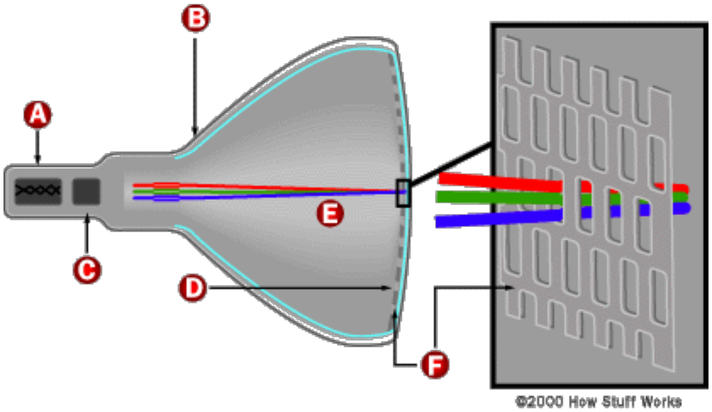
VGA!



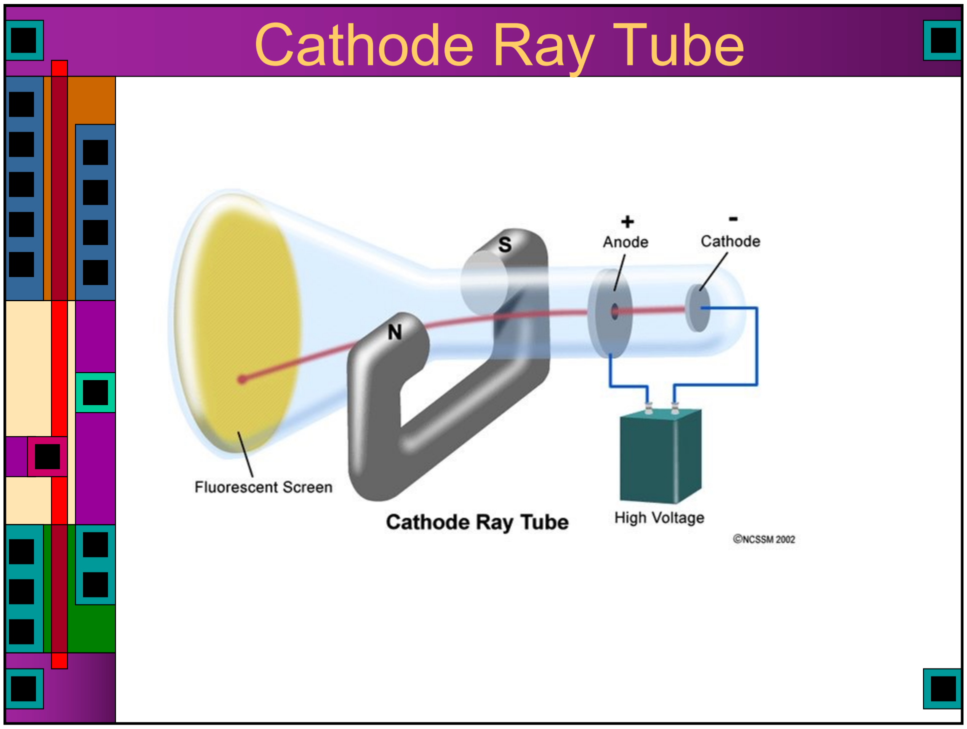
©2000 How Stuff Works

1

Display Technology

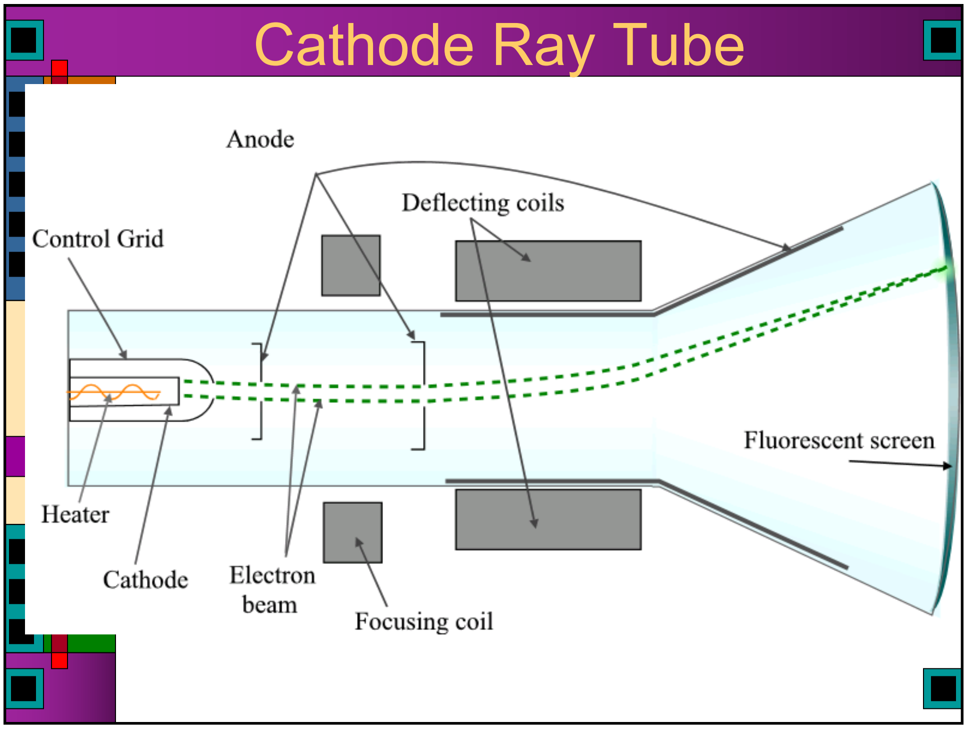
- Images stolen from various locations on the web...

2



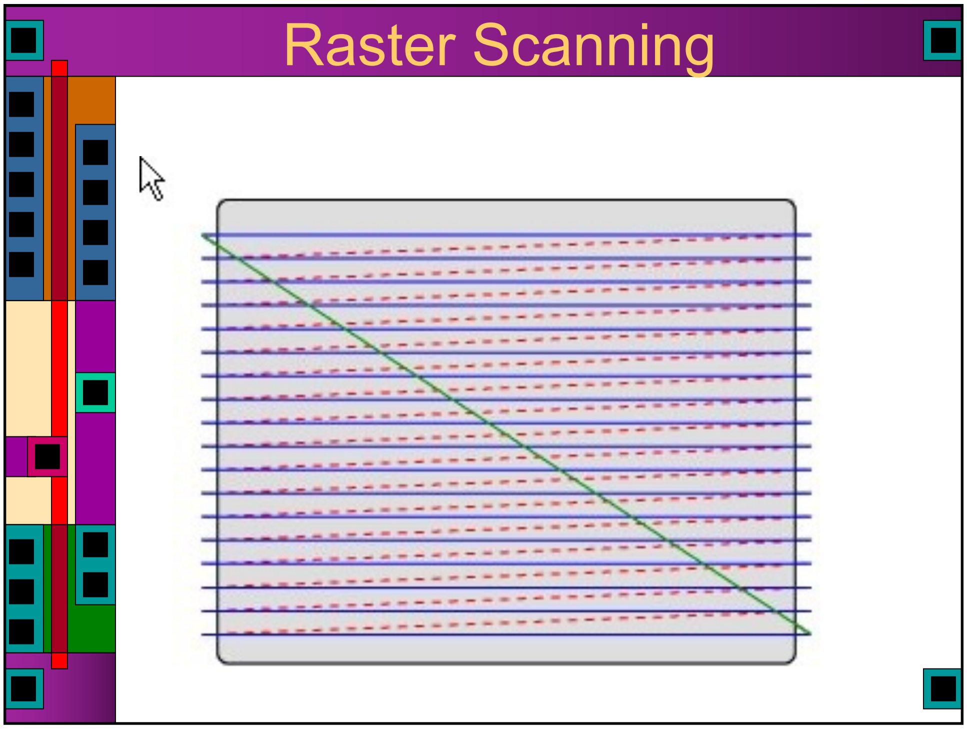
Cathode Ray Tube

3



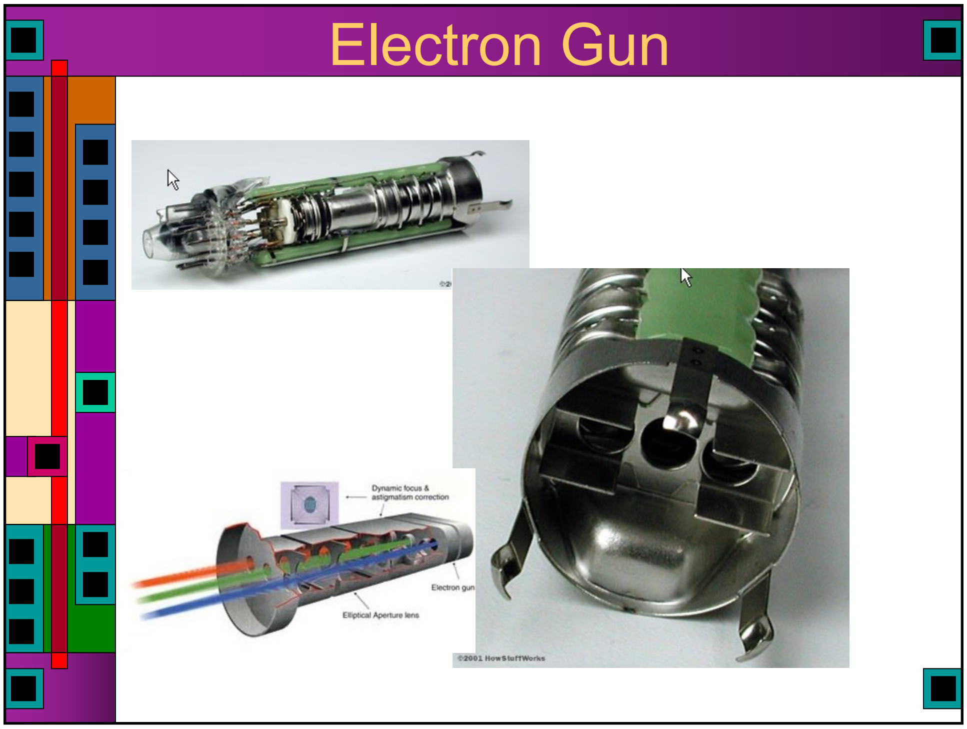
Cathode Ray Tube

4



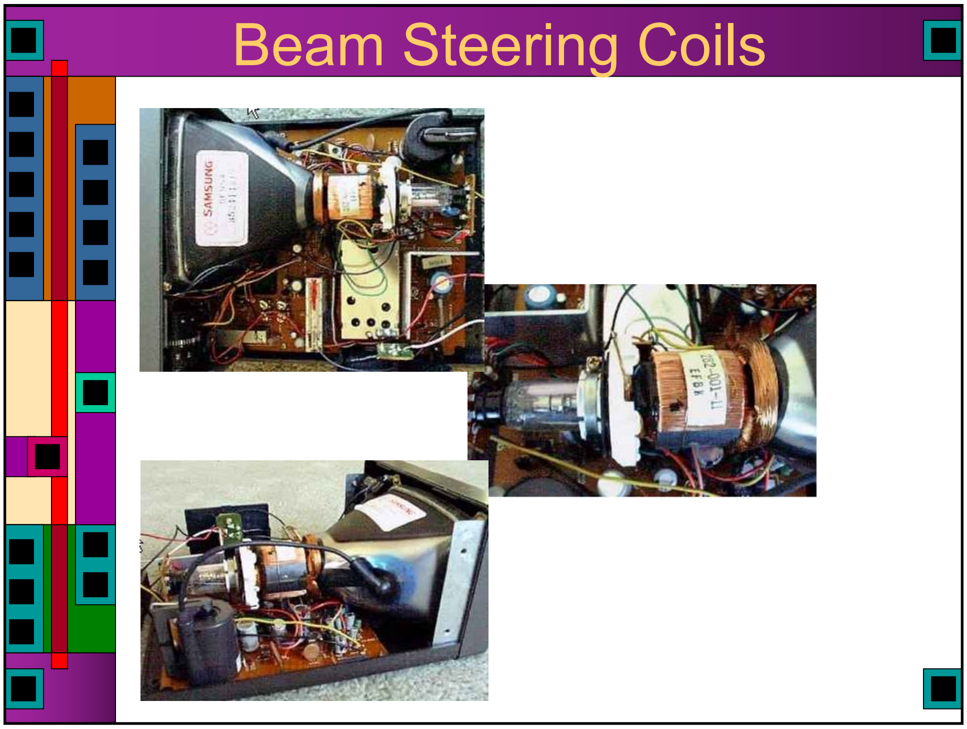
Raster Scanning

5



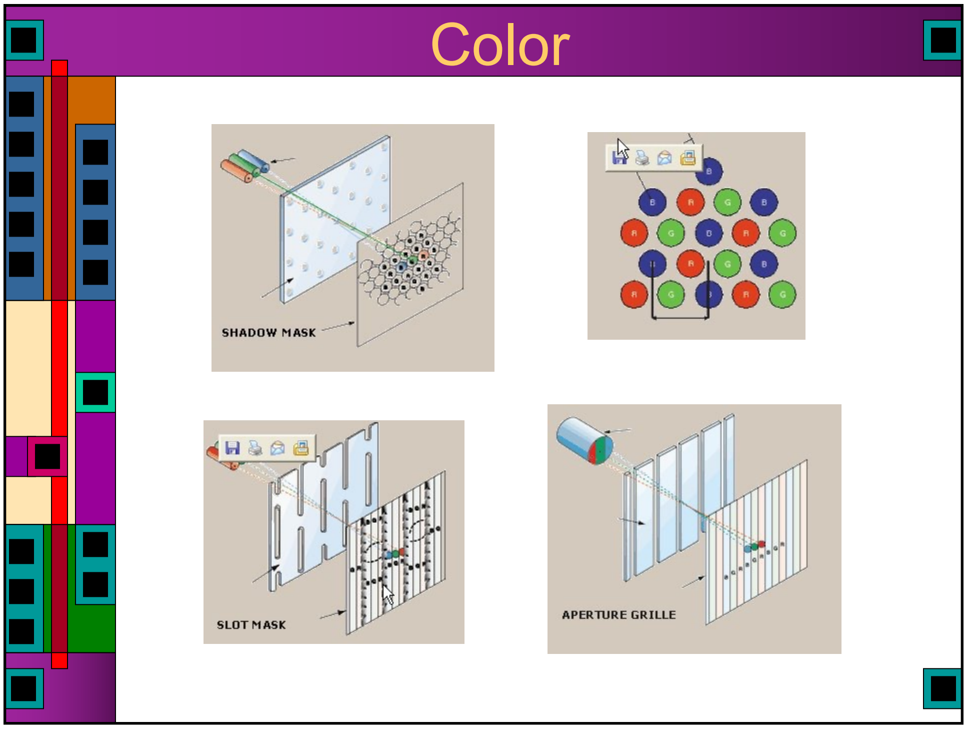
Electron Gun

6



Beam Steering Coils

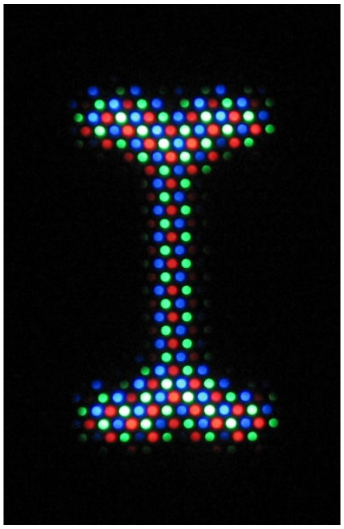
7

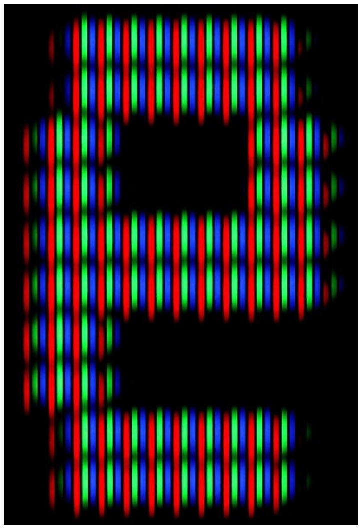


Color

8

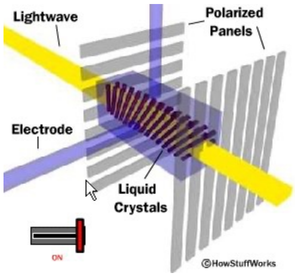
Shadow Mask and Aperture Grille

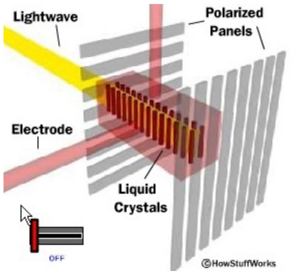




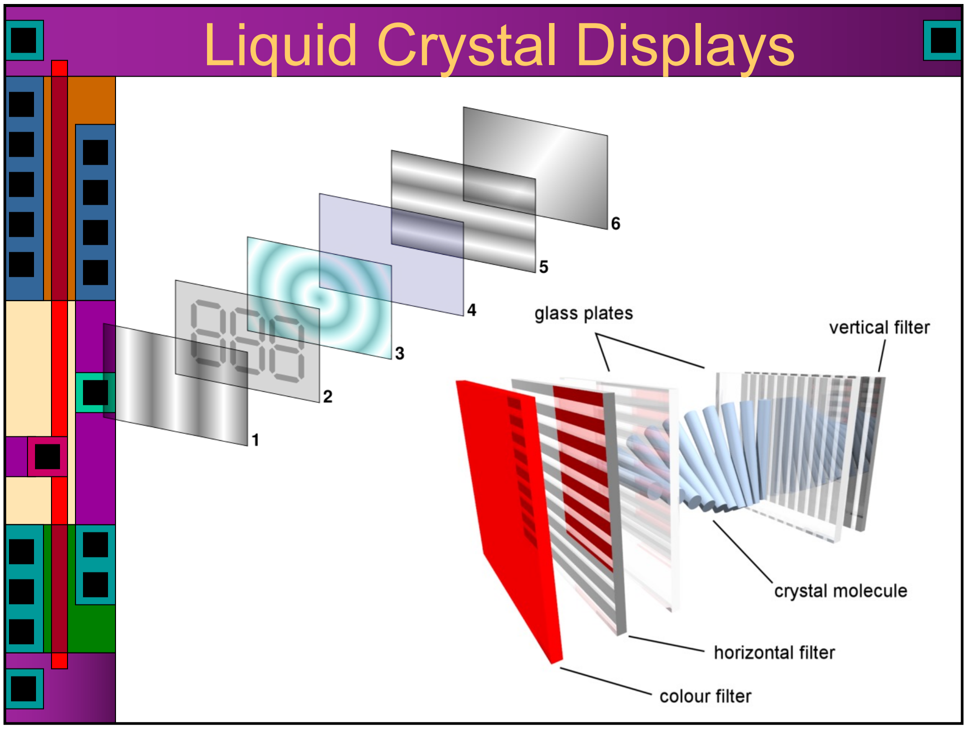
9

Liquid Crystal Displays



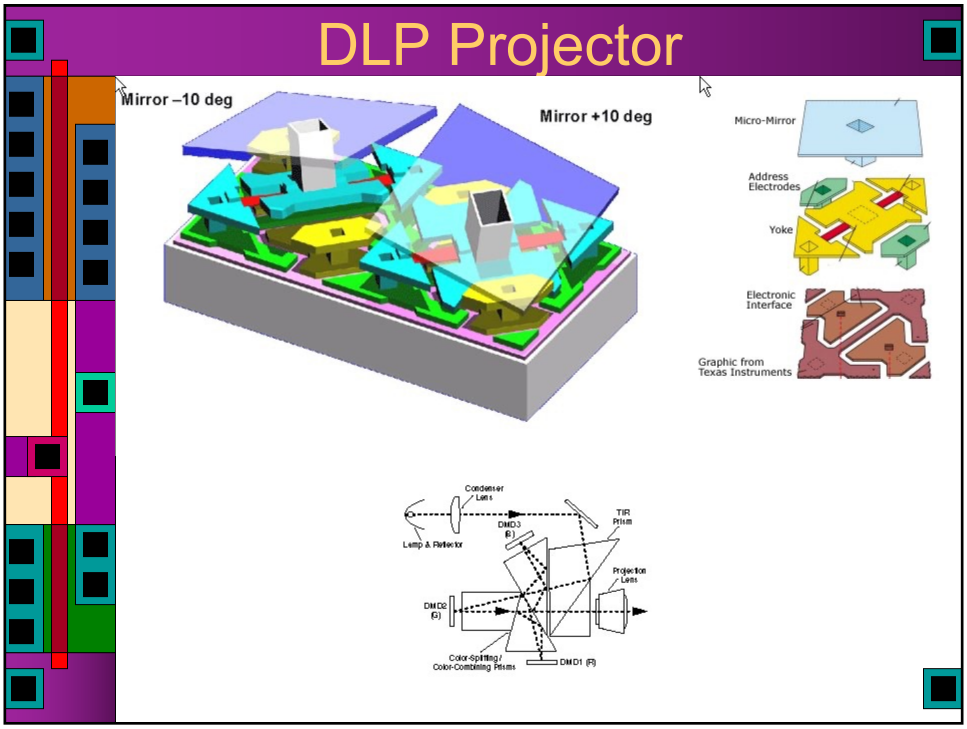


10



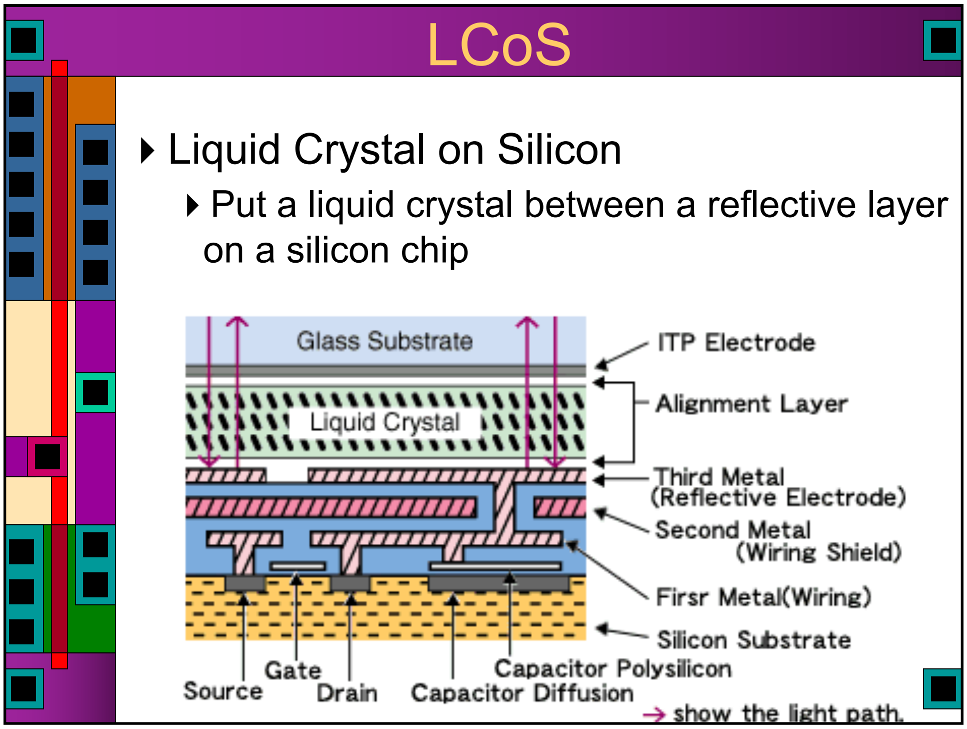
Liquid Crystal Displays

11



DLP Projector

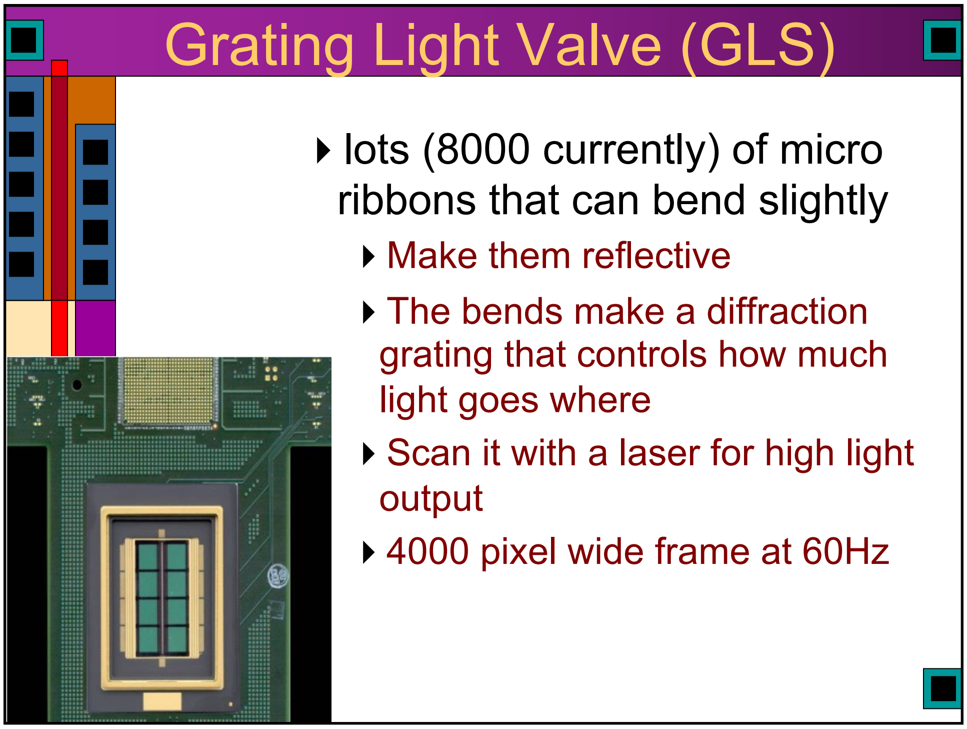
12



LCoS

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

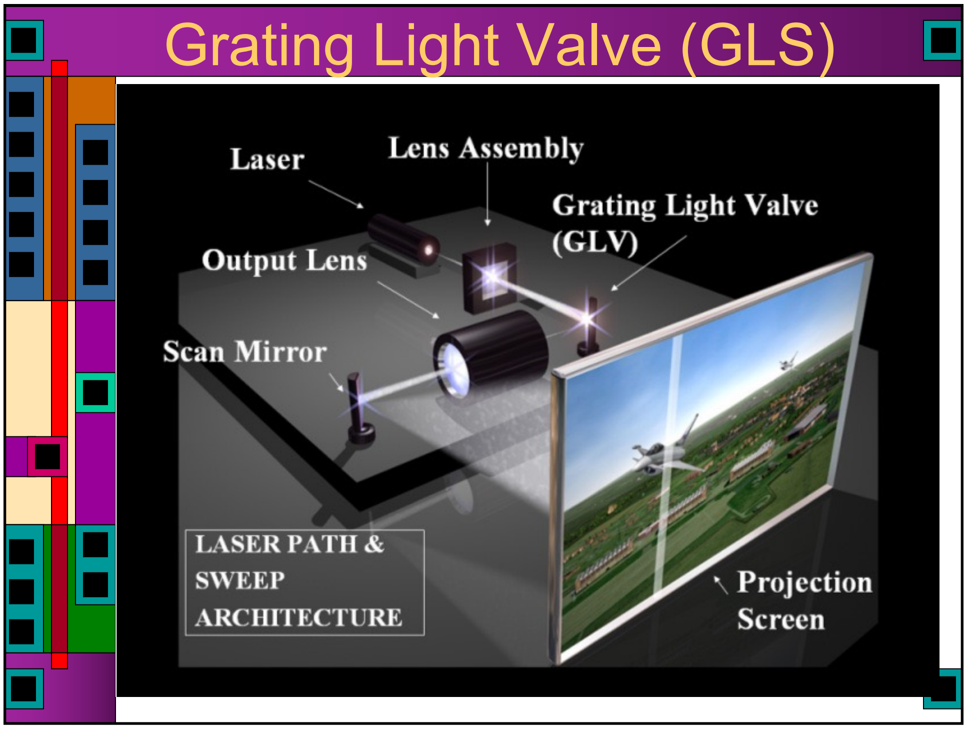
13



Grating Light Valve (GLS)

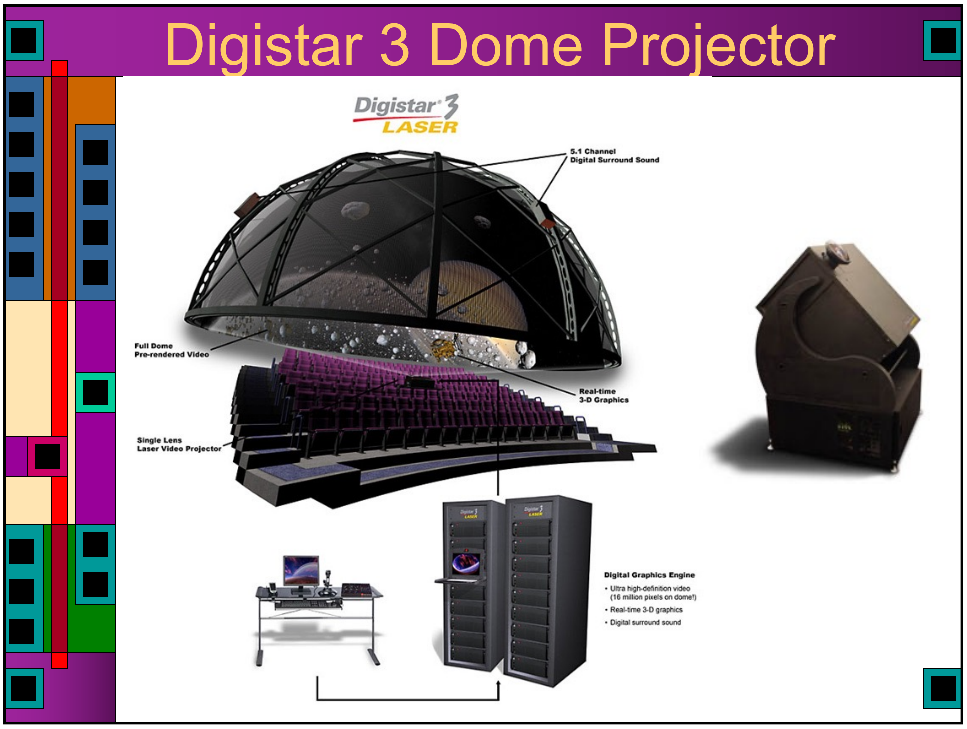
- lots (8000 currently) of micro ribbons that can bend slightly
  - Make them reflective
  - The bends make a diffraction grating that controls how much light goes where
  - Scan it with a laser for high light output
  - 4000 pixel wide frame at 60Hz

14



Grating Light Valve (GLS)

15



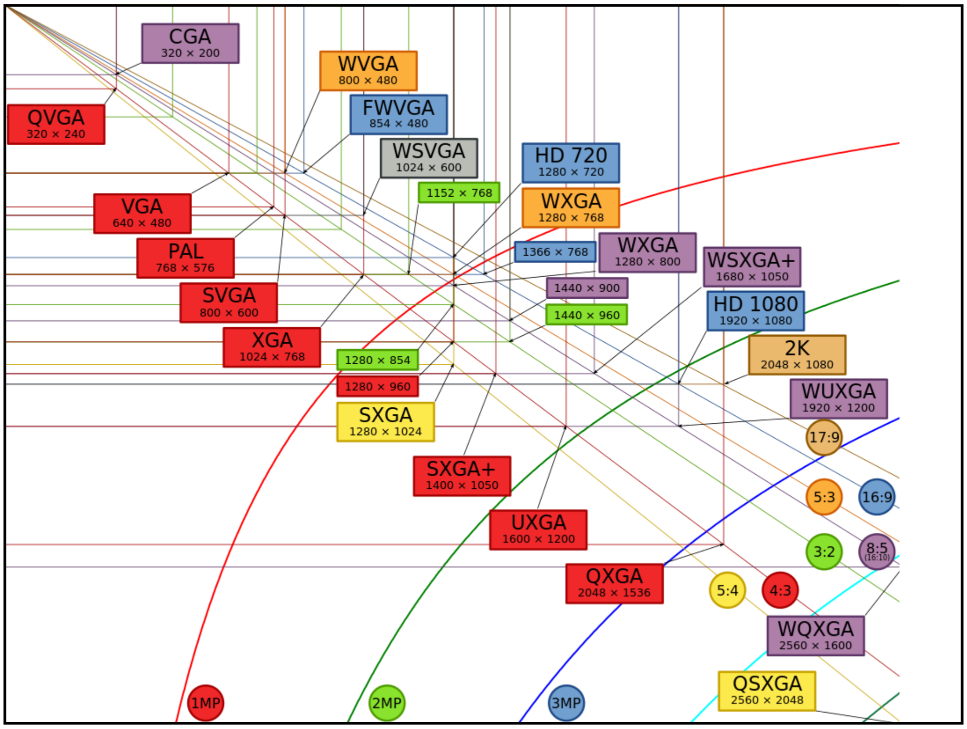
Digistar 3 Dome Projector

16

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

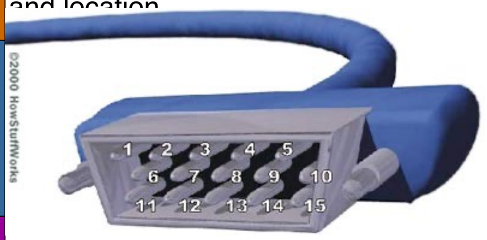
17

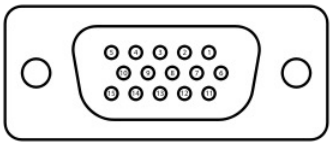


18

VGA Connector

and location

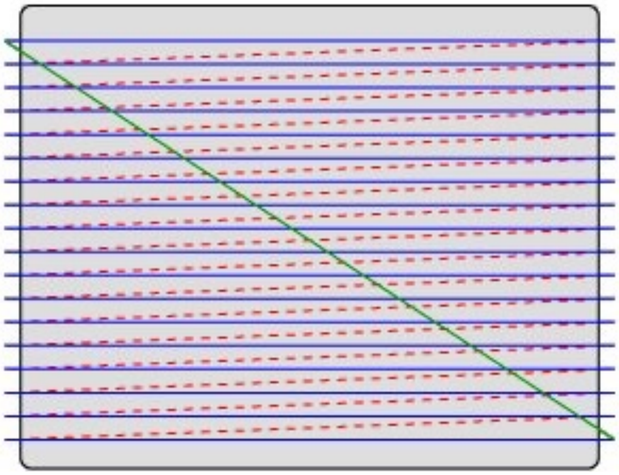




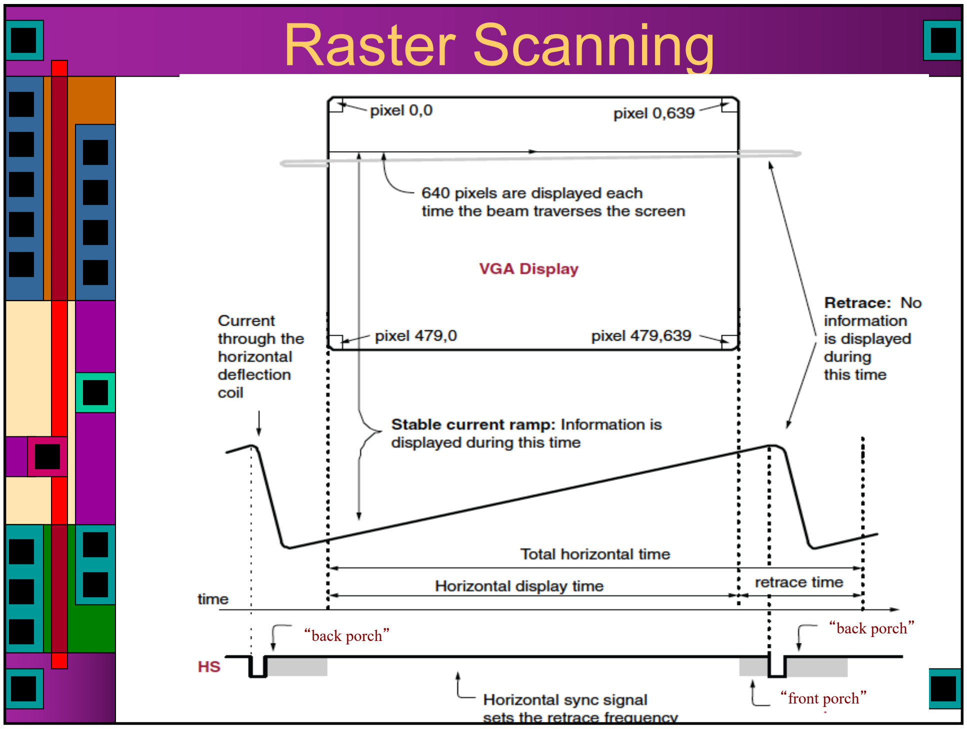
1: Red out	**6**: Red return (ground)	**11**: Monitor ID 0 in
2: Green out	**7**: Green return (ground)	**12**: Monitor ID 1 in or data from display
3: Blue out	**8**: Blue return (ground)	**13**: Horizontal Sync
4: Unused	**9**: Unused	**14**: Vertical Sync
5: Ground	**10**: Sync return (ground)	**15**: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

19

Raster Scanning



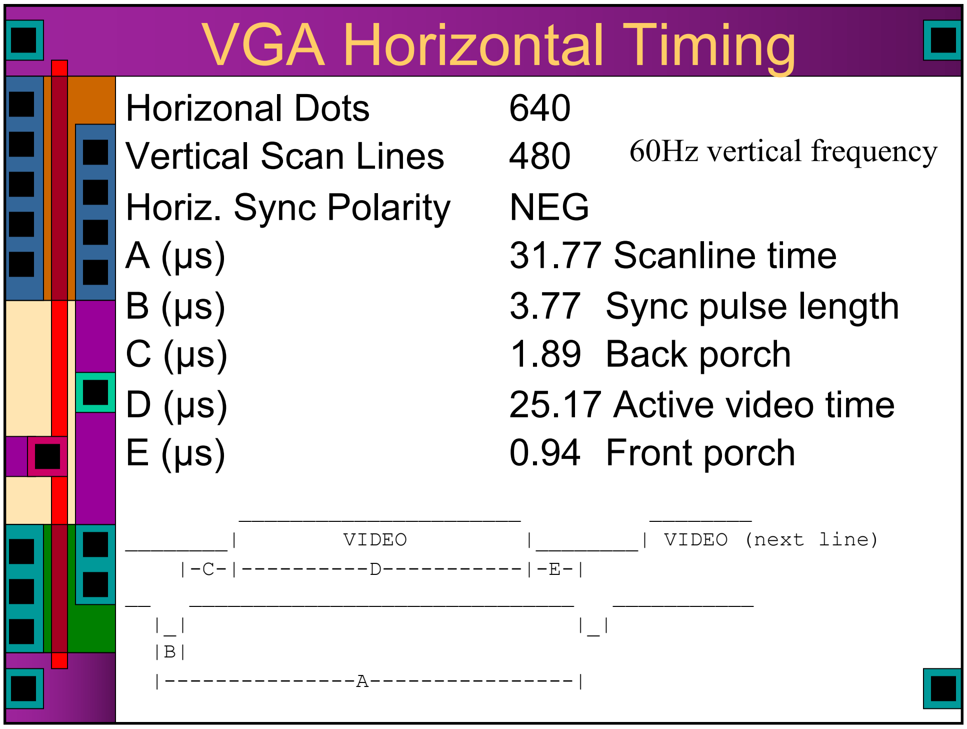
20



Raster Scanning

pixel 0,0
pixel 0,639
640 pixels are displayed each time the beam traverses the screen
VGA Display
Current through the horizontal deflection coil
pixel 479,0
pixel 479,639
Retrace: No information is displayed during this time
Stable current ramp: Information is displayed during this time
Total horizontal time
Horizontal display time
retrace time
time
"back porch"
"back porch"
HS
Horizontal sync signal sets the retrace frequency
"front porch"

21

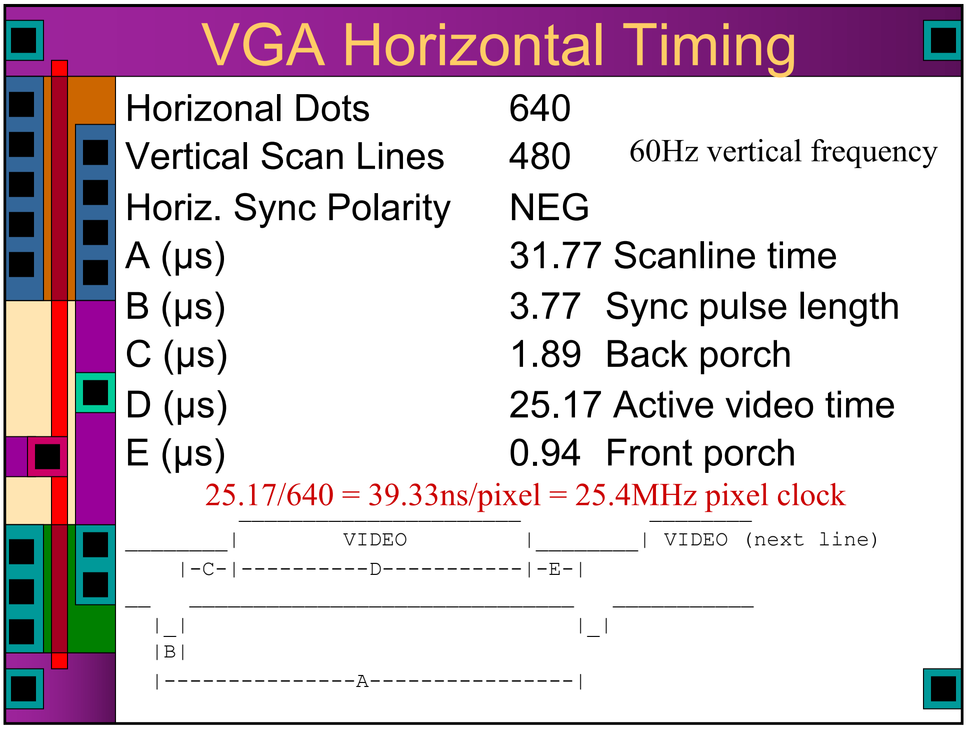


VGA Horizontal Timing

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----------------|
```

22

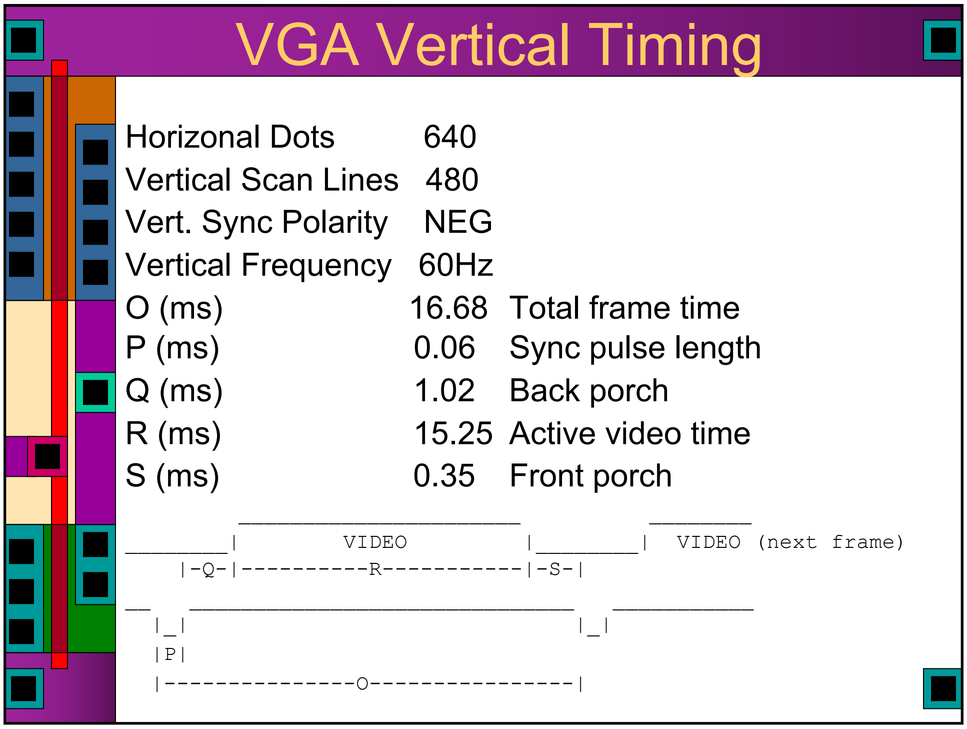


VGA Horizontal Timing

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

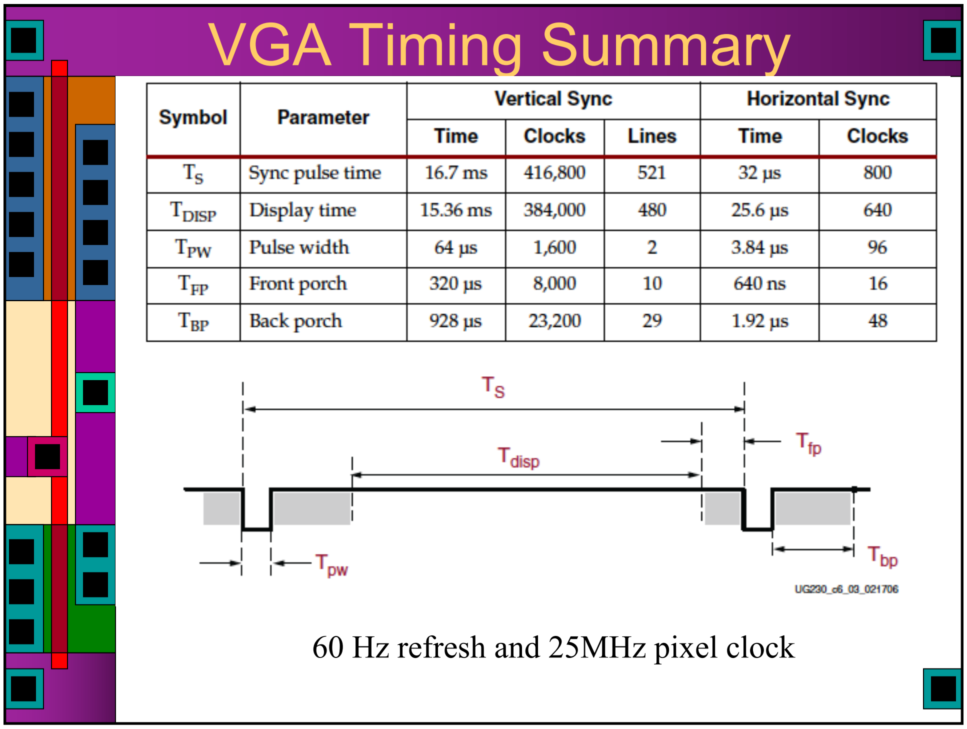
```
         ______________________          ________
________|        VIDEO         |________| VIDEO (next line)
    |-C-|----------D-----------|-E-|
__   ______________________________   ___________
  |_|                              |_|
  |B|
  |---------------A----------------|
```



VGA Vertical 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----------------|
```

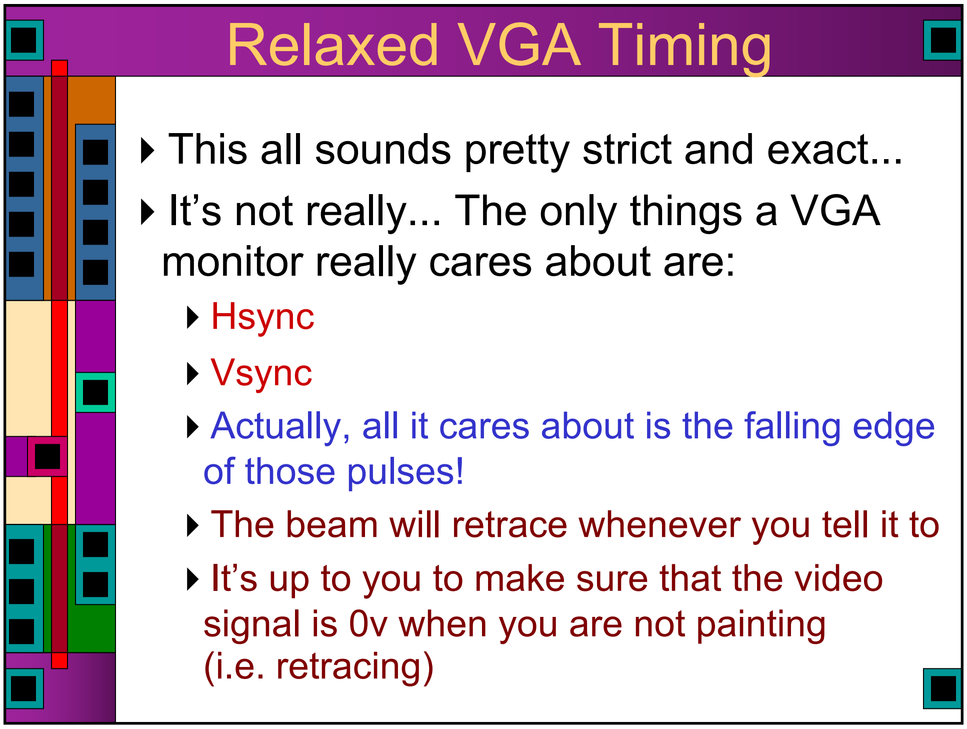


VGA Timing Summary

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

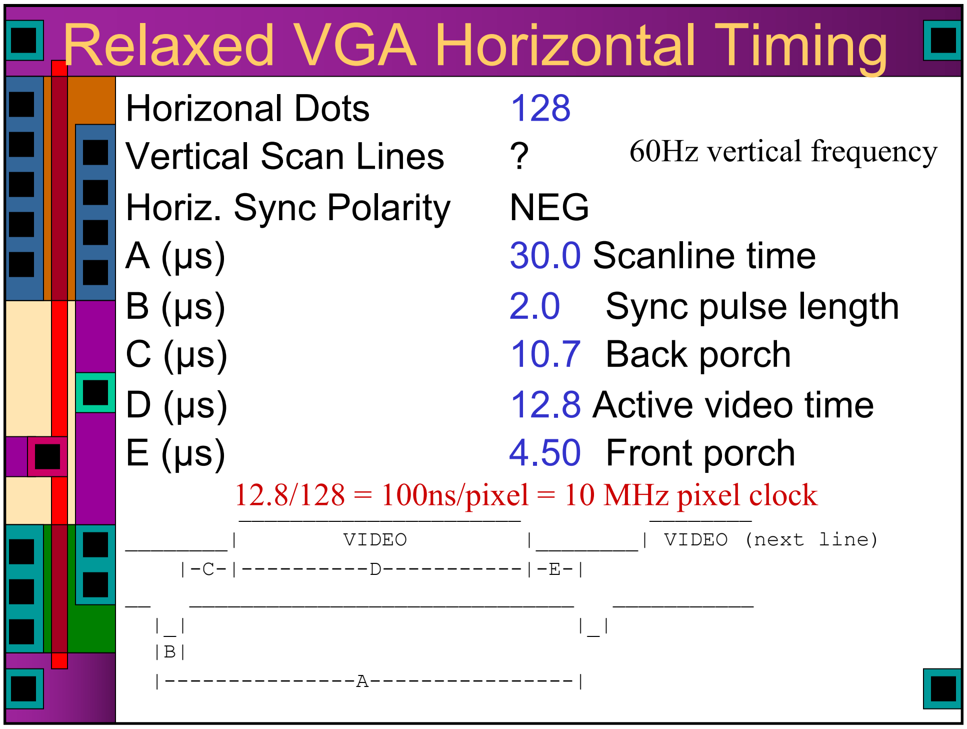
25



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)

26



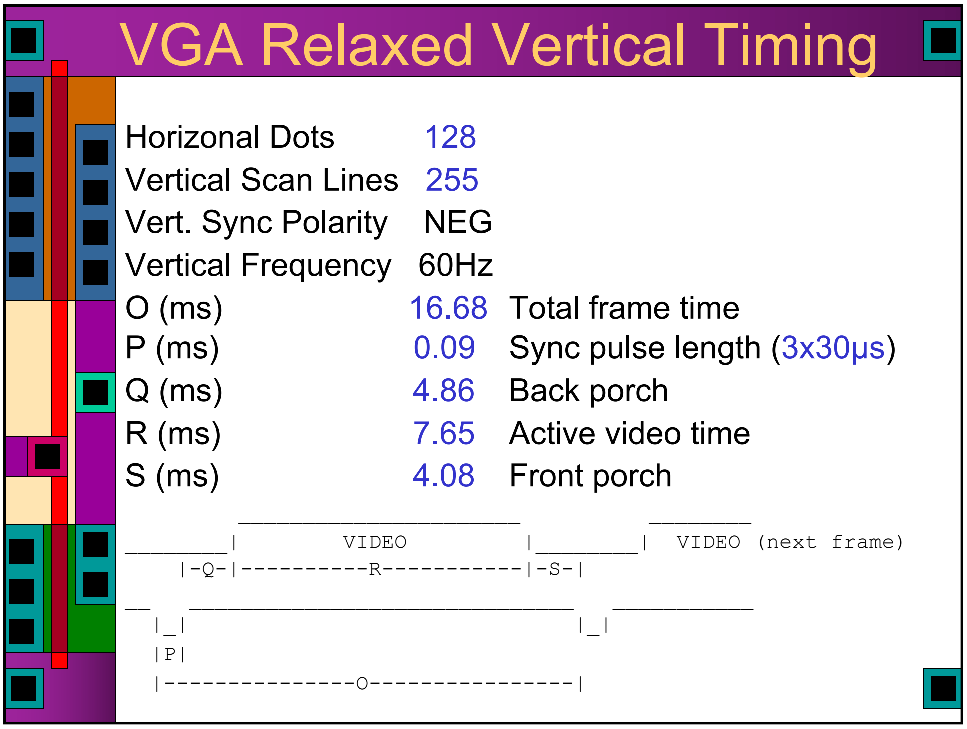
Relaxed VGA Horizontal Timing

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----------------|
```

27

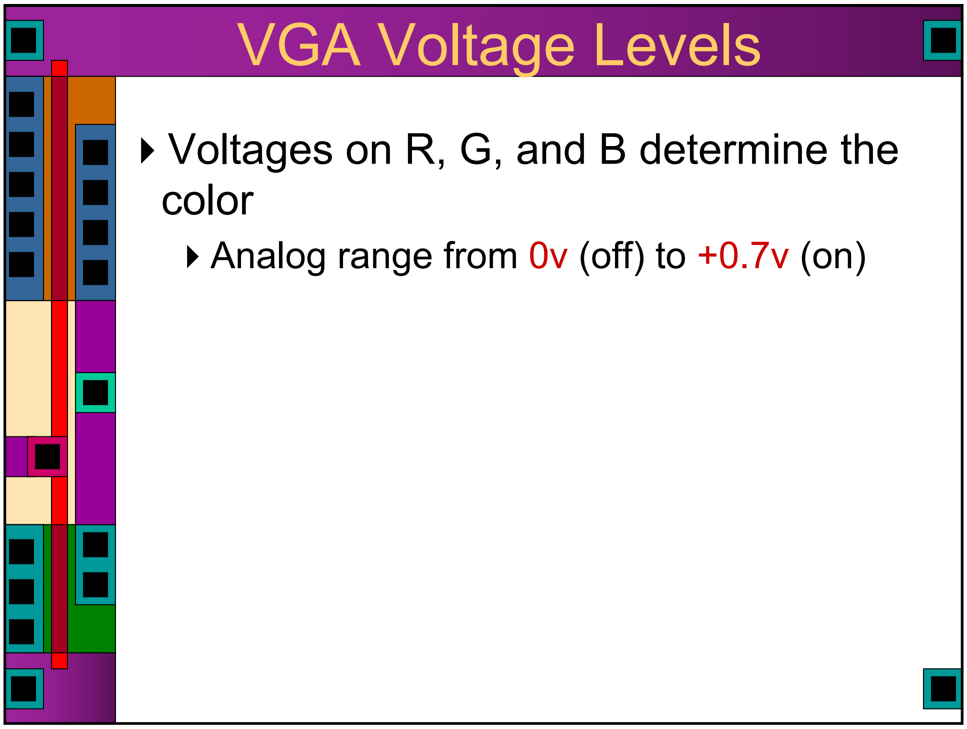


VGA Relaxed Vertical Timing

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----------------|
```

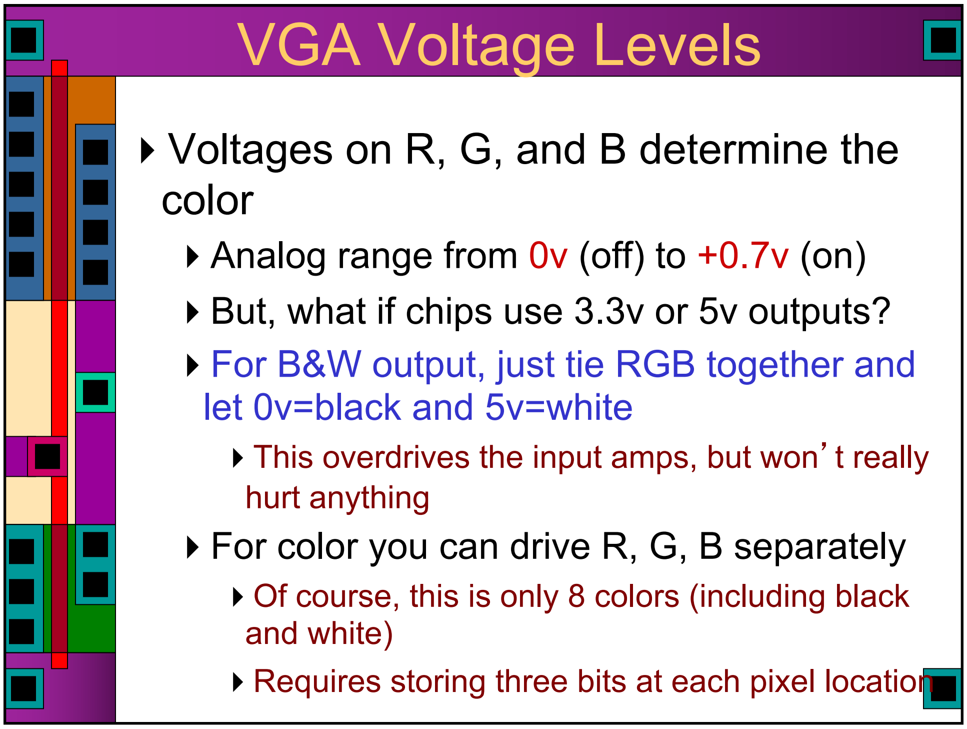
28



VGA Voltage Levels

- Voltages on R, G, and B determine the color
  - Analog range from 0v (off) to +0.7v (on)

29

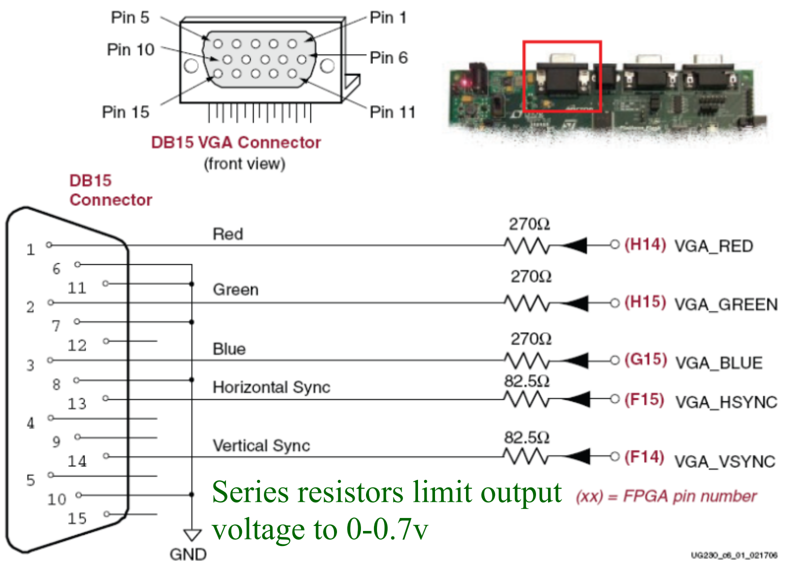


VGA Voltage Levels

- Voltages on R, G, and B determine the color
  - Analog range from 0v (off) to +0.7v (on)
  - But, what if chips use 3.3v or 5v outputs?
  - For B&W output, just tie RGB together and let 0v=black and 5v=white
    - This 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

30

VGA on Spartan3e Starter



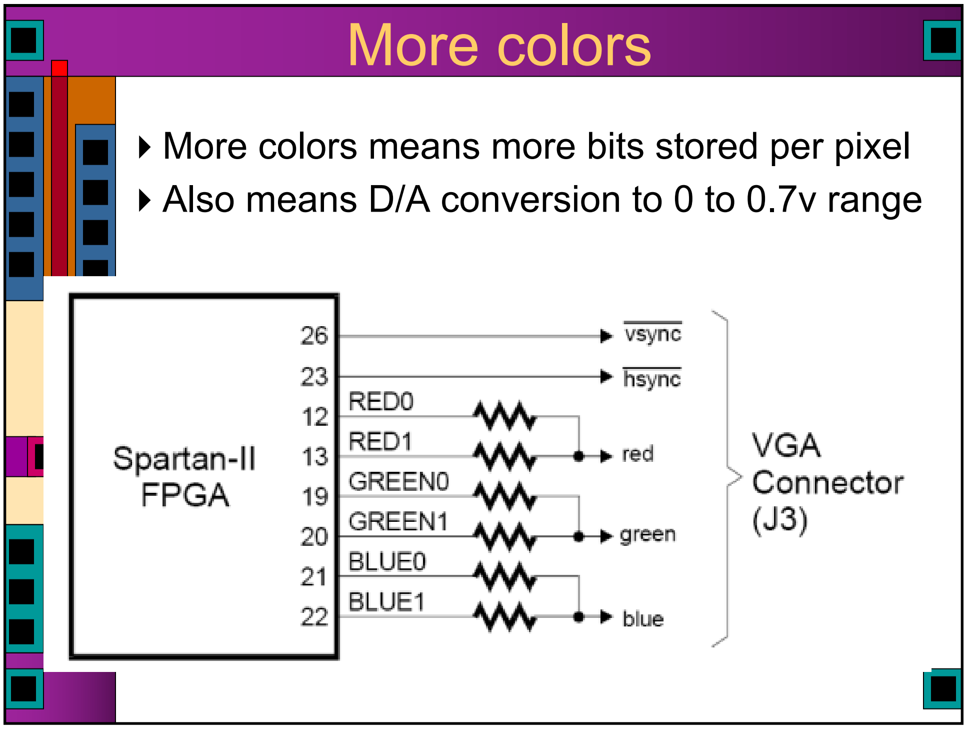
Series resistors limit output voltage to 0-0.7v

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

VGA on Spartan3e Starter

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

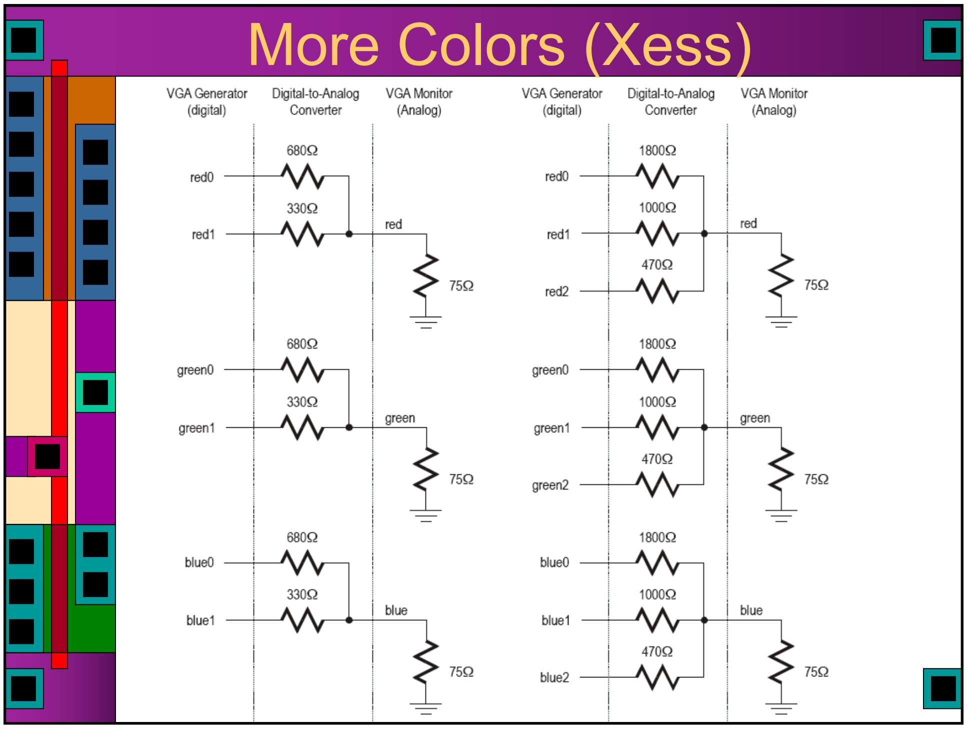
VGA_RED	VGA_GREEN	VGA_BLUE	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White



More colors

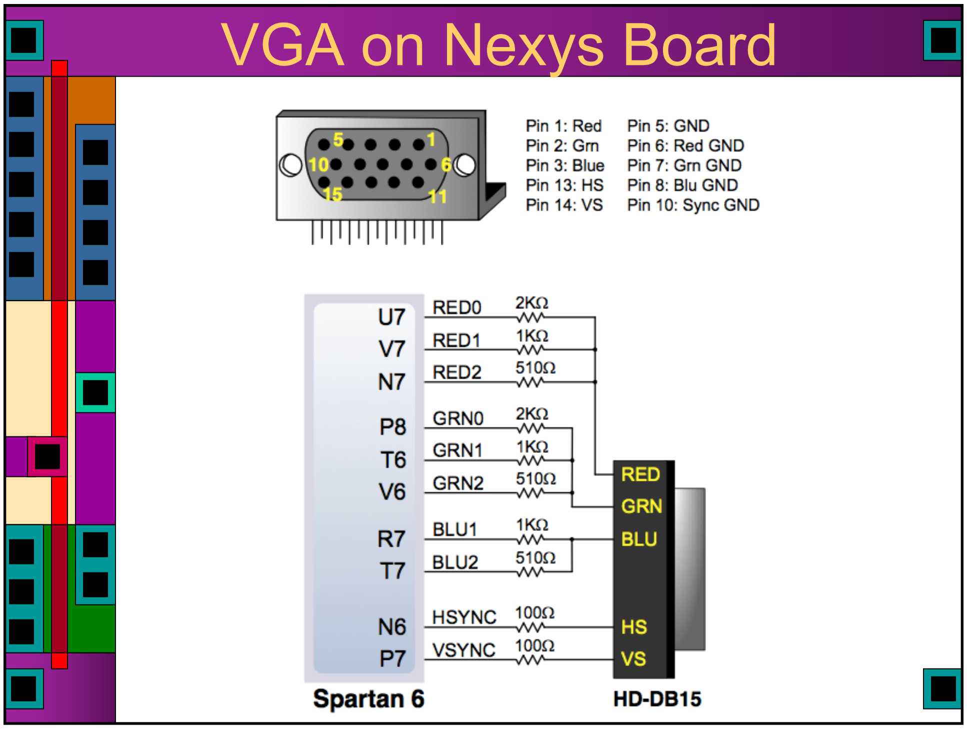
- More colors means more bits stored per pixel
- Also means D/A conversion to 0 to 0.7v range

33



More Colors (Xess)

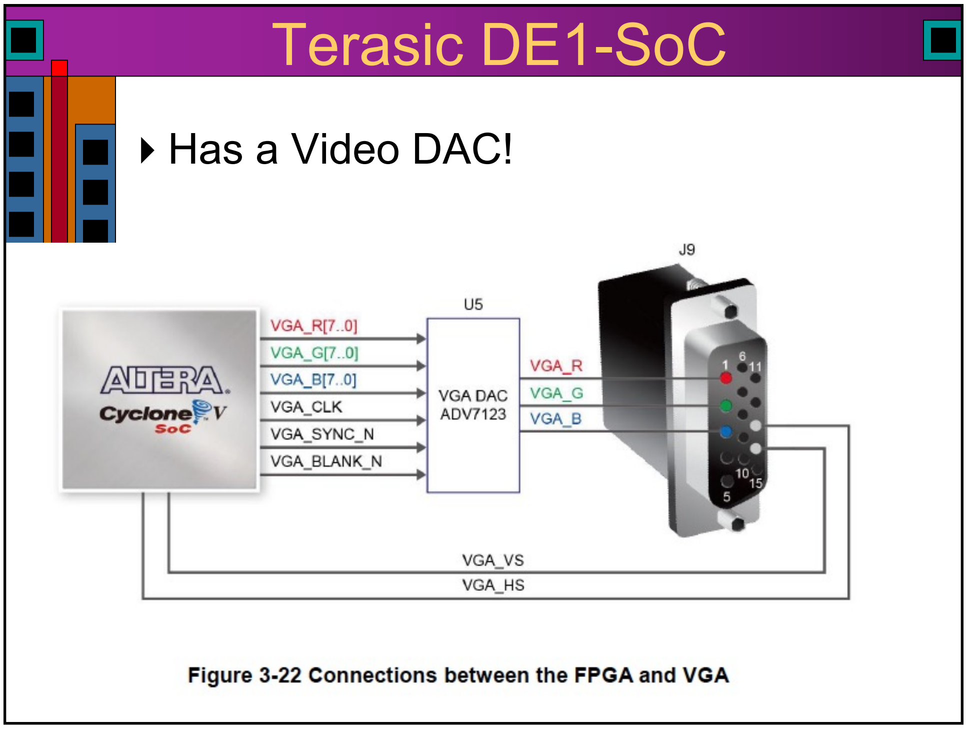
34



VGA on Nexys Board

Pin 1: Red	Pin 5: GND
Pin 2: Grn	Pin 6: Red GND
Pin 3: Blue	Pin 7: Grn GND
Pin 13: HS	Pin 8: Blu GND
Pin 14: VS	Pin 10: Sync GND

35



Terasic DE1-SoC

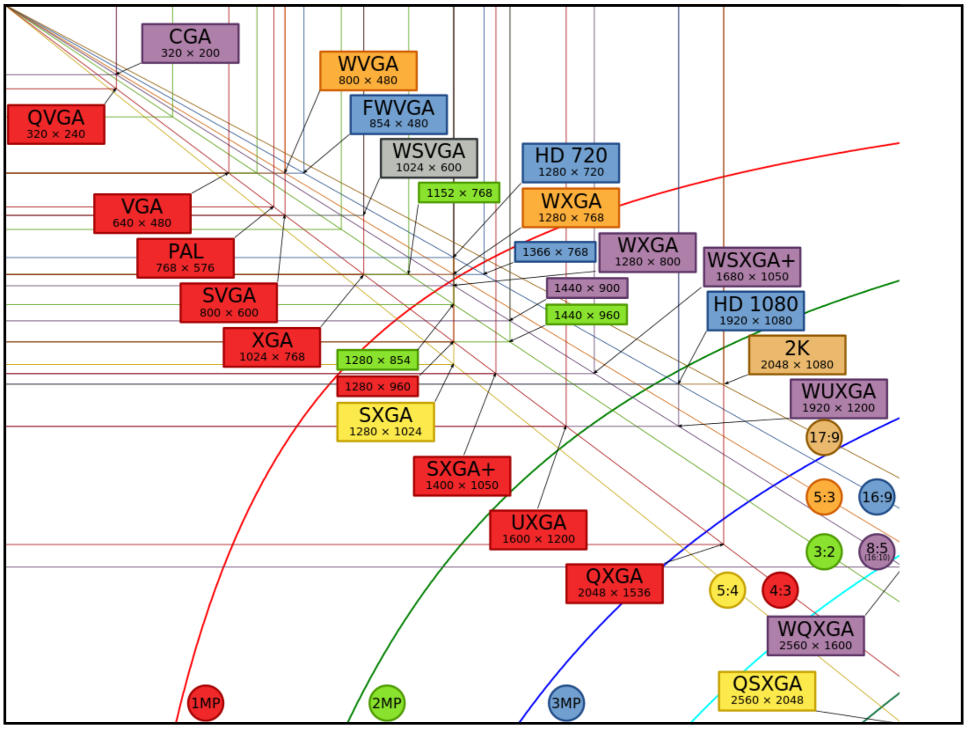
- Has a Video DAC!

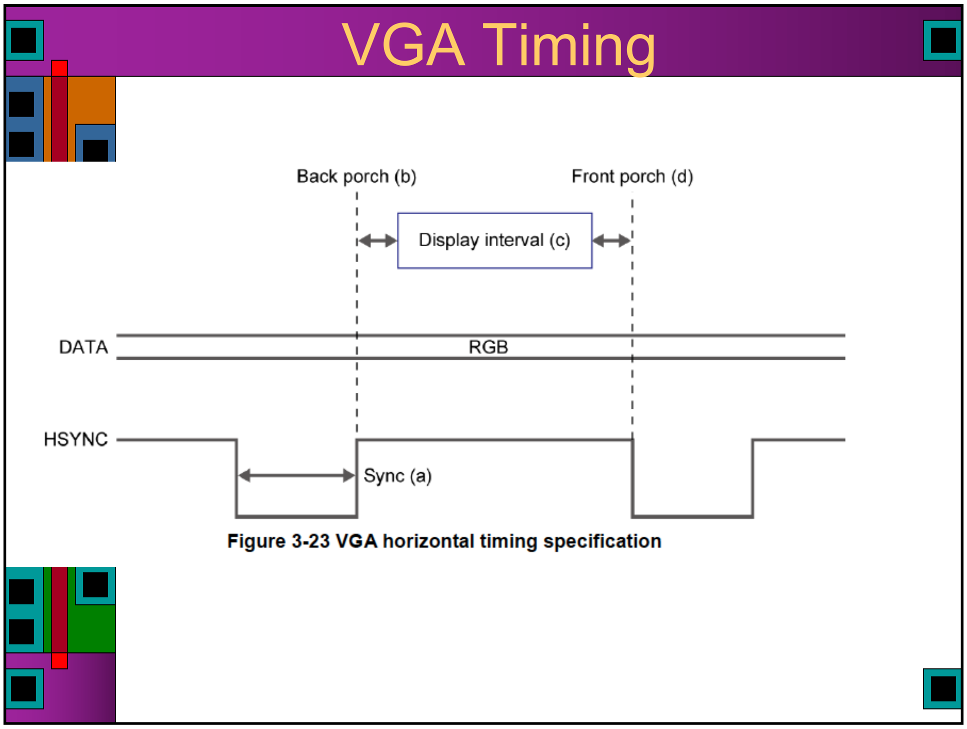
Figure 3-22 Connections between the FPGA and VGA

36

DE1-SoC

- Can support 1280x1024 (SXGA)
- Signals transmitted at 100MHz
  - A huge improvement over past year's boards…
  - But it means you can end up spending a lot more effort on display output…

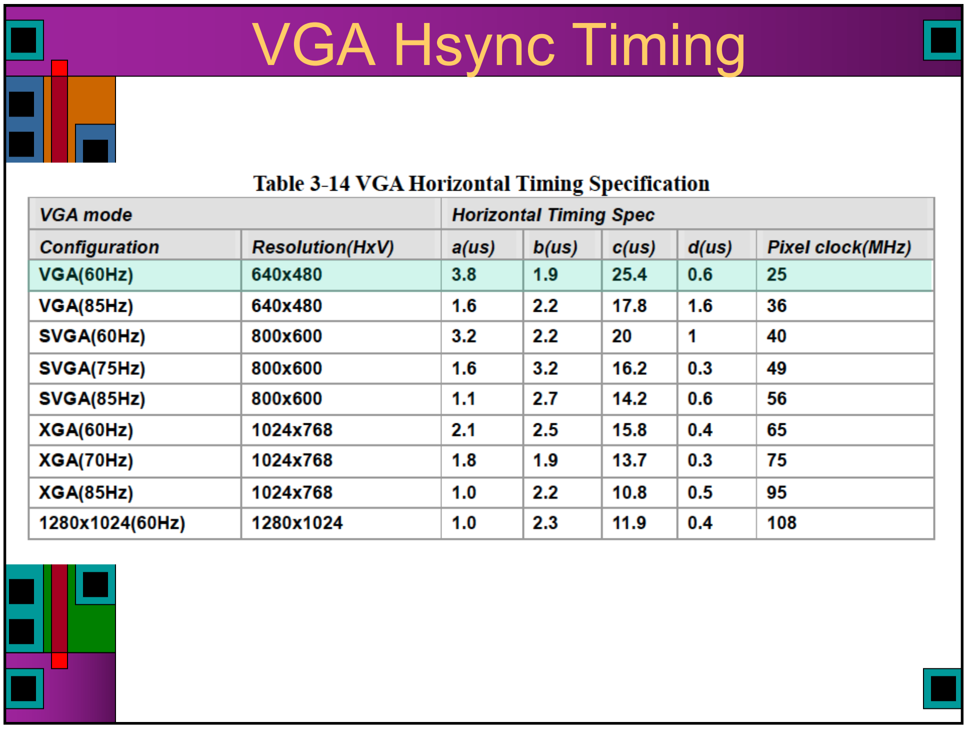




VGA Timing

Figure 3-23 VGA horizontal timing specification

39



VGA Hsync Timing

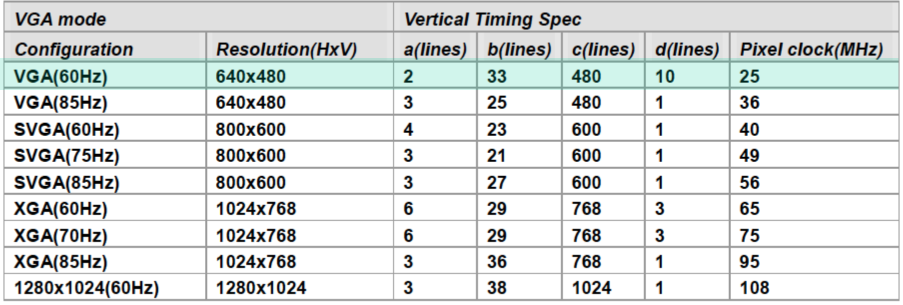
Table 3-14 VGA Horizontal Timing Specification

VGA mode		*Horizontal Timing Spec*				
Configuration	*Resolution(HxV)*	*a(us)*	*b(us)*	*c(us)*	*d(us)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	3.8	1.9	25.4	0.6	25
VGA(85Hz)	640x480	1.6	2.2	17.8	1.6	36
SVGA(60Hz)	800x600	3.2	2.2	20	1	40
SVGA(75Hz)	800x600	1.6	3.2	16.2	0.3	49
SVGA(85Hz)	800x600	1.1	2.7	14.2	0.6	56
XGA(60Hz)	1024x768	2.1	2.5	15.8	0.4	65
XGA(70Hz)	1024x768	1.8	1.9	13.7	0.3	75
XGA(85Hz)	1024x768	1.0	2.2	10.8	0.5	95
1280x1024(60Hz)	1280x1024	1.0	2.3	11.9	0.4	108

40

VGA Vsync Timing

Table 3-15 VGA Vertical Timing Specification

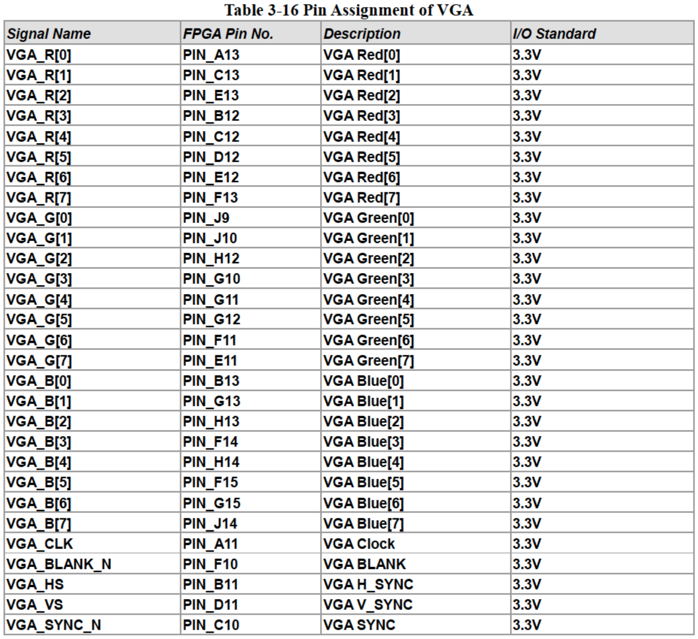


VGA mode		*Vertical Timing Spec*				
Configuration	*Resolution(HxV)*	*a(lines)*	*b(lines)*	*c(lines)*	*d(lines)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	2	33	480	10	25
VGA(85Hz)	640x480	3	25	480	1	36
SVGA(60Hz)	800x600	4	23	600	1	40
SVGA(75Hz)	800x600	3	21	600	1	49
SVGA(85Hz)	800x600	3	27	600	1	56
XGA(60Hz)	1024x768	6	29	768	3	65
XGA(70Hz)	1024x768	6	29	768	3	75
XGA(85Hz)	1024x768	3	36	768	1	95
1280x1024(60Hz)	1280x1024	3	38	1024	1	108

41

Pin Assignment

Table 3-16 Pin Assignment of VGA



Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
VGA_R[0]	PIN_A13	VGA Red[0]	3.3V
VGA_R[1]	PIN_C13	VGA Red[1]	3.3V
VGA_R[2]	PIN_E13	VGA Red[2]	3.3V
VGA_R[3]	PIN_B12	VGA Red[3]	3.3V
VGA_R[4]	PIN_C12	VGA Red[4]	3.3V
VGA_R[5]	PIN_D12	VGA Red[5]	3.3V
VGA_R[6]	PIN_E12	VGA Red[6]	3.3V
VGA_R[7]	PIN_F13	VGA Red[7]	3.3V
VGA_G[0]	PIN_J9	VGA Green[0]	3.3V
VGA_G[1]	PIN_J10	VGA Green[1]	3.3V
VGA_G[2]	PIN_H12	VGA Green[2]	3.3V
VGA_G[3]	PIN_G10	VGA Green[3]	3.3V
VGA_G[4]	PIN_G11	VGA Green[4]	3.3V
VGA_G[5]	PIN_G12	VGA Green[5]	3.3V
VGA_G[6]	PIN_F11	VGA Green[6]	3.3V
VGA_G[7]	PIN_E11	VGA Green[7]	3.3V
VGA_B[0]	PIN_B13	VGA Blue[0]	3.3V
VGA_B[1]	PIN_G13	VGA Blue[1]	3.3V
VGA_B[2]	PIN_H13	VGA Blue[2]	3.3V
VGA_B[3]	PIN_F14	VGA Blue[3]	3.3V
VGA_B[4]	PIN_H14	VGA Blue[4]	3.3V
VGA_B[5]	PIN_F15	VGA Blue[5]	3.3V
VGA_B[6]	PIN_G15	VGA Blue[6]	3.3V
VGA_B[7]	PIN_J14	VGA Blue[7]	3.3V
VGA_CLK	PIN_A11	VGA Clock	3.3V
VGA_BLANK_N	PIN_F10	VGA BLANK	3.3V
VGA_HS	PIN_B11	VGA H_SYNC	3.3V
VGA_VS	PIN_D11	VGA V_SYNC	3.3V
VGA_SYNC_N	PIN_C10	VGA SYNC	3.3V

42

Terasic DE1-SoC

- Video DAC ADV7123

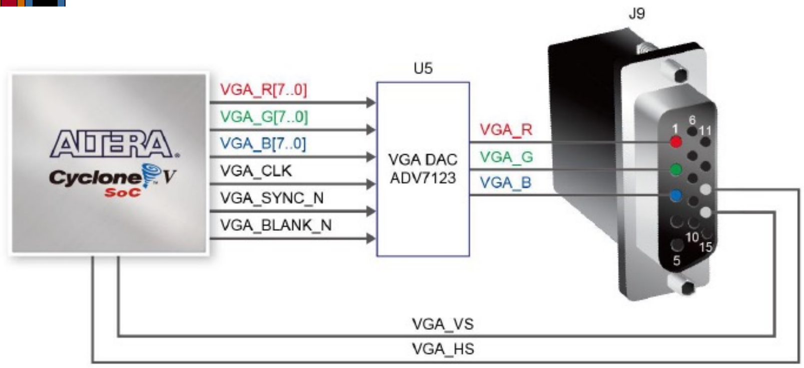


Figure 3-22 Connections between the FPGA and VGA

ADV7123 video DAC

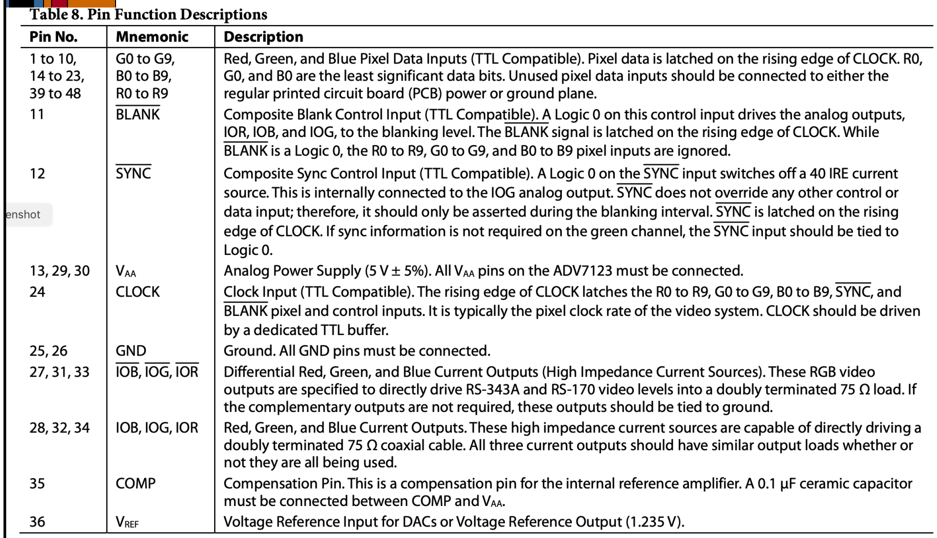


Table 8. Pin Function Descriptions

Pin No.	Mnemonic	Description
1 to 10, 14 to 23, 39 to 48	G0 to G9, B0 to B9, R0 to R9	Red, Green, and Blue Pixel Data Inputs (TTL Compatible). Pixel data is latched on the rising edge of CLOCK. R0, G0, and B0 are the least significant data bits. Unused pixel data inputs should be connected to either the regular printed circuit board (PCB) power or ground plane.
11	$\overline{\text{BLANK}}$	Composite Blank Control Input (TTL Compatible). A Logic 0 on this control input drives the analog outputs, IOR, IOB, and IOG, to the blanking level. The $\overline{\text{BLANK}}$ signal is latched on the rising edge of CLOCK. While $\overline{\text{BLANK}}$ is a Logic 0, the R0 to R9, G0 to G9, and B0 to B9 pixel inputs are ignored.
12	$\overline{\text{SYNC}}$	Composite Sync Control Input (TTL Compatible). A Logic 0 on the $\overline{\text{SYNC}}$ input switches off a 40 IRE current source. This is internally connected to the IOG analog output. $\overline{\text{SYNC}}$ does not override any other control or data input; therefore, it should only be asserted during the blanking interval. $\overline{\text{SYNC}}$ is latched on the rising edge of CLOCK. If sync information is not required on the green channel, the $\overline{\text{SYNC}}$ input should be tied to Logic 0.
13, 29, 30	V_{AA}	Analog Power Supply (5 V ± 5%). All V_{AA} pins on the ADV7123 must be connected.
24	CLOCK	Clock Input (TTL Compatible). The rising edge of CLOCK latches the R0 to R9, G0 to G9, B0 to B9, $\overline{\text{SYNC}}$, and $\overline{\text{BLANK}}$ pixel and control inputs. It is typically the pixel clock rate of the video system. CLOCK should be driven by a dedicated TTL buffer.
25, 26	GND	Ground. All GND pins must be connected.
27, 31, 33	$\overline{\text{IOB}}$, $\overline{\text{IOG}}$, $\overline{\text{IOR}}$	Differential Red, Green, and Blue Current Outputs (High Impedance Current Sources). These RGB video outputs are specified to directly drive RS-343A and RS-170 video levels into a doubly terminated 75 Ω load. If the complementary outputs are not required, these outputs should be tied to ground.
28, 32, 34	IOB, IOG, IOR	Red, Green, and Blue Current Outputs. These high impedance current sources are capable of directly driving a doubly terminated 75 Ω coaxial cable. All three current outputs should have similar output loads whether or not they are all being used.
35	COMP	Compensation Pin. This is a compensation pin for the internal reference amplifier. A 0.1 µF ceramic capacitor must be connected between COMP and V_{AA}.
36	V_{REF}	Voltage Reference Input for DACs or Voltage Reference Output (1.235 V).

ADV7123 video DAC

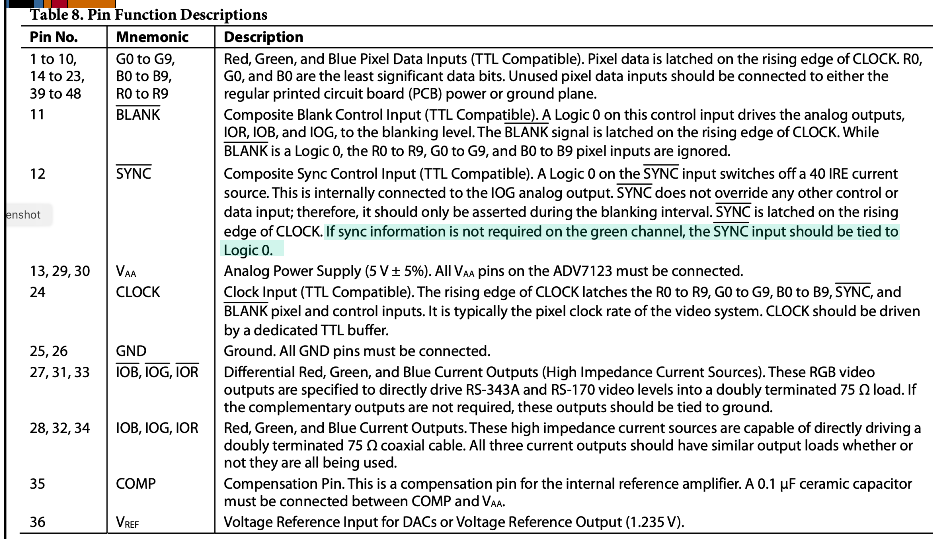
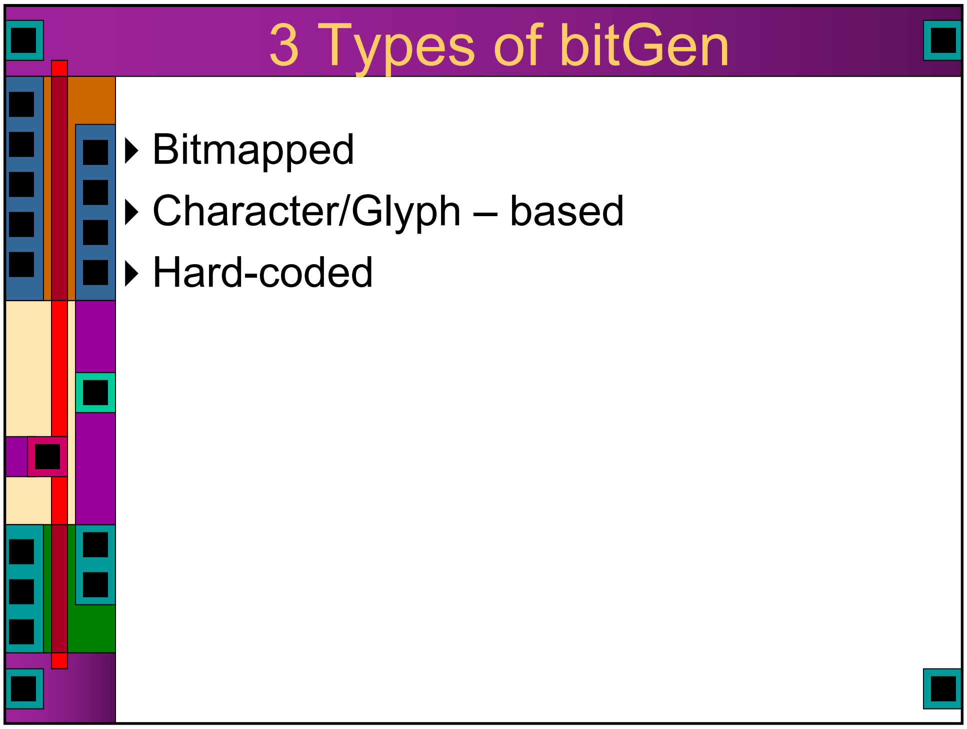


Table 8. Pin Function Descriptions

Pin No.	Mnemonic	Description
1 to 10, 14 to 23, 39 to 48	G0 to G9, B0 to B9, R0 to R9	Red, Green, and Blue Pixel Data Inputs (TTL Compatible). Pixel data is latched on the rising edge of CLOCK. R0, G0, and B0 are the least significant data bits. Unused pixel data inputs should be connected to either the regular printed circuit board (PCB) power or ground plane.
11	$\overline{\text{BLANK}}$	Composite Blank Control Input (TTL Compatible). A Logic 0 on this control input drives the analog outputs, IOR, IOB, and IOG, to the blanking level. The $\overline{\text{BLANK}}$ signal is latched on the rising edge of CLOCK. While $\overline{\text{BLANK}}$ is a Logic 0, the R0 to R9, G0 to G9, and B0 to B9 pixel inputs are ignored.
12	$\overline{\text{SYNC}}$	Composite Sync Control Input (TTL Compatible). A Logic 0 on the $\overline{\text{SYNC}}$ input switches off a 40 IRE current source. This is internally connected to the IOG analog output. $\overline{\text{SYNC}}$ does not override any other control or data input; therefore, it should only be asserted during the blanking interval. $\overline{\text{SYNC}}$ is latched on the rising edge of CLOCK. If sync information is not required on the green channel, the $\overline{\text{SYNC}}$ input should be tied to Logic 0.
13, 29, 30	V_{AA}	Analog Power Supply (5 V ± 5%). All V_{AA} pins on the ADV7123 must be connected.
24	CLOCK	Clock Input (TTL Compatible). The rising edge of CLOCK latches the R0 to R9, G0 to G9, B0 to B9, $\overline{\text{SYNC}}$, and $\overline{\text{BLANK}}$ pixel and control inputs. It is typically the pixel clock rate of the video system. CLOCK should be driven by a dedicated TTL buffer.
25, 26	GND	Ground. All GND pins must be connected.
27, 31, 33	$\overline{\text{IOB}}$, $\overline{\text{IOG}}$, $\overline{\text{IOR}}$	Differential Red, Green, and Blue Current Outputs (High Impedance Current Sources). These RGB video outputs are specified to directly drive RS-343A and RS-170 video levels into a doubly terminated 75 Ω load. If the complementary outputs are not required, these outputs should be tied to ground.
28, 32, 34	IOB, IOG, IOR	Red, Green, and Blue Current Outputs. These high impedance current sources are capable of directly driving a doubly terminated 75 Ω coaxial cable. All three current outputs should have similar output loads whether or not they are all being used.
35	COMP	Compensation Pin. This is a compensation pin for the internal reference amplifier. A 0.1 µF ceramic capacitor must be connected between COMP and V_{AA}.
36	V_{REF}	Voltage Reference Input for DACs or Voltage Reference Output (1.235 V).

VGA Breakdown

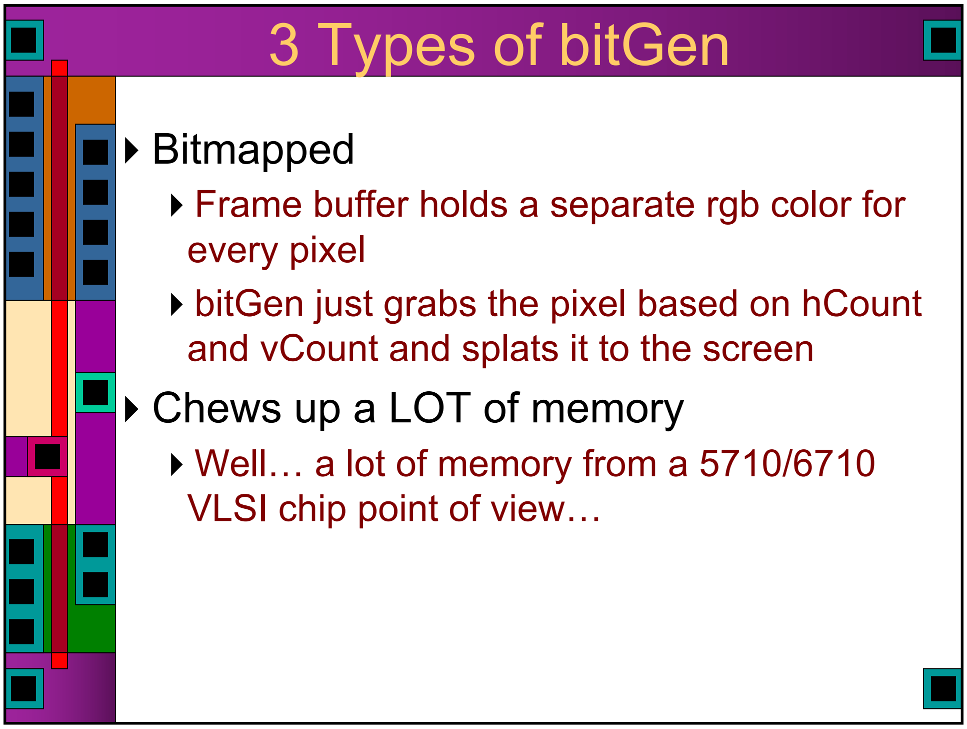
- vgaControl
  - Generate timing pulses at the right time
  - hSync, vSync, bright, hCount, vCount
- bitGen
  - Based on bright, hCount, vCount, turn on the bits



3 Types of bitGen

- Bitmapped
- Character/Glyph – based
- Hard-coded

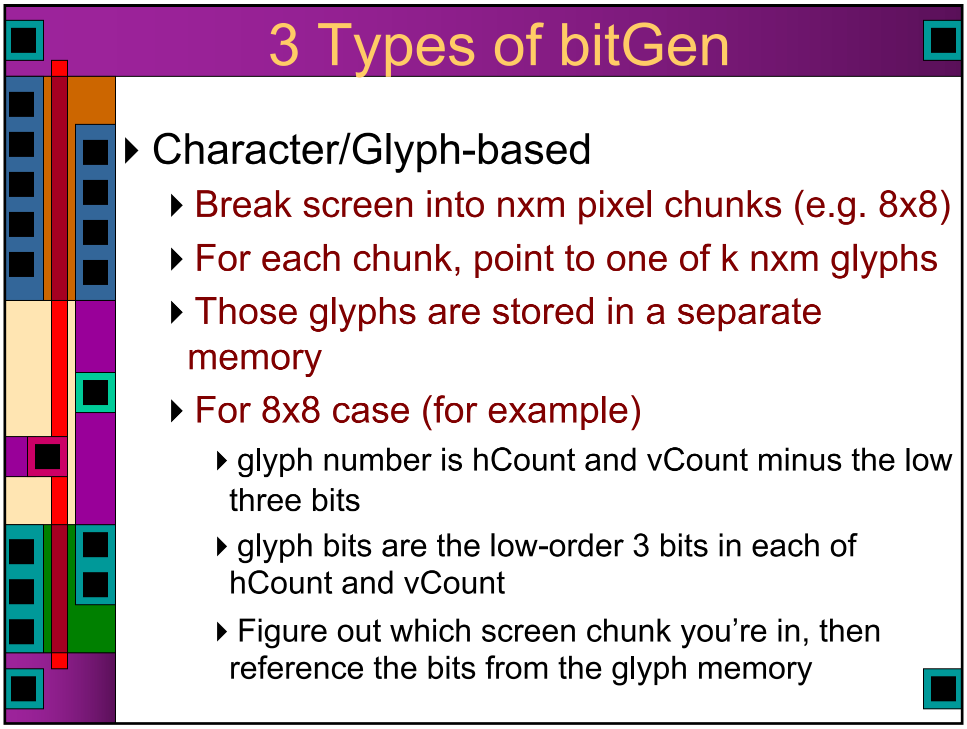
47



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
  - Well… a lot of memory from a 5710/6710 VLSI chip point of view…

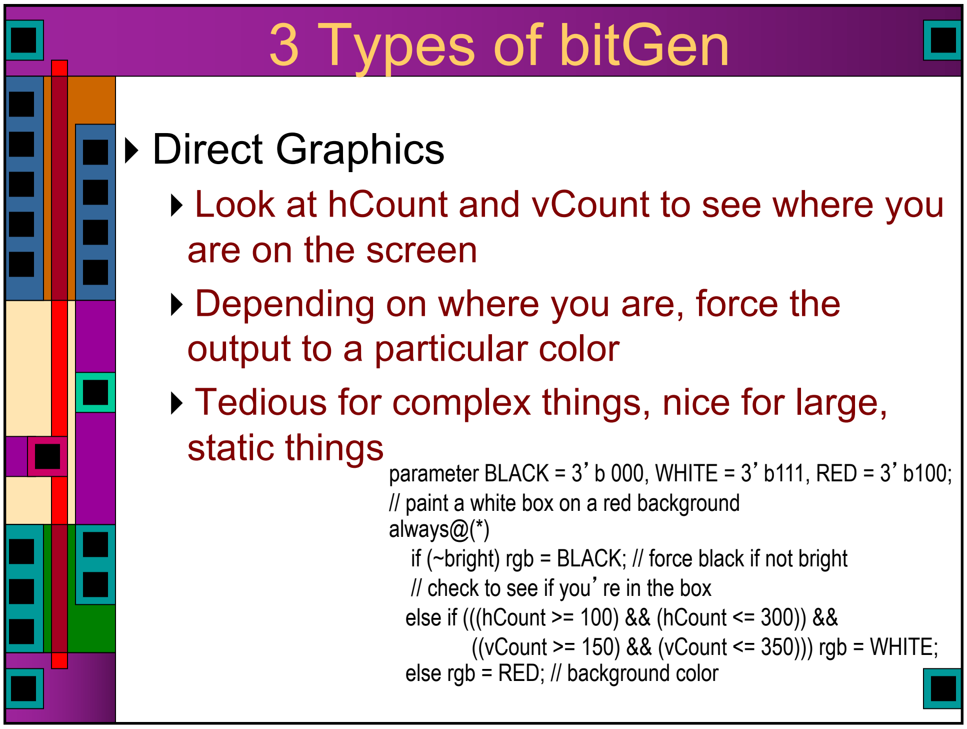
48



3 Types of bitGen

- Character/Glyph-based
  - Break screen into nxm pixel 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

49

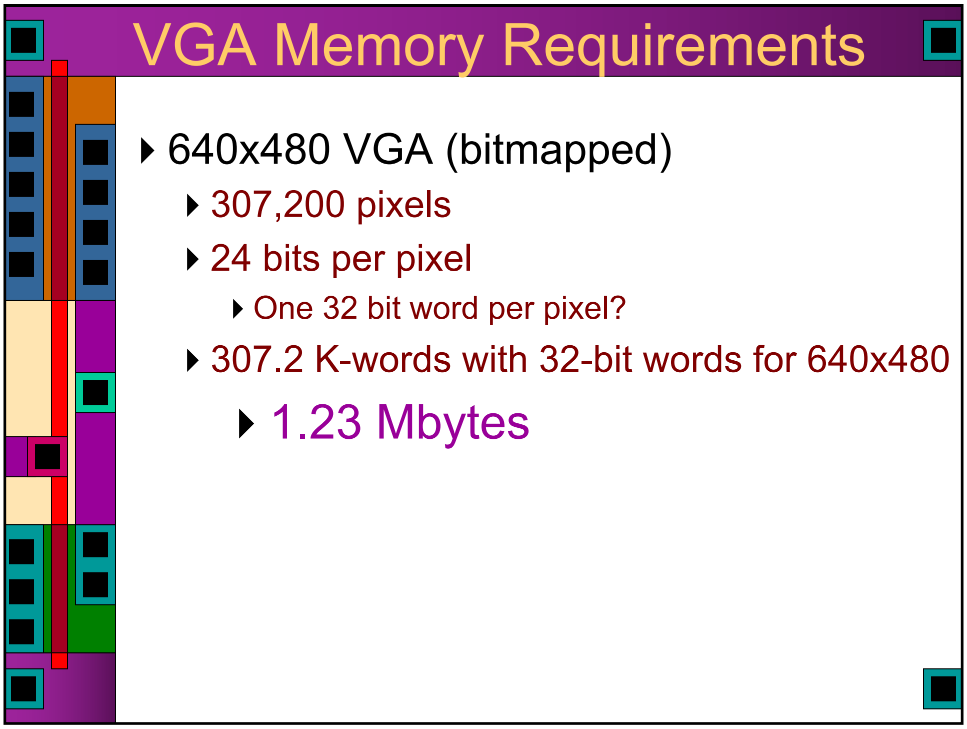


3 Types of bitGen

- 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
```

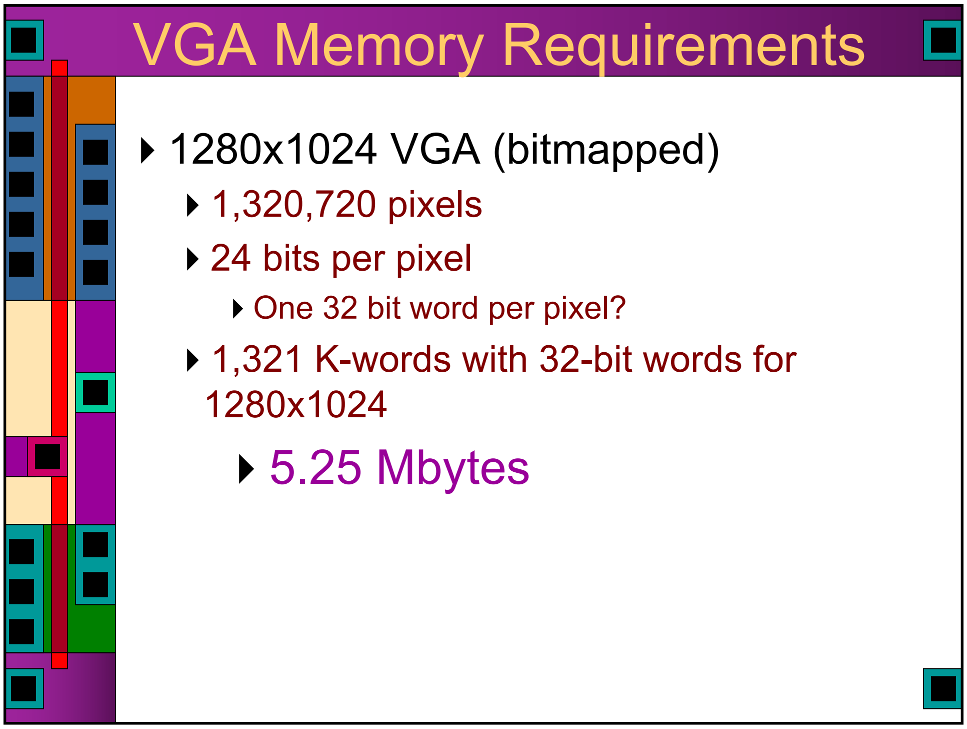
50



VGA Memory Requirements

- 640x480 VGA (bitmapped)
    - 307,200 pixels
    - 24 bits per pixel
        - One 32 bit word per pixel?
    - 307.2 K-words with 32-bit words for 640x480
        - 1.23 Mbytes

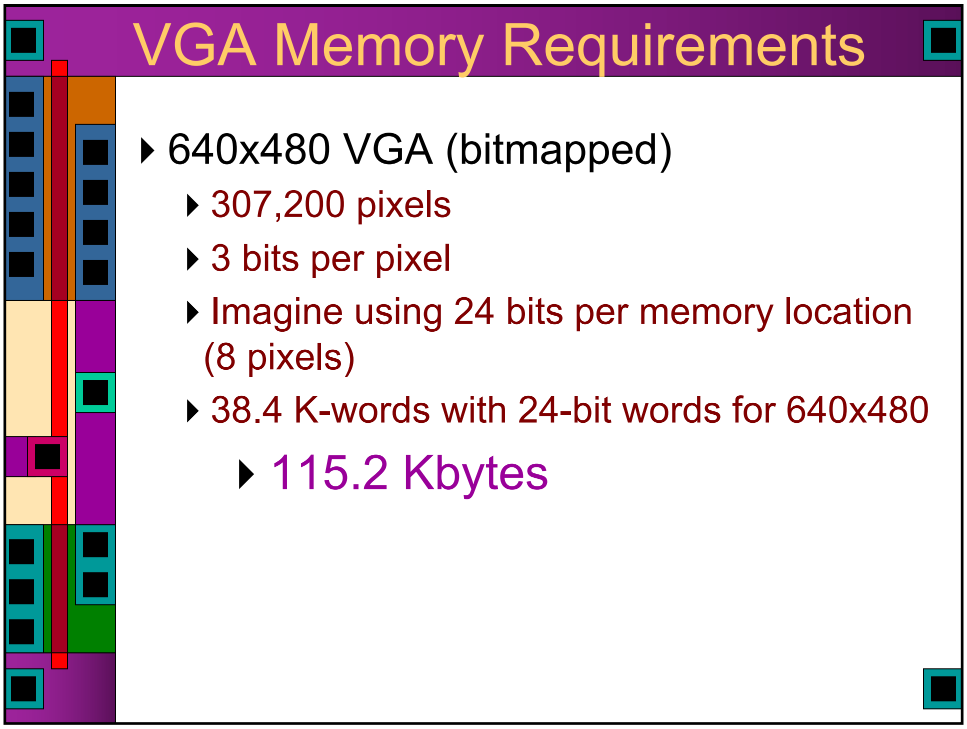
51



VGA Memory Requirements

- 1280x1024 VGA (bitmapped)
    - 1,320,720 pixels
    - 24 bits per pixel
        - One 32 bit word per pixel?
    - 1,321 K-words with 32-bit words for 1280x1024
        - 5.25 Mbytes

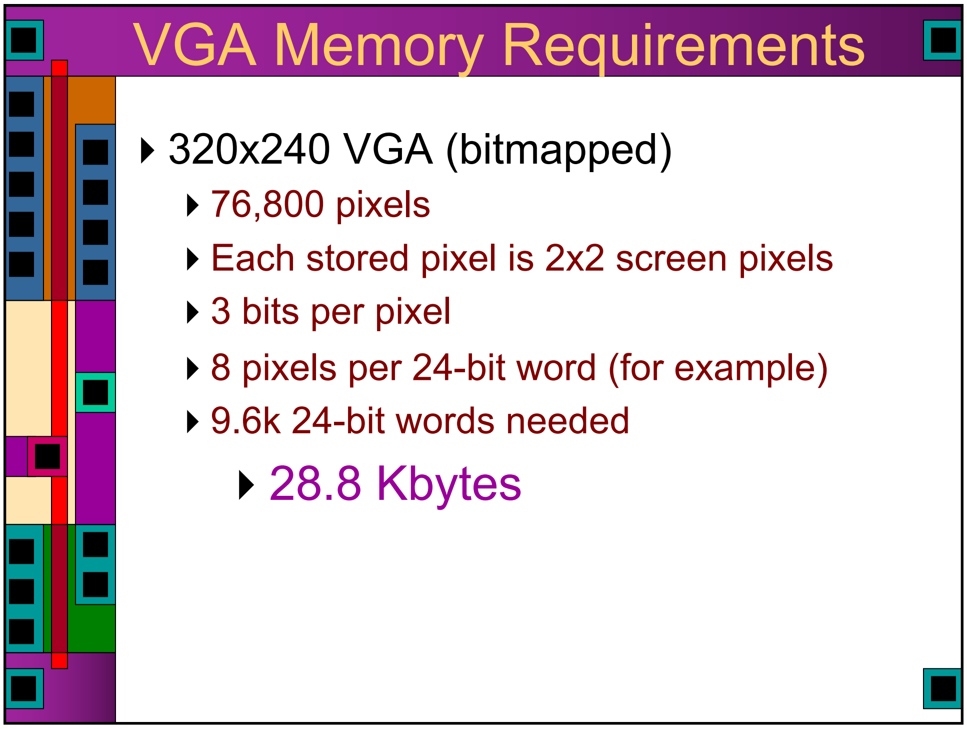
52



VGA Memory Requirements

- 640x480 VGA (bitmapped)
  - 307,200 pixels
  - 3 bits per pixel
  - Imagine using 24 bits per memory location (8 pixels)
  - 38.4 K-words with 24-bit words for 640x480
    - 115.2 Kbytes

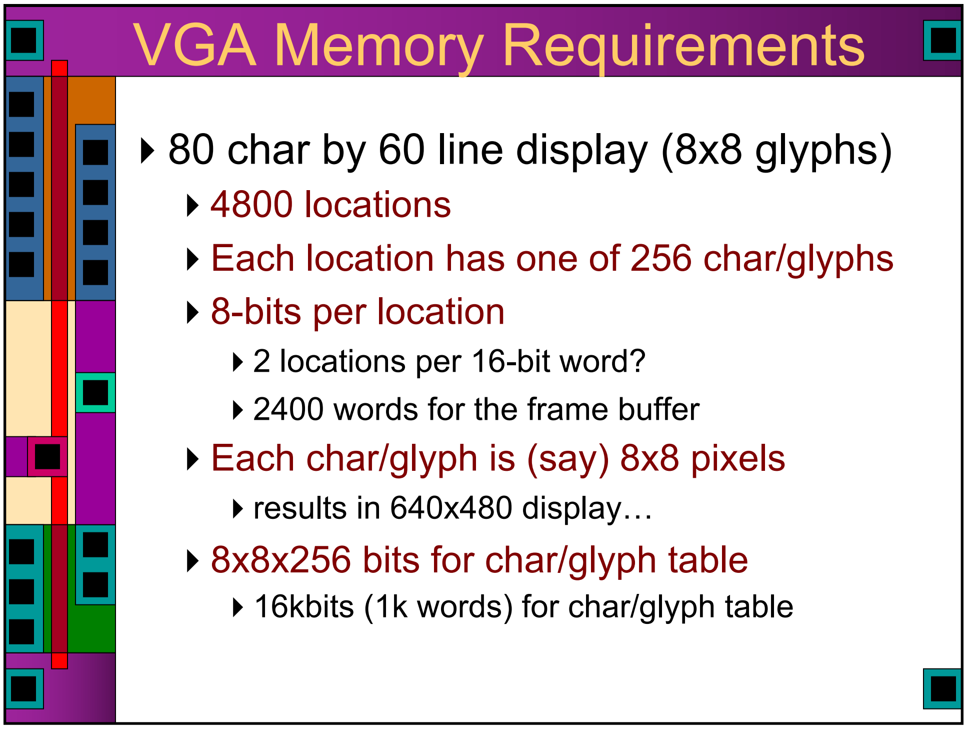
53



VGA Memory Requirements

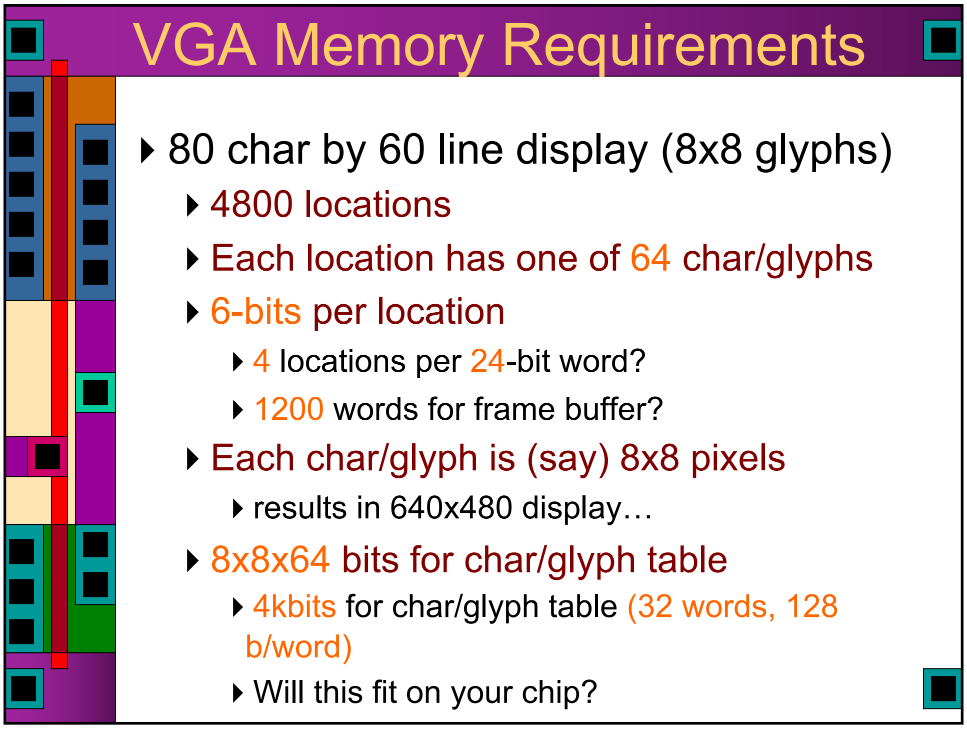
- 320x240 VGA (bitmapped)
  - 76,800 pixels
  - Each stored pixel is 2x2 screen pixels
  - 3 bits per pixel
  - 8 pixels per 24-bit word (for example)
  - 9.6k 24-bit words needed
    - 28.8 Kbytes

54



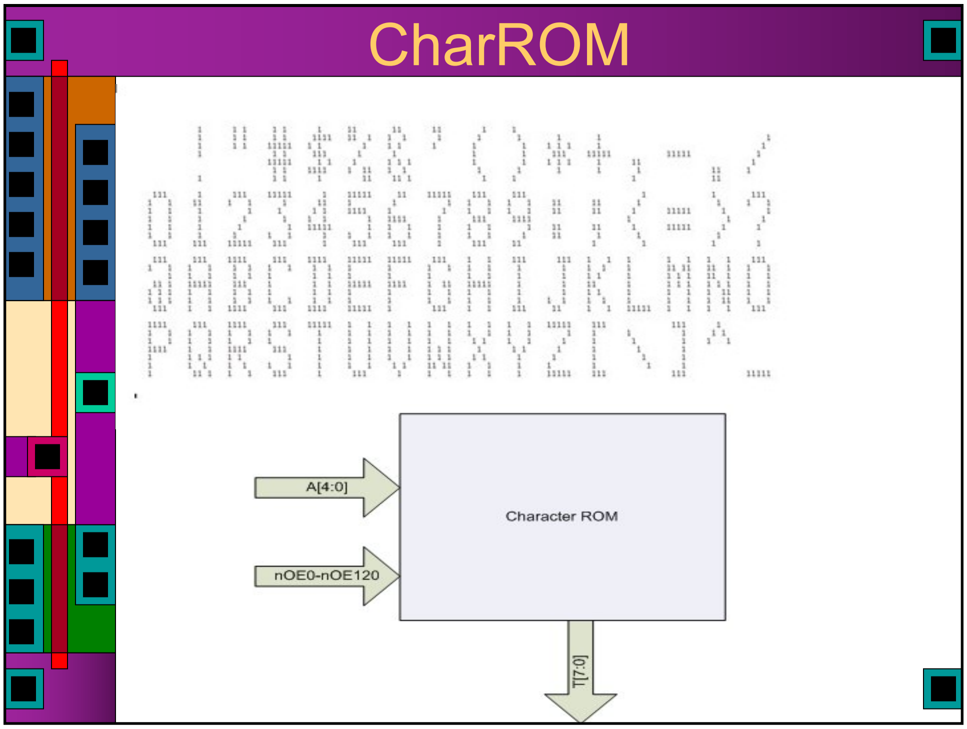
VGA Memory Requirements

- 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 16-bit word?
    - 2400 words for the 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



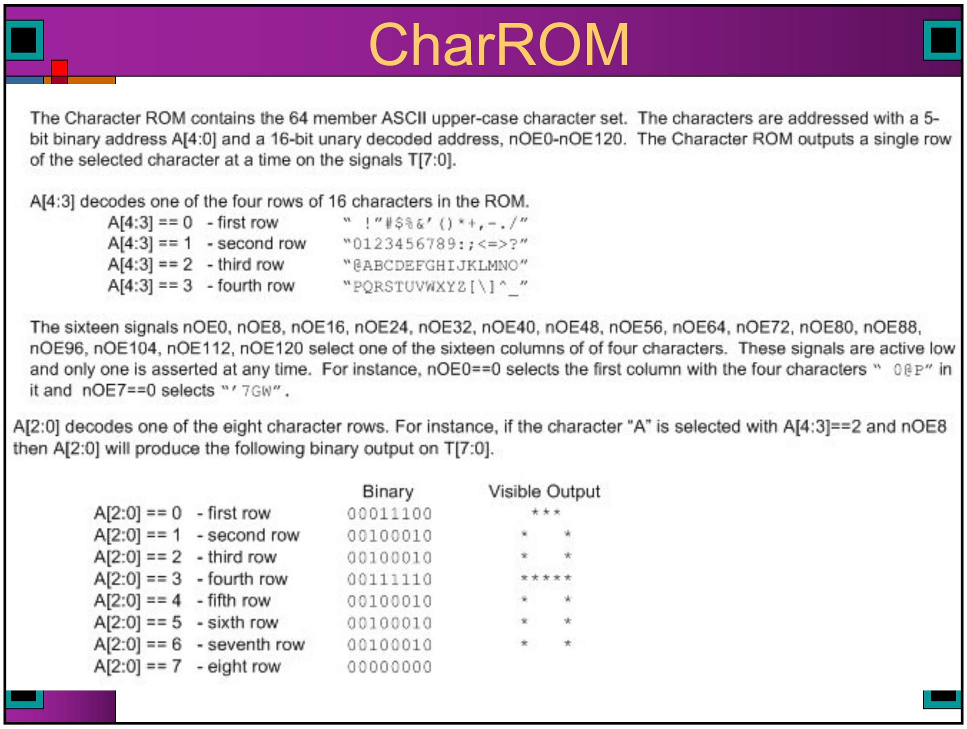
VGA Memory Requirements

- 80 char by 60 line display (8x8 glyphs)
  - 4800 locations
  - Each location has one of 64 char/glyphs
  - 6-bits per location
    - 4 locations per 24-bit word?
    - 1200 words for frame buffer?
  - Each char/glyph is (say) 8x8 pixels
    - results in 640x480 display…
  - 8x8x64 bits for char/glyph table
    - 4kbits for char/glyph table (32 words, 128 b/word)
    - Will this fit on your chip?



CharROM

57



CharROM

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.

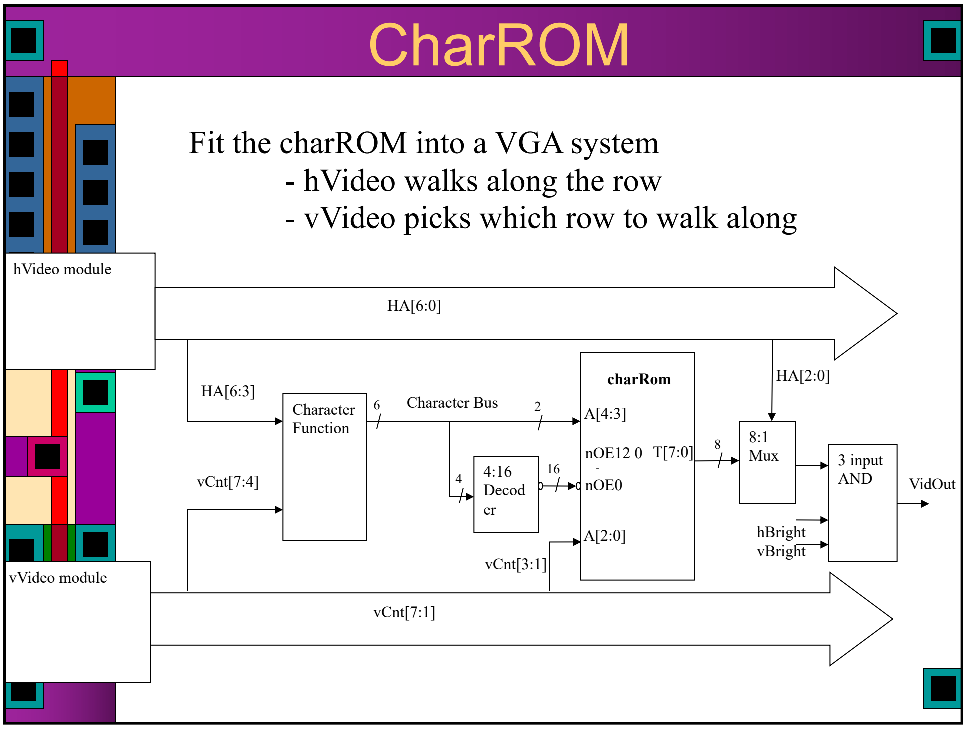
```
A[4:3] == 0   - first row      " !"#$%&'()*+,-./"
A[4:3] == 1   - second row     "0123456789:;<=>?"
A[4:3] == 2   - third row      "@ABCDEFGHIJKLMNO"
A[4:3] == 3   - fourth row     "PQRSTUVWXYZ[\]^_"
```

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
```

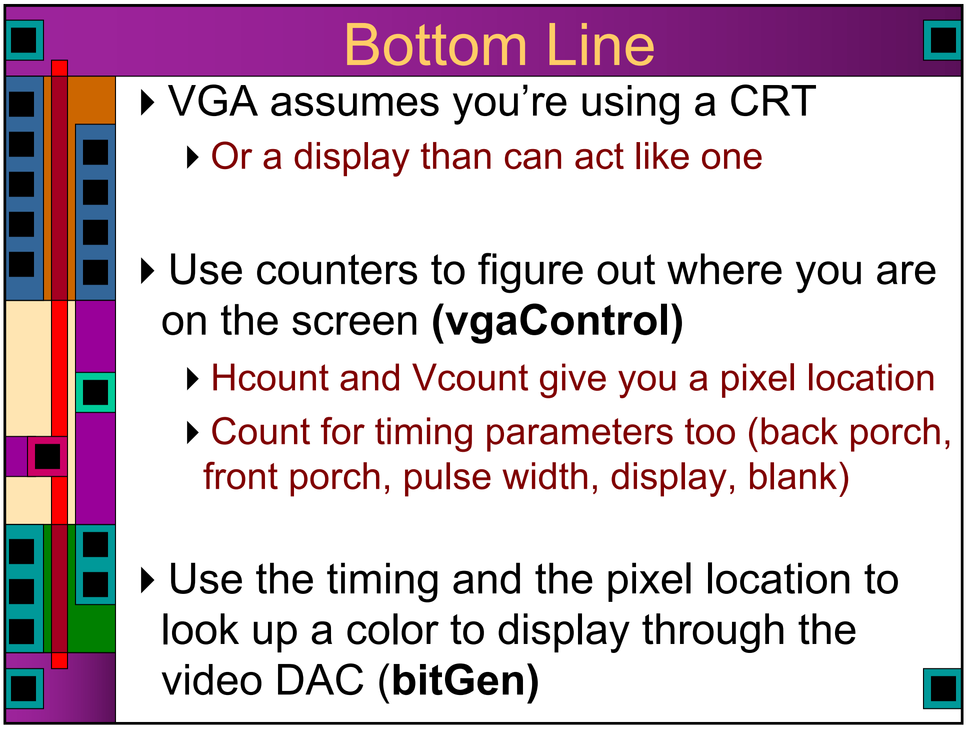
58



CharROM

Fit the charROM into a VGA system
- hVideo walks along the row
- vVideo picks which row to walk along

59



Bottom Line

- VGA assumes you're using a CRT
  - Or a display than can act like one
- Use counters to figure out where you are on the screen **(vgaControl)**
  - Hcount and Vcount give you a pixel location
  - Count for timing parameters too (back porch, front porch, pulse width, display, blank)
- Use the timing and the pixel location to look up a color to display through the video DAC (**bitGen**)

60

Terasic DE1-SoC

- Video DAC ADV7123

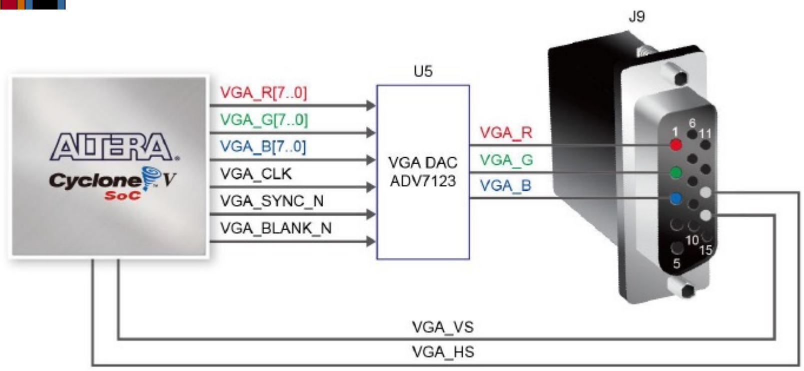


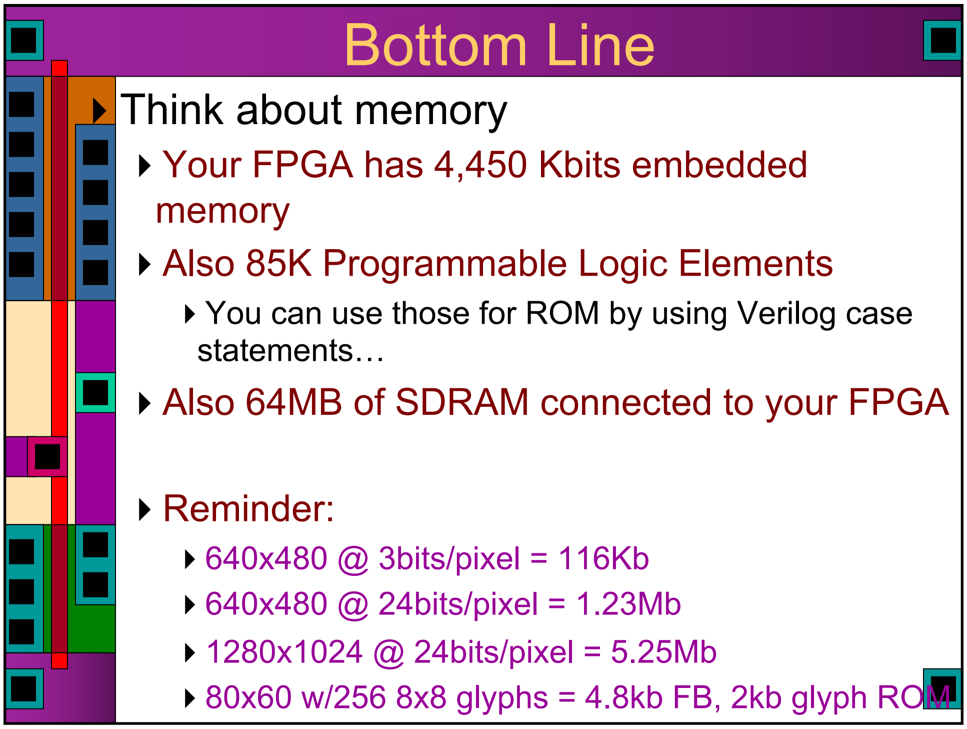
Figure 3-22 Connections between the FPGA and VGA

61

Bottom Line

- Choose a display size
  - That sets your frame buffer size
- Or choose a glyph size, and how many glyphs
  - That sets the size of your glyph ROM
  - And the size of your frame buffer to hold glyph numbers
- Start counting!

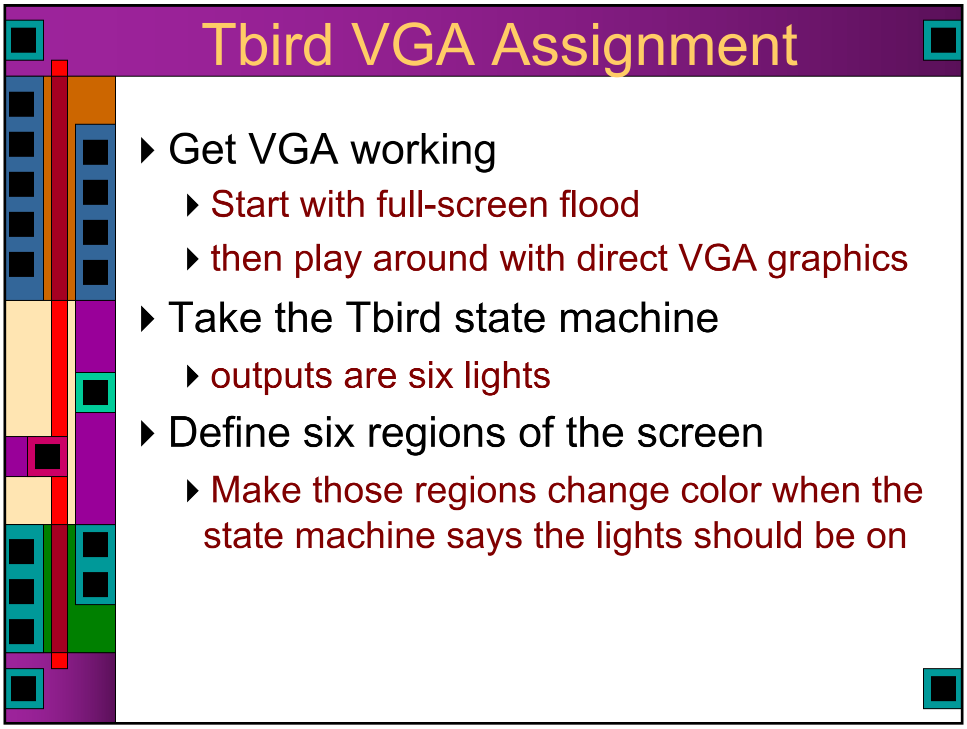
62



Bottom Line

- Think about memory
  - Your FPGA has 4,450 Kbits embedded memory
  - Also 85K Programmable Logic Elements
    - You can use those for ROM by using Verilog case statements…
  - Also 64MB of SDRAM connected to your FPGA
  - Reminder:
    - 640x480 @ 3bits/pixel = 116Kb
    - 640x480 @ 24bits/pixel = 1.23Mb
    - 1280x1024 @ 24bits/pixel = 5.25Mb
    - 80x60 w/256 8x8 glyphs = 4.8kb FB, 2kb glyph ROM

63



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

64