Area of Memory cell	**2.69 um x 2.38 um = 6.4022squm**
Aspect Ratio of Memory cell	**1.13**
Memory array area	**7422.8449squm**
Memory area per bit	**5.79squm**
Total Memory area	**13162.2146um^2 (136.580um x 96.370 um)**
Total Memory area per bit	**10.28um^2**
Aspect Ratio of total memory	**0.7**
Worst case Write time	**323ps**
Worst case Read time	**617ps**
Operating Frequency	**2.7 GHz**

INTRODUCTION:

Static random access memory (SRAM) is a type of volatile semiconductor memory meaning it stores data binary logic ‘1’ and ‘0’ as long as it is electrically powered. It uses bi-stable latching circuitry made of transistors to store each bit. SRAM works without refreshing, unlike Dynamic RAM (DRAM) which must be periodically refreshed. SRAM is faster and more reliable than the more common DRAM. But it is more expensive to produce than DRAM and due to this, SRAM is often used only as a memory cache.

PROJECT DESCRIPTION:

- Design and layout a 128 word SRAM (word size is 10bits) using the IBM 130nm process. The key design tools used are Cadence’s Virtuoso for layout editing, DRC (for design rule checking), LVS (layout versus netlist, for verifying that the layout matches the schematic netlist) and circuit simulation (for measuring the read/write times).
- An output capacitance of 30fF is used for all outputs when simulating for delays.
- All input signals, and clocks are provided by inverters sized: pMOS=0.75um and nMOS=0.25um.

SRAM Implementation

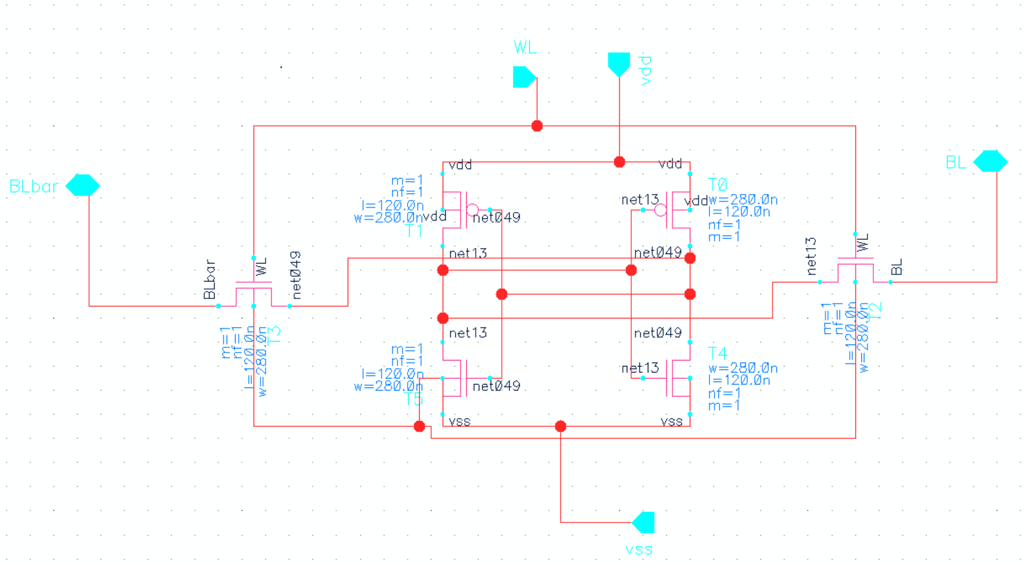
128 word SRAM has 32*40 memory cells. The word size is given to be 10 bits. Therefore, it has 2^5 rows and 10 columns with 4 words being stored in a row. Hence, we need a 5 bit address line to access one of the rows/word lines and a 2 bit address line to access one of the four words.

An SRAM architecture consists of the following blocks:

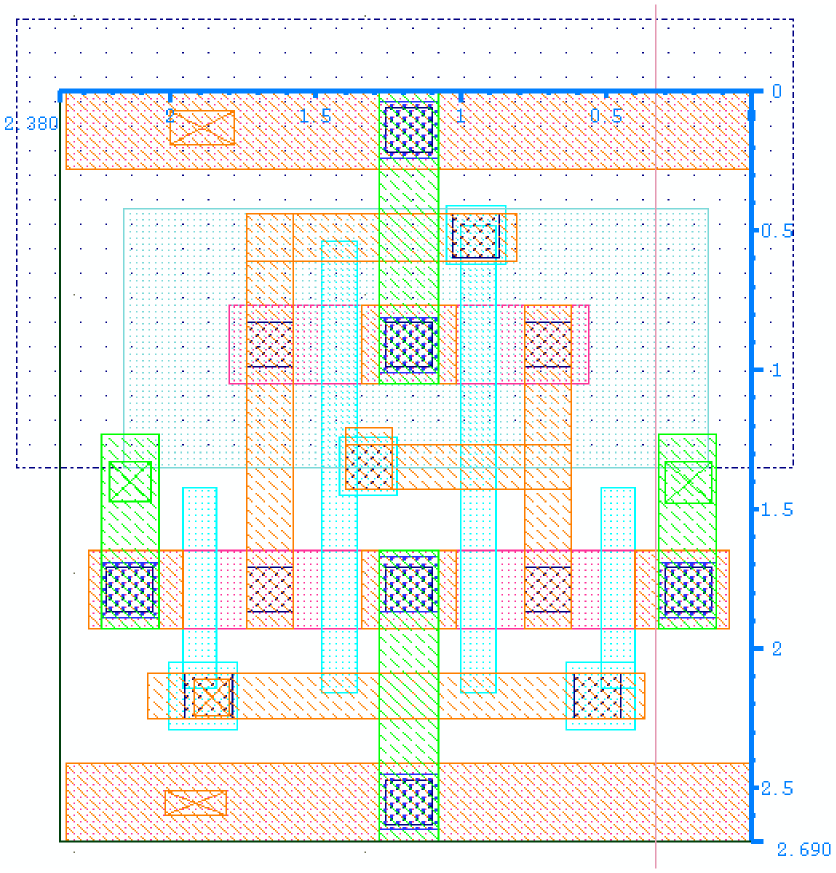
(a) 6T (Six-Transistor) Memory cell Array
(b) Row decoder
(c) Column decoder
(d) Column Multiplexer
(e) Sense amplifier
(f) Write driver
(g) Pre charge circuit
(h) Clock and Write driver circuits

6T (Six transistors Memory Cell) [Library name: tutorial, Cell name: mcell]:

6T cell consists of a latch and two NMOS which are driven by the word line. One such block constitutes a bit. Read and Write operations are performed on this block. The two NMOs transistors are turned on by the row decoder. The schematic and layout of the 6T cell are shown below.



6T Schematic



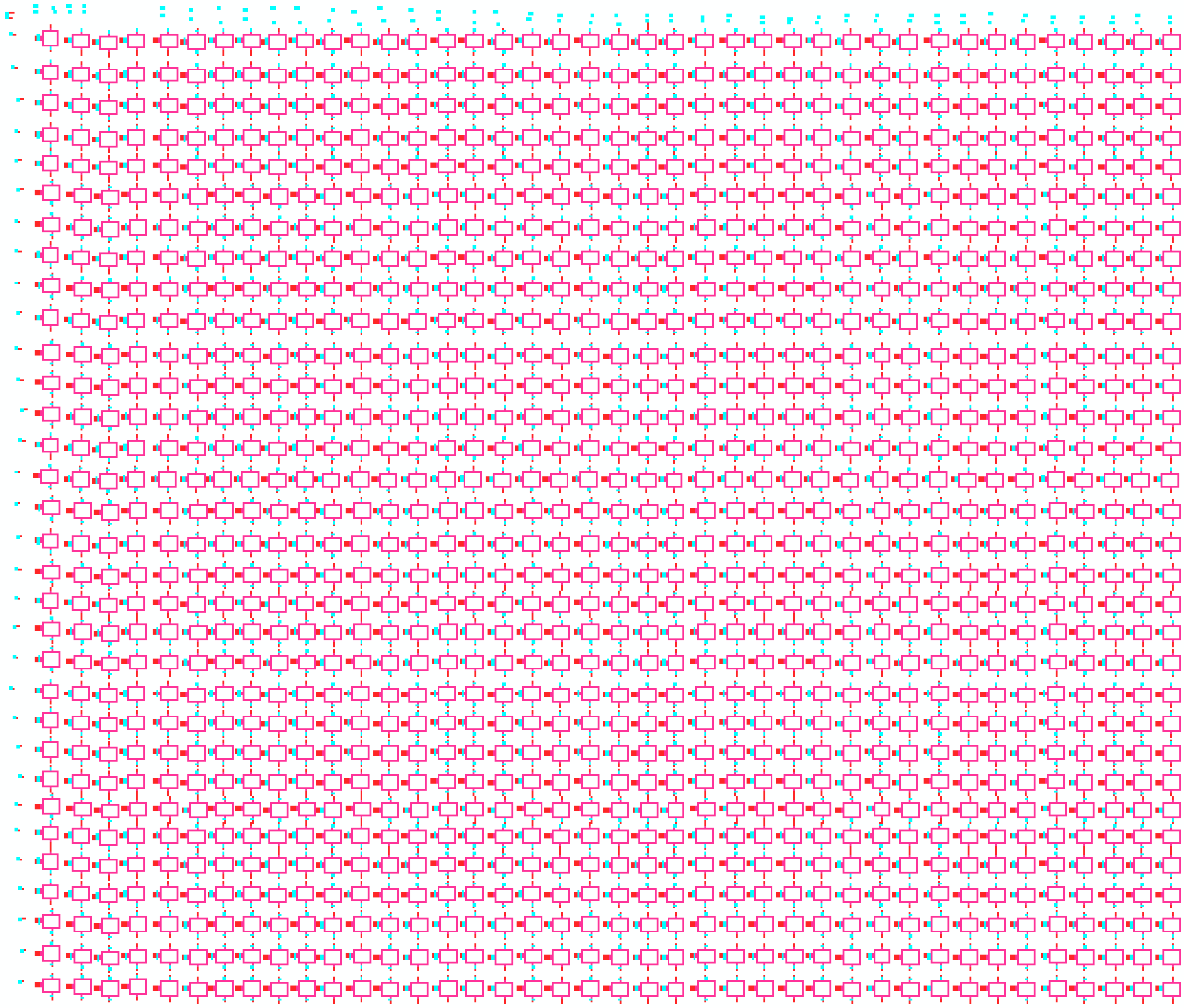
6T Layout

Memory Cell Array [Library name: tutorial, Cell name: 1280cells]:

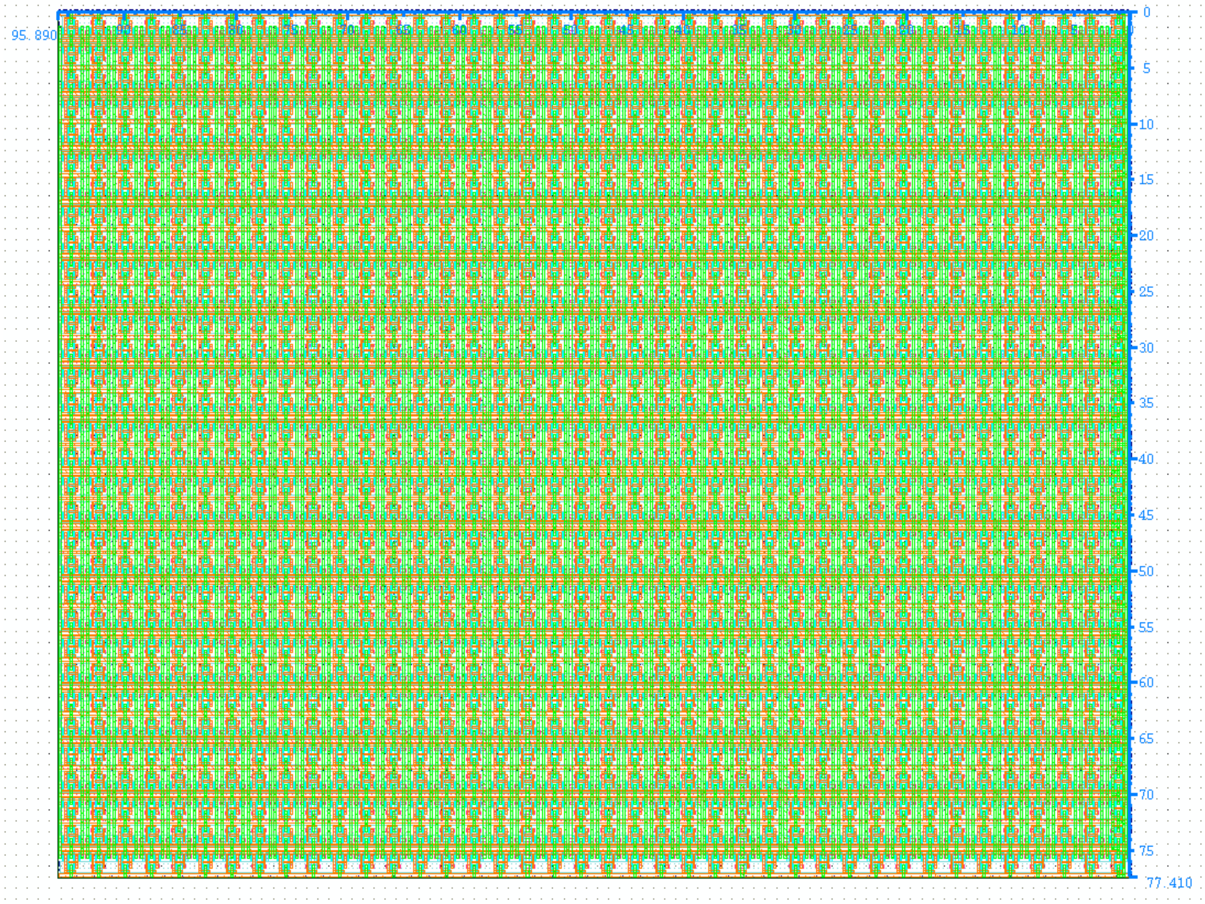
Memory Array is made from the Memory cells by placing it row wise and column wise. The number of cells per row and column depends upon the memory architecture and memory size.

Height of the cell: 2.69um
Width of the cell : 2.38um
The area of a single 6T cell is found to be 6.4022um^2 (2.69 * 2.38).
Aspect ratio: Height/Width = 1.13

Memory Array Schematic:



Memory Array Layout:



Height = 77.410um
Width = 95.89um
Area = 77.410*95.89= 7422.8449squm
Area per bit = 5.79squm

ROW DECODER [Library name: tutorial, Cell name: row_final]:

Row decoder is used to select the required row in the memory array. The required wordline is activated based on the address given to the decoder. In this project, we used a pre decoded 5X32 row decoder.

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

We are having 6 inputs (address bits and address enable) to the row decoder. The design has been implemented by using 3-input nand gates followed by inverters. By using the necessary information from the single 6T 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 = 44.86fF
Cwire = 19.04fF
Cload = Cpoly + Cwire = 63.9fF = 31.95um

G for two 3-input NAND gates, inverter and 2-input NAND gate = 3.7
B= 16
H=31.95

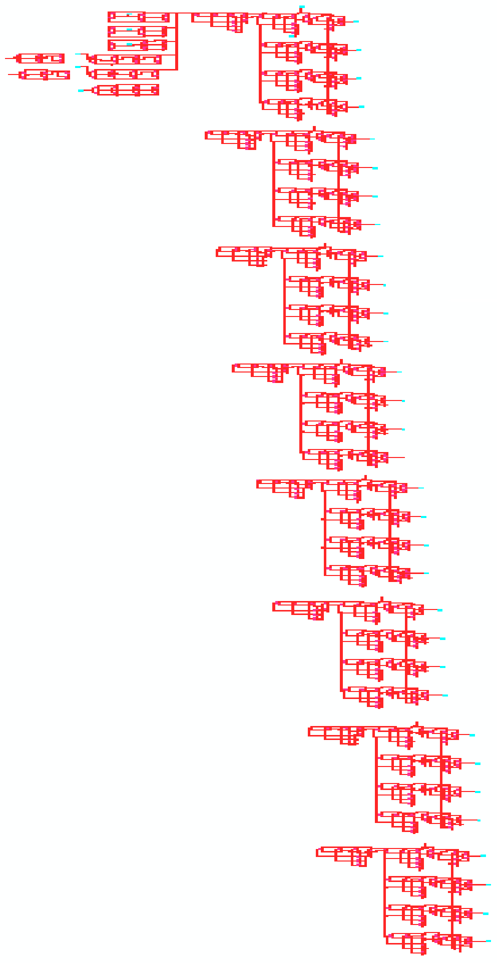
F= GBH= 1891.44
N = $\log_{3.6} F$ = 6. But it is considered to be 7
$\hat{f} = F^{\frac{1}{N}}$ = 2.938

Transistor sizing

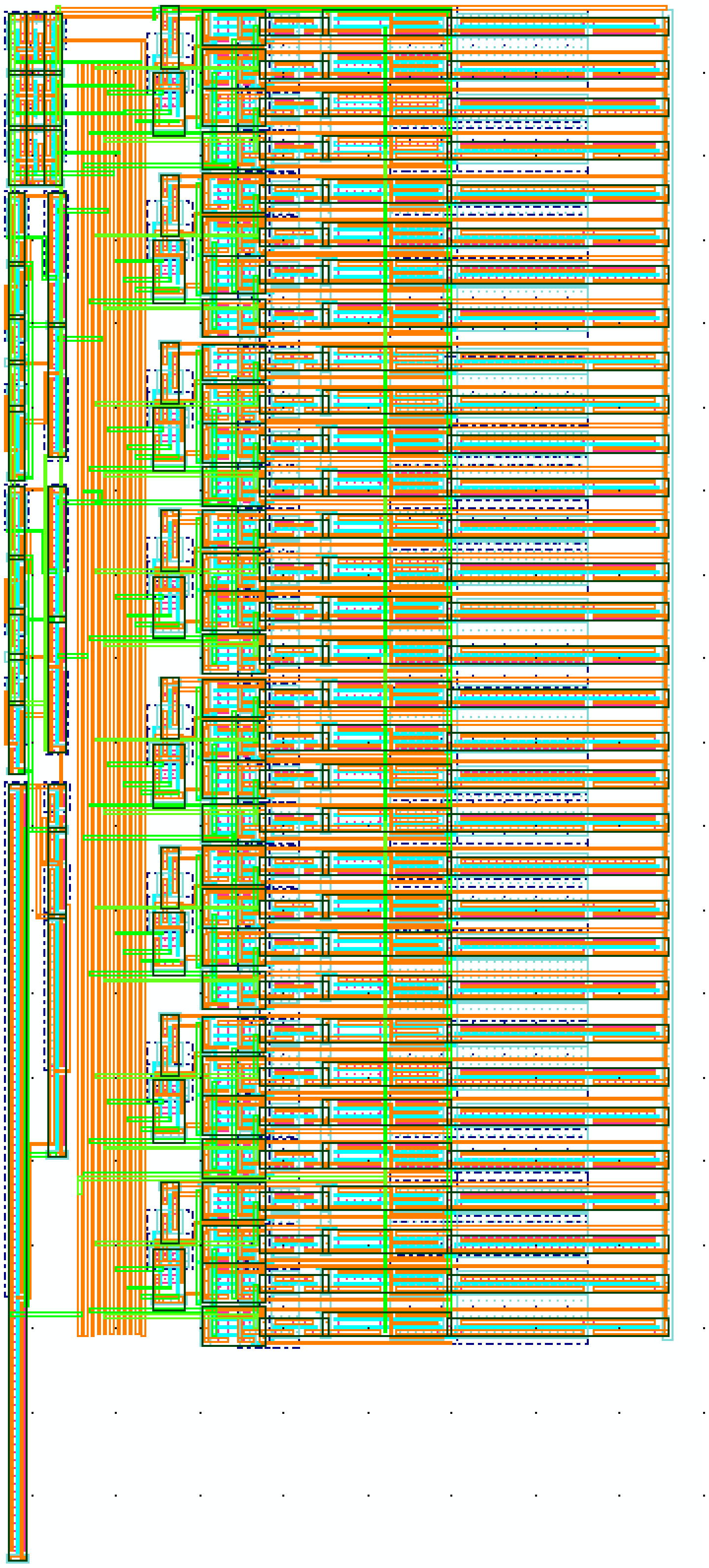
Inverter #1(rightmost): wp= 7.24um, wn= 3.62um
NAND2: wp= 2.47um, wn= 2.47um
Inverter #2: wp= 1.12um, wn= 0.56um
NAND3: wp= 0.38um, wn= 0.57um
Inverter #3: wp= 0.86um, wn=0.43um
NAND3: wp= 0.29um, wn=0.44um
Inverter #4: wp= 0.66um, wn=0.33um
Inverter #5: wp= 1.14um, wn=0.57um

The inputs to the row decoder are addr0, addr1, addr2, addr3, addr4, addr5 and addr_en. The outputs of row decoders are given to the word lines of memory lines.

Row Decoder Schematic:



Row Decoder Layout:



COLUMN DECODER [Library name: tutorial, Cell name: Cdec. (Cdec#2e)]:

After precharging all the bitlines to a high voltage, the next step is to select the columns that will be involved in the read or write operation.
The Column Decoder required in the SRAM design is designed as follows.

Calculations

Cpoly = 22.4fF
Cwire = 19.04 fF
Cload = Cpoly + Cwire = 41.44 fF= 20.71um
F=GBH= 110.45
N= 4 Stages
$\hat{f} = F^{\frac{1}{N}} = 3.24$

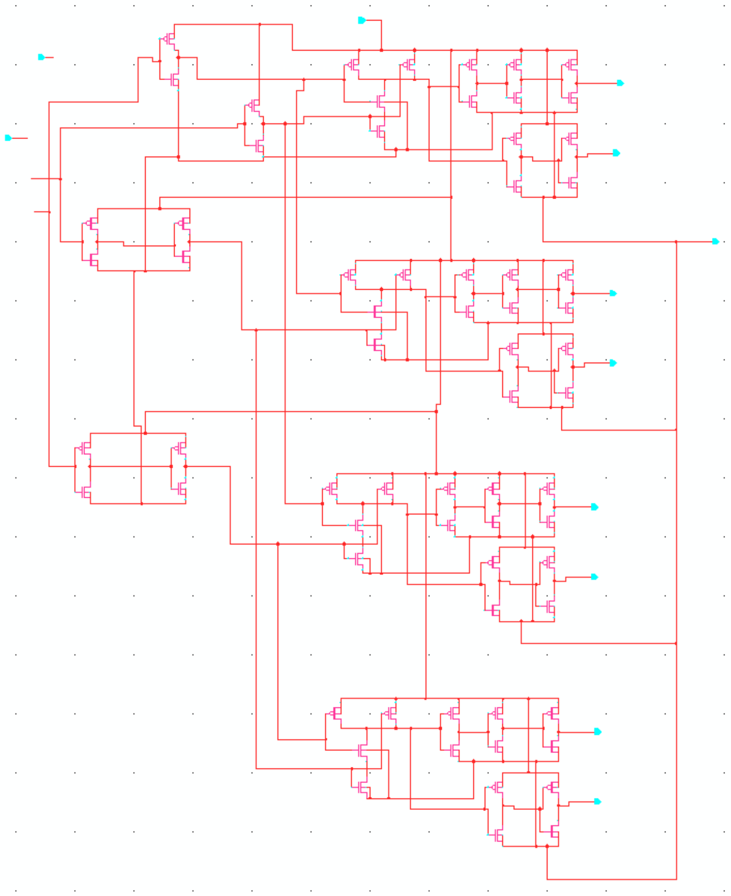
Transistor sizing

Inverter #1(rightmost): wp= 4.26um, wn= 2.13um
Inverter #2: wp= 2.37um, wn= 1.18um
Inverter #3: wp= 1.31um, wn=0.66um
NAND2: wp= 0.81um, wn= 0.81um
Inverter #4: wp= 0.67um, wn=0.33um
Inverter #5: wp= 1.2um, wn=0.6um

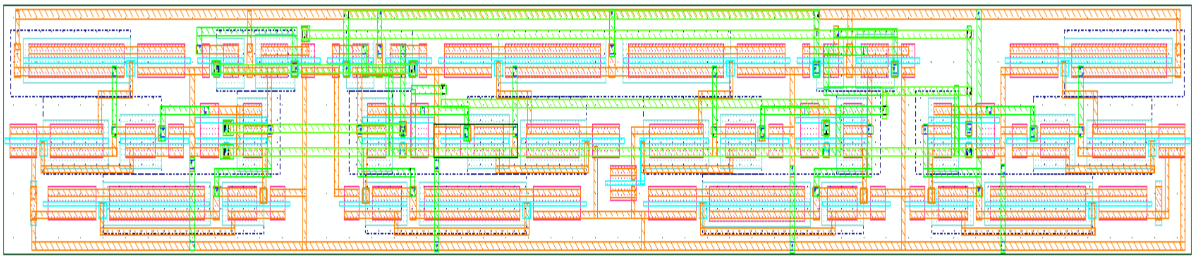
Inputs to the column decoder: addr5, addr6

Outputs : y1, y1bar, y2, y2bar, y3, y3bar, y4, y4bar (which are fed to the column MUX).

Column Decoder Schematic:



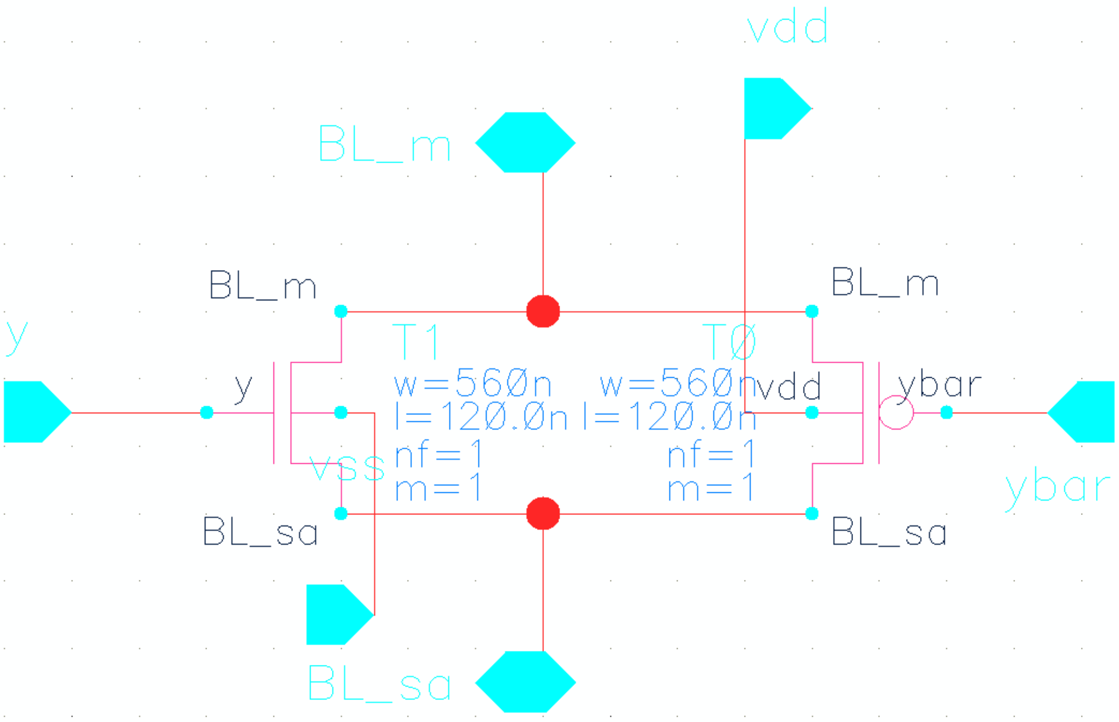
Column Decoder Layout:



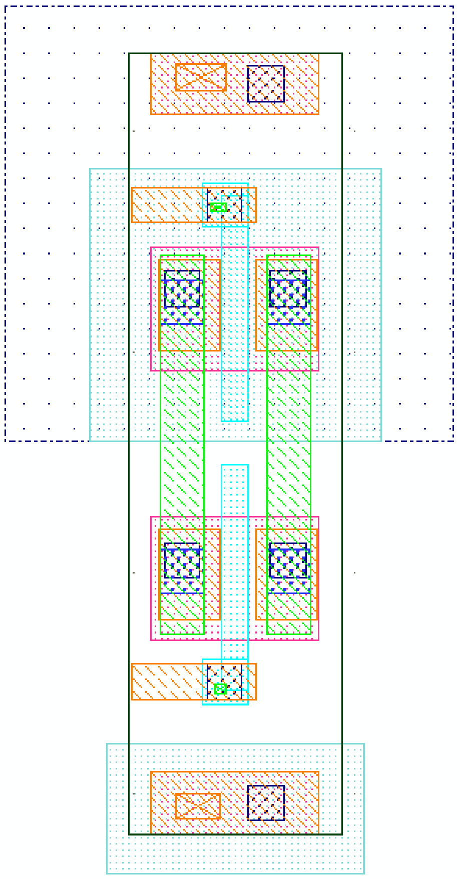
Column Multiplexers [Library name: tutorial, Cell name: pass_trans]:

CMOS Transmission gates are used in order to select the columns. This is done because PMOS device is better at transmitting a logic ‘1’ while the NMOS device is better at transmitting a logic ‘0’. Since the bitlines are near VDD during a read operation, PMOS is turned ON during a read operation leaving the NMOS device OFF. During a write operation, a bitline is pulled to a low voltage. Therefore, PMOS is turned OFF and the NMOS is turned ON. The transmission gates are also sized for optimal speed.

The schematic of the Transmission gate:



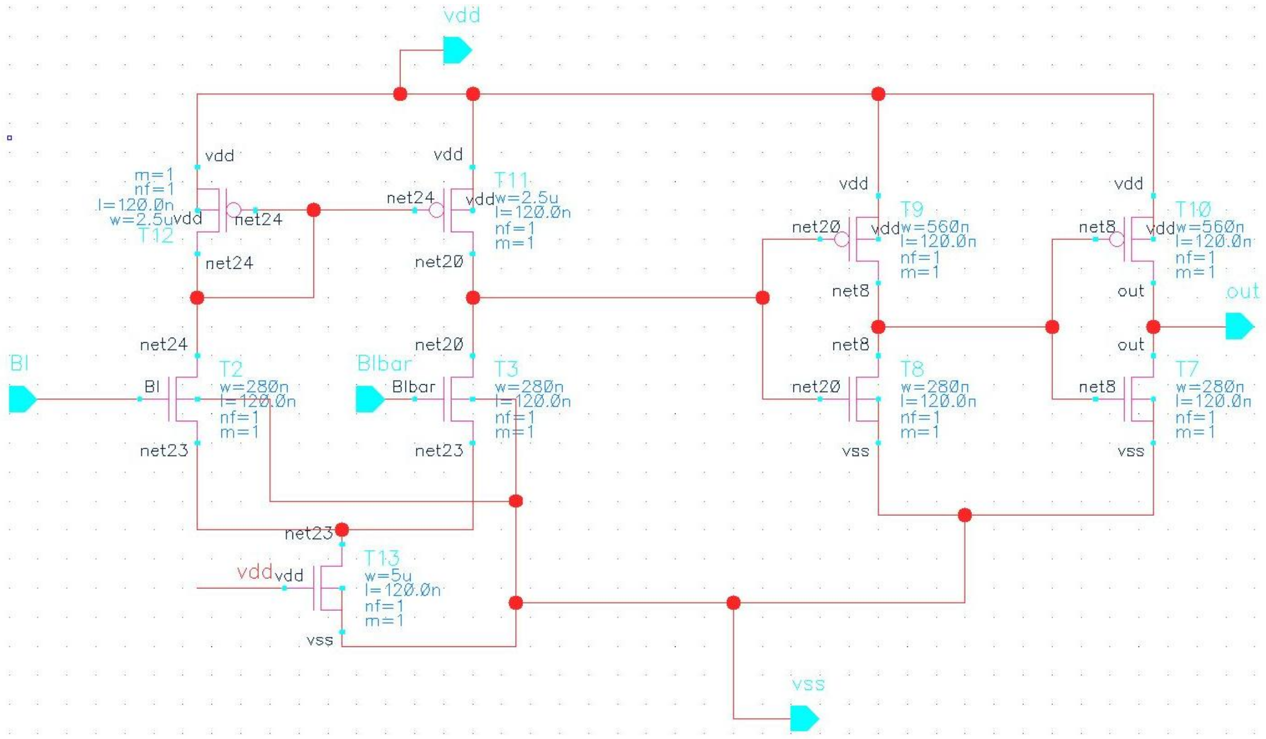
The layout of the Transmission gate:



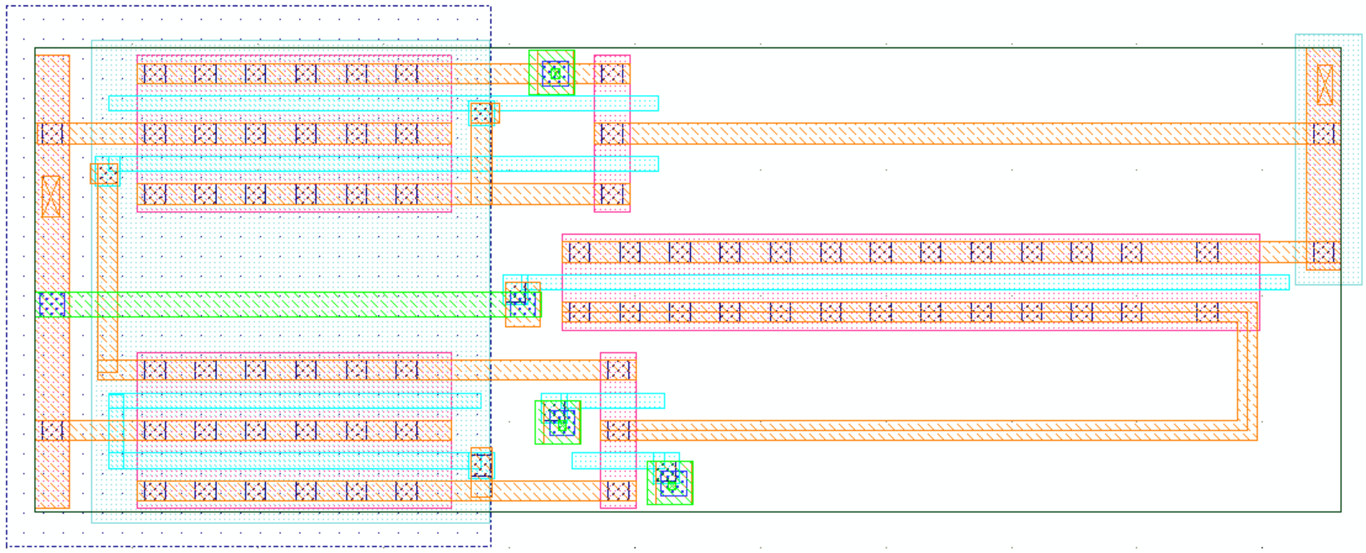
SENSE AMPLIFIER [Library name: tutorial, Cell name: sense_amp]:

We used a Differential Amplifier on the bit lines to sense the voltage differences between BL and BLBAR. The values that are passed through pass transistors are sensed by the sense amplifier.

Sense Amplifier Schematic:



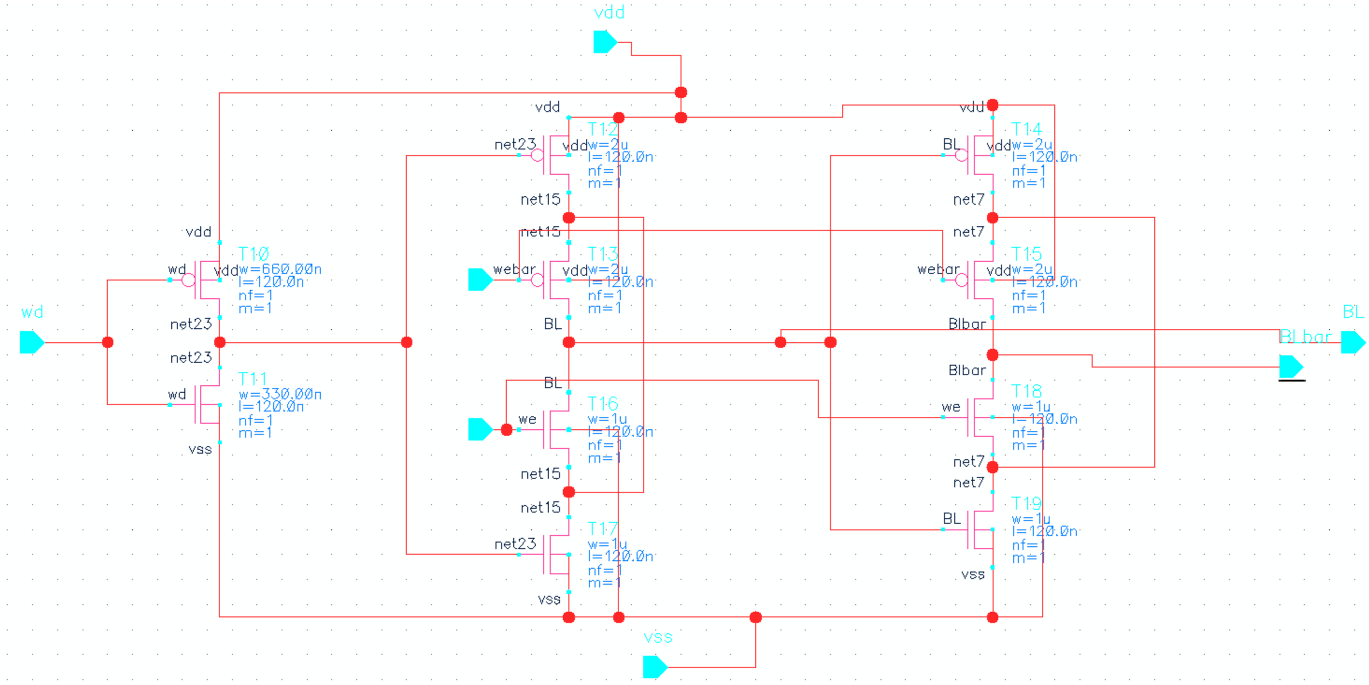
Sense Amplifier Layout:



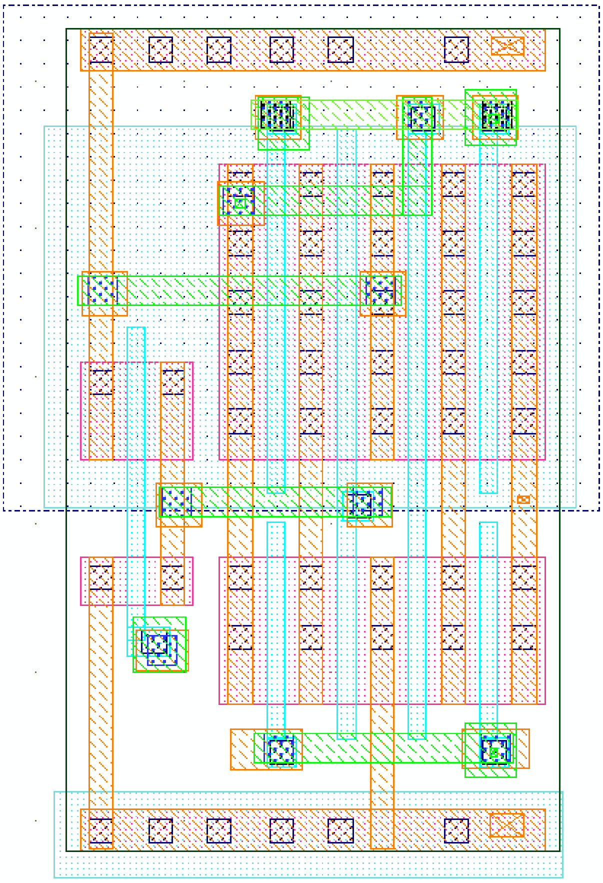
WRITE DRIVER [Library name: tutorial, Cell name: write_circuit]:

During clock precharging wr input must be disabled. Before, write operation, one of the bitlines must be driven high and the other low based on the data bit that is being written., When wr signal goes high, write operation is done and the 10 bit data can be written by giving required bit values to the corresponding input bits wd. These values are then passed through a set of tri-state inveretrs that are attached to the BL and BLbar lines so that the data bit will be written into the corresponding memory cell. Both the read and write operations are occured only during evaluation.

The schematic of Write Driver:



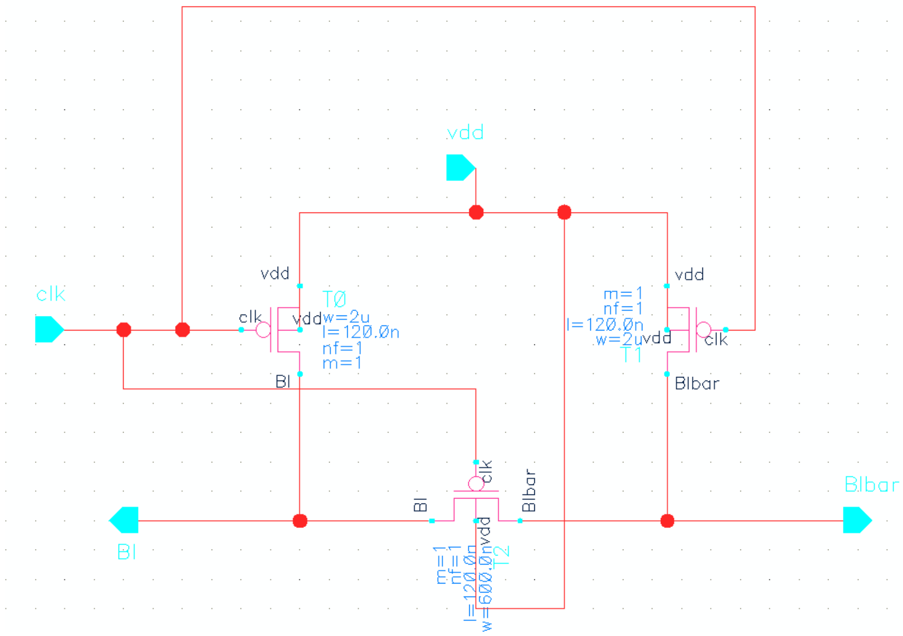
The layout of Write Driver:



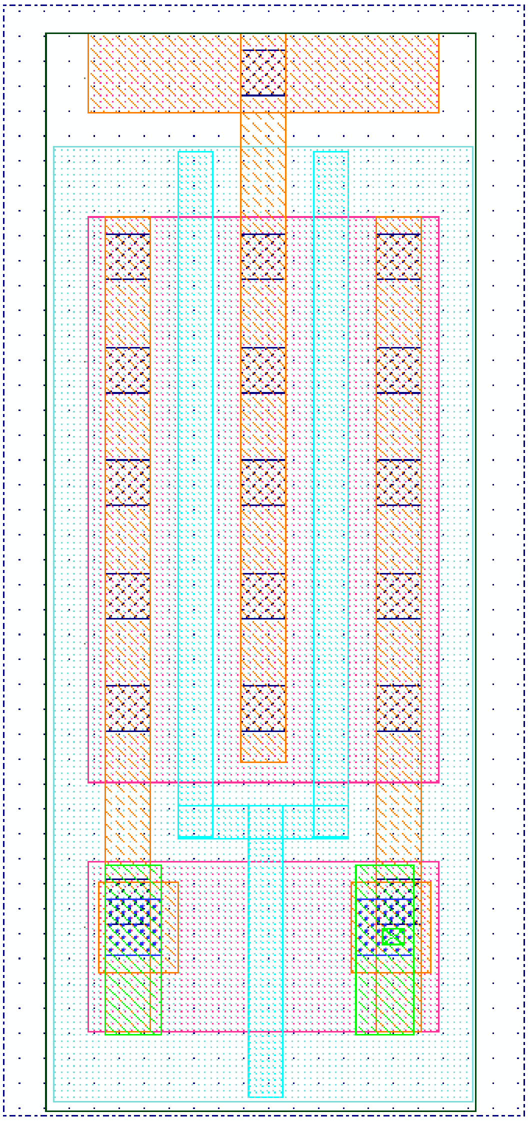
PRECHARGE CIRCUIT [Library name: tutorial, Cell name: pre_charge]:

In both read and write operations, the bitlines are initially pulled up to a high voltage near VDD. This is Accomplished using a Precharge circuit. As shown in the schematic of precharge circuit, a clocked input ‘clk’ is applied to the two pull-ups and to a third transistor (balance transistor), connected between the two bitlines to equalize their voltage levels. When the wordline signal goes high, one bitline remains high and the other falls at a linear rate until wordline goes low. The difference between the bitlines is fed into a sense amplifier that is triggered when the differential voltage exceeds a certain threshold.

The schematic of Precharge circuit:



The layout of Precharge circuit:



Other Auxiliary circuits:

Write Buffer [Library name: tutorial, Cell name: Wbuff]:

Since the wr signal drives two nfets in each column, a total of 20 nfets will be driven by wr. In addition, it drives inverter that is used to generate its complement that is given to the pfets of the pass transistor pair. Hence the buffer circuit for wr must be suitably sized so that it drives the required load. The transistor sizing is as given below.

Cpoly= 80fF

Cwire= 19.04fF

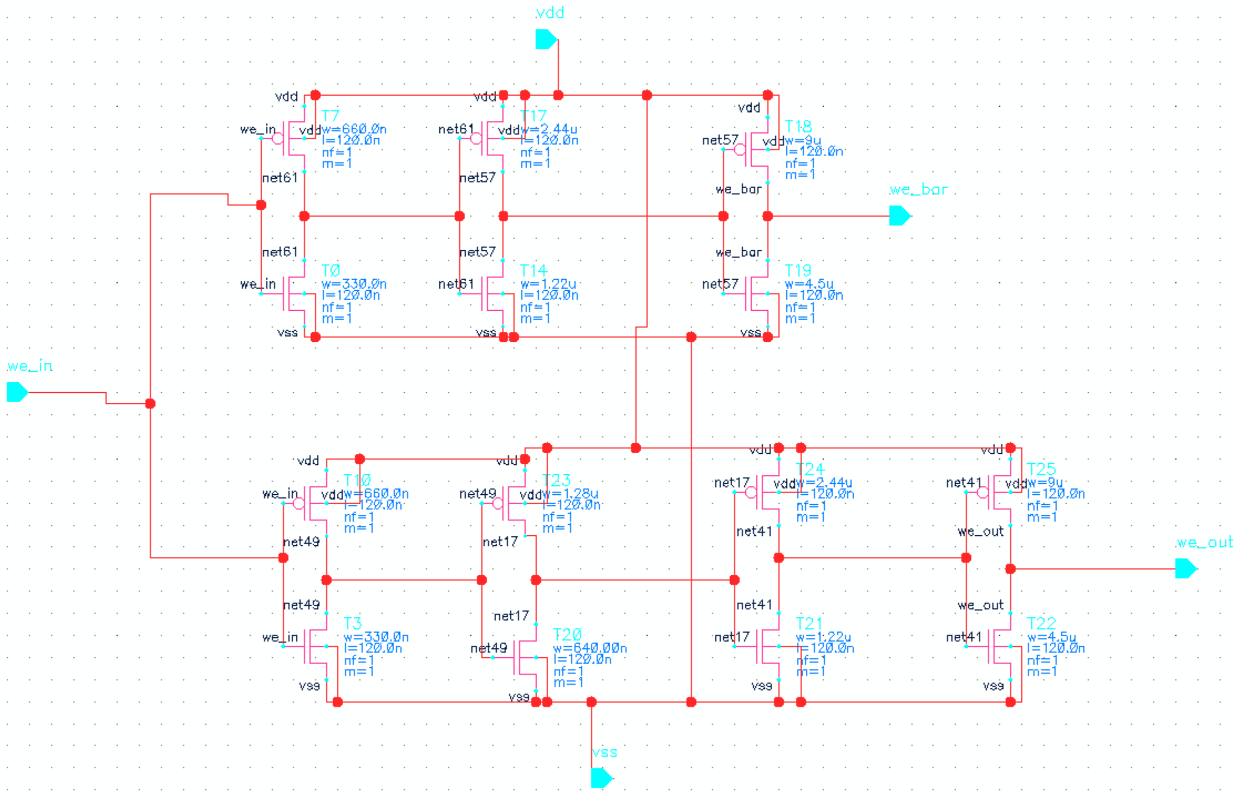
Cload= Cpoly + Cwire = 99.04fF= 49.52um

Number of stages, N = log 49.52/ log 3.6 ≈ 3

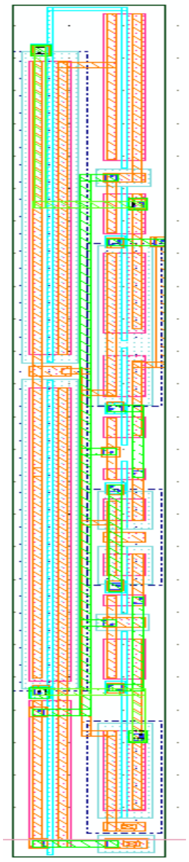
f = $F^{1/N}$ = 3.67

Inverter #1(rightmost): wp= 9.67um, wn= 4.83um
Inverter #2: wp= 2.54m, wn= 1.27um
Inverter #3: wp= 1.31um, wn=0.7um
Inverter #4: wp= 0.66um, wn=0.33um

The schematic of write buffer:



The layout of write buffer:



Clock Driver circuit [Library name: tutorial, Cell name: clkbuf]:

Since we have used a clocked precharge circuit to charge the bitlines, it is even necessary to size the clock buffer circuit. The sizing of the transistor is as follows:

Cpoly= 256fF

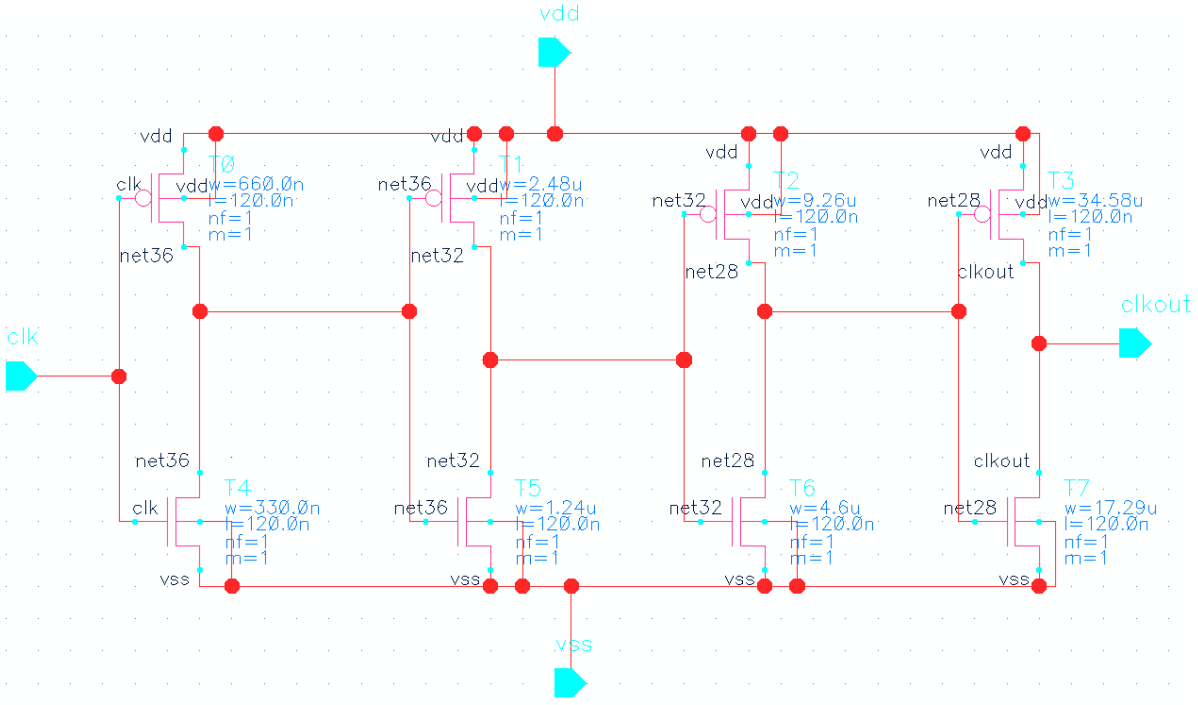
Cwire= 19.04fF

Cload= Cpoly + Cwire = 275.04fF= 137.52um

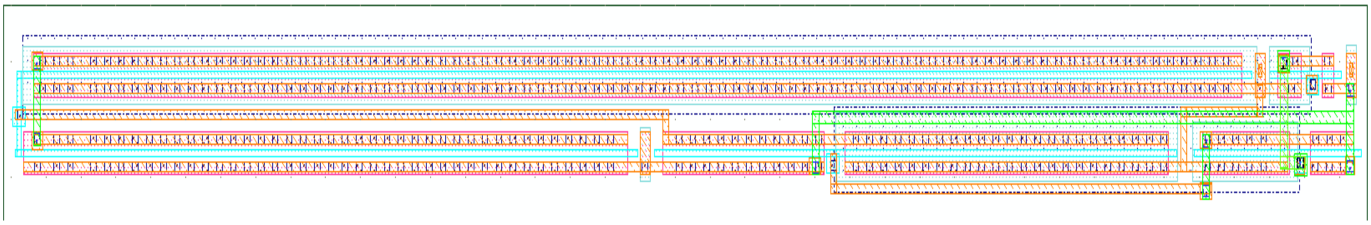
Number of stages, N = log 137.52/ log 3.6 ≈ 4

$f = F^{1/N} = 3.42$

The schematic of clock driver circit:



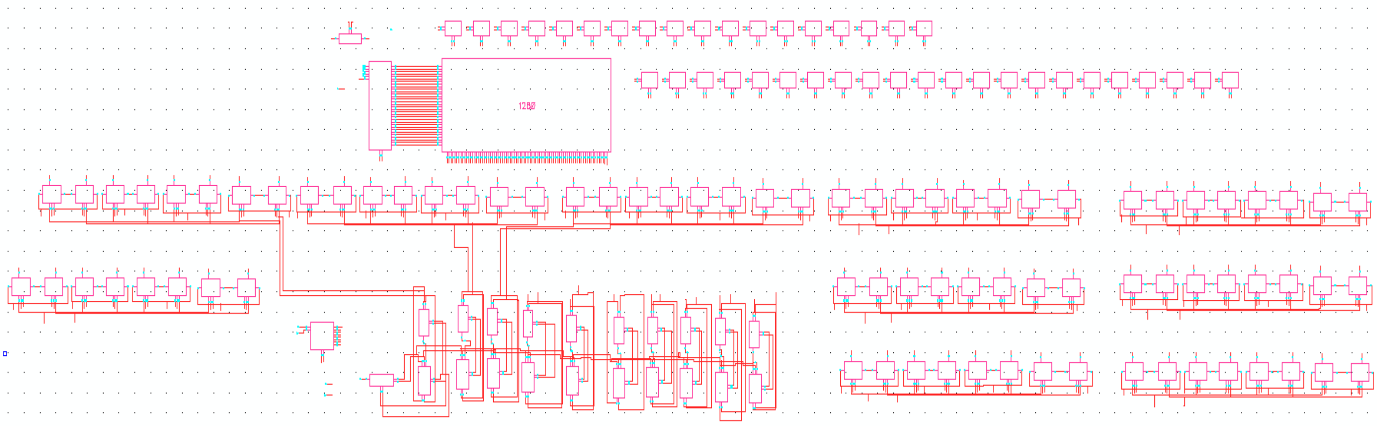
The layout of Clock Driver circuit:



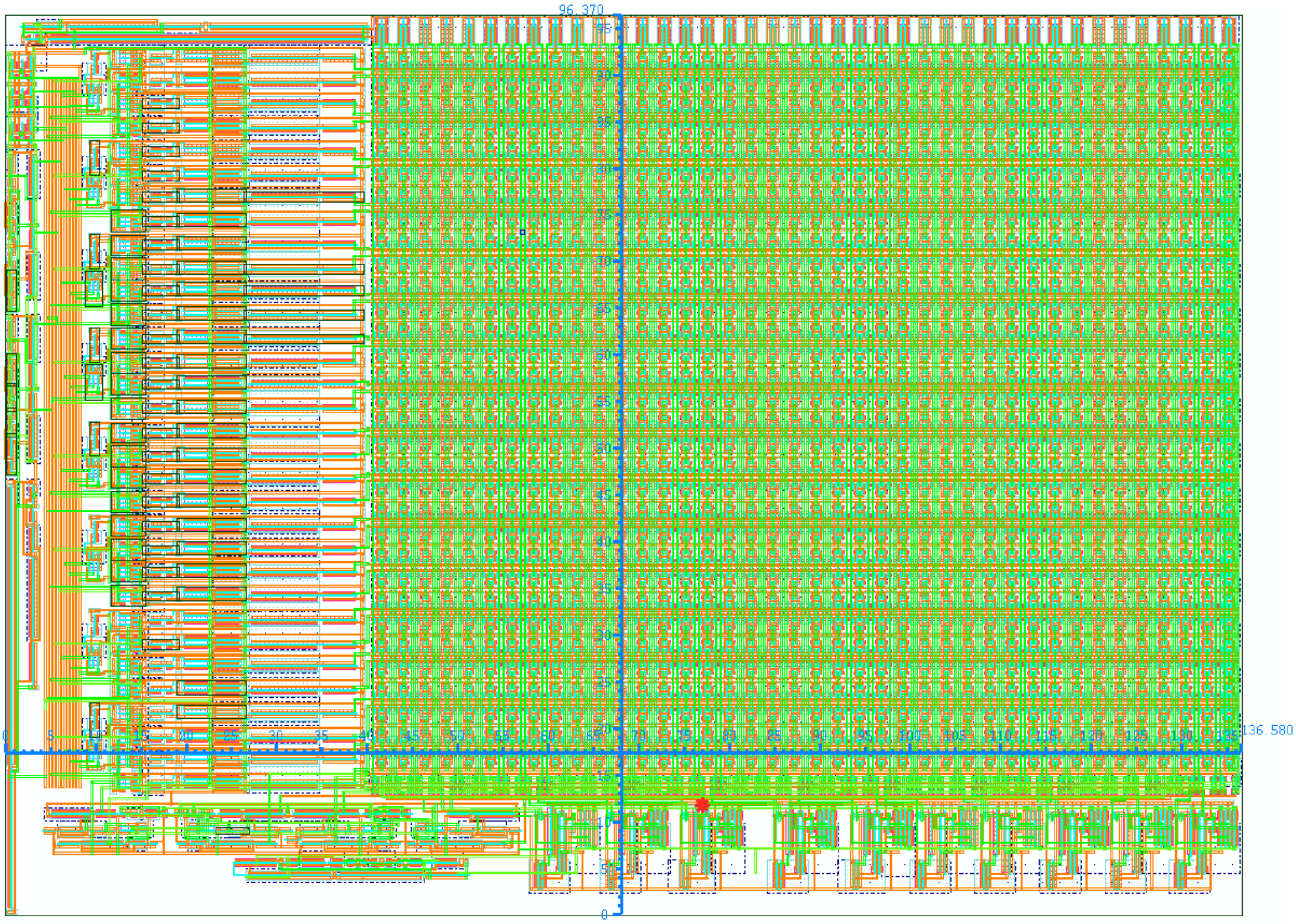
SRAM schematic and layout [Library name: tutorial, Cell name: SRAMf]:

Once all the peripheral circuits are designed, all of the units are then integrated to the memory cell array. The complete SRAM schematic and layout including precharge circuit, clock driver circuit, row decoder, memory array, column decoder, sense amplifier, write driver circuit and the write buffer circuit are as shown below.

Complete schematic of SRAM:



Complete layout of SRAM:

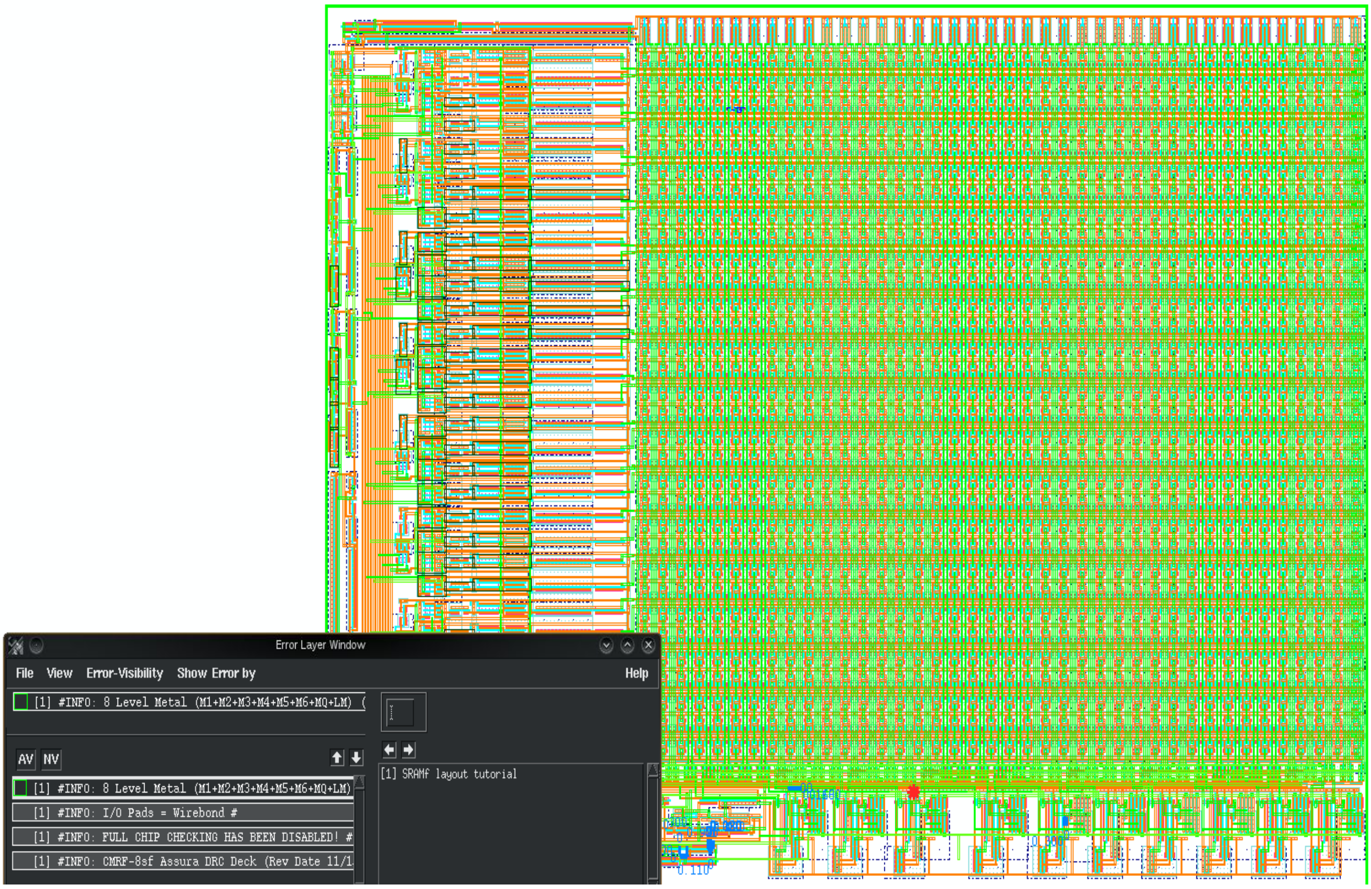


The total area of the design is **136.580 * 96.370 = 13162.2146um^2** . Therefore, total area that accounts for one bit is given by, **Area /bit = 13162.2146/ 1280 = 10.28um^2**

DRC and LVS reports

When the designed layout is simulated in cadence, there were no DRC errors. Then the functionality was tested by comparing the Layout versus Schematic(LVS). DRC report and matched LVS report are shown below.

The snapshot of DRC report:

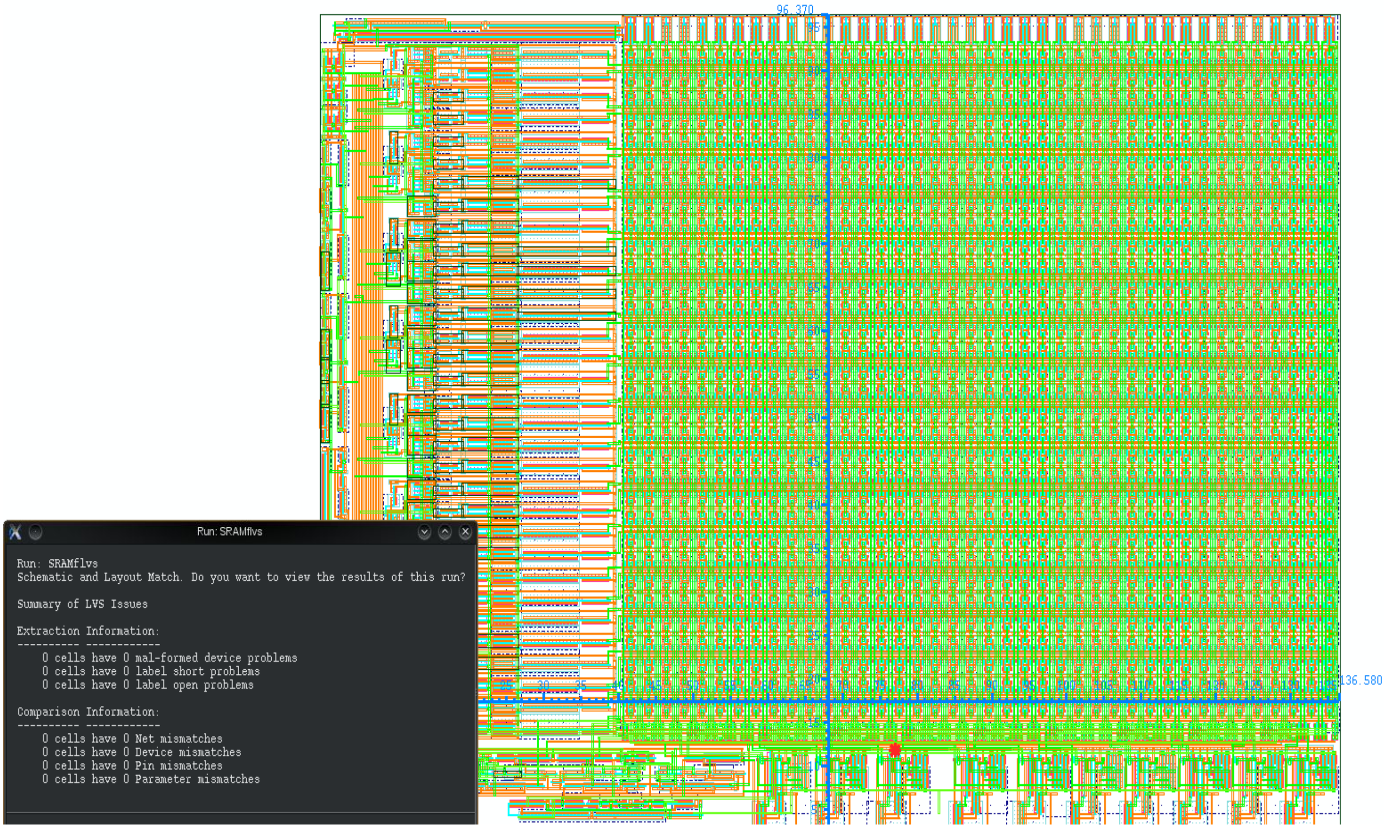


```
[1] #INFO: 8 Level Metal (M1+M2+M3+M4+M5+M6+MQ+LM) (
```

```
[1] #INFO: 8 Level Metal (M1+M2+M3+M4+M5+M6+MQ+LM)
[1] #INFO: I/O Pads = Wirebond #
[1] #INFO: FULL CHIP CHECKING HAS BEEN DISABLED! #
[1] #INFO: CMRF-8sf Assura DRC Deck (Rev Date 11/1
```

```
[1] SRAMf layout tutorial
```

The snapshot of LVS report:



```
Run: SRAMflvs
Schematic and Layout Match. Do you want to view the results of this run?

Summary of LVS Issues

Extraction Information:
---------- -----------
    0 cells have 0 mal-formed device problems
    0 cells have 0 label short problems
    0 cells have 0 label open problems

Comparison Information:
---------- -----------
    0 cells have 0 Net mismatches
    0 cells have 0 Device mismatches
    0 cells have 0 Pin mismatches
    0 cells have 0 Parameter mismatches
```

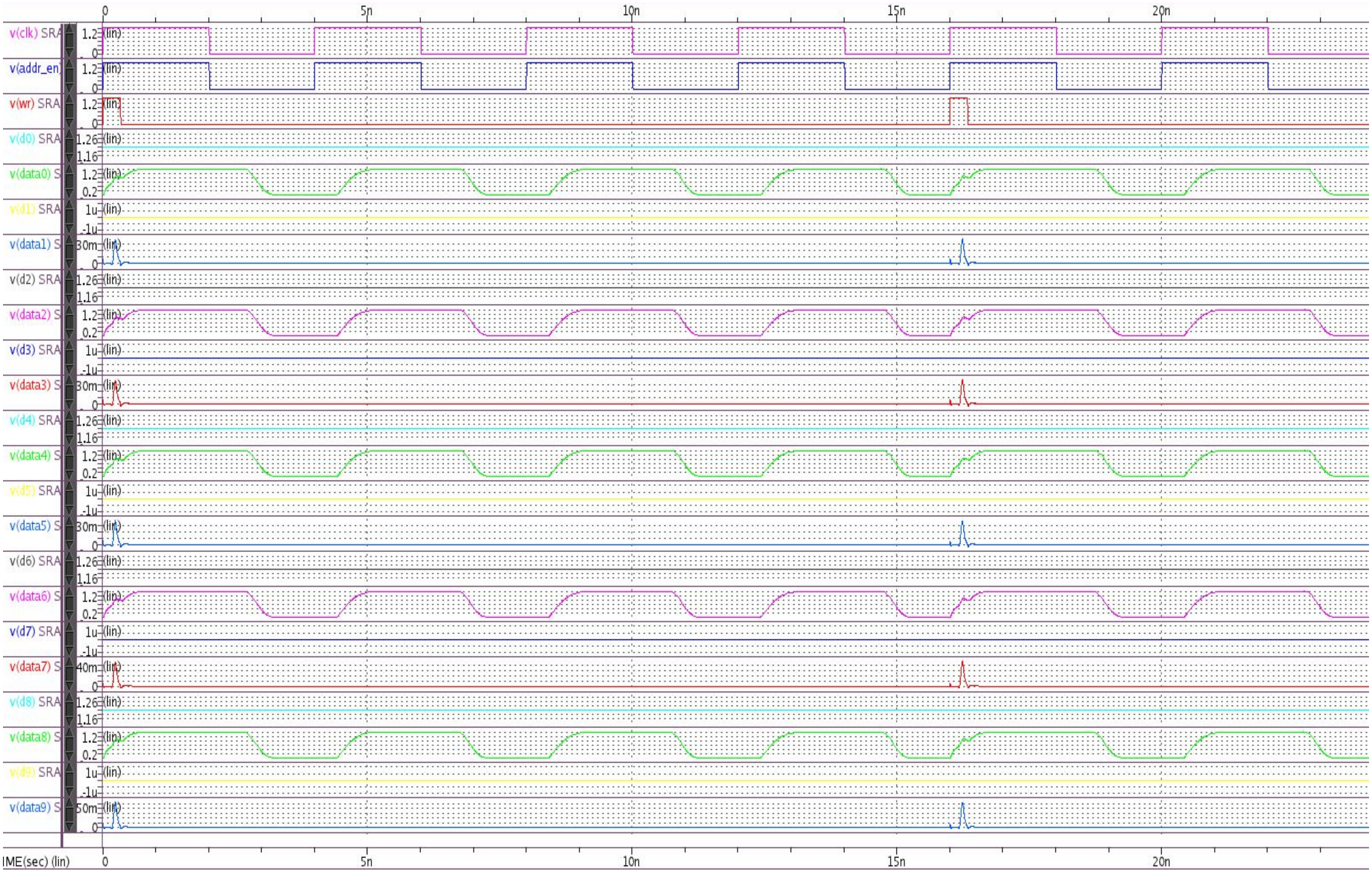
Results and Simulations

The functionality of the SRAM is tested by writing a 10 bit data word 1010101010 into the last row (row 11) and column 2nd of each super column of the design and then reading the written value in the next clock cycle.

When the clk is low, all the bitlines are precharged to VDD. During evaluation, when the clk and wr signals are high, the write operation takes place and word bits are written into the corresponding memory cell depending on the row and column address. The worst case write time delay is the time when the data bits that are being written are read in next cycle.

After one cycle, clk and wr go low, and the bitlines are again precharged high. During this clock cycle, as wr is low, the bits that were written in the previous clock cycle are read. It can be seen that all the bits are correctly read and the read time delay is also measured by 50 - 50% delay between addr_en and data bits being read. The simulated waveforms are as shown below. As shown below, the data bits are correctly written into the memory cell.

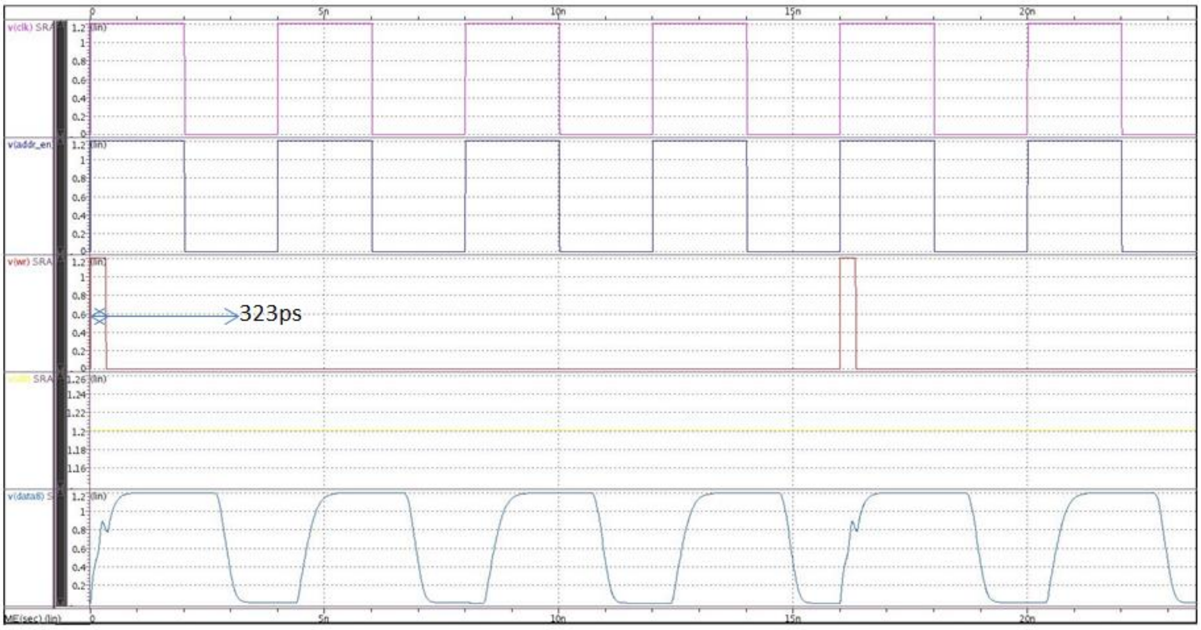
The simulated waveform:



Write time

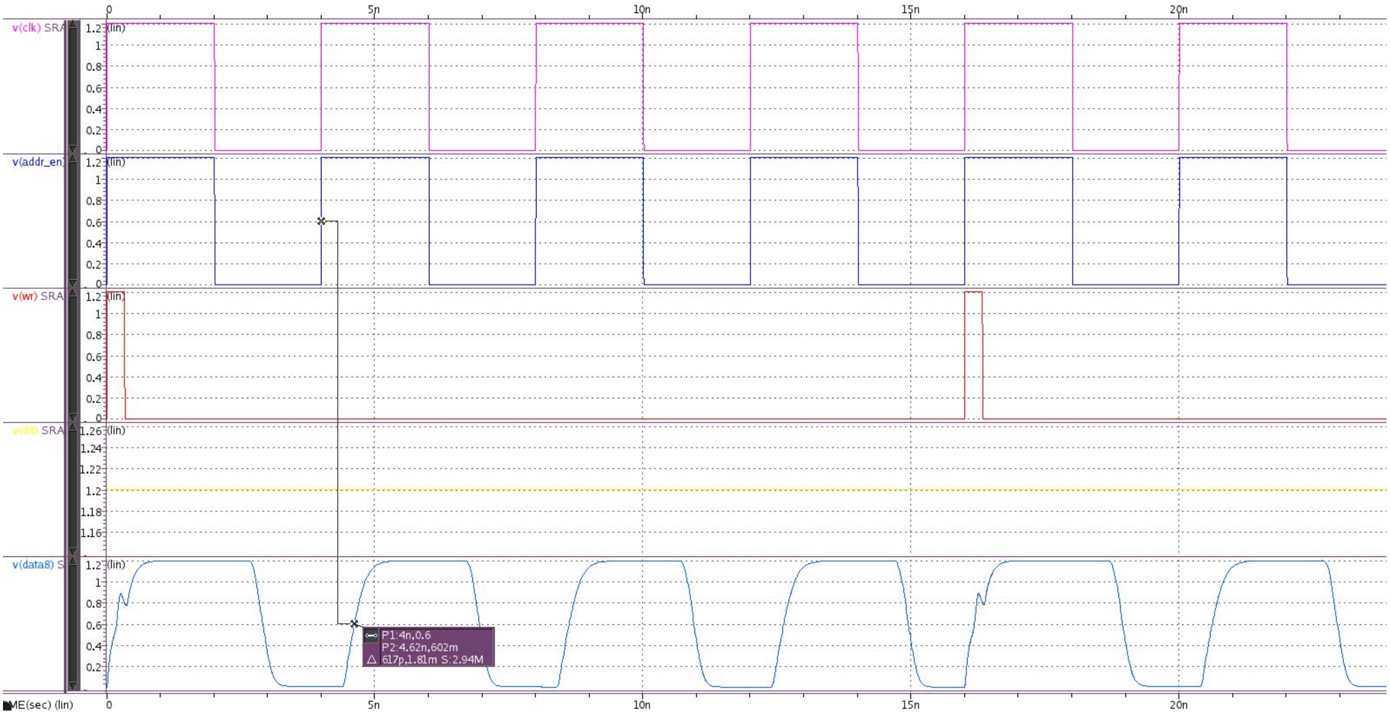
For the designed SRAM, it seems that writing a ‘1’ to the last bit in the first row yields the worst case write time since it is the farthest memory cell to access both by row decoder and also the column decoder and hence it suffers a large load capacitance. The worst case write time delay is the time when the data bits that are being written are read in next cycle.

The worst case write time is found to be: 323ps



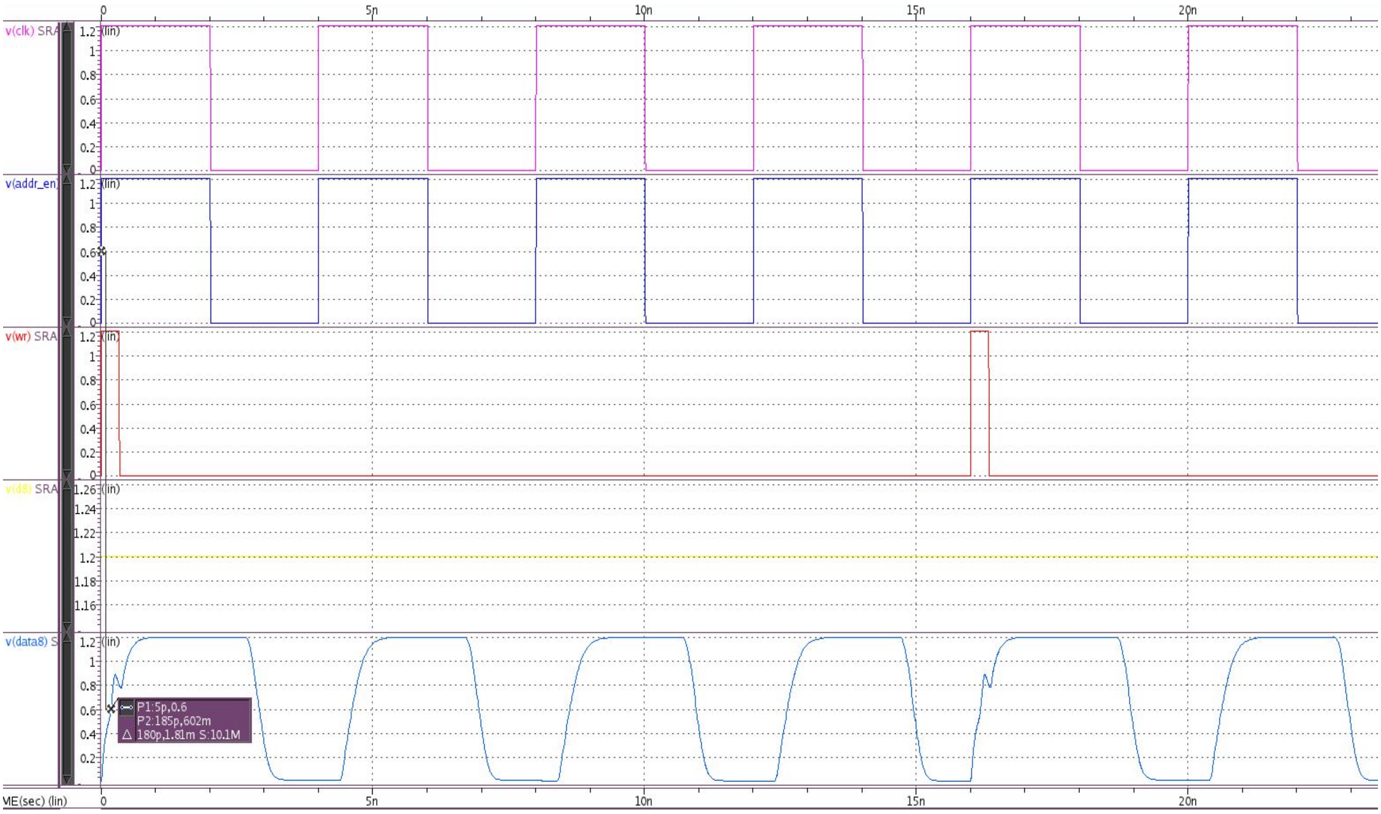
Read time

Worst case read time is found by reading the data values from the word line 0 and bit 32. It is measured by 50 - 50% delay between addr_en and data bits being read. The worst case read time is found to be: 617ps



Operating Frequency

Write time measured from the instant addr_en is activated to the instant the data is written into the cell is found to be 180ps. Operating frequency= 1/(2*180p) = 2.7 GHz



Conclusion

Aspect ratio = 0.70

All the inputs are sized to minimise the delay

Worst case Write time = 323ps

Worst case Read time = 617ps