

CSE 140 Lecture 11

Standard Combinational Modules

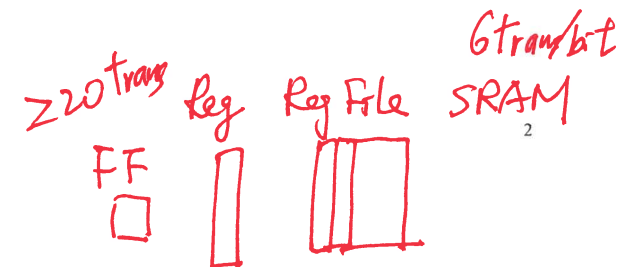
Uniform
Widely used
Recognized

CK Cheng
CSE Dept.
UC San Diego

Part III - Standard Combinational Modules

- Introduction
- Decoder
 - Behavior, Logic, Usage
- Encoder
- Multiplexer (Mux)
 - Behavior, Logic, Usage
- Demultiplexer (DeMux)

Universal Set?



Part III - Standard Combinational Modules

Signal Transport

- Decoder: Decode address
- Encoder: Encode address
- Multiplexer (Mux): Select data by address
- Demultiplexer (DeMux): Direct data by address
- Shifter: Shift bit location

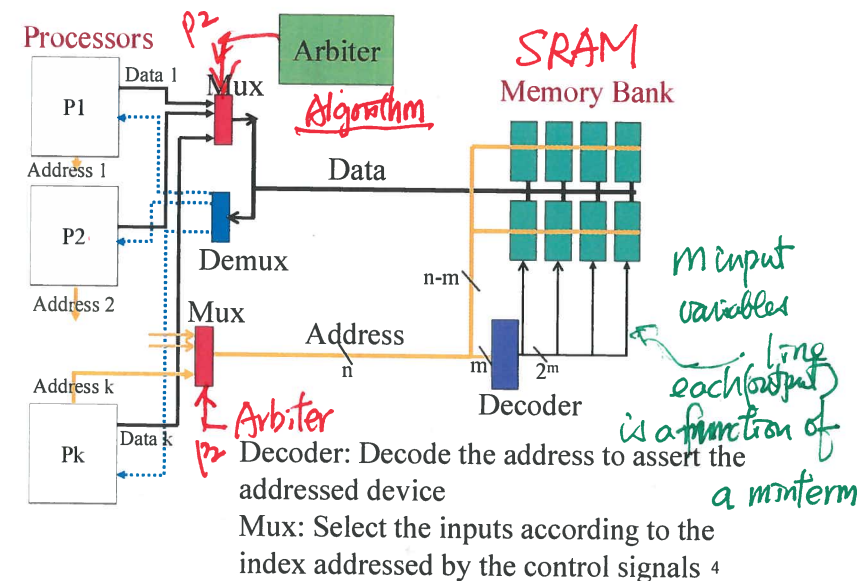
Data Operator

- Adder: Add two binary numbers
- Multiplier: Multiply two binary numbers

Address, Data

Enable Device ← Address
Device → " " Address: Device → Databus

Interconnect: Decoder, Encoder, Mux, DeMux

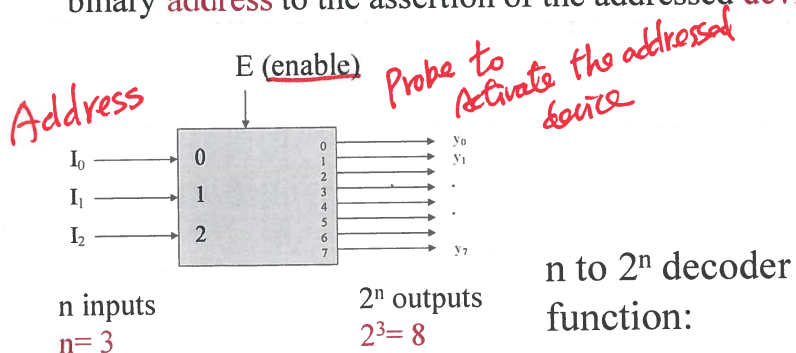


1. Decoder

- Definition
- Logic Diagram
- Application (Universal Set)
- Tree of Decoders

5

Decoder Definition: A digital module that converts a binary **address** to the assertion of the addressed **device**



$$\begin{cases} y_i = 1 \text{ if } E = 1 \text{ \& } (I_2, I_1, I_0) = i \\ y_i = 0 \text{ otherwise} \end{cases}$$

$E = \begin{cases} 0: \text{deactivate the decoder} \Rightarrow y_i = 0 \forall i \\ 1: \text{one \& only one output is true} \end{cases}$

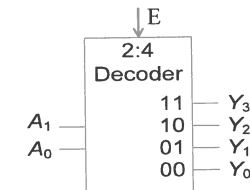
Decoder: Definition

- A device that decodes
- An electronic device that converts signals from one form to another
- A machine that converts a coded text into ordinary language
- A device or program that translates encoded data into its original format
- All of the above

6

1. Decoder: Definition

- N inputs, 2^N outputs
- One-hot outputs: only one output HIGH at most



$E = 1$

i	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1	0
1	0	1	0	0	0	1
2	1	0	0	1	0	0
3	1	1	1	0	0	0

only one of the output is true.

minterm $\rightarrow$ only one entry is true for each output.

8

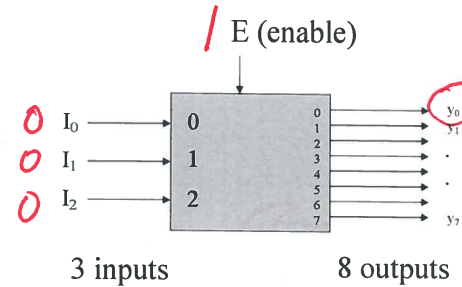
1. Decoder: Definition

Example: A 3-input decoder has how many outputs?

- A. 2 outputs
- B. 4 outputs
- C. 8 outputs
- D. 10 outputs

↓^E

Decoder Definition



Example: For a 3-input decoder, suppose $(E, I_2, I_1, I_0) = (1, 0, 0, 0)$, then $(y_7, y_6, \dots, y_0)$ is equal to:

- A. (00000000)
- B. (00000001)**
- C. (00000010)
- D. (01000000)
- E. (10000000)

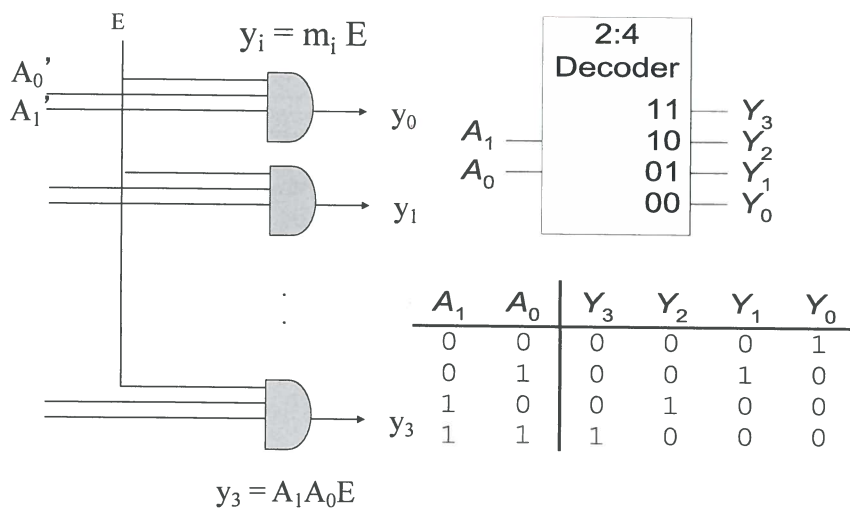
Handwritten red text: y7, ..., y0

9

10

Decoder: Logic Diagram (Inside a decoder)

$y_0 = 1$ if $(A_1, A_0) = (0, 0)$ & $E = 1$

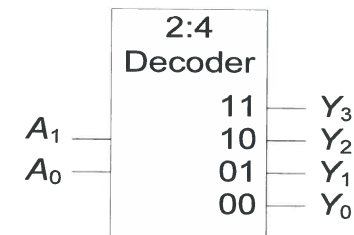


11

1. Decoder: Definition

PI Q: What is the output $Y_{3:0}$ of the 2:4 decoder for $(A_1, A_0) = (1, 0)$?

- A. (1, 1, 0, 0)
- B. (1, 0, 1, 1)
- C. (0, 0, 1, 0)
- D. (0, 1, 0, 0)



12

Decoder Application: universal set {Decoder, OR}

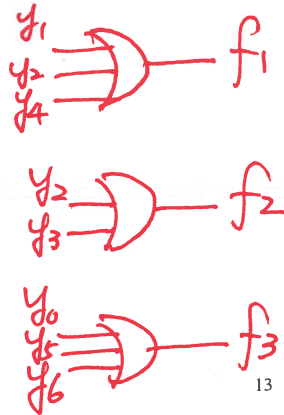
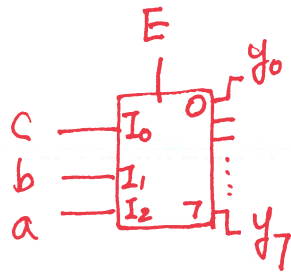
Example:

Implement the following functions with a 3-input decoder and OR gates.

i) $f_1(a,b,c) = \Sigma m(1,2,4)$

ii) $f_2(a,b,c) = \Sigma m(2,3)$

iii) $f_3(a,b,c) = \Sigma m(0,5,6)$



Decoder Application: universal set {Decoder, OR}

Decoder produces minterms when $E=1$.

We can use an OR gate to collect the minterms to cover the On-set.

For the Don't Care-Set, we can just ignore the terms.

14

Decoder Application: universal set {Decoder, OR}

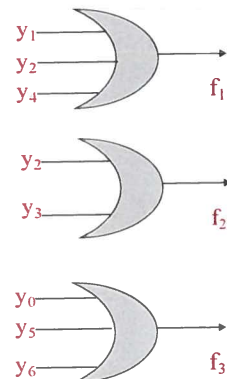
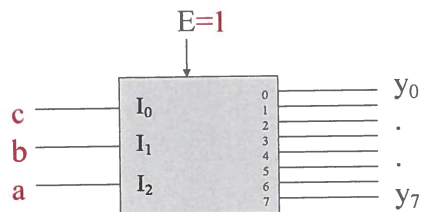
Example: Implement functions

i) $f_1(a,b,c) = \Sigma m(1,2,4) + \Sigma d(0,5)$,

ii) $f_2(a,b,c) = \Sigma m(2,3) + \Sigma d(1,4)$,

iii) $f_3(a,b,c) = \Sigma m(0,5,6)$

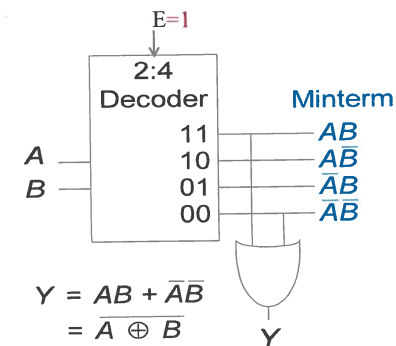
with a 3-input decoder and OR gates.



15

Decoders

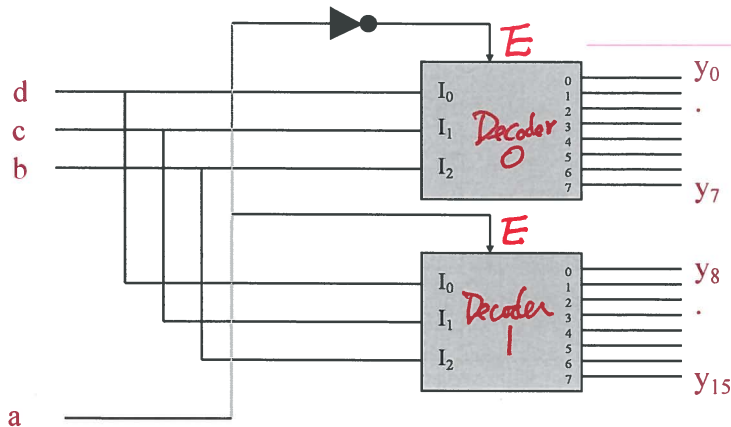
- OR minterms



16

Tree of Decoders: Scale up the size of the decoders using a tree structure

Implement a 4-2⁴ decoder with 3-2³ decoders.



Voting machine



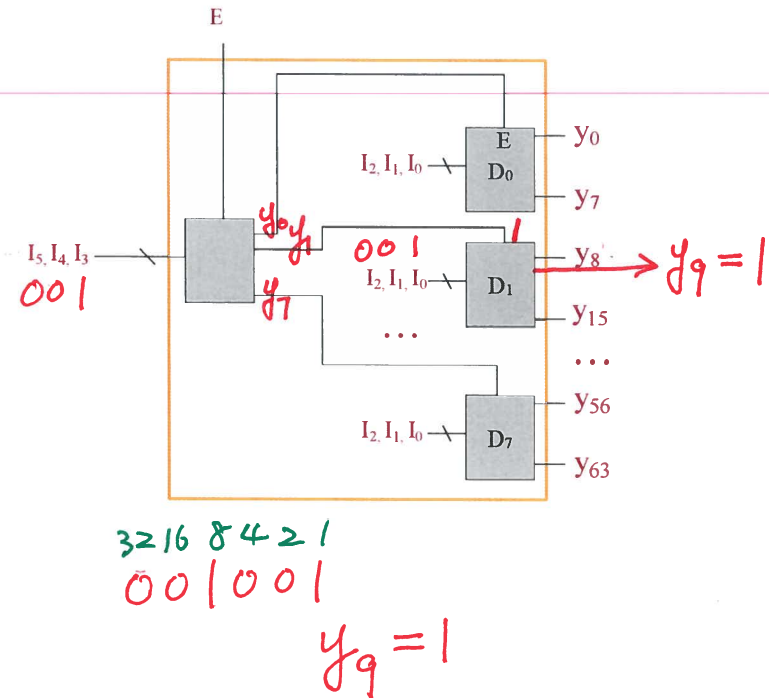
true or false
true if # inputs which are true ≥ 1
bit counter

Construction: A four variable switching function $f(a,b,c,d)$ can be implemented using which of the following?

- A. 1:2 decoders and OR gates
- B. 2:4 decoders and OR gates
- C. 3:8 decoders and OR gates
- D. All of the above
- E. None of the above

Tree of Decoders

Implement a 6-2⁶ decoder with 3-2³ decoders.



2. Encoder

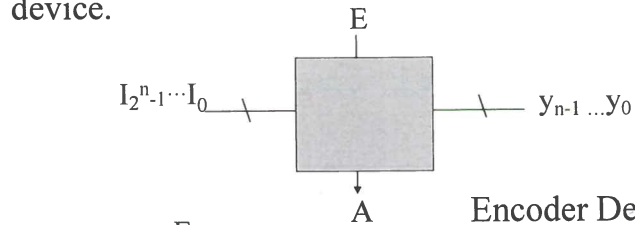
- Definition
- Logic Diagram
- Priority Encoder

Definition of Encoder

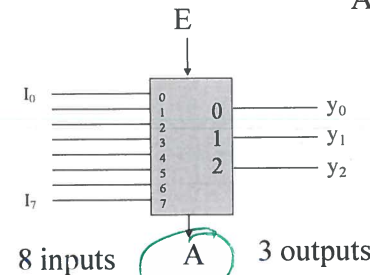
- A. Any program, circuit or algorithm which encodes
- B. In digital audio technology, an encoder is a program that converts an audio WAV file into an MP3 file
- C. A device that convert a message from plain text into code
- D. A circuit that is used to convert between digital video and analog video
- E. All of the above

21

Encoder Definition: A digital module that converts the assertion of a **device** to the binary **address** of the device.



Encoder Description:



At most one $I_i = 1$.

$(y_{n-1}, \dots, y_0) = i$ if $I_i = 1$ & $E = 1$

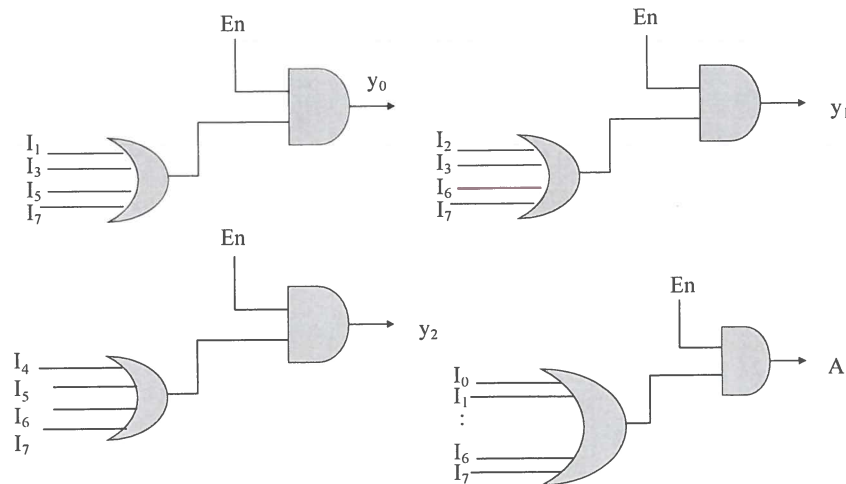
$(y_{n-1}, \dots, y_0) = 0$ otherwise.

$A = 1$ if $E = 1$ and one i s.t. $I_i = 1$

$A = 0$ otherwise.

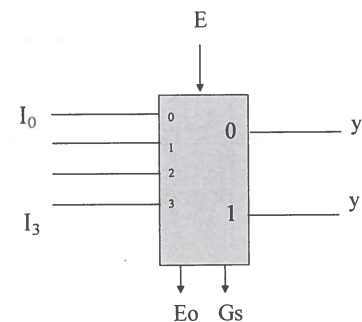
22

Encoder: Logic Diagram



23

Priority Encoder:

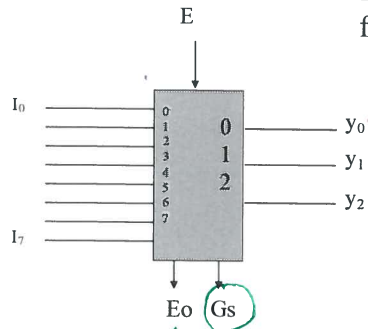


24

Priority Encoder: Definition

Description: Input ($I_{2^n-1}, \dots, I_0$), Output ($y_{n-1}, \dots, y_0$)

$(y_{n-1}, \dots, y_0) = i$ if $I_i = 1$ & $E = 1$ & $I_k = 0$
for all $k > i$ (high bit priority) or
for all $k < i$ (low bit priority).



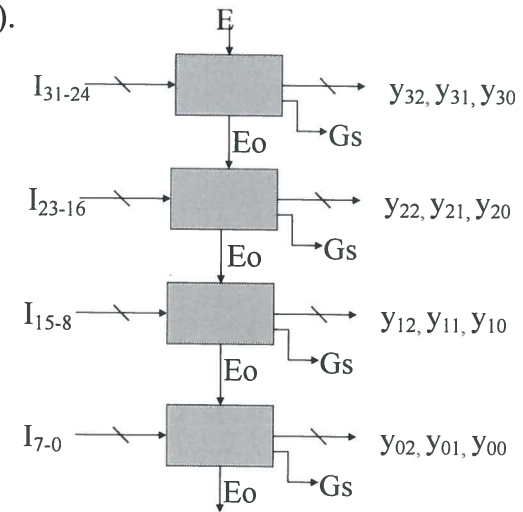
$E_o = 1$ if $E = 1$ & $I_i = 0$ for all i ,
 $G_s = 1$ if $E = 1$ & $\exists i$ s.t. $I_i = 1$.

(G_s is like A_s and E_o passes on enable).

$\downarrow E=1 \text{ but } I_i=0 \forall i$

25

Priority Encoder: Implement a 32-input priority encoder with 8 input priority encoders (high bit priority).



26

CSE 140 Lecture 12

Standard Combinational Modules

CK Cheng
CSE Dept.
UC San Diego

1

Part III. Standard Modules

Interconnect Modules:

1. Decoder, 2. Encoder
- 3. Multiplexer, 4. Demultiplexer**

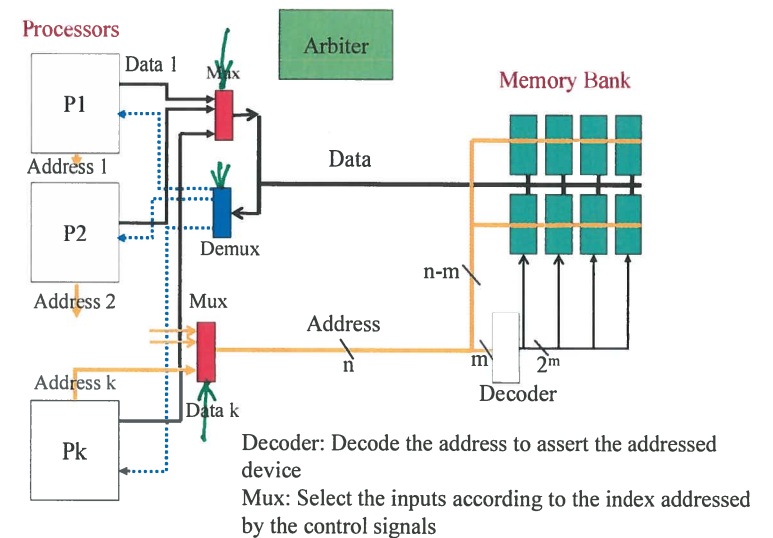
2

Multiplexer

- Definition
- Logic Diagram
- Application

3

Interconnect: Decoder, Encoder, Mux, DeMux



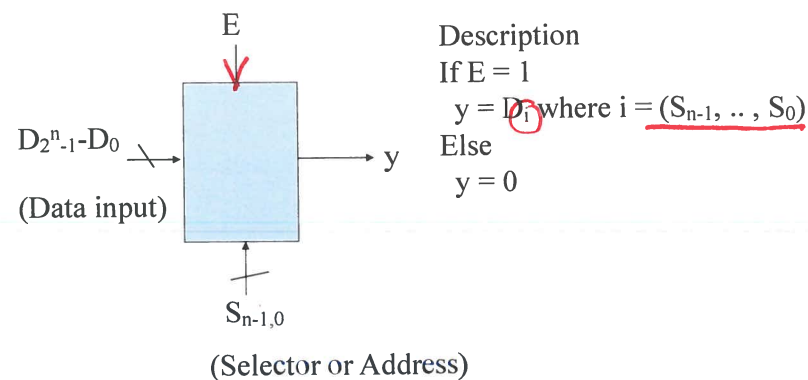
4

Multiplexer Definition

- A. A device that interleaves two or more activities
- B. A communications device that combines several signals for transmission over a single medium
- C. A logic circuit that sends one of several inputs out over a single output channel.**
- D. The circuit that uses a common communications channel for sending two or more messages or signals.
- E. All of the above

5

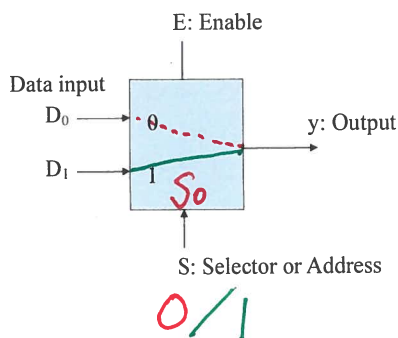
3. Mux (Multiplexer) Definition: A digital module that selects one of data inputs according to the binary address of the selector.



6

Multiplexer (Mux): Definition

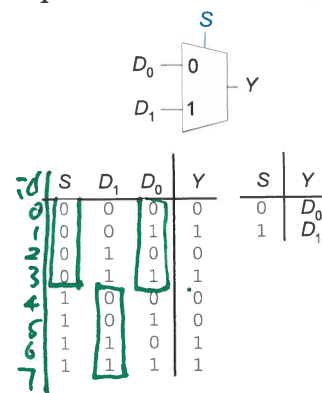
- Selects between one of N inputs to connect to the output.
- $\log_2 N$ -bit select input – control input



7

Multiplexer (Mux): Definition

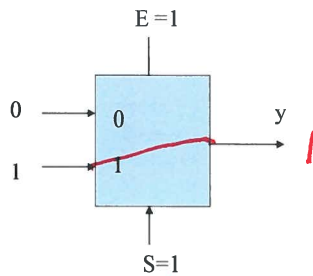
- Selects between one of N inputs to connect to the output.
- $\log_2 N$ -bit select input – control input
- Example: **2:1 Mux**



8

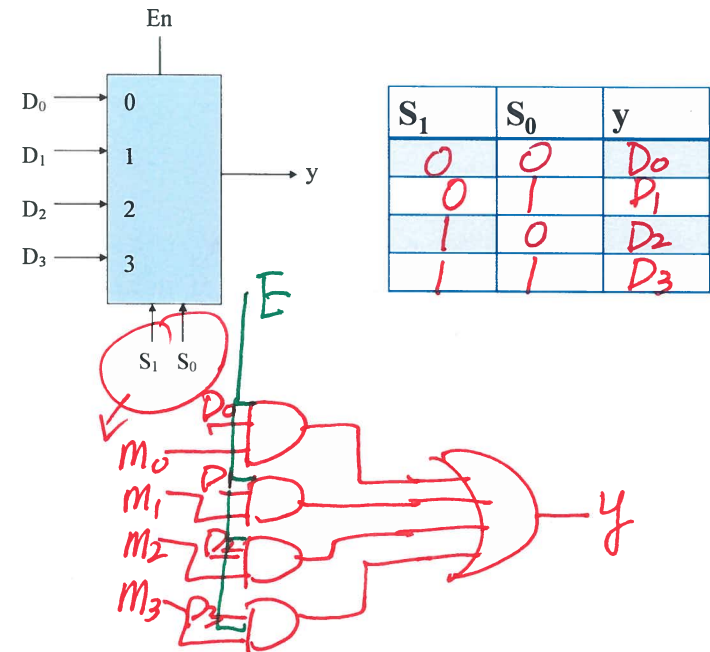
What is the output of the following MUX?

- A.0
- B.1
- C.Can't say



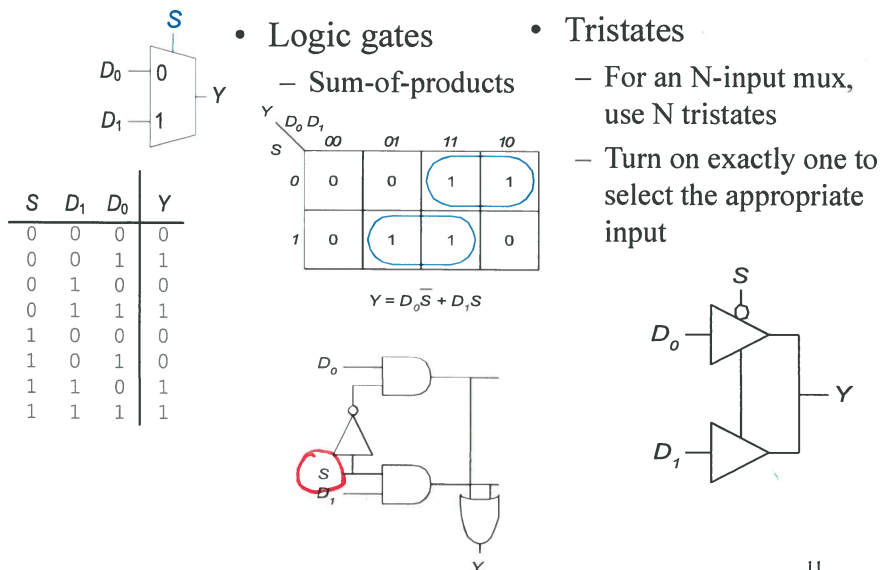
9

Multiplexer Definition: 4-input mux



10

Multiplexer: Logic Diagram



11

Multiplexer Application: universal set {Mux}

We use selector to decompose the function into smaller functions (less number of variables), which follows

Shannon's expansion.

We simplify the decomposed functions using **K-map**, which follows **consensus theorem**.

12

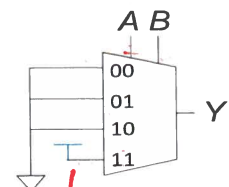
Multiplexer Application

Intel
Universal Set
Xilinx

FPGA CBL

Combinational Building Block

- Mux for a Boolean function with truth table as input
- Building blocks of FPGA (Field Programmable Gate Array).



For the logic diagram on left, output Y is

- A. AB
- B. (AB)'
- C. A+B
- D. (A+B)'
- E. None of the above

Handwritten truth table for the 2-to-1 Mux:

AB	Y
00	0
01	0
10	0
11	1

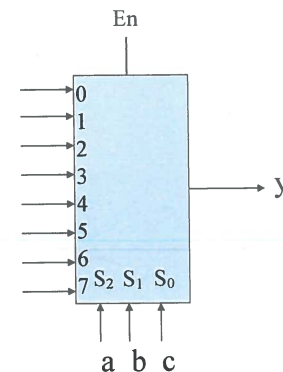
$\Rightarrow AB$

13

Multiplexer Application: universal set {Mux}

Example 1: Given $f(a,b,c) = \sum m(0,1,7) + \sum d(2)$, implement with an 8-input Mux.

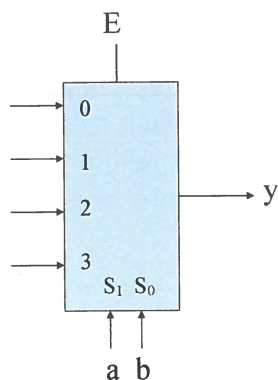
id	abc	f
0	000	1
1	001	1
2	010	-
3	011	0
4	100	0
5	101	0
6	110	0
7	111	1



14

Example 2: Given $f(a,b,c) = \sum m(0,1,7) + \sum d(2)$, implement with 4-input Muxes.

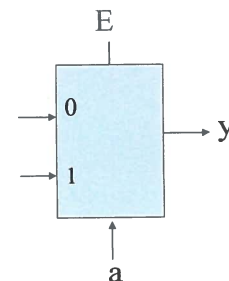
ab	c=0	c=1	D
00			D_0
01			D_1
10			D_2
11			D_3



15

Example 3: Given $f(a,b,c) = \sum m(0,1,7) + \sum d(2)$, implement with 2-input Muxes.

a\bc	00	01	10	11	D(b,c)
0	1	1	-	0	D_0
1	0	0	0	1	D_1



16