XMAT: A 6T XOR-MAT based 2R-1W SRAM for High Bandwidth Network Applications

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Ramandeep Kaur
VLSI and Embedded Systems
IIIT Delhi

A thesis submitted for the degree of
Master of Technology

2015 June

1. Reviewer: Dr. Alexander Fell

2. Reviewer: Harsh Rawat

Day of the defense:

Signature from head of M.Tech committee:

Abstract

System on Chips (SoC) targeted for high performance network applications such as Internet routers, require low latency memories combined with large storage capacities to maintain high throughputs and fast packet forwarding capabilities. To meet these demands, Dual Port SRAM (DP-SRAM) consisting of 8 transistors (T), are integrated into the SoC. However in contrast to 6T Single Port SRAMs (SP-SRAM), DP-SRAMs suffer from a limited performance, large area consumption, read-write instabilities and constraints regarding the memory capacity. In this paper an SP-SRAM based memory architecture is proposed which is able to execute two reads, one write or alternatively one read and one write within a clock cycle by combining a dedicated memory bank for XOR calculations with a Memory Association Table (MAT). In comparison to DP-SRAM the new design shows an improvement of 21%, 11% and 5% in access time, cycle time and power reduction for a 20% chance of contention respectively for a memory capable of storing 1024 words of 64 bit depth each.

This thesis is dedicated to my grandmother, who taught me that even the largest task can be accomplished if it is done one step at a time.

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 mates in IIIT Delhi: Rahul Malhotra and Abhishek Jain for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we had in the last two years.

Last but not the least, I would like to thank my family for supporting me spiritually throughout my life.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

Core routers are the key element of the Internet forwarding packets to their destinations. Due to the exponentially increasing volume of traffic especially at the dawn of Internet-of-Things (IoT), these routers are equipped with specialized high performance networking System-on-Chip (SoC) to maintain quality of service(1). Apart from a fast decision logic and Lookup Table (LuT) determining which route a packet should be forwarded to, packets need to be stored temporarily in case a particular route is congested. To cater to the demand of fast transmission speeds and low latency, the SoCs are therefore equipped with specifically designed memory components in the form of SRAM in order to optimize system performance(2).

Within the SoC the memory needs to fulfill very different tasks. For instance, the packet buffer is implemented as a First-in First-out (FIFO) and characterized by a storage capacity of several megabytes(3). Since the buffer is in the path of traversing packets, a high throughput needs to be achieved. The packet buffer consists of multiple FIFO such as Virtual Output Queues (VOQ) where a packet buffer holds one FIFO for each output queue in a shared memory router(4). It is possible that successive arriving packets belong to different queues and depart in a different order than their arrival. Thus, the packet buffer processing memory should essentially be a 2R, 1R-1W type of DP-SRAM to facilitate FIFO handling with improved efficiency and low latency read. From figure 1.1, packet P0 arrives which needs to be routed to port 0 and is written into the FIFO. In the next cycle, P0 has to be read from port 0 however, it is blocked due to insufficient output port resources i.e. the port is already experiencing packet

congestion. P0 cannot be routed at the moment and is still in the packet buffer. In the same cycle P1 arrives and needs to be routed to Port 1 and is written into FIFO joining P0. The router keeps on checking for port 0 to become available. Port 0 and 1 become available in the next cycle and P0 and P1 can be forwarded at the same time to different ports because of the 2R facility in the buffer design. Next, P2 arrives and is forwarded to Port 2 in the following cycle.

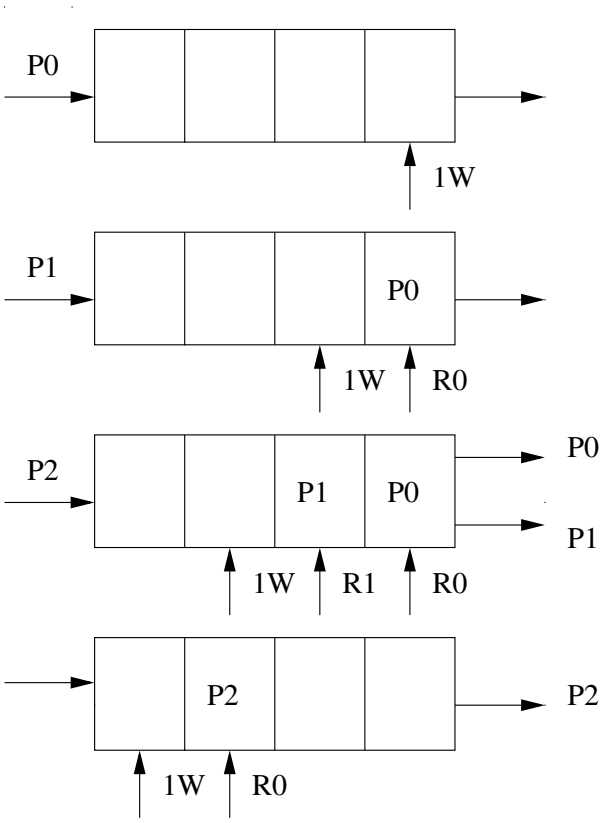


Figure 1.1: Routing of packets using a 2R, 1R-1W packet buffer

These requirements are in contrast to the LuT, which checks the destination address of each packet against a physical output port of the router. This access is read dominated, since these LuTs are rarely updated(5). However while the destination is looked up, the packet is stalled which increases its latency. To minimize this bottleneck the SRAM for the LuT is optimized for low-latency random access and short access times.

Multi-port memory, such as Dual-Port (DP) memory, unlike its Single-Port (SP) memory counterpart, is generally characterised by its ability to access the memory on one port while concurrently reading or writing data to the memory on another port. Hence, the DP memory is provided with two independent sets of address, data, and control lines. DP-SRAM allows a parallel access to the memory in contrast to Single-port Static Random Access Memory (SP-SRAM) where the read-write operations are forced to be sequential. Classically, DP-SRAM designs have been implemented with a dedicated wordline and bitline for each port. The area of the memory is dominated by the number of metal lines (wordlines and bitlines) running through it, thus, duplicating them results in significant increase in the silicon area in comparison to a SP-SRAM. However, higher throughput is achieved with increasing number of access ports of the SRAM(6). Although memories with more number of read and write ports are essential for high throughput applications requiring parallel architectures, their implementation area and cost are high and thus, are used sparingly in designs.

Figure 1.2 shows a schematic of a 6 Transistor (6T) SP-SRAM cell. The transistors M5 and M6 are pass transistors which aid in accessing the bit stored in the back-to-back latches formed by the transistors M1-M3 and M2-M4(7). For higher cell stability of the SP cell, the width of the pull-down transistors M1, M2 should be the largest and the

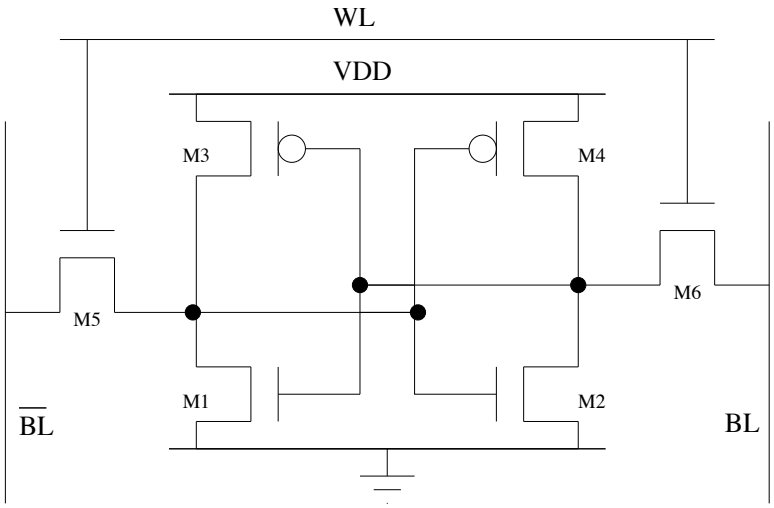


Figure 1.2: A conventional single-port SRAM cell

pull-up transistors M3, M4 should be the weakest. The 8T 2RW DP cell is an extension to the 6T cell with two additional pass transistors and a dedicated wordline and bitline for each port as shown in figure 1.3. This cell can process the access requests from the two ports in a single memory clock cycle independent of each other(8). The 2RW DP cell has greater flexibility than 1 Read - 1 Write (1R-1W) cell (shown in figure 2.1) because it is capable of performing 2 reads, 2 writes or 1 read-write simultaneously in a single cycle. An alternative 1R-1W 8T DP-SRAM cell, illustrated in figure 2.1 is a slight variation of the 2RW 8T cell and has a separated write and read only port. This cell has the advantage of a lower width of the pull-down transistors M1, M2 which helps to get a considerable improvement in read and write margin. Read and write margins can be defined as the lowest wordline voltage at which read or write operations can take place. However, this cell can only perform a concurrent read-write operation in a single memory clock cycle.

The additional access port in DP-SRAM comes at the cost of reduced density, read-write stability issues and increased read-write power. Because of the simultaneous access of the two ports, the DP-SRAM spends much more power due to charging and discharging of two pairs of bitlines during each cycle, when the two ports simultaneously access the memory at the same frequency. As a result of device scaling, increased threshold voltage (V_t) variation of transistors due to random dopant fluctuations degrades the operating margin of embedded SRAM with a subsequent performance reduction at minimum supply voltage (VDD_{min}). In addition, the VDD_{min} of DP-SRAM is more vulnerable to such variations than that of 6T SP-SRAM, due to its dual-port access behaviour(9). Practically, the VDD_{min} of DP-SRAM is higher, when both the ports concurrently access cells in the same row. This is called the “read-stability issue”. Also, if both the wordlines (WL1, WL2) in figure 1.3 are triggered simultaneously, the storage node of one latch cannot be immediately discharged to ground level because of the disturbance from the other bitline BL2(10). In this case the storage node retains the previous data even after the wordlines are deactivated, resulting in a write failure. This is called the “write-stability issue”.

DP-SRAM with concurrent 1R-1W or alternatively dual-read (2R) capabilities in the same clock cycle can meet these challenges. However, a conventional 8T DP-SRAM

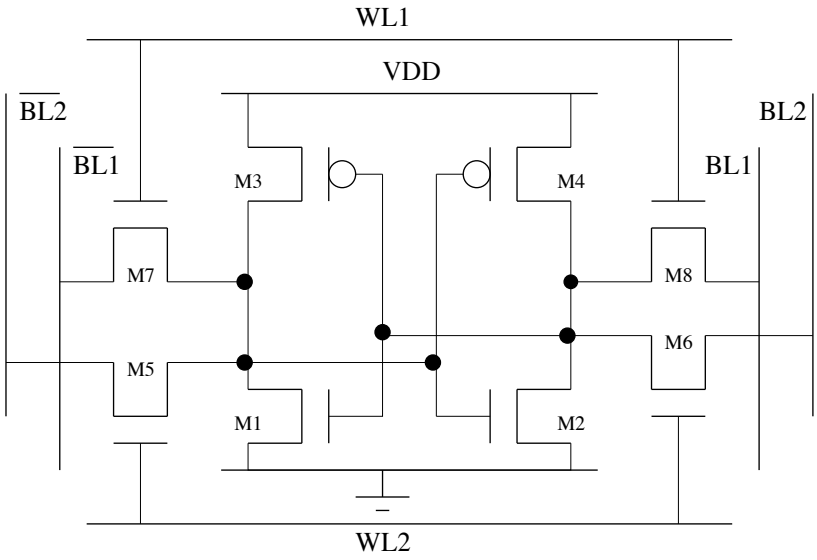


Figure 1.3: An 8T 2RW dual-port SRAM cell

cell supporting 2 concurrent read and write accesses (2RW), suffers from various drawbacks such as limited performance, large area consumption, read-write stability issues when a read/write operation is executed in the same row, and constraints in available memory capacity(11). In this work, an SRAM architecture is introduced using multiple SP-SRAM in its core allowing 1R-1W or 2R accesses within the same clock cycle satisfying the requirements of packet buffers and LuT. In addition using SP-SRAM to mimic the behavior of DP-SRAM avoids the read-write stability issues and results in larger memory capacities available with higher operating frequencies(12). The architecture is described in detail in section 3 after introducing related work in section 2. In section 4 the performance, power and area results are compared between DP-SRAM, selected proposals found in literature and the proposed architecture, while section 5 concludes the paper.

2

Related Work

Several techniques have been proposed in literature targeting the broad range requirements for memories in networking applications. (1) introduces a dual-port 1-read, 1-write memory, which significantly reduces the probability of congestion at the bank and architecture levels. In this work a two-port bank design in addition to the two-port cell design supports the large bandwidth and bank parallelism critical for high-performance networking applications. The performance of this memory is benchmarked using queuing theory and statistical information obtained by processing network packet traces using Packet-Bench(13).

An alternate approach is to modify the structure of the 8T cell itself and add write-assist or read-assist schemes to overcome the stability issues as proposed in (9, 14). Read and write assist schemes such as Bit Line Equalizing (BLE) and Shifted bit line access are implemented into the 8T DP-SRAM cell to minimize the timing skew between the activation of both word lines (WL) and support simultaneous port access by including BLE switches(8). The activation of the write-port WL before the read-port WL causes a positive skew, leading to a write disturbance. Adding a write-assist 8T cell (WA8T) at the top and bottom edges of a DP-SRAM cell array can eliminate the write-disturbance especially for high-speed (short WL pulse width) applications(9). However, this further increases the cell and design complexity resulting in higher area and power requirements of the overall memory system.

In (15) the traditional 8T DP-SRAM cell is altered to address the aforementioned instabilities. The 2RW functionality is reduced to support a 1R-1W operation only resulting in a reduced width of the pull-down transistors M1 and M2. Therefore a considerable improvement in read and write margins has been observed(16).

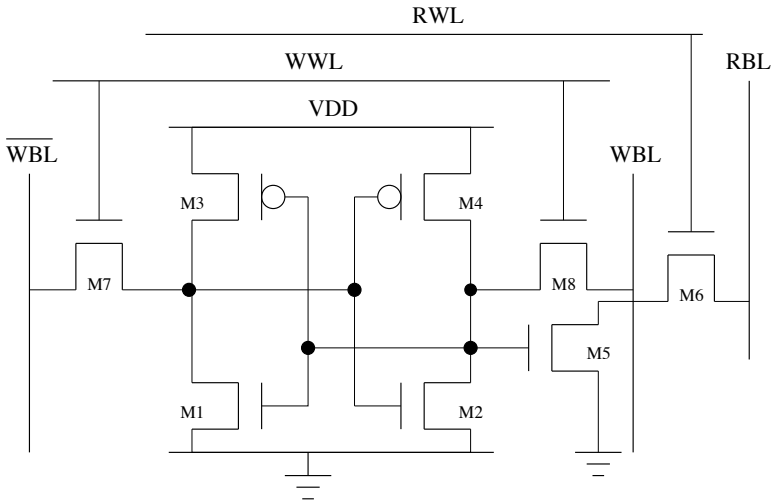


Figure 2.1: An 8T 1R-1W DP-SRAM cell

Instead of manipulating SRAM cells, a pseudo DP-SRAM can be implemented on an architectural level by operating a 6T SP-RAM at double the frequency of the external clock(17). While this pseudo DP-SRAM is externally clocked only once, a concurrent read-write operation in which the read is followed by the write, is executed internally on the SP-SRAM. This technique greatly reduces the area required for the memory compared to an actual DP-SRAM of same capacity. However it also reduces the maximum achievable external operating frequency. For further information refer to Appendix A.

In (18) a replication based technique is proposed in which the SP-SRAM banks are instantiated several times. Two concurrent operations are then executed on different memory banks. An additional memory is updated to indicate which bank contains the recently written data. While this approach does not assume an external clock at half the possible memory clock, the replication increases the area requirements. In addition only half of the total memory capacity is used at any given point in time. For further

information refer to Appendix B.

(19) introduces an inverted exclusive OR (XOR) based approach for a multi-read memory design utilizing a 4-port XOR bank, which stores the encoded version of all data that has been written in corresponding rows. In addition, it contains a small XOR contents table to keep track of which memory banks have valid data. The memory implemented is a 'read once' system wherein data is destroyed upon read. This is because it needs to be guaranteed that an empty memory bank is always available for each address in case data needs to be written. Otherwise, in addition to updating the XOR bank with the new content of the respective row, the old data needs to be removed from the corresponding entry in the XOR bank, a step which can be skipped. However this multi-port XOR bank increases the memory area further, adds complexity to the memory design in addition to read-write instabilities.

In (16) an XOR based pipelined 2R, 1W memory architecture is proposed using only 6T SRAM cells. To form a dual-port memory with a capacity of W_{max} words, three single-port memory banks of the size of $W_{max}/2$ are used. One of these banks stores the row-wise exclusive OR of the other two physical memory banks to retrieve the data in case of contention. Although the area requirements are less than 8T DP-SRAM, a concurrent read-write operation cannot be accomplished which excludes this approach from networking applications and FIFO packet buffers as mentioned in section 3. Moreover, the output of this XOR approach is pipelined increasing the overall packet latency. For further information refer to Appendix C.

While the DP-SRAM cell imposes constraints on memory capacity and clock frequencies, the approaches using SP-SRAM in their memory subsystem do not support all required operations in network applications of 2R and 1R-1W. Therefore a design is introduced, combining several approaches to satisfy the requirements.

3

Proposed Design

In this paper an SP-SRAM based memory architecture is proposed which is able to execute two reads (2R) or alternatively one read and one write 1R-1W within a clock cycle by combining a dedicated memory bank for XOR calculations with a MAT. To facilitate the 2R operation an XOR based approach is adopted as described in (16). To perform a 1R-1W operation concurrently in a single cycle a variety of techniques can be used such as a pseudo DP SRAM or a replication based technique as introduced in section 2. Alternatively a MAT based technique can be used similar to a virtual memory mapping table technique as described in (19). The next sections discuss the techniques for the 2R and 1R-1W operation separately which are finally combined to form the proposed design.

3.1 XOR supported Memory

The XOR based technique is introduced in (18, 20) which is similar to the approach described in (16). To obtain a memory capacity with W_{max} words, the memory is equally divided into two memory banks (bank A and B) with a capacity of $W_{max}/2$ words each as shown in figure 3.1. Memory bank A stores data for the addresses from 0 to $W_{max}/2 - 1$ and B for the addresses $W_{max}/2$ to $W_{max} - 1$.

A third memory bank XOR is designed to store the XOR values of A and B such

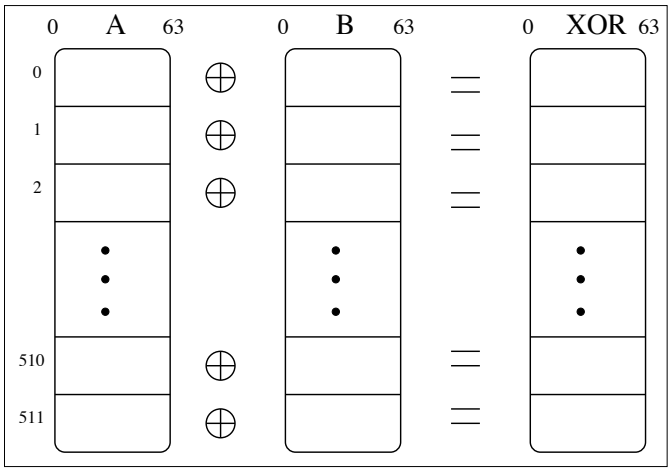


Figure 3.1: Physical memory banks in XOR technique

that each row

$$row_{XOR,a} = row_{A,a} \oplus row_{B,a} \tag{3.1}$$

where a represents the address with $0 \leq a < W_{max}/2$.

Due to the bitwise exclusive OR operation in equation 3.1, any unknown third value of a row can be retrieved by performing an XOR over the remaining two known operands. The algorithm described in listing 1, covers both contention and contention free memory access cases. The most significant bits (MSB) of addresses a_1 and a_2 are extracted and compared. If they differ, both read accesses point to different memory banks and therefore both reads can be executed on the respective bank in a clock cycle using the remaining least significant bits of each address after MSB extraction. However if the MSB are equal, the first read of address a_1 is allowed to proceed, while the data stored at a_2 needs to be reconstructed using data from the memory banks B and XOR. The waveform of XOR based design is shown in C.2 where read and write both operations are implemented in two cycles.

Algorithm 1 2R operation using XOR based technique

Precondition: 2R operation is executed

Precondition: Each row of bank A and bank B are $\oplus$ in XOR bank

```
1: function DUAL-READ(a_1, a_2)
2:     if msb(a_1) ≠ msb(a_2) then
3:         q_1 ← read(A, a_1)
4:         q_2 ← read(B, a_2)
5:     else
6:         q_1 ← read(A, a_1)
7:         q_2 ← read(B, a_2) ⊕ read(XOR, a_2)
8:     end if
9:     return q_1, q_2
10: end function
```

3.2 Memory Association Table

On an architectural level, a 1R-1W memory is created by implementing a memory-mapping technique(19, 21) called Memory Association Table (MAT) which maps a virtual memory address to an address of the physical memory bank. For a memory capacity of W_{max} words, three SP-SRAM of a size of $W_{max}/2$ words are required. In addition the MAT consists of 6 D-flip-flops for each of the $W_{max}/2$ rows of the SP-SRAMs. These flip-flops store identifiers unique to each bank, into three fields (LOW, HIGH and EMPTY) as shown in figure 3.2. LOW stores the identity of the SP-SRAM which contains the data from 0 to $W_{max}/2-1$, while the field HIGH holds the identifier of the memory bank which contains data for the addresses from $W_{max}/2$ to $W_{max}-1$. Finally, the field EMPTY represents the memory bank which contains invalid data currently and can therefore be used for a write operation, if a contention occurs. In addition to these 3 physical SP-SRAM and MAT, a controller in form of a chip select unit chooses the memory bank where the read-write operation is to be performed. A contention management unit detects, if the 1R and 1W operations point both to the same or different memory banks.

For instance, if a $1\text{k} \times 64$ 1R-1W memory is instantiated, three 512×64 SP-SRAM

and a MAT with 512 rows is required. Initially, all rows of the association table are initialized to 00-01-10 indicating that the first $W_{max}/2$ addresses are stored in memory bank A, while the other $W_{max}/2$ addresses are served by memory bank B. Memory bank C is either empty or contains invalid data. This memory bank is selected only in case a contention is encountered.

If a single read operation is executed, the corresponding field of the MAT depending on the given address is simply read. The chip select unit chooses that particular memory bank and the operation is performed. A single write operation is executed in a similar manner. However in case of a contention, the field EMPTY of the MAT is read and subsequently the memory bank which currently contains invalid data, is identified for the write operation. The fields are updated in the next clock cycle to reflect the activation of the alternative memory bank. The algorithm for concurrent 1R-1W is summarized in listing 2.

For instance, a 1R-1W operation arrives on addresses 2 and 3 respectively after a reset. Row 2 and 3 of the MAT are read and both LOW fields contain the identifier 00 referring to memory bank A. Thus, a contention is detected. While the read operation is executed in memory bank A, the field EMPTY of the third row is read and the write operation is shifted to memory bank C. After the completion of the operation, row 3 of the MAT is updated to 10-01-00 indicating that address 3 is to be read from memory bank C from now onwards.

3.3 2R, 1R-1W XMAT Memory

The proposed design combines the two techniques to form a dual-port memory which can perform both 1R-1W or 2R in a single clock cycle, a necessity for packet switching routers. The block diagram of the proposed design is shown in figure 3.2. It consists of three SP-SRAM of $W_{max}/2$ word capacity, a 1R-1W XOR bank as described in section 3.1, and a MAT of $W_{max}/2$ rows made from D-flip-flops introduced in section 3.2. Additionally a contention management and a chip select unit detect and resolve memory

Algorithm 2 1R-1W operation using MAT

Precondition: Execute a concurrent 1R-1W

```
function READ-WRITE(RADDR, RWADDR, D)
    field_read ← (msb(RADDR) = 1) ? HIGH : LOW
    field_write ← (msb(RWADDR) = 1) ? HIGH : LOW
    bank_read ← MAT(RADDR, field_read)
    if field_read = field_write then
        bank_write ← MAT(RWADDR, EMPTY)
    else
        bank_write ← MAT(RWADDR, field_write)
    end if
    Q1 ← read(bank_read, RADDR)
    write(bank_write, RWADDR, D)
    return Q1
end function
```

bank conflicts.

While a memory together with a MAT can support the dual port operation 1R-1W, the 2R functionality is added by including an XOR memory bank into the design. If a 1R-1W operation needs to be executed, the corresponding identifier is looked up in the MAT and different memory banks are chosen, in case a conflict is detected. In this design after each write, the appropriate entry in the XOR memory needs to be updated additionally.

If the conflict occurs during a 2R operation, one read is served by reading the corresponding memory bank directly. However the data of the second read needs to be reconstructed by determining which memory bank holds the active data. This is achieved by reading the appropriate non-EMPTY field in the MAT, followed by an exclusive OR operation of the read data with the value stored in the XOR memory bank.

The waveform in figure C.2 illustrates the sequence of operations in the proposed

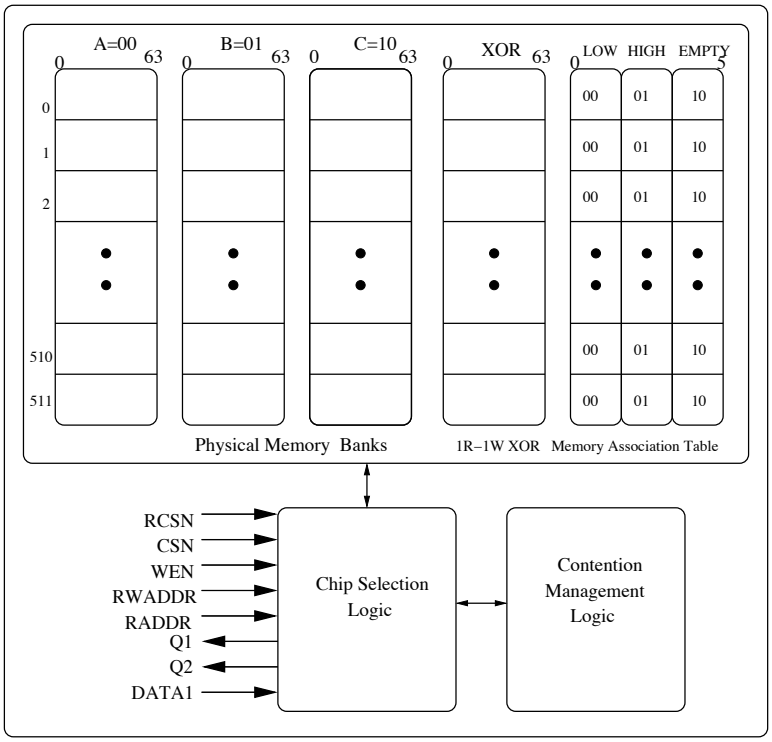
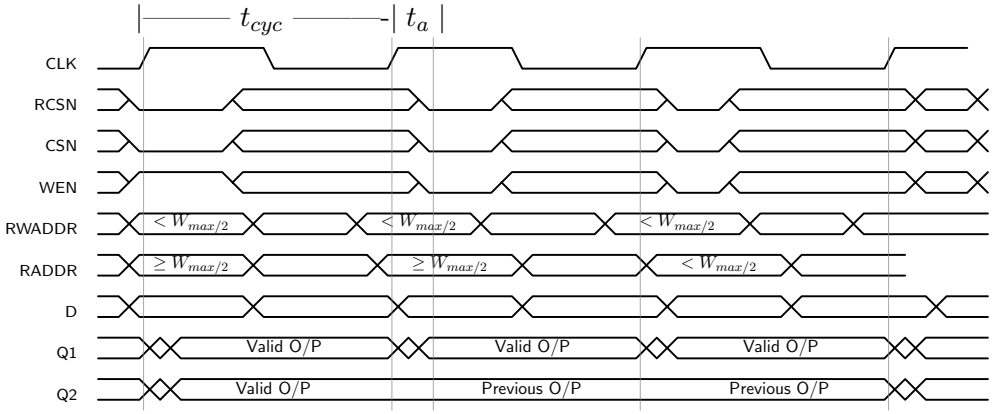


Figure 3.2: Proposed Design Diagram for a 1024×64 capacity



Operation	2R with contention	1R-1W w/o contention	1R-1W with contention
Execution Order	read_A, read_B, read_{XOR}	read_A, read_B, write_C, write_{XOR}	read_A, read_B, read_{XOR}, write_C, write_{XOR}

Figure 3.3: Waveform of proposed 2R-1W memory showing 2R and 1R-1W operation sequences with and without contention

design in case of a 2R or 1R-1W operation with contention and in contention-free cases and the SP memories that will be utilized for the same. For instance, to execute a 1R-1W operation, the signals of RCSN, CSN and WEN are set to 0 (active low), and the read and write addresses are passed to RADDR and RWADDR respectively, while D carrying the data to be stored, is setup. After the memory access time t_a the output of the read is latched at Q1. This 1R-1W operation with contention activates all the memories present in figure 3.2.

4

Results and Benchmarking

In this section, the proposed Exclusive OR - Memory Association Table (XMAT) implementation is compared to the conventional 8T 2RW and 1R-1W DP-SRAM, 6T SP-SRAM, simple XOR based 2R and MAT based 1R-1W implementations considering access time (t_a), cycle time (t_{cyc}), total area occupied and 2R/1R-1W power consumption using 28nm Ultra Thin Body and Box (UTBB)-Fully Depleted Silicon on Insulator (FDSOI) technology. The design has been implemented in Verilog Hardware Description Language (HDL) and synthesised using Synopsys Design Compiler. Power and performance characteristics have been obtained using Synopsys Primetime. The conventional 2RW and 1R-1W DP-SRAM is generated directly using dual-port compilers to a maximum capacity of 2048 words, 64 bits each. Traditionally, if both SP-SRAM and DP-SRAM are used together in a design, the performance is limited by the DP-SRAM because of its reduced operation speed. However, the comparison with the 6T SP-SRAM of similar capacity shows that the proposed design outperforms the SP-SRAM in some cases and the performance is no longer limited by the DP-SRAM. The conventional DP-SRAM can be directly generated using the dual-port compiler to a maximum capacity of 2048 words, 64 bits each. However, SP-SRAM can be generated up to a capacity of 8192 words of 64 bits. The comparison for capacities greater than the above sizes has been accomplished by replicating and combining two small memory banks to form one large DP or SP memory. Memory access time is the time taken by the output (Q) of the read instruction to reach a valid stable state (shown in figure C.2). Cycle time (t_{cycle}) includes the clk-to-memory activation time plus the clk-to-Q time. Memory activation time is the amount of time consumed in address decoding

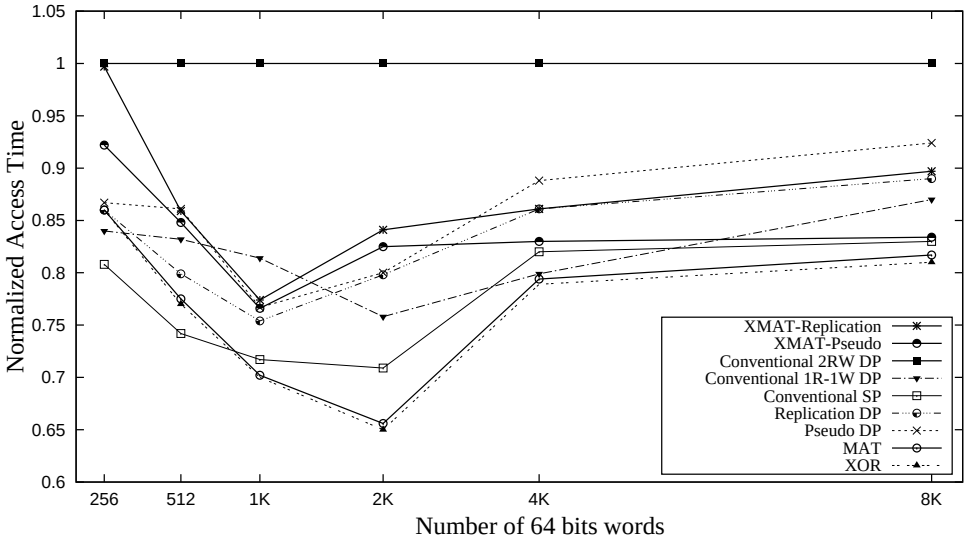
and the additional chip select logic time in our design.

The proposed design requires a 1R-1W type of XOR memory as shown in figure 3.2 which can be implemented in two ways: high density for area optimization and high performance. The area-optimized 1R-1W SRAM (XMAT-Pseudo in figure 4.1) is implemented using the pseudo DP-SRAM as mentioned in the section 2 which results in an area comparable to SP-SRAM and thus gives an area-compact result at the cost of a reduced performance. The high performance 1R-1W memory (XMAT-Replication) is a replication based design which occupies a larger area at a higher clock frequencies. Therefore the designer has the option to choose a memory architecture depending on the particular application and constraints.

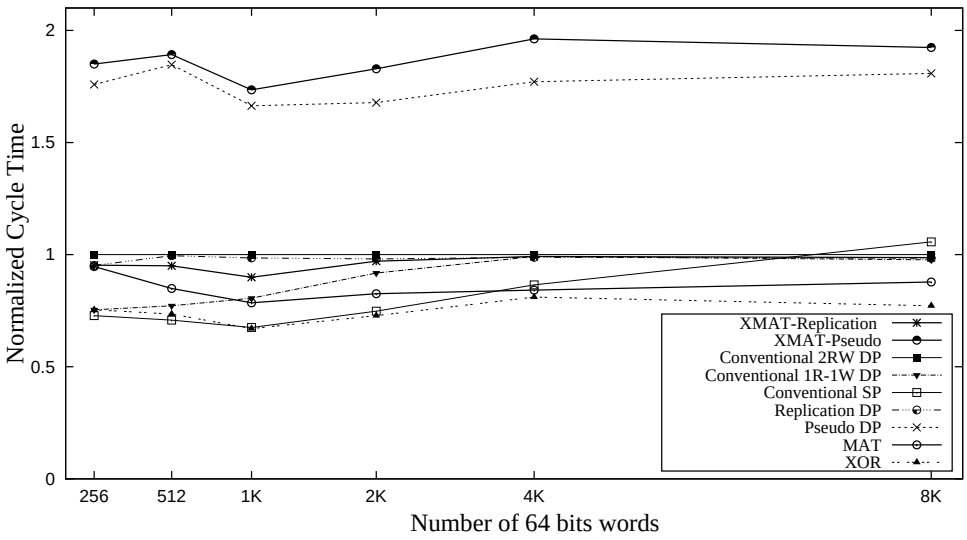
Figure 4.1a and 4.1b show an improvement of 21% in t_a and 11% in t_{cyc} for an XMAT-Replication in comparison to 8T DP-SRAM with a $1k \times 64$ capacity. The pure XOR and MAT based memories have similar t_a as both utilize SP-SRAM memories with a comparable combinational delay. For larger capacities ($W_{max} >$ 4k words) these designs are better in performance than even SP-SRAM of similar size as internally they use similar SP-SRAM of half the capacity. The cycle time of XMAT-Pseudo is 62% greater that of the 8T DP-SRAM due to the higher internal frequency which limits the maximum achievable clock rate externally as mentioned in section 2.

In figure 4.1c the area consumption is plotted. As it can be observed for both XMAT-Pseudo and XMAT-Replication the required area is 42% and 59% larger respectively when compared with the traditional 8T DP-SRAM. This is due to the MAT based 1R-1W which alone occupies 8% more area than 8T DP-SRAM for a $1k \times 64$ capacity SRAM. XOR based 2R design occupies 19% lesser area on an average for a similar capacity. Pseudo and conventional 6T SRAM occupy the least area of all the implemented designs.

Figure 4.1d and 4.1 show the power consumption for a 2R and 1R-1W operation in each clock cycle. The power has been calculated for the worst-case i.e. when continuously contention read operations are taking place. XMAT-Pseudo and XMAT-Replication design consume 5% and 1.15% more power compared to conventional 8T



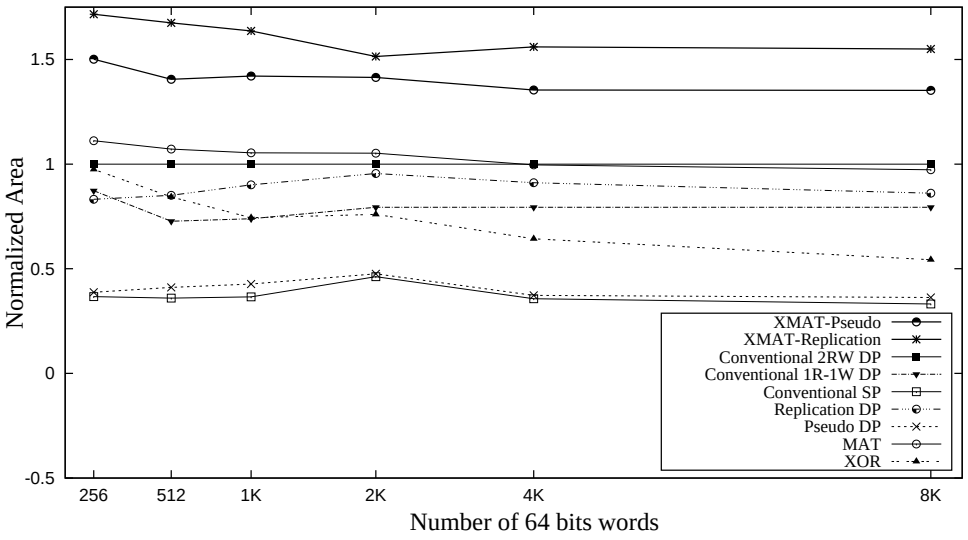
(a) Memory Access Time (t_a)



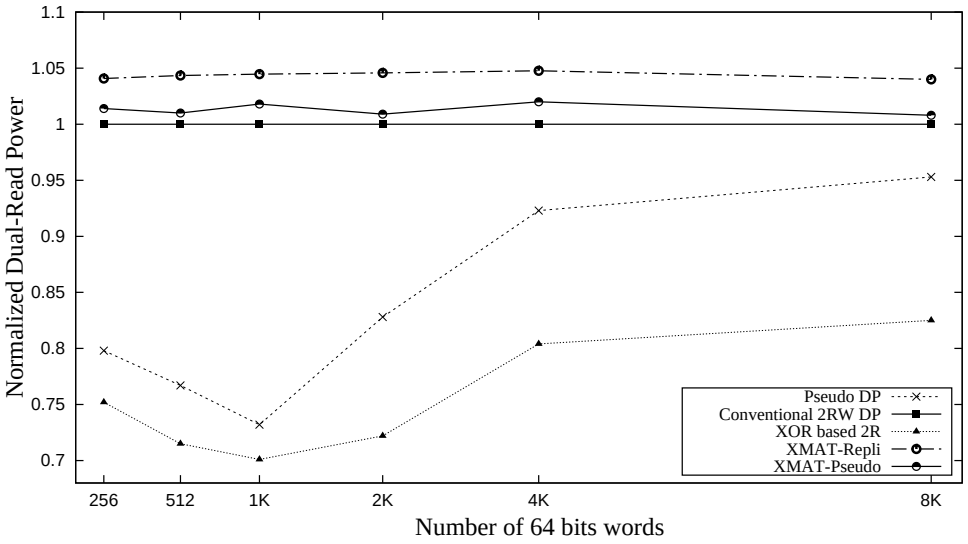
(b) Memory Cycle Time (t_{cyc})

Figure 4.1: Comparison of the conventional SP-SRAM, 8T 2RW, 1R-1W DP-SRAM, XOR based 2R, MAT based and the proposed design variants XMAT-Pseudo and XMAT-Replication on t_a and t_{cyc} in a clock cycle

DP-SRAM. The single read-write power is approximately 2.75 times higher than conventional 8T DP-SRAM because in the worst case condition each read-write operation demands a read operation on all the physical memory banks present and the concurrent



(c) Total Memory Area



(d) Total Dual-Read Power in worst cases

Figure 4.0: Comparison of the conventional SP-SRAM, 8T 2RW, 1R-1W DP-SRAM, XOR based 2R, MAT based and the proposed design variants XMAT-Pseudo and XMAT-Replication on area and 2R power in a clock cycle

write operation on the XOR bank as well as the previously EMPTY memory bank as shown in figure C.2.

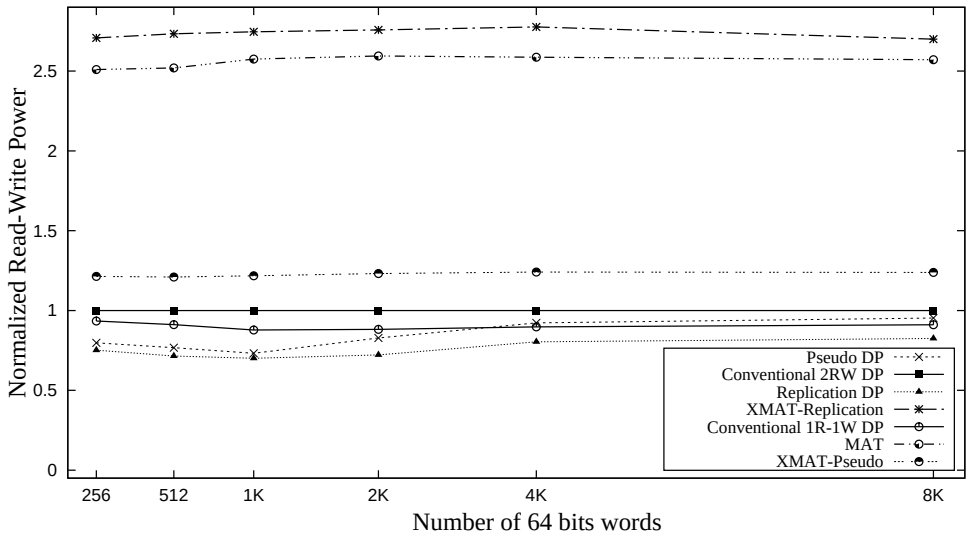


Figure 4.1: Comparison of the conventional SP-SRAM, 8T 2RW, 1R-1W DP-SRAM, XOR based 2R, MAT based and the proposed design variants XMAT-Pseudo and XMAT-Replication on 1R-1W power in a clock cycle

Probability of	**Total Power in**	
Contention	**2R, 1R-1W**	**DP-SRAM**
0.5	1.66	1
0.4	1.48	1
0.3	1.22	1
0.2	0.95	1
0.1	0.89	1

Table 4.1: Total Read-Write Power with different contention probabilities for a 2048 words, 64 bit memory

Although both XMAT implementations require considerable more area and in worst cases consume more power due to its complexity, they show an average improvement of 13% in terms of memory access times and in case of XMAT-Replication a 0.8% also in the operating frequency. In addition due to the SP-SRAM instantiations in XMAT, larger memory capacities become available eliminating read-write instabilities inherently.

4.1 Application Benchmarks in Network Routers

While XMAT consumes more power compared to the traditional DP-SRAM, the XMAT memory has been integrated into a router as a use case application.

As stated earlier the packet buffer is implemented as a large FIFO memory acting as a queue as shown in figure 4.2 in which the data written at the write pointer address is read out first. In the beginning of operation the read pointer refers to the same memory bank as the write pointer and contentions occur. However if the FIFO begins to fill, after $W_{max}/2$ entries, the read and write pointer refer to different memory banks and the read and write operation are executed without contention. Therefore for a large capacity queues(22) and high traffic volumes the probability of contention decreases. The power calculated in figure 4.1, is the worst-case power consumption in which back-to-back conflicting 1R-1W operations take place. Table 4.1 shows the sum of read and write power with different contention probabilities. It is apparent that an increase in read-write power is not that significant in applications with lesser probability of contention. A power gain of 5% compared to traditional DP-SRAM is achieved when the probability of contention is 20%.

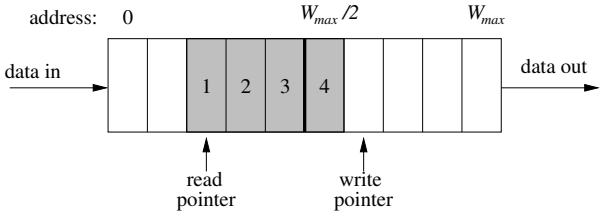


Figure 4.2: A standard FIFO

5

Conclusion

This paper proposes a 6T XMAT based 2R, 1R-1W SRAM architecture for high-performance networking applications such as packet switching, queue management to manage high data traffic throughput in Internet routers. It can execute dual-read or alternatively one read-write within a clock cycle by combining a dedicated memory bank for XOR calculations with a Memory Association Table. This design shows an improvement of 21%, 11% and 5% in access time, cycle time and power reduction respectively in case of a contention probability of 20% for a memory capable of storing 1024 words of 64 bit depth each. Since Single Port memories are used to mimic the behavior of a Dual Port memory, larger memory capacities can be instantiated and read-write instabilities inherent to Dual Port SRAM, are not observed.

References

[1] Jiayin Li, David B. Dgien, Nathan Altay Hunter, Yirong Zhao, and Kartik Mohanram. **Two-Port PCM Architecture for Network Processing**. *IEEE Trans. VLSI Syst*, **23**(10):2135–2148, 2015. 1, 7

[2] Jahangir Hasan, Satish Chandra, and TN Vijaykumar. **Efficient use of memory bandwidth to improve network processor throughput**. In *Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on*, pages 300–311. IEEE, 2003. 1

[3] Ali Keshavarzi, Dinesh Maheshwari, Derwin Mattos, Ravi Kapre, Sandeep Krishnegowda, Morgan Whately, and Sudhir Gopalswamy. **Directions in future of SRAM with QDR-WideIO for high performance networking applications and beyond**. In *Custom Integrated Circuits Conference (CICC), 2014 IEEE Proceedings of the*, pages 1–6. IEEE, 2014. 1

[4] Sundar Iyer, Ramana Rao Kompella, and Nick McKeown. **Designing packet buffers for router linecards**. *IEEE/ACM Transactions on Networking (ToN)*, **16**(3):705–717, 2008. 1

[5] Aaron Brown, Dan Chian, Nishat Mehta, Yannis Papaefstathiou, Josh Simer, Trevor Blackwell, Michael D Smith, and Woodward Yang. **Using MML to simulate multiple dualported SRAMs: Parallel routing lookups in an ATM switch controller**. In *Workshop on Mixing Logic and DRAM, Denver, CO*. Citeseer, 1997. 2

[6] Koji Nii, Yasumasa Tsukamoto, Makoto Yabuuchi, Yasuhiro Masuda, Susumu Imaoka, Keiichi Usui, Shigeki Ohbayashi, Hiroshi Makino, and Hirofumi Shinohara. **Synchronous ultra-high-density 2RW dual-port 8T-SRAM with circumvention of simultaneous common-row-access**. *Solid-State Circuits, IEEE Journal of*, **44**(3):977–986, 2009. 3

[7] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, 4th edition, 2010. 3

[8] 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**. In *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*, pages 99–100, June 2010. 4, 7

[9] Jui-Jen Wu, Meng-Fan Chang, Shau-Wei Lu, R. Lo, and Q. Li. **A 45-nm Dual-Port SRAM Utilizing Write-Assist Cells Against Simultaneous Access Disturbances**. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, **59**(11):790–794, Nov 2012. 4, 7

[10] Y. Ishii, H. Fujiwara, S. Tanaka, Y. Tsukamoto, K. Nii, Y. Kihara, and K Yanagisawa. **A 28 nm Dual-Port SRAM Macro With Screening Circuitry Against Write-Read Disturb Failure Issues**. *Solid-State Circuits, IEEE Journal of*, **46**(11):2535–2544, Nov 2011. 4

[11] Leland Chang, David M Fried, Jack Hergenrother, Jeffrey W Sleight, Robert H Dennard, Robert K Montoye, Lidija Sekaric, Sharee J McNab, Anna W Topol, Charlotte D Adams, et al. **Stable SRAM cell design for the 32 nm node and beyond**. In *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on*, pages 128–129. IEEE, 2005. 5

[12] Hiroki Noguchi, Shunsuke Okumura, Yusuke Iguchi, Hidehiro Fujiwara, Yasuhiro Morita, Koji Nii, Hiroshi Kawaguchi, and Masahiko Yoshimoto. **Which is the Best Dual-Port SRAM in 45-nm Process Technology? - 8T, 10T single end, and 10T differential**. In *Integrated Circuit Design and Technology and Tutorial, 2008. ICICDT 2008. IEEE International Conference on*, pages 55–58. IEEE, 2008. 5

[13] Ramaswamy Ramaswamy and Tilman Wolf. **PacketBench: A Tool for Workload Characterization of Network Processing**, October 13 2003. 7

[14] Hao-Yu Yang, Chen-Wei Lin, Chao-Ying Huang, Ching-Ho Lu, Chen-An Lai, Mango C.-T. Chao, and Rei-Fu Huang. **Testing methods for a write-assist disturbance-free dual-port SRAM**. *2014 IEEE 32nd VLSI Test Symposium (VTS)*, 2010. 7

REFERENCES

[15] Leland Chang, David M Fried, Jack Hergenrother, Jeffrey W Sleight, Robert H Dennard, Robert K Montoye, Lidija Sekaric, Sharee J McNab, Anna W Topol, Charlotte D Adams, et al. **Stable SRAM cell design for the 32 nm node and beyond**. In *VLSI Technology, 2005. Digest of Technical Papers. 2005 Symposium on*, pages 128–129. IEEE, 2005. 8

[16] Ramandeep Kaur, Harsh Rawat, and Alexander Fell. **A 6T SRAM Cell Based Pipelined 2R/1W Memory Design Using 28nm UTBB-FDSOI**. In *2015 28th IEEE International System-on-Chip Conference (SOCC) (SOCC 2015)*, pages 320–325, Beijing, P.R. China, September 2015. 8, 9, 11

[17] Charles Eric LaForest and J Gregory Steffan. **Efficient multi-ported memories for FPGAs**. In *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, pages 41–50. ACM, 2010. 8

[18] Charles Eric Laforest, Ming G Liu, Emma Rae Rapati, and J Gregory Steffan. **Multi-ported memories for FPGAs via XOR**. In *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays*, pages 209–218. ACM, 2012. 8, 11, 33

[19] S. Iyer and S.T. Chuang. **System and method for storing multiple copies of data in a high speed memory system**, January 13 2015. US Patent 8,935,507. 9, 11, 13, 33, 37

[20] S.G. Block, T. Zhou, M.I. Grinchuk, A.A. Bolotov, and L.D. Ivanovic. **Multi-read port memory**, September 18 2014. US Patent App. 13/833,691. 11

[21] Piyush Jain, Harsh Rawat, and Gangaikondan Subramani Visweswaran. **Cache memory system with simultaneous read-write in single cycle**, January 28 2014. US Patent App. 14/166,003. 13

[22] Jim Warner. **Packet Buffer**. 24

[23] S.S.L. Chang. **Multiple-Read Single-Write Memory and Its Applications**. *Computers, IEEE Transactions on*, **C-29**(8):689–694, Aug 1980. 37

[24] Nan-Chun Lien, Ching-Te Chuang, and Wen-Rong Wu. **Method for resolving simultaneous same-row access in Dual-Port 8T SRAM with asynchronous dual-clock operation**. In *SOC Conference (SOCC), 2013 IEEE 26th International*, pages 105–109, Sept 2013.

[25] Liang Wen, Zhentao Li, and Yong Li. **Single-ended, Robust 8T SRAM Cell for Low-voltage Operation**. *Microelectron. J.*, **44**(8):718–728, August 2013.

[26] Dao-Ping Wang, Hon-Jarn Lin, and Wei Hwang. **A Two-Write and Two-Read Multi-Port SRAM with Shared Write Bit-Line Scheme and Selective Read Path for Low Power Operation**. *Journal of Low Power Electronics*, **9**(11):9–22, Nov 2013.

[27] Brian Tierney Michael Smitasin.

[28] Charles Eric LaForest and J Gregory Steffan. **Efficient multi-ported memories for FPGAs**. In *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, pages 41–50. ACM, 2010. 31

Appendix A

Pseudo Dual-Port Memory

In this section the TDM based approach for designing the DP-SRAM is discussed. A TDM approach can be applied to an SP-SRAM to reuse its access ports in a different time slot. For any memory design the speed of the memory can be traded for the area by operating the memory at an internal clock frequency that is a faster multiple of the external clock frequency, giving the illusion of having more ports than are actually supported. Thus, it is also called Pseudo DP-SRAM or multi-pumping as it gives an illusion of being dual-port. Multi-pumping can be applied to any memory design to multiply its read and write ports.

In one clock cycle of the external clock, the single-ported memories can perform 2 memory accesses, which from the system's perspective, is equivalent to having a dual-port memory. This architecture is shown in figure A.1 for a 1K×64 memory capacity. To create a 1K×64 pseudo dual-port memory an SP-SRAM of similar size is instantiated. Two concurrent accesses are broken down into two set accesses. The first set of access is presented to the SP-SRAM immediately, while the second set is stored in registers. In the second-half of the external clock cycle, the results of the first set of access are stored in registers, while the second set of accesses are presented to the SP-SRAM. At the end of a complete system cycle, both sets of accesses are complete.

A number of designs utilize multi-pumping to gain additional access ports while keeping area overhead minimal. It uses the same capacity memory as SP-SRAM with extra control logic and registers to steer the data in and out of the SP-SRAM. The

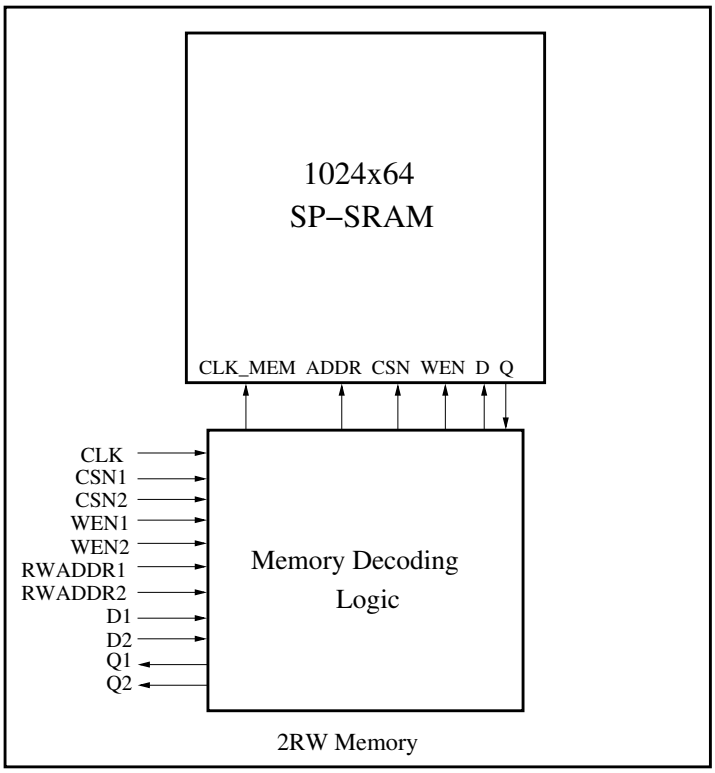


Figure A.1: Block Diagram for the TDM implemented 1K×64 DP-SRAM

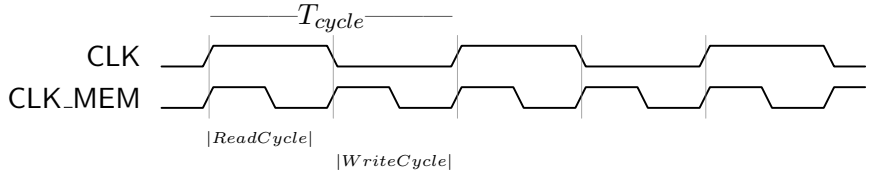


Figure A.2: Waveform of the Pseudo DP-SAM scheme

limitation, however, is that each increase in the number of ports dramatically reduces the maximum external operating frequency of the memory. The system needs to run slow enough so that the memory can run at multiples of the system clock. This constraint usually makes it difficult to run the memories at more than twice the system clock. Figure A.2 shows the operations in each cycle.

Appendix B

Replication based Dual-Port Memory

This section discusses the replication based 1R-1W DP-SRAM design. (28) proposed the first multi-ported memory design based on replication of the SP-SRAM and Live Value Table (LVT) to keep track of the recently written data. The basic idea of an LVT design is to augment a banked design with the ability to connect each read port to the most-recently written bank for a given memory address. To create a replication based DP-SRAM of W_{max} words, two SP-SRAM of W_{max} words and a single-bit MAT or an LVT of W_{max} words is instantiated as show in figure B.1. The memory blocks for each read and write port are replicated, while keeping read and write as separate ports, and an LVT or MAT as discussed in section 3. When a read operation is executed, the read port looks up the memory address in the MAT, which returns the previously-written identity of the memory bank. This is used to select the most-recently-written value from one of the memory banks.

Actually, the LVT itself is a multi-ported SRAM with the same memory words and number of writing ports as the implemented multi-port memory. However, since the LVT stores only bank identity, the data width of the LVT table is only $log_2 n$ where n is the number of banks, which is equal to the number of writing ports. Since an LVT is a narrow, multi-port memory, it is implemented with registers and is pure logic based. Furthermore, the LVT doesn't have write data, instead it writes a fixed bank ID for each port as described in figure B.1. Since there are two banks it stores either 0 or 1

as bank identity.

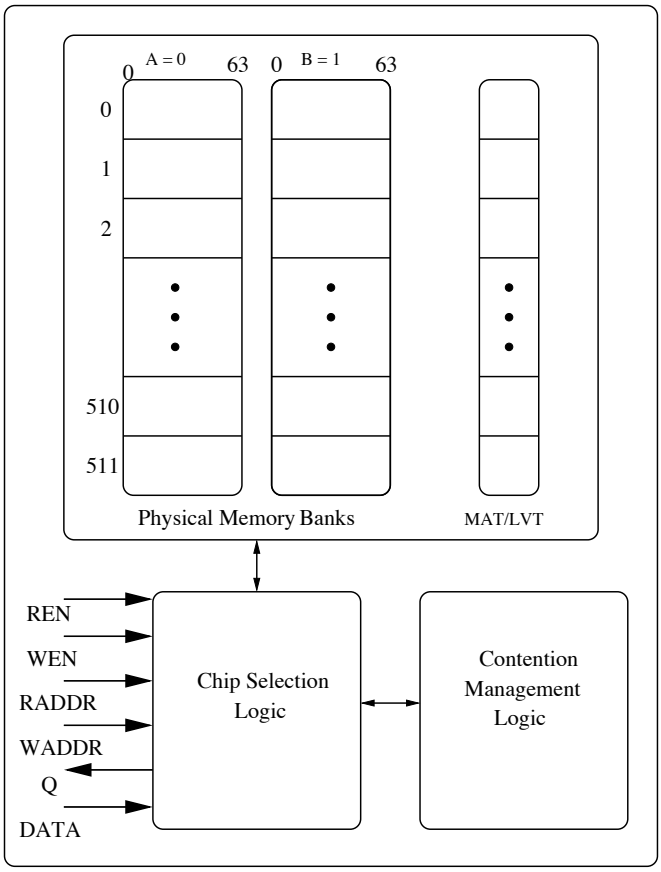


Figure B.1: Block Diagram for Replication based 1024×64 DP-SRAM

Replication based designs implements memory more efficiently and have an operating frequency closer to that of the SP-SRAM itself. Thus, it can achieve higher frequencies than the multi-pumping approach. While this approach does not assume an external clock at half the possible memory clock, the replication increases the area requirements. In addition only half of the total memory capacity is used at any given point in time.

Appendix C

XOR based 2R/1W Design

In this section an XOR based design to overcome the drawbacks of the previous approaches discussed in the last section. (18) and (19) introduce an XOR based approach for designing a multi-read memory. However, (18) describes it for a Field Programmable Gate Array (FPGA) which uses a Block Random Access Memory (RAM) that is inherently dual-port and (19) makes use of a 4-ported XOR memory to design a multi-read memory. Using multi-port memories further increases the memory area and adds complexity to the memory design in addition to read-write instability. Thus, a favoured solution will be to design a multi-port using only a single-port memory in the core.

The design, illustrated in figure 3.2, utilizes only 6T single-port banks to realize a dual-read, single write (2R/1W) memory. It consists of two separate ports, 1 Read-Write and 1-Read only port with different sets of address (RWADDR and RADDR), control (CSN, WEN and REN) and output (Q1 and Q2) lines and a single data line (D) to write to the memory. Both the ports are clocked at the same frequency using a single clock line (CLK). The proposed dual-port memory design comprises of three single-port memory banks A, B and an XOR bank, each of which is half the capacity of the memory to be built. For instance, to generate a dual-port memory capacity of 1K words (W_{max}) of 64 bits each (shown in figure 3.2), the three single-port memory banks would be of the size of 512 words ($W_{max}/2$) of 64 bits each. Each row of the XOR bank contains the data from both the banks A and B in XOR'ed form. Bank A contains data of the first 512 words (0-511) and bank B stores data of rows 512-1023.

C. XOR BASED 2R/1W DESIGN

This design exploits the property of XOR to effectively avert memory access conflicts.

Let

$$C = A \oplus B$$

then,

$$A \oplus C = B, \qquad \text{because } A \oplus (A \oplus B) = B$$

and

$$B \oplus C = A, \qquad \text{because } B \oplus (A \oplus B) = A$$

Classically, if two access requests point to the same SP-SRAM bank, it was solved by either memory halts or memory stalls (queuing the subsequent accesses). In this case, the read or write to the memory would no longer be guaranteed to execute in a fixed time. However, the proposed design boosts the read-write parallelism by creating multiple copies of the same data in two different banks. The contention management logic is operatively coupled to the A and B memory bank via the memory decoding logic to help resolve the memory access conflicts as shown in C.1. It does so by determining the locations to retrieve data from the single-port memory cells, thereby avoiding collision between concurrent memory accesses. However, the dual-read and write operations take time and thus, by construction to avoid bank conflicts during operations the output data from the proposed DP-SRAM is pipelined and is stabilized after 1 clock cycle latency. The result of the read is stored in registers and is used in later cycles.

When the first memory address (RWADDR) and second memory address (RADDR) execute two read accesses on separate memory banks, both the reads are directly made from the respective banks. For example, if two concurrent read operations are received for the addresses 255 and 1011 which are located in different memory banks the reads can be made directly from SP memory bank A and B respectively.

However, when the first (RWADDR) and the second memory address (RADDR) are associated with the same memory bank, the first data is retrieved from that memory bank and the second data is reconstructed using data from the second and the XOR memory bank. For instance, let two reads access the addresses 522 and 1002. In this

case, it is not possible to get the output of the read in a single clock cycle due to memory stalls. Thus to avoid the stall, first set of data is read-out from the memory address directly from bank B and the second read data is reconstructed using bank A and the XOR memory bank. The output of both the reads is pipelined and the read-out data is latched in the next cycle. This kind of approach essentially means that we are creating multiple copies of the same data but in a compressed form instead of directly replicating it. This kind of multiple storage of same data comes at a cost of writing the data twice:

- Writing data to a memory corresponding to RWADDR and reading the data from

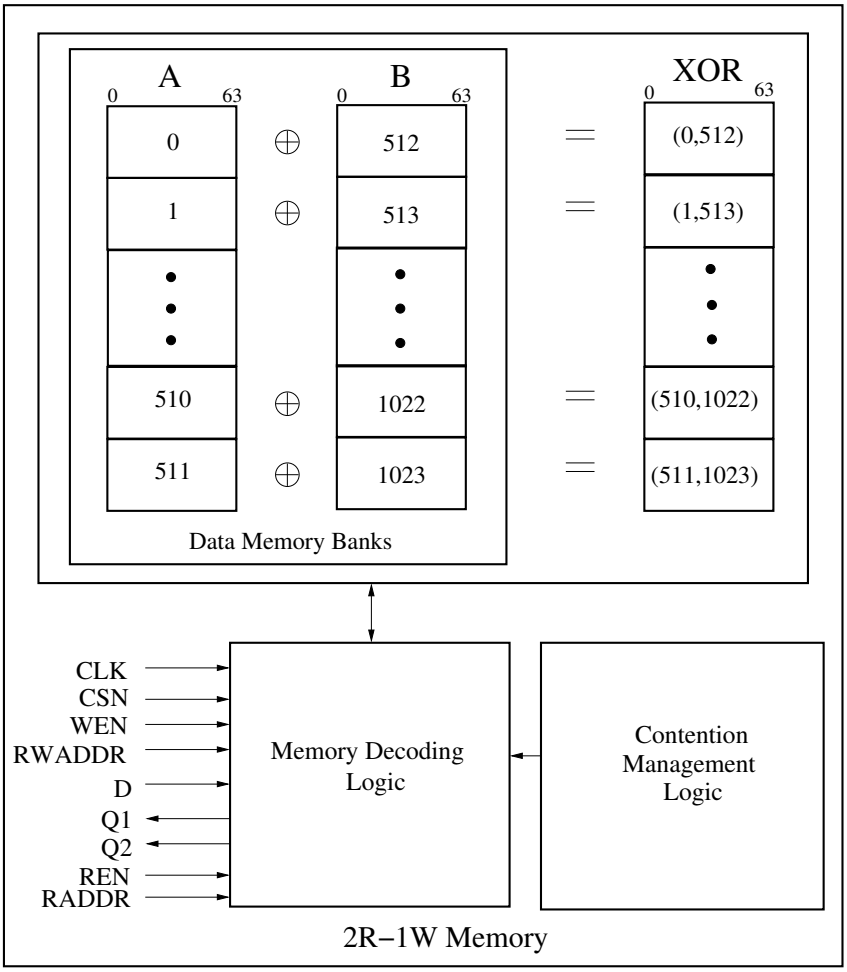


Figure C.1: Proposed Design Block Diagram

the address RWADDR $+ \left|W_{max}/2\right|$. This data is then XOR'ed with the new data.

- Writing the XOR'ed data in the single-port XOR memory bank.

Dual-Read operation from the same memory bank will include the following operations to be executed:

- Reading data from the memory bank corresponding to the address RADDR and also, reading from memory address RWADDR $+ \left|W_{max}/2\right|$.
- Reading XOR memory bank corresponding to the RWADDR address and XOR them to obtain the output of the second read.

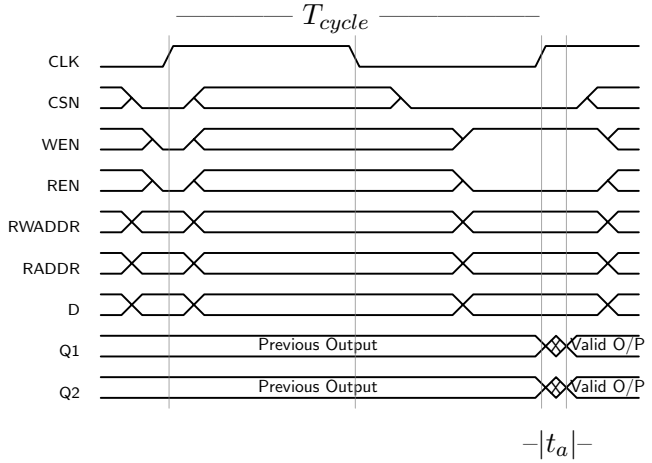


Figure C.2: A single cycle waveform of proposed 2R-1W memory with pipelined output

The waveform in figure C.2 illustrates a single-cycle operation of the proposed dual-port design. The control, data and address lines have a finite setup and hold time which depends on how fast the chip selection logic is decoding the operations to be performed on the memory banks, after input from the contention management logic. After a finite access time (t_a) or clk-to-Q time, the pipelined outputs on Q1 and Q2 are stabilized and latched. The advantage of using a pipelined output is two-fold. First, it helps to reduce the penalty in the access time and cycle time due to latching of the output in the next consecutive cycle. Secondly, the use of multi-port memories is avoided as

opposed to the memory design in (19).

The design however, has a limitation that a concurrent read-write operation cannot be executed in a single memory cycle. This is due to the fact that during a write operation all the memory banks A, B and the XOR bank are in use. There is no idle bank to read from. The restriction of writing capability to one port is not necessarily a handicap situation as mentioned in (23).

Declaration

I herewith declare that I have produced this work without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such.

The thesis work was conducted from July 2014 to June 2015 under the joint supervision of Dr Alexander Fell at IIIT Delhi and Harsh Rawat at STMicroelectronics.

New Delhi