Computer Organization & Architecture

Dr. Sonu Lamba

Department of Computer Science and Engineering
The LNM Institute of Information Technology Jaipur

August 6, 2020

Data Representation & Computer Arithmetic

- What is Data Representation?
- Why is required?

Data Representation in Computer Systems

- Representation of a number in a computer system $\rightarrow$ string of bits.
- Various types of number representation techniques for digital number representation
  - Binary number system
  - Octal number system
  - Decimal number system
  - Hexadecimal number system
- Digital Computers use Binary number system to represent all types of information inside the computers.
- Computer memory location merely stores a binary pattern.
- A byte is a group of eight bits
  - It is the smallest possible addressable unit of computer storage
- A word is a contiguous group of bytes.
  - Word sizes (length) of 16, 32, or 64 bits are most common.

Binary Number Representations for Integers

- Two types of Integers:
  - Unsigned Integers: 0/ +ve.
  - Signed Integers: 0/+ve/-ve
- Three representation schemes had been proposed for signed integers:
  1. Sign-Magnitude representation: $-(2^{(k-1)} - 1) to (2^{(k-1)} - 1)$, for k bits.
  2. 1's Complement representation: $-(2^{(k-1)} - 1) to (2^{(k-1)} - 1)$, for k bits.
  3. 2's Complement representation: $-(2^{(k-1)}) to (2^{(k-1)} - 1)$, for k bits.

Signed Number Representation

- Positive values have identical representations in all systems, but negative values have different representations.
- In all the three schemes, the most-significant bit (msb) is called the sign bit.
    - represent the sign of the integer: 0 for positive integers and 1 for negative integers.
- The magnitude of the number is interpreted differently in different schemes.

Example

Decimal	Unsigned	SNM	1's	2's

Example

Addition of Unsigned Integers

- Add bit pairs starting from the low-order (right) end of the bit vectors, propagating carries toward the high-order (left) end.

$$
\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array}
\qquad
\begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array}
\qquad
\begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array}
\qquad
\begin{array}{r} 1 \\ +\ 1 \\ \hline 1\,0 \end{array}
$$

Carry-out

- Example:

- Note: Only overflow can occur
- Can we subtract unsigned number?

Addition of Signed Integers

- Add the values and discard any carry-out bit.
- The sum will be the algebraically correct value if within the range.

```
  0010 (+2)      0100 (+4)      (+7)
+ 0011 (+3)    + 1010 (-6)    + (-3)
-----------    -----------    ------
```

```
  1011 (-5)      +7
+ 1110 (-2)      +6
-----------    ----
```

1000 (-8)

1001 (-7)

(-1)

+ (-6)

Sign Extension: ?

Subtraction of Signed Integers

- To perform X - Y , form the 2's-complement of Y , then add it to X using the add rule.

(a)

$$\begin{array}{rl} 1101 & (-3) \\ -\,1001 & (-7) \\ \hline \end{array} \Rightarrow \begin{array}{rl} 1101 & \\ +\,0111 & \\ \hline 0100 & (+4) \end{array}$$

(b)

$$\begin{array}{rl} 0110 & (+6) \\ -\,0011 & (+3) \\ \hline \end{array} \Rightarrow$$

(c)

$$\begin{array}{rl} 1001 & (-7) \\ -\,0001 & (+1) \\ \hline \end{array}$$

ADDITION :-

If $(+A) + (+B) = -C$

$(-A) + (-B) = +C$

(a)

$$\begin{array}{rr} & -7 \\ + & -6 \\ \hline \end{array} \qquad \begin{array}{r} 1001 \\ 1010 \\ \hline \end{array}$$

(b)

$$\begin{array}{rr} & (+7) \\ + & (+4) \\ \hline \end{array}$$

Overflow in Subtraction :-

if $(+A) - (-B) = -C$

$(-A) - (+B) = +C$

Ex: $(+7)$ 0111

$-(-6)$ 1010

Booth's Multiplication Algorithm