

कम्प्यूटर, डेटा के निरूपण के लिए बाइनरी भाषा का प्रयोग करता है। ये बाइनरी भाषा 0 और 1 से मिलकर बनी होती है। उपयोगकर्ता कम्प्यूटर को जो भी डेटा या निर्देश इनपुट के रूप में देता है या कम्प्यूटर से जो भी आउटपुट प्राप्त करता है, वह अक्षर, संख्या, संकेत, ध्वनि या वीडियो के रूप में होता है। इन सभी डेटा या निर्देशों को पहले बाइनरी भाषा में बदलना पड़ता है अर्थात् डेटा को 0 और 1 के रूप में प्रस्तुत करना पड़ता है। इस प्रक्रिया को 'डेटा निरूपण' कहते हैं।

संख्या पद्धति (Number System)

संख्या पद्धति के अन्तर्गत विभिन्न प्रकार की संख्याओं का समूह होता है, जिसका प्रयोग कम्प्यूटर में किसी डेटा/निर्देश को व्यक्त करने के लिए करते हैं। कम्प्यूटर को डेटा या निर्देश अलग-अलग संख्या पद्धति में दिया जाता है और कम्प्यूटर अलग-अलग संख्या पद्धति में डेटा को निरूपित करता है, किन्तु आन्तरिक रूप से किसी कार्य को करने के लिए कम्प्यूटर बाइनरी भाषा का ही प्रयोग करता है।

संख्या पद्धति के प्रकार (Types of Number System)

कम्प्यूटर सिस्टम द्वारा प्रयोग की जाने वाली संख्या पद्धतियाँ मुख्यतः चार प्रकार की होती हैं

(1) बाइनरी संख्या पद्धति

2. दशमलव या दशमिक संख्या प्रणाली (Decimal Number System)

दैनिक जीवन में प्रयुक्त होने वाली संख्या पद्धति को दशमिक या दशमलव संख्या प्रणाली कहा जाता है। इस संख्या प्रणाली में 0, 1, 2,

3, 4, 5, 6, 7, 8, और 9 ये दस संकेत मान (Symbol Value) होते हैं। जिस कारण इस संख्या प्रणाली का आधार 10 होता है। दशमलव प्रणाली का स्थानीय मान (Positional Value) संख्या के दाई से बाई दिशा में आधार (Base) 10 की घात की वृद्धि के क्रम के रूप में होता है। दशमलव प्रणाली के स्थानीय मानों को निम्न प्रकार से समझा जा सकता है।

स्थान (दाई से बाई ओर) (Position)	स्थानीय मान (Positional Value)	10 की घात (Power of 10)
1	इकाई (1)	10^0
2	दहाई (10)	10^1
3	सैंकड़ा (100)	10^2
4	हजार (1000)	10^3
5	दस हजार (10,000)	10^4
6	लाख (1,00,000)	10^5

- (2) दशमलव संख्या पद्धति
- (3) ऑक्टल संख्या पद्धति
- (4) हेक्साडेसीमल संख्या पद्धति

1. बाइनरी या द्वि-आधारी संख्या प्रणाली (Binary Number System)

इस संख्या प्रणाली में केवल दो अंक होते हैं- 0 (शून्य) और 1 (एक)। जिस कारण इसका आधार 2 होता है। इसलिए इसे द्वि-आधारी या बाइनरी संख्या प्रणाली कहा जाता है। जिस प्रणाली में कम्प्यूटर की मुख्य पद्धति बनती है, वह स्विच की तरह कार्य करती है। स्विच की केवल दो स्थितियाँ होती हैं- ऑन (ON) तथा ऑफ (OFF)। इसके अलावा कोई तीसरी स्थिति सम्भव नहीं है। इस आधार पर कम्प्यूटर संख्या प्रणाली में

3. ऑक्टल या अष्ट-आधारी संख्या प्रणाली

29

(Octal Number System)

ऑक्टल संख्या प्रणाली में 0, 1, 2, 3, 4, 5, 6 और 7 इन आठ अंको का प्रयोग किया जाता है। जिस कारण इसका आधार 8 होता है। इन अंकों के मुख्य मान दशमलव संख्या प्रणाली की तरह ही होते हैं। ऑक्टल संख्या प्रणाली इसलिए सुविधाजनक है, क्योंकि इसमें किसी भी बाइनरी संख्या को छोटे रूप में लिख सकते हैं।

उदाहरण के लिए

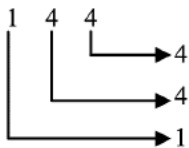
ऑक्टल (Octal)	बाइनरी (Binary)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

इस प्रणाली का प्रयोग मुख्यतः माइक्रो कम्प्यूटरों में किया जाता है। आधार 8 होने का कारण अष्टमिक संख्या प्रणाली में अंकों के स्थानीय मान दाईं ओर से बाईं ओर क्रमशः आठ गुने होते जाते हैं। अर्थात् $8^0, 8^1, 8^2, 8^3, \dots$ आदि तथा ऑक्टल बिन्दु दाईं ओर क्रमशः

$8^{-1}, 8^{-2}, 8^{-3}, \dots$ आदि होते हैं।

उदाहरण के लिए ऑक्टल संख्या $(144)_8$ का अर्थ है

संकेत मान	अंको के मान	स्थानीय मान (दाईं से बाईं दिशा में)	ऑक्टल संख्या के अंको का मान
-----------	-------------	--	-----------------------------



$$8^0 = 1$$

$$8^1 = 8$$

$$8^2 = 64$$

$$4 \times 1 = 4$$

$$4 \times 8 = 32$$

$$1 \times 64 = \underline{64}$$

$$\underline{100}$$

$$\therefore (144)_8 = (100)_{10}$$

नोट किसी भी संख्या को प्रदर्शित करते समय संख्या को कोष्ठकों में बन्द करके दाईं ओर नीचे उसका आधार (Base) लिख दिया जाता है, जिससे हमें पता चलता है कि दी गई संख्या किस प्रणाली की है।

4. हेक्सा-डेसीमल या षट्दशमिक संख्या प्रणाली (Hexadecimal Number System)

हेक्सा-डेसीमल शब्द दो अक्षरों से मिलकर बना है हेक्सा + डेसीमल। हेक्सा से तात्पर्य छः तथा डेसीमल से तात्पर्य दस से होता है। अतः इस संख्या प्रणाली में कुल सोलह [16] (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) अंक होते हैं। इसके मुख्य मान क्रमशः 0 से 15 तक होते हैं, परन्तु 10, 11, 12, आदि को दो अलग-अलग अंक न समझ लिया जाए, इसलिए हम अंकों 10, 11, 12, 13, 14 और 15 के स्थान पर क्रमशः A, B, C, D, E, तथा F अक्षर लिखते हैं। इस प्रकार इस प्रणाली में दस अंक तथा छः वर्णों का प्रयोग किया जाता है, जो निम्नलिखित हैं

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

हेक्सा-डेसीमल संख्या प्रणाली में अंको के स्थानीय मान दाईं ओर से बाईं

ओर 16 के गुणकों में बढ़ते हैं;

जैसे- $16^0, 16^1, 16^2, 16^3, \dots$ आदि इसी प्रकार हेक्सा-डेसीमल बिन्दु के बाद इसके स्थानीय। मान 16 के गुणकों में घटते हैं; जैसे- $16^{-1}, 16^{-2}, 16^{-3}, 16^{-4}, \dots$ आदि।

उदाहरण के लिए हेक्सा-डेसीमल $(F6A4)_{16}$ का अर्थ है

संकेत मान	अंको के मान	स्थानीय मान (दाईं से बाईं दिशा में)	ऑक्टल संख्या के अंको का मान
-----------	-------------	--	-----------------------------

F	6	A	4			
				→ 4	$16^0 = 1$	$1 \times 4 = 4$
			→ 10		$16^1 = 16$	$16 \times 10 = 160$
		→ 6			$16^2 = 256$	$256 \times 6 = 1536$
	→ 15				$16^3 = 4096$	$4096 \times 15 = \underline{61440}$

$$\therefore (F6A4)_{16} = (63140)_{10}$$

हम किसी भी संख्या को किसी भी संख्या प्रणाली में लिख सकते हैं।

उदाहरण के लिए, पहली 16 संख्याओं को दशमलव, ऑक्टल, बाइनरी तथा हेक्सा-डेसीमल में निम्न प्रकार लिखा जाता है-

दशमलव	बाइनरी	ऑक्टल	हेक्सा-डेसीमल
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

30

संख्या पद्धतियों का आपस में परिवर्तन (Conversion between Number System)

एक पद्धति से दूसरी संख्या पद्धतियों में परिवर्तन आवश्यक होता है, क्योंकि उपयोगकर्ता (User) द्वारा इनपुट किया गया डेटा दशमलव संख्या पद्धति में होता है। जिसके बाद कम्प्यूटर इस इनपुट किए गए डेटा को उस संख्या पद्धति

में बदल देता है, जिसमें उसे सुविधा हो। एक डिजिटल कम्प्यूटर सिस्टम में एक समय में तीन या चार संख्या पद्धतियों का प्रयोग किया जाता है। इसी कारण से संख्या पद्धतियों को आपस में परिवर्तित कराया जाता है।

विभिन्न संख्या पद्धतियों को आपस में परिवर्तित करने की चर्चा आगे की गई है

1. दशमलव का अन्य संख्या पद्धतियों में परिवर्तन (Conversion of Decimal System to Other Number Systems)

दशमलव (पूर्णांक) को बाइनरी में बदलने के लिए निम्नलिखित स्टेप्स का अनुसरण करना चाहिए

स्टेप 1 दशमलव को बाइनरी संख्या में परिवर्तित करने के लिए बाइनरी संख्या के आधार 2 से दशमलव संख्या को भाग देते हैं।

स्टेप 2 इसमें जो शेषफल आता है, उसे दाएँ लिख लेते हैं तथा जो भागफल आता है, उसे फिर से 2 से भाग देते हैं। तत्पश्चात् यही क्रिया दोहराते हैं।

स्टेप 3 यह प्रक्रिया तब तक चलती रहती है, जब तक कि भागफल शून्य न हो जाए।

स्टेप 4 प्राप्त शेषफलों को नीचे से ऊपर के क्रम में लिखा जाता है। यही शेषफल दिए गए दशमलव संख्या के तुल्य बाइनरी संख्या है।

(ii) दशमलव (भिन्नांक) का बाइनरी में परिवर्तन (Conversion of Decimal Fraction into Binary)

दशमलव (भिन्नांक) को बाइनरी में बदलने के लिए निम्नलिखित स्टेप्स

का अनुसरण करना चाहिए

स्टेप 1 दशमलव बिन्दु वाली दशमलव संख्या को बाइनरी संख्या में बदलने के लिए हम दशमलव संख्या को बाइनरी के आधार चिन्ह 2 से गुणा करते हैं।

स्टेप 2 प्राप्त पूर्णांक (1 या 0) को दाईं ओर लिखते हैं तथा प्राप्त भिन्नांक को फिर से 2 से गुणा करते हैं। तत्पश्चात् यही क्रिया दोहराते हैं।

स्टेप 3 यही प्रक्रिया तब तक चलती है, जब तक या तो भिन्नांक 0 रह जाता है या इच्छित स्थानों तक बिट भर जाते हैं।

स्टेप 4 ये पूर्णांक बाइनरी बिन्दु के पश्चात् ऊपर से नीचे के क्रम में लिखे जाते हैं।

उदाहरण के लिए

दशमलव संख्या $(53.6875)_{10}$ को बाइनरी में बदलना।

53	0.6875
↓	↓
पूर्णांक	भिन्नांक
(Integer)	(Fraction)

पूर्णांक भाग 53 का बाइनरी परिवर्तन

$$53 \div 2 = 26$$

शेष 1

$$26 \div 2 = 13$$

शेष 0

$$13 \div 2 = 6$$

शेष 1

$$6 \div 2 = 3$$

शेष 0

$$3 \div 2 = 1$$

शेष 1

$$1 \div 2 = 0$$

शेष 1



$$\therefore (53)_{10} = (110101)_2 \dots \dots \dots (i)$$

तथा भिन्नांक भाग 0.6875 का बाइनरी परिवर्तन

$$0.6875 \times 2 = 1.3750$$

पूर्णांक 1

$$0.3750 \times 2 = 0.7500$$

पूर्णांक 0

$$0.7500 \times 2 = 1.5000$$

पूर्णांक 1

$$0.5000 \times 2 = 1.0000$$

पूर्णांक 1



$$\therefore (0.6875)_{10} = (0.1011)_2 \dots \dots \dots (ii)$$

समी (i) व (ii) को जोड़ने पर

$$(53.6875)_{10} = (110101.1011)_2$$

(iii) दशमलव (पूर्णांक) का ऑक्टल में परिवर्तन (Conversion of Decimal Integer to Octal)

दशमलव संख्या को ऑक्टल में बदलने की विधि ठीक उसी प्रकार होती है, जिस प्रकार दशमलव संख्या को बाइनरी में बदलने की होती है। अन्तर केवल यही होता है कि जहाँ दशमलव संख्या को बाइनरी में बदलने के लिए 2 से बार-बार भाग देते हैं, वहीं ऑक्टल में बदलने के

लिए 8 से बार-बार भाग देते हैं।

(iv) दशमलव (भिन्नांक) का ऑक्टल में परिवर्तन (Conversion of Decimal Fraction to Octal)

दशमलव भिन्नांक को ऑक्टल में बदलने के लिए निम्नलिखित स्टेप्स का अनुसरण करना चाहिए

31

स्टेप 1 दशमलव बिन्दु वाली दशमलव संख्या को ऑक्टल संख्या में बदलने के लिए हम दशमलव संख्या को ऑक्टल के आधार चिन्ह 8 से गुणा करते हैं।

स्टेप 2 प्राप्त पूर्णांक (1 या 0) को दाईं ओर लिखते हैं तथा भिन्नांक को फिर से 8 से भाग देते हैं। तत्पश्चात् यही क्रिया दोहराते हैं।

स्टेप 3 यही प्रक्रिया तब तक चलती है, जब तक या तो भिन्नांक 0 रह जाता है या इच्छित स्थानों तक बिट भर जाते हैं।

स्टेप 4 ये पूर्णांक ऑक्टल बिन्दु के पश्चात् ऊपर से नीचे के क्रम में लिखे जाते हैं।

उदाहरण के लिए

दशमलव संख्या $(423.03125)_{10}$ का ऑक्टल में परिवर्तन

423	0.03125
↓	↓
पूर्णांक	भिन्नांक
(Integer)	(Fraction)

पूर्णांक भाग $(423)_{10}$ का ऑक्टल में परिवर्तन

$$423 \div 8 = 52$$

$$\text{शेष} \quad 7 \quad \uparrow$$

$$52 \div 8 = 6$$

$$\text{शेष} \quad 4 \quad \uparrow$$

$$6 \div 8 = 0$$

$$\text{शेष} \quad 6 \quad \uparrow$$

शेषों को नीचे से ऊपर के क्रम में लिखने पर = 647

$$(423)_{10} = (647)_{10} \quad \dots (i)$$

तथा भिन्नांक भाग $(0.03125)_{10}$ का ऑक्टल में परिवर्तन

$$0.03125 \times 8 = 0.250 \quad \text{पूर्णांक} \quad 0 \quad \downarrow$$

$$0.250 \times 8 = 2.00 \quad \text{पूर्णांक} \quad 2 \quad \downarrow$$

$$\therefore (0.03125)_{10} = (0.02)_8 \dots (ii)$$

समी (i) व (ii) को जोड़ने पर

$$(423.03125)_{10} = (647.02)_8$$

(v) दशमलव (पूर्णांक) का हेक्सा-डेसीमल में परिवर्तन
(Conversion of Decimal Integer into Hexa-Decimal)

दशमलव संख्या को हेक्सा-डेसीमल में बदलने की विधि ठीक उसी प्रकार होती है, जिस प्रकार दशमलव संख्या को बाइनरी में बदलने की होती है। अन्तर केवल यही होता है कि इसमें 2 की जगह 16 से बार-बार भाग देते हैं।

(vi) दशमलव (भिन्नांक) का हेक्सा-डेसीमल में परिवर्तन
(Conversion of Decimal Fraction into Hexa-decimal)

दशमलव (भिन्नांक) को हेक्सा-डेसीमल में बदलने के लिए निम्नलिखित स्टेप्स का अनुसरण करना चाहिए

स्टेप 1 दशमलव बिन्दु वाली दशमलव संख्या को हेक्सा-डेसीमल

संख्या में बदलने के लिए हम दशमलव संख्या को हेक्सा-डेसीमल के आधार चिन्ह 16 से गुणा करते हैं।

स्टेप 2 प्राप्त पूर्णांक (1 या 0) को दाईं ओर लिखते हैं और भिन्नांक को फिर से 16 से भाग देते हैं। तत्पश्चात् यही क्रिया दोहराते हैं।

स्टेप 3 यही प्रक्रिया तब तक चलती है, जब तक या तो भिन्नांक 0 रह जाता है या इच्छित स्थानों तक बिट भर जाते हैं।

स्टेप 4 ये पूर्णांक हेक्सा-डेसीमल बिन्दु के पश्चात् ऊपर से नीचे के क्रम में लिखे जाते हैं।

उदाहरण के लिए

दशमलव संख्या $(2863.225)_{10}$ को हेक्सा-डेसीमल में बदलना

$$\begin{array}{cc} 2863 & 0.225 \\ \downarrow & \downarrow \\ \text{पूर्णांक} & \text{भिन्नांक} \\ \text{(Integer)} & \text{(Fraction)} \end{array}$$

पूर्णांक भाग $(2863)_{10}$ का हेक्सा-डेसीमल में परिवर्तन

$2863 \div 16 = 178$	शेष	$15 = F$	\uparrow
$178 \div 16 = 11$	शेष	2	
$11 \div 16 = 0$	शेष	$11 = B$	

$$\therefore (2863)_{10} = (B2F)_{16} \quad \dots(i)$$

तथा भिन्नांक भाग $(0.225)_{10}$ का हेक्सा-डेसीमल परिवर्तन

$0.225 \times 16 = 3.6$	पूर्णांक	3	\uparrow
-------------------------	----------	---	------------

$$0.6 \times 16 = 9.6$$

पूर्णांक 9

$$\therefore (0.225)_{10} = (0.39)_{16} \quad \dots(ii)$$

समी (i) वा (ii) को जोड़ने पर

$$(2863.225)_{10} = (B2F.39)_{16}$$

2. बाइनरी का अन्य संख्या पद्धतियों में परिवर्तन (Conversion of Binary Number System to other Number System)

(i) बाइनरी (पूर्णांक) का दशमलव में परिवर्तन (Conversion of Binary Integer into Decimal)

किसी बाइनरी संख्या को दशमलव प्रणाली में परिवर्तित करने के लिए हम प्रत्येक बिट को उनके स्थानीय मान से गुणा करके जोड़ देते हैं। प्राप्त संख्या ही बाइनरी का दशमलव में परिवर्तन होता है।

32

(ii) बाइनरी (भिन्नांक) का दशमलव में परिवर्तन (Conversion of Binary Fraction into Decimal)

बाइनरी बिन्दु से पहले अर्थात् पूर्णांक (Integer) के स्थानीय मान दाईं ओर से दशमलव में क्रमशः 1, 2, 4, 8, 16, 32, 64 आदि होते हैं तथा बाइनरी बिन्दु से दाईं ओर अर्थात् भिन्नांक (Fraction) की बिटों के स्थानीय मान क्रमशः $2^{-1}, 2^{-2}, 2^{-3} \dots \dots \dots$ आदि होते हैं

अर्थात् 0.5, 0.25, 0.125, 0.0625 आदि।

उदाहरण के लिए

बाइनरी संख्या $(1001101.01101)_2$ का दशमलव में बदलना

1001101 0.01101



पूर्णांक

भिन्नांक

पूर्णांक भाग 1001101 का दशमलव में परिवर्तन

संकेत मान	अंकों के मान	स्थानीय मान (दाई से बाई दिशा में)	बाइनरी संख्या के अंको का मान
1	1	$2^0 = 1$	$1 \times 1 = 1$
0	0	$2^1 = 2$	$0 \times 2 = 0$
1	1	$2^2 = 4$	$1 \times 4 = 4$
1	1	$2^3 = 8$	$1 \times 8 = 8$
0	0	$2^4 = 16$	$0 \times 16 = 0$
0	0	$2^5 = 32$	$0 \times 32 = 0$
1	1	$2^6 = 64$	$1 \times 64 = 64$
			योग <u>77</u>

$$\therefore (1001101)_2 = (77)_{10} \quad \dots(i)$$

तथा भिन्नांक भाग 0.01101 का दशमलव में परिवर्तन

$$0.01101 = 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}$$

$$= 0 + \frac{1}{4} + \frac{1}{8} + 0 + \frac{1}{32}$$

$$= 0 + 0.25 + 0.125 + 0 + 0.03125$$

$$= (0.40625)_{10}$$

$$\therefore (0.01101)_2 = (0.40625)_{10} \quad \dots (ii)$$

समी (i) व (ii) को जोड़ने पर

$$(1001101.01101)_2 = (77.40625)_{10}$$

(iii) बाइनरी (पूर्णांक) का ऑक्टल में परिवर्तन (Conversion of Binary Integer into Octal)

चूँकि, बाइनरी संख्या का आधार 2 तथा ऑक्टल संख्या का आधार 8 होता है और हम जानते हैं,

$$8 = 2 \times 2 \times 2 = 2^3$$

इसलिए, प्रत्येक ऑक्टल अंक को तीन बाइनरी अंको अर्थात् बिटों में बदल सकते हैं और लगातार 3 बिटों के प्रत्येक समूह को एक ऑक्टल संख्या में बदला जा सकता है। बाइनरी का ऑक्टल में परिवर्तन के लिए निम्नलिखित सूची का प्रयोग किया जाता है।

ऑक्टल अंक	बाइनरी अंक
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

इस सूची की सहायता से किसी भी बाइनरी संख्या को सरलता से ऑक्टल में परिवर्तित किया जाता है। इसके लिए निम्नलिखित स्टेप्स का अनुसरण करना चाहिए

स्टेप 1 दी गई बाइनरी संख्या के दाईं ओर से तीन-तीन बिटों के समूह बनाते हैं।

स्टेप 2 आवश्यकता पड़ने पर सबसे बाईं ओर तीन बिट का समूह पूरा करने के लिए शून्य अंक बढ़ाए जा सकते हैं।

स्टेप 3 तत्पश्चात् प्रत्येक समूह के अनुसार ऑक्टल अंक का मान रखते हैं।

(iv) बाइनरी (भिन्नांक) का ऑक्टल में परिवर्तन (Conversion of Binary Fraction into Octal)

बाइनरी (भिन्नांक) को ऑक्टल में परिवर्तित करने के लिए ऊपर दी गई सारणी का ही प्रयोग किया जाता है। इसके लिए निम्नलिखित स्टेप्स का अनुसरण करे

- स्टेप 1** दी गई संख्या में बाइनरी बिन्दु के बाईं ओर से तीन-तीन बिटों के समूह बनाते हैं।
- स्टेप 2** आवश्यकता पड़ने पर सबसे दाईं ओर शून्य अंक बढ़ाए जा सकते हैं।
- स्टेप 3** तत्पश्चात् प्रत्येक समूह के अनुसार ऑक्टल अंक का मान रखते हैं।

33

उदाहरण के लिए

बाइनरी संख्या $(1110111001.1000011)_2$ का ऑक्टल में परिवर्तन

तीन बिटों का समूह
पूरा करने के लिए

तीन बिटों का समूह
पूरा करने के लिए

$$\begin{array}{ccccccc} \uparrow & & & & & & \uparrow \\ \underbrace{001} & \underbrace{110} & \underbrace{111} & \underbrace{001} & . & \underbrace{100} & \underbrace{001} & \underbrace{100} \\ 1 & 6 & 7 & 1 & & 4 & 1 & 4 \end{array}$$

$$\therefore (1110111001.100001100)_2 = (1671.414)_8$$

(V) बाइनरी (पूर्णांक) का हेक्सा-डेसीमल में परिवर्तन
(Conversions of Binary Integer to Hexadecimal)

चूँकि, बाइनरी संख्या का आधार 2 होता है तथा हेक्सा-डेसीमल संख्या का आधार 16 होता है और हम जानते हैं

$$16 = 2 \times 2 \times 2 \times 2 = 2^4$$

अतः प्रत्येक हेक्सा-डेसीमल अंक को चार बाइनरी अंको अर्थात् बिटों में बदल सकते हैं और लगातार चार बिटों के प्रत्येक समूह को एक हेक्सा-डेसीमल संख्या में बदला जा सकता है। बाइनरी हेक्सा-डेसीमल में परिवर्तन के लिए निम्नलिखित सूची का प्रयोग किया जाता है, जिसमें सभी हेक्सा-डेसीमल संख्या के बराबर 4 बाइनरी अंक दिए गए हैं

हेक्सा-डेसीमल अंक	बाइनरी अंक
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

इस सूची की सहायता से किसी भी बाइनरी संख्या को सरलता से हेक्सा-डेसीमल में परिवर्तित किया जाता है। इसके लिए निम्नलिखित स्टेप्स का अनुसरण करें

स्टेप 1 दी गई बाइनरी संख्या के दाईं ओर से चार-चार बिटों के समूह बनाते हैं।

स्टेप 2 आवश्यकता पड़ने पर सबसे बाईं ओर शून्य अंक बढ़ाए जा सकते हैं।

स्टेप 3 तत्पश्चात् प्रत्येक समूह के अनुसार हेक्सा-डेसीमल का मान रखते हैं।

(vi) बाइनरी (भिन्नांक) का हेक्सा-डेसीमल में परिवर्तन (Conversion of Binary Fraction to Hexadecimal)

बाइनरी (भिन्नांक) को हेक्सा-डेसीमल में परिवर्तन करने के लिए ऊपर दी गई सारणी-2 का ही प्रयोग किया जाता है। इसके लिए निम्नलिखित स्टेप्स का अनुसरण करें

स्टेप 1 दी गई संख्या के बाईं ओर से (बाइनरी बिन्दु के तुरन्त बाद) चार-चार बिटों के समूह बनाते हैं।

स्टेप 2 आवश्यकता पड़ने पर सबसे दाईं ओर शून्य अंक बढ़ाए जा सकते हैं।

स्टेप 3 तत्पश्चात् प्रत्येक समूह के अनुसार हेक्सा-डेसीमल का मान रखते हैं।

उदहारण के लिए

बाइनरी संख्या $(110011011101.10110011)_2$ का हेक्सा-डेसीमल में परिवर्तन

$$\underline{1100} \underline{1101} \underline{1101} . \underline{1011} \underline{0011}$$

$$12 = C \quad 13 = D \quad 13 = D \quad 11 = B \quad 3$$

$$\therefore (110011011101.10110011)_2 = (CDD\overline{B}3)_{16}$$

नोट बाइनरी संख्या को हेक्सा-डेसीमल या ऑक्टल में बदलने के लिए पहले इन्हें डेसीमल संख्या में बदलकर फिर हेक्सा-डेसीमल या ऑक्टल संख्या में बदला जा सकता है।

3. ऑक्टल का अन्य संख्या पद्धतियों में परिवर्तन (Conversion of Octal to Other Number Systems)

(i) ऑक्टल (पूर्णांक) का दशमलव में परिवर्तन (Conversion of Octal Integer to Decimal)

ये विधि भी ठीक उसी प्रकार होती है, जिस प्रकार बाइनरी का दशमलव में परिवर्तन। इसमें 2 की जगह 8 के अंक के बढ़ते हुए क्रम में गुणा करते हैं।

34

(ii) ऑक्टल (भिन्नांक) का दशमलव में परिवर्तन (Conversion of Octal Fraction to Decimal)

ऑक्टल बिन्दु से पहले अर्थात् पूर्णांक (Integer) के स्थानीय मान दाईं ओर से दशमलव में क्रमशः 1, 8, 64, 512, 4096 आदि होते हैं तथा दाईं ओर अर्थात् भिन्नांक (Fraction) के अंको के स्थानीय मान क्रमशः $8^{-1}, 8^{-2}, 8^{-3}, \dots$ आदि होते हैं अर्थात् 0.125, 0.015625, 0.0019531 आदि।

उदाहरण के लिए ऑक्टल संख्या $(427.235)_8$ को दशमलव में बदलना

427	0.235
↓	↓
पूर्णांक	भिन्नांक

पूर्णांक भाग 427 का दशमलव में परिवर्तन

संकेत मान	अंकों के मान	स्थानीय मान (दाई से बाई दिशा में)	बाइनरी संख्या के अंको का मान
4	2	7	$7 \times 1 = 7$
		2	$2 \times 8 = 16$
		4	$4 \times 64 = 256$
			योग <u>279</u>

$$\therefore (427)_8 = (279)_{10}$$

तथा भिन्नांक भाग (0.235) का दशमलव में परिवर्तन

$$0.235 = 2 \times 8^{-1} + 3 \times 8^{-2} + 5 \times 8^{-3}$$

$$= 2 \times \frac{1}{8} + 3 \times \frac{1}{8^2} + 5 \times \frac{1}{8^3}$$

$$= 2 \times \frac{1}{8} + 3 \times \frac{1}{64} + 5 \times \frac{1}{512}$$

$$= \frac{128 + 24 + 5}{512} = \frac{157}{512} = 0.306640$$

$$\therefore (427.235)_8 = (279.306640)_{10} \text{ (लगभग)}$$

(iii) ऑक्टल (पूर्णांक) और (भिन्नांक) का बाइनरी में परिवर्तन (Conversion of Octal (Integer) and

(fraction) into Binary)

ऑक्टल का बाइनरी में परिवर्तन अत्यधिक सरल हैं। इसमें केवल प्रत्येक ऑक्टल अंक के स्थान पर उसके बराबर तीन बिटों का समूह लिख देते हैं।

उदाहरण के लिए

ऑक्टल संख्या (325.614) का बाइनरी में परिवर्तन

$$\therefore (325.614)_8 = (\underline{011} \underline{010} \underline{101} . \underline{110} \underline{001} \underline{100})_2$$

या $(325.614)_8 = (\underline{11010101} . \underline{1100011})_2$

नोट ऑक्टल को बाइनरी में बदलने के लिए दी गई संख्या को पहले दशमलव संख्या में बदलते हैं, जिसके बाद उसे बाइनरी में बदल सकते हैं।

(iv) ऑक्टल संख्या का हेक्सा-डेसीमल संख्या में परिवर्तन (Conversion of Octal to Hexa-Decimal)

ऑक्टल संख्या को हेक्सा-डेसीमल संख्या में बदलने के लिए निम्नलिखित विधि का प्रयोग करते हैं

1. दी गई ऑक्टल संख्या के प्रत्येक अंक को बाइनरी नम्बर में बदलते हैं।
 2. पुनः प्राप्त बाइनरी संख्या को हेक्सा-डेसीमल संख्या में बदलते हैं।
- उदाहरण के लिए (7632)₈ को हेक्सा-डेसीमल संख्या में बदलो।

$$\begin{array}{cccc} 7 & 6 & 3 & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 111 & 110 & 011 & 010 \\ \therefore (7632)_8 = (111110011010)_2 \\ 1111 & 1001 & 1010 \\ \downarrow & \downarrow & \downarrow \\ 15 & 9 & 10 \end{array}$$

$$\therefore (7632)_8 = (F_9A)$$

नोट ऑक्टल को हेक्सा-डेसीमल में बदलने के लिए एक और विधि का प्रयोग किया जा सकता है, जिसमें ऑक्टल को पहले दशमलव में फिर दशमलव से हेक्सा-डेसीमल में परिवर्तित किया जाता है।

4. हेक्सा-डेसीमल का अन्य संख्या पद्धतियों में परिवर्तन (Conversion of Hexadecimal to Other Number Systems)

(i) हेक्सा-डेसीमल का दशमलव में परिवर्तन (Conversion of Hexa-Decimal into Decimal)

यह विधि भी ठीक उसी प्रकार होती है, जिस प्रकार बाइनरी का दशमलव में परिवर्तन। इसमें 2 की जगह 16 के अंक के बढ़ते हुए क्रम में गुणा करते हैं।

उदाहरण के लिए हेक्सा-डेसीमल (1B4) दशमलव में परिवर्तन

संकेत मान	अंकों के मान	स्थानीय मान (दाई से बाई दिशा में)	बाइनरी संख्या के अंको का मान
1	B	4	
		→ 4	$16^0 = 1$ $1 \times 4 = 4$
		→ 11	$16^1 = 16$ $16 \times 11 = 176$
		→ 1	$16^2 = 256$ $256 \times 1 = 256$
			योग <u>436</u>

$$\therefore (1B4)_{16} = (436)_{10}$$

इसी प्रकार हेक्साडेसीमल भिन्नांक का परिवर्तन भी उसी प्रकार कर सकते हैं जिस प्रकार बाइनरी भिन्नांक का दशमलव में परिवर्तन करते हैं। परन्तु इसमें 2 की जगह 16 का प्रयोग करते हैं।

(ii) हेक्सा-डेसीमल (पूर्णांक तथा भिन्नांक) का बाइनरी में परिवर्तन (Conversion of Hexa-decimal (Integer and Fraction) into Binary)

हेक्सा-डेसीमल का बाइनरी में परिवर्तन अत्यधिक सरल है। इसमें केवल प्रत्येक हेक्सा-डेसीमल के स्थान पर उसके बराबर चार बिटों के समूह लिख देते हैं।

उदाहरण के लिए हेक्सा-डेसीमल D7A का बाइनरी में परिवर्तन

$$(D7A)_{16} = (\underline{1101} \underline{0111} \underline{1010})_2$$

या $(D7A)_{16} = (110101111010)_2$

इसी प्रकार, हेक्सा-डेसीमल (भिन्नांक) को भी बाइनरी में बदला जा सकता है।

(iii) हेक्सा-डेसीमल संख्या का ऑक्टल संख्या में परिवर्तन (Hexa-decimal to Octal)

हेक्सा-डेसीमल संख्या को ऑक्टल संख्या में बदलने के लिए निम्नलिखित स्टेप्स का प्रयोग करते हैं।

स्टेप 1 दी हुई हेक्सा-डेसीमल संख्या के प्रत्येक अंक को बाइनरी संख्या में बदलो।

स्टेप 2 प्राप्त बाइनरी संख्या को ऑक्टल संख्या में बदलो।

उदाहरण के लिए $(AC2D)_{16}$ को ऑक्टल संख्या में बदलो

A	C	2	D		
1010	1100	0010	1101		
$(AC2D)_{16} = (1010110000101101)_2$					
001	010	110	000	101	101

1 2 6 0 5 5

$$\therefore (AC2D)_{16} = (126055)_8$$

नोट हेक्सा-डेसीमल को ऑक्टल में बदलने के लिए एक और विधि का प्रयोग किया जा सकता है। जिसमें हेक्सा-डेसीमल को पहले दशमलव में फिर दशमलव को ऑक्टल में परिवर्तित किया जाता है।

कम्प्यूटर कोड्स (Computer Codes)

कम्प्यूटर प्रत्येक प्रकार के कैरेक्टर जैसे कि अल्फाबेट संख्या या कोई चिन्ह स्टोर कर सकता है। इन सभी कैरेक्टरों के निरूपण (Representation) के लिए बाइनरी संख्या पद्धति पर आधारित एक विशेष प्रकार के कोड की आवश्यकता होती है, जिसे कम्प्यूटर कोड कहा जाता है।

कम्प्यूटर कोड्स विभिन्न प्रकार के होते हैं, जिनमें से कुछ निम्नलिखित हैं

1. बाइनरी कोडेड डेसीमल (Binary Coded Decimal - BCD)

BCD कोड में प्रत्येक अंक को प्रस्तुत करने के लिए चार बिट्स के समूह का प्रयोग करते हैं। इसका प्रयोग 0 से 9 तक की संख्या को चार बिट्स के बाइनरी संख्या में निरूपित करने के लिए करते हैं।

BCD सिस्टम में किसी संख्या के आकार की कोई सीमा नहीं होती है अर्थात् संख्या चाहे कितनी ही बड़ी क्यों न हो, प्रत्येक अंक को चार बिट्स के समूह में कम्प्यूटर में प्रदर्शित करते हैं।

जैसे कि 1 4 3 9 2 4

1 4 3 9 2 4

0001 0100 0011 1001 0010 0100

$$\therefore (143924)_{10} = (0001\ 0100\ 0011\ 1001\ 0010\ 0100)_{BCD}$$

2. अमेरिकन स्टैंडर्ड कोड फॉर इन्फॉर्मेशन इण्टरचेन्ज

(ASCII) (American Standard Code for Information Interchange)

स्टैंडर्ड कैरेक्टर कोड का प्रयोग किसी प्रोग्राम द्वारा डेटा को स्टोर करने तथा उसका प्रयोग करने के लिए किया जाता है।

ASCII कोड दो प्रकार के होते हैं।


(i) **ASCII-7** यह एक 7-बिट स्टैंडर्ड कोड है। जिसके कारण कुल $2^7 = 128$ कैरेक्टर को निरूपित किया जा सकता है।


(ii) **ASCII-8** यह एक 8-बिट स्टैंडर्ड कोड है। इसमें $2^8 = 256$ प्रकार के कैरेक्टर को निरूपित किया जा सकता है। यह ASCII-7 का बदला हुआ प्रारूप है।

3. एक्सटेंडेड बाइनरी कोडेड डेसीमल इण्टरचेंज कोड (EBCDIC-Extended Binary Coded Decimal Interchange Code)

EBCDIC में, कैरेक्टर 8 बिट्स के समूह से निरूपित होते हैं। इसका प्रयोग किसी भी प्रकार के कम्प्यूटर में सूचनाओं को स्टोर करने के लिए किया जाता है। इसमें $2^8 = 256$ प्रकार के कैरेक्टर को निरूपित किया जा सकता है।

इन्हें भी जानें

 BCD सिस्टम IBM कॉर्पोरेशन द्वारा विकसित किया गया था।

 UNICODE में डेटा में किसी संकेत (Symbol) को प्रस्तुत करने के लिए 16 बिट्स का प्रयोग होता है। ये अंग्रेजी के अक्षरों के अलावा किसी भी प्रकार का वैज्ञानिक संकेत, अनेक प्रकार की भाषा जैसे कि चाइनीज (Chinese), जैपेनीज (Japanese) आदि को भी निरूपित करता है।

साइन बिट (Sign bit) का प्रयोग किसी संख्या के धनात्मक (Positive) या ऋणात्मक (Negative) होने का संकेत है। यह बिट Most Significant Bit (MSB) बिट कहलाती है तथा संख्या के बाइनरी निरूपण में सबसे पहली बिट होती है। यदि बिट 0 है तो संख्या धनात्मक है। यदि बिट 1 है तो संख्या ऋणात्मक है।

36

6

कम्प्यूटर सॉफ्टवेयर (Computer Software)

एक कम्प्यूटर सिस्टम अनेक इकाइयों का एक समूह होता है, जो एक या अनेक लक्ष्यों की प्राप्ति हेतु बनाया जाता है। उदाहरणार्थ-प्रयोगशाला भी एक सिस्टम है, जिसका लक्ष्य विविध प्रकार के शोध करना है तथा जिसकी अनेक इकाइयाँ; वैज्ञानिक शोधार्थी और वैज्ञानिक उपकरण इत्यादि हैं। इसी प्रकार कम्प्यूटर भी एक सिस्टम है, जिसका लक्ष्य विविध प्रकार के कार्य करना है तथा जिसकी इकाइयाँ हार्डवेयर तथा सॉफ्टवेयर हैं।

सॉफ्टवेयर (Software)

सॉफ्टवेयर, प्रोग्रामिंग भाषा में लिखे गए निर्देशों अर्थात् प्रोग्रामों की वह शृंखला है, जो कम्प्यूटर सिस्टम के कार्यों को नियन्त्रित करता है तथा कम्प्यूटर के विभिन्न हार्डवेयरों के बीच समन्वय स्थापित करता है, ताकि किसी विशेष कार्य को पूरा किया जा सके। इसका प्राथमिक उद्देश्य डेटा को सूचना में परिवर्तित करना है। सॉफ्टवेयर के निर्देशों के अनुसार ही हार्डवेयर कार्य करता है। इसे प्रोग्रामों का समूह भी कहते हैं।

दूसरे शब्दों में, “कम्प्यूटरों में सैकड़ों की संख्या में प्रोग्राम होते हैं, जो अलग-अलग कार्यों के लिए लिखे या बनाए जाते हैं। इन सभी प्रोग्रामों के समूह को सम्मिलित रूप से ‘सॉफ्टवेयर’ कहा जाता है।”

सॉफ्टवेयर के प्रकार (Types of Software)

सॉफ्टवेयर को उसके कार्यों तथा संरचना के आधार पर दो प्रमुख भागों में विभाजित किया गया है

1. सिस्टम सॉफ्टवेयर

2. एप्लीकेशन सॉफ्टवेयर



1. सिस्टम सॉफ्टवेयर (System Software)

जो प्रोग्राम कम्प्यूटर को चलाने, उसको नियन्त्रित करने, उसके विभिन्न भागों की देखभाल करने तथा उसकी सभी क्षमताओं का अच्छे से उपयोग करने के लिए लिखे जाते हैं, उनको सम्मिलित रूप से 'सिस्टम सॉफ्टवेयर' कहा जाता है। सामान्यतः सिस्टम सॉफ्टवेयर कम्प्यूटर के निर्माता द्वारा ही उपलब्ध कराया जाता है। वैसे यह बाद में बाजार से भी खरीदा जा सकता है। कम्प्यूटर से हमारा सम्पर्क या संवाद सिस्टम सॉफ्टवेयर के माध्यम से ही हो पाता है। दूसरे शब्दों में कम्प्यूटर हमेशा सिस्टम सॉफ्टवेयर के नियन्त्रण में ही रहता है, जिसकी वजह से हम सीधे कम्प्यूटर से अपना सम्पर्क नहीं बना सकते। वास्तव में सिस्टम सॉफ्टवेयर के बिना कम्प्यूटर से सीधा सम्पर्क नामुमकिन है, इसलिए सिस्टम सॉफ्टवेयर उपयोगकर्ता की सुविधा के लिए ही बनाया जाता है। सिस्टम सॉफ्टवेयर से हमें बहुत सुविधा हो जाती है, क्योंकि वह कम्प्यूटर

को अपने नियन्त्रण में लेकर हमारे द्वारा बताए गए कार्यों को कराने तथा प्रोग्रामों का सही-सही पालन करने के दायित्व अपने ऊपर ले लेता है। सिस्टम सॉफ्टवेयर में वे प्रोग्राम शामिल होते हैं, जो कम्प्यूटर सिस्टम को नियन्त्रित (Control) करते हैं और उसके विभिन्न भागों के बीच उचित तालमेल बनाकर कार्य करते हैं।

कार्यों के आधार पर सिस्टम सॉफ्टवेयर को दो भागों में बाँटा गया है—सिस्टम मैनेजमेन्ट प्रोग्राम और डवलपिंग सॉफ्टवेयर

(i) सिस्टम मैनेजमेन्ट प्रोग्राम (System Management Program)

ये वे प्रोग्राम होते हैं, जो सिस्टम का प्रबन्धन (Management) करने के काम आते हैं। इन प्रोग्राम्स का प्रमुख कार्य इनपुट आउटपुट तथा मैमोरी युक्तियों और प्रोसेसर के विभिन्न कार्यों का प्रबन्धन करना है। ऑपरेटिंग सिस्टम, डिवाइस ड्राइवर्स तथा सिस्टम यूटिलिटीज, सिस्टम मैनेजमेन्ट प्रोग्राम्स के प्रमुख उदाहरण हैं।

(a) ऑपरेटिंग सिस्टम (Operating System)

इसमें वे प्रोग्राम शामिल होते हैं जो कम्प्यूटर के विभिन्न अवयवों के कार्यों को नियन्त्रित करते हैं, उनमें समन्वय स्थापित करते हैं तथा उन्हें प्रबन्धित (Manage) करते हैं। इसका प्रमुख कार्य उपयोगकर्ता (User) तथा हार्डवेयर के मध्य एक समन्वय स्थापित करना है।

ऑपरेटिंग सिस्टम कुछ विशेष प्रोग्रामों का ऐसा व्यवस्थित समूह है, जो किसी कम्प्यूटर के सम्पूर्ण क्रियाकलापों को नियन्त्रित रखता है। यह कम्प्यूटर के साधनों के उपयोग पर नजर रखने और उन्हें व्यवस्थित करने में हमारी सहायता करता है। ऑपरेटिंग सिस्टम आवश्यक होने पर अन्य प्रोग्रामों को चालू करता है, विशेष सेवाएँ देने वाले प्रोग्रामों का मशीनी भाषा में अनुवाद करता है और उपयोगकर्ताओं की इच्छा के अनुसार

आउटपुट निकालने के लिए डेटा का प्रबन्धन करता है। वास्तव में यह प्रोग्रामों को कार्य करने के लिए एक आधार उपलब्ध कराता है। उदाहरण एम एस डॉस, विण्डोज XP/2000/98, यूनिक्स, लाइनेक्स इत्यादि ऑपरेटिंग सिस्टम के कुछ उदाहरण हैं।

ऑपरेटिंग सिस्टम के कार्य (Functions of Operating System)

1. कम्प्यूटर तथा उसके उपयोगकर्ता के बीच संवाद (Communication) स्थापित करना।
2. कम्प्यूटर के सभी उपकरणों को नियन्त्रण में रखना तथा उनसे काम लेना।