

Lecture Schedule  
CPSC / ECE 3710, Fall 2006

This class consists of designing a computer system consisting of a CPU, a memory subsystem, and an I/O system. The lecture plans are as below. If we do not finish covering all the topics slated for a lecture, we will leave it up to you to study the remaining topics whose handouts will be kept online - unless you *ask* for certain things to be discussed. The schedule below is guaranteed to slip, but we have some slack built in!

We shall conduct designs using the ISE tool suite for design capture, and ModelSim for simulation. A link to version 7.1 has been kept on the class webpage. We shall make our comments with respect to this version. Some of you may work with later versions; I don't know much about these more recent versions (suffices to say that they occupy more disk space).

We shall also be using the VHDL hardware description language for most of our work. You may conduct designs entirely as schematics, as VHDL + Schematics, or entirely as VHDL. We may occasionally require schematics to be produced from your existing VHDL descriptions. Generating such schematics is easy and automatic, using ISE (if in doubt, ask the TAs).

Lecture 1:8-24 provides an overview of the CPU of choice, namely a subset of CR-16. It also provides points for expanding the instruction set, adding features such as interrupts, etc. We suggest that you develop the processor in three stages:

- Develop a thin slice
- Finish the required baseline instruction set
- Finally add enhancements according to our rough guidelines (feel free to propose your own additions)

Lecture 2:8-31 provides an overview of the memory subsystems used. We will discuss

- SDRAM and its controller
- Block RAMs

Lecture 3:9-7 provides an overview of the I/O systems used. These include

- VGA,
- PS2, and
- UART

Lecture 4:9-14 provides an overview of the assembler and simulator that are required. We will also discuss upload and download methods. The items specifically discussed are

1. Simple assembler syntax and FAQs
2. Simulator specification
3. Hex load formats
  - (a) For memory upload / download

(b) Others?

Lecture 5:9-21 helps you plan and organize your design activities. It contains these details:

1. Deriving a microarchitecture
2. Hardware flowcharts (planning micro-ops, merging them for max bus utilization, parallelism, and cycle count reduction)
3. Turning HW flowcharts into state machines
4. Clocking and timing disciplines
  - (a) Latches and flip-flops
  - (b) Single- and dual-edge clocking (prefer the former)
  - (c) Standard asynchronous handshakes
  - (d) Other ad-hoc asynchronous and timed handshakes and interactions

Lecture 6:9-28 helps you finish your “thin-slice” milestone.

1. Finish simulator and HW implementation of thin slice using Block RAMs
2. Finish assembler
3. Migrate to SDRAM
4. Show how it can work with the VGA
5. Design “consoles”
  - (a) VGA based
  - (b) LED based

Lecture 7:10-12 talks about design debugging and optimization. It discusses unit and integration testing ideas. It also provides ideas for projects. Finally it provides ideas pertaining to giving successful project demos.

1. Planning subsystems
2. Cost of various shift instructions
  - (a) Combo or sequential (students have latitude here)
3. Interface protocols
4. Unit testing plans
5. Integration testing plans

Lecture 8:10-19 discusses advanced topics, including:

1. Lectures on modern CPUs

2. Multicores, Transaction Memories, and Programming
3. Multiprocessors
4. Fancy memory subsystems
5. I/O systems such as PCI Express and Infiniband

Some general items of wisdom from past experience include these:

1. Assess the “hardest” thing to pull off. Get it under control sufficiently early (meaning, do not get bogged down by it, but do not put it off either).
2. Leave enough time to play with and enjoy your CPU. Therefore, set a deadline for yourselves that is at least a few weeks before the final deadline.