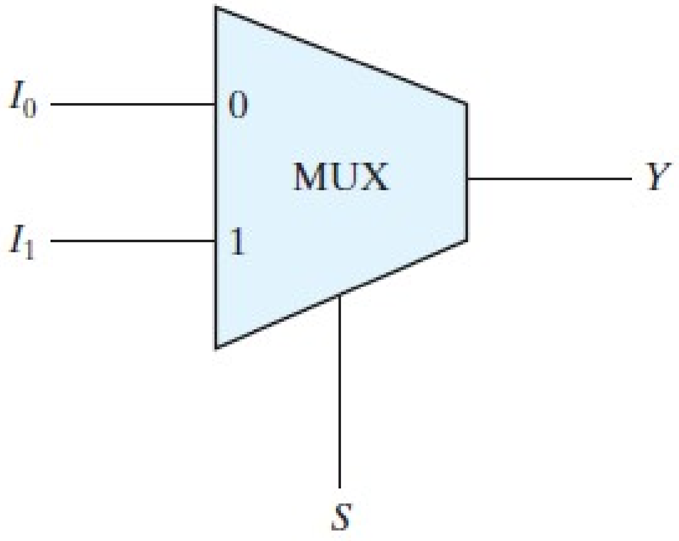
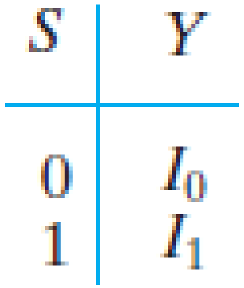
Comparators, Multiplexers

Multiplexers

- Directs one of multiple inputs to a single output
- Select lines determine what input is switched to the output. Passes an input to output
- Max # of inputs = , where n is # of select lines

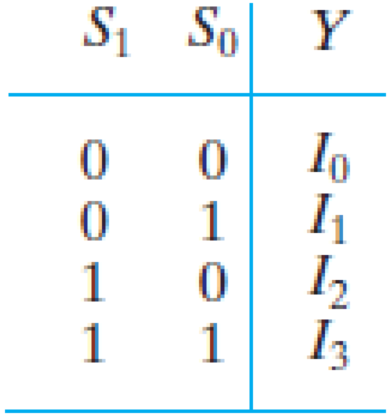




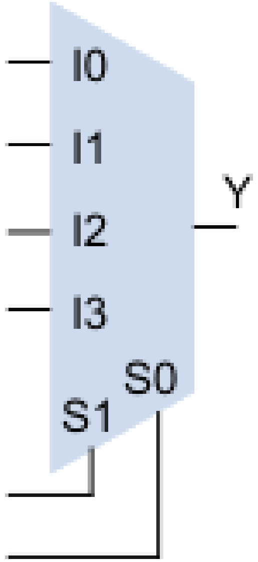
S	Y
0	I_0
1	I_1

Multiplexer

* 4 inputs need 2 select lines. is the MSB

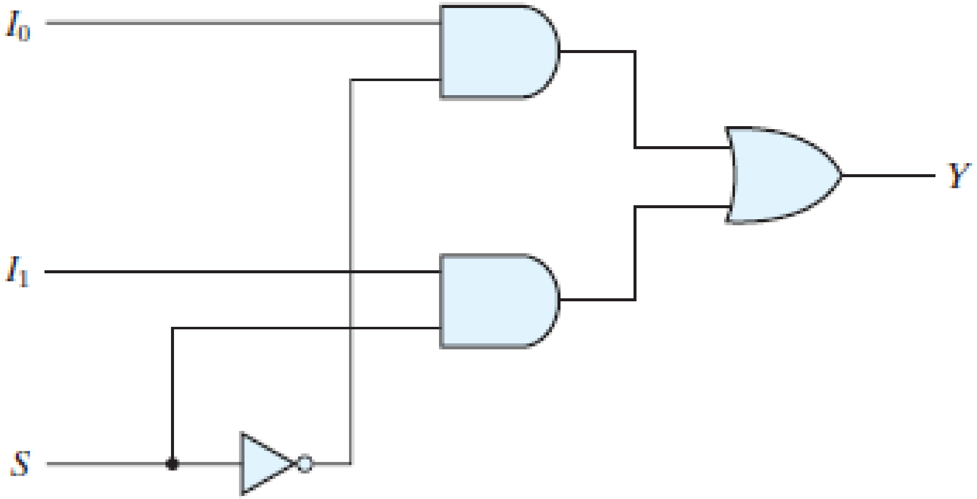


S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

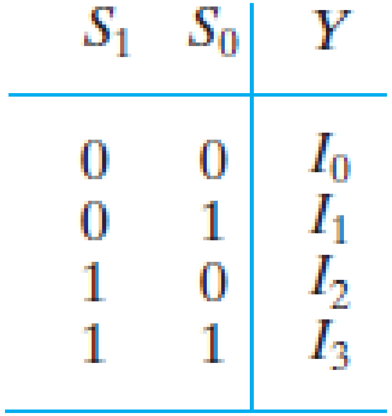


What’s Inside a Multiplexer

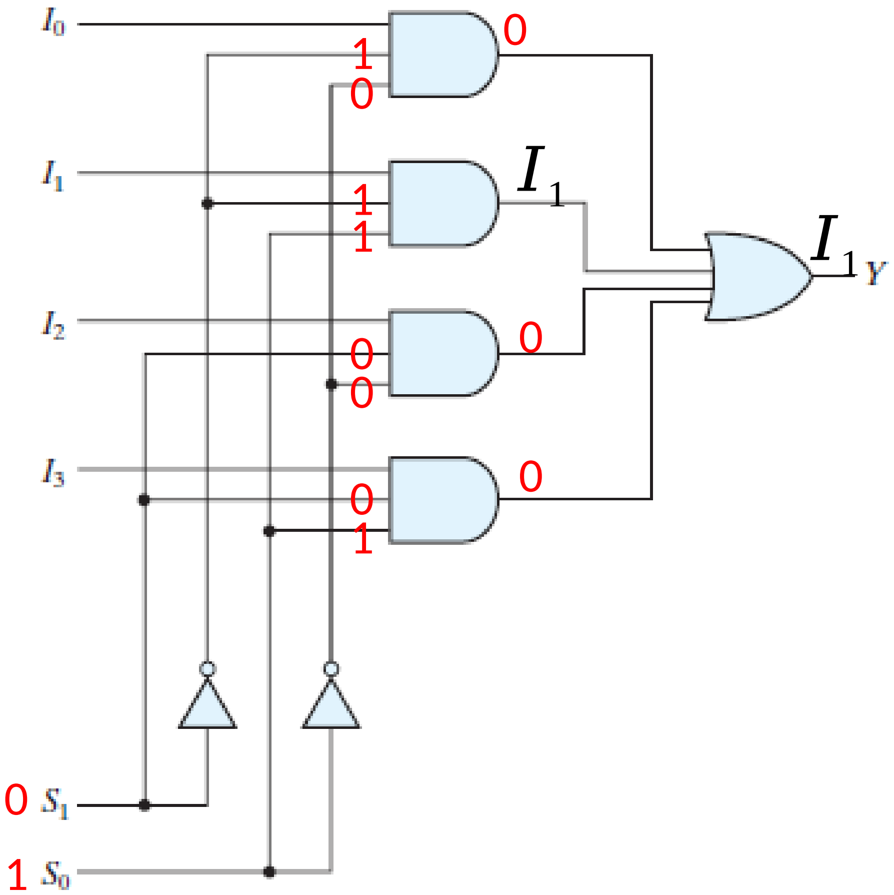
- Law of AND -> A AND 1 is A
- Law of OR -> A OR 0 is A
- Select lines are used to activate only one AND gate



4 Input

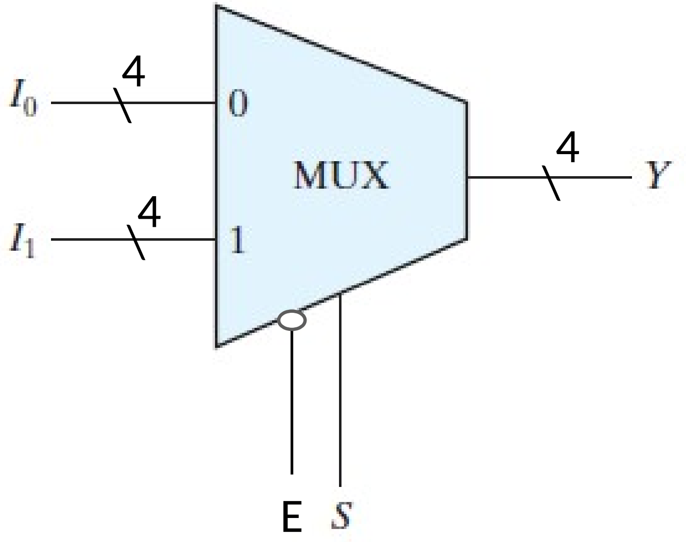


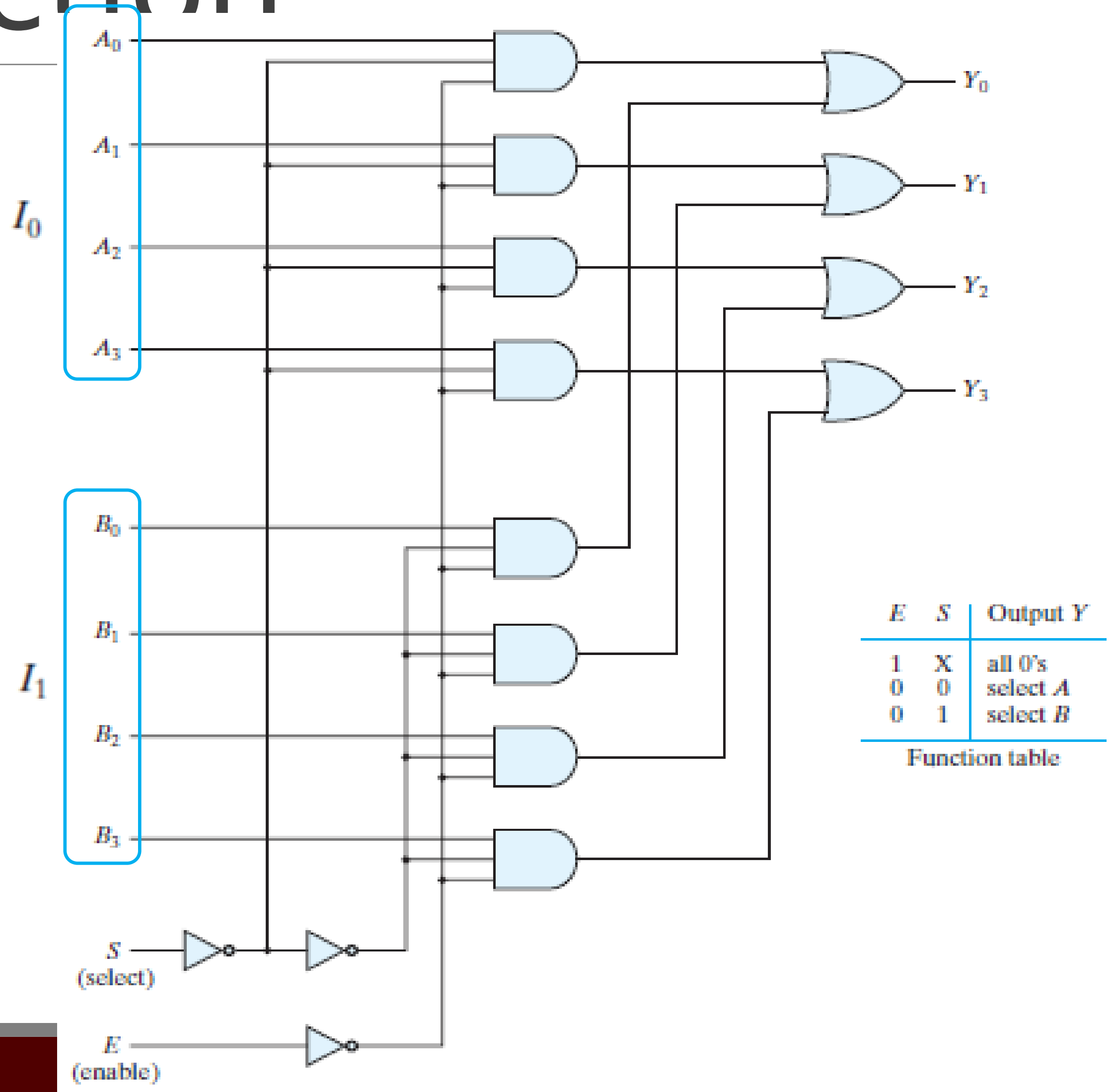
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Multiple Bit Selection

- Multiple bits can be passed to output
- Group A is one input of multiple bits
- Group B is another input of multiple bits
- 1 bit select line with active low enable





E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

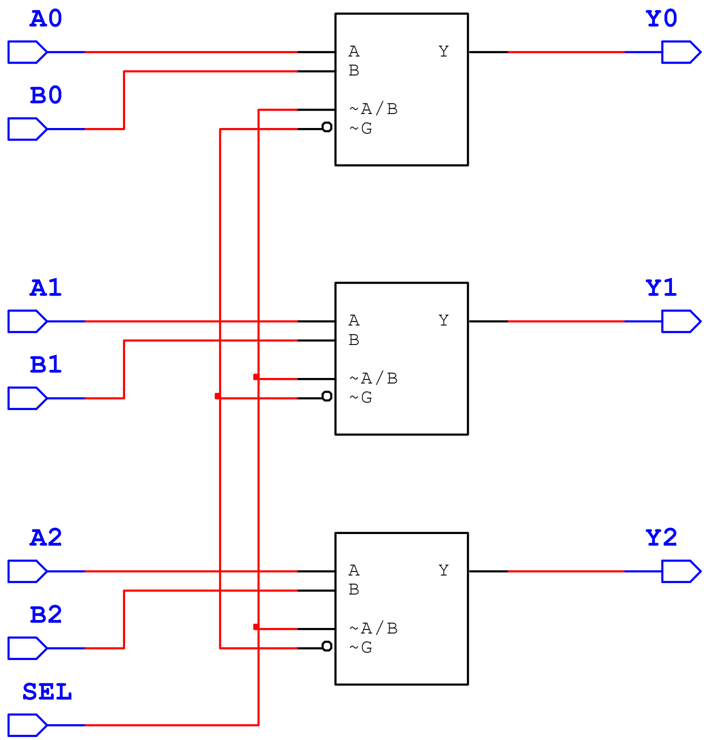
Function table

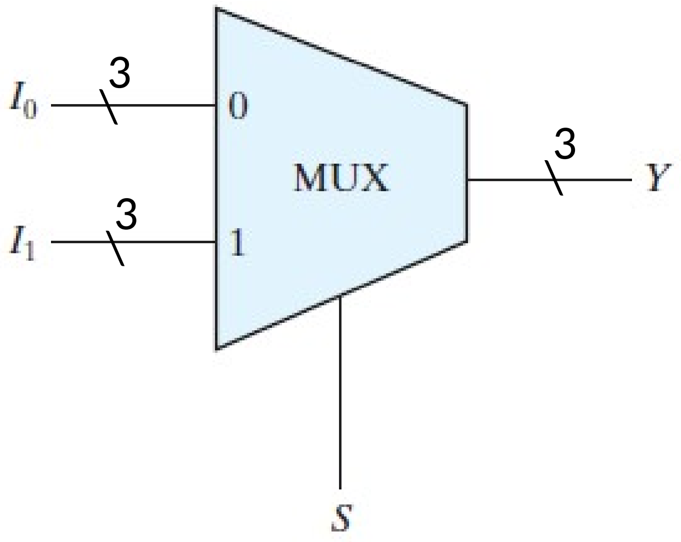
Using Parallel Multiplexers

- Placing multiplexers in parallel will allow for bus inputs
- The # of multiplexers in parallel is the number of bits on each bus input
- Each multiplexer will switch one digit of the multiple bit number

Parallel Multiplexer

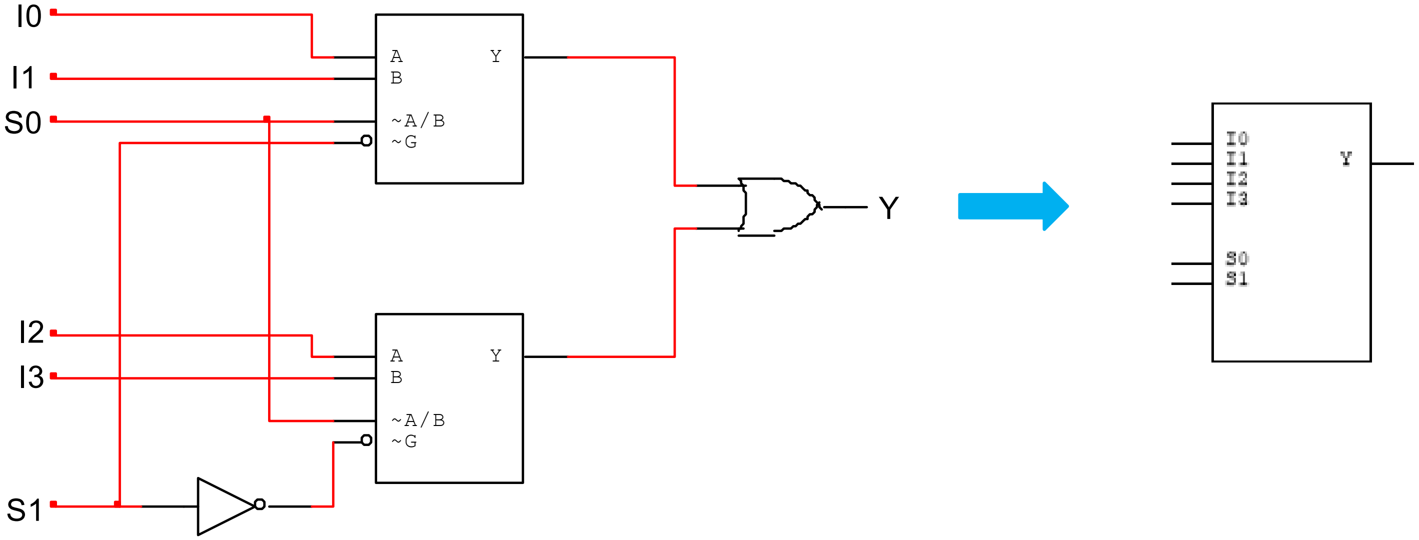
* Allows a two 3-bit inputs to be switched to 3 – bit output





Cascading Multiplexers

- Use enable on multiplexer to create additional select line
- Connect remaining select lines together
- OR outputs from each multiplexer

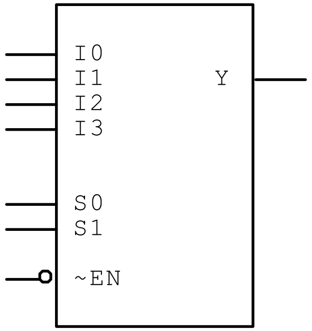


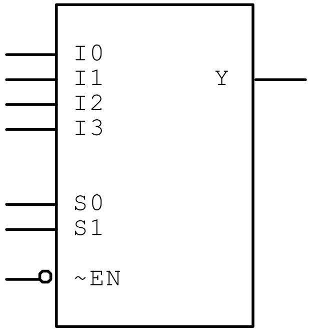
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Cascade Multiplexer

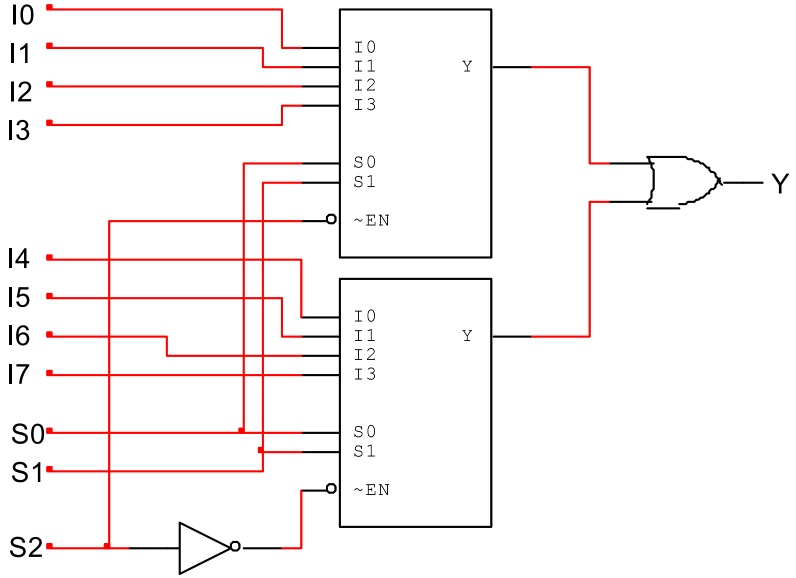
- How would a 4 input multiplexer be cascaded to create a 8 input multiplexer?

S2	S1	S0	Y
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7





Cascade Multiplexer



Implement Combinational Logic Circuits

* Multiplexers can implement outputs of truth table

* One multiplexer for each output

* The number of select lines on a multiplexer is number of inputs - 1

- LSB will be input to multiplexer
- Remaining bits will be inputs to the select lines
- 3 input truth table uses 2 select lines which is a 4 input multiplexer

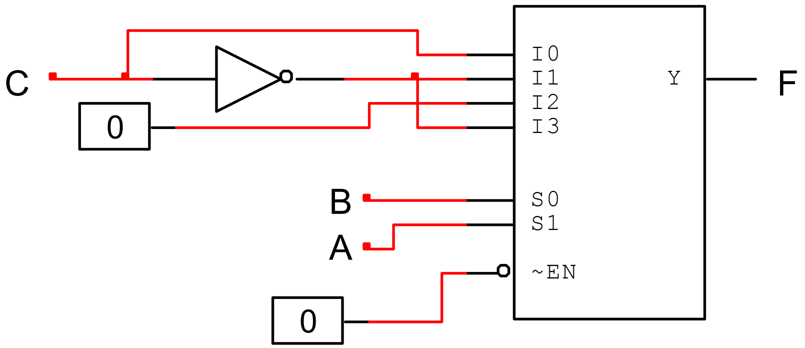
* Divide truth table in groups of 2, and compare how LSB changes to output. That becomes input to multiplexer

Example

Implement truth table with multiplexer

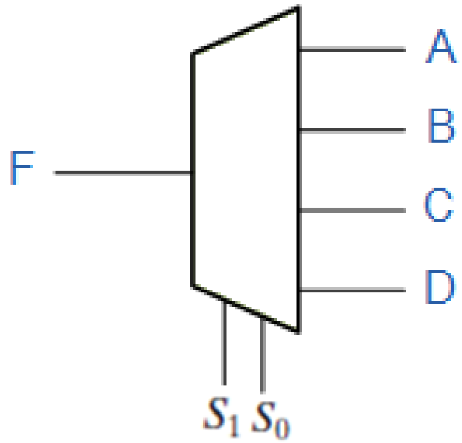
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

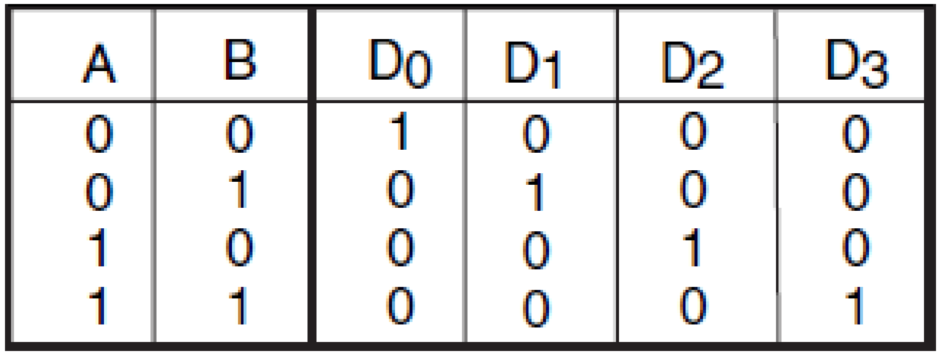
Example



Demultiplexer

- Directs one input to one of multiple output lines
- Number of max outputs is 2^n, where n is # of select lines

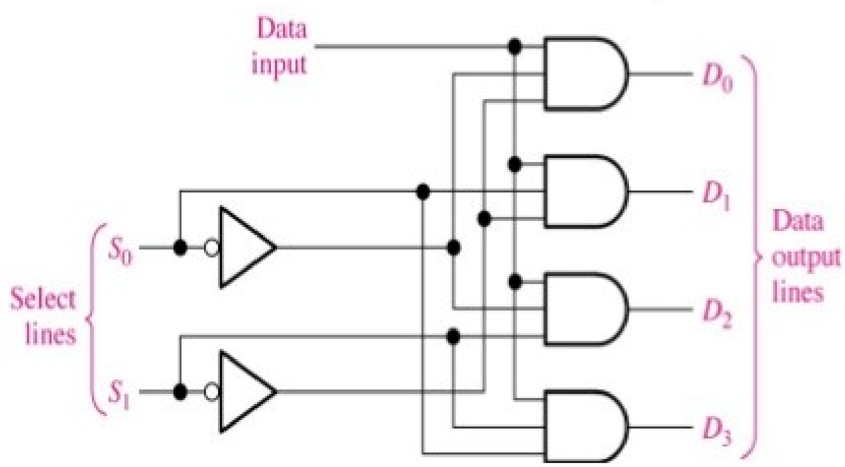


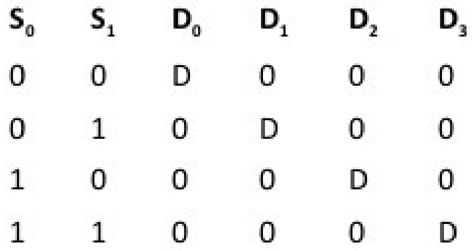


A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

What’s Inside a Demultiplexer

* AND gates to pass input to an output. NOT gates for select lines





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

Multiplexer & Demultiplexer Uses

* Communication systems

* Microcontroller peripherals and functions

* Signal routing between destinations

* Switching functions in electronics

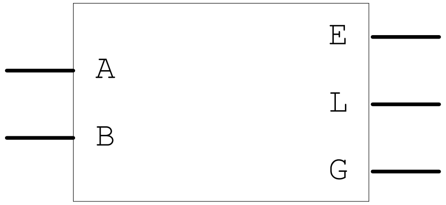
Comparator

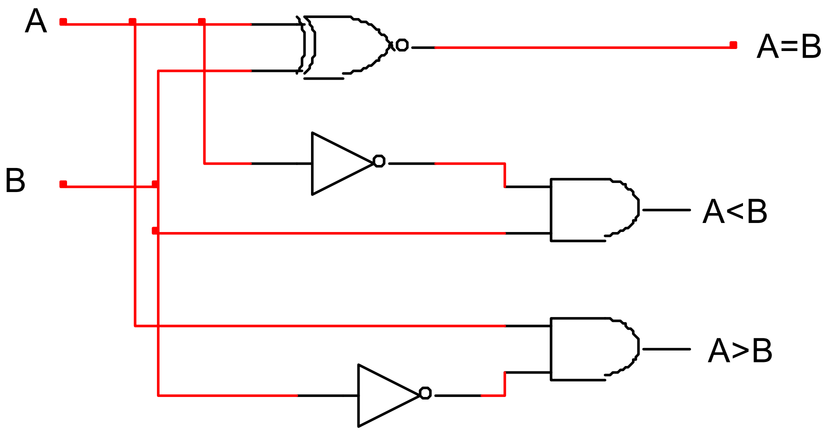
- Logic device that compares two numbers (A and B) and determines the following:
  - A = B
  - A < B
  - A > B
- N – bit comparator
  - N is number of bits in each number to compare

Making a 1-Bit Comparator

Truth table of 1-bit comparator

A	B	A = B	A < B	A > B
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0





1-bit comparator circuit

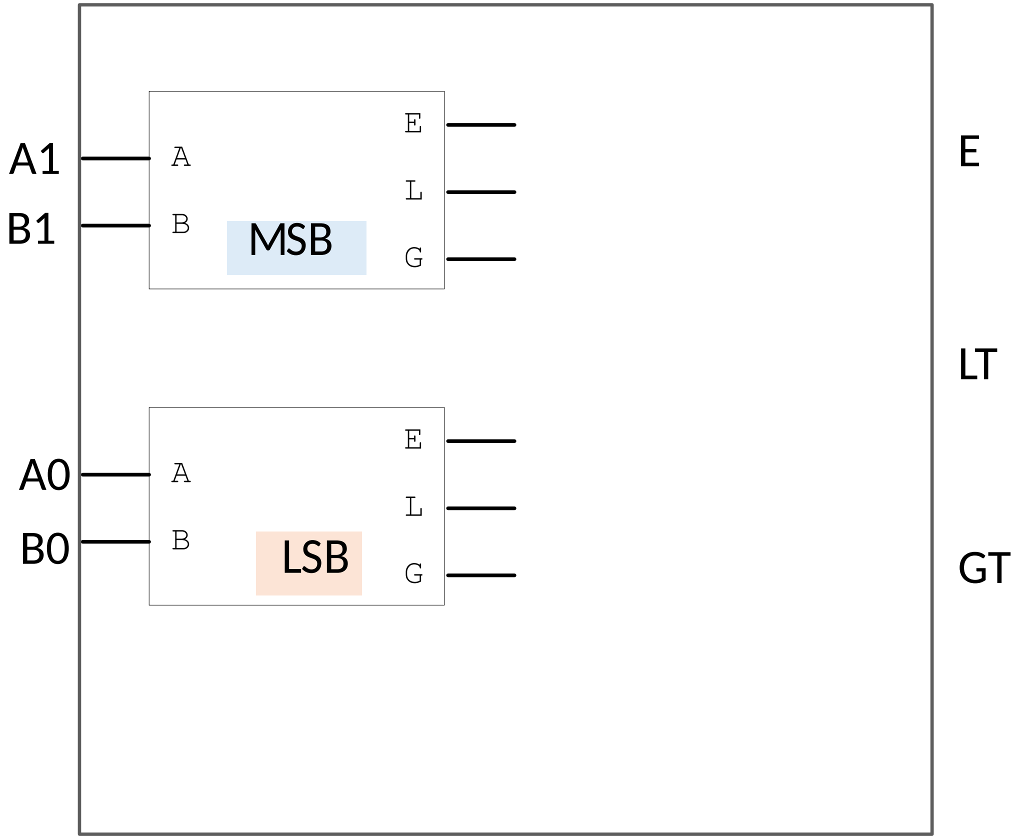
Cascading 1-Bit Comparator

- Cascading two 1-bit comparators will make a 2-bit comparator
- One comparator will compare the MSB bits and the other the LSB bits

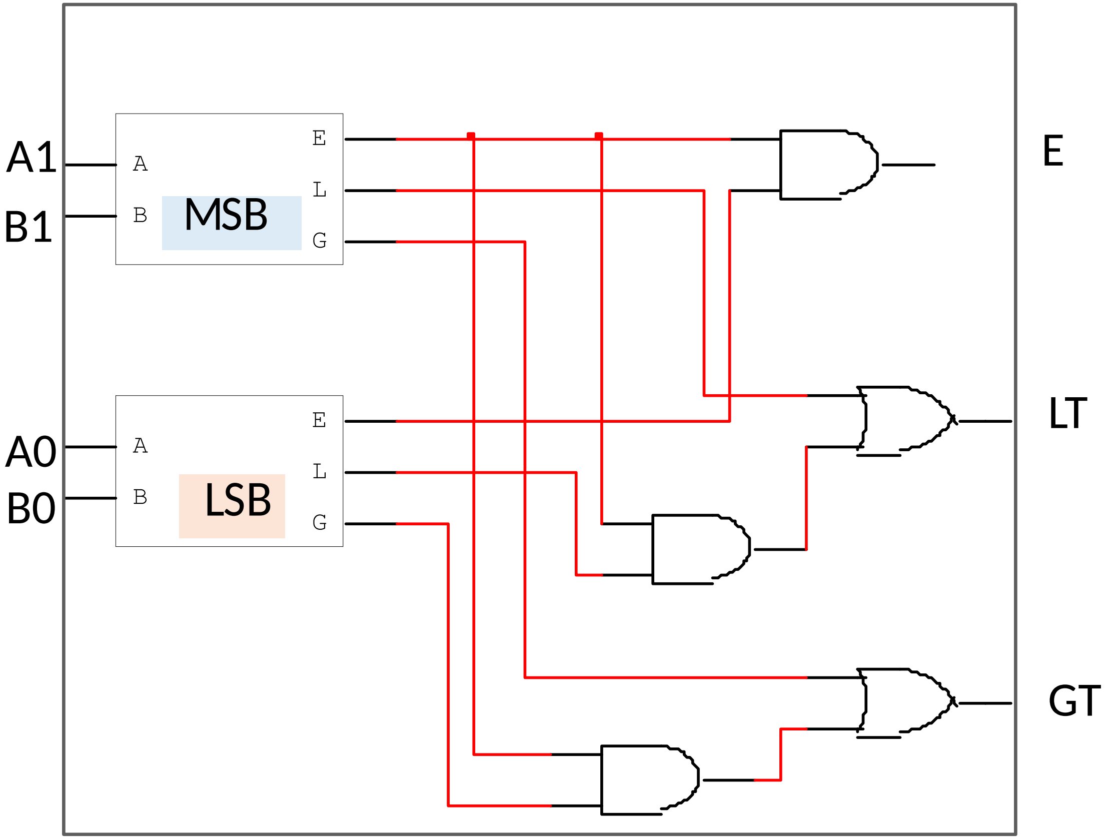
Possible Scenarios

A1	B1	A0	B0	
0	0	0	0	A=B
0	1	1	1	A<B
1	1	0	1	A<B
1	0	0	0	A > B
0	0	1	0	A > B

MSB
LSB

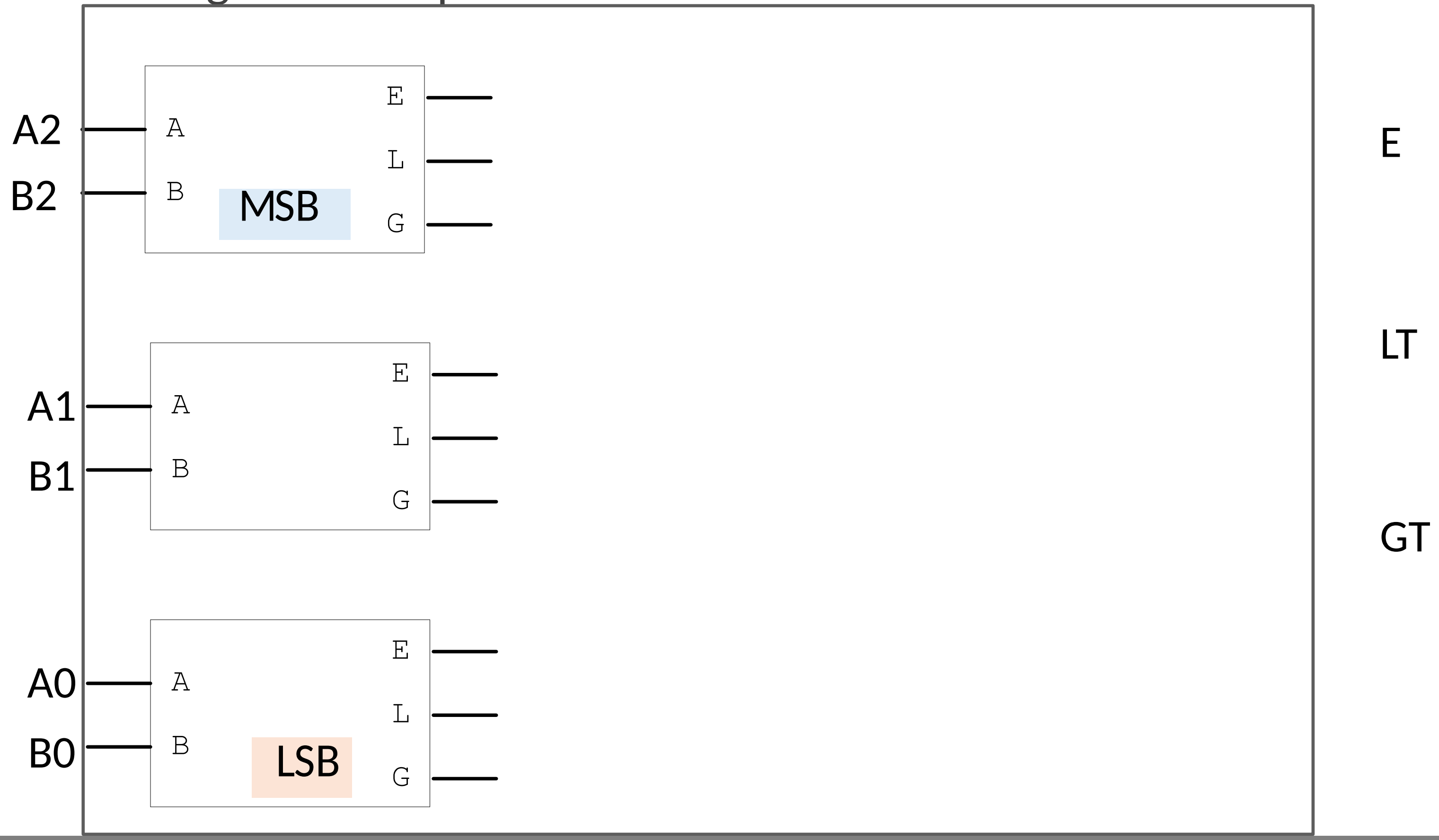


2-Bit Comparator

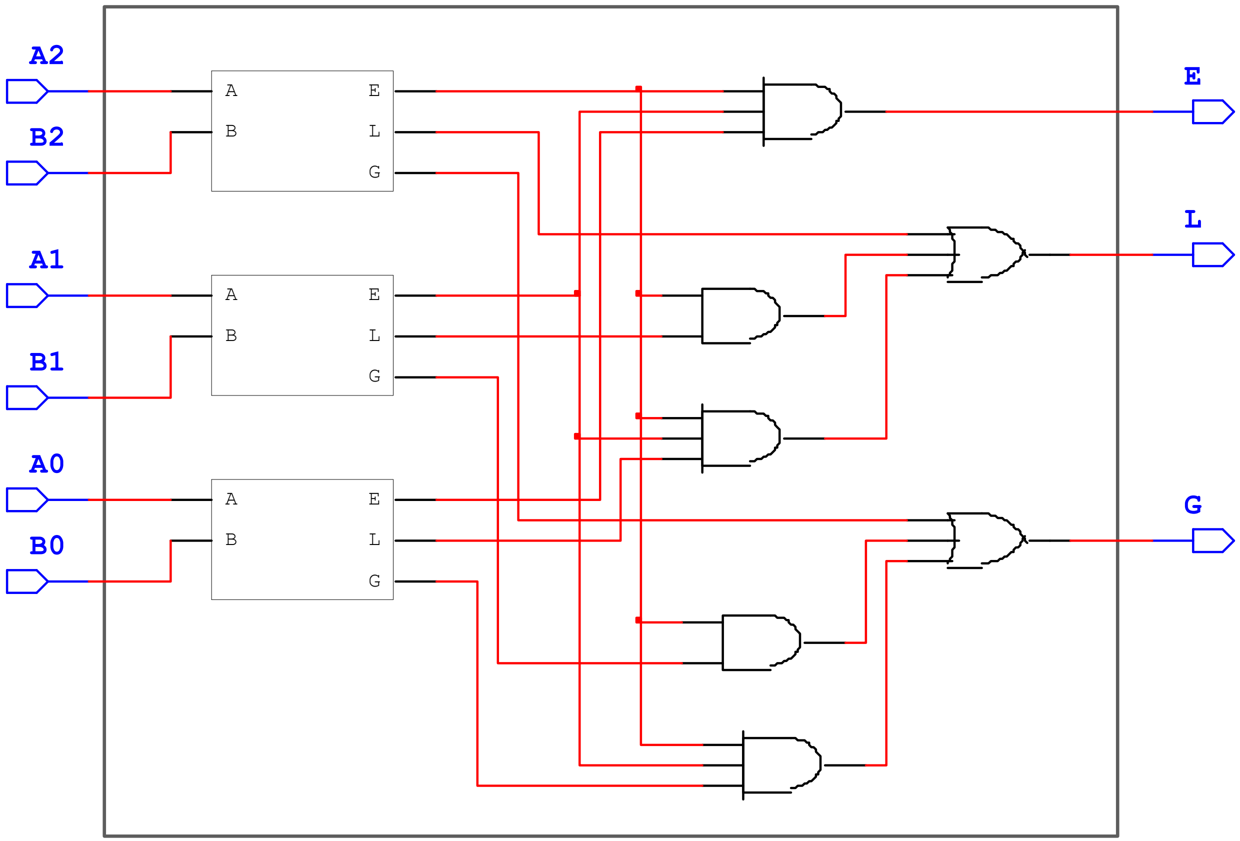


3 – Bit Comparator

Design a 3-bit comparator using 1-bit comparators

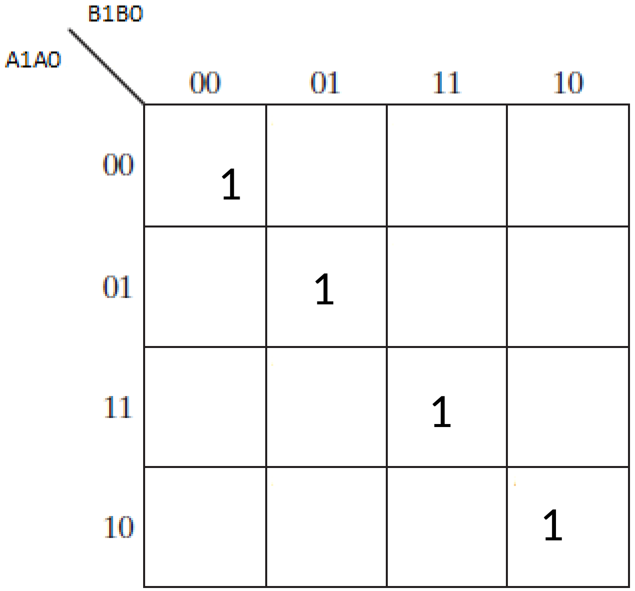


3 – Bit Comparator



Making a 2-bit Comparator

A1	A0	B1	B0	A = B	A < B	A > B
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	1	0	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	1	0	0



A1A0 \ B1B0	00	01	11	10
00	1			
01		1		
11			1	
10				1

A = B

2 – bit Comparator

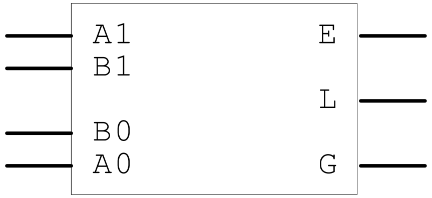
A1A0 \ B1B0	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

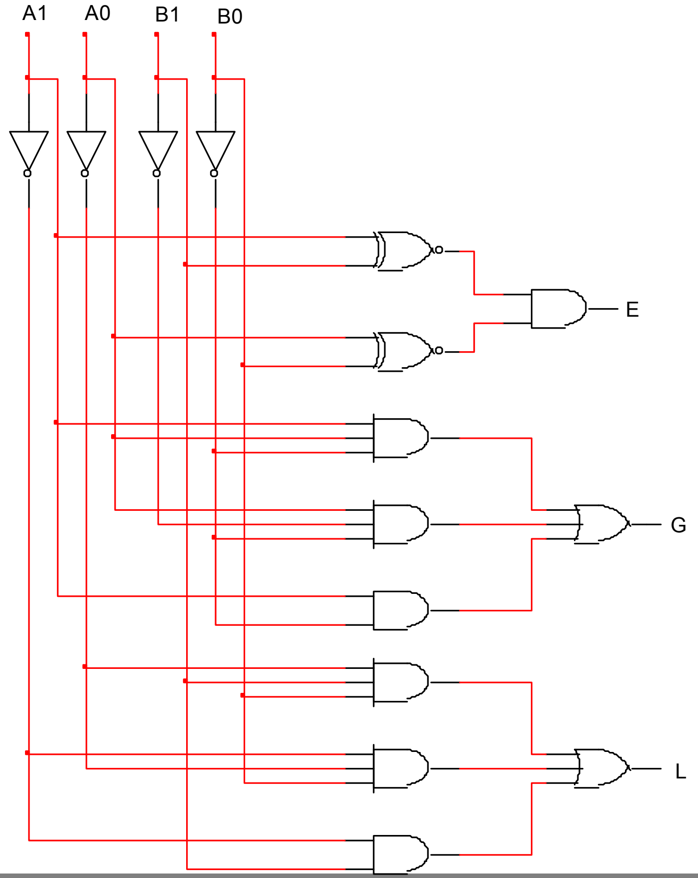
A > B

A1A0 \ B1B0	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

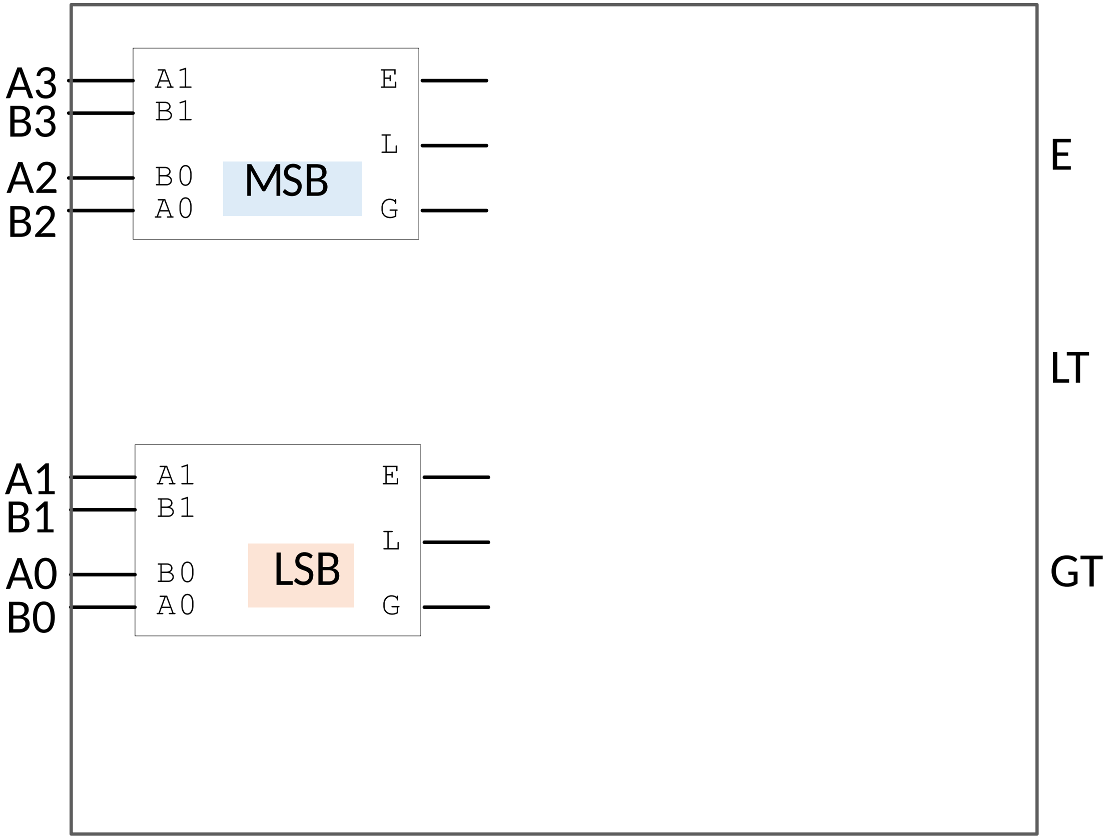
A < B

2 – bit Comparator





Cascading to make 4-bit



The condition for A<B can be possible in the following four cases:

If A3 = 0 and B3 = 1

If A3 = B3 and A2 = 0 and B2 = 1

If A3 = B3, A2 = B2 and A1 = 0 and B1 = 1

If A3 = B3, A2 = B2, A1 = B1 and A0 = 0 and B0 = 1

The condition of A>B can be possible in the following four cases:

If A3 = 1 and B3 = 0

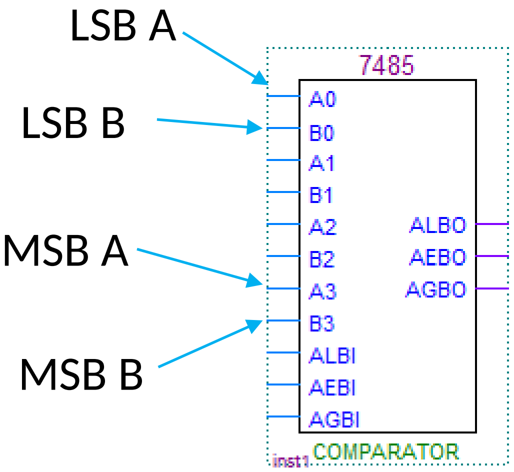
If A3 = B3 and A2 = 1 and B2 = 0

If A3 = B3, A2 = B2 and A1 = 1 and B1 = 0

If A3 = B3, A2 = B2, A1 = B1 and A0 = 1 and B0 = 0

Comparator Schematics

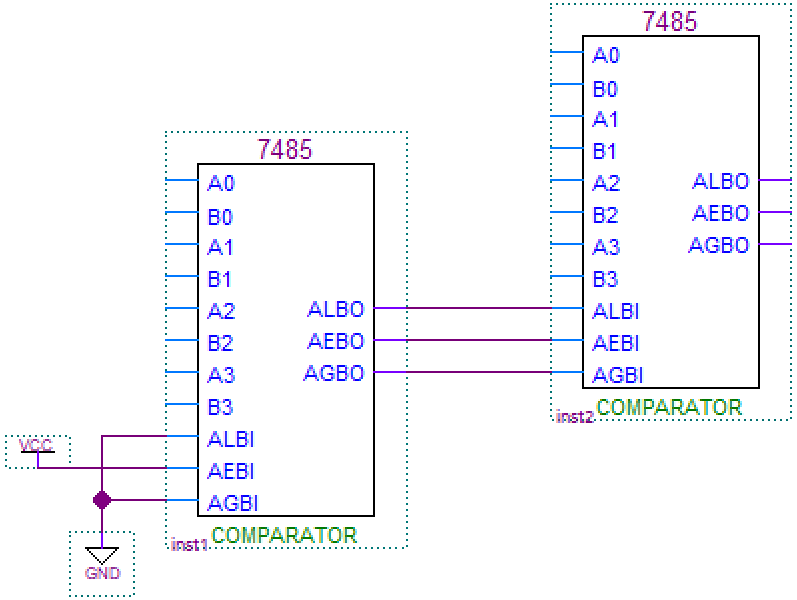
* 4 – bit comparator


LSB A

LSB B

MSB A

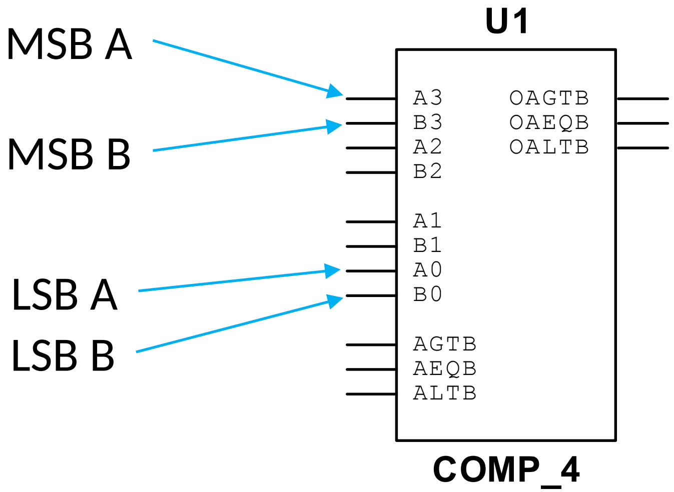
MSB B

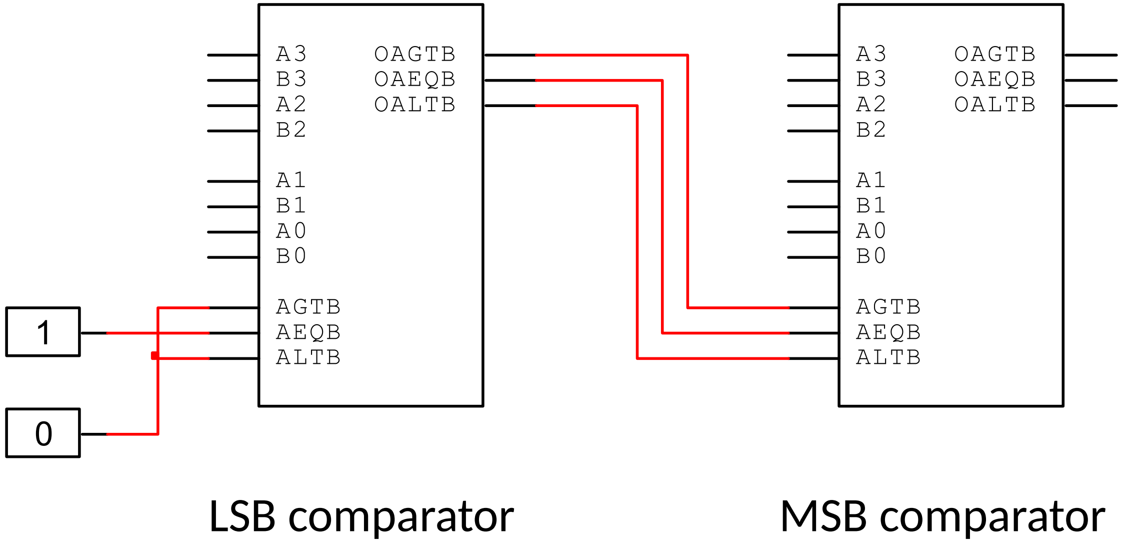


Cascading Comparators

Comparator Schematics

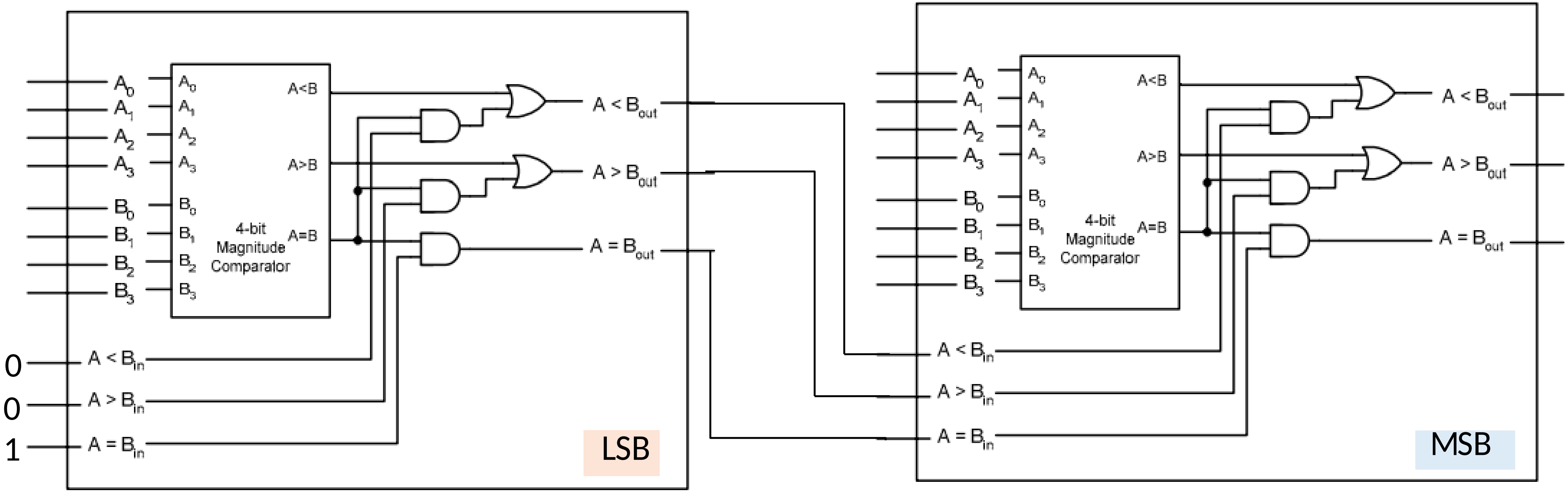
- 4 – bit comparator





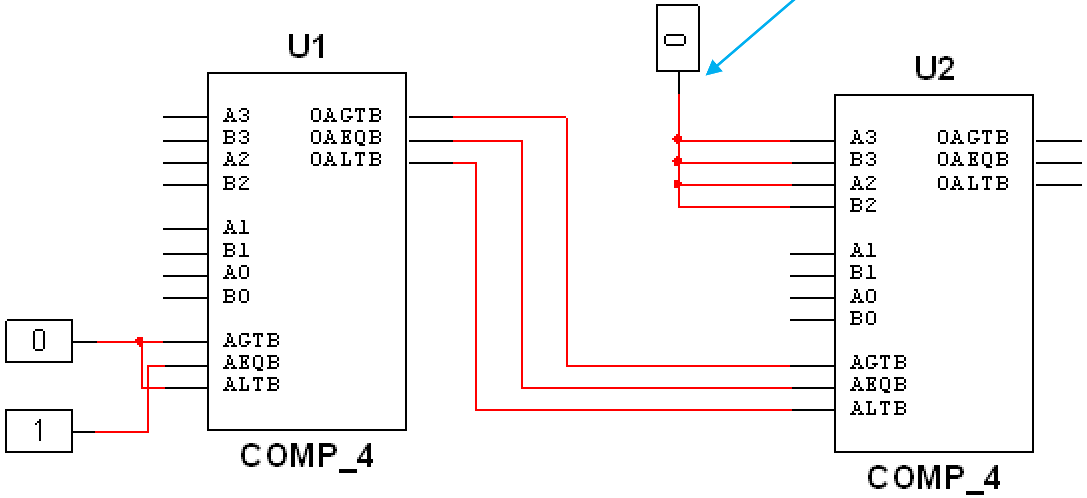
Expansion Inputs

* Result of LSB comparator are carried into the MSB comparator



Unused Inputs

- Need two 4-bit comparators to compare 6-bit numbers
- Not all inputs will be used, unused inputs are inactive



Equivalent to padding 0s on MSB