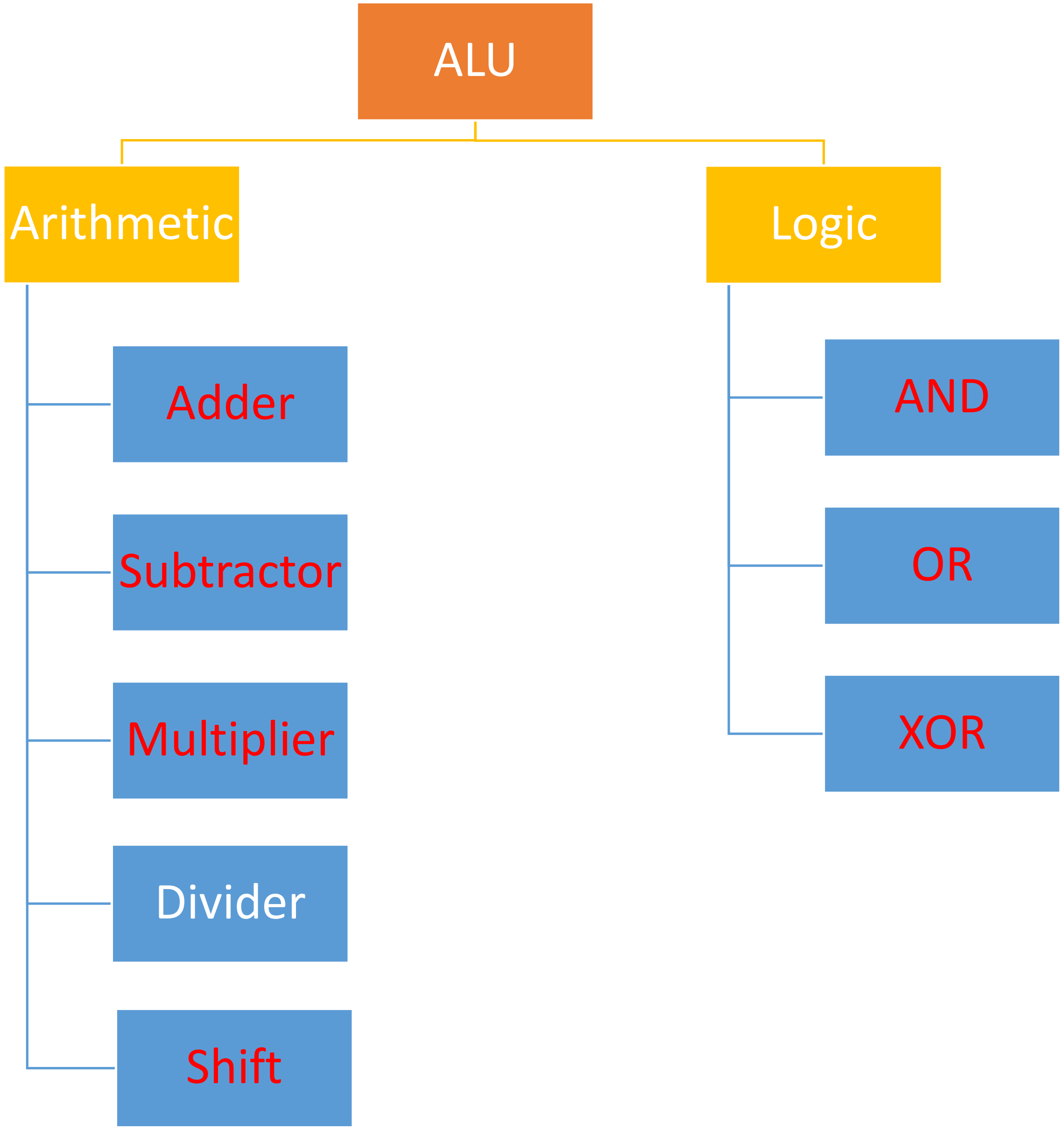
ALU Design III

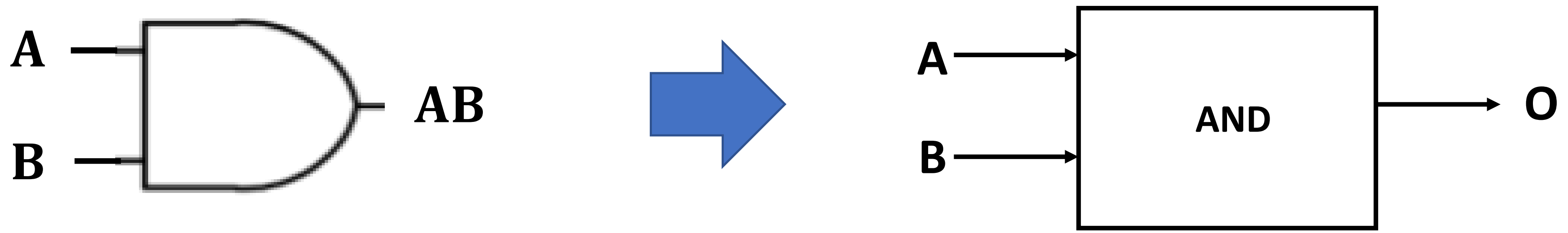
Nahin Ul Sadad
Lecturer
CSE, RUET


ALU
Arithmetic
Logic
Adder
Subtractor
Multiplier
Divider
Shift
AND
OR
XOR

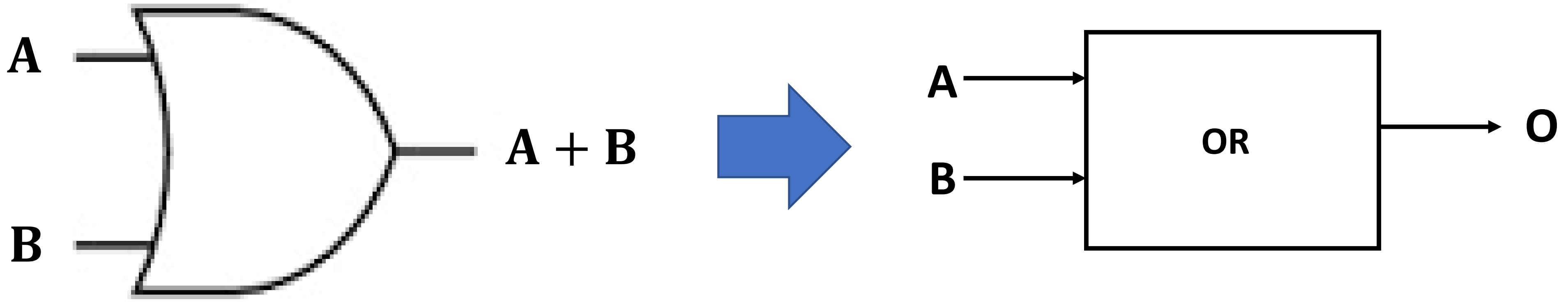
1-bit ALU design

s_1

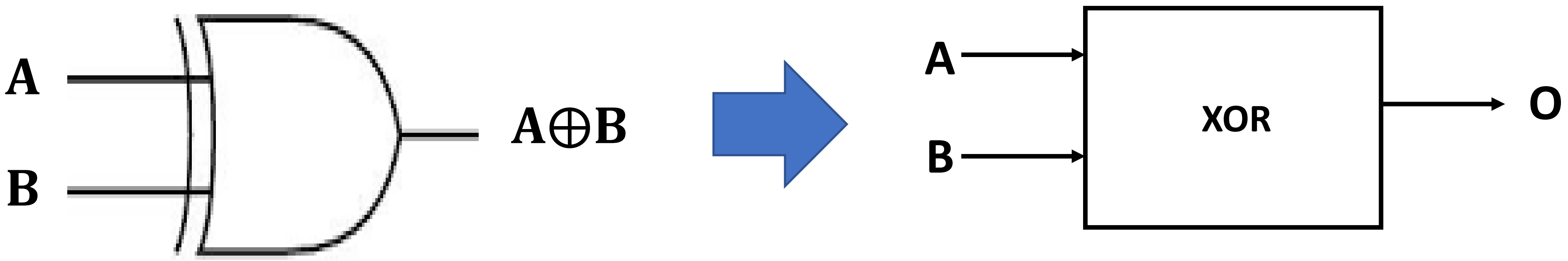
1-bit AND gate



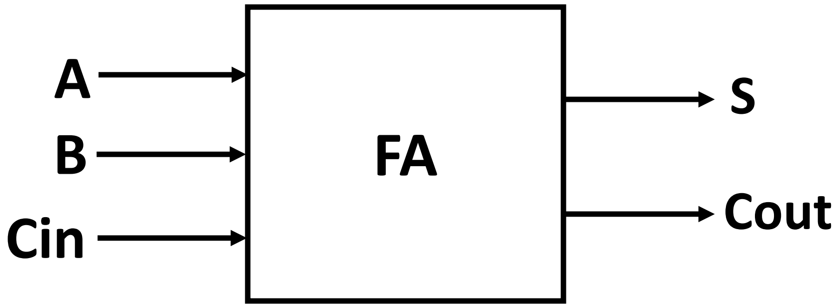
1-bit OR gate



1-bit XOR gate



1-bit Adder

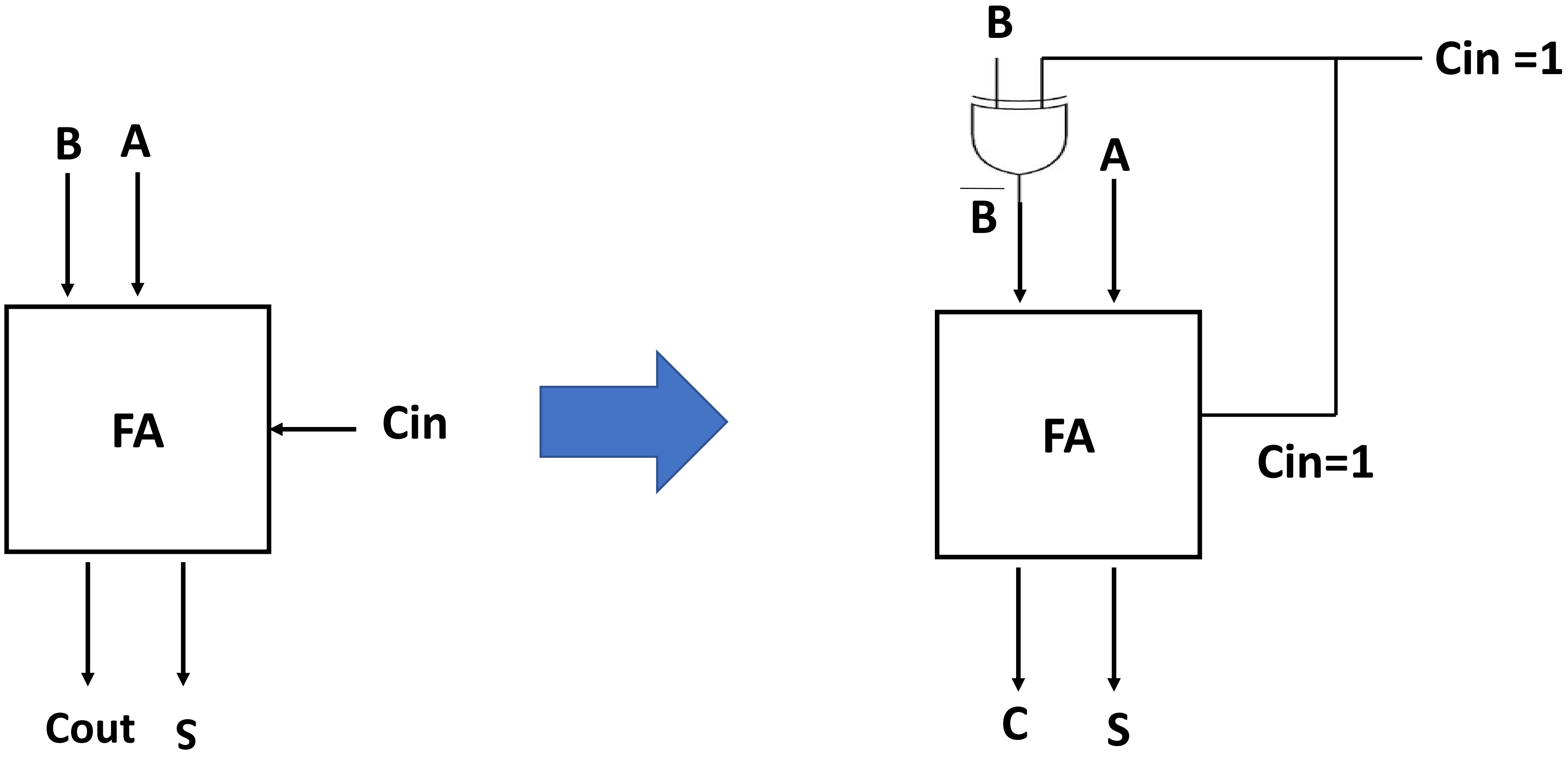


A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

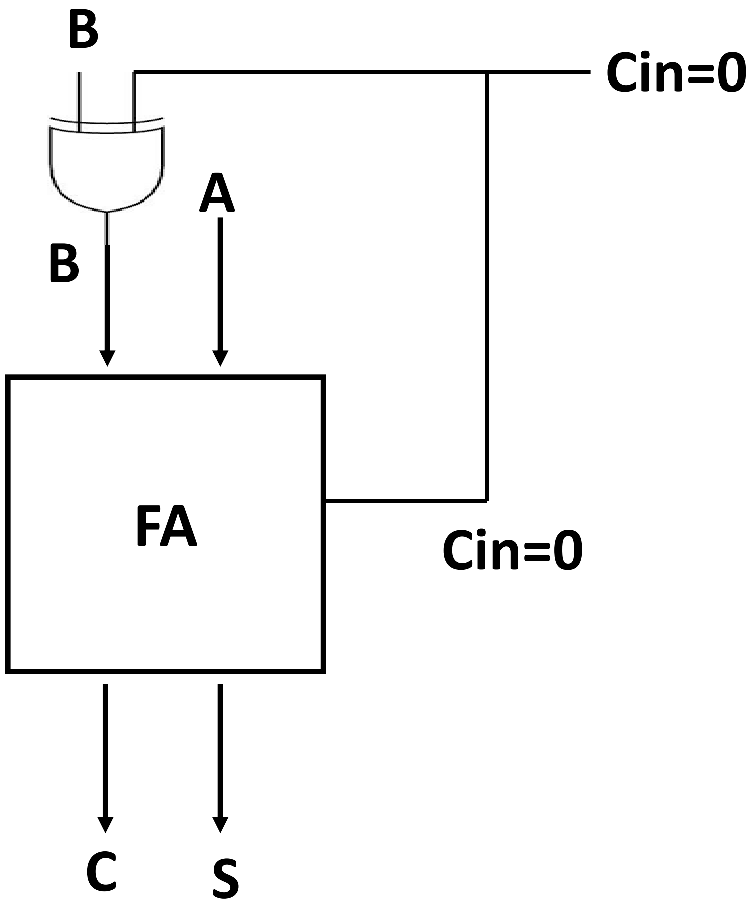
$$S = A\,\overline{B}\,\overline{Cin} + \overline{A}\,B\,\overline{Cin} + \overline{A}\,\overline{B}\,Cin + A\,B\,Cin$$

$$Cout = A\,B + A\,C + B\,C$$

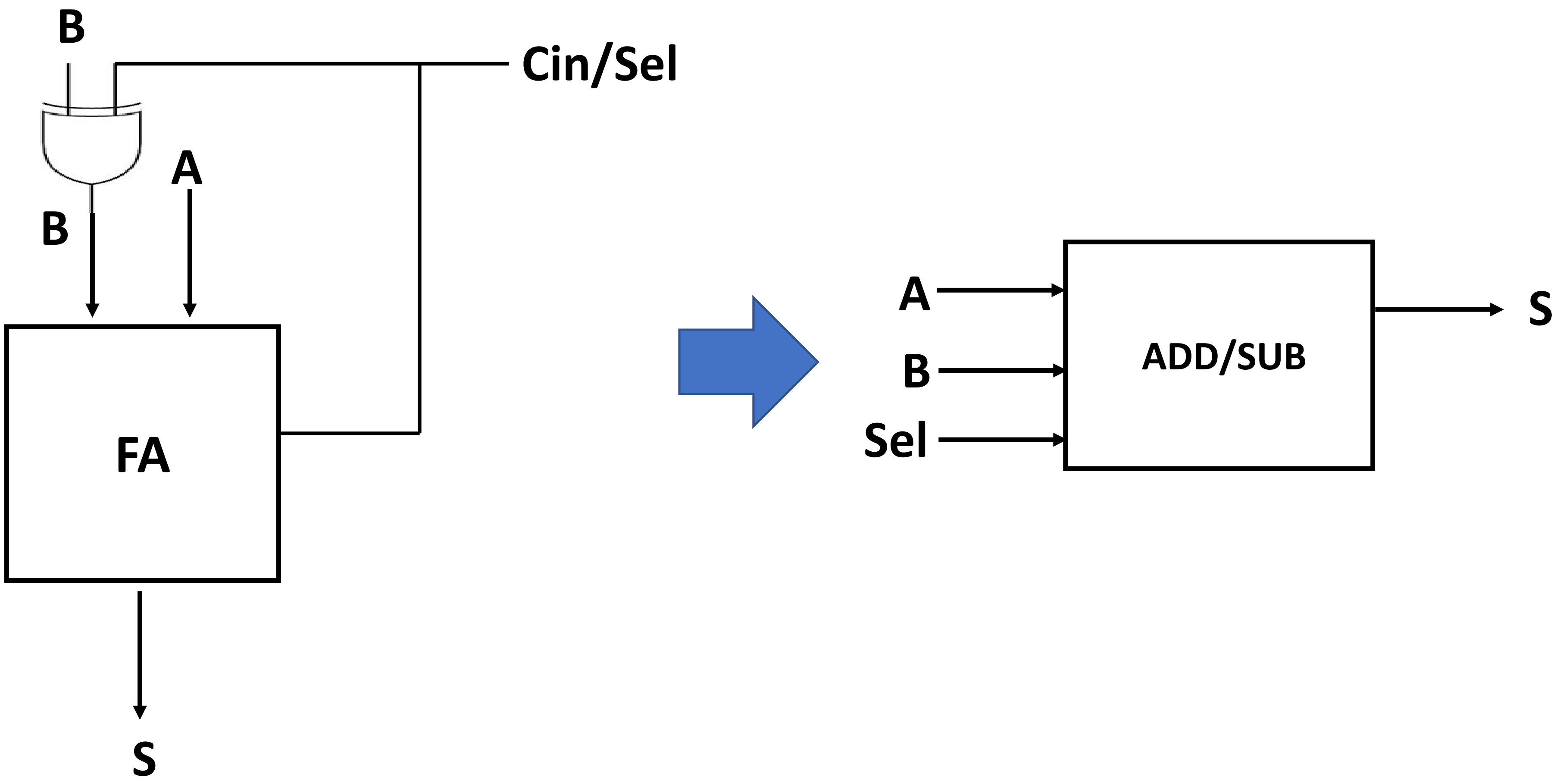
1-bit Subtractor



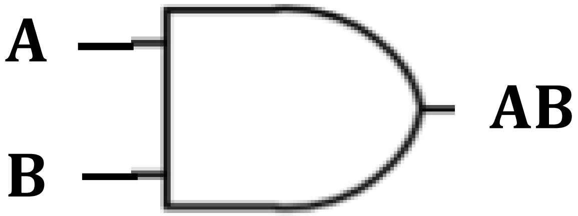
1-bit Adder (Modified)



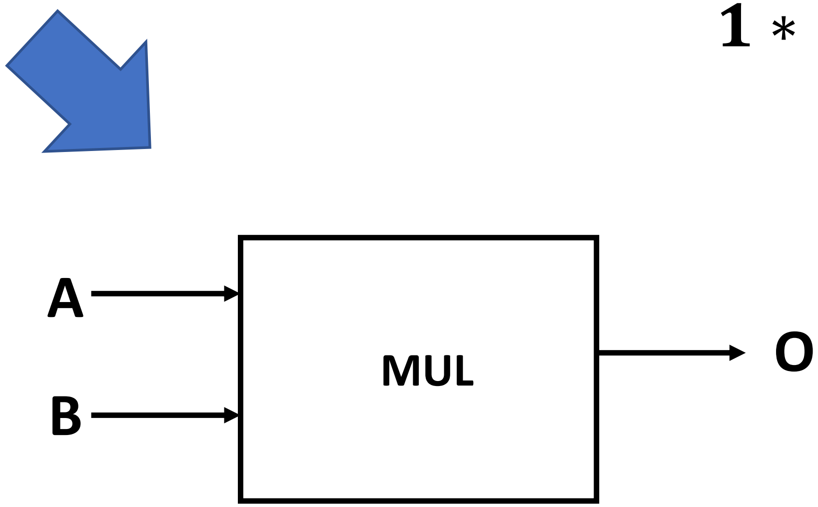
1-bit Adder/Subtractor



1-bit Multiplier

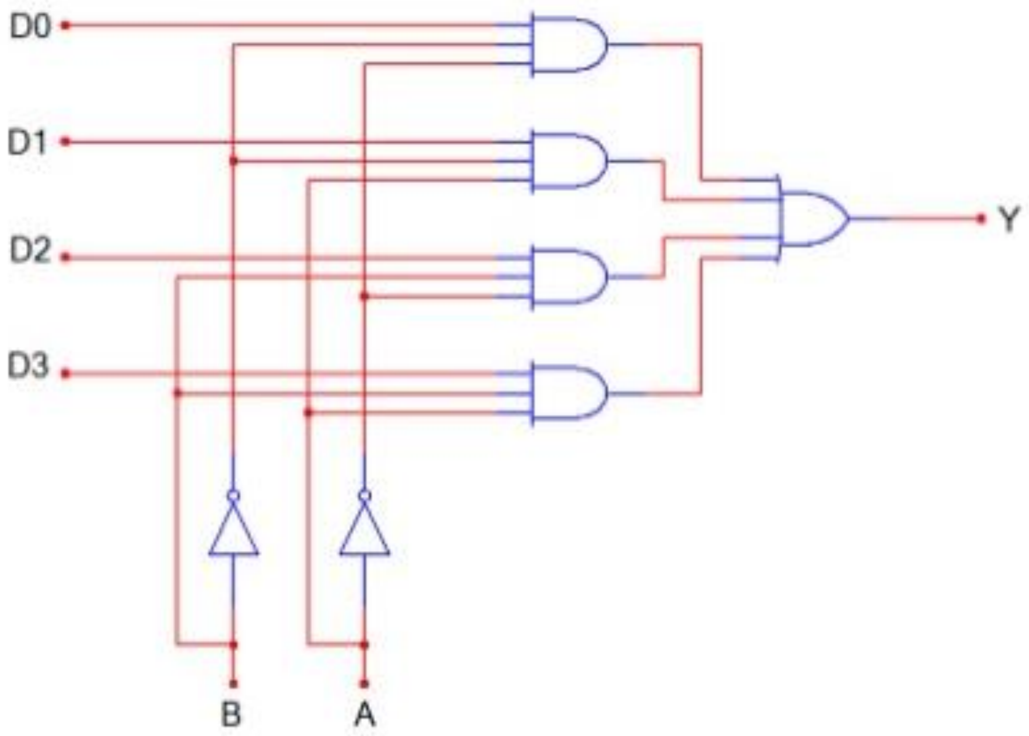


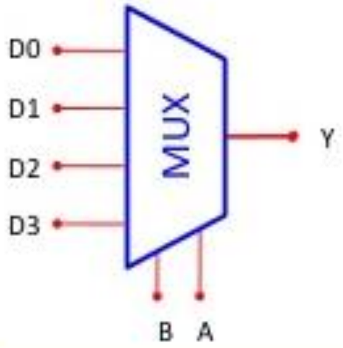
$$\begin{array}{ccccc} A & & B & & M \\ 0 & * & 0 & = & 0 \\ 0 & * & 1 & = & 0 \\ 1 & * & 0 & = & 0 \\ 1 & * & 1 & = & 1 \end{array}$$



1-bit Shifter

4-to-1 Multiplexer (MUX)

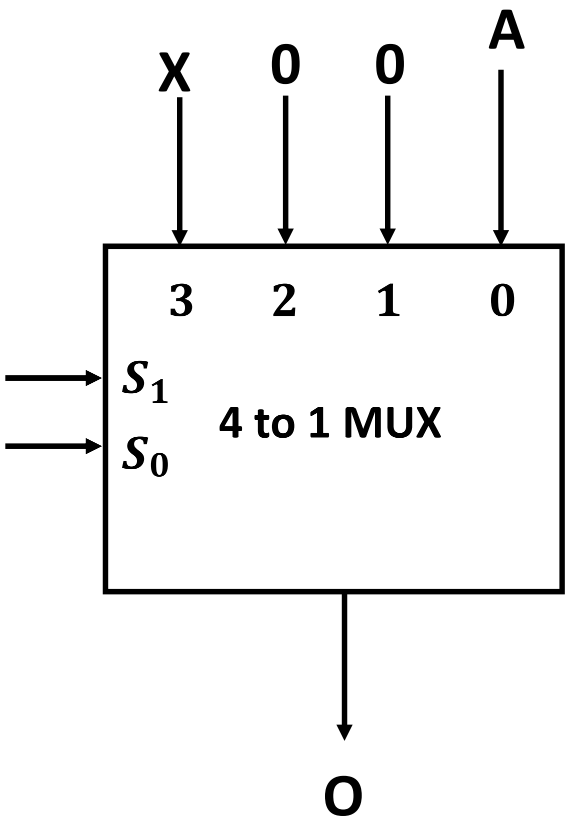




B	A	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$$\mathbf{Y = \bar{A}.\bar{B}.D0 + \bar{A}.B.D1 + A.\bar{B}.D2 + A.B.D3}$$

1-bit Shifter

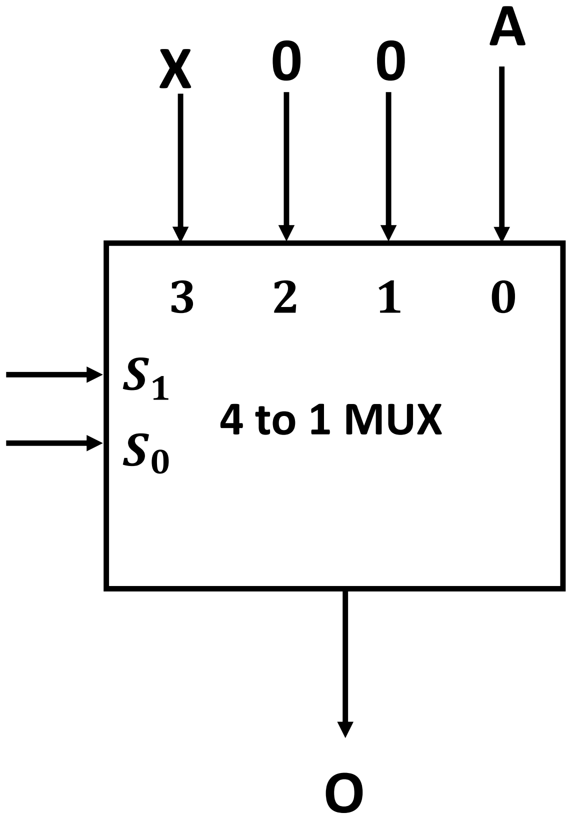


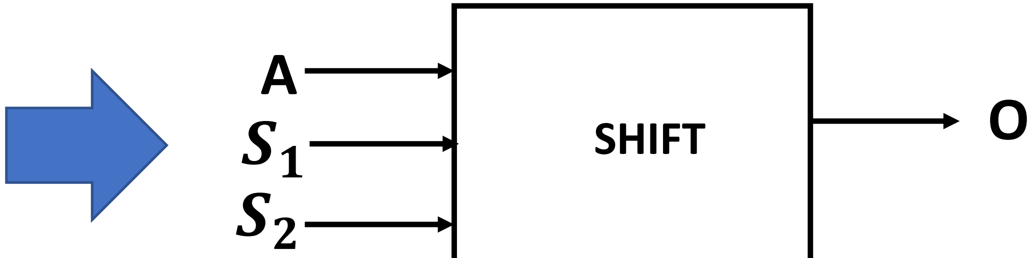
S_1	S_2	Output	Operation
0	0	A	No Shift
0	1	0	Left Shift
1	0	0	Right Shift
1	1	X	X

Input: A
Right shift: 0A
Left shift: A0

Input: 1
Right shift: 01
Left shift: 10

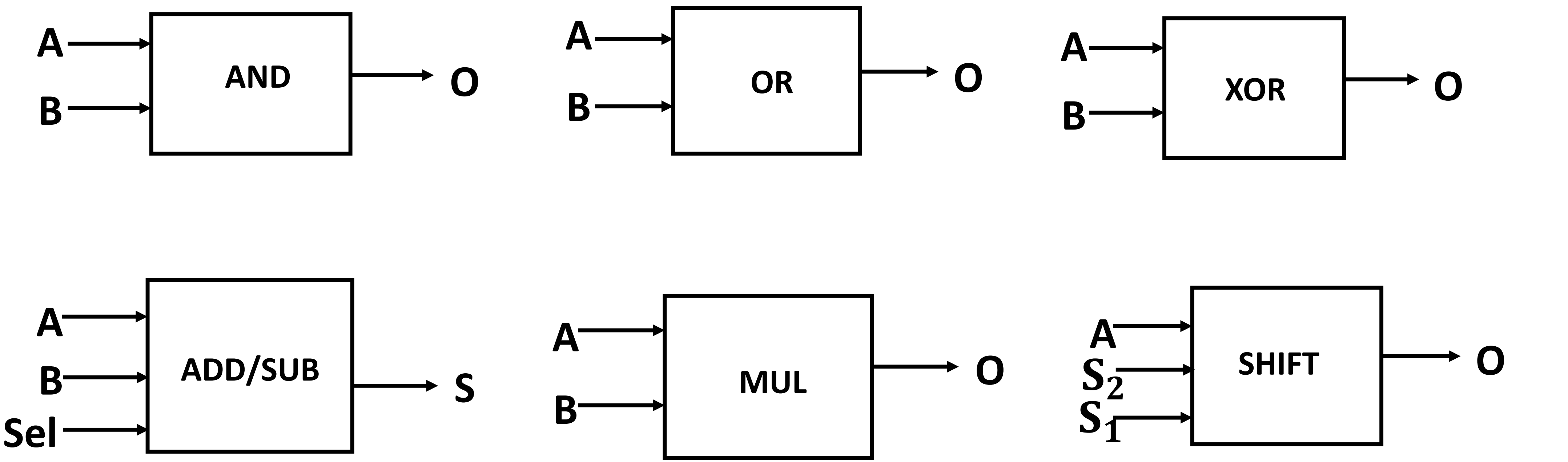
1-bit Shifter





S_1	S_2	Output	Operation
0	0	A	No Shift
0	1	0	Left Shift
1	0	0	Right Shift
1	1	X	X

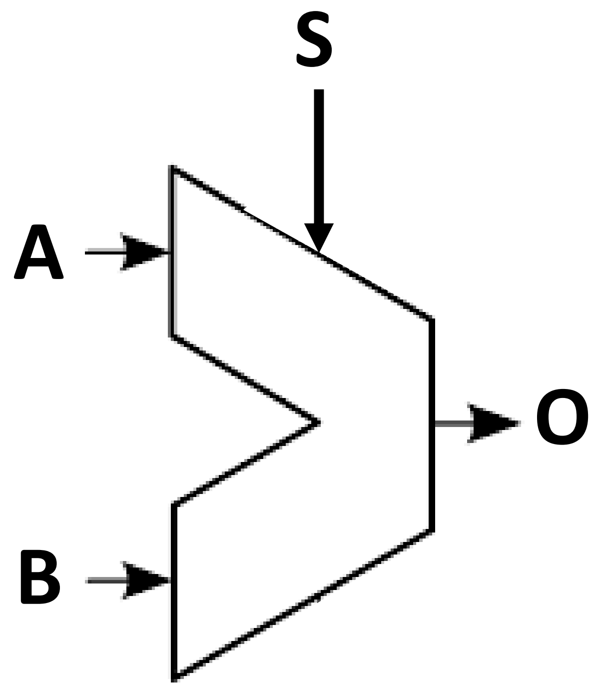
All the circuits so far



If $\mathbf{sel = 0,\ \ S = ADD}$
If $\mathbf{sel = 1,\ \ S = SUB}$

If $\mathbf{S_1 = 0, S_0 = 1, S = LEFT\ SHIFT}$
If $\mathbf{S_1 = 1, S_0 = 0, S = RIGHT\ SHIFT}$

ALU Circuit



1-bit CPU

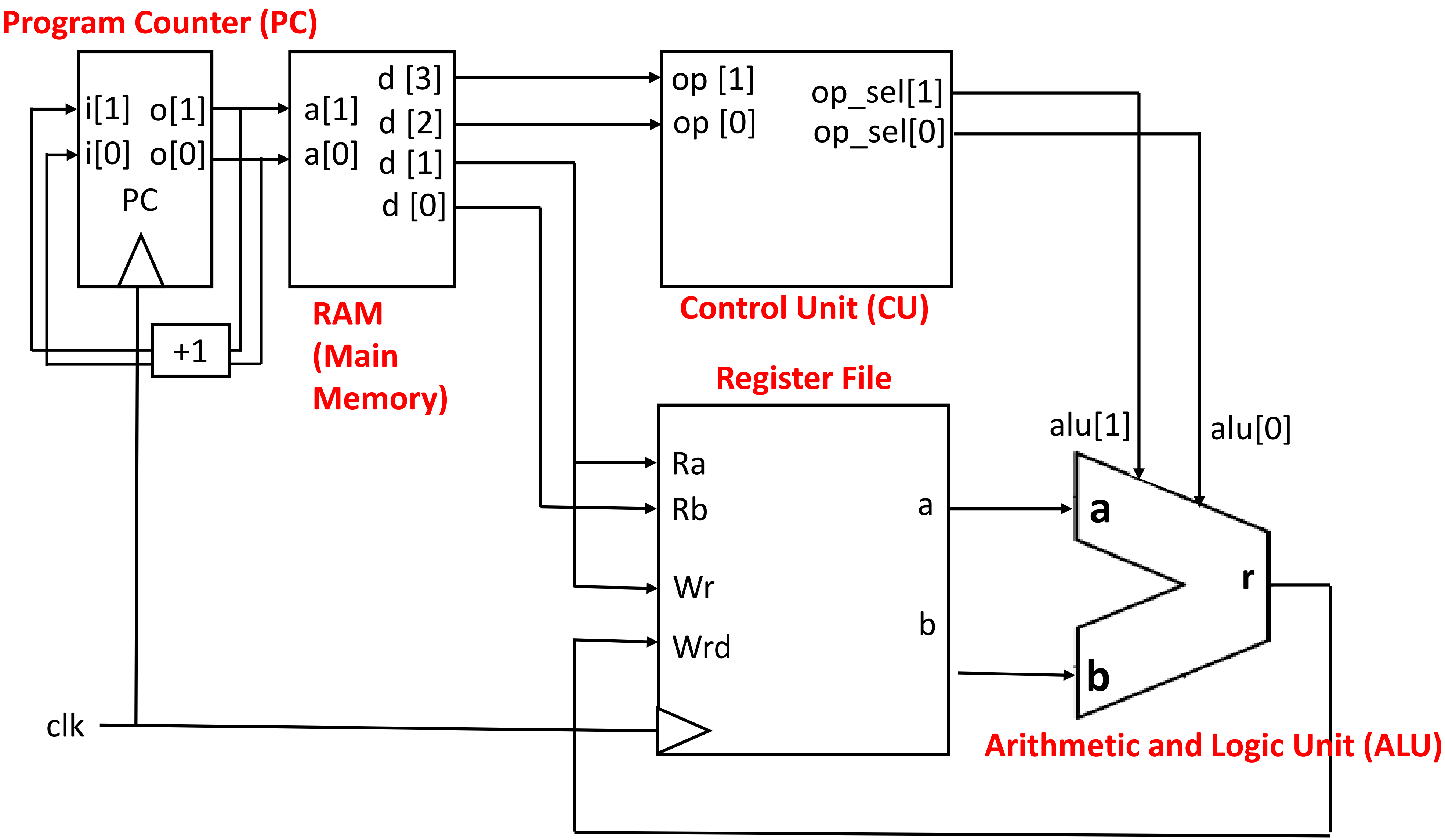
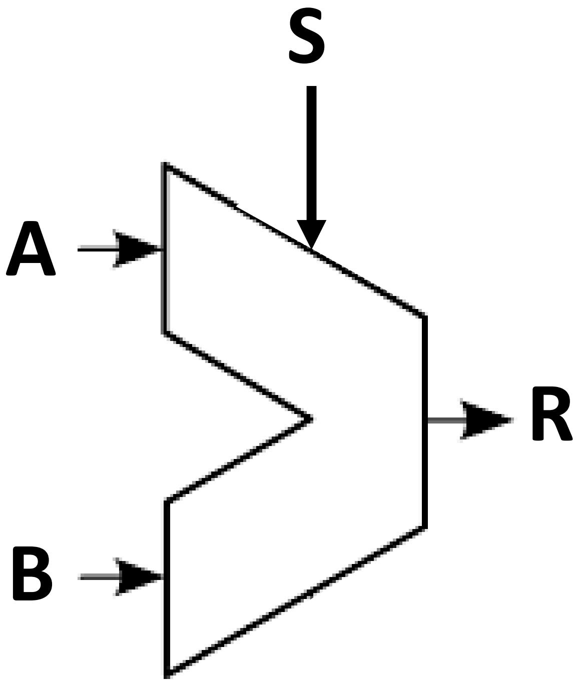


Figure: 1-bit CPU

ALU Circuit



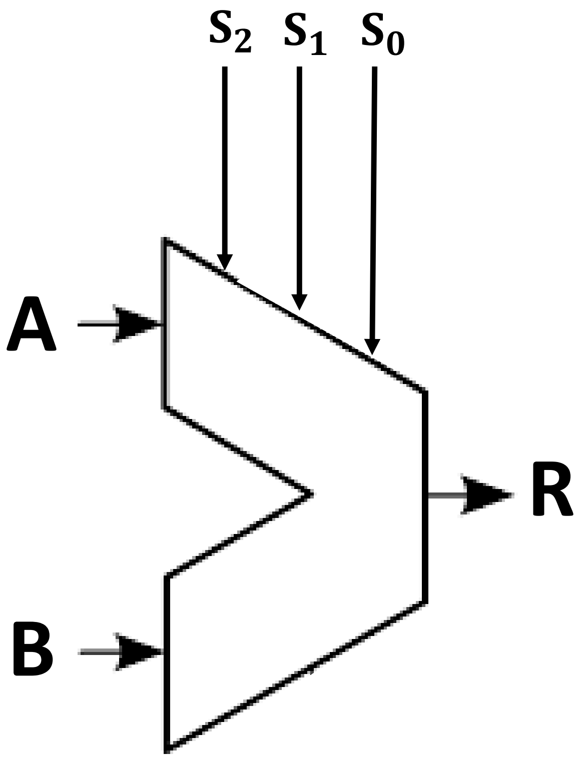
Available Operations:
ADD,
SUB,
MUL,
AND,
OR,
XOR,
LEFT SHIFT,
RIGHT SHIFT

All operations will
be executed at the same.

Only one operation must be
selected by Control Unit.

Total Operations: 8

1-bit ALU Circuit



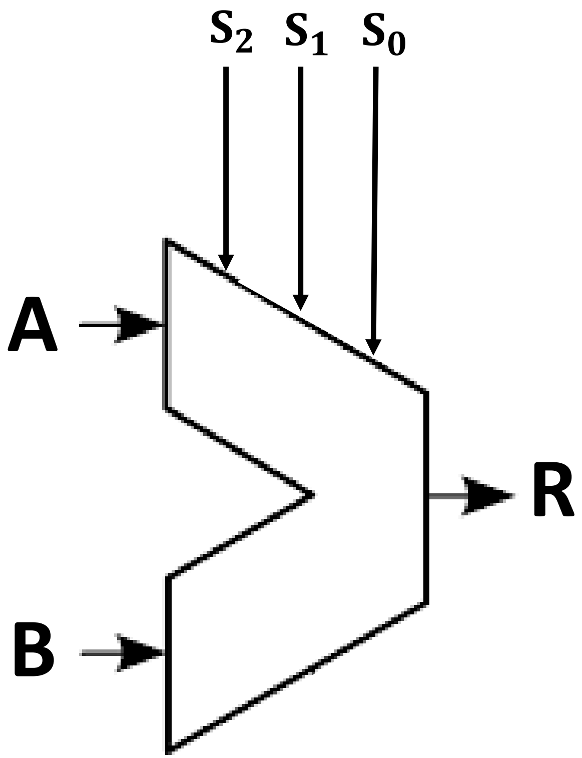
Available Operations:

ADD,
SUB,
MUL,
AND,
OR,
XOR,
LEFT SHIFT,
RIGHT SHIFT

Total Operations: 8

So, selection line must support at least 8 combinations.

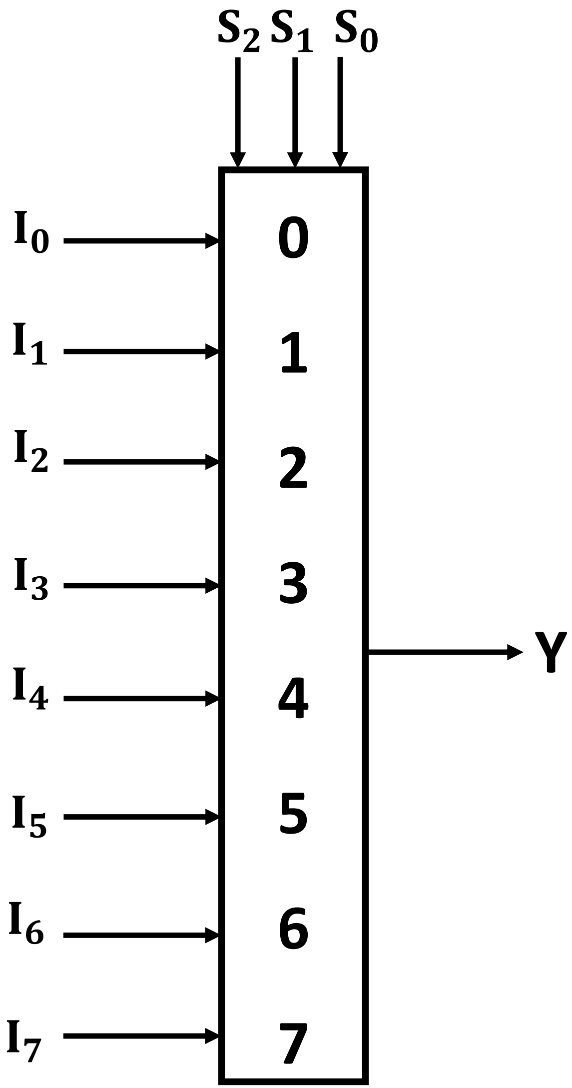
1-bit ALU Circuit



We will need a MUX to select our expected output.

As there are eight operations in total, we will use 8 to 1 MUX

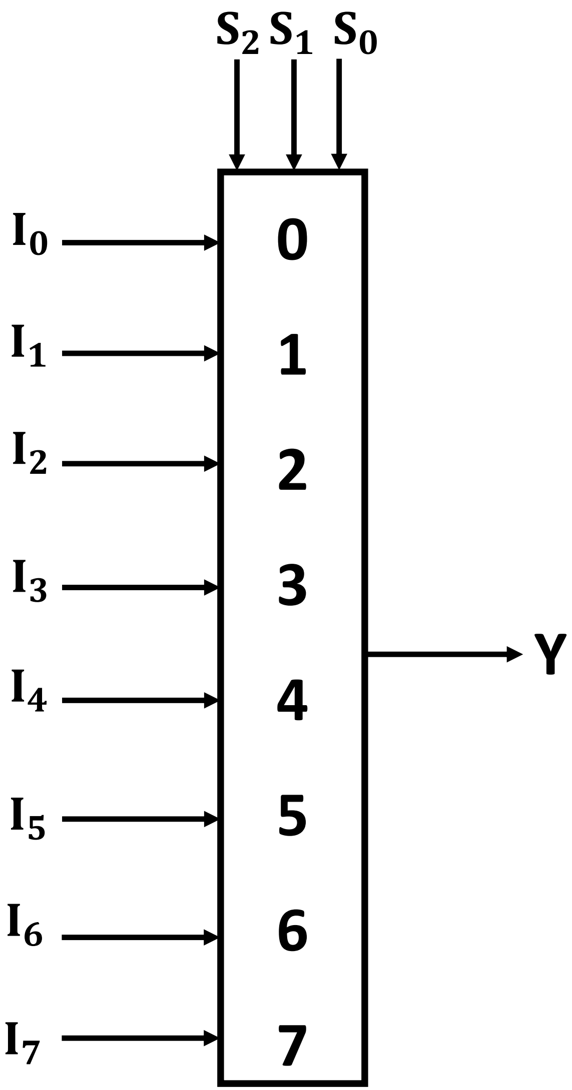
1-bit ALU Circuit
8 to 1 MUX



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

$$Y = \overline{S_2}.\overline{S_1}.\overline{S_0}.I_0 + \overline{S_2}.\overline{S_1}.S_0.I_1 + \overline{S_2}.S_1.\overline{S_0}.I_2 + \overline{S_2}.S_1.S_0.I_3 + S_2.\overline{S_1}.\overline{S_0}.I_4 + S_2.\overline{S_1}.S_0.I_5 + S_2.S_1.\overline{S_0}.I_6 + S_2.S_1.S_0.I_7$$

1-bit ALU Circuit

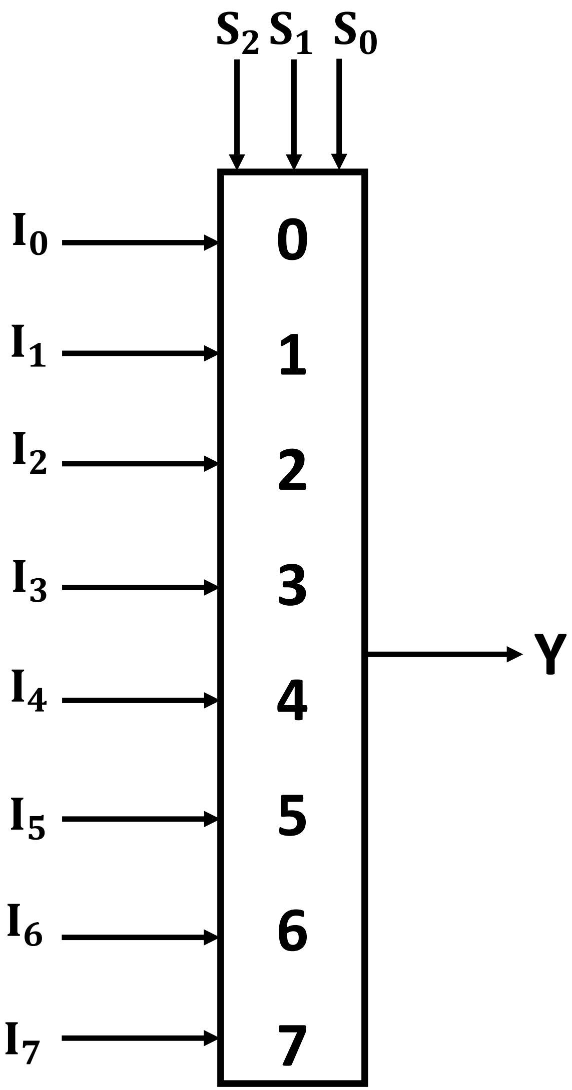


Operation	S_2	S_1	S_0	O
AND	0	0	0	I_0
OR	0	0	1	I_1
XOR	0	1	0	I_2
LEFT SHIFT	0	1	1	I_3
RIGHT SHIFT	1	0	0	I_4
ADD	1	0	1	I_5
SUB	1	1	0	I_6
MUL	1	1	1	I_7

If $\mathbf{sel} = \mathbf{0}, \ \mathbf{S} = \mathbf{ADD}$
If $\mathbf{sel} = \mathbf{1}, \ \mathbf{S} = \mathbf{SUB}$

If $\mathbf{S_1} = \mathbf{0}, \mathbf{S_0} = \mathbf{1}, \mathbf{S} = \mathbf{LEFT\ SHIFT}$
If $\mathbf{S_1} = \mathbf{1}, \mathbf{S_0} = \mathbf{0}, \mathbf{S} = \mathbf{RIGHT\ SHIFT}$

1-bit ALU Circuit

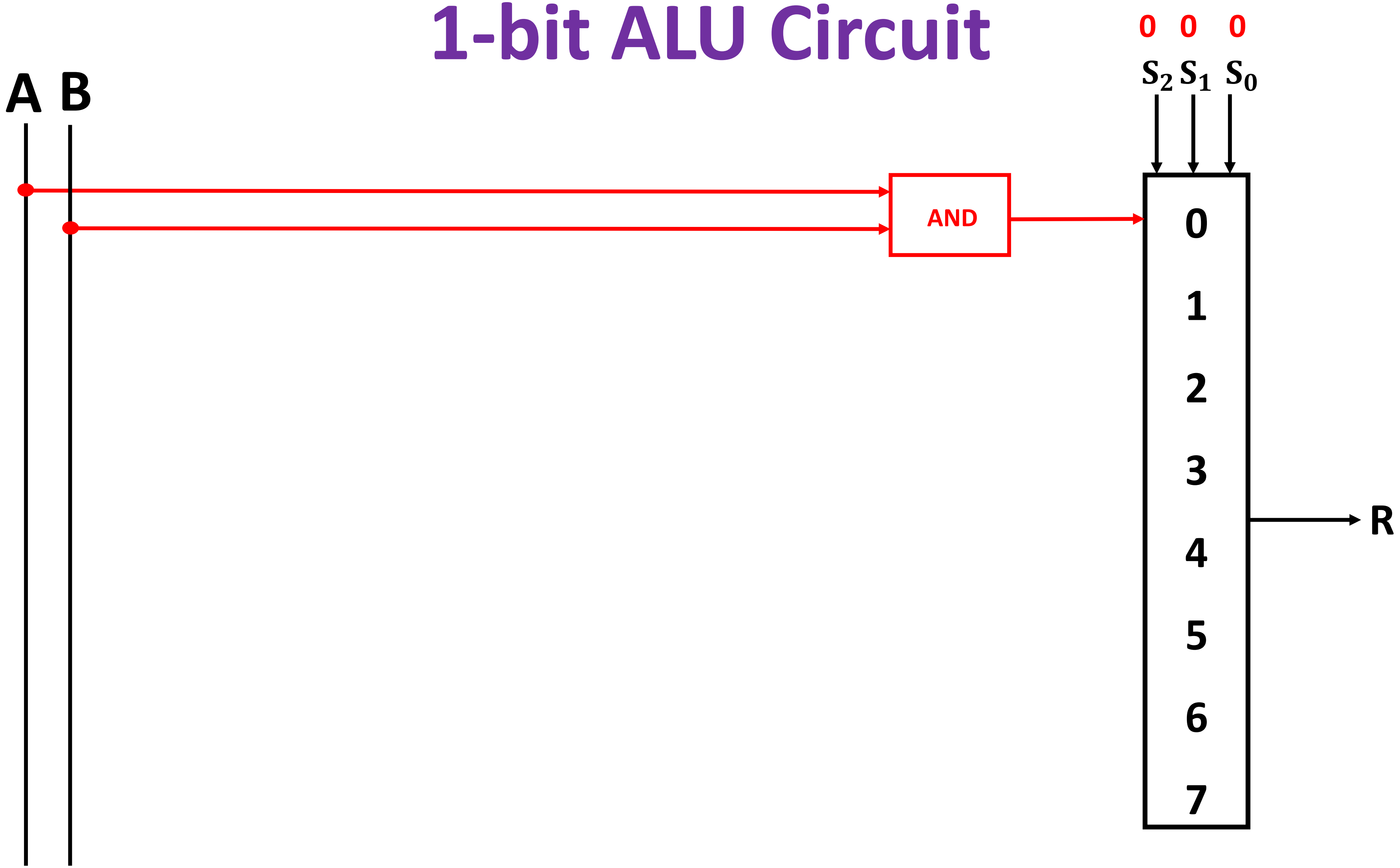


Operation	S_2	S_1	S_0	O
AND	0	0	0	I_0
OR	0	0	1	I_1
XOR	0	1	0	I_2
LEFT SHIFT	0	1	1	I_3
RIGHT SHIFT	1	0	0	I_4
ADD	1	0	1	I_5
SUB	1	1	0	I_6
MUL	1	1	1	I_7

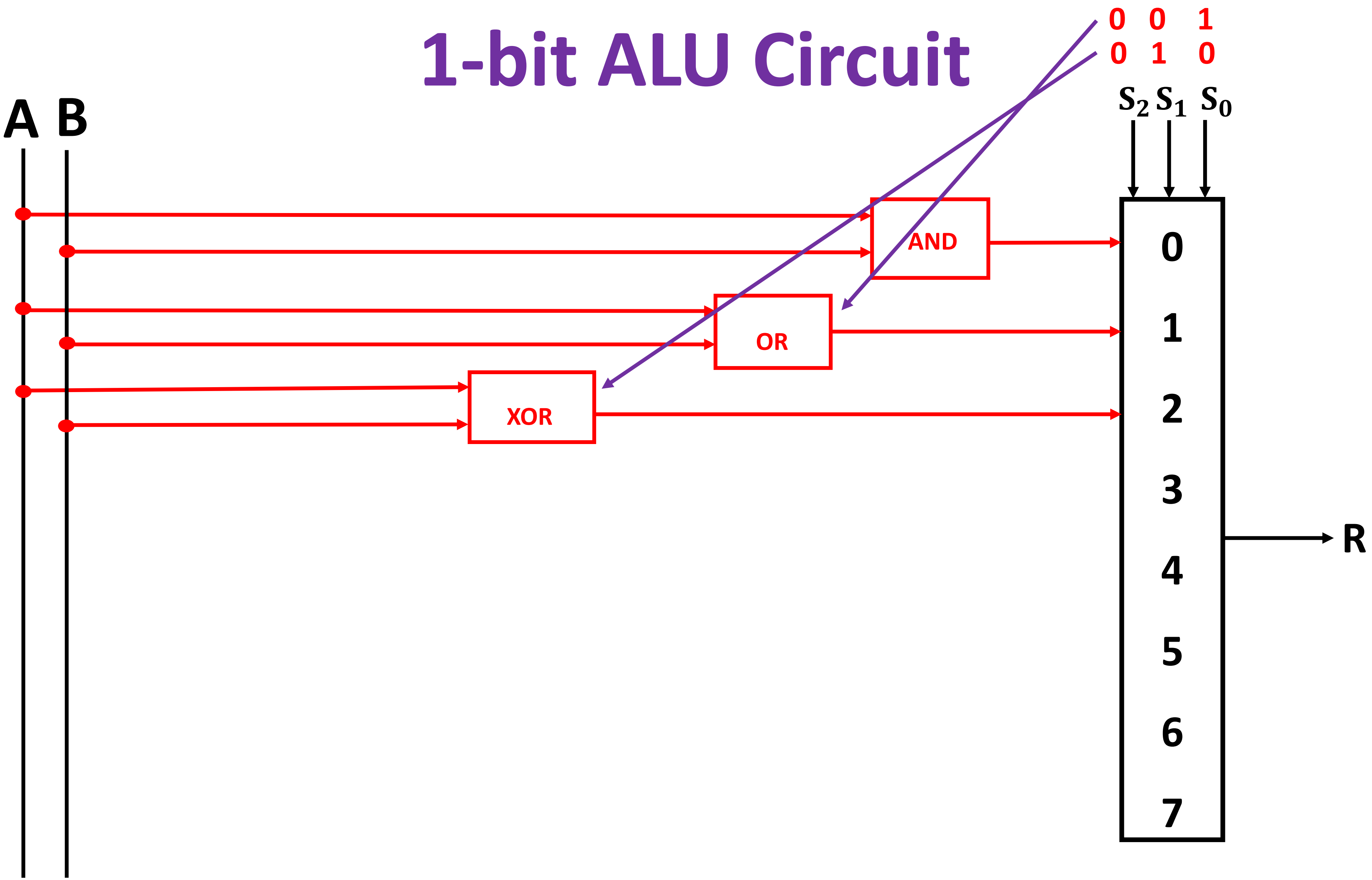
If $\mathrm{sel} = 0,\ S = ADD$
If $\mathrm{sel} = 1,\ S = SUB$

If $S_1 = 0, S_0 = 1, S = \text{LEFT SHIFT}$
If $S_1 = 1, S_0 = 0, S = \text{RIGHT SHIFT}$

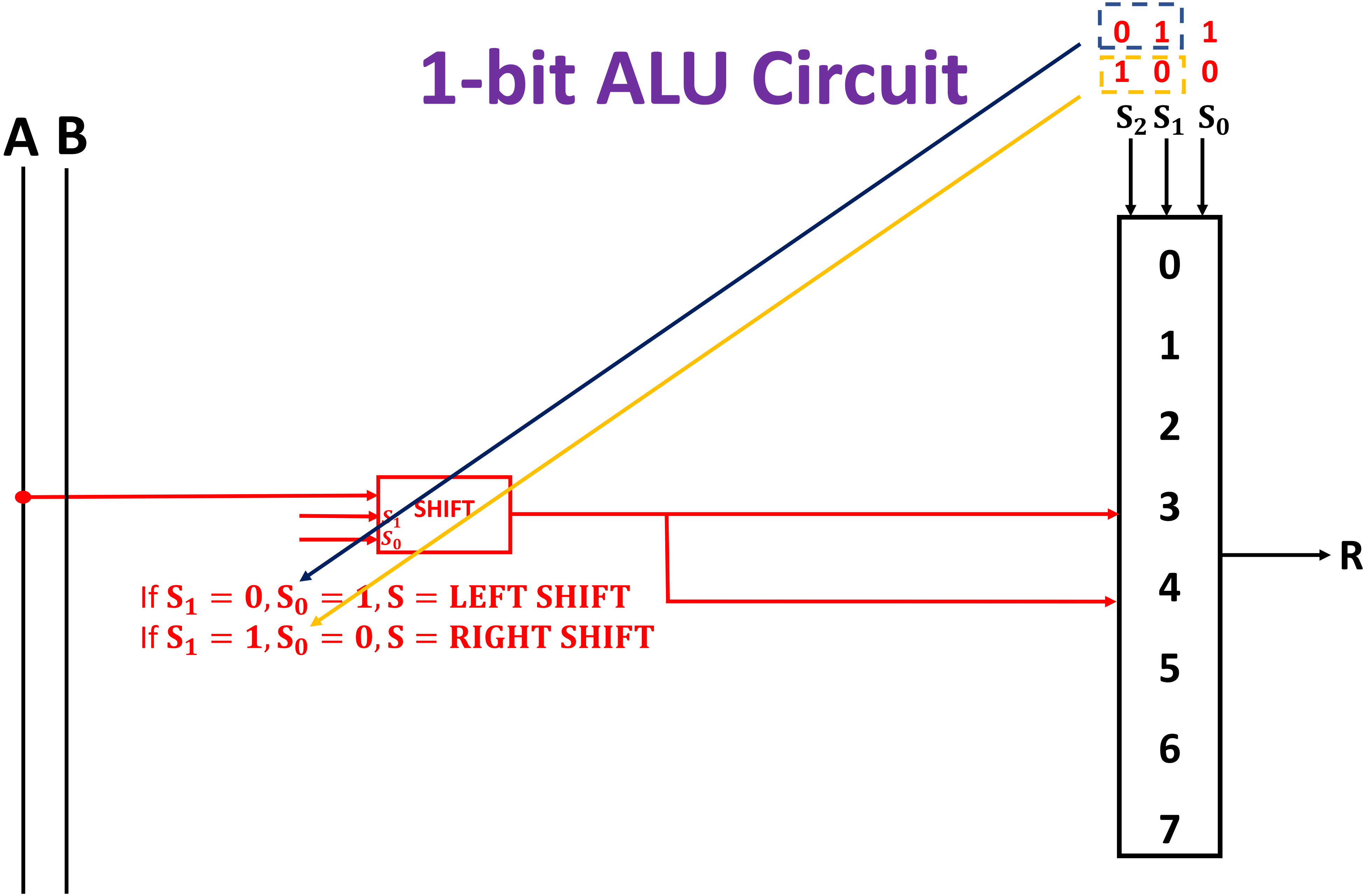
1-bit ALU Circuit



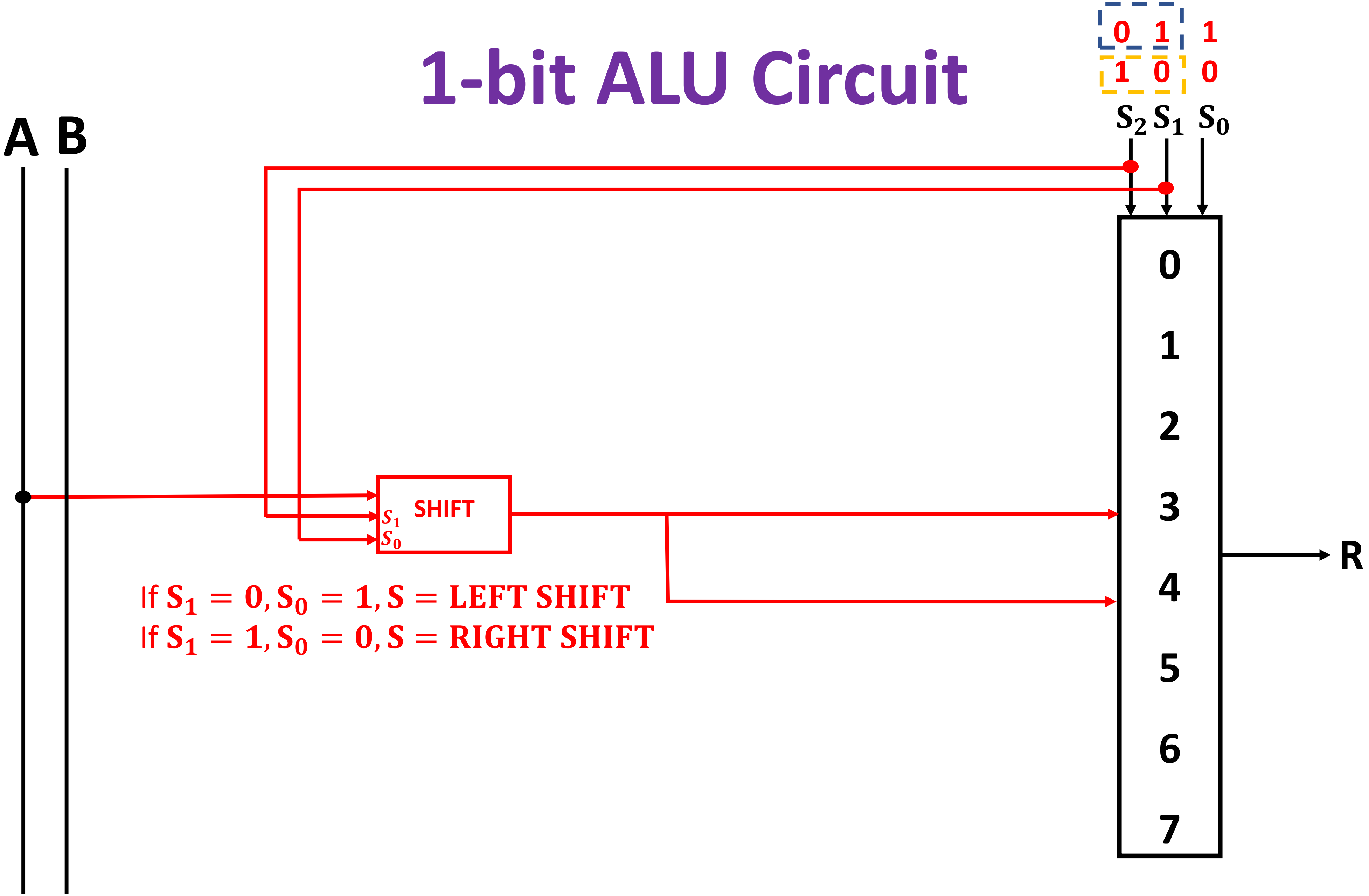
1-bit ALU Circuit



1-bit ALU Circuit



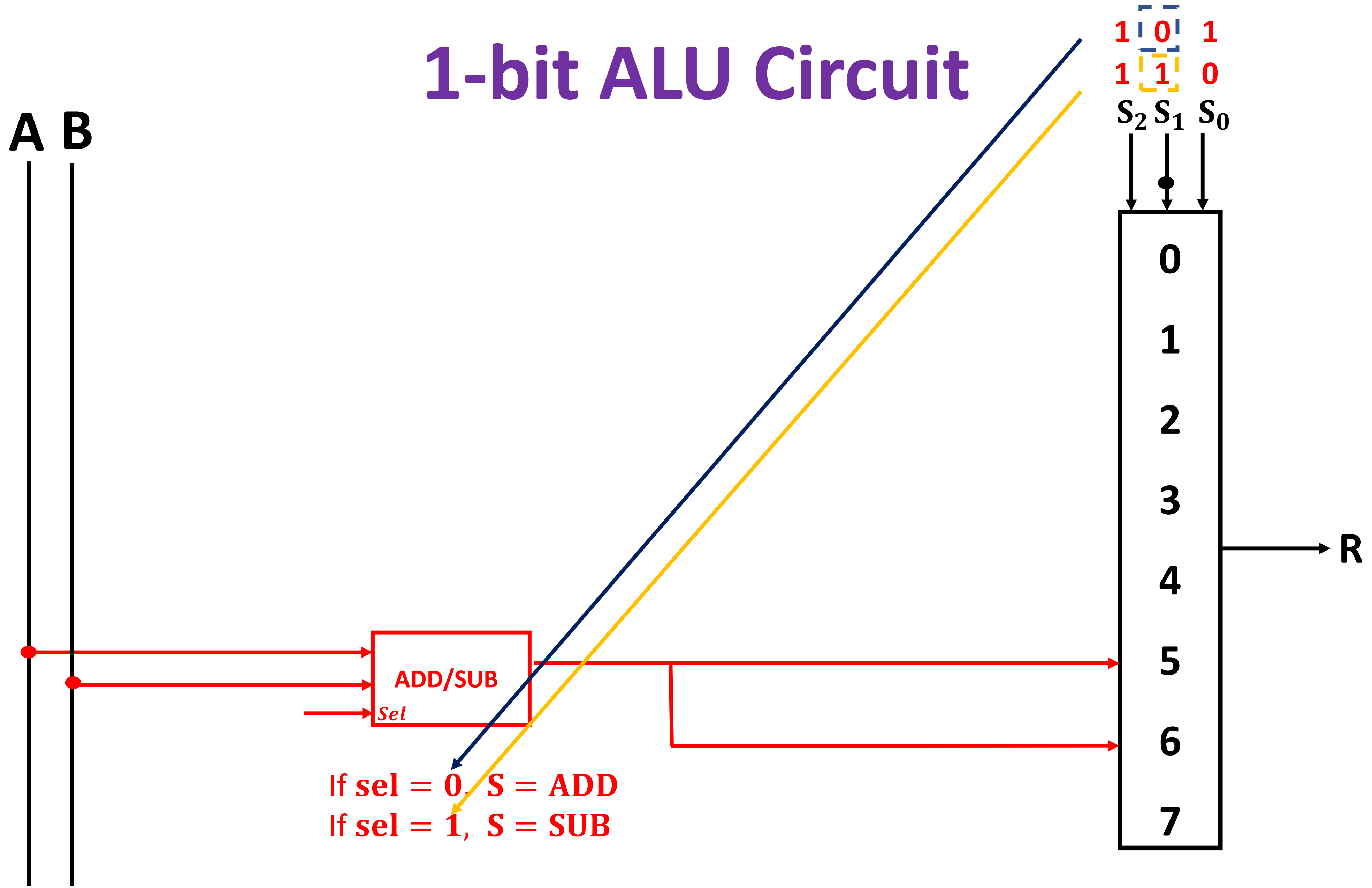
1-bit ALU Circuit



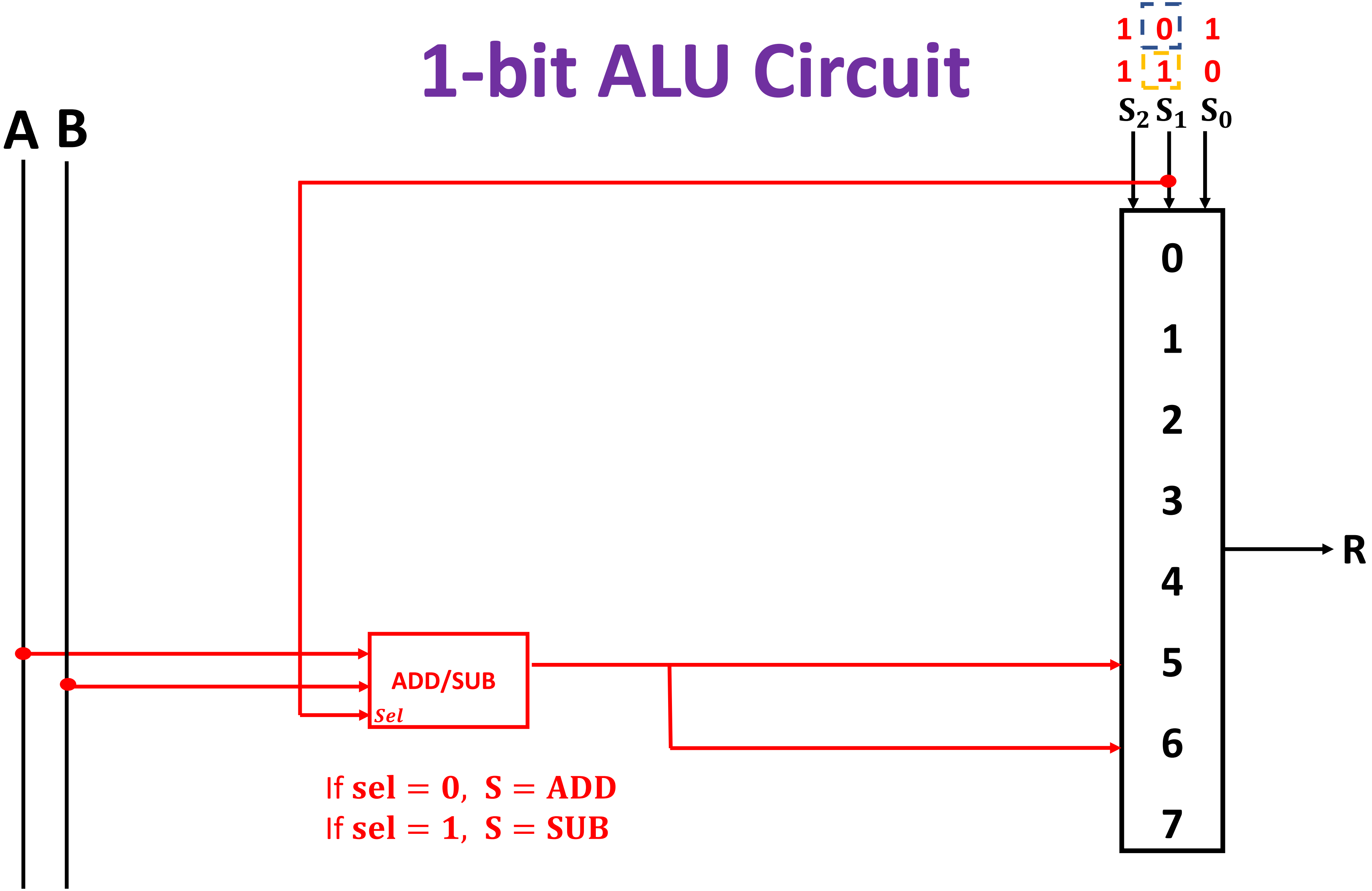
If $S_1 = 0, S_0 = 1, S = \text{LEFT SHIFT}$

If $S_1 = 1, S_0 = 0, S = \text{RIGHT SHIFT}$

1-bit ALU Circuit

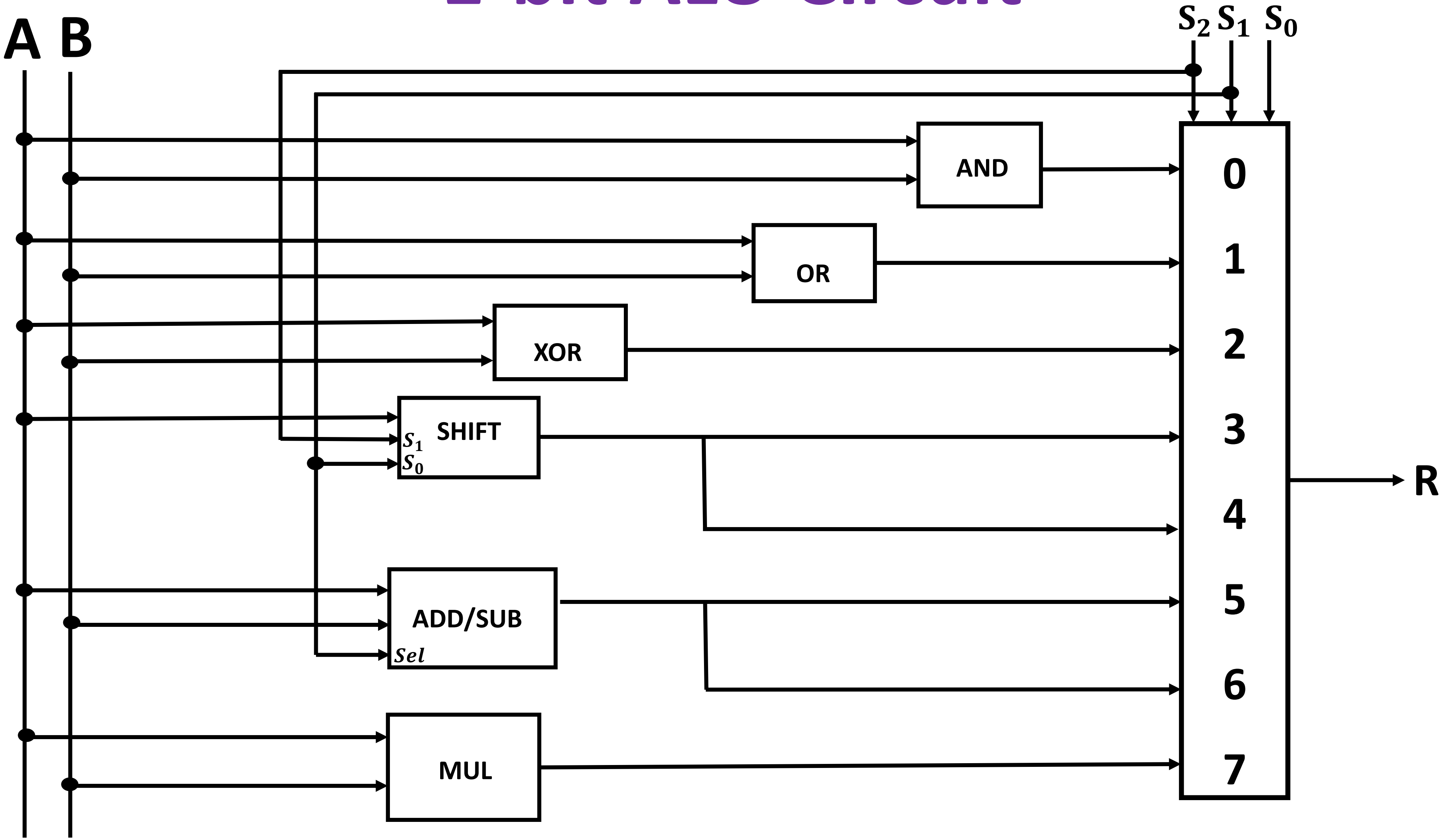


1-bit ALU Circuit



If $\mathbf{sel = 0,\ S = ADD}$
If $\mathbf{sel = 1,\ S = SUB}$

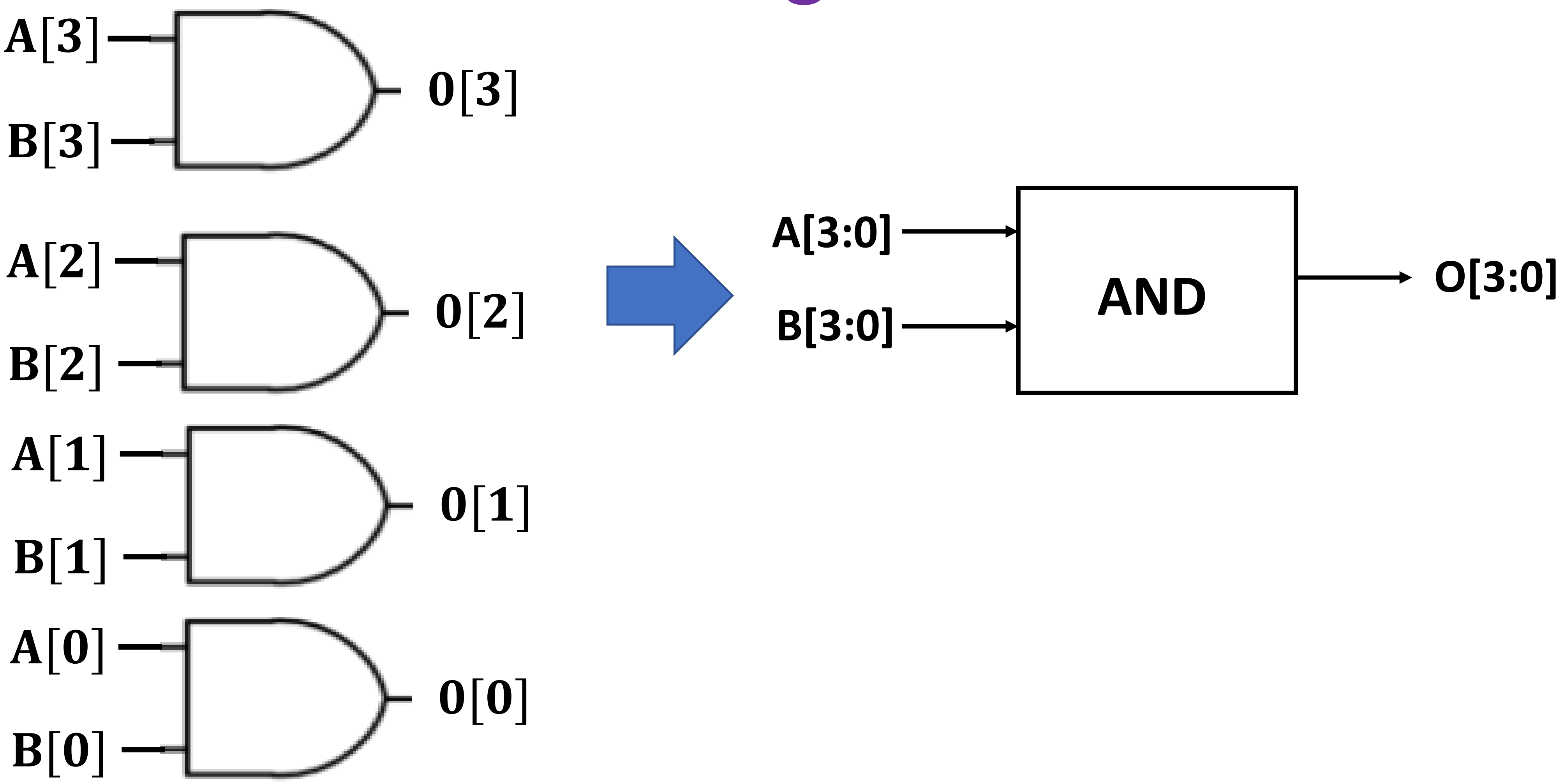
1-bit ALU Circuit



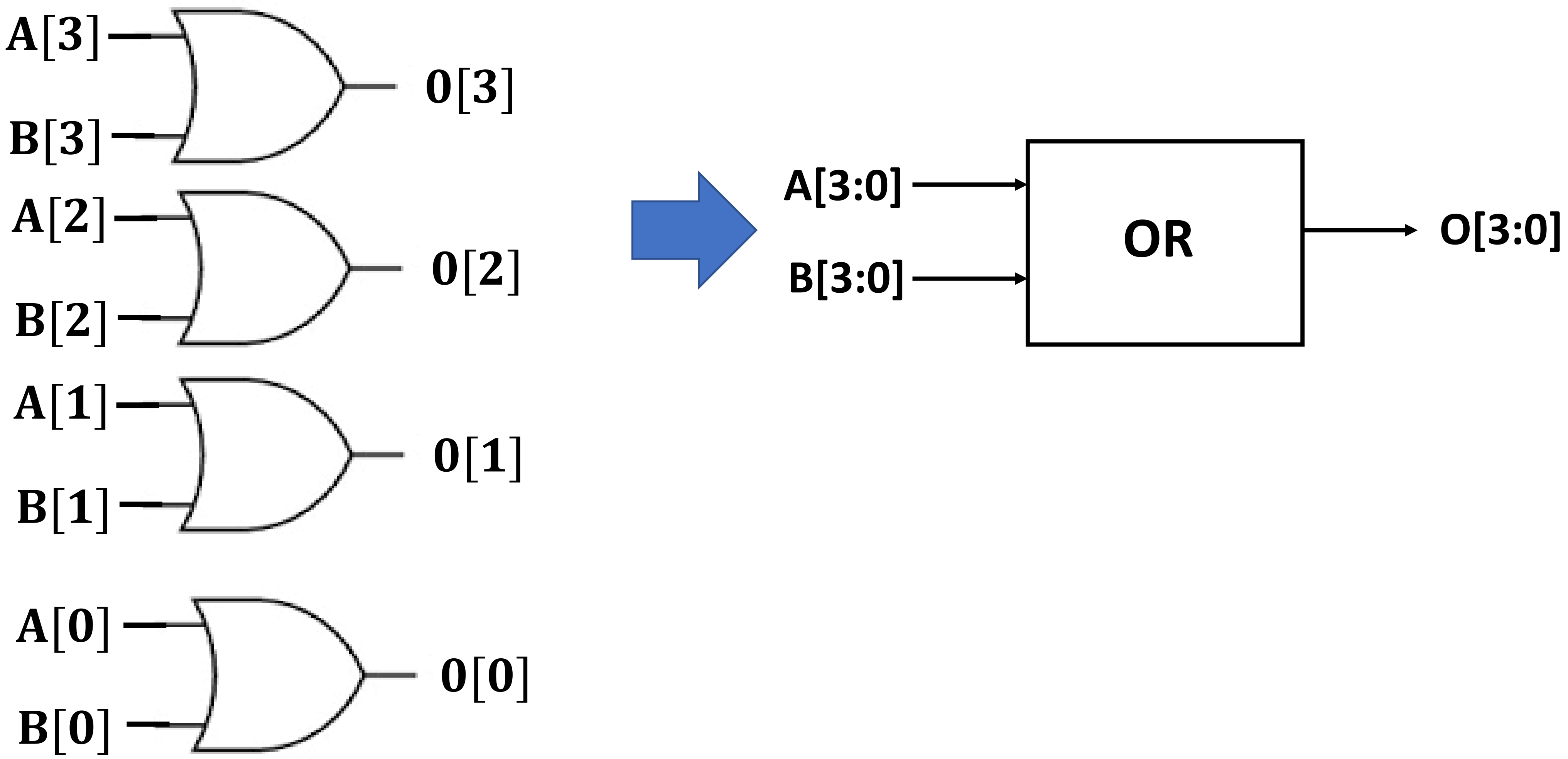
4-bit ALU design

s_1

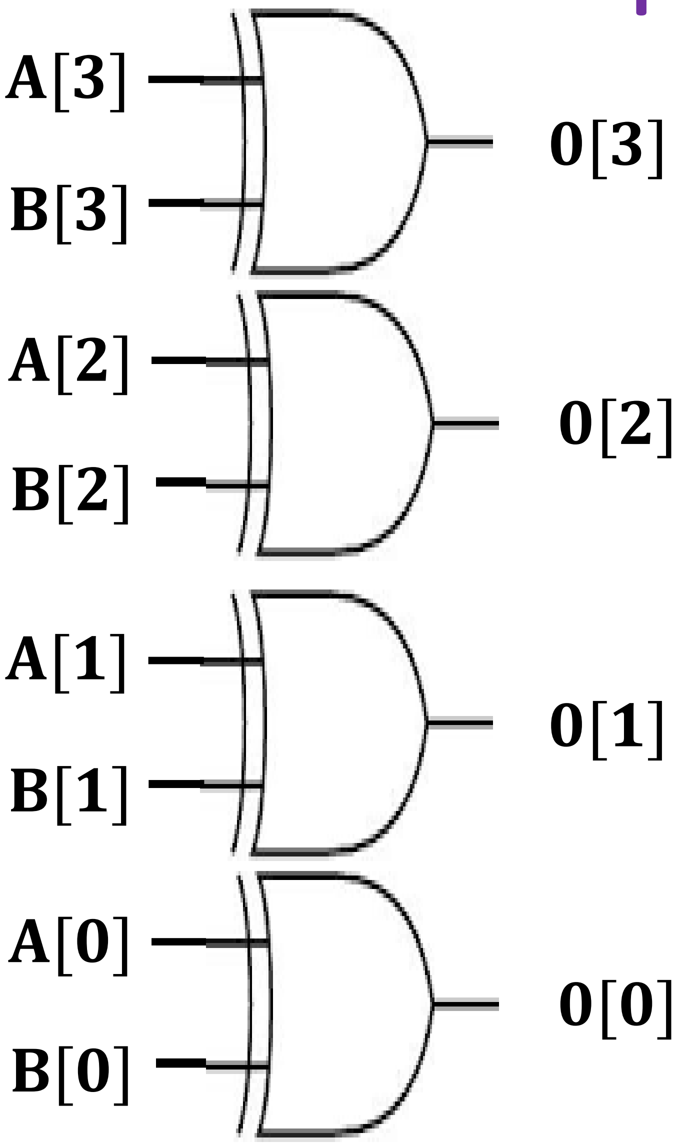
4-bit AND gate

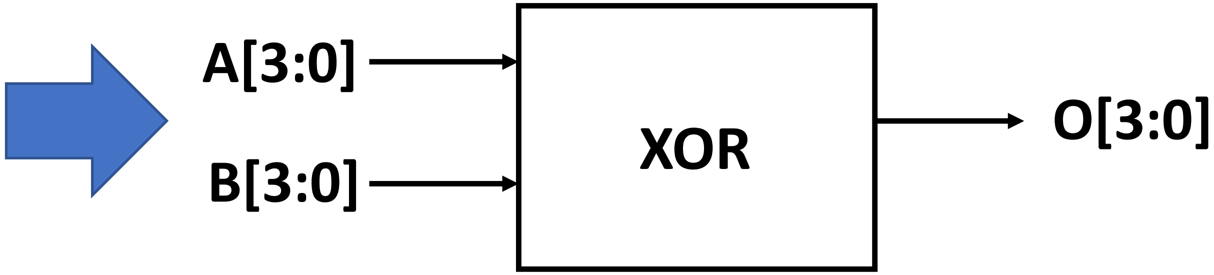


4-bit OR gate

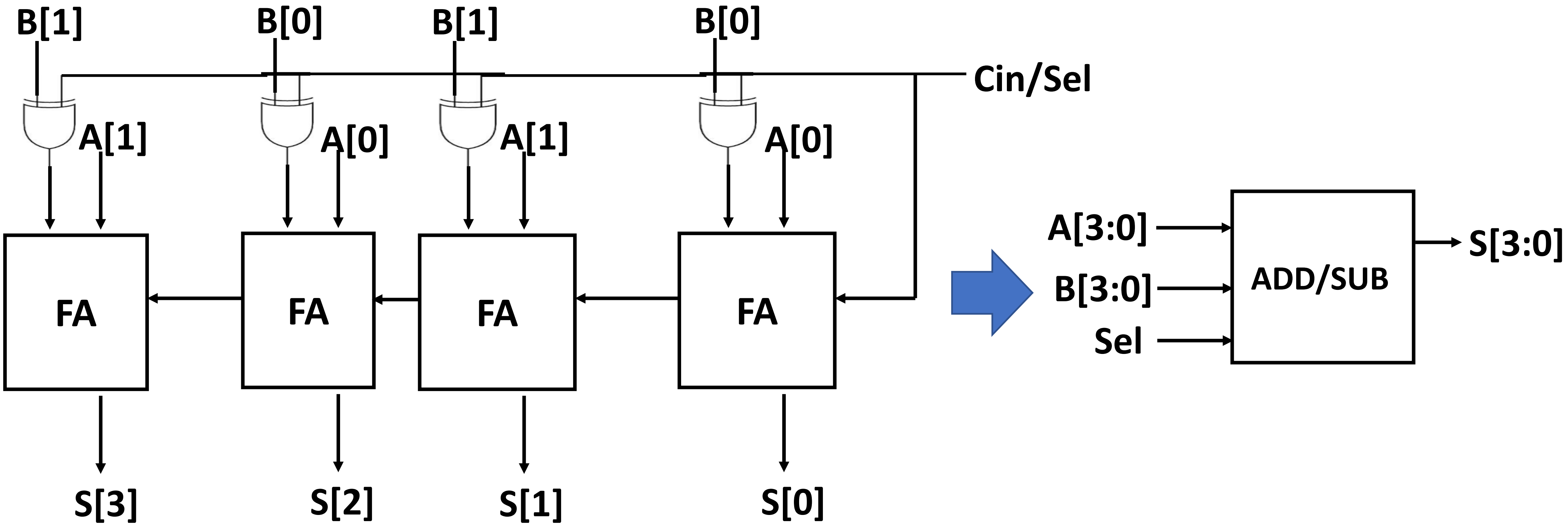


4-bit OR gate



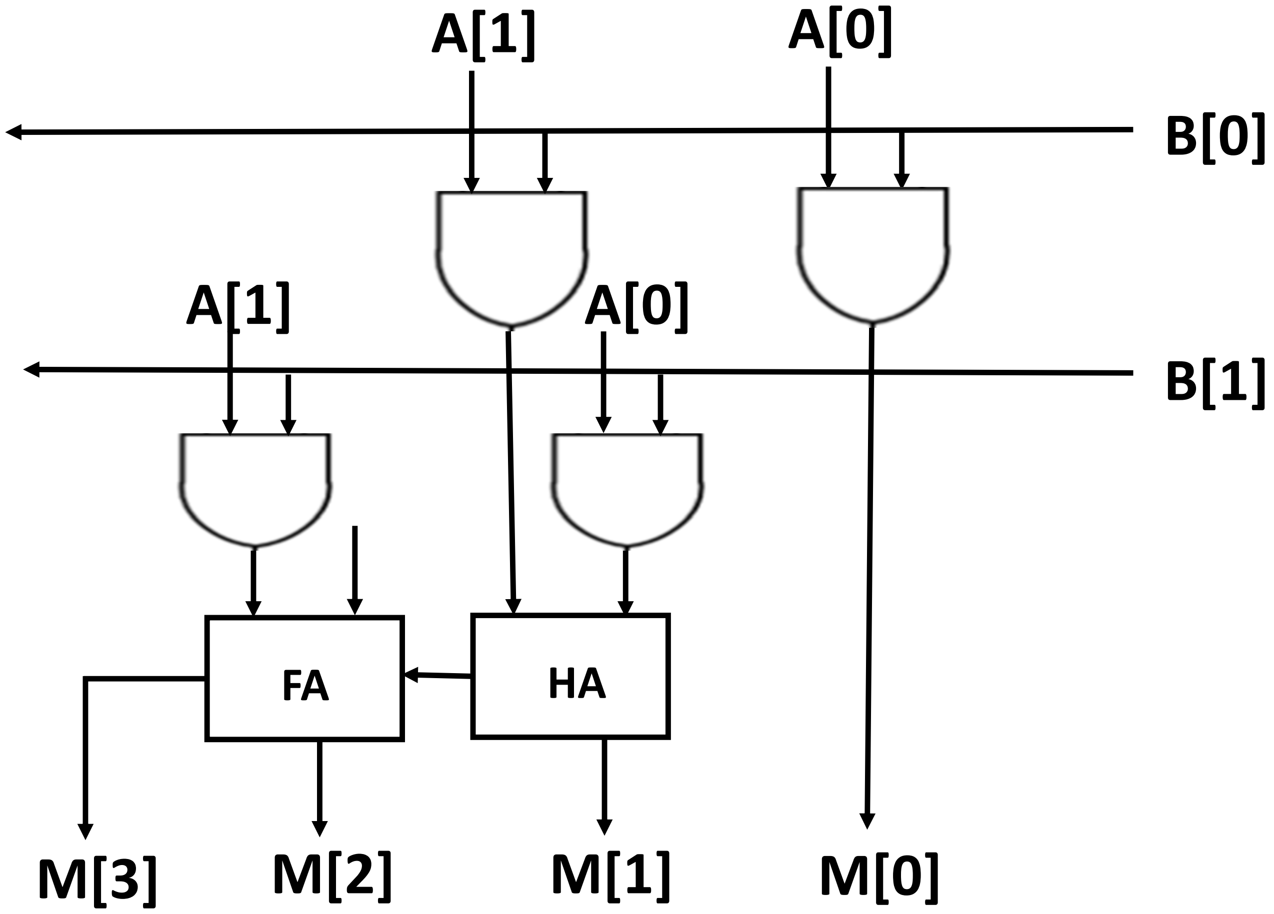


2-bit Adder/Subtractor



4-bit Multiplier

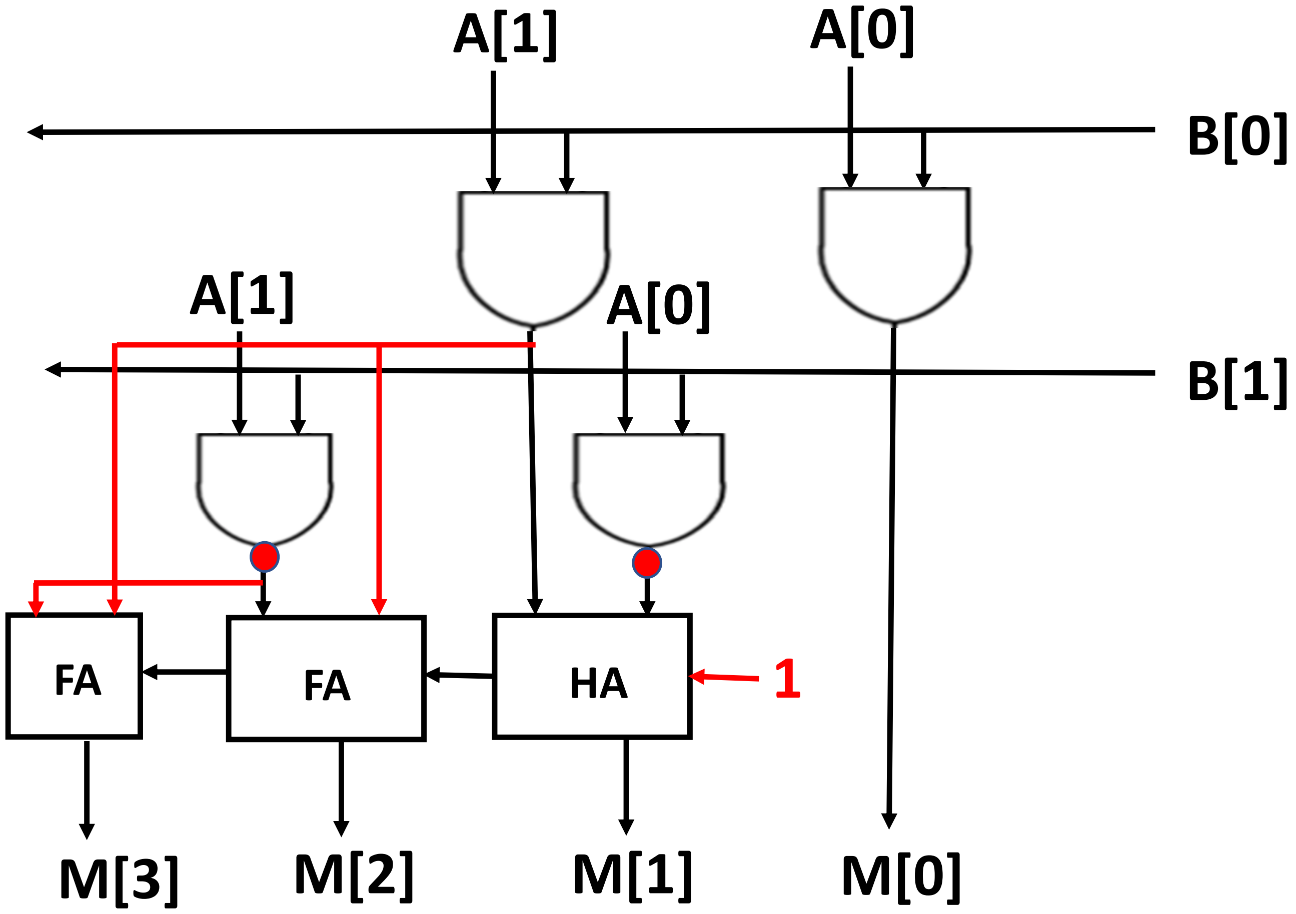
$$\begin{array}{cccc} & & A_1 & A_0 \\ & & B_1 & B_0 \\ \hline & & A_1B_0 & A_0B_0 \\ & A_1B_1 & A_0B_1 & \\ \hline M_3 & M_2 & M_1 & M_0 \end{array}$$



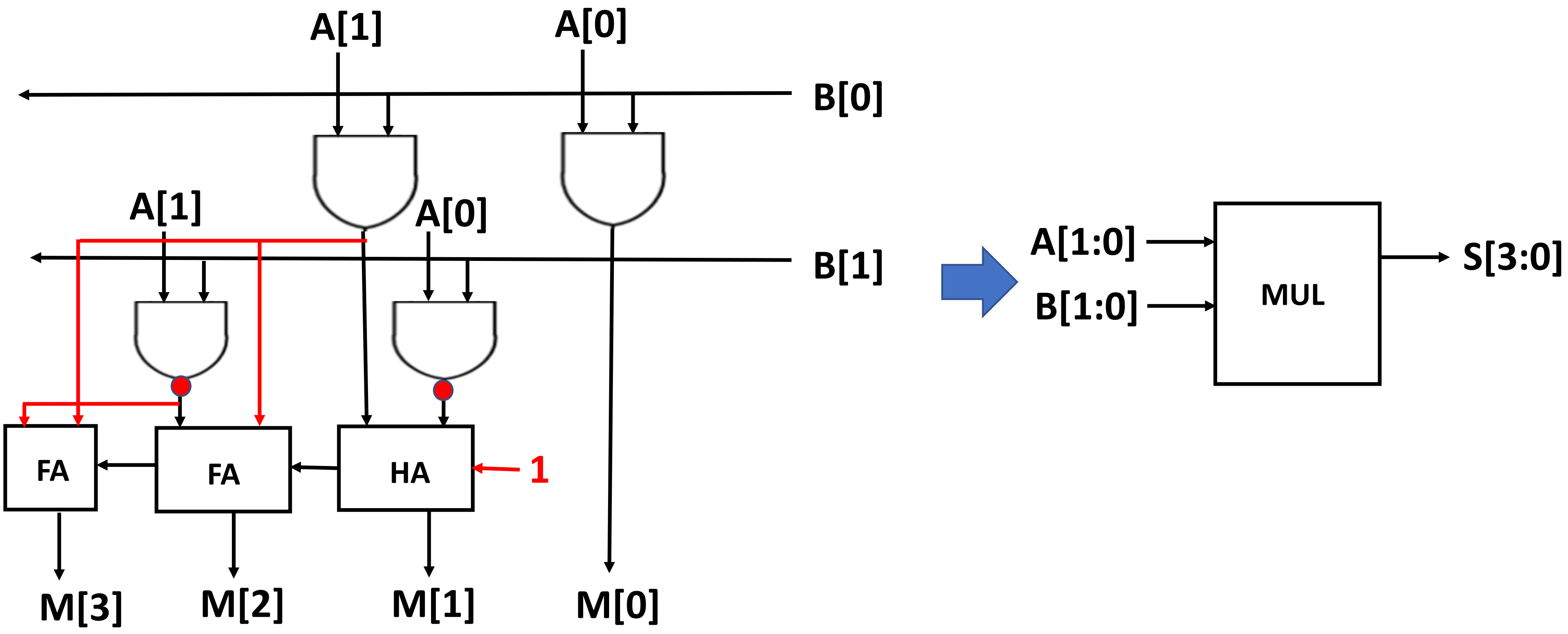
4-bit Multiplier (Signed Multiplier)

		A_1	A_0
		B_1	B_0
		A_1B_0	A_0B_0
	A_1B_1	A_0B_1	
M_3	M_2	M_1	M_0

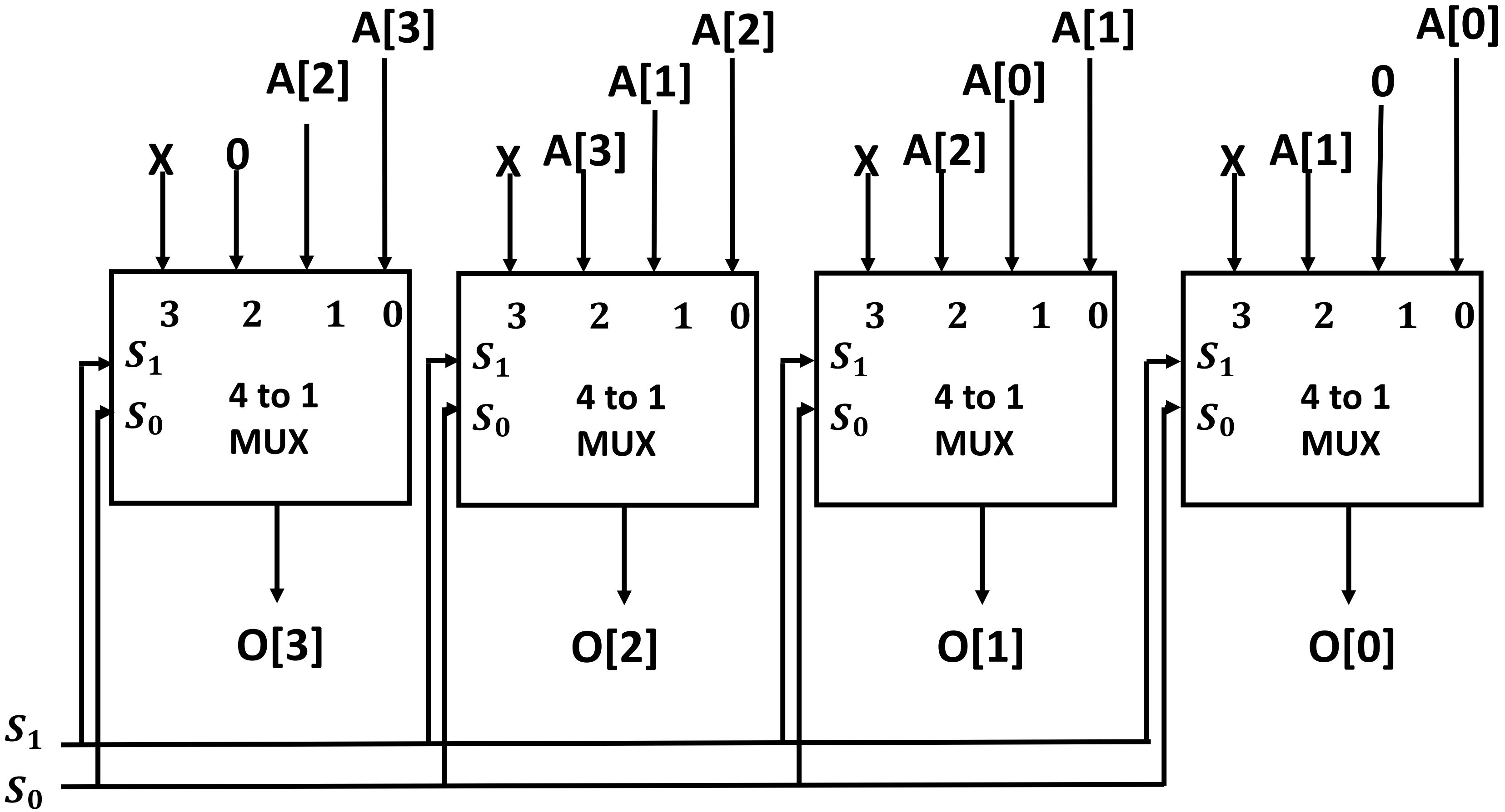
		A_1	A_0
		B_1	B_0
$+\ A_1B_0$	A_1B_0	A_1B_0	A_0B_0
$-\ A_1B_1$	A_1B_1	A_0B_1	
M_3	M_2	M_1	M_0



4-bit Multiplier (Signed Multiplier)



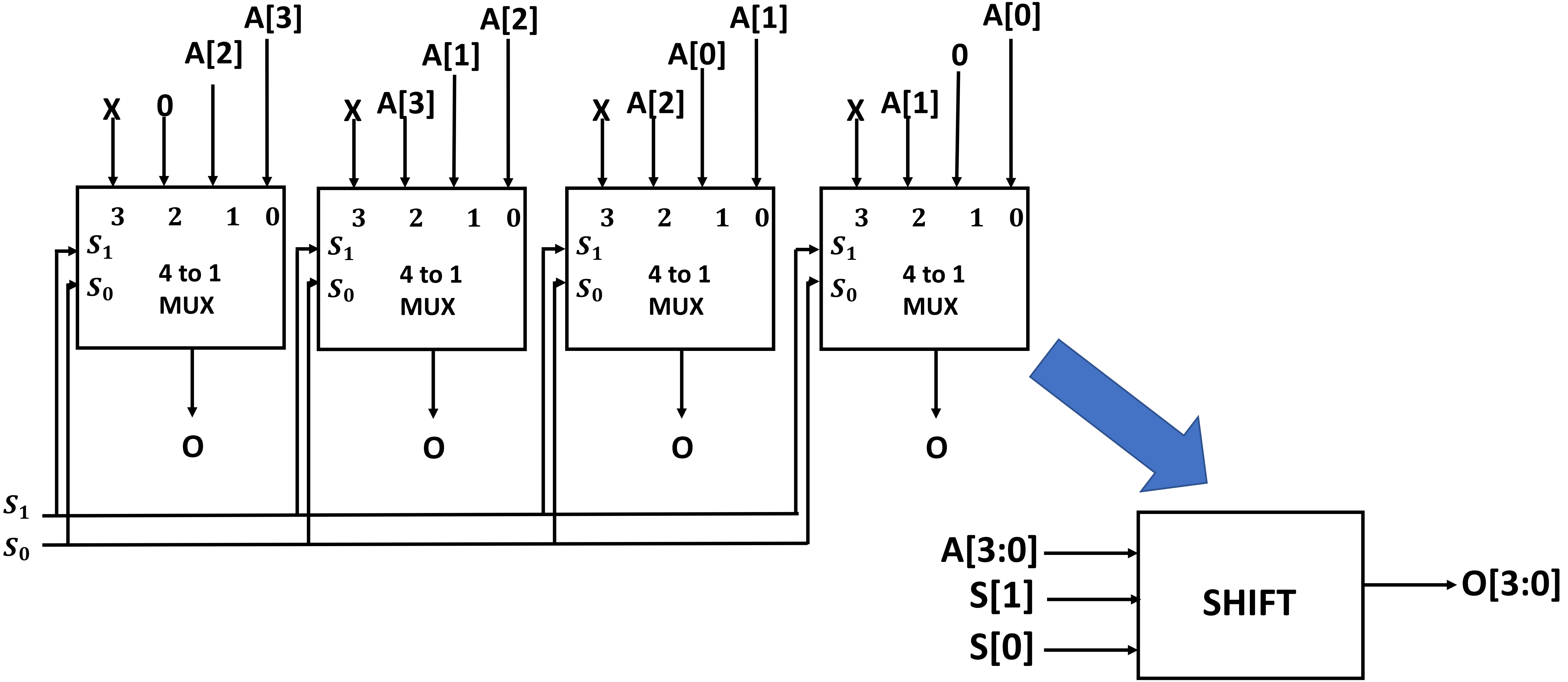
4-bit Shifter



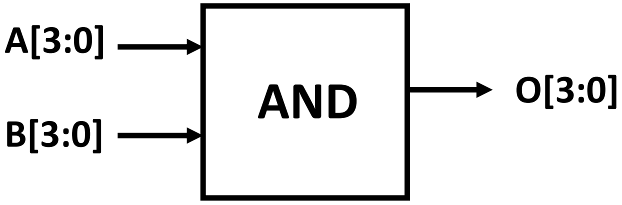
S_1	S_2	Output	Operation
0	0	**A[3]A[2]A[1]A[0]**	No Shift
0	1	**A[2]A[1]A[0] 0**	Left Shift
1	0	**0 A[3]A[2]A[1]**	Right Shift
1	1	X	X

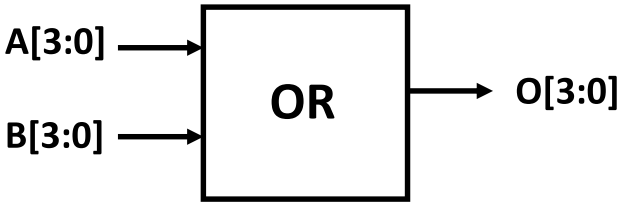
Input: A[3]A[2]A[1]A[0]
Right shift: 0 A[3]A[2]A[1]
Left shift: A[2]A[1]A[0] 0

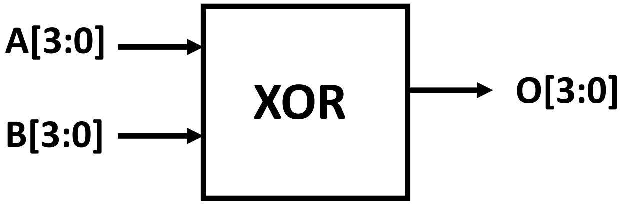
4-bit Shifter

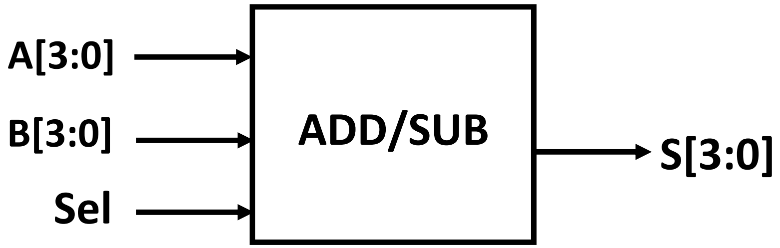


All the circuits so far

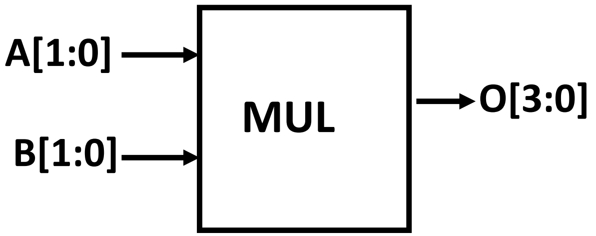


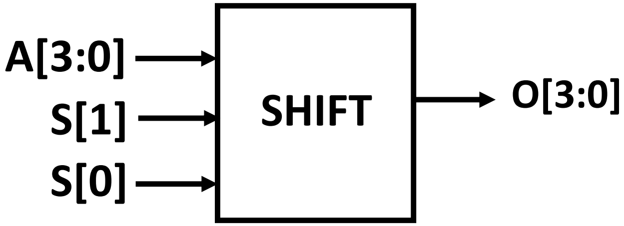






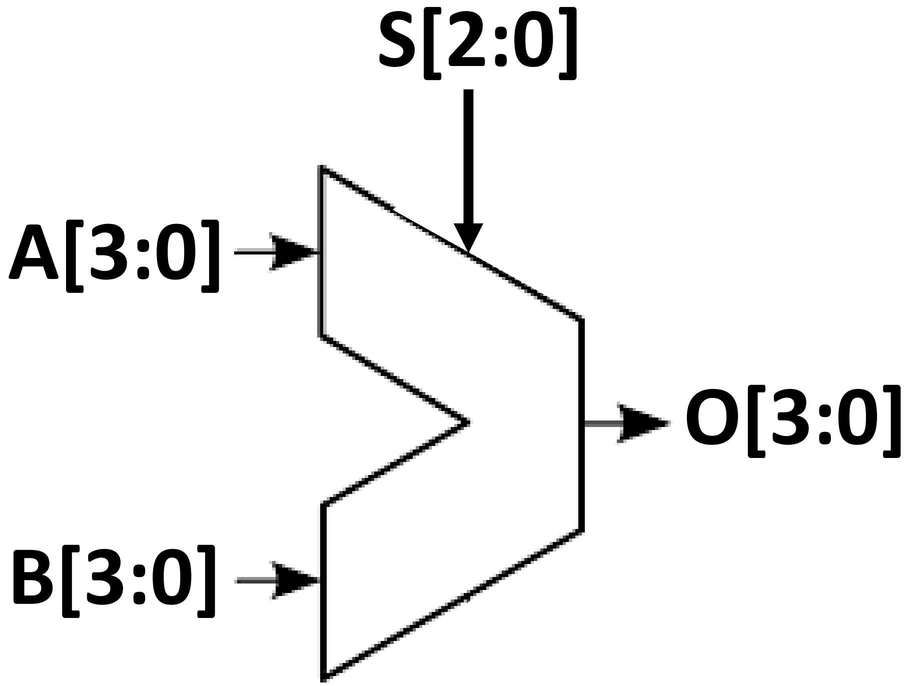
If $\mathbf{sel = 0,\ \ S = ADD}$
If $\mathbf{sel = 1,\ \ S = SUB}$



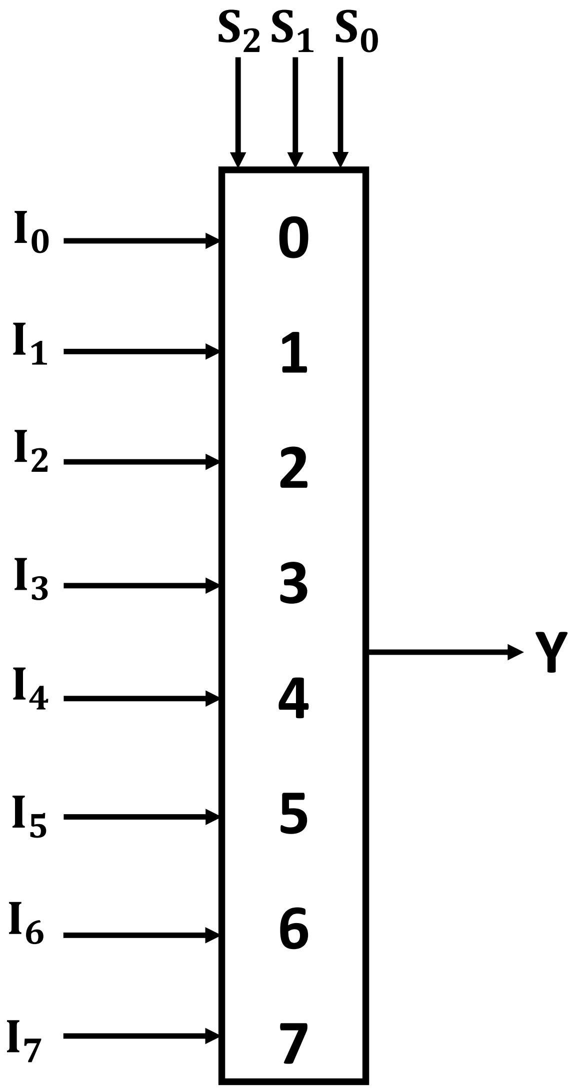


If $\mathbf{S_1 = 0, S_0 = 1, S = LEFT\ SHIFT}$
If $\mathbf{S_1 = 1, S_0 = 0, S = RIGHT\ SHIFT}$

4-bit ALU Circuit

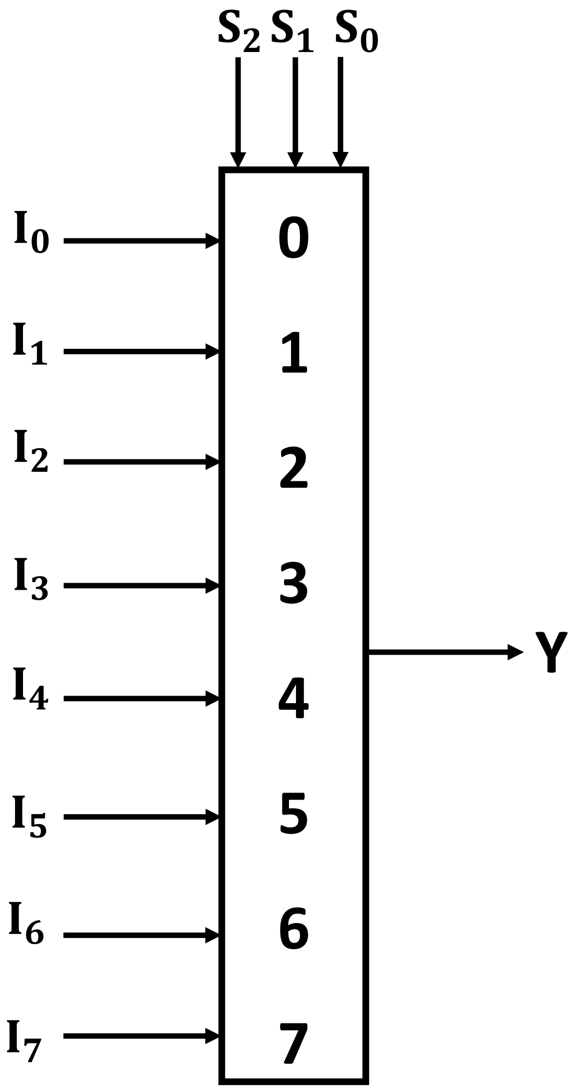


4-bit ALU Circuit
8 to 1 MUX



But this MUX can handle only 1 bit.

How can we build 4-bit MUX?



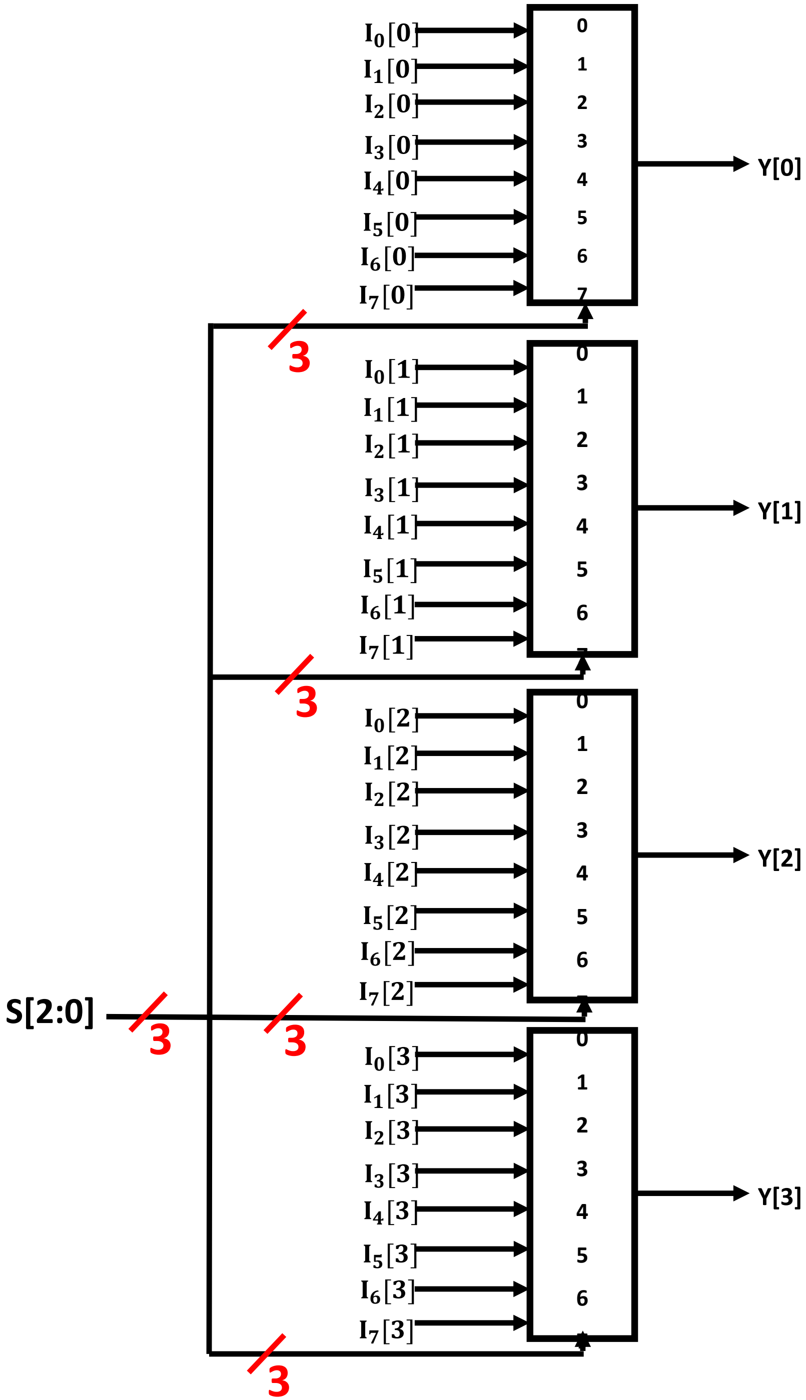
4-bit ALU Circuit

8 to 1 MUX

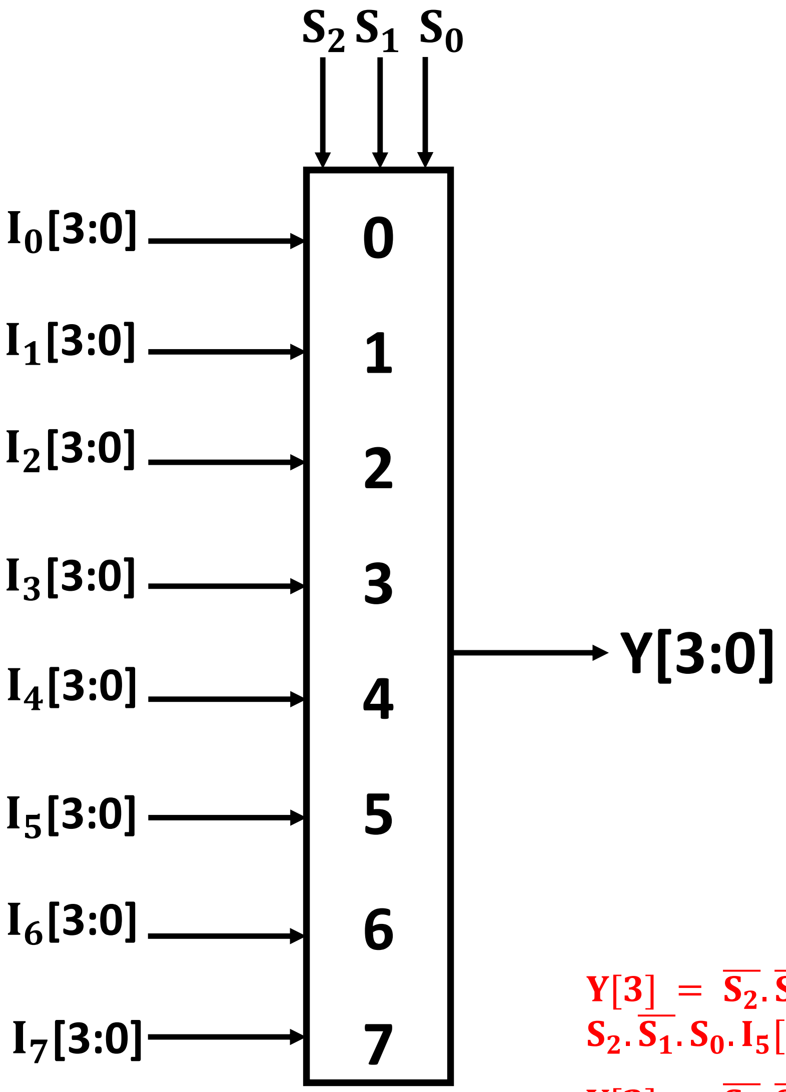
But this MUX can handle only 1 bit.
How can we build 4-bit MUX?

Solution is to use FOUR 8 to 1 MUX for 4 input lines and 4 output lines.

4-bit ALU Circuit
4-bit 8 to 1 MUX



4-bit ALU Circuit
8 to 1 MUX

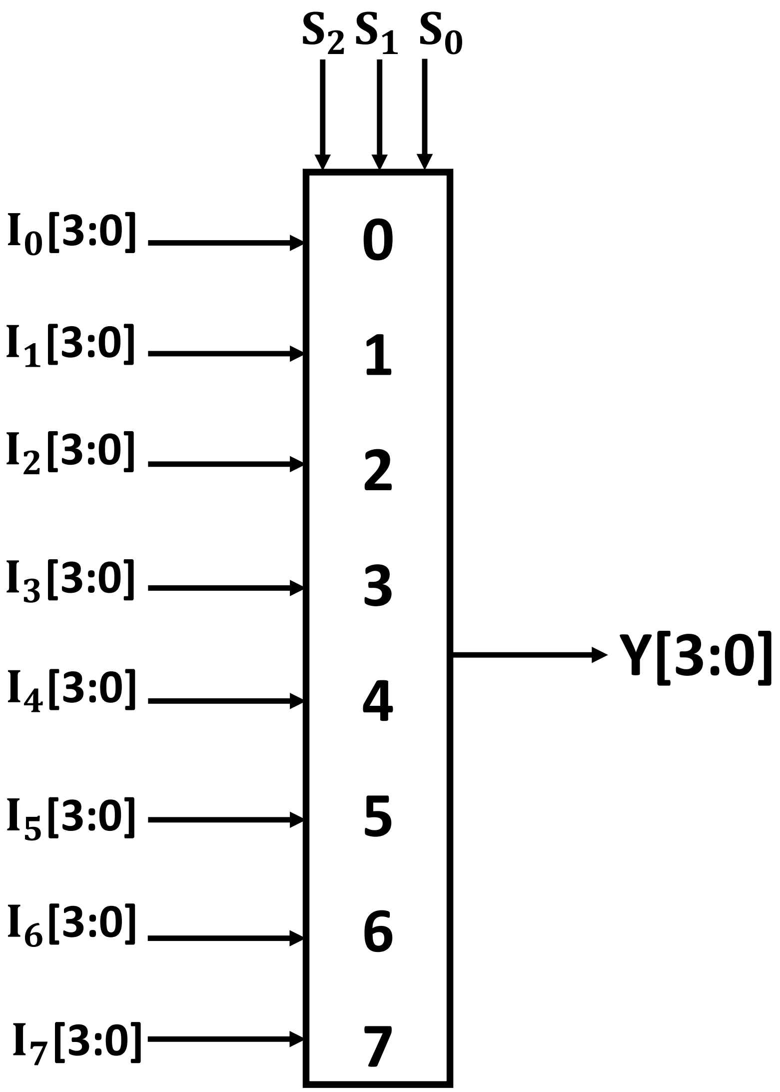


S_2	S_1	S_0	Y[3]	Y[2]	Y[1]	Y[0]
0	0	0	$I_0[3]$	$I_0[2]$	$I_0[1]$	$I_0[0]$
0	0	1	$I_1[3]$	$I_1[2]$	$I_1[1]$	$I_1[0]$
0	1	0	$I_2[3]$	$I_2[2]$	$I_2[1]$	$I_2[0]$
0	1	1	$I_3[3]$	$I_3[2]$	$I_3[1]$	$I_3[0]$
1	0	0	$I_4[3]$	$I_4[2]$	$I_4[1]$	$I_4[0]$
1	0	1	$I_5[3]$	$I_5[2]$	$I_5[1]$	$I_5[0]$
1	1	0	$I_6[3]$	$I_6[2]$	$I_6[1]$	$I_6[0]$
1	1	1	$I_7[3]$	$I_7[2]$	$I_7[1]$	$I_7[0]$

$$Y[3] = \overline{S_2}.\overline{S_1}.\overline{S_0}.I_0[3] + \overline{S_2}.\overline{S_1}.S_0.I_1[3] + \overline{S_2}.S_1.\overline{S_0}.I_2[3] + \overline{S_2}.S_1.S_0.I_3[3] + S_2.\overline{S_1}.\overline{S_0}.I_4[3] + S_2.\overline{S_1}.S_0.I_5[3] + S_2.S_1.\overline{S_0}.I_6[3] + S_2.S_1.S_0.I_7[3]$$

$$Y[2] = \overline{S_2}.\overline{S_1}.\overline{S_0}.I_0[2] + \overline{S_2}.\overline{S_1}.S_0.I_1[2] + \overline{S_2}.S_1.\overline{S_0}.I_2[2] + \overline{S_2}.S_1.S_0.I_3[2] + S_2.\overline{S_1}.\overline{S_0}.I_4[2] + S_2.\overline{S_1}.S_0.I_5[2] + S_2.S_1.\overline{S_0}.I_6[2] + S_2.S_1.S_0.I_7[2]$$

1-bit ALU Circuit

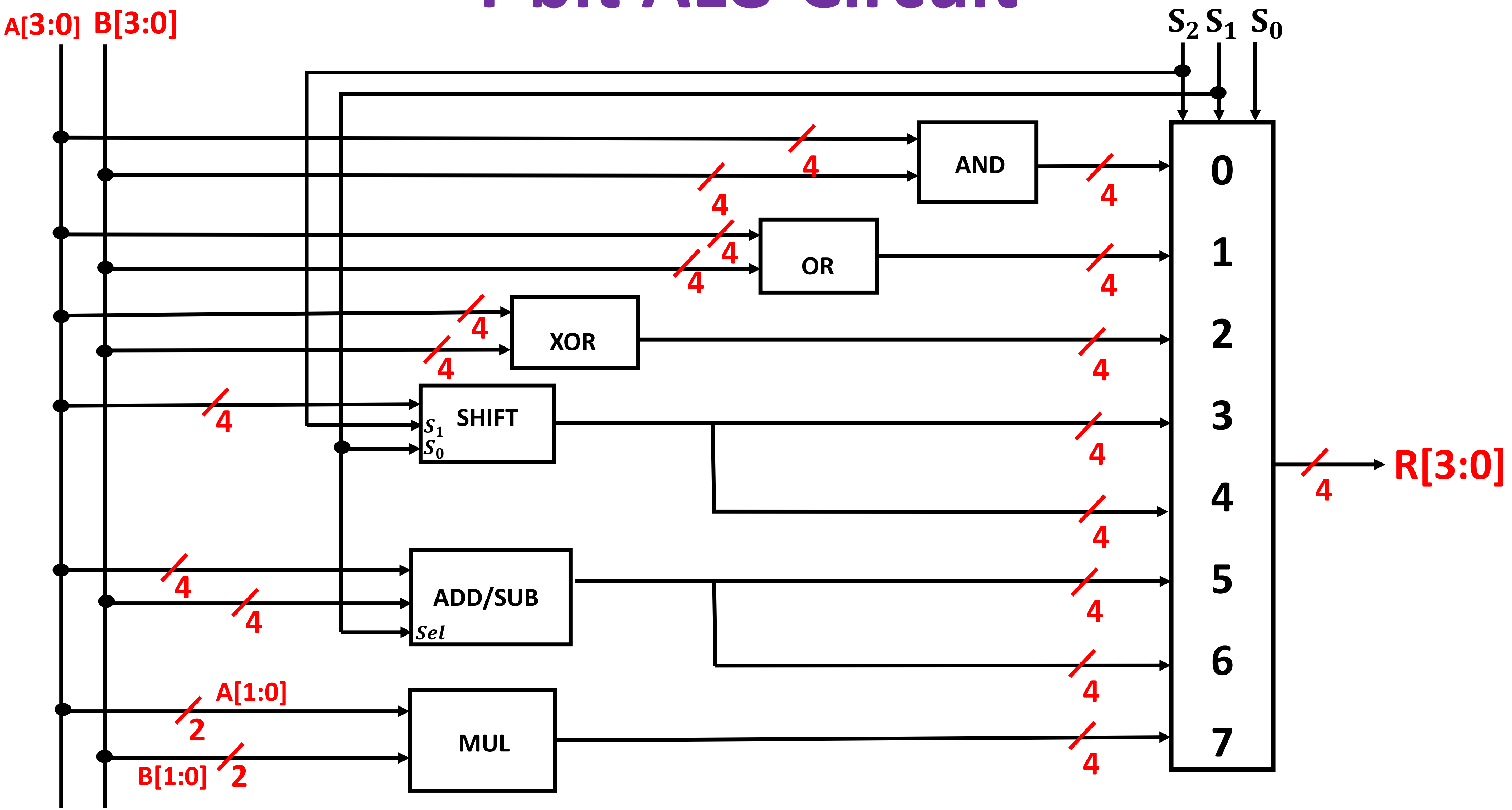


Operation	S_2	S_1	S_0	Y[3]	Y[2]	Y[1]	Y[0]
AND	0	0	0	$I_0[3]$	$I_0[2]$	$I_0[1]$	$I_0[0]$
OR	0	0	1	$I_1[3]$	$I_1[2]$	$I_1[1]$	$I_1[0]$
XOR	0	1	0	$I_2[3]$	$I_2[2]$	$I_2[1]$	$I_2[0]$
LEFT SHIFT	0	1	1	$I_3[3]$	$I_3[2]$	$I_3[1]$	$I_3[0]$
RIGHT SHIFT	1	0	0	$I_4[3]$	$I_4[2]$	$I_4[1]$	$I_4[0]$
ADD	1	0	1	$I_5[3]$	$I_5[2]$	$I_5[1]$	$I_5[0]$
SUB	1	1	0	$I_6[3]$	$I_6[2]$	$I_6[1]$	$I_6[0]$
MUL	1	1	1	$I_7[3]$	$I_7[2]$	$I_7[1]$	$I_7[0]$

If $\text{sel} = 0,\ S = \text{ADD}$
If $\text{sel} = 1,\ S = \text{SUB}$

If $S_1 = 0, S_0 = 1, S = \text{LEFT SHIFT}$
If $S_1 = 1, S_0 = 0, S = \text{RIGHT SHIFT}$

4-bit ALU Circuit



Home Work:
Design a 2-bit ALU that supports following operations: AND, LEFT SHIFT, MUL & SUB.

Thank You ☺