Lecture 25 Last lecture

- Logistics
  - HW9 and Ant extra credit problem are due now
  - All lab must be done by 6pm today.
  - Review session tomorrow CSE403 11:30am
  - Final Exam 6/9 here 8:30am
- Today
  - Review materials from the entire class
  - Announcement of final exam extra credit problem
  - Evaluation: me and Nikhil (leave last 15 min for this)
  - Tomorrow: can take more questions and go over examples.

What you should know

- Combinational logic basics
  - Binary/hex/decimal numbers
  - Ones and twos complement arithmetic
  - Truth tables
  - Boolean algebra
  - Basic logic gates
  - Schematic diagrams
  - Timing diagrams
  - de Morgan's theorem
  - AND/OR to NAND/NOR logic conversion
  - K-maps (up to 4 variables), logic minimization, don't cares
  - SOP, POS
  - Minterm and maxterm expansions (canonical, minimized)

What you should know

- Combinational logic applications
  - Combinational design
    - Input/output encoding
    - Truth table
    - K-map
    - Boolean equation
    - Schematics
  - Multiplexers/demultiplexers
  - PLAs/PALs
  - ROMs
  - Adders

What you should know

- Sequential logic building blocks
  - Latches (R-S and D)
  - Flip-flops (D and T)
  - Latch and flip-flop timing (setup/hold time, prop delay)
  - Timing diagrams
  - Asynchronous inputs and metastability
  - Registers

What you should know

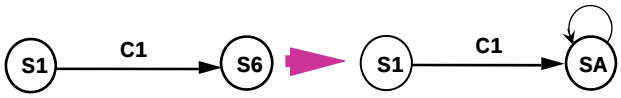
- Counters
  - Timing diagrams
  - Shift registers
  - Ring counters
  - State diagrams and state-transition tables
  - Counter design procedure
    1. Draw a state diagram
    2. Draw a state-transition table
    3. Encode the next-state functions
    4. Implement the design
  - Self-starting counters

What you should know

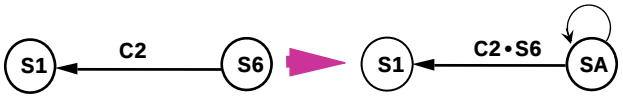
- Finite state machines
  - Timing diagrams (synchronous FSMs)
  - Moore versus Mealy versus synchronized Mealy
  - FSM design procedure
    1. State diagram
    2. state-transition table
    3. State minimization
    4. State encoding
    5. Next-state logic minimization
    6. Implement the design
  - State minimization
  - One-hot / output-oriented encoding
  - FSM design guidelines
    - Separate datapath and control
  - Pipelining, retiming partitioning basics

Partitioning rules

Rule #1: Source-state transformation
Replace by transition to idle state (SA)

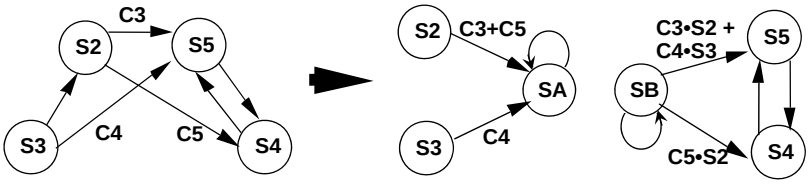


Rule #2: Destination state transformation
Replace with exit transition from idle state

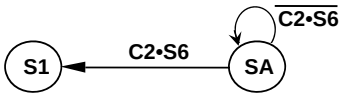


Partitioning rules (con't)

Rule #3: Multiple transitions with same source or destination
Source ⇒ Replace by transitions to idle state (SA)
Destination ⇒ Replace with exit transitions from idle state



Rule #4: Hold condition for idle state
OR exit conditions and invert



Example: 6 state up/down counter (con't)

- Count sequence S_0, S_1, S_2, S_3, S_4, S_5
  - S_2 goes to S_A and holds, leaves after S_5
  - S_5 goes to S_B and holds, leaves after S_2
  - Down sequence is similar