Today
- Checkpoint 4 Goals
  - Block Diagram
  - Mixer Internals
  - Announcements
  - Interfacing with the Mixer

Checkpoint 4 (1)
- Dummy Vector Overlay
  - We like colorful arrows
  - This time we just want dummy arrows
  - Make a creative pattern
  - This checkpoint is trivial
  - It can be done in 8 lines of verilog
  - We'll tell you dam near everything
  - WORK ON YOUR PROJECT DESIGN

Checkpoint 4 (2)

Checkpoint 4 (3)
- Simple addition to checkpoint 3
  - No modifications to any existing modules
  - VectorBlend simply inserted after encoder
  - Add a line or two to encoder to tell blend which line its in/blanking
  - We’ve done all the major work for you

Checkpoint 4 (4)
- Use our overlay unit
  - Its tested and efficient
  - Feed in a test pattern of vectors
  - Vector values from dip switches
  - Counter to change values across the screen
  - We’ll use this module for real next time
**Vector Blend (1)**
- Why our unit?
  - Time
  - Bitmaps aren't easy the first time
  - Ours is relatively efficient
- How do we make arrows
  - Chroma and Luma Bitmaps
  - Bitmap scaled by intensity
  - Scaled Bitmap mixed with video

**Vector Blend (2)**
![Diagram of Vector Blend](image)

**Vector Blend (3)**
- Bitmaps
  - Luma and Chroma Bitmaps are just intensities
  - Actual color is added by VectorBlend module
  - Intensity from bitmap is scaled by Intensity input
- Scaled intensity is multiplied by "BLUE"
- Video in is multiplied by inverse of scaled intensity
- Video out is sum of scaled "BLUE" and scaled video in

**Vector Blend (4)**
- Why is this so hard?
  - Bitmap data must be converted from .BMP
    ![Bitmap Conversion](image)
  - Making initialized memory is a pain
  - Multiplying efficiently is tricky
  - Alpha Blending is tricky unless you know how

**Announcements (1)**
- Checkpoint3 Due 11/4 @ 8pm
  - You may checkoff through Friday 11/7 for 50% credit
- Checkpoint4 Due Tuesday 11/10
  - Do this early and take some time off
- NO CHECKPOINT5!
- Project Due Wed 11/26
  - +5 Bonus for Early Completion on 11/21

**Announcements (2)**
- Project Due Wed 11/26
  - +5 Bonus for Early Completion on 11/21
- **CHANGE IN PROJECT SPEC**
  - Instead of 30Hz vector updates: 1Hz!
  - This means one field of vectors computed each second
  - 30Hz full motion vectors are +5 bonus
Announcements (3)

- Design Reviews
  - Saw some nice ASMDs
  - We need to collect a copy of them!
- Next Review
  - Detailed RTL
  - Shell Verilog
  - You should be basically done with your design

Vector Blend Interface (1)

<table>
<thead>
<tr>
<th>Vector to Display</th>
<th>Direction</th>
<th>4</th>
<th>Direction of Vector to Display</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intensity</td>
<td>3</td>
<td>Intensity of Vector to Display</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input from Encoder</th>
<th>Row</th>
<th>4</th>
<th>Row of current vector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Col</td>
<td>3</td>
<td>Column of current vector (pixel pair)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Blanking</th>
<th>1</th>
<th>In blanking?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Video</th>
<th>VideoIn</th>
<th>8</th>
<th>Video from encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VideoOut</td>
<td>8</td>
<td>Video to ADV7194</td>
</tr>
</tbody>
</table>

| General | Clock | 1 | VE_CLOCK |

Vector Blend Interface (2)

- Video In/Out is simple
  - VectorBlend knows how to modify video
  - This keeps your encoder intact
  - If you can’t figure out the clock, you’re in trouble
  - The rest is a little tougher

Vector Blend Interface (3)

- Direction and Intensity
  - These are the inputs that specify a vector
  - Start with a dipswitch
  - Maybe use a counter
- Blanking
  - Should be 1'b1 during ALL blanking
  - This disables the vector blend unit
  - You’ll need this to make the borders...

Vector Blend Interface (4)

- Row
  - 16 rows per vector
  - Based on VCOUNT from your encoder
    - Perhaps with an offset
    - Row = VCOUNT[? :?] + ?
  - Remember fields!
    - Odd rows occur in one field, even rows in another

Vector Blend Interface (5)

- Column
  - 8 cols per vector
    - One per 4byte pixel pair
  - Based on HCOUNT
    - Perhaps with an offset
    - Col = HCOUNT[? :?] + ?
Vector Blend Interface (6)

- Remember you must do vectors a line at a time
  - Cycle through 39 vectors on row 0, then 39 vectors on row (1,2, what?)
- Don’t forget the borders!
  - Four pixel border on all sides
  - You must take this into account with blanking

Where to Start

- Think a bit
  - We went over most of the details today
  - What do you need to add to your encoder?
- Read the Verilog
  - You don’t have to this time
  - Its decently commented
  - With some thinking!

And now...

- We will check off Checkpoint3
  - You have till Tuesday to checkoff
  - 50% credit through friday
- Questions?
  - Stick around and ask
  - Stick around and listen, you might hear something very useful.