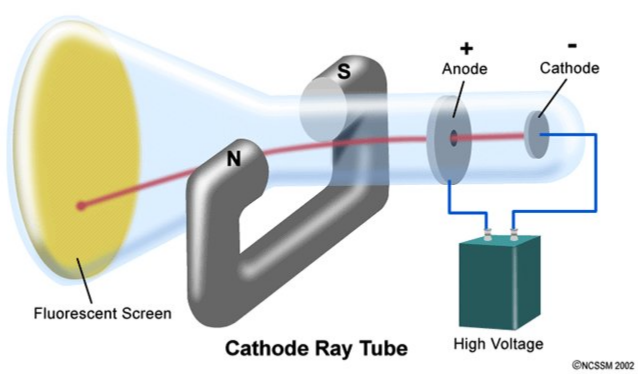
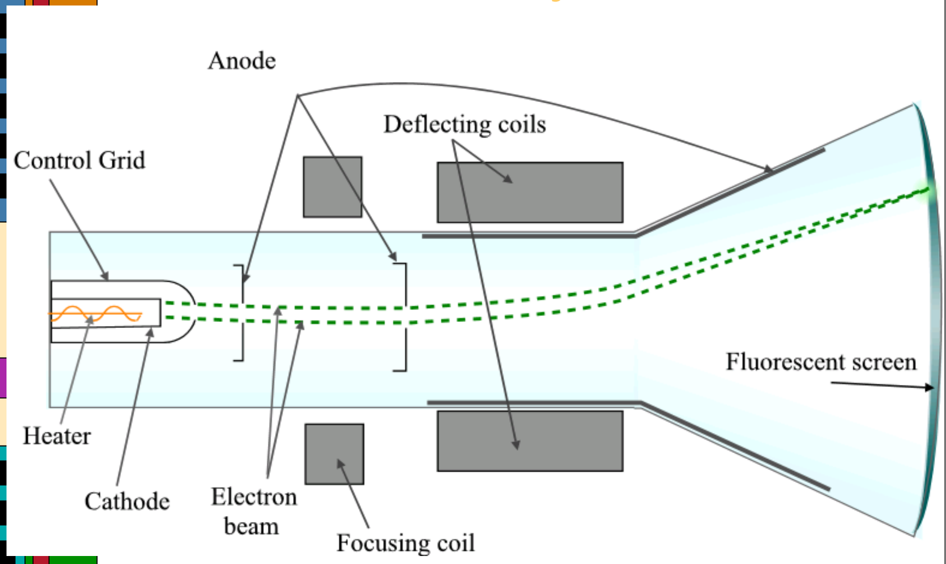
Display Technology

- Images stolen from various locations on the web...

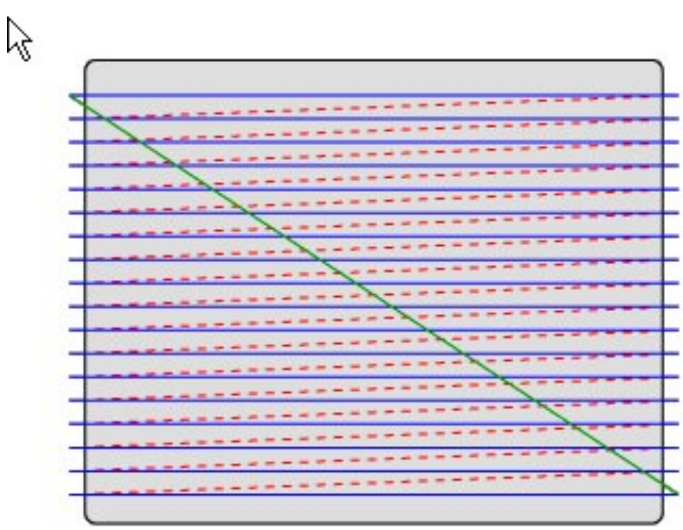
Cathode Ray Tube



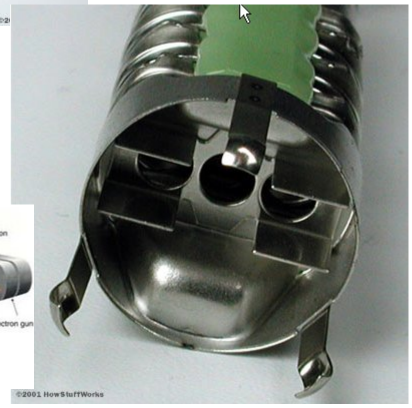
Cathode Ray Tube

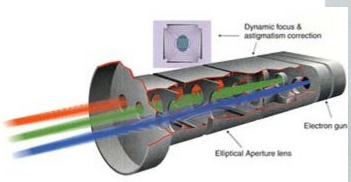


Raster Scanning

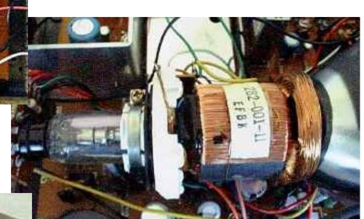


Electron Gun

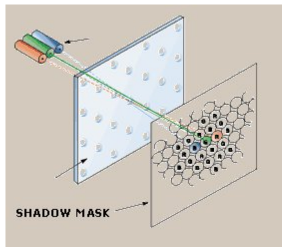


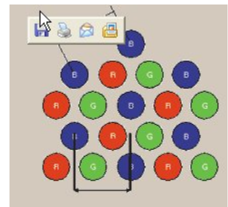


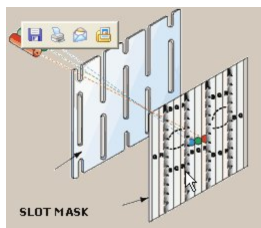
Beam Steering Coils

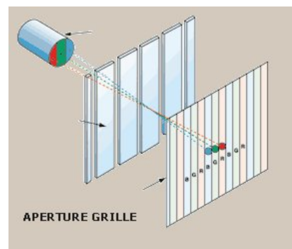


Color

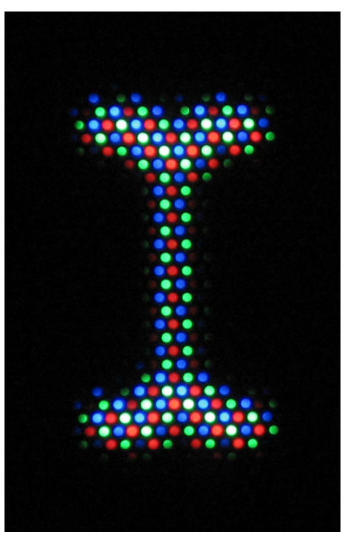


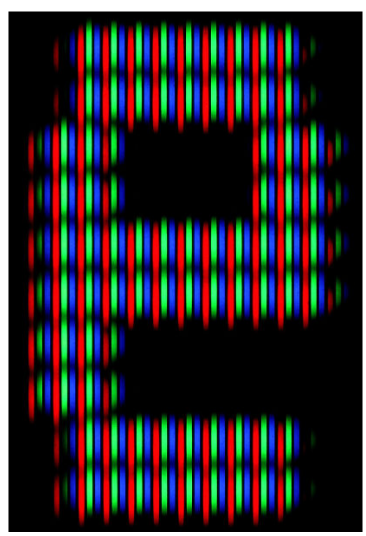




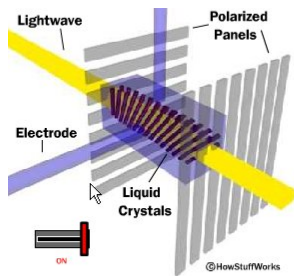


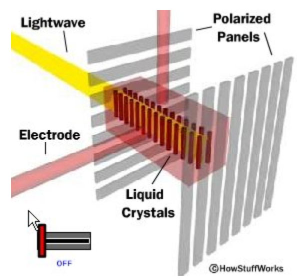
Shadow Mask and Aperture Grille



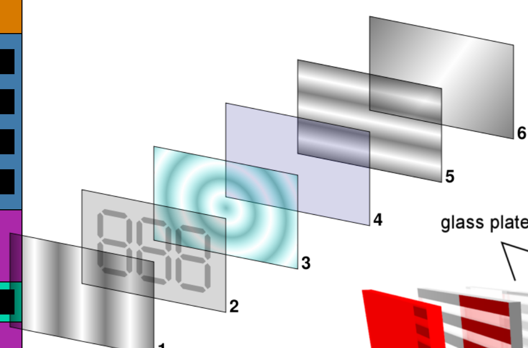


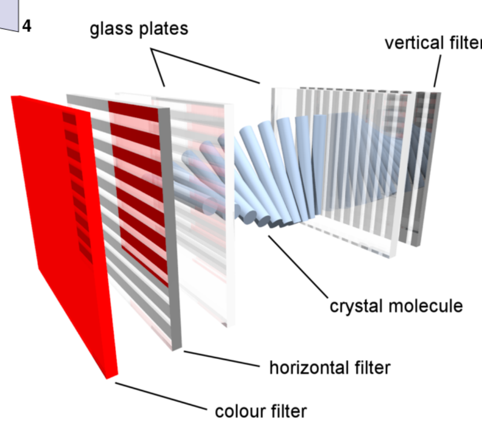
Liquid Crystal Displays



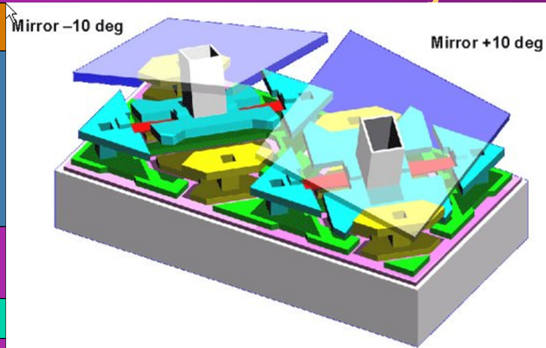


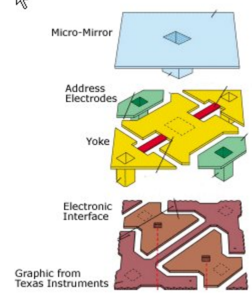
Liquid Crystal Displays

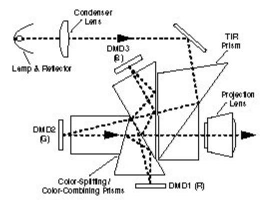




DLP Projector

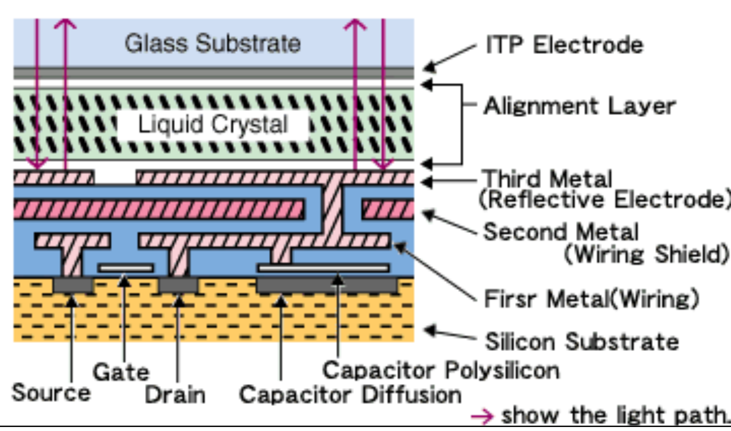




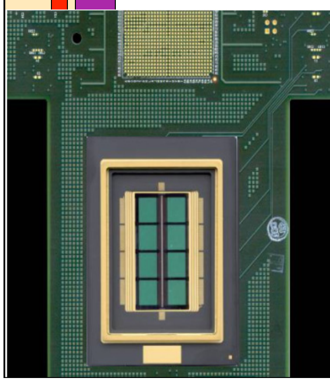


LCoS

- Liquid Crystal on Silicon
  - Put a liquid crystal between a reflective layer on a silicon chip

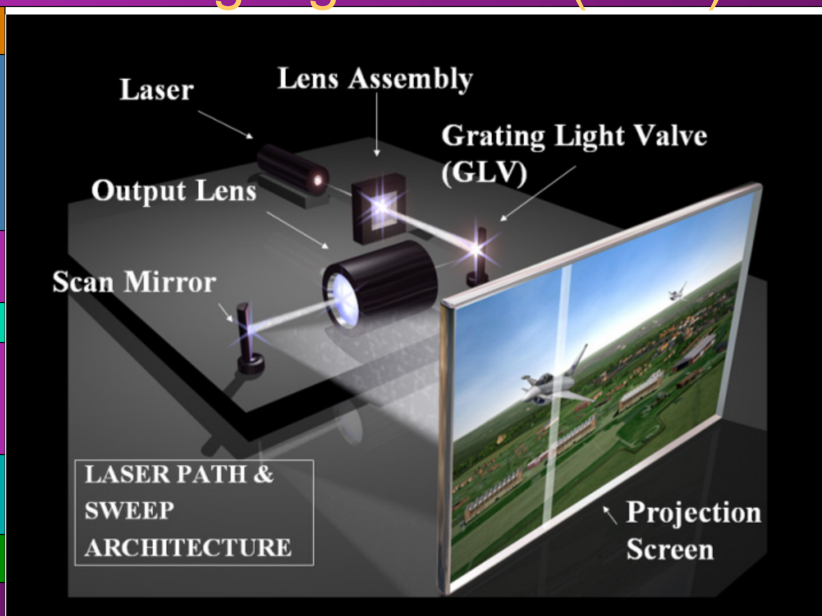


Grating Light Valve (GLS)

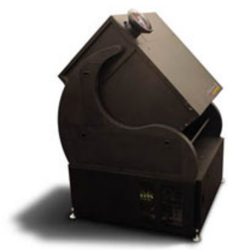


- lots (8000 currently) of micro ribbons that can bend slightly
  - Make them reflective
  - The bends make a diffraction grating that controls how much light where
  - Scan it with a laser for high light output
  - 4000 pixel wide frame ever 60Hz

Grating Light Valve (GLS)



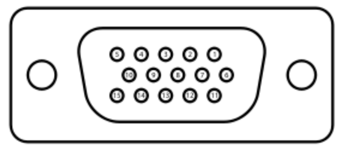
Digistar 3 Dome Projector



VGA

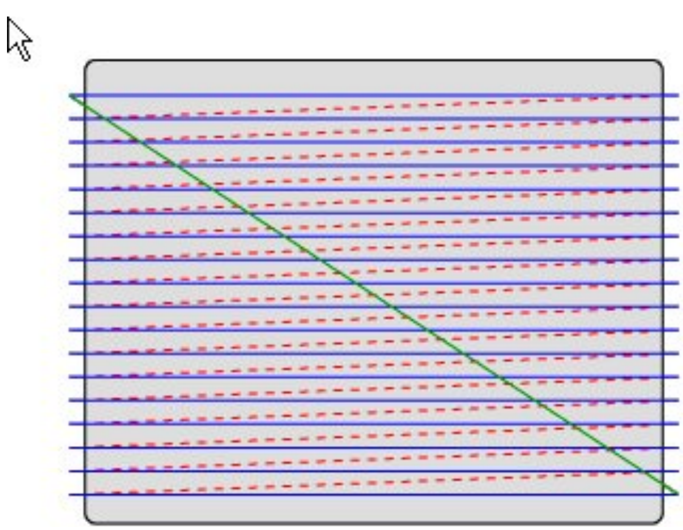
- Stands for Video Graphics Array
- A standard defined by IBM back in 1987
  - 640 x 480 pixels
  - Now superseded by much higher resolution standards...
- Also means a specific analog connector
  - 15-pin D-subminiature VGA connector

VGA Connector

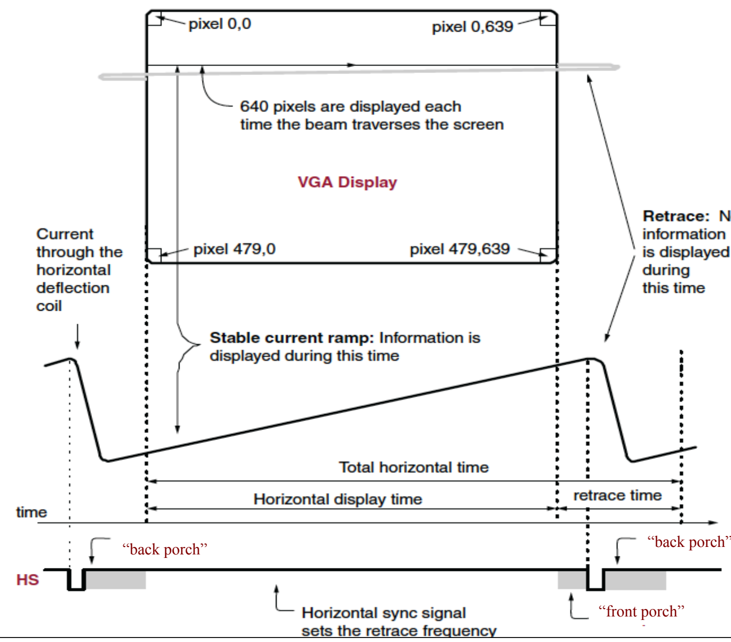


1: Red out	**6**: Red return (ground)	**11**: Monitor ID 0 in
2: Green out	**7**: Green return (ground)	**12**: Monitor ID 1 in or data from display
3: Blue out	**8**: Blue return (ground)	**13**: Horizontal Sync
4: Unused	**9**: Unused	**14**: Vertical Sync
5: Ground	**10**: Sync return (ground)	**15**: Monitor ID 3 in or data clock

Raster Scanning



Raster Scanning



VGA Timing

Horizonal Dots	640
Vertical Scan Lines	480 60Hz vertical frequency
Horiz. Sync Polarity	NEG
A (µs)	31.77 Scanline time
B (µs)	3.77 Sync pulse length
C (µs)	1.89 Back porch
D (µs)	25.17 Active video time
E (µs)	0.94 Front porch

```
               ______________________          ________
________|             VIDEO           |________| VIDEO (next line)
    |-C-|----------D-----------|-E-|
__   ______________________________   ___________
  |_|                              |_|
  |B|
  |---------------A----------------|
```

VGA Timing

Horizonal Dots	640	
Vertical Scan Lines	480	60Hz vertical frequency
Horiz. Sync Polarity	NEG	
A (μs)	31.77	Scanline time
B (μs)	3.77	Sync pulse length
C (μs)	1.89	Back porch
D (μs)	25.17	Active video time
E (μs)	0.94	Front porch

25.17/640 = 39.33ns/pixel = 25.4MHz pixel clock

```
         _______________________         ________
________|          VIDEO          |_______|  VIDEO (next line)
    |-C-|----------D-----------|-E-|
__   _____________________________   __________
  |_|                              |_|
  |B|
  |---------------A----------------|
```

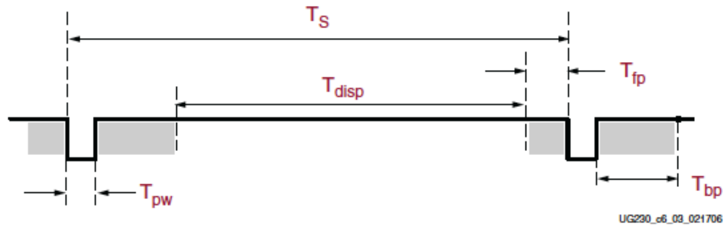
VGA Timing

Horizonal Dots	640	
Vertical Scan Lines	480	
Vert. Sync Polarity	NEG	
Vertical Frequency	60Hz	
O (ms)	16.68	Total frame time
P (ms)	0.06	Sync pulse length
Q (ms)	1.02	Back porch
R (ms)	15.25	Active video time
S (ms)	0.35	Front porch

```
         _______________________         ________
________|          VIDEO          |_______|   VIDEO (next frame)
    |-Q-|----------R-----------|-S-|
__   _____________________________   __________
  |_|                              |_|
  |P|
  |---------------O----------------|
```

VGA Timing Summary

Symbol	Parameter	Vertical Sync			Horizontal Sync	
		Time	Clocks	Lines	Time	Clocks
T_S	Sync pulse time	16.7 ms	416,800	521	32 µs	800
T_{DISP}	Display time	15.36 ms	384,000	480	25.6 µs	640
T_{PW}	Pulse width	64 µs	1,600	2	3.84 µs	96
T_{FP}	Front porch	320 µs	8,000	10	640 ns	16
T_{BP}	Back porch	928 µs	23,200	29	1.92 µs	48



60 Hz refresh and 25MHz pixel clock

Relaxed VGA Timing

- This all sounds pretty strict and exact...
- It's not really... The only things a VGA monitor really cares about are:
  - Hsync
  - Vsync
  - Actually, all it cares about is the falling edge of those pulses!
  - The beam will retrace whenever you tell it to
  - It's up to you to make sure that the video signal is 0v when you are not painting (i.e. retracing)

Relaxed VGA Timing

Horizonal Dots	128	
Vertical Scan Lines	?	60Hz vertical frequency
Horiz. Sync Polarity	NEG	
A (μs)	30.0	Scanline time
B (μs)	2.0	Sync pulse length
C (μs)	10.7	Back porch
D (μs)	12.8	Active video time
E (μs)	4.50	Front porch

12.8/128 = 100ns/pixel = 10 MHz pixel clock

```
         ___________________________     _________
________|          VIDEO            |_________|  VIDEO (next line)
    |-C-|----------D----------------|-E-|
__   ______________________________    ___________
  |_|                              |_|
  |B|
  |---------------A----------------|
```

VGA Timing

Horizonal Dots	128	
Vertical Scan Lines	255	
Vert. Sync Polarity	NEG	
Vertical Frequency	60Hz	
O (ms)	16.68	Total frame time
P (ms)	0.09	Sync pulse length (3x30μs)
Q (ms)	4.86	Back porch
R (ms)	7.65	Active video time
S (ms)	4.08	Front porch

```
         ___________________________     _________
________|          VIDEO            |_________|  VIDEO (next frame)
    |-Q-|----------R----------------|-S-|
__   ______________________________    ___________
  |_|                              |_|
  |P|
  |---------------O----------------|
```

VGA Voltage Levels

- Voltages on R, G, and B determine the color
  - Analog range from 0v (off) to +0.7v (on)
  - But, our pads produce 0-3.3v outputs!

VGA Voltage Levels

- Voltages on R, G, and B determine the color
  - Analog range from 0v (off) to +0.7v (on)
  - But, our pads produce 0-3.3v outputs!
  - For B&W output, just drive RGB together and let 0v=black and 3.3v=white
    - overdrives the input amps, but won't really hurt anything
  - For color you can drive R, G, B separately
    - Of course, this is only 8 colors (including black and white)
    - Requires storing three bits at each pixel location

VGA on Spartan3e Starter

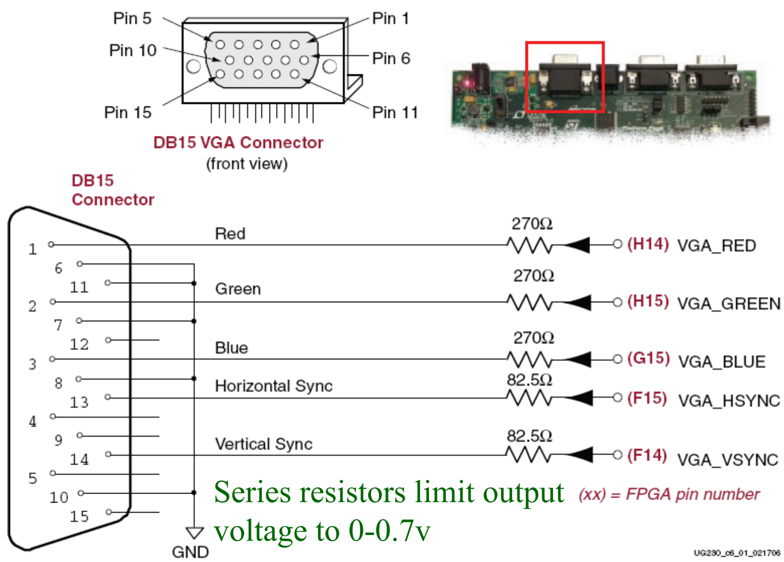


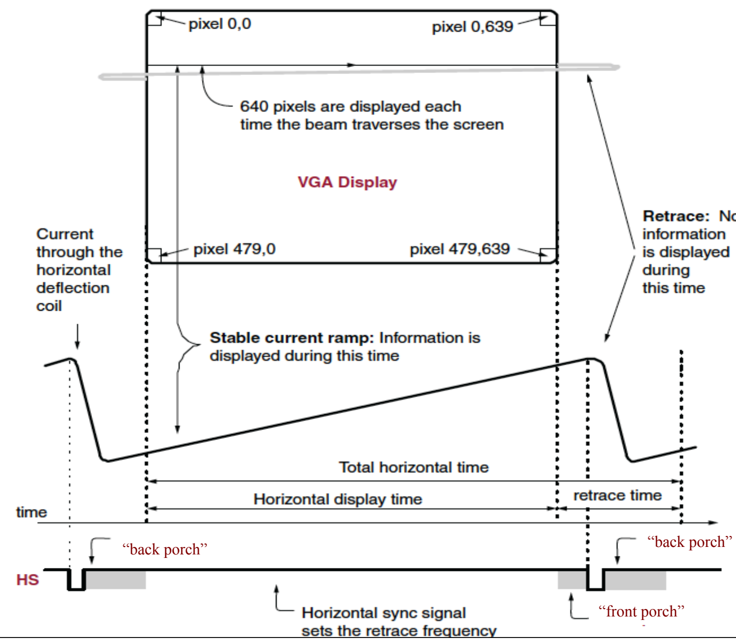
Figure 6-1: **VGA Connections from Spartan-3E Starter Kit Board**

VGA on Spartan3e Starter

Table 6-1: **3-Bit Display Color Codes**

VGA_RED	VGA_GREEN	VGA_BLUE	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Raster Scanning



VGA on Spartan3e Starter

Table 6-2: **640x480 Mode VGA Timing**

Symbol	Parameter	Vertical Sync			Horizontal Sync	
		Time	Clocks	Lines	Time	Clocks
T_S	Sync pulse time	16.7 ms	416,800	521	32 µs	800
T_{DISP}	Display time	15.36 ms	384,000	480	25.6 µs	640
T_{PW}	Pulse width	64 µs	1,600	2	3.84 µs	96
T_{FP}	Front porch	320 µs	8,000	10	640 ns	16
T_{BP}	Back porch	928 µs	23,200	29	1.92 µs	48

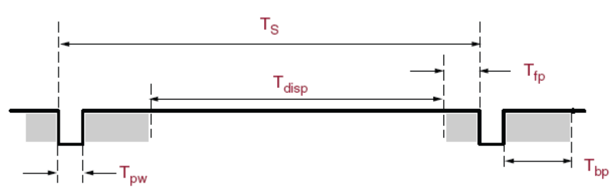


Figure 6-3: **VGA Control Timing**

The signal timings in Table 6-2 are derived for a 640-pixel by 480-row display using a 25 MHz pixel clock and 60 Hz ± 1 refresh. Figure 6-3 shows the relation between each of the timing symbols. The timing for the sync pulse width (T_{PW}) and front and back porch intervals (T_{FP} and T_{BP}) are based on observations from various VGA displays. The front and back porch intervals are the pre- and post-sync pulse times. Information cannot be displayed during these times.

VGA on Spartan3e Starter

Figure 6-4 provides the UCF constraints for the VGA display port, including the I/O pin assignment, the I/O standard used, the output slew rate, and the output drive current.

```
NET "VGA_RED"   LOC = "H14" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_GREEN" LOC = "H15" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_BLUE"  LOC = "G15" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_HSYNC" LOC = "F15" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
NET "VGA_VSYNC" LOC = "F14" | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST ;
```

Figure 6-4: **UCF Constraints for VGA Display Port**

VGA Assignment

- vgaControl
  - Generate timing pulses at the right time
  - hSync, vSync, bright, hCount, vCount
- bitGen
  - Based on bright, hCount, vCount, turn on the bits

3 Types of bitGen

- Bitmapped
  - Frame buffer holds a separate rgb color for every pixel
  - bitGen just grabs the pixel based on hCount and vCount and splats it to the screen
  - Chews up a LOT of memory

3 Types of bitGen

- Character/Glyph-based
  - Break screen into nxm piexl chunks (e.g. 8x8)
  - For each chunk, point to one of k nxm glyphs
  - Those glyphs are stored in a separate memory
  - For 8x8 case (for example)
    - glyph number is hCount and vCount minus the low three bits
    - glyph bits are the low-order 3 bits in each of hCount and vCount
    - Figure out which screen chunk you’re in, then reference the bits from the glyph memory

3 Types of bitGen

- Direct Graphics
  - Look at hCount and vCount to see where you are on the screen
  - Depending on where you are, force the output to a particular color
  - Tedious for complex things, nice for large, static things

```
parameter BLACK = 3'b 000, WHITE = 3'b111, RED = 3'b100;
// paint a white box on a red background
always@(*)
   if (~bright) rgb = BLACK; // force black if not bright
   // check to see if you're in the box
  else if (((hCount >= 100) && (hCount <= 300)) &&
            ((vCount >= 150) && (vCount <= 350))) rgb = WHITE;
  else rgb = RED; // background color
```

VGA Memory Requirements

- Remember, Spartan3e has 20 18kbit Block RAMs
  - i.e. 20k addresses where each address is a 16-bit (or 18 bit) word
  - But, 16 bits of address = 64k addresses
  - So, you can't use all the address space with just Block RAMs

VGA Memory Requirements

- 640x480 VGA (bitmapped)
  - 307,200 pixels
  - 3 bits per pixel
  - 6 pixels per 18-bit word
  - 50k locations for 640x480
  - Oops – we only have 20k, and you need some space for code and other data…

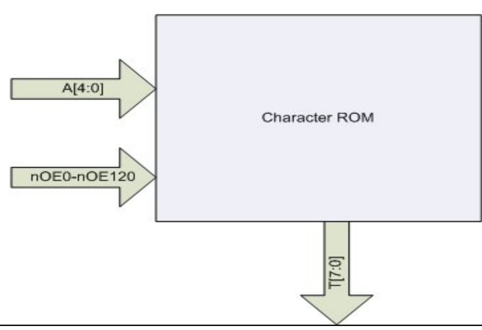
VGA Memory Requirements

- 320x240 VGA (bitmapped)
  - 76,800 pixels
  - Each stored pixel is 2x2 screen pixels
  - 3 bits per pixel
  - 6 pixels per 18-bit word
  - 12.5k 18-bit words needed
  - Much more realistic… 7.5k left over for code/ data

VGA Memory Requirements

- 80 char by 60 line display (8x8 glyphs)
  - 4800 locations
  - Each location has one of 256 char/glyphs
  - 8-bits per location – 2 locations per word
  - 2400 addresses for frame buffer
  - Each char/glyph is (say) 8x8 pixels
    - results in 640x480 display…
  - 8x8x256 bits for char/glyph table
    - 16kbits (1k words) for char/glyph table

Character Example…



64 characters
each 8x8 pixels

Character Example…

The Character ROM contains the 64 member ASCII upper-case character set. The characters are addressed with a 5-bit binary address A[4:0] and a 16-bit unary decoded address, nOE0-nOE120. The Character ROM outputs a single row of the selected character at a time on the signals T[7:0].

A[4:3] decodes one of the four rows of 16 characters in the ROM.

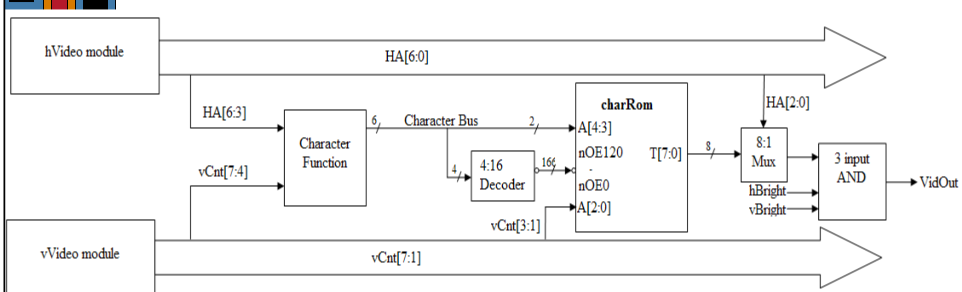
A[4:3] == 0	- first row	" !"#$%&'()*+,-./"
A[4:3] == 1	- second row	"0123456789:;<=>?"
A[4:3] == 2	- third row	"@ABCDEFGHIJKLMNO"
A[4:3] == 3	- fourth row	"PQRSTUVWXYZ[\]^_"

The sixteen signals nOE0, nOE8, nOE16, nOE24, nOE32, nOE40, nOE48, nOE56, nOE64, nOE72, nOE80, nOE88, nOE96, nOE104, nOE112, nOE120 select one of the sixteen columns of of four characters. These signals are active low and only one is asserted at any time. For instance, nOE0==0 selects the first column with the four characters " 0@P" in it and nOE7==0 selects "'7GW".

A[2:0] decodes one of the eight character rows. For instance, if the character "A" is selected with A[4:3]==2 and nOE8 then A[2:0] will produce the following binary output on T[7:0].

		Binary	Visible Output
A[2:0] == 0	- first row	00011100	***
A[2:0] == 1	- second row	00100010	* *
A[2:0] == 2	- third row	00100010	* *
A[2:0] == 3	- fourth row	00111110	*****
A[2:0] == 4	- fifth row	00100010	* *
A[2:0] == 5	- sixth row	00100010	* *
A[2:0] == 6	- seventh row	00100010	* *
A[2:0] == 7	- eight row	00000000	

Character Example…



Remember the skier/chicken/tron example?
That used character/glyph graphics similar to this…

Tbird VGA Assignment

- Get VGA working
  - Start with full-screen flood
  - then play around with direct VGA graphics
- Take the Tbird state machine
  - outputs are six lights
- Define six regions of the screen
  - Make those regions change color when the state machine says the lights should be on

Other I/O (more details later)

- LCD display
  - 2-line 16-char display
  - Reasonably easy to use, once you can do it under program control!
  - Reading and writing memory-mapped 8-bit registers
- PS/2 mouse/keyboard port
- RS323 connector and level converter
- DAC
  - 12 bit unsigned resolution – four outputs
- ADC
  - Dual-channel – 14 bit resolution
- Seven-segment LCDs
  - Already in your kits…

See the Starter Board users guide for more details!