# Asynchronous 1R-1W Dual-Port SRAM by using Single-Port SRAM

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY DELHI

K Bharath

VLSI and Embedded Systems

IIIT-Delhi

A thesis submitted for the degree of

*Master of Technology*

December 2016

1. Reviewer: Dr. Alexander Fell

2. Reviewer: Harsh Rawat

Day of the defense:

Signature from head of M.Tech committee:

# Abstract

With the advancement in technology nodes, the number of components operating in different clock domains on System on Chip (SoC) increases. To support the processing of data between these components, the demand of an asynchronous multi-port memory on SoC is rising. This paper introduces an asynchronous multi-port memory with dedicated write and read ports. The memory architecture is based on the Single-Port SRAM (SP-SRAM) that can be generated in larger capacities with good performance compared to the Dual-Port SRAM (DP-SRAM). The proposed design has been evaluated by comparing existing dual-port 1R-1W and 2RW designs in Ultra Thin Body and Box Fully Depleted Silicon on Insulator (UTBB-FDSOI) technology. A 2048 words of 64 bit memory shows 15%, 35%, 28% and 4.5% improvement in read power, write power, read-write power and performance respectively over conventional 1R-1W DP-SRAM with equal area. The same size memory with area optimization technique shows 50% area advantage over conventional 1R-1W DP-SRAM but with degradation in performance.

This thesis is dedicated to my teachers, who taught and enchoraged me throughout my life how to lead and learn from each and every aspect.

# Acknowledgements

Foremost, I would like to express my hearty thanks and indebtedness to my guide Dr. Alexander Fell for his encouragement, enthusiasm, insightful suggestions, comments and hard questions.

I would like to express my deepest gratitude to my mentor Mr. Harsh Rawat for the continuous support in the research, for his patience, motivation and immense knowledge.

My sincere thanks also goes to Mr. Anuj Grover for offering me this opportunity in the group leading me to this diverse and exciting project.

I thank my fellow batch mate in IIIT Delhi: Renduchinthala Anusha for the stimulating discussions and for the help to write this thesis document.

Last but not the least, my family and dear friends. None of this would have been possible without their love. And the one above all of us, the omnipresent God for giving me the strength to plod on, thank you so much everyone.

# Contents

# List of Figures

## LIST OF FIGURES

# List of Tables

## LIST OF TABLES

# 1

# Introduction

With the advancement in technology nodes, more functionalities are implemented on System on Chips (SoCs) by adding more components. To sustain the data processing among these components and to avoid high look-up latency due to off-chip memory with limited inputs and outputs (I/Os), large memories are integrated on the chip itself. According to the International Technology Roadmap for Semiconductors (ITRS), embedded memories occupy a large portion of SoC area (1). Therefore, to have the large capacity of memory within minimized area, the bit-cells of SRAM are scaled down, which makes the memory cells more prone to process variations. These variations have an impact on the performance of the bit-cell increasing the gap between the performance of the devices on SoC and the memory. Also, due to the tremendous technological upsurge, a boom in multimedia applications is witnessed. These multimedia applications (i.e. mobiles, setup box) demand parallel and back to back data processing for which large memories are to be shared at the centre of the chip(2). Further, these applications necessitate different clock domain data transfers and need low latency, high throughput, and high performance. SP-SRAM despite showing a good performance, does not satisfy the aforementioned requirements. Hence, asynchronous multi-port memories are required.

Unlike SP-SRAMs which can access only one memory location in a clock cycle resulting in sequential operations, multi-port memories have more than one port and can access multiple memory locations at a time with the ability to perform read and write operations simultaneously. Dual-Port SRAM (DP-SRAM) falls in this category, consisting of two ports for simultaneous operations. Each port of DP-SRAM consists of

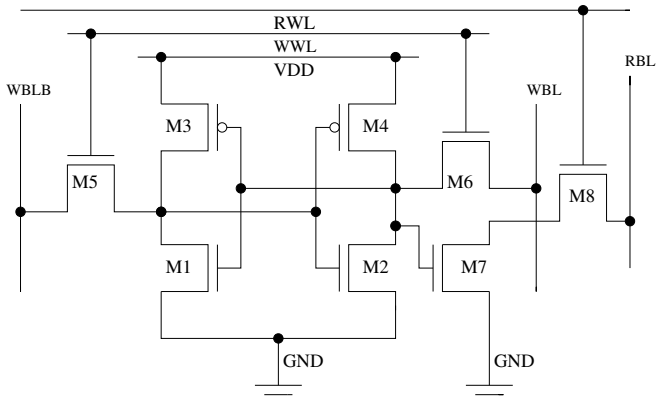


**Figure 1.1:** An 8T 1R-1W Dual-Port SRAM cell

dedicated bit and word lines leading to the addition of extra metal layer in silicon. To control these lines, in total 8 transistors (T) are required in DP-SRAM, thus increasing the chip area.

The existing DP-SRAM cell allows to do read and write operations on each port and hence is named as a 2RW (2 Read Write) cell. Drawbacks of this cell are instability and size. Figure 1.1 shows another 8T DP-SRAM cell with two ports, one is dedicated for read and another is dedicated for write operations. Therefore, this cell is called as 1R-1W cell. In a 1R-1W cell, M7-M8 and Read Bit-Line (RBL) are used for the read port while the other port acts as SP-SRAM port for a write only operation. Unlike 2RW DP-SRAM, 1R-1W memory can perform only 1R-1W in one clock cycle. But 1R-1W DP-SRAM has the advantage of higher read and write stability over a 2RW DP-SRAM cell (refer to appendix A). On the other hand a 1R-1W cell consumes more power than that of SP-SRAM because of 8T and single ended differential amplifier (3).

This paper proposes a dual clock 1R-1W DP-SRAM using SP-SRAM, which gives better performance, with a reduction in power and area, to replace existing conventional 1R-1W DP-SRAM. The paper is organised as follows: Section II presents related work addressing 1R-1W DP-SRAM problems and drawbacks of existing solutions. Section III describes the proposed design architecture designed using SP-SRAM and its implementation. Section IV shows the results in terms of performance, area and power and followed by the conclusion in section V.

# 2

# Related work

In existing conventional 1R-1W and 2RW DP-SRAMs, each port works on a different clock (4) as these ports are independent from each other. However, if both ports try to access the same memory location, a contention occurs and the data integrity is lost. For example, in 1R-1W memory, there is a possibility that both ports access same address location for reading and writing data. If the ports are operating on different clocks, the arrival of operation timings may be different. If a read is received before the write, it may read the old data. Similarly, if the read arrives after write, it may read the new data or a combination of both the old data and new data. This leads to a contention.

A solution to this problem is to allow one of the ports to proceed and to access the cell, while the other port is blocked and rescheduled. This will be indicated to the processor through a flag. In next clock cycle, the blocked operation will be executed. This results in a sequential execution of the two operations which reduces the throughput. Moreover, 1R-1W DP-SRAM operates at lower speeds because of its cell architecture. To overcome the speed disadvantage, banking architecture technique (5) can be used at the cost of an increase in area.

Several techniques were proposed in the literature to overcome disadvantages of power and area consumptions of conventional 1R-1W DP-SRAM. Time Division Multiplexing (TDM) (6)-(7) and Replica based designs are among those creating extra read and write ports. TDM based memory read and write operations are sequential with respect to the internal clock of the memory although it seems like a parallel execution with respect to the SoC clock, as the internal memory clock is twice the highest clock of the accessing device in the SoC. Hence it does not experience contention problems and

greatly reduces area. However, this memory is reduced to the domain of low frequency applications. Despite the fact that this memory has two ports, it has only one clock as input and hence is limited to synchronous data transfers as explained in appendix C.

In the Replica based technique(8), a DP-SRAM with a capacity of $W$ words is designed using capacity blocks of $W/2$ or $W/4$ words with an additional empty block of same size. The empty block is used when read and write operations access the same address location or the same block. This additional memory block enables 1R-1W operation within the same clock cycle. Hence the system throughput is unaffected. It gives better performance and lower power consumption because it is designed by using SP-SRAMs with half or one-fourth the size of the total DP-SRAM capacity. However this design cannot be instantiated, if read and write ports require two different clock frequencies, just like the TDM based memory design as explained in appendix D.

Both TDM and Replica based techniques can be used to design a 1R-1W Single Clock DP-SRAM (SC-DP-SRAM) but do not support read and write operations at different frequencies, which is a prerequisite for multi-clock domains often found in SoCs. Therefore they cannot replace 8T 1R-1W DP-SRAM shown in figure 1.1. The proposed design addresses this requisite and can be integrated in a multi-clock environment. It offers two input signals for clock domains, while at its core, an SC-SP-SRAM implemented.

# 3

# Proposed Design

In this section, an improved design for a 1R-1W SC-DP-SRAM is proposed to solve the drawbacks of existing designs which were discussed in the previous section. This work includes the modification of the SC-DP-SRAM to enable the multi-clock domain operation using a synchronization mechanism among the clock domains. The techniques that are available for this clock domain crossing, are two stage flip-flop synchronization, handshake protocol based synchronization and First In First Out (FIFO) based synchronization. The flip-flop synchronizers have the disadvantage of incoherency (9) and data loss while on the other hand the handshake protocol based synchronizers have the drawback of high latencies (10). In contrast, the FIFO based synchronizers used in the proposed design, have the ability to overcome all these issues but at the cost of area.

### 3.0.1 FIFO Synchronizer

FIFO buffers are used as a synchronizer between two clock domains. The depth of a FIFO depends on the read and write clock frequencies of the FIFO. If the write clock frequency of the FIFO ($f_{wclkf}$) is higher than the read clock frequency of the FIFO ($f_{rclkf}$), data overflows can occur which lead to loss of data. In addition, if data is written burst by burst with a particular amount of delay, based on the delay, depth widely varies with every frequencies change of the FIFO buffer (11). But in case where $f_{wclkf} < f_{rclkf}$, the maximum required depth of the FIFO buffer is only four as per the equation 3.1 which is derived by the explanation of the FIFO operation below.

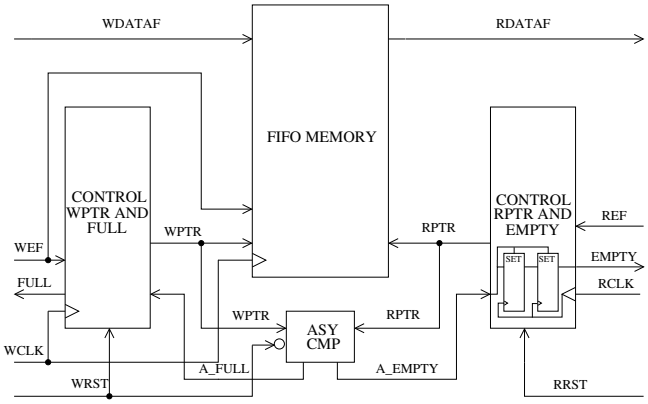


**Figure 3.1:** The basic block diagram of an asynchronous FIFO

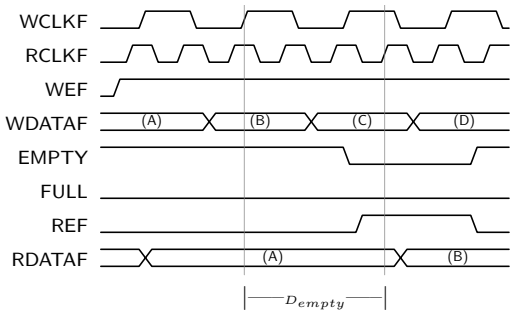


**Figure 3.2:** FIFO timing diagram

A generic FIFO structure (12) is shown in the figure 3.1. The data signals of the FIFO buffer are RDATAF (read data), WDATAF (write data). The control signals are WEF (write enable of FIFO buffer), REF (read enable of FIFO buffer), RRST (read clock reset), WRST (write clock reset). The clock signals are RCLK (read clock) and WCLK (write clock). The EMPTY and FULL signals show whether the FIFO buffer is empty or full respectively. The timing diagram of basic write and read operations considering a read clock faster than the write clock is shown in figure 3.2. For a write operation into the FIFO buffer, it takes one write clock cycle of FIFO ($T_{wclkf}$). It is because when there are parallel read and write operations going on and read frequency is much higher than write frequency and FIFO is undergoing read operations, there is a possibility that the data being written into the FIFO during write operation has to be given to a module reading from FIFO and this leads to the invalid data transmission.

After this write operation to show the data availability, EMPTY signal takes two to

three read clock cycles ($T_{rclkf}$) to change its status which is shown in the figure 3.2 as $D_{empty}$. This is because of two flip-flop synchronizer used in the FIFO buffer as shown in the figure 3.1. The depth of the FIFO buffer for $f_{wclkf} < f_{rclkf}$ is to be large enough to avoid overwriting of old data before it is delivered to the read port. Hence the depth depends on the number of write operations between the time taken to write data into the FIFO buffer and the EMPTY signal to trigger REF. The formula for depth $d$ of the FIFO is shown in the equation 3.1.

$$d = 1 + 3\left(\frac{f_{wclkf}}{f_{rclkf}}\right) \tag{3.1}$$

where

- a depth of at least 1 is required to write data into the FIFO buffer.
- plus additional 3 storage slots to observe status change in the EMPTY signal after write operation.

The maximum depth ($d_{max}$) of the FIFO buffer for $f_{wclkf} < f_{rclkf}$ is obtained, when both $f_{rclkf}$ and $f_{wclkf}$ are equal. From equation 3.1,

$$d_{max} = 1 + 3 * 1 = 4. \tag{3.2}$$

Due to the upper bound of the depth $d_{max}$ for $f_{wcklf} < f_{rclkf}$, which is not applicable for $f_{wclkf} > f_{rclkf}$ as discussed earlier, a data transfer occurs always from the slower write to a faster read clock in the proposed design. Hence the data receiver (SC-DP-SRAM) always operates on the faster ($f_{rclkf}$) clock. According to the clock domain condition, two designs are needed: The first design is for read clock frequency higher than the write clock frequency of the memory ($f_{wclkm} < f_{rclkm}$) while the second design is for write clock frequency higher than the read clock frequency of the memory ($f_{wclkm} > f_{rclkm}$). Both the designs work when both the clocks are equal including a potential phase shift between them.

### 3.0.2 Proposed Designs

In this subsection the proposed designs for $f_{wclkm} < f_{rclkm}$ and $f_{wclkm} > f_{rclkm}$ are explained.

#### 3.0.2.1 Proposed Design for $f_{wclkm} < f_{rclkm}$

The block diagram of the design for $f_{wclkm} < f_{rclkm}$ is shown in figure 3.3. Control signals of this design are WEASY (asynchronous write enable) and REASY (asynchronous read enable). The data signals are WDATA (write data), RDATA (read data), RADD (read address) and WADD (write address). It has two blocks, consisting of the FIFO buffer and SC-DP-SRAM. As discussed above, the FIFO buffer acts as a synchronizer from the slower ($f_{wclkm}$) to the faster ($f_{rclkm}$) clock domain, while the SC-DP-SRAM operates at the frequency $f_{rclkm}$. After the transfer of data through the FIFO buffer, the data is stored in the SC-DP-SRAM. The timing diagram of the read and write operations of the proposed design with $f_{wclkm} < f_{rclkm}$ is shown in figure 3.4.

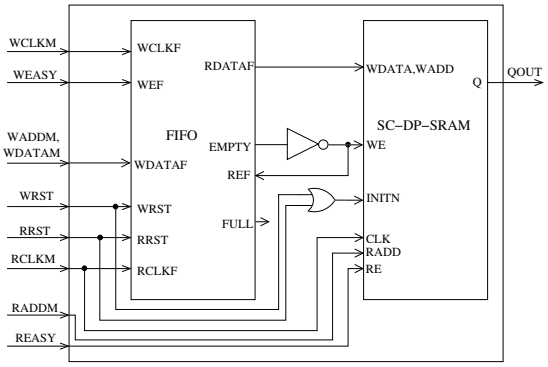


**Figure 3.3:** Proposed design for $f_{wclkm} < f_{rclkm}$

The sequence of write and read operations in this design:

- Address and data are written into the FIFO buffer in 1 $T_{wclkm}$.
- Based on the data availability, the EMPTY signal status changes. It takes 2 to 3 read clock cycles ($T_{rclkm}$) for the change, for any frequency combination of read and write clocks of proposed memory because of FIFO buffer architecture as explained in section 3.0.1 (12).
- The EMPTY signal triggers the REF (Read Enable of FIFO buffer) and RE (Read Enable of SC-DP-SRAM). WDATA is written into the SC-DP-SRAM at WADD. This takes 1 $T_{rclkm}$.
- This entire operation takes maximum of 1 $T_{wclkm}$ plus upto 4 $T_{rclkm}$ for any read and write frequency combination of the memory.

- For the read operation, it takes only one $T_{rclkm}$ like the normal read in DP-SRAM since the read clock is directly connected.

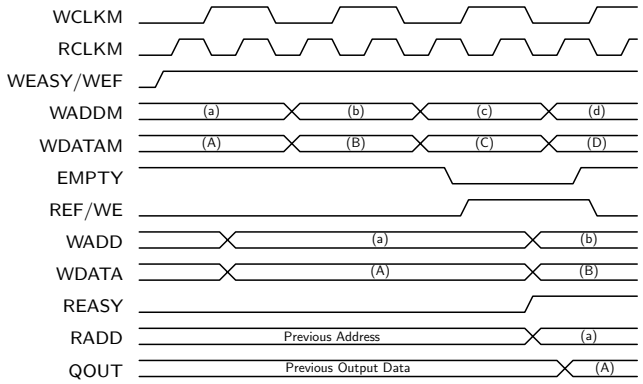


**Figure 3.4:** Timing diagram of proposed design for $f_{wclkm} < f_{rclkm}$

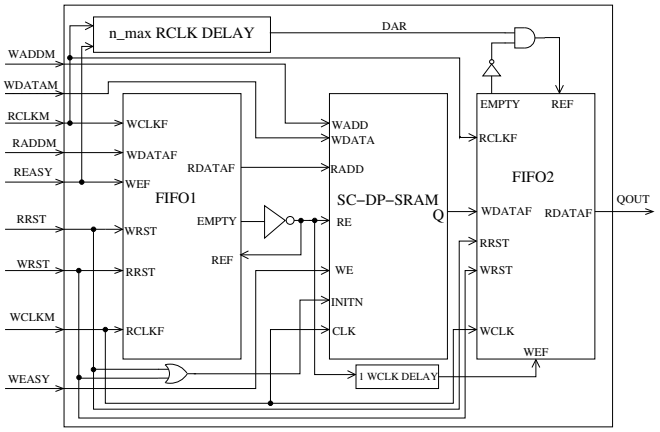


**Figure 3.5:** Proposed design for $f_{wclkm} > f_{rclkm}$

### 3.0.2.2 Proposed Design for $f_{wclkm} > f_{rclkm}$

The block diagram of the proposed design for $f_{wclkm} > f_{rclkm}$ is shown in the figure 3.5. Its control and data signals are the same as in the previous design. It consists of two FIFO buffers and one SC-DP-SRAM. In this design, as SC-DP-SRAM works on the faster write clock, the read address needs to be transferred from the read clock domain to write clock domain. After reading from SC-DP-SRAM, the data will be in the write clock domain, but it has to be in read clock domain. To transfer it to

read clock domain, another FIFO buffer (FIFO2) is used. But FIFO2 transfers data from faster write clock ($f_{wclkm}$) to slower read clock domain ($f_{rclkm}$) which conflicts the statements regarding the upper bound of $d_{max}$ in section 3.0.1. But those constraints do not have any effect on the data, because the $f_{rclkm}$ is slower than the clock the memory operates ($f_{wclkm}$), and hence the addresses where to read from, cannot be fed into the memory system fast enough to fill FIFO2 buffer. Therefore, no overflow of data occurs.

Figure 3.5 shows that the delayed REASY input signal is connected to FIFO2 to trigger the read operation of FIFO2 along with EMPTY signal of FIFO2. This connectivity is used to maintain constant delay from change in input (RADD_M) to change in output (QOUT) of the memory for getting output at known time because the delivery time of data to read port of FIFO2 may vary based on the time taken by EMPTY signals of FIFO1 and FIFO2 which leads to difficulty in data transmission for particular cycle.

The timing diagram of a single write followed by a read operation is shown in figure 3.6.

The sequence of write and read operations in this design:

- Similar to read operation in the proposed design for $f_{wclkm} < f_{rclkm}$ in section 3.0.2.2, the write operation in this design takes $T_{wclkm}$ since SC-DP-SRAM directly runs on the write clock.
- For the read operation, RADD_M should be written to the FIFO1, which takes 1 $T_{rclkm}$.
- Based on data availability, the EMPTY status changes. It takes 2 to 3 $T_{wclkm}$ ($= T_{rclkf}$) for any frequency combination of the read and write clock because of FIFO buffer architecture as explained in section 3.0.1 (12). It is followed by a read from the SC-DP-SRAM, requiring 1 $T_{wclkm}$.
- After the read operation in the SC-DP-SRAM, the data needs to be transferred from the write back to the read clock domain. To enable this, the data is written into FIFO2 and WEF of FIFO2 is triggered by one $T_{wclkm}$ delayed RE of SC-DP-SRAM. To write the data into FIFO2, it takes 1 $T_{wclkm}$.
- Update of the EMPTY signal status takes again 2 to 3 $T_{wclkm}$.

- REF of the FIFO2 is triggered when the EMPTY signal of FIFO2 is inactive and REASY signal delayed by $n \times T_{rclkm}$ (delayed asynchronous read enable (DAR) signal) is active.
- The first read operation takes $n \times T_{rclkm}$ and subsequent read operations take only 1 $T_{rclkm}$.

The delay of $n \times T_{rclkm}$ considered for REASY signal is the maximum data path latency from the input REASY of the proposed design to the time taken for EMPTY signal of the FIFO2 to change its status. The latency ($n \times T_{rclkm}$) calculation from the input RADD_M of the proposed design to the EMPTY signal of FIFO2 is shown in equation 3.3.

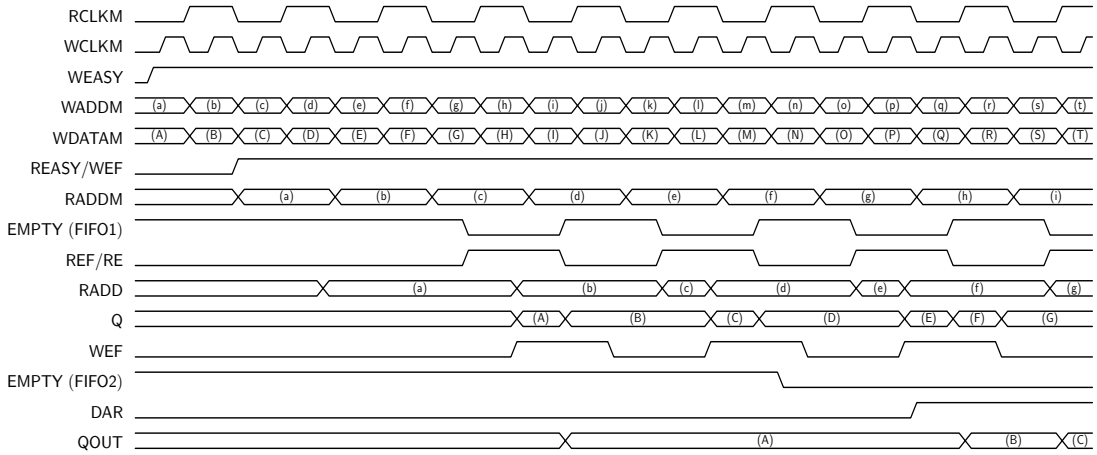


**Figure 3.6:** Timing diagram of proposed design for $f_{wclkm} > f_{rclkm}$

$$\begin{aligned} n = 1\ T_{rclkm} + p\ T_{wclkm} + 1\ T_{wclkm} + 1\ T_{wclkm} \\ + q\ T_{rclkm}\ \text{ with } p, q \in \{2, 3\} \end{aligned} \tag{3.3}$$

Reasons for each value of latecy:

- 1st summand: For writing into the FIFO1.
- 2nd summand: The delay pertained to the status change of EMPTY signal of FIFO1 (12).
- 3rd summand: To read from the SC-DP-SRAM.

- $4^{\text{th}}$ summand: For writing into the FIFO2.
- $5^{\text{th}}$ summand: The delay pertained to status change of EMPTY signal of FIFO1 (12).

**Example 1.** If $f_{wclkm} = 2 \times f_{rclkm}$: Minimum latency ($n_{min}$) from equation 3.3 is

$$\begin{aligned} n_{min} &= 3\ T_{rclkm} + 4\ T_{wclkm} \\ &= 5\ T_{rclkm} \end{aligned} \tag{3.4}$$

Maximum latency from equation 3.3 is

$$\begin{aligned} n_{max} &= 4\ T_{rclkm} + 5\ T_{wclkm} \\ &= 6.5\ T_{rclkm} \end{aligned} \tag{3.5}$$

As shown in figure 3.5, REASY is delayed by maximum time $n_{max}$. It is due to if the read operation takes a delay of $n_{max}$ and REASY is delayed only by $n_min$, DAR will arrive at the input of the AND gate before the EMPTY signal becomes active. This misses to trigger of required read operation in FIFO2.

The depth of FIFO depends on the time taken to store the data into FIFOs, time taken for EMPTY signal status change and time for the arrival of feedback signals. Depth of FIFO2 ($d_{FIFO2}$) is shown in equation 3.6.

$$d_{FIFO2} = 1\ + 3\ + (n_{max} - n_{min}) \tag{3.6}$$

Reasons for each value of the depth in equation 3.6:

- $1^{\text{st}}$ summand: Write operation into FIFO2.
- $2^{\text{nd}}$ summand: Writing data during change in EMPTY signal status.
- $3^{\text{rd}}$ summand: To prevent data loss, during the time gap between enabling of the EMPTY signal (when it follows minimum data path latency $n_{min}$) and arrival of DAR ($n_{max}$) at the AND gate. This is the difference between maximum and minimum latencies calculated in equations 3.4 and 3.5.

The depth of the FIFO2 buffer is highest when $f_{rclkm} = f_{wclkm}$, as the value of $n_{max} - n_{min}$ is maximum.

$$\begin{aligned} max(d_{FIFO2}) &= 1 + 3 + 2 \\ &= 6 \end{aligned} \tag{3.7}$$

Applying the FIFO synchronizers extends the SC-DP-SRAM to an asynchronous dual clock-dual port SRAM (DC-DP-SRAM) utilizing only SP-SRAM with extra logic.

# 4

# Results

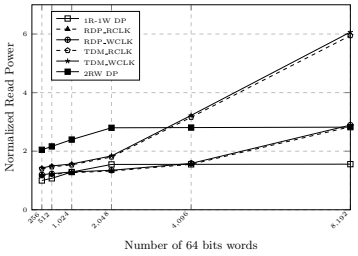


**Figure 4.1:** Memory Read Power

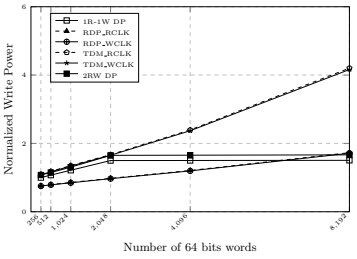


**Figure 4.2:** Memory Write Power

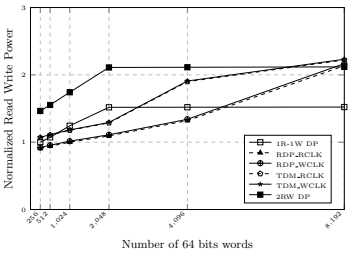


**Figure 4.3:** Memory Read-Write Power

In this section, the proposed asynchronous DP-SRAM designed by using SP-SRAM, is compared with existing conventional 1R-1W DP-SRAM and 2RW DP-SRAM designs in terms of power, performance and area using Ultra-Thin Body and Box Fully Depleted Silicon on Insulator (UTBB-FDSOI) technology for both $f_{wclkm} < f_{rclkm}$ and $f_{wclkm} > f_{rclkm}$. In literature the proposed DP-SRAMs are designed using SP-SRAMs with Time Division Multiplexing (TDM) (6) and Replica based (8) techniques. The designs are implemented in Verilog HDL, synthesized in Synopsys Design compiler (DC) for area and clock frequency calculation and power is calculated by using Synopsys Primetime. The capacities of memory designs considered are 256, 512, 1024, 2048, 4096 and 8192 words of 64 bit width each. The size of the conventional 1R-1W DP-SRAM and 2RW DP-SRAM is limited to 2048 words of 64 bits. To generate 4096 and 8192 sizes of conventional 1R-1W DP-SRAM, the 2048 word sized memory is duplicated multiple times and organized such that it can be addressed as a single memory bank and results are calculated. All values shown in the graphs are normalized to the

readings of conventional 1R-1W DP-SRAM with 256 words of 64 bits capacity. For the two proposed designs for $f_{wclkm} < f_{rclkm}$ and $f_{wclkm} > f_{rclkm}$, the slower clock is set to half the frequency of faster clock.

Figure 4.4 shows the cycle time comparison among various designs. It is observed that for RDP_RCLK (Replica based Dual port memory with the read clock faster than the write clock) and RDP_WCLK (Replica based Dual port memory with the write clock faster than the read clock) cycle times are reduced by 16.5% and 4.5% for 1024 and 2048 words capacity compared to conventional 1R-1W DP-SRAM of same sizes. With increase in memory size, gain in cycle time increases for RDP_RCLK and RDP_WCLK memories till a capacity of 1024 words because they are designed by using three SP-SRAMs, each with a capacity of half the size of the memory to be built. For the sizes larger than 1024 words, the gain decreases because of the change in 1R-1W DP-SRAM architecture as 2048 word memory includes Bank-4 architecture (5) and the same is replicated for designing 4096 and 8192 memories whereas till 1024, 1R-1W DP-SRAM is designed using Bank-2 architecture. The maximum operating frequencies of both TDM_RCLK and TDM_WCLK memories are approximately half of that of an SP-SRAM showing 131% higher cycle time compared to 1R-1W DP-SRAM for the capacity of 256 words of 64 bits.

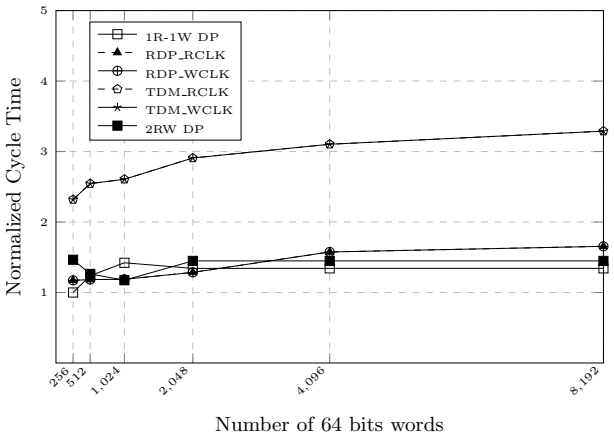


**Figure 4.4:** Memory Cycle Time

Figure 4.5 shows the area comparison of various designs. The maximum area gains of 50% and 47% are observed for TDM_RCLK and TDM_WCLK memories respectively over the 1R-1W DP-SRAM for the capacity of 2048 words and the minimum area gain

is observed at 256 word capacity for both the designs. The area gain rises with increase in capacity because of the usage of SP-SRAMs. RDP_RCLK has almost equal area as that of the 1R-1W DP-SRAM whereas RDP_WCLK occupies 3% more area compared to 1R-1W DP-SRAM for the capacity of 2048 words. The area gain is noticed for the sizes larger than 2048 words for both the designs. For the capacity of 8192 words, both RDP_RCLK and RDP_WCLK show area gains of 10.5% and 9.7% compared to that of 1R-1W DP-SRAM of same capacity. The area decreases with the rise in the memory capacity because of the increase in advantage of using half sized SP-SRAM for both RDP_RCLK and RDP_WCLK designs.

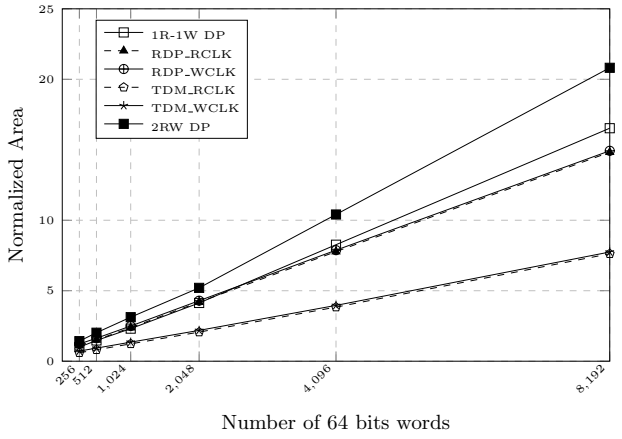


**Figure 4.5:** Total Memory Area

Read, write and read-write power are defined as the power consumed by the memory for one read, write and read-write operations respectively. Figures 4.1, 4.2 and 4.3 show the comparison of read, write and read-write powers respectively for various designs considered. RDP_RCLK and RDP_WCLK memories consume 15%, 35% and 28% lesser read, write and read-write powers respectively than 1R-1W DP-SRAM for 2048 words capacity. For both RDP_RCLK and RDP_WCLK designs, the read, write and read-write power gains increase with the rise in the memory capacity till 2048 words as shown in the figures because of the usage of half of the memory capacity sized SP-SRAMs. For the designs with the capacity from 2048 to 8192 words, the power gains decrease as 1R-1W DP-SRAM with the capacity of 4096 and 8192 words are designed with replicas of 2048 memories. Whereas for TDP_RCLK and TDP_WCLK designs, read and write power consumptions are higher than that of the 1R-1W DP-SRAM and with

the increase in size, the power consumption rises because of its TDM architecture as explained in appendix C (7). For TDM_RCLK and TDM_WCLK, an improvement of 15% is observed in read-write power over 1R-1W DP-SRAM for the 2048 word capacity because of the usage of SP-SRAM. For the designs with the capacity from 2048 to 8192 words, the power gains decrease as 1R-1W DP-SRAM for 4096 and 8192 words are designed with replicas of 2048 memories.

Conventional 2RW DP-SRAM is also compared with 1R-1W DP-SRAM and it is clear from the results that the proposed RDP designs are better in read, write and read-write power by 52%, 40%, 28% respectively and by 20% and 19% in performance and area respectively, whereas TDM designs are better in area, read power and read-write power by 60%, 35% and 39% compared to 2RW DP-SRAMs.

The drawbacks of the proposed designs are, $f_{wclkm} < f_{rclkm}$ design write operation takes more than one write clock cycle. $f_{wclkm} > f_{rclkm}$ takes more than one read clock cycle for first read operation.

# 5

# Conclusion

This paper proposes a novel DP memory with asynchronous read and write clocks using a single port memory to provide the data storage functionality. The results show that compared to existing dual port memories which suffer from read instability, limited bank sizes and data contention, the replica based proposed design, for both scenarios, shows a higher performance along with decreased power consumption over conventional 1R-1W DP-SRAM with the capacity of 2048 words of 64 bits. The TDM based proposed design is area efficient compared to conventional 64 bit DP-SRAM with 2048 word capacity for both the designs. Additionally, the proposed design is scalable to large memory capacities which is not possible with the available dual-port memories.

# References


[1] Y. Zorian. **Embedded memory test and repair: infrastructure IP for SoC yield**. *Proceedings International Test Conference*, pages 340–349, 2002. 1

[2] D Schwaderer and P Martin. **Solving SoC shared memory resource challenges**. *Sonics Inc., Jun*, 2003. 1

[3] Yibin Ye, Muhammad Khellah, and Dinesh Somasekhar. **Evaluation of Differential vs. Single-Ended Sensing and Asymmetric Cells in 90nm Logic Technology for On-Chip Caches**. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 963–966, 2006. 2

[4] CYPRESS. **Understanding Asynchrnous Dual-Port RAMs**. URL: http://www.eet-china.com/ARTICLES/2001MAR/PDF/2001MAR29_MEM_AN1092.PDF?SOURCES=DOWNLOAD. 3

[5] Tom Granberg. *Handbook of Digital Techniques for High-Speed Design*. Pearson Education, 1 edition, 2007. 3, 16

[6] C. E. LaForest and J. G. Steffan. **Efficient multi-ported memories for FPGAs**. *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, pages 41–50, 2010. ACM. 3, 15

[7] Jonathan Dama and Andrew Lines. **Pseudo dual-port SRAM and a shared memory switch using multiple memory banks and a sideband memory**, February 2003. US Patent 8370557 B2. 3, 18

[8] Sundar Iyer and Shang Tse Chuang. **System and method for storing data in a virtualized high speed memory system**, April 2013. US Patent 8,433,880 B2. 4, 15

[9] Tejas Dave and Amit Jain and Divyanshu Jain. **Synchronizer techniques for multi-clock domain SoCs & FPGAs**. URL: http://www.edn.com/electronics-blogs/day-in-the-life-of-a-chip-designer/4435339/Synchronizer-techniques-for-multi-clock-domain-SoCs. 5

[10] Mohit Arora. *The Art of Hardware Architecture*. Springer, 1 edition, 2011. 5

[11] Putta Satish. **CALCULATION OF FIFO DEPTH**. URL: http://pgschool.gtu.ac.in/moodle/pluginfile.php/1850/mod_forum/post/391/FIFODEPTHCALCULATIONMADEEASY2.pdf. 5

[12] Clifford E. Cummings and Peter Alfke. **Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons**. URL: http://www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO2.pdf. 6, 8, 10, 11, 12

[13] N. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley, 4 edition, 2010. 23

[14] Y. Ishii, H. Fujiwara, K. Nii, H. Chigasaki, O. Kuromiya, T. Saiki, A. Miyanishi, and Kihara. **A 28-nm dual-port SRAM macro with active bitline equalizing circuitry against write disturb issue**. *VLSI circuits, IEEE Sympoism on*, pages 99–100, June 2010. 25

[15] Ramandeep Kaur. *XMAT: A 6T XOR-MAT based 2R-1W SRAM for High Bandwidth Network Applications*. Master's thesis, IIIT-D, New Delhi, 2015. 27, 29

# Appendix A

# 6T SRAM cell

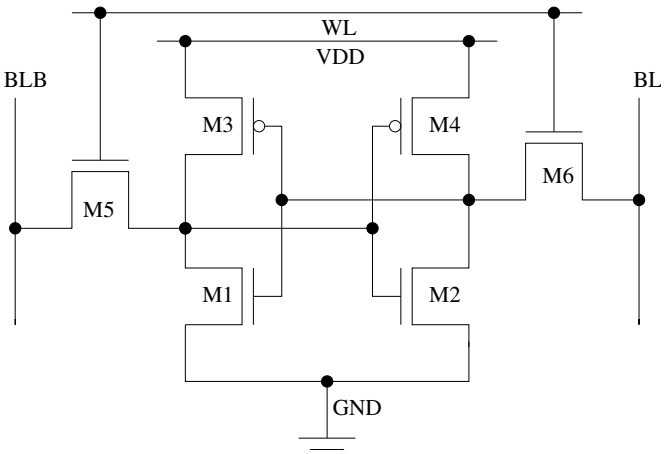


**Figure A.1:** A 6T SRAM cell

The basic 6T SP-SRAM cell is shown in figure A.1. A 6T SRAM cell acts as a latch with back to back connected inverters (13). It can be accessed through two M5-M6 access transistors. M1-M2 are pull-down transistors, which are used in the read operation. M3-M4 are pull-up transistors, these are used in a write operation of a bit-cell. For stability of the SP-SRAM, the width of M1-M2 and M3-M4 transistors needs to be larger than M3-M4 and M5-M6 transistors respectively. As 6T SP-SRAM has only one port, it allows access to a single memory location.

# Appendix B

# 8T 2RW DP-SRAM cell

A DP-SRAM cell is shown in figure B.1. It is an extension of 6T SP-SPAM cell and consists of two additional transistors as compared to 6T SP-SRAM. They are used to achieve the dual port functionality. In one memory clock cycle, DP-SRAM supports two operations through two access ports independently (14). For stability, the size of M1-M2 transistors in DP-SRAM are larger than that of the M1-M2 transistors found in SP-SRAM because of the two read operations in DP-SRAM. They create read bump at the storage node, which is twice compared to the read bump of a 6T SP-SRAM. The read bump might lead to a bit flip in the cell and makes the cell less stable.

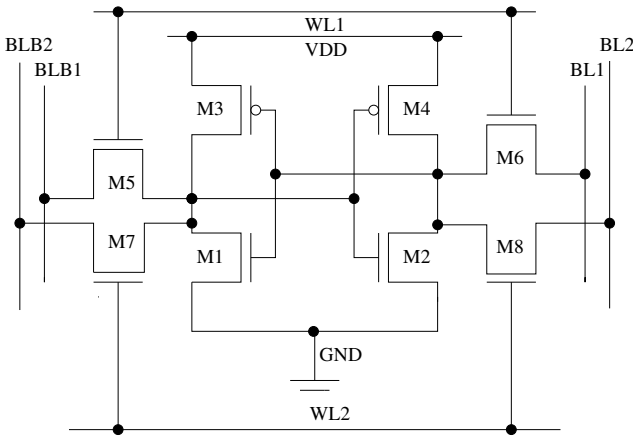


**Figure B.1:** An 8T 2RW Dual-Port SRAM cell

# Appendix C

# Time Division Multiplexing based DP-SRAM

In this section, the design of a synchronous 1R-1W DP-SRAM is discussed using the Time Division Multiplexing (TDM) technique. An extra access port is created with the concept of *accessing ports in different time slots*. Therefore the memory operates on internal clock frequency at multiples of external clock frequency. TDM technique can be applied to any memory design, to multiply its read and write ports.

The block diagram of TDM technique reproduced from (15) and shown in figure C.1, utilizes SP-SRAM to store data. It has two ports, one is dedicated to the read and the other one to the write operation, with two sets of address and control lines. As shown in figure C.2, the external clock frequency is half of the internal clock frequency of the memory. Therefore, two concurrent operations are serialized and executed sequentially from the SoC. While serializing the operations, the sequence of operations is fixed, the memory always reads in the first half of external clock cycle followed by the write operation. In the first half the data for the write operation is always stored in the registers temporarily (called as dummy write operation which is the reason for a higher power consumption).

If the number of ports increases, the external clock frequency decreases, which leads to performance degradation. Hence only for low frequency applications this type of memory can be used. In addition it has a disadvantage of single clock input and thus it is limited to synchronous data transfer.

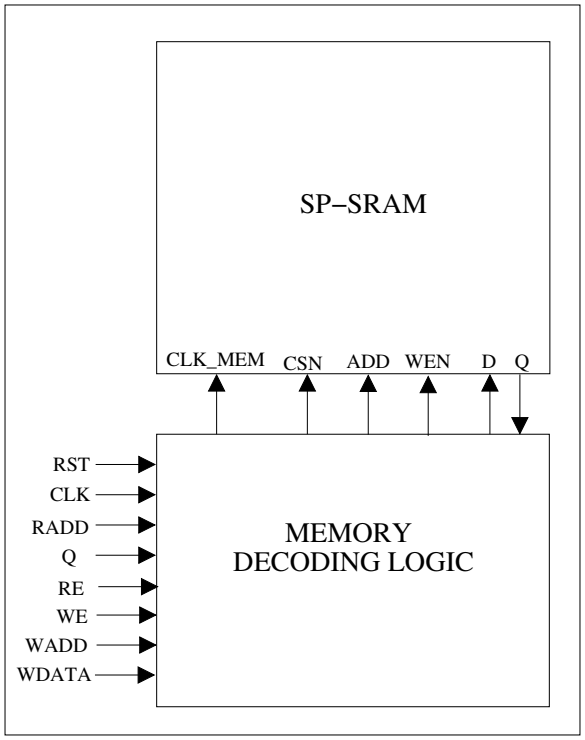


**Figure C.1:** Block diagram of TDM technique

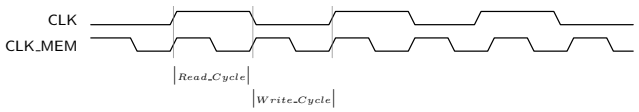


**Figure C.2:** Timing diagram of TDM Technique

# Appendix D

# Replica based DP-SRAM

In this section, the design of a synchronous 1R-1W DP-SRAM using the Replica based technique, is discussed. In the Replica based technique, DP-SRAM with a word capacity

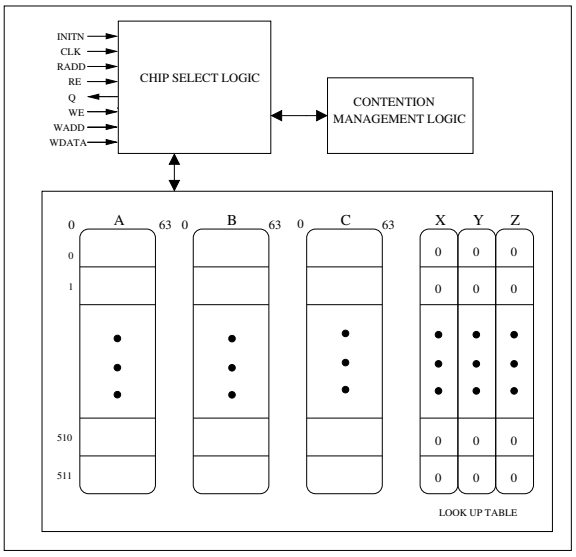


**Figure D.1:** Block diagram of Replica based t echnique

of $W$ is designed using $W/2$ or $W/4$ capacity memory blocks with an additional empty memory block of $W/2$ or $W/4$ size, and a Look Up Table (LUT). The LUT is used to map a virtual memory address to physical memory address. For $W/2$ blocks, the LUT needs 3 D-flip-flops for each of the $W/2$ rows of the SP-SRAMs. The block diagram of the memory design using the Replica based technique reproduced from (15), is shown in figure D.1. It has a capacity of 1024 words of 64 bits designed with 512 words of 64

bits sized blocks. Block $A$ and $B$ store the data for addresses $[0, W/2)$ and $[W/2, W)$ respectively.

In addition to these 3 SP-SRAMs and the LUT, it has two blocks: chip select logic and contention management logic. The chip select unit decides the bank in which read-write operations have to be performed. The contention management logic detects whether the 1R-1W operation points to same or different memory bank. After initialization of the memory, LUT contains 0-0-0 as shown in the block diagram (figure D.1).

**Table D.1:** LUT data indication table

| LUT data | | | Block holding address range | |
|---|---|---|---|---|
| **X** | **Y** | **Z** | $[0, W/2)$ | $[W/2, W)$ |
| 0 | 0 | 0 | A | B |
| 0 | 0 | 1 | NA | |
| 0 | 1 | 0 | NA | |
| 0 | 1 | 1 | A | C |
| 1 | 0 | 0 | C | B |
| 1 | 0 | 1 | NA | |
| 1 | 1 | 0 | C | A |
| 1 | 1 | 1 | B | C |

As shown in table D.1, if the value of $X$ holds 0, the data of the lower address range is available in block $A$ and if it is 1, it may or may not hold the data of the higher address range depending on the flat in $Y$ and $Z$. Similarly, block $Y$ indicates the availability of higher address range data in block $B$. If $Z$ holds 0, it indicates that block $C$ holds lower address range data, and if it is 1, block $C$ holds higher address range data. But if both $X$ and $Z$ hold 0, the higher priority is given to $X$ and lower address range data is considered to be in block $A$ and not in block $C$. Similarly, if $Y$ and $Z$ hold 0 and 1 respectively, the higher address range data is present in block $B$ not in block $C$.

For example, let us consider that both read and write operation need to be executed at address 0 and the LUT has an initial data set of 0-0-0 for this address. The write operation occurs at the address 0 of block $C$ as 0-0-0 values in LUT indicate that the

data of addresses 0 and 512 are available at $A$ and $B$ blocks respectively and $C$ block holds invalid data. Therefore the data is read from block $A$, while the new data is written into block $C$ which now holds valid data, whereas the data in block $A$ becomes invalid. This change of validity is marked by changing the LUT values to 1-0-0 for address 0.