Lecture 4 – Binary Representation

Chapter 1

Representation of negative numbers

- In ordinary arithmetic, a negative number is indicated by a minus sign and a positive number by a plus sign
- Computers must represent everything with binary digits
- It is customary to represent the sign with a bit placed in the leftmost position of the number
- The convention is to make the sign bit 0 for positive and 1 for negative
- This can be done using:
    1. Signed magnitude representation
    2. Signed complement representation
        1. Signed 1's complement representation
        2. Signed 2's complement representation

Signed-magnitude representation

- In this notation, the number consists of a magnitude and a symbol (+ or -) or a bit (0 or 1) indicating the sign

- This is similar to the representation of signed numbers used in ordinary arithmetic

- Eg: 01001 represents +9, and 11001 represents -9 in signed magnitude representation
  - -9 is obtained from +9 by changing only the sign bit from 0 to 1

- Weird: +0 is represented as 0000 and minus 0 is represented as 1000. So, two representations for zero – inefficient and may cause errors

Signed complement representation

- In digital hardware, it is more convenient to use *signed complement* system, for representing negative numbers
- In this system, a negative number is indicated by its complement
  - signed-complement system negates a number by taking its complement
- Since positive numbers always start with 0 (plus) in the leftmost position (in all representations), it follows that the complement will always start with a 1, indicating a negative number
- In signed-1's-complement, -9 is obtained by taking the 1's complement of all the bits of +9 (01001), including the sign bit, ie, 10110
- The signed-2's-complement representation of -9 is obtained by taking the 2's complement of +9 , including the sign bit, ie 10111

Signed complement representation

- Find signed 2's complement representation in 4 bits:
- +3

  0011
- -7

  $(7)_{10}$ = 0111 => -7 = 1001
- 0

  0000

- -39 (00100111) in 8 bit:
  - 10100111 (signed magnitude)
  - 11011000 (signed 1's complement)
  - 11011001 (signed 2's complement)

Signed complement representation

Find the decimal numbers for the following signed 2's complement representation in 4 bits:

- $(1100)_2$
- $(1111)_2$
- $(0000)_2$
- $(1000)_2$

Interpretations for 4 bit binary numbers

Decimal	Signed-2's Complement	Signed-1's Complement	Signed Magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

Signed addition

- If the numbers are represented in memory in 2’s complement form, the sign of the sum takes care of itself! The result is also in 2’s complement representation
- The sign bit is to be included in the addition and if there is a carry, it is discarded
- Examples in 4-bit signed 2’s complement representation:

$$\begin{array}{rr} +\ 6 & 00000110 \\ +13 & 00001101 \\ \hline +19 & 00010011 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ +13 & 00001101 \\ \hline +\ 7 & 00000111 \end{array}$$

$$\begin{array}{rr} +\ 6 & 00000110 \\ -13 & 11110011 \\ \hline -\ 7 & 11111001 \end{array} \qquad \begin{array}{rr} -\ 6 & 11111010 \\ -13 & 11110011 \\ \hline -19 & 11101101 \end{array}$$

Signed addition

- In order to obtain a correct answer, we must ensure that the result has a sufficient number of bits to accommodate the sum

- If we start with two *n*-bit numbers and the sum occupies *n* + 1 bits, we say that an overflow occurs

$$\begin{array}{rr} 01111101 & 125 \\ +\,00111010 & +\;\;58 \\ \hline 1\underbrace{0110111} & 183 \end{array}$$

Sign incorrect

Magnitude incorrect

Signed subtraction

- Subtraction of two signed binary numbers when negative numbers are in 2’s-complement form is simple and can be stated as follows:
  - 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 is discarded
- This works because: $M - N = M + (-N)$
- Example : 8 – 3 = 8 + (-3)

	00001000	Minuend (+8)
	+ 11111101	2’s complement of subtrahend (−3)
Discard carry ⟶	**1 00000101**	Difference (+5)