Project Overview

- Play ping pong with a physical paddle as input to hit a virtual ball displayed on screen
- Detect paddle using image processing
- Calculate paddle characteristics
- Display image detected by camera
- Display ball bouncing in three dimensions based on paddle
- Allow multiplayer games running on two FPGA’s
Technical Overview

NTSC Conversion and Storage

Camera Data

30 bits of ycrcb per pixel

X, Y coords
10 bits each

Image Processing:
Find paddle pixels and use them for calculations

9 bits of Tilt
(-180 to +180 degrees in two's complement),
8 bits of Velocity

To Other FPGA

Game Data from FPGA 2

Input Mux

Game Logic

Draw Static Features, Ball, Opponent

RGB for X,Y Query

Lab 5 XVGA Module

Monitor

Multiplier?
Virtual Ping Pong Vision

Mark Stevens
Vision Overview

Camera Data -> NTSC Conversion and Storage

- 30 bits of ycrcb per pixel
- X, Y coords 10 bits each

19 bit write address

18 bits of rgb per pixel

X, Y read request (19 bit address)

Stored RGB on ZBT Memory, Accessed by FSM

9 bits of Tilt (-180 to +180 degrees in two's complement)

8 bits of Velocity

X, Y coords for upper left and lower right corners (10 bits of x and y each)

Image Processing:
Find paddle pixels and use them for calculations

18 bits of rgb per pixel
NTSC Conversion and Storage

- Blocks from other semesters (zbt_6111, ntsc_decode, adv7185init) communicate with camera and ZBT.
- Altered ntsc2zbt and vram_display store two pixels per address in ZBT.
- New block converts Y, Cr, Cb to RGB.
Filter incoming data based on Y, Cr, Cb thresholds
Filter out noise
Output coordinates of corners and total area once per frame
Use relative positions of coordinates to calculate paddle tilt and velocity
Game Logic and Output

Zach Clifford
Game Logic stores ball location in \((X,Y,Z)\)

Every frame updates position and translates to pixel location for Draw Interface

Draw Interface places game elements on screen
Multiplayer Mode

- Moves relevant data between labkits
- Need to communicate one global ball position
- Also need to send paddle data because cameras are reversed
<table>
<thead>
<tr>
<th>Schedule</th>
<th></th>
<th>11/12</th>
<th>11/19</th>
<th>11/26</th>
<th>12/3</th>
<th>12/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Video from Camera</td>
<td></td>
<td>Working in grayscale</td>
<td>Refine blocks</td>
<td>Store RGB instead of grayscale</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detect and Filter Color</td>
<td></td>
<td>Working</td>
<td>Refine</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculate positions in image</td>
<td></td>
<td>Start</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculate Tilt, velocity</td>
<td></td>
<td>Start</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drawing Interface</td>
<td></td>
<td>Working</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics, Collisions</td>
<td></td>
<td>Works w/o paddle</td>
<td>Add paddle position</td>
<td>Add paddle tilt, velocity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labkit Communications</td>
<td></td>
<td>Start</td>
<td>Refine Serial Module</td>
<td>Multiplayer Works</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?