Conversion of SOP form to Standard SOP form or Canonical SOP form

1) $f(A,B,C) = AB + AC + BC$

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

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

$$\boxed{f(A,B,C) = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC}$$

$= \sum m(7,6,5,3)$

2) $f(A,B) = A + \bar{A}B$

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

$$\boxed{f(A,B) = AB + A\bar{B} + \bar{A}B}$$

$= \sum m(3,2,1)$

$X + \bar{X} = 1$

$X \cdot 1 = X$

$X + X = X$

Conversion of POS form to standard POS form or Canonical POS form

1) $f(A,B) = (\bar{A} + B).(A)$

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

$$\boxed{f(A,B) = (\bar{A} + B).(A + B).(A + \bar{B})}$$

$$= \Pi M(2,0,1)$$

1) $X.\bar{X} = 0$

2) $X + 0 = X$

3) $X + YZ = (X+Y).(X+Z)$

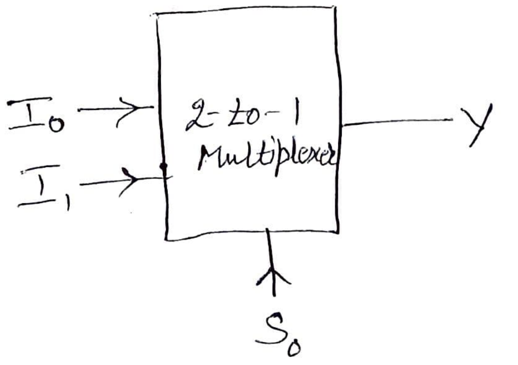
2) $F = (\bar{p} + q + r) * (\bar{q} + r + \bar{s}) * (p + \bar{q} + \bar{r} + s)$

$$= (\bar{p} + q + r + s.\bar{s}) * (\bar{q} + r + \bar{s} + p.\bar{p}) * (p + \bar{q} + \bar{r} + s)$$

$$= (\bar{p} + q + r + s) * (\bar{p} + q + r + \bar{s}) * (\bar{q} + r + \bar{s} + p) * (\bar{q} + r + \bar{s} + \bar{p}) * (p + \bar{q} + \bar{r} + s)$$

$$= \Pi M(8,9,10,11,5)$$

MULTIPLEXER



S_0	Y
0	I_0
1	I_1

$$Y = I_0 \overline{S_0} + I_1 S_0$$

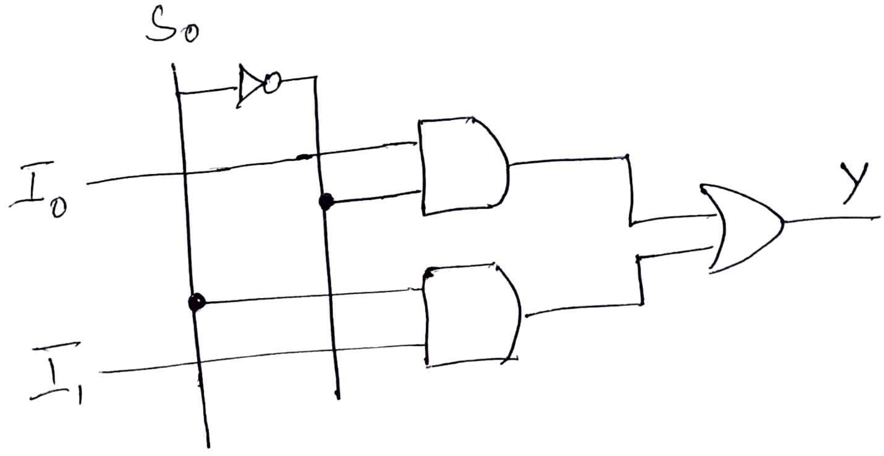
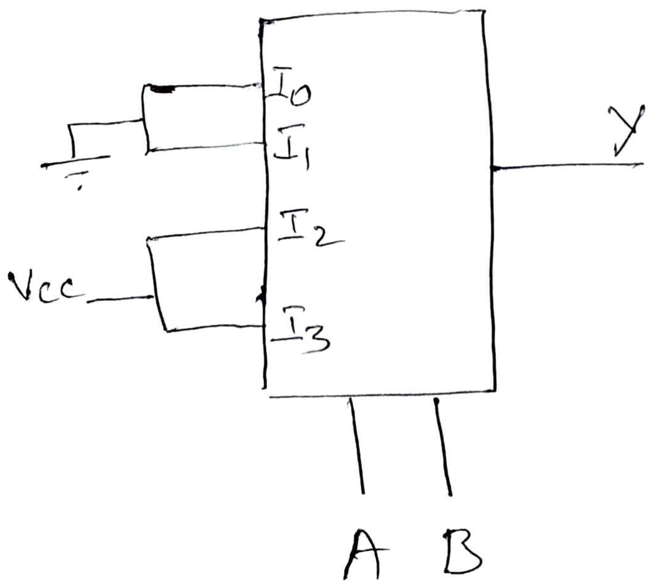


Figure: 2-to-1 MUX

a) Realize $f(A,B) = \sum m(2,3)$ using 4-to-1 MUX



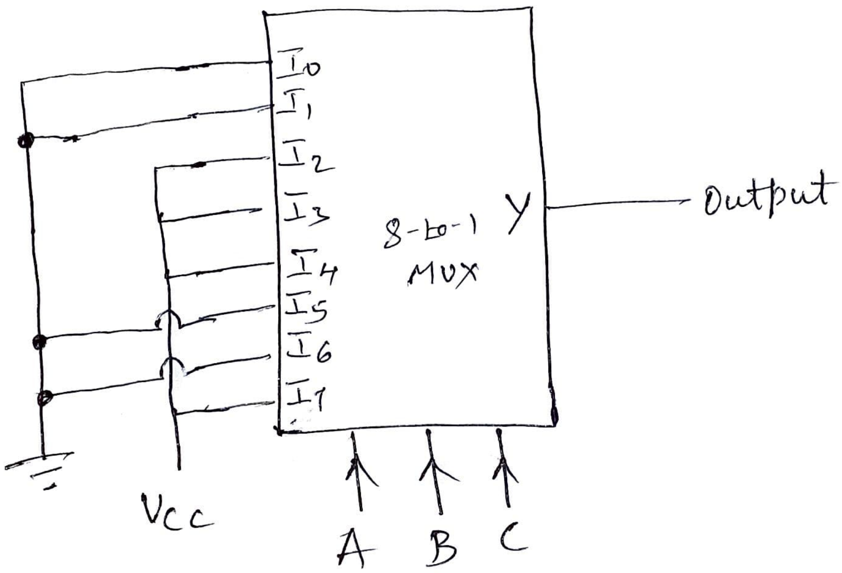
1) Realize $Y = \bar{A}B + \bar{B}\bar{C} + ABC$ using an 8-to-1 MUX

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

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

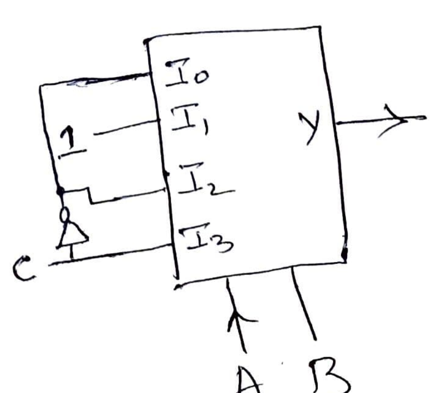
$$= \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + ABC$$

$$= \sum m(3, 2, 4, 0, 7)$$



2) Realize $f(A,B,C) = \sum m(0,2,3,4,7)$ using 4-to-1 Mux

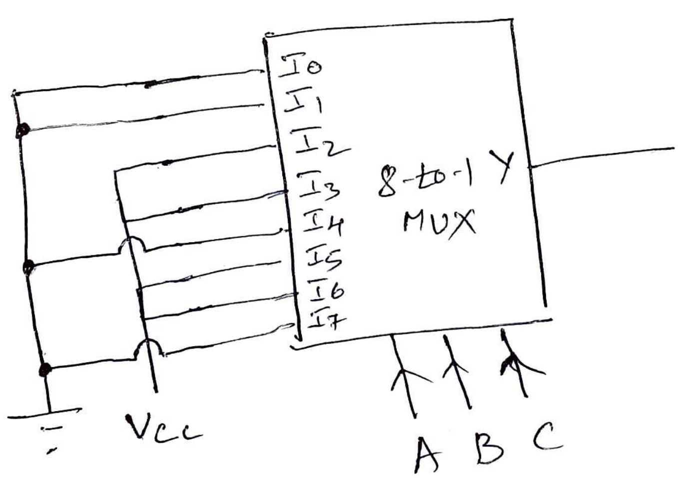
A	B	C	Y	
0	0	0	1	$\bar{C} - I_0$
0	0	1	0	
0	1	0	1	$1 - I_1$
0	1	1	1	
1	0	0	1	$\bar{C} - I_2$
1	0	1	0	
1	1	0	0	$C - I_3$
1	1	1	1	

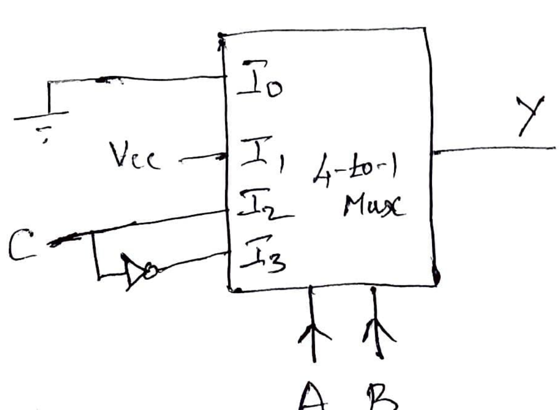


Realize $f(A,B,C) = \sum m(2,3,5,6)$ using

i) 4-to-1 Mux

A	B	C	Y		
0	0	0	0	0	— I_0
0	0	1	0		
0	1	0	1	1	— I_1
0	1	1	1		
1	0	0	0	C	— I_2
1	0	1	1		
1	1	0	1	$\bar{C}$	— I_3
1	1	1	0		





Design 4-to-1 Multiplexer using only 2-to-1 Multiplexers

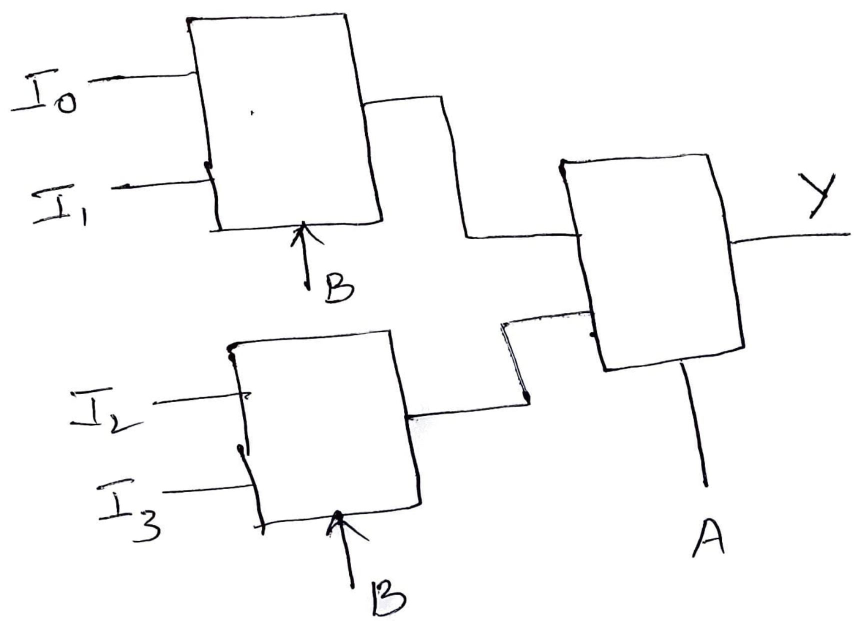
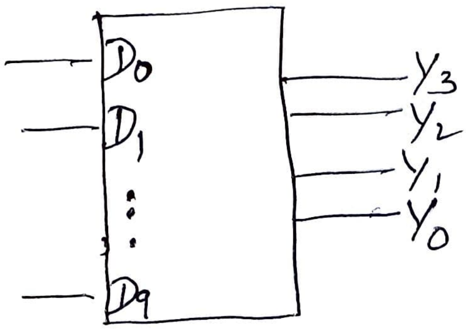


Figure: Realization of higher order multiplexers using lower orders

Encoder

An encoder is a device which converts familiar numbers or characters or symbols into a coded format.

Decimal to ~~Binary~~ BCD Encoder



D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

$$Y_3 = D_8 + D_9$$

$$Y_2 = D_4 + D_5 + D_6 + D_7$$

$$Y_1 = D_2 + D_3 + D_6 + D_7$$

$$Y_0 = D_1 + D_3 + D_5 + D_7 + D_9$$

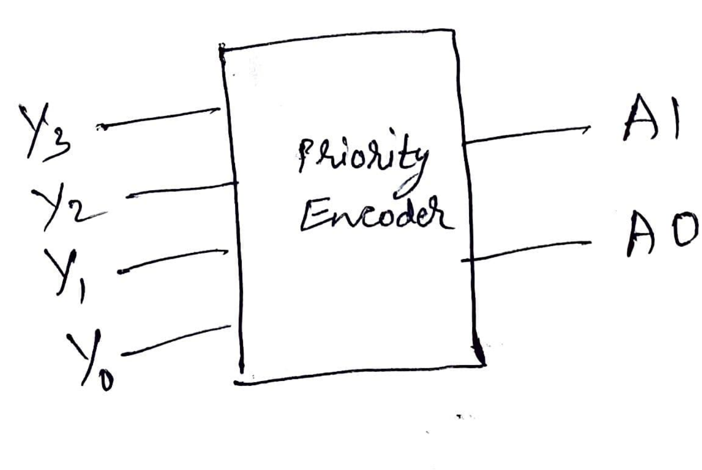
Priority Encoder

A priority encoder is a circuit or algorithm that compresses multiple binary inputs into a smaller number of outputs. The output of a priority encoder is the binary representation of the original number starting from zero of the most significant input bit

highest priority — Lowest Priority

Y_3	Y_2	Y_1	Y_0	A1	A0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

Fig: 4-to-2 Priority Encoder



$Y_3Y_2 \backslash Y_1Y_0$	00	01	11	10
00	X	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Y_2

Y_3

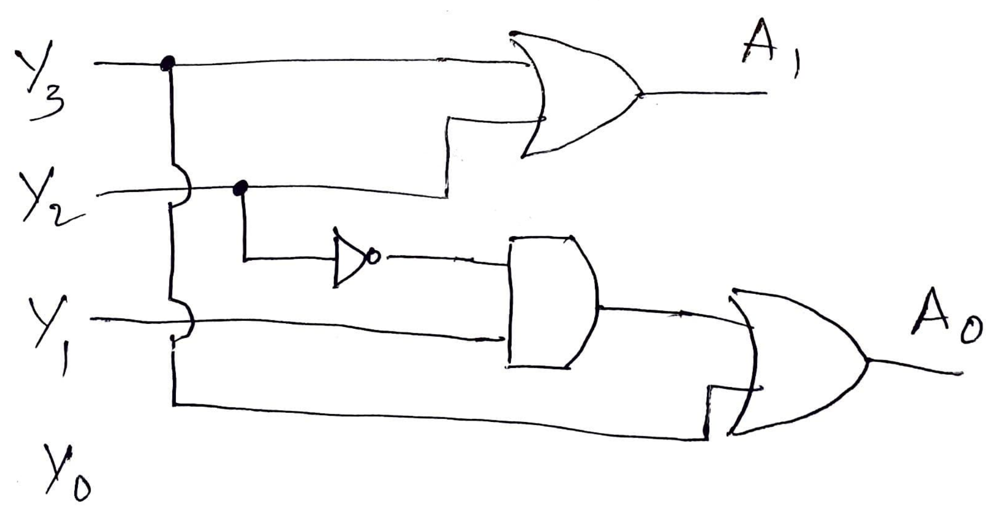
$$A_1 = Y_3 + Y_2$$

$\bar{Y_2}Y_1$

$Y_3Y_2 \backslash Y_1Y_0$	00	01	11	10
00	X	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

Y_3

$$A_0 = Y_3 + \bar{Y_2}Y_1$$



<u>Uses of Encoders</u>

1. To translate Decimal values to the binary
2. Priority Encoders is used for detecting interrupts in microprocessor application

Decoder

A decoder is a combinational circuit that converts n lines of input into 2^n lines of output

Let's take an example of 2-to-4 line decoder

X	Y	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$D_0 = \overline{X}\,\overline{Y}$

$D_1 = \overline{X}Y$

$D_2 = X\overline{Y}$

$D_3 = XY$

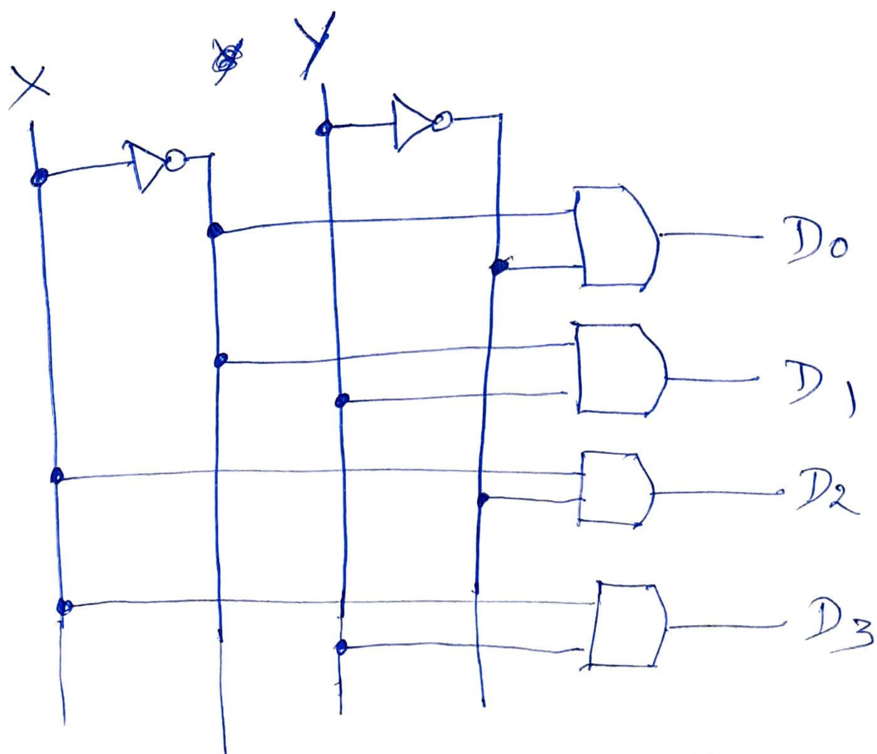


Figure: 2-to-4 Line Decoder

Binary Coded Decimal to Decimal Decoder

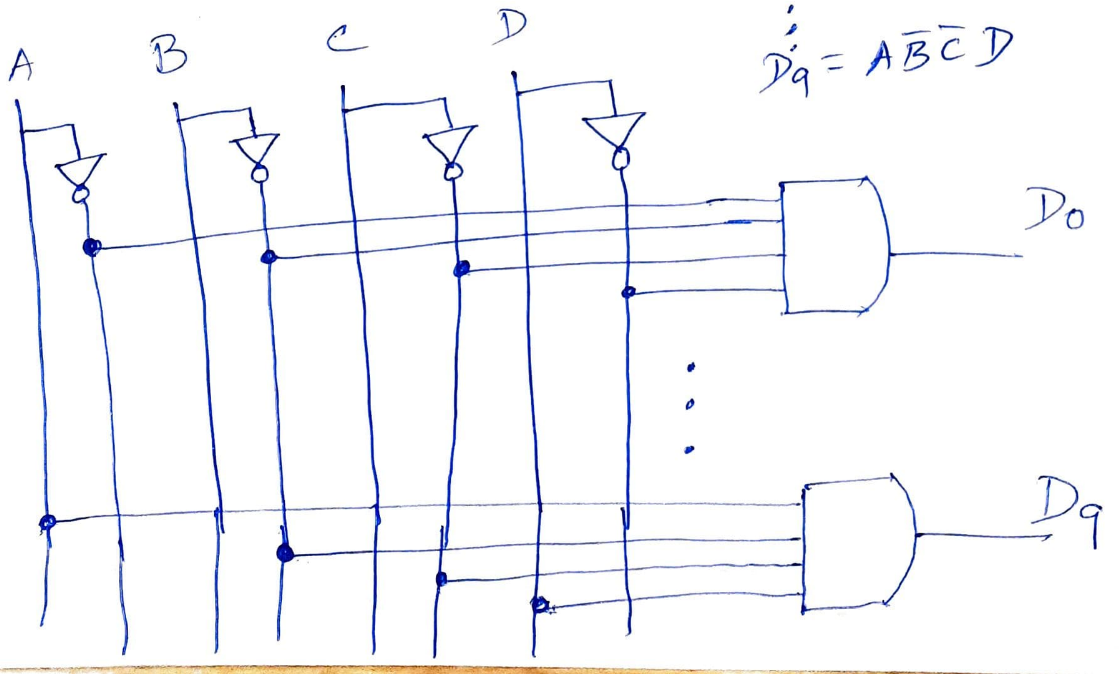
Decoders are used to get the decimal digit corresponding to a specific input combination

	A	B	C	D	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9
0 →	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1 →	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2 →	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3 →	0	0	1	1	0	0	0	1	0	0	0	0	0	0
4 →	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5 →	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6 →	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7 →	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8 →	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9 →	1	0	0	1	0	0	0	0	0	0	0	0	0	1

$D_0 = \bar{A}\bar{B}\bar{C}\bar{D}$

$\vdots$

$D_9 = A\bar{B}\bar{C}D$



Applications of Decoders

1. Binary to Decimal Decoder

Decoders are used to get the decimal digit corresponding to a specific input combination

2. Address Decoders.

Decoders is widely used to decode the particular memory location in the computer memory system.

3. Instruction Decoder

Decoders are used to decode the program instructions in order to activate the specific control lines such that different operations in the ALU of the CPU are carried out.

Decoder

Half Adder

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

$Sum = \sum m(1,2)$

$Carry = \sum m(3)$