Deekshith Anantha

Raleigh, NC | +1(984) 382-1068 | ananthadeekshith@gmail.com | linkedin.com/in/deekshith-anantha | Portfolio

EDUCATION

North Carolina State University, Raleigh **Aug 2025 - May 2027**
Master of Science (MS), Computer Engineering *4.00/4.00*

- **Coursework:** Microprocessor Architecture, Architecture of Parallel Computers, Advanced Microarchitecture, Embedded Systems Architecture, Neural Networks, Object-Oriented Design and Development, Operating Systems, Compiler Optimization and Scheduling

RV College of Engineering, Bangalore, India **Aug 2019 - Jun 2023**
Bachelor of Engineering, Electronics and Telecommunication *7.9/10*

TECHNICAL SKILLS

- **Programming & Core Languages**: C++, Embedded C, Python, Robot framework, Assembly Language, Shell Scripting, Ruby, object, Object Oriented Programming, Data Structures and Algorithms
- **Embedded Systems & Real-Time Systems**: Bare-Metal Programming, RTOS, Microcontrollers, Digital Signal Processing, Interrupt Handling, Low-Level Driver Development, Firmware Architecture, Bootloaders, Memory Management, UART, Arinc, MATLAB, Simulink
- **Hardware & Debugging**: Lauterbach Trace32, GDB, ARM Cortex-M, GPIO, Logic Analyzer, Keil, Polyspace, Astrée, Unit Testing-, Hardware-in-the-Loop (HIL) Testing
- **Build and Deployment Tools**: Makefile, Jenkins, Git, Jira, Continuous Integration & Deployment (CI/CD), Linux/Unix
- **Software Development Lifecycle**: Agile Methodologies, Requirement-Based Development, Documentation, Component-based development

WORK EXPERIENCE

AIRBUS

Firmware Engineer **Jul 2023 - Jul 2025**

Contributed to the development of firmware for the Flight Warning Computer on Next-Gen Single Aisle Aircraft (A320s), demonstrating C/C++ and embedded systems expertise.

- Exposed to hardware-software integration, requirement-based testing, and real-test bench debugging using Trace32 and GDB, which aligns with rigorous firmware testing and debugging practices.
- Solely responsible for performing Static Analysis of the Flight Warning application using the Astrée tool to ensure compliance and software reliability.
- Engineered a CI/CD pipeline leveraging Jenkins, achieving a 50% reduction in build and deployment time and substantially optimizing the development workflow.
- Integrated the SDAC (System Data Acquisition Concentrator) test bench configuration with the Flight Warning Computer by introducing a LUA based graphical user interface, mitigating resource constraints, and increasing productivity by 30%.

Engineering Intern **Mar 2023 - Jul 2023**

Designed automation scripts using Python to streamline firmware configuration and verification under DO-178C guidelines.

- Developed an auto-process tool to automate the Change and Configuration Management process, aligning with systematic firmware documentation practices.
- Automated migration of 34 software components to the latest toolchain using Python and Shell scripting, reducing migration time by 80%.
- Enhanced cockpit simulation by optimizing over 15,000 signals using Python profiling, leading to a 15% boost in computational performance and reduced system latency.

PROJECTS

Cache and Memory Hierarchy Simulator with Stream Buffer Prefetching (C++)

- Developed a configurable simulator for multi-level cache hierarchies (L1, L2) supporting adjustable size, associativity, and block parameters.
- Implemented a stream buffer prefetching unit to model speculative data access and analyzed performance metrics such as hit/miss rates and average access time. Evaluated system-level trade-offs in latency, bandwidth, and energy using CACTI models, reinforcing concepts in performance modeling and firmware-level optimization.

Superscalar Pipeline Simulator (C++)

- Developed a cycle-accurate simulator for an out-of-order (OOO) superscalar processor to model dynamic instruction scheduling.
- Implemented core microarchitectural components essential for OOO execution, including a Reorder Buffer (ROB), a unified Issue Queue (IQ), and a Rename Map Table (RMT). Modeled the complete multi-stage pipeline (Fetch, Decode, Rename, Register Read, Dispatch, Issue, Execute, Writeback, and Retire) to resolve true data dependencies an
- Analyzed processor performance by processing instruction trace files to generate detailed cycle-by-cycle timing reports and calculate final metrics like Instructions Per Cycle (IPC).

Branch Predictor Simulator (C++)

- Designed and implemented a RISC-V branch prediction simulator in C++ supporting bimodal, gshare, and hybrid predictor architectures.
- Modeled dynamic two-bit counter tables and chooser logic to measure misprediction rates, accuracy, and pipeline efficiency across configurations. Optimized simulator runtime and validated against benchmark traces to study trade-offs in accuracy, hardware complexity, and pipeline performance.

Adaptive Modulation and Detection using Machine Learning (Matlab)

- Designed a machine learning model to perform automatic modulation classification (AMC) on signals using adaptive modulation techniques.
- Developed and trained a deep neural network (DNN) to predict the modulation scheme (e.g., QPSK, 16-QAM, 64-QAM) of a received signal, overcoming the errors and quality loss of traditional demodulation.
- Utilized Python, MATLAB, and Jupyter Notebook to pre-process signal datasets, build the ML architecture, and train the model to minimize loss.
- Evaluated model performance by comparing its classification accuracy against other established architectures, including CNN, LSTM, and PCA.