Digital System Design

Lecture 6

Binary System Operations and Representation of Negative Numbers

Objective:

1. **Binary system operations.**
2. **Representation of negative numbers.**
3. **Two's complement addition and subtraction.**
4. **One's complement addition and subtraction.**

1) Binary system operations

a) Binary Addition:

Example:-

	1	1	1	0	1		← carries
	1	1	1	1	0	1	
+							
	1	0	1	1	0	1	
1	1	0	1	0	1	0	

Addition table

	Sum	Carry
0 + 0 =	0	0
0 + 1 =	1	0
1 + 0 =	1	0
1 + 1 =	0	1

b) Binary subtraction:

Example:-

	B	borrow ↔	001111100
First number →	X	229	11100101
	-		-
Second number →	Y	46	00101110
	x-y = 183		10110111

⇩

Subtraction table

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = 1 (and the borrow =1)

After the first borrow, the new subtraction for this column is 0-1, so we must borrow again.

Must borrow 1, yielding the new subtraction 10-1 = 1

			0	10	1	1	10	10	
X	229	1	1	1	0	0	1	0	1
-									
Y	46	0	0	1	0	1	1	1	0
x-y	183	1	0	1	1	0	1	1	1

The borrow goes through three columns to reach a borrowable 1.

100 = 011 (the modified bits), + 1 (the borrow)

c) Binary multiplication:

Example:-

				1	0	1	1	1
			*		1	0	1	0
				0	0	0	0	0
	+		1	0	1	1	1	
		0	0	0	0	0		
	1	0	1	1	1			
	1	1	1	0	0	1	1	0 (result)

Multiplication table

0 * 0 = 0

0 * 1 = 0

1 * 0 = 0

1 * 1 = 1

2) Representation of negative numbers

- There are many ways to represent negative numbers:
  - ✓ **Signed-magnitude system.**
  - ✓ **Complement number systems.**
    - **Signed-magnitude representation:**
  - In signed-magnitude system, the number consists of a ***magnitude*** and a ***symbol indicating whether the magnitude is positive or negative***.
  - In binary system: ***extra bit position to represent the sign*** (sign bit): (***MSB***) is used.

Sign bit: 0 = plus, 1 = minus.

Example:-

$0\,1010101_2 = +85_{10}$, $11010101_2 = -85_{10}$

Sign bit   magnitude

$01111111_2 = +127_{10}$ $11111111_2 = -127_{10}$

$00000000_2 = +0_{10}$ $10000000_2 = -0_{10}$

- **Complement number systems:**

➢ Complement number system negates a number by talking its ***complement*** as defined by the system.

➢ There are two complement number systems that can be used:
  - ***Two's complement system***
  - ***One's complement system.***

Two's complement system:

➢ The two's complement of an **n-digit** number **D** is obtained by:

Subtracting the number from $\mathbf{r^n}$

$$r^n - D$$

  - **r-** The base of the system.

➢ This can be accomplished by ***complementing the individual digits of*** D, ***and adding 1 to the result.***
  - ➢ In decimal system, it's called the **10's complement**.
  - ➢ For binary numbers, it's called **two's complement**.
  - ➢ The ***MSB*** of a number in this system is used as the ***sign bit***.

Example 1:

$17_{10} = 00010001_2$

⇓ Complement bits

11101110

\+ 1

$11101111 = -17_{10}$

Example 2:

$-99_{10} = 10011101_2$

⇓ complement bits

01100010

\+ 1

$01100011_2 = 99_{10}$

Example 3:

$0_{10} = 00000000_2$

⇓

11111111

\+ 1

X (1) $00000000_2 = 0_{10}$

Important Note:
The number 0 has one representation using two's complement.

One's complement representation:

- In one's complement representation the complement of an **n-digit** number **D** is obtained by: ***Subtracting the number from*** $\mathbf{r^n - 1}$

$$(r^n - 1) - D$$

- This can be accomplished by ***complementing the individual digits of*** D, ***without adding 1*** as in two's complement systems.
  - In decimal system it's called **9's complement**.
  - In binary system it's called **1's complement.**

Example 1

$$\mathbf{17_{10} = 00010001_2}$$

⇩

$$\mathbf{11101110_2 = 17_{10}}$$

Example 2

$$\mathbf{-99_{10} = 10011100_2}$$

⇩

$$\mathbf{01100011_2 = 99_{10}}$$

in one's complement :

- **MSB is used as sign digit.**
- **The number 0 has two representations using one's complement:**

00000000	positive zero.
11111111	negative zero.

3) Two's complement addition and subtraction

Example 1: ***adding two positive numbers: +15 +27***

	0	0000	1111	
+	0	0001	1011	
	0	0010	1010	$= (42)_{10}$ → (okay)

↓

Sign bit

Example 2: ***adding two positive numbers with overflow: +78 +85***

$+78 + 85 = (163)_{10}$ (we need 8 bits +1 for sign bit)

	0	1	0	0	1	1	1	0
+	0	1	0	1	0	1	0	1
	١	0	1	0	0	0	1	1

Sign or overflow?

Example 3 ***43 -25:***

	00101011
+	11100111
1	00010010

Ignore

Example 4 ***25- 43:***

	0	0011 001
+	1	1010 101
	1	1101 110

Example 5 ***-15 -27:***

	1	1110001
+	1	1100101
1	1	1010110

Ignore

4) One's complement addition and subtraction

- **Note:** *the same as two's complement addition and subtraction, except that the carry value must be added to the result in LSB bit.*

Example 1: 43-25:

```
            00101011
   +        11100110
          ____________
(Carry)  100010001
                   1   (carry added)
          ____________
            00010010
```

Example 2: 25- 43:

```
            00011001
   +        11010100
          ____________
            11101101
```