LEO-1 Homebrew Computer

Visual Display Unit (VDU)

Rev 1.0, 29th May 2017, John Croudy

The LEO-1 VDU is a fairly simple video board designed to work with the LEO-1. It should also work with any other 16-bit computer or an 8-bit computer with extra hardware to enable 16-bit memory writes. The design was inspired by (and partly based on) the video section of the Ohio Scientific Superboard II. Part of the counter chain and the character generation was based on the Superboard, but the rest was designed by me. Features include:

- Character-based progressive (non-interlaced) composite video output.
- 2048 16-bit words of video memory directly addressable by the LEO-1 CPU. This is organized as 2048 x 8-bit character data bytes and 2048 x 8-bit character attribute bytes.
- 40 video modes consisting of 4 horizontal modes combined with 10 vertical modes in any combination.

Memory

The 2048 words of video memory are implemented using two modern 128k RAM chips. This is because these chips are easier to get and use less power than the real 2K chips that would have been used in the late 1970s. The circuit could easily be modified to use any RAM chips that happen to be available. Conversely, the character ROM is implemented with an old Am27C64 UV-erasable EPROM (8K) but the circuit could easily be modified to use any modern EEPROM. Only 4096 bytes are needed to get 256 characters with 16 rows in each.

Video signal

The video output is a simplified monochrome composite video signal with a 64 microsecond scan line, a 4.6 microsecond horizontal blanking pulse and a 68 microsecond vertical blanking pulse. This follows neither the PAL nor the NTSC specification but is close enough to work anyway. The vertical refresh rate is around 60Hz which is more like NTSC than PAL. Most TVs and video monitors are not terribly fussy about the precision of the signal. The horizontal sync pulse is the most important, and is usually 4.7 microseconds long, but anything from 4 to 5 microseconds seems to work. What's more, a simple vertical blank pulse is all that's needed to get a progressive scan. With modern screens there's no need for the complicated equalization pulses that are supposed to appear in the vertical blank period. The voltage levels are 0v for sync, about 0.3v for black, and about 1v for white. A resistor ladder forms a simple D to A converter which generates 16 grey levels. The sync pulses are inserted by using them to switch the main signal on and off. This forms the complete composite signal which works well with a 75 ohm composite video monitor.

References

http://www.batsocks.co.uk/readme/video_timing.htm
http://martin.hinner.info/vga/pal.html
http://www.rickard.gunee.com/projects/video/pic/howto.php

Method of operation

The VDU operates by sequentially reading out memory locations, converting the data found there to character data, and shifting that data out to form a 'pixel mask'. This mask is then used to choose either the background or foreground colour value to send to the display. The 8-bit character data comes from the *Data RAM* (U26) and the two 4-bit colour values come from the *Attribute RAM* (U27). A pre-programmed EPROM (U28) is used to look up the character pixel values from the character code fetched from Data RAM.

The timing of the entire system is derived from a *main clock* (CLK) which is programmable to be either 16MHz, 12MHz, 8MHz or 4MHz. Pixels are output to the screen at a rate defined by the *pixel clock* (PIXCLK) which is half the frequency of the main clock. The main clock drives a counter chain (U15, U16, U17) which generates the pixel clock and a 10-bit horizontal counter value (HC0 to HC9). HC0 to HC2 represent the pixel position within a character (see below) while HC3 to HC8 form a 6-bit code that is fed to the RAM address bus (RAMA0 to RAMA5) giving a maximum of 64 characters on each line.

Character pixel generation

The 3-bit value from HC0 to HC2 is inverted by U32 and fed to NAND gate U34 along with the pixel clock. When the value resets to zero, the output of U34 goes low (in sync with the pixel clock going high) and this causes the shift register (U29) to latch the output of the character ROM. Subsequent pixel clocks shift the value out of the shift register and into *attribute mux* U31. This mux chooses between two 4-bit values derived from the output of the Attribute RAM (U27) which is latched by register U30 on every clock tick. Thus, the character data shifted out of the shift register is used as a 1-bit mask to choose either the background or foreground colour stored in the Attribute RAM at the same address as the character itself. The output of the attribute mux is a 4-bit value which is converted by a resistor ladder (R4 to R11) to a voltage representing the grey level. This in turn is fed to transistor Q2 to produce the 'picture' part of the video signal. As HC0 to HC2 continually count from 0 to 7, a pixel row of the current character block is sent out to the screen.[1] When they reset to zero at the end of the character, HC3 increments, the next character on the line is addressed and the sequence repeats until the end of the line is reached.

Scan line termination

The topmost 3 bits of the horizontal counter (HC7 to HC9) are fed to comparator U25 and compared with the value stored in the Horizontal Control Register (U22). When the values are equal, /HRES goes low to indicate the end of the scan line. This causes the horizontal counter

[1] Character blocks are therefore always 8 pixels wide regardless of the graphics mode.

to be reset to zero and generates the horizontal sync pulse (/HS) by triggering a one-shot (U4.2) with a timing of about 4.6 microseconds.[2]

Character row generation

The horizontal sync pulse is also used to clock U18 which is the *character row counter*. This counter drives the low four bits of the Character ROM (U28) address bus and causes the ROM to output the rows of whichever character is being generated. This character is chosen by the ROM's A4 to A11 lines (8 bits) and is determined by the output of the Data RAM at that moment. Thus, for each line scanned, a different row of each character is scanned out. The number of rows in each character depends on the value *VCODE1* in the low 4 bits of the Vertical Control Register (U21). This value is compared by comparator U23 with the output of the character row counter (U18). When the values are equal, the character row counter is reset to zero and the *character line counter* (U19) is incremented. The value of the character line counter (VC0 to VC4) is used to control the top 5 bits of the RAM address giving a maximum of 32 possible character rows on the screen. The value *VCODE2* in the high 4 bits of the Vertical Control Register (U21) is compared by comparator U24 with the value of the character line counter (U19). When they are equal *and* VC4 is high, the /FLYBACK signal is asserted.

Vertical flyback

The /FLYBACK signal causes the character line counter to be reset to zero. At the same moment, two vertical sync pulses (and their complements) are generated, one (/VS) by one-shot U4.1 with a length of 68 microseconds, and another much longer one (VBL) by one-shot U3. /VS is combined with /HS (by U2) to produce the complete sync signal which is fed to transistor Q1 and on to the video output. VBL is sent, along with its complement, to an output port where it may be sensed by the host computer system in order to determine when it is 'safe' to write to the video RAM. This signal is asserted roughly from the time the vertical blank starts until the electron beam reappears at the top of the screen. If the video RAM is written to at any other time, it will cause interference in the form of 'snow' to appear on the screen.

[2] Note that the memory is read out from the very start of the line until the horizontal blank at the end of the line. This means that the required *front porch* and *back porch* in the video signal rely on the memory being zero in the regions that are off the screen. If there are characters in these regions, they will not be displayed but the screen image may be distorted due to the front and/or back porches being at the wrong level.

Video RAM and Control Register access

When the host computer wants to access the video RAM, it must assert `/RAMEN` and then assert `/OE` (read) or `/WE` (write). These signals are processed by inverter U1 and AND gate U2 to produce the `/RAMOE` signal used by the RAM chips and also to control the direction and output-enable of bus transceivers U9 and U10. By treating the video board as a memory-mapped device whose ‘chip enable’ is `/RAMEN`, the host computer will access video RAM during normal read and write operations. The same is true of the Control Registers which use `/CTRLEN`, except this is a write-only operation that stores the data bus into Control Registers U21 and U22.

TODO: Address muxes

Horizontal modes

Four horizontal modes are available. The H Control Code specifies the mode and must be 1, 2, 3 or 4. Any other value will result in a blank display. This code selects one of four frequencies for the main clock which in turn provides the pixel timing. Each frequency corresponds to a certain number of pixels across the 64-microsecond scan line.

Mode	HCODE	Clock freq (MHz)	Pixel clock (MHz)	Pixels on line	H Control Code
0	001	4	2	128	01
1	010	8	4	256	02
2	011	12	6	384	03
3	100	16	8	512	04

Vertical modes

Ten vertical modes are available. The V Control Code specifies the mode and must be a specific value for the mode as specified in the table below. Most other values will result in an unstable display which may damage some video monitors. The control code is an 8-bit value which is interpreted as two 4-bit values, *VCODE1* and *VCODE2*. The former corresponds to the character height in pixels and the latter to the number of character rows minus 16. These values are used to reset the vertical counters at the right moment to generate the required character height and row count. This rather peculiar set of modes is the result of the fairly simple design using 4-bit comparators and has an unfortunate side-effect: there is no mode that provides characters that are exactly 8 or 16 pixels high. Furthermore, none of the vertical refresh rates are exactly 60Hz, but most video monitors can handle some variation with adjustment of the vertical hold control.

Mode	VCODE1	VCODE2	Char height	Char Rows	Lines	Vertical refresh frequency (Hz)	V Control Code
0	1001	1101	9	29	261	59.9	D9
1	1001	1100	9	28	252	62.0	C9
2	1010	1010	10	26	260	60.1	AA
3	1010	1001	10	25	250	62.5	9A
4	1011	1000	11	24	264	59.2	8B
5	1100	0110	12	22	264	59.2	6C
6	1100	0101	12	21	252	62.0	5C
7	1101	0100	13	20	260	60.1	4D
8	1110	0010	14	18	252	62.0	2E
9	1111	0001	15	17	255	61.3	1F