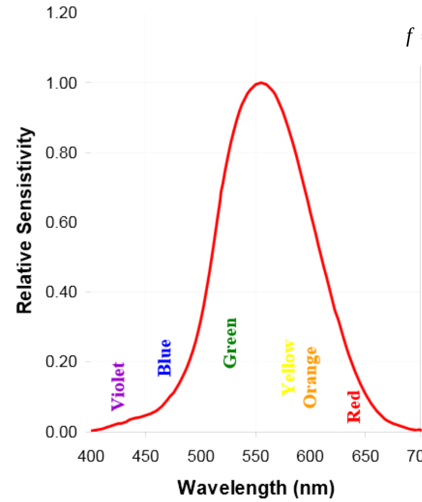
Video

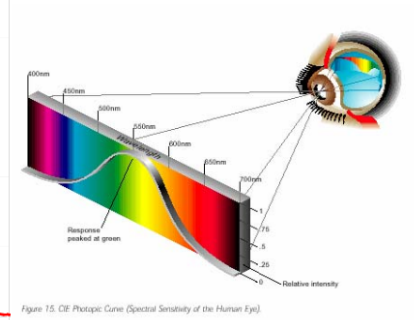
- Display technologies
- Generating video sync signals
- Decoding NTSC video
  -- color space conversions
- Generating pixels
  -- test patterns
  -- character display
  -- sprite-based games

Lab #3 issued – due Tue 9/30

Human Eye— Spectral Response

$$f = \frac{c}{wavelength} = \frac{2.9979*10^8}{600*10^{-9}} = 500*10^{12}\ (terahertz)$$





Display Terminology

Term	Definition
Pixel	Picture element - The smallest unit that can be addressed to give color and intensity
Pixel Matrix	Number of rows by column for the display
Aspect Ratio	Ratio of display width to display height: 4:3, 16:9
Resolution (ppi)	Number of pixels per unit length (pixel per inch)
Frame Rate (Hz)	Number of frames displayed per second
Viewing Angle (°)	Angular range over which images can viewed without distortion
Diagonal Size	Length of display diagonal
Contrast Ratio	Ratio of highest luminance (brightest) to lowest luminance (darkest)
TFT	Thin Film Transistor (narrow viewing angle)
IPS	In-plane Switch (wide viewing angle)
E-Ink ©	Electrophoretic Display

Display Capacity

Resolution	Pixel	Aspect Ratio	Products
VGA	640x480	4:3	
SVGA	800x600	4:3	
XGA	1024x768	4:3	iPad, iPad Mini
SXGA	1280x1024	4:3	
HD TV	1920x1080	16:9	
iPad Retina	2048x1536	4:3	iPad Air, iPad Mini Retina
Macbook Retina	2560x1600	16:10	13" Macbook Pro
Kindle Fire	1920x1200		HDX 7" (3rd Generation)

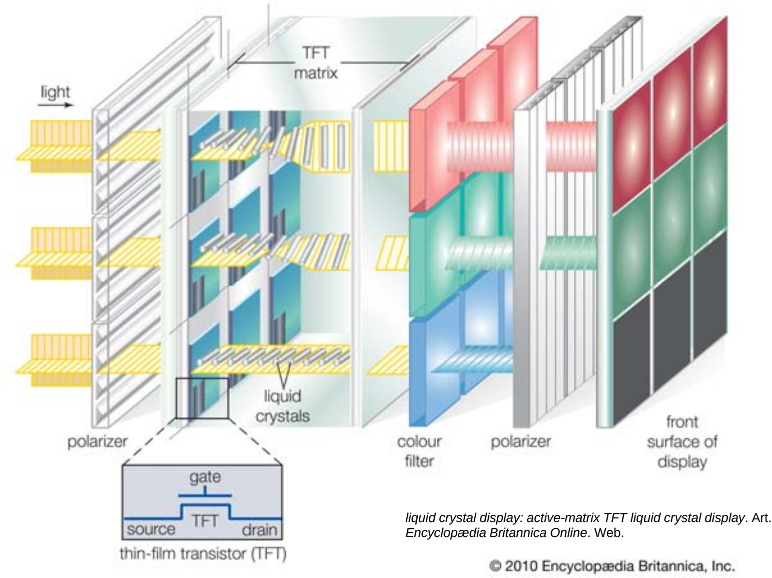
Display Types

- Emissive Display
  - Liquid Crystal Display (LCD)
    - requires backlight source,
    - constant power
  - Cathode Ray Tube (CRT)

- Reflective Display
  - Electrophoretic Display (E-Ink)*
    - Ultra Low Power – displays are bi-stable, drawing power only when updating the display.
    - Viewable in sunlight – ambient light reflected from display

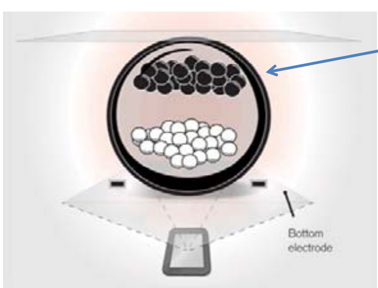
*Prof Joseph Jacbson, MIT

TFT LCD



liquid crystal display: active-matrix TFT liquid crystal display. Art. *Encyclopædia Britannica Online*. Web.

E-Ink *



When a positive or negative electric field is applied, corresponding positively charged white particles or negatively charged black particles move to the top of the microcapsule where they become visible to the viewer.

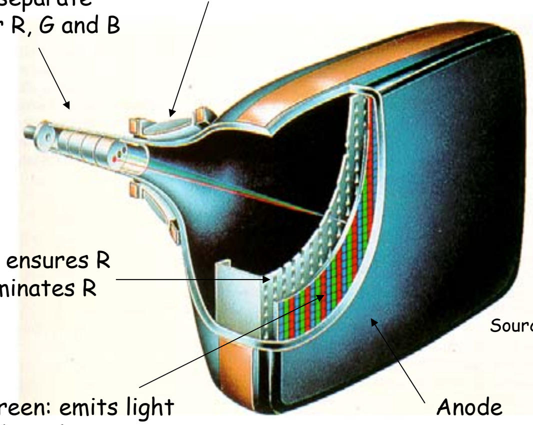
This makes the surface appear white or black at that spot.

*http://www.eink.com/technology.html

Background: Cathode Ray Tubes

Deflection coil (aka yoke): magnetically steers beam in a left-to-right top-to-bottom pattern. There are separate H and V coils.

Cathode: separate beams for R, G and B



Shadow mask: ensures R beam only illuminates R pixels, etc.

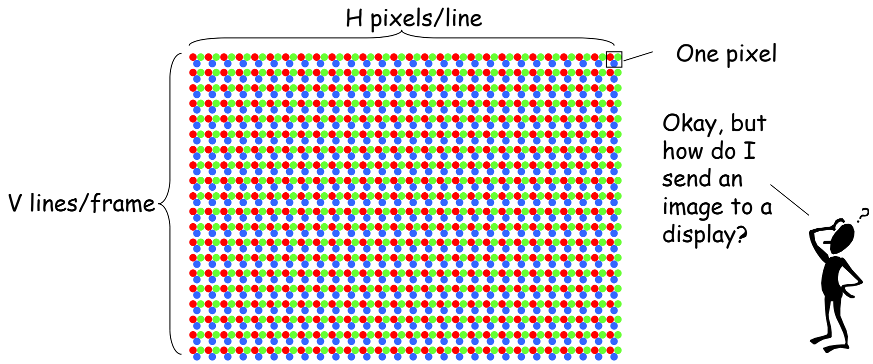
Source: PixTech

Phosphor Screen: emits light when excited by electron beam, intensity of beam determines brightness

Anode

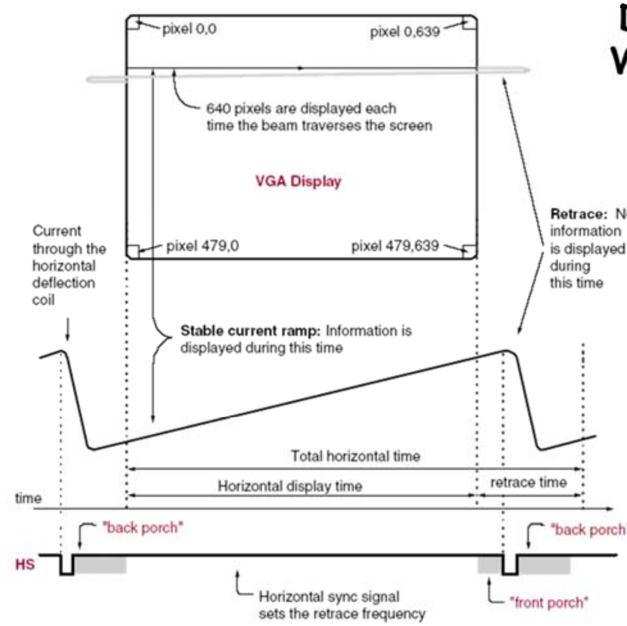
The CRT: Generalized Video Display

Think of a color video display as a 2D grid of picture elements (pixels). Each pixel is made up of red, green and blue (RGB) emitters. The relative intensities of RGB determine the apparent color of a particular pixel.



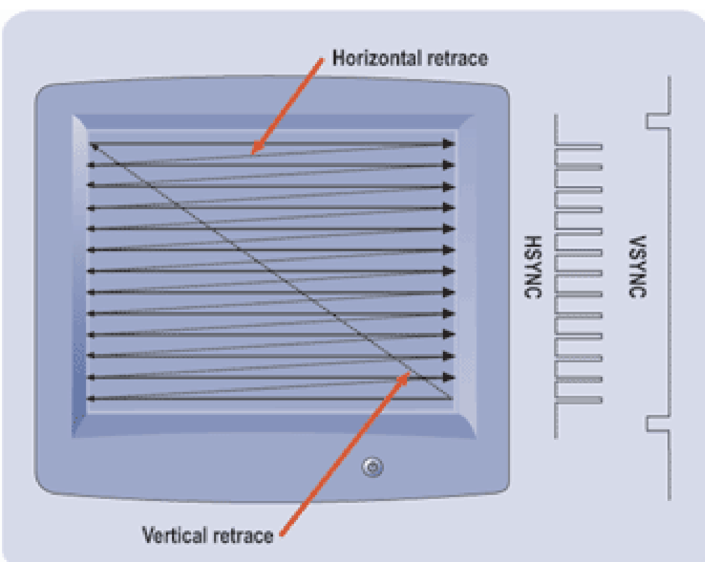
Traditionally H/V = 4/3 or with the advent of high-def 16/9.
Lots of choices for H,V and display technologies (CRT, LCD, …)

Deflection Waveforms



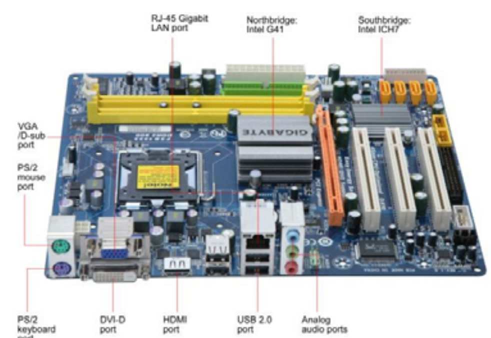
Source: Xilinx Spartan-3 Starter Kit Board User Guide

Sync Signals (HS and VS)



Video Evolution

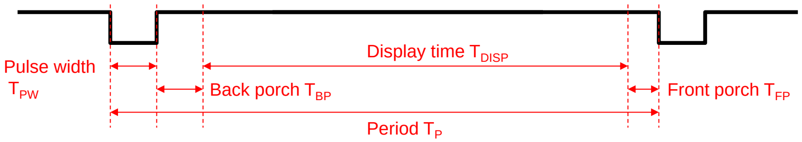
- VGA (Video Graphics Array) standard being replaced by DVI (Digital Visual Interface) and HDMI (High Definition Multimedia Interface). HDMI ~ DVI + Audio.



GIGABYTE GA-G41M-ES2H

Sync Signal Timing

The most common ways to send an image to a video display (even displays that don't use deflection coils, eg, LCDs) require you to generate two sync signals: one for the horizontal dimension (HS) and one for the vertical dimension (VS).



Format		CLK	P	PW	BP	DISP	FP
VGA	HS (pixels)	25Mhz	794	95	47	640	13
	VS (lines)	--	528	2	33	480	13
XGA	HS (pixels)	65Mhz	1344	136	160	1024	24
	VS (lines)	--	806	6	23	768	9

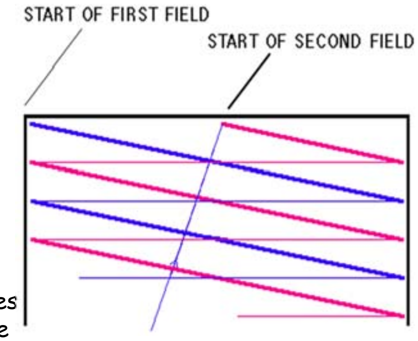
Interlace

Non-interlaced (aka progressive) scanning:

- VS period is a multiple of HS period
- Frame rate >= 60Hz to avoid flicker

Interlaced scanning:

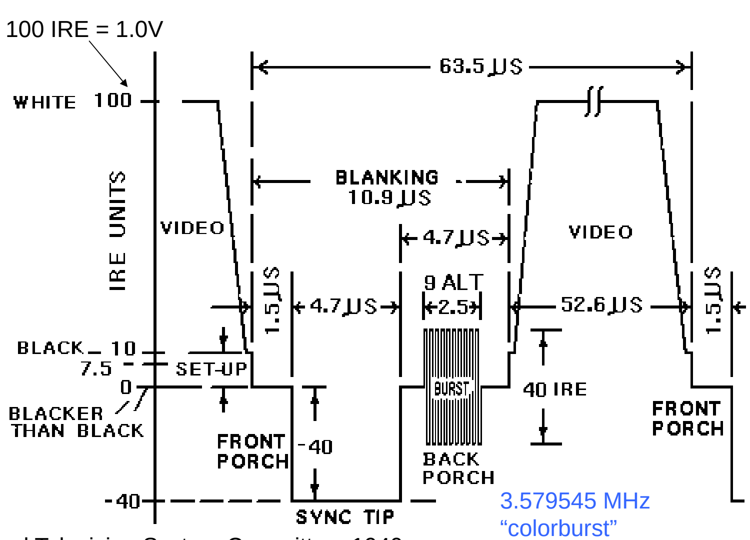
VS period is *not* a multiple of HS period, so successive vertical scan are offset relative to horizontal scan, so vertical position of scan lines varies from frame to frame.



NTSC example:

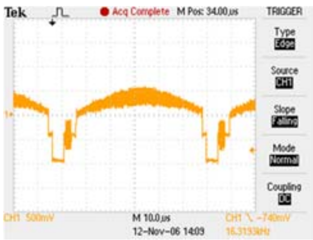
- •525 total scan lines (480 displayed)
- •2 fields of 262.5 scan lines (240 displayed). Field rate is 60Hz, frame rate = 30Hz

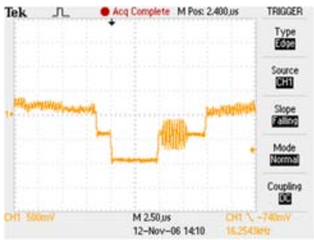
NTSC*: Composite Video Encoding

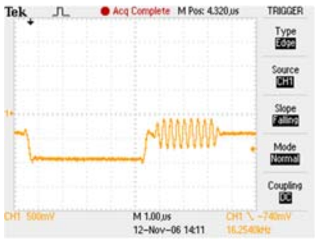


*National Television System Committee: 1940

Source: http://www.ntsc-tv.com

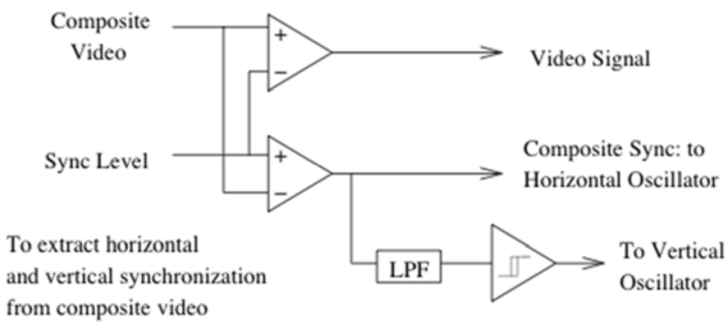






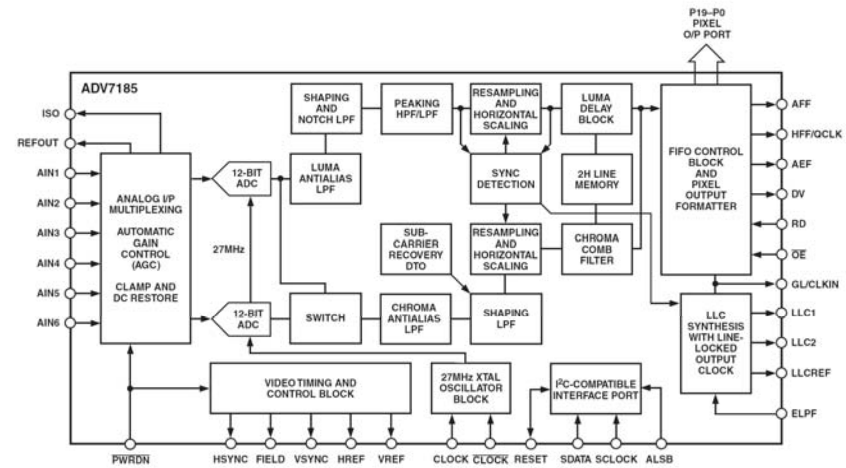
Video Capture: Signal Recovery

- Composite video has picture data and both syncs.
  - Picture data (video) is above the sync level.
  - Simple comparators extract video and composite sync.
- Composite sync is fed directly to the horizontal oscillator.
- A low-pass filter is used to separate the vertical sync.
  - The edges of the low-passed vertical sync are squared up by a Schmidt trigger.



Labkit: ADV7185 NTSC Decoder

- Decodes NTSC and PAL video (composite or S-video)
- Produces CCIR656 (10-bit) or CCIR601 (8-bit) digital data



Labkit: ADV7185 NTSC Decoder

- Decodes NTSC and PAL video (composite or S-video)
- Produces CCIR656 (10-bit) or CCIR601 (8-bit) digital data

BLANKING PERIOD			TIMING REFERENCE CODE				720 PIXELS YUV 4 : 2 : 2 DATA									TIMING REFERENCE CODE				BLANKING PERIOD		
...	80	10	FF	00	00	SAV	C_B0	Y0	C_R0	Y1	C_B2	Y2	...	C_R718	Y719	FF	00	00	EAV	80	10	...

Pixel 1: Y1,C_B0,C_R0

Pixel 0: Y0,C_B0,C_R0

```
8-bit SAV/EAV code: 1FVHabcd
10-bit SAV/EAV code: 1FVHabcd00
F = field (0: field 1/odd, 1: field 2/even)
V = vsync (0 for SAV)
H = hsync (0 for SAV)
a = V^H
b = F^H
c = F^V
d = F^V^H
8h'80, 10'h200 = start of even field
8h'C7, 10'h31C = start of odd field
```

8-bit data:
Y in range 16-235;
C_R, C_B in range 16-240 (offset by 128)

10-bit data:
Y in range 64-943;
C_R, C_B in range 64-963 (offset by 512)

YCrCb to RGB (for display)

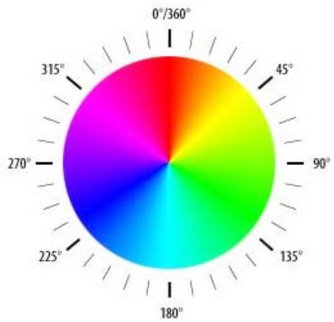
- 8-bit data
  - R = 1.164(Y – 16) + 1.596(Cr – 128)
  - G = 1.164(Y – 16) – 0.813(Cr – 128) – 0.392(Cb – 128)
  - B = 1.164(Y – 16) + 2.017(Cb – 128)
- 10-bit data
  - R = 1.164(Y – 64) + 1.596(Cr – 512)
  - G = 1.164(Y – 64) – 0.813(Cr – 512) – 0.392(Cb – 512)
  - B = 1.164(Y – 64) + 2.017(Cb – 512)
- Implement using
  - Integer arithmetic operators (scale constants/answer by 2^{11})
  - 5 BRAMs (1024x16) as lookup tables for multiplications

http://www-mtl.mit.edu/Courses/6.111/labkit/video.shtml

Video Feature Extraction

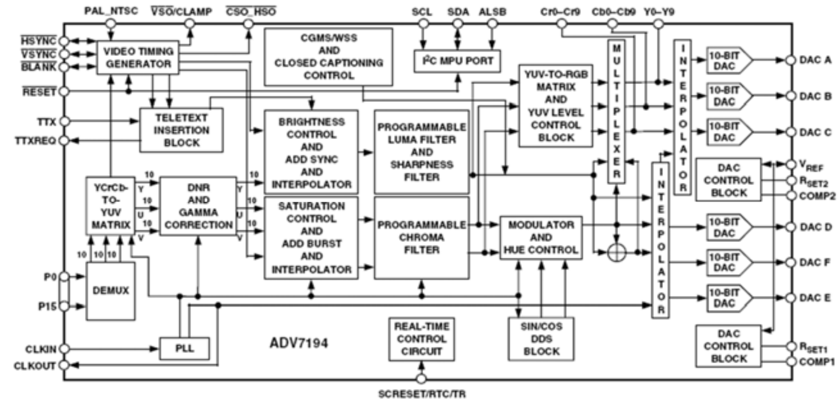
- A common technique for finding features in a real-time video stream is to locate the center-of-mass for pixels of a given color
  - Using RGB can be a pain since a color (eg, red) will be represented by a wide range of RGB values depending on the type and intensity of light used to illuminate the scene. Tedious and finicky calibration process required.
- Consider using a HSL/HSV color space
  - H = hue (see diagram)
  - S = saturation, the degree by which color differs from neutral gray (0% to 100%)

  - L = lightness, illumination of the color (0% to 100%)



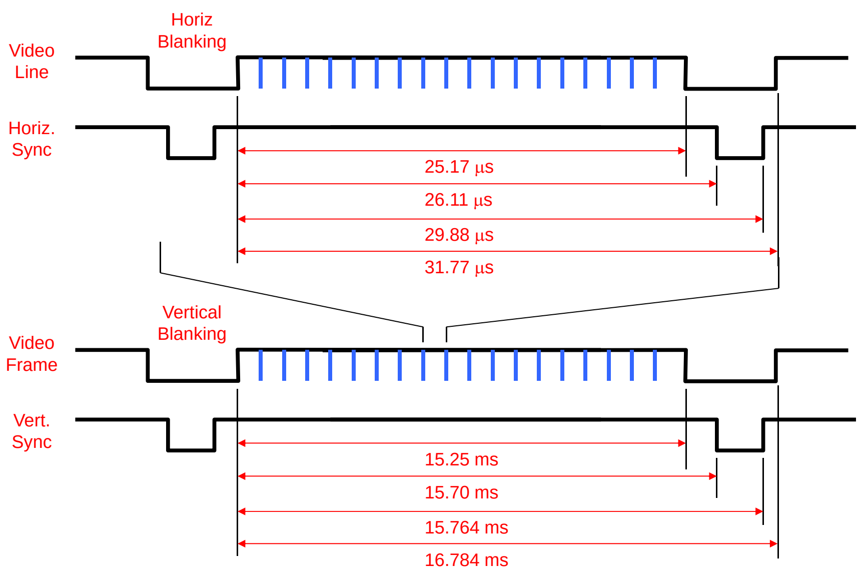
- Filter pixels by hue!

Labkit: AD7194 Digital Video Encoder



CCIR 601/656 4:2:2 digital video data → analog baseband TV signal

VGA (640x480) Video

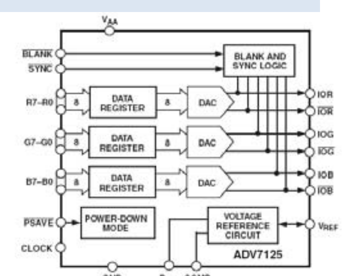


Labkit: ADV7125 Triple DAC (VGA)

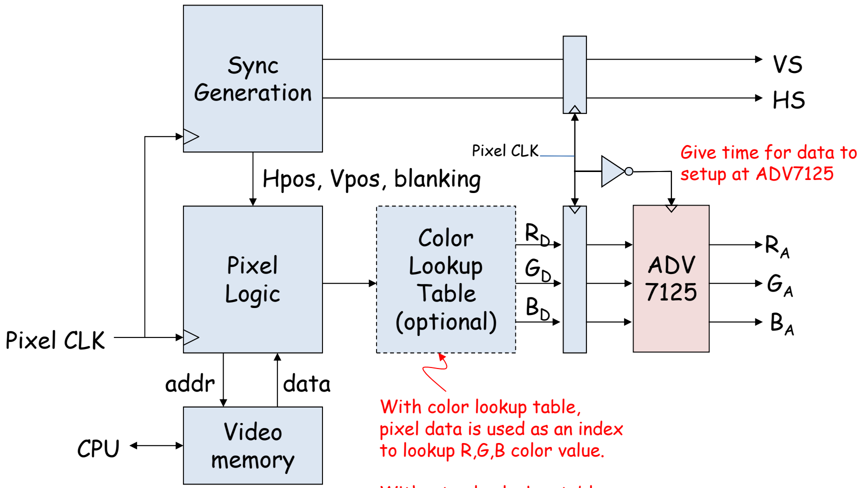
- Two Challenges:
  - (1) Generate Sync Signals
    - Sync signal generation requires precise timing
    - Labkit comes with 27 MHz clock
    - Use phase-locked-loops (PLL) to create higher frequencies
    - Xilinx FPGA's have a "Digital Clock Manager" (DCM)

```
DCM pixel_clock(.CLKIN(clock_27mhz),.CLKFX(pixel_clock));
// synthesis attribute CLKFX_DIVIDE of pixel_clock is 10
// synthesis attribute CLKFX_MULTIPLY of pixel_clock is 24
// 27MHz * (24/10) = 64.8MHz
```

  - (2) Generate Video Pixel Data (RGB)
    - Use ADV7125 Triple DAC
    - Send 24 bits of R,G,B data at pixel clock rate to chip
    - Create pixels either in real time
    - Or using dual port RAM
    - Or from character maps
    - Or ...?



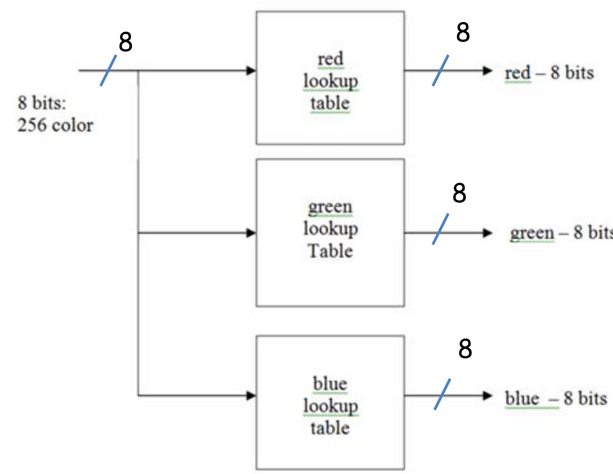
Generating VGA-style Video



With color lookup table, pixel data is used as an index to lookup R,G,B color value.

Without color lookup table, pixel data is used directly as R,G,B value (aka "true color")

Lookup Table



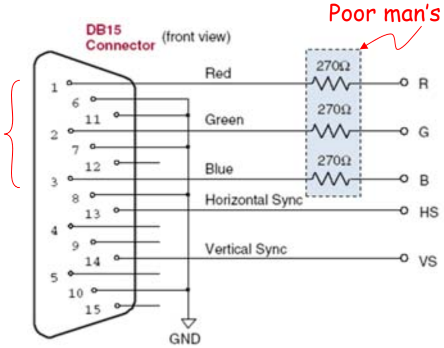
4 bit – 16 colors

8 bit – 256 colors

24 bit – 16M colors

Simple VGA Interface for FPGA

Poor man's Video DAC



Your circuitry should produce TTL-level signals (3.3V high level)

HS, VS are active-low signals.

R, G, B are active-high.

Shown: a simple "8-color" scheme

The R, G and B signals are terminated with 75 Ohms to ground inside of the VGA monitor. So when you drive your 3.3V signal through the 270 Ohm series resistor, it shows up at the monitor as 0.7V – exactly what the VGA spec calls for.

$$0.7V = \left(\frac{75}{75+270}\right)(3.3V)$$

Verilog: XVGA Display (1024x768)

```
module xvga(clk,hcount,vcount,hsync,vsync);
  input clk;             // 64.8 Mhz
  output [10:0] hcount;
  output [9:0] vcount;
  output hsync, vsync;
  output [2:0] rgb;

  reg hsync,vsync,hblank,vblank,blank;
  reg [10:0] hcount;      // pixel number on current line
  reg [9:0] vcount;       // line number

  wire hsyncon,hsyncoff,hreset,hblankon; // next slide for generation
  wire vsyncon,vsyncoff,vreset,vblankon; // of timing signals

  wire next_hb = hreset ? 0 : hblankon ? 1 : hblank; // sync & blank
  wire next_vb = vreset ? 0 : vblankon ? 1 : vblank;

  always @(posedge clk) begin
    hcount <= hreset ? 0 : hcount + 1;
    hblank <= next_hb;
    hsync <= hsyncon ? 0 : hsyncoff ? 1 : hsync;   // active low

    vcount <= hreset ? (vreset ? 0 : vcount + 1) : vcount;
    vblank <= next_vb;
    vsync <= vsyncon ? 0 : vsyncoff ? 1 : vsync;   // active low
  end
```


XVGA (1024x768) Sync Timing

```
// assume 65 Mhz pixel clock

// horizontal: 1344 pixels total
// display 1024 pixels per line
assign hblankon = (hcount == 1023);  // turn on blanking
assign hsyncon = (hcount == 1047);   // turn on sync pulse
assign hsyncoff = (hcount == 1183);  // turn off sync pulse
assign hreset = (hcount == 1343);    // end of line (reset counter)

// vertical: 806 lines total
// display 768 lines
assign vblankon = hreset & (vcount == 767);   // turn on blanking
assign vsyncon = hreset & (vcount == 776);    // turn on sync pulse
assign vsyncoff = hreset & (vcount == 782);   // turn off sync pulse
assign vreset = hreset & (vcount == 805);     // end of frame
```


Video Test Patterns

- Big white rectangle (good for "auto adjust" on monitor)

```
always @(posedge clk) begin
   if (vblank | (hblank & ~hreset)) rgb <= 0;
   else
    rgb <= 24'bFFF;
end
```

- Color bars

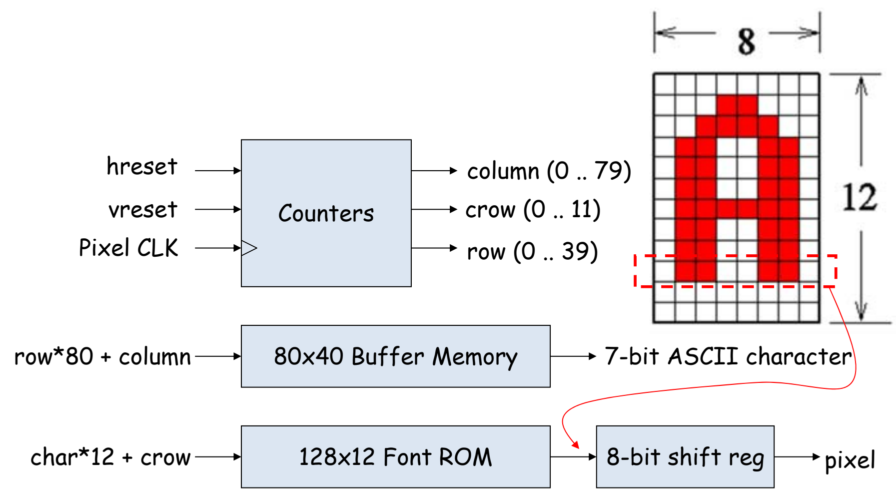
```
always @(posedge clk) begin
   if (vblank | (hblank & ~hreset)) rgb <= 0;
   else
    rgb <= {8{hcount[8]}, 8{hcount[7]},
            8{hcount[6]}};
end
```

RGB	*Color*
000	black
001	blue
010	green
011	cyan
100	red
101	magenta
110	yellow
111	white

rgb is 24 bits wide; 8 R, 8 G, 8 B

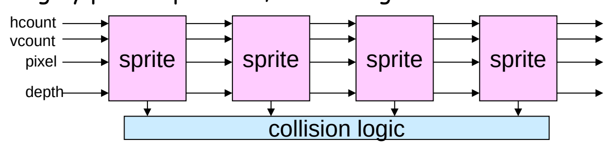
Character Display

(80 columns x 40 rows, 8x12 glyph)

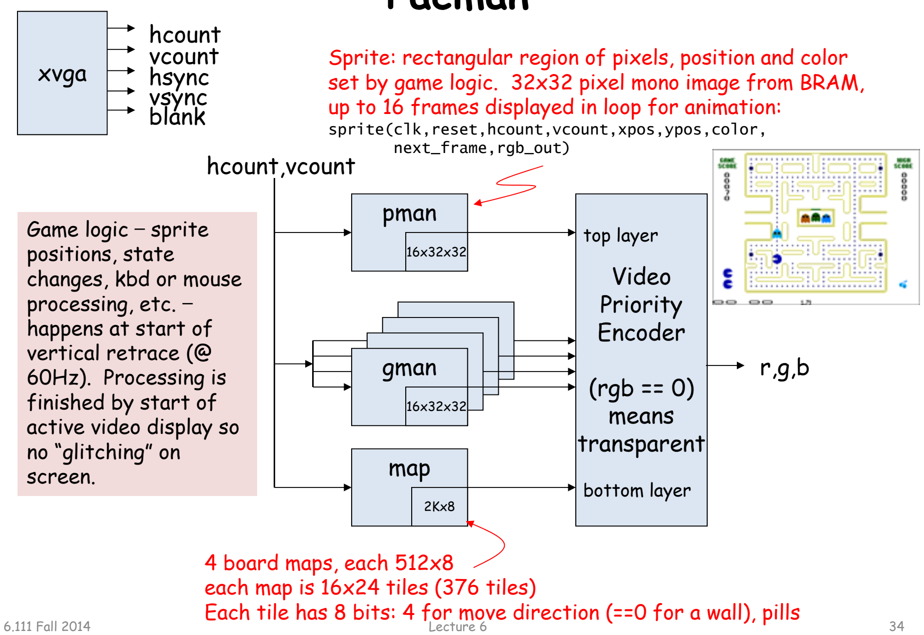


Game Graphics using Sprites

- Sprite = game object occupying a rectangular region of the screen (it's bounding box).
  - Usually it contains both opaque and transparent pixels.
  - Given (H,V), sprite returns pixel (0=transparent) and depth
  - Pseudo 3D: look at current pixel from all sprites, display the opaque one that's in front (min depth): see sprite pipeline below
  - Collision detection: look for opaque pixels from other sprites
  - Motion: smoothly change coords of upper left-hand corner
- Pixels can be generated by logic or fetched from a bitmap (memory holding array of pixels).
  - Bitmap may have multiple images that can be displayed in rapid succession to achieve animation.
  - Mirroring and 90° rotation by fooling with bitmap address, crude scaling by pixel replication, or resizing filter.



Pacman



Video Memory

- For complex video (images, computer generated graphics) a bitmap of the image is stored in memory – the frame buffer
  - each memory location represents one pixel
  - memory size = row × colomn × color depth × z
  - labkit ZBT memory can be used as frame buffer
    - 2 banks of 512K x 36 RAM
- For smooth video, two frame buffers are used – one for the display and one for updating. Buffers are switch during video retrace.
- Dedicated graphics accelerator and high speed memory used in high performance graphics.

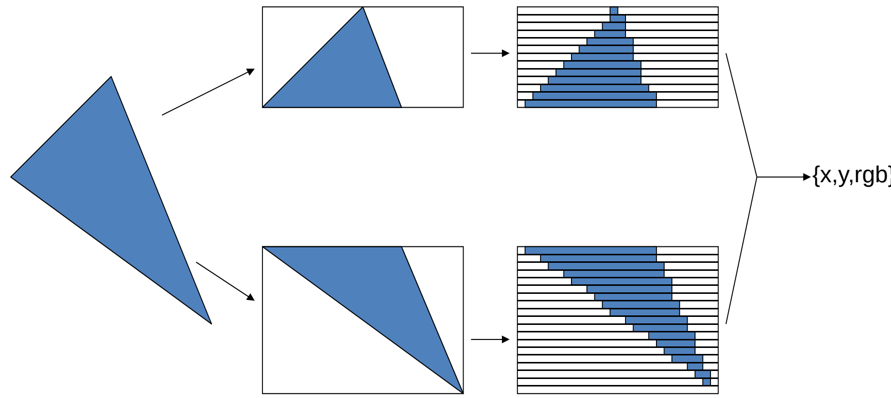
Rubik's Cube Solver

Katharine Daly, Jack Hutchinson
Fall 2013

3D Pong

Igor Ginzburg - Spring 2006

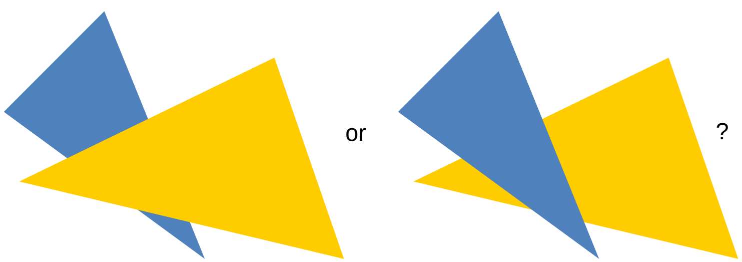
Graphics Generation



Graphics stored as triangles or polygons

Credit: Igor Ginzburg

Z-Buffer – Painter's Algorithm



- Buffer z-coordinate in addition to RGB for each pixel
- Compare z-coordinates before storing a new pixel color

Credit: Igor Ginzburg

Ray Tracer

Sam Gross, Adam Lerer - Spring 2007