ENGR 2218 – Digital Logic
Binary Arithmetic

Binary Addition

0 + 0 = Sum of 0 with carry 0
0 + 1 = Sum of 1 with carry 0
1 + 0 = Sum of 1 with carry 0
1 + 1 = Sum of 0 with carry 1

- When there is a carry over, a third bit is added

0 + 0 + 0 = Sum of 0 with carry 0
0 + 0 + 1 = Sum of 1 with carry 0
0 + 1 + 0 = Sum of 1 with carry 0
0 + 1 + 1 = Sum of 0 with carry 1
1 + 0 + 0 = Sum of 1 with carry 0
1 + 0 + 1 = Sum of 0 with carry 1
1 + 1 + 0 = Sum of 0 with carry 1
1 + 1 + 1 = Sum of 1 with carry 1

Binary Subtraction

0 – 0 = 0
1 – 0 = 1
1 – 1 = 0
0 – 1 = Borrow: 10 – 1 = 1

Binary Multiplication

- Works just like regular multiplication

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

Binary Division

- Works just like long division
- Have a remainder if the numbers don't divide evenly

Negative Numbers using the 2's Complement System

- First find the 1's Complement (change the 0's to 1's, and the 1's to 0's)
- Add a binary 1 to the Least Significant Bit (LSB)

Binary Number 0010 1001
1's complement 11010110
+1

2's comp 11010111

2's Complement System

- The Most Significant Bit (MSB) is treated as a negative magnitude bit
  - For 8 bits: -2^7
- If the MSB = 0, then the binary number is positive

- If the MSB = 1, then the number is negative

Binary Addition with Signed Numbers
- When adding the MSB's, there can be a final carry that should be discarded

Binary Subtraction with Signed Numbers
- Take the 2's Complement of the subtracting number (subtrahend) and add

Binary Multiplication with Signed Numbers
- Must maintain partial sums since no more than two binary numbers can be added at a time
- Steps
    1.) Convert any negative numbers to positive using the 2's complement
    2.) Multiply
    3.) If the final answer should be negative, then take the 2's complement of the final product. Otherwise, leave the final product alone.

(+) × (+) = (+)
(+) × (−) = (−)
(−) × (+) = (−)
(−) × (−) = (+)

Binary Division with Signed Numbers
- Works like long division
- Follow the same steps as with multiplication

Hexadecimal Addition
- Letters need to be converted to a decimal number
- If the sum is greater than or equal to 16, then subtract 16 and carry a 1

```
   1
  0238      10      12
+ 04AC
 ------
  06E4
```

$$\begin{array}{r} 8 \\ +12 \\ \hline 20 \\ -16 \\ \hline 4 \end{array} \qquad \begin{array}{r} 10 \\ 3 \\ +1 \\ \hline 14 \end{array}$$

16's Complement
Subtract the hexadecimal number from all F's, and add 1 to the LSB

0238 => 16's comp

$$\begin{array}{r} FFFF \\ -\ 0238 \\ \hline FDC7 \\ +1 \\ \hline \boxed{FDC8} \end{array}$$

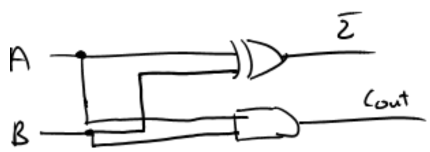
<u>Adders</u>
- Digital Circuit that performs binary addition

<u>Half Adder</u>
- Adds two bits

A	B	Sum (Σ)	Carry (C_{out})
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
		XOR	AND

$\Sigma = A \oplus B$

$C_{out} = AB$



<u>Full Adder</u>
- Includes an input carry

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

Σ

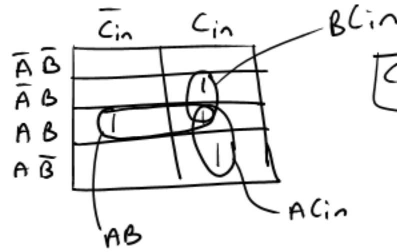
	$\overline{C_{in}}$	C_{in}
$\bar{A}\bar{B}$		1
$\bar{A}B$	1	
AB		1
$A\bar{B}$	1	

$\Sigma = \bar{A}\bar{B}C_{in} + \bar{A}B\overline{C_{in}} + ABC_{in} + A\bar{B}\overline{C_{in}}$

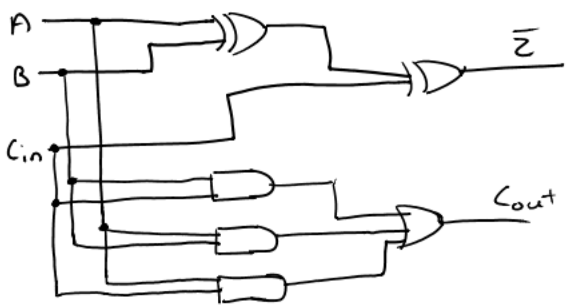
or

$\boxed{\Sigma = (A \oplus B) \oplus C_{in}}$

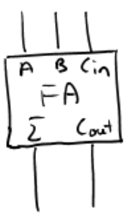
C_{out}



$$\boxed{C_{out} = BC_{in} + AB + AC_{in}}$$

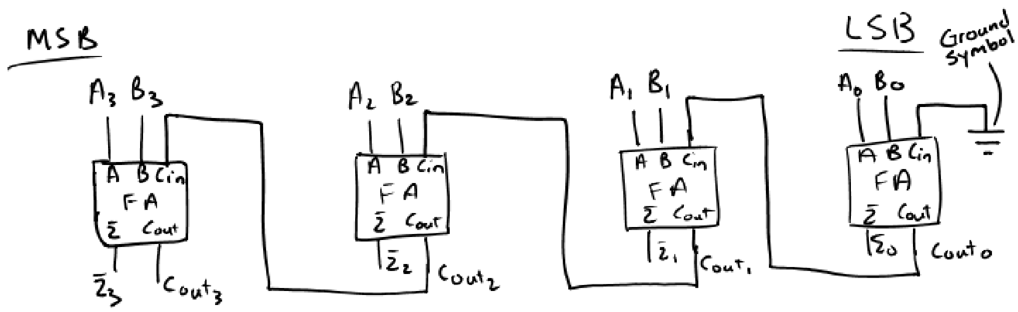


Block Diagram

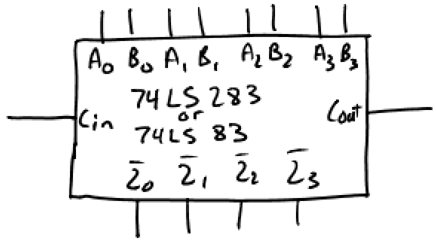


<u>Parallel Adders</u>
- Combine (Cascade) multiple full adders

$$\begin{array}{rcccc}
 & {\scriptstyle C_{out_2}}\,A_3 & {\scriptstyle C_{out_1}}\,A_2 & {\scriptstyle C_{out_0}}\,A_1 & A_0 \\
+ & B_3 & B_2 & B_1 & B_0 \\
\hline
C_{out_3} & \Sigma_3 & \Sigma_2 & \Sigma_1 & \Sigma_0
\end{array}$$



4-bit Parallel Binary Adder



- Can cascade to 4-bit adders to create an 8-bit adder