Lecture : 14

Introduction to Computing

Instructor Name : Aasim Ali

Punjab University College of Information Technology

Important

- Before starting the lecture:
  - You must remain obedient and respect giving towards your TAs
  - They are like next-to-teacher for students
  - You must call them by adding some respectful titles, instead of calling them by Name only
  - You may be knowing much more even than me, but the relationship that Allah has made between us is that of teacher-student, and that bounds you to respect me;
    - same applies to TA-student relationship

Binary Arithmetic

Binary Number System

- Binary system is used internally by almost all modern computers and computer-based devices such as mobile phones, because of its straightforward implementation in digital electronic circuitry using logic gates.
- Modern computer uses Logic gates to implement Boolean functions. And using logic gates computer perform addition, subtraction, multiplication, division and many other mathematical operations.

Binary Arithmetic

Binary Arithmetic

- Arithmetic in Binary system is like arithmetic in other numeral system. Addition, subtraction, multiplication, and division can be performed on binary numerals.

Addition:

- The simplest arithmetic operation in binary is addition. Adding two "1" digits produces a digit "0", while 1 will have to be added to carry. This is similar to what happens in decimal when certain single-digit numbers are added together; if the result equals or exceeds the value of the radix (10), the digit to the left is incremented.

Binary Arithmetic

- For example we have two numbers 1101 and 10111. Addition of binary numbers will be:

```
 1 1 1 1 1      (carried digits)
   0 1 1 0 1
 + 1 0 1 1 1
 ----------------
 =1 0 0 1 0 0 = 36
```

Addition Table:

	0	1
0	0	1
1	1	10

Binary Arithmetic

Multiplication:

- Multiplication in binary is similar to its decimal counterpart. Two numbers A and B can be multiplied by partial products: for each digit in B, the product of that digit in A is calculated and written on a new line, shifted leftward so that its rightmost digit lines up with the digit in B that was used. The sum of all these partial products gives the final result.

Multiplication Table:

	0	1
0	0	0
1	0	1

```
      1 0 1 1    (A)
    × 1 0 1 0    (B)
    -----------
      0 0 0 0   ← B is zero
+   1 0 1 1     ← B is one
+  0 0 0 0
+ 1 0 1 1
-------------------
=1 1 0 1 1 1 0
```

Binary Arithmetic

Subtraction:

- Subtraction works in much the same way as addition of two binary numbers. Subtracting a "1" digit from a "0" digit produces the digit "1", while 1 will have to be subtracted from the next column. This is known as borrowing. The principle is the same as for carrying.
- For example:

```
   *   * * *    (starred columns are borrowed from)
  1 1 0 1 1 1 0
 −   1 0 1 1 1
-------------------
 = 1 0 1 0 1 1 1
```


Binary Arithmetic

Division:

- Division in binary is similar to its decimal counterpart. Here, the divisor is 101, while the dividend is 11001. The procedure is the same as that of decimal long division; here, the divisor 101 goes into the first three digits 110 of the dividend one time, so a "1" is written on the top line. This result is multiplied by the divisor, and subtracted from the first three digits of the dividend; the next digit (a "1") is included to obtain a new three-digit sequence:

```
        1 0 1
      ________
1 0 1 | 1 1 0 0 1
      − 1 0 1
      -------
        0 1 0
      − 0 0 0
      -------
          1 0 1
        − 1 0 1
        -------
              0
```


Binary Arithmetic

One's Complement (also known, **inversion** of a binary number**):**

- The ones' complement of a binary number is defined as the value obtained by inverting all the bits in the binary representation of the number (swapping 0's for 1's and vice-versa). The ones' complement of the number then behaves like the negative of the original number in most arithmetic operations
- For example:

	Number	Negation
1	0001	1110
2	0010	1101
3	0011	1100
4	0100	1011
5	0101	1010
6	0110	1001
7	0111	1000

Binary Arithmetic

Two's Complement
(Computer's way of storing **negative integer**):

- The two's complement of an N-bit number is defined as the complement with respect to 2^N, in other words the result of subtracting the number from 2^N. This is also equivalent to taking the ones' complement and then adding one, since the sum of a number and its ones' complement is all 1 bits. The two's complement of a number behaves like the negative of the original number in most arithmetic.

- For example in 4 bit notation, 2's complement of 5 is:
  5* (* represents 2's complement) = 1011
  Inverting bits and adding 1: 1010 + 1 = 1011

Binary Arithmetic

Subtraction using Two's Complement:

- A number can be positive or negative. Computer differentiate between numbers by using its most significant bit (left most).
- For example 12 in 8-bit format is 0000 1100. Now if it is -12 then computer would store this number in the form of 2's complement.
- As we learned how to find 2's complement of a number. 2's complement of 12 in 8-bits will be: 1111 0100.
- Left most bit is 1, which shows that number is negative. If left most bit is 0, then it shows that number is positive.

Binary Arithmetic

Subtraction using Two's Complement:

- Subtraction can be done using 2's complement. For example we have to find value of 29 – 12 in 8-bit binary.
- Then we can this expression as:
  29 + (-12). This shows subtraction of a number is equals to addition of the 2's complement in that number.

- Try the same method on the following:
  5 – 12 (in 8-bit system)
  5 – 4 (in 3-bit system)
  4 – 5 (in 3-bit system)

Q: Evaluate 29-12 using two's complement:

```
29   =   0001 1101
12*  =   1111 0100
        -----------------
        1 0001 0001
```

In 8-bit system the 9th bit (current left most bit) is overflow, and cannot be stored, hence ignored.

Remaining number is 00010001_2 which is 17_{10}

Why using Two's complement?

- Allowing –ve values reduces the value range of a bit-set (Number of combinations remains unchanged, but the value ranges differ)
- Do we have negative numbers on speedometer?
- Without negative: 8-bits allow 0 to 255 (256 **unsigned** values)
- With negative: 8-bits are partitioned as (1+7) bits, allowing half space for negative values (128 **signed** [+/-] values)
- One's complement has problem of having two [+/-] combinations for zero [00000000/11111111], which is loss of storage capacity, and an invalid proposition:
  -127, -126, -125, … , -2, -1, -0, 0, 1, 2, …, 125, 126, 127
- Two's complement uses all combinations for distinct values:
  -128, -127, -126, … , -2, -1, 0, 1, 2, … , 126, 127