Synchronous Sequential Logic

Sequential vs Combinational Ckts

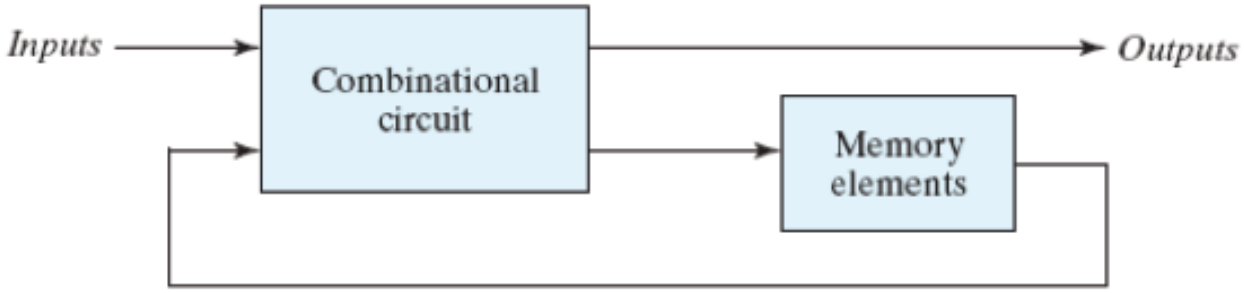
- Recall definition of combinational logic
  - Outputs are a function of inputs
  - Combinational delay (each gate has some propagation delay)

- Sequential circuits
  - Outputs depend on inputs and previous values of outputs
    - » Outputs depend on previous state of the circuit
    - » State is stored in memory elements (registers, latches, fliplops)

Sequential Circuits

- So far: combinational circuits
  - Output of circuit depends only on inputs
- Most real-world circuits also contain memory
  - Pocket calculators
  - Counters
  - Computer systems, etc.

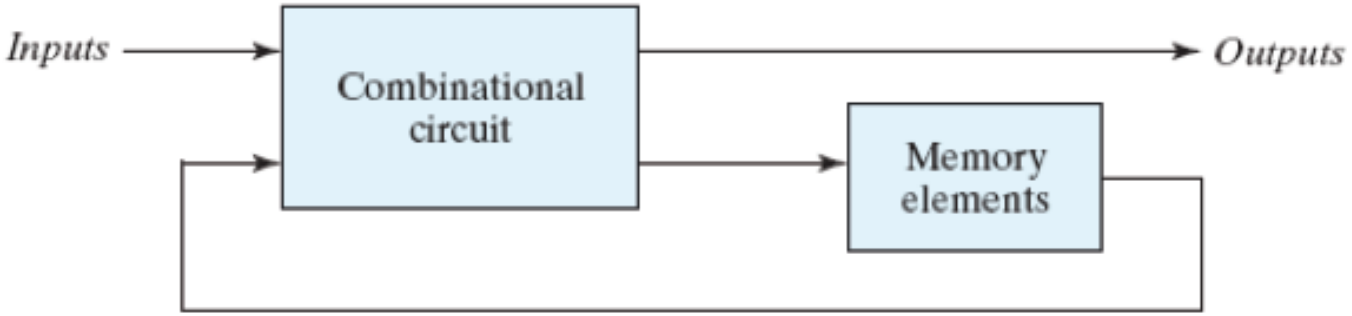
- Sequential circuit:



Synchronous Sequential Circuits

- Synchronous sequential circuits
  - Storage elements change only at discrete instances of time
- Timing controlled by “clock”
  - Clock generator provides train of clock pulses:

- Clocked (synchronous) sequential circuit:



Storage Elements

- Binary storage device capable of storing one bit
- Latch = *level-sensitive* device
  - State changes with input when enabled (e.g., when clock = 1)
  - Holds last input value when disabled (when clock = 0)
- Flip-flop = *edge-triggered* device
  - State of flip-flop can only change during clock *transition*
  - Example: Flip-flops change on rising/falling edge of clock

- Why change on an edge?
  - Couldn't we change state while clock is 1?
  - That would be a latch!
- Edge is moment in time, state is duration
  - Feedback would continue during clock being 1, causing possible race conditions

Level-sensitive vs Edge-triggered

- Latches are level-sensitive

(a) Response to positive level

- Flip-flops are edge-sensitive

(b) Positive-edge response

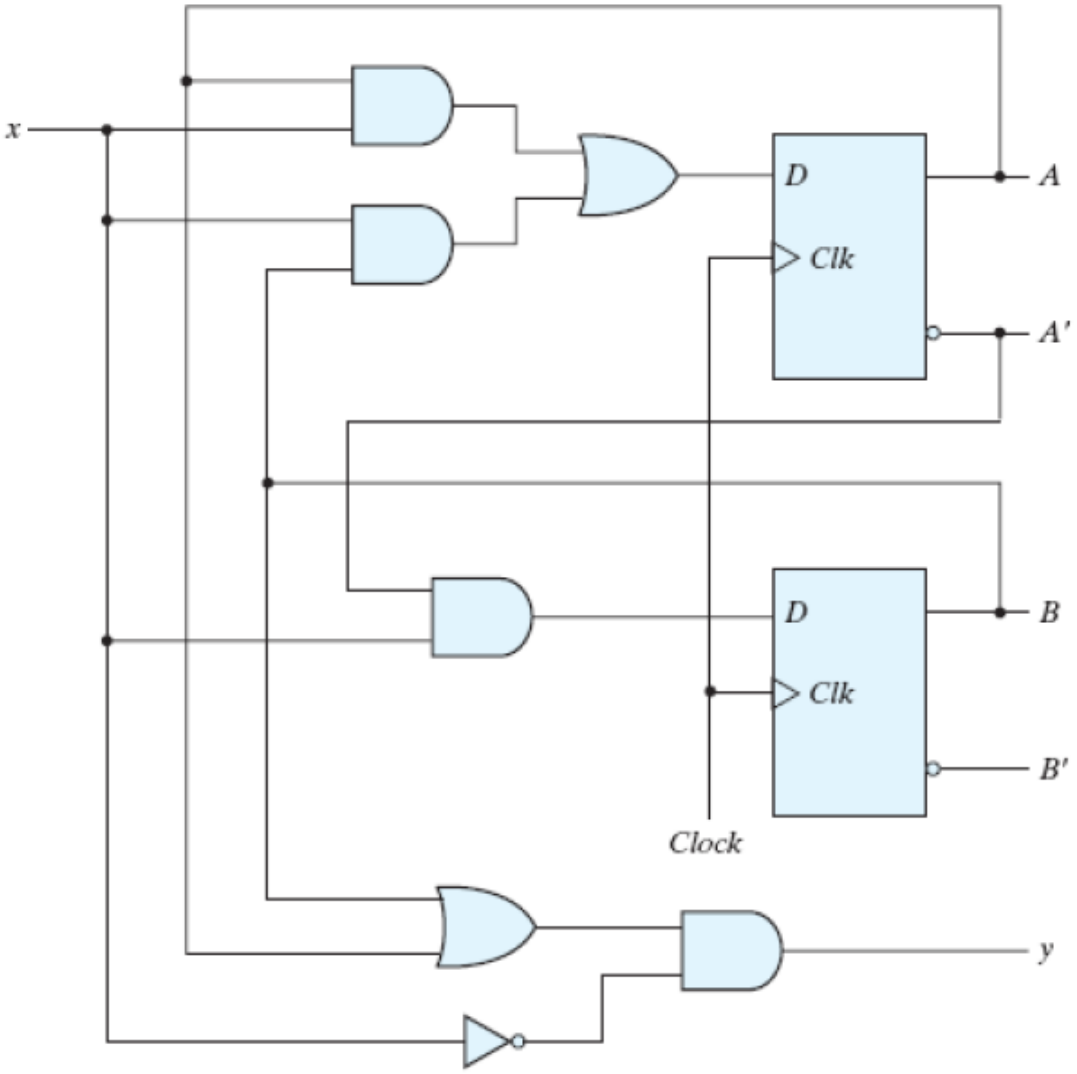
(c) Negative-edge response

Analysis of Sequential Circuits

- Behavior of clocked sequential circuit determined by
  - Inputs
  - Outputs
  - State of flip-flops
- Analysis process
  - Consider all combinations of
    - » Inputs
    - » Flip-flop states
  - Determine next state and output of circuit
- Concept of a Finite State Machine (FSM)
- Methods
  - State equations
  - State table
  - State diagram

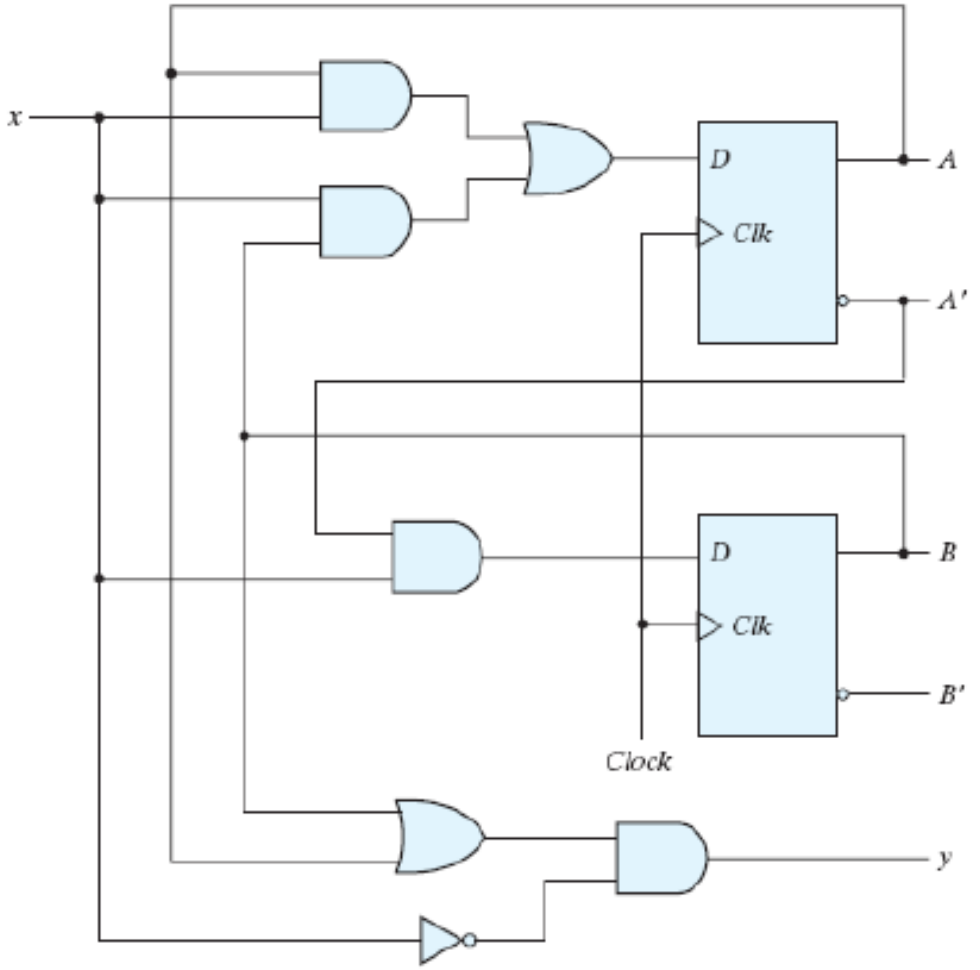
Example Circuit

- Sequential circuit
  - Input: x
  - Output: *y*
  - Flip-flops:
    - » 2 D-type *A* and *B*
- When is *y=1*?
  - Very difficult to answer
  - Systematic analysis necessary



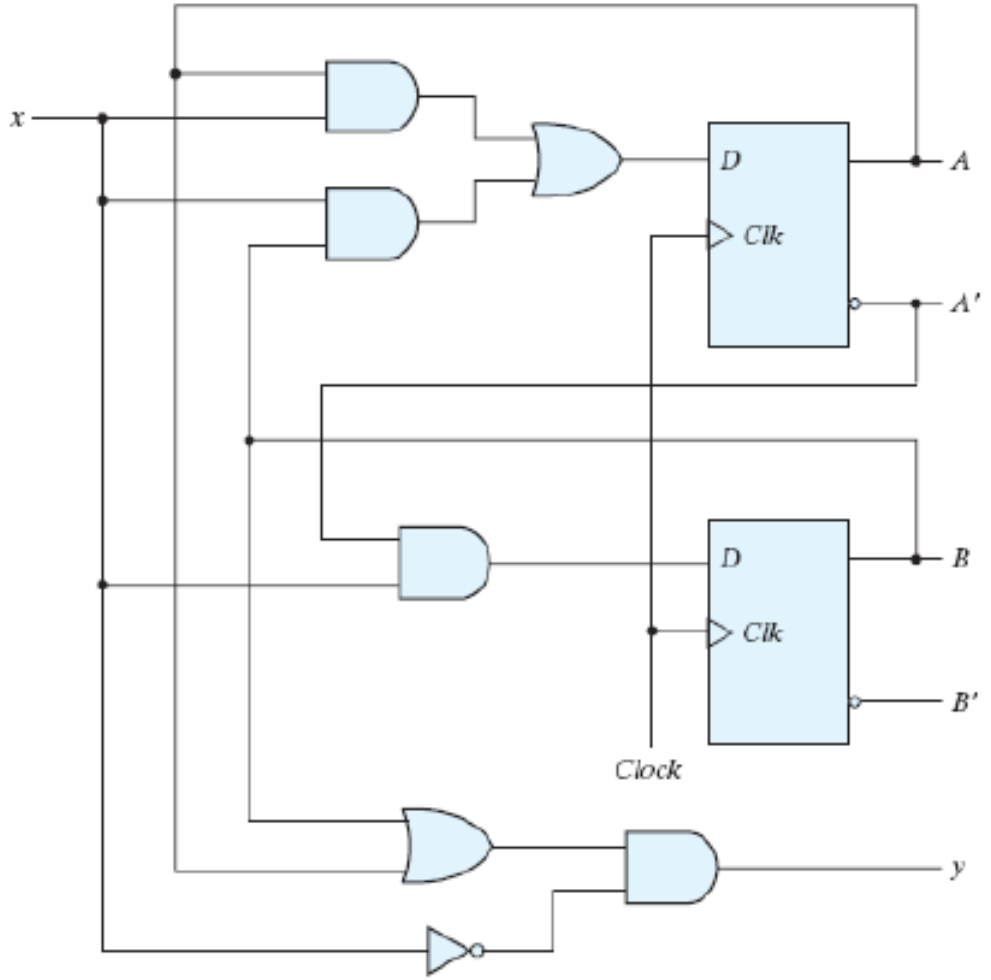
State Equations

- State equation specifies next state
  - Function of current state and inputs
- State equation for flip-flops:
  - $A(t+1) = A(t)\,x(t) + B(t)\,x(t)$
  - $B(t+1) = A'(t)\,x(t)$
- Output expression:
  - $y(t) = (A(t) + B(t))\,x'(t)$
- Simplified:
  - $A(t+1) = A\,x + B\,x$
  - $B(t+1) = A'\,x$
  - $y = (A+B)\,x'$



Flip-flop Input Equations

- Similar to state equations
  - Specifies type of flip-flop used
- Example:
  - $D_A = A\,x + B\,x$
  - $D_B = A'\,x$
  - $y = (A + B)\,x'$



State Table

- What needs to be considered in table?
  - Inputs
  - State of flip-flops
  - Next state of flip-flops
  - Outputs

  Inputs and State of flip-flops: "left side" of table

  Next state of flip-flops and Outputs: "right side" of table

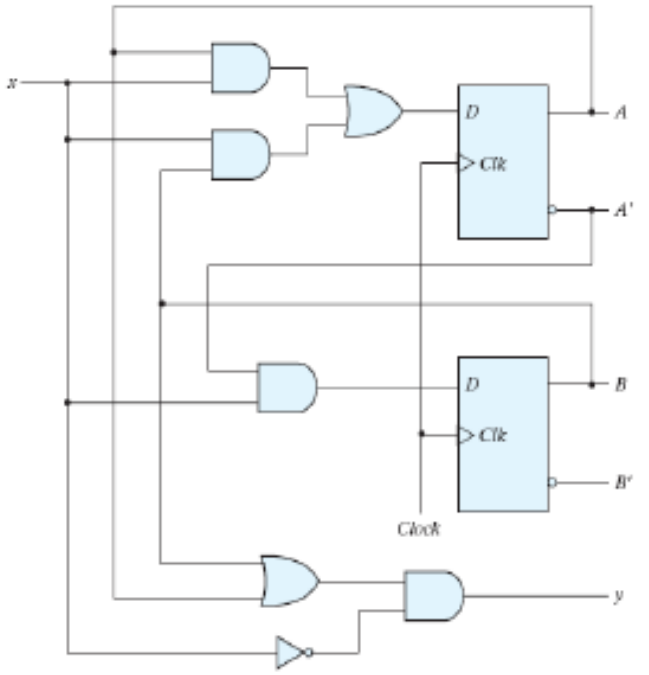
- How many entries in state table?
  - *n* inputs
  - *m* states
  - Total of 2^{m+n} entries

- For every entry
  - Determine flip-flop change by input and current state
    - » State equation
  - Determine the output
    - » Output equation

State Table

- State table

current state		input	next state		output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



State Table

- Alternate form:

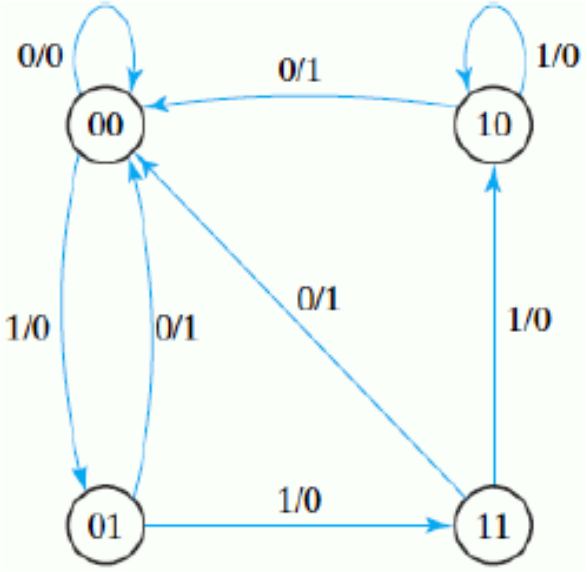
current state	next state		output	
	x=0	x=1	x=0	x=1
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

- Note that state combinations can be concatenated
  - *AB = 00* instead of *A = 0* and *B = 0*

State Diagram

- State transitions represented as graph
  - Vertices indicate states
  - Edges represent transitions
    - » Edge annotation: "*x/y*" meaning input is *x* and output is *y*

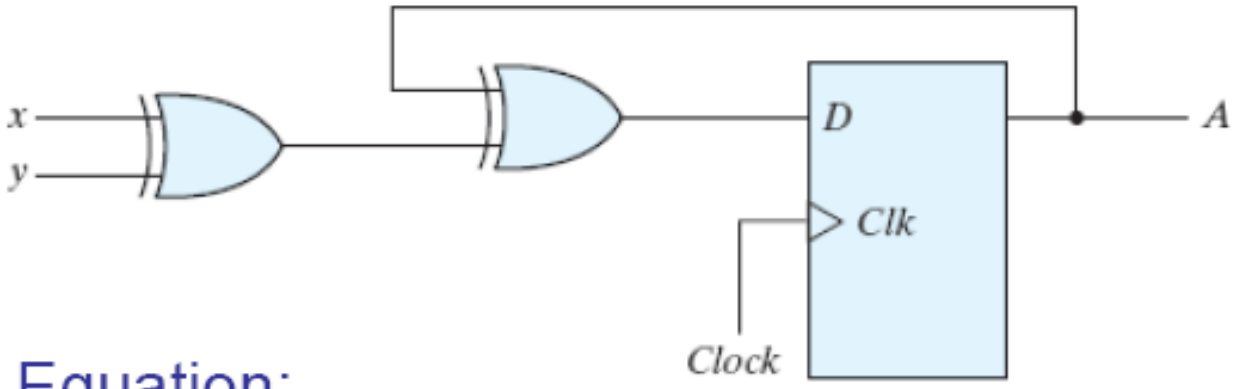
- Easiest generated from state table



current state	next state		output	
	x=0	x=1	x=0	x=1
AB	AB	AB	y	y
00	00	01	0	0
01	00	11	1	0
10	00	10	1	0
11	00	10	1	0

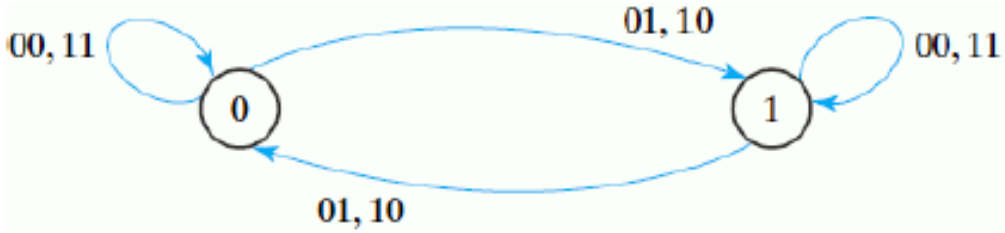
Analysis Example

- Circuit:



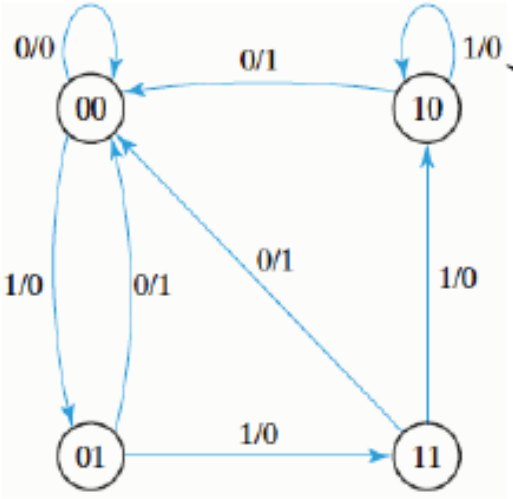
Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

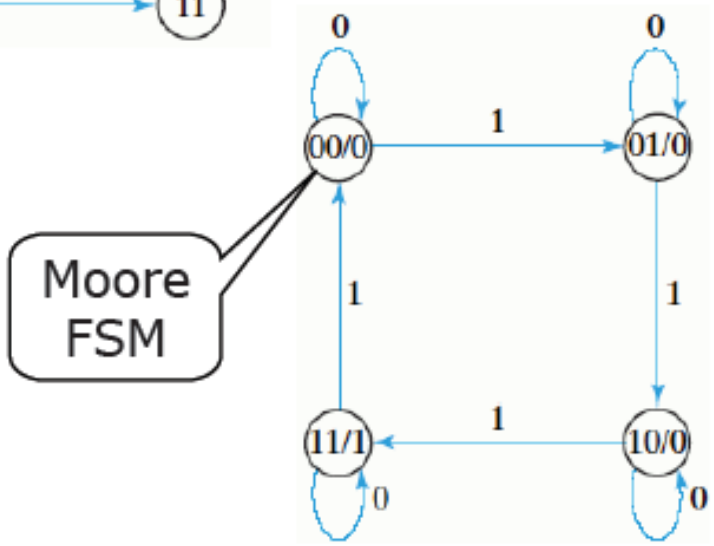
- Equation:
  - $A(t+1) = A \oplus x \oplus y$ (state equation)
  - $D_A(t+1) = A \oplus x \oplus y$ (flip-flop equation)
- State table:
- State diagram:
  - Note: no outputs



Finite State Machines

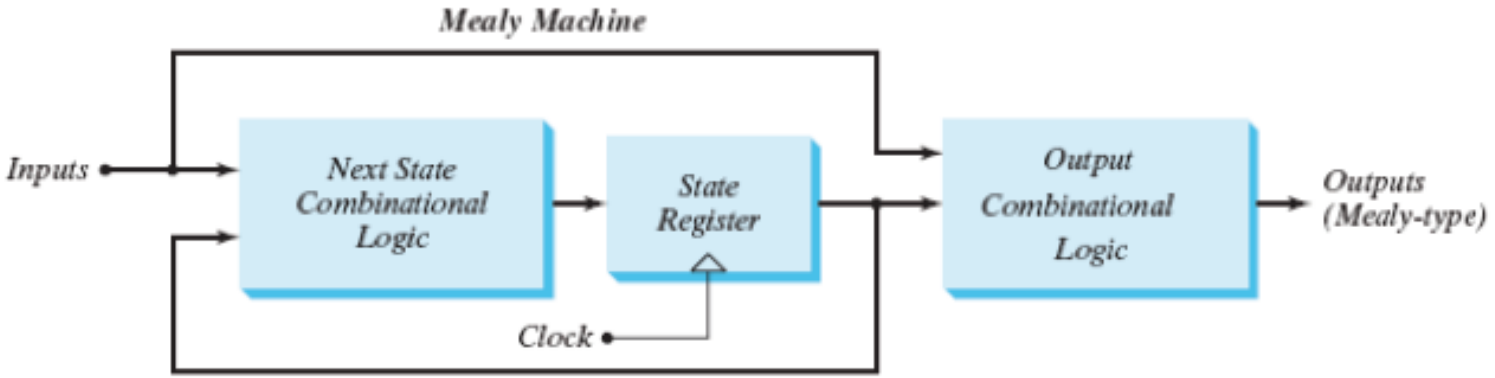
- State diagrams are representations of *Finite State Machines* (FSM)
  - Two flavors of FSMs:
    - » Mealy FSM
    - » Moore FSM
- Difference:
  - How output is determined
- Mealy FSM
  - Output depends on input and state
  - Output is not synchronized with clock
    - » can have temporarily unstable output
- Moore FSM
  - Output depends only on state

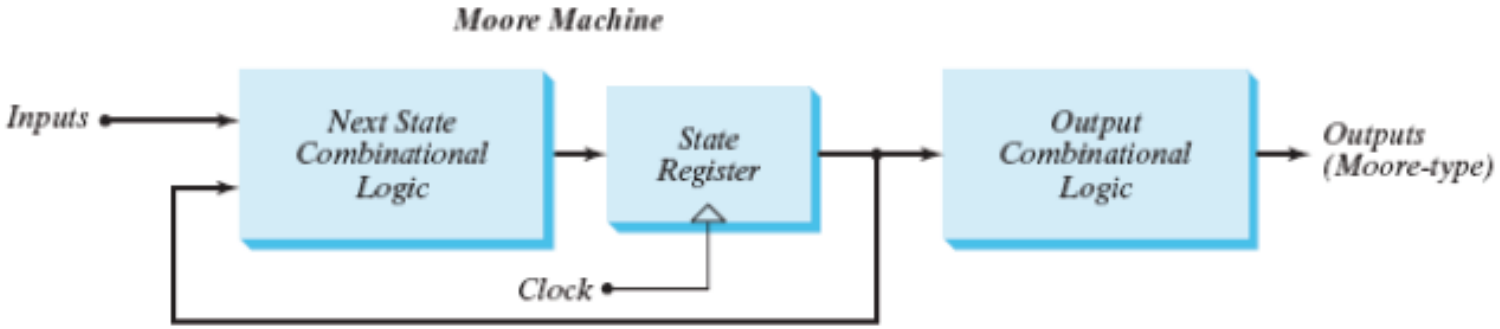




Two Flavors of FSM

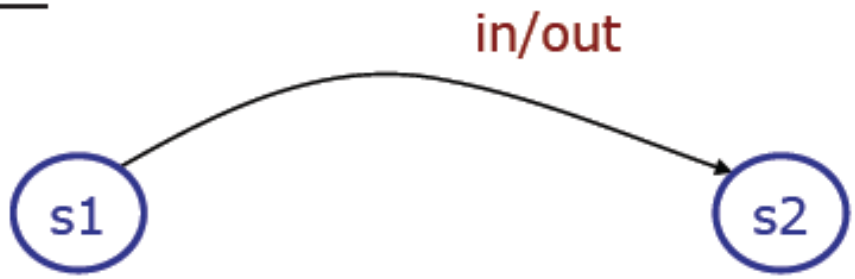
- Mealy vs Moore machine

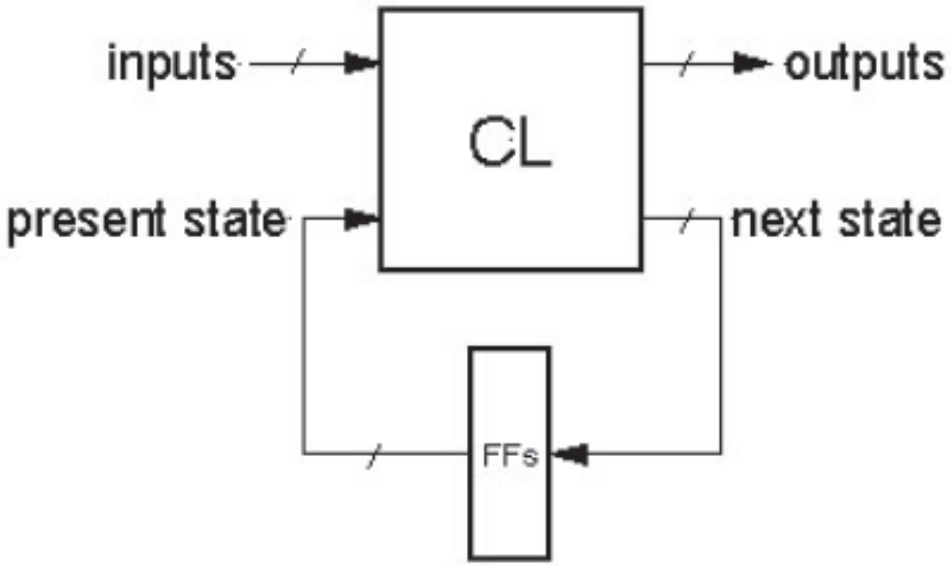




Mealy Machine

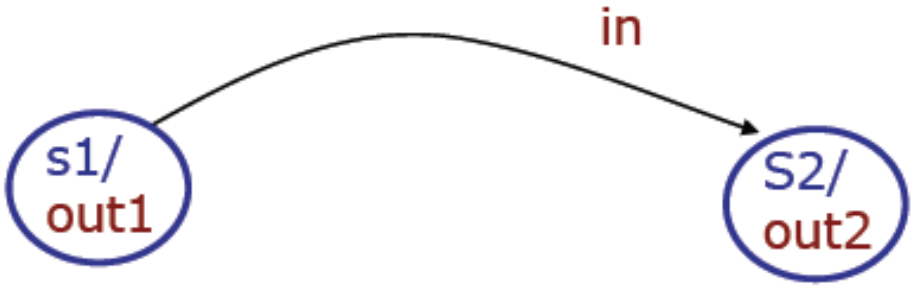
- Output based on state and present input
  - Output changes during transition

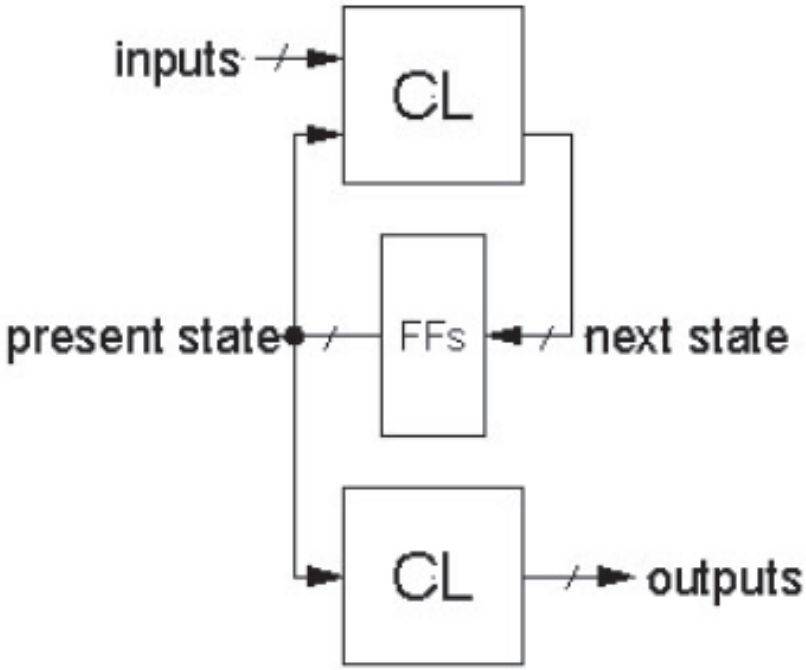




Moore Machine

- Output based on state only
  - Output is associated with state





Design of Sequential Circuits

- How can we design a sequential circuit?
  - E.g., circuit that detects 3 or more consecutive 1's in input

- Design procedure:
  1. Derive state diagram from description
  2. Reduce number of states if necessary
  3. Assign binary values to states
  4. Obtain binary coded state table (transition table)
  5. Choose type of flip-flops
  6. Derive flip-flop input equations and output equations
  7. Draw logic diagram

- Steps 1 & 3 require insight

- Steps 2, 4–7 can be automated
  - Design that follows well-defined procedure called synthesis

Canonical form of Sequential Circuits

- Graphs are hard to compare
  - Arbitrary state names
  - Arbitrary coding
- Graphs are generally very difficult to deal with
  - Determining if two graphs are identical is the "graph isomorphism problem"
    - » No known algorithm exists that can determine isomorphism in polynomial time for arbitrary graphs
    - » Problem reduces to trying every possible matching
    - » Requires exponential time (very, very long for large graphs)
- Canonical form of sequential circuit would require solution to graph isomorphism problem
- No canonical form for sequential circuits exists

State Assignment

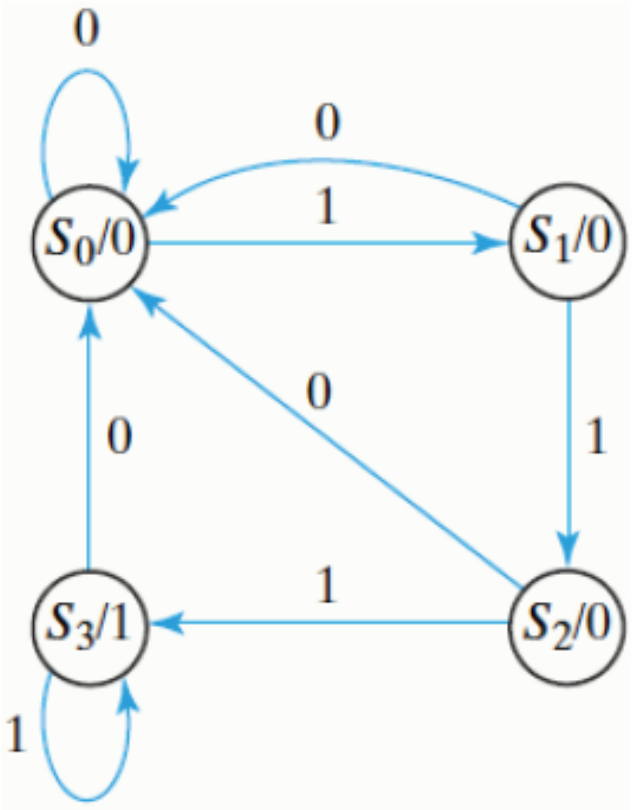
- States are represented by flip-flop values in circuit
  - Need to encode state in binary
- What is the minimum number of flip-flops that are necessary to encode m states?
  - Need at least $\lceil log_2(m) \rceil$ bits ($\lceil\ \rceil$ is "ceiling" function)
- Many possible encodings:
- Terminology:
  - "State table" uses uncoded states
  - "Transition table" uses coded states

State	Binary	Gray	One-hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

Example 1: Sequence Detector

- Circuit specification:
  - Design a circuit that outputs a 1 when three consecutive 1s have been applied to input, and 0 otherwise."

- Step 1: derive state diagram
  - What should a state represent?
    - » E.g., "number of 1's seen so far"
  - Moore or Mealy FSM?
    - » Both possible
    - » Chose Moore to simplify diagram
  - State diagram:
    - » State S0: zero 1s detected
    - » State S1: one 1 detected
    - » State S2: two 1s detected
    - » State S3: three 1s detected

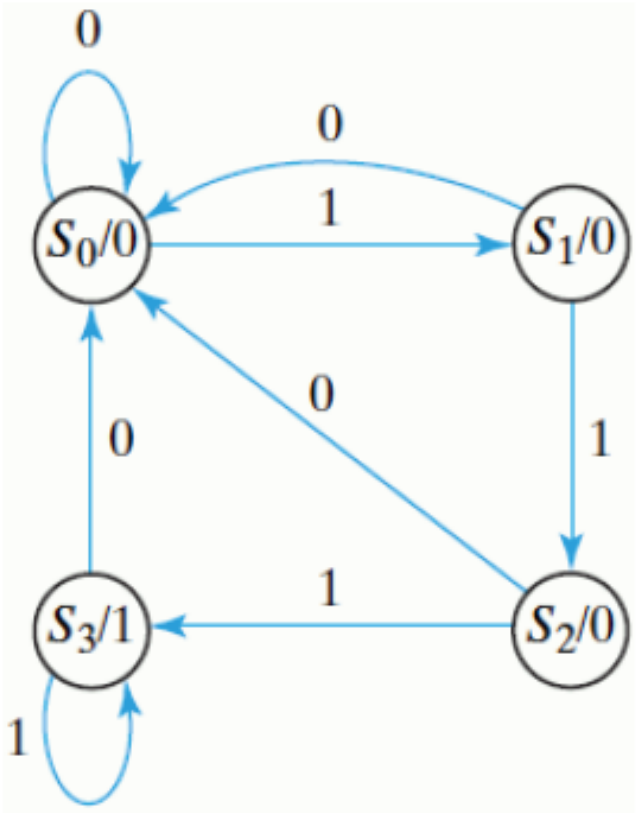


Example 1: Sequence Detector

- Step 2: reduce number of states
  - State table:

current state	next state x=0	next state x=1	output
S_0	S_0	S_1	0
S_1	S_0	S_2	0
S_2	S_0	S_3	0
S_3	S_0	S_3	1

  - Which states are equivalent?
    - » None – no state reduction possible



- Step 3: state assignment
  - Two flip-flops
  - Binary state coding

Example 1: Sequence Detector

- Step 4: Binary coded state table
  - Name flip-flops A and B

current state		next state				output
		x=0		x=1		
A	B	A	B	A	B	
0	0	0	0	0	1	0
0	1	0	0	1	0	0
1	0	0	0	1	1	0
1	1	0	0	1	1	1

- Step 5: Choose type of flip-flops
  - E.g., D flip-flop
  - Characteristic equation: $Q(t+1)=D_Q$

Example 1: Sequence Detector

- Step 6: derive flip-flop input equations and output equation
  - Use state table

  - $A(t+1) = D_A(A,B,x) = \Sigma(3,5,7)$

  - $B(t+1) = D_B(A,B,x) = \Sigma(1,5,7)$

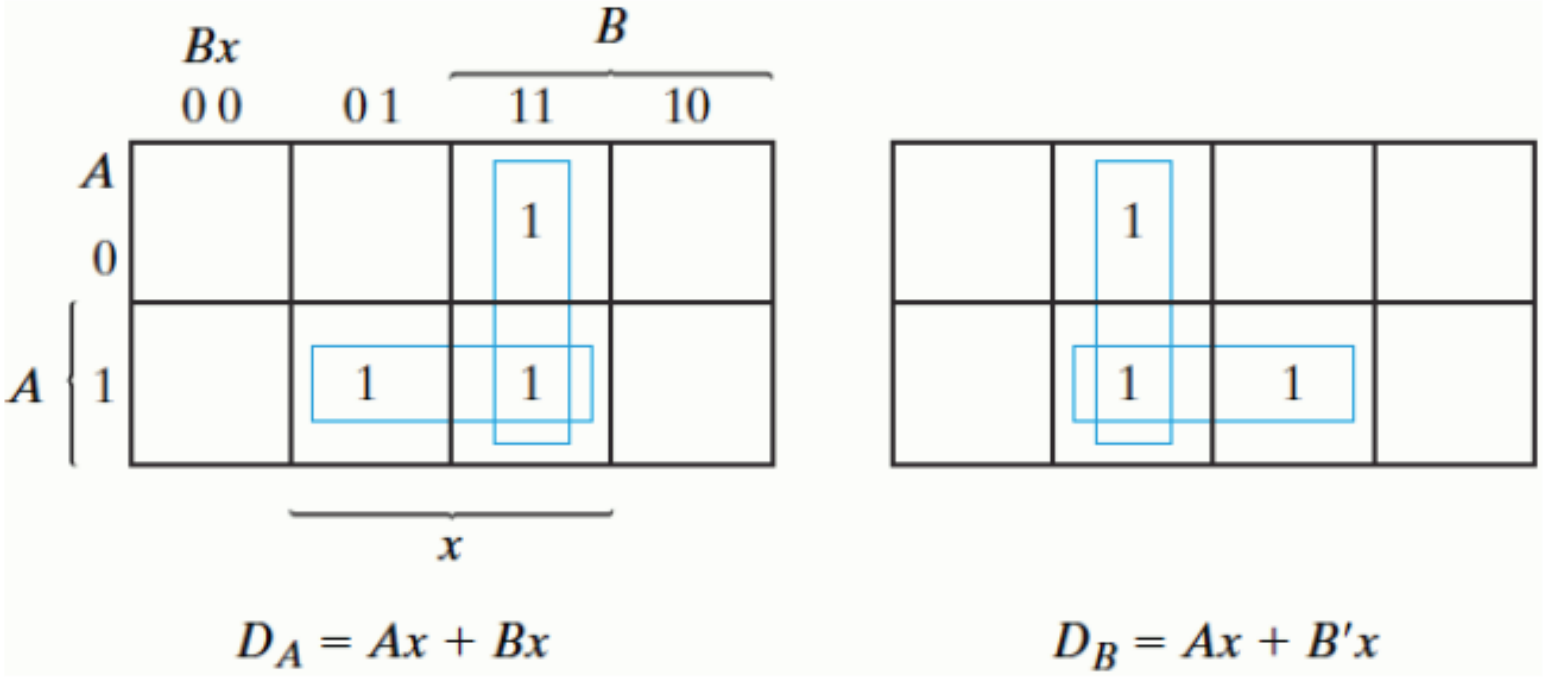
  - $y(A,B,x) = \Sigma(6,7)$ or $y(A,B) = \Sigma(3)$

current state		input	next state		output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Example 1: Sequence Detector

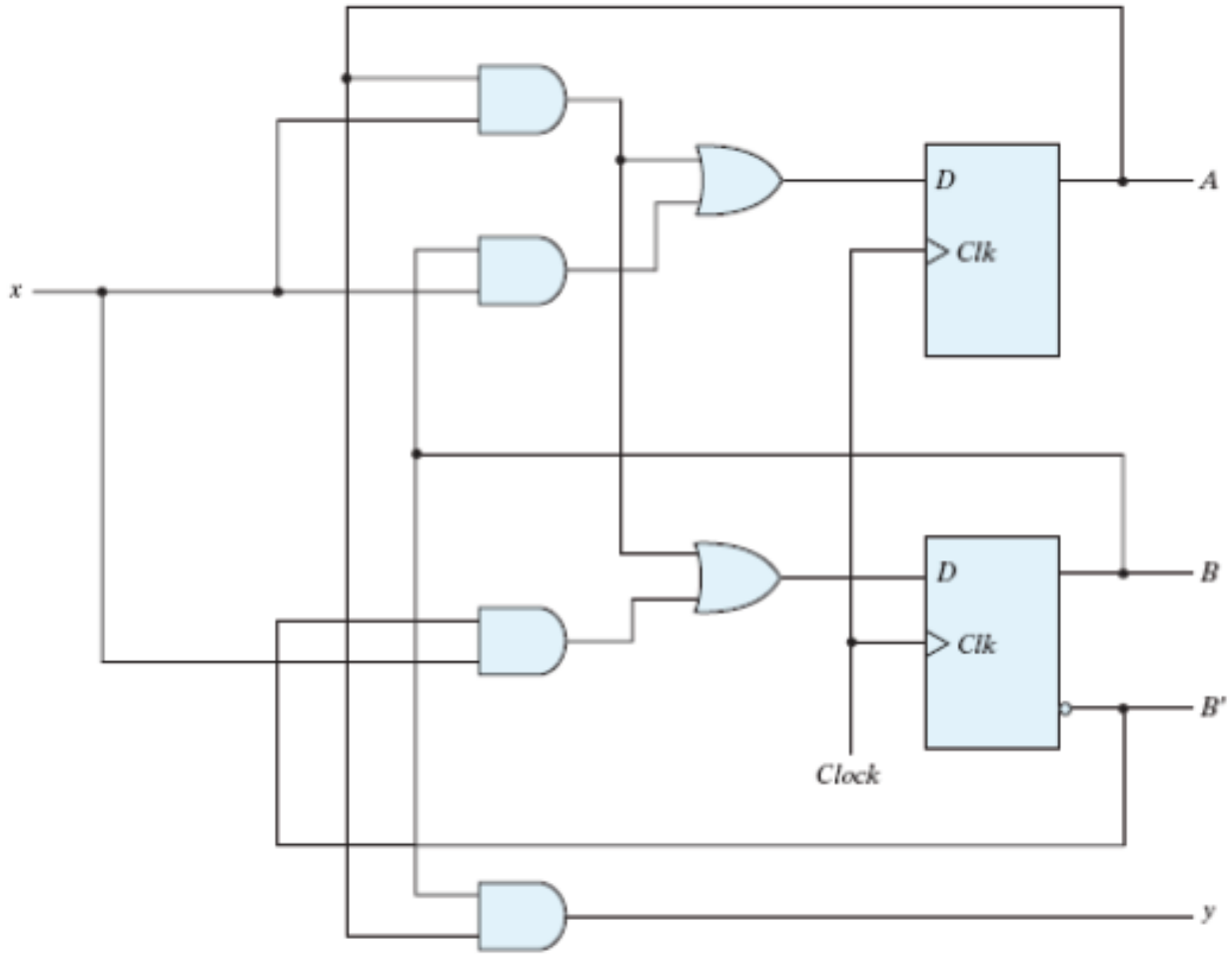
- Step 6b: minimize equations
  - $A(t+1) = \Sigma(3,5,7)$
  - $B(t+1) = \Sigma(1,5,7)$
  - $y(A,B) = \Sigma(3)$ – easy: $y = AB$

- Karnaugh maps for A and B:

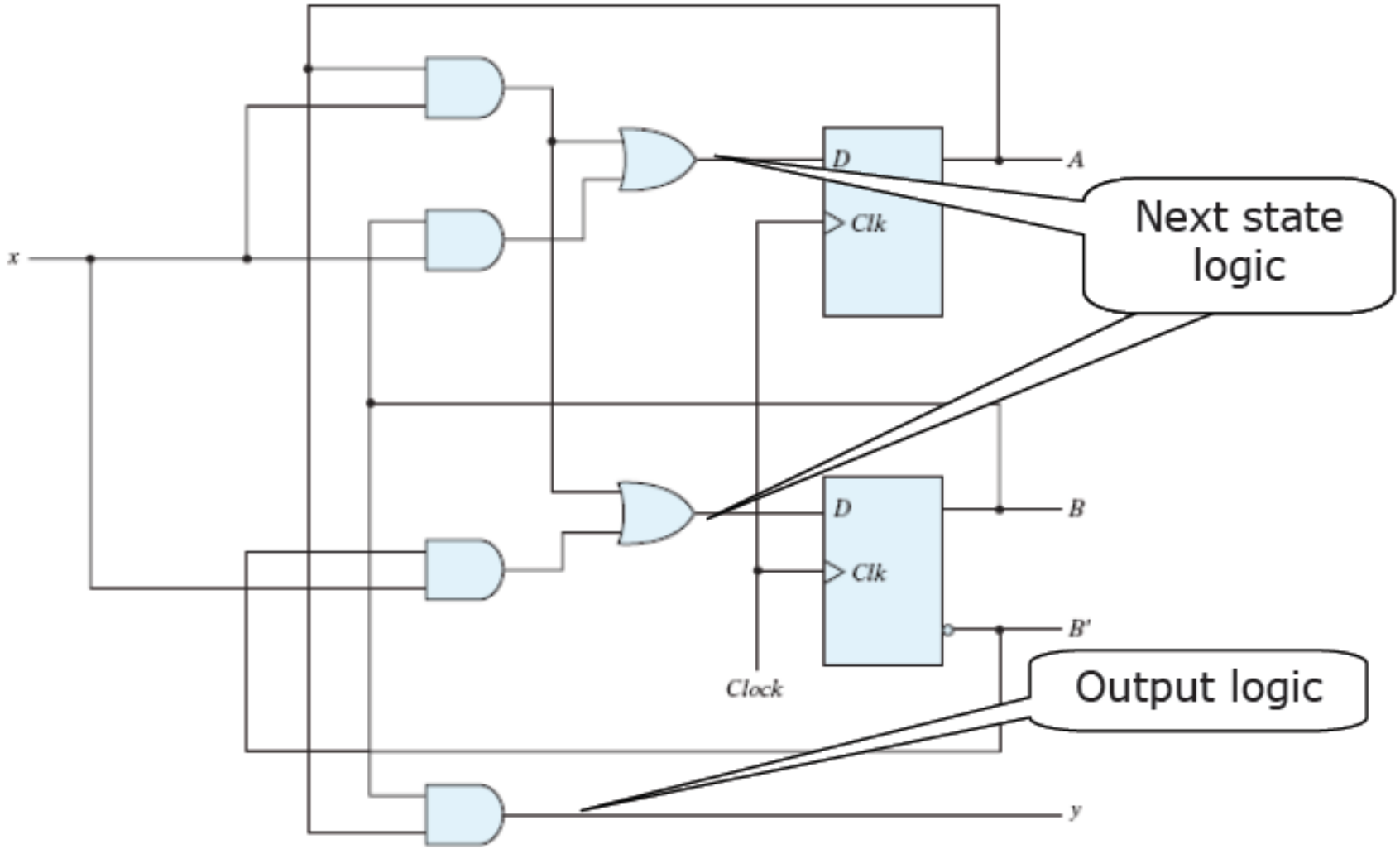


Example 1: Sequence Detector

- Step 7: Circuit diagram
  - $D_A = Ax + Bx$
  - $D_B = Ax + B'x$
  - $Y = AB$



Terminology



Questions - 1

- (Q.5.5) A sequential circuit with two D flip-flops A and B, two inputs x and y, and one output z is specified by the following next state and output equations

  A(t+1)=x'y+xB

  B(t+1)=x'A+xB

  Z=A

  a) Draw the logic diagram of the circuit.
  b) List the state table for the sequential circuit.
  c) Draw the corresponding state diagram.

Questions - 2

- (Q.5.12) Design a sequential circuit with two D flip-flops A and B and one input x_in.
  a) When x_in=0, the state of the circuit remains the same. When x_in=1, the circuit goes through the state transitions from 00 to 01, to 11, to 10, back to 00, and repeats.
  b) When x_in=0, the state of the circuit remains the same. When x_in=1, the circuit goes through the state transitions from 00 to 11, to 01, to 10, back to 00, and repeats.

Questions - 3

- (Q.5.14) Design a sequential circuit with two D flip-flops A and B and two inputs E and F.
  - If E=0, the circuit remains in the same state regardless of the value of F.
  - When E=1 and F=1, the circuit goes through the state transitions from 00 to 01, to 10, to 11, back to 00, and repeats.
  - When E=1 and F=0, the circuit goes through the state transitions from 00 to 11, to 10, to 01, back to 00, and repeats.
    - (Up and down counter with enable. Count up when F=1, count down when F=0.)

Questions - 4

- (Q.5.15) A sequential circuit has three flip-flops A,B,and C; one input x_in; and one output y_out. The state diagram is shown at right. The circuit is to be designed by treating the unused states as don't-care conditions. Analyze the circuit obtained from the design to determine the effect of the unused states.