



Virtual Ping Pong



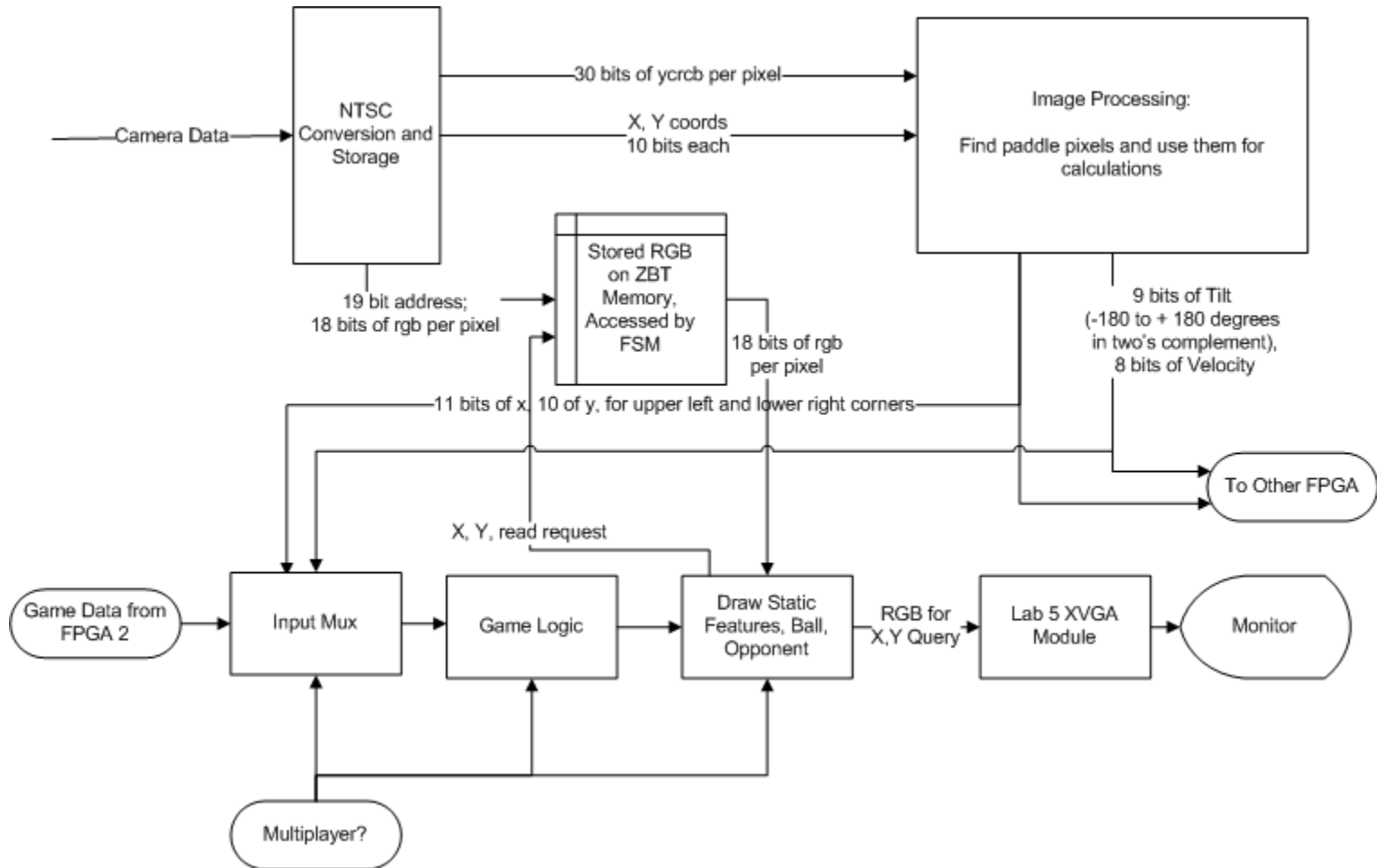
Zach Clifford, Mark Stevens

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



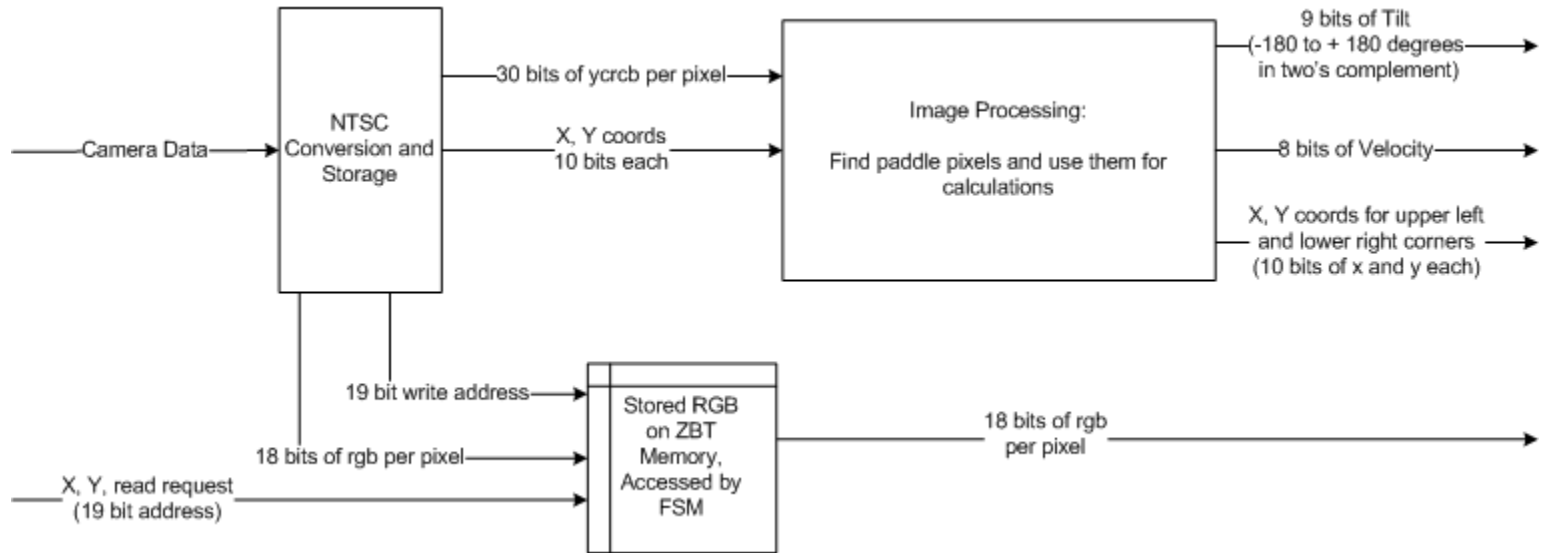
Technical Overview



Virtual Ping Pong Vision

Mark Stevens

Vision Overview



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

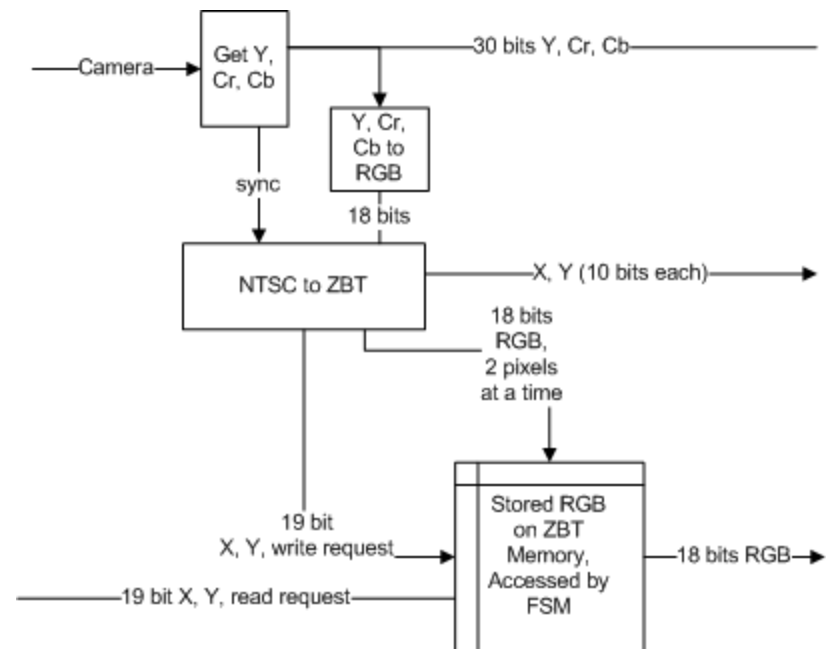
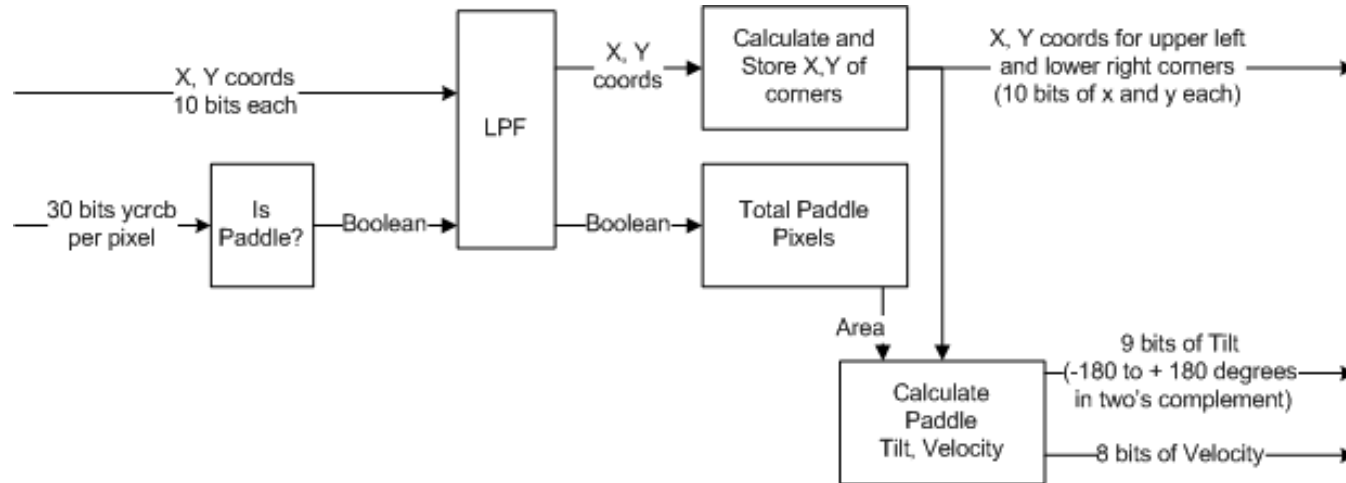


Image Processing



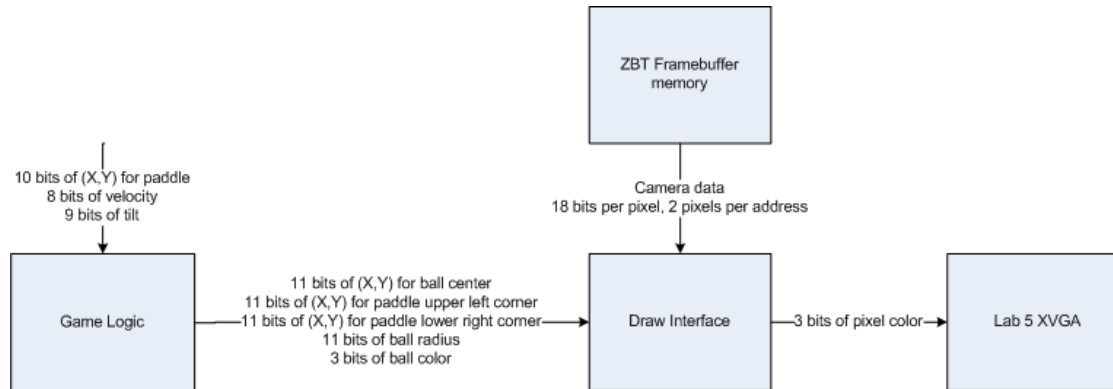
- ▶ Filter incoming data based on Y, Cr, Cb thresholds
 - ▶ Filter out noise
 - ▶ Output coordinates of corners and total area once per frame
 - ▶ Use relative positions of coordinates to calculate paddle tilt and velocity
-



Game Logic and Output

Zach Clifford

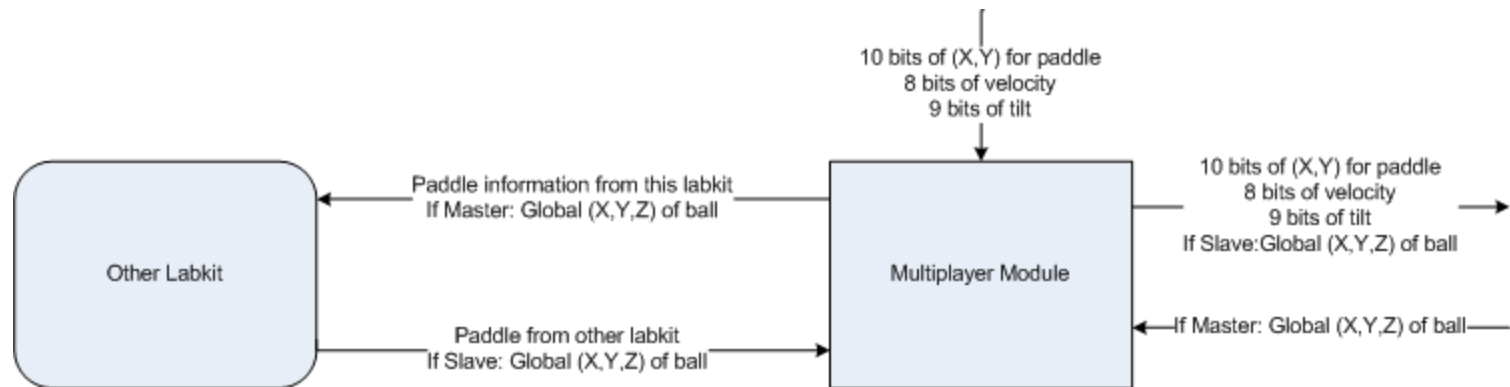
Output Stage



- ▶ 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
-



Schedule

Week of	11/12	11/19	11/26	12/3	12/10
Display Video from Camera	Working in grayscale	Refine blocks		Store RGB instead of grayscale	
Detect and Filter Color	Working	Refine			
Calculate positions in image	Start	Complete			
Calculate Tilt, velocity		Start	Complete		
Drawing Interface	Working				
Physics, Collisions	Works w/o paddle	Add paddle position	Add paddle tilt, velocity		
Labkit Communications	Start	Refine Serial Module		Multiplayer Works	

Questions?

