EECS 150 Fall 2003

Lab Lecture 8
Checkpoint2
10/17/2003

Greg Gibeling
Adapted from lecture by Santor Pinto

Today
- Checkpoint 2 Goals
  - Block Diagram
  - Components
  - Announcements
  - Video Decoder
  - Digital Video
  - Video Encoder

Checkpoint2 (1)
- You will get video running on the board
  - We're starting small
  - Just one line of video displayed many times

Checkpoint2 (2)

Checkpoint2 (3)
- We will give you almost everything
  - SRAM Block
  - Asynchronous FIFO
  - Decoder
- You must build the encoder
  - We'll get you started
  - Build it well, it's a big part of your project

SRAM Block (1)
- This is a very simple piece of memory
  - SDRAM would be nice, but it's hard to use
  - For now we'll make our lives a little easier
  - Checkpoint3 will be integrating SDRAM into this checkpoint
**SRAM Block (2)**
- “linememory”
- Will contain one line of video: the first ACTIVE line
- Dual ported synchronous memory
  - Separate read and write ports
  - COULD be separately clocked, but we’ll use a FIFO
  - Synchronous read and write!

**SRAM Block (3)**
- 8 bits of address
  - Video is normally 720 pixels, you’ll only use 320.
  - Drop every other pixel until you have 320, then ignore the rest.
- 8 bits of data
  - Ignore Chroma, we want black and white
  - Set Chroma to 0x80 on output

**Async FIFO (1)**
- Similar to a Synchronous FIFO
- In and out ports are clocked separately
- These are very hard to make!
  - Gray Code for the Counters
  - All kinds of sync issues
  - Dual Ported SRAM
- AINIT instead of SINIT for Reset

**Async FIFO (2)**
- FIFO must never be empty or full
  - Full -> You will lose data
  - Empty -> You will create garbage data
- How do we know when to read/write?
  - Remember that it doesn't matter when we read the data from the SRAM!
  - It does matter when the encoder needs data

**Announcements**
- Checkpoint2 is relatively easy
  - We want you to think about the project (Design Review)
  - You will need to do this one well, bugs will cost you lots of time later so SIMULATE
- Due 10/24
- Sign up for Design Review
- Checkpoint1 Checkoffs...

**Decoder (1)**
- Decoder
  - Takes in raw ITU-R BT.656 video
  - Extracts sync signals
  - Outputs video in two pixel pairs
    - Remember the chroma subsampling?
    - This is a relatively simple module please read the verilog
**Decoder (2)**

<table>
<thead>
<tr>
<th>General Inputs</th>
<th>Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>RST</td>
<td>Reset</td>
</tr>
<tr>
<td>EN</td>
<td>Enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video Data In</th>
<th>F: Raw Data Input</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Decoder Control Outputs</th>
<th>RST_OUT: Reset Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO</td>
<td>Sync</td>
</tr>
<tr>
<td>SDA</td>
<td>PC Config Data Line</td>
</tr>
<tr>
<td>SCLK</td>
<td>PC Config Clock Line</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decoded Signals</th>
<th>CIF: Start Even Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAV</td>
<td>End Active Video</td>
</tr>
<tr>
<td>SAV</td>
<td>Start Active Video</td>
</tr>
<tr>
<td>VALID</td>
<td>Data Out Valid</td>
</tr>
<tr>
<td>DATA_OUT</td>
<td>Video, samples of video</td>
</tr>
</tbody>
</table>

---

**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 Synch State Machine (1)**

---

**Video Capture State Machine (2)**

---

**Encoder (1)**

- You must generate all sync signals
- The SRAM and FIFO must only contain valid, active video data
  - Bandwidth and memory requirements
  - FIFO/Clock Problems?
- We've given you a shell for your encoder
Encoder (2)
- Two Basic States
  - SENDI2C
  - SENDVIDEO (stay for ever)
- Everything else can be counter based
  - Vertical Counter
  - Horizontal Counter

Encoder (3)
- ADV7194 Initialization using I²C
  - Requires only 2 wires
    - Serial Data (Bidirectional)
    - Clock (Driven by master)
  - Runs at up to 400kHz
  - Bidirectional
  - Too complicated for the time you have, so we'll give it to you
  - Read datasheet for more

Where to Start
- Reread “Video in a Nutshell”
  - Useful Documents section on website
- Read the Verilog
  - Vid_Dec.V
  - FPGA_TOP.V (You will modify this)
  - Vid_Enc.V (You will modify this)

And now...
- We will check off Checkpoint1
  - You have till 4pm to be checked off
  - Its due today
- Partners – Come see me after lecture
- Questions?
  - Stick around and ask
  - Stick around and listen, you might hear something very useful.