

# Design of a Low-Leakage Low-Power 2-Kbit Custom SRAM for a Medical Implant

Institute for Integrated Circuits  
Hamburg University of Technology

**Project work**  
for the attainment of the academic degree  
Master of Science

submitted by

**Rafiul Islam**

born on April 22, 1994 in Dhaka

August 18, 2021

**First examiner:** Prof. Dr.-Ing. Matthias Kuhl  
**Second examiner:** Prof. Dr.-Ing. Wolfgang Krautschneider

**Supervisor** M. Sc. -Ing. Mohamed Faragalla

# **Declaration of authorship**

I certify that I have written this work independently and that I have not used any sources or aids other than those indicated. Both content and verbatim content have been marked as such. The work has not been submitted in the same or similar form to any other examination office. I agree to the dissemination of my work for scientific purposes.

.....

Place, Date

.....

Signature

# Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>Abstract</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 6T SRAM Unit Cell . . . . .	1
1.2 Basic SRAM Design Parameter . . . . .	1
1.2.1 Cell Stability . . . . .	2
1.2.2 Low Leakage Operation . . . . .	3
<b>2 SRAM Core Design</b>	<b>4</b>
2.1 Leakage Reduction Technique . . . . .	4
2.1.1 Power Gating Technique . . . . .	4
2.1.2 Body Biasing Technique of MOSFET . . . . .	5
2.1.3 Short Channel Effect of the MOSFET . . . . .	6
2.2 Inverter characterization . . . . .	7
2.3 Cell Stability Estimation . . . . .	10
<b>3 SRAM Peripheral Circuits</b>	<b>14</b>
3.1 SRAM System . . . . .	14
3.2 Multiplexers and Decoders . . . . .	16
3.2.1 Column Multiplexer . . . . .	16
3.2.2 Row Decoder . . . . .	17
3.3 Write Driver . . . . .	17
3.4 Sense Amplifier . . . . .	19
3.5 Precharge Circuit . . . . .	20
3.6 Control Unit . . . . .	20
3.6.1 Address Transition Detection . . . . .	22

3.6.2	Word Line enable and Reset Signal Generation . . . . .	22
3.6.3	Write enable Signal Generation . . . . .	23
3.6.4	Read enable Signal Generation . . . . .	23
3.7	SRAM Floorplan . . . . .	24
3.8	SRAM Layout . . . . .	25
<b>4</b>	<b>Verification and Simulation Result</b>	<b>26</b>
4.1	Control Unit Simulation . . . . .	26
4.2	SRAM Unit Cell Simulation . . . . .	27
4.3	Stand-by Current Simulation . . . . .	29
4.4	Verification of the SRAM . . . . .	29
4.5	SPI Interface . . . . .	31
<b>5</b>	<b>Conclusion</b>	<b>34</b>
<b>Bibliography</b>		<b>35</b>
<b>A Appendix</b>		<b>37</b>
<b>B Appendix</b>		<b>39</b>

# List of Figures

1.1 A 6T SRAM cell schematic [1] . . . . .	2
1.2 The standard setup for SNM definition [2] . . . . .	2
1.3 Graphical representation of SNM [1] . . . . .	3
2.1 Power gating technique . . . . .	4
2.2 Variable Threshold CMOS (VTCMOS)) [3] . . . . .	5
2.3 Conduction band energy versus lateral distance(normalized to the channel length) [4] . . . . .	6
2.4 PMOS leakage current . . . . .	7
2.5 NMOS leakage current . . . . .	8
2.6 Difference in Subthreshold Current . . . . .	10
2.7 NML versus access transistor dimension . . . . .	11
2.8 NMH versus access transistor dimension . . . . .	11
2.9 Write static noise margin . . . . .	12
2.10 Hold static noise margin . . . . .	12
2.11 Read static noise margin . . . . .	13
2.12 SRAM unit cell layout . . . . .	13
3.1 SRAM unit cell with peripheral circuit . . . . .	14
3.2 Timing diagram . . . . .	15
3.3 Block diagram of the multiplexer . . . . .	17
3.4 Circuit Diagram of 2:1 word multiplexer . . . . .	17
3.5 Circuit Diagram of Write Driver . . . . .	18
3.6 Waveform for the write driver . . . . .	18
3.7 Circuit diagram of sense amplifier . . . . .	19
3.8 Precharge Circuit . . . . .	20
3.9 Waveform for precharge Circuit . . . . .	21
3.10 Block diagram of control operation . . . . .	21
3.11 Address transition detection circuit . . . . .	22

3.12	Read/write positive transition detection circuit . . . . .	22
3.13	Word Line Enable and Reset Signal Generation circuit . . . . .	23
3.14	Write Enable Signal Generation circuit . . . . .	23
3.15	Read Enable Signal Generation circuit . . . . .	23
3.16	SRAM floorplan . . . . .	24
3.17	Layout of toal SRAM system . . . . .	25
4.1	Test-bench for control unit . . . . .	26
4.2	waveform of the control unit . . . . .	27
4.3	Waveform of the verification of the SRAM . . . . .	28
4.4	SRAM Core leakage current . . . . .	29
4.5	Waveform of the verification of the SRAM . . . . .	30
4.6	Operation of MOSI in SPI . . . . .	31
4.7	Operation of MISO in SPI . . . . .	32
4.8	SPI waverform for SRAM . . . . .	33
4.9	Expanded SPI waverform for SRAM . . . . .	33

# List of Tables

2.1	Differences in leakage current . . . . .	9
2.2	Static Noise Margin estimation . . . . .	10
4.1	Data for the verification of the SRAM . . . . .	30

# Abstract

This work presents a full design of a low-leakage and low-power Static Random Access Memory (SRAM). The SRAM is to be used in an implantable electronic chip to store invasively recorded biological data. Thus, minimizing the leakage and standby currents of the system is of high importance to maximize the operation time of the electronic implant.

The property of mitigation of short channel effects of the metal-oxide-semiconductor (MOS) transistors is analyzed to lower the leakage currents by proper sizing for the dimensions of the transistors. The leakage current of an SRAM unit cell i.e., one-bit storage cell, is reduced to  $87 \text{ fA}$  compared to  $702 \text{ fA}$  if standard-logic gates are used. Moreover, dual supply voltages are used for the storage core and the peripheral circuitry in order to eliminate the leakage current of the access circuits when the memory is in standby mode.

The building blocks of the SRAM core and peripheral circuits for accessing the storage cells as well as interface circuits for external communication are analyzed in this report. A design of 2-Kbit SRAM and serial-to-parallel interface (SPI) is realized and taped-out in  $180 \text{ nm}$  CMOS process. The total area of the SRAM system is  $500 \mu\text{m} \times 355 \mu\text{m}$ . The leakage current of the memory core is found to be approximately  $231 \text{ pA}$ .

# 1 Introduction

SRAM is type of volatile memory which is mainly used for the cache memory in the microprocessors, mainframe computers, engineering workstations and memory in the hand held devices due to high speed and low power consumption [5]. Being a volatile memory, it always requires the power to keep the data. However, when powered from energy storage elements such as batteries or capacitors, it becomes essential to reduce the standby or leakage current of the SRAM cells to ensure reliable operation under long data-holding intervals. In principle, SRAMs consist of core cells where data are stored and peripheral circuit to access the core cells. For applications where the SRAM operate in standby modes, it is reasonable to disconnect the peripheral circuits from the power source to eliminate their leakage current when the SRAM is not to be accessed. As a result, this project work focuses on designing an asynchronous SRAM with low leakage current and with the ability to switch off the peripheral circuits.

## 1.1 6T SRAM Unit Cell

A 6T SRAM unit cell, shown in Fig. 1.1, includes two cross-coupled inverters formed by two pull-up transistors (PU), two pull-down transistors (PD) and two pass transistors (PG) for accessing the cell. In order to select the particular cell from the array, word-lines (WL) and bit-lines (BL and BLP) are used [1]. During both read and write operation, at first the bitlines are precharged to  $V_{DD}$  or  $V_{DD}/2$  and after then, the Word Line (WL) is asserted to have access in the storage node of the SRAM cell.

## 1.2 Basic SRAM Design Parameter

There are few basic parameters which needs to be considered while designing the SRAM such as cell stability, memory operation speed and low power operation. For this project, memory operation speed is not highly important parameter since as a requirement, the

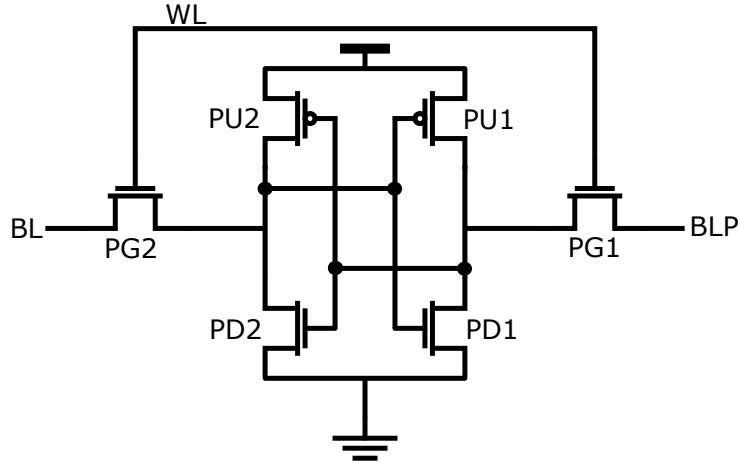


Figure 1.1: A 6T SRAM cell schematic [1]

SRAM read and write operation frequency is not very high. The write operation has the frequency of 4 MHz and the read operation has the frequency of 1 MHz.

### 1.2.1 Cell Stability

The parameter by which cell stability of SRAM is measured is called Static Noise Margin (SNM). It is defined as the maximum value of DC noise voltage that can be tolerated by SRAM cell without changing the stored bit. The equivalent circuit for SNM definition is shown in Fig. 1.2. The two DC noise voltage sources ( $V_n$ ) are placed in series with the inverters at the internal nodes of the cell. As ( $V_n$ ) increases, the stability of the SRAM cell changes. A 6T-SRAM cell has two stable state in read mode because it must retain its state in the presence of both precharged bitline voltages. However, it has only one stable state in write mode of operation because write SNM means the minimum bit line voltage to flip the state of the cell [2].

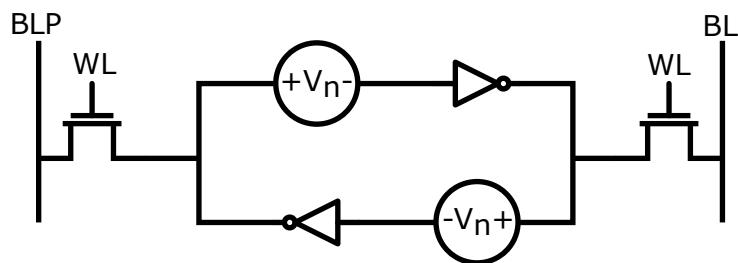


Figure 1.2: The standard setup for SNM definition [2]

A butterfly curve can be formed by plotting the voltage transfer characteristics of the cell inverters. The points at which the maximum square drawn are the logical stable points

where  $dV_{out}/dV_{in} = -1$  of a bistable inverter pair. SNM can be calculated as:  $SNM = \min(NM_L, NM_H)$  and  $SNM' = \min(NM'_L, NM'_H)$  where  $NM_L, NM_H, NM'_L, NM'_H$  are defined in figure 1.3 [1].

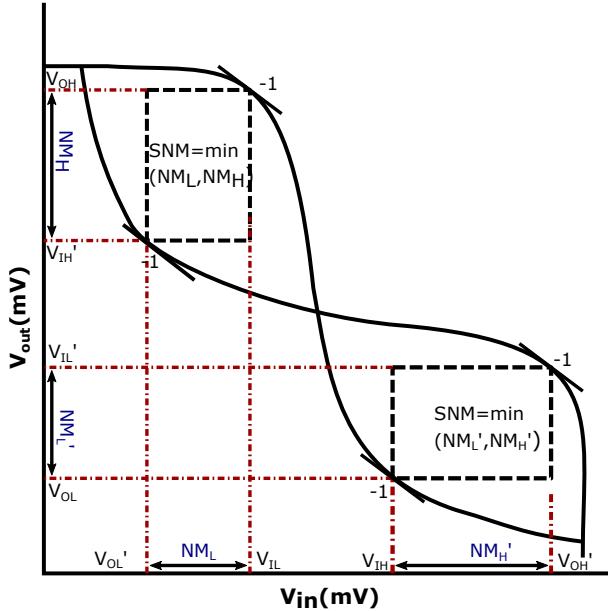


Figure 1.3: Graphical representation of SNM [1]

### 1.2.2 Low Leakage Operation

There are many methods proposed to reduce the leakage current of SRAM without the data hazard. The most common two techniques used are: source biasing and dynamic  $V_{DD}$  [6]. There are obviously some advantages and disadvantages in all of the approaches. However, the trade-off between the merits and demerits of these methods mostly depend on the purpose of the application of the memory and how it will be used in the system. For this project, Minimization of short channel effect approach has been used to reduce the power during the sleep mode.

The report is organized as follows:

- Chapter 2: describes few leakage reduction technique, the theory of short channel effect, minimizing the effect to form unit cell and the SRAM array.
- Chapter 3: discusses all the peripheral circuits required for the proper functioning of the SRAM.
- Chapter 4: presents all the verification and simulation results of the SRAM.

# 2 SRAM Core Design

This chapter describes few leakage reduction technique in brief, the short channel effect of the MOSFET, utilization of this characteristics for the inverter characterization of the 6T SRAM cell and finally the formation of the bit cell and the array of the SRAM.

## 2.1 Leakage Reduction Technique

### 2.1.1 Power Gating Technique

The conventional power gating devices can be classified into two main categories: header and footer devices. They are also known as dynamic voltage source and source biasing technique respectively. Header device is set by inserting PMOS sleep transistors between real voltage source and virtual voltage source and footer is set by inserting NMOS sleep transistors between real ground and virtual ground shown in Fig. 2.1. The power gating devices are controlled by the power control unit of the SRAM [7].

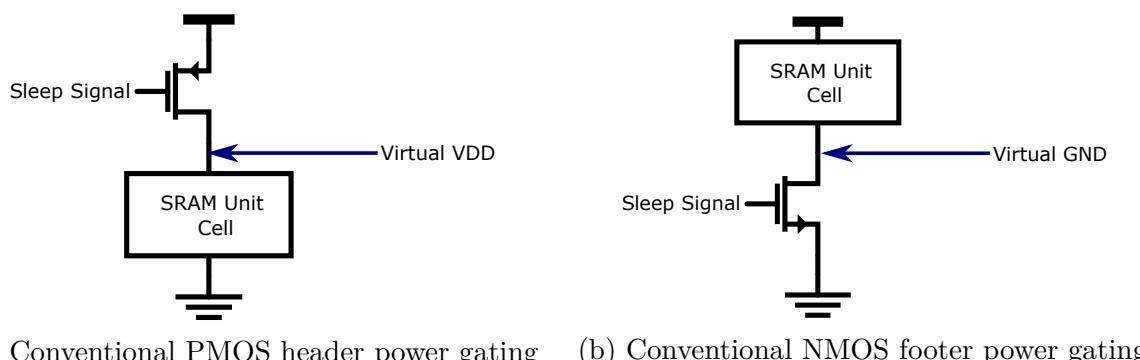


Figure 2.1: Power gating technique

In case of header device, a critical voltage level called data retention voltage must be defined before lowering the supply voltage. It is the minimum voltage level where the data is not destroyed during standby period [7]. As a result, the leakage power becomes lower during sleep mode. Although it reduces sub-threshold, gate and band-to-band tunneling leakage,

the bit-line leakage current cannot be reduced in this technique. Another drawback is the necessity of dedicated switchable supply voltage [6].

In case of footer device, the source line voltage is raised to generate a negative  $V_{GS}$  which leads to the body effect in NMOS transistors and hence, lowers the leakage current [6]. Unlike the header device, footer device is able to reduce the bit-line leakage as well. However, a DC-DC converter is required to lift the ground voltage [8]. A multiple threshold-voltage CMOS (MTCMOS) can be additional technique to reduce the leakage current. In this technique, the sleep control transistors are designed with high  $V_{th}$  transistors where the rest of the SRAM unit cell is designed with low  $V_{th}$  transistors [3].

### 2.1.2 Body Biasing Technique of MOSFET

In general, the body biasing is varying the substrate voltage of the transistor at different voltage levels. It is also known as Variable Threshold CMOS (VTCMOS) technique shown in Fig. 2.2. To achieve different threshold voltages, a self-substrate bias circuit is used to control the body bias. In the active mode, a nearly zero body bias is applied. While in the standby mode, a deeper reverse body bias is applied. The substrate bias voltage has a direct impact on the magnitude of the threshold voltage of a particular transistor which in turn affects the leakage current. The threshold voltage of MOSFET changed by varying  $V_{SB}$  can be expressed by following equation [9]

$$V_{th} = V_{th0} + \gamma(\sqrt{|2\phi_F - V_{BS}|} - \sqrt{|2\phi_F|}) \quad (2.1)$$

where,  $V_{th0}$  is the threshold voltage of the device without body bias,  $\gamma$  is the coefficient

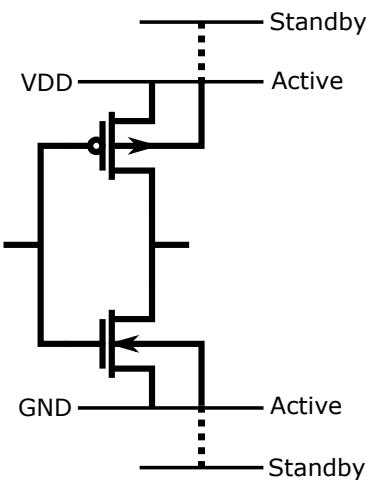


Figure 2.2: Variable Threshold CMOS (VTCMOS)) [3]

of body effect. The Body bias technique may reduce the sub-threshold leakage if the substrate voltage for NMOS can be increased. In the reverse body bias scheme, the depletion width of the diode formed by source and substrate is increased which results in an increase of the threshold voltage. A higher gate voltage is required for inverting the substrate of the MOSFET. Therefore, an optimized reverse body bias is capable of reducing the overall leakage current [10].

### 2.1.3 Short Channel Effect of the MOSFET

The short channel effect is one of the important characteristics of short channel device which needs to be considered while designing. It is a phenomenon occurred when MOSFET's threshold voltage decreases due to the reduction in channel length. The effect of short channel of MOSFET can be understood considering potential barrier at the surface between the source and drain shown in figure 2.3.

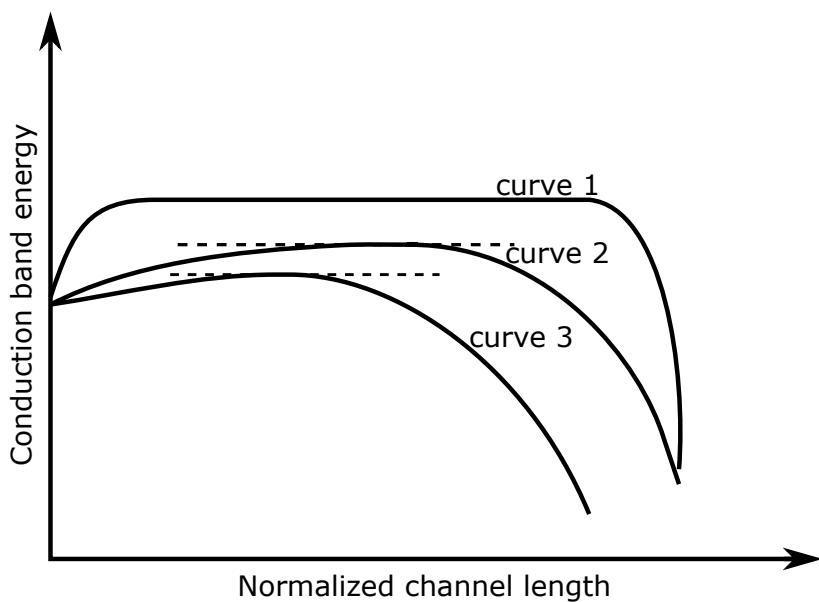


Figure 2.3: Conduction band energy versus lateral distance(normalized to the channel length) [4]

When the transistor is turned off, the current flowing is prevented by the potential barrier. However, a limited number of electrons are still able to escape the source over the barrier and collected by the drain which is basically known as the subthreshold leakage current. When the transistor is turned on, the structure of potential barrier depends on the channel length. In case of long channel MOSFET, the potential barrier is flat over the most part of the device similar to the curve 1 of Fig. 2.3. On the other hand, in case of the short

channel MOSFET, the potential barrier between the source and the drain is lowered due to the penetration of the field into the middle of the channel which causes the considerable increase in subthreshold current. The curve 2 of Fig. 2.3 depicts this situation. Finally, the curve 3 of Fig. 2.3 represents another phenomenon called ‘drain-induced barrier lowering (DIBL)’. This situation occurs to the short channel device when a high drain voltage is applied which lowers the potential barrier even more [4]. In a nutshell, it is clear that the increase in channel length of the device reduces the subthreshold leakage current. This property has been used to find out the MOSFET dimensions with very low leakage current. However, this procedure naturally produces a discussion or the trade off between the area versus the leakage current of the MOSFET which will be discussed in the following sections.

## 2.2 Inverter characterization

Figure 2.4 and 2.5 depict the PMOS and NMOS leakage current respectively.

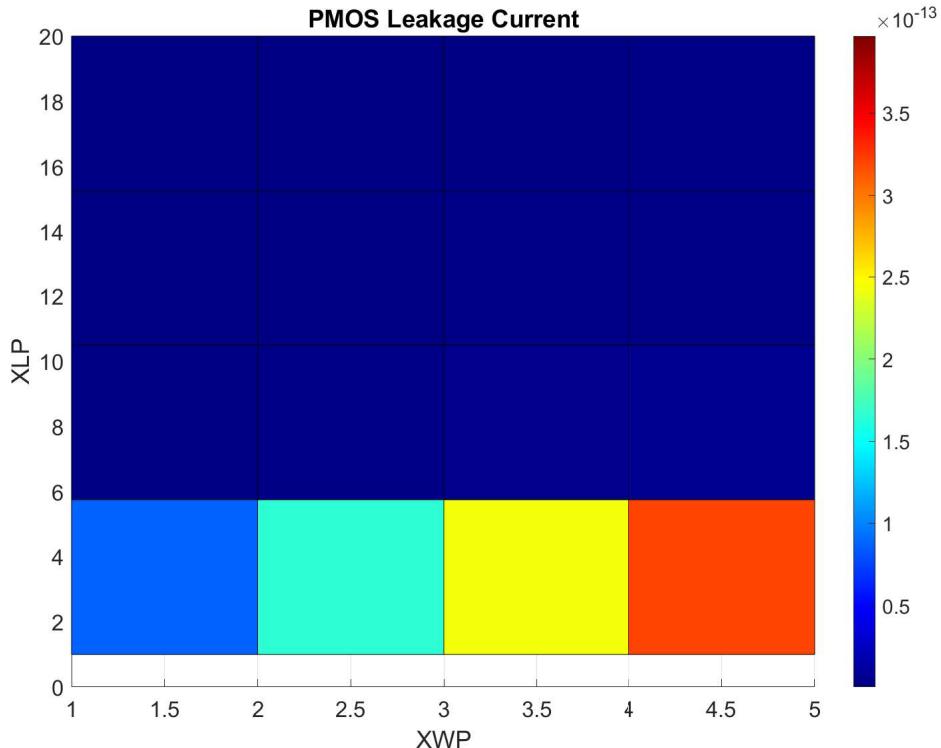


Figure 2.4: PMOS leakage current

The x-axis and y-axis of both figure are defined as follows:

- XWP: PMOS Width Multiplier
- XLP: PMOS Length Multiplier
- XWN: NMOS Width Multiplier
- XLN: NMOS Length Multiplier

This graphs in Fig. 2.4 and 2.5 are generated by taking a unit dimension for the MOSFET and swiped over the multiplier to find out the point where the leakage current is lower with the least dimension possible. Both Fig. 2.4 and 2.5 show that the increase in channel length of the MOSFET is reducing the leakage current as discussed in section 2.1.3. Also, the width of the MOSFET has no influence in leakage current reduction as expected.

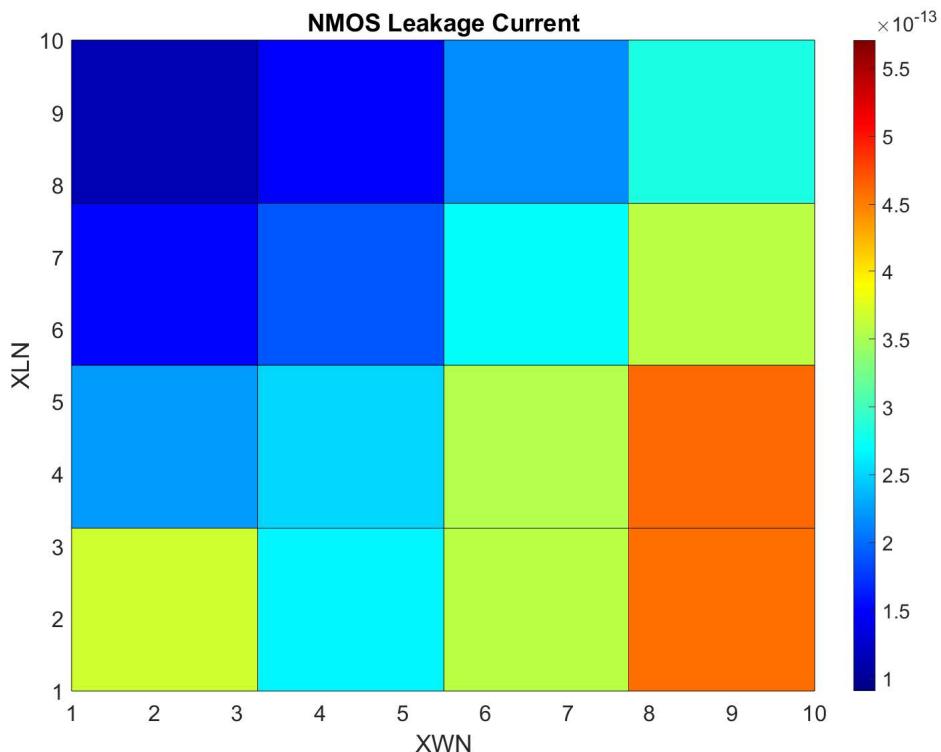


Figure 2.5: NMOS leakage current

Finally, the inverter dimensions are selected with respect to Fig. 2.4 and 2.5 as follows:

- PMOS Width: 1.6 um
- PMOS Length: 1.5 um
- NMOS Width: 1 um
- NMOS Length: 5.2 um

Since the selected inverter area and leakage current is already known from figure 2.4 and 2.5, these parameters are compared to the standard cell library(provided by the technology vendor) inverter's area and leakage current. The comparison is shown in table 2.1.

Table 2.1: Differences in leakage current

	Standard Library Inverter		This Design
	Inverter-1	Inverter-2	
Area ( $\mu m^2$ )	10.39	22.93	21.33
NMOS leakage current (fA)	1466	702	87.5
PMOS leakage current (fA)	500	149	2.24

There is another important observation from the table 2.1 that the PMOS leakage current significantly reduces with the increase in length compared to the NMOS leakage current. This issue is further examined by plotting the subthreshold leakage current equation. The equation of subthreshold leakage current is [11],

$$I_D = \beta_{eff} \cdot (n - 1) \cdot V_T^2 \cdot \exp \frac{V_{GS} - V_{th}}{n \cdot V_T} \cdot (1 - \exp \frac{-V_{DS}}{V_T}) \quad (2.2)$$

where,  $\beta_{eff} = \mu_{eff} \cdot C_{ox} \cdot \frac{W}{L}$ , n= Ideality factor,  $V_T$  = Thermal Voltage,  $V_{th}$  = Threshold Voltage. Equation 2.2 can be expressed as the ratio of the subthreshold current of the NMOS to that of the PMOS assuming that both transistors have the same  $\beta_{eff}$  and terminals voltages,  $|V_{GS}|$  and  $|V_{DS}|$ .

$$\frac{I_{D,n}}{I_{D,p}} = \frac{(n_n - 1)}{(n_p - 1)} \frac{\exp \frac{V_{GS} - V_{th,n}}{n_n \cdot V_T}}{\exp \frac{V_{GS} - V_{th,p}}{n_p \cdot V_T}} \quad (2.3)$$

Figure 2.6 representing equation 3.3 shows the ratio of the subthreshold current of the NMOS to that of the PMOS versus the difference of their absolute threshold voltages with various ideality factor considering both transistors have the same  $\beta_{eff}$  and terminals voltages,  $|V_{GS}|$  and  $|V_{DS}|$ . This figure represents the analytical inspection of the subthreshold leakgae current equation to find out the parameter having the effect most in enhancing the leakage current. In essence, one PMOS and one NMOS with similar  $\beta_{eff}$  were chosen and swept over the the difference in absolute threshold voltage of PMOS and NMOS, and the ideality factor. It is clear from Fig. 2.6 that the change in threshold voltage between a NMOS and a PMOS has the more significant impact on the ratio of subthreshold leakage current rather than the ideality factor.

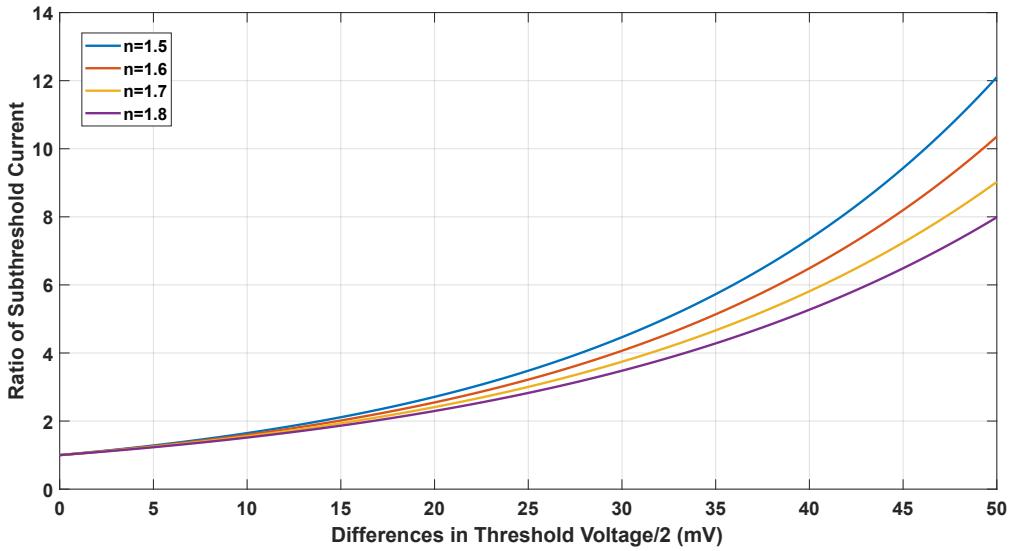


Figure 2.6: Difference in Subthreshold Current

## 2.3 Cell Stability Estimation

It is already discussed in the section 1.2.1 that the SRAM cell stability is measured by Static Noise Margin (SNM). However, the SNM of the SRAM cell depends on the relative strengths of the access, pull-up and pull-down transistors (see Fig. 1.1). In order to operate the SRAM properly, the strengths of these three transistors from strongest to weakest are pull-down, access, and pull-up in sequence [6]. The dimension of the access transistor of the unit cell were swept for finding out the best dimension for the cell stability. Figure 2.7 and 2.8 show the variation in noise margins due to variation in dimensions of the access transistor in the unit cell. From this two figures, an optimum point was chosen as the dimension of the access transistor.

Since the unit cell is designed, all types of SNM of the SRAM cell are now determined from Fig. 2.9, 2.10 and 2.11. Table 2.2 presents the numerical value of the SNM.

Table 2.2: Static Noise Margin estimation

	Static Noise Margin	
	Write Stability	Read Stability
NML	502.9 mV	246.3 mV
NMH	1.375 V	650.3 mV
NMLP		246.3 mV
NMHP		650.3 mV

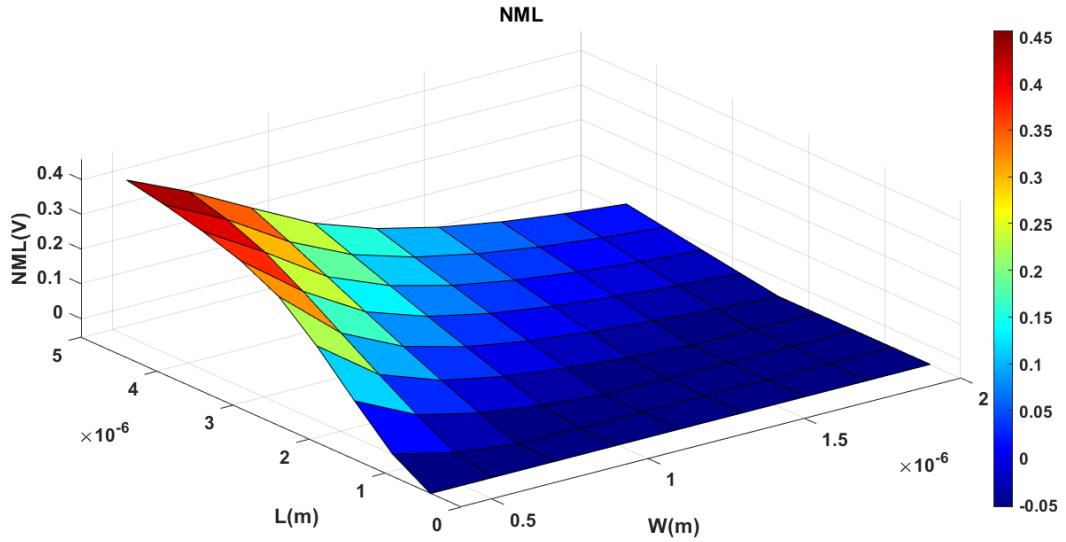


Figure 2.7: NML versus access transistor dimension

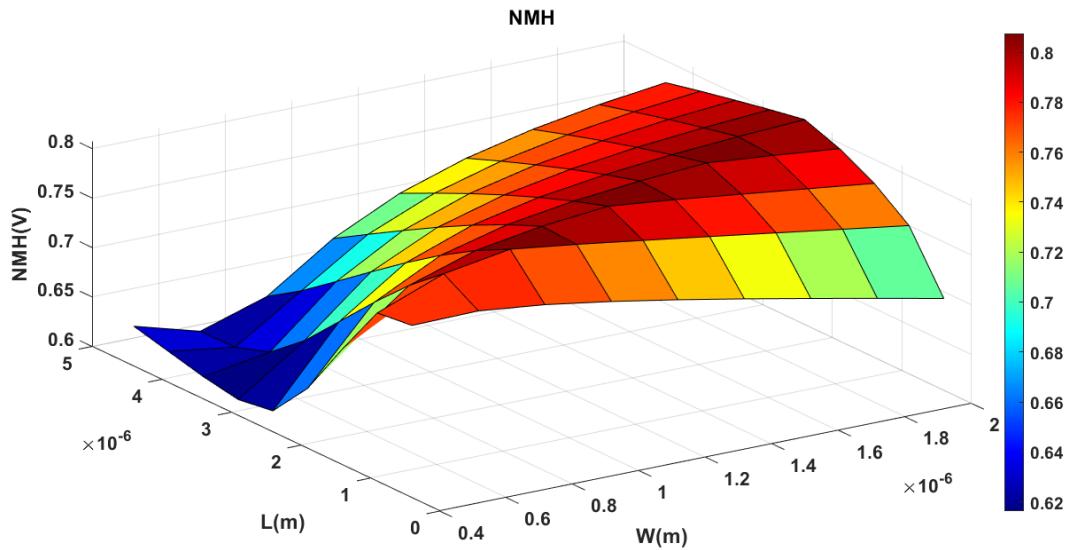


Figure 2.8: NMH versus access transistor dimension

The unit cell layout is shown in Fig. 2.12. The layout is designed symmetrically such that it can be easily merged with other unit cells and have the least possible area. The unit cell area dimension is  $7.74 \mu m \times 7.12 \mu m$ .

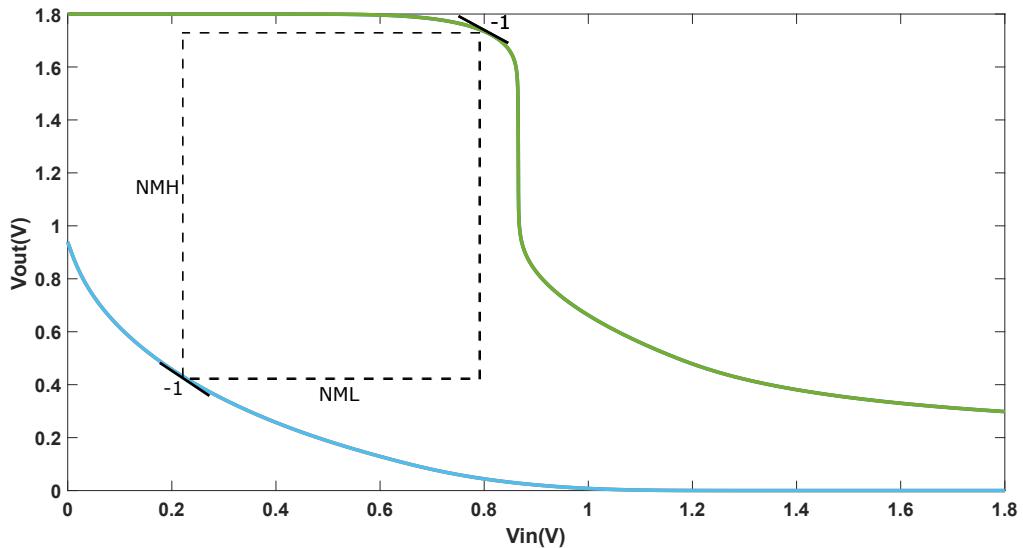


Figure 2.9: Write static noise margin

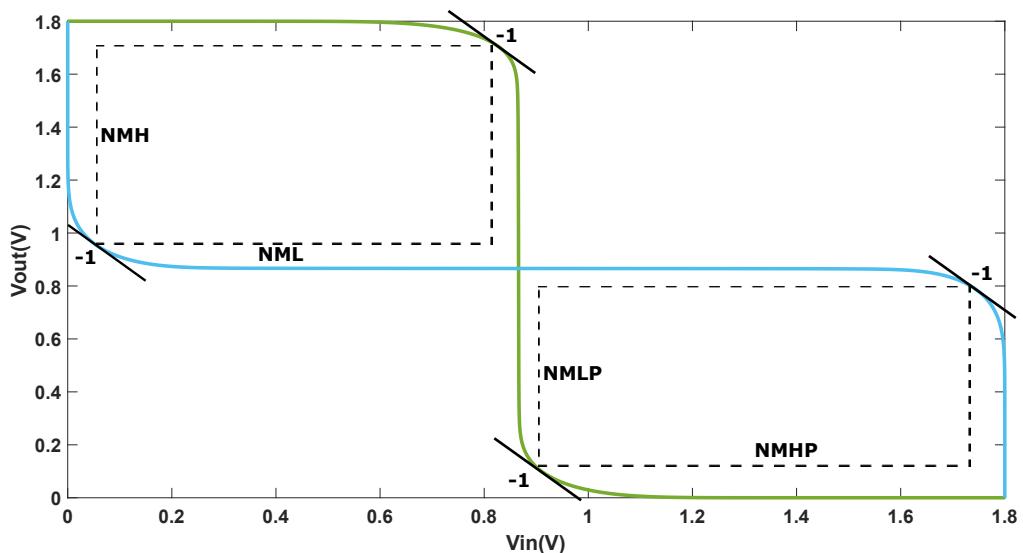


Figure 2.10: Hold static noise margin

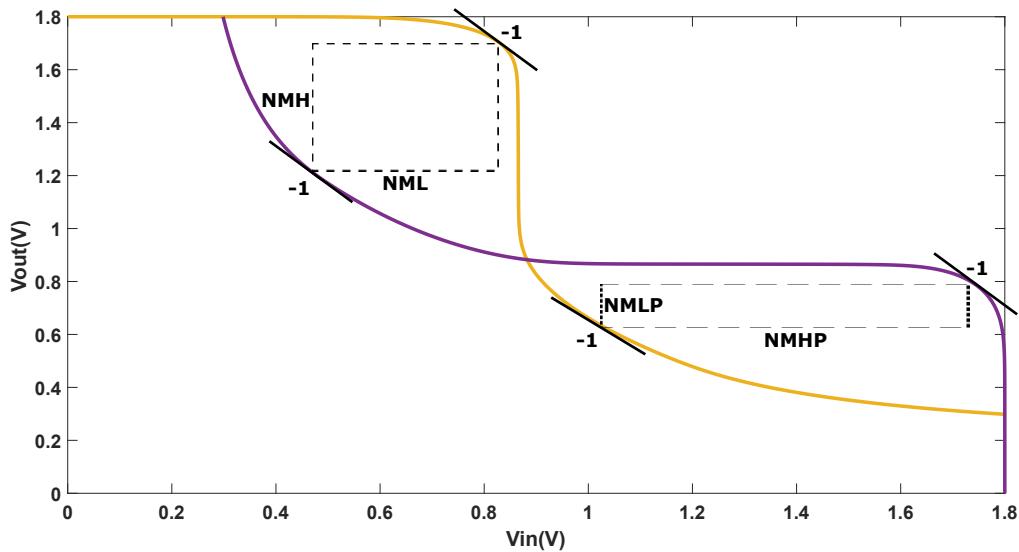


Figure 2.11: Read static noise margin

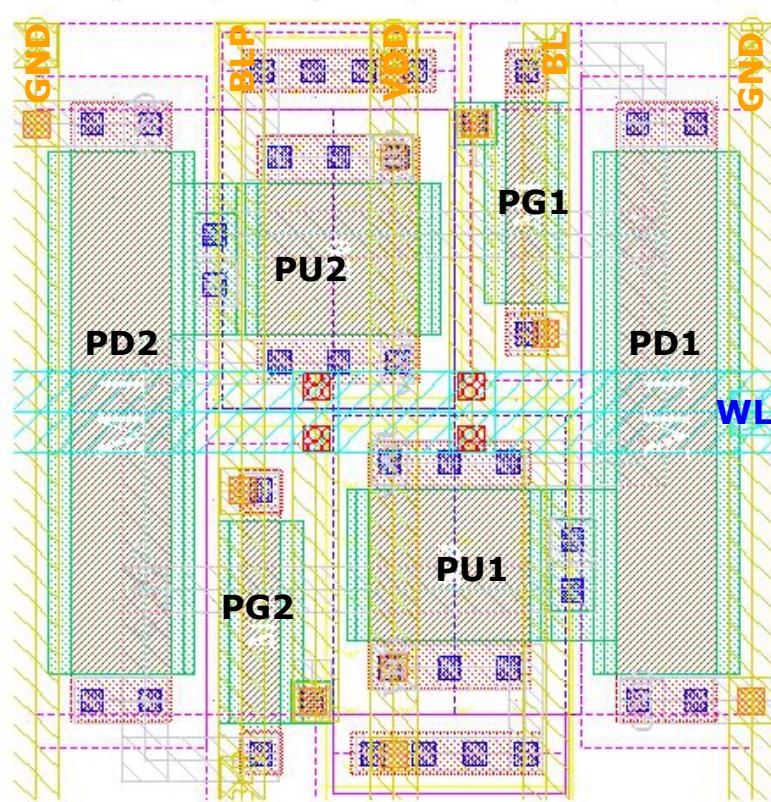


Figure 2.12: SRAM unit cell layout

# 3 SRAM Peripheral Circuits

This chapter describes the required peripheral circuits for the SRAM to function properly. How the peripheral circuits are controlled for a single cycle is a matter of discussion in this chapter.

## 3.1 SRAM System

Figure 3.1 shows the peripheral circuits encircling an SRAM unit cell.

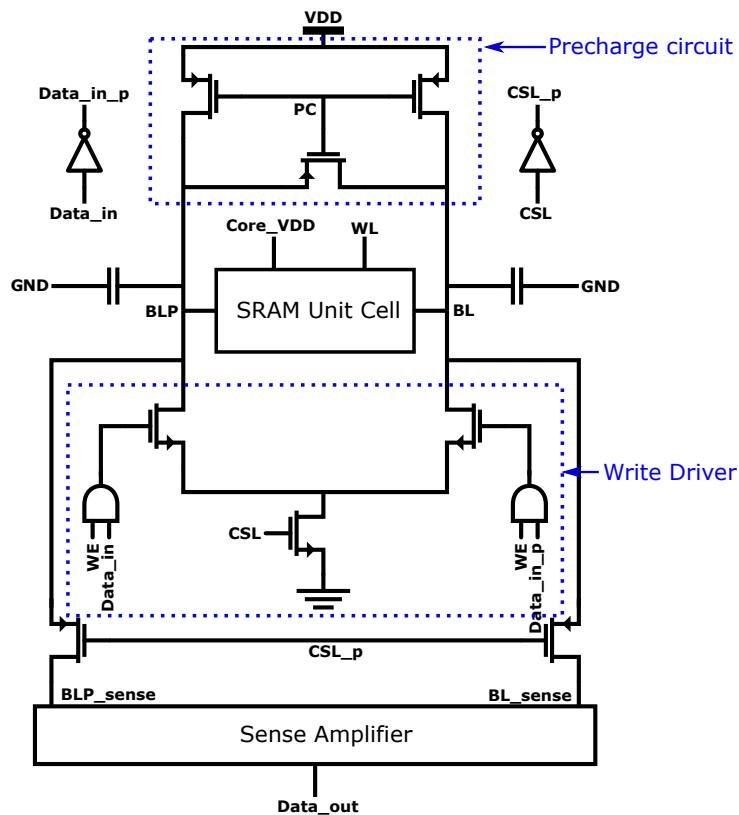


Figure 3.1: SRAM unit cell with peripheral circuit

Figure 3.2 exhibits the conceptual write and read timing diagram for a unit cell. Both read and write operation are triggered by the ‘Address Transition Detection (ATD)’ circuit

and after that, by lowering the ‘*precharge (PC)*’ signal. In case of write operation, the ‘*Write Enable (WE)*’ and ‘*Column Selection (CSL)*’ are enabled to activate the write driver. ‘*Word Line (WL)*’ is enabled a little bit later allowing one of the bit-lines pulling down to 0 V. Hence, the cell may flip depending on the ‘*data\_in*’ signal [5]. The events are shown in Fig. 3.2a.

In case of read operation, the ‘*WL*’ signal is activated after the ‘*PC*’ signal is lowered. Depending on the stored value, either ‘*BL*’ or ‘*BLP*’ start to drop its voltage below  $V_{DD}/2$ . ‘*Sense Amplifier Enable (SAE)*’ signal is enabled to activate the sense amplifier in order to capture the bit-line changes and generate a valid logic value [5]. The detailed timing diagram can be found in Fig. 3.2b.

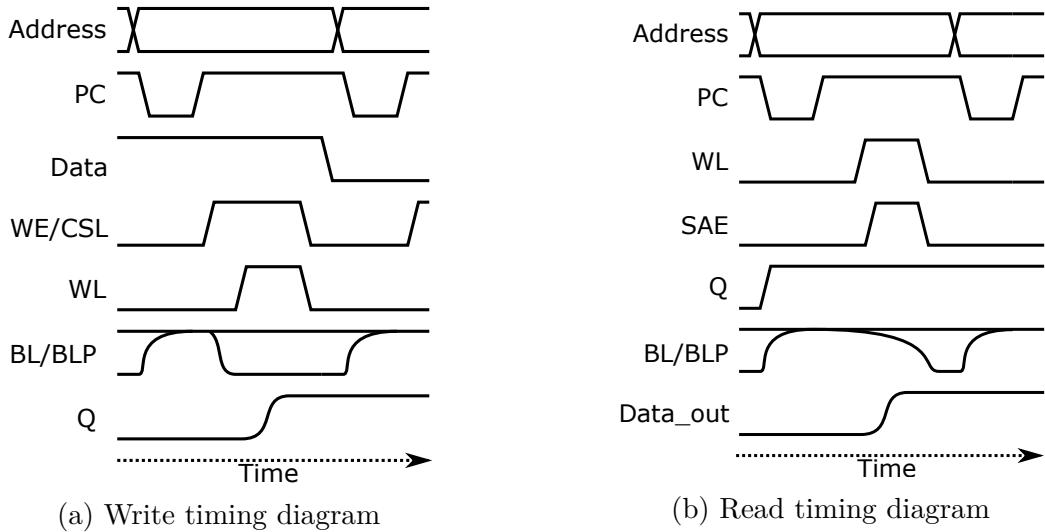


Figure 3.2: Timing diagram

In case of SRAM array, it is connected in three ways to the peripheral circuits: one row decoder and two column multiplexers as shown in Fig. 3.16. One column multiplexer connects the SRAM to the bit line precharge circuit and another column multiplexer connects the SRAM array to the read driver/sense amplifier and the write driver. Depending on the read/write command, the sense amplifier or the wrte driver are selected.

The precharge circuit is used to precharge the bit lines at the begining of the operation. Write driver is used to write data in the memory and sense amplifier is used to read data from the memory. An ‘AND block’ is used for two reasons. It helps generating the word line pulse width needed for the operation and also, it strengthens the word line signals. The control unit communicates with the outside of the system by taking commands and it generates the controlling signal for allowing the peripheral circuits to perfrom sequentially.

## 3.2 Multiplexers and Decoders

Since the memory capacity for this design is defined as 2-Kbit and the word length is 8-bit, it can be divided as  $2^5 * 2^6 = 2^{11} = 2\text{-Kbit}$ . So, the row address width can be determined as  $\log 2^5 = 5$  and the column address width can be calculated as  $\log(\frac{2^6}{2^3}) = 3$ . Hence, the SRAM will have a row decoder of 32 rows and comulmn multiplexer for 64 bit-lines. Additional decoders are necessary to control the multiplexers according to the input address.

### 3.2.1 Column Multiplexer

For the column multiplexer, a heterogeneous tree structure was used which is generally a multi-level tree in which each node contains the same or different number of branches. This structure has been used because it gives the privilege to use variety of multiplexer of different size and has considerably low propagation delay than the binary tree structure. The propagation delay of the heterogenous tree structure can be given by the following equation of Elmore delay model [12]:

$$t_d = \sum_{i=1}^k iS_iRC + \sum_{i=1}^{k-1} iRC \quad (3.1)$$

where,  $k$  denotes the the number of partitioned stages and  $S_i$  denotes the number of switches combined together at Stage  $i$ . In Fig. 3.3, a two-level multiplexer is shown with 64 inputs or 8 words. The propagation delay for this particular structure has been calculated as  $11RC$  ( $R$ : on resistance of the MOSFET and  $C$ : output capacitance of the MOSFET) from equation 3.1. On the other hand, the propagation delay of the binary tree structure is given by the following equation [12]:

$$t_d = \frac{1}{2} \log n(1 + 3 \cdot \log n)RC \quad (3.2)$$

where,  $n$  is the number of multiplexer input. For multiplexing from 8 words, the propagation delay is  $15RC$  calculated from equation 3.2. Heterogenous tree structure has low propagation delay compared to other multiplexer structure, however, it consumes larger area than other multiplexer structure [12].

In Fig. 3.4, the schematic implementation of 2-to-1 word multiplexer is shown where a word has been multiplexed out of 2 words [6]. The 4-to-1 multiplexer needed for the SRAM is also designed in the similar way.

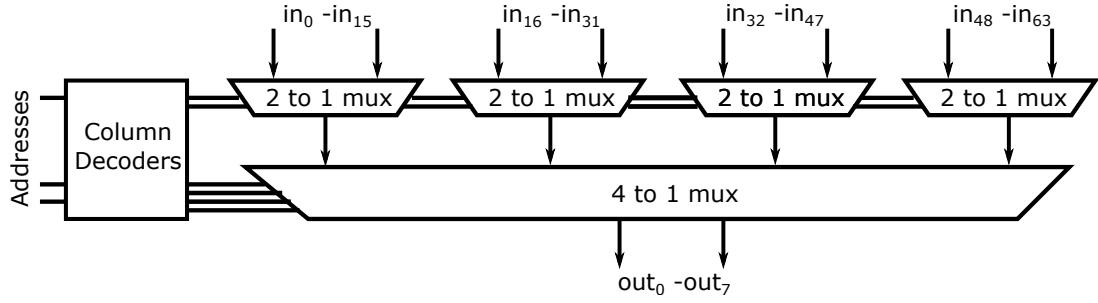


Figure 3.3: Block diagram of the multiplexer

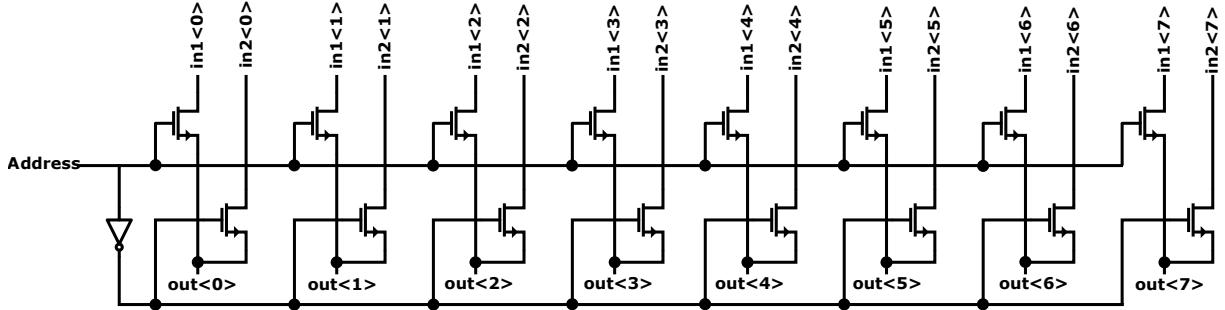


Figure 3.4: Circuit Diagram of 2:1 word multiplexer

### 3.2.2 Row Decoder

As the SRAM array has 32 rows, it would be possible to build a decoder of 5-input logic gates. In that case, 32 logic gates would be needed where each gate would be a 5-input logic gate. Every address has to drive these 32 logic gates every time which would increase the propagation delay of the decoder. That's why a multi-level row decoder has been used by using the cascade of the gates. The advantage is that this structure reduces the area of the decoder and propagation delay [6]. A 3-to-8 decoder and a 2-to-4 decoder have been used as a predecoder and are integrated as final decoder to produce 32 word lines. The coding implementation can be found in appendix A.

## 3.3 Write Driver

Figure 3.5 shows the write driver. The writing operation is generally done by discharging one of the bit lines. Since N-MOSFET has been used for the write driver and it has high electron mobility, discharging time of bit lines would not be an issue. Minimum dimensions of the write driver transistors are used. Figure 3.6 shows the transient waveforms for the writing operation and highlights that the bit-line is discharged from 913 mV to 1.32  $\mu$ V

### 3 SRAM Peripheral Circuits

in less than 10 ns. Given that the required writing speed of the SRAM is at 4 MHz, the precharging strength with minimally sized transistors is accepted.

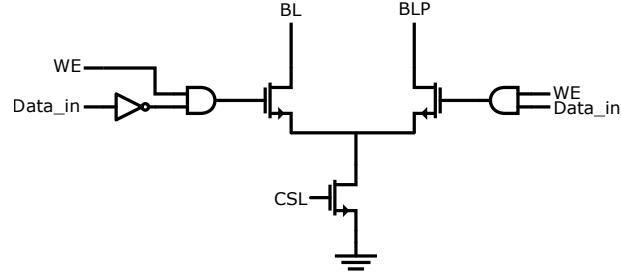


Figure 3.5: Circuit Diagram of Write Driver

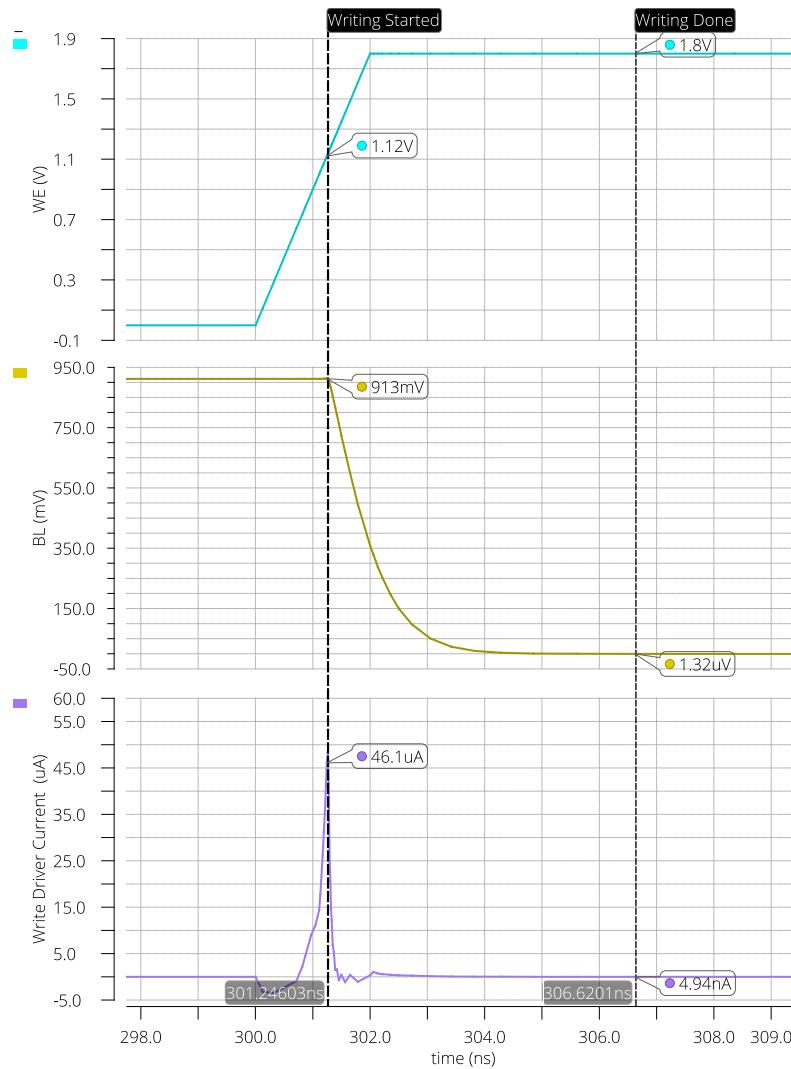


Figure 3.6: Waveform for the write driver

### 3.4 Sense Amplifier

In order to read a word or 8 bits of data, 8 sense amplifiers have been used. Among various configuration listed in [6], latch based sense amplifier, shown in Fig. 3.7, has been used due to being fast and ability to full swing rapidly. This type of sense amplifier uses the regenerative effect of the CMOS inverter to generate the valid voltage levels. Proper activation time of the sense amplifier is necessary to have low voltage operation. The output full swing voltage can be written as the function of bit line capacitance, propagation delay and initial input voltage which needs to be sensed. The equation is given by [13]:

$$\Delta V_{SA} = \Delta V_{SA}(0) \exp^{-\frac{t \cdot g_m}{C_{SA}}} \quad (3.3)$$

$C_{SA}$  is the single bit-line capacitance which consists of source/drain capacitance of the NMOS access transistor, wire capacitance, and source/drain contact capacitance [14]. The value of  $C_{SA}$  is found as around  $90 \text{ fF}$  from the RC extracted view of the SRAM cell. It is possible to calculate the  $g_m$  for the MOSFET from equation 3.3. Initial voltage  $V_{SA}(0) = 200 \text{ mV}$  to full swing  $V_{SA} = 1.8 \text{ V}$  has been considered within sensing time of  $t_{sense} = 10 \text{ ns}$ . After getting the transconductance  $g_m$ , it is possible to calculate the aspect ratio of the MOSFET by following equation:

$$g_m = \left(\frac{W}{L}\right) \cdot k'_n \cdot \left(\frac{V_{DD}}{2} - V_{th}\right) \quad (3.4)$$

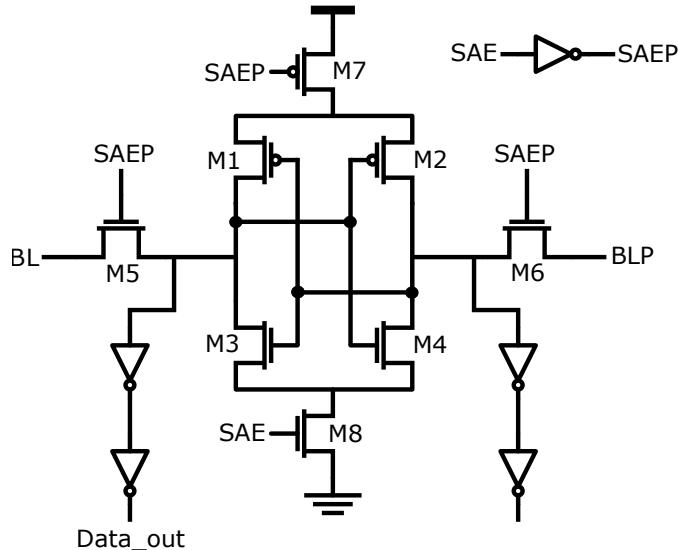


Figure 3.7: Circuit diagram of sense amplifier

### 3.5 Precharge Circuit

Eight precharge circuit have been used to precharge the bit-lines through the column multiplexer. The precharge voltage can be  $V_{DD}$  or  $V_{DD}/2$ , however, the  $V_{DD}/2$  is used because it reduces both the time and effort to precharge the bit lines. The circuit diagram is shown in Fig. 3.8 which is activated by lowering the ‘PC’ signal to ‘0’ logic level. ‘M1’ and ‘M2’ transistors allow to pull-up the bit lines to  $V_{DD}/2$  and ‘M3’ transistor assists equalizing the bit line voltage [6]. Approximately 40 ns is needed to charge the bit lines from ‘0’ to  $V_{DD}/2$  and the current consumption is almost 17  $\mu A$  shown in the waveform in Fig. 3.9. This precharging time window is chosen considering the write operation cycle which is 250 ns.

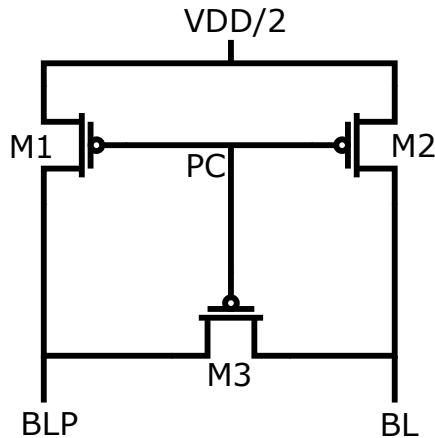


Figure 3.8: Precharge Circuit

### 3.6 Control Unit

Figure 3.10 shows the block diagram of the overall control system of the SRAM. Any read or write operation is triggered by the signal transition detection circuit. ‘Word Line enable and reset generation’ will be activated following by the ‘Precharge Signal’. Depending on the discharging time of the bit lines, this module will generate a ‘Word Line Enable ( $WL\_en$ )’ signal, a ‘reset’ signal and a ‘Sense Amplifier Enable Start ( $SA\_start$ )’ signal. After that, the ‘write enable signal generation’ or the ‘sense amplifier enable generation’ circuit will be activated depending on the command ‘read’ or ‘write’. If ‘write’ command was already given, then the ‘write enable signal generation’ circuit would activate and generate a ‘Write Enable (WE)’ signal. Otherwise, if ‘read’ command was given, the ‘sense amplifier enable generation’ circuit would activate and with the intersection of ‘ $SA\_start$ ’

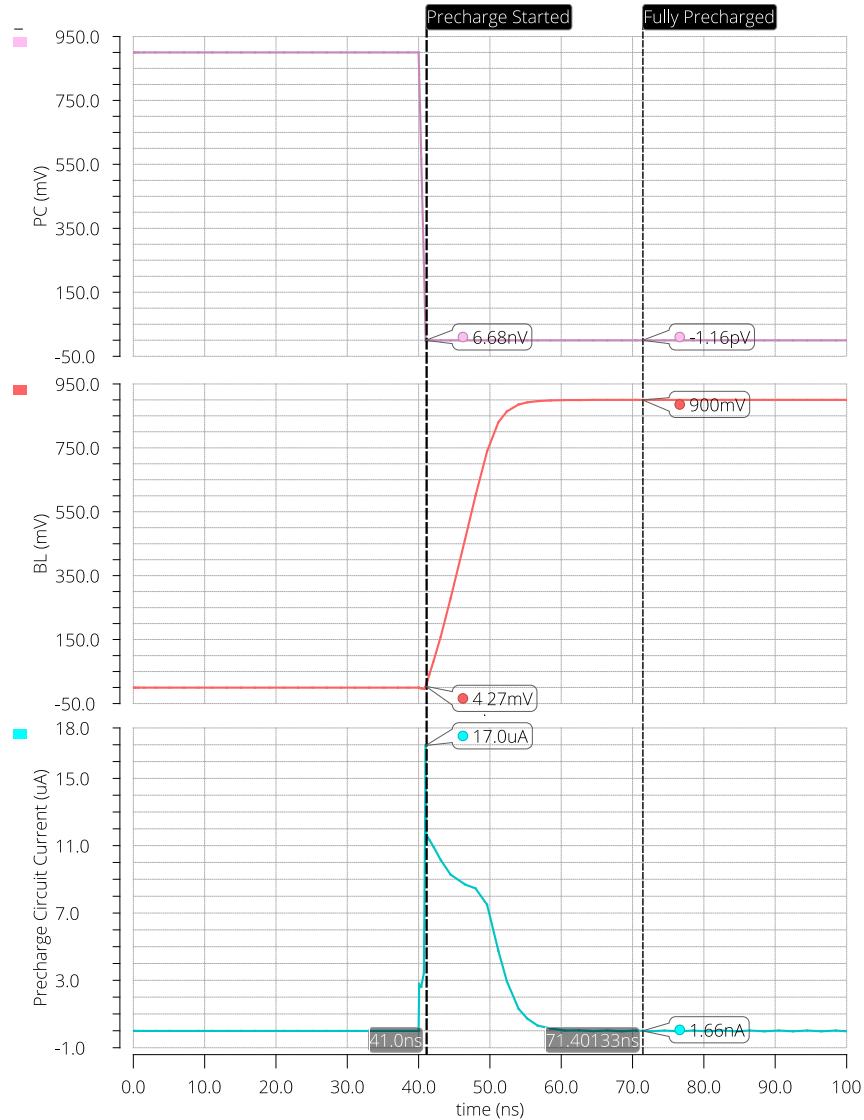


Figure 3.9: Waveform for precharge Circuit

signal, it would generate a ‘Sense Amplifier Enable (SAE)’ signal. In the following section, a brief discussion on each component is given.

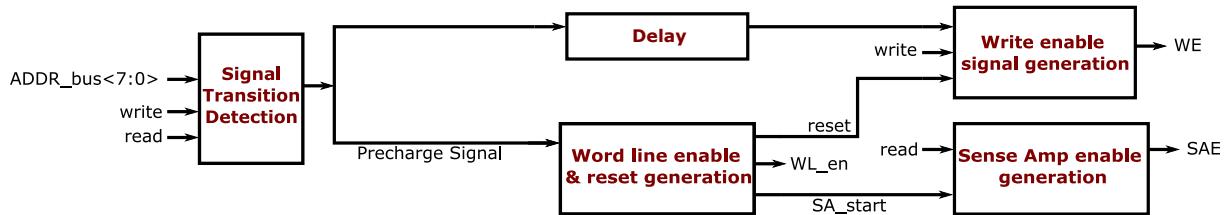


Figure 3.10: Block diagram of control operation

### 3.6.1 Address Transition Detection

In order to trigger the SRAM, an address transition detection circuit was designed with a XOR gate and delay buffers shown in Fig. 3.11. When there will be a transition in the address bus, the circuit will immediately generate a signal and start pre-charging the bit-lines. Furthermore, as a requirement of the design, a ‘read/write signal transition detection’ circuit, shown in Fig. 3.12 was also necessary which was integrated in the ‘precharge signal generation’ circuit.

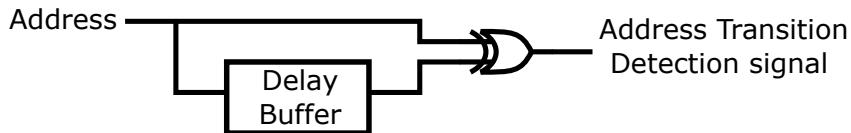


Figure 3.11: Address transition detection circuit

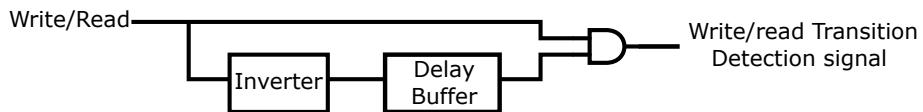


Figure 3.12: Read/write positive transition detection circuit

### 3.6.2 Word Line enable and Reset Signal Generation

Figure 3.13 shows the model of the ‘word line enable signal generation’ circuit. A dummy unit cell is a dummy SRAM bit cell where a logical ‘1’ is already stored in one of the storing node during the precharge period. So, it can be used to get the strating time of the discharging of the bit line capacitance so that a ‘*word line enable (WL\_en)*’ signal can be generated while necessary. Delay buffers have been used to have the pulse width of the ‘*WL\_en*’ signal. The pulse width of this signal or the number of delay buffers have been determined depending on the charging and discharging time of the bit line capacitance for the correct functionality of the SRAM.

‘*SA\_start*’ signal is pulled out between the delay buffers when the voltage difference between the bit lines is at 200 mV, minimum. Finally, an SR-flip flop has been used to generate the ‘*word line enable (WL\_en)*’ signal taking ‘*precharge signal*’ as the clock for the flip-flop. The ‘*Reset*’ signal for this flip-flop is also used for other modules of the control unit.

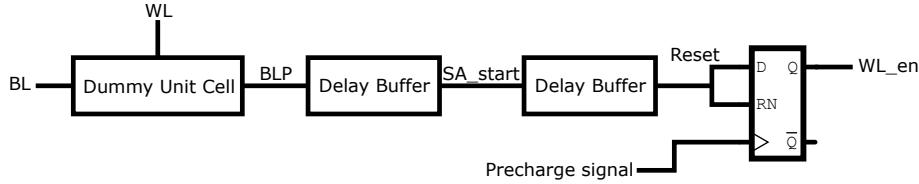


Figure 3.13: Word Line Enable and Reset Signal Generation circuit

### 3.6.3 Write enable Signal Generation

Write enable signal generation circuit only consists of an SR flip-flop shown in Fig. 3.14. ‘write’ command from outside of the system is taken as the input ‘D’ of the flip-flop and the ‘Reset’ is produced by the ‘word line enable generation circuit’ described in section 3.6.2.

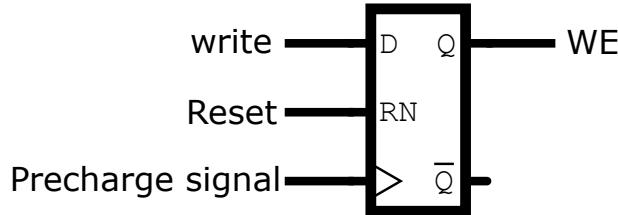


Figure 3.14: Write Enable Signal Generation circuit

### 3.6.4 Read enable Signal Generation

Read enable signal generation circuit consists of an SR-flip flop as well as a delay buffer shown in Fig. 3.15. Delay buffer is used to create the pulse width of the ‘Sense Amplifier Enable(SAE)’ signal for 10 ns. ‘SA\_start’ signal denotes the starting time of activation of the sense amplifier generated by the ‘word line enable generation circuit’ described in section 3.6.2.

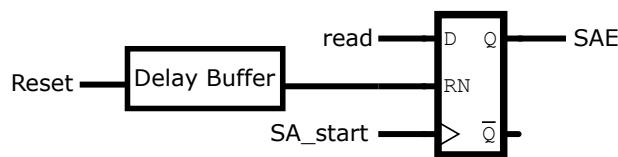


Figure 3.15: Read Enable Signal Generation circuit

## 3.7 SRAM Floorplan

Figure 3.16 shows the imprecise idea of the SRAM floorplan. The SRAM has the dimension of approximately 500 um x 355 um with the peripherals.

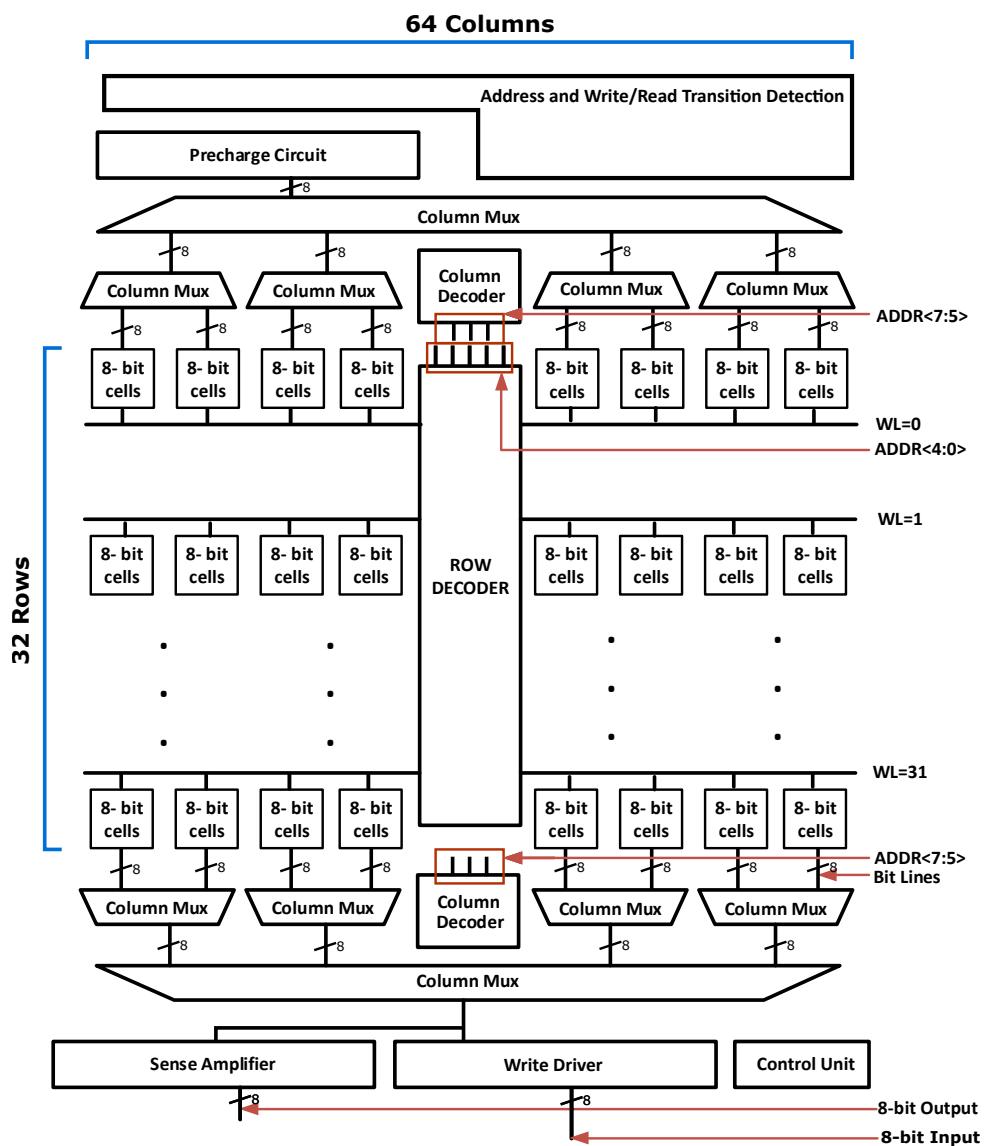


Figure 3.16: SRAM floorplan

## 3.8 SRAM Layout

Figure 3.17 shows the final layout of the total SRAM system indicating different module's placement.

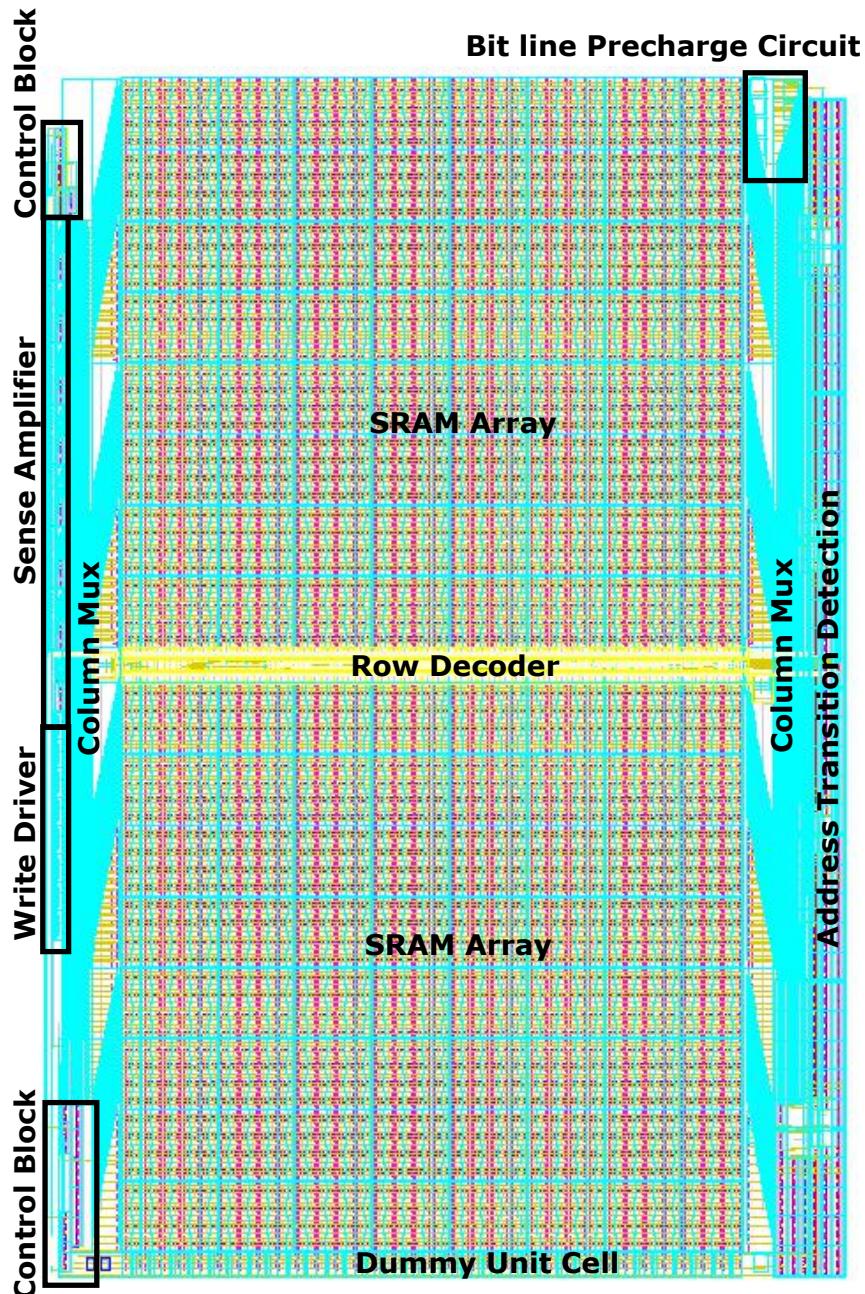


Figure 3.17: Layout of total SRAM system

# 4 Verification and Simulation Result

This chapter describes all the simulation results of the SRAM, verification of the SRAM and finally, the verification for the SPI interface for the SRAM.

## 4.1 Control Unit Simulation

Figure 4.1 shows the test-bench for the control unit. Control unit is initiated by generating a precharge signal from the Address Transition Detection (ATD) circuit. The control unit signal waveforms are shown in Fig. 4.2. For this simulation, ‘*write=1*’ means writing down in the SRAM otherwise, reading from the memory. An SRAM unit cell has been used for the testing. When ‘*write=1*’, ‘WE’ pulse is generated and when ‘*write=0*’, ‘SAE’ signal is generated. ‘Reset’ signal has been created to reset ‘WE’ signal. Output data can be obtained when ‘SAE’ signal is enabled.

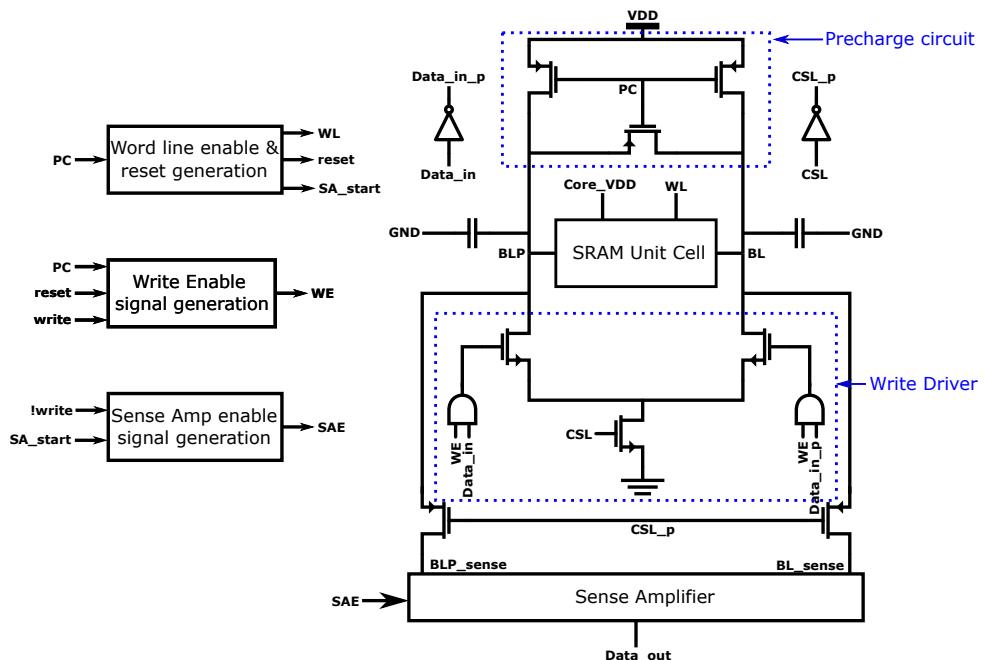


Figure 4.1: Test-bench for control unit

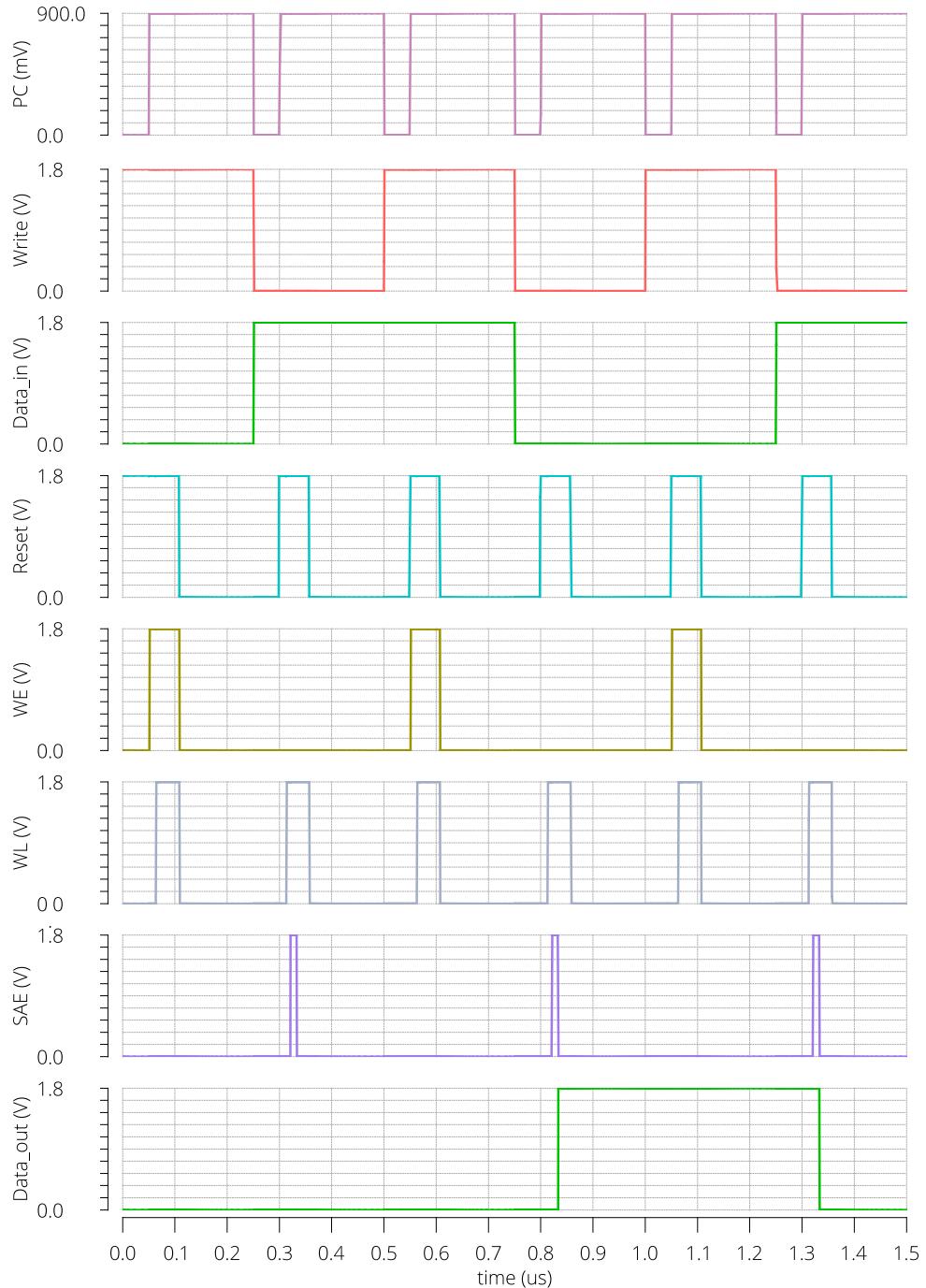


Figure 4.2: waveform of the control unit

## 4.2 SRAM Unit Cell Simulation

Figure 4.5 shows the waveform generated by the testbench (see Fig. 3.1) for the unit cell. As expected, the leakage current of the unit cell has been found approximately  $87 \text{ fA}$ .

#### 4 Verification and Simulation Result

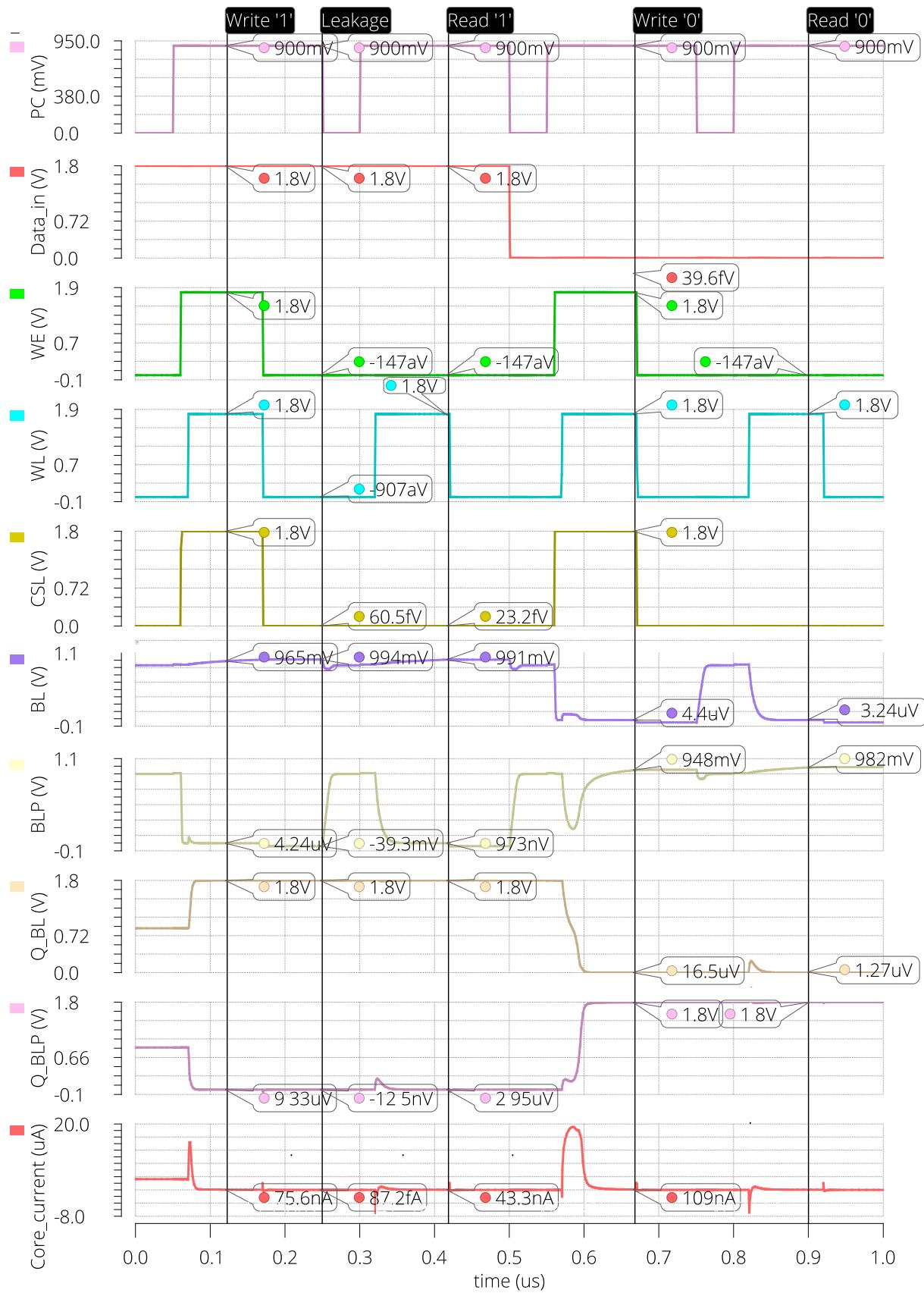


Figure 4.3: Waveform of the verification of the SRAM

### 4.3 Stand-by Current Simulation

The leakage current of the full SRAM array was found out from the full SRAM with the peripherals. A transient simulation for 1 ms was performed shown in Fig. 4.4. Few read/write operation was carried out for first few micro seconds. The stand-by current was recorded in the rest of the non-functioning period which is approximately  $231 \text{ pA}$ . The read cycle consumes approximately  $500 \mu\text{W}$  averaged over  $1000 \text{ ns}$  and the write cycle consumes approximately  $650 \mu\text{W}$  averaged over  $250 \text{ ns}$

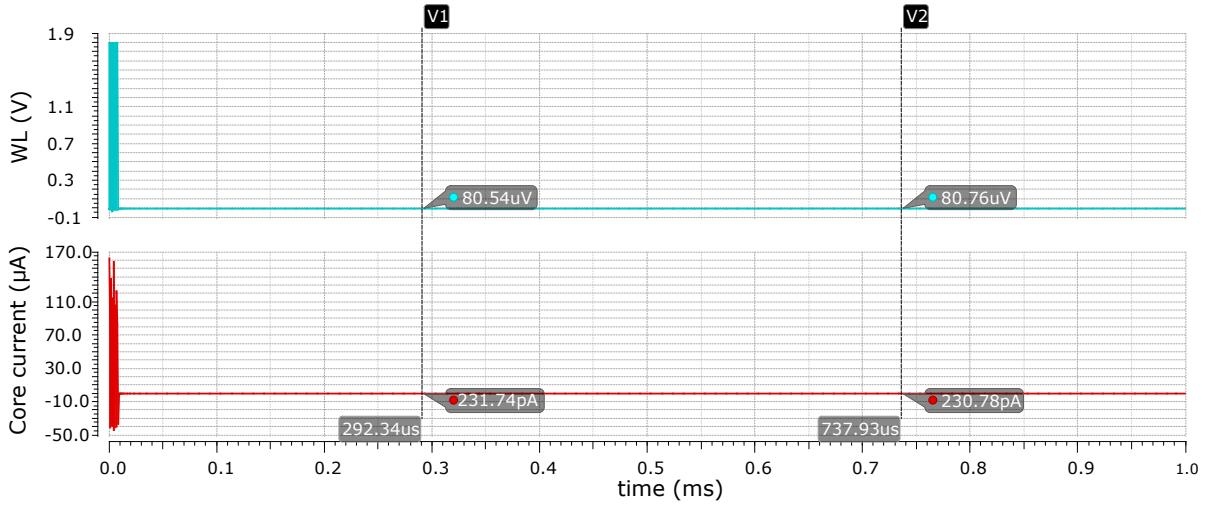


Figure 4.4: SRAM Core leakage current

### 4.4 Verification of the SRAM

Table 4.1 shows the snippet of the testing plan for the full SRAM system and Fig. 4.5 shows the waveform of execution of SRAM operation corresponding to table 4.1. The data which was written, was given by the ‘*Data in<0>*’ and ‘*Data in<7>*’. The row ‘*Time*’ gives the input timing of the address, data and commands in the Fig. 4.5. In the table 4.1, ‘*W*’ denotes the write and ‘*R*’ denotes the read operation. The data were written in various addresses, overwritten in the same address in different times and read out in different times. The expected output row in the table 4.1 shows the data to be expected in the specific time and it was matched in the waveform in Fig. 4.5. The simulation was executed with the RC extracted view of the full SRAM system.

#### 4 Verification and Simulation Result

Table 4.1: Data for the verification of the SRAM

Data_in[0]	0				1				0				1				0				1			
Data_in[7]	1				0				1				0				1				0			
Time(μs)	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2	2.25	2.5	2.75	3	3.25	3.5	3.75	4	4.25	4.5	4.75	5	5.25	5.5	
Addresses/Command	W	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	W	R	
ADDR[7:0]	20	00	20	20	00	C0	A0	BF	C0	1F	BF	3F	1F	20	3F	00	20	32	00	12	32	16	04	
Expected Output																								
Data_out[0]			0		0		0		1		1		0		0		1		1		0		0	
Data_out[7]			1		1		1		0		0		1		1		0		0		1		1	

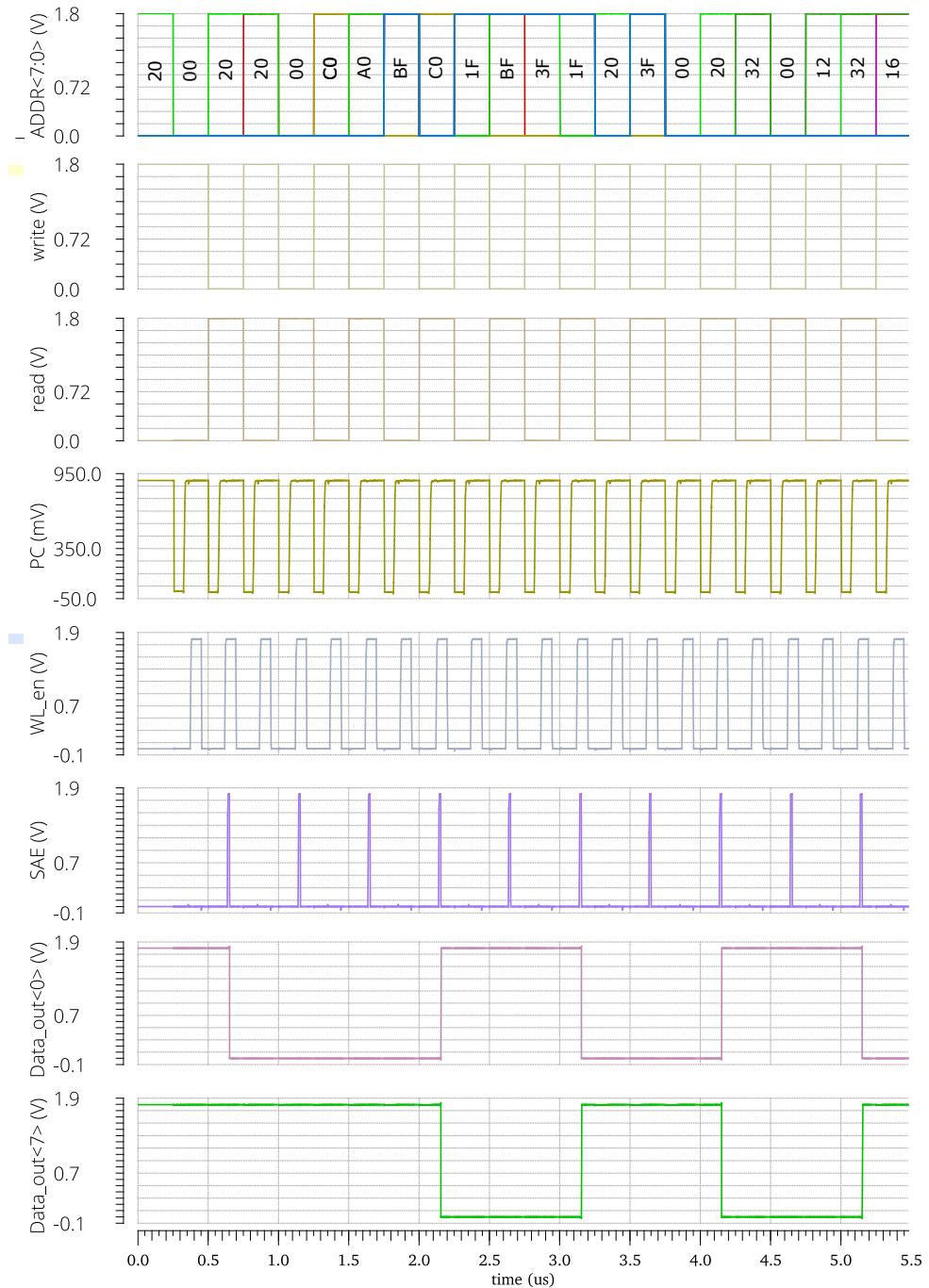


Figure 4.5: Waveform of the verification of the SRAM

## 4.5 SPI Interface

Three Serial Peripheral Interface (SPI) have been used for communication of the SRAM system to minimize the number of access pads for the SRAM test structure. One for data incoming and outgoing and the other two have been used for write address and read address communication. All of the SPI interfaces used are the slave SPI interface. Figure 4.6 and 4.7 demonstrate the operation of the serial-parallel data conversion and vice-versa. The graph data were exported from Cadence and processed in MATLAB. The mode of SPI operation depends on the master SPI interface.

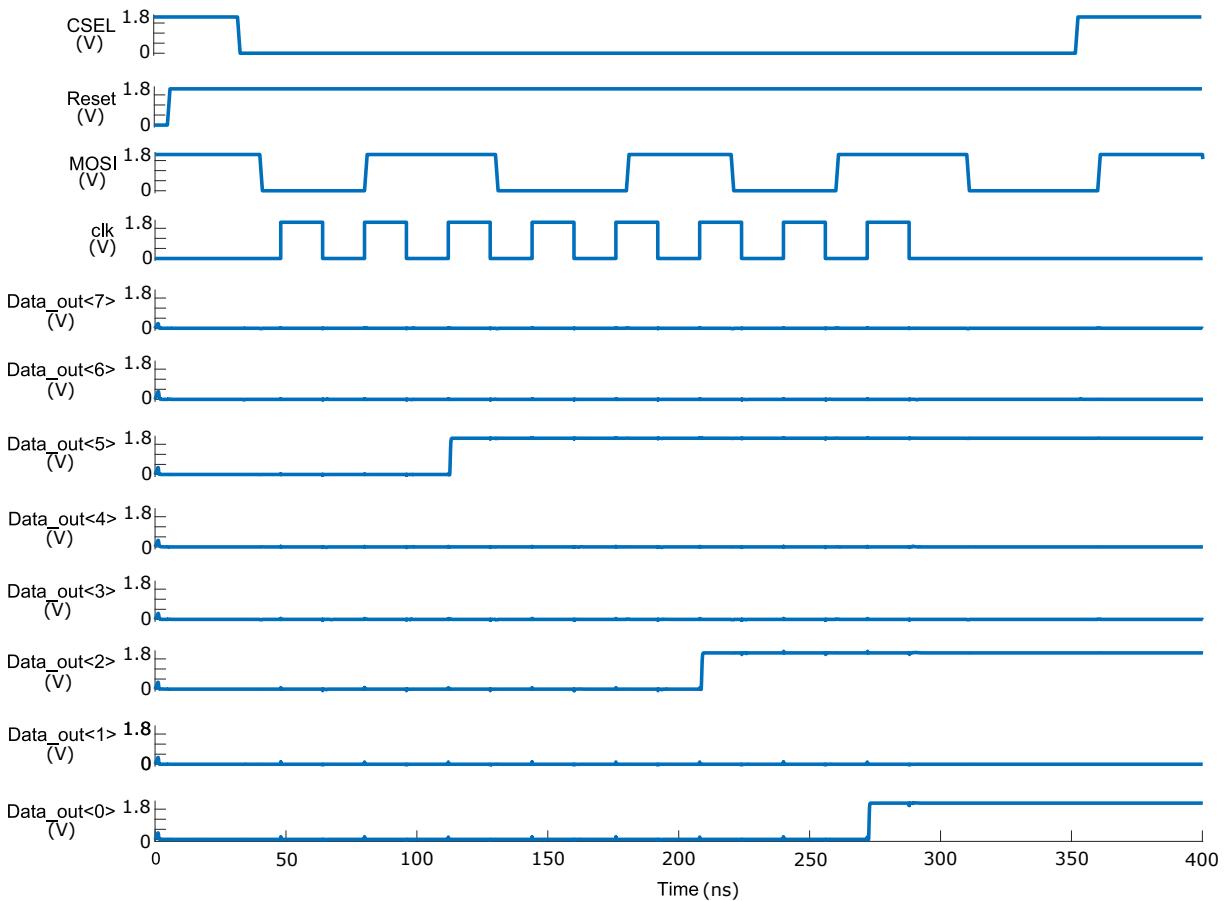


Figure 4.6: Operation of MOSI in SPI

Figure 4.8 shows the waveform of SRAM verification after including the SPI interface with the SRAM. A string of data has been passed through the MOSI and stored in the particular address of the SRAM. The data were read later by the MISO. Figure 4.9 shows the closer view of Fig. 4.8 from  $0.7\mu s$  to  $0.9\mu s$ . Reading and writing operation is executed together spontaneously in the same clock cycle. In case of the address SPI interface, an extra latch has been used in the slave design to make the address bits available together

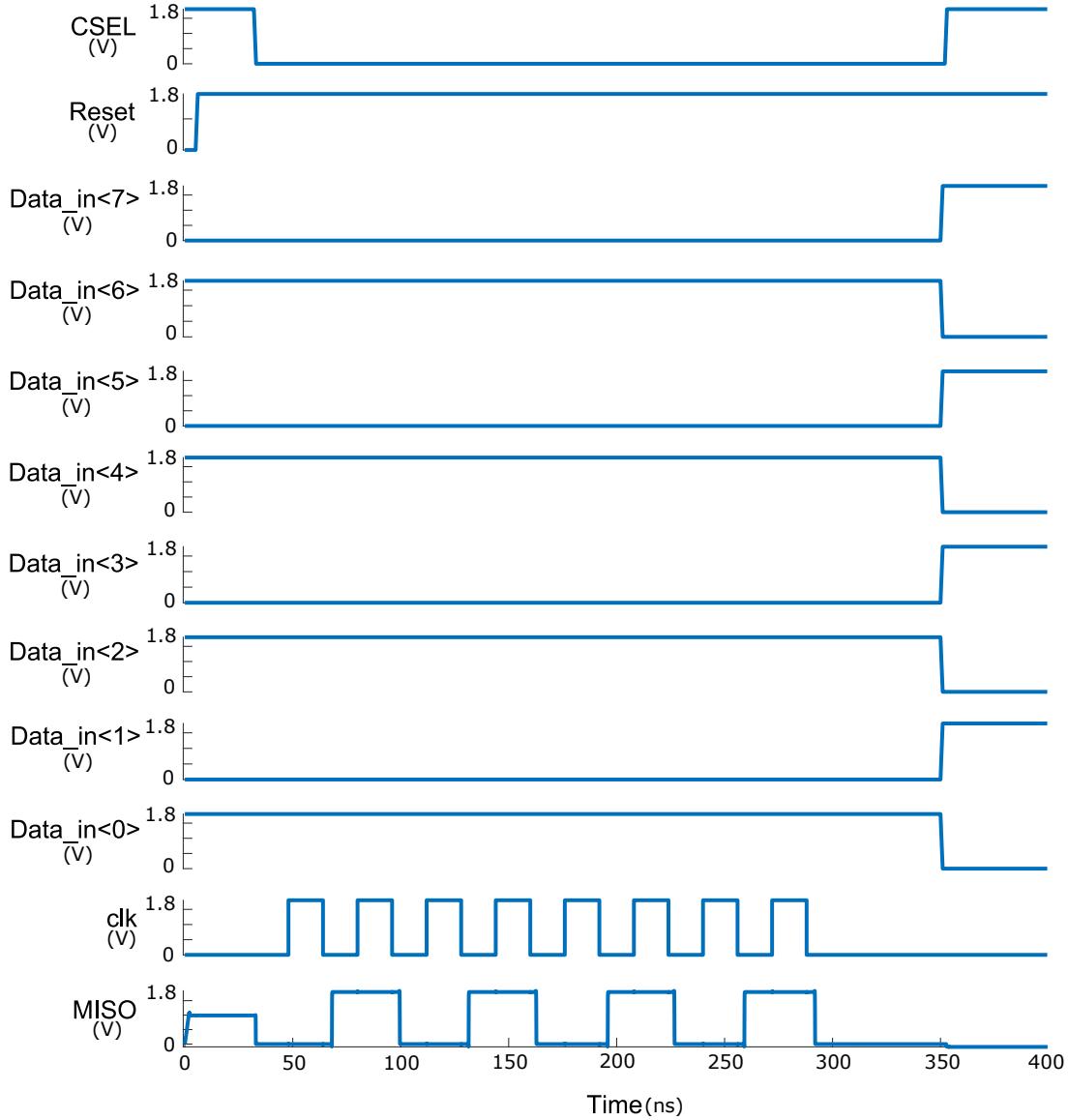


Figure 4.7: Operation of MISO in SPI

in the MOSI so that Address Transition Detection (ATD) works properly. ‘*CLK\_Data*’ and ‘*CSEL\_Data*’ represent the clock sent by the master SPI and chip select/slave select respectively. The simulation results associated with the SPI interface were generated from the RC extracted view of the SRAM.

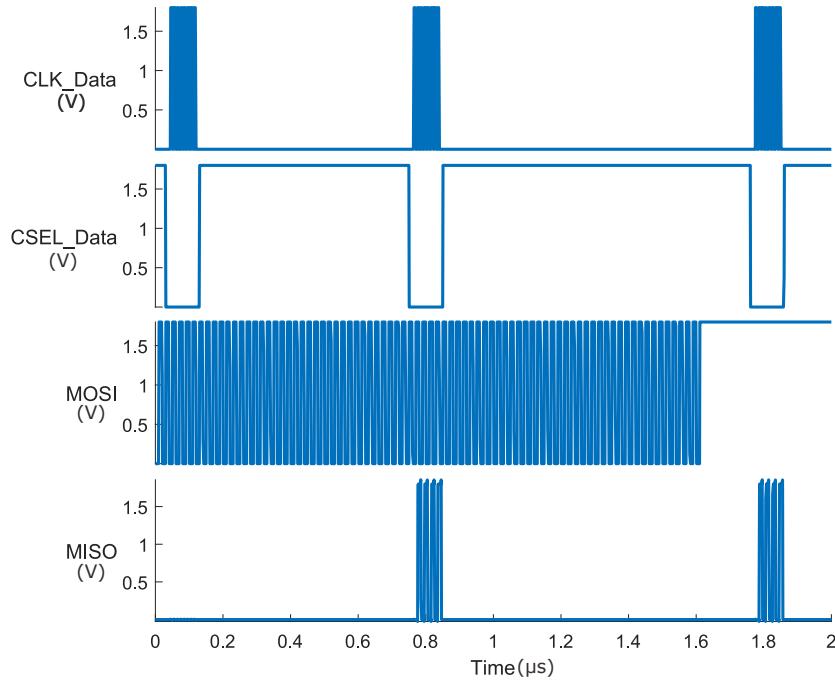


Figure 4.8: SPI waveform for SRAM

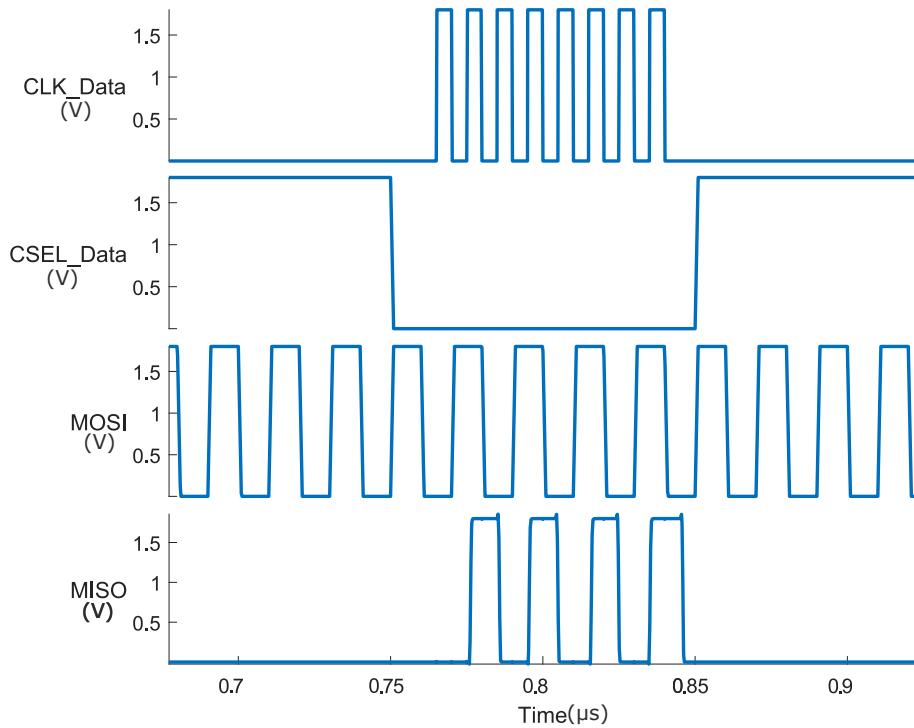


Figure 4.9: Expanded SPI waveform for SRAM

## 5 Conclusion

The first phase of the SRAM design was comprised of the study of the leakage current in the MOSFET and the formation of a unit cell considering the SRAM parameters described in chapter 2. The last phase of the SRAM design was about all the peripheral circuits and the integration of the circuits with the SRAM array described in chapter 3.

As a purpose of medical implant, apart from low power operation, hardware security can be further topic of discussion for the SRAM. The SRAM-based Physical Unclonable Function(PUF) can be used to generate and store a secret key [1]. Hence, it might be reasonable to find out the mismatches in the generated bits of the SRAM due to process variability.

# Bibliography

- [1] M. R. Faragalla, M. A. Ewais, H. F. Ragai, M. S. Badran, and H. H. Issa, “Impact of process variability on FinFET 6T SRAM cells for physical unclonable functions (PUFs),” in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 2017, pp. 31–36.
- [2] E. Grossar, M. Stucchi, K. Maex, and W. Dehaene, “Read stability and write-ability analysis of SRAM cells for nanometer technologies,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 11, pp. 2577–2588, 2006.
- [3] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.
- [4] T. H. N. Yuan Taur, *Fundamentals of Modern VLSI Devies*. Cambridge University Press, 2009.
- [5] V. L. Sangeeta Singh, “Read and write stability of 6T SRAM,” *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 3, 2014.
- [6] M.-B. Lin, *Introduction to VLSI Systems*. CRC Press, 2012.
- [7] C.-H. Hua, T.-S. Cheng, and W. Hwang, “Distributed data-retention power gating techniques for column and row co-controlled embedded SRAM,” in *2005 IEEE International Workshop on Memory Technology, Design, and Testing (MTDT’05)*, 2005, pp. 129–134.
- [8] K.-S. Min, K. Kanda, and T. Sakurai, “Row-by-row dynamic source-line voltage control (RRDSV) scheme for two orders of magnitude leakage current reduction of sub-1-V-V/sub DD/ SRAM’s,” in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED ’03.*, 2003, pp. 66–71.

## Bibliography

---

- [9] M. Zabeli, N. Caka, M. Limani, and Q. Kabashi, “The impact of MOSFET’s physical parameters on its threshold voltage,” in *Proceedings of the 6th Conference on Microelectronics, Nanoelectronics, Optoelectronics*, ser. MINO’07. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007, p. 54–58.
- [10] C. Duari and S. Birla, “Low leakage SRAM cell with improved stability for IoT applications,” *Procedia Computer Science*, vol. 171, pp. 1469–1478, 2020, third International Conference on Computing and Network Communications (CoCoNet’19). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050920311364>
- [11] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.
- [12] M.-B. Lin, “On the design of fast large fan-in CMOS multiplexers,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 8, pp. 963–967, 2000.
- [13] K. Abbas, *Handbook of Digital CMOS Technology, Circuits, and Systems*. Springer, 2020.
- [14] A. Karandikar and K. Parhi, “Low power SRAM design using hierarchical divided bit-line approach,” in *Proceedings International Conference on Computer Design. VLSI in Computers and Processors (Cat. No.98CB36273)*, 1998, pp. 82–88.

# A Appendix

2 to 4 decoder (predecoder):

```
1 module row_2_to_4_dec(
2   input [1:0] ADDR,
3   output reg [3:0] row);
4
5   always @(ADDR)
6   begin
7     case (ADDR)
8       2'b00 : row <= 4'b0001;
9       2'b01 : row <= 4'b0010;
10      2'b10 : row <= 4'b0100;
11      2'b11 : row <= 4'b1000;
12    endcase
13  end
14 endmodule
```

3 to 8 decoder (predecoder):

```
1 module row_3_to_8_dec(
2   input [2:0] ADDR,
3   output reg [7:0] row );
4
5   always @(ADDR)
6   begin
7     case (ADDR)
8       3'b000 : row <= 8'b00000001;
9       3'b001 : row <= 8'b00000010;
10      3'b010 : row <= 8'b00000100;
11      3'b011 : row <= 8'b00001000;
12      3'b100 : row <= 8'b00010000;
13      3'b101 : row <= 8'b00100000;
```

```
14      3'b110 : row <= 8'b01000000;
15      3'b111 : row <= 8'b10000000;
16      endcase
17  end
18 endmodule
```

Final decoder:

```
1 module row_dec(ADDR, row);
2   input [4:0] ADDR;
3   output [31:0] row;
4   wire [7:0] x;
5   wire [3:0] y;
6
7   row_3_to_8_dec x1(ADDR[4:2],x[7:0]);
8   row_2_to_4_dec x2(ADDR[1:0],y[3:0]);
9
10  genvar i;
11  generate
12    for(i=0;i<8;i=i+1)
13
14    begin
15      assign row[4*i]=x[i]&y[0];
16      assign row[(4*i)+1]=x[i]&y[1];
17      assign row[(4*i)+2]=x[i]&y[2];
18      assign row[(4*i)+3]=x[i]&y[3];
19    end
20  endgenerate
21 endmodule
```

# B Appendix

SPI Slave Interface:

```
1 module digi_test_interface(RST, SCLK, CSEL, MISO, MOSI, DATA_IN_P, ↵
2   DATA_OUT_P);
3
4 parameter BIT_COUNT = 8;
5 input SCLK, RST;
6 input CSEL;
7 input MOSI;
8 input [BIT_COUNT-1:0] DATA_IN_P;
9 output [BIT_COUNT-1:0] DATA_OUT_P;
10 reg [0:BIT_COUNT-1] in_buff;
11 output MISO;
12
13 reg [0:BIT_COUNT-1] shift_load_reg;
14 reg [2:0] count;
15 genvar i;
16
17 always @(negedge RST or negedge CSEL) begin
18   if(!RST) shift_load_reg<=0;
19   else shift_load_reg<=DATA_IN_P;
20 end
21
22 always @(negedge SCLK or posedge CSEL or negedge RST) begin
23   if(CSEL==1||RST==0) count<=0;
24   else count<=count+1;
25 end
26 assign MISO = (CSEL)? 1'bz: shift_load_reg[count];
27
28
```

```
29 always @(posedge SCLK or negedge RST) begin
30   if(!RST) in_buff<=0;
31   else begin
32     if(!CSEL) in_buff[count]<=MOSI;
33   end
34 end
35
36 for (i=0; i < BIT_COUNT; i=i+1) begin
37   assign DATA_OUT_P[BIT_COUNT-1-i]=in_buff[i];
38 end
39
40 endmodule
```