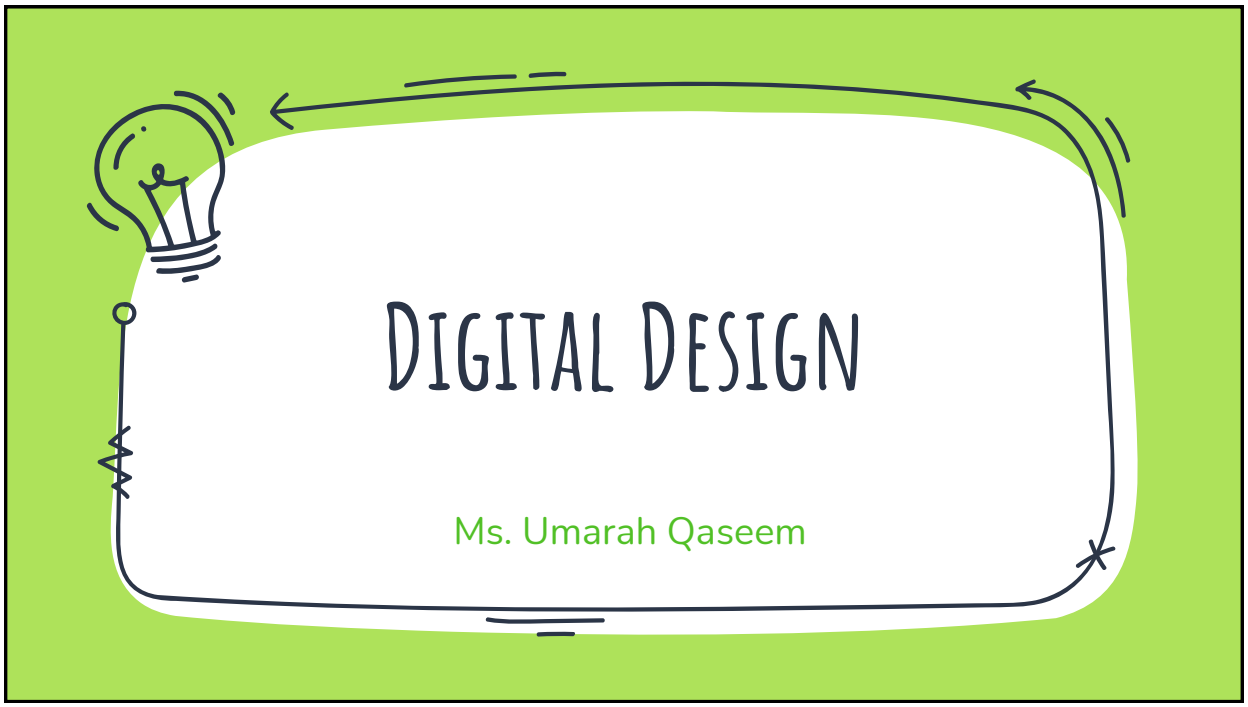
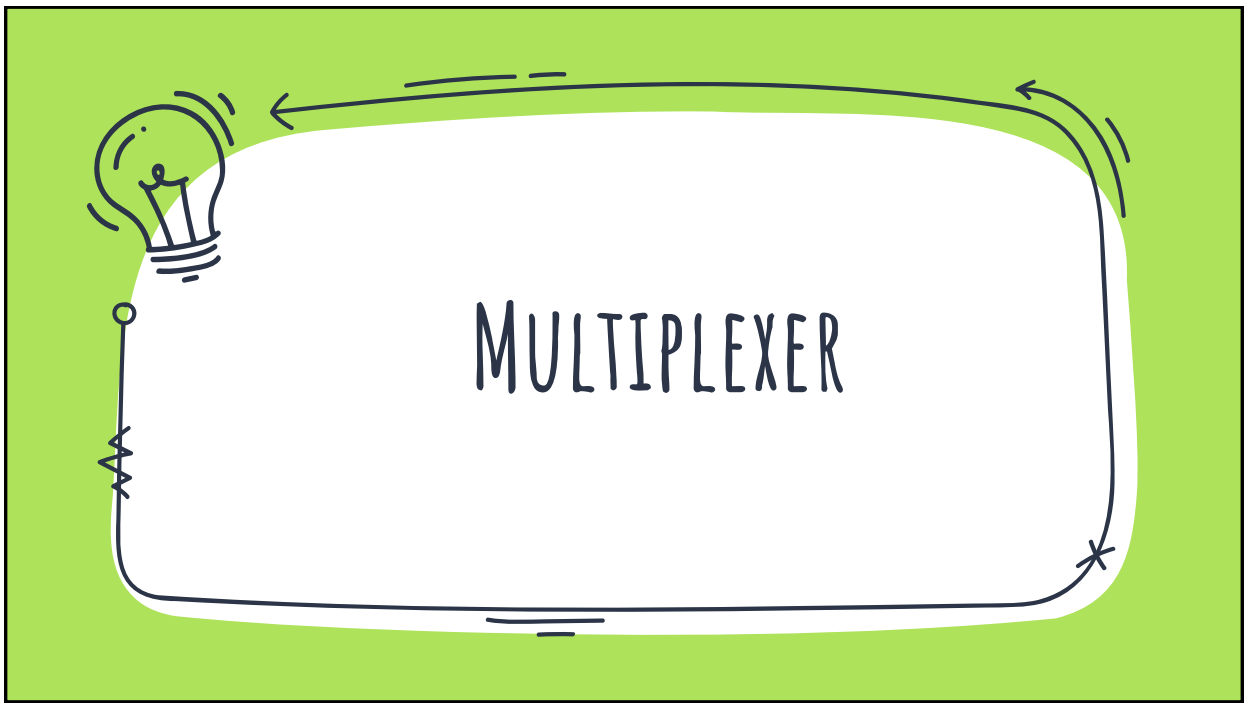



1




4

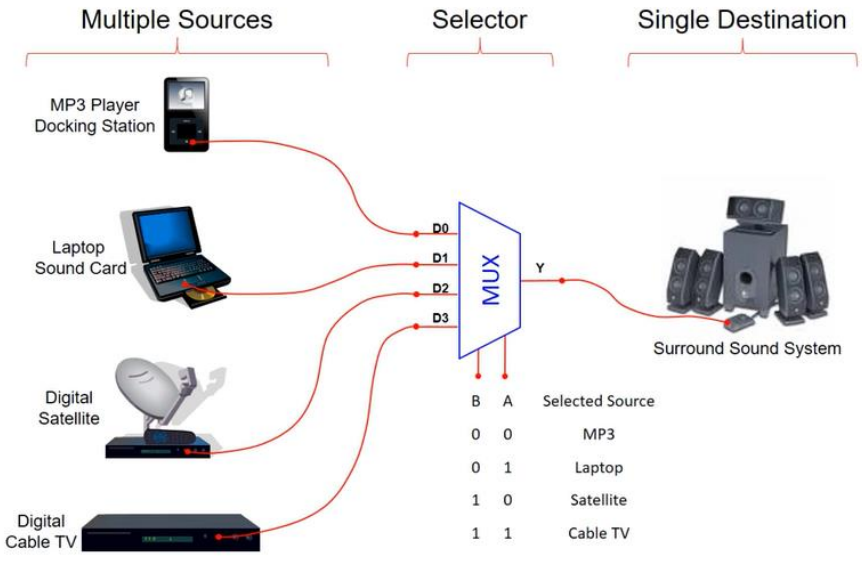
# Multiplexer

A multiplexer (MUX) selects one of multiple input signals and passes it to the output.

5

5

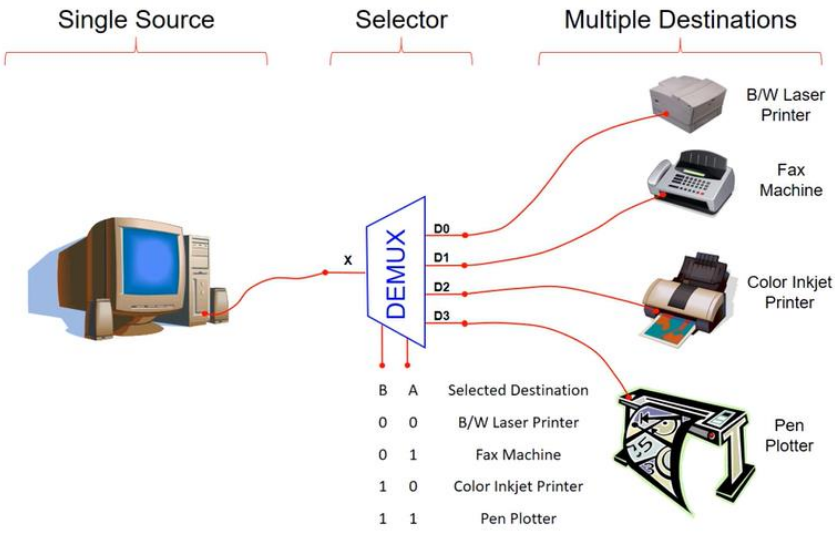
# Multiplexer




6

6

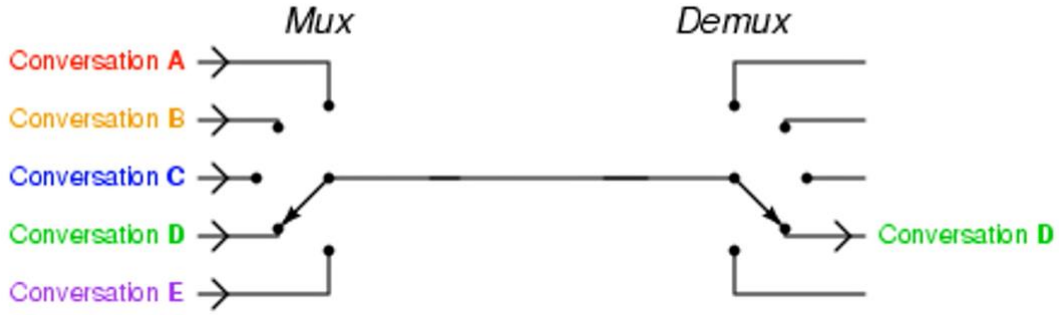
# DEMULTIPLEXER




| B | A | Selected Destination |
|---|---|---|
| 0 | 0 | B/W Laser Printer |
| 0 | 1 | Fax Machine |
| 1 | 0 | Color Inkjet Printer |
| 1 | 1 | Pen Plotter |

7

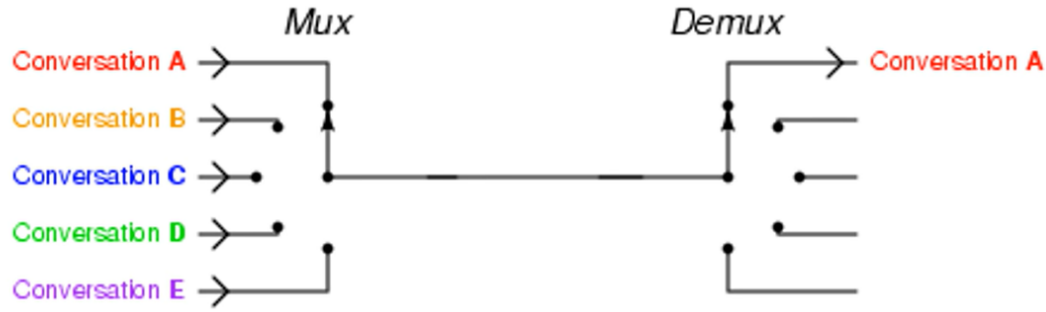
7




The basic function of a multiplexer: combining multiple inputs into a single data stream. On the receiving side, a demultiplexer splits the single data stream into the original multiple signals.

8

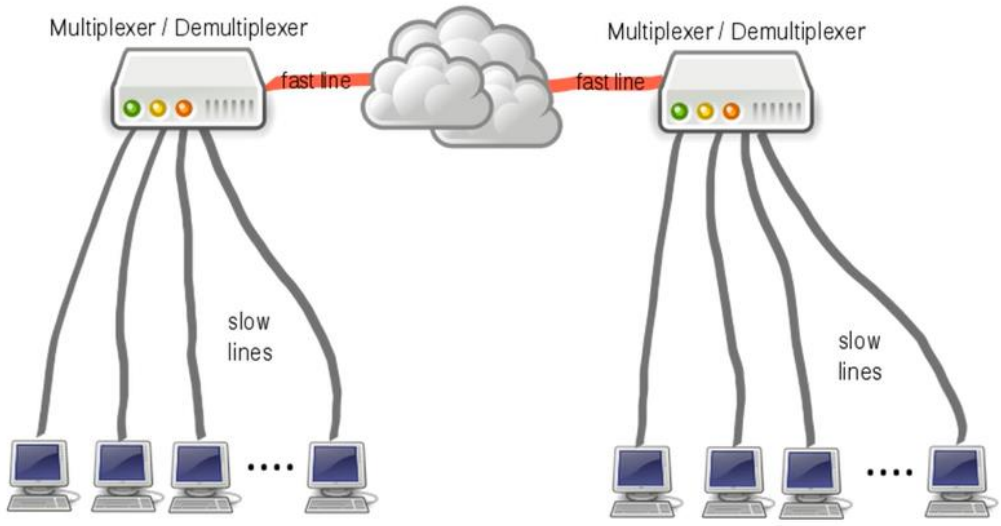
8




The basic function of a multiplexer: combining multiple inputs into a single data stream. On the receiving side, a demultiplexer splits the single data stream into the original multiple signals.

9

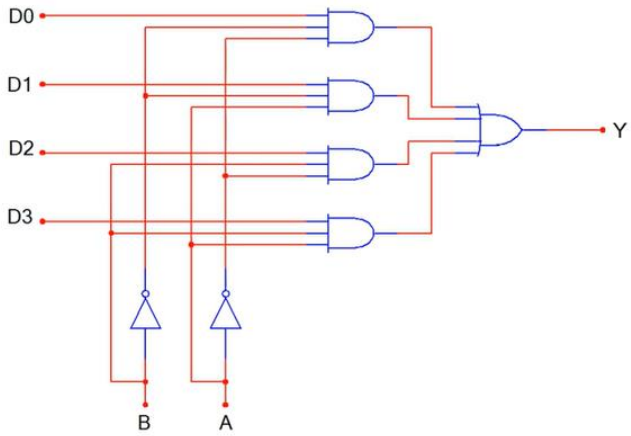
9

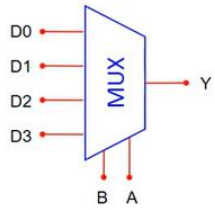



10

10

# 4-to-1 Multiplexer (MUX)



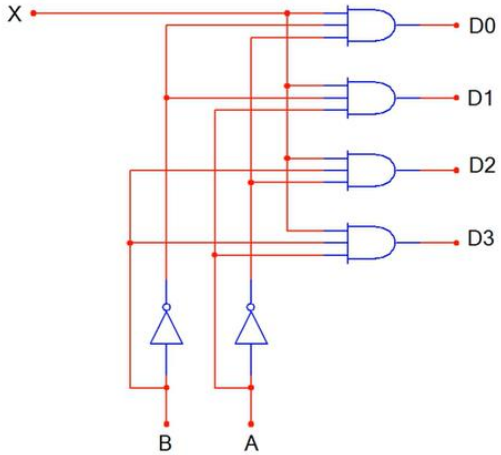



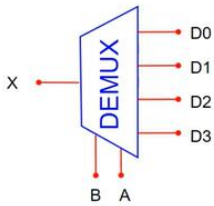

| B | A | Y |
|---|---|---|
| 0 | 0 | D0 |
| 0 | 1 | D1 |
| 1 | 0 | D2 |
| 1 | 1 | D3 |

11

11

# 1-to-4 De-Multiplexer (DEMUX)



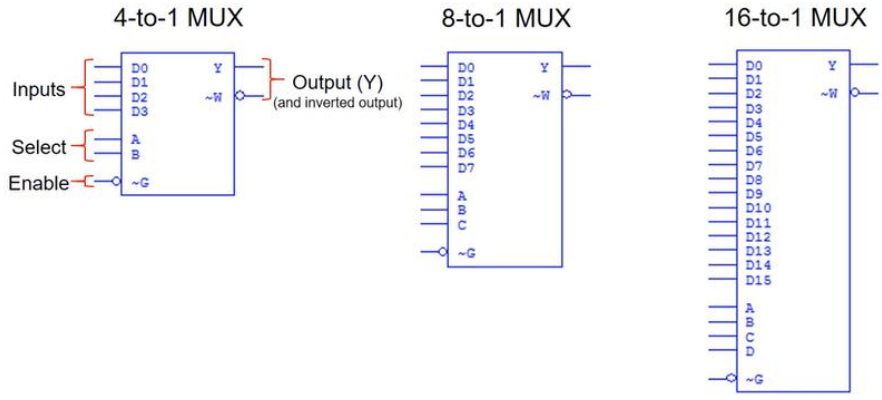




| B | A | D0 | D1 | D2 | D3 |
|---|---|---|---|---|---|
| 0 | 0 | X | 0 | 0 | 0 |
| 0 | 1 | 0 | X | 0 | 0 |
| 1 | 0 | 0 | 0 | X | 0 |
| 1 | 1 | 0 | 0 | 0 | X |

12

12

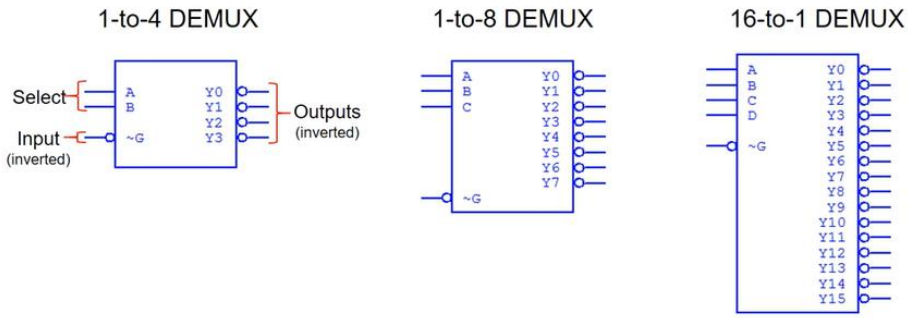
# Medium Scale Integration MUX




13

13

# Medium Scale Integration DEMUX



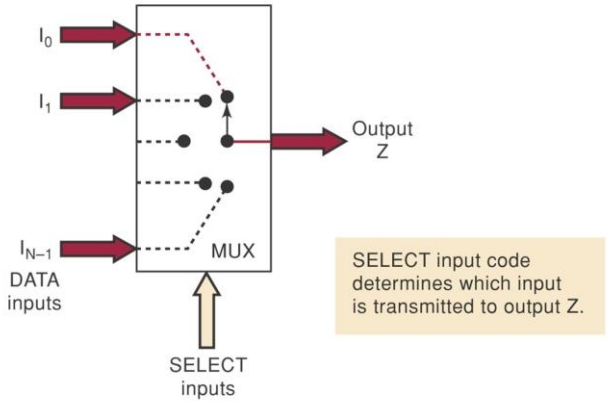

Note : Most Medium Scale Integrated (MSI) DEMUXs , like the three shown, have outputs that are inverted. This is done because it requires few logic gates to implement DEMUXs with inverted outputs rather than no-inverted outputs.

14

14

# Multiplexer (Data Selector)

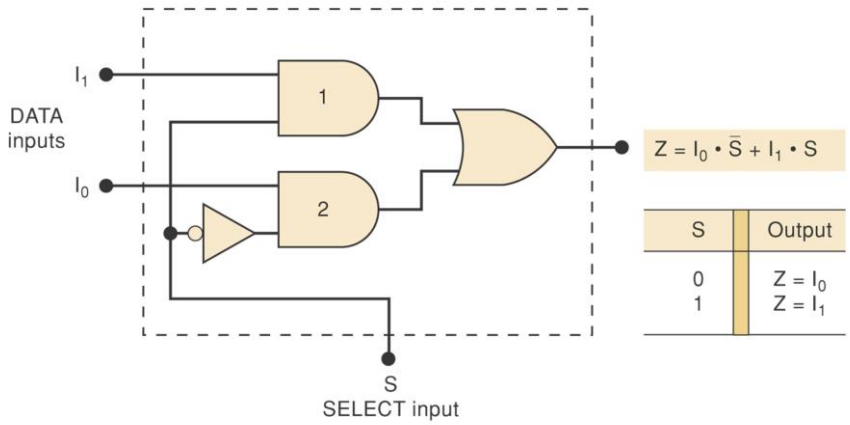
- Selects one of a set of inputs to pass on to output
- Binary control code, n lines - Choose from $2^n$ inputs




15

15

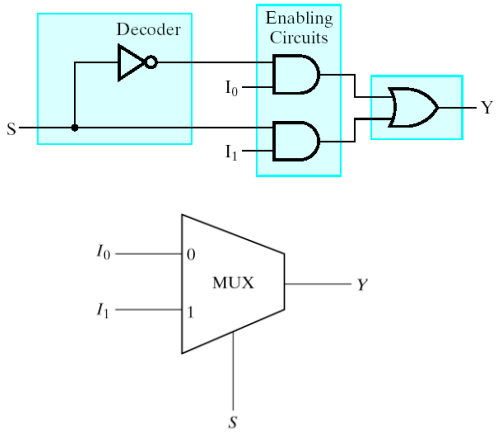
# 2-To-1 Line MUX




16

16

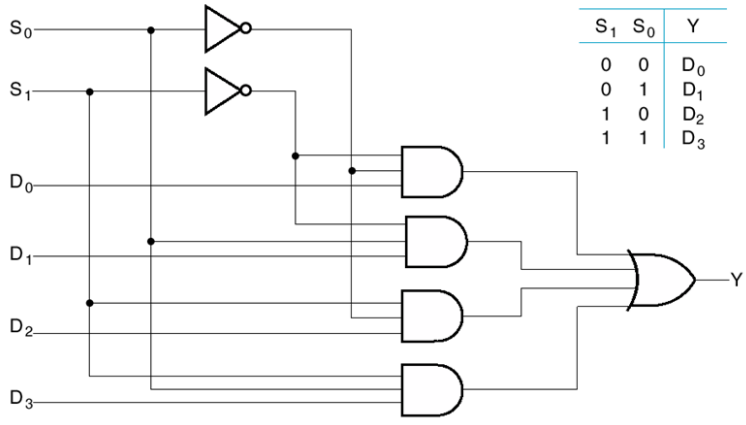
## 2-To-1 Line MUX




17

17

## 4-To-1 Line MUX



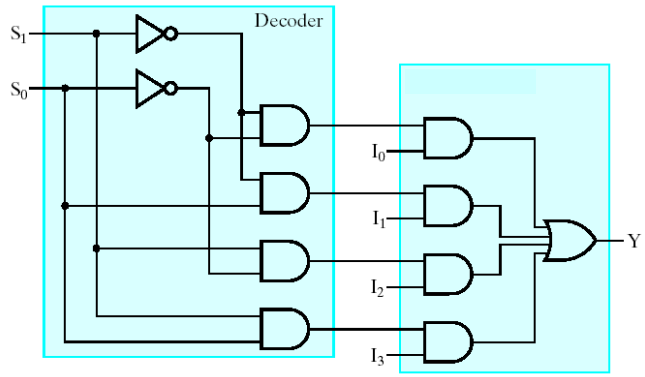

Function table

| $S_1$ | $S_0$ | Y |
|---|---|---|
| 0 | 0 | $D_0$ |
| 0 | 1 | $D_1$ |
| 1 | 0 | $D_2$ |
| 1 | 1 | $D_3$ |

18

18

# 4-to-1 line MUX



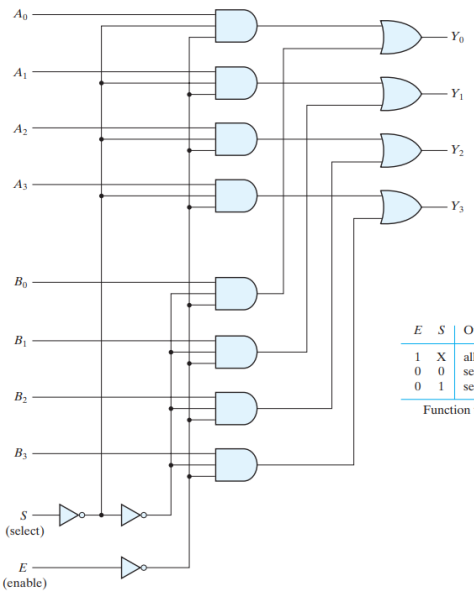

19

19

# Boolean function implementation

- Although the circuit contains four 2-to-1-line multiplexers, we are more likely to view it as a circuit that selects one of two 4-bit sets of data lines.
- As shown in the function table, the unit is enabled when E = 0.
- 
- Then, if S = 0, the four A inputs have a path to the four outputs.
- If, by contrast, S = 1, the four B inputs are applied to the outputs.
- The outputs have all 0′ s when E = 1, regardless of the value of S .

160 Chapter 4 Combinational Logic




| E | S | Output Y |
|---|---|---|
| 1 | X | all 0's |
| 0 | 0 | select A |
| 0 | 1 | select B |

Function table

**FIGURE 4.26**
Quadruple two-to-one-line multiplexer

20

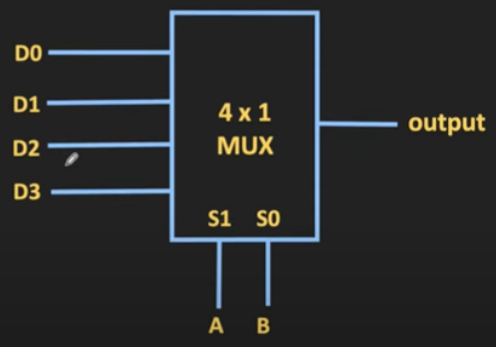
# Implementation of any function through multiplexers

21

## Implementation of any function through Multiplexer

**Truth Table** $F(A, B, C) = \sum m(0, 2, 3, 7)$

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |



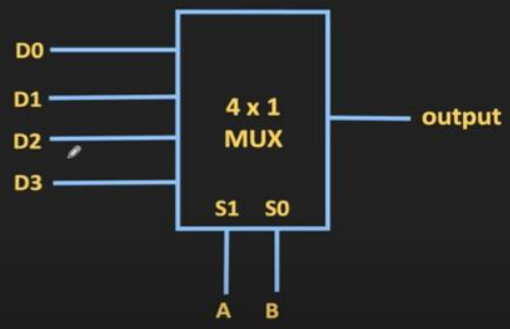

22

# Example 1:

$F(A, B, C) = \sum m(0, 2, 3, 7)$

**Truth Table**

| A | B | C | F | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 1 | 1 | |



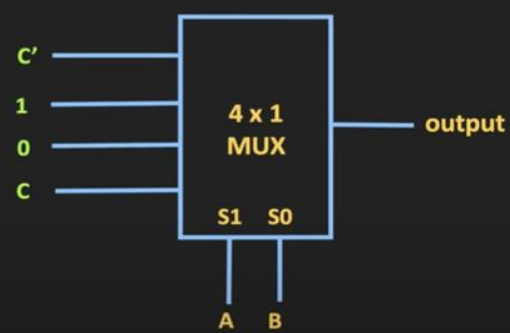

23

# Example 1:

$F(A, B, C) = \sum m(0, 2, 3, 7)$

**Truth Table**

| A | B | C | F | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | F = C' |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | F = 1 |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | F = 0 |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 0 | F = C |
| 1 | 1 | 1 | 1 | |

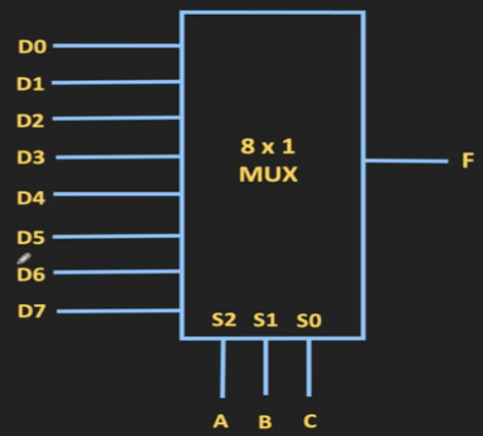



24

## Example 2:

| A | B | C | D | F | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | |
| 1 | 0 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 1 | 1 | |

$F(A, B, C, D) = \sum m(1, 3, 6, 9, 12, 14, 15)$

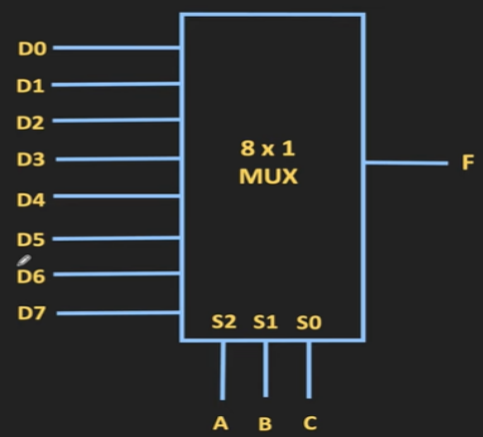



25

## Example 2:

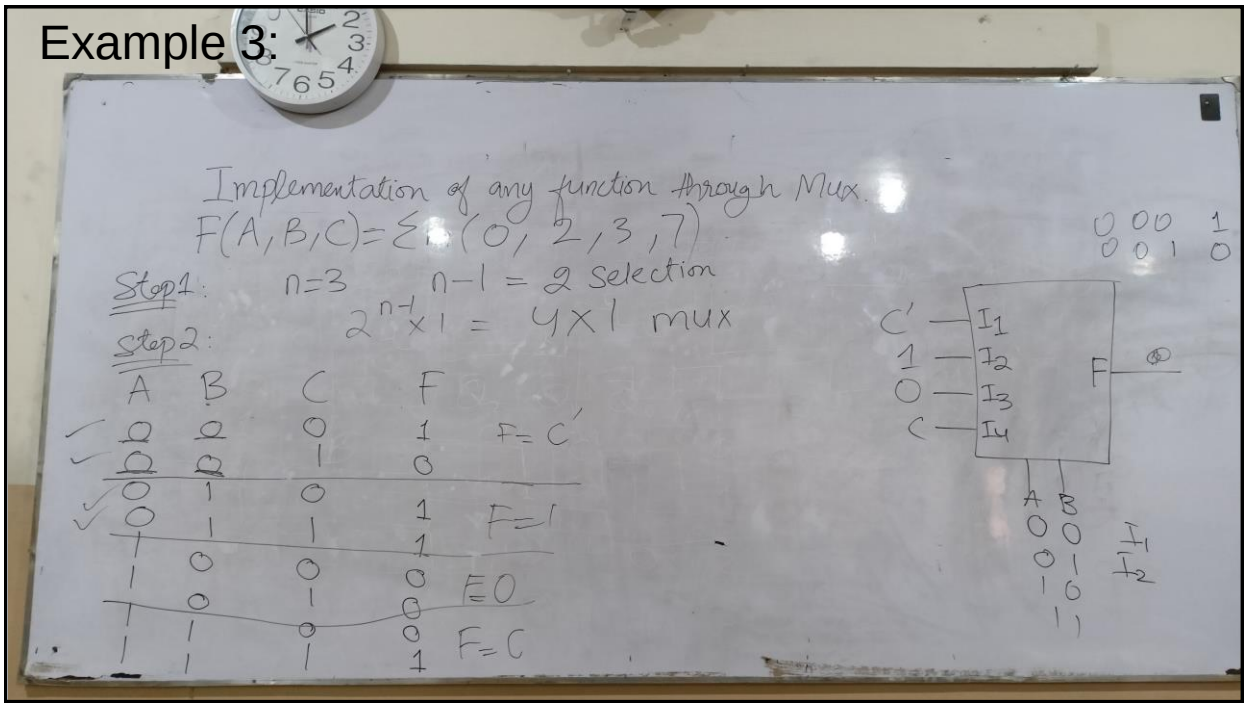
| A | B | C | D | F | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | F = D |
| 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 0 | F = D |
| 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 0 | 0 | 0 | F = 0 |
| 0 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 1 | F = D' |
| 0 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | F = D |
| 1 | 0 | 0 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | F = 0 |
| 1 | 0 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 0 | 1 | F = D' |
| 1 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 1 | 0 | 1 | F = 1 |
| 1 | 1 | 1 | 1 | 1 | |

$F(A, B, C, D) = \sum m(1, 3, 6, 9, 12, 14, 15)$



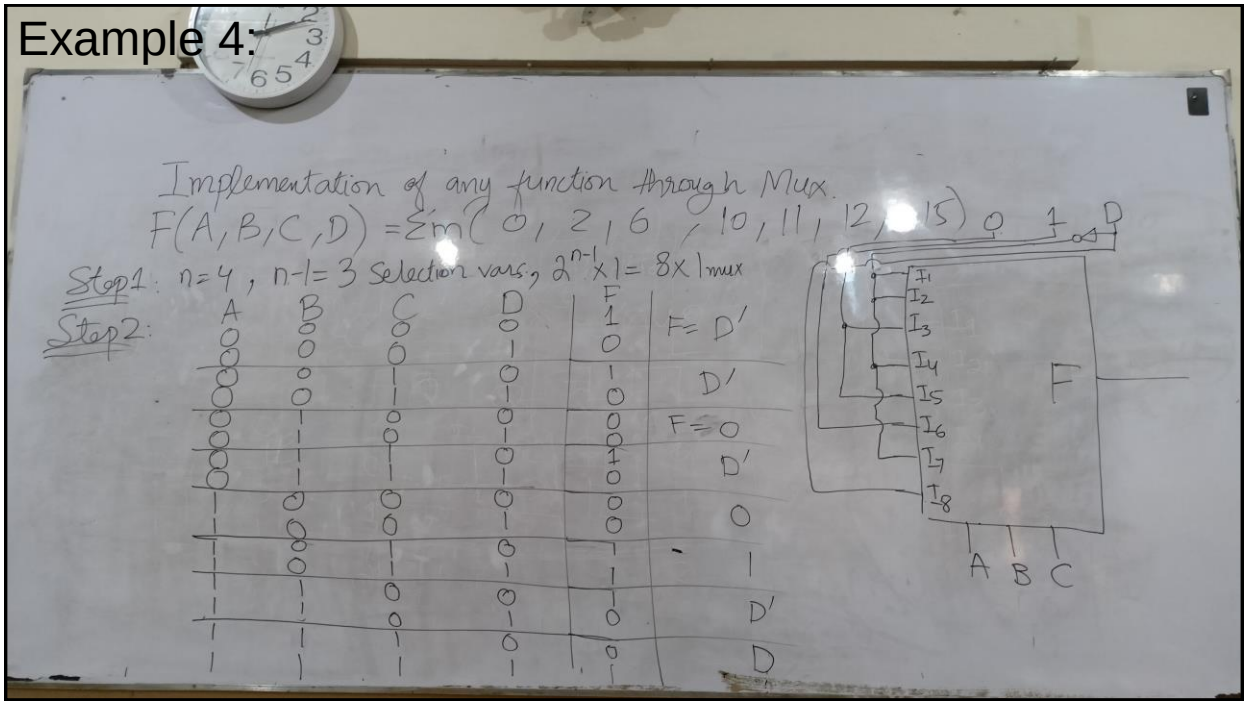

26

## Example 3:




27

## Example 4:

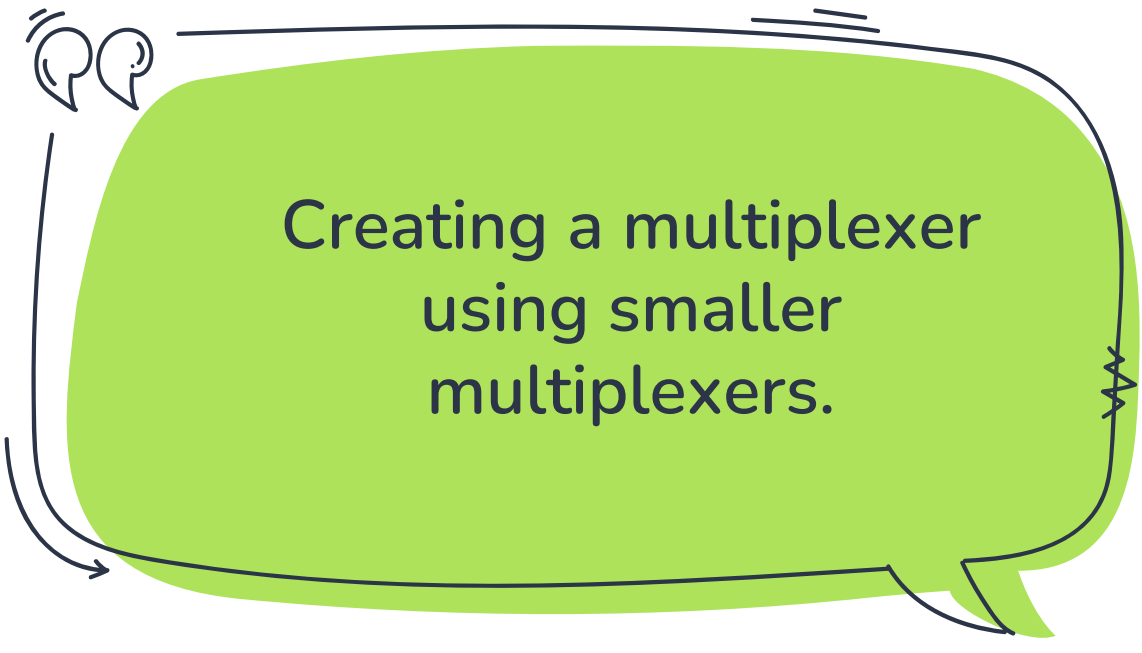



28

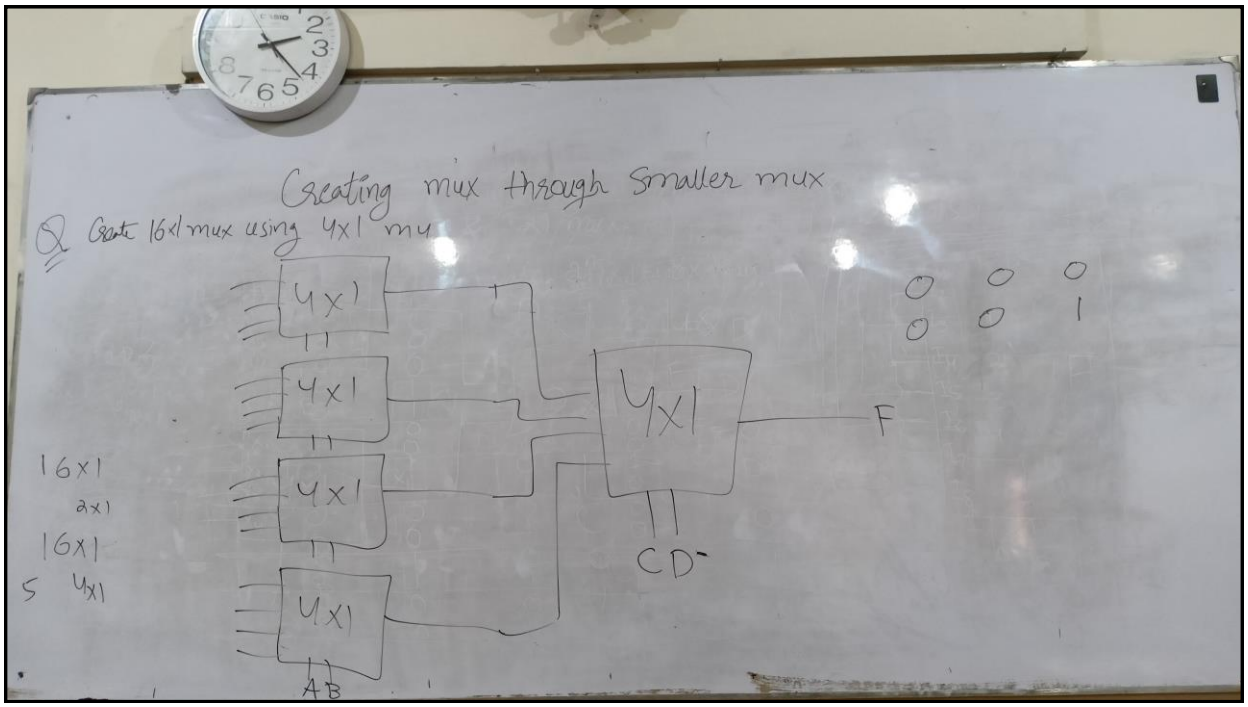
## Implementation of any function through Multiplexer
## Activity: Solve it yourself

| A | B | C | F | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | |

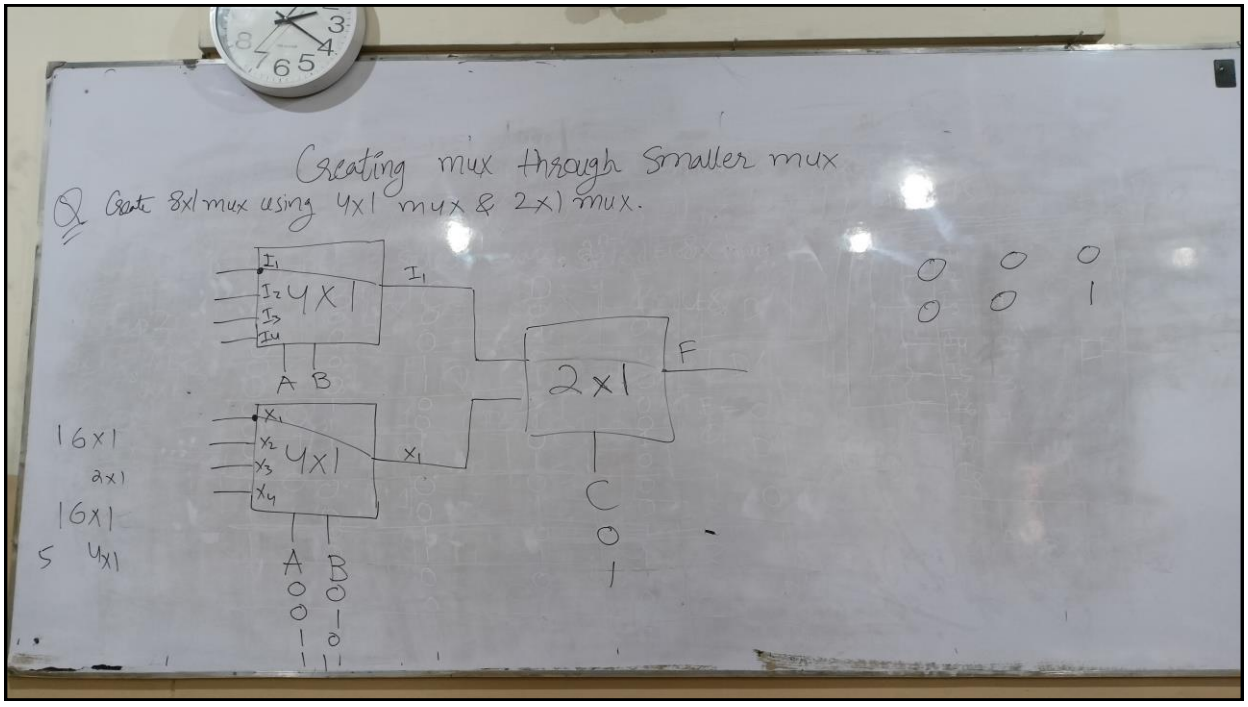
29



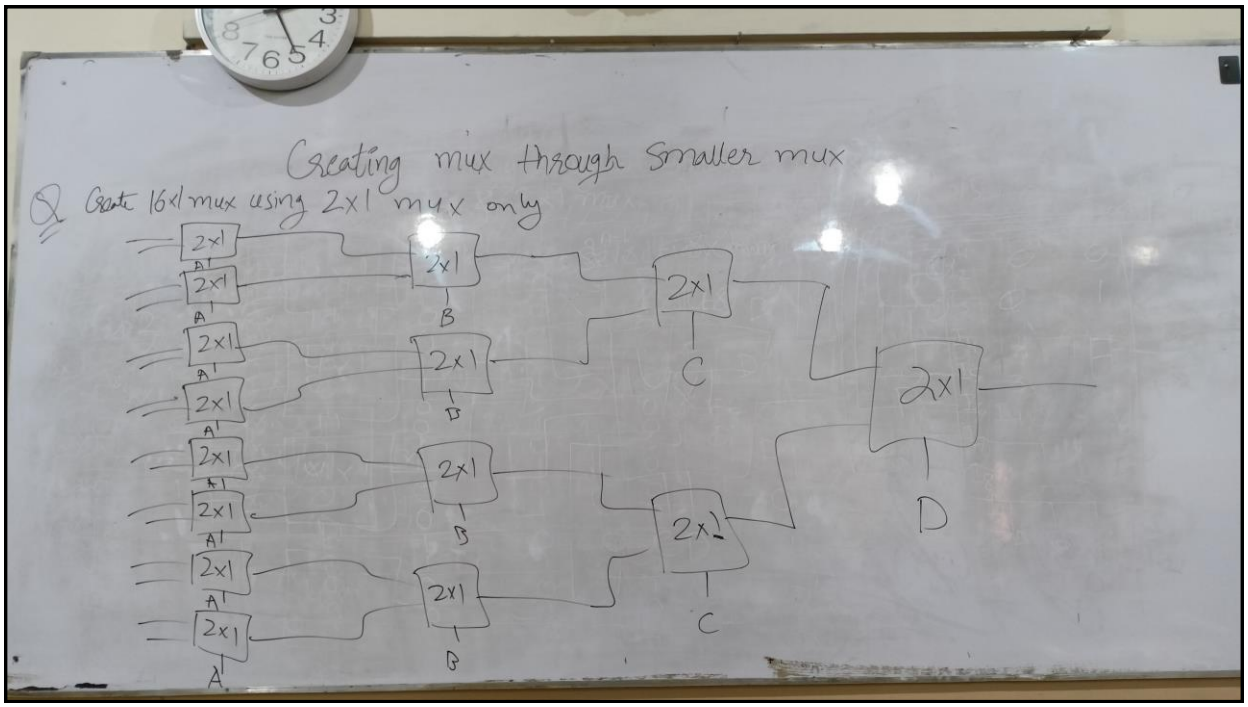

30


Creating mux through Smaller mux
Q Create 16x1 mux using 4x1 mux
4x1
4x1
4x1
4x1
4x1
F
CD
AB
16x1
2x1
16x1
5 4x1
0 0 0
0 0 1

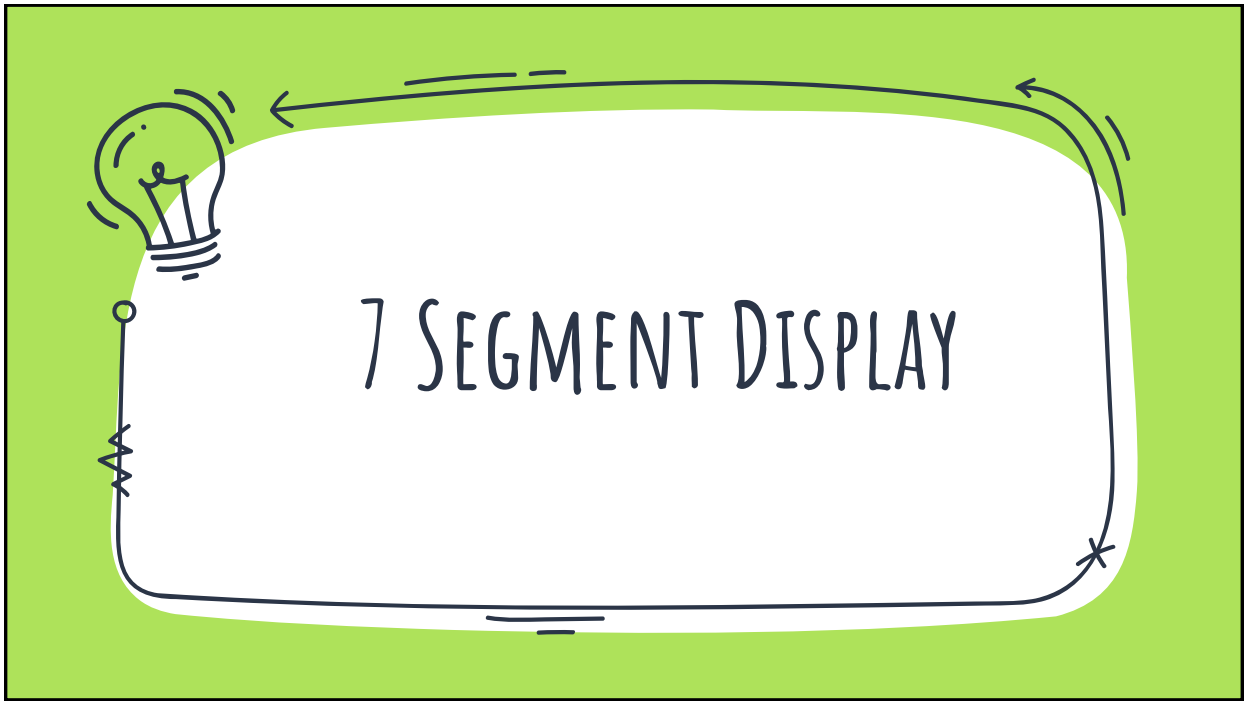
31


Creating mux through Smaller mux
Q Create 8x1 mux using 4x1 mux & 2x1 mux.
I1
I2
I3
I4
4X1
A B
I1
2x1
F
C
0
1
X1
X2
X3
X4
4X1
X1
A 0 0 1 1
B 0 1 0 1
16x1
2x1
16x1
5 4x1
0 0 0
0 0 1

32

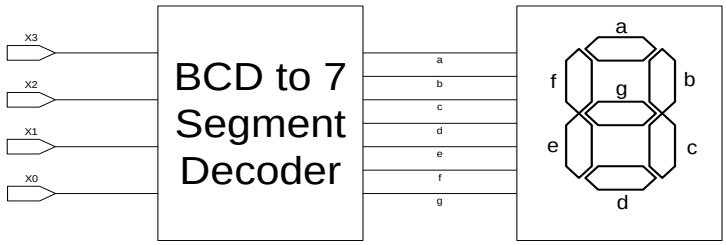

Creating mux through Smaller mux
Q Create 16x1 mux using 2x1 mux only
2x1
A
B
C
D

33


7 Segment Display

37

# BCD to 7 Segment Display

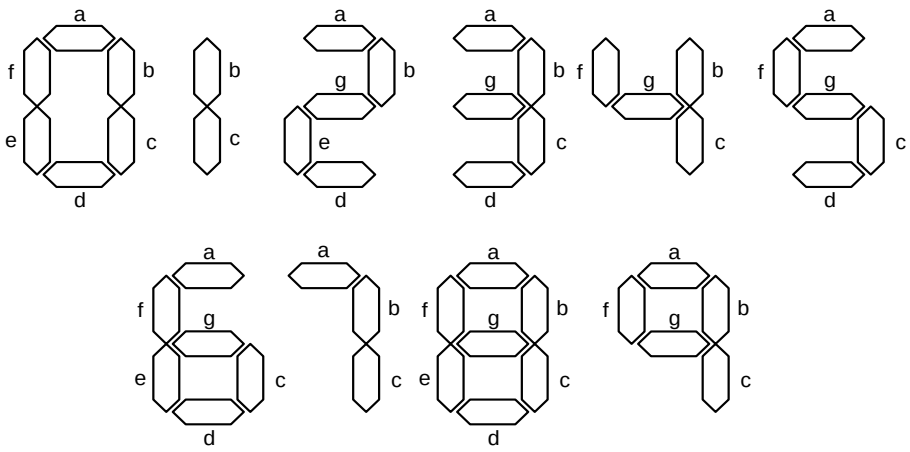



Design the logic circuitry that will drive a seven segment LED display and will be able to represent numbers from 0 to 9

38

38

# Possible numbers and their representation on 7 segment display




39

39

# Truth Table

| X3 | X2 | X1 | X0 | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | x | x | x | x | x | x | x |
| 1 | 0 | 1 | 1 | x | x | x | x | x | x | x |
| 1 | 1 | 0 | 0 | x | x | x | x | x | x | x |
| 1 | 1 | 0 | 1 | x | x | x | x | x | x | x |
| 1 | 1 | 1 | 0 | x | x | x | x | x | x | x |
| 1 | 1 | 1 | 1 | x | x | x | x | x | x | x |

40

40

# References


- Chapter 4 – Digital Design Morris Mano
- Logic and Computer Design Fundamentals, 4e, Power Point Slides, 2008 Pearson Education Inc.
- Digital Design – Amirali Baniasad
- Template is taken from slides carnival.


Slides Carnival

44