DESIGN OF SRAM-BASED 8T-CELL FOR MEMORY ALIAS TABLE

Saleh Abdel-Hafeez[1, 2], Sanabel Otoom[1] and Muhannad Quwaider[1]

[1]Jordan University of Science and Technology, Dept. Computer Engineering, Irbid 22110, Jordan
[2]Sabbatical at Qassim University, Dept. of Computer Eng., College of Computer, Qassim, Buraydah, Saudi Arabia

ABSTRACT

Memory Alias Table exploits a major role in Register Renaming Unit (RRU) for maintaining the translation between logical registers to physical registers for the given instruction(s). This work presents the design of the memory Alias Table based on the 8T-Cell with multiport write, read, and content-addressable operation for 2-WAY three operands machine cycle. Results show that four read ports operate simultaneously within a half-cycle, while two-write ports operate simultaneously within the other half-cycle. The operation of content-addressable with two parallel ports is managed during the half-cycle of the read phase; thus, the three operations occur within a single cycle without latency. HSPICE simulations conduct 32-rows x 6-bit with 21T-Cell memory Alias Table that has 4-read ports, 2-write ports, and 2-content-addressable ports using a standard 65 nm/1V CMOS process. Simulations reveal that the proposed design operates within a one-cycle of 1 GHz consuming an average power of 0.87 mW.

KEYWORDS

Content-Addressable,8T-Cell SRAM, 2-WAY Instructions Cycle, Memory Alias Table, Register Renaming Unit

1. INTRODUCTION

The hardware of instruction-level parallelism constitutes several out-of-order units within the pipeline structure. These units facilitate the execution of multiple instructions within the stages of the pipeline to reduce the dependencies between instructions; and thus, improving the overall Amdahl's law measure performance [1]-[4]. One of these stages of the pipeline is the instruction dispatch unit that allocates mapping of the logical to the physical registers since the computations were held in physical implementations. Register Renaming Unit (RRU) holds this mapping as well as deallocates back the physical to the logical registers, wherein the number of the physical registers is greater than the number of the logical registers [5]-[7].

Register renaming unit constitutes primarily three tables with several comparators and priority encoders to orchestrates the mapping between logical and physical registers; and thus, increase the performance of out-of-order (OoO) speculative execution unit. The three tables are Alias Table, Physical Table, and Architectural Table [8]-[11]. The Alias Table is indexed by the logical register number and holds the mapping to the physical registers. The Physical Table shows the availability of each physical register. That is, if the allocated bit is set, the corresponding physical register is being mapped by a destination logical register of the assigned instruction;

David C. Wyld et al. (Eds): NLPTA, EDU, DSA, IoTE, VLSI, DPPR, ACITY, AIAA, CNDC - 2021
pp. 87-106, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111907

otherwise, the corresponding physical register is free and can be allocated by a destination logical register of incoming instructions. The Architecture Table records all completed evaluated physical registers and their actual data.

Register renaming tables are considered high-cost design since each table requires several and multi-operations within a single cycle; besides, tables require to maintain low power consumption and short critical path for each operation [12]. Consequently, as the number of instructions per cycle increase and parallelisms becomes prominent in modern processors, the tables required to hold more parallel operations, and thus, having more complex circuits [13]. Therefore, the memory array of multiport SRAM cells considers an essential element in constituting the tables array, in which the cell structure provides several parallel operations with relatively reduce circuit complexity [14][15]. Still, comparators along with priority encoders in a form of prefix tree structure hold a large burden of design overhead area, wherein content-addressable memory (CAM) cell can provide a major role in reducing these complexities of design circuitry as well as maintaining the required operation efficiently [16]. Consequently, the Register Renaming tables have been investigated in several realizations of organizational memory structures. Some works focus on SRAM-based rather than CAM-based implementations for more scalable and energy-efficient at the cost of extra-overhead comparators and priority encoders [14][15]. Other recent works leverage the benefit of CAM to match the contents in a parallel fashion, such as hybrid SRAM-CAM structure [17][18] that result in reducing the latency cycles; and thus, improve throughput.

Another essential factor in the memory array is the type of cell structure. Some of the SRAM-base structures use the 6T-Cell [19], where the read and write ports share the same input-output bus; and thus, narrowing the noise margin. Subsequently, a highly sensitive sense amplifier with constant bias current is essential to boost the speed of the read operation. Other use the 9T-Cell structure [20], in which the back-to-back inverters of the write access port are disconnected by intermediate transistors to reduce the contention during write access mode. Thus, improving the power consumption for the write operation. However, this factor comes for the cost of double the size of the memory array as the authors mention due to the high cost of control logic that requires to generate the appropriate signal voltage level for the intermediate cell's transistor. Further attempts of SRAM-base cell is the use of 10T-Cell [21] and 7T-Cell [22] that are often used for ultra-low supply voltage in the order 0.5 V or even less for low power consumption with low operating frequency. A comparative study of the abovementioned memory cells exploits the characteristics of each cell and its effect on the overall memory array [23][24].

Nowadays, the 8T-Cell commonly use in most processors and graphics semiconductor companies in the field of CACHE and Shared Memory [25][26]. The 8T-Cell with sperate read and write ports structure considers an essential component for Register Renaming tables, wherein simultaneous read and write operations at different rows are needed. Therefore, multiport for write and read can realize fast access time with a full rail noise margin and low power consumptions [27][28]. Additionally, the 8T-Cell can be adapted to verities and a wide scale of semiconductor technologies, while maintaining its low-power and high-speed features [29]. Therefore, the 8T-Cell is considered in this work for constituting the Register Renaming tables. The 8T-Cell is reconfigured for multiport read and multiport write; besides, the CAM circuitry has added to the 8T-Cell forming a hybrid SRAM-CAM cell. Further facilitation of 8T-Cell reduces design time to market by introducing an automation algorithm from schematic layout to physical layout with the support of 8T-Cell library standard cell components [30][31].

As a result, the prior work on the Register Renaming unit architectural development [32] is extended to pay attention to the Alias Table circuit development as the most complex circuitry, which exploits several simultaneous operations within a single cycle [33]. The Alias Table

constitutes a memory array of SRAM-CAM cells, where the SRAM is based on the 8T-Cell and the CAM is based on the two pass-gate transistors forming XOR logic connected to another two pass-gate transistors of the match line. The SRAM part of the cell has four read ports and two write ports, while the CAM part has two content-addressable ports, results in a total of twenty-one transistors cell. The match line is pre-charged to the power supply, where any mismatch between cell content and the coming data from the mask register drops the match line voltage to the ground. Thus, if all cells in a particular row match the data of the mask register, the match line for that particular row preserves the power supply pre-charge value, which enables the Tri-state buffer to release the row index to the output match port. In summary, the proposed Alias Table circuit has the following key features:

- The Alias Table leverages the 8T-Cell SRAM structure that is suitable for continued low-cost CMOS technology with high-speed and low-power operations.
- The Alias Table conducts write, read, and content-addressable match index within a single clock cycle.
- The Alias Table provides four parallel read operations.
- The Alias Table provides two parallel write operations.
- The Alias Table provides two parallel content-addressable operations.
- The content-addressable operation compares all rows in parallel with the mask register and releases the associated match index address.

The remainder of this paper is organized as follows. Section II discusses the design of the SRAM-CAM cell circuit structure. Section III realizes the overall Alias Table circuit architecture with the proposed memory array of SRAM-CAM cells. Section IV gives the estimated overall critical path delay for each operation, while section V provides the HSPICE simulations and verifications. Additionally, section VI illustrates some performance features along with some comparison of recent works. The conclusion is given in Section VII.

2. SRAM-CAM CELL CIRCUIT

The cell given in Figure 1 is designed based on the well-known 8T-Cell circuit with the addition of a new CAM structure. The cell combines three circuits operations that are - four parallel read ports, two parallel write ports, and two parallel content-addressable ports. All three operations are independent of each other's and have separate ports, given a rail-to-rail noise margin operation that is attractive for continued technology scaling with low power supply voltage and high-speed operation. Moreover, the ports within the same operation are activated in parallel since they are gated with separate pass-gate transistors. TABLE 1 depicts the input/output signals' abbreviations and descriptions for the 21T-Cell.

The two write ports have a similar circuit structure to the 8T-Cell write port, where the back-to-back inverters are flipped by input bit and inverted input bit. Subsequently, two input bits are gated by two independent pass-gate transistors from each side of the back-to-back inverters given the write circuit with eight transistors count. Each write port is associated with two pass-gate transistors that were gated by a write decoder. Therefore, the two separate write decoders activate the two separate write ports, where each port has a separate data bit.

TABLE 1. 21T-Cell signals definitions and abbreviations

Input-Output Signals	Representations
DA	First Input
DA!	First Inverted Input
DB	Second Input
DB!	Second Inverted Input
WDA	First Write Word Line Decoder
WDB	Second Write Word Line Decoder
RDA	First Read Word Line Decoder
RDB	Second Read Word Line Decoder
RDC	Third Read Word Line Decoder
RDD	Fourth Read Word Line Decoder
OA	First Output
OB	Second Output
OC	Third Output
OD	Fourth Output
MLA	First Match Line
MLB	Second Match Line
MA	First Mask Register
MA!	First Inverted Mask Register
MB	Second Mask Register
MB!	Second Inverted Mask Register
CLK	System Clock

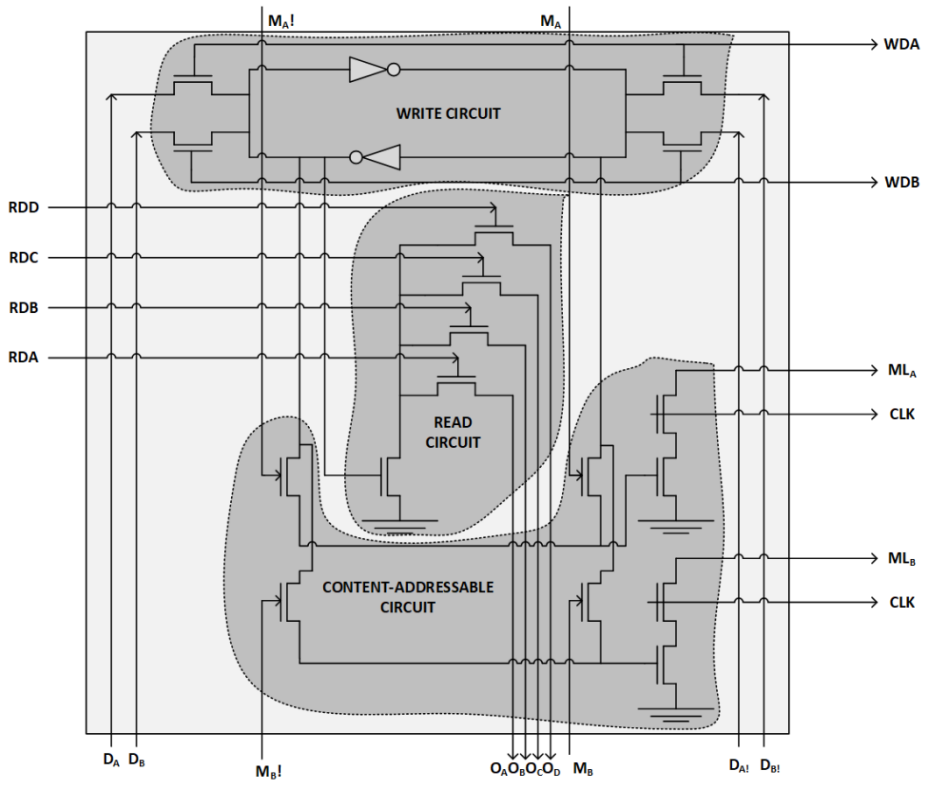

Figure 1. The circuit structure of the proposed SRAM-CAM 21T-Cell

Subsequently, two different data bits can be written to two different cells simultaneously of the same column of the memory array. A priority encoder is realized at each two input data bits in order to assure no contention of bits values is stored simultaneously at the same cell. Thus, precludes any possibility of two different data bits to be written simultaneously at the same cell.

The four read ports exploit the same structure of the 8T-Cell read port, and they can operate in parallel since each port has a separate pass-gate transistor gated with a separate decoder. Subsequently, the read circuit has five transistors. The discharge-transistor for all the four pass-gate transistors is designed with a larger width ratio than in the 8T-Cell single read port since it has more diffusion capacitances at its drain node. The four output ports are pre-charged to the power supply voltage through the output buffers. Once the read ports' decoders are enabled, the output ports are either discharged to the ground or preserved the pre-charged voltage, based on the cell stored value.

The CAM circuit compares the cell value with an inverted mask bit value and vice versa by using two pass-gate transistors with a common drain forming an XOR logic structure. Additionally, the second content-addressable port has a similar structure, but the cell value is compared with the second mask bit that is related to the second mask register. For brevity of discussion, the detailed operation of one port content-addressable is presented since the second port has similar behavior. The common drain out of the XOR logic is directed to the gate of discharge pass-gate transistors, which is connected with the match line through a gated clock pass-gate transistor. During the low

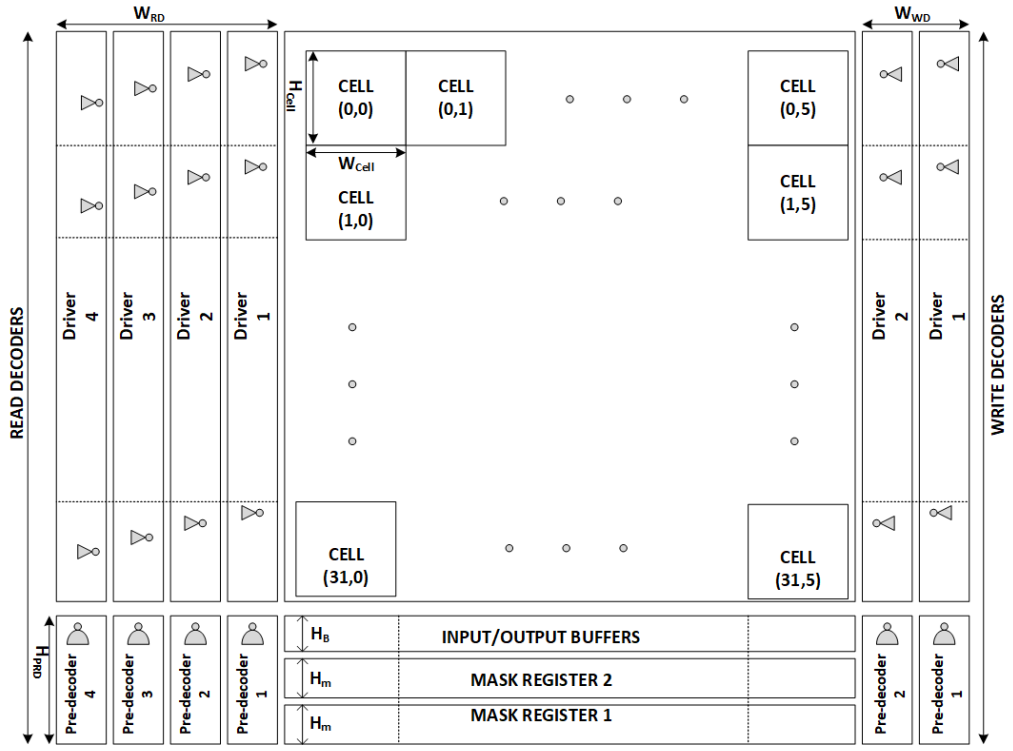

Figure 2. The architectural diagram of the proposed Alias Table

Phase of the clock, the match line is pre-charged to the power supply voltage. Consequently, during the high phase of the clock, the match line is either discharged to the ground or preserve the pre-charge value.

As a result, the cell contains two write ports, four read ports, and two content-addressable ports with twenty-one transistors count. The cell used similar geometry sizes (L/W) for the well-known 8T-Cell for write and read operations with similar layout features of combining three metals materials that detail some highlights on the cell layout structure and geometry sizes [34][35].

3. ALIAS TABLE CIRCUIT ARCHITECTURE

The study for the Register Renaming unit shows a machine with two simultaneous fetch instructions, where each instruction has three operands, describes the requirements for Alias Table design. The proposed Alias Table constitutes the memory array of size 32-row X 6 bit. Such that, the 32-row presents the logical indices, while the 6-bit presents the physical indices. Some machines present 7-bit instead of 6-bit per each row in order to allocate the last bit for availability [16][18]. However, in this context,the 6-bit is used for the brevity of discussion and ease of understanding. Each cell in the memory array is a hybrid SRAM-CAM of 21T-Cell describes in section II since the Alias Table would require four simultaneous reads, two simultaneous writes, and two simultaneous contents-addressable. An additional constraint is required to maintain all three operations within one cycle of the pipeline stage. Figure 2 demonstrates the topology block diagram of the overall Alias Table realizes four read decoders, two write decoders, a memory array of 21T-Cell, two mask registers, and input-output buffers with priority encoders.

As clearly illustrated in Figure 2, depicting the geometry sizes of the 21T-Cell, which is the height (H_{Cell}) and the width (W_{Cell}), the memory array can be estimated as the height of 32 x H_{Cell} and the width as 6 x W_{Cell}. The four read decoders' drivers are aligned with the pitch size of the cell height (H_{Cell}), where the width of the read decoders (W_{RD}) can simply be estimated. Similarly, the two write decoders' driver width (W_{WD}) is depicted.

On the other hand, each bit width of the mask register is aligned with the pitch cell width (W_{Cell}), while the mask bit height (H_M) is estimated. Similarly, each input/output buffer is aligned with the pitch cell width and the buffer height (H_B) is estimated. The read pre-decoders height (H_{PRD}) is aligned with the height of the input/output bus of the array (H_M+H_B), where the width of the read pre-decoder is aligned with the decoders width (W_{RD}) as clearly shown in Figure 2. Similarly, the write pre-decoder width is aligned with the write decoder width (W_{WD}), and the height of the write pre-decoder is aligned with the input/output bus of the array (H_M+H_B). As a result, the total layout geometry of the Alias Table circuitry can be estimated and measure, as well be depicted in simulation section V for the given technology parameters.

4. CIRCUIT AND TIMING ANALYSIS

The circuit detail of each operation is considered in the following subsections along with some highlights on the time delay model for each operation by considering only the critical path. Subsequently, the acronym "TD_{NAME}" is referred to the time delay, where the subscript "NAME" is referred to the circuit's component involved in the critical path. Furthermore, the scalability of the approach is evaluated using the timing of all critical paths concerning a one-unit gate delay (GD), which provides an analysis that is independent of technology factors for direct comparison purposes [36]. In simulation section V, the 65 nm/1 V technology parameters and HSPICE simulator are used to cross-verify the derived critical path for each operation. The proposed design uses only basic CMOS logic gates structure with basic width/length sizes to provide design layout clarity and cost-effectiveness for continued technology scaling.

4.1. Read Circuit Architecture and Timing

The Alias Table shown in Figure 3 presents the read portion circuit architecture. Each cell in the memory array has four read ports that can be accessed in parallel by the four read decoders due to a separate pass-gate transistor on each port. Therefore, all the cells within a column share the same four read ports lines; thus, each column has a bus of four output lines. Each output line has an output buffer that has a pre-charge PMOS transistor, given a total of twenty-four output bus lines. During the low phase of a clock, all the twenty-four output bus lines are pre-charged by PMOS transistors to a supply voltage, which is known as a pre-charge phase. Inversely, all the twenty-four output bus lines are disabled from pre-charging during the high phase of a clock, which is known as an evaluate-phase. During the evaluation phase, each read decoder enables a one-row port of the memory array, and thus, the four decoders enable four-row ports of the memory array. The four-row ports can be a particular row of the memory array or separate rows of the memory array since the read decoders along with the read ports are independent of each other.

Therefore, the read access time is started by the rising edge of the clock (CLK), such that, the total read access time for fetching the data from a particular row's cells to output bus bits is estimated
as follow:

$$TD_{Racc} = TD_{Decoder} + TD_{Row} + TD_{Cell\text{-}line} + TD_{Buffer} \quad (1)$$

That is, the TDDecoder is the decoder time delay since all four decoders are activated in parallel and each decoder has a separate 5-bit input address bus. Each read decoder has a simple structure of pre-decoders and simple inverter drivers, in which the pre-decoder has a simple prefix-tree of NAND-Gates where the last gate is gated with the clock. Thus, the decoder time delay is

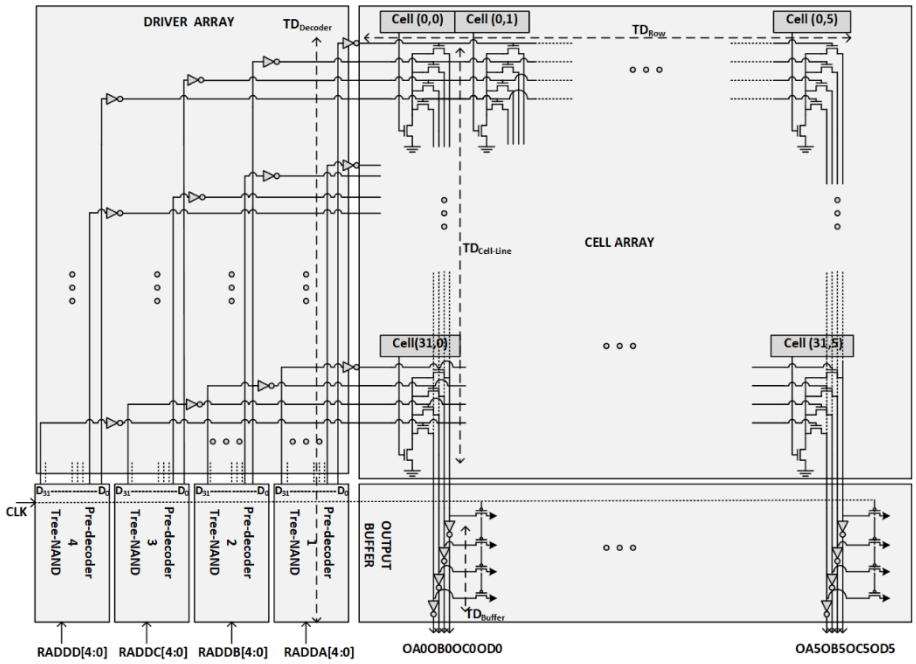

Figure 3. The proposed four parallel ports read circuitry with critical path

approximated into $TD_{Decoder}$ = 5 GDs to consider the worst-case scenario; wherein the decoder delay is usually designed to be even less than 5 GDs for a similar memory size [34]. Furthermore, each read word line is driven by a simple inverter from the decoders' drivers' circuit, and each read wordline has a parasitic of six capacitive gates as clearly shown in Figure 3. Consequently, the read wordline time delay can be approximated into TD_{Row} = 3 GDs as a worst-case scenario.

Once the four read word lines are activated, the data is fetched into the output lines; and thus, the worst-case scenario is to propagate the data to the output lines passing through thirty-two rows, where each row has one parasitic capacitance of type depletion region of NMOS transistor as illustrated in Figure 3. Thus, each bit line of the output bus has a total of thirty-two type parasitic capacitances of the depletion of NMOS transistors. HSPICE simulations shows the depth can be approximated with $TD_{Cell-line}$ = 8 GDs as a worst-case scenario. The bit line output buffer constitutesa simple inverter with a simple pre-charge PMOS transistor, or more often a simple latch to holdthe data for a complete clock cycle for other interfacing components. Subsequently, the latch accesstime is only TD_{Buffer} = 1 GD. As a result, the total read access time estimated by Eq. (1) is:

TDRacc = 5 GDs + 3 GDs + 8 GDs + 1 GDs = 17 GDs.

4.2. Write Circuit Architecture and Timing

The write circuit shown in Figure 4 exploits the Alias Table for the write operation. The detailed of the write portion of the cell in the memory array is addressed, which contains two parallel

write ports that are activated by two separate write decoders. Therefore, each column has two input data bit lines, giving a total input data bus of twelve-bit lines. A priority encoder depicted in Figure 5 streamlines the input data at each column to prevent different data written to the same cell; however,

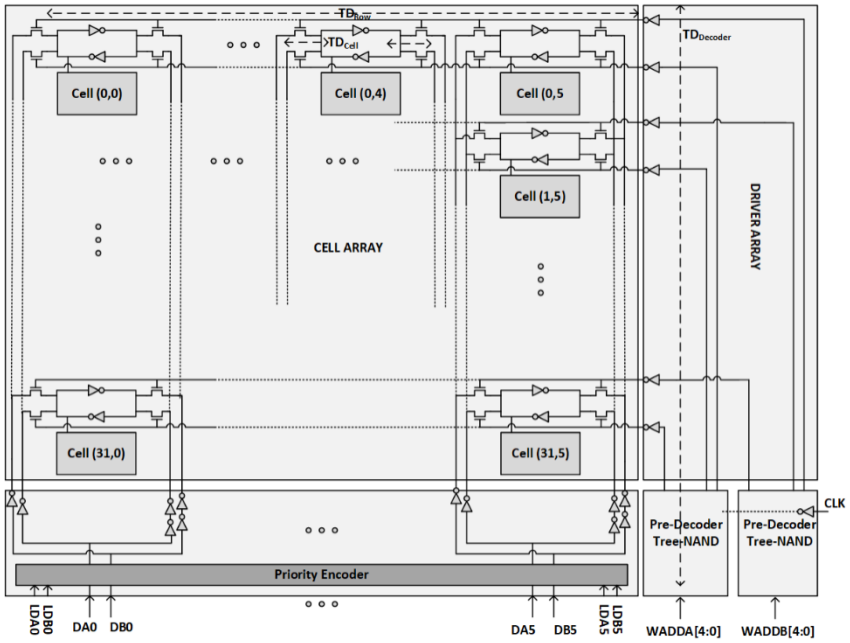

Figure 4. The proposed two parallel ports write circuitry with critical path

the priority encoder permits different data written in different cells. Thus, a total of six priority encoders exploits the input data buffer to leverage the advantages of two write operations to two different rows simultaneously and prevent the contention of different data to the same row.

The write operation begins during the low phase of the clock, where the access time can simply be derived by the total number of GDs as follow:

$$TD_{Wacc} = TD_{Decoder} + TD_{Row} + TD_{Cell} \quad (2)$$

That is, the $TD_{Decoder}$ is the decoder delay time, which is similar to the read decoder structure. Thus, the write decoder delay time is $TD_{Decoder}$ = 5 GDs as a worst-case scenario. Furthermore, each writes word line is associated with parasitic capacitances of type capacitive NMOS gates of count twelve since there are six columns of the memory array as clearly shown in Figure 4. Consequently, the write word line time delay can be approximated into TD_{Row} = 6 GDs as a worst-case scenario. Moreover, each bit of input data is propagated through the column of data bit lines waiting for the particular rows to be enabled by the write decoder, where the propagation time occurs in parallel with the decoder time. Therefore, the write delay cell time is only the count time to flip the cells of the row, which is approximated as TD_{Cell} = 1 GD. As a result, the write access time using Eq. (2) is:

TD_{Wacc} = 5 GDs + 6 GDs + 1 GDs = 12 GDs.

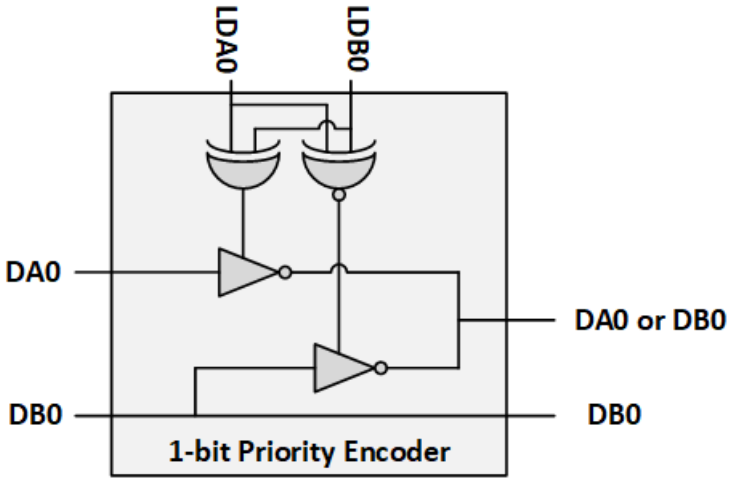

Figure 5. The proposed Priority Encoder circuitry

4.3. Content-Addressable Circuit Architecture and Timing

The Alias Table in Figure 6 illustrates the content-addressable circuit operations, which constitutes the CAM portion of the cell in the memory array, the match lines, the two mask registers, and the Tri-state output buffers. Two match lines per row are supposed to be compared simultaneously with all bits of the mask registers; however, only one match line per row is presented for the brevity of discussion. Each match line on each row is pre-charged to a power supply by a PMOS transistor during the low phase of the clock. During the high phase of the clock, each bit of the mask register is broadcasted to all CAM cells in the associated column of the memory array by two mask lines. Subsequently, each CAM cell in the column examines the mask bit against its content, and thus, the CAM discharges or preserves the pre-charge voltage of the associated match line based on the comparison with the cell's content. If all CAM cells in the row hold the match line voltage; then, the mask register matches the content of that particular row cells. Thus, the match line of that particular row enables the associated memory array index through a Tri-state buffer as illustrated in Figure 6. This matched index propagates to the match-output port of the Alias Table.

The content-addressable of the Alias Table evaluates the match address index at the high phase of the clock, giving the access time as follow:

$$TD_{Cacc} = TD_{FF} + TD_{Mb\text{-}line} + TD_{XOR} + TD_{Match\text{-}line} + TD_{Tri\text{-}state} + TD_{Buff\text{-}out} \quad (3)$$

That is, the TD_{FF} is the D-Type Flip-Flop access time of mask register, TD_{XOR} is the cell's pass-gate access time, $TD_{Tri\text{-}state}$ is the Tri-state buffer access time, and $TD_{Buff\text{-}out}$ is the output buffer access time, which all are more-less have the same access time delay TD = 1 GDs. The delay of the mask bit lines propagated through each column, and thus, passing through thirty-two cells of the parasitic type NMOS gate, in which two parasitic NMOS gate per cell since there are two match lines per cell. Consequently, each match bit lines heavily inherited with sixty-four parasitic gate capacitances. However, the mask bit lines are driven from the mask register, which has an output buffer on each of the mask bit line. Therefore, the estimated delay time for the mask bit line is $TD_{MB\text{-}line}$ = 8 GDs. On the other hand, each match line has six parasitic capacitances of type depletion NMOS transistors, where the estimated delay is $TD_{Match\text{-}line}$ = 6 GDs since the worst-case delay is to discharge the match line by only one cell through two in

series pass-gate NMOS transistors. As a result, the content-addressable access time delay is derived from Eq. (3) as:

TD_{Cacc} = 1 GD + 6 GD + 1 GDs + 8 GDs + 1 GDs + 1 GDs = 18 GDs.

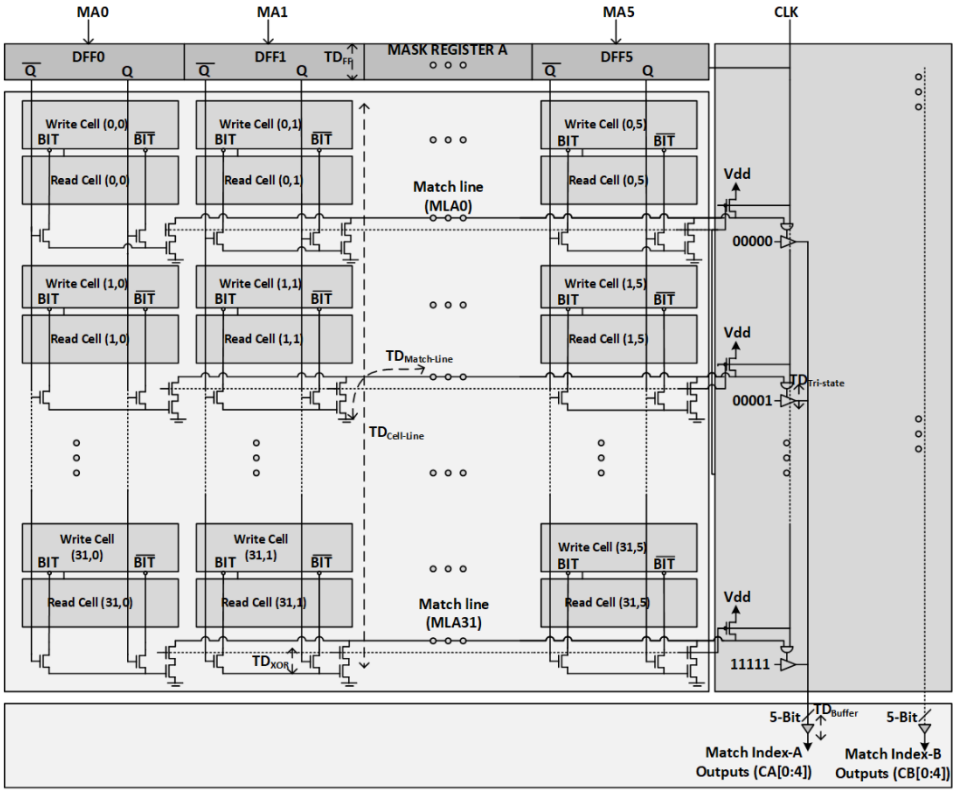

Figure 6: The proposed two parallel ports content-addressable circuitry with critical path. Second port circuitry is omitted for ease of understanding, as described.

5. HSPICE SIMULATIONS

The memory Alias Table with the derived SRAM-CAM cell of 21 transistors is designed and tested of memory array size 32-row x 6-bit for two write ports, four read ports, and two content-addressable ports. Based on the proposed design architecture, the read, and the content-addressable occur during the high phase of the clock, while the write occurs during the low phase of the clock, given all three operations within a single clock cycle. All timing delay values, total power consumption, and total transistor counts are collected based on the cost-effective CMOS transistor level of 65-nm Taiwan Semiconductor Manufacturing Company (TSMC) technology with a 1 V power supply [37] using an HSPICE simulator [38]. Each standard GD estimates at 0.005 ns, but the delay model assumes GD = 0.02 ns as a precaution measure. Although all distributed fan-in and fan-out logic circuits are composed with a four-gate tree structure and minimum geometry (W/L) sizes.

The HSPICE simulations of memory Alias Table realizes two simultaneous write operations, four simultaneous read operations, and two simultaneous content-addressable operations, where all operations occur within a single cycle of the clock. Exhaustive simulations verify the corner cells of the memory array, which are the upper right-hand corner, the upper left-hand corner, the lower right-hand corner, and the lower left-hand corner. Besides, verifying setup and hold time

between signals and clock as well as the pre-charge time. The parasitic model for all signals' wires is estimated based on three layers of metals and 65-nm TSMC technology. Further simulations detail can be found in [34][35]. However, for the brevity of discussion and ease of understanding, the worst-case time delay is shown for each operation as only one simulation. Wherein, the reader can easily follow and verify.

5.1. Read Simulations

The simulation in Figure 7 is conducted for the design read circuitry of the Alias Table, which realizes in the circuit diagram of Figure 3. Assume address "0" assigns to read decoders "A" and "B", while address "31" assigns to read decoders "C" and "B". Since there is only six output bus due to six column and every bus has four read ports, the release output ports from the memory array row "0" are- "OA0, OB0, OA1, OB1, OA2, OB2, OA3, OB3, OA4, OB4, OA5, OB5"; additionally, the release output ports from row "31" are-"OC0, OD0, OC1, OD1, OC2, OD2, OC3, OD3, OC4, OD4, OC5, OD5". Figure 7 presents only the last read ports of corner cell row "0",
which is "OA5, OB5", and row "31", which is "OC5, OD5", instead of showing the complete ports of rows "0" and "31". Subsequently, only the read ports "A and B" of Cell(0,5) and "C and D" of Cell(31,5) are presented for the last column of the memory array as a worst-case timing scenario.

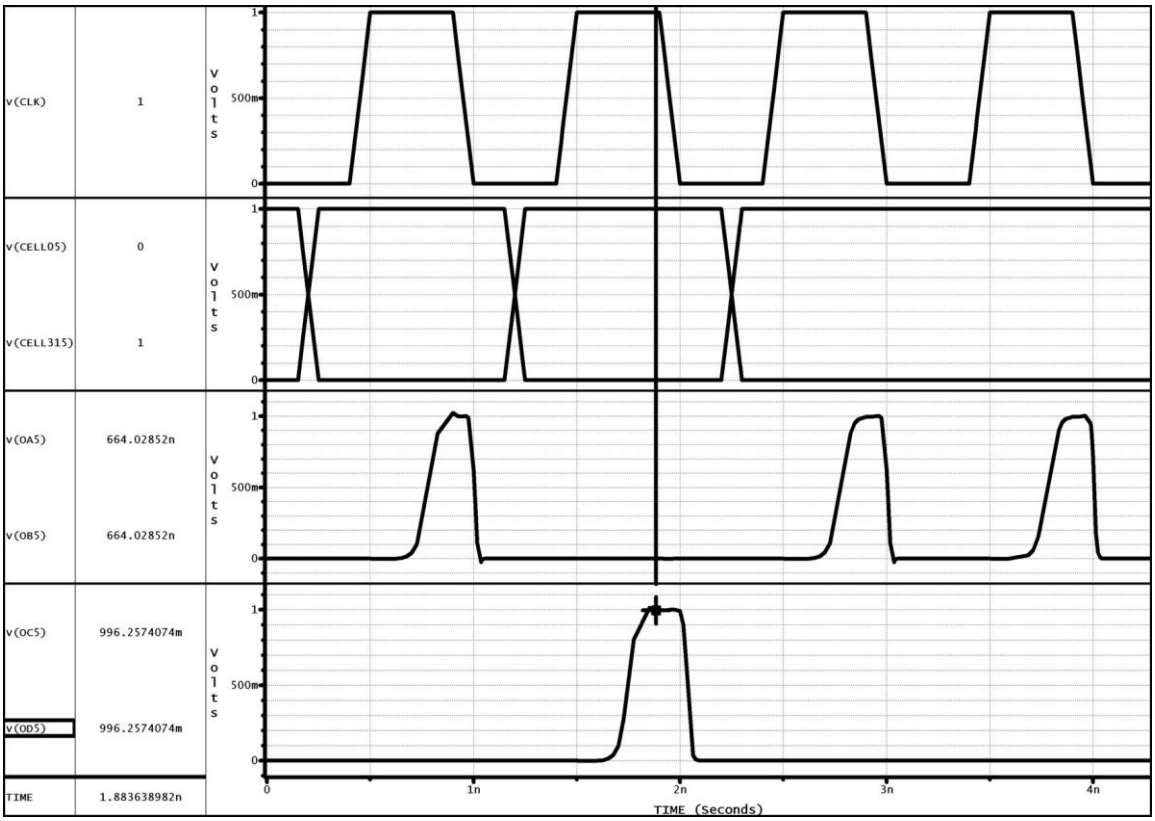

Figure 7. HSPICE simulation for four parallel ports read operation. Two ports fetch from row "0"and two ports fetch from row "31", where only last column is presented, as described.

The first row of Figure 7 gives the system clock running at 1 GHz, where the second row shows the data stored in Cell(0,5) and Cell(31,5), which are opposite to each other to give more simulation clarity to the reader. The third row shows the outputs "OA5, OB5" at about 0.33 ns

from the clock rising edge. These outputs present the fetch data value from Cell(0,5) using the decoders A, B that select row ”0” (read world line "0" is not shown in the simulation). The fourth row shows the outputs “OC5, OD5” at about 0.3 ns, which present the fetch data value from Cell(31,5) since the decoders C, D were selecting row ”31” (read world line "31" is not shown in the simulation). Notice, when the clock is asserted low the output bus shows a “0” value since all read lines hold the pre-charge voltage “VDD”, which were inverted at the read output buffer bus. The read output bus shows the data is fetched out at less than 0.35 ns that is very close to the derived read cycle time in Eq. (1), which is:

Read Access-Time = 0.02 ns/GD * 17 GDs = 0.34 ns.

As a result, the estimated delay model has more conservative results than the simulation model since the delay model scenario assumes more parasitic to further adhere to the safety margin delay. In either case, the read operation can safely operate at the half cycle of 1 GHz since the other half cycle holds the write cycle in parallel with pre-charge time.

5.2. Write Simulations

The write simulation in Figure 8 is conducted for the design write circuitry of the Alias Table, which realizes in Figure 4. Assume address "0" assigns to write decoder "A" and address "31" assigns to write decoder "B". Thus, the write world line "0" of port "A" (WLA0) and the write world line "31" of port "B" (WLB31) are enabled. The data input ports of the memory array are- “DA0, DB0, DA1, DB1, DA2, DB2, DA3, DB3, DA4, DB4, DA5, DB5”, which propagate through all the columns of the memory array to store on rows "0" and "31". Such that, data of port "A" stored on row "0" cells, while data of port "B" stored on row "31" cells.

Figure 8 presents the write simulation of Cell(0,5) and Cell(31,5) of the last column of the memory array as a worst-case timing scenario. The first row of Figure 8 gives the system clock running at 1 GHz. The second row shows the write word lines of rows "0" and "31" (WLA0, WLB31), which are asserted at the falling edge of the clock and de-asserted at the rising edge of the clock consuming a delay of about 0.12 ns. The third row shows the data input ports “DA5, DB5”, which propagates through the last column of the memory array scanning for the active word lines WLA0 and WLB5. The last two signals ("Cell05" and "C315") show the stored data of Cell05 using data port "A" and Cell315 using data port "B". The write simulations show the data is stored at less than 0.18 ns, result in a close write access time to the derived write cycle time in Eq. (2), which is:

Write Access-Time = 0.02 ns/GD * 10 GDs = 0.2 ns.

Consequently, the write operation is active at the low half cycle of 1 GHz, while the read operation is active at the high half cycle of 1 GHz. The pre-charge mode for the read and the content-addressable is active during the low half cycle of 1 GHz.

5.3. Content-Addressable Simulations

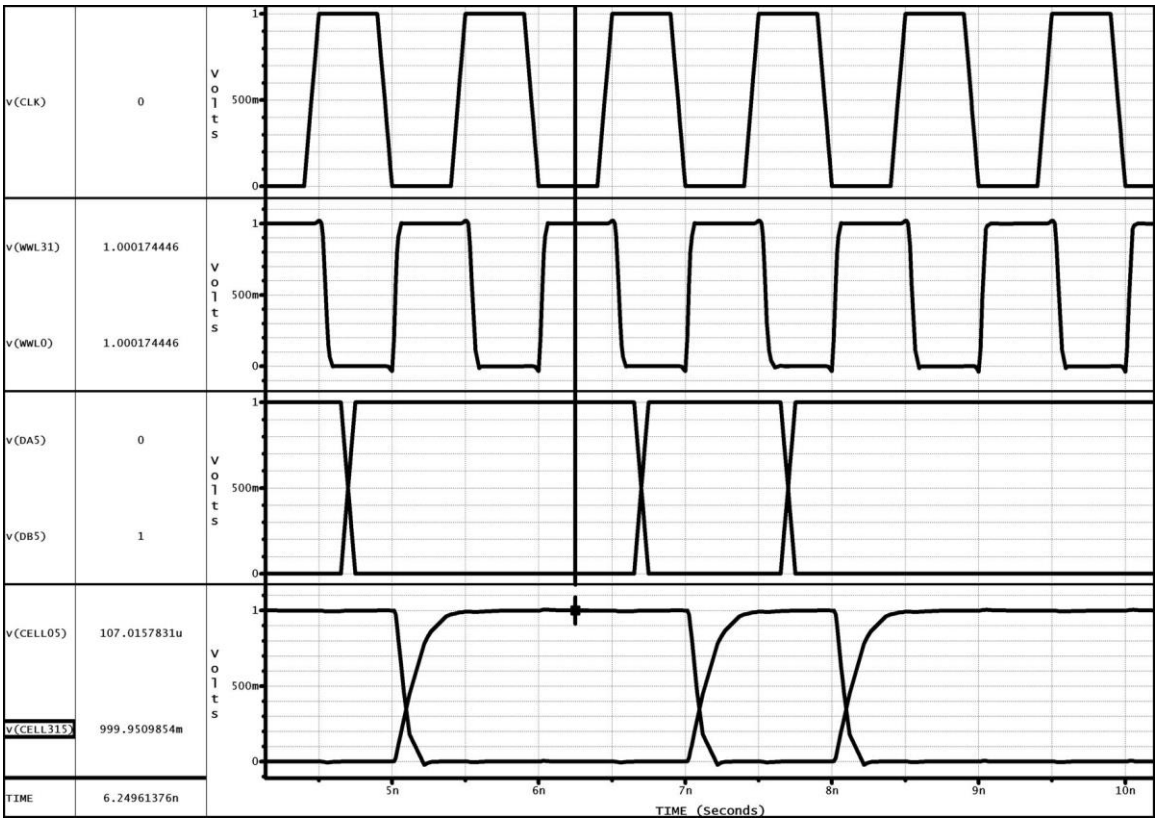

Figure 8. HSPICE simulation for two parallel ports write operation. Two ports fetch from row “0” and two ports fetch from row “31”, where only last column is presented, as described.

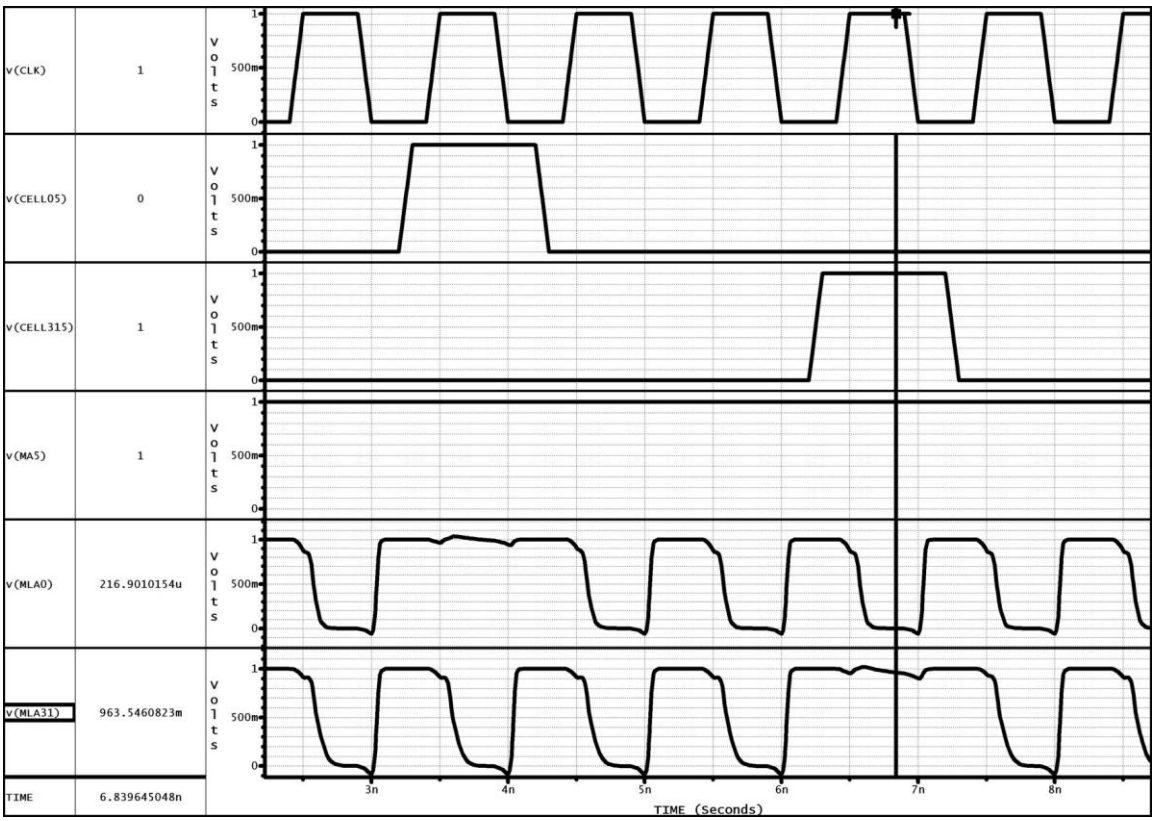

Figure 9: HSPICE simulation for two parallel ports content-addressable operation. Two ports fetch from row “0” and two ports fetch from row “31”, where only the last column is presented, as described.

The simulation of the content-addressable operation in Figure 9 realizes the circuitry of the Alias Table shown in Figure 6, which shows only one port of content-addressable for ease of understanding and brevity of discussion since the other port has a similar operation. The content of the mask register is compared against all rows of the memory array simultaneously, where the row that has the matching contents activates its match line. Subsequently, the active match line permits its associated index address through the Tri-state buffer to propagate to the output port of the content-addressable. The simulation in Figure 9 presents the content-addressable operation, wherein the mask register A is assumed to match the content of row "0" during some time and row "31" during some other time. During the matches, the match line maintains the pre-charge VDD; and thus, enables the Tri-state buffers to release the particular match index value to the output bus of content-addressable. As a result, the output match index bus "CA[0:4]" releases indices "00000" during the match of contents' row "0", and releases indices "11111" during the match of contents' row "31".

The first row of Figure 9 gives the system clock running at 1 GHz, where the second row shows the cells contents Cell(0,5) and Cell(31,5) of only the last cells of rows "0" and row "31" (Column 5) for ease of visuality and brevity of discussion. The third row shows the mask register “A” content of the last bit (MA5). Consequently, the fourth row of simulation shows the match line "0" and "31" of port A (MLA0) and (MLA31), respectively. The MLA0 is high during the match of Cell(0,5) and the high phase of the clock; otherwise, the MLA0 is low during the high phase of clock. Similarly, the MLA31 is high during the match of Cell(31,5) and the high phase of the clock; otherwise, the MLA31 is low during the high phase of clock. Both match lines are high during the low phase of the clock since they are pre-charged during this phase; thus, all match line are high during the low phase of the clock.

The result shows the index is released at less than 0.35 ns, which is very close to the derived content-addressable access time in Eq. (3), that is:

Content-Addressable Access-Time = 0.02 ns/GD * 18 GDs = 0.36 ns.

As a result, the content-addressable operation can safely operate at the half cycle of 1 GHz during the high phase of the clock.

6. RESULTS AND COMPARISONS

The HSPICE simulations conducted in the previous section verifies that the proposed Alias Table exploits the read, write, and content-addressable operations in one cycle without any latency cycle. Such that, the four read ports and two content-addressable ports have occurred in parallel during the high phase of the clock, while the two write ports and pre-charge mechanisms have occurred during the low phase of the clock. Additionally, the simulations showed that the worst operation delay time is less than 20 GDs as a conservative delay measure. Thus, considering a technology factor of 65 nm, the clock cycle of the design can safely run at 1 GHz with a slew rate of 0.1 ns/1V.

Moreover, the Alias Table design has efficient power consumption saving factors. One of the essential factors, the design operates at a 1 V power supply and can further scale for continued CMOS technologies. Additionally, all components have constructed using CMOS transistors with a 65 nm channel length and widths ranging from 3 μm to 5 μm, except for the inverters drivers' widths that are Wp = 10 μm and Wn = 7 μm. Another major contributing factor for low power design is the use of an SRAM-CAM memory array of 21T-Cell structure that is based on the 8T-Cell with a standard geometry size of 65 nm from Intel [25]. The 21T-Cell SRAM-CAM operates at 1V power supply with no sense amplifier obviating a biasing current, which considers a major source of continuous drawn power. Besides, the 21T-Cell provides separate ports for each operation, avoiding a charge contention that minimizes the current density path from power supply to ground, and provides a rail-to-rail noise margin.

TABLE 2 summarizes the comparison of the characteristics between several state-of-the-art Memory Alias Table structures. Key factors such as power consumption, operating speed, performance with latency cycle, operation ports, and cell structure have been discussed and presented. This comparison evaluates the design's factors independent of the underlying technology factor since it is challenging to find comparable designs with the same technical

TABLE 2. Comparison between prior works and the proposed Memory Alias Table design

	Design Structure	**Memory Cell Structure/ ISA**	**Read Ports**	**Write Ports**	**Content-Addressable**	**Characteristics Pros/Cons**	
[7]	Pipeline Design to reduce power and offset low speed	6T-Cell with sensing Amp/ 4-WAY	12-Ports	4-Ports	1-Port Outside memory array	Moderate speed large power Two latencies	Overhead area of pipeline structure to offset low speed
[14]	Design new memory cell, where each cell contains priority	7T-Cell with Pre-charge/ 4-WAY	8-Ports	4-Ports	1-Port Outside memory array	Low Speed Low power One latency	Expensive cell with very large area and slow

	encoders and multiplexers						design
[13]	FIFO memory array equals the number of logical registers.	6T-Cell with FIFO Structure/ 4-WAY	8-Ports	4-Ports	1-Port Outside memory array	High speed Large power Four Latencies	Not efficient in mapping for free list
Our Work	SRAM-CAM based 8T-CELL with separate ports and decoders for each operation	21T-Cell based on 8T-Cell/ 2-WAY	4-Ports	2-Ports	2-Ports Within memory array	High speed Low power No latencies	The design is structured for 2-Way processor and can be expanded for 4-Way processor

Parameters and specifics. However, the comparison still provides insights about relative power consumption, speed, scalability, and design latency throughput. Noticeably, some of the counterpart designs realize content-addressable operation outside the memory array by having a prefix-tree structure of comparators and priority encoders, which worsen the critical path delay and area overhead. In general, designs inherit several latency cycles to compensate for high-frequency operations, and thus, maintain performance from degradation.

The design in [7] is based on the 6T-Cell structure with 4-ports and 12-ports for write and read operations, respectively. Subsequently, the cell has a narrow noise margin between writes and reads, requiring a high-sensitive sense amplifier, which usually draws large biasing currents to leverage the operation speed. Additionally, the design uses an internal pipeline structure in order to remedy the slow clock-cycle for the cost of extra latency cycles. The design further disadvantage harvests large power consumption, and not suitable for continued CMOS scaling technology that requires 1 V or less for the power supply voltage.

The design in [14] operates at the power supply 1 V; however, it is 3X slower than the proposed design. The memory array structure is based on the 7T-Cell that is commonly known for efficient power consumption in the trade of low-speed operation. Each cell contains a comparator and a priority encoder to exploit content-addressable operation; results in a large area cost. The design still requires several latency cycles to offset the low-speed operations; which influence the overall throughput.

The design in [13] exploits a First-in First-out (FIFO) memory array that only reads or writes from the next location. Thus, the design suffers from the low utilization and fragmentation of the memory array. Additionally, the data dependencies of the write and the read operations are accomplished through several priority encoders with multiplexers outside the FIFO array. Furthermore, the content-addressable exploits a prefix-tree structure of comparators outside the FIFO array. The design requires four latency cycles to holds the three operations. Moreover, the FIFO array is based on the 6T-Cell that is known for its low efficiency on power consumption and technology scaling in comparison with the 8T-Cell.

7. CONCLUSION

In this work, the Memory Alias Table circuit design is proposed with size 32-row x 5-bit that exploits two write parallel ports, four read parallel ports, and two content-addressable parallel ports. The high phase of the clock holds the read and the content-addressable operations, while the low phase of the clock holds the write and the pre-charge operations. Therefore, the design operates on a single clock cycle and obviates any latency degradation. The cell of the memory array has an SRAM-CAM structure with 21 transistors. The SRAM portion of the cell carries all advantages of the 8T-Cell SRAM-based and constitutes four parallel read ports and two parallel write ports.

The CAM portion of the cell is based on XOR pass gate logic with a pre-charged match line, which shares with all cells in the same row. Every row has two match ports that receive data for comparison from two mask registers. The mask register broadcasts its content to all rows of the memory array. Concurrently, each row examines its cells' content with the content of the mask register and affect its match line value. If the match line maintains its pre-charge value, then there is a match; and thus, the 5-bit index of that particular row is released. Therefore, content-addressable circuitry precludes a large tree of comparator logic structure. Additionally, the input buffer of the memory array has a priority encoder to alleviate write-after-write data dependencies. In extensive HSPICE simulations, the results show that the clock cycle of 20 standard CMOS gate delays (i.e., independent of technology parameters) can compensate for the three operations without any latency cycles. As a result, the Memory Alias Table operates at clock cycle 1 GHz with a 1 V power supply based on 65 nm technology surpasses most of the current release designs by 3X-to-5X. Furthermore, the proposed design operates with a 1 V power supply and offers continued technology scaling as an attractive feature for low power design.

REFERENCES

[1] K. Patsidis, C. Nicopoulos, G. Ch. Sirakoulis, G. Dimitrakopoulos, “RISC-V2: A scalable RISC-V Vector Processor,” IEEE International Symposium on Circuits and Systems (ISCAS), Sevilla, Spain, Oct. 10-21, 2020.

[2] D. Leibholz and R. Razdan, “The Alpha 21264: A 500 MIPS Out-of-Order Execution Microprocessor,” Proc. Compcon, IEEE Computer Society Press, San Jose, CA. USA, Feb. 23-26, 1997.

[3] K. Moore, Samuel. Breaking the multicore bottleneck. IEEE Spectrum. 53. 16-17. 10.1109/MSPEC.2016.7607015, 2016.

[4] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann; 4th edition, Sept. 27, 2006.

[5] J. P. Shen and M. H. Lipasti, Modern Processor Design: Fundamentals of Superscalar Processors, Waveland Press, Inc., 2013.

[6] M. Postiff;D. Greene;T. Mudge, “The store-load address table and speculative register promotion,” Proceedings 33rd Annual IEEE/ACM International Symposium on Microarchitecture. MICRO-33, Monterey, CA, USA, 2002.

[7] E. Safi;A. Moshovos;A. Veneris, “Two-Stage, Pipelined Register Renaming,”IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 19, Issue 10, pp. 1926-1931, 2011.

[8] D. Sima, “The design space of register renaming techniques,” IEEE Micro, Vol. 20, Issue: 5, pp. 70-83, 2000.

[9] G. Kucuk;O. Ergin;D. Ponomarev;K. Ghose, “Reducing power dissipation of register alias tables in high-performance processors,” IEE Proceedings - Computers and Digital Techniques, Vol. 152, Issue: 6, 2005.

[10] T. N. Buti; R. G. McDonald; Z. Khwaja; A. Ambekar; H. Q. Le; W. E. Burky; B. Williams, “Organization and Implementation of the Register Renaming Mapper for Out-of-Order IBM POWER4 Processors,” IBM Journal of Research and Development, Vol. 49, Issue. 1, pp. 167 – 188, 2005.

[11] S. Petit; R. Ubal; J. Sahuquillo; P. López, "Efficient Register Renaming and Recovery for High-Performance Processors," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, Vol. 22, Issue 7, pp. 1506 – 1514, 2014.
[12] R. Sangireddy, "Fast and low-power processor front-end with reduced rename logic circuit complexity," IEEE International Symposium on Circuits and Systems, Island of Kos, Greece, May 21-24, 2006.
[13] C. Müller-Schloer, W. Karl, and S. Yehia, "Complexity-Effective rename table design for rapid speculation recovery," In International Conference on Architecture of Computing Systems, Springer, Berlin, Heidelberg, pp.15–24, 2010.
[14] De Gloria, Alessandro, and Mauro Olivieri. "An application specific multi-port RAM cell circuit for register renaming units in high speed microprocessors." Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on. Vol. 4. IEEE, 2001.
[15] A. M S Abdelhadi; G. G. F. Lemieux, "Modular Switched Multiported SRAM-Based Memories," ACM Transactions on Reconfigurable Technology and Systems, Vol. 9, Issue 3, pp. 1–26, July 2016.
[16] Yen-Jen Chang; Kun-Lin; TsaiYu-Cheng; ChengMeng-Rong; Lu, "Low-power ternary content-addressable memory design based on a voltage self-controlled fin field-effect transistor segment," Computers & Electrical Engineering, Elsevier, Vol. 81, pp. 528-540, January 2020.
[17] Saleh Abdel-Hafeez, Shadi M. Harb, William R. Eisenstadt, "Low-Power Content Addressable Memory With Read/Write And Matched Mask Ports", PATMOS 2007, LNCS 4644, Gothenburg, Sweden, pp. 75–85, Sep. 2007.
[18] S. Petit;R. Ubal;J. Sahuquillo;P. López, "A power-aware hybrid RAM-CAM renaming mechanism for fast recovery," IEEE International Conference on Computer Design (ICCD), Nov. 2009.
[19] S. S. Ensan, M. H. Moaiyeri,B. Ebrahimi, S. Hessabi, A. Afzali-Kusha, "A low-leakage and high-writable SRAM cell with back-gate biasing in FinFET technology," Springer, Journal of Computational Electronics (2019), Vol.18, pp. 519–526, March 2019.
[20] A. K. Singh; M. M. Seong; C. M. R. Prabhu, "A data aware 9T static random-access memory cell for low power consumption and improved stability," International Journal of Circuit Theory and Applications, Wiley, Vol.8, Issue 4, Jan. 2013.
[21] G. Prasad; N. Kumari; B. Chandra; M. Ali, "Design and statistical analysis of low power and high speed 10T static random-access memory cell," International Journal of Circuit Theory and Applications, Wiley, Vol. 48, Issue 8, May 2020.
[22] W. Hussain; S. M. Jahinuzzaman, "A 7T SRAM bit-cell for low-power embedded memories," Proceedings of the 21st Edition of the great lakes symposium on Great lakes symposium on VLSI (GLSVLSI '11), pp 121–126, May 2011.
[23] Y. Sharma, A. Singh, A. Pandey, "Comparative Design and Analysis and CMOS SRAM Cell," International Conference on Signal Processing and Communication (ICSC), NOIDA, India, March 7-9, 2019.
[24] H. Zhu; V. Kursun, "A comprehensive comparison of superior triple-threshold-voltage 7-transistor, 8-transistor, and 9-transistor SRAM cells," IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne VIC, Australia, June 1-5, 2014.
[25] Saleh Abdel-hafeez and Sarathy P. Sribhashyam, "System and Method for Efficiently Implementing a Double Data Rate Memory Architecture", US patent No. 6356509, March 15, 2002.
[26] A Nand Tech (Intel I7): http://www.anandtech.com/show/2594/10, 2002.
[27] K. Nii, Y. Masuda, M. Yabuuchi, Y. Tsukamoto, S. Ohbayashi, S. Imaoka, M. Igarashi, K. Tomita, N. Tsuboi, H. Makino, K. Ishibashi, H. Shinohara, "A 65 nm Ultra-High-Density Dual-Port SRAM with 0.71um/sup ~/ 8T-Cell for SoC," Symposium on VLSI Circuits, Digest of Technical Papers, Honolulu, HI, USA, 2006.
[28] Y. Chen, M. Fan, P. Hu, P. Su, C. Chuang, "Ultra-low voltage mixed TFET-MOSFET 8T SRAM cell," Proceedings of the 2014 international symposium on Low power electronics and design (ISLPED '14), pp 255–25, August 2014.
[29] M. Patnalaa, A. Yadava, J. Williamsb, A. Gopinatha, B. Nutterb, T. Ytterdalc, M. Rizkalla, "Low power-high speed performance of 8T static RAM cell within GaN TFET, FinFET, and GNRFET technologies – A review," Solid-State Electronics, Elsevier, Vol. 163, January 2020.
[30] Y. Kumar, P. Gupta, "External memory layout vs. schematic," ACM Transactions on Design Automation of Electronic Systems, Vol. 14, Issue 2, pp. 1–20, March 2009.
[31] A. Teman, D. Rossi, P. Andreas, P. Meinerzhagen, L. Benini, A. P. Burg, "Power, Area, and Performance Optimization of Standard Cell Memory Arrays Through Controlled Placement," ACM

Transactions on Design Automation of Electronic Systems, Volume 21, Issue 4, Article No.: 59, pp 1–25, Sept. 2016.
[32] Saleh Abdel-hafeez, Muhanad Quader, sanabel alotoom, "Alias Table Memory Circuit for Register Renaming Unit", IEEE 10th International Conference on Information and Communication Systems (ICICS) , IEEE, Jordan, June 2019.
[33] H. Tabani, Jose-Maria Arnau, J. Tubella, A. Gonzalez, "A Novel Register Renaming Technique for Out-of-Order Processors," IEEE International Symposium on High Performance Computer Architecture (HPCA), Vienna, Austria, Feb. 24-28, 2018.
[34] S. Abdel-Hafeez and A. Matalkah, "CMOS Eight-Transistor Memory Cell for Low-Dynamic-Power High-speed Embedded SRAMS," Journal of Circuits, Systems and Computers, World Scientific, Vol. 17, No. 5, pp. 845-863,Oct. 2008.
[35] S. Abdel-Hafeez, M. Shatnawi, and A. Gordon-Ross, "A Double Data Rate 8t-Cell SRAM Architecture for Systems-On-Chip," IEEE 14Th International Symposium on System-on-Chip, Tampere, Finland, October 11-12, 2012.
[36] H. Jooypa, D. Dideban, "Impact analysis of statistical variability on the accuracy of a propagation delay time compact model in nano-CMOS technology," Springer, Journal of Computational Electronics (2018), Vol. 17, pp. 192-204, Dec. 2017.
[37] Taiwan Semiconductor Manufacturing Corporation, 0.65 μm CMOS ASIC Process Digests, Hsinchu, Taiwan [Online], 2005.
[38] Synopsys, HSPICE, Mountain View, CA [Online], 2016.

AUTHORS

Saleh Abdel-hafeez received his BSEE, MSEE, and Ph.D. in Computer Engineering in the field of VLSI design, the USA. In 1997, he joined S3.inc as a member of their technical staff, where he performed IC circuit design related to cache memory, digital I/O, and ADCs. He has three patents (6,265,509; 6,356,509; 20040211982A1) in the field of IC design. Currently, he is a Professor in the College of Computer and Information Technology, University of Science and Technology, Jordan. His research interests include circuits and architectures for low power and high-performance VLSI. Prof. Abdel-hafeez is a former chairman of the computer engineering department.

Sanabel Otoom received her BSEE and MSEE in Computer Engineering. In 2019, she publishes a paper about Register Alias Table at the 10th International Conference on Information and Communication Systems. Currently, she is seeking an operationality to start her Ph.D. Her research interests include circuits and architectures for high-performance processors and memory.

Quwaider, Muhannad earned his Ph.D. and M.S. at Michigan State University in East Lansing, the USA, and his B.S. at Jordan University of Science and Technology, Jordan. Dr. Quwaider served as vice-dean of the Faculty of Computer and Information Technology from 2018 to 2020 and a chairman of the Computer Engineering department at Jordan University of Science and Technology since 2018. Additionally, He served as a steering committee and TPC chair for the International Conference on Information and Communication Systems (ICICS) from 2012 to 2020. His current research interests include the broad area of wireless data networking, cloud computing, the internet of things, low-power network protocols, high-performance circuit design, and body area network.