(i) combinational circuits.

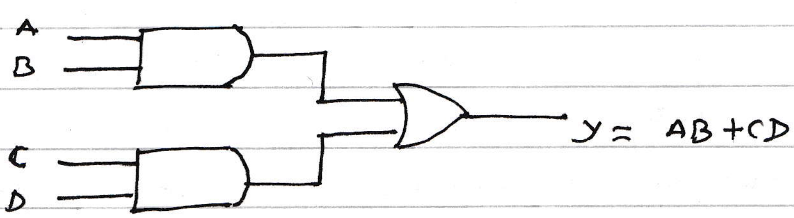
(ii) sequential circuits.

Combinational circuits	Sequential circuits.
(1) The combinational circuits are logical circuits in which the output at any instant of time depends only on the combination of inputs ~~and does not require memory~~ at that instant and not on the past value of inputs or outputs.	(1) sequential circuits are the logic circuits in which the output depends on the present state of inputs, previous output and the sequence in which inputs are applied.
(2) combinational circuits doesnot require memory.	(2) sequential circuits requires memory.
(3) Combinational circuits consists of logical gates.	(3) sequential circuits consists of memory element with combinational circuit.
(4) e.g. Adder, substractor, multiplexer, demultiplexer etc.	(4) flip-flops, registers, counters etc.

Introduction to SOP and POS tecnique.

The relation between the inputs and outputs of the logical or digital circuit can be expressed mathematically by an expression called as Boolean expression.

e.g. $Y = AB + CD$ ——— (1)



In eq^n (1), y is output and A, B, C, D are inputs called literals or variables.

Any logical expression can be expressed in following two standard forms:

(1) SUM OF PRODUCT (SOP) form.

(2) PRODUCT OF SUM (POS) form.

$y = ABC + ABC + ABC$ —②

$y = \bar{A}B + \bar{A}B\bar{C} + AC$ —③

In standard SOP form, each product term includes all the variables in complemented or uncomplimented form.

eqⁿ ① and ③ are not standard SOP form. eqⁿ ② is in standard SOP form because each product term in this eqⁿ contain all the variables (A, B, C) either complemented or uncomplemted form.

Each individual term in eqⁿ ② is called minterm (m)

Conversion of SOP equation into standard SOP form.

① let $y = \bar{A}B + A\bar{C} + BC$ —①

As each product term doesnot contain all the variables above eqⁿ is not standard SOP form.

$\therefore y = \bar{A}B(C+\bar{C}) + A\bar{C}(B+\bar{B}) + BC(A+\bar{A})$

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

$y = \bar{A}BC + \bar{A}B\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + ABC + \bar{A}BC$

In above eqⁿ $\bar{A}BC$ is a redundant term, so one term is dropped out.

$\therefore y = \bar{A}BC + \bar{A}B\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + ABC$ —②

This expression is in standard SOP form.

② convert $y = AB + A\bar{B}C$

The variable C is missing in first product term.

$y = AB(C+\bar{C}) + A\bar{B}C$

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

This expression is in standard SOP form.

③ find the minterm for following boolean expression

$y(A,B,C) = A\bar{B} + \bar{C}$

In first term C is missing and in second term BC is missing

$\therefore y = A\bar{B}(C+\bar{C}) + \bar{C}(A+\bar{A})(B+\bar{B})$

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

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

$A\bar{B}\bar{C}$ is redundant term $\therefore$ drop it

m_0 (ABC), m_1 (ABC), m_2 (ABC), m_3 (ABC),

m_4 $(\bar{A}\bar{B}\bar{C})$ and m_5 $(\bar{A}B\bar{C})$.

Minterms – The output of AND gate is called product term or minterm.

Product of sum expression.

When two or more ~~prod~~ sum are multiplied (product) then the resulting expression is a product of sum (POS) form.

e.g.

$$(A+\bar{B})\cdot(\bar{A}+B+C) \quad \text{——} \ (1)$$

$$(A+B+C)\cdot(\bar{A}+B)\cdot(\bar{A}+\bar{C}) \quad \text{——} \ (2)$$

$$(A+B)\cdot(\bar{A}+B)\cdot(\bar{B}+\bar{A}) \quad \text{——} \ (3)$$

eqn (1) and (2) are not standard POS form but eqn (3) is in standard POS form because each sum contains all the variables in complemented or uncomplemented form.

Each individual term in equation (3) is called Maxterm (M)

Maxterm: The output of OR gate is called sumterm or Maxterm.

(1) Convert following expression into standard POS equation.

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

Soln In given expression there are four variables A, B, C, D. In first term D (or $\bar{D}$) is missing and in second term A ($\bar{A}$) is missing. Therefore adding $D\cdot\bar{D}$ and $A\cdot\bar{A}$ in first and second term resp.

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

$$\because A\bar{A} = D\bar{D} = 0$$

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

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

Redundand term

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

This expression is in standard POS form.

3 variables, $2^3 = 8$ diff — " —

4 variables, $2^4 = 16$ diff " and so on.

The procedure for constructing truth-table for standard SOP expression is as follows –

(1) find number of variables in the domain of given expre-ssion.

(2) List all possible combinations of binary values of the variables in the expression.

(3) write 1 in the output column for which binary values that make the standard product term equal to 1 and place a 0 for all the remaining binary combination.

e.g. let $y = \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

Here 3 variables therefore 8 diff. combinations.

Input Variables A	B	C	output	Product term (minterm)
0	0	0	0	m_0
0	0	1	1	$m_1 \rightarrow \bar{A}\bar{B}C$
0	1	0	0	m_2
0	1	1	0	m_3
1	0	0	0	$m_4 \rightarrow A\bar{B}\bar{C}$
1	0	1	1	$m_5 \rightarrow$
1	1	0	0	m_6
1	1	1	1	$m_7 \rightarrow ABC$

Conversion of POS expression into truth table format

The procedure for constructing truth-table for standard POS expression is as follows –

(1) find number of variables in the domain of given expression

(2) List all possible combinations of binary values of the variables in the expression.

(3) write 0 in the output column for which binary values that make the sum term equal to 0 and Place a 1 for all the remaining binary combination.

Input Variables A	B	C	output	Sum term (Maxterm)	
0	0	0	0	→ M_0	$(A+B+C)$
0	0	1	1	M_1	
0	1	0	0	M_2	$(A+\overline{B}+C)$
0	1	1	1	M_3	
1	0	0	1	M_4	
1	0	1	0	M_5	$(\overline{A}+B+\overline{C})$
1	0	0	0	M_6	$(\overline{A}+\overline{B}+C)$
1	1	1	1	M_7	

Karnaugh Map

A Karnaugh map is a visual display of the fundamental products needed for a sum of products solution.

2 variable K'map.

A	B	Y	
0	0	0	m_0
0	1	0	m_1
1	0	1	m_2
1	1	1	m_3

(a) Truth-table

	$\overline{B}$	B
$\overline{A}$	0	0
A	1	1

$y = \Sigma m\,(2,3)$

(b) K-map for 2-variable.

3-variable K-map

A	B	C	Y	
0	0	0	0	m_0
0	0	1	0	m_1
0	1	0	1	m_2
0	1	1	0	m_3
1	0	0	0	m_4
1	0	1	0	m_5
1	1	0	1	m_6
1	1	1	1	m_7

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}B$	1	0
AB	1	1
$A\overline{B}$	0	0

$y = \Sigma m\,(2,6,7)$

A	B	C	D	y	
0	0	0	0	0	(m_0)
0	0	0	1	1	(m_1)
0	0	1	0	0	(m_2)
0	0	1	1	0	(m_3)
0	1	0	0	0	(m_4)
0	1	0	1	0	(m_5)
0	1	1	0	1	(m_6)
0	1	1	1	1	(m_7)
1	0	0	0	0	(m_8)
1	0	0	1	0	(m_9)
1	0	1	0	0	(m_{10})
1	0	1	1	0	(m_{11})
1	1	0	0	0	(m_{12})
1	1	0	1	0	(m_{13})
1	1	1	0	1	(m_{14})
1	1	1	1	0	(m_{15})

(a) Truth table

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	0	1	1
AB	0	0	0	1
$A\bar{B}$	0	0	0	0

(b) K-map for 4 variable.

$$y = \sum m(1, 6, 7, 14)$$

pairs :- A group of two adjustant (vertically or horizontally) 1's forms a pair. pair eliminates one variable.

e.g. $y = ABC + AB\bar{C}$

Three variable k-map

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	0
AB	1	1
$A\bar{B}$	0	0

pair

pair elimates 'c' variable.

we have

$$y = ABC + AB\bar{C}$$

$$= AB(C + \bar{C})$$

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

$$y = AB(1)$$

$$\boxed{y = AB}$$

variables and their complements.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	1	1	1	1
$A\bar{B}$	0	0	0	0

quad.

$y = AB$ (eliminate CD)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	0	0	1	1
$A\bar{B}$	0	0	1	1

quad

$y = AC$ (elimate B & D)

Octate - Octate is a group of eight 1's. An octate eliminates <u>three</u> variables and their complements.

e.g.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	0	0
AB	1	1	1	1
$A\bar{B}$	1	1	1	1

octate.

$y = A$

K-map method for Simplifying Boolean equation

1. Enter a 1 on the K-map for each fundamental product that produces a 1 output in the truth-table. Enter 0s otherwise.
2. Encircle the octate, quad & pairs. Remember to roll and overlap to get largest groups possible.
3. If any isolated 1's remain, encircle it.
4. Eliminate any redundant group.
5. Write the boolean equation by ORing the products corrousponding to the encircled groups.

$y = ABCD + ABCD + ABCD + ABCD$

Sol- K-map of above expression is

	$\bar C\bar D$	$\bar C D$	CD	$C\bar D$
$\bar A\bar B$	0	0	0	0
$\bar A B$	1	1	0	0
AB	1	1	0	0
$A\bar B$	0	0	0	0

quad.

from K-map

$y = B\bar C$

Proof- $y = \bar A B\bar C\bar D + AB\bar C\bar D + \bar A B\bar C D + AB\bar C D$

$= \bar A B\bar C(D+\bar D) + AB\bar C(\bar D + D)$

$= \bar A B\bar C + AB\bar C \quad \because D+\bar D = 1$

$= B\bar C(\bar A + A)$

$\boxed{y = B\bar C} \quad \because \bar A + A = 1$

Thus variable A and D are eliminated.

② Map the following standard expression on a K-map.

(a) $\bar A\bar B C + \bar A B\bar C + AB\bar C + ABC$

Solution- The K-map is as

	$\bar C$	C
$\bar A\bar B$	0	1
$\bar A B$	1	0
AB	1	1
$A\bar B$	0	0

(b) $y = \bar A\bar B\bar C\bar D + ABCD + \bar A BCD + A\bar B CD + \bar A\bar B CD + AB\bar C\bar D$

Solution- The K-map is as

	$\bar C\bar D$	$\bar C D$	CD	$C\bar D$
$\bar A\bar B$	1	0	1	0
$\bar A B$	0	0	1	0
AB	1	0	1	0
$A\bar B$	0	0	1	0

Sol:- K – Map is as

	$B\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	1	1
AB	0	0
$A\bar{B}$	0	0

pair

$y = \bar{A}B$

Proof -

We have

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

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

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

$y = \bar{A}B(1)$

$\therefore \quad y = \bar{A}B$

(4) Map the expression $A + AC\bar{D} + \bar{A}BC\bar{D}$ on K-map.

Solⁿ Given expression is not standard SOP eqⁿ

$\therefore \quad y = A + AC\bar{D} + \bar{A}BC\bar{D}$

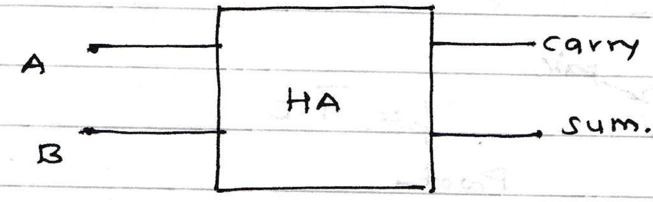
$y = A(B+\bar{B})(C+\bar{C})(D+\bar{D}) + A(B+\bar{B})C\bar{D} + \bar{A}BC\bar{D}$

$y = (AB + A\bar{B})(CD + C\bar{D} + \bar{C}D + \bar{C}\bar{D}) + ABC\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D}$

$= ABCD + \underline{ABC\bar{D}} + AB\bar{C}D + AB\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}\bar{C}D$

$+ A\bar{B}\bar{C}\bar{D} + \underline{ABC\bar{D}} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D}$

has two inputs (A, B) and two outputs (SUM, CARRY).
It is used to add the least significant column.



(a) Block diagram.

Inputs		Outputs	
A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Truth table.

since the half adder has two outputs (sum and carry) One k-map is plotted for each output.

	$\overline{B}$	B
$\overline{A}$	0	0
A	0	1 ← AB

(c) k-map for carry

	$\overline{B}$	B
$\overline{A}$	0	1 ← $\overline{A}B$
A	1 ($A\overline{B}$)	0

(d) k-map for sum.

outputs-

$$\text{carry} = AB$$

$$\text{sum} = \overline{A}B + A\overline{B}$$

$$= A \oplus B$$

$$\therefore\ y = \underbrace{AB}_{\text{carry}} + \underbrace{A \oplus B}_{\text{sum}}$$

∴ logic diagram is

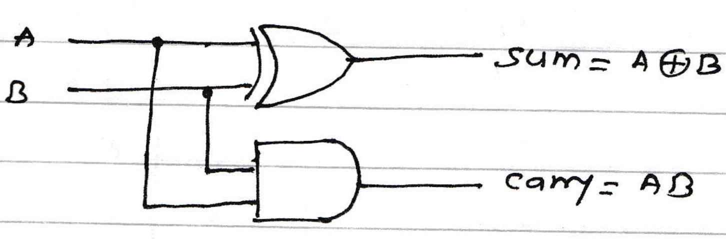
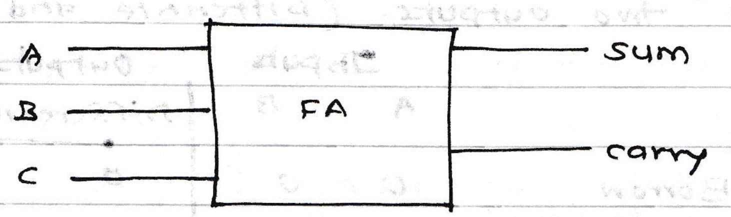


fig Half Adder logic diagram.



The truth-table for full Adder is

A	B	C	carry	sum
0	0	0	0	0
0	0	1	0	1
0	[illegible]	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Since the full Adder has two outputs (sum and carry)

K-maps are

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	0
$\bar{A}B$	0	1
AB	1	1
$A\bar{B}$	0	1

K-map for carry

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	1	0
AB	0	1
$A\bar{B}$	1	0

K-map for sum

$carry = AB + BC + AC$

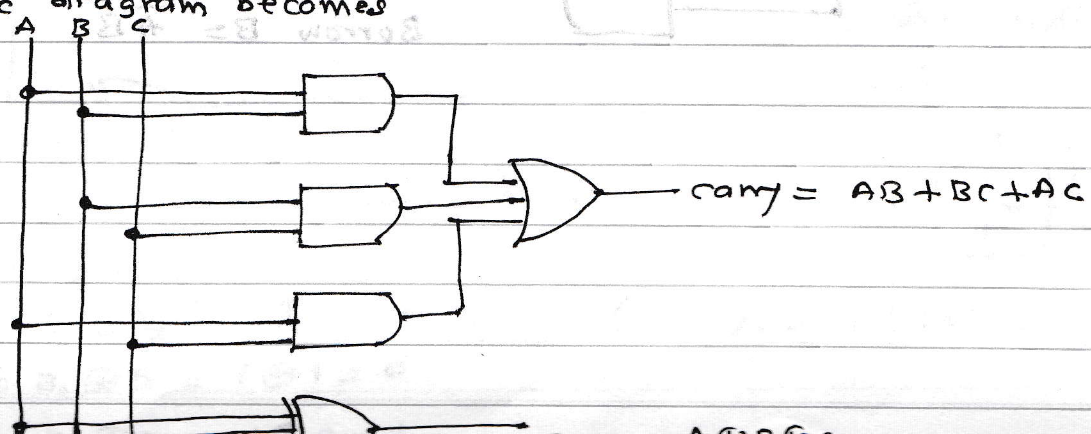
$\cancel{= A \oplus B \oplus C}$

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

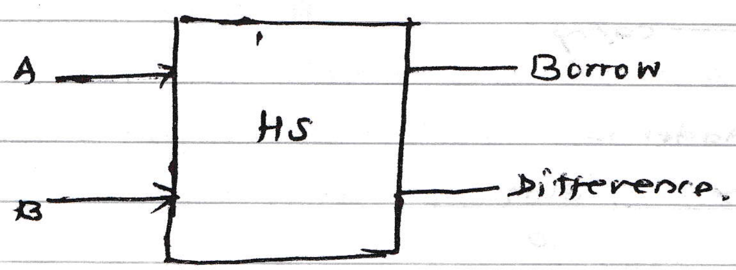
$= \bar{A}(\bar{B}C * B\bar{C}) * (AB + A$

$= A \oplus B \oplus C$

∴ logic diagram becomes



Half substractor is a combinational logic circuit with two inputs and two outputs. (Difference and Borron)



(a) Block diagram.

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(b) Truth-table.

since full Adder has two outputs

	$\overline{B}$	B
$\overline{A}$	0	1 ← $\overline{A}B$
A	1	0

$A\overline{B}$

(c) K-Map for difference output

	$\overline{B}$	B
$\overline{A}$	0	1 ← $\overline{A}B$
A	0	0

(d) K-map for borrow output.

Boolean expression for difference $D = \overline{A}B + A\overline{B}$

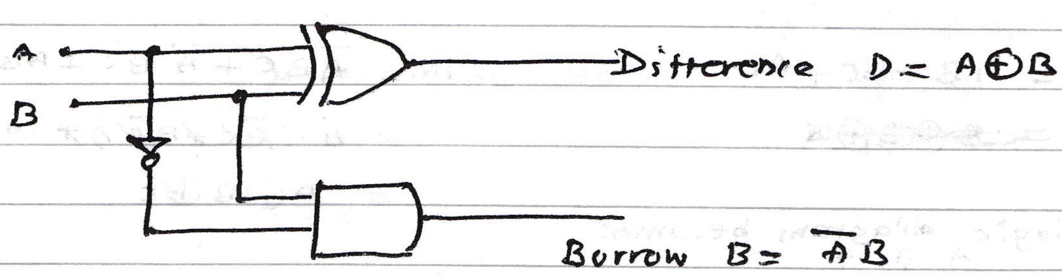
$D = A \oplus B$

" " for Borrow $B = \overline{A}B$

∴ ~~Boolean expression for HS is~~

$\cancel{B} = \cancel{A \oplus B}$

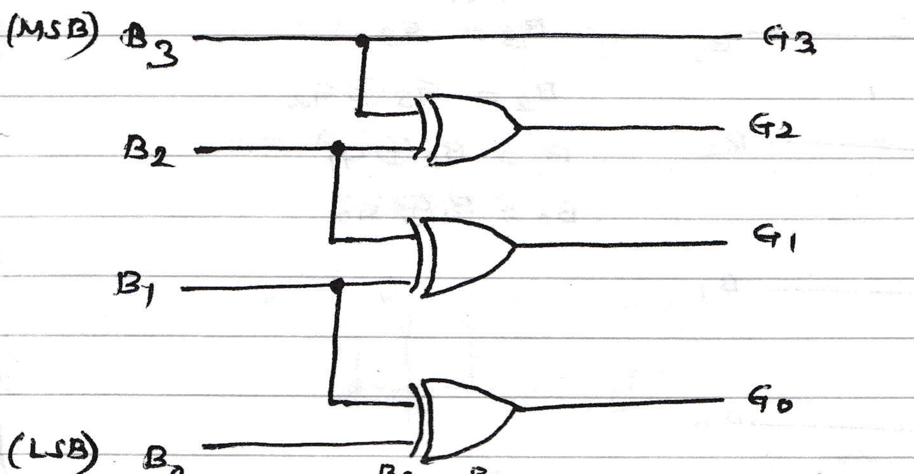
∴ Logic circuit for HS is



weights. It is not an arithmatic code. It is a four bit numeric code in which decimal numbers 0 to 15 are represented by four bit binary code.

Decimal	Gray code	Binary code
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 1	0 0 1 0
3	0 0 1 0	0 0 1 1
4	0 1 1 0	0 1 0 0
5	0 1 1 1	0 1 0 1
6	0 1 0 1	0 1 1 0
7	0 1 0 0	0 1 1 1
8	1 1 0 0	1 0 0 0
9	1 1 0 1	1 0 0 1
10	1 1 1 1	1 0 1 0
11	1 1 1 0	1 0 1 1
12	1 0 1 0	1 1 0 0
13	1 0 1 1	1 1 0 1
14	1 0 0 1	1 1 1 0
15	1 0 0 0	1 1 1 1

Binary to Gray conversion-



i.e

$G_3 = B_3$

$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

$G_0 = B_1 \oplus B_0$

Ex-OR Gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

e.g. $(\overset{B_3}{1}11\overset{B_0}{0})_2 = (\underline{\qquad})_{Gray}$

$G_3 = B_3 = 1$

$G_2 = B_3 \oplus B_2 = 1 \oplus 1 = 0$ $\therefore (1110)_2 = (1001)_{Gray}.$

$G_1 = B_2 \oplus B_1 = 1 \oplus 1 = 0$

Sol^n The Binary for (15) is 1111.

$\therefore B_3 = 1, \quad B_2 = 1, \quad B_1 = 1$ and $B_0 = 1$

$G_3 = B_3 = 1$

$G_2 = B_3 \oplus B_2 = 1 \oplus 1 = 0$

$G_1 = B_2 \oplus B_1 = 1 \oplus 1 = 0$

$G_0 = B_1 \oplus B_0 = 1 \oplus 1 = 0$

$\therefore (1111)_2 = (1000)_{Gray}$.

③ find Gray code for (111001100)

Sol^n

We have $B_8 = 1, B_7 = 1, B_6 = 1, B_5 = 0, B_4 = 0, B_3 = 1, B_2 = 1$

$B_1 = 0$ and $B_0 = 0$

$G_8 = B_8 = 1$

$G_7 = B_8 \oplus B_7 = 1 \oplus 1 = 0$

$G_6 = B_7 \oplus B_6 = 1 \oplus 1 = 0$

$G_5 = B_6 \oplus B_5 = 1 \oplus 0 = 1$

$G_4 = B_5 \oplus B_4 = 0 \oplus 0 = 0$

$G_3 = B_4 \oplus B_3 = 0 \oplus 1 = 1$

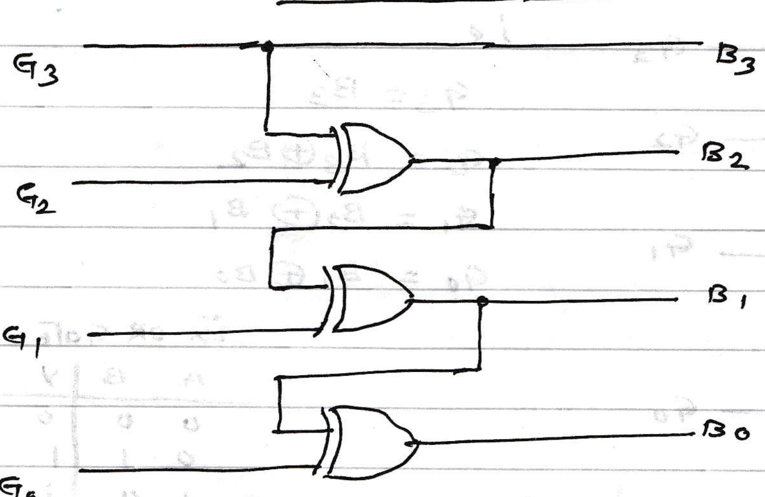
$G_2 = B_3 \oplus B_2 = 1 \oplus 1 = 0$

$G_1 = B_2 \oplus B_1 = 1 \oplus 0 = 1$

$G_0 = B_1 \oplus B_0 = 0 \oplus 0 = 0$

$\therefore (111001100)_2 = (100101010)_{Gray}$

Gray to Binary conversion



i.e.

$B_3 = G_3$

$B_2 = G_3 \oplus G_2$

$B_1 = B_2 \oplus G_1$

$B_0 = B_1 \oplus G_0$

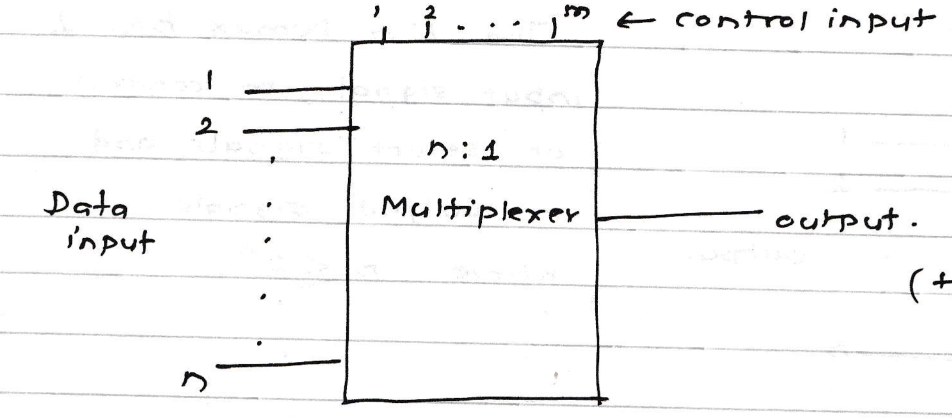
e.g. $(1000)_{Gray} = (\_\_\_\_\_)_{Gray}$.

$B_3 = G_3 = 1$

$B_2 = G_3 \oplus G_2 = 1 \oplus 0 = 1$ $\quad \therefore (1000)_{Gray} = (1111)_2$

$B_1 = B_2 \oplus G_1 = 1 \oplus 0 = 1$

output. Thus it is also called a data selector and control inputs are termed as select inputs.



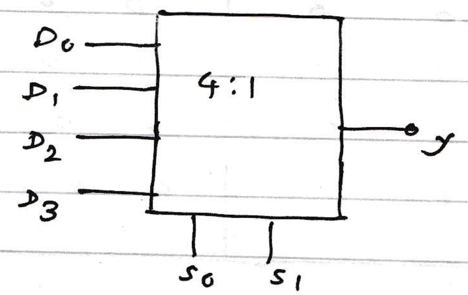
(Here $n \leq 2^m$)

(a) mux block diagram.

4 : 1 Multiplexer :

4:1 MUX has 4 data input and only one output.

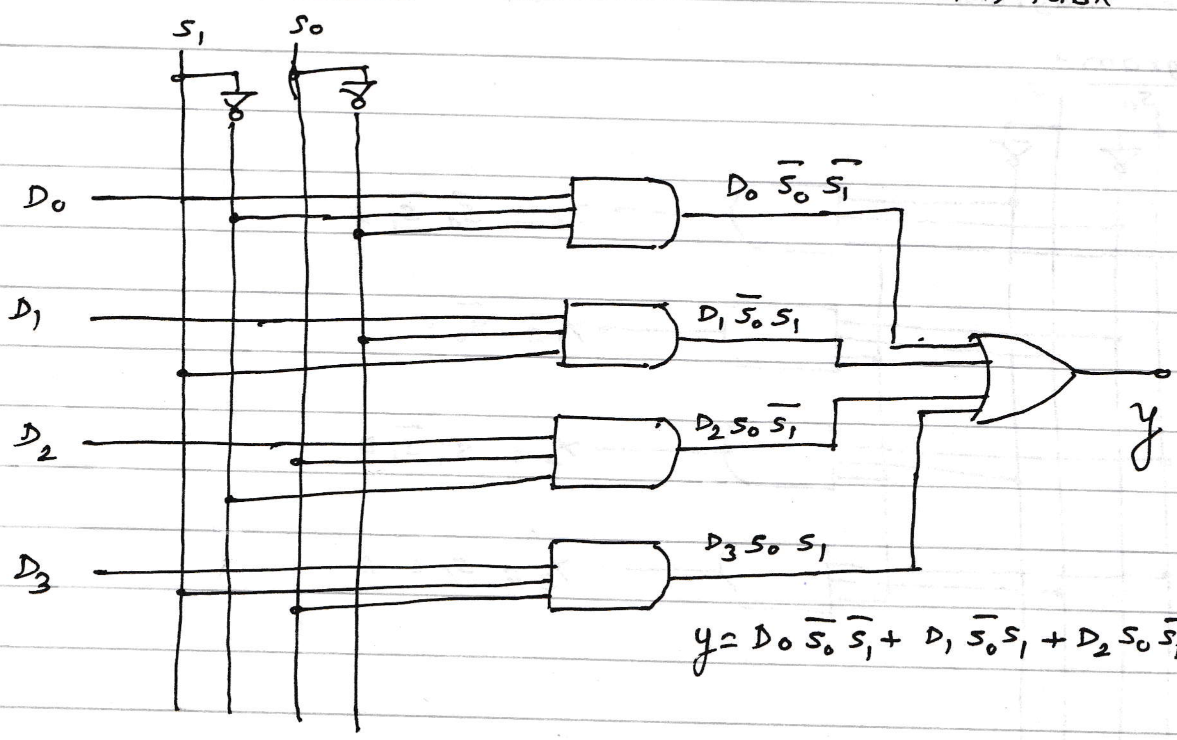
The number of control inputs are 2. (S_0 and S_1)



4:1 MUX

Input S_0	Input S_1	output y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Truth table



$D_0 \bar{S_0} \bar{S_1}$

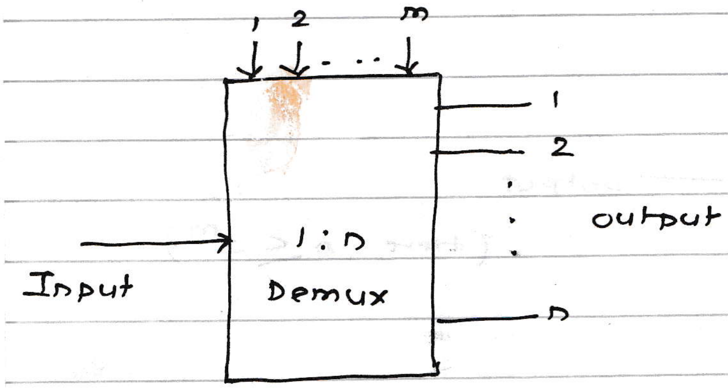
$D_1 \bar{S_0} S_1$

$D_2 S_0 \bar{S_1}$

$D_3 S_0 S_1$

$$y = D_0 \bar{S_0} \bar{S_1} + D_1 \bar{S_0} S_1 + D_2 S_0 \bar{S_1} + D_3 S_0 S_1$$

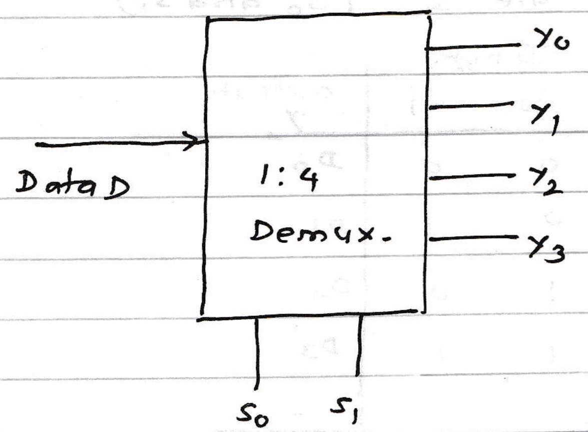
applying control signals, we can steer the input signal to one of the output lines.



The 1:n Demux has 1 input signal, m control or select signals and n output signals where $n \leq 2^m$.

(a) Block diagram.

1:4 Demux - This Demux has 1 input, 4 outputs and 2 control inputs ($2^2 = 4$ outputs).



(a) block diagram

Select input		outputs			
S_0	S_1	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

(b) Truth-table.

Logic diagram -