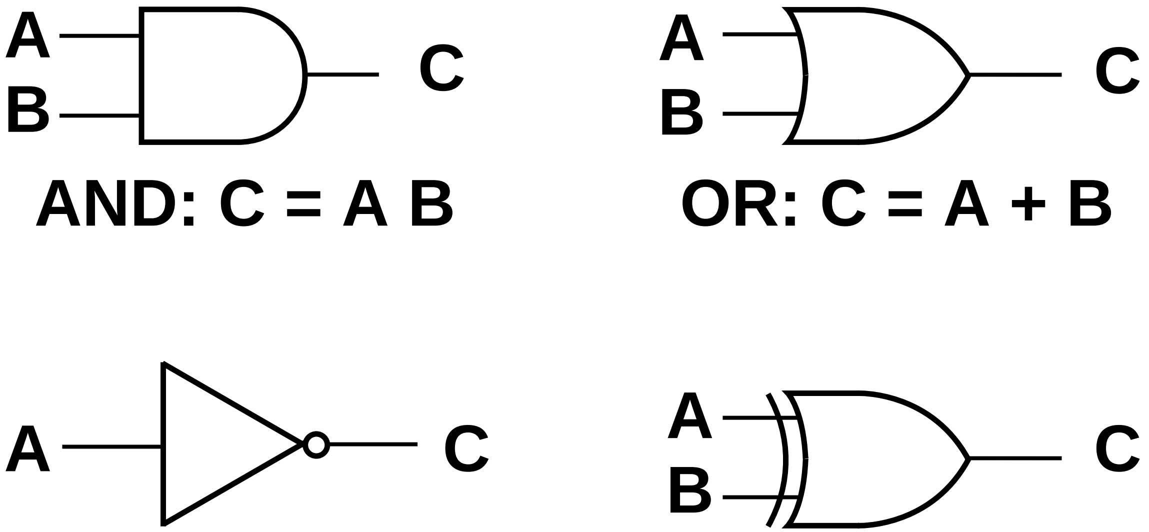
Figure 1-1 Basic Gates



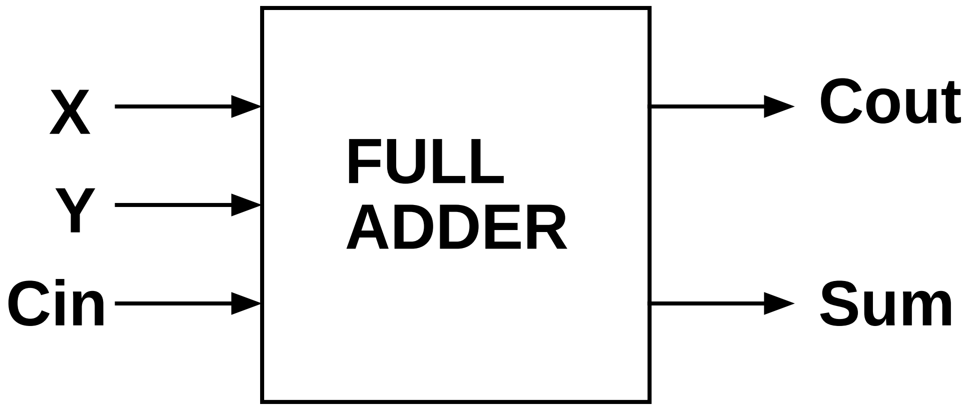
AND: $C = A\,B$

OR: $C = A + B$

NOT: $C = A'$

EXCLUSIVE OR: $C = A \oplus B$

Figure 1-2 Full Adder



(a) Full adder module

X	Y	Cin	Cout	Sum
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

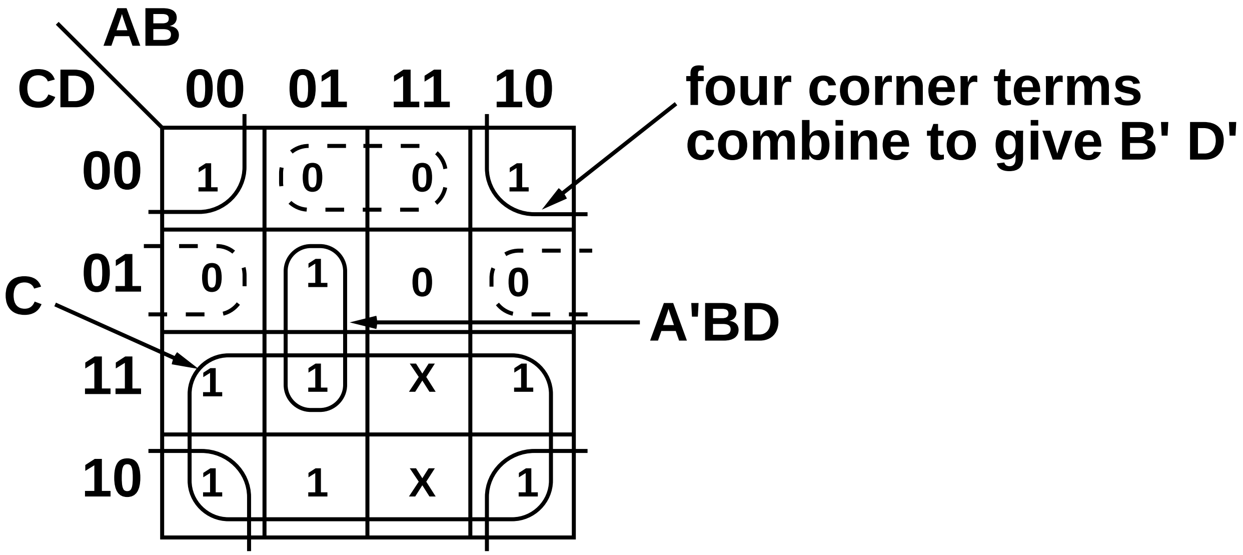
(b) Truth Table

$$\text{Sum} = X'Y'\text{Cin} + X'Y\text{Cin}' + XY'\text{Cin}' + XY\text{Cin} = X \oplus Y \oplus \text{Cin}$$

$$\text{Cout} = X'Y\text{Cin} + XY'\text{Cin} + XY\text{Cin}' + XY\text{Cin} = XY + X\text{Cin} + Y\text{Cin}$$

Figure 1-3 Four-Variable Karnaugh Maps

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10



CD \ AB	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	1	X	1
10	1	1	X	1

four corner terms combine to give B' D'

C

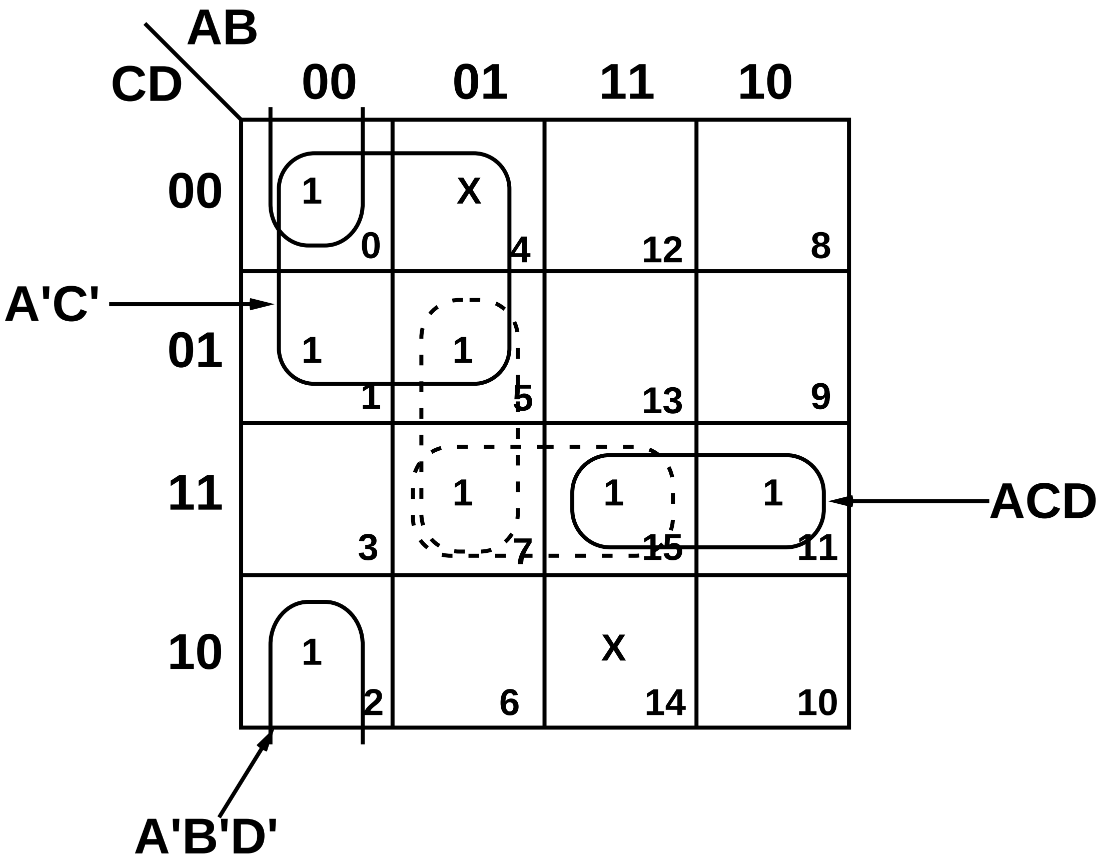
A'BD

$F = \sum m(0,2,3,5,6,7,8,10,11) + \sum d(14,15)$

$= C + B'D' + A'BD$

$= (B' + C + D)(B + C + D')(A' + B')$

Figure 1-4 Selection of Prime Implicants



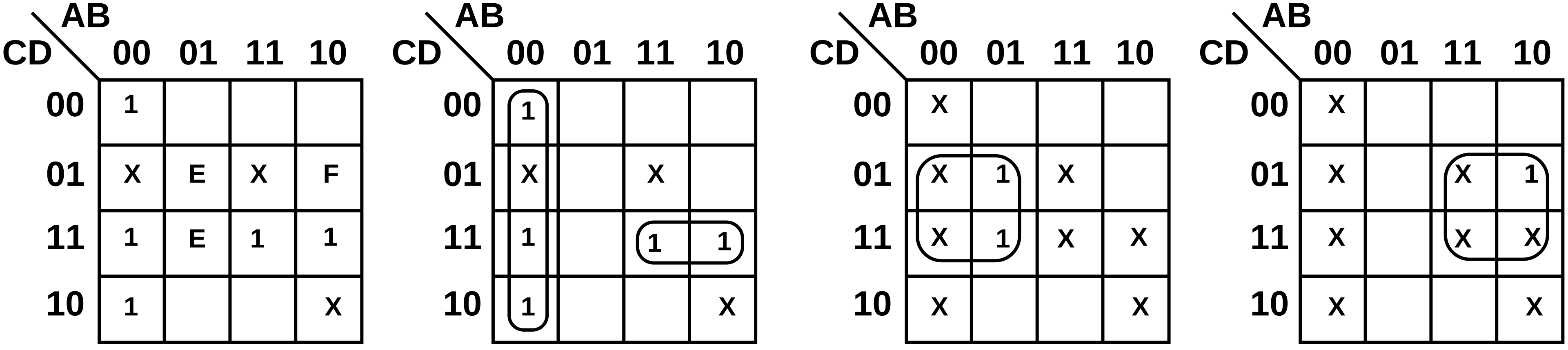
CD \ AB	00	01	11	10
00	1 0	X 4	12	8
01	1 1	1 5	13	9
11	3	1 7	1 15	1 11
10	1 2	6	X 14	10

F = A'C' + A'B'D' + ACD + A'BD

or

F = A'C' + A'B'D' + ACD + BCD

Figure 1-5
Simplification Using Map-Entered Variables



CD \ AB	00	01	11	10
00	1			
01	X	E	X	F
11	1	E	1	1
10	1			X

G

CD \ AB	00	01	11	10
00	1			
01	X		X	
11	1		1	1
10	1			X

E = F = 0

$MS_0 = A'B' + ACD$

CD \ AB	00	01	11	10
00	X			
01	X	1	X	
11	X	1	X	X
10	X			X

E = 1, F = 0

$MS_1 = A'D$

CD \ AB	00	01	11	10
00	X			
01	X		X	1
11	X		X	X
10	X			X

E = 0, F = 1

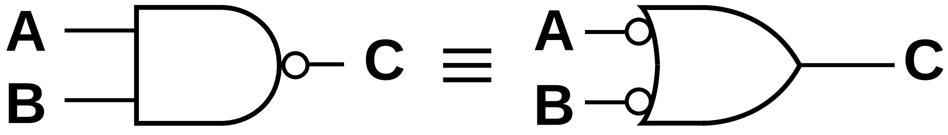
$MS_2 = AD$

$$G = MS_0 + EMS_1 + FMS_2$$

$$= A'B' + ACD + EA'D + FAD$$

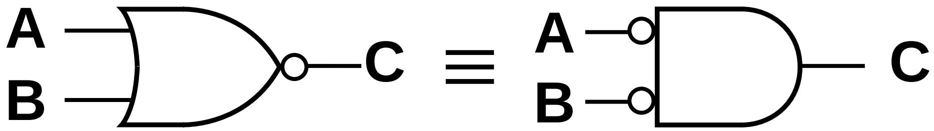
Figure 1-6 NAND and NOR Gates

NAND:



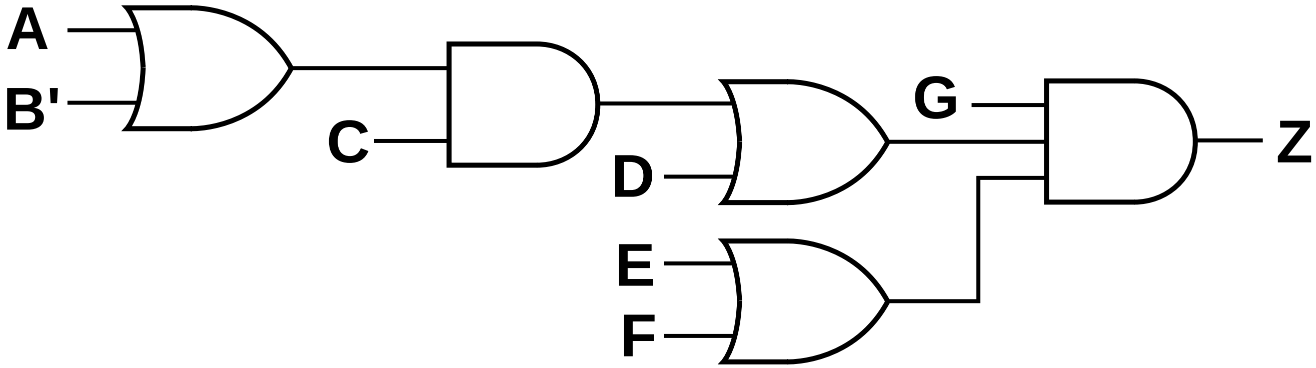
$C = (AB)' = A' + B'$

NOR:

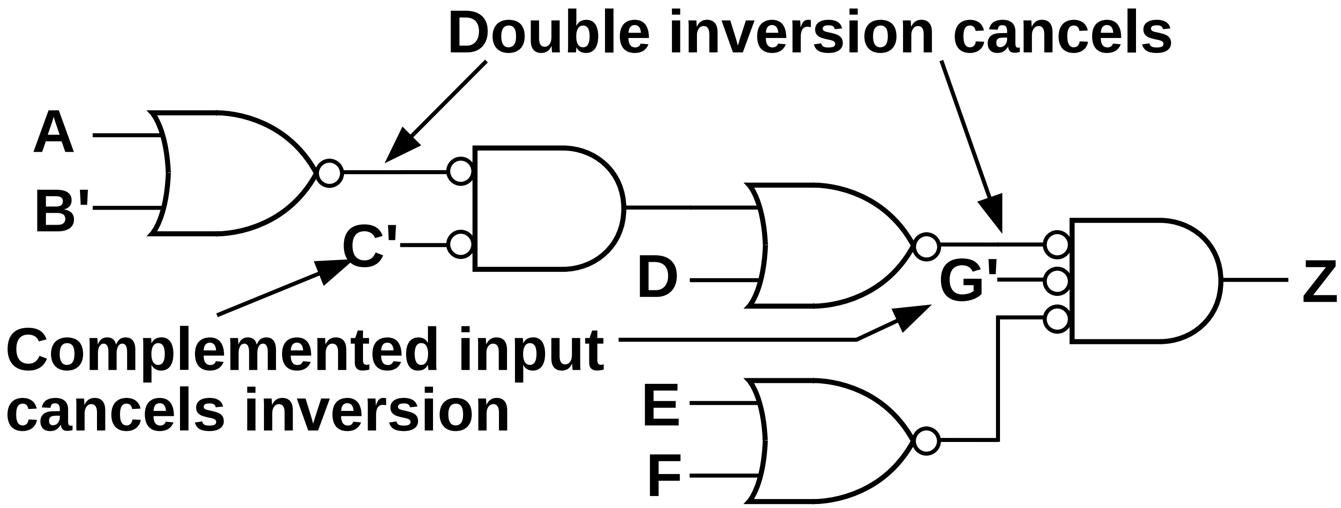


$C = (A+B)' = A'B'$

Figure 1-7 Conversion to NOR Gates

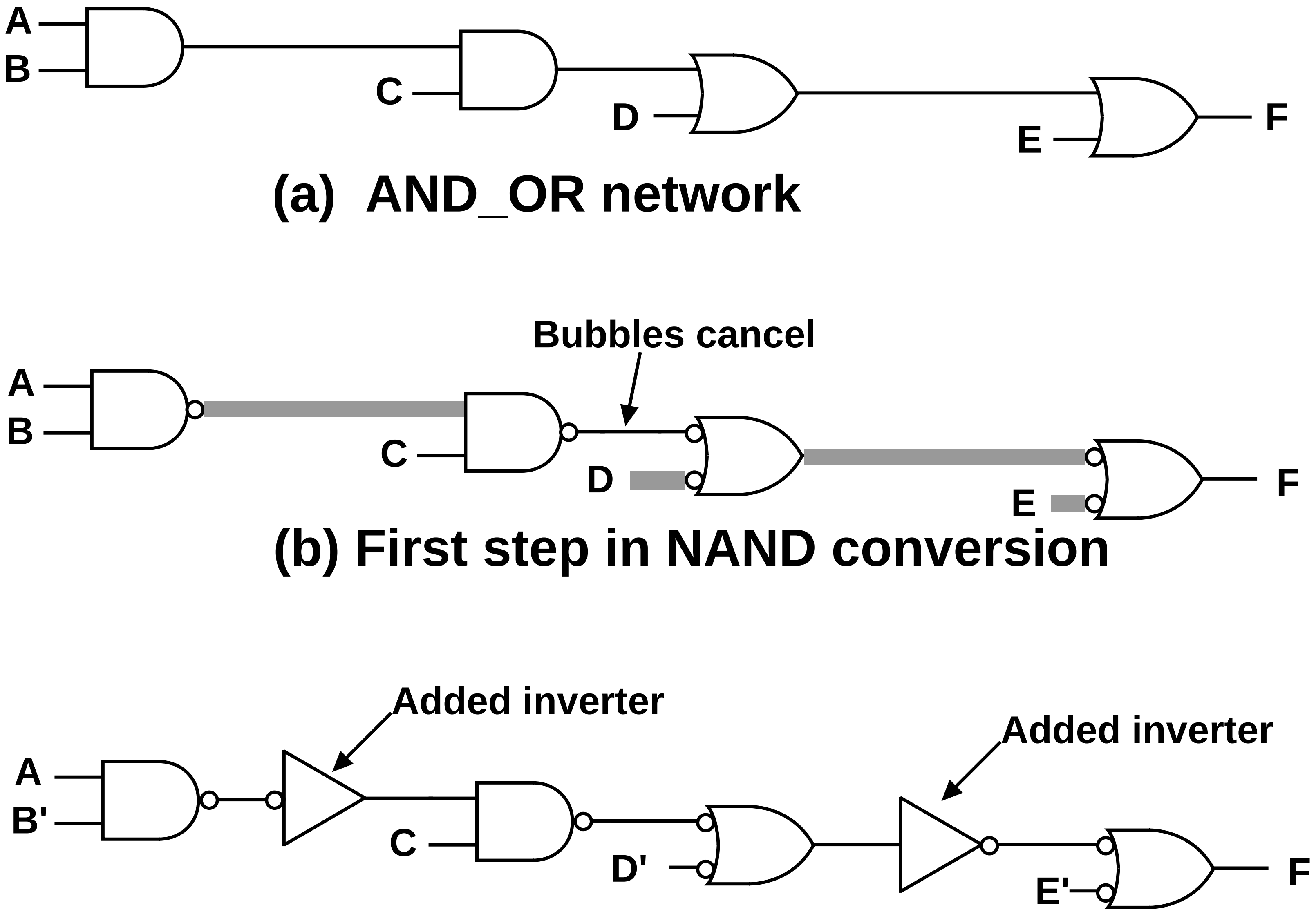


(a) AND-OR network



(b) Equivalent NOR-gate network

Figure 1-8
Conversion of AND-OR Network to NAND Gates



(a) AND_OR network

Bubbles cancel

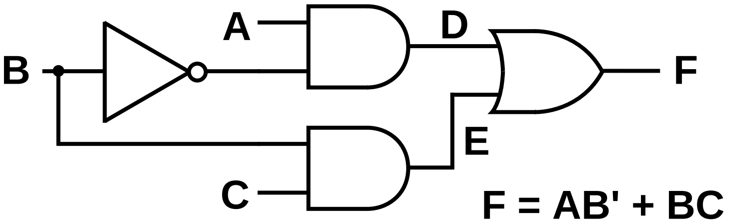
(b) First step in NAND conversion

Added inverter

Added inverter

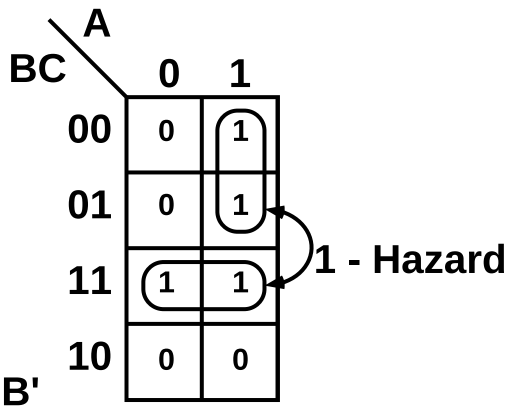
(c) Completed conversion

Figure 1-9 Elimination of 1-Hazard

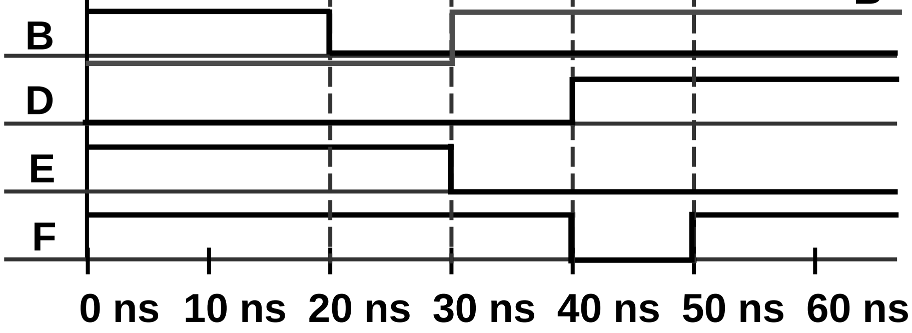


F = AB' + BC

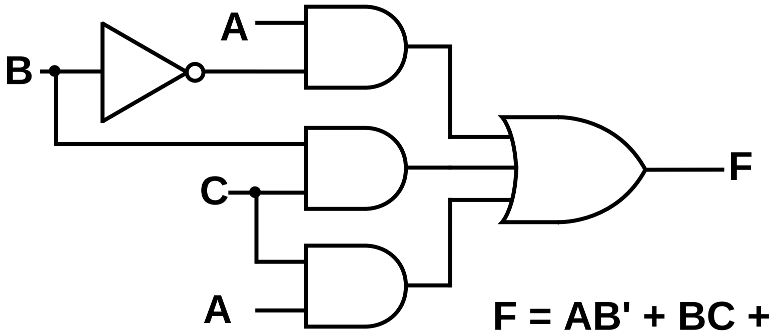
(a) Network with 1-hazard



BC \ A	0	1
00	0	1
01	0	1
11	1	1
10	0	0

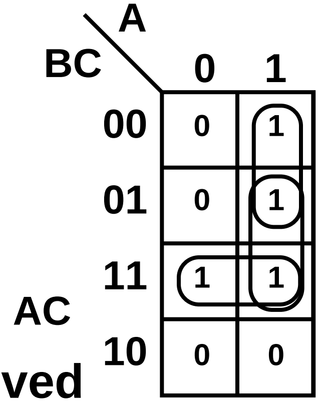


(b) Timing Chart



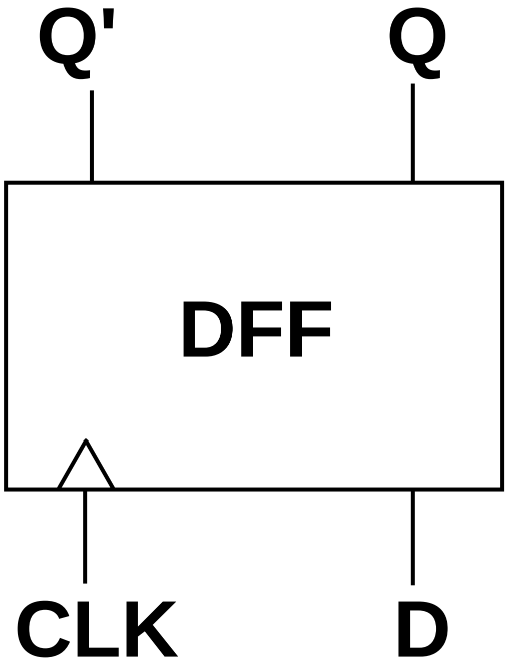
F = AB' + BC + AC

(c) Network with hazard removed



BC \ A	0	1
00	0	1
01	0	1
11	1	1
10	0	0

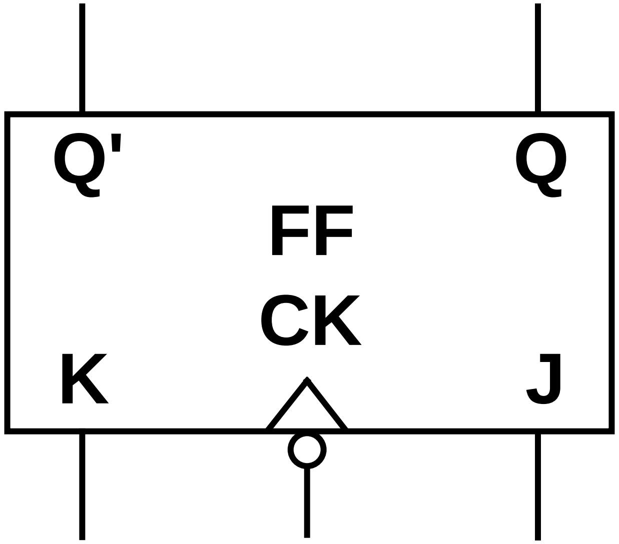
Figure 1-10
Clocked D Flip-flop
with Rising-edge Trigger



D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1

$$Q^+ = D$$

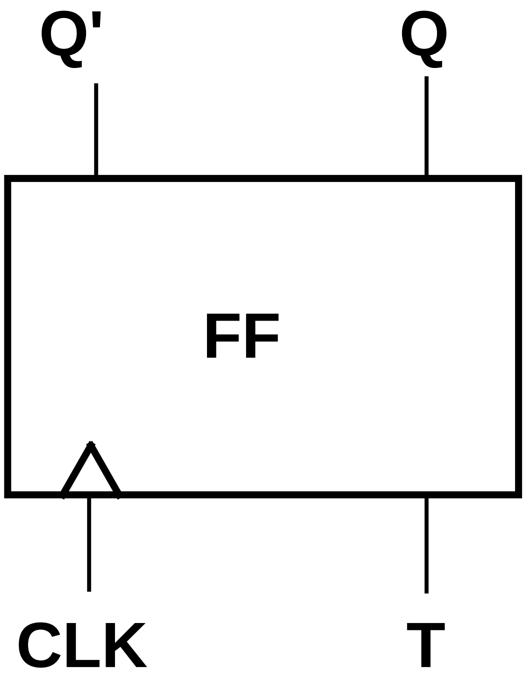
Figure 1-11 Clocked J-K Flip-flop



J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$$Q^+ = JQ' + K'Q$$

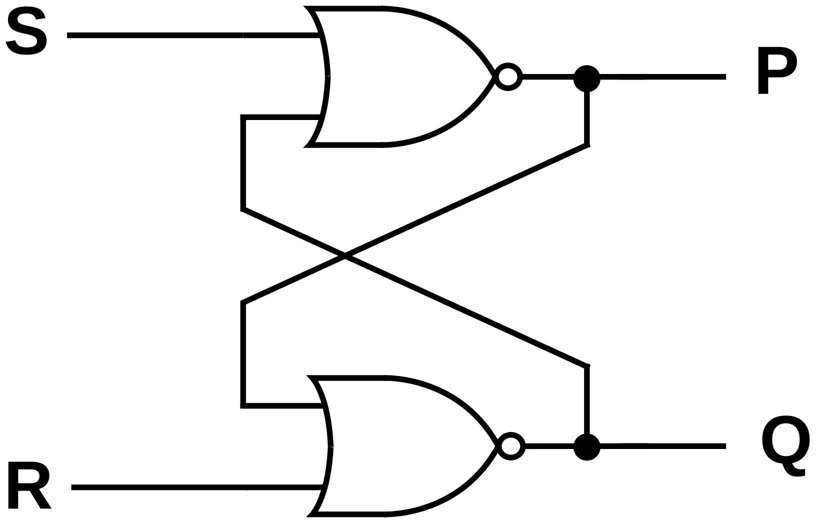
Figure 1-12 Clocked T Flip-flop



T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

$$Q^+ = QT' + Q'T = Q \oplus T$$

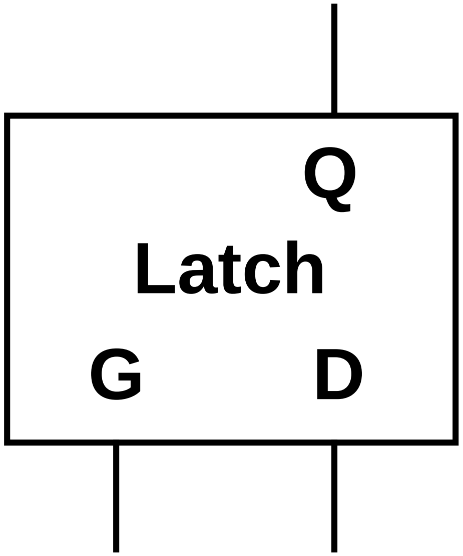
Figure 1-13 S-R Latch



S	R	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	–
1	1	1	–

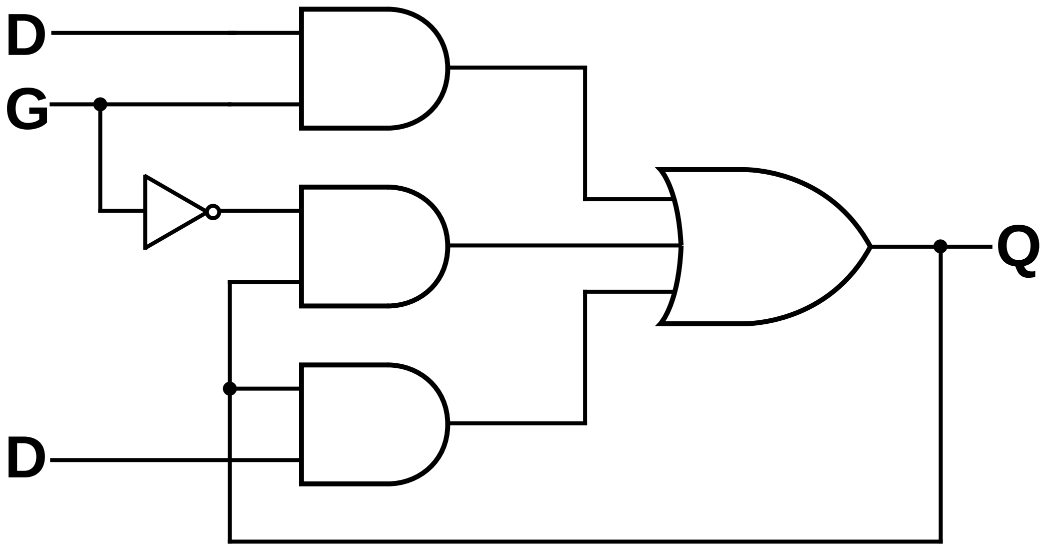
$$Q^+ = S + R'Q$$

Figure 1-14 Transparent D Latch



G	D	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 1-15 Implementation of D Latch



$Q^{+} = DG + G'Q + (DQ)$

Figure 1-16 General Model of Mealy Sequential Machine

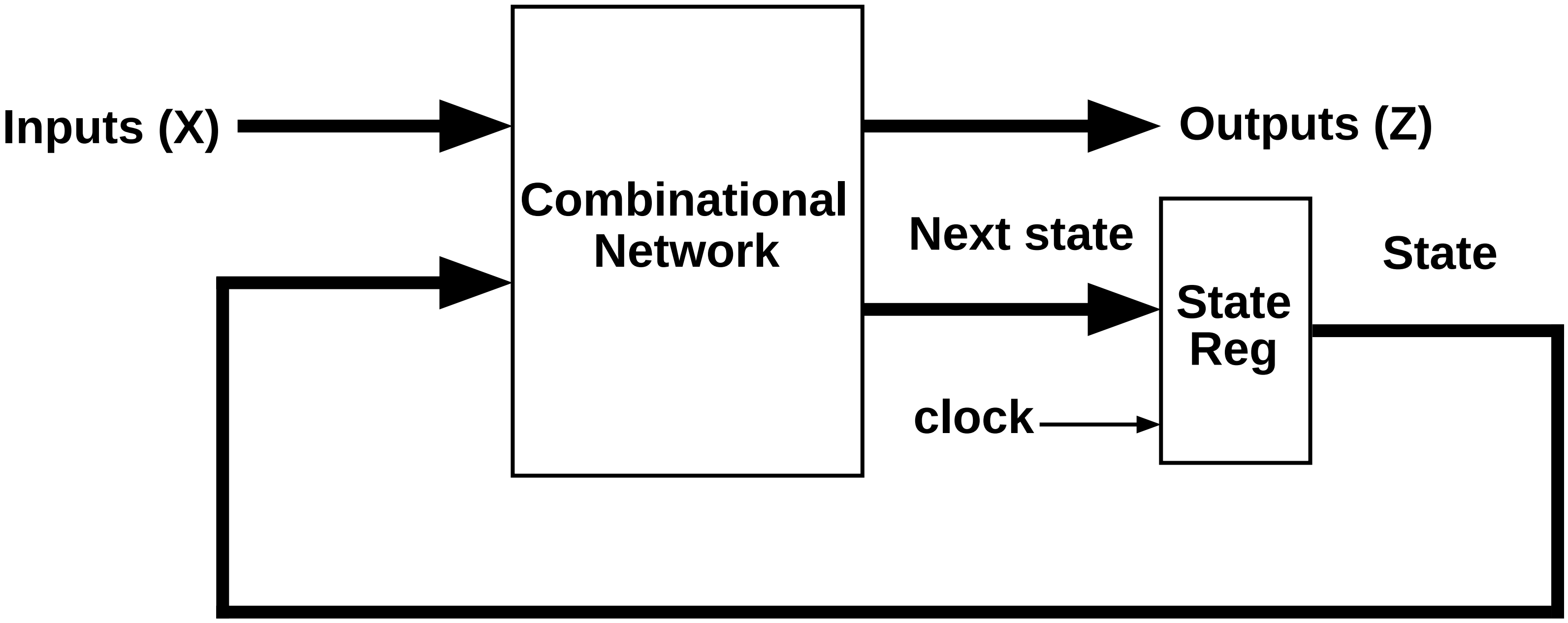
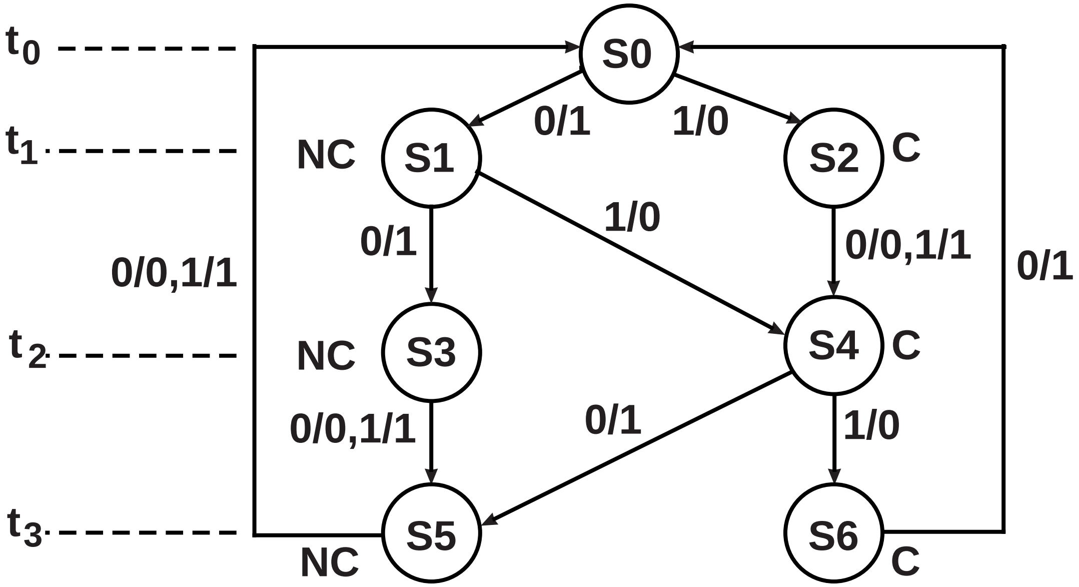


Figure 1-17 State Graph and Table for Code Converter



(a) Mealy state graph

	NS		Z	
PS	X = 0	X = 1	X = 0	X = 1
S0	S1	S2	1	0
S1	S3	S4	1	0
S2	S4	S4	0	1
S3	S5	S5	0	1
S4	S5	S6	1	0
S5	S0	S0	0	1
S6	S0	–	1	–

(b) State Table

From Page 20

I. States which have the same next state (NS) for a given input should be given adjacent assignments (look at the columns of the state table).

II. States which are the next states of the same state should be given adjacent assignments (look at the rows).

III. States which have the same output for a given input should be given adjacent assignments.

I. (1,2) (3,4) (5,6) (in the X=1 column, S_1 and S_2 both have NS S_4;
in the X=0 column, S_3 & S_4 have NS S_5, and S_5 & S_6 have NS S_0)

II. (1,2) (3,4) (5,6)   (S_1 & S_2 are NS of S_0; S_3 & S_4 are NS of S_1;
and S_5 & S_6 are NS of S_4)

III. (0,1,4,6) (2,3,5)

Q2 Q3 \ Q1	0	1
00	S0	S1
01		S2
11	S5	S3
10	S6	S4

Figure 1-18(a)
State Assignment Map

Figure 1-17(b) State Table

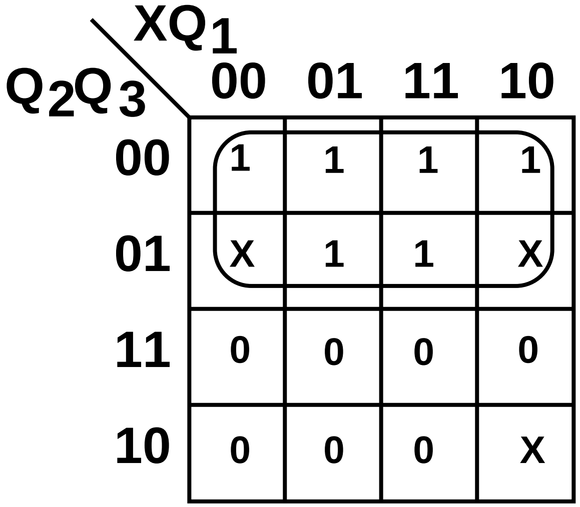
	NS		Z	
PS	X=0	X=1	X=0	X=1
S0	S1	S2	1	0
S1	S3	S4	1	0
S2	S4	S4	0	1
S3	S5	S5	0	1
S4	S5	S6	1	0
S5	S0	S0	0	1
S6	S0	–	1	–

Figure 1-18(b) Transition Table

	$Q1^+Q2^+Q3^+$		Z	
Q1Q2Q3	X=0	X=1	X=0	X=1
000	100	101	1	0
100	111	110	1	0
101	110	110	0	1
111	011	011	0	1
110	011	010	1	0
011	000	000	0	1
010	000	xxx	1	x
001	xxx	xxx	x	x

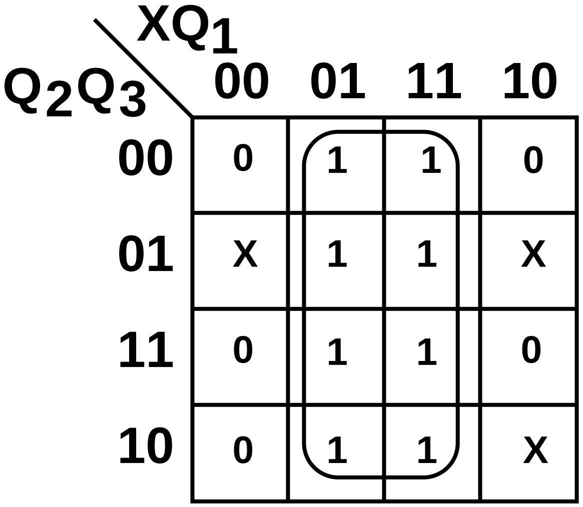
S0 = 000, S1 = 100, S2 = 101, S3 = 111, S4 = 110, S5 = 011, S6 = 010

Figure 1-19
Karnaugh Maps for Figure 1-17



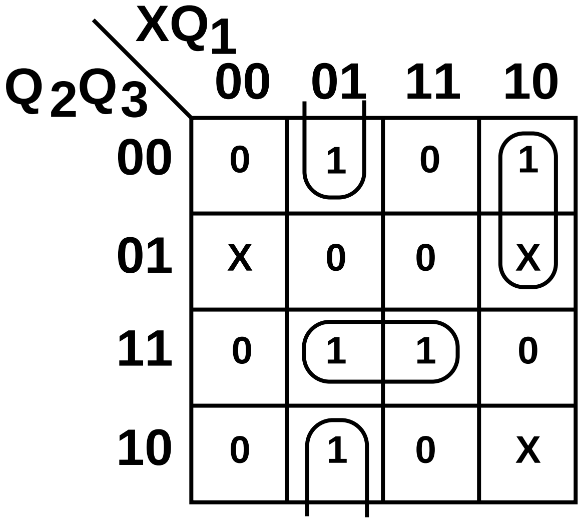
	XQ_1			
Q_2Q_3	00	01	11	10
00	1	1	1	1
01	X	1	1	X
11	0	0	0	0
10	0	0	0	X

$D_1 = Q_1^+ = Q_2'$



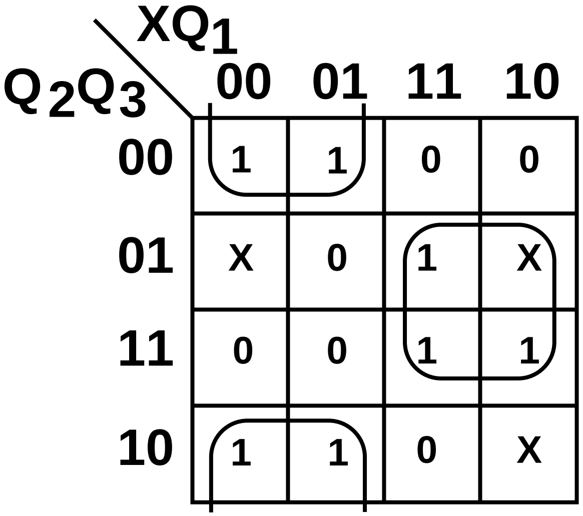
	XQ_1			
Q_2Q_3	00	01	11	10
00	0	1	1	0
01	X	1	1	X
11	0	1	1	0
10	0	1	1	X

$D_2 = Q_2^+ = Q_1$



	XQ_1			
Q_2Q_3	00	01	11	10
00	0	1	0	1
01	X	0	0	X
11	0	1	1	0
10	0	1	0	X

$D_3 = Q_3^+ = Q_1Q_2Q_3 + X'Q_1Q_3' + XQ_1'Q_2'$



	XQ_1			
Q_2Q_3	00	01	11	10
00	1	1	0	0
01	X	0	1	X
11	0	0	1	1
10	1	1	0	X

$Z = X'Q_3' + XQ_3$

Figure 1-20   Realization of Code Converter

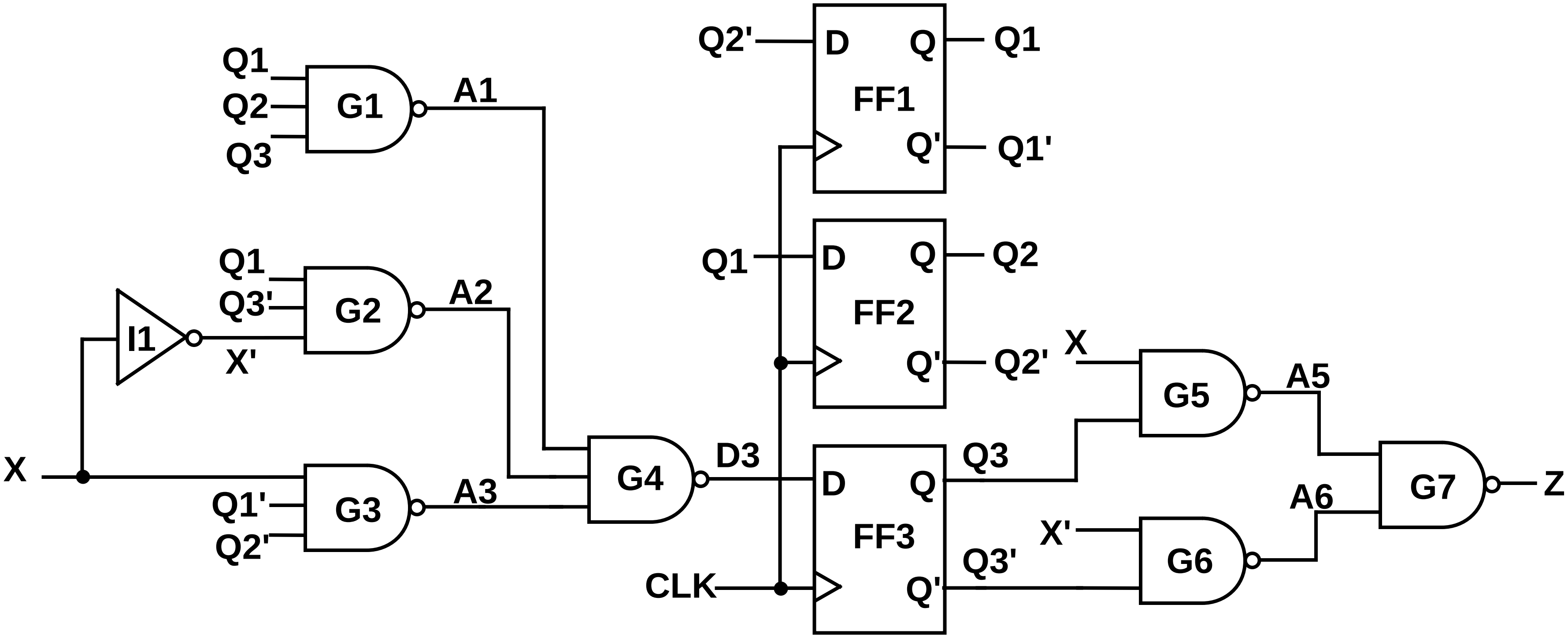
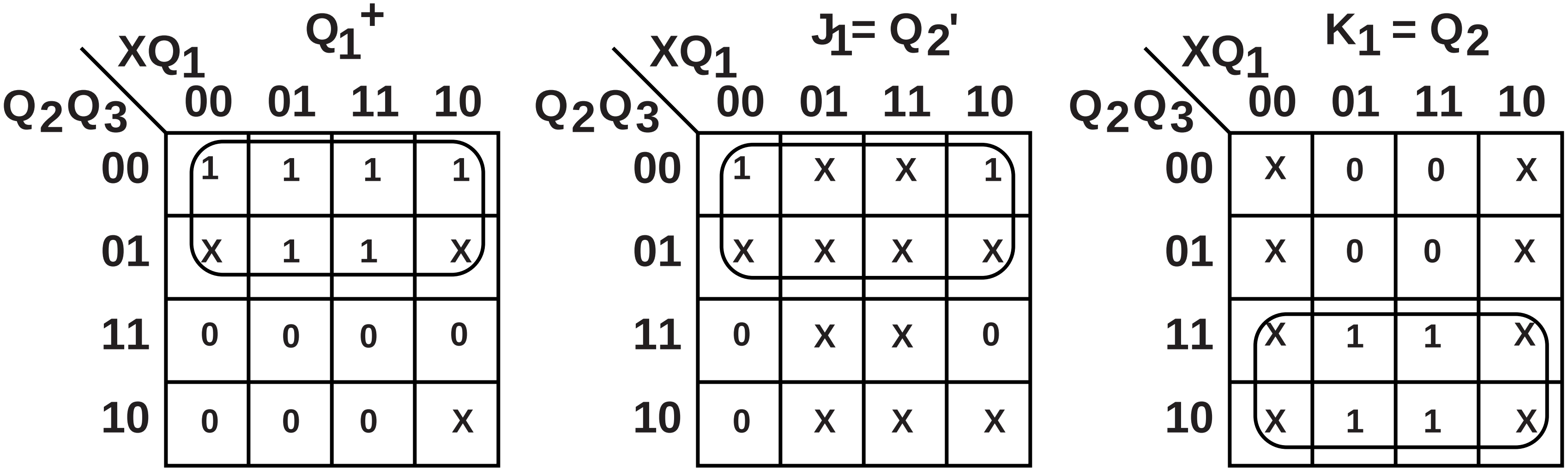


Figure 1-21 Derivation of J-K Input Equations



Q_1^+

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	1	1	1	1
01	X	1	1	X
11	0	0	0	0
10	0	0	0	X

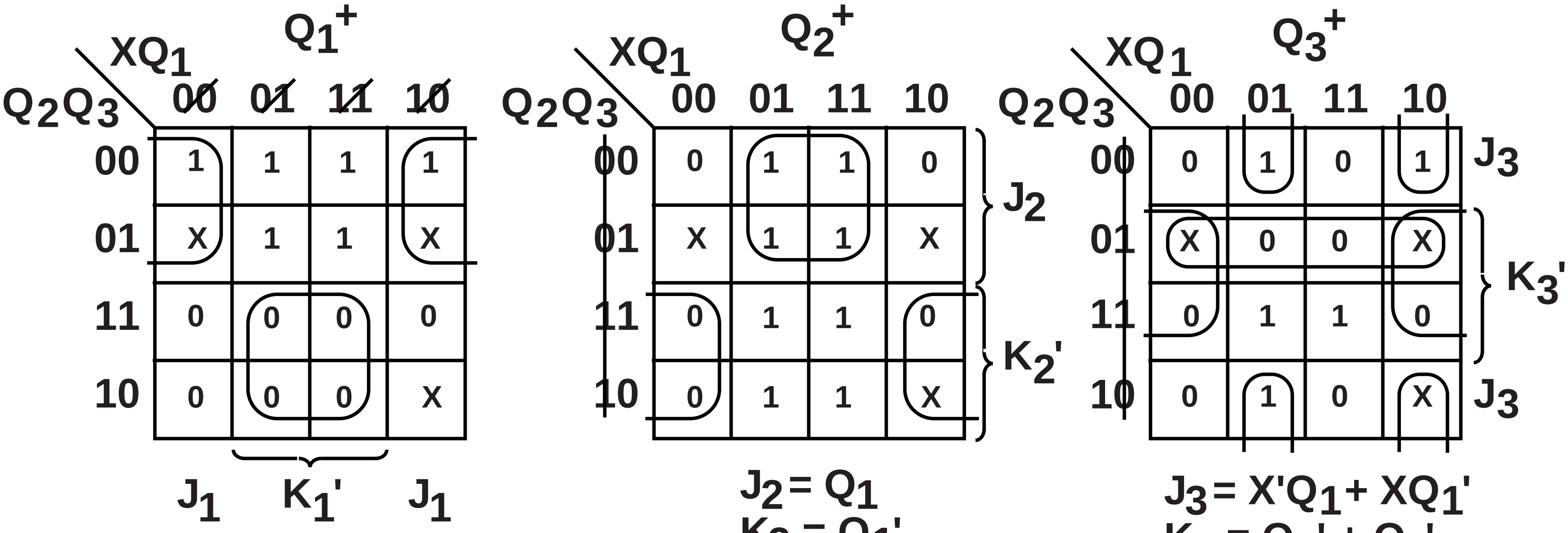
$J_1 = Q_2'$

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	1	X	X	1
01	X	X	X	X
11	0	X	X	0
10	0	X	X	X

$K_1 = Q_2$

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	X	0	0	X
01	X	0	0	X
11	X	1	1	X
10	X	1	1	X

(a) Derivation using separate J-K map



Q_1^+

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	1	1	1	1
01	X	1	1	X
11	0	0	0	0
10	0	0	0	X

$J_1 = Q_2'$ $K_1 = Q_2$

Q_2^+

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	0	1	1	0
01	X	1	1	X
11	0	1	1	0
10	0	1	1	X

$J_2 = Q_1$

$K_2 = Q_1'$

Q_3^+

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	0	1	0	1
01	X	0	0	X
11	0	1	1	0
10	0	1	0	X

$J_3 = X'Q_1 + XQ_1'$

$K_3 = Q_1' + Q_2'$

(b) Derivation using the shortcut method

Figure 1-22
Coding Schemes for Serial Data Transmission

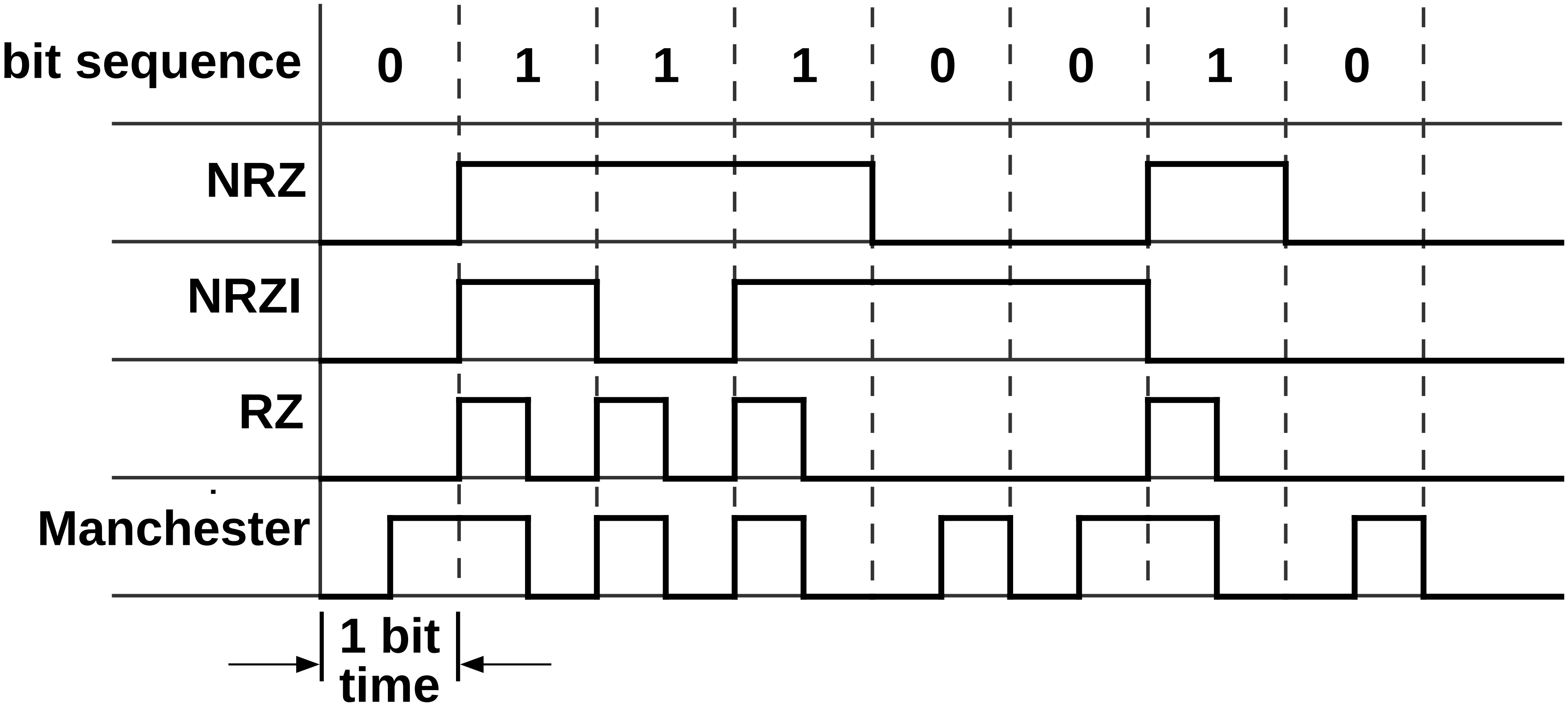
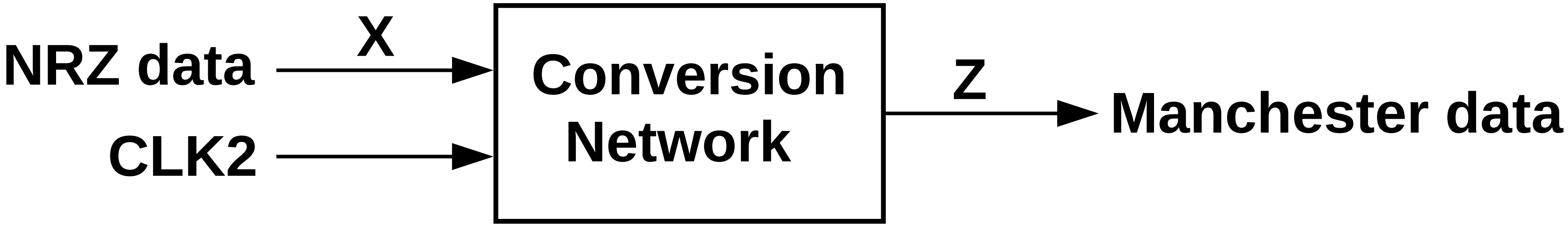


Figure 1-23 Moore network for NRZ-to-Manchester Conversion



NRZ data

X

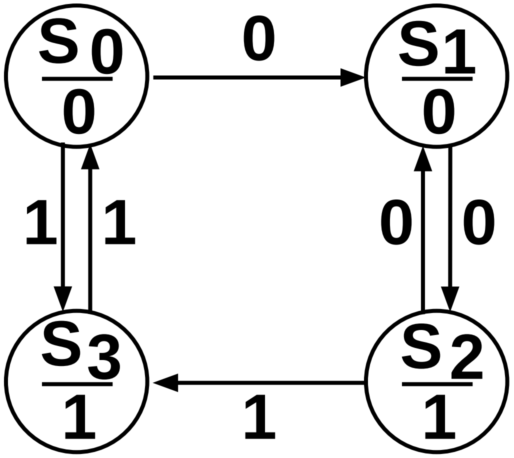
CLK2

Conversion Network

Z

Manchester data

(a) Conversion network



(b) State Graph

Present State	Next State		Present Output (Z)
	X = 0	X = 1	
S_0	S_1	S_3	0
S_1	S_2	–	0
S_2	S_1	S_3	1
S_3	–	S_0	1

(c) State table

Figure 1-24 Timing for Moore Network

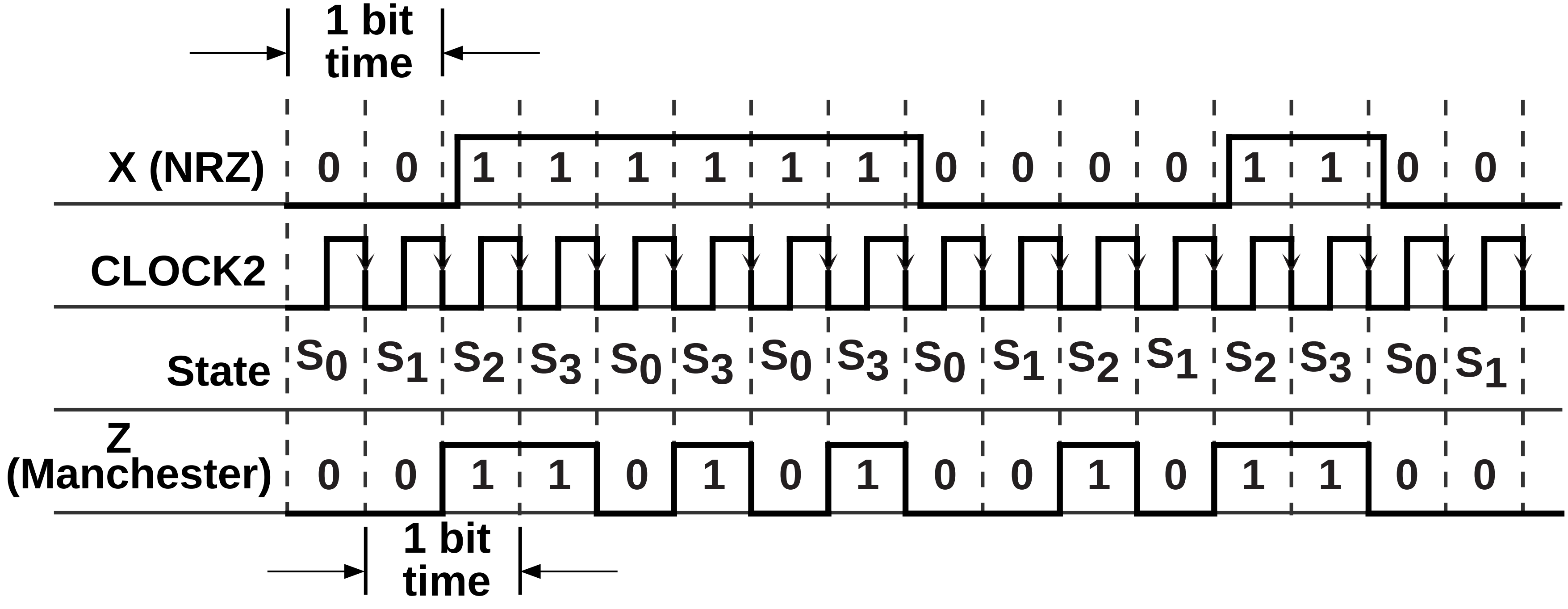
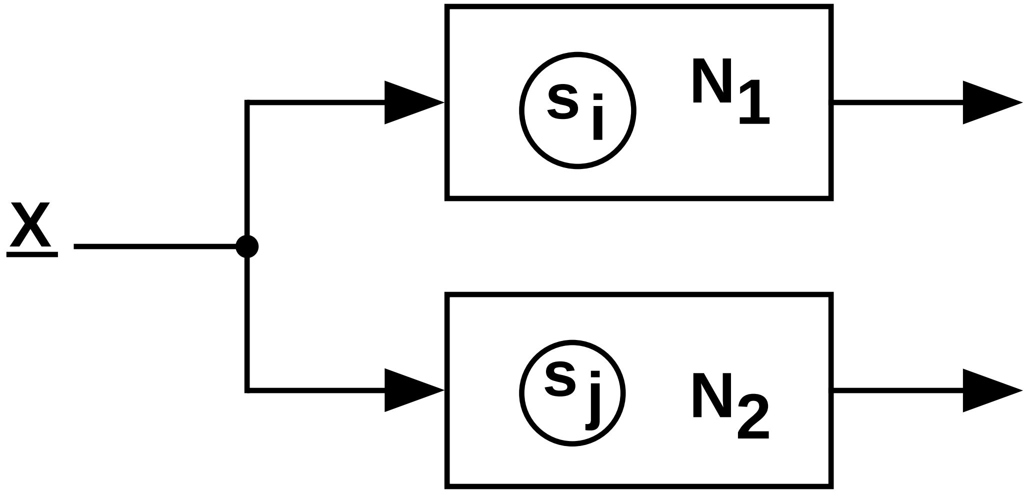


Figure 1-25 Determination of Equivalent States



$\underline{Z}_1 = \lambda_1(s_i, \underline{X})$

$\underline{Z}_2 = \lambda_2(s_j, \underline{X})$

$s_i \equiv s_j$ iff $\underline{Z}_1 = \underline{Z}_2$

for every input sequence $\underline{X}$

Figure 1-26(i) State Table Reduction

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

b	c - d e - f					
c	f - g	a - d e - g				
d	b - c f - g	e - g	a - b			
e	X	X	X	X		
f	X	X	X	X	a - b	
g	X	X	X	X	c - e b - g	e - f a - g
	a	b	c	d	e	f

a ≡ b iff c ≡ d and e ≡ f

Figure 1-26(ii) State Table Reduction

	a	b	c	d	e	f
b	c - d e - f					
c	f - g	a - d e - g				
d	b - c f - g	e - g	a - b			
e	☒	☒	☒	☒		
f	☒	☒	☒	☒	a - b	
g	☒	☒	☒	☒	c - e b - g	e - f a - g

a ≡ b iff c ≡ d and e ≡ f

	a	b	c	d	e	f
b	c - d e - f					
c	~~f - g~~	~~a - d~~ ~~e - g~~				
d	~~b - c~~ ~~f - g~~	~~e - g~~	a - b			
e	☒	☒	☒	☒		
f	☒	☒	☒	☒	a - b	
g	☒	☒	☒	☒	~~c - e~~ ~~b - g~~	~~e - f~~ ~~a - g~~

a º b, c º d, e º f

Figure 1-26(iii) State Table Reduction

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

a ≡ b, c ≡ d, e ≡ f

Present State	X = 0	1	X = 0	1
a	c	e	0	0
c	a	g	0	0
e	e	a	0	1
g	c	g	0	1

Final Reduced Table

Figure 1-27 Timing Diagram for Code Converter

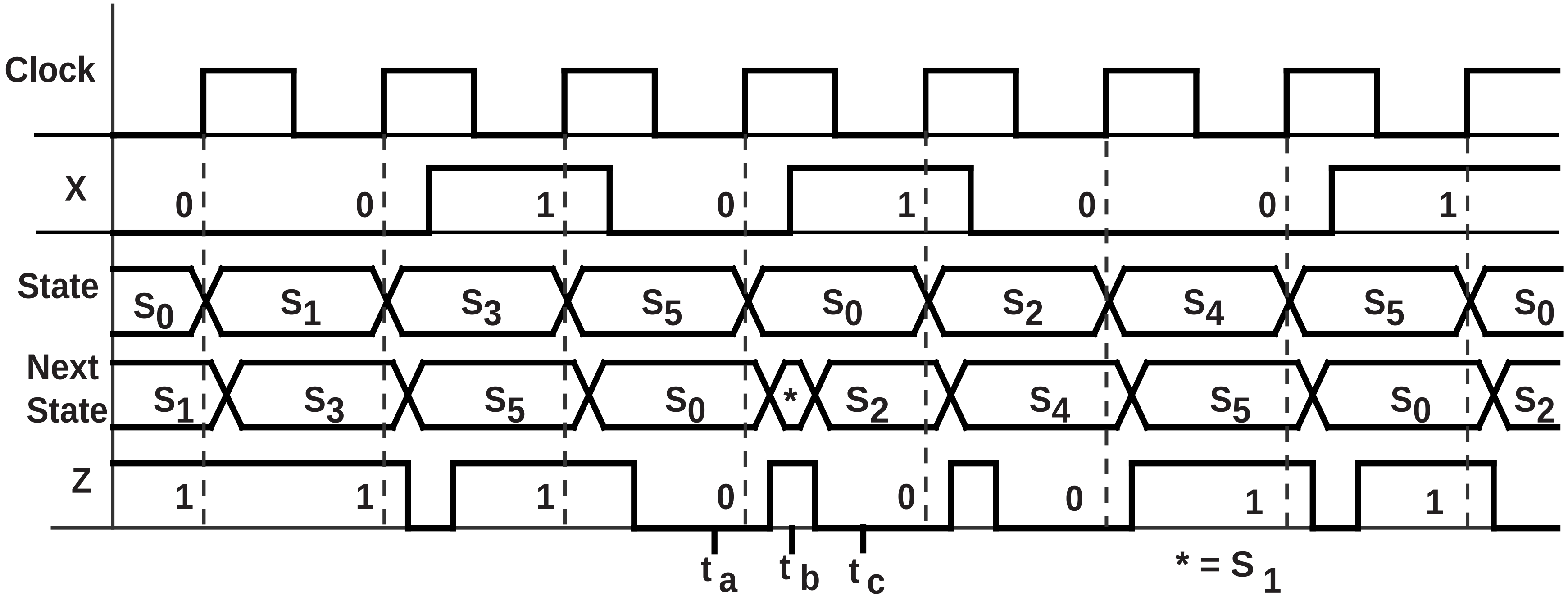


Figure 1-28 Timing Diagram for Figure 1-20

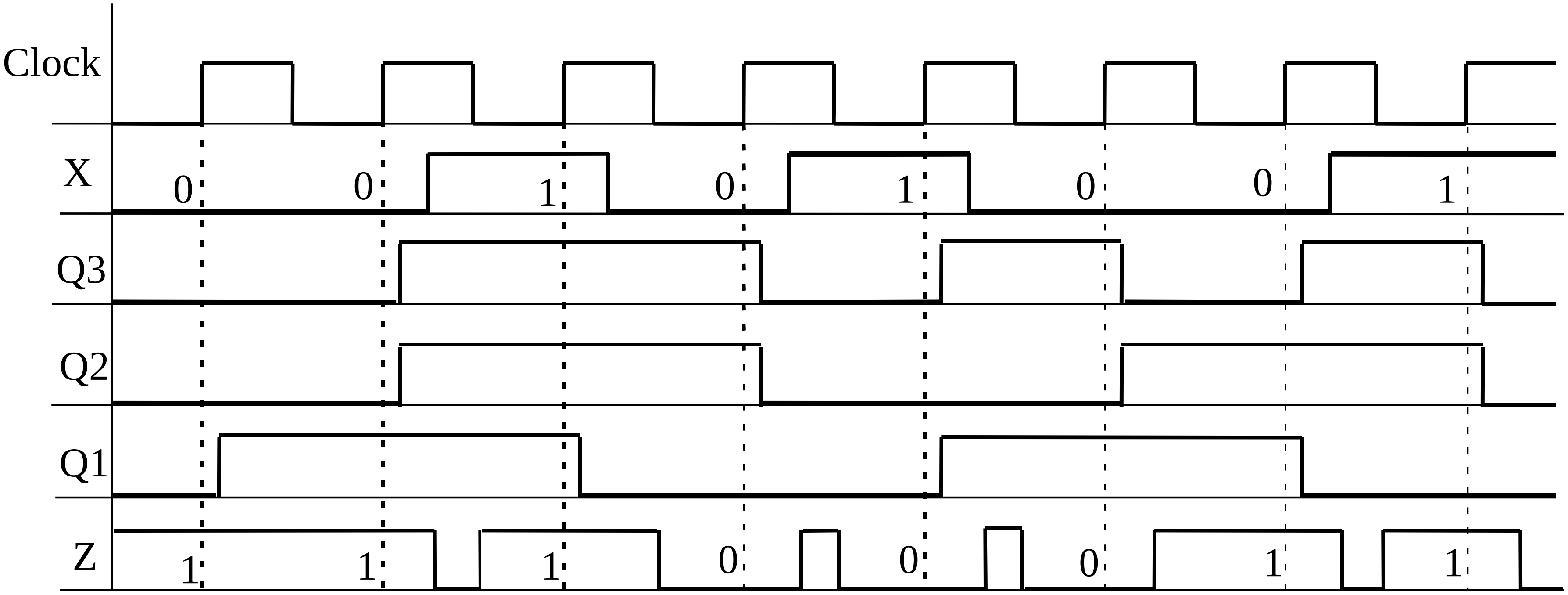


Figure 1-29 Setup and Hold Times for D Flip-flop

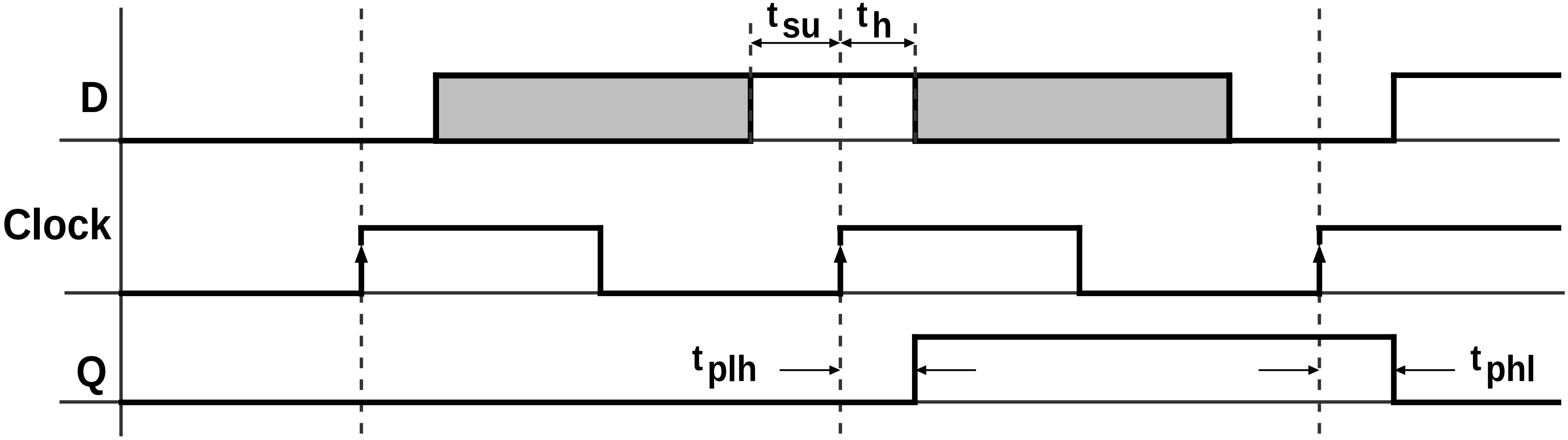


Figure 1-30 Setup and Hold Timing for Changes in X

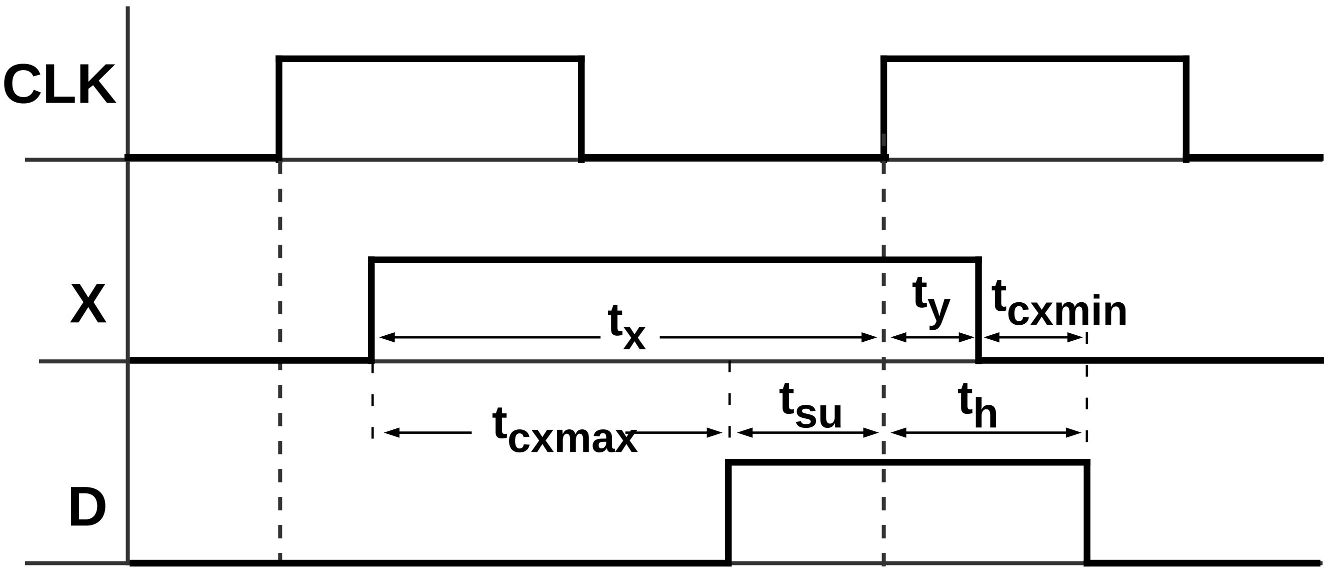


Figure 1-31 Synchronous Digital System

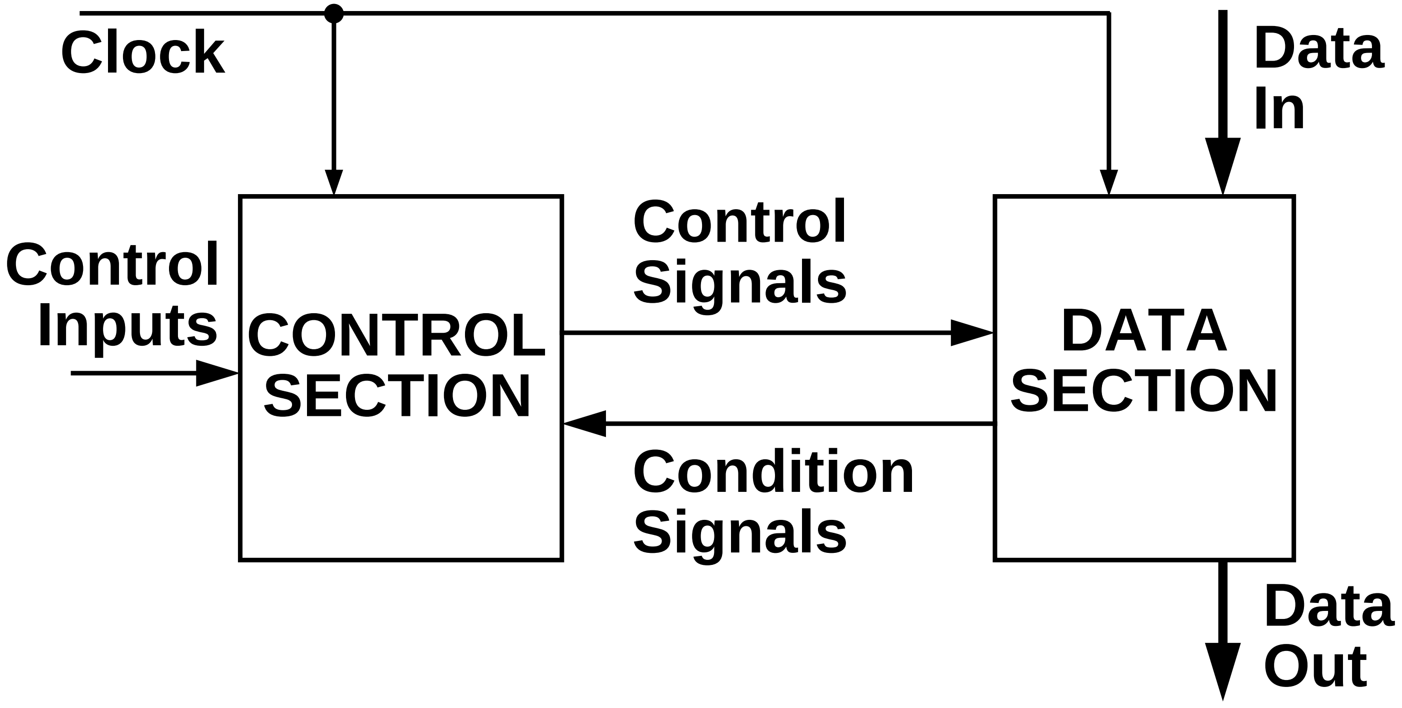


Figure 1-32 Timing Chart for System with Falling-Edge Devicves

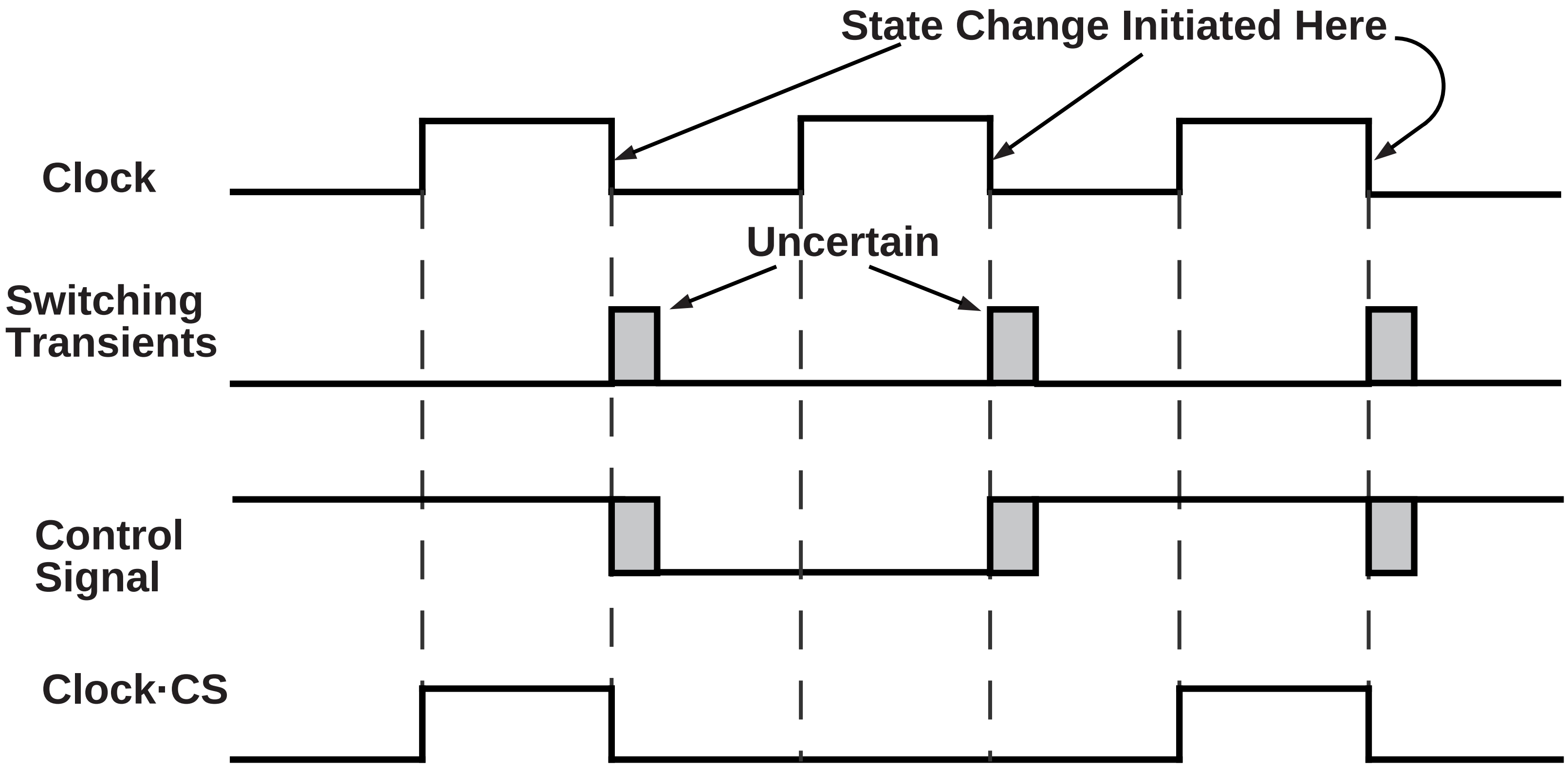
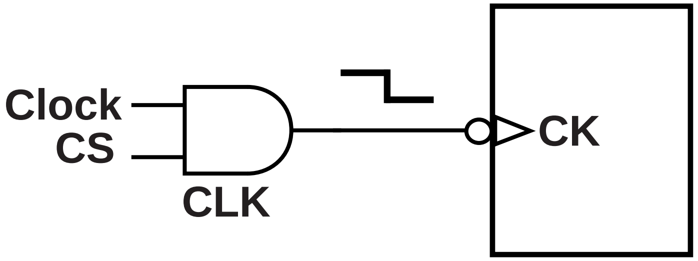
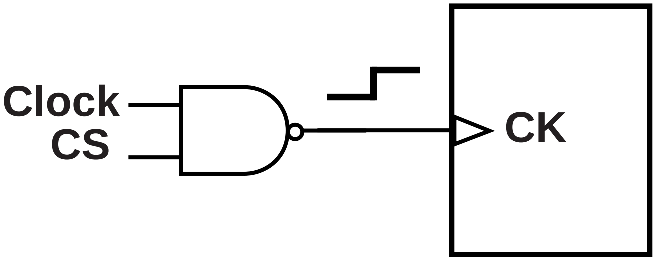


Figure 1-33 Gated Control Signal



(a) Faling-edge device



(b) Rising-edge device

Figure 1-34 Timing Chart with Rising-Edge Devices

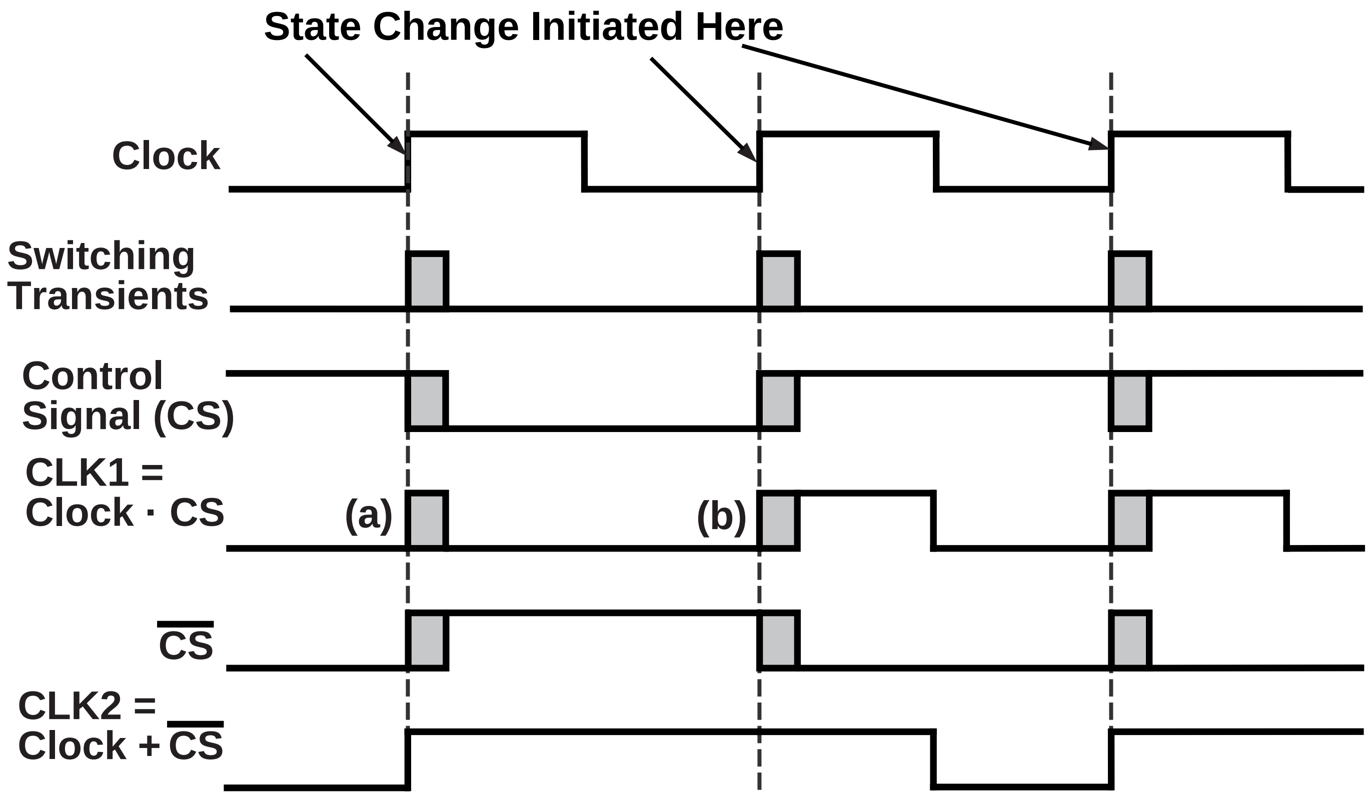


Figure 1-35 Incorrect Design

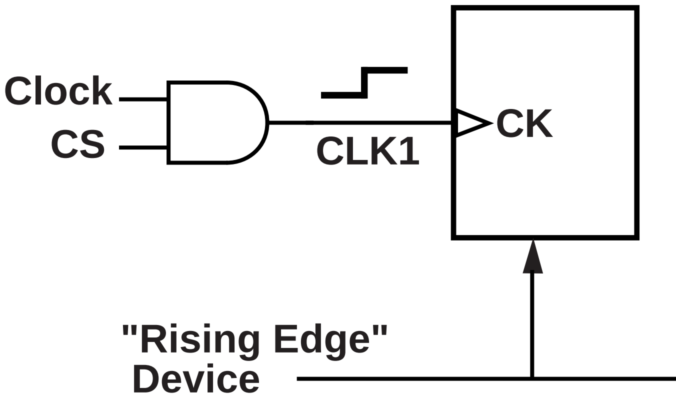
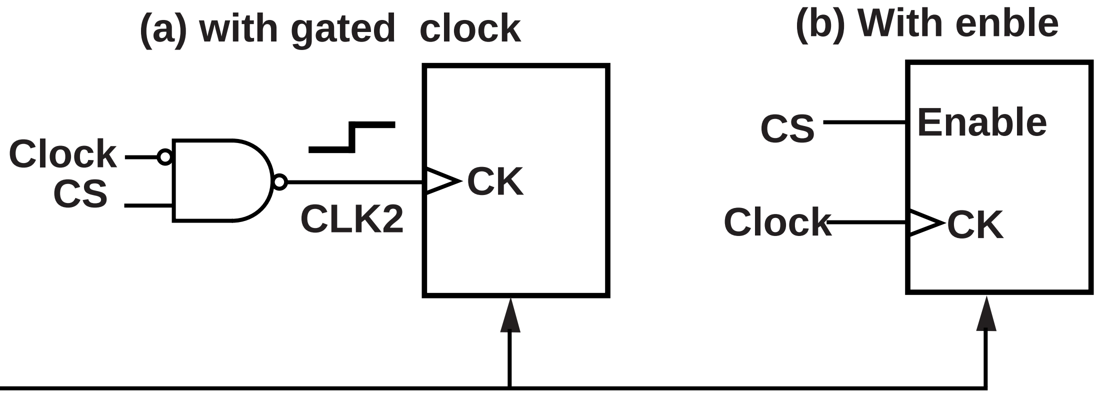


Figure 1-36 Correct Design


(a) with gated clock

(b) With enble

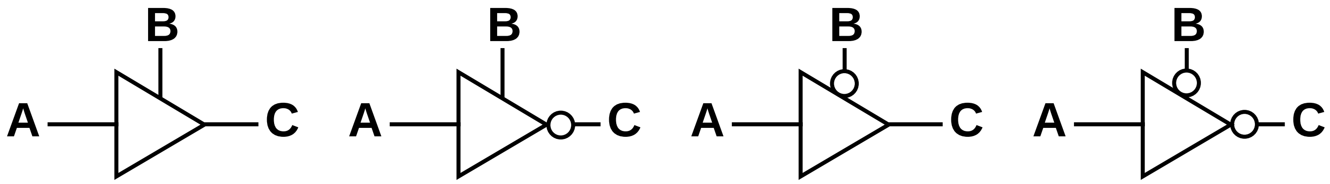
Synchronous Design Principals (from page 34)

Method: All clock inputs to flip-flops, registers, counters, etc. are driven directly from the system clock or from the clock ANDed with a control signal.

Result: All state changes occur immediately following the active edge of the clock signal.

Advantage: All switching transients, switching noise, etc. occur between clock pulses and have no effect on system performance.

Figure 1-37 Four Kinds of Tristate Buffers



B	A	C
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1

(a)

B	A	C
0	0	Hi-Z
0	1	Hi-Z
1	0	1
1	1	0

(b)

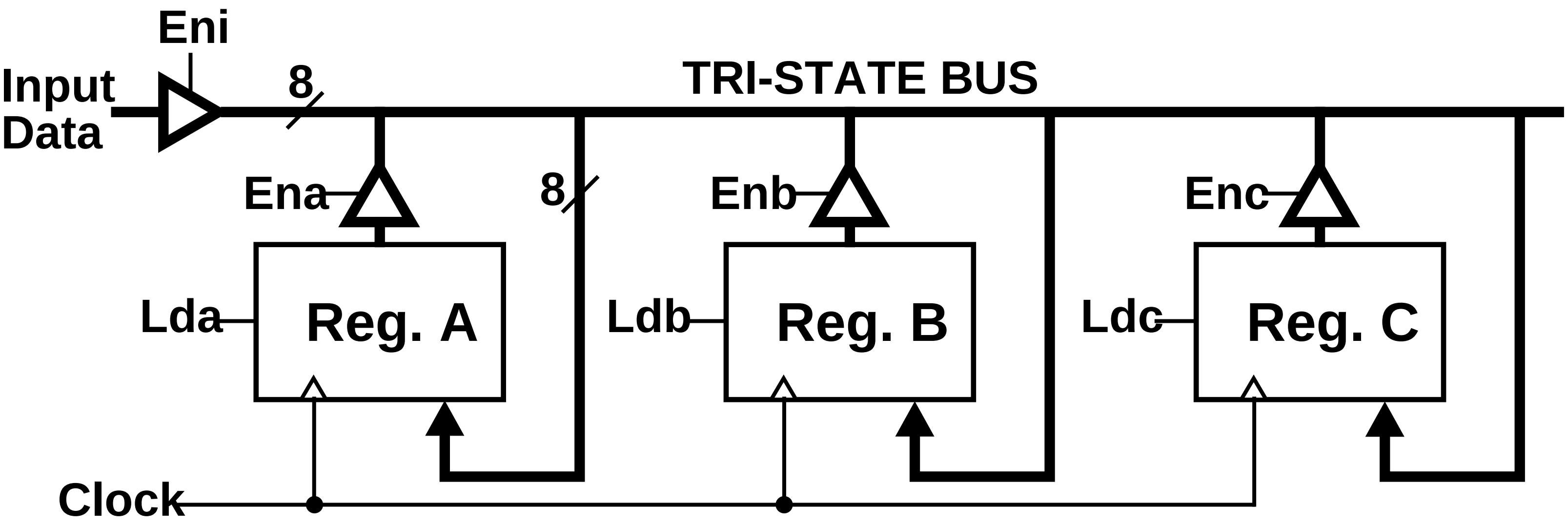
B	A	C
0	0	0
0	1	1
1	0	Hi-Z
1	1	Hi-Z

(c)

B	A	C
0	0	1
0	1	0
1	0	Hi-Z
1	1	Hi-Z

(d)

Figure 1-38 Data Transfer Using Tristate Bus


Eni
Input
Data
8
TRI-STATE BUS
Ena
8
Enb
Enc
Lda
Reg. A
Ldb
Reg. B
Ldc
Reg. C
Clock