Digital Electronics

(Second Year B. Tech program in Computer Engineering)

Digital Electronics

(Second Year B. Tech program in Computer Engineering)

Semester-III (w.e.f. 2021-22)														
Sr	Course Category	Course Code	Course Title	Teaching Scheme			Evaluation Scheme					Total	Credits
							Continuous Assessment (CA)				ESE		
							TA	Term Test 1 (TT1)	Term Test 2 (TT2)	Average (TT1 & TT2)			
				L	T	P	[A]			[B]	[C]	[A+B+C]	
7	ES	ESCO3050T	Digital Electronics	3			20	15	15	15	65	100	3
8	ESL	ESCO3050L	Digital Electronics Laboratory			2	25				25	50	1

Digital Electronics (ESCO3050T)

Teaching Scheme	**Examination Scheme**
Lectures : 03 Hrs./week	Term Test : 15 Marks
Credits : 03	Teacher Assessment : 20 Marks
	End Sem Exam : 65 Marks
	Total Marks : 100 Marks

Course Objectives:

1. To introduce the fundamental concepts and methods for design of Digital circuits and a prerequisite for Computer Organization and Architecture, Microprocessor Systems.
2. To provide the concept of designing Combinational and Sequential circuits.
3. To provide basic knowledge of how digital building blocks are described in VHDL.

CO	Course Outcomes	Blooms Level	Blooms Description
CO1	Explain different Number Systems and their conversions.	L2	Understand
CO2	Examine and minimize Boolean expressions.	L4, L6	Analyze, Create
CO3	Design and analyze Combinational circuits.	L6, L4	Create, Analyze
CO4	Design and analyze Sequential circuits.	L6, L4	Create, Analyze
CO5	Design and analyze Counters and Registers.	L6, L4	Create, Analyze
CO6	Explain concept of Programming Logic Devices.	L2	Understand

Course Contents

Unit-I Number Systems and Codes 07 Hrs.

Introduction to Number System and Conversions: Binary, Octal, Decimal and Hexadecimal Number Systems, Binary Arithmetic: Addition, Subtraction (1's and 2's Complement), Multiplication and Division.Octal and Hexadecimal Arithmetic: Addition and Subtraction (7's and 8's Complement Method for Octal) and (15's and 16's Complement Method for Hexadecimal).

Codes: Gray Code, BCD Code, Excess-3 Code, Error Detection and Correction: Hamming Codes.

Unit-II Boolean Algebra and Logic Gates 08 Hrs.

Theorems and Properties of Boolean Algebra, Boolean Functions, Boolean Function Reduction using Boolean Laws, Canonical Forms, Standard SOP and POS Form. Basic Digital Gates: NOT, AND, OR, NAND, NOR, EX-OR, EX-NOR, Positive and Negative Logic, K-Map Method: 2-variable, 3-variable, 4-variable, Don't-Care Conditions, Quine-McCluskey Method, NAND-NOR Realization.

Unit-III Combinational Logic Design 08 Hrs.

Introduction: Half and Full Adder, Half and Full Subtractor, 4-bit Ripple Adder, Look-Ahead Carry Adder, 4-bit Adder and Subtractor, 1- digit BCD Adder, Multiplexers, Multiplexer Tree, Demultiplexers, Demultiplexer Tree, Encoders, Priority Encoder, Decoders. Comparators: 1-bit, 2-bit, 4-bit Magnitude Comparators, ALU IC 74181.

Unit-IV Sequential Logic Design 09 Hrs.

Application of Sequential Logic, Introduction: S-R Latch, S-R, D, J-K, T Flip-Flops, Truth Tables and Excitation Tables of all Types, Race-Around Condition, Master-Slave J-K Flip-Flops, Flip-Flop Conversion.

Unit-V Counters 07 Hrs.

Design of Asynchronous and Synchronous Counters, Modulus of the Counters, UP/DOWN Counter, Shift Registers: SISO, SIPO, PIPO, PISO Bi-directional Shift Register, Universal Shift Register, Ring and Twisted-Ring/Johnson Counter, Sequence Generator.

Unit-VI Programming Logic Devices 03 Hrs.

Concepts of Programmable Array Logic (PAL) and Programming Logic Array (PLA). Introduction to Sensors.

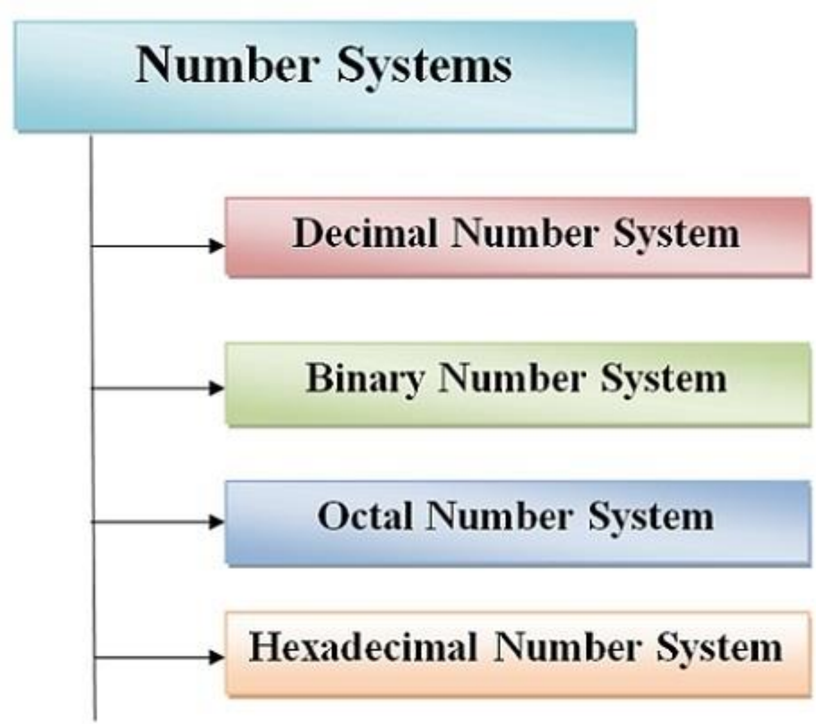
Text Books:

1. R. P. Jain, Modern Digital Electronics, 4^{th} Edition, McGraw-Hill Education.
2. M. Morris Mano, Digital Logic and Computer Design, 1^{st} Edition, PHI.
3. Norman Balabanian, Digital Logic Design Principles, Student Edition, Wiley Publications.
4. J. Bhasker, VHDL Primer, 3^{rd} Edition, Pearson Education.

Reference Books:

1. Donald P. Leach and Albert Paul Malvino, Digital Principles and Applications, 8^{th} Edition, McGraw- Hill Education.
2. Yarbrough John M, Digital Logic Applications and Design, 2016 Edition, Cengage Learning.
3. Douglas L. Perry, VHDL Programming by Example, 4^{th} Edition, McGraw-Hill Education.

Number System:



Numbering System		
System	Base	Digits
Binary	2	0, 1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Definitions:

- Hexadecimal: *"Hexadecimal" means "based on 16" (From Greek hexa: "six" and Latin decima: "a tenth part").*
- Decimal: *Based on 10; Example: the numbers we use in everyday life are decimal numbers, because there are 10 of them (0,1,2,3,4,5,6,7,8 and 9).*
- Denary: *Same as Decimal – Base 10*
- Binary: *The word binary comes from "Bi-" meaning two. We see "bi-" in words such as "bicycle" (two wheels) or "binocular" (two eyes). Binary only uses 2 digits; 1 & 0*
- Octal: *An Octal Number uses only these 8 digits: 0, 1, 2, 3, 4, 5, 6 and 7 Examples:*
  - *10 in Octal equals 8 in the Decimal Number System.*
  - *167 in Octal equals 119 in the Decimal Number System.*

  Also called Base 8.

Decimal to Binary Conversion

Binary to Decimal Conversion

Decimal to Octal Conversion

Octal to Decimal Conversion

Decimal to Hexadecimal Conversion

Hexadecimal to Decimal Conversion

Hexadecimal to Binary Conversion

Binary to Hexadecimal Conversion

Octal to Binary Conversion

Binary to Octal Conversion

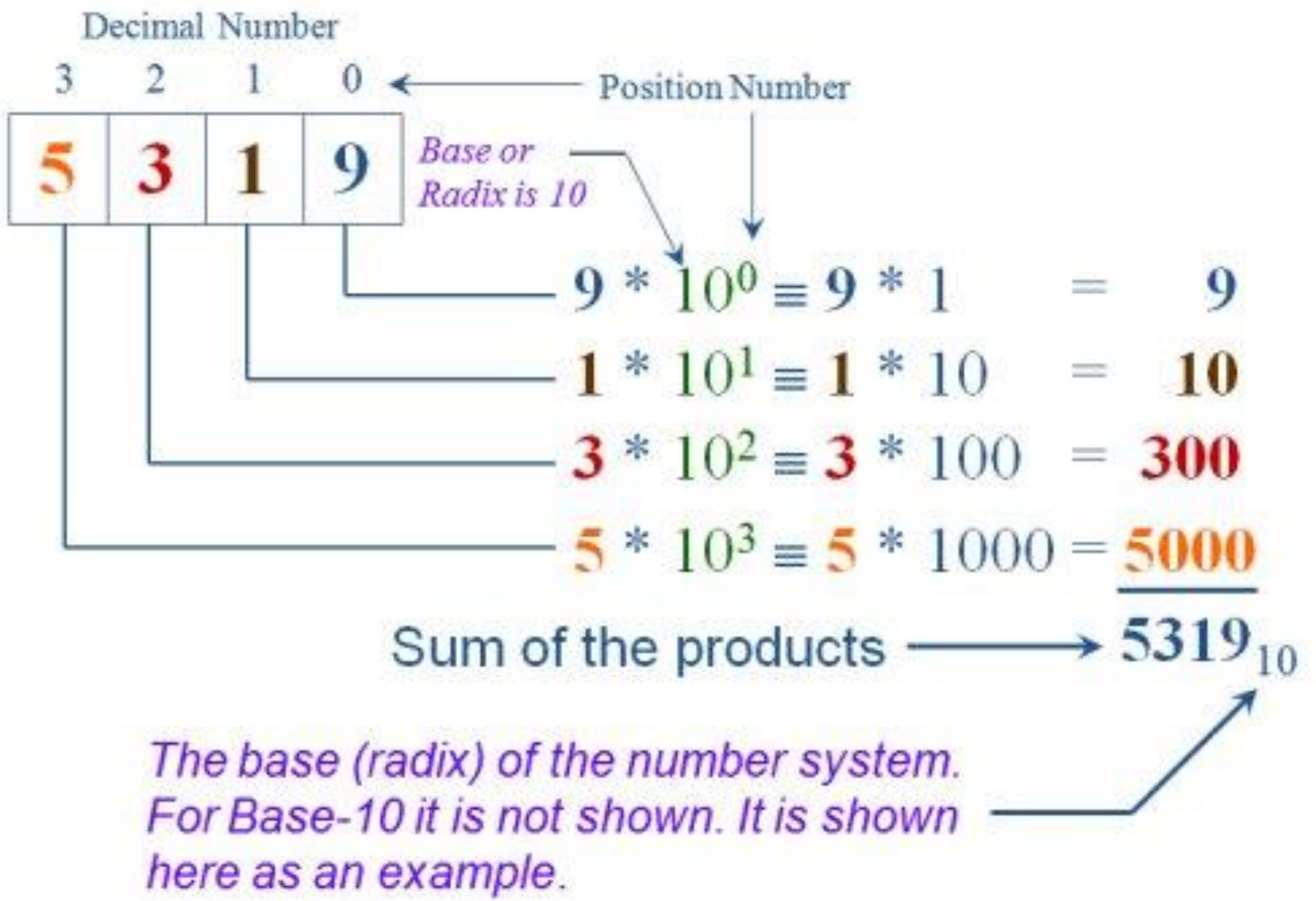
Hexadecimal to Octal Conversion

Octal to Hexadecimal Conversion

Decimal Number System:

Decimal number system has ten-digits represented by 0,1,2,3,4,5,6,7,8,9. Any decimal number can be represented by these digits and since there are ten-digits, therefore the base or radix of this number system is 10.

Decimal Number System:



Binary Number System:

What is Binary?

The binary number system is a base-2 number system. This means it only has two numbers: 0 and 1. The number system that we normally use is the decimal number system. It has 10 numbers: 0-9.

In binary number system we have two digits 0 and 1. Computer represents all kinds of data and information in binary numbers. It includes audio, graphics, video, text and numbers. The base of binary number system is 2.

Binary Number System:

CONVERSION OF A DECIMAL NUMBER INTO BINARY NUMBER

<u>DIVISION METHOD:</u>

i. Divide the number by 2.

ii. Put the reminder on right side.

iii. Keep dividing the quotient till we get the quotient less than 2.

iv. Record reminders from bottom to top to get the answer.

<u>Example:</u> $(21)_{10}$

	Number	Reminder
2	21	
2	10	1
2	5	0
2	2	1
	1	0

Answer: $(21)_{10} = (10101)_2$

Decimal to Binary Number System:

CONVERTING DECIMAL TO BINARY

- To convert a decimal integer into binary, keep dividing by 2 until the quotient is 0. Collect the remainders in *reverse* order
- To convert a fraction, keep multiplying the fractional part by 2 until it becomes 0. Collect the integer parts in forward order
- Example: 162.375: So, $(162.375)_{10} = (10100010.011)_2$

162 / 2	= 81	rem 0
81 / 2	= 40	rem 1
40 / 2	= 20	rem 0
20 / 2	= 10	rem 0
10 / 2	= 5	rem 0
5 / 2	= 2	rem 1
2 / 2	= 1	rem 0
1 / 2	= 0	rem 1

0.375 x 2 = 0.750
0.750 x 2 = 1.500
0.500 x 2 = 1.000

Binary Number System:

COUNTING IN BINARY NUMBERS

Decimal	Binary	Decimal	Binary
0	0	12	1100
1	1	13	1101
2	10	14	1110
3	11	15	1111
4	100	16	10000
5	101	17	10001
6	110	18	10010
7	111	19	10011
8	1000	20	10100
9	1001	21	10101
10	1010	22	10110
11	1011	23	10111

Binary to Decimal Number System:

Binary Digits [bits] :		1	1	0	1
Place Value :		2^3	2^2	2^1	2^0
Decimal Value :		8	4	2	1
Final value :	=	$8 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 13_{10}$			

Binary to Decimal Number System:

- In base 2 (binary): 1101.101

  The binary number 1101.101 has the following digits:

Power of 2	3	2	1	0		−1	−2	−3
Binary digit	1	1	0	1	.	1	0	1

Hence, its decimal value can be calculated as:

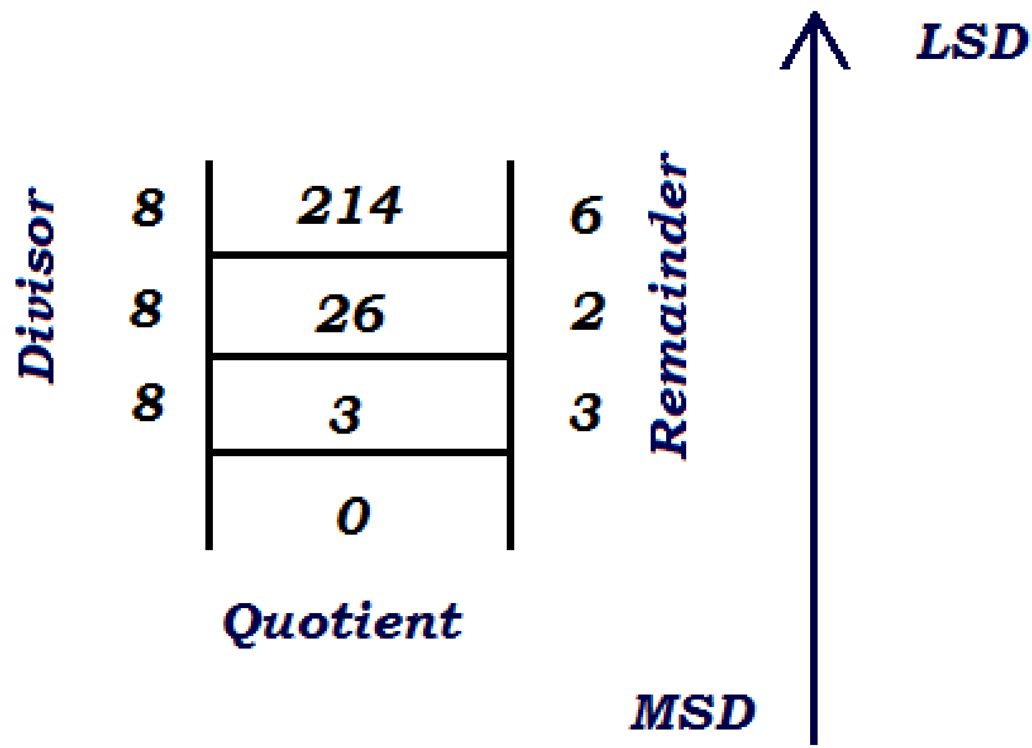
$$\begin{aligned}1101.101_2 &= 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}\\ &= 1\times 8 + 1\times 4 + 0\times 2 + 1\times 1 + 1\times 0.5 + 0\times 0.25 + 1\times 0.125\\ &= 8 + 4 + 0 + 1 + 0.5 + 0 + 0.125\\ &= 13.625_{10}\end{aligned}$$

It is now seen that 1101, which is to the left of the radix point, is the binary representation of the decimal number 13. To the right of the radix point is 101, which is the binary representation of the decimal fraction 625/1000 (or 5/8).

Octal Number System:

An octal number system has eight-digits represented as 0,1,2,3,4,5,6,7. The base of octal number system is 8.

Find the Octal equivalent for Decimal 214



MSD - most significant digit
LSD - least significant digit

Therefore, the Octal equivalent for decimal 214 is 326

Octal to Decimal Conversion

Find the equivalent decimal number for octal 143_8

Place values	8^2	8^1	8^0
Octal	1	4	3
Conversion	$1x8^2$	$4x8^1$	$3x8^0$
Decimal	64	+ 32	+ 3

99

Hexadecimal Number System:

The hexadecimal system has 16-digits, which are represented as 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. The base of hexadecimal number system is 16.

Decimal to hexadecimal

- The conversion method of decimal to hexadecimal is the same as that of decimal to binary except that the base taken is 16 instead of 2.
- For example, to convert 765.245_{10} to the hexadecimal equivalent, do the following:

Integer Part

16	765
16	47 - 13
16	2 - 15
	0 - 2

Fractional Part

0.245 x 16
3.920 x 16
14.720 x 16
11.520

$765.245_{10} = 2FD.3EB_{16}$

Find the Hex equivalent for the Decimal 3509

Divisor	Quotient	Remainder
16	3509	5
16	219	11
16	13	13
	0	

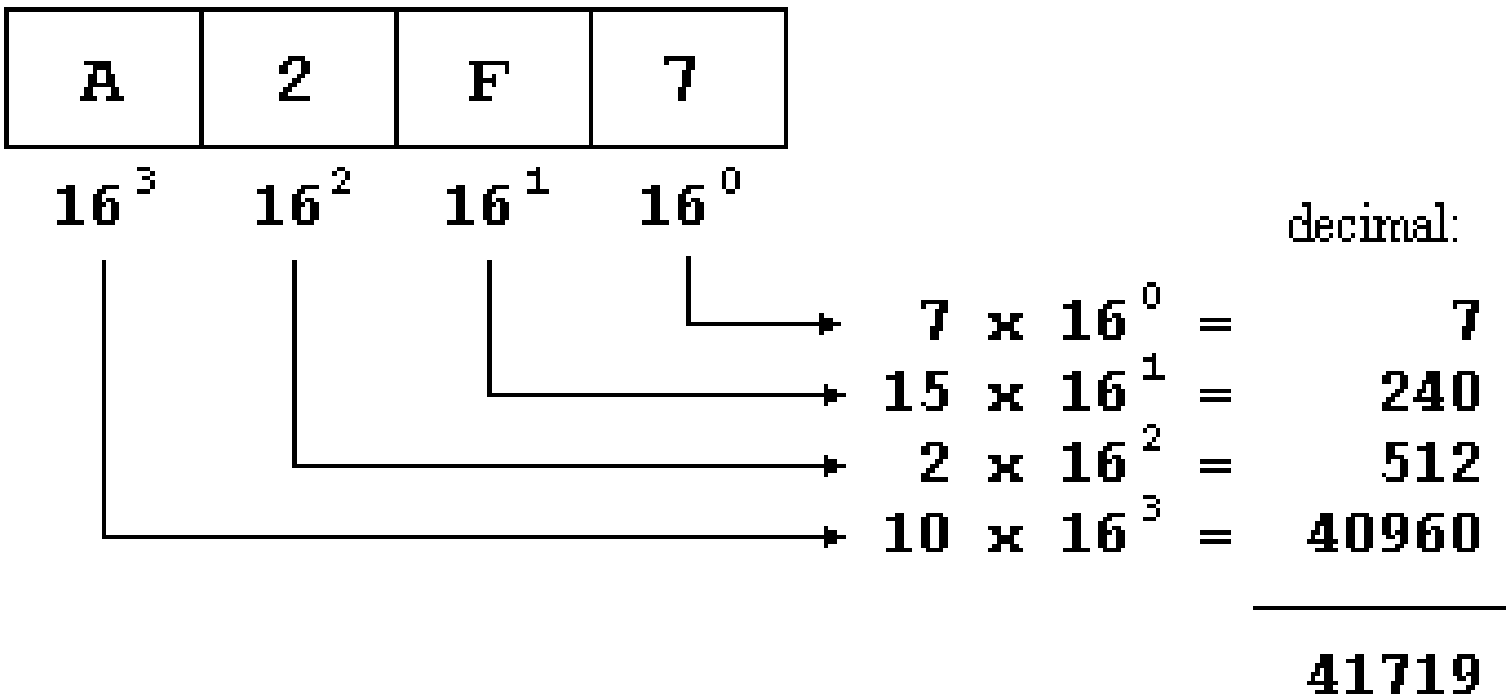
LSD

MSD

MSD - most significant digit
LSD - least significant digit

For Hex value 13 = D, 11 = B & 5 = 5
Therefore, the equivalent Hex number for decimal 3509 is DB5

Hexadecimal to Decimal Conversion



Find the Hex Equivalent for Binary 1011010

101	1010
group 2	group 1

Group 2 containing only 3 bits, so add 0 to the left

0101	1010
⇩	⇩
5	A

Binary 01010010 is equal to 5A

$01010010_2 = 5A_{16}$

Hex to Binary Number Conversion

Convert the hexadecimal $9DB5_{16}$ to its binary equivalent.

1. Separate the digits of the given hexadecimal, if more than 1 digit.

9 D B 5

2. Find the equivalent binary number for each digit of hex number, add 0's to the left if any of the binary number is shorter than 4 bits.

9	D	B	5
1001	1110	1011	0101

3. Write the all groups binary numbers together, maintaining the same group order provides the equivalent binary for the given hexadecimal.

1001111010110101

Result

$$9DB5_{16} = 1001111010110101_2$$

Octal to Binary Conversion

Convert the octal 7631_8 to its binary equivalent.

1. Separate the digits of the given octal number, if it contains more than 1 digit.

7	6	3	1

2. Find the equivalent binary number for each digit of octal number. Add 0's to the left if any of the binary equivalent is shorter than 3 bits.

7	6	3	1
111	110	011	001

3. Write the all group's binary numbers together, maintaining the same group order provides the equivalent binary for the given octal number.

111110011001

Result

$$7631_8 = 111110011001_2$$

Binary to Octal Conversion

Convert the binary number 111110011001_2 to its octal equivalent.

1. Separate the digits of a given binary number into groups from right to left side, each containing 4 bits.

   111 110 011 001

2. Find the equivalent octal number for each group.

111	110	011	001
7	6	3	1

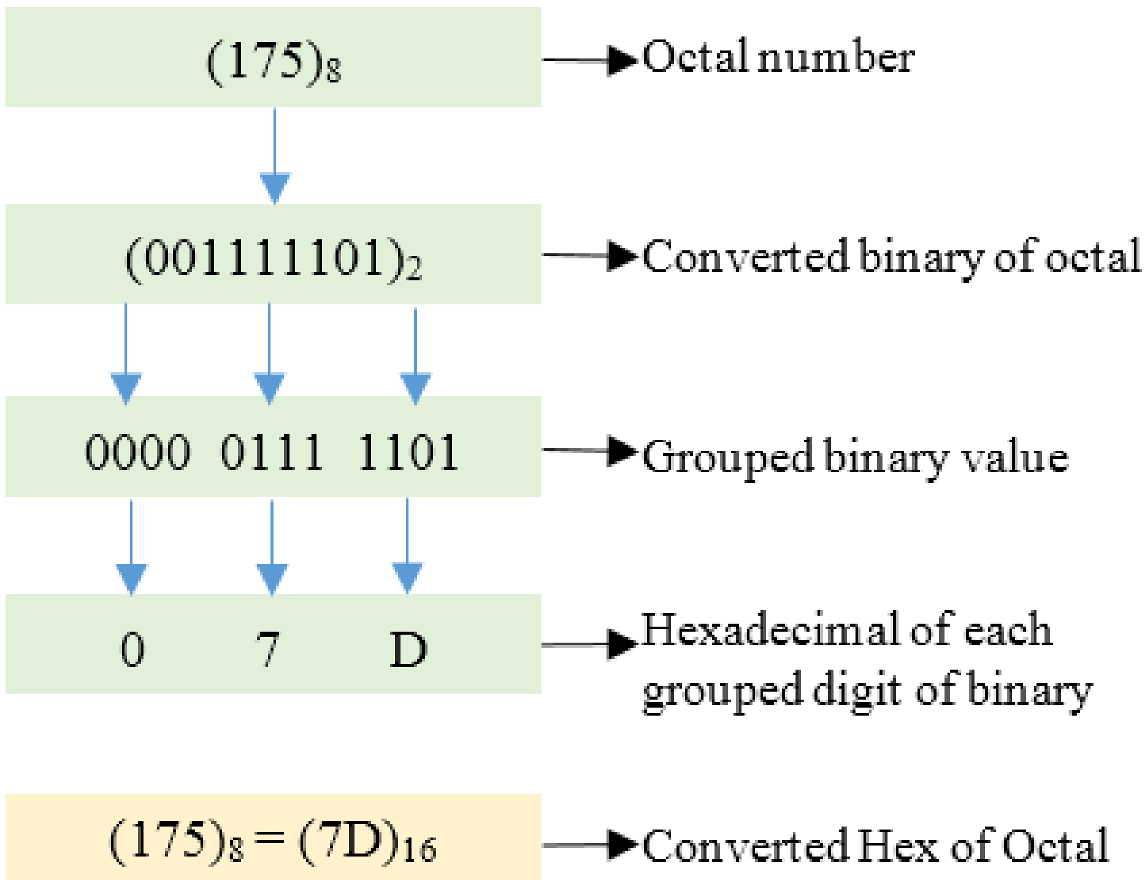
3. Write the all group's octal numbers together, maintaining the group order provides the equivalent octal number for the given binary.

   7631

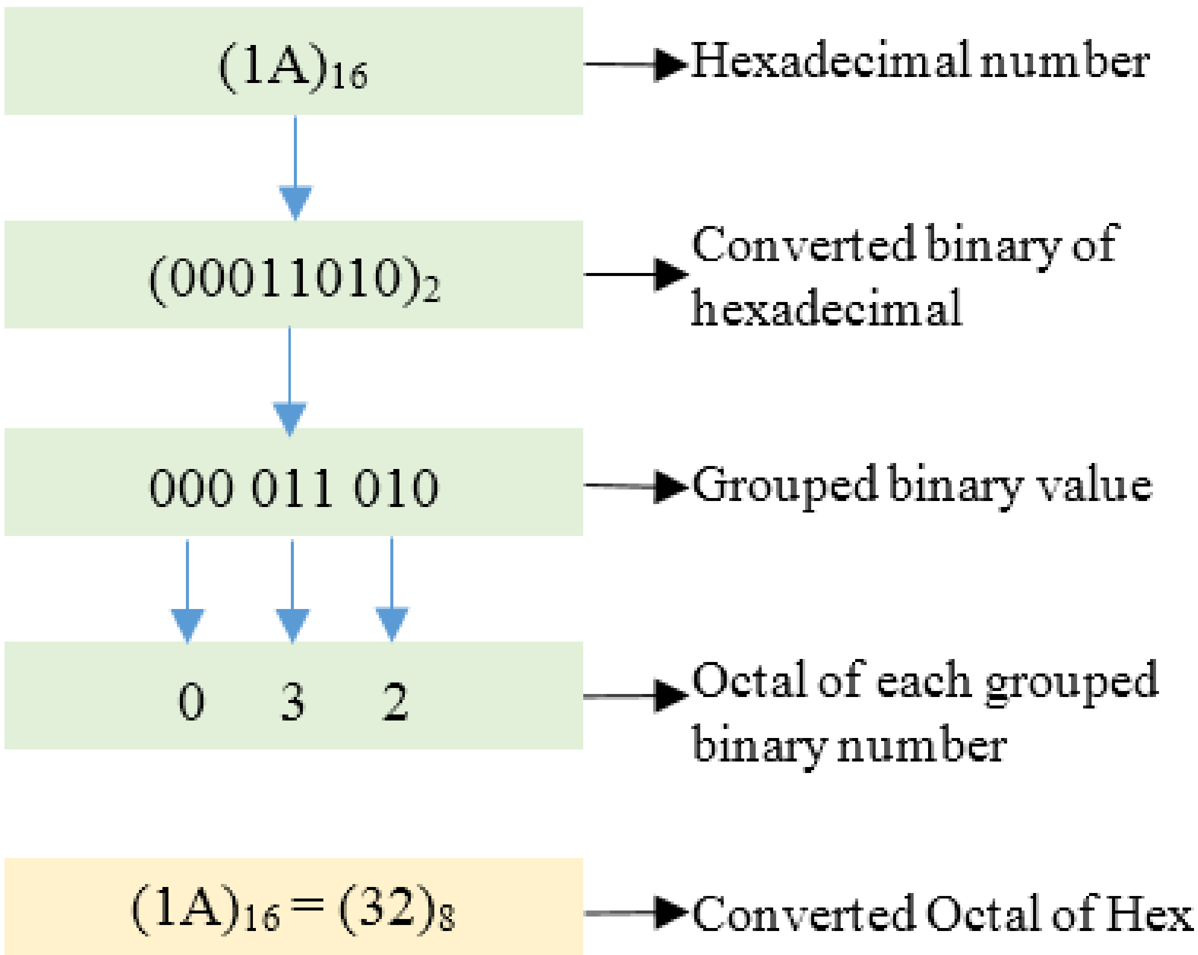
Result

$$111110011001_2 = 7631_8$$

Octal to Hexadecimal Conversion



Hexadecimal to Octal Conversion



One's Complement

Invert all bits. Each 1 becomes a 0, and each 0 becomes a 1.

Original Value	One's Complement
0	1
1	0
1010	0101
1111	0000
11110000	00001111
10100011	01011100
11110000 10100101	00001111 01011010

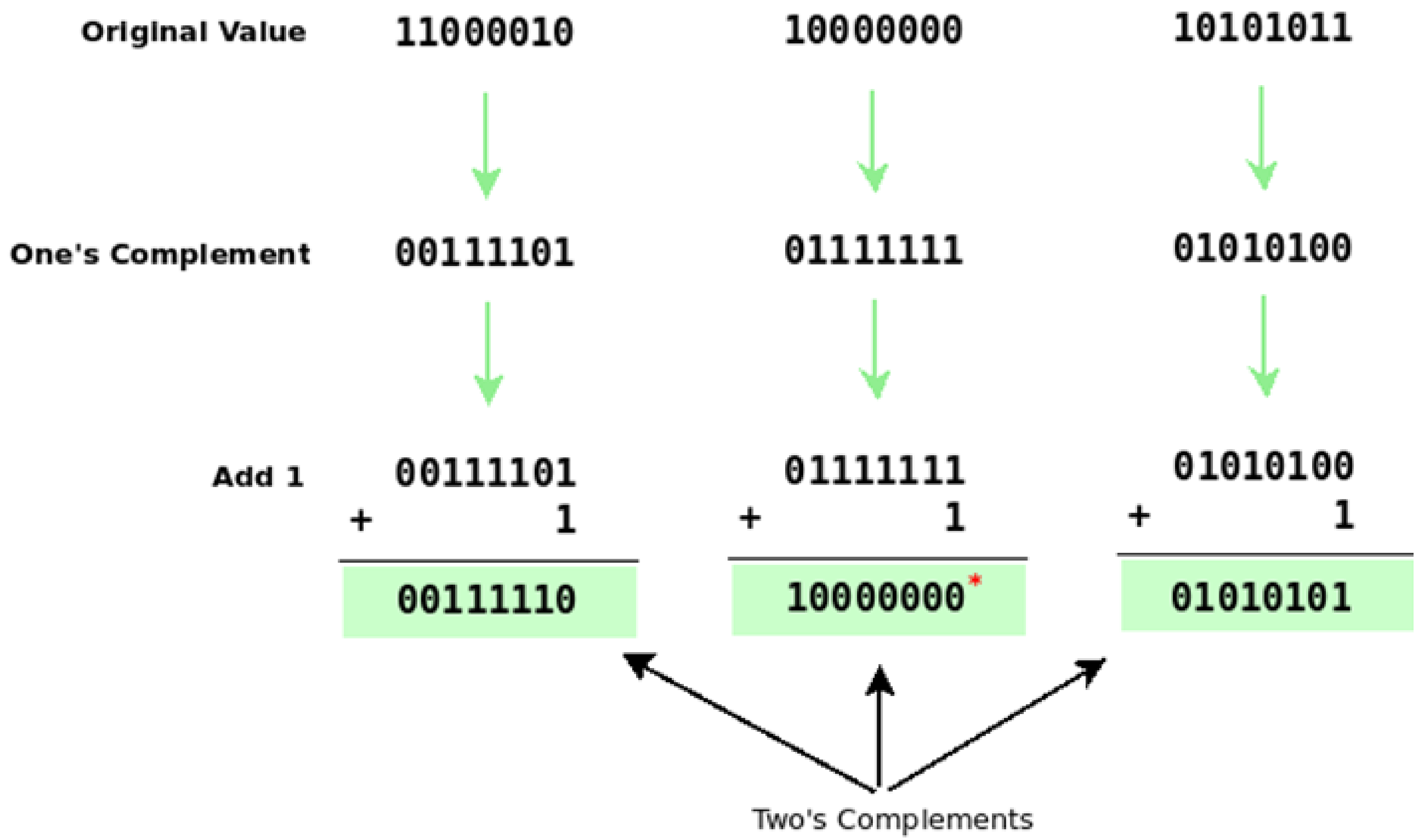
For example: Let's find out the 2's complement of given 8-bit number

$$
\begin{array}{rl}
 & 00101001 \\
\\
 & 11010110 \quad \text{Invert the bits} \\
+ & 00000001 \quad \text{Then, add 1} \\
\hline
 & 11010111
\end{array}
$$

The 2's complement of 00101001 is **11010111**

Two's Complement

First, find the one's complement of a value, and then add 1 to it.



2's Complement Examples

Example #1

5 = 00000101

↓↓↓↓↓↓↓↓

11111010

+1

-5 = 11111011

Complement Digits

Add 1

Example #2

-13 = 11110011

↓↓↓↓↓↓↓↓

00001100

+1

13 = 00001101

Complement Digits

Add 1

Subtraction using 2's complement

a) Subtraction of smaller number from a larger number:-

1. Calculate the 2's complement of a smaller number.
2. add the 2's complement to the larger number.
3. If carry comes in the MSB , discard the carry .

$$(+7)_{10} = 0000\ 0111$$
$$(-3)_{10} = 1111\ 1101$$

$$(+4)_{10} = 0000\ 0100$$

Again, $(+4)_{10} = 0000\ 0100$
$$(-3)_{10} = 1111\ 1101$$

$(+1)_{10} = 0000\ 0001$ = remainder

Binary Addition

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

0011010 + 001100 = 00100110

	11	carry
	0011010	$= 26_{10}$
+	0001100	$= 12_{10}$
	0100110	$= 38_{10}$

Binary Addition

	Binary	*Decimal*
A	**110011**	**51**
B	**1101**	**13**
A + B	**1000000**	**64**

Binary Subtraction

A	B	Sub	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- 1011011 − 10010 = 1001001

100010110 − 1111010 = 10011100

One’s complement addition

- To add one’s complement numbers:
  - First do unsigned addition on the numbers, *including* the sign bits.
  - Then take the carry out and add it to the sum.
- Two examples:

$$
\begin{array}{rr@{\qquad}r}
 & 0111 & (+7) \\
+ & 1011 & +\ (-4) \\
\hline
1 & 0010 & \\
\\
 & 0010 & \\
+ & 1 & \\
\hline
 & 0011 & (+3)
\end{array}
\qquad\qquad
\begin{array}{rr@{\qquad}r}
 & 0011 & (+3) \\
+ & 0010 & +\ (+2) \\
\hline
0 & 0101 & \\
\\
 & 0101 & \\
+ & 0 & \\
\hline
 & 0101 & (+5)
\end{array}
$$

- This is simpler and more uniform than signed magnitude addition.

Numerical

Represent the decimal number 27 in binary form using

(i) Binary Code
(ii) BCD code
(iii) Excess-3 code
(iv) Gray code
(v) Octal code
(vi) Hexadecimal code

Solution

(i) The decimal number is converted into straight binary form. Its value is 11011
(ii) Each digit of the decimal number is coded using 4-bit BCD code as given below

0010 0111

(iii) Each digit of the decimal number is coded using 4-bit Excess-3 code as given below

0101 1010

(iv) 5 bits are required to represent 27, therefore, 5-bit Gray code is constructed and 27 is represented as 10110.
(v) $(27)_{10} = (33)_8 = 011\ 011$
(vi) $(27)_{10} = (1B)_{16} = 0001\ 1011$

Numerical

Represent the decimal numbers (a) 396 and (b) 4096 in binary form in

(i) Binary code (straight binary)
(ii) BCD code
(iii) Excess-3 code
(iv) Octal code
(v) Hex code

Solution

(a) (i) $396 = 110001100$
(ii) $396 = 001110010110$
(iii) $396 = 011011001001$
(iv) $396 = (614)_8 = 110001100$
(v) $396 = (18C)_{16} = 000110001100$

(b) (i) $4096 = 1000000000000$
(ii) $4096 = 0100000010010110$
(iii) $4096 = 01110011\ 11001001$
(iv) $4096 = (10000)_8 = 001\ 000\ 000\ 000\ 000$
(v) $4096 = (1000)_{16} = 0001\ 0000\ 0000\ 0000$

Octal Arithmetic

Add $(23)_8$ and $(67)_8$.

Solution

$$
\begin{array}{rr}
23 = & 010011 \\
(+)67 = & 110111 \\
\hline
(112)_8 = & 1001\ 010
\end{array}
$$

Octal Arithmetic

Subtract (a) $(37)_8$ from $(53)_8$
(b) $(75)_8$ from $(26)_8$

Solution

Using 8-bit representation,

(a)

$$
\begin{array}{rcrl}
(53)_8 & = & 00101011 & \\
-\,(37)_8 & = & (+)\ 11100001 & \text{Two's complement of } (37)_8 \\
\hline
(14)_8 & = & 100001100 &
\end{array}
$$

Discard carry

(b)

$$
\begin{array}{rcrl}
(26)_8 & = & 00010110 & \\
-\,(75)_8 & = & (+)\ 11000011 & \text{Two's complement of } (75)_8 \\
\hline
-\,(47)_8 & & 11011001 & \text{Two's complement of result}
\end{array}
$$

Two's complement of 11011001 = 00 100 111 = $(47)_8$

Hexadecimal Arithmetic

Solution

$$
\begin{array}{rcr}
7F = & & 01111111 \\
\underline{(+)\ BA} = & & \underline{10111010} \\
(139)_{16} = & & 100111001
\end{array}
$$

Hexadecimal Arithmetic

Subtract (a) $(5C)_{16}$ from $(3F)_{16}$
(b) $(7A)_{16}$ from $(C0)_{16}$

Solution

(a)

$$\begin{array}{rrl} 3F = & 00111111 & \\ -5C = & (+)10100100 & \text{Two's complement of } (5C)_{16} \\ \hline -1D = & 11100011 & \text{Two's complement of result} \end{array}$$

Two's complement of 11100011 = 0001 1101 = $(1D)_{16}$

(b)

$$\begin{array}{rrl} C0 = & 11000000 & \\ -7A = & (+)\ 10000110 & \text{Two's complement of } (7A)_{16} \\ \hline 46 = & 101000110 & \end{array}$$

Discard carry

8's Complement

r's complement = (r-1)'s complement + 1

i) 3675

$$
\begin{array}{lr}
\text{7's complement} = & 7777 \\
 & -\ 3675 \\
\hline
 & 4102 \\
 & +\ 1 \\
\hline
\text{8's complement} = & 4103 \\
\hline
\end{array}
$$

ii) 2057.34

$$
\begin{array}{lr}
\text{7's complement} = & 7777.77 \\
 & -\ 2057.34 \\
\hline
 & 5720.43 \\
 & +\ 1 \\
\hline
\text{8's complement} = & 5720.44 \\
\hline
\end{array}
$$

16's Complement

i) 7BA

15's complement =	FFF
	− 7BA
	845
	+ 1
16's complement =	846

ii) F20.3AE

15's complement =	FFF.FFF
	− F20.3AE
	0DF.C51
	+ 1
16's complement =	DF.C52

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

r's complement = (r−1)'s complement + 1

Octal Subtraction using 7's Complement Method

Step 1: Find 7's complement of subtrahend

Step 2: Add first number (minuend) and 7's complement of subtrahend

Step 3: If carry is generated, add the carry with LSB of the sum. If there is no carry,take 7's complement for the sum and assign negative sign.

Case 1: Carry is produced

Ex 1: $(412)_8 - (263)_8 = (\,?\,)$

https://www.youtube.com/watch?v=l7BXPLv8Kd4

Octal Subtraction using 7's Complement Method

Step 1: Find 7's complement of subtrahend

Step 2: Add first number (minuend) and 7's complement of subtrahend

Step 3: If carry is generated, add the carry with LSB of the sum. If there is no carry, take 7's complement for the sum and assign negative sign.

Case 1: Carry is produced

Ex 1 : $(4\,1\,2)_8 - (2\,6\,3)_8 = (\,?)$

x y

1)
$$\begin{array}{r} 7\,7\,7 \\ 2\,6\,3 \; (-) \\ \hline \underline{5\,1\,4} \end{array} \leftarrow 7\text{'s}$$

2)
$$\begin{array}{r} 4\,1\,2 \\ 5\,1\,4 \; (+) \\ \hline \boxed{1}\,1\,2\,\underline{6} \end{array}$$

carry

3)
$$\begin{array}{r} 1\,2\,6 \\ 1 \\ \hline \underline{1\,2\,7} \end{array}$$

$\boxed{\text{Ans: } 127_8}$

https://www.youtube.com/watch?v=I7BXPLv8Kd4

Octal Subtraction using 7's Complement Method

Case 2 : No carry is produced

Ex 2: $(2\,6\,3)_8 - (4\,1\,2)_8 = (?)$

x y

1)
$$\begin{array}{r} 7\;7\;7 \\ 4\;1\;2^{(-)} \\ \hline 3\;6\;5 \end{array} \leftarrow 7\text{'s}$$

2)
$$\begin{array}{r} {}^{1}\;{}^{1}\;\; \\ 2\;6\;3 \\ 3\;6\;5^{(+)} \\ \hline 6\;5\;0 \end{array} \leftarrow \text{sum}$$

3)
$$\begin{array}{r} 7\;7\;7 \\ 6\;5\;0^{(-)} \\ \hline -1\;2\;7 \end{array}$$

$8\,\underline{|8}$ $8\,\underline{|13}$

1-0   1-5

$\rightarrow \begin{array}{cc} 1 & 0 \\ \hline c & s \end{array}$ $\begin{array}{cc} 1 & 5 \\ \hline c & s \end{array}$

$\boxed{\text{ANS}: -127_8}$

https://www.youtube.com/watch?v=I7BXPLv8Kd4

Octal Subtraction using 8's Complement Method

Step 1: Find 8's complement of subtrahend

Step 2: Add first number (minuend) and 8's complement of subtrahend

Step 3: If carry is produced in the addition, discard the carry. If there is no carry, take 8's complement of the sum and assign negative sign.

Case 1: Carry is produced

Ex : $(3\,7\,2)_8 - (1\,4\,4)_8 = (\,?\,)_8$

https://www.youtube.com/watch?v=0Bbz5yvZG5s

Octal Subtraction using 8's Complement Method

Step 1: Find 8's complement of subtrahend

Step 2: Add first number (minuend) and 8's complement of subtrahend

Step 3: If carry is produced in the addition, discard the carry. If there is no carry, take 8's complement of the sum and assign negative sign.

Case 1: Carry is produced

Ex : $(3\,7\,2)_8 - (1\,4\,4)_8 = (\,?\,)_8$

x y

1)
$$\begin{array}{r} 7\,7\,7 \\ 1\,4\,4\ (-) \\ \hline 6\,3\,3 \leftarrow 7\text{'s} \\ 1 \\ \hline 6\,3\,4 \leftarrow 8\text{'s} \end{array}$$

2)
$$\begin{array}{r} ^{1}\quad \\ 3\,7\,2 \\ 6\,3\,4\ (+) \\ \hline \boxed{\cancel{1}}\,2\,2\,6 \\ \hline \end{array}$$

$$\boxed{226_8}$$

https://www.youtube.com/watch?v=0Bbz5yvZG5s

Octal Subtraction using 8's Complement Method

Case 2: No carry is produced

Ex : $(144)_8 - (372)_8 = (?)_8$

X Y

1)

$$\begin{array}{r} 777 \\ 372 \; (-) \\ \hline 405 \leftarrow 7's \\ 1 \; (+) \\ \hline 406 \leftarrow 8's \\ \hline \end{array}$$

2)

$$\begin{array}{r} 1 \quad \\ 144 \\ 406 \\ \hline 552 \leftarrow sum \\ \hline \end{array}$$

3)

$$\begin{array}{r} 777 \\ 552 \; (-) \\ \hline 225 \leftarrow 7's \\ 1 \\ \hline -226 \leftarrow 8's \\ \hline \end{array}$$

$$\boxed{-226_8}$$

https://www.youtube.com/watch?v=0Bbz5yvZG5s

Hexadecimal Subtraction using 15's Complement Method

Step 1: Find 15's complement of subtrahend

Step 2: Add first number(minuend) and 15's complement of subtrahend

Step 3: If carry is generated,add the carry with LSB of the sum.If there is no carry ,take 15's complement for the sum and assign negative sign

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Case 1: Carry is produced

EX 1: $(B01)_{16} - (98F)_{16}$ ----> $(?)_{16}$

https://www.youtube.com/watch?v=On7YB46rcl0

Hexadecimal Subtraction using 15's Complement Method

Step 1: Find 15's complement of subtrahend

Step 2: Add first number(minuend) and 15's complement of subtrahend

Step 3: If carry is generated,add the carry with LSB of the sum.If there is no carry ,take 15's complement for the sum and assign negative sign

Case 1: Carry is produced

EX 1: $(B\,0\,1)_{16} - (9\,8\,F)_{16}$ ----> $(\,?\,)_{16}$

X Y

1)
$$\begin{array}{r} 15\;15\;15 \\ 9\;\;8\;\;F \;(-) \\ \hline 6\;\;7\;\;0 \end{array} \leftarrow 15's$$

2)
$$\begin{array}{r} B\;0\;1 \\ 6\;7\;0 \;(+) \\ \hline \boxed{1}\,1\;7\;1 \\ 1 \;(+) \\ \hline 1\;7\;2_{16} \end{array} \leftarrow$$

https://www.youtube.com/watch?v=On7YB46rcl0

Hexadecimal Subtraction using 15's Complement Method

Case 2 : No carry is produced

EX 2 : $(6\ 9\ A)_{16} - (C\ 1\ 3)_{16} = (\ ?\ )_{16}$

$X \qquad Y$

1)

$$\begin{array}{r} 15\ 15\ 15 \\ C\ \ 1\ \ 3 \ (-) \\ \hline 3\ \ E\ \ C \end{array} \leftarrow 15\text{'s}$$

2)

$$\begin{array}{r} {}^{1}6\ {}^{1}9\ A \\ 3\ E\ C \ (+) \\ \hline A\ 8\ 6 \end{array}$$

3)

$$\begin{array}{r} 15\ 15\ 15 \\ A\ \ 8\ \ 6 \ (-) \\ \hline -\ 5\ \ 7\ \ 9 \end{array}$$

$$\boxed{-579_{16}}$$

https://www.youtube.com/watch?v=On7YB46rcl0

Hexadecimal Subtraction using 16's Complement Method

- Step 1: Find 16's complement of subtrahend
- Step 2: Add first number(minuend) and 16's complement of subtrahend
- Step 3: If carry is produced, discard the carry ,Otherwise, take 16's complement for the sum and assign negative sign.

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Case 1: Carry is produced

EX 1 : $(C\,B\,1)_{16} - (9\,7\,1)_{16} = (\,?\,)_{16}$

X Y

https://www.youtube.com/watch?v=mwy3Osz9xsc

Hexadecimal Subtraction using 16's Complement Method

Step 1: Find 16's complement of subtrahend

Step 2: Add first number(minuend) and 16's complement of subtrahend

Step 3: If carry is produced, discard the carry ,Otherwise, take 16's complement for the sum and assign negative sign.

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Case 1: Carry is produced

EX 1 : $(CB1)_{16} - (971)_{16} = (?)_{16}$

X Y

1)
$$\begin{array}{r} 15\,15\,15 \\ 9\;7\;1\;(-) \\ \hline 6\;8\;E \leftarrow 15's \\ 1\;(+) \\ \hline 6\;8\;F \leftarrow 16's \end{array}$$

2)
$$\begin{array}{r} C\;B\;1 \\ 6\;8\;F\;(+) \\ \hline \cancel{\boxed{1}}\,3\;4\;0 \end{array}$$

$\rightarrow 340_{16}$

https://www.youtube.com/watch?v=mwy3Osz9xsc

Hexadecimal Subtraction using 16's Complement Method

Case 2 : No carry is produced

EX 2: $(9\,7\,1)_{16} - (C\,B\,1)_{16} = (?)_{16}$

X Y

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

1)
$$\begin{array}{r l} 15\ 15\ 15 & \\ C\ \ B\ \ 1 & (-) \\ \hline 3\ \ 4\ \ E & \leftarrow 15\text{'s} \\ 1 & (+) \\ \hline 3\ \ 4\ \ F & \leftarrow 16\text{'s} \\ \hline \end{array}$$

2)
$$\begin{array}{r} {}^{1} \\ 9\ \ 7\ \ 1 \\ 3\ \ 4\ \ F \\ \hline -\ C\ \ C\ \ 0 \\ \hline \end{array}$$

3)
$$\begin{array}{r l} 15\ 15\ 15 & \\ C\ \ C\ \ 0 & (-) \\ \hline 3\ \ 3\ \ F & \leftarrow 15\text{'s} \\ 1 & (+) \\ \hline -\ 3\ \ 4\ \ 0 & \leftarrow 16\text{'s} \\ \hline \end{array}$$

$$\boxed{-340_{16}}$$

https://www.youtube.com/watch?v=mwy3Osz9xsc

Numericals for practice

Convert hexadecimal value 16 to decimal.

Convert the following decimal number 187 to 8-bit binary.

Convert binary 111111110010 to hexadecimal.

Convert the following binary 101010 number to decimal.

Convert the binary number 1001.0010_2 to decimal.

Convert $8B3F_{16}$ to binary

Convert decimal 64 to binary.

Convert hexadecimal value C1 to binary.

Convert the following octal number 17 to decimal.

Numericals for practice

Convert the binary number 010111100 to octal.

What is binary number for octal 45_8?

The sum of 11101 + 10111 equals

Convert the binary number 10011010 to decimal.

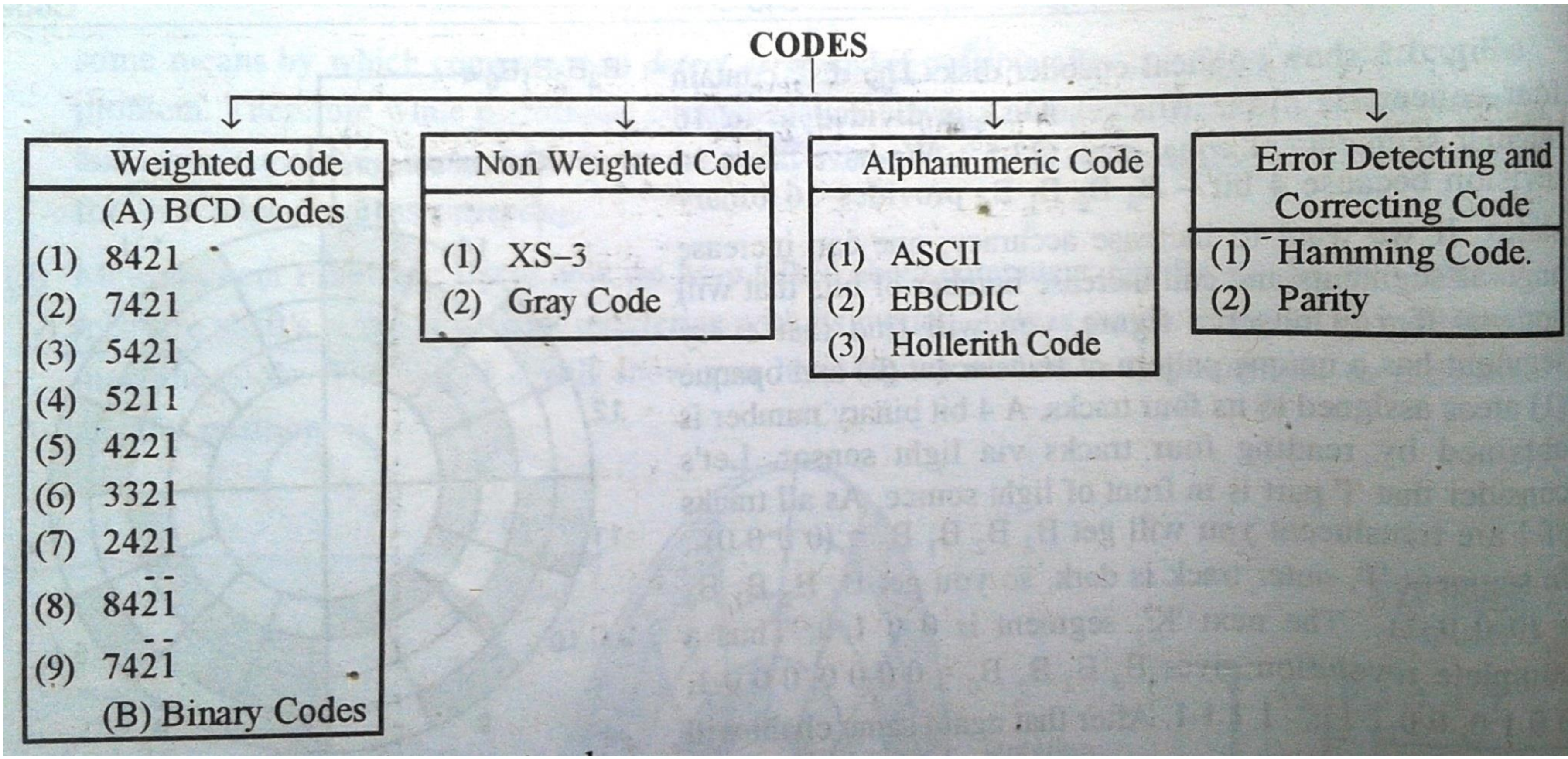
The decimal number 188 is equal to the binary number?

Convert the following binary number 001101011 to octal

Convert decimal 213 to binary

The decimal number for octal 74_8 is?

Codes- Classification



Weighted Code:-

In weighted code, each digit position has a weight or value. The sum of all digits multiplied by a weight gives the total amount being represented.

We can express any decimal number in tens, hundreds, thousands and so on.

Eg:- Decimal number 4327 can be written as

$$4327 = 4000+300+20+7$$

In the power of 10, it becomes

$$4327 = 4(10^3)+3(10^2)+2(10^1)+7(10^0)$$

BCD or 8421 is a type of weighted code where each digit position is being assigned a specific weight.

BCD Code (Binary Coded Decimal)

Decimal digit	BCD Code		
	8 4 2 1	4 2 2 1	5 4 2 1
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 0 0	1 0 0 0	0 1 0 0
5	0 1 0 1	0 0 1 1	1 0 0 1
6	0 1 1 0	1 1 0 0	1 0 1 0
7	0 1 1 1	1 1 0 1	1 0 1 1
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0

Decimal	8 4 2 1 2nd digit	8 4 2 1 1st digit	B C D
11	0 0 0 1	0 0 0 1	
12	0 0 0 1	0 0 1 0	
13	0 0 0 1	0 0 1 1	
14	0 0 0 1	0 1 0 0	
:	:	:	
19	0 0 0 1	1 0 0 1	
20	0 0 1 0	0 0 0 0	
21	0 0 1 0	0 0 0 1	
22	0 0 1 0	0 0 1 0	
23	0 0 1 0	0 0 1 1	
24	0 0 1 0	0 0 1 1	
:	:	:	
29	0 0 1 0	1 0 0 1	
30	0 0 1 1	0 0 0 0	
31	0 0 1 1	0 0 0 1	
:	:	:	

BCD Code (Binary Coded Decimal)

Decimal	8 4 2 1	7 4 2 1	5 4 2 1	5 2 1 1	4 2 2 1	3 3 2 1	2 4 2 1	$8\ 4\ \bar{2}\ \bar{1}$	$7\ 4\ \bar{2}\ \bar{1}$
0	0000	0000	0000	0000	0000	0000	0000	0000	0000
1	0001	0001	0001	0001	0001	0001	0001	0111	0111
2	0010	0010	0010	0011	0010	0010	0010	0110	0110
3	0011	0011	0011	0101	0011	0011	0011	0101	0101
4	0100	0100	0100	0111	1000	0101	0100	0100	0100
5	0101	0101	1000	1000	0111	1010	1011	1011	1010
6	0110	0110	1001	1010	1100	1100	1100	1010	1001
7	0111	1000	1010	1100	1101	1101	1101	1001	1000
8	1000	1001	1011	1110	1110	1110	1110	1000	1111
9	1001	1010	1100	1111	1111	1111	1111	1111	1110

Non-weighted code:-

In non-weighted code, there is no positional weight i.e. each position within the binary number is not assigned a prefixed value. No specific weights are assigned to bit position in non –weighted code.

The non-weighted codes are:-

a) The Gray code

b) The Excess-3 code

b(1)	b(2)	b(3)	b(4)	b(5)	
1	1	1	0	1	binary
1	0	0	1	1	gray
g(1)	g(2)	g(3)	g(4)	g(5)	
b(1)	b(1) xor b(2)	b(2) xor b(3)	b(3) xor (4)	b(4) xor b(5)	

The Gray code:-

It is non weighted code in which each number differs from previous number by a single bit.

Decimal	Binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

Excess 3 code

- The Excess-3 code is also called as XS-3 code.
- It is non-weighted code used to express decimal numbers.
- The Excess-3 code words are derived from the 8421 BCD code words adding 3 to it.

Decimal	BCD 8 4 2 1	Excess-3 BCD + 0011
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

Decimal					XS-3 Code				
					2^3	2^2	2^1	2^0	
0					0	0	1	1	0 + 3 = 3
1					0	1	0	0	1 + 3 = 4
2					0	1	0	1	2 + 3 = 5
3					0	1	1	0	3 + 3 = 6
4					0	1	1	1	4 + 3 = 7
5					1	0	0	0	5 + 3 = 8
6					1	0	0	1	6 + 3 = 9
7					1	0	1	0	7 + 3 = 10
8					1	0	1	1	8 + 3 = 11
9					1	1	0	0	9 + 3 = 12
10	0	1	0	0	0	0	1	1	(1 + 3)(0 + 3) = 4/3
11	0	1	0	0	0	1	0	0	(1 + 3)(1 + 3) = 4/4
12	0	1	0	0	0	1	0	1	(1 + 3)(2 + 3) = 4/5
13	0	1	0	0	0	1	1	0	(1 + 3)(3 + 3) = 4/6
⋮		⋮				⋮			⋮
19	0	1	0	0	1	1	0	0	(1 + 3)(9 + 3) = 4/12
20	0	1	0	1	0	0	1	1	(2 + 3)(0 + 3) = 5/3
21	0	1	0	1	0	1	0	0	
⋮		⋮				⋮			⋮
29	0	1	0	1	1	1	0	0	

Table 2.8 ***Various Binary Codes***

Decimal Number	Binary				BCD				Excess-3				Gray			
	B_3	B_2	B_1	B_0	D	C	B	A	E_3	E_2	E_1	E_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	1
3	0	0	1	1	0	0	1	1	0	1	1	0	0	0	1	0
4	0	1	0	0	0	1	0	0	0	1	1	1	0	1	1	0
5	0	1	0	1	0	1	0	1	1	0	0	0	0	1	1	1
6	0	1	1	0	0	1	1	0	1	0	0	1	0	1	0	1
7	0	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0
8	1	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0
9	1	0	0	1	1	0	0	1	1	1	0	0	1	1	0	1
10	1	0	1	0									1	1	1	1
11	1	0	1	1									1	1	1	0
12	1	1	0	0									1	0	1	0
13	1	1	0	1									1	0	1	1
14	1	1	1	0									1	0	0	1
15	1	1	1	1									1	0	0	0

Alphanumeric Code – ASCII Code

Character	ASCII code	Binary code
null character	0	0 0 0 0 0 0 0
a	97	1 1 0 0 0 0 1
b	98	1 1 0 0 0 1 0
c	99	1 1 0 0 0 1 1
A	65	1 0 0 0 0 0 1
B	66	1 0 0 0 0 1 0
C	67	1 0 0 0 0 1 1
%	37	0 1 0 0 1 0 1
+	43	0 1 0 1 0 1 1
0	48	0 1 1 0 0 0 0
1	49	0 1 1 0 0 0 1
Delete	127	1 1 1 1 1 1 1

Alphanumeric Code – EBCDIC Code

Special characters	EBCDIC
<	01001011
(	01001100
+	01001101
/	01001110
&	01010000
:	01111011
#	01111011
@	01111100
‘	01111101
=	01111110
“	01111111
,	01101011
%	01101100
-	01101101
>	01101110

Contd..

Alphabetic	EBCDIC
A	11000001
B	11000010
C	11000011
D	11000100
E	11000101
F	11000110
G	11000111
H	11001000
I	11001001
J	11010001
K	11010010
L	11010011
M	11010100
N	11010101
O	11010110
P	11010111

Error Detecting Code – Parity Code

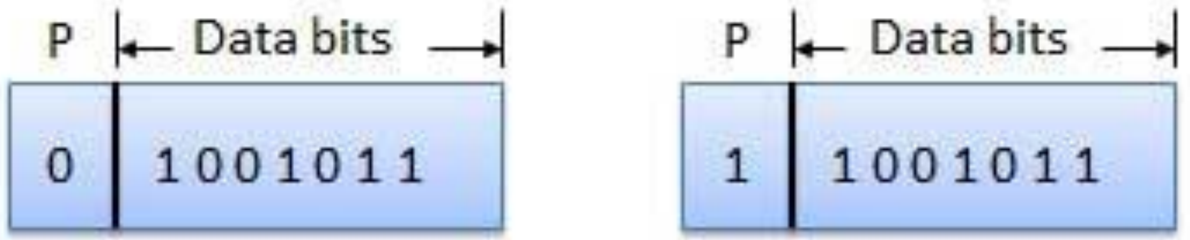


Fig. (a)

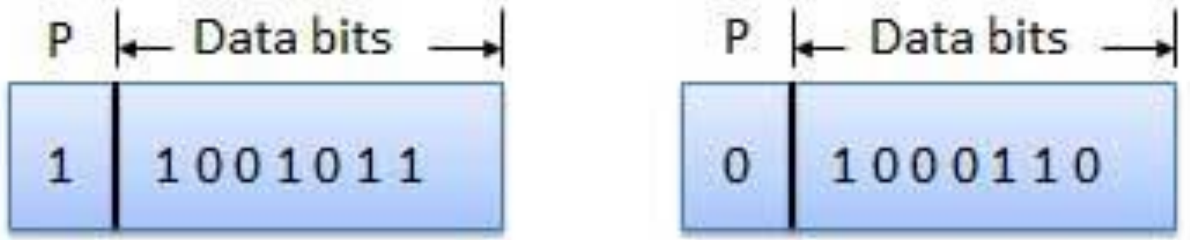
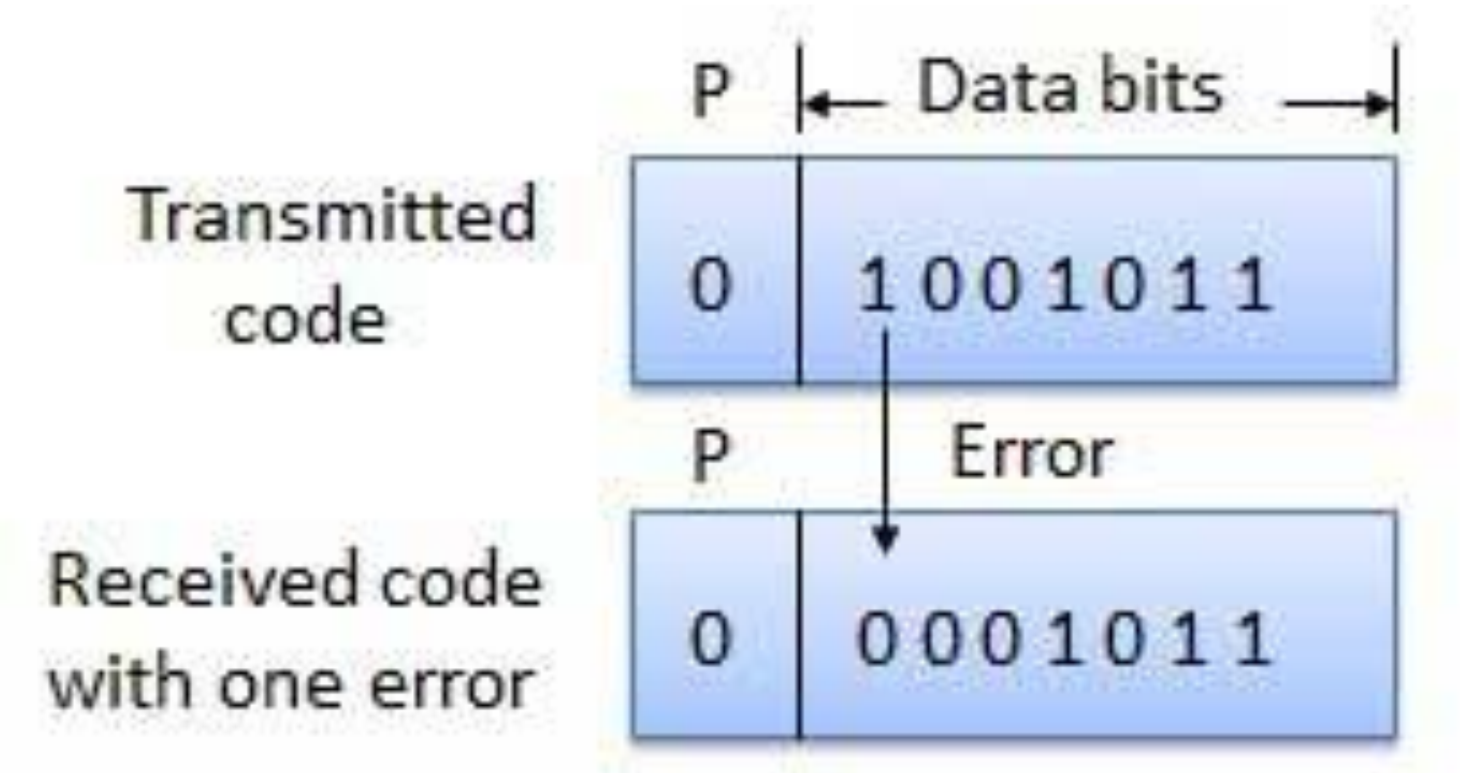


Fig. (b)

Error Detecting Code – Parity Code



Limitation:

1. **Suitable for detection of only one error but not for multiple error**
2. **It can't correct the error**

Error Correcting Code – Hamming Code

code word 1: 1 1 0 1 0 1 0 0

code word 2: 0 1 0 1 1 1 1 0

Hamming Distance – 3

Hamming Weight - code word 1 – 4

Hamming Weight - code word 1 – 5

Error Correcting Code – Hamming Code

7	6	5	4	3	2	1	How to calculate parity bits
D_7	D_6	D_5	P_4	D_3	P_2	P_1	This represents the full codeword
D_7	D_6	D_5	P_4				P_4 – Even parity of D_7 D_6 D_5
D_7	D_6			D_3	P_2		P_2 – Even parity of D_7 D_6 D_3
D_7		D_5		D_3		P_1	P_1 – Even parity of D_7 D_5 D_3

7 bit Hamming Code Format

Error Correcting Code – Hamming Code

Generate 7 bit hamming code for given data bits 1011

7	6	5	4	3	2	1
D_7	D_6	D_5	P_4	D_3	P_2	P_1

P1 bit = Even parity of D7 D5 D3 = 1 1 1 1

P2 bit = Even parity of D7 D6 D3 = 1 0 1 0

P4 bit = Even parity of D7 D6 D5 = 1 0 1 0