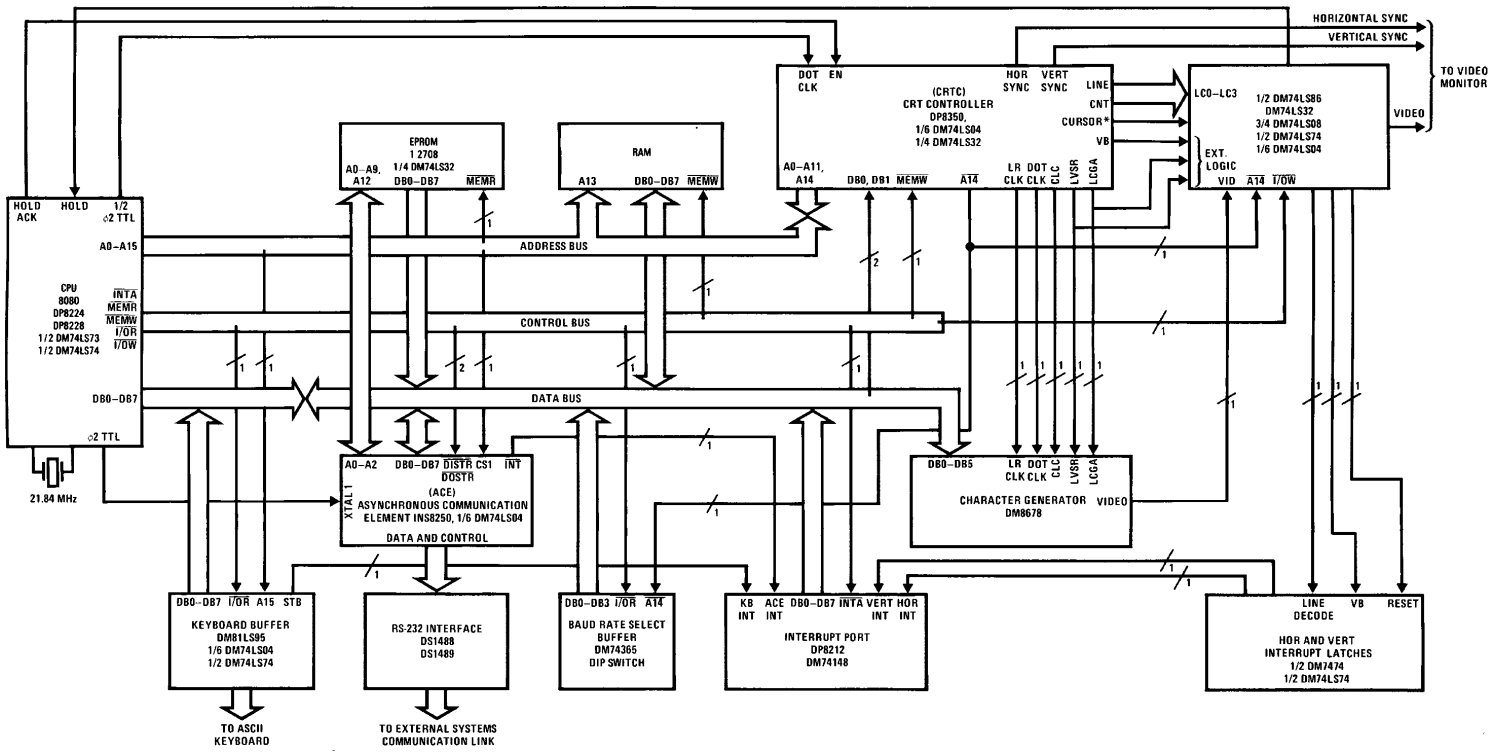
Application of the terminal's power supply resets the microprocessor, the communication element, and the CRT controller. Resetting the ACE is necessary to clear the interrupt. Resetting the CRTC is not absolutely necessary since the microprocessor loads the cursor and top of page registers in the initialization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

RRD-B30M105/Printed in U. S. A.



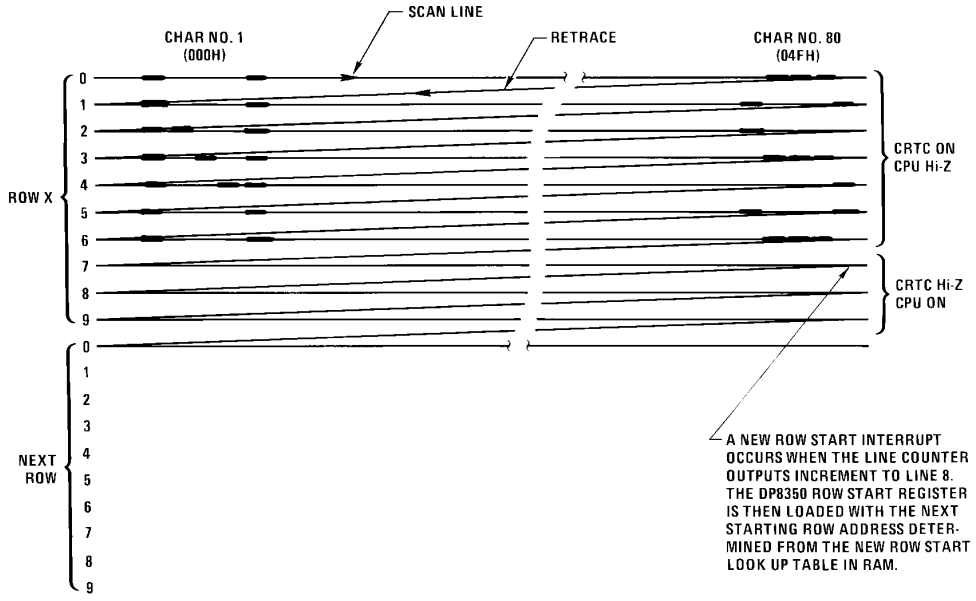
TL/F/5866–1

*The cursor is internally pipelined by the CRTC to allow for access time of the RAM and the character generator.

FIGURE 1. Video Data Terminal Detailed Block Diagram

Abbreviations:

LR CLK	Line Rate Clock
CLC	Clear Line Counter
LVSR	Load Video Shift Register
LCGA	Latch Character Generator Address
Line CNT	Line Counter
EN	Enable
VID	Video
KB INT	Keyboard Interrupt
VB	Vertical Blanking



TL/F/5866–2

FIGURE 2. Row Start Interrupting and Multiplexing the 8080 with the DP8350

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.

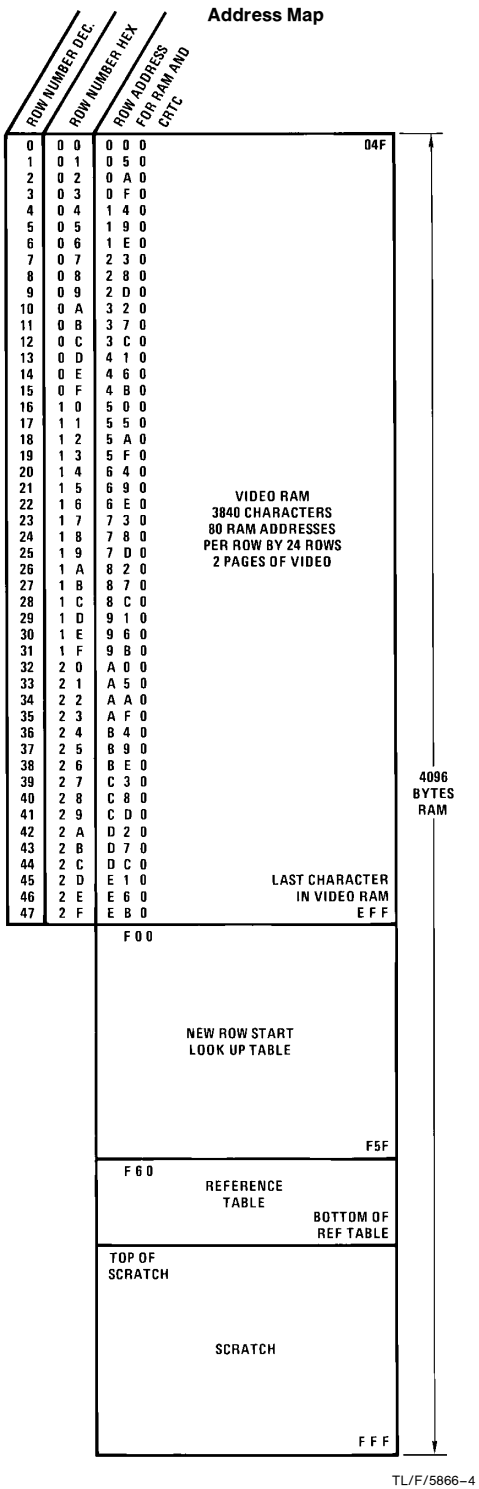
The row start look-up table, *(Figure 4),* is loaded up by a simple algorithm that loads and adds the data for referencing a row number to that row's starting address. The reference table, *(Figure 5),* is initialized next by direct loading. This table provides the CPU with top of page, bottom of page,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



TL/F/5866–4

FIGURE 3. RAM Organization

Memory Reference Tables

Page 1

Row Number		NRS High					NRS Low				
		Address				Row Data	Address				Row Data
Dec	Hex										
0	0 0	3	F	0	0	3 0	3	F	3	0	0 0
1	0 1	3	F	0	1	3 0	3	F	3	1	5 0
2	0 2	3	F	0	2	3 0	3	F	3	2	A 0
3	0 3	3	F	0	3	3 0	3	F	3	3	F 0
4	0 4	3	F	0	4	3 1	3	F	3	4	4 0
5	0 5	3	F	0	5	3 1	3	F	3	5	9 0
6	0 6	3	F	0	6	3 1	3	F	3	6	E 0
7	0 7	3	F	0	7	3 2	3	F	3	7	3 0
8	0 8	3	F	0	8	3 2	3	F	3	8	8 0
9	0 9	3	F	0	9	3 2	3	F	3	9	D 0
10	0 A	3	F	0	A	3 3	3	F	3	A	2 0
11	0 B	3	F	0	B	3 3	3	F	3	B	7 0
12	0 C	3	F	0	C	3 3	3	F	3	C	C 0
13	0 D	3	F	0	D	3 4	3	F	3	D	1 0
14	0 E	3	F	0	E	3 4	3	F	3	E	6 0
15	0 F	3	F	0	F	3 4	3	F	3	F	B 0
16	1 0	3	F	1	0	3 5	3	F	4	0	0 0
17	1 1	3	F	1	1	3 5	3	F	4	1	5 0
18	1 2	3	F	1	2	3 5	3	F	4	2	A 0
19	1 3	3	F	1	3	3 5	3	F	4	3	F 0
20	1 4	3	F	1	4	3 6	3	F	4	4	4 0
21	1 5	3	F	1	5	3 6	3	F	4	5	9 0
22	1 6	3	F	1	6	3 6	3	F	4	6	E 0
23	1 7	3	F	1	7	3 7	3	F	4	7	3 0

Page 2

Row Number		NRS High					NRS Low				
		Address				Row Data	Address				Row Data
Dec	Hex										
24	1 8	3	F	1	8	3 7	3	F	4	8	8 0
25	1 9	3	F	1	9	3 7	3	F	4	9	D 0
26	1 A	3	F	1	A	3 8	3	F	4	A	2 0
27	1 B	3	F	1	B	3 8	3	F	4	B	7 0
28	1 C	3	F	1	C	3 8	3	F	4	C	C 0
29	1 D	3	F	1	D	3 9	3	F	4	D	1 0
30	1 E	3	F	1	E	3 9	3	F	4	E	6 0
31	1 F	3	F	1	F	3 9	3	F	4	F	B 0
32	2 0	3	F	2	0	3 A	3	F	5	0	0 0
33	2 1	3	F	2	1	3 A	3	F	5	1	5 0
34	2 2	3	F	2	2	3 A	3	F	5	2	A 0
35	2 3	3	F	2	3	3 A	3	F	5	3	F 0
36	2 4	3	F	2	4	3 B	3	F	5	4	4 0
37	2 5	3	F	2	5	3 B	3	F	5	5	9 0
38	2 6	3	F	2	6	3 B	3	F	5	6	E 0
39	2 7	3	F	2	7	3 C	3	F	5	7	3 0
40	2 8	3	F	2	8	3 C	3	F	5	8	8 0
41	2 9	3	F	2	9	3 C	3	F	5	9	D 0
42	2 A	3	F	2	A	3 D	3	F	5	A	2 0
43	2 B	3	F	2	B	3 D	3	F	5	B	7 0
44	2 C	3	F	2	C	3 D	3	F	5	C	C 0
45	2 D	3	F	2	D	3 E	3	F	5	D	1 0
46	2 E	3	F	2	E	3 E	3	F	5	E	6 0
47	2 F	3	F	2	F	3 E	3	F	5	F	B 0

FIGURE 4. New Row Start Look Up Table

Function	Address	Data	Initialized Data
Last Row #	3F60	XY	17
8080 Row #	3F61	XY	00
First Row #	3F62	XY	00
Character #	3F63	XY	00
CRTC Row #	3F64	XY	00
Row Save #	3F65	XY	00
Temp. 1	3F66	XY	00
Temp. 2	3F67	XY	00

FIGURE 5. Reference Table

Command		Function
OUT	40	Clear new row start and vertical interrupt latches
IN	80	Read keyboard
IN	40	Read baud rate select switch

FIGURE 6. Input/Output Space

Device	Address
ROM	0000 to 0FFF
RAM	3000 to 3FFF
CRTC	5000 to 5FFF
ACE	9000 to 9007

*Direct device selecting was used to minimize the system component count.

FIGURE 7. CPU Addressing Space

Row Number		NRS High					NRS Low				
		Address				Row Data	Address				Row Data
Dec	Hex										
32	2 0	3	F	2	0	3 A	3	F	5	0	0 0

Row Start Address for Row 20H.

3XXX Selects RAM.

5XXX Selects CRTC.

FIGURE 8. Example from the New Row Start Look Up Table

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

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.

KEYBOARD INTERRUPT

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.

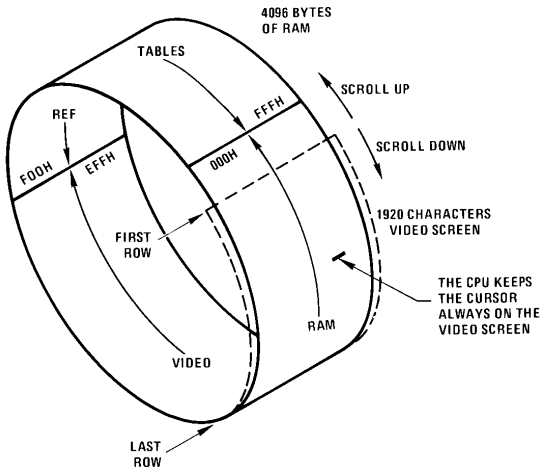
ACE INTERRUPT

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.



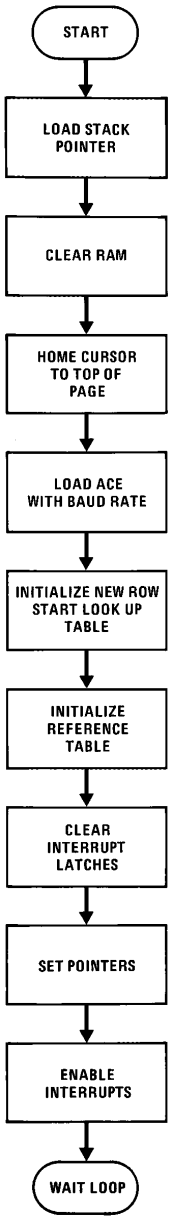
The video screen is allowed to scroll only through the video RAM (000H to EFFH). The CPU keeps the video screen within these bounds by loading the new row start register with that address range only (row 00H to 2FH).

TL/F/5866–5

FIGURE 9. Drum Analogy for the RAM

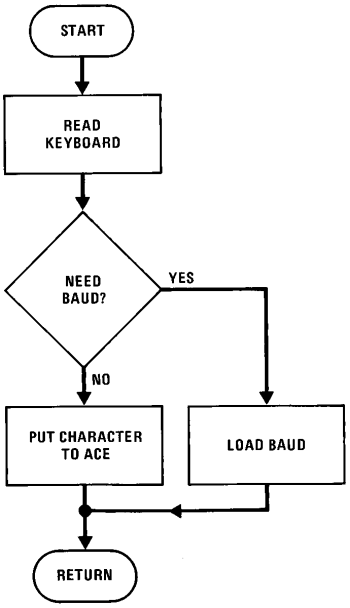
DP8350/8080 Video Data Terminal Basic Software Flow Chart

Initialization



TL/F/5866–6

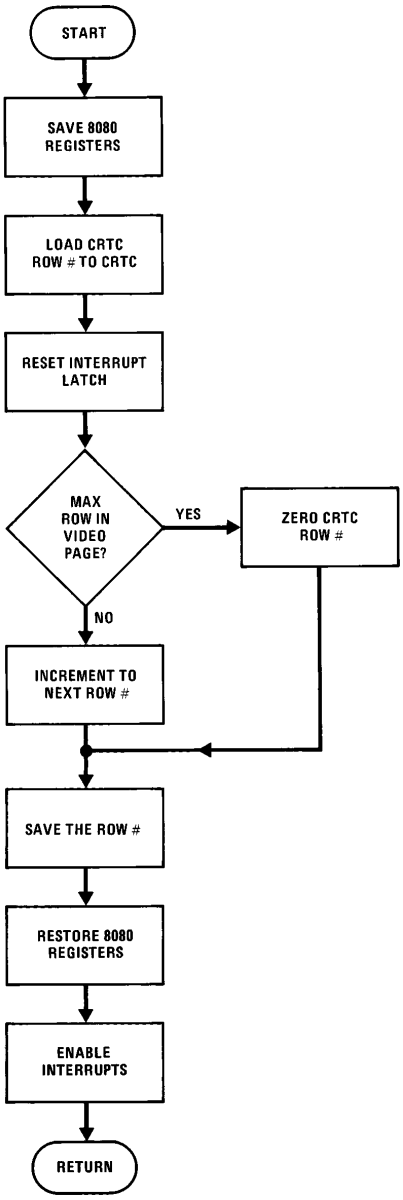
Keyboard Interrupt



TL/F/5866–7

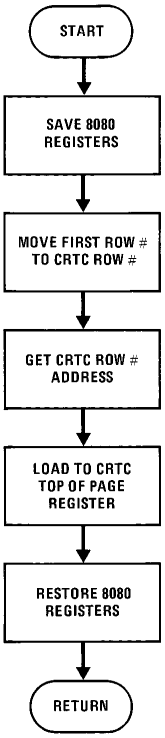
DP8350/8080 Video Data Terminal Basic Software Flow Chart (Continued)

New Row Start Interrupt



TL/F/5866–8

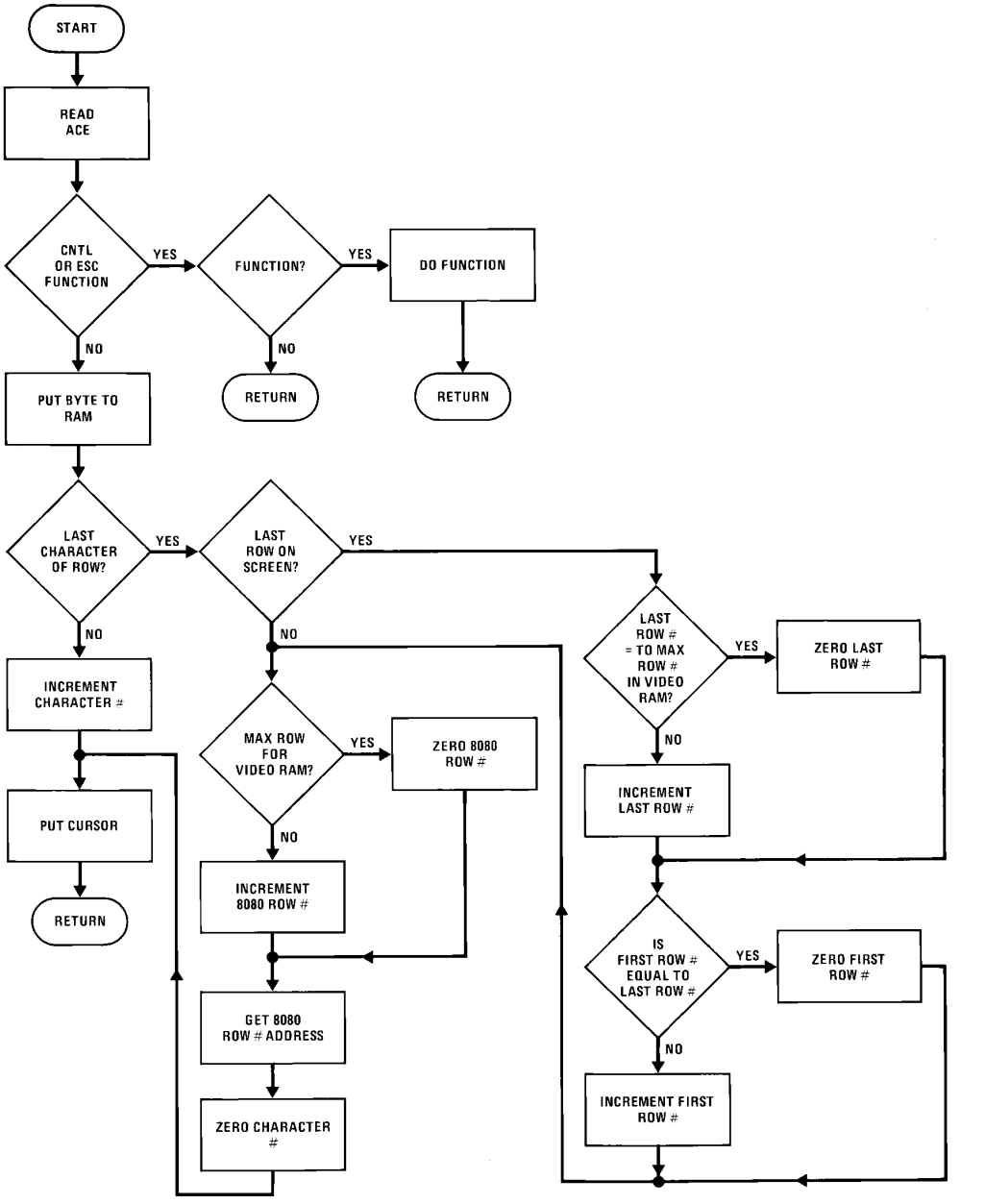
Vertical Interrupt



TL/F/5866–9

DP8350/8080 Video Data Terminal Basic Software Flow Chart (Continued)

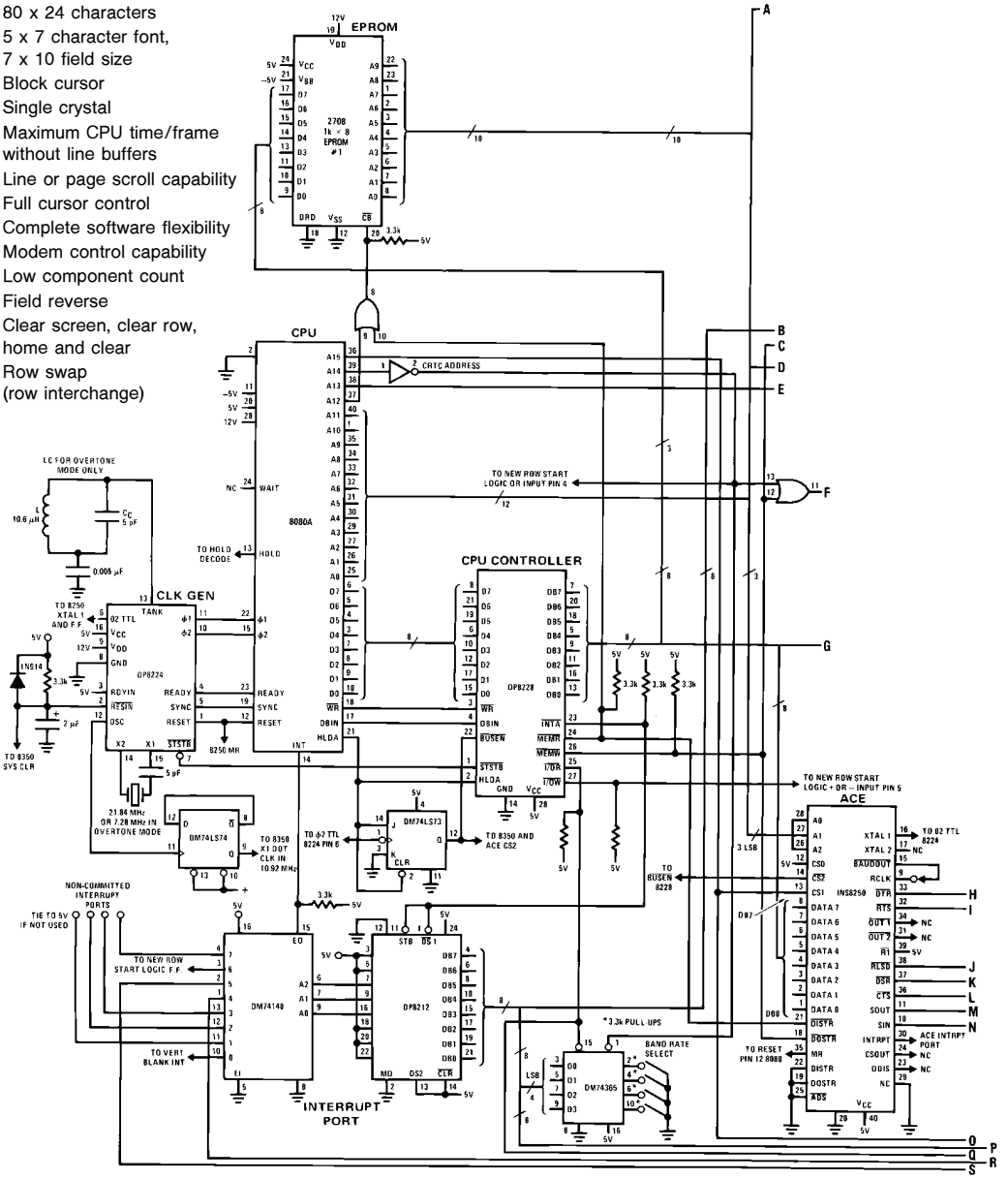
ACE Interrupt



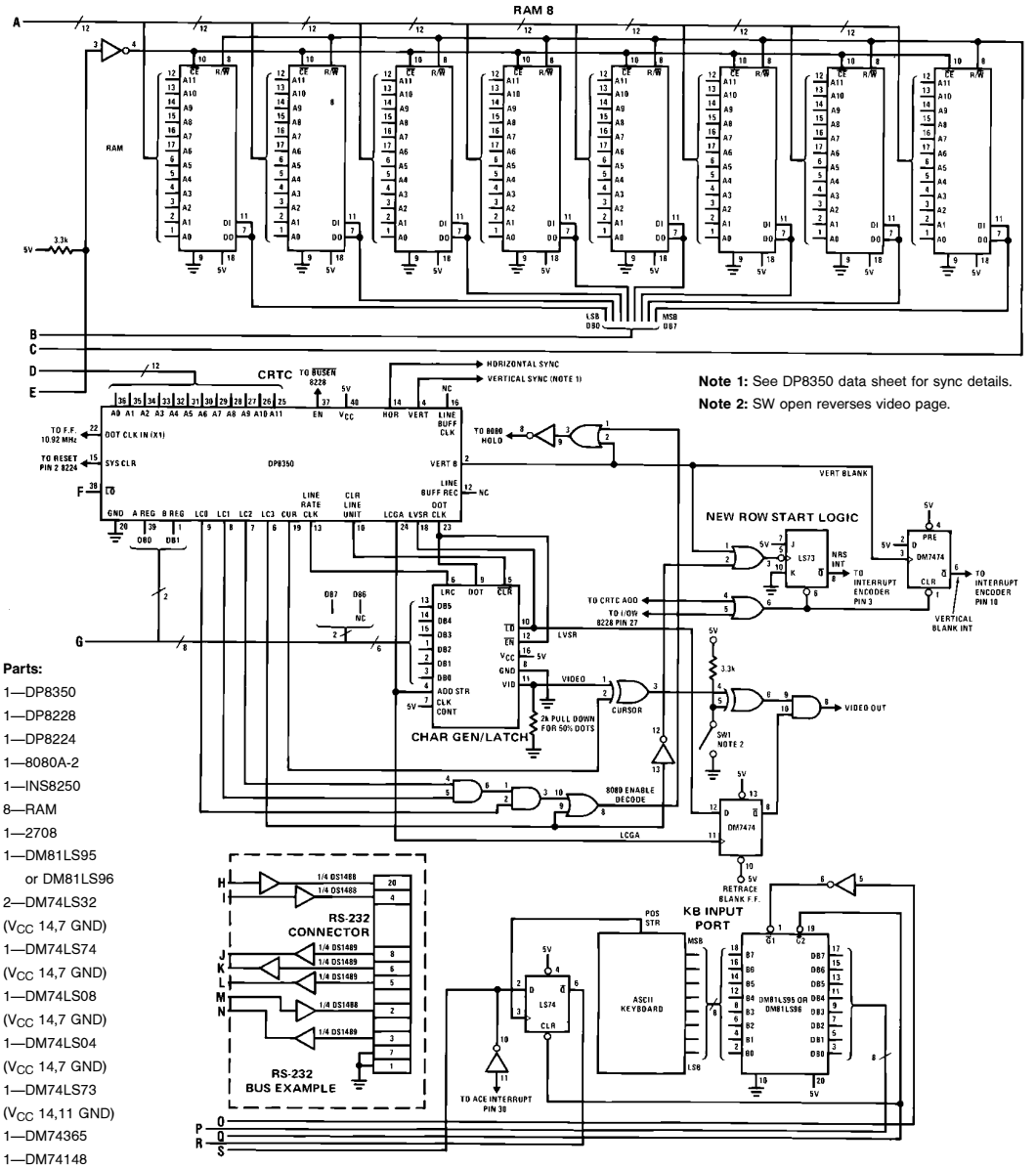
TL/F/5866–10

FEATURES

- Keyboard input port
- Serial I/O up to 9600 baud
  - 4 kbytes RAM
  - 1 kbyte ROM
- 2 video pages
- 80 x 24 characters
- 5 x 7 character font, 7 x 10 field size
- Block cursor
- Single crystal
- Maximum CPU time/frame without line buffers
- Line or page scroll capability
- Full cursor control
- Complete software flexibility
- Modem control capability
- Low component count
- Field reverse
- Clear screen, clear row, home and clear
- Row swap (row interchange)



TL/F/5866–11



Note 1: See DP8350 data sheet for sync details.

Note 2: SW open reverses video page.

Parts:

1—DP8350
1—DP8228
1—DP8224
1—8080A-2
1—INS8250
8—RAM
1—2708
1—DM81LS95
or DM81LS96
2—DM74LS32
(V_{CC} 14,7 GND)
1—DM74LS74
(V_{CC} 14,7 GND)
1—DM74LS08
(V_{CC} 14,7 GND)
1—DM74LS04
(V_{CC} 14,7 GND)
1—DM74LS73
(V_{CC} 14,11 GND)
1—DM74365
1—DM74148
1—DM7474
1—DP8212
1—DM74LS86
1—char. gen/latch
2—Res. arrays,
3.3k
1—21.84 MHz
Xtal
Bypass capacitors
on all parts

TL/F/5866–12

```
                        .TITLE CRTC ,'8080A 02/15/78'

              ;** NATIONAL SEMICONDUCTOR'S
              ;   SERIES PROGRAMMABLE CRT CONTROLLER BOARD **

              ;AL BRILLIOTT-JIM TROUTNER

      0060    LASTROW =               060
      0061    ROW8080 =               061
      0062    FIRSTRO =               062
      0063    CHARNUM =               063
      0064    CRTCROW =               064
      0065    ROWSAVE =               065
      0066    TEMP1   =               066
      0067    TEMP2   =               067
      0068    IMASK   =               068

0000                  .=0000
0000 F3      START:   DI                      ;DISABLE INTERRUPTS
0001 31FF3F           LXI     SP,03FFF         ;LOAD STACK POINTER
0004 C33B00           JMP     INIT             ;JUMP TO INITILIZE ROUTINE
0007                  .=0008
0008 C32502           JMP     NEWRO            ;NEW ROW START INTERRUPT
000B                  .=0010
0010 C34A01           JMP     INTACE           ;ACE INTERUPT
0013                  .=0018
0018 C33601           JMP     INTKB            ;KEYBOARD INTERUPT
001B                  .=0038
0038 C34F02           JMP     VERTI            ;VERTICAL INTERUPT
003B 210030  INIT:    LXI     H,03000          ;1ST RAM ADDRESS
003E 0E20             MVI     C,020            ;ASCII SPACE INTO C REG
0040 3E3F             MVI     A,03F            ;MAX RAM ADDRESS
0042 71      CLRAM:   MOV     M,C              ;ASCII SPACE INTO MEM
0043 23               INX     H                ;NEXT RAM ADDRESS
0044 BC               CMP     H                ;MAX RAM ADDRESS?
0045 C24200           JNZ     CLRAM            ;IF NO THEN NEXT ADD.
0048 0E00             MVI     C,000
004A 3E40             MVI     A,040
004C 71      CLRAM1:  MOV     M,C
004D 23               INX     H
004E BC               CMP     H
004F C24C00           JNZ     CLRAM1
0052 CD8700           CALL    HMCUR            ;GO TO CUR HOME ROUTINE
0055 CD9300           CALL    BAUD             ;GO TO BAUD LOAD ROUTINE

                      ;NEW ROW START LOOK UP TABLE GENERATION

0058 21003F           LXI     H,03F00          ;N.R.S.  HIGH ADDRESS
005B 11303F           LXI     D,03F30          ;N.R.S.  LOW ADDRESS
005E 010030           LXI     B,03000          ;N.R.S.  ADDRESS DATA
0061 70      NRS:     MOV     M,B              ;STORE TO N.R.S.  DATA TABLE ''
0062 79               MOV     A,C              ;N.R.S.  DATA LOW TO ACC.
0063 12               STAX    D                ;STORE TO N.R.S.  DATA TABLE L
0064 C650             ADI     050              ;ACC READY FOR NEXT LOAD
0066 4F               MOV     C,A              ;ACC TO N.R.S.  DATA HIGH
0067 78               MOV     A,B              ;N.R.S.  DATA TO ACC
0068 CE00             ACI     000              ;ADD CARRY BIT TO DATA HIGH
006A 47               MOV     B,A              ;MOVE RESULT TO N.R.S.  DATA H
006B 2C               INR     L                ;INCREMENT N.R.S.  HIGH ADD
006C 1C               INR     E                ;INCREMENT N.R.S.  LOW ADD
006D 7B               MOV     A,E              ;N.R.S.  ADD LOW TO ACC
006E FE60             CPI     LASTROW          ;MAX TABLE ADDRESS
0070 C26100           JNZ     NRS              ;IF FALSE JUMP

                      ;REFERENCE TABLE INITILIZE

0073 3E17             MVI     A,017            ;LAST ROW NUMBER TO ACC.
0075 12               STAX    D                ;STORE TO REFERENCE TABLE

                      ;CLEAR PERIPHERAL INTERRUPT FLOPS

0076 D340             OUT     040              ;N.R.S.  INTERRUPT CLEAR
0078 DB80             IN      080              ;KEYBOARD INTERRUPT CLEAR

                      ;SET UP POINTERS

007A 11603F           LXI     D,03F60          ;POINT D-E TO REFERENCE TABLE
007D 210030           LXI     H,03000          ;POINT H-L TO 1ST RAM LOCATI^
0080 010090           LXI     B,09000          ;POINT B-C TO ACE

                      ;WAIT LOOP FOR INTERUPTS

0083 FB      BACK:    EI                       ;ENABLE INTERRUPTS
0084 C38300           JMP     BACK             ;LOOP UNTIL INTERRUPTED

                      ;HOME UP CURSOR

0087 210050  HMCUR:   LXI     H,05000          ;POINT B-C TO CRTC
008A 3E02             MVI     A,002            ;T.O.P.  REGISTER SELECT
008C 77               MOV     M,A              ;T.O.P.  LOAD
008D 3C               INR     A                ;CURSOR REGISTER SELECT
008E 77               MOV     M,A              ;CURSOR LOADS TO T.O.P.
008F 210030           LXI     H,03000          ;POINT H-L TO 1ST RAM ADD.
0092 C9               RET                      ;RETURN
```

TL/F/5866–13

Continued Next Page

```
 96                             ;BAUD RATE SELECT
 97
 98 0093 D5      BAUD:   PUSH    D               ;SAVE D-E REGISTERS
 99 0094 DB40            IN      040             ;READ BAUD SELECT CODE
100 0096 E60F            ANI     00F             ;ZERO THE HIGH ORDER 4 BITS
101 0098 FE00            CPI     000
102 009A CAD400          JZ      B110            ;110 BAUD ROUTINE
103 009D FE01            CPI     001
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
108 00A9 CAE600          JZ      B600            ;600 BAUD ROUTINE
109 00AC FE04            CPI     004
110 00AE CAEC00          JZ      B1200           ;1200 BAUD ROUTINE
111 00B1 FE05            CPI     005
112 00B3 CAF200          JZ      B1800           ;1800 BAUD ROUTINE
113 00B6 FE06            CPI     006
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           ;3600 BAUD ROUTINE
119 00C5 FE09            CPI     009
120 00C7 CA0A01          JZ      B4800           ;4800 BAUD ROUTINE
121 00CA FE0A            CPI     00A
122 00CC CA1001          JZ      B7200           ;7200 BAUD ROUTINE
123 00CF FE0B            CPI     00B
124 00D1 CA1601          JZ      B9600           ;9600 BAUD ROUTINE
125
126                             ;BAUD RATE SET UP ROUTINES
127
128 00D4 116305 B110:   LXI     D,00563         ;110 BAUD DIVISOR
129 00D7 C31C01          JMP     ACELD           ;GO TO ACE LOAD ROUTINE
130 00DA 11F303 B150:   LXI     D,003F3         ;150 BAUD DIVISOR
131 00DD C31C01          JMP     ACELD
132 00E0 11F901 B300:   LXI     D,001F9         ;300 BAUD DIVISOR
133 00E3 C31C01          JMP     ACELD
134 00E6 11FC00 B600:   LXI     D,000FC         ;600 BAUD DIVISOR
135 00E9 C31C01          JMP     ACELD
136 00EC 117E00 B1200:  LXI     D,0007E         ;1200 BAUD DIVISOR
137 00EF C31C01          JMP     ACELD
138 00F2 115400 B1800:  LXI     D,00054         ;1800 BAUD DIVISOR
139 00F5 C31C01          JMP     ACELD
140 00F8 114C00 B2000:  LXI     D,0004C         ;2000 BAUD DIVISOR
141 00FB C31C01          JMP     ACELD
142 00FE 113F00 B2400:  LXI     D,0003F         ;2400 BAUD DIVISOR
143 0101 C31C01          JMP     ACELD
144 0104 112A00 B3600:  LXI     D,0002A         ;3600 BAUD DIVISOR
145 0107 C31C01          JMP     ACELD
146 010A 112000 B4800:  LXI     D,00020         ;4800 BAUD DIVISOR
147 010D C31C01          JMP     ACELD
148 0110 111500 B7200:  LXI     D,00015         ;7200 BAUD DIVISOR
149 0113 C31C01          JMP     ACELD
150 0116 111000 B9600:  LXI     D,00010         ;9600 BAUD DIVISOR
151 0119 C31C01          JMP     ACELD
152
153                             ;ACE LOAD ROUTINE
154
155 011C 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           ;POINT TO BAUD HIGH
159 0124 7A              MOV     A,D             ;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             ;GET BAUD LOW
163 0129 02              STAX    B               ;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               ;PUT TO ACE
167 012E 0E01            MVI     C,001           ;INTERRUPT ENABLE REG
168 0130 79              MOV     A,C             ;SELECT RECEIVED DATA INTERR
169 0131 02              STAX    B               ;LOAD IT
170 0132 0E00            MVI     C,000           ;RESTORE B-C ACE POINTER
171 0134 D1              POP     D               ;RESTORE D-E REGISTERS
172 0135 C9              RET                     ;RETURN
173
174                             ;KEYBOARD INTERRUPT ROUTINE
175
176 0136 DB80   INTKB:  IN      080             ;READ KEYBOARD
177 0138 FB              EI                      ;ENABLE INTERRUPTS
178 0139 FE05            CPI     005             ;NEED BAUD RATE? (CNTL E)
179 013B 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                     ;RETURN
186
```

TL/F/5866–14

Continued Next Page

```
187                     ;ACE INTERUPT ROUTINE
188
189 014A 0A      INTACE: LDAX    B               ;LOAD ACE DATA BYTE TO ACC.
190 014B FB             EI                      ;ENABLE INTERRUPTS
191 014C FE7E           CPI     07E
192 014E CA7001         JZ      FUNC            ;TEST FOR ESC COMAMD
193 0151 FE7F           CPI     07F             ;TEST FOR DEL COMAND
194 0153 CA7001         JZ      FUNC
195 0156 5F             MOV     E,A             ;SAVE CHAR IN REG.E
196 0157 E660           ANI     060             ;MASK OUT BITS FOR CNTL TEST
197 0159 CA7001         JZ      FUNC            ;IF ZERO JMP TO CNTL FUNC
198 015C 3A683F         LDA     03F68           ;LOAD INVERT MASK
199 015F B3             ORA     E               ;OR MASK AND CHAR
200 0160 77             MOV     M,A             ;STORE DATA BYTE TO RAM
201
202                     ;ADVANCE CURSOR
203
204 0161 1E63   ADCUR:  MVI     E,CHARNUM       ;POINT B-C TO CHAR #
205 0163 1A             LDAX    D               ;LOAD CHAR # TO ACC.
206 0164 23             INX     H               ;NEXT CHAR LOCATION
207 0165 FE4F           CPI     04F             ;LAST CHAR OF ROW?
208 0167 CABE01         JZ      NXRO            ;IF TRUE JUMP TO NEXT ROW
209 016A C601           ADI     001             ;INCREMENT CHAR #
210 016C 12             STAX    D               ;STORE CHAR # TO RAM REF.
211 016D C3B301         JMP     PCUR            ;PUT CURSOR
212
213                     ;TEST FOR FUNCTION
214
215 0170 7B     FUNC:   MOV     A,E
216 0171 FE01           CPI     001             ;HOME AND CLEAR CNTL A (SOH)
217 0173 CA0000         JZ      START           ;
218 0176 FE0D           CPI     00D             ;CARRAGE RETURN
219 0178 CA6E02         JZ      CR
220 017B FE11           CPI     011             ;SAVE ROW # CNTL Q (DC1)
221 017D CA7B02         JZ      SAVRO
222 0180 FE0C           CPI     00C             ;ADVANCE CURSOR CNTL L (FF)
223 0182 CA6101         JZ      ADCUR
224 0185 FE02           CPI     002             ;HOME UP CNTL B (STX)
225 0187 CAA402         JZ      HOME
226 018A FE1A           CPI     01A             ;SWAP   CNTL Z (SUB)
227 018C CAB502         JZ      SWAP
228 018F FE0A           CPI     00A             ;LINEFEED
229 0191 CA8D02         JZ      LF
230 0194 FE08           CPI     008             ;BACKSPACE CNTL H (BS)
231 0196 CAE002         JZ      BS
232 0199 FE0B           CPI     00B             ;UP CURSOR CNTL K (VT)
233 019B CAF102         JZ      UPCUR
234 019E FE18           CPI     018             ;CLEAR ROW CNTL X (CAN)
235 01A0 CA3003         JZ      CLROW
236 01A3 FE07           CPI     007             ;RING BELL CNTL G (BEL)
237 01A5 CA4503         JZ      BELL
238 01A8 FE12           CPI     012             ;INVERT NEXT CNTL R (DC2)
239 01AA CA4803         JZ      IVERTN
240 01AD FE13           CPI     013             ;INVERT ROW CNTL S (DC3)
241 01AF CA5403         JZ      IVERTR
242 01B2 C9             RET                     ;RETURN
243
244                     ;STORE CURSOR TO CRTC FROM H-L REGISTERS
245
246 01B3 7C     PCUR:   MOV     A,H             ;H REG TO ACC.
247 01B4 C620           ADI     020             ;SET H-L REG TO CRTC ADD.
248 01B6 67             MOV     H,A             ;H IS CRTC ADD.
249 01B7 3603           MVI     M,003           ;CURSOR REGISTER SELECT
250 01B9 7C             MOV     A,H             ;H REG SET BACK TO VIDIO RAM
251 01BA D620           SUI     020             ;ADDRESS
252 01BC 67             MOV     H,A
253 01BD C9             RET                     ;RETURN
254
255
256                     ;LAST ROW ON SCREEN
257
258 01BE CDDC01 NXRO:   CALL    NXRO1           ;GO TO NEXT ROW SUBROUTINE
259 01C1 CDF301         CALL    ZCHAR           ;ZERO CHARACTER
260 01C4 E5     CLROW3: PUSH    H               ;SAVE H,L
261 01C5 1E60           MVI     E,LASTROW       ;POINT D,E TO LASTROW
262 01C7 1A             LDAX    D
263 01C8 C601           ADI     001             ;POINT AC TO FIRST ROW OFF SC
264 01CA FE30           CPI     030             ;CK IF LAST ROW IN RAM
265 01CC CAD701         JZ      ROZERO
266 01CF CD8302 LOOP5:  CALL    LDHL1           ;LOAD H,L WITH ADD.  OF LASTRO
267 01D2 CD3B03         CALL    CLROW2
268 01D5 E1             POP     H               ;RESTORE H,L
269 01D6 C9             RET
270
271 01D7 3E00   ROZERO: MVI     A,000           ;LOAD ROW ZERO
272 01D9 C3CF01         JMP     LOOP5
273
```

TL/F/5866–15

Continued Next Page

```
                        ;NEXT ROW
                        
01DC 1E60    NXRO1:     MVI     E,LASTROW       ;POINT D-E REG TO LAST ROW
01DE 1A                 LDAX    D               ;PUT LAST ROW # TO ACC.
01DF EB                 XCHG                    ;EXCHANGE H-L WITH D-E
01E0 23                 INX     H               ;H-L IS NOW AT 8080 ROW #
01E1 BE                 CMP     M               ;COMPARE LAST ROW # WITH
01E2 CA0502             JZ      SCROLL          ;8080 ROW # IF TRUE SCROLL
                        
                        ;INCREMENT 8080 ROW #
                        
01E5 3E2F    INCRO:     MVI     A,02F           ;TEST FOR MAX ROW AND
01E7 BE                 CMP     M               ;JUMP TO ZERO ROW IF TRUE
01E8 CAFB01             JZ      ZROW            ;ZERO ROW
01EB 34                 INR     M               ;INCREMENT THE 8080 ROW #
01EC EB                 XCHG                    ;POINT H-L TO CHAR #
01ED 1E61               MVI     E,ROW8080
01EF CD8202             CALL    LDHL
01F2 C9                 RET                     ;RETURN
                        
                        ;ZERO CHARACTER
                        
01F3 3E00    ZCHAR:     MVI     A,000           ;PUT CHAR # TO ZERO
01F5 32633F             STA     03F63           ;AND STORE
01F8 C3B301             JMP     PCUR            ;GO TO PUT CURSOR ROUTINE
                        
                        ;ZERO 8080 ROW #
                        
01FB 3600    ZROW:      MVI     M,000           ;8080 ROW # TO ZERO
01FD 2E00               MVI     L,000           ;N.R.S.  ADDRESS HIGH
01FF 56                 MOV     D,M             ;N.R.S.  DATA HIGH TO D REG
0200 2E30               MVI     L,030           ;N.R.S.  ADDRESS LOW
0202 5E                 MOV     E,M             ;N.R.S.  DATA LOW TO E REG
0203 EB                 XCHG                    ;EXCHANGE H-L WITH D-E
0204 C9                 RET                     ;RETURN
                        
                        ;ROW SCROLL
                        
0205 2B      SCROLL:    DCX     H               ;POINT H-L TO LAST ROW#
0206 3E2F               MVI     A,02F           ;BEFORE SCRATCH TABLES.
0208 BE                 CMP     M               ;TEST FOR THE LAST ROW.
0209 CA1902             JZ      ZLRO            ;JUMP TO ZERO LAST ROW IF TR
020C 34                 INR     M               ;INCREMENT TO NEXT ROW.
                        
                        
                        
020D 2E62    ROLO:      MVI     L,FIRSTRO       ;POINT H-L TO FIRST ROW#
020F BE                 CMP     M               ;IS FIRST LOW = TO LAST ROW.
0210 CA1E02             JZ      ZFRO            ;JUMP TO ZERO FIRST R
0213 34                 INR     M               ;INCREMENT TO NEXT ROW
0214 2E61               MVI     L,ROW8080       ;POINT H-L TO 8080 ROW
0216 C3E501             JMP     INCRO           ;GO TO INCREMENT ROW ROUTINE
                        
                        
                        
0219 3600    ZLRO:      MVI     M,000           ;PUT LAST ROW# TO ZERO
021B C30D02             JMP     ROLO            ;GO TO ROUTINE FOR FIRST ROW
                        
                        
021E 3600    ZFRO:      MVI     M,000           ;PUT FIRST ROW# TO ZERO
0220 2E61               MVI     L,ROW8080       ;POINT H-L TO 8080 ROW
0222 C3E501             JMP     INCRO           ;GO TO INCREMENT ROW ROUTINE
                        
                        ;NEW ROW START INTERRUPT
                        
0225 F5      NEWRO:     PUSH    PSW             ;SAVE ACC AND FLAGS
0226 E5                 PUSH    H               ;SAVE H-L REG
0227 D5                 PUSH    D
0228 11643F             LXI     D,03F64         ;POINT D-E TO CRTCROW #
022B 1A                 LDAX    D               ;LOAD ACC WITH CRTC ROW #
022C 5F                 MOV     E,A             ;N.R.S.  DATA ADD HIGH TO E
022D 1A                 LDAX    D               ;ROW DATA HIGH INTO ACC
022E C620               ADI     020
0230 67                 MOV     H,A             ;N.R.S.  DATA ADD HIGH INTO H
0231 7B                 MOV     A,E
0232 C630               ADI     030             ;ACC TO N.R.S.  DATA LOW
0234 5F                 MOV     E,A             ;N.R.S.  DATA ADD LOW TO E REG
0235 1A                 LDAX    D               ;ROW DATA LOW TO ACC
0236 6F                 MOV     L,A             ;N.R.S DATA ADD LOW INTO L
0237 3601               MVI     M,001           ;STORE N.R.S.  TO CRTC
0239 D340               OUT     040             ;RESET N.R.S.  AND VERT INTER
023B 1E64               MVI     E,CRTCROW
023D 1A                 LDAX    D
023E FE2F               CPI     02F             ;TEST FOR CRTC MAX ROW
0240 CA4A02             JZ      ZCRTC           ;IF TRUE ZERO ACC
0243 3C                 INR     A               ;INCREMENT TO NEXT ROW
0244 12      LOOP:      STAX    D               ;STORE NEXT ROW NUMBER
0245 D1                 POP     D
0246 E1                 POP     H               ;RESTORE H-L REG
0247 F1                 POP     PSW             ;RESTORE ACC AND FLAGS
0248 FB                 EI
0249 C9                 RET                     ;RETURN
                        
                        
```

TL/F/5866–16

Continued Next Page

```
                        ;ZERO CRTCROW
                        
024A 3E00    ZCRTC:     MVI     A,000           ;ZERO ACC
024C C34402             JMP     LOOP
                        
                        ;VERTICAL INTERRUPT
                        
024F F5      VERTI:     PUSH    PSW             ;SAVE ACC AND FLAGS
0250 E5                 PUSH    H               ;SAVE H REG
0251 D5                 PUSH    D
0252 1E62               MVI     E,FIRSTRO       ;POINT D-E TO FIRST ROW #
0254 1A                 LDAX    D               ;LOAD 1ST ROW # INTO ACC
0255 1E64               MVI     E,CRTCROW       ;POINT D-E TO CRTCROW #
0257 12                 STAX    D               ;UPDATE CRTCROW #
0258 E63F               ANI     03F             ;REMOVE MARKER
025A 5F                 MOV     E,A             ;POINT H L TO CRTC FIRST ROW
025B 1A                 LDAX    D
025C C620               ADI     020
025E 67                 MOV     H,A
025F 7B                 MOV     A,E
0260 C630               ADI     030
0262 5F                 MOV     E,A
0263 1A                 LDAX    D
0264 6F                 MOV     L,A
0265 3602               MVI     M,002           ;STORE TOP OF PAGE
0267 D340               OUT     040
0269 D1                 POP     D
026A E1                 POP     H
026B F1                 POP     PSW             ;RESTORE ACC AND FLAGS
026C FB                 EI
026D C9                 RET                     ;RETURN
                        
                        ;CARRAGE RETURN
                        
026E 1E63    CR:        MVI     E,CHARNUM       ;POINT D-E TO CHAR #
0270 3E00               MVI     A,000
0272 12                 STAX    D
0273 1E61               MVI     E,ROW8080
0275 CD8202             CALL    LDHL
0278 C3B301             JMP     PCUR            ;CURSOR TO THE BEGINNING OF R
                        
                        ;SAVE ROW
                        
027B 1E61    SAVRO:     MVI     E,ROW8080       ;POINT D-E TO 8080 ROW#
027D 1A                 LDAX    D               ;PUT 8080 ROW # TO ACC
027E 1E65               MVI     E,ROWSAVE       ;POINT D-E TO ROW SAVE
0280 12                 STAX    D               ;STORE ROW SAVE # IN REF TAB'
0281 C9                 RET                     ;RETURN
                        
             ;          H-L ROW DATA LOAD ROUTINE
                        
0282 1A      LDHL:      LDAX    D               ;LOAD ACC WITH D-E DATA
0283 5F      LDHL1:     MOV     E,A             ;POINT D-E TO N.R.S.  DATA HI'
0284 1A                 LDAX    D               ;ROW # TO N.R.S.  DATA HIGH
0285 67                 MOV     H,A             ;ROW # TO H REG
0286 7B                 MOV     A,E             ;PUT 1ST ROW # TO ACC
0287 C630               ADI     030             ;ACC TO N.R.S.  ADD LOW
0289 5F                 MOV     E,A             ;POINT D-E TO N.R.S.  DATA LOW
028A 1A                 LDAX    D               ;ROW # TO N.R.S.DATA LOW
028B 6F                 MOV     L,A             ;ROW # TO L REG
028C C9                 RET                     ;RETURN
                        
                        ;LINEFEED
                        
028D CDDC01  LF:        CALL    NXRO1           ;DO NEXT ROW SUBROUTINE
0290 CDC401             CALL    CLROW3          ;OFF SCREEN CLEAR ROW ROUTINE
0293 1E61               MVI     E,ROW8080       ;MOVE REFERENCE ROW # TO H-L
0295 CD8202             CALL    LDHL            ;LOAD H-L
0298 3A633F  ADDCH:     LDA     03F63           ;CHAR # TO ACC
029B 85                 ADD     L               ;ADD THE CHAR # TO THE
029C 6F                 MOV     L,A             ;FIRST ROW ADDRESS.
029D 7C                 MOV     A,H             ;IF A CARRY OCCURED ADD TO
029E CE00               ACI     000             ;THE DATA HIGH
02A0 67                 MOV     H,A             ;H-L POINTS TO LINE FED ROW
02A1 C3B301             JMP     PCUR            ;PUT CURSOR TO LINE FED ROW
                        
                        ;HOME CURSOR TO T.O.P.
                        
02A4 1E62    HOME:      MVI     E,FIRSTRO       ;POINT D-E TO 1ST ROW
02A6 1A                 LDAX    D               ;STORE FIRSTROW TO ROW8080
02A7 1E61               MVI     E,ROW8080
02A9 12                 STAX    D
02AA CD8302             CALL    LDHL1           ;MOVE REFERENCE ROW TO H-L
02AD 3E00               MVI     A,000           ;PUT CHAR # BACK
02AF 323F63             STA     03F63           ;TO ZERO
02B2 C3B301             JMP     PCUR            ;PUT CURSOR HOME
```

TL/F/5866–17

Continued Next Page

```
455                             ;SWAP ROWS
456
457 02B5 1E65    SWAP:   MVI     E,ROWSAVE       ;POINT D-E TO ROW SAVE # AND
458 02B7 CD8202         CALL    LDHL            ;PUT IN H-L REG.
459 02BA 22663F         SHLD    03F66           ;STORE ROW SAVE # TO TEMP 1 *
460 02BD 1E61           MVI     E,ROW8080       ;POINT D-E TO 8080 ROW # AND
461 02BF CD8202         CALL    LDHL            ;PUT ADDRESS IN H-L REG
462 02C2 1E65           MVI     E,ROWSAVE       ;POINT D-E TO ROW SAVE # AND
463 02C4 1A             LDAX    D               ;PUT IN ACC
464 02C5 5F             MOV     E,A             ;8080 ROW # TO ADD HIGH
465 02C6 7C             MOV     A,H             ;STORE 8080 ROW # TO N.R.S.
466 02C7 12             STAX    D               ;DATA HIGH.
467 02C8 7B             MOV     A,E
468 02C9 C630           ADI     030
469 02CB 5F             MOV     E,A             ;PUT 8080 ROW # TO
470 02CC 7D             MOV     A,L             ;N.R.S.  DATA LOW
471 02CD 12             STAX    D               ;8080 ROW # IS NOW IN ROW SA'
472 02CE 2A663F         LHLD    03F66           ;PUT ROW SAVE # BACK TO H-L
473 02D1 1E61           MVI     E,ROW8080       ;COMENT SAME AS ABOVE
474 02D3 1A             LDAX    D
475 02D4 5F             MOV     E,A
476 02D5 7C             MOV     A,H
477 02D6 12             STAX    D
478 02D7 7B             MOV     A,E
479 02D8 C630           ADI     030
480 02DA 5F             MOV     E,A
481 02DB 7D             MOV     A,L
482 02DC 12             STAX    D
483 02DD C39802         JMP     ADDCH           ;JUMP TO ADD CHAR.
484
485                             ;BACK SPACE
486
487 02E0 1E63    BS:    MVI     E,CHARNUM       ;POINT THE D-E REG TO CHAR #
488 02E2 1A             LDAX    D               ;AND PUT IN ACC
489 02E3 FE00           CPI     000             ;TEST FOR THE CHAR # =
490 02E5 CAEE02         JZ      UPROW           ;TO ZERO.  JUMP IF TRUE
491 02E8 3D             DCR     A               ;DECREMENT CHAR #
492 02E9 12             STAX    D               ;STORE DECREMENTED CHAR #
493 02EA 2B             DCX     H               ;DEC H-L FOR NEW CURSOR LOCA)
494 02EB C3B301         JMP     PCUR            ;PUT CURSOR IN DECREMENTED LO
495
496                             ;NEXT ROW UP
497
498 02EE 3E4F    UPROW: MVI     A,04F           ;MOVE THE CHAR #
499 02F0 12             STAX    D               ;TO 50H AND STORE IT.
500
501                             ;MOVE CURSOR UP
502
503 02F1 EB      UPCUR: XCHG                    ;POINT H-L TO 8080 ROW AND D-
504 02F2 2E61           MVI     L,ROW8080       ;TO NEW CURSOR LOCATION.
505 02F4 7E             MOV     A,M             ;TEST IF NEXT UP CURSOR WILL
506 02F5 23             INX     H               ;BE ON THE FIRST ROW.
507 02F6 BE             CMP     M               ;IF TRUE JUMP TO
508 02F7 CA0803         JZ      UPSCL           ;UP SCROLL ROUTINE.
509 02FA 2B             DCX     H               ;POINT H-L BACK TO 8080 ROW #
510
511 02FB FE00    BACK1: CPI     000             ;IF 8080 ROW # IS EQUAL TO
512 02FD CA1E03         JZ      R048            ;ZERO JUMP TO ROW 48 ROUTINE.
513 0300 35             DCR     M               ;DECREMENT 8080 ROW #
514
515 0301 EB      LOOP1: XCHG                    ;POINT H-L TO NEW CURSOR LOCA
516 0302 CD8202         CALL    LDHL            ;AND D-E TO 8080 ROW #.  JUMP
517 0305 C39802         JMP     ADDCH           ;TO ADD CHARACTER ROUTINE.
518
519 0308 7E      UPSCL: MOV     A,M             ;PUT FIRST ROW # INTO ACC.
520 0309 FE00           CPI     000             ;TEST IF FIRST ROW # IS = TO
521 030B CA2403         JZ      FR048           ;ZERO.  IF TRUE JUMP TO ROW
522 030E 35             DCR     M               ;48 ROUTINE.
523
524 030F 2E60    LOOP2: MVI     L,LASTROW
525 0311 7E             MOV     A,M
526 0312 FE00           CPI     000
527 0314 CA2A03         JZ      LR048
528 0317 35             DCR     M
529
530 0318 2E61    LOOP3: MVI     L,ROW8080       ;POINT H-L TO 8080 ROW #
531 031A 7E             MOV     A,M             ;AND LOAD TO ACC
532 031B C3FB02         JMP     BACK1
533
534 031E 3E2F    R048:  MVI     A,02F           ;CHANGE 8080 ROW #
535 0320 77             MOV     M,A             ;TO 23D AND STORE
536 0321 C30103         JMP     LOOP1           ;JUMP TO POINTER EXCHANGE ROU
537
538 0324 3E2F    FR048: MVI     A,02F           ;
539 0326 77             MOV     M,A
540 0327 C30F03         JMP     LOOP2
541
542 032A 3E2F    LR048: MVI     A,02F           ;PUT THE 1ST ROW TO
543 032C 77             MOV     M,A             ;17H.
544 032D C31803         JMP     LOOP3           ;JUMP TO 8080 ROW # STORE
545
```

TL/F/5866–18

Continued Next Page

```
546                     ;CLEAR ROW ROUTINE
547
548 0330 CD3603 CLROW:  CALL    CLROW1
549 0333 C36E02         JMP     CR
550
551 0336 1E61   CLROW1: MVI     E,ROW8080
552 0338 CD8202         CALL    LDHL            ;PUT ROW DATA IN H-L REG
553 033B 3E50   CLROW2: MVI     A,050           ;INTILIZE LOOP COUNTER.
554 033D 3620   LOOP4:  MVI     M,020           ;STORE ASCII SPACE IN MEM.
555 033F 3D             DCR     A               ;DECREMENT LOOP COUNTER.
556 0340 C8             RZ                      ;RETURN IF ZERO BIT IS SET.
557 0341 23             INX     H               ;NEXT LOCATION
558 0342 C33D03         JMP     LOOP4           ;CLEAR NEXT LOCATION.
559
560 0345 D301   BELL:   OUT     001             ;RING BELL
561 0347 C9             RET
562
563 0348 AF     IVERTN: XRA     A
564 0349 1E68           MVI     E,IMASK         ;POINT D,E TO MASK
565 034B 1A             LDAX    D
566 034C 17             RAL                     ;CK BIT 8 STATUS
567 034D DA5203         JC      RESET
568 0350 3E80           MVI     A,080           ;INVERT BIT 8
569 0352 12     RESET:  STAX    D               ;STORE OUT NEW MASK
570 0353 C9             RET
571
572 0354 E5     IVERTR: PUSH    H
573 0355 1E61           MVI     E,ROW8080
574 0357 CD8202         CALL    LDHL            ;LOAD 1ST ADD. OF 8080ROW TO
575 035A 1E50           MVI     E,050           ;SET COUNTER
576 035C 7E     LOOP6:  MOV     A,M             ;GET CHAR.
577 035D 17             RAL                     ;CK BIT 8 STATUS AND INVERT
578 035E DA7003         JC      RESET1
579 0361 1F             RAR
580 0362 F680           ORI     080             ;MASK BIT 8 HIGH
581 0364 77     BACK2:  MOV     M,A             ;STORE MOD. CHAR TO MEM
582 0365 23             INX     H               ;POINT TO NEXT MEM
583 0366 7B             MOV     A,E
584 0367 FE01           CPI     001
585 0369 CA7603         JZ      DONE            ;RETURN IF COUNT = ZERO
586 036C 1D             DCR     E               ;DEC. COUNTER
587 036D C35C03         JMP     LOOP6
588
589 0370 1F     RESET1: RAR
590 0371 E67F           ANI     07F             ;RESET BIT 8
591 0373 C36403         JMP     BACK2
592
593 0376 E1     DONE:   POP     H
594 0377 C9             RET
595      0000            END    START
```

TL/F/5866–19

```
A       0007    ACELD   011C    ADCUR   0161    ADDCH   0298
B       0000    B110    00D4    B1200   00EC    B150    00DA
B1800   00F2    B2000   00F8    B2400   00FE    B300    00E0
B3600   0104    B4800   010A    B600    00E6    B7200   0110
B9600   0116    BACK    0083    BACK1   02FB    BACK2   0364
BAUD    0093    BELL    0345    BS      02E0    C       0001
CHARNU  0063    CLRAM   0042    CLRAM1  004C    CLROW   0330
CLROW1  0336    CLROW2  033B    CLROW3  01C4    CR      026E
CRTCRO  0064    D       0002    DONE    0376    E       0003
FIRSTR  0062    FR048   0324    FUNC    0170    H       0004
HMCUR   0087    HOME    02A4    IMASK   0068    INCRO   01E5
INIT    003B    INTACE  014A    INTKB   0136    IVERTN  0348
IVERTR  0354    L       0005    LASTRO  0060    LDHL    0282
LDHL1   0283    LF      028D    LOOP    0244    LOOP1   0301
LOOP2   030F    LOOP3   0318    LOOP4   033D    LOOP5   01CF
LOOP6   035C    LR048   032A    M       0006    NEWRO   0225
NRS     0061    NXRO    01BE    NXRO1   01DC    PCUR    01B3
PSW     0006    RESET   0352    RESET1  0370    RO48    031E
ROLO    020D    ROW808  0061    ROWSAV  0065    ROZERO  01D7
SAVRO   027B    SCROLL  0205    SP      0006    START   0000
SWAP    02B5    TEMP1   0066 *  TEMP2   0067 *  UPCUR   02F1
UPROW   02EE    UPSCL   0308    VERTI   024F    ZCHAR   01F3
ZCRTC   024A    ZFRO    021E    ZLRO    0219    ZROW    01FB

 NO ERROR LINES
SOURCE CHECKSUM = 403F
OBJECT CHECKSUM = 0F51
INPUT FILE     1:CRT80A.SRC ON JIMFM
OBJECT FILE    1:CRT80A.LM ON JIMFM
```

TL/F/5866–20

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.

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

AN-199

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:

1. 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.
2. 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 Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408