Mealy and Moore Machines

ECE 152A – Winter 2012

Reading Assignment

- Brown and Vranesic
  - 8 Synchronous Sequential Circuits
    - 8.3 Mealy State Model

Reading Assignment

- Roth
  - 13 Analysis of Clocked Sequential Circuits
    - 13.1 A Sequential Parity Checker
    - 13.2 Analysis by Signal Tracing and Timing Charts
    - 13.3 State Tables and Graphs
    - 13.4 General Models for Sequential Circuits

Finite State Machines

- Thus far, sequential circuit (counter and register) outputs limited to state variables
- In general, sequential circuits (or Finite State Machines, FSM's) have outputs in addition to the state variables
  - For example, vending machine controllers generate output signals to dispense product, provide change, illuminate displays, etc.

Finite State Machines

- Two types (or models) of sequential circuits (or finite state machines)
  - Mealy machine
    - Output is function of present state and present input
  - Moore machine
    - Output is function of present state only
- Analysis first, then proceed to the design of general finite state machines

Analysis by Signal Tracing and Timing Diagrams

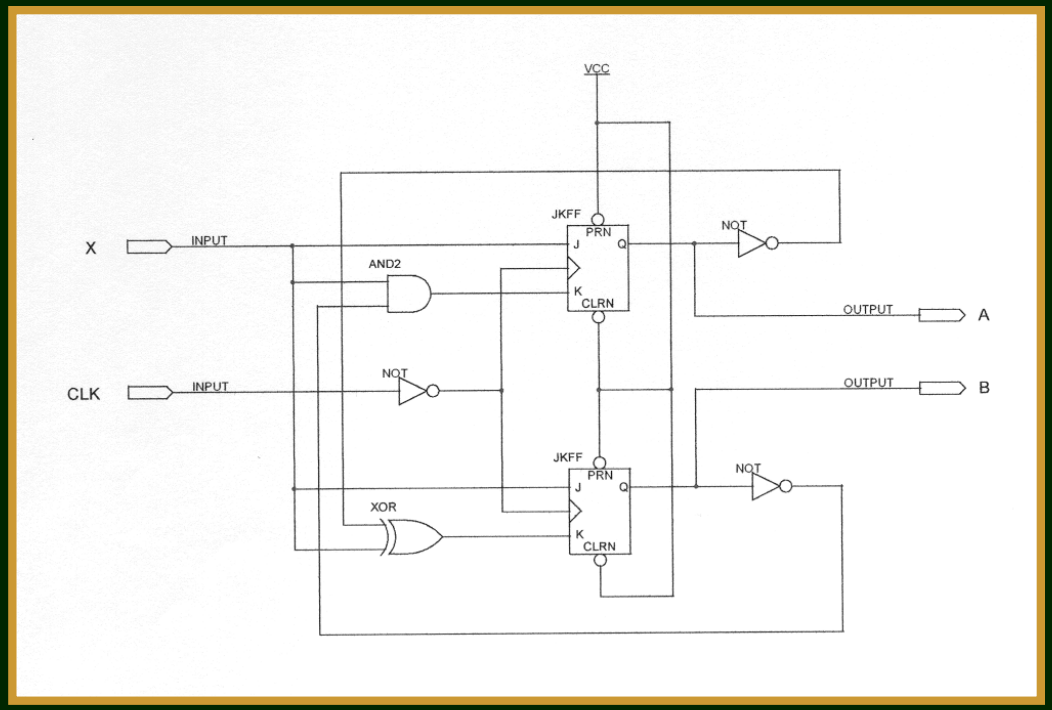
- Timing Analysis
  - Determine flip-flop input equations
  - Determine output equations
    - Mealy or Moore model
  - Generate timing diagram illustrating circuit's response to a particular input sequence
    - Outputs as well as to state

Moore Network Example

- Implemented with falling edge triggered (by way of external inverter) JK flip-flops
- Schematic (following slide)
  - $J_A = x \quad K_A = xB'$
  - $J_B = x \quad K_B = x \text{ XOR } A' = xA + x'A'$
  - $z = B$ (function of present state only)

Moore Network Example

- Schematic



Moore Network Example

- Timing Diagram and Analysis
  - Initial conditions:
    - A = B = z = 0
  - Input sequence:
    - x = 10101
  - All state and output transitions occur after the falling clock edge
    - Assumes x changes on rising edge
      - Best case assumption for satisfying setup and hold time

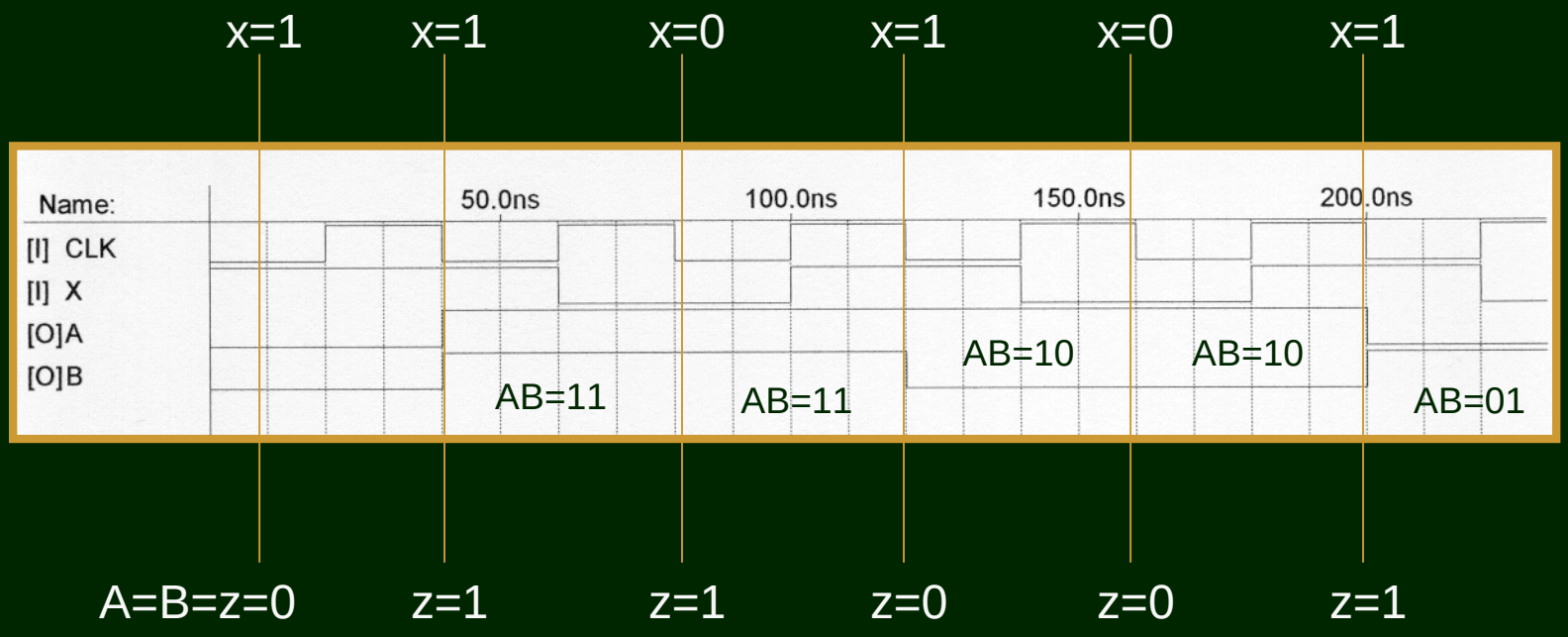
Moore Network Example

- Timing Diagram (Functional Simulation)

$J_A = x$ $K_A = xB'$

$J_B = x$ $K_B = x \text{ XOR } A' = xA + x'A'$

$z = B$

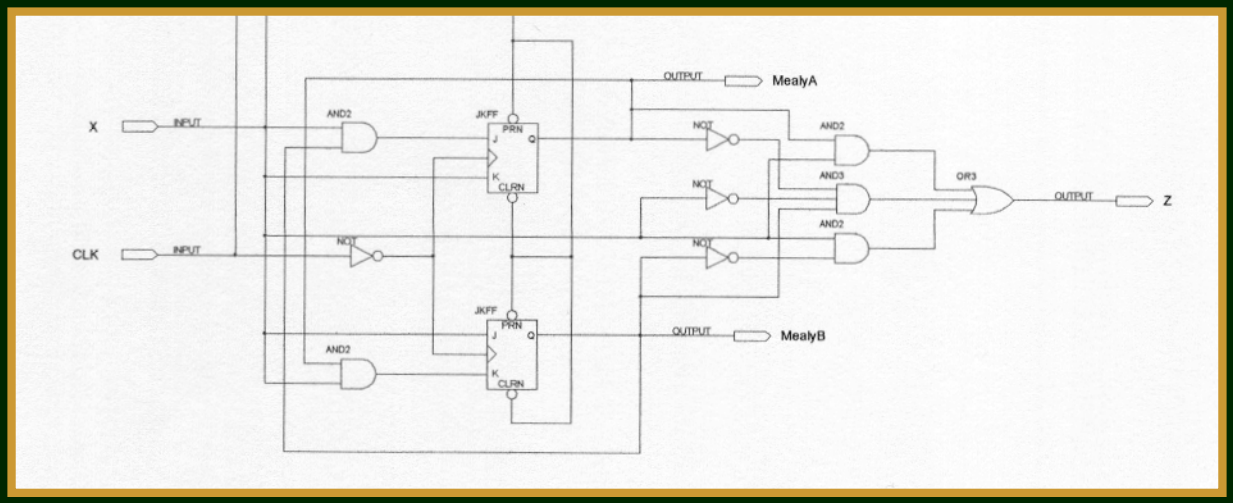


Mealy Network Example

- Implemented with falling edge triggered (by way of external inverter) JK flip-flops
- Schematic (following slide)
  - $J_A = xB$ $K_A = x$
  - $J_B = x$ $K_B = xA$
  - $z = xB' + xA + x'A'B$
    - function of present state and present input

Mealy Network Example

- Schematic



Mealy Network Example

- Timing Diagram and Analysis
  - Initial conditions:
    - A = B = 0
    - z = 1
  - Input sequence:
    - x = 10101
  - Analysis again assumes x changes on rising edge of clock
  - All state transitions occur after the falling clock edge (as with Moore machine)

Mealy Network Example

- Timing Diagram and Analysis (cont)
  - Output transitions occur in response to both input and state transitions
    - “glitches” may be generated by transitions in inputs
    - Moore machines don’t glitch because outputs are associated with present state only
  - Assumes gate delays to output(s) much shorter than clock period
    - All outputs stable before occurrence of active clock edge

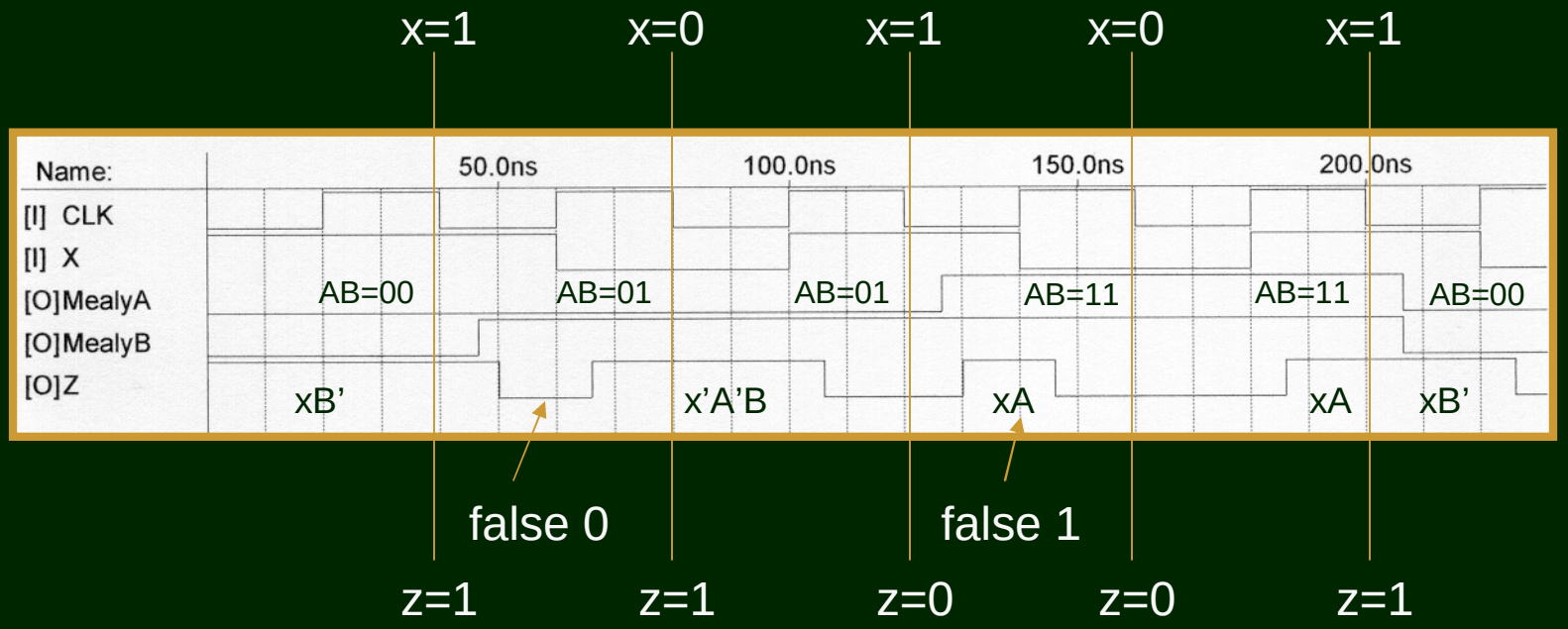
Mealy Network Example

- Timing Diagram (Timing Simulation)

$J_A = xB$ $\quad K_A = x$

$J_B = x$ $\quad K_B = xA$

$z = xB' + xA + x'A'B$



Mealy Machines and Glitches

- In synchronous network, glitches don't matter
  - All data transfers occur around common, falling (or rising) clock edge
    - Register transfer operations
    - Outputs sampled only on active clock edge
  - Output is stable before and after active clock edge
    - Setup and hold times satisfied

FSM Outputs & Timing - Summary

- For Moore machine, output is valid after state transition
  - Output associated with stable present state
- For Mealy machine, output is valid on occurrence of active clock edge
  - Output associated with transition from present state to next state
  - Output in Mealy machine occurs one clock period before output in equivalent Moore machine

Derivation of State Tables and Diagrams

- Timing diagram illustrates the sequential circuit's response to a particular input sequence
  - May not include all states and all transitions
- In general, analysis needs to produce state diagram and state table
- Reverse of design process
  - Begin with implementation, derive state diagram

Derivation of State Tables and Diagrams

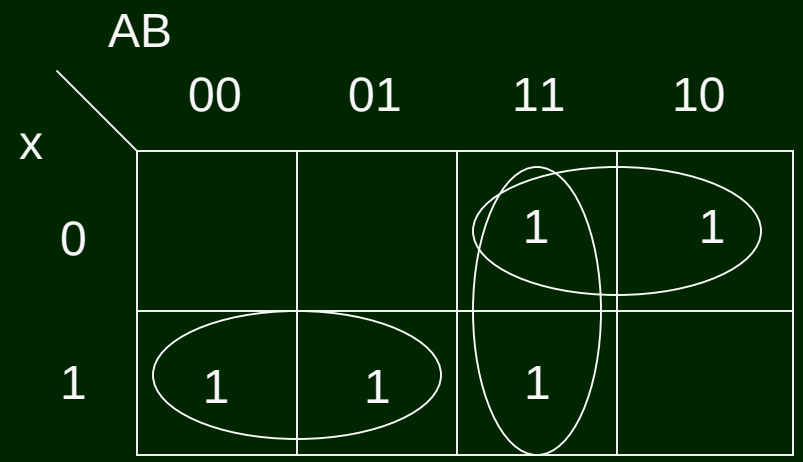
- Returning to Moore machine example
  - Flip-Flop inputs and circuit output functions
    - $J_A = x$ $K_A = xB'$
    - $J_B = x$ $K_B = x \text{ XOR } A' = xA + x'A'$
    - $z = B$ (function of present state only)
- Begin with characteristic equation for JK Flip-Flop
  - $Q^+ = JQ' + K'Q$

Derivation of State Tables and Diagrams

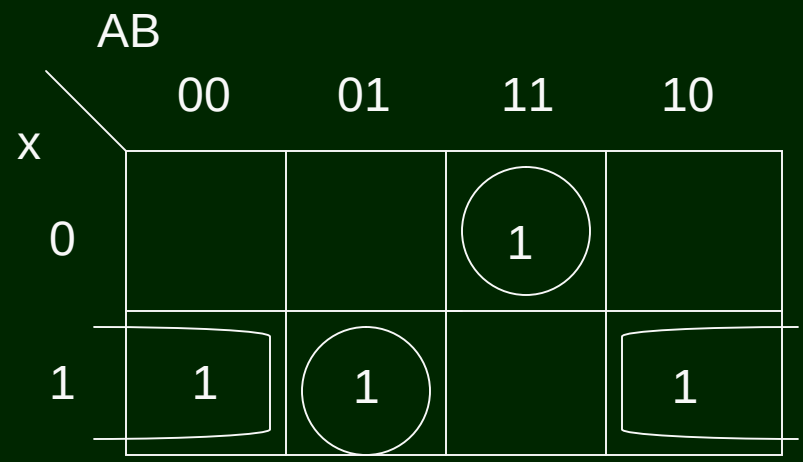
- Using characteristic function, generate next state equations and maps for each flip flop
  - $Q^+ = JQ' + K'Q \rightarrow A^+ = J_AQ' + K_A'Q$
  - $A^+ = xA' + (xB')' A = xA' + x'A + AB$
  - $Q^+ = JQ' + K'Q \rightarrow B^+ = J_BQ' + K_B'Q$
  - $B^+ = xB' + (x \text{ xor } A')'B = xB' + xA'B + x'AB$

Derivation of State Tables and Diagrams

- Next State Maps



$A^+ = xA' + x'A + AB$



$B^+ = xB' + xA'B + x'AB$

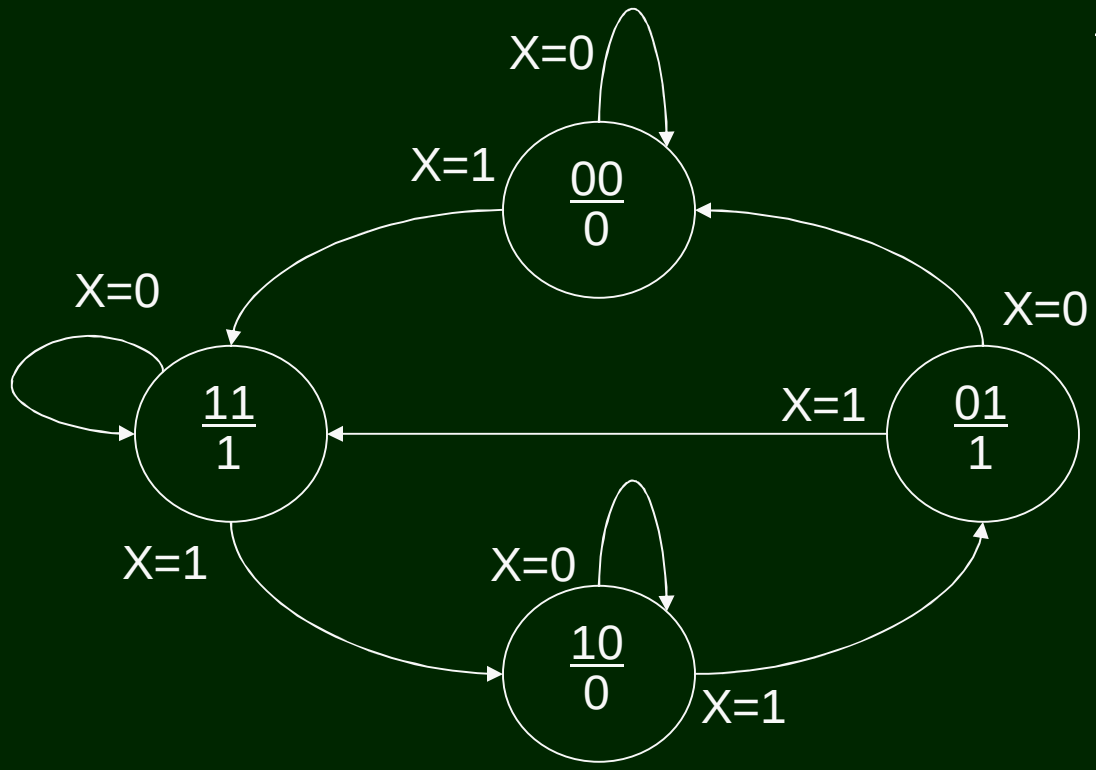
Derivation of State Tables and Diagrams

- State Table

	NS		
PS	X=0	X=1	
AB	AB	AB	z (=B)
00	00	11	0
01	00	11	1
10	10	01	0
11	11	10	1

Derivation of State Tables and Diagrams

- State Diagram



	NS		
PS	X=0	X=1	
AB	AB	AB	z (=B)
00	00	11	0
01	00	11	1
10	10	01	0
11	11	10	1

Derivation of State Tables and Diagrams

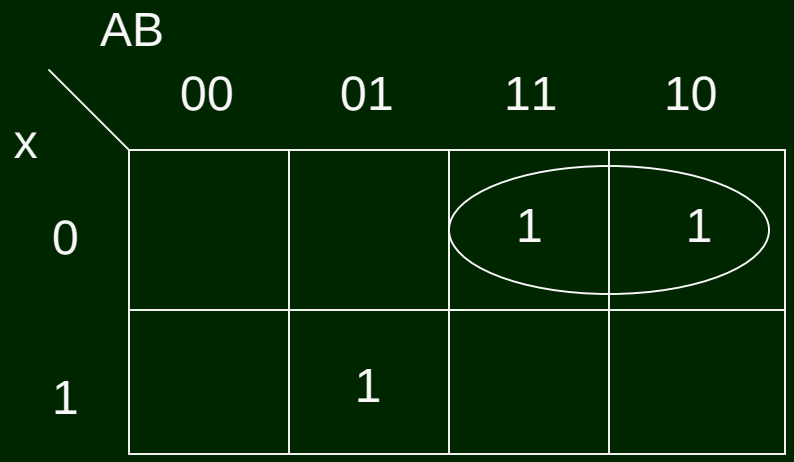
- Mealy machine example
  - Flip-Flop inputs and circuit output functions
    - $J_A = xB$ $K_A = x$
    - $J_B = x$ $K_B = xA$
    - $z = xB' + xA + x'A'B$
- Once again, begin with characteristic Equation for JK Flip-Flop
  - $Q^+ = JQ' + K'Q$

Derivation of State Tables and Diagrams

- Generate next state equations and maps for each flip flop
  - $Q^+ = JQ' + K'Q \rightarrow A^+ = J_AQ' + K_A'Q$
  - $A^+ = xBA' + x'A$
  - $Q^+ = JQ' + K'Q \rightarrow B^+ = J_BQ' + K_B'Q$
  - $B^+ = xB' + (xA)'B = xB' + x'B + A'B$

Derivation of State Tables and Diagrams

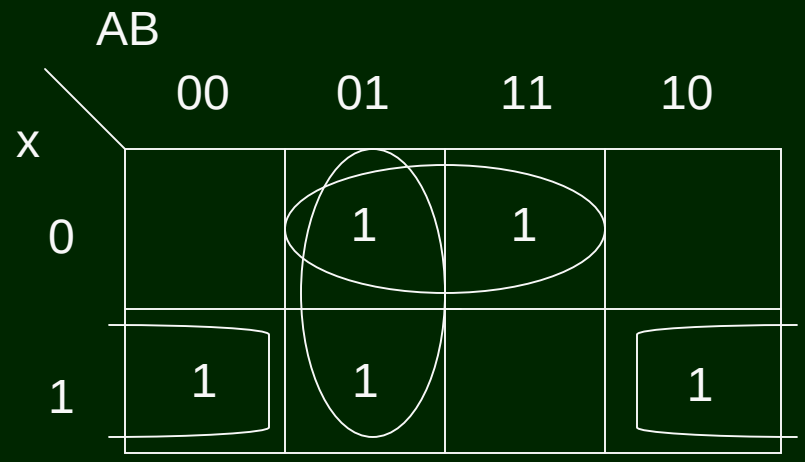
- Next state and output maps

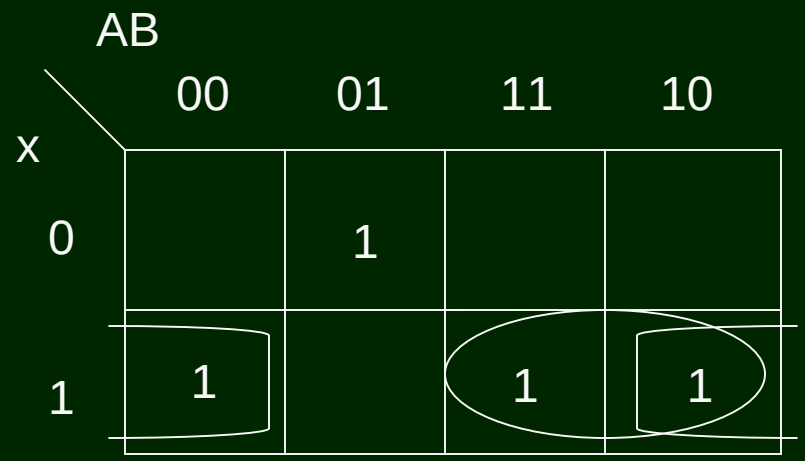


$A^+ = xBA' + x'A$

$B^+ = xB' + x'B + A'B$

$z = xB' + xA + x'A'B$





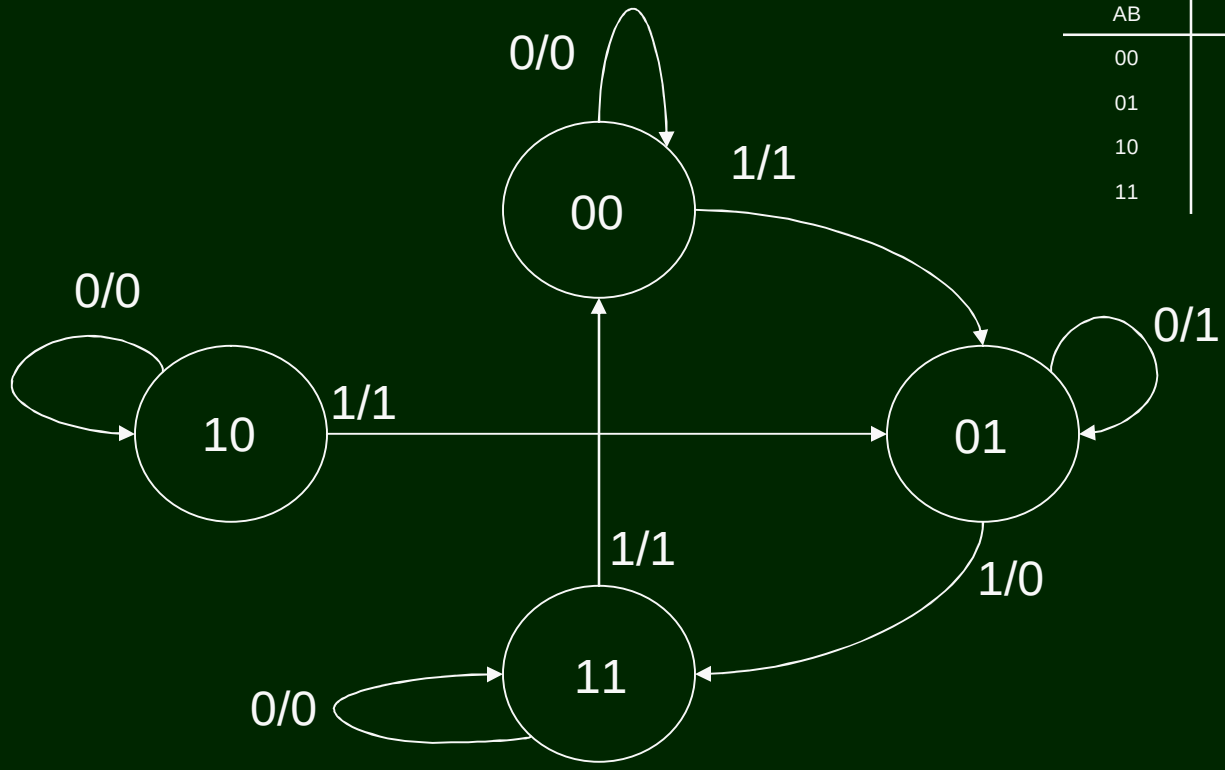
Derivation of State Tables and Diagrams

- State Table

	NS	
PS	x=0	x=1
AB	AB,z	AB,z
00	00,0	01,1
01	01,1	11,0
10	10,0	01,1
11	11,0	00,1

Derivation of State Tables and Diagrams

- State Diagram



	NS	
PS	x=0	x=1
AB	AB,z	AB,z
00	00,0	01,1
01	01,1	11,0
10	10,0	01,1
11	11,0	00,1