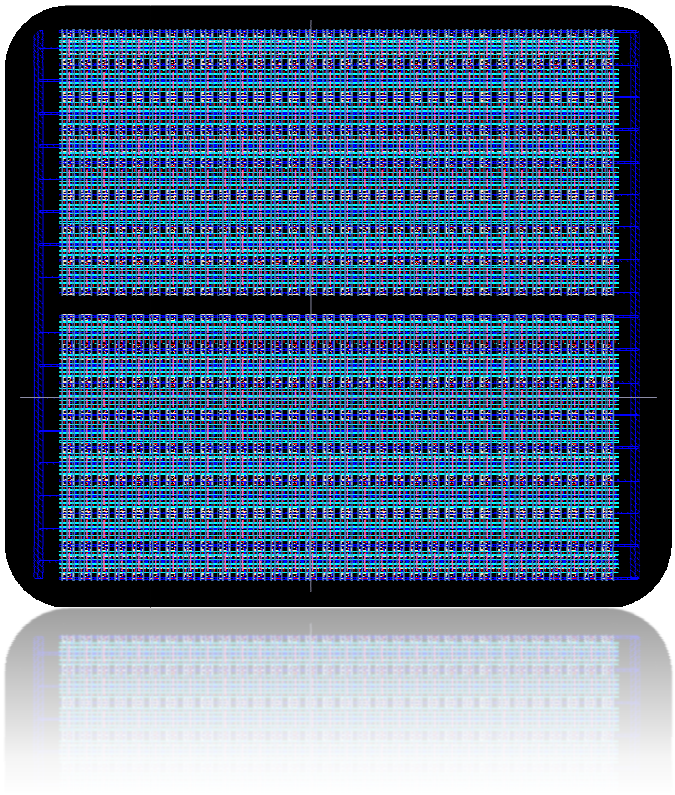
CHALMERS



Optimized Register File Implementation

Master of Science Thesis in Integrated Electronic System Design

Akshay Vijayashekar

Hasan Ali

Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2011

Optimized Register File Implementation

Akshay Vijayashekar and Hasan Ali

Examiner: Professor Per Larsson-Edefors

VLSI Research Group
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden

Department of Computer Science and Engineering
Göteborg, Sweden, June 2011

Abstract

The register file is a critical component in any CPU design. As it is accessed almost every cycle, it contributes significantly to key chip characteristics. It is a vital component to optimize for best performance, area and power dissipation. The purpose of this thesis is to implement a full custom, low power and area efficient register file for an Atmel 32-bit microcontroller. The benchmark design is a flip-flop based register file. This thesis work is carried out in two phases – Literature study phase and Implementation phase.

During the literature study phase, a number of cell topologies for SRAM based register files were explored. A few different cell topologies were selected for implementation and the power dissipation was estimated. Apart from different cell topologies, different register files architectures such as time-multiplexed and partitioned register files have been implemented. Simulation results show that significant area and power savings can be achieved by designing a SRAM based full custom register file, instead of making a design based on flip-flops.

Table of contents

List of figures

List of abbreviations

BL	Bit Lines
DRAM	Dynamic Random Access memory
MUX	Multiplexer
NMOS	N-channel Metal Oxide Semiconductor
PMOS	P-channel Metal Oxide Semiconductor
PVT	Process, Voltage, Temperature
PWL	Piecewise Linear
RBL	Read Bit Line
RWL	Read Word Line
SRAM	Static Random Access Memory
TG	Transmission Gate
WBL	Write Bit Line
WL	Word Line
WWL	Write Word Line

1. Introduction

The aim of this master's thesis is to implement a stand-alone full custom register file with lower power and less area compared to standard cell based register file. The size of the register file is 32 words of 32 bits each with two write ports and four read ports. The frequency of operation is 100 MHz. Atmel 180nm technology libraries are used for simulations.

1.1 Memory hierarchy

The memory hierarchy is a pyramid structure that is commonly used to illustrate the significant differences among memory types as shown in Figure 1.1 [1]. At the top of the hierarchy is the register file. It is accessed in every clock cycle during any program execution and also has the shortest latency time. As the register file is accessed frequently, even a small decrease in the power of the register file will directly translate to less power dissipation of the complete system.

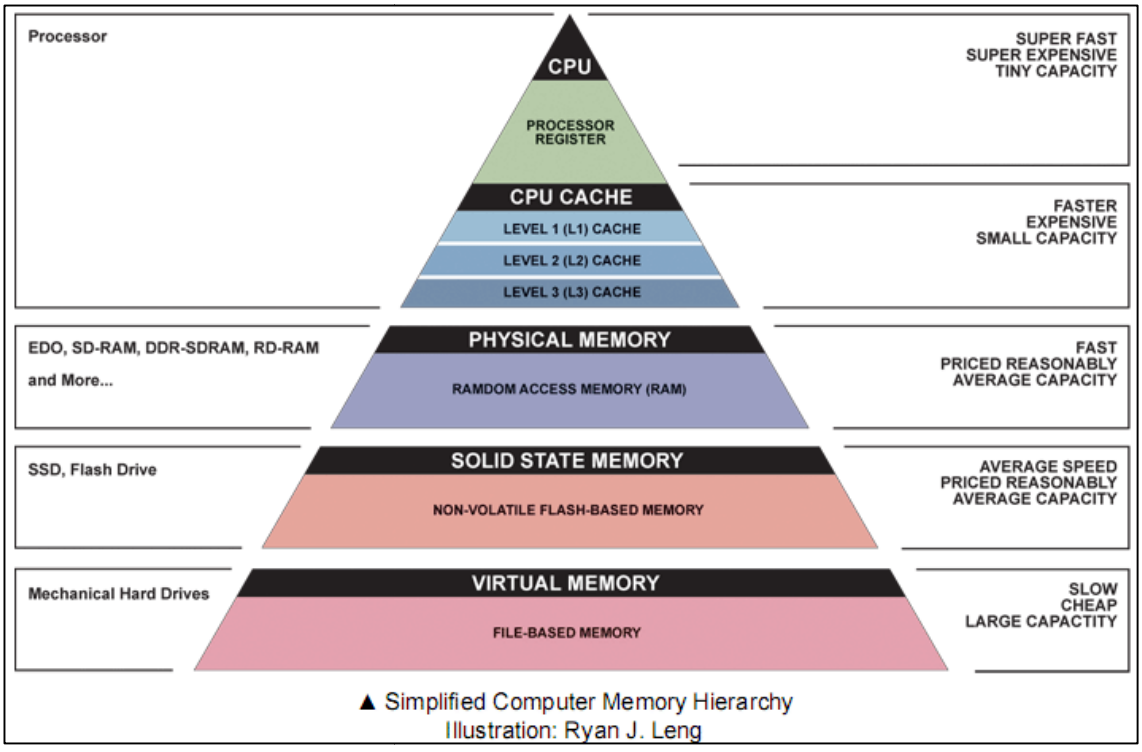


Figure 1. 1 Typical memory hierarchy

1.2 Standard cell based register file

A standard cell based multiport register file uses flip-flops as a basic storage unit defined in the standard cell library. Figure 1.2 shows the interface of a standard cell based register file with two read ports and four write ports and its placement between two pipeline stages.

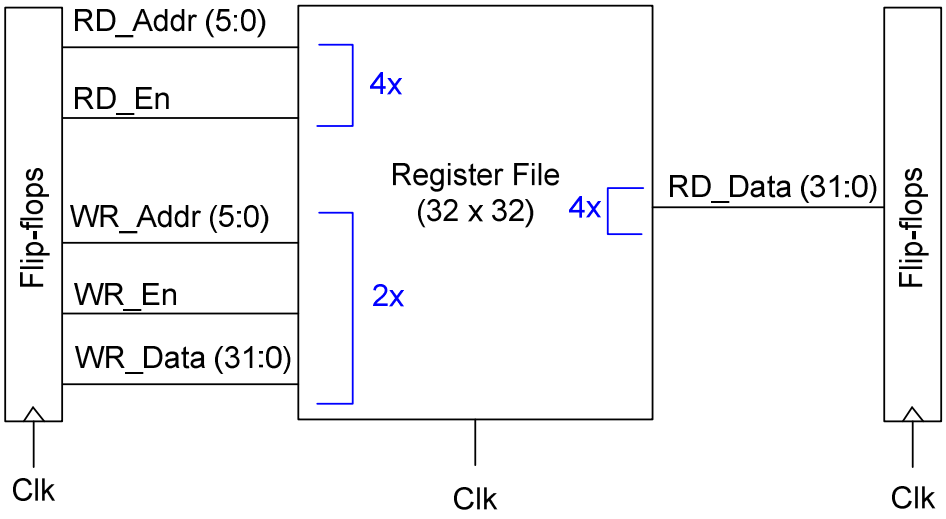


Figure 1. 2 Interface of a standard cell based register file implementation

A standard cell based register file requires huge multiplexers to perform the read operation. The number of these large multiplexers is directly proportional to the number of read ports. The flow to obtain power and area values for this design is discussed in Section 4.1. These values are used as a benchmark for the full custom register file implementation.

1.3 Memory alternatives

During the literature study phase it was noted that there are two common types memory implementations

- Dynamic Random Access Memory (DRAM)
- Static Random Access Memory (SRAM)

The following sub-section gives more details about DRAM and SRAM and also discusses which type of memory can be used for a custom based register file.

1.3.1 DRAM

The DRAM is a type of random access memory which stores a bit of data using a NMOS transistor and a capacitor [3]. When the capacitor is charged, the data saved in it is logic ‘1’, otherwise a logic ‘0’ is saved. Since the capacitor can hold its charge only for a

certain period of time, a DRAM based memory has to be refreshed periodically. This is the reason for calling such memories dynamic.

The advantage of using DRAMs is its simple bit cell structure which is just a transistor and a capacitor. Owing to this, these memories can reach high integration densities. On the downside, these circuits are slow and have an overhead of being refreshed regularly.

1.3.2 SRAM cell

The SRAM cell stores a bit of data using six transistors. A six Transistor (6T) SRAM bit cell is as shown in Figure 1.3 [2].

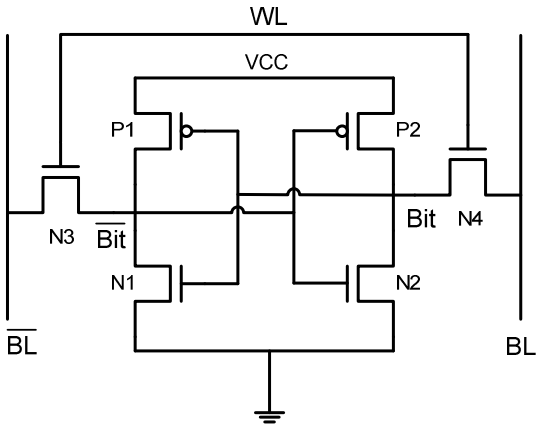


Figure 1. 3 6T SRAM bit cell

The 6T SRAM cell saves one bit using two cross coupled inverters. The value stored by these cross coupled inverters is read or over-written using two access transistors N3 and N4. In Figure 1.3, Word Lines (WLs) enable the access transistors during read or write cycle. The read and write happens through the Bit Lines (BL and $\overline{BL}$).

Suppose logic '1' is to be saved in the SRAM cell. BL would be charged to logic '1' and $\overline{BL}$ would be discharged to logic '0'. These bit lines should be actively driven during the write cycle and the drivers should be strong enough to overwrite the feedback inverters. When WL is asserted, the access transistors are enabled and the value on the BL is written inside the cell. The same procedure is for writing logic '0'.

Suppose logic'1' is stored at 'Bit'. To read data out from the SRAM cell, initially BL and $\overline{BL}$ are charged to the supply voltage, V_{CC}, and then are put in high-impedance state. WL is then asserted and the access transistors are enabled. Logic values at 'Bit' and $\overline{Bit}$ are transferred to the bit lines and the voltage at $\overline{BL}$ will start to drop. As SRAM cells have highly capacitive bit lines, the cell is generally not able to swing the bit line to the full logic

level. Typically sense amplifiers are used to detect the small variations on the bit lines and to give a proper logic output [7].

1.3.3 Comparison between SRAM and DRAM

During the initial study phase it was observed that it is very common for register files to use SRAM based cells. The SRAM cells occupy more area and consume more power than the DRAM cells. The DRAM is slow and it requires a refreshing circuitry. The overhead due to the refreshing circuitry and relatively slow operating speed of DRAM make them an unfavorable choice in register files.

1.4 SRAM based cell design challenges

The sizing of the six transistors in a SRAM bit cell is a delicate procedure. There are many conflicting criteria that need to be considered.

The size of each bit cell has to be as small as possible so as to increase the density of the memory. Next, the access and the pull down NMOS transistors must be big enough to discharge the highly capacitive bit line in a small duration of time. On the other hand, making these transistors big increases the bit line capacitance and thereby affects its discharge time. Also, big access transistors increase the bit line leakage current and thus increase the static power and reduce the bit line noise immunity.

Keeping into consideration the above discussion, the six transistors in the SRAM cell have to be sized so as to get the balance between proper functionality and low power dissipation.

1.5 Power dissipation in SRAM based memories

The total power of the register file is given by the equation

$$P_{total} = P_{dynamic} + P_{static}$$

In the 180nm technology node, it is assumed that the leakage power is insignificant and therefore the major contributor to total power is the dynamic power. This dynamic power is given by the equation below.

$$P_{dynamic} = \alpha f_{clk} V_{DD}^{2} C$$

It has to be noted here that the dynamic power is directly proportional to the capacitance. An SRAM cell based register file has capacitance on the word lines and bit lines which are major contributors to the dynamic power.

2. SRAM cell based register file

2.1 Architecture of SRAM cell based register file

Figure 2.1 shows the typical architecture of 32x32 SRAM cell based register file with one read port and one write port [5].

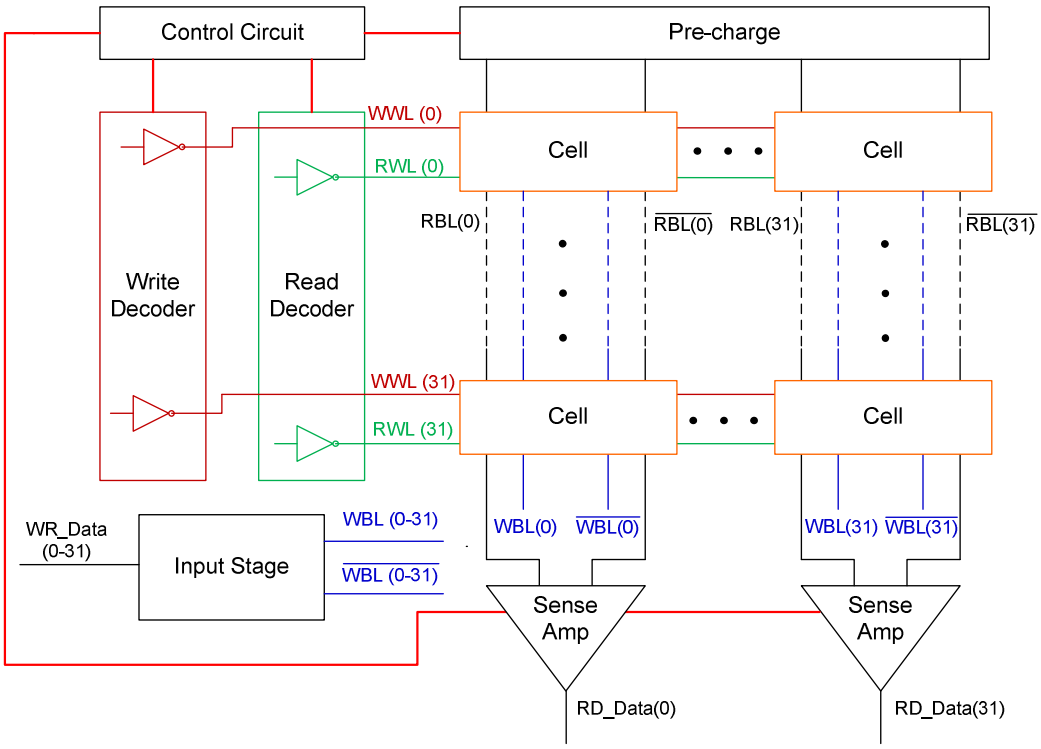


Figure 2. 1 Architecture of a typical register file

A register file has the following major components.

- Decoders
- Storage cell array
- Control circuit
- Output stage (Sense amplifiers in Figure 2.1)

The input data bits, WR_Data, which are to be written into particular word of the register file, are put on the write bit lines (WBLs) by the drivers in the 'Input Stage' module. The data that is read out of the register file on the read bit lines (RBLs). Sense amplifiers detect voltage variation in the RBLs and toggle the output of read port (named as RD_Data) accordingly.

There is a separate decoder for each read and write port as shown in Figure 2.1. A few design implementations for the storage cell array are discussed in Section 2.3. The

storage cell array also consists of a pre-charge circuit and memory cells. Read bit lines are generally pre-charged to a value between the voltage supply and the reference voltage. Read bit lines often do not swing to full logic levels. Sense amplifiers convert the low swing on the read bit lines into full logic levels [5]. All of these components are discussed in detail in the subsequent sections of the report.

From Figure 2.1 it may be noted that the bit lines and word lines are long wires that run across the whole register file. If these parallel wires are placed very close to each other then the data integrity might be compromised due to crosstalk. To avoid cross coupling, bit lines and word lines can be shielded with power lines or the spacing between them should be more than the minimum metal spacing allowed by the CMOS process [4].

Instruction scheduling hardware and software components should ensure that one word is written by one write port only. If more than one write port access the same word in the same cycle there should be some mechanism to disable one of them. If the same address is to be written and read in the same cycle, then bypass multiplexers should be used to pass the data to the read ports directly.

2.2 Register file cell circuit topologies

The bit cells in a register file have dedicated read and write ports unlike SRAM cells [4]. This section presents some of the cell configurations that were seen during the initial literature study phase. The write and read topologies are discussed in separate subsections. A bit cell with a single bit line per read port and write port is referred to as a single-ended cell while the one with two bit lines per read port and write port are called differential-ended cells.

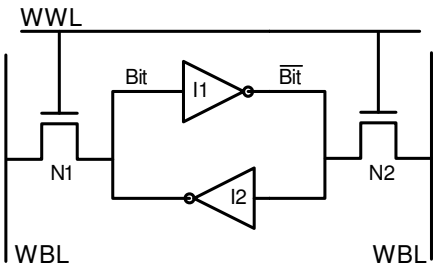
The target register file implementation of this thesis has two write ports and four read ports. For each cell topology, the number of transistors, the number of bit lines and the number of word lines per cell are listed with each cell schematic. The number of transistors excludes the two feedback inverters that are an integral part of every cell.

2.2.1 Write topologies

For write topologies, the number of transistors required per cell is listed for one and two write ports.

Design 1

The write topology shown in Figure 2.2 is differential-ended and is similar to the 6T SRAM cell write topology [4]. Transistors N1 and N2 shall be replicated for each write port. This cell has one word line and two bit lines per write port. WWL refers to write word line and WBL refers to write bit line.

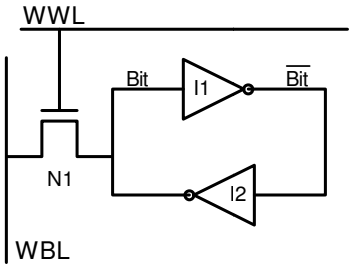


Write Ports	1	2
Bit Lines	2	4
Word Lines	1	2
Transistors	2	4

Figure 2. 2 Write port configuration – Design 1

Design 2

The write topology for the bit cell shown in Figure 2.3 uses only one pass transistor instead of two unlike the previous design [5]. This topology requires one word line and one bit line for each write port. Inverter I2 should be a weak driver so that N1 can easily override it. Since NMOS is not good at passing a logic '1', to write a logic '1' through N1, strong write bit line drivers and a relatively large N1 is required. This is one of the drawbacks of this cell topology. This write topology is discussed in more detail in Section 4.4.

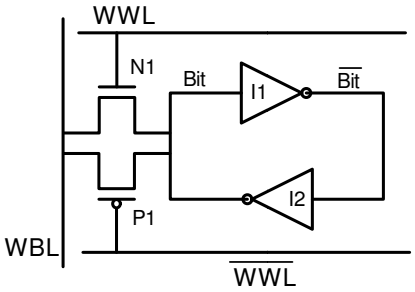


Write Ports	1	4
Bit Lines	1	4
Word Lines	1	4
Transistors	1	4

Figure 2. 3 Write port configuration – Design 2

Design 3

Figure 2.4 shows another bit cell which is similar to Design 2 but uses a transmission gate instead of NMOS pass transistor [4]. The transmission gate helps to write a strong logic '1' and logic '0' and increases the noise margin. Unlike Design 2, this cell requires two word lines per write port. This configuration solves the problem of writing logic '1' but with an increase in the number of word lines.

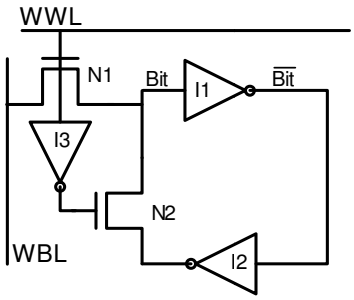


Write Ports	1	2
Bit Lines	1	2
Word Lines	2	4
Transistors	2	4

Figure 2. 4 Write port configuration – Design 3

Design 4

The cell configuration shown in Figure 2.5 is a single-ended structure which requires one word line [4]. The word-line inverse is generated using I3 and it disables the feedback path of the inverters during the duration of the write cycle. This allows N1 to write a logic '0' or logic '1' using an NMOS pass transistor. Inverter I3 can also be moved outside the cell but in that case an extra word line is required per write port. For two write ports, I3 would be replaced by a NOR gate. Both of the word lines would be fed to a NOR-gate and the output of the NOR gate would be connected to gate of N2. This write topology is discussed in detail in Section 4.5.

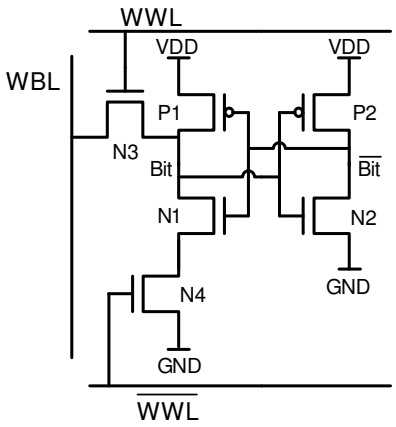


Write Ports	1	2
Bit Lines	1	2
Word Lines	1	2
Transistors	4	7

Figure 2. 5 Write port configuration – Design 4

Design 5

The write topology shown in Figure 2.6 is similar to Design 4. The feedback inverter is disabled during the write cycle by disabling N4 [4]. This configuration requires two word lines for one write port and three word lines for two write ports. The inverse of both word lines can be fed to an AND gate and the output of AND gate would drive the gate of the N4 transistor of each cell in the memory word.

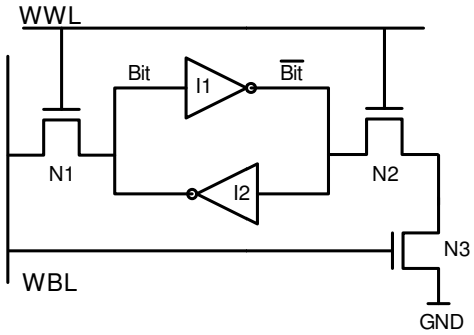


Write Ports	1	2
Bit Lines	1	2
Word Lines	2	4
Transistors	2	3

Figure 2. 6 Write port configuration – Design 5

Design 6

The write configuration depicted in Figure 2.7 uses one word line and one bit line per write port [4, 5]. The transistor N1 allows writing a strong logic '0'. When logic '1' is to be written to the cell, N2 and N3 pull down bit inverse to logic '0'. This configuration allows using small transistors for write port and uses minimum number of bit lines and word lines.

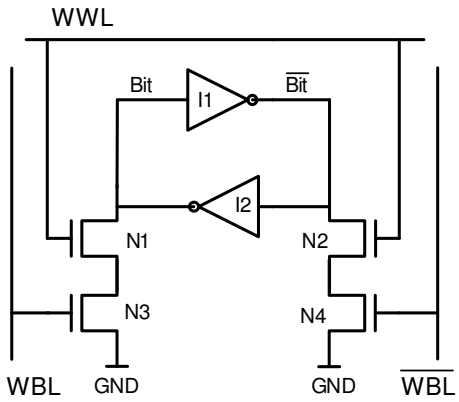


Write Ports	1	2
Bit Lines	1	2
Word Lines	1	2
Transistors	3	6

Figure 2. 7 Write port configuration – Design 6

Design 7

Figure 2.8 shows a write configuration which uses four transistors, one word line and two bit lines per write port [4]. When the value of the write bit line and the inverse of the write bit line is logic '0' then the cell keeps its old data. This can be useful in designs which deal with variable word length where it might be required to mask out certain bits in a word such as byte-wide write operations.



Write Ports	1	2
Bit Lines	2	4
Word Lines	1	2
Transistors	4	8

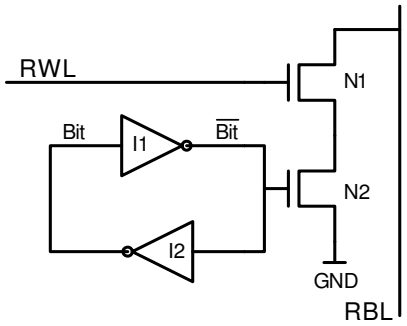
Figure 2. 8 Write port configuration – Design 7

2.2.2 Read topologies

A few read topologies have been listed below. The number of transistors required per cell is listed for one and four read ports.

Design 1

Figure 2.9 shows a read topology which requires pre-charge of the read bit line, RBL before the read starts [4, 5]. When the read word line, RWL, is enabled the bit line is either pulled down to '0' using N1 and N2 or is left floating depending upon the data stored in the cell. This read topology is immune to destructive read problem as the cell storage value is isolated from the read bit line.



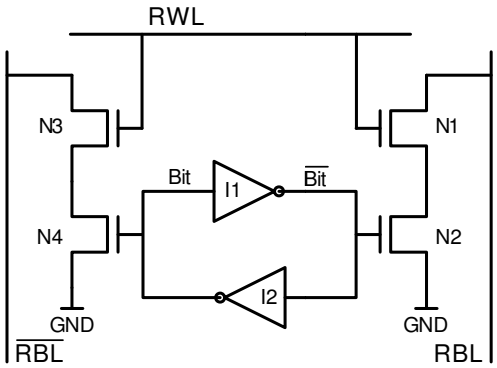
Read Ports	1	4
Bit Lines	1	4
Word Lines	1	4
Transistors	2	8

Figure 2. 9 Read port configuration – Design 1

It is preferred that transistors inside the cell are small as this reduces the overall size of the cell array. Looking at Figure 2.9, it can be noted that sometimes N1 and N2 cannot discharge the highly capacitive bit line completely during the read cycle. In such cases a differential read topology is required which is explained in Design 2.

Design 2

Figure 2.10 shows a differential read topology. Each read port has two bit lines. This read topology also requires a pre-charge circuit. This topology is used when the cell cannot discharge the bit line completely during the read cycle. The design requires sense amplifiers to read out a full logic level. Different types of sense amplifiers are discussed in more detail in Section 2.4.



Read Ports	1	4
Bit Lines	2	8
Word Lines	1	4
Transistors	4	16

Figure 2. 10 Read port configuration – Design 2

2.3 Single-ended vs. differential-ended register file

Register file cells mentioned in Section 2.3 have both single-ended and differential-ended read and write topologies. Differential-ended topologies allow operation at much higher clock rate than single-ended topologies and provide robust noise margins. On the other hand, they increase the number of wires inside the cell and make the cell larger. Single-ended topologies generally have a smaller cell, less number of wires and are intrinsically slower than the differential-ended topologies [4, 6].

In the single-ended read topology, discussed in Design 1 of Section 2.2.2, there is a probability of 0.5 that the read bit line would not switch at all, considering that reading out a logic ‘0’ or a logic ‘1’ is equiprobable. Less switching on the read bit line leads to less power dissipation. So it is preferable to use single-ended read topology in low power register files if the timing constraints can be met [5]. Differential-ended techniques inherently take more power because there is always some swing on the bit lines irrespective of whether a logic ‘1’ or logic ‘0’ is read out. With an increase in the number of words, it takes more time for the cell to change the logic level on bit lines. In this case the sense amplifiers are a better option because they do not require full voltage swing of the bit line to read properly. If the number of words is up to 64, then single-ended topology is a good choice [5].

It was decided that initially single-ended designs would be used in simulations as the size of the target register file is small (< 64 words). If the timing constraints cannot be met then the differential-ended designs would be considered.

2.4 Types of sense amplifier

Sense amplifiers are generally used with differential read topology. These circuits can detect small voltage variations in the differential bit lines and give proper logic level at the output. Following are the two major types of sense amplifiers used in register files [6].

- Voltage Mode Sense Amplifier
- Current Mode Sense Amplifier

Most of the power dissipation in the read cycle is due to the voltage swing on the read bit lines and sense amplifier. The power dissipation by these components can be approximated using the following equation [6].

$$P_{bl} + P_{SenseAmp} = V^2_{bl,swing} * C_{bl} + V_{DD} * I_{SenseAmp}$$

$V_{bl,swing}$ is the voltage swing in the read bit lines and $I_{SenseAmp}$ is the current consumed by the sense amplifier. Voltage mode sense amplifiers require relatively large voltage swing on the bit lines but the current consumed by them is less than the current mode sense amplifiers. On the other hand, current mode sense amplifiers require a small voltage swing on the bit lines but they consume more current than the voltage mode sense amplifiers. Current mode sense amplifiers are mostly used in a SRAMs where the bit line capacitance is huge whereas voltage mode sense amplifiers are preferred in low-power register files [6].

3. Simulation of register file

3.1 Simulation environment

A SPICE transistor netlist of the register file was developed for simulating its behavior and to extract power values of the design. HSPICE [8] was used for initial simulations at Chalmers. Later, Eldo [9] was used to simulate the SPICE transistor netlists at Atmel. Atmel 180nm technology data was used for all simulations. The nominal supply voltage used is 1.8V and the frequency of operation is 100MHz.

3.2 Operating conditions

The effect of process, voltage and temperature (PVT) variations were considered in the simulations. The following process corners were considered.

- Nominal PMOS / nominal NMOS (typical case)
- Slow PMOS / slow NMOS (worst case)
- Fast PMOS / fast NMOS (best case)
- Fast PMOS / slow NMOS case (fast/slow case)
- Slow PMOS / fast NMOS case (slow/fast case)

Simulations were run at the following PVT conditions.

- All process corners at 1.8V, 25^{o} C
- All process corners at 1.6V, 100^{o} C
- All process corners at 1.95V, -40^{o} C

3.3 Transistor models

The following transistor models were used.

- **Low leakage, high threshold transistors**
  These were used for most of the simulations

- **RAM specific transistors**
  These are specially designed transistors to be used in RAMs. Simulation results with these transistors are discussed in Appendix C.

Apart from the length and width of transistor, the following parameters were defined for all the transistor models in the SPICE netlists. The notations of 'drain length' and 'width' are as shown in Figure 3.1.

- Source Area (AS) = drain length * width
- Drain Area (AD) = drain length * width
- Source Perimeter (PS) = (2*drain length) + width
- Drain Perimeter (PD) = (2*drain length) + width

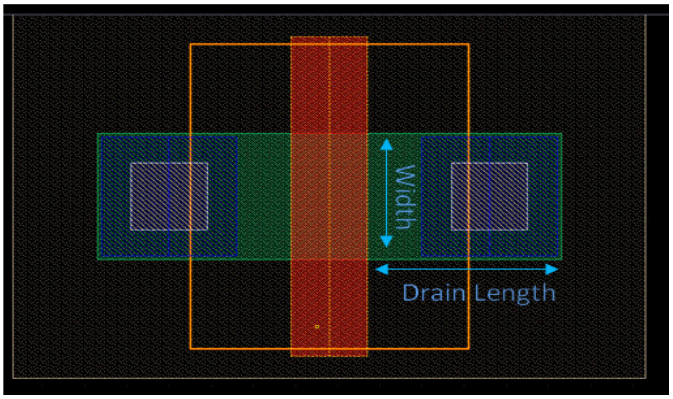


Figure 3. 1 Layout of transistor

3.4 Test bench in SPICE

To verify the functionality of the register file a test bench was developed in SPICE. The input buses are driven using 'sigbus' along with the keyword 'pattern' and 'setbus' commands. These commands enable to define the input buses using hexadecimal values. Signals like clock, read enable, write enable are generated using Piecewise Linear (PWL) sources. Rise and fall times of 0.1ns are used for all simulations unless otherwise stated. The logic level of the output buses is checked by the test bench using the 'checkbus' command 1ns before the next positive edge of clock. Voltage levels less than 10% of the supply voltage, V_{CC} are considered as 'logic 0' and greater than 90% is considered as 'logic 1'. Figure 3.2 shows the test vectors used to calculate the power for standard cell implementation and full custom implementation. The power is calculated between 11ns and 102ns.

Time (ns)	READ				WRITE			
	Addr Port A/B	Data Port A/B	Addr Port C/D	Data Port C/D	Addr Port A	Data Port A	Addr Port B	Data Port B
11 – 20	-	-	-	-	0	1s	30	0s
21 – 30	0	1s	30	0s	1	0s	31	1s
31 – 40	1	0s	31	1s	0	0s	30	1s
41 – 50	0	0s	30	1s	1	1s	31	0s
51 – 60	1	1s	31	0s	0	1s	30	0s
61 – 70	0	1s	30	0s	1	0s	31	1s
71 – 80	1	0s	31	1s	0	0s	30	1s
81 – 90	0	0s	30	1s	1	1s	31	0s
91 – 100	1	1s	31	0s	-	-	-	-

Figure 3. 2 Test vectors used to calculate power of register file

3.5 Assumptions in SPICE transistor netlists

In the SPICE transistor netlist the clock signal is generated using a Piecewise Linear (PWL) source. The duty cycle of the clock is 50%. The SPICE transistor netlist depends on the positive and negative edge of the clock for proper functionality. A clock duty cycle compensation circuit was developed which enabled the functionality of the register file to be dependent on the positive edge of the clock alone; the circuit is discussed in Appendix B. The change in duty cycle of the clock signal was considered during the last stages of the thesis and hence the design implementations discussed in the report do not use this circuit.

All the input buses are driven at the start of the clock cycle and the arrival time is the same for all input signals. However, in real systems the arrival time of inputs may vary due to the placement of pipeline flip-flops which drive the input of the register file. To avoid this problem, flip-flops which drive the inputs to the register file can be made part of the full custom register file. However, in the SPICE netlists, no flip-flops are considered at the input of the register file.

3.6 Local clock generations and control circuit

As specified before, the input clock frequency is 100 MHz. For correct functionality of the register file some signals are needed to be generated locally such as Pre-charge (PC), Read Word Line (RWL), Write Word Line (WWL), etc. A higher frequency clock is generated by delaying the 100 MHz clock and use of logic gates.

To delay the clock, delay cells from the Atmel special cells library are used. The delay of these logic gates vary a lot across the corners and it was a big challenge to generate

the signals to ensure proper functionality across the PVT corners. The transistors were sized appropriately to make sure the register file worked across all the corners.

3.7 Modeling of wire capacitance

Figure 3.3 shows the simplified architecture which was modeled in SPICE for a register file with one read port and one write port. SRAM based designs have long wires that run across the whole length and width of the register file. Since the SPICE transistor netlist doesn't consider the parasitics of these wires implicitly, they need to be modeled as well. To simulate the effect of wiring in the netlists, capacitors are added to all the netlist nodes like bit lines, word lines and all other wires that run across the register file. The estimation of capacitance for the SPICE transistor netlists is discussed in Chapter 4. The outputs of the register file are connected to flip flops of the next pipeline stage. The load of flip flops on these output lines is modeled using capacitor 'C_FF'. The value of C_FF is considered to be 10fF throughout this report. The lines in blue show the signals generated by the control circuit. 'CBL' and 'CWL' are added to model the wire capacitance due to bit lines and word lines respectively. This interface of the register file is similar to the standard cell implementation interface which is discussed in Section 1.2.

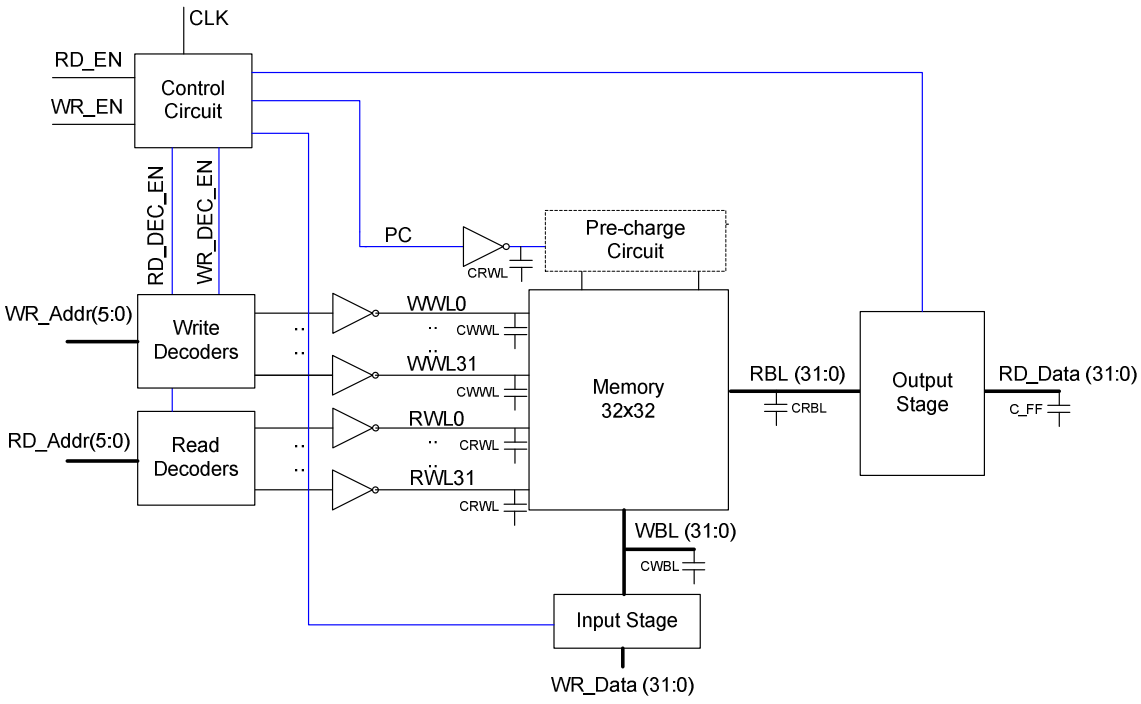


Figure 3. 3 Architecture of simulated register file

By default, the 'Input Stage' is an inverter, referred to as a driver for input data bits and the 'Output Stage' is a set of two inverters called a buffer. However a few design implementations have different circuits for input and output stages. These are discussed in

Chapters 4 and 5. The resistance was assumed to have no impact on timing or power in the SPICE transistor netlists.

3.8 Decoders

The decoders were designed using AND gates. The 5-address bits and their inverse are assumed to be running across the length of the register file. The capacitance of these address lines are modeled by CRBL as shown in Figure 3.4. It is considered to be the same as the capacitance on read bit lines as the length of both the wires is the same.

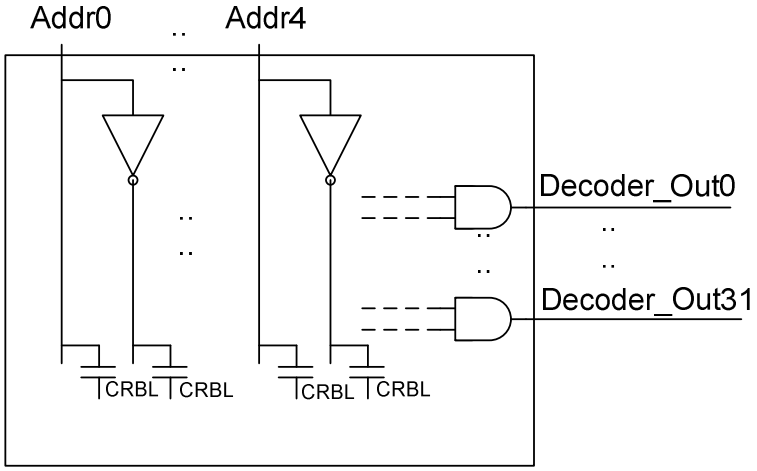


Figure 3. 4 Decoder

4. Implementation of designs

4.1 Benchmark design

The benchmark for a full custom register file is a standard cell implementation of the register file. The interface diagram of a 4 Read / 2 Write port standard cell implementation is shown in Figure 1.2. A standard cell register file, based on flip-flops, has been described using Verilog. Atmel 180nm technology data was used to extract the area and power values. The test stimuli used to obtain power values for this design is same as the test vectors discussed in Section 3.4. Figure 4.1 shows the flow for extraction of area and power values for the standard cell register file implementation.

The Verilog description, modeling the register file, is first synthesized. It is then followed by Place and Route. The clock tree distribution is done next which distributes the clock signals to all the elements. As this design is based on flip-flops, the clock network has to be routed to 1024 flip-flops and it takes about 10% of the power. Clock Tree Synthesis (CTS) is then performed where the clock signal is routed to all the elements that require it. Post CTS, the timing is extracted from PrimeTime to run the simulation. After extracting the VCD (Value Change Dump) file, which is a record of the signal switching activity, the power for this register file is extracted from PrimeTime PX. Using the procedure mentioned above, the power of the standard cell based register file implementation working at a frequency of 100 MHz with 1.8V power supply is **2850 µW**. The gate count for the design is **16,728**. Figure 4.2 shows the power consumed by different elements in the register file. The combinational logic, which contributes to 65% of the power, is mostly the huge multiplexers which are used to connect the registers to the input write ports and output read ports. The memory core is essentially made up of 1024 flip-flops.

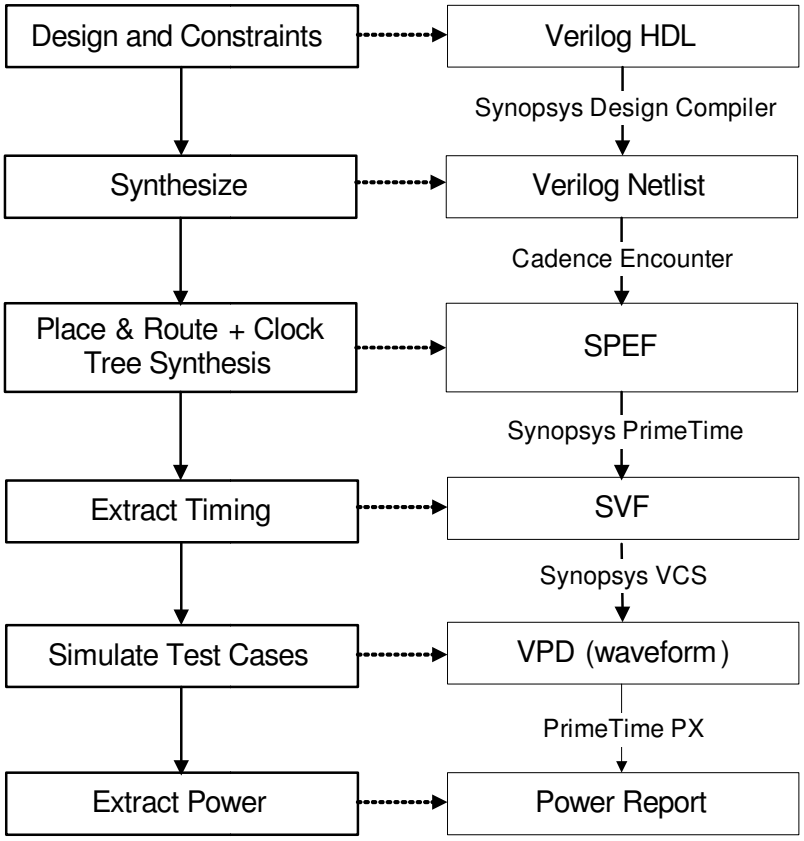


Figure 4. 1 Flow for area and power extraction for a standard cell register file implementation

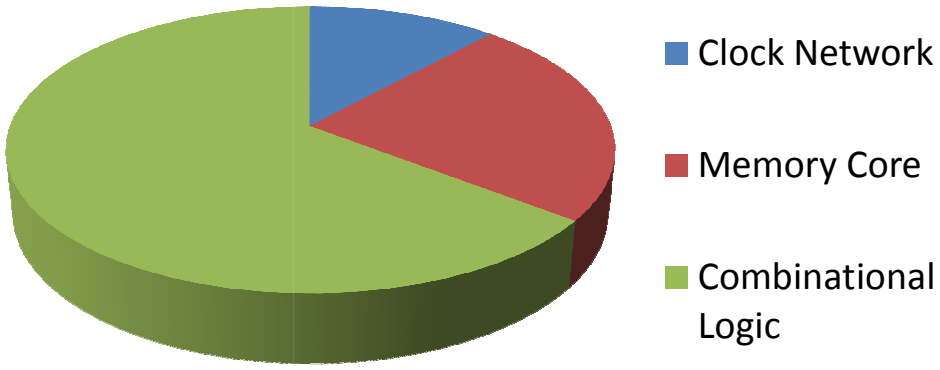


Figure 4. 2 Power analysis of standard cell register file

The area value obtained from the layout of the standard cell register file implementation is **144,400 µm^2**. The layout is as shown in Figure 4.3.

Despite the technological data and test stimuli used for simulations of standard cell implementation are the same as that of the full custom register file, there is a slight possibility that the power comparison may not be 100% accurate. This could be because the

flow to obtain the power number is different for standard cell and custom register file. In this design, the standard cell one, there is no load at the output, unlike in the full custom design. It is however assumed that the primary output load would not have a big impact on the power of standard cell implementation.

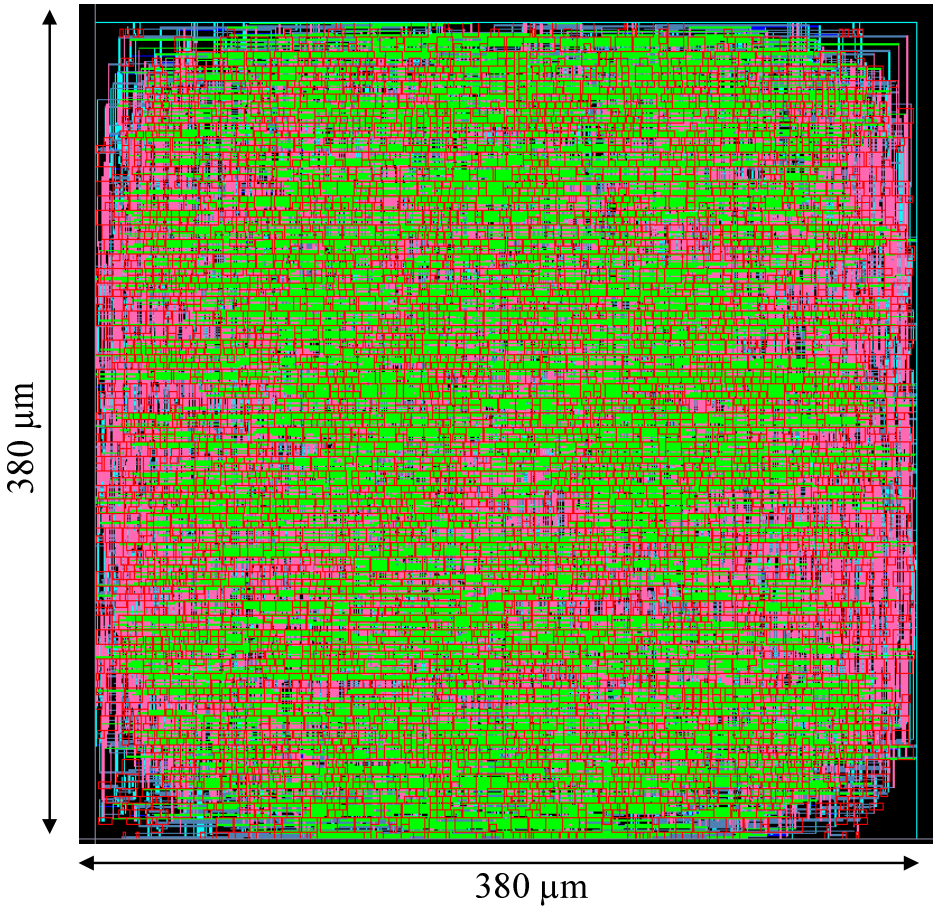


Figure 4. 3 Layout of standard cell register file

4.2 SRAM based register file - Design 1

The first cell design chosen for SRAM based register file is shown in Figure 4.4. Its main features are as listed below.

- Single-ended read and write topology
- Immune to destructive read

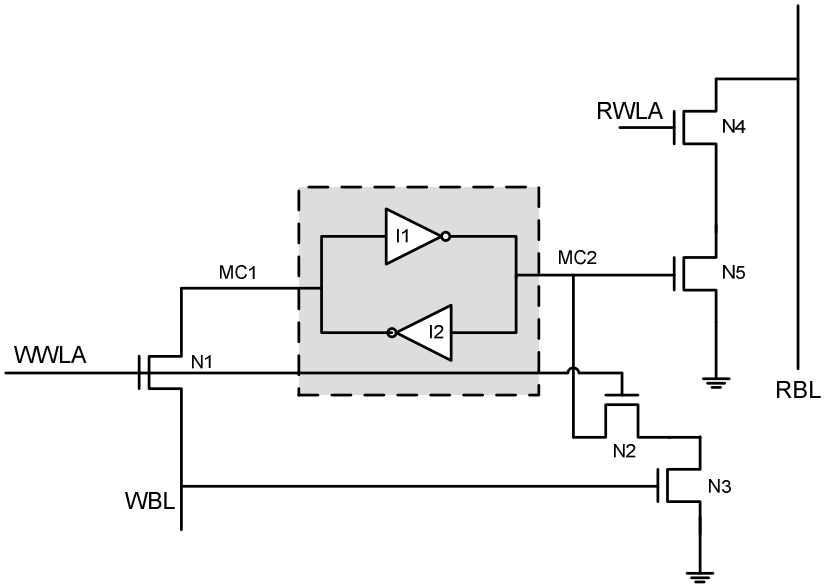


Figure 4. 4 Memory cell for Design 1

4.2.1 Functionality

This design uses three transistors per write port and two transistors per read port. When a logic '0' is to be written to the MC1 node, N1 acts as a pass transistor for the data in bit on WBL and the value is written inside the cell. When a logic '1' is to be written to the MC1 node, N2 and N3 pull down the MC2 node to logic '0'. This indirectly writes the logic '1' at the MC1 node.

This cell structure requires external pre-charge of the read bit lines for the read operation. The bit line should be charged to the supply voltage before the read starts. If the value at MC2 node is '0' then there would be no discharge path for the read bit line and the highly capacitive bit line retains logic '1'. If logic '1' is stored at MC2 node, it provides a discharge path for the bit line through N4 and N5 and the read bit line gets discharged to a logic '0'. This type of read topology is immune to destructive read problem as N4 and N5 isolate the bit line and MC2 node. A 4 Read / 2 Write port bit cell has 18 transistors as shown in Figure 4.5.

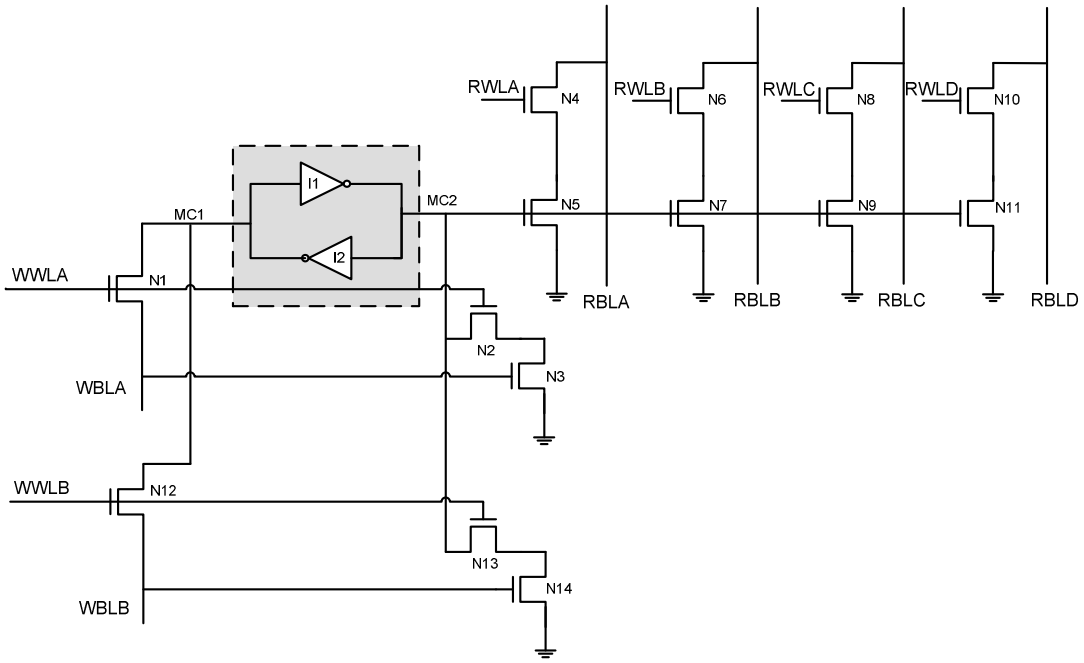


Figure 4. 5 Design 1 memory cell with 4R/2W configuration

4.2.2 Architecture

The architecture of the register file is similar to the architecture shown in Figure 3.3. A pre-charge circuit charges the bit line to the supply voltage during the read operation. It consists of one PMOS for each read bit line. The duration of pre-charge is controlled by a signal generated by the control circuit.

In huge SRAM memories, it is preferred to have small transistors so that the overall size of the memory is kept at a minimum. These small transistors typically cannot drive the read bit lines to the full logic levels during the read cycle and sense amplifiers are thus used to detect the voltage changes on the read bit lines. The size of the register file is very small compared to such SRAM memories, so the capacitance associated with the bit lines is also small. In the simulations it was observed that the minimum size transistor can easily discharge the read bit line within the read cycle. To keep the output stage simple, two inverters were used at the output of the bit lines. For the input stage, inverters were used to drive the write bit lines.

4.2.3 Estimation of parasitics

To estimate the parasitics, a cell was laid out using Cadence Virtuoso [10], as shown in Figure 4.6. The cell has dimensions of 13 µm x 5.5 µm. Initially, parasitics of 0.1 fF/µm were considered as a rule of thumb. As it was realized that the power contributed by the parasitics was very significant, an effort was made to get closer to the realistic parasitic numbers. It should also be noted that the transistors sizes in the layout were not the same as

in SPICE simulations. As transistor size shrinks are easy to be performed in late layout stages, the transistor sizes are a bit bigger in the layouts to accommodate for any changes in the SPICE transistor netlist.

A 32x32 memory was laid out as shown in Figure 4.7 and parasitics of bit lines and word lines were extracted from the layout. A major factor contributing to the wire capacitance were the neighboring wires of the same metal layer. The values of extracted parasitics used in the SPICE transistor netlist are listed below.

- Read bit line = 35 fF
- Write bit line = 50 fF
- Read word line = 80 fF
- Write word line = 100 fF

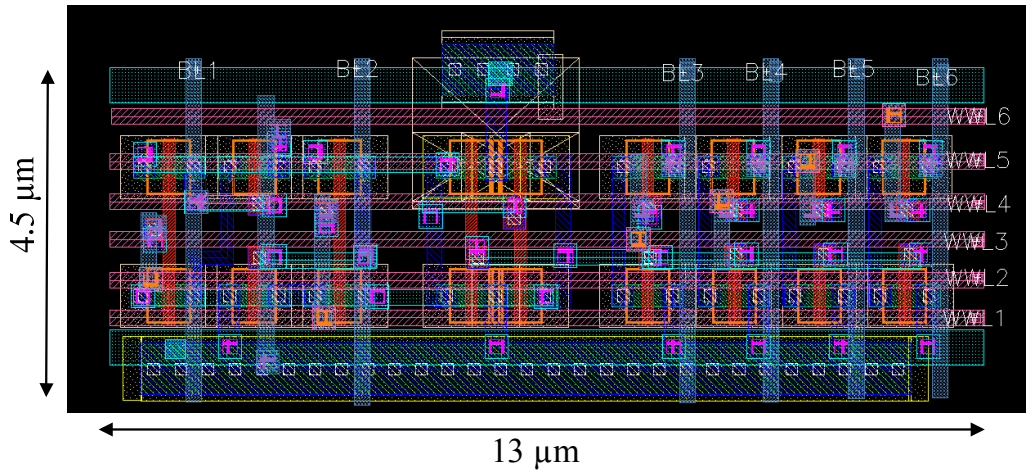


Figure 4. 6 Layout of cell of Design1

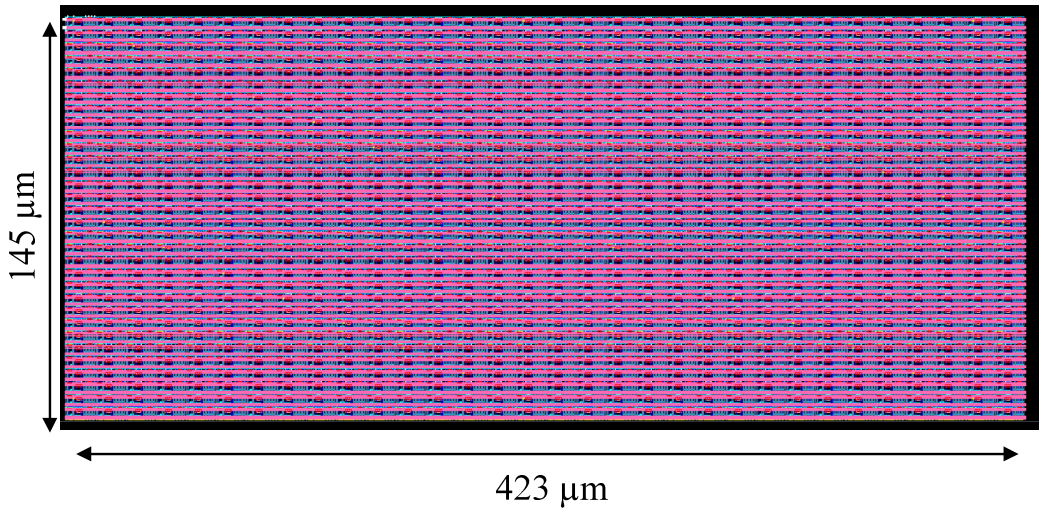


Figure 4. 7 Layout of 32x32 memory array of Design1

4.2.4 Power analysis

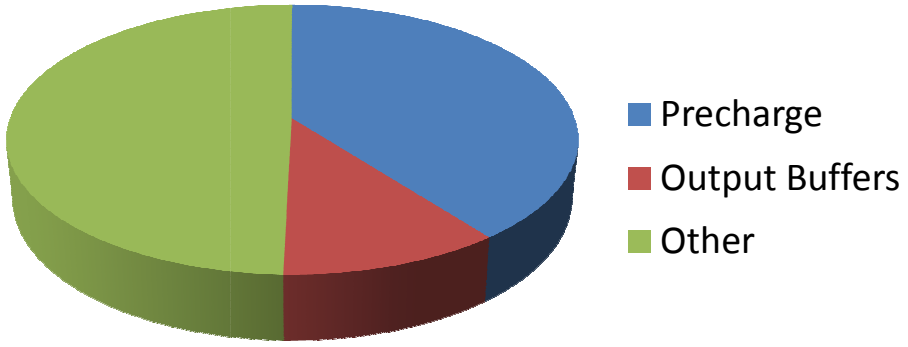


Figure 4. 8 Power analysis of Design 1

The power value obtained for this design implementation was **2649 µW**. The power of this design was quite close to the power of standard cell implementation. Figure 4.8 shows the split up of power for the design. Pre-charge represents the power associated with the read bit lines and it is quite significant part of the total power.

4.2.5 Sense amplifier

It is clear from the Figure 4.8 that the read bit lines contribute to a lot of power. As there are four read bit lines, a reduction in the voltage swing of the read bit lines may lead to significant power savings. To limit the bit line voltage, a different pre-charge circuit as shown in Figure 4.9, two NMOS transistors were used to charge the bit line. These NMOS transistors were able to charge the bit line to about 300mV during the pre-charge time for this particular test case. The intention of this simulation was to see the impact on power if the voltage swing on the bit line can be limited. Inverters cannot be used at the output stage if the bit lines do not have full voltage swing, thus the sense amplifier shown in Figure 4.10 was used as the output circuit to get a valid logic output. These simulations also helped to give an idea about the working and power of the sense amplifier.

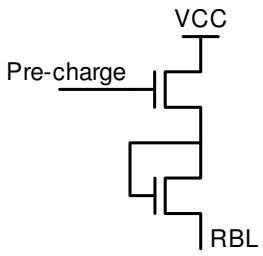


Figure 4. 9 Pre-charge circuit to limit bit line swing

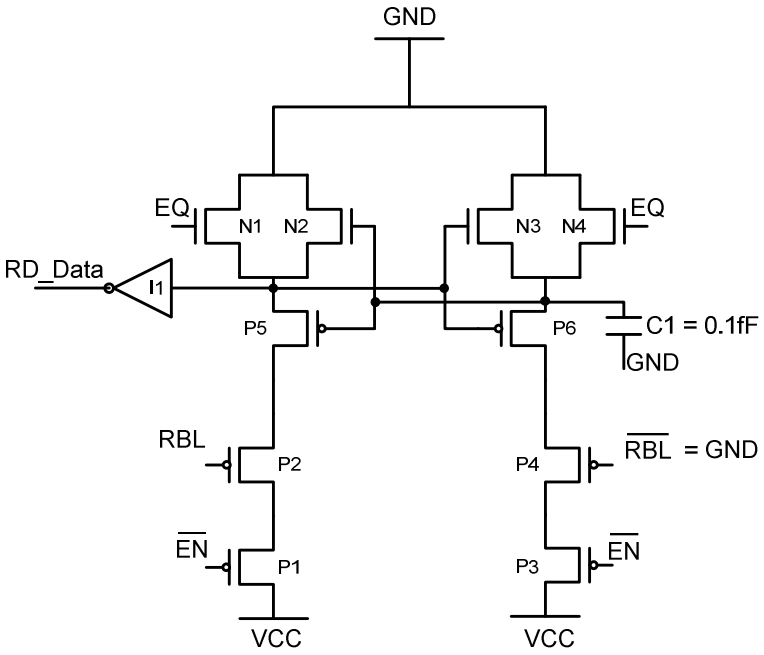


Figure 4. 10 Sense amplifier circuit

The sense amplifier design shown here is inherently meant for differential read topologies. For proper working of the sense amplifier a reference voltage of about 150 mV needs to be generated. Initially a piecewise linear voltage source was used in Eldo to generate the reference voltage. It was observed that if ground (represented as GND) is used as a reference and a C1 is added to the sense amplifier, the sense amplifier works in all but 1.95 V, -40 C, fast/slow corner. It is believed that the problem can be solved by proper sizing of transistors. The power consumed by the sense amplifier is **765 µW**. Some of the power associated with the sense amplifiers is due to the EN and EQ control lines that run across the whole register file. The read word line capacitance of 80fF is used to model the wire capacitance of these control lines.

4.2.5.1 Power analysis of sense amplifier

The power consumed by Design 1 with a reduced swing on bit lines and sense amplifiers is **2448 µW**. Sense amplifiers are very useful in RAMs where the bit line capacitance is much higher than this design. They can detect very small variations in the bit line and don't require the bit line to switch to the whole logic level. Figure 4.11 shows the power distribution of this design. The power associated with the read bit lines is reduced but the sense amplifier takes a lot of power and this nullifies the impact of power saving that was obtained while deploying reduced bit line swing.

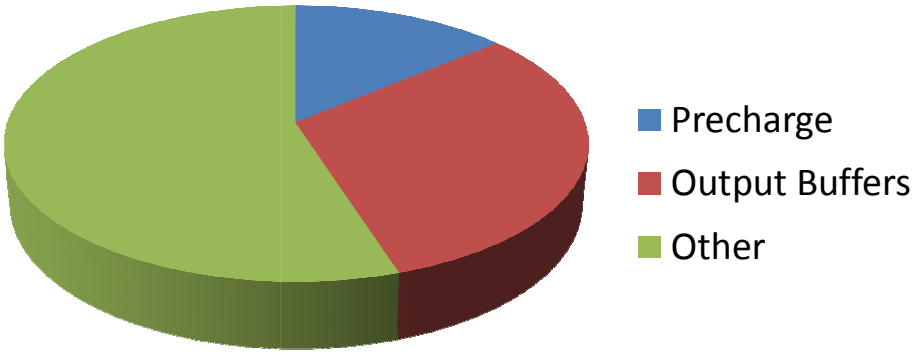


Figure 4. 11 Power analysis of Design 1 using sense amplifier

4.3 SRAM based register file - Design 2

Based on the observations from the Design 1, it was noticed that the read bit lines contribute to a significant amount of power. The motivation of Design 2 was to limit voltage on bit lines so as to save power. The main features of this design are listed below.

- Single-ended read and write topology
- Reduced bit line swing

4.3.1 Functionality

In the simulations of Design 1, it was observed that the minimum sized transistor can easily discharge the bit line during the read cycle. In Design 2, the cell drives the bit line actively through an NMOS during the read cycle. This eliminated the pre-charge circuitry and also limited the swing on the read bit lines. The cell is shown in Figure 4.12.

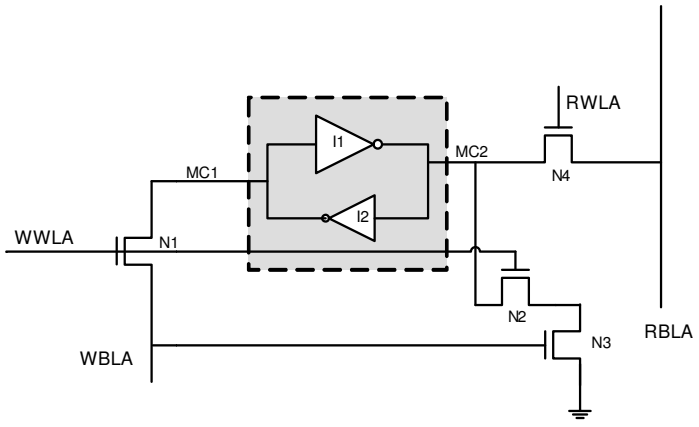


Figure 4. 12 Bit Cell for Design 2 – 1R/1W

The bit cell employs three NMOS transistors for writing and one NMOS for reading. The write topology here is the same as that of Design 1. During the read cycle, when logic '0' is saved at MC2 it is passed to the bit line. If MC2 has logic '1', a threshold voltage drop is lost and the read bit lines rise up to 1.2V only. This limits the voltage swing on the bit lines. Figure 4.13 shows the bit cell with four read ports and two write ports. This design has 14 transistors per bit cell.

As the bit cell actively drives the read bit lines, this cell structure was prone to errors related to data integrity during the read cycle. The phenomenon of destructive read is explained in the following sub section.

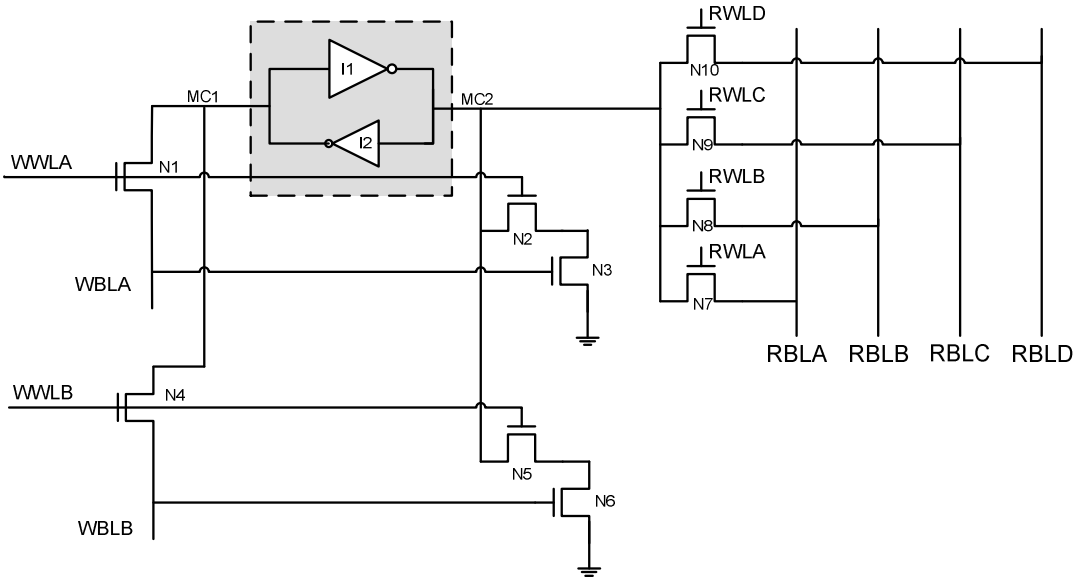


Figure 4. 13 Bit Cell for Design 2 – 4R/2W

4.3.2 Destructive read problem

To elaborate this problem, let us consider the bit cell shown in Figure 4.13. Consider the voltage on all the four bit lines is logic '0' before reading and the logic at MC2 node is '1'. When all the read ports are reading the same bit cell, the four read pass transistors are turned on and it takes some time for I1 to charge the highly capacitive bit lines. But in the meantime the voltage at MC2 node might drops significantly. This can result in flipping of MC1 node due to I2 and the value saved in the cell is overwritten. Three ways are suggested to minimize this problem.

- Increase the drive strength of I1
- Reduce the bit line capacitance
- Turn on the read pass NMOS transistor slowly by reducing the drive strength of the word line signals RLWA, RWLB, RWLC and RWLD

To resolve this issue, the drive strength of inverter I1 driving the read bit lines was increased. Later sections of the report also discuss the reduction of bit line capacitance by layout optimization and partitioning. In the actual implementation, the drive strength of I1 is three times more than the drive strength of inverter I2. After this sizing, the inverter could drive up to three read bit lines without the destructive read problem. If this design is to be considered, the hardware and software components should ensure that a maximum of three read ports read out from the same word in the same cycle.

4.3.3 Architecture

The architecture of this design is quite similar to the architecture of Design 1. The changes are listed below.

- Pre-charge is removed
- Modified output stage

In Design 1, inverters and sense amplifiers were used in the output stage. It was observed that the power dissipation of the buffers was less compared to sense amplifiers. Sense amplifiers are capable of converting a small voltage swing on the read bit lines to full swing but the high power consumption do not make them a favorable choice. If inverters are used then these will contribute to the short circuit power when a logic '1' is read as the swing on the bit lines is only up to 1.2 V in this case. Initial simulations were done using inverters at the output stage. It was observed that despite the short circuit current in the inverters, the power of this design was less than Design 1 but the short circuit power becomes more prominent when the respective read port is idle. The different circuits that were tried to find an optimal circuit for the output stage are discussed in the next sub-section.

4.3.4 Output stage

The following sub-section discusses some of the variants of keeper circuits.

4.3.4.1 Simple keeper circuit

A simple keeper circuit is shown in Figure 4.14. It was used to avoid the short circuit current in the inverter by swinging the input of the inverter to full logic levels. Although it eliminated the short circuit current, it forced the highly capacitive read bit line to swing to full logic levels thus eliminating the advantage of this particular design altogether.

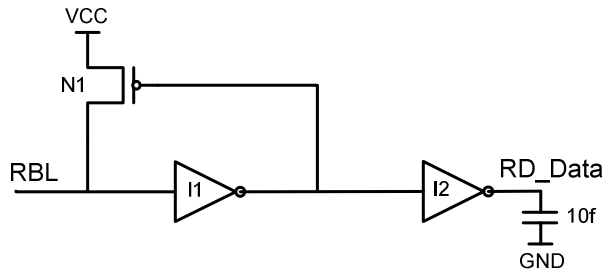


Figure 4. 14 Simple keeper circuit

4.3.4.2 Keeper with tri-state inverter

A combination of tri-state inverter with a keeper was tried to limit the bit line voltage while eliminating the short circuit current as shown in Figure 4.15. But the tri-state inverter with weak logic input is unable to overwrite the minimum sized keeper transistors P3 and N3.

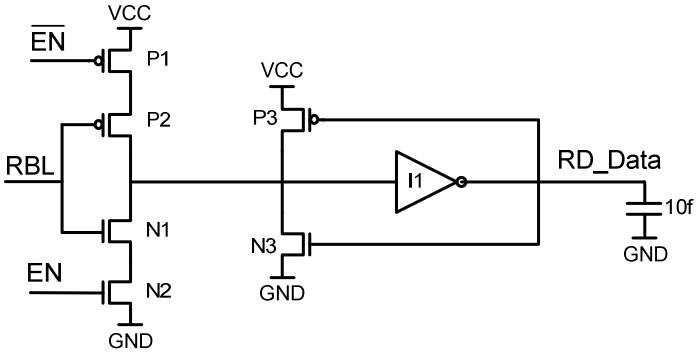


Figure 4. 15 Tri-state inverter and keeper for output stage

4.3.4.3 Keeper with enable and tri-state inverter

Figure 4.16 shows a tri-state inverter and a keeper circuit with enable. During the read cycle, the tri-state inverter is turned on and the keeper is disabled. After the read cycle, the tri-state is turned off and the keeper is turned on. This circuit limits the duration of short circuit current at the inverter to the left and also isolates the bit line voltage swing while always maintaining a proper logic level at the output. The width of NMOS of the tri-state inverter is more than the PMOS to compensate for the weak logic '1' input. One major drawback of this circuit is that it requires two control signals per read port that pass through the entire width of the register file. These wires are modelled by a capacitive load equivalent to the read word line. In total it adds eight extra control lines to the register file. A few more design alternatives for the output stage are mentioned in Appendix A.

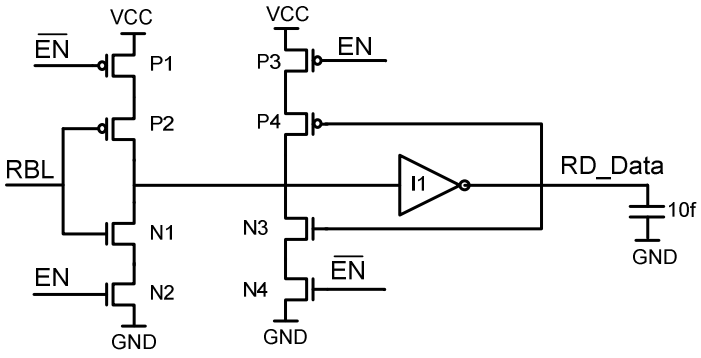


Figure 4. 16 Tri-state inverter and keeper with enable

4.3.5 Estimation of parasitics

The layout of a cell for four read ports and two write ports was made in Cadence Virtuoso. Figure 4.17 shows the layout of the cell of Design 2. The extracted values of parasitics are given below.

- Read Bit Lines = 30 fF
- Write Bit lines = 50 fF
- Read Word Lines = 105 fF
- Write Word Lines = 110 fF

It should be noted that the layout of this design has almost the same dimensions as that of Design 1. Even though it has less number of transistors, the parasitics obtained here are more than in Design 1. Apart from dimensions, the placement of transistors and wires is very important for having a minimum capacitance.

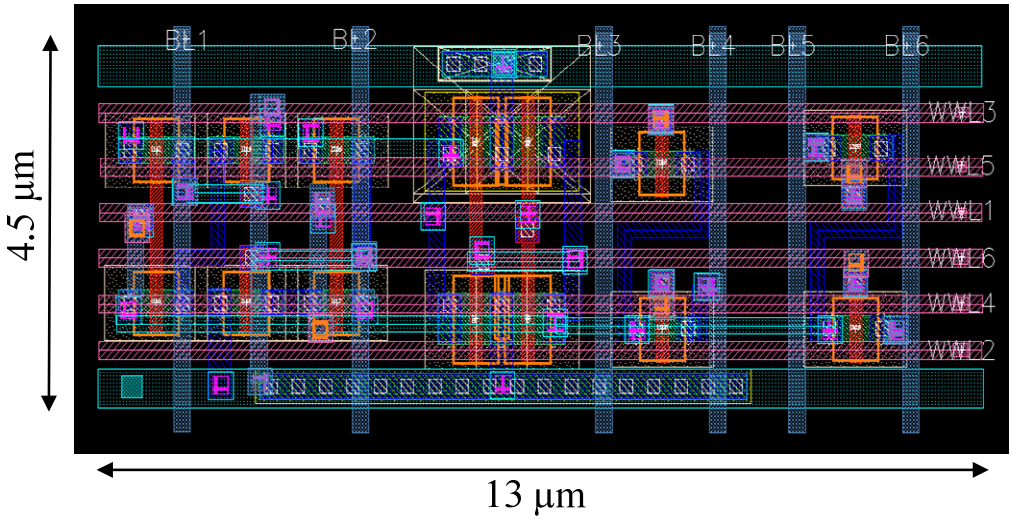


Figure 4. 17 Layout of a 4R/2W bit cell - Design 2

4.3.6 Power analysis

The power value obtained for this design is **2140 µW**. Figure 4.18 shows the power distribution. This design is a compromise between the reduced swing of 330 mV and a full VCC swing. Also the output stage is a compromise between buffers and sense amplifiers. It should be noted that the power is less than both the versions of Design 1 – with and without sense amplifiers.

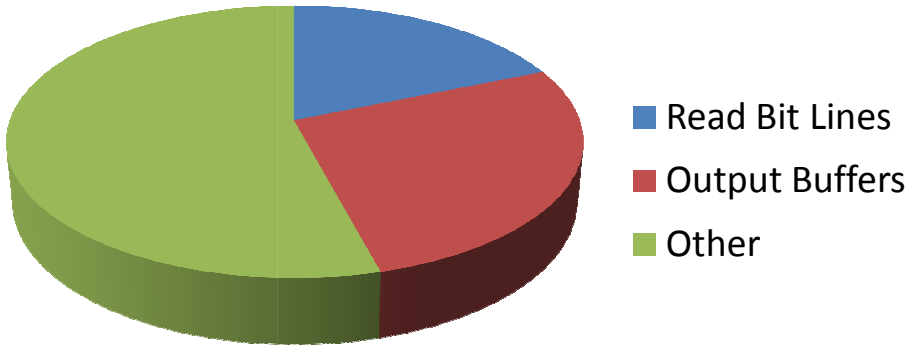


Figure 4. 18 Power Analysis of Design 2

4.4 SRAM based register file - Design 3

Figure 4.19 shows the cell structure of Design 3. Key features of this design are as listed below.

- Single-ended read and write topology
- Less transistors for write port

This cell structure only uses one pass transistor for each write port. It reduces the number of transistors inside the cell compared to the previous designs. The write structure is dependent only on N1 and this transistor has to overcome the logic driven by inverter I2. Hence, a very wide N1 is required to be able to write logic '0' or logic '1'. Moreover, NMOS is slow at passing a logic '1' which leads to short circuit power in the duration of write cycle. This increases the overall power for this cell design.

Although this cell configuration has less number of transistors per write port, this design was not considered for the register file due to the reasons mentioned above.

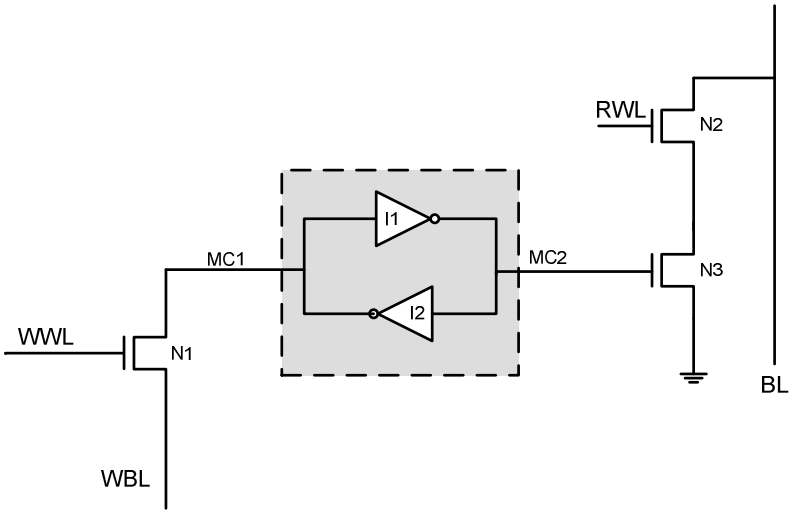


Figure 4. 19 Cell of Design 3 – 1R/1W

4.5 SRAM based register file - Design 4

The bit cell for Design 4 is as shown in Figure 4.20. Key features of this design are listed below.

- One bit line per read port and write port
- One word line per read port and two word lines per write port

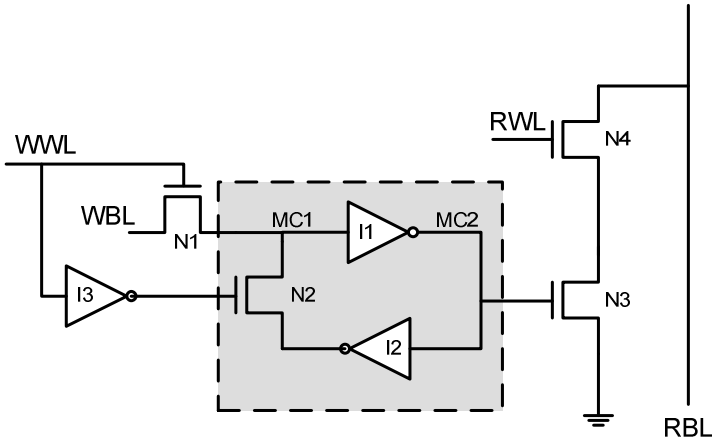


Figure 4. 20 Cell of Design 4 – 1R/1W

Inverter I3 provides the inverse of the write word line. One inverter can be used for one row of bit cell i.e., one memory word. The write topology here uses only one pass transistor N1, similar to the previous design. But, during the write cycle, inverter I2 is disconnected from the feedback loop by turning off N2 which makes the writing into the bit cell faster.

For a four read / two write configuration, inverter I3 would be replaced by a NAND gate. The number of transistors per cell in this design is less but this design has some major issues listed below.

- During the write cycle, there is a threshold voltage drop across N1 and it leads to short circuit power in inverter I1.
- After the write cycle, there is always a threshold voltage drop across N2 inside the cell which leads to short circuit power in inverter I1.
- The number of word lines is more in comparison with other designs discussed in this chapter. This will lead to an increase in the number of high capacitance lines and thereby an increase in power associated with the word lines.
- To eliminate the short circuit current in inverter I1, N1 and N2 were replaced with transmission gates. This increased wiring and number of transistors inside the cell. Moreover, with the addition of transmission gates inside the cell, voltage spikes were observed at the MC1 node. These spikes hamper the integrity of the data stored in the bit cell.

Due to the above mentioned problems, this cell design was not considered for further simulations.

4.6 Conclusion

Considering the various designs discussed in this chapter, Design 2 was found to be the one with the least power dissipation. Hence this design was pursued for further optimization techniques described in Chapter 5.

5. Design techniques for optimization

In the previous chapter different cell topologies were discussed. This chapter explores different architectural optimization techniques that can be done to make the register file a low power, less area unit. Time multiplexing of read and write ports and partitioning of the register file would be discussed in this chapter. These techniques have been implemented on the best design from Chapter 4, that is Design 2.

5.1 Time multiplexed register file

Time multiplexing is a design technique in which the bit cell has fewer ports than the register file. As mentioned earlier, in this thesis the target register file has four read ports and two write ports. In a fully time multiplexed register file, the bit cell would have only two read ports and one write port while the register file will still have four read ports and two write ports externally. This is achieved by making the internal cell array to work at twice the frequency of the system clock.

5.1.1 Necessity for time multiplexing

The initial study of the existing time multiplexed memories suggested that time multiplexed design reduces the area of the cell array while the switching is almost the same as that of a non-time multiplexed design. There is an area and power overhead for the extra control circuit though. The extra control circuit and some sizing adjustment of transistors due to higher frequency may add to extra power.

It was decided to use the time multiplexed register file owing to the factors and observations mentioned below.

- It was observed that parasitics contribute to a lot of power of the register file. Reducing the number of ports inside the cell will reduce the overall cell size by almost 50% and thereby reducing the size of the memory cell array. This will lead to shorter bit lines and word lines.
- The number of bit lines and word lines will be reduced to half. This will reduce the coupled capacitance of the wires associated with neighbouring wires of the same metal layer. It will also be helpful to reduce cross talk.
- The huge capacitance on the read bit lines is one of the factors responsible for destructive read problem. Reducing the bit line capacitance would help in solving this problem.
- During the previous simulations, the read transistors were turned on only for 4 ns in the time period of 10ns. There was a lot of idle time during each clock cycle. So it was assumed that operating the cell array at twice the frequency would not require any resizing of transistors inside the cell array.

5.2 Read time multiplexed design

In the first phase of time multiplexing, only the read ports were time multiplexed.

5.2.1 Bit cell

In this configuration, the bit cell has two read ports and two write ports as shown in Figure 5.1.

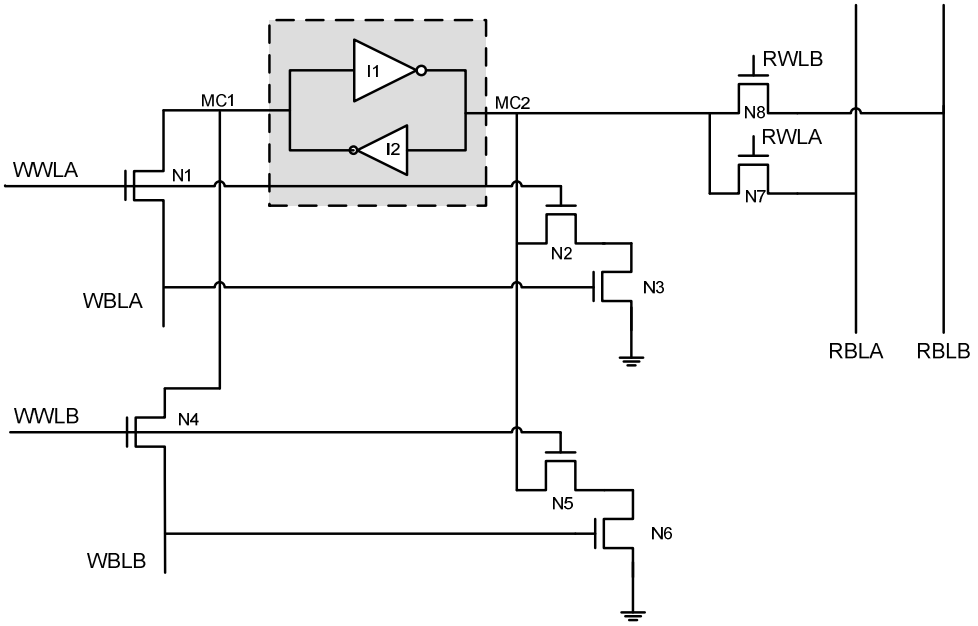


Figure 5. 1 Bit cell of read time multiplexed design

5.2.2 Layout

To estimate the bit line and word line parasitics for this design, a layout was designed as shown in Figure 5.2. In essence, two read ports, that is two pass NMOS transistors have been removed. The values of the parasitics extracted from this layout are listed below.

- Read Bit Lines = 30 fF
- Write Bit lines = 50 fF
- Read Word Lines = 55 fF
- Write Word Lines = 75 fF

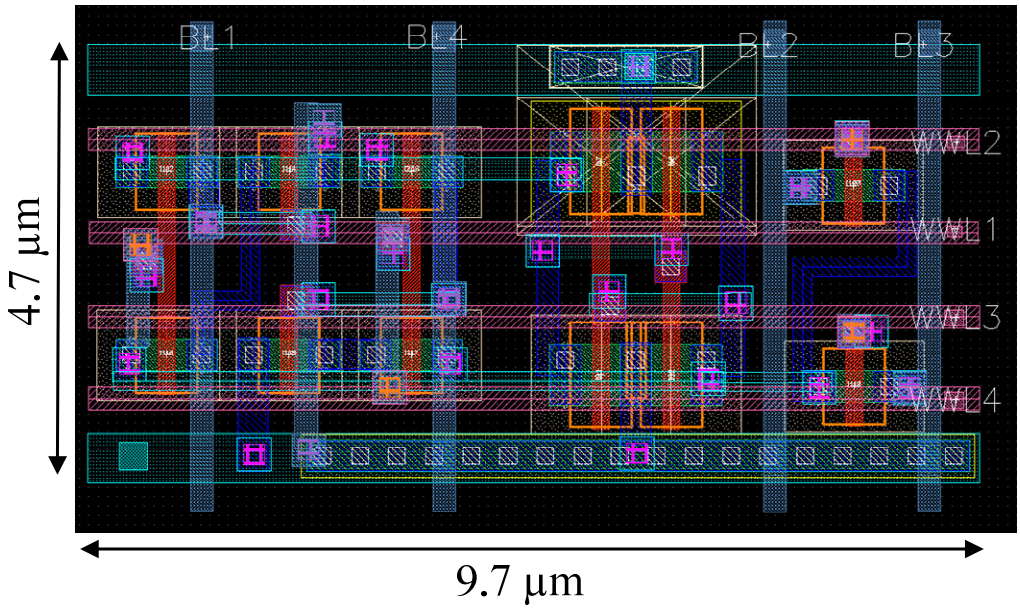


Figure 5. 2 Layout of a read time multiplexed (2R/2W) bit cell

5.2.3 Functionality

This design has two read ports working at twice the frequency of the clock. Each read port effectively reads twice in a clock period of 10ns. The read port A of the bit cell (as in Figure 5.1) reads out the data on external read port A during the first 5ns while in the second half of the cycle, it reads the data out on external read port C from the corresponding port addresses. Similarly, the read port B of the bit cell reads out the data on external read port B during the first 5ns while in the second half of the cycle it reads data on external read port D from the corresponding port addresses. The design still uses four read decoders and the enable signals of the decoders are controlled using the control circuitry to produce the read word line signals as explained before. An alternate would be to use two decoders and multiplex the read address bits going to the decoder.

The data that is read in the first half of the cycle is latched until the next read by that port, using the keeper structure discussed in Section 4.3.4.3. The data read in the second half of the clock pulse is latched as well until another read is performed by the corresponding port. Figure 5.3 gives an idea as to how read time multiplexing is achieved for the test case mentioned in Section 3.4. The read word line signals for two different words are shown as plots in green and blue. It can be seen that two words are read in one clock cycle. Read bit line is represented by the yellow waveform and has a voltage swing between 0 and 1.2V. At the output of the ports A and C, the data has a proper logic level as shown by the plots which are orange in colour.

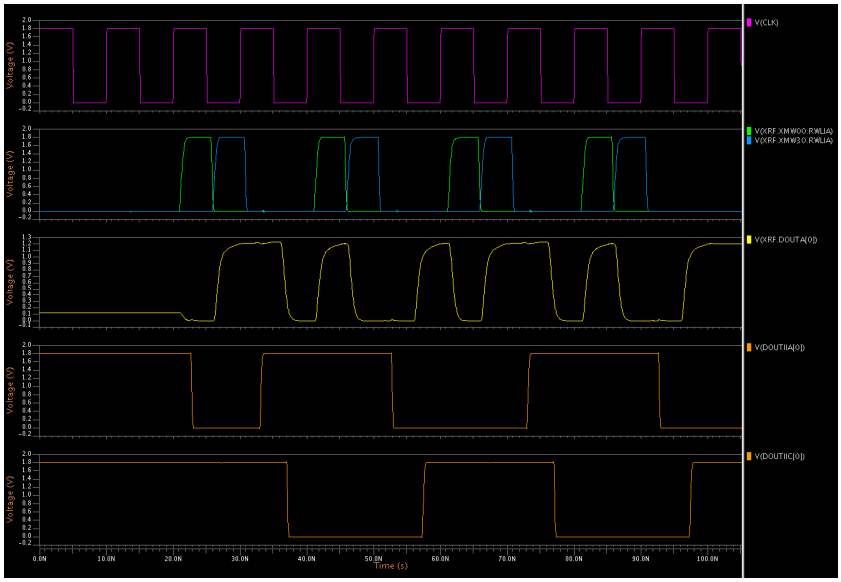


Figure 5. 3 Waveforms for read time multiplexing

5.2.4 Design challenges

The output structure discussed in section 4.3 is used in this design as well. This circuit has been adapted to function as a de-multiplexer so as to perform read time multiplexing. Introduction of the de-multiplexer led to unnecessary switching on the output ports. This was removed by controlling the enable signals of the output circuit.

5.2.5 Power analysis

The reduction in power comes from the fact that only two read ports do the necessary function of four read ports, thereby reducing the number of transistors in a bit cell and thus influencing in the reduction of word line capacitance. Also, the number of read bit lines is reduced from four to two, thus reducing the coupled capacitance on these lines. Reduced capacitance on the wires decreases the power consumed by parasitics. But there is also a slight increase in the power consumed by logic due to extra control signals. The power of this design is **1963 µW**.

5.3 Read/write time multiplexed design

The design from Section 5.2 was carried forward to time-multiplex the write ports as well.

5.3.1 Bit cell

The bit cell for design with read/write time multiplexing is as shown in Figure 5.4. Here, the bit cell has one write port and two read ports.

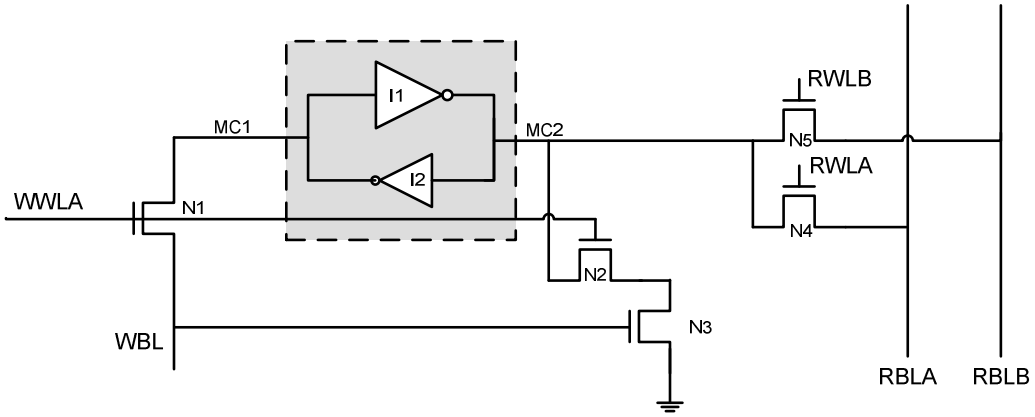


Figure 5. 4 Cell of read/write time multiplexed design

5.3.2 Layout

The layout of this cell is shown in Figure 5.5. The extracted parasitics for this layout are given below.

- Read Bit Lines = 30 fF
- Write Bit lines = 35fF
- Read Word Lines = 35 fF
- Write Word Lines = 40 fF

Since we eliminate one more bit line, there is a noticeable change on the write bit line capacitance. This is mainly due to the fact that the bit lines are sparse. As there are less number of transistors in the bit cell, the cells dimensions are much smaller as a result of which the word lines also have reduced capacitance values.

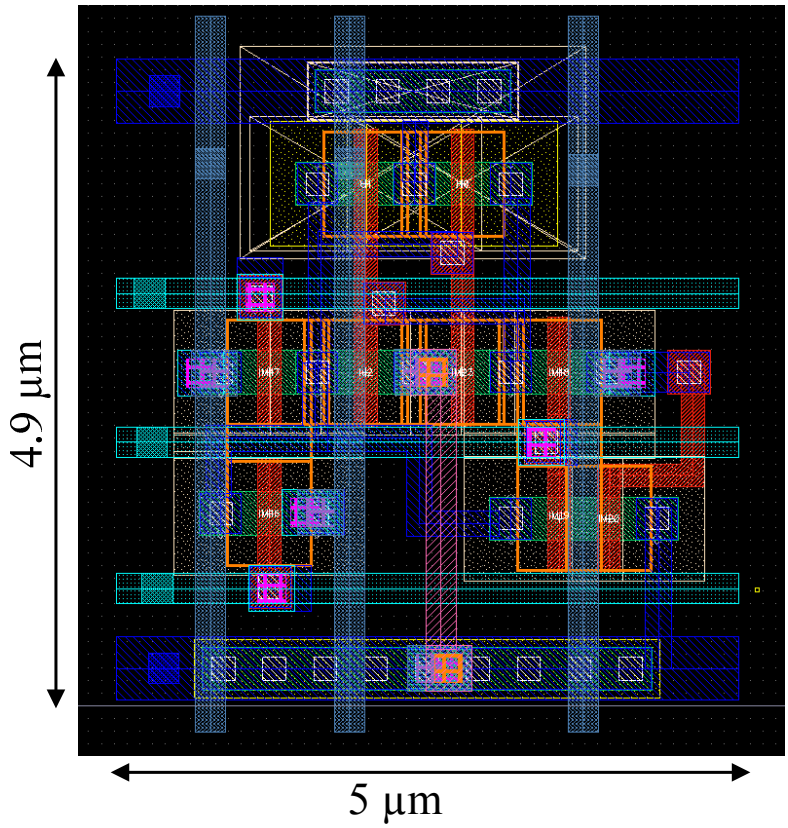


Figure 5. 5 Layout of a read/write time multiplexed (2R/1W) bit cell

5.3.3 Functionality

This design has a single write port, working at double the frequency of the system clock, which performs the functionality of two write ports. The read structure is the same as in Design 1. The data at write port A is written in the first half of the clock cycle while in the second half, the data at port B is written. A multiplexer, the design of which is discussed in detail in Section 5.3.4, is used to put the correct data on the bit lines. This design has two write address decoders. The word lines, which are signals coming out of the decoders are multiplexed.

The plots shown in Figure 5.6 give an indication as to how time multiplexing is handled with respect to a write port. The words ‘00’ and ‘30’ are written in the same clock cycle as the write word lines for these words indicate (shown by yellow and orange waveforms). Likewise, the words ‘01’ and ‘31’ are written in the same clock cycle (shown by blue and pink waveforms).

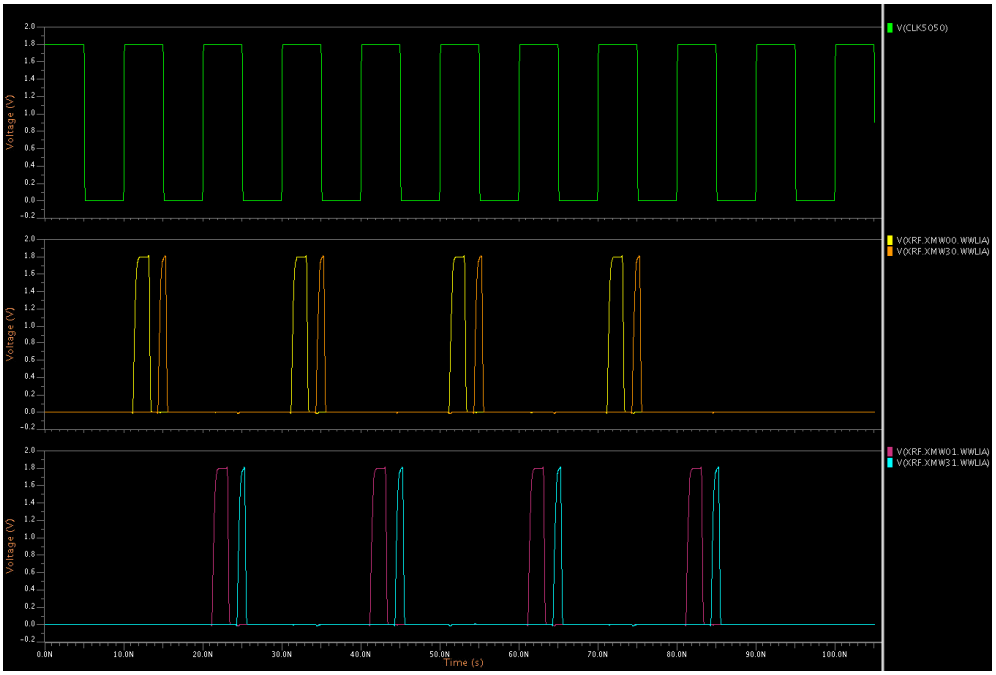


Figure 5. 6 Waveforms for read time multiplexing

5.3.4 Design challenges

The major challenge in this design was to ensure that the voltage is stable at the write bit lines before the write word line is asserted.

The discussion on the functioning of multiplexers used gives an insight into the problems that could likely occur. Initially the multiplexer shown in Figure 5.7 was used for every bit of input data from the write port. DINA and DINB are input data bits that are to be written. MUXENA and MUXENB signals are generated by the control circuit. WBL is the write bit line of the register file. At the cross-over point of the MUX enable signals, MUXENA and MUXENB, a voltage glitch is observed on the output of the MUX and causes the write bit line to toggle abruptly. At this point in time, the AND gate with MUXENA as input is disabled and while the AND gate with input MUXENB is enabled. The voltage glitch is as noticed in the Figure 5.8 (green waveform). The voltage glitch on the write bit line not only increases the power of the register file, but might also end up writing the wrong value to the bit cell if it occurs during the time when write word line is asserted. This is likely to occur during the slow process corners.

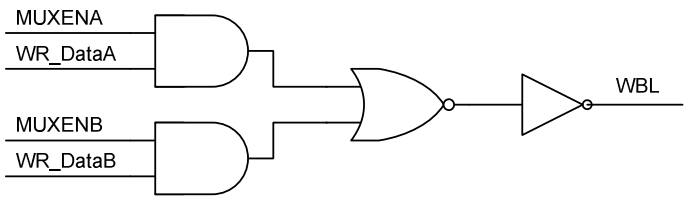


Figure 5. 7 Multiplexer using standard cells

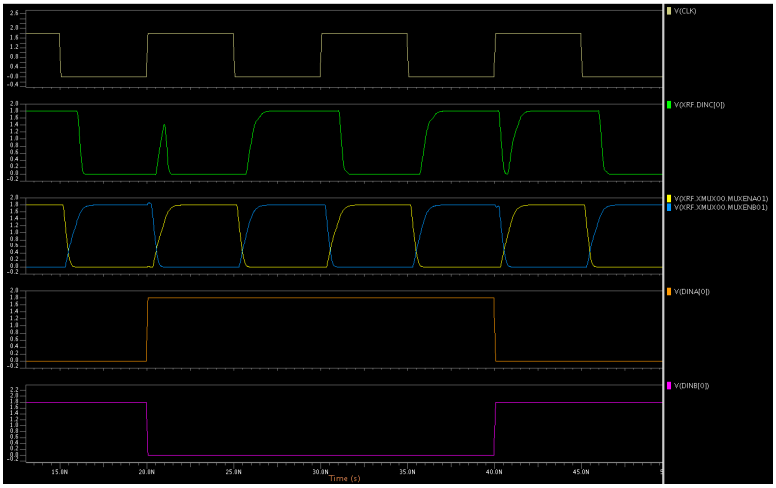


Figure 5. 8 Waveforms showing the voltage glitch on write bit line

This necessitated a change in the multiplexer structure and the control signals. The voltage glitch was sorted out using the multiplexer structure shown in Figure 5.9.

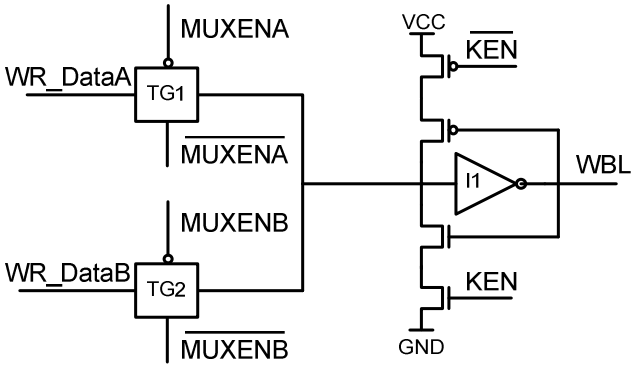


Figure 5. 9 MUX structure to eliminate voltage glitch on write bit lines

This circuit enables TG1 during the first 2.5ns of the positive edge of clock and enables TG2 during the first 2.5ns of the negative edge of the clock. KEN is enabled whenever TG1 and TG2 are disabled. This circuit avoids the glitch at the write bit line but at a cost of six extra control lines. These control lines span the entire length of the register file and thus have a capacitance equal to that of the word lines on them. These six active word lines capacitances switching during a write cycle contribute to a lot of power.

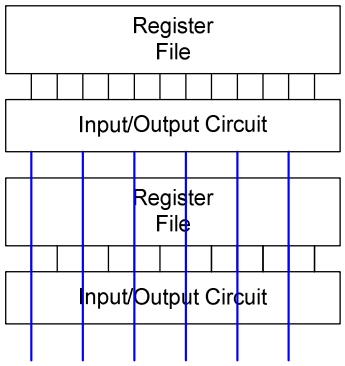
5.3.5 Power analysis

The advantage of this design is the reduced write bit line capacitance and the reduced word line capacitances owing to the small bit cell size. However, this advantage is negated by the addition of extra logic, which is the multiplexer structure with six control signals, at the write bit line. Because of the addition of multiplexer logic, this design consumes slightly more power than the read time multiplexed version. The power of this design is **2020 µW**.

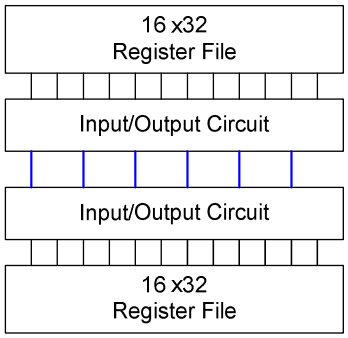
5.4 Partitioning

A partitioned register file is essentially two 16x32 register files. Partitioning of the register file helps to reduce the length of the bit lines and thereby reducing the power associated with parasitic capacitance. The word line capacitances still remain the same though. Reducing the bit line capacitance also helps to solve the destructive read problem.

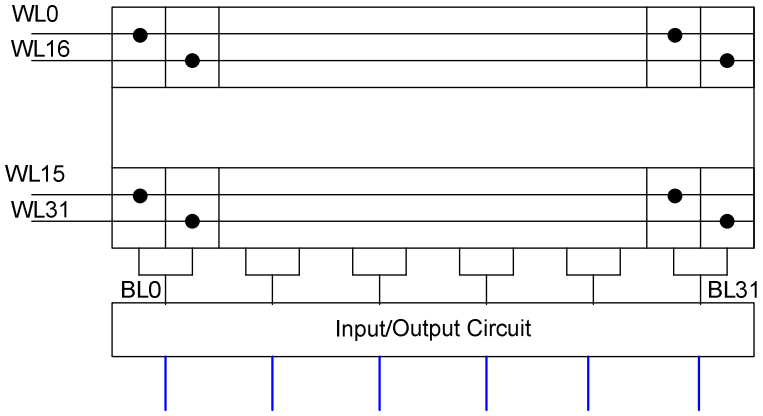
5.4.1 Configurations for partitioning



(a) Configuration 1



(b) Configuration 2



(c) Configuration 3

Figure 5. 10 Design configurations for partitioning are shown in (a), (b) and (c)

The partitioning of a register file can be done in several possible ways based on where the input and output pins can be placed. A few possible architectures are as shown in Figure 5.10.

5.4.1.1 Configuration 1

In this configuration, the entire register file is split into two and arranged on top of each other. The bit lines are reduced by half. But if all the output pins of the read ports are

required at one end, the bit lines of the top half of the register file have to span across the entire length. The only saving in this configuration is in the bit lines of the bottom half of the register file.

5.4.1.2 Configuration 2

In configuration 1, if the lower half of the register file is flipped horizontally then all input/output pins would be in the centre. In this way the bit lines are reduced to half. The problem in this case would be that we have 128 output connections, 32 from each read port, in addition to the input connections as well. Extra wiring and routing in center may cause routing congestion and increase the power for this configuration globally. Considering that the Place and Route tool will be able to manage routing this many number of pins in an efficient manners, this configuration seems to have the best advantage.

5.4.1.3 Configuration 3

In this design configuration, the two halves of the register file, top and bottom ones are interleaved. This reduces the length of the bit lines by half, but at the same time increases the word line length by twice.

5.4.2 Selection of design alternative

Based on the discussion above, configuration 2 was selected for partitioning for the obvious reason that the bit lines of all ports could be reduced without having to compensate for anything extra, unlike the other two configurations.

The design with read time multiplexing has lower power than the read/write time multiplexed version. So naturally it was considered for partitioning. But also considering the fewer number of transistors in the read/write time multiplexed design and optimizations that were possible in the layout, both these designs were pursued for partitioning.

5.5 Partitioning of register file

Before heading to partitioning the register file, estimates were made to find the expected power for these designs after partitioning. The bit line capacitance was reduced to half its original value and 16 words, which were not accessed for reading or writing by the test stimuli, were removed from the SPICE transistor netlist. This is similar to simulating a 16x32 register file with the same test vectors as before. The extra logic overhead was ignored for the estimation of power. These projected values have been compared with the obtained power dissipation values in Figure 5.11.

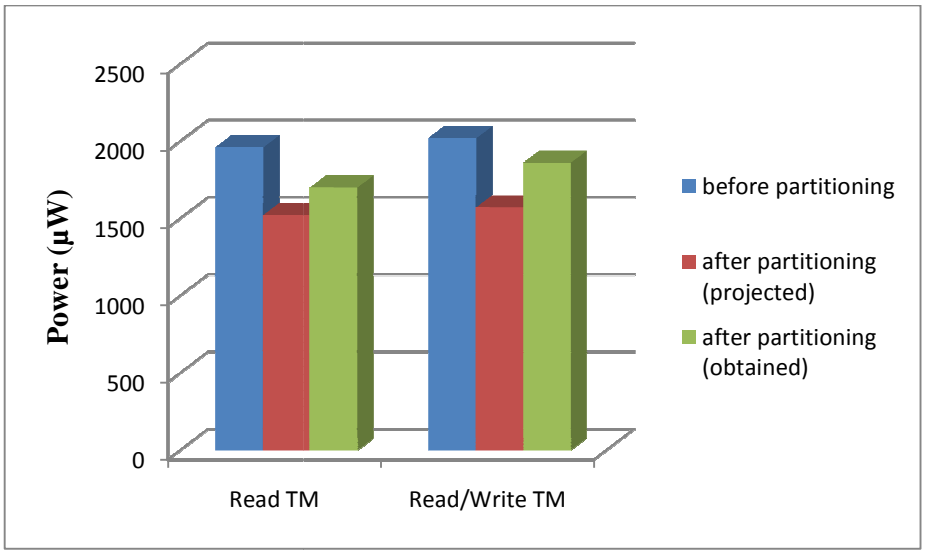


Figure 5. 11 Comparing projected and obtained power values for partitioning

After partitioning, the power of the read time multiplexed design is **1700 µW** and that of the read/write time multiplexed design is **1845 µW**. It has to be noted that the number of transistors in a bit cell of the read/write time multiplexed design is less compared to the other design.

5.6 Overview of the register file design

Figure 5.12 shows the memory cell of the final design that was simulated. After partitioning the bit line capacitance decreased significantly. The problem of destructive read was minimized due to less read bit line capacitance and less number of physical read ports for each cell. In the final cell design, all transistors are minimum sized except for P1 and N3. P1 is slightly bigger to provide some margin for destructive read. N5 is slightly larger to facilitate the writing of logic '0' to the cell across all PVT corners.

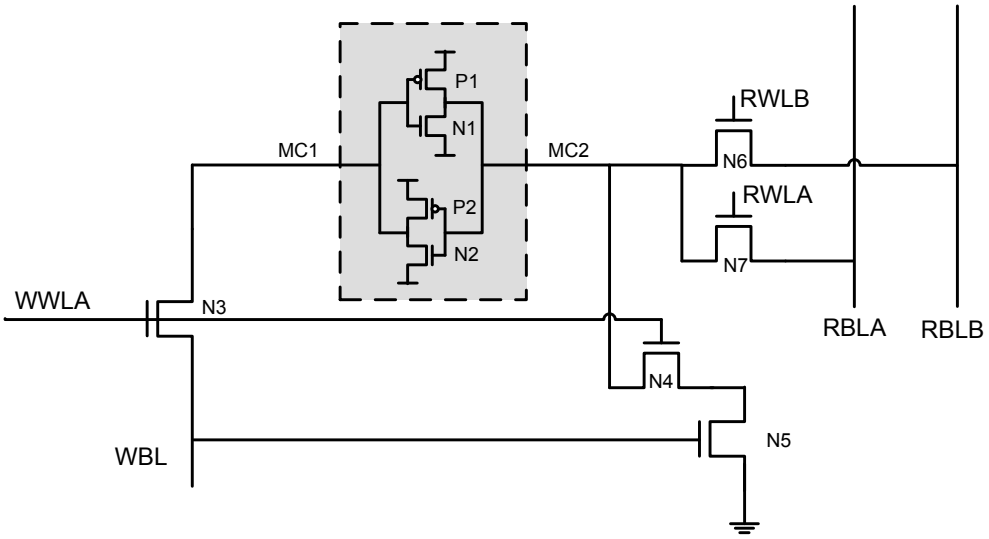


Figure 5. 12 Memory cell of the final design

Figure 5.13 shows the input stage for read/write time multiplexed and partitioned design. WR_DataA and WR_DataB hold the value of data to be written into the memory word. Since the concept of time multiplexing is used, there is only one physical write port which performs the function of two write ports. The period of 10ns is split into two halves of 5ns each. In the first 5ns, the data on WR_DataA is written while in the second half, data at the port WR_DataB is written. This multiplexing of data from two ports to one port, called DIN, is performed by the logic marked as 'MUX for TM'. As the entire register file is partitioned, there are two write bit lines, one of which accesses words 0-15 while the other accesses words 16-31. The data on DIN has to be moved to the corresponding bit line based on the write address of port A or port B depending on whether the write is happening in the first or the second half of the clock cycle. This is performed by the logic 'DE-MUX for partitioning'. The data on DINC is thus split into DIN1I and DIN2I nodes, which are the write bit lines.

Figure 5.14 shows the logic involved in the output stage. DOUTA1 and DOUTA2 are the read bit lines coming from the two halves of the partitioned register file. As the swing on the read bit lines is 0V-1.2V, only an NMOS has been used as a multiplexer instead of TG for partitioning. This multiplexer combines the two halves of the read bit lines. This node, DOUTA is then given to the output circuit of the ports A and C where the data having a proper logic level is eventually read out. The total power consumed by this design is **1840 µW**.

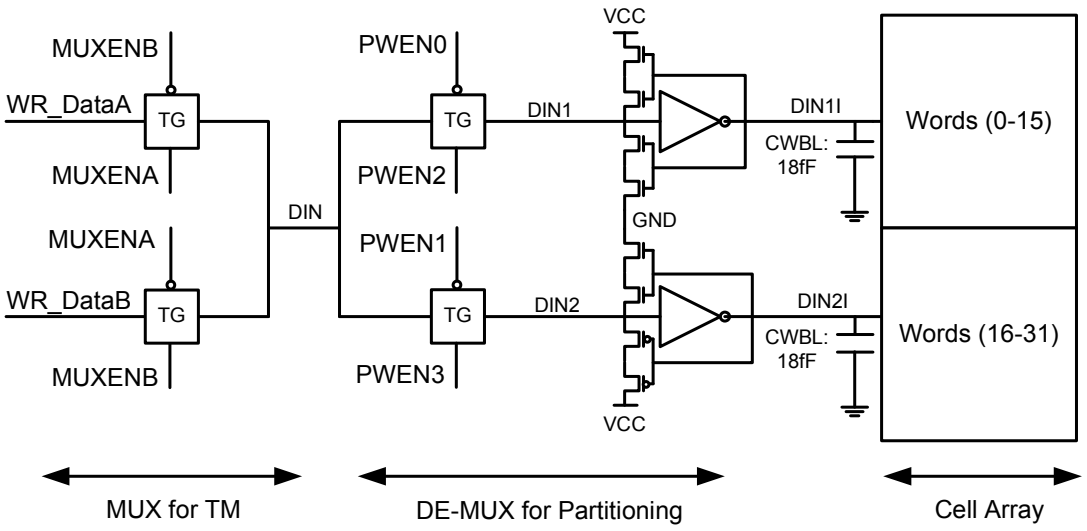


Figure 5. 13 Input stage final design

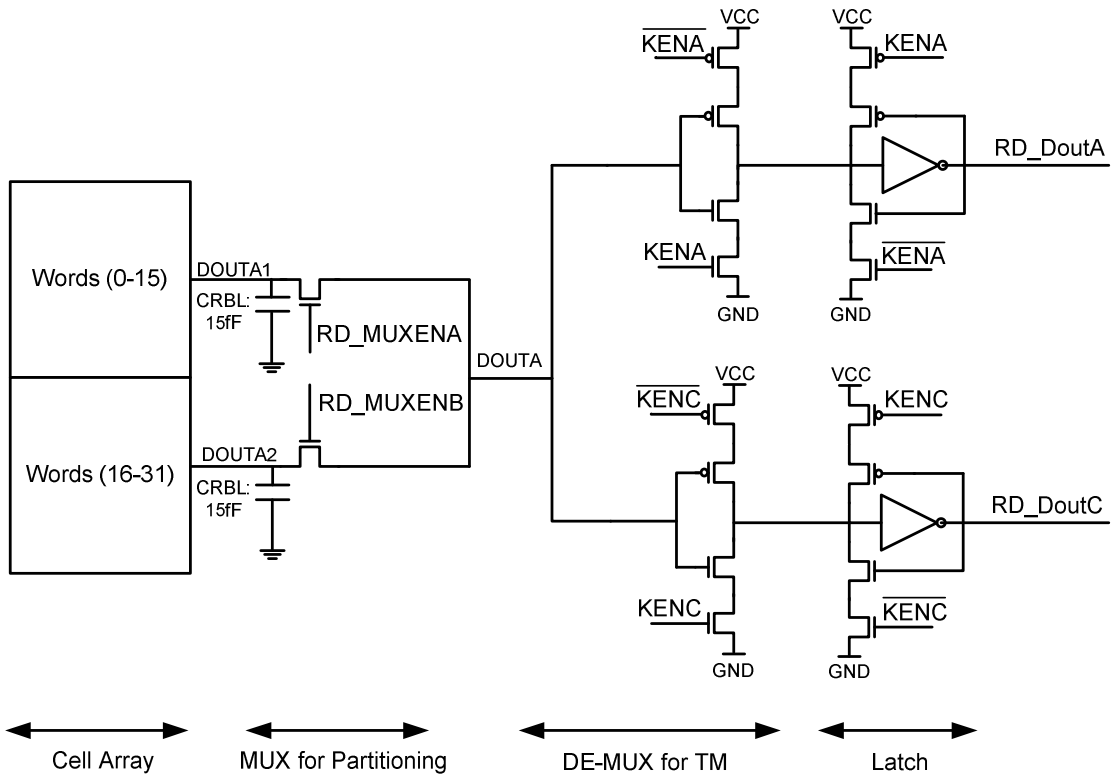


Figure 5. 14 Output stage of the final design

5.7 Simulation with extracted netlist of memory core

Until now all the power values were obtained by simulating the SPICE netlist of register file and the wire parasitics were modelled by adding capacitors in the SPICE transistor netlist. To verify the completeness of the SPICE transistor netlist, the layout of a

32x32 cell array was made for the design discussed in Section 5.6. Figure 5.5 shows the cell layout for this design. The transistor sizes in the layout and schematic are identical. The complete 32x32 partitioned register file memory array is shown in Figure 5.15. Pins were added to all the bit lines and word lines and the SPICE netlist of memory array was extracted with parasitics using Cadense Virtuoso. The SPICE netlist of the memory core was extracted and this replaced the hand written memory core in the SPICE transistor netlist. This was then simulated with the logic surrounding the core to verify that the power value obtained in the previous sections is based on the right assumptions. 'Decoupled C' extraction of the memory core was tested at all PVT conditions specified in Section 3.2 and the functionality was verified. The power value for this design was 1830 µW which is very close to 1840 µW - obtained from the SPICE transistor netlist only. This simulation verified that the assumptions made about the parasitics were quite accurate. To further verify the functionality and timing implications, the memory core was also extracted using 'Coupled C' and 'Decoupled RC' options. The register file was tested with both of these netlists at 1.8V, 25 C, nominal corner and the functionality was verified. There was no significant power change with these simulations.

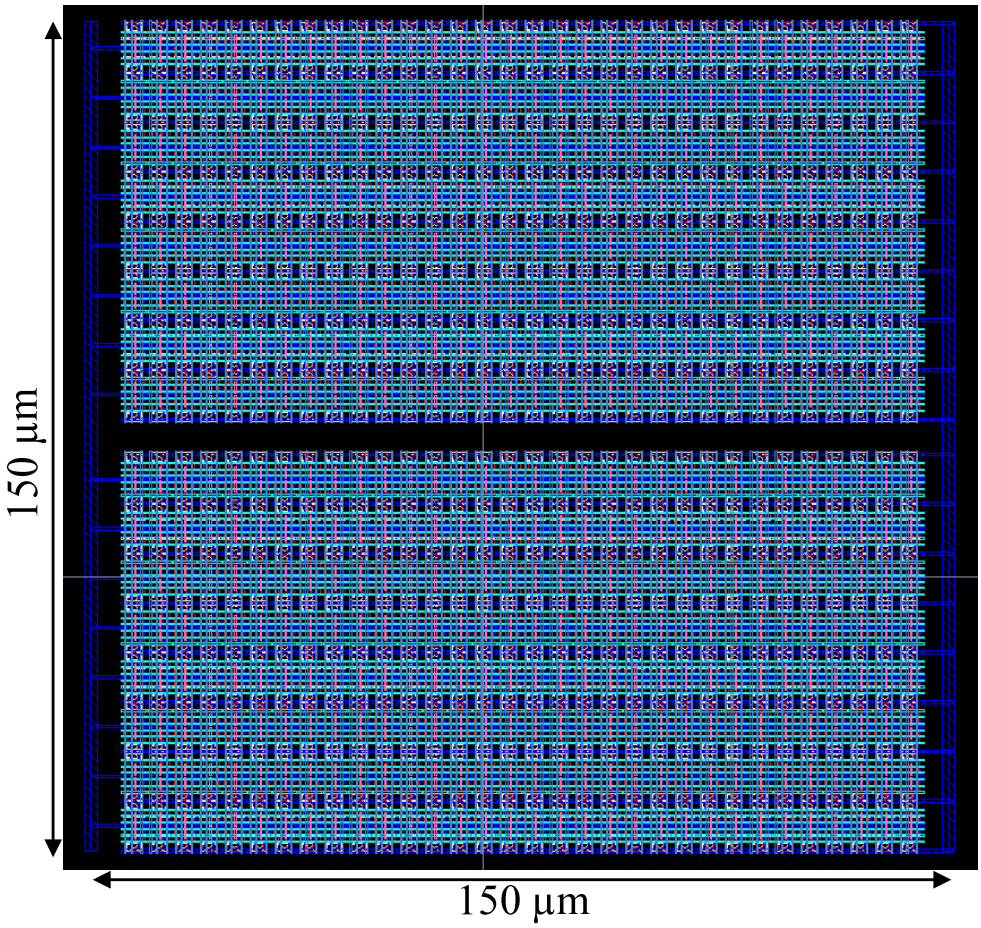


Figure 5. 15 Layout of partitioned register file

To evaluate the impact of RC delay on the functionality of register file the resistance of write word line was extracted and its value was approximately 51 ohm. For the layout specified in Section 5.3.2 the write word line capacitance is 35fF. The RC delay on the word line would be around 1.78ps. This delay would have no impact on timing or power for the register file.

6. Summary and results

A full custom register file has been designed, implemented and verified with area and power optimizations in mind. This chapter summarizes the area and power of all the designs that have been dealt with in the previous chapters.

6.1 Area

The area of an entire register file is expressed in terms of the number of elements in the designs. As only the layout of the memory core was designed, only the memory core could be expressed in terms of μm^2 leaving out the rest of the circuitry around the core. As this would not give a full estimate about the area, the number of elements was chosen as a better yardstick for comparison.

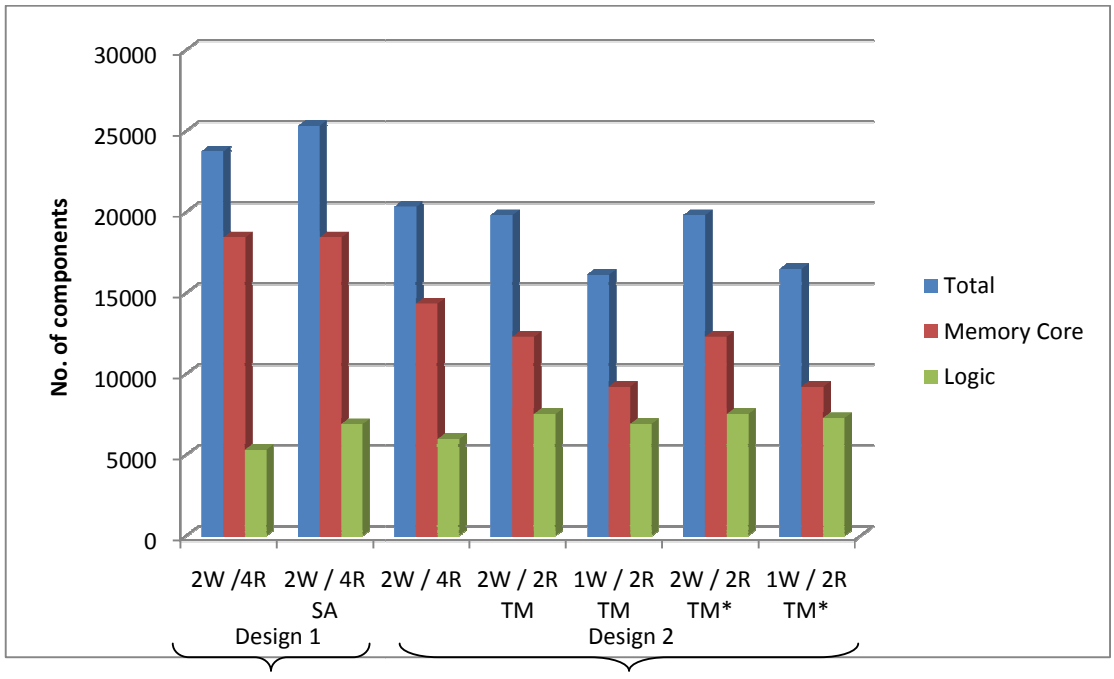


Figure 6. 1 Comparison of number of elements for different architectures.

The graph shown in Figure 6.1 gives an estimate of the number of components in Designs 1 and 2 in accordance with Chapter 4. The bar in blue color gives the total number of elements while the two bars next to it give a split up of the number of elements in the memory core and the logic around it respectively. Design 1 has two implementations, one with a full swing on the bit lines while the other being reduced swing on the bit lines with sense amplifiers denoted by SA in the graph. As anticipated, the memory core for both of them is the same but the second implementation has more logic and so the number of components is more. Moving further to Design 2, we notice a reduction in the memory core

which is because the number of transistors per read port has been reduced. Further on, with time multiplexed designs, the number of read/write ports reduces and so the memory core further decreases while the logic increases because of increasing complexity of input and output stages required to implement these techniques. In partioned designs, the number of elements in the core are the same as in the respective time multiplexed versions. This is because partitioning only adds extra logic for control while the memory core is unchanged.

6.2 Power

Like area, the power has also been summarized in the same manner as shown in the Figure 6.2. The power dissipated is split up into power dissipation due to parasitics and power dissipation due to logic for different architectures.

As with Design 1, we notice that more than half the power is contributed by the parasitics. This is mainly because of the huge bit cell structure. In the next implementation of the same design, where the swing on the read bit lines is reduced (0-300mV), the power due to parasitics reduces significantly while at the same time, the power due to logic increases

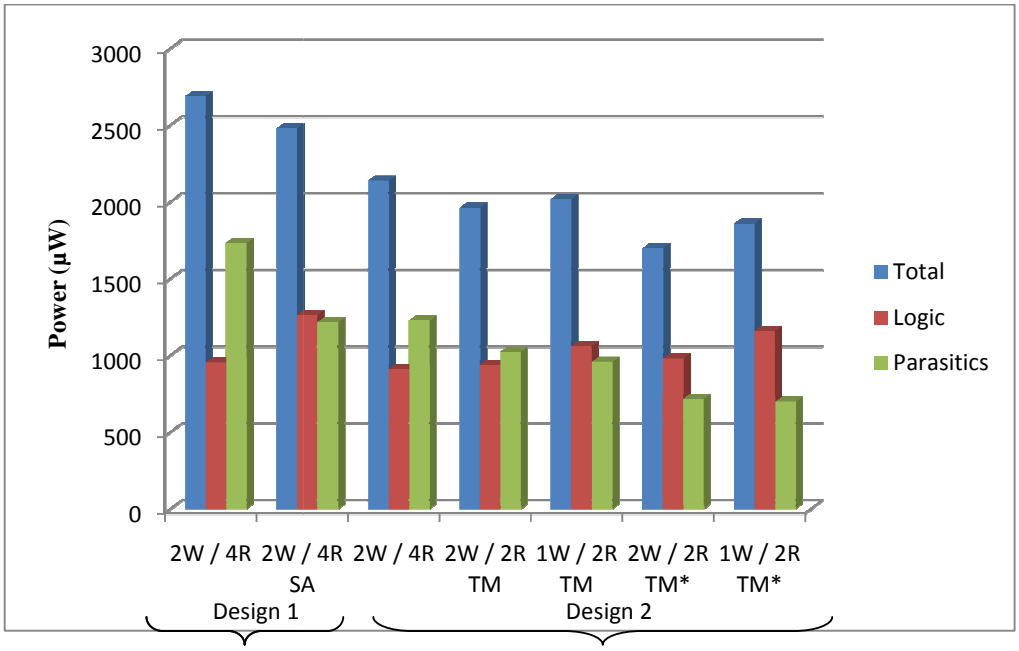


Figure 6. 2 Power split up for logic and parasitic for different architectures.

due to the sense amplifiers in the design. In Design 2, the bit lines have reduced voltage swing (0-1.2V) which reduces the effect of parasitics on power while at the same time the output circuit adds to the power dissipated by the logic. With the implementation of time multiplexing on Design 2, the parasitics power reduces further while the power due to logic increases. In the partitioned implementations it is observed that the power dissipated by logic elements is more than half the total power disspation while the portion dissipated by

parasitics has greatly reduced. This is in total contrast with the first implementation of Design 1. In essence, it is very important to find a balance between power consumed by logic and parasitics.

6.3 Discussion

The two versions - read time multiplexed with partitioning and read/write time multiplexed with partitioning, mentioned in Section 5.5, can be chosen as the best design alternatives. There is a trade-off between power and area to pick up one of these as the best implementation. With power as a yardstick, the read time multiplexed design is a better choice. If area is an important consideration, then the read/write time multiplexed design would be the more viable option. Also, another consideration should be the number of transistors per bit cell. The lesser the number of transistors, the more compact is the layout. A compact layout directly translates to shorter wires and less power dissipation due to parasitics. As the read time multiplexed version has a higher number of transistors than the other version, optimizations in layout to reduce the parasitics are limited. On the other hand, the layout of the read/write time multiplexed design has more room for improvement.

7. Conclusion

The comparison of power and area numbers of a full custom, read/write time multiplexed, partitioned register file and standard cell implementation are as shown in the Figure 7.1. The power of the standard cell register file design is **2850 µW** while that of a full custom register file is **1840 µW** which is a 37% reduction.

As discussed earlier, the full custom design layout was made only for the memory core, so the total area has to be estimated. The total number of elements in the full custom, read/write time multiplexed, partitioned design is 16486 out of which 9216 belong to the memory core. The area of the memory core is 22500 μm^2. It is assumed that each of the remaining 7270 elements also takes the same amount of area as those of the memory core. Considering this assumption, the total area of the full custom register file is **45000 μm^2**. The estimated area of the full custom register file is 68% less than the standard cell register file.

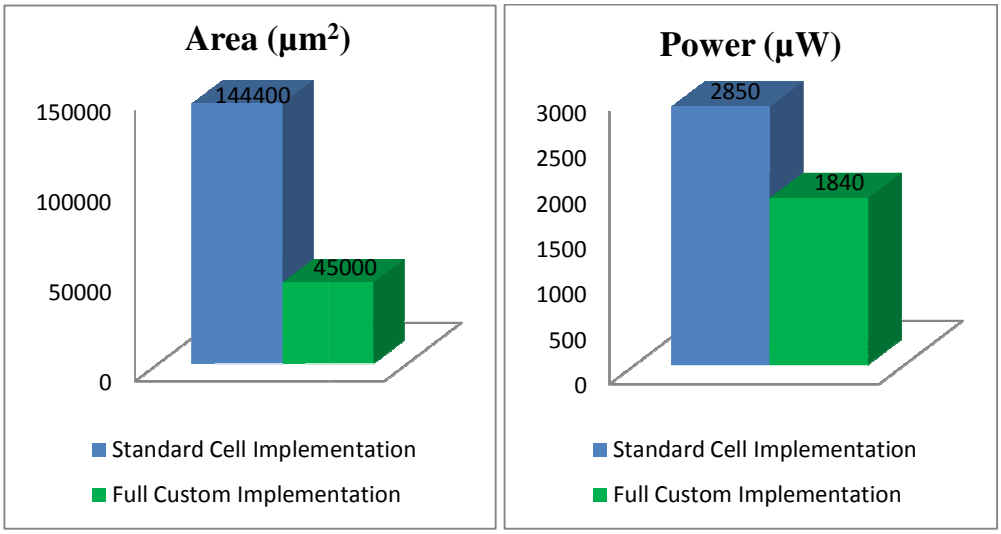


Figure 7. 1 Area and power comparison of a standard cell implementation with a full custom implementation

7.1 Future work

The following tasks are proposed for further optimizations and to get low power and less area values.

- **Data Forwarding and write protection handling**

In case the same memory word is accessed for both writing and reading in the same cycle, the output would be unpredictable. In this thesis it has been assumed that this situation would not occur. If data forwarding is necessary a separate hardware module needs to be implemented or it should be handled by the scheduling hardware and software. Moreover, if more than one write port tries to write same word in the same cycle, the results would be

unpredictable. A mechanism needs to be developed to make sure that no word is accessed by more than one write port.

- **Improvements in the layout**

As has already been noticed, the parasitics contribute significantly to the overall power. The values of parasitics were extracted from the layout of the memory core. The layouts of the memory cells can be made more compact. This directly translates to reduction in parasitics which in turn reduces the overall power dissipation.

- **Use of SRAM transistors**

SRAM transistors are specifically meant for memories. A full custom layout using these transistors could be made and simulated. Further discussion on this topic is done in Appendix C.

- **Custom layout of complete register file**

During the course of the entire project, only the layout of the memory core has been done. Even though the parasitics in the circuits outside the memory core have been accounted for in the SPICE transistor netlists, layout of the complete register file is needed to be done. Simulation of the extracted SPICE netlist from the layout of the complete register file will also verify the area and power values estimated from the SPICE transistor netlist.

References

[1] Leng, R. J., *Computer Memory Hierarchy* [Online]. Available: http://www.bit-tech.net/hardware/memory/2007/11/15/the_secrets_of_pc_memory_part_1/3 [Accessed: June 26, 2011]

[2] *Static Random Access Memory* [Online]. Available: http://en.wikipedia.org/wiki/Static_random-access_memory [Accessed: Jan 20, 2011]

[3] GojkoBabić, "Register File Design and Memory Design," Class Lecture [Online]. Available : http://www.cse.ohio-state.edu/~teodores/download/teaching/cse675.au08/Cse675.02.E.MemoryDesign_part1.pdf [Accessed : Jan 20, 2011]

[4] Nestoras Tzartzanis, "Static Memory Design," in *High-Performance Energy-Efficient Microprocessor Design*, Springer, 2006, pp. 89-119.

[5] T. Jau, W. Yang and C. Chang, "Analysis and Design of High Performance, Low Power Multiple Ports Register Files", in *IEEE Asia Pacific Conf. on Circuits and Systems*, Singapore, Dec. 2006, pp. 1453-1456.

[6] Shenglong Li, Zhaolin Li and Fang Wang, "Design of a High-Speed Low-Power Multiport Register File", in *Asia Pacific Conf. on Postgraduate Research in Microelectronics and Electronics*, Jan. 2009, pp. 408-411.

[7] A. Bellaouar, Mohamed I. Elmasry, "Low-Power CMOS Random Access Memory Circuits," in *Low-power Digital VLSI Design: Circuits and Systems*, Kluwer Academic Publishers, 1999, pp. 339.

[8] *HSPICE ® Simulation and Analysis User Guide,* Synopsys, March 2006.

[9] *Eldo User's Manual*, Mentor Graphics Corp., March 2005.

[10] *Virtuoso ® Layout Editor User Guide – Product Version 5.0,* Cadence Design Systems, Inc., January 2003.

Appendix A - Output circuit

Sense amplifiers detect small voltage variations on bit lines and convert them to proper logic levels. These circuits are often used with differential-ended bit lines. In this report, all designs implementations use single-ended read topology. In some of the designs, the voltage swing on the read bit lines does not translate to proper logic levels. In these designs, circuits similar to sense amplifiers are required to convert small voltage variations into the proper logic levels. These circuits are refered to as output stage throughout the report. Some of the possible implementations for output stage are mentioned in Section 4.3.4. This section specifies some more variants of output stage that were simulated.

The output circuits discussed below were tried to be implemented with the time multiplexed and partitioned design explained in Chapter 5. In this design implementation, the swing on the read bit line is between 1.2V and 0V. 1.2V provides a weak logic '1' on RBL. This has to be converted to a proper logic level '1' of 1.8V. Following subsections discusses few output circuits that were simulated, but not used in final designs due to various problems.

Output circuit 1

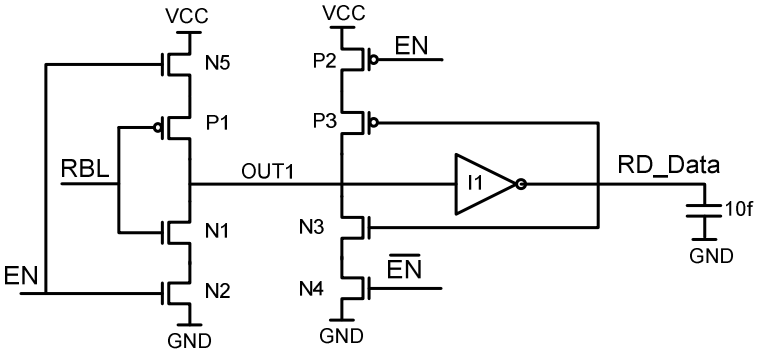


Figure A. 1 Output circuit 1

The output stage shown in Figure A.1 uses NMOS (N5) in series with the PMOS (P1). As discussed earlier, voltage swing on RBL is between 0V and 1.2V. Voltage swing on node 'OUT1' is between 0V and 1.4V. This circuit reduces the short circuit power in the tri-state inverter by shifting the input flipping voltage below 'VCC/2'. But this causes short circuit power in inverter I1 during the duration of the read cycle. The power dissipated by this circuit was less compared to output circuit shown in Figure 4.16.

The downside of this circuit is that, when supply voltage of 1.6V is used, it could not meet the timing requirements. At lower voltages the transistors are slow. The graphs shown in the Figure A.2 give an idea as to why this happens. The top most waveform, in blue, is the swing on the RBL. The next two waveforms are the control signals EN and its inverse as shown in the Figure A.1. The waveform in green, is the voltage at the node OUTI. Let us look at the time when there is a voltage spike impulse at this node. Voltage on the node

OUT1 is expected to rise up to 1.4V during the read cycle. But, instead it only rise up to 700 mV. This does not toggle the output of inverter I1 and the value at RD_Data in incorrect.

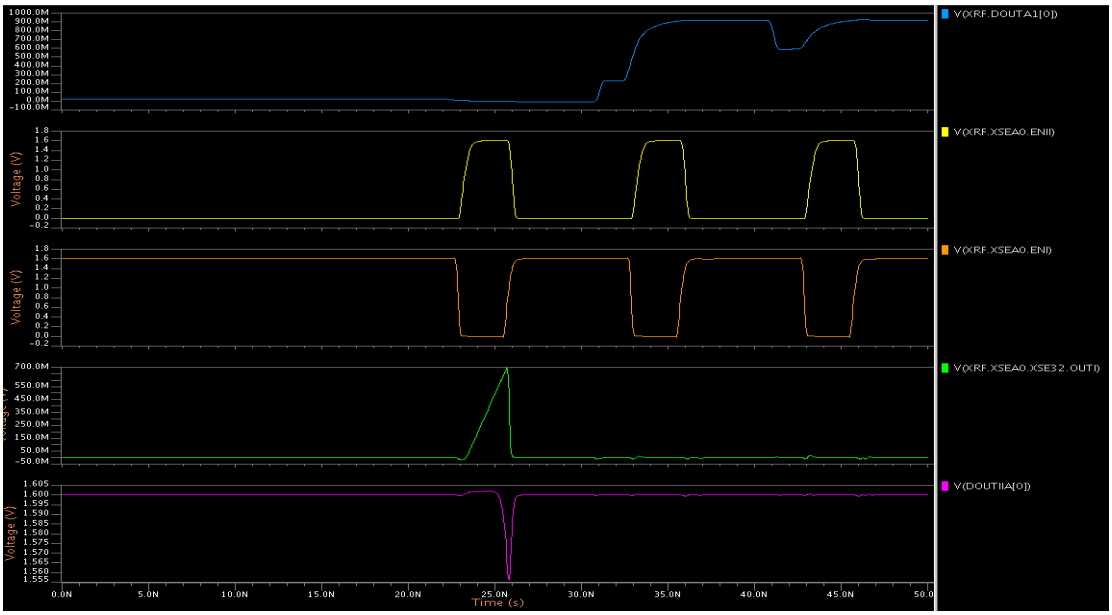


Figure A. 2 Waveforms for output circuit 1

Output circuit 2

Figure A.3 shows another configuration for an output circuit which is based on domino logic. This circuit eliminates the short circuit power that was present in the Output Circuit 1.

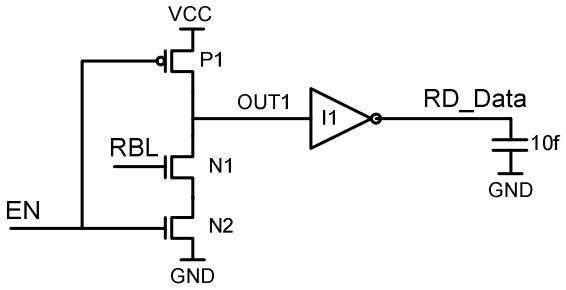


Figure A. 3 Output circuit 2

The required output stage has to perform function of latch for time multiplexed design implementations. To make a latch, a TG and a keeper circuit with enable is used as shown in Figure A.4. It is important to notice that all the control signals run through the entire width of the register file and a capacitance equal to RBL is added to all control signals. Power dissipation of this circuit is approximately 25% more that the circuit used in Section 4.3.4.3.

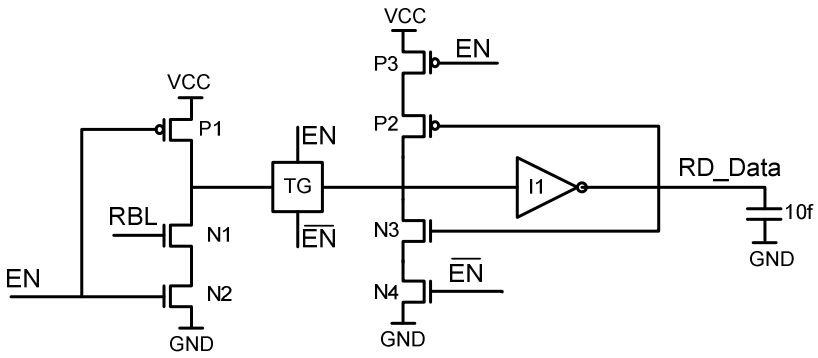


Figure A. 4 Output circuit 3

Appendix B – Clock correction circuit

Design implementation discussed in the report depend on the positive and negative edge of the clock to generate local control signals. In the simulations, 50% duty cycle clock is used to simulte the designs. But in reality the duty cycle of clock on the chip can vary between 30% and 70%.

This problem can be solved by using a duty cycle correction circuit as shown in Figure B.1. This circuit is basically a positive edge detection circuit. Irrespective of the duty cycle of the clock, the width of the 'CLK_OUT' pulse can be controlled by the delay circuit. Delay circuit uses special delay cells provided by the Atmel libraries. The behavior of delay cells varies across the PVT corners. The delay cells were adjuted to generate clock of 50% duty cycle in worst case timing corner. The duty cycle of the CLK_OUT pulse varies from 30% in the best case timing corner to 50% in the worst case timing corner.

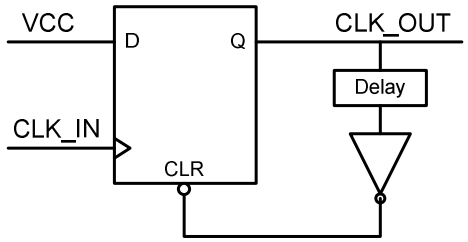


Figure B. 1 Clock duty cycle correction circuit

This circuit was tested with the time multiplexed design discussed in Section 5.3. It worked well across all the PVT corners. Power consumed by this circuit is approximately **80 µW**. This circuit is not used in the final versions of implementations, due to lack of time, but gives an idea about the power consumed by this module.

Appendix C – Register file with SRAM transistors

All the SPICE transistor netlist simulations discussed in the previous sections have been performed by using low leakage, high threshold-voltage transistors. These transistors have limitations on the minimum width. In the read/write time multiplexed and partitioned design, a majority of the transistors have a minimum width and the timing constraints were easily met.

To make the bit cell smaller, the width of transistors could be reduced even further than what the high voltage transistors allow. For this purpose, the RAM specific transistors could be used. These transistors can be used with slightly lesser widths than the minimum size allowed for the high voltage transistors.

The highly symmetrical memory module can be made very dense using the RAM transistors. As the size of the memory core decreases, the parasitics reduce as well and there is a reduction in power dissipation. Also, since these transistors are small, the write data driver and the word line drivers can be slightly smaller in these designs.

A memory core made with the RAM transistors was modeled in SPICE and tested for the read/write time multiplexed and partitioned design. Since a layout using these transistors wasn't made, the parasitics of the bit cell design implementation with high voltage transistors were only used. This design was tested at all the process corners. An overall power reduction of approximately **100 µW** was obtained.

The value of power dissipation obtained from these simulations was not very accurate. This is because the parasitics used weren't extracted from the bit cell layout made of RAM transistors. But, this simulation gives an idea as to how much power saving can be expected with RAM specific transistors.