COEN 212: DIGITAL SYSTEMS DESIGN I
Lecture 7: Common Combinational Logic Circuits (Adders and Multipliers)

Instructor: Dr. Reza Soleymani, Office: EV-5.125,
Telephone: 848-2424 ext.: 4103.

Lecture 7: Objectives of this lecture

- In this lecture, we talk about:
  - Adders.
  - Subtractors.
  - Multipliers and,
  - Magnitude Comaparators.
- In the next lecture, we will talk about:
  - Decoders, Encoders, Multiplexers

Lecture 7:
Reading for this lecture

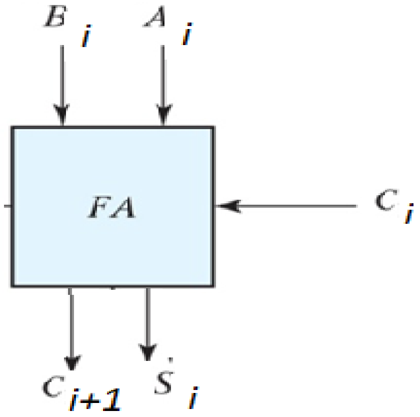
- Digital Design by M. Morris R. Mano and Michael D. Ciletti, 6th Edition, Pearson, 2018:
  - Chapter 4 (4.5 to 4.8)

Lecture 7: Binary Adders

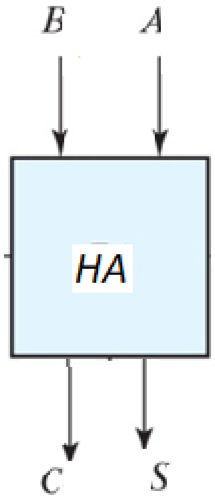
- Binary numbers are added bit by bit.
- To add two bits, we need a Full Adder.
- It takes in two bits and a carry and outputs a sum and a carry:

- A full adder can be implemented either directly or using two half adder. A half adder has two inputs and two outputs.

Full Adder:



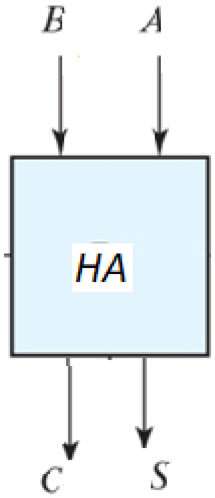
Half Adder:



Lecture 7:
Addition: HA

- Truth table for a half adder is:

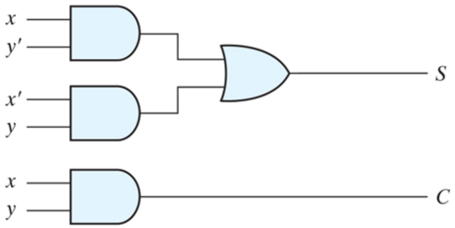
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



- So:

$$S = A'B + AB' = A \oplus B$$
$$C = AB$$

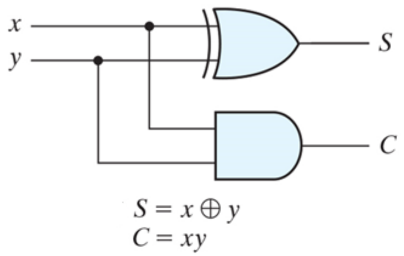
- The implementation is:



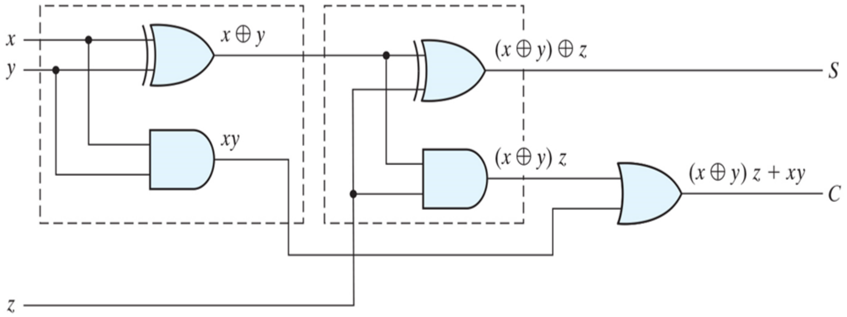
$S = xy' + x'y$
$C = xy$

Lecture 7:
Addition: HA

- Half adder can also be implemented using XOR:



- Implementing a FA using two HA's:



Lecture 7:
Addition: FA implementation uisng HA's

- Half adder can also be implemented using XOR:

$$S_i = A_i \oplus B_i \oplus C_i$$

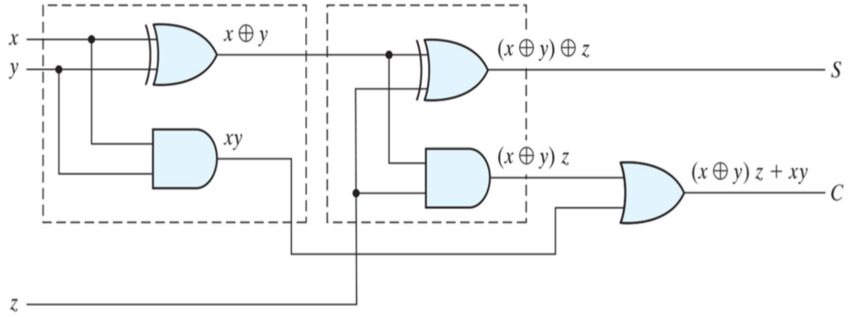
- Also, : $C_{i+1} = A_i B_i + A_i C_i + B_i C_i$

$$= A_i(B_i + C_i) + B_i(A_i + C_i)$$

$$= A_i(B_i + B_i' C_i) + B_i(A_i + A_i' C_i)$$

$$= C_i(A_i B_i' + A_i' B_i) + A_i B_i$$

$$= C_i(A_i \oplus B_i) + A_i B_i$$



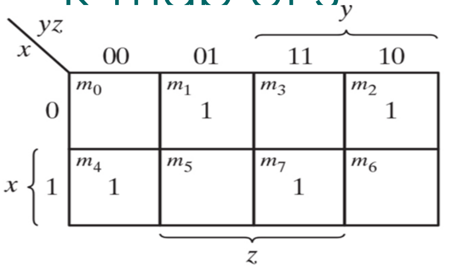
Lecture 7:

Addition: FA direct implementation

- Truth Table of FA:

x	y	z	C	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

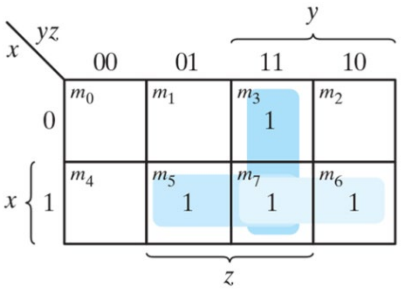
K-map of S



$S = x'y'z + x'yz' + xy'z' + xyz$

$$S = x'y'z + x'yz' + xy'z' + xyz$$

K-map of C

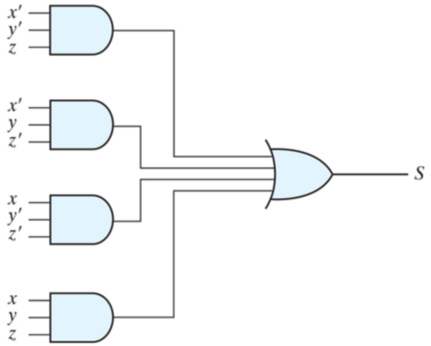


$C = xy + xz + yz$

$$C = xy + xz + yz$$

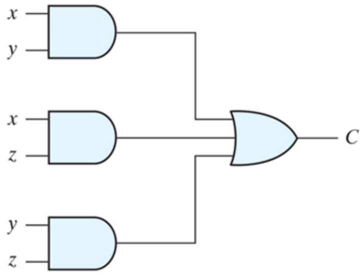
Lecture 7:
Addition: FA direct implementation

- The implementation for S



$$S = x'y'z + x'yz' + xy'z' + xyz$$

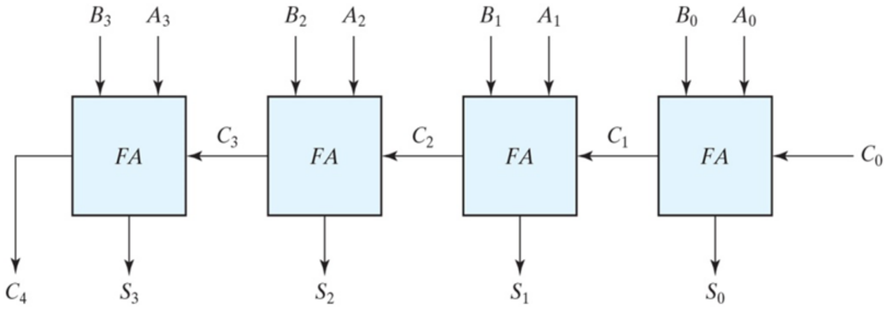
- The implementation for C



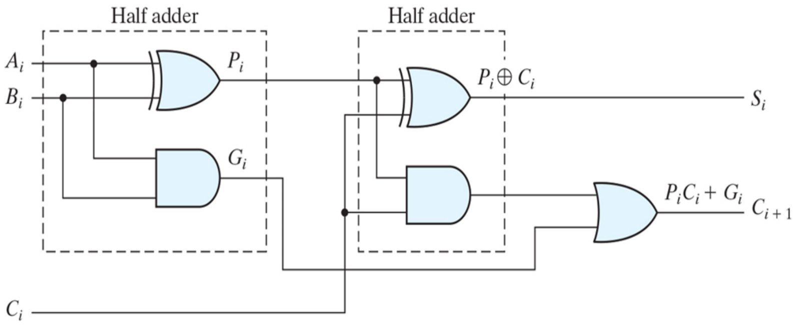
$$C = xy + xz + yz$$

Lecture 7:
Binary adders/carry propagation

- Four-bit adder: Delay $2m = 2 \times 4 = 8$



- Carry Lookahead

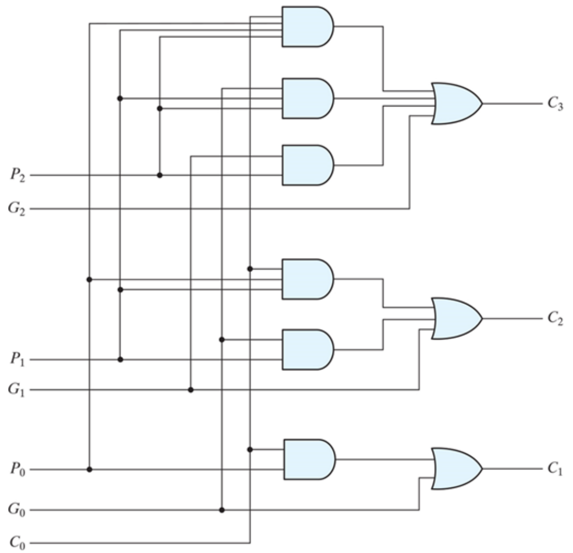


Lecture 7: Binary adders/carry Lookahead

- Note that G_i generates a carry when both the inputs A_i and B_i are 1 regardless of the value of C_i.
- G_i is called a carry generator.
- P_i decides whether a carry will propagate from stage i to stage $i + 1$. It is called a carry propagator.
- Note that P_i and G_i are generated from A_i and B_i in one gate delay.
- Now, let's consider the example of 4-bit adder.
- $C_0 =$ the input carry
- $C_1 = G_0 + P_0C_0$
- $C_2 = G_1 + P_1C_1 = G_1 + P_1(G_0 + P_0C_0) = G_1 + P_1G_0 + P_1P_0C_0$
- $C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

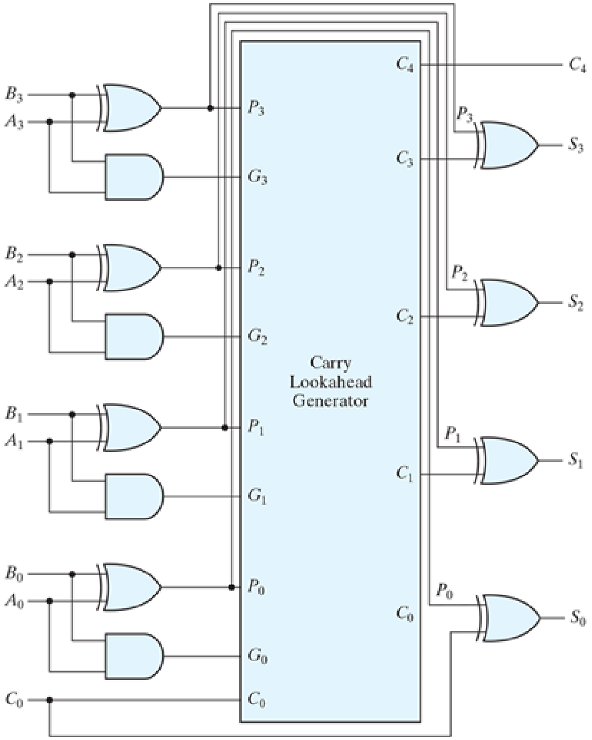
Lecture 7: Binary adders/carry Lookahead

- carry lookahead generator:



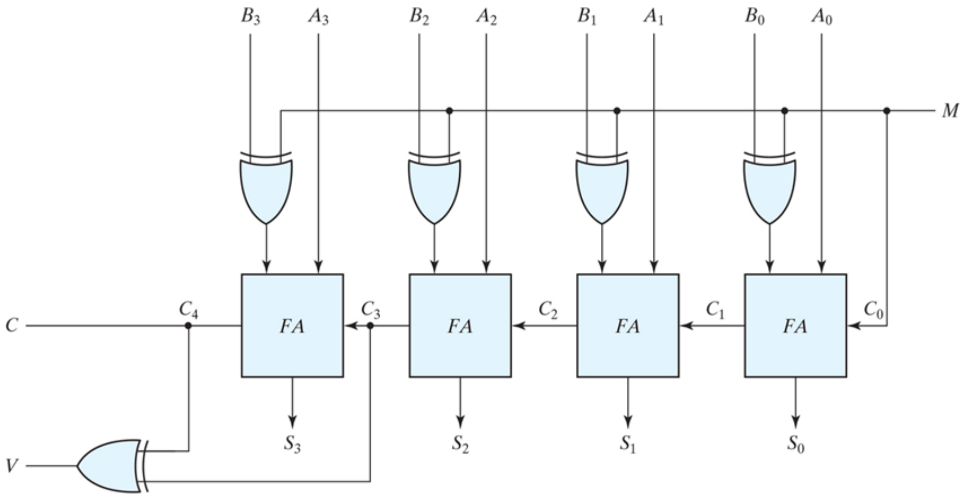
Lecture 7:
Binary adders/carry Lookahead

- carry lookahead generator:



Lecture 7: Subtraction:

- Adder/Subtractor:



- When $M = 0$, we get $A+B$.
- When $M = 1$, we get A-B.
- V=1 flags an overflow.

Lecture 7: Subtraction: Example

- Example: 8-bit adder.
- Numbers are from -127 to +127 (from 11111111 to 01111111).
- The result may fall out of range.
- Example: Adding +60 and +70 = 130 > 127

+60 00111100

+70 01000110

+130 10000010 → -2

sign bit

- Subtraction +70 from -60

-60 11000100

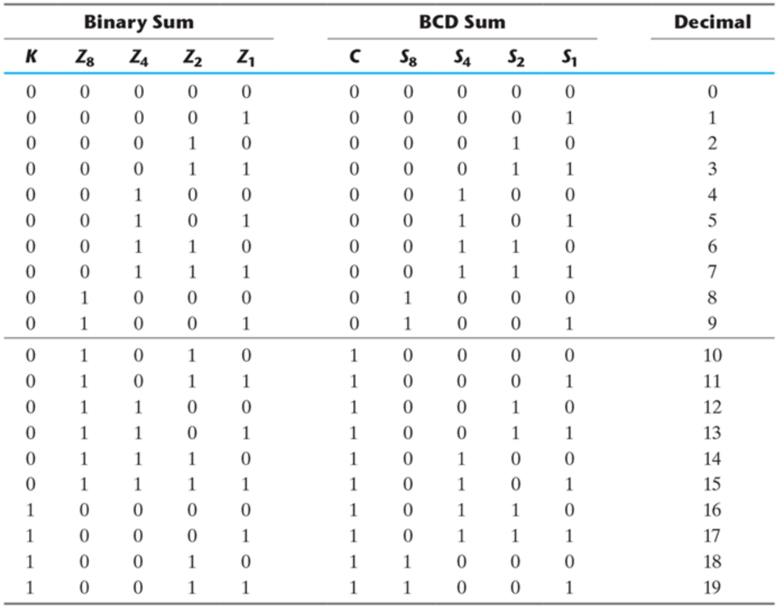
-70 10111010

-130 1 01111110 → +1

overflow

Lecture 7:
BCD Adder

- Adding two decimal digits and a carry, we get a number between 0 and 19.



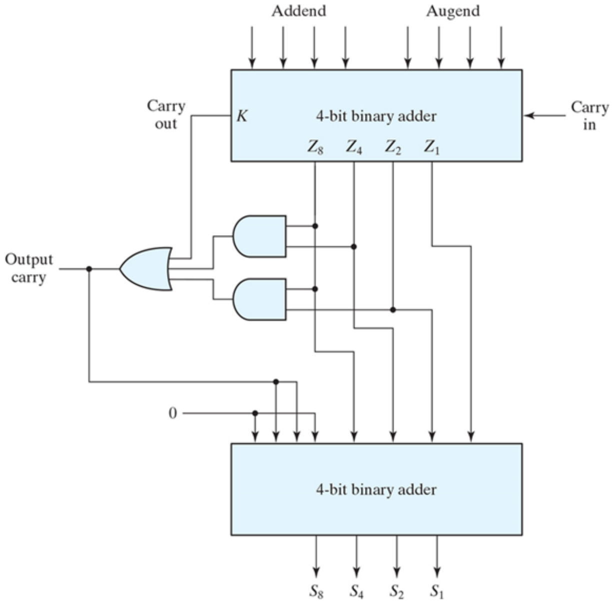
Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

Lecture 7: BCD Adder

- $C = 1$ whenever $K = 1$ or $Z_8 = 1$.
- $C = 0$ when $Z_8 = 1$ and both Z_4 and Z_2 are zero.
- So,

$$C = K + Z_8(Z_2 + Z_4) = K + Z_8Z_4 + Z_8Z_2$$

- When $C = 1$, we need to add 6 (0110) to $Z_8Z_4Z_2Z_1$ to get $S_8S_4S_2S_1$.



Lecture 7: Binary multiplication

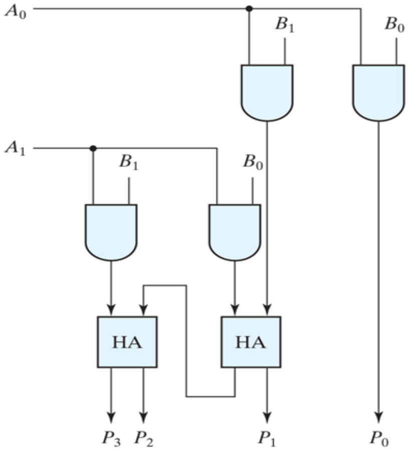
- $C = 1$ whenever $K = 1$ or $Z_8 = 1$.

$$\begin{array}{cccc} & & B_1 & B_0 \\ & & A_1 & A_0 \\ \hline & & A_0B_1 & A_0B_0 \\ & A_1B_1 & A_1B_0 & \\ \hline P_3 & P_2 & P_1 & P_0 \end{array}$$

multiplying B_1B_0 by A_0

multiplying B_1B_0 by A_1 and shifting the result

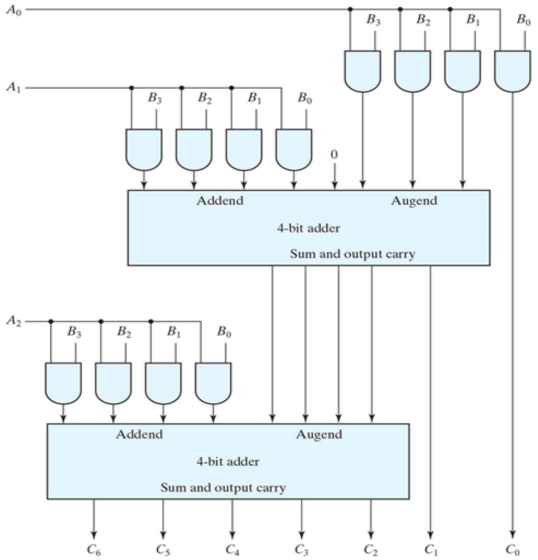
- 2-by-2 multiplier:



Lecture 7: Binary multiplication

- Example: 4-bit by 3-bit multiplier

$$
\begin{array}{ccccccc}
 & & & B_3 & B_2 & B_1 & B_0 \\
 & & & & A_2 & A_1 & A_0 \\
\hline
 & & & A_0B_3 & A_0B_2 & A_0B_1 & A_0B_0 \\
 & & A_1B_3 & A_1B_2 & A_1B_1 & A_1B_0 & \\
 & A_2B_3 & A_2B_2 & A_2B_1 & A_2B_0 & & \\
\hline
C_6 \quad C_5 & & C_4 & C_3 & C_2 & C_1 & C_0
\end{array}
$$



Lecture 7: Magnitude Comparator

- It compares two numbers $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$ and decides whether $A = B, A > B$ or $A < B$.
- $A = B$ if and only if $A_3 = B_3, A_2 = B_2, A_1 = B_1, A_0 = B_0$.
- $A_3 = B_3$ if $X_3 = A_3B_3 + A_3'B_3'$.
- Similarly $X_i = A_iB_i + A_i'B_i'$, $i = 0, 1, 2, 3$ shows $A_i = B_i$.
- So: $(A = B) = X_3X_2X_1X_0$.
- Case of: $A > B$: $A > B$
- If A_3 is equal to 1 and $B_3 = 0$. So, if $A_3B_3' = 1$, then $A > B$.
- If $A_3 = B_3$, i.e., if $X_3 = 1$ and $A_2 = 1$ and $B_2 = 0$, i.e., if $X_3A_2B_2' = 1$.
- if $X_3X_2A_1B_1' = 1$ or $X_3X_2X_1A_0B_0' = 1$.

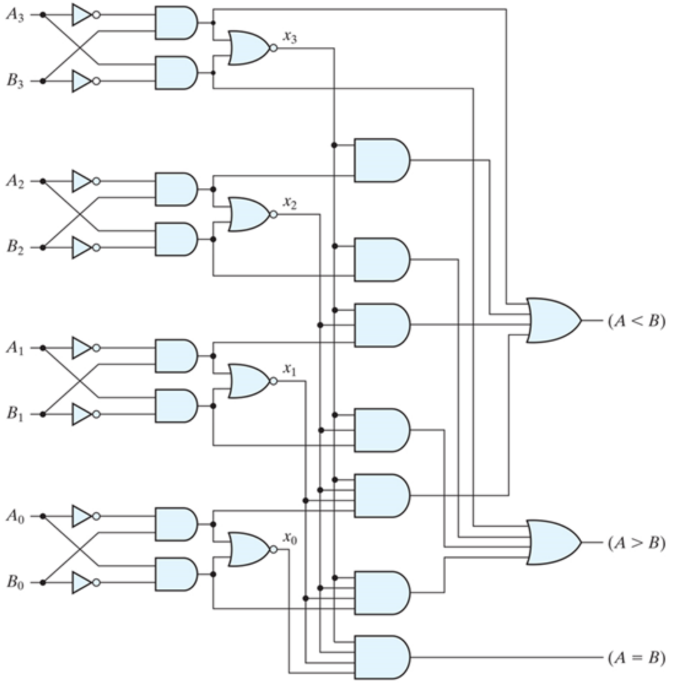
$$(A > B) = A_3B_3' + X_3A_2B_2' + X_3X_2A_1B_1' + X_3X_2X_1A_0B_0'$$

and

$$(A < B) = A_3'B_3 + X_3A_2'B_2 + X_3X_2A_1'B_1 + X_3X_2X_1A_0'B_0$$

Lecture 7: Magnitude Comparator

- 4-bit Magnitude Comparator:



Lecture 7: Knowledge Check

- **Question 1:** To add two 8-bit numbers, the number of Half-Adders needed is:
- a) 8, b) 15, c) 16, d) 17
- **Question 2:** If the gate delay is 10 ns (nano seconds), what would be the delay in adding two 8 bit numbers using carry look ahead?
- a) 40 ns, b) 160 ns, c) 80 ns, d) 20 ns
- **Question 3:**