EECT 7325 VLSI DESIGN

256 WORD DUAL SRAM DESIGN

Area of Memory cell	**3.42um x 2.38 um = 8.1396 squm**
Aspect Ratio of Memory cell	**1.436**
Memory array area	**102.860 x 151.210 = 15553.4606 squm**
Memory area per bit	**6.459**
Total Memory area	**213.520 x 170.231 = 36347.723 squm**
Total Memory area per bit	**17.74**
Aspect Ratio of total memory	**0.797**
Worst case Write time	**For '0' = 451 ps For '1' = 467 ps**
Worst case Read time	**For '0' = 20.6 ns For '1' = 1.59 ns**

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction to SRAM

A 256 word Dual SRAM (word size is 8 bits) is designed using 130nm CMOS technology. Worst case read and write times are calculated for the output load as 25fF.

An SRAM cell consists of the following blocks:
(a) 8T (Eight transistor) Memory cell
(b) Row decoder
(c) Column decoder
(d) Precharger and clock
(d) Sense amplifier
(e) Write driver

1.2 SRAM Architecture

The SRAM array with the given specifications is made up of 256 words, which is split into 8 words per row, with 32 rows depending on the aspect ratio. Each word is made up of 8 bits. There are 32 word lines, hence we have a 5x32 row decoder.

Number of columns= 64

Number of rows =32

1.3 8T (Eight transistor Memory Cell)

The 8T Memory cell employs two more transistors to access read bitline. The additional two transistors are added to reduce leakage current.

The main purpose of 8T cell, for separate write/read and word signal lines as shown in the figure 1, to separate data storage and data output lines.

Write access to cell occurs through the write access transistors and from write bitlines, wbl and wblb. Read access to the cell is through read access transistors and controlled by the read wordline, rwl. Read bitline, rbl is precharged prior to the read access. The wordline for read is distinct from the write wordline.

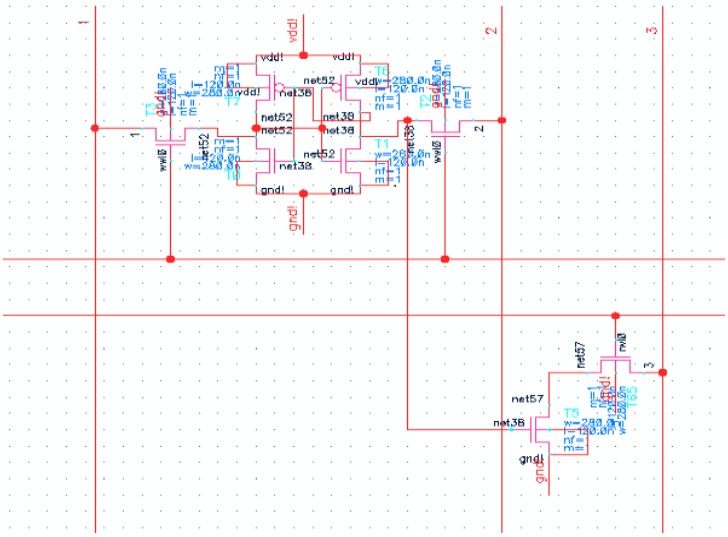

Fig.1 Memory cell schematic

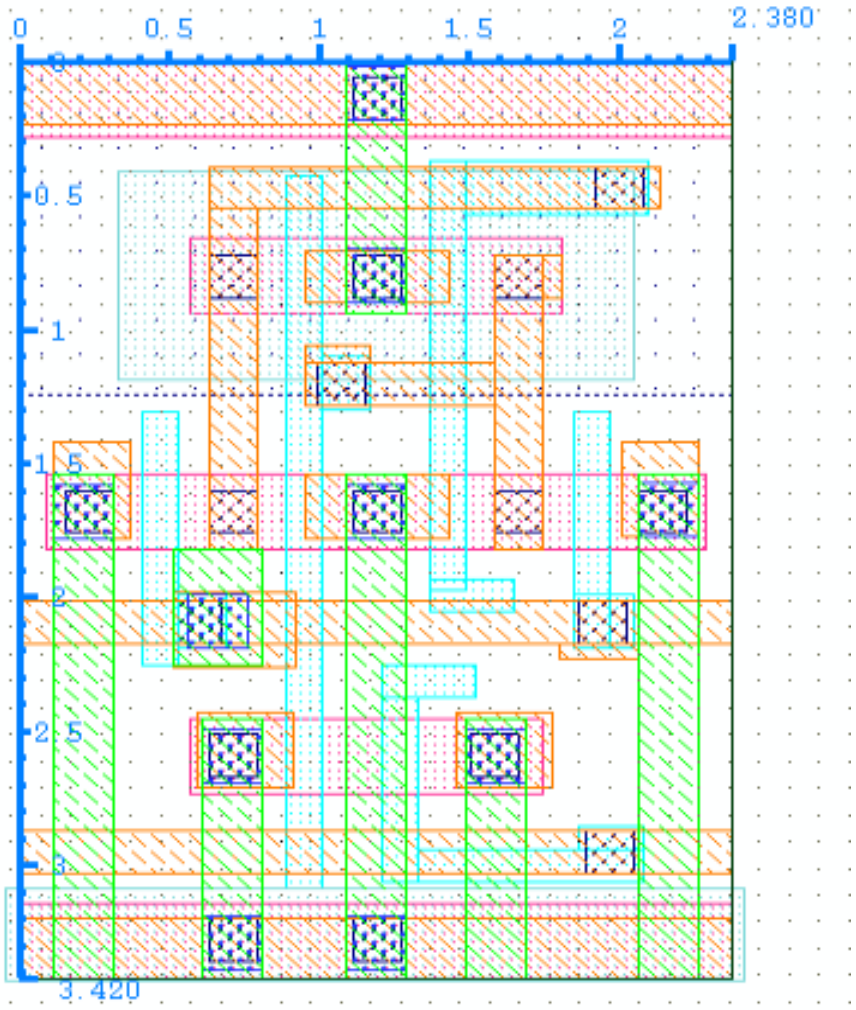

Fig.2 Memory Cell Layout

1.4 Memory array:

The Memory Array consists of 2048 memory cells, placed in 32 rows and 64 columns.

Memory cell:

Width of the cell = 2.38 um
Height of the cell =3.42 um
Area of 8T cell = 8.1396 sq um
Aspect Ratio = 1.4369

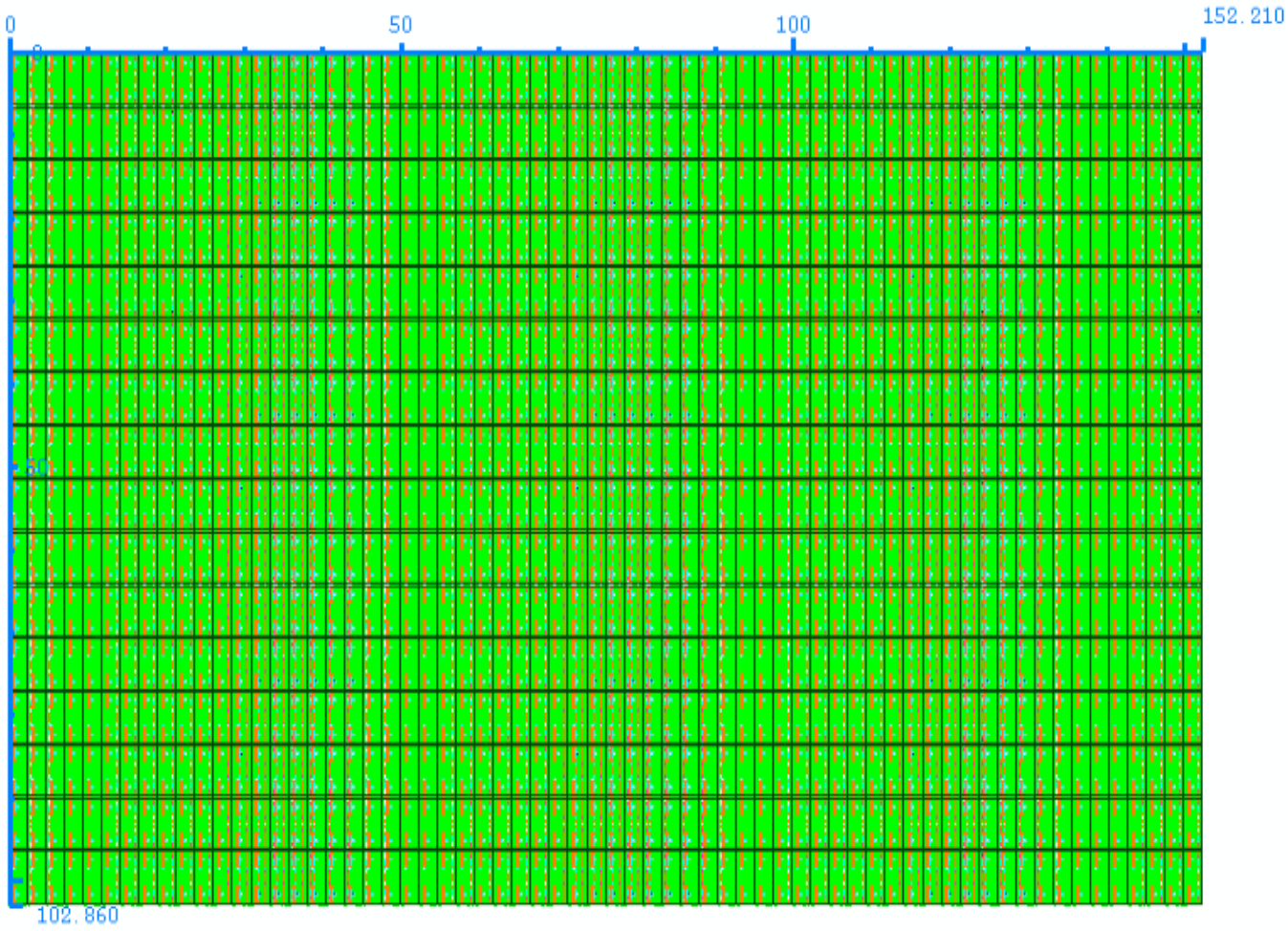

Fig.3 Memory Array Layout

Width = 102.860 um
Height = 151.210 um
Area = 102.860 x 151.210 = 15553.4606 squm
Area per bit = 6.459 squm/bit

2. ROW DECODERS

The row decoder will activate the wordlines, wwl in our memory cell. In this project, we used a pre decoded row decoder. Two row decoders are used, one for read and the other for write.

2.1 Pre Decoded Style row decoder and Sizing of the transistors in the Row decoder

We are having 5 inputs to the row decoder. The design has been implemented by using two 3 input nand gates, one 2 input nor gate and an inverter. By using the necessary information from the single 8T cell, and by using logical effort, we determined the optimum number of stages and then the sizes of the transistors used in the row decoder in the following way.

CPoly = 32 x 2fF x 2x0.28 = 35.84 fF
Cwire = 32 x 2.38 x 0.2fF = 15.232 fF
Cload = Cpoly + Cwire = 51.072 fF = 25.5um

By simulating in Hspice, the Cload was changed to 53.5fF (26.75um)

G for two 3-input NAND gate, inverter and 2-input NOR gate = 125/27
B= 32
H=26.75
From F= GBH= 3962.96
N = $\log_{3.6} F$ = 6 (Considered)
$\hat{f} = F^{\frac{1}{N}}$ = 3.978

2.2 Transistor sizing

Inverter 1(rightmost) = 6.724 um
NAND3= 2.817 um
NOR2= 4.721 um
NAND3= 1.978 um
Inverter4= 1.988 um

The inputs to the row decoder are addr0, addr1, addr2, addr3, addr4 and addr0b, addr1b, addr2b, addr3b, addr4b. The outputs of row decoders are fed to pass transistors to activate the rows.

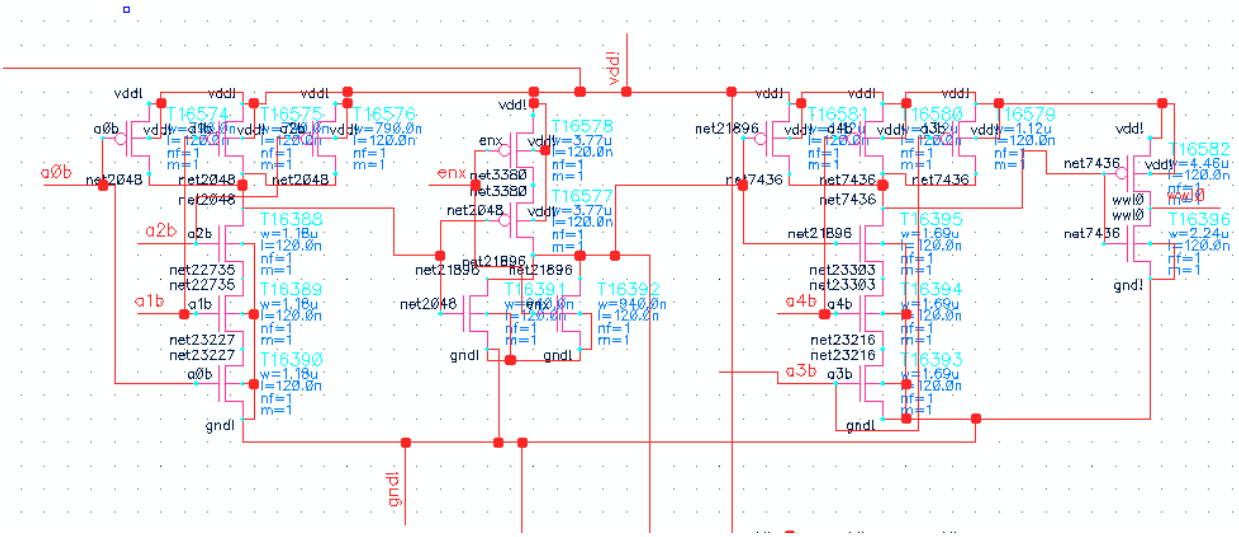

Fig.4 Schematic of Row Decoder (Single row)

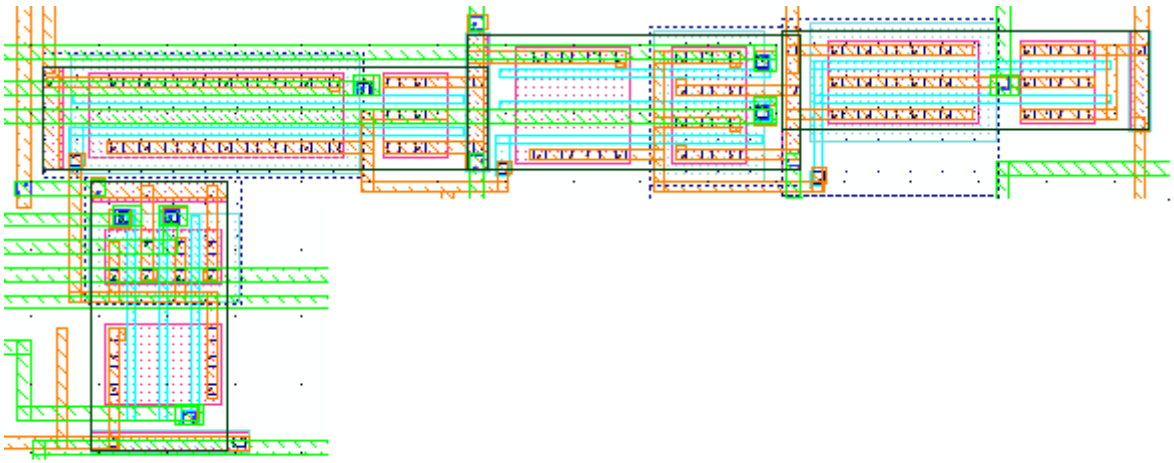

Fig.5 Layout of Row decoder (Single row)

3. COLUMN DECODERS

The column decoder will activate the pass transistors, which drives the word bitlines, wbl and wblb. Two column decoders are used, one for read and the other for write.

3.1 Sizing of the transistors in the column decoder

Calculations:

Cpoly = 2 x 2fF x 8= 32fF
Cwire = (2.38 um x 32 x 2) + (8 x 0.16 x 15) x 0.2fF = 34.304 fF
Cload = Cpoly + Cwire = 66.304 fF = 33.152 um
F=GBH= 1064.38
N= 6 Stages
$\hat{f} = F^{\frac{1}{N}}$ = 3.195

3.2 Transistor sizing

Inverter 1 (Right most) = 10.410 um
Inverter 2= 6.516 um
Inverter 3= 2.039 um
NAND4= 1.276 um
Inverter 5= 1.59

The outputs of the column decoders will be given as inputs to the pass transistors.

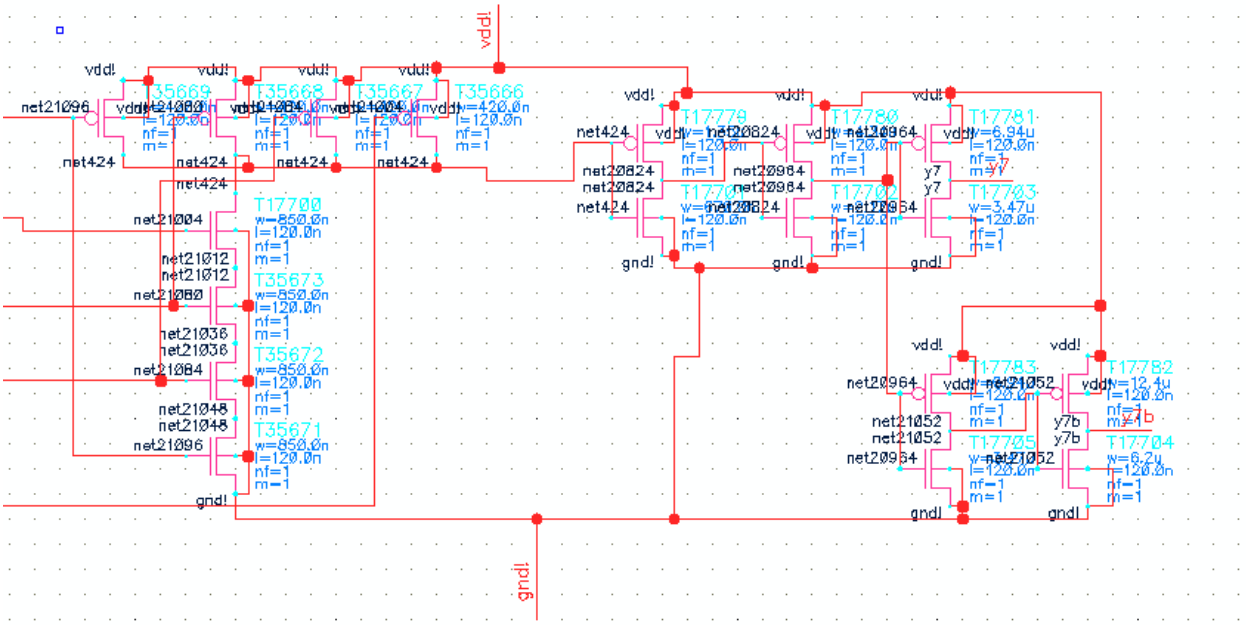

Fig.6 Schematic of Column decoder

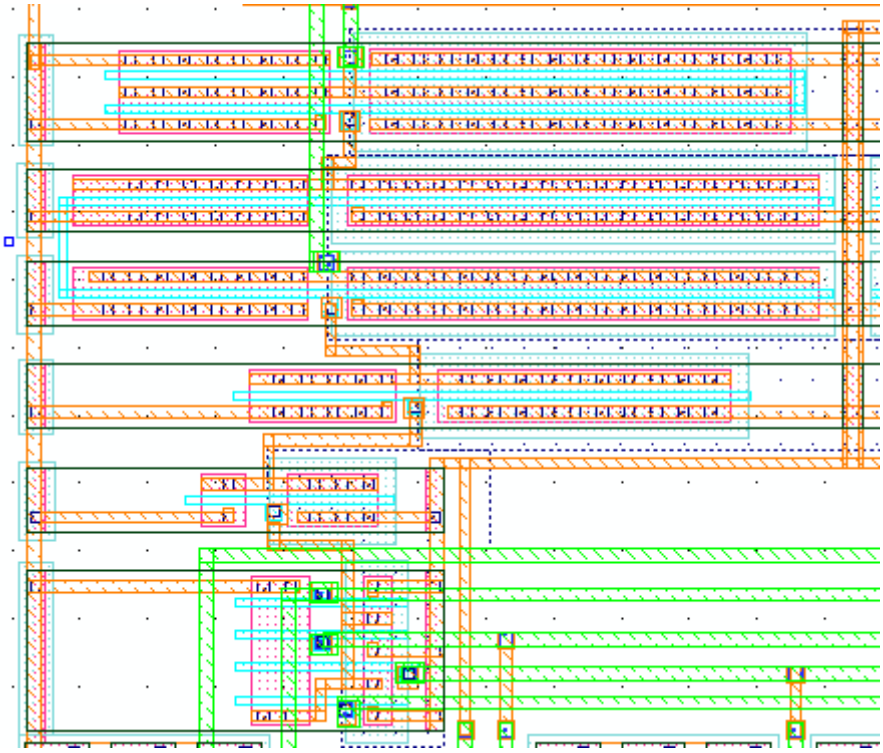

Fig.7 Layout of Column Decoder

4. SENSE AMPLIFIER

We used a Differential Amplifier on the bit lines to sense the voltage differences between wbl and wblb. The values are passed through pass transistors and are sensed by the sense amplifier. **We used transistors of sizes 0.84 um for PMOS and 0.28 um for NMOS in the circuit**. We have given a read bitline, rbl as input to the bottom most NMOS of the circuit.

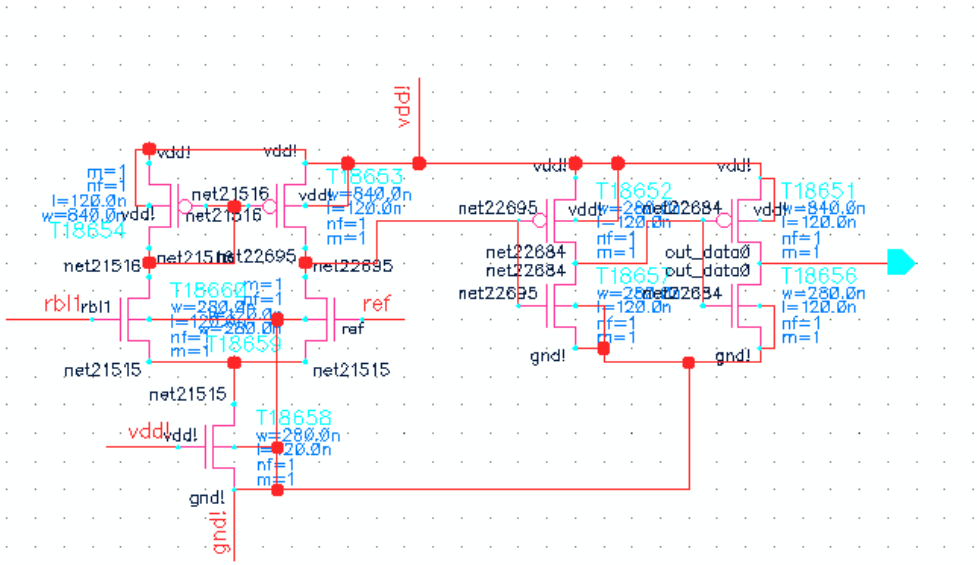

Fig.8 Schematic of Sense Amplifier

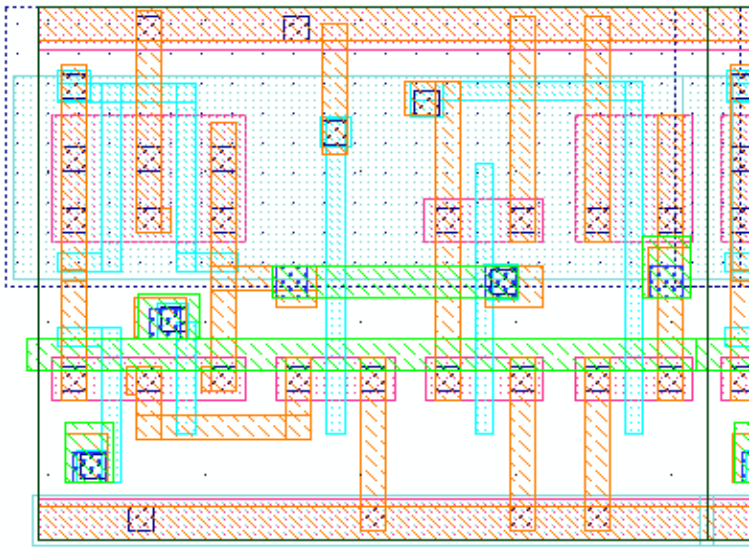

Fig.9 Layout of Sense Amplifier

5. WRITE BUFFERS

We used 2 Tri-state Transistors of sizes 6 um for PMOS and 6 um for NMOS.

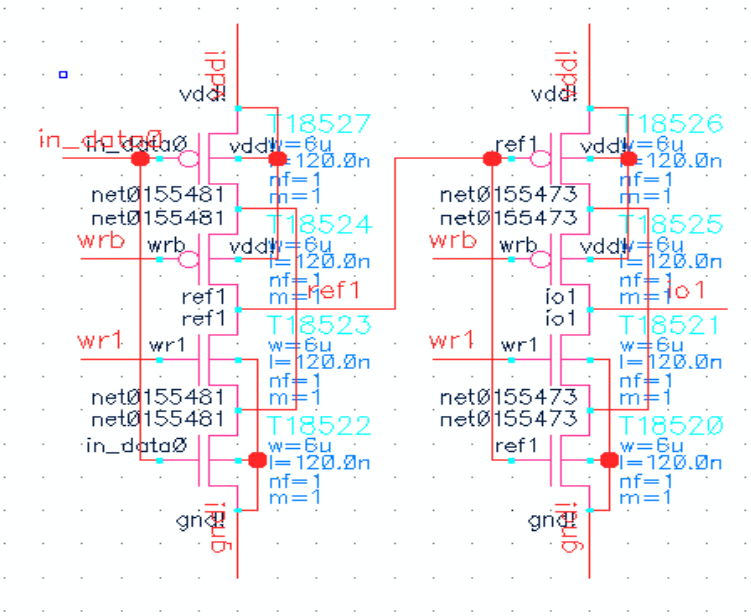

Fig.10 Schematic of Write Buffer

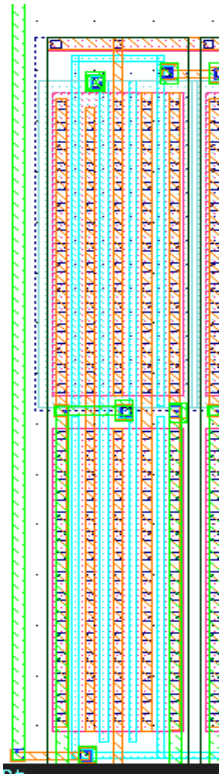

Fig.11 Layout of Write Buffer

6. AUXILIARY CIRCUITS

6.1 Pass transistors

The size of PMOS and NMOS is 1 um.

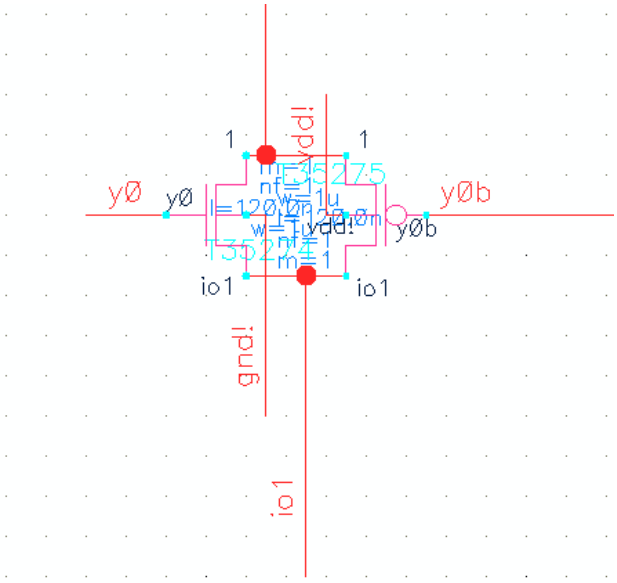

Fig.12 Schematic of Pass Transistor

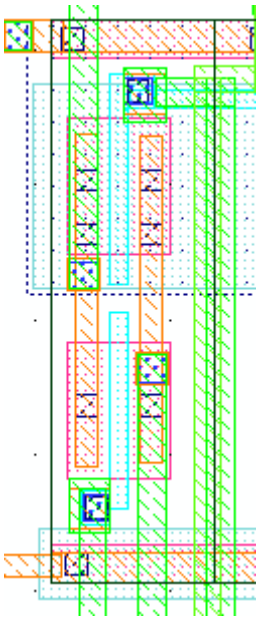

Fig.13 Layout of Pass Transistor

6.2 Pre-charger

The pre charger is used for bit line, wbl and wblb conditioning, which means to pre charge the bit line to V_{DD} before the next read or writes operation. Two PMOS transistors size is taken as 2um for good conditioning of bit line and bit line bars.

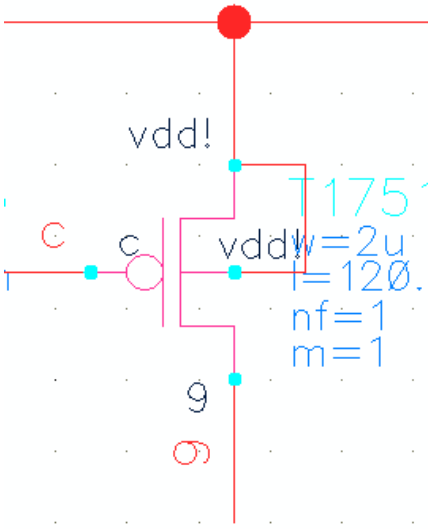

Fig.14 Schematic of Pre-charger

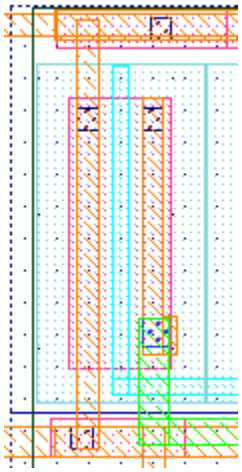

Fig.15 Layout of Pre-charger

6.3 Clock Buffer

A clock buffer is used to provide the clock input to the pre chargers. Input to the clock is defined in the Hspice code. At the pre charge state all the bit lines will be going to Vdd.

Calculations:

C poly = 2 x 2fF x 3 x 64 = 768 fF

C wire = 0.2fF x 152.27 = 30.454 fF
Cload = 798.54 fF = 399.232 um
F= GBH = 399.232 um
N = 5
$\hat{f} = F^{\frac{1}{N}} = 3.313$

Transistor sizes:
Inverter 1 (Right most) = 120.50 um
Inverter 2 = 36.37 um
Inverter 3 = 10.978 um
Inverter 4 = 3.313 um

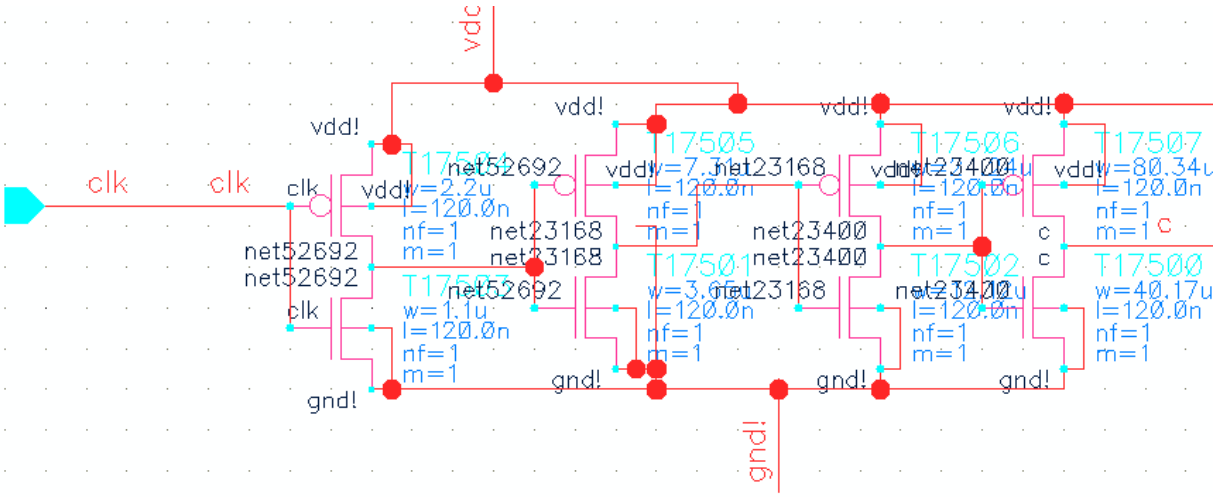

Fig.16 Schematic of Clock Buffer

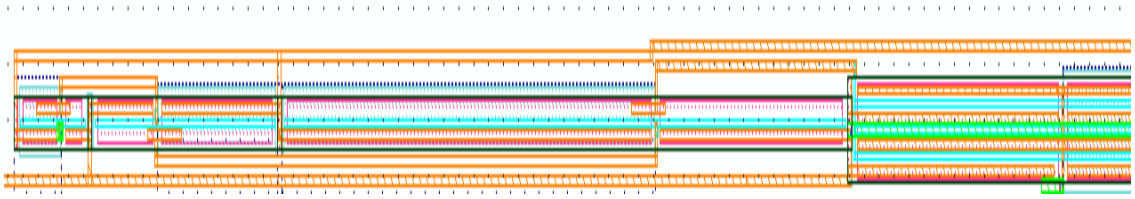

Fig.17 Layout of Clock Buffer

6.4 Input Drivers for address bits and address enable addr_en, read enable read_en:

Sizing for address bits: addr0, addr1, addr2, addr3, addr4
Cload = 1.978 um
$\hat{f} = F^{\frac{1}{N}} = 3.978$
$sqrt\ \hat{f} = 1.994$
Inverter 1 (Rightmost) = 1.988 um
Inverter 2 = 1.988 um
Inverter 3 = 3.967 um

Sizing for address enable: addr_en1 and addr_en2

Cload = 4.721 um

$\hat{f} = F^{\frac{1}{N}} = 3.355$

Inverter 1 (Rightmost) = 11.257 um

Inverter 2 = 3.355 um

Inverter 3 = 3.967 um

$sqrt\ \hat{f} = 1.831$

Inverter 1 (Rightmost) = 20.626 um

Inverter 2 = 11.265 um

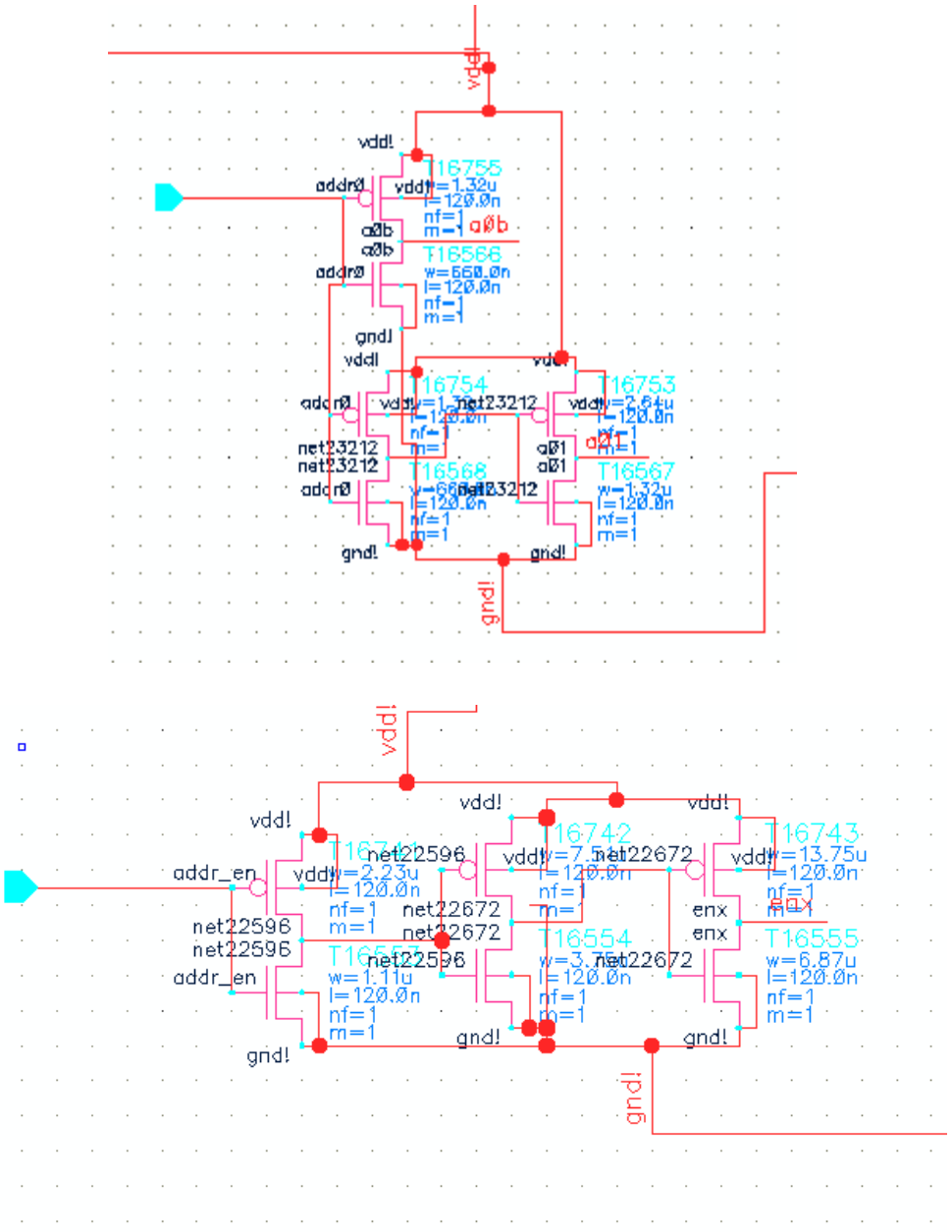

Fig.18 Schematic for Address bits and Address enable

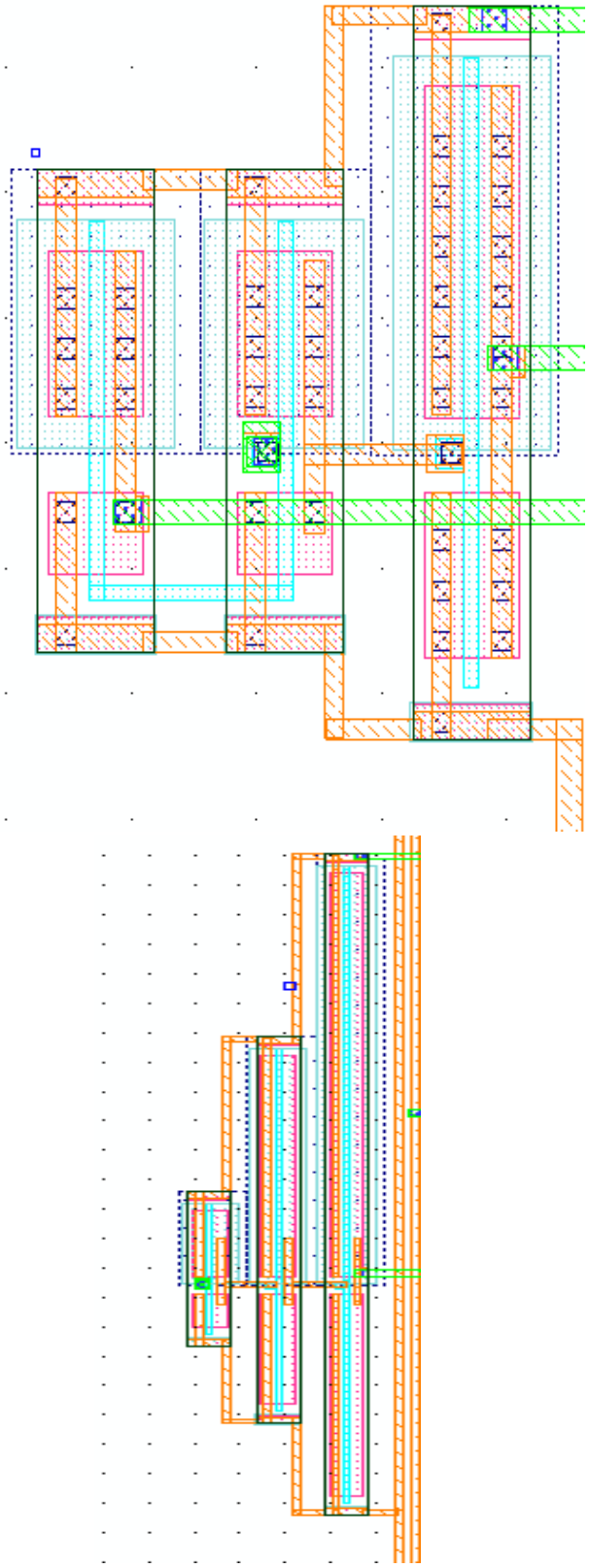

Fig.19 Layout for Address bits and Address enable

Sizing for address bits: addr5, addr6, addr7
Cload = 1.276 um
$\hat{f} = F^{\frac{1}{N}} = 3.195$
$sqrt\ \hat{f} = 1.787$
Inverter 1 (Rightmost) = 1.59 um
Inverter 2 = 1.59 um
Inverter 3 = 0.714 um

Sizing for read enable: read_en1, read_en2
Cload = 1.276 um
$\hat{f} = F^{\frac{1}{N}} = 2.169$
Inverter 1 (Rightmost) = 4.706 um
Inverter 2 = 2.169 um

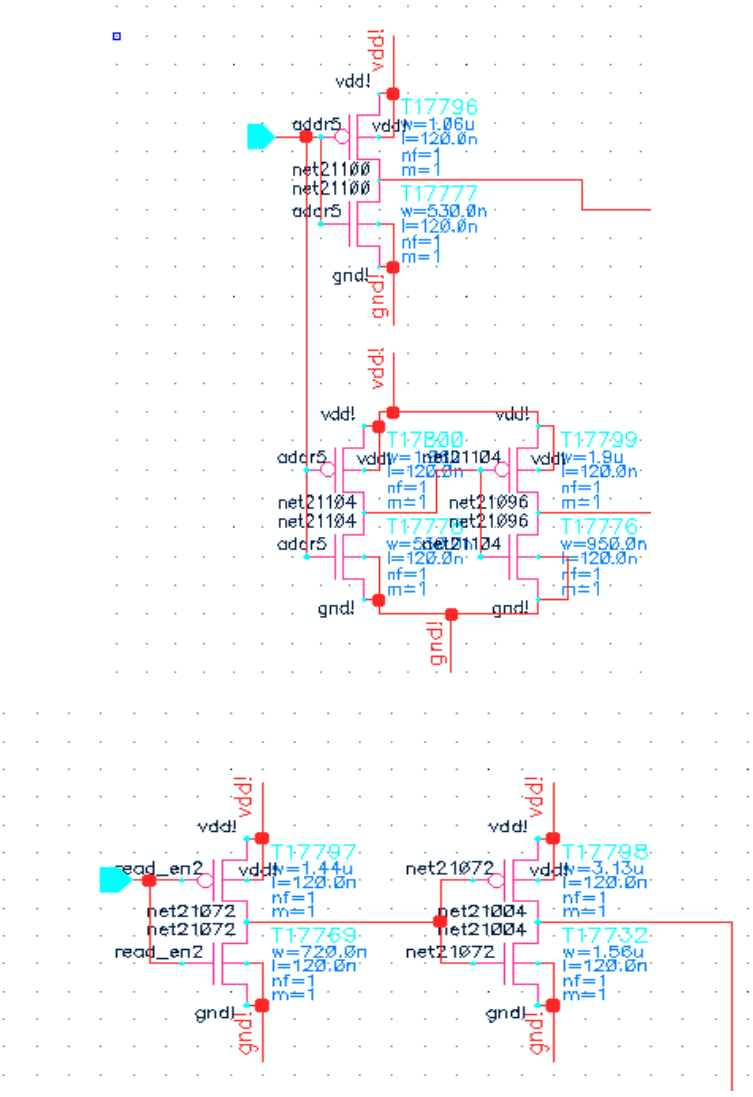

Fig.20 Schematic of Address bits and Read enables

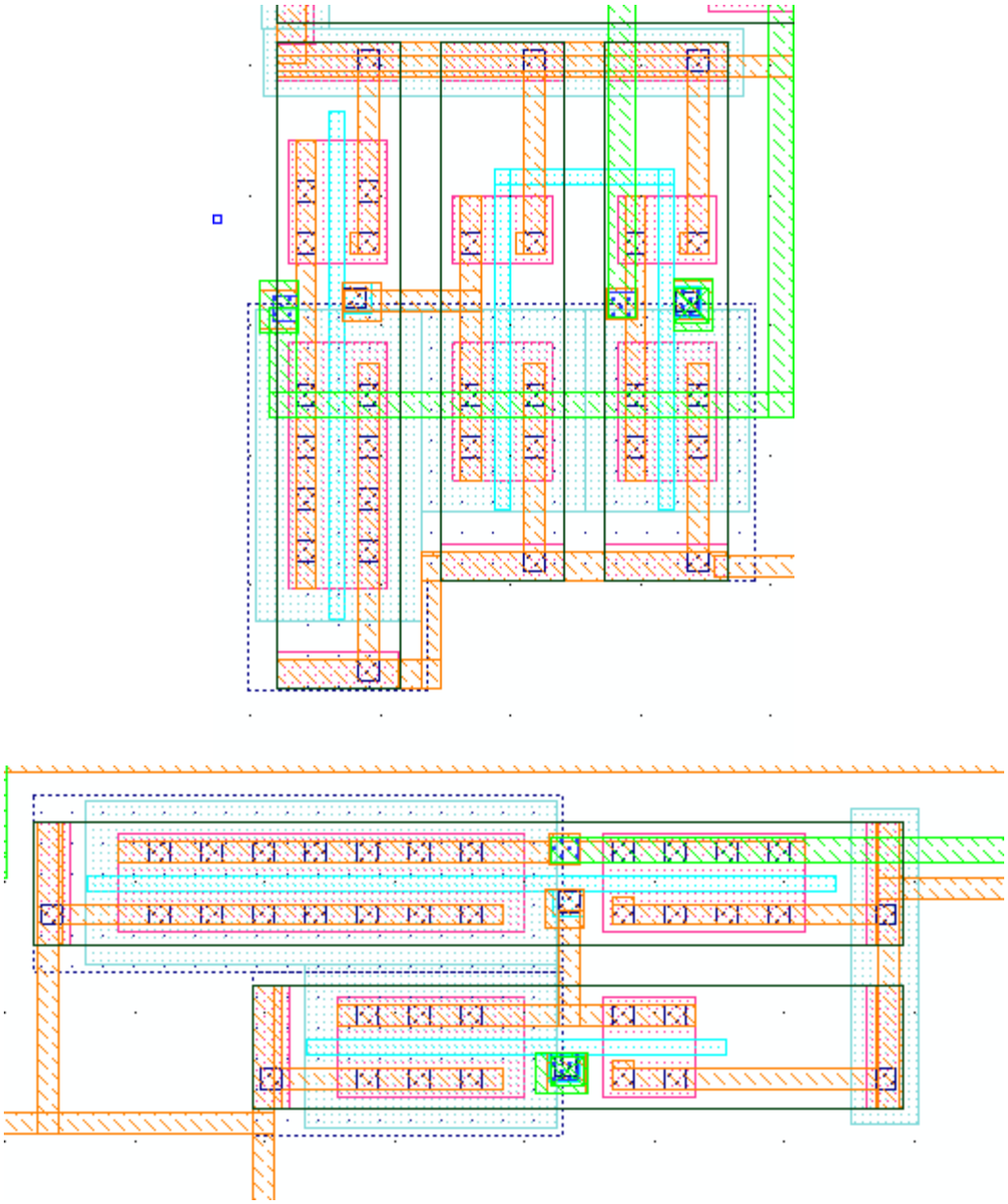

Fig.21 Layout of Address bits and Read enables

7. DRC AND LVS OF SRAM DESIGN

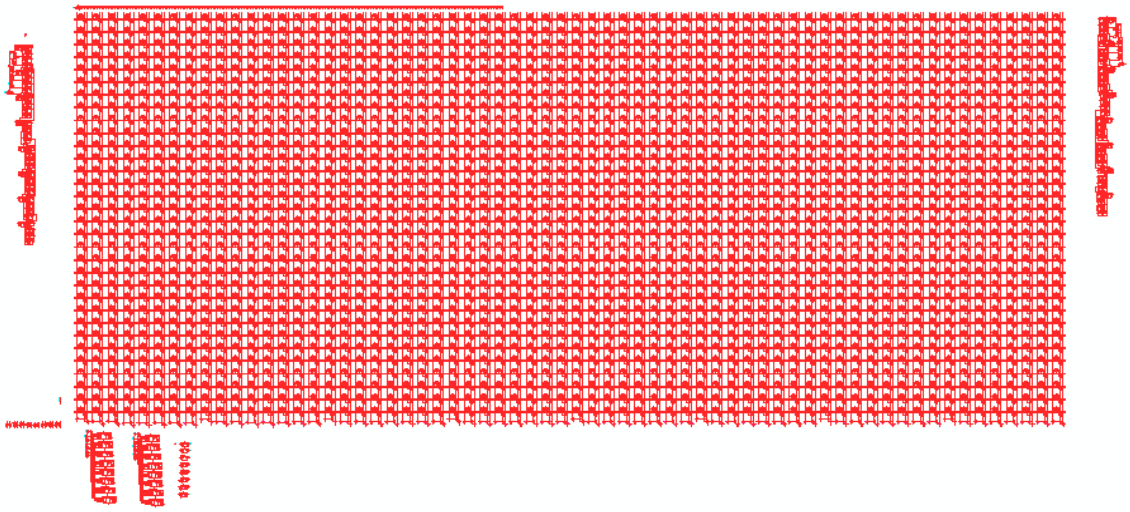

Fig.22 Schematic of Memory Cell Array

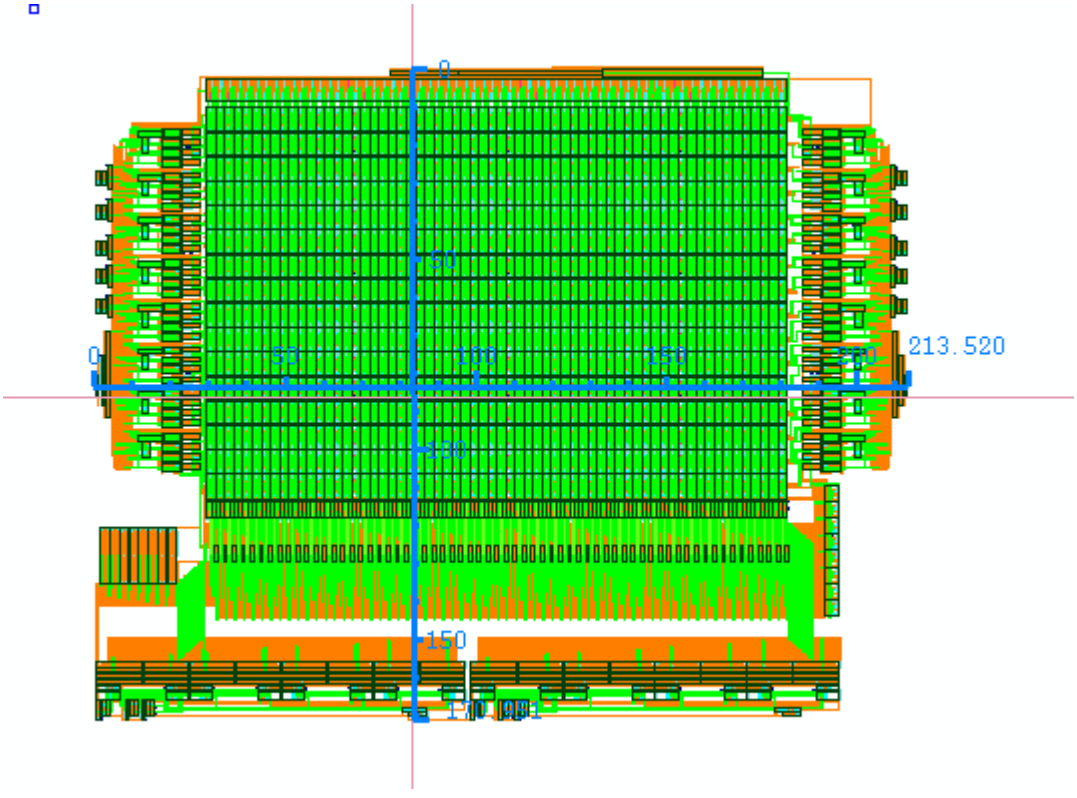

Fig.23 Layout of Memory Cell Array

Total Area = (213.520 um x 170.231 um) = 36347.723 sq um

Total area per bit = 17.747sq um

Aspect ratio = 0.797

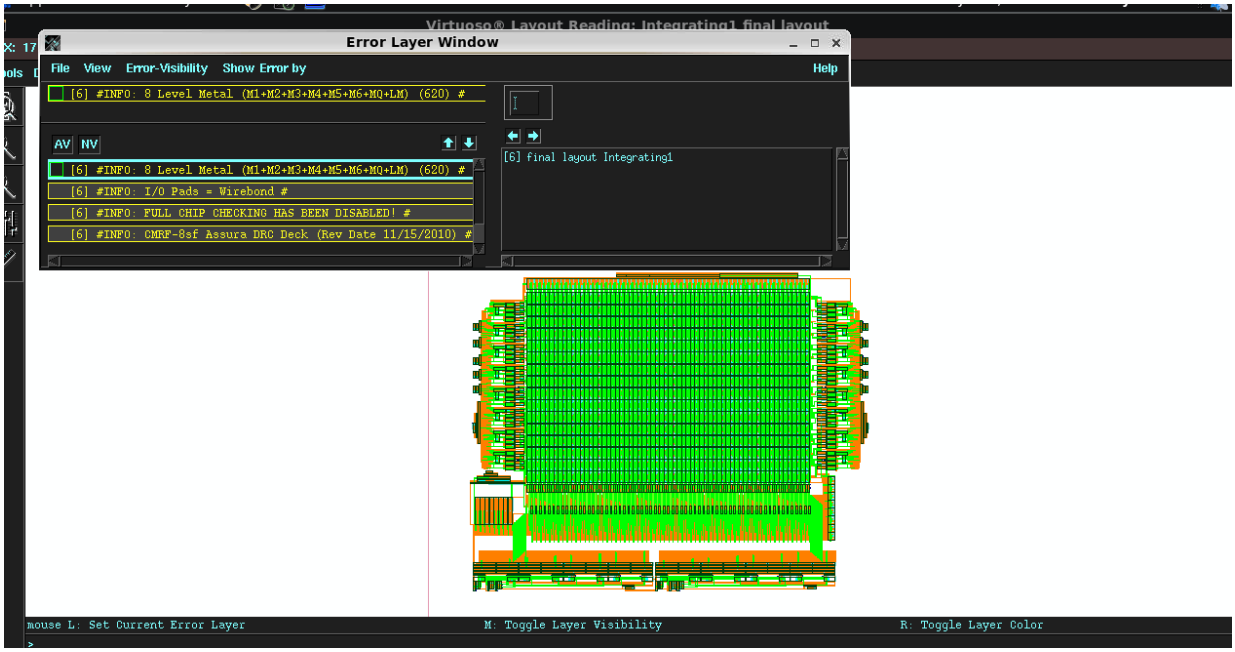

Fig.24 DRC Report

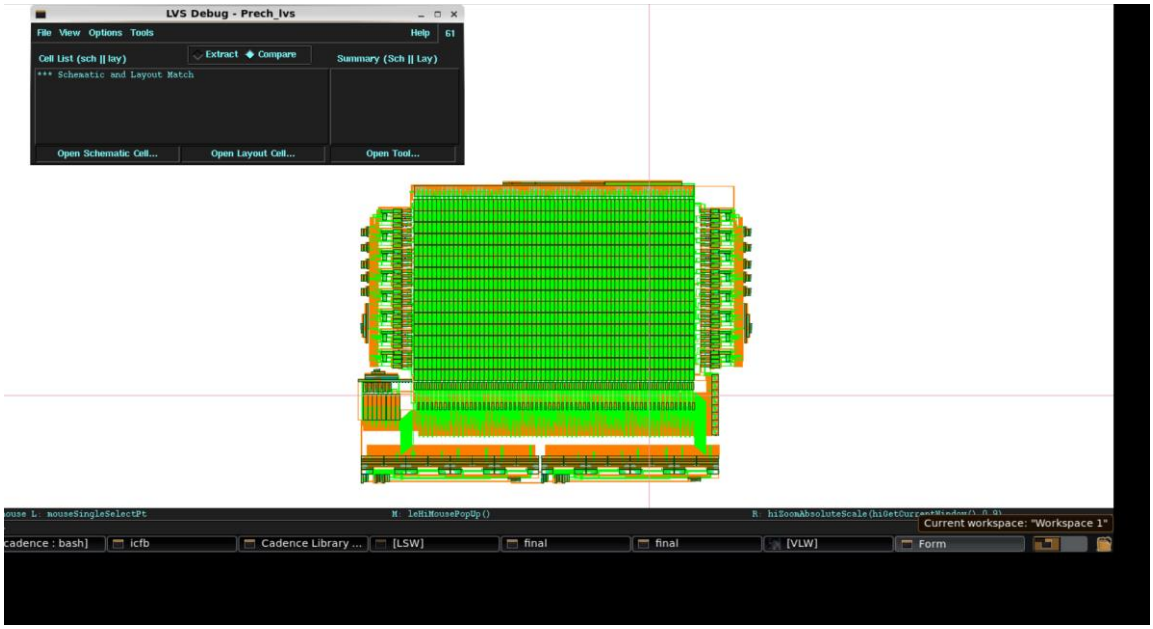

Fig.25 LVS Report

8. TIMING ANALYSIS

8.1 Hspice Code

```
$
$
$transistor model

.include "/home/cad/kits/IBM_CMRF8SF-
LM013/IBM_PDK/cmrf8sf/V1.2.0.0LM/HSPICE/models/model013.lib_inc"

.include "working.sp"

.option post runlvl=5

xi addr0 addr1 addr2 addr3 addr4 addr5 addr6 addr7 addr_en addr_en1
+ baddr0 baddr1 baddr2 baddr3 baddr4 baddr5 baddr6 baddr7 clk in_data0 in_data1
+ in_data2 in_data3 in_data4 in_data5 in_data6 in_data7 out_data0 out_data1
+ out_data2 out_data3 out_data4 out_data5 out_data6 out_data7 read_en1 read_en2
+ ref wr NEW

vdd vdd! gnd! 1.2V
Vref ref gnd! 0.6V
Cout_1 out_data0 0 25fF
Cout_2 out_data1  0 25fF
Cout_3 out_data2 0 25fF
Cout_4 out_data3 0 25fF
Cout_5 out_data4 0 25fF
Cout_6 out_data5 0 25fF
Cout_7 out_data6 0 25fF
Cout_8 out_data7 0 25fF

**********Address lines for row and column decoders1

Va0 addr0 gnd! PWL 0ns 1.2V
Va1 addr1 gnd! PWL 0ns 1.2V
Va2 addr2 gnd! PWL 0ns 1.2V
Va3 addr3 gnd! PWL 0ns 1.2V
Va4 addr4 gnd! PWL 0ns 1.2V
Va5 addr5 gnd! PWL 0ns 1.2V
Va6 addr6 gnd! PWL 0ns 1.2V
Va7 addr7 gnd! PWL 0ns 1.2V

**********Address lines for row and column decoders2
```

```
Vb0 baddr0 gnd! PWL 0ns 1.2V
Vb1 baddr1 gnd! PWL 0ns 1.2V
Vb2 baddr2 gnd! PWL 0ns 1.2V
Vb3 baddr3 gnd! PWL 0ns 1.2V
Vb4 baddr4 gnd! PWL 0ns 1.2V
Vb5 baddr5 gnd! PWL 0ns 1.2V
Vb6 baddr6 gnd! PWL 0ns 1.2V
Vb7 baddr7 gnd! PWL 0ns 1.2V

**********Data values to be written

Vd1 in_data0 gnd! PWL (0ns 1.2V 125ns 1.2V 125.2ns 0V)
Vd2 in_data1 gnd! PWL (0ns 0V 125ns 0V 125.2ns 1.2V)
Vd3 in_data2 gnd! PWL (0ns 1.2V 125ns 1.2V 125.2ns 0V)
Vd4 in_data3 gnd! PWL (0ns 0V 125ns 0V 125.2ns 1.2V)
Vd5 in_data4 gnd! PWL (0ns 1.2V 125ns 1.2V 125.2ns 0V)
Vd6 in_data5 gnd! PWL (0ns 0V 125ns 0V 125.2ns 1.2V)
Vd7 in_data6 gnd! PWL (0ns 1.2V 125ns 1.2V 125.2ns 0V)
Vd8 in_data7 gnd! PWL (0ns 0V 125ns 0V 125.2ns 1.2V)

**********Values for Read enable, Write enable and CLK
V1 wr  gnd! PWL (0ns 0V 50ns 0V 50.2ns 1.2V 60ns 1.2V 60.2ns 0V 150ns 0V 150.2ns 1.2V 160ns
1.2V 160.2ns 0V)
V2 addr_en gnd! PWL (0ns 0V 50ns 0V 50.2ns 1.2V 70ns 1.2V 70.2ns 0V 150ns 0V 150.2ns 1.2V 170ns
1.2V 170.2ns 0V)
V3 addr_en1 gnd! PWL (0ns 0V 80ns 0V 80.2ns 1.2V 100ns 1.2V 100.2ns 0V 180ns 0V 180.2ns 1.2V
200ns 1.2V 200.2ns 0V)
V4 read_en1 gnd! PWL (0ns 0V 80ns 0V 80.2ns 1.2V 100ns 1.2V 100.2ns 0V 180ns 0V 180.2ns 1.2V
200ns 1.2V 200.2ns 0V)
V5 read_en2 gnd! PWL (0ns 0V 50ns 0V 50.2ns 1.2V 70ns 1.2V 70.2ns 0V 150ns 0V 150.2ns 1.2V
170ns 1.2V 170.2ns 0V)
V6 clk gnd! PWL(0ns 0V 50ns 0V 50.2ns 1.2V 100ns 1.2V 100.2ns 0V 150ns 0V 150.2ns 1.2V 200ns
1.2V 200.2ns 0V 250ns 0V 250.2ns 1.2V 300ns 1.2V 300.2ns 0V)


.tr 0.01ns 500ns

.end
```

8.2 Testing the correct functionality:

To check the correct functionality, writing and reading of word 10101010 is checked.

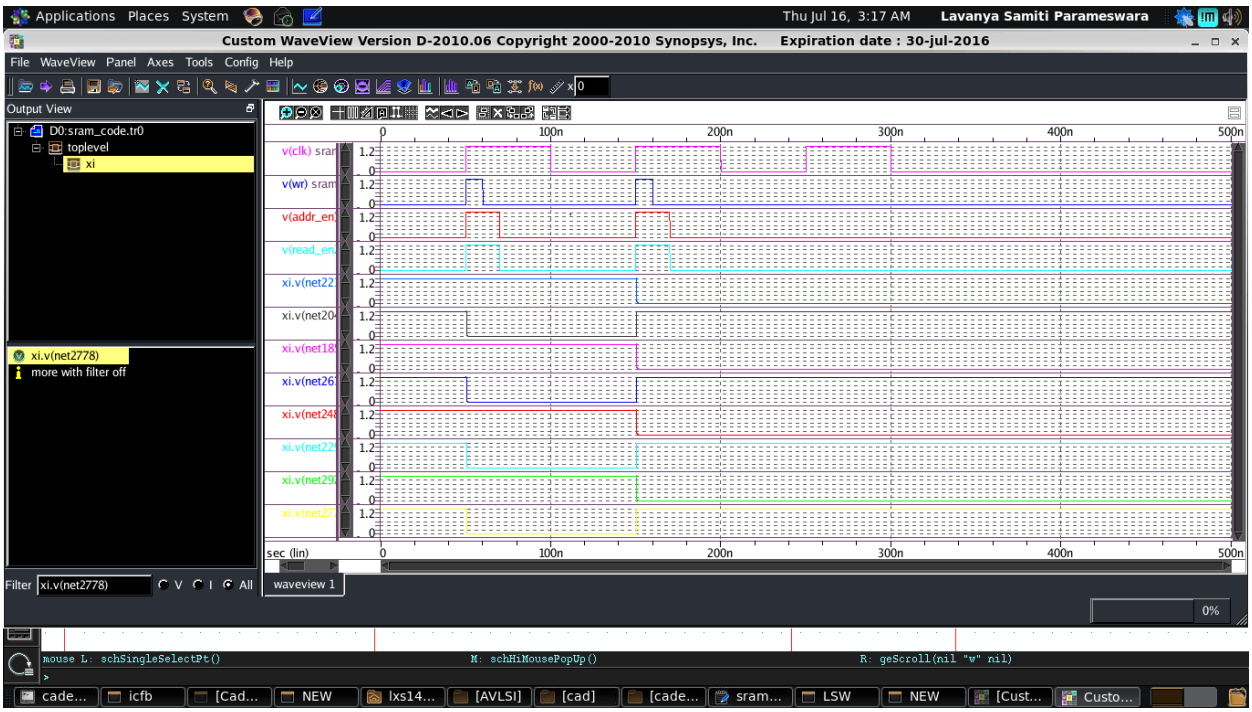

Fig.26 Functionality check of write 10101010

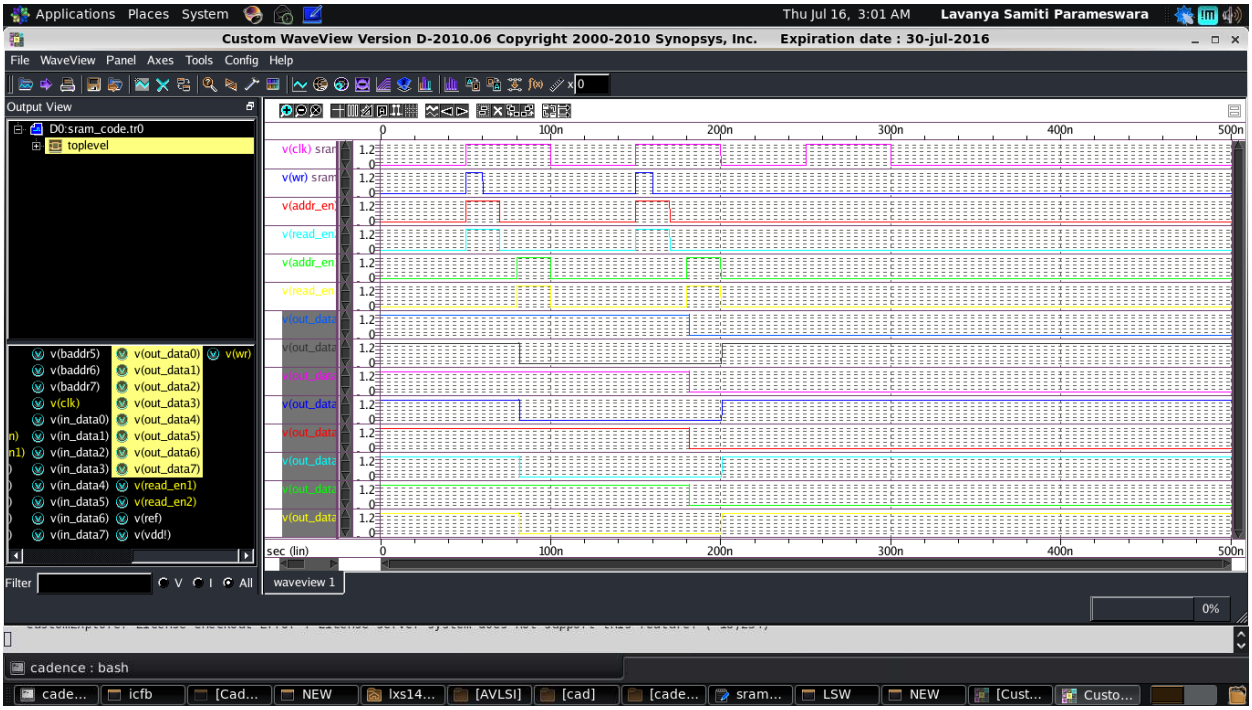

Fig.27 Functionality check of read 10101010

8.3 Worst case writing time for ‘0’ = 451 ps and for ‘1’ = 467 ps

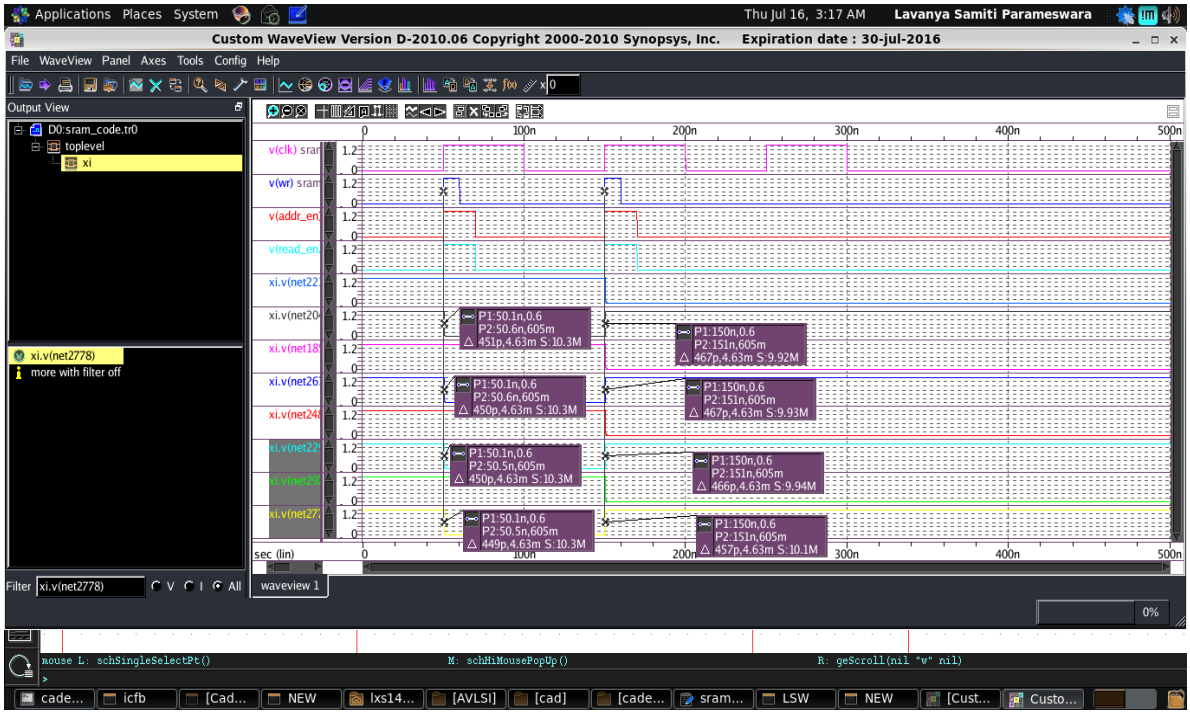

Fig.28 Worst Case Write Time for Zero and One

8.4 Worst case reading time for ‘0’ = 20.6 ns and for ‘1’ = 1.59ns

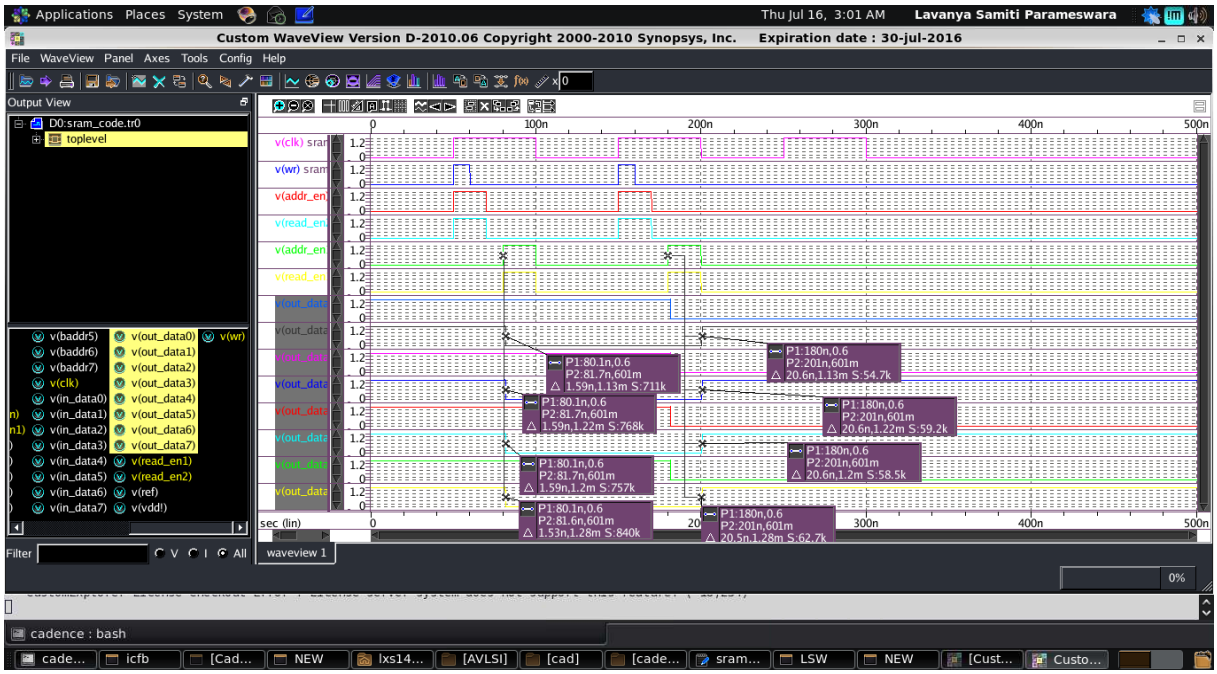

Fig.29 Worst Case Read Time for Zero and One

8.5 Operating Frequency = 1GHz

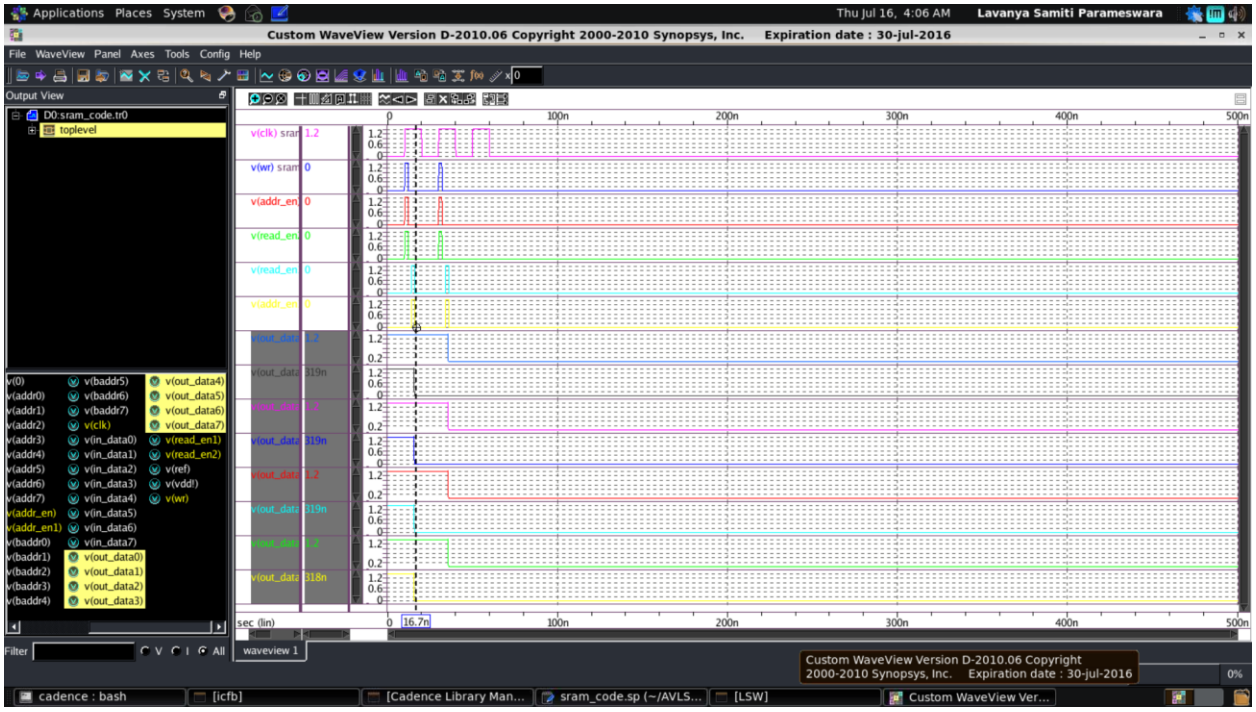

Fig. 30 Operating frequency

9. RESULTS AND CONCLUSIONS

A 256 word SRAM was designed with an area of 36347.723 sq um.

The Worst Case Read and Write Times are found to be 20.6 ns and 467 ps.

The operating frequency is 1GHz.