

Combinational Logic

Dr. E. Paul Braineard

Logic Circuits

- Combinational circuit
 - Logic gate circuit, whose output at a particular time instant is dependent on input combination at that particular time instant
- Sequential circuit
 - Logic gate circuit + Storage element
 - Output depends on the input combination and the state of the memory element

Combinational Circuit

- Logic circuit that consists of interconnection of logic gates
- For m inputs, the possible combinations of binary inputs = 2^m
- For each possible input combination, there is one possible value at the output
- A combinational circuit can be described by truth table



Block diagram of combinational circuit

Combinational Circuits

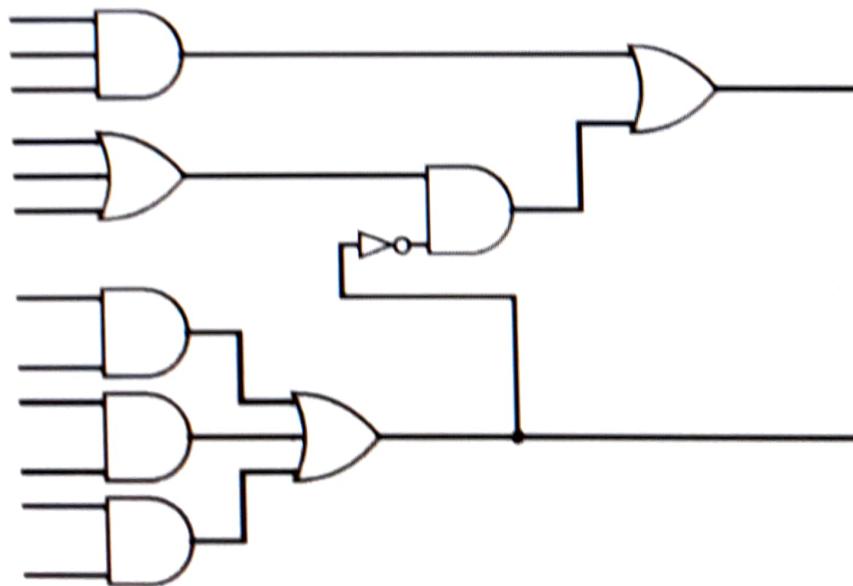
- Binary Adder, Subtractor
- Decimal Adder
- Binary multiplier
- Magnitude comparator
- Decoder
- Encoder
- Multiplexer

Analysis of Combinational Circuit

- Determine the function the circuit implements
- Start with
 - Logic circuit
 - Boolean functions
 - Truth table
 - Explanation of the circuit operation
- Make sure the circuit is not sequential
- Make sure there is no feedback path or memory elements

Procedure to obtain Boolean function from logic circuit

- Label all gate first stage outputs with unique names
- Proceed to the next stage, till you reach final output



Combinational Circuit

$$P = ABC$$

$$Q = A + B + C$$

$$R = AB$$

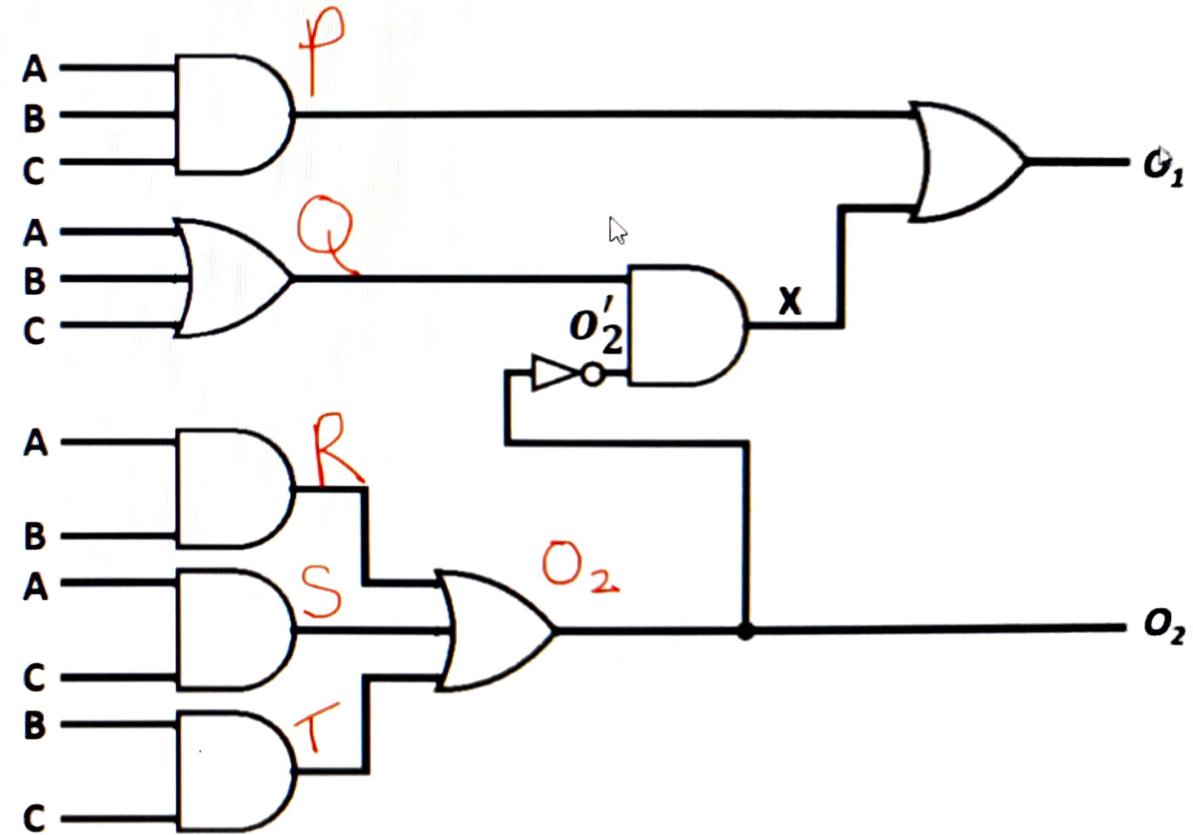
$$S = AC$$

$$T = BC$$

$$O_2 = R + S + T$$

$$O_2 = AB + AC + BC$$

$$O_1 = P + X = (ABC) + (QO_2)$$



Code conversion example combinational logic circuit

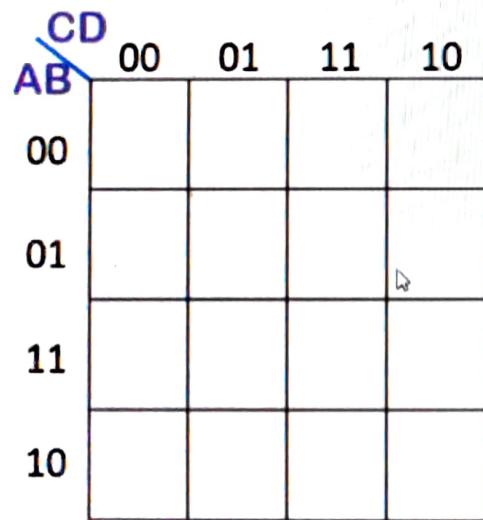
Decimal	Binary	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100



Input BCD				Output Excess-3			
A	B	C	D	P	Q	R	S
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

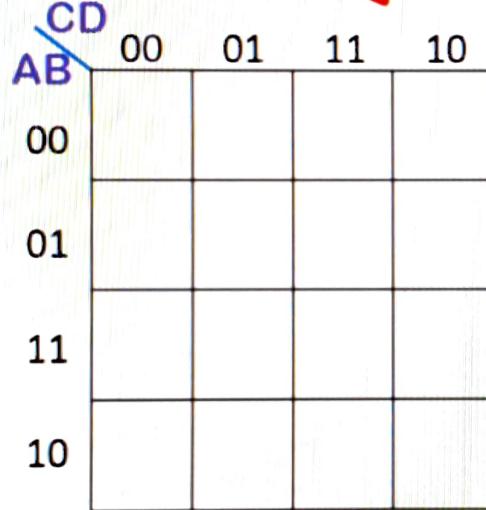
K-maps for the output variables

P

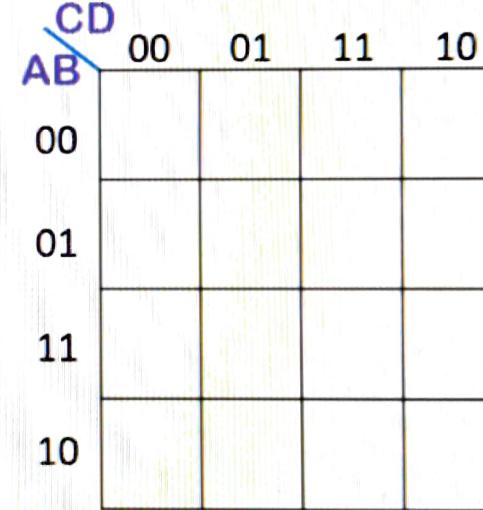


m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

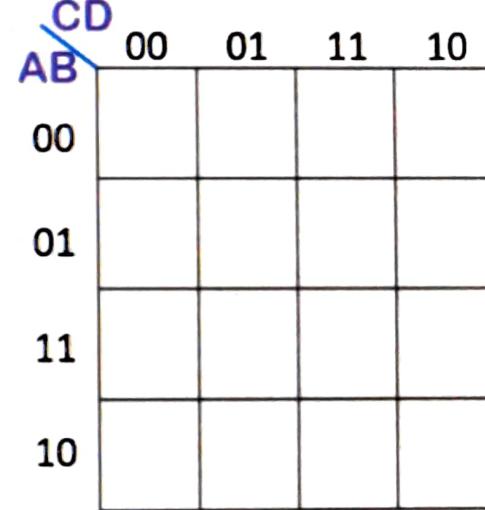
Q



R



S



Don't care conditions

- The six bit combinations not listed beyond 1001 for input are don't-care combinations
- These values have no meaning in BCD and we assume that they will never occur in actual operation of the circuit
- Therefore we have the liberty to take either 0 or 1
- Don't care is represented by X
- Idea is to get simple circuit

Binary	Excess-3
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

Four-variable K-map

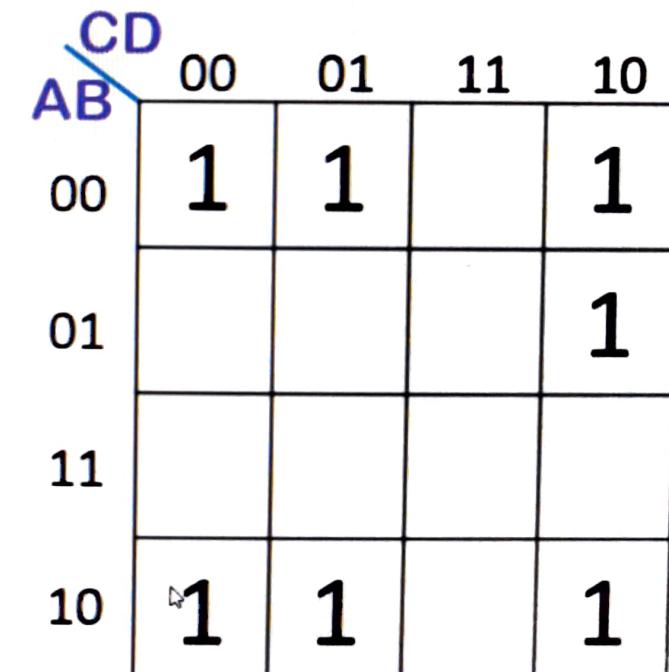
Simplify the Boolean function given in standard SOP form

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$

Add missing variables

$$F = A'B'C'(D+D') + (A+A')B'CD' + A'BCD' + AB'C'(D+D')$$

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}



K-maps for the output variables

P

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

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

Q

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

R

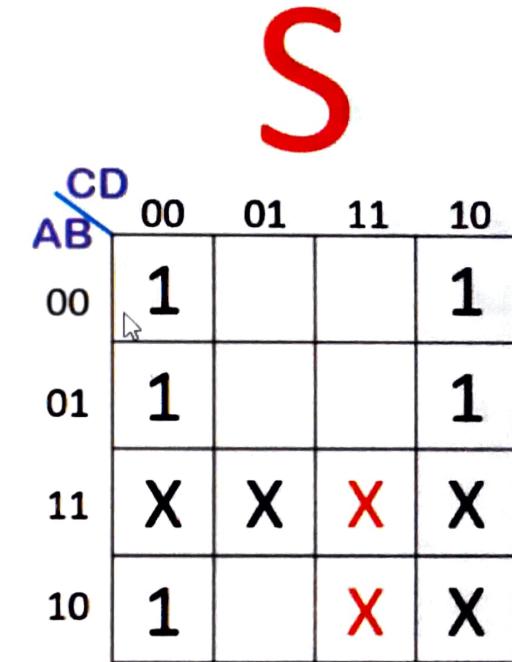
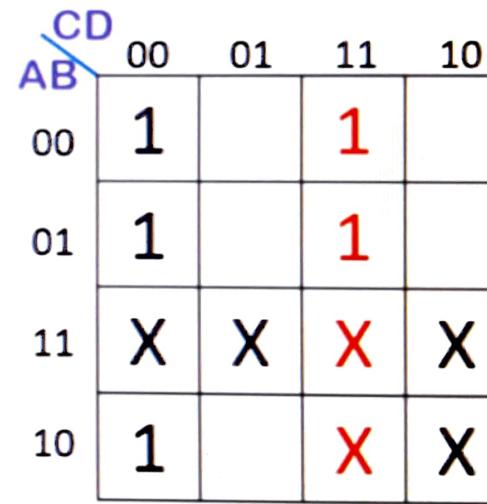
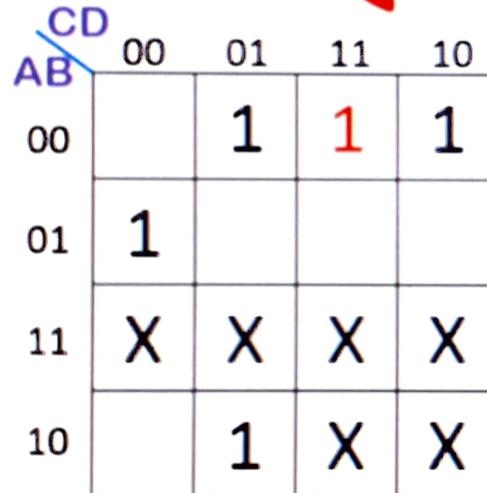
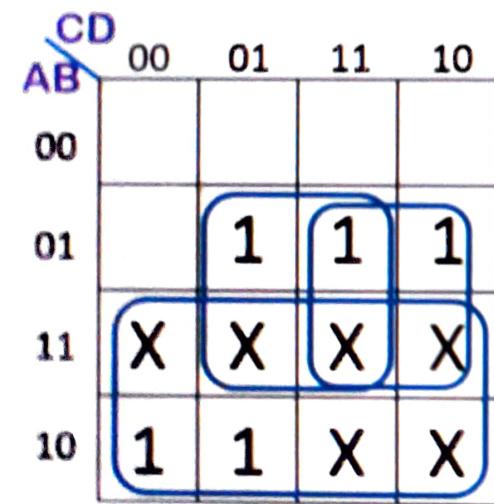
AB\CD	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

S

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

K-maps for the output variables

$$P = \overline{A} + \overline{B}P + BC$$



Full Adder

Input			Output	
A	B	C	Carry	Sum
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

Two-bit multiplier

		$A_1 A_0$	$\times B_1 B_0$	
		$A_1 B_0$	$A_0 B_0$	Partial products
$A_1 B_1$	$A_0 B_1$			
P_3	P_2	P_1	P_0	

$$P_0 = A_0 B_0$$

$$P_1 = A_1 B_0 + A_0 B_1$$

$$P_2 = A_1 B_1$$

$$P_3 = \text{Carry out from } P_2$$

2 bit \times 2bit
multiplier

Product

$P_3 \ P_2 \ P_1 \ P_0$

A_1	A_0	B_1	B_0	P_3	P_2	P_1	P_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Magnitude comparator

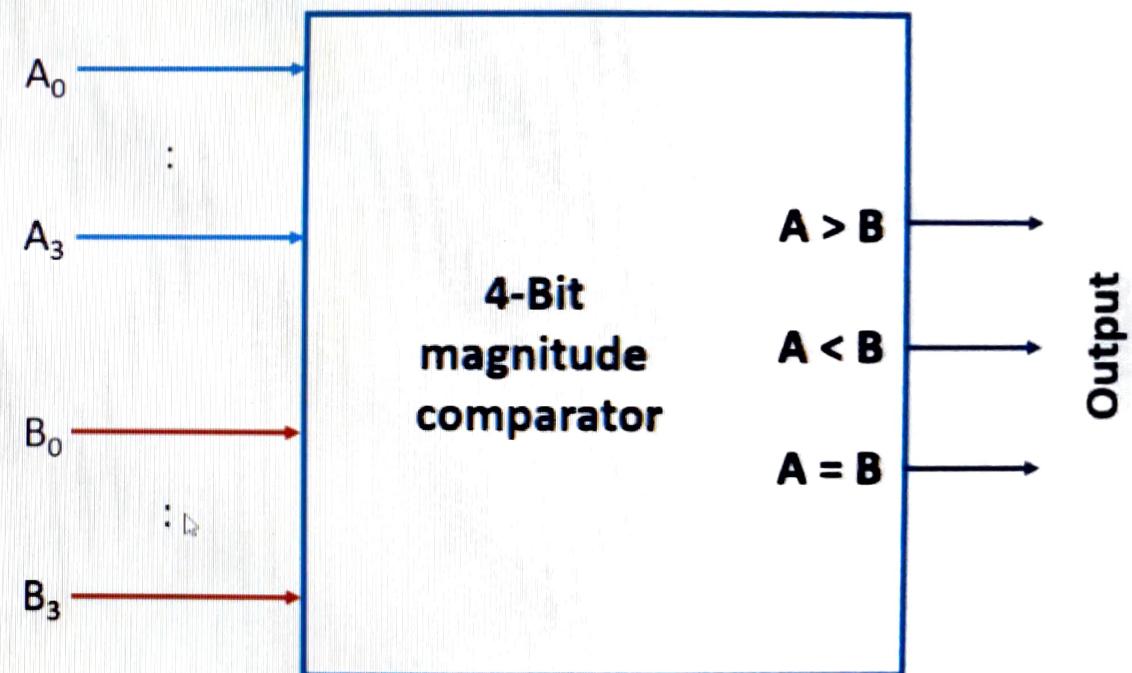
- Comparison of two numbers to determine whether a number is
 - $>$
 - $<$
 - $=$the other number
- A magnitude comparator is a combinational circuit that compares two numbers A and B and results in a output to indicate

$A > B$

$A < B$

$A = B$

4-Bit Magnitude comparator



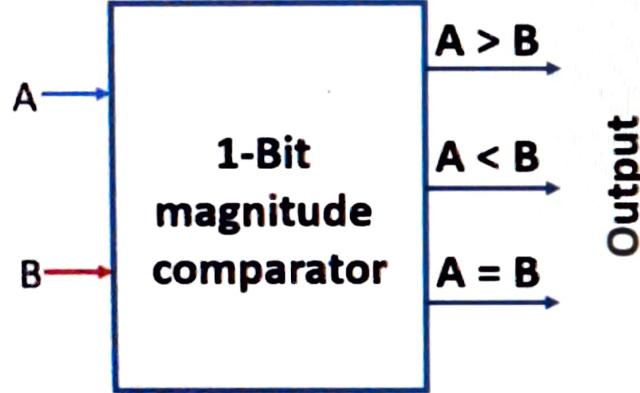
4-Bit Magnitude comparator

A3 & B3	A2 & B2	A1 & B1	A0 & B0	A > B	A < B	A = B
A3 > B3				1		0
A3 < B3					1	0
A3 = B3	A2 > B2			1		0
A3 = B3	A2 < B2				1	0
A3 = B3	A2 = B2	A1 > B1		1		0
A3 = B3	A2 = B2	A1 < B1			1	0
A3 = B3	A2 = B2	A1 = B1	A0 > B0	1		0
A3 = B3	A2 = B2	A1 = B1	A0 < B0		1	0
A3 = B3	A2 = B2	A1 = B1	A0 = B0			1

Decoder

- A combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines

1-Bit Magnitude comparator

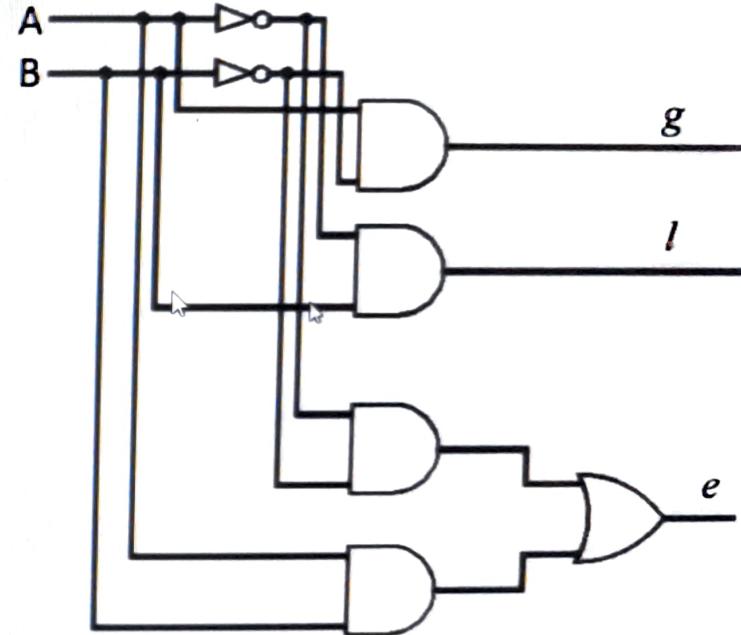


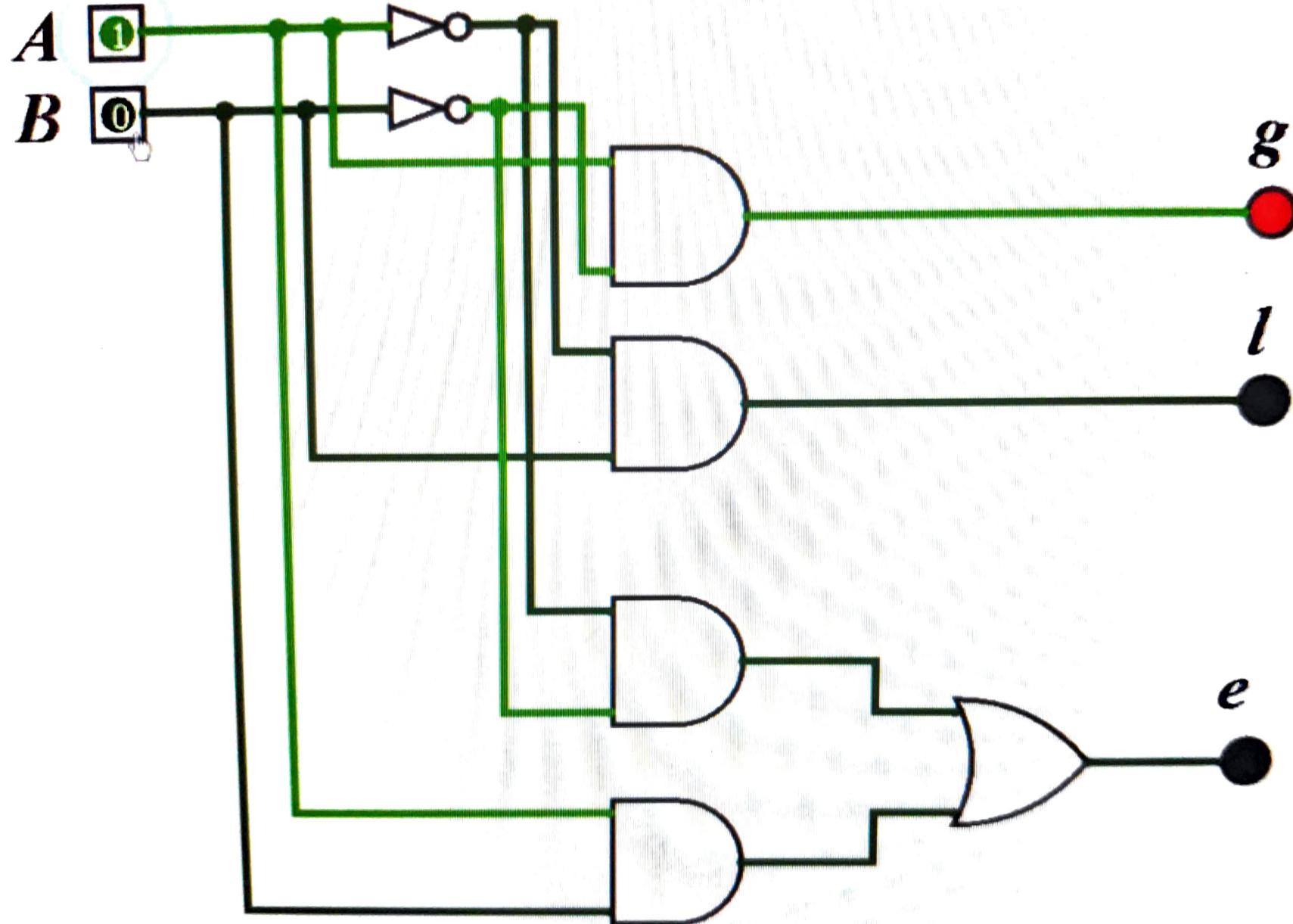
Inputs		Output		
A	B	g $A > B$	l $A < B$	e $A = B$
0	0			1
0	1		1	
1	0	1		
1	1			1

$$g = A\bar{B}$$

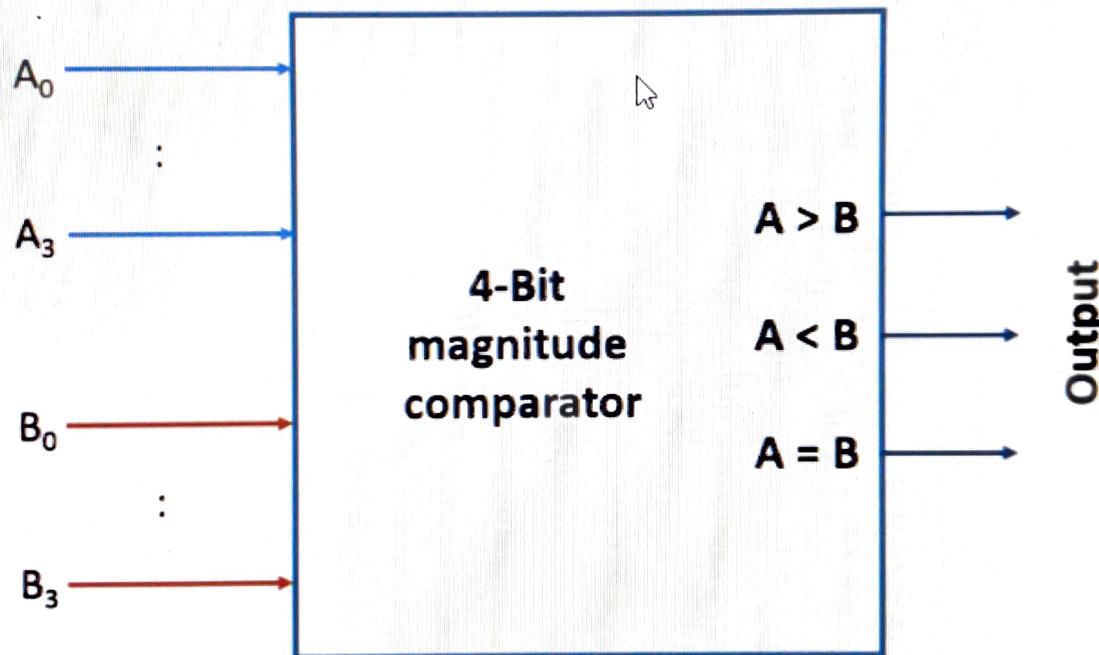
$$l = \bar{A}B$$

$$e = \bar{A}\bar{B} + AB = A \oplus B$$





4-Bit Magnitude comparator



4-Bit Magnitude comparator

A3 & B3	A2 & B2	A1 & B1	A0 & B0	g A > B	I A < B	e A = B
A3 > B3				1		
A3 < B3					1	
A3 = B3	A2 > B2			1		
A3 = B3	A2 < B2				1	
A3 = B3	A2 = B2	A1 > B1		1		
A3 = B3	A2 = B2	A1 < B1			1	
A3 = B3	A2 = B2	A1 = B1	A0 > B0	1		
A3 = B3	A2 = B2	A1 = B1	A0 < B0		1	
A3 = B3	A2 = B2	A1 = B1	A0 = B0			1

4-Bit Magnitude comparator

MSB ↓
 A₃ A₂ A₁, A₀ 1000 ✓
 B₃ B₂ B₁, B₀ 0999

A ₃ & B ₃	A ₂ & B ₂	A ₁ & B ₁	A ₀ & B ₀	g A > B	l A < B	e A = B
A ₃ > B ₃				1		
A ₃ < B ₃					1	
A ₃ = B ₃	A ₂ > B ₂			1		
A ₃ = B ₃	A ₂ < B ₂				1	
A ₃ = B ₃	A ₂ = B ₂	A ₁ > B ₁		1		
A ₃ = B ₃	A ₂ = B ₂	A ₁ < B ₁			1	
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ > B ₀	1		
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ < B ₀		1	
A ₃ = B ₃	A ₂ = B ₂	A ₁ = B ₁	A ₀ = B ₀			1

4-Bit Magnitude comparator: Equal



$$e = (A_3 \odot B_3) \bullet (A_2 \odot B_2) \bullet (A_1 \odot B_1) \bullet (A_0 \odot B_0)$$

4-Bit Magnitude comparator: Greater

A3 & B3	A2 & B2	A1 & B1	A0 & B0
A3 > B3			

$$A_3 > B_3 \rightarrow A_3 = 1 \text{ and } B_3 = 0 \rightarrow A_3 \bar{B}_3$$

A3 = B3	A2 > B2		
---------	---------	--	--

$$(A_3 \odot B_3) \text{ AND } A_2 = 1 \text{ AND } B_2 = 0 \rightarrow (A_3 \odot B_3) \cdot A_2 \bar{B}_2$$

$$A_2 \bar{B}_2$$

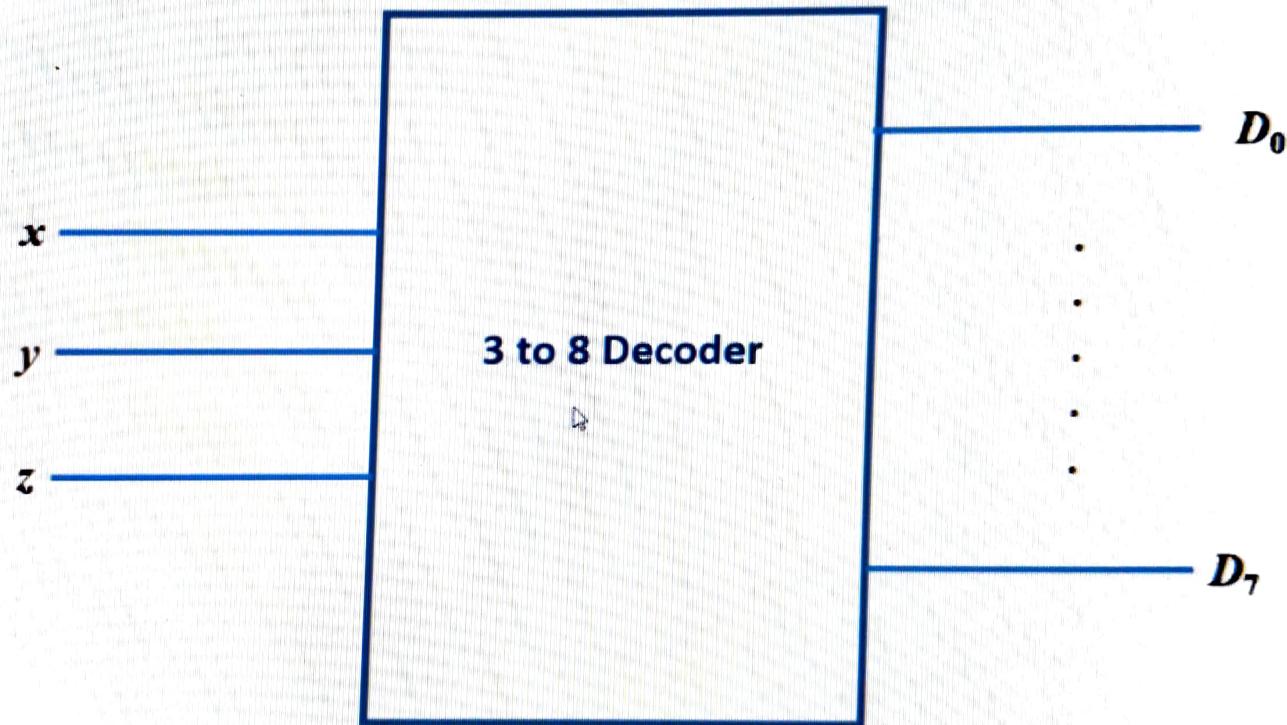
A3 = B3	A2 = B2	A1 > B1	
A3 = B3	A2 = B2	A1 = B1	A0 > B0

$$(A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot A_1 \bar{B}_1$$

$$(A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot A_0 \bar{B}_0$$

$$g = A_3 \bar{B}_3 + (A_3 \odot B_3) \cdot A_2 \bar{B}_2 + (A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot A_1 \bar{B}_1 + (A_3 \odot B_3) \cdot (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot A_0 \bar{B}_0$$

Decoder



Decoder

- A combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

$$D_0 = \bar{x} \bar{y} \bar{z}$$

$$D_1 =$$

$$D_2 =$$

$$D_3 =$$

$$D_4 =$$

$$D_5 =$$

$$D_6 =$$

$$D_7 =$$

Encoder

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1								0	0	0
	1							0	0	1
		1						0	1	0
			1					0	1	1
				1				1	0	0
					1			1	0	1
						1		1	1	0
							1	1	1	1

$$x = D_4 + D_5 + D_6 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$z = D_1 + D_3 + D_5 + D_7$$