1. Bus arbitration
    a. How to arbitrate a bus
    b. Methods of bus arbitration
        i. Centralized
        ii. Distributed
    c. Timing mechanisms
        i. Synchronous
        ii. Asynchronous
2. Interrupts
    a. Will be covered again in ECS 150
    b. Why do we need interrupts?
    c. How can we deal with I/O?
        i. Busy waiting
        ii. Polling
        iii. Interrupt

d. What do we need to implement an interrupt?
   i. Must preserve current state

   ii. Jump to the correct interrupt service routine / subroutine (ISR) based on the interrupt type

   iii. Interrupt needs to be invisible, so current state can be restored correctly

e. Changes we need to make to support interrupts (incomplete list)
   i. Modify our original program order of Fetch, Decode, Execute

   ii. Add a place in memory to store the ISR code / instructions

   iii. Support different types of interrupts

   iv. Need to be able to enable and disable interrupts

   v. Need to know what to load into PC

   vi. Need to add an Interrupt Service Routine (ISR)

   vii. Need a Return from Interrupt (RTI) instruction

3. Classification of interrupts
    a. Examples of possible interrupts
    b. How to classify interrupts
        i. By timing (with the clock)
            1. Synchronous (deterministic)
            2. Asynchronous (nondeterministic)
        ii. Source of interrupt
            1. User request
            2. Coerced
        iii. Masking
            1. User maskable
            2. Non-maskable
        iv. Location (or time) in instruction
            1. Within an instruction
            2. Between instructions
        v. Result
            1. Resume
            2. Terminate