classmate

Date
Page

25th June '2020. Unit : 2

Representation of Data and Register Transfer with Microoperations.

* **Number System :-** Electronic and Digital systems may use a variety of different number systems.

Ex.

Decimal, Hexadecimal, Octal, Binary.

→ A number system of base, or radix, r is a system that uses distinct symbols for r digits. Numbers are represented by a string of digit symbols.

→ A number N in base or radix b can be written as:

$(N)_b = d_{n-1}\, d_{n-2} \ldots d_1\, d_0 \,.\, d_{-1}\, d_{-2} \ldots d_{-m}$.

In the above, d_{n-1} to d_0 is integer part, then follows a radix point, and then d_{-1} to d_{-m} is fractional part.

Scanned with CamScanner

d_{n-1} : Most significant bit (MSB)
d_{-m} : Least significant bit (LSB)

* Decimal Numbering System :-

As the decimal number is a weighted number, converting from decimal to binary (base 10 to base 2) will also produce a weighted binary number with the right-hand most bit being the ~~Le~~ Least Significant Bit or LSB, and the left-hand most bit being the Most Significant Bit or MSB.

Example.

$(174)_{10} = (10101110)_2$

2	174	
2	87	0
2	43	1
2	21	1
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

26th June '2020.

* Binary to Decimal :-

1. $(11001010)_2 = (202)_{10}.$

$= [(1 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)]$

$= [(1 \times 128) + (1 \times 64) + (1 \times 8) + (1 \times 2)]$

$= 128 + 64 + 8 + 2$

$= 202.$

2. $(1010.1011)_2 = (10.6875)_{10}.$

$= [(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) . + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4})]$

$= [(1 \times 8) + (1 \times 2) + . (1 \times 0.5) + (1 \times 0.125) + (1 \times 0.0625)]$

$= 8 + 2 + . (0.5 + 0.125 + 0.0625)$

$= 10 + . (0.6875)$

$= 10.6875.$

3. $(100101)_2 = (37)_{10}.$

$= [(1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)]$

Scanned with CamScanner

$= [(1 \times 32) + (1 \times 4) + (1 \times 1)]$

$= 32 + 4 + 1$

$= 37.$

4. $(111101000)_2 = (488)_{10}.$

$= [(1 \times 2^8) + (1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + \cancel{(0+2} (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)]$

$= [(1 \times 256) + (1 \times 128) + (1 \times 64) + (1 \times 32) + (1 \times 8)]$

$= 256 + 128 + 64 + 32 + 8.$

$= 488$

2nd Method,

Exponents :	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Place value :	256	128	64	32	16	8	4	2	1
Bits :	1	1	1	1	0	1	0	0	0
Value :	256 +	128 +	64 +	32		+ 8			= 488

5. $(10001110)_2 = (142)_{10}.$

$= [(1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)]$

$= [(1 \times 128) + (1 \times 8) + (1 \times 4) + (1 \times 2)]$

$= 128 + 8 + 4 + 2$

$= 142.$

6. $(10110101)_2 = (181)_{10}.$

$= [(1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)]$

$= [(1 \times 128) + (1 \times 32) + (1 \times 16) + (1 \times 4) + (1 \times 1)]$

$= 128 + 32 + 16 + 4 + 1$

$= 181.$

* Decimal to Binary :-

1. $(65)_{10} = (1000001)_2.$

2	65	
2	32	1
2	16	0
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

2. $(1792)_{10} = (11100000000)_2$

2	1792	
2	896	0
2	448	0
2	224	0
2	112	0
2	56	0
2	28	0
2	14	0
2	7	0
2	3	1
2	1	1
	0	1

3. $(458)_{10} = (111001010)_2$

2	458	
2	229	0
2	114	1
2	57	0
2	28	1
2	14	0
2	7	0
2	3	1
2	1	1
	0	1

Scanned with CamScanner

29th June '2020.

* Binary to Hexadecimal :-

1. $(10101)_2 = (15)_{16}$.

$\underbrace{0001}_{1}\ \underbrace{0101}_{5}$

Bits 8421 ← 2^4

2. $(11101010)_2 = (EA)_{16}$.

$\underbrace{1110}_{14=E}\ \underbrace{1010}_{10=A}$

3. $(101001.1011)_2 = (29.B)_{16}$.

$\underbrace{0010}_{2}\ \underbrace{1001}_{9}.\ \underbrace{1011}_{11=B}$

* Hexadecimal to Binary :-

1. $(3FA7)_{16} = (11111110100111)_2$.

3	F = 15	A = 10	7
↓	↓	↓	↓
0011	1111	1010	0111

2. $(15)_{16} = (10101)_2$.

1 → 0001   5 → 0101.

* Hexa to Decimal :-

1. $(5A8)_{16} = (1448)_{10}$.

$= [(5 \times 16^2) + (10 \times 16^1) + (8 \times 16^0)]$

$= [(5 \times 256) + (10 \times 16) + (8 \times 1)]$

$= 1280 + 160 + 8$

$= 1448$.

2. $(2C03)_{16} = (11267)_{10}$.

$= [(2 \times 16^3) + (12 \times 16^2) + (0 \times 16^1) + (3 \times 16^0)]$

$= [(2 \times 4096) + (12 \times 256) + (3 \times 1)]$

$= 8192 + 3072 + 3$

$= 11267$ / if e than ans: 11779.

3. $(95)_{16} = (149)_{10}$.

$= [(9 \times 16^1) + (5 \times 16^0)]$

$= [(9 \times 16) + (5 \times 1)]$

= 144 + 5

= 149.

* Binary to Octal :-

1. $(101001.1011)_2 = (51.54)_8$.

$\underbrace{101}_{5}\ \underbrace{001}_{1}\ .\ \underbrace{101}_{5}\ \underbrace{100}_{4}$ Bits : 4 2 1 → 2^3

2. $(1010101)_2 = (125)_8$.

$\underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{101}_{5}$

3. $(10011011)_2 = (233)_8$.

$\underbrace{010}_{2}\ \underbrace{011}_{3}\ \underbrace{011}_{3}$.

* Octal to Binary :-

1. $(51.54)_8 = (101001.1011)_2$.

5 → 101, 1 → 001, . 5 → 101, 4 → $\cancel{0}\cancel{1}$100.

☆ Binary Addition :-

* It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.

Case	A+B	Sum	Carry.
1	0+0	0	0
2	0+1	1	0
3	1+0	1	0
4	1+1	0	1
5	1+1+1	1	1

if 1+0+1
sum = 0
carry = 1.

Example, ~> Addition,

0011010 + 001100 = 00100110.

```
    1  1             ← carry.
  0 0 1 1 0 1 0  =  26₁₀
+ 0 0 0 1 1 0 0  =  12₁₀.
  0 1 0 0 1 1 0  =  38₁₀.
```

if total digits are diff. in expre. add 0 at MSB.

* Sums : Binary Addition :

1. 10001 + 11101 = 101110.

```
  1         1     ← carry.
  0 1 0 0 0 1  =  17₁₀
+ 0 1 1 1 0 1  =  29₁₀
  1 0 1 1 1 0  =  46₁₀.
```

2. 101101 + 11001 = 1000110.

$$
\begin{array}{rcl}
1\,1\,1\quad\;\;1 & \leftarrow & \text{carry} \\
0\,1\,0\,1\,1\,0\,1 & = & 45_{10} \\
+\;0\,0\,1\,1\,0\,0\,1 & = & 25_{10} \\
\hline
1\,0\,0\,0\,1\,1\,0 & = & 70_{10}.
\end{array}
$$

3. 1011001 + 111010 = 10010011.

$$
\begin{array}{rcl}
1\,1\,1\,1\quad\quad\quad & \leftarrow & \text{carry} \\
0\,1\,0\,1\,1\,0\,0\,1 & = & 89_{10} \\
+\;0\,0\,1\,1\,1\,0\,1\,0 & = & 58_{10}. \\
\hline
1\,0\,0\,1\,0\,0\,1\,1 & = & 147_{10}.
\end{array}
$$

30th June, 2020.

4. 1110 + 1111 = 11101.

$$
\begin{array}{rcl}
1\,1\,1\quad\quad & \leftarrow & \text{carry}. \\
1\,1\,1\,0 & = & 14_{10} \\
+\;1\,1\,1\,1 & = & 15_{10} \\
\hline
1\,1\,1\,0\,1 & = & 29_{10}.
\end{array}
$$

5. 10111 + 110101 = 1001100.

$$
\begin{array}{rcl}
1\,1\quad 1\,1\,1\quad & \leftarrow & \text{carry}. \\
0\,1\,0\,1\,1\,1 & = & 23_{10} \\
+\;1\,1\,0\,1\,0\,1 & = & 53_{10} \\
\hline
1\,0\,0\,1\,1\,0\,0 & = & 76_{10}
\end{array}
$$

6. 11011 + 1001010 = 1100101

$$\begin{array}{rccccccccl} & & & 1 & 1 & & 1 & & & \leftarrow \text{carry.} \\ & 0 & 0 & 1 & 1 & 0 & 1 & 1 & = & 27_{10} \\ + & 1 & 0 & 0 & 1 & 0 & 1 & 0 & = & 74_{10} \\ \hline & 1 & 1 & 0 & 0 & 1 & 0 & 1 & = & 101_{10}. \end{array}$$

* Binary Subtraction :-

* Subtraction and Borrow, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

← and borrow 1 from the next more significant bit

Example ~> Subtraction.

(i). 101 from 1001.

$$\begin{array}{rcccc} & 0 & 1 & & \\ & \cancel{1} & {}^{1}0 & 0 & 1 \\ - & & 1 & 0 & 1 \\ \hline & 0 & 1 & 0 & 0 \end{array} \quad \leftarrow \text{Borrow.}$$

∴ Answer: 1001 - 101 = 100.

Scanned with CamScanner

(ii) 1010101.10 from 1111011.11

← Borrow.

$$\begin{array}{r} 1\;1\;1\;\overset{0}{\cancel{1}}\;{}^{1}0\;1\;1\,.\,1\;1 \\ -\;1\;0\;1\;0\;1\;0\;1\,.\,1\;0 \\ \hline 0\;1\;0\;0\;1\;1\;0\,.\,0\;1 \end{array}$$

∴ Answer : 1111011.11 − 1010101.10 = 100110.01.

(iii) 11010.101 from 101100.011.

← Borrow.

$$\begin{array}{rl} \overset{0}{\cancel{1}}\;{}^{1}0\;1\;\overset{0}{\cancel{1}}\;\overset{1}{\cancel{0}}\;\overset{1}{\cancel{0}}\,.\,{}^{1}0\;1\;1 & = 44.375_{10} \\ -\;1\;1\;0\;1\;0\,.\,1\;0\;1 & = 26.625_{10} \\ \hline 0\;1\;0\;0\;0\;1\,.\,1\;1\;0 & = 17.75_{10}. \end{array}$$

∴ Answer: 101100.011 − 11010.101 = 10001.110.

* Sums : Binary Subtraction :

1. 1011011 − 10010 = 1001001

~~1011011~~

~~1001~~

← Borrow.

$$\begin{array}{rl} 1\;0\;1\;1\;0\;1\;1 & = 91_{10} \\ 1\;0\;0\;1\;0. & = 18_{10} \\ \hline 1\;0\;0\;1\;0\;0\;1 & = 73_{10}. \end{array}$$

2. 1010110 - 101010 = 101100.

$$
\begin{array}{rl}
 & \text{0 0} \quad \leftarrow \text{Borrow.} \\
 & \cancel{1}\,{}^{1}0\,\cancel{1}\,{}^{1}0\,1\,1\,0 = 86_{10} \\
- & 1\,0\,1\,0\,1\,0 = 42_{10} \\
\hline
 & 0.1\,0\,1\,1\,0\,0 = 44_{10}.
\end{array}
$$

3. 1000101 - 101100 = 11001

$$
\begin{array}{rl}
 & \text{0 1 1} \quad \leftarrow \text{Borrow.} \\
 & \cancel{1}\,\cancel{0}\,\cancel{0}\,{}^{1}0\,1\,0\,1 = 69_{10} \\
- & 1\,0\,1\,1\,0\,0 = 44_{10}. \\
\hline
 & 0.0\,1\,1\,0\,0\,1 = 25_{10}.
\end{array}
$$

4. 100010110 - 111010 = 100011100.

$$
\begin{array}{rl}
 & \text{0 1 1 1 } {}^{1}0 \quad \leftarrow \text{Borrow} \\
 & \cancel{1}\,\cancel{0}\,\cancel{0}\,\cancel{0}\,\cancel{1}\,{}^{1}0\,1\,1\,0. = 278_{10} \\
- & 1\,1\,1\,1\,0\,1\,0. = 122_{10}. \\
\hline
 & 0.1\,0\,0\,1\,1\,1\,0\,0 = 156_{10}
\end{array}
$$

5. 101101 - 100111 = 110.

$$
\begin{array}{rl}
 & \text{0 } {}^{1}0 \quad \leftarrow \text{Borrow.} \\
 & 1\,0\,\cancel{1}\,\cancel{1}\,{}^{1}0\,1 = 45_{10} \\
- & 1\,0\,0\,1\,1\,1 = 39_{10} \\
\hline
 & 0.0\,0\,1\,1\,0 = 6_{10}.
\end{array}
$$

6. 1110110 - 1010111 = 11111

```
         0  1 0 1 1 0 1 0        ← Borrow.
  1  1   1  0 1 1  0  =  118₁₀
- 1  0   1  0 1 1  1  =   87₁₀
  ---------------------
  0. 0   1  1 1 1  1  =   31₁₀
```

4.

3rd July' 2020.

7. 00100101 - 00010001 = 10100.

```
     0   1                          Borrow.
  0 0 1 0 0 1 0 1   :  37₁₀
- 0 0 0 1 0 0 0 1   :  17₁₀
  ----------------
  0.0 0 1 0 1 0 0   :  20₁₀.
```

8. 00110011 - 00010110 = 11101.

```
      0  1 0  1                    Borrow.
  0 0 1  1 0  0 1 1   :  51₁₀
- 0 0 0  1 0  1 1 0.  :  22₁₀
  -----------------
  0 0 0  1 1  1 0 1   :. 29₁₀.
```

★ Floating Point Numbers :-

31	30 — 23	22 — 0
S	Exponent	Significand
1 bit.	8 bits	23 bits.

Example,

Mantissa — Exponent

$$1.0_{two} \times 2^{-1}$$

"binary point" — radix (base).

* Normalization of floating Point Numbers:

→ Before a floating-point binary number can be stored correctly, its mantissa must be normalized.

→ The process is basically the same as when normalizing a floating-point decimal number. For example, decimal 1234.567 is normalized as 1.234567×10^3 by moving the decimal point so that only one digit appears before the decimal.

→ Rules:

1. Before decimal point only 1 digit should be there.

2. Before decimal point it has to be 1 not 0.

3. The exponent expresses the number of positions the decimal point was moved left (positive exponent) or moved right (negative exponent).

→ Similarly, the floating-point binary value 1101.101 is normalized as 1.101101×2^3 by moving the decimal point 3 positions to the left, and multiplying by 2^3. Here are some examples of normalizations:

Binary Value	Normalized As	Exponent
1101.101	1.101101	3
.00101	1.01	-3
1.0001	1.0001	0
10000011.0	1.00000110.	7.

6th July '2020.

* Eight-Bit Binary Addition Example :

→ Here are some examples of binary addition performed on eight-bit unsigned numbers. Just fill out each number to eight bits, and force the sum to fit as well. If it does not fit, this is considered an "overflow", and will be accompanied by a one bit carried out of the 128's place, a "carryout". With unsigned numbers, overflow and carryout always ~~occ~~ occur together, thought this is not true for two's complement addition.

For example,

1. 1011 + 10010 = 11101

1 ← carry.

0 0 0 0 1 0 1 1
\+ 0 0 0 1 0 0 1 0
0. 0 0 1 1 1 0 1

1. → No ~~over~~ overflow: sum is correct.

2. 1010110 + 110100 = 10001010.

1 1 1 1 ← carry.

0 1 0 1 0 1 1 0
\+ 0 0 1 1 0 1 0 0
1 0 0 0 1 0 1 0

→ No overflow: sum is correct.

3. 11001011 + 1011010 = 100100101

1 1 1 1 ~~0~~ 1 ← carry.

1 1 0 0 1 0 1 1
\+ 0 1 0 1 1 0 1 0
0 0 1 0 0 1 0 1

→ Overflow. The carry-out is discarded, and the sum is not correct.

4. 1010001 + 11001 = 1101010.

1 1 ← carry.

0 1 0 1 0 0 0 1
\+ 0 0 0 1 1 0 0 1
0 1 1 0 1 0 1 0

5. 110001 + 11100100 = 100010101.

	1	1	1						← carry.
	0	0	1	1	0	0	0	1	
+	1	1	1	0	0	1	0	0	
	0	0	0	1	0	1	0	1	

→ Overflow. The carry-out is discarded, and the sum is not correct.

6. 10011011 + 1001010 = 11100101.

			1	1		1			← carry.
	1	0	0	1	1	0	1	1	
+	0	1	0	0	1	0	1	0	
	1	1	1	0	0	1	0	1	

→ No overflow: Sum is ~~overflow~~ correct.

* Complement of Numbers : M Imp

1. (r-1)'s
2. r's

→ If the number is binary, then we use 1's complement and 2's complement. But in case, when the number is a decimal number, we will use the 9's and 10's complement.

→ 1. (r-1)'s Complement :-

I. 1's complement :

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's complement is as follows.

Given number →	1	0	1	0	1
1's complement →	0	1	0	1	0

→ For Example,

Original	1's Complement
10011001	01100110
10000001	01111110
11110000	00001111
11111111	00000000
00000000	11111111

II. 9's Complement :

The 9's complement is used to find the subtraction of the decimal numbers.

→ The 9's complement of a number is calculated by subtracting each digit of the number by 9.

→ For Example, suppose we have a number, 1423, and we want to find the 9's complement of the number. For this, we subtract each digit of the number 1423 by 9. So the 9's complement of the number 1423 is 9999 - 1423 = 8576.

→ There are two possible cases when we subtract the number using 9's complement.

a.) For subtracting the smaller number from the larger number using 9's complement, calculate 9's complement of smaller number or negative number, and by adding complement and first number, the result will come in the formation of carry. At last, we will add this carry to the result obtained previously.

(i) for smaller one find 9's com., (ii) Add com. in the larger no., (iii) if carry is there, add it into the result.

When subtrahend is smaller than the minuend

General Subtraction	Subtraction using 9's Complement.
$\begin{array}{r} 841 \\ -\ 329 \\ \hline 512 \end{array}$ $\quad\begin{array}{r} 999 \\ -329 \\ \hline 670 \end{array}$	$\begin{array}{r} 841 \\ +\ 670 \\ \hline 1\,511 \\ +\ 1 \\ \hline 512 \end{array}$ ← (9's com. of 329)

→ Examples,

1. 718 - 123 = 595

General Subtraction	Subtraction using 9's Complement
$\begin{array}{r} 718 \\ -\ 123 \\ \hline 595 \end{array}$ $\quad \begin{array}{r} 999 \\ -\ 123 \\ \hline 876 \end{array}$	$\begin{array}{r} 718 \\ +\ 876 \\ \hline 1594 \\ +\ \ \ 1 \\ \hline 595. \end{array}$

2. 215 - 155 = 60.

General Subtraction	Subtraction using 9's Complement
$\begin{array}{r} 215 \\ -\ 155 \\ \hline 60 \end{array}$ $\quad \begin{array}{r} 999 \\ -\ 155 \\ \hline 844 \end{array}$	$\begin{array}{r} 215 \\ +\ 844 \\ \hline 1059 \\ +\ \ \ 1 \\ \hline 60. \end{array}$

3. 1234 - 1000 = 234

$\begin{array}{r} 1234 \\ -\ 1000 \\ \hline 0.234 \end{array}$ $\quad \begin{array}{r} 9999 \\ 1000 \\ \hline 8999 \end{array}$	$\begin{array}{r} 1234 \\ +\ 8999 \\ \hline 10233 \\ +\ \ \ \ 1 \\ \hline 0234. \end{array}$

7th July '2020.

b. For subtracting the larger number from the smaller number using 9's complement, calculate 9's complement of negative number, and by adding complement and first number, than no carry is generated then result is negative and make 9's complement of result and put negative sign.

When subtrahend ~~is~~ is greater than minuend.

General Subtraction	Subtraction using 9's Complement
841 - 983 - 142. 999 -983 016	841 + 016 ← (9's com. of 983 857 (NO carry) ↓ - 142 (9's com. of result

→ Examples,

1. 18 - 72 = -54

General Subtraction	Subtraction using 9's Complement
18 - 72 - 54. 99 - 72 27	1 18 + 27 45 ↓ - 54 99 - 45 54

2. $228 - 485 = -257$

General Subtraction	Subtraction using 9's Complement
$\begin{array}{r} 228 \\ -485 \\ \hline -257 \end{array}$	$\begin{array}{r} 228 \\ +514 \\ \hline 742 \\ \downarrow \\ -257 \end{array}$
$\begin{array}{r} 999 \\ -485 \\ \hline 514 \end{array}$	$\begin{array}{r} 999 \\ -742 \\ \hline 257 \end{array}$

3. $571.52 - 837.81 = -266.29$

General Subtraction	Subtraction using 9's complement
$\begin{array}{r} 571.52 \\ -837.81 \\ \hline -266.29 \end{array}$	$\begin{array}{r} 571.52 \\ +162.18 \\ \hline 733.70 \\ \downarrow \\ -266.29 \end{array}$
$\begin{array}{r} 999.99 \\ -837.81 \\ \hline 162.18 \end{array}$	$\begin{array}{r} 999.99 \\ -733.70 \\ \hline 266.29 \end{array}$

→ 2. (r)'s Complement :-

I. 2's complement :

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

Scanned with CamScanner

→ 2's complement = 1's complement + 1

→ <u>Example</u> of 2's complement is as follows

Given Number →	1	0	1	0	1
1's complement →	0	1	0	1	0
Add 1 +					1
2's complement →	0	1	0	1	1

→ <u>Examples</u>.

1. <u>1010101010</u> = 0101010110.

1's complement → 0101010101
Add 1 + 1
2's complement → 0101010110

2. <u>1110011000</u> = <u>0001101000</u>.

1's complement → 0001100111
Add 1 + 1
2's complement → 0001101000

3. <u>11111000</u> = <u>00001000</u>.

1's complement → 00000111
Add 1 + 1
2's complement → 00001000

4. 11100 1101 = 00011001 1

1's complement → 0 0 0 1 1 0 0 1 0
Add 1 + 1
2's complement → 0 0 0 1 1 0 0 1 1

II. 10's Complement:

The 10's complement is used to find the subtraction of the decimal numbers.

→ The 10's complement of a number is calculated by subtracting each digit by 9 and then adding 1 to the result. Simply, by adding 1 to it's 9's complement we can get it's 10's complement value.

→ For example, suppose we have a number 1423, and we want to find the 10's complement of the number. For this, we find the 9's complement of the number 1423 that is 9999 - 1423 = 8576, and now we will add 1 to the result. So the 10's complement of the number 1423 is 8576 +1 = 8577.

→ There are two possible cases when we subtract the numbers using 10's comple-ment.

a.) For subtracting the smaller number from the larger number using 10's complement, calculate 10's complement of smaller number or negative number, and by adding complement and first number, the result will come in the formation of carry, then ignore ~~carry~~ carry.

when subtrahend is smaller than the minuend

General Subtraction	Subtraction using 10's Complement
821 − 413 = 408	821 + 587 (10's com. of 413) = 1408 (ignore the carry) → 408
999 − 413 = 586 + 1 = 587	

→ Examples,

1. 984 - 599 = 385.

General Subtraction	Subtraction using 10's Complement
984 − 599 = 385.	984 + 401 = 1385 → 385
999 − 599 = 400 + 1 = 401	

2. $7542 - 3167 = 4375$

General Subtraction	Subtraction using 10's Complement
$\begin{array}{r} 7542 \\ -\ 3167 \\ \hline 4375 \end{array}$ $\begin{array}{r} 9999 \\ -3167 \\ \hline 6832+1 \\ =6833 \end{array}$	$\begin{array}{r} 7542 \\ +\ 6833 \\ \hline 14375 \end{array}$ ↓ 4375.

3. $7188 - 3049 = 4139.$

General Subtraction	Subtraction using 10's Complement
$\begin{array}{r} 7188 \\ -\ 3049 \\ \hline 4139 \end{array}$ $\begin{array}{r} 9999 \\ -3049 \\ \hline 6950+1 \\ =6951 \end{array}$	$\begin{array}{r} 7188 \\ +\ 6951 \\ \hline 14139 \end{array}$ ↓ 4139.

b.) For subtracting the larger number from the smaller number using 10's complement, first calculate 10's complement of negative number, and by adding complement and first number, than no carry is generated than result is negative and make 10's complement of result and put negative sign.

When subtrahend is greater than the minuend.

General Subtraction	Subtraction using 10's Complement
325 - 641 - 316. 999 - 641 358 + 1 = 359	325 + 359 (10's com. 641) 684 (No carry) ↓ - 316. (10's com result)

→ Examples,

1. 670 - 831 = -161

General Subtraction	Subtraction using 10's Complement
670 - 831 - 161 999 - 831 168 + 1 = 169	670 + 169 839 ↓ - 161. 999 - 839 160 + 1 = 161

2. 571.52 - 837.81 = -266.29

571.52 - 837.81 - 266.29 999.99 - 837.81 162.18 + 1 = 163.18	571.52 + 163.18 734.70 ↓ - 266.29 999.99 - 734.70 265.29 + 1 = 266.29

3. 155 - 215 = -60

General Subtraction		Subtraction using 10's complement	
155 - 215 - 60.	999 -215 784+1 = 785	11 155 +785 940 ↓ -60	999 -940 059+1 = 60

4. 150 - 2100 = -1950.

General Subtraction		Subtraction using 10's Complement	
150 - 2100 -1950	9999 -2100 7899+1 = 7900	1 150 +7900 8050 ↓ -1950.	9999 -8050 1949+1 = 1950

10th July' 2020.

* Subtraction using 1's Complement:

These are the following steps to subtract two binary numbers using 1's complement.

- In the first step, find the 1's complement of the subtrahend. (negative number).

- Next. Add the complement number with minuedend. (first number).
- If got a carry, add the carry to its LSB.
- If carry is not generated than take 1's complement of the result which will be negative.

Example.

(i) 10101 - 00111.

→ We take 1's complement of subtrahend 00111. which comes out 11000 Now sum them. So,

→ 10101 + 11000 = 1 01101

→ In the above result, we get the carry bit 1, so add this to the LSB of a given result. i.e., 01101 + 1 = 01110, which is the answer. (Normal Subtraction ans. will be same

(ii) 10101 - 10111

→ We take 1's complement of subtrahend 10111, which comes out 01000. Now add both of the numbers. So,

→ 10101 + 01000 = 11101.

→ In the above result, we didn't get the carry bit. So calculate the 1's complement of the result, i.e., 00010, which is the negative number and the final answer.

→ Examples,

1. 110101 - 100101 = 10000

→ 1's complement of 100101 is 011010.

→ Adding both,

```
   1 1
   1 1 0 1 0 1
 + 0 1 1 0 1 0
 1 0 0 1 1 1 1
```

→ Here carry is generated, adding carry.

```
     1 1 1 1
   0 0 1 1 1 1
 +           1
   0 1 0 0 0 0
```

→ Answer is: 110101 - 100101 = 010000.

2. 101011 - 111001 = -1110.

→ 1's complement of 111001 is 000110.

→ Adding both,

$$
\begin{array}{r}
 1\ 1\ 1 \\
1\ 0\ 1\ 0\ 1\ 1 \\
+\ \underline{0\ 0\ 0\ 1\ 1\ 0} \\
1.\ 1\ 0\ 0\ 0\ 1
\end{array}
$$

→ Here carry is not generated, so 1's complement of result is 001110. which is negative.

→ Answer: 101011 - 111001 = -1110.

3. 1011.001 - 110.10 = 100.101

→ 1's complement of 0110.100 is 1001.011

→ Adding both,

$$
\begin{array}{r}
1\ \ 1\ 1\ \ 1\ 1 \\
1\ 0\ 1\ 1.\ 0\ 0\ 1 \\
+\ \underline{1\ 0\ 0\ 1.\ 0\ 1\ 1} \\
1\ 0\ 1\ 0\ 0.\ 1\ 0\ 0
\end{array}
$$

→ Here carry is generated, adding carry.

$$
\begin{array}{r}
0\ 1\ 0\ 0.\ 1\ 0\ 0 \\
+\ \underline{1} \\
1\ 0\ 0.\ 1\ 0\ 1
\end{array}
$$

→ Answer: 1011.001 - 110.10 = 100.101.

4. 10110.01 - 11010.10 = -100.01

→ 1's complement of ~~xxxxx.xx~~ 11010.10 is 00101.01

→ Adding both,

$$\begin{array}{r} 11 \\ 10110.01 \\ +\ 00101.01 \\ \hline 11011.10 \end{array}$$

→ Here carry is not generated, so 1's complement of result is 00100.01 which is negative.

→ Answer: 10110.01 - 11010.10 = -100.01.

* Subtraction using 2's complement:

These are the following steps to subtract two binary numbers using 2's complement.

- In the first step, find the 2's complement of the subtrahend.
- Add the complement number with the minuend.
- If we get the carry by adding both the numbers, then we discard this carry and the result is positive.
- If carry is not generated than take 2's complement of the result which will be negative.

Example,

(i) 10101 - 00111.

→ We take 2's complement of subtrahend 00111, which is 11001. Now, sum them. So,

→ 10101 + 11001 = 1 01110.

→ In the above result, we get the carry bit 1. So we discard this carry bit and remaining is the final result and a positive number.

(ii) 10101 - 10111

→ We take 2's complement of subtrahend 10111, which comes out 01001. Now, we add both of the numbers. So,

→ 10101 + 01001 = 11110.

→ In the above result, we didn't get the carry bit. So calculate the 2's complement of the result. i.e., 00010. It is the negative number and the final answer.

→ Examples,

1. 110110 - 10110. = ______

→ 2's complement of 10110 is 01010.

→ Adding both,

$$
\begin{array}{r}
1\ 1\ 1\ 1\ 1\quad \\
1\ 1\ 0\ 1\ 1\ 0 \\
+\quad 1\ 0\ 1\ 0\ 1\ 0 \\
\hline
\underline{1}\ 1\ 0\ 0\ 0\ 0\ 0
\end{array}
$$

→ Here carry is generated. So we will discard it and remaining part will be our answer.

→ Answer: 110110 - 10110 = 100000.

2. 10110 - 11010 = -100.

→ 2's complement of 11010 is 00110.

→ Adding both,

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

→ Here carry is not generated, so 2's complement of result is 100 which is negative.

→ Answer: 10110 - 11010 = -100.

3. 1010.11 - 1001.01 = 1.10.

→ 2's complement of 1001.01 is 0110.11.

→ Adding both,

$$\begin{array}{r} {\scriptstyle 1\,1\,1\quad 1\,1} \\ 1010.11 \\ +\ \underline{0110.11} \\ \underline{1}0001.10 \end{array}$$

→ Here carry is generated. So we will discard it and remaining part will be our answer.

→ Answer: 1010.11 - 1001.01 = 1.10.

4. 10100.01 - 11011.10 = -00111.01.

→ 2's complement of 11011.10 is 00100.10.

→ Adding both,

$$\begin{array}{r} {\scriptstyle 1} \\ 10100.01 \\ +\ \underline{00100.10} \\ 11000.11 \end{array}$$

→ Here carry is not generated, so 2's complement of result is 00111.01. which is negative.

→ Answer: 10100.01 - 11011.10 = -111.01.

* Positive Signed Binary Numbers -

$$\overbrace{0\,|\,0\,1\,1\,0\,1\,0\,1}^{\text{8-bit word}}$$

Positive sign bit

Magnitude bits.

* Negative Signed Binary Numbers -

$$\overbrace{1\,|\,0\,1\,1\,0\,1\,0\,1}^{\text{8-bit word}}$$

Negative sign bit

Magnitude bits.

A. Addition of a positive number and a negative number

We consider the following cases.

* **Case I** : When the positive number has a greater magnitude.

→ In this case the carry which will be generated is discarded and the final result is the result of addition.

→ The following examples will illustrate this method in binary addition using 2's Complement :

→ In a 5-bit register find the sum of the following by using 2's complement :

Example,

(i) +1011 and -0101.

$\boxed{1}$	1			1	1
	0	1	0	1	1
+	1	1	0	1	1
$\boxed{1}$	0	0	1	1	0

carry sign ans.

→ Solution :

+1011 ⇒ 0 1011

-0101 ⇒ 1 1011 (2's complement).

(carry 1 discarded). 0 0110.

→ Hence the sum is +0110.

(ii) +0111 and -0011.

	1	1	1	1	
	0	0	1	1	1
+	1	1	1	0	1
$\boxed{1}$	$\boxed{0}$	0	1	0	0

carry sign ans.

→ Solution :

+0111 ⇒ 0 0111

-0011 ⇒ 1 1101 (2's complement).

(carry 1 discarded) 0 0100.

→ Hence the sum is +0100.

* → Case II : When the negative number is greater.

→ The following examples will illustrate this method in binary addition using 2's comple.

→ In a 5-bit register find the sum of the following by using 2's complement.

(i). +0011 and -0101.

	0	0	0	1	1
+	1	1	0	1	1
	1	1	1	1	0
	sign		ans.		

→ Solution:

+0011 ⇒ 0 0011
-0101 ⇒ 1 1011 (2's com.)
(No carry gener.) 1 1110

→ 2's complement of 1110 is 0010.

→ Hence the required sum is -0010.

(ii) +0100 and -0111.

	0	0	1	0	0
+	1	1	0	0	1
	1	1	1	0	1
	sign		ans.		

→ Solution:

+0100 ⇒ 0 0100
-0111 ⇒ 1 1001
(No carry gener.) 1 1101

→ 2's complement of 1101 is 0011.

→ Hence the required sum is -0011.

→ When the negative number is greater no carry will be generated in the sign bit. The result of addition will be negative and the final result is obtained by taking 2's complement of the magnitude bits of the result. (This happens in case II).

13th July '2020.

B. When the numbers are negative.

~~when two negative numbers~~

→ When two negative numbers are added a carry will be generated from the sign bit which will be discarded. 2's complement of the magnitude bits of the operation will be the final sum.

→ The following examples will illustrate this method in binary addition using 2's complement:

→ In a 5-bit register find the sum of the following by using 2's complement:

Examples,

(i) -0011 and -0101.

→ Solution:

-0011	⇒	1 1101	(2's complement)
-0101	⇒	1 1011.	(2's complement).
(carry 1 discarded).		1. 1000	

→ 2's complement of 1000 is 1000.

→ Hence the required sum is -1000.

(iii) -0111 and -0010.

→ Solution:

- 0111 ⇒ 1 1001 (2's complement)
- 0010 ⇒ 1 1110 (2's complement)
(carry 1 discarded). 1 0111

→ 2's complement of 0111 is 1001.

→ Hence the require sum is -1001.

Examples, (For both A & B)

1. -39 + 92 = 53.

→ Binary of -39 = 0100111.
→ Binary of 92 = 1011100.
→ 2's complement of 0100111 (-39) is 1011001.

- 0100111 ⇒ 1 1011001
+ 1011100 ⇒ 0 1011100.
(carry 1 discarded.) 0 0110101

→ Hence the sum is +0110101 = +53.

→ Carryout without overflow. Sum is correct.

2. -19 + -7 = -26.

→ Binary of -19 = 0010011.
Binary of -7 = 0000111.

→ 2's complement of 0010011(19) is 1101101.
2's complement of 0000111(7) is 1111001.

- 0010011	⇒	1 1101101
- 0000111	⇒	1 1111001
(carry 1 discarded).		1 1100110

→ 2's complement of 1100110 is 0011010.

→ Hence the sum is -0011010 = -26.

→ Carry out without overflow. Sum is correct.

3. 44 + 45 = 89.

→ Binary of 44 = 0101100
Binary of 45 = 0101101.

+ 0101100	⇒	0 0101100
+ 0101101	⇒	0 0101101
(No carry generated).		0 1011001

→ Hence the sum is +1011001 = +89.

→ No overflow nor carry out.

4. 104 + 45 = 149.

→ Binary of 104 = 1101000.
Binary of 45 = 0101101

+ 1101000 ⇒ 0 1 1 0 1 0 0 0
+ 0101101 ⇒ 0 0 1 0 1 1 0 1.
(carry is generataed). 1 0 0 1 0 1 0 1

→ Overflow, no carry out. Sum is not correct.

5. -75 + 59 = -16.

→ Binary of -75 = 100 1011
Binary of 59 = 011 1011.

→ 2's complement of 1001011 (75) is 0110101

- 1001011 ⇒ 1 0 1 1 0 1 0 1
+ 0111011 ⇒ 0 0 1 1 1 0 1 1.
1 1 1 1 0 0 0 0

→ 2's complement of 1110000 is 0010000.

→ Hence the sum is -10000 = -16.

→ No overflow nor carryout.

6. -103 + -69 = -172.

→ Binary of -103 = 1100111
Binary of -69 = 100 0101.

→ 2's complement of 1100111 (103) is 0011001
2's complement of 1000101 (69) is 0111011.

- 1100111 ⇒ 1 0011001
- 1000101 ⇒ 1 0111011.
(carry is generated). 0 1010100

→ Overflow with incidental carryout. Sum is not correct.

107 10 + (-3) = 7.

→ Binary of 10 = 0001010.
Binary of -3 = 0000011.

→ 2's complement of 0000011 (3) is 1111101.

+ 0001010 ⇒ 0 0001010.
- 0000011 ⇒ 1 1111101.
(carry 1 discarded). 0. 0000111

→ Hence the sum is + 0000111 = +7.

→ Carryout without overflow, Sum is correct.

8. 127 + 1 = 128.

→ Binary of 127 = 1111111
Binary of 1 = 0000001.

+ 1111111 ⇒ 0 1111111
+ 0000001 ⇒ 0 0000001
1 0000000

→ Overflow, no carryout. Sum is not correct.

9. -1 + 1 = 0.

→ Binary of -1 = 0000001
Binary of 1 = 0000001.

→ 2's complement of 0000001 (-1) is 1111111.

- 00000001 ⇒ 1 1111111
+ 0000001 ⇒ 0 0000001
(carry 1 discarded). 0. 0000000

→ Hence the sum is +0000000 = 0.

→ Carryout without overflow. Sum is correct.