Deekshith Anantha

Raleigh, NC | +1(984) 382-1068 | ananthadeekshith@gmail.com | https://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, Embedded Systems Architecture, Object-Oriented Design and Development, Advanced Microarchitecture, Architecture of Parallel Computers, Compiler Optimization and Scheduling, Operating Systems

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

TECHNICAL SKILLS

- **Programming & Hardware**: C++, Embedded C, Python, Object Oriented Programming, Data Structures, Algorithms, Robot framework, Assembly Language, Shell Scripting, Ruby, UART, ARINC, RTOS, Microcontrollers, Bare-Metal Programming, Memory Management
- **Systems Engineering & V&V**: Systems Integration, Hardware-Software Integration (HSI), Hardware-in-the-Loop (HIL) Testing, System Performance Analysis, DO-178C Compliance, Firmware Architecture, Configuration Management
- **Build and Deployment Tools**: Makefile, Jenkins, Git, Jira, Continuous Integration & Deployment (CI/CD), Linux/Unix
- **Simulation, Modeling & Analysis Tools**: MATLAB, Simulink, Astrée/Polyspace (Static Analysis), CACTI, Lauterbach Trace32, GDB, Logic Analyzer
- **Software Development Lifecycle**: Agile Methodologies, V-model , Requirement-Based Development & Testing, Documentation, Component-based development

WORK EXPERIENCE

AIRBUS

Systems Software Engineer **Jul 2023 - Jul 2025**

Contributed to the development of firmware for the Flight Warning Computer (FWC) on Next-Gen Single Aisle Aircraft (A320s), demonstrating C/C++ and Systems Life-Cycle expertise.
- Executed **hardware-software integration (HSI)** and **requirement-based testing** for the FWC, performing system-level debugging on real-test benches using **Trace32** and **GDB** to **verify and validate** system behavior against design specifications.
- Solely responsible for performing **Static Analysis** of the Flight Warning application to ensure **system reliability** and **regulatory compliance.**
- Optimized the **systems verification workflow** by engineering a **CI/CD** pipeline (Jenkins), reducing the build-and-test cycle by **50%** and enabling faster design iteration.
- Managed **system interfaces** by integrating the SDAC (System Data Acquisition Concentrator) test bench with the FWC, developing a new GUI based on LUA script that increased test team productivity by 30%.

Engineering Intern **Mar 2023 - Jul 2023**

Developed Python automation tools to support the Verification and Validation (V&V) process for avionics firmware, ensuring strict adherence to DO-178C safety-critical standards.
- Developed an auto-process tool to automate the **Change and Configuration Management** process, aligning with systematic firmware documentation practices.
- Improved **systems-level Configuration Management (CM)** and documentation by developing a tool to automate the change control and migration process for 34 software components.
- Conducted **system performance analysis** on a cockpit simulation environment and optimized 15,000 signals to reduce **system latency** and boost computational performance by 15%.

PROJECTS

Cache and Memory Hierarchy Simulator with Stream Buffer Prefetching
- 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
- 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
- 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
- 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.