Common Pitfalls Regarding Transition Properties

- Only one condition should be true
  - For all transitions leaving a state
  - Else, which one?
- One condition must be true
  - For all transitions leaving a state
  - Else, where go?

Verifying Correct Transition Properties

- Can verify using Boolean algebra
  - Only one condition true: AND of each condition pair (for transitions leaving a state) should equal 0
  - One condition true: OR of all conditions of transitions leaving a state) should equal 1

Example:

\[ a \cdot a' = a + a'b \]
\[ = a \cdot (1 + b) + a'b \]
\[ = a + ab + a'b \]
\[ = a + (a + a')b \]
\[ = a + b \]

Fails! Might not be 1 (i.e., \( a=0 \), \( b=0 \))

Evidence that Pitfall is Common

- Recall code detector FSM
  - We “fixed” a problem with the transition conditions
  - Do the transitions obey the two required transition properties?
    - Consider transitions of state
      - Start, and the "only one true" property

Simplifying Notations

- FSMs
  - Assume unassigned output implicitly assigned 0
- Sequential circuits
  - Assume unconnected clock inputs connected to same external clock
More on Flip-Flops and Controllers

- Other flip-flop types
  - SR flip-flop: like SR latch, but edge triggered
  - JK flip-flop: like SR (S=J, R=K)
    - But when JK=11, toggles
  - T flip-flop: JK with inputs tied together
    - Toggles on every rising clock edge
      - Previously utilized to minimize logic outside flip-flop
      - Today, minimizing logic to such extent is not as important
  - D flip-flops are thus by far the most common

Non-Ideal Flip-Flop Behavior

- Can’t change flip-flop input too close to clock edge
  - Setup time: time that D must be stable before edge
    - Else, stable value not present at internal latch
  - Hold time: time that D must be held stable after edge
    - Else, new value doesn’t have time to loop around and stabilize in internal latch

Metastability

- Violating setup/hold time can lead to bad situation known as metastable state
  - Metastable state: Any flip-flop state other than stable 1 or 0
    - Eventually settles to one or other, but we don’t know which
    - For internal circuits, we can make sure observe setup time
      - But what if input comes from external (asynchronous) source, e.g., button press?
  - Partial solution
    - Insert synchronizer flip-flop for asynchronous input
    - Special flip-flop with very small setup/hold time
      - Doesn’t completely prevent metastability

Initial State of a Controller

- All our FSMs had initial state
  - But our sequential circuit designs did not
    - Can accomplish using flip-flops with reset/set inputs
      - Shown circuit initializes flip-flops to 01
      - Designer must ensure reset input is 1 during power up of circuit
      - By electronic circuit design
Glitching

- Glitch: Temporary values on outputs that appear soon after input changes, before stable new output values.
- Designer must determine whether glitching outputs may pose a problem.
  - If so, may consider adding flip-flops to outputs.
  - Delays output by one clock cycle, but may be OK.

Active Low Inputs

- We’ve assumed input action occurs when input is 1.
  - Some inputs are instead active when input is 0 -- “active low”.
  - Shown with inversion bubble.
  - So to reset the shown flip-flop, set R=0. Else, keep R=1.

Design Challenge

- Design Challenge
  - Determine what is wrong with the following FSM, and design a possible corrected version using your best guess as to the original intent.

Due Next Lecture (as announced in class)
1 point extra credit (Homework)