Gesture Recognition
Remote Control

Wolfe Styke
6.111 Final Project
The Gesture Creators: Wands

• Each Wand
  – 1 sphere of LEDs attached to end of wand
  – 1 button to turn on/off LEDs

• Two Wands
  – different color LEDs for more gestures

• Set list of gestures which will be understood by labkit hardware
The Gesture Interpreter: Labkit (Part 1)

Purpose: To allow user to create gestures with wands that create corresponding changes in the state of a device normally controlled by IR remotes.

• Input:
  – A ntsc video camera

• Output:
  – infrared (IR) transmitter for controlling external devices (tv)

• Internals:
  – ZBT memory:
    • to store incoming frames from camera
  – Image Filters:
    • to find presence and location of each wand in frames
  – Gesture Identification:
    • to determine gesture based on samples of wand movement
  – IR Signal Finder:
    • to look up the corresponding code for each gesture identified
  – IR Signal Generator
    • to create the IR signal sent to the IR transmitter output
The Gesture Interpreter: Labkit (Part 2) (time permitting)

Purpose: Allow device to be reprogrammed to function like any remote control.

• Input:
  – IR Receiver
  – A labkit button

• Internals:
  – IR signal analyzer and storage
    • to analyze a pre-determined sequence of signals from a new IR remote control
    • to store this set of signals in the BRAM memory
All modules clocked with the 27MHz clock of the ad7185.

Block Diagram for Gesture Recognition Remote Control

Wolfe Styke, 6.111
Video Interpretation (Pt. 1)

• The ad7185 module
  – Clock: 27 MHz – used for all modules
  – Every four clock cycles:
    • Video info sent in order: Cr, Y, Cb, Y
      – Luminance sampling rate: 13.5 MHz
      – Color sampling rate: 6.5 MHz
  – Will be reused from existing module designs.
ZBT Memory

- ZBT Memory
  - controlled via frame storage and retrieval module
  - takes in frames from video interpreter
  - passes frames out to the filters
Filters

• Filter (input: frame)
  >> determines if and where a wand is lit up <<
  – luminance -> detect presence of wand
  – color -> detect which wand is gesturing
  – output:
    • coordinates of pixels where wand being tracked resides
Center of Mass

• Input: Coordinates of pixels where wand is found

• Internals:
  – Detect presence of wand by finding a jump in total number of pixels of a certain color or luminance
  – Find average hcount and vcount of pixels noticed
  – Implementation Idea:
    • stop analyzing the data for a frame after the line with the largest number of pixels signifying a wand are detected, then use that line’s hcount and the average the vcounts showing pixels for that line

• Output:
  – Coordinates of wand’s center
  – Ready signal signifying presence of a wand
Gesture Generator

• Input (from each wand identifier):
  – ready signal (signifying presence of wand)
  – coordinates of wand

• Internals:
  – Sample coordinates of each wand roughly 30 times a sec.
  – Classify movements of each wand between samples in terms of simple directions (up, down, left, right)
  – Map list (or 2 lists from 2 wands) of directions to a pattern associated with a gesture

• Output:
  – Gesture chosen from a set list of gestures
    • 1 Wand (either wand a or wand b)
      – aUp, aDown, aLeft, aRight,
      – bUp, bDown, bLeft, bRight
    • 2Wands
      – Both (Up, Down, Left, Right)
      – Traveling (away and towards) each other
Gesture to Signal

• Input:
  – Gesture
  – Reset button

• Internals
  – BRAM memory to store IR signal data for each gesture
  – A lookup table for picking the address in the BRAM to reference for a given signal
  – A Bram Control module for giving control of the BRAM to the second part of the system (adding new signal data) when used

• Output:
  – IR Signal Data
New IR Remote Functionality

• Input:
  – IR Receiver
  – Switch on Labkit (Reprogram)

• Internals:
  – Receiving a specific sequence of signals (from IR remote)
    • ie (Vol. Up, Down, Chan. Up, Down, etc…)
  – Stores each signal in BRAM memory through gaining control of the memory via the BRAM Control module

• Output (BRAM Memory update)
  – Address of BRAM memory to update
  – Signal to update memory with
  – write enable, which also signifies the gaining of control of the BRAM memory
Signal Generator

• Input:
  – Signal Data

• Internals:
  – convert signal data to wire output highs and lows recognizable to IR Transmitter
  – emulate timing pattern for ones and zeros according to the remote control pattern

• Output:
  – wire user1[2] to IR transmitter to device controlled (TV)
# Timeline

<table>
<thead>
<tr>
<th>Item To Complete</th>
<th>Date to Complete By</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZBT Memory</td>
<td>11/17</td>
<td></td>
</tr>
<tr>
<td>Filters</td>
<td>11/19</td>
<td></td>
</tr>
<tr>
<td>Center of Mass</td>
<td>11/22</td>
<td></td>
</tr>
<tr>
<td>Gesture Generator</td>
<td>11/24</td>
<td></td>
</tr>
<tr>
<td>Gesture to Signal</td>
<td>11/29</td>
<td></td>
</tr>
<tr>
<td>New IR Remote Functionality</td>
<td>If time</td>
<td>Extra Functionality</td>
</tr>
<tr>
<td>Signal Generator</td>
<td>12/1</td>
<td></td>
</tr>
</tbody>
</table>