Lab Lecture 11

Tips and Tricks

11/7/2003

Greg Gibeling

Today

- Debugging
- Reset & Debouncing
- Announcements
- Signal Conditioning
- Timing Analyzer
  - Constraints
  - Kramnik

Debugging (1)

- Debugging Algorithm
  - Hypothesis: What’s broken?
  - Control: Give it controlled test inputs
  - Expected Output: What SHOULD it do?
  - Observe: Did it work right?
  - If it broke: THAT’S GREAT!
    - If we can’t break anything like this then the project must be working...

Debugging (2)

- Don’t debug randomly
  - Just changing things at random often makes things look fixed
  - It won’t really help
  - Debug systematically
  - Your first design may be the best
  - “1000 CS150 students at a 1000 typewriters...”
  - What can you do?

Debugging (3)

- High Level Debugging
- Localize the problem
  - Is it the decoder? The sync fifo? SDRAM? Async? Encoder?
  - Test Patterns
    - Lets you easily isolate the broken component
  - Freeze Frame
    - Divides project in half and lets you check both halves

Debugging (4)

- Simulate the broken component(s)
  - Writing test benches takes less time than sitting around wondering why its broken
  - Everyone hates writing testbenches
    - (Even me)
    - Get used to it
  - What if the simulation works?
    - LOGIC ANALYZER!
Debugging (5)
- Using the logic analyzer
  - The most reliable tool you have
    - When used properly
  - Use the triggers effectively
    - Trigger on recurring sequences
    - Trigger on errors
    - An unstable display is useless
  - Compare logic analyzer to simulation

Debugging (6)
- Your best debugging tool is logic
  - If your decoder, FIFOs and SDRAM work then what's probably broken?
- Question all your assumptions!
  - Just because you think it's true doesn't mean it is
  - 90% of debugging time is wasted debugging the wrong problem otherwise
  - Given solutions and modules may not work the way you expect!

Debugging (7)
- Before you change anything
  - Understand exactly what the problem is
  - Find an efficient solution
  - Evaluate alternative solutions
- After the change
  - Fixes may make things worse sometimes
  - May uncover a second bug
  - May be an incorrect fix
  - Repeat the debugging process

Debugging (8)
- Ask around
  - Someone else may have had the same bug
  - They'll probably at least know about where the problem is
  - Different bugs may produce the same results
- TAs
  - The TAs know common problems
  - We've also made a lot of the mistakes

Reset & Debouncing (1)
- A lot of people have reset problems
  - Is it debounced?
  - Why isn't debouncing enough?
  - What's the solution?
    - Make it longer!

Reset & Debouncing (2)
- Picture sync problems...
  - Why does this happen?
    - Decoder, SDRAM and Encoder must be synced on reset
**Reset & Debouncing (2)**
- How do we sync everything?
  - SDRAM doesn't write when Sync FIFO is over half empty (data_count)
  - SDRAM reads can start before the first write. Why?
  - Encoder doesn't start until Async FIFO is half full for the first time
- Questions?

**Announcements**
- Checkpoint3 Due 11/4 @ 8pm
  - You may checkoff through 4pm Today 11/7 for 50% credit
- Checkpoint4 Due Monday 11/10
  - We'd be happy to check you off early
- NO CHECKPOINT5!
- Project Due Wed 11/26
  - +5 Bonus for Early Completion on 11/21

**Signal Conditioning (1)**
- Off-by-a-cycle Errors
  - Shorten a Pulse
  - Lengthen a Pulse
  - Shift a Pulse
  - Remember the Edge Detector?
    - It's something like that...

**Signal Conditioning (2)**
- Shorten a Pulse
  - 5 Cycles -> 4 Cycles
    - Input
    - ![Diagram](image1)
    - Output
    - Any guesses?
    - What if we delayed the input?

**Signal Conditioning (3)**
- Shorten a Pulse
  - Input
  - ![Diagram](image2)
  - Output = In & In_Delayed

**Signal Conditioning (4)**
- Lengthen a Pulse
  - Input
  - ![Diagram](image3)
  - Output = In | In_Delayed
Timing Analyzer (1)
- Timing Constraints in Synplify
  - Xilinx will attempt to match them
  - Will tell you if it fails
  - They make PAR run slower
  - A constraint will not make your circuit faster
- Better to just let things PAR
- Check the timing when we’re done

Timing Analyzer (2)
- “Analyze Post Place and Route Static Timing (Timing Analyzer)”
- Implement Design -> Place & Route -> Generate Post Place and Route Static Timing
- This will tell you your minimum period
- If its too big then what?
  - Simplify your circuit
  - Constraints probably won’t do it

Kramnik
- Allow log in from off campus
- You can run ModelSim from home
- Not much point to Synplify/Xilinx
  - https://iesg.eecs.berkeley.edu/remote
- Log into kramnik.eecs.berkeley.edu
- Demo...

And now...
- We will check off Checkpoint4
  - You have till Monday for full credit
- Checkpoint3 for 50% credit
- Questions?
  - Stick around and ask
  - Stick around and listen, you might hear something very useful.