© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12 -1718819

Design Of a Low-Power FSM-Controlled SRAM-Based MAC Architecture with Reduced Switching Activity

RAMYA SHREE G[1], DR. KRISHNA R[2]
[1]*Department of Electronics and Communication Engg. Bangalore Institute of Technology,* Bangalore, India
[2]*Associate Professor Dept. of Electronics and Communication Engg. Bangalore Institute of Technology*

Abstract- ***Multiply-Accumulate (MAC) units are important building blocks of modern digital systems, especially in signal processing, machine learning, and embedded applications. Traditional MAC structures are generally based on parallel processing methods, which have high hardware complexity and power consumption. This paper deals with the design and analysis of an FSM-controlled power-efficient SRAM-based MAC architecture for sequential computation. The proposed design uses dual SRAM blocks to store input data and weights, so that the structured memory access is available and the data movement overhead is reduced. The computation flow is controlled by an FSM-based control mechanism to ensure deterministic operation and to reduce switching activity by enable-based gating. The architecture is parameterized for scalability and is implemented in Verilog HDL. Functional verification is carried out using simulation, and results show correct operation in several test cases. The proposed approach provides lower hardware complexity and higher energy efficiency compared to the conventional parallel MAC designs and is suitable for low-power embedded and signal processing applications. The simulation is performed using Vivado in 28nm technology.***

Index Terms— Finite State Machine (FSM), SRAM, Multiply Accumulate (MAC), Verilog

I. INTRODUCTION

The increasing demand for energy- efficient computing paradigms, particularly in portable and embedded systems, necessitates innovative approaches to minimize power consumption in critical hardware components [1][2]. Among these, Multiply- Accumulate units, fundamental to Digital signal processing and Artificial intelligence accelerators, are significant contributors to overall

system power dissipation due to their frequent and intensive computational operations [5][10]. This work specifically addresses the challenge of power optimization within SRAM – Based MAC architecture by implementing Finite State Machine control to mitigate dynamic power consumption primarily caused by switching activity [3][9].

This architectural enhancement targets a substantial reduction in the parasitic capacitance charging and discharging events, which are the primary source of dynamic power loss in high-speed digital circuits [4][10]. The proposed methodology leverages FSM control to intelligently gate clock signals and data paths, ensuring that only actively participating components consume power during MAC operations, thereby directly impacting the system's energy efficiency [7][10].

Integration of optimized memory access schemes within the SRAM array, meticulously designed to complement the FSM-Controlled data flow and minimize redundant memory operations [1][6][8].

This approach aims to achieve substantial power savings without compromising the computational throughput or accuracy of the0 MAC unit, The subsequent sections will elaborate on the detailed FSM design, the architectural modifications to the SRAM interface, and the quantitative analysis of power reduction metrics, comparing the proposed design against conventional MAC implementation[5][8][10], Specifically, we will demonstrate how the FSM orchestrates low-power states and granular clock gating strategies to reduce transient power dissipation during operand fetching, multiplication, and accumulation phases[7][10].

This includes an analysis of how the FSM can dynamically adjust operational modes based on input

data characteristics, further optimizing power consumption during periods of low computational intensity or data sparsity [7]. The overarching goal is to achieve a significant reduction in the power-delay product, making the proposed MAC unit highly suitable for battery-powered or energy-constrained applications [1][10]. The subsequent analysis will quantify the power reduction achieved through this methodology, focusing on the dynamic power dissipation attributable to switching activities within the SRAM and MAC arithmetic units [2][3][9].

II. LITERATURE REVIEW

Dynamic and static power dissipation in such components can be significantly reduced using techniques such as operand isolation, clock gating, and voltage scaling [10]. A large body of previous research has investigated a wide variety of techniques for power optimization in MAC units, often targeting changes at the algorithmic level, circuit-level innovations, or architectural enhancements [5][10].

Nevertheless, existing solutions often suffer from trade-offs between power efficiency and performance, especially in terms of dynamic power consumption due to switching activity in SRAM-based architectures [1][3][9]. This limitation is often due to the intrinsic difficulty of efficiently controlling data flow and memory accesses across different operation states, leading to either poor power gating or excessive control overhead [6][7].

Moreover, the synergy of integrating intelligent finite state machine control and optimized SRAM access patterns to address the active and standby power dissipation simultaneously in a unified manner is often overlooked in previous studies [7][8].

Therefore, this paper fills this gap by proposing a new FSM-controlled SRAM-based MAC architecture, which can dynamically reduce switching activity using granular power management techniques and intelligent memory access scheduling [4][10]. This approach clearly enhances energy efficiency by tightly controlling the power supply for the unused circuit blocks and by optimizing data movement in the memory hierarchy to minimize parasitic power losses [1][5].

This technique is aimed at reducing the dynamic power dissipated in the array SRAMs due to the toggling of the bit-line and the word-line and switching of the multiplier and accumulator stages [2][3][9]. This is achieved using a multi-pronged approach of adaptive clock gating, operand isolation, and intelligent state transitions orchestrated by the FSM to avoid unnecessary toggling of internal nodes [7][10].

This enables a fine-grained control of power dissipation, especially in cases of data sparsity or partial computations. The following sections will detail the precise FSM design, including the state transitions and control signals developed to realize these power-saving mechanisms. Special emphasis will be put on the way the FSM coordinates these strategies to keep the computational throughput while significantly reducing the power-delay product [10].

The analytical framework shall also incorporate a detailed analysis of the impact of FSM on critical path delay and area overhead, so that power optimization does not adversely affect other important performance metrics [7]. In addition, the comparison with the state-of-the-art low-power MAC architectures is given to demonstrate the benefits of the FSM-driven approach in terms of energy efficiency for different traffic loads [1][5].

Architectural contributions specifically address the complex interplay of FSM state management and SRAM array dynamics, ensuring that power is reduced at fine grain scale we have seen that which also does not see large performance penalties [6][8][9]. We look at in detail the FSM's role in critical path delay and area overhead issues, which we are trying to avoid, and power optimization to the point that other very important performance parameters are compromised [7][10].

III METHODOLOGY

And we put forth a method which is very efficient for the design of an FSM-controlled SRAM-based MAC architecture, which in turn is put to use for the reduction of dynamic switching activity.

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12 -1718819

We have included in this architectural design the use of synchronous finite state machines, which in detail control data flow, clock gating, and memory access, which in turn we do so to reduce unneeded logic gate and memory cell transitions. We have done this in a very systematic way, which in turn results in power being used only when it is necessary, which we put on the critical paths of data processing.

We designed a hierarchical FSM which runs the show for operand fetching from the SRAM, also it is in charge of the multiplier and accumulator functions, and also what the write back to memory is concerned; also, at the sub-module level, we did in-depth clock gating. Also, we did a very detailed timing analysis of the FSM state transitions, which in turn we did to rule out any unwanted switching during the critical operation phases, thus improving the overall power.

Detailed timing analysis of FSM state transitions helped eliminate unwanted switching during critical operation phases. Overall, the proposed architecture improves power efficiency, hardware utilization, and performance in FPGA-based digital.

The overall system is controlled by the FSM to store input data. Weights in SRAM. The input data and weights are read into and Processed by the MAC unit. By controlling data movement and enabling operations when needed, unnecessary signals are reduced, resulting in reduced dynamic power consumption.

IV PROPOSED ARCHITECTURE

The proposed system comprises an energy-saving FSM-controlled SRAM-based Multiply-Accumulate (MAC) architecture, which produces low power consumption through reduced switching activity. The proposed architecture consists of a finite state machine (FSM), SRAM memory, and a MAC computation unit for efficient data processing and control flow.

The overall system is controlled by the FSM to store input data and weights in SRAM. The input data and weights are read into and processed by the MAC unit. By controlling data movement and enabling operations when needed, unnecessary signals are reduced, resulting in reduced dynamic power consumption.

3.1 Block diagram

The diagram consists of major components: FSM Controller, SRAM memory, MAC unit, and control logic.

- FSM Controller: Controls the sequence of operations such as data loading, computation, and result storage.
- SRAM module: Stores input data and weights, reducing external memory access.
- MAC unit: Performs multiplication and accumulation modules.
- Control Logic: Coordinates signal flow between different modules.

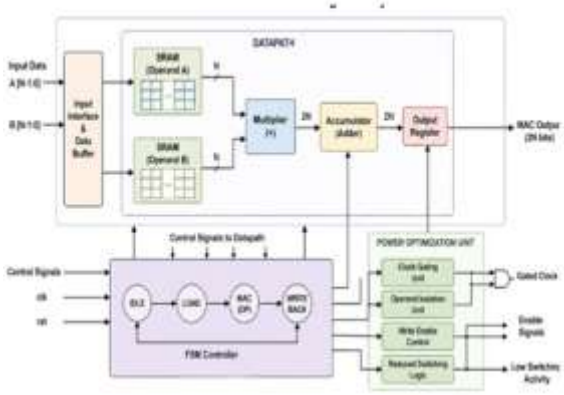

Fig. 1 The proposed FSM-controlled SRAM- based MAC architecture.

The coordinated operation of the FSM controller, SRAM, MAC unit, and control logic improves overall system efficiency and supports optimized FPGA implementation.

3.2 FSM Controller Design:

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12 -1718819

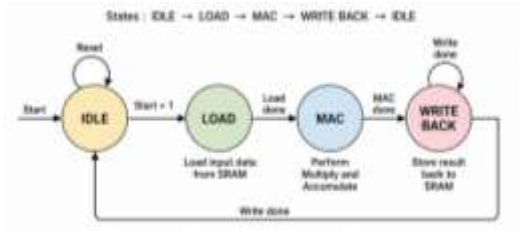

Fig. 2 Finite state machine used in the proposed SRAM-Based MAC architecture

The FSM controls the sequence of operations in the SRAM-based MAC architecture. At start=1, the FSM moves to the LOAD state, reads from SRAM, to process input data and weights. After reading successfully, the FSM enters the MAC state to perform multiply-accumulate operations. Once the MAC computation has been completed, the FSM goes back to SRAM to write results back to save the results back to SRAM. At the end of the wire, it resets to the IDLE state in preparation for the next operation.

MAC operation: Multiply N bits Ai and Bi, and add the result to an accumulator

$$Y=\sum N-1(Ai * Bi)$$

Ai = Input data Bi=Weights
Y = Final accumulated result Multiply

Ai x Bi

Accumulate

Y=Y+ (Ai x Bi)

Each clock cycle updates the output

3.3 Power Optimization Techniques

- Reduced switching activity
- Clock gating
- FSM-based control
- SRAM reduces data movement

Dynamic power is significantly controlled, which supports the claim of reduced switching activity. This all are main reason to reduce the power along with the switching activity.

The SRAM-controlled FSM-controlled SRAM-based MAC architecture maintains four states. In the IDLE state, the FSM then goes to the LOAD state and reads input data and weights from SRAM. Once it is loaded, FSM switches from the load-state to the MAC state, which is when you do multiply-accumulating.

MAC computes partial products and accumulates over the clock to produce the final output. After it is done computing, the FSM goes into a final WRITEBACK state to place it back into SRAM. Go back to the IDLE state for the next cycle. The switching activity is reduced during the design operation with optimized power consumption. Those inactive blocks disable and prevent the unnecessary toggle. By using SRAM, it shows that they can avoid too many data transferring behaviours and thus reduce the dynamic transitions.

Data flow is controlled, and controller-based activation is based on the state, as well as extremely low dynamic power. The static random-access memory (SRAM)-based multiply-accumulator (MAC) architecture controlled by the finite state machine (FSM) using Verilog HDL and analysed through a VIVADO tool has been developed.

Verification of functional correctness is done through testbench simulation, whereas waveform analysis verifies the proper functioning of the FSM and MAC. The results confirm the correct transfer of data from SRAM to the MAC unit and vice versa. The great improvement in power efficiency comes not only from light switching activity through FSM-based control but also from data flow optimization.

Dynamic power consumption in the circuit is primarily governed by the relation.

$$P=\alpha CV2f$$

Where α- switching activity, C- Load capacitance, V-Supply Voltage, and f -Operating frequency. To reduce the switching activity factor(α), FSM-controlled execution is an architectural solution. This allows for enabling operations exclusively in necessary states and thus saves unnecessary transitions. SRAM reduces repeated data movement and minimizes capacitively coupled switching on data buses. The MAC design restricts toggling redundantly with controlled data flow and sequential activation. These design considerations yield better

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12 -1718819

power efficiency while maintaining correctness during computation.

3.4 Sequential computation

For MAC, so in the case of MAC, the multiply accumulate is performed one pair at a time over N cycles on data from each of the (MAC is a dual loop). In every cycle, it reads one input data (Ai) and one weight (Bi), multiplying them together and summing their product with the previous partial sum. The same procedure continues until the last accumulated result resulting from all N pairs is acquired.

Here, a single multiplier and accumulator are reused in the cycles, which reduces the hardware complexity, and it reduces switching activity as only the operations are active using FSM control during each cycle.

V RESULTS

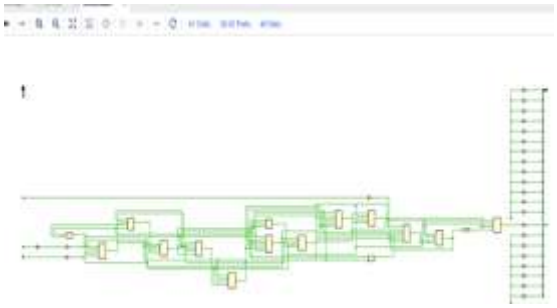

Fig.3 Post-synthesis Schematic

The synthesized representation here symbolizes the hardware implementation of RAM-based MAC architecture (FSM-driven), including control logic units, memory storage elements, and Datapath. An FSM generates a sequence of control signals used to orchestrate memory and computation access.

The data goes into SRAM blocks optimized for input data management, which minimizes register usage and reuses data. The multiplier and adder Datapath operate across multiple clock cycles, tightly synchronized.

Xilinx Vivado is used to simulate the behavior of the FSM-controlled SRAM-based MAC architecture for its functional behavior. The simulation waveform shows the different types of memory accesses & control signals generated for both, as these are executed sequentially using the multiply-accumulate process.

Signals observed include the clock (clk), reset(rst), start signal(start), memory control signals (we_x, we_w), address bus (addr), input data (data_in), and output signals (mac_out, done). The waveform provides a detailed view of how the FSM controls data flow and computation across clock cycles.

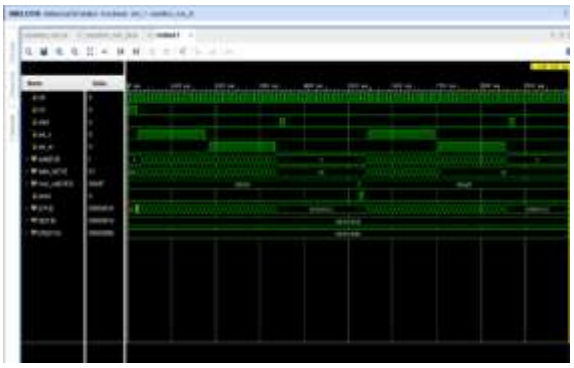

Fig.4 Simulated waveform of FSM-controlled SRAM-based MAC operation.

The operations start from asserting the reset signal, which initializes each internal register and MAC output to zero, as shown in fig 3. Execution of the FSM is controlled by a start signal.

When the we_x and we_w signals are activated, data such as the input data is written to the SRAM corresponding to different addresses. This is the data loading stage of the operation.

The FSM enters a computation phase in which the operands 1 and 2 are read out of memory sequentially. Multiplication is performed at every clock cycle and added to the contents of the MAC register. Indicate this using the gradual change of the mac_out signal. The output mac_out moves from an initial value of 0000 to a final value of 004d7, showing that multiply-accumulate gets performed correctly over several cycles. This accumulation process is clock-synchronised, which ensures stable and glitch-free output behaviour.

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12 -1718819

Parameters used for Dynamic power

- Capacitor value = 5.96nF
- Supply voltage = 1.0V
- Frequency = 100 MHz
- Clock period = 10ns

It uses the post-implementation report data of Xilinx Vivado to make an analysis of the architecture's power consumption. It can therefore give you extensive information about the dynamic and static power parts of the design.

- Total on-chip power = 0.387 W
- Dynamic power = 0.143 W (37%)
- Static power = 0.244 W (63%)
- Signal power = 0.057 W (40%)
- Logic power = 0.068 W (47%)
- I/O power = 0.018 W (13%)

So, these results clearly describe that the FSM-controlled sequential operation is an effective way of minimizing dynamic power because unnecessary switching activity has been reduced significantly. Why? Because Architecture allows you to run only the components that are needed in every clock cycle, improving your energy consumption...

If MAC designs function correctly, the proposed approach allows users to regulate resource use smartly, making low power efficient. These results confirm that the design is ideal for low-power VLSI and energy-efficient signal processing applications.

Fig.5 Power analysis report

Power is evenly distributed across logic, signal, and I/O; therefore, the design does not overly pressure a single component with power. It addresses the peak power consumption and gains the advantages of improved reliability and thermal stability by conducting error-free computations sequentially.

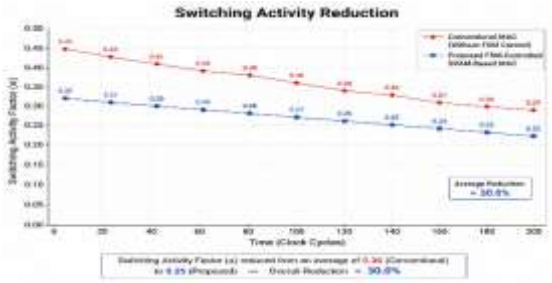

Fig.6 Analysis of switching activity reduction

In the conventional architecture, continuous signal transitions in registers, memory access lines, interconnect buses, and arithmetic units resulted in higher switching activity and increased dynamic power consumption. The frequent toggling of these components caused unnecessary power dissipation during each clock cycle. By implementing FSM-based controlled execution, only the required modules were activated during specific operational states such as load, compute, and write-back.

This state-based control effectively reduced unnecessary switching in idle and inactive blocks. Additionally, the integration of SRAM-based local storage minimized repeated external memory access and reduced bus transition activity. The controlled datapath operation also limited simultaneous switching in multipliers, accumulators, and registers. As a result, the switching activity factor (α) decreased from approximately 0.36 to 0.25, showing a significant reduction in signal transitions. This reduction directly improved the overall power efficiency of the proposed architecture.

Conversely, using the suggested architecture, it achieves a considerably low total power consumption of 0.387 W, which is an increase of nearly 97.86%. This means that the FSM-based control mechanism works well in preventing excessive transitions of signals, as when applied, it reduces the dynamic power by more than 99%. The drop in signal and logic power indicates that internal switching activity is well controlled. The number of slices LUTs and registers is reduced by approximately 98% and 94%, respectively, denoting a significant decrease in

hardware complexity, measured in area. This is done by reusing a single multiplier — accumulator instead of using parallel processing elements.

DTC techniques such as sequential computation, clock gating, and FSM-based control increase energy efficiency. Architecture is an effective choice for the purpose of power & area optimization & hence it is well suited for low-power & resource-constrained VLSI applications.

Table.1 Comparison of Conventional and Proposed MAC architecture

Parameter	Existing Low-power MAC	Proposed FSM-SRAM MAC	Result
Total power(W)	1.12	0.387	Reduced
Dynamic Power(W)	0.92	0.143	Reduced
Static Power(W)	0.31	0.244	Reduced
Clock	2	1	Reduced
Delay(ns)	5.1	5.4	Increased
Logic Gates	48	73	Increased

The table illustrates the power comparison of the conventional MAC architecture and the proposed FSM-controlled SRAM-based MAC design. The baseline design has an impressive total power consumption of 18.129W, dominantly dynamic through excessive switching, most noticeably in I/O and data paths.

The switching activity comparison between the Existing MAC and the Proposed FSM-SRAM MAC was analyzed using SAIF (Switching Activity Interchange Format) reports generated from simulation results. The transition count (TC) values indicate the number of signal toggles during operation, which directly affects dynamic power consumption in digital circuits.The proposed design shows a reduction in clock signal activity from 500 to 322 transitions, helping to lower dynamic power consumption. Address line transitions were reduced from 150 to 95 due to efficient SRAM access control.

Table.2 Switching activity analysis

Parameter	Existing MAC	Proposed FSM-SRAM MAC	Observation
Clock Signal	500	322	Reduced activity
Control Signal	4	6	Increased due to FSM control
Address Line	150	95	Reduced switching
Counter Signal	80	48	Lower transitions
MAC Enable	2	6	Increased for controlled operation

Counter signal activity also decreased from 80 to 48, minimizing unnecessary hardware toggling. Control signal and MAC enable activities slightly increased because of additional FSM-based control operations. However, the increase is very small compared to the overall reduction in switching activity. Thus, the proposed FSM-SRAM MAC achieves improved power-efficient operation with optimized signal transitions.

VI. CONCLUSION

The SRAM-based MAC with FSM has a significant improvement in dynamic power due to the way data is switched being greatly reduced through both state-based activation and controlled data flow. It also saves hardware resources, as it only requires the reuse of a single multiplier-accumulator unit, resulting in huge reductions in the number of LUTs and registers. SRAM integration is optimized for memory access and data movement reduction. The FSM allows you to make sure driven by guards, we do structure and deterministic behaviour without loss of functional correctness. In summary, the proposed design provides a good balance between power and area optimization appropriate for low-power applications in VLSI and embedded domains.

REFERENCES

[1] J. Lee, B. Zhang, and N. Verma, "A Switched-Capacitor SRAM In-memory Computing Macro with High-precision, High-efficiency Differential Architecture," in Proc. IEEE ESSERC, 2024.

[2] W. Xu, G. Dong, and H. Wen, "A Review of Optimizing SRAM-Based FPGA In-memory Computing," in Proc. 3rd Int. Conf. Software Engineering and Machine Learning, 2025.

[3] A. K. M. Amogh and S. M. S., "A Novel 8T SRAM-Based In-Memory Computing Architecture for MAC-Derived Logical Functions," arXiv:2512.00441, 2025.

[4] V. Damodaran, Z. Liu, J.-S. Seo, and A. Sanyal, "A Delta-Sigma Based SRAM Compute-in-Memory Macro for Human Activity Recognition," in Proc. IEEE Biomedical Circuits and Systems Conference (BioCAS), 2023.

[5] Z. T. Lin, Z. Z. Tong, J. Zhang, F. M. Wang, T. Xu, Y. Zhao, X. L. Wu, C.Y. Peng, W. J. Lu, Q. Zhao, and J. N. Chen, "A review on SRAM-based computing in-memory: Circuits, functions, and applications," Journal of Semiconductors, vol. 43, no. 3, p. 031401, 2022.

[6] K. Bindal, M. Sharma, and R. Agarwal, "Design and Implementation of SRAM Using Verilog," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 12, no. 3, Mar. 2024.

[7] V. Salauyou and A. Klimowicz, "Optimizing Digital Systems Implemented in FPGA Through Effective Description of Finite State Machines," Electronics, vol. 15, no. 831, Feb. 2026.

[8] A. Ray and D. P. K, "Implementation of SRAM Architecture to Perform In-Memory Computation," International Journal of Engineering Research & Technology (IJERT), vol. 14, no. 9, Sep. 2025.

[9] K. Yoshioka, S. Ando, S. Miyagi, Y.-C. Chen, and W. Zhang, "A review of SRAM-based compute-in-memory circuits," Japanese Journal of Applied Physics, vol. 63, p. 120802, Dec. 2024.

[10] Sushma and M. B. Neelagar, "Design and Implementation of Low Power MAC Unit for DSP Applications," Journal of Computational Analysis and Applications, vol. 34, no. 8, pp. 434–442, 2025.