

May 7, 1997

# **Design of a 32-bit SRAM using MOSIS 0.6 $\mu m$ CMOS technology**

Rajamohana Hegde, Soo Lee, Michael Liwanag and John Strologas

## A. Introduction

In this paper we describe the procedure followed for the transistor level design and layout of a 32-bit Static-RAM array. We designed it in such a way that it is as fast as possible with the constraint that the average Power consumption during read or write it is less than 1 mW. The average Power is defined as the sum of the Power consumption in the 4 possible operations (Read 1, Read 0, Write 1, Write 0), divided by four. In every step of our design we simulated the circuits using SPICE. We also used the same simulation program to verify the correct performance of our layout, according to the specifications, by simulating the SPICE code extracted by the IC station (the Computer Aided Design software we used to produce our layout).

## B. The memory cell

The most important building block of any RAM or ROM is the memory cell. In our design we used the standard SRAM memory cell, which can be seen in Figure 1.

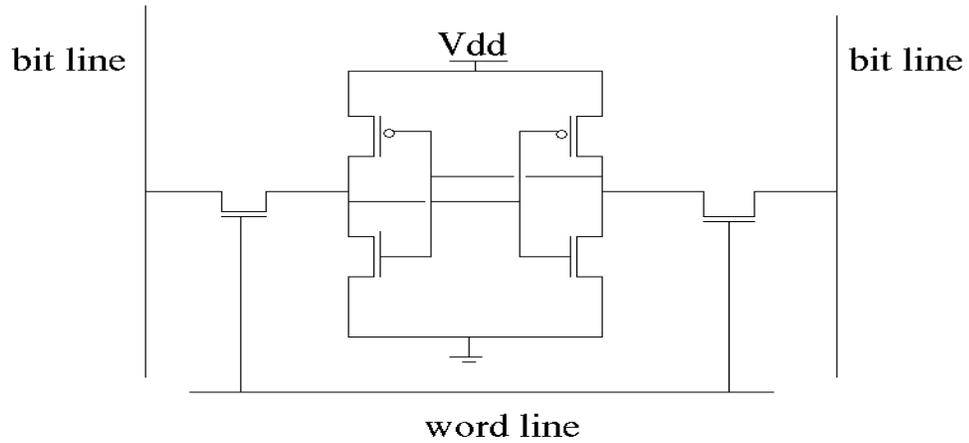


Figure 1: Standard configuration of the SRAM memory cell

While obtaining the layout of the memory cell, we moved all the pMOS transistors up and all the nMOS transistors down to minimize the area covered. In Figure 2, we the SRAM memory cell after this topological modification is shown.

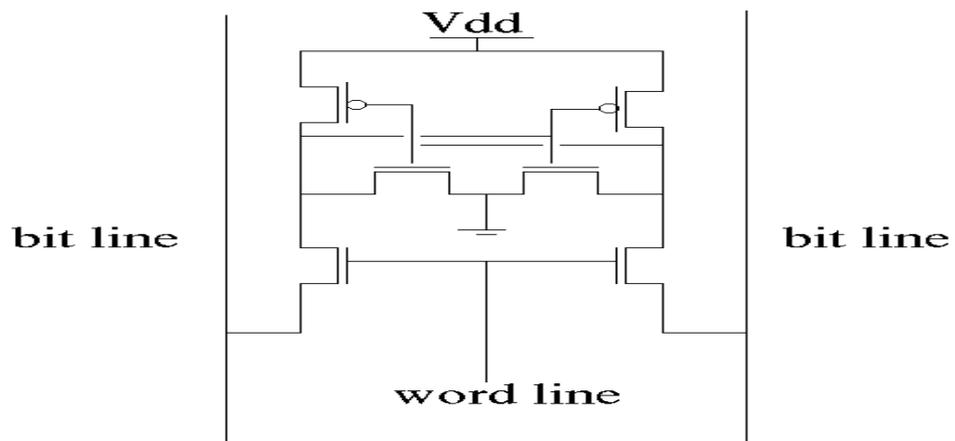


Figure 2: Configuration of the SRAM memory cell useful for minimum area layout

The final layout is shown in Figure 3.



```

cp_1 1 0 6.986f ic=0v
cp_2 2 0 7.042f
cp_5 5 0 7.173f
cp_7 7 0 3.603f
* distributed parasitics:
* devices:
m0 2 4 1 2 p l=0.6u w=1.8u ad=3.901p as=3.901p pd=6.134u ps=6.134u
m1 2 1 4 2 p l=0.6u w=1.8u ad=3.901p as=3.901p pd=6.134u ps=6.134u
m2 1 7 3 0 n l=0.6u w=1.8u ad=8.713p as=3.853p pd=8.781u ps=6.081u
m3 5 4 1 0 n l=0.6u w=4.5u ad=10.98p as=17.46p pd=9.381u ps=16.58u
m4 4 1 5 0 n l=0.6u w=4.5u ad=17.46p as=10.98p pd=16.58u ps=9.381u
m5 4 7 6 0 n l=0.6u w=1.8u ad=8.713p as=3.853p pd=8.781u ps=6.081u

vg 5 0 0v
vdd 2 0 5v

vw1 7 0 PULSE(0 5 8n 0 0 5n 40n)
vb1 6 0 PULSE(5 0 5n 0 0 5n 40n)

fp 30 0 vdd 0.0125
rp 30 0 100k
cp 30 0 100p

.model n NMOS LEVEL=3 PHI=0.700000 TOX=1.0000E-08 XJ=0.200000U TPG=1 VTO=0.7812
+DELTA=2.4510E-01 LD=4.0510E-08 KP=1.8847E-04 UO=545.8 THETA=2.5170E-01
+RSH=2.1290E+01
+GAMMA=0.6200 NSUB=1.3810E+17 NFS=7.0710E+11 VMAX=1.8610E+05 ETA=2.2420E-02
+KAPPA=9.6720E-02 CGDO=3.66E-10 CGSO=3.66E-10 CGBO=4.0161E-10 CJ=5.4E-04 MJ=0.6
+CJSW=1.5000E-10 MJSW=0.32 PB=0.99

```

```

.model p PMOS LEVEL=3 PHI=0.700000 TOX=1.0000E-08 XJ=0.200000U TPG=-1
+VTO=-0.9197
+DELTA=2.4830E-01 LD=6.7120E-08 KP=4.4546E-05 U0=129.0 THETA=1.7800E-01
+RSH=3.4290E+00
+GAMMA=0.5230 NSUB=9.8260E+16 NFS=6.4990E+11 VMAX=3.0560E+05 ETA=1.7820E-02
+KAPPA=6.3410E+00 CGDO=3.66E-10 CGS0=3.66E-10 CGB0=4.2772E-10 CJ=9.3191E-04
+MJ=0.51 CJSW=1.5E-10 MJSW=0.193 PB=0.95
.tran 1ns 80ns uic
.end

```

(Node 30 is used to measure the power consumption)

In Figure 4 we can see the D voltage during write 1 (it goes high) and in Figure 5 we can see the  $\bar{D}$  during the same operation (it goes low). The blue line is the write signal in the word line. Figure 6 gives as the power consumption for just one SRAM cell which is about .1 mW in our case.

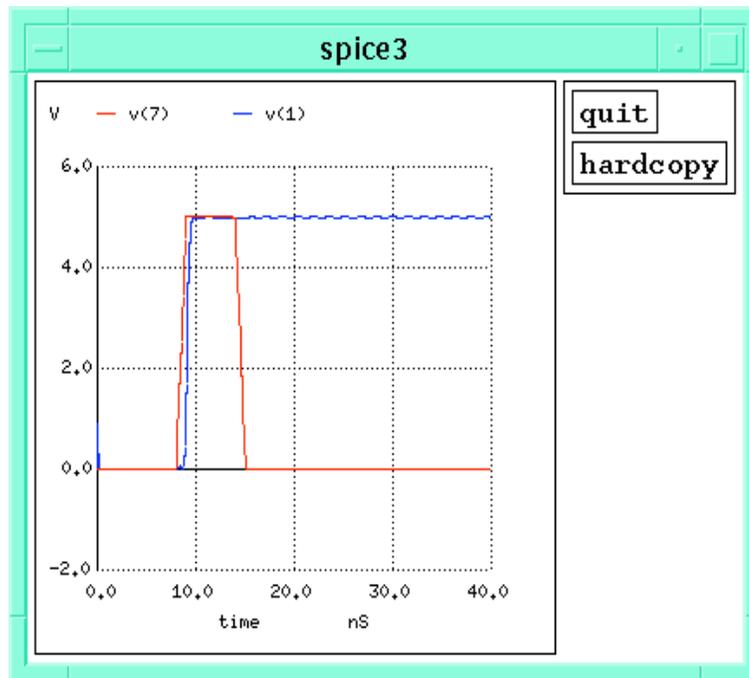


Figure 4: During Write 1, D goes high

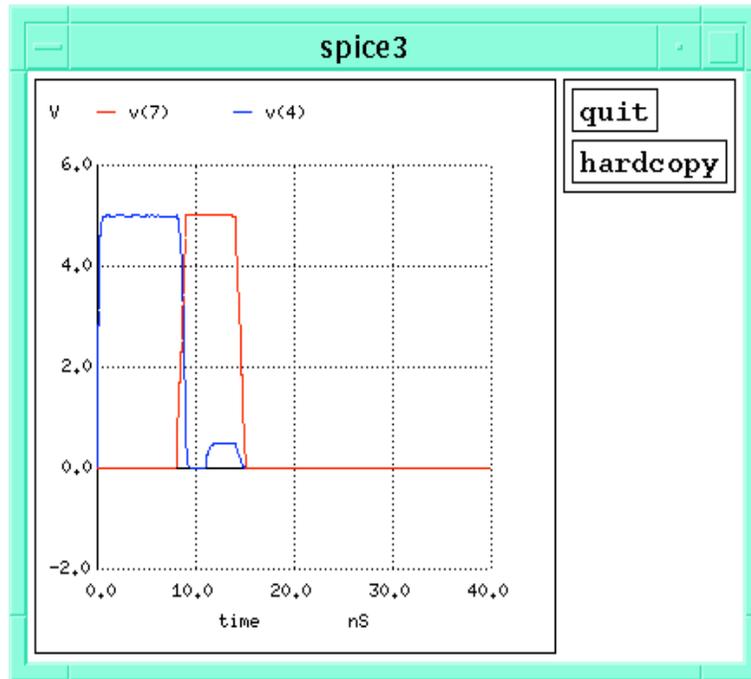


Figure 5: During Write 1,  $\bar{D}$  low

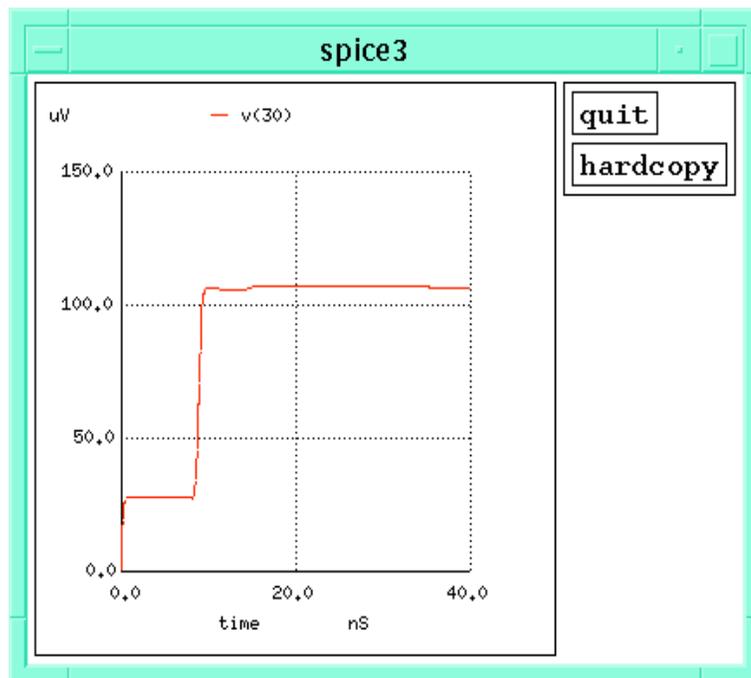


Figure 6: The power consumption of our SRAM cell is 0.1 mW

## C. Memory array

To design the memory array, using the memory cell we already designed, we must partition our array correctly. We must thus study the time delay of the array, under different configurations.

The row time delay for an array with  $2^x$  columns (where  $x$  is an integer between 0 and 5 in our case) is given by the formula:

$$t_{row} = 0.38 \cdot R_{row}^{cell} \cdot C_{row}^{cell} \cdot (2^x)^2 \quad (1)$$

where,

$$\begin{aligned} R_{row}^{cell} &= 2.6\Omega \cdot (\text{number of polysilicon word line squares per cell}) \\ &= 2.6 \cdot 49\Omega = 127.4\Omega \end{aligned} \quad (2)$$

is the row resistance per unit cell and

$$\begin{aligned} C_{row}^{cell} &= 2 \cdot C_{ox} \cdot W_p \cdot L_p + C_{poly \text{ line}} \\ &= 2 \cdot \frac{\epsilon_{ox}}{T_{ox}} \cdot W_p \cdot L_p + 8.7 \cdot 10^{-5} F/m^2 \cdot 196(\mu m)^2 \\ &= 9.83 fF \end{aligned} \quad (3)$$

is the row capacitance per unit cell.  $W_p$  and  $L_p$  are the width and length of the pass transistors. Both of them are connected to the word line and they contribute to the total capacitance of the word line per cell.

Using equations (1), (2) and (3) we get

$$t_{row} = 1.197 \cdot 10^{-12} \cdot 2^{2x} \quad (4)$$

Now we will calculate the column delay time. This is given by the formula:

$$t_{col} = C_{col}^{cell} \cdot \Delta V / I \cdot 2^{(5-x)}(5)$$

where  $C_{col}^{cell}$  is the column capacitance per unit cell,  $\Delta V$  is the voltage drop of one of the two bit lines (during a Read or Write operation), which can be detected by the

sense amplifier. A good estimation is  $\Delta V=0.5$  V. On the other hand  $I$  is the current which is responsible for this drop and it is around 1 mA, as we can see from Figure 7 (which is the result of the SPICE simulation of one SRAM cell).

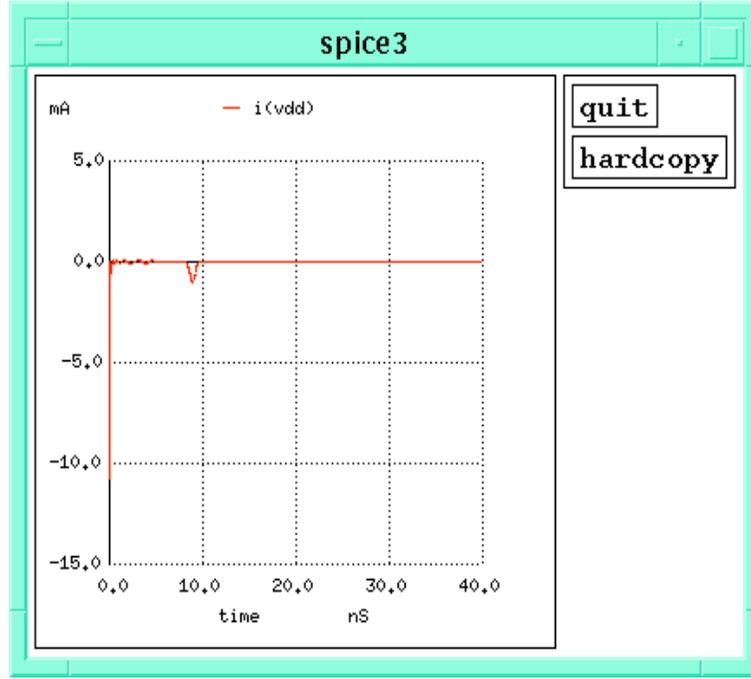


Figure 7: The discharge current during a Write operation

The column capacitance per memory cell is the sum of the pass-transistor parasitic capacitance and the metal line capacitance, as we can see from the following relation:

$$C_{col}^{cell} = C_{db} + C_{gd} + C_{metall}$$

$$= 5.806 fF + 114 \cdot 0.6 \mu m \cdot 2 pF/cm = 19.486 pF(6)$$

where the parasitic capacitance of the pass transistor is given in the SRAM cell extracted SPICE code.

Using formulas (5) and (6) we get:

$$t_{col} = 9.743 \cdot 10^{-12} \cdot 2^{(5-x)} \quad (7)$$

We can now complete our partition. We plot the sum of the row and column time as a function of  $x$  and we choose the minimum to be our design's delay time.

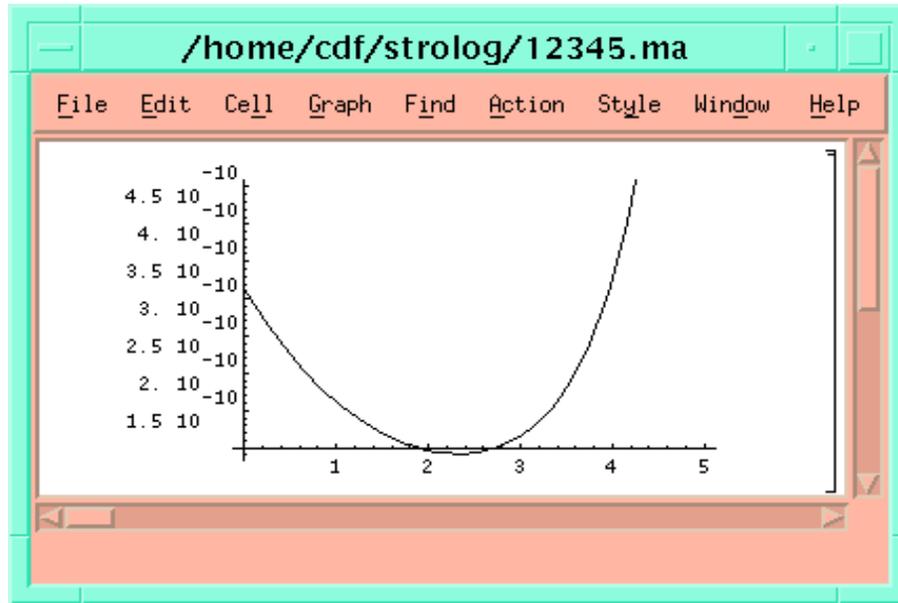


Figure 8: The plot of the total delay time as a function of the partition variable  $x$

We see that the delay time is minimum for  $x$  close to 2, that means that our array must have  $2^2 = 4$  columns. So our final result is that the SRAM array is going to be (8 rows by 4 columns) [or (3 by 2) in the address bits].

In Figure 9 we can see our array. We chose pull-up pMOS transistors of small (W/L) to reduce the power consumption (W/L=1.2/6).

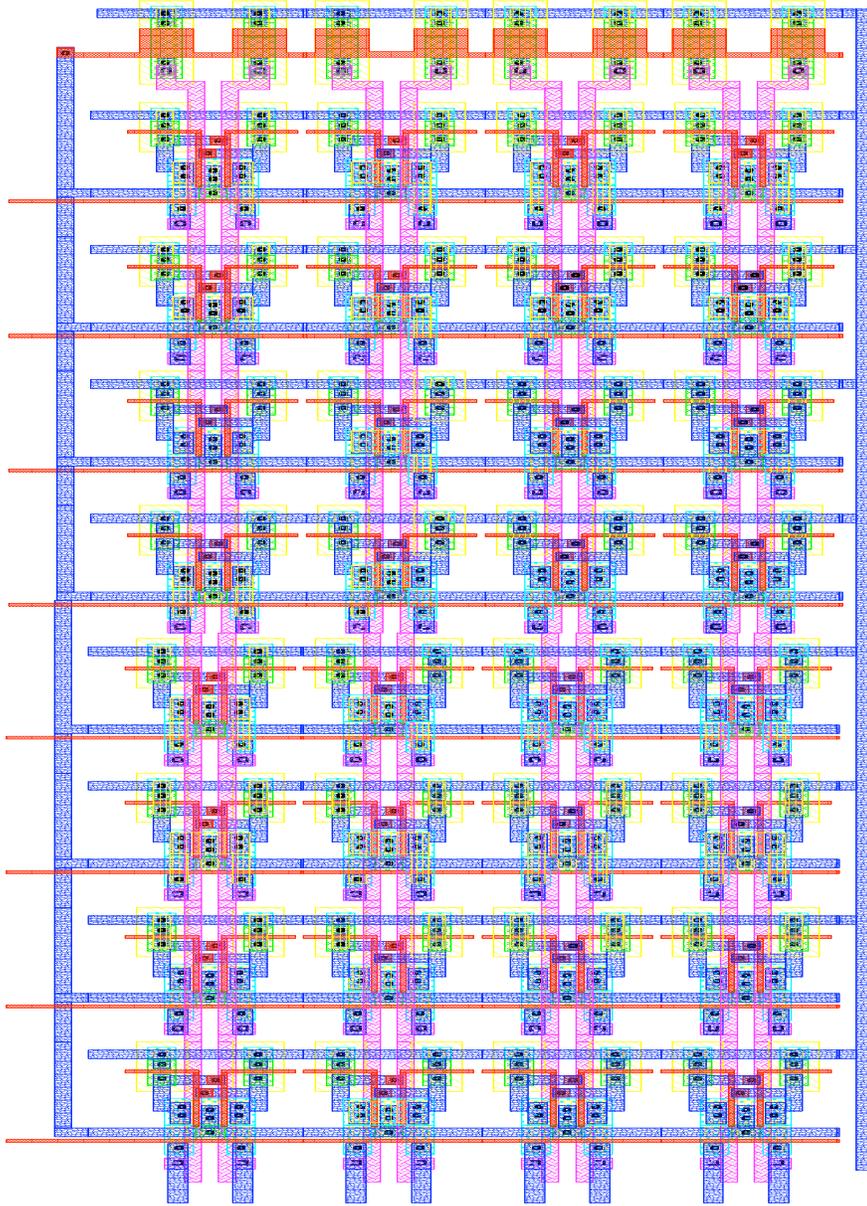


Figure 9: The SRAM array

## D. The Row and Column Decoder

In the row and column decoder, there is one inverter to obtain  $CLK$  and  $\overline{CLK}$ , three  $D_{latches}$  for the row decoder and two  $D_{latches}$  for the column decoder. First we had to decide which model to use for our project. There are two types of models which are  $NAND_{model}$ , and  $NOR_{model}$  for row and column decoder. We decided to use the  $NOR_{model}$ , because it consumes the power less than  $NAND_{model}$ . (the worst case of NOR is less than NAND). Also we had two choices for the  $NOR_{model}$ . We could use the static gates or dynamic gates. Because of low power consumption, we decided to use the dynamic one. However, we found out that the input can not change while the clock is high, so in the dynamic gate, we connected  $\overline{CLK}$ . Because of the connection of the  $\overline{CLK}$ , we had to change the model to  $NAND_{model}$  and add the inverter at the end to get the right output. The schematics of the row and column decoders we designed can be seen in Figures 10 and 11 respectively.

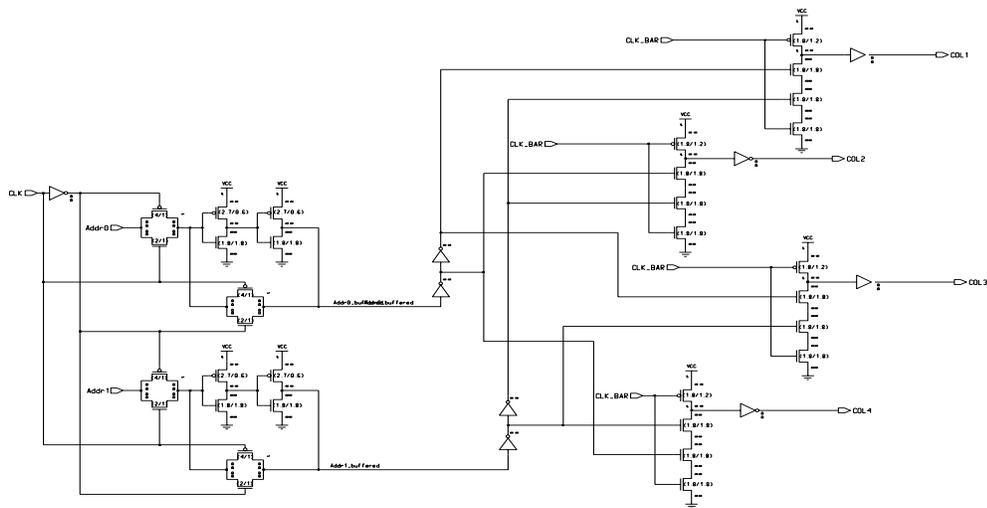


Figure 10: The Column Decoder Schematic



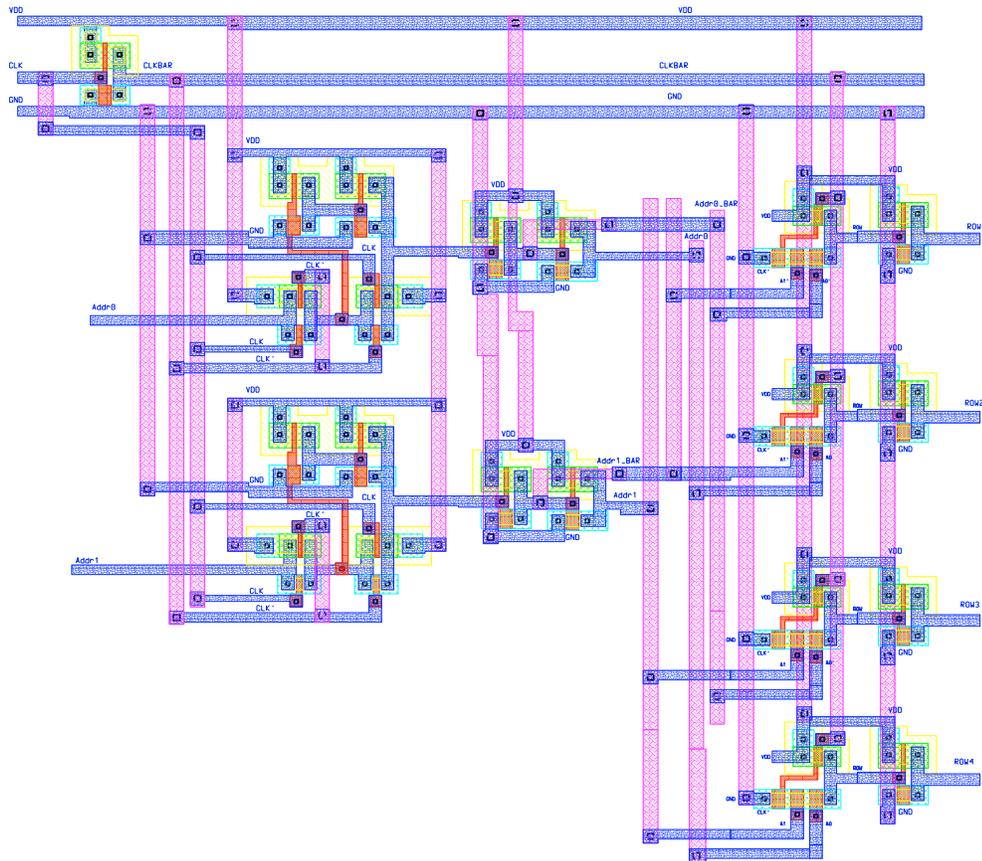


Figure 12: The column decoder layout

Finally we extracted the SPICE files of the decoders (using the IC station) and we simulated them. The following is a part of the SPICE file where the inputs are specified.

```
vdd 2 0 5
vgnd 1 0 0
vA1 5 0 5 pulse(0 5 50ns 1ns 1ns 40ns 80ns)
vA0 6 0 0 pulse(0 5 30ns 1ns 1ns 20ns 40ns)
vclk 3 0 5 pulse(5 0 20ns 1ns 1ns 10ns 20ns)
```

The Figures 14 and 15 show the output of the row and column decoders respectively.

Note that  $v(3)$  is CLK and  $v(30)$  is the output. When the clock is high the decoder read the input and when the CLK is low ( $\overline{CLK}$  is high) it evaluates the

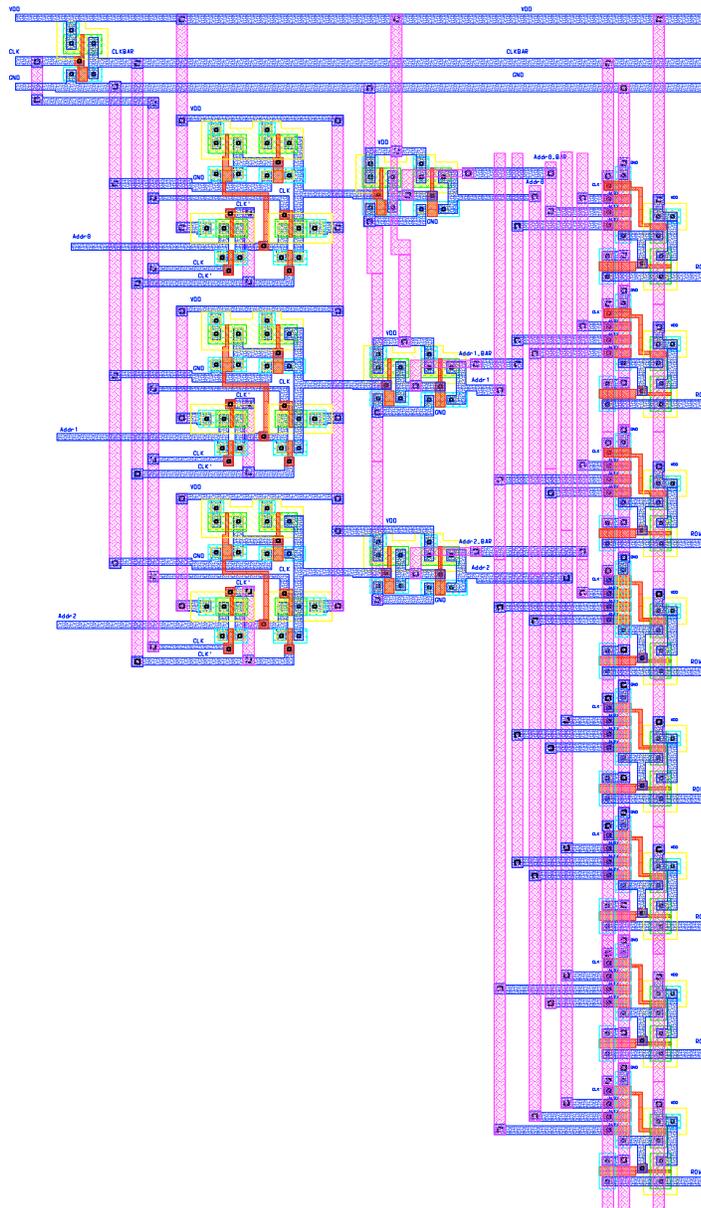


Figure 13: The row decoder layout

output. Between 40ns to 60ns the output is high (5 volt) while the clock is low. It is easy to see the all the outputs are high while  $\overline{CLK}$  is high. Figures 16 and 17 give us the power consumption for the row and the column decoders respectively. The average Power consumption equals the height of each small step and it is considerably low - less than 0.2 mW.

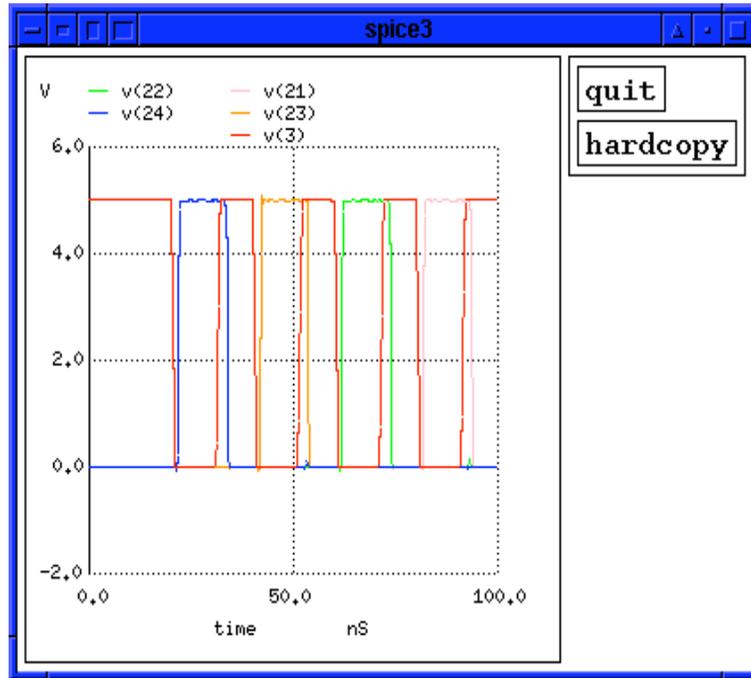


Figure 14: Output of the row decoder versus the clock.

## E. The Sense Amplifier

The configuration of the differential amplifier employed in this circuit to sense the output of the RAM cell is shown in figure 18. It consists of a cross coupled pair of PMOS transistors that rapidly produce a differential output signal. In the original sense amplifier circuit, the body terminals of the two n-mos devices are connected to their respective source terminals. This can be done only in a technology that allows creation of both n and p tubs. As we employ a p-well process, the circuit has been modified by connecting the substrates of all the nmos devices to ground. To offset the degradation due to the grounded substrates, the relative sizing of the transistors has been modified. The transistor sizes employed are shown in the schematic.

The layout of the sense amplifier and its performance plots are shown in figures 19 and 20 respectively. The voltages v(1) and v(2) are the outputs and v(6), v(7) are the outputs. Note that the two outputs do not provide a complete voltage swing of 0 to 5v. Hence, inverted output of the sense amplifier is connected to a inverter buffer

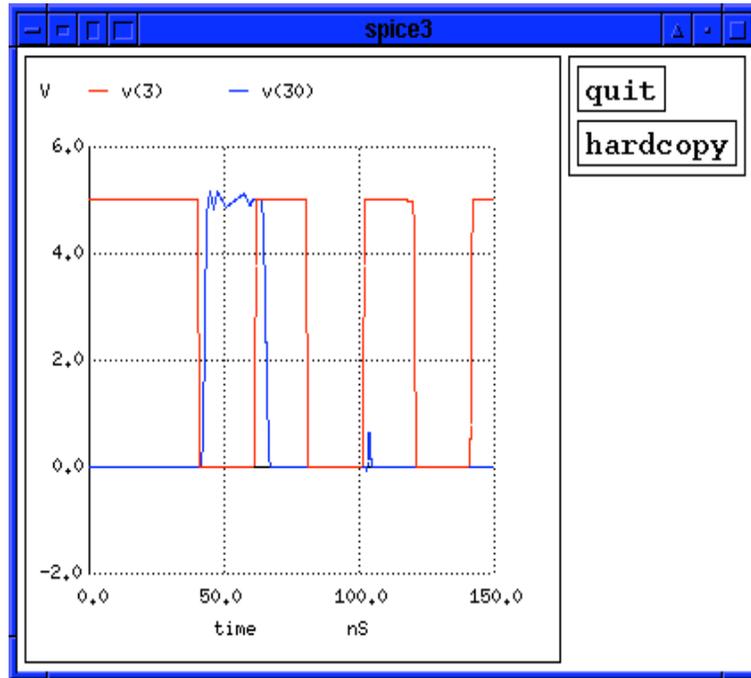


Figure 15: Output of the column decoder versus the clock

in order to improve the voltage swing.

## F. The Final Design

After we designed all the parts of our SRAM, we will put everything together. The block diagram of the whole SRAM can be seen in Figure 27. The corresponding layout is in Figure 28. We extracted the SPICE file and we ran it. The operation is exactly what we expected for all the four operations (Write 1, Write 0, Read 1, Read 0). For example Figures 21 and 22 show the  $DATA$  and  $\overline{DATA}$  for the Write 1

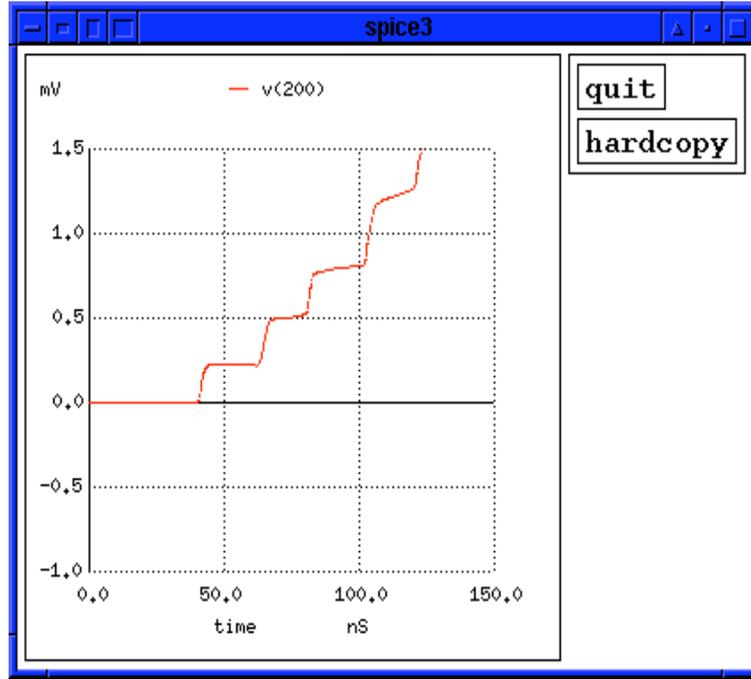


Figure 16: Power consumption for the row decoder

operation. Figures 24 and 25 show the bit-lines  $C$  and  $\bar{C}$  respectively for the Read 1 case ( as we expected,  $\bar{C}$  is pulled down - and let us actually Read 1- and  $C$  remains high ). The blue line (v(3)) is the clock of our dynamic design.

The Power consumption for a Read operation is given in Figure 23. The Power consumption for a Write operation can be seen in Figure 26. We see that  $Power_{Write1} = Power_{Write0} \approx 1.2mW$  and  $Power_{Read1} = Power_{Read0} \approx 0.7mW$ . So the overall Power consumption is about 0.95 mW (the sum for all operations divided by four). This is a very satisfactory result, since we met the specifications.

Finally we include the timing diagrams for our circuit. Figures 29 and 30 show the Write and Read timing diagrams respectively. Note that as we have employed dynamic logic gates in our decoders, the addresses are valid only for half the clock cycle. Hence time period of the clock is adjusted such that both Read and Write

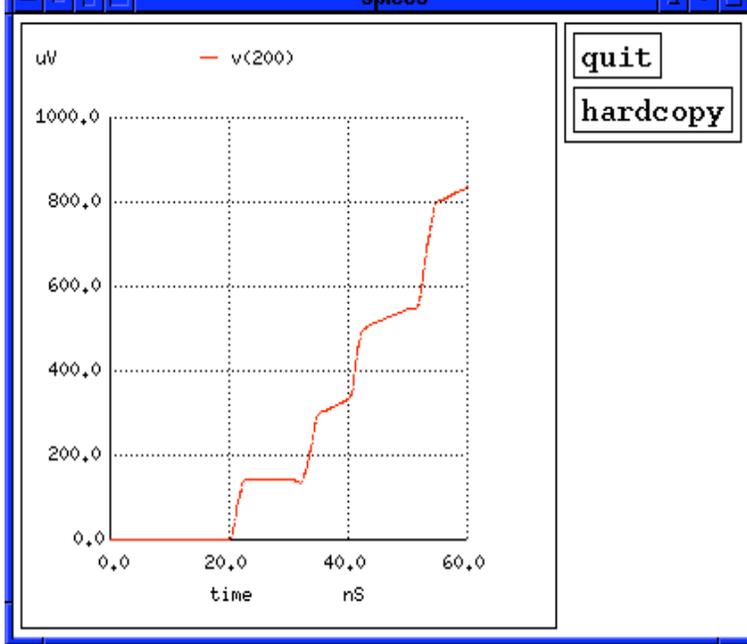


Figure 17: Power consumption for the column decoder

operations can be accomplished in half a clock cycle. Two control signals, namely Read Enable and Write Enable are provided to have the data available for Read and Write operations at appropriate times. Both the input and output signals have been buffered to prevent loading on external as well as internal circuitry.

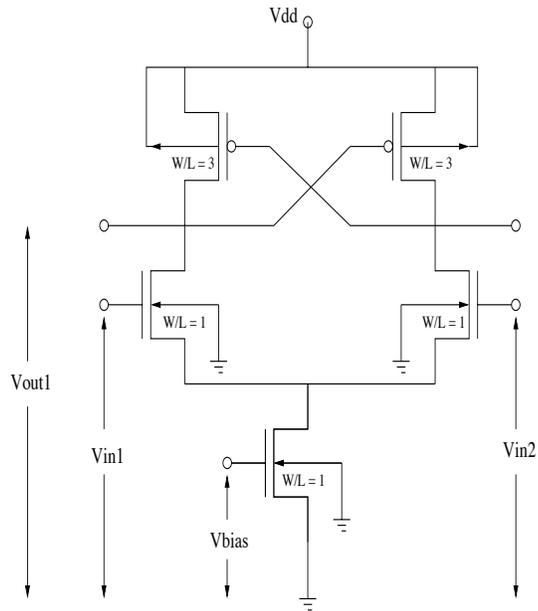


Figure 18: Schematic of the differential sense amplifier

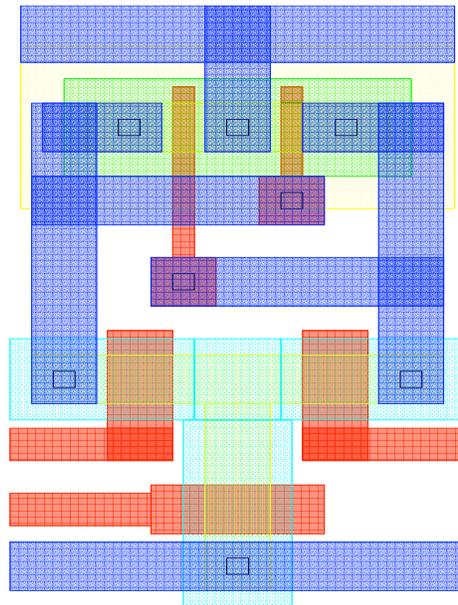


Figure 19: Layout of the differential sense amplifier

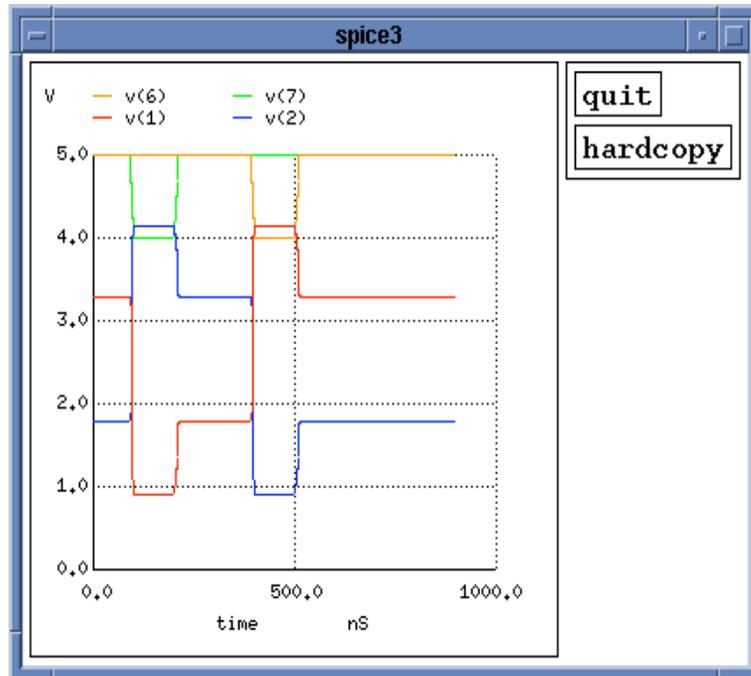


Figure 20: Performance of the sense amplifier

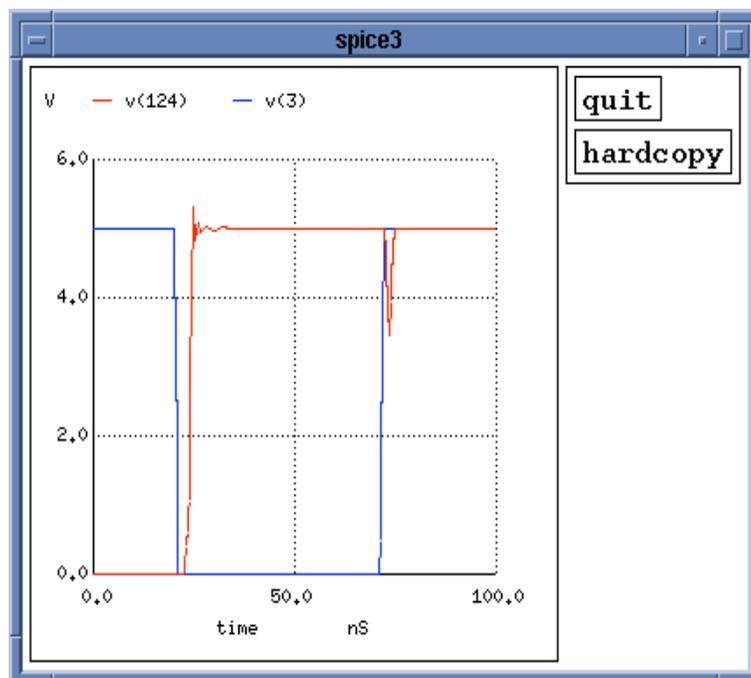


Figure 21: During Write 1, D goes high

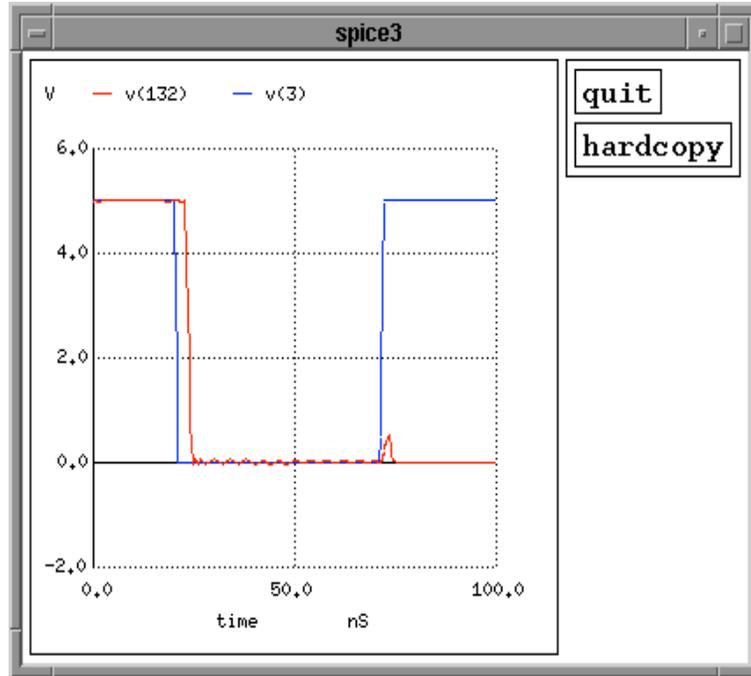


Figure 22: During Write 1,  $\bar{D}$  goes low

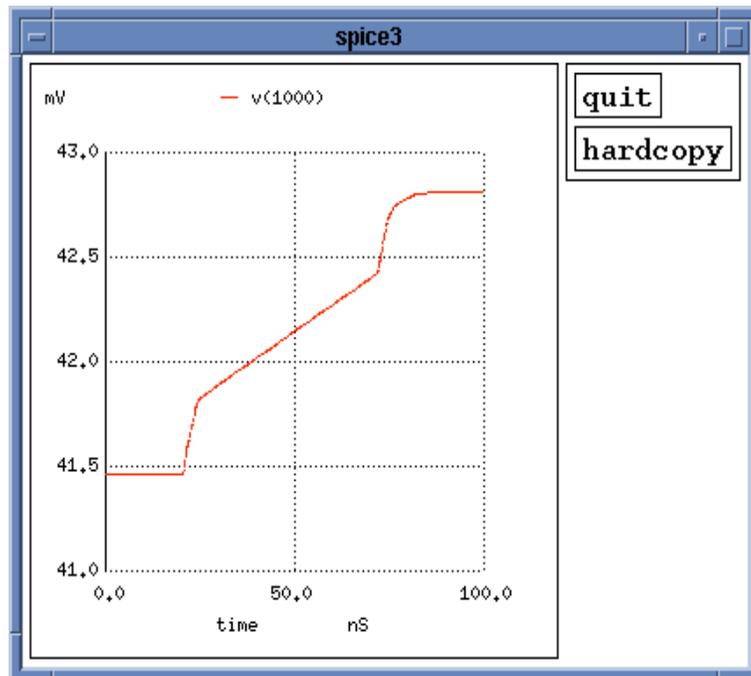


Figure 23: Power consumption for a Write operation

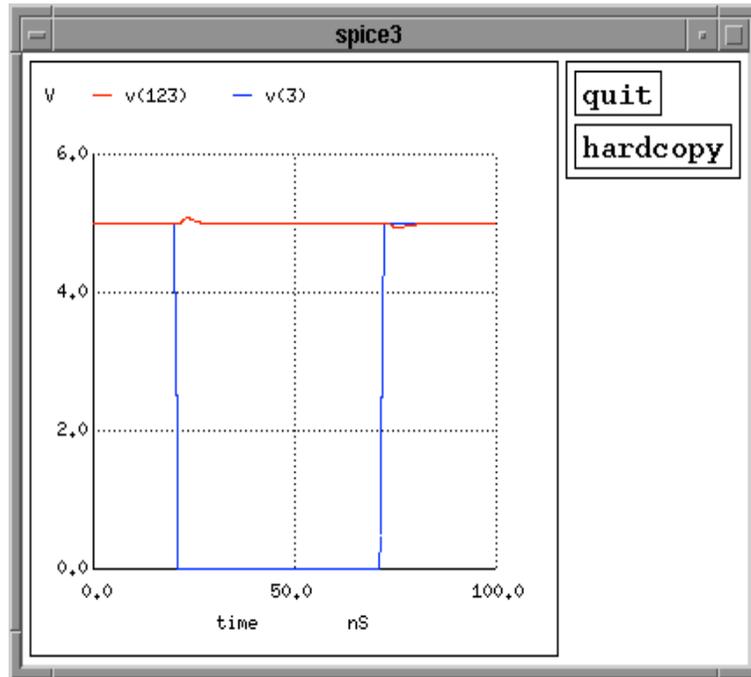


Figure 24: During Read 1, C remains high

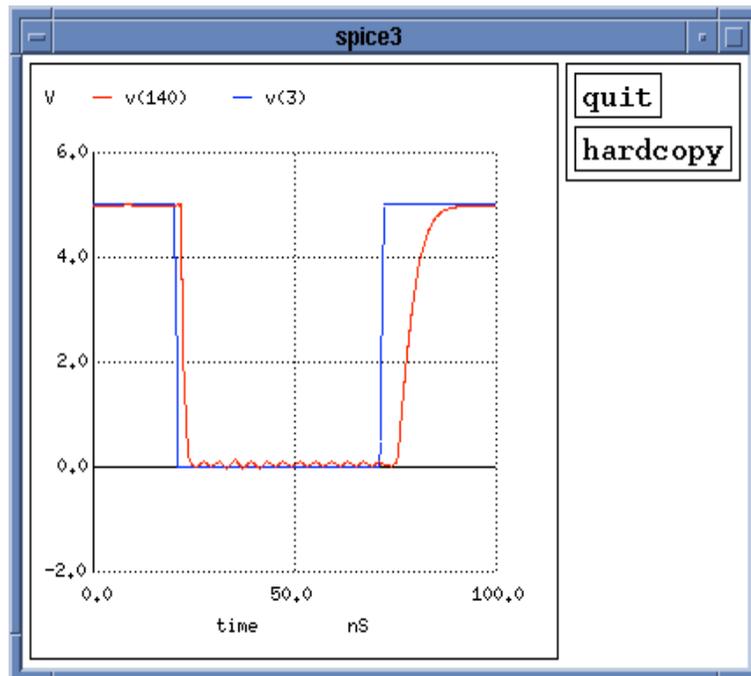


Figure 25: During Read 1,  $\bar{C}$  goes low

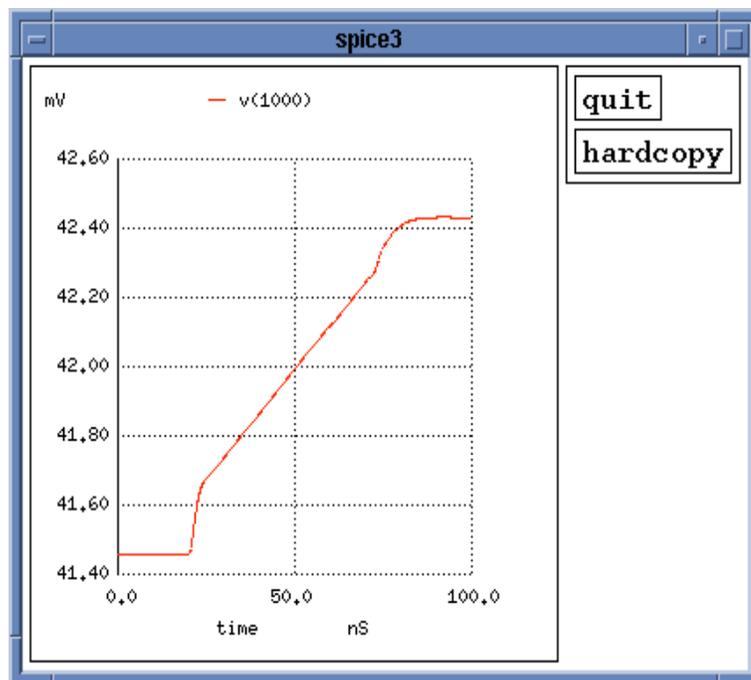


Figure 26: Power consumption for a Read operation

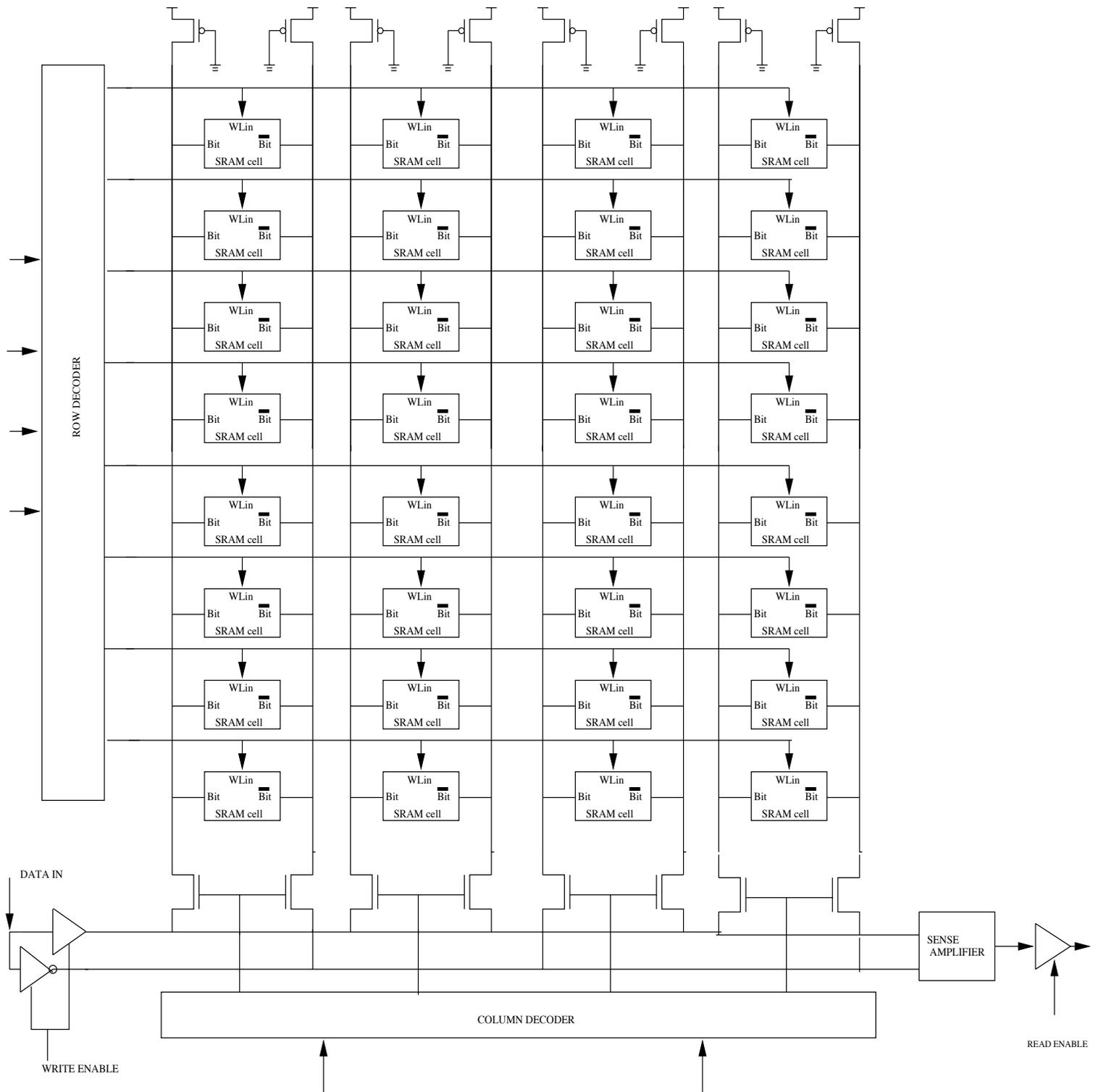


Figure 27: Schematic of our SRAM design

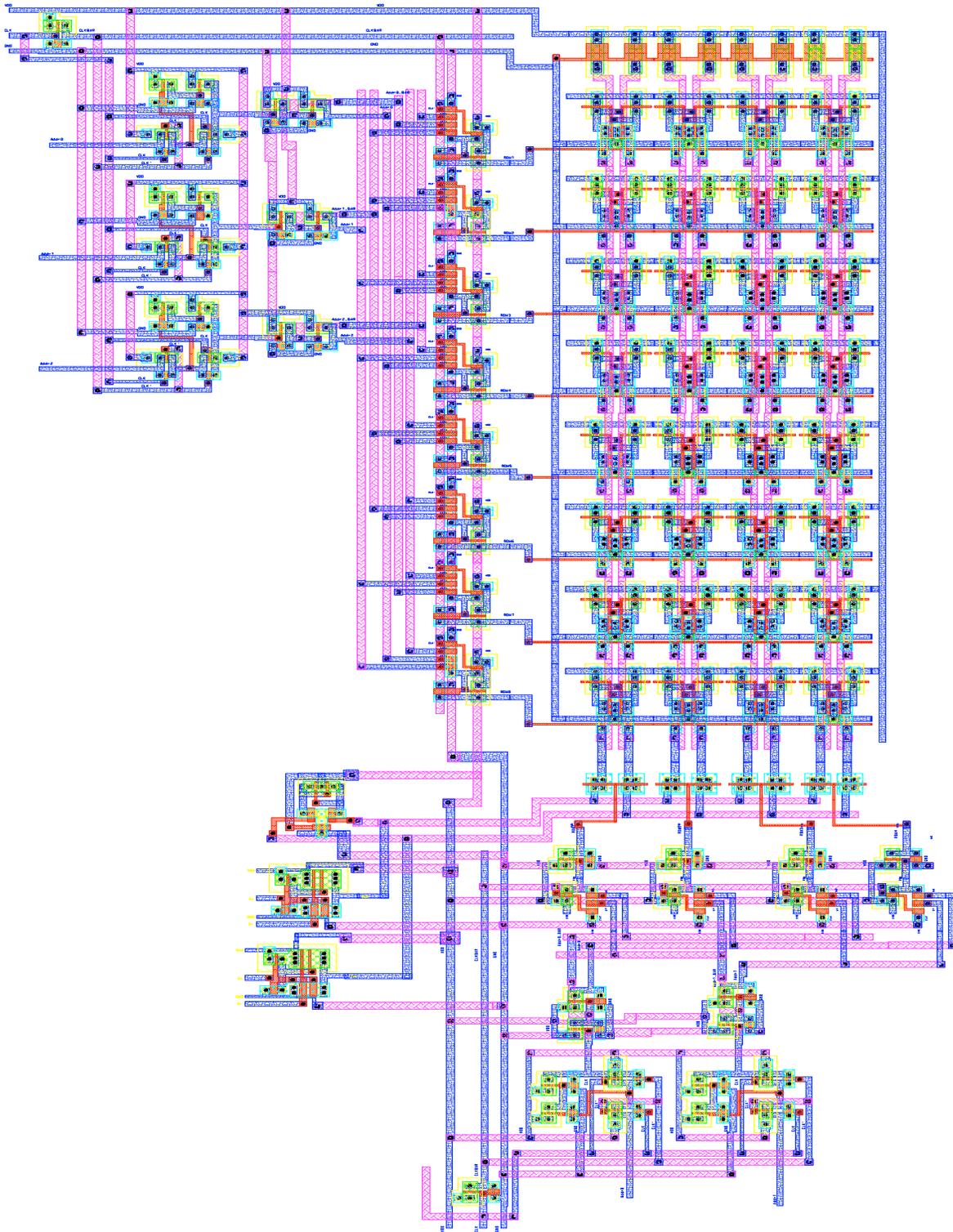


Figure 28: Total layout of our SRAM

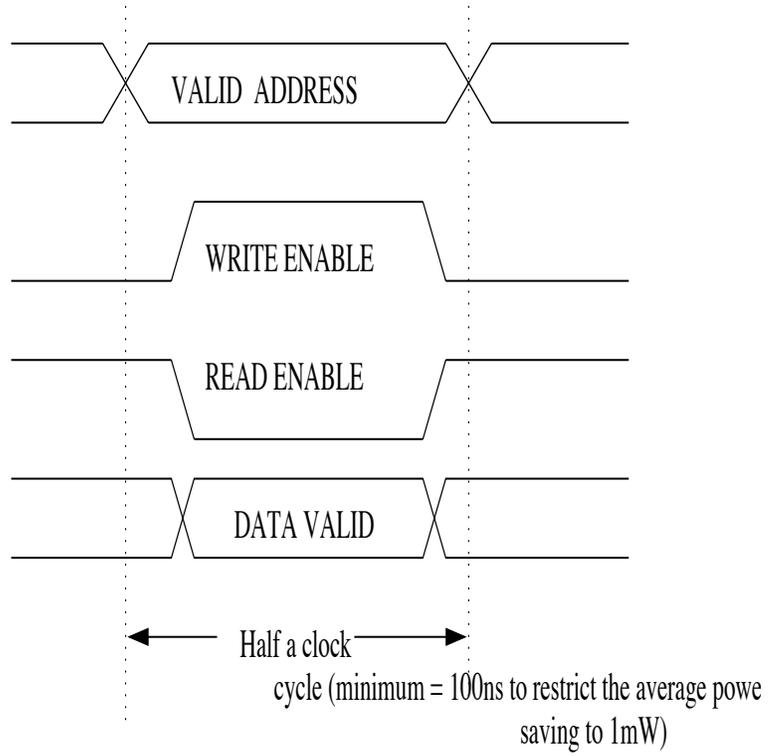


Figure 29: Timing diagram for the Write operation

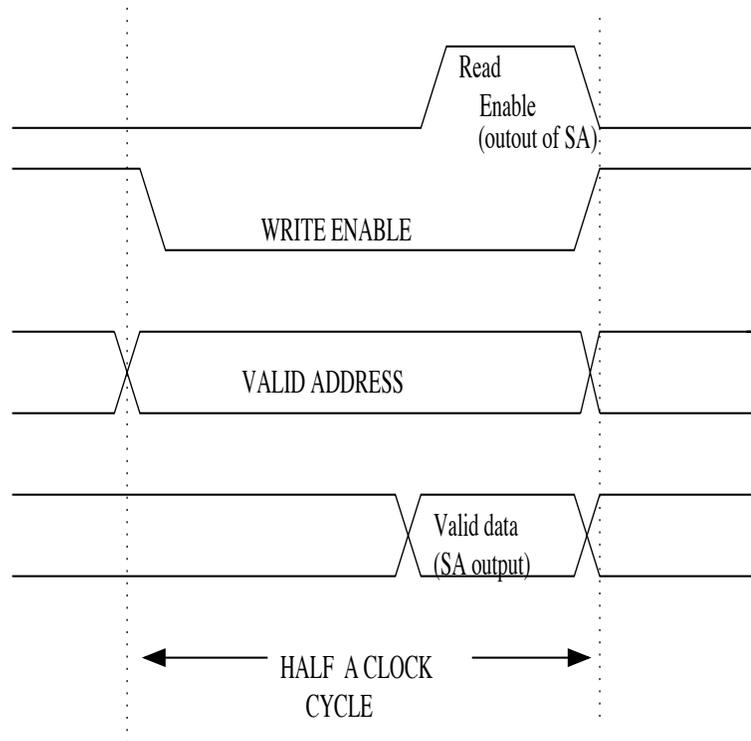


Figure 30: Timing diagram for the Read operation