Chapter 1
CMOS Digital Design

The demand for electronic and multimedia devices is increasing exponentially. This demand in-turn has propelled the need for memory chips to process instructions, store data and other multimedia content. Some of the most common memory structures used for faster data and program memory access are Static (SRAM) and Dynamic (DRAM) memory.

In this chapter, a 6 Transistor CMOS based SRAM memory chip of 1 KB capacity is designed and simulated. The complete chip along with SRAM cells array and circuit elements is designed using SPICE program. Simulations for the design are done using LTspice. Schematic and Layout for a single SRAM cell is also designed using Cadence Schematic and Virtuoso tool respectively. An estimation of parasitic resistance and capacitance values for the layout drawn for the SRAM cell is extracted Vituoso.

The prerequisite to approach this chapter would be an adequate background of CMOS digital circuits, Spice programming and basic knowledge of IC layout design.

1.1 Design of CMOS SRAM Cell and Array

1.1.1 Plan of SRAM Cell and Array

Static Random Access Memory (SRAM) is a type of semiconductor memory. The word static indicates that the memory retains its contents as long as power remains applied. 'Random Access' means that the location in the memory can be written to or read from in any order regardless of the memory location that was last accessed [1]. The SRAM cell has the capability to store one bit data as long as the power is continuously applied. Hence SRAM's are volatile memory devices. An array of eight SRAM cells can store 1 byte of data. Considering this unit of 8-bit SRAM array, a number of these structures can be replicated to build a large memory block. In this chapter, an SRAM memory of 1 KB is designed using 6 Transistor (6T) CMOS SRAM cell.

V.A. Chandrasetty, *VLSI Design: A Practical Guide for FPGA and ASIC Implementations*, SpringerBriefs in Electrical and Computer Engineering,
DOI 10.1007/978-1-4614-1120-8_1, © Springer Science+Business Media, LLC 2011

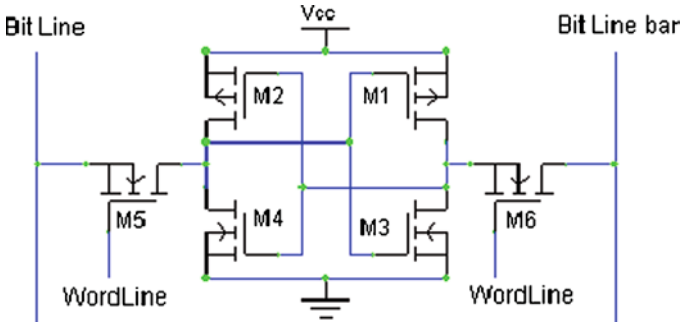

Fig. 1.1 6T CMOS SRAM cell

1.1.2 Design of 6 Transistor SRAM Cell

An SRAM cell can store one bit data on four transistors that form two cross-coupled inverters [1]. This storage cell has two stable states which are used to denote 0 and 1. Two additional access transistors serve to control the access to storage cell during read and write operations. Thus, a combination of 6 transistors is used to store one bit data.

Access to the cell is enabled by the word line (WL) which controls the two access transistors. They are used to transfer data for both read and write operations by connecting the bit lines (BL and BL bar). Although the two bit lines are not necessary, both the signal and its inverse are typically provided since it improves noise margins. The symmetric structure of SRAM's allows for differential signaling, which makes small voltage swings more easily detectable. A schematic of 6T CMOS SRAM cell is shown in Fig. 1.1.

1.1.3 Simulations of SRAM Cell

An SRAM cell has three different states of operation: *standby* when the circuit is idle, *reading* when the data has been requested and *writing* when updating the contents. Each states are discussed with respect to the Fig. 1.1 as follows [2]:

***Standby*:**
If the word line is not asserted, the access transistors M5 and M6 disconnect the cell from bit lines. The two cross coupled inverters formed by M1–M4 will continue to reinforce each other as long as they are disconnected from the outside world.

***Reading*:**
Assuming that the content of memory is 1, when the word line is asserted, the state stored in the cell is transferred to the bit line which is then read on the data output port. If the memory content was a 0, the opposite would happen.

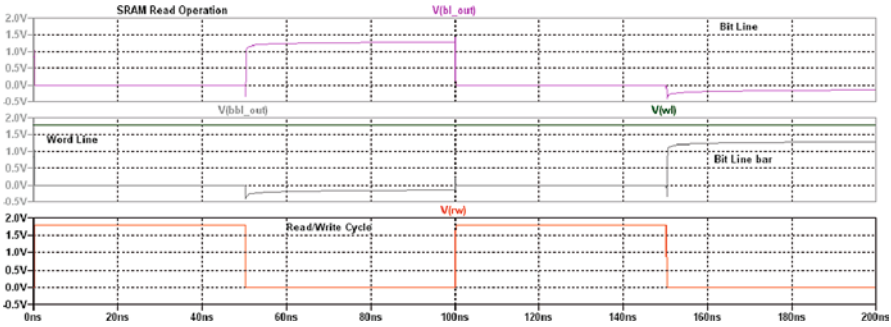

Fig. 1.2 Spice simulations for SRAM cell read operation

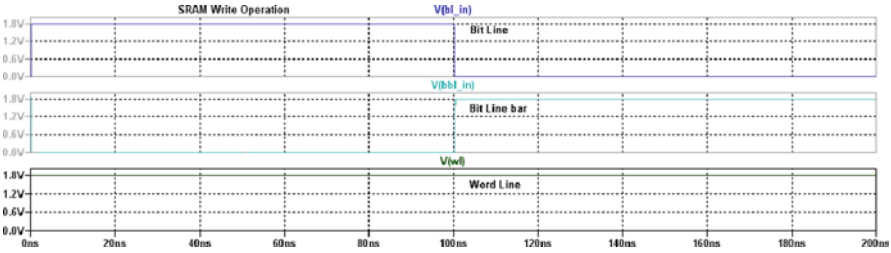

Fig. 1.3 Spice simulations for SRAM cell write operation

***Writing*:**
The start of a write cycle begins by applying the value to be written to the bit lines. The word line is asserted to store the input data on to the cell. The bit line input drivers are designed to be much stronger than the relatively weak transistors in the cell itself, so that they can easily override the previous state of the cross-coupled inverters. Careful sizing of the transistors in the SRAM cell is needed to ensure proper operation.

The spice simulation for SRAM cell Read operation is shown in Fig. 1.2.

The spice simulation for SRAM cell Write operation is shown in Fig. 1.3.

1.1.4 Layout of SRAM Cell

The layout for SRAM cell is drawn using Cadence Virtuoso for 180 nm technology. The layout is successfully completed with Design Rule Checks (DRC) and Layout versus Schematic (LVS) evaluation as well. A snapshot of the layout of SRAM cell is shown in Fig. 1.4. The resistance and capacitance parasitic parameters are extracted from the layout using Cadence Virtuoso.

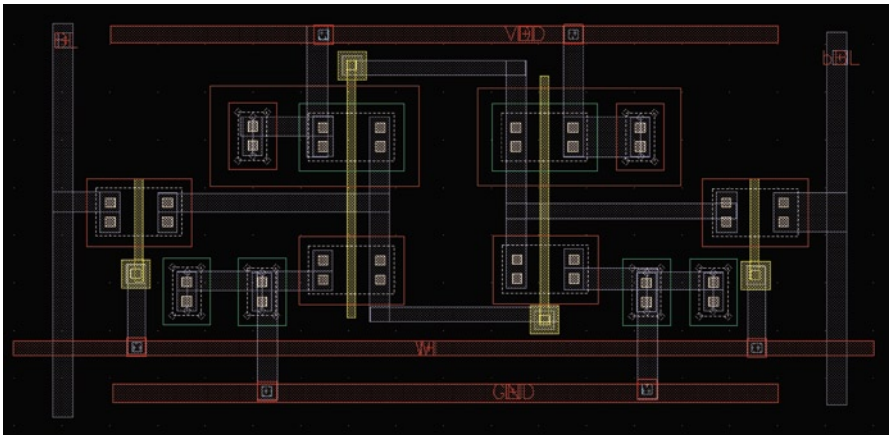

Fig. 1.4 Layout of 6T SRAM cell

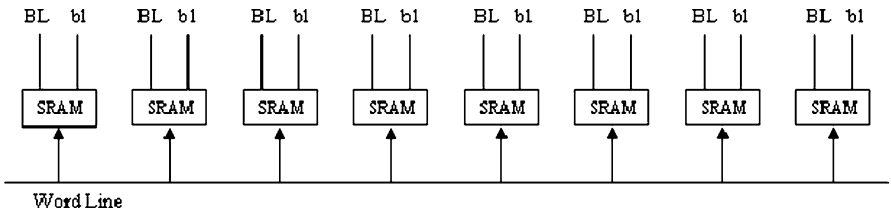

Fig. 1.5 Block diagram of 8-bit SRAM array

1.1.5 Design of SRAM Array

For this design, 1 KB SRAM chip with 8-bit data I/O is required. Since one bit data can be stored in a single SRAM cell, an array of 8 cells should satisfy the requirement. Hence 1,024 such arrays are required to build 1 KB memory chip. A block diagram representing an 8-bit memory SRAM array is shown in Fig. 1.5.

1.1.6 Simulation of SRAM Array

An SRAM array (8-bit) is selected or activated by the row and column decoder based on the input address. The spice simulation for SRAM array section is shown in Fig. 1.6.

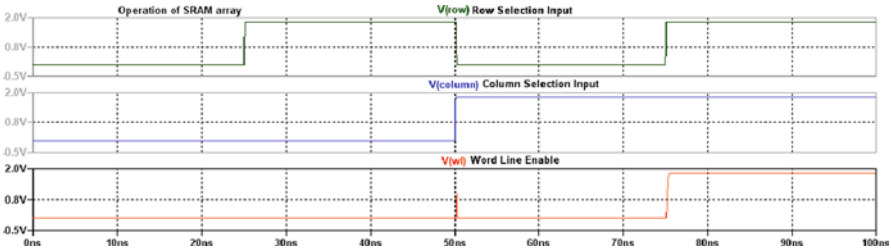

Fig. 1.6 Spice simulations for section of SRAM array

1.2 Design of SRAM Chip Circuit Elements

1.2.1 SRAM Chip Circuit Elements

The 6T CMOS SRAM chip requires various circuit elements to execute the desired memory operations. In this section, a complete SRAM chip circuitry elements such as, address decoder, sense amplifier, pre-charge circuit and data I/O control logic is designed using LTspice. The detailed design, schematic and simulation of these circuit elements are discussed in the following sections:

1.2.1.1 Address Decoder

The Address Decoder is nothing but a simple logic circuitry used to select and enable the memory cells in the SRAM array corresponding to the input address value. In this section, 1 KB SRAM is required to be designed. Hence it requires 10 address bits to cover entire 1 KB memory area. A 5:32 NAND based decoder is designed as row decoder to access 32 bytes of memory area and another 5:32 decoder is used as column decoder in order to access 32 such 32 bytes of memory areas. There by achieving the desired access to 1 KB memory. The schematic of NAND based 5:32 decoder is shown in Fig. 1.7.

The 5:32 NAND based decoder is designed and simulated using LTspice [3]. The simulation results for 5:32 NAND based decoder are shown in Fig. 1.8.

1.2.1.2 Sense Amplifier

A Sense amplifier is an essential circuit in memory chips to speed up the *Read* operation. Due to large arrays of SRAM cells, the resulting signal in the event of *Read* operation has a much lower voltage swing [4]. To compensate for that swing, a sense amplifier is used to amplify voltage coming off Bit Line and ~Bit Line. The voltage coming out of sense amplifier has a full swing voltage of (0–1.8 V). Sense Amplifier also helps reduce the delay times and power dissipation in the overall

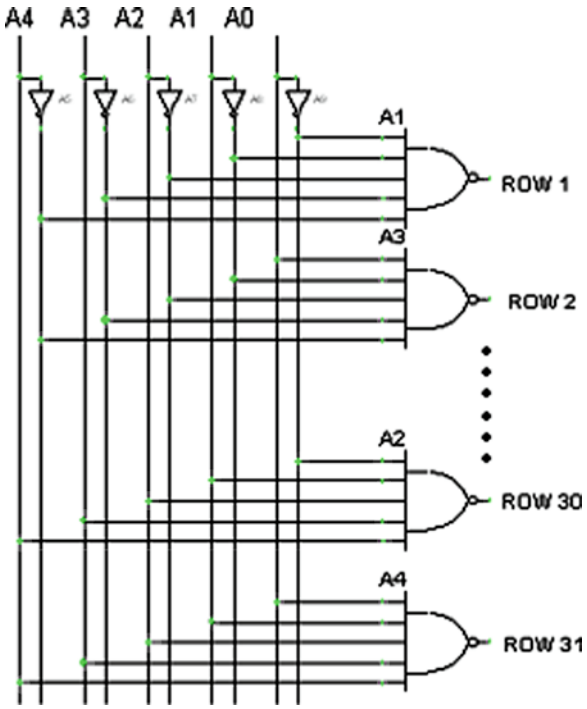

Fig. 1.7 Schematic of NAND based 5:32 row/column decoder

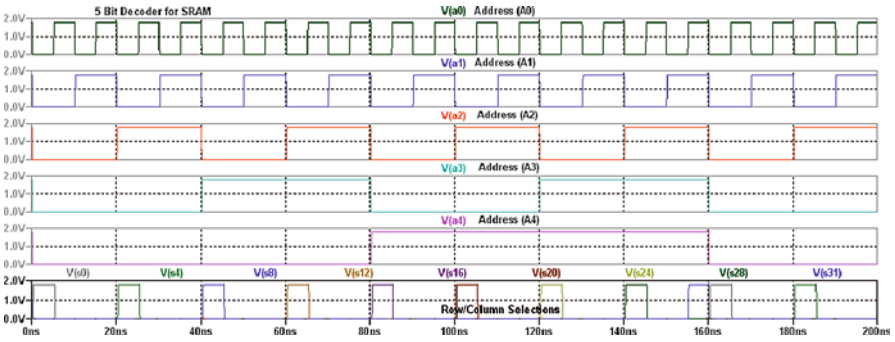

Fig. 1.8 Spice simulations of 5:32 NAND based row/column decoder

SRAM chip. There are many versions of sense amplifier used in memory chips. The one that is designed in this chapter is a Cross-coupled Sense Amplifier. The schematic of the same is shown in Fig. 1.9.

The Cross-coupled/Feedback Sense amplifier is designed and simulated using LTspice. The LTspice simulations for the same are shown in Fig. 1.10.

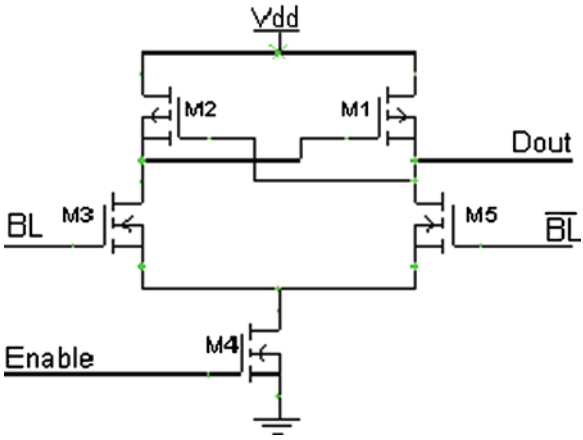

Fig. 1.9 Schematic of cross-coupled sense amplifier

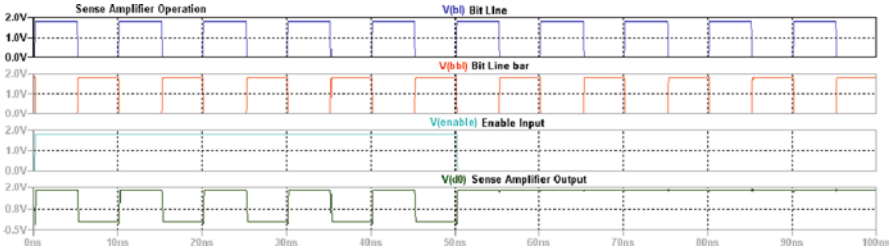

Fig. 1.10 Spice simulations of cross-coupled sense amplifier

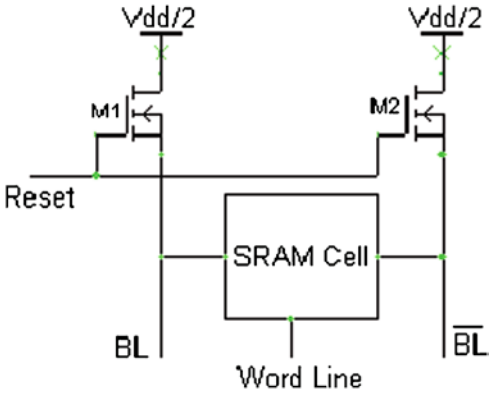

Fig. 1.11 Schematic of a pre-charge circuit

1.2.1.3 Pre-Charge Circuit

Safe read and write operations require a modification of the memory array and timing sequence based on a pre-charge circuit [5]. The schematic of a pre-charge circuit is shown in Fig. 1.11. The usual voltage of pre-charge is VDD/2. Before reading or

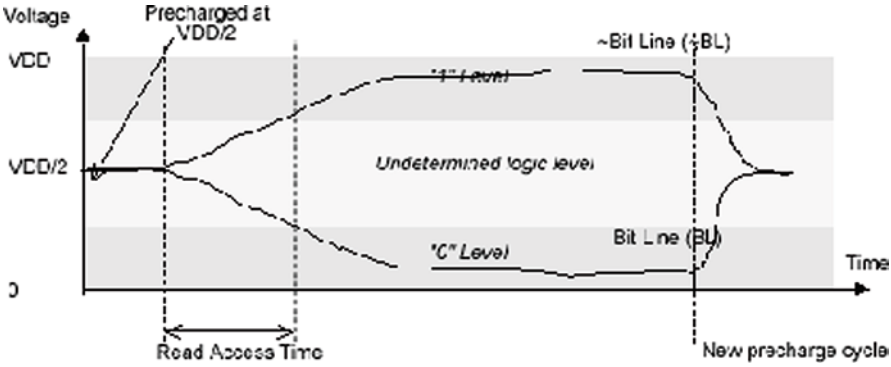

Fig. 1.12 Read cycle using pre-charge circuit

writing to the memory, the bit lines are tied to VDD/2 using appropriate pass gates. When reading, the BL and ~BL diverge from VDD/2 and reach the "1" or "0" levels after a short time. As the SRAM cells are based on active devices, the memories usually provide the fastest read and write access times. A simple pre-charge circuit consists of a NMOS or PMOS. The drain is connected to VDD/2 and the source to the bit line. The pre-charging on bit lines is done whenever a Reset is triggered. The Read cycle using pre-charge circuit is shown in Fig. 1.12 [5].

1.2.1.4 Data I/O Control Logic

The Data Input/Output control logic block is responsible for latching Input data to the SRAM memory cells and also latching the data that needs to be read on the output data ports from the specified address in the SRAM.

The Input data control block is basically a data routing block. Data from the input pins is passed into the block and then transferred to the memory cell array via the buffer circuit and a pass transistor. The pass transistor controls the flow of data into the memory cell array.

The Output data control block is a simple controlled buffer circuit. A tri-state inverter is used to control the flow of data to the Data Out pins on the SRAM chip. When Read is enabled, the tri-state transistors are turned off and prevent data from entering in to the SRAM chip to write. The data is accessed at the specified address on SRAM and latched on the data output pins via Sense amplifier. The I/O control logic block is shown in Fig. 1.13.

1.2.2 Design of Complete SRAM Chip

An SRAM chip with 1 KB memory can be built using 32 blocks of 32 bytes array. The design of circuit elements required to support the operation of SRAM chip is

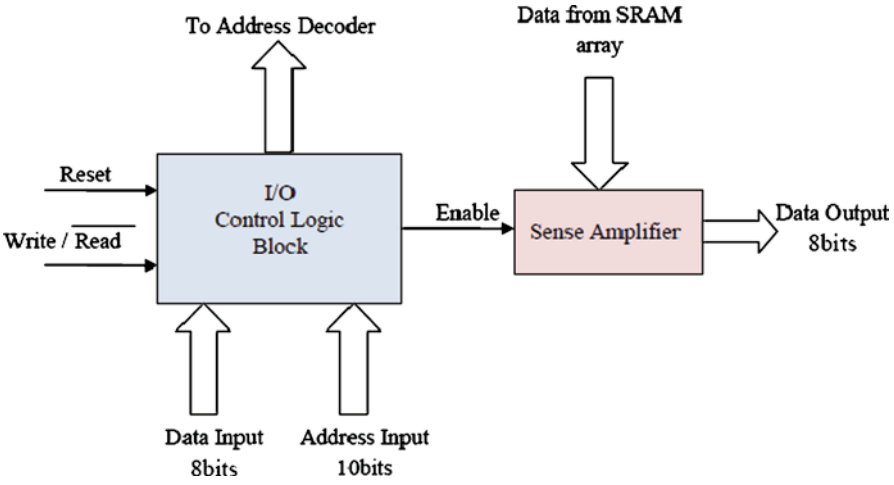

Fig. 1.13 Block diagram of I/O control logic block for 1 KB SRAM chip

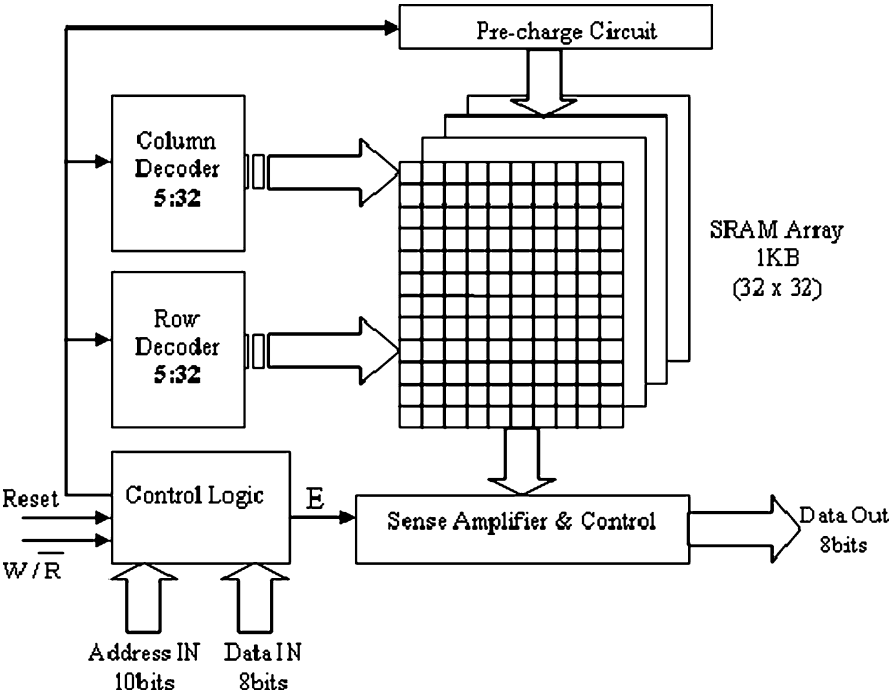

Fig. 1.14 Complete plan of 1 KB SRAM chip

discussed in Sect. 1.2.1. Using these memory cell arrays and circuit elements, a complete 1 KB CMOS based SRAM chip can be designed. In this section, 6T CMOS 1 KB SRAM chip is designed as per the plan shown in Fig. 1.14.

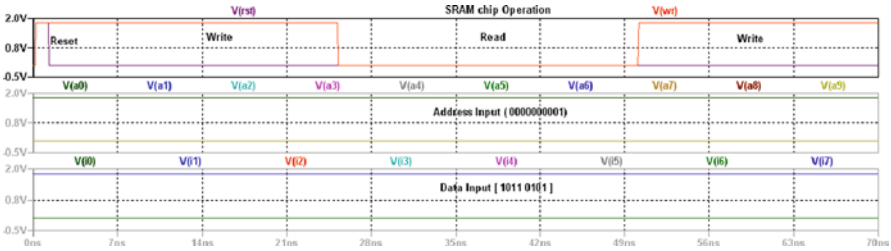

Fig. 1.15 Spice simulations for complete SRAM chip operation

The approximate chip area required for the designed 1 KB CMOS based SRAM chip including circuitry elements can be calculated as follows:

No. of MOSFET used in the design (Approx.) = 56,000
Area of a single MOSFET [NMOS/PMOS – average] (Approx.) = 20 μ²
Total area = 56,000 × 20 μ² = 1,120,000 μ²
Estimated chip area for the designed 1 KB SRAM chip = 1.12 mm²

1.2.3 Simulations of Complete SRAM Chip

The complete 1 KB SRAM chip is designed and simulated using LTspice [6]. The LTspice simulations for the same are shown in Fig. 1.15.

1.2.4 Delay Extraction for SRAM Chip Write/Read Operation

The Write delay and Read access times are extracted for the designed SRAM chip from the simulations.

The Write delay time is measured when Write is enabled until when the data appears on the data bit lines. From the LTspice simulations shown in Fig. 1.16, it can be inferred that the *Write delay time* is 0.24 ns.

The Read Access time is measured from when Read is enabled until when the data appears on the data output lines. From the LTspice simulations shown in Fig. 1.17, it can be inferred that the *Read Access time* is 0.16 ns.

1.2.5 Re-Design of SRAM Chip for Low Power Consumption

The power consumption is very important factor that needs to be considered while designing a chip. It is evident that the SRAM chip is operational whenever the

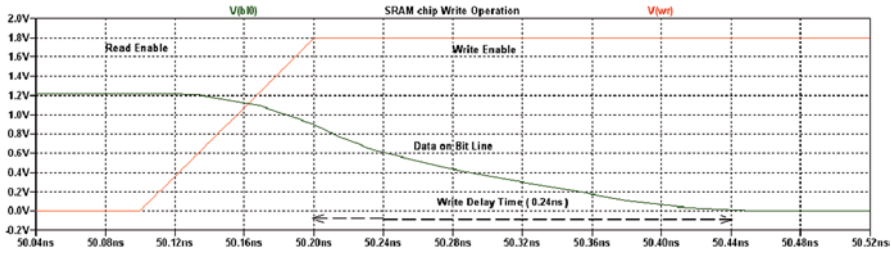

Fig. 1.16 Spice simulation of SRAM chip to measure write delay time

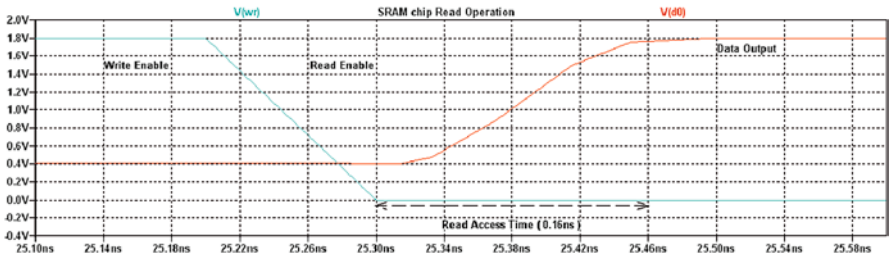

Fig. 1.17 Spice simulation of SRAM chip to measure read access time

word-line is asserted for read/write operation. The current passes through the cell during read/write operation as long as the word-line is asserted. Hence the power consumption in the chip is directly proportional to the time during which the word-line is asserted.

Based on the above hypothesis, certain measures can be taken by implementing appropriate logic to optimize the power consumption. One of the approaches to the solution for the above mentioned problem is to incorporate clock based assertion of word-line. The word-line may be asserted only for a short and optimized duration for which the write or read operation can be performed completely. Hence the power consumption can be reduced to a certain extent.

The pre-charge voltage that is applied on the bit lines also can be optimized to minimize the power consumption. The duration for which the charge applied on the bit lines may be optimized so that it is just sufficient enough for the sense amplifiers to sense the voltage levels at the shortest time.

Various other measures may be taken based on the floor plan of the transistor, layout, dimensions of transistors, and other factors etc. to optimize power consumption. Additional circuitry also may be incorporated to obtain an optimized and lowest power consuming SRAM chips.

Appendix

A. SPICE code for SRAM circuit elements

```
* Interface components subcircuits
* File Name: subckt_comp.sp

**** Sense Amplifier unit subcircuit
.subckt sense_amplifier Enable BL bBL output src gnd

M1 output A src src cmosp l=0.18u w=0.72u
M2 A output src src cmosp l=0.18u w=0.72u

M3 A BL B gnd cmosn l=0.18u w=0.36u
M4 output bBL B gnd cmosn l=0.18u w=0.36u
M5 B Enable gnd gnd cmosn l=0.18u w=0.36u

.ends

**** Sense Amplifier row subcircuit
.subckt sense_amplifier_row E BL0 bBL0 BL1 bBL1 BL2 bBL2 BL3 bBL3
BL4 bBL4 BL5 bBL5 BL6 bBL6 BL7 bBL7 D0 D1 D2 D3 D4 D5 D6 D7 VDD gnd

X0 E BL0 bBL0 N0 VDD gnd sense_amplifier
X1 E BL1 bBL1 N1 VDD gnd sense_amplifier
X2 E BL2 bBL2 N2 VDD gnd sense_amplifier
X3 E BL3 bBL3 N3 VDD gnd sense_amplifier
X4 E BL4 bBL4 N4 VDD gnd sense_amplifier
X5 E BL5 bBL5 N5 VDD gnd sense_amplifier
X6 E BL6 bBL6 N6 VDD gnd sense_amplifier
X7 E BL7 bBL7 N7 VDD gnd sense_amplifier

X8  N0 D0 VDD gnd buffer
X9  N1 D1 VDD gnd buffer
X10 N2 D2 VDD gnd buffer
X11 N3 D3 VDD gnd buffer
X12 N4 D4 VDD gnd buffer
X13 N5 D5 VDD gnd buffer
X14 N6 D6 VDD gnd buffer
X15 N7 D7 VDD gnd buffer

X16 E Dis VDD gnd inverter

M1 D0 Dis gnd gnd cmosn l=0.18u w=0.36u
M2 D1 Dis gnd gnd cmosn l=0.18u w=0.36u
M3 D2 Dis gnd gnd cmosn l=0.18u w=0.36u
M4 D3 Dis gnd gnd cmosn l=0.18u w=0.36u
M5 D4 Dis gnd gnd cmosn l=0.18u w=0.36u
M6 D5 Dis gnd gnd cmosn l=0.18u w=0.36u
M7 D6 Dis gnd gnd cmosn l=0.18u w=0.36u
M8 D7 Dis gnd gnd cmosn l=0.18u w=0.36u

.ends
```

```
   **** 5 bit Decoder subcircuit
   .subckt decoder_5bit A0 A1 A2 A3 A4 s0 s1 s2 s3 s4 s5 s6 s7 s8 s9
s10 s11 s12 s13 s14 s15 s16 s17 s18 s19 s20 s21 s22 s23 s24 s25 s26
s27 s28 s29 s30 s31 VDD gnd

   X0 A0 bA0 VDD gnd inverter
   X1 A1 bA1 VDD gnd inverter
   X2 A2 bA2 VDD gnd inverter
   X3 A3 bA3 VDD gnd inverter
   X4 A4 bA4 VDD gnd inverter

   X5  bA4 bA3 bA2 bA1 bA0 s0  VDD gnd and5
   X6  bA4 bA3 bA2 bA1  A0 s1  VDD gnd and5
   X7  bA4 bA3 bA2  A1 bA0 s2  VDD gnd and5
   X8  bA4 bA3 bA2  A1  A0 s3  VDD gnd and5
   X9  bA4 bA3  A2 bA1 bA0 s4  VDD gnd and5
   X10 bA4 bA3  A2 bA1  A0 s5  VDD gnd and5
   X11 bA4 bA3  A2  A1 bA0 s6  VDD gnd and5
   X12 bA4 bA3  A2  A1  A0 s7  VDD gnd and5
   X13 bA4  A3 bA2 bA1 bA0 s8  VDD gnd and5
   X14 bA4  A3 bA2 bA1  A0 s9  VDD gnd and5
   X15 bA4  A3 bA2  A1 bA0 s10 VDD gnd and5
   X16 bA4  A3 bA2  A1  A0 s11 VDD gnd and5
   X17 bA4  A3  A2 bA1 bA0 s12 VDD gnd and5
   X18 bA4  A3  A2 bA1  A0 s13 VDD gnd and5
   X19 bA4  A3  A2  A1 bA0 s14 VDD gnd and5
   X20 bA4  A3  A2  A1  A0 s15 VDD gnd and5
   X21  A4 bA3 bA2 bA1 bA0 s16 VDD gnd and5
   X22  A4 bA3 bA2 bA1  A0 s17 VDD gnd and5
   X23  A4 bA3 bA2  A1 bA0 s18 VDD gnd and5
   X24  A4 bA3 bA2  A1  A0 s19 VDD gnd and5
   X25  A4 bA3  A2 bA1 bA0 s20 VDD gnd and5
   X26  A4 bA3  A2 bA1  A0 s21 VDD gnd and5
   X27  A4 bA3  A2  A1 bA0 s22 VDD gnd and5
   X28  A4 bA3  A2  A1  A0 s23 VDD gnd and5
   X29  A4  A3 bA2 bA1 bA0 s24 VDD gnd and5
   X30  A4  A3 bA2 bA1  A0 s25 VDD gnd and5
   X31  A4  A3 bA2  A1 bA0 s26 VDD gnd and5
   X32  A4  A3 bA2  A1  A0 s27 VDD gnd and5
   X33  A4  A3  A2 bA1 bA0 s28 VDD gnd and5
   X34  A4  A3  A2 bA1  A0 s29 VDD gnd and5
   X35  A4  A3  A2  A1 bA0 s30 VDD gnd and5
   X36  A4  A3  A2  A1  A0 s31 VDD gnd and5

   .ends
```

B. SPICE code for RC parasitic extraction from SRAM cell layout

```
   * LINUX           Sat Mar 24 15:11:27 2007
   * PROGRAM advgen
   * SPICE LIBRARY

   .SUBCKT sram_cell VDD GND BL WL bBL
   * Caps2d version: 8
   * TRANSISTOR CARDS

   MN0    net5#4  WL#3 BL#1    GND#1   nmos    L=0.18U W=1U + effW=1e-06
   MI3/MN0 net5#2  net12#2    GND#3   GND#1   nmos    L=0.18U W=1U     +
effW=1e-06
   MI2/MN0 net12#7 net5#7     GND#4   GND#1   nmos    L=0.18U W=1U     +
effW=1e-06
   MN1    net12#4 WL#6 bBL#1  GND#1   nmos    L=0.18U W=1U + effW=1e-06
   MI3/MP0 net5#5  net12      VDD#2   VDD#1   pmos    L=0.18U W=1U     +
effW=1e-06
   MI2/MP0 net12#6 net5#6     VDD#5   VDD#4   pmos    L=0.18U W=1U     +
effW=1e-06

   *
   * RESISTOR AND CAP/DIODE CARDS
   *

   Rg1  WL#3    WL#2       45.2083     $poly
   Rg2  net12   net12#2   116.2500     $poly
   Rg3  net12   net12#3    53.9583     $poly
   Rg4  net5#6  net5#7    115.4167     $poly
   Rg5  net5#7  net5       35.6250     $poly
   Rg6  WL#6    WL#5       46.2500     $poly
   Rf1  BL#1    BL          1.0564     $mt1
   Rf2  WL#1    WL#2        2.4110     $mt1
   Rf3  GND#1   GND#2       2.4778     $mt1
   Rf4  GND#1   GND#3       0.2601     $mt1
   Rf5  VDD#1   VDD#2       0.3244     $mt1
   Rf6  VDD#2   VDD#3       2.4814     $mt1
   Rf7  net5    net5#2      1.2832     $mt1
   Rf8  net5#2  net5#3      0.3333     $mt1
   Rf9  net5#3  net5#4      1.0512     $mt1
   Rf10 net5#3  net5#5      0.3238     $mt1
   Rf11 VDD#4   VDD#5       0.1669     $mt1
   Rf12 VDD#5   VDD#6       2.5082     $mt1
   Rf13 GND#1   GND#4       0.3203     $mt1
   Rf14 GND#1   GND#5       2.4776     $mt1
   Rf15 net12#4 net12#5     1.3201     $mt1
   Rf16 net12#5 net12#6     0.3653     $mt1
   Rf17 net12#6 net12#3     1.4158     $mt1
   Rf18 net12#5 net12#7     0.2855     $mt1
   Rf19 WL#4    WL#5        2.4070     $mt1
   Rf20 bBL#1   bBL         0.9190     $mt1
   Re1  VDD#6   VDD         0.2832     $mt2
   Re2  VDD     VDD#3       1.2223     $mt2
   Re3  GND#5   GND         0.7362     $mt2
   Re4  GND     GND#2       1.3788     $mt2
   Re5  WL#4    WL          2.2416     $mt2
   Re6  WL      WL#1        2.0018     $mt2
```

```
*
*        CAPACITOR CARDS
*

C1    VDD      GND     2.038E-16
C2    BL       GND     2.661E-16
C3    WL       GND     6.774E-16
C4    bBL      GND     2.865E-16
C5    net12    GND     1.597E-16
C6    net5     GND     4.261E-16
C7    net5#6   GND     2.298E-16
C8    WL#6     GND     7.296E-17
C9    net5#7   GND     1.507E-16
C10   net12#2  GND     2.056E-16
C11   WL#3     GND     7.003E-17
C12   WL#5     GND     3.171E-16
C13   net12#3  GND     4.754E-16
C14   WL#2     GND     3.062E-16
C15   WL#4     GND     7.411E-16
C16   VDD#6    GND     4.331E-16
C17   VDD#3    GND     6.310E-16
C18   WL#1     GND     6.933E-16
C19   VDD#4    GND     1.635E-16
C20   bBL#1    GND     7.459E-16
C21   net12#4  GND     2.806E-16
C22   net12#7  GND     1.575E-16
C23   net5#2   GND     3.270E-16
C24   net5#4   GND     2.873E-16
C25   BL#1     GND     7.836E-16
C26   VDD#5    GND     1.549E-16
C27   net12#6  GND     3.979E-16
C28   net5#5   GND     1.381E-16
C29   VDD#2    GND     1.350E-16
C30   VDD#1    GND     1.746E-16
C31   net5#3   GND     3.524E-16
C32   net12#5  GND     3.510E-16

.ENDS sram_cell
```

References

1. Kang S, Leblebici Y (2003) CMOS digital integrated circuits, 3rd edn. Tata McGraw-Hill, Boston
2. Static Random Access Memory Interface (2007) EE Herald. http://www.eeherald.com/section/design-guide/esmod15.html. Accessed 4 June 2007
3. LT Spice User Guide (2006) Linear technology. http://LTspice.linear-tech.com/software/scad3.pdf. Accessed 10 August 2006
4. Mehata K, Zinkovski I (2002) CSE447: Design of 1 KB SRAM chip. The Pennsylvania State University. http://www.cedcc.psu.edu/khanjan/vlsihome.htm. Accessed 4 June 2007
5. Static RAM Memory (2006) Institut National des Sciences. https://intranet.insa-toulouse.fr/view/422/content/static_ram.html. Accessed 10 August 2006
6. ECE558: Spice Simulations (2006) University of Massachusetts http://www-unix.ecs.umass.edu/~zzeng/ece558/spice_www/spice.html. Accessed 10 August 2006