

SEQUENTIAL NETWORKS (CH 8)

STUDY

AT LOGIC LEVEL:

SN IS A COLLECTION OF

SIMPLER COMB AND SEQ NETS

AT CIRCUIT LEVEL:

PHYSICAL IMPL. OF ELEM SER MODULES

(BINARY CELL;

GATED LATCH & FLIP-FLOP

COLLECTION OF FFs \rightarrow REGISTER

CANONICAL FORM (HUFFMAN-MOORE)

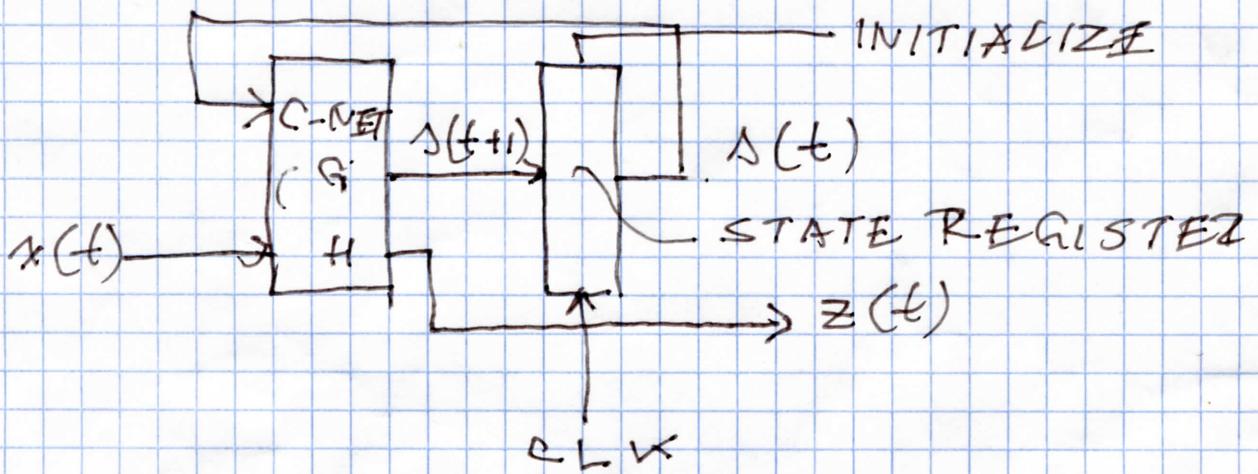
DIRECT IMPL. OF STATE DESCRIPTION

$$s(t+1) = f(s(t), x(t))$$

$$z(t) = h(s(t), x(t))$$

CONSISTS OF:

- A STATE REGISTER (STORES $s(t)$)
- A COMB. NET: IMPLEMENTS f AND h
- THE CLOCK CLK



SEQUENTIAL NETWORKS

- CANONICAL FORM OF SEQUENTIAL NETWORKS
- LATCHES AND EDGE-TRIGGERED CELLS. D FLIP-FLOP
- TIMING CHARACTERISTICS
- ANALYSIS AND DESIGN OF CANONICAL NETWORKS
- SR, JK and T FLIP-FLOP
- ANALYSIS OF FLIP-FLOP NETWORKS
- DESIGN OF FLIP-FLOP NETWORKS. EXCITATION FUNCTIONS
- SPECIAL STATE ASSIGNMENTS: ONE-FLIP-FLOP-PER-STATE AND SHIFT-ING REGISTER

CANONICAL FORM OF SEQUENTIAL NETWORKS (Huffman-Moore)

STATE-TRANSITION FUNCTION $s(t+1) = G(s(t), x(t))$
 OUTPUT FUNCTION $z(t) = H(s(t), x(t))$

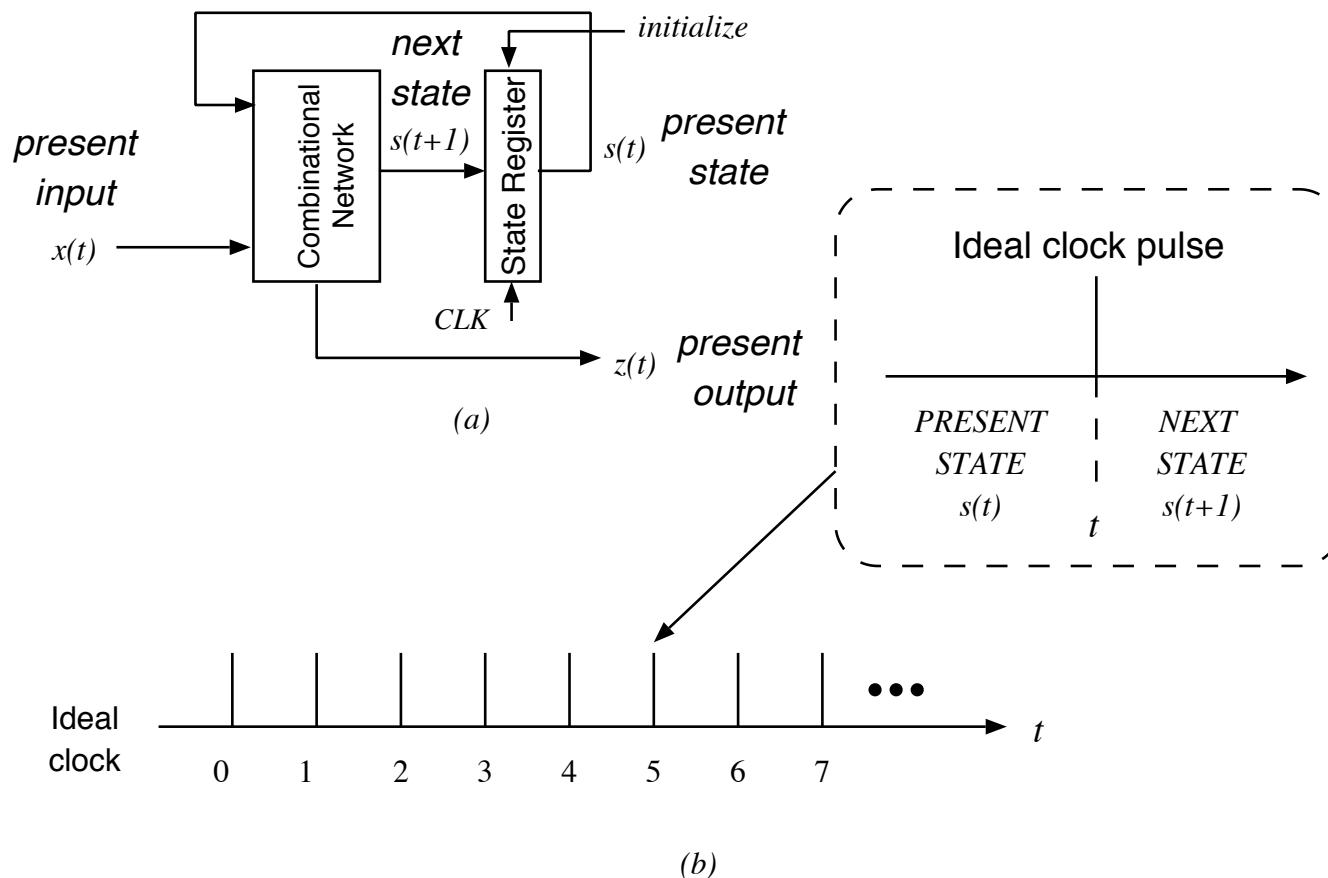
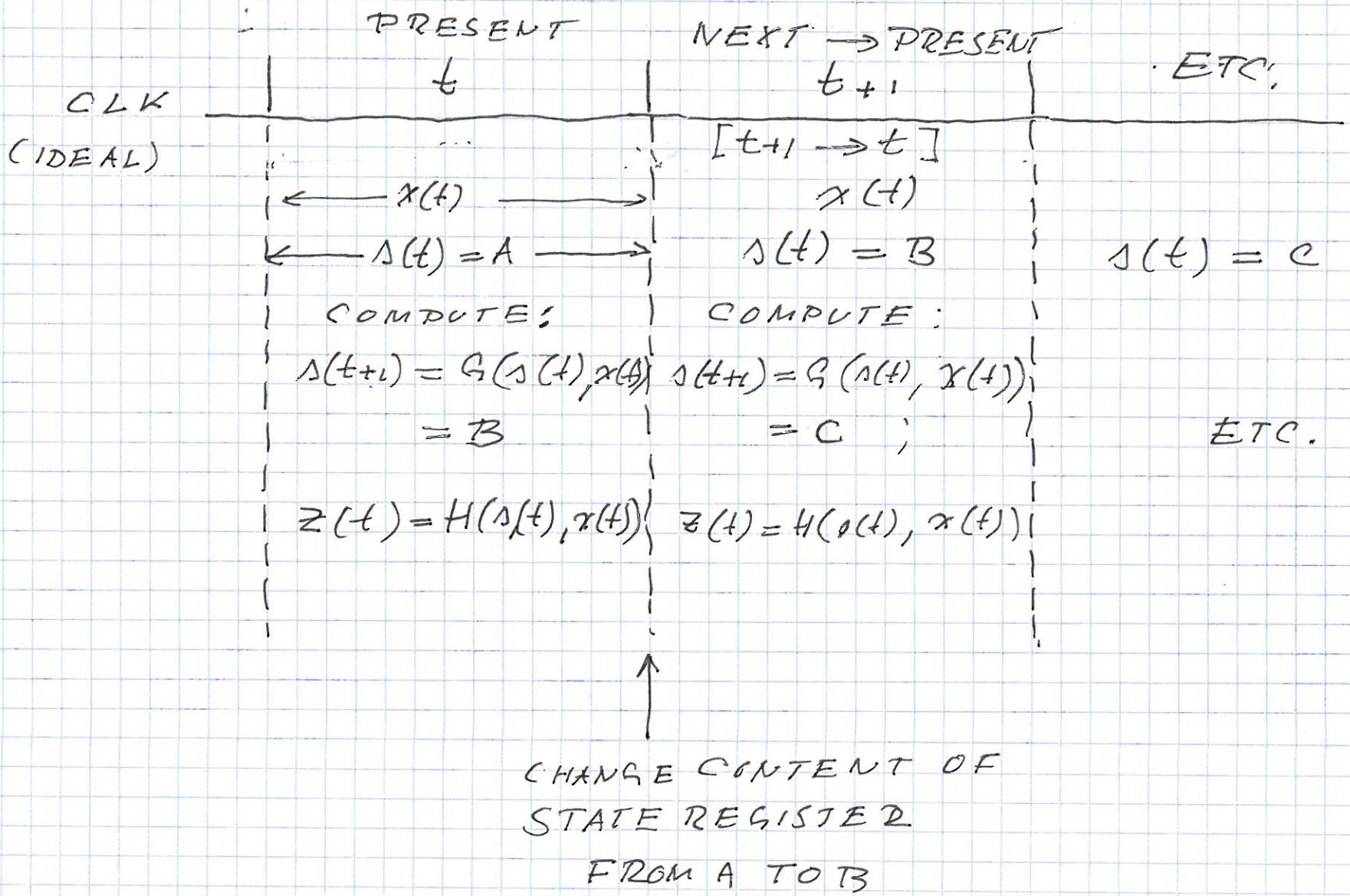


Figure 8.1: a) CANONICAL IMPLEMENTATION OF SEQUENTIAL NETWORK. b) IDEAL CLOCK SIGNAL AND ITS INTERPRETATION.

CS M5A

TIMING OF STATE EQUATIONS



MEALY AND MOORE MACHINES

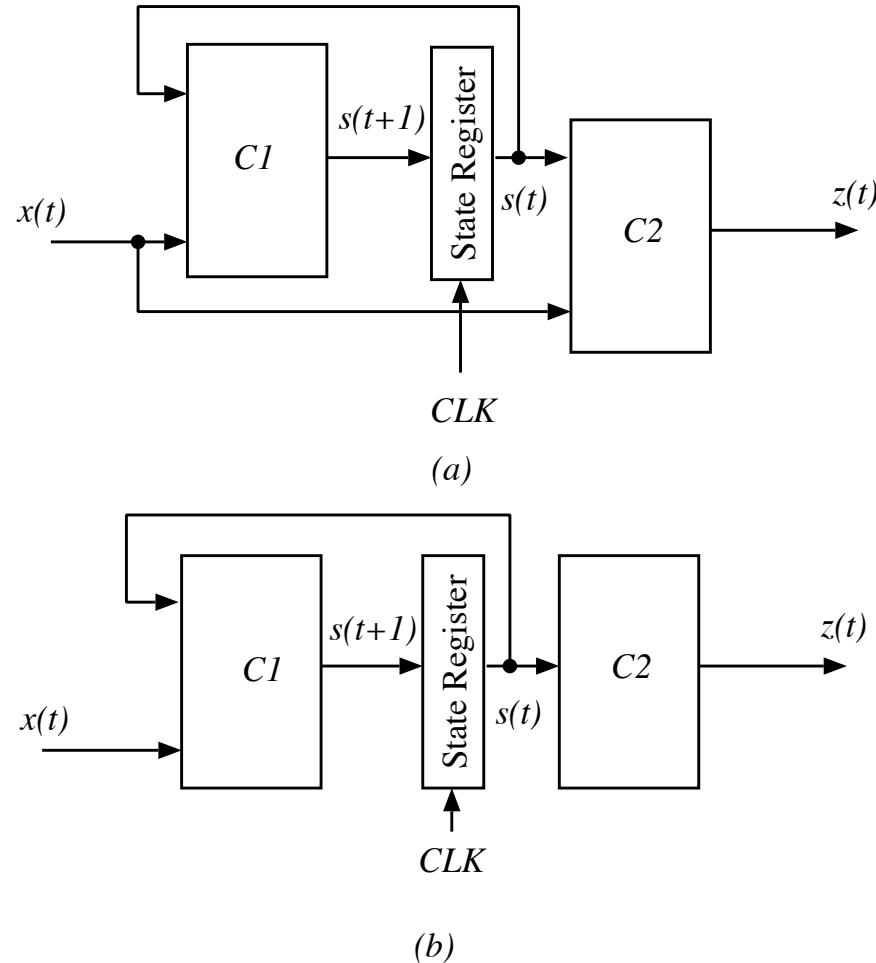
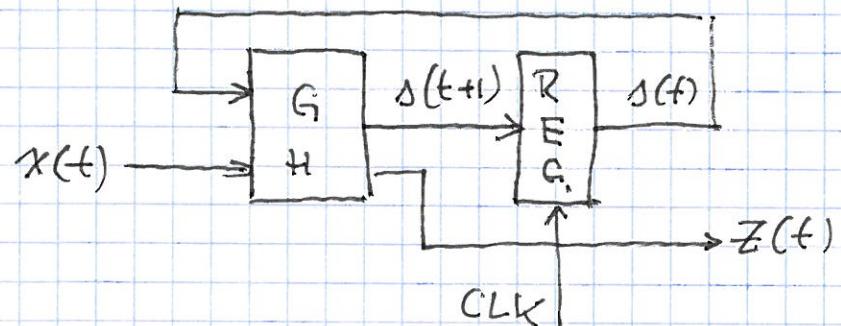


Figure 8.2: CANONICAL IMPLEMENTATIONS: a) MEALY MACHINE. b) MOORE MACHINE.

CS 451A

EXAMPLE OF INPUT, STATE & OUTPUT
BEHAVIOR

$$\begin{aligned} s(t+1) &= G(s(t), x(t)) \\ z(t) &= H(s(t)) \end{aligned} \quad \left. \begin{array}{l} \text{MOORE} \\ \text{MACHINE} \end{array} \right\}$$

PS	$x(t)$	$z(t)$
S_0	a b c	P
S_1	$S_0 S_1 S_1$	Σ
S_2	$S_2 S_3 S_0$	Σ
S_3	$S_0 S_1 S_2$	P

NS

 S_2 - INITIAL

t	$x(t)$	$s(t)$	$s(t+1)$	$z(t)$
0	a	$\boxed{S_2}$	S_2	g
1	b	S_2	S_3	g
2	c	S_3	S_2	p
3	a	S_2	S_2	g
4				

"LOAD" STATE REGISTER
ON CLOCK

HIGH-LEVEL AND BINARY IMPLEMENTATIONS

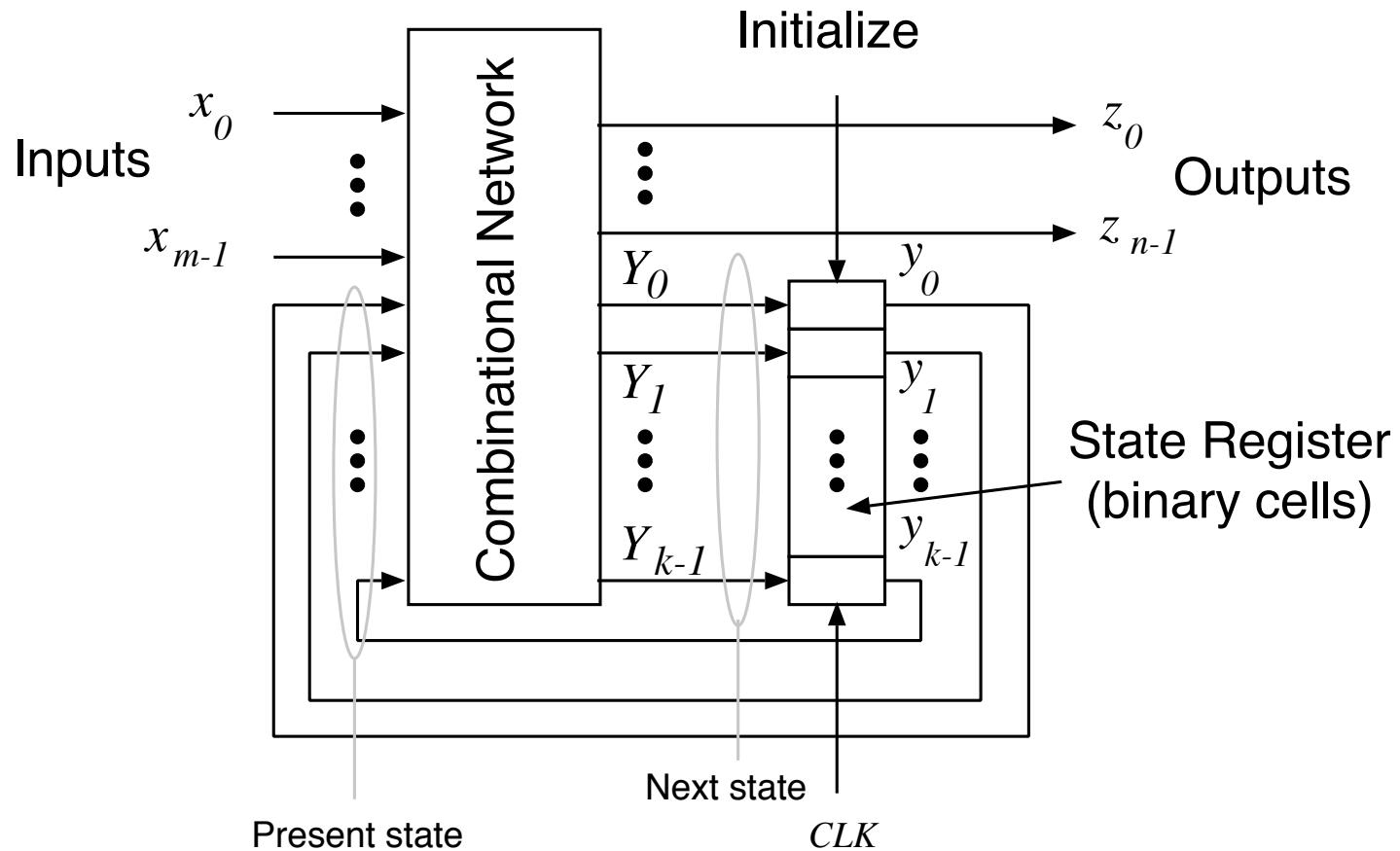


Figure 8.3: CANONICAL IMPLEMENTATION WITH BINARY VARIABLES.

EXAMPLE 8.1

INPUT: $\underline{x}(t) = (x_1, x_0), \quad x_i \in \{0, 1\}$

OUTPUT: $z(t) \in \{0, 1\}$

STATE: $\underline{y}(t) = (y_3, y_2, y_1, y_0), \quad y_i(t) \in \{0, 1\}$

INITIAL STATE: $\underline{y}(0) = (0, 0, 0, 0)$

FUNCTION: THE TRANSITION AND OUTPUT FUNCTIONS

$$Y_3 = y_2 x'_1 x_0$$

$$Y_2 = (y_1 + y_2)x'_0 + y_3 x_1$$

$$Y_1 = (y_0 + y_3)x'_1 x_0 + (y_0 + y_1)x_1$$

$$Y_0 = (y_0 + y_3)x'_0 y_1 x'_1 x_0 + y_2 x_1$$

$$z = y_3 + y_2 + y_1 + y_0$$

EXAMPLE 8.1 (cont.)

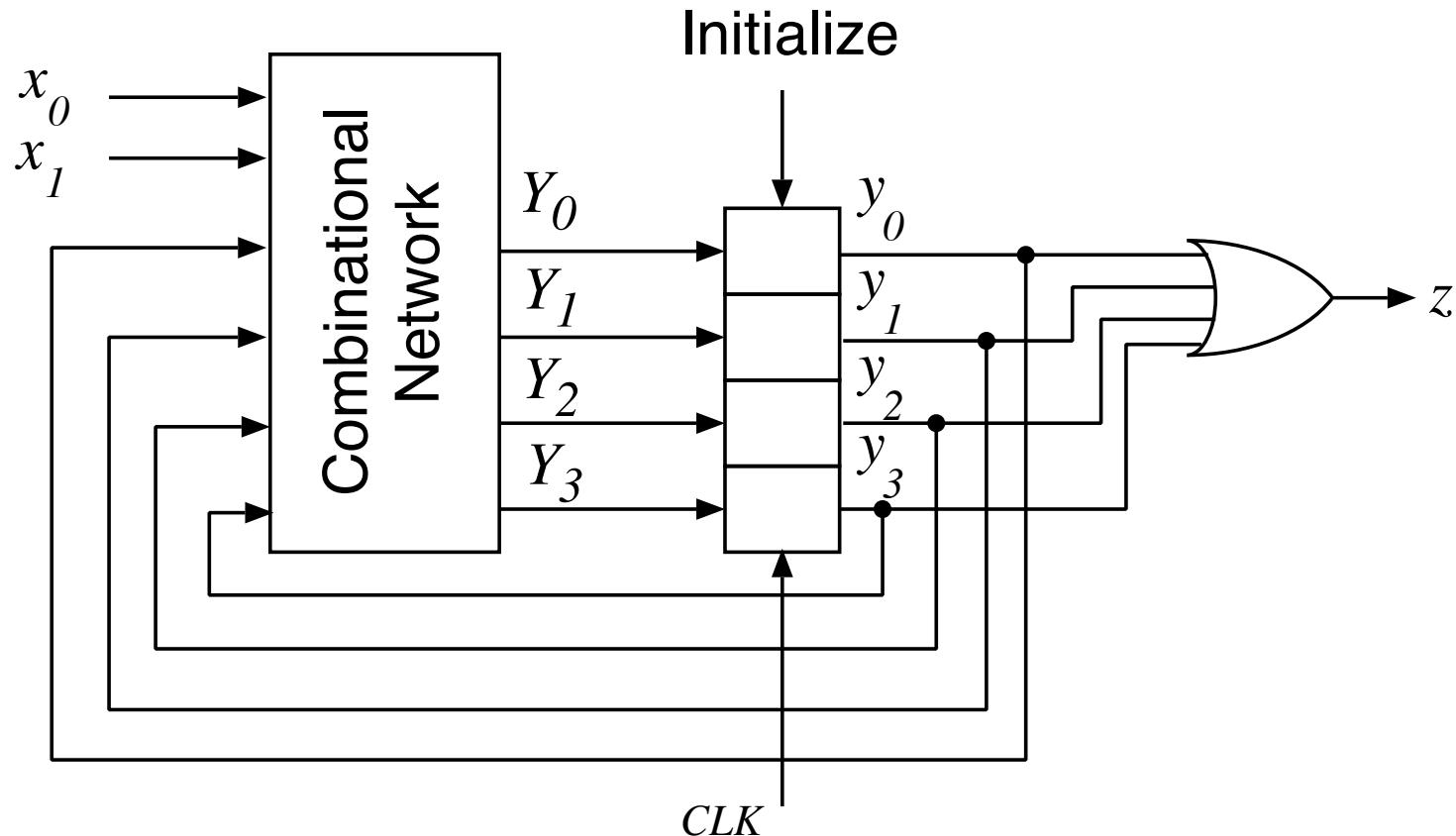


Figure 8.4: CANONICAL NETWORK FOR EXAMPLE 8.1.

CLOCK

- CLOCK PERIOD T
- CLOCK FREQUENCY $f = 1/T$
- (CLOCK) PULSE WIDTH t_w

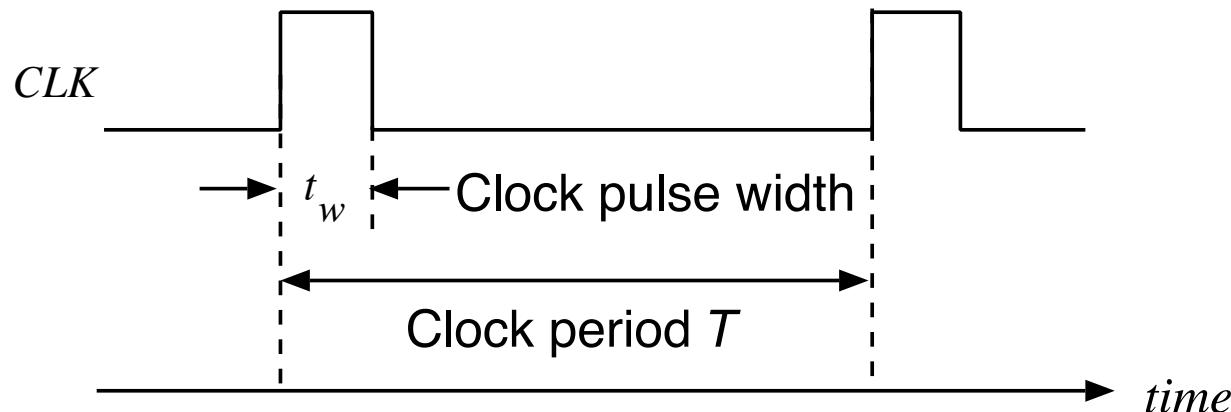


Figure 8.5: PULSE WIDTH AND CLOCK PERIOD.

$f = 1 \text{ Hz}$ (Hertz): ONE CHANGE / SEC

 f

$$(\text{Kilo}) 1 \text{ KHz} = 10^3 \text{ Hz}$$

$$T = 1/f$$

$$10^{-3} \text{ sec} = 1 \text{ ms} \text{ (MILLISECOND)}$$

$$(\text{mega}) 1 \text{ MHz} = 10^6 \text{ Hz}$$

$$10^{-6} \text{ sec} = 1 \mu\text{s} \text{ (MICROSECOND)}$$

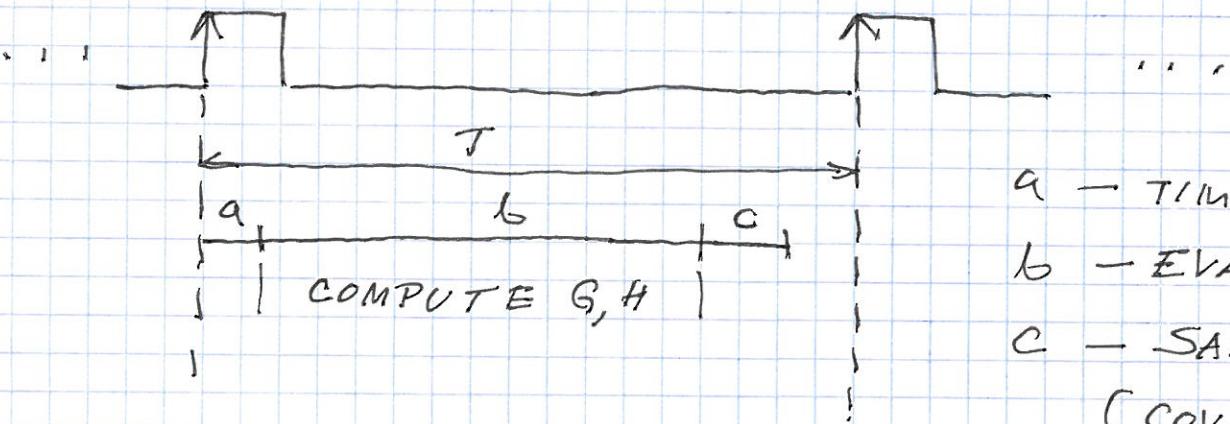
$$(\text{Giga}) 1 \text{ GHz} = 10^9 \text{ Hz} = 10^3 \text{ MHz} \quad 10^{-9} \text{ sec} = 1 \text{ ns} \text{ (NANOSECOND)}$$

$$(\text{Tera}) 1 \text{ THz} = 10^{12} \text{ Hz} = 10^3 \text{ GHz} \quad 10^{-12} \text{ sec} = 1 \text{ ps} \text{ (PICOSECOND)}$$

$$2 \text{ GHz CLOCK; } T = 0.5 \text{ ns} = 500 \text{ ps}$$

$$4 \text{ GHz} \quad \equiv \quad T = 0.25 \text{ ns} = 250 \text{ ps}$$

$$10 \text{ GHz} \quad // \quad T = 0.1 \text{ ns} = 100 \text{ ps}$$



a - TIME TO STABILIZE STATE \rightarrow

b - EVALUATE STATE EQUATIONS

c - SAFETY MARGIN

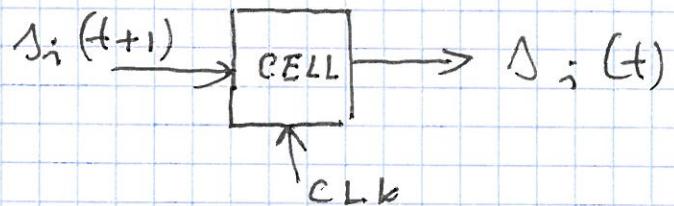
(COVERS VARIATIONS IN DELAYS)

CS MSIA

BASIC BINARY CELL



FUNCTION: STORE A STATE VARIABLE s_i .



- CHANGE ITS VALUE ON CLOCK
- s_i DOES NOT CHANGE BETWEEN SUCCESSIVE CLOCK PULSES

BEFORE CLOCK: AFTER CLOCK

$s(t)$

0

0

1

1

$s(t)$

0

1

0

1

GATED LATCH - FIRST TRY

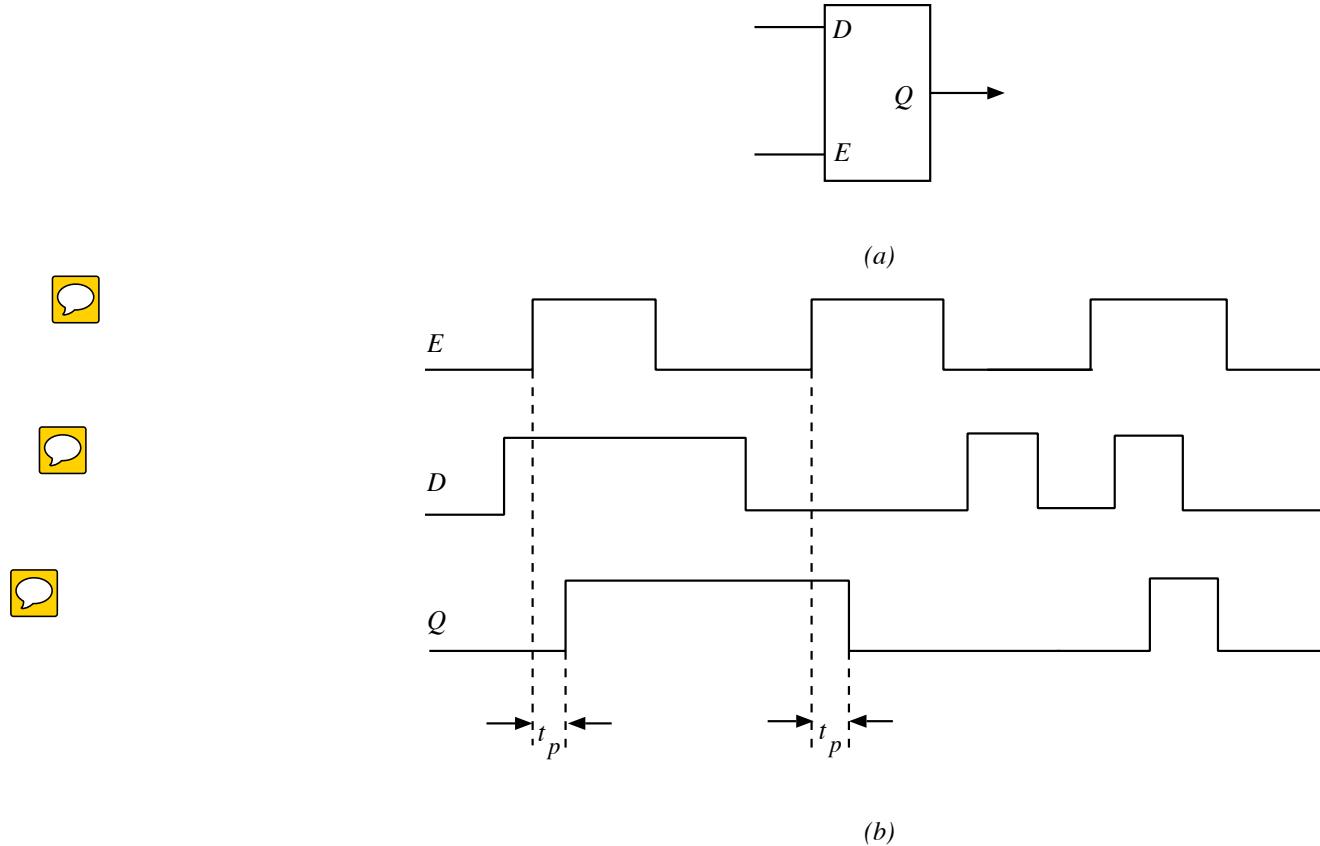
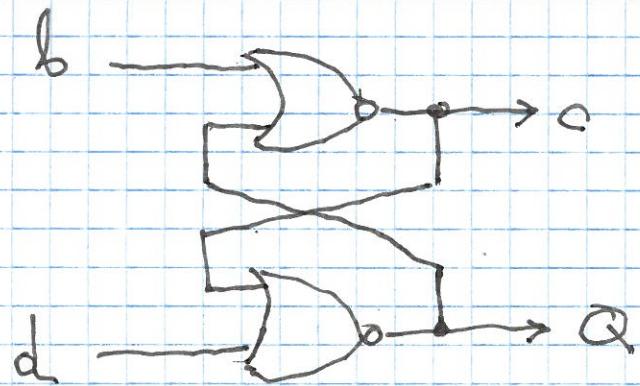


Figure 8.6: a) GATED-LATCH. b) TIMING BEHAVIOR.

$$Q(t + t_p) = D(t) \cdot E(t) + Q(t) \cdot E'(t)$$

- LEVEL-SENSITIVE: when $E = 1$ then $Q = D$

- IDEA:



$b(t)$	0	0	1	1
$d(t)$	0	1	0	1
<hr/>	<hr/>			
$c(t+\Delta)$	$Q'(t)$	1	0	0
$Q(t+\Delta)$	$C'(t)$	0	1	0
				X
				not allowed
				$\Rightarrow b \cdot d \neq 1$

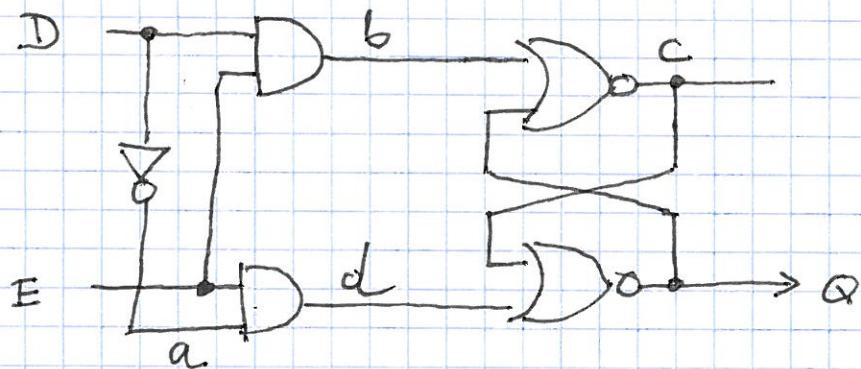
HOLD PREVIOUS STATE

SET Q TO 0

SET Q TO 1

$$Q(t) = C'(t) \quad - \text{COMPLEMENTED OUTPUTS}$$

- SPECIFY b & d AS FUNCTIONS OF E (ENABLE) AND D (DATA)
- $b = E \cdot D$ $d = E \cdot D'$



$$\begin{aligned} b &= D \cdot E & a &= D' \\ d &= D' \cdot E \end{aligned}$$

$E = 1$ $b = D$ $d = D'$ $a = D'$

$$D = 1 \Rightarrow \begin{cases} b = 1 \Rightarrow C = 0 \Rightarrow \\ d = 0 \end{cases} \xrightarrow{\text{LOOP}} \boxed{Q = 1} \quad \therefore Q = D$$

$$D = 0 \Rightarrow \begin{cases} b = 0 \xrightarrow{\text{LOOP}} \\ d = 1 \Rightarrow \boxed{Q = 0} \Rightarrow C = 1 \end{cases} \quad \uparrow \quad \downarrow \quad \therefore Q = D$$

SO, WHEN $E = 1$, $Q = D$ (OUTPUT IS INPUT-LEVEL SENSITIVE)

CHANGE E TO 0, KEEP D UNCHANGED UNTIL $E = 0$

$E = 0$ $b = 0$ $d = 0$

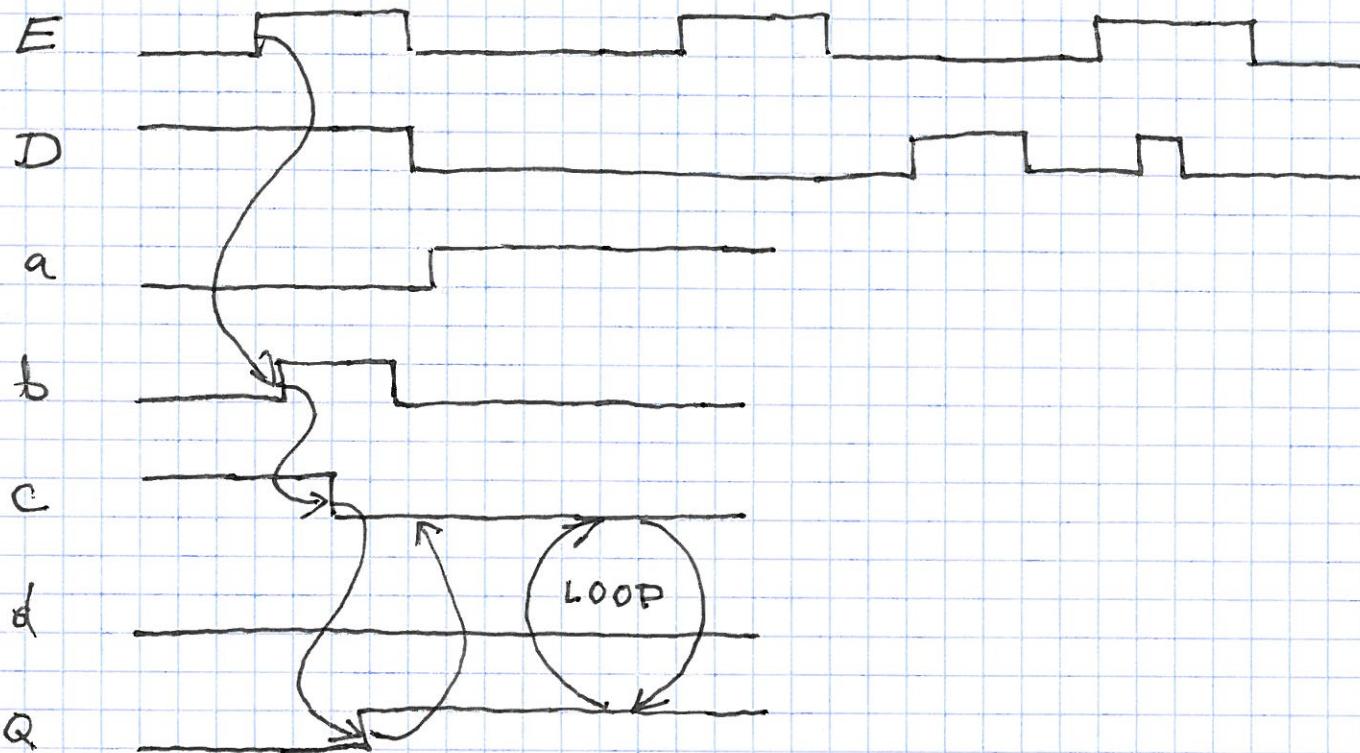
IF $C = 0$ THEN $\boxed{Q = 1} \Rightarrow C = 0$ STABLE

IF $C = 1$ THEN $\boxed{Q = 0} \Rightarrow C = 1$ STABLE

REMAINS STABLE AS LONG AS $E = 0$

CS MSIA

NOR-NOR LATCH TIMING DIAGRAM



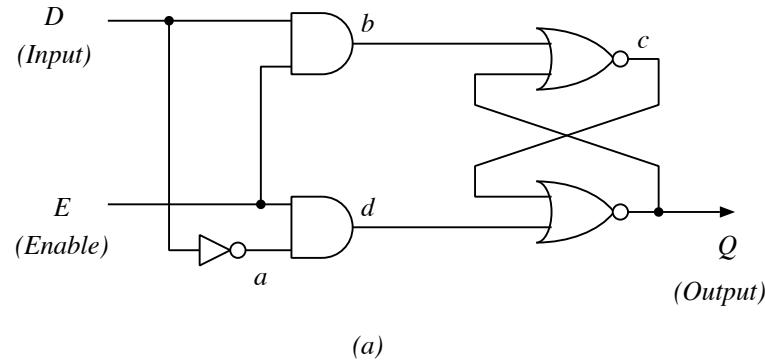
(ASSUME
 $Q=0$)

HOLD

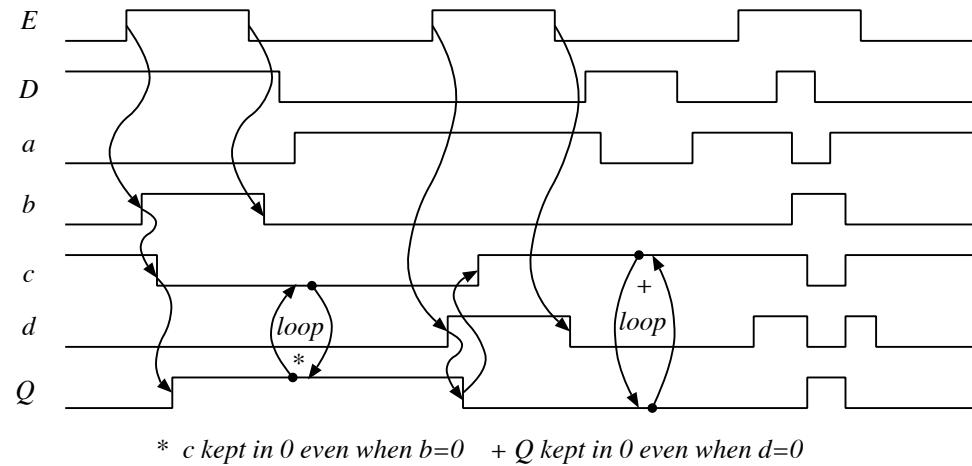
STATE

- FINISH THE TIMING DIAGRAM

NOR-NOR LATCH



(a)



(b)

Figure 8.7: a) IMPLEMENTATION OF GATED-LATCH WITH NOR GATES. b) TIMING DIAGRAM.

LIMITATIONS OF GATED-LATCH

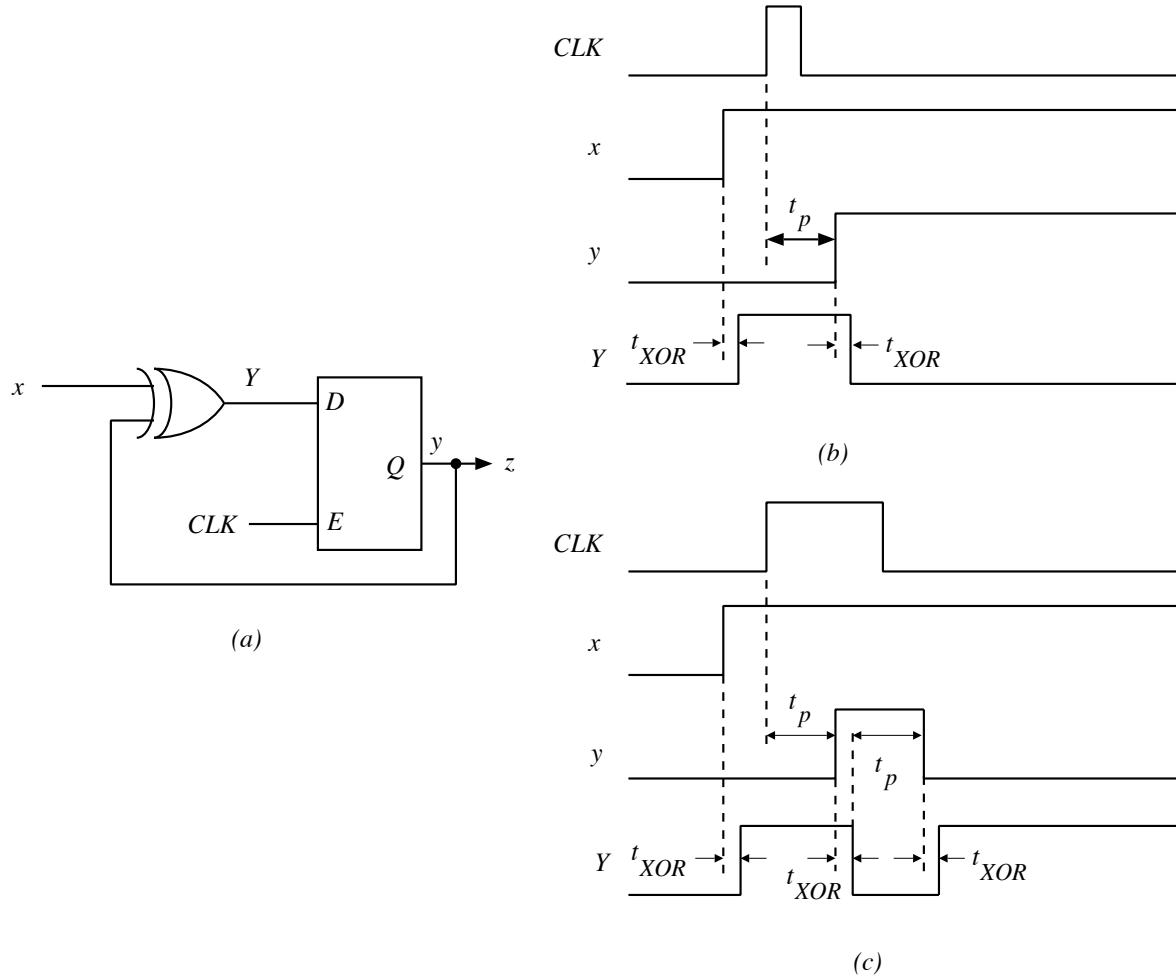
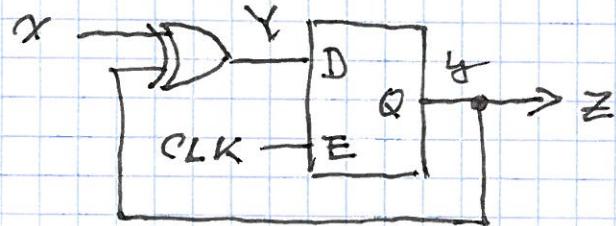


Figure 8.8: a) SEQUENTIAL NETWORK. b) CORRECT TIMING BEHAVIOR. c) INCORRECT TIMING BEHAVIOR.

CS M51A

LATCH BEHAVIOR DEPENDS ON
CLOCK PULSE (ENABLE) WIDTH
AND PROP. DELAY OF G NETWORK

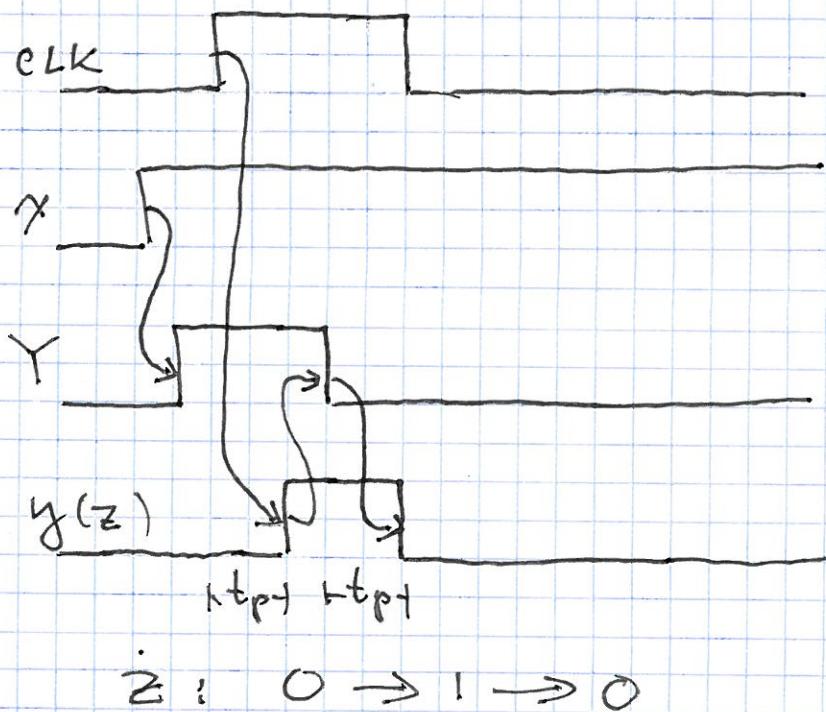
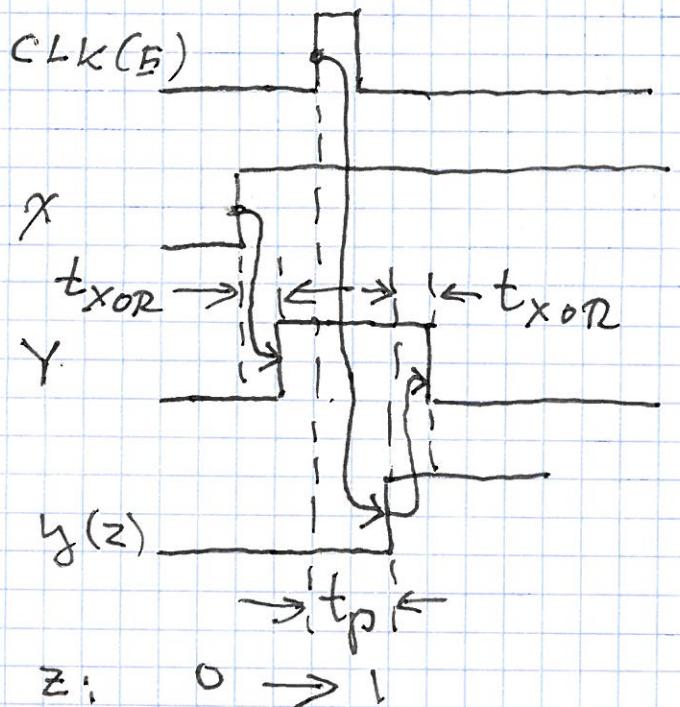


CORRECT
BEHAVIOR

PS	X(+)	Z = Y
Y	0	1
0	0	1
1	1	0

NS Y

INCORRECT
BEHAVIOR



A SOLUTION: EDGE-TRIGGERED CELL

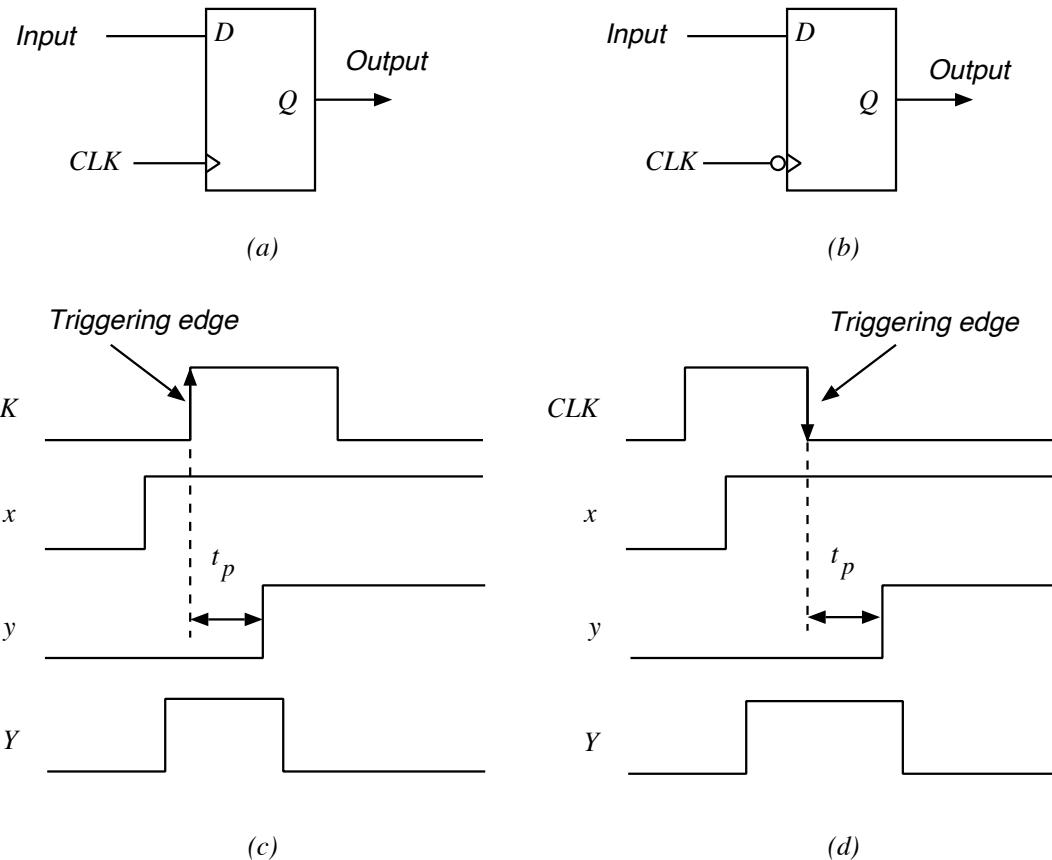


Figure 8.9: EDGE-TRIGGERED CELL: a) LEADING-EDGE-TRIGGERED CELL. b) TRAILING-EDGE-TRIGGERED CELL. c) LEADING-EDGE-TRIGGERED CELL IN NETWORK OF Figure 8.8. d) TRAILING-EDGE-TRIGGERED CELL IN NETWORK OF Figure 8.8.

ONE CHANGE PER CLOCK

SWAP A AND B:

$$T \leftarrow A$$

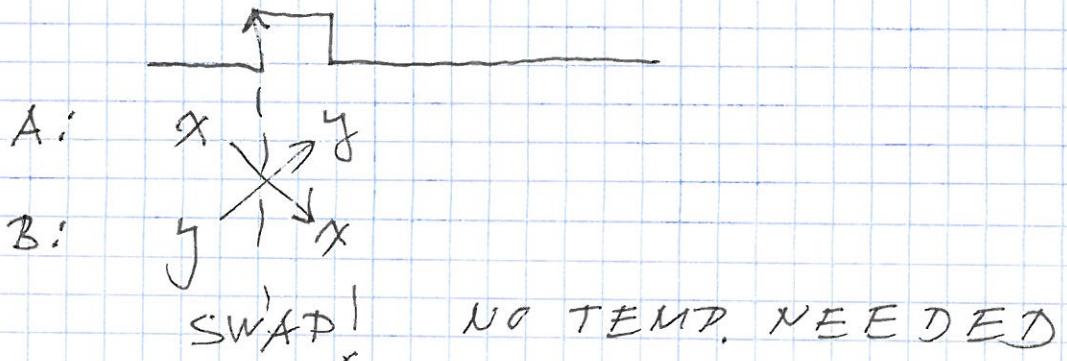
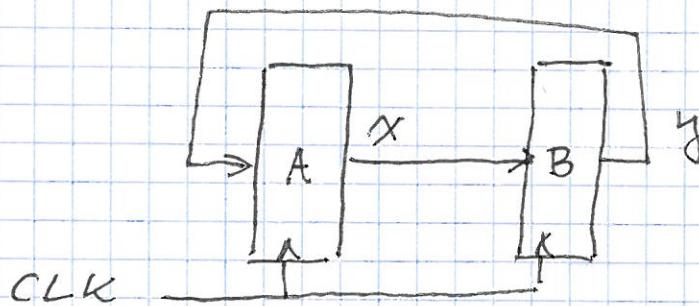
$$A \leftarrow B \quad (A=B)$$

$$B \leftarrow T \quad (B=A)$$

COULD BE DONE AS

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

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





IMPLEMENTATION: MASTER-SLAVE CELL

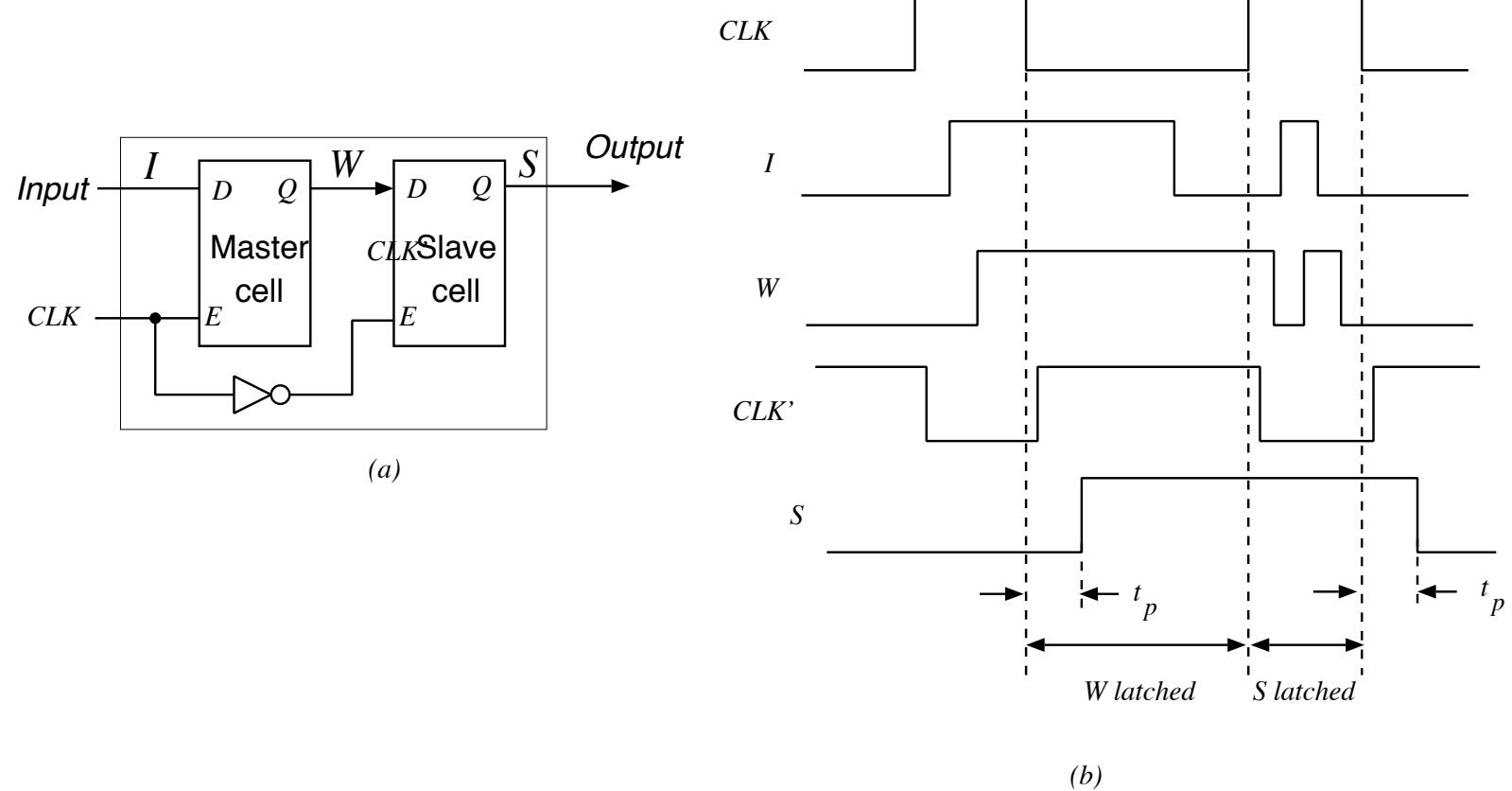


Figure 8.10: a) MASTER-SLAVE IMPLEMENTATION OF TRAILING-EDGE-TRIGGERED CELL. b) MASTER-SLAVE STATE CHANGE PROCESS.

PRACTICAL BASIC CELL: D flip-flop

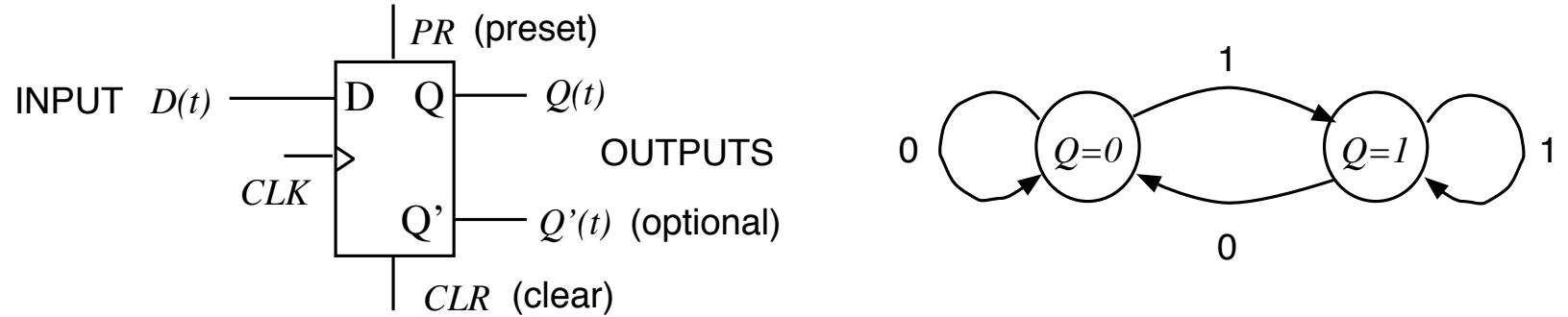


Figure 8.11: D FLIP-FLOP AND ITS STATE DIAGRAM.

$PS = Q(t)$	$D(t)$	
	0	1
0	0	1
1	0	1
		$NS = Q(t + 1)$

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

TIMING PARAMETERS OF A BINARY CELL

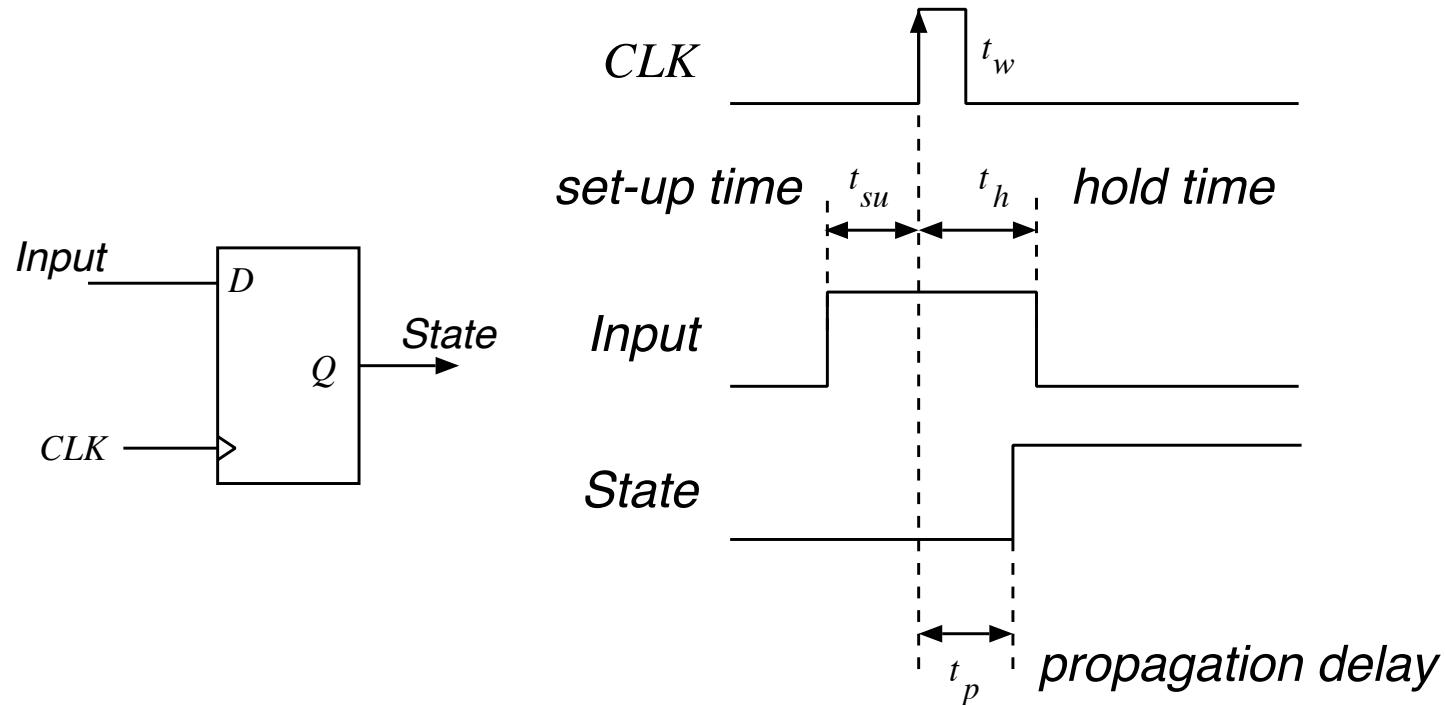


Figure 8.12: TIME BEHAVIOR OF CELL.

CHARACTERISTICS OF A CMOS D flip-flop

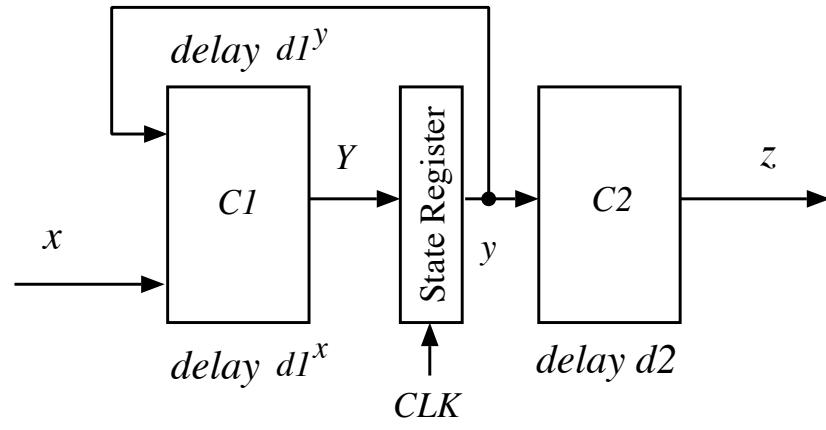
Delays					Input factor	Size
t_{pLH} [ns]	t_{pHL} [ns]	t_{su} [ns]	t_h [ns]	t_w [ns]	[std. loads]	[equiv. gates]
$0.49 + 0.038L$	$0.54 + 0.019L$	0.30	0.14	0.2	1	6

L : output load of the flip-flop

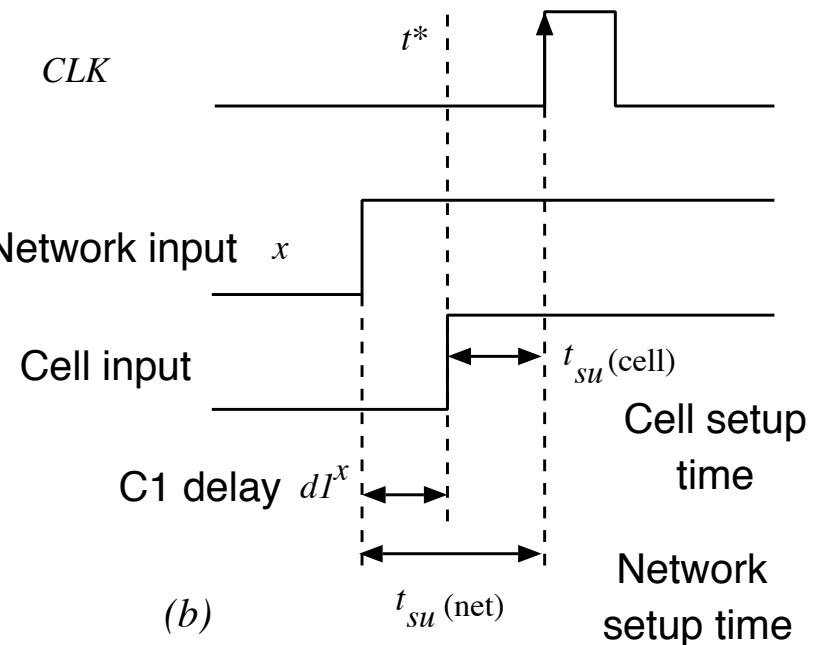
- THIS FLIP-FLOP HAS ONLY THE UNCOMPLEMENTED OUTPUT

TIMING CHARACTERISTICS OF SEQUENTIAL NETWORKS¹⁶

- NETWORK SET-UP TIME: $t_{su}^x(\text{net}) = d1^x + t_{su}(\text{cell})$



(a)



(b)

Figure 8.13: TIMING FACTORS IN SEQUENTIAL NETWORKS: a) THE NETWORK. b) NETWORK SET-UP TIME.

TIMING FACTORS (cont.)

- NETWORK HOLD TIME: $t_h(\text{net}) = t_h(\text{cell})$

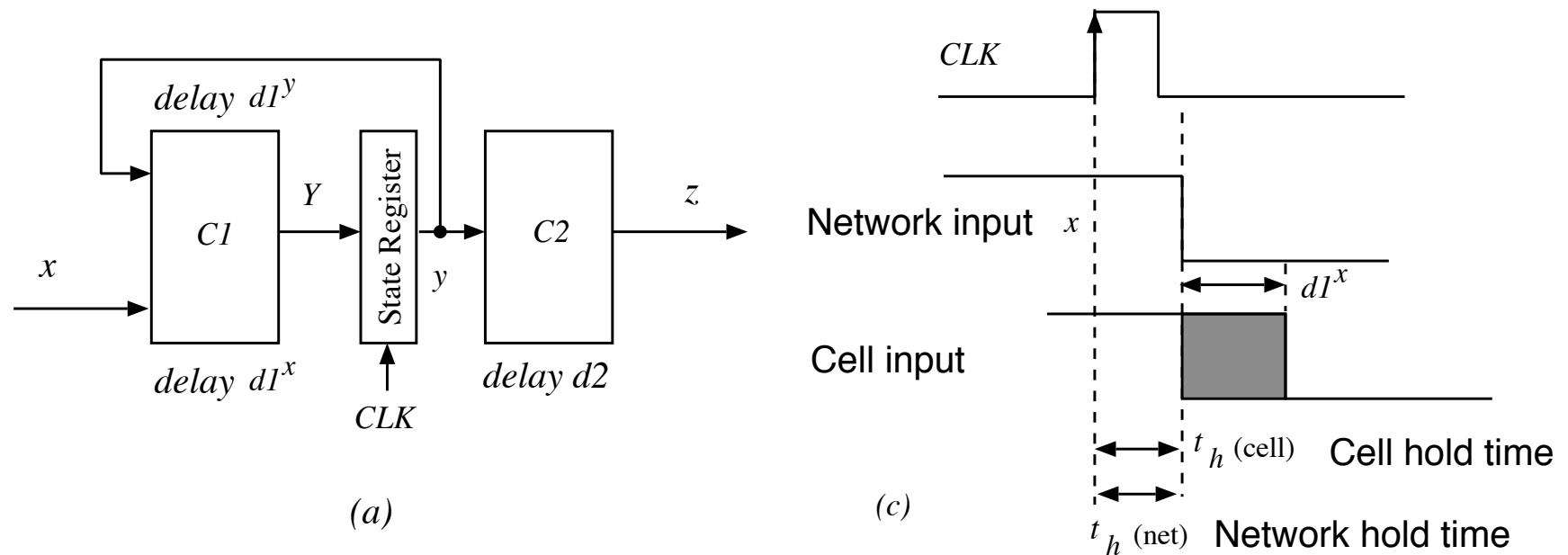


Figure 8.14: TIMING FACTORS IN SEQUENTIAL NETWORKS: a) THE NETWORK. c) NETWORK HOLD TIME.

TIMING FACTORS (Cont.)

- NETWORK PROPAGATION DELAY: $t_p(\text{net}) = t_p(\text{cell}) + d2$

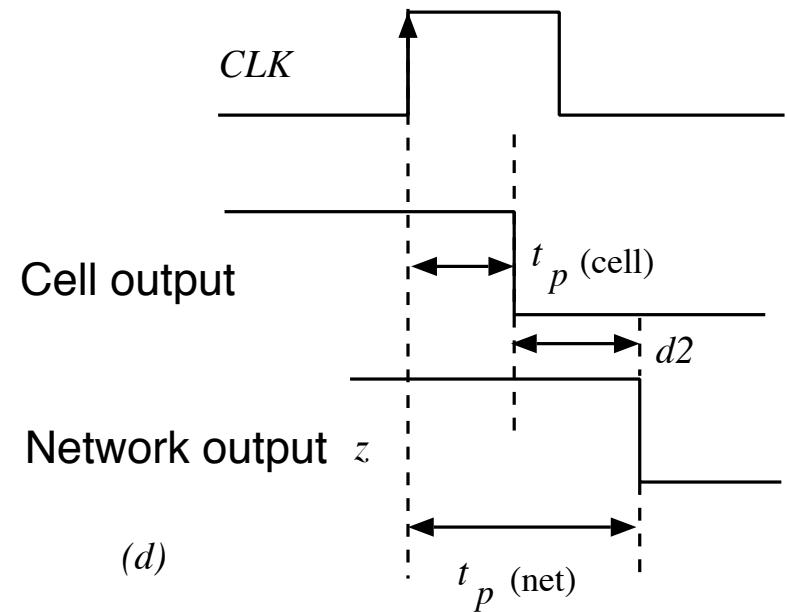
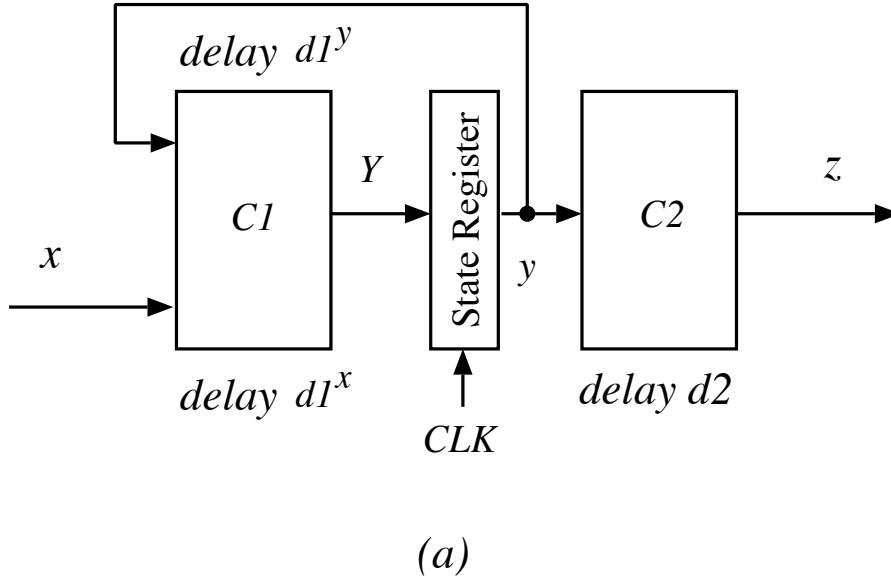


Figure 8.14: TIMING FACTORS IN SEQUENTIAL NETWORKS: a) THE NETWORK. d) NETWORK PROPAGATION DELAY.



MAXIMUM CLOCK FREQUENCY

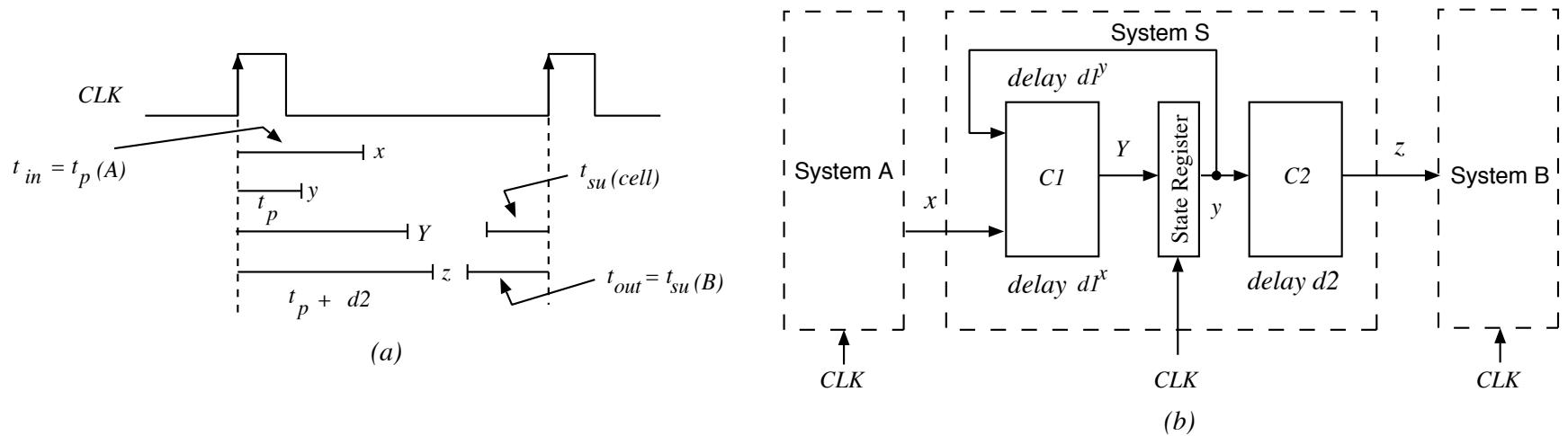


Figure 8.15: MAXIMUM CLOCK FREQUENCY: a) CLOCK PERIOD AND SIGNAL DELAYS. b) THE NETWORK.

- t_{in} - TIME BETWEEN TRIGGERING EDGE OF CLOCK AND STABILIZATION OF INPUT x
- t_{out} - TIME BETWEEN STABILIZATION OF OUTPUT z AND NEXT CLOCK TRIGGERING EDGE

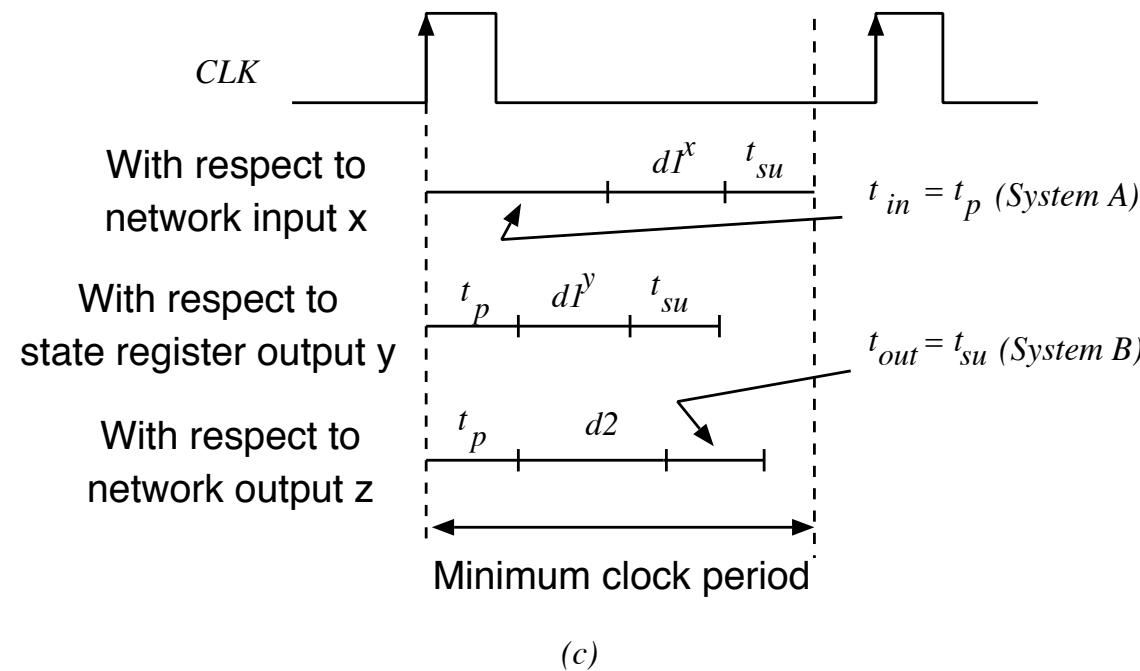
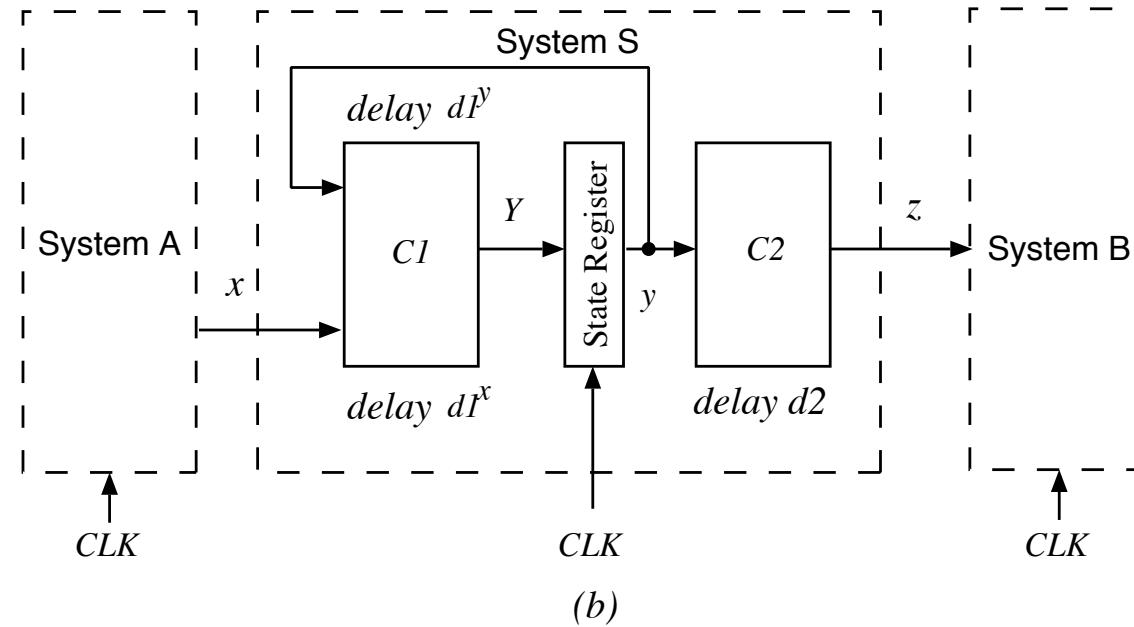


Figure 8.15: MAXIMUM CLOCK FREQUENCY: b) THE NETWORK. c) MINIMUM CLOCK PERIOD.

MAXIMUM CLOCK FREQUENCY (cont.)

$$T_{\min} = 1/f_{\max}$$

$$T_{\min} = \max[(t_{in} + t_{su}^x(net)), (t_p(cell) + t_{su}^y(net)), (t_p(net) + t_{out})]$$

$$t_h(cell) \leq t_p(cell)$$

$$T_{\min} = \max[(t_{in} + d1^x + t_{su}(cell)), (t_p(cell) + d1^y + t_{su}(cell)), (t_p(cell) + d1^z + t_{su}(cell))]$$

EXAMPLE 8.3

DETERMINE THE MAXIMUM CLOCK FREQUENCY

$$d1^x = d1^y = 2.5\text{ns}$$

$$d2 = 3\text{ns}$$

$$t_{su} = 0.3\text{ns}$$

$$t_p = 1\text{ns}$$

$$t_{in} = 2\text{ns}$$

$$t_{out} = 3\text{ns}$$

THE MINIMUM CLOCK PERIOD

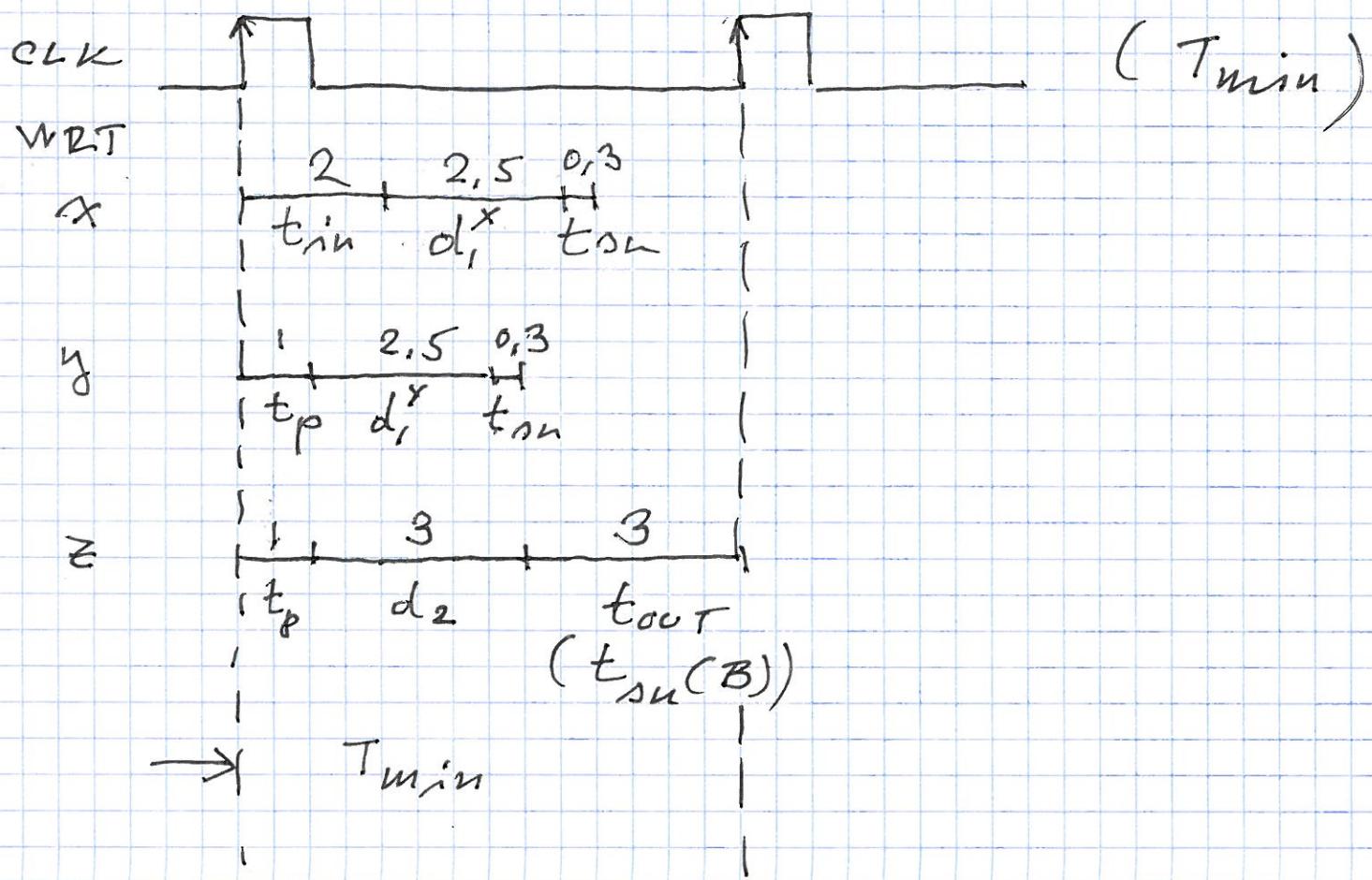
$$T_{\min} = \max[(2 + 2.5 + 0.3), (1 + 2.5 + 0.3), (1 + 3 + 3)] = 7[\text{ns}]$$

THE MAXIMUM FREQUENCY

$$f_{\max} = \frac{1}{7 \times 10^{-9}} \approx 140(\text{MHz})$$

CS M51A

EXAMPLE 8.3 :

 f_{\max} 

$$T_{min} = \max(4,8, 3,8, 7) = 7$$

$$f_{\max} = \frac{1}{T_{min}} = \frac{1}{7 \times 10^{-9}} \approx 140 \text{ MHz}$$

CLOCK SKEW

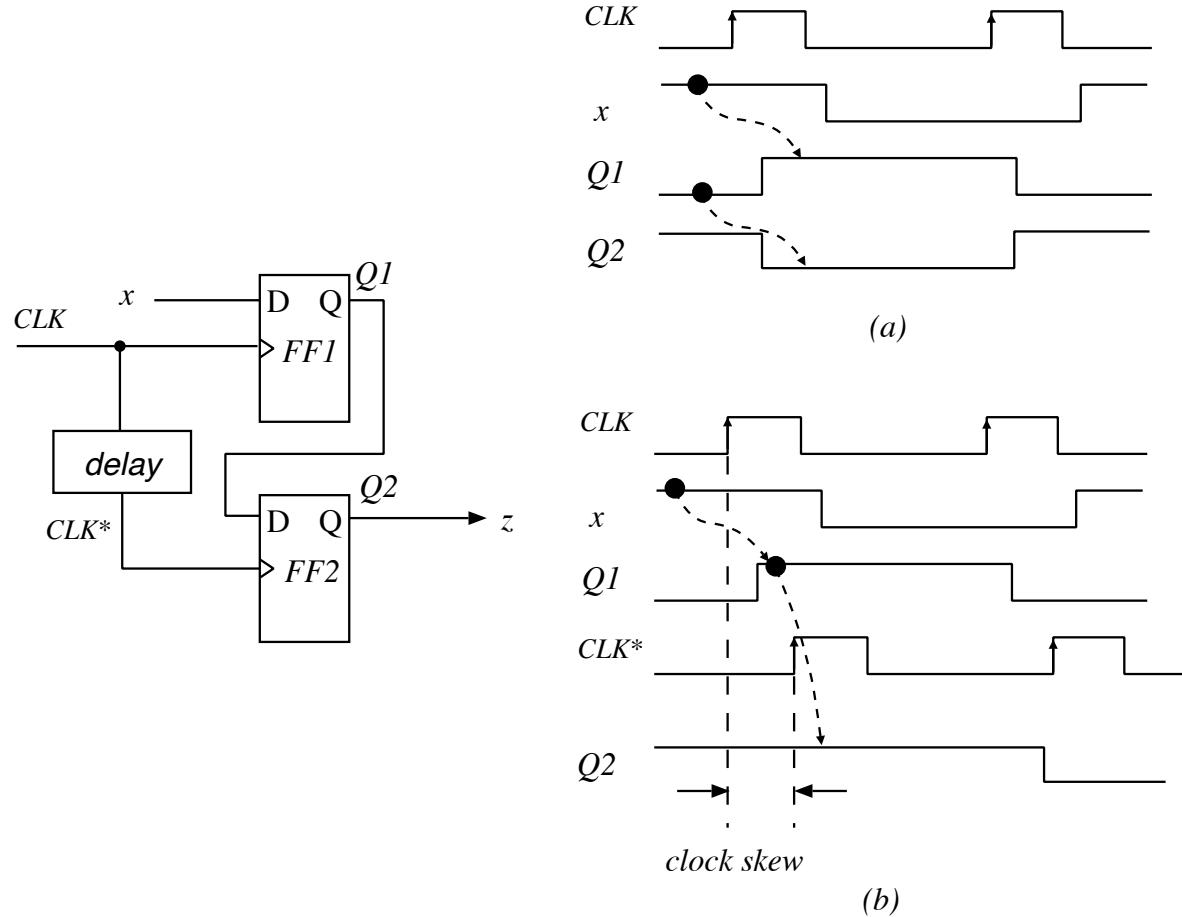
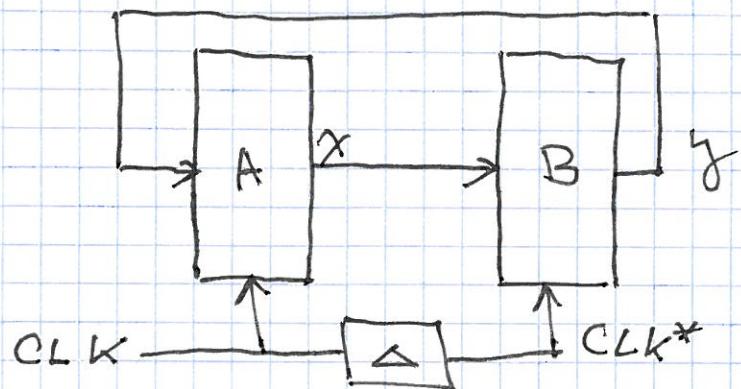


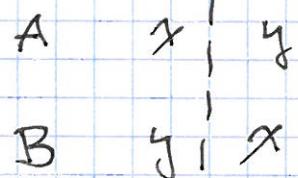
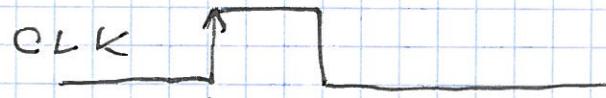
Figure 8.16: a) NETWORK BEHAVIOR WITHOUT CLOCK SKEW. b) NETWORK BEHAVIOR WITH INADMISSIBLE CLOCK SKEW.

AN EXAMPLE

SWAP A AND B

 $\Delta = 0$ (NO SKEW)

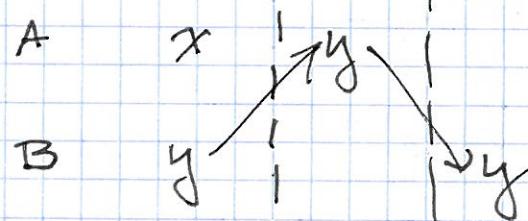
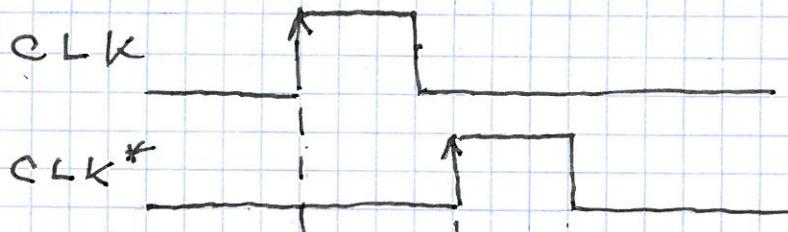
$$\text{CLK}^* = \text{CLK}$$



CORRECT

 $\Delta \neq 0$ (SKew)

CLK* COMES AFTER CLK



X - LOST

ANALYSIS OF CANONICAL SEQUENTIAL NETWORKS

24

1. ANALYZE COMBINATIONAL NETWORK
DETERMINE THE TRANSITION AND OUTPUT FUNCTIONS

2. DETERMINE HIGH-LEVEL SPECIFICATION OF STATE DESCRIPTION OUTPUT FUNCTIONS.

3. IF DESIRED (OR REQUIRED), DETERMINE TIME BEHAVIOR

EXAMPLE 8.4: ANALYSIS

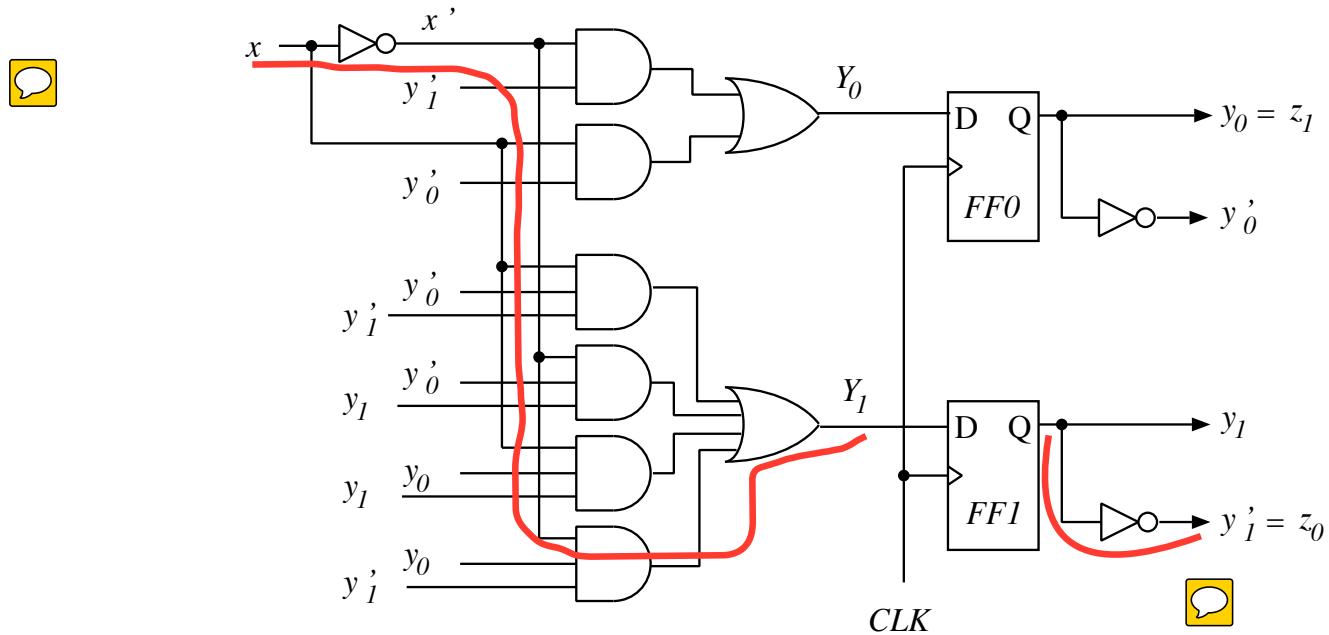


Figure 8.17: SEQUENTIAL NETWORK IN Example 8.4.

State transition

$$Y_0 = x'y'_1 + xy'_0$$

$$Y_1 = xy'_0y'_1 + x'y'_0y_1 + xy_0y_1 + x'y_0y'_1$$

Output

$$z_0 = y'_1$$

$$z_1 = y_0$$

EXAMPLE 8.4 (cont.)

- STATE-TRANSITION AND OUTPUT FUNCTIONS:

PS	Input		
y_1y_0	$x = 0$	$x = 1$	
00	01	11	01
01	11	00	11
10	10	01	00
11	00	10	10

	Y_1Y_0	z_1z_0
NS		Output



- CODES:

x	x
0	a
1	b

$z_1 z_0$	z
00	c
01	d
10	e
11	f

$y_1 y_0$	s
00	S_0
01	S_1
10	S_2
11	S_3

EXAMPLE 8.4 (cont.)

- HIGH-LEVEL SPECIFICATION:

Input: $x(t) \in \{a, b\}$

Output: $z(t) \in \{c, d, e, f\}$

State: $s(t) \in \{S_0, S_1, S_2, S_3\}$

Initial state: $s(0) = S_2$

Functions: The state-transition and output functions

PS	$x(t) = a$	$x(t) = b$	
S_0	S_1	S_3	d
S_1	S_3	S_0	f
S_2	S_2	S_1	c
S_3	S_0	S_2	e
	NS		$z(t)$

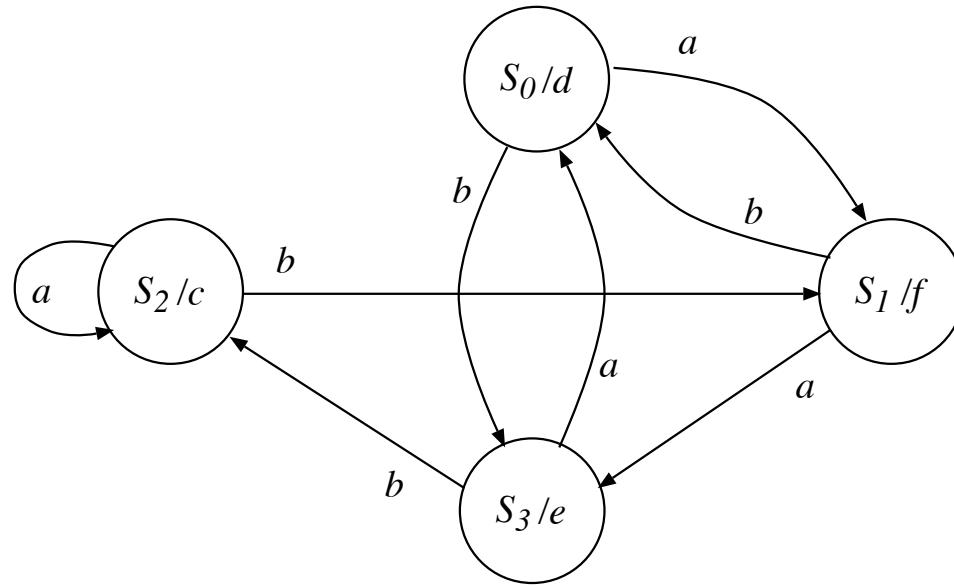


Figure 8.18: a) STATE DIAGRAM FOR SEQUENTIAL NETWORK.

$x(t)$	a	a	b	a	b	b	a	b	a	a	b	b	b	a
$s(t)$	S_2	S_2	S_2	S_1	S_3	S_2	S_1	S_3	S_2	S_2	S_2	S_1	S_0	S_3
$z(t)$	c	c	c	f	e	c	f	e	c	c	c	f	d	e

Figure 8.18: b) A sequence of input-output pairs.

OBJECTIVES:

$$a) (X: 0 \rightarrow 1) \text{ & } b) (Z_0: 1 \rightarrow 0)$$

**NOT RELATED
DELAYS**

— OTHER CASES: SIMIL.

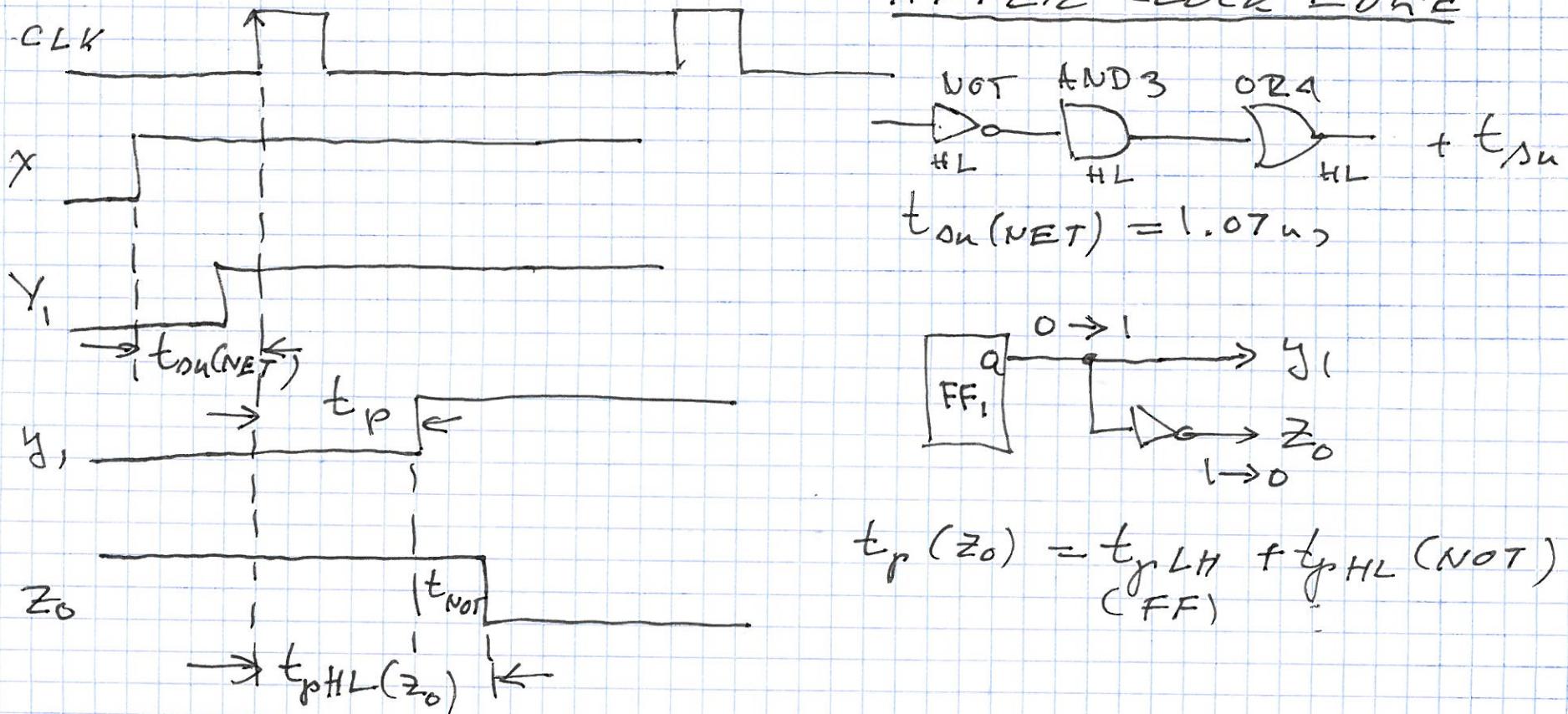
a) FIND $t_{\text{on}}(\text{NET})$

— LATEST TIME BEFORE
CLOCK EDGE FOR X
TO CHANGE

b) FIND $t_p(\text{OUTPUT})$

— DELAY TO PRODUCE OUTPUT

AFTER CLOCK EDGE



PROPAGATION DELAY x to z_0 :

INPUT LOAD FACTORS:

$$I_x = 4$$

SET-UP TIME:

$$\begin{aligned} t_{su}(net) &= t_{pHL}(\text{NOT}) + t_{pHL}(\text{AND3}) \\ &\quad + t_{pHL}(\text{OR4}) + t_{su} \\ &= (0.05 + 0.017 \times 3) + (0.18 + 0.018) \\ &\quad + (0.45 + 0.025) + 0.3 \\ &= 1.07 \text{ [ns]} \end{aligned}$$

HOLD TIME:

$$t_h(net) = 0.14 \text{ [ns]}$$

PROPAGATION DELAY:

$$\begin{aligned} t_p(z_0) &= t_{pLH}(\text{FF}) + t_{pHL}(\text{NOT}) \\ &= (0.49 + 0.038 \times 3) \\ &\quad + (0.05 + 0.017 \times (L + 3)) \\ &= 0.70 + 0.017L \text{ [ns]} \\ &\quad (\text{load of NOT is } L + 3, \text{ load of FF is 3}) \end{aligned}$$

SIZE:

$$\begin{aligned} &= 6 \times 2 + 2 + 3 + 2 \times 6 + 3 \times 1 \\ &= 32 \text{ equivalent gates.} \end{aligned}$$

1. TRANSFORM THE TRANSITION AND OUTPUT FUNCTIONS
2. SPECIFY A STATE REGISTER TO ENCODE THE REQUIRED NUMBER OF STATES
3. DESIGN THE REQUIRED COMBINATIONAL NETWORK

EXAMPLE 8.5: DESIGN

Input: $x(t) \in \{a, b, c\}$

Output: $z(t) \in \{0, 1\}$

State: $s(t) \in \{A, B, C, D\}$

Initial state: $s(0) = A$

Functions: The state-transition and output functions

PS	Input		
	$x = a$	$x = b$	$x = c$
A	C,0	B,1	B,0
B	D,0	B,0	A,1
C	A,0	D,1	D,0
D	B,0	A,0	D,1
	NS, z		

EXAMPLE 8.5 (cont.)

- CODING:

Input code			State code		
x	x_1	x_0	s	y_1	y_0
a	0	1	A	0	0
b	1	0	B	1	0
c	1	1	C	0	1
			D	1	1

TO SIMPLIFY THE PROCESS, MAKE BINARY STATE

TABLE USE K-MAP LABELS

PS	a	b	c		PS	a	c	b		y_1, y_0	00	01	11	10	x_1, x_0	
00 A	0, 1	1, 0	1, 1		00 A	0, 0	0, 1	1, 1	1, 0	00	0, 1, 0	1, 0, 0	1, 0, 1	1, 0, 1	0, 1, 0	
1, 0 B	D, 0	B, 0	A, 1		01 C	-	C, 0	B, 0	B, 1	00	-	0, 1, 0	1, 0, 0	1, 0, 1	1, 0, 1	0, 0, 0
0, 1 C	A, 0	D, 1	D, 0		11 D	-	A, 0	D, 0	D, 1	01	-	0, 0, 0	1, 1, 0	1, 1, 1	1, 1, 1	1, 0, 0
1, 1 D	B, 0	A, 0	D, 1		10 B	-	B, 0	D, 1	A, 0	11	-	1, 0, 0	1, 1, 1	0, 0, 0	0, 0, 0	1, 0, 0

	x_0				
y_1	0	1	0	1	y_0
	0	1	1	0	
0	1	0	1	0	0
1	0	1	0	1	1
	1	0	0	1	

	x_0				
y_1	0	1	0	0	y_0
	1	0	0	0	
0	1	1	0	0	0
1	0	0	0	0	1
	0	1	0	0	

	x_0				
y_1	0	0	1	0	y_0
	1	0	0	1	
0	1	0	0	0	0
1	0	0	0	0	1
	0	1	0	0	

$$\begin{aligned}
 Y_1 &= x_0' y_0' + y_1' x_1 + y_1 x_1' \\
 &\quad + y_1 y_0 x_0 \\
 Y_0 &= y_0' x_1' + y_0 x_1 x_0 \\
 &\quad + y_1' y_0 x_1 \\
 Z &= y_1' x_0' + y_1 x_1 x_0
 \end{aligned}$$

EXAMPLE 8.5 (cont.)

- STATE-TRANSITION AND OUTPUT FUNCTIONS



PS	x_1x_0		
y_1y_0	01	10	11
00	01,0	10,1	10,0
10	11,0	10,0	00,1
01	00,0	11,1	11,0
11	10,0	00,0	11,1

	Y_1Y_0, z
	NS, Output

Y_1	$\frac{x_0}{x_1}$
y_1	$\frac{-011}{-101}$

Y_0	$\frac{x_0}{x_1}$
y_1	$\frac{-100}{-100}$

z	$\frac{x_0}{x_1}$
y_1	$\frac{-001}{-010}$

- NEXT-STATE AND OUTPUT EXPRESSIONS

$$Y_1 = y'_1 x_1 + y_1 x'_1 + y'_0 x'_0 + y_0 x_1 x_0$$

$$Y_0 = y'_0 x'_1 + y'_1 y_0 x_1 + y_0 x_1 x_0$$

$$z = y'_1 x'_0 + y_1 x_1 x_0$$

EXAMPLE 8.5 (cont.)

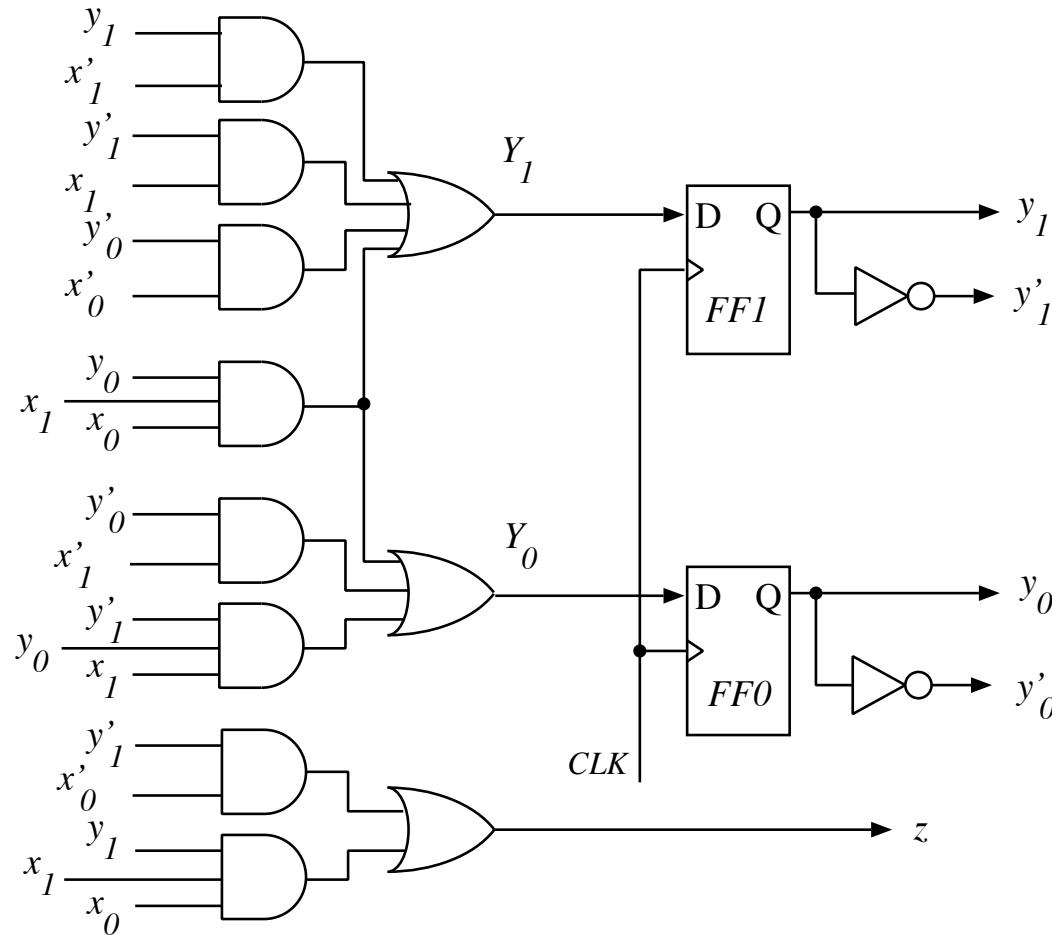
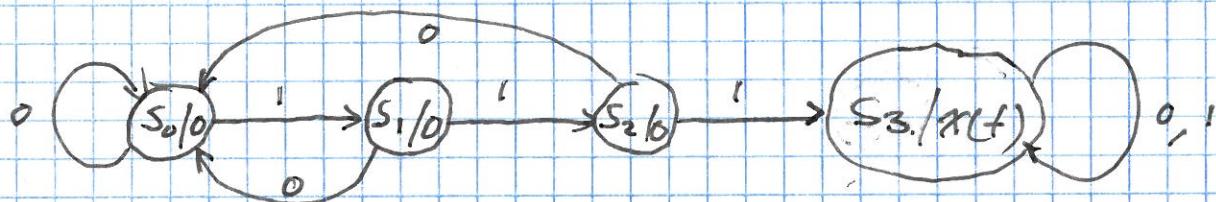


Figure 8.19: SEQUENTIAL NETWORK IN Example 8.5.

EXTRA EXAMPLE:

$$z(t) = \begin{cases} 0 & \text{UNTIL THE FIRST INSTANCE OF} \\ & \text{3 CONSECUTIVE 1'S HAS BEEN} \\ & \text{RECEIVED} \\ x(t) & \text{AFTERWARDS} \end{cases}$$

x	0	0	1	1	0	1	1	1	0	0	1	...		
z	0	0	0	0	0	0	0	0	1	1	0	0	1	...



PS	$x(t)$	z
$y_1 y_0$	0	1
S_0	00	00
S_1	01	00
S_2	10	10
S_3	11	11

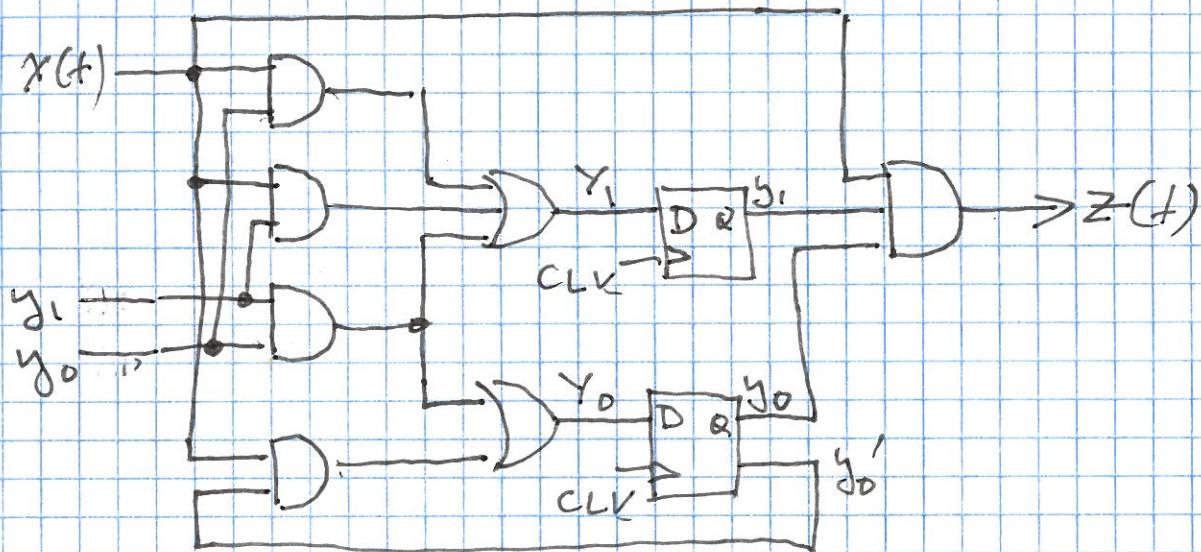
y_1 , y_0
NS

$$y_1: \begin{array}{|c|c|c|c|} \hline & x & & \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 \\ \hline \end{array} \quad y_1 = y_1 x + y_0 x + y_0 y_0$$

$$y_0: \begin{array}{|c|c|c|c|} \hline & x & & \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline \end{array} \quad y_0 = y_0 x + y_1 y_0$$



CANONICAL DESIGN



- HAVE ADDITIONAL FUNCTIONALITY WHICH MAY SIMPLIFY OVERALL DESIGN

CONSIDER D FLIP-FLOP

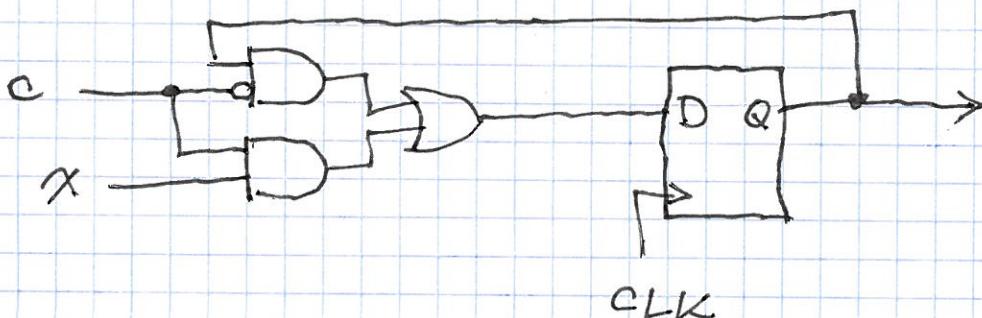
$$Q(t+1) = D(t)$$

HOW DO YOU KEEP STATE UNCHANGED?

- INHIBIT CLOCK (GATED CLOCK)

→ NOT DESIRABLE,
EXTRA DELAY, ETC.

- RELOAD THE PRESENT STATE IF NO CHANGE DESIRED



CLK — PRESENT
ALWAYS

$$C = \begin{cases} 1 & \text{CHANGE STATE} \\ 0 & \text{KEEP STATE} \end{cases}$$

CLK

$$CLK^* = \begin{cases} CLK & \text{IF } C = 1 \\ 0 & \text{IF } C = 0 \end{cases}$$

→ NO CHANGE

— WASTEFUL ACTIVITY
(WASTED ENERGY)
WHEN RELOADING
SAME STATE

SR FLIP-FLOP

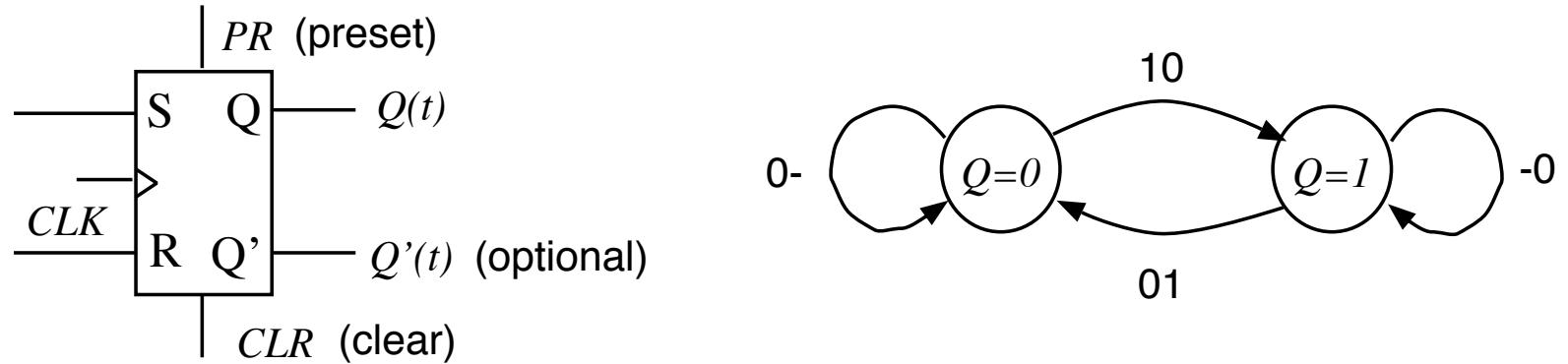


Figure 8.20: SR FLIP-FLOP AND ITS STATE DIAGRAM.

$PS = Q(t)$		$S(t)R(t)$			
		00	01	10	11
0		0	0	1	-
1		1	0	1	-
				$NS = Q(t + 1)$	

$$Q(t+1) = Q(t)R'(t) + S(t) \text{ restriction: } R(t) \cdot S(t) = 0$$

JK FLIP-FLOP

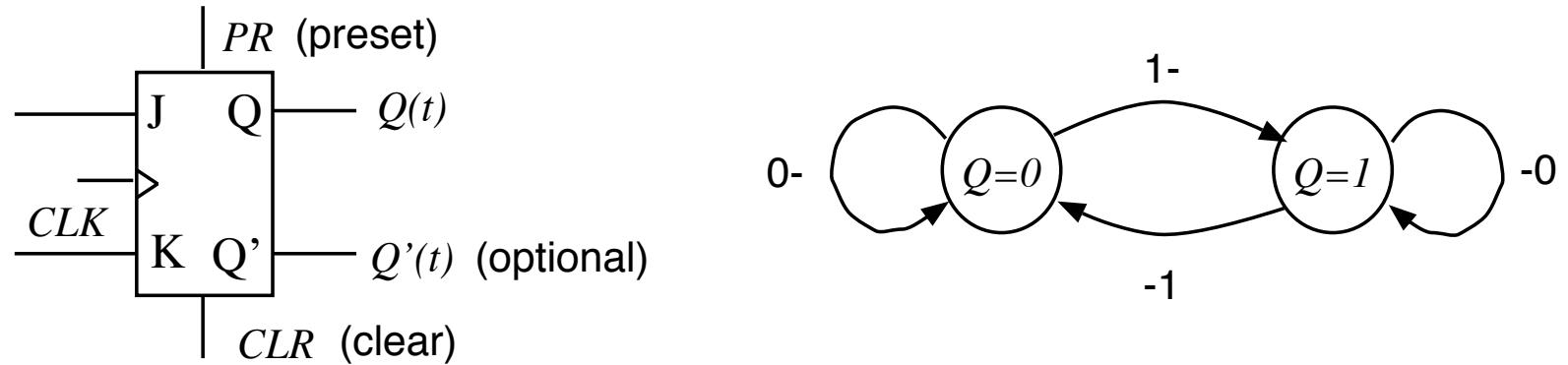


Figure 8.21: JK FLIP-FLOP AND ITS STATE DIAGRAM.

$PS = Q(t)$		$J(t)K(t)$			
		00	01	10	11
0		0	0	1	1
1		1	0	1	0
				$NS = Q(t + 1)$	

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

T (Toggle) FLIP-FLOP

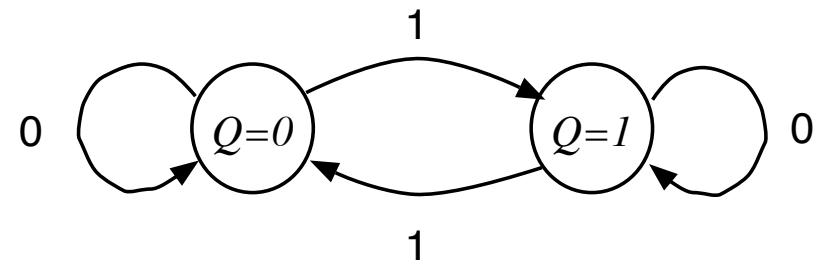
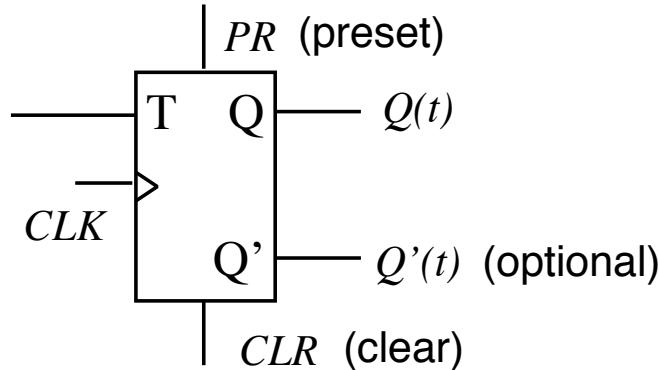


Figure 8.22: T FLIP-FLOP AND ITS STATE DIAGRAM.

$PS = Q(t)$	$T(t)$	
	0	1
0	0	1
1	1	0
$NS = Q(t + 1)$		

$$Q(t + 1) = Q(t) \oplus T(t)$$

IMPLEMENTING ONE FF TYPE WITH ANOTHER

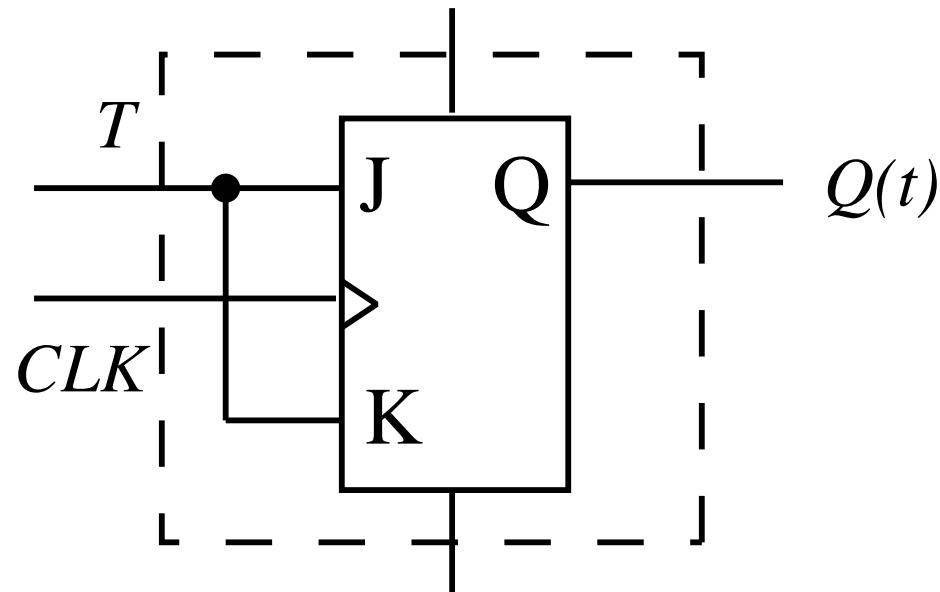
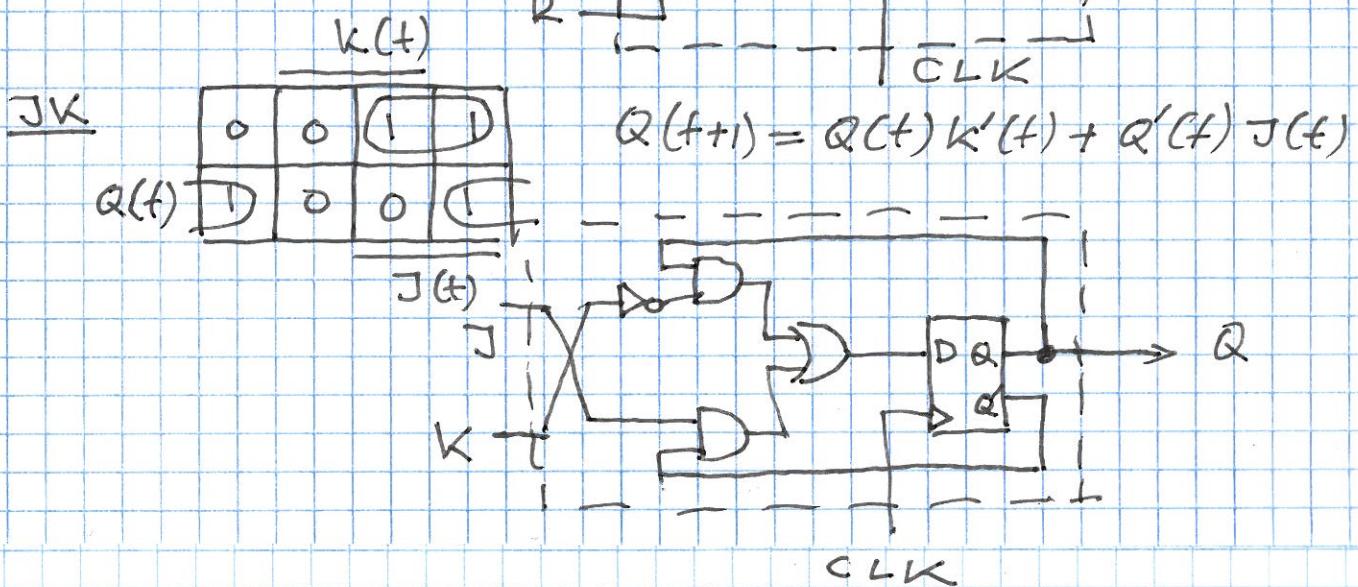
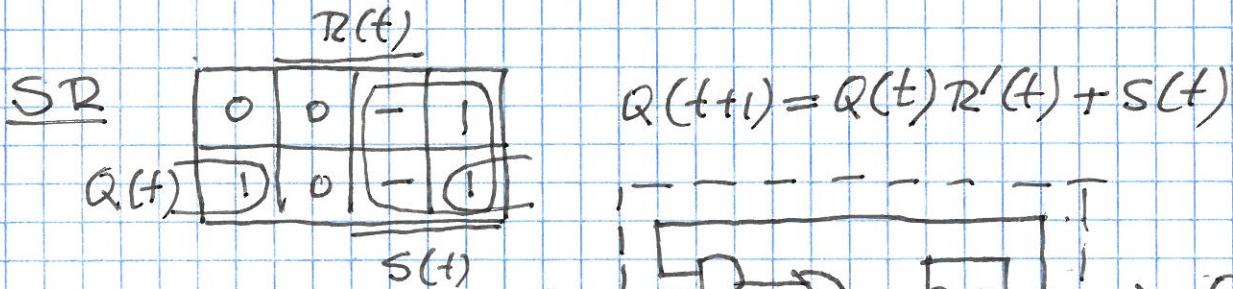


Figure 8.23: T FLIP-FLOP IMPLEMENTED WITH JK FLIP-FLOP.

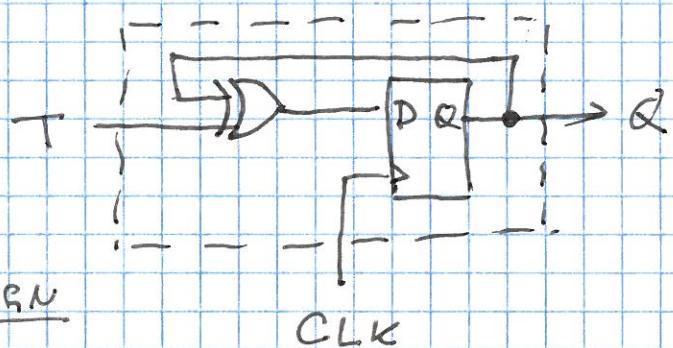
CANONICAL DESIGN OF FLIP-FLOPS

— BINARY CELL USED IS D-FLIP-FLOP



<u>I</u>	$T(t)$
$Q(t)$	$\begin{array}{ c c } \hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array}$

$$Q(t+1) = Q(t) \oplus T(t)$$



EXERCISE:

<u>USE AS B-CELL</u>	<u>DESIGN</u>
SR	D, JK, T
JK	D, SR, T
T	D, SR, JK



ANALYSIS OF NETWORKS WITH FLIP-FLOPS

1. OBTAIN THE TRANSITION FUNCTION OF THE NETWORK
 - (a) DETERMINE THE INPUTS TO THE FLIP-FLOPS
 - (b) USE THE TRANSITION FUNCTION OF THE FLIP-FLOPS
TO DETERMINE THE NEXT STATE
2. OBTAIN THE OUTPUT FUNCTION
3. DETERMINE A SUITABLE HIGH-LEVEL SPECIFICATION

CHARACTERISTICS OF A FAMILY OF CMOS FLIP-FLOPS

42

FF type	Delays					Input factor	Size
	t_{pLH} [ns]	t_{pHL} [ns]	t_{su} [ns]	t_h [ns]	t_w [ns]		
D	$0.49 + 0.038L$	$0.54 + 0.019L$	0.30	0.14	0.20	1	6
JK	$0.45 + 0.038L$	$0.47 + 0.022L$	0.41	0.23	0.20	1	8

L : output load of the flip-flop

These flip-flops have only uncomplemented outputs

EXAMPLE 8.6: ANALYSIS

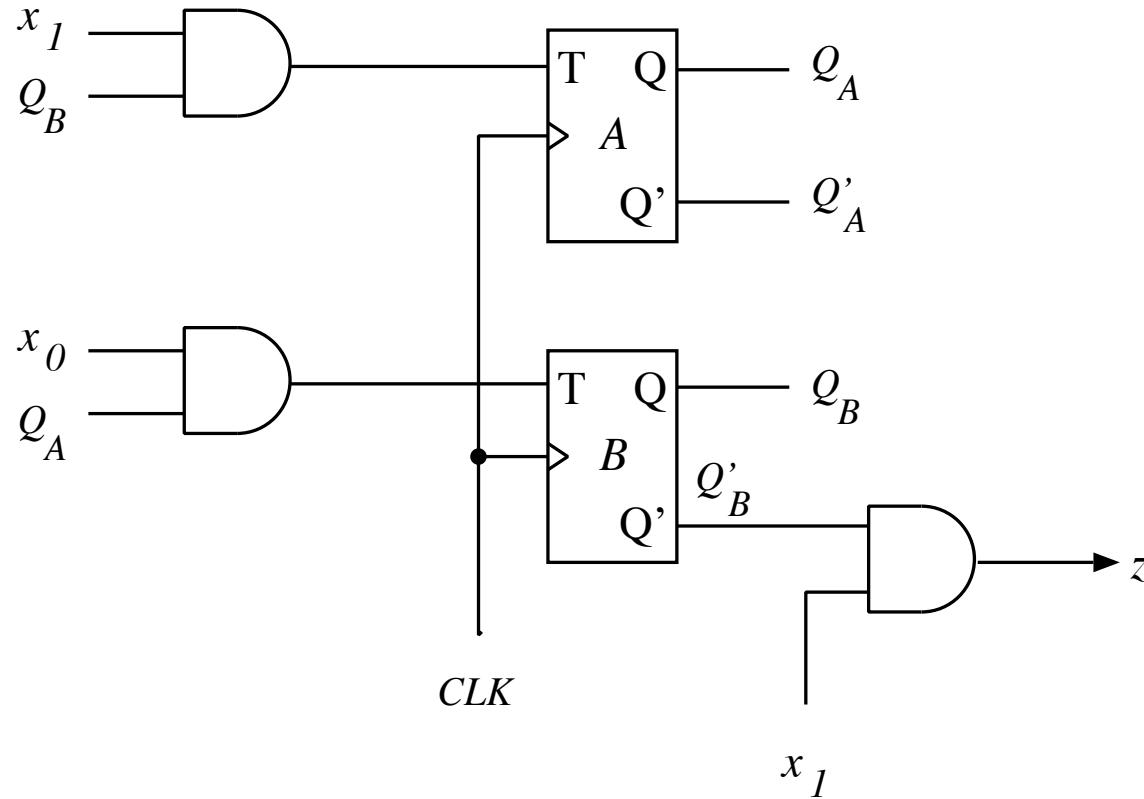


Figure 8.24: SEQUENTIAL NETWORK FOR Example 8.6.

$$\begin{aligned}
 T_A &= x_1 Q_B & Q_A(t+1) &= Q_A(t) \oplus x_1 Q_B(t) \\
 T_B &= x_0 Q_A & Q_B(t+1) &= Q_B(t) \oplus x_0 Q_A(t) \\
 z(t) &= x_1(t) Q'_B(t)
 \end{aligned}$$

EXAMPLE 8.6 (cont.)

- STATE-TRANSITION AND OUTPUT FUNCTIONS

PS	Input							
$Q_A Q_B$	$x_1 x_0$				$x_1 x_0$			
	00	01	10	11	00	01	10	11
00	00	00	00	00	0	0	1	1
01	01	01	11	11	0	0	0	0
10	10	11	10	11	0	0	1	1
11	11	10	01	00	0	0	0	0
	$Q_A Q_B$				z			
	NS				Output			

EXAMPLE 8.6 (cont.)

- CODING:

Q_A	Q_B	s	x_1	x_0	x
0	0	S_0	0	0	a
0	1	S_1	0	1	b
1	0	S_2	1	0	c
1	1	S_3	1	1	d

EXAMPLE 8.6 (cont.)

- HIGH-LEVEL DESCRIPTION:

INPUT: $x(t) \in \{a, b, c, d\}$

OUTPUT: $z(t) \in \{0, 1\}$

STATE: $s(t) \in \{S_0, S_1, S_2, S_3\}$

INITIAL STATE: $s(0) = S_0$

Functions: the state-transition and output functions

PS	x				x			
	a	b	c	d	a	b	c	d
S_0	S_0	S_0	S_0	S_0	0	0	1	1
S_1	S_1	S_1	S_3	S_3	0	0	0	0
S_2	S_2	S_3	S_2	S_3	0	0	1	1
S_3	S_3	S_2	S_1	S_0	0	0	0	0
	NS				z			

EXAMPLE 8.7: ANALYSIS

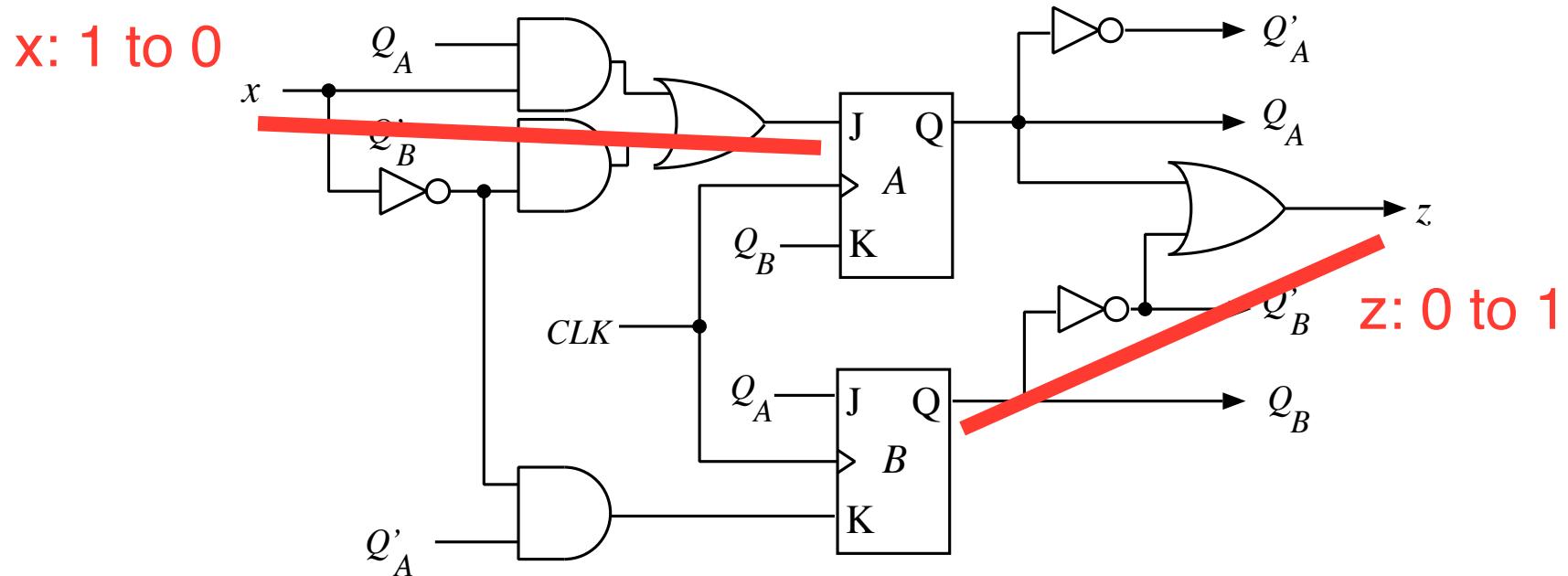


Figure 8.25: SEQUENTIAL NETWORK FOR Example 8.7

$$\begin{array}{ll} J_A = x'Q'_B + xQ_A & K_A = Q_B \\ J_B = Q_A & K_B = x'Q'_A \end{array}$$

$$z = Q_A + Q'_B$$

EXAMPLE 8.7 (cont.)

$$\begin{aligned} J_A &= x'Q'_B + xQ_A & K_A &= Q_B \\ J_B &= Q_A & K_B &= x'Q'_A \\ z &= Q_A + Q'_B \end{aligned}$$

$$\begin{aligned} Q_A(t+1) &= Q_A K'_A + Q'_A J_A \\ &= Q_A Q'_B + Q'_A (x'Q'_B + xQ_A) \\ &= Q'_B (Q_A + x') \end{aligned}$$

$$\begin{aligned} Q_B(t+1) &= Q_B K'_B + Q'_B J_B \\ &= Q_B (x + Q_A) + Q'_B Q_A \\ &= Q_B x + Q_A \end{aligned}$$

EXAMPLE 8.7 (cont.)

- STATE-TRANSITION AND OUTPUT FUNCTIONS

PS	NS		Output z
	$x = 0$	$x = 1$	
$Q_A Q_B$	$Q_A Q_B$	$Q_A Q_B$	
00	10	00	1
01	00	01	0
10	11	11	1
11	01	01	1

- STATE CODING

Q_A	Q_B	S
0	0	S_0
0	1	S_1
1	0	S_2
1	1	S_3

EXAMPLE 8.7 (cont.)

- HIGH-LEVEL DESCRIPTION

INPUT: $x(t) \in \{0, 1\}$

OUTPUT: $z(t) \in \{0, 1\}$

STATE: $s(t) \in \{S_0, S_1, S_2, S_3\}$

INITIAL STATE: $s(0) = S_0$

Functions: The state-transition and output functions

PS	Input		
	$x = 0$	$x = 1$	
S_0	S_2	S_0	1
S_1	S_0	S_1	0
S_2	S_3	S_3	1
S_3	S_1	S_1	1
	NS		z

EXAMPLE 8.7 (cont.)

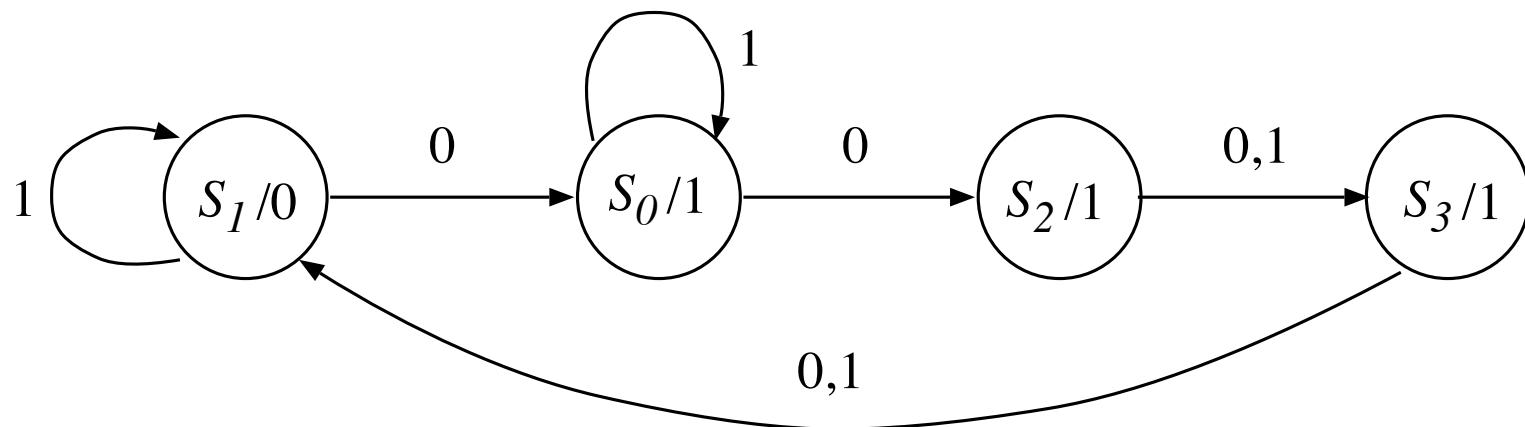


Figure 8.26: STATE DIAGRAM IN Example 8.7.

OTHER CHARACTERISTICS (Example 8.7)

INPUT LOAD FACTOR: $I_x = 2$

SET-UP TIME:

$$\begin{aligned}
 t_{su}(net) &= t_{pLH}(\text{NOT}) + t_{pLH}(\text{AND}) + t_{pLH}(\text{OR}) \\
 &\quad + t_{su}(\text{FF}) \\
 &= (0.02 + 0.038 \times 2) + (0.15 + 0.037) \\
 &\quad + (0.12 + 0.037) + 0.41 \\
 &= 0.86 \text{ [ns]}
 \end{aligned}$$

x: 1 to 0
other case similar

HOLD TIME: $t_h(net) = 0.23 \text{ [ns]}$

PROPAGATION DELAY:

$$\begin{aligned}
 t_p(net) &= t_{pHL}(\text{FF}) + t_{pLH}(\text{NOT}) + t_{pLH}(\text{OR}) \\
 &= (0.47 + 0.022 \times 2) + (0.02 + 0.038 \times 2) \\
 &\quad + (0.12 + 0.037L) \\
 &= 0.73 + 0.037L \text{ [ns]}
 \end{aligned}$$

z: 0 to 1

SIZE:

$$\begin{aligned}
 &= 3 + 2 \times 5 + 8 \times 2 \\
 &= 29 \text{ equivalent gates}
 \end{aligned}$$

DESIGN OF SEQUENTIAL FLIP-FLOP NETWORKS

- EXCITATION FUNCTION $E(Q(t), Q(t + 1))$

FROM	TO	INPUTS SHOULD BE
$Q(t) = 0$	$Q(t + 1) = 0$	$S(t) = 0, R(t) = dc$
$Q(t) = 0$	$Q(t + 1) = 1$	$S(t) = 1, R(t) = 0$
$Q(t) = 1$	$Q(t + 1) = 0$	$S(t) = 0, R(t) = 1$
$Q(t) = 1$	$Q(t + 1) = 1$	$S(t) = dc, R(t) = 0$

EXCITATION FUNCTIONS

D flip-flop

<i>PS</i>	<i>NS</i>	
	0	1
0	0	1
1	0	1
	<i>D(t)</i>	

SR flip-flop

<i>PS</i>	<i>NS</i>	
	0	1
0	0-	10
1	01	-0
	<i>S(t)R(t)</i>	

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

JK flip-flop

<i>PS</i>	<i>NS</i>	
	0	1
0	0-	1-
1	-1	-0
	<i>J(t)K(t)</i>	

T flip-flop

<i>PS</i>	<i>NS</i>	
	0	1
0	0	1
1	1	0
	<i>T(t)</i>	

$$T(t) = Q(t) \oplus Q(t + 1)$$

THE DESIGN PROCEDURE

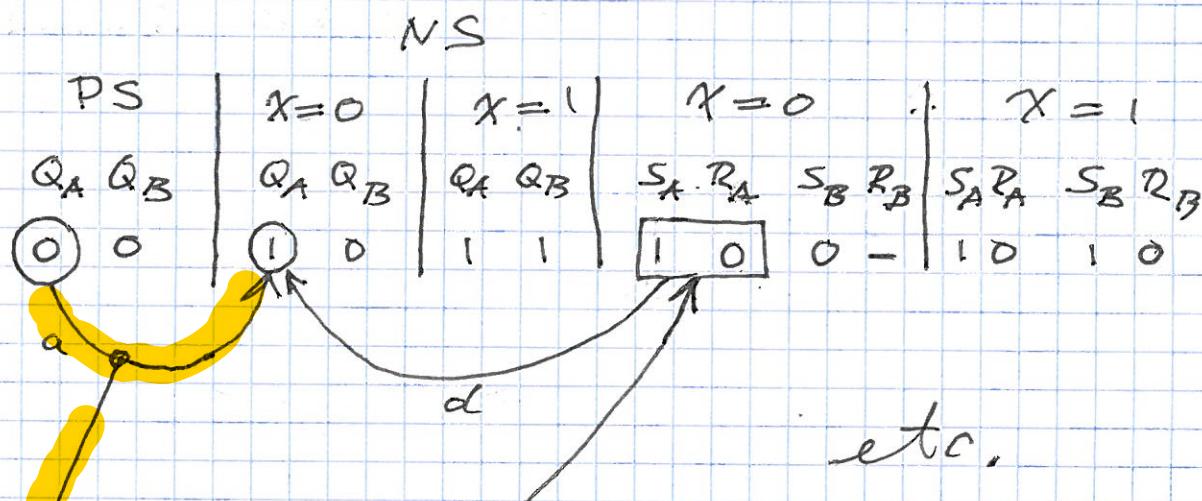
1. OBTAIN A BINARY DESCRIPTION OF THE SYSTEM
2. SELECT THE TYPE OF FLIP-FLOP
3. DETERMINE THE INPUTS TO THE FLIP-FLOPS (use the excitation function)
4. DESIGN A COMBINATIONAL NETWORK

CS MSIA

FF EXCITATION FUNCTION

- GIVEN STATE TABLE, FOR EACH $Q(t)$ AND EACH INPUT WE KNOW THE CORRESPONDING $Q(t+1)$.
 - WHAT ARE THE FF INPUTS TO ACHIEVE $Q(t) \rightarrow Q(t+1)$
- EXAMPLE :

PS	X		
Q_A	Q_B	0	1
0 0		10	11
0 1		00	10
1 0		11	00
1 1		00	01



SR E-FUNCTION

PS	NS
-	0
0	0 -
1	0 1 - 0

$S(t) R(t)$

EXAMPLE 8.8: DESIGN MODULO-5 COUNTER

- USE T FLIP-FLOPS

Input: $x(t) \in \{0, 1\}$

Output: $z(t) \in \{0, 1, 2, 3, 4\}$

State: $s(t) \in \{S_0, S_1, S_2, S_3, S_4\}$

Initial state: $s(0) = S_0$

Functions: Counts modulo-5, i.e.,
 $(0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0\dots)$,

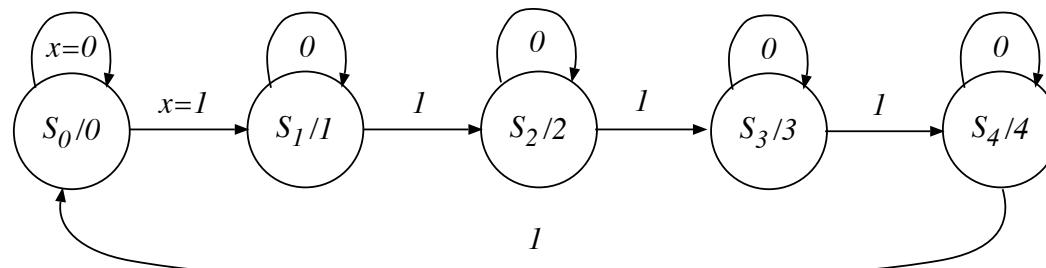


Figure 8.27: STATE DIAGRAM FOR Example 8.8.

EXAMPLE 8.8 (cont.)

z	z_2	z_1	z_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0

PS	Input		Input	
$Q_2Q_1Q_0$	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	000	001
001	001	010	000	011
010	010	011	000	001
011	011	100	000	111
100	100	000	000	100

	NS	$T_2T_1T_0$
--	------	-------------

DON'T CARES: 5, 6, AND 7

EXAMPLE 8.8 (cont.)

sm – STATE MAP

sm.	<u>x</u>	Q ₂	Q ₁	Q ₀
0	0	1	1	
2	2	3	3	
6	6	7	7	
4	4	5	5	

T ₂ .	<u>x</u>	Q ₂	Q ₁	Q ₀
0	0	0	0	
0	0	1	0	
-	-	-	-	
0	1	-	-	

T ₁ .	<u>x</u>	Q ₂	Q ₁	Q ₀
0	0	1	0	
0	0	1	0	
-	-	-	-	
0	0	-	-	

T ₀ .	<u>x</u>	Q ₂	Q ₁	Q ₀
0	1	1	0	
0	1	1	0	
-	-	-	-	
0	0	-	-	

$$T_2 = xQ_2 + xQ_1Q_0$$

$$T_1 = xQ_0$$

$$T_0 = xQ'_2$$

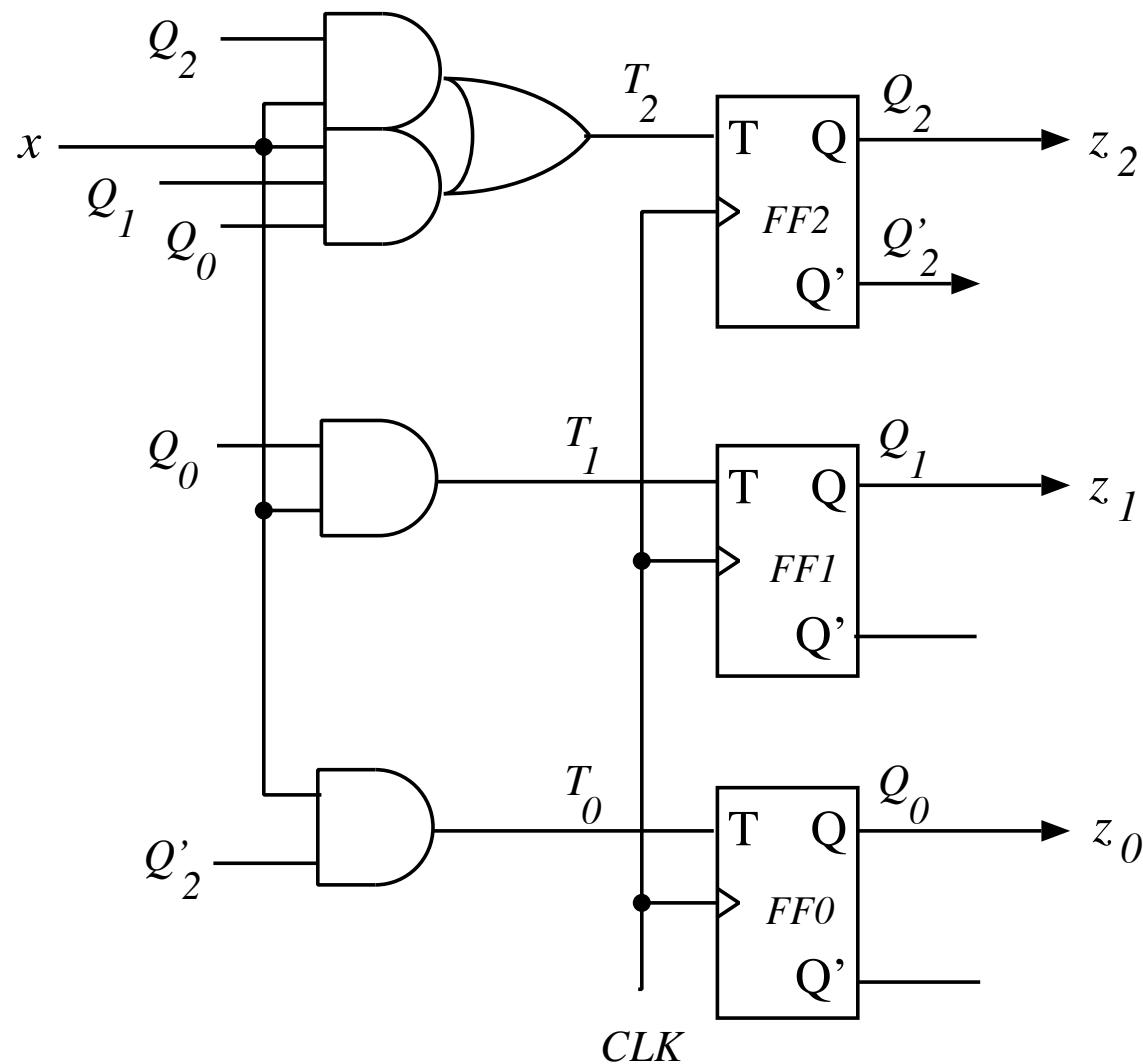


Figure 8.28: SEQUENTIAL NETWORK IN Example 8.8.

EXAMPLE 8.9: DESIGN

Input: $\underline{x}(t) = (x_1, x_0), x_i \in \{0, 1\}$

Output: $z(t) \in \{0, 1\}$

State: $s(t) \in \{a, b, c, d\}$

Initial state: $s(0) = a$

Functions: The transition and output functions

PS		$x_1 x_0$		
		01	10	11
	a	b,0	c,1	c,0
	b	a,0	d,1	d,0
	c	d,0	c,0	a,1
	d	c,0	a,0	d,1
		NS, z		

EXAMPLE 8.9 (CONT.)

State	$Q_1 Q_0$	PS	$x_1 x_0$			$Q(t)$	$Q(t+1)$	S	R
		$Q_1 Q_0$	01	10	11				
a	00	00	01	10	10	0	0	0	-
b	01	01	00	11	11	0	1	1	0
c	10	10	11	10	00	1	0	0	1
d	11	11	10	00	11	1	1	-	0
			NS						

S_1	x_0	Q_0
-	0 1 1	-
-	0 1 1	-
-	- - 0	-
-	- 0 -	-
S_0	$x_0 \quad x_1$	Q_0
-	1 0 0	-
-	0 - -	-
-	0 - 0	-
-	1 0 0	-

R_1	x_0	Q_0
-	- 0 0	-
-	- 0 0	-
-	0 0 1	-
-	0 1 0	-
R_0	$x_0 \quad x_1$	Q_0
-	0 - -	-
-	1 0 0	-
-	1 0 1	-
-	0 - -	-

$$\begin{aligned}
 S_1 &= x_1 Q'_1 \\
 R_1 &= x'_0 Q_1 Q_0 + x_1 x_0 Q_1 Q'_0 \\
 S_0 &= x'_1 Q'_0 \\
 R_0 &= x'_1 Q_0 + x'_0 Q_1
 \end{aligned}$$

x_0				
-	0	0	1	
-	0	0	1	Q_0
-	0	1	0	
-	0	1	0	
				x_1

THE OUTPUT EXPRESSION IS

$$z = x'_0 Q'_1 + x_1 x_0 Q_1$$

EXAMPLE 8.9 (cont.)

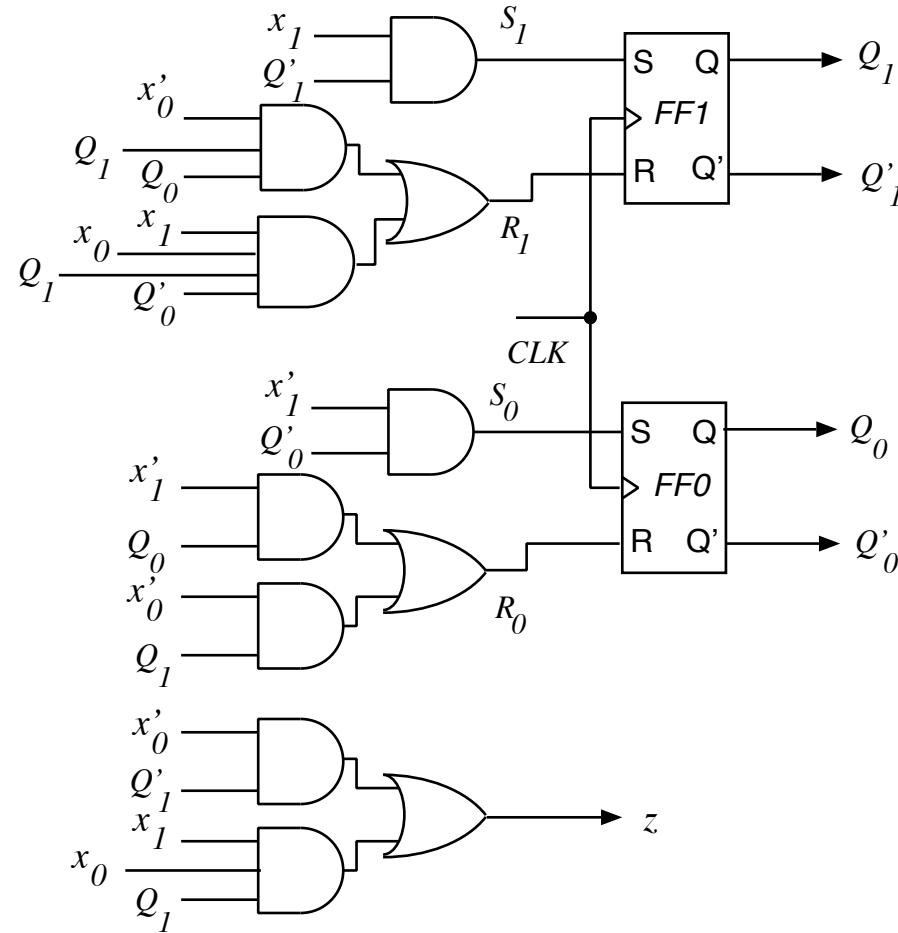


Figure 8.29: SEQUENTIAL NETWORK IN Example 8.9.

SPECIAL STATE ASSIGNMENTS

- ONE FLIP-FLOP PER STATE

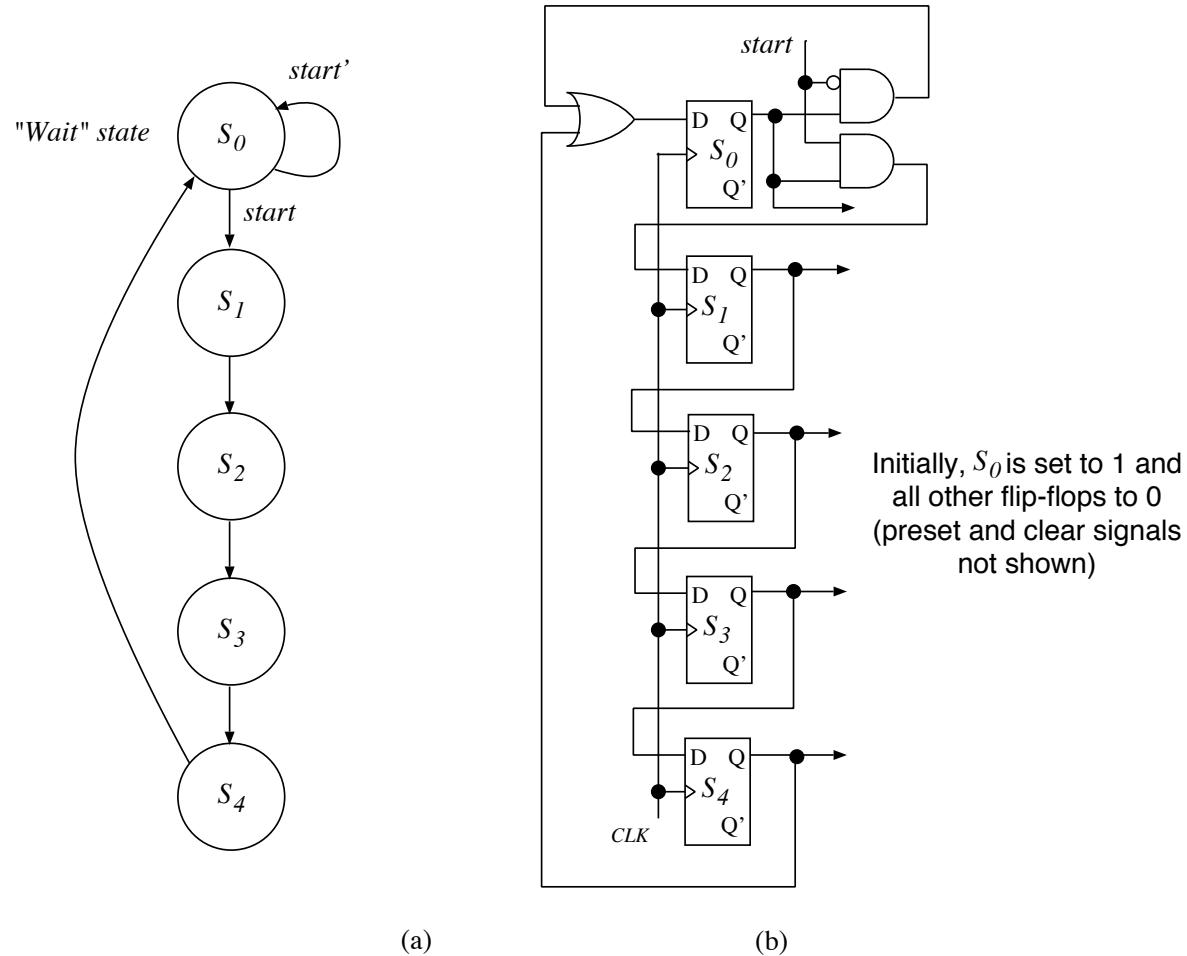
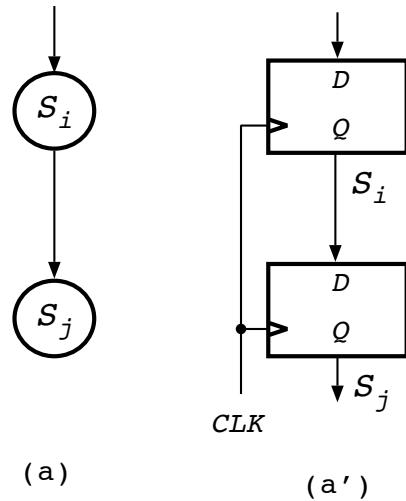


Figure 8.30: ONE FLIP-FLOP PER STATE APPROACH: a) STATE DIAGRAM. b) IMPLEMENTATION (Outputs omitted).

PRIMITIVES FOR “one-flip-flop-per-state” APPROACH



Determined by
inputs

Predecessor States

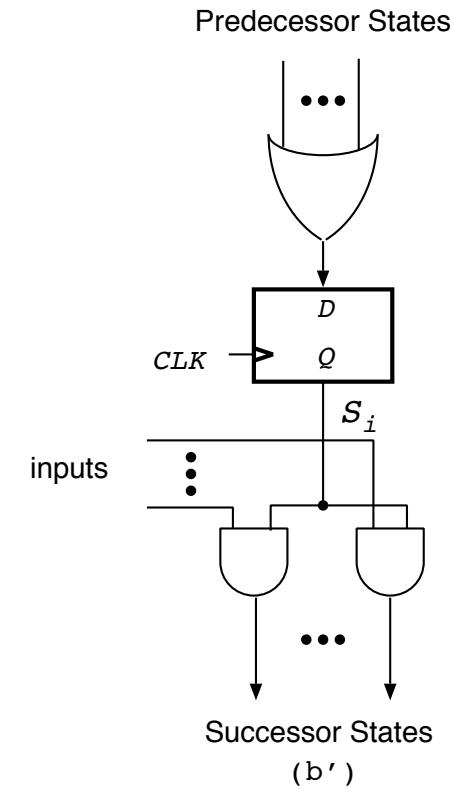
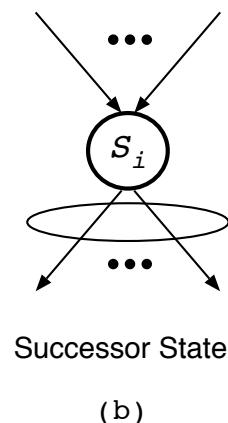
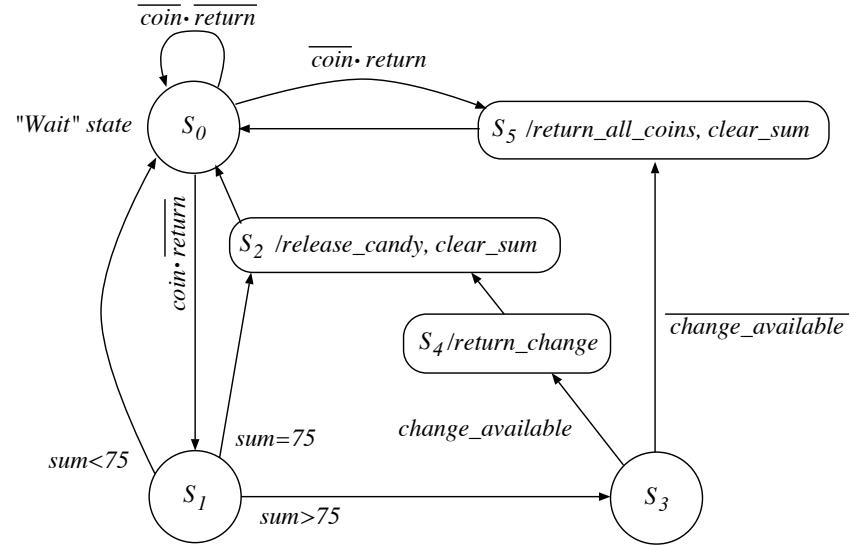
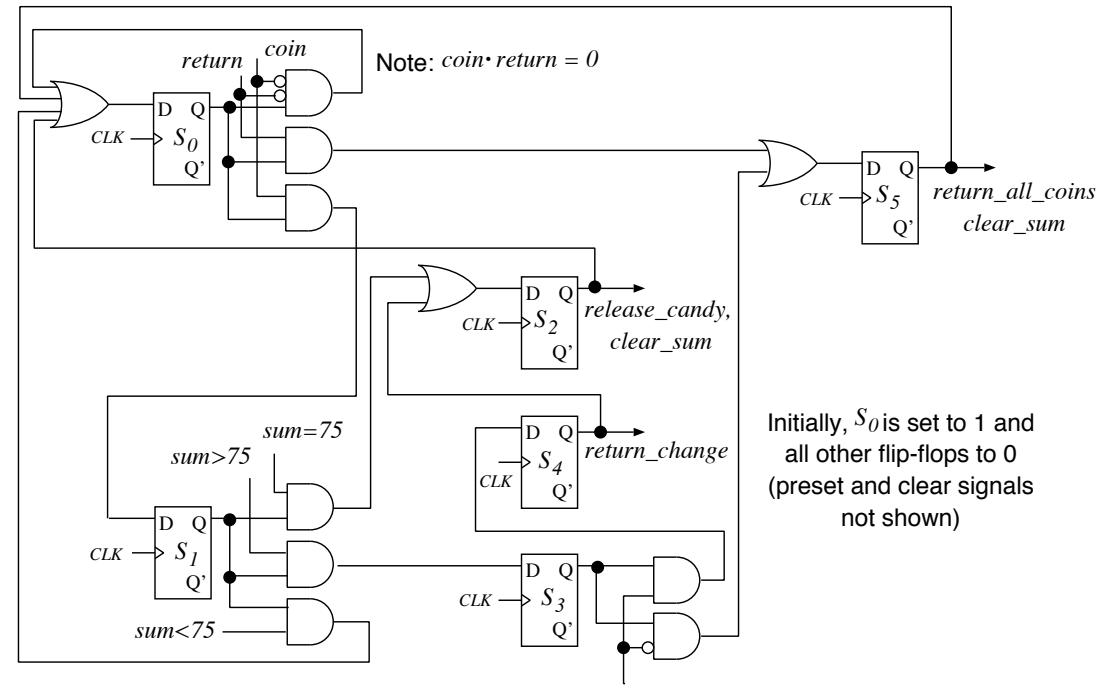


Figure 8.31: PRIMITIVES FOR THE “ONE-FLIP-FLOP-PER-STATE” APPROACH.

CONTROLLER FOR SIMPLE VENDING MACHINE



(a)



(b)

Figure 8.32: A ONE-FLIP-FLOP-PER-STATE IMPLEMENTATION OF A CONTROLLER FOR VENDING MACHINE: a) STATE DIAGRAM. b) IMPLEMENTATION.

SHIFTING STATE REGISTER: Example 8.10

INPUT: $x(t) \in \{0, 1\}$
 OUTPUT: $z(t) \in \{0, 1\}$

FUNCTION: $z(t) = \begin{cases} 1 & \text{if } x(t-3, t) = 1101 \\ 0 & \text{otherwise} \end{cases}$

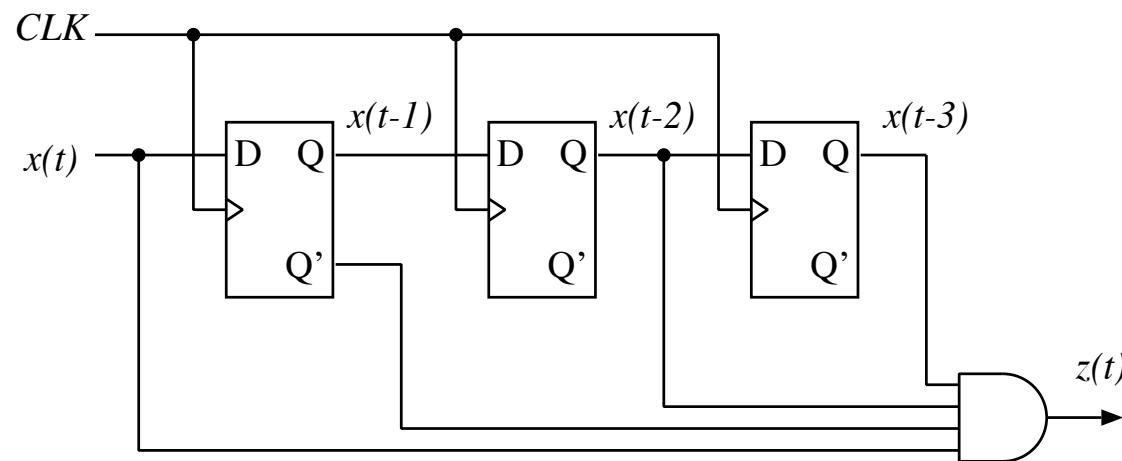
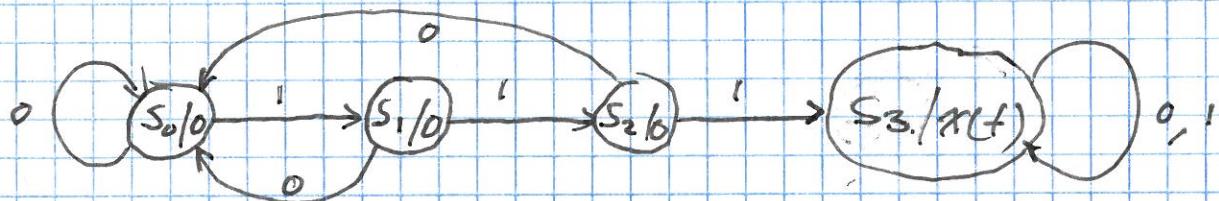


Figure 8.33: IMPLEMENTATION OF PATTERN RECOGNIZER IN EXAMPLE 8.10.

EXTRA EXAMPLE:

$$z(t) = \begin{cases} 0 & \text{UNTIL THE FIRST INSTANCE OF} \\ & \text{3 CONSECUTIVE 1'S HAS BEEN} \\ & \text{RECEIVED} \\ x(t) & \text{AFTERWARDS} \end{cases}$$

x	0	0	1	1	0	1	1	1	0	0	1	...		
z	0	0	0	0	0	0	0	0	1	1	0	0	1	...



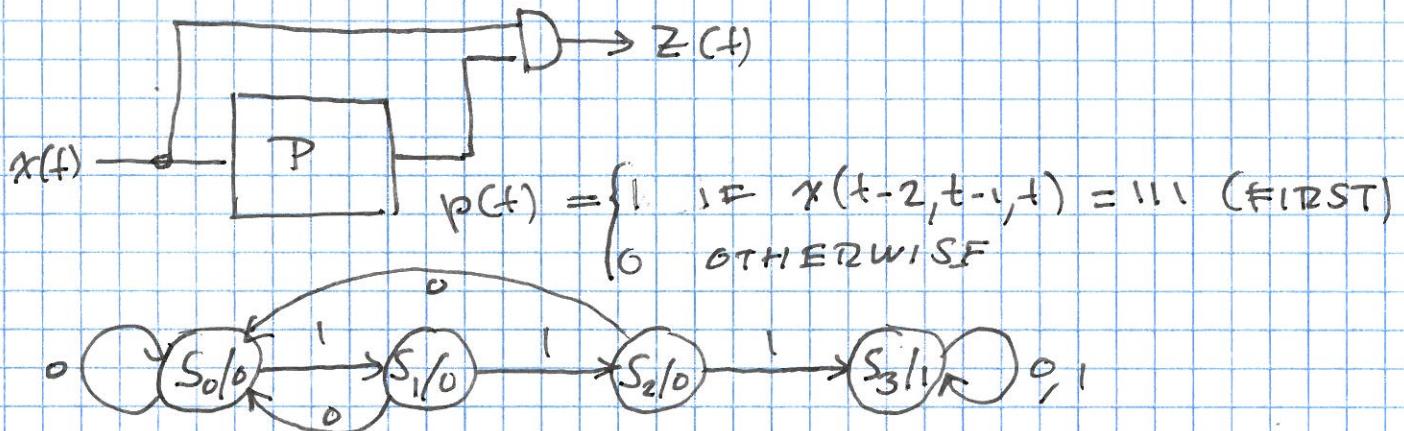
PS	$x(t)$	z
$y_1 y_0$	0	1
S_0	00	00
S_1	01	00
S_2	10	10
S_3	11	11

y_1 , y_0
NS

$$y_1: \begin{array}{|c|c|c|c|} \hline & x & & \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline \end{array} \quad y_1 = y_1 x + y_0 x + y_0 y_0$$

$$y_0: \begin{array}{|c|c|c|c|} \hline & x & & \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline \end{array} \quad y_0 = y_0 x + y_1 y_0$$

ALTERNATIVE DESIGN



USE FINITE-MEMORY DESIGN

