32x16 SRAM in 90nm CMOS Technology

ENGR848: Digital VLSI Design

(Spring 2021)

Team members

Amir Modan(918234621)

Jinal Jogesh Apte(920820066)

Design Goal and Constraints

The purpose of this project is to design a 32X16 SRAM having minimum chip area and access time along with minimum power consumption.

Design constraints:

- Address bus size of 5 bits
- Input data bus of size 16 bits
- Output data bus of size 16 bits
- The following electrical constraints all at nominal supply voltage of 1.2 Volts and worst-case temperature of 110ºC:
  - Minimum read SNM of 150 mVolts
  - Minimum bit-line differential of 100 mVolts in the read mode when the sense amplifier is enabled.
  - Maximum standby power of 200 µWatts

INTRODUCTION

RAM (Random Access Memory) is the internal memory of the CPU for storing data, program, and program result. It is a read/write memory which stores data until the machine is working. RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. RAM is small, both in terms of its physical size and in the amount of data it can hold. There are various different types of memories as shown in the figure but we will mainly focus on the Volatile Memories.

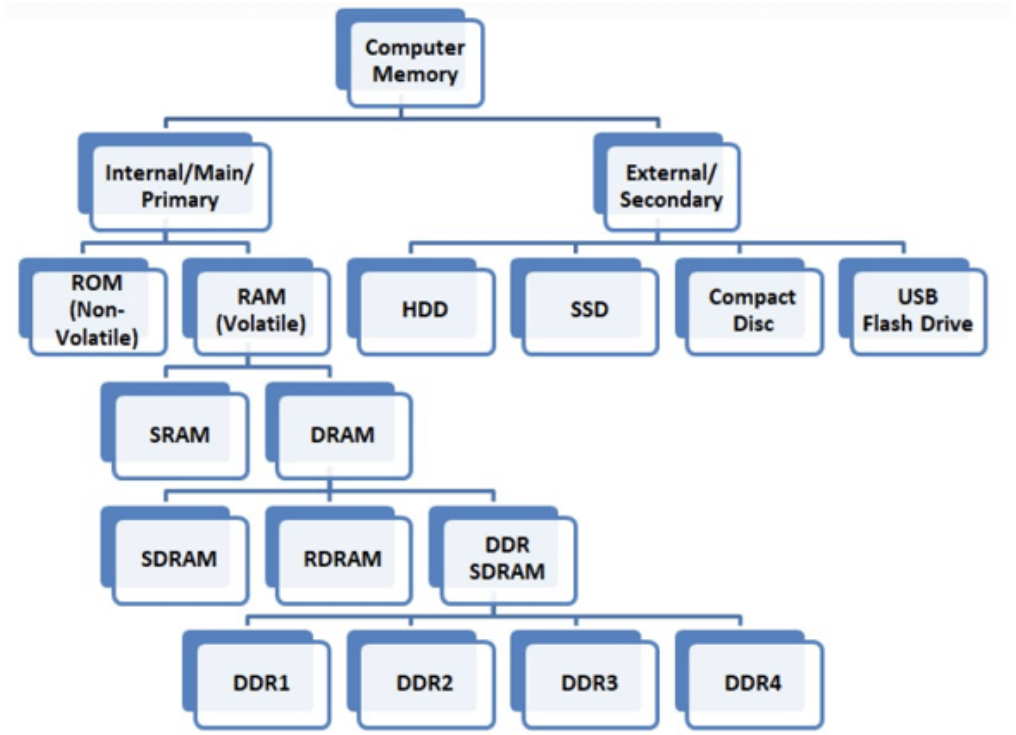


Figure 1 - Memory Types

Volatile Memories are mainly divided into two SRAM and DRAM .

DRAM

DRAM, unlike SRAM, must be continually refreshed in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory as it is cheap and small. All DRAMs are made up of memory cells, which are composed of one capacitor and one transistor. Characteristics of Dynamic RAM are as follows :

- Short data lifetime
- Needs to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM
- Smaller in size
- Less expensive
- Less power consumption.

SRAM

The word static indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not be refreshed on a regular basis.

There is extra space in the matrix, hence SRAM uses more chips than DRAM for the same amount of storage space, making the manufacturing costs higher. SRAM is thus used as cache memory and has very fast access.

Characteristic of Static RAM are as follows :

- Long life
- No need to refresh
- Faster
- Used as cache memory
- Large size
- Expensive
- High power consumption

In this Project we will mainly focus on the SRAM .

Top level block diagram of the design

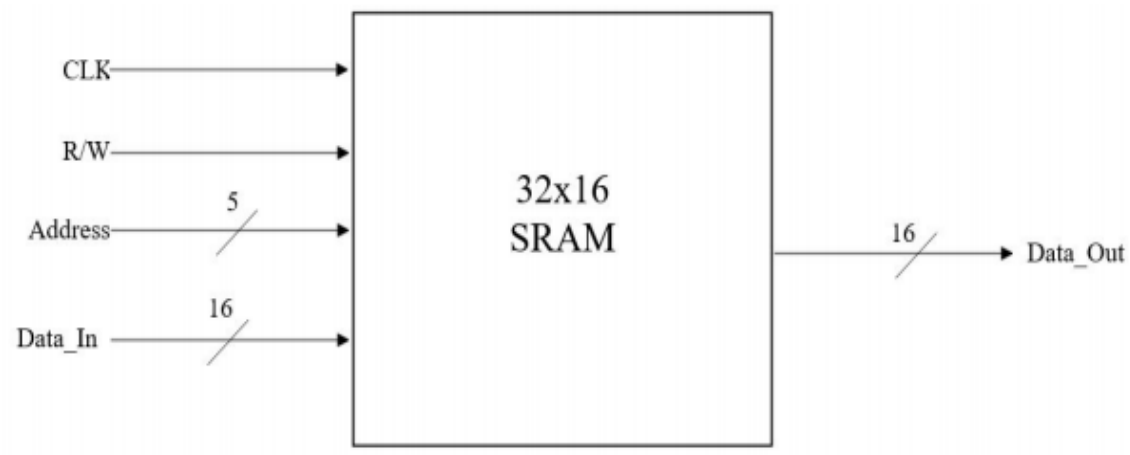


Figure 2 - SRAM Black Box Design

The top level of the design consists of the following parameters

- Clock Signal
- Read and Write Signal
- Address Bus (5 bits)
- Data Input (16 bits)
- Data Output (16 bits)

Design partitioning (Block Diagram)

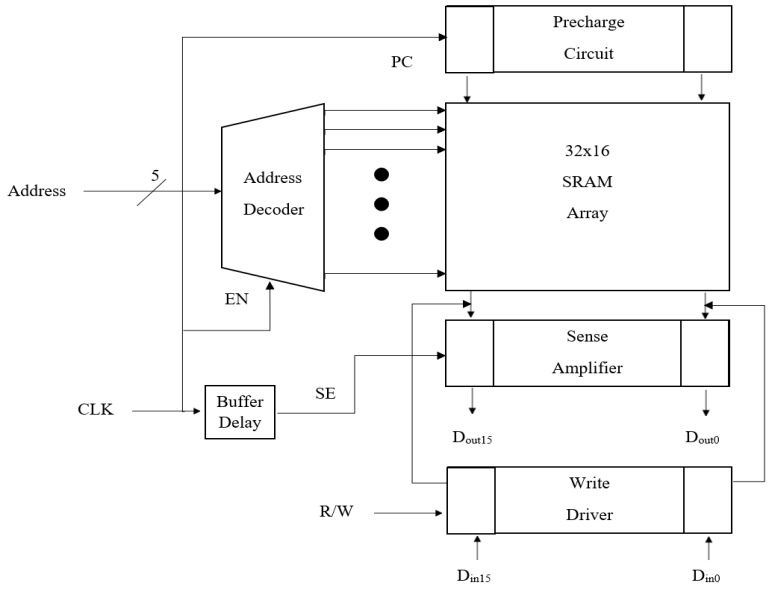


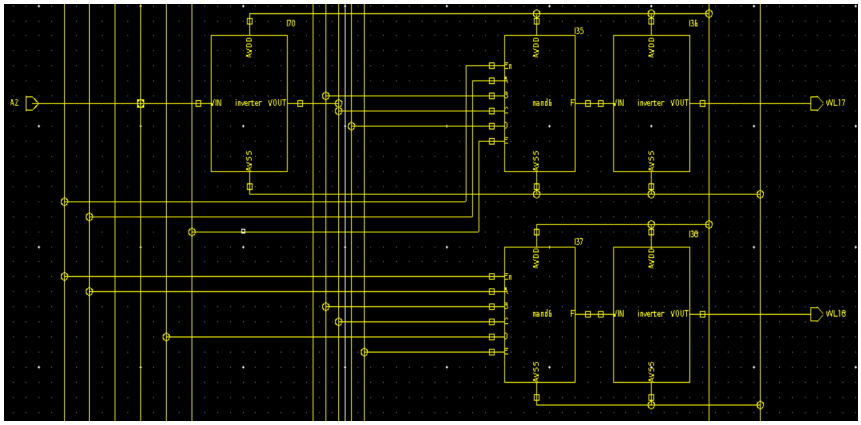
Figure 3 - SRAM Architecture

The figure shown above is the Design partitioning block diagram . It consists of the following .

- Address Decoder
- SRAM Array
- Sense Amplifier
- Write Driver
- Precharge Circuit

We will look at each of the following along with its design schematic layout and verification in further points.

Address Decoder



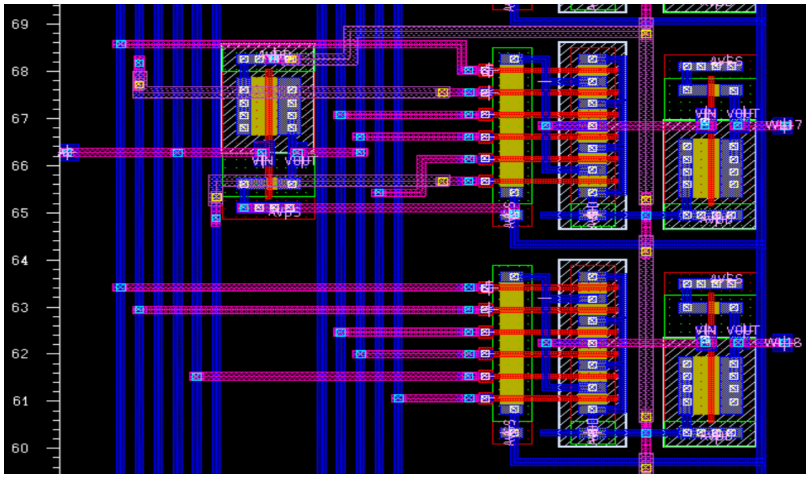


Figure 4 - Decoder Schematic (Top) and Layout (Bottom); Area = 12.63µm x 146.97µm

A 5-Input Decoder will be responsible for generating the addresses of the memory which we want to read/write to. The Decoder is implemented using a NAND implementation in which an address is selected only if the corresponding address bits are present at the input of the Decoder. While this has higher fan-in, the use of parallelism for the PMOS may offer better performance than a NOR implementation which will have 6 PMOS in series. An additional 6th input will be applied to the NAND gate to be used as an enable signal. The reason for this is so that no address will be selected when the SRAM is performing pre-charge on the bit-lines. Additionally, each NAND gate will need to be followed by an inverter so that the selected address is active high. Verification of the Decoder is performed post-layout in Figure 5, where Address 31 is first selected, then others are selected after.

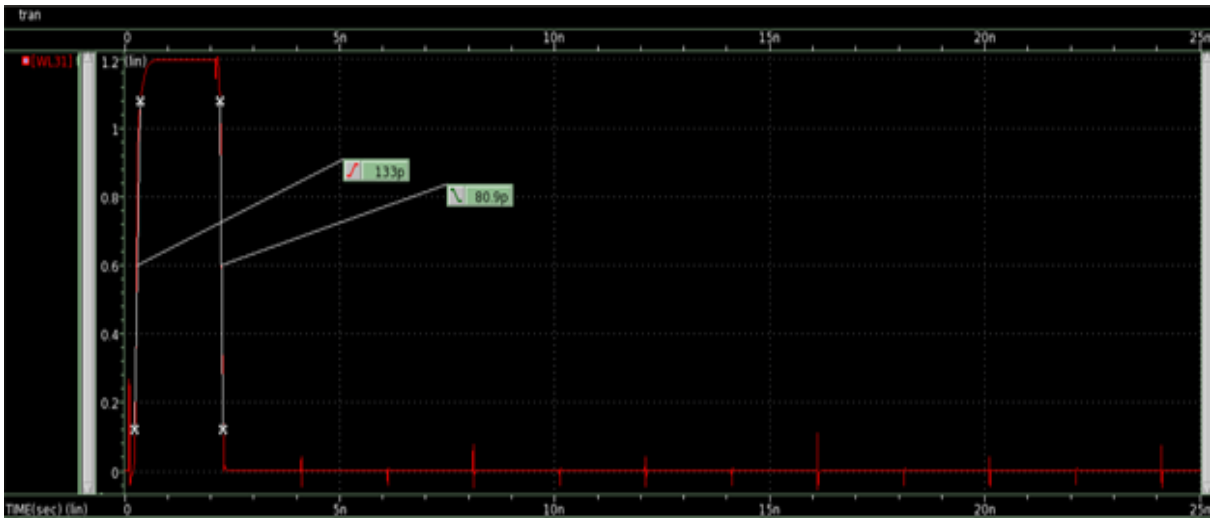


Figure 5 - Decoder Verification

SRAM Cell

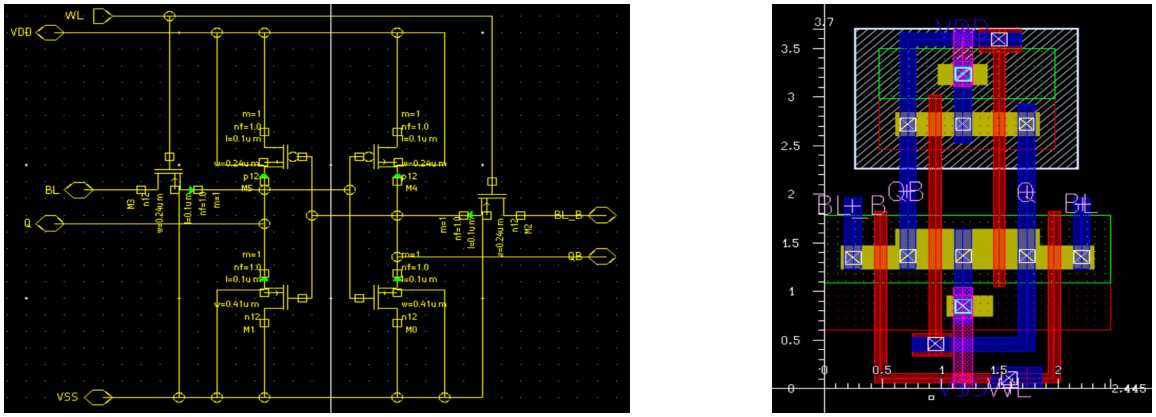


Figure 6 - SRAM Cell Schematic (Left) and Layout (Right); Area = 2.445µm x 3.7µm

The SRAM cell will be responsible for storing the data we write to the SRAM. It will be capable of storing one bit statically, that is, using an inverter feedback mechanism to refresh itself rather than dynamic storage which requires periodic refresh. This, however, requires 6 transistors, making it less dense than its dynamic counterparts.

Reliability of an SRAM cell can be quantitated by its Static Noise Margin (SNM). SNM is an indication of how much noise an SRAM cell can withstand before the data being stored is corrupted. For this project, we are required to meet a minimum of 150mV SNM which is being met according to our analysis in Figure 7.

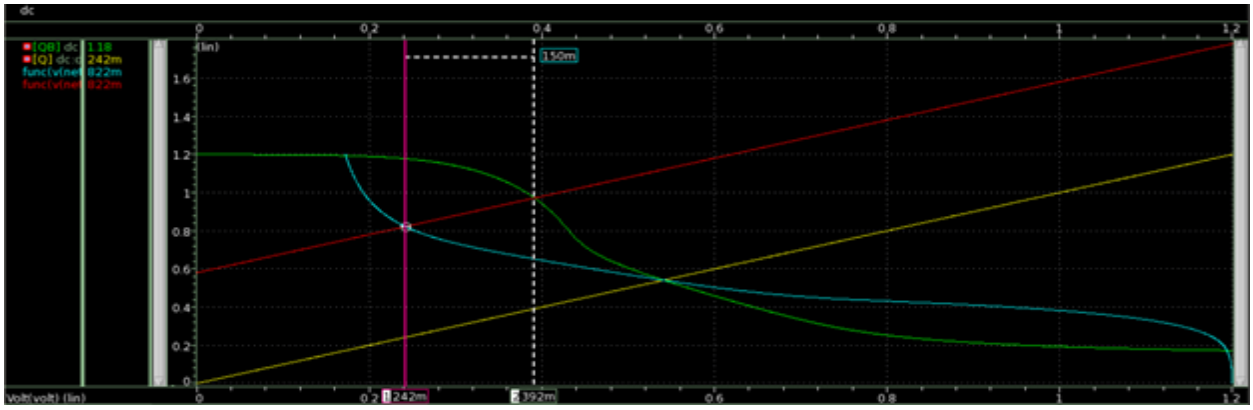


Figure 7 - SRAM Cell SNM

Another metric of the SRAM cell is how quickly it is able to pull both bit-lines so that the differential between the two bit-lines is high enough for a Read Operation by the Sense Amplifier, known as its bit-line delay. We are required by the design specifications to achieve a 100mV differential between the two bit-lines before we can perform amplification. The SRAM cell requires 191ps to generate this differential as shown in Figure 8.

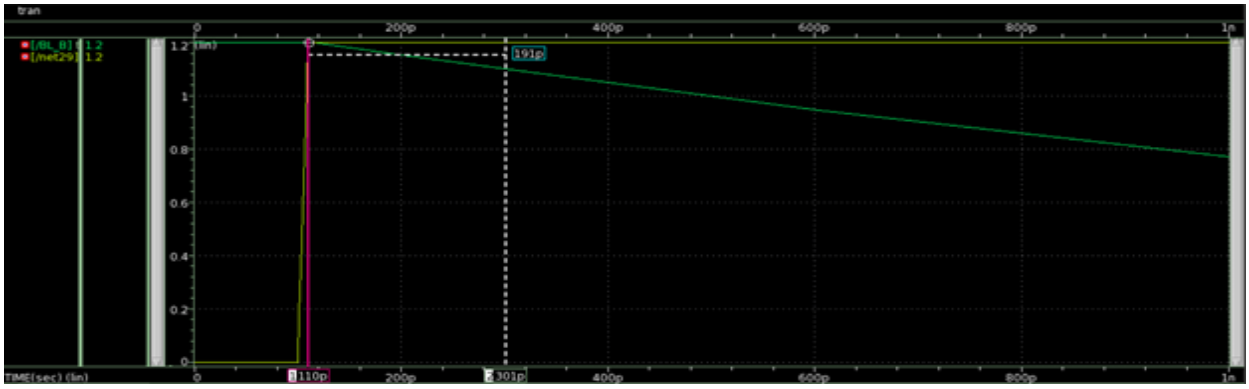


Figure 8 - SRAM Cell Bit-Line Delay

SRAM ARRAY

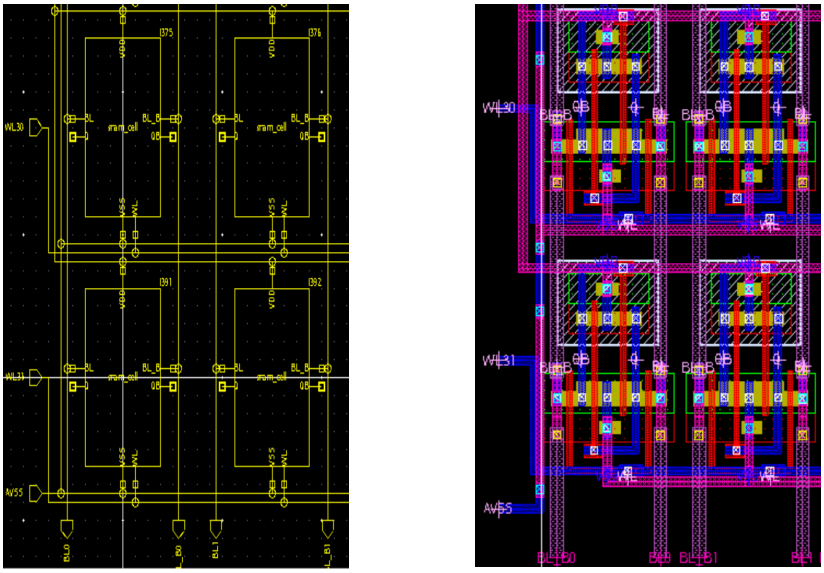


Figure 9 - SRAM Array Schematic (Left) and Layout (Right); Area = 43.575µm x 139.485µm

The previously discussed SRAM Cell will be duplicated into an array that is 32x16 cells long. This means that there will be 32 addresses that one can read from / write to and each address contains a word that is 16 bits long. This accounts to a total of 512 cells in our SRAM design, capable of storing 512 bits (64 Bytes).

SENSE AMPLIFIER

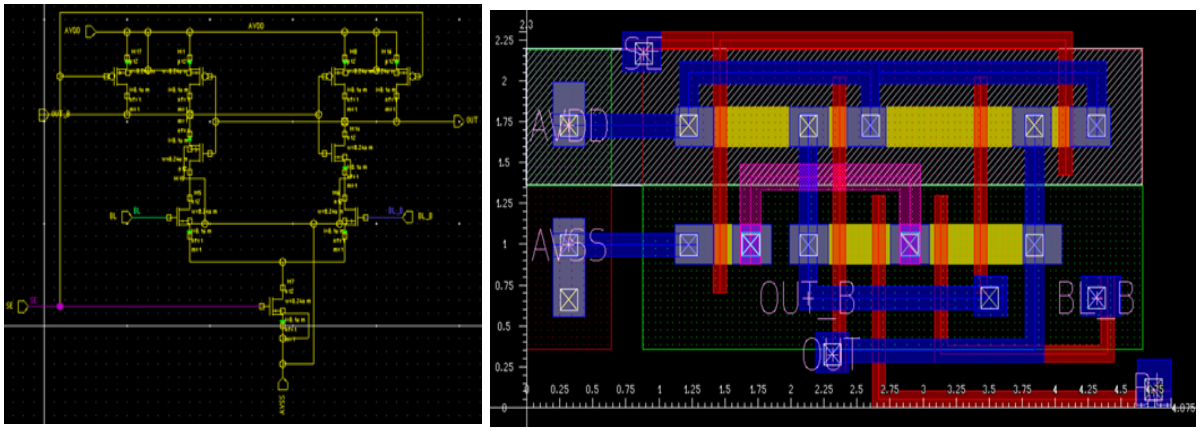


Figure 10 - Sense Amplifier Schematic (Left) and Layout (Right); Area = 4.075µm x 2.3µm

The Sense Amplifier will amplify the differential between Bit-Lines when SE is high. This will decrease the time needed for a read operation which would usually take rather long if the SRAM Cell pulled the Bit-Lines on its own. For our design, we will activate the Sense Amplifier only after the SRAM Cell has developed a 100mV Differential between the two Bit-Lines. On the other hand, the Sense Amplifier will output VDD when it is disabled.

The Sense Amplifier's performance can be characterized by its delay which is how long it takes for the amplified output to surpass 50% transition after the Sense Amplifier is enabled. According to Figure 11, this time is 73.5ps with Parasitics applied which should be low enough for the frequency we are operating at.

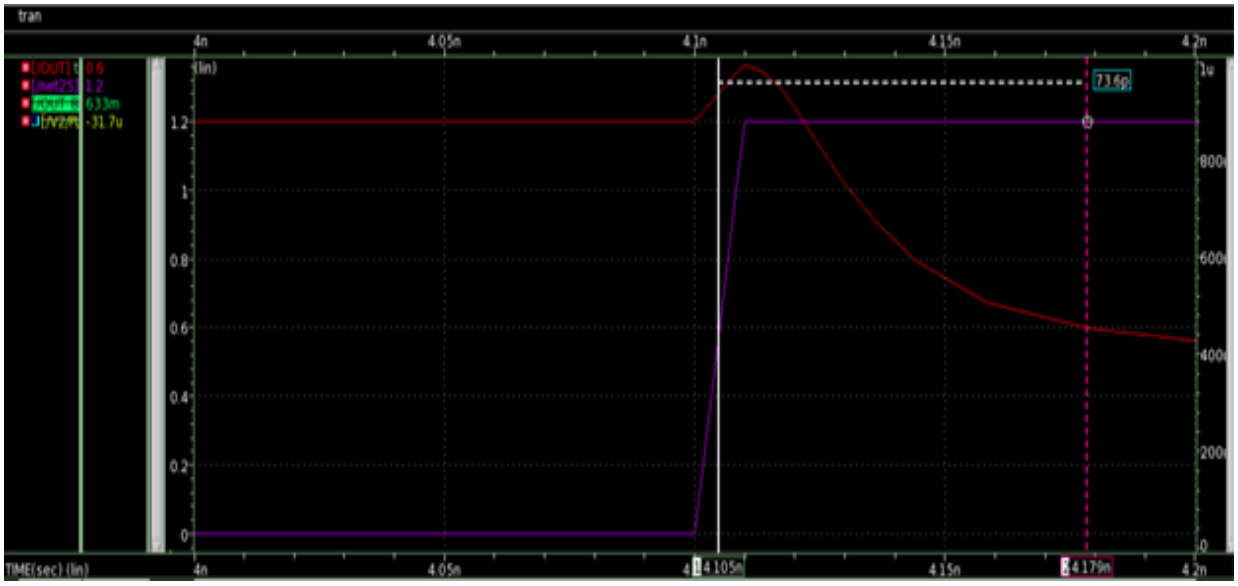


Figure 11 - Sense Amplifier Delay

PRECHARGE CIRCUIT

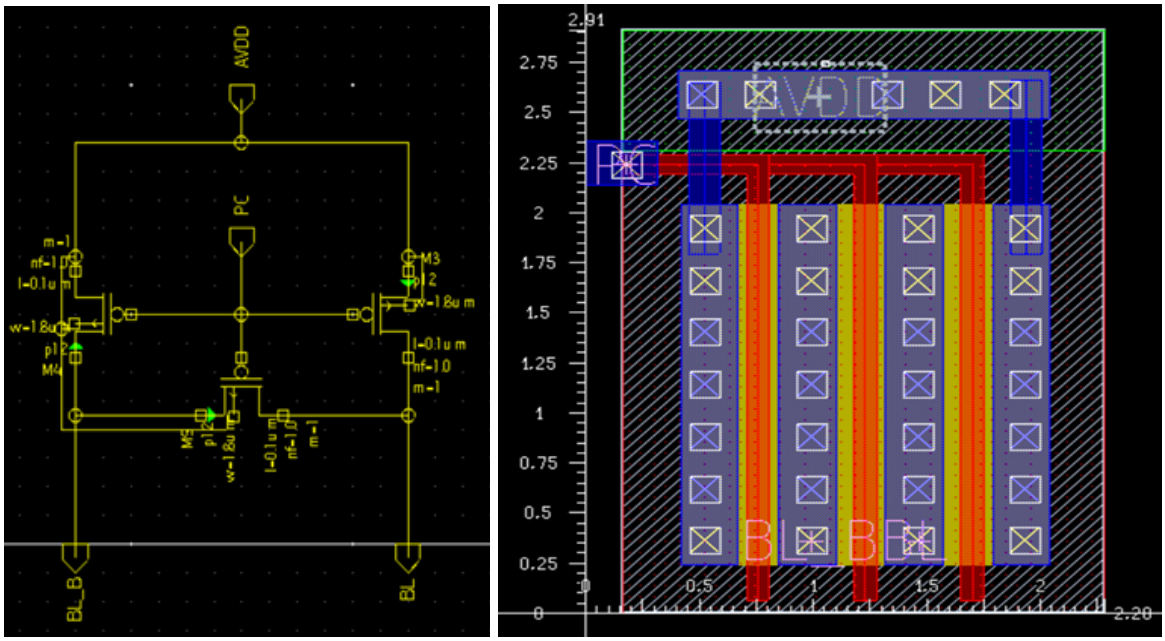


Figure 12 - Pre-Charge Schematic (Left) and Layout (Right); Area = 2.20µm x 2.91µm

The Pre-Charge circuit will be responsible for pulling both bit-lines to VDD when the clock of the entire SRAM is low. This signal will be the same signal that will enable the Decoder. Note that the Pre-Charge Circuit will be active only when the PC signal is low. Therefore, the Pre-Charge will never be active at the same time as the decoder, preventing any unwanted behavior due to both being active.

However, we would like the delay of the Pre-Charge Circuit to be less than that of the Decoder so that precharging does not add to the overall latency of the circuit. Figure 13 demonstrates that the delay of both components is equivalent, preventing any additional latency in the system due to the Pre-Charge Circuit. Note that this came at the Area expense of having to supersize the PMOS Transistors of the Pre-Charge Circuit as seen in Figure 12.

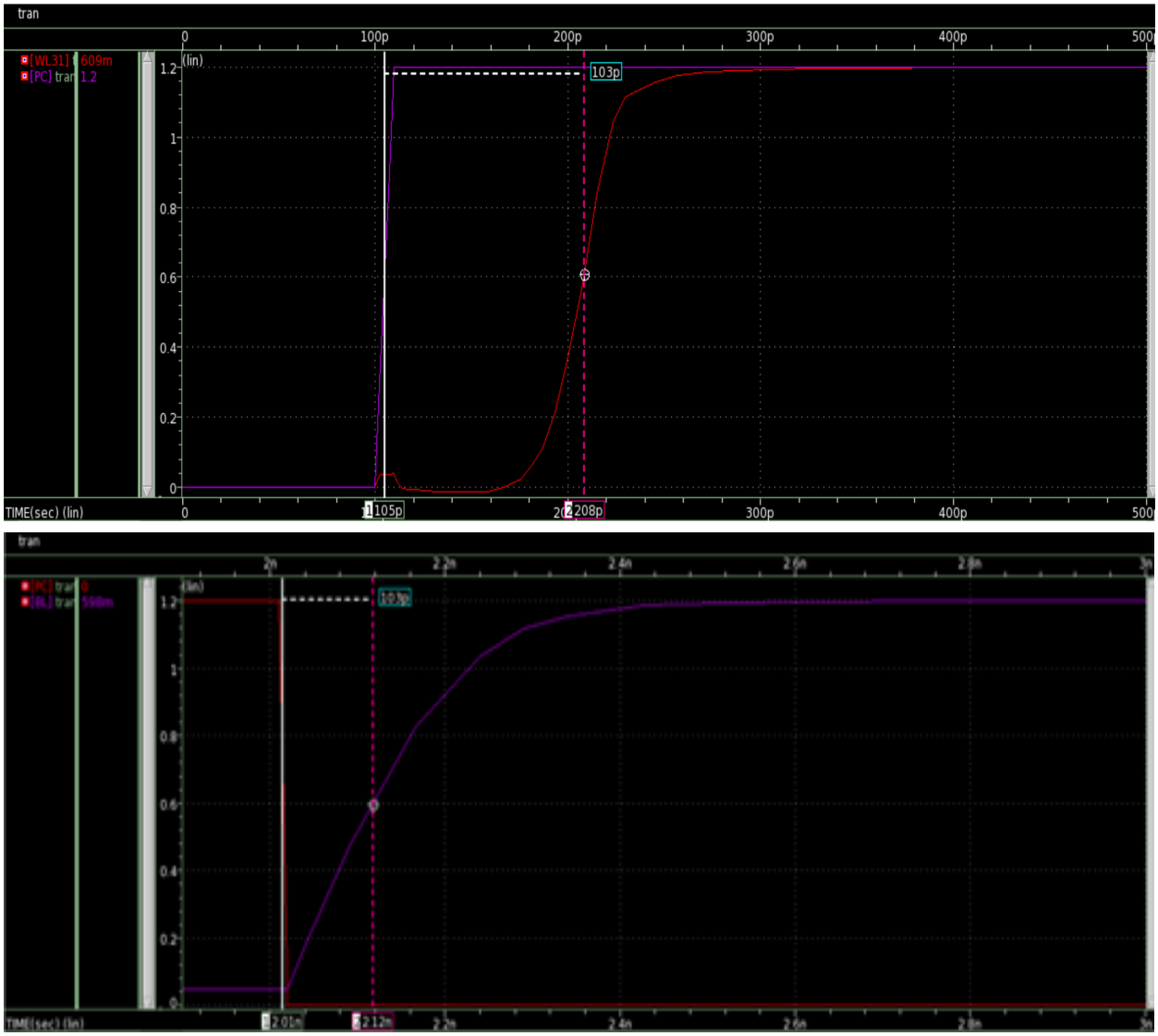


Figure 13 - Decoder Delay (Top) and Pre-Charge Delay (Bottom)

WRITE DRIVER

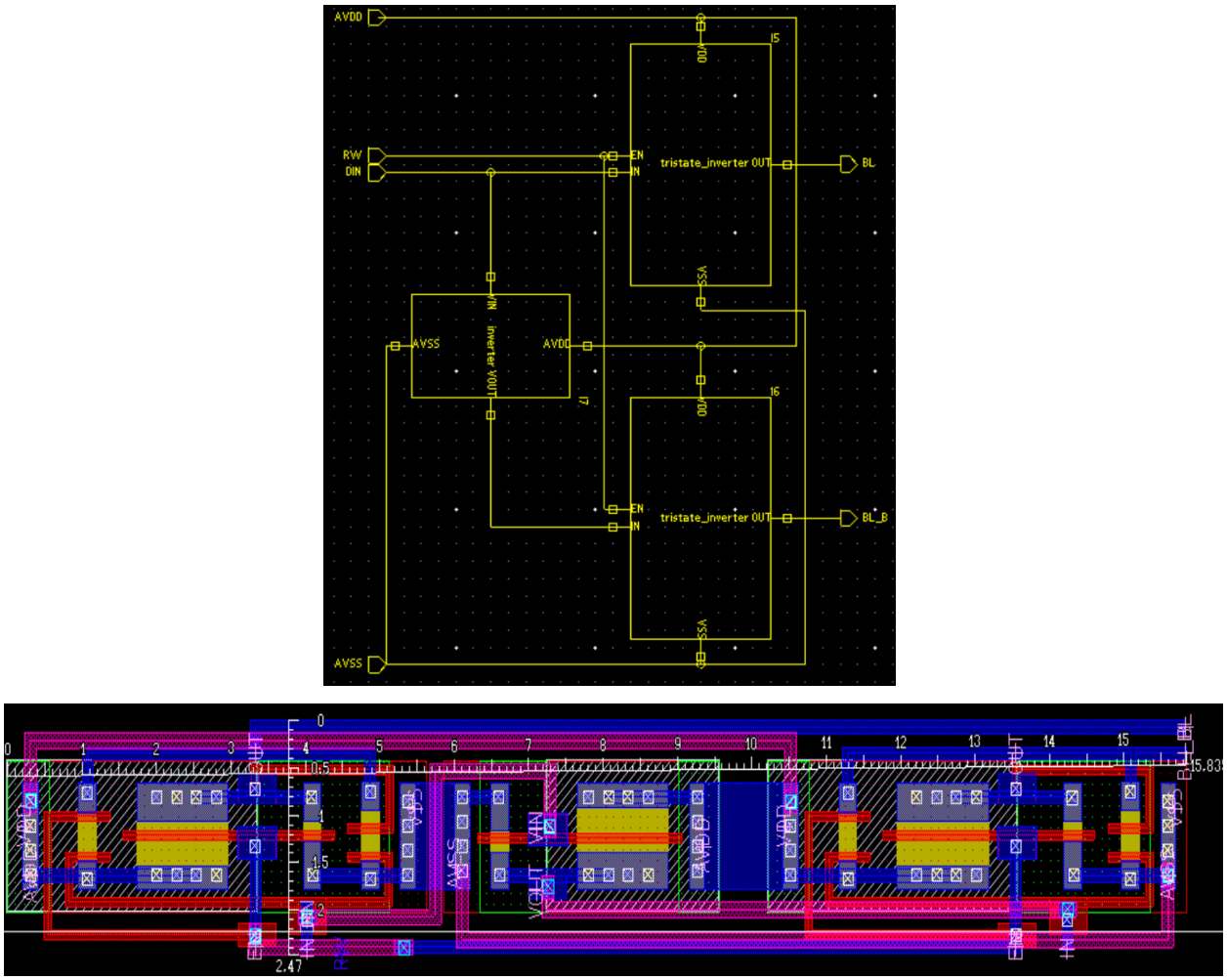


Figure 14 - Write Driver Schematic (Top) and Layout (Bottom); Area = 2.47µm x 15.83µm

The Write Driver will be capable of writing data provided at its input to the Bit-Lines in the SRAM Array. The write driver is designed using two Tri-State Inverters as well as an Inverter for inverting the data that will be written to BL_B. When R/W is low, the Write Driver will be active and its Tri-State Inverters will pull the Bit-Lines to whatever we are trying to store. Furthermore, the word selected by the decoder will be overwritten by the data from the Write Driver. During a Read Operation, that is when R/W is high, the Tri-State Inverters will output their floating state and the Bit-Lines will not be affected by the Write Driver.

SRAM

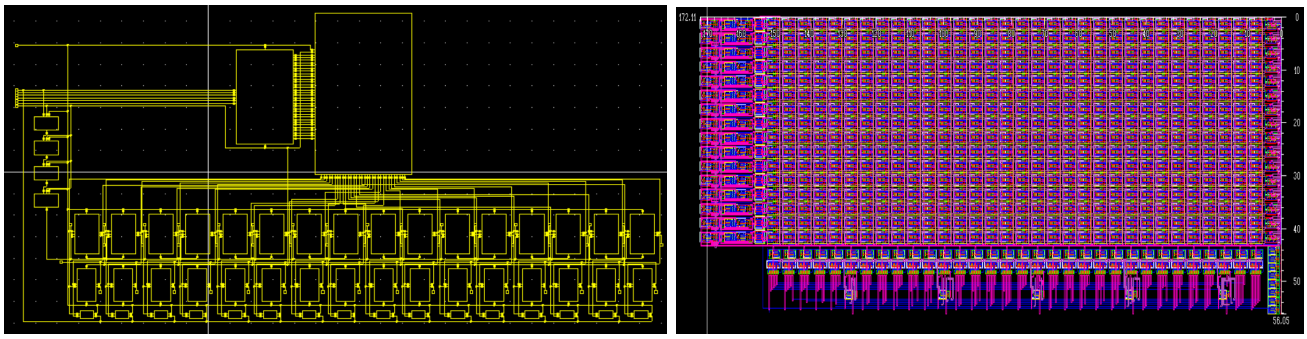


Figure 15 - SRAM Schematic (Top) and Layout (Bottom); Area = 56.05µm x 172.11µm

Integrating all components previously discussed, we get the design shown in Figure 15. At this stage, we must also determine how much we should delay the Sense Enable signal such that we can achieve a 100mV differential before amplification which would take 191ps. The global CLK signal will be delayed using 4 Buffers, producing a 212ps delay shown in Figure 16.

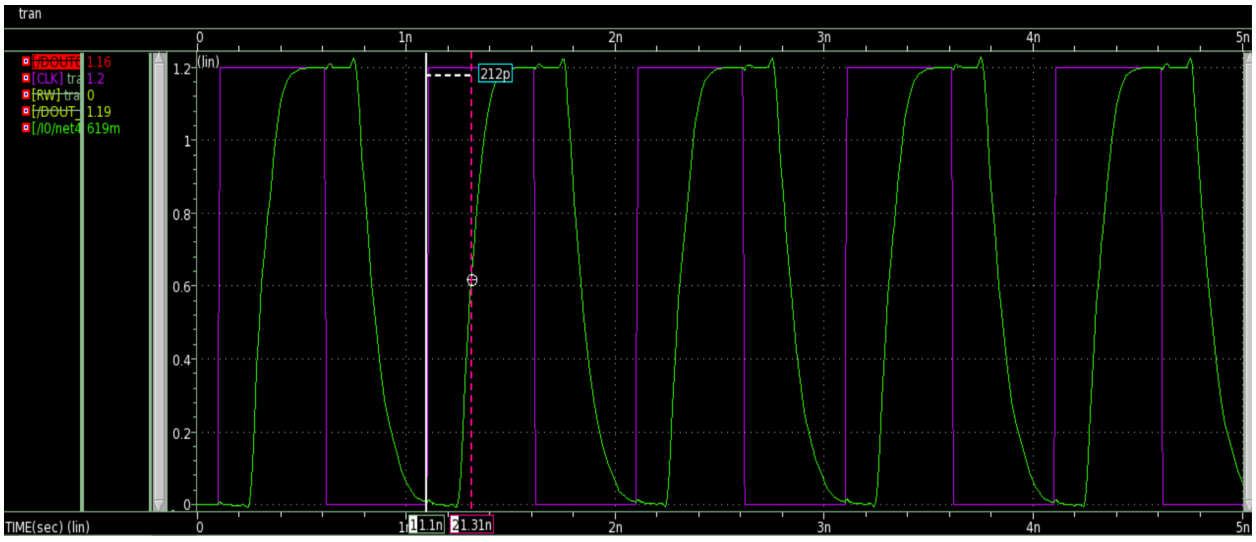


Figure 16 - Sense Amplifier Delayed Enable

We simulate the integrated SRAM design in Figure 17 at 500 MHz. Some of the behaviors we had predicted earlier are apparent in this waveform such as the fact that both outputs from the Sense Amplifier are always high when CLK is low, regardless of what is being stored in the

memory. We can see that the SRAM behaves correctly, first storing a word of value 0 to Address 0, then that same word being read out to us later on.

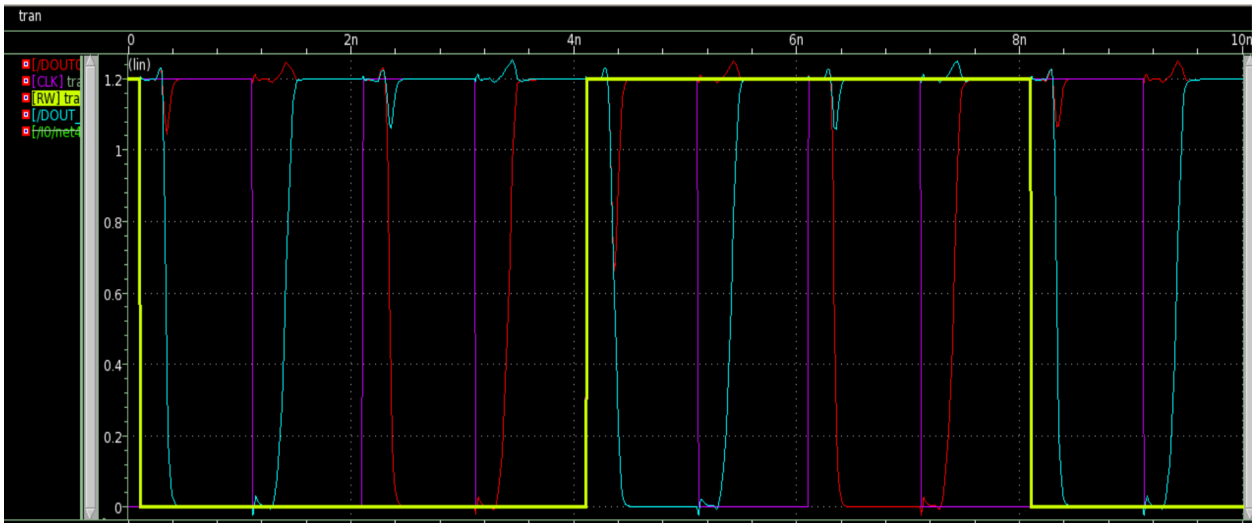


Figure 17 - SRAM Verification Pre-Layout

Layout also passes DRC and LVS successfully as per the reports shown in Figure 18.

```
                        LVS Compare Results: PASS

                         ####   ###    ####   ####
                        #   #  #   #  #      #
                        ####   #####  #####  #####
                        #      #   #      #      #
                        #      #   #  ####   ####

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

                      DRC and Extraction Results: CLEAN

                         ####  #      #####   ###   #   #
                        #      #      #      #   #  ##  #
                        #      #      ####   #####  # # #
                        #      #      #      #   #  #  ##
                         ####  #####  #####  #   #  #   #

================================================================================


--------------------------------------------------------------------------------
ICV Execution
--------------------------------------------------------------------------------


                                IC Validator (R)

          Version K-2015.12-SP2-7 for linux64 - Dec 16, 2016 cl#3476058

                      Copyright (c) 1996 - 2016 Synopsys, Inc.
     This software and the associated documentation are proprietary to Synopsys,
   Inc. This software may only be used in accordance with the terms and conditions
   of a written license agreement with Synopsys, Inc. All other use, reproduction,
              or distribution of this software is strictly prohibited.

Called as: icv -f openaccess -i mylibrary -c sram -oa_view layout -oa_lib_defs /courses/engr848/engr848-16/lib.defs -s /courses/engr848/engr848-
16/synopsys_custom/sram.icv.lvs/sram.custom_compiler.sp -sf SPICE -stc sram -oa_dm5 -vue /courses/engr848/engr848-
16/synopsys_custom/sram.icv.lvs/rules.lvs.9m_saed90_lvs.lvs.rs
```

Figure 18 - SRAM DRC and LVS Results

Returning to the delay needed for the Sense Amplifier, we observe in Figure 19, whether the 100mV Bit-Line differential is still being met with Parasitics applied. As expected, the 100mV

differential is still being met, with the Sense Amplifier being enabled 414ps after the clock rises.

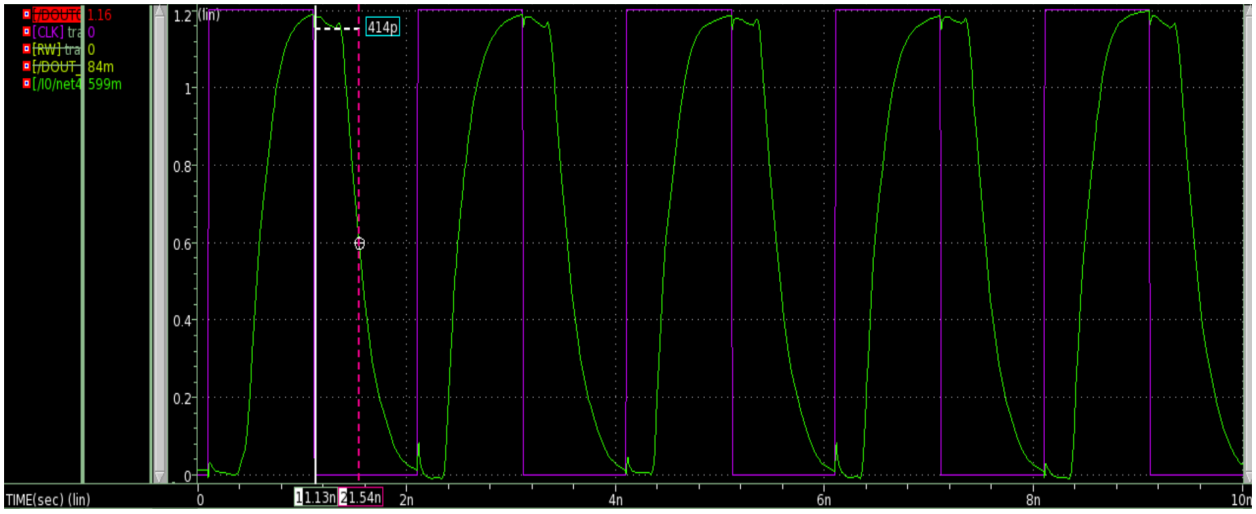


Figure 19 - Sense Amplifier Delayed Enable with Parasitics

We perform the same Verification on the SRAM after extraction of parasitics whose results are shown in Figure 20. Due to the increase in delay from the Sense Enable as well as other components, the Sense Amplifier will have less time to produce a stable output before being deactivated, resulting in the waveform exhibiting more fluctuation from the waveform produced in Figure 17.

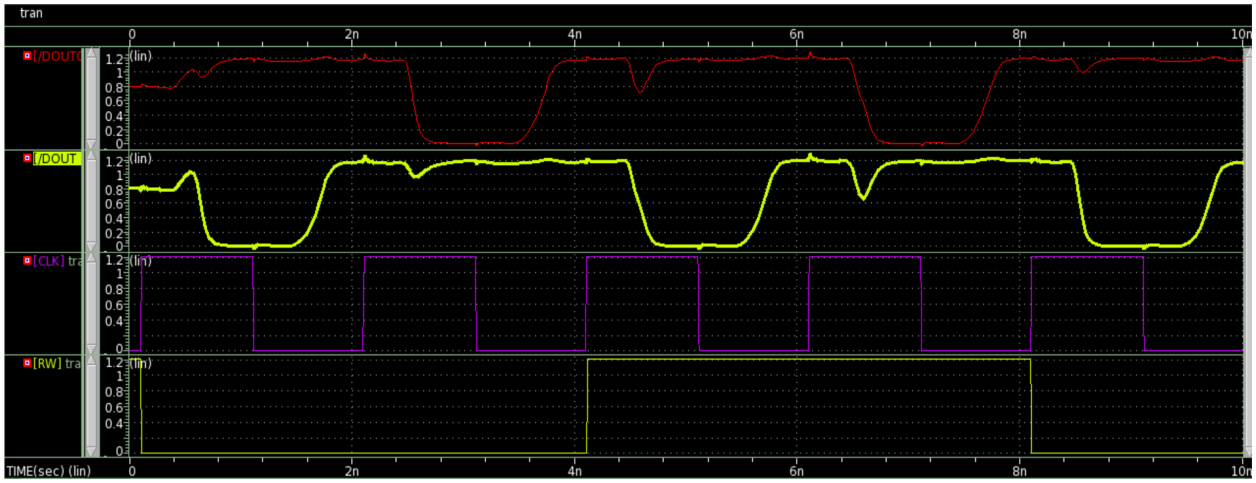


Figure 20 - SRAM Verification Post-Layout @500MHz

Nevertheless, we can further increase the Clock Frequency of the SRAM and still achieve correct results as shown in Figure 21. In fact, we can increase the Clock Frequency as long as it remains below the SRAM's read-access time which is shown to be 470ps when reading 0 and 484ps when reading 1. This allows for a maximum frequency of 2.07GHz!

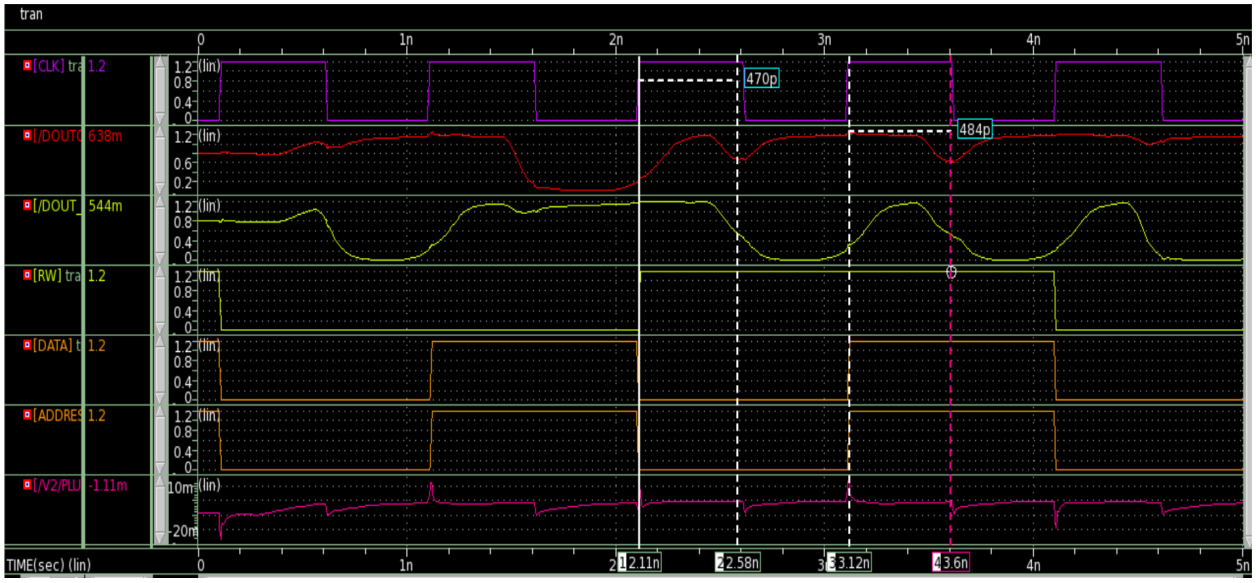


Figure 21 - SRAM Verification Post-Layout @1GHz

It is important that our design also maintains a low Power Consumption for it to be viable in a consumer setting. Our design is able to achieve a static power consumption of 168µW, so its power consumption won't be too high when the SRAM and its inputs are remaining idle. Active Power consumption was found to be 2.628mW, describing how much power is consumed when the SRAM is performing several write and read operations to different addresses such as the behavior shown in Figure 22.

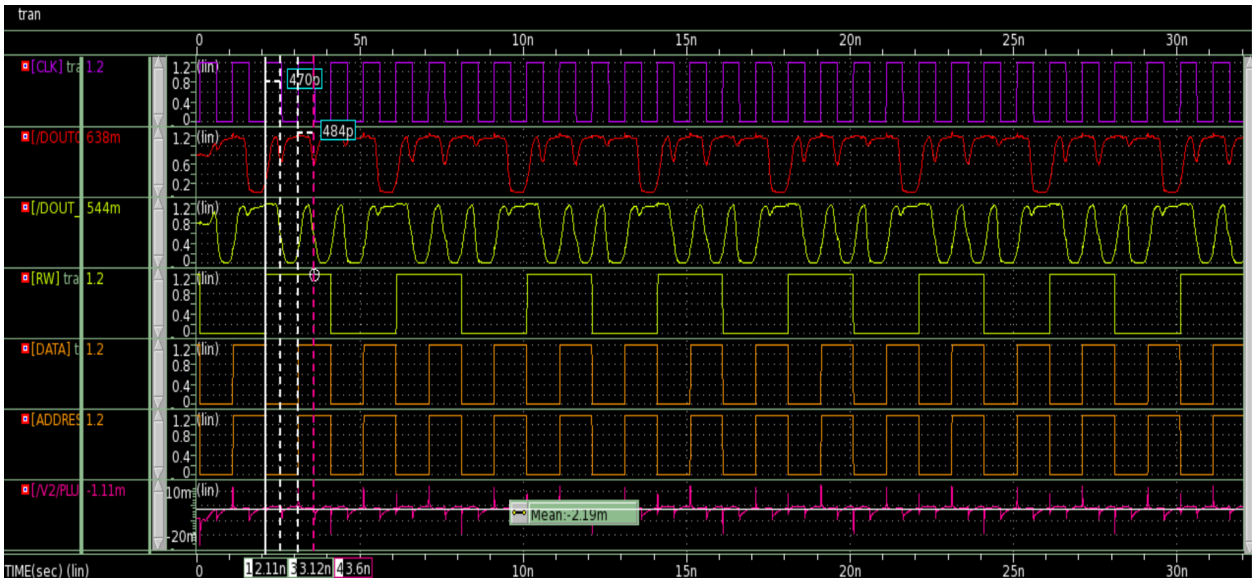


Figure 22 - SRAM Power Verification Post-Layout @1GHz

Conclusion

In this project, it requires working carefully from step to step to have the right outcome. For each logic gate or each designed component, the schematic needs to be checked thoroughly from power to delay. Especially, the sizing needs to be done, and parts need to be integrated together before the layout to have proper outcome.