Sequential Logic

State Machines

Kumaravel Jagasivamani
Feb. 19, 2024

Overview
Part 3

- Glitch with single output bit and multiple input bits
- Finite State Machine
- Alarm example
  - Glitch
    - Flip-flop for outputs
    - Gray code
  - Mealy vs. Moore

Glitch

- Z = A’B’ + AC
- ABC changes from 001 to 101
  - A changes from 0 to 1
- Z should remain at 1
- Glitch: unexpected change in output for a short period of time
  - Z could go to 0 if A’B’ changes from 1 to 0 before AC changes from 0 to 1
- Add B’C term to avoid glitch
  - Z = A’B’ + AC + B’C
- Alternative solution:
  - Add a flip-flop for Z to ensure glitch doesn’t show up at flip-flop output

A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

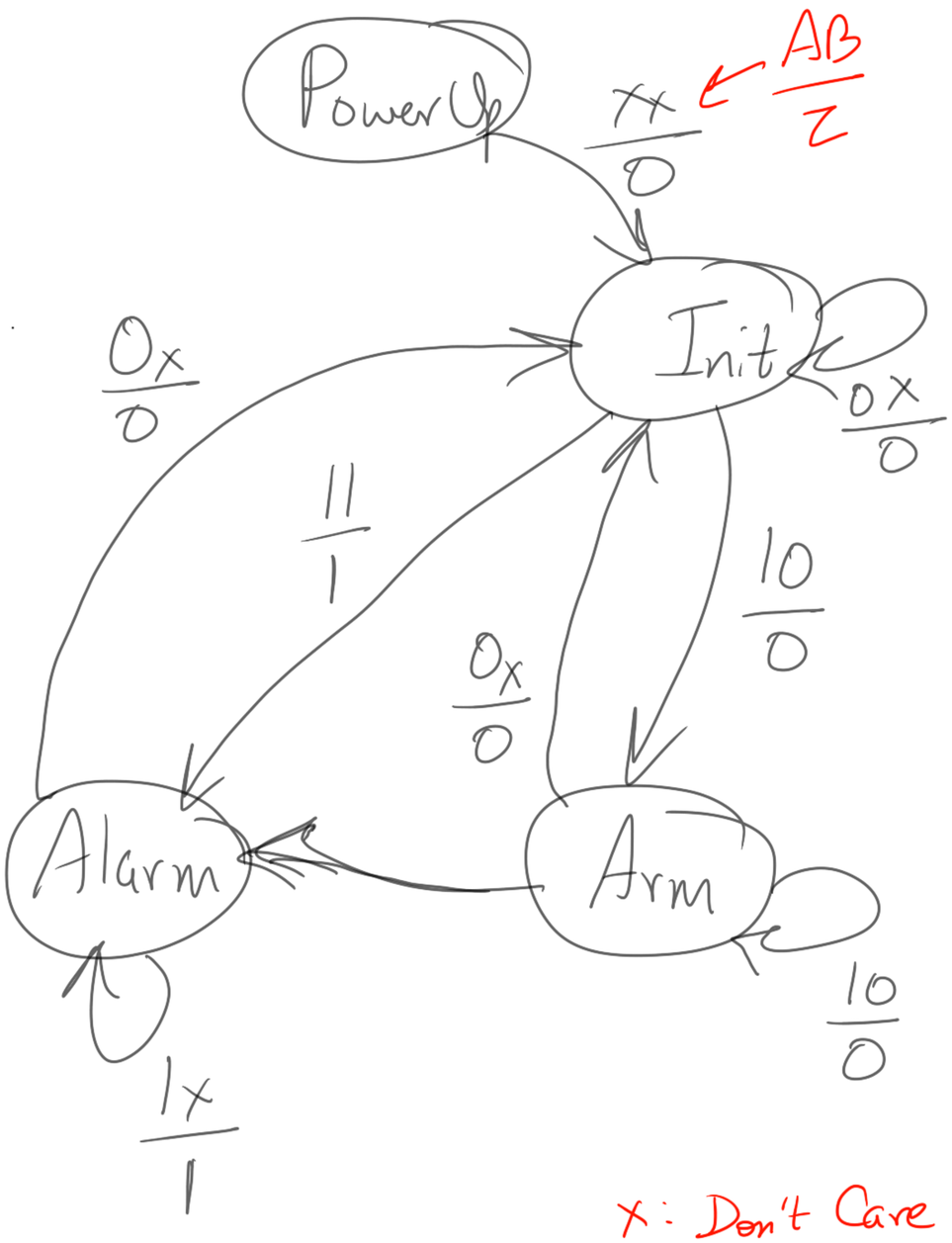
Finite State Machine

Design Steps

- Obtain requirements of system
- Create state transition diagram
- Identify number of bits needed to store state information
- Assign binary values for each state
- Create truth table
- Use K-Maps to write equation
- Draw block diagram
    - Flip-flops (registers) for each bit of state
    - Combinational logic to update the D input of each bit of state
    - Combinational logic to generate the output of system

Alarm Example

- States:
  - Power-Up: 00
  - Init: 01
  - Wait: 10
  - Alarm: 11
- Inputs:
  - A: Arm the alarm
  - B: Sensor detection
- Output:
  - Z: Alarm sound



Alarm Example

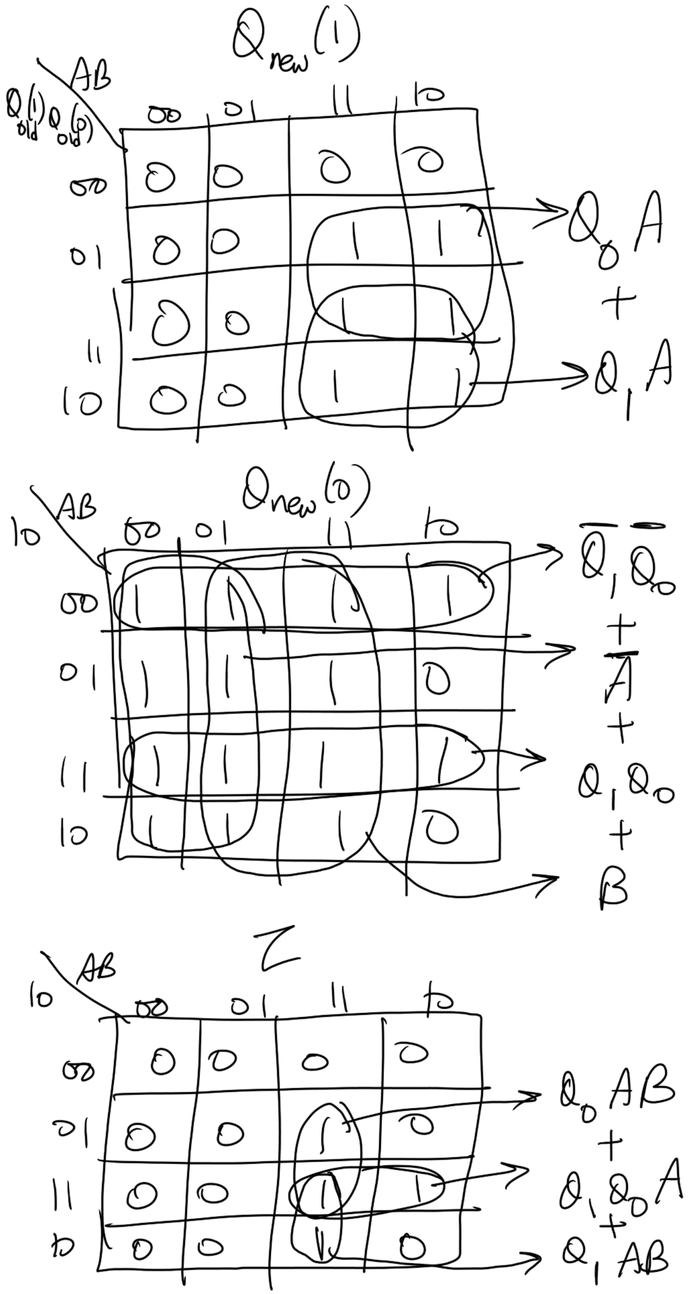
Truth Table

$Q_{old}(1)$	$Q_{old}(0)$	A	B	$Q_{new}(1)$	$Q_{new}(0)$	Z
0	0	0	0	0	1	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	1	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Alarm Example

K-Maps

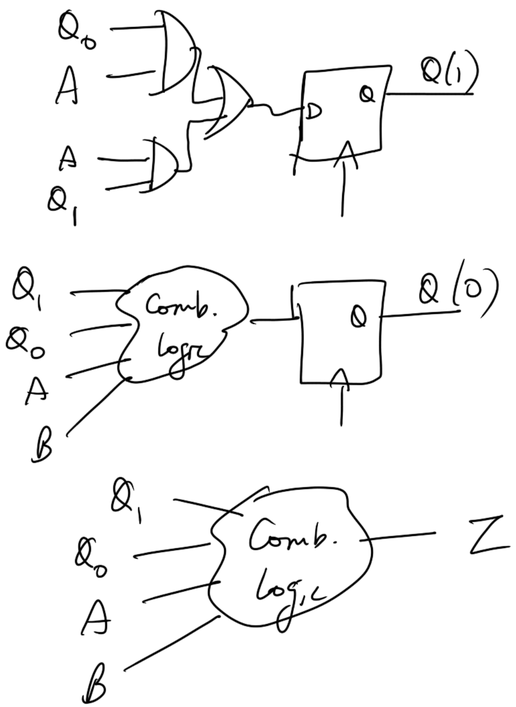
- Qnew(1) = Q0A + Q1A
- Qnew(0) = Q1’Q0’ + A’ + Q1Q0 + B
- Z = Q0AB + Q1Q0A + Q1AB
-



Alarm Example

Block Diagram

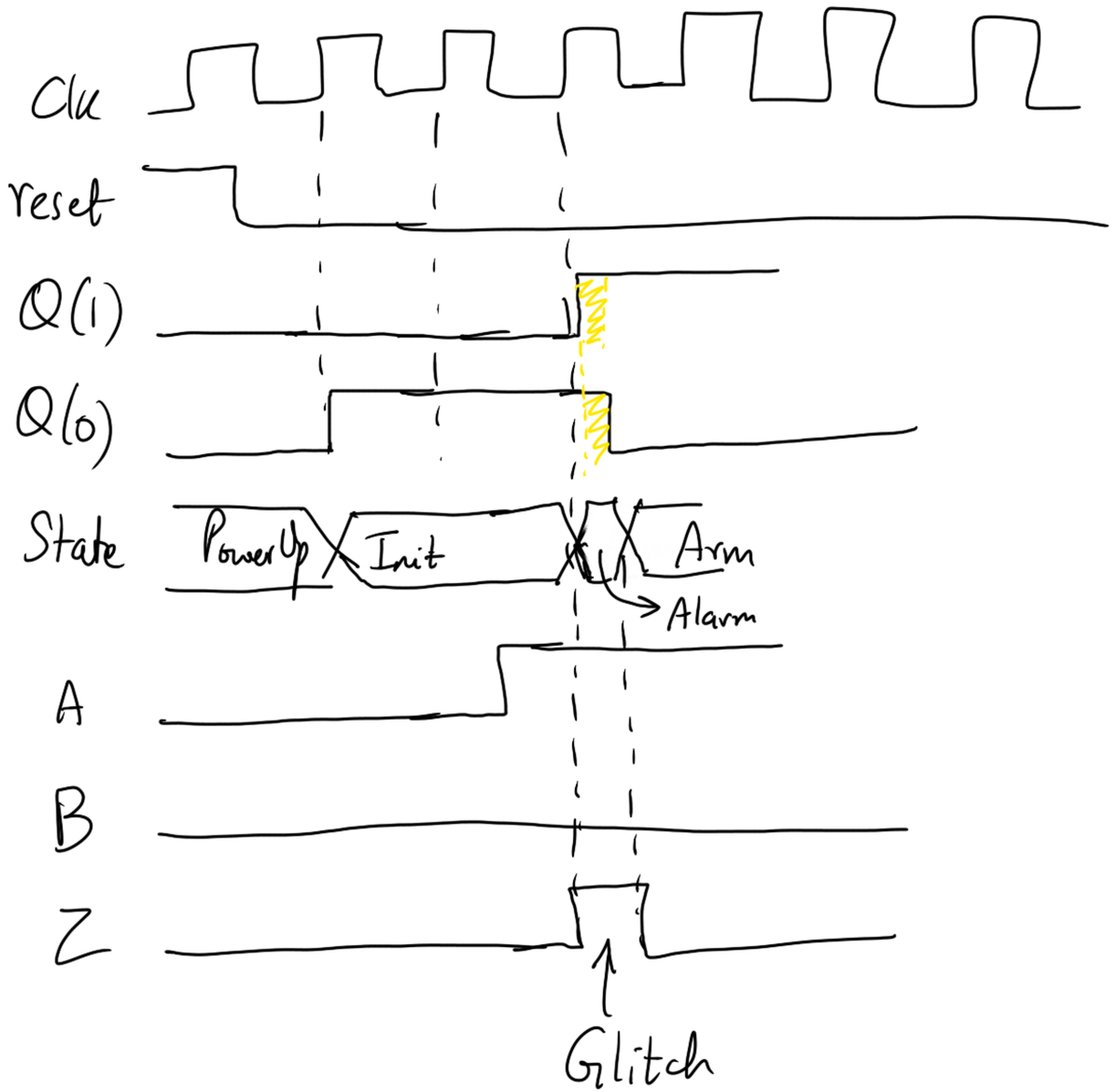
- Qnew(1) = Q0A + Q1A
- Qnew(0) = Q1’Q0’ + A’ + Q1Q0 + B
- Z = Q0AB + Q1Q0A + Q1AB
-



Alarm Example

Glitch

- Transition of A: 0->1 with state 01
  - Bit 0: 1->0
  - Bit 1: 0->1
- State: 01 -> 11 -> 10
  - State bits are independent
  - State could erroneously go to Alarm state for a brief period



Alarm Example
Ways to avoid glitch

- Register the output
  - Add a flip-flop for Z
- Gray code for state assignment
  - Problem caused because both bits of state changed when transitioning from Init -> Arm
  - Allow only bit of state to change for transitions to avoid problem
    - PowerUp: 000
    - Init: 001
    - Arm: 011
    - Alarm: 101
    - Add an extra state to transition from Arm to Alarm with value of 111

Moore vs Mealy FSM

- Mealy: output depends on both state and input
- Moore: output only depends on state
  - Changes/Glitches in input doesn’t affect output
  - Simpler design than Mealy
    - Changes in input take time to reflect on output

Alarm Example
Moore FSM

- States:
  - Power-Up: 00
  - Init: 01
  - Wait: 10
  - Alarm: 11
- Moore:
  - Z = Q1Q0
  - Only depends on state
- Mealy:
  - Z = Q0AB + Q1Q0A + Q1AB
  - Depended on both state and inputs