Digital and Logic Circuit

Module 5

Binary Codes, Registers

CILO 5.
Comprehend Binary Codes to understand Digital Computers

Topics

- Binary Codes
- Other Codes
  - Alphanumeric Codes
  - ASCII Code
  - Error Detection Code
  - EBCDIC - Extended BCD (Binary Coded Decimal) Interchange Code
  - BCD or 8421 (BINARY CODED DECIMAL)
  - +3 or EO3 (EXCESS THREE CODE)
  - GRAY CODE or GC (REFLECTED CODE)

Binary Codes

- Computer in its task to do processing uses different and special coding to represent data. This chapter dwells on these coding and also the basic data representation of data in computer world.

  - BITS
    - abbreviation for BINARY DIGITS.
    - smallest information in a computer system.
    - expressed in the form of 0’s and 1’s.
    - Examples:

      02 = 010 12 = 110 102 = 210   10001101010101012

  - NIBBLE
    - string of 4 bits.
    - half of a BYTE.
    - Examples:

      10102 = 1010 or A16   11112 = 1510 or F16

Binary Codes

- Computer in its task to do processing uses different and special coding to represent data. This chapter dwells on these coding and also the basic data representation of data in computer world.

  - BYTE
    - string of 8 bits.
    - basic unit of binary information.
    - It usually forms a character.
    - Example:

      0100 10002 = 6510 = 4116 = A (letter or character, which is different from A16)
  - 1 KB (KiloByte) < > approximately 1000 (thousand) bytes (but not equal to)
    - 1024 bytes
    - 1024 bytes * 8bits/byte = 8192 bits
  - 1 MB (MegaByte) < > approximately 1,000,000 (million) bytes (but not equal to)
    - 1 KB * 1 KB = 1,048,576 bytes
    - 1,048,576 bytes * 8 bits / byte = 8,388,608 bits

Binary Codes

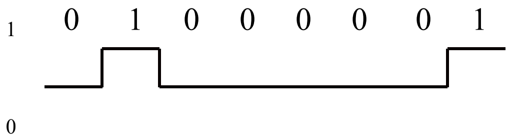
- Computer in its task to do processing uses different and special coding to represent data. This chapter dwells on these coding and also the basic data representation of data in computer world.

  - 1 GB (GigaByte) < > approximately 1,000,000,000 (billion) bytes (but not equal to)
    - » 1 KB * 1 KB * 1 KB = 1,073,741,824 bytes
    - » 1,073,741,824 bytes * 8 bits / byte = 8,589,934,592 bits

  - 1 TB (TerraByte) < > approximately 1,000,000,000,000 (trillion) bytes (but not equal to)
    - » 1 MB * 1 MB = 1.099511628 x 10 12 bytes
    - » 1.099511628 x 10 12 bytes * 8 bits/byte = 8.796093022 x 10 12 bytes

DIGITAL - pertains to anything in the form of digits (which is usually in the form of binary).

$A = 41_{16} = 0100\ 0000_2 =$



AlphaNumeric Code

- The binary codes used to represent alphanumeric data.
- The codes write alphanumeric data, including letters of the alphabet, mathematical symbols, numbers, and punctuation marks, in a form that is easily understood by a computer.

Char.	Zone	Numeric	Hex	Char.	Zone	Numeric	Hex	Char.	Zone	Numeric	Hex
0	0011	0000	30	A	0100	0001	41	P	0101	0000	50
1		0001	31	B		0010	42	Q		0001	51
2		0010	32	C		0011	43	R		0010	52
3		0011	33	D		0100	44	S		0011	53
4		0100	34	E		0101	45	T		0100	54
5		0101	35	F		0110	46	U		0101	55
6		0110	36	G		0111	47	V		0110	56
7		0111	37	H		1000	48	W		0111	57
8		1000	38	I		1001	49	X		1000	58
9	0011	1001	39	J		1010	4A	Y		1001	59
				K		1011	4B	Z	0101	1011	5A
				L		1100	4C				
				M		1101	4D				
				N		1110	4E				
				O	0100	1111	4F				

ASCII Code

- ASCII (pronounced as ask-ee) is the standard code for the alphanumeric character set. ASCII stands for American Standard Code for Information Interchange). It uses 7 bits to code 128 characters and represented by X0 through X6, with X6 the most significant bit.

ASCII control characters

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de linea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	7Fh	DEL	(delete)

ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

theASCIIcode.com.ar

Extended ASCII characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ö	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

ASCII Code

- The control characters are being used for routing of data and arranging the printed text into a prescribed format. There are three types of control characters:

1. FORMAT EFFECTORS – are characters that control the layout of printing. Examples: BS (backspace), HT (horizontal tab) and CR (carriage return)

2. INFORMATION SEPARATORS – are used to separate the data into divisions – for example, paragraphs and pages. Examples: RS (record separator) and FS (file separator)

3. COMMUNICATION CONTROL – are used during the transmission of text from one location to the other. Examples: STX (start of text) and ETX (end of text)

Error Detection Code

- To detect and find errors in data communication processing and transferring data, an eighth bit is sometimes added to the ASCII character to indicate its PARITY. A PARITY BIT is an extra bit included with a message to make the total number of 1's either ODD or EVEN.

  ODD PARITY - if the total number of 1's is ODD.

  EVEN PARITY - if the total number of 1's is EVEN.

- Example:

	With EVEN PARITY	With ODD PARITY
ASCII A = 100 0001	0100 0001	1100 0001
ASCII T = 101 0100	1101 0100	0101 0100

Error Detection Code

- The parity bit is used and of great help in detecting errors during the transmission of information from one location to another. Assuming that even parity is used, the simplest case is handled as follows:

  1. an even parity bit is generated at the sending end for each character
  2. the 8-bit characters that include the parity bits are transmitted to their destination
  3. the parity of each character is then checked at the receiving end
  4. if the parity of the received character is not even, it means that at least one bit has changed its value during the transmission.

Extended BCD (Binary Coded Decimal) Interchange Code

- A. coding system used to represent characters-letters, numerals, punctuation marks, and other symbols in computerized text.
- A character is represented in EBCDIC by eight bit. EBCDIC mainly used on IBM mainframe and IBM mid-range computer operating systems. Each byte consists of two nibbles, each four bits wide. The first four bits define the class of character, while the second nibble defines the specific character inside that class.
- EBCDIC is different from, and incompatible with, the ASCII character set used by all other computers. The EBCDIC code allows for 256 different characters. For personal computers, however, ASCII is the standard. If you want to move text between your computer and a mainframe, you can get a file conversion utility that will convert between EBCDIC and ASCII.

Table 4 - EBCDIC Codes

Symbol	EBCDIC	Symbol	EBCDIC
(space)	01000000	?	01101111
!	01011010	@	01111100
"	01111111	A	11000001
#	01111011	B	11000010
$	01011011	C	11000011
%	01101100	D	11000100
&	01010000	E	11000101
'	01111101	F	11000110
(	01001101	G	11000111
)	01011101	H	11001000
*	01011100	I	11001001
+	01001110	J	11010001
,	01101011	K	11010010
-	01100000	L	11010011
.	01001011	M	11010100
/	01100001	N	11010101
0	11110000	O	11010110
1	11110001	P	11010111
2	11110010	Q	11011000
3	311110011	R	11011001
4	11110100	S	11100010
5	11110101	T	11100011
6	11110110	U	11100100
7	11110111	V	11100101
8	11111000	W	11100110
9	11111001	X	11100111
:	01111010	Y	11101000
;	01011110	Z	11101001
<	01001100	[	01001010
=	01111110	\	NONE
>	01101110	]	01011010

BCD or 8421 (BINARY CODED DECIMAL)

- The BCD is used to represent decimal numbers in its convenient form. To write the decimal digits 0 – 9 in binary, it is necessary to use 4 binary digits

- TAKENOTE : Basis is DECIMAL base.

1. $893_{10} = 1000\ 1001\ 0011_{BCD}$

   $$\begin{array}{ccc} 8421 & 8421 & 8421 \\ 8 = 1000 & 9 = 1001 & 3 = 0011 \end{array}$$

2. $123_8 = 83_{10}$

   $= 1000\ 0011_{BCD}$

   $$\begin{array}{cc} 8421 & 8421 \\ 8 = 1000 & 3 = 0011 \end{array}$$

+3 or EO3 (EXCESS THREE CODE)

- Derived in the same way as the BCD code except that before conversion into binary each digit is increased by 3.
- TAKENOTE : Basis is DECIMAL base.

5. $893_{10} = 1011\ 1100\ 0110_{+3}$

$$\begin{aligned} & \quad 8421 \\ 8+3 &= 1011 \\ 9+3 &= 1100 \\ 3+3 &= 0011 \end{aligned}$$

6. $123_8 = 83_{10}$

$$\begin{aligned} &= 1011\ 0011_{+3} \\ & \quad 8421 \\ 8+3 &= 1011 \\ 3+3 &= 0011 \end{aligned}$$

Gray Code or GC (Reflected Code)

- A sequence of binary numbers in which one and only one digit changes in succession. It is also called the "UNIT DISTANCE" code and is used in K-Maps.

- TAKENOTE : Basis is BINARY base.

  - Conversion from BINARY to GRAY CODE
    1. The 1st bit of the gray code is the SAME as the 1st bit of the binary number.
    2. The 2nd bit of the gray code equals the Exclusive OR of the 1st and 2nd bit, that is, it will be a 1 if the binary bits are DIFFERENT, 0 if they are the SAME.

  - Conversion from GRAY CODE to BINARY
    1. The 1st binary bit is the same as the 1st gray bit.
    2. The 2nd gray bit is 0, the 2nd binary bit is the SAME as the 1st binary bit, if the 2nd gray bit is 1, the 2nd bit is the INVERSE of the 1st binary bit

**See recording for example conversion.*