অধ্যায়-২ সংখ্যা পদ্ধতি ও অ্যারিথমেটিক কোডসমূহ

SOS অতি সংক্ষিপ্ত প্রশ্নোত্তর ঃ

****১। সংখ্যা পদ্ধতি (Number System) কাকে বলে?**

বাকাশিবো- ২০১৯'পরি

উত্তর ঃ কোনো সংখ্যাকে উপস্থাপন বা প্রকাশ করার জন্য যে সকল পদ্ধতি ব্যবহার করা হয়, তাকে সংখ্যা পদ্ধতি বা Number System বলে।

***** ২। সংখ্যা পদ্ধতির বেস বা ভিত্তি কী?**

বাকাশিবো- ২০১০, ১৫'পরি, ১৬, ১৬'পরি, ২৪'পরি

উত্তর ঃ কোনো Number System কে প্রকাশ করার জন্য যতগুলো Digit বা অঙ্ক ব্যবহার করা হয়, তাকে ঐ Number System এর বেস বা ভিত্তি বা Radix বলে।

যেমন ঃ Decimal Number System কে প্রকাশ করার জন্য 0 থেকে 9 এই 10 টি অঙ্ক বা ডিজিট ব্যবহার করা হয়। সুতরাং Decimal এর base 10.

**** ৩। LSB ও MSB বলতে কী বুঝ?** বাকাশিবো- ২০০৯, ১২

উত্তর ঃ LSB ঃ LSB এর পূর্ণরূপ Least Significant Bit. Binary Number System এর সর্বডানের Bit কে LSB বলে।

MSB ঃ MSB এর পূর্ণরূপ Most Significant Bit. Binary Number System এর সর্ববামের Bit কে MSB বলে।

যেমন ঃ MSB **1010** LSB

**** ৪। বিট ও কোডের মধ্যে মূল পার্থক্য কী?** বাকাশিবো- ২০০৯, ১৫'পরি

উত্তর ঃ বিট বলতে দুইটি বাইনারি Digit 0 এবং 1 কে বুঝায়। আর কোড হচ্ছে তথ্য সম্বলিত সংকেত।

***** ৫। Gray Code বলতে কী বুঝ?** বাকাশিবো- ২০১০'পরি, ১২, ১৬

উত্তর ঃ এটি এমন এক ধরনের কোড যাতে একটি সংখ্যা হতে পরবর্তী সংখ্যায় যেতে মাত্র একটি বিটের পরিবর্তন ঘটে। এটি বিশেষ ধরনের রিফ্লেকটেড কোড।

ব্যবহার ঃ এটি Analog to Digital Converter Circuit এবং Input Output Device এ ব্যবহৃত হয়।

**** ৬। Gray Code কেন ব্যবহার করা হয়?** বাকাশিবো- ২০০৮, ০৯, ১০

উত্তর ঃ বাইনারী সংখ্যাকে কোডে রূপান্তরের জন্য Gray Code ব্যবহৃত হয়।

**** ৭। $(59)_{10}$ কে Gray Code এ রূপান্তর কর।**

বাকাশিবো- ২০১২'পরি, ১৬, ২৫'পরি

উত্তর ঃ $(59)_{10} = \begin{array}{cccccc} 1 & 1 & 1 & 0 & 1 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 0 & 1 & 1 & 0 \end{array}$

$(59)_{10}$ এর বাইনারি $(111011)_2$

[MSB সরাসরি বসবে, পাশাপাশি দুই bit যোগ হবে, Carry যদি থাকে তবে তা বাদ যাবে]

**** ৮। $(110001)_2$ কে Gray Code এ রূপান্তর কর।**

বাকাশিবো-২০১২, ১৩

উত্তর ঃ $\begin{array}{cccccc} 1 & 1 & 0 & 0 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array}$

**** ৯। $(54)_{10}$ কে Excess-3 Code এ রূপান্তর কর।**

বাকাশিবো- ২০১৩'পরি, ১৫'পরি

উত্তর ঃ

$$\begin{array}{r} 5 \\ +3 \\ \hline 8 \end{array} \qquad \begin{array}{r} 4 \\ +3 \\ \hline 7 \end{array}$$

8 এর 4 Bit এর Binary = 1000

7 এর 4 Bit এর Binary = 0111

$\therefore (54)_{10} = (10000111)_{Ex\text{-}3}$

***** ১০। Hamming Code কাকে বলে?**

বাকাশিবো- ২০০৩, ০৬, ১৫, ১৮, ১৮'পরি

উত্তর ঃ 1950 সালে রিচার্ড হামিং এই কোড আবিষ্কার করেন। এটি ত্রুটি সনাক্ত ও সংশোধনের জন্য বহুল ব্যবহৃত একটি পদ্ধতি। Data transfer করার সময় যদি কোন Bit Position এ error থাকে তাহলে Hamming Code এর মাধ্যমে তা সনাক্ত করা এবং সংশোধন করা যায়।

***** ১১। ASCII Code কী?** বাকাশিবো- ২০১২, ১৩, ১৪'পরি, ২৪'পরি

উত্তর ঃ ১৯৬৫ সালে Robert William bener সাত বিটের ASCII Code উদ্ভাবন করেন। এর পূর্ণরূপ American Standard Code for Information Interchange. এই 7 Bit এর বাম দিকের তিনটি Bit কে জোন এবং ডান দিকের চারটি Bit কে সংখ্যাসূচক Bit ধরা হয়। তবে সর্ব বামে একটি Parity Bit যোগ করে 8 Bit এ পরিণত করা হয়। 7 Bit দ্বারা $2^7 = 128$ টি অক্ষর প্রকাশ করা যায়।

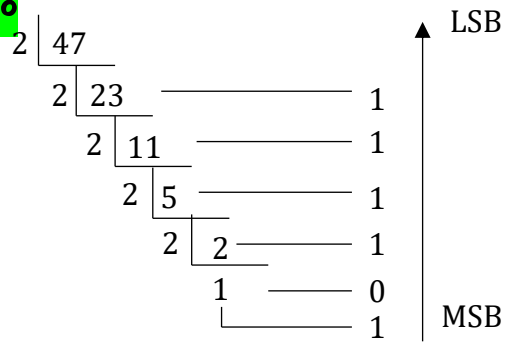
যেমন ঃ 0 = 48, A = 65, a = 97

*** ১২। $(47.85)_{10}$ কে বাইনারীতে রূপান্তর কর।

বাকাশিবো- ২০১৮'পরি, ১৫'পরি, ২৩'পরি, ২৫'পরি

Same as, $(13.15)_{10}$

উত্তর ঃ



$0.85 \times 2 = 1.7 \rightarrow 1$
$0.7 \times 2 = 1.4 \rightarrow 1$
$0.4 \times 2 = 0.8 \rightarrow 0$
$0.8 \times 2 = 1.6 \rightarrow 1$
$0.6 \times 2 = 1.2 \rightarrow 1$

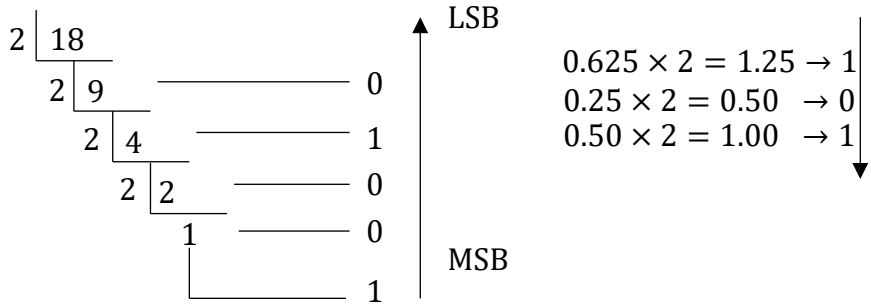
$$\therefore (47.85)_{10} = (101111.11011)_2$$

SOS সংক্ষিপ্ত প্রশ্নোত্তর ঃ

***** ১। $(18.625)_{10}$ এর সমতুল্য বাইনারী সংখ্যা নির্ণয় কর।**

বাকাশিবো- ২০১৩'পরি, ১৪'পরি, ১৫'পরি

উত্তর ঃ

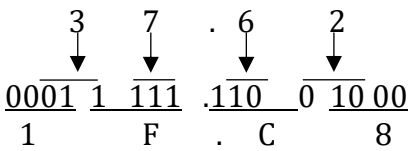


$$\therefore (18.625)_{10} = (10010.101)$$

**** ২। $(37.62)_8$ কে হেক্সের সংখ্যায় রূপান্তর কর।**

বাকাশিবো-২০১২, ১৩, ১৪'পরি

উত্তর ঃ



$$\therefore (37.62)_8 = (1F.C8)_{16}$$

***** ৩। 2's Complement পদ্ধতিতে $(9)_{10}$ থেকে $(11)_{10}$ বিয়োগ কর।** বাকাশিবো- ২০১০'পরি, ১৬

উত্তর ঃ $(9)_{10} = (1001)_2$

$(11)_{10} = (1011)_2$

মনে করি, A = 1001

B = 1011

$\therefore B' = 0\,1\,0\,0$
$\quad\quad\quad +1$
$B'' = 0\,1\,0\,1$ [1's Complement]

$A = 1\,0\,0\,1$
$B'' = 0\,1\,0\,1$
$0\,1\,1\,1\,0$ [2's Complement]
$= 0\,0\,0\,1$
$\quad\quad +1$
$-0\,0\,1\,0$

[Carry 0 আসায় যোগফলকে 2's Complement করতে হবে]

[1's Complement]

Ans : $-(0010)_2$

***** ৪। $(86)_{10}$ এবং $(95)_{10}$ কে BCD 8421 Coding পদ্ধতিতে যোগ কর।** বাকাশিবো- ২০১৫'পরি, ১৩'পরি

উত্তর ঃ

86 এর BCD = 1 0 0 0 0 1 1 0

95 এর BCD = 1 0 0 1 0 1 0 1

Binary Sum = 1 0 0 0 1 1 0 1 1

\+ 0 1 1 0 0 1 1 0

BCD Sum = 0 0 0 1 1 0 0 0 0 0 0 1

1 8 1

Note ঃ BCD যোগের ক্ষেত্রে Binary যোগফল 9 (1001) এর চেয়ে বেশি হলেই অতিরিক্ত 6 (0110) যোগ করতে হবে।

***** ৫। $(111011)_2$ কে Gray Code এ রূপান্তর কর।**

বাকাশিবো- ২০১০'পরি, ১৬, ১৬'পরি

উত্তর ঃ

1 1 1 0 1 1

1 0 0 1 1 0 Gray Code

***** ৬। $(68)_{10}$ কে Gray Code এ রূপান্তর কর।**

বাকাশিবো- ২০১৬, ১৮, ১৮'পরি

উত্তর ঃ $(68)_{10} = (1000100)_2$

1 0 0 0 1 0 0

1 1 0 0 1 1 0

SOS রচনামূলক প্রশ্নোত্তর ঃ

***** ১। Hamming Code এর সাহায্যে Error Correction ও Detection নির্ণয় পদ্ধতি উদাহরণসহ বর্ণনা কর।**

উত্তর ঃ Hamming Code ঃ 1950 সালে রিচার্ড হামিং এই কোড আবিষ্কার করেন। এটি ত্রুটি সনাক্ত ও সংশোধনের জন্য বহুল ব্যবহৃত একটি পদ্ধতি। Data Transfer করার সময় যদি কোনো Bit Position এ error থাকে তাহলে Hamming Code এর মাধ্যমে তা সনাক্ত করা এবং সংশোধন করা যায়।

এতে n সংখ্যক ডাটার মধ্যে k সংখ্যক Parity Bit যুক্ত করে (n + k) সংখ্যক Bit এর Data তৈরি করা হয়।

উদাহরণঃ মনে করি, 1011001 একটি 7 Bit এর Data.

যার ১ম, ২য়, ৪র্থ ও ৮ম পজিশনে Parity Bit বসবে।

Hamming Bit গুলি নির্ণয় করার জন্য যে সকল স্থানে 1 আছে তাদেরকে যতগুলো Parity Bit বসানো হয়েছে তত bit এর Binary তে রূপান্তর করে যোগ করতে হবে। অথবা X-OR Operation করতে হবে। (যোগ করলে Carry বাদ যাবে)

11	10	9	8	7	6	5	4	3	2	1
1	0	1	P_3	1	0	0	P_2	1	P_1	P_0

$$\begin{array}{rl} 11 = & 1\ 0\ 1\ 1 \\ 9 = & 1\ 0\ 0\ 1 \\ 7 = & 0\ 1\ 1\ 1 \\ \underline{3 =} & \underline{0\ 0\ 1\ 1} \\ & 0\ 1\ 1\ 0 \\ & P_3\ P_2\ P_1\ P_0 \end{array}$$

X-OR: বিজোড় সংখ্যক Input
1 হলে Output 1, অন্যথায়, Output 0.

এখন Parity Bit এর মান বসাতে হবে।

11	10	90	8	7	6	5	4	3	2	1
1	0	1	0	1	0	0	1	1	1	0

আবার যে সকল অবস্থানে 1 আছে তাদেরকে binary তে আগের মত করে রূপান্তর করতে হবে। এবং X- OR Operation করতে হবে।

11	= 1	0	1	1
9	= 1	0	0	1
7	= 0	1	1	1
4	= 0	1	0	0
3	= 0	0	1	1
2	= 0	0	1	0
	0	0	0	0

যেহেতু X-OR Operation এ প্রতিটি Parity Bit এর মান 0 এসেছে, সুতরাং কোনো Error নেই। যদি 0 বাদে 1 আসত তাহলে যে যে Position এ Parity Bit এর মান 1 এসেছে, সেই পজিশনগুলির মান error থাকতো। তাই Hamming Code ব্যবহার করে ত্রুটি সনাক্ত করণ ও সংশোধন করা যায়।