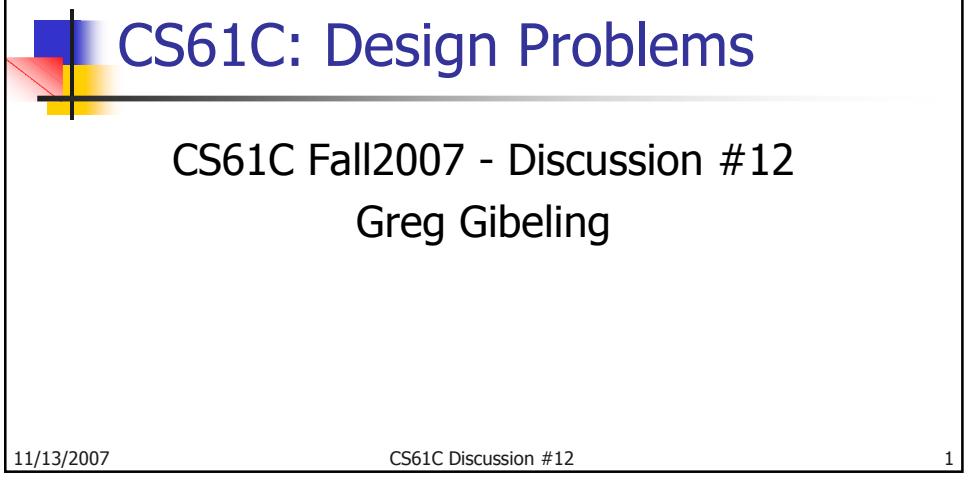
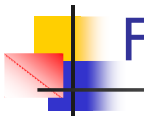


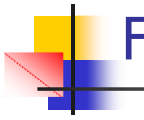
Problems & Rules

- Problems
  - FSM & Logic
    - FSMs/Basic
    - FSMs/FIFO Design
    - Non-CPU SDS
  - Processors & ISAs
    - Memory-To-Memory Processor
    - Microcode
    - Out-Of-Order
    - No-branch processor
    - MultiProcessor Locking
  - Stump the TA
- Rules
  - Previous spokespeople: one group
  - Anyone should be able to explain your solution



FSMs/Basic

- Basic Problem
  - Expand your FSM from HW8 to a 5bit output
    - 00001, 00010, 00100, 01000, 10000, 01000…
  - Add an input to reverse the direction
- Results
  - State encoding/State transition diagram
  - Timing Diagram
  - Schematic/Block Diagram
- Advanced
  - Write some Verilog
  - Consider the FSMs/FIFO Design problem



FSMs/FIFO Design

- Basic Problem
  - Design the control logic for a FIFO

```
module FIFOControl(Clock, Reset, WriteEnableIn,
  WriteEnableOut, WriteAddress, WriteAck,
  ReadEnableIn, ReadEnableOut, ReadAddress, ReadAck);
```

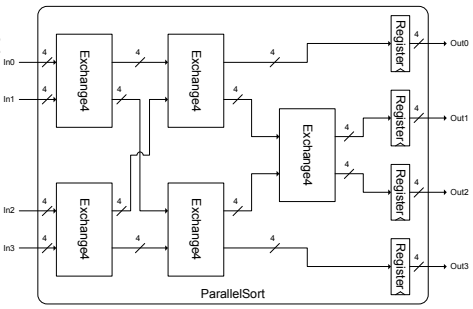
  - Stores the elements in a memory:

```
module RAM(WriteAddress, WriteData, WriteEnable,
  ReadAddress, ReadData, ReadEnable)
```

    - Implements a circular buffer (head: first empty, tail: last full slot)
  - Corner Cases
    - Read & Write on same cycle
    - Full & Empty
- Results
  - Schematic
  - Timing Diagram
  - Proof/argument of corner case handling

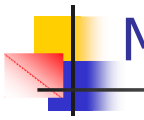
Non-CPU SDS

- Basic Problems
  - Pipeline ParallelSort
  - Design SequentialSort
    - Use only a single exchange module!
    - You'll need FSM for control…
- Results
  - Schematics
  - Timing diagams
- Advanced
  - Code some Verilog
  - Testbench Design?



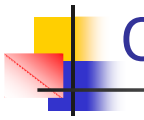
Memory-To-Memory Processor

- Basic Problem
  - No register file
  - All operands must be stored in data memory
- Results
  - Instruction format
  - Block Diagram/Schematic
  - Timing diagram
- Advanced
  - Pipelining



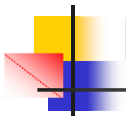
Microcode

- Basic Problem
  - Add support for microcoded instructions to figure 5.17/5.24/proj4
  - Microcode uses multiple small (perhaps non-MIPS) instructions to implement one large instruction
    - Any string operation
    - GCD
- Results
  - Schematic
  - Microcode instruction format
  - Timing Diagram
  - Controller & State
- Advanced
  - Try and implement a second instruction using microcode



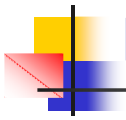
Out Of Order

- Basic Problem
  - Design an out-of-order processor
    - Run short instructions while waiting for long ones
    - Assume memory access & shift take ~10 cycles
  - Account for dependencies (RAW, WAW, WAR)
- Results
  - Schematic (use hierarchy)
  - Timing diagram, showing a simple program
  - Proof/argument that it works properly
- Advanced
  - Execute multiple instructions at once



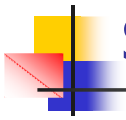
No-Branch Processor

- Basic Problem
  - Design a processor without a standard branch instruction
    - Still need to support control flow changes
    - Could optimize 1 instruction branch bodies?
- Results
  - ISA
    - Instruction Set
    - Register File characteristics
  - Performance estimates/arguments
- Advanced
  - Schematic, Timing Diagram
  - Read about IA64/Itanium



MultiProcessor Locking

- Basic Problem
  - A shared memory system is one in which two CPUs share the same memory, but they execute different code
  - We'd like to be able ensure that only one of them has access to e.g. the CD-RW drive at once
  - Implement support for locking which allows us to guarentee this
- Results
  - ISA (New Instructions)
  - Block Diagram
  - Coherent argument/proof of operation
- Advanced
  - Make it efficient!



Stump the TA

- Basic Problem
  - Any CS61C-related problemYou have ~45min to write it, I get ~5min to solve Should cover the material we've talked about so far
  - You don't need to know the answer
- Results
  - A clear description of the problem
  - A list of the results you expect for a correct solution
- Advanced
  - Figure out the solution
  - Make sure I can't just make something up