SPECIFICATION OF SEQUENTIAL SYSTEMS

- SYNCHRONOUS SEQUENTIAL SYSTEMS
- TWO TYPES: MEALY AND MOORE MACHINES
- TIME BEHAVIOR: I/O SEQUENCES
- STATE TABLE AND STATE DIAGRAM
- STATE MINIMIZATION

CS MSIA

SEQUENTIAL SYSTEMS

EXAMPLE: SERIAL ADDER

INPUT: $x(t) \in \{0, 1, \ldots, 9\}$

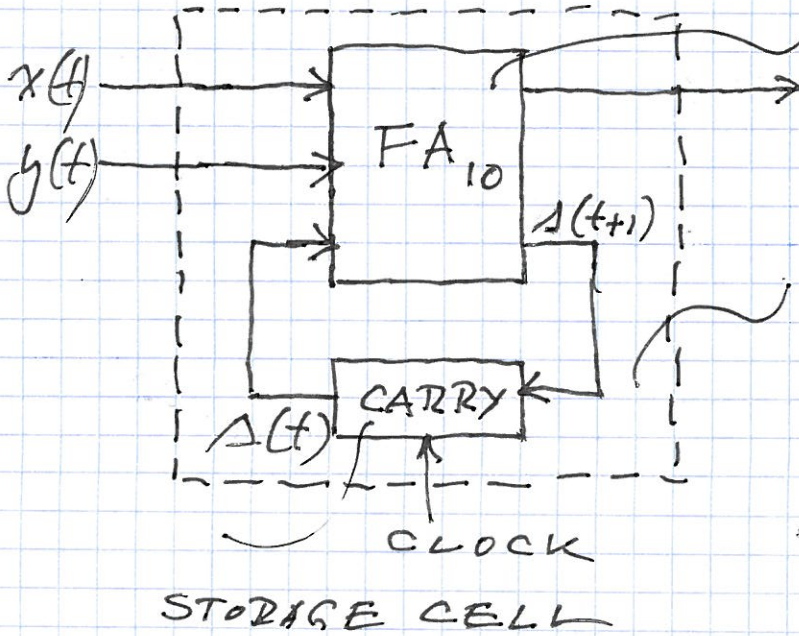
$y(t) \in \{0, 1, \ldots, 9\}$

STATE: $s(t) \in \{0, 1\}$

OUTPUT: $z(t) = (x(t) + y(t) + \text{PRESENT\_CARRY } s(t)) \bmod 10$

$w(t)$

NEXT_CARRY $s(t+1) = \begin{cases} 1 & \text{IF } w(t) \geq 10 \\ 0 & \text{OTHERWISE} \end{cases}$



SERIAL ADDER (SEQ. SYSTEM)

$s(t)$ – STATE OF THE SYSTEM

DEFINITION

$$z(t) = F(x(0,t))$$

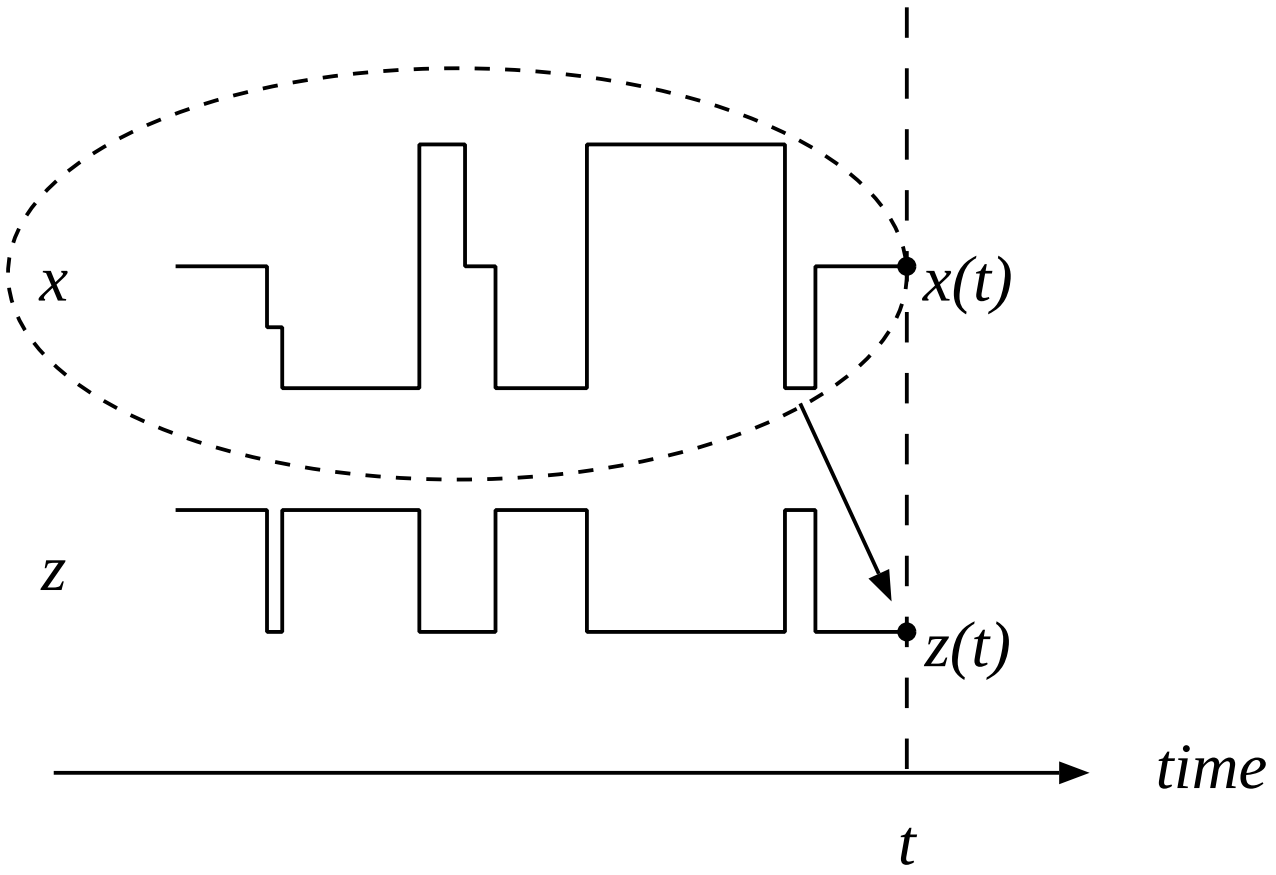


Figure 7.1: INPUT AND OUTPUT TIME FUNCTIONS.

SYNCHRONOUS AND ASYNCHRONOUS SYSTEMS

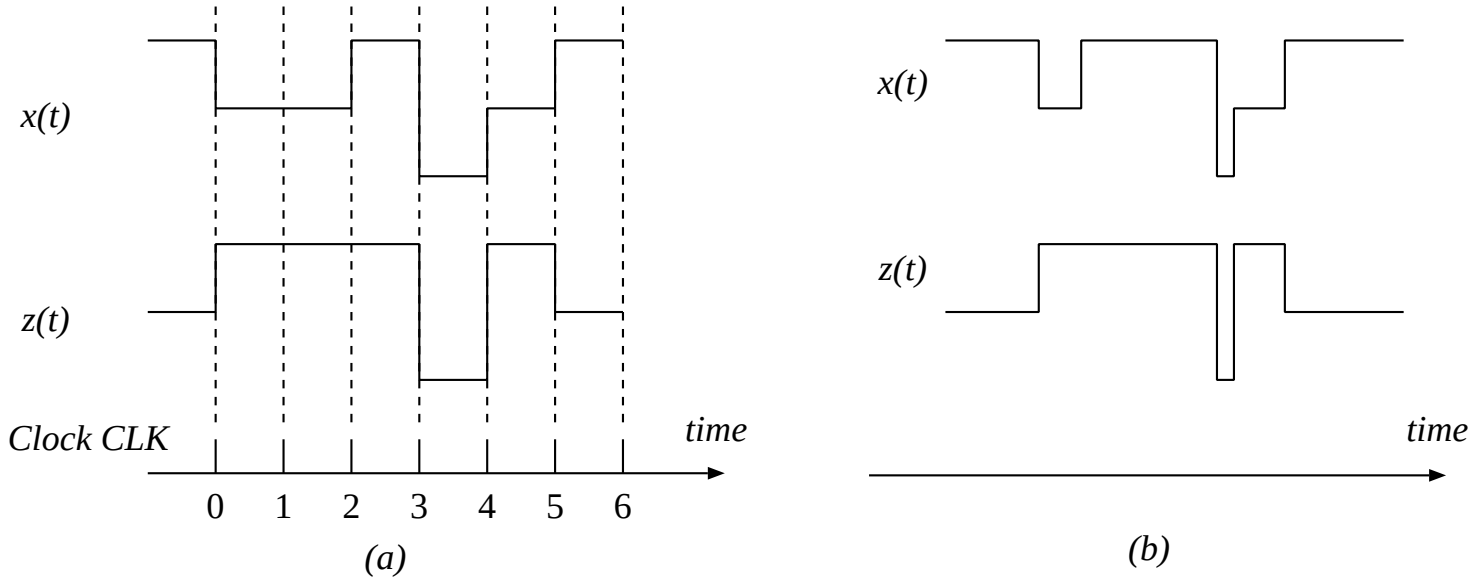


Figure 7.2: a) SYNCHRONOUS BEHAVIOR. b) ASYNCHRONOUS BEHAVIOR.

- CLOCK
- I/O SEQUENCE $x(t_1, t_2)$

$$x(2,5) = aabc$$

$$z(2,5) = 1021$$

Example 7.1: SERIAL DECIMAL ADDER

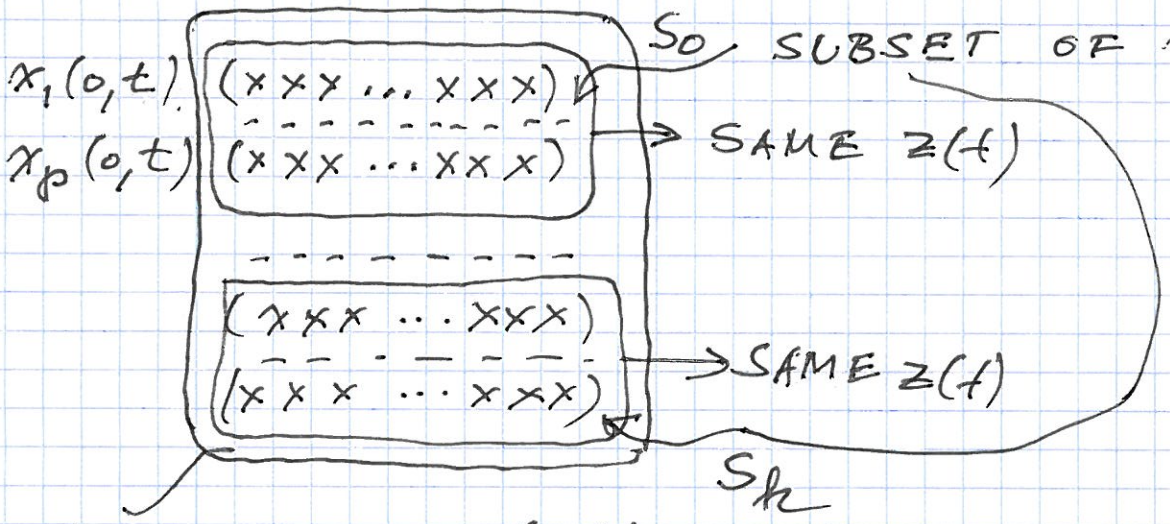
x	1638753
y	3652425
z	5291178

- LEAST-SIGNIFICANT DIGIT FIRST (at t=0)

t	0	1	2	3	4	5	6
x(t)	3	5	7	8	3	6	1
y(t)	5	2	4	2	5	6	3
z(t)	8	7	1	1	9	2	5

CS MSIA A KEY IDEA: STATE

$z(t) = f(x(0,t))$ – INTRACTABLE



a) A SET OF ALL INPUT SEQUENCES HAS A <u>FINITE</u> NUMBER OF SUBSETS S_k

b) SEQUENCES WITH THE SAME EFFECT ON $z(t)$, $t \geq t_0$ BELONG TO THE <u>SAME CLASS</u> (SUBSET)

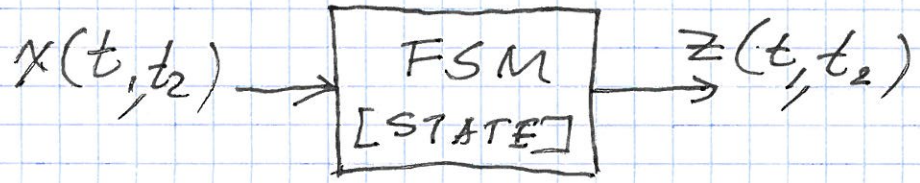
c) THE CLASS KEPT IN AN AUXILLIARY VARIABLE S CALLED <u>STATE</u>

IF SYSTEM AT TIME t IS IN STATE $S(t) = S_k$ THAN THE INPUT SEQUENCE BELONGS TO SUBSET (CLASS) S_k

CS MS1A STATE: (CONT,)

— IF THE NUMBER OF CLASSES (STATES) IS FINITE → WE HAVE A FINITE STATE SYSTEM OR FINITE STATE MACHINE (FSM)

— AN FSM IS A TRANSDUCER:
TRANSFORMS INPUT SEQUENCES INTO OUTPUT SEQUENCES, KEEPING AND UPDATING FSM STATE INTERNALLY



— TO DETERMINE THE OUTPUT & NEXT STATE, IT IS SUFFICIENT TO KNOW PRESENT STATE AND PRESENT INPUT

STATE DESCRIPTION

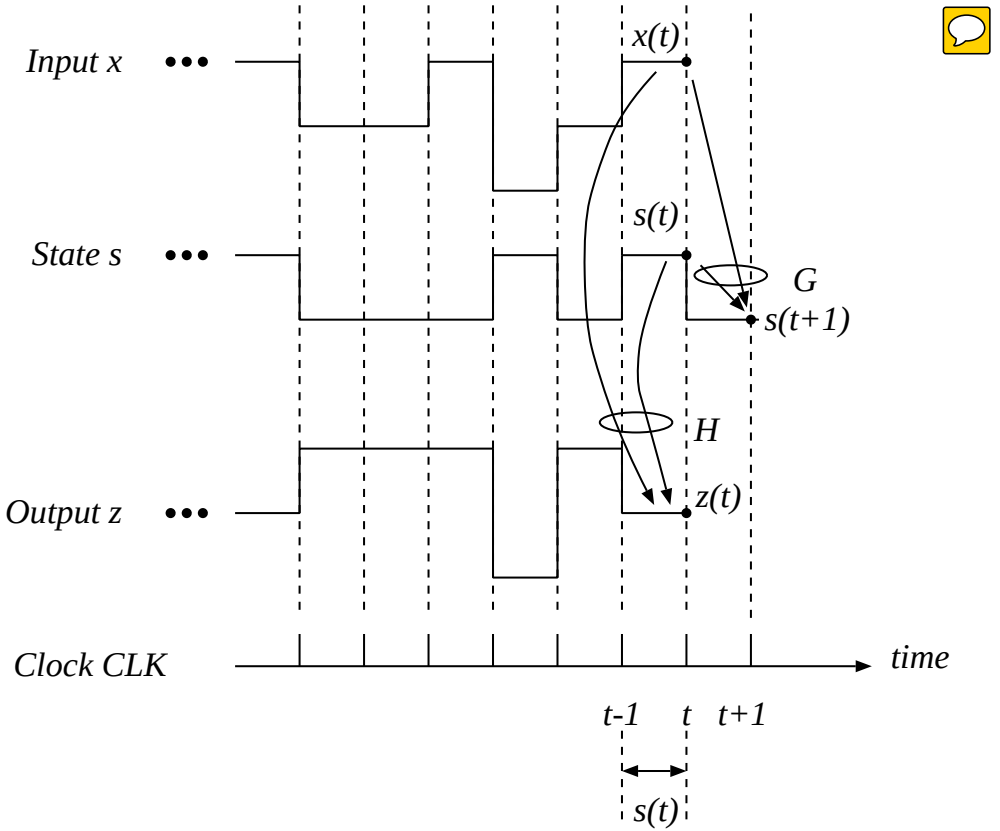


Figure 7.3: OUTPUT AND STATE TRANSITION FUNCTIONS

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

OUTPUT FUNCTION $z(t) = H(s(t), x(t))$

Example 7.3: STATE DESCRIPTION OF SERIAL ADDER

INPUT: $x(t), y(t) \in \{0, 1, ..., 9\}$
OUTPUT: $z(t) \in \{0, 1, ..., 9\}$
STATE: $s(t) \in \{0, 1\}$ (the carry)
INITIAL STATE: $s(0) = 0$

FUNCTIONS: THE TRANSITION AND OUTPUT FUNCTIONS

$$s(t+1) = \begin{cases} 1 \ \textbf{if} \ \ x(t) + y(t) + s(t) \geq 10 \\ 0 \ \textbf{otherwise} \end{cases}$$

$$z(t) = (x(t) + y(t) + s(t)) \ mod \ 10$$

CS MSIA

SEQUENTIAL SYSTEMS

EXAMPLE: SERIAL ADDER

INPUT: $x(t) \in \{0, 1, \ldots, 9\}$

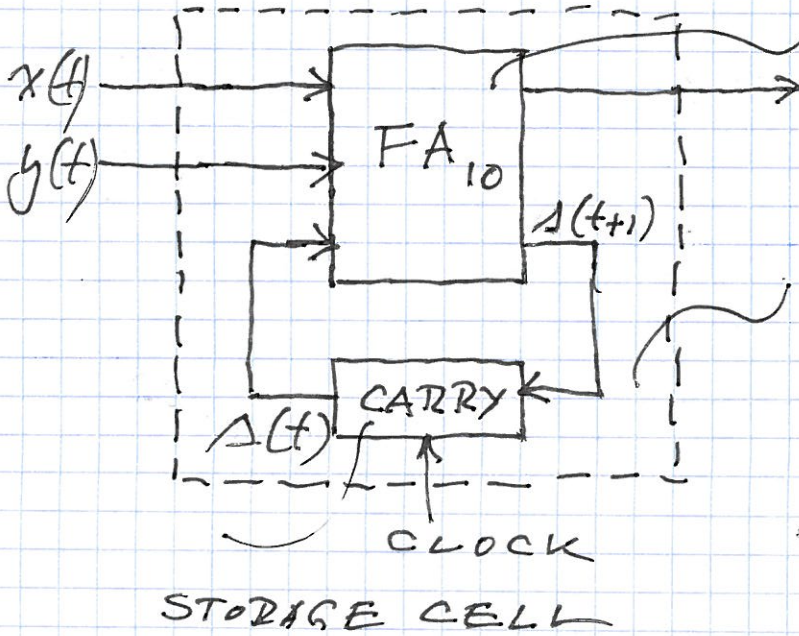
$y(t) \in \{0, 1, \ldots, 9\}$

STATE: $s(t) \in \{0, 1\}$

OUTPUT: $z(t) = (x(t) + y(t) + \text{PRESENT\_CARRY } s(t)) \bmod 10$

$w(t)$

NEXT_CARRY $s(t+1) = \begin{cases} 1 & \text{IF } w(t) \geq 10 \\ 0 & \text{OTHERWISE} \end{cases}$



SERIAL ADDER (SEQ. SYSTEM)

$s(t)$ – STATE OF THE SYSTEM

EXAMPLE OF SERIAL ADDITION

t	0	1	2	3	4	5	6
x(t)	3	5	7	8	3	6	1
y(t)	5	2	4	2	5	6	3
s(t)	0	0	0	1	1	0	1
z(t)	8	7	1	1	9	2	5

Example 7.4: ODD/EVEN

TIME-BEHAVIOR SPECIFICATION:

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

FUNCTION: $z(t) = \begin{cases} 1 \textbf{ if } \ x(0,t) \text{ contains an even number of } b'\text{s} \\ 0 \textbf{ otherwise} \end{cases}$

I/O SEQUENCE:

t	0	1	2	3	4	5	6	7
x, z	$a, 1$	$b, 0$	$b, 1$	$a, 1$	$b, 0$	$a, 0$	$b, 1$	$a, 1$

Example 7.4: STATE DESCRIPTION OF ODD/EVEN

INPUT: $x(t) \in \{a, b\}$
OUTPUT: $z(t) \in \{0, 1\}$
STATE: $s(t) \in \{\text{EVEN, ODD}\}$
INITIAL STATE: s(0)= EVEN

FUNCTIONS: TRANSITION AND OUTPUT FUNCTIONS

PS	$x(t) = a$	$x(t) = b$
EVEN	EVEN, 1	ODD, 0
ODD	ODD, 0	EVEN, 1
	$NS,\ z(t)$	

MEALY AND MOORE MACHINES

Mealy machine

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

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

Moore machine

$$z(t) = H(s(t))$$

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

- EQUIVALENT IN CAPABILITIES

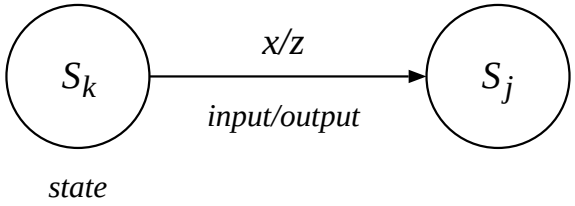
Example 7.5: MOORE SEQUENTIAL SYSTEM

INPUT: $x(t) \in \{a, b, c\}$
OUTPUT: $z(t) \in \{0, 1\}$
STATE: $s(t) \in \{S_0, S_1, S_2, S_3\}$
INITIAL STATE: $s(0) = S_0$

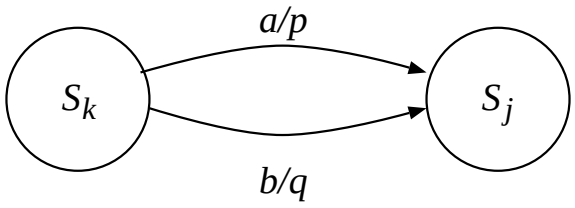
FUNCTIONS: TRANSITION AND OUTPUT FUNCTIONS:

PS	Input			
	a	b	c	
S_0	S_0	S_1	S_1	0
S_1	S_2	S_0	S_1	1
S_2	S_2	S_3	S_0	1
S_3	S_0	S_1	S_2	0
	NS			Output

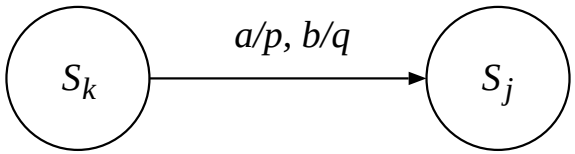
REPRESENTATION OF STATE-TRANSITION AND OUTPUT FUNCTIONS WITH STATE DIAGRAM



(a)



Complete state diagram



Simplified state diagram

(b)

Figure 7.4: (a) STATE DIAGRAM REPRESENTATION. (b) SIMPLIFIED STATE DIAGRAM NOTATION.

Example 7.6

FUNCTIONS: THE TRANSITION AND OUTPUT FUNCTIONS

$s(t)$	$x(t)$	
	a	b
S_0	S_1, p	S_2, q
S_1	S_1, p	S_0, p
S_2	S_1, p	S_2, p
	$s(t+1), z(t)$	

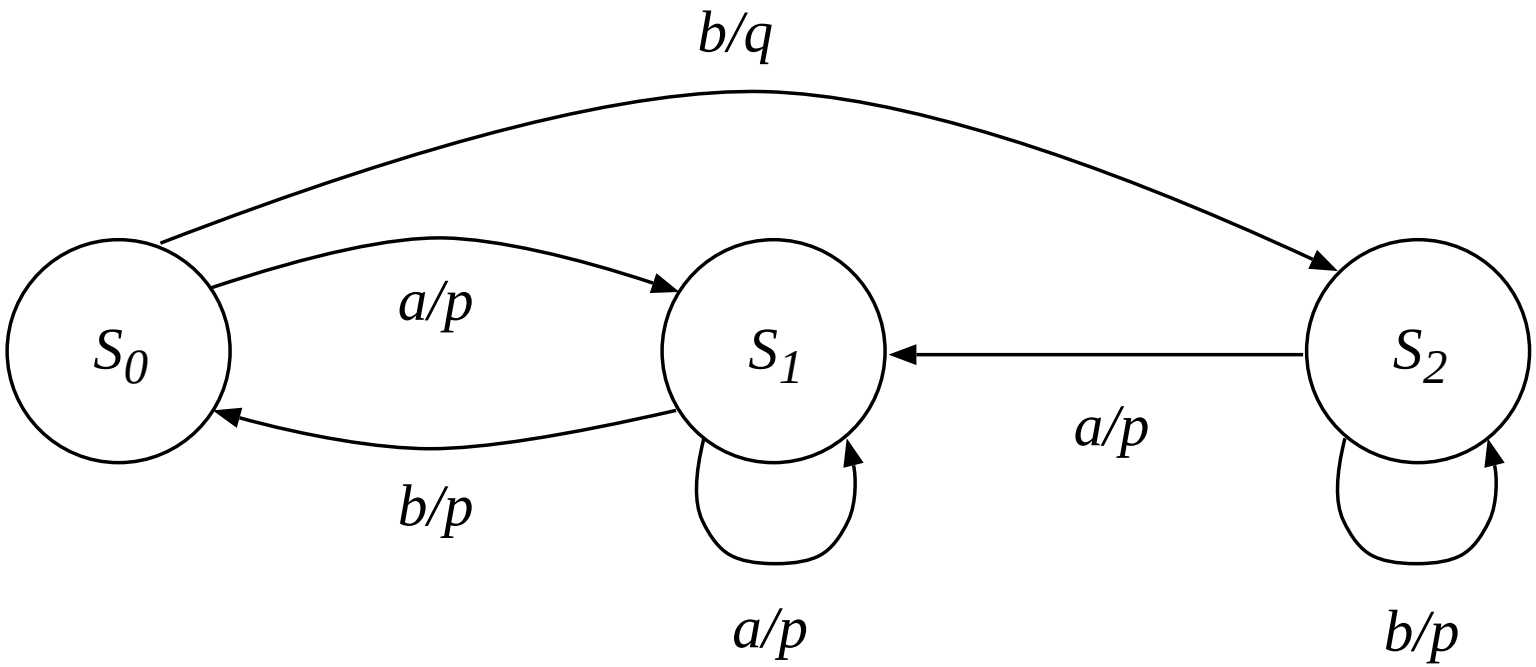


Figure 7.5: STATE DIAGRAM FOR EXAMPLE 7.6.

STATE DIAGRAM FOR A MOORE MACHINE

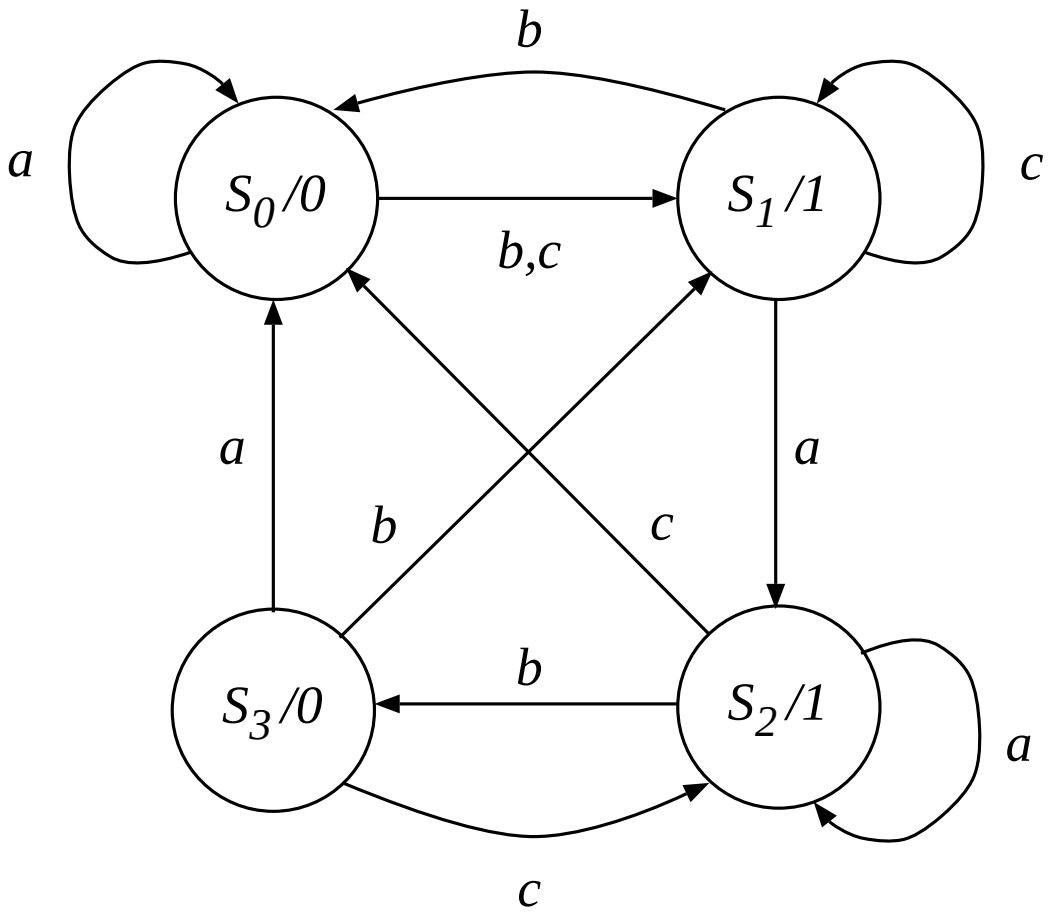


Figure 7.6: STATE DIAGRAM FOR EXAMPLE 7.5

Example 7.7: USE OF CONDITIONAL EXPRESSIONS

INPUT: $x(t) \in \{0, 1, 2, 3\}$
OUTPUT: $z(t) \in \{a, b\}$
STATE: $s(t) \in \{S_0, S_1\}$
INITIAL STATE: $s(0) = S_0$

FUNCTIONS: THE TRANSITION AND OUTPUT FUNCTIONS

$$s(t+1) = \begin{cases} S_0 \textbf{ if } & (s(t) = S_0 \\ & \quad \textbf{and } [x(t) = 0 \textbf{ or } x(t) = 2]) \\ & \quad \textbf{or } (s(t) = S_1 \textbf{ and } x(t) = 3) \\ S_1 \textbf{ otherwise} & \end{cases}$$

$$z(t) = \begin{cases} a \textbf{ if } & s(t) = S_0 \\ b \textbf{ if } & s(t) = S_1 \end{cases}$$

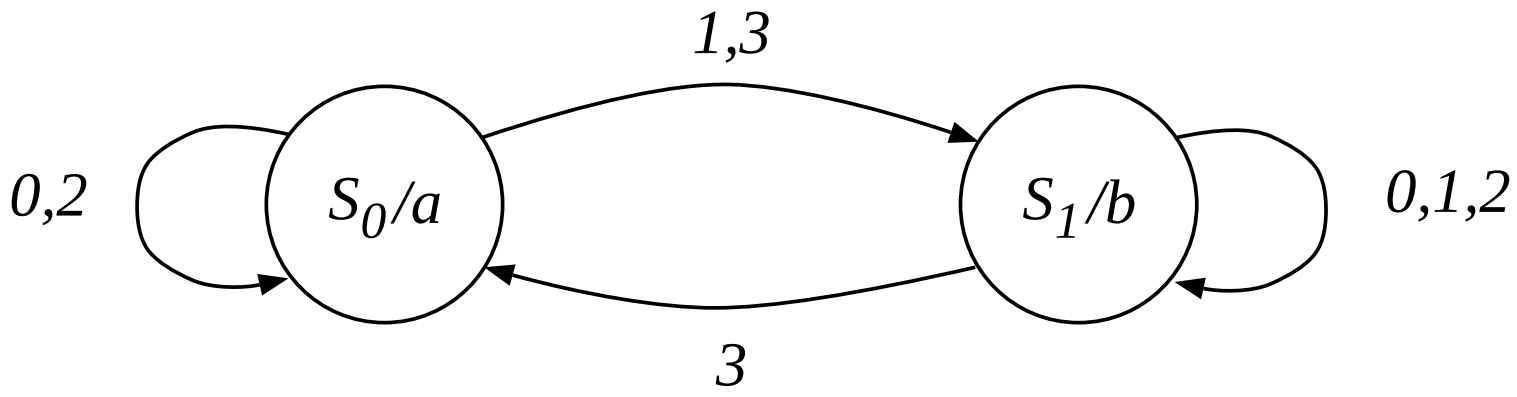


Figure 7.7: STATE DIAGRAM FOR EXAMPLE 7.7

STATE NAMES

Example 7.8: INTEGERS AS STATE NAMES

A MODULO-64 COUNTER

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

OUTPUT: $z(t) \in \{0, 1, 2, \ldots, 63\}$

STATE: $s(t) \in \{0, 1, 2, \ldots, 63\}$

INITIAL STATE: $s(0) = 0$

FUNCTIONS: THE TRANSITION AND OUTPUT FUNCTIONS

$$s(t+1) = [s(t) + x(t)] \bmod 64$$
$$z(t) = s(t)$$

Example 7.9: VECTORS AS STATE NAMES

INPUT: $e(t) \in \{1, 2, \ldots, 55\}$

OUTPUT: $z(t) \in \{0, 1, 2, \ldots, 55\}$

STATE: $\underline{s}(t) = (s_{55}, \ldots, s_1), \quad s_i \in \{0, 1, 2, \ldots, 99\}$

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

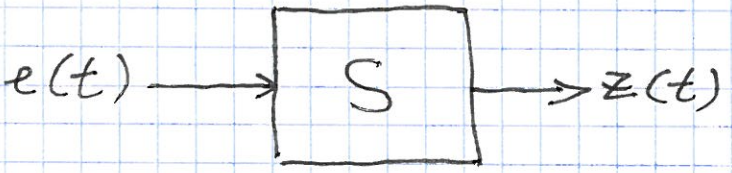
FUNCTIONS: THE TRANSITION AND OUTPUT FUNCTIONS

$$s_i(t+1) = \begin{cases} [s_i(t) + 1] \bmod 100 & \textbf{if} \quad e(t) = i \\ & \qquad i = 1, 2, \ldots, 55 \\ s_i(t) & \textbf{otherwise} \end{cases}$$

$$z(t) = \begin{cases} i \ \textbf{if} \quad e(t) = i \ \textbf{and} \ s_i(t) = 99 \\ 0 \ \textbf{otherwise} \end{cases}$$

CS MSIA EXAMPLE 7.9

A SEQUENTIAL SYSTEM KEEPING TRACK OF EVENTS



55 EVENTS: $e(t) \in \{1, 2, \ldots, 55\}$

OUTPUT:

- INDEX OF EVENT HAPPENING 100 TIMES

$$z(t) = \begin{cases} i & \text{IF } e(t)=i \ \& \ \Delta_i(t)=99 \\ 0 & \text{OTHERWISE} \end{cases}$$

$z(t) \in \{0, 1, 2, \ldots, 55\}$

t	$e(t)$	Δ_{55}	Δ_{54}	.....	Δ_{20}	....	Δ_2	Δ_1	$z(t)$
0	2	0	0		0		0	0	0
1	20	0	0		0		1	0	0
2	2	0	0		1		1	0	0
3	2	0	0		1		2	0	0
4	54	0	0		1		3	0	0
5	1	0	1		1		3	0	0
6	15	0	1		1		3	1	0
⋮	⋮	⋮		etc.			⋮	⋮	⋮
175	(54)	17	(99)				25	13	(54)
176			0						
⋮									

- THERE ARE 55×100 STATES
- IF SINGLE STATE VARIABLE $\Delta(t) \in \{0, \ldots, 5499\}$

$\Rightarrow$ HARD TO MANAGE! TRY :)

$e(175) = 54$
$\Delta_{54}(175) = 99$ } 100×

$z(175) = 54$

$\Delta_{54}(176) = 0$

TIME BEHAVIOR AND FINITE-STATE MACHINES

- STATE DESCRIPTION $\Rightarrow$ I/O SEQUENCE (Example 7.10)

  INITIAL STATE: $s(0) = S_2$
  FUNCTIONS: TRANSITION AND OUTPUT FUNCTIONS

PS	$x(t)$			
	a	b	c	
S_0	S_0	S_1	S_1	p
S_1	S_2	S_0	S_1	q
S_2	S_2	S_3	S_0	q
S_3	S_0	S_1	S_2	p
	NS			$z(t)$

I/O SEQUENCE

t	0	1	2	3	4
x	a	b	c	a	
s	S_2	S_2	S_3	S_2	S_2
z	q	q	p	q	

TIME BEHAVIOR ⇒ STATE DESCRIPTION

- NOT ALL TIME-BEHAVIORS ARE REALIZABLE

$$z(t) = \begin{cases} 1 \ \textbf{if} \ \ x(0,t) \text{ has same number of } 0'\text{s and } 1'\text{s} \\ 0 \ \textbf{otherwise} \end{cases}$$

$s(t) =$ DIFFERENCE BETWEEN NUMBER OF 1'S AND 0'S

$$s(t+1) = \begin{cases} s(t)+1 \ \textbf{if} \ \ x(t) = 1 \\ s(t)-1 \ \textbf{otherwise} \end{cases}$$

$$z(t) = \begin{cases} 1 \ \textbf{if} \ \ s(t) = 0 \\ 0 \ \textbf{otherwise} \end{cases}$$

$\Rightarrow$ DIFFERENCE UNBOUNDED: NOT A FINITE-STATE SYSTEM

PROCEDURE FOR OBTAINING FSM FROM TIME BEHAVIOR[23]

1. DETERMINE A SET OF STATES REPRESENTING REQUIRED EVENTS

2. DETERMINE THE TRANSITION FUNCTION

3. DETERMINE THE OUTPUT FUNCTION

- Example 7.11

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

- PATTERN DETECTOR $\Rightarrow$ DETECT SUBPATTERNS

State	indicates that
S_{init}	Initial state; also no subpattern
S_1	First symbol (1) of pattern has been detected
S_{11}	Subpattern (11) has been detected
S_{110}	Subpattern (110) has been detected

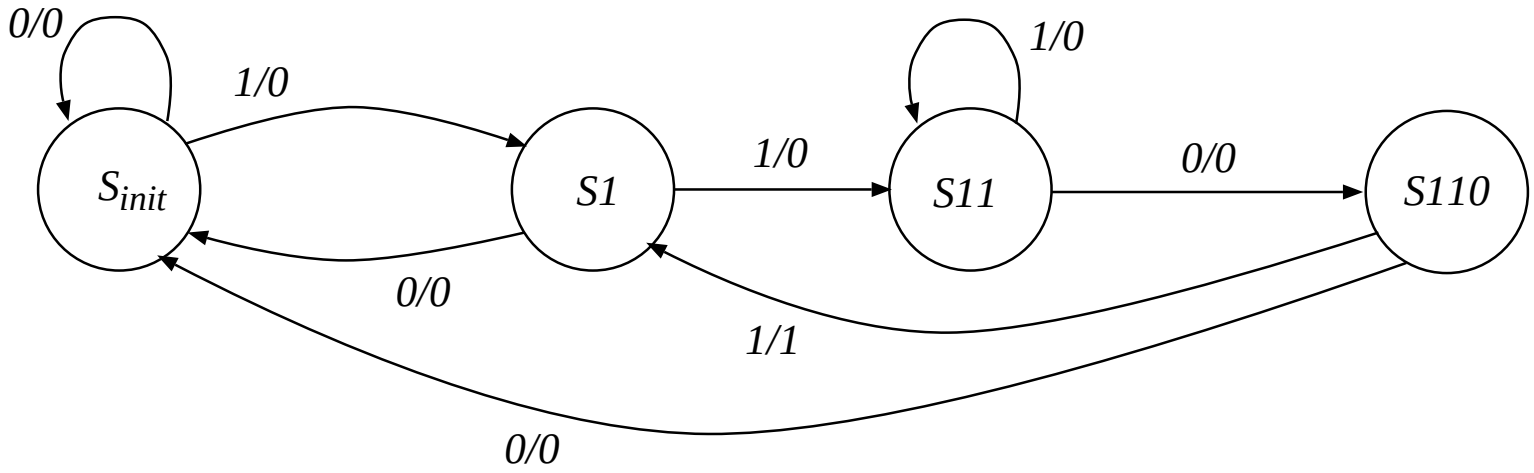


Figure 7.8: STATE DIAGRAM FOR Example 7.11

CS M51A EXAMPLE

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

FUNCTION:

$$z(t) = \begin{cases} 0 & \text{UNTIL THE FIRST INSTANCE OF 3 CONSECUTIVE 1s HAS BEEN RECEIVED} \\ x(t) & \text{AFTERWARDS} \end{cases}$$

$x(t)$	0	0	1	1	0	1	1	1	0	1	1	0	0	1	...	
$z(t)$	0	0	0	0	0	0	0	0	0	1	1	0	0	1	...	

STATES:

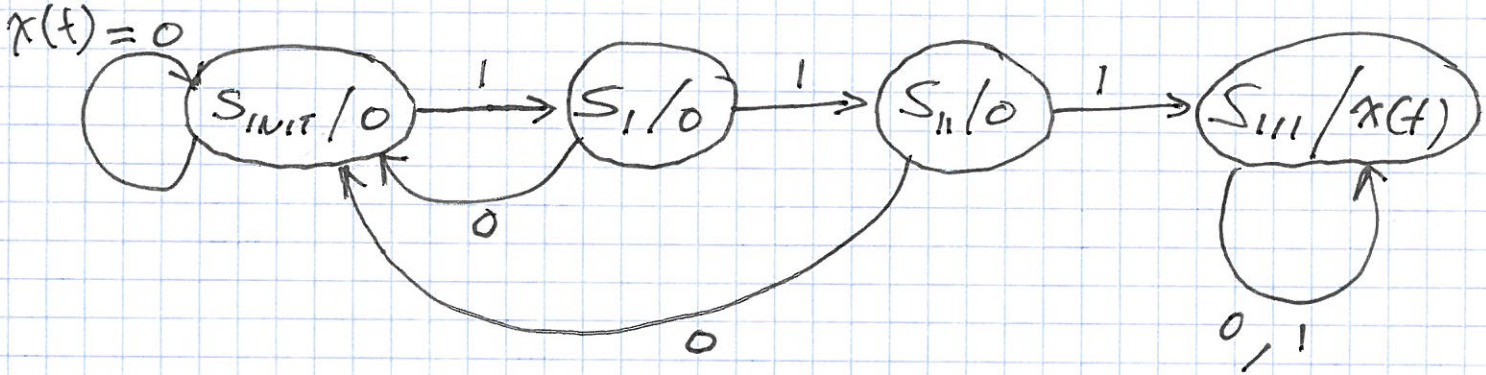
MOORE MACHINE

S_{INIT} INITIAL STATE: NO SUBPATTERN RECEIVED

S_1 FIRST "1" RECEIVED

S_{11} TWO CONSEC. 1s RECEIVED

S_{111} THREE CONSEC. 1s "



CS M51A FINITE-MEMORY SEQ. SYSTEM

$$z(t) = F(\underbrace{x(t-m+1, t)}_{\text{LAST } m \text{ INPUTS}})$$

EX. 7,12

$$z(t) = \begin{cases} p & \text{IF } x(t-3, t) = aaba \\ \xi & \text{OTHERWISE} \end{cases}$$

FINITE MEMORY OF LENGTH $m=4$

OVERLAP OK

$x(t)$	a	b	a	b	b	a	a	b	a	b	a	a	b	a	a	b	a	a	a	a	b
$z(t)$	ξ	ξ	ξ	ξ	ξ	ξ	ξ	ξ	p	ξ	ξ	ξ	ξ	p	ξ	ξ	p	ξ	ξ	ξ	ξ

ALL FM SS ARE FINITE STATE SYSTEMS

NOT ALL FINITE STATE SYSTEMS ARE FM SS

$$z(t) = \begin{cases} 1 & \text{IF \#1's IN } x(0, t) \text{ IS EVEN} \\ 0 & \text{OTHERWISE} \end{cases}$$

PS	$x(t)$ = 0	$x(t)$ = 1
EVEN	EVEN, 1	ODD, 0
ODD	ODD, 0	EVEN, 1

NS, Z

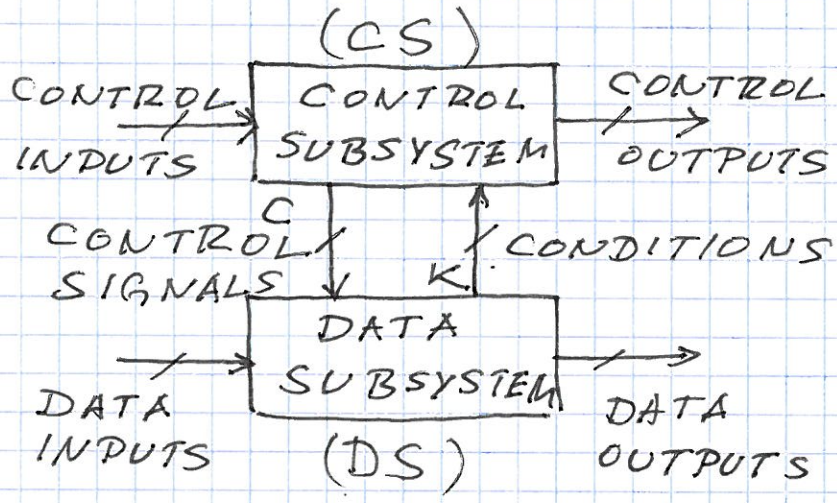
UNBOUNDED INPUT LENGTH $\neq$ FM SS

CONTROLLERS

- THE STATE DESCRIPTION IS PRIMARY
- FSM PRODUCING CONTROL SIGNALS
- CONTROL SIGNALS DETERMINE ACTIONS PERFORMED IN OTHER PARTS OF SYSTEM
- *AUTONOMOUS*: FIXED SEQUENCE OF STATES, INDEPENDENT OF INPUTS

CS M51A DIGITAL SYSTEMS: CONTROL

GENERAL DIGITAL SYSTEM

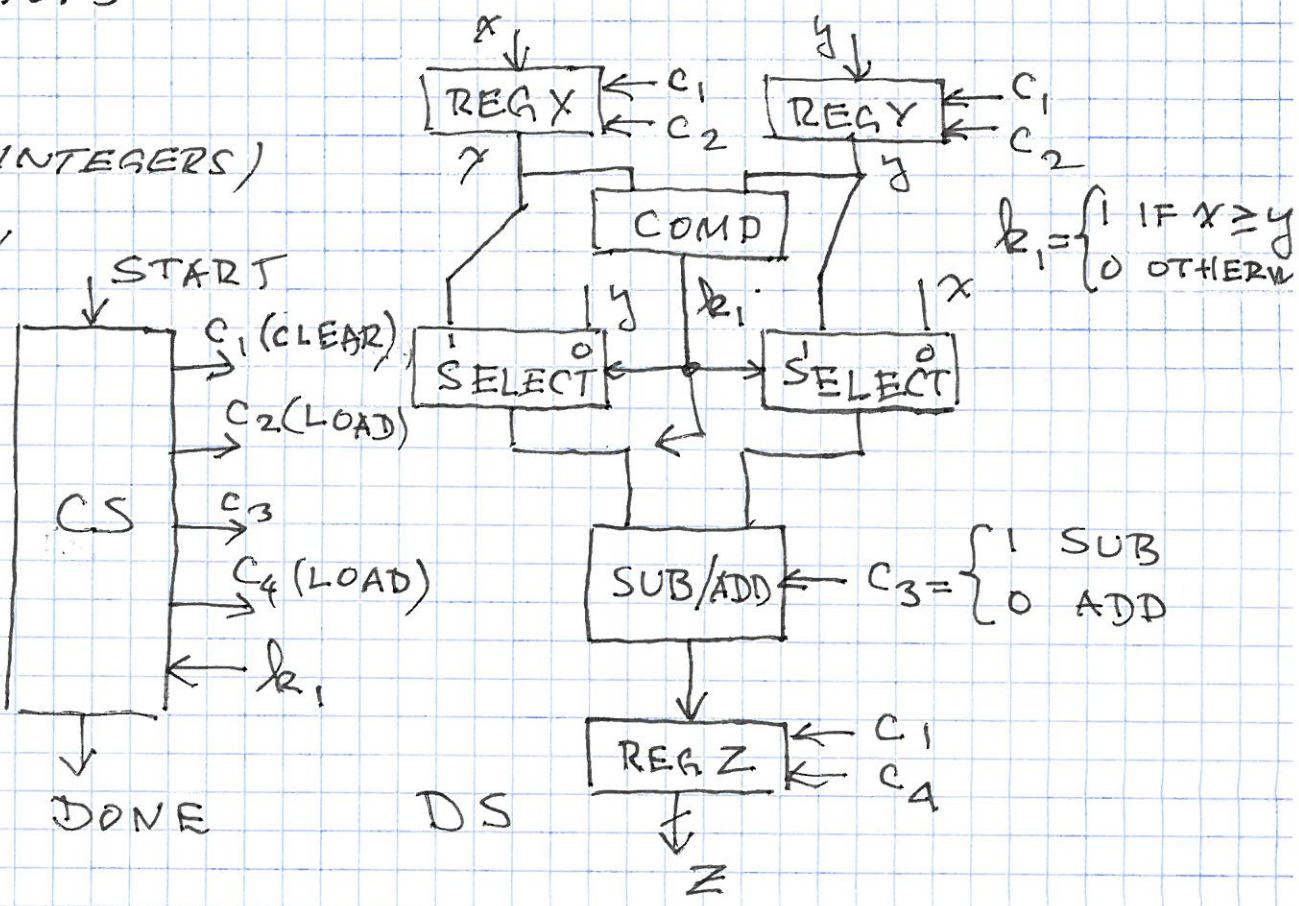


CONTROL SIGNALS $C_1, C_2 \ldots C_m$
- CONTROL ACTIONS IN DS

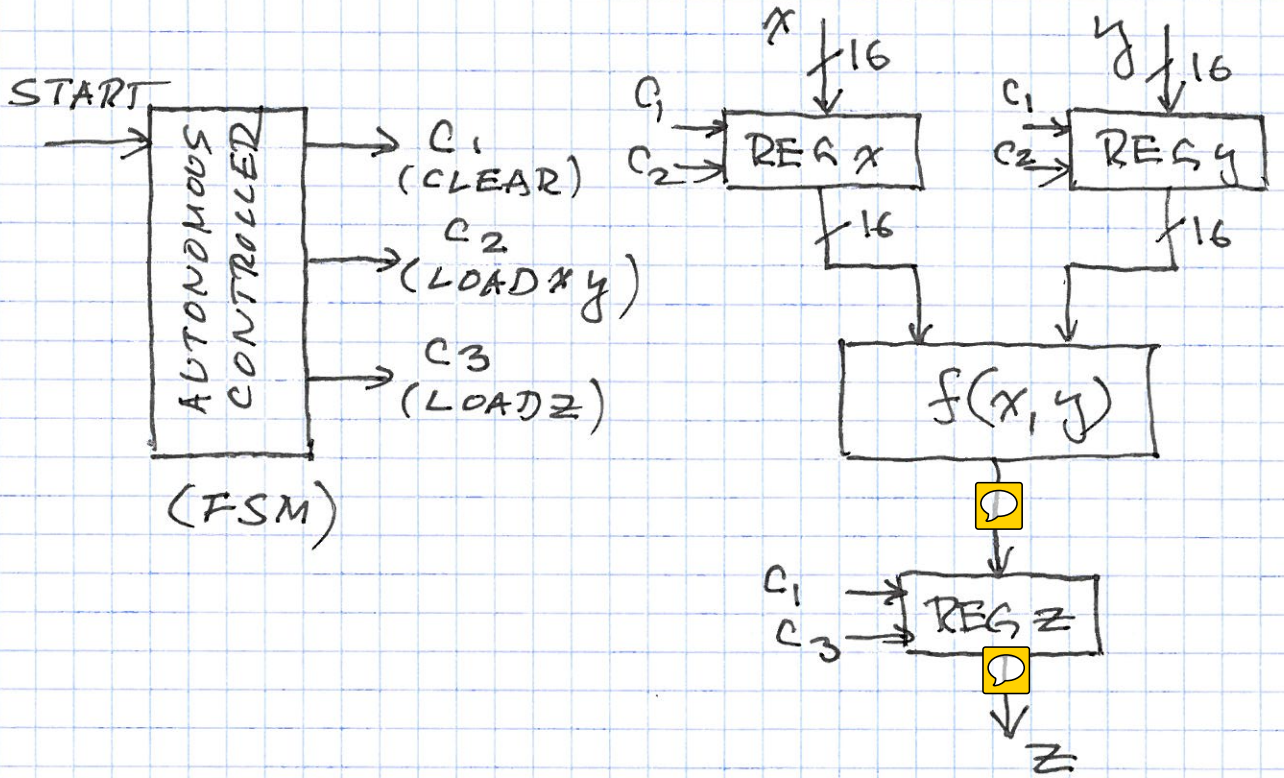
CONDITIONS $k_1, k_2, \ldots, k_p$
- ACTIVATE SOME CONTROL SIGNALS
- CONTROL FLOW OF ACTIONS

SIMPLE EXAMPLE:

1. INPUT x, y (POS. INTEGERS)
2. IF $x \geq y$ THEN
3. $z = x - y$
   ELSE
   $z = x + y$
4. OUTPUT z



CS MSIA AUTONOMOUS CONTROLLER



AUTONOMOUS CONTROLLER

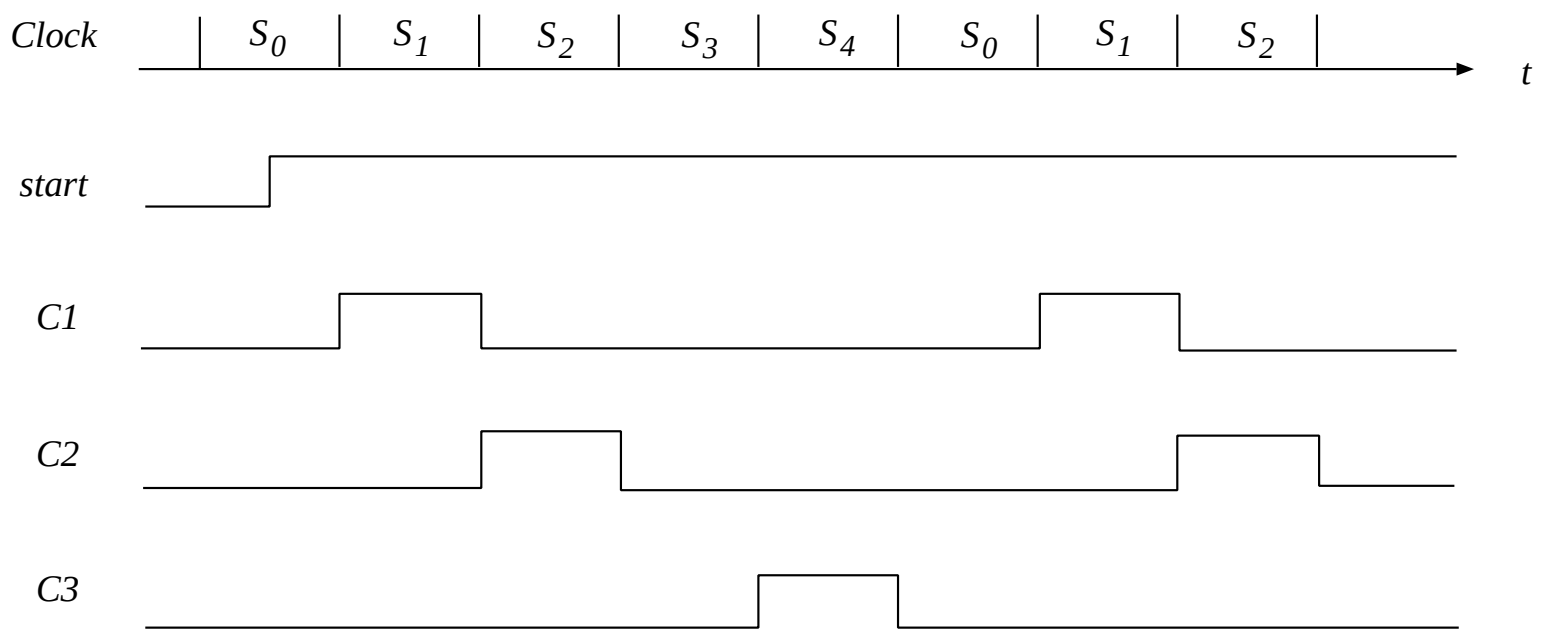


Figure 7.9: AUTONOMOUS CONTROLLER: TIMING DIAGRAM.

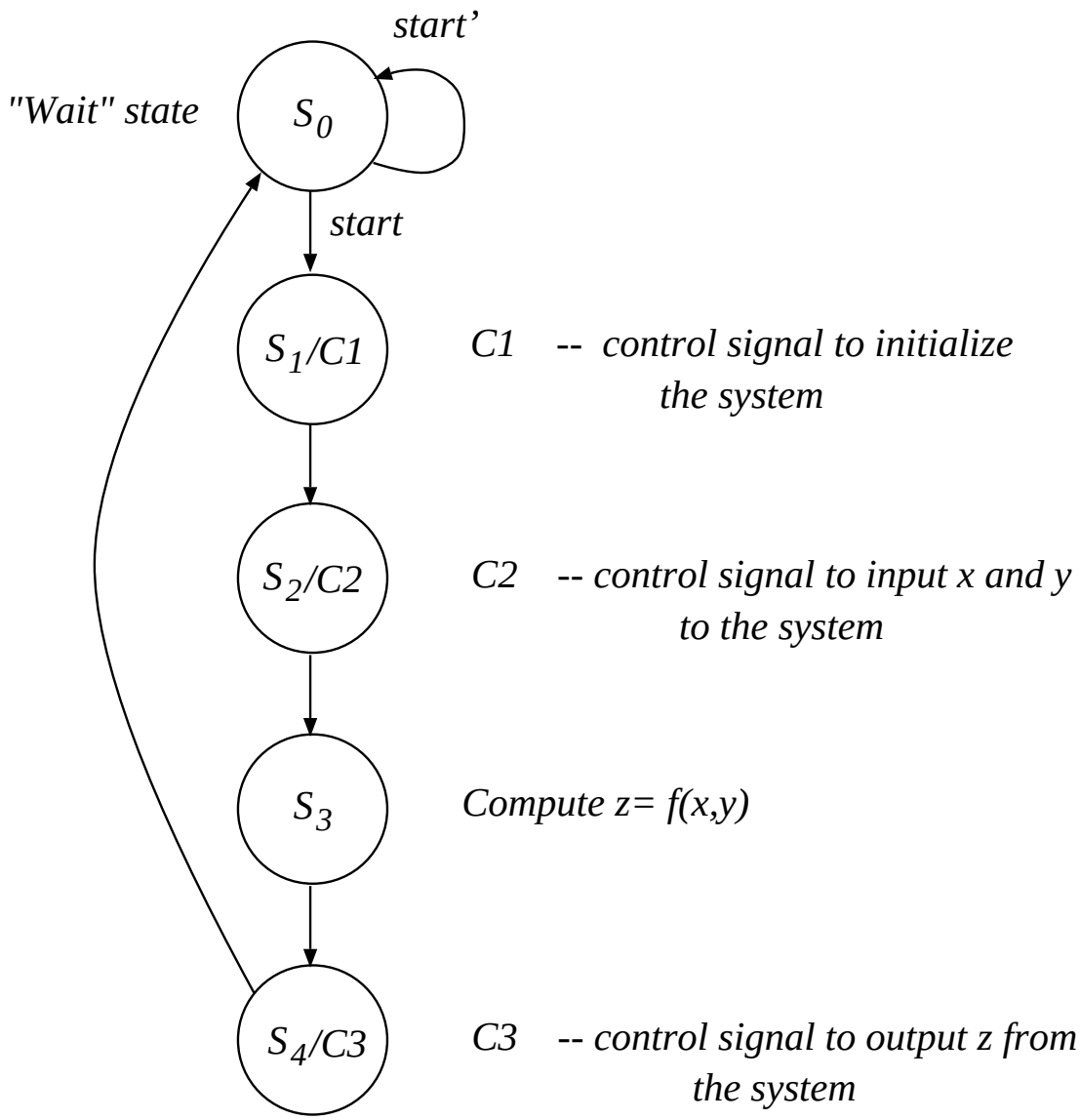


Figure 7.10: AUTONOMOUS CONTROLLER: STATE DIAGRAM.

GENERAL CONTROLLER

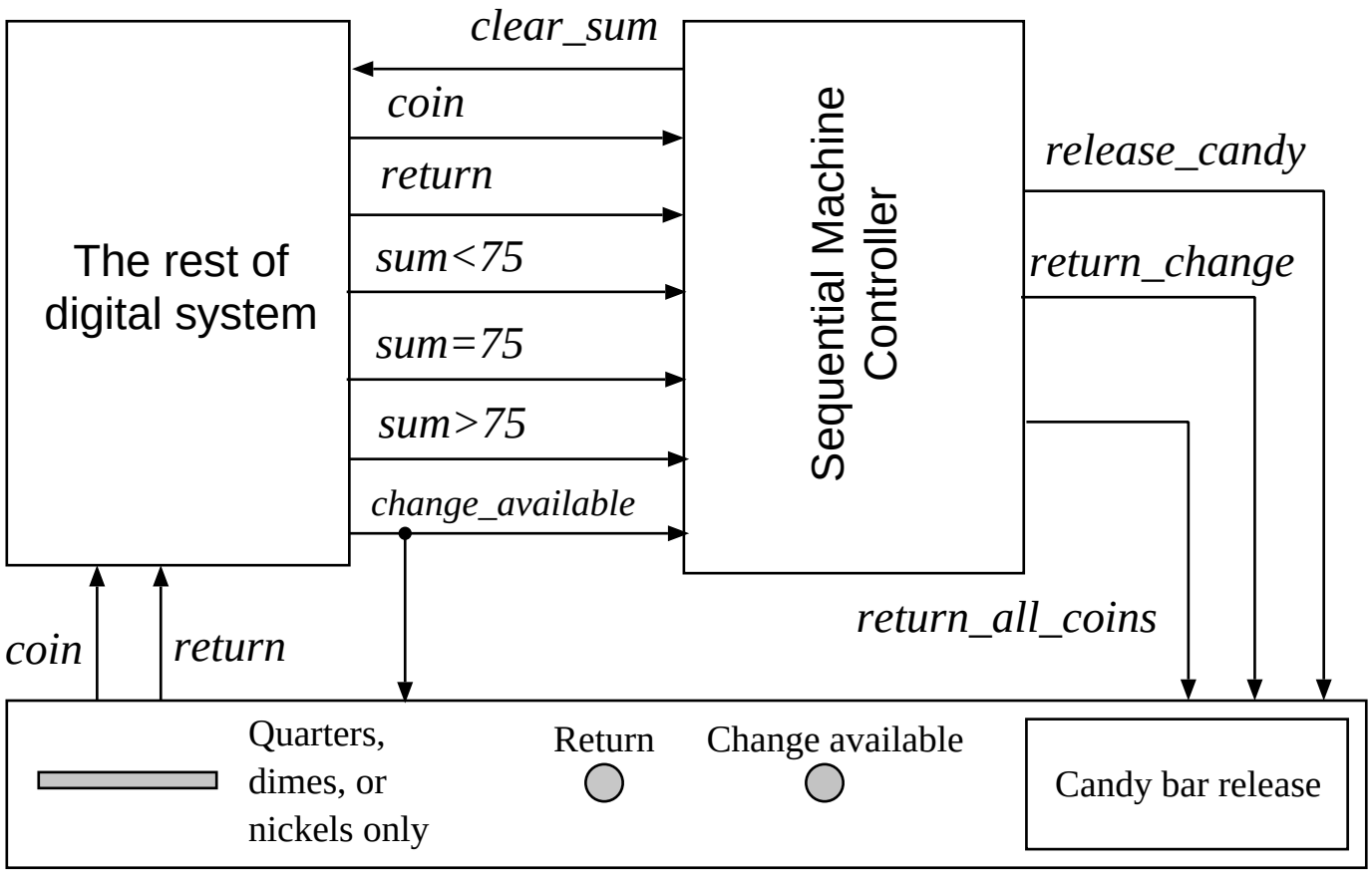


Figure 7.11: CONTROLLER FOR SIMPLE VENDING MACHINE: BLOCK DIAGRAM.

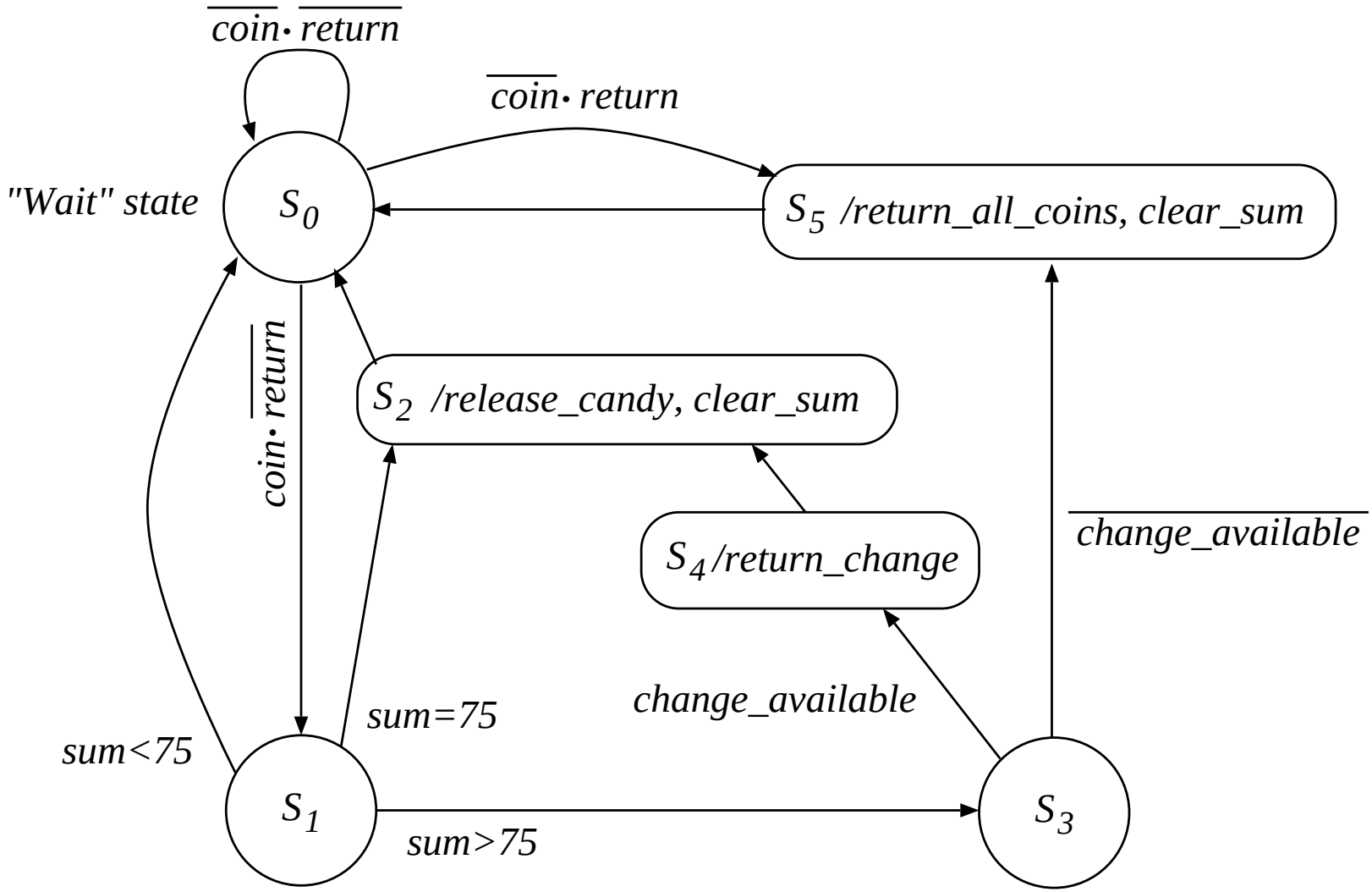


Figure 7.12: CONTROLLER FOR SIMPLE VENDING MACHINE: STATE DIAGRAM.

EQUIVALENT SEQUENTIAL SYSTEMS: SAME TIME BEHAVIOR

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

FUNCTION: $z(t) = \begin{cases} 1 \ \textbf{if} \ \ x(t-2, t) = 101 \\ 0 \ \textbf{otherwise} \end{cases}$

t	0	1	2	3	4	5	6	7	8
x	0	0	1	0	1	0	1	0	0
z	0	0	0	0	1	0	1	0	0

INITIAL STATE DIAGRAM and REDUCED STATE DIAGRAM

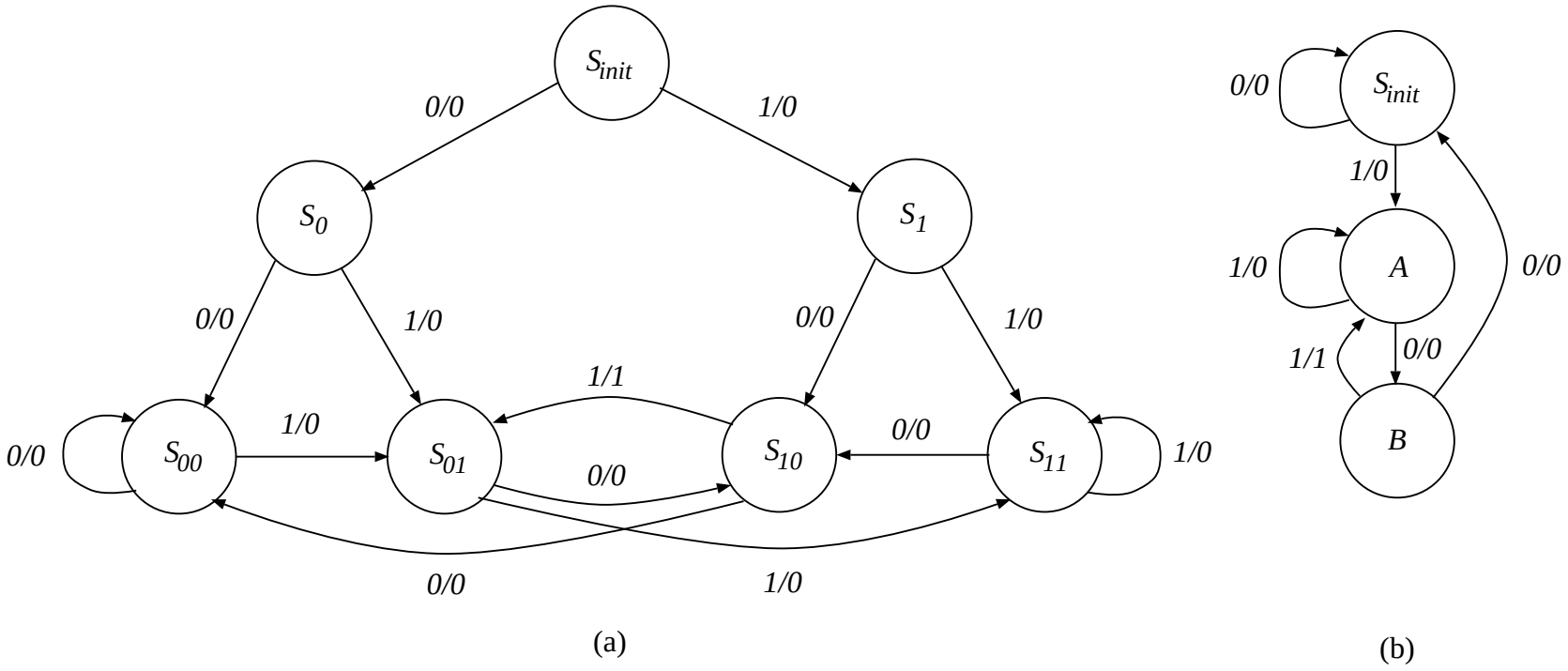


Figure 7.13: a) STATE DIAGRAM WITH REDUNDANT STATES; b) REDUCED STATE DIAGRAM

MINIMIZATION OF STATES

- A DERIVATION OF FSM MAY PRODUCE MORE STATES THAN NECESSARY
  → THESE STATES ARE REDUNDANT
  - A RED. STATE S_k HAS AN EQUIV. STATE S_j

EXAMPLE:

FSM_A: $\{S_0, S_1, S_2, S_3, S_4, S_5\}$ 6 STATES

FSM_B: $\{\{S_0, S_3\}, \{S_1, S_2, S_5\}, S_4\}$ 3 CLASSES

$\{S_a, S_b, S_c\}$ 3 STATES

MIN FSM

FSM_a AND FSM_b HAVE THE SAME TIME BEHAVIOR (FOR THE SAME INITIAL STATE)

EQUIVALENT STATES

- k-DISTINGUISHABLE STATES: DIFF. OUTPUT SEQUENCES

$$z(x(t, t+k-1), S_v) \neq z(x(t, t+k-1), S_w)$$

EXAMPLE:

State	$x(3,6)$	$z(3,6)$
S_1	0210	0011
S_3	0210	0001

- k-EQUIVALENT STATES: NOT DISTINGUISHABLE FOR SEQUENCES OF LENGTH k
- P_k: PARTITION OF STATES INTO k-EQUIVALENT CLASSES
- EQUIVALENT STATES – NOT DISTINGUISHABLE FOR ANY k

CS MSI A

EQUIVALENT STATES

DISTINGUISHABLE STATES:

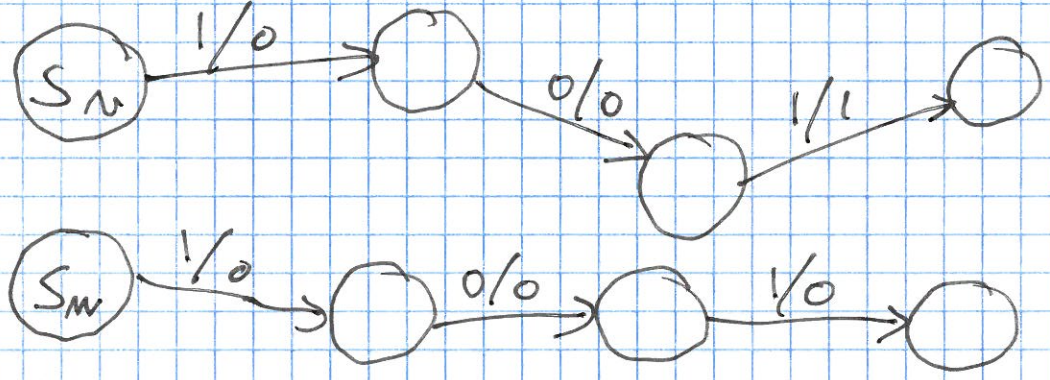
- AT LEAST ONE INPUT SEQUENCE THAT PRODUCES DIFF. OUTPUT SEQUENCE

k-DISTINGUISHABLE STATES:

$$z(x(t, t+k-1), S_v) \neq z(x(t, t+k-1), S_w)$$

$k=3$ $\quad z(x(t, t+1, t+2), S_v) \neq z(x(t, t+1, t+2), S_w)$

S_v, S_w ARE 3-DISTINGUISHABLE



$z((1\,0\,1), S_v) = 0\,0\,1$

$z((1\,0\,1), S_w) = 0\,0\,0$ $\neq$

Example 7.14

INPUT: $x(t) \in \{a, b, c\}$
OUTPUT: $z(t) \in \{0, 1\}$
STATE: $s(t) \in \{A, B, C, D, E, F\}$
INITIAL STATE: $s(0) = A$

FUNCTIONS: TRANSITION AND OUTPUT

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

Example 7.14 (cont.)

- A and B ARE 1-DISTINGUISHABLE BECAUSE

$$z(b, A) \neq z(b, B)$$

- A and C ARE 1-EQUIVALENT BECAUSE

$$z(x(t), A) = z(x(t), C), \quad for\ all\ x(t) \in I$$

- A and C ARE ALSO 2-EQUIVALENT BECAUSE

$$
\begin{array}{lllll}
z(aa, A) & = & z(aa, C) & = & 00 \\
z(ab, A) & = & z(ab, C) & = & 01 \\
z(ac, A) & = & z(ac, C) & = & 00 \\
z(ba, A) & = & z(ba, C) & = & 10 \\
z(bb, A) & = & z(bb, c) & = & 10 \\
z(bc, A) & = & z(bc, C) & = & 11 \\
z(ca, A) & = & z(ca, C) & = & 00 \\
z(cb, A) & = & z(cb, C) & = & 00 \\
z(cc, A) & = & z(cc, C) & = & 01
\end{array}
$$

PROCEDURE TO MINIMIZE NUMBER OF STATES

Obtain P_1: DIRECTLY FROM OUTPUT FUNCTION

From P_i to P_{i+1} ...

1. P_{i+1} IS A REFINEMENT OF P_i
   (states (i+1)-equiv. must also be i-equiv.)

P_i	$(A, B, C)(D)$	
	possible	not possible
P_{i+1}	$(A, C)(B)(D)$	$(A, D)(B)(C)$

   FOR (i+1)-EQUIVALENT STATES S_v and S_w

$$z(x(t, t+i), S_v) = z(x(t, t+i), S_w)$$

   FOR ARBITRARY $x(t, t+i)$

   THEN $z(x(t, t+i-1), S_v) = z(x(t, t+i-1), S_w)$

   EXAMPLE: $z(abcd, S_v) = z(abcd, S_w) = 1234$

   THEN $z(abc, S_v) = z(abc, S_w) = 123$

$(i+1)$-EQUIVALENT STATES

2. TWO STATES ARE (i+1)-EQUIVALENT IF AND ONLY IF

   a) THEY ARE i-EQUIVALENT, and

   b) FOR ALL $x \in I$, THE CORRESPONDING NEXT STATES ARE i-EQUIVALENT

   PROOF:

   IF PART:

   - SINCE THE STATES ARE i-EQUIVALENT, THEY ARE ALSO 1-EQUIVALENT
   - THEREFORE, IF THE NEXT STATES ARE i-EQUIVALENT, THE STATES ARE (i+1)-EQUIVALENT

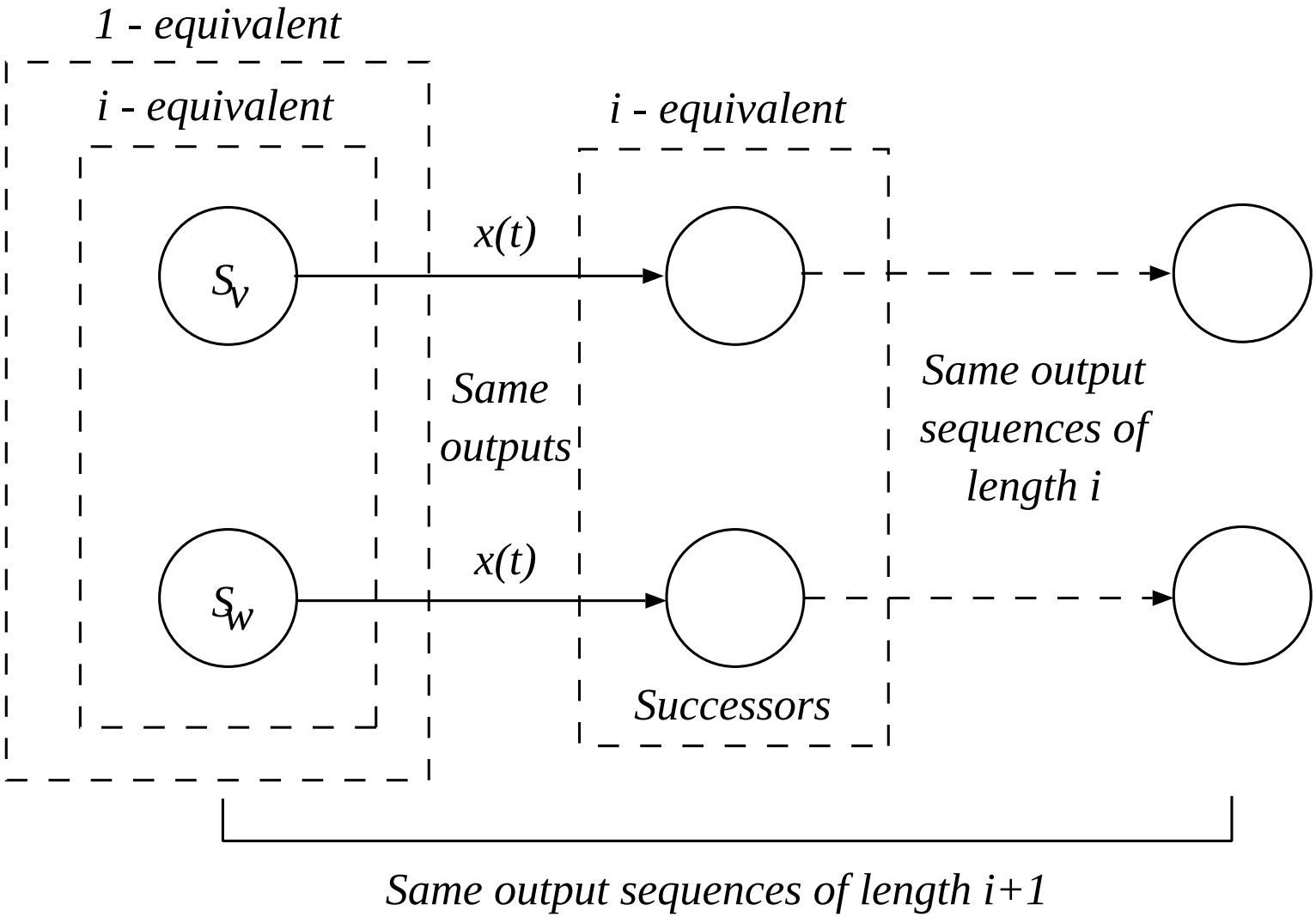


Figure 7.14: ILLUSTRATION OF $(i+1)$-EQUIVALENCE RELATION.

$(i+1)$-EQUIVALENT STATES (cont.)

ONLY IF PART: BY CONTRADICTION

- IF FOR SOME INPUT a THE NEXT STATES ARE NOT i-EQUIVALENT THEN THERE EXISTS A SEQUENCE T OF LENGTH i SUCH THAT THESE NEXT STATES ARE DISTINGUISHABLE.

THEREFORE,

$$z(aT, S_v) \neq z(aT, S_w)$$

$\rightarrow$ S_v AND S_w NOT (i+1)-EQUIVALENT

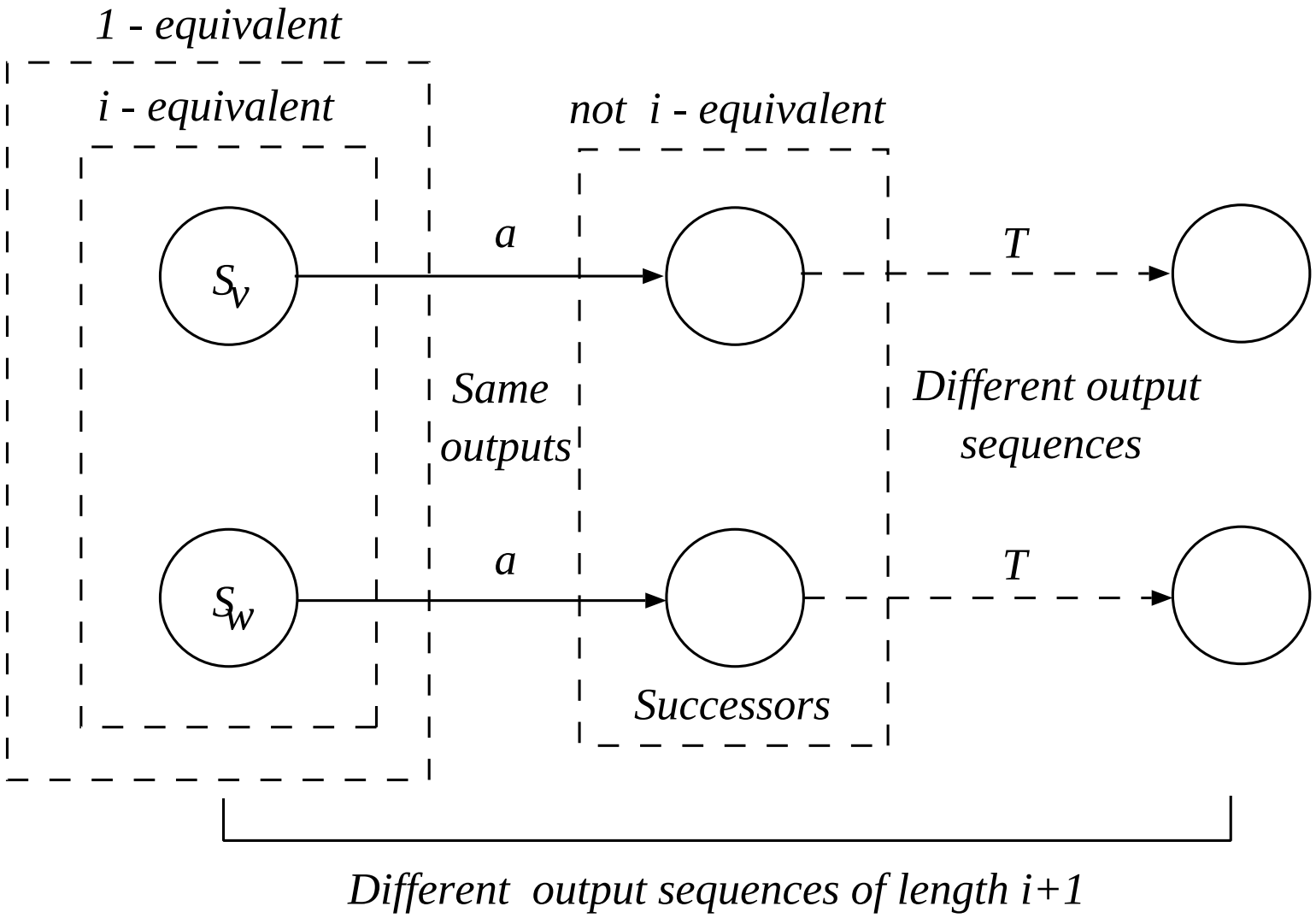


Figure 7.15: ILLUSTRATION OF $(i+1)$-EQUIVALENCE RELATION.

WHEN TO STOP?

- STOP WHEN P_{i+1} IS THE SAME AS P_i
  - THIS IS THE EQUIVALENCE PARTITION
  - THE PROCESS ALWAYS TERMINATES

PROCEDURE: SUMMARY

1. OBTAIN P_1 (look at the outputs)

2. OBTAIN P_{i+1} FROM P_i
   BY GROUPING STATES THAT ARE i-EQUIVALENT
   AND WHOSE CORRESPONDING SUCCESSORS
   ARE i-EQUIVALENT

3. TERMINATE WHEN $P_{i+1} = P_i$

4. WRITE THE REDUCED TABLE

Example 7.14

INPUT: $x(t) \in \{a, b, c\}$
OUTPUT: $z(t) \in \{0, 1\}$
STATE: $s(t) \in \{A, B, C, D, E, F\}$
INITIAL STATE: $s(0) = A$

FUNCTIONS: TRANSITION AND OUTPUT

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

Example 7.15

PS	$x(t) = a$	$x(t) = b$	$x(t) = c$
A	0	1	0
B	0	0	1
C	0	1	0
D	0	0	1
E	0	1	0
F	0	0	1
	$NS,\ z$		

- 1-EQUIVALENT IF SAME "row pattern"

$$P_1 = (A, C, E)\ \ (B, D, F)$$

Example 7.15 (cont.)

- NUMBER THE CLASSES IN P_1

- TWO STATES ARE IN THE SAME CLASS OF P_2
  IF THEIR SUCCESSOR COLUMNS HAVE THE SAME NUMBERS

	1			2		
P_1	$(A,$	$C,$	$E)$	$(B,$	$D,$	$F)$
a	1	1	1	2	2	2
b	2	2	2	2	2	1
c	2	2	2	1	1	2

BY IDENTIFYING IDENTICAL COLUMNS OF SUCCESSORS, WE GET $P_2 = (A, C, E)\ \ (B, D)\ \ (F)$

Example 7.15 (cont.)

- SAME PROCESS TO OBTAIN THE NEXT PARTITION:

P_2	1 $(A,$	$C,$	$E)$	2 $(B,$	$D),$	3 (F)
a	1	1	1	3	3	
b	2	2	3	2	2	
c	2	2	3	1	1	

$$P_3 = (A, C) \ \ (E) \ \ (B, D) \ \ (F)$$

- SIMILARLY, WE DETERMINE $P_4 = (A, C)\ (E)\ (B, D)\ (F)$

BECAUSE $P_4 = P_3$ THIS IS ALSO THE EQUIVALENCE PARTITION P

Example 7.15 (cont.)

THE MINIMAL SYSTEM:

PS	$x = a$	$x = b$	$x = c$
A	$E, 0$	$B, 1$	$B, 0$
B	$F, 0$	$B, 0$	$A, 1$
E	$A, 0$	$F, 1$	$F, 0$
F	$B, 0$	$A, 0$	$F, 1$
	NS, z		

THE STATES CAN BE RELABELLED

CS M51A STATE MINIMIZATION EXAMPLE

PS	x=0	x=1
A	B, 0	A, 0
B	F, 0	E, 1
C	D, 0	B, 0
D	B, 0	A, 0
E	C, 0	B, 1
F	A, 0	E, 0
G	E, 0	G, 0

NS, Z

7 STATES

$P_1 =$
0
1

$P_2 =$
0
1

$P_3 =$
0
1

S = (

BINARY SPECIFICATION OF SEQUENTIAL SYSTEMS

- THE STATE CODING IS CALLED *STATE ASSIGNMENT*
- CODING FUNCTIONS:

$$\begin{array}{lll} \text{INPUT} & C_I: & I \rightarrow \{0,1\}^n \\ \text{OUTPUT} & C_O: & O \rightarrow \{0,1\}^m \\ \text{STATE} & C_S: & S \rightarrow \{0,1\}^k \end{array}$$

Example 7.16

PS	$x = a$	$x = b$	$x = c$
A	$E, 0$	$B, 1$	$B, 0$
B	$F, 0$	$B, 0$	$A, 1$
E	$A, 0$	$F, 1$	$F, 0$
F	$B, 0$	$A, 0$	$F, 1$
	NS, z		

BINARY CODING

Input code

$x(t)$	$x_1(t)x_0(t)$
a	00
b	01
c	10

Output code

$z(t)$	
0	0
1	1

State assignment

$s(t)$	$s_1(t)s_0(t)$
A	00
B	01
E	10
F	11

- THE RESULTING BINARY SPECIFICATION:

$s_1(t)s_0(t)$	$x_1x_0 = 00$	$x_1x_0 = 01$	$x_1x_0 = 10$
00	$10, 0$	$01, 1$	$01, 0$
01	$11, 0$	$01, 0$	$00, 1$
10	$00, 0$	$11, 1$	$11, 0$
11	$01, 0$	$00, 0$	$11, 1$
	$s_1(t+1)s_0(t+1),\ z$		

LABELING ARCS WITH SWITCHING EXPRESSIONS

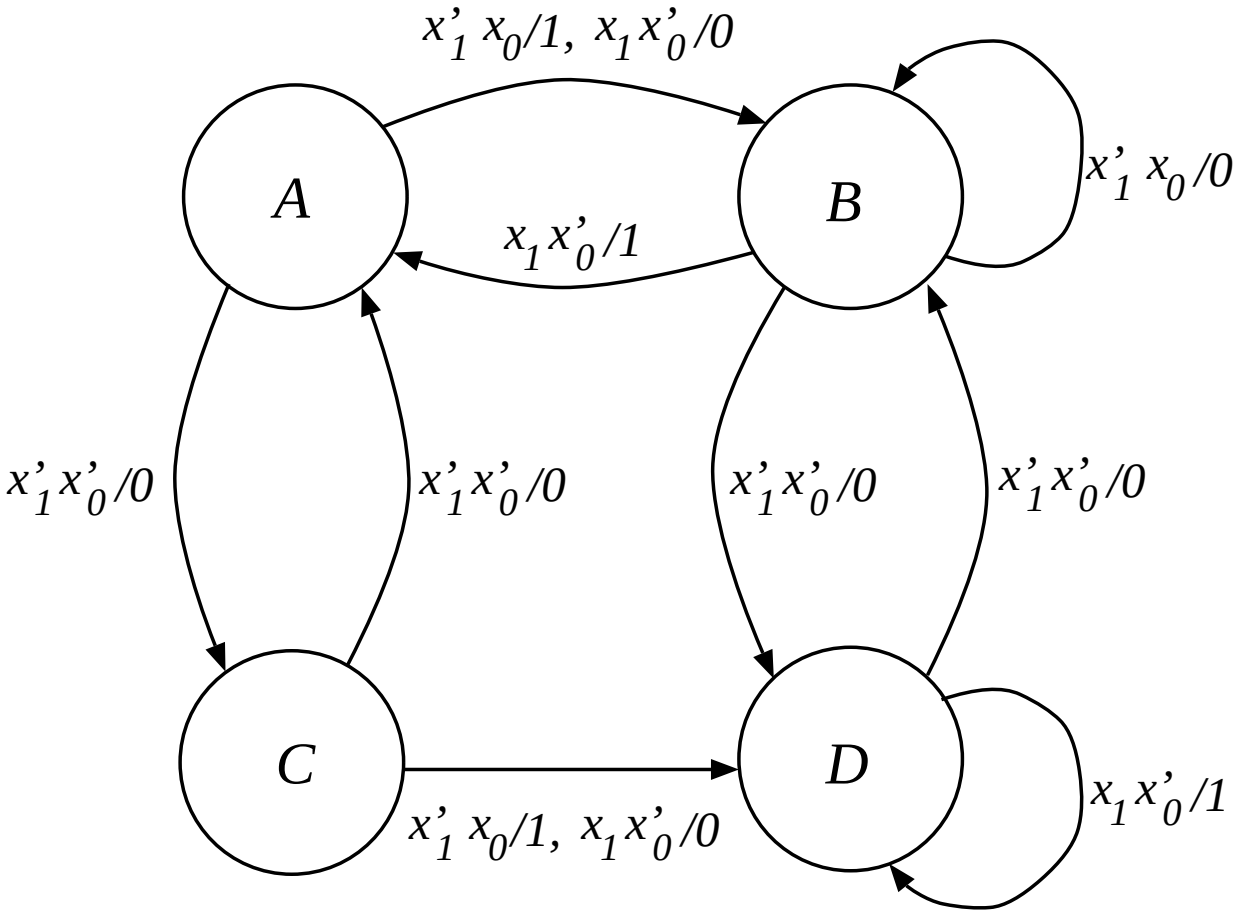


Figure 7.16: SWITCHING EXPRESSIONS AS ARC LABELS

SPECIFICATION OF DIFFERENT TYPES OF SEQUENTIAL SYSTEMS

MODULO-p COUNTER: 0, 1, 2, ..., p-1, 0, 1, ...

$$
\begin{aligned}
z(t) &= [\sum_{i=0}^{t} x(i)] \; mod \; p \\
s(t+1) &= [s(t) + x(t)] \; mod \; p \\
z(t) &= s(t) \quad \text{(if same coding)}
\end{aligned}
$$

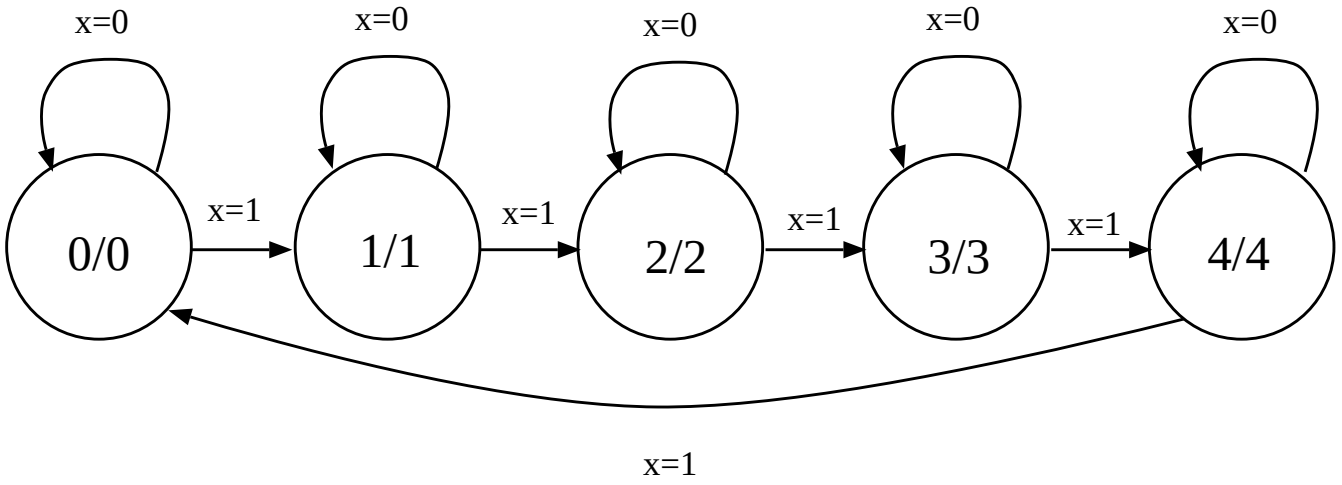


Figure 7.17: STATE DIAGRAM OF A MODULO-5 COUNTER

CS MSIA

PATTERN RECOGNIZERS

PATTERN $P = (p_0, p_1, \ldots, p_{m-1})$ — FINITE MEMORY

$p_i \in I$

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

FUNCTION:

$$z(t) = \begin{cases} 1 & \text{IF } x(t-m+1, t) = P \\ 0 & \text{OTHERWISE} \end{cases}$$

OVERLAPPED P's OK

STATE DESCRIPTION? TWO POSSIBILITIES

a) REPRESENT STATE WITH A VECTOR OF M-1 ELEMENTS

b) STATE REPRESENTS A PARTIAL PATTERN RECOGNITION

(EXAMPLE 7.11)

CS M51A PATTERN RECOGNIZER : APPROACH (a)

STATE VECTOR: $\underline{\Delta} = (\Delta_{m-2}, \Delta_{m-3}, \ldots, \Delta_1, \Delta_0)$

M-1 COMPONENTS PATTERN P OF LENGTH M

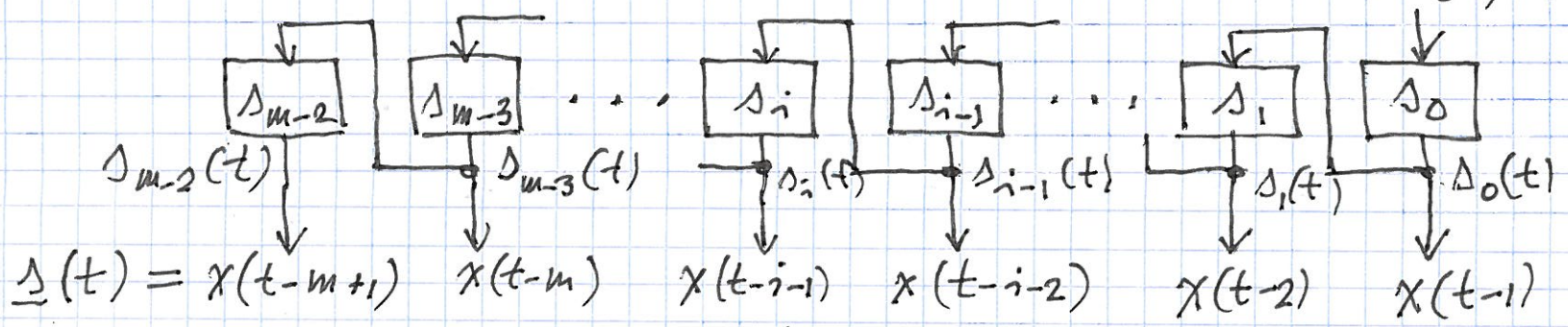
STATE EQUATIONS;

$$\Delta_i(t+1) = \Delta_{i-1}(t), \quad 1 \le i \le m-2 \qquad \text{"SHIFT" LEFT}$$

$$\Delta_0(t+1) = x(t)$$

$$\Rightarrow \underline{\Delta}(t) = (x(t-m+1), \ldots, x(t-2), x(t-1))$$

STATE STORES THE LAST M-1 INPUT SYMBOLS



TO GET $\underline{\Delta}(t+1)$, "SHIFT" LEFT $\underline{\Delta}(t)$

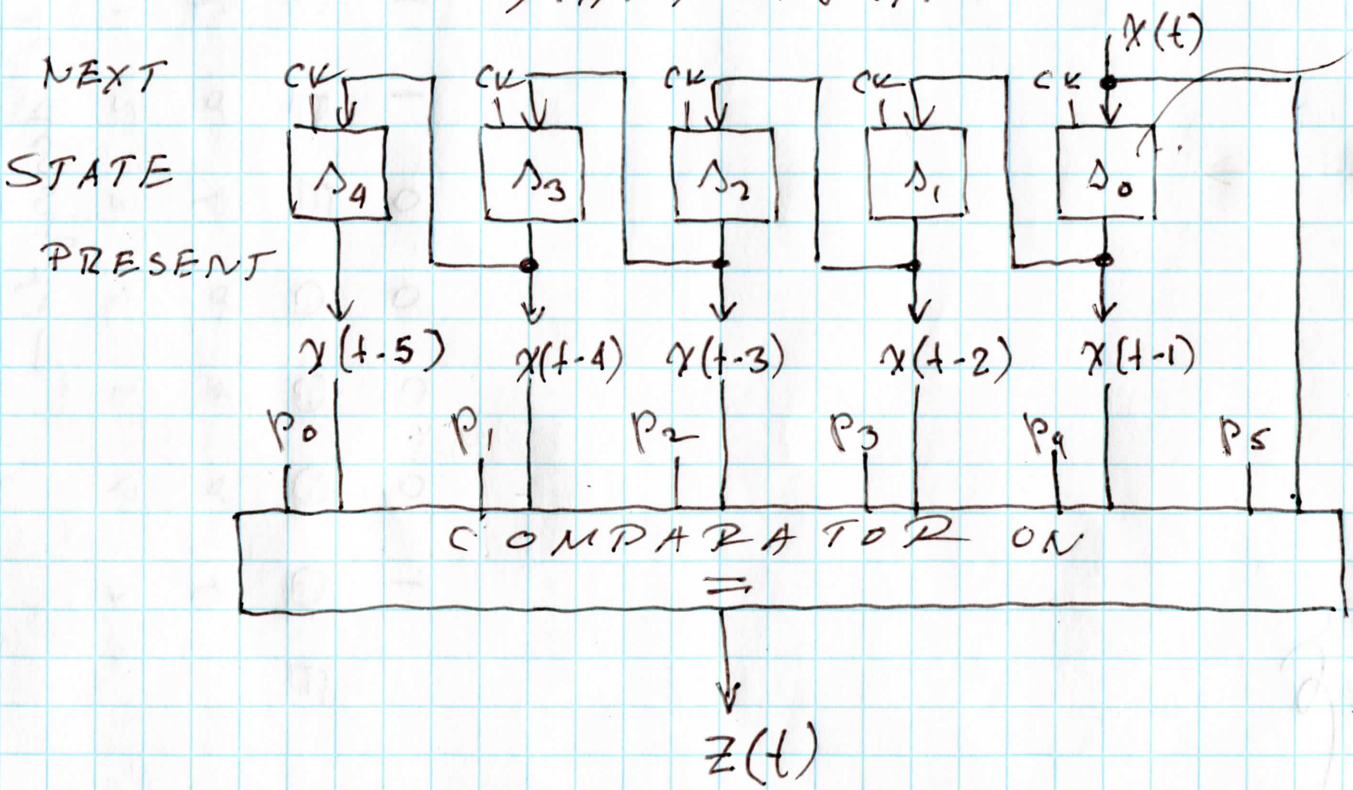
THEN

$$z(t) = \begin{cases} 1 & \text{IF } (\underline{\Delta}(t), x(t)) = P \\ 0 & \text{OTHERWISE} \end{cases}$$

CS M51A PATTERN RECOGNIZER USING STATE VECTOR

EXAMPLE

$$P = (p_0, p_1, p_2, p_3, p_4, p_5)$$



CS MSIA EXAMPLE 7.17

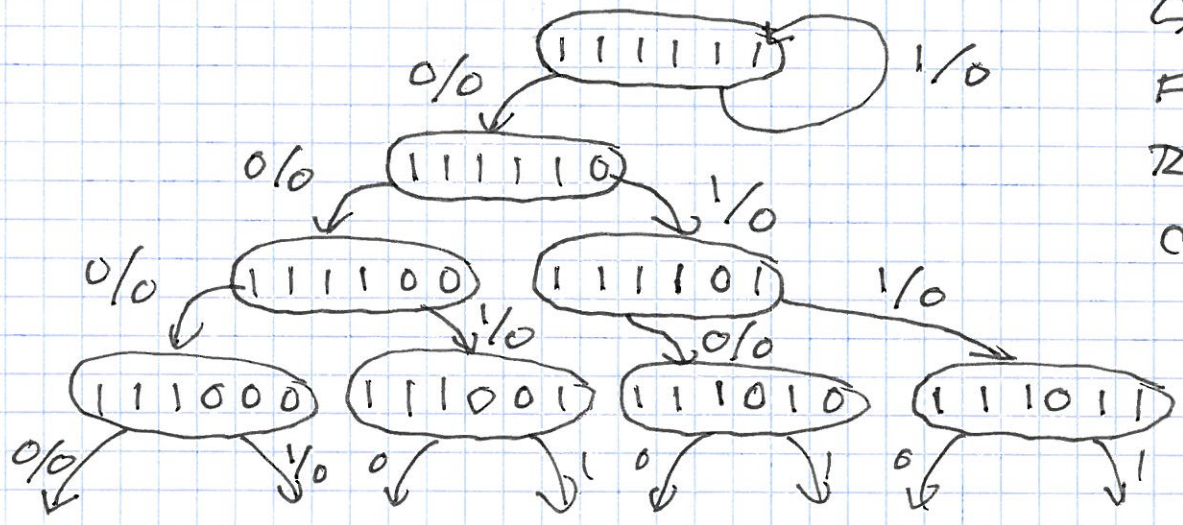
DETERMINE STATE DESCRIPTION TO RECOGNIZE

$P = 0101011$ $(m=7)$

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

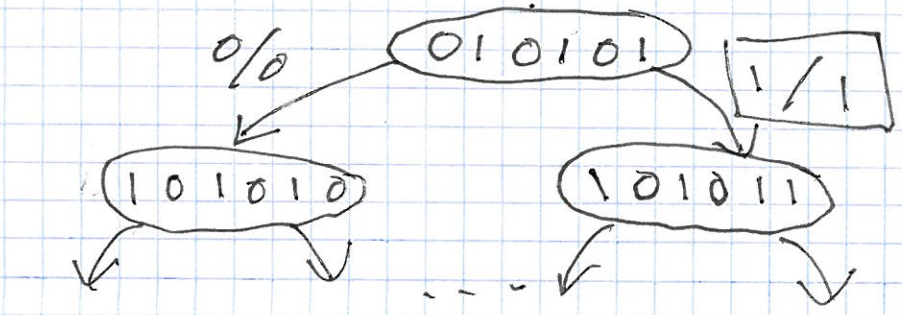
STATE: $\Delta(t) = (\Delta_5, \Delta_4, \Delta_3, \Delta_2, \Delta_1, \Delta_0)$

INIT. STATE $\Delta(0) = (1\,1\,1\,1\,1\,1)$



GENERAL, WORKS FOR ALL PATTERN RECOGNIZERS, COMPLICATED BUT ... HAS A SIMPLE AND GENERAL IMPLEMENTATON (CHAPTER 8)

$(2^{m-1}$ STATES$)$ $2^6 = 64$ STATES



PATTERN RECOGNIZER

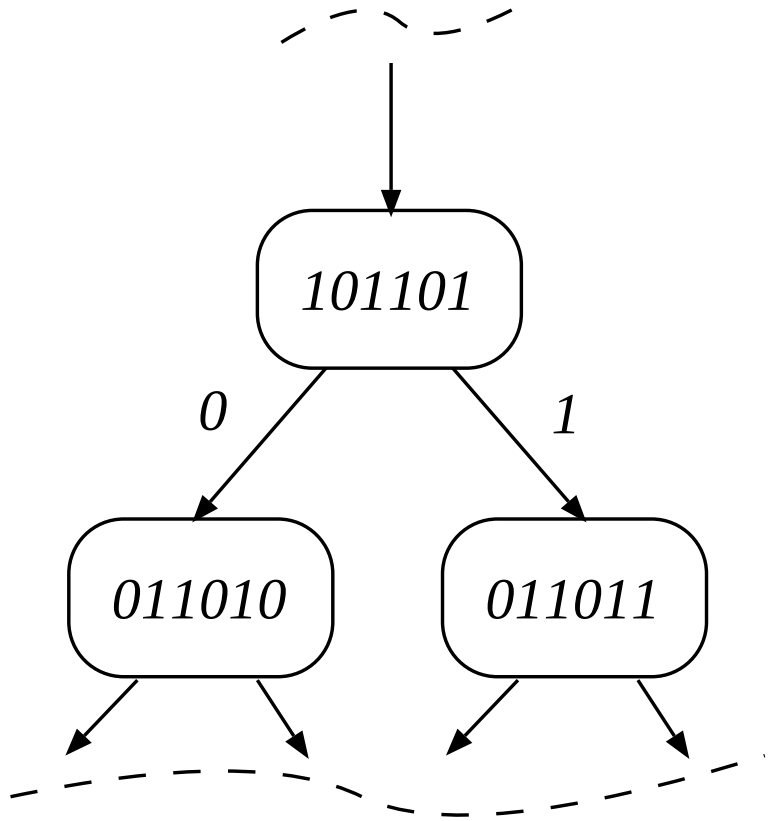


Figure 7.18: FRAGMENT OF STATE DIAGRAM OF PATTERN RECOGNIZER

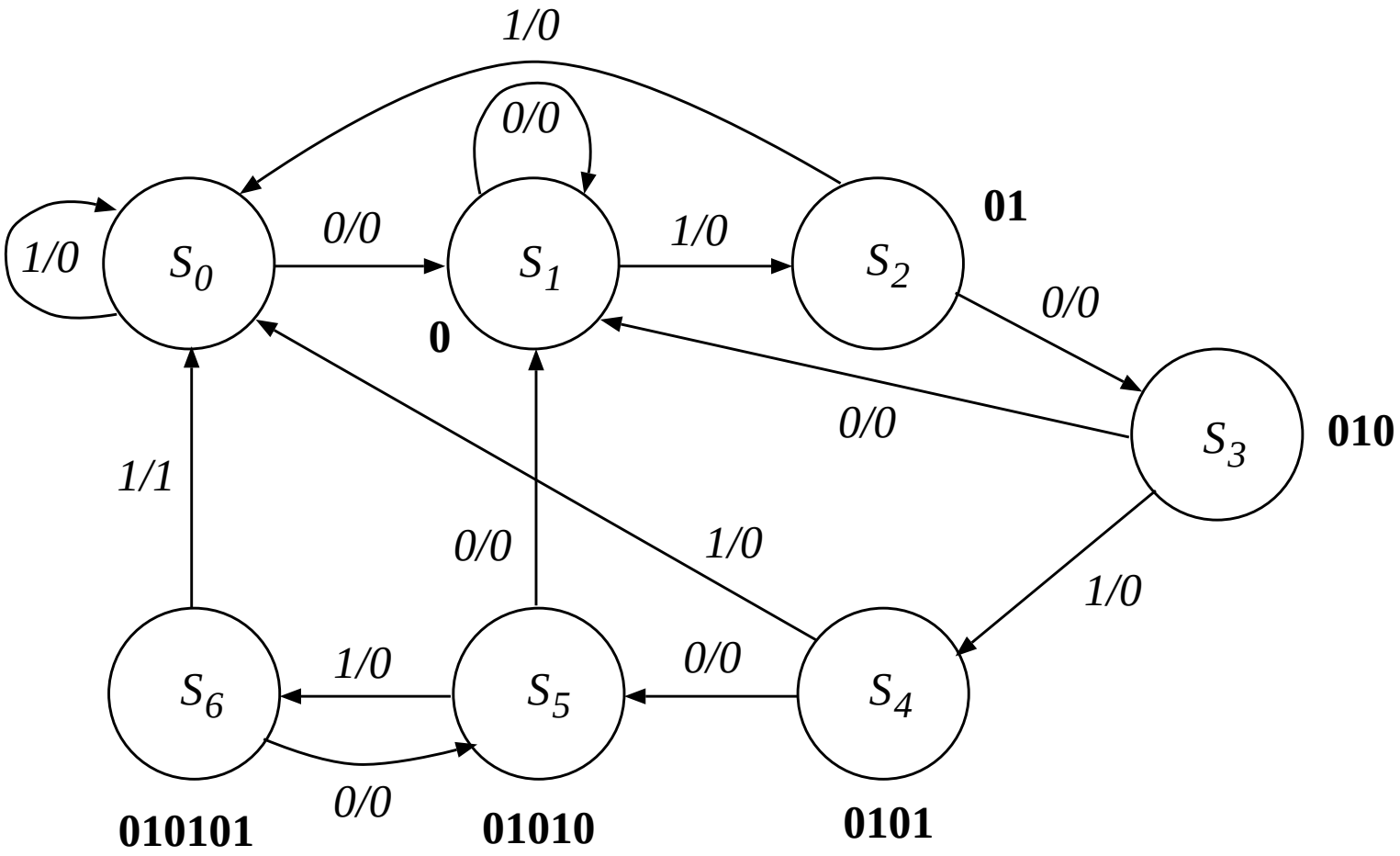
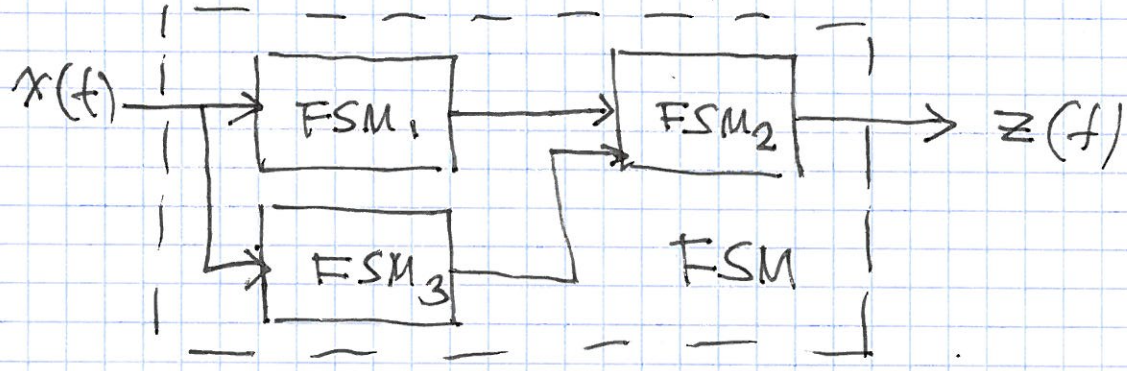


Figure 7.19: STATE DIAGRAM OF A PATTERN RECOGNIZER

CSMSIA FSM AS COMPOSITION OF FSM 1

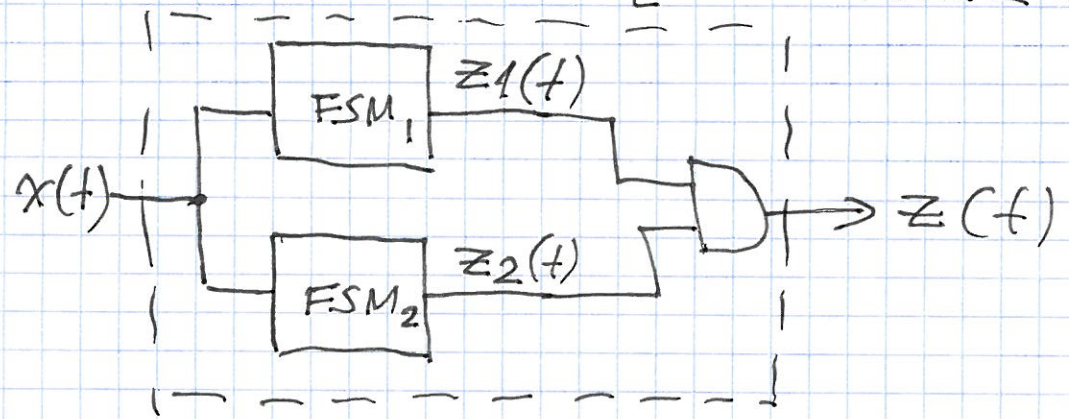


EXAMPLE: P1 AND P2 ARE PATTERNS

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

FUNCTION:

$$z(t) = \begin{cases} 1 & \text{IF P1 HAPPENED 6 AT LEAST 6 TIMES AND P2 // 2 // 2 //} \\ 0 & \text{OTHERWISE} \end{cases}$$



$$z1(t) = \begin{cases} 1 & \#P1 \geq 6 \\ 0 & \text{OTHERWISE} \end{cases}$$

$$z2(t) = \begin{cases} 1 & \#P2 \geq 2 \\ 0 & \text{OTHERWISE} \end{cases}$$