AN-199 A Low Component Count Video Data Terminal Using the DP8350 CRT Controller and the INS8080 CPU
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 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 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 (FFFFH) 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 00H to the cursor and the top of page registers in the DP8350 CRT. 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).
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
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 registers 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 entire routine takes 1 scan line of time, approximately 64 $\mu$s. 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.
### Memory Reference Tables

#### FIGURE 4. New Row Start Look Up Table

<table>
<thead>
<tr>
<th>Row Number</th>
<th>NRS High</th>
<th>NRS Low</th>
<th>Address</th>
<th>Row Data</th>
<th>Address</th>
<th>Row Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 0</td>
<td>3 0</td>
<td>3 F 3 0</td>
<td>0 0</td>
<td>3 F 3 0</td>
<td>0 0</td>
</tr>
<tr>
<td>1</td>
<td>0 1</td>
<td>3 0</td>
<td>3 F 3 1</td>
<td>5 0</td>
<td>3 F 3 1</td>
<td>5 0</td>
</tr>
<tr>
<td>2</td>
<td>0 2</td>
<td>3 0</td>
<td>3 F 3 2</td>
<td>A 0</td>
<td>3 F 3 2</td>
<td>A 0</td>
</tr>
<tr>
<td>3</td>
<td>0 3</td>
<td>3 0</td>
<td>3 F 3 3</td>
<td>F 0</td>
<td>3 F 3 3</td>
<td>F 0</td>
</tr>
<tr>
<td>4</td>
<td>0 4</td>
<td>3 1</td>
<td>3 F 3 4</td>
<td>4 0</td>
<td>3 F 3 4</td>
<td>4 0</td>
</tr>
<tr>
<td>5</td>
<td>0 5</td>
<td>3 1</td>
<td>3 F 3 5</td>
<td>9 0</td>
<td>3 F 3 5</td>
<td>9 0</td>
</tr>
<tr>
<td>6</td>
<td>0 6</td>
<td>3 1</td>
<td>3 F 3 6</td>
<td>E 0</td>
<td>3 F 3 6</td>
<td>E 0</td>
</tr>
<tr>
<td>7</td>
<td>0 7</td>
<td>3 2</td>
<td>3 F 3 7</td>
<td>3 0</td>
<td>3 F 3 7</td>
<td>3 0</td>
</tr>
<tr>
<td>8</td>
<td>0 8</td>
<td>3 2</td>
<td>3 F 3 8</td>
<td>B 0</td>
<td>3 F 3 8</td>
<td>B 0</td>
</tr>
<tr>
<td>9</td>
<td>0 9</td>
<td>3 2</td>
<td>3 F 3 9</td>
<td>D 0</td>
<td>3 F 3 9</td>
<td>D 0</td>
</tr>
<tr>
<td>10</td>
<td>0 A</td>
<td>3 3</td>
<td>3 F 3 A</td>
<td>2 0</td>
<td>3 F 3 A</td>
<td>2 0</td>
</tr>
<tr>
<td>11</td>
<td>0 B</td>
<td>3 3</td>
<td>3 F 3 B</td>
<td>7 0</td>
<td>3 F 3 B</td>
<td>7 0</td>
</tr>
<tr>
<td>12</td>
<td>0 C</td>
<td>3 3</td>
<td>3 F 3 C</td>
<td>C 0</td>
<td>3 F 3 C</td>
<td>C 0</td>
</tr>
<tr>
<td>13</td>
<td>0 D</td>
<td>3 4</td>
<td>3 F 3 D</td>
<td>1 0</td>
<td>3 F 3 D</td>
<td>1 0</td>
</tr>
<tr>
<td>14</td>
<td>0 E</td>
<td>3 4</td>
<td>3 F 3 E</td>
<td>6 0</td>
<td>3 F 3 E</td>
<td>6 0</td>
</tr>
<tr>
<td>15</td>
<td>0 F</td>
<td>3 4</td>
<td>3 F 3 F</td>
<td>B 0</td>
<td>3 F 3 F</td>
<td>B 0</td>
</tr>
<tr>
<td>16</td>
<td>1 0</td>
<td>3 5</td>
<td>3 F 4 0</td>
<td>0 0</td>
<td>3 F 4 0</td>
<td>0 0</td>
</tr>
<tr>
<td>17</td>
<td>1 1</td>
<td>3 5</td>
<td>3 F 4 1</td>
<td>5 0</td>
<td>3 F 4 1</td>
<td>5 0</td>
</tr>
<tr>
<td>18</td>
<td>1 2</td>
<td>3 5</td>
<td>3 F 4 2</td>
<td>A 0</td>
<td>3 F 4 2</td>
<td>A 0</td>
</tr>
<tr>
<td>19</td>
<td>1 3</td>
<td>3 5</td>
<td>3 F 4 3</td>
<td>F 0</td>
<td>3 F 4 3</td>
<td>F 0</td>
</tr>
<tr>
<td>20</td>
<td>1 4</td>
<td>3 6</td>
<td>3 F 4 4</td>
<td>4 0</td>
<td>3 F 4 4</td>
<td>4 0</td>
</tr>
<tr>
<td>21</td>
<td>1 5</td>
<td>3 6</td>
<td>3 F 4 5</td>
<td>9 0</td>
<td>3 F 4 5</td>
<td>9 0</td>
</tr>
<tr>
<td>22</td>
<td>1 6</td>
<td>3 6</td>
<td>3 F 4 6</td>
<td>E 0</td>
<td>3 F 4 6</td>
<td>E 0</td>
</tr>
<tr>
<td>23</td>
<td>1 7</td>
<td>3 7</td>
<td>3 F 4 7</td>
<td>3 0</td>
<td>3 F 4 7</td>
<td>3 0</td>
</tr>
</tbody>
</table>

### FIGURE 6. Input/Output Space

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT</td>
<td>40</td>
</tr>
<tr>
<td>IN</td>
<td>80</td>
</tr>
<tr>
<td>IN</td>
<td>40</td>
</tr>
</tbody>
</table>

### FIGURE 5. Reference Table

<table>
<thead>
<tr>
<th>Device</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>0000 to 0FFF</td>
</tr>
<tr>
<td>RAM</td>
<td>3000 to 3FFF</td>
</tr>
<tr>
<td>CRTC</td>
<td>5000 to 5FFF</td>
</tr>
<tr>
<td>ACE</td>
<td>9000 to 9007</td>
</tr>
</tbody>
</table>

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

### FIGURE 7. CPU Addressing Space

<table>
<thead>
<tr>
<th>Row Number</th>
<th>NRS High</th>
<th>NRS Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0 2</td>
<td>3 F 2 0 A 3 F 5 0 0 0</td>
</tr>
</tbody>
</table>

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

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 (CNTLE), 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, initializing 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).
DP8350/8080 Video Data Terminal Basic Software Flow Chart

Initialization

START

LOAD STACK POINTER

CLEAR RAM

HOME CURSOR TO TOP OF PAGE

LOAD ACE WITH BAUD RATES

INITIALIZE NEW ROW START LOOK UP TABLE

INITIALIZE REFERENCE TABLE

CLEAR INTERRUPT LATCHES

SET POINTERS

ENABLE INTERRUPTS

WAIT LOOP

Keyboard Interrupt

START

READ KEYBOARD

NEED BAUD?

YES

PUT CHARACTER TO ACE

NO

LOAD BAUD

RETURN

TL/F/5866-7

TL/F/5866-6
New Row Start Interrupt

**Start**

1. SAVE 8080 Registers
2. LOAD CRTC ROW = TO CRTC
3. RESET INTERRUPT LATCH
4. **IF MAX ROW IN VIDEO PAGE?**
   - YES: ZERO CRTC ROW =
   - NO: INCREMENT TO NEXT ROW =
5. SAVE THE ROW =
6. RESTORE 8080 Registers
7. ENABLE INTERRUPTS
8. **RETURN**

Vertical Interrupt

**Start**

1. SAVE 8080 Registers
2. MOVE FIRST ROW = TO CRTC ROW =
3. GET CRTC ROW = ADDRESS
4. LOAD TO CRTC TOP OF PAGE REGISTER
5. RESTORE 8080 Registers
6. **RETURN**
DP8350/8080 Video Data Terminal Basic Software Flow Chart (Continued)

ACE Interrupt

START

READ ACE

CNTL OR ESC FUNCTION

YES

FUNCTION?

YES

GO FUNCTION

NO

RETURN

PUT BYTE TO RAM

LAST CHARACTER OF ROW?

YES

LAST ROW ON SCREEN?

YES

RETURN

NO

INCREMENT CHARACTER

PUT CURSOR

RETURN

MAX ROW FOR VIDEO RAM?

YES

INCREMENT BBD ROW

NO

ZERO BBD ROW

IS FIRST ROW = TO MAX ROW = IN VIDEO RAM?

YES

ZERO FIRST ROW

INCREASE LAST ROW

NO

IS FIRST ROW = EQUAL TO LAST ROW?

YES

INCREMENT FIRST ROW

NO

GET BBD ROW = ADDRESS

ZERO CHARACTER

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)
Parts:
1—DP8350
1—DP9228
1—DP8224
1—INS8250
8—RAM
1—2706
1—DM74LS95
1—DM74LS96
2—DM74LS32
2—DM74LS74
2—DM74LS04
2—DM74LS73
2—DM74LS32
2—DM74LS74
2—DM74LS04
2—DM74LS73
2—DM74LS32
1—char. gen/latch
2—Res. arrays,
3.3k
1—21.84 MHz
Xtal
Bypass capacitors
on all parts

Note 1: See DP8350 data sheet for sync details.
Note 2: SW open reverses video page.
TITLE CRTC / 8080A 02/15/78

;** NATIONAL SEMICONDUCTOR
;** SERIES PROGRAMMABLE CRT CONTROLLER BOARD **
;AL BRILLIOTT - JM TROUTNER

160000 FX START D1 //DISABLE INTERRUPTS
160004 33FF20 //INITIALIZE ROUTINE
160008 C26F30 //NEW ROW START INTERRUPT
16000C D8001 //INTERRUPT
160010 001B //KEYBOARD INTERRUPT
160014 C36F00 //VERTICAL INTERRUPT
160018 33FF00 //HORIZONTAL INTERRUPT
16001C 320000 //NEW ROW START LOB UP TABLE GENERATION
16001E 320000 CALL //GO TO CUR HOME ROUTINE
160020 3FF000 CALL //GO TO BRD LOAD ROUTINE
160024 000000 //NEW ROW START LOCK UP TABLE GENERATION
160028 000000 //LAST ROW NUMBER TO ACC
16002C 000000 //STORE TO REFERENCE TABLE
160030 000000 //CLEAR PERIPHERAL INTERRUPT FLOPS
160034 000000 //STORE TO REFERENCE TABLE
160038 000000 //SET UP POINTERS
16003C 000000 //WAIT LOOP FOR INTERRUPTS
160040 000000 //ENABLE INTERRUPTS
160044 000000 //LOOP UNTIL INTERRUPTED
160048 000000 //RETURN

Continued Next Page
; DAUD RATE SELECT
17
MB 9072 DS DAUD: PUSH D ; SAVE D-E CARDS
99 0049 0040 IN 040 ; READ DAUD SELECT CODE
196 0049 006F AND 00F ; ZERO THE HIGH ORDER 4 BITS
101 008F FEOO CFI 000
102 008F CD400 JZ 0110 ; 110 DAUD ROUTINE
103 008F FE00 CFI 001
104 008F CD400 JZ 0150 ; 150 DAUD ROUTINE
105 008E FE00 CFI 002
106 008F CD400 JZ 0300 ; 300 DAUD ROUTINE
107 00AE FE03 CFI 003
108 00AF CB400 JZ 0600 ; 600 DAUD ROUTINE
109 00AF CE04 CFI 004
110 00AF CE00 JZ 01200 ; 1200 DAUD ROUTINE
111 00B1 FE00 CFI 005
112 00B3 CF000 JZ 01800 ; 1800 DAUD ROUTINE
113 00B5 FE00 CFI 006
114 00B6 CB000 JZ 02000 ; 2000 DAUD ROUTINE
115 00B8 FE00 CFI 007
116 00B9 CF000 JZ 02400 ; 2400 DAUD ROUTINE
117 00BC FE00 CFI 008
118 00C2 CE00 JZ 03600 ; 3600 DAUD ROUTINE
119 00C3 FE00 CFI 009
120 00C7 C0A00 JZ 04800 ; 4800 DAUD ROUTINE
121 00CA FE00 CFI 00A
122 00CC CE00 JZ 07200 ; 7200 DAUD ROUTINE
123 00CF FE00 CFI 00B
124 00D1 CA100 JZ 09600 ; 9600 DAUD ROUTINE
125
126
; DAUD RATE SET UP ROUTINES
127
128 00D4 116305 B100 LXI D.005E3 ; 110 DAUD DIVIDER
129 00D4 033C01 JMP ACED ; 030 TO ACE LOAD ROUTINE
130 00D4 113F00 B150 LXI D.003F3 ; 150 DAUD ROUTINE
131 00D4 031C01 JMP ACED
132 00D4 110F01 B200 LXI D.001F9 ; 200 DAUD DIVIDER
133 00D4 033C01 JMP ACED
134 00D4 110F00 B600 LXI D.000FC ; 600 DAUD DIVIDER
135 00D4 031C01 JMP ACED
136 00D4 112E00 B1200 LXI D.0007E ; 1200 DAUD DIVIDER
137 00D4 031C01 JMP ACED
138 00D4 115400 B1800 LXI D.00054 ; 1800 DAUD DIVIDER
139 00D4 031C01 JMP ACED
140 00D4 116400 B2000 LXI D.0004C ; 2000 DAUD DIVIDER
141 00D4 033C01 JMP ACED
142 00D4 113F00 B2400 LXI D.0003F ; 2400 DAUD DIVIDER
143 0100 031C01 JMP ACED
144 0100 113900 B3600 LXI D.0002A ; 3600 DAUD DIVIDER
145 0107 031C01 JMP ACED
146 0100 112000 B4800 LXI D.00020 ; 4800 DAUD DIVIDER
147 0100 031C01 JMP ACED
148 0100 111500 B7200 LXI D.00015 ; 7200 DAUD DIVIDER
149 0100 031C01 JMP ACED
150 0100 111000 B9600 LXI D.00010 ; 9600 DAUD DIVIDER
151 0100 031C01 JMP ACED
152
153 ; ACE LOAD ROUTINE
154
155 0101 010390 ACHEL LXI D.09002 ; POINT B C TO ACE
156 0101 0303 MVU A.083 ; INITIATE DAUD LOAD - 8 BITS
157 0102 0211 SXR B ; DC INIT DAUD LOAD
158 0102 02E0 MVI C.001 ; POINT TO DAUD HIGH
159 0102 07A4 MOV A B ; GET DAUD HIGH
160 0102 0202 SXR B ; STORE DAUD HIGH TO ACE
161 0102 0600 MVI C.000 ; POINT ACE TO DAUD LOW
162 0102 0798 MOV A B ; GET DAUD LOW
163 0102 0292 SXR B ; STORE DAUD LOW TO ACE
164 0102 0203 MVI C.003 ; RESET CLR TO ZERO
165 010C 79 MOV A C ; INIT ACE T/R
166 010C 0211 SXR B ; PUT TO ACE
167 010C 02E0 MVI C.001 ; INTERRUPT ENABLE REG
168 010C 0309 MOV A C ; SELECT RECEIVED DATA INTERRUPT
169 010C 0211 SXR B ; LOAD IT
170 010C 02E0 MVI C.000 ; RESTORE D-C ACE POINTER
171 0134 04B4 POP D ; RESTORE C-E REGISTERS
172 0135 099 RET ; RETURN
173
174 ; KEYBOARD INTERRUPT ROUTINE
175
176 0136 0880 INTKB IN 080 ; READ KEYBOARD
177 0136 08FF EI ; ENABLE INTERRUPTS
178 0136 0F03 CFPI 005 ; NEED BAUD RATE? (CNTL E)
179 0136 0800 CFI 006 ; IF YES GO TO DAUD ROUTINE
180 0136 0E2F CFPI 012 ; INVERT NEXT CNTL R
181 0136 04A000 CFI 015 ; INVERT CNTL S
182 0136 04B000 CFI 015 ; INVERT ROW CNTL S
183 0136 04C000 JZ 0FFTA
184 0136 082F SXR B ; STORE BYTE TO ACE
185 0136 099 RET ; RETURN
186

Continued Next Page
187  ACE INTERRUPT ROUTINE
188
189  014A  DA  INTACE  LDAX  D  ;LOAD ACE DATA BYTE TO ACC
190  014B  FB  E1  ;ENABLE INTERRUPTS
191  014C  FE  CF  07E  ;TEST FOR ESC COMMAND
192  014D  90  C0  07F  ;TEST FOR DEL COMMAND
193  014E  90  C0  07F  ;TEST FOR SPACE COMMAND
194  014F  90  C0  07F  ;RESTORE H.L
195  0150  9F  MOV  E,A  ;SAVE CHAR IN REG E
196  0151  7F  ANI  06H  ;MASK OUT BITS FOR CNTL TEST
197  0152  9D  C700  JZ  FUNC  ;IF ZERO JMP TO CNTL FUNC
198  0153  9D  C66F  LD  03F6  ;LOAD INVERT MASK
199  0154  53  80  E  ORL  E  ;OR MASK AND CHAR
200  0155  60  77  M,A  ;STORE DATA BYTE TO RAM
201
202  209  ;ADVANCE CURSOR
203
204  0161  IE0  ADCUR:  MOV  E,CHARDUM  ;POINT E-C TO CHAR #
205  0162  1A  LDAX  D  ;LOAD CHAR # TO ACC
206  0163  23  INX  M  ;NEXT CHAR LOCATION
207  0164  0F  PEE  04F  ;LAST CHAR OF ROW
208  0165  0E  C0  050  JZ  ADDR  ;IF TRUE JMP TO NEXT ROW
209  0166  0A  56  AD  001  INCR CHAR #
210  0167  12  STAX  D  ;STORE CHAR # TO RAM REF
211  0168  30  CJ2001  JMP  PCUR  ;PUT CURSOR TO CURSOR
212
213  214  ;TEST FOR FUNCTION
215
216  0170  7B  90  JNZ  FE01  ;FUNC MOV A,E  ;HOME AND CLEAR CNTL A (SOH)
217  0171  7C  C0000  JZ  START  ;
218  0172  7E  FE00  CPI  000  ;CARRIAGE RETURN
219  0173  7F  C0E2  JZ  CR  ;
220  0174  7F  FE11  CPI  011  ;SAVE ROW # CNTL O (DC1)
221  0175  80  C0D2  JZ  SAVE  ;
222  0176  80  FE0C  CPI  00C  ;ADVANCE CURSOR CNTL L (FF)
223  0177  82  C101  JZ  ADCUR  ;
224  0178  85  FE02  CPI  002  ;HOME UP CNTL B (1STX)
225  0179  87  C440  JZ  HOME  ;
226  017A  86  FE1A  CPI  01A  ;SWAP CNTL Z (SUB)
227  017B  8C  C500  JZ  SMAP  ;
228  017C  8F  FE0A  CPI  00A  ;LINEFEED
229  017D  91  D80E  JZ  LF  ;
230  017E  94  FE08  CPI  008  ;BACKSPACE CNTL H (BS)
231  017F  96  C0E1  JZ  BS  ;
232  0180  99  FE0B  CPI  00B  ;UP CURSOR CNTL K (VT1)
233  0181  9B  C502  JZ  UPCR  ;
234  0182  9E  FE18  CPI  018  ;CLEAR CNTL X (CAN)
235  0183  9E  C303  JZ  CLGW  ;
236  0184  9E  FE07  CPI  007  ;RING DEL CNTL Q (BEL)
237  0185  9E  C453  JZ  BELL  ;
238  0186  9F  FE12  CPI  012  ;INSERT NEXT CNTL R (DC2)
239  0187  9F  C480  JZ  IVERTR  ;
240  0188  AD  E013  CPI  013  ;INSERT ROW CNTL S (DC3)
241  0189  AF  C540  JZ  IVERTR  ;
242  018A  29  RET  ;RETURN
243
244  ;STORE CURSOR TO CRTC FROM H-L REGISTERS
245
246  018B  7C  PCUR:  MOV  A,H  ;H REG TO ACC
247  018C  63  C020  ADI  020  ;SET H-L REG TO CRTC ADD
248  018D  67  MOV  H,A  ;H IS CRTC ADD
249  018E  36  30  CPI  003 ;CURSOR REGISTER SELECT
250  018F  7C  MOV  A,H  ;A REG SET BACK TO VIDEO RAM
251  0190  62  20  SUI  020  ;ADDRESS
252  0191  67  MOV  H,A  ;
253  0192  00  RET  ;RETURN
254
255  256  ;LAST ROW ON SCREEN
257
258  0199  0B  CDD01  NYRO  ;CALL NYRO1  ;GO TO NEXT ROW SUBROUTINE
259  019C  0B  CDD01  NYRO  ;CALL NYRO1  ;ZERO CHARACTER
260  019F  0B  CDD01  NYRO  ;SAVE H-L
261  01C5  EE0  JZ  LASTROW  ;POINT D-E TO LASTROW
262  01C7  1A  LDAX  D  ;LASTROW
263  01C8  C501  ADI  001  ;POINT AC TO FIRST ROW OFF SC
264  01C9  FE00  CPI  030  ;CF IF LAST ROW IN RAM
265  01CC  C470  JZ  ROZERO  ;
266  01CF  C6302  LOOPS:  CALL  LEMI  ;LOAD H-L WITH ADD. OF LASTROW
267  01D2  C602  CALL  LEMI  ;
268  01DC  E1  ROP  H  ;RESTORE H-L
269  01DF  69  RET  ;
270  01E5  GE0  ROZERO,  MOV  A,000  ;LOAD ROW ZERO
271  01ED  CDD01  JMP  LOOPS  ;LOOPS
272
273
Continued Next Page
Continued Next Page
Continued Next Page
Continued Next Page
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 hexadecimal 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
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.
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI’s standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions: