Interactive Video Game

Tim Li

Leo Hwang
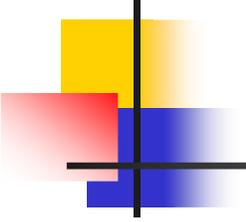
The Project

- A video game where the user interacts by making gestures into a camera
- Video game elements are superimposed onto the person's outline, then displayed
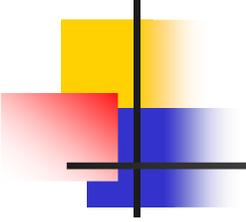
Initial Goals

1. Video output for display
2. Video input for user control
3. An outline detection scheme to capture the user's actions
4. A game with which the user can interact
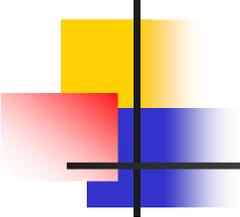
Results

- Similar to initial goals, but
  - User holds a coloured flag instead of making gestures
  - Video output is not as clean as we would want

Final Design
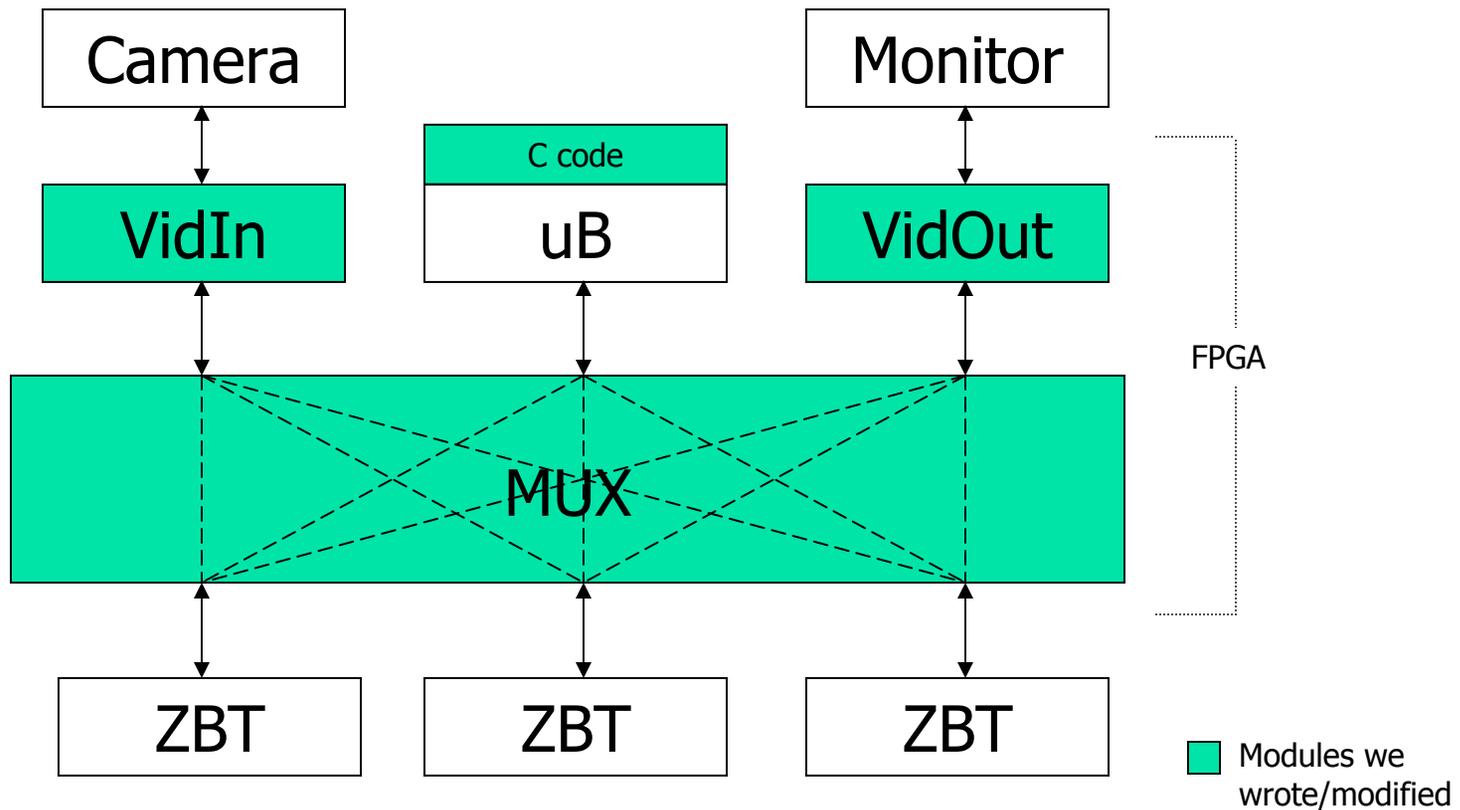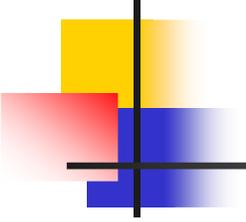
Project Blocks

- **Video Output (hardware)**
  - From Xilinx Multimedia Board Example
  - Modifed to work in XPS
- **Video Input (hardware)**
  - Took Lesley's video_input block and the standalone ZBT controller
  - Made a Verilog piece to connect the two parts and deciphered the signals correctly
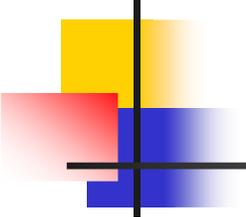
Project Blocks (cont.)

- ZBT bank switcher (hardware+software)
  - Custom block, acts as a big MUX between three ZBT RAM chips
  - Rotates modules' (camera/processing/monitor) connections between the memories
  - Switch buffers when a signal is toggled from the Microblaze (via GPIO)
  - Switching is aligned to vertical blanking interval
  - Good, because modules can work without modification and in parallel
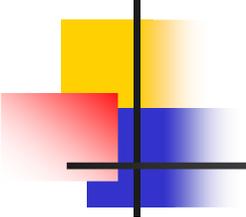
Project Blocks (cont.)

- Detecting user's motion (software)
  - User holds and moves a red flag
  - Implemented a detection algorithm which gives a rough location of the red flag
- Video Game (software)
  - Takes input from the user detection block
  - Draws elements into the ZBT memory (via EMC)
  - Rotates memory buffers by toggling the ZBT bank switch block
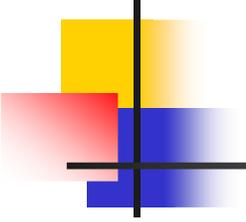
Design Process

- Try to get blocks to work on their own
  - Integrate by adding bit by bit together
  - Test in between each addition

1. Video Output was working first in ISE
2. Modified so it would work in XPS
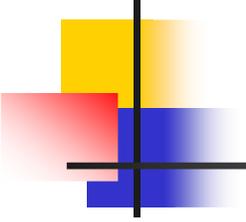
# Design Process (cont.)

3. Added a simple version of the ZBT bank switcher

4. Video Input working in ISE

5. Modified to work in XPS (like for video out)

6. Extended the ZBT bank switcher to accommodate for video input

Design Process Illustration

Video Out

Test

Video Out ISE

Modify and Test

Video Out XPS

2xZBT MUX

Video In

Test

Video In ISE

Modify and Test

Video In XPS

Combine and Test

uB controlled Video Out

Modify MUX into 3x and Test

uB controlled Video In and Out

Design Process Illustration (cont.)

```
          │
          ▼
┌─────────────────────────────┐     ┌──────────────────────────┐
│ uB controlled Video In and Out │     │ User Detection Algorithm │
└─────────────────────────────┘     └──────────────────────────┘
          │                                   │
          │                         Combine and Test
          ▼                                   ▼
┌─────────────────────────────┐
│ uB controlled Video In and Out │
│      with User Detection       │     ┌──────────────────────────┐
└─────────────────────────────┘     │        Video Game         │
          │                         └──────────────────────────┘
          │                                   │
          │                         Combine and Test
          ▼                                   ▼
┌─────────────────────────────┐
│     Interactive Video Game     │
└─────────────────────────────┘
```
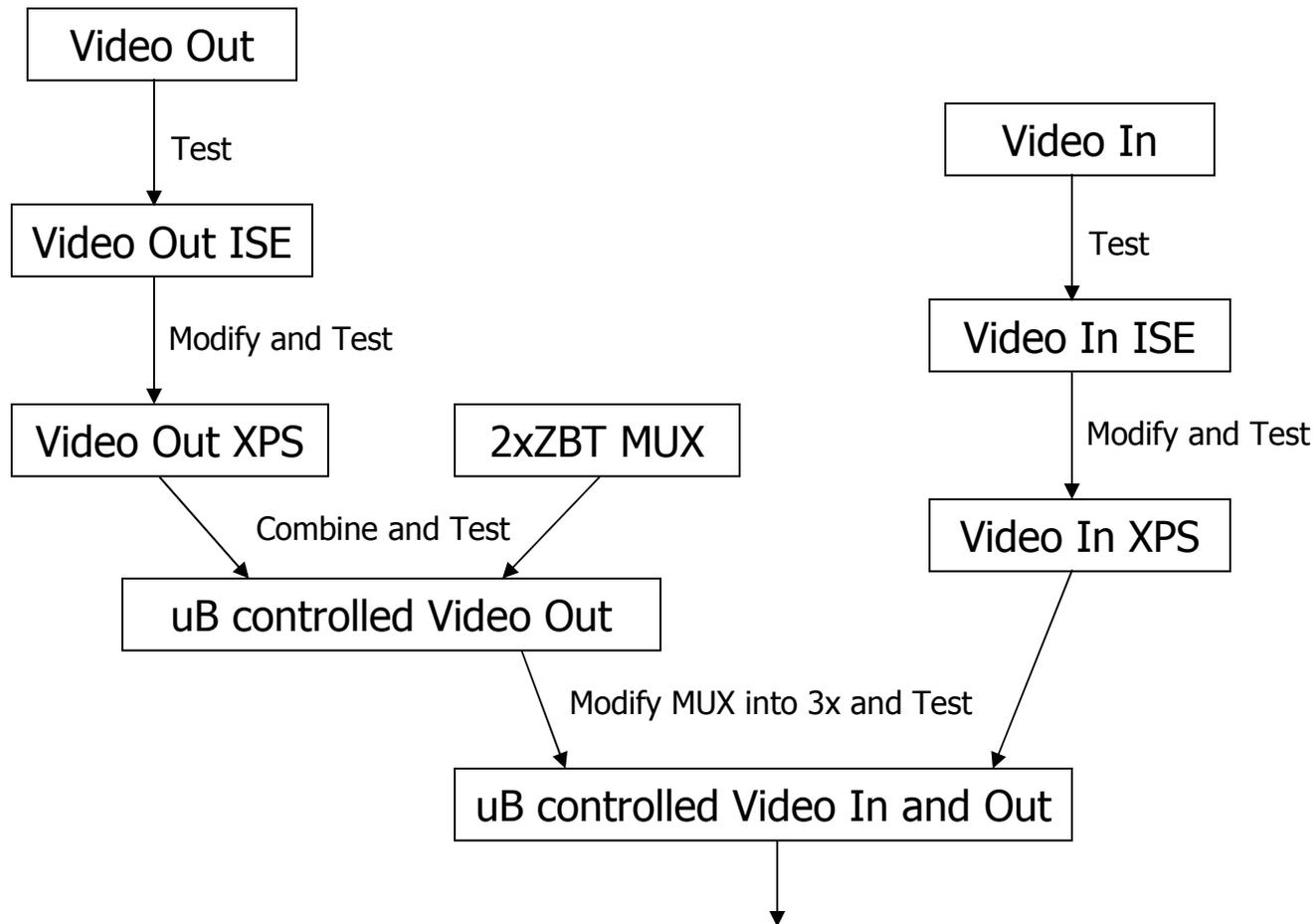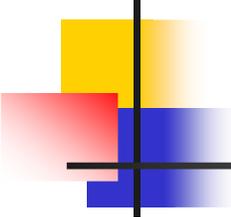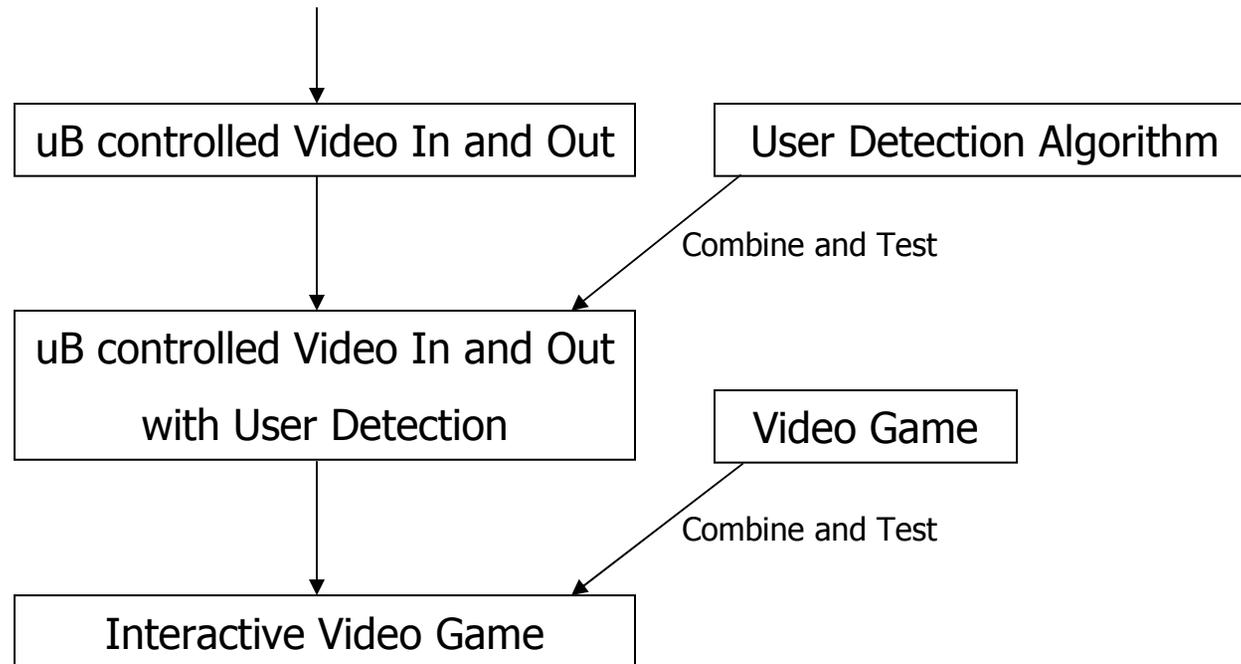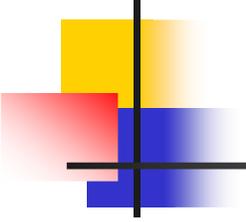
What We Learned

- How to use Xilinx tools and the organization of files

- Tools have room for a lot of improvement

- NTSC format

- Some Verilog

- ZBT memory control
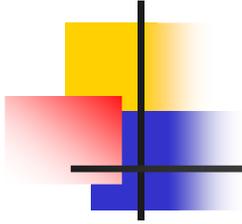
Demonstration
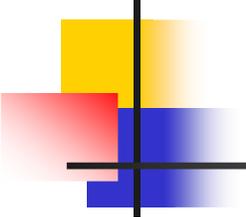
Problems Encountered

- Video output and input cores working in ISE, but cannot import into XPS
- Problems with video output showing random dots
- Video input showing false colours at edges
- Program data segment getting overwritten
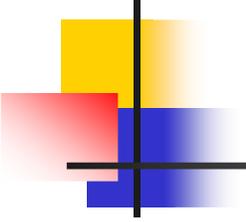
Video In/Out Cores and XPS

- Want to use cores in XPS, but get many errors
- Solutions:
  - Convert HDL module names to lowercase
  - Replace OBUFs with assign
  - Separate IOBUFs into separate _I, _O, _T wires
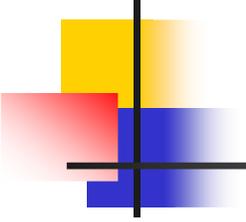
Dots in Video Output

- Dots would appear in the video output, when controlled by Microblaze program
- Solution:
  - Our output frequency was slightly off the specs, 40.5Hz instead of 40Hz -> attached DCM multiplier/divider to 50MHz clock
  - Resulted in less dots appearing
  - Also synchronized drawing program to VSYNC signal
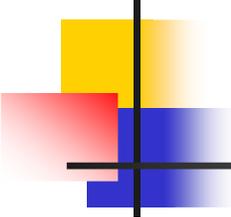
Video Input and False Colours

- False colours esp. around edges
- Possible Solution:
  - Modify YCbCr -> RGB module
  - But this was not pursued, in order to work on more critical pieces
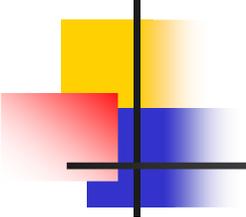
Memory Getting Overwritten

- Using the ZBT EMC, variables in a program would be reset while executing
- Solution:
  - Wizard project creation placed data segment into ZBT memory, which was cleared on commands like ClearScreen()
  - Forced the placement of text/data/stack/etc into local memory by modifying <program>LinkScr