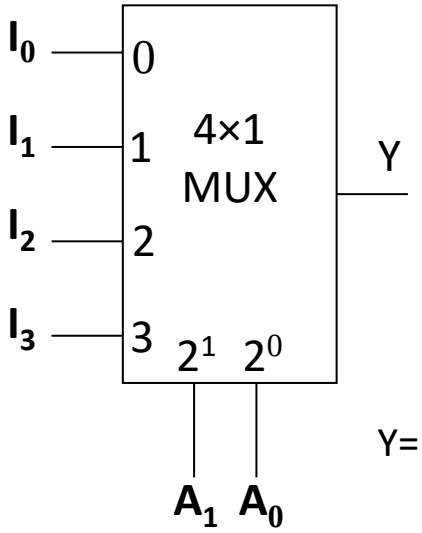
Section 6. Combinational Circuits (Continued)

- **Multiplexer (MUX) / Data Selector**
- **Demultiplexer (DEMUX) / Data Distributor**
- **Seven-Segment Display and Decoder Circuit**

Multiplexer (MUX) / Data Selector

Multiplexers are combinational circuits that have n data selection inputs, 2^n data input lines, and 1 output. Depending on the values of the data selection inputs, they transfer one of the data inputs to the output.

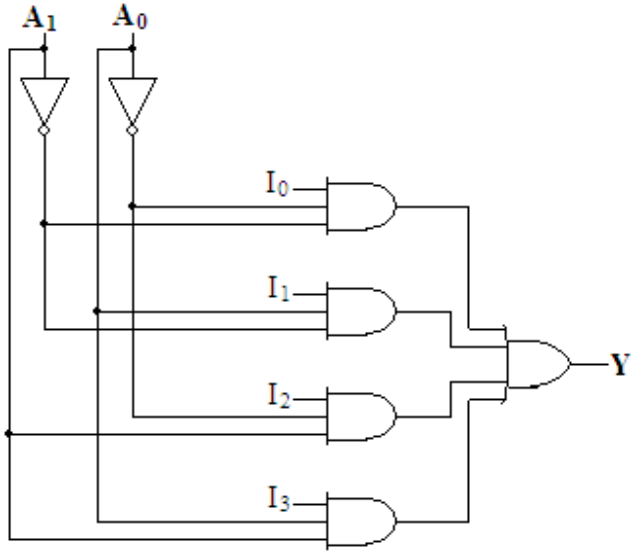
Block diagram, logic circuit, and the truth table for a 4×1 multiplexer (MUX) are shown below.



A_1	A_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$Y = I_0.A_1'.A_0' + I_1.A_1'.A_0 + I_2.A_1.A_0' + I_3.A_1.A_0$

Block diagram of a 4×1 MUX



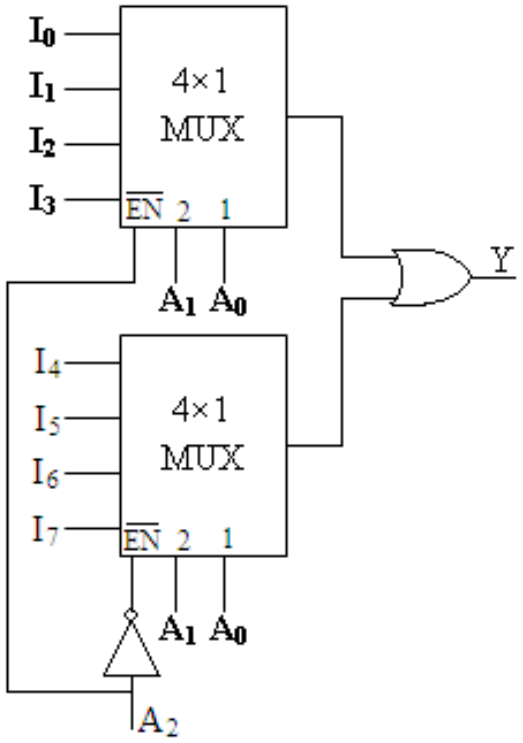
Logic circuit of a 4×1 MUX

Multiplexer (MUX) / Data Selector

By using multiplexers with an **"enable"** input, multiplexers with more data inputs can be created.

Example: Let's create an 8×1 multiplexer by using two 4×1 multiplexers with an enable input.

	A_2	A_1	A_0	Y
Upper MUX active	0	0	0	I_0
	0	0	1	I_1
	0	1	0	I_2
	0	1	1	I_3
Lower MUX active	1	0	0	I_4
	1	0	1	I_5
	1	1	0	I_6
	1	1	1	I_7



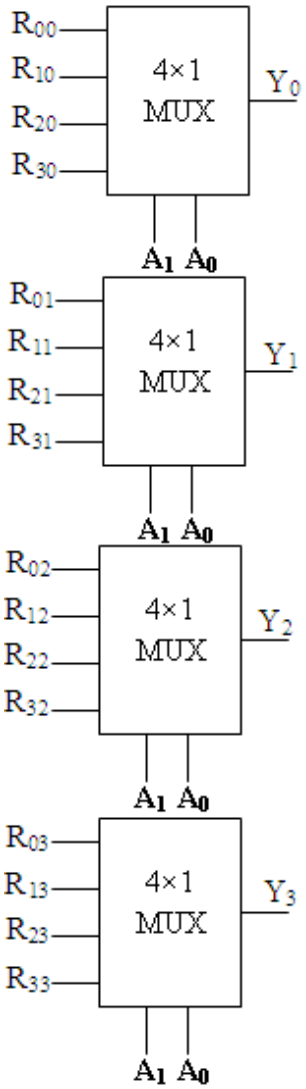
Multiplexer (MUX) / Data Selector

Multiplexers can be used to create a common data bus.

Example: If we want to connect 4 registers to a 4-bit data bus, the circuit shown on the right can be used.

Here, R_0, R_1, R_2, and R_3 are 4-bit registers.

If $A_1A_0 = 00$, the contents of the R_0 register (R_{00}, R_{01}, R_{02}, R_{03}) will be transferred to the bus.



Multiplexer (MUX) / Data Selector

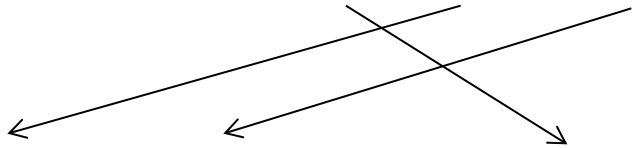
Multiplexers can also be used to implement logical expressions. A combinational circuit with n inputs and 1 output can be implemented using a $2^{n-1}\times1$ multiplexer, which has (n-1) data selection inputs and 2^{n-1} data input lines.

Example: Let's implement the function f(a,b,c)=abc+a'b'+bc' using a 4×1 MUX by selecting the data selection inputs as a and b.

Since a and b are selected as the data selection variables, all terms should be expanded to include these variables.

f(a,b,c)=abc+a'b'+(a+a')bc'= abc+a'b'+abc'+a'bc'= ab + a'b' + a'bc'

Output expression for the MUX is:



$$Y= I_0.a'.b'+ I_1.a'.b+ I_2.a.b'+ I_3.a.b$$

Then, $I_0 = 1$, $I_1 = c'$, $I_2 = 0$, $I_3=1$

Example: (continued)

f(a,b,c) = ab+a'b'+a'bc'

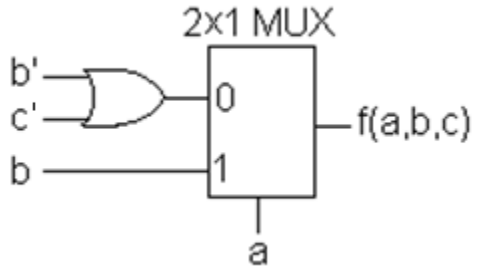
If we want to implement the same logical expression using a 2×1 MUX with "a" as the data selection input, we will need to use additional gates.

We need to write the f function in terms of a and a'.

f(a,b,c) = ab+a'b'+a'bc' = ab+a'(b'+bc') = ab+a'(b'+c')

Output expression for the MUX is: $Y = I_0.a' + I_1.a$

Then, $I_0 = b'+c'$ and $I_1 = b$



Pay attention that we had to use an extra OR gate.

Multiplexer (MUX) / Data Selector

A systematic approach has been developed to implement a logical expression given in standard product form using a multiplexer with the appropriate number of data selection inputs.

Example: Let the logical expression of a combinational circuit with 3 inputs and 1 output be $f(a,b,c)=\Sigma(1,3,4,7)$. Suppose we want to implement this logical expression using a 4×1 MUX with b and c as the selection inputs.

a \ bc	00	01	10	11
0	m_0	m_1	m_2	m_3
1	m_4	m_5	m_6	m_7
Connection	**a**	**a'**	**0**	**1**
MUX Data Inputs	I_0	I_1	I_2	I_3

The marked minterms are used to define the MUX's I_0, I_1, I_2, and I_3 inputs as follows.

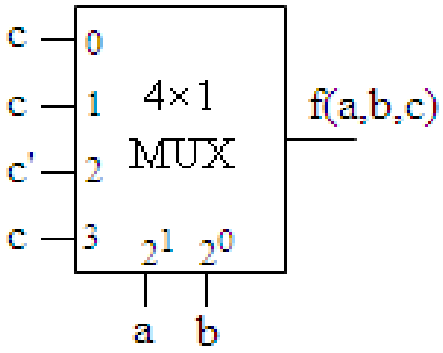
If the marked minterm corresponds to a row where a is 1, the connection value is a; if a is 0, the connection value is a'; if both rows are marked, the connection value is 1; and if neither is marked, the connection value is 0.

Example: (continued)

f(a,b,c)=Σ(1,3,4,7)

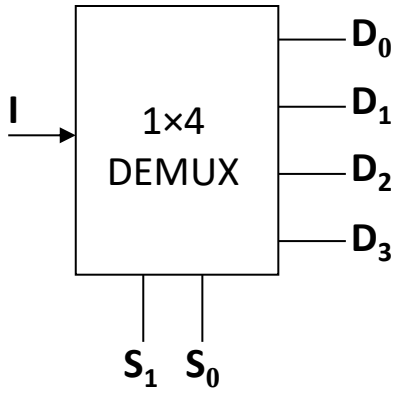
Let's implement the same logical expression by taking a and b as selection inputs.

ab / c	00	01	10	11
0	m_0	m_2	m_4	**m_6**
1	m_1	m_3	m_5	**m_7**
Connection	**c**	**c**	**c'**	**c**
MUX Data Inputs	**I_0**	**I_1**	**I_2**	**I_3**



Demultiplexer (DEMUX) / Data Distributor

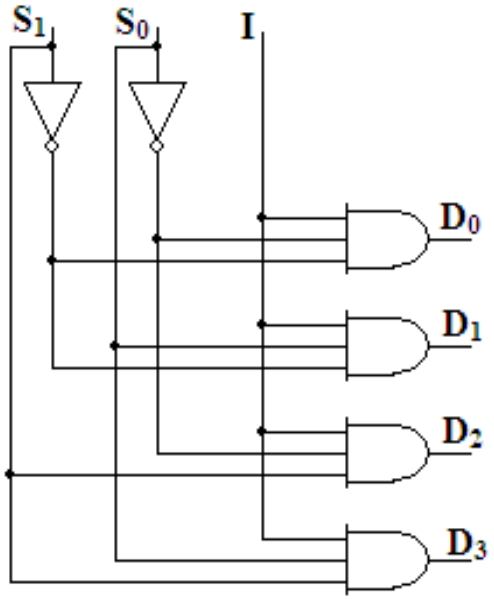
Demultiplexers are combinational circuits with n select lines, 1 input, and 2^n outputs. The number formed by the select lines indicates which output the input will be directed to. Below is the block diagram, truth table, and the logic circuit of a 1×4 DEMUX.



Block diagram of
1×4 DEMUX

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

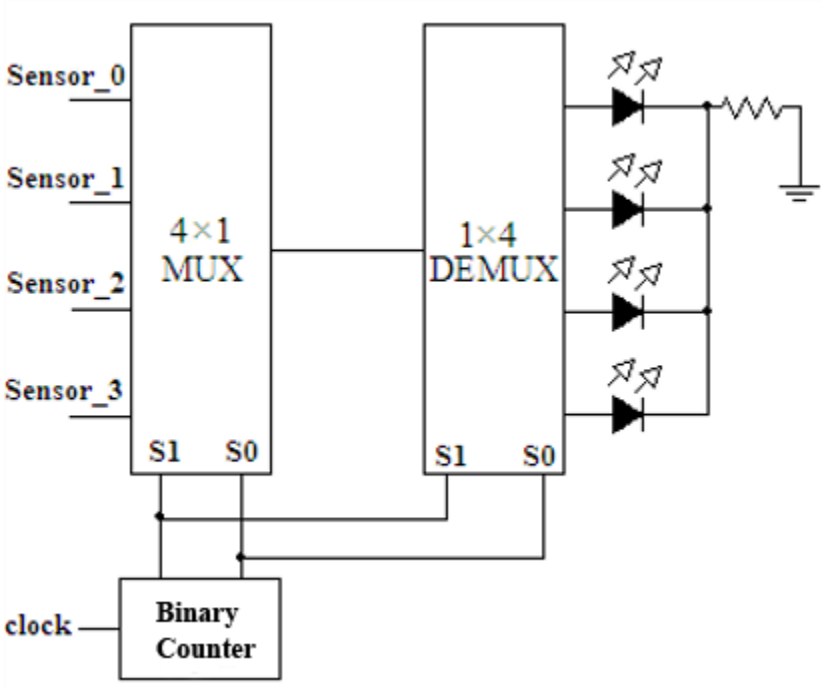
Truth table for
1×4 DEMUX



Logic diagram of
1×4 DEMUX

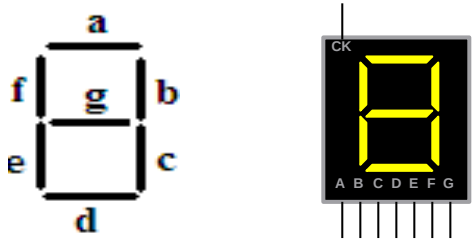
Demultiplexer (DEMUX) / Data Distributor

Example: MUX and DEMUX can be used together, especially for transmitting data to remote locations. For example, information from 4 sensors can be monitored from a distant display panel. If data from sensor 1 is received, the LEDs at the DEMUX output will blink. The counter used ensures that the information at the MUX inputs is transmitted in sequence. The frequency of the counter also determines the blinking interval of the LEDs.

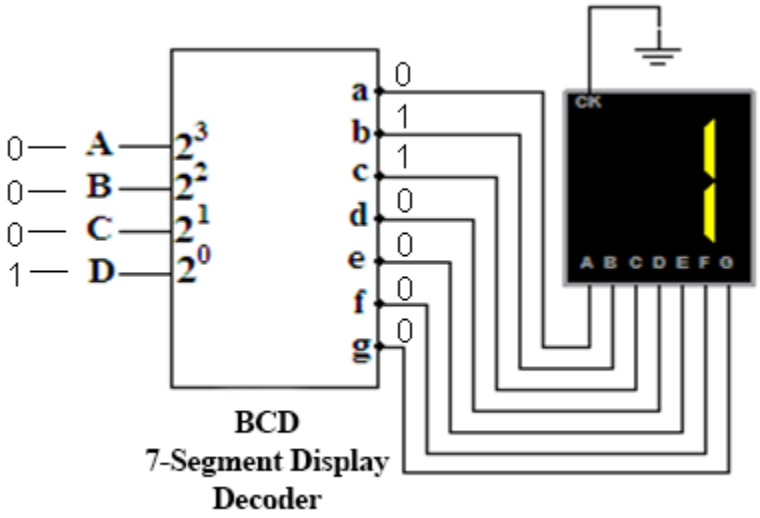


7-Segment Display and Decoder Circuit

Seven-segment displays are commonly used in logic circuits, especially to display decimal numbers. They are formed by combining 7 LEDs. They can be produced as common anode or common cathode. A decoder is needed to drive this display. In this section, a decoder from BCD to a 7-segment display will be designed. This decoder can be obtained as an integrated circuit (IC).



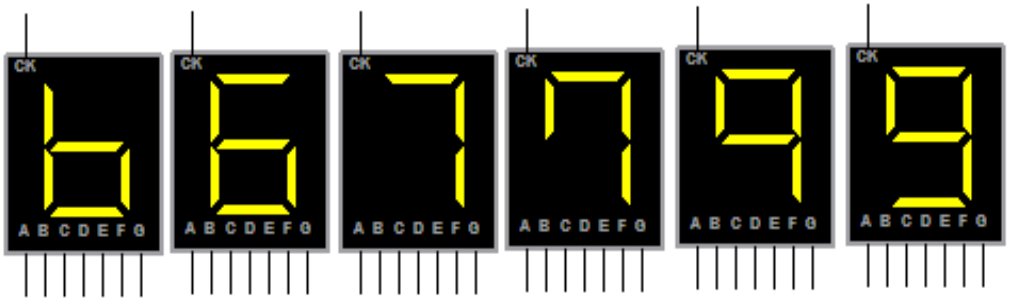
Seven-segment display



Block diagram of a decoder for a seven-segment display

Design of the Decoder for a Seven-Segment Display

To design the decoder, we can either create the truth table or directly map the outputs onto the Karnaugh map. Considering that 6, 7, and 9 can have two different representations, the appropriate representation will be selected during the design phase to simplify the outputs as much as possible.



a

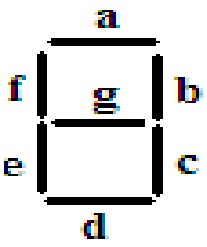
CD \ AB	00	01	11	10
00	1		x	1
01		1	x	1
11	1	1	x	x
10	1	x	x	x

a=A+C+BD+B'D'

b

CD \ AB	00	01	11	10
00	1	1	x	1
01	1		x	1
11	1	1	x	x
10	1		x	x

b=B'+CD+C'D'



Design of the Decoder for a Seven-Segment Display

c

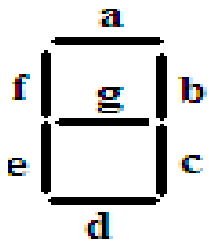
AB / CD	00	01	11	10
00	1	1	x	1
01	1	1	x	1
11	1	1	x	x
10		1	x	x

c=B+C'+D

d

AB / CD	00	01	11	10
00	1		x	1
01		1	x	x
11	1		x	x
10	1	1	x	x

d=B'D'+CD'+B'C+BC'D



Design of the Decoder for a Seven-Segment Display

e

CD \ AB	00	01	11	10
00	1		x	1
01			x	
11			x	x
10	1	1	x	x

e=B'D'+CD'

f

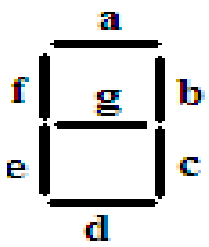
CD \ AB	00	01	11	10
00	1	1	x	1
01		1	x	1
11		x	x	x
10		1	x	x

f=A+B+C'D'

g

CD \ AB	00	01	11	10
00		1	x	1
01		1	x	1
11	1		x	x
10	1	1	x	x

g=A+B'C+CD'+BC'



Design of the Decoder for a Seven-Segment Display