VidFX – Video Effects Processor

- Video stream is decoded, processed, and redisplayed
- Commands are sent by you from Linux server
- The receiver gets a command from the networked Linux server. It decodes the command and dynamically changes the video processing
- Your particular board is addressed by your private MAC address.
- Everyone (working in groups of 2) will design, implement, debug, and demo a VidFX system on the Calinx board.
Calinx Board

- Video & Audio Ports
- AC '97 Codec & Power Amp
- Video Encoder & Decoder
- Flash Card & Micro-drive Port
- Four 100 Mb Ethernet Ports
- 8 Meg x 32 SDRAM
- Quad Ethernet Transceiver
- Prototype Area
- Seven Segment LED Displays
- Xilinx Virtex 2000E
Outline

1. Rough calculations / feasibility
2. Network Side
   - LANs
   - Network stacks
   - Ethernet
   - 125 Network Architecture
   - VidFX Packet Format
   - Calinx Network Interface
3. Video Side
   - Digital Video Basics
   - Example Video Standards
   - Calinx Video Interface
     (encoder/decoder)
   - Video Effects Processing
4. Frame-Buffer Design
5. VidFX High-level Organization
6. Review of Schedule and Design
   Checkpoints
Digital Video Basics

• Pixel Array:
  – A digital image is represented by a matrix of values where each value is a function of the information surrounding the corresponding point in the image. A single element in an image matrix is a picture element, or pixel. A pixel includes info for all color components.
  – The array size varies for different applications and costs. Some common sizes shown to the right.

• Frames:
  – The illusion of motion is created by successively flashing still pictures called frames.
Refresh Rates & Scanning

- The human perceptual system can be fooled into seeing continuous motion by flashing frames at a rate of around 20 frames/sec or higher.
  - Much lower and the movement looks jerky and flickers. TV in the US uses 30 frames/second (originally derived from the 60Hz line current frequency).
- Images are generated on the screen of the display device by “drawing” or scanning each line of the image one after another, usually from top to bottom.
- Early display devices (CRTs) required time to get from the end of a scan line to the beginning of the next. Therefore each line of video consists of an active video portion and a horizontal blanking interval portion.
- A vertical blanking interval corresponds to the time to return from the bottom to the top.
  - In addition to the active (visible) lines of video, each frame includes a number of non-visible lines in the vertical blanking interval.
  - The vertical blanking interval is used these days to send additional information such as closed captions and stock reports.
Interlaced Scanning

- Early inventors of TV discovered that they could reduce the flicker effect by increasing the *flash-rate* without increasing the *frame-rate*.

- Interlaced scanning forms a complete picture, the frame, from two *fields*, each comprising half the scan lines. The second field is delayed half the frame time from the first.

- The first field, *odd field*, displays the odd scan lines, the second, *even field*, displays the even scan lines.

- Non-interlaced displays are call *progressive scan*. 
Pixel Components

• A natural way to represent the information at each pixel is with the brightness of each of the primary color components: red, green and blue (RBG).
  – In the digital domain we could transmit one number for each of red, green, and blue intensity.

• Engineers had to deal with issue when transitioning from black and white TV to color. The signal for black and white TV contains the overall pixel brightness (a combination of all color components).
  – Rather than adding three new signals for color TV, they decided to encode the color information in two extra signals to be used in conjunction with the B/W signal for color receivers and could be ignored for the older B/W sets.

• The color signals (components) are color differences, defined as:
  B-Y and R-Y, where Y is the brightness signal (component).

• In the digital domain the three components are called:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td><strong>luma</strong>, overall brightness</td>
<td></td>
</tr>
<tr>
<td>C_B</td>
<td><strong>chroma</strong>, Y-B</td>
<td></td>
</tr>
<tr>
<td>C_R</td>
<td><strong>chroma</strong>, Y-R</td>
<td></td>
</tr>
</tbody>
</table>

• Note that it is possible to reconstruct the RGB representation if needed.

• One reason this representation survives today is that the human visual perceptual system is less sensitive to spatial information in chrominance than it is in luminance. Therefore chroma components are usually **subsampled** with respect to luma component.
Chroma Subsampling

Variations include subsampling horizontally, both vertically and horizontally.
Chroma samples are coincident with alternate luma samples or are sited halfway between alternate luma samples.
Common Interchange Format (CIF)

Example 1: commonly used as output of MPEG-1 decoders.

- Common Interchange Format (CIF)
- Developed for low to medium quality applications. Teleconferencing, etc.
- Variations:
  - QCIF, 4CIF, 16CIF
- Examples of component streaming:
  - line i: \[ Y \ C_R \ Y \ Y \ C_R \ Y \ Y \ ... \]
  - line i+1: \[ Y \ C_B \ Y \ Y \ C_B \ Y \ Y \ ... \]

Alternate (different packet types):
- line i: \[ Y \ C_R \ Y \ C_B \ Y \ C_R \ Y \ C_B \ Y \ ... \]
- line i+1: \[ Y \ Y \ Y \ Y \ Y \ Y \ Y \ ... \]

Bits/pixel:
- 6 components / 4 pixels
- 48/4 = 12 bits/pixel

<table>
<thead>
<tr>
<th>Frame size</th>
<th>352 x 288</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame rate</td>
<td>30 /sec</td>
</tr>
<tr>
<td>Scan</td>
<td>progressive</td>
</tr>
<tr>
<td>Chroma subsampling</td>
<td>4:2:0</td>
</tr>
<tr>
<td>Chroma alignment</td>
<td>interstitial</td>
</tr>
<tr>
<td>Bits per component</td>
<td>8</td>
</tr>
<tr>
<td>Effective bits/pixel</td>
<td>12</td>
</tr>
</tbody>
</table>
ITU-R BT.601 Format

The Calinx board video encoder supports this format.

- Formerly, CCIR-601. Designed for digitizing broadcast NTSC (national television system committee) signals.
- Variations:
  - 4:2:0
  - PAL (European) version
- Component streaming:
  line i: \[ \begin{array}{ccc}
  Y & C_B & Y \\
  C_R & Y & C_B \\
  \end{array} \]
  line i+1: \[ \begin{array}{ccc}
  Y & C_B & Y \\
  C_R & Y & C_B \\
  \end{array} \]
- Bits/pixel:
  - 4 components / 2 pixels
  - 40/2 = 20 bits/pixel

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame size</td>
<td>720 x 487</td>
</tr>
<tr>
<td>Frame rate</td>
<td>29.97 /sec</td>
</tr>
<tr>
<td>Scan</td>
<td>interlaced</td>
</tr>
<tr>
<td>Chroma subsampling</td>
<td>4:2:2</td>
</tr>
<tr>
<td></td>
<td>2:1 in X only</td>
</tr>
<tr>
<td>Chroma alignment</td>
<td>coincident</td>
</tr>
<tr>
<td>Bits per component</td>
<td>10</td>
</tr>
<tr>
<td>Effective bits/pixel</td>
<td>20</td>
</tr>
</tbody>
</table>
Calinx Video Decoder

- Analog Devices ADV7185

- Takes NTSC (or PAL) video signal on analog side and outputs ITU601/ITU656 on digital side.
  - Many modes and features not used by us.
  - VidFX project will use default mode: no initialization needed.

- Generates 27MHz clock synchronized to the output data.
- Digital input side connected to Virtex pins.
- Analog output side wired to on board connectors or headers. Camera connection through “composite video”.

The diagram shows the block diagram of the ADV7185 video decoder, with inputs and outputs labeled for both the analog and digital sides.
Calinx Video Encoder

- Analog Devices ADV7185

- Supports:
  - Multiple input formats and outputs
  - Operational modes, slave/master
  - VidFX project will use default mode: ITU-601 as slave s-video output

- Digital input side connected to Virtex pins.
- Analog output side wired to on board connectors or headers.
- I²C interface for initialization:
  - Wired to Virtex.
  - We will supply this as a pre-designed module.
ITU-R BT.656 Details

- Interfacing details for ITU-601.
  - Pixels per line = 858
  - Lines per frame = 525
  - Frames/sec = 29.97
  - Pixels/sec = 13.5 M
  - Viewable pixels/sec = 720
  - Viewable lines/frame = 487
- With 4:2:2 chroma sub-sampling, need to send 2 words/pixel (1 Y and 1 C).
- words/sec = 27M, Therefore encoder runs off a 27MHz clock.
- Control information (horizontal and vertical synch) is multiplexed on the data lines.
- Encoder data stream show to right:

---

Note 1 – Sample identification numbers in parentheses are for 625-line systems where these differ from those for 525-line systems. (See also Recommendation ITU-R BT.803.)

Spring 2003 EECS150 lec12-proj2
ITU-R BT.656 Details

- Control is provided through “End of Video” (EAV) and “Start of Video” (SAV) timing references.
- Each reference is a block of four words: FF, 00, 00, <code>
- The <code> word encodes the following bits:
  F = field select (even or odd)
  V = indicates vertical blanking
  H = 1 if EAV else 0 for SAV
- Horizontal blanking section consists of repeating pattern 80 10 80 10 …
Video Effects Processing

Required processing:

1. **Freeze Frame** – upon receiving “freeze” command from the network, the currently displayed frame is held static. Subsequent “unfreeze” command will restore display to normal (intervening frames are lost).

2. **Zoom-out** - display quarter-sized image in center of screen.

Optional (extra-credit):

1. **Variable zoom out**: zoom-out command with a parameter indicating the zoom out amount.

2. **Black and white**: Eliminate the color components and display a black and white picture.

3. **Inverse video**: Subtracting the actual component value from the maximum component value.

4. **Zoom In**: Either simple line and pixel doubling or linear interpolation (higher quality).

5. **Dynamic change of aspect ratio**: Change the on screen aspect ratio by scaling one or the other dimension.

6. **Your proposal** – *make sure to get approval from teaching staff.*
Image Scaling (zoom-out)

- One technique is to replace every block of 2x2 pixels by a single pixel representing the original block of 4.
  a) Could just use one of the original pixels (lower right pixel, for instance).
  b) Different (integer) scale factors can be achieved using larger blocks.
  c) Better technique would use all four original pixels, for instance in an average calculation.

- Other more complex and higher-quality techniques are possible but not necessary for VidFX. The minimum required technique is a) above.
Frame Buffer

- Many video application designs are eased through the use of a frame buffer.
- It holds one frame of video as it is being displayed.
- Logically, the buffer can considered to have two ports – one for writing from the video source (camera in case of VidFX) and one for reading for video display.

For VidFX it permits the “freeze frame” command.

- Frame buffers are used on PC video display cards to decouple video generation from display.

Frame Buffer will not fit in memory internal to Virtex FPGA
- Virtex 2000E has 160 blockRAMs (4K bits each) = 655,360 bits.
- Buffer to be implemented using external SDRAM with “control logic” implemented on the FPGA.

from camera

Frame Buffer

720 x 487
= 350,640 Bytes

to video display
Active Lines per Frame

Tuesday’s Lecture said 507, today 487!?

• NTSC:
  525 lines/frame total
  487 active, 38 blanking
  – odd field:
    lines 4-19 (16 lines) blanking
    lines 20-263 (244) active
    lines 264-265 (2) blanking  
      262 lines
  – even field:
    lines 266-282 (17) blanking
    lines 283-525 (243) active
    Lines 1-3 (3) blanking  
      263 lines

  525 lines total

• Our camera generates 525 total lines per frame, but 507 of these are active.
  – 507-487 = 20 extra lines of the NTSC frame when the camera sends active video.
  – Which lines?
  – Doesn’t really matter:
    • Use control bits in video stream (V bit) to indicate active video versus blanking.
    • Implement frame buffer large enough to hold active camera data (507 lines).
Checkpoints

1. **Video Decoder:** (1 week, check-off by week of 3/10)
   
   Decode the incoming video stream and compute a value representing each one of the 3 video components. Display that value on the 7-segment LEDs

2. **Video Encoder:** (2 weeks, check-off by week of 3/31)
   
   Initialize blockRAM memory with video test pattern, display it on monitor. Tests video display control and data-path. Checks understanding of memory-based video control.

3. **SDRAM Test:** (2 weeks, check-off by week of 4/14)
   
   Write and read data patterns to SDRAM.

4. **Network Test:** (1 week, check-off by week of 4/21)
   
   Receive packets from network, display payload on LED display. Tests receiver MAC and packet filter/parser

5. **Integration, debugging, improvements.** (3 weeks, final check-off by week of 5/5) Extra credit for early check-off.
References for Video

1. For general background on video and digital video: Charles Poynton, "A technical Introduction to Digital Video", Chapter 1.

2. www.inforamp.net/~poynton/PDFs/TIDV/Basic_principle.pdf
3. Calinx board user's manual (on class website)

4. Tom Oberheim's "CS150 Board Digital Video in a Nutshell"

5. VideoNutshell.doc on class website

6. The official specification supported by the video encoder on the Calinx board: ITU specifications 656 & 601

   ITU656.doc, ITU601.doc on class website

7. Video encoder datasheet: ADV7194.pdf on class website
8. Video decoder datasheet: ADV7185.pdf on class website