

Radix _4 Modified Booth Multiplier

Basic Algorithm mechanism
of Radix - 4 Booth multiplier

(Step 1)

- We have to provide the proper signed format of multiplier (Q) and multiplicant (M).

Ex. Suppose, we want to multiply $25 \times (-27)$

$$(25)_{10} \rightarrow 00011001 \quad (M's complement form)$$

$$(-27)_{10} \rightarrow 11100101 \quad (27 \text{ which represents actually } -27)$$

- We have to select or provide

$$Q = 00011001$$

$$M = 11100101$$

(Step 2)

- In this step, we are studying about - Multiplier's bit pattern. According to the special bit pattern of a particular group of Multiplier (Q), we decide specific Arithmetic operation will be performed with multiplicant (M).

→ Here, we are going to design (8×8) - Radix-4 Booth Multiplier.

$$Q = Q_7 Q_6 Q_5 Q_4 Q_3 Q_2 Q_1 Q_0 \quad [8-1]$$

This is initial bit it is always keeping in 2's complement state

$$Q = (Q_7 Q_6 Q_5 Q_4 Q_3 Q_2 Q_1 Q_0 Q_1)$$

Here total 4-group is possible and each group contains 3-bit at a time. So, Every group has 8 possible combination according to that specific combination particular Arithmetic operation should be performed with multiplicant.

This is some below in table:

	Q_{i+3}	Q_{i+1}	Q_i	$Q_i M +$
0	0	0	0	$0 \times M$
1	0	0	1	$1 \times M$
2	0	1	0	$1 \times M$
3	0	1	1	$-1 \times M$
4	1	0	0	$-2 \times M$
5	1	0	1	$-1 \times M$
6	1	1	0	$-1 \times M$
7	1	1	1	$0 \times M$

→ Specifically it is signified that if the bit sequence is 000 then corresponding we have to multiply 0 @ the multiplicant (M).

Similarly corresponding all the product is the same below.

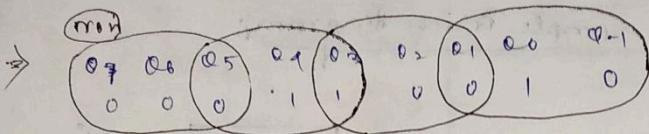
(Example)

Let's example for better understanding.

$$Q = (25)_{10} \rightarrow 00011001$$

$$M = (-27)_{10} \rightarrow 11100101$$

Q_{-1} (bit) is assigned 0, always or initially.



Here M is the multiplier which contains 8 bit.

$$\begin{matrix} M_7 & M_6 & M_5 & M_4 & M_3 & M_2 & M_1 & M_0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{matrix}$$

$$\text{Q7 Q6 Q5} \Rightarrow 1 \times M = \boxed{11100101} \Rightarrow M_7 M_6 M_5 M_4 M_3 M_2 M_1 M_0 \\ (= 27)$$

$$\text{Q3 Q2 Q1} \Rightarrow -2 \times M = \boxed{\begin{matrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{matrix}} \Rightarrow M_7 M_6 M_5 M_4 M_3 M_2 M_1 M_0 \\ (= -27)$$

$$\text{Q5 Q4 Q3} \Rightarrow +2 \times M = \boxed{\begin{matrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 2 \times (-27) \\ = -54 \end{matrix}} \Rightarrow M_7^2 M_6^2 M_5^2 M_4^2 M_3^2 M_2^2 M_1^2 M_0^2$$

$$\text{Q2 Q1 Q0} \Rightarrow 0 \times M = \boxed{00000000} \Rightarrow M_7^3 M_6^3 M_5^3 M_4^3 M_3^3 M_2^3 M_1^3 M_0^3$$

Now, we have to perform the basic addition operation, which is successively able to give the correct result.

	M_7^0	M_6^0	M_5^0	M_4^0	M_3^0	M_2^0	M_1^0	M_0^0
\oplus	M_7^0	M_6^0	M_5^0	M_4^0	M_3^0	M_2^0	M_1^0	M_0^0
\oplus	M_7^2	M_6^2	M_5^2	M_4^2	M_3^2	M_2^2	M_1^2	M_0^2
\oplus	M_7^3	M_6^3	M_5^3	M_4^3	M_3^3	M_2^3	M_1^3	M_0^3
	R_5	R_4	R_3	R_2	R_1	R_0	P_1	P_2

+ finish
now
final y/
result

$+$	11111111	11100101	00000000	111111010100101
	1111110000110101010			
	11111100010101010			
	000000000000000			
	11111110101011101			

1111111

$$\begin{array}{r} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \hline 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array}$$

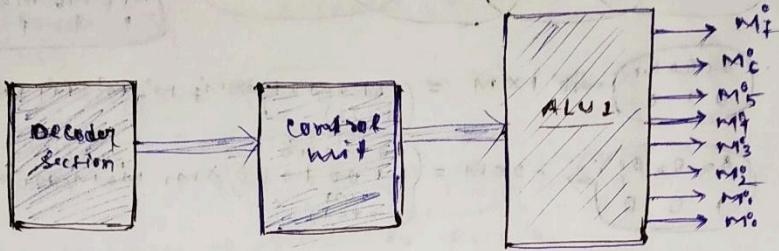
So, the Result is

$$1111111010100101 (17bit)$$

which represents or corresponds decimal value = -675.

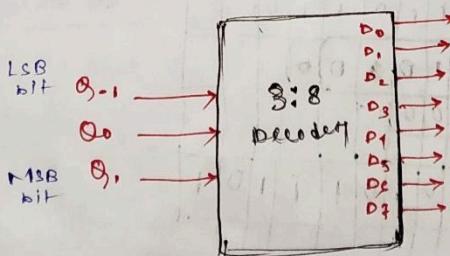
circuit diagram - portion

→ 1st our aim is produced ($M_7^o M_6^o M_5^o M_4^o M_3^o M_2^o M_1^o M_0^o$)
 (the simple Block diagram)



- (A) Now we will discuss about ALU1. int+min part
 We will perform 8 arithmetic operation according to the specific binary sequence. So, there are so many control bits available like x, y, z, cin, cout. We will discuss of ALU1 which will actually decide which operation should be done to which arithmetic operation. It should be controlled by entire whole things should be controlled by control unit section. Decoder section helps to decide the specific binary pattern of the multiplexer (Q). This is the basic operating mechanism of this section.

Decoder section

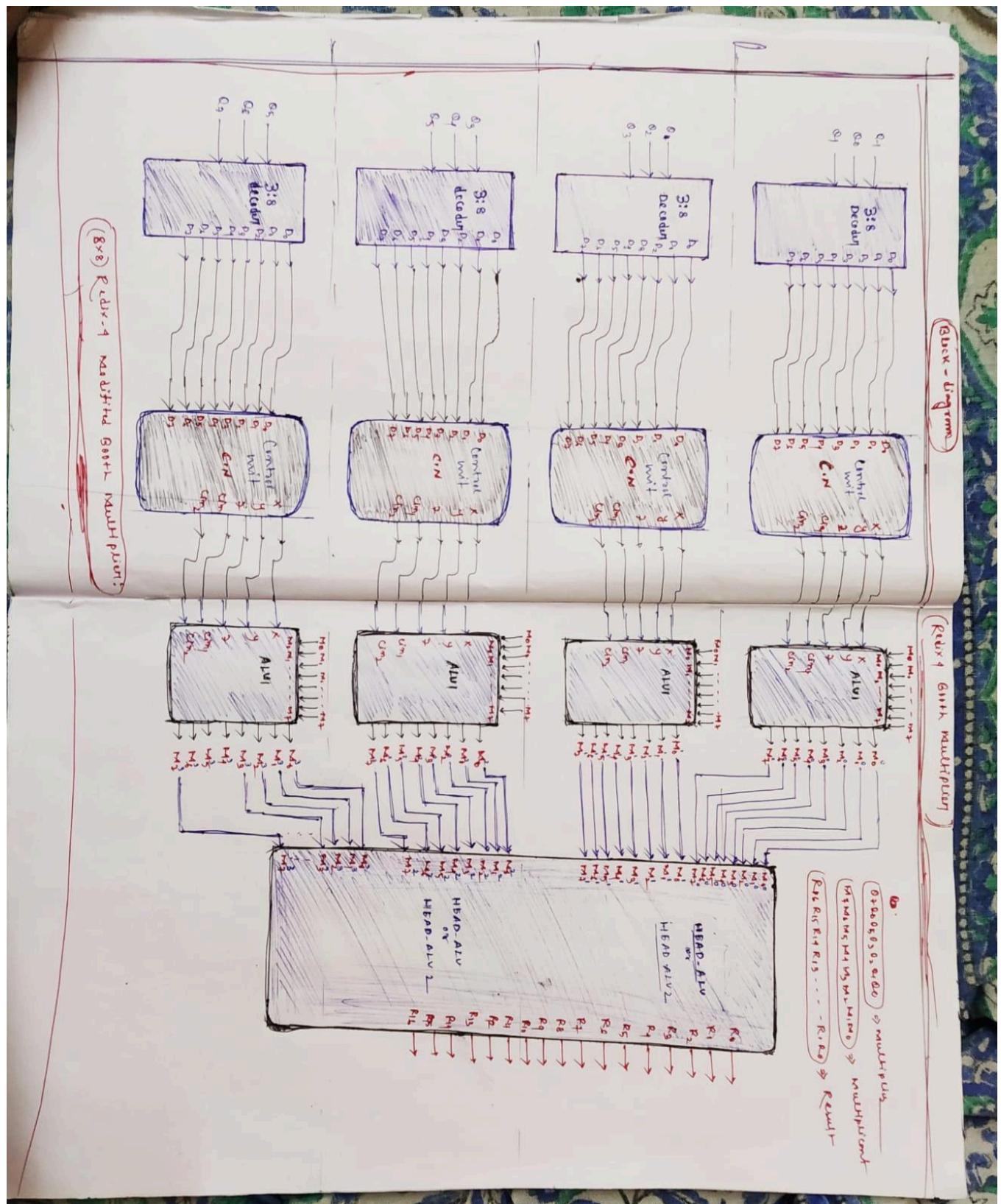


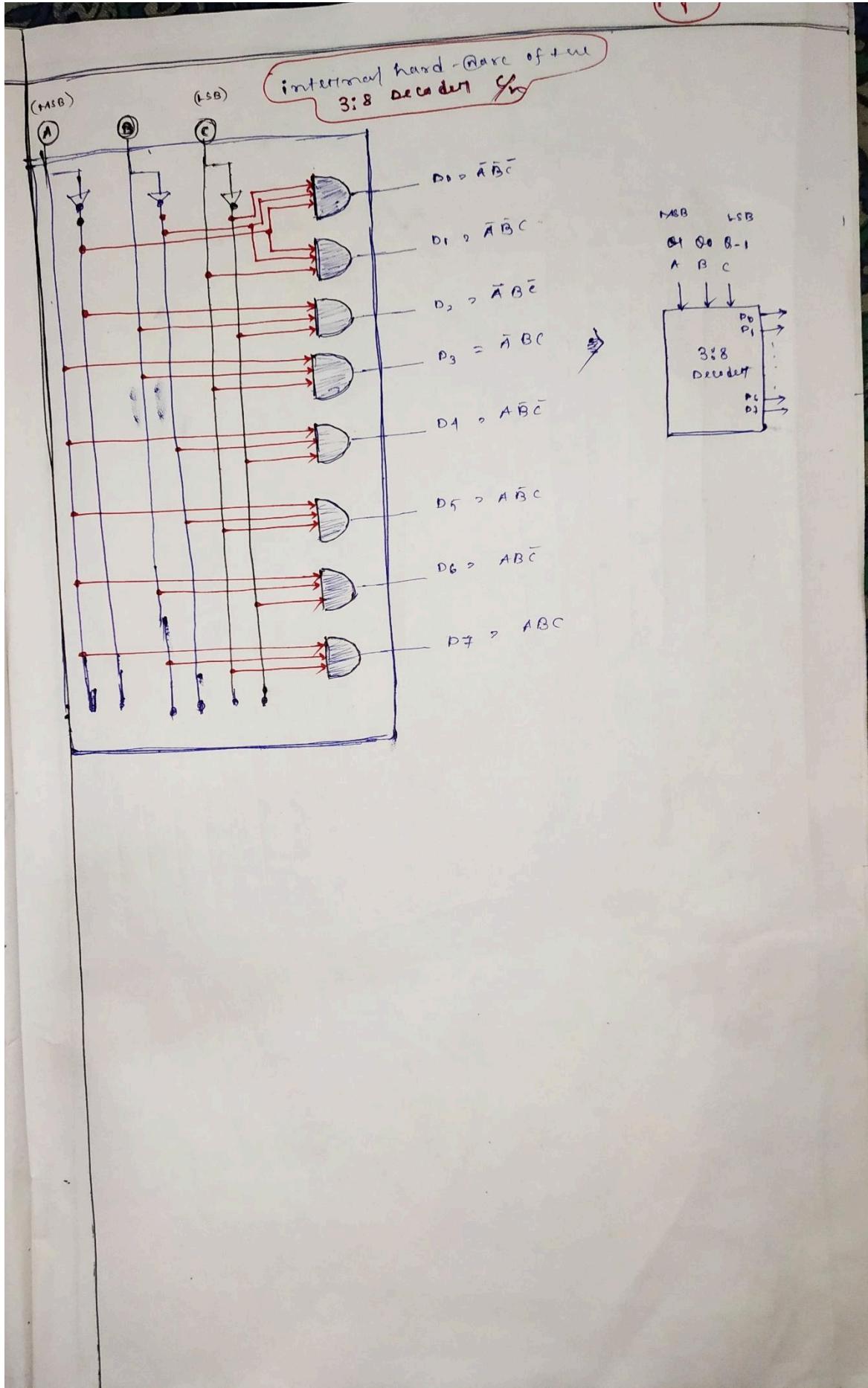
According to the (3:8) decoder mechanism. When $Q_{-1} = 1$, once any of Q_P is active then all the Q_P should be also active.

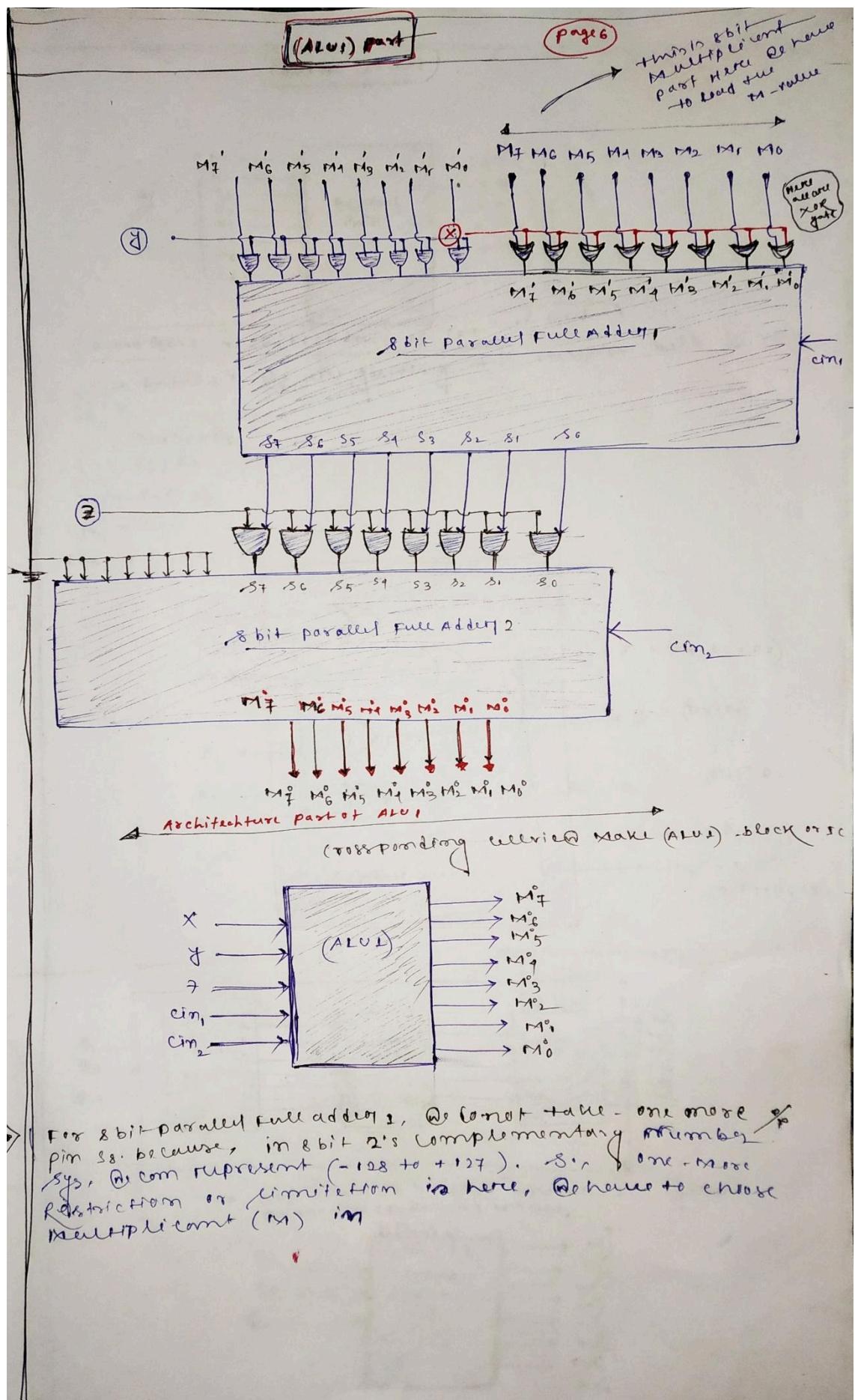
Ex. $Q_{-1} = 0$
 $Q_0 = 0$
 $Q_1 = 1$

$$Q_1 \oplus Q_{-1} \Rightarrow Q_1 = 1 \text{ result}$$

then all the $A_D = 0$
 $i = 0, 1, 2, 3, 5, 6, 7$







page 7

truth-table

(all these pins are pining from Deuden's 60)

Table 2

1/p Pins of two
(ALU) HardWare

	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	X	Y	Z	Cin ₁	Cin ₂	
R ₀	1	0	0	0	0	0	0	0	X	X	0	0	0	$\rightarrow 0 \times M$
R ₁	0	1	0	0	0	0	0	0	0	0	1	0	0	$\rightarrow 1 \times M$
C ₂	0	0	1	0	0	0	0	0	0	0	1	0	0	$\rightarrow 1 \times M$
C ₃	0	0	0	1	0	0	0	0	0	1	1	0	0	$\rightarrow 2 \times M$
C ₄	0	0	0	0	1	0	0	0	1	1	1	1	1	$\rightarrow -2 \times M$
C ₅	0	0	0	0	0	1	0	0	1	0	1	0	1	$\rightarrow -1 \times M$
C ₆	0	0	0	0	0	0	1	0	1	0	1	0	1	$\rightarrow -1 \times M$
C ₇	0	0	0	0	0	0	0	1	X	X	0	0	0	$\rightarrow 0 \times M$

Once we have to multiply 2 with M (multiplicant)
here, we don't use any shifting mechanism.
just M is added with it self $(M+M) = 2M = 2 \times M$

Again, once we have to multiply by -2 with M then
we are doing $-M + -M = -2M$ operation

Let's discuss about C₆ column.

D₀ D₁ D₂ D₃ D₄ D₅ D₆ D₇
0 0 0 0 0 0 1 0

→ This bit sequence
signifies D₆ pin is
off & rest seems
we have to perform
 $-1 \times M$ operation

Now, corresponding control pin decides $(-1 \times M) - 1$ operation

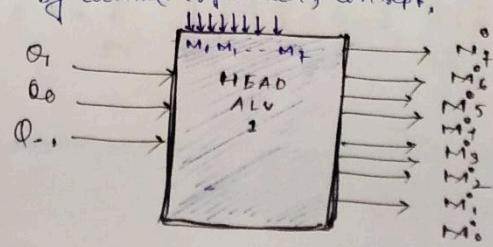
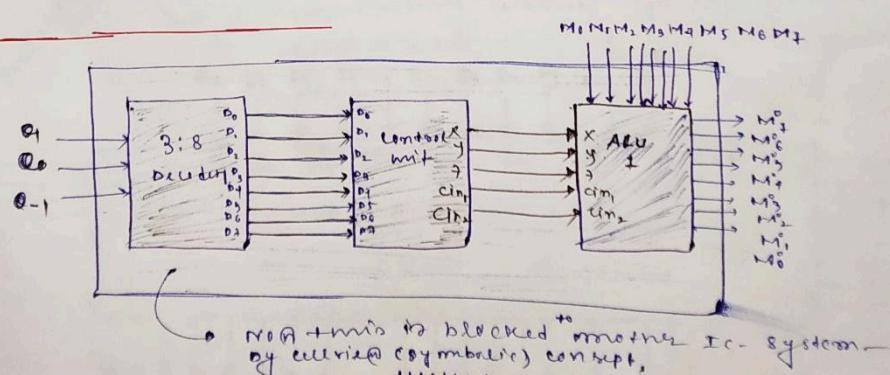
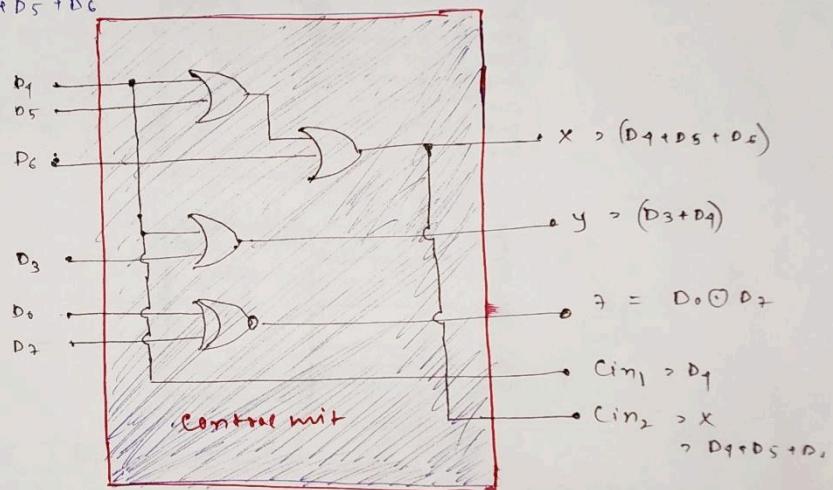
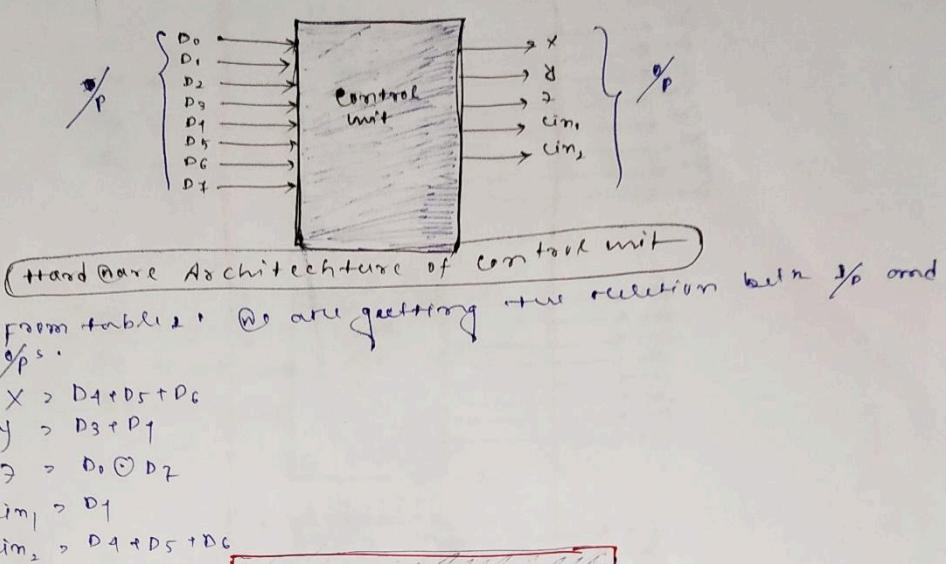
→ its corresponding control pin:
X Y Z Cin₁ Cin₂

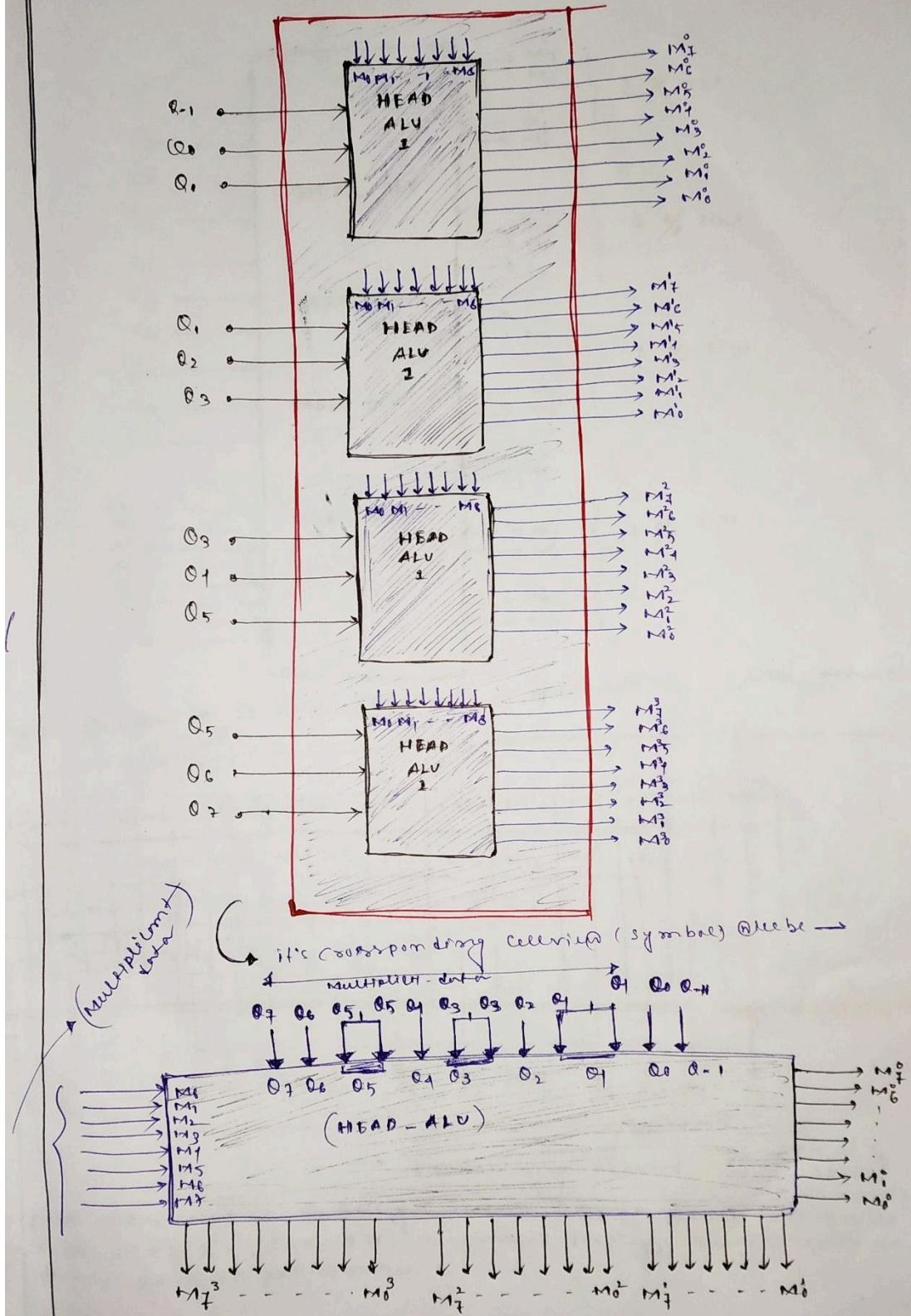
∴ X = 1
Y = 0
Z = 1
Cin₁ = 0
Cin₂ = 1

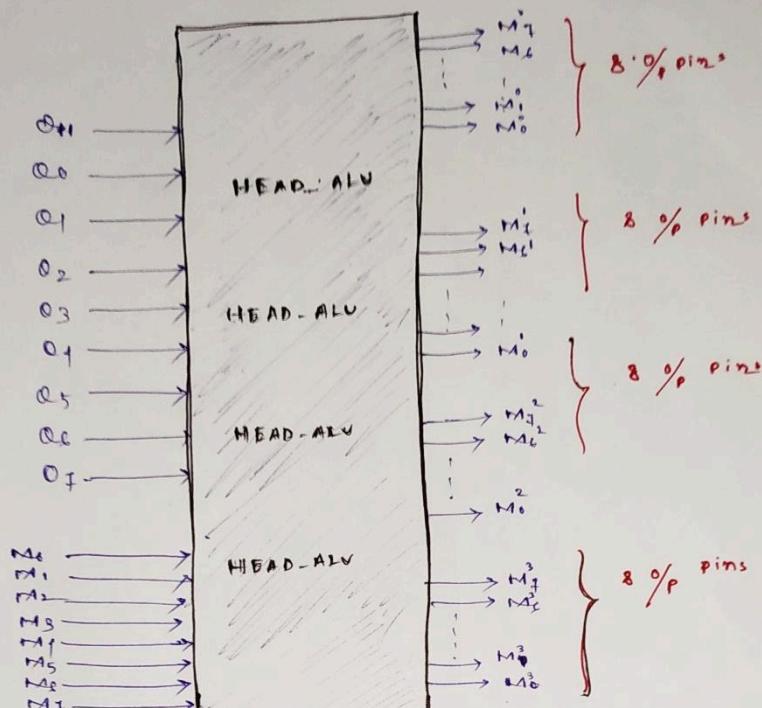
→ Apply this logic to all
Architecture part of
ALU
→ Now you will see finally
 $(-1 \times M)$ operation has been performed

Control circuit

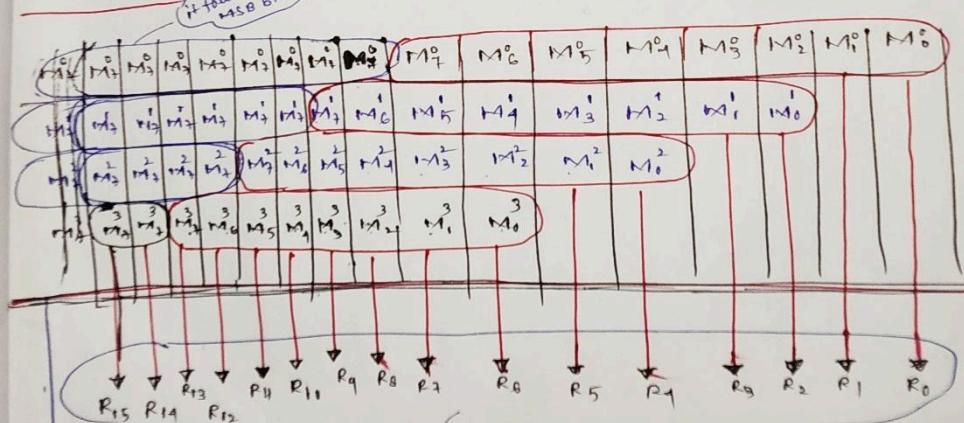
(Page 7)







(a) We have to do the addition operation like that



→ This is my actual \oplus (final 8bit)
from (HEAD-ALU) IC - @ 8@U get total 32% which is required
then all 16% should be arranged in Fig 2 term. Now to
do two Addition performance.

(Page 10)

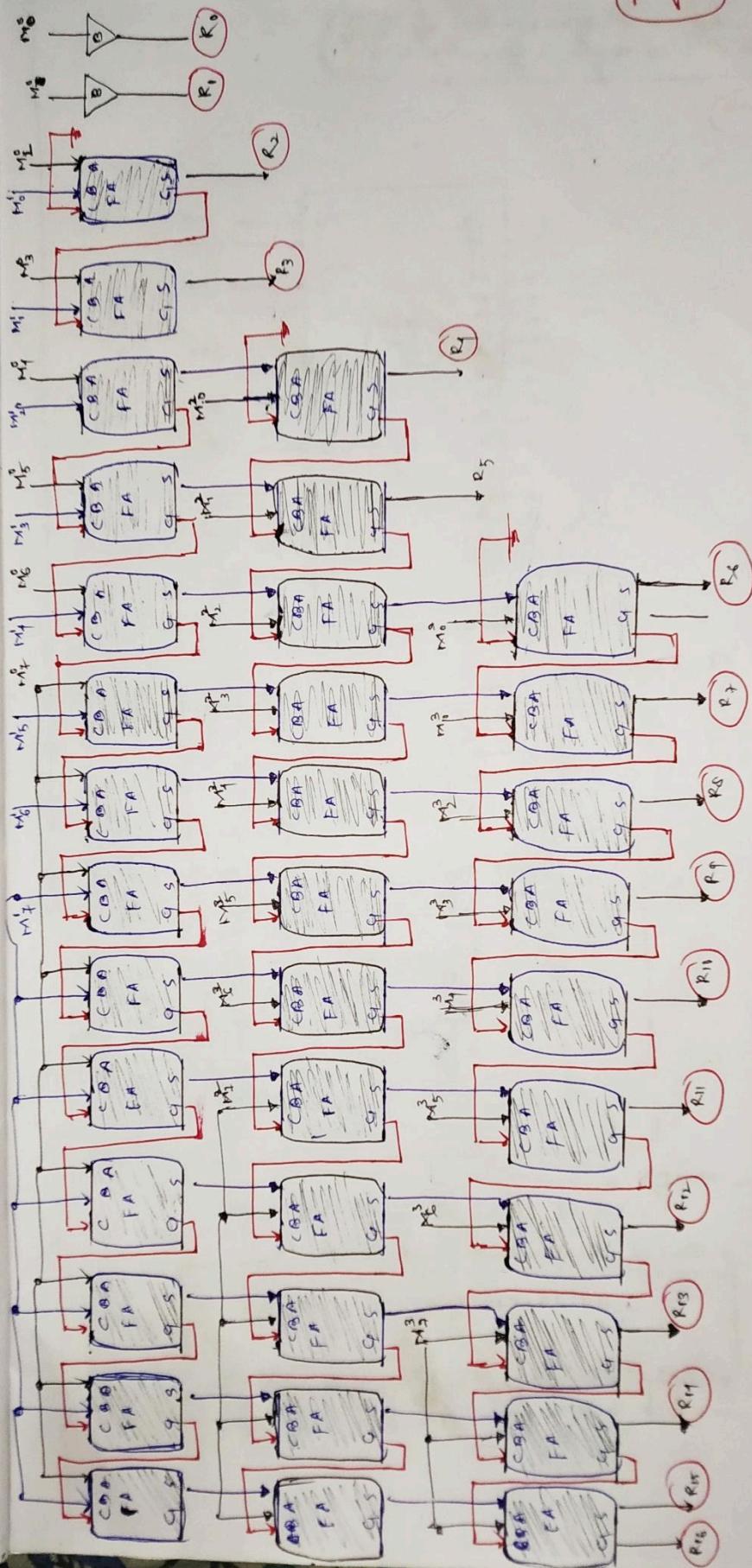
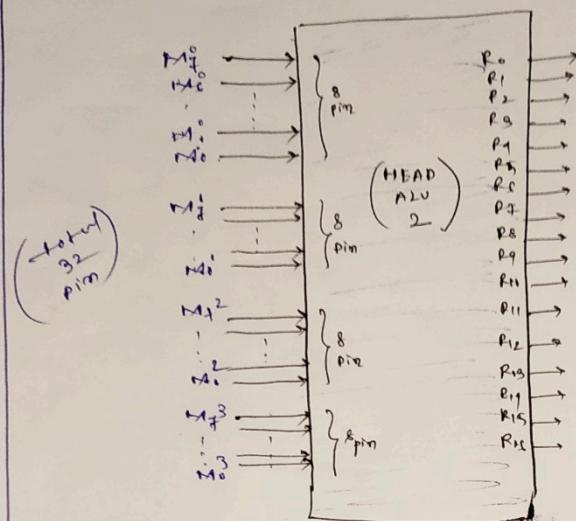


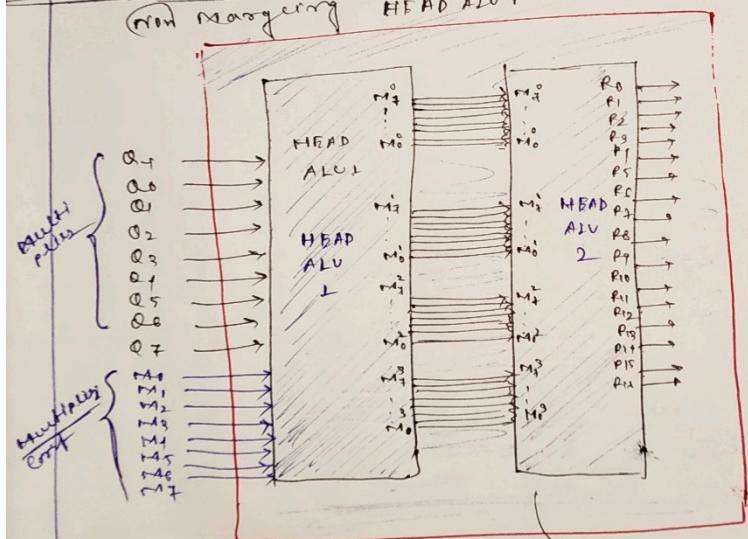
Fig 2

(Now we are going to design
two cell width (symbolic) of HQ2)

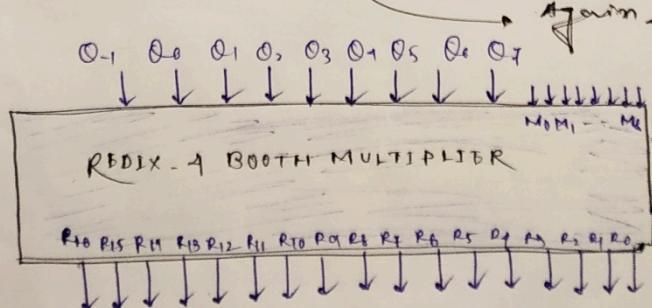
(Page 11)

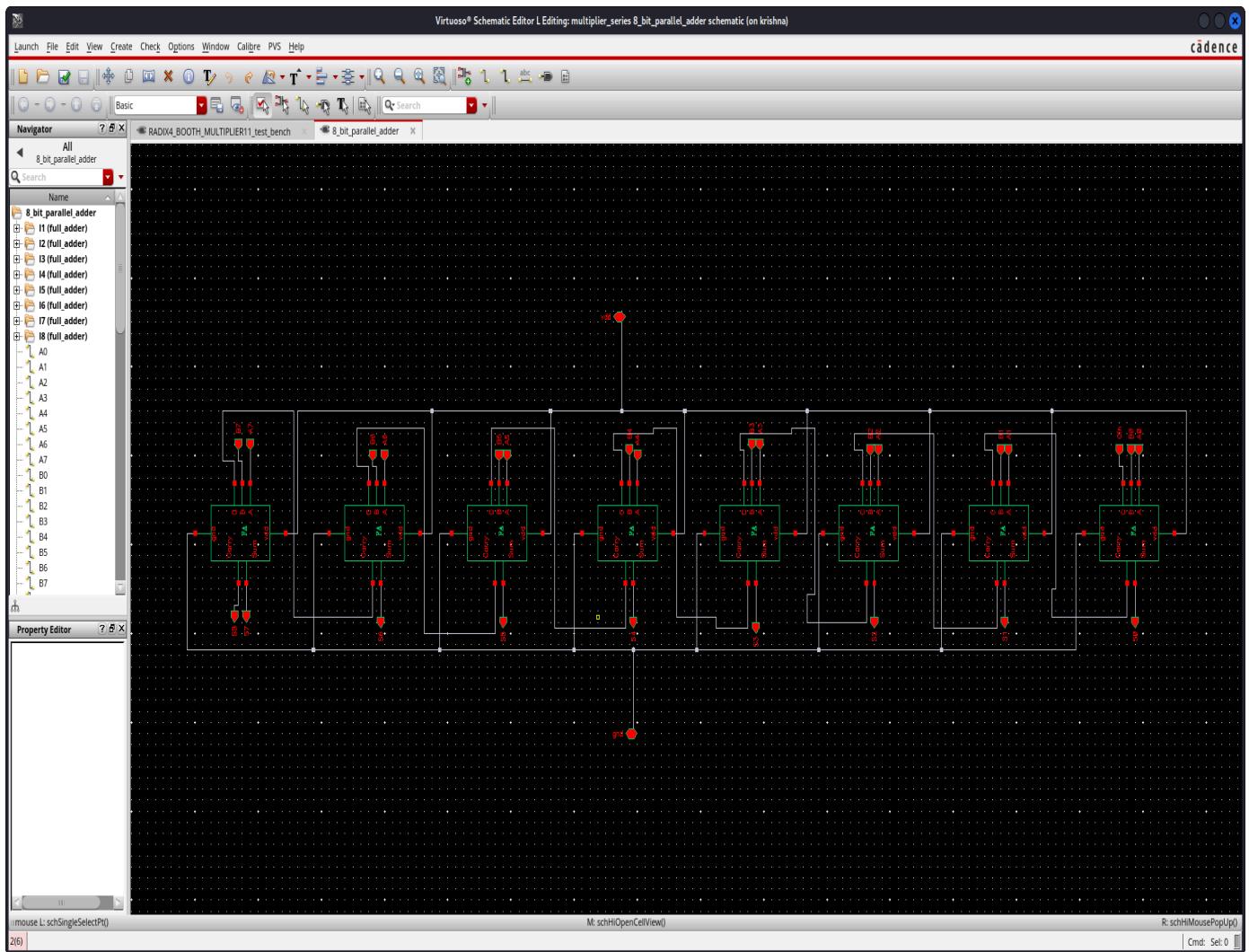


(Now managing HEAD ALU 1 and HEAD ALU 2 IC

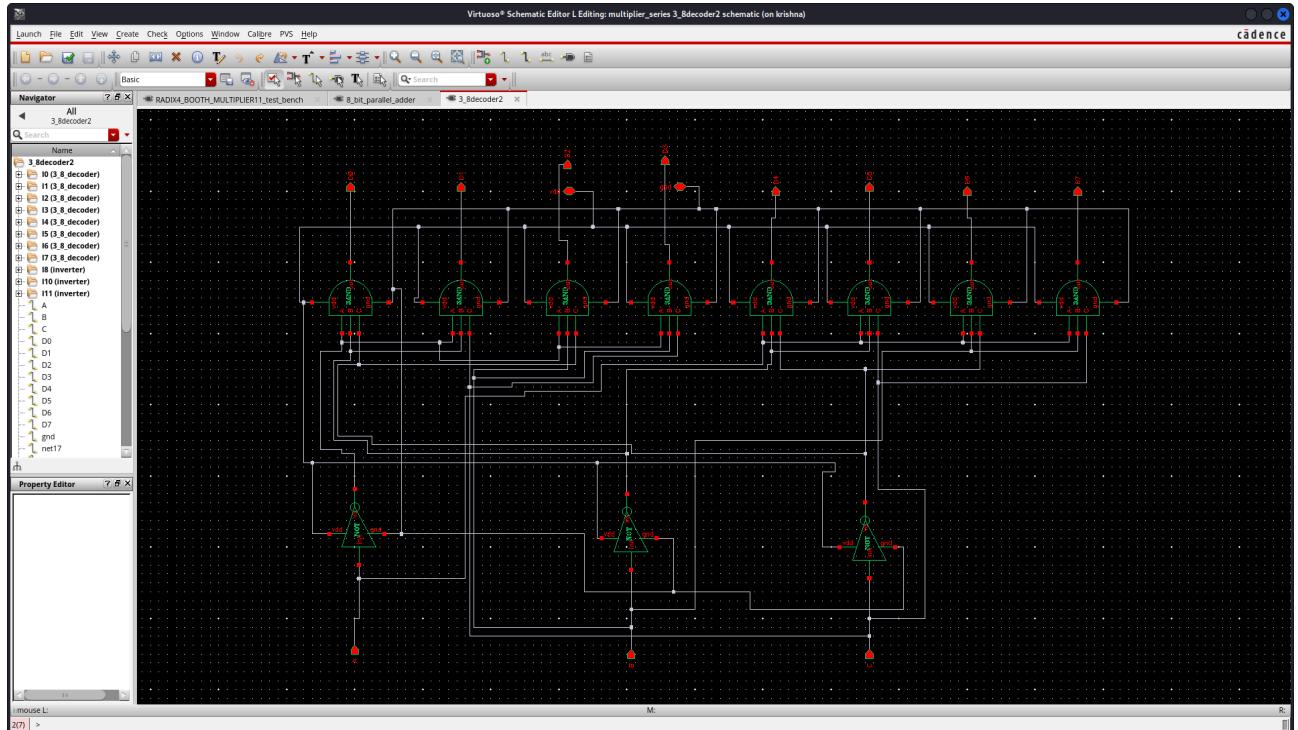


Again - converted it
into another
IC by using (symbol)



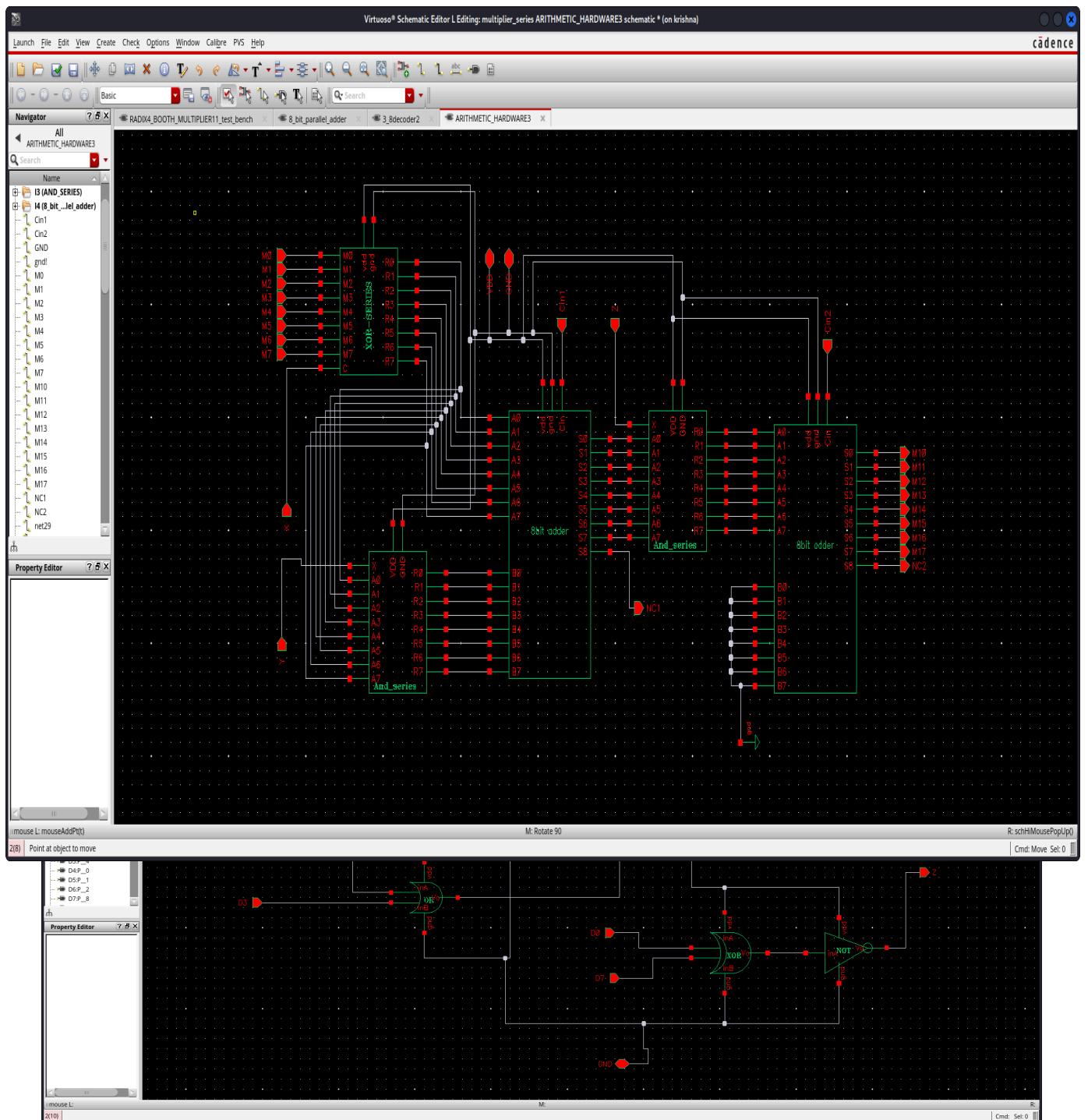


Internal circuit of 8-bit parallel adder

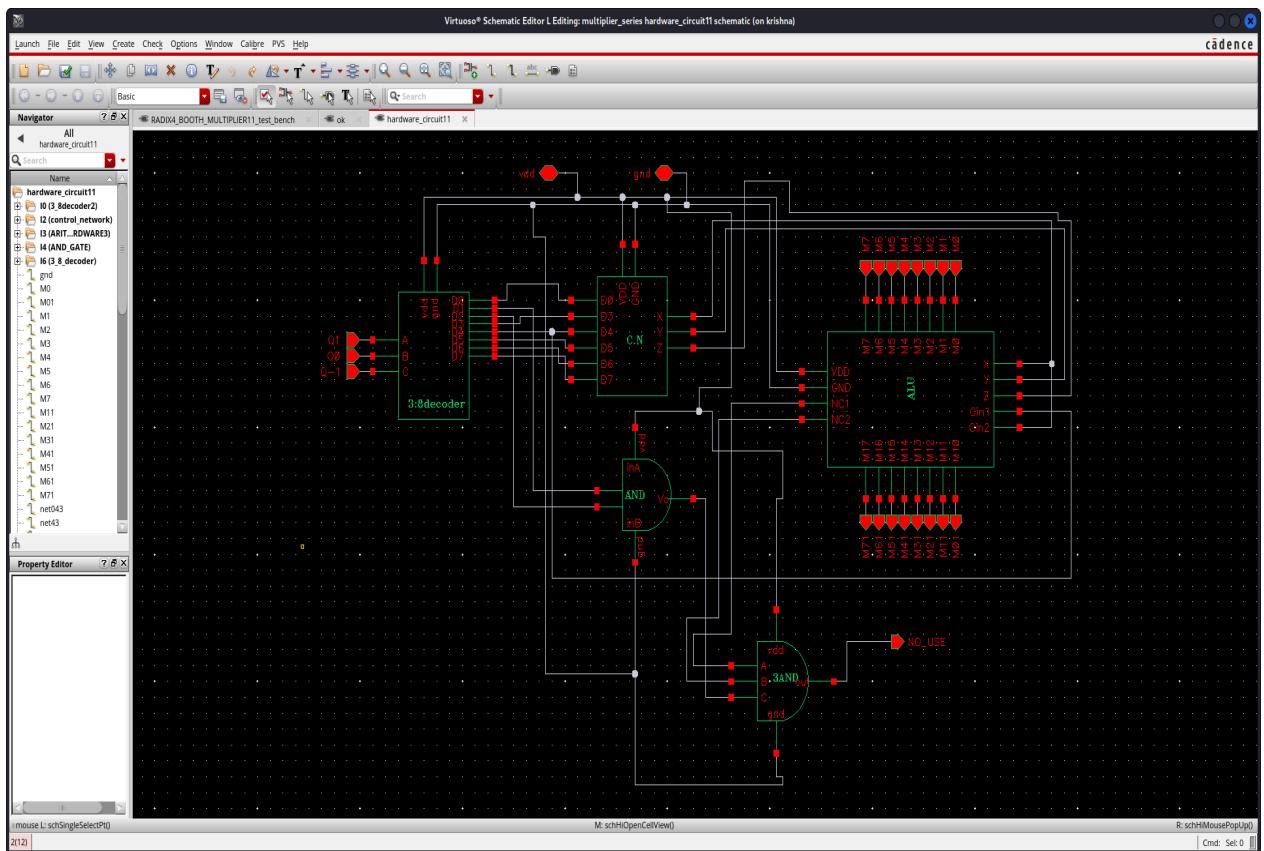


Internal hardware of 3:8 Decoder

Internal circuit of control unit

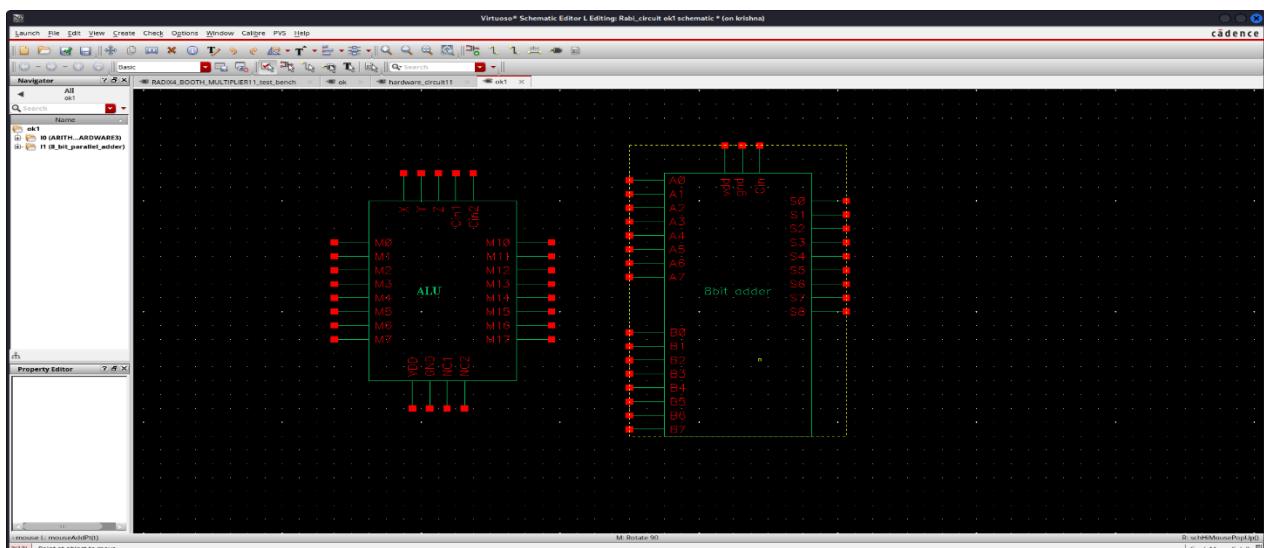


Interface of ALU 1 network

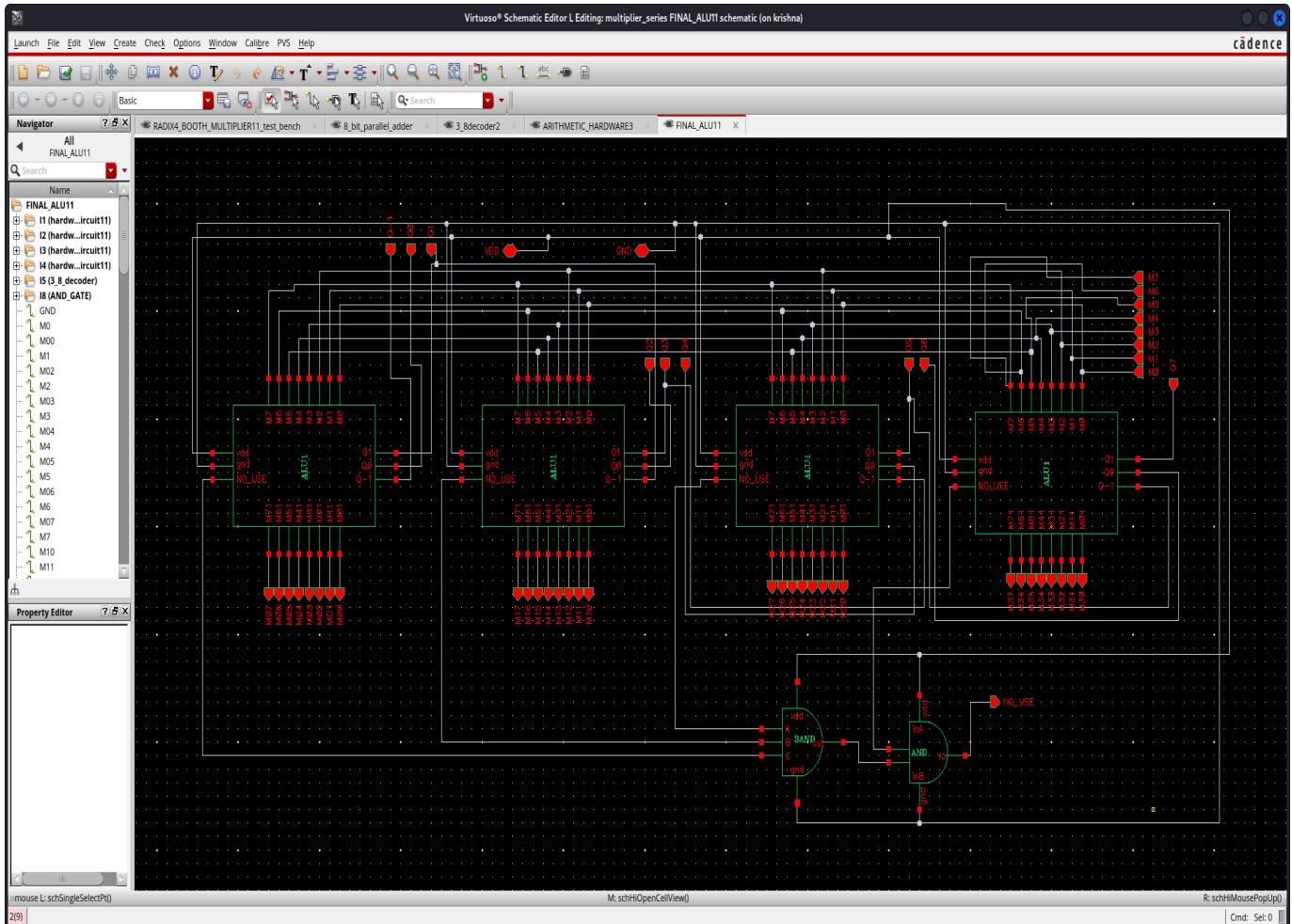


Internal networking of single ALU1 interface

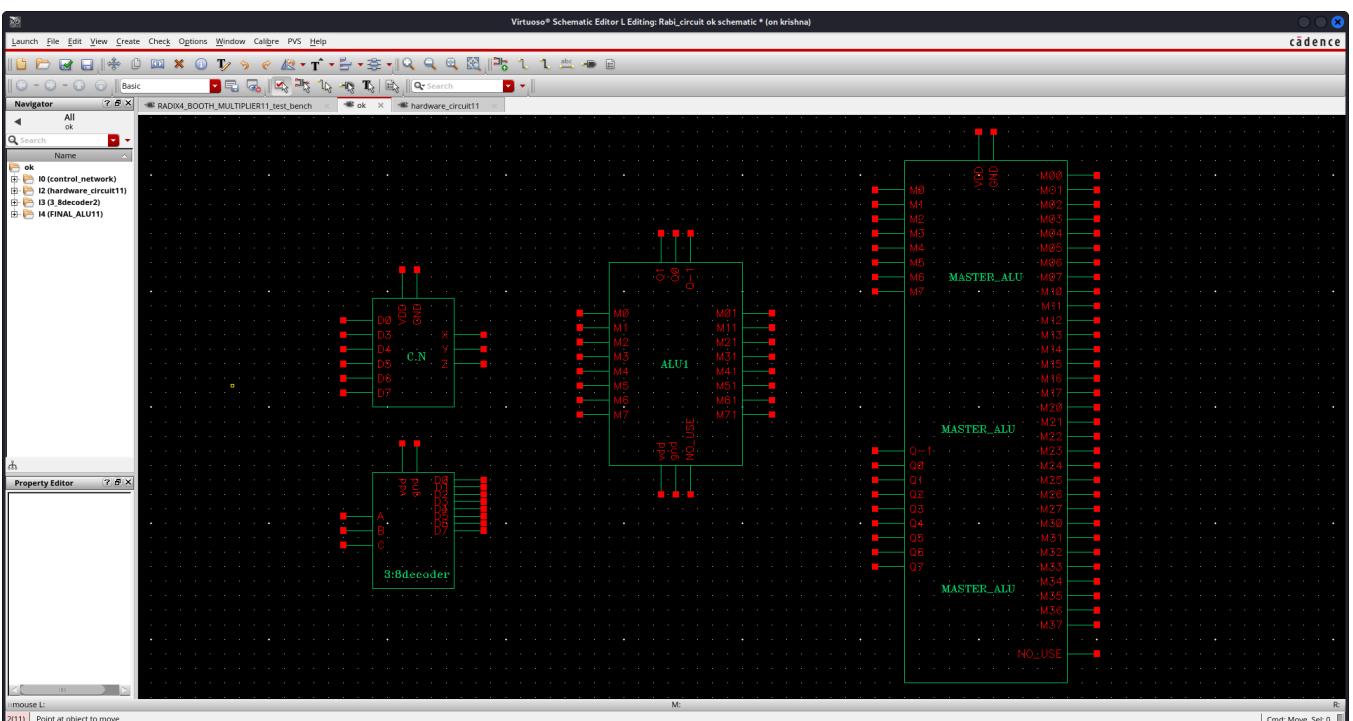
In this circuit diagram, according to the binary sequence of



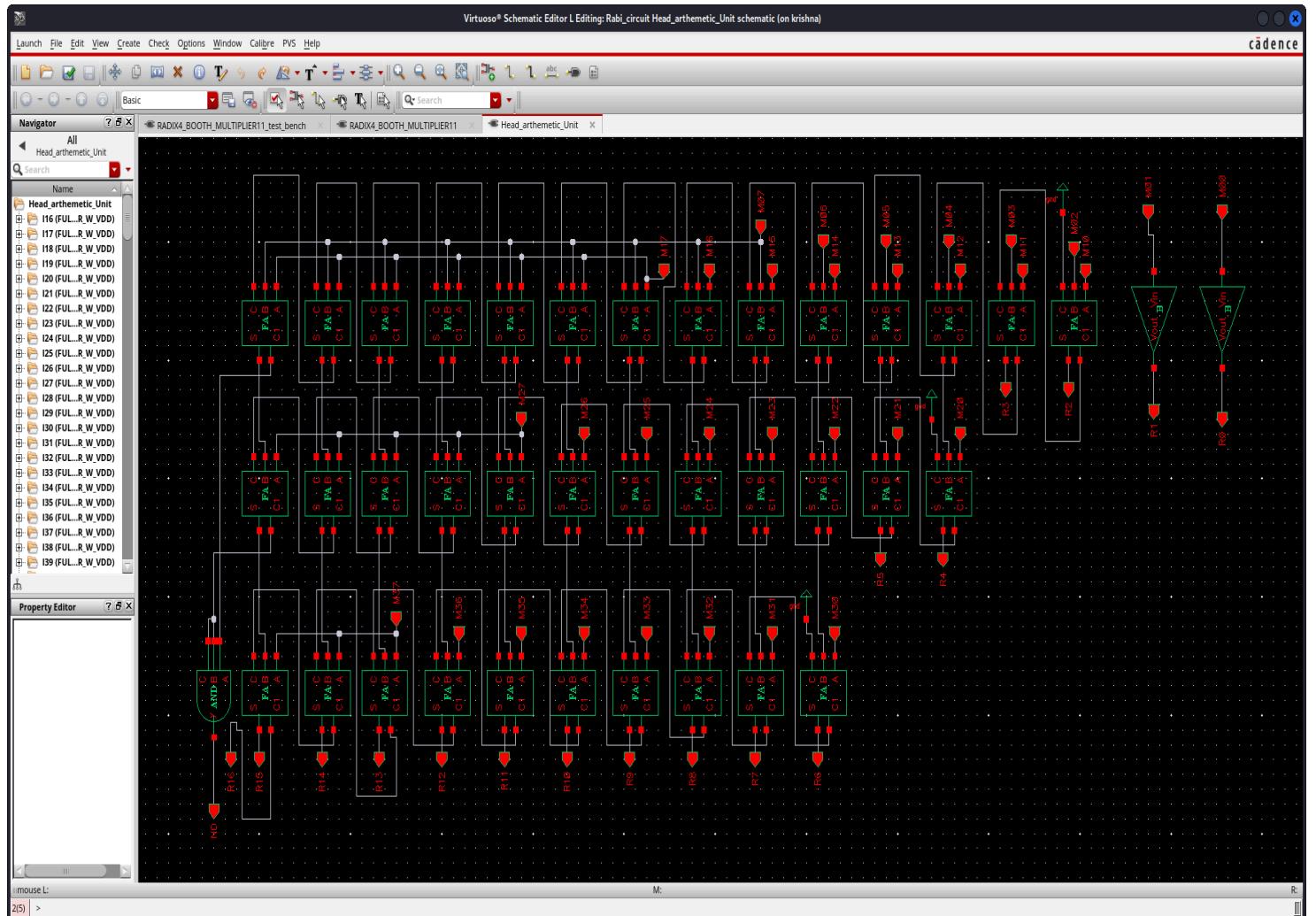
Q-1,Q0,Q1 cross bonding specified arithmetic operation
should be performed through ALU1_IC.



Symbolic representation of ALU1 and 8 bit parallel adder

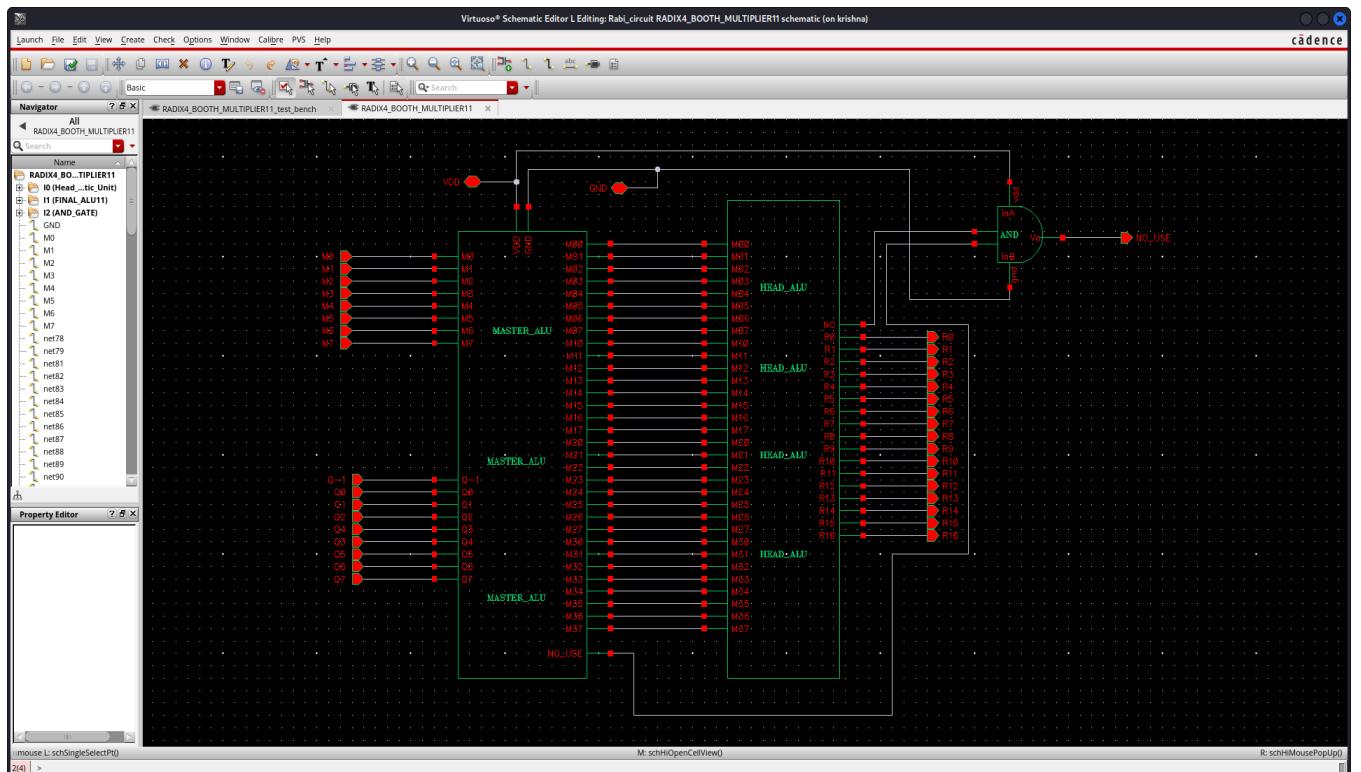
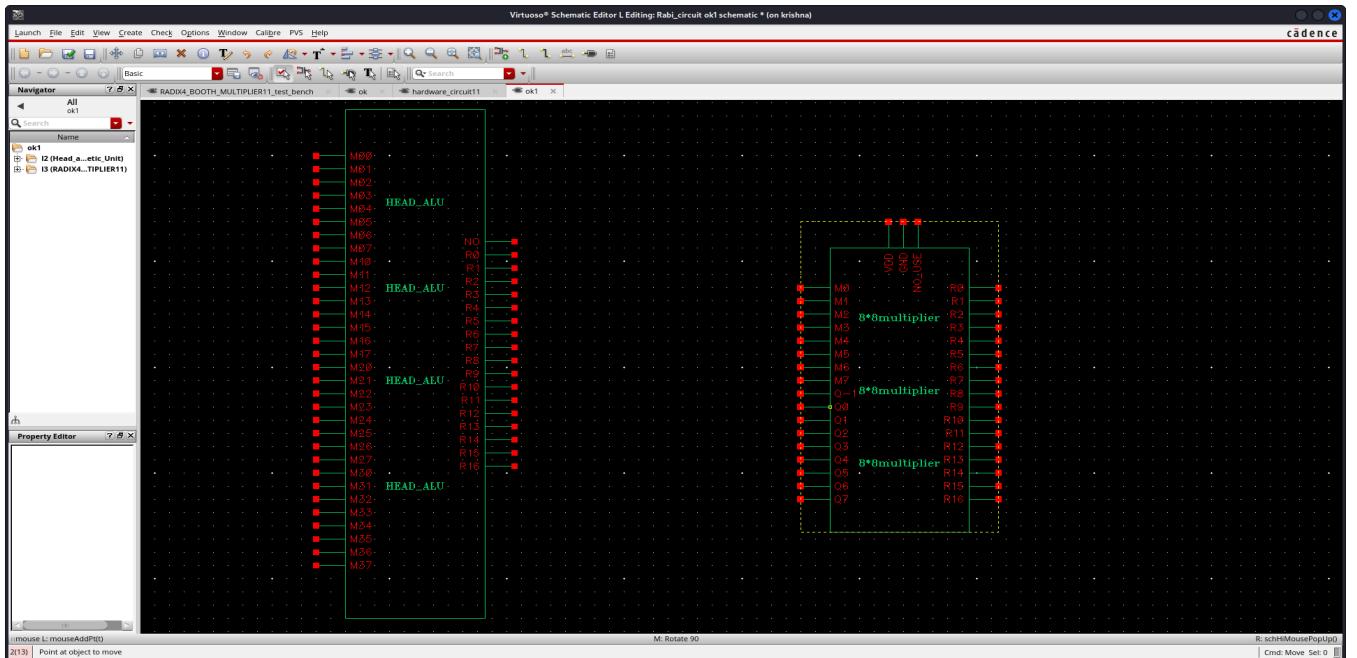


Internal circuit interface of master ALU



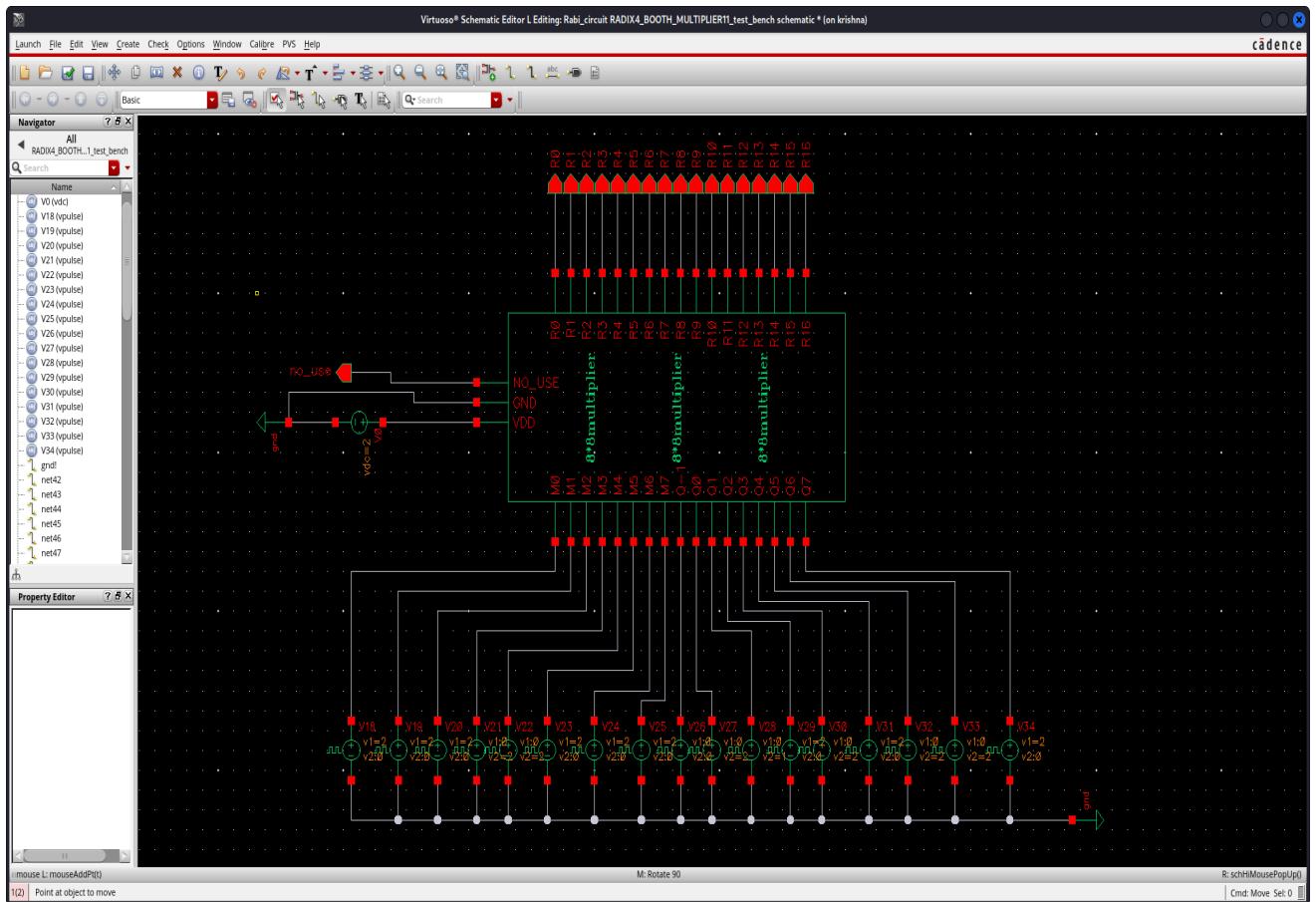
Symbolic form of Maser ALU and ALU1, 3:8 decoder and control unit. Here Master ALU=HEAD ALU 1 both are same.

Internal circuit interface of HEAD_ALU 2 or HEAD ALU



This is the symbolic form of head alu 2 and 8*8

**Joining interface of master alu and head alu
which make 8*8 radix 4 booth multiplier.**



Testing interface of this multiplier