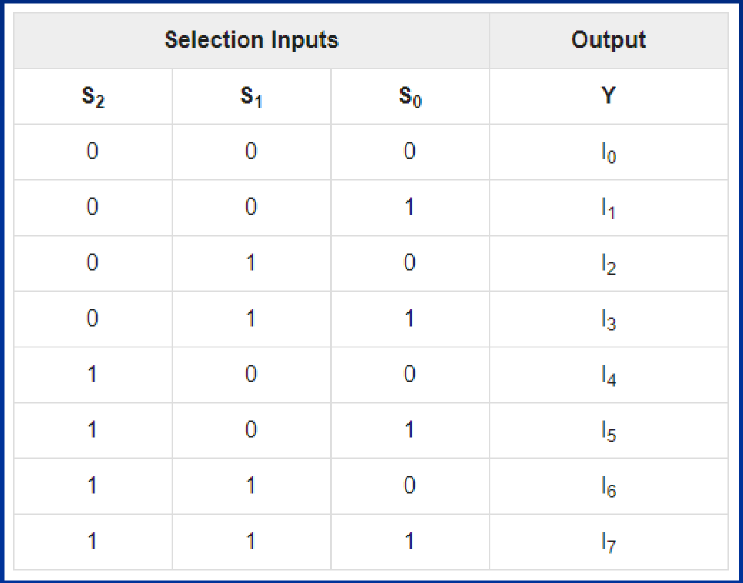
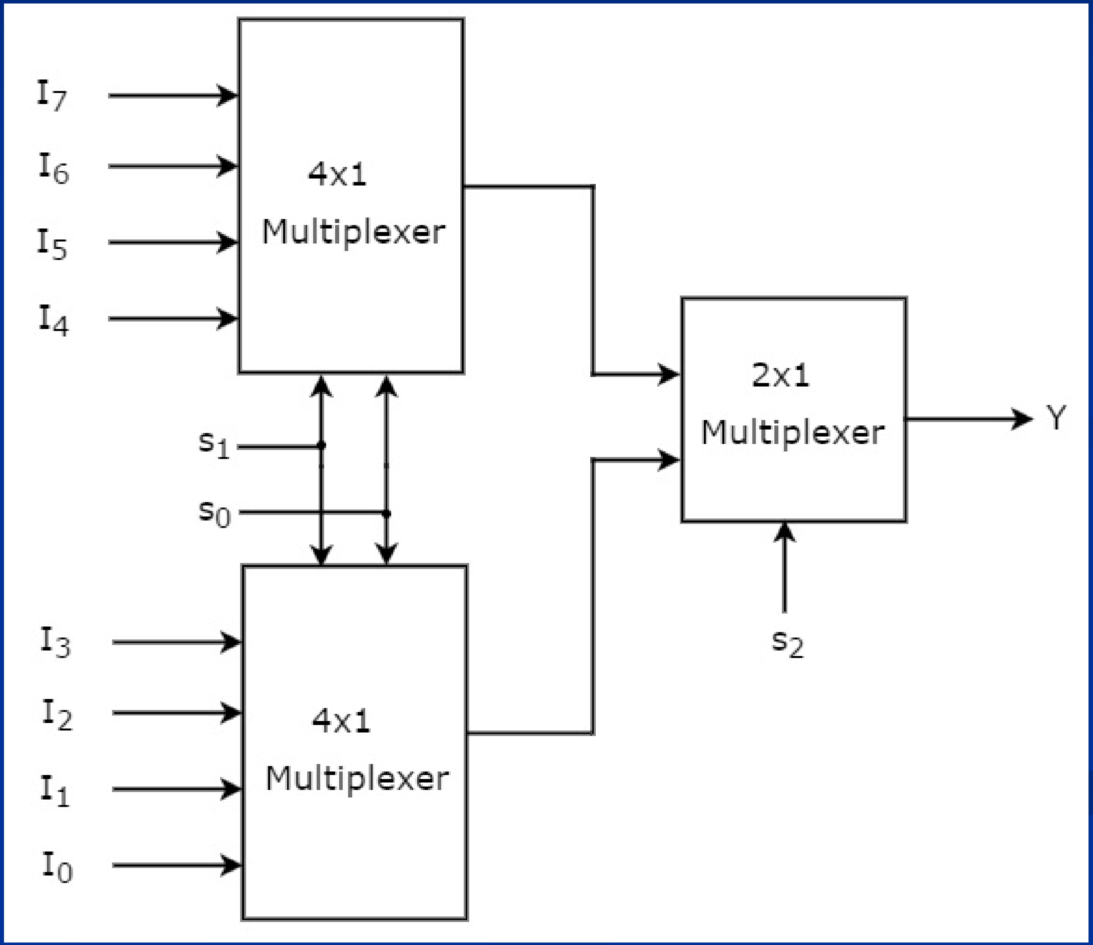
Implementation of Higher-order Multiplexers using Lower-order Multiplexers

Implementation of 8x1 Multiplexer using 4x1 Multiplexers and 2x1 Multiplexer

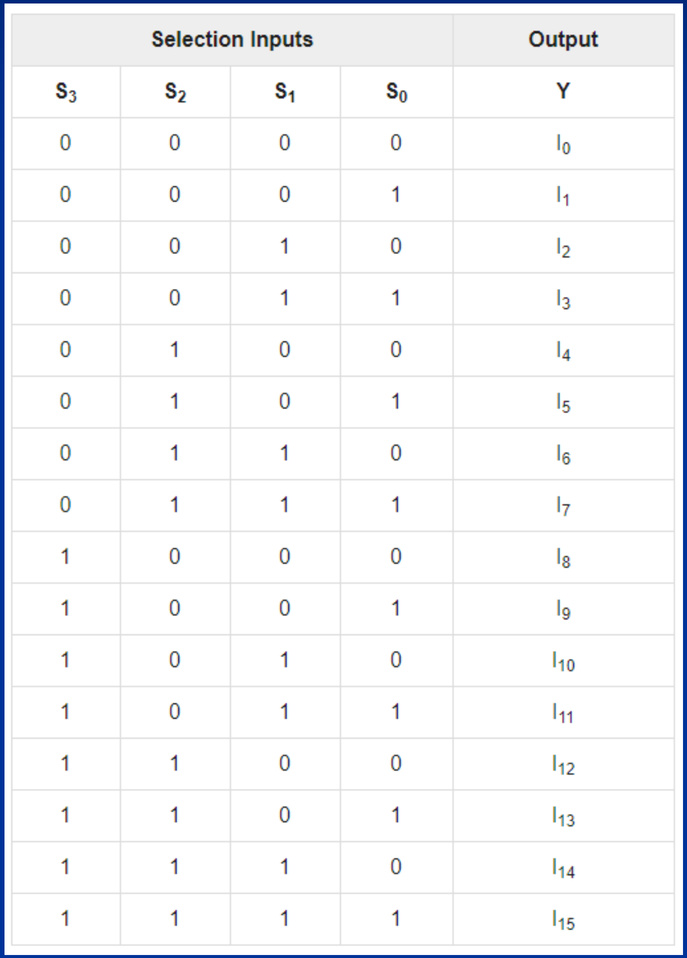


Selection Inputs			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

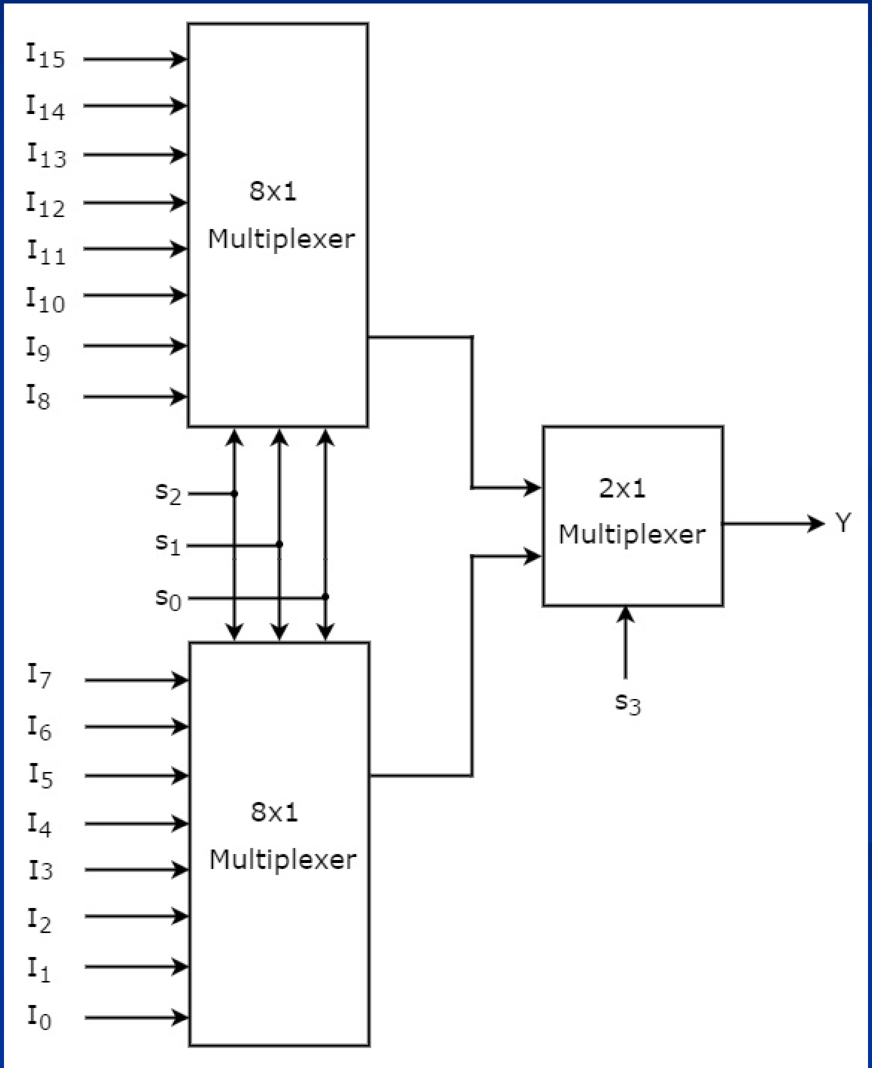


Implementation of Higher-order Multiplexers using Lower-order Multiplexers (cont.)

Implementation of 16x1 Multiplexer using 8x1 Multiplexers and 2x1 Multiplexer



Selection Inputs				Output
S_3	S_2	S_1	S_0	Y
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}



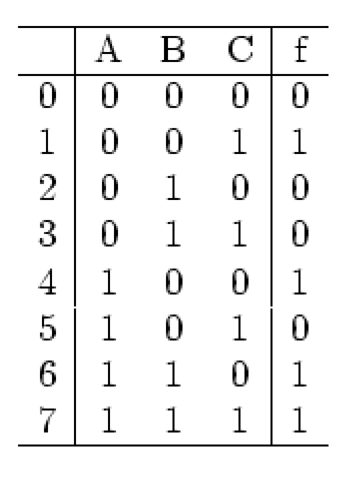
Implementing Boolean Functions with Multiplexers

- Any Boolean function of *n* variables can be implemented using a 2^n-to-1 multiplexer.
- The SELECT signals generate the minterms of the function.

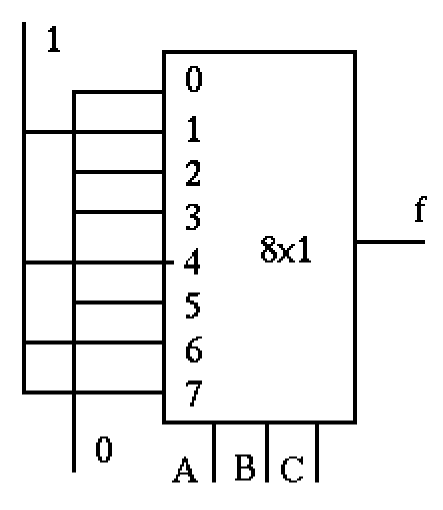
Implement function

Implementing Boolean Functions with Multiplexers (cont.)

Implementation of $f(A,B,C) = \sum(1,4,6,7)$



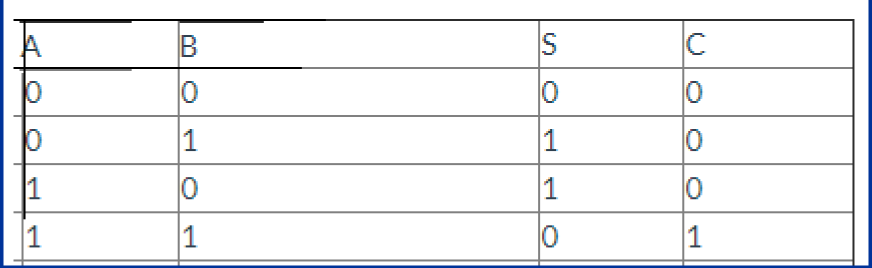
	A	B	C	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1



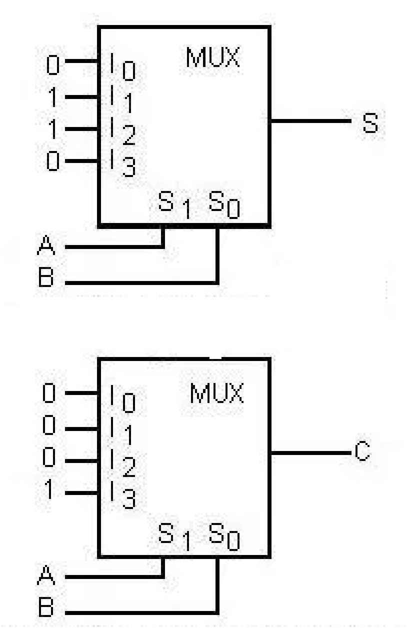
Implement function

Implementing Boolean Functions with Multiplexers (cont.)

Implementation of Half Adder using 4x1 Muxes



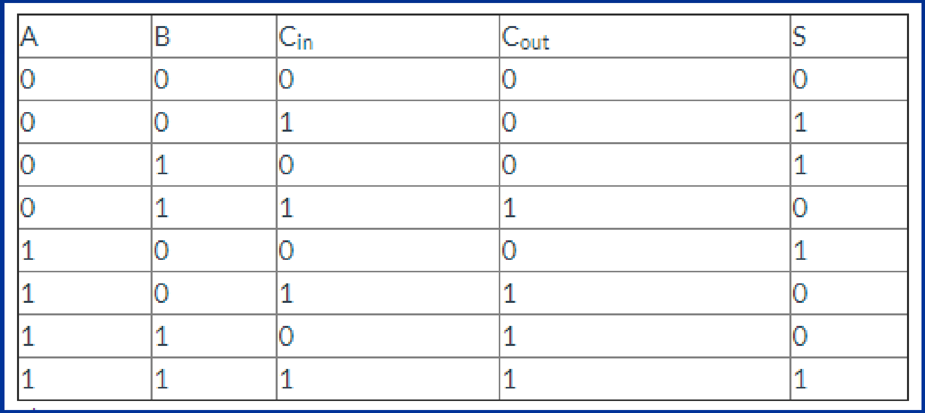
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



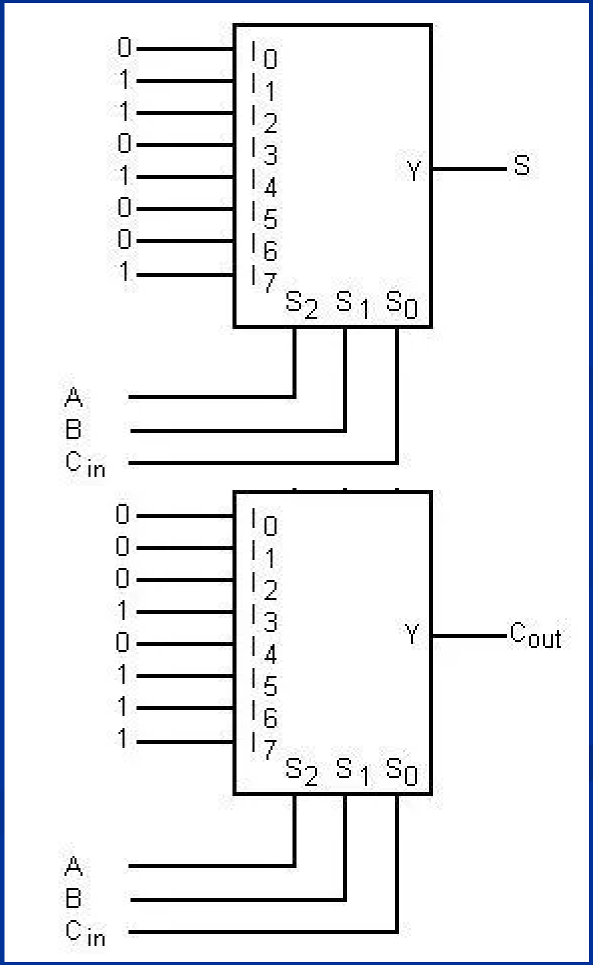
Implement function

Implementing Boolean Functions with Multiplexers (cont.)

Implementation of Full Adder using 8x1 Muxes



A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1


0
1
1
0
1
0
0
1
I_0
I_1
I_2
I_3
I_4
I_5
I_6
I_7
Y
S
S_2
S_1
S_0
A
B
C_{in}
0
0
0
1
0
1
1
1
I_0
I_1
I_2
I_3
I_4
I_5
I_6
I_7
Y
C_{out}
S_2
S_1
S_0
A
B
C_{in}