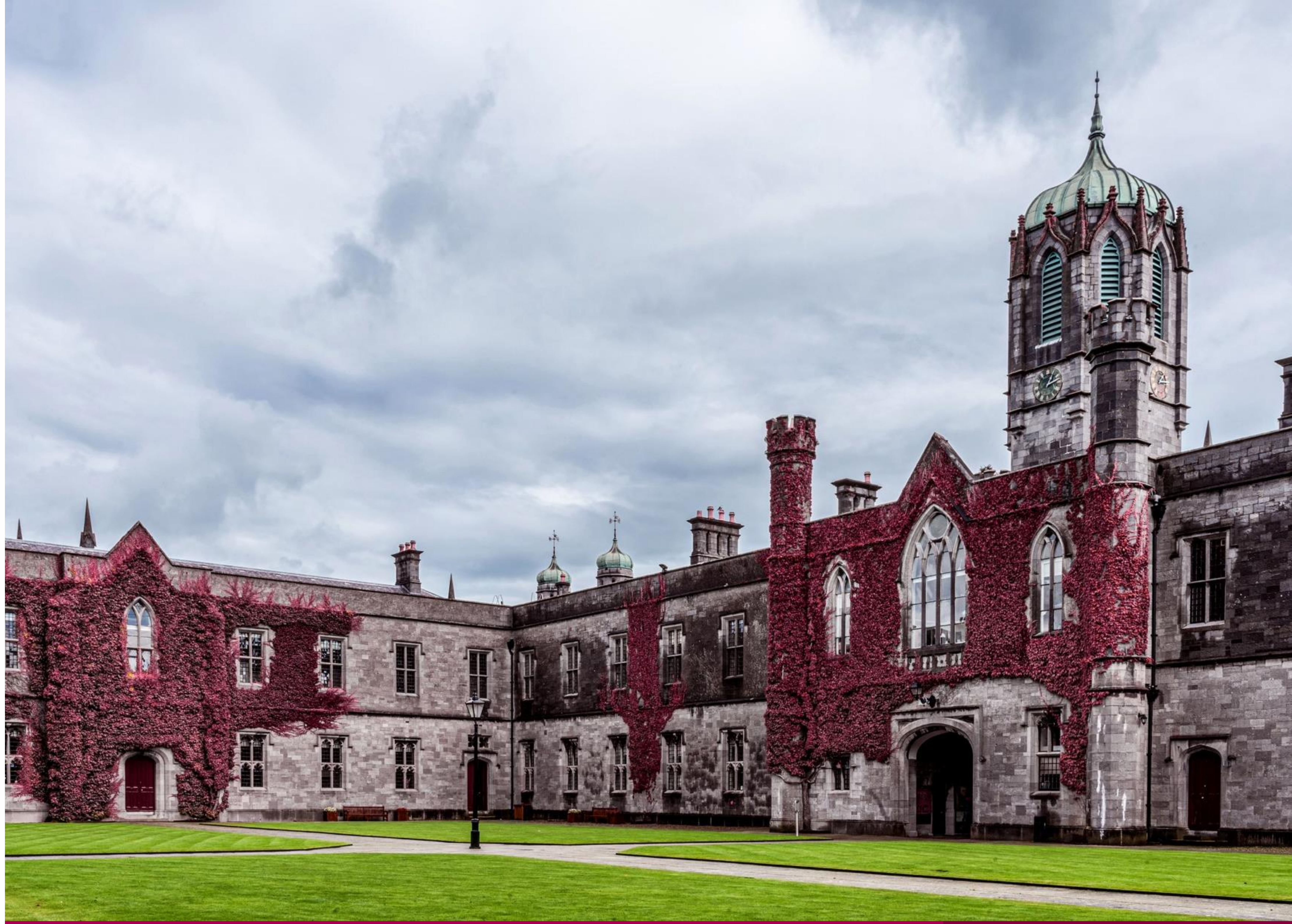
Ollscoil na Gaillimhe
University of Galway

CT101 Computing Systems

Dr. Bharathi Raja Chakravarthi
Lecturer-above-the-bar
Email: bharathi.raja@universityofgalway.ie



University ofGalway.ie

Complements of Numbers

- Complements are used in digital computers to **simplify the subtraction operation and for logical manipulation**.

- Simplifying operations leads to **simpler, less expensive circuits** to implement the operations.

- **Two types** of complements for each base-r system:
  - the radix complements (**r’s complement**)
  - the diminished radix complements (**(r - 1)’s complement**)

- Value of base r is substituted in the name, then
  - 2’s complement and 1’s complement
  - 10’s complement and 9’s complement

Diminished Radix Complement

- **(r - 1)'s** complement of N is **$(r^n - 1) - N$**

Where,
N - number
r - base
n - digits

- **For decimal numbers**,
r = 10 and **r - 1 = 9**, 9's complement of N is

$(10^n - 1) - N$

- In this case, **10^n** represents a number that consists of a single **1** followed by **n 0's.**

10^n - 1 is a represented by n 9's

Decimal 9's Complement

- **For example,**
  - ❖ if **n = 4**,

    $10^4 = 10{,}000$ and $10^4 - 1 = 9999_{10}$.

- Here are some numerical examples:
  - ❖ 9’s complement of $\mathbf{546700_{10}}$ is

$$
\begin{array}{r}
999999_{10} \\
-\ 546700_{10} \\
\hline
\mathbf{453299_{10}}
\end{array}
$$

  - ❖ 9’s complement of $\mathbf{012398_{10}}$ is

$$
\begin{array}{r}
999999_{10} \\
-\ 012398_{10} \\
\hline
\mathbf{987601_{10}}
\end{array}
$$

Diminished Radix Complement

- **For binary numbers**,
  r = 2 and **r - 1 = 1**, so the 1's complement of N is

  $$(2^n - 1) - N$$

- Again, 2^n is represented by a binary number that consists of a 1 followed by n 0's.

  $2^n - 1$ is a binary number represented by n 1's

Diminished Radix Complement

- **For example**,

  if **n = 4**,

  $2^4 = (10000)_2$ and $2^4 - 1 = 15_{10} = (1111)_2$

- Thus, the 1's complement of a binary number is obtained by subtracting each digit from 1.

- when subtracting binary digits from 1, causes the bit to change from 0 to 1 or from 1 to 0

  1 - 0 = 1 or **1 - 1 = 0**

- Therefore, the 1's complement of a binary number is formed by changing 1's to 0's and 0's to 1's.

Binary 1's Complement

- For $r = 2$, $N = 01110011_2$, $n = 8$ (8 digits), we have:
  $(r^n - 1) = 256 - 1 = 255_{10}$ or $\mathbf{11111111_2}$

- The 1's complement of $\mathbf{01110011_2}$ is then:

$$
\begin{array}{r}
1111\ 1111_2 \\
-\ 0111\ 0011_2 \\
\hline
\mathbf{10001100_2}
\end{array}
$$

Radix Complement

- r's complement of N is

  $\mathbf{r^n - N}$ for $N \neq 0$ and **0** for $N = 0$.

  Where,
  N - number
  r - base
  n - digits

- Radix complement is obtained **by adding 1** to the Diminished Radix Complement

$$\mathbf{r^n - N = [(r^n - 1) - N] + 1}$$

Decimal 10's Complement

The 10’s complement of decimal 2389 is

$$\underbrace{\begin{array}{r} 9999_{10} \\ -\ 2389_{10} \\ \hline \mathbf{7610_{10}} \end{array}}_{\textbf{9's complement}}$$

$$\underbrace{\begin{array}{r} 7610_{10} \\ +\qquad 1_{10} \\ \hline \mathbf{7611_{10}} \end{array}}_{\textbf{10's complement}}$$

Decimal 10's Complement

For Example, 10's complement of 012398

$$
\left.\begin{array}{r} 999999_{10} \\ -\ 012398_{10} \\ \hline \mathbf{987601_{10}} \end{array}\right\} \textbf{9's complement}
$$

$$
\left.\begin{array}{r} 987601_{10} \\ +\quad\quad 1_{10} \\ \hline \mathbf{987602_{10}} \end{array}\right\} \textbf{10's complement}
$$

Solve the problem

Find the 10's complement of $\mathbf{246700_{10}}$.

Solve the problem

Find the 10's complement of $\mathbf{246700_{10}}$.

$$\begin{array}{r} 999999_{10} \\ -\ 246700_{10} \\ \hline \mathbf{753299_{10}} \end{array} \quad \text{9's complement}$$

$$\begin{array}{r} 753299_{10} \\ +\ \quad 1_{10} \\ \hline \mathbf{753300_{10}} \end{array} \quad \text{10's complement}$$

Binary 2's Complement

The 2’s complement of binary 101100 is

$$\begin{array}{r} 111111_2 \\ -\ 101100_2 \\ \hline \mathbf{010011_2} \end{array} \quad \textbf{1's complement}$$

$$\begin{array}{r} 010011_2 \\ +\quad\quad 1_2 \\ \hline \mathbf{010100_2} \end{array} \quad \textbf{2's complement}$$

Solve the problem

Find the 2's complement of **1101100_2**.

Solve the problem

Find the 2's complement of $\mathbf{1101100_2}$.

$$\begin{array}{r} 1111111_2 \\ -\ 1101100_2 \\ \hline \mathbf{0010011_2} \end{array}$$

1's complement

$$\begin{array}{r} 0010011_2 \\ +\qquad 1_2 \\ \hline \mathbf{0010100_2} \end{array}$$

2's complement

Efficient 2's Complement

Given: an n-bit binary number:

$$\mathbf{a_{n-1}\, a_{n-2} \ldots a_{i+1}\, 1\underline{0}\ldots\underline{00}}$$

Where for some digit position i, a_i is 1 and all digits to the right are 0, form the 2's complement value this way:

- ✓ Leave a_i equal to 1 (unchanged),
- ✓ Leave rightmost digits 0 (unchanged)
- ✓ Complement all other digits to the left of a_i(0 replaces 1, 1 replaces 0)

The complement of the complement restores the number to its original value.

Note: the r's complement of N is $\mathbf{r^n - N}$, so that the complement of the complement is $\mathbf{r^n - (r^n - N) = N}$ and is equal to the original number.

Efficient 2's Complement

- **First 1 from right**

0110101110001110000

- **Complement leftmost digits**

<u>**1001010001110**</u>0100000

0110100111100

replaced

<u>**1001011000**</u>100

1000000000000

unchanged

1000000000000

Subtraction with Complements

Subtracting two n-digit unsigned numbers, **M-N** in base **r**:

1. Add the **M** to the **r's** complement of **N**.

$$M + (r^n - N) = (M - N) + r^n$$

2. If **M ≥ N**, the sum will produce an end carry, i.e., $\mathbf{r^n}$, which can be discarded to produce M - N

3. If **M < N**, the sum does not produce an end carry. Apply r's complement on the sum & place a –ve sign in front.

$$\left.\begin{array}{c} r^n - (N - M) \\ \text{or} \\ -(r^n + (M-N)) \end{array}\right\} \text{r's complement of } \mathbf{(N - M)}$$

Example

Find **543_{10} - 123_{10}**

M **N**

1. 10's complement of 123_{10}:

$$\begin{array}{r} 1000 \\ -\ 123 \\ \hline 877 \end{array}$$

2. Add the two:

$$\begin{array}{r} 543 \\ +\ 877 \\ \hline \boxed{1}\boxed{420} \end{array}$$

carry

3. Since M ≥ N, we discard the carry.

Ans. **420**

Example

Find **123_{10} - 543_{10}**

M **N**

1. 10's complement of 543_{10}:

$$\begin{array}{r} 1000 \\ -\ 543 \\ \hline 457 \end{array}$$

2. Add the two:

$$\begin{array}{r} 123 \\ +\ 457 \\ \hline \boxed{580} \end{array}$$

No carry

3. Since M < N, Perform r's complement

$$\begin{array}{r} 1000 \\ -\ 580 \\ \hline -\ 420 \end{array}$$

Place –ve sign in front

Example

Compute $\mathbf{1010100_2 - 1000011_2}$

$$\underset{M}{\mathbf{1010100_2}} - \underset{N}{\mathbf{1000011_2}}$$

1. 2's complement of 1000011_2:

$$\begin{array}{c} \mathbf{1000011} \\ \downarrow \\ \mathbf{0111101} \end{array}$$

2. Add the two:

$$\begin{array}{r} \mathbf{1010100} \\ \mathbf{0111101} \\ \hline \boxed{\mathbf{1}}\ \boxed{\mathbf{0010001}} \end{array}$$

carry

3. Since M ≥ N, we discard the carry.

Ans. **0010001**

Example

Compute $\mathbf{1000011_2 - 1010100_2}$

M **N**

1. 2's complement of 1010100_2: **1010100** → **0101100**

2. Add the two:

$$
\begin{array}{r}
\mathbf{1000011} \\
\mathbf{+\ 0101100} \\
\hline
\boxed{\mathbf{1101111}}
\end{array}
$$

No carry

3. Since M < N, Perform r's complement **0010001**

Place –ve sign in front → **Ans. - 0010001**

Signed Binary Numbers

- Positive numbers and zero can be represented by unsigned n-digit, radix r numbers.
- We need a representation for negative numbers.
- To represent a sign (+ or -) we need exactly one more bit of information (1 binary digit gives $2^1 = 2$ elements which is exactly what is needed).
- The most significant bit (MSB) is interpreted as a sign bit as shown below:

$$sa_{n-2} \ldots a_2a_1a_0$$

Where:
s = 0 for Positive numbers
s = 1 for Negative numbers
ai are 0 or 1

Signed Binary Numbers

Decimal	Signed-2's Complement	Signed-1's Complement	Signed Magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

Interpreting the Other Digits

Given n binary digits,

- the digit with **weight 2(n-1) is the sign** and
- the digits with **weights 2(n-2) down to 2(0) represents 2(n-1)** distinct elements.

There <u>two popular ways</u> to interpret the other digits:

1. Signed-Magnitude
2. Signed-Complement
    a) Signed One's Complement
    b) Signed Two's Complement

Signed-magnitude representation

01001 → 9 (unsigned) or +9 (signed)

leftmost bit is 0 denoted positive

Positive

11001 → 25 (unsigned) or -9 (signed)

leftmost bit is 1
denotes negative

bits represent binary 9

Negative

Signed complement representation

Signed 1’s complement representation of -9:

```
  111 11
   01001
  ------
  1 0110
```

$\boxed{1}\,\boxed{0110}$

denotes -ve

1’s complement of +9

Signed 2’s complement representation of -9:

```
  1 0110   ← 1’s complement of +9
       1
  ------
  1 0111
```

$\boxed{1}\,\boxed{0111}$

denotes -ve

2’s complement of +9

Binary Codes

- A binary code represents text, computer processor instructions, or any other data using a two-symbol system.

- The two-symbol system used is often "**0**" and "**1**" from the binary number system.

- The binary code assigns a pattern of binary digits, also known as bits, to each character, instruction, etc.

- For example, a binary string of eight bits (which is also called a byte) can represent any of 256 possible values and can, therefore, represent a wide variety of different items.

https://en.wikipedia.org/wiki/Binary_code

Binary-Coded Decimal Code

- It is commonly known as BCD.
- BCD code is a weighted code, so in this code each digit is assigned a specific Weight according to its position.
- BCD code is also known as 8421 code.
- This is because 8,4,2, and 1 are the weights of the four bits of the BCD code.
- The weight of the LSB is 2^0 or 1, next higher order 2^1 or 2 and next 2^2 or 4 and MSB is 2^3 or 8.

Binary-Coded Decimal Code

- To represent 10 decimal digits, it is necessary to use atleast 4 binary bits.
- For each decimal digits (0 to 9) is represented by unique combination of bits
- So, there will be six unused or invalid combination (10 to 15) in BCD code.

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

Decimal to BCD Number

$(12)_{10} = (?)_2$

$(12)_{10}$

0001 0011

$$\mathbf{(12)_{10} = (00010010)_{BCD}}$$

BCD to Decimal number

$\mathbf{(1100111)_{BCD} = (?)_{10}}$

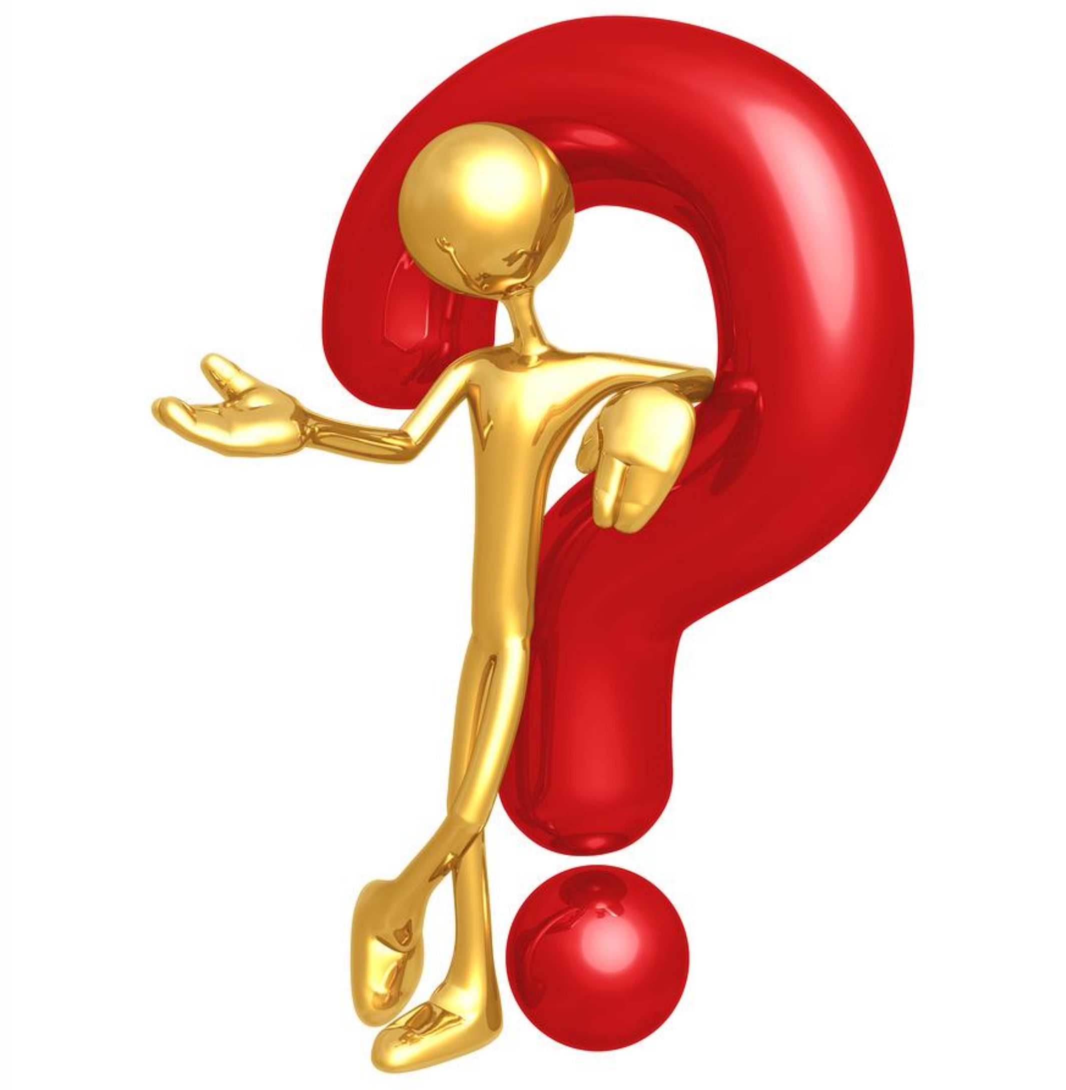
0001 0110

6 7

$(0110010110)_{BCD} = \mathbf{(67)_{10}}$

Solve the problem

Convert BCD to Decimal number $\mathbf{(0110010110)_{BCD} = (?)_{10}}$



Solve the problem

Convert BCD to Decimal number $\mathbf{(0110010110)_{BCD} = (?)_{10}}$

$$\underline{0001}\quad\underline{1001}\quad\underline{0110}$$

$$\downarrow\qquad\quad\downarrow\qquad\quad\downarrow$$

$$1\qquad\quad 9\qquad\quad 6$$

$(0110010110)_{BCD} = \mathbf{(196)_{10}}$

BCD Addition

$(8)_{10} + (4)_{10} = (?)_{BCD}$

$$
\begin{array}{lrr}
\text{BCD of 8} & & 1000 \\
\text{BCD of 4} & & 0100 \\
\hline
 & & \boxed{1100} \\
\text{Add 6} & & 0110 \\
\hline
 & \underbrace{0001}_{\mathbf{1}} & \underbrace{0010}_{\mathbf{2}}
\end{array}
$$

Invalid BCD

$(8)_{10} + (4)_{10} = (0001\ \ 0010)_{BCD} = (12)_{10}$

Gray Code

- Gray code is a non- weighted code and is a special case of unit- distance code.
- In unit distance code, bit patterns for two consecutive numbers differ in only one bit position. These codes are also called as cyclic codes
- The gray code is also called reflected code.

Gray Code

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

Other Decimal codes

Four difference binary codes for the Decimal digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Unused bit combi-nations	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

ASCII Character Code

- ASCII stands for American Standard Code for Information Interchange.
- ASCII code is the numerical representation of a characters.
- The table right shows the code for each character.

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	000	001	010	011	100	101	110	111
0000	NUL	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

Error-Detecting Code

- When the digital information is transmitted from one circuit to another circuit an error may occur.

- This means the signal corresponding to 0 may change to 1 or vice-versa due to presence of noise.

- To maintain data integrity between transmitter and receiver, extra bit or more than one bit are added in the data.

Error-Detecting Code

- These extra bits allow the detection and sometimes the correction of error in the data.
- The data along with the extra bit/ bits form the code.
- Codes which allow only error detection are called error detecting codes and codes which allow error detection and correction are called error detecting and correcting codes.

Error-Detecting Code

Parity bit:

- It is an extra bit included with a message to make the total no. of 1s either odd or even.
- The message including the parity bit is transmitted and then checked at the receiving end for errors.
- An error is detected if the checked party does not correspond with the one transmitted.

References

- Computer Organization and Architecture Designing for Performance Tenth Edition by William Stallings
- Digital Design With an Introduction to the Verilog HDL FIFTH EDITION by M Morris, M. and Michael, D., 2013.

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Thank *you*