Unit - 2 : Data Representation

1. Definition of Information

Information वह data होता है जो processed और organized रूप में किसी व्यक्ति या system के लिए useful और meaningful बन जाता है। Raw data अपने आप में meaningful नहीं होता, लेकिन जब उसे प्रोसेस करके किसी कार्य के लिए उपयोगी बना दिया जाता है, तब वह information कहलाता है।

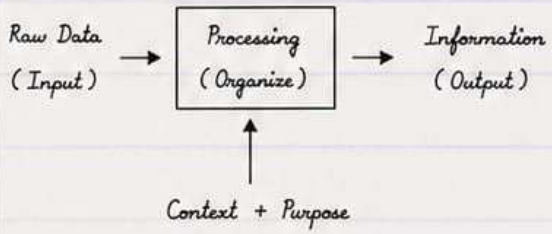
Key Points

- Information, raw data से अधिक useful और meaningful होता है।
- यह decision making में help करता है।
- सभी Information context और purpose पर depend करती है।
- Information को आसानी से समझा और share किया जा सकता है।
- यह knowledge प्राप्त करने और समस्या का समाधान निकालने के लिए आवश्यक है।

Data vs Information

Data	Information
Raw facts and figures	Processed और organized data
Meaningful नहीं होता	Meaningful और useful होता है
Input के रूप में होता है	Output के रूप में होता है
उदाहरण: 25, 35, 45	उदाहरण: Average age is 35 years
Decision making में direct help नहीं करता	Decision making में help करता है

Data to Information Process



Characteristics of Good Information

- Accurate – सही और त्रुटि रहित हो।
- Relevant – उद्देश्य के अनुसार प्रासंगिक हो।
- Timely – समय पर उपलब्ध हो।
- Complete – सभी आवश्यक जानकारी शामिल हो।
- Clear – समझने में आसान हो।
- Reliable – भरोसेमंद स्रोत से प्राप्त हो।

Example

- Data: 72, 68, 90, 85, 76
  यह raw marks है।
- Information: Student का average marks 78.2 है
  यह processed और meaningful information है, जिससे performance का पता चलता है।

☆ **Summary** : Information, raw data का processed और organized रूप होता है जो किसी कार्य के लिए useful और meaningful होता है। यह decision making, knowledge प्राप्त करने और communication में महत्वपूर्ण भूमिका निभाता है।

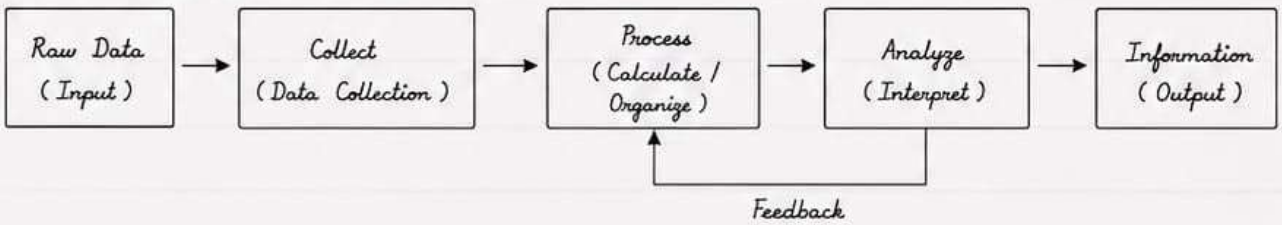
Unit - 2 : Data Representation

3. Difference Between Data and Information

Data और Information आपस में related होते हैं लेकिन इनका nature और use different होता है। Data raw facts होते हैं जो processing से पहले के form में होते हैं, जबकि Information processed और organized data होते हैं जिनका meaning होता है और decision making में help करते हैं।

Basis	Data	Information
1. अर्थ (Meaning)	Data का कोई specific meaning नहीं होता।	Information meaningful होती है और purpose को clearly बताती है।
2. Nature	Data raw, unprocessed और unorganized होता है।	Information processed, organized और structured होता है।
3. Form	Data numbers, words, symbols या figures के form में होता है।	Information tables, reports, charts या documents के form में होती है।
4. Processing	Data को processing की आवश्यकता होती है।	Information processing के बाद प्राप्त होती है।
5. Use	Data का use direct decision making में नहीं होता।	Information decision making और problem solving के लिए use होती है।
6. Example	72, 68, 90, 85, 76 (marks)	Average marks = 78.2% (यह student की performance बताता है।
7. Dependence	Data independent facts होते हैं।	Information data पर depend करती है और उसके analysis से बनती है।
8. Volume	Data की मात्रा बहुत अधिक हो सकती है।	Information concise और useful होती है, क्योंकि यह filtered और summarized होती है।

Data to Information Process



☆ **Summary** : Data raw facts होते हैं जिनका कोई direct meaning नहीं होता, जबकि Information processed और organized data होते हैं जिनका meaning होता है और जो decision making में उपयोगी होते हैं। सही information प्राप्त करने के लिए data को collect, process और analyze करना आवश्यक होता है।

Unit - 2 : Data Representation

4. Importance of Binary Number System

Binary Number System (Base-2) सबसे महत्वपूर्ण number system है क्योंकि एक Computer के अंदर सभी प्रकार की processing, storage और communication binary form (0 और 1) में ही होती है। इसके बिना digital computer का कार्य संभव नहीं है।

Key Points

① Computer की hardware circuits दो states (ON/OFF) में काम करती हैं, जो 0 और 1 को represent करते हैं।

② Binary system में data को store, process और transfer करना आसान होता है।

③ यह system reliable होता है क्योंकि इसमें केवल दो digits होते हैं, जिससे errors कम होते हैं।

④ Binary form में data का representation कम cost और कम power में किया जा सकता है।

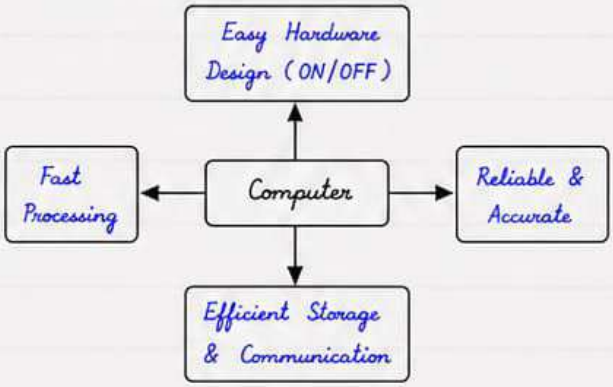
⑤ सभी प्रकार के digital devices (Computer, Mobile, Calculator, etc.) binary system पर ही आधारित होते हैं।

⑥ Programming, logical operations और digital communication binary system के द्वारा ही संभव होते हैं।

Decimal vs Binary Number System

Basis	Decimal (Base-10)	Binary (Base-2)
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1
Base	10	2
Place Value	$10^0, 10^1, 10^2, \ldots$	$2^0, 2^1, 2^2, \ldots$
Use	Humans के लिए आसान	Computers के लिए उपयुक्त
Hardware	Multiple states	Two states (ON/OFF)
Error	अधिक संभावना	कम संभावना

Why Binary is Best for Computers?



Examples

- Decimal Number : $25 = 2 \times 10 + 5 \times 1 = 25$
- Binary Number : $11001_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 8 + 0 + 0 + 1 = 25$

☆ **Summary :** Binary Number System digital computers की foundation है। यह simple, reliable, cost effective और efficient है। सभी digital data (text, numbers, images, audio, video) को अंततः binary form (0 और 1) में ही convert करके process किया जाता है।

Unit - 2 : Data Representation

5. Various Number Systems

किसी भी quantity या value को represent करने के लिए अलग-अलग number systems का use किया जाता है। Computer के अंदर सभी data binary form (0 और 1) में process होता है, लेकिन humans के लिए decimal system सबसे आसान है। इसलिए हमें एक system से दूसरे system में conversion का ज्ञान होना चाहिए।

Key Points

- विभिन्न number systems का use अलग-अलग purposes के लिए होता है।
- हर number system की अपनी base (radix) होती है।
- Computer internally binary system (base-2) का use करता है।
- एक number system से दूसरे system में conversion आवश्यक होता है।
- Number systems को generally चार भागों में classify किया जाता है।

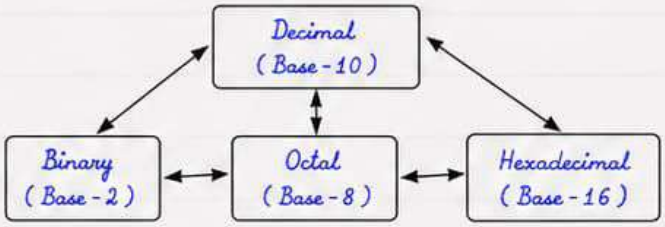
Various Number Systems

Number System	Base (Radix)	Digits	Use / विशेषताएँ
1. Binary System	2	0, 1	Computer का internal system, electronic circuits के लिए best.
2. Octal System	8	0 - 7	Short hand representation के लिए उपयोगी, 3 bits को represent करता है।
3. Decimal System	10	0 - 9	Human friendly system, सामान्य गणना के लिए सबसे सरल।
4. Hexadecimal System	16	0 - 9, A - F	Compact representation के लिए, 4 bits को represent करता है।

Relationship Between Systems

- 1 Octal digit = 3 Binary digits
- 1 Hexadecimal digit = 4 Binary digits
- 1 Decimal digit = 3.322 Binary digits (approx.)

Conversion Between Number Systems



Examples

- $(25)_{10} = (11001)_2 = (31)_8 = (19)_{16}$
- $(75)_{10} = (1001011)_2 = (113)_8 = (4B)_{16}$

☆ **Summary :** विभिन्न number systems का ज्ञान computer science का मूल आधार है। Binary system computer का internal language है, जबकि decimal system humans के लिए आसान है। Conversion के बिना data को समझना और process करना संभव नहीं है।

Unit - 2 : Data Representation

6. Conversion from Decimal to Binary

Decimal Number System (Base-10) में numbers 0 से 9 तक के digits से represent होते हैं, जबकि Binary Number System (Base-2) में केवल 0 और 1 digits का use होता है। किसी भी decimal number को binary number में convert करना बहुत जरूरी होता है क्योंकि computer internally binary form (0 और 1) में ही data को process करता है।

Conversion Method (Division by 2)

Decimal number को binary में convert करने के लिए बार-बार 2 से division करते हैं और हर step का remainder (0 या 1) लिखते हैं। अंत में जो remainder bottom से top की तरफ पढ़ने पर मिलता है, वही binary number होता है।

Steps

① Decimal number को 2 से divide करें।

② Quotient और Remainder (0 या 1) लिखते जाएँ।

③ Quotient को फिर से 2 से divide करें।

④ यह process तब तक दोहराएँ जब तक Quotient 0 न हो जाए।

⑤ Remainders को bottom से top की तरफ पढ़ें, वही Binary number है।

	Dividend		Quotient	Remainder
2	25	→	12	1
2	12	→	6	0
2	6	→	3	0
2	3	→	1	1
2	1	→	0	1

(Stop here)

Read Bottom to Top

Examples

(i) Convert 13_{10} to Binary

2	13	→ 6	Remainder →	1
2	6	→ 3	Remainder →	0
2	3	→ 1	Remainder →	1
2	1	→ 0	Remainder →	1

(Stop)

$$\boxed{13_{10} = 1101_2}$$

(Read bottom to top)

(ii) Convert 25_{10} to Binary

2	25	→ 12	Remainder →	1
2	12	→ 6	Remainder →	0
2	6	→ 3	Remainder →	0
2	3	→ 1	Remainder →	1
2	1	→ 0	Remainder →	1

(Stop)

$$\boxed{25_{10} = 11001_2}$$

(Read bottom to top)

☆ **Summary** : Decimal number को binary में convert करने के लिए 2 से बार-बार division किया जाता है और remainders (0 या 1) को bottom से top की तरफ पढ़ा जाता है। यह method simple, fast और computers के लिए very useful है।

Unit - 2 : Data Representation

7. Conversion from Binary to Decimal

Binary Number System (Base - 2) में केवल 0 और 1 digits होते हैं। किसी binary number को decimal number में convert करने के लिए हम हर bit के position value (place value) के साथ multiply करते हैं और सभी products को add करते हैं। जो final sum प्राप्त होता है, वही decimal number होता है।

Place Value in Binary Number

Binary number को right से left की तरफ powers of 2 दिए जाते हैं।

Position	7	6	5	4	3	2	1	0
Power of 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Value	128	64	32	16	8	4	2	1

Steps

① Binary number के हर bit के ऊपर उसके position की power of 2 लिखें। (right से left)

② प्रत्येक bit (0 या 1) को उसके corresponding power of 2 के साथ multiply करें।

③ जिन bits की value 0 है, उनके products 0 होंगे (उन्हें ignore कर सकते हैं)।

④ सभी non-zero products को add करें।

⑤ जो final sum प्राप्त होगा, वही decimal number है।

Examples

(i) $(101101)_2$ को decimal में convert करें।

Binary →	1	0	1	1	0	1
Position →	2^5	2^4	2^3	2^2	2^1	2^0
Value →	32	16	8	4	2	1
Multiply →	1×32	0×16	1×8	1×4	0×2	1×1
Product →	32	0	8	4	0	1

Add → $32 + 0 + 8 + 4 + 0 + 1 = 45$

$$(101101)_2 = (45)_{10}$$

(ii) $(1100101)_2$ को decimal में convert करें।

Binary →	1	1	0	0	1	0	1
Position →	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value →	64	32	16	8	4	2	1
Multiply →	1×64	1×32	0×16	0×8	1×4	0×2	1×1
Product →	64	32	0	0	4	0	1

Add → $64 + 32 + 0 + 0 + 4 + 0 + 1 = 101$

$$(1100101)_2 = (101)_{10}$$

☆ **Summary** : Binary number को decimal में convert करने के लिए प्रत्येक bit को उसकी power of 2 (position value) के साथ multiply किया जाता है। जिन bits की value 1 होती है, केवल उनके products को add किया जाता है। यह method computer के internal data representation और calculation के लिए बहुत useful है।

Unit - 2 : Data Representation

8. Binary Number into Hexadecimal Number

Hexadecimal Number System (Base-16) में 16 symbols (0-9 और A-F) होते हैं। Binary number को hexadecimal number में convert करने के लिए binary digits को 4-4 bits के groups (nibble) में divide किया जाता हैं, क्योंकि 4 bits (0000 से 1111) कुल 16 combinations बनाते हैं, जो hexadecimal के 16 symbols (0-9, A-F) के बराबर होते हैं।

Conversion Method (Grouping by 4 bits)

① Binary number को right से left की ओर 4-4 bits के groups में divide करें।
② यदि left side में 4 bits से कम bits हों, तो उसे left में 0's add करके 4 bits पूरा करें।
③ हर 4 bits group को उसके equivalent hexadecimal digit से replace करें।
④ Left से right की ओर लिखने पर final hexadecimal number प्राप्त होता हैं।

4 Bits (Nibble) to Hexadecimal Mapping

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

Examples

(i) $(11010110)_2$ को Hexadecimal में convert करें।

Step 1: 4-4 bits के groups बनाएं (right से left)

1101 0110

Step 2: हर group को hexadecimal digit से replace करें।

1101 → D, 0110 → 6

Step 3: Left से right लिखें → $(D6)_{16}$

(ii) $(101111001011)_2$ को Hexadecimal में convert करें।

Step 1: 4-4 bits के groups बनाएं

1011 1100 1011

Step 2: हर group को hexadecimal digit से replace करें।

1011 → B, 1100 → C, 1011 → B

Step 3: Left से right लिखें → $(BCB)_{16}$

☆ **Summary** :
- Binary number को hexadecimal में convert करने के लिए 4-4 bits के groups (nibble) बनाएं जाते हैं।
- 4 bits कुल 16 combinations (0000 से 1111) बनाते हैं, जो 16 hexadecimal symbols (0-9 और A-F) के बराबर होते हैं।
- यह method fast, simple और memory efficient होतो हैं, इसलिए computer systems में hexadecimal का use अधिक किया जाता हैं।

Unit - 2 : Data Representation

9. Hexadecimal Number into Binary Number System

Hexadecimal Number System (Base - 16) में 16 symbols (0 - 9 और A - F) होते हैं। Hexadecimal number को binary number में convert करने के लिए प्रत्येक hexadecimal digit को उसके equivalent 4 - bit binary group से replace किया जाता है, क्योंकि 1 hexadecimal digit = 4 binary bits ($2^4 = 16$) को represent करता है।

Conversion Method (Replace each hex digit by 4 bits)

① हर hexadecimal digit को उसके equivalent 4 - bit binary group से replace करें।

② Left से right की ओर लिखते जाएँ और सभी 4 - bit groups को साथ लिखें।

③ कोई और calculation की आवश्यकता नहीं होती, केवल direct replacement होता है।

④ यदि आवश्यक हो तो final binary number को 4 - 4 bits के groups में लिखें।

Hexadecimal to Binary Mapping

Hexadecimal	0	1	2	3	4	5	6	7	8	9	B	C	D	E	F
Binary (4-bit)	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1110	1111

Examples

(i) $(3A7)_{16}$ को Binary में convert करें।

Step 1 : हर hexadecimal digit को 4 - bit binary group से replace करें।

Hexadecimal : 3 A 7

↓ ↓ ↓

Binary (4-bit) : 0011 1010 0111

Step 2 : सभी groups को साथ लिखें।

$\therefore$ $(3A7)_{16} = (0011\ 1010\ 0111)_2$

(Group by 4 bits)

(ii) $(2F9B)_{16}$ को Binary में convert करें।

Step 1 : हर hexadecimal digit को 4 - bit binary group से replace करें।

Hexadecimal : 2 F 9 B

↓ ↓ ↓ ↓

Binary (4-bit) : 0010 1111 1001 1011

Step 2 : सभी groups को साथ लिखें।

$\therefore$ $(2F9B)_{16} = (0010\ 1111\ 1001\ 1011)_2$

(Group by 4 bits)

Note :

- हर hexadecimal digit हमेशा 4 bits में represent होता है।
- इस conversion में कोई addition, multiplication या calculation नहीं होती।
- यह method simple, fast और error - free होता है।

☆ **Summary :**

- Hexadecimal number को binary में convert करने के लिए हर digit को 4 - bit binary group से replace किया जाता है।
- कुल 16 hexadecimal digits (0 - 9, A - F) होते हैं और प्रत्येक digit 4 bits (0000 से 1111) को represent करता है।
- यह conversion method computer systems में memory address, data storage और communication के लिए बहुत उपयोगी है।

Unit - 2 : Data Representation

10. Memory Addressing and its Importance

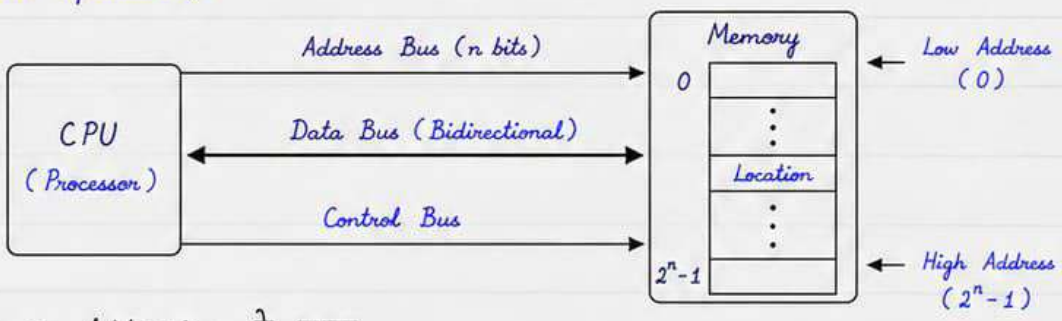
1. Memory Addressing क्या है?

Memory Addressing वह प्रक्रिया है जिसके द्वारा Computer की Main Memory के हर Location (Byte) को एक unique Address दिया जाता है। CPU इन Addresses का use करके Memory से Data read या write करता है।

यदि Address सही होगा, तभी सही Data प्राप्त होगा।

2. Memory Addressing कैसे काम करता है?

CPU किसी Data को access करने के लिए, उस Location का Address generate करता है और उसे Address Bus के द्वारा Memory में भेजता है। Memory उस Address को decode करके संबंधित Data को Data Bus के द्वारा वापस CPU को भेजती है (Read Operation) या CPU से Data प्राप्त करके उस Address पर store कर देती है (Write Operation).



3. Memory Addressing के प्रकार

① Byte Addressing - हर Byte को अलग Address दिया जाता है।

② Word Addressing - हर Word (2 या 4 Byte) को एक Address दिया जाता है।

③ Block Addressing - Memory को Blocks में divide करके हर Block को Address दिया जाता है।

④ Segment Addressing - Memory को Segments में divide करके Addressing की जाती है।

4. Memory Addressing का महत्व (Importance)

- सही Data तक पहुँचने के लिए Addressing अनिवार्य होता है।
- यह Memory Management और Data Organization में मदद करता है।
- Program Execution, Data Retrieval और Storage के लिए महत्वपूर्ण है।
- Large Memory Space को efficiently access करने में सहायक होता है।

☆ **Summary :** Memory Addressing Computer System का एक मूल आधार है। इसके द्वारा CPU Memory के हर Location को uniquely identify करता है और सही Data को read या write करता है। यही Process Computer की Speed, Accuracy और Efficiency के लिए बहुत महत्वपूर्ण होता है।

Unit - 2 : Data Representation

11. ASCII Coding System

1. ASCII क्या है ?

ASCII का पूरा नाम American Standard Code for Information Interchange होता है। यह एक Character Coding System है जिसका use Computer में Textual Data को represent करने के लिए किया जाता है। ASCII Code 7-bit का होता है, इसलिए इससे $2^7 = 128$ अलग-अलग Characters को represent किया जा सकता है।

2. ASCII Code की विशेषताएँ

① ASCII एक 7-bit Code होता है, इसलिए कुल 128 (2^7) Codes होते हैं (0 से 127 तक)।

② इसमें 0 - 31 और 127 Codes Control Characters के लिए होते हैं।

③ 32 - 126 Codes Printable Characters (Letters, Digits, Symbols) के लिए होते हैं।

④ यह Code Communication और Data Interchange के लिए Standard है।

⑤ ASCII का Extended रूप 8-bit में भी उपयोग होता है (Extended ASCII) जिसमें कुल 256 (2^8) Codes होते हैं।

3. ASCII Code Table (मुख्य भाग)

Decimal	Character	Binary (7-bit)
0	NUL	0000000
1	SOH	0000001
2	STX	0000010
⋮	...	...
31	US	0011111

↓ Control Characters (0 - 31)

Decimal	Character	Binary (7-bit)
32	(space)	0100000
33	!	0100001
34	"	0100010
⋮	⋮	...
126	~	1111110

↓ Printable Characters (32 - 126)

Decimal	Character	Binary (7-bit)
127	DEL	1111111

↓ Control Character (127)

4. उदाहरण (Examples)

Character	A	B	C	a	b	c	0	1	2	@	#	$	(space)
Decimal	65	66	67	97	98	99	48	49	50	64	35	36	32
Binary (7-bit)	1000001	1000010	1000011	1100001	1100010	1100011	0110000	0110001	0110010	1000000	0100011	0100100	0100000

5. ASCII का उपयोग (Importance)

- यह Computer में Text Data को Store, Process और Communicate करने के लिए आवश्यक है।
- सभी Programming Languages ASCII Codes का उपयोग करती हैं।
- यह विभिन्न Devices और Systems के बीच Data Communication को Standard बनाता है।
- यह Character Representation को सरल, Reliable और Portable बनाता है।

☆ **Summary** : ASCII एक 7-bit Character Coding System है जो कुल 128 Characters को represent करता है। यह Control और Printable Characters दोनों को define करता है और Computer में Textual Data Representation और Communication के लिए बहुत महत्वपूर्ण है।

Unit - 2 : Data Representation

12. EBCDIC Coding System

1. EBCDIC क्या है ?

EBCDIC का पूरा नाम "Extended Binary Coded Decimal Interchange Code" होता है। यह एक Character Coding System है जिसे मुख्य रूप से IBM Computers के लिए develop किया गया था। यह ASCII का ही एक Extended रूप है, जिसमें कुल 8-bit (256) Codes होते हैं।

2. EBCDIC Code की विशेषताएँ

① EBCDIC एक 8-bit Code (1 Byte) होता है, जिसमें $2^8 = 256$ Codes होते हैं (00 से FF तक)।

② यह Code मुख्य रूप से IBM Systems (Mainframe Computers) में use होता है।

③ इसमें 0-9, A-Z, a-z, Symbols और Control Characters के Codes शामिल होते हैं।

④ यह Code Data Communication और Interchange के लिए Standard प्रदान करता है।

⑤ ASCII की तुलना में इसमें Alphabetic Characters के Codes अलग होते हैं।

3. EBCDIC Code Table (मुख्य भाग)

Character	EBCDIC (Hex)
0	F0
1	F1
2	F2
...	...
9	F9

Digits (0-9)

Character	EBCDIC (Hex)
A	C1
B	C2
C	C3
...	...
Z	E9

Uppercase Letters (A-Z)

Character	EBCDIC (Hex)
a	81
b	82
c	83
...	...
z	A9

Lowercase Letters (a-z)

Note : अलग-अलग EBCDIC Code Pages (जैसे - 037, 1047, 1140) में कुछ Codes बदल सकते हैं, लेकिन मूल संरचना समान रहती है।

4. उदाहरण (Examples)

Character	A	B	C	a	b	c	0	1	2	@	(space)
EBCDIC (Hex)	C1	C2	C3	81	82	83	F0	F1	F2	7C	40
EBCDIC (Binary)	1100 0001	1100 0010	1100 0011	1000 0001	1000 0010	1000 0011	1111 0000	1111 0001	1111 0010	0111 1100	0100 0000

5. EBCDIC का महत्व (Importance)

- यह IBM Systems में Data Processing और Storage के लिए Standard है।
- यह Data Interchange और Communication को आसान बनाता है।
- Large Scale Systems (Mainframe) में Data को Organize और Manage करने के लिए उपयोगी है।
- यह Character Representation को Efficient और Reliable बनाता है।

☆ **Summary** : EBCDIC एक 8-bit Character Coding System है जिसमें कुल 256 Codes होते हैं। यह मुख्य रूप से IBM Computers में use होता है और Data Representation, Storage एवं Communication के लिए बहुत महत्वपूर्ण होता है।