

# Microprocessors

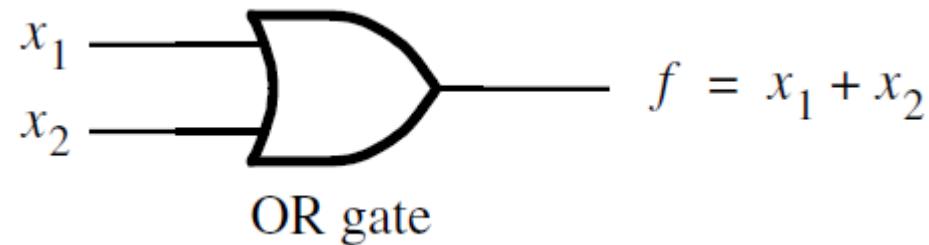
Tuba Ayhan

MEF University

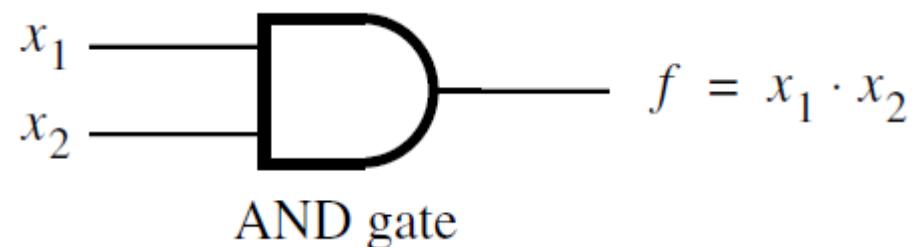
## Logic Circuits

Computer Organization and Embedded Systems, Hamacher et. al

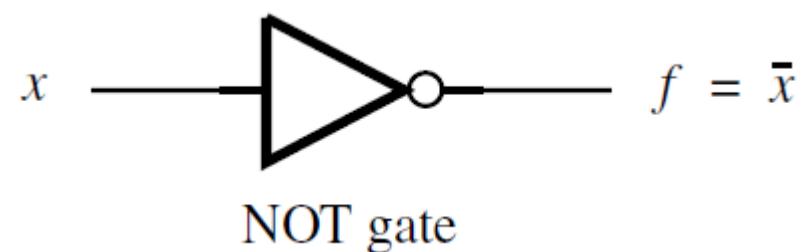
# Logic Gates



$x_1$	$x_2$	$f(x_1, x_2) = x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1



$x_1$	$x_2$	$f(x_1, x_2) = x_1 \cdot x_2$
0	0	0
0	1	0
1	0	0
1	1	1



$x$	$f(x)=x'$
0	1
1	0

# NAND and NOR gates

$x_1$	$x_2$	$f$
0	0	1
0	1	1
1	0	1
1	1	0

$x_1$	$x_2$	$f$
0	0	1
0	1	0
1	0	0
1	1	0

$$f = x_1 \uparrow x_2 = \overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$$

$$f = x_1 \downarrow x_2 = \overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$$

$$\overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$$

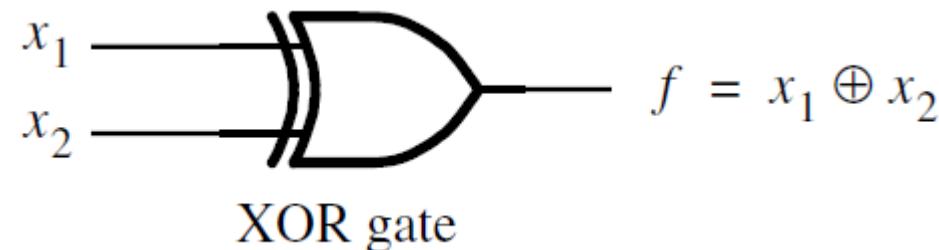


(a) NAND



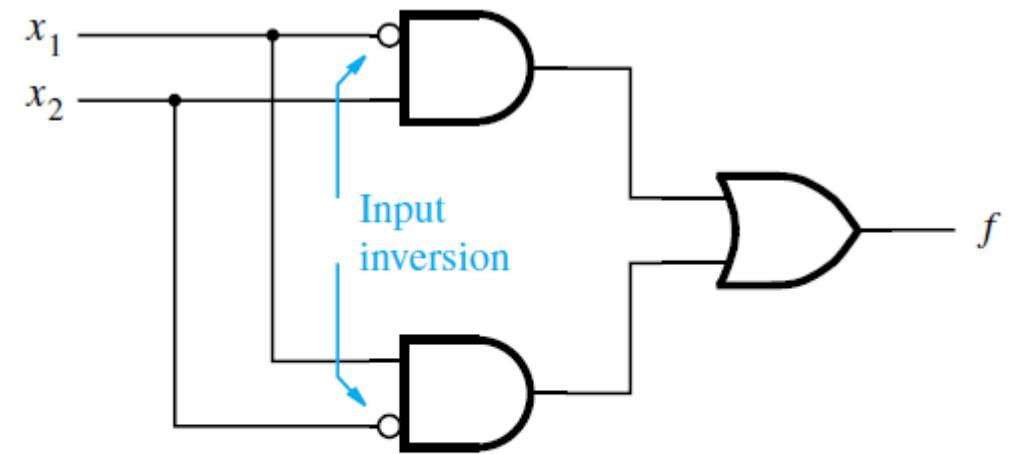
(b) NOR

$$\text{XOR: } f = x_1' \cdot x_2 + x_1 \cdot x_2'$$



$x_1$	$x_2$	$\bar{x}_1 \cdot x_2$	$x_1 \cdot \bar{x}_2$	$f = \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2$ $= x_1 \oplus x_2$
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

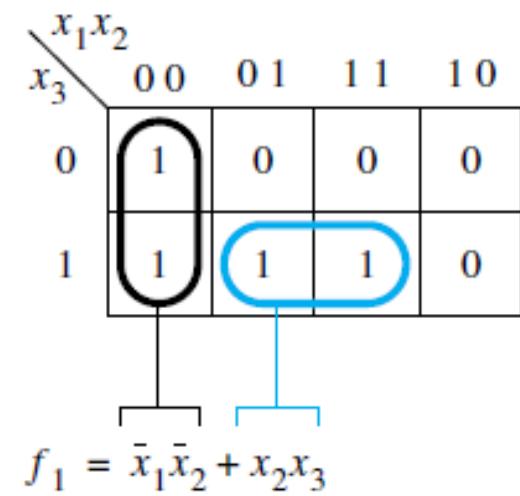
(b) Truth table construction of  $\bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2$



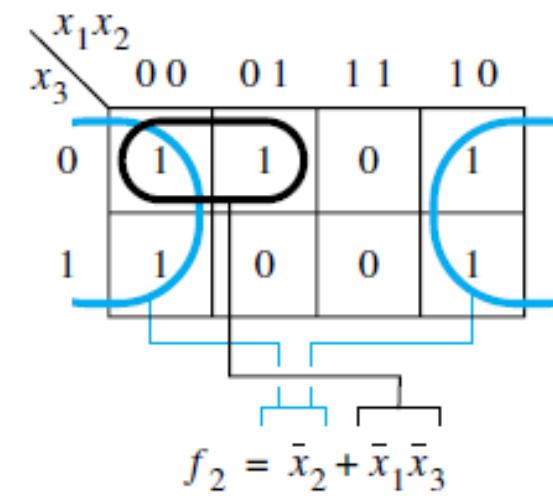
(a) Network for the XOR function

# Karnaugh maps

$x_1$	$x_2$	$x_3$	$f_1$	$f_2$
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

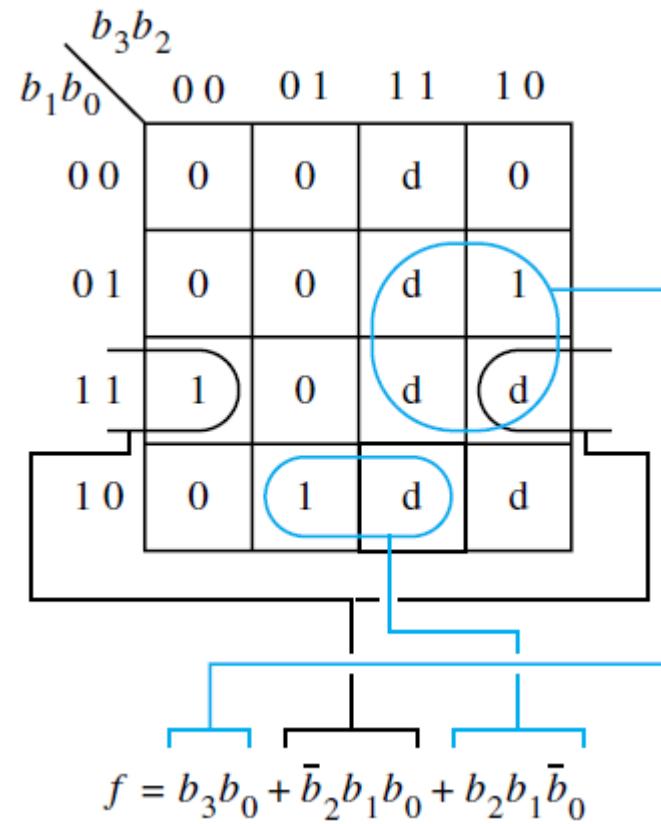


MEF University - EE308 Microprocessors, Tuba Ayhan



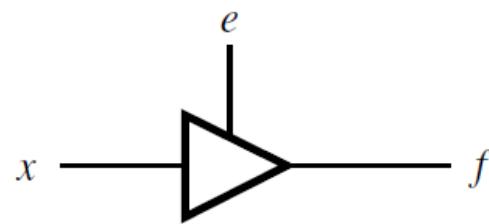
(a) Three-variable maps

Decimal digit represented	Binary coding $b_3 \ b_2 \ b_1 \ b_0$	$f$
0	0 0 0 0	0
1	0 0 0 1	0
2	0 0 1 0	0
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	0
6	0 1 1 0	1
7	0 1 1 1	0
8	1 0 0 0	0
9	1 0 0 1	1
unused	1 0 1 0	d
	1 0 1 1	d
	1 1 0 0	d
	1 1 0 1	d
	1 1 1 0	d
	1 1 1 1	d

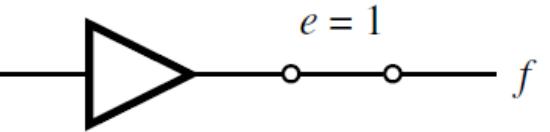
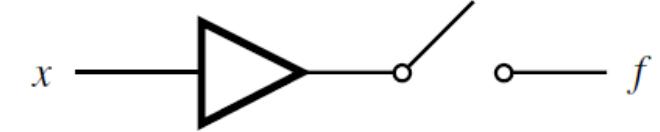


# Tri-state buffer

- A tri-state buffer has three states:
- 0
- 1
- Z (high-impedance): the output terminal is electrically disconnected from the input.



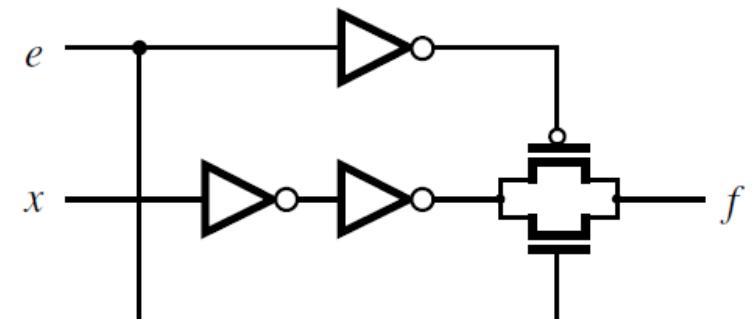
(a) Symbol



(b) Equivalent circuit

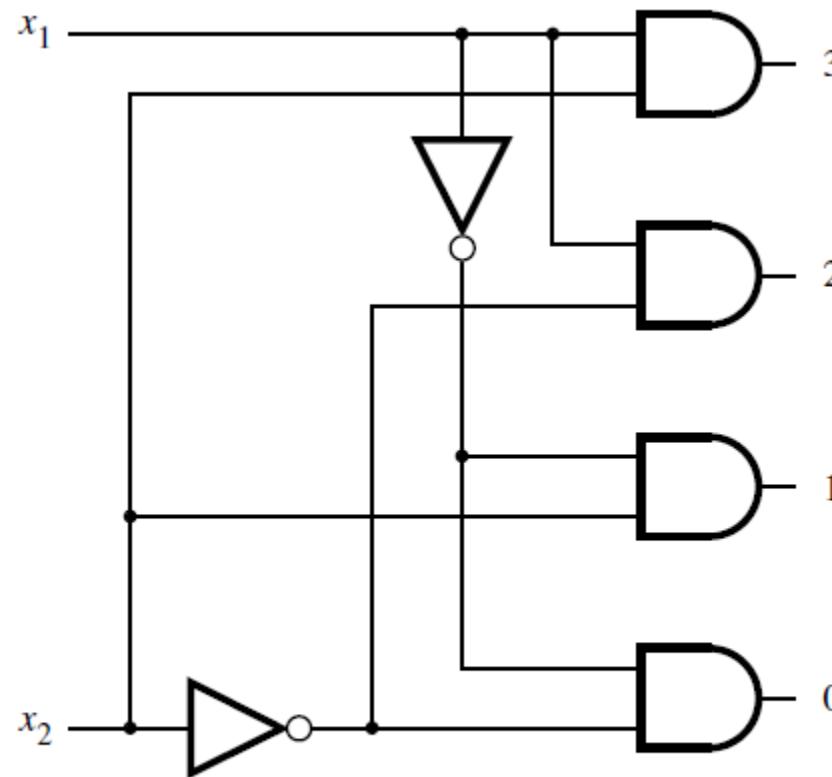
$e$	$x$	$f$
0	0	Z
0	1	Z
1	0	0
1	1	1

(c) Truth table

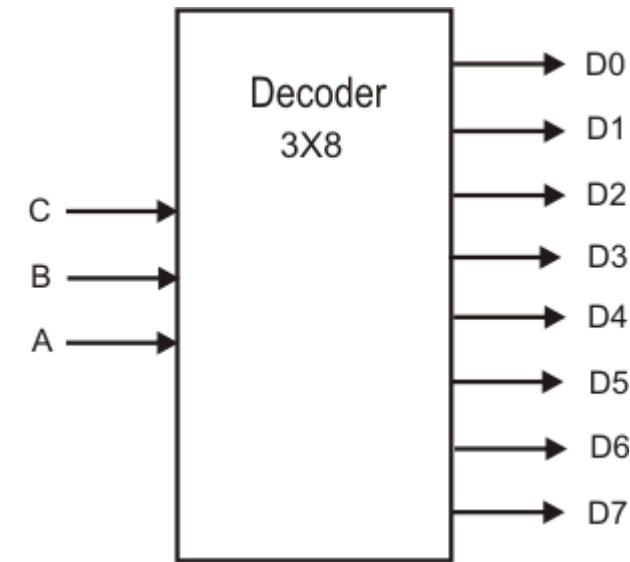


(d) Implementation

# Decoder: $n \times 2^n$



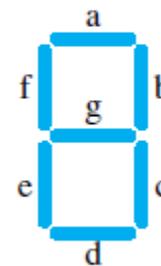
$x_1$	$x_2$	Active output
0	0	0
0	1	1
1	0	2
1	1	3



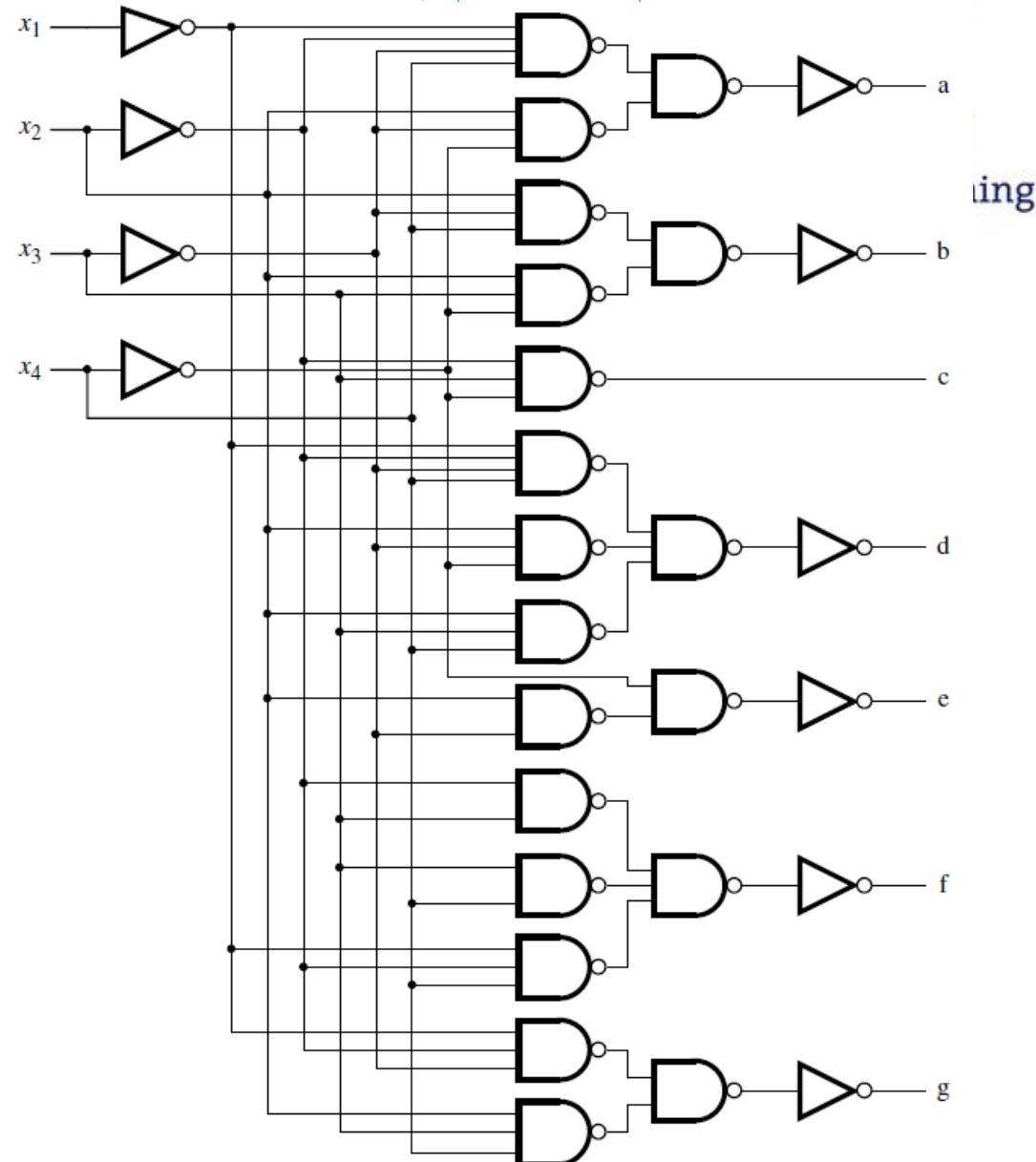
A three-input to eight-output decoder

**Figure A.35** A two-input to four-output decoder.

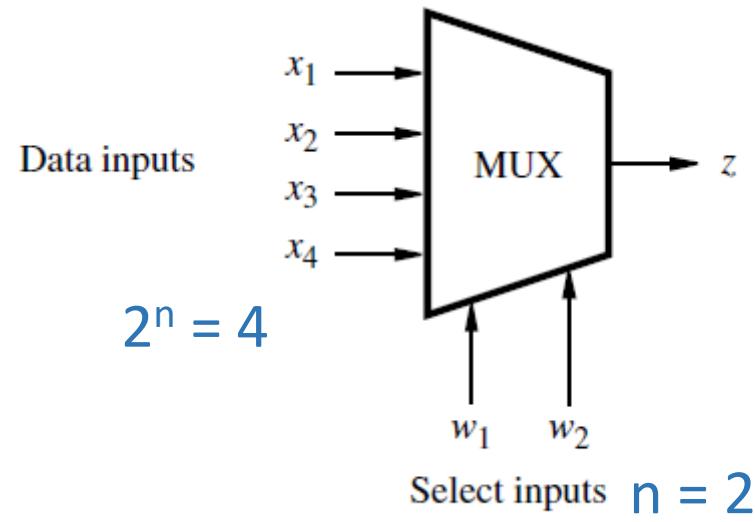
# BCD to 7-segment decoder



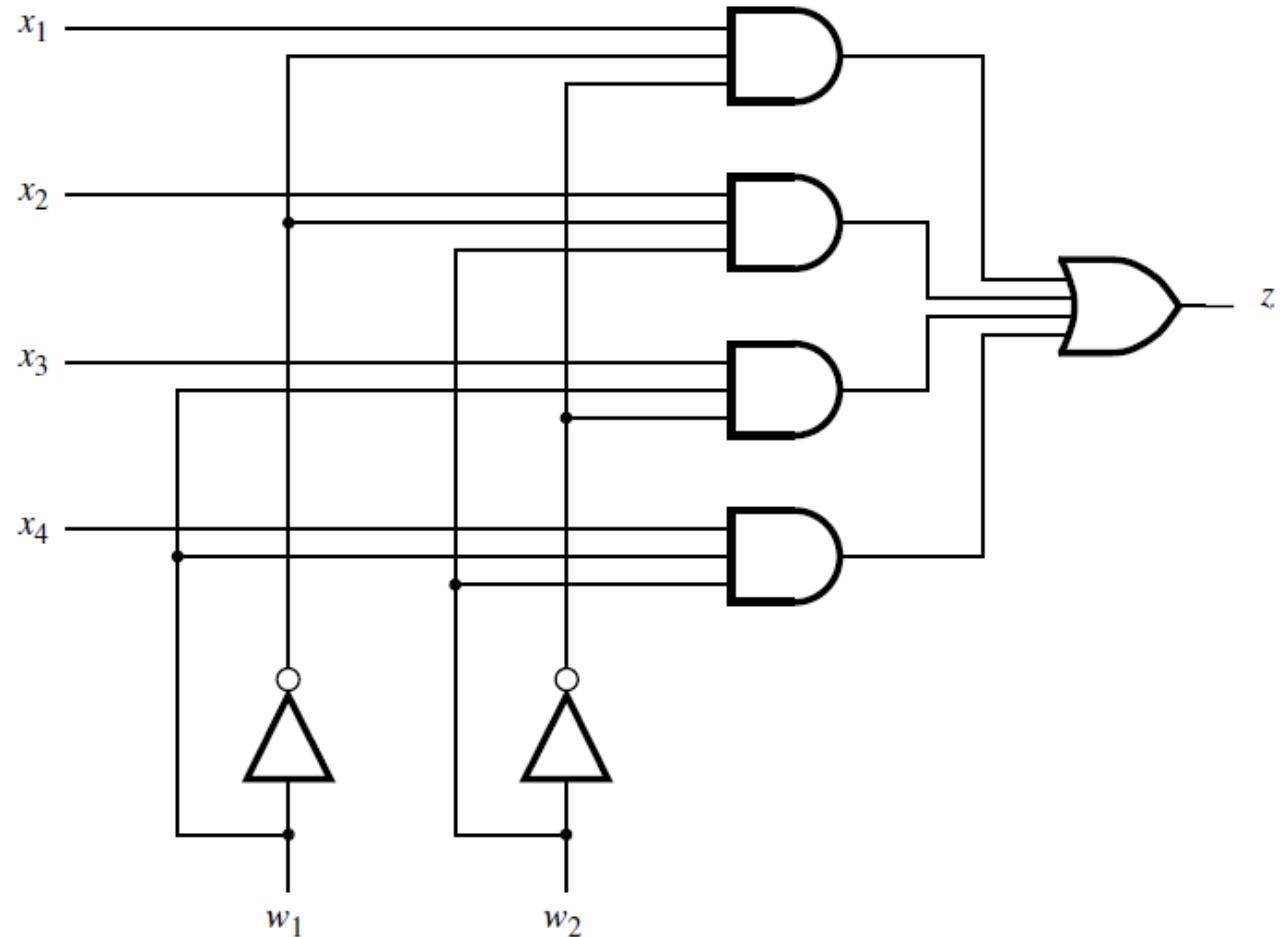
No.	$x_1$	$x_2$	$x_3$	$x_4$	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



# Multiplexer: $2^n$ -to-1



$w_1$	$w_2$	$z$
0	0	$x_1$
0	1	$x_2$
1	0	$x_3$
1	1	$x_4$



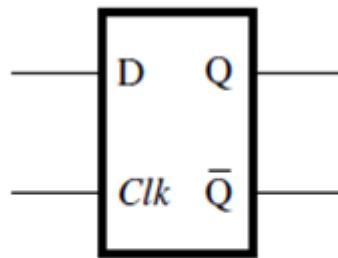
# Microprocessors

Tuba Ayhan

MEF University

## Sequential Logic Circuits

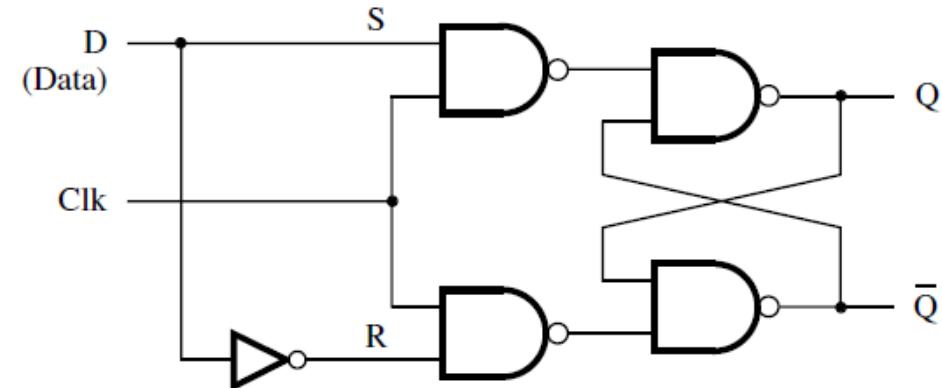
Computer Organization and Embedded Systems, Hamacher et. al



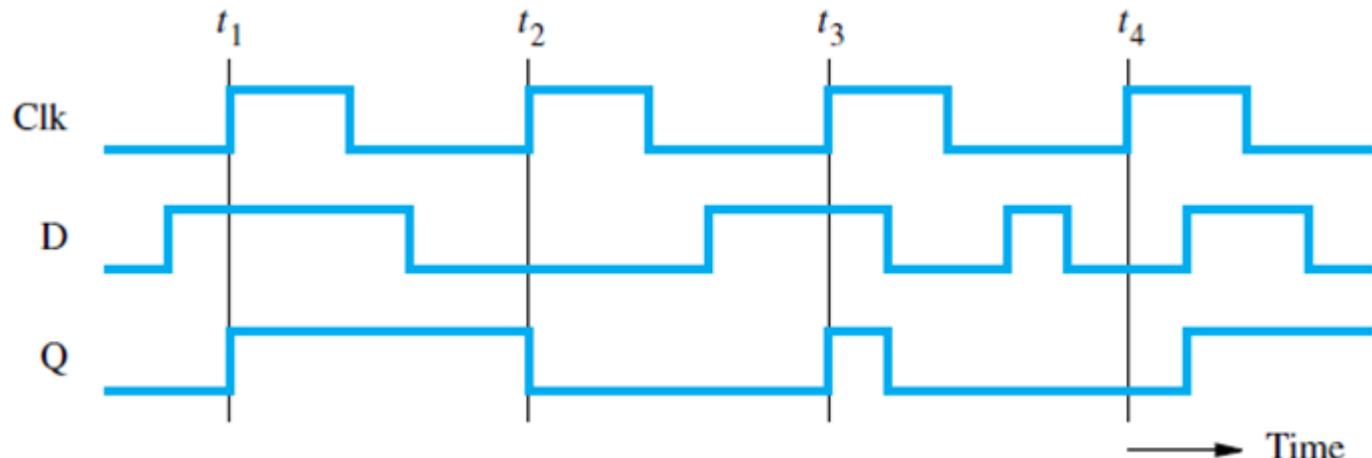
(c) Graphical symbol

Clk	D	$Q(t+1)$
		$Q(t)$
0	x	
1	0	0
1	1	1

(b) Truth table



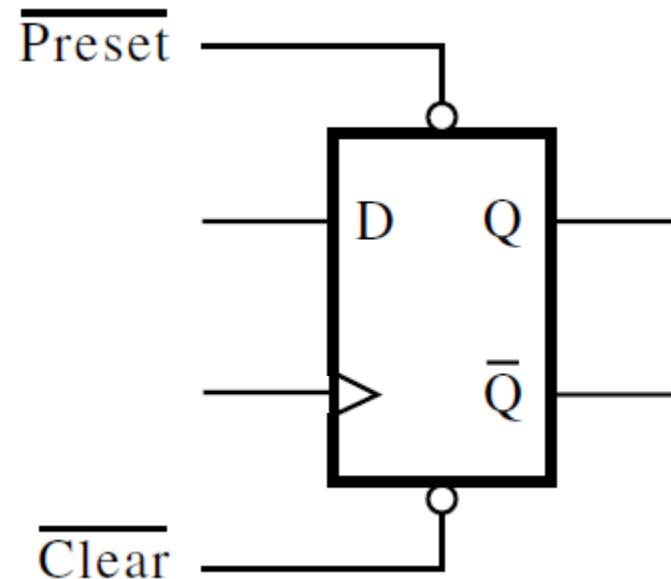
(a) Circuit



(d) Timing diagram

# Pos. Edge triggered D ff with preset and clear

Your Freedom in Learning



**FUNCTION TABLE**

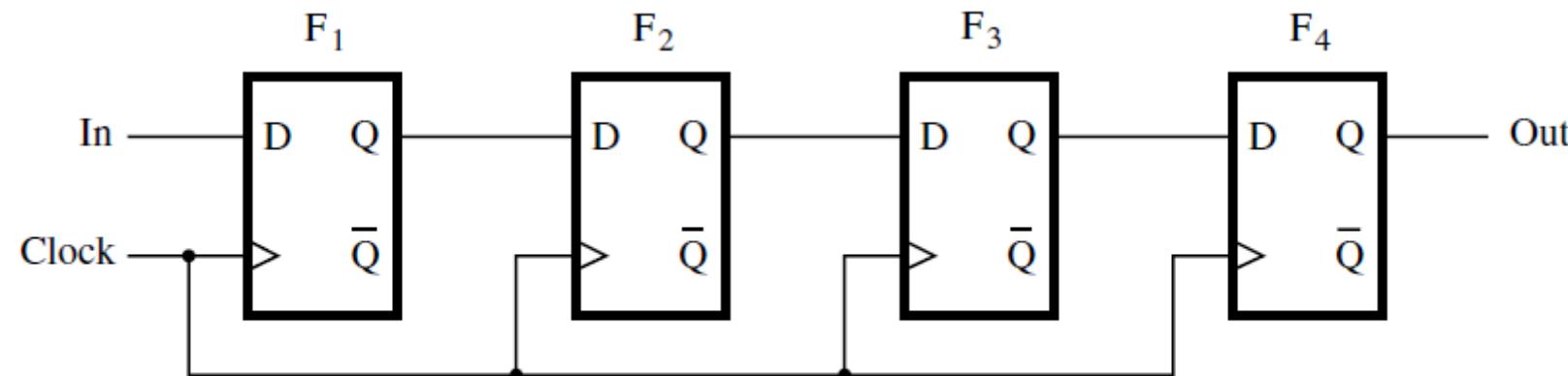
INPUTS				OUTPUTS	
PRE	CLR	CLK	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	$H^{(1)}$	$H^{(1)}$
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	$Q_0$	$\bar{Q}_0$

- (1) This configuration is nonstable; that is, it does not persist when  $\overline{\text{PRE}}$  or  $\overline{\text{CLR}}$  returns to its inactive (high) level.

*Ref. Datasheet*

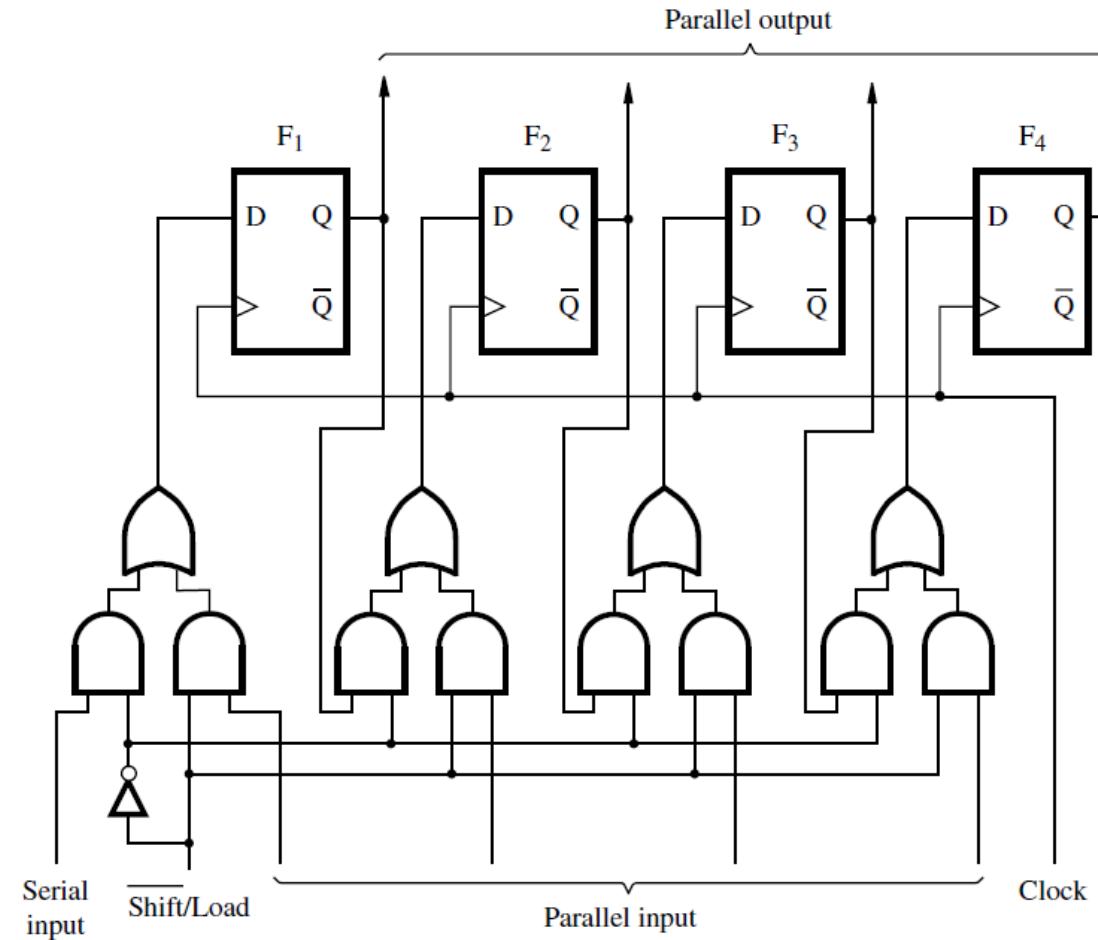
# Registers and shift registers

- *Register:* A number of flip-flops arranged into a common structure
- Operation of ffs in a register is synchronized by a common clock.
  - Data are written (loaded) into or read from all flip-flops at the same time.

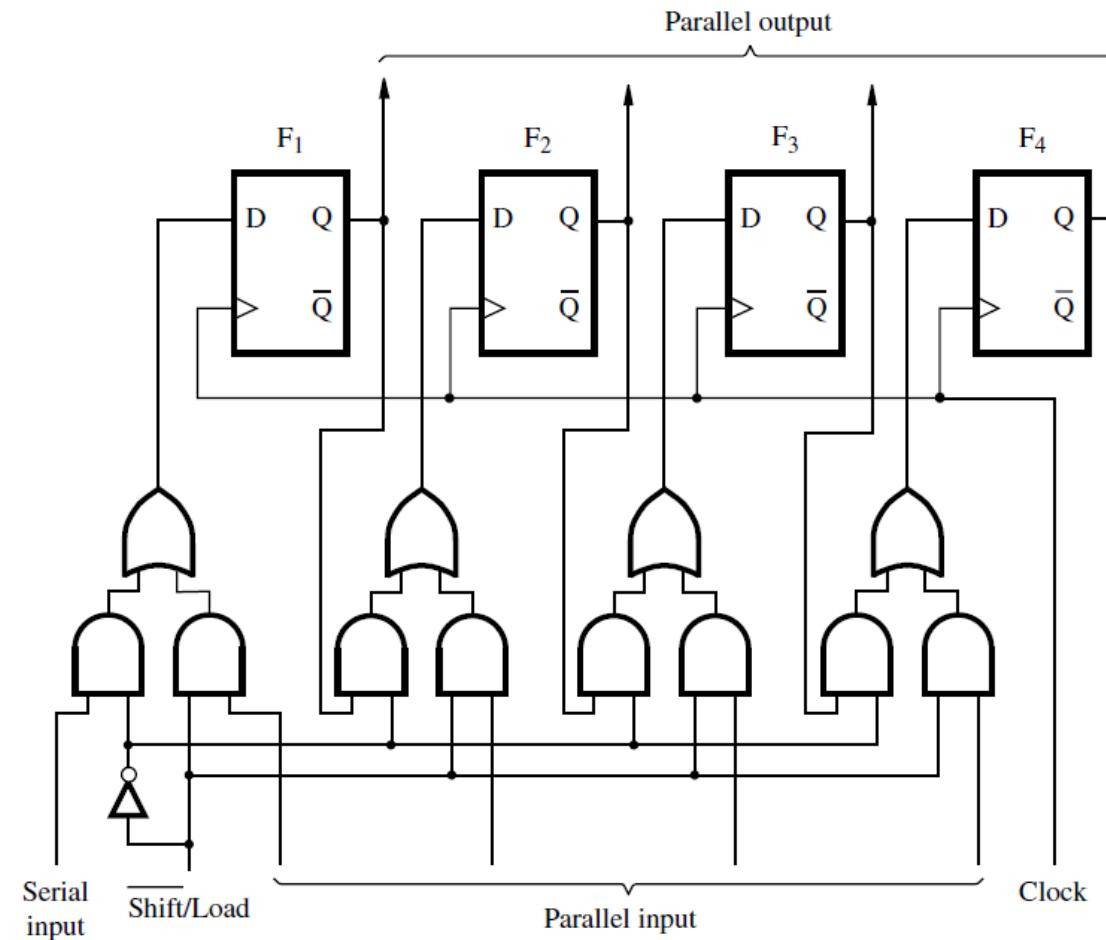


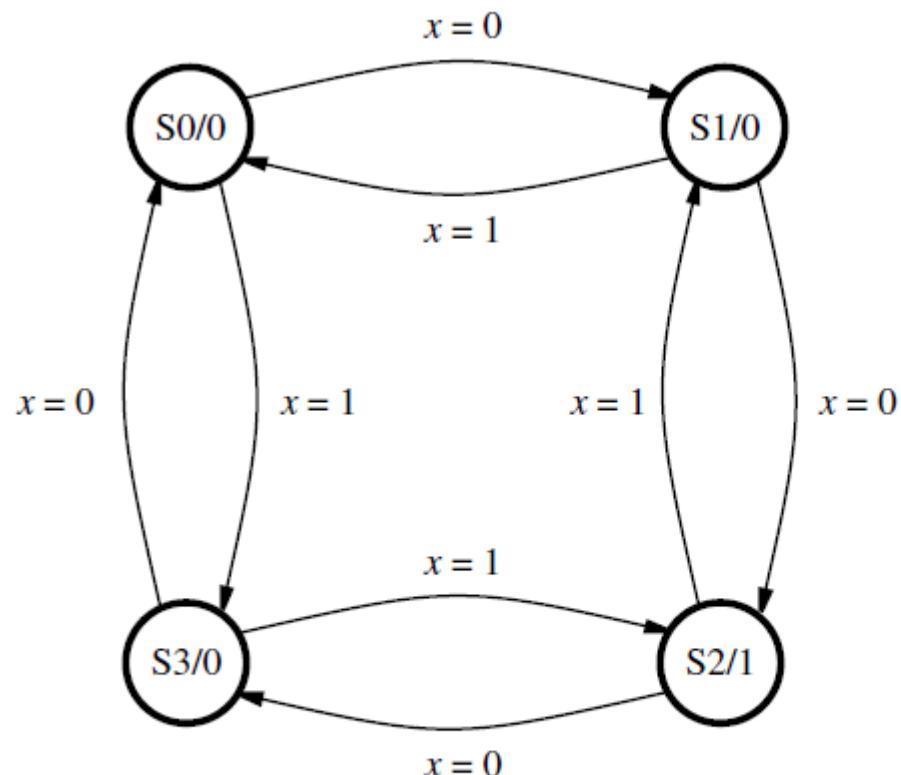
**Figure A.32** A simple shift register.

# What is this?



# Parallel-Access shift register





**Figure A.46** State diagram of a mod-4 up/down counter that detects the count of 2.

Present state	Next state		Output $z$
	$x = 0$	$x = 1$	
S0	S1	S3	0
S1	S2	S0	0
S2	S3	S1	1
S3	S0	S2	0

**Figure A.47** State table for the example of the up/down counter.