Project Introduction & SMIPS150

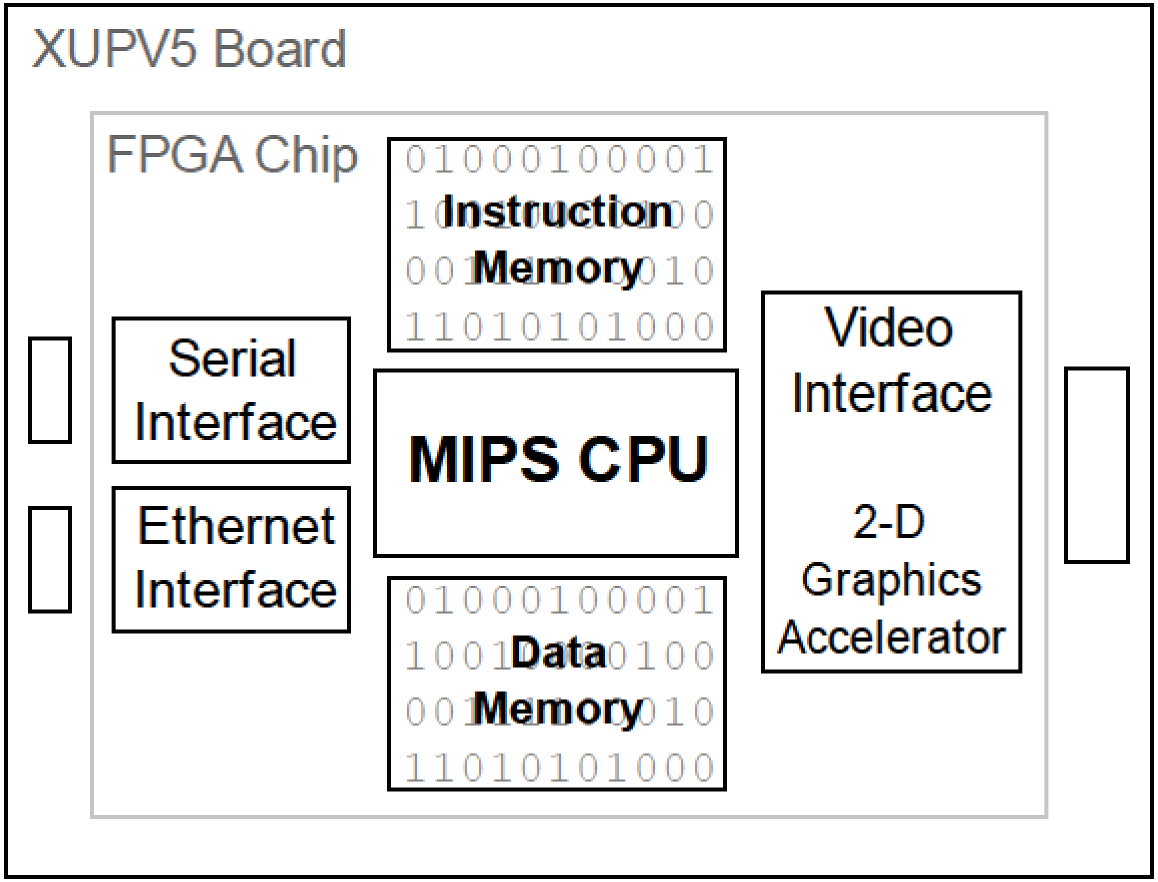
UCB EECS150 Spring 2010

Lab Lecture #6

Agenda

- Project Overview
- Project Logistics
- Administrative Info

CS150 Spring 2010 Project: SMIPS150



Project Overview 1

- Components
  - ‘S’MIPS 3-stage CPU (w/ a compiler)
  - Serial & Ethernet interfaces
  - Frame buffer in off-chip SRAM
  - Graphics (triangle) accelerator engine

Project Overview 2

- Checkpoint 1: SMIPS Processor
  - “You will implement a subset of the MIPS ISA”
  - In a 3-stage pipeline
  - Will be split over several weeks

More on this later…

Project Overview 3

- Checkpoint 2: I/O Interfaces
  - Serial interface
    - Send commands from terminal (PUTTY) to CPU
    - A proxy shell to your CPU
    - We will give you a basic “boot monitor”
  - Ethernet interface
    - Send code (files), images, video streams, etc
    - Boot monitor will support TFTP out of the box

Project Overview 4

- Checkpoint 3: Frame Buffer in SRAM
  - Eventual goal: show pictures on the TV using DVI
  - Spring 2009: show an *entire* frame in block RAM
  - **Problem:** This uses all of the block RAMs!
    - 768x1024 resolution
    - Small color map (4-bit color base)
  - Spring 2010: store a frame in off-chip SRAM
    - 800x600 resolution
    - 16-bit color (no color map)

Project Overview 5

- Checkpoint 4: Triangle Engine
  - One way to do it…
    - Have your processor perform `sw`’s to draw dots
    - Processor can’t do other work while drawing
    - Very slow!
  - Another way to do it…
    - Queue up triangle *vertices* using the processor
    - Have the actual draw/fill done by dedicated hardware
    - A simple GPU-like approach, basically

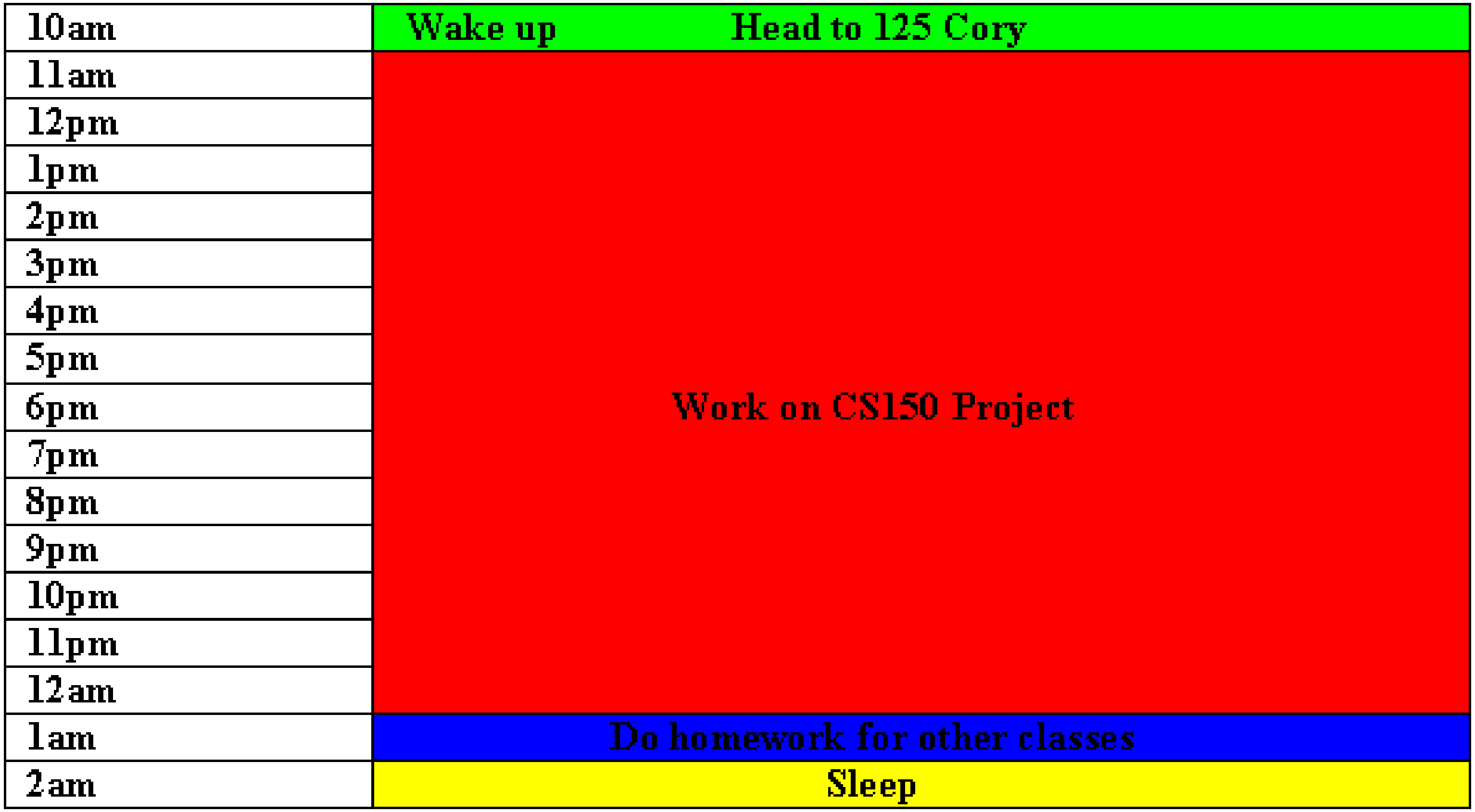
Project Overview 5

This semester’s schedule

2/28-3/6	Lab 5 Due, Checkpoint 1 Design Review
3/7-3/13	Checkpoint 1
3/14-3/20	
3/21-3/27	Spring Break
3/28-4/3	Checkpoint 1 Due, Checkpoint 2
4/4-4/10	Checkpoint 2 Due, Checkpoint 3
4/11-4/17	Checkpoint 3 Due, Checkpoint 4 Design Review
4/18-4/24	Checkpoint 4
4/25-5/1	Checkpoint 4 Due, Checkpoint 5, **Early Check-off**
5/2-5/8	Checkpoint 5 Due, **Regular Check-off**

Project Overview 6

- Courtesy of Chen… (“your everyday schedule for 10 weeks”)



10am	Wake up Head to 125 Cory
11am	Work on CS150 Project
12pm	
1pm	
2pm	
3pm	
4pm	
5pm	
6pm	
7pm	
8pm	
9pm	
10pm	
11pm	
12am	
1am	Do homework for other classes
2am	Sleep

Project Logistics 1

- Partners
  - You should have a partner from lab 5
  - *Segway:* Does anyone **NOT** have a partner for lab 5?
  - You can start with a new partner for the project
  - Or keep the one you currently have…
  - Regardless, you **must have** a partner
  - Exception: “odd numbered” student

Project Logistics 2

- Partners (continued)
  - Golden rule: “ALWAYS expect your partner to accomplish absolutely nothing, but DO expect your partner to cover for you when you are slammed by 3 midterms the day before check-off”
  - The above is a *useful* working relationship
  - Where can this go wrong?

    Don’t take the same classes!

Project Logistics 3

- Design Reviews
  - Lab 5 was practice
  - During the project they count as homeworks
- This last week…
  - People had questions
  - People didn’t have design documents ready
  - Design documents were scribbles “on a napkin”

Project Logistics 4

- Project Collaboration
  - Collaborate! That is what this lab is for.
  - Don’t share code (except with your partner)
- Collaboration extends to…
  - The newsgroup
  - Discussion section

Project Logistics 5

- Project Submission
  - Everything goes through SVN
  - You and your partner will have a shared repository
  - Check in code regularly
    - Reverting might actually come in handy now…

Administrivia

- Lecture room change!
  - – We are moving to 306 Soda
  - – Effective immediately
- Computer logins (after lab lecture)

SMIPS150 1

- 3-stage pipeline
- Open-ended design
  - We impose the ISA
  - You come up with the design
- The 3-stage constraint is there for a reason…
- Feeling uneasy? Collaborate!
- Don’t look at the book’s implementation
  - 5 stages
  - Not our ISA anyway

SMIPS150 2

- Restrictions
  - Must be a 3-stage pipe
  - CPI = 1 (i.e. no stalling)
  - Must run @ 100 Mhz

- The less area, the better (you will be graded on this)
- No extra credit for running @ > 100 Mhz

SMIPS150 3

- Testing
  - Compiler will come live at some point
  - But you don't want to test with that…
  - SPIM / MARS
    - Build basic / single-instruction tests
    - Once each instruction works alone, test whole programs
    - Don't test in hardware
    - Use Modelsim as much as possible!

SMIPS150 4

- Parting words
  - More details will be released in a spec (over the weekend)
  - For now, finish lab 5

- You have 3 weeks to get SMIPS working
- Make it work and make it work well!

Acknowledgements & Contributors

Slides developed by Chris Fletcher & John Wawrzynek (2/2010).

This work is based in part on slides by:

Chen Sun (2008-2009)

Chris Fletcher (2008-2009)

Greg Gibeling (2003-2005)

This work has been used by the following courses:

– UC Berkeley CS150 (Spring 2010): Components and Design Techniques for Digital Systems