ECE550 HW5

Cache Controller
Schedule

- Homework Intro
- How to get started
- Timing Requirements
- Debug with the hardware interface
- Debug with SignalTap
Given Cache Organization:

- 32-bit Address
- Direct-mapped (1-way/associativity=1)
- 64B cache block (6 bits for byte select/offset)
- 128 sets (7 bits for index)
- \( \# \text{ of Tag bits} = 32 - 6 - 7 = 19 \)
  
  - tag.vhd also includes dirty and valid bits, making tag data 21 bits
Homework Introduction

Cache controller (**cache.vhd**) needs to:

- Initializes Cache, Tells Tester (**cache_sim.vhd**) whenever it’s ready.
- Accepts load/store requests, checks tags/valid bits and then...
Homework Introduction

Cache Hit/Miss Handling


- **Clean Miss** -> Replace cache block with new data, then returns/updates

- **Dirty Miss (The block to be replaced has new information)**
  - 1. Cast out old dirty block
  - 2. Replace with new block
  - 3. Returns data/Updates data
What do I do….?

- Start Early
- Read the homework write up carefully
- Design FSM (for loads only) → **Remember VGA Controller?**
- Implement FSM (for loads only) → **Feel free to reuser your own counter/adder**...
- Test and Make Sure Cache Controller Can Handle Loads
- Updates FSM to Support Loads and Stores
- Retest
- Improve and Optimize for Extra Credits & Retest
Timing Requirements

Cache Controller Need to Operate at a Minimum of 75MHz Clock

How to find out if the design has met the requirement?

- Set PLL output clock to designed frequency

Compiles and see if the target frequency passed time requirements
Timing Requirements

How to make your design faster?

- Use TimeQuest Analyzer to Look for **Critical Path**
- Simplify Logics on Critical Path
- Use Better State Encoding (e.g. One Hot)
- etc....
Debugging ...........

ModelSim!

- Easiest way to find out what’s wrong
Debugging

SignalTap

- Look at signals in hardware
Debugging

Hardware Interface

→ matter of last resort, not recommended, but still useful


Q & A