Number Representation Tutorial Session – **2** (and answers)

1. Give the 1's complement of each binary number:
   a. 00011010 b. 11110111 c. 10001101

   Give using three methods, the 2's complement of each of the binary numbers:
   a. 01010 b. 11100 c. 10001 d.10101.01

Solution:

1's complement (00011010) =11100101; 1's complement (11110111) =00001000;
1's complement (10001101) =01110010
2's complement (01010) = 100000 - 01010 = 10110;
2's complement (01010) = 1's complement (01010) + 1 = 10101 + 1 = 10110
2's complement (11100) = 100000 - 11100 = 00100;
2's complement (11100) = 1's complement (11100) + 1 = 00011 + 1 = 00100
2's complement (10001) = 100000 - 10001 = 01111;
2's complement (10001) = 1's complement (10001) + 1 = 01110 + 1 = 01111
2's complement (10101,01) = 100000,00 - 10101,01 = 01010,11;
2's complement (10101,01) = 1's complement (10101,01) + 0,01 = 01010,10 + 0,01 = 01010,11

2. Give the decimal value of the signed binary number 10010101, expressed in 8-bit sign-magnitude representation.
   Give the decimal value of the signed binary numbers 010101 and 110101 expressed in 6-bit two's complement representation.

Solution :

- 8-bit sign-magnitude representation of the signed binary number 10010101 = -21 ;
- 6-bit 2's complement signed representation of the signed binary number 010101 = +21 ;
- 6-bit 2's complement signed representation of the signed binary number 110101
  = - -(110101) = - 2's complement (110101) = -(001011) = -11

3. Assume numbers are represented in 8-bit twos complement representation. Show the calculation of the following:
   a. 6 + 13 b. -6 + 13 c. 6 – 13 d. -6 – 13 s.

Solution :

+6	00000110	−6	11111010	+6	00000110	−6	11111010
+13	00001101	+13	00001101	−13	11110011	−13	11110011
+19	00010011	+7	00000111	−7	11111001	−19	11101101

4. Give the value of the following number in 8-bit 1's complement representation; 8-bit 2's complement signed representation and 8-bit sign-magnitude representation.
   a. -32 b. +128 c. -128 d. +127

   Give the decimal value of the signed number $(B7)_{16}$ expressed in 8-bit two's complement representation.

Solution :

Number	**8-bit sign-magnitude representation**	**8-bit 2's complement signed representation**
-32	10100000	11100000
+128	No representation	No representation
-128	No representation	10000000

$(B7)_{16}$ = 10110111 =- - (10110111) = - two's complement (10110111) = - (01001001) = -73

5. Add the following using 2's complement representation in 8-bit register. Also check overflow/underflow.
   a. 15 − 6 b. 16 −24 c. -5 - 9 d. 125 + 58 e. - 62 – 89

   Perform, in signed binary using 2's complement notation, the operation: 0011 x 1011, check the result by performing the operation in decimal. Conclusion.

Solution:

* 15-6=15+Cà2(6)=00001111+Cà2(00000110)=00001111+11111010

	1	1	1	1	1	1	1			
		0	0	0	0	1	1	1	1	
+		1	1	1	1	1	0	1	0	
=	1	0	0	0	0	1	0	0	1	= +9

Retenue à ignorer

* 16-24=16+Cà2(24)=00010000+Cà2(00011000)=00010000+11101000

	0	0	0	1	0	0	0	0	
+	1	1	1	0	1	0	0	0	
=	1	1	1	1	1	0	0	0	= -Cà2(11111000) = -00001000 = -8

* -5-9 = Cà2(5)+Cà2(9) = Cà2(00000101)+Cà2(00001001) = 11111011+11110111

	1	1	1	1	1	1	1	1		
		1	1	1	1	1	0	1	1	
+		1	1	1	1	0	1	1	1	
=	1	1	1	1	1	0	0	1	0	= -Cà2(11110010) = -00001110 = -14

Retenue à ignorer

* 125+58 = 01111101+00111010

	1	1	1	1					
	0	1	1	1	1	1	0	1	
+	0	0	1	1	1	0	1	0	
=	1	0	1	1	0	1	1	1	Dépassement de capacité (Overflow)

* -62-89 = Cà2(62)+Cà2(89) = Cà2(00111110)+Cà2(01011001) = 11000010+10100111

	1					1	1			
		1	1	0	0	0	0	1	0	
+		1	0	1	0	0	1	1	1	
=	1	0	1	1	0	1	0	0	1	Dépassement de capacité (Overflow)

Retenue à ignorer ;

* 0011 x 1011 = Cà2 (0011 x 0101)

					0	1	0	1	
x					0	0	1	1	
					0	1	0	1	
+				0	1	0	1	.	
=	0	0	0	0	1	1	1	1	= 15

0011 x 1011 = Cà2 (0011 x 0101) = Cà2(00001111) = 11110001

In decimal: 3 x (-5) = -(3 x 5) = -15

Conclusion: Multiplication is performed between absolute values, the result is final if the operands have the same sign. If the operands have different signs, the result is complemented by 2.

6. Give the value of the following number in 8-bit 1's complement representation; 8-bit 2's complement signed representation and 8-bit sign-magnitude representation.

   a. +88 b. -88 c. -127 d. +127

Solution :

Number	**+88**	**-88**	**-127**	**+127**
8-bit 1's complement representation of	01011000	10100111	10000000	01111111
8-bit 2's complement signed representation of	01011000	10101000	10000001	01111111
8-bit sign-magnitude representation of	01011000	11011000	11111111	01111111

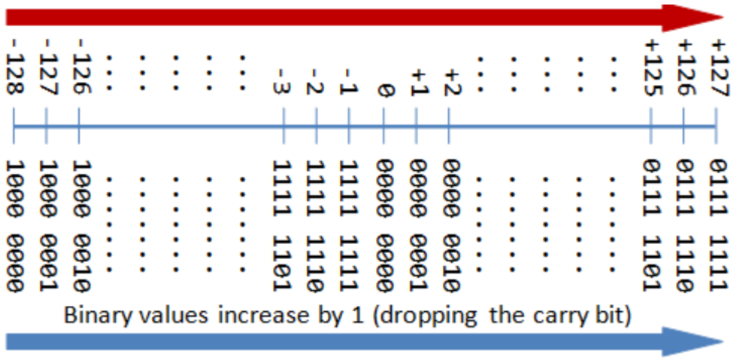
7. Add the following using 2’s complement representation in 8-bit register. Also check overflow/underflow.

   b. +45+(−65) b. −27+(−101) c. +27+101 d. −103+(−69)

Solution:

Detecting Overflow/Underflow

Carry Bit	Sign Bit	Status
0	0	No Overflow/Underflow
1	1	
0	1	Overflow
1	0	Underflow



2's Complement Representation

a. +45+(−65)

Represent 45 in binary → 00101101

Represent −65 by 2's complement → 11101100

0	0	1	1	1	1	1	1		
	0	0	1	0	1	1	0	1	45
	1	0	1	1	1	1	1	1	-65
	1	**1**	**1**	**0**	**1**	**1**	**0**	**0**	-20

Carry into Sign-Bit = 0

Carry out of Sign-Bit = 0

Therefore, no overflow

b. −27+(−101)

Represent −27 by 2's complement → 11100101

Represent −101 by 2's complement → 10011011

1	1	1	1	1	1	1	1		
	1	1	1	0	0	1	0	1	-27
	1	0	0	1	1	0	1	1	-101
	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	-128

Carry into Sign-Bit = 1

Carry out of Sign-Bit = 1

Therefore, no overflow

c. +27+101

Represent 27 in binary → 00011011

Represent 101 in binary → 01100101

0	1	1	1	1	1	1	1	1	
	0	0	0	1	1	0	1	1	27
	0	1	1	0	0	1	0	1	101
	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	~~128~~

Carry into Sign-Bit = 1

Carry out of Sign-Bit = 0

Therefore, overflow

d. −103+(−69)

Represent −103 by 2's complement → 10011001

Represent −69 by 2's complement → 10111011

1	0	1	1	1		1	1		
	1	0	0	1	1	0	0	1	-103
	1	0	1	1	1	0	1	1	-69
	0	**1**	**0**	**1**	**0**	**1**	**0**	**0**	~~-172~~

Carry into Sign-Bit = 0

Carry out of Sign-Bit = 1

Therefore, underflow