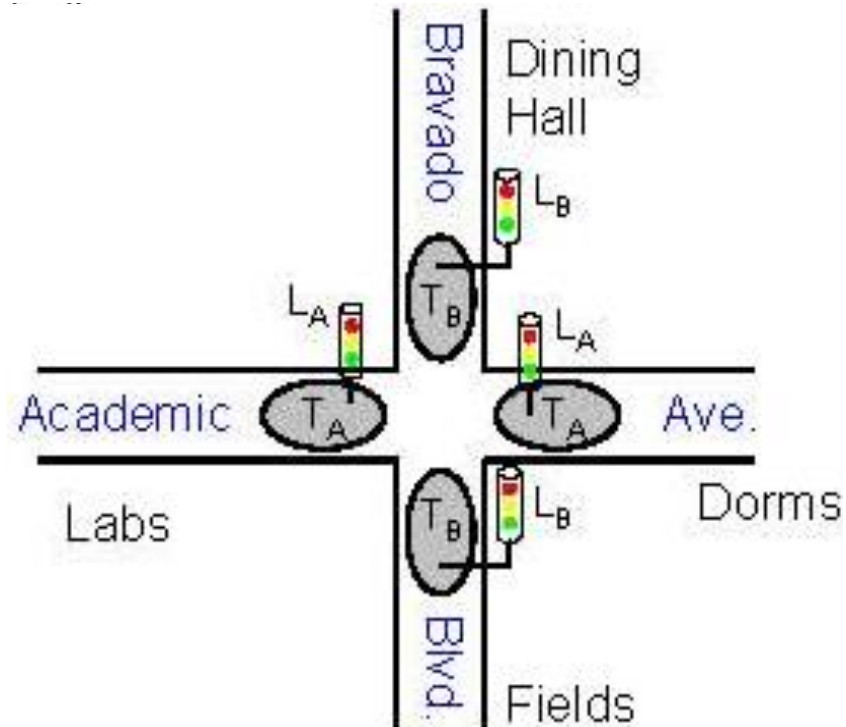


FSM Word Problems

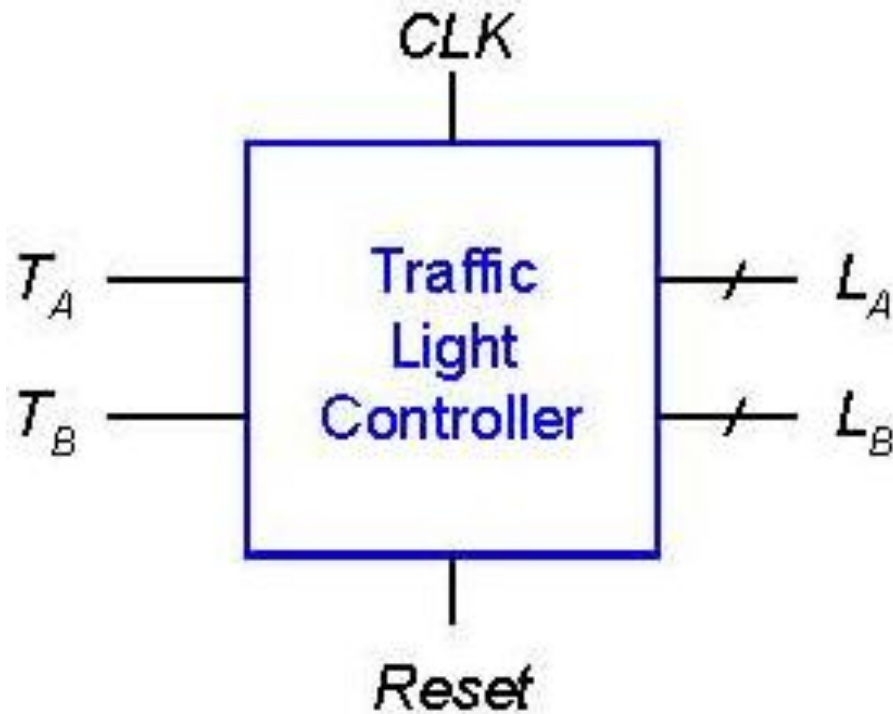
FSM Example

- Traffic light controller
 - Traffic sensors: T_A , T_B (TRUE when there's traffic)
 - Lights: L_A , L_B
 - If $L_A = \text{GREEN}$, then L_A remains GREEN as long as A has cars (i.e. $T_A = 1$)
 - If $L_B = \text{GREEN}$, then L_B remains GREEN as long as B has cars (i.e. $T_B = 1$)
 - Light colors change
(GREEN, RED) \rightarrow
(YELLOW, RED) \rightarrow
(RED, GREEN)

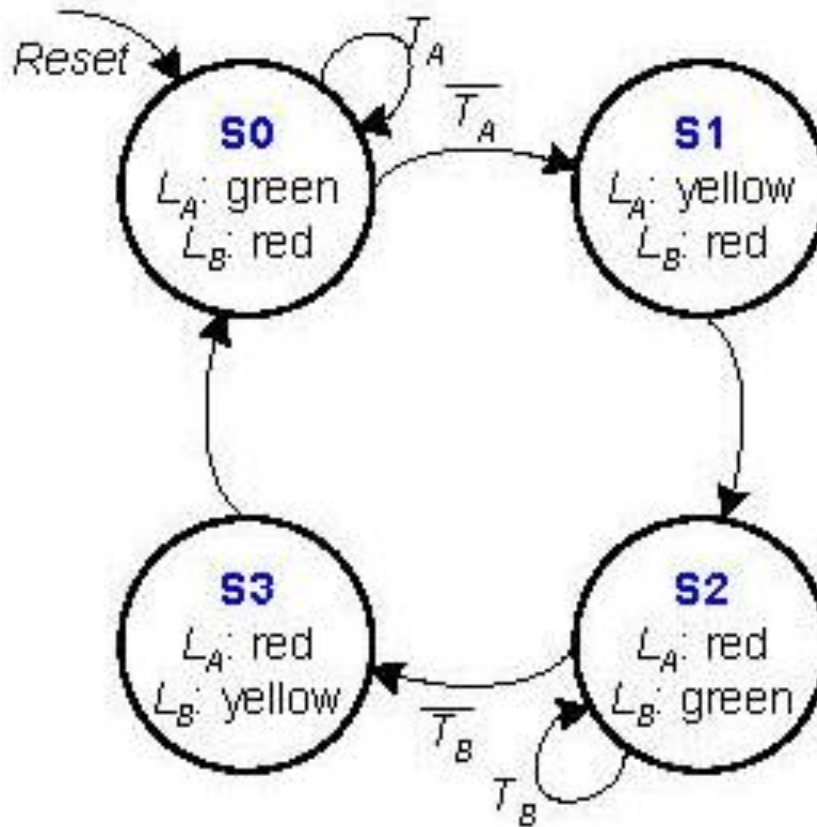


FSM Blackbox

- Inputs: CLK, T_A , T_B
- Outputs: L_A , L_B

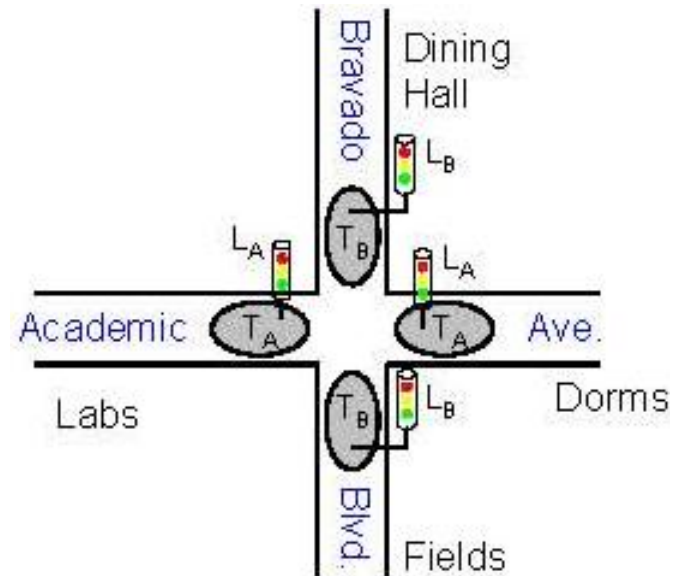


FSM State Transition Diagram



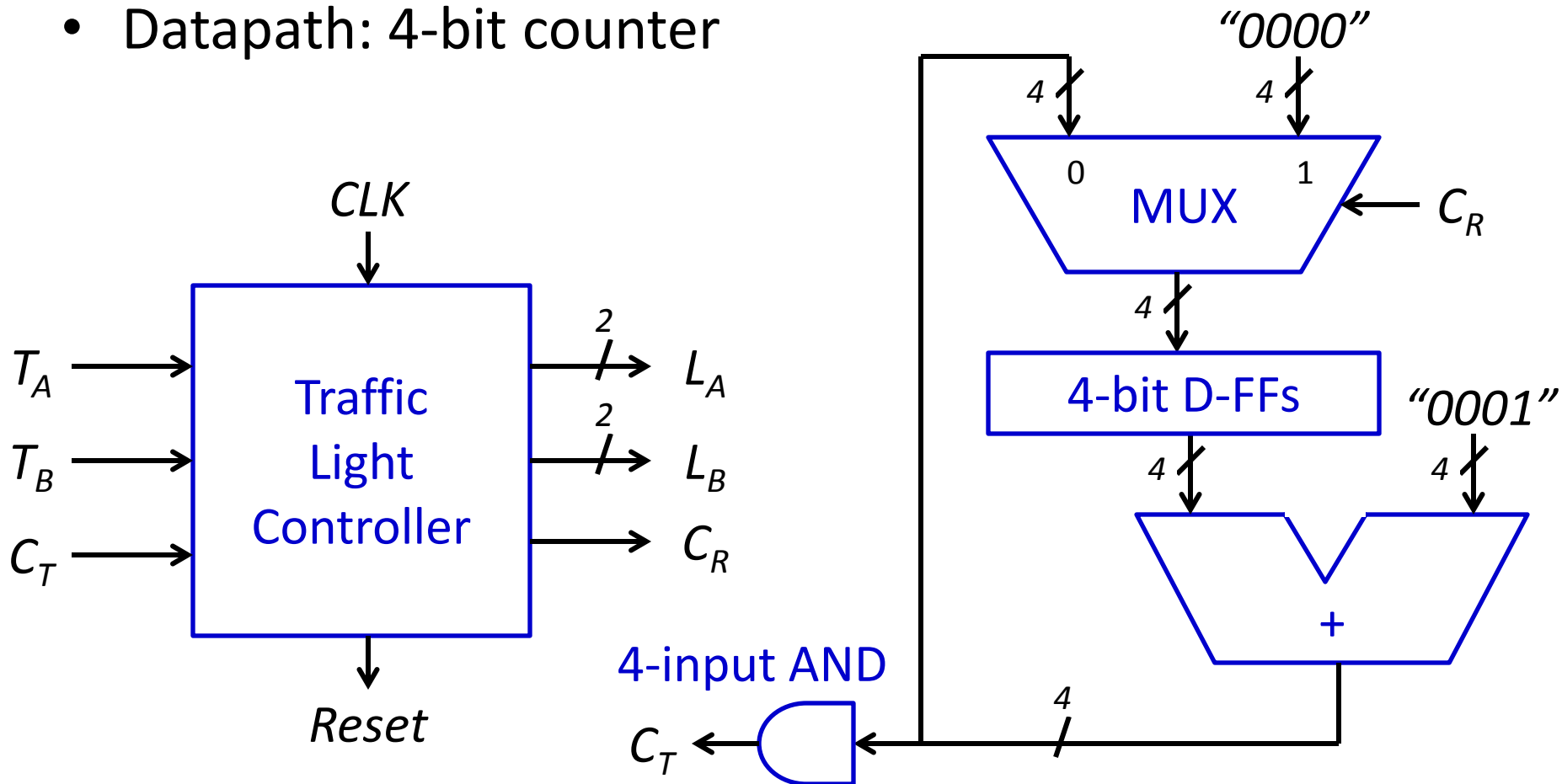
Modified FSM Example

- If L_A is GREEN: Case
 - No cars on either A or B: change light after 16 cycles
 - Cars waiting on both A and B: change light after 16 cycles
 - Cars waiting only on A: L_A stays GREEN
 - Cars waiting only on B: change light
 - Light colors change (GREEN, RED) \rightarrow (YELLOW, RED) \rightarrow (RED, GREEN)
- If L_B is GREEN: Case ...

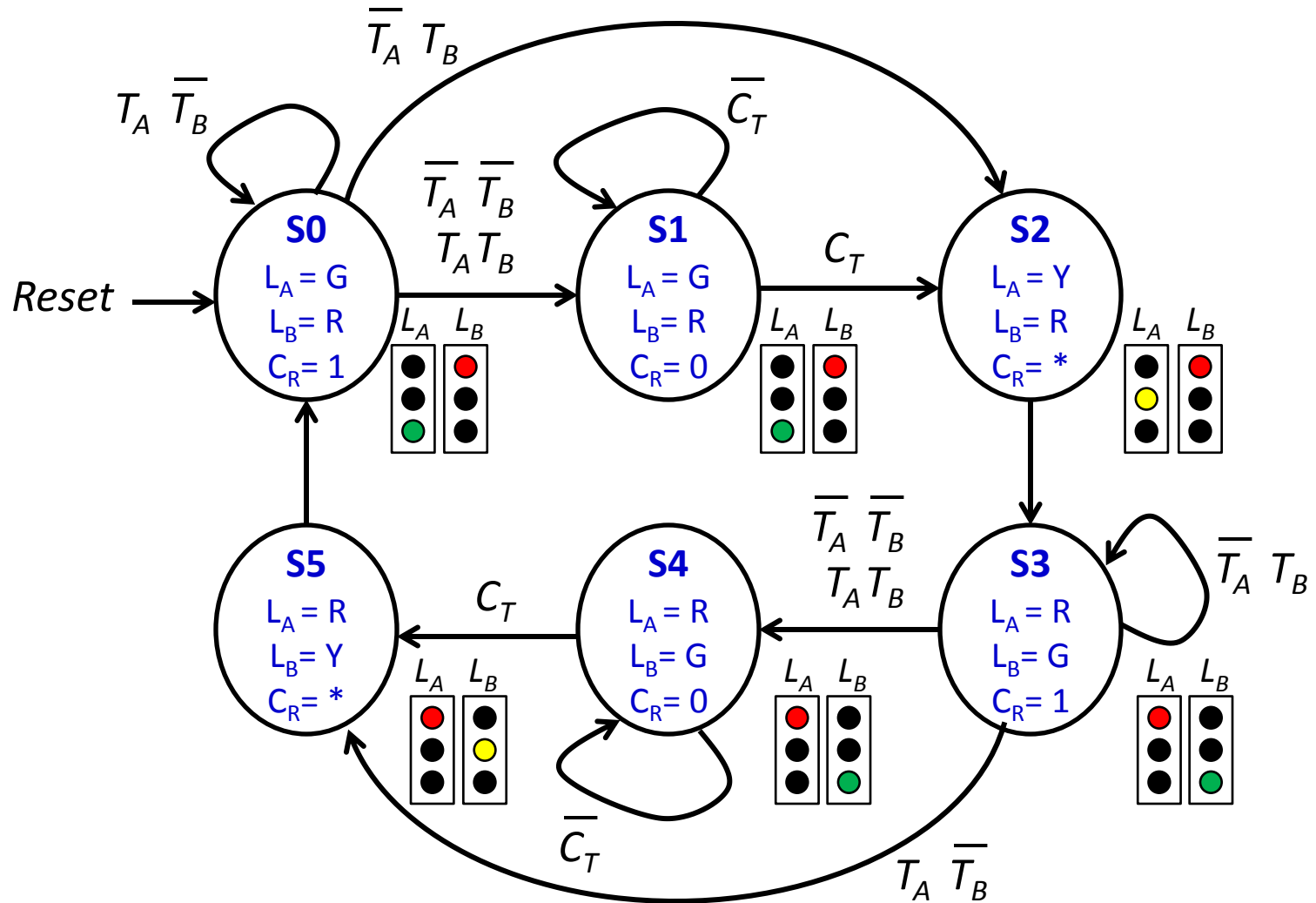


FSM Blackbox

- FSM Inputs: CLK, T_A , T_B , C_R (to reset counter)
- FSM Outputs: L_A , L_B , C_T (TRUE if count is 15, all 1's)
- Datapath: 4-bit counter

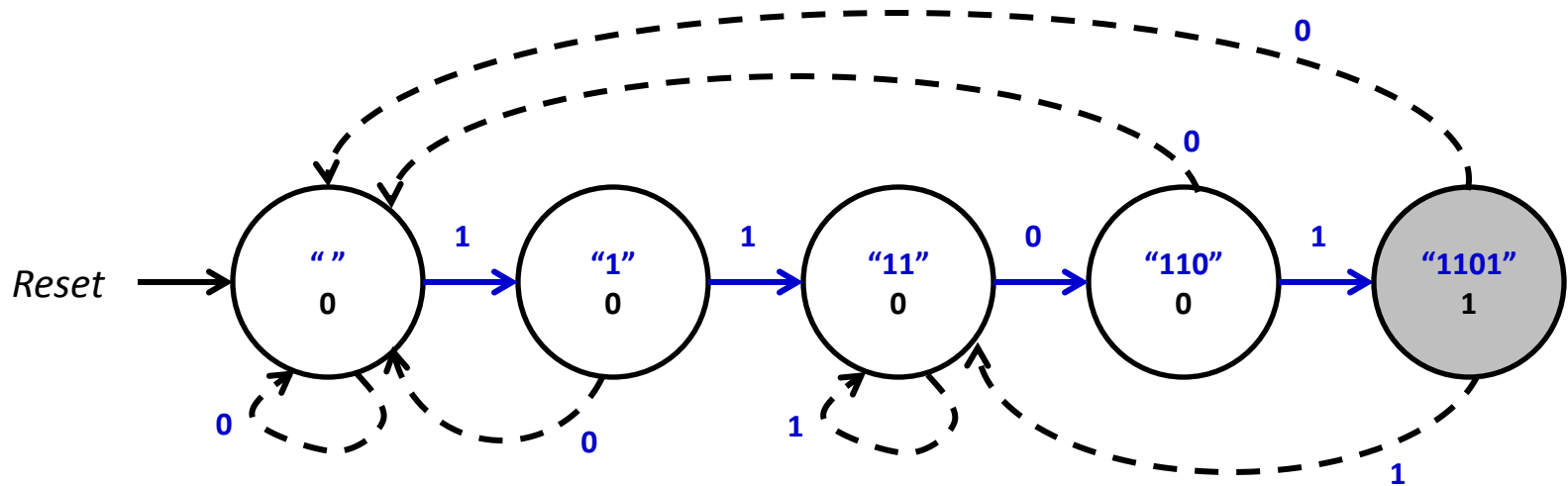


FSM State Transition Diagram



String Recognition

- Simple case: recognize 1101

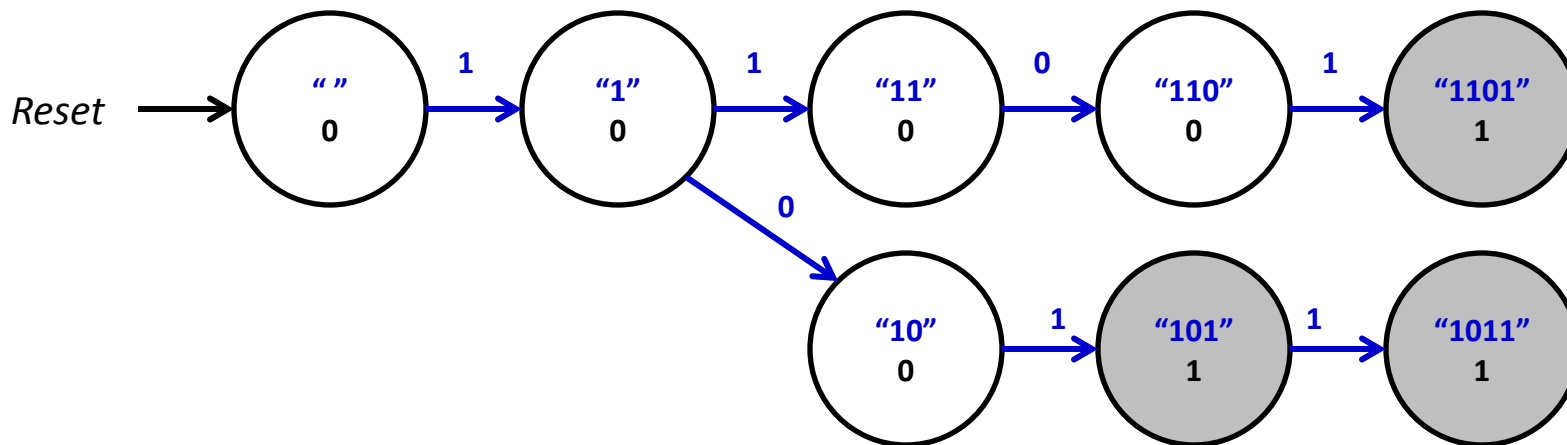


String Recognition

- General rules:
 1. Create string matching tree
 2. State output = 1 if a “pattern” is contained along the path
 3. Add failure edges: find longest “suffix” of string seen so far and transition to the corresponding “prefix” state

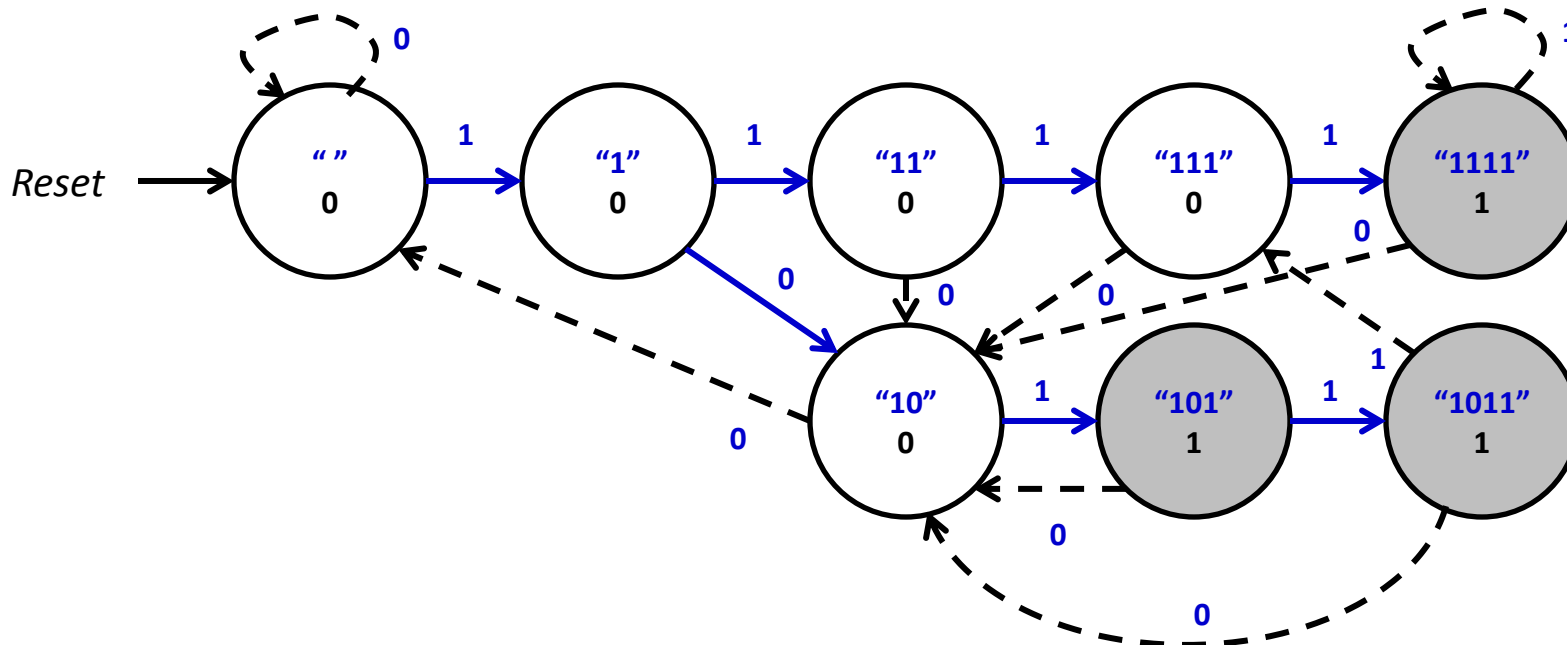
String Recognition

- Example: 1101, 1011, 101
 1. Create string matching tree
 2. State output = 1 if a “pattern” is contained along the path
 3. Add failure edges: find longest “suffix” of string seen so far and transition to the corresponding “prefix” state



String Recognition

- Example: 1111, 1011, 101
 1. Create string matching tree
 2. State output = 1 if a “pattern” is contained along the path
 3. Add failure edges: find longest “suffix” of string seen so far and transition to the corresponding “prefix” state



String Recognition using Shift Registers

- Example: 1111, 1011, 101
 1. Use N bit shift register if longest pattern has N bits
 2. Create an AND gate for pattern
 3. OR together AND gates
 4. Caveat: result is invalid for the first N cycles

