NAAC ACCREDITED "B++" (CGPA 2.89)

YOGODA SATSANGA EDUCATIONAL INSTITUTIONS

Yogoda Satsanga Mahavidyalaya

Jagannathpur, Dhurwa, Ranchi-834004

www.ysmranchi.net

ysmranchi4@gmail.com

Course :Computer System Architecture

Class : Sem-1

Lesson :Digital Code

By :Goutam Sanyal

CODES

A set of 4 distinct numbers can be represented by 2-bit codes s.t. each number in the set is assigned exactly one of the combinations/codes in {00,01,10,11}

CODES (cont.)

- Representations of info (set) obtained by associating one or more <u>codewords</u> (a binary pattern/string) with each element in the set

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

- *n-bit* binary code: a group of *n* bits that can encode up to 2^n distinct elements

Non-numeric Binary Codes

- **Given *n* binary digits (called bits), a binary code is a mapping from a set of represented elements to a subset of the 2^n binary numbers.**
- **Code 100 is not used**

Color	Binary Number
Red	000
Orange	001
Yellow	010
Green	011
Blue	101
Indigo	110
Violet	111

To encode *m* distinct elements with an *n-bit* code: $2^n >= m$

Representing Numeric and Characters Using Standard Binary codes

Code to represent numeric

- BCD

Code to represent Character

- EBCDIC – developed for the IBM 360 and used in all IBM mainframes since then
  - An 8-bit representation for 256 characters
- ASCII – used in just about every other computer
  - A 7-bit representation plus the high-order bit used for parity
- Unicode – newer representation to include non-Latin based alphabetic characters
  - 16 bits allow for 65000+ characters
  - It is downward compatible with ASCII, so the first 128 characters are the same as ASCII

Binary-Coded Decimal (BCD)

- A decimal code: Decimal numbers (0..9) are coded using 4-bit distinct binary words
- Observe that the codes 1010 .. 1111 (decimal 10..15) are NOT represented (invalid BCD codes)

Decimal	BCD code Representation
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Convert each of the following decimal numbers to BCD:

(a) 35 **(b)** 98 **(c)** 170 **(d)** 2469

Solution

(a) $\overbrace{\mathbf{0011}}^{3\downarrow}\overbrace{\mathbf{0101}}^{5\downarrow}$

(b) $\overbrace{\mathbf{1001}}^{9\downarrow}\overbrace{\mathbf{1000}}^{8\downarrow}$

(c) $\overbrace{\mathbf{0001}}^{1\downarrow}\overbrace{\mathbf{0111}}^{7\downarrow}\overbrace{\mathbf{0000}}^{0\downarrow}$

(d) $\overbrace{\mathbf{0010}}^{2\downarrow}\overbrace{\mathbf{0100}}^{4\downarrow}\overbrace{\mathbf{0110}}^{6\downarrow}\overbrace{\mathbf{1001}}^{9\downarrow}$

Convert each of the following BCD codes to decimal:

(a) 10000110 **(b)** 001101010001 **(c)** 1001010001110000

Solution

(a) 1000 0110 → **8** **6**

(b) 0011 0101 0001 → **3** **5** **1**

(c) 1001 0100 0111 0000 → **9** **4** **7** **0**

BCD Addition

Step 1: Add the two BCD numbers, using the rules for binary addition in Section 2–4.

Step 2: If a 4-bit sum is equal to or less than 9, it is a valid BCD number.

Step 3: If a 4-bit sum is greater than 9, or if a carry out of the 4-bit group is generated, it is an invalid result. Add 6 (0110) to the 4-bit sum in order to skip the six invalid states and return the code to 8421. If a carry results when 6 is added, simply add the carry to the next 4-bit group.

Add the following BCD numbers:

(a) 0011 + 0100 (b) 00100011 + 00010101

(c) 10000110 + 00010011 (d) 010001010000 + 010000010111

Solution

The decimal number additions are shown for comparison.

(a)

0011	3
+ 0100	+ 4
0111	7

(b)

0010	0011	23
+ 0001	0101	+ 15
0011	**1000**	38

(c)

1000	0110	86
+ 0001	0011	+ 13
1001	**1001**	**99**

(d)

0100	0101	0000	450
+ 0100	0001	0111	+ 417
1000	**0110**	**0111**	**867**

Add the following BCD numbers:

(a) 1001 + 0100 (b) 1001 + 1001

(c) 00010110 + 00010101 (d) 01100111 + 01010011

(a)		1001		9
		+ 0100		+4
		1101	Invalid BCD number (>9)	13
		+ 0110	Add 6	
	0001	**0011**	Valid BCD number	
	↓	↓		
	1	3		

(b)		1001		9
		+ 1001		+ 9
	1	0010	Invalid because of carry	18
		+ 0110	Add 6	
	0001	**1000**	Valid BCD number	
	↓	↓		
	1	8		

(c)

```
      0001      0110                                        16
    + 0001      0101                                      + 15
      0010      1011   Right group is invalid (>9),         31
                         left group is valid.
              + 0110   Add 6 to invalid code. Add
                         carry, 0001, to next group.
      0011      0001   Valid BCD number
        ↓         ↓
        3         1
```

(d)

```
             0110        0111                                 67
           + 0101        0011                               + 53
             1011        1010   Both groups are invalid (>9)  120
           + 0110      + 0110   Add 6 to both groups
    0001     0010        0000   Valid BCD number
      ↓        ↓           ↓
      1        2           0
```

Excess-3 code

- This code is also self-complementing like 2421 code.
- Although this code is not weighted, it has an arithmetic relationship with the BCD code.
- The code word for each decimal digit is the corresponding BCD code word plus 0011_2.

$0010 = 2$ in BCD

$+ \; 0011_2$

$= 0101 = 2$ in excess-3

The Excess-3- Code

DECIMAL	BCD	EXCESS-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

(a) 13

	1	3	
+	3	3	
	4	6	
	0100	0110	Excess-3

(b) 430

(b)	4	3	0	
	3	3	3	
	7	6	3	
	0111	0110	0011	Excess-3

ASCII character code

- We also need to represent letters and other symbols → alphanumeric codes

- ASCII = American Standard Code for Information Interchange. Also know as Western European

- It contains 128 characters:
  - 94 printable (26 upper case and 26 lower case letters, 10 digits, 32 special symbols)
  - 34 non-printable (for control functions)

- Uses 7-bit binary codes to represent each of the 128 characters(Since 2^7=128)

ASCII Table

	$B_7B_6B_5$							
$B_4B_3B_2B_1$	000	001	010	011	100	101	110	111
0000	NULL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB		7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	')	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Unicode

- Established standard (16-bit alphanumeric code) for international character sets
- Since is 16-bit, it has 65,536 codes
- Represented by 4 Hex digits
- ASCII is between 0000_{16} .. $007B_{16}$

Parity

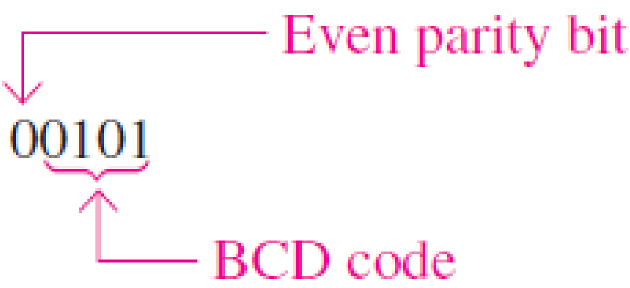
- The method of parity is widely used as a method of error detection.
  - Extra bit known as parity is added to data word
  - The new data word is then transmitted.
- Two systems are used:
  - Even parity: the number of 1's must be even.
  - Odd parity: the number of 1's must be odd.

The BCD code with parity bits.

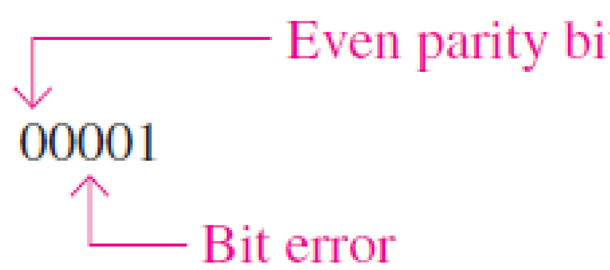
Even Parity		Odd Parity	
P	**BCD**	*P*	**BCD**
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

Error Detection

- A parity bit provides for the detection of a single bit error but cannot check for two errors in one group.
- For instance, let's assume that we wish to transmit the BCD code 0101.
- The total code transmitted, including the even parity bit, is



Now let's assume that an error occurs in the third bit from the left (the 1 becomes a 0).



Because an even number of 1s does not appear in the code when it is received, an error is indicated.

ASCII Parity Bit

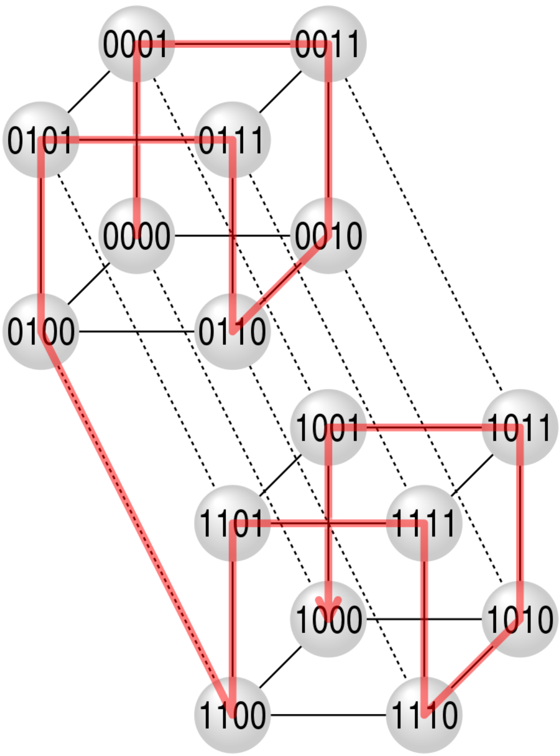
- Parity coding is used to detect errors in data communication and processing
- An 8^{th} bit is added to the 7-bit ASCII code
- Even (Odd) parity: set the parity bit so as to make the number of 1’s in the 8-bit code even (odd)

ASCII Parity Bit (cont.)

- For example:
  - Make the 7-bit code 1011011 an 8-bit even parity code → 11011011
  - Make the 7-bit code 1011011 an 8-bit odd parity code → 01011011
- Both even and odd parity codes can detect an odd number of error. An even number of errors goes undetected.

Gray Code

Bell Labs researcher Frank Gray introduced the term *reflected binary code* in his 1947 patent application, remarking that the code had "as yet no recognized name".He derived the name from the fact that it "may be built up from the conventional binary code by a sort of reflection process".



Gray Code

The **Gray code** is unweight and is not an arithmetic code; that is, there are no specific weights assigned to the bit positions.
The important feature of the Gray code is that ***it exhibits only a single bit change from one code word to the next in sequence***.

This property is important in many applications, such as shaft position encoders, converting analog values into digital values

Four-bit Gray code.

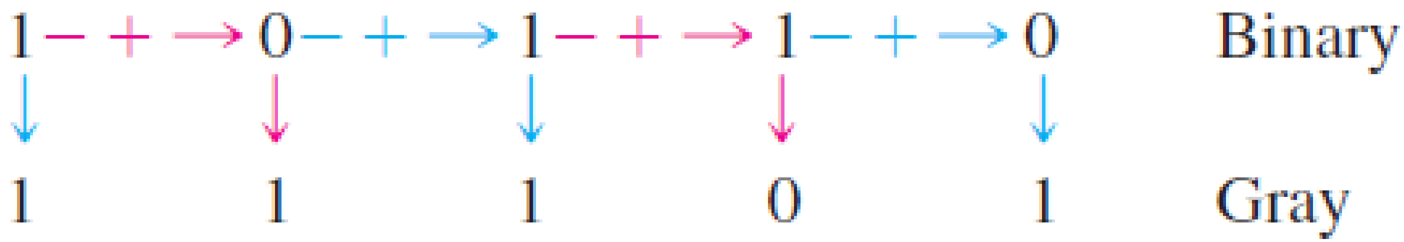
Decimal	Binary	Gray Code	Decimal	Binary	Gray Code
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Binary-to-Gray Code Conversion

1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.

2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

For example, the conversion of the binary number 10110 to Gray code is as follows:



The Gray code is 11101.

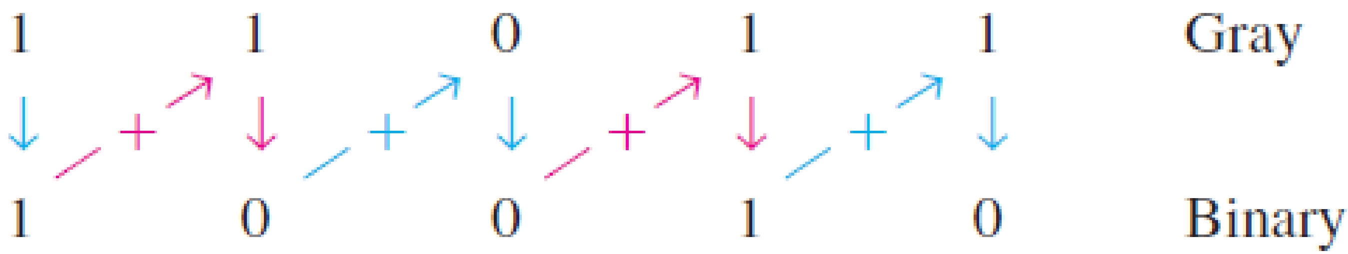
Gray-to-Binary Code Conversion

1. The most significant bit (left-most) in the binary code is the same as the corresponding

bit in the Gray code.

2. Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

For example, the conversion of the Gray code word 11011 to binary is as follows:



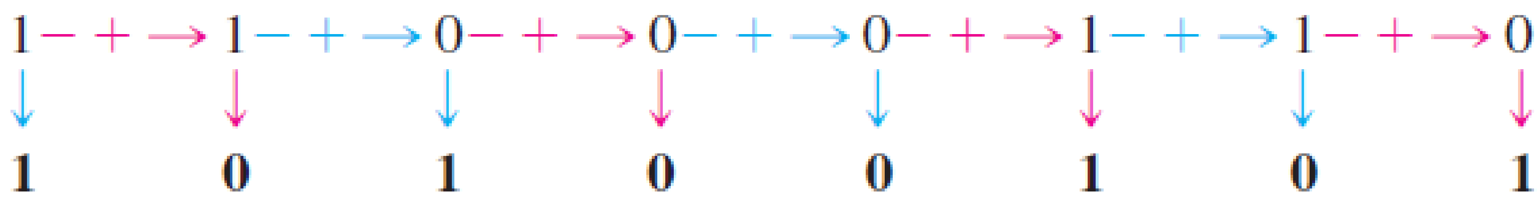
The binary number is 10010.

(a) Convert the binary number 11000110 to Gray code.

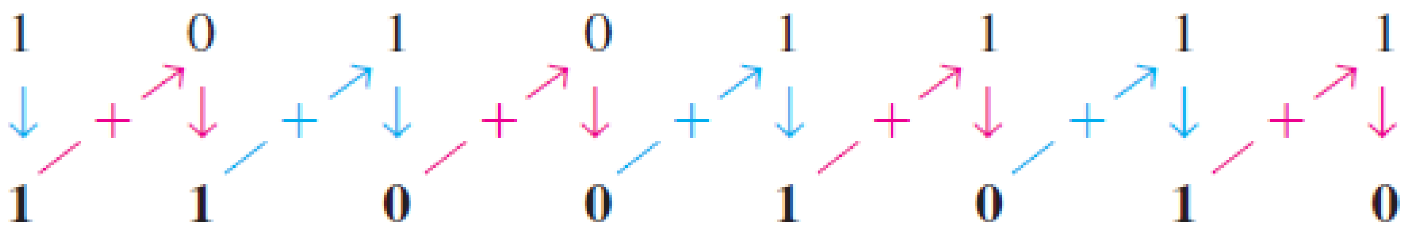
(b) Convert the Gray code 10101111 to binary.

Solution

(a) Binary to Gray code:



(b) Gray code to binary:



Thank You