Data Representation

सीबीएसई पाठ्यक्रम पर आधारित

कक्षा -11

अध्याय - 13

द्वारा:

संजीव भदौरिया

स्नातकोत्तर शिक्षक (संगणक विज्ञान)

के० वि० बाराबंकी (लखनऊ संभाग)

परिचय

- जैसा कि हम जानते हैं की कंप्यूटर में कोई भी डाटा binary के रूप में store होता है | इसीलिए कंप्यूटर से सम्बंधित कार्यों के साथ digital शब्द जोड़ दिया जाता है | और कंप्यूटर में संग्रहीत डाटा को हम digital data कहते हैं |
- इस अध्याय में हम दितल तकनीकों के बारे में पढेंगे की कैसे कंप्यूटर में डाटा को प्रदर्शित किया जाता है |
- मानव अपने दैनिक जीवन में एक संख्या पद्यति (number system) को अपनाता है उसका नाम है दशमलव संख्या प्रणाली , इसी प्रकार से कंप्यूटर जिन संख्या पद्यतियों को अपनाता है उनमे binary number system, Octal number system और hexadecimal number system हैं | जिन्हें हम digital number system कहते हैं |
- तो आइये in पद्यतियों के बारे में जानते हैं -

Number Systems		
System	Base	Digits
Binary	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

Decimal number system

- Decimal system 10 अंकों या संकेतों से मिलकर बना है जो की निम्न हैं | इन्हें Digit कहा जाता है |
  - 0,1,2,3,4,5,6,7,8,9
- इस system का आधार 10 होता है तथा इसे निम्न प्रकार दर्शाते हैं -
  - $(1249)_{10}$
- यह एक प्रकार की स्थानीय मान (positional value) पद्यति है जिसमे किसी अंक की कीमत उसके स्थान के अनुसार होती है – जैसे 526 संख्या में 5 का मान 500 है, 2 का मान 20 है तथा 6 का मान 6 है | (सैकड़ा(hundreds) – दहाई(tens) – इकाई(ones) के अनुसार 6 इकाई स्थान पर है , 2 दहाई स्थान पर है तथा 5 सैकड़े के स्थान पर है)
- हम उपरोक्त उदाहरण को निन्वत भी लिख सकते हैं -
- $526 = 5 \times 10^2 + 2 \times 10^1 + 6 \times 10^0$
- $25.32 = 2 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 2 \times 10^{-2}$
- सबसे बाएं वाला अंक MSD (Most Significant Digit) कहलाता है |
- सबसे दायें वाला अंक LSD (Least Significant Digit) कहलाता है |

Binary Number System

- Binary system 2 अंकों या संकेतों से मिलकर बना है जो की निम्न हैं | इन्हें bit कहा जाता है | 0,1
- इस system का आधार 2 होता है तथा इसे निम्न प्रकार दर्शाते हैं -
  - $(1001010101)_2$
- Digital systems में decimal system का प्रयोग कर पाना नामुमकिन है | अतः कंप्यूटर सिस्टम के लिए binary system का प्रयोग सार्थक होता है क्योंकि दो voltage लेवल को बनाये रखने के लिए circuit बनाना अत्यंत आसान होता है |
- हम उदाहरण से binary number लिखना सीख सकते हैं -
- $1010 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
- $10.11 = 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
- सबसे बाएं वाला अंक MSB (Most Significant Bit) कहलाता है |
- सबसे दायें वाला अंक LSB (Least Significant Bit) कहलाता है |

Octal Number System

- Octal system 8 अंकों या संकेतों से मिलकर बना है जो की निम्न हैं |
  - 0,1,2,3,4,5,6,7
- इस system का आधार 8 होता है तथा इसे निम्न प्रकार दर्शाते हैं -
  - $(1675)_8$
- हम उदाहरण से Octal Number लिखना सीख सकते हैं -
- $147 = 1 \text{ X } 8^2 + 4 \text{ X } 8^1 + 7 \text{ X } 8^0$
- $13.46 = 1 \text{ X } 8^1 + 3 \text{ X } 8^0 + 4 \text{ X } 8^{-1} + 6 \text{ X } 8^{-2}$

Hexadecimal Number System

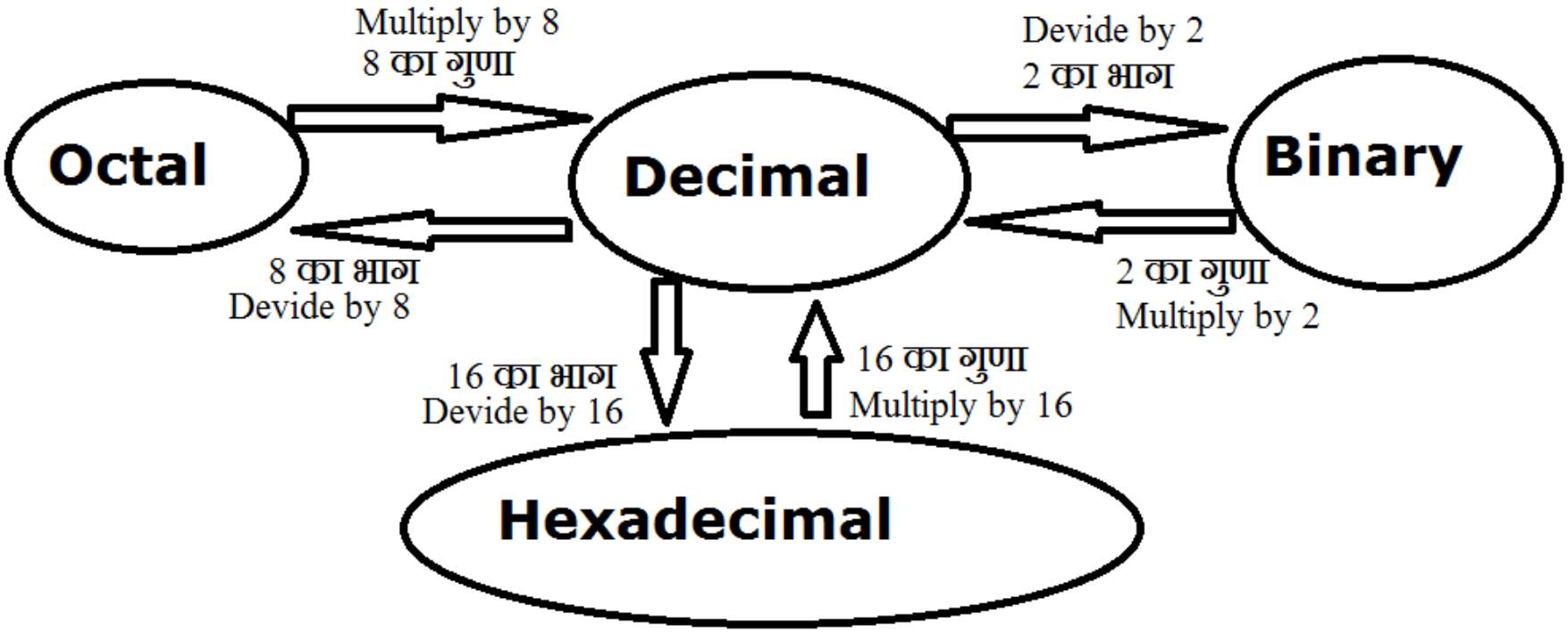
- Hexadecimal system 16 अंकों या संकेतों से मिलकर बना है जो की निम्न हैं | 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- यहाँ A=10, B=11, C=12, D=13, E=14, F=15 होगा |
- इस system का आधार 16 होता है तथा इसे निम्न प्रकार दर्शाते हैं -
  - $(16A7B5)_{16}$
- हम उदाहरण से Hexadecimal Number लिखना सीख सकते हैं -
- $1A7 = 1 \text{ X } 16^2 + 10 \text{ X } 16^1 + 7 \text{ X } 16^0$
- $1B.A6 = 1 \text{ X } 16^1 + 11 \text{ X } 16^0 + 10 \text{ X } 16^{-1} + 6 \text{ X } 16^{-2}$

विभिन्न Number System के मध्य सम्बन्ध

Hexadecimal	Octal	Decimal	Binary
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	10	8	1000
9	11	9	1001
A	12	10	1010
B	13	11	1011
C	14	12	1100
D	15	13	1101
E	16	14	1110
F	17	15	1111

Number Conversion

- Numbers को एक पद्यति से दूसरी पद्यति में बदलने के लिए निम्न चित्र का अनुसरण करें |



Decimal से Binary

- इसके लिए decimal संख्या में 2 से भाग देते हैं और शेषफल को एक क्रम में लगा देते हैं फिर भागफल में 2 का भाग लगते हैं और शेषफल को पुनः क्रम में लगा देते हैं यह कार्य तब तक करते जाते हैं जब तक भाग देते-देते भागफल 0 न आजाये | शेषफल के क्रम में लगाने से binary तैयार हो जाती है | (क्योंकि 2 से किसी भी संख्या में भाग दें तो शेषफल 0 या 1 ही आएगा |)
- उदहारण के लिए $(259)_{10}$ को binary में बदलते हैं -

2	259	Remainder	
2	129	1	LSB
2	64	1	
2	32	0	
2	16	0	
2	8	0	
2	4	0	
2	2	0	
2	1	0	
	0	1	MSB

$$(259)_{10} = (100000011)_2$$

इसका एक और अन्य तरीका भी हो सकता है |

Decimal से Binary

- इसके लिये आप 2^n की एक तालिका (table) बना लें जो बहुत आसान है |

2^n	n	2^{-n}
1	0	1.00000000000000000000000000000000
2	1	0.50000000000000000000000000000000
4	2	0.25000000000000000000000000000000
8	3	0.12500000000000000000000000000000
16	4	0.06250000000000000000000000000000
32	5	0.03125000000000000000000000000000
64	6	0.01562500000000000000000000000000
128	7	0.00781250000000000000000000000000
256	8	0.00390625000000000000000000000000
512	9	0.00195312500000000000000000000000
1024	10	0.00097656250000000000000000000000
2048	11	0.00048828125000000000000000000000
4096	12	0.00024414062500000000000000000000
8192	13	0.00012207031250000000000000000000
16384	14	0.00006103515625000000000000000000
32768	15	0.00003051757812500000000000000000
65536	16	0.00001525878906250000000000000000
131072	17	0.00000762939453125000000000000000
262144	18	0.00000381469726562500000000000000
524288	19	0.00000190734863281250000000000000
1048576	20	0.00000095367431640625000000000000
2097152	21	0.00000047683715820312500000000000
4194304	22	0.00000023841857910156200000000000
8388608	23	0.00000011920928955078100000000000

2^n	n	2^{-n}
16777216	24	0.00000005960464477539060000000
33554432	25	0.00000002980232238769530000000
67108864	26	0.00000001490116119384770000000
134217728	27	0.00000000745058059692383000000
268435456	28	0.00000000372529029846191000000
536870912	29	0.00000000186264514923096000000
1073741824	30	0.00000000093132257461547900000

अब $(200)_{10}$ को binary में बदलते हैं |

हल: 2^n की तालिका में 200 से नीचे सबसे बड़ा number है 128

128 = 10000000

200 – 128= 72 से नीचे 64 = 1000000

72 – 64 = 8 से नीचे 8 = 1000

कुल जोड़ 200 = 11001000

अतः $(200)_{10} = (11001000)_2$

Decimal से Binary

- यदि संख्या दशमलव में हो तो दशमलव के पूर्व का तरीका पूर्ववत रहेगा| दशमलव के बाद की संख्यामे 2 का गुणा करने पर (.) से पहले वाला अंक binary के साथ (.) लगा कर लिख देते हैं | और यह प्रक्रिया tab तक दोहराते हैं जब तक इच्छित उत्तर न मिल जाए |
- उदहारण के लिए $(259.25)_{10}$ को binary में बदलते हैं -

2	259	Remainder	
2	129	1	LSB
2	64	1	
2	32	0	
2	16	0	
2	8	0	
2	4	0	
2	2	0	
2	1	0	
	0	1	MSB

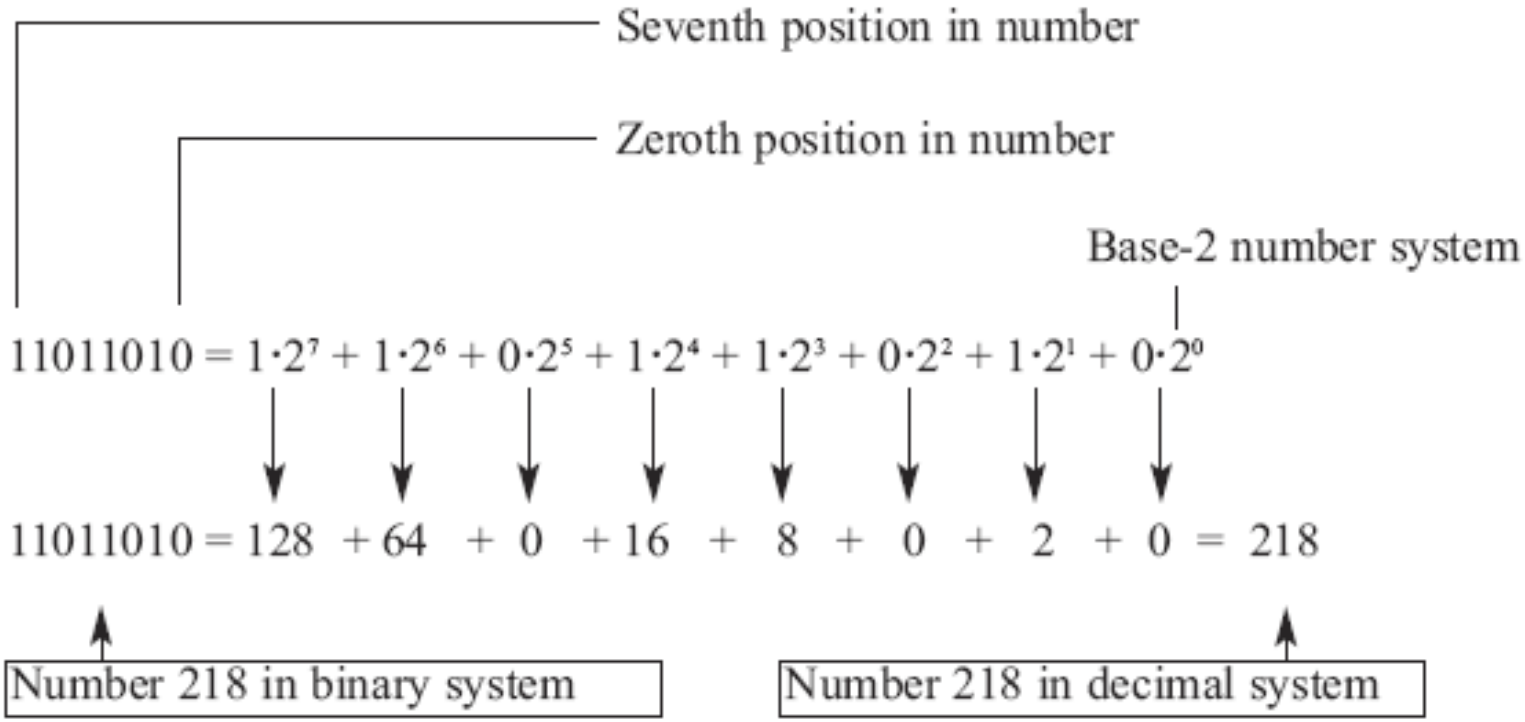
$(259.25)_{10} = (100000011.01)_2$

0.25 X 2 = **0**.50 यहाँ से दशमलव के पहले का 0 उठाकर binary के साथ लिख देंगे|

0.50 X 2 = **1**.00 यहाँ से दशमलव के पहले का 1 उठाकर binary के साथ लिख देंगे|

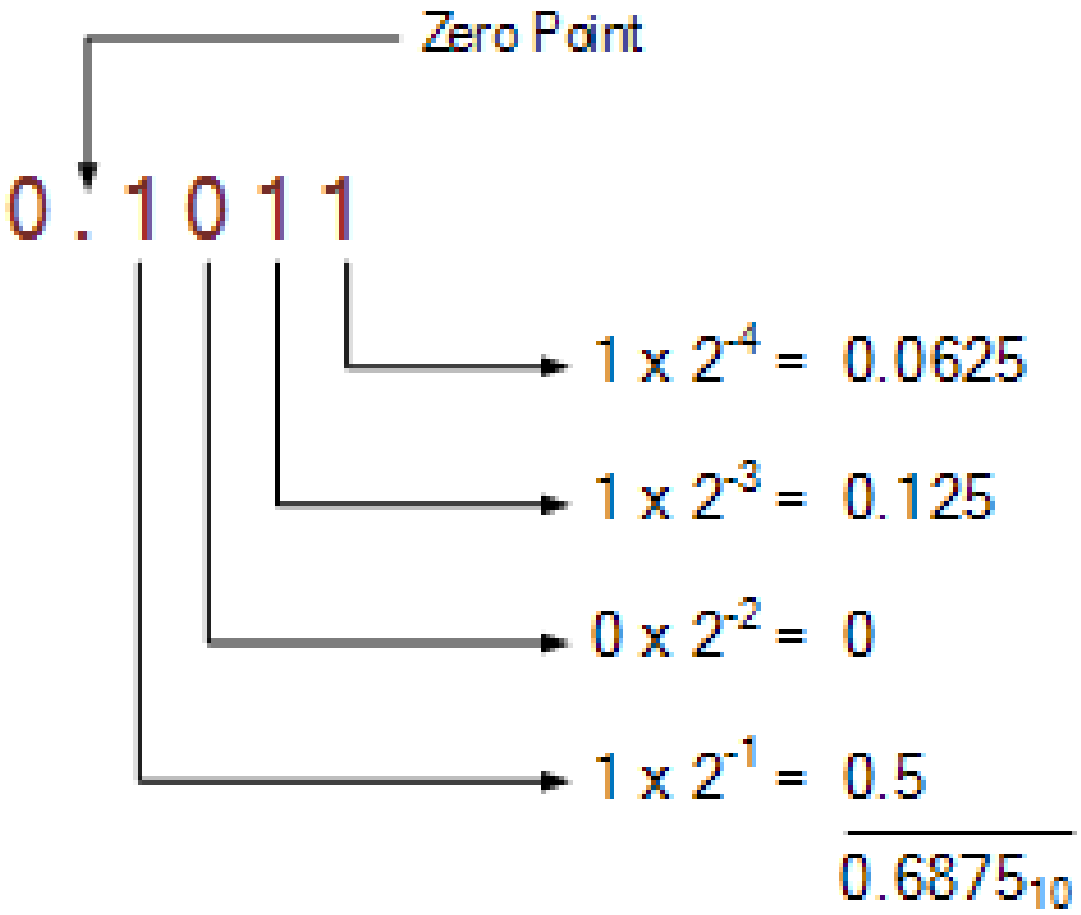
Binary से Decimal

- इसके लिए पहले binary के प्रत्येक bit में 2 का उसके स्थानानुसार घात(Power) लगा कर गुणा करते हैं |
- फिर जो expression आये उसे गणितीय विधि से हल कर लेते हैं |
- उदाहरण के लिए हम $(11011010)_2$को decimal में बदलते हैं



Binary से Decimal

- यदि binary दशमलव के साथ हो तो उसको बदलने के लिए निम्न उदाहरण पर ध्यान दें |



Decimal से Octal

- इसके लिए decimal संख्या में 8 से भाग देते हैं और शेषफल को एक क्रम में लगा देते हैं फिर भागफल में 8 का भाग लगते हैं और शेषफल को पुनः क्रम में लगा देते हैं यह कार्य तब तक करते जाते हैं जब तक भाग देते-देते भागफल 0 न आजाये | शेषफल के क्रम में लगाने से octal तैयार हो जाता है | (क्योंकि 8 से किसी भी संख्या में भाग दें तो शेषफल 0 से 7 के बीच में ही आएगा |)
- उदहारण के लिए $(239)_{10}$ को Octal में बदलते हैं -

8	239
8	29
8	3
	0

4 ← First Remainder

5 ← Second Remainder

3 ← Third Remainder

Read Up

$(239)_{10} = (354)_8$

0.513 x 8 = **4**.104 — 4

0.104 x 8 = **0**.832 — 0

0.832 x 8 = **6**.656 — 6

0.656 x 8 = **5**.248 — 5

0.248 x 8 = **1**.984 — 1

$(0.513)_{10} = (0.40651...)_8$

Complete answer is $(152.512)_{10} = (230.40651...)_8$

Decimal Number: $(540.125)_{10}$

8	540	
8	67	4
8	8	3
8	1	0
	0	1

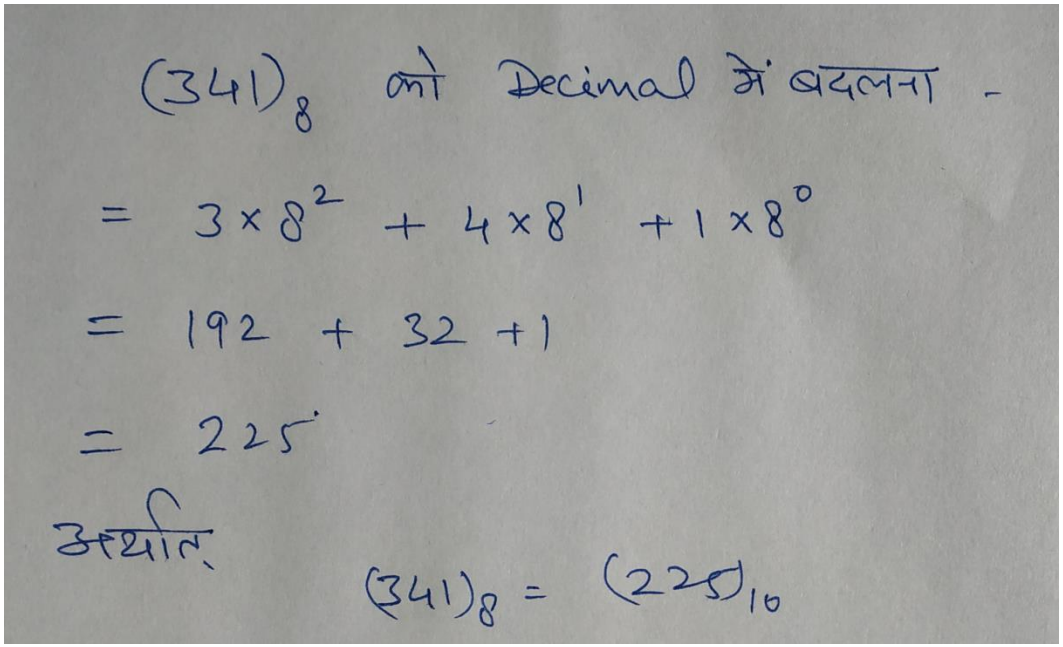
$0.125 \times 8 = 0$ with a carry of 1

$(0.125)_{10} = (0.1)_8$

$(540)_{10} = (1034.1)_8$ Octal Number

Octal से Decimal

- इसके लिए पहले Octal के प्रत्येक digit में 8 का उसके स्थानानुसार घात(Power) लगा कर गुणा करते हैं |
- फिर जो expression आये उसे गणितीय विधि से हल कर लेते हैं |
- उदाहरण के लिए हम $(341)_8$ को decimal में बदलते हैं

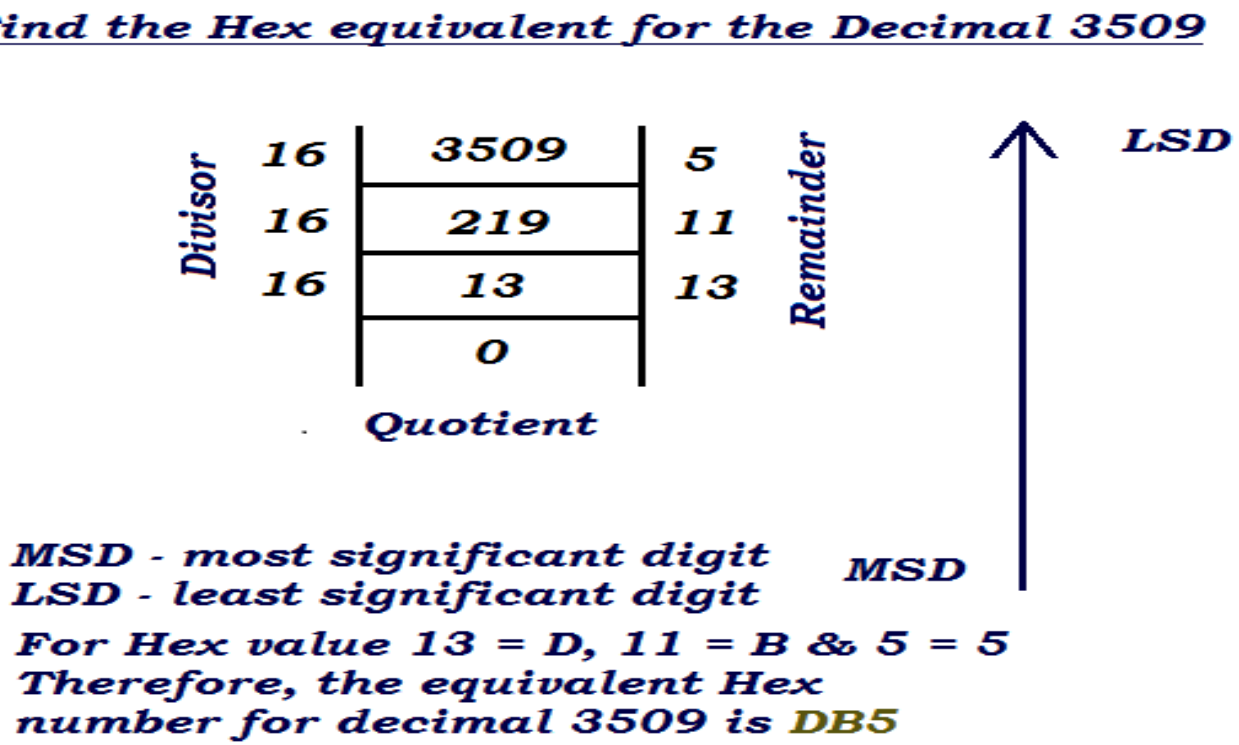


- एक अन्य उदाहरण -

$$24.6_8 = 2 \text{ X } 8^1 + 4 \text{ X } 8^0 + 6 \text{ X } 8^{-1} = 20.75_{10}$$

Decimal से Hexadecimal

- इसके लिए decimal संख्या में 16 से भाग देते हैं और शेषफल को एक क्रम में लगा देते हैं फिर भागफल में 16 का भाग लगते हैं और शेषफल को पुनः क्रम में लगा देते हैं यह कार्य तब तक करते जाते हैं जब तक भाग देते-देते भागफल 0 न आजाये | शेषफल के क्रम में लगाने से Hexadecimal तैयार हो जाता है | (क्योंकि 16 से किसी भी संख्या में भाग दें तो शेषफल 0 से 15 के बीच में ही आएगा| 10 की जगह A, 11 की जगह B . . . लिख देंगे)



Hex से Decimal

- इसके लिए पहले Hex के प्रत्येक digit में 16 का उसके स्थानानुसार घात(Power) लगा कर गुणा करते हैं |
- फिर जो expression आये उसे गणितीय विधि से हल कर लेते हैं |
- उदाहरण के लिए हम निम्न को decimal में बदलते हैं

- $356_{16} = 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 = 768 + 80 + 6 = 854_{10}$
- $2AF16 = 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0$
- $= 512 + 160 + 15$
- $= 687_{10}$
- एक अन्य उदाहरण -

$56.08_{16} = 5 \times 16^1 + 6 \times 16^0 + 0 \times 16^{-1} + 8 \times 16^{-2}$

$= 80 + 6 + 0 + 8/256$

$= 86 + 0.03125$

$= 86.03125$

Octal ⟺ binary

- इसके लिए आप octal से decimal और फिर जो संख्या प्राप्त हो उसको binary में बदल सकते हैं |
- दूसरे तरीके के लिए हम निम्न तालिका पर ध्यान देते हैं

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$(235)_8 = (010011101)_2$

सिर्फ octal के बराबर का binary हर octal digit की जगह लिख देते हैं |

$(101110100)_2 = (564)_8$

तीन तीन के समूह बनाकर तालिका देखकर binary के सामने की octal डिजिट को क्रम से रख देते हैं |

Hex	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Hex ⟺ binary

- इसके लिए आप Hex से decimal और फिर जो संख्या प्राप्त हो उसको binary में बदल सकते हैं |
- दूसरे तरीके के लिए हम निम्न तालिका पर ध्यान देते हैं

$$(2A5)_{16} = (001010100101)_2$$

सिर्फ Hex के बराबर का binary हर Hex digit की जगह लिख देते हैं |

$$(000101110100)_2 = (174)_{16}$$

तीन तीन के समूह बनाकर तालिका देखकर binary के सामने की hex डिजिट को क्रम से रख देते हैं |

Octal ⟷ Hex

- इसके लिए आप octal तालिका से binary और फिर जो संख्या प्राप्त हो उसको Hex में बदल सकते हैं | और Hex से octal मे बदलने के लिए इसका ठीक उल्टा करना होगा |
- दूसरे तरीके के लिए हम निम्न तालिका पर ध्यान देते हैं

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$(347)_8 = (\quad)_{16}$ के बदलाव के लिए हमें निम्न कार्य करने होंगे
- पहले binary में बदलो
$(347)_8 = (011100111)_2$
अब binary में दाहिनी ओर से चार-चार bit के समूह बनाकर Hex में बदलो |

$(011100111)_2 = (E7)_{16}$

Hex	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Binary में Unsigned Integers

- एक unsigned integer कोई भी सकारात्मक संख्या या शून्य (0) हो सकती है |
- कोई भी negative (-ve) संख्या unsigned integer नहीं हो सकती|
- Unsigned Integer 0 से $2^n - 1$ तक हो सकती है |

n (bits)	न्यूनतम मान	अधिकतम मान
8	0	2^8-1 (=255)
16	0	2^{16}-1 (=65535)
32	0	2^{32}-1 (=4,294,967,295)
64	0	2^{64}-1 (=18,446,744,073,709,551,615)

Binary Addition

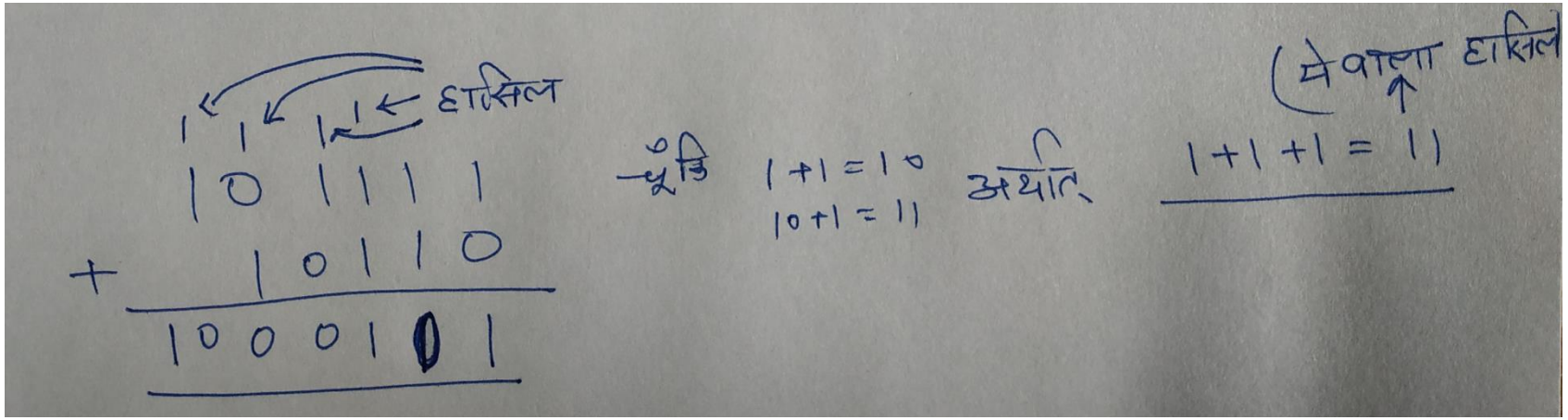
- Binary में योग के लिए निम्न बातों का ध्यान रखना होगा |

  0 + 0 = 0

  0 + 1 = 1

  1 + 0 = 1

  1 + 1 = 11 यहाँ यह ध्यान रखने योग्य है की इसमें 1 हासिल(Carry) मिल जाता है | जो कि अगले स्थान के bits के साथ जुड़ता है |



ASCII Code

- कोई भी कंप्यूटर कितने संकेतों या अक्षरों को समझ सकता है यह निर्भर करता है character set पर |
- Character set के अपने कुछ मानक हैं जिन्हें character set code कहते हैं | जैसे - ASCII, ISCII, UNICODE इत्यादि |
- ASCII (American Standard Code for Information Interchange) अधिकतर micro computers, mini computers और कुछ mainframe computers में किया जाता है |
- ASCII code दो रूपों में उपलब्ध है - ASCII – 7 और ASCII – 8
- ASCII – 7 code एक संकेत या charachter के लिए 7 bits का एक समूह प्रयोग करता है | जिसके द्वारा $2^7 = 128$ विभिन्न characters को प्रयोग कर सकते हैं |
- ASCII – 8 code एक संकेत या charachter के लिए 8 bits का एक समूह प्रयोग करता है | जिसके द्वारा $2^8 = 256$ विभिन्न characters को प्रयोग कर सकते हैं |

ASCII Code

Name	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

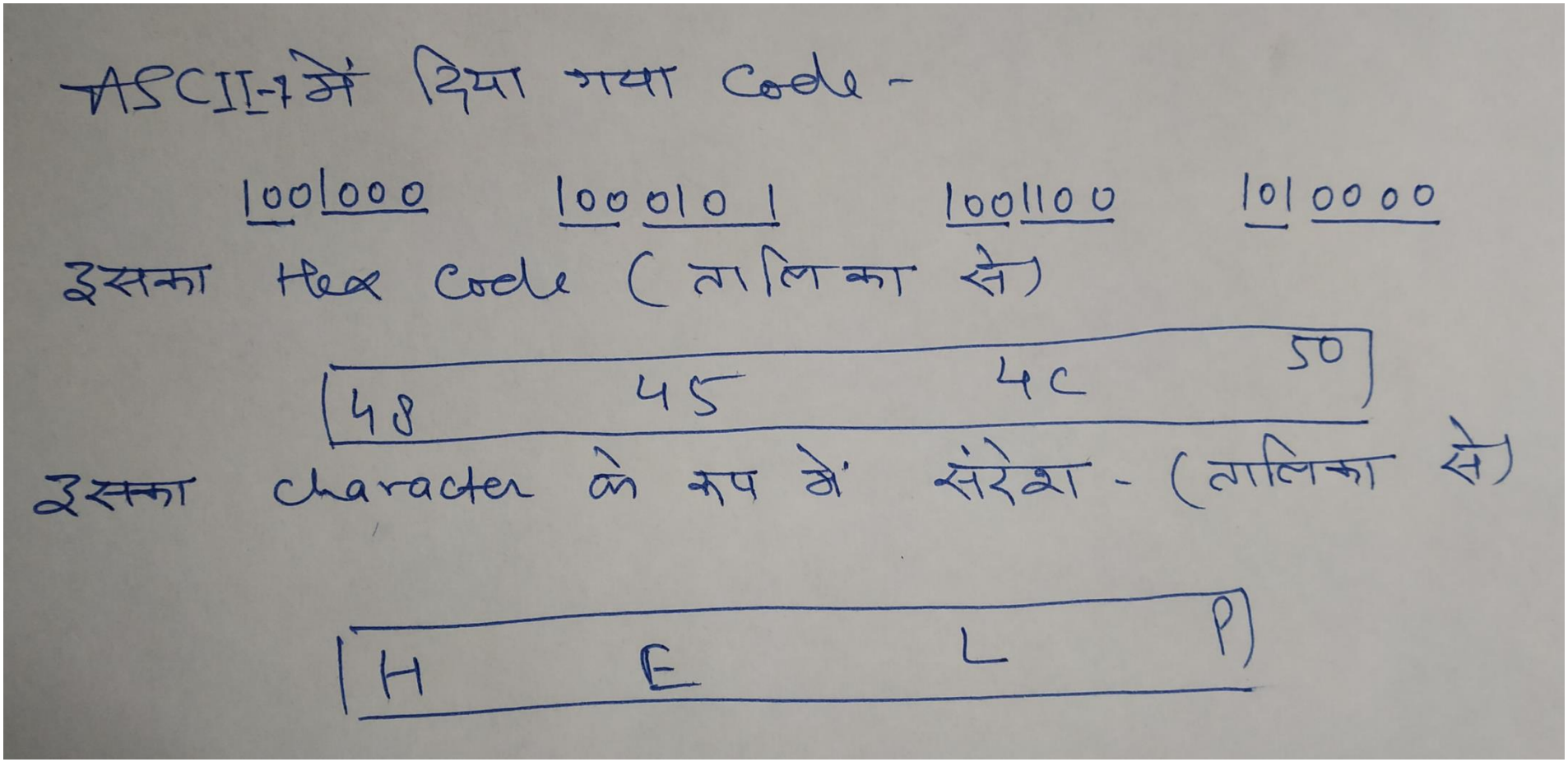
Name	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

Name	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	802
S	53	083
T	54	084
U	55	085
V	56	086

Name	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

इनकी सहायता से characters को पचाना जा सकता है |

ASCII Code


ASCII-7 में दिया गया Code -
1001000 1000101 1001100 1010000
इसका Hex Code (तालिका से)
48 45 4C 50
इसका character के रूप में संदेश - (तालिका से)
H E L P

ISCII Code

- यह Indian Standard Code for Information Interchange है जो की भारत में विकसित किया गया है |
- यह लगभग समस्त भारतीय भाषाओं के अक्षरों और संकेतों को पहचानता है |
- यह ASCII लिपियों के अलावा अन्य भारतीय भाषाओँ की प्रतिलिपियों को पहचान लेगा|
- यह भी 8 bit के समूह में कार्य करता है |

Unicode

- यह Universal Character Set होता है तथा 32 bit के समूह में एक संकेत या character को प्रदर्शित करता है|
- इसमें दुनिया की सभी भाषाओँ की प्रतिलिपि के संकेतों और अक्षरों को समावेशित करने की क्षमता है |
- जब तक Unicode विक्सित नहीं हुआ था तब तक कई प्रकार के encoding system प्रचलन में थे |
- इन्टरनेट पर Unicode की वजह से कई प्रकार की भाषा सम्बन्धी समस्याएं दूर हो गयीं |

Unicode अपने characters को दर्शाने के लिए विभिन्न encoding systems का प्रयोग करता है जैसे –

1. UTF – 8 (Unicode Transformation Format) – 8
   a) UTF – 8 – 1 Octet (8 bits) Representation
   b) UTF – 8 – 2 Octet (16 bits) Representation
   c) UTF – 8 – 3 Octet (24 bits) Representation
   d) UTF – 8 – 4 Octet (32 bits) Representation
2. UTF – 32

धन्यवाद

और अधिक पाठ्य-सामग्री हेतु निम्न लिंक पर क्लिक करें -

www.pythontrends.wordpress.com