SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECB212 – DIGITAL SIGNAL PROCESSING

II YEAR/ IV SEMESTER

UNIT 4 – FINITE WORD LENGTH EFFECTS

TOPIC – **REPRESENTATION OF NUMBERS – FIXED POINT AND FLOATING POINT REPRESNETATION**

FINITE WORD LENGTH EFFECTS IN DIGITAL FILTERS

- The fundamental operations in the various computational procedure like convolution, Spectral estimation etc.. in DSP are multiplication and addition
- These operations are performed using the samples of input sequence, impulse response and the coefficients of the difference equation governing the system
- The informations (or numbers) used for computation are called input data and the results of computation are called output data. The input and output data are stored in registers in a digital system

FINITE WORD LENGTH EFFECTS IN DIGITAL FILTERS

- The registers are the basic storage device in digital system. The maximum size of the binary information (or data) that can be stored in a register is called register word length
- For Example: When a register stores an 8-bit data then its word length is 8-bit . For storing the input data in registers they have to be quantized and coded in binary
- The quantization and coding depends on their register word length Eg: When the register word length is 8-bit, we can generate 2^8 = 256 binary codes and so we have 256 quantized levels

FINITE WORD LENGTH EFFECTS IN DIGITAL FILTERS

- The register is used to store the result is 8-bit, then the result has to be truncated or rounded to accommodate in the register
- This makes the system nonlinear, leads to limit cycle behaviour. The effect of truncation or rounding can be represented interms of an additive error signal, which is called roundoff noise
- The effects due to finite precision representation of numbers in a digital system are commonly referred to as Finite Word Length Effects

FINITE WORD LENGTH EFFECTS IN DIGITAL FILTERS

- The following are some of the finite word length effects in digital filters
- Errors due to quantization of input data by A/D (Analog to Digital) Converter
- Errors due to quantization of filter coefficients
- Errors due to rounding the product in multiplication
- Errors due to overflow in addition
- Limit cycles

REPRESENTATION OF NUMBERS IN DIGITAL SYSTEMS

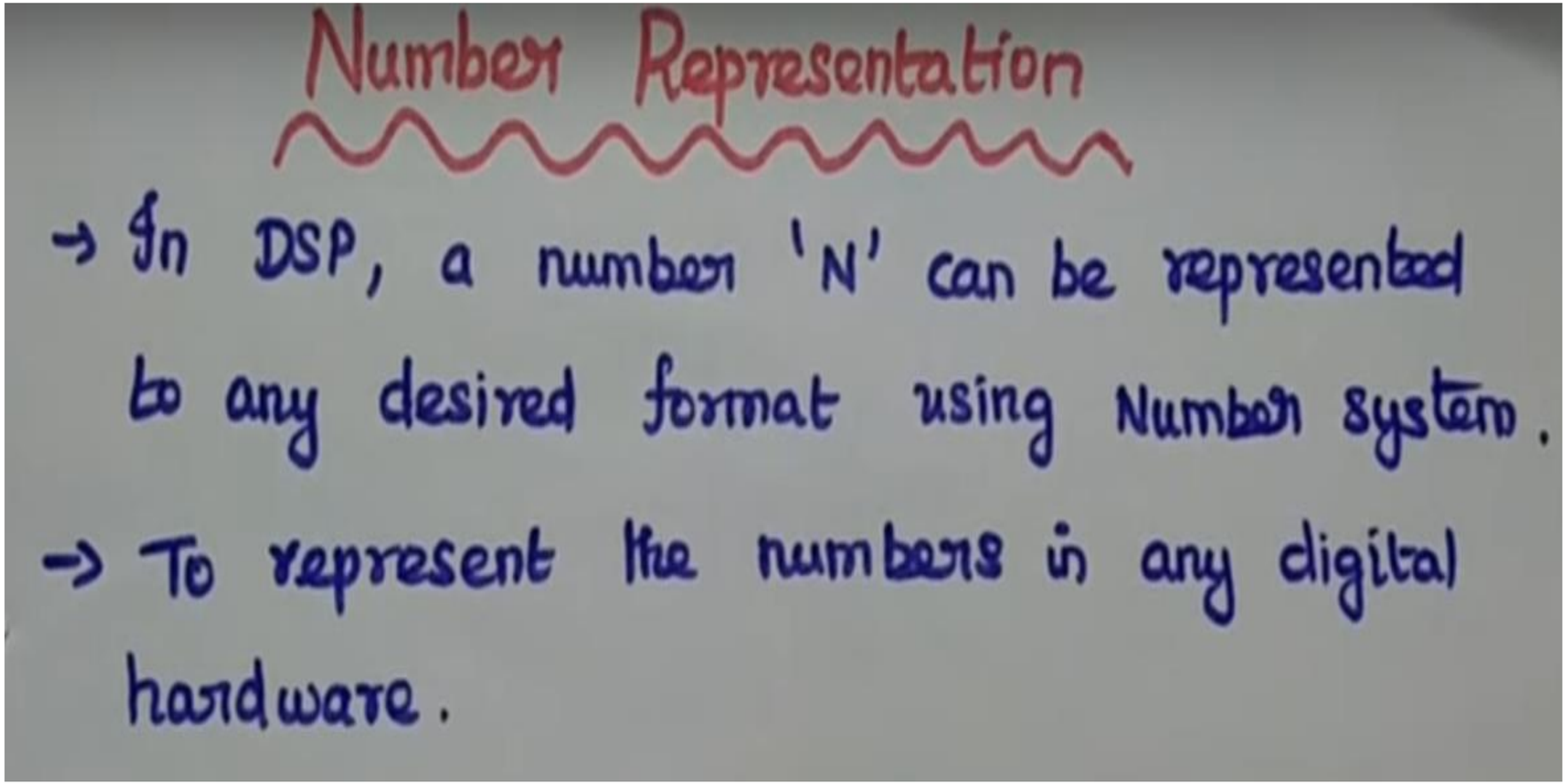
- **Binary Codes:** The binary codes are framed using the numeric symbols ‘0’ and ‘1’
- Each digit of the binary code is called bit. The size of the binary code is specified in terms of number of bits
- In digital system the binary codes are used to represent any information like text, Images, Numbers, etc...
- When decimal numbers are represented in binary codes, the size of the code will decide the range of numbers that can be represented in binary

Number Representation in DSP

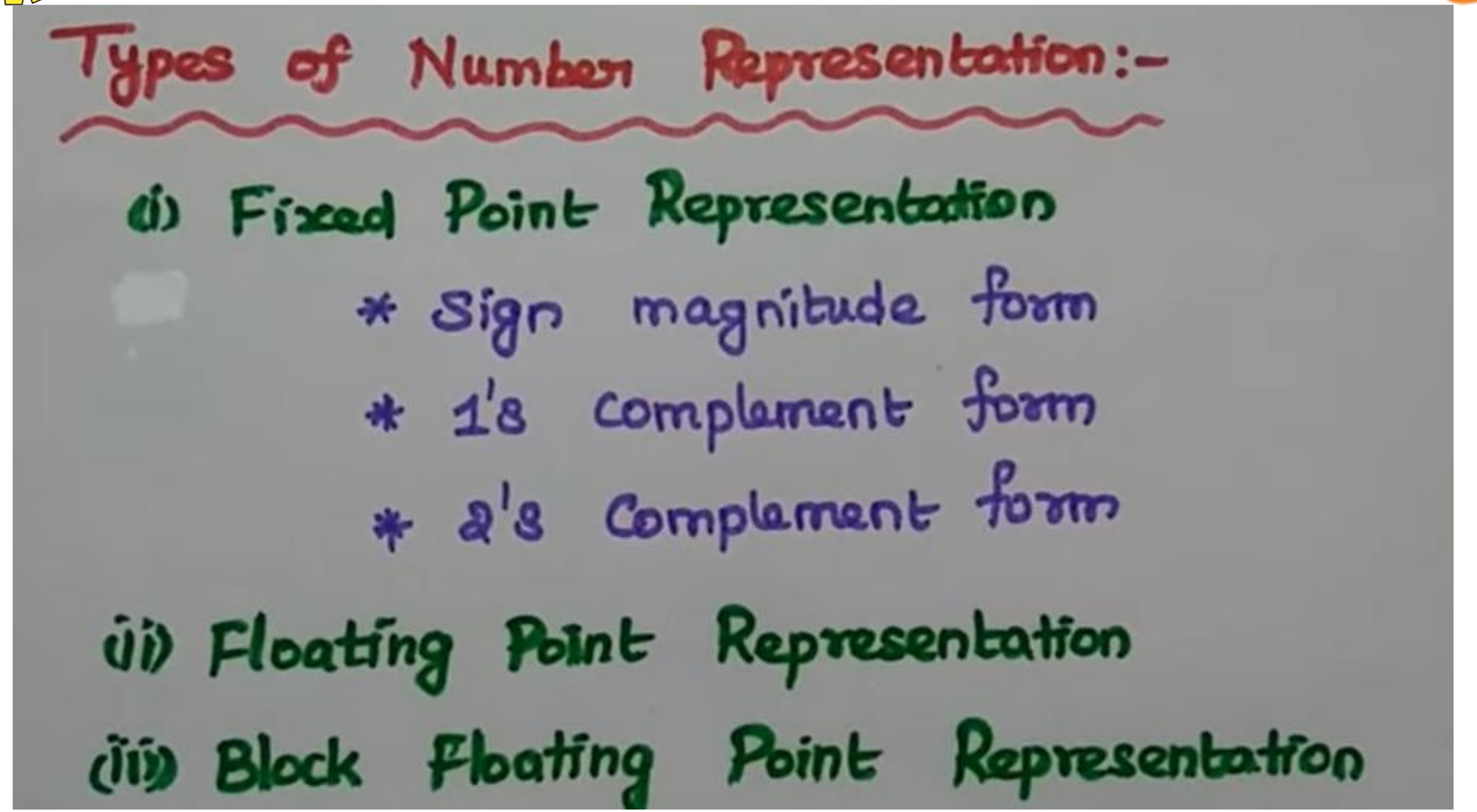
Fixed Point Representation	Floating Point Representation
* Fast & Inexpensive Implemantation	Slow & Expensive Implementation.
* Limited Dynamic range	Large Dynamic range
* Round off errors occur only for addition	Round off errors can occur with both addition & multiplication
* Overflow occurs in addition process	Overflow does not occur.
* Low power consumption	High power consumption
* Less flexible	More flexible

* **Fixed Point**
* **Floating Point**
* **Block Floating Point**

REPRESENTATION OF NUMBERS IN DIGITAL SYSTEMS



REPRESENTATION OF NUMBERS IN DIGITAL SYSTEMS



REPRESENTATION OF NUMBERS IN DIGITAL SYSTEMS

When $n = 1$; $2^n = 2^1 = 2$

Binary codes :

0

1

When $n = 2$; $2^n = 2^2 = 4$

Binary codes :

0 0

0 1

1 0

1 1

When $n = 3$; $2^n = 2^3 = 8$

Binary codes :

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

When $n = 4$; $2^n = 2^4 = 16$

Binary codes :

0 0 0 0

0 0 0 1

0 0 1 0

0 0 1 1

0 1 0 0

0 1 0 1

0 1 1 0

0 1 1 1

1 0 0 0

1 0 0 1

1 0 1 0

1 0 1 1

1 1 0 0

1 1 0 1

1 1 1 0

1 1 1 1

REPRESENTATION OF NUMBERS IN DIGITAL SYSTEMS

- When 4 – bit binary is used to represent unsigned decimal integers then the range is 0 to 2^4 -1 = 0 to 15_{10}
- When 4 – bit binary is used to represent signed decimal integers in sign – magnitude format then the range is -(2^3 -1) to +(2^3 -1) = -7_{10} to $+7_{10}$
- When 4 – bit binary is used to represent unsigned decimal fraction in fixed point representation then the range is 0 to 1- 2^{-4} =0 to 15/16= 0 to 0.9375_{10}
- When 4 – bit binary is used to represent signed decimal fraction in fixed point sign – magnitude format, then the range is -(1-2^{-3}) to +(1-2^{-3}) = -7/8 to +7/8 = -0.875_{10} to 0.875_{10}

BINARY REPRESENTATION OF DECIMAL NUMBERS

Binary Code	Unsigned decimal integer	Signed decimal integer	Unsigned decimal fraction	Signed decimal fraction
0 0 0 0	0	0	0/16 = 0	0/8 = 0
0 0 0 1	1	1	1/16 = 0.0625	1/8 = 0.125
0 0 1 0	2	2	2/16 = 0.1250	2/8 = 0.250
0 0 1 1	3	3	3/16 = 0.1875	3/8 = 0.375
0 1 0 0	4	4	4/16 = 0.2500	4/8 = 0.500
0 1 0 1	5	5	5/16 = 0.3125	5/8 = 0.625
0 1 1 0	6	6	6/16 = 0.3750	6/8 = 0.750
0 1 1 1	7	7	7/16 = 0.4375	7/8 = 0.875
1 0 0 0	8	–0	8/16 = 0.5000	–0/8 = –0
1 0 0 1	9	–1	9/16 = 0.5625	–1/8 = –0.125
1 0 1 0	10	–2	10/16 = 0.6250	–2/8 = –0.250
1 0 1 1	11	–3	11/16 = 0.6875	–3/8 = –0.375
1 1 0 0	12	–4	12/16 = 0.7500	–4/8 = –0.500
1 1 0 1	13	–5	13/16 = 0.8125	–5/8 = –0.625
1 1 1 0	14	–6	14/16 = 0.8750	–6/8 = –0.750
1 1 1 1	15	–7	15/16 = 0.9375	–7/8 = –0.875

RADIX NUMBER SYSTEM

$$178.25_{10} = (1 \times 10^2) + (7 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (5 \times 10^{-2})$$

$$= (d_{-2} \times 10^2) + (d_{-1} \times 10^1) + (d_0 \times 10^0) + (d_1 \times 10^{-1}) + (d_2 \times 10^{-2})$$

$$= \sum_{i=-2}^{2} d_i \, r^{-i} \quad ; \text{where, } r = 10$$

$$111.11_2 = (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2})$$

$$= (d_{-2} \times 2^2) + (d_{-1} \times 2^1) + (d_0 \times 2^0) + (d_1 \times 2^{-1}) + (d_2 \times 2^{-2})$$

$$= \sum_{i=-2}^{2} d_i \, r^{-i} \quad ; \text{where, } r = 2$$

RADIX NUMBER SYSTEM

- Any number can be represented as

$$\text{Number, } N = \sum_{i=-A}^{B} d_i \, r^{-i}$$

- A – Number of Integer digits
- B – Number of Fraction digits
- r – Radix or Base
- d_i - $_i^{th}$ digit of the Number
- In digital systems the numbers are represented in binary, in which the radix r=2

$$\text{Binary number, } N = \sum_{i=-A}^{B} d_i \, 2^{-i}$$

RADIX NUMBER SYSTEM

- The binary digit d_{-A} is called the Most Significant Digit (MSD) and the binary digit d_B is called the Least Significant Digit (LSD) of the binary Number N
- The binary point between the digits d_0 and d_1 doesnot exist physically in the digital system. The binary digit is also known as bit.
- For the fraction format of binary numbers the equation can be modified as

$$\text{Binary fraction number, } N = \pm \sum_{i=1}^{B} d_i \, 2^{-i}$$

$$\text{or Binary fraction number, } N = \sum_{i=0}^{B} d_i \, 2^{-i}$$

RADIX NUMBER SYSTEM

- The two major methods of representing binary numbers are Fixed Point Representation and Floating Point Representation.
- In Fixed Point Representation the digits allotted for integer part and fraction part are fixed and so the position of binary point is fixed. Since the number of digits is fixed it is impossible to represent too large and too small numbers by fixed point representation.
- Therefore the range of numbers that can be represented in fixed point representation for a given binary word size is less when compared to floating point representation

RADIX NUMBER SYSTEM

- In Floating Point Representation the binary point can be shifted to desired position so that number of digits in the integer part and fraction part of the number can be varied. This leads to larger range of number that can be represented in floating point representation
- In Fixed Point Representation there are three different formats for representing negative binary fraction numbers. They are

1. Sign-Magnitude Format
2. One's Complement Format
3. Two's Complement Format

RADIX NUMBER SYSTEM

- In Fixed Point Representation there is only one unique way of representing positive binary fraction number

$$N_p = 0.d_1d_2.....d_B$$

$$= (0 \times 2^0) + (d_1 \times 2^{-1}) + (d_2 \times 2^{-2}) + ... + (d_B \times 2^{-B})$$

$$= \sum_{i=0}^{B} d_i\, 2^{-i} \quad ;\text{where } d_0 = 0$$

$$= (0 \times 2^0) + \sum_{i=1}^{B} d_i\, 2^{-i}$$

- The most significant digit d_0 is set to zero to represent the positive sign. For negative numbers the most significant digit d_0 is one to represent the negative sign

SIGN MAGNITUDE FORMAT

- In Sign magnitude format the negative value of a given number differ only in sign bit (i.e., digit d_0) The sign digit d_0 is zero for positive number and one for negative number

$$\therefore \text{Negative binary fraction number, } N_N = (1 \times 2^0) + \sum_{i=1}^{B} d_i 2^{-i}$$

- The range of decimal fraction numbers that can be represented in B –bit fixed point Sign – magnitude format is

$$-\left[1-2^{-(B-1)}\right] \text{ to } +\left[1-2^{-(B-1)}\right] \quad ; \text{with step size } = \frac{1}{2^{B-1}}$$

SIGN MAGNITUDE FORMAT

- The range of decimal fraction numbers that can be represented in B –bit fixed point Sign – magnitude format is

$$-\left[1-2^{-(B-1)}\right] \text{ to } +\left[1-2^{-(B-1)}\right] \quad ; \text{with step size } = \frac{1}{2^{B-1}}$$

When B = 4,

$$\text{Range} = -\left[1-2^{-(4-1)}\right] \text{ to } +\left[1-2^{-(4-1)}\right] = -\left[1-\tfrac{1}{8}\right] \text{ to } +\left[1-\tfrac{1}{8}\right] = -\tfrac{7}{8} \text{ to } +\tfrac{7}{8}$$

$$= -0.875_{10} \text{ to } +0.875_{10}$$

$$\text{Step size } = \frac{1}{2^{4-1}} = \frac{1}{2^3} = \frac{1}{8} = 0.125_{10}$$

DECIMAL EQUIVALENT OF 4-BIT BINARY NUMBERS IN FIXED POINT REPRESENTATION

Binary number in fixed point representation			Decimal
Sign-magnitude	One's complement	Two's complement	Equivalent
0 0 0 0	0 0 0 0	0 0 0 0	0
0 0 0 1	0 0 0 1	0 0 0 1	1/8 = 0.125
0 0 1 0	0 0 1 0	0 0 1 0	2/8 = 0.250
0 0 1 1	0 0 1 1	0 0 1 1	3/8 = 0.375
0 1 0 0	0 1 0 0	0 1 0 0	4/8 = 0.500
0 1 0 1	0 1 0 1	0 1 0 1	5/8 = 0.625
0 1 1 0	0 1 1 0	0 1 1 0	6/8 = 0.750
0 1 1 1	0 1 1 1	0 1 1 1	7/8 = 0.875
1 0 0 0	1 1 1 1	———	–0
1 0 0 1	1 1 1 0	1 1 1 1	–1/8 = –0.125
1 0 1 0	1 1 0 1	1 1 1 0	–2/8 = –0.250
1 0 1 1	1 1 0 0	1 1 0 1	–3/8 = –0.375
1 1 0 0	1 0 1 1	1 1 0 0	–4/8 = –0.500
1 1 0 1	1 0 1 0	1 0 1 1	–5/8 = –0.625
1 1 1 0	1 0 0 1	1 0 1 0	–6/8 = –0.750
1 1 1 1	1 0 0 0	1 0 0 1	–7/8 = –0.875
		1 0 0 0	–8/8 = –1.000

ONE'S COMPLEMENT FORMAT

- The positive number is same in all the formats of fixed point representation and it is given by $(0 \times 2^0) + \sum_{i=1}^{B} d_i 2^{-i}$
- In one's complement format the negative of the given number is obtained by bit by bit complement of its positive representation. The complement of a digit d_i can be obtained by subtracting the digit from one

$$\text{Complement of } d_i = \bar{d}_i = (1 - d_i)$$

- From eqn $(0 \times 2^0) + \sum_{i=1}^{B} d_i 2^{-i}$ if we set the sign bit to one and replace d_i by $(1- d_i)$

$$\therefore \text{ Negative binary fraction number in one's complement } N_{1c} = (1 \times 2^0) + \sum_{i=1}^{B} (1 - d_i)2^{-i}$$

TWO'S COMPLEMENT FORMAT

- The positive number is same in all the formats of fixed point representation and it is given by $(0 \times 2^0) + \sum_{i=1}^{B} d_i 2^{-i}$
- In two's complement format the negative of the given number is obtained by taking one's complement of its positive representation and then adding one to the least significant bit. If we add $_{1x2}{}^{-B}$ the we get two's complement format for negative numbers

$$\therefore \text{Negative binary fraction number in two's complement} \Big\} N_{2c} = (1 \times 2^0) + \sum_{i=1}^{B} (1 - d_i)2^{-i} + (1 \times 2^{-B})$$

TWO'S COMPLEMENT FORMAT

- The two's complement format provides single representation for zero, whereas the sign-magnitude and one's complement format has two representation for zero. Hence, the two's complement format for binary number system is practically used in all digital systems
- The range of decimal fraction numbers that can be represented in B-bit fixed point two's complement format is

$$-1 \text{ to } +\left[1-2^{-(B-1)}\right] \quad ;\text{with step size } = \frac{1}{2^{B-1}}$$

When B = 4,

$$\text{Range} = -1 \text{ to } +\left[1-2^{-(4-1)}\right] = -1 \text{ to } +\left[1-\frac{1}{8}\right] = -1 \text{ to } +\frac{7}{8} = -1 \text{ to } +0.875_{10}$$

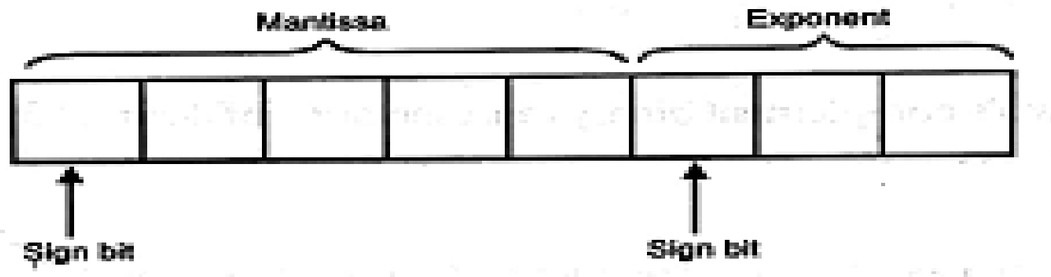
$$\text{Step size } = \frac{1}{2^{4-1}} = \frac{1}{2^3} = \frac{1}{8} = 0.125_{10}$$

FLOATING POINT REPRESENTATION

- The floating point representation is employed to represent larger range of numbers in a given binary word size. The floating point number is represented as
- Floating Point Number, $\mathbf{N_r = M \times 2^E}$
- M is called Mantissa and it will be in binary fraction format. The value of M will be in the range $0 \leq M \leq 1$
- E is called Exponent and it is either a positive or negative integer
- In floating point representation both mantissa and exponent uses one bit for representing sign. Usually the leftmost bit in mantissa and exponent is used to represent the sign

FLOATING POINT REPRESENTATION

- “1” is the leftmost bit position represents negative sign and “0” in the leftmost bit position represents positive sign
- The floating point representation is explained by considering a five bit mantissa and three bit exponent with a total size of eight bits
- In mantissa the leftmost bit is used to represent the sign and other four bits are used to represent a binary fraction number
- In exponent the leftmost bit is used to represent the sign and other two bits are used to represent a binary integer number



ASSESSMENT

1. Define register word length.
2. What is meant by round off noise.
3. Summarize the finite word length effects in digital filters.
4. How to represent any number in radix number system?
5. List the two methods of representing binary numbers.
6. Mention the three different formats for representing negative binary fraction numbers.
7. Define Floating Point representation.

THANK YOU