**National Institute of Technology Karnataka, Surathkal**

EC383

Mini Project in VLSI Design



**SRAM Design and Analysis**

**End-Semester Report**

3 December 2020

***Authors***

Rahul M Hanchate - 181EC135

Bhaskar Vyas - 181EC109

***Supervisors***

Prof. M. S. Bhat

Prof. Nikhil K S

Prof. Sushil Kumar Pandey

# **Introduction**

SRAM or Static Random Access Memory has occupied a large market in modern SoC designs. The use of SRAM in the ASIC domain is increasing at a much larger pace than ever before. In order to understand various aspects of chip design and its manufacturing processes, we need to understand the design and operation of SRAM.

The growing gap between the MicroProcessor Unit (MPU) cycle time and memory access time necessitated the introduction of several levels of memory hierarchy in modern processors. To reduce the entire dependency on the whole memory, the cache system is there and this system should be of relatively higher speed, compared to other memories, to avoid relevant latencies. That’s why the cache area in our computer or in general SoCs is occupied by SRAM and there’s always a need to optimize overall performance.

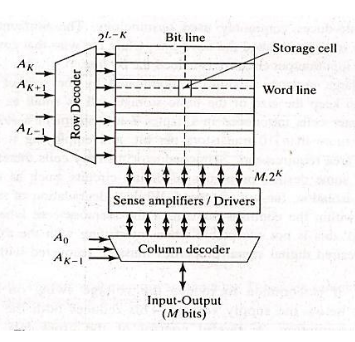
# **Objective**

Our main objective was to analyze and design a basic SRAM architecture using 130nm pharosc technology with the help of MAGIC, IRSIM and other open source tools available. We built a 256 bit memory system (16 x 16 array) of 4 address bits with a 4 bit data line. We implemented various changes to certain building blocks of SRAM such as changing conventional latch based ways of sense amplifiers to modern SAs by incorporating changes in it and to analyse various techniques of read and write operations.

# **SRAM Architecture**

The basic SRAM architecture consists of:

* SRAM cell
* Precharge Circuit
* Sense Amplifier Circuit
* Write Driver Circuit
* Row & Column Decoders
* Timing Blocks
* Buffers with Data Bus (I/O line)



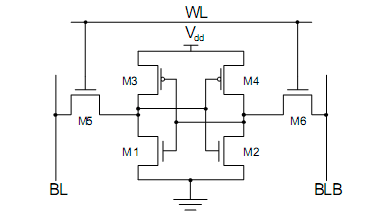
Typically, SRAM block structure can operate in a bit oriented or word oriented way. When each address addresses a single bit, it is called as bit oriented, whereas addressing multiple bits is word oriented addressing.

As we are supposed to build a N x N addressed SRAM structure, we need to have both row and column decoders which help us accessing a particular bit line of the SRAM block structure. These give rise to 2N x 2N lines in which SRAM cells are interconnected with each other. It has various precharges attached to the bit lines to charge it to a specific voltage level. The sense amplifier amplifies the analog differential voltage developed by the bit lines while reading and helps to stick it near digital output signal levels. Write drivers, as the name suggests, writes the bit wherever required and drives the bit lines using discharge from the precharge to ground. Other parts include output buffers for the bit lines, timing blocks (majority of modern SRAMs are self timed) and data buses (I/O blocks). Some of the SRAM parts can be modified to global, semi-global and local levels to enhance the performance and speed of read and write.

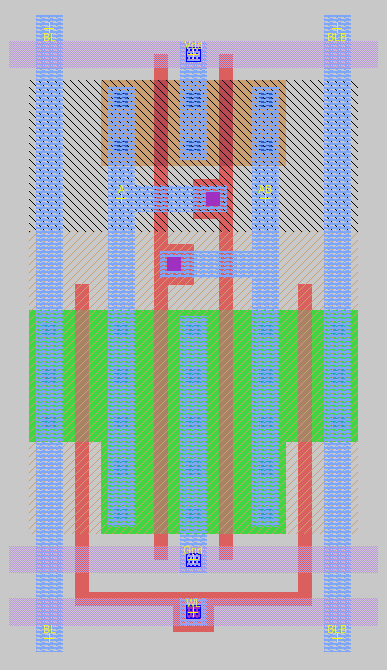
# **SRAM Design**

## **SRAM Cell**

There are various ways in which we can implement a SRAM cell. The main factors are number of transistors, propagation delay, power consumption and scalability. We looked across these factors and chose to implement the standard 6-Transistor SRAM cell. The figure of it is given below:



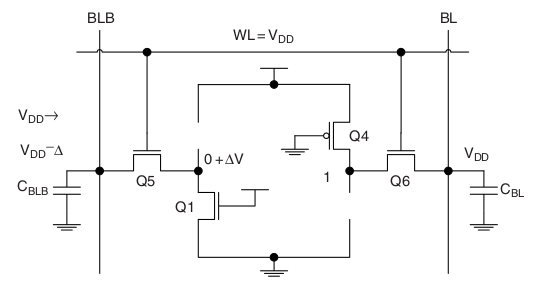
We designed the 6T SRAM cell on MAGIC Layout Editor using pharosc 130nm technology. The below figure represents the layout of 6T SRAM standard cell.



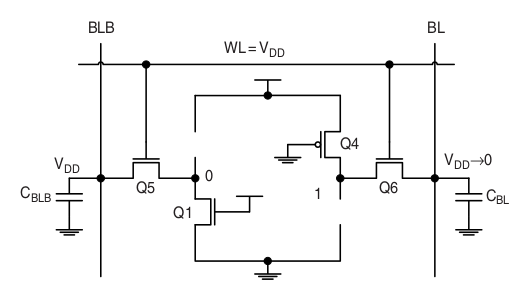
In SRAM, for any operation to be performed, the word line should be high. To perform read operation, initially some memory would be there. Bitlines act as output lines and these bit lines are initially pre-charged i.e there will be a node voltage at both bitlines. Since inner 4 transistors form 2 cross coupled inverters, hence the value stored in those would be complementary.

As word line is raised, the pre-charged voltage of the bit line, whose nearby inverter holds low value at input, will get lowered by a particular voltage margin, which will result in a slight rise at that inverter vallue, and this variation between both bit lines will be picked up by sense amplifier and it will transfer the stored logic further.

However, one consideration should be there that while performing reading operation, the increase in lower voltage of one inverter doesn't switch the voltage of another inverter. This can be done keeping appropriate width to length ratios of transistors involved.



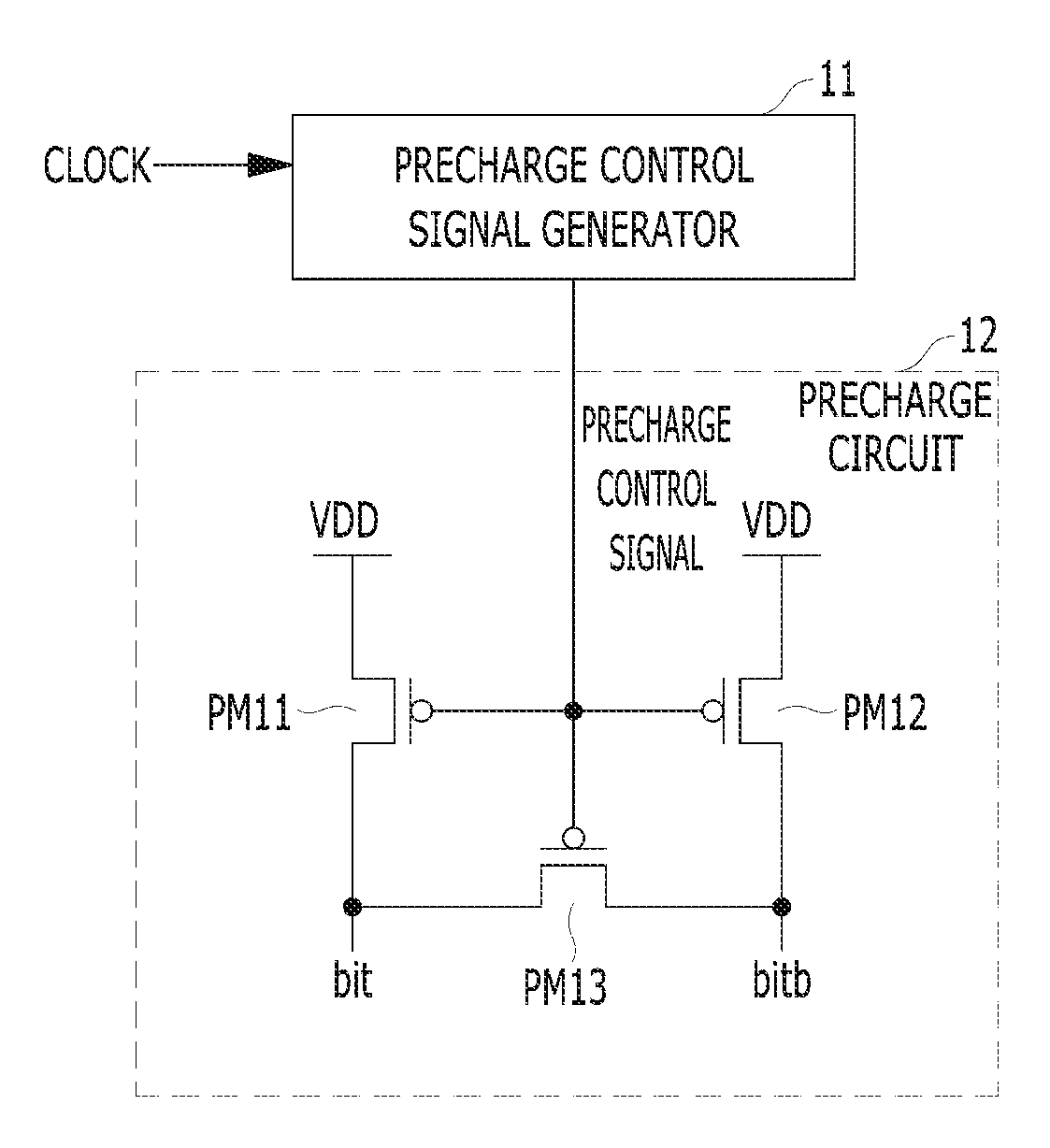
Similarly as read operation, to write value to the cell, we need to raise the word line. But here bitlines will act as input lines. Using a write driver, one of the bit lines will be driven to ground potential, resulting the same for one of cross coupled inverters then the opposite inverter’s value will also switch, resulting in effective overwriting of present data.



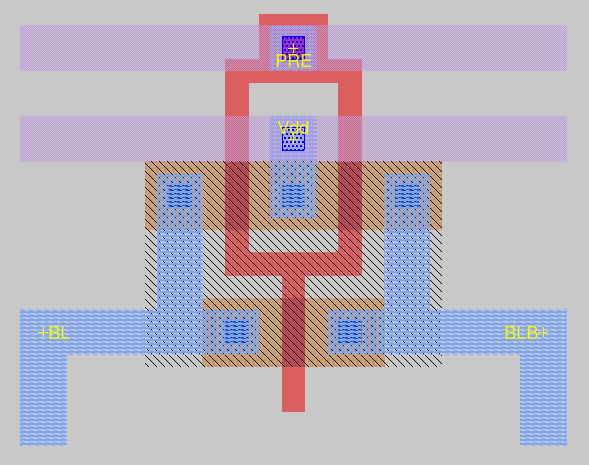
## **Precharge Circuit**

Precharge circuit is one of the most important parts of SRAM. Its job is to charge up the bit and complementary bit lines to VDD. It enables the bit lines to be charged during all times except during read and write operations. As the sensing operation begins after enabling the SAE pin, setting the operating point by precharging and equalisation of both bit lines to precharged levels.

Below figure shows the schematic of a standard precharge circuit.



The layout for the corresponding circuit has been designed in MAGIC:



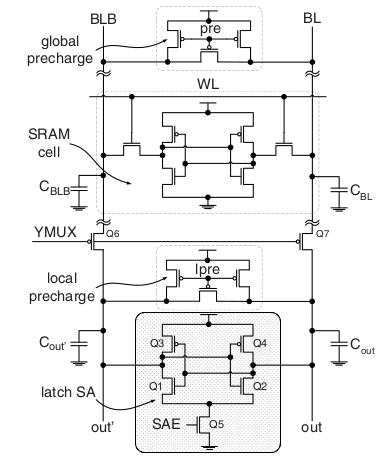
## **Sense Amplifier Circuit (Conventional Latch-Based)**

A sense amplifier is an active circuit that reduces the time of signal propagation from an accessed memory cell to the logic circuit located at the periphery of the memory cell array, and converts the arbitrary logic levels occurring on a bitline to the digital logic levels of the peripheral Boolean circuits.

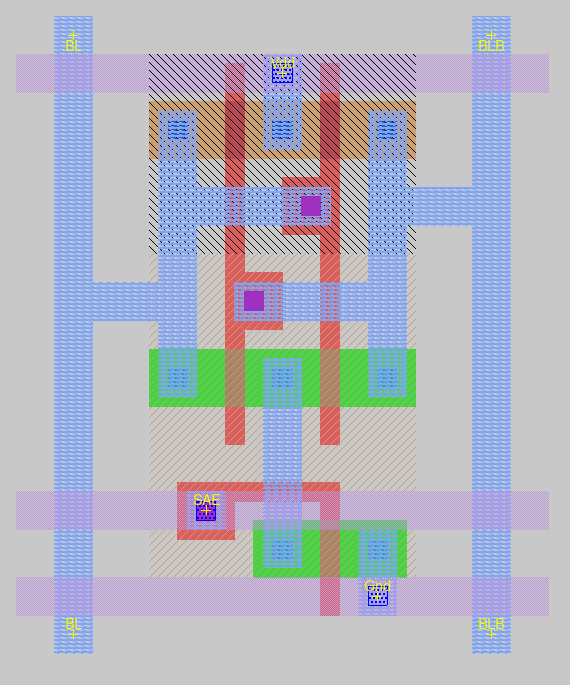
The SA circuit has to operate within the conditions that are set by operation margins. These margins constraints the minimum and maximum input voltage levels as well as the minimum gain for sense amplifiers.

There are some layout area restrictions for SAs in memory designs. Here, in case of SRAM, SA layout should fit in the decoder pitch when a multiplicity of bitlines are connected to a single sense amplifier.

Below is a typical circuit in which a latch-type sense amplifier.



The following figure represents the layout of the same circuit.



## **Voltage Mode Sense Amplifiers (VMSAs) and Current Mode Sense Amplifiers (CMSAs)**

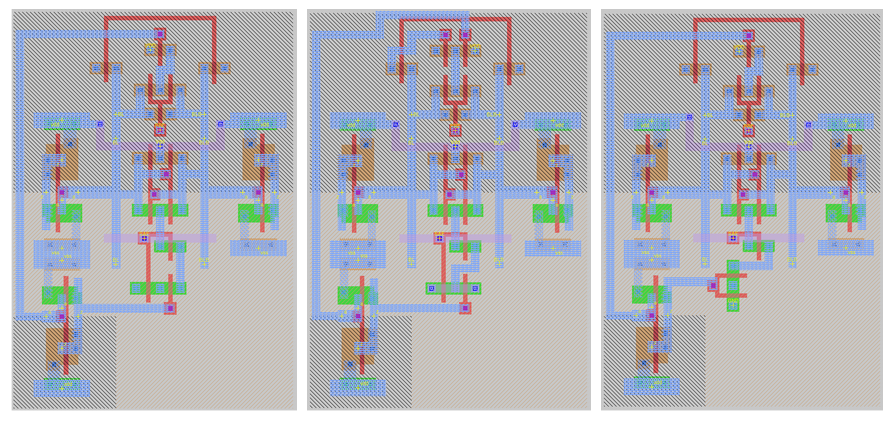
The performance of VMSA is limited by the time required to develop a differential voltage on high capacitance SRAM bit lines.

Current sensing technique is faster than voltage mode technique because the bit line swing decreases in current mode sense amplifiers during read operation as compared to voltage mode sensing technique.

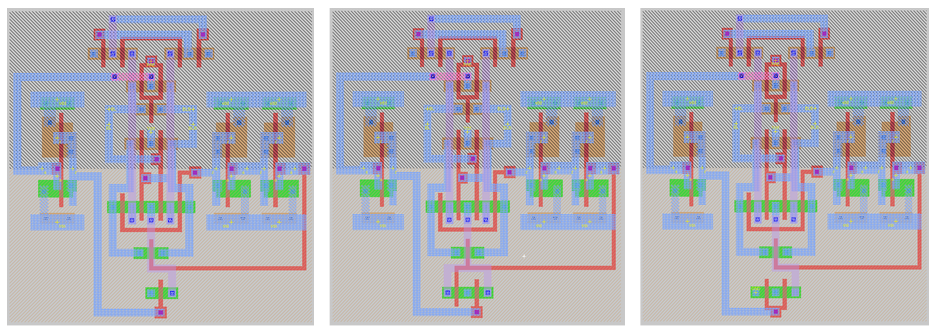
The following techniques were analysed for

1. MTCMOS Technique
   1. The operational block connected to the virtual GND line are allotted lower threshold voltage value.
   2. This line pursued the main GND rail through a high threshold voltage transistor called sleep transistor.
   3. In active-mode, the sleep signal is ‘ON’.
   4. But in standby-mode, sleep signal revolve off the high threshold transistor.
2. Sleepy Stack Technique
   1. The sleepy-stack approach combines the sleep and stack approaches.
   2. This technique divides the single transistor into two half-size transistors.
   3. Between the divide transistors, one of the sleep transistors will be added in parallel.
   4. It also adds sleep transistor to disconnect the power supply and ground from the network so that there is no power consumption in off mode.
3. Footer Stack Technique
   1. In this structure, one HVT is stacked in two transistors with a width twice that of the original one.
   2. Leakage power is less leaky with more than one-off transistor as compared to a one-off transistor in the path.
   3. This technique is the great extent to reduce leakage power as compared to other techniques.

The following layouts represent MTCMOS, Sleepy Stack and Footer VMSAs respectively.



The following layouts represent MTCMOS, Sleepy Stack and Footer CMSAs respectively.

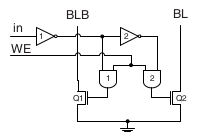
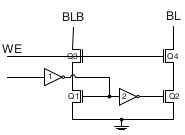
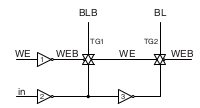


## 

## **Write Driver Circuit**

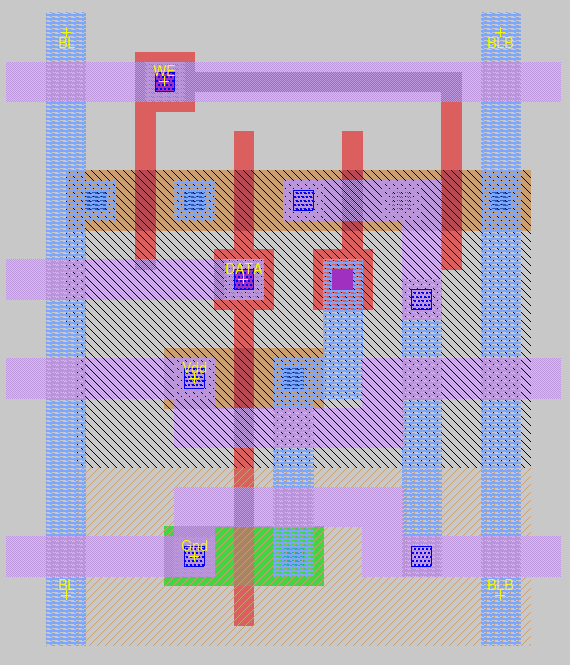
The main job of a SRAM write driver circuit is to discharge one of the bit lines quickly from precharged levels to below the write margin required in the SRAM cell. The circuit is enabled by the WE (Write Enable) pin which drives the bit line using full swing discharge from precharge to ground.

There are some ways in which we can implement a write driver circuit. They are listed below:



We have used the second circuit which contains two NMOS transistors placed upon another with write enable pin governing the above two transistors and data input below.

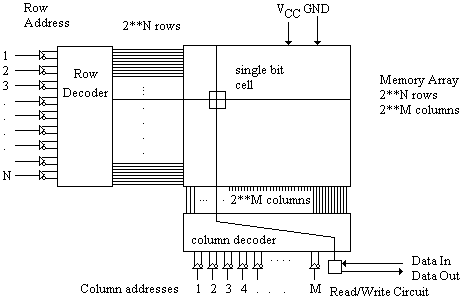
The following figure shows the implementation of a simple SRAM Write Driver (second circuit shown above)



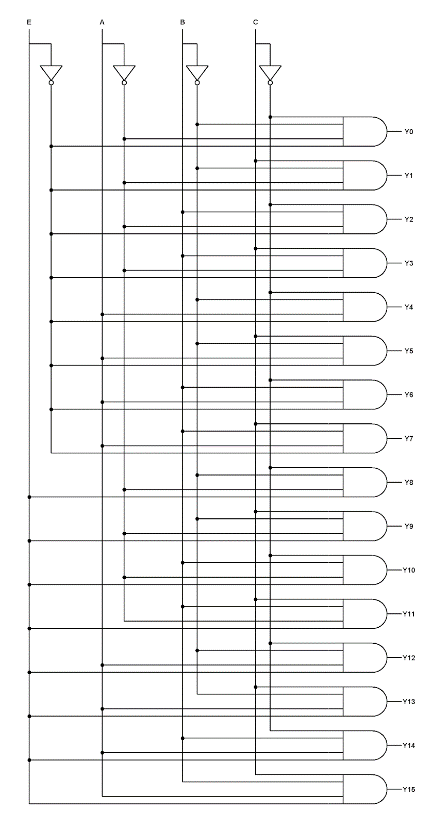
## **Row and Column Decoders**

To help look up a specific address of memory, or in other sense, access a particular or series of SRAM cells, we need to use address decoders. The memory size is defined by the total number of bit lines required to access a particular or words of bits.

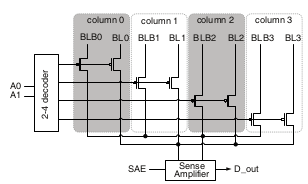
The following figure depicts the way decoders are connected to the SRAM array.



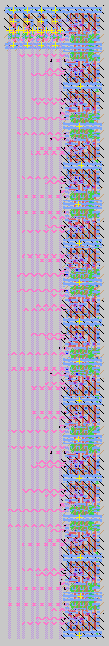
The following figure depicts the schematic circuit for a 4:16 row decoder.



The following figure depicts the connection of a column decoder with all bitlines



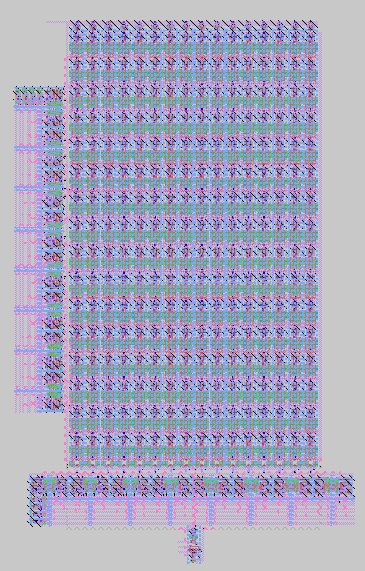
The following figure depicts the layout of a 4:16 decoder.



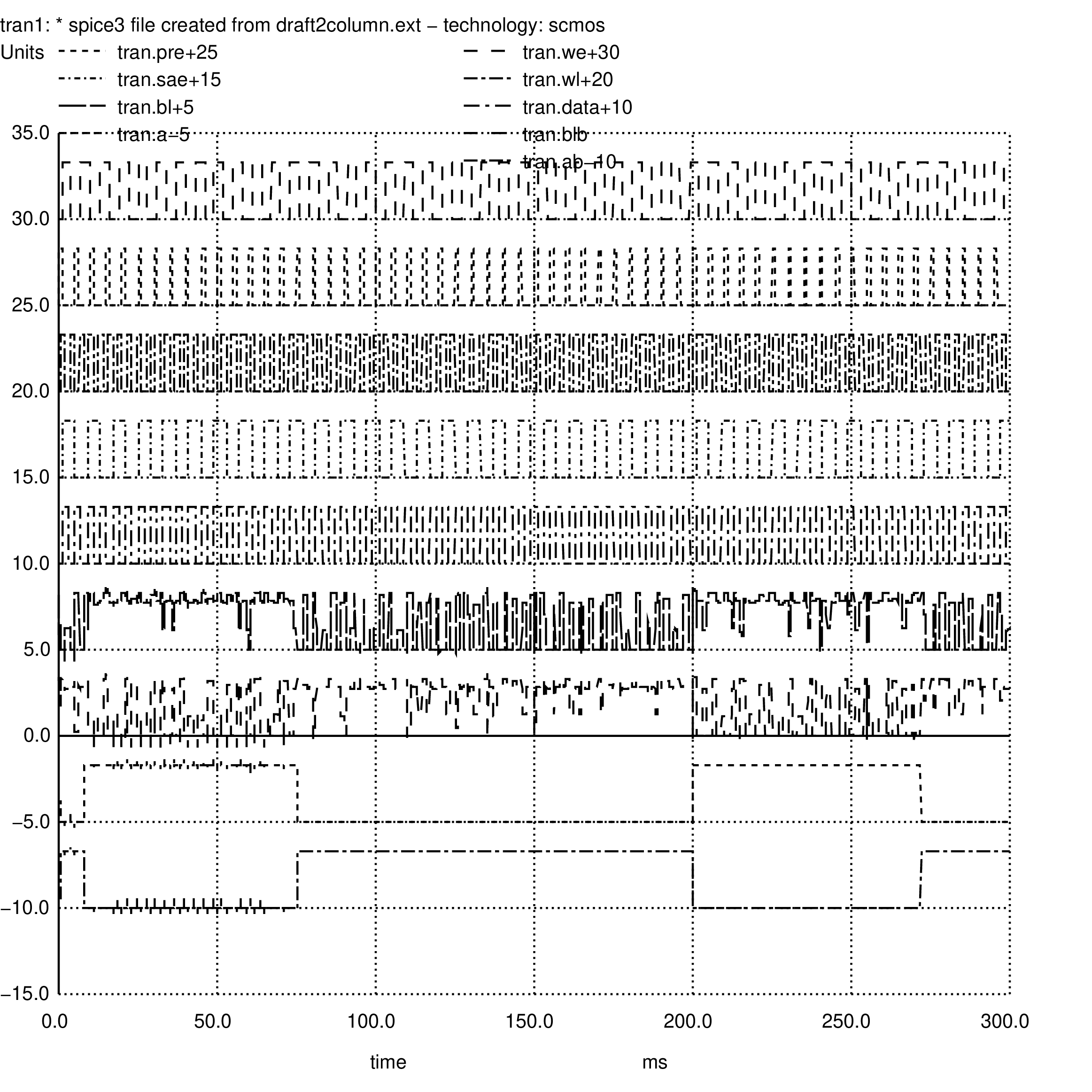
# **Methodology and Results**

The following figure shows the final 256 bit SRAM array layout of 4 address bits with a 4 bit data line, created in MAGIC Layout editor tool.

The total average area of the layout is 22,25,223 sq microns (1183 × 1881). The total delay is estimated to be 15.5 ns. The total sum of nodal capacitance is found out to be 32.48 pF.



Read and Write operation for a particular SRAM cell using MTCMOS VMSA in Ngspice has been depicted below:



Some more observations:

* SRAMs with CMSA consume less power (1.54 mW) compared to SRAMs with VMSA (1.87 mW).
* Footer Stack is found out to be the most efficient technique (1.54 mW) while implementing a SA when compared to MTCMOS (1.56 mW) and Sleep Stack (1.65 mW).
* As the time step increases, the power consumed increases exponentially.
* One fourth of the space in the layout is unutilised.

# **Conclusion Drawn and Future Improvements**

* The time steps have to be shorter in order to lessen dynamic power consumption. It should also be ensured that the time step chosen shouldn’t be too short for the output to appear and stabilize.
* In order to further improve performance, we can add global and local precharges. Decoders can be made multi-stage as this approach is proven to be more power efficient.
* High sensitivity to transistor mismatches in a current-mirror SA causes increased offsets. So to compensate for possible offsets and maintain reliable sensing, the minimum differential voltage must be increased, slowing down the sensing.
* Timing Control Schemes can be introduced.
* Develop a full-fledged processor for a specific use (Deep Learning, Electronic Devices, etc).

# **References**

* Andrei Pavlov and Manoj Sachdev. 2008. *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies: Process-Aware SRAM Design and Test(1st. ed.).* Springer Publishing Company, Incorporated.
* Y. Tao and W. Hu, "Design of Sense Amplifier in the High Speed SRAM," *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Xi'an*, 2015, pp. 384-387, doi: 10.1109/CyberC.2015.32.
* A. Abdollahi, F. Fallah and M. Pedram, "A Robust Power Gating Structure and Power Mode Transition Strategy for MTCMOS Design," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 1, pp. 80-89, Jan. 2007, doi: 10.1109/TVLSI.2007.891093.
* K. Sridhara, G. S. Biradar and R. Yanamshetti, "Subthreshold leakage power reduction in VLSI circuits: A survey," *2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur*, 2016, pp. 1120-1124, doi: 10.1109/ICCSP.2016.7754326.
* Diagrams: Andrei Pavlov, Manoj Sachdev - CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies\_ Process-Aware SRAM Design and Test (2008, Springer)