Muhammad Farhan Azmine

540-557-8751 muhammadfarhan@vt.edu [Website] [Github] [linkedin.com/Muhammad Farhan Azmine]

Education

Virginia Tech (GPA: 4.00 / 4.00) Blacksburg & Alexandria, VA
PhD in Computer Engineering (Direct PhD; MS completed) *Aug 2022 – Dec 2026 (Expected)*

Bangladesh University of Engineering and Technology (BUET) *Dhaka, Bangladesh*
Bachelors of Science in Electrical & Electronics Engineering *Jan 2013 – Sep 2017*

- **Relevant Coursework:** Advanced Digital Design, Advanced Computer Architecture, Testing VLSI Techniques, VLSI Device Modeling, Advanced Analog IC Design, Deep Learning, Advanced Machine Learning, Computation in Data Science

Publications

- **Muhammad F. A.**, Li, R., Sharma, G., & Yi, Y. (2025). SpikeSpec: On-Chip Learning Neuromorphic Accelerator for Spectrum Sensing. *IEEE Transaction CAD*, Feb 2025.
- Li, R., **Muhammad F. A.**, Sharma, G., & Yi, Y. (2025). Efficient Digital Architecture of Spiking Encoders. In *Proc. ISQED 2025*.
- Lin, C., **Muhammad F. A.**, Liang, Y., & Yi, Y. (2024). Neuro-Inspired AI Accelerator for 6G Networks. *Front. Comput. Neurosci.*
- Lin, C., **Muhammad F. A.**, & Yi, Y. (2023). Accelerating Wireless Communications with FPGA-Based AI. In *ICCAD 2023*.

Work Experience

Research Assistant *(Functioning as RTL Engineer – Complex Digital System Design & Verification)* *June 2023 – present*

An ultra efficient DSP based RTL chip implementation of hidden layer alternative for Recurrent Neural Network

Summer 24-Spring 25

- Achieved **100% design validation** through hardware–software co-integration on Zynq SoC, enabling real-time sample delivery to FPGA (PL) and result capture via ZynQ AXI-UART (PS), with a custom C++ parser for automated CSV input processing and output generation.
- **Decreased dynamic power consumption by 65%** by using adder tree DSP48 structure to replace CLB based matrix adder-multiplier which resulted in 86% LUT and 64% FF reduction
- **Cut wire delay by 3.2ns** through **fanout optimization** to boost operating frequency.
- **Maximized throughput by 240 MHz** by replacing LUT logic with DSP48E2 IP cores and adding pipeline registers to the critical path.

RL based Spiking Neural Network RTL architecture design with On-Chip learning for Spectrum-Sensing

[Github link] *Fall 22 - Spring 24*

- **Achieved 99.83% accuracy** in fixed-point AI modeling of neural network using **OOP based Python** for algorithm verification
- **RTL Area optimization by 60%** through resource sharing for **adders & LUTs** using **serialization technique** with **SIPO-shift registers**.
- **Throughput increase** by **58 MHz** with critical path balancing between **priority encoder & exponential approximator**
- **Reduced latency** by **50%** using simple **dual port memory ram** in read-then-write mode for weight learning update
- Improved performance accuracy **by 3.88%** against baseline accelerator through **priority encoder and fixed-point exponential approximator** for advanced weight update engine design
- **Linear Feedback Shift Register** based learning rule implementation for **stochastic** weight update to model **Hebbian Learning**

An efficient recurrent neural network (RNN) inference chip design for MIMO OFDM symbol detection

[Github link] *Spring 23-Fall 23*

- **Performed 100% AI model development** using **C++ simulator** $\langle 16, 10 \rangle$ fixed-point format with template libraries like **std::vector**
- **Reduced IP area usage by 33.3%** through **DSP48E1 IP** integration at RTL-level for inference MAC operation
- **Increased data transfer throughput 5x** by Ethernet-MAC IP integration with target accelerator design at 125 MHz frequency
- **Boosted design frequency by 100 MHz** by implementing **Clock Domain Crossing** to achieve 200 MHz frequency for target accelerator through synchronizing with Ethernet PHY communication at 125 MHz using **Ping-pong buffer, CDC AXI-handshake IPs and Asynchronous FIFOs**
- **Improved verification coverage by 30%** compared to baseline BIST testbench in frame data transfer between Ethernet-PHY and target accelerator by creating over 4 protocol-variant Ethernet frame stimulus patters including error-injection and backpressure scenarios

Technical Skills

- **Techniques:** RTL design, STA, Power optimization, Clock-Domain-Crossing, UVM, DFT, FPGA-IP Integration, ASIC implementation
- **Languages:** SystemVerilog, Verilog, Python (OOP), C++, Tcl, Linux Shell
- **Tools:** Cadence Suite, TensorFlow, PyTorch, Vivado, Modelsim, Quartus
- **Concepts:** SVA Formal Verification, AXI-DMA, AXI4, UART, Ethernet-TCP/UDP, SPI, VGA, RISC V ISA, Cache memory mapping