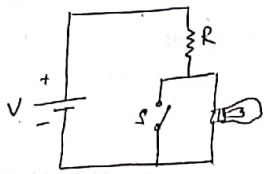
UNIT-2

Logic Operation, Error detection & Correction Codes

Basic Logic Operations:

Logical operator NOT / INVERT:

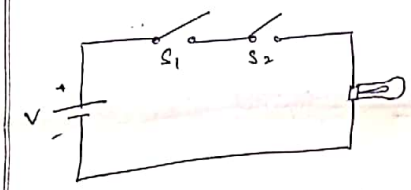
The inversion (or complementing or negation) operator is written as a bar over its argument. Sometimes it is written "NOT". Thus the inverse of A is $\bar{A}$ or NOT A.



Input	Output
'S' open (Low)	Lamp ON (High)
'S' closed (High)	Lamp OFF (Low)

Logical Operator AND:

It is denoted by '.' A·B may be read as A and B.

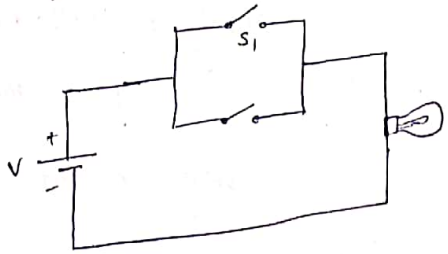


The lamp is ON only when both the switches S_1 & S_2 closed simultaneously.

Input		output
S_1	S_2	
Open (L)	Open (L)	OFF (L)
open (L)	closed (H)	OFF (L)
Closed (H)	Open (L)	OFF (L)
Closed (H)	Closed (H)	ON (H)

Logical Operator OR:

It is represented with '+'. A+B is read as A or B.



If either or both of two separate switches S_1 & S_2 are closed, The lamp will ON. When both S_1 & S_2 are OFF, the lamp will be OFF.

Input		output
S_1	S_2	
L	L	OFF (L)
L	H	H
H	L	H
H	H	H

To implement these logical operations we need some digital circuits called as gates (or) logic gates.

Scanned by CamScanner

Logic Gates :

Logic gates are the basic elements to design a digital system. The electronic gate is a circuit that is able to operate on a number of binary inputs inorder to perform a particular logical function. The type of gates are NOT, AND, OR, NAND, NOR, EX-OR & EX-NOR. Except EX-NOR gate, they are available in monolithic integrated circuit form.

Inverter : NOT gate :

The inverter changes one logic level to its opposite level. It changes a logic 1 to a logic 0 and logic 0 to a logic 1.

Logic Symbol :

The bubble appearing on the output is the negation & inversion indicator.

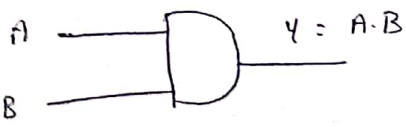
Truth Table :

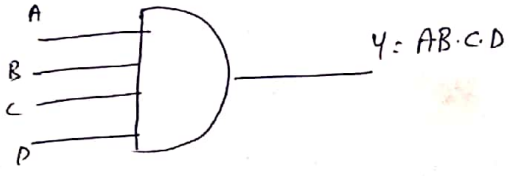
Input	Output
0	1
1	0

AND gate :

The AND gate performs logical multiplication, more commonly known as the AND function. The AND gate may have two & more inputs and a single output.

Logic Symbol :





Scanned by CamScanner

Truth Table

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

When both the inputs are high then only the output of AND gate is high otherwise low.

OR gate

The OR gate performs logical addition, more commonly known as the OR function. It has two or more inputs & one output.

Logic Symbol

Truth Table

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

If any one of the input is high then the output of OR gate is high. When all the inputs are low then only output of OR gate is low.

Boolean Theorems :

Boolean Algebra

It is a set of rules, laws and Theorems by which logical operations can be mathematically expressed.

Boolean Algebra is an Algebra $(B; \cdot, +, ', 0, 1)$ consisting of a set B. Consists atleast two elements 0 and 1 together with three operators AND, OR & NOT.

Scanned by CamScanner

Postulates :

1. **Closure** .

A set B is closed with respect to a binary operator If for every pair of elements of B, the binary operator specifies a rule for obtaining a unique element of B.

2. **Commutative** :

The binary operators + and · on a set B are said to be commutative whenever

$$\left.\begin{aligned} x \cdot y &= y \cdot x \\ x + y &= y + x \end{aligned}\right\} \text{ for all } x, y \in B$$

3. **Associative** :

The binary operators + and · on a set B are said to be associative whenever

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

$$(x + y) + z = x + (y + z)$$

4. **Distributive**

A binary operator + and · on a set B, is said to be Distributive whenever

$$x \cdot (y + z) = x \cdot y + x \cdot z \quad \text{for all } x, y, z \in B$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

Identity element :

A set S is said to have an identity element with respect to a binary operation * and + on S if there exists an element $e \in S$ with the property :

$$e * x = x * e = x \quad \text{for every } x \in S$$

$$e + x = x + e = x \quad \text{for every } x \in S.$$

Scanned by CamScanner

Note: The element '1' is an identity element with respect to operation $\cdot$ on set

$$1 * x = x * 1 = x$$

The element '0' is an identity element with respect to operation $+$ on set

$$0 + x = x + 0 = x.$$

Inverse:

A set S having the identity element e with respect to a binary operator $*$ and $+$ is said to have an inverse whenever, for every $x \in S$, there exists an element $y \in S$ such that:

$$x * y = e$$
$$x + y = e$$

Note: In the set of integers with $e = 1$, the inverse of an element x is $\left(\frac{1}{x}\right)$ then $x \cdot \frac{1}{x} = 1$.

In the set of integers with $e = 0$, the inverse of an element x is $x + (-x) = 0$.

Complement and Dual of logical expressions:

Duality Theorem:

In duality theorem we can change one boolean expression into another boolean expression by

1. Changing each OR sign to an AND sign.
2. Changing each AND sign to an OR sign.
3. Complementing any 0 & 1 appearing in the expression.

Ex: Find the dual form of $F_1 = x'y z' + x'y'z$ and $F_2 = x(y'z' + yz$

Sol: Given that $F_1 = x'y z' + x'y'z$

dual of $F_1 = (x' + y + z')(x' + y' + z)$

$F_2 = x(y'z' + yz)$

dual of $F_2 = x + (y' + z')(y + z)$

Scanned by CamScanner

De Morgan's Theorems :

Demorgan suggested two theorems.

1) $\overline{AB} = \bar{A} + \bar{B}$

The complement of a product is equal to the sum of the complements.

2) $\overline{A+B} = \bar{A} \cdot \bar{B}$

The complement of a sum is equal to the product of Complements.

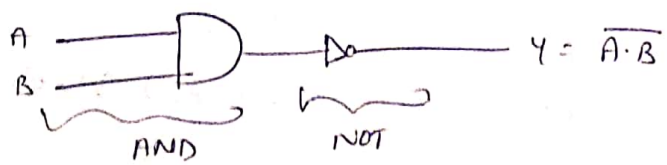
Rules :

1. $x + 0 = x$
2. $x \cdot 1 = x$
3. $x + x' = 1$
4. $x \cdot x' = 0$
5. $x + x = x$ } Idempotency
6. $x \cdot x = x$ }
7. $x + 1 = 1$
8. $x \cdot 0 = 0$
9. $(x')' = x$
10. $x + xy = x$
11. $x(x+y) = x$

NAND gate :

The term NAND implies an AND function with a complemented (inverted) output.

Logic Symbol :



(a)

Scanned by CamScanner

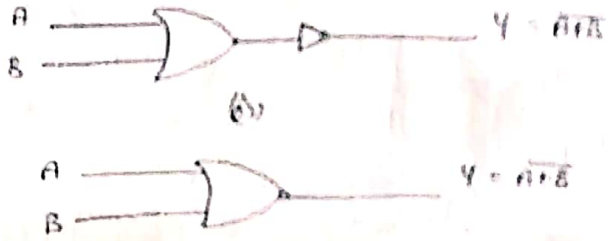
Truth Table

Input		output
A	B	$Y = A\bar{B}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR gate :

The term NOR is implies an OR function with an inverted output

Logic Symbol



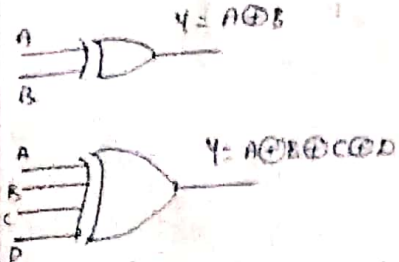
Truth Table :

Input		Output
A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

EX-OR gate

The Ex-OR gate abbreviation is Exclusive- OR gate. An It has two & more inputs & one output

Logic Symbol



Truth Table

Inputs		output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

The o/p of Ex-OR gate is high when there is odd number of 1's present in the i/p.

Scanned by CamScanner

Ex-NOR gate:

The term Ex-NOR is implies that NOT Exclusive OR gate.

Logic Symbol:

(A)

Truth Table:

Input		Output
A	B	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Standard SOP and POS forms:

Boolean function is expressed in Standard form. In this configuration, the terms that form the function may contain one, two & any number of literals. There are two types of Standard form:

The Sum of products and Product of Sums.

The Sum of products is a Boolean expression containing AND terms, called product terms, of one & more literals each. The Sum denotes the ORing of these terms.

Ex: $F_1 = y' + xy + x'y z'$

The above expression has three product terms of one, two and three literals each respectively. Their Sum is in effect an OR operation.

Scanned by CamScanner

A product of Sums is a Boolean expression containing OR terms, called sum terms. Each term may have any no. of literals. The product denotes the ANDing of these terms.

Ex: $F_2 = x(y'+z)(x'+y+z'+w)$

The above expression has three sum terms of one, two and four literals each. The product is an AND operation.

A Boolean function may be expressed in a non-standard form.

Ex: $F_3 = (AB + CD)(A'B' + C'D')$

It is neither in sum of products nor in product of sums. It can be changed to a standard form by using the distributive law to remove the parantheses thus

$$F_3 = A'B'CD + ABC'D'$$

Algebraic Simplification :

→ Reduce $\bar{A}B\bar{C}\bar{D} + B\bar{C}\bar{D} + B\bar{C}\bar{D} + B\bar{C}D$

Sol: $\bar{A}B\bar{C}\bar{D} + B\bar{C}\bar{D} + B\bar{C}\bar{D} + B\bar{C}D$

$= B\bar{C}\bar{D}(\bar{A}+1) + B\bar{C}\bar{D} + B\bar{C}D$

$= B\bar{C}\bar{D} + B\bar{C}\bar{D} + B\bar{C}D$

$= B\bar{D}(C+\bar{C}) + B\bar{C}D$

$= B\bar{D} + B\bar{C}D$

$= B(\bar{D} + \bar{C}D)$

$= B(\bar{D} + \bar{C})$

→ Reduce $AB + \overline{AC} + AB\bar{C}(AB+C)$

$= AB + \overline{AC} + AABB\bar{C} + AB\bar{C}\cdot C$

$= AB + \overline{AC} + AB\bar{C}$

$= AB + \bar{A} + \bar{C} + AB\bar{C}$

Scanned by CamScanner

$= \bar{A} + B + \bar{C} + ABC$

$= \bar{A} + ABC + B + \bar{C}$

$= \bar{A} + \bar{B}C + B + \bar{C}$

$= \bar{A} + B + \bar{C} + BC$

$= \bar{A} + B + \bar{C} + \bar{B}$

$= \bar{A} + \bar{C} + 1$

$= \bar{A} + 1$

$= 1$

→ Reduce $\overline{\bar{A}B + \bar{A} + AB}$

$= \overline{\bar{A} + \bar{B} + \bar{A} + AB}$

$= \overline{\bar{A} + \bar{B} + AB}$

$= \overline{\bar{A} + A + \bar{B}}$

$= \overline{\bar{B} + 1}$

$= \overline{1}$

$= 0$

→ Simplify the expression $a + a\bar{b} + ab\bar{c} + abcd$.

$a + a\bar{b} + ab\bar{c} + abcd$

$= a\,[1 + \bar{b} + b\bar{c} + bcd + \quad]$

$= a\,[1]$

$= a$

→ $(x+y)\,(x'(y'+z'))' + x'y' + x'z'$

$(x+y)\,(x + (y'+z')') + x'y' + x'z'$

$(x+y)\,(x + yz) + x'y' + x'z'$

$x + xy + xyz + yz + x'y' + x'z'$

$x(1+y) + yz(1+x) + x'y' + x'z'$

$x + yz + x'y' + x'z'$

$x + y' + yz + x'z'$

$x + x'z' + y' + yz$

$x + z' + y' + z$ $\qquad x + x'y' + z' + z$

$1 + y' + 1 \quad = 1$

Scanned by CamScanner

→ Given that $AB' + A'B = C$, Show that $AC' + A'C = B$

$AC' + A'C$

$A(AB' + A'B)' + A'(AB + A'B)$

$A[(AB')'(A'B)'] = A'B$

$A[(A' + B)(A + B')] + A'B$

$A[AB + A'B'] + A'B$

$AB + A'B$

$B(A + A')$

$= B$

→ Find the complement of following expressions.

a) $xy' + x'y$

$\overline{xy' + x'y} = \overline{x \oplus y}$

$= x \odot y$

$= x'y' + xy$

b) $[(AB' + C)D' + E]$

$[(AB' + C)D' + E]'$

$[(AB' + C)D']' \cdot E'$

$[(AB' + C)' + D] \cdot E'$

$[(AB')' \cdot C' + D] E'$

$[(A' + B)C' + D] \cdot E'$

→ Using de Morgan's theorem, Convert the following boolean expression to equivalent expression that have only OR.

$F = x'y' + x'z + y'z$

$= (x + y)' + (x + z')' + (y + z')'$

$F = (y + z')(x + y)(y' + z)$

$= [(y + z')' + (x + y)' + (y' + z)']'$

Scanned by CamScanner

→ Convert the Boolean expressions that contains only logic expression.

$$F = x'y' + x'z + y'z$$

$$= [(x'y')' (x'z)' (y'z)]'$$

$$F = (y+z') (x+y) (y+z)$$

$$= (y'z)' (x'y')' (yz')'$$

Gray Code :

The code which exhibits only a single bit change from one code number to the next is known as 'Gray code'. This code is also known as 'reflective code'.

Binary to Gray Conversion :

→ Write the MSB of the binary as it is.

→ Perform the Ex-OR operation MSB to the next lower significant bit of binary number.

→ Continue the same process till all the binary digits are added.

Ex: Convert $(1001)_2$ into Gray Code

Binary number 1 0 0 1

Gray code 1 1 0 1

Gray to Binary Conversion :

→ Write down the MSB as it is

→ Perform Ex-OR operation on result bit & next lower Significant bit.

→ Continue same process until last LSB.

Ex: Convert $(1110)_{Gray}$ to binary

Gray Code 1 1 1 0

Binary number 1 0 1 1

Scanned by CamScanner

Decimal Number	Binary Number	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

Error detection Codes :

In binary system the input & output may be transmitted through some form of communication medium such as wires & radio waves. Any external noise introduced into a physical communication medium changes bit values from 0 to 1 & vice-versa. An error detection code can be used to detect errors during transmission. The detected error cannot be corrected, but its presence is indicated.

Parity Checking Even Parity, Odd Parity :

Parity bit is an extra bit that is transmitted along with the actual data to detect the errors. The parity bit is added such that the no. of 1's in data can be made either even or odd. The parity check is based on the use of the parity bit. In 'Even Parity' the no. of 1's including parity bit is always even. In 'Odd Parity' the no. of 1's including Parity bit is always odd.

Scanned by CamScanner

Message	P (odd)	P (even)
0000	1	0
0001	0	1
0010	0	1
0011	1	0
0100	0	1
0101	1	0
0110	1	0
0111	0	1
1000	0	1
1001	1	0
1010	1	0
1011	0	1
1100	1	0
1101	0	1
1110	0	1
1111	1	0

Let us consider an even parity and assume that a sequence of data 1001, 1101, 0110, 1010 is to be transmitted. The parity bit that are sent will be 0, 1, 0 and 0. Due to some reason, there exists an error in received bits.

Transmitted		Received	
Data	Parity bit	Data	Parity bit
1001	0	1001	0
1101	1	1101	1 ← error
0110	0	0110	0
1010	0	1010	0

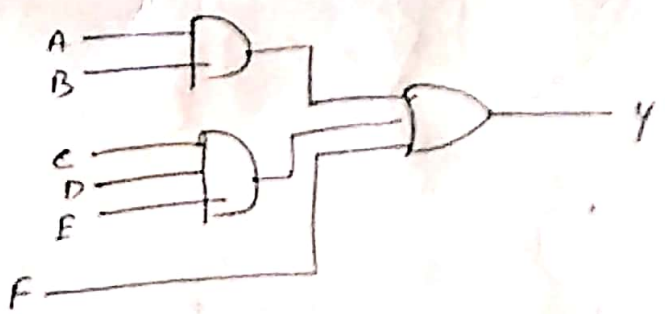
It can be noticed that there is error in second data reception. At the receiver parity is checked for even parity and found that there is an error. But exact location of the error is not found. Same method is adapted for odd parity also.

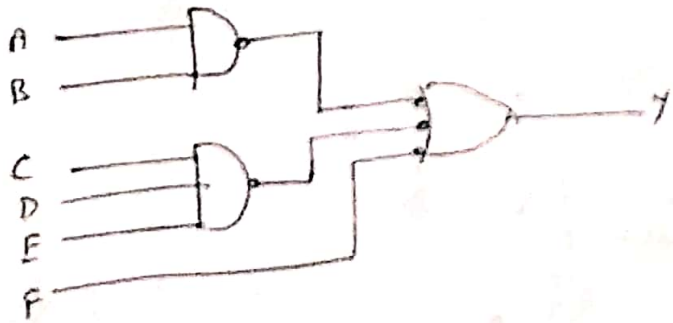
Scanned by CamScanner

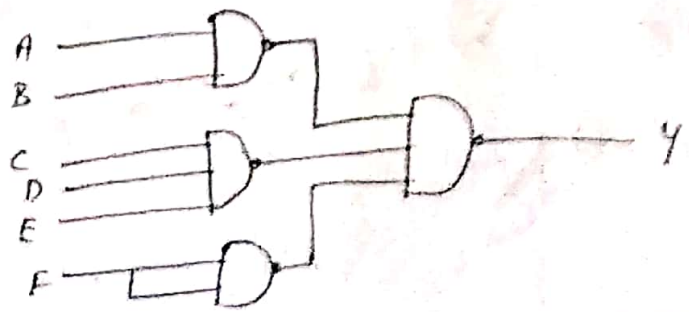
2 Level NAND - NAND implementation

- Simplify the given logic expression and convert it in the SOP form
- Draw logic circuit using AND, OR and NOT gate
- Replace every AND gate by a NAND gate, every OR gate by a bubbled OR gate and NOT gate by a NAND inverter
- Replace bubbled OR gate by NAND gate.

→ Implement following boolean function using only NAND gate: $Y = AB + CDE + F$

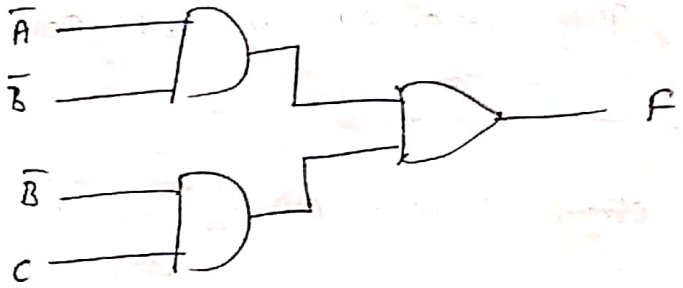


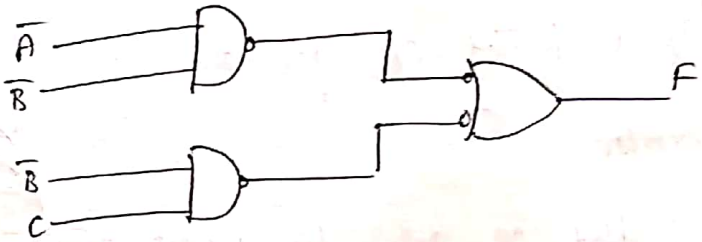


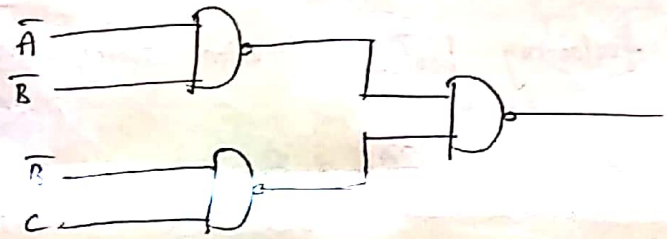


Scanned by CamScanner

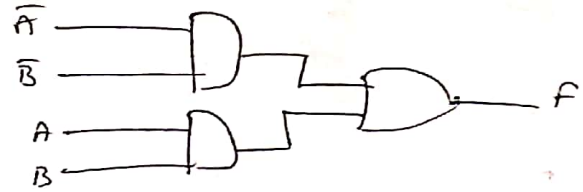
→ $F = \bar{A}\bar{B} + \bar{B}C$

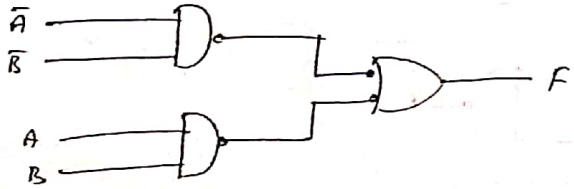


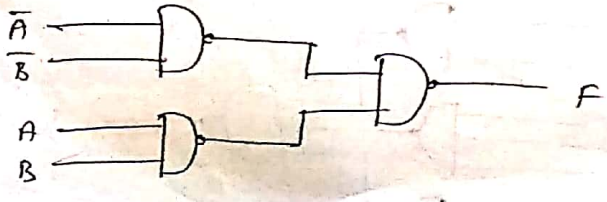




→ $F = \bar{A}\bar{B} + AB$







Scanned by CamScanner

This code is also known as 8-4-2-1 Code & simply BCD code. 8, 4, 2, and 1 are the weights of the four bits of the binary code.

2421 Code :

Here the weightage of four bits are given as 2, 4, 2 and 1.

Decimal Number	8421	2421	5421	5211
0	0000	0000	0000	0000
1	0001	0001	0001	0001
2	0010	0010	0010	0100
3	0011	0011	0011	0101
4	0100	0100	0100	0111
5	0101	0101	0101	1000
6	0110	0110	0110	1001
7	0111	0111	0111	1100
8	1000	1110	1011	1110
9	1001	1111	1100	1111

Un-Weighted Codes :

These codes are given without any positional value for the bits. Examples of un-weighted codes are

1. Excess-3 Code
2. Gray Code.

Excess-3 Code :

This is another form of BCD code, in which each decimal digit is coded into a 4-bit binary code. The code for each decimal digit is obtained by adding decimal 3 to the natural BCD code of the digit.

Scanned by CamScanner

Decimal Number	8421 (BCD)	Excess-3 Code
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

Gray Code:

The code which exhibits only a single bit change from one code number to the next is known as 'Gray Code'. This code is also known as 'reflective code'.

Alphanumeric Codes:

Many applications of digital systems require the handling of data that consists of not only of numbers, but also the letters of the alphabets and certain special characters. Hence to transfer the information one should use a code which contains letters, numbers and other symbols. This types of codes are known as 'Alphanumeric Codes'. For each character, number, special character a unique sequence of 0's and 1's are given.

In this following two codes normally used

1. Extended BCD Interchange Code (EBCDIC)
2. American Standard Code for Information Interchange (ASCII)

Scanned by CamScanner

Error - Correcting Codes :

Hamming Code :

Hamming code not only provides the detection of a bit error, but also identifies which bit is in error so that it can be corrected. Thus Hamming code is called error detecting and correcting code.

No. of Parity Bits :

If no. of information bits is 'x', then the no. of parity bits, P is determined by the following relationship :

$$2^P \geq x + P + 1$$

Locations of the parity bits in the Code :

The parity bits P_k are to be located at the positions, in which the position number must be power of '2'.

Ex: If x = 4, and message = 1010

Then the length of code word = x + P

$$x = 4$$

$$2^P \geq x + P + 1$$

If $P = 1$

$$2^1 \geq 4 + 1 + 1$$

$$2 \geq 6 \quad \times$$

If $P = 2$

$$2^2 \geq 4 + 2 + 1$$

$$4 \geq 7 \quad \times$$

If $P = 3$

$$2^3 \geq 4 + 3 + 1$$

$$8 \geq 8 \quad \checkmark$$

$\therefore P = 3$

$\therefore$ Length of Code word = $4 + 3 = 7$.

Scanned by CamScanner

Calculation of Parities :

7	6	5	4	3	2	1
1	0	1	P_4	0	P_2	P_1

	P_4 2 P_1
0 →	0 0 0
1 →	0 0 1
2 →	0 1 0
3 →	0 1 1
4 →	1 0 0
5 →	1 0 1
6 →	1 1 0
7 →	1 1 1

P_1 is calculated as the positions 1, 3, 5, 7 establish required parity.

P_2 is calculated as the positions 2, 3, 6, 7 establish required parity.

P_4 is calculated as the positions 4, 5, 6, 7 establish required parity.

Assume even parity

P_1 = even parity of 1, 3, 5, 7 = P_1 011

$\therefore P_1 = 0$

P_2 = even parity of 2, 3, 6, 7 = P_2 001

$\therefore P_2 = 1$

P_4 = even parity of 4, 5, 6, 7 : P_4 101

$\therefore P_4 = 0$

Then Code word is 1 0 1 0 0 1 0

If we received code word 1 0 1 1 0 1 0 (7 6 5 4 3 2 1)

To find the error position recalculate parity bits.

P_1 = 1, 3, 5, 7 = 0 0 1 1 = 0 ↑

P_2 = 2, 3, 6, 7 = 1 0 0 1 = 0 | $(100)_2 = 4$.

P_3 = 4, 5, 6, 7 = 1 1 0 1 = 1

$\therefore$ There was error in 4th position

Scanned by CamScanner

→ Consider a message 1001101 is transmitted through the channel. Obtain the redundancy bits of transmitting unit. Assume bit no's has been change. How to locate it.

Message = 1001101

$x = 7$

$2^4 \geq 7+4+1$

$p = 4$

11	10	9	8	7	6	5	4	3	2	1
1	0	0	P_8	1	1	0	P_4	1	P_2	P_1

P_1 = 1, 3, 5, 7, 9, 11

P_2 = 2, 3, 6, 7, 10, 11

P_4 = 4, 5, 6, 7

P_8 = 8, 9, 10, 11

Assume Choose odd Parity

P_1 : P_1 1 0 1 0 1 = 0

P_2 : P_2 1 1 1 0 1 = 1

P_4 = P_4 0 1 1 = 1

P_8 = P_8 1 0 0 = 0

Transmitted Code word 1000 1101110

Assume if there is an error in 8th position.

Received code coord 1001 1101110

P_1 = 1, 3, 5, 7, 9, 11 = 010101 = 0

P_2 = 2, 3, 6, 7, 10, 11 = 111101 = 0

P_4 = 4, 5, 6, 7 = 1011 = 0

P_8 = 8, 9, 10, 11 = 1001 = 1

$(1000)_2 = (8)_{10}$

∴ Error present in eighth position.

Scanned by CamScanner