Chapter1: Digital Systems and Binary Numbers

Lecture4- Study Signed Numbers, Perform Subtraction using Complements

Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

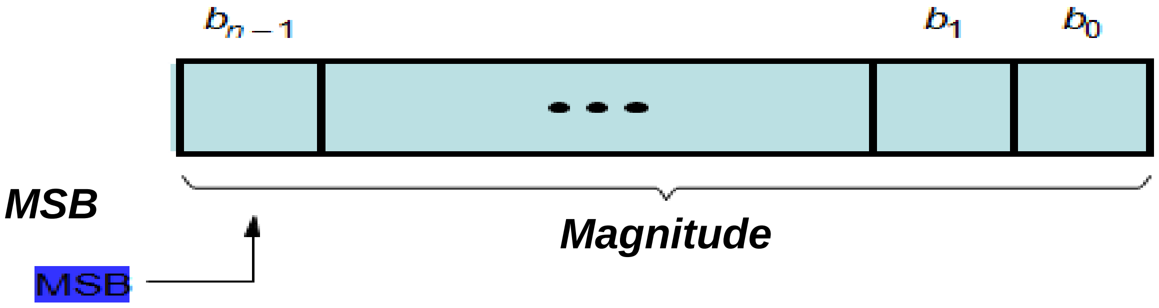
Objectives

- Study Signed Numbers
- Perform Subtraction of Signed Numbers using Complements

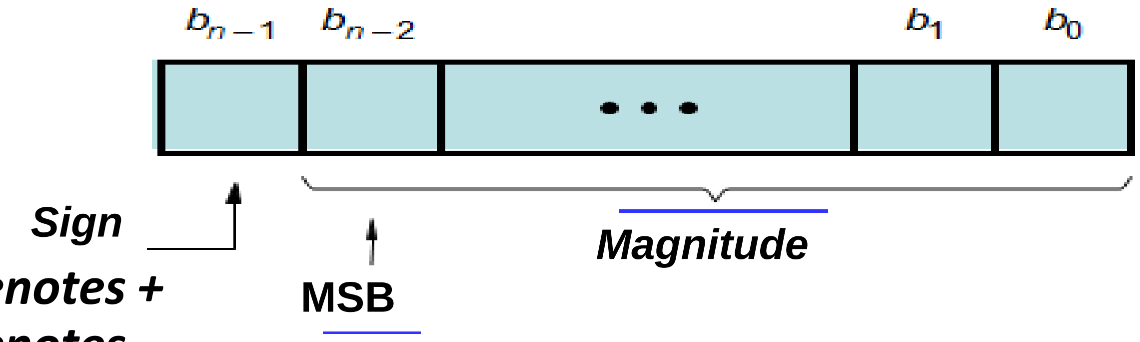
Signed Binary Numbers

- In ordinary arithmetic a negative number is indicated by minus sign and positive number by plus sign. This is not possible in computers, because of hardware limitations since computers must represent everything with binary digits. There are two methods to do this:
  - The *signed magnitude convention* uses the left-most bit to represent the sign (0 for positive and 1 for negative).
  - The *signed complement system* negates a number by taking its complement.
    - It could be either*, 1's complement* representation or *2's complement* representation.

Representations of Negative Numbers



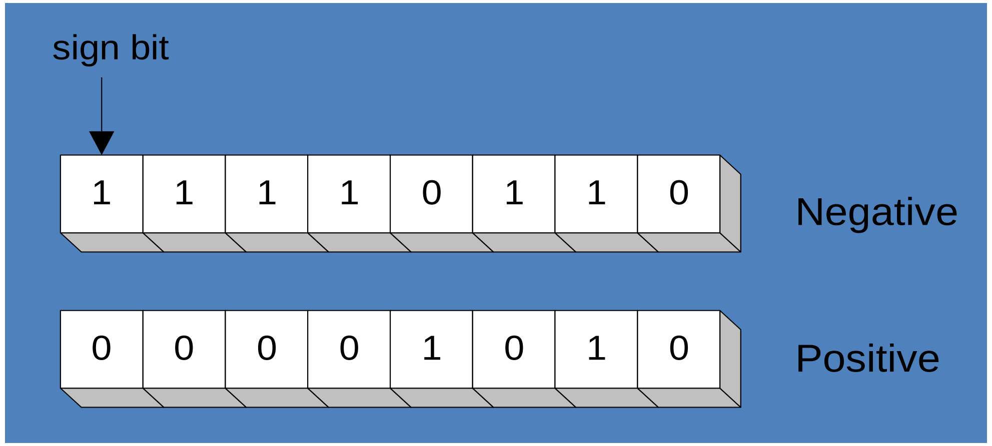
(a) Unsigned number



0 denotes +
1 denotes -

(b) Signed number

Signed Magnitude Convention



- The signed magnitude convention uses the left-most bit to represent the sign (0 for positive and 1 for negative).
  - The user determines whether the number is signed or unsigned
  - If the binary number is signed, then the leftmost bit represents the sign and the rest of the bits represent the number
  - If the binary number is unsigned then the leftmost bit is the most significant bit of the number
  - For example:
    - 01001 can be considered as 9 (unsigned binary) or a +9 because the left most bit is zero.
    - On the other hand, the string of bits 11001 represents binary equivalent of 25 when considered as unsigned number or as –9 when considered as signed number

Signed Complement System

- The Signed Complement System negative number is indicated by its complement (Complement of positive number)
  - Positive numbers always start with 0 (plus), its complement (representing negative number) will always start with 1
  - Signed complement system can use either 1's complement or 2's complement.
  - For example:
    - +9 is represented only as 00001001 but –9 can be represented as:
      - ✓11110110 Signed 1's complement representation
      - ✓11110111 Signed 2's complement representation

Number Representations

- The following is the representation for +11:
  - ➢ 00001011
- The following are different methods for representing -11:
  - ➢ Signed magnitude: 10001011
  - ➢ Signed-1's-complement: 11110100
  - ➢ Signed-2's-complement: 11110101

Signed Binary Numbers

+N	Positive integers (all systems)	-N	Negative integers		
			Sign and magnitude	2's complement N^*	1's complement $\overline{N}$
+0	0000	-0	1000	-	1111
+1	0001	-1	1001	1111	1110
+2	0010	-2	1010	1110	1101
+3	0011	-3	1011	1101	1100
+4	0100	-4	1100	1100	1011
+5	0101	-5	1101	1011	1010
+6	0110	-6	1110	1010	1001
+7	0111	-7	1111	1001	1000
		-8	-	1000	-

Arithmetic Addition (Signed-Magnitude System)

- The addition of two signed binary numbers in the signed-magnitude system follows the rules of ordinary arithmetic
- If the signs are the same, we add the two magnitudes and give the sum the common sign
- If the signs are different, we subtract the smaller magnitude from the larger and give the result the sign of the larger magnitude
- For example:
  - -35+(-12)=-(35+12)=- 47
  - -35+(+12)=-(35-12)=-23
  - +35+(+12)=+(35+12)=+47
  - +35+(-12)=+(35-12)=+23

Arithmetic Addition (Signed 2's Complement system)

- This system doesn't require the comparison of the signs and the magnitudes (as in signed-magnitude system), but only addition.
- The addition of two signed binary numbers with negative numbers represented in signed-2's complement form is obtained from addition of the two numbers, including their sign bits.
- A carry out of the sign-bit position is discarded, provided there is no over flow. Here the left-most bit of the result shows sign.
- If the sum is negative, it will always be in 2's complement form.

2's Complement Addition Rules

Addition of 2's complement Numbers

Case 1	$+3$	0011	
	$+4$	0100	
	$+7$	0111	(correct answer)
Case 2	$+5$	0101	
	$+6$	0110	
		1011	← wrong answer because of **overflow** (+11 requires 5 bits including sign)
Case 3	$+5$	0101	
	-6	1010	
		1111	(correct answer)
Case 4	-5	1011	
	$+6$	0110	
		(1)0001	← correct answer when the carry from the sign bit is ignored (this is *not* an overflow)

An overflow occurs since SUM$\geq 2^{n-1}$

2’s Complement Addition Rules

Addition of 2’s complement Numbers

Case 5

– 3	1101	
– 4	1100	
– 7	(1)1001 ←	correct answer when the last carry is ignored (this is *not* an overflow)

Case 6

– 5	1011	
– 6	1010	
	(1)0101 ←	wrong answer because of overflow (-11 requires 5 bits including sign)

An overflow occurs since SUM >2^{n-1}

Arithmetic Operations with Signed Numbers

Using the signed number notation with negative numbers in 2's complement form simplifies addition and subtraction of signed numbers.

Rules for **addition**: Add the two signed numbers. Discard any final carries. The result is in signed form.

Examples:

$$\begin{array}{r} 00011110 = +30 \\ 00001111 = +15 \\ \hline 00101101 = +45 \end{array} \qquad \begin{array}{r} 00001110 = +14 \\ 11101111 = -17 \\ \hline 11111101 = -3 \end{array} \qquad \begin{array}{r} 11111111 = -1 \\ 11111000 = -8 \\ \hline \cancel{1}\,11110111 = -9 \end{array}$$

Discard carry

Arithmetic Operations with Signed Numbers

Note that if the number of bits required for the answer is exceeded, overflow will occur. This occurs only if both numbers have the same sign. The overflow will be indicated by an incorrect sign bit.

Two examples are:

01000000 = +64	10000001 = –127
01000001 = +65	10000001 = –127
10000001 = ~~- 127~~	00000010 = ~~+2~~

Wrong! The answer is incorrect and the sign bit has changed.

Arithmetic Operations with Signed Numbers

Rules for **subtraction**: 2's complement the subtrahend and add the numbers. Discard any final carries. The result is in signed form.

Repeat the examples done previously, but subtract:

$$\begin{array}{rl} 00011110 & (+30) \\ -\ \underline{00001111} & -(+15) \end{array} \qquad \begin{array}{rl} 00001110 & (+14) \\ -\ \underline{11101111} & -(-17) \end{array} \qquad \begin{array}{rl} 11111111 & (-1) \\ -\ \underline{11111000} & -(-8) \end{array}$$

2's complement subtrahend and add:

$$\begin{array}{rl} 00011110 & = +30 \\ \underline{11110001} & = -15 \\ \cancel{1}00001111 & = +15 \end{array} \qquad \begin{array}{rl} 00001110 & = +14 \\ \underline{00010001} & = +17 \\ 00011111 & = +31 \end{array} \qquad \begin{array}{rl} 11111111 & = -1 \\ \underline{00001000} & = +8 \\ \cancel{1}00000111 & = +7 \end{array}$$

Discard carry

Discard carry

1's Complement Addition Rules

Addition of 1's complement Numbers

Case 3

$+5$		0101	
-6		1001	
-1		1110	(correct answer)

Case 4

		1010	
-5		0110	
$+6$	(1)	0000	
		1	(end-around carry)
		0001	(correct answer, *no* overflow)

Case 5

		1100	
-3		1011	
$+ -4$	(1)	0111	
		1	(end-around carry)
		1000	(correct answer, *no* overflow)

Note: Case 1 and 2 are same for both complement methods

1's Complement Addition Rules

Addition of 1's complement Numbers

Case 6

$$
\begin{array}{rcr}
 & & 1010 \\
-5 & & \underline{1001} \\
\underline{-6} & (1) & 0011 \\
 & \lfloor\!\longrightarrow & \underline{1} \\
 & & 0100
\end{array}
$$

(end-around carry)

(wrong answer because of overflow)

$$Case\,4: \quad -A+B \ \ (\text{where } B > A)$$

$$\overline{A}+B=(2^n-1-A)+B=2^n+(B-A)-1$$

$$Case\,5: \quad -A-B \ \ (A+B<2^{n-1})$$

$$\overline{A}+\overline{B}=(2^n-1-A)+(2^n-1-B)=2^n+[2^n-1-(A+B)]-1$$

Addition Examples

Addition of 1's complement Numbers

$$
\begin{array}{rrl}
 & 11110100 & (-11) \\
 & \underline{11101011} & \underline{+(-20)} \\
(1) & 11011111 & \\
 & \underline{1} & \text{(end-around carry)} \\
 & 11100000 = & -31
\end{array}
$$

Addition of 2's complement Numbers

$$
\begin{array}{rr}
11111000 & (-8) \\
\underline{00010011} & \underline{+19} \\
\cancel{(1)}00001011 & = +11
\end{array}
$$

(end carry discarded)

Arithmetic Subtraction

- Subtraction can be performed by simply converting the equation into an addition formula.
  - Take the 2’s complement of the subtrahend (including the sign bit) and add it to the minuend (including the sign bit)
  - A carry out of the sign bit position is discarded, Provided there is no overflow.
  - Note: Subtraction operation can be changed to an addition operation if the sign of the subtrahend is changed. This is easily done by taking it’s 2’s complement as demonstrated in the following relationship

$$(\pm A) - (+B) = (\pm A) + (-B)$$

$$(\pm A) - (-B) = (\pm A) + (+B)$$

  - For example (–6) – (–13) in binary

(11111010 –11110011)

=11111010+00001101=100000111

=00000111 (+7) Removing the end carry

The End