

ECE212/212s Lecture Notes

Lecture: Chapter 5

Dr. Clay Gloster, Jr.

Representation of Negative Numbers

We will present three systems used for representing negative integers. They are:

- Sign and Magnitude
- One's Complement
- Two's Complement

We will use 4-bit numbers to illustrate these systems, however, the student should be able to represent n-bit numbers using any of these systems.

In all three systems, the leftmost bit is the sign bit.

If the sign bit is 0, then the number is positive.

If the sign bit is 1, then the number is negative.

Sign and Magnitude (Signed Magnitude)

The leftmost bit is the sign and the remaining bits are the magnitude of the number.

0 000	+0
0 001	+1
0 010	+2
0 011	+3
0 100	+4
0 101	+5
0 110	+6
0 111	+7
1 000	-0
1 001	-1
1 010	-2
1 011	-3
1 100	-4
1 101	-5
1 110	-6
1 111	-7

Addition and Subtraction using Sign and Magnitude numbers.

Subtraction is simply addition of negative numbers, therefore, we need only discuss addition in detail.

Case 1: The sign of the two operands is the same.

Simply add the two magnitudes and the sign of the result is the sign of the two operands.

Examples:

$$\begin{array}{r} +3 \quad 0 \ 011 \\ + \quad +2 \quad 0 \ 010 \\ \hline +5 \quad 0 \ 101 \end{array}$$

$$\begin{array}{r} -3 \quad 1 \ 011 \\ + \quad -2 \quad 1 \ 010 \\ \hline -5 \quad 1 \ 101 \end{array}$$

Case 2: The sign of the two operands is different.

Subtract the number with smaller magnitude number from the one with larger magnitude. The sign of the result is the sign of the number with the larger magnitude.

Examples:

$$\begin{array}{r} +3 \quad 0 \ 011 \\ + \quad -2 \quad 1 \ 010 \\ \hline +1 \quad 0 \ 001 \end{array}$$

$$\begin{array}{r} -3 \quad 1 \ 011 \\ + \quad +2 \quad 0 \ 010 \\ \hline -1 \quad 1 \ 001 \end{array}$$

Observation: Any adder circuit for sign and magnitude numbers must include a subtractor and a comparator.

One's Complement

Positive numbers are the same as in sign and magnitude representation.
Negative numbers are the bitwise complement of the positive number.

$$-N = \overline{N}$$

For 4-bit numbers, $-2 = \overline{0010} = 1101$.

Considering 4-bit binary numbers, $-2 = 2^4 - 1 - 2$.

Considering 4-bit binary numbers, $-3 = 2^4 - 1 - 3$.

Considering 4-bit binary numbers, $-N = 2^4 - 1 - N$.

Considering n-bit binary numbers, $-N = 2^n - 1 - N$.

0 000	+0
0 001	+1
0 010	+2
0 011	+3
0 100	+4
0 101	+5
0 110	+6
0 111	+7
1 000	-7
1 001	-6
1 010	-5
1 011	-4
1 100	-3
1 101	-2
1 110	-1
1 111	-0

Addition and Subtraction using One's Complement numbers.

Subtraction is simply addition of negative numbers, therefore, we need only discuss addition in detail.

Case 1: The sign of the two operands is the same.

Simply add the numbers.

If there is a carry, then add one to the result.

(End around carry condition.)

Examples:

$$\begin{array}{r}
 +3 \quad 0 \ 011 \\
 + \quad +2 \quad 0 \ 010 \\
 \hline
 +5 \quad 0 \ 101
 \end{array}$$

$$\begin{array}{r}
 -3 \quad 1 \ 100 \\
 + \quad -2 \quad 1 \ 101 \\
 \hline
 11 \ 001 \\
 + \quad \quad 0 \ 001 \\
 \hline
 + \quad -5 \quad 1 \ 010
 \end{array}$$

Case 2: The sign of the two operands is different.

Simply add the numbers.

If there is a carry, then add one to the result.

(End around carry condition.)

Examples:

$$\begin{array}{r}
+3 \quad 0 \ 011 \\
+ \quad -2 \quad 1 \ 101 \\
\hline
 10 \ 000 \\
 0 \ 001 \\
\hline
+1 \quad 0 \ 001 \\
\\
-3 \quad 1 \ 100 \\
+ \quad +2 \quad 0 \ 010 \\
\hline
-1 \quad 1 \ 111
\end{array}$$

Observation: Subtraction in one's complement requires an adder and circuitry that adds one when there is a carry. (End around carry condition.)

Two's Complement

Positive numbers are the same as in sign and magnitude representation.
 Negative numbers are the bitwise complement of the positive number + 1.

$$-N = \overline{N} + 1$$

For 4-bit numbers, $-2 = \overline{0010} + 1 = 1110$.

Considering 4-bit binary numbers, $-2 = 2^4 - 2$.

Considering 4-bit binary numbers, $-3 = 2^4 - 3$.

Considering 4-bit binary numbers, $-N = 2^4 - N$.

Considering n-bit binary numbers, $-N = 2^n - N$.

0 000	+0
0 001	+1
0 010	+2
0 011	+3
0 100	+4
0 101	+5
0 110	+6
0 111	+7
1 000	-8
1 001	-7
1 010	-6
1 011	-5
1 100	-4
1 101	-3
1 110	-2
1 111	-1

Addition and Subtraction using Two's Complement numbers.

Subtraction is simply addition of negative numbers, therefore, we need only discuss addition in detail.

Case 1: The sign of the two operands is the same.

Simply add the numbers.

Ignore the carry out!!.

Examples:

$$\begin{array}{r} +3 \quad 0 \ 011 \\ + \quad +2 \quad 0 \ 010 \\ \hline +5 \quad 0 \ 101 \end{array}$$

$$\begin{array}{r} -3 \quad 1 \ 101 \\ + \quad -2 \quad 1 \ 110 \\ \hline + \quad -5 \quad 11 \ 011 \end{array}$$

Case 2: The sign of the two operands is different.

Simply add the numbers.

Ignore the carry out.

Examples:

$$\begin{array}{r} +3 \quad 0 \ 011 \\ + \quad -2 \quad 1 \ 110 \\ \hline +1 \quad 1 \ 0 \ 001 \end{array}$$

$$\begin{array}{r} -3 \quad 1 \ 101 \\ + \quad +2 \quad 0 \ 010 \\ \hline -1 \quad 1 \ 111 \end{array}$$

Observation: Subtraction in two's complement requires an adder only.

Overflow Conditions

Overflow occurs during addition of two numbers when the sign of the two operands is the same and the sign of the result is different.

Case 1: If we add two positive numbers and get a negative result, overflow has occurred.

Case 2: If we add two negative numbers and get a positive result, overflow has occurred.

Examples (Two's complement):

$$\begin{array}{r} +3 \quad 0 \ 011 \\ + \quad +6 \quad 0 \ 110 \\ \hline -6 \quad 1 \ 010 \end{array}$$

$$\begin{array}{r} -3 \quad 1 \ 101 \\ + \quad -6 \quad 1 \ 010 \\ \hline +7 \quad 0 \ 111 \end{array}$$

Observation: The carry out and the carry in of the MSB are different. In general overflow occurs when this happens.

Half Adder

The half adder circuit is shown in the figure.

Half Adder

A_i	B_i	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A_i	0	1
B_i	0	1
0	0	1
1	1	0

$$\text{Sum} = \overline{A_i} B_i + A_i \overline{B_i}$$

$$= A_i \oplus B_i$$

A_i	0	1
B_i	0	0
0	0	0
1	0	1

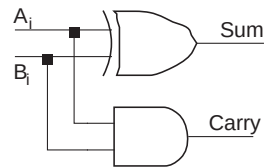
$$\text{Carry} = A_i B_i$$


Figure 1: A half adder

Full Adder

The full adder circuit is shown in the figure.

Full Adder

A	B	CI	S	CO
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

	A B			
CI	00	01	11	10
S	0	1	0	1
	1	0	1	0

	A B			
CI	00	01	11	10
CO	0	0	1	0
	1	1	1	1

$$S = CI \text{ xor } A \text{ xor } B$$

$$CO = B CI + A CI + A B = CI (A + B) + A B$$

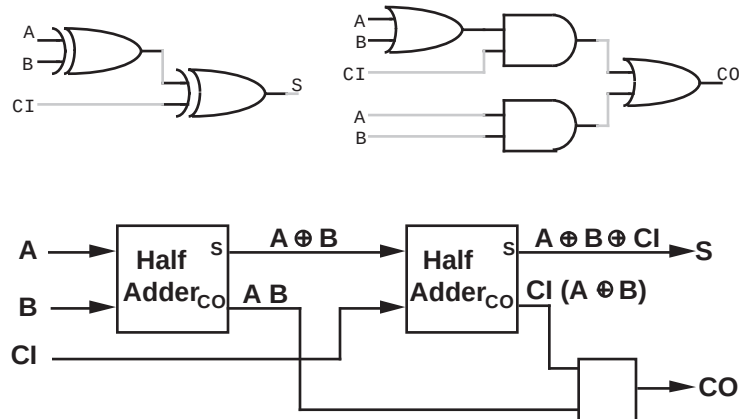
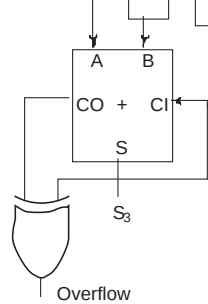


Figure 2: A full adder

Full Address



Adder/Subtractor

$$A_0 \quad B_0 \quad \overline{B_0}$$


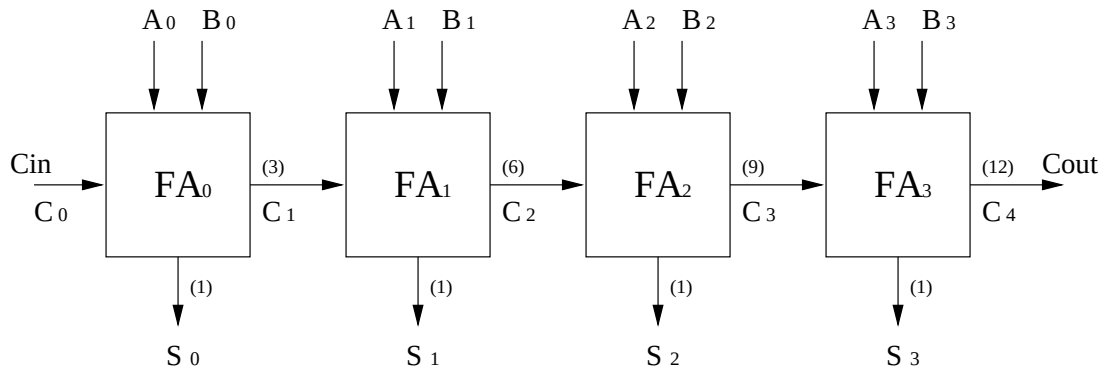


Figure 5: 4-bit Ripple Carry Adder

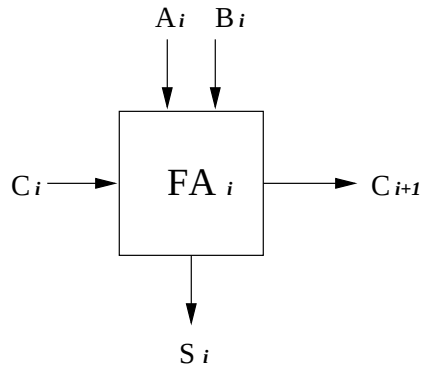


Figure 6: 1-bit Full Adder

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

$$\begin{aligned} C_{i+1} &= B_i \cdot C_i + A_i \cdot C_i + A_i \cdot B_i \\ &= C_i(A_i + B_i) + A_i B_i \end{aligned}$$

If all gate delays are 10ns each, what is the maximum clock speed that we can run this adder?

Worst case delay of adder is 120ns.

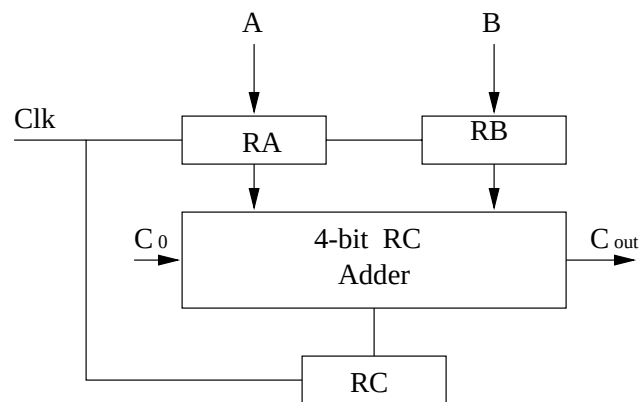
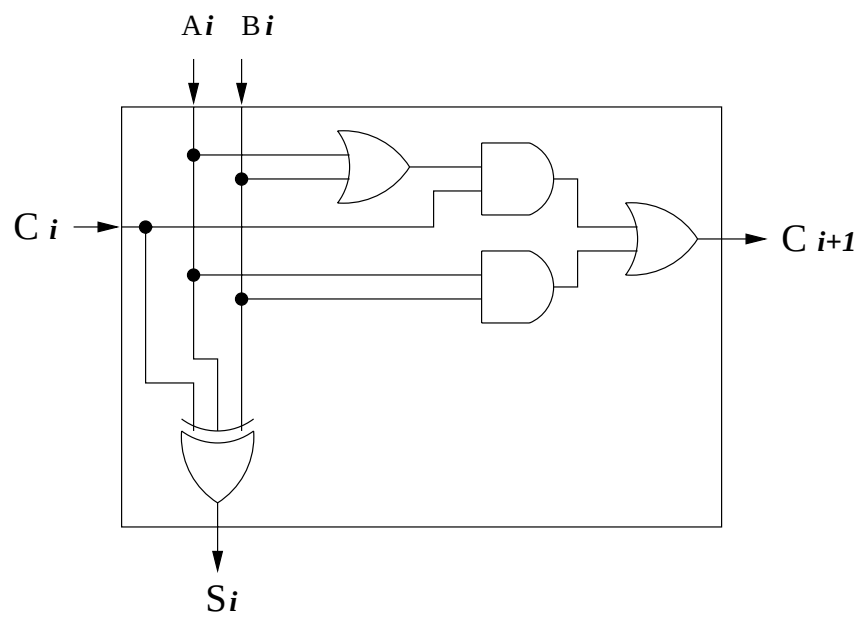
$$\begin{aligned} \text{Min } T &= 120ns \\ f &= \frac{1}{T} = \frac{1}{120 \times 10^{-9}s} = 8.33 \times 10^6 \text{ Hz} \\ &= 8.33 \text{ MHz} \end{aligned}$$

Carry Look Ahead Adders

Problem with RD adders: C_{i+1} depends on C_i from previous full adder.

If the carry value could be determined without having to ripple through all previous stages, then worst case delay could be decreased. (Max clock speed would increase.)

When do we need a carry?



$$\begin{array}{ccccccc}
& C_3 & C_2 & C_1 & C_0 & = & 0 \\
& A_3 & A_2 & A_1 & A_0 & & \\
+ & B_3 & B_2 & B_1 & B_0 & & \\
\hline
& S_3 & S_2 & S_1 & S_0 & &
\end{array}$$

a) When both $A_i + B_i$ are 1, C_{i+1} is 1 regardless of C_i . (Generate Condition) $A_i \cdot B_i$

b) If either A_i or B_i is 1, C_{i+1} is 1 only when C_i is 1. (Propagate condition) $A_i \oplus B_i$

In summary: C_i is 1 if it meets the generate condition or if it meets the propagate condition and C_i was 1.

- $C_{i+1} = G_i + P_i C_i$

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

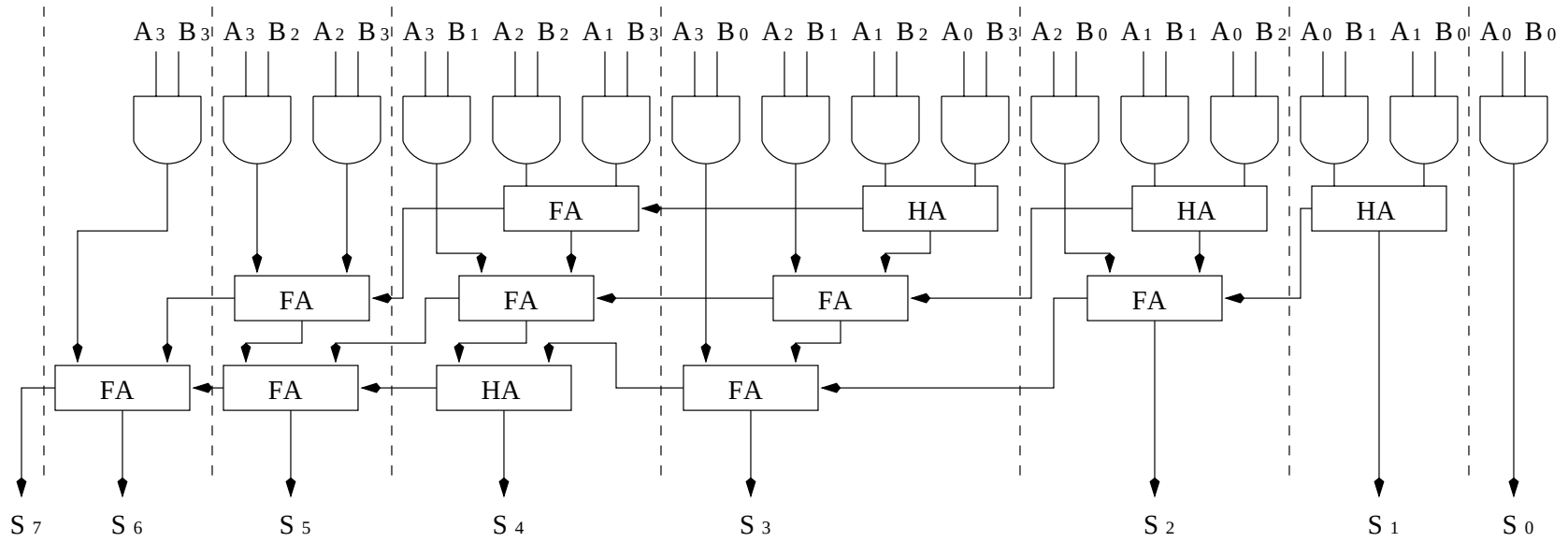
$$C_{i+1} = G_i + P_i C_i$$

$$i = 0 \quad C_1 = G_0 + P_0 C_0$$

$$\begin{aligned}
i = 1 \quad C_2 &= G_1 + P_1 C_1 \\
&= G_1 + P_1 (G_0 + P_0 C_0) \\
&= G_1 + P_1 G_0 + P_1 P_0 C_0
\end{aligned}$$

$$\begin{aligned}
i = 2 \quad C_3 &= G_2 + P_2 C_2 \\
&= G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0) \\
&= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0
\end{aligned}$$

$$\begin{aligned}
i = 3 \quad C_4 &= G_3 + P_3 C_3 \\
&= G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0) \\
&= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0
\end{aligned}$$



See Page 267