PowerPoint Slides

to accompany

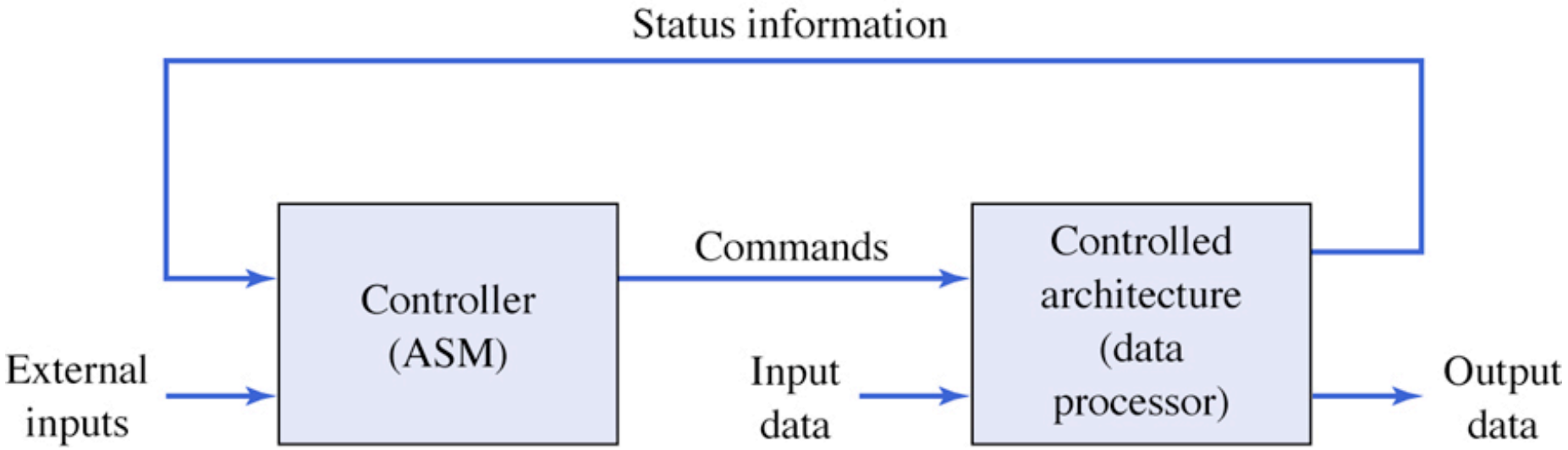
Digital Principles and Design

Donald D. Givone

Chapter 8
Algorithmic State Machines

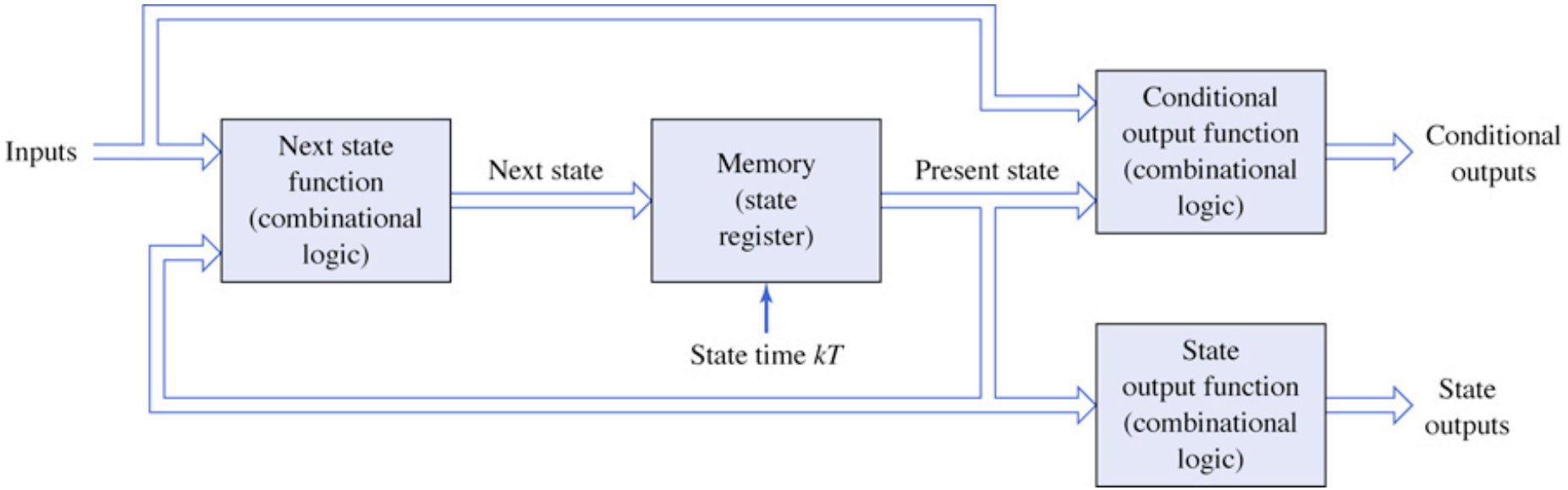
Partitioning of a digital system.

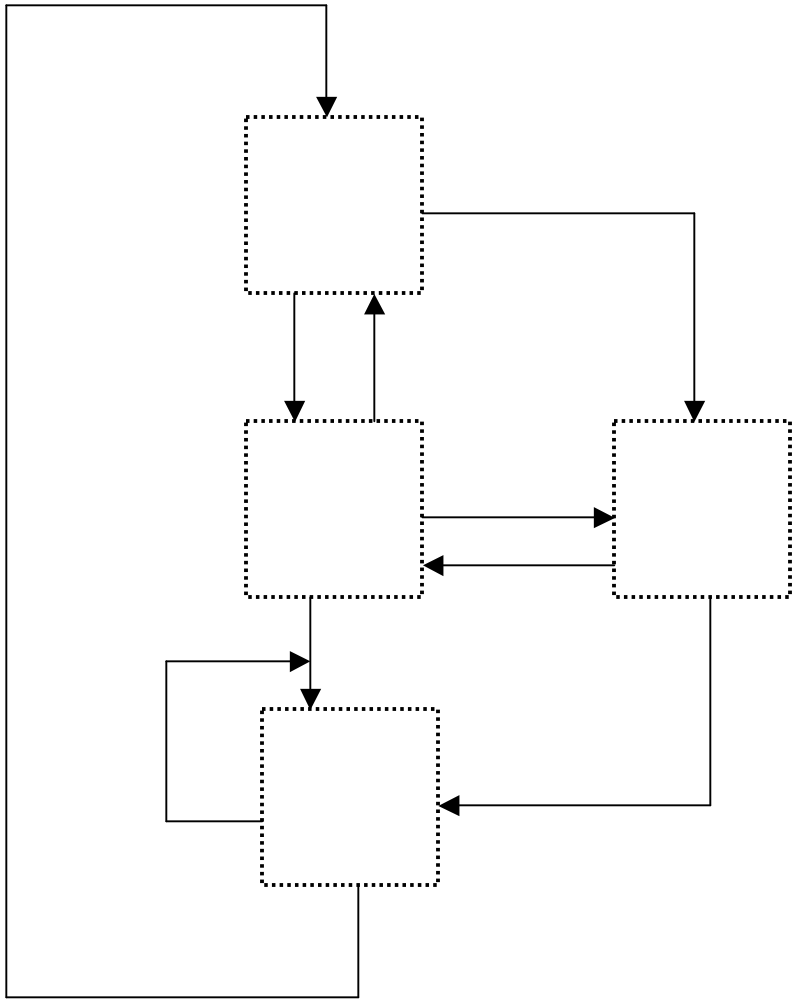
Figure 8.1



Model of an algorithmic state machine.

Figure 8.2





Blocks define states

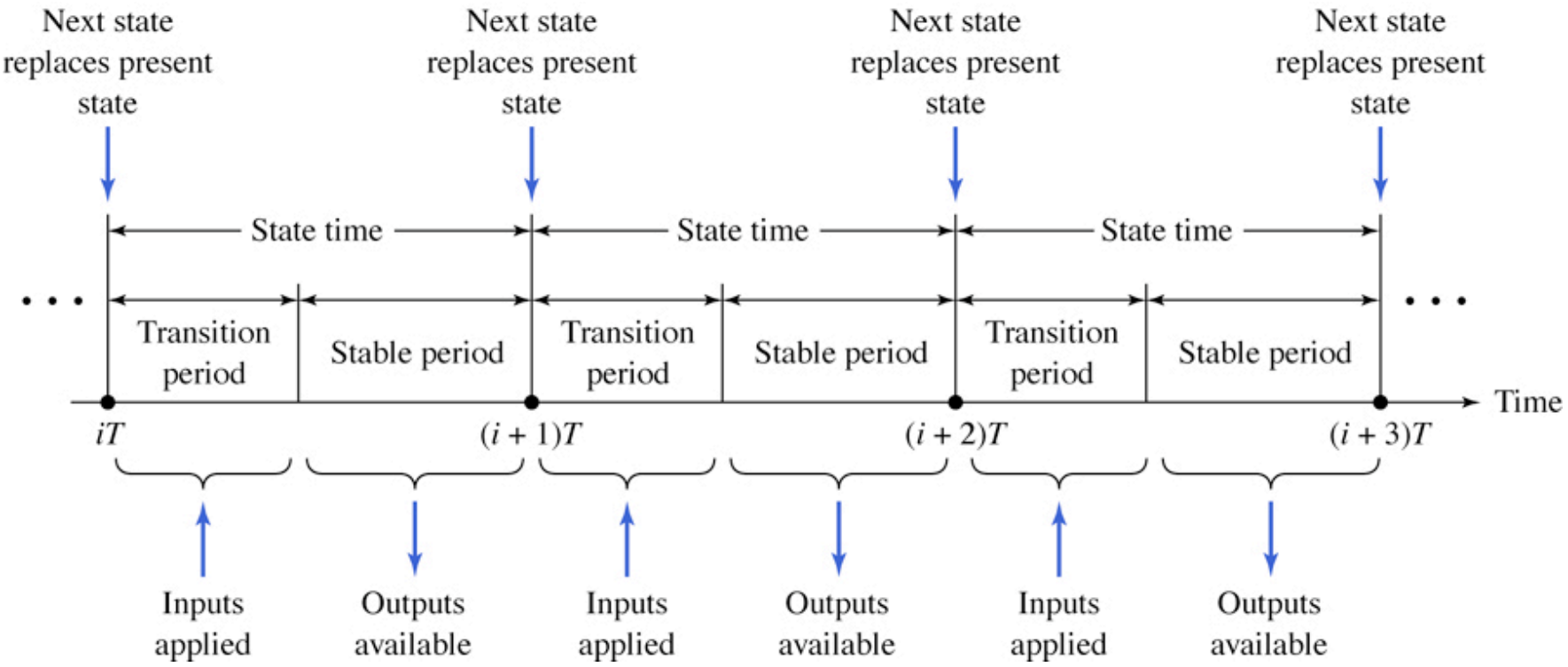
Transition & output logic

contained within

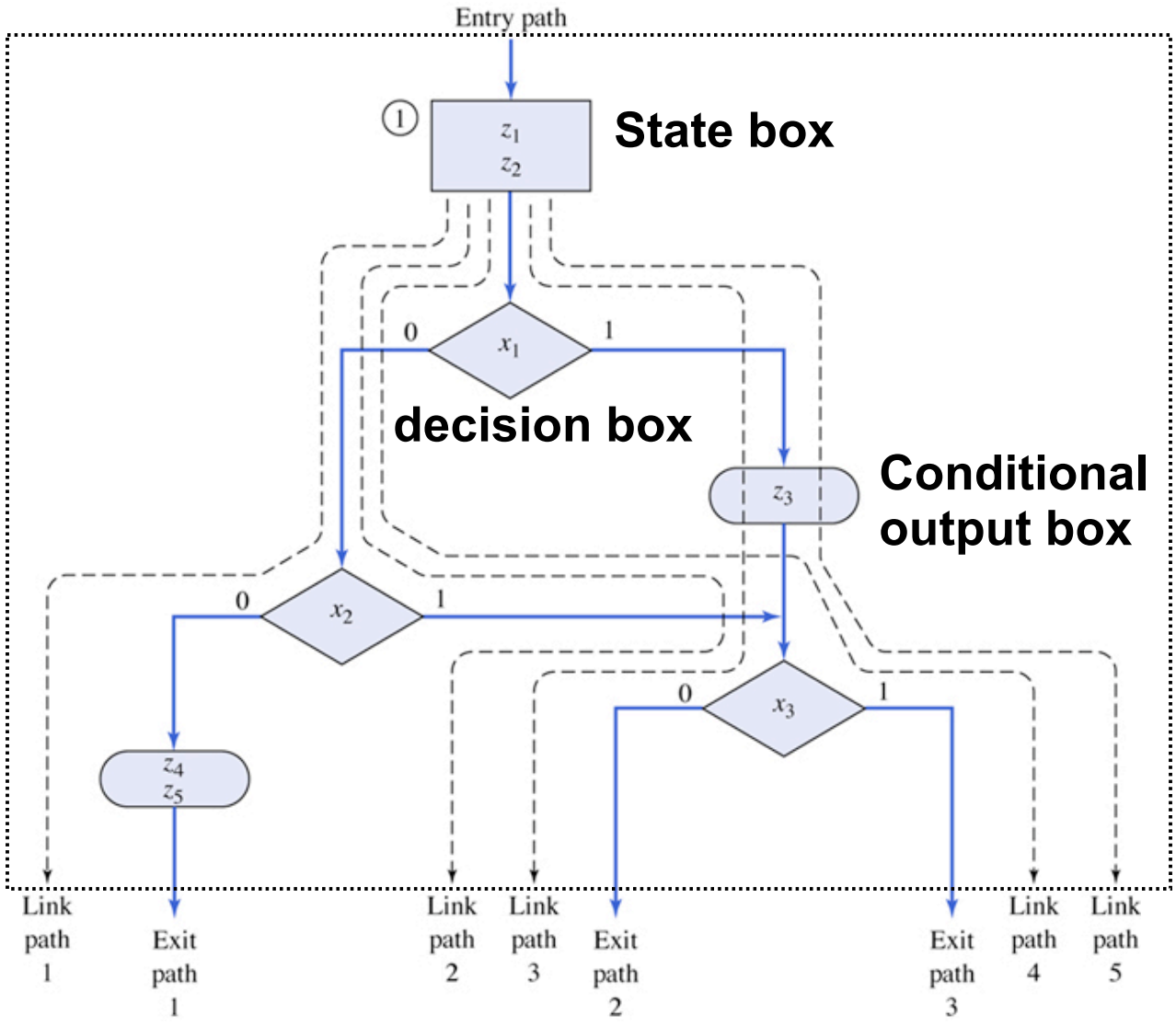
Compare with Mealy/Moore

Timing of an algorithmic state machine.

Figure 8.3

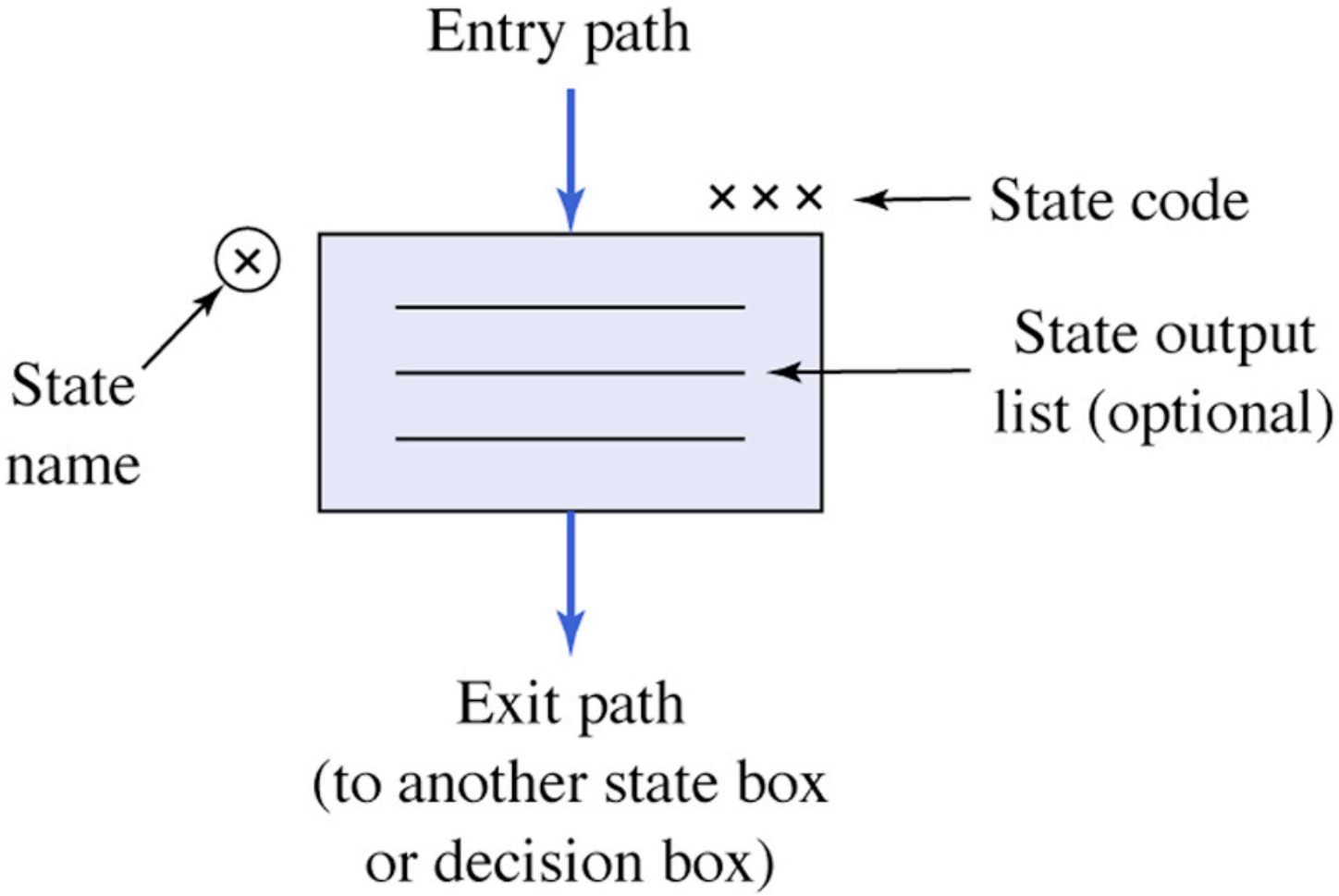


Example ASM Block



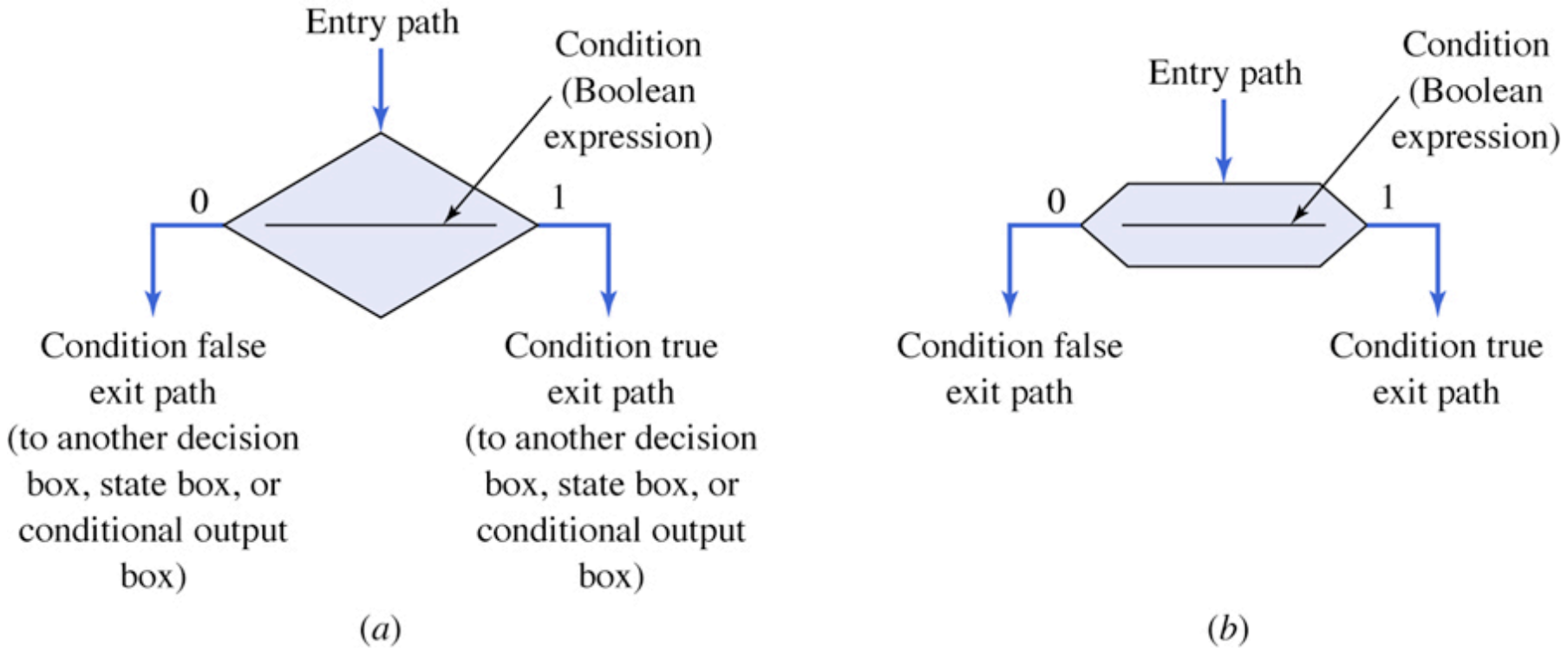
The state box.

Figure 8.4



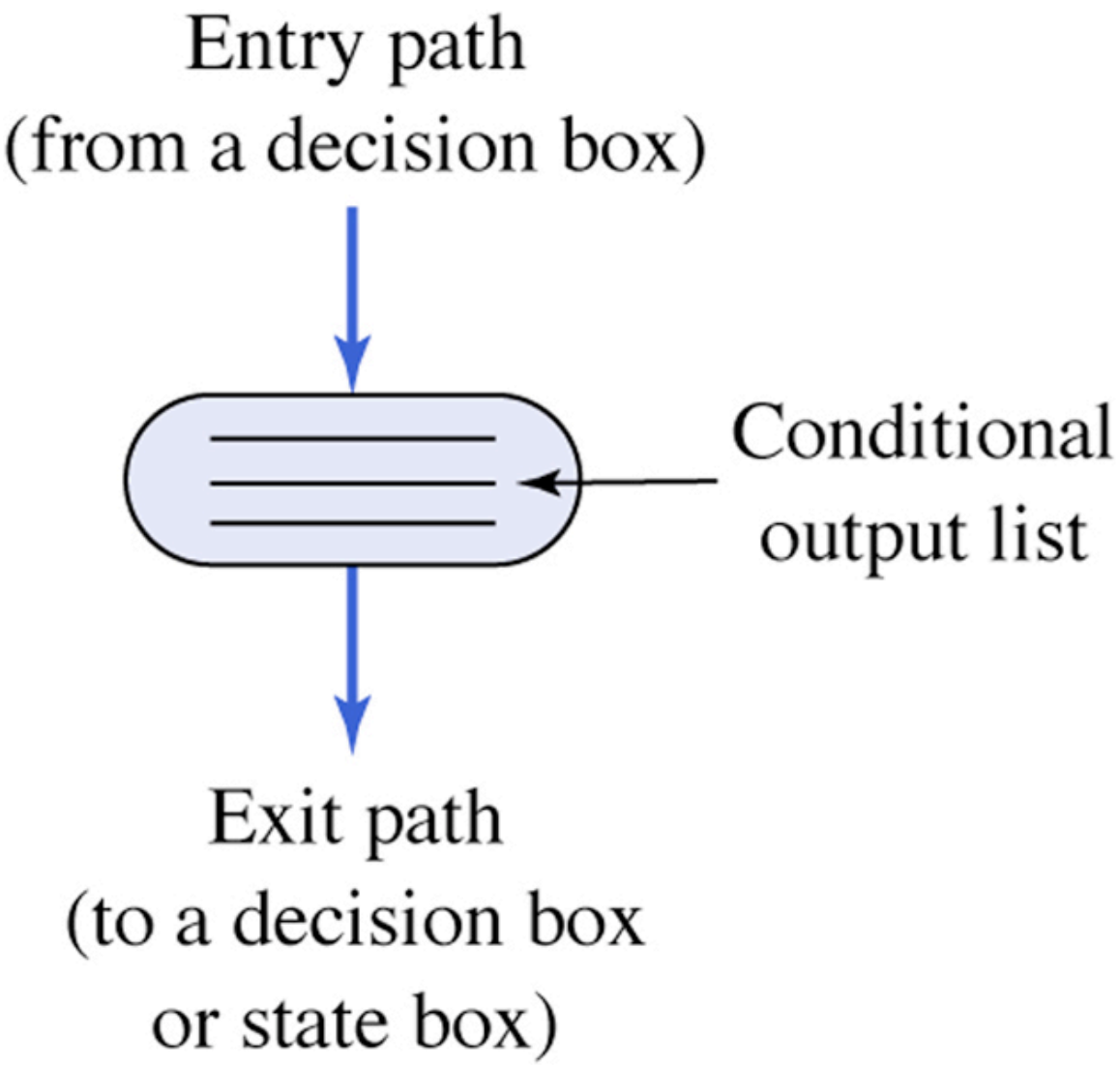
The decision box. (*a*) Symbol. (*b*) Alternate symbol.

Figure 8.5



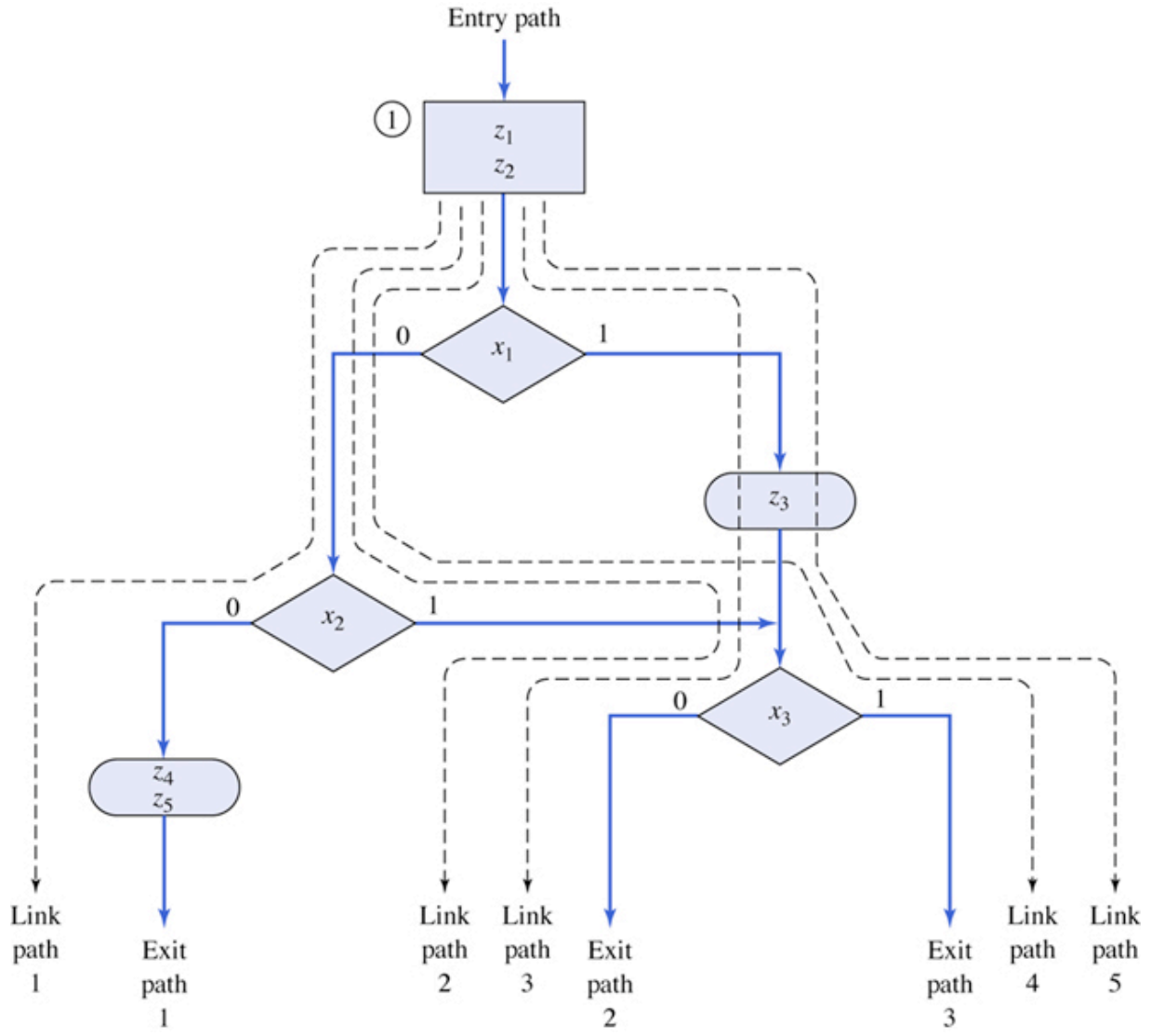
The conditional output box.

Figure 8.6

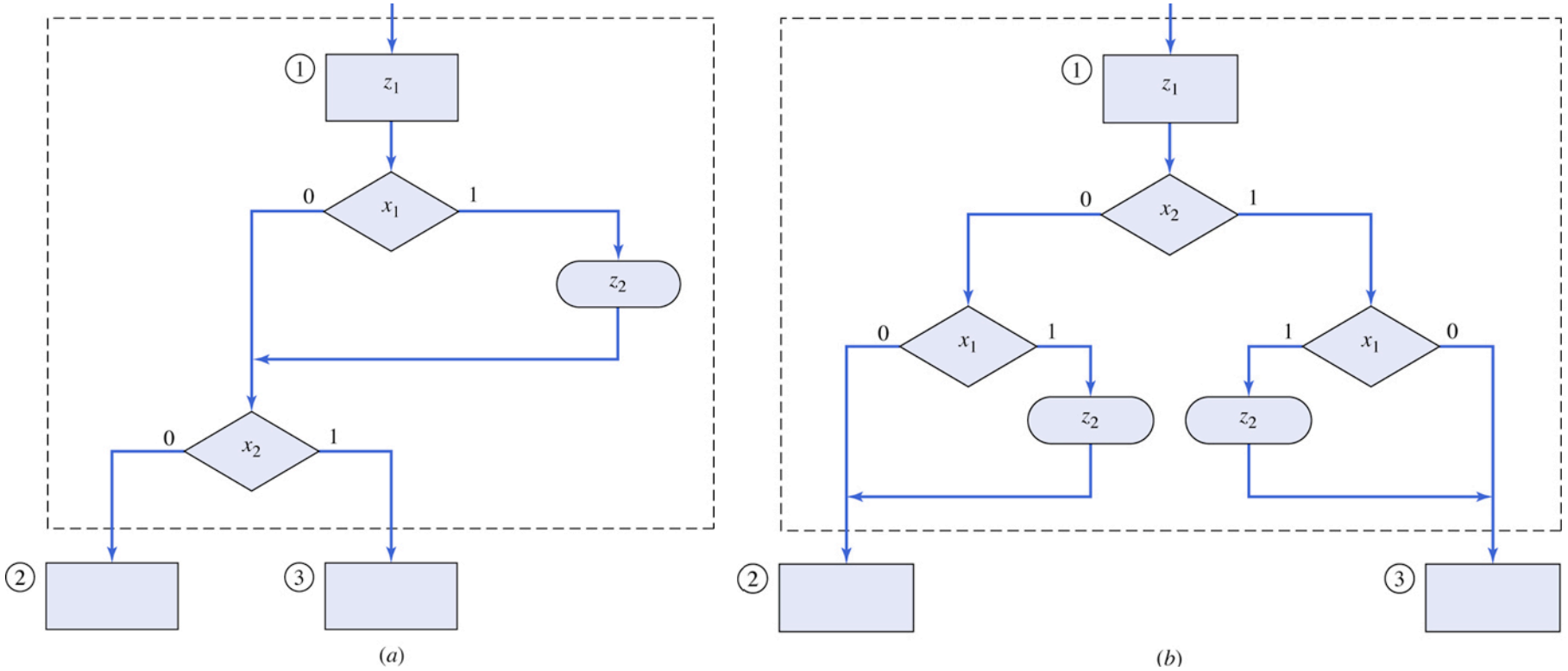


Example of an ASM block and its link paths.

Figure 8.7



Two equivalent ASM blocks.

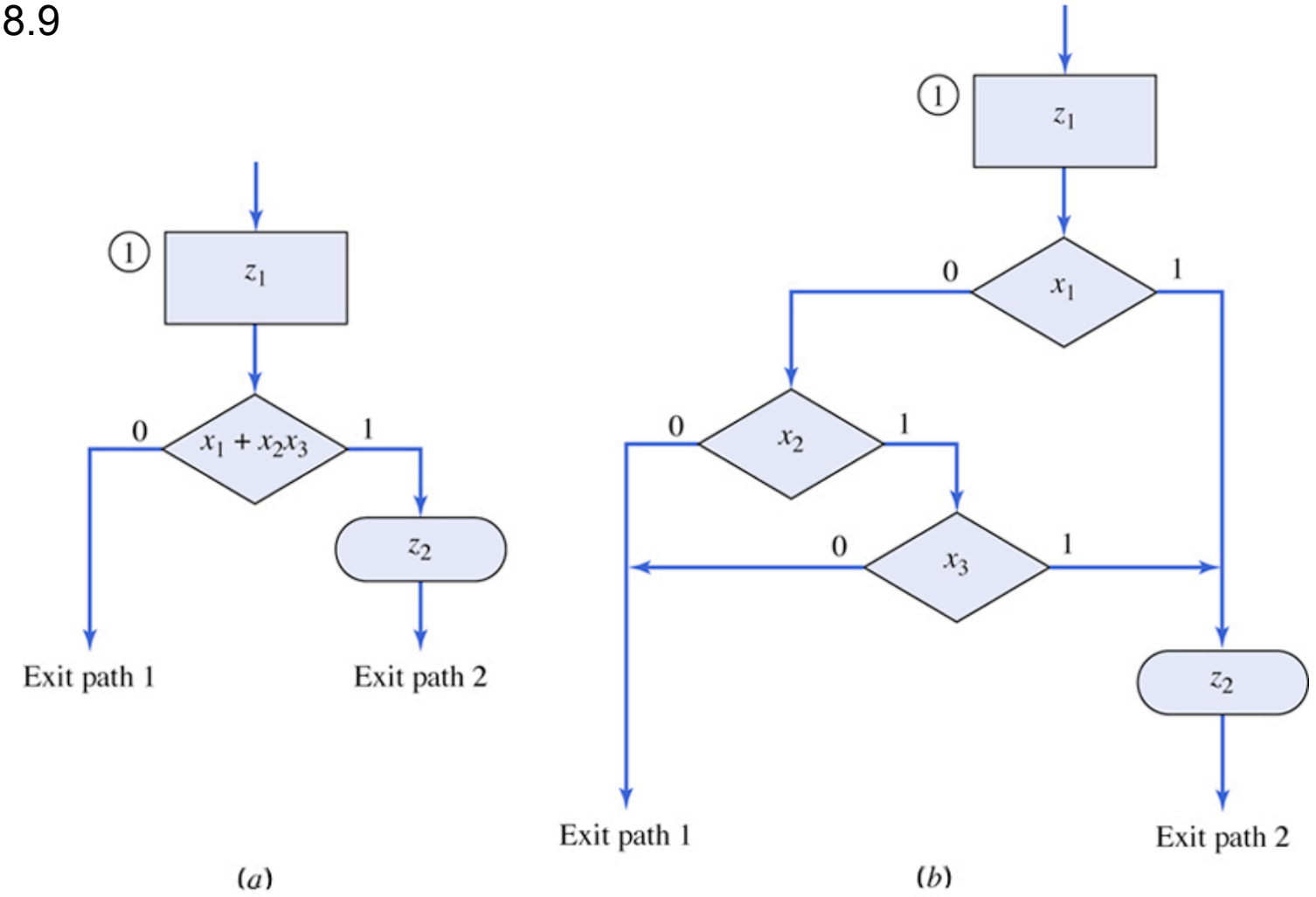


Two blocks are equivalent if

- same state output variables named in state box
- for every setting of input values
  - the same next state is chosen
  - the same set of output variables are named in the set of conditional output boxes traversed

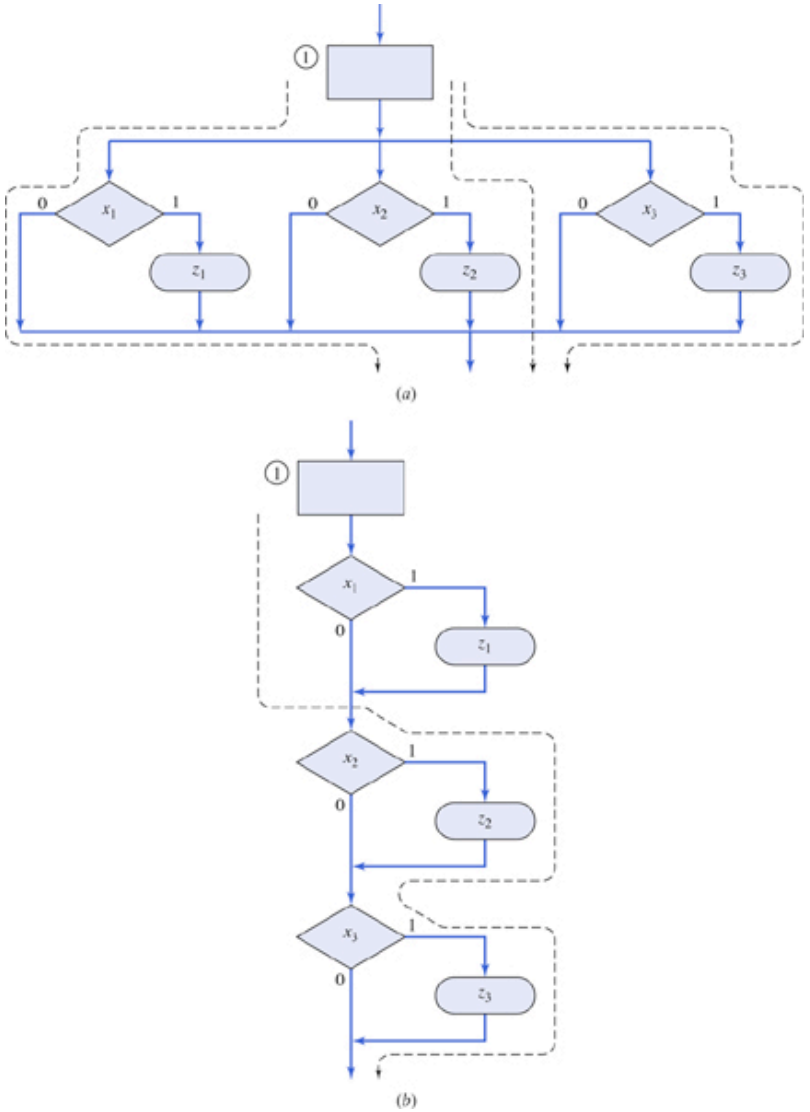
Two equivalent ASM blocks. (*a*) Using a single decision box. (*b*) Using several decision boxes.

Figure 8.9



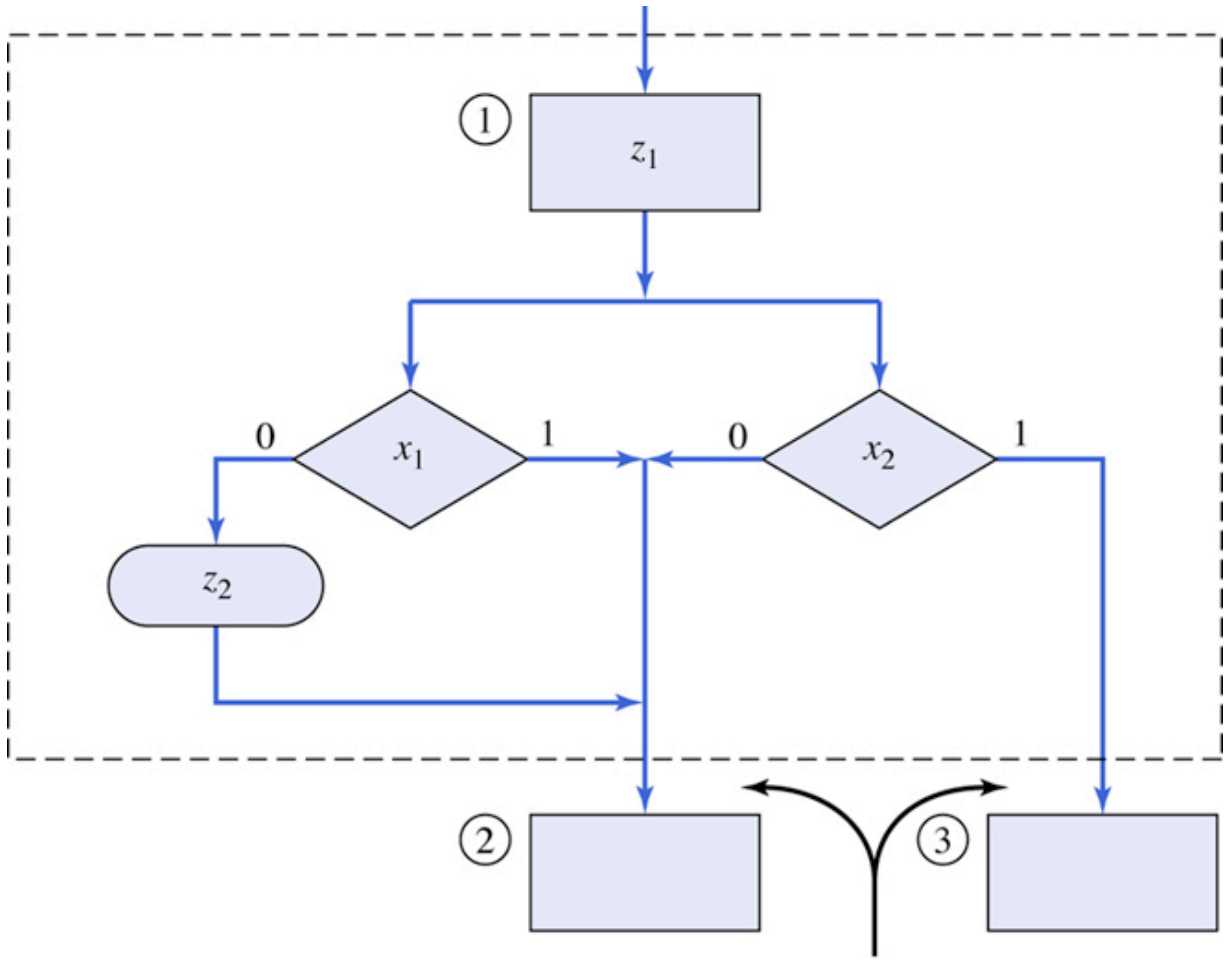
Two equivalent ASM books blocks. (*a*) Parallel decision boxes. (*b*) Serial decision boxes.

Figure 8.10



Invalid ASM block having nonunique next states.

Figure 8.11

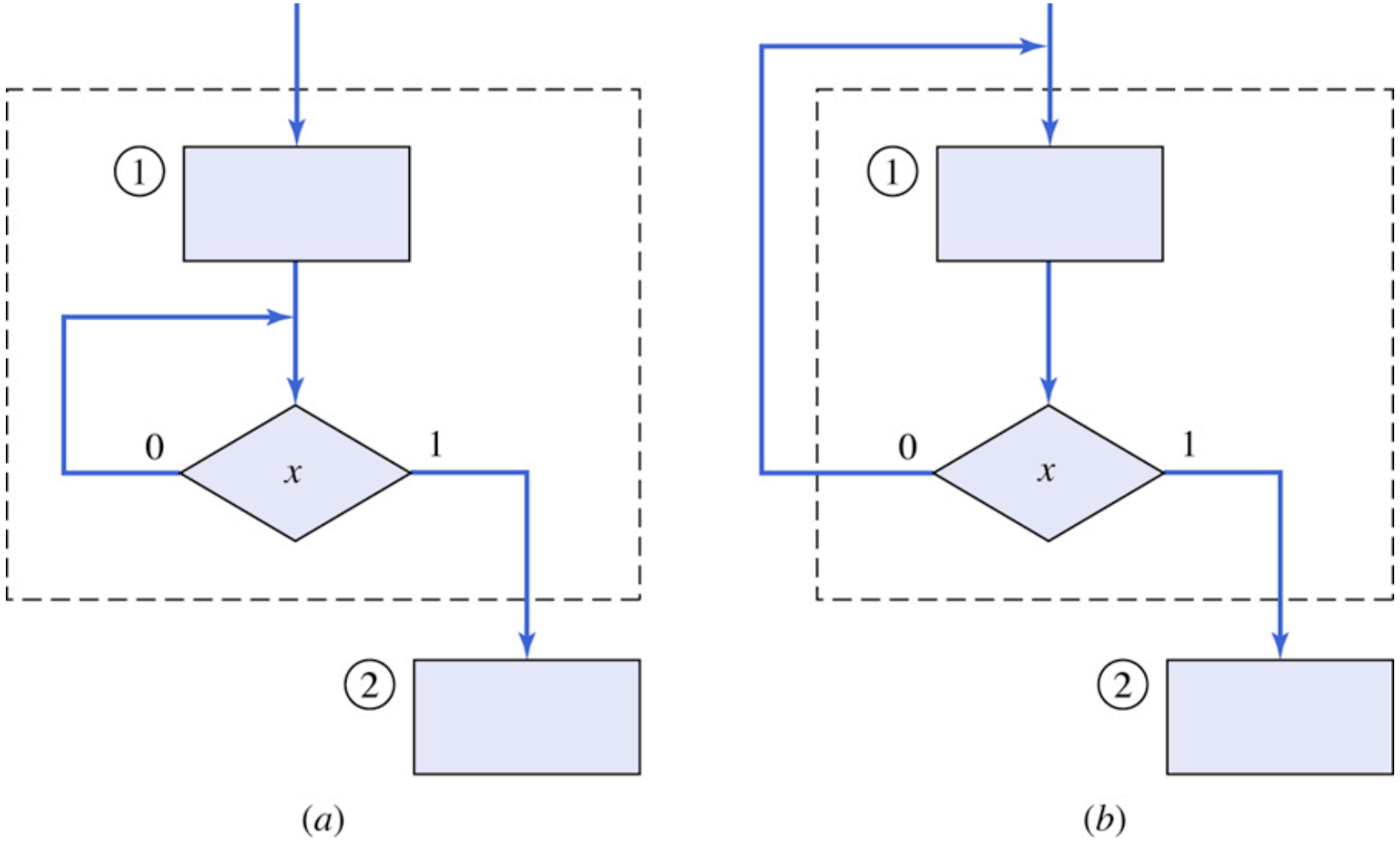


Both exits selected when

both inputs are 1

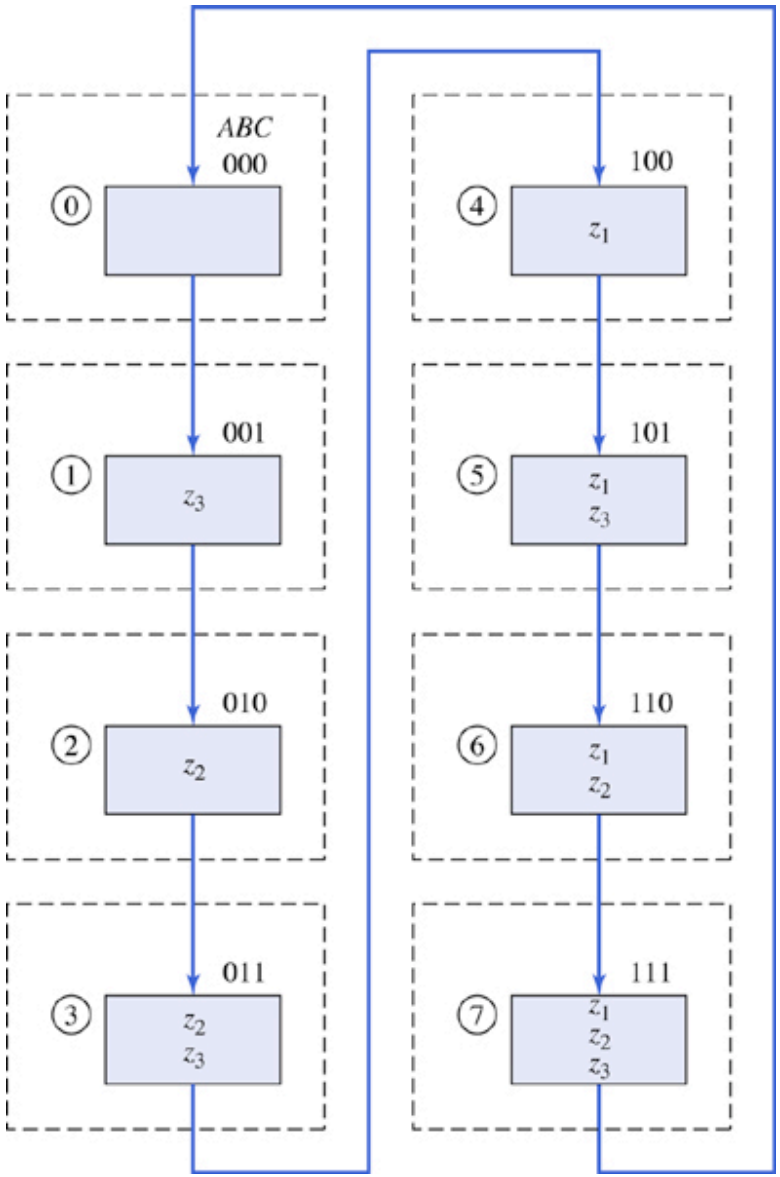
Looping. (*a*) Incorrect. (*b*) Correct.

Any closed loop must contain at least one state box



ASM chart for a mod-8 binary counter.

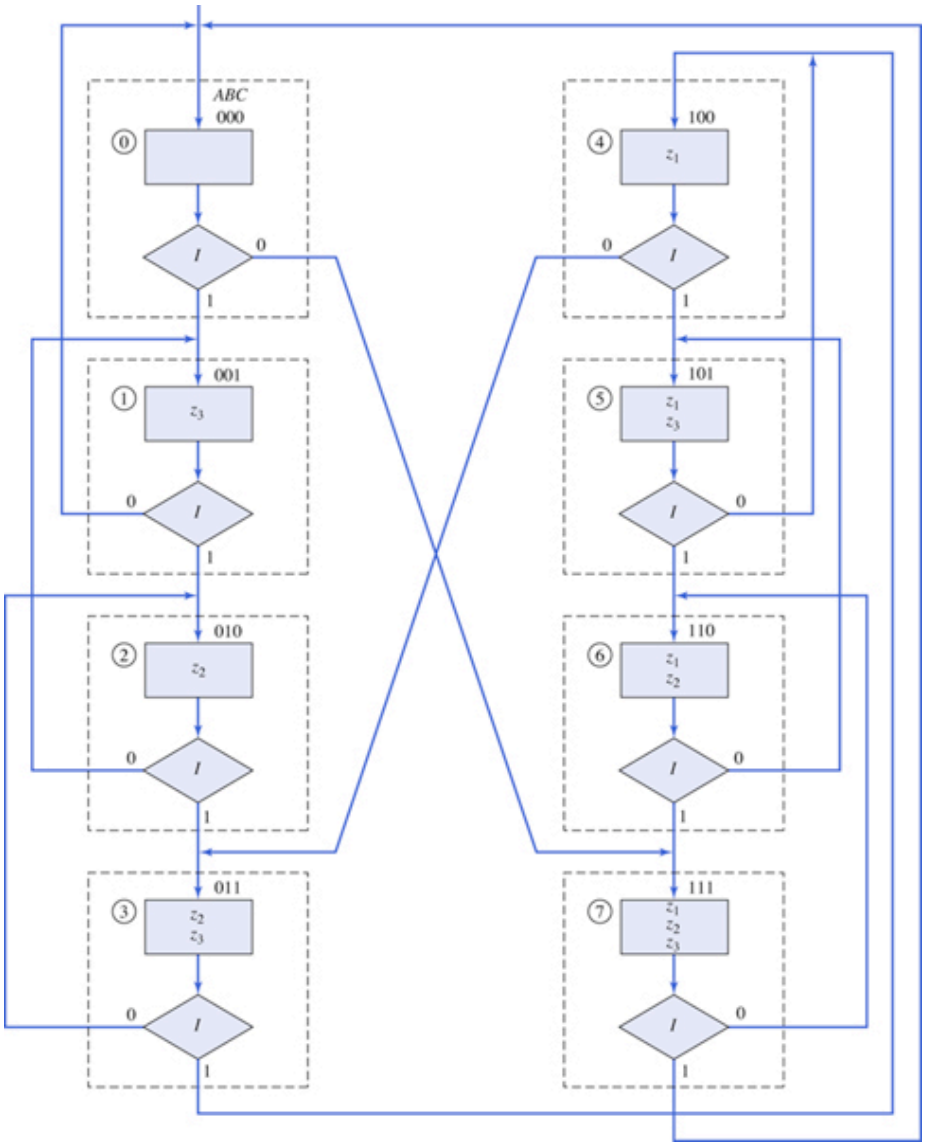
Figure 8.13



State output is state code

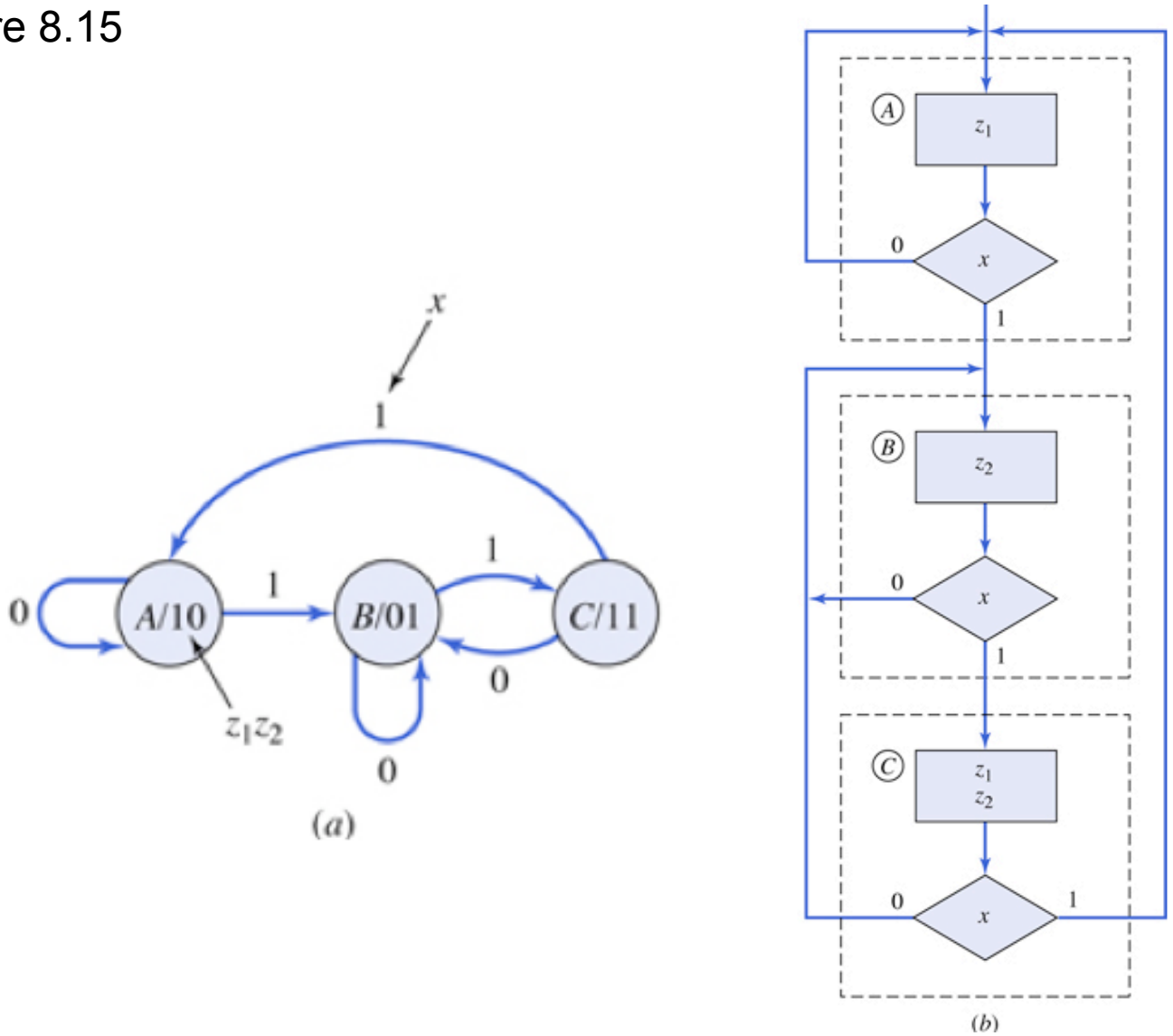
ASM chart for a mod-8 binary up-down counter.

Input I controls direction



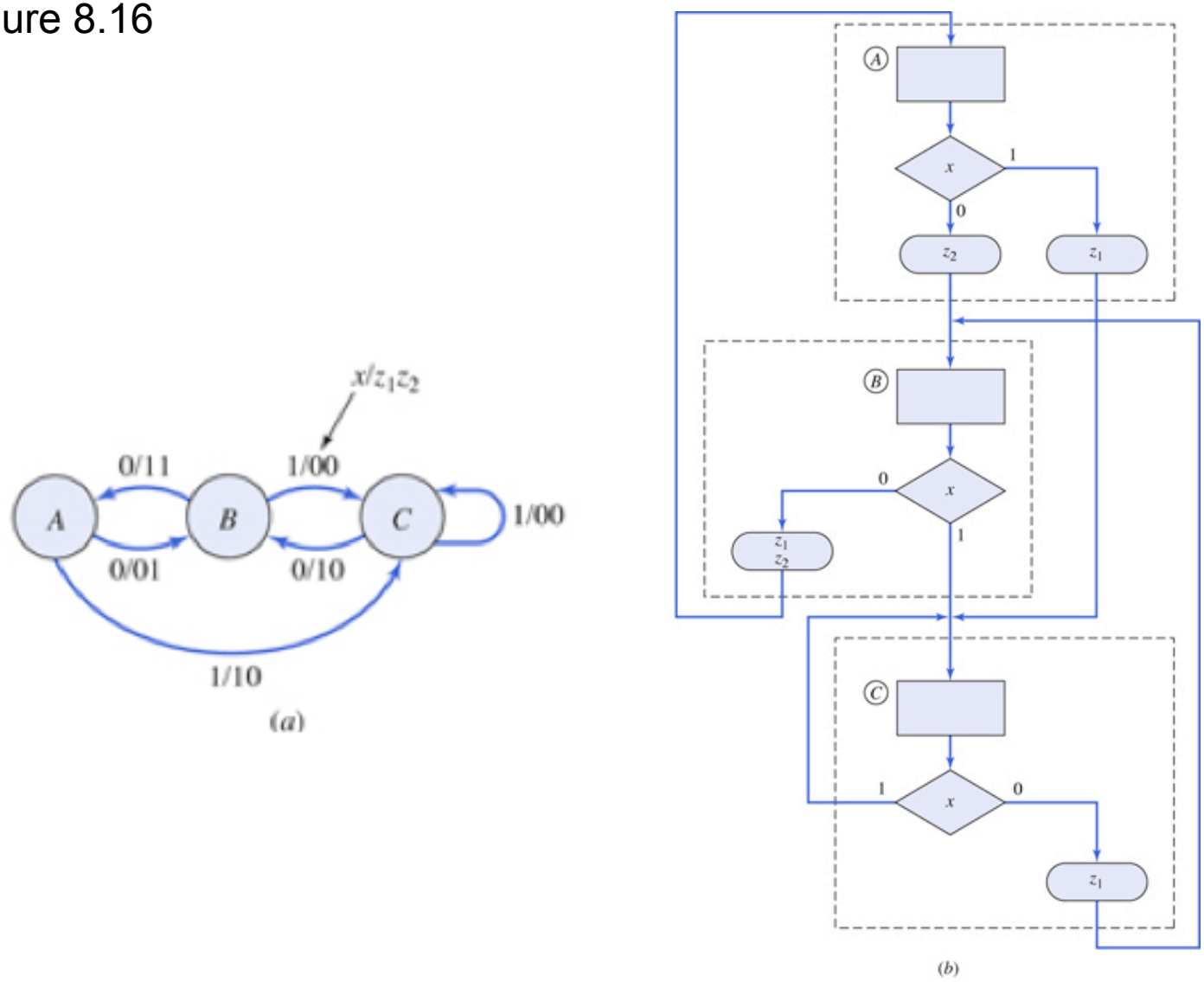
Moore sequential network. (*a*) State diagram. (*b*) ASM chart.

Figure 8.15



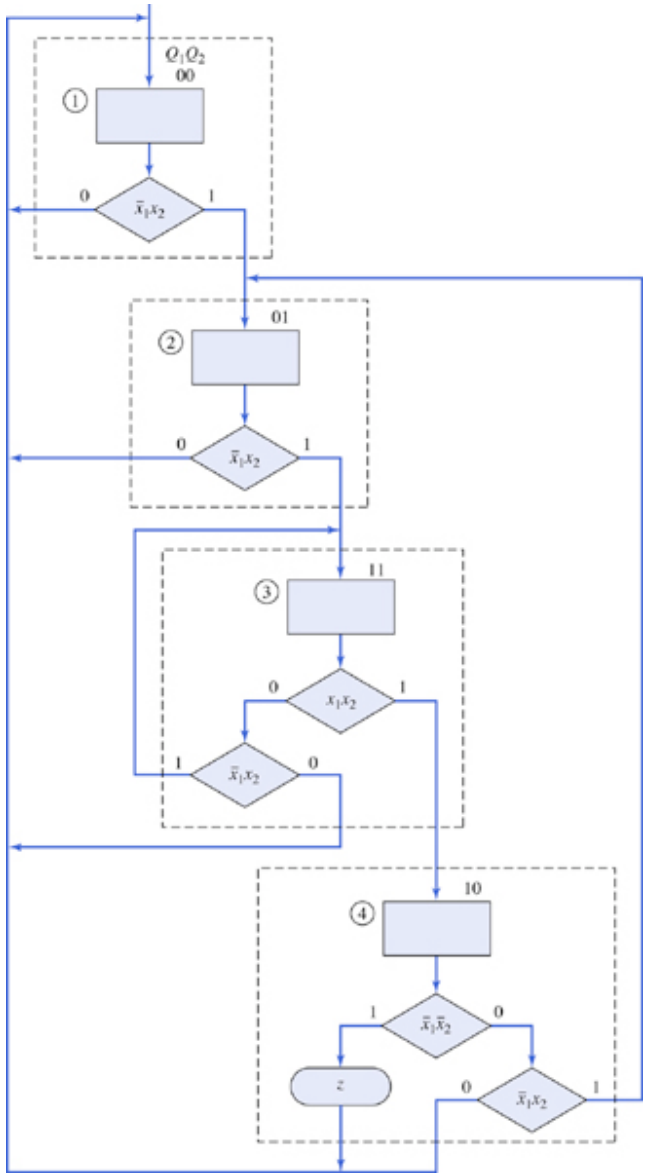
Mealy sequential network. (*a*) State diagram. (*b*) ASM chart.

Figure 8.16



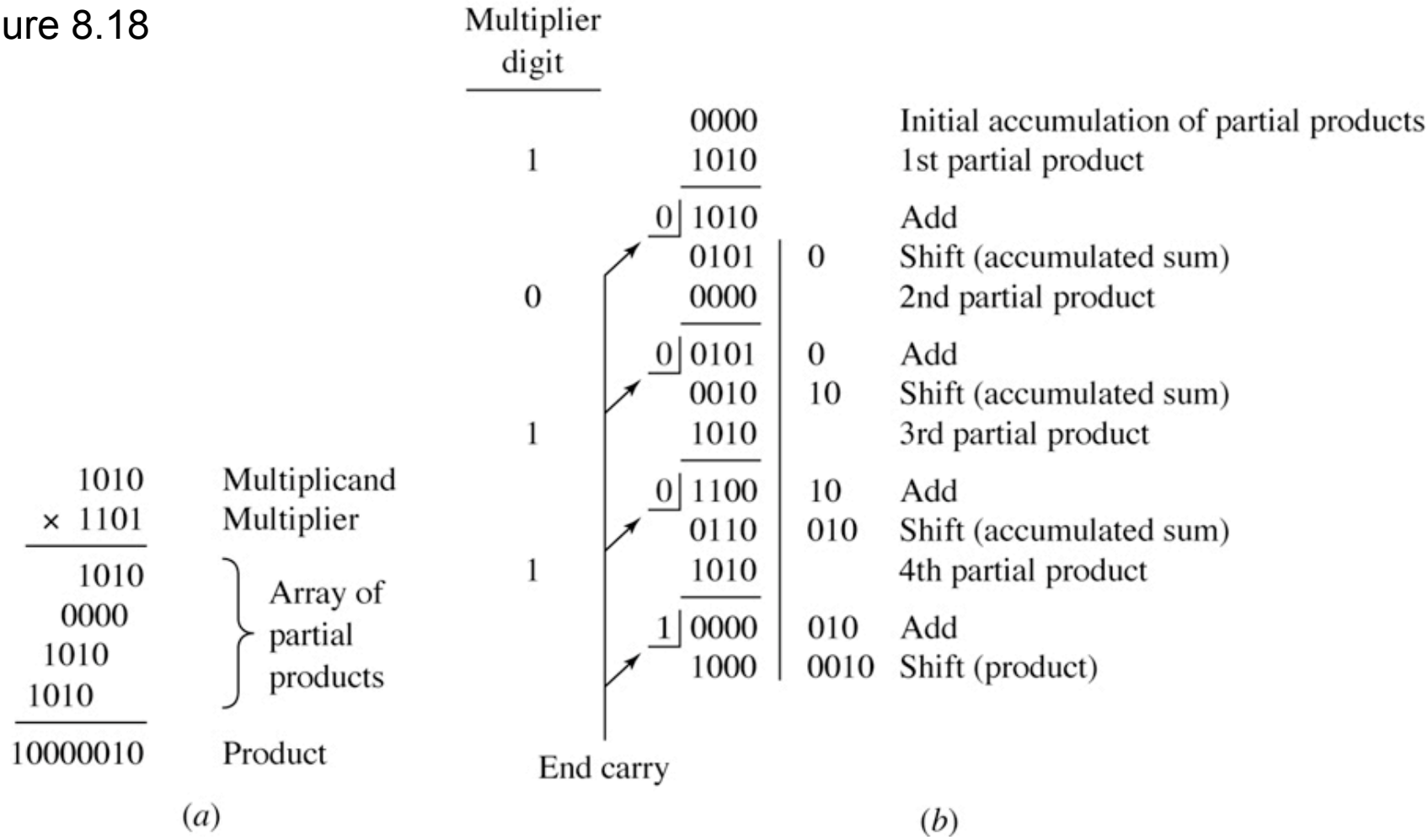
ASM chart to recognize the sequence x_1x_2 = 01,01,11,00.

Figure 8.17



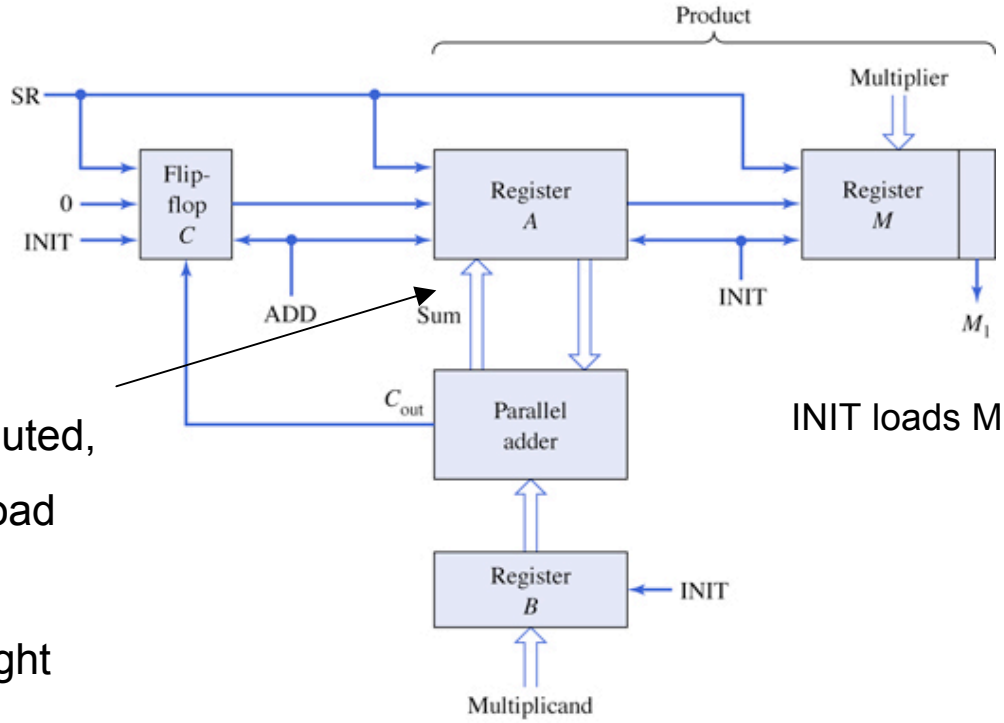
Binary multiplication. (*a*) Pencil-and-paper approach. (*b*) Add-shift approach.

Figure 8.18



Architecture for a binary multiplier.

Figure 8.19

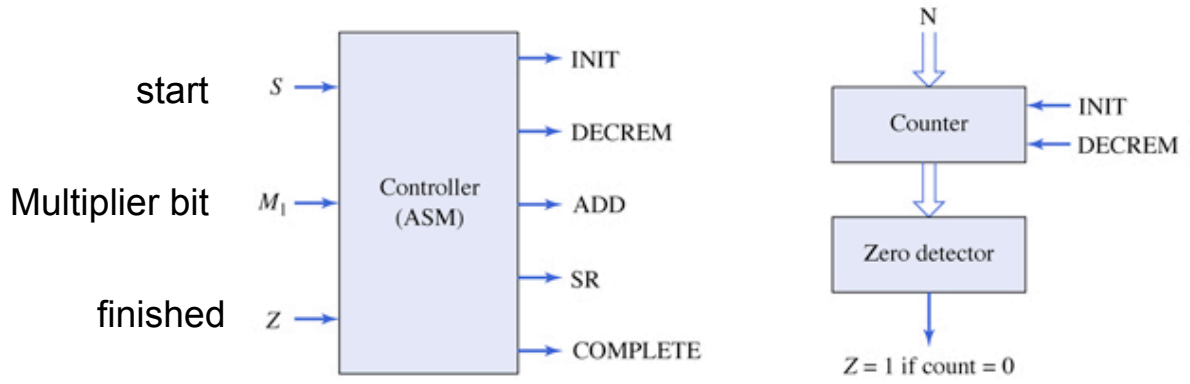


Sum always computed,

ADD tells A,C to load

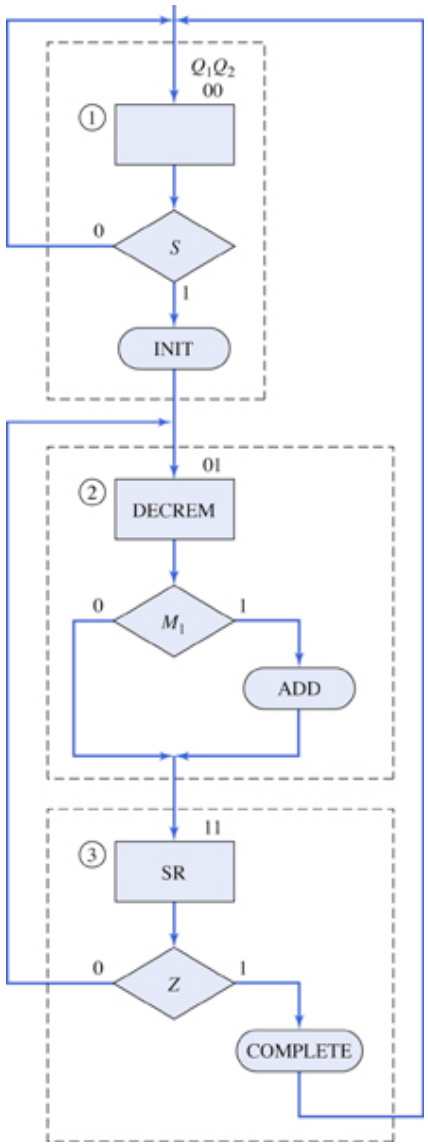
SR shifts C A M right

INIT loads M,B, Counter, clears C

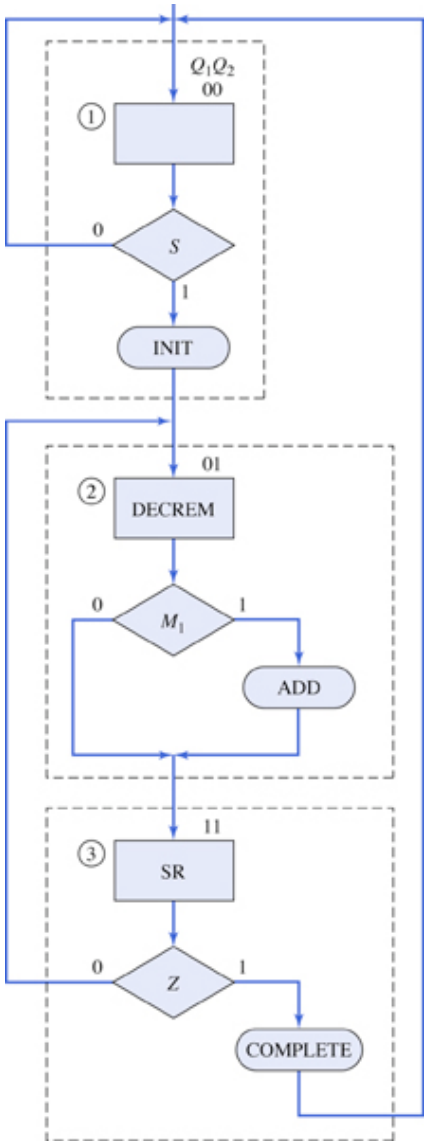


ASM chart for a binary multiplier.

Figure 8.20



Assigned ASM Table



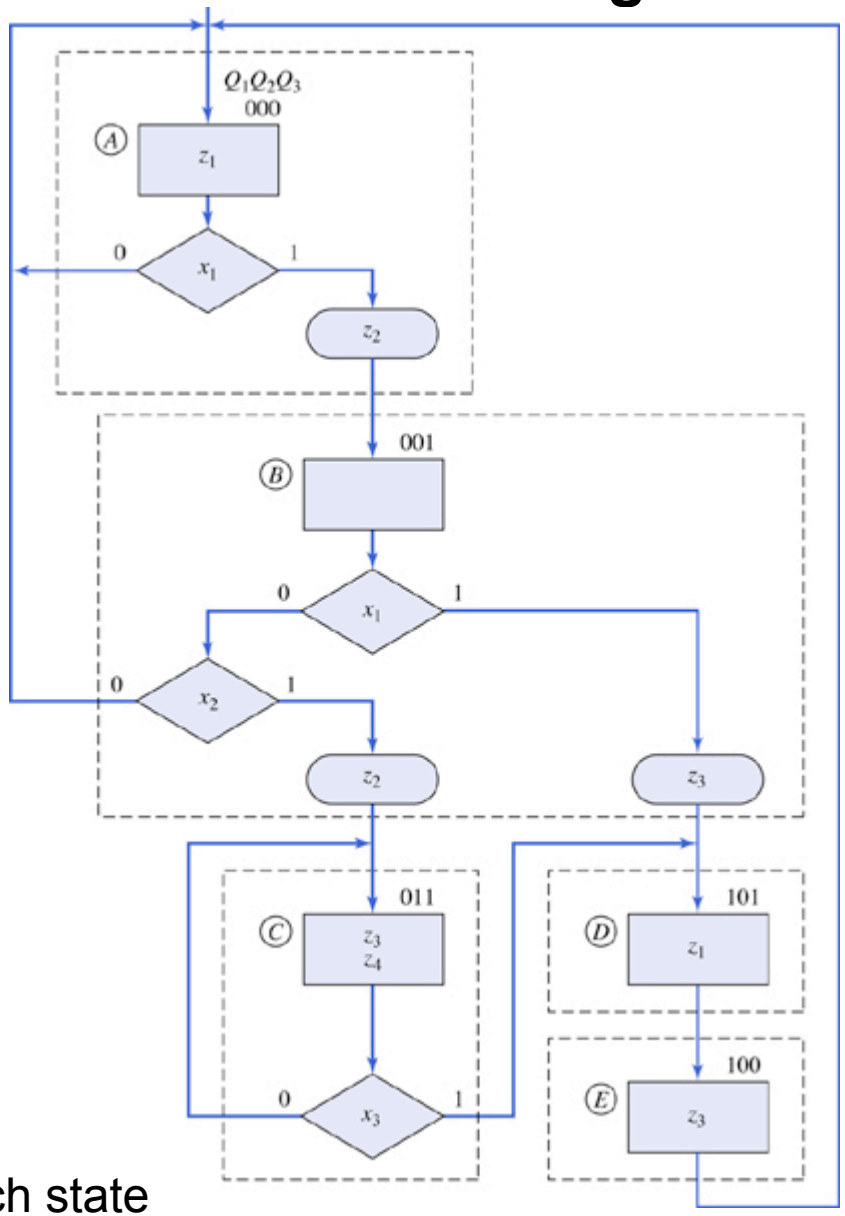
Link path	Present state Sym	Q_1	Q_2	Inputs S	M_1	Z	Next state Sym	Q_1^+	Q_2^+	Outputs INIT	DECREM	ADD	SR	COMPLETE
L_1	1	0	0	0	–	–	1	0	0	0	0	0	0	0
L_2	1	0	0	1	–	–	2	0	1	1	0	0	0	0
L_3	2	0	1	–	0	–	3	1	1	0	1	0	0	0
L_4	2	0	1	–	1	–	3	1	1	0	1	1	0	0
L_5	3	1	1	–	–	0	2	0	1	0	0	0	1	0
L_6	3	1	1	–	–	1	1	0	0	0	0	0	1	1

(*b*)

Note grouping of link paths

An ASM chart to illustrate state assignment.

Figure 8.21



A ->A,B

B->A,C,D

C->C,D

D->E

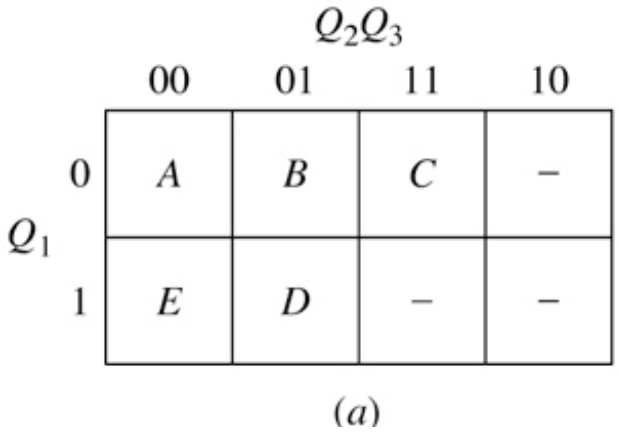
E->A

3 state bits needed

Assign codes to each state

A minimum state locus assignment for the ASM chart of Fig. 8.21. (*a*) State-assignment map. (*b*) State locus.

Figure 8.22



(*a*)

State transition $A \longrightarrow B = 000 \longrightarrow 001$ 1 bit change
State transition $B \longrightarrow A = 001 \longrightarrow 000$ 1 bit change
State transition $B \longrightarrow C = 001 \longrightarrow 011$ 1 bit change
State transition $B \longrightarrow D = 001 \longrightarrow 101$ 1 bit change
State transition $C \longrightarrow D = 011 \longrightarrow 101$ 2 bit changes
State transition $D \longrightarrow E = 101 \longrightarrow 100$ 1 bit change
State transition $E \longrightarrow A = 100 \longrightarrow 000$ 1 bit change

Total = 8 = state locus

(*b*)

Table 8.1 ASM tables for Fig. 8.21. (*a*) ASM transition table. (*b*) Assigned ASM transition table

Link path	Present state	Inputs x_1	x_2	x_3	Next state	Outputs z_1	z_2	z_3	z_4
L_1	A	0	–	–	A	1	0	0	0
L_2	A	1	–	–	B	1	1	0	0
L_3	B	0	0	–	A	0	0	0	0
L_4	B	0	1	–	C	0	1	0	0
L_5	B	1	–	–	D	0	0	1	0
L_6	C	–	–	0	C	0	0	1	1
L_7	C	–	–	1	D	0	0	1	1
L_8	D	–	–	–	E	1	0	0	0
L_9	E	–	–	–	A	0	0	1	0

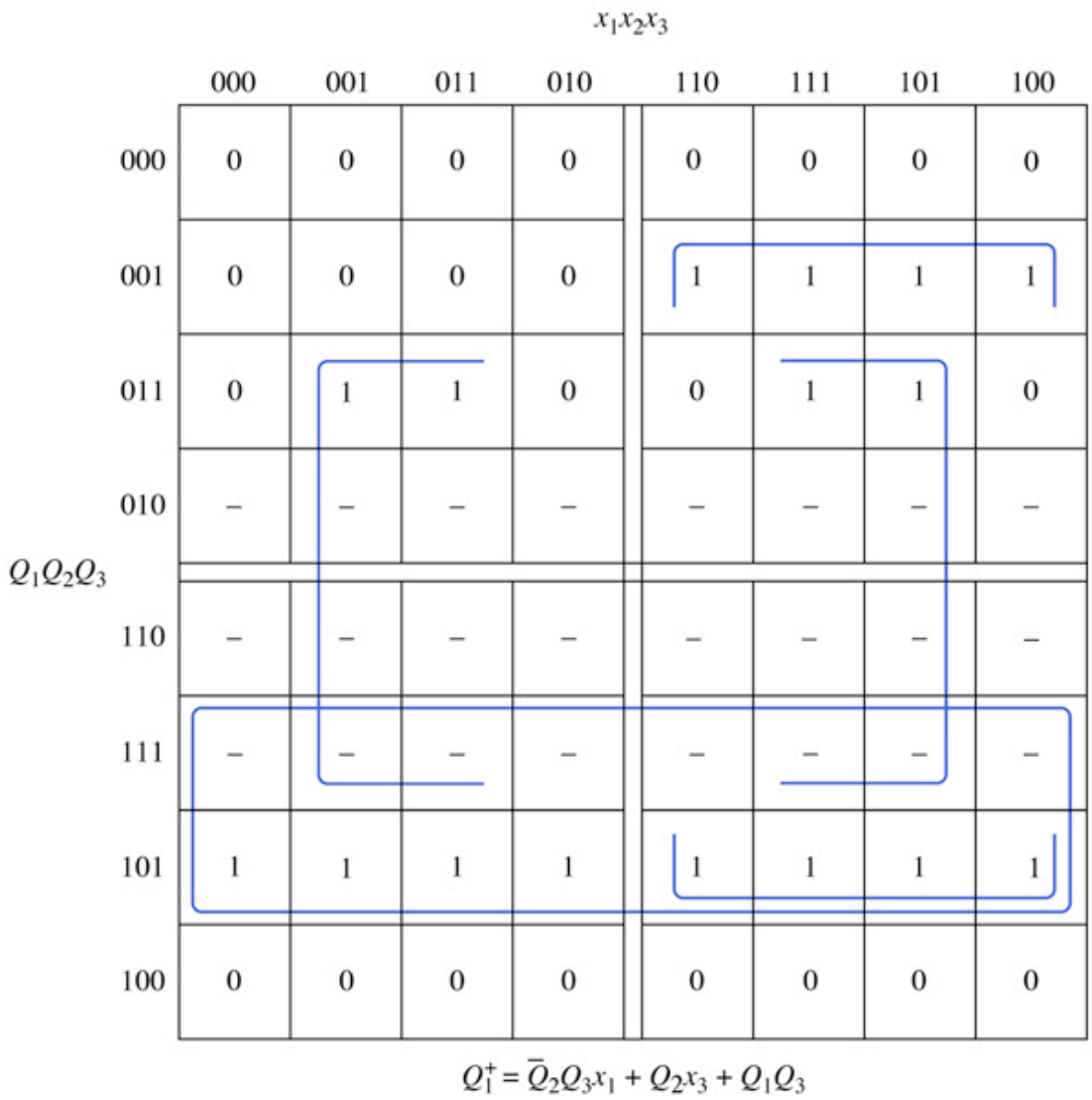
(*a*)

Link path	Present state Sym	Q_1	Q_2	Q_3	Inputs x_1	x_2	x_3	Next state Sym	Q_1^+	Q_2^+	Q_3^+	Outputs z_1	z_2	z_3	z_4
L_1	A	0	0	0	0	–	–	A	0	0	0	1	0	0	0
L_2	A	0	0	0	1	–	–	B	0	0	1	1	1	0	0
L_3	B	0	0	1	0	0	–	A	0	0	0	0	0	0	0
L_4	B	0	0	1	0	1	–	C	0	1	1	0	1	0	0
L_5	B	0	0	1	1	–	–	D	1	0	1	0	0	1	0
L_6	C	0	1	1	–	–	0	C	0	1	1	0	0	1	1
L_7	C	0	1	1	–	–	1	D	1	0	1	0	0	1	1
L_8	D	1	0	1	–	–	–	E	1	0	0	1	0	0	0
L_9	E	1	0	0	–	–	–	A	0	0	0	0	0	1	0

(*b*)

Karnaugh map for simplifying the Q_1^+ function of Table 8.1*b*.

Figure 8.23



ASM Excitation Table defines FF input equations

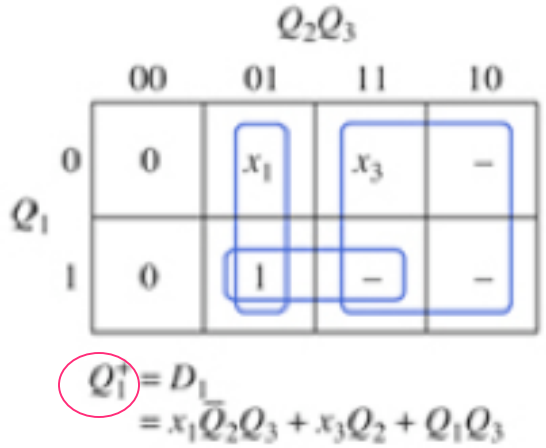
Table 8.5 ASM excitation table for Table 8.1*b*

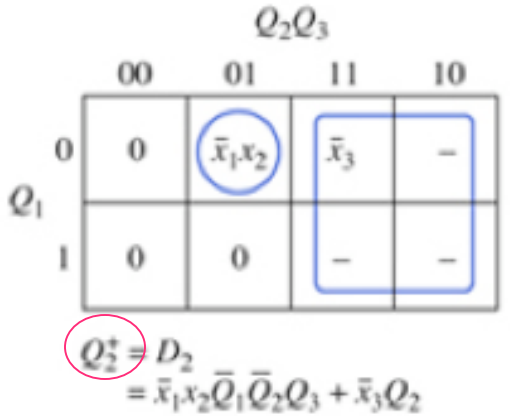
Link path	Present state				Inputs			Next state				Excitations						Outputs			
	Sym	Q_1	Q_2	Q_3	x_1	x_2	x_3	Sym	Q_1^+	Q_2^+	Q_3^+	J_1	K_1	J_2	K_2	J_3	K_3	z_1	z_2	z_3	z_4
L_1	A	0	0	0	0	–	–	A	0	0	0	0	–	0	–	0	–	1	0	0	0
L_2	A	0	0	0	1	–	–	B	0	0	1	0	–	0	–	1	–	1	1	0	0
L_3	B	0	0	1	0	0	–	A	0	0	0	0	–	0	–	–	1	0	0	0	0
L_4	B	0	0	1	0	1	–	C	0	1	1	0	–	1	–	–	0	0	1	0	0
L_5	B	0	0	1	1	–	–	D	1	0	1	1	–	0	–	–	0	0	0	1	0
L_6	C	0	1	1	–	–	0	C	0	1	1	0	–	–	0	–	0	0	0	1	1
L_7	C	0	1	1	–	–	1	D	1	0	1	1	–	–	1	–	0	0	0	1	1
L_8	D	1	0	1	–	–	–	E	1	0	0	–	0	0	–	–	1	1	0	0	0
L_9	E	1	0	0	–	–	–	A	0	0	0	–	1	0	–	0	–	0	0	1	0

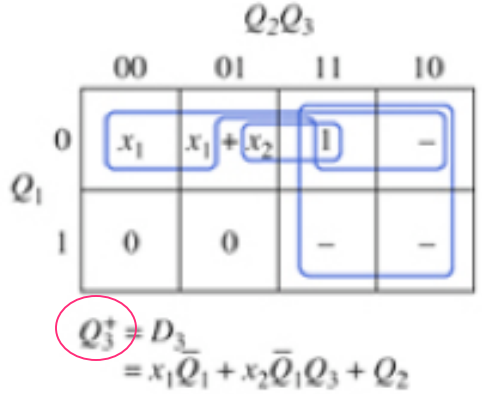
Using variable-entered Karnaugh maps to obtain a discrete-gate realization with clocked *D* flip-flops

Link path	Present state Sym	Q_1	Q_2	Q_3	Inputs x_1	x_2	x_3	Next state Sym	Q_1^+	Q_2^+	Q_3^+	Outputs z_1	z_2	z_3	z_4
L_1	A	0	0	0	0	–	–	A	0	0	0	1	0	0	0
L_2	A	0	0	0	1	–	–	B	0	0	1	1	1	0	0
L_3	B	0	0	1	0	0	–	A	0	0	0	0	0	0	0
L_4	B	0	0	1	0	1	–	C	0	1	1	0	1	0	0
L_5	B	0	0	1	1	–	–	D	1	0	1	0	0	1	0
L_6	C	0	1	1	–	–	0	C	0	1	1	0	0	1	1
L_7	C	0	1	1	–	–	1	D	1	0	1	0	0	1	1
L_8	D	1	0	1	–	–	–	E	1	0	0	1	0	0	0
L_9	E	1	0	0	–	–	–	A	0	0	0	0	0	1	0

(*b*)

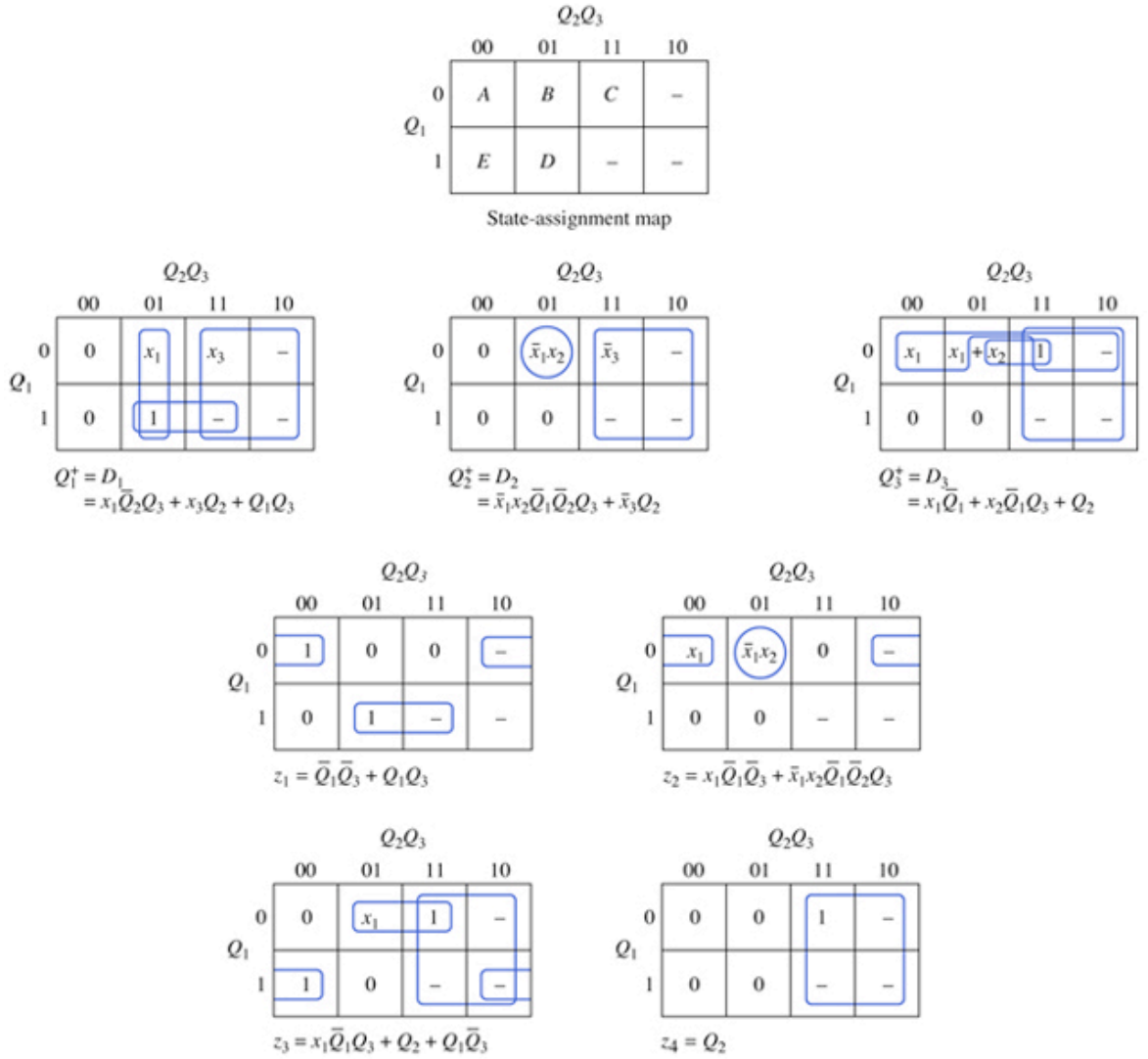




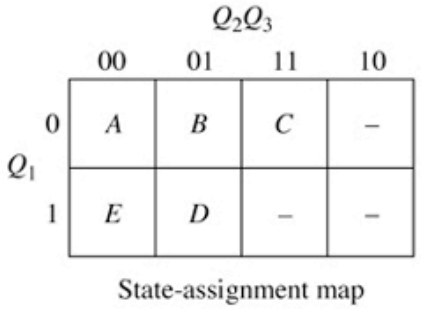


Using variable-entered Karnaugh maps to obtain a discrete-gate realization with clocked *D* flip-flops for the ASM chart of Fig. 8.21.

Figure 8.25

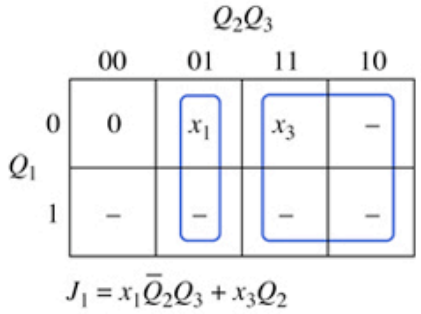


Using variable-entered Karnaugh maps to obtain a discrete-gate realization with clocked *JK* flip-flops for the ASM chart of Fig. 8.21.



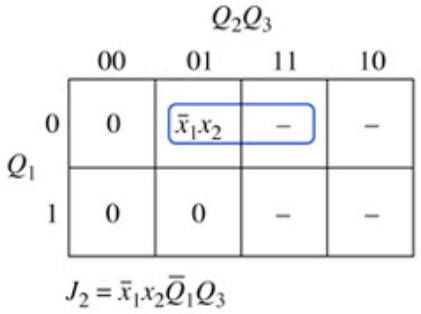
Q_1 \ Q_2Q_3	00	01	11	10
0	A	B	C	–
1	E	D	–	–

State-assignment map



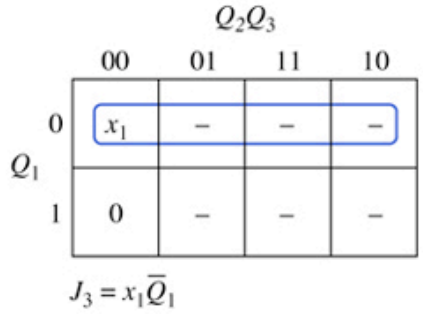
Q_1 \ Q_2Q_3	00	01	11	10
0	0	x_1	x_3	–
1	–	–	–	–

$J_1 = x_1\bar{Q}_2Q_3 + x_3Q_2$



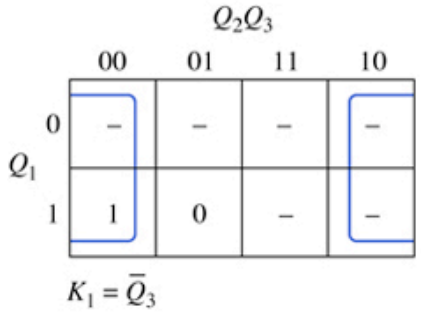
Q_1 \ Q_2Q_3	00	01	11	10
0	0	$\bar{x}_1x_2$	–	–
1	0	0	–	–

$J_2 = \bar{x}_1x_2\bar{Q}_1Q_3$



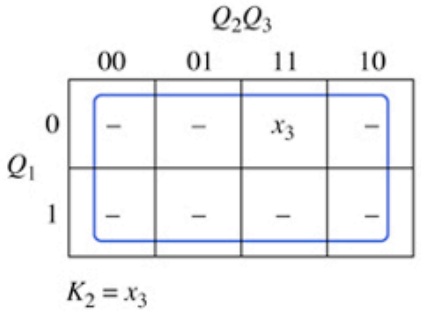
Q_1 \ Q_2Q_3	00	01	11	10
0	x_1	–	–	–
1	0	–	–	–

$J_3 = x_1\bar{Q}_1$



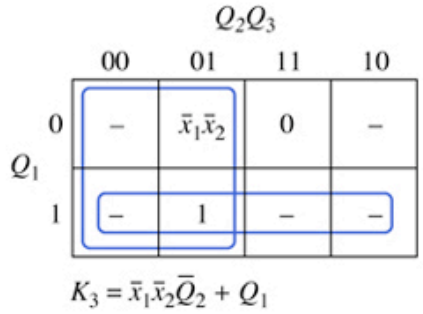
Q_1 \ Q_2Q_3	00	01	11	10
0	–	–	–	–
1	1	0	–	–

$K_1 = \bar{Q}_3$



Q_1 \ Q_2Q_3	00	01	11	10
0	–	–	x_3	–
1	–	–	–	–

$K_2 = x_3$



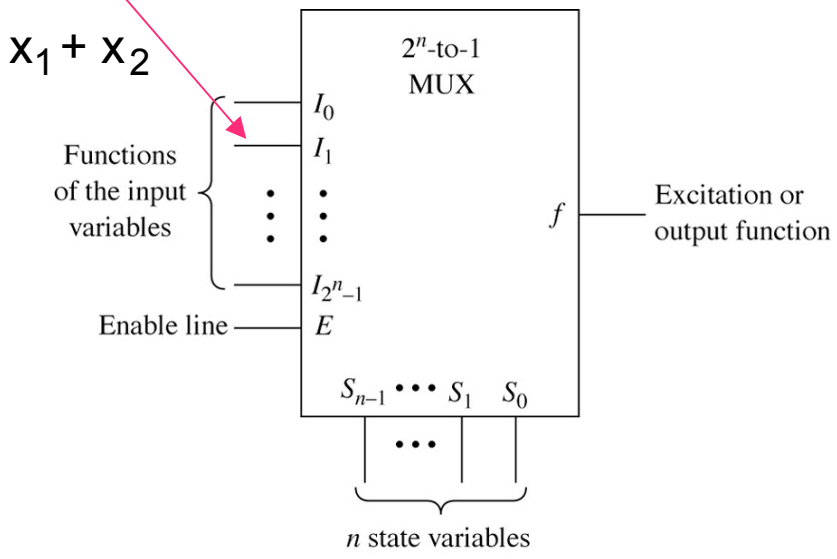
Q_1 \ Q_2Q_3	00	01	11	10
0	–	$\bar{x}_1\bar{x}_2$	0	–
1	–	1	–	–

$K_3 = \bar{x}_1\bar{x}_2\bar{Q}_2 + Q_1$

Assignment of inputs to a multiplexer for each excitation and output function.

Link path	Present state Sym	Q_1	Q_2	Q_3	Inputs x_1	x_2	x_3	Next state Sym	Q_1^+	Q_2^+	Q_3^+	Outputs z_1	z_2	z_3	z_4
L_1	A	0	0	0	0	–	–	A	0	0	0	1	0	0	0
L_2	A	0	0	0	1	–	–	B	0	0	1	1	1	0	0
L_3	B	0	0	1	0	0	–	A	0	0	0	0	0	0	0
L_4	B	0	0	1	0	1	–	C	0	1	1	0	1	0	0
L_5	B	0	0	1	1	–	–	D	1	0	1	0	0	1	0
L_6	C	0	1	1	–	–	0	C	0	1	1	0	0	1	1
L_7	C	0	1	1	–	–	1	D	1	0	1	0	0	1	1
L_8	D	1	0	1	–	–	–	E	1	0	0	1	0	0	0
L_9	E	1	0	0	–	–	–	A	0	0	0	0	0	1	0

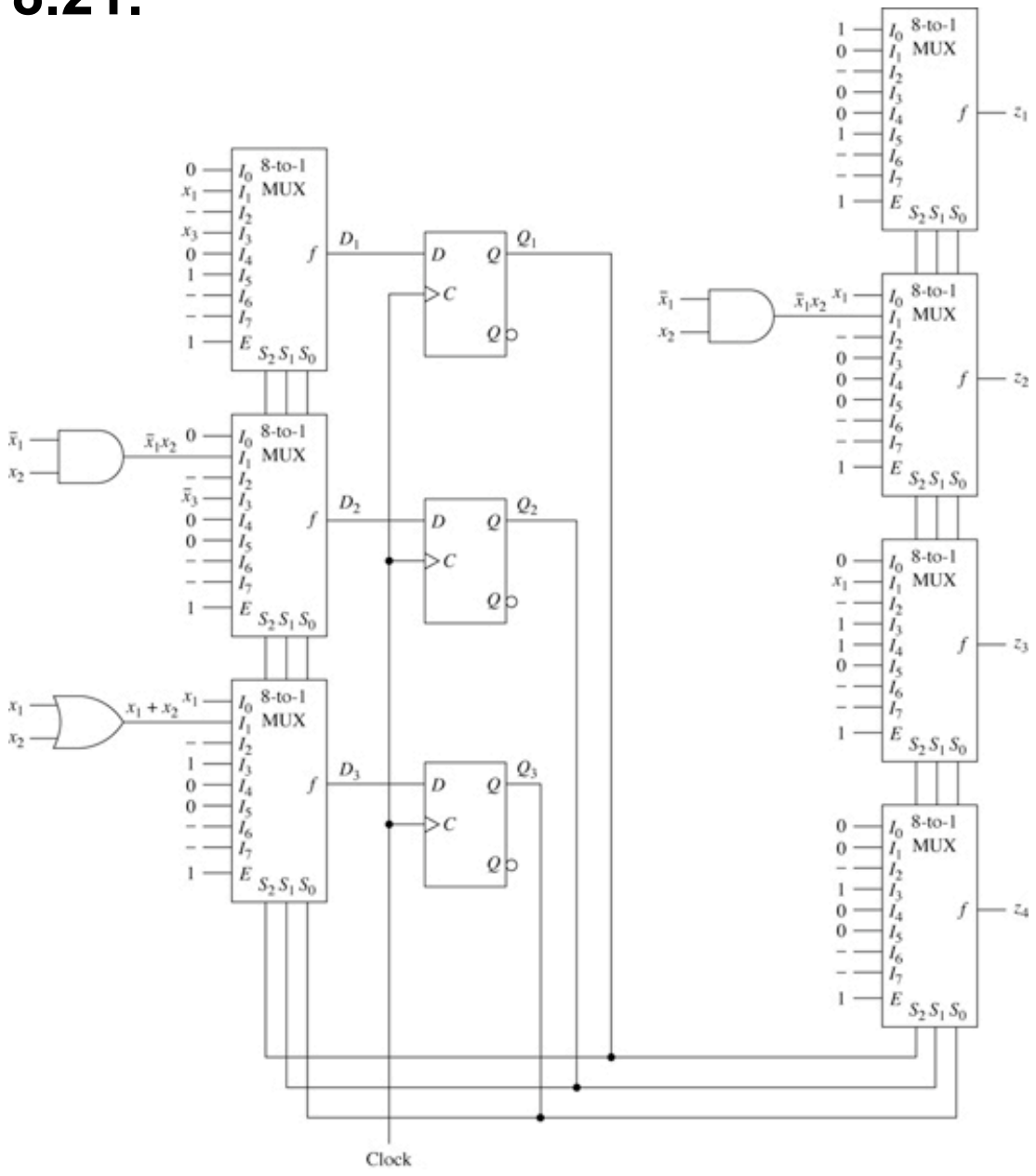
(b)



One mux per function

Each input function corresponds to cell function in variable-entered K-map

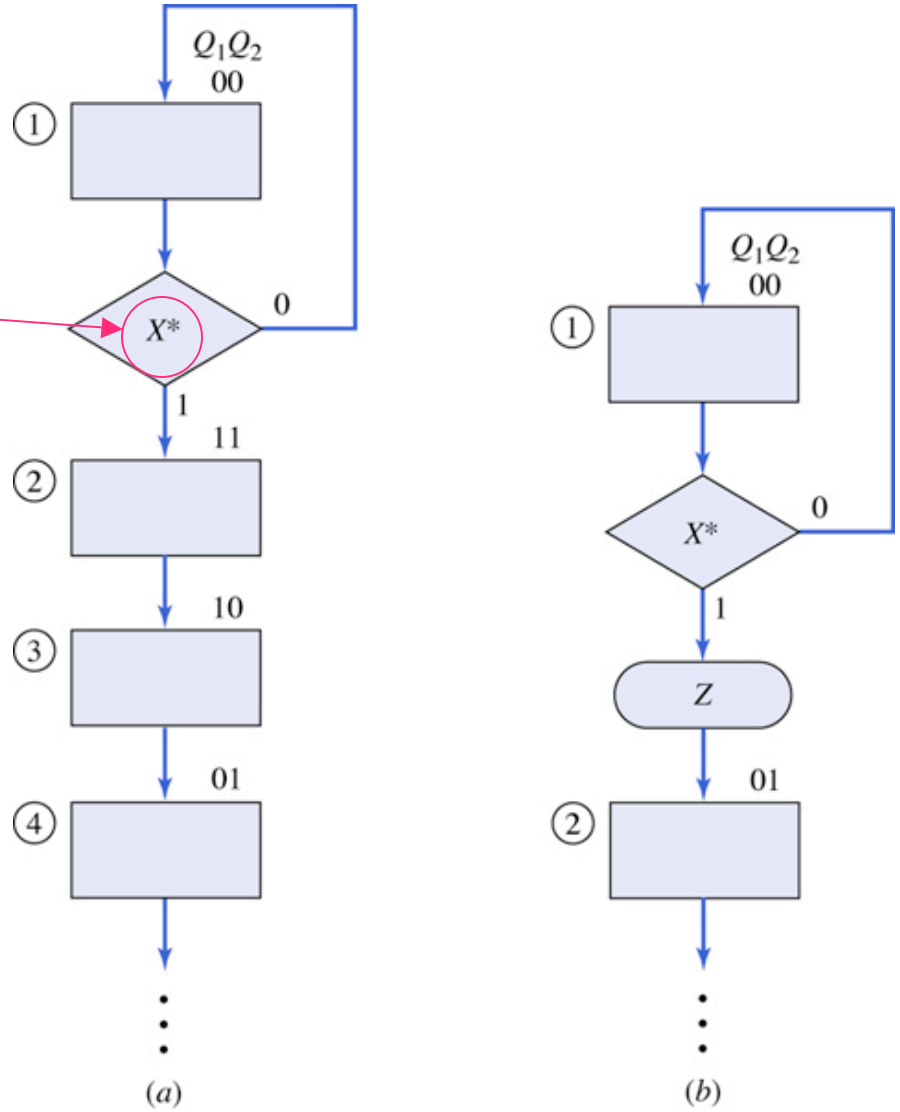
Multiplexer realization with clocked *D* flip-flops for the ASM chart of Fig. 8.21.



Fragments of ASM charts illustrating problems associated with asynchronous inputs. (*a*) Transition race. (*b*) Output race.

Figure 8.31

* Means asychronous w.r.t. ASM clock



Using a clocked *D* flip-flop to synchronize an asynchronous input.

Figure 8.32