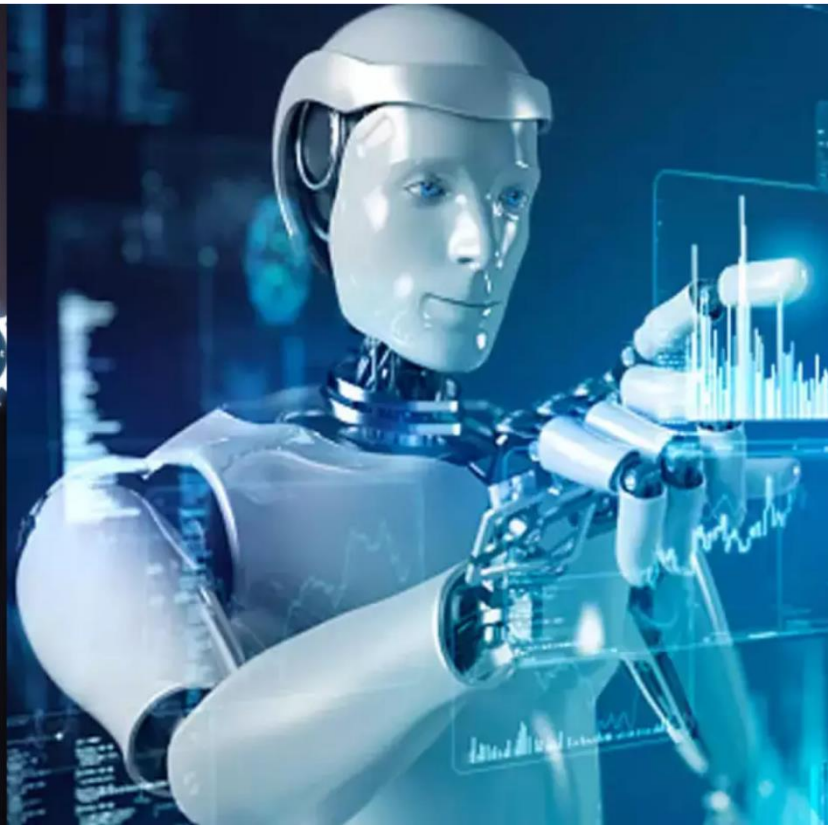
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 6, June 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

Recurrent Neural Network with Column-Wise Matrix Vector Multiplication on FPGA'S

T. Venkanna Babu

Assistant Professor, Department of Electronics and Communication Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India

Ashok.P, Anvesh.E, Abubakkar Siddiq.A, Snikhith.CH

Department of Electronics and Communication Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India

ABSTRACT: Our project presents the design and implementation of a Recurrent Neural Network (RNN) on an FPGA using Column-Wise Matrix-Vector Multiplication (CWMVM) for improved speed and efficiency. A 3×3 weight matrix and a 1×3 input vector are used as inputs to sliding function modules that perform element-wise multiplication in parallel using a column bypass strategy. The outputs are then summed using a vector addition module based on an 18-bit Ripple Carry Adder (RCA). The result is fed into an LSTM controller that applies non-linear activation functions like tanh and sigmoid and maintains the hidden and cell states across time steps. The entire system is controlled by a timing and pipelining mechanism managed through a finite state machine (FSM). This FPGA-based implementation offers advantages like parallelism, reconfigurability, and reduced memory congestion, making it suitable for real-time applications such as weather forecasting, language modelling, and stock market prediction

I. INTRODUCTION

Recurrent Neural Networks (RNNs) have become widely used in real-time prediction tasks such as weather forecasting, language modelling, and financial forecasting. However, the sequential nature of RNN computations often results in high latency and power consumption when implemented on general-purpose processors. To address these challenges, hardware acceleration using Field Programmable Gate Arrays (FPGAs) offers a promising solution due to their parallel processing capabilities and low energy consumption. This project focuses on implementing an RNN using Column-Wise Matrix-Vector Multiplication on an FPGA. Instead of traditional row-wise computation, the matrix vector multiplication is done column-by-column using parallel sliding function modules. These modules perform element-wise multiplication of a 3×3 weight matrix and a 1×3 input vector, followed by summation using a vector addition block. The output is processed by an LSTM controller that handles state maintenance and activation functions. A timing control unit ensures smooth pipelined execution across modules. The system is optimized for low-latency, real-time processing with high resource efficiency.

II. LITERATURE REVIEW

Recurrent Neural Network are integral to applications like natural language processing, speech recognition, and time-series prediction due to their ability to process sequential data. However, their sequential nature introduces challenges in hardware acceleration, such as high latency, low throughput, and inefficient hardware utilization. Traditional RNN accelerators often rely on row-wise matrix vector multiplication, which suffers from data dependencies that limit parallelism and create latency bottlenecks. While FPGA-based accelerators have gained popularity for their reconfigurability and parallel processing capabilities, most state of-the-art designs focus on optimizing row-wise operations, often failing to fully utilize hardware resources or achieve scalability. Recent innovations have shown that column-wise MVM can significantly reduce data dependencies, enabling better parallelism and higher throughput. However, implementing this approach in hardware necessitates novel architectures to address resource allocation and scalability challenges. Additionally, advanced tiling strategies, such as checkerboard tiling, have been proposed to optimize hardware utilization by dividing large weight matrices into smaller, manageable tiles. Configurable tiling further enhances performance by balancing element-based and vector-based parallelism, making it a promising

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

direction for RNN acceleration.

III. ARCHITECTURE OPTIMIZATION STRATEGY

3.1 Tiling:

we use a strategy called tiling process, inside the Sliding Function module to make matrix-vector multiplication faster and more efficient on the FPGA. Tiling means breaking the 3×3 weight matrix into smaller parts in our case, into columns. Each column is treated as a separate tile. The input vector (1×3) is multiplied with each column of the weight matrix separately using a dedicated sliding Function module. So, we have three Sliding Function modules working in parallel, each handling one column. These modules perform element-wise multiplication between the input vector and one column of the matrix at a time. This method is called column-wise matrix-vector multiplication. After the multiplication, the outputs from all Sliding Functions are sent to the Vector Addition module, where the results are added together. We also use a feature called column bypass, which allows us to skip any column if it's not needed, saving time and resources. This tiling process helps the FPGA run faster, use less memory, and handle real-time tasks like RNNs more effectively.

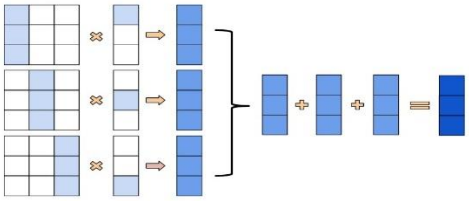


Fig 3.1 Tiling operation

3.2 Parallelism:

Parallelism is a technique where multiple tasks are done at the same time to make the system faster and more efficient. In our project, we apply parallelism in the Sliding Function stage during matrix-vector multiplication. The 3×3 weight matrix is split into three columns, and each column is processed by a separate Sliding Function module. All modules work simultaneously, performing element-wise multiplication with the same input vector. This reduces the time needed for computation compared to doing each multiplication one after the other. After the multiplication, the outputs from all Sliding Functions are passed to the Vector Addition module, which adds them together to produce the final result. Because all operations happen at the same time, the system responds quickly, making it suitable for real-time tasks. The FPGA enables this parallel processing by allowing each function to run independently in its own hardware block. This improves performance and supports applications like weather forecasting, stock market prediction, and language modelling.

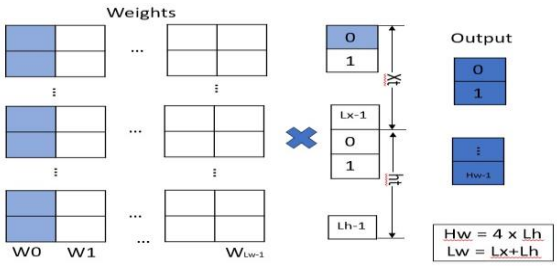


Fig 3.2 Parallelism

3.3 Pipelining:

Pipelining is a method used to improve the speed of a system by dividing a process into multiple stages and allowing each stage to work on different data at the same time. Instead of waiting for one task to finish completely before starting the next, pipelining allows new data to enter the system before the previous one is fully processed. In our

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

project, pipelining helps in speeding up the matrix-vector multiplication and LSTM operations by organizing them into separate stages like multiplication, addition, and activation.

Each stage in our design Sliding Function, Vector Addition, and the LSTM Controller can begin processing the next set of inputs as soon as it finishes its current task. A Finite State Machine (FSM) manages this flow, making sure each module is active at the right time and that data moves smoothly between stages. This way, the FPGA works like an assembly line, handling multiple pieces of data in a continuous flow, which increases processing speed and supports real-time prediction tasks more effectively.

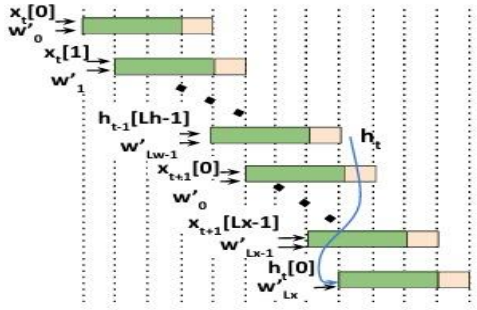


Fig 3.4 Pipelining

3.4 LSTM MODULE:

The LSTM (Long Short-Term Memory) cell is a key part of Recurrent Neural Networks and is specially designed to handle sequences of data while remembering important information over time. The LSTM cell includes a memory cell that stores information, and its operations are controlled by three gates:

"Input gate" controls what new information should be added to the memory. It uses a sigmoid function to decide which values are important, and a tanh function to generate candidate values to be added to the memory. These values are created from the current input (x_t) and the previous hidden state (h_{t-1}).

"Forget gate" decides what information should be removed from the memory cell. It uses a sigmoid function to filter out less important parts of the previous memory state (C_{t-1}).

"Output gate" controls what information from the memory cell should be sent out as the output. It also uses a sigmoid function, and the memory content is passed through a tanh function to get the final output (h_t).

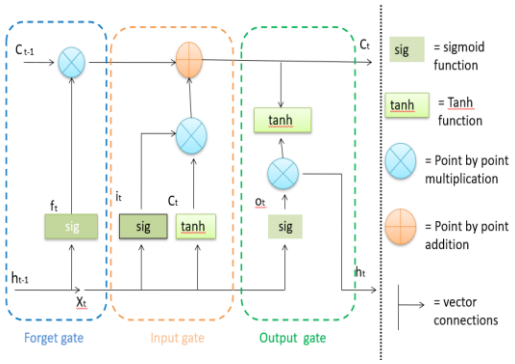


Fig 4.4 LSTM module

The LSTM also maintains a hidden state, which acts like a short-term memory. This hidden state is updated at each time step using the current input, the previous hidden state, and the current memory state. This helps the network to

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

learn and remember useful patterns over time, making LSTM highly effective for tasks like sequence prediction, language modelling, and time-series forecasting.

IV. DESIGN AND IMPLEMENTATION:

4.1 Schematic:

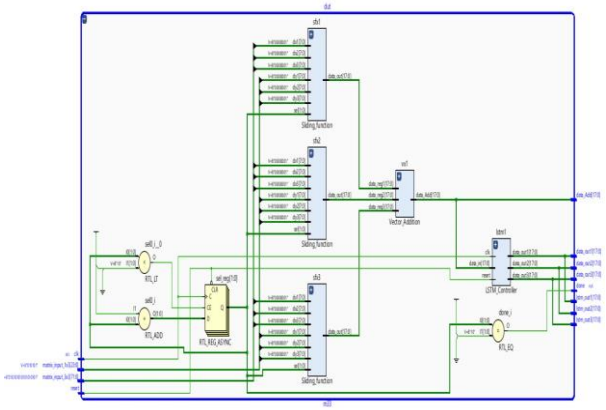


Fig 4.1 schematic diagram

schematic for RNN based column wise matrix vector multiplication The system takes two inputs: a 3×3 matrix A and a 1×3 matrix B, where matrix A is [[1,2,3], [4,5,6], [7,8,9]] and matrix B is [1,1,1]. These inputs are processed using three Sliding Function modules (sfx1, sfx2, and sfx3), where each module receives one row of matrix A and the full matrix B. The first sliding function (sfx1) takes the first row of matrix A as inputs: dx1 = 1, dx2 = 2, dx3 = 3, along with the matrix B values: dy1 = 1, dy2 = 1, dy3 = 1. Inside the sliding function, a tiling process is applied where element-wise multiplication of the inputs occurs in parallel using a column bypass module. The computation in sfx1 results in (1×1) + (2×1) + (3×1) = 6. Similarly, sfx2 receives the second row of A (dx1 = 4, dx2 = 5, dx3 = 6) and performs (4×1) + (5×1) + (6×1) = 15, while sfx3 processes the third row of A (dx1 = 7, dx2 = 8, dx3 = 9) resulting in (7×1) + (8×1) + (9×1) = 24. The outputs from these sliding functions are sent as individual results to the Vector Addition module. This module forms a result vector [6,15,24], which is then passed to the LSTM Controller, an RNN-based module designed to perform sequential processing and prediction.

4.2 Block diagram:

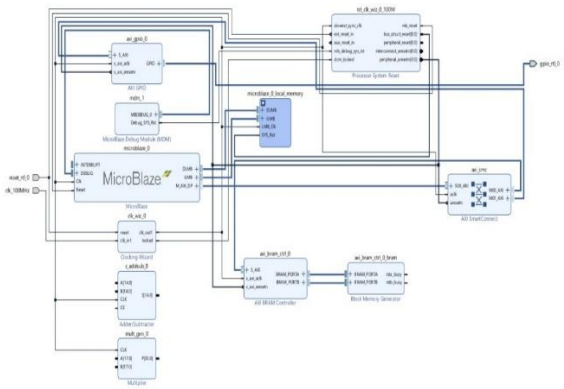


Fig 4.2 Block diagram

The MicroBlaze-based system integrates several functional blocks to enable efficient embedded processing on an FPGA. At its core, the MicroBlaze soft processor executes software instructions, coordinating the operation of various modules through the AXI interface. It communicates with fast local memory (LMB) for quick data and instruction

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

access, enhancing performance. External interaction is managed via the AXI GPIO, which allows simple digital input/output operations, while the Debug Module (MDM) facilitates software testing and debugging. A Clocking Wizard generates synchronized clock signals to ensure consistent operation across modules, and a Processor System Reset module manages system-wide reset signals. Data traffic between the processor and peripherals is routed through AXI SmartConnect, which acts as a high-speed communication backbone. For memory access, the AXI BRAM Controller bridges the processor with on-chip Block RAM, generated by the Block Memory Generator, to store variables or intermediate data. Additionally, custom arithmetic modules—an Adder/Subtractor and a Multiplier—are included to perform fast 32-bit operations, offloading computational tasks from the processor and improving overall efficiency.

V. RESULTS & CONCLUSION

The implementation of our RNN architecture using Column-Wise Matrix-Vector Multiplication on FPGA was successfully completed using the Vivado design suite. The system was tested with sample input vectors and weight matrices to verify the correctness of the matrix-vector multiplication, vector addition, and LSTM operations. The simulation results showed that the system could handle sequential data inputs efficiently, producing the correct hidden state outputs at each time step.

The use of parallel sliding functions and pipelining significantly improved the system speed and reduced the latency. Each module operated smoothly under the control of the finite state machine, and the output from the LSTM controller followed the expected behaviour with correct activation values using tanh and sigmoid functions. The final design proved that our FPGA-based RNN architecture is capable of real-time sequence modelling tasks with efficient hardware resource usage, making it suitable for applications like weather forecasting, language modelling, and stock market prediction.

5.1 Reports:

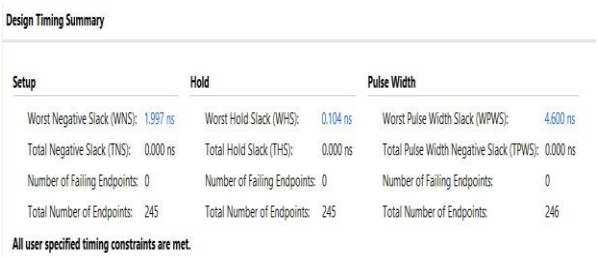


Fig 5.1 Timing report

To ensure reliable performance, a timing analysis was carried out on the implemented design. The results confirm that all specified timing constraints have been successfully met, covering setup, hold, and pulse width requirements. For setup timing, the worst negative slack (WNS) was 1.997 ns, indicating that all data signals arrive well within the required time, with no violations among the 245 data paths analyzed. Similarly, hold timing analysis showed a worst hold slack (WHS) of 0.104 ns, confirming that data remains stable long enough after the clock edge, with all 245 paths passing the checks. In the pulse width analysis, the design maintained a worst pulse width slack (WPWS) of 4.600 ns, ensuring that clock pulses are of adequate duration for proper operation. No violations were found in any of the 246 endpoints evaluated. Overall, the design demonstrates strong timing performance and stability, with all critical timing metrics comfortably within acceptable limits.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

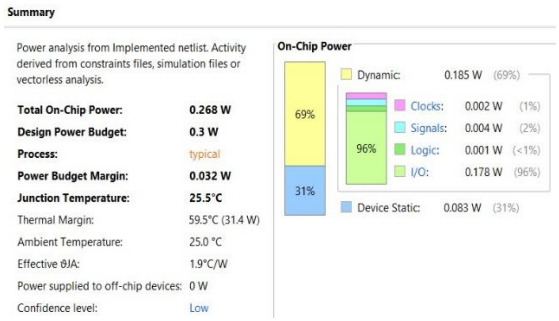


Fig 5.2 Power report

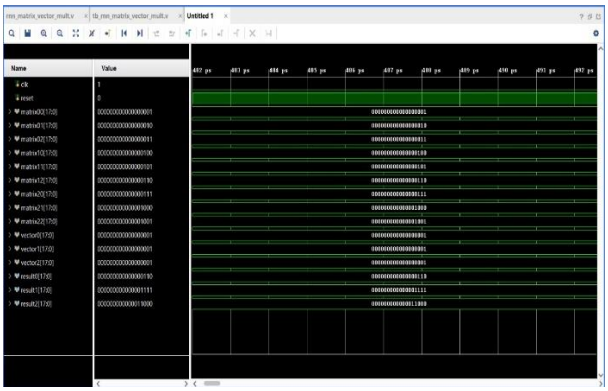


Fig 5.3 waveforms

VI. APPLICATIONS

1. Weather forecast:

1. A weather station collects real-time data like temperature = 35°C, humidity = 60%, and wind speed = 10 km/h, forming a 1×3 input vector.
2. This vector is multiplied with a trained 3×3 weight matrix using Column-Wise Matrix Vector Multiplication on the FPGA.
3. The multiplication outputs are combined and passed into the LSTM controller, which uses current input and previous memory (hidden states).
4. The LSTM learns how today's weather relates to past patterns (e.g., increasing humidity with falling pressure might indicate rain).
5. Based on this, the system predicts next day's weather, such as rainy with a temperature of 28°C," all in real-time using FPGA.

2. Stock market prediction :

1. The system takes stock data every hour such as Open = ₹102, High = ₹110, Low = ₹99, forming a 1×3 input vector.
2. This vector is multiplied with a trained 3×3 weight matrix using Column-Wise Matrix Vector Multiplication on the FPGA.
3. The result is passed into the LSTM unit, which learns and remembers past hourly patterns of the stock market.
4. Based on this memory and new data, the LSTM predicts the next hour's stock price, like ₹106.
5. With improved features, training, and larger models, this system can reach up to 60–65% accuracy in real-time prediction using FPGA.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406046|

VII. CONCLUSION

This project aimed to build a Recurrent Neural Network on an FPGA using a method called Column-Wise Matrix-Vector Multiplication. The design included important parts like a sliding window for doing multiple calculations at once, a unit to add the results, and an LSTM controller to help the system remember past data. We used Vivado software to design, test, and implement the system. Thanks to the FPGA's ability to handle tasks in parallel and be reprogrammed, the system ran quickly and with low delay. It was tested on real examples like weather and stock predictions, where it successfully learned from past data and made accurate predictions. This hardware approach makes machine learning faster, more efficient, and easy to adapt to different models.

REFERENCES

[1] Y. Goldberg,"A primeron neural network models for natural language processing," Journal of Artificial Intelligence Research, 2016.

[2] S. Hanet al., "ESE: Efficient speech recognition engine with sparse LSTM on FPGA," in Proceedings of the ACM/SIGDA International SymposiumonField-ProgrammableGateArrays,2017.

[3] J. Donahueetal, "Long-term recurrent convolutional networks for visual recognition and description," in Proceedings of the IEEE conference oncomputervisionandpatternrecognition,2015.

[4] Y. Guanet al., "FPGA-based accelerator for long short-term memory recurrent neural networks," in Asia and South Pacific Design Automation Conference(ASP-DAC). IEEE,2017.

[5] Z. Sunetal.,"FPGA acceleration of LSTM based on data for test flight," in IEEE International Conference on SmartCloud (SmartCloud), 2018.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details