PRELIM PROGRAMMING SPECS VDM-2/GRAPHIC DISPLAY

BY LEE FELSENSTEIN
Golemics, Inc.
1407 Addison St.
Berkeley, CA 94702

Dear Dr. Dobb's:

I am trying to find a home for an orphan design. The VDM-2 currently exists as a prototype S-100 printed circuit built partly to specifications set by Processor Technology, which so recently made a meteoric attempt to achieve greatness by liquidating. Being but a simple engineer, I stolidly completed the design (with a few small changes) and displayed the result at SigGraph recently. Now it remains to find a manufacturer, especially one who will agree to make a version available as an S-100 Sol retrofit.

From my own experience, the best designs and systems are the result of playing around, rather than "serious work." I have almost completed my own hardware-related playing around – now it's time for the software fun. Accordingly, I am submitting for publication the complete set of software specifications for the current version of the VDM-2.

Note that the final version remains to be laid out. I am committed to making an S-100 version available somehow, but the design can be adapted for almost any bus.

Readers: Please *write me your comments!*

The VDM-2 would have existed six months ago it I had had the good sense to stop waiting for the front office turkeys to make up their minds. As it was, I finally told them what I would and would not design. Now they've lost their front office, but maybe we don't need them anyway. – *Lee Felsenstein*

Introduction

The VDM-2 was designed as a sophisticated 80 X 24 text display for use in Sol and other S-100 computers. Of prime importance in setting the specifications has been the desire to allow for the fullest use of the graphics-related capabilities inherent in the memory-mapped type of display. In addition, provision has been made in the hardware for use of the VDM-2 in an extended video environment.

Graphics-related capabilities include the alternate writeable character font, row-column screen addressing, contiguous accessible video in both dimensions across the screen and "glitch-free" CPU access.

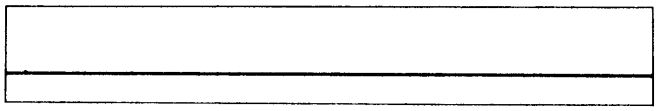
Alphanumeric features include extended attributes such as underline, half intensity, blinking and font select, ROM resident character set using EPROM-compatible character generator, selectable character or cursor blink, selectable text line modulus and smooth scrolling in either direction.

Video-related features include video lock capability among several units, memory disable for efficient multi-unit operation, and gen-lock capability to external video.

The VDM-2 is intended to be used in an interactive system mode and provides for the possibility of quote windowing, mixture with live and animated video, and joystick screen control. Programming for the VDM-2 should be done with the enjoyment of the user in mind.

Screen Format

The VDM-2 occupies an address space of 4096 bytes, organized as 32 rows of 128 columns each. Since the screen displays only 1920 characters at most, columns 81 through 128 (50h through 7Fh) are not displayed and must not be written by the CPU. Similarly, rows 25 through 32 (18h through 1Fh) must not be written.



This is because the board contains memory for only 2048 bytes in its base page, and the mapping necessary to compact the memory space results in non-monotonic relationships between address and physical memory location. In simpler language, anything written into forbidden memory areas will turn up at another apparent location within the visible area.

There is, however, an area of memory which may be utilized by the CPU without any interference with the display. This area exists as a series of eight groups of sixteen contiguous addresses each, starting at C30h and recurring at intervals of 128 bytes (the next group starts at CBOh and runs through CBFh—the lowest three hex digits of the memory address are specified because the highest digit is the base address of the unit). These locations may be used for temporary storage of parameters specific to the operation of the display. Together with the memory disable feature, this allows the use of re-entrant code for the display driver in many applications involving multiple displays.

In such an arrangement, all of the VDM-2's may be addressed to the same 4K base address. Bit 0 of the control output register will prevent the memory decoder of the device from responding when set = 1. I/O references are unaffected by this "DISABLE" bit.

All memory and I/O references to the VDM-2 cause no interruption in the screen display.

Screen Memory Definition

The screen memory stores one 12-bit word for each character location on the screen. These words are accessed by the CPU as determined by the status of the PAGE bit (bit 2) of the control register. A bit value of 0 gives the CPU access to the low-order eight bits of each word. In this case, the low-order seven bits of the CPU data word access the character number—equivalent to the ASCII value if the character of an ASCII character font is used. The high order bit of the CPU word controls the cursor for that character—video will be inverted in that character cell if this bit is set = 1.

With PAGE set = 1 in the control register, the CPU accesses the high-order four bits of the character word. These bits are accessed as the four least significant bits of the CPU data word and their functions are:

bit 0 – underline: the cursor is inverted on scans nine and ten of the character
bit 1 – half intensity:
bit 2 – blink: blinks video at 512 msec period if the BLINK bit of the control register is set = 0; blinks the cursor if a cursor is set at that character location and the BLINK bit of the control register is set = 1
bit 3 – writeable font.

These bits are active when set = 1 in the CPU data byte, and affect only the character location where written. Both pages may be written and read by the CPU, and the PAGE bit of the control register remains at its previous value until changed by a CPU reference or reset to 0 by a bus initialization.

Writeable Font

In addition to the standard 128 ASCII character dot patterns stored in the ROM font memeory, the VDM-2 stores another 128 character dot patterns in a 2048 byte random-access memory. These may be called up on a character-by-character basis by the high order bit of the character data word as described above.

While the raster of the VDM-2 is interlaced, the same video is fed to both fields. This results in a duplication of the video on each pair of interlaced scans and a finer apparent display in the vertical dimension. The video uses 240 scans to display either 24 or 20 character lines.

In the 20 line mode, the character font consists of a dot matrix eight dots wide by twelve scans high. In the 24 line mode the lower two scans of the matrix are truncated to produce a matrix ten scans high. The display mode is selected by the status of bit 4 (24) of the control register. 20 line mode will be selected when this bit is set = 0 or after initialization.

The writeable font memory is accessed by the CPU when the FONT bit of the control register (bit 1) is set = 1. In this mode, the low order seven bits of the address specify the character number to be accessed. The eighth bit of address is ignored, and the ninth through twelfth bits of the address specifies the scan number accessed. The scans are numbered top to bottom, zero through eleven (Oh through Ah). The CPU data byte will be accessed to or from the scan specified. The least significant bit of the data byte will correspond to the right hand dot of the matrix, and the most significant bit to the left hand dot. A bit set = 1 will turn on video in that dot location.

Some persons familiar with graphics programming may quarrel with this assignment of bits horizontally, pointing out that screen co-ordination has historically had its origin point at the upper left hand corner. This scheme has been chosen to make life easier for the inexperienced programmer trying to convert a visual pattern into memory contents. With least significant bit to the right, the pattern becomes a graphic representation of two hexadecimal digits commonly used to represent memory contents. The user need only memorize the hex bit patterns and enter the desired two hex bytes to the memory.

The deletion of the eighth memory address bit puts the scan number fully in the high byte of the memory address, which is manipulated in the H register of most 8 bit microprocessors. A font created and stored in the writeable font memory may be transferred to a 2716 type EPROM through a bus-resident programmer, providing the address is "closed up" by shifting the four scan bits right one place.

Sixteen scans per character pattern may be stored and read by the CPU, but only the top twelve will be visible.

Scrolling

"Scrolling" here refers to the mapping of the text line in memory to the character line on the screen. Mapping of a given character line to a given scan line is considered below under "crawling."

Upon power-up initialization, the memory text line having line address 0 appears as the first character line at the top of the screen. An output to the screen port with data bit 4

set = 0 will increment the memory text line number displayed on the top character line of the screen. The memory text line formerly at the top of the screen will appear as the bottom line on the screen. This occurs regardless of the display mode (20 or 24 line screen).

It is important to note that in the 20 line mode, the highest four memory text lines (20 through 23) do not participate in the scrolling, although they are accessible by the CPU. These lines are used by the "subtext" command, which is described below.

Scrolling down is accomplished by scrolling up N-1 times, where N is the number of character lines displayed on the screen. Since the scrolling hardware is read internally at the leading edge of the vertical display signal VDISP, scrolling should be held off until immediately after this event. VDISP is made available to the CPU on bit 1 of the status port. The CPU should store the last value of this bit and test the new value to detect a change from 0 to 1.

The CPU may reset the scrolling hardware by outputting to the screen port with bit 5 set = 0. This will override any data on bit 4 and will cause the top line of the display to be memory text line 0.

Crawling

The VDM-2 includes the hardware necessary for the movement of the display vertically in increments of one scan. When synchronized with the scrolling, this feature allows for smooth text movement up or down the screen as if the screen were a window gliding over an unbroken column of text.

In such crawling it is ideal if the top displayed line is gradually eclipsed by the top of the screen while the new bottom line emerges gradually at the same time. Since in the 24 line mode of the VDM-2 the partially obscured top and bottom lines are in fact the same text line in memory, this ideal condition is not realized. Instead, a one-line "preblank" curtain descends over the top line at such strategic times.

Crawling is controlled by bits 0 through 3 of the screen port. Bit 0 sets the direction (0 = up, 1 = down), and bits 1 and 2 select one of four crawl rates. Three of these are generated in the hardware as submultiples of the 60 Hz vertical rate. One (both bits = 1) provides for an externally generated crawl clock, which may be, for example, an oscillator whose frequency is controlled by a joystick. Bit 3 is the "go" bit which initiates a crawl sequence when set = 0 during a screen port output.

Upon issuance of a "crawl up" sequence, the display will wait until the onset of VDISP and will then appear to "jump down" one character line. The old top line will still appear at the top of the display, but it will be displaced down by one character line. The display will gradually crawl back up to its original state, advancing one scan for each crawl clock. When the starting point is reached, bit 0 of the status port (CRAWL DONE) will set = 1 (it sets = 0 immediately upon issuance of a crawl sequence command).

It should be obvious that if, immediately following the issuance of a "crawl up" command and before the onset of VDISP, the new bottom line is written into the old top line and the display is scrolled up one line, the result will be the disappearance of the top line and a slow scroll up, with the old top line appearing at the bottom of the screen with its new contents.

When a "crawl down'. sequence is initiated, the subsequent onset of VDISP will cause the display to move down one scan, with the bottom scan of the bottom line obscured. This downward movement will continue, one scan for each crawl clock, until the bottom line has been completely obscured. The CRAWL DONE bit will now return t-o = 1. At this point the new top line should be written into the old bottom line and the screen should be scrolled down one line. The newly written line will not become visible until after a new "crawl down" sequence is initiated.

If the newly-written top line is that last of a sequence and it is desired that the crawling stop, issue the crawl down command and follow it immediately with an output to the screen port having the direction bit = 0 (up) and the GO bit = 1. The new line will appear at the top of the screen and on the next crawl clock will move up one scan. CRAWL DONE will then set = 1.

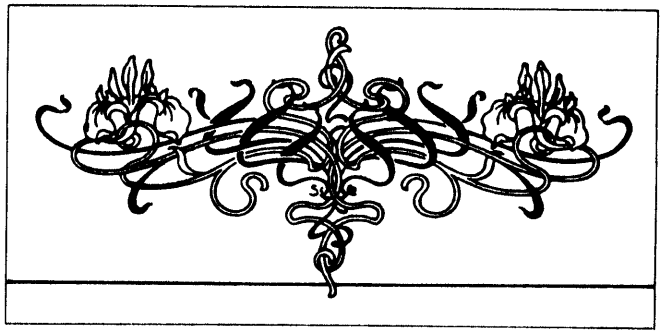
When the external crawl clock is selected it is useful to have an externally provided "go request" and "direction request" set of signals so that text movement may be controlled without use of the standard input channel such as the keyboard. Bits 4 and 5 of the status port are connected to the external device connector and are read by the VDM-2 when the status port is read. These bits are advisory to the CPU and have no function in the VDM-2 hardware. They will show a default "1" value when no external device is connected.

Subtext

In the section on scrolling it was mentioned that four memory text lines remain unseen when the VDM-2 is in the 20 line mode. These text lines may still be accessed by the CPU, and constitute a "subtext" area for information of secondary utility. Upon command from the CPU these lines can be made to appear at the bottom of the screen obscuring one or more of the text lines normally displayed.

Bit 5 of the control register (SUBTEXT 0) will cause the display of one or more of these lines when set = 1. The number of subtext lines to be displayed is determined by the binary value of bits 6 and 7 of the control register (SUBTEXT 1 and 2). Taking bit 7 as the most significant bit, values 0, 1, 2, and 3 represent 1, 2, 3, and 4 displayed subtext lines. These lines always end at the bottom of the screen and the topmost line is always memory text line 20. The subtext lines do not respond to crawl or scroll commands.

Subtext lines are intended for uses such as prompting, system status information display, menus, "jack-in-the-box" quick comebacks, etc. Use of the subtext should be avoided in the 24 line mode, since the subtext lines will be duplicates of lines visible elsewhere on the screen.



Interrupts

Interrupts have been left unimplemented in the current prototype S-100 version of the VDM-2. Since interrupts are a system consideration and since the VDM-2 is currently without a system, many configurations are possible.

As a minimum, the CRAWL DONE bit could cause an interrupt by its onset when interrupts are enabled. This would allow smooth scrolling with a minimum of CPU attention.

Bit 7 of the screen port has been tentatvely designated as INTERRUPT ENABLE. Its exact mode of functioning awaits the final design of the device.

```
BIT SUMMARY  --  OUTPUT PORTS

Control port

 bit no.   Name          Function

   0       DISABLE     - Prevents memory decoder from responding
                         when set = 1
   1       FONT        - Causes font memory (writeable) to re-
                         spond to CPU memory references when
                         set = 1
   2       PAGE        - Causes high order four bits of screen
                         memory character word to respond to
                         CPU memory references when set = 1
   3       BLINK       - Determines whether cursor or video blinks
                         on characters having page 1 bit 2 set = 1.
                         0 causes video blink, 1 causes cursor blink.
   4       24          - Determines number of character lines per
                         screen. 0 causes 20 lines of 12 scan each.
                         1 causes 24 line display of 10 scans each.
   5       SUBTEXT 0   - Controls display of subtext lines at
                         bottom of screen. 1 = subtext displayed.
   6       SUBTEXT 1   - Least significant bit pair
   7       SUBTEXT 2   - Most significant bit of pair
                         Bit pair value determines number of
                         subtext lines to be displayed. Number of
                         lines is one plus binary value of bit pair.

Screen port

 bit no.   Name          Function

   0       DIRECTION   - Sets direction of crawl. 0 = up.

   1       Speed 0
   2       SPEED 1     - As a binary pair (bit 1 least significant)
                         sets one of four crawl speeds:
                            value     speed
                              0       60 scans/ sec.
                              1       30 scans/sec.
                              2       15 scans/ sec.
                              3       externally clocked
   3       GO          - Intitiates crawl operation when set
                         = 0
   4       SCROLL      - Increments the number of the memory text
                         line displayed at the top of the screen.
                         0 = inorement
   5       RESET       - When set = 0 causes memory text line 0
                         to be displayed at top of screen.
                         Overrides effect of bit 4.
   7       INTERRUPT
             ENABLE    - reserved for use in interrupt version


BIT SUMMARY -- INPUT PORT

 Status port

bit no.     name          function

0       CRAWL DONE      Sets = 0 when crawl is in progress.

1       VDISP           Sets = 1 during vertical display time
                        (240 scans)

2       CRAWL CLOCK     Sets = 1 during crawl clock signal.
                        Crawl advances one scan at onset of
                        VDISP following leading edge of crawl
                        clock.

3       unused

4       EXT. CRAWL RQST   Sets = 0 when external crawl-control
                          hardware is requesting text movement

5       EXT CRAWL DIRECTION   sets = 0 when external crawl
                              control hardawre requests crawl
                              down.

        (note- bits 4,5 are set = 1 when no external device
               is connected.)
```

Continued from pg. 35

Copyrights

As usual, I assert the free-diffusion-clause copyright defined in *DDJ* #32 and earlier articles. I recommend this to algorithmaniacs of my ilk, since it encourages users to let you know of unexpected applications *they* have found for your creations.

Hal Gordon is 60 years old, has a Ph.D (Biology, Harvard, 1947), and has been at UC – Berkeley since 1947. He is a low-level programmer in more ways than one, having learned how to use a CPU directly on an early KIM-1 and only recently upgraded to a SYM-1. His (still unrealized) dream is to equip these boards with sensors and effectors so that they can control real-time biological experiments. He sympathizes with the anti-AI views of Joseph Weizenbaum on the limits of human-language-oriented programs, believing that it's more instructive to interact with books and the right kind of human mind.

```
     (listing of subroutine MIXSIM, without addresses since
       it is fully relocatable)

A6 DF    MIXSIM LDX MEMEX    (load prev. G-index into X)
CA              DEX          (decrement X for next G)
30 0D           BMI RESET    (if = $FF, go reset X to 02)
B5 E1           LDA RND+1,X  (load G+1 seed into Acc.)
D0 0B           BNE SIMRND   (if ≠ 0, run normal G logic)
B5 E2           LDA RND+2,X  (load G+2 seed into Acc.)
F0 07           BEQ SIMRND   (if = 0, run normal G)
B5 E0           LDA RND,X    (load G seed into Acc.)
18              CLC          (clear carry for branch)
90 10           BCC DEJUM    (output pseudo-G number)
A2 02    RESET  LDX #$02     (reset X to 02)

B5 E0    SIMRND LDA RND,X    (load G seed into Acc.)
0A              ASL A        (X 2)
0A              ASL A        (X 4)
38              SEC          (set carry to add 1)
75 E0           ADC RND,X    (X 5, + 1)
95 E0           STA RND,X    (store new seed)

18              CLC          (clear carry for addition)
75 E3           ADC RND+3,X  (add addend for G variant)

10 02           BPL STOREX   (bypass DEJUM if bit 7 = 0)
49 7F    DEJUM  EOR #$7F     (complement lowest 7 bits)

86 DF    STOREX STX MEMEX    (store G-index value)
60              RTS          (exit subroutine)

     Zero-page locations used (all must be initialized,
       cf. text for restrictions):

DF   MEMEX  (must be 02 or 01 or 00)

E0   RND    (seed of G0)
E1   RND+1  (seed of G1)
E2   RND+2  (seed of G2)

E3   RND+3  (addend of G0, must not be = 00)
E4   RND+4  (addend of G1, ≠ RND+3 and ≠ RND+5)
E5   RND+5  (addend of G2, ≠ RND+3 and ≠ RND+4)
```