6845 Video Display Board
By Data Systems 68

The 6845 video display board (VDS) provides your SS-50 based system with memory mapped video display. If you add a keyboard for data entry, You will have your terminal built into your computer and won't need a separate one. To simplify the installation into your system, software drivers are provided that are overlays to SBUG-E(tm). These drivers reside totally within the 2k SBUG-E(tm) space. To complete the installation, a simple patch to FLEX(tm) will vector to your new OUTCH routine.

The VDB is based on the Motorola MC6845 CRT controller chip, U26. A 2716 font EPROM, 2k of display RAM and a handful of TTL latches buffers gates and shift registers support the 6845 for use on the SS-50 bus. The display RAM resides entirely within the 65k address space so that a program can write characters directly to a displayed page.

Description:

The address bus A1-A10, is multiplexed between the 6845 and the 2k display RAM (U15-18) by TTL switches U11-12, and 19. The 6845 puts out data which is latched by a pair of 74175 quad latches (U8-9). This data is input as address information to the font EPROM (U14). The lower 4 address bits for U14 are row addresses from the 6845 and the data that appears as output from the EPROM is the character row information for a given scan line. This parallel data from the EPROM is lorded into a 74166 shift register (U13) where it is clocked out on a scan line. This output is the video which when combined with sync and cursor information from the 6845 becomes the composite video output to the CRT monitor.

Scanned and edited by Michael Holley Nov 28, 2000
Data Systems 68 Document Circa 1982

A 4-position dip switch places the display RAM and the controller registers at a 4k boundary. For the moment, choose this boundary at $E000. This places the 6845 controller registers at $E7FE-E7FF and the display ram from $E800-$EFFF. These addresses are decoded basically by 74133, an 8-input NAND gate. Page decoding for the display ram is provided by the 74LS00 (U10) a demultiplexer. The $E000 addressing boundary is decoded by a 7485 comparitor (U5) from the 4 contact dip switch and address lines A12-15.

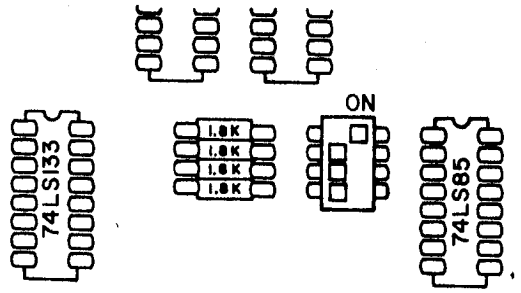
Data from the SS-50 system bus can also write and read the display RAM through a pair of 8835 bus transceivers (U6-7). This allows direct use of the screen by such programs as the SCREDITOR(tm).

A local 12.56 MHz crystal oscillator (U20) is counted down by a 74163 binary counter (U21). This generates the scan rate timing for the 74166 shift register (U13) which clocks a row of character onto a given scan line. This same divided down clock is input to the 6845 where sync is generated. Horizontal and vertical sync and cursor position data are all ORed to generate the composite video. The actual sync timing cursor characteristics and character placement are determined from the initialization lookup table in the software drivers provided.

Getting it running:

Once the board is built, check it carefully for solder shorts and proper chip placement. Pay particular attention to pin 1 on each IC. Leave out the 6845 chip for the initial checkout. Plug the board into the bus for power and check the output of the 5 volt regulator, U27. It should be 5 volts plus or minus 5%. If all is well, remove the board from the system with system power off and install the 6845 IC.

Now set the dip switch. The most significant bit of the address is the lower switch (closest to the edge connector). The upper 4 bits of the address determine the boundary that we are setting.



Notice that an on switch is a binary zero.

Address	#1 – A12	#2 A13	#3 A14	#4 A14
0000	On	On	On	On
1000	Off	On	On	On
2000	On	Off	On	On
3000	Off	Off	On	On
4000	On	On	Off	On
5000	Off	On	Off	On
6000	On	Off	Off	On
7000	Off	Off	Off	On
8000	On	On	On	Off
9000	Off	On	On	Off
A000	On	Off	On	Off
B000	Off	Off	On	Off
C000	On	On	Off	Off
D000	Off	On	Off	Off
E000	**On**	**Off**	**Off**	**Off**
F000	Off	Off	Off	Off

That's all there is to building up the board. All that remains is to include the software drivers into SBUG-E(tm) and make the small patch in FLEX(tm) to vector to the new OUTCH routine called decode. The enclosed assembly listings show the OUTCH replacement and the simple patch to FLEX(tm). The patch to SBUG-E (tm) deletes the DATRAM routines and the cassette load and punch routines to make room. Just burn a new EPROM with the modified SBUG-E(tm) and you're off and running.

NOTE

U14 is a 2716 EPROM which is the character generator you may want to program your own custom prom or a 2716 with character set are available for Data Systems "68" or you can use the listing supplied and burn your own.

The patch to FLEX(tm) is a simple JSR to your decode in SBUG-E(tm). Use the FLEX(tm) utilities FIX or DISKEDIT to make the 7 byte change.

It should be noted that the listing supplied are to configure for a 80 by 24 display.

That's it, now go forth and enjoy the flexibility of your new video display board from data systems 68.

FLEX is a trademark of Technical Systems Consultants and SBUG-E is trademark of Southwest Technical Products Corp. SCREDITOR is a trademark of Alford & Associates

```
                    * This patch to the FLEX(tm) output routine links
                    * The new video display drivers to FLEX(tm).

          FF35      DECODE   EQU     $FF35 NEW OUTEE S/R

   D38B                      ORG     $D38B FLEX (TM) OUTPUT S/R

   D38B   34 17     OUTPUT   PSH     X, A, B, CC
   D38D   BD FF35            JSR     DECODE
   D390   35 97              PUL     X, R, B, CC, PC (RTS)
                             END
```

Output Connector

+5V	1	14	+5V
Composite In	2	13	Vert. Sync. Out
Composite In	3	12	Horz. Sync. Out
Composite In	4	11	Video out
120 Ohm	5	10	120 Ohm
	6	9	Composite Video
Ground	7	8	Ground

For composite video jumper 2 to 13, 3 to 12 and 4 to 11.

QUANTITY	PART
	RESISTORS
1	330 Ohm 1/4w
1	220 Ohm 1/4w
1	120 Ohm 1/4w
1	180 Ohm 1/4w
1	47 Ohm 1/4w
4	1.8 K Ohm 1/4w
1	10 K Ohm 1/4w
	CAPACITORS
7	0.1 uF Ceramic
1	39 PF Silver Mica
1	47 uF Electrolytic
1	18 uF Electrolytic
1	0.47 uF Ceramic
	Miscellaneous
1	12.576 MHz Crystal
1	2N3904 Transistor
1	1N914 Diode
1	40-Pin Socket
1	24-Pin Socket
4	18-Pin Sockets
12	16-Pin Sockets
9	14-Pin Sockets
5	10-Pin Female Molex Connectors
1	THM 6060 Heat Sink
2	6-32 Screws
2	6-32 Nuts
	INTEGRATED CIRCIUTS
1	LM309K (U27)
1	74166 (U13)
2	7404 (U20) (U1)
1	74163 (U21)
1	2716 (U14)
3	74LS18 (U2) (U3) (U22)
3	74LS175 (U8, U9, U25)
4	2114 (U15-18)
1	7401 (U23)
1	74LS88 (U10)
1	74LS133 (U4)
3	74157 (U11, 12, 19)
1	7486 (U24)
1	7485 (U5)
2	8835 (U6, 7)
1	6845 (26)

```
  NAM 6845 TERMINAL DRIVER

* MODIFIED TO BE CORESIDENT WITH SBUG E (TM)
* DAT RAM, KC LOAD AND KC PUNCH ARE DELETED
* FROM SBUG   4/18/82

        OPT PAG
        ORG    $FF00

* POWER ON ENTRY
RESET   LDS    #STKADD
        JSR    CLEAR
CRTIZ   LDX    #CRTTAB
        CLRA
CRT1    STAA   CRTTAB    SET CRT CONTROLLER ADDRESS
        LDAB   0,X
        STAB   CRTCDR
        INX              READ CRT INITIALZATION
        INC    A         TABLE INTO CONTROLLER
        CMPA   #$10
        BNE    CRT1
        LDX    #0
        STX    CURPOS
        STX    STARAD
        STX    LINEAD
        STX    CCOUNT
        LDS    #$DFC0
        LDAB   #$FF      TURN ON ECHO, NON ZERO
        STAB   $DFE2
        JMP    ESCSET    SETUP RETURN FOR ESCAPE

*
* CONTROL CHARACTER DECODE
DECODE  BITA   #$60
        BEQ    DECOD1
* PUT CHARACTER IN DISPLAY
        LDAB   CURPOS
        ANDB   #$07
        ORAB   #$E8
        STAB   CUTEMP
        LDAB   CURPOS+1
        STAB   CUTEMP+1
        LDX    CUTEMP
        STAA   0,X
        LDX    CURPOS
        INX
        STX    CURPOS
        INC    CCOUNT
        LDAA   CCOUNT
        CMPA   CLINE
        BPL    CRLF
        BRA    UPDATE

DECOD1  LDX    #DECTAB-1
        CLRB
NEXT1   INCB
        INX
        CPX    #DECEND
        BEQ    DECOD
        CMPA   0,X
```

```
        BNE    NEXT1
        LDX    #DECADD-2
NEXT2   INX
        INX
        DECB
        BNE    NEXT2
        LDX    0,X
        JSR    0,X
DECOD   RTS

* UPDATE CRT CONTROLLER REGISTERS
UPDATE  PSHB
        LDAB   #$0C
        LDX    #STARAD
UDATE1  LDAA   0,X
        STAB   CRTCAR
        STAA   CRTCDR
        INX
        INCB
        CMPB   #$10
        BNE    UDATE11
        PULB
        RTS

*CARRAGE RETURN
RTURN1  LDX    LINEAD
        STX    CURPOS
        CLR    CCOUNT
        BRA    UPDATE

*CARRAGE RETURN LINE FEED
CRLF    BSR    RTURN1
        JMP    LFEED

        ORG    $FC09
*LINE FEED
LFEED   LDX    #STARAD
        LDAA   3,X
        ADDA   CLINE
        STAA   3,X
        BCC    LFEED1
        INC    2,X

LFEED1  LDAA   7,X
        ADDA   CLINE
        STAA   7,X
        STAA   CUTEMP+1
        LDAA   6,X
        ADCA   #$00
        STAA   6,X

* CLEAR NEXT LINE
        ANDA   #$07
        ORAA   #$E8
        STAA   CUTEMP
LFEED3  LDX    CUTEMP
        LDAA   #$20
        LDAB   CLINE
LFEED4  STAA   0,X
        INX
```

```
        CPX    #MEMEND+1
        BNE    LFEED5
        LDX    #MEMSTR
LFEED5  DECB
        BNE    LFEED4

* CHECK FOR LAST LINE ON SCREEN
        LDX    #STARAD
        LDAA   5,X
        CMPA   LSCREN
        BNE    LFEED2
        INC    5,X
        BRA    SCROLU
LFEED2  INC    5,X
        JMP    UPDATE

*SCROLL UP
SCROLU  LDX    #STARAD
        DEC    5,X
        LDAA   1,X
        ADDA   CLINE
        STAA   1,X
        BCC    SCROL1
        INC    0,X
SCROL1  JMP    UPDATE

* BACK SPACE
BACKSP  CLRB
        DEC    CCOUNT
        BPL    BACK1
        LDAA   CLINE
        STAA   CCOUNT
        DECA
        ADDA   CURPOS+1
        ADCB   CURPOS
        STAB   CURPOS
        STAA   CURPOS+1
        JMP    UPDATE
BACK1   LDX    CURPOS
        DEX
        STX    CURPOS
        JMP    UPDATE

* CLEAR SCREEN
CLEAR   LDX    #MEMSTR   STARTING DISPLAY MEM.
        LDAA   #$20      SPACE
CLEAR1  STAA   0,X
        INX
        CPX    #MEMEND+1
        BNE    CLEAR1
        RTS

* HOME CURSOR TOP LEFT
HOMEUP  LDX    STARAD
        STX    CURPOS
        STX    LINEAD
        LDX    #0
        STX    CCOUNT
        JMP    UPDATE
```

```
* CTR PARAMETERS
CLINE   FCB    80        COULMNS PER LINE
LSCREN  FCB    23        LINES PER PAGE ON CRT

DECTAB  FCB    $0A
        FCB    $0D
        FCB    $08
        FCB    $0B
        FCB    $0C
        FCB    $1B
DECEND  EQU    *


        ORG    $FEDF

*CRT INITIALIZATION TABLE
CRTTAB  FCB    99,80,88,8
        FCB    25,2,24,24,00

        FCB    $9,$60,$8
        FDB    $0100,$0100

*COMMAND TABLE
DECADD  FDB    LFEED
        FDB    RTURN1
        FDB    BACKSP
        FDB    CLEAR
        FDB    HOMEUP
        FDB    CURSOR

MEMSTR  EQU    $E800
MEMEND  EQU    $EFFF
CRTCAR  EQU    $E7FE
CRTCDR  EQU    $E7FF
STKADD  EQU    $F7FF
CURSOR  EQU    $F610

*RAM LOCATIONS
        ORG    $F600
STARAD  RMB    2
CURPOS  RMB    2
CCOUNT  RMB    1
LCOUNT  RMB    1
LINEAD  RMB    2
CUTEMP  RMB    2

        ORG    $FDE7

ESCSET  LDAB   #$39      SETUP DUMMY RTS FOR ESC
        STAB   CURSOR

* REPLACE OUTCH S/R IN SBUGE
        ORG    $FFDF

OUTCH   PSH    X,X,B,CC
        JSR    DECODE
        PUL    X,A,B,CC,PC (RTS)

        END
```

```
* Data Systems 68 Character Font For 2716

FCB 18,18,18,18,18,18,18,18
FCB 18,18,00,00,00,00,00,00
FCB 00,00,00,00,FF,FF,00,00
FCB 00,00,00,00,00,00,00,00
FCB 18,18,18,18,FF,FF,18,18
FCB 18,18,00,00,00,00,00,00
FCB 00,00,00,00,F8,F8,18,18
FCB 18,18,00,00,00,00,00,00
FCB 18,18,18,18,F8,F8,00,00
FCB 00,00,00,00,00,00,00,00
FCB 18,18,18,18,1F,1F,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,00,1F,1F,18,18
FCB 18,18,00,00,00,00,00,00
FCB 00,08,08,3E,08,08,00,3E
FCB 00,00,00,00,00,00,00,00
FCB 00,00,08,04,3E,04,08,00
FCB 00,00,00,00,00,00,00,00
FCB AA,55,AA,55,AA,55,AA,55
FCB AA,55,00,00,00,00,00,00
FCB 00,00,08,00,3E,00,08,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,08,08,2A,1C,08,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,00,00,0F,0F,0F
FCB 0F,0F,00,00,00,00,00,00
FCB 00,00,00,00,00,F0,F0,F0
FCB F0,F0,00,00,00,00,00,00
FCB F0,F0,F0,F0,F0,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 0F,0F,0F,0F,0F,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB FF,FF,FF,FF,FF,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 0F,0F,0F,0F,0F,0F,0F,0F
FCB 0F,0F,00,00,00,00,00,00
FCB FF,FE,FC,F8,F0,F0,E0,C0
FCB 80,80,00,00,00,00,00,00
FCB 00,00,00,00,FF,FF,18,18
FCB 18,18,00,00,00,00,00,00
FCB 18,18,18,18,F8,F8,18,18
FCB 18,18,00,00,00,00,00,00
FCB 18,18,18,18,FF,FF,00,00
FCB 00,00,00,00,00,00,00,00
FCB 18,18,18,18,1F,1F,18,18
FCB 18,18,00,00,00,00,00,00
FCB 81,C3,66,3C,18,18,3C,66
FCB C3,81,00,00,00,00,00,00
FCB 01,03,06,0C,18,18,30,60
FCB C0,80,00,00,00,00,00,00
FCB 80,C0,60,30,18,18,0C,06
FCB 03,01,00,00,00,00,00,00
FCB FF,FF,00,00,00,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB FF,FF,00,00,00,00,00,00
FCB C0,C0,C0,C0,C0,C0,C0,C0
FCB C0,C0,00,00,00,00,00,00
```

```
FCB 03,03,03,03,03,03,03,03
FCB 03,03,00,00,00,00,00,00
FCB 00,1E,3C,3C,1C,0C,0C,0C
FCB 00,00,00,00,00,00,00,00
FCB FF,7F,3F,1F,0F,0F,07,03
FCB 01,01,00,00,00,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,08,08,08,08,08,00,08
FCB 00,00,00,00,00,00,00,00
FCB 00,14,14,14,00,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,14,14,3E,14,3E,14,14
FCB 00,00,00,00,00,00,00,00
FCB 00,08,1E,28,1C,0A,3C,08
FCB 00,00,00,00,00,00,00,00
FCB 00,30,32,04,08,10,26,06
FCB 00,00,00,00,00,00,00,00
FCB 00,08,14,14,18,2A,24,1A
FCB 00,00,00,00,00,00,00,00
FCB 00,08,08,10,00,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,04,08,10,10,10,08,04
FCB 00,00,00,00,00,00,00,00
FCB 00,10,08,04,04,04,08,10
FCB 00,00,00,00,00,00,00,00
FCB 00,00,08,2A,1C,2A,08,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,08,08,3E,08,08,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,00,00,00,18,18
FCB 08,10,00,00,00,00,00,00
FCB 00,00,00,00,3E,00,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,00,00,00,18,18
FCB 00,00,00,00,00,00,00,00
FCB 00,01,02,04,08,10,20,40
FCB 00,00,00,00,00,00,00,00
FCB 00,1C,22,26,2A,32,22,1C
FCB 00,00,00,00,00,00,00,00
FCB 00,08,18,08,08,08,08,1C
FCB 00,00,00,00,00,00,00,00
FCB 00,1C,22,02,04,08,10,3E
FCB 00,00,00,00,00,00,00,00
FCB 00,3E,04,08,04,02,22,1C
FCB 00,00,00,00,00,00,00,00
FCB 00,04,0C,14,24,3E,04,04
FCB 00,00,00,00,00,00,00,00
FCB 00,3E,20,3C,02,02,22,1C
FCB 00,00,00,00,00,00,00,00
FCB 00,0C,10,20,3C,22,22,1C
FCB 00,00,00,00,00,00,00,00
FCB 00,3E,02,04,08,10,10,10
FCB 00,00,00,00,00,00,00,00
FCB 00,1C,22,22,1C,22,22,1C
FCB 00,00,00,00,00,00,00,00
FCB 00,1C,22,22,1E,02,04,18
FCB 00,00,00,00,00,00,00,00
FCB 00,00,18,18,00,18,18,00
FCB 00,00,00,00,00,00,00,00
```

```
FCB 00,00,18,18,00,18,18,08
FCB 10,00,00,00,00,00,00,00
FCB 00,02,04,08,10,08,04,02
FCB 00,00,00,00,00,00,00,00
FCB 00,00,00,3E,00,3E,00,00
FCB 00,00,00,00,00,00,00,00
FCB 00,20,10,08,04,08,10,20
FCB 00,00,00,00,00,00,00,00
FCB 00,1C,22,02,04,08,00,08
FCB 00,00,00,00,00,00,00,00
END
```