

CpE358/CS381

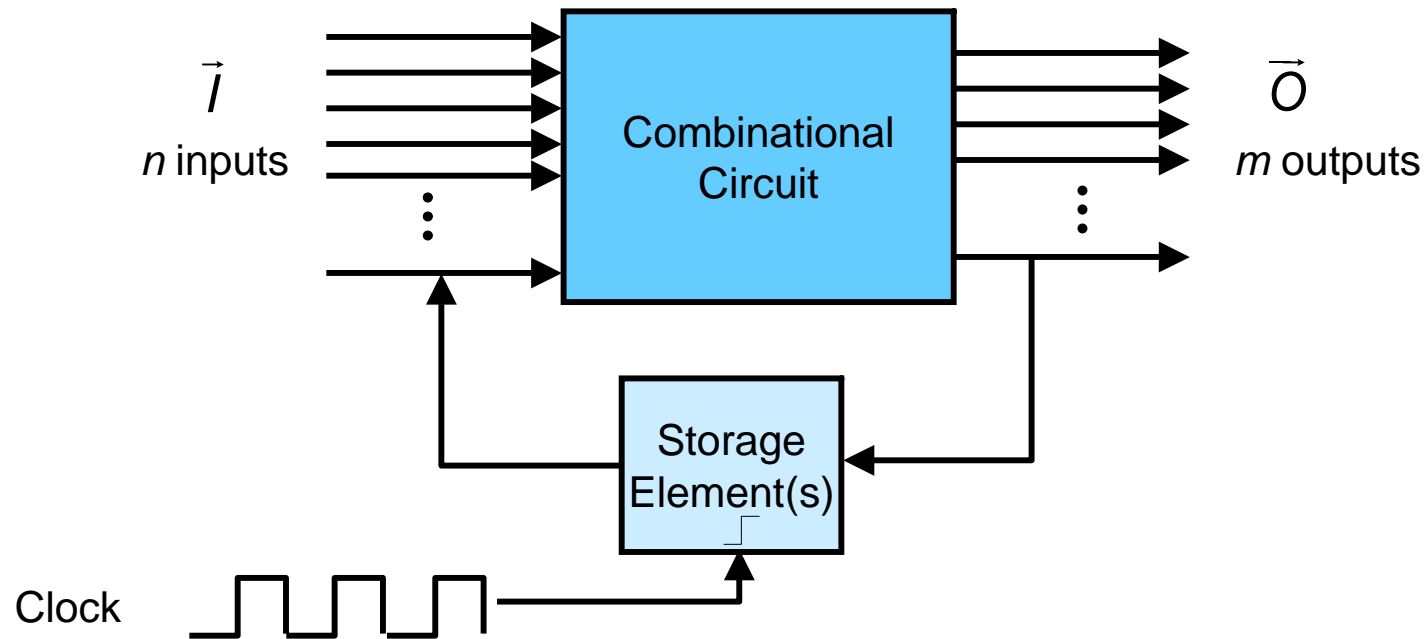
**Switching Theory and
Logical Design**

Class 7

Today

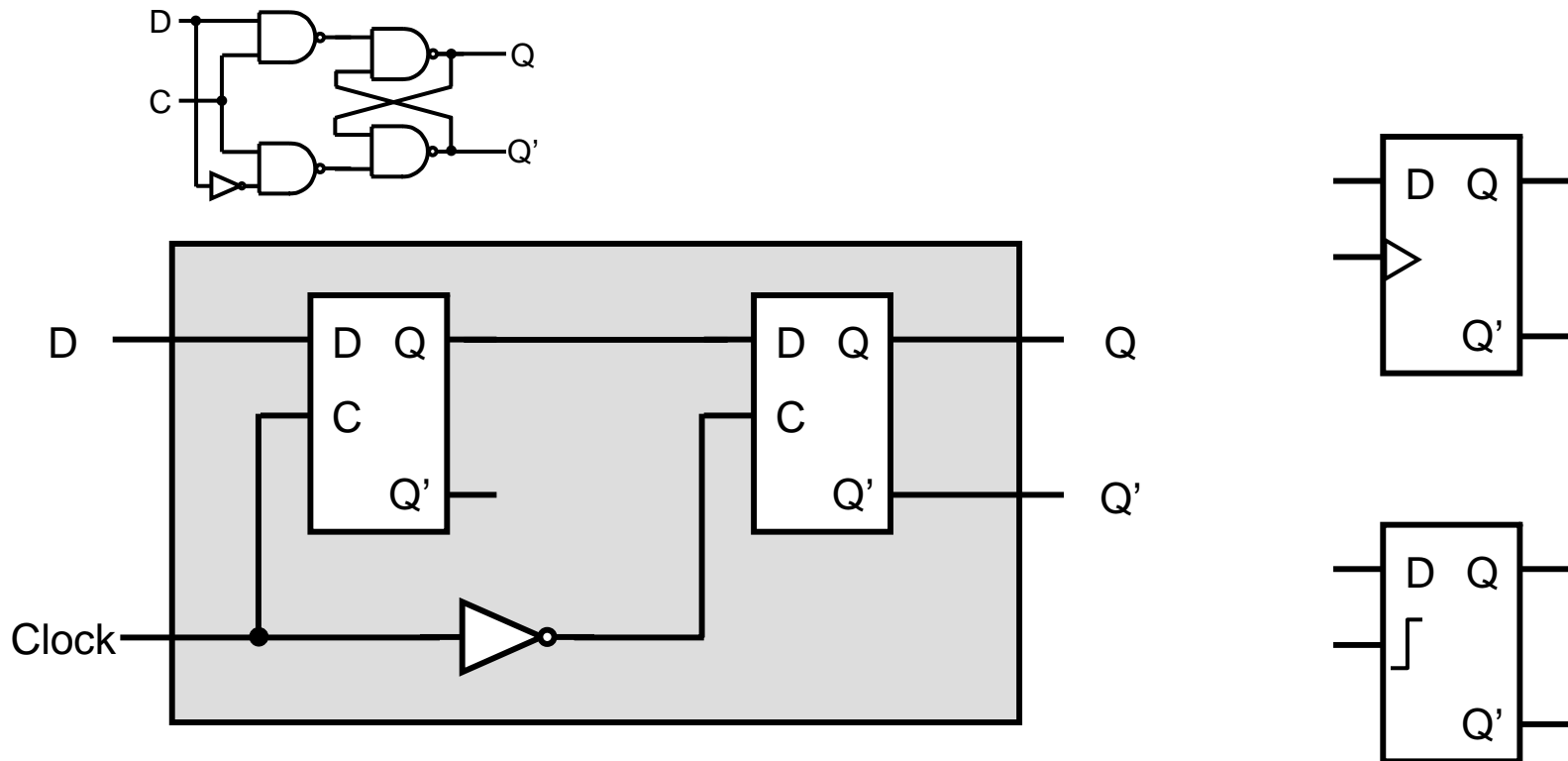
- Fundamental concepts of digital systems (Mano Chapter 1)
- Binary codes, number systems, and arithmetic (Ch 1)
- Boolean algebra (Ch 2)
- Simplification of switching equations (Ch 3)
- Digital device characteristics (e.g., TTL, CMOS)/design considerations (Ch 10)
- Combinatoric logical design including LSI implementation (Chapter 4)
- Flip-flops and state memory elements (Ch 5)
- **Sequential logic analysis and design (Ch 5)**
- Hazards, Races, and time related issues in digital design (Ch 9)
- Synchronous vs. asynchronous design (Ch 9)
- Counters, shift register circuits (Ch 6)
- Memory and Programmable logic (Ch 7)
- Minimization of sequential systems
- Introduction to Finite Automata

Synchronous Sequential Circuit



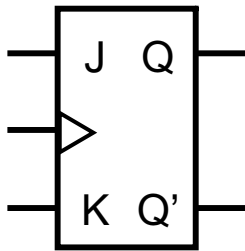
Master-Slave D Flip-Flop

$$Q(t+1) = D$$



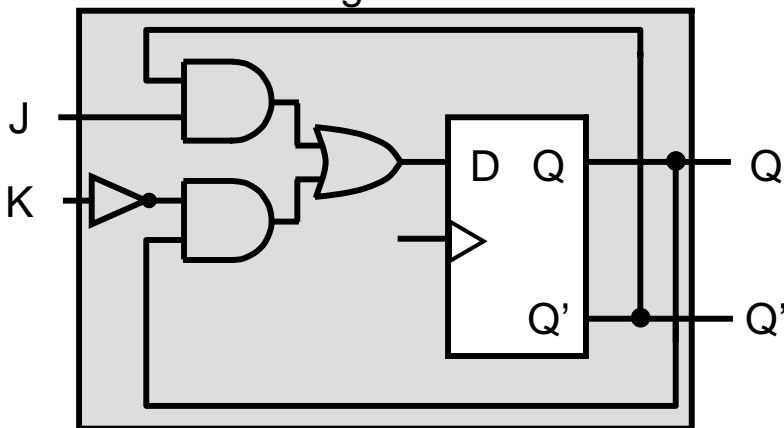
J-K Flip Flop

$$Q(t+1) = JQ' + K'Q$$

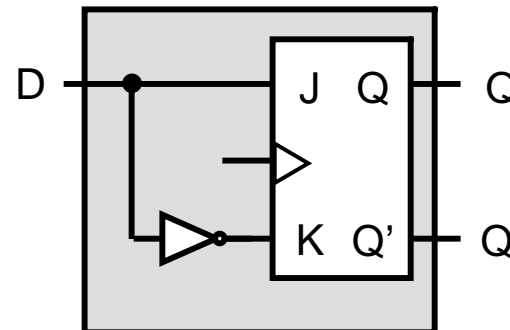


J	K	Q_{i+1}	Meaning
0	0	Q_i	No change
1	0	1	Clockin 1
0	1	0	Clockin 0
1	1	Q_i'	Toggle

Constructing a J-K from a D

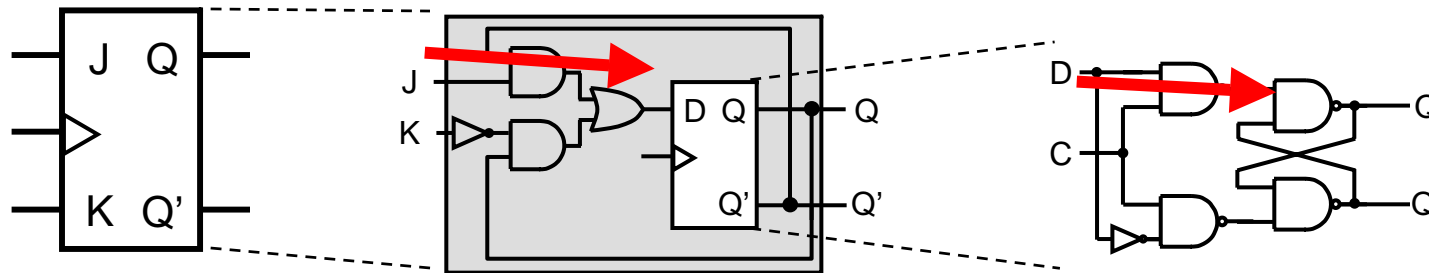


Constructing a D from a J-K



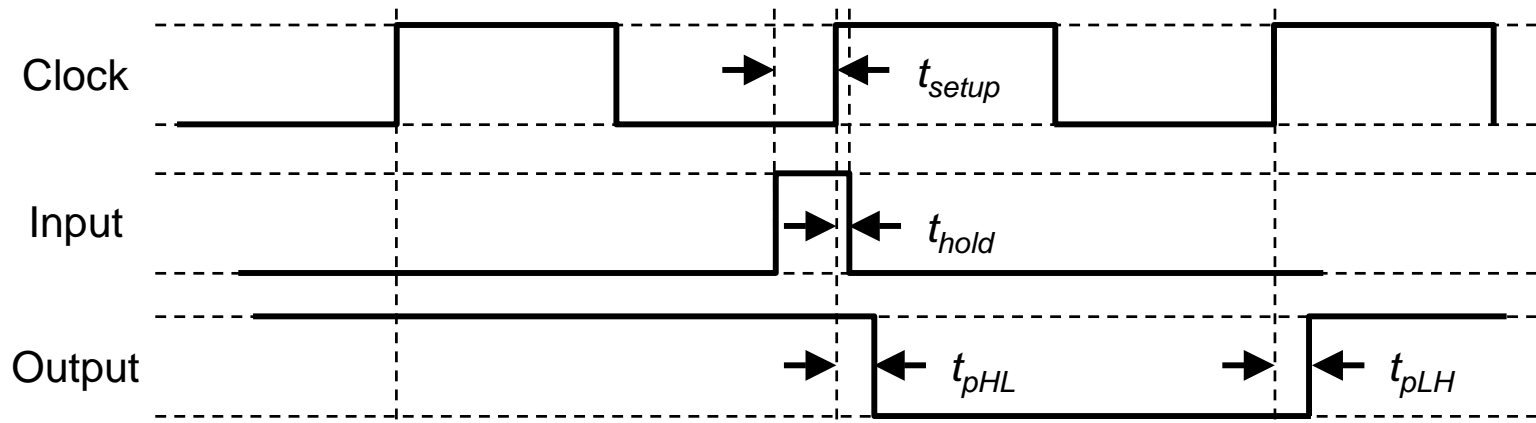
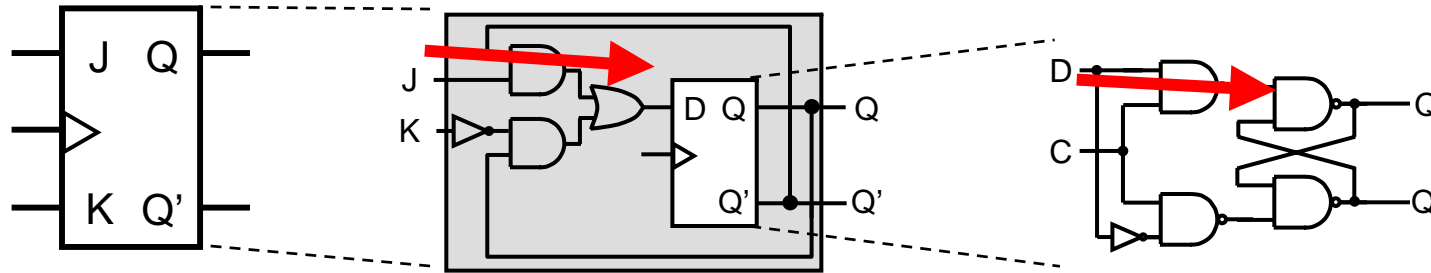
Flip-flop Timing Considerations

- There are propagation delays due to the internal logic of any flip-flop



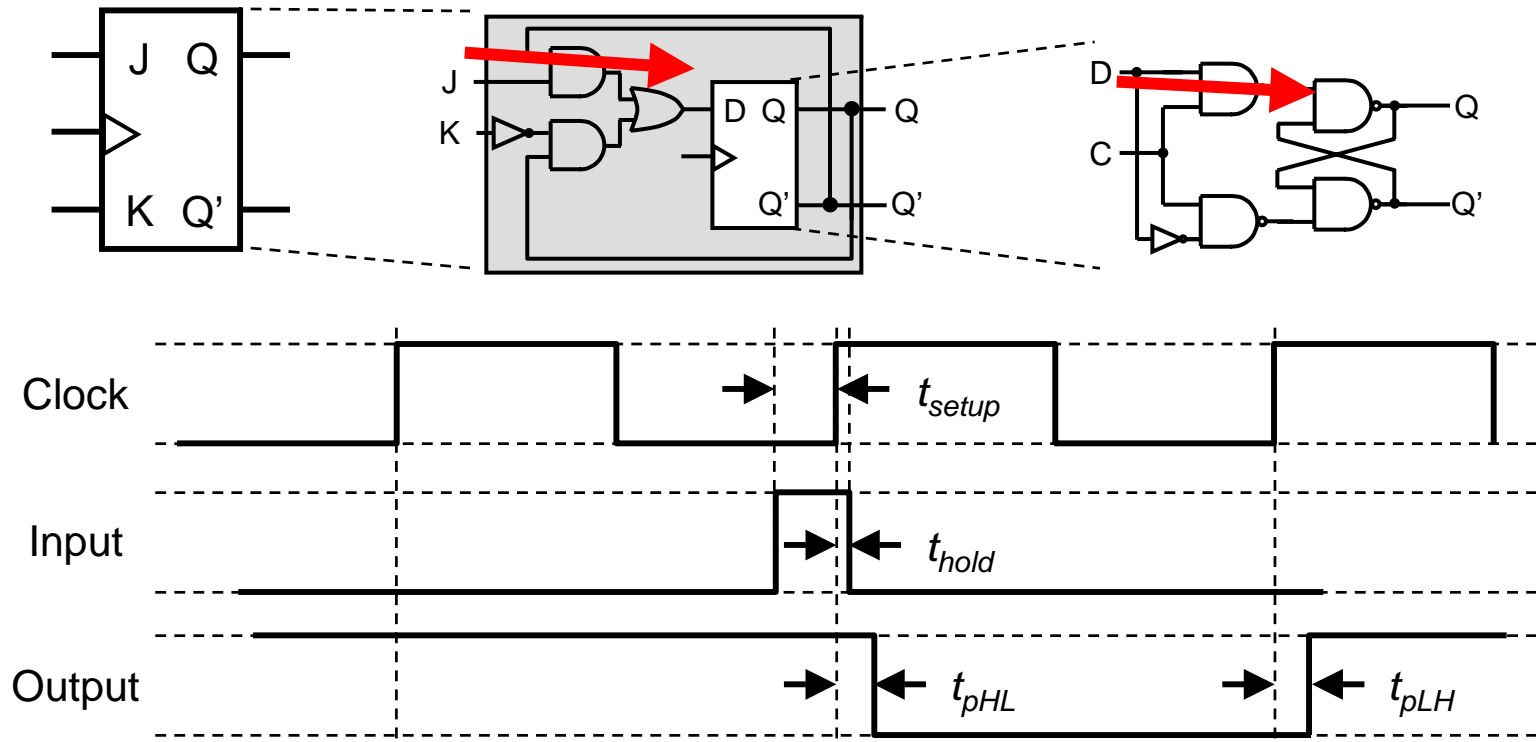
Flip-flop Timing Considerations

- There are propagation delays due to the internal logic of any flip-flop



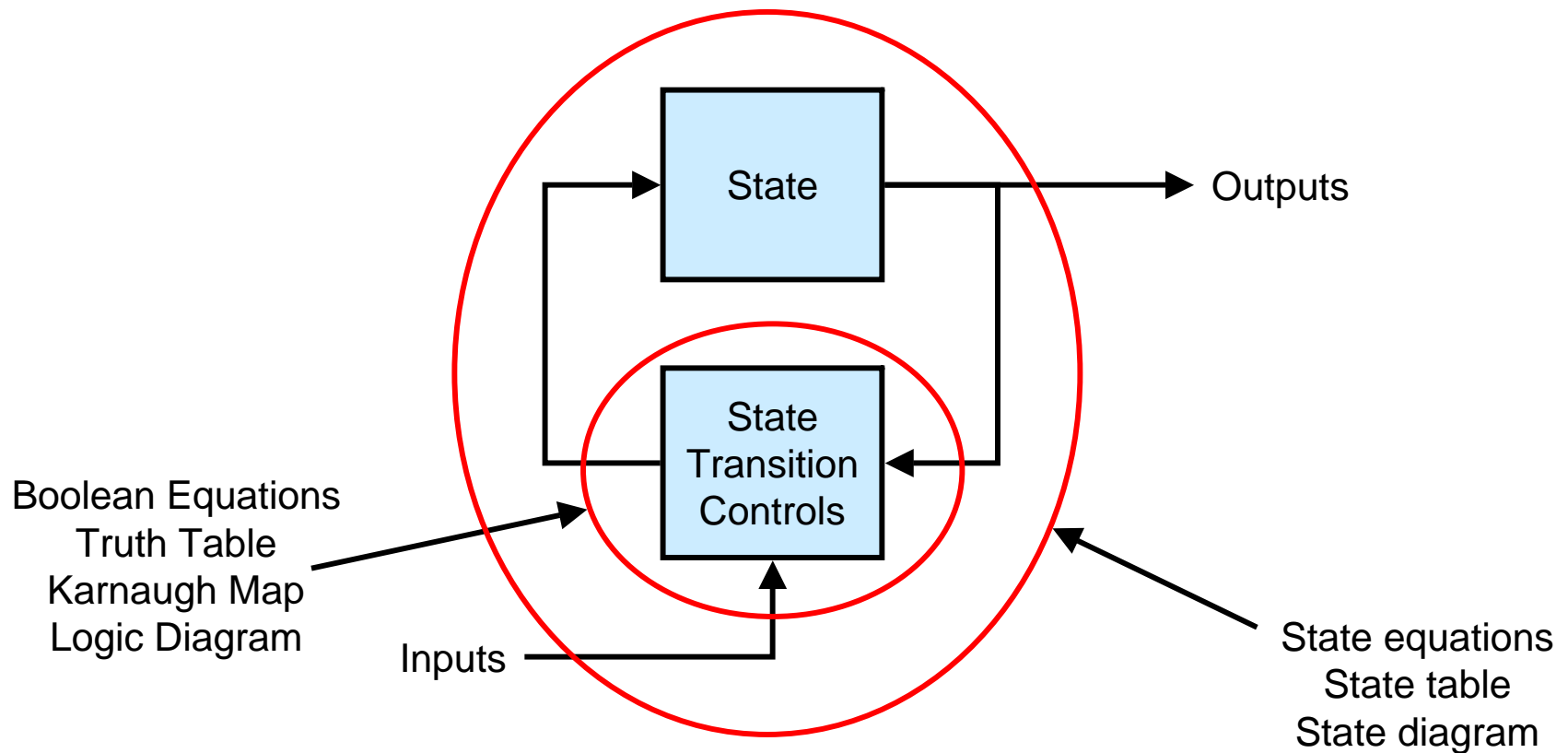
Flip-flop Timing Considerations

- There are propagation delays due to the internal logic of any flip-flop

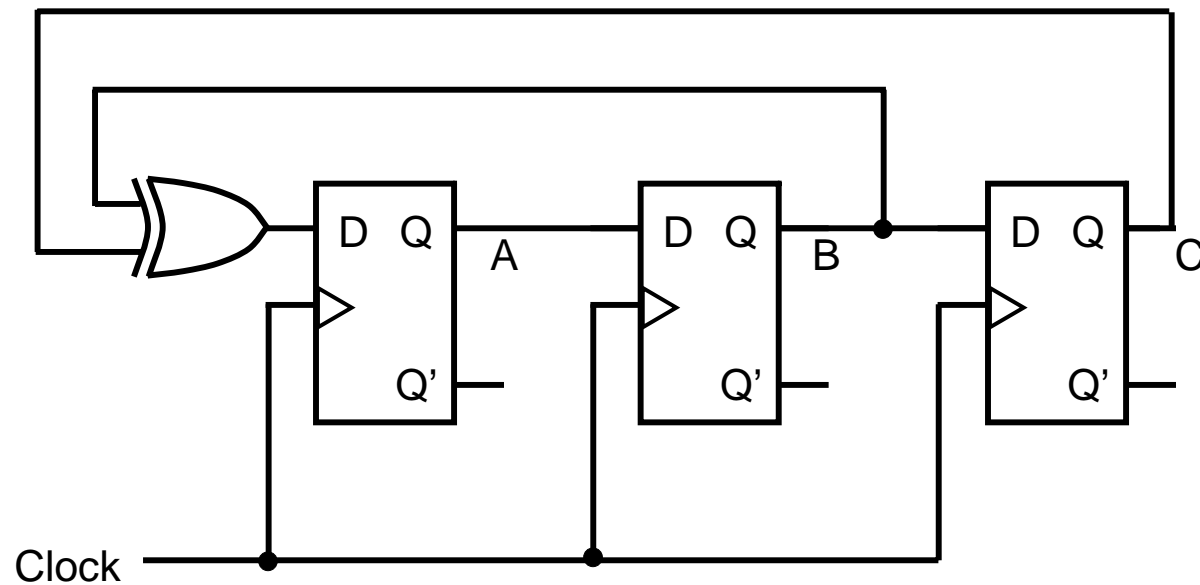


- **Device operation may not be as expected if setup and hold time requirements are not observed**

Abstraction of Sequential Circuit

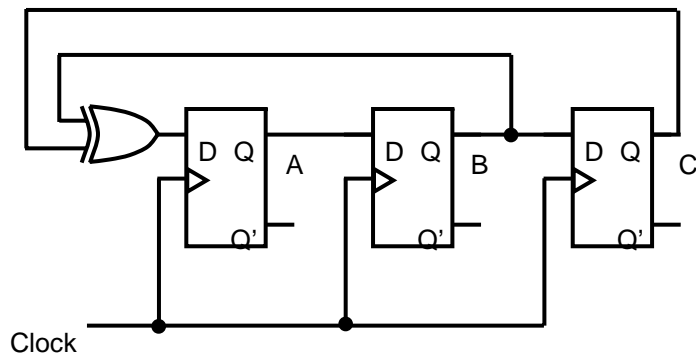


Typical Sequential Circuit With No Inputs



Typical Sequential Circuit With No Inputs

- Description by State Equations



$$A(t+1) = B(t) \oplus C(t)$$

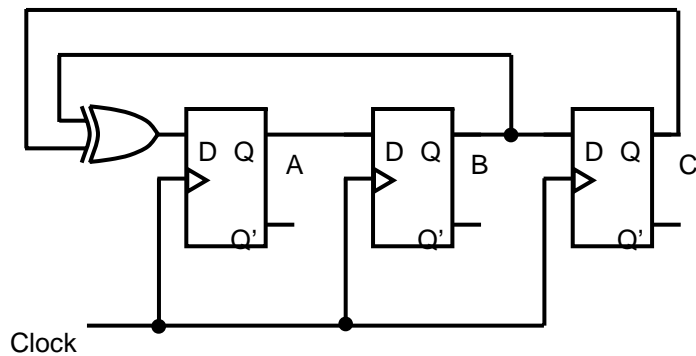
$$B(t+1) = A(t)$$

$$C(t+1) = B(t)$$

$$\begin{aligned} A(t+1) &= B(t+1) = A(t) \\ C(t+1) &= B(t) \end{aligned}$$

Typical Sequential Circuit With No Inputs

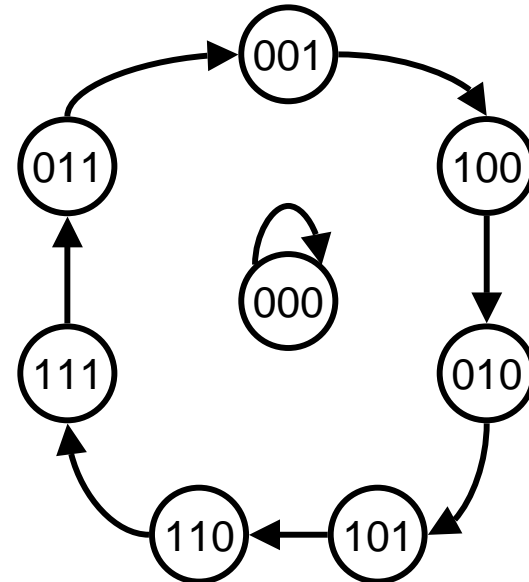
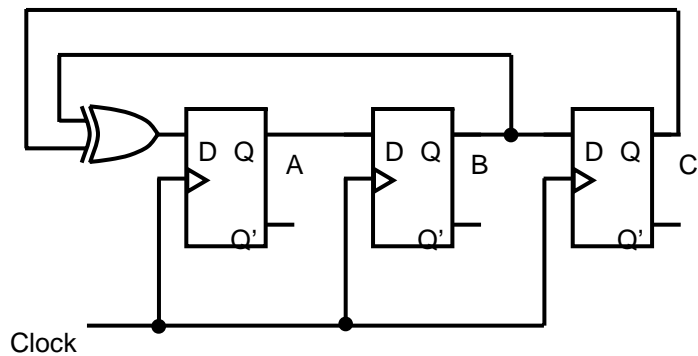
- Description by State Table



Present State			Input	Next State			Output
A	B	C	-	A	B	C	-
0	0	0	-	0	0	0	-
0	0	1	-	1	0	0	-
0	1	0	-	1	0	1	-
0	1	1	-	0	0	1	-
1	0	0	-	0	1	0	-
1	0	1	-	1	1	0	-
1	1	0	-	1	1	1	-
1	1	1	-	0	1	1	-

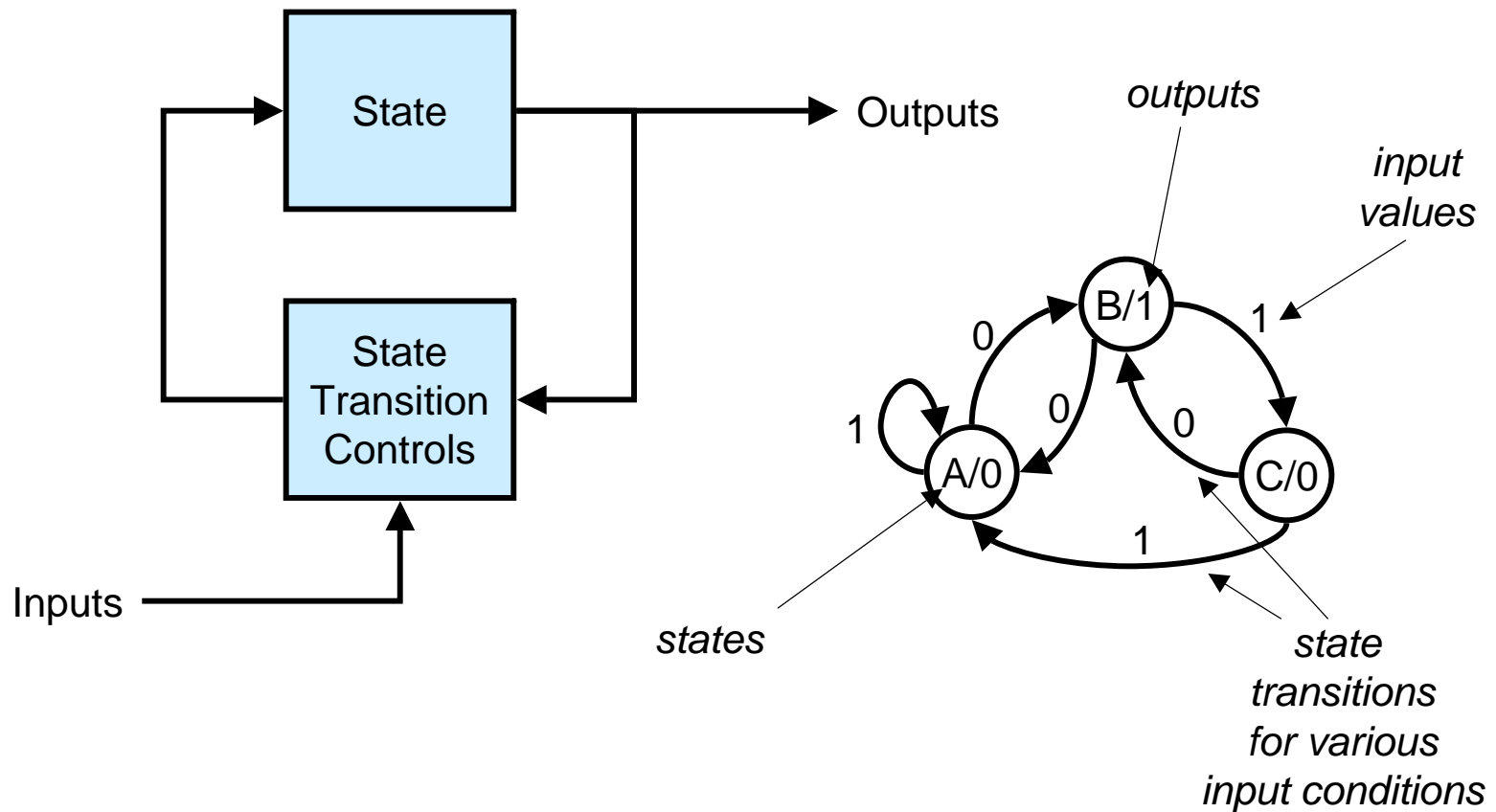
Typical Sequential Circuit With No Inputs

- Description by State Diagram



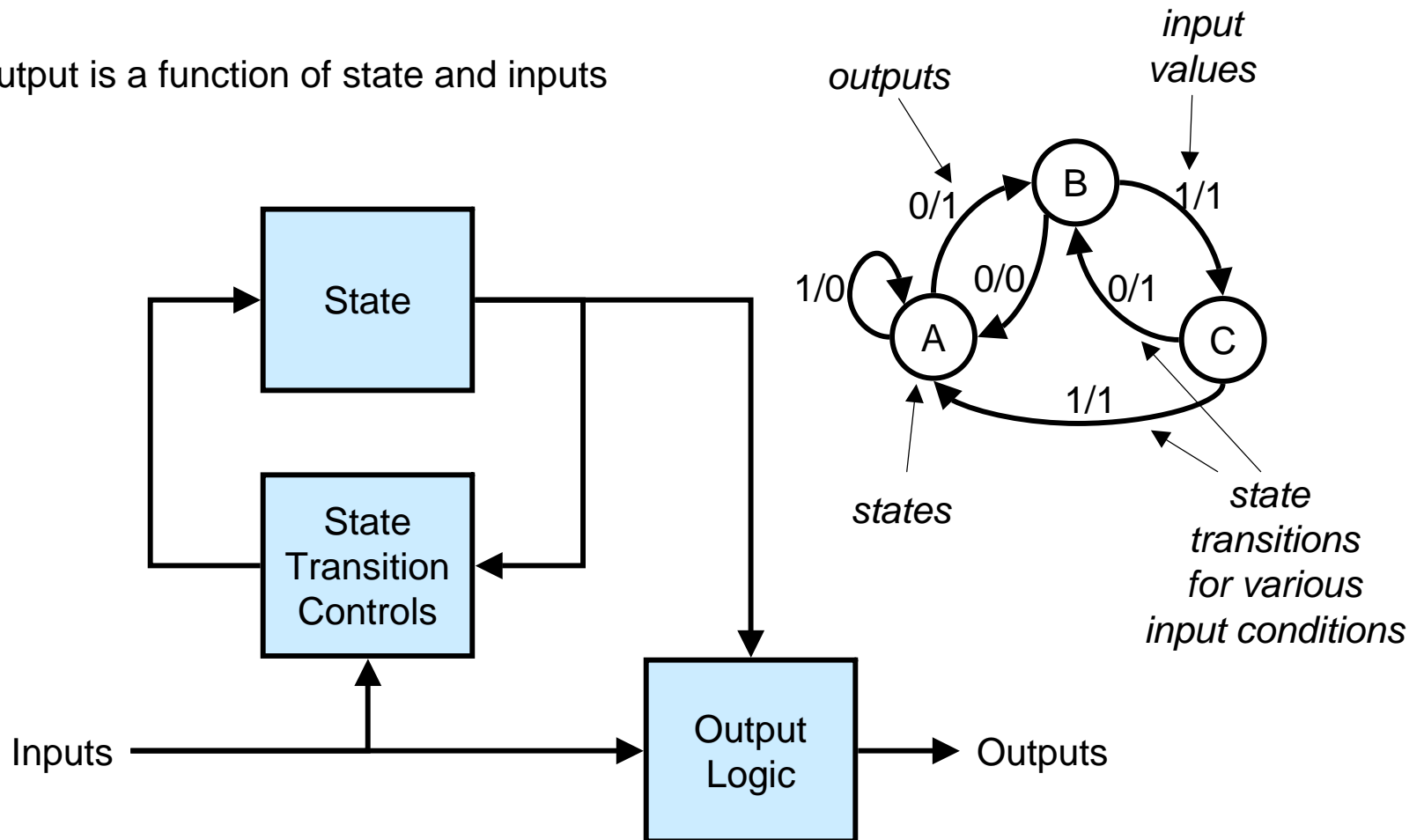
Sequential Machine – Moore Models

- Output is a function of state only



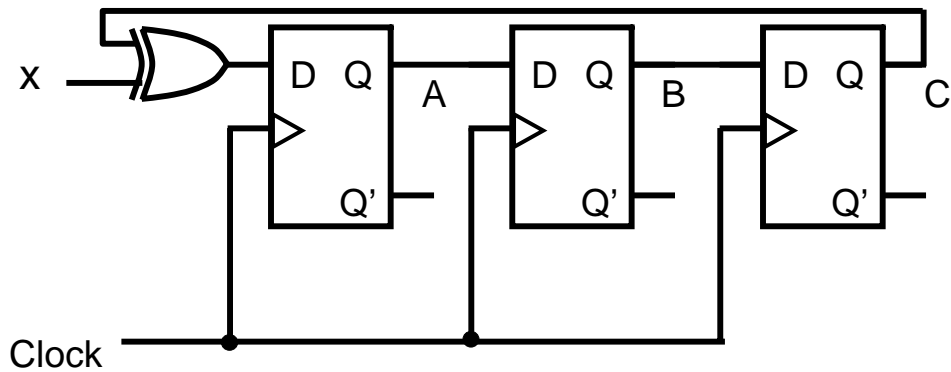
Sequential Machine – Mealy Models

- Output is a function of state and inputs



Typical Sequential Circuit

- Description by State Equations



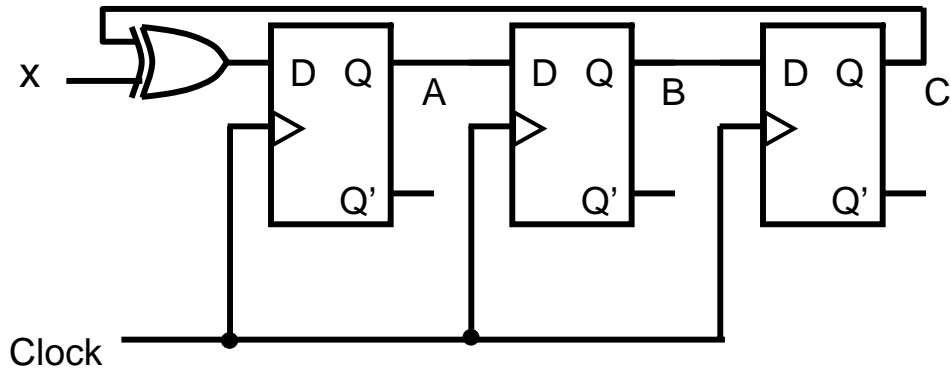
$$A(t+1) = x \oplus C(t)$$

$$B(t+1) = A(t)$$

$$C(t+1) = B(t)$$

Typical Sequential Circuit

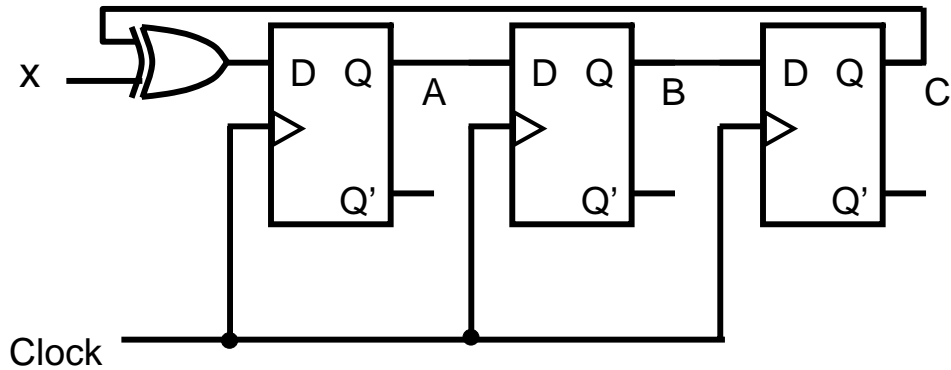
- Description by State Table



Present State			Input X	Next State			Output
A	B	C		A	B	C	
0	0	0	0				-
0	0	0	1				-
0	0	1	0				-
0	0	1	1				-
0	1	0	0				-
0	1	0	1				-
0	1	1	0				-
0	1	1	1				-
1	0	0	0				-
1	0	0	1				-
1	0	1	0				-
1	0	1	1				-
1	1	0	0				-
1	1	0	1				-
1	1	1	0				-
1	1	1	1				-

Typical Sequential Circuit

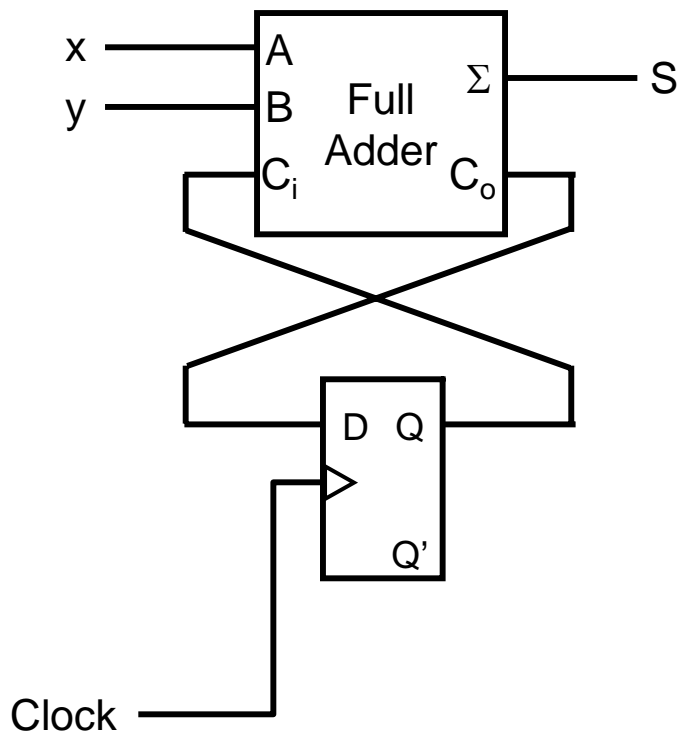
- Description by State Table



Present State			Input X	Next State			Output
A	B	C		A	B	C	
0	0	0	0		0	0	-
0	0	0	1		0	0	-
0	0	1	0		0	0	-
0	0	1	1		0	0	-
0	1	0	0		0	1	-
0	1	0	1		0	1	-
0	1	1	0		0	1	-
0	1	1	1		0	1	-
1	0	0	0		1	0	-
1	0	0	1		1	0	-
1	0	1	0		1	0	-
1	0	1	1		1	0	-
1	1	0	0		1	1	-
1	1	0	1		1	1	-
1	1	1	0		1	1	-
1	1	1	1		1	1	-

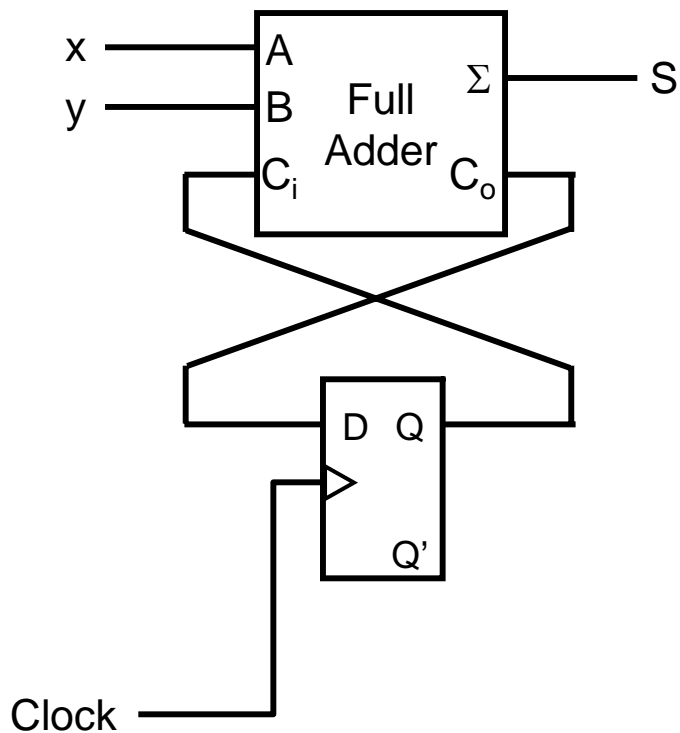
Problem 5-7

- Derive **state equations**, state table and state diagram
- **Is this a Mealy or Moore machine?**
- What function does this sequential circuit perform?



Problem 5-7

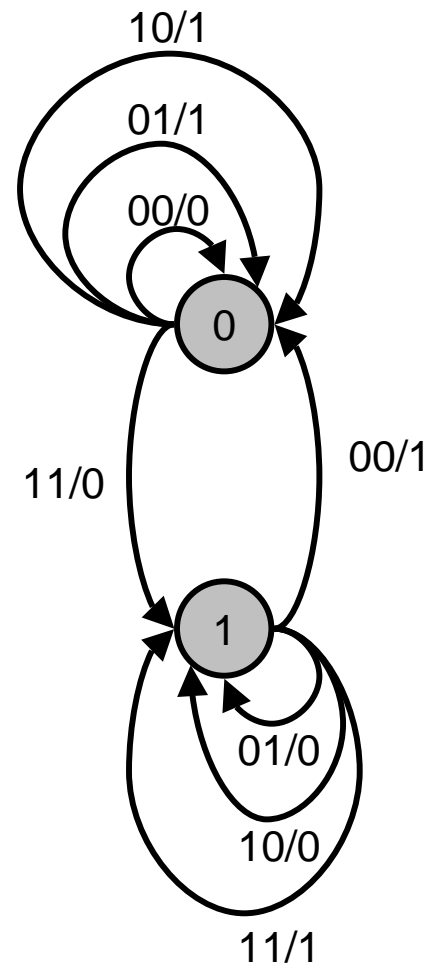
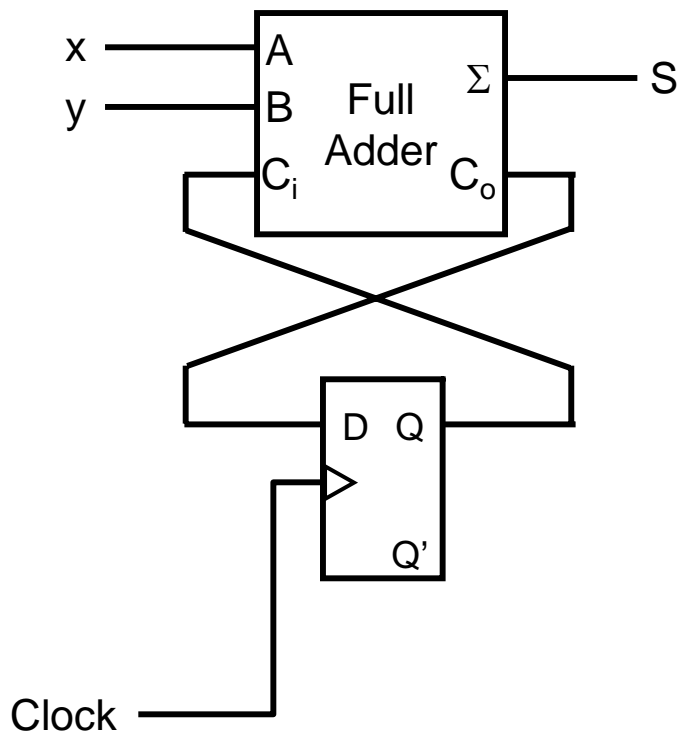
- Derive **state equations**, state table and state diagram
- Is this a Mealy or Moore machine?
- What function does this sequential circuit perform?



Present State	Input		Next State	Output
$C_i(t)$	X	Y	$C_i(t+1)$	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

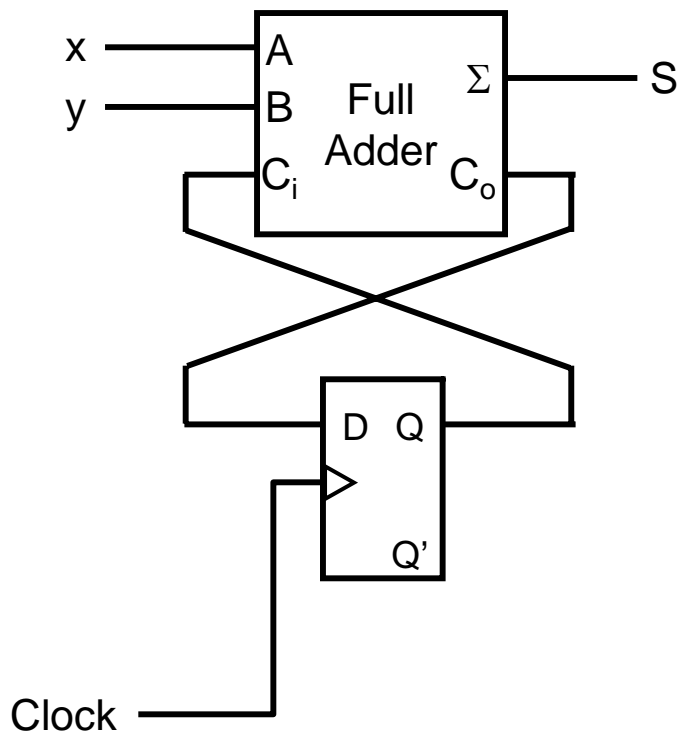
Problem 5-7

- Derive **state equations**, state table and state diagram
- Is this a Mealy or Moore machine?
- What function does this sequential circuit perform?



Problem 5-7

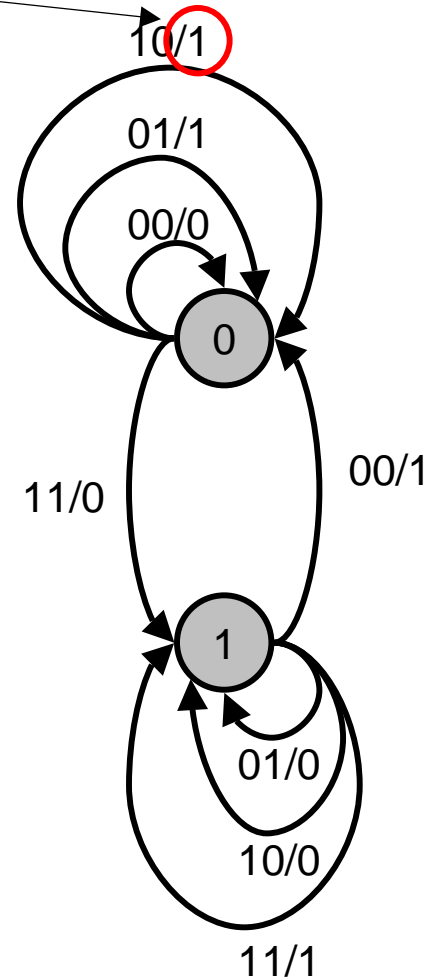
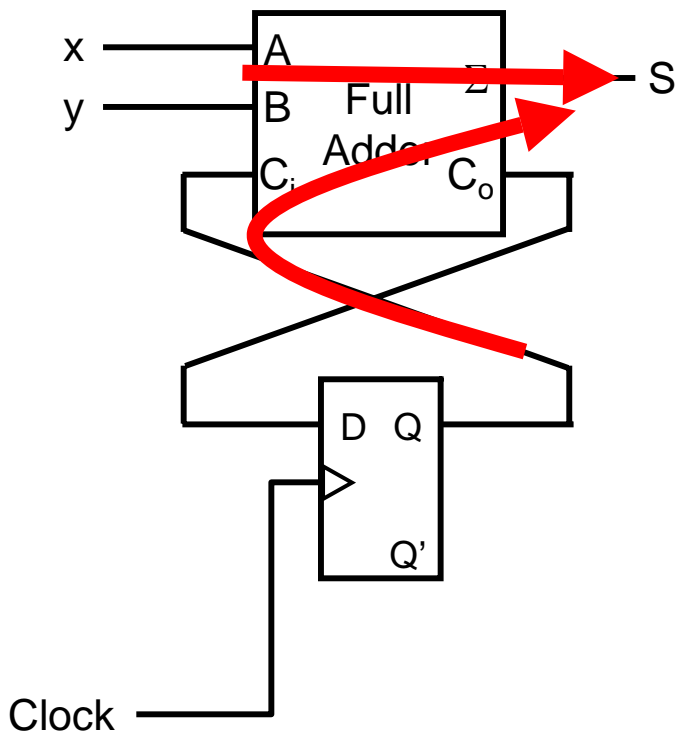
- Derive state equations, state table and state diagram
- Is this a Mealy or Moore machine?
- What function does this sequential circuit perform?



$$C_i(t+1) = C_o(t) = xy + xC_i(t) + yC_i(t)$$
$$S = x \oplus y \oplus C_i$$

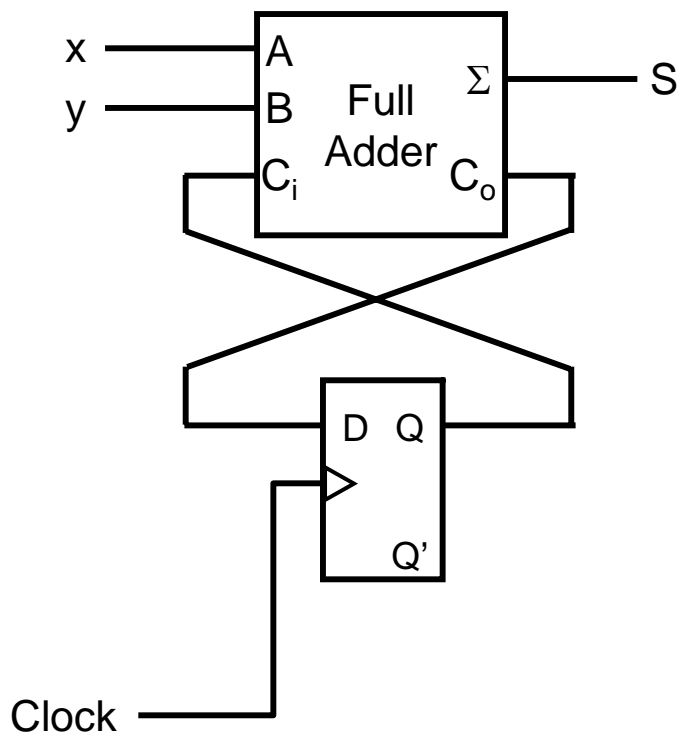
Problem 5-7

- Derive **state equations**, state table and state diagram
- **Is this a Mealy or Moore machine?**
- What function does this sequential circuit perform?



Problem 5-7

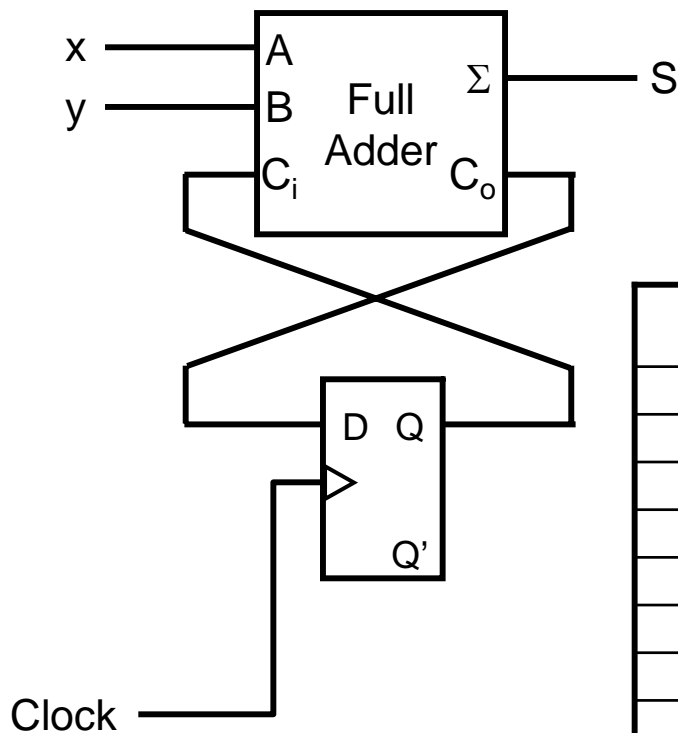
- Derive **state equations**, state table and state diagram
- Is this a **Mealy** or **Moore machine**?
- **What function does this sequential circuit perform?**



1. Assume initial state is 0
2. Assume x and y change in sync with clock
3. Consider the input sequences:
x = 010010
y = 110100

Problem 5-7

- Derive **state equations**, state table and state diagram
- Is this a **Mealy** or **Moore** machine?
- What function does this sequential circuit perform?



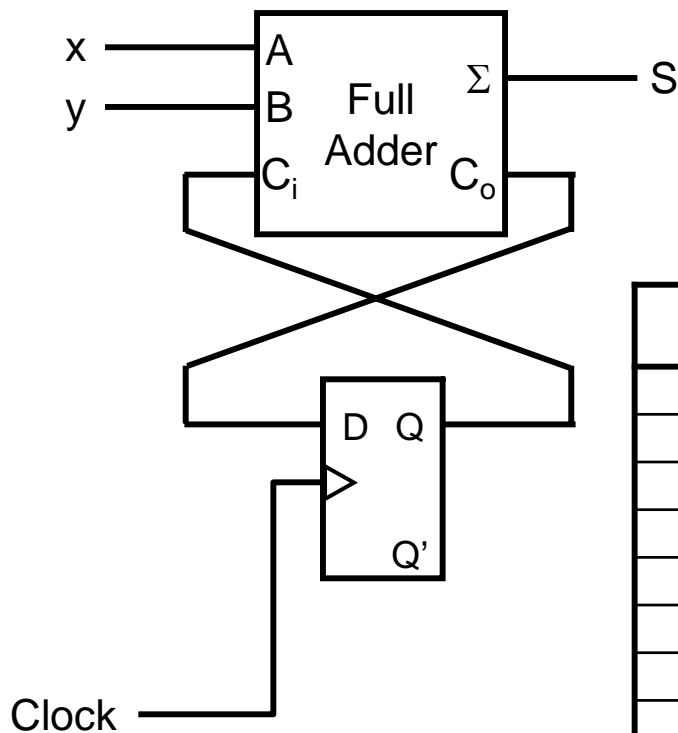
1. Assume initial state is 0
2. Assume x and y change in sync with clock
3. Consider the input sequences:
 $x = 010010$
 $y = 110100$

Present State	Input		Next State	Output
	X	Y		
$C_i(t)$	X	Y	$C_i(t+1)$	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_i	x	y	S	C_o
0	0	1		
	1	1		
	0	0		
	0	1		
	1	0		
	0	0		

Problem 5-7

- Derive **state equations**, state table and state diagram
- Is this a **Mealy** or **Moore** machine?
- **What function does this sequential circuit perform?**



1. Assume initial state is 0
2. Assume x and y change in sync with clock
3. Consider the input sequences:
 $x = 010010$
 $y = 110100$

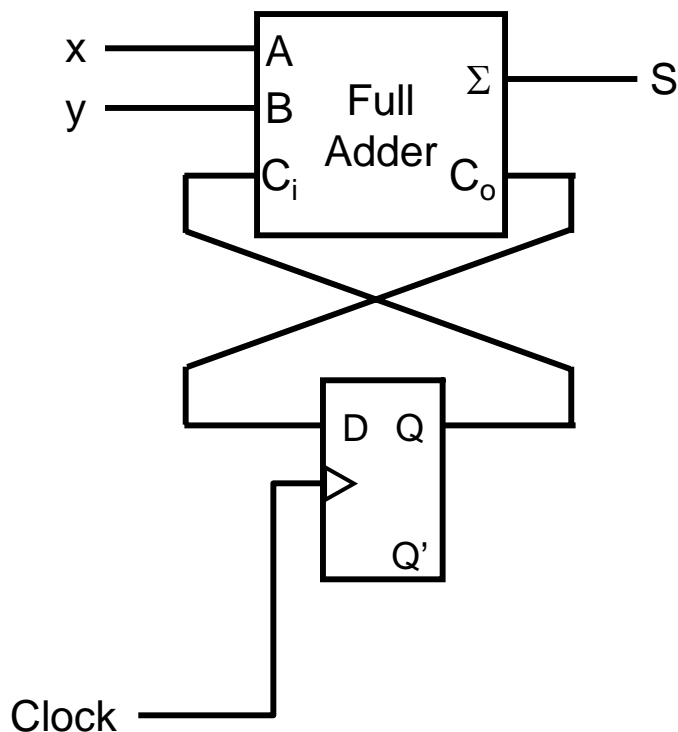
Present State	Input		Next State	Output
	X	Y		
$C_i(t)$	X	Y	$C_i(t+1)$	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_i	x	y	S	C_o
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	0	0	0	0

$S = 101110$

Problem 5-7

- Derive **state equations**, state table and state diagram
- Is this a **Mealy** or **Moore machine**?
- **What function does this sequential circuit perform?**



1. Assume initial state is 0
2. Assume x and y change in sync with clock
3. Consider the input sequences:
 $x = 010010$
 $y = 110100$

Notice:

$$\begin{array}{r}
 010010 \quad 18 \\
 +001011 \quad +11 \\
 \hline
 011101 \quad 29
 \end{array}$$

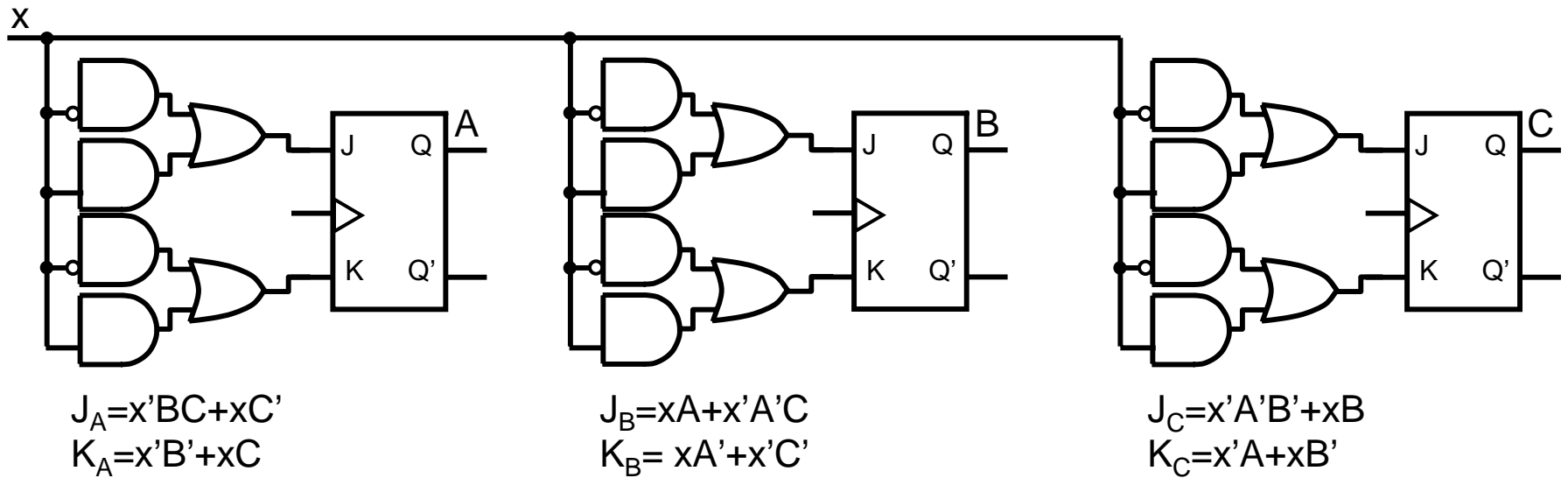
This is a Serial Adder
Data is entered LSB first

C_i	x	y	S	C_o
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	0	0	0	0

$$S = 101110$$

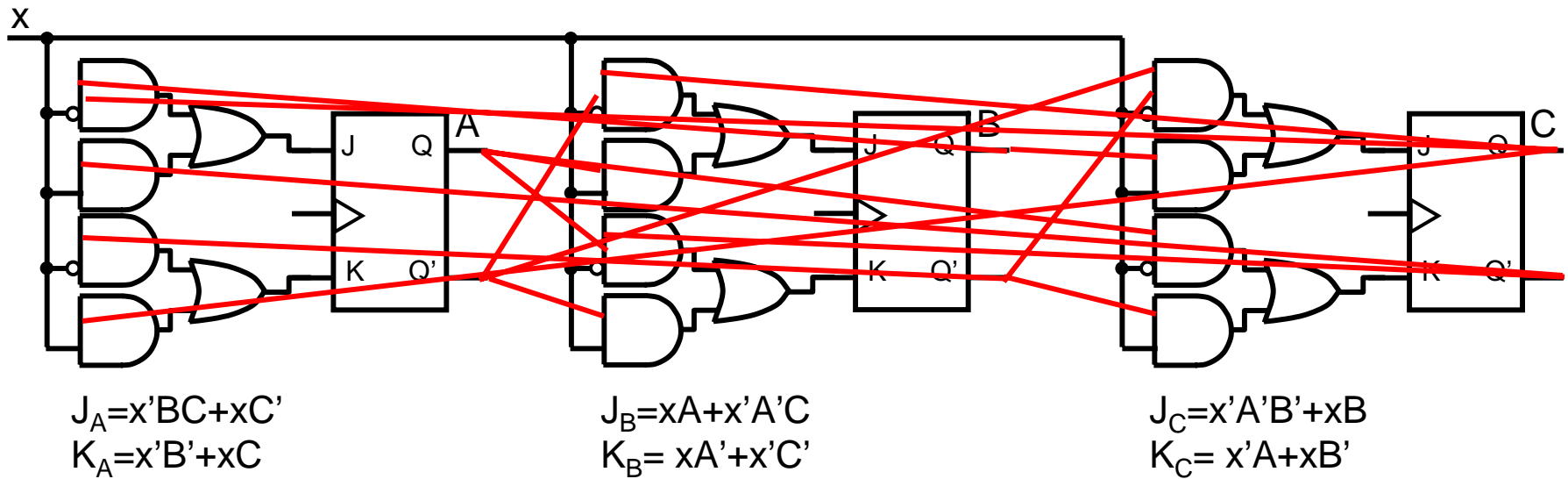
Another Analysis Example

- What function does this circuit perform?



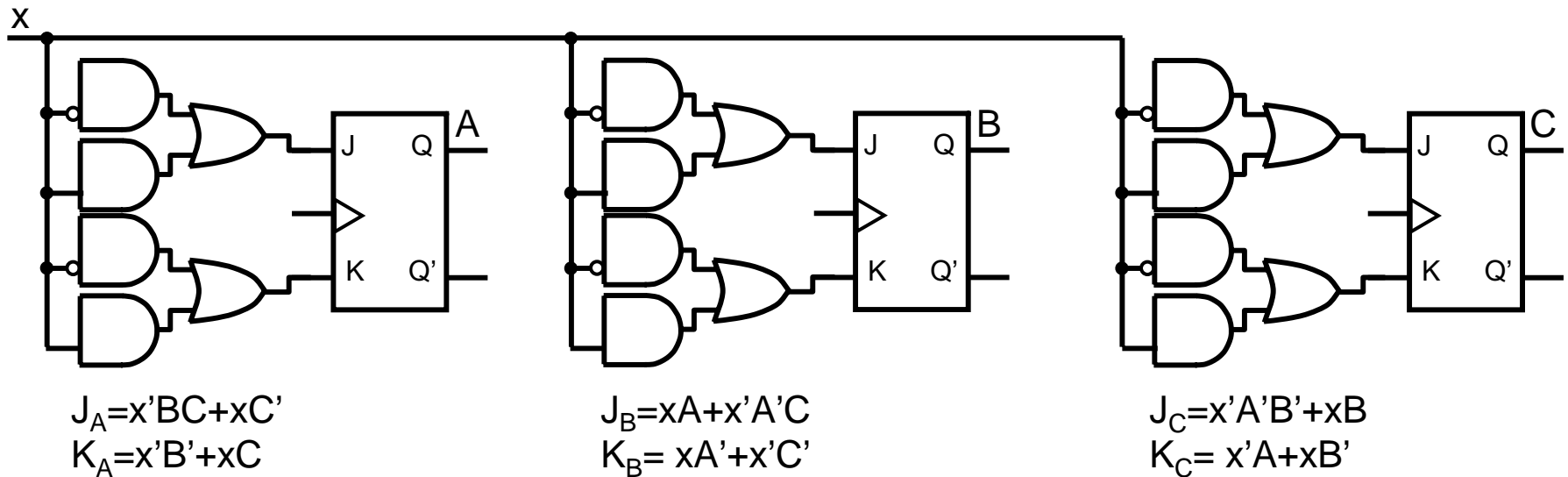
Another Analysis Example

- What function does this circuit perform?



Another Analysis Example

- What function does this circuit perform?



- Recognize these characteristics of a J-K flip-flop:

$$0 \xrightarrow{0/x} 0$$

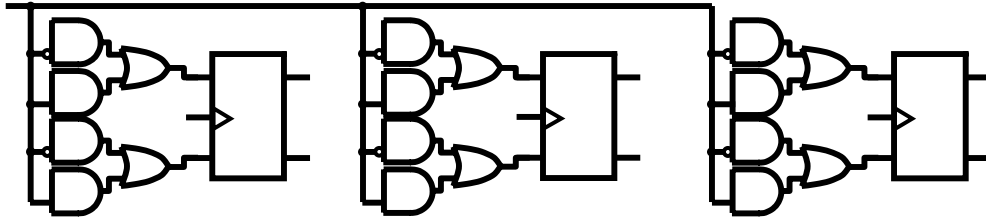
$$0 \xrightarrow{1/x} 1$$

$$1 \xrightarrow{x/1} 0$$

$$1 \xrightarrow{x/0} 1$$

Another Analysis Example

- What function does this circuit perform?

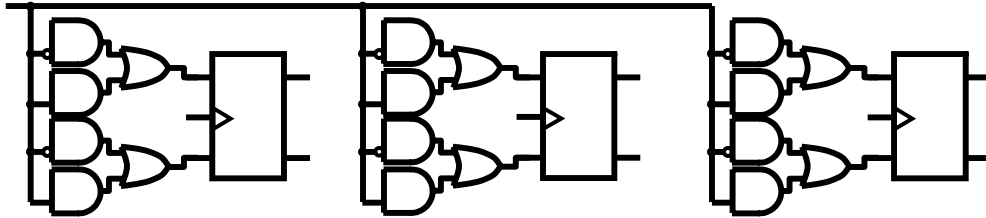


$$\begin{aligned}
 J_A &= x'BC + xC' & J_B &= xA + x'A'C & J_C &= x'A'B' + xB \\
 K_A &= x'B' + xC & K_B &= xA' + x'C' & K_C &= x'A + xB'
 \end{aligned}$$

State ABC	X	J _A	K _A	A _{t+1}	J _B	K _B	B _{t+1}	J _C	K _C	C _{t+1}	Next State
111	0	1	0	1	1	0	1	1	1	0	110
111	1	0	1	0	1	0	1	1	0	1	011

Another Analysis Example

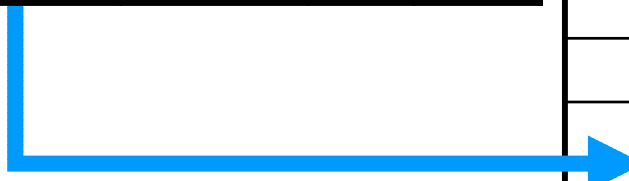
- What function does this circuit perform?



$$\begin{aligned}
 J_A &= x'BC + xC' & J_B &= xA + x'A'C & J_C &= x'A'B' + xB \\
 K_A &= x'B' + xC & K_B &= xA' + x'C' & K_C &= x'A + xB'
 \end{aligned}$$

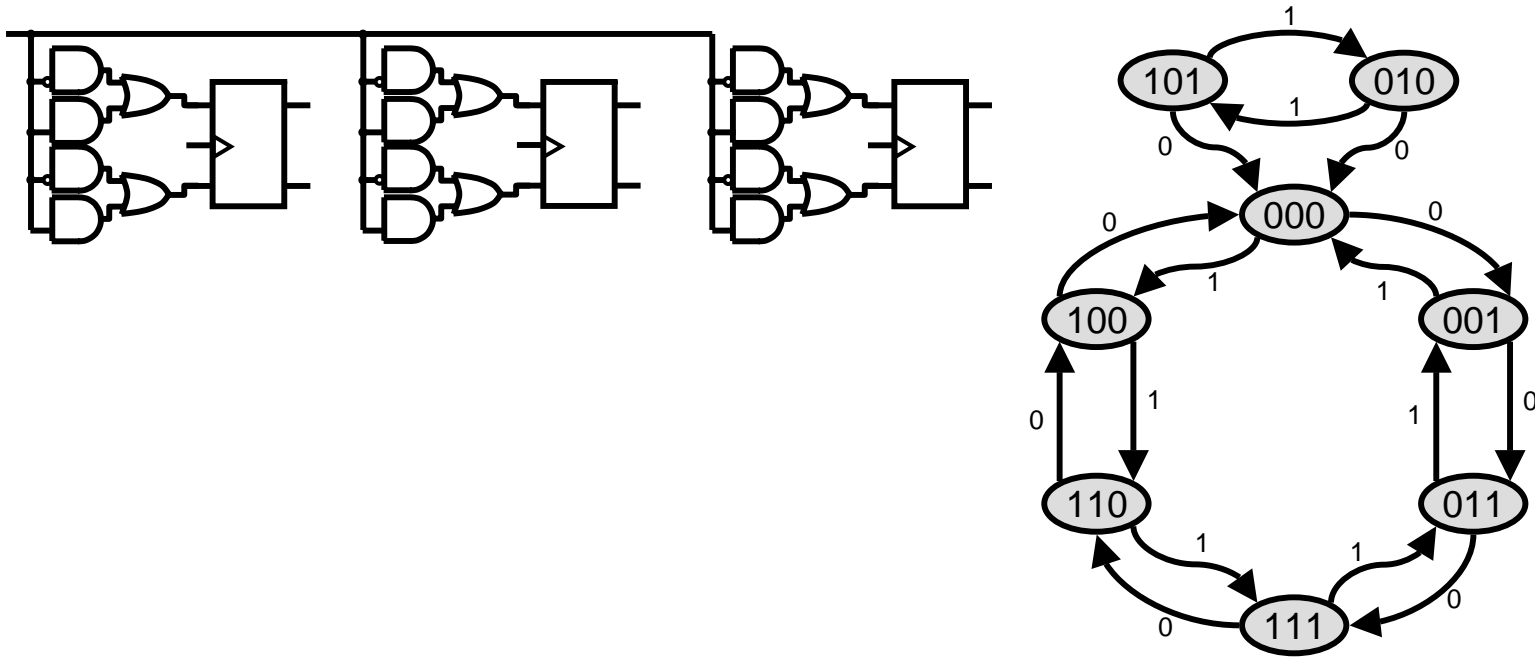
State ABC	X	J_A	K_A	A_{t+1}	J_B	K_B	B_{t+1}	J_C	K_C	C_{t+1}	Next State
111	0	1	0	1	1	0	1	1	1	0	110
111	1	0	1	0	1	0	1	1	0	1	011

Present State	Input	Next State/Output
ABC	X	ABC (t+1)
000	0	001
000	1	100
001	0	011
001	1	000
010	0	000
010	1	101
011	0	111
011	1	001
100	0	000
100	1	110
101	0	000
101	1	010
110	0	100
110	1	111
111	0	110
111	1	011

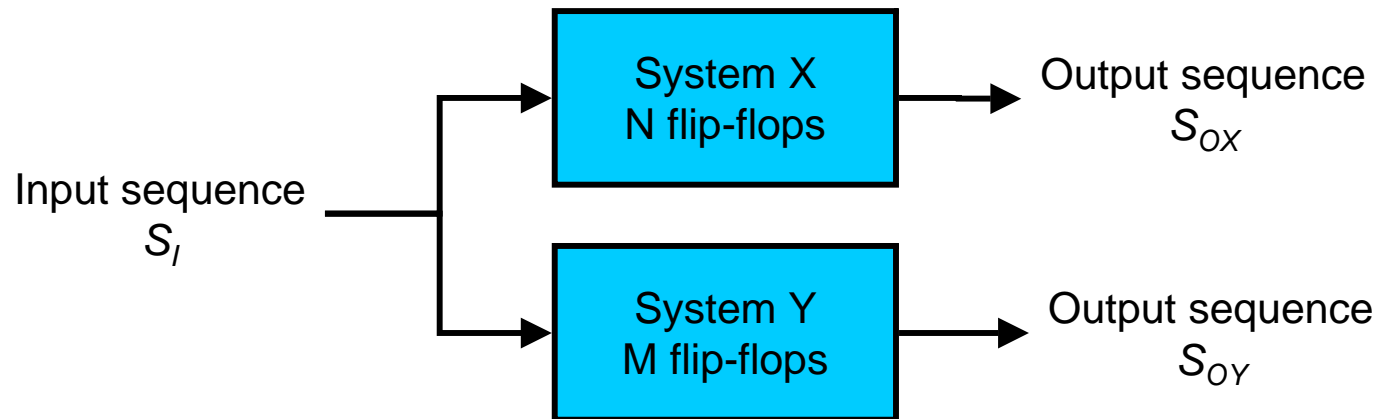


Another Analysis Example

- What function does this circuit perform?



State Reduction



$$S_{OX} = X(S_I)$$

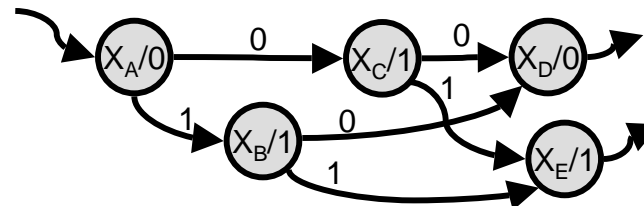
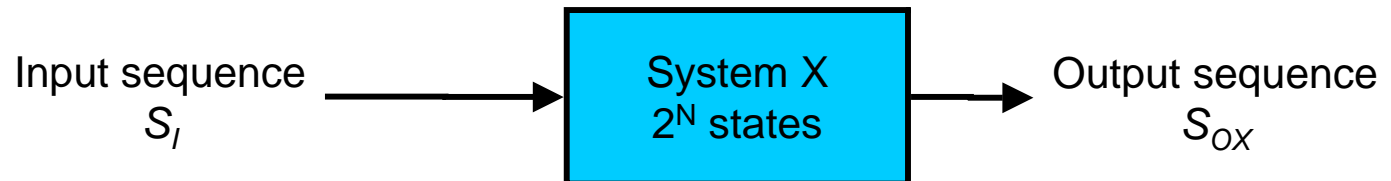
$$S_{OY} = Y(S_I)$$

$$(X \equiv Y) \Leftrightarrow (\forall S_I, X(S_I) = Y(S_I))$$

- If $M < N$, Y is a state-reduced version of X

State Reduction

- If there is a System Y with fewer states than X that produces the same output as X, some of the states of X must be equivalent to each other. Y can be generated by eliminating all the equivalent states.
- If there are two states (B&C) that, given the same starting state (A) and inputs, arrive in equivalent states with the same output, those states (B&C) are equivalent

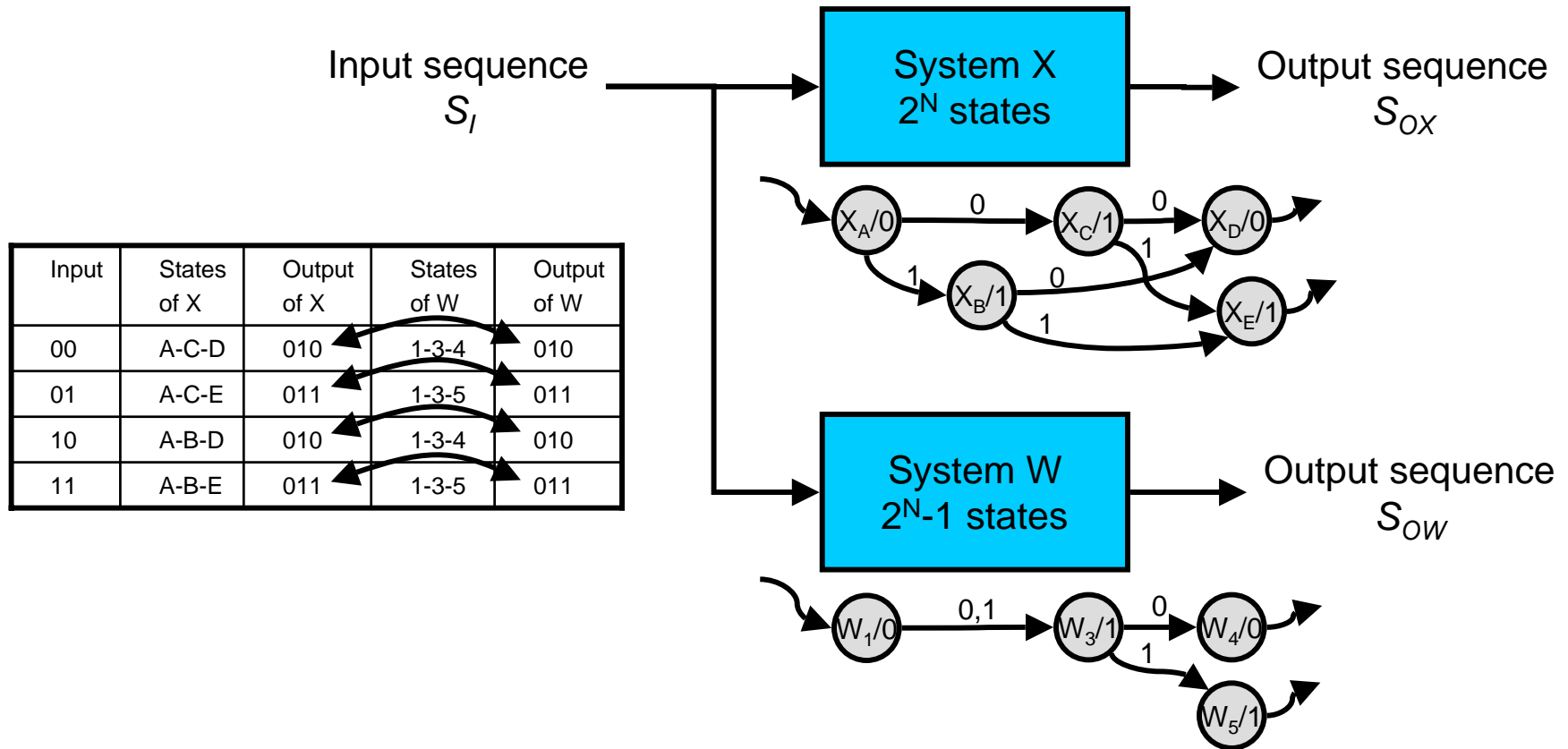


Input	States of X	Output of X	States of W	Output of W
00	A-C-D	010	1-3-4	010
01	A-C-E	011	1-3-5	011
10	A-B-D	010	1-3-4	010
11	A-B-E	011	1-3-5	011

States B and C
are equivalent

State Reduction

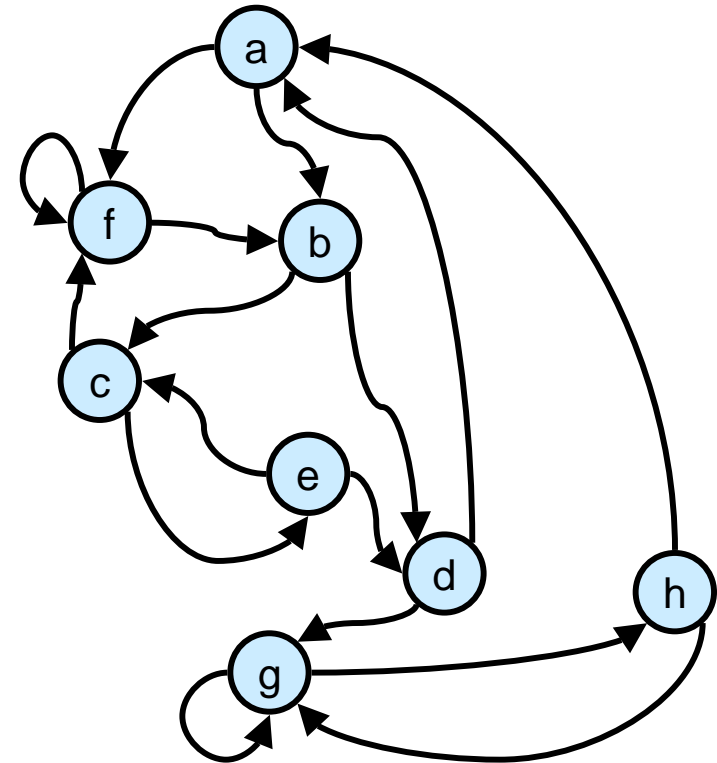
- If there is a System Y with fewer states than X that produces the same output as X, some of the states of X must be equivalent to each other. Y can be generated by eliminating all the equivalent states.
- If there are two states (B&C) that, given the same starting state (A) and inputs, arrive in equivalent states with the same output, those states (B&C) are equivalent



State Reduction – Problem 5-12

- Reduce the number of states in this state table and tabulate the reduced state table:

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0



State Reduction – Problem 5-12

- Reduce the number of states in this state table and tabulate the reduced state table:

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

- Start by assuming all states are equivalent. Create a single equivalence set:
{abcdefgh}

State Reduction – Problem 5-12

- Reduce the number of states in this state table and tabulate the reduced state table:

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

- Start by assuming all states are equivalent. Create a single equivalence set:

{abcdefgh}

- Without changing states, apply inputs x=0 and x=1 to distinguish states:

{abce} {dh} {f} {g}

State Reduction – Problem 5-12

- Equivalence sets:

{abce} {dh} {f} {g}

- Can {abce} be separated? Can {dh}?

- Consider the next-states that follow:

- Are they in separate sets?

$$\{dh\} \xrightarrow{0} \{g\}$$

$$\{dh\} \xrightarrow{1} \{a\}$$

- {dh} cannot be separated

$$\{abce\} \xrightarrow{0} \{f\}\{d\}$$

$$\{abce\} \xrightarrow{1} \{bce\}$$

- {ac} {be} can be separated, since with a 0 input, they go to distinct states (d and f)

- Equivalence sets:

{ac} {be} {dh} {f} {g}

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

State Reduction – Problem 5-12

- Equivalence sets:

{ac} {be} {dh} {f} {g}

- Again, consider the next-states that follow:

- Are they in separate sets

$\{ac\} \xrightarrow{0} \{f\}$

$\{ac\} \xrightarrow{1} \{be\}$

$\{be\} \xrightarrow{0} \{d\}$

$\{be\} \xrightarrow{1} \{c\}$

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

- These sets cannot be separated, so the equivalence sets are:

{ac} {be} {dh} {f} {g}

State Reduction – Problem 5-12

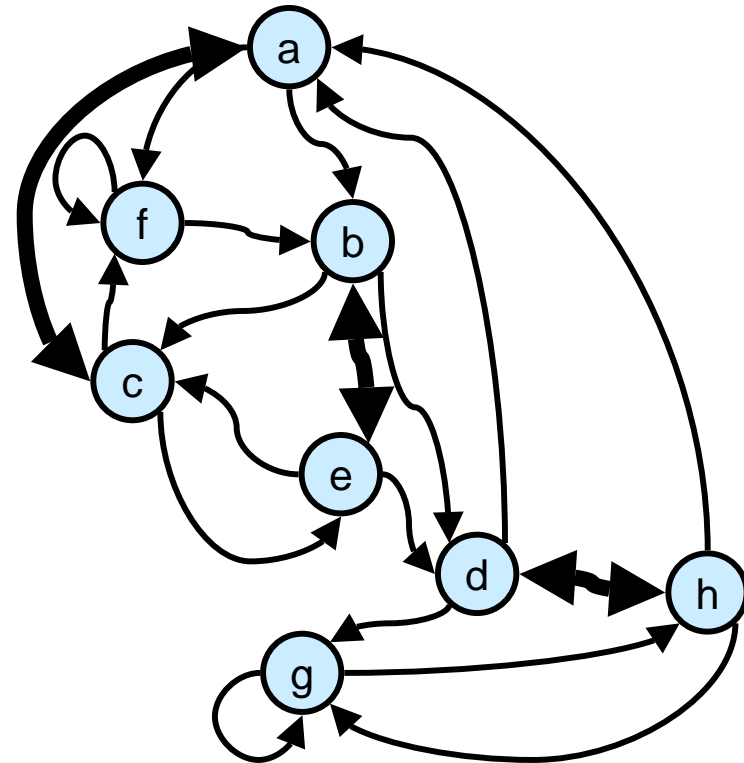
- Equivalence sets:

{ac} {be} {dh} {f} {g}

- The reduced state table is:

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
ac	f	be	0	0
be	dh	ac	0	0
C	f	e	0	0
dh	g	ac	1	0
E	d	e	0	0
F	f	b	1	1
G	g	dh	0	1
H	g	a	1	0

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

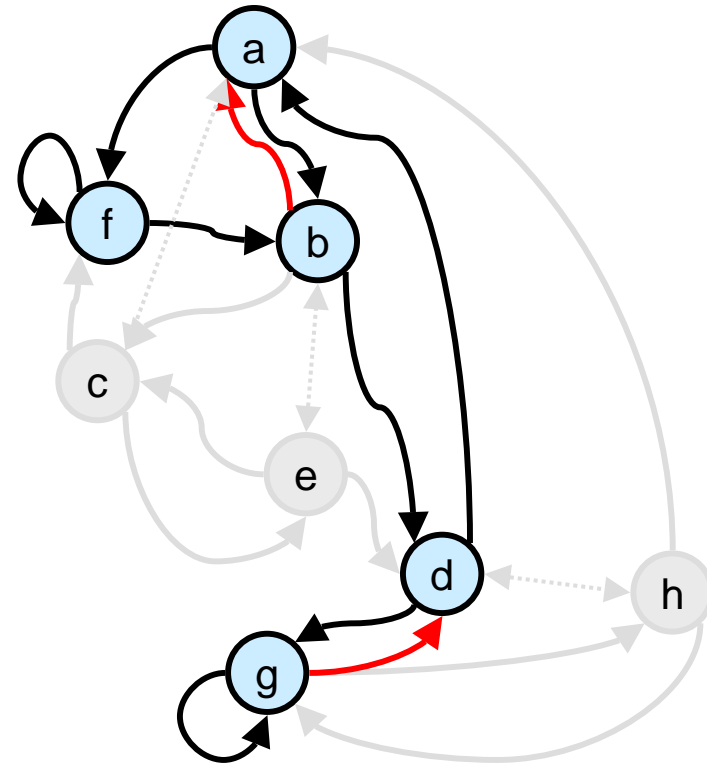


State Reduction – Problem 5-12

- The reduced state table is:

Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

- 8 states required at least 3 FFs
- 5 states requires at least 3 FFs



Implications of State Reduction

- System B has N states
- System C has $N-K$ states ($K > 0$)
 - Which is preferable?

Implications of State Reduction

- System B has N states
- System C has $N-K$ states ($K>0$)
 - Which is preferable?

B requires at least $\lceil \log_2(N) \rceil$ flip-flops

C requires at least $\lceil \log_2(N - K) \rceil$ flip-flops

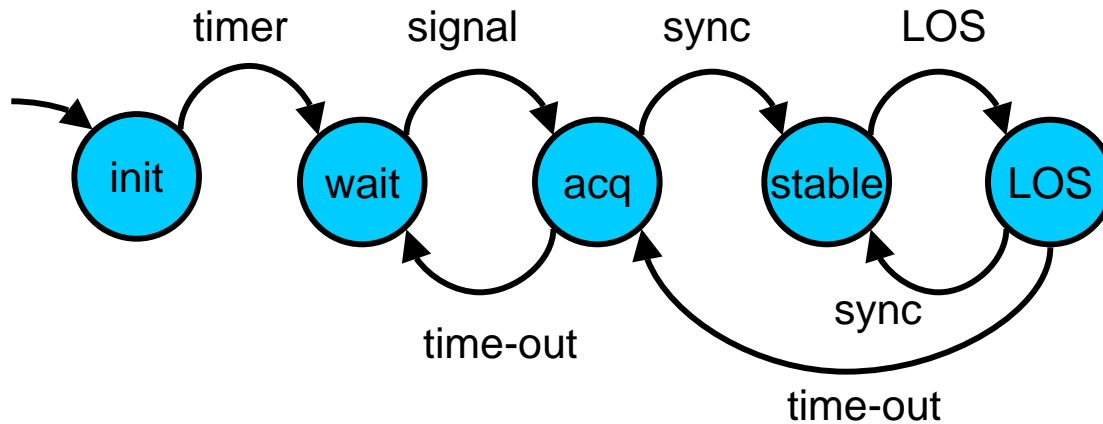
Is $\lceil \log_2(N - K) \rceil < \lceil \log_2(N) \rceil$?

Not unless $(N - K) \leq 2^M < N$

- Which requires more combinatorial logic?
 - Possibly C.

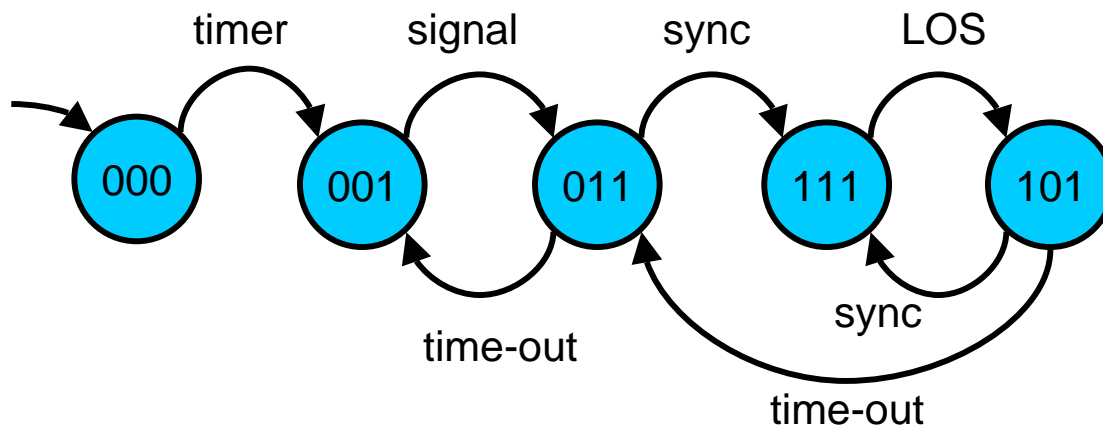
State Assignment

- Start with either symbolic names or binary numbers for states, depending on task



For N states,
number of bits
(FF's) needed is:

$$B \geq \lceil \log_2 N \rceil$$



Look for state assignments that have meaning. E.g.,

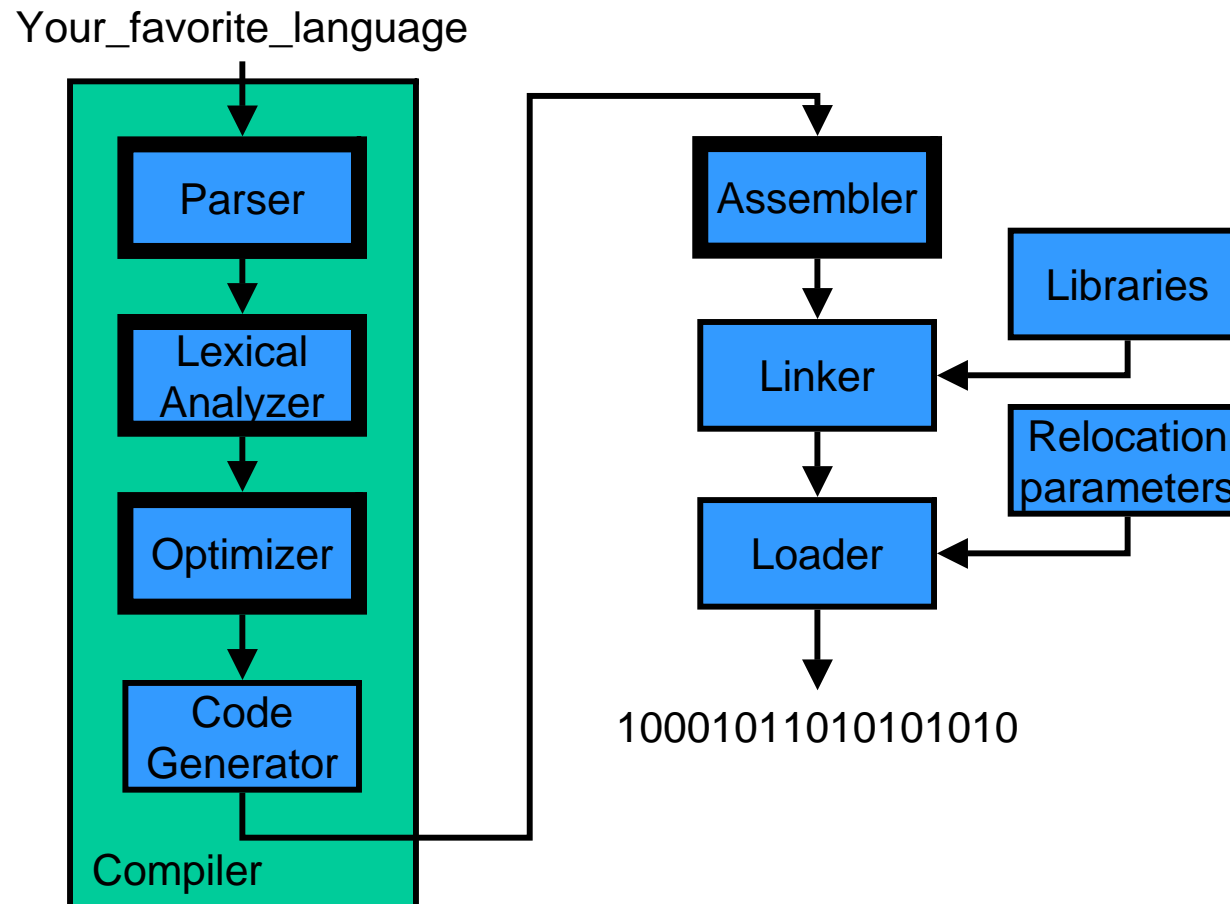
- 000 initial state
- xx1 not init
- x1x signal present
- 1xx signal present

An Aside – A Brief Discussion of “Formal Languages”

- How does a computer understand what the following means?

```
float inner_product(coeff, values)
{
    float sum = 0;
    for(index=0; index<N; index++)
    {
        sum += *coeff++ * *values++;
    }
    return(sum);
}
```

Translating Code into 1's and 0's



An Excerpt from the Algol-60 Formal Language Specification

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<letter> ::= a|b|c|d|e|f|g|h| ... |x|y|z|A|B|C|D ... |X|Y|Z

<delimiter> ::= <operator> | <separator> | <bracket> | <declarator> |
<specifier>

<identifier> ::= <letter> | <identifier> <letter> | <identifier> <digit>

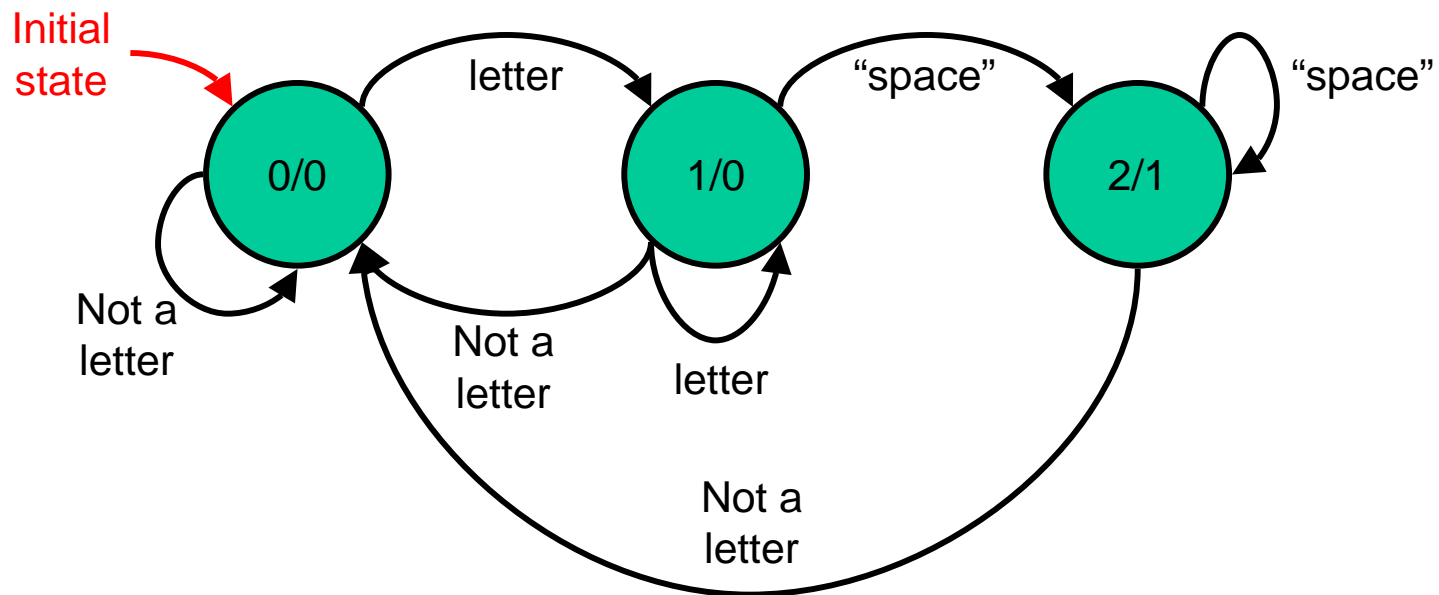
<procedure_identifier> ::= <identifier>

<actual_parameter> ::= <string> | <expression> | <array_identifier> |
<switch_identifier> | <procedure_identifier>

.
. .
.

Finite State Machines and Sequential Logic

- Recognize a string of letters that ends with one or more “space” characters. Preceding non-letter symbols are ignored



- The Finite State Machine that recognizes Formal Languages is described with the same state diagram we would use to design a sequential circuit

Summary

- Fundamental concepts of digital systems (Mano Chapter 1)
- Binary codes, number systems, and arithmetic (Ch 1)
- Boolean algebra (Ch 2)
- Simplification of switching equations (Ch 3)
- Digital device characteristics (e.g., TTL, CMOS)/design considerations (Ch 10)
- Combinatoric logical design including LSI implementation (Chapter 4)
- Flip-flops and state memory elements (Ch 5)
- **Sequential logic analysis and design (Ch 5)**
- Hazards, Races, and time related issues in digital design (Ch 9)
- Synchronous vs. asynchronous design (Ch 9)
- Counters, shift register circuits (Ch 6)
- Memory and Programmable logic (Ch 7)
- Minimization of sequential systems
- Introduction to Finite Automata

Homework 7 – due in Class 9

- As always, show all work
- Problems 5-2, 5-11