Digital Circuits

ECS 371

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Lecture 11

Office Hours:
BKD 3601-7
Monday 9:00-10:30, 1:30-3:30
Tuesday 10:30-11:30

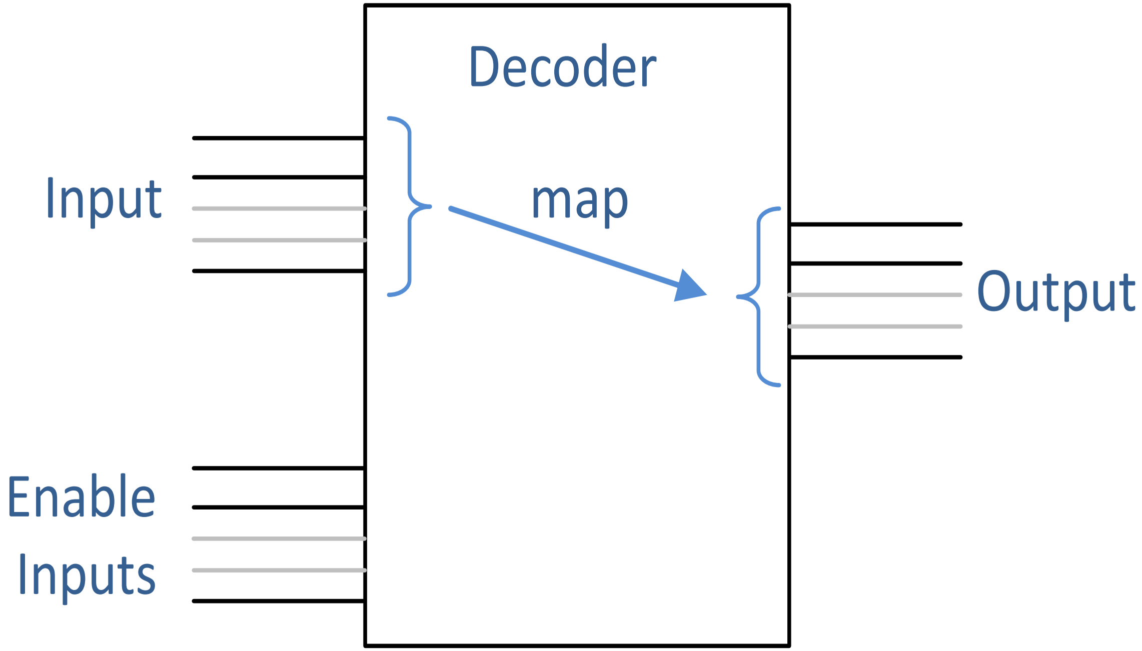
ECS371.PRAPUN.COM

Signals and their Active Levels

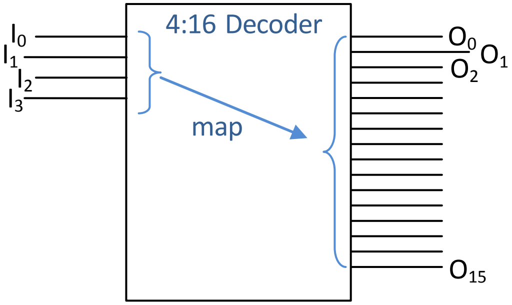
- Each input and output in a logic circuit should have a descriptive alpha-numeric label, the signal’s name.
- A signal is **active high** if it performs the named action or denotes the named condition when it is HIGH (H) or 1.
- A signal is **active low** if it performs the named action or denotes the named condition when it is LOW (L) or 0.
- If not specified, assume active-high signal.
- A signal is said to be **asserted** when it is at its active level.
- A signal is said to be **negated** (or, sometimes, **deasserted**) when it is not at its active level.
- A **bus** is a collection of two or more related signal lines.

Binary-to-Decimal Decoder

- The input has *n* bits
- The output has 2^n bits. Only one bit is asserted at any time.
  - Also known as "1-out-of-*m*" (where $m = 2^n$)
- Zero or more EN (enable) lines



4:16 Decoder



Input				Output															
I_3	I_2	I_1	I_0	O_{15}	O_{14}	O_{13}	O_{12}	O_{11}	O_{10}	O_9	O_8	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: All signals here are active-HIGH.

4:16 Decoder with Active-LOW outputs

Input				Output															
I_3	I_2	I_1	I_0	O_{15}_L	O_{14}_L	O_{13}_L	O_{12}_L	O_{11}_L	O_{10}_L	O_9_L	O_8_L	O_7_L	O_6_L	O_5_L	O_4_L	O_3_L	O_2_L	O_1_L	O_0_L
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

4:16 Decoder with EN

Input					Output															
EN	I_3	I_2	I_1	I_0	O_{15}	O_{14}	O_{13}	O_{12}	O_{11}	O_{10}	O_9	O_8	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: All signals here are active-HIGH.

4:16 Decoder

(Active-HIGH EN)
Active-LOW output

Input					Output															
EN	I_3	I_2	I_1	I_0	O_{15}_L	O_{14}_L	O_{13}_L	O_{12}_L	O_{11}_L	O_{10}_L	O_9_L	O_8_L	O_7_L	O_6_L	O_5_L	O_4_L	O_3_L	O_2_L	O_1_L	O_0_L
0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

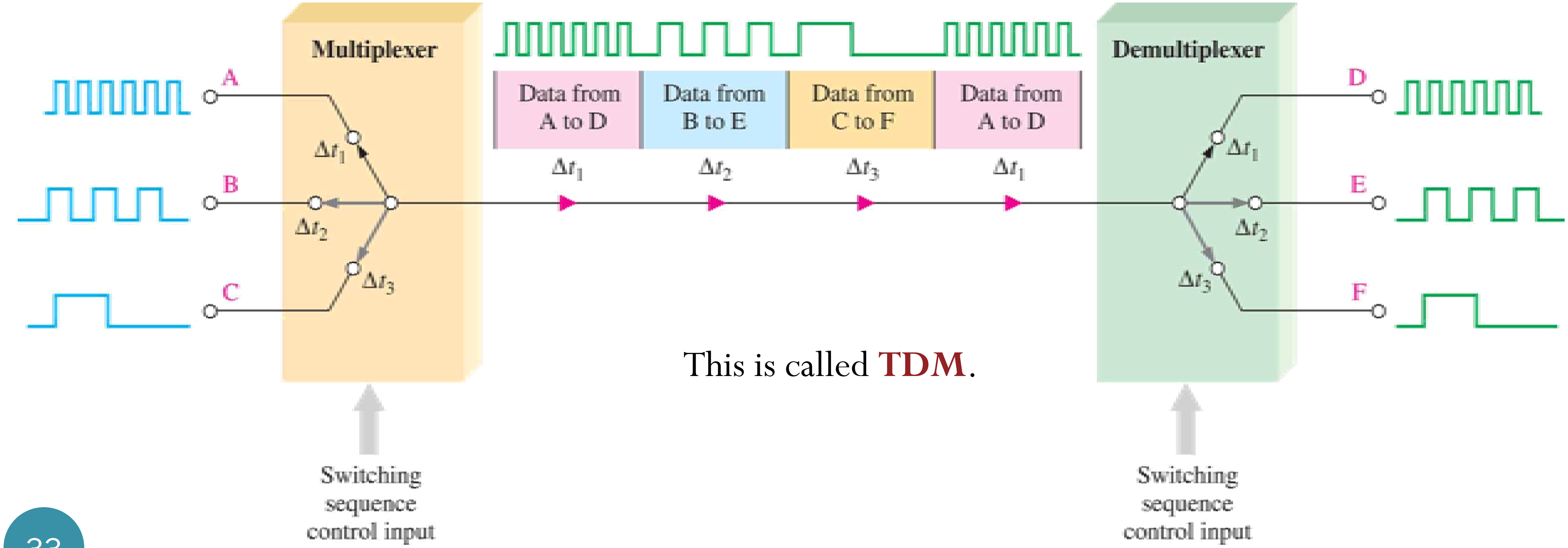
4:16 Decoder

Active-LOW EN
Active-LOW output

Input					Output															
EN_L	I_3	I_2	I_1	I_0	O_{15}_L	O_{14}_L	O_{13}_L	O_{12}_L	O_{11}_L	O_{10}_L	O_9_L	O_8_L	O_7_L	O_6_L	O_5_L	O_4_L	O_3_L	O_2_L	O_1_L	O_0_L
1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Multiplexing/Demultiplexing

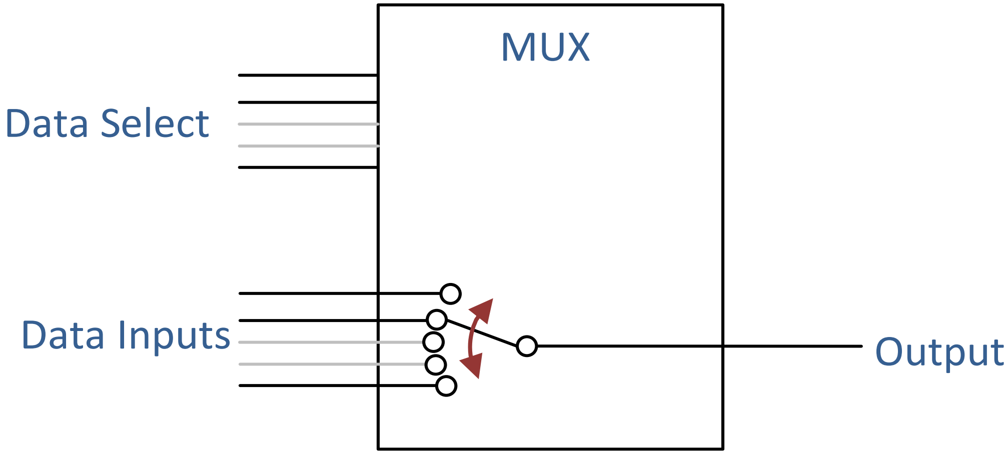
- The **multiplexer**, or **mux** for short, is a logic circuit that switches digital data from several input lines onto a single output line in a specified time sequence.
- The **demultiplexer (demux)** is a logic circuit that switches digital data from one input line to several output lines in a specified time sequence



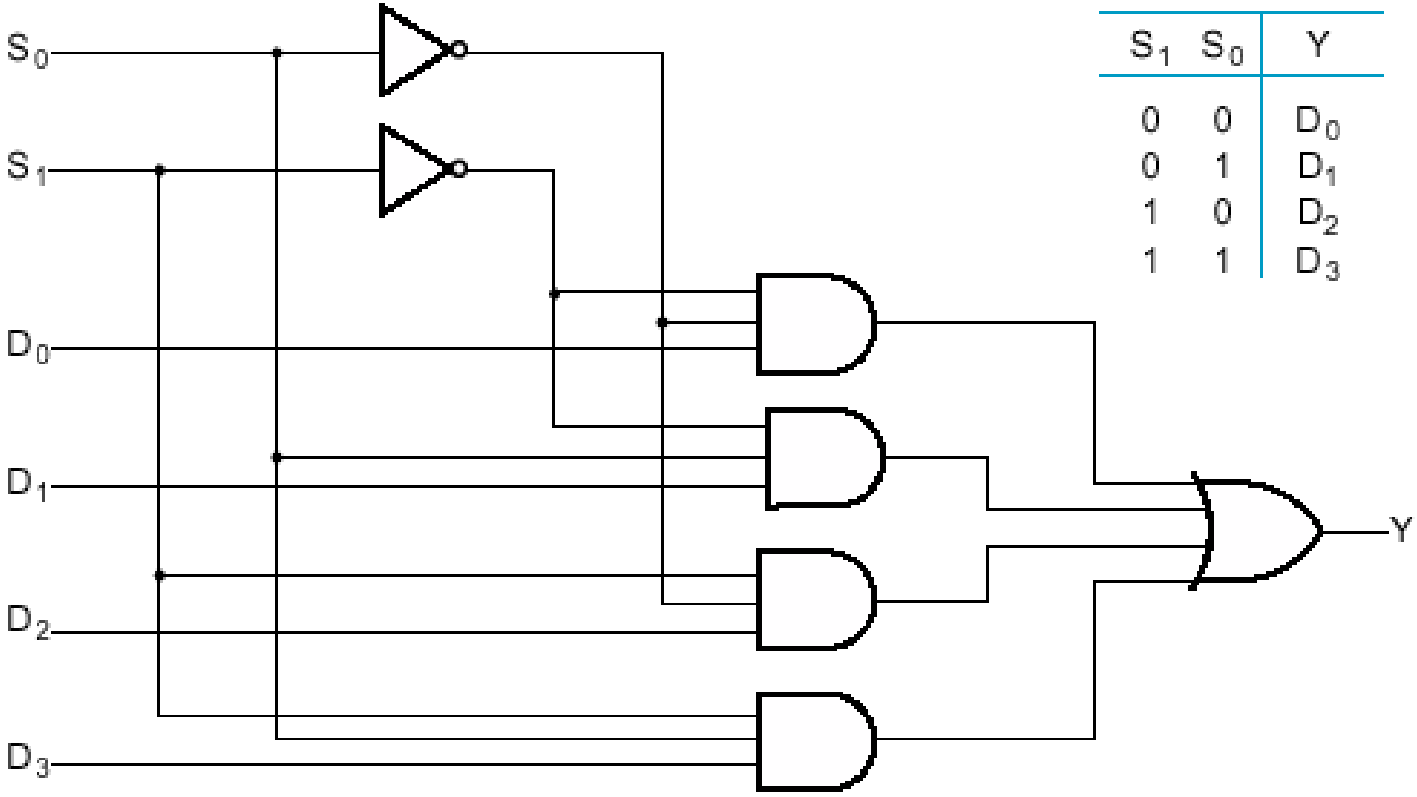
This is called **TDM**.

Multiplexer (Data Selector)

- Select binary information from one of many input lines and directs the information to a single output line.
- Allow digital information from several sources to be routed onto a single line for transmission over that line to a common destination.
- Basic multiplexer has
  1. Data-input lines
  2. Single output line.
  3. Data-select inputs



4-to-1-line multiplexer (4:1 MUX)



Function table

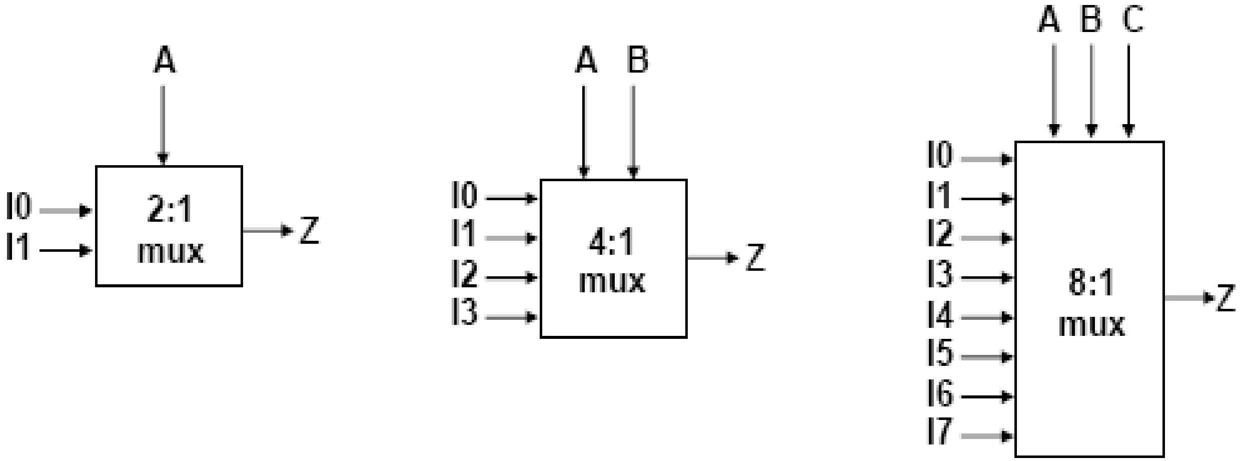
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Example

2:1 mux:Z = A'I0 + AI1

4:1 mux:Z = A'B'I0 + A'BI1 + AB'I2 + ABI3

8:1 mux:Z = A'B'C'I0 + A'B'CI1 + A'BC'I2 + A'BCI3
+ AB'C'I4 + AB'CI5 + ABC'I6 + ABCI7


A
I0
I1
2:1 mux
Z
A B
I0
I1
I2
I3
4:1 mux
Z
A B C
I0
I1
I2
I3
I4
I5
I6
I7
8:1 mux
Z