CS M51A
Logic Design of Digital Systems
Winter 2021

Some slides borrowed and modified from:
M.D. Ercegovac, T. Lang and J. Moreno, Introduction to Digital Systems.

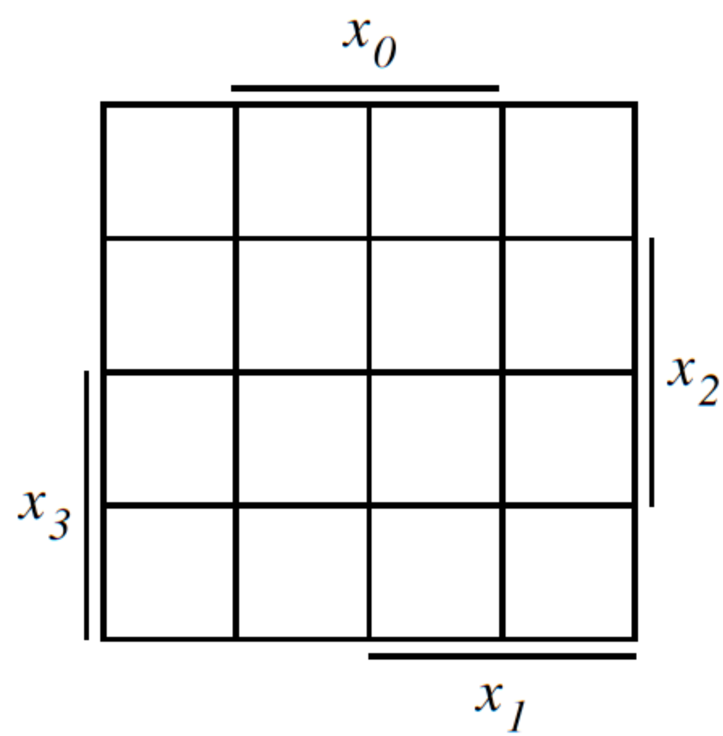
kARNAUGH MAPS

- 2-DIMENSIONAL ARRAY OF CELLS
- n VARIABLES $\longrightarrow 2^n$ CELLS

- REPRESENTING SWITCHING FUNCTIONS
- REPRESENTING SWITCHING EXPRESSIONS
- GRAPHICAL AID IN SIMPLIFYING EXPRESSIONS

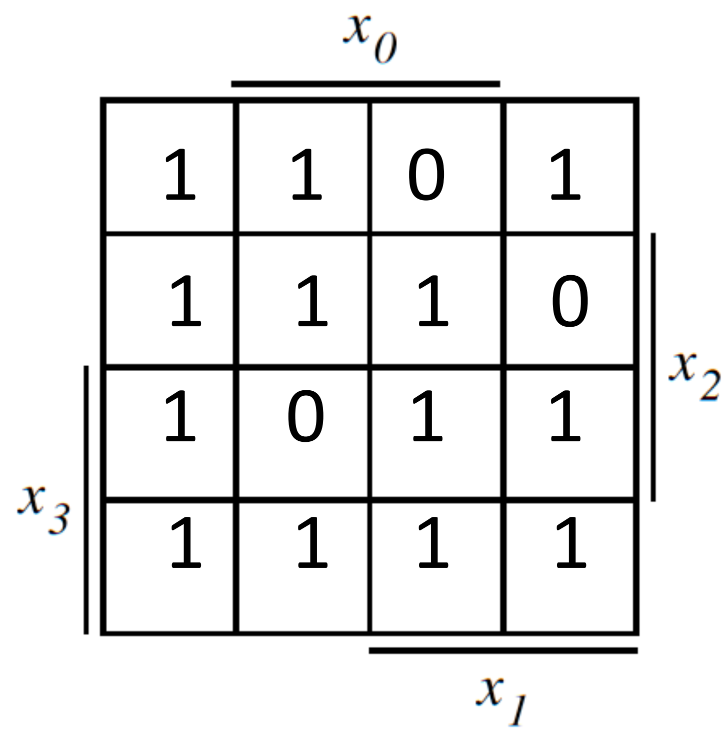
Simplifying using K-Map

$$F = x_3x_2x'_1x'_0 + x_3x_2x_1x'_0 + x'_3x_2x_1x'_0 + X'_3x_2x'_1x'_0 + x'_3x'_2x_1x_0 + x'_3x'_2x'_1x_0$$



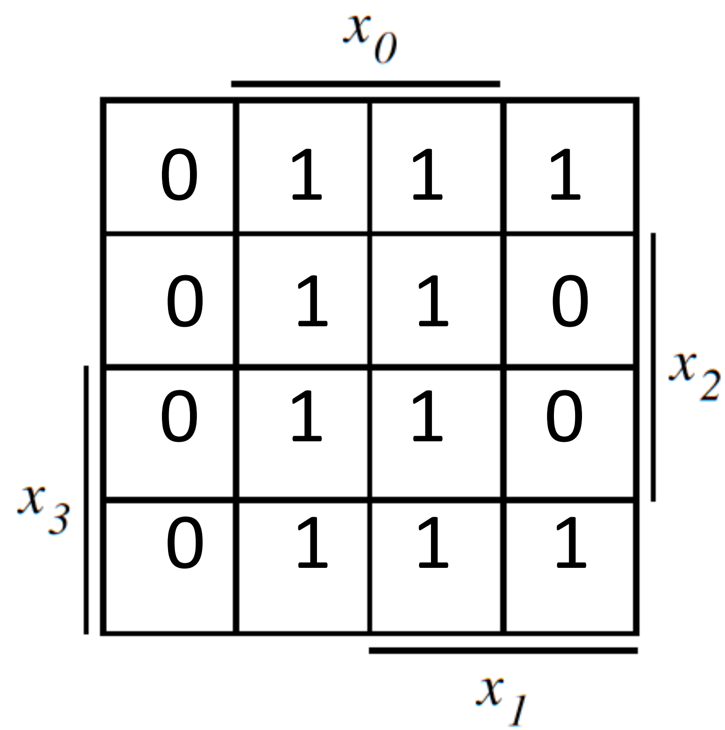
Presenting **Product of Sums** using K-Map

F=



Simplifying PRODUCT of SUMs - Examples

F=?



Clicker Question

Which one is the simplest correct expression?

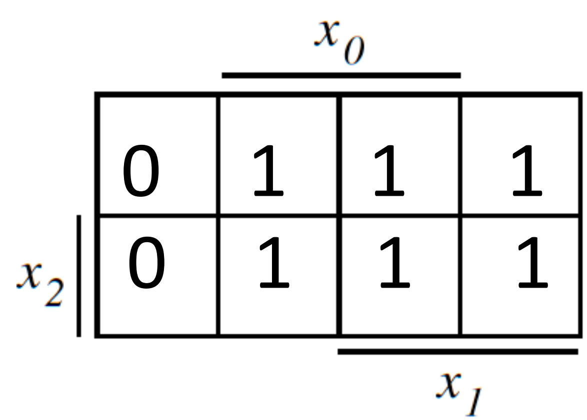
a) $F = x'_2 x'_1 x_0 + x_2 x_1 x'_0$

b) $F = x_2 x_1 x_0 + x_2 x'_1 x'_0$

c) $F = x_1 + x_0$

d) $F = x_2 + x_1$

e) none



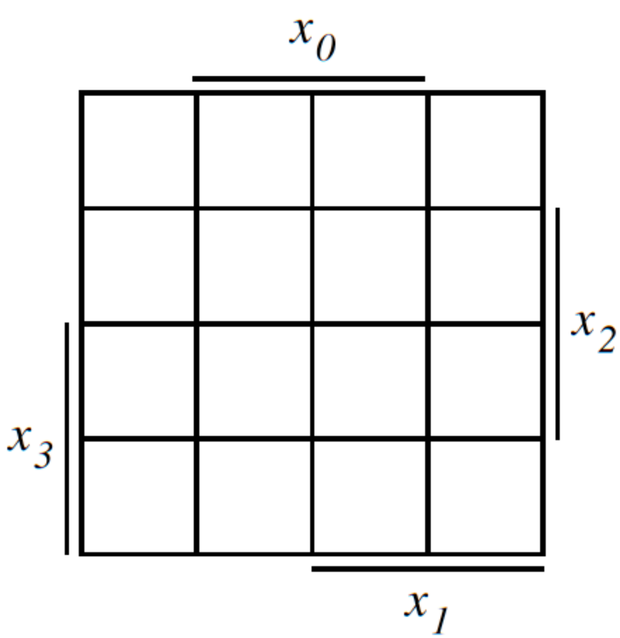
MINIMAL TWO-LEVEL GATE NETWORK DESIGN: EXAMPLE

Input: $x \in \{0, 1, 2, ..., 9\}$, coded in BCD as

$$\underline{x} = (x_3, x_2, x_1, x_0),\ \ x_i \in \{0, 1\}$$

Output: $z \in \{0, 1\}$

Function: $z = \begin{cases} 1 \ \textbf{if} \ \ x \in \{0, 2, 3, 5, 8\} \\ 0 \ \textbf{otherwise} \end{cases}$



MIN SP: $z =$

MIN PS: $z =$

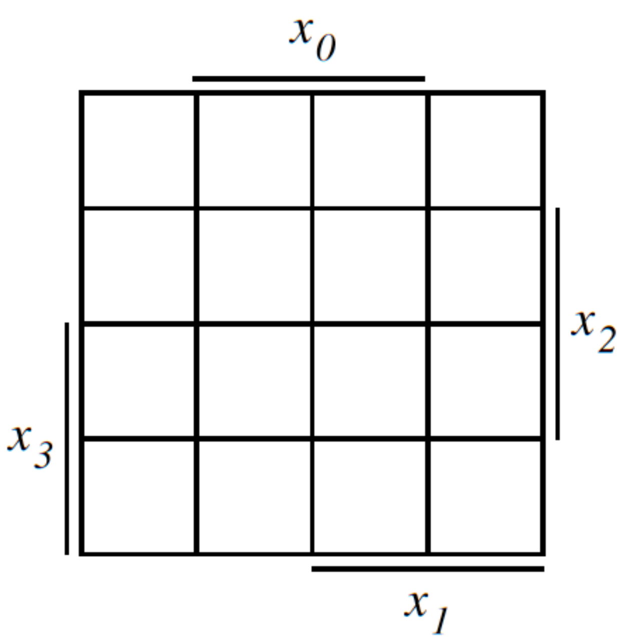
Gate Level Design

MINIMAL TWO-LEVEL GATE NETWORK DESIGN: EXAMPLE

Input: $x \in \{0, 1, 2, ..., 15\}$
represented in binary code by $\underline{x} = (x_3, x_2, x_1, x_0)$

Output: $z \in \{0, 1\}$

Function: $z = \begin{cases} 1 & \textbf{if} \;\; x \in \{0, 1, 3, 5, 7, 11, 12, 13, 14\} \\ 0 & \textbf{otherwise} \end{cases}$



MIN SP: $z =$

MIN PS: $z =$

Gate Level Design

DESIGN OF MULTIPLE-OUTPUT TWO-LEVEL GATE NETWORKS

- SEPARATE NETWORK FOR EACH OUTPUT: NO SHARING

Inputs: $(x_2, x_1, x_0), \quad x_i \in \{0, 1\}$
Output: $z \in \{0, 1, 2, 3\}$
Function: $z = \Sigma_{i=0}^{2} x_i$

1. THE SWITCHING FUNCTIONS IN TABULAR FORM ARE

x_2	x_1	x_0	z_1	z_0
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

2. THE CORRESPONDING K-MAPS ARE

$$z_1 \qquad \begin{array}{c|cccc|} & & \overline{\quad x_0 \quad} & & \\ \hline & 0 & 0 & 1 & 0 \\ x_2 & 0 & 1 & 1 & 1 \\ \hline & & & \underline{\quad x_1 \quad} & \end{array} \qquad\qquad z_0 \qquad \begin{array}{c|cccc|} & & \overline{\quad x_0 \quad} & & \\ \hline & 0 & 1 & 0 & 1 \\ x_2 & 1 & 0 & 1 & 0 \\ \hline & & & \underline{\quad x_1 \quad} & \end{array}$$

3. MINIMAL SPs:

$$\begin{aligned} z_1 &= x_2x_1 + x_2x_0 + x_1x_0 \\ z_0 &= x_2'x_1'x_0 + x_2'x_1x_0' + x_2x_1'x_0' + x_2x_1x_0 \end{aligned}$$

4. MINIMAL PSs:

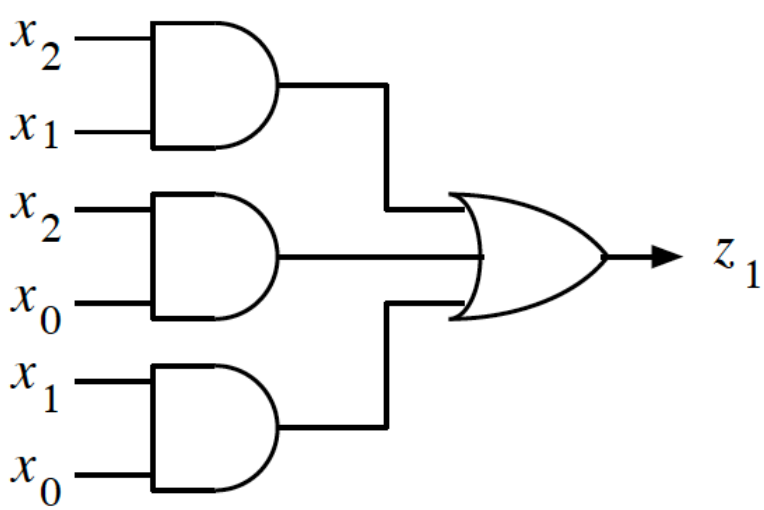
$$\begin{aligned} z_1 &= (x_2 + x_0)(x_2 + x_1)(x_1 + x_0) \\ z_0 &= (x_2 + x_1 + x_0)(x_2 + x_1' + x_0') \\ &\quad (x_2' + x_1 + x_0')(x_2' + x_1' + x_0) \end{aligned}$$

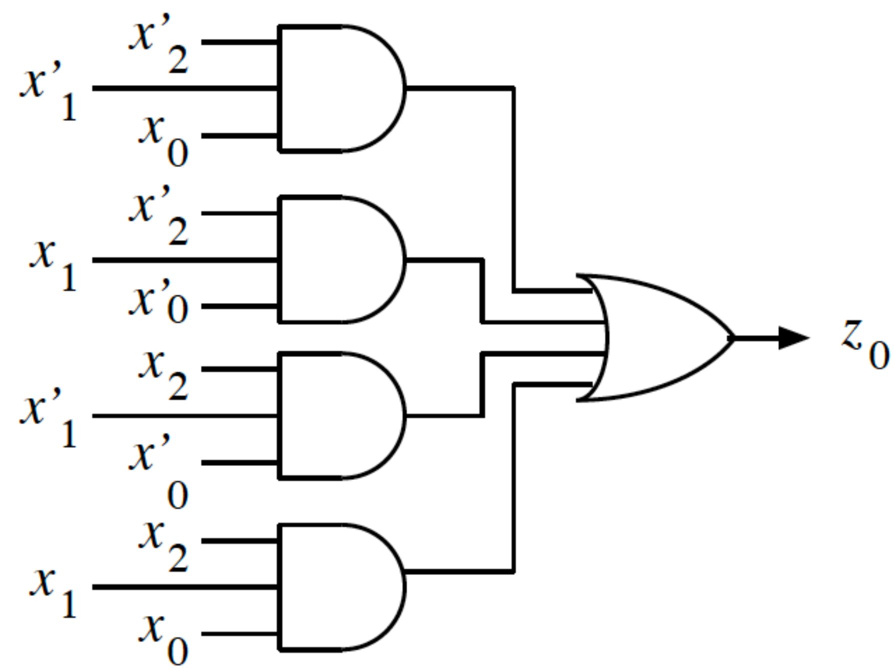
5. SP AND PS EXPRESSIONS HAVE THE SAME COST

Gate Level Design

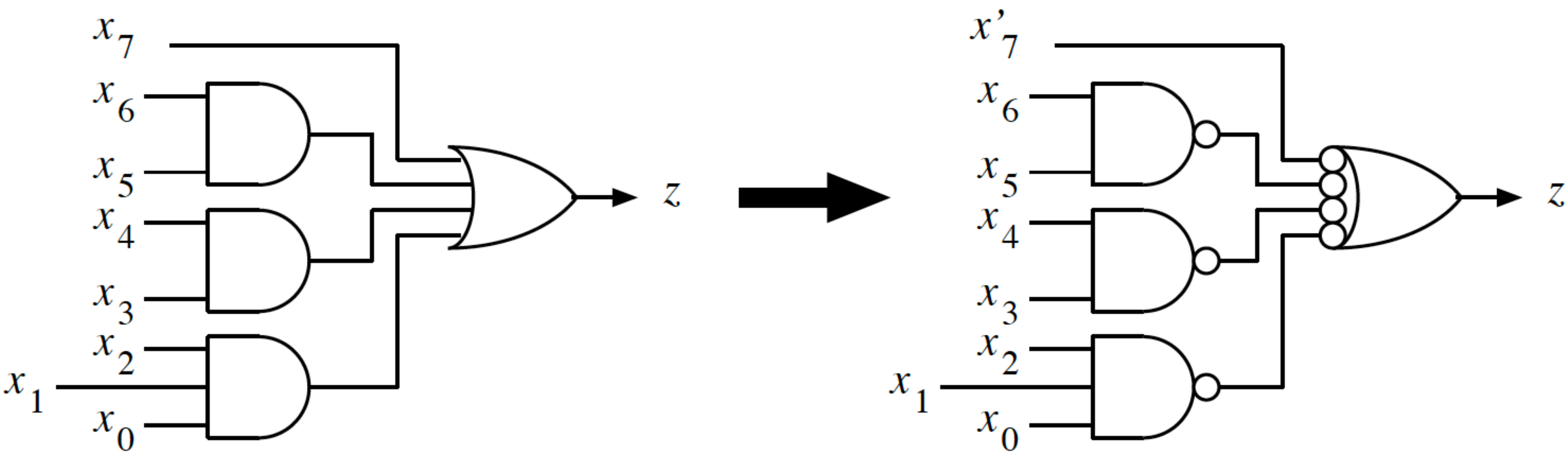
$$
\begin{aligned}
z_1 &= x_2x_1 + x_2x_0 + x_1x_0 \\
z_0 &= x_2'x_1'x_0 + x_2'x_1x_0' + x_2x_1'x_0' + x_2x_1x_0
\end{aligned}
$$

$$
\begin{aligned}
z_1 &= (x_2 + x_0)(x_2 + x_1)(x_1 + x_0) \\
z_0 &= (x_2 + x_1 + x_0)(x_2 + x_1' + x_0') \\
&\quad (x_2' + x_1 + x_0')(x_2' + x_1' + x_0)
\end{aligned}
$$

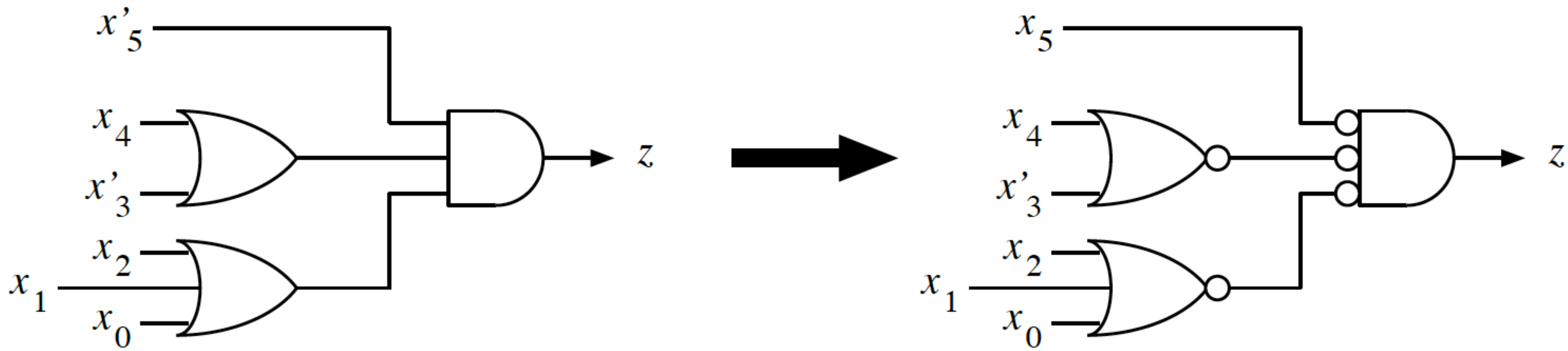




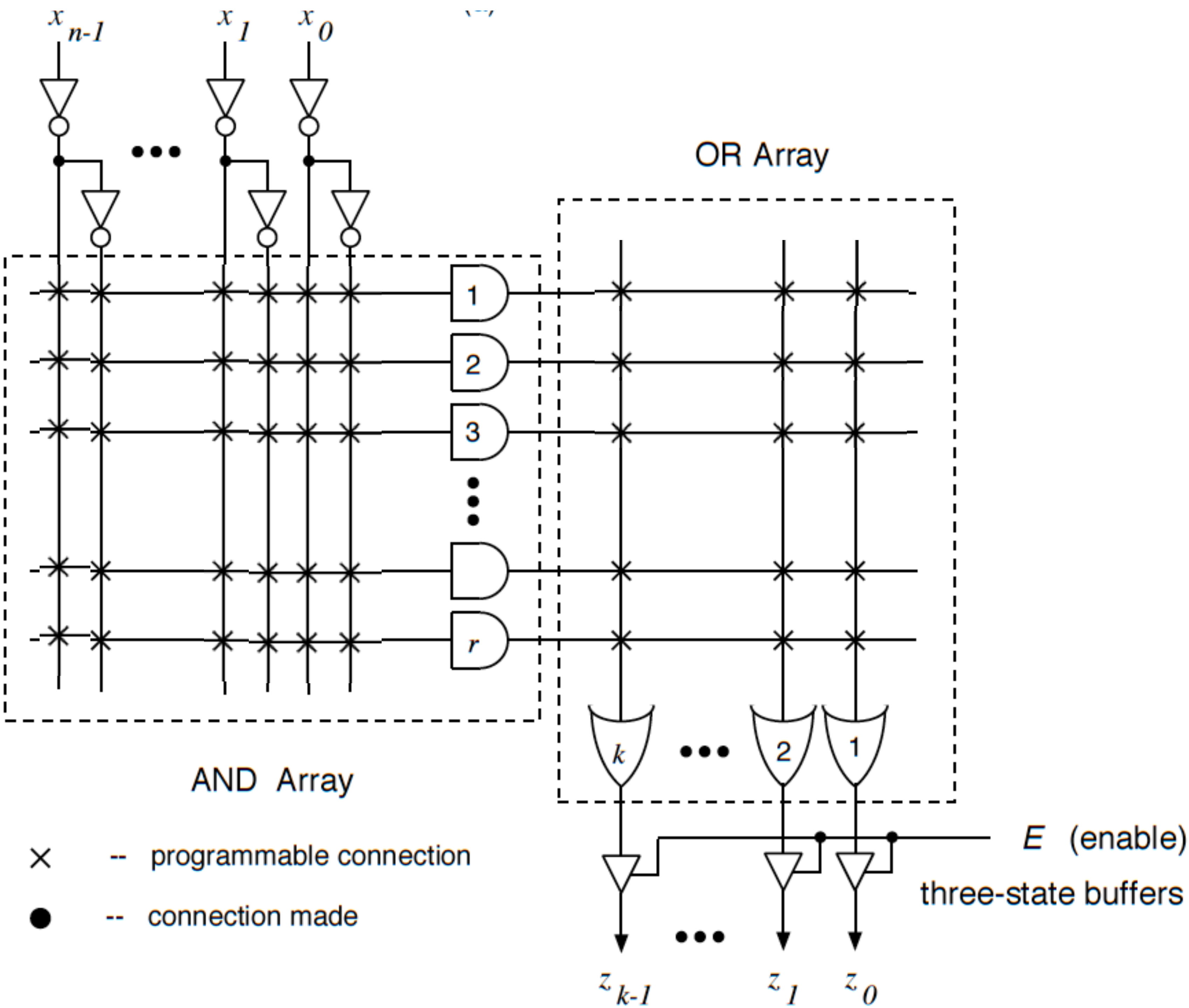
AND-OR network to NAND network



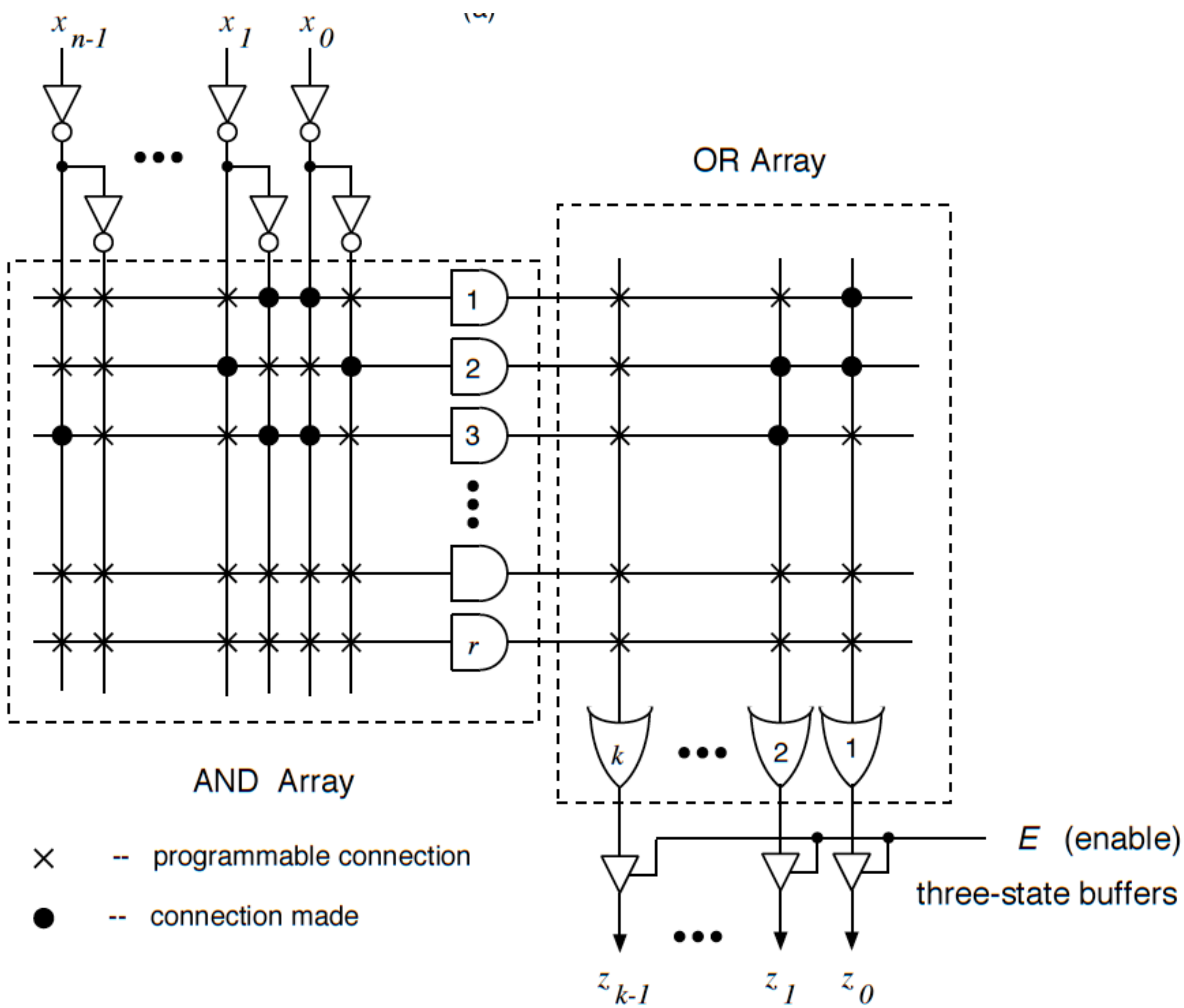
OR-AND network to NOR network



PROGRAMMABLE LOGIC ARRAY



x_{n-1}
x_1
x_0
OR Array
1
2
3
r
k
2
1
AND Array
× -- programmable connection
● -- connection made
E (enable)
three-state buffers
z_{k-1}
z_1
z_0



x_{n-1}
x_1
x_0
OR Array
1
2
3
r
k
2
1
AND Array
E (enable)
three-state buffers
× -- programmable connection
● -- connection made
z_{k-1}
z_1
z_0

Clicker Question

Which one is correct?

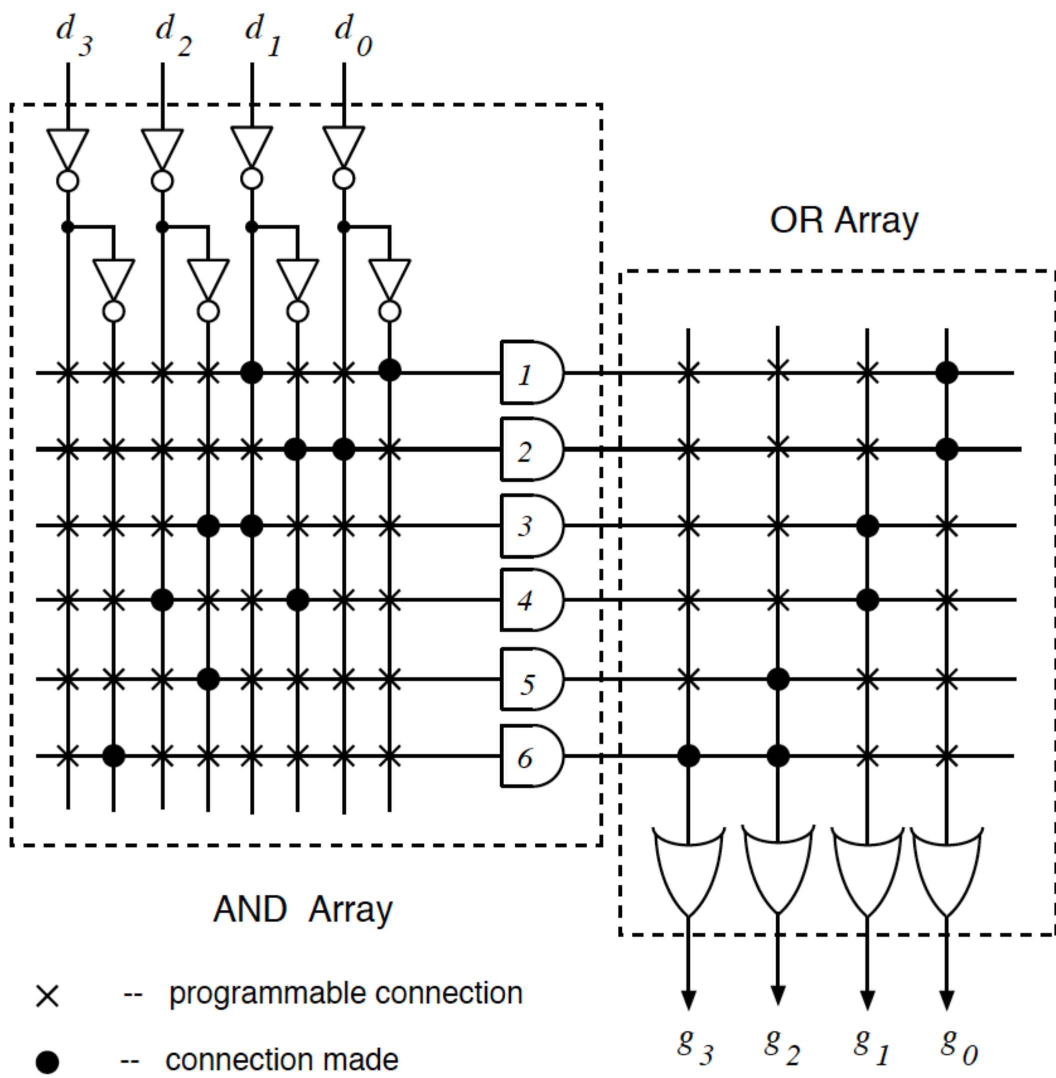
a) $g_0 = d_1d'_0+d'_1d_0$

b) $g_0 = d_1d_0$

c) $g_0 = d_1d'_0+d'_1d'_0$

d) $g_0 = d_1d_0+d'_1d'_0$

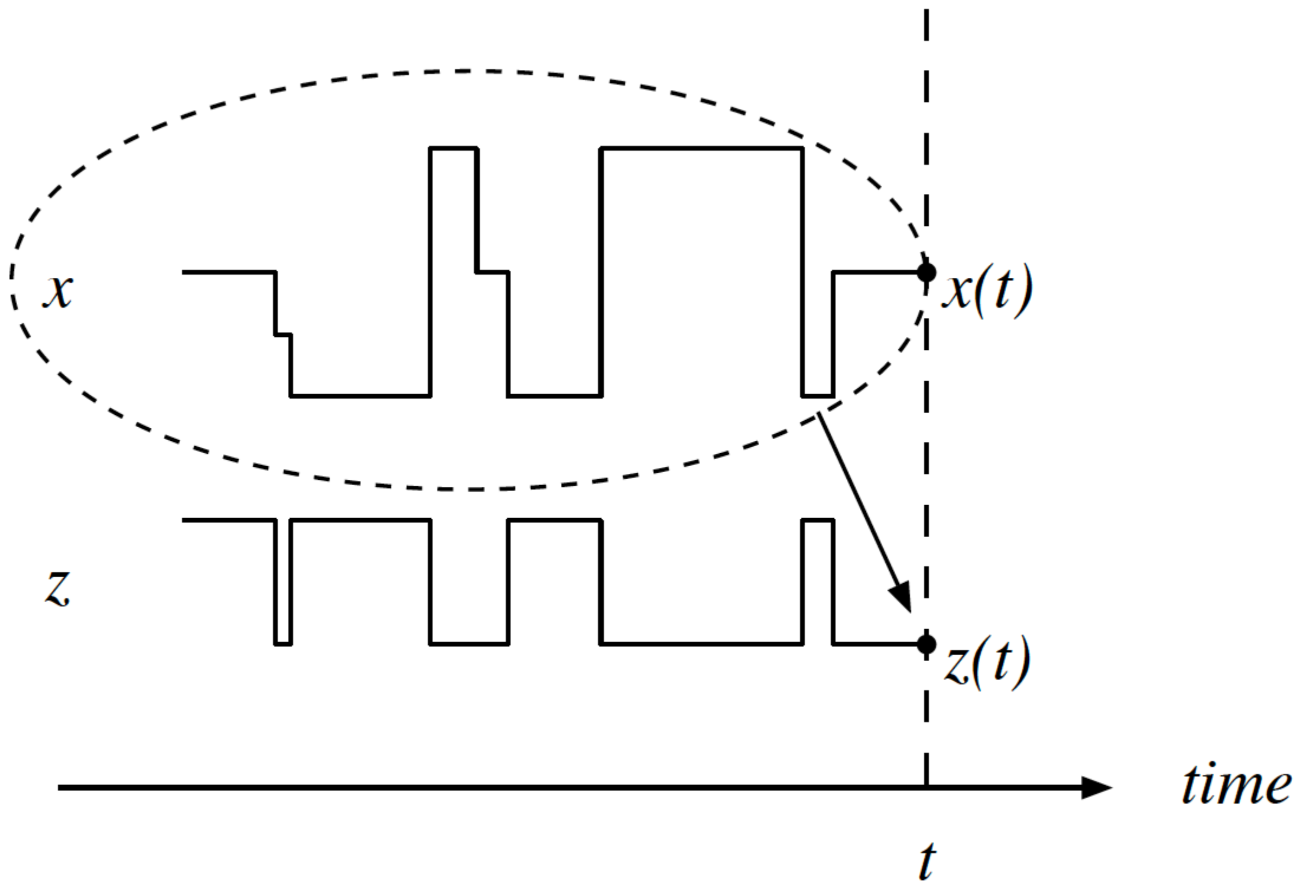
e) none



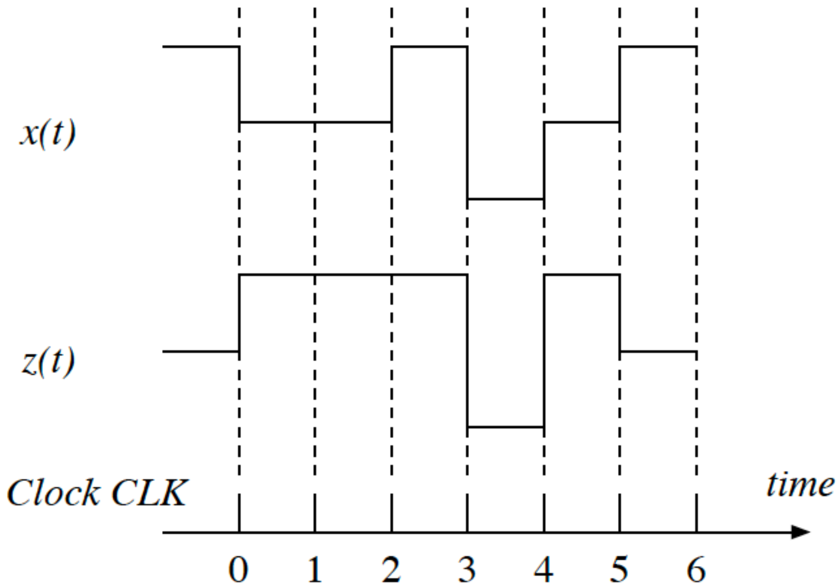
SEQUENTIAL SYSTEMS

DEFINITION

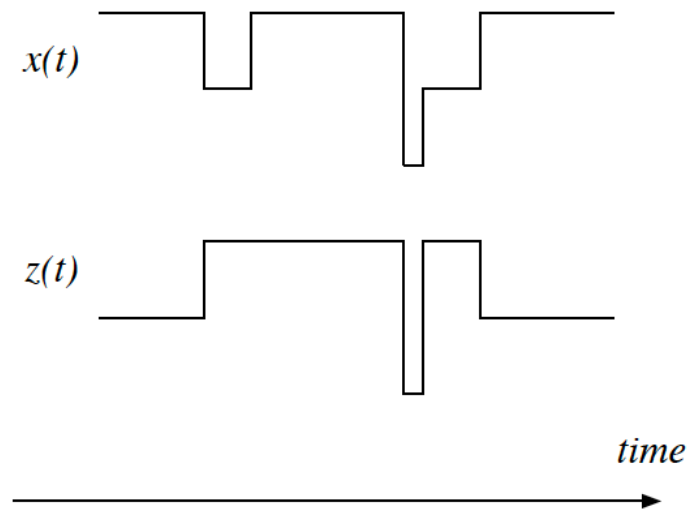
$$z(t) = F(x(0,t))$$



SYNCHRONOUS AND ASYNCHRONOUS SYSTEMS



(a)



(b)

Example ODD/EVEN

TIME-BEHAVIOR SPECIFICATION:

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

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

I/O SEQUENCE:

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

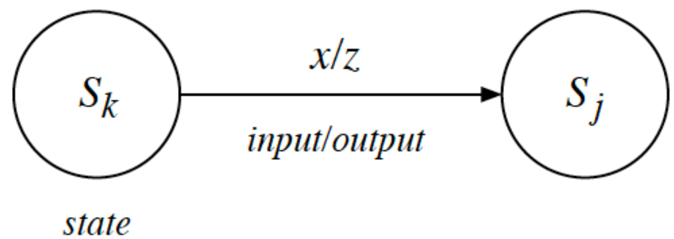
STATE DESCRIPTION OF ODD/EVEN

Input: $x(t) \in \{a, b\}$
Output: $z(t) \in \{0, 1\}$
State: $s(t) \in \{\text{EVEN, ODD}\}$
Initial state: s(0)= EVEN

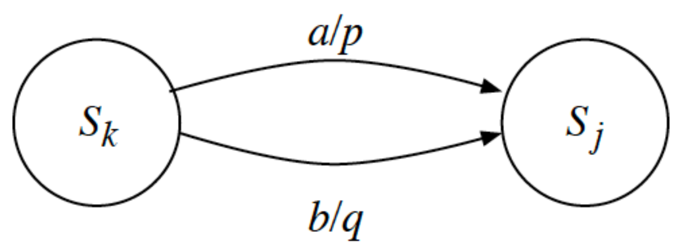
Functions: Transition and output functions

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

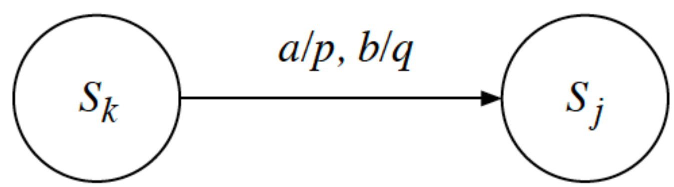
REPRESENTATION OF STATE-TRANSITION AND OUTPUT FUNCTIONS WITH STATE DIAGRAM



(a)



Complete state diagram



Simplified state diagram

(b)

What is the state diagram for this example?

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

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

Example: Traffic Light

- Output signals: NSlight, EWlight
- Input signals: NScar, EWcar
- State names: NSgreen, EWgreen (no yellow for now)
- Functionality: want light to change only if car is waiting at red light