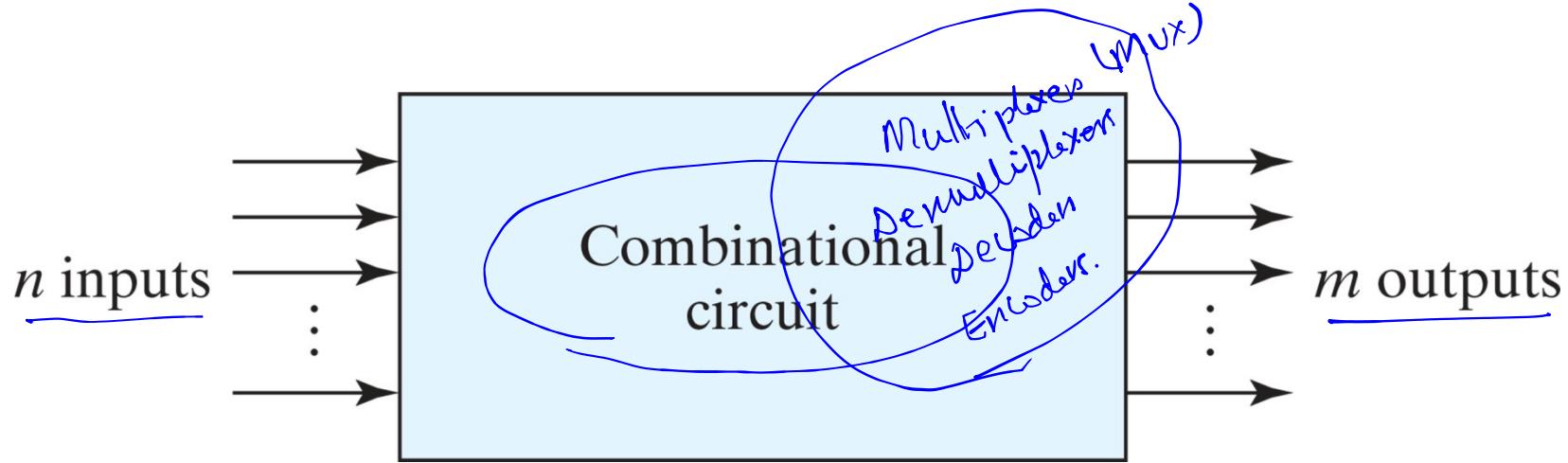




## Functionally Complete Sets:

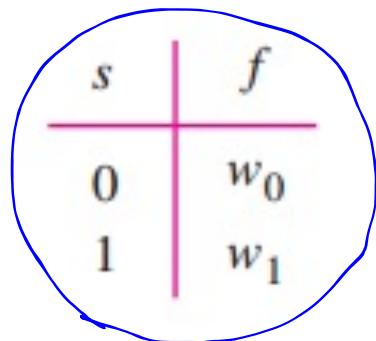
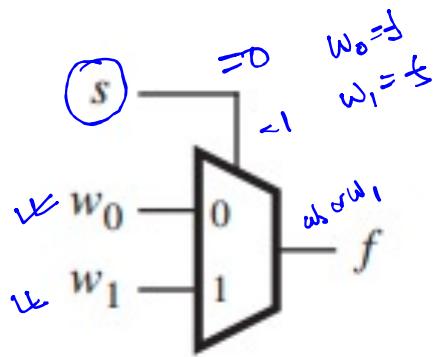
1. {NAND}
  2. {AND, NOT}
  3. {NOR}
  4. {OR, NOT}
  5. {AOI} --- AND OR INVERT  $(\overline{A \cdot B + C \cdot D})$
  6. {OAI} --- OR AND INVERT  $((A + B) \cdot (C + D))$
- easy to design*
- any function*
- there are other gates - logic functions*

## Combinational Circuits:



## Multiplexer also called MUX:

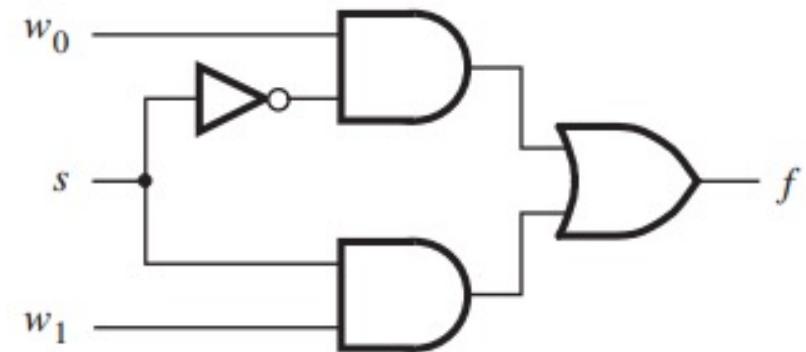
A multiplexer helps in transmitting, selectively, any of the several available signal inputs to the only available output.



2:1 Multiplexer

$$f = \bar{s} \cdot w_0 + s \cdot w_1$$

Realization



## 4:1 Multiplexers

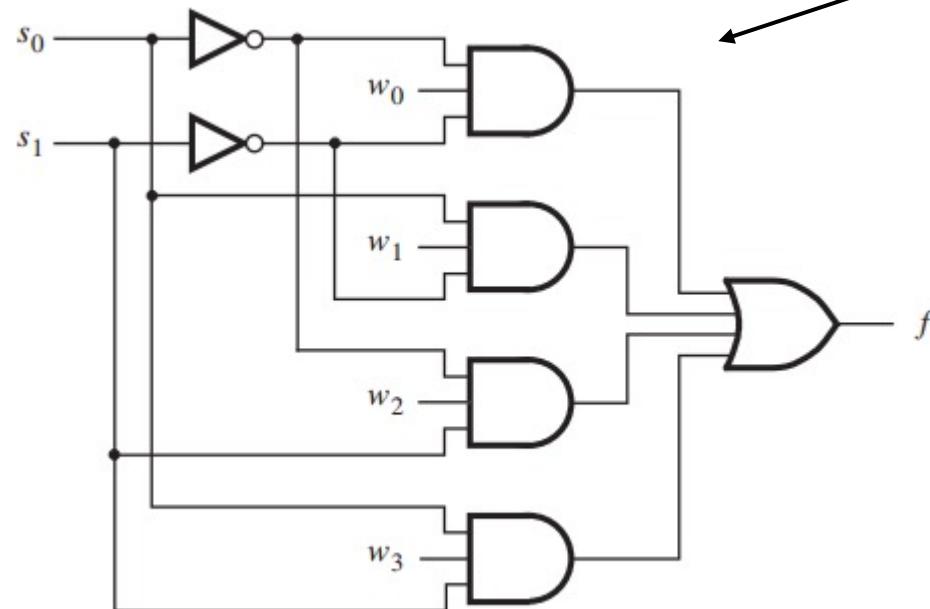
$s_1$	$s_0$	$f$
0	0	$w_0$
0	1	$w_1$
1	0	$w_2$
1	1	$w_3$

Shannon's theorem.

$$f = \bar{s}_1 \bar{s}_0 w_0 + \bar{s}_1 s_0 w_1 + s_1 \bar{s}_0 w_2 + s_1 s_0 w_3$$

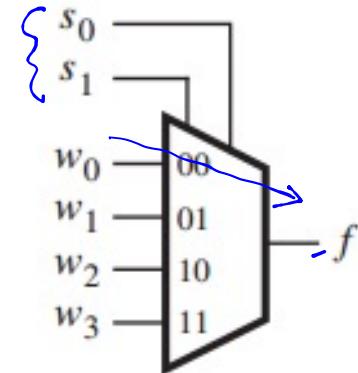
↓ Shannon's Theorem

$$f = \bar{s}_1 \cdot (\bar{s}_0 w_0 + s_0 w_1) + s_1 \cdot (\bar{s}_0 w_2 + s_0 w_3)$$



control  
inputs

Symbol



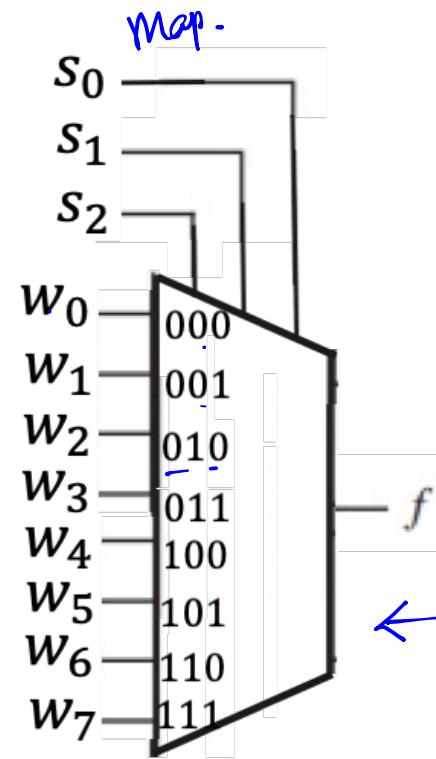
4:1 MUX

## 8:1 Multiplexers:

For chip designer.

Truth Table

$s_2$	$s_1$	$s_0$	$f$
0	0	0	$w_0$
0	0	1	$w_1$
0	1	0	$w_2$
0	1	1	$w_3$
1	0	0	$w_4$
1	0	1	$w_5$
1	1	0	$w_6$
1	1	1	$w_7$



map the Truth Table  
minimize nothing  
design is comfortable  
while designing in  
Bread Board or  
PC card

$$f = \bar{s}_2 \bar{s}_1 \bar{s}_0 w_0 + \bar{s}_2 \bar{s}_1 s_0 w_1 + \bar{s}_2 s_1 \bar{s}_0 w_2 + \bar{s}_2 s_1 s_0 w_3 + s_2 \bar{s}_1 \bar{s}_0 w_4 + s_2 \bar{s}_1 s_0 w_5 + s_2 s_1 \bar{s}_0 w_6 + s_2 s_1 s_0 w_7$$

$$f = \bar{s}_2 \cdot (\bar{s}_1 \bar{s}_0 w_0 + \bar{s}_1 s_0 w_1 + s_1 \bar{s}_0 w_2 + s_1 s_0 w_3) + s_2 \cdot (\bar{s}_1 \bar{s}_0 w_4 + \bar{s}_1 s_0 w_5 + s_1 \bar{s}_0 w_6 + s_1 s_0 w_7)$$

$$f = \bar{s}_2 \cdot (\bar{s}_1 \cdot (\bar{s}_0 w_0 + s_0 w_1) + s_1 \cdot (\bar{s}_0 w_2 + s_0 w_3)) + s_2 \cdot (\bar{s}_1 \cdot (\bar{s}_0 w_4 + s_0 w_5) + s_1 \cdot (\bar{s}_0 w_6 + s_0 w_7))$$

# Boolean Functions Using Multiplexers

- Shannon's Expansion Theorem. ----- Nested

variable  
entered

$2^{n-1}$

use this  
extensively

$$f(w_{n-1}, \dots, w_0) = \overline{w_0} \cdot f(w_{n-1}, \dots, w_1, 0) + w_0 \cdot f(w_{n-1}, \dots, w_1, 1)$$

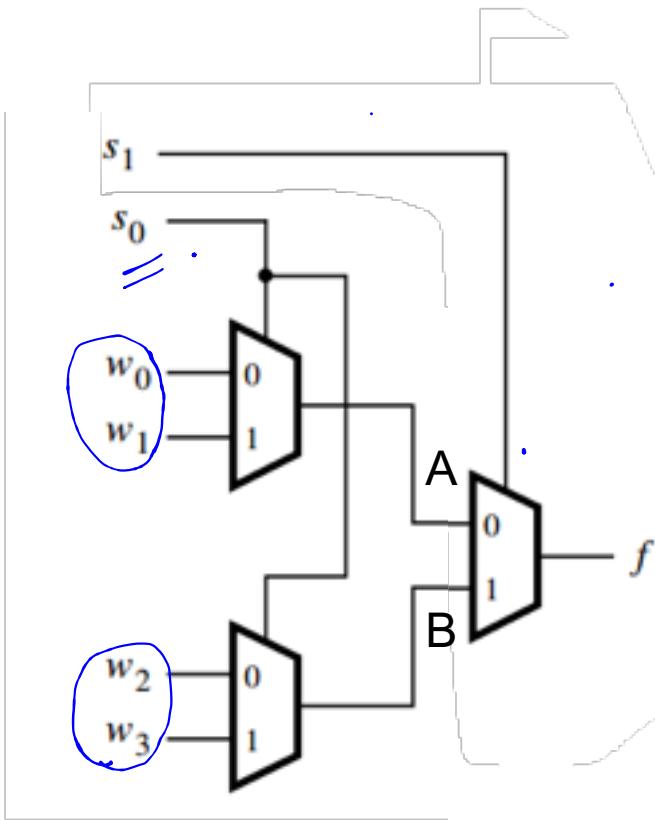
$$\begin{aligned} f(w_{n-1}, \dots, w_0) &= \overline{w_1} \cdot \overline{w_0} \cdot f(w_{n-1}, \dots, w_2, 0, 0) + \overline{w_1} \cdot w_0 \cdot f(w_{n-1}, \dots, w_2, 0, 1) \\ &\quad \vdots \\ &= w_1 \cdot \overline{w_0} \cdot f(w_{n-1}, \dots, w_2, 1, 0) + w_1 \cdot w_0 \cdot f(w_{n-1}, \dots, w_2, 1, 1) \end{aligned}$$

$$\begin{aligned} f(w_{n-1}, \dots, w_0) &= \overline{w_2} \cdot \overline{w_1} \cdot \overline{w_0} \cdot f(w_{n-1}, \dots, w_3, 0, 0, 0) + \overline{w_2} \cdot \overline{w_1} \cdot w_0 \cdot f(w_{n-1}, \dots, w_3, 0, 0, 1) \\ &= \overline{w_2} \cdot w_1 \cdot \overline{w_0} \cdot f(w_{n-1}, \dots, w_3, 0, 1, 0) + \overline{w_2} \cdot w_1 \cdot w_0 \cdot f(w_{n-1}, \dots, w_3, 0, 1, 1) \\ &= w_2 \cdot \overline{w_1} \cdot \overline{w_0} \cdot f(w_{n-1}, \dots, w_3, 1, 0, 0) + w_2 \cdot \overline{w_1} \cdot w_0 \cdot f(w_{n-1}, \dots, w_3, 1, 0, 1) \\ &= w_2 \cdot w_1 \cdot \overline{w_0} \cdot f(w_{n-1}, \dots, w_3, 1, 1, 0) + w_2 \cdot w_1 \cdot w_0 \cdot f(w_{n-1}, \dots, w_3, 1, 1, 1) \end{aligned}$$

## 4:1 Multiplexer Using 2:1 Multiplexer

$$f = \bar{s}_1 \cdot (\bar{s}_0 w_0 + s_0 w_1) + s_1 \cdot (\bar{s}_0 w_2 + s_0 w_3)$$

A                            B



# Multiplexer (HW)

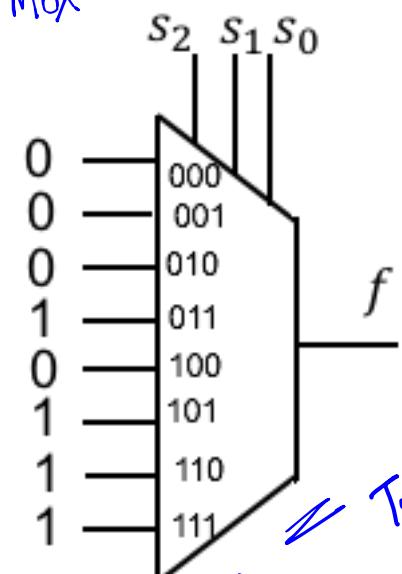
- Draw gate level circuit diagram (only AND, OR, NOT gates) for 4:1 and 8:1 multiplexers
- Design 8:1 multiplexer using only 4:1 multiplexers
- Design 8:1 multiplexer using 4:1 and 2:1 multiplexers
- Design 8:1 multiplexer using only two 4:1 multiplexers and basic gates
- Similarly, design various combinations of 16:1 multiplexers

## Logic Functions Using Multiplexers:

Realize a 3-input majority function using 4:1 and 2:1 Multiplexers

$s_2$	$s_1$	$s_0$	$f$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

8:1 MUX



are map it  
Truth Table  
complete set

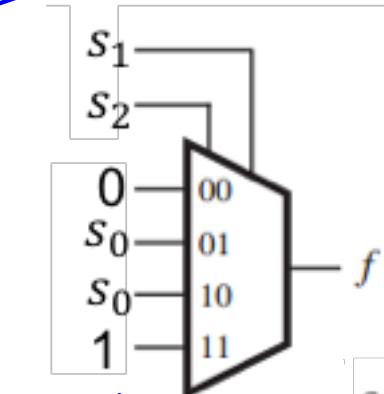
Truth Table  
map useful

$s_2$	$f$
0	$s_1 \cdot s_0$
1	$s_1 + s_0$

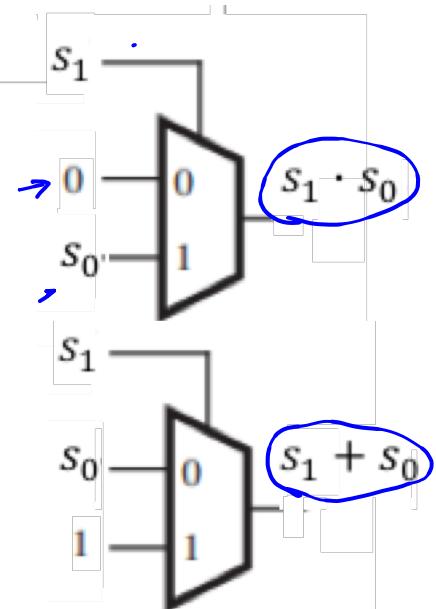
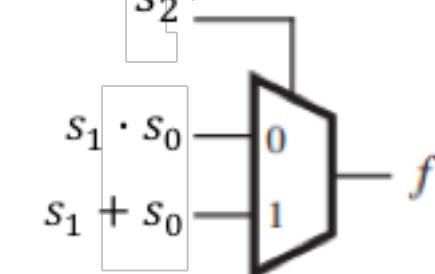
4:1

variable  
entered  
Table.

$s_2$	$s_1$	$f$
0	0	0
0	1	$s_0$
1	0	$s_0$
1	1	1



resource  
utilization

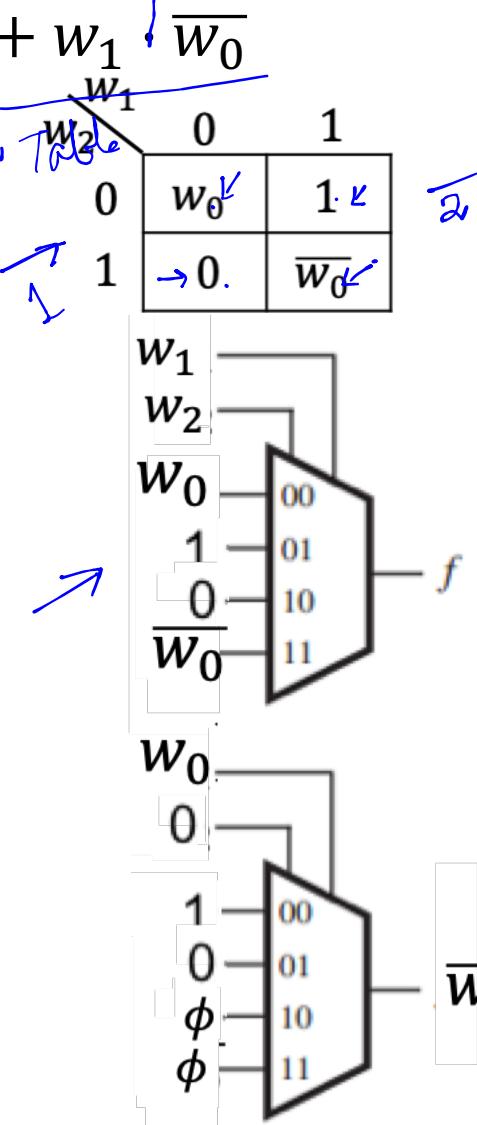
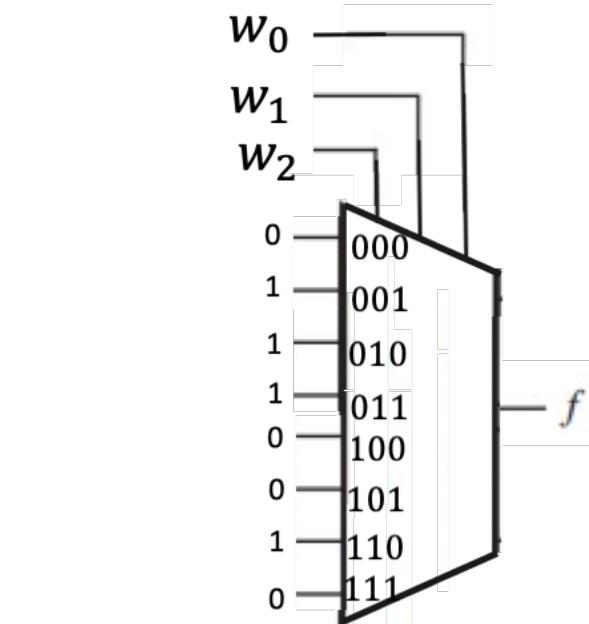


## Basic Functions Using Multiplexers:

$$f(w_2, w_1, w_0) = \overline{w_2} \cdot w_0 + w_1 \cdot \overline{w_0}$$

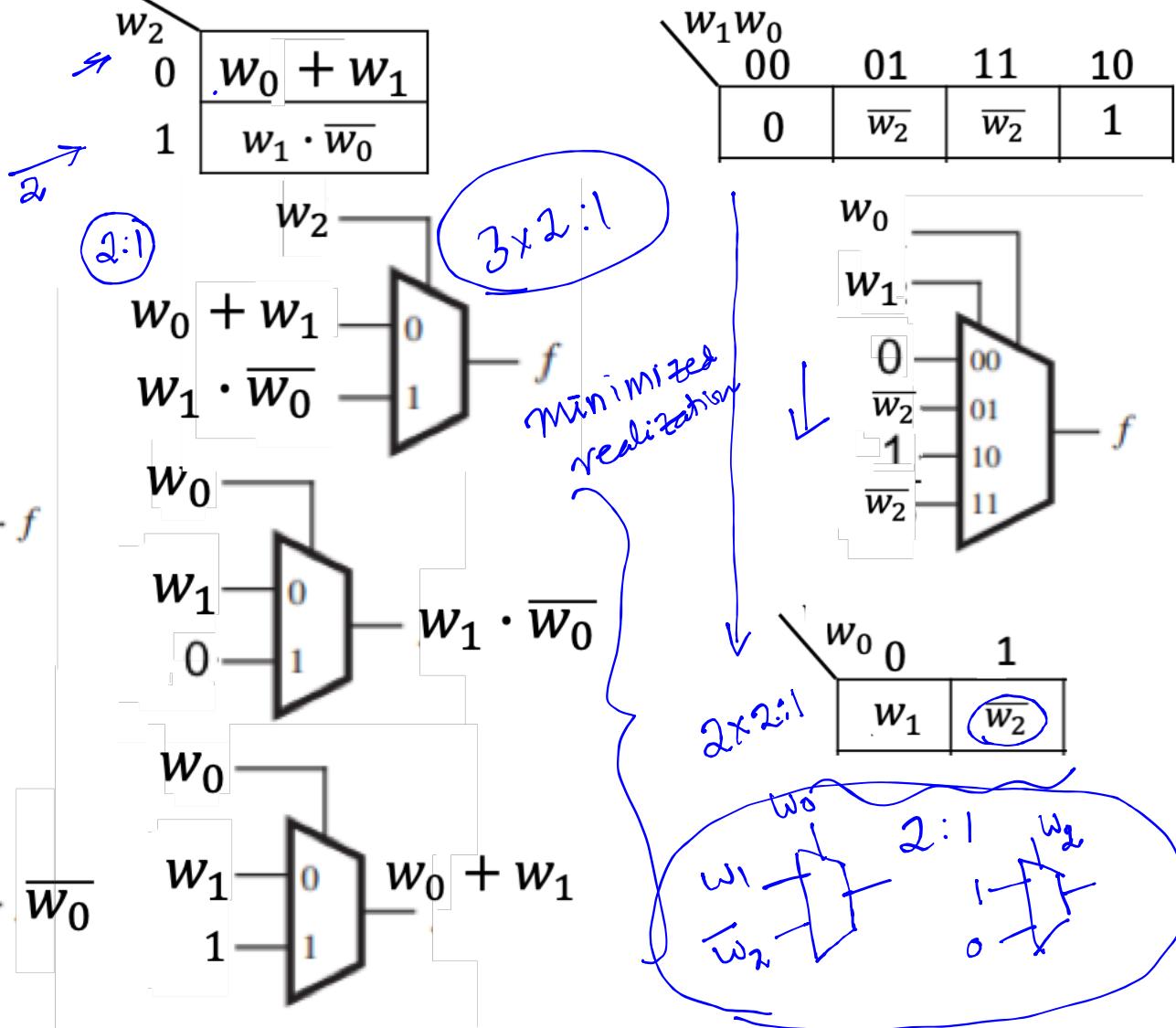
Truth Table:

$w_1 w_0$		00	01	11	10
$w_2$	0	0	1	1	1
1	0	0	0	0	1



IIITD ECE 111 Section A

Alternative:



## Boolean Functions Using Multiplexers:

- $f(w, x, y) = \sum m(3, 5, 6, 7)$  using minimum number of 2:1 MUXs (Don't use any logic gates)

w	x	y	00	01	11	10
0	0	0	0	0	1	0
1	0	1	1	1	1	1

positive terms  
complements

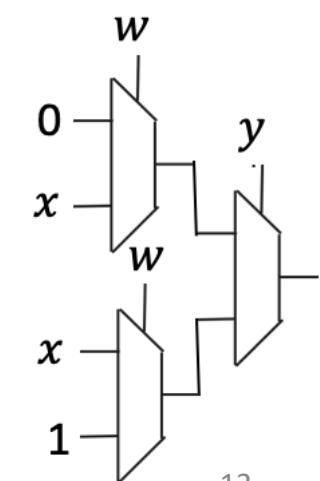
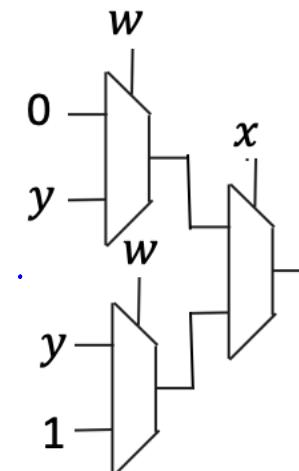
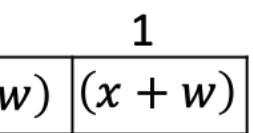
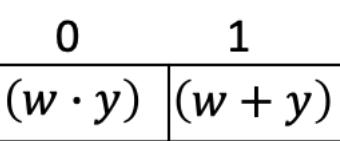
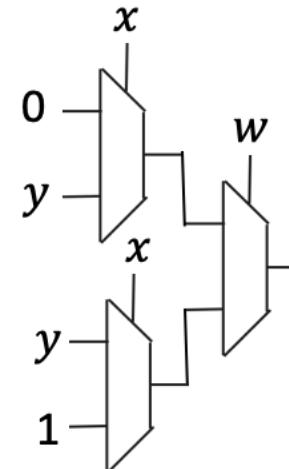
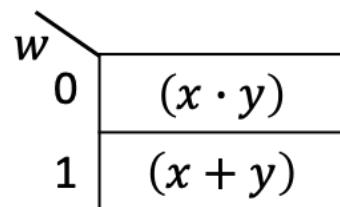
$$f(w, x, y) = \underline{wy} + \underline{wx} + \underline{xy}$$

Shannon decomposition

$$= \bar{w} \cdot (x \cdot y) + w \cdot (x + y)$$

$$= \bar{x} \cdot (w \cdot y) + x \cdot (w + y)$$

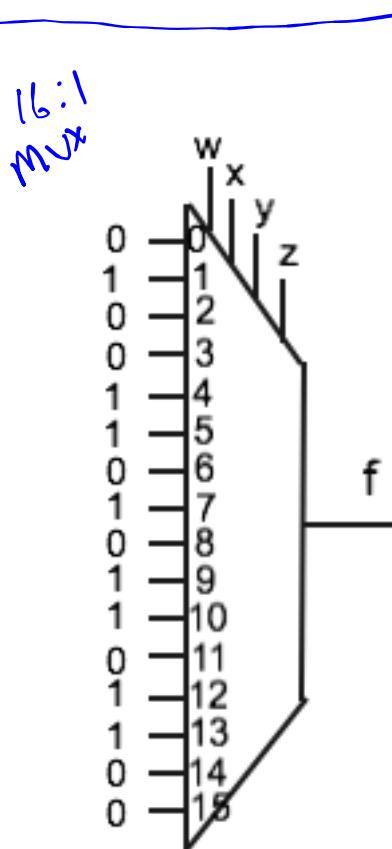
$$= \bar{y} \cdot (x \cdot w) + y \cdot (x + w)$$



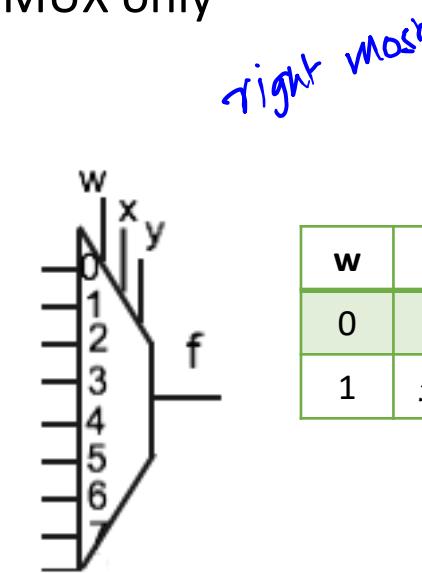
# Logic Functions Using Multiplexers:

- $f(w, x, y, z) = \sum m(1, 4, 5, 7, 9, 10, 12, 13)$  using 4:1 MUX and 2:1 MUX only

w	x	y	z	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

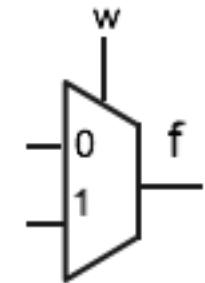


w	x	y	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

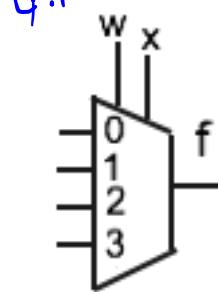


TT K-map

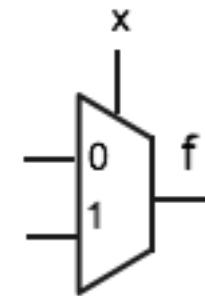
w	f
0	$x(\bar{y} + z) + \bar{y}z$
1	$x\bar{y} + \bar{x}yz + \bar{y}z$



w	x	f
0	0	$\bar{y}z$
0	1	$\bar{y} + z$
1	0	$\bar{y}z + y\bar{z}$
1	1	$\bar{y}$

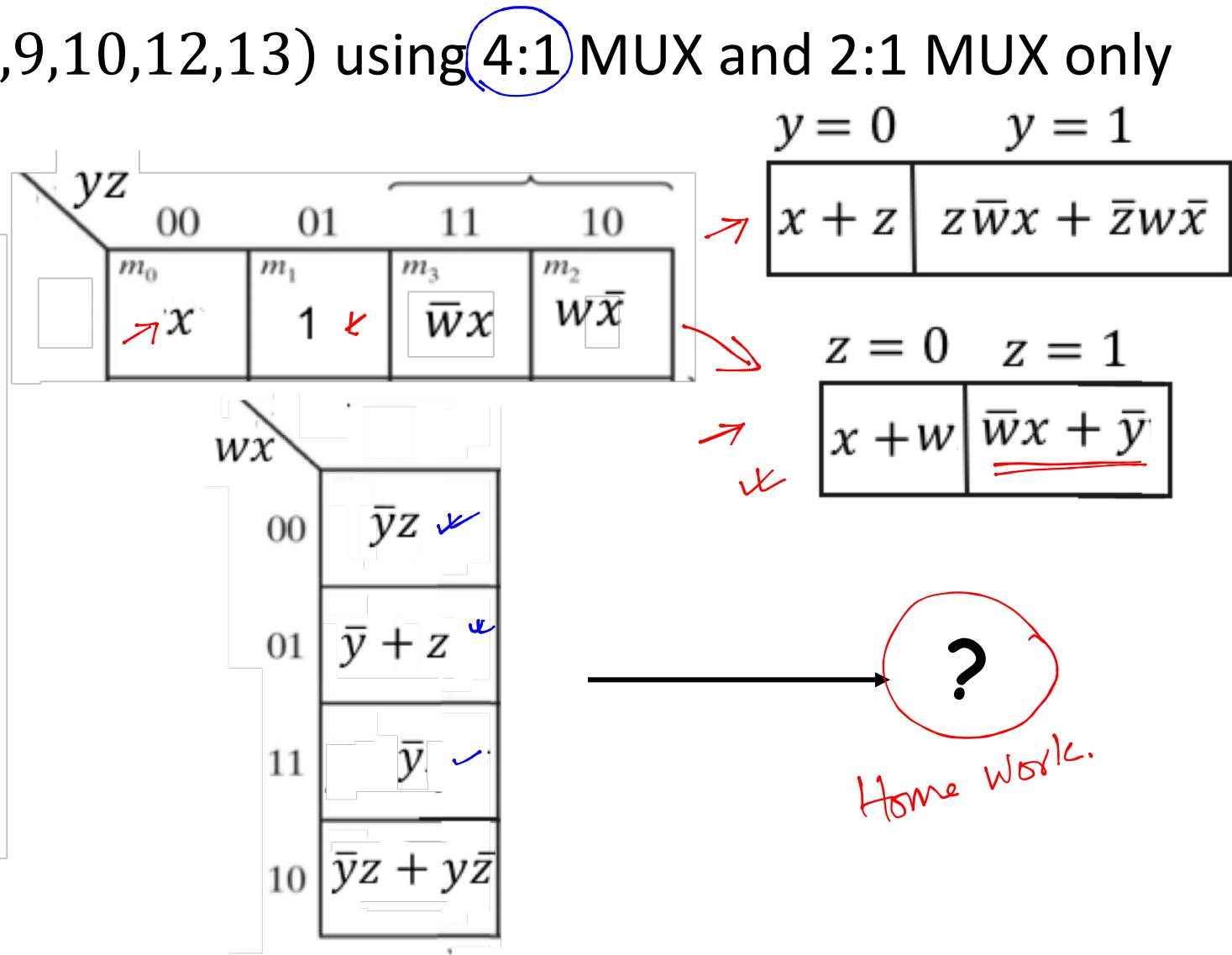
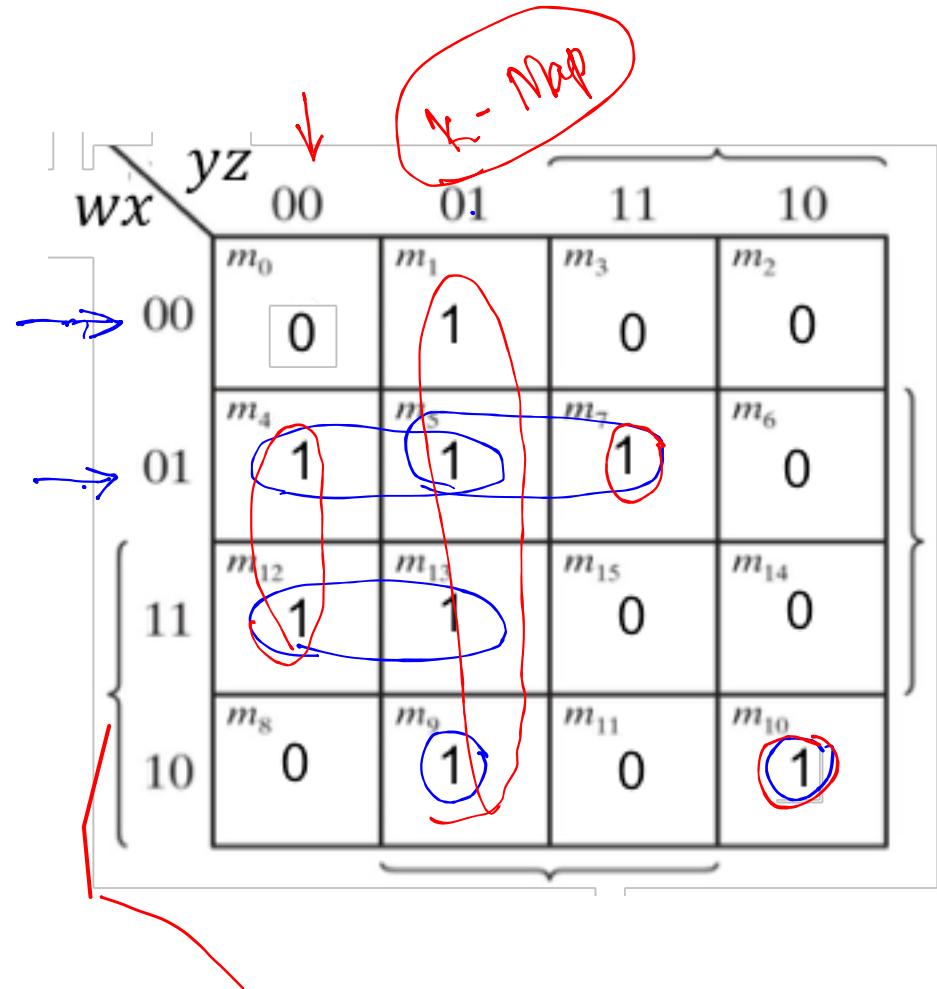


x	f
0	$\bar{y}z + wy\bar{z}$
1	$\bar{y} + \bar{w}z$



## Boolean Functions Using Multiplexers:

- $f(w, x, y, z) = \sum m(1, 4, 5, 7, 9, 10, 12, 13)$  using 4:1 MUX and 2:1 MUX only



## Boolean Circuits Using Multiplexers (HW)

- $f(w, x, y, z) = \sum m (1, 4, 5, 7, 9, 12, 13)$  using 8:1 MUX and logic gates only
- Design 1-bit full adder using 4:1 multiplexers
- Design a 2-input XNOR gate using 4:1 multiplexer
- Design a 2-input XNOR gate using 2:1 multiplexers

## Decoder:

- A  $n$ -to- $m$ -line decoder is a combinational circuit that converts binary information from  $n$  input lines to a maximum of  $m \leq 2^n$  unique output lines.
- Their purpose is to generate the  $2^n$  (or fewer) minterms of  $n$  input variables.

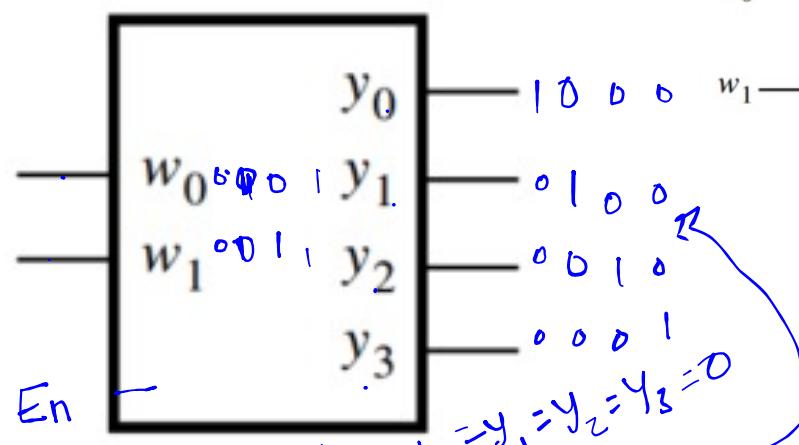
2:4 Decoder: ↙ → 2 bits are input

Truth Table:

		4 lines			
$w_1$	$w_0$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

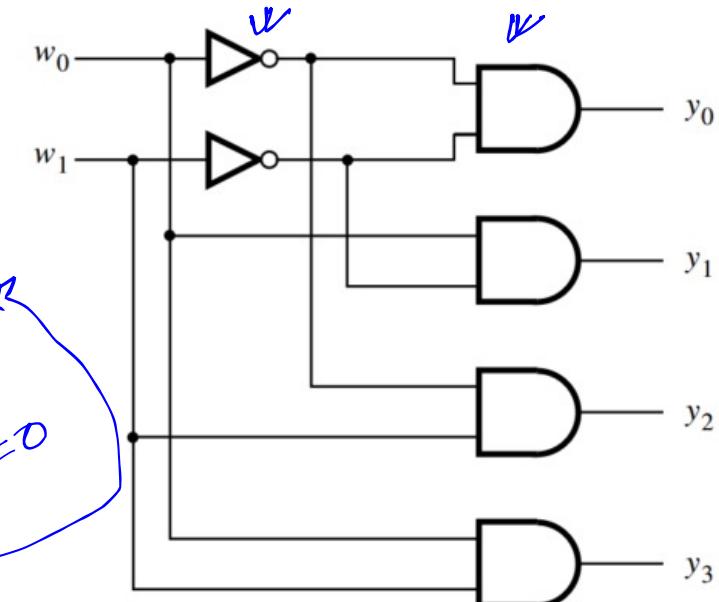
↙ 2 : 4 Decoder

Functional Block Diagram:



Coding → reduces inputs  
Decoding → expands the input combination

Schematic Diagram:



## 2:4 Decoder with Enable input:

$111111 \rightarrow 9$

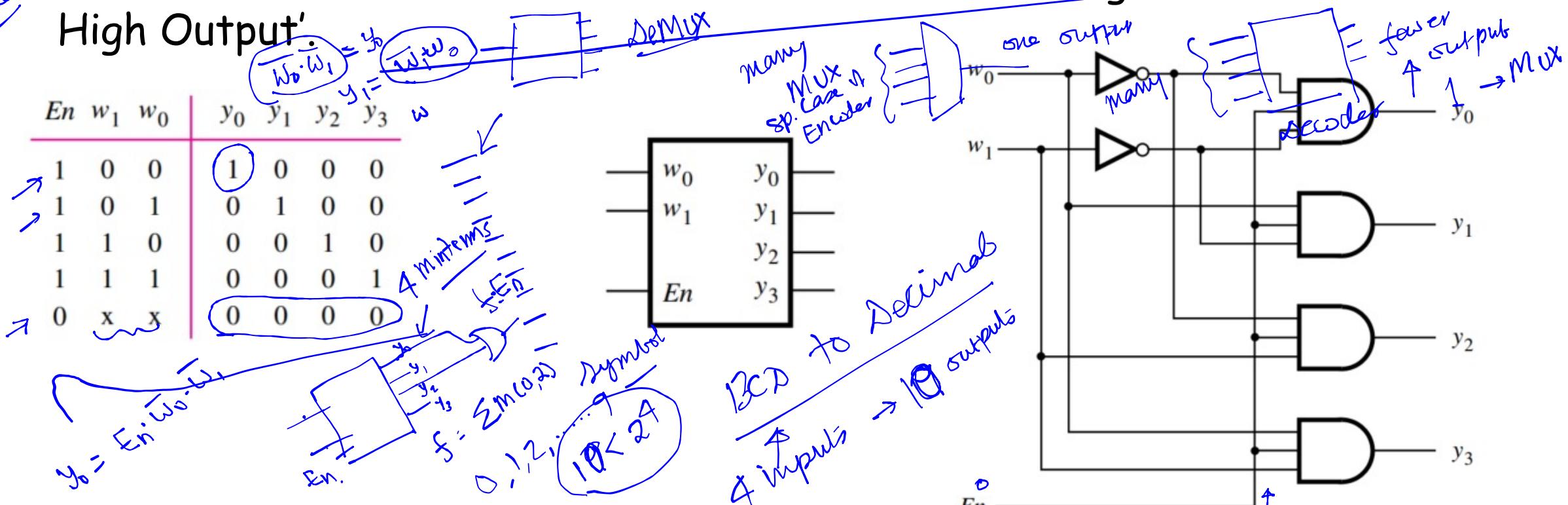
$10 \rightarrow 101111 \rightarrow 5$

$1111 \rightarrow 4$

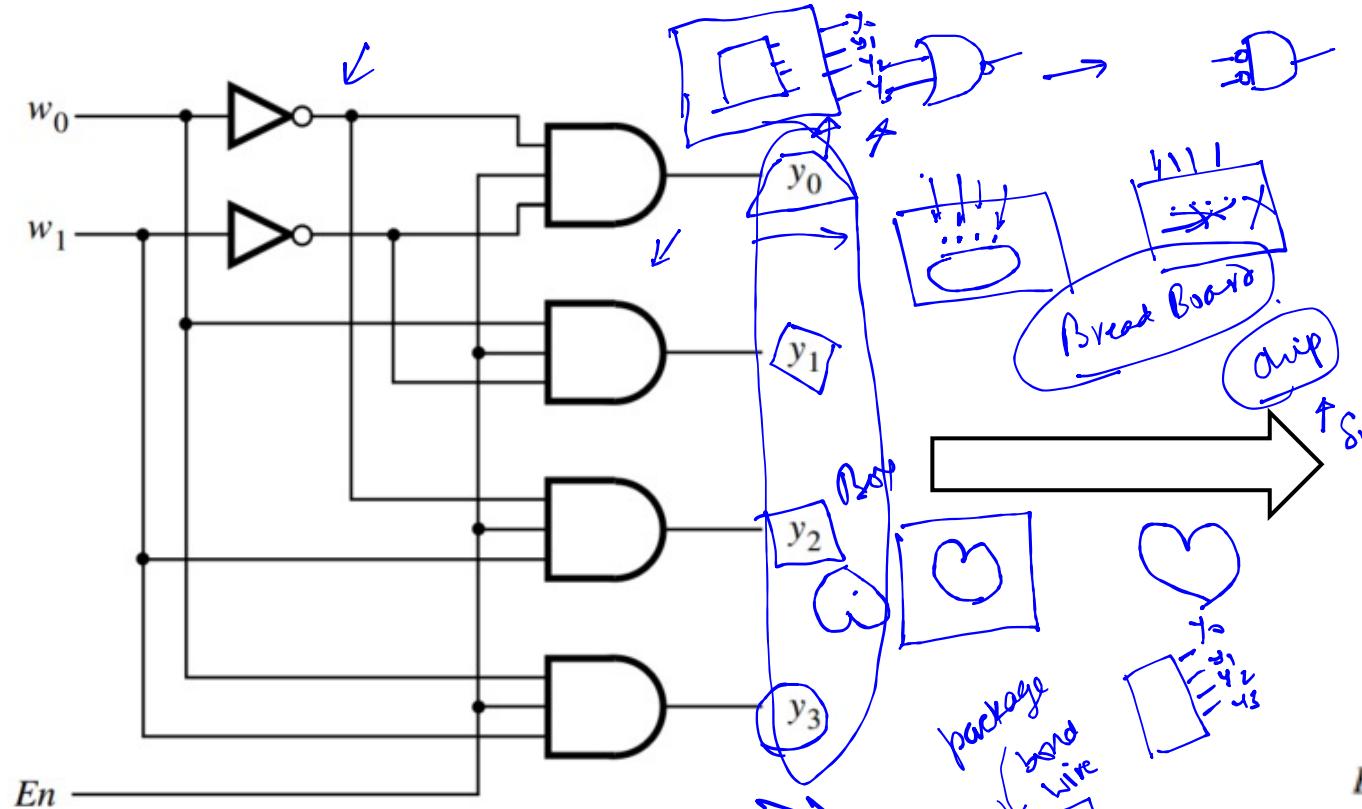
BCD to Seven segment display  
reduce the size decoder

(mp3).jpg .mpf

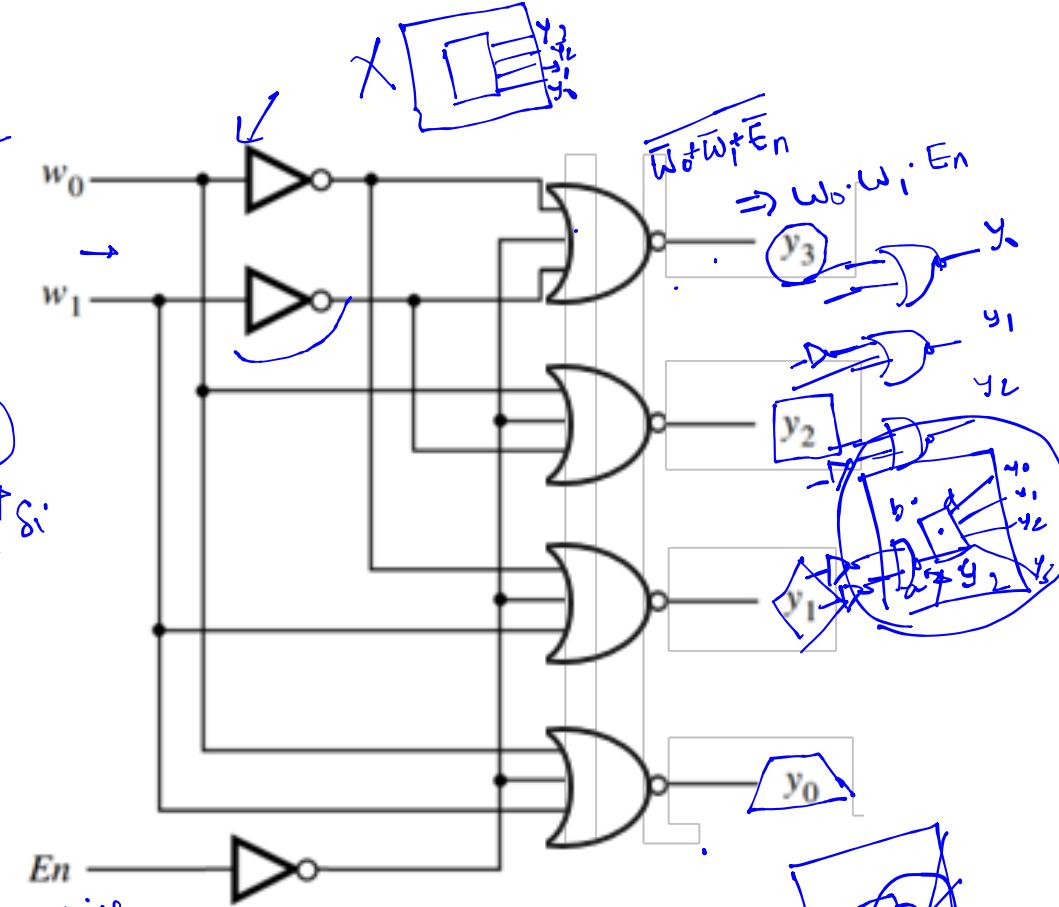
- An n-to-m-line decoder with Enable input is a combinational circuit that converts binary information from  $n$  input lines to a maximum of  $m \leq 2^n$  unique output lines only when it is enabled. When not enabled all outputs are set to logic 0.
- The decoders discussed so far are called 'Active High Enable' and 'Active High Output'.



## 2:4 Decoder with Active High Enable :

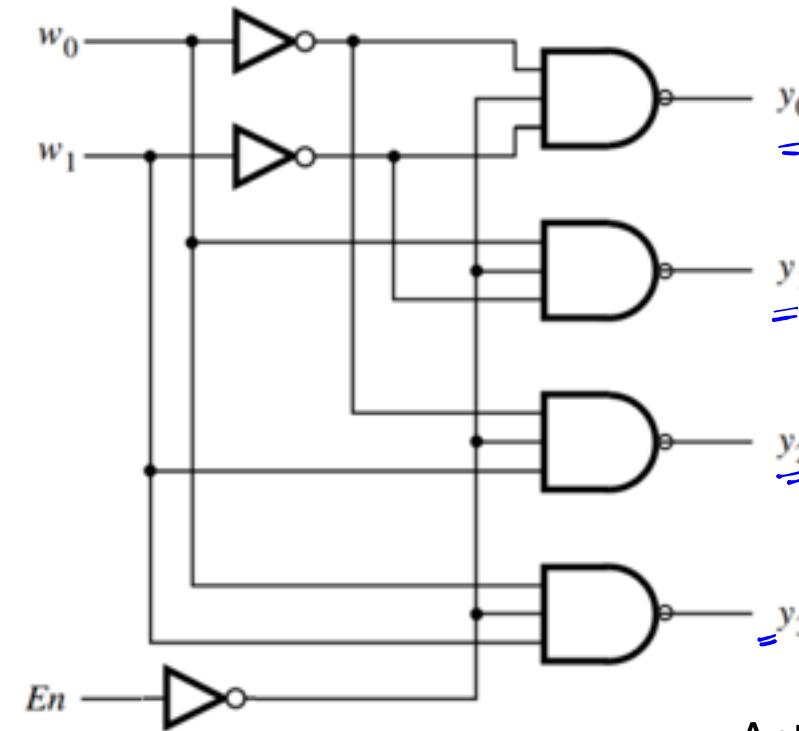
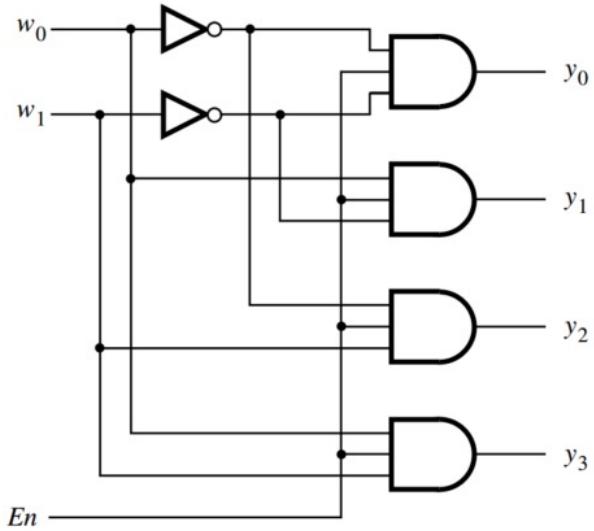


Decoder with Active High  
Enable and Active High Output  
using AND Gates



Decoder with Active High  
Enable and Active High  
Output using NOR Gates

## 2:4 Decoder with Enable Using NAND Gates:



$En$	$w_1$	$w_0$	$y_0$	$y_1$	$y_2$	$y_3$
1	0	0	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Active Low Enable

Active Low Output

