



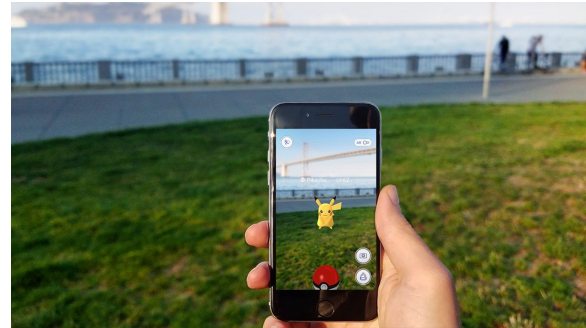
# Motion Capturing System for Game Control: COD Edition

Emanuel Perez  
Ziwen Jiang



# OVERVIEW

Virtual entertainment, such as VR and AR, has been a rising trend in technology industry. Interactive gaming attracts great attention.



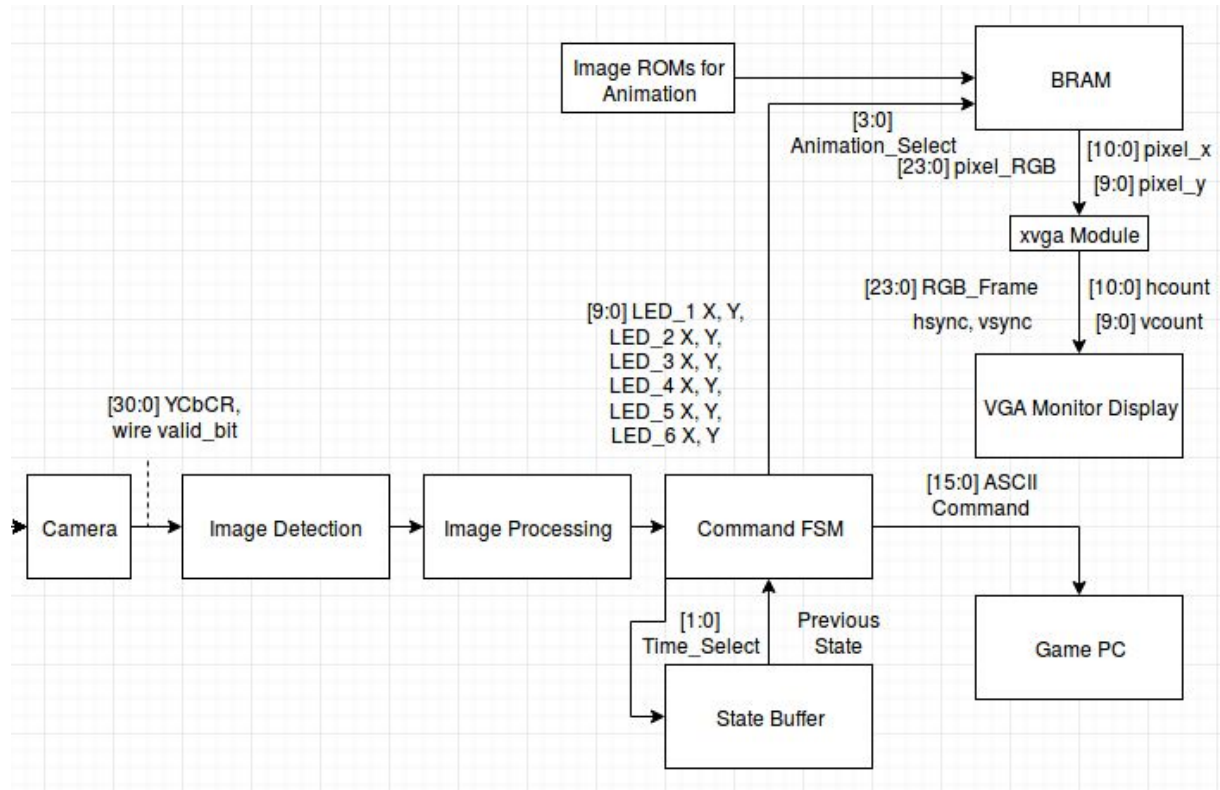
However...



Why stick when you have a suit?

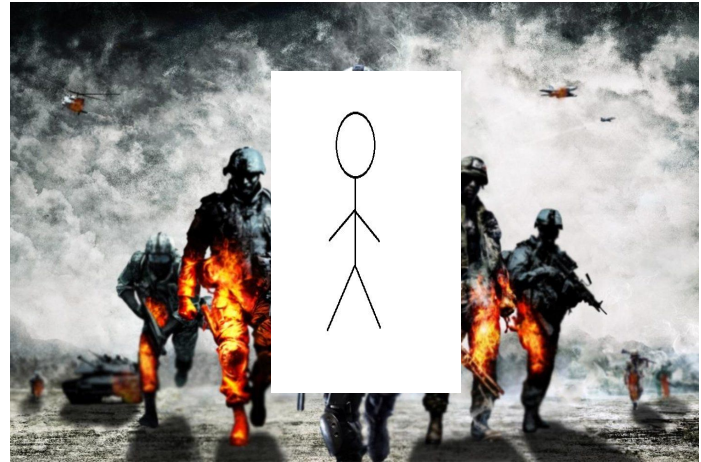
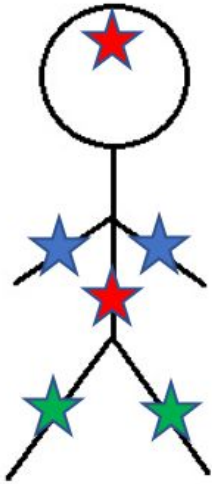


# BLOCK DIAGRAM



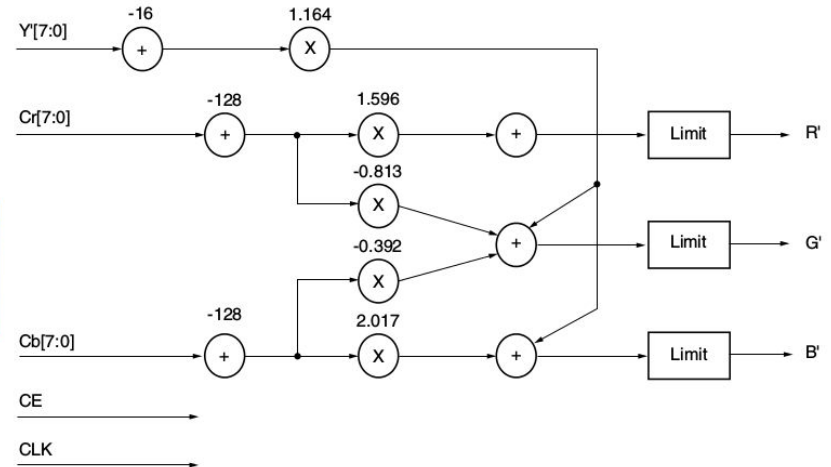
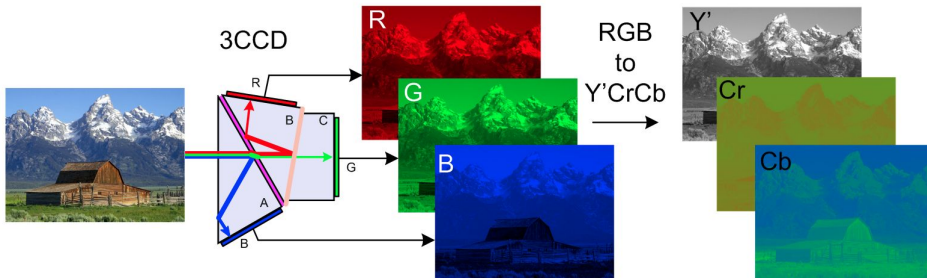
# PHYSICAL SETUP

- NTSC Camera
- 3 pairs of Brightly Colored LEDs
- VGA Monitor



# Module 1: Image Detection

- Functionality: Pixel by pixel, converts YCbCr to RGB and classifies color
- Inputs: [7:0] Y, Cb, Cr
- Outputs: [1:0] color
- Relationships
  - $R' = 1.164 ( Y' - 16 ) + 1.596 ( Cr - 128 )$
  - $G' = 1.164 ( Y' - 16 ) - ( 0.813 ) ( Cr - 128 ) - 0.392 ( Cb - 128 )$
  - $B' = 1.164 ( Y' - 16 ) + 2.017 ( Cb - 128 )$



# Module 1: Image Detection

- Functionality: Pixel by pixel, converts YCbCr to RGB and classifies color
- Inputs: [7:0] Y, Cb, Cr
- Outputs: [2:0] color
- Create threshold values so as to distinguish between 3 different LED colors
  - PSEUDO CODE:

```
if(R>R_TH && B <B_TL && G<G_TL) color <= 0; //red
```

```
if(R<R_TL && B >B_TH && G<G_TL) color <= 1; //blue
```

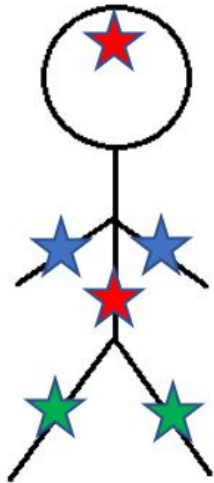
```
if(R<R_TL && B <B_TL && G>G_TH) color <= 2; //green
```

- Need values for TH (high threshold) and TL (low threshold) for each color.



# Module 2: Image processing

- Functionality: Determines outline of colors and calculates centroid location for each LED
- Inputs: [1:0] color
- Outputs: [9:0] LED\_1X, LED\_1Y, ETC.



(X, Y)

# Module 3 Command Converter

- Track the trajectory of each LED marker by storing x, y coordinates across certain time frame in BRAM
- Compare the position change to determine the movement actions
- We will have different action outputs:
  - Basic: up, down, left, right, jump
  - Stretching Goal: hand waving, kneeling down

# Pseudocode

LED\_1\_x\_dif <= LED\_1\_x\_start - LED\_1\_x\_end

LED\_1\_y\_dif <= LED\_1\_y\_start - LED\_1\_y\_end

...

If (LED\_1\_y\_dif > jump\_thresh)

    Action <= jump;

Else If (LED\_3\_y\_start < LED\_2\_y\_start && LED\_3\_y\_end > LED\_2\_y\_end)

    Action <= left\_hand\_waving

**More complex actions will be decided based on the relative positions change between different markers.**

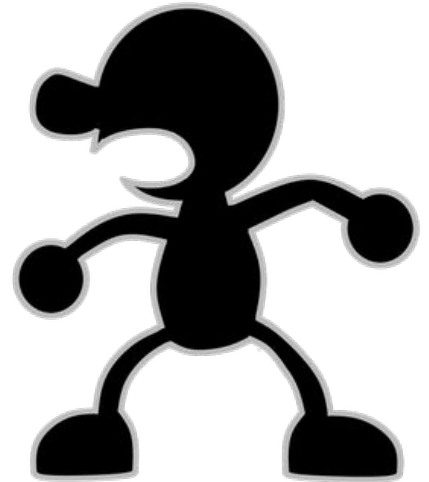
# Module 4 Video Display

There will be two kinds of displays:

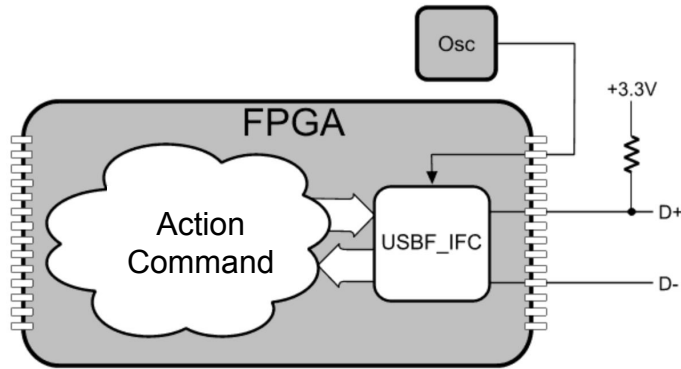
1. Game character display on monitor through VGA
2. Game on external computer controlled by FPGA

# Game Character Display

- Image ROMs that composes the animation will be preloaded into BRAM
- Video display module will select which set of images to display based on the action command from Command Converter Module
- Xvga module will convert to the VGA signal for display



# External Game Control



## XESS XSA-3S1000 Board Demo

- USBF\_IFC is the top-level interface component
- USB\_DRVR component within USBF\_IFC is instantiated to act as the transceiver driving the USB differential data lines
- The module sits and waits for tokens and packets to arrive from the host, interrupts the interface when setup requests or data are ready, and sends whatever data is loaded or requested by the interface



# External Game Control

HID Conversion  
through  
Hyperterminal  
Reading



# Timeline

Task/Time	Week 1 [11/6-11/12]	Week 2 [11/13-11/19]	Week 3 [11/20-11/ 26]	Week 4 [11/27-12/3]	Week 5 [12/4-12/10]	Week 6 [12/11-12/17]
Preliminary Investigation	Both					
Module 1	EP	EP				
Module 2		EP	EP			
Integration			Both			
Module 3	ZJ	ZJ				
Module 4		ZJ	ZJ			
Integration 2				Both		
Testing					Both	
Buffer Time/Stretch Goals					Both	
Demo/Final Presentation						Both



# Troubleshooting

- Memory management during command converter and video display
- Noise management during image processing

Questions?