Lab Lecture for Lab #5
Caches and Real Memory

Jack Kang

cs152-ta@imail.eecs.berkeley.edu

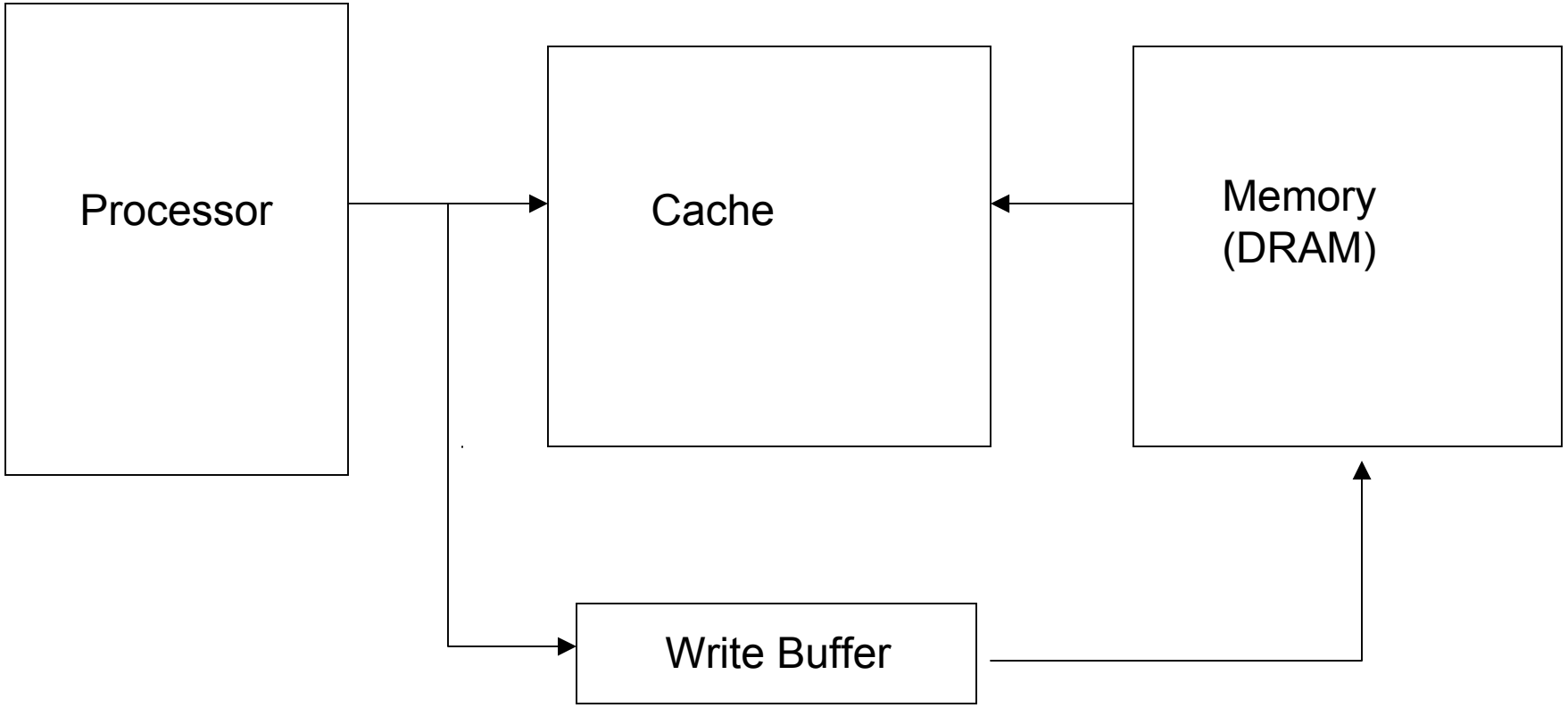
DRAM Controller

- We are giving you a working module that will handle initialization, do 1 word burst, 1 word reads, and auto refresh.
- Do what you wish with this module.
- Thank Professor Patterson for this.
  - He's probably watching the webcast as we speak.
- You are still responsible for reading and understanding the RAM.
- Simulation file called mt48lc8m16a2.v

Write Buffer

- If we have a write-through scheme, how slow are writes to cache??
  - We're forced to stall, even though there are no dependencies in the pipeline!
- Let's put a write buffer in.
  - Store the outstanding writes to the buffer.
  - Write it to the DRAM while we have time.
  - Still, it's possible for the Write Buffer to be full. Typical Size of write buffer: 4.
  - Works fine if: Store Frequency << 1/DRAM write cycle

Write Buffer



Arbiter

- Only one thing can use the DRAM at once.
  - But we have 3 things that may make requests:
    - Inst Mem
    - Data Mem
    - Write Buffer
  - The arbiter must decide who gets to go, and in what order.
  - Probably should be a FSM…
  - Be Careful!!! Think it through! This is difficult!

Memory Mapped IO

- Read the spec!
  - Make sure that iooutput.trace and ioinput.trace both work
  - The same module should work both in simulation and synthesis
- One new thing:
  - Cycle counter, can be read from 0xFFFFFFFC

Lvl 0 Boot

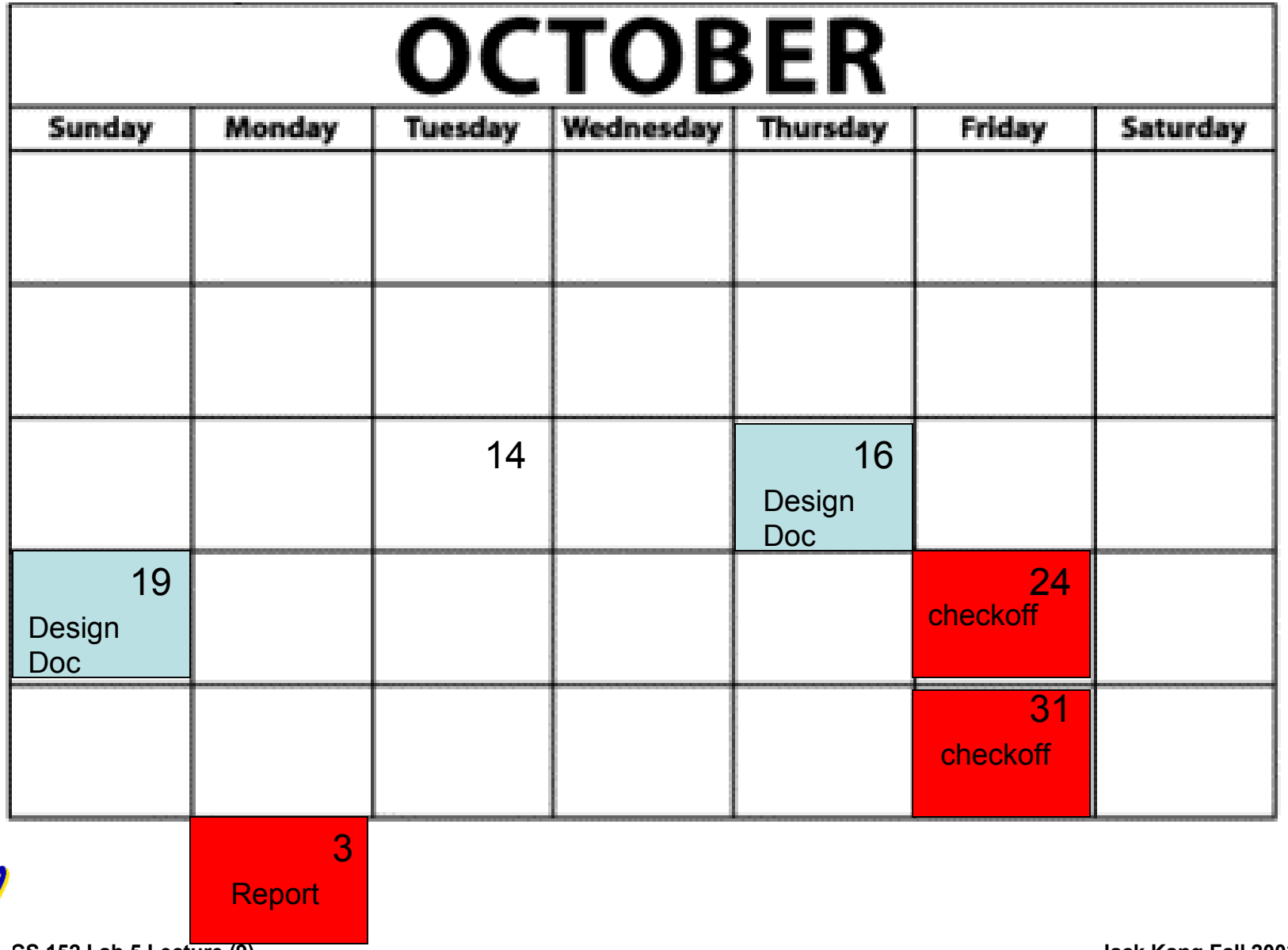
- Set of instructions that loads in your instructions and data.
- Use mipsasm and mipsconvert as normal, but at the very top of your instructions, into your contents file, you need to add the length and the address before running boardRAMcreatewin

  Length (number of instructions– in HEX)
  Address (starting address of your code—probably 0x00000000)
  Inst1
  Inst2
  Inst3
  …
- Think about how the lvl 0 Boot loads in your values. Where does your boardRAM module fit in the processor?

Extra Credit

- Extra Credit means you don't have to do it.
- If you're doing a burst, it's probably a good idea to start off designing that way
- Used to be able to do a write-back cache. Too difficult, so we are saving it for the final project.
- Only working processors will get extra credit!

Schedule



Tips on avoiding another Lab 4

- Implementation is fast, debugging is slow
  - Set aside more time than you think for debugging. Use the trace outputs!
  - Consider writing tests before your implementation is done—it will help you organize your design.
- Understand the Specifications
  - If unclear, ask! Memio module is a good example from lab 4.
- We are working on getting more computers