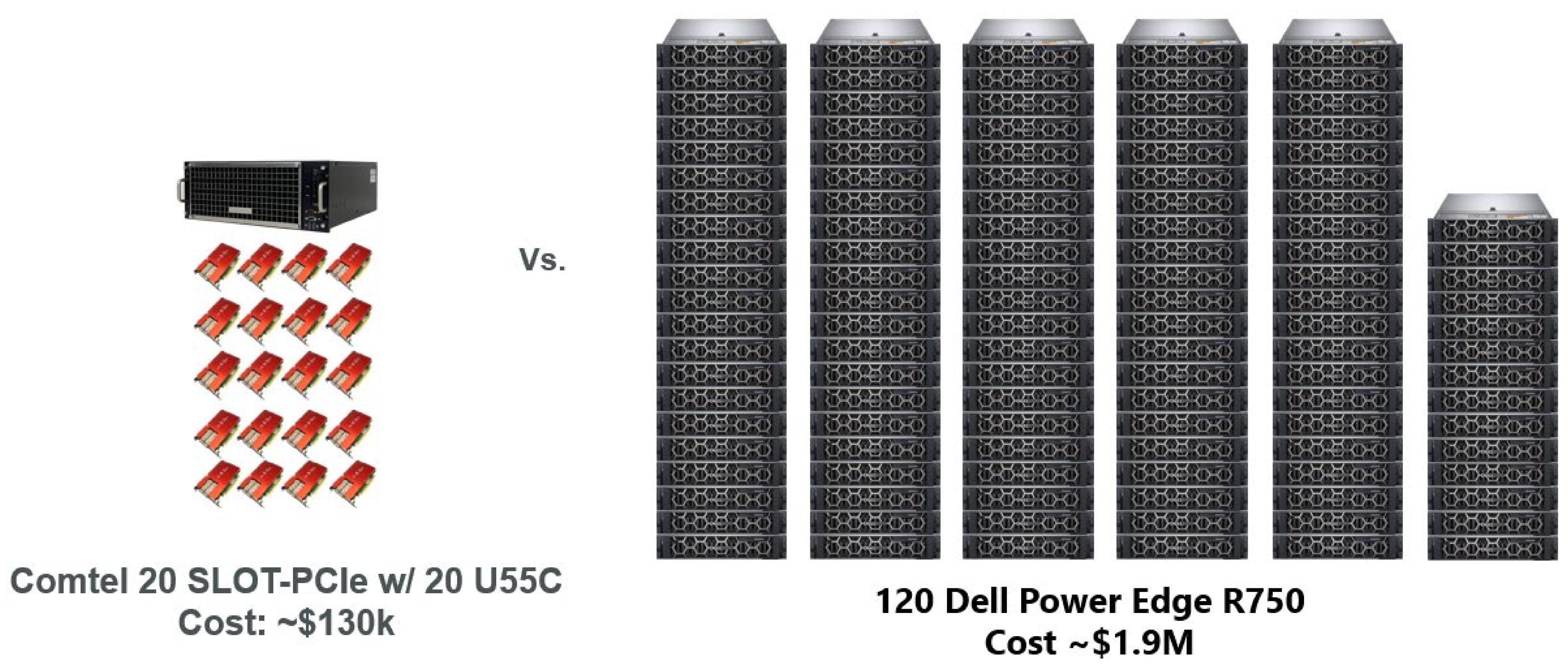
Phase 1 SBIR: Multi-tenancy Platforms using AWS F1 Instance

Juan Deaton juan.deaton@alignup.space, Christian Rodriquez Chris@alignup.space, Sharif Siddiqui Sharif@alignup.space, Phillip Choi Phillip.Choi@intelsat.com

Challenge and Approach

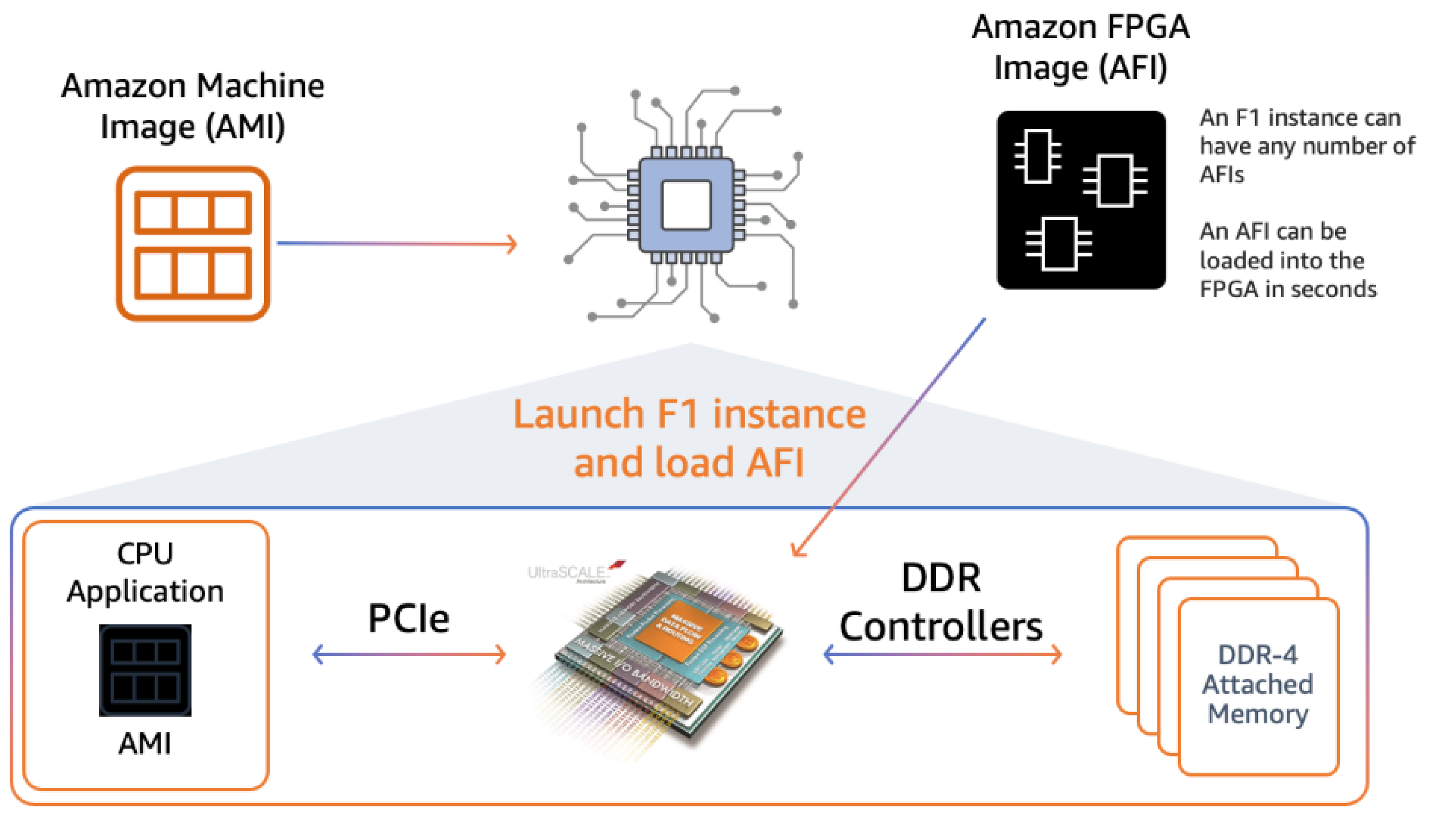
Ensuring multi-tenancy at satellite terminals and allowing multiple end-users and customers to share the same platform while receiving tailored access network services are key technical challenges. Additionally, this approach emphasizes elastic provisioning leveraging AWS F1 instances underlying computing resources for multi-tenants creating virtualization-capable remote head-ends catering to diverse services. This project is to examine feasibility of using F1 instances for multi-tenancy architectures as a reference solution for multi-tenant virtualized adaptive digital IF waveforms by relying on heterogeneous compute architectures. CPU and FPGA heterogeneous architecture are key to advancing satellite industry towards high performance solutions to deploy waveform virtualization.



AWS F1 Instances

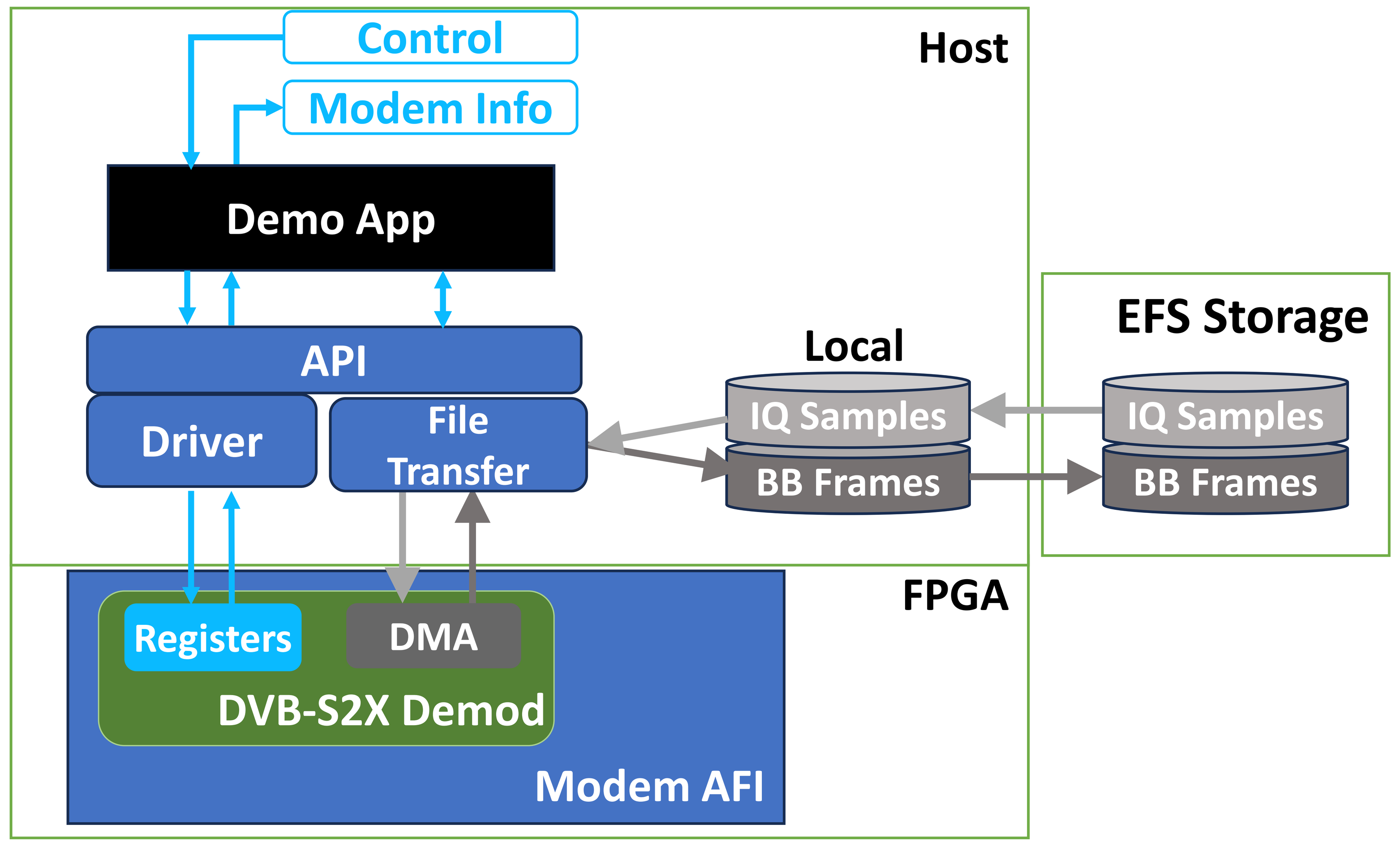
Amazon AWS F1 instances are specialized EC2 instances designed for hardware acceleration using FPGAs. These instances allow developers to deploy custom FPGA-based hardware for a wide range of applications, such as high-performance computing, machine learning, genomics, and financial modeling. F1 instances offer a flexible and scalable environment where users can develop and optimize custom logic while leveraging AWS's cloud infrastructure. With tools like the AWS FPGA Development Kit (FDK), users can accelerate workloads by creating and running hardware-optimized algorithms directly on the FPGA hardware within the F1 instances.

Instance Size	FPGA count	FPGA memory DDR-4 (GiB)	Number of host CPUs (vCPUs)	Host memory (GiB)	NVMe instance storage (GB)	Network bandwidth
f1.2xlarge	1	4 x 16	8	122	470	Up to 10 Gbps
f1.4xlarge	2	8 x 16	16	244	940	Up to 10 Gbps
f1.16xlarge	8	32 x 16	64	976	4 x 940	25 Gbps



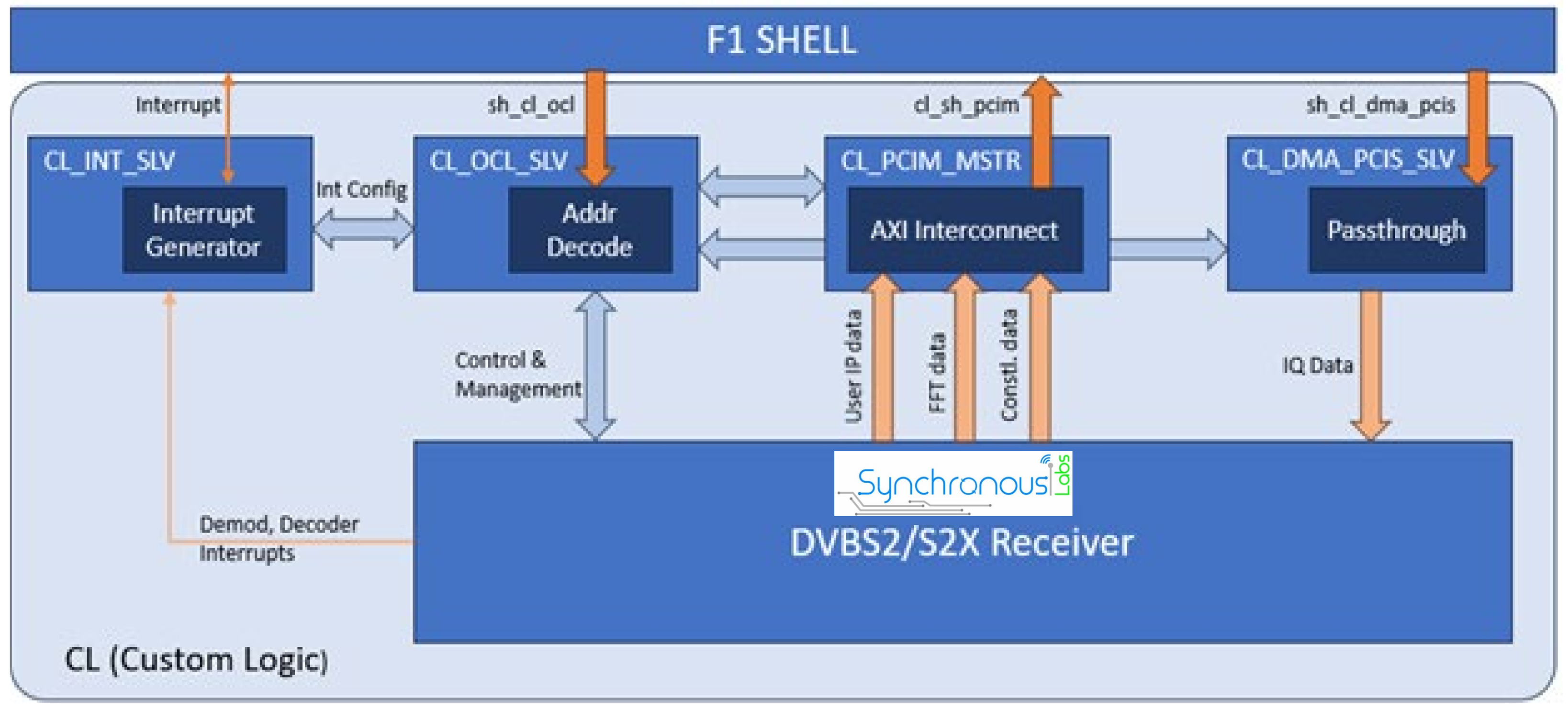
Design Approach

A DVB-S2X demodulation system is implemented on an AWS F1 instance. The system's core components include a Demo Application, API, Driver, and the DVB-S2X demodulator (Modem AFI) on the FPGA. The workflow involves retrieving IQ samples from EFS storage, processing them through the demodulator, and storing the resulting baseband frames back in EFS. The Demo App interfaces with the system to access status parameters, control the demodulator, and provide real-time monitoring capabilities. This architecture enables efficient cloud-based signal processing and demonstrates the integration of FPGA acceleration with AWS cloud services for SATCOM applications.



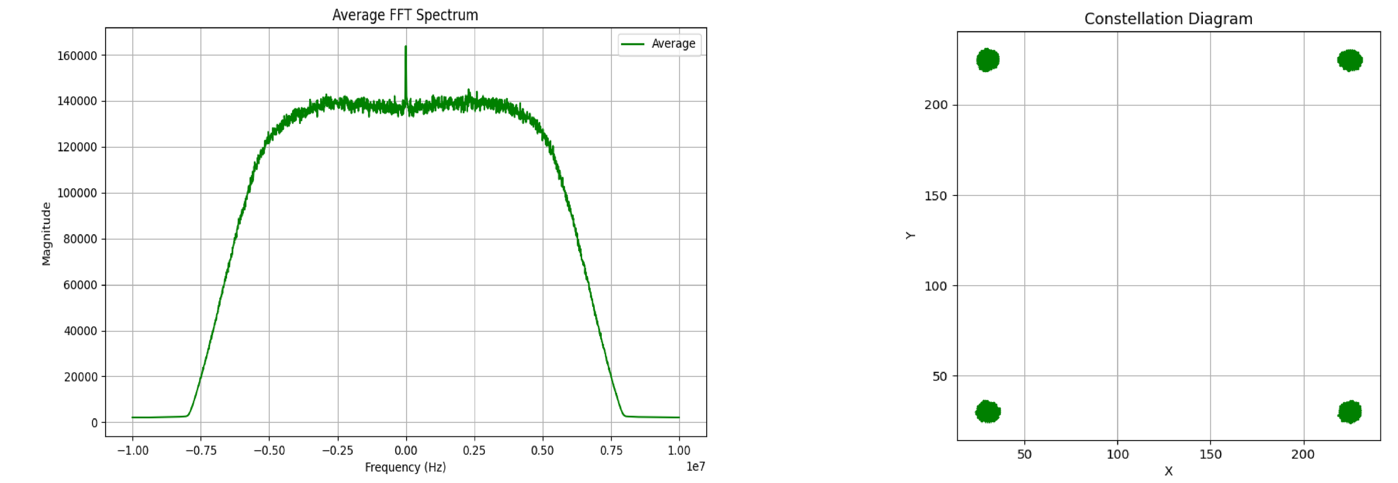
Synchronous IP Integration

The integration of the Synchronous Labs DVB-S2X receiver into the AWS F1 instance involves connecting key AXI interfaces between the custom logic (CL) and the F1 shell. The receiver leverages four primary interfaces for communication:sh_cl_ocl (AXI-Lite) – Used for configuration, control, and management of the receiver via register access.sh_cl_dma_pcis (AXI4 slave) – Handles high-speed IQ data transfers from the host to the receiver.cl_sh_pcim (AXI4 master) – Sends output data streams (e.g., User IP, FFT, constellation data) back to the host.Interrupts – Allows the DVB-S2X receiver to signal events (e.g., demodulator and decoder interrupts) to the shell for efficient processing and management.This architecture facilitates high-performance communication between the host and the FPGA while maintaining flexibility and scalability for cloud-based acceleration on EC2 F1 instances.

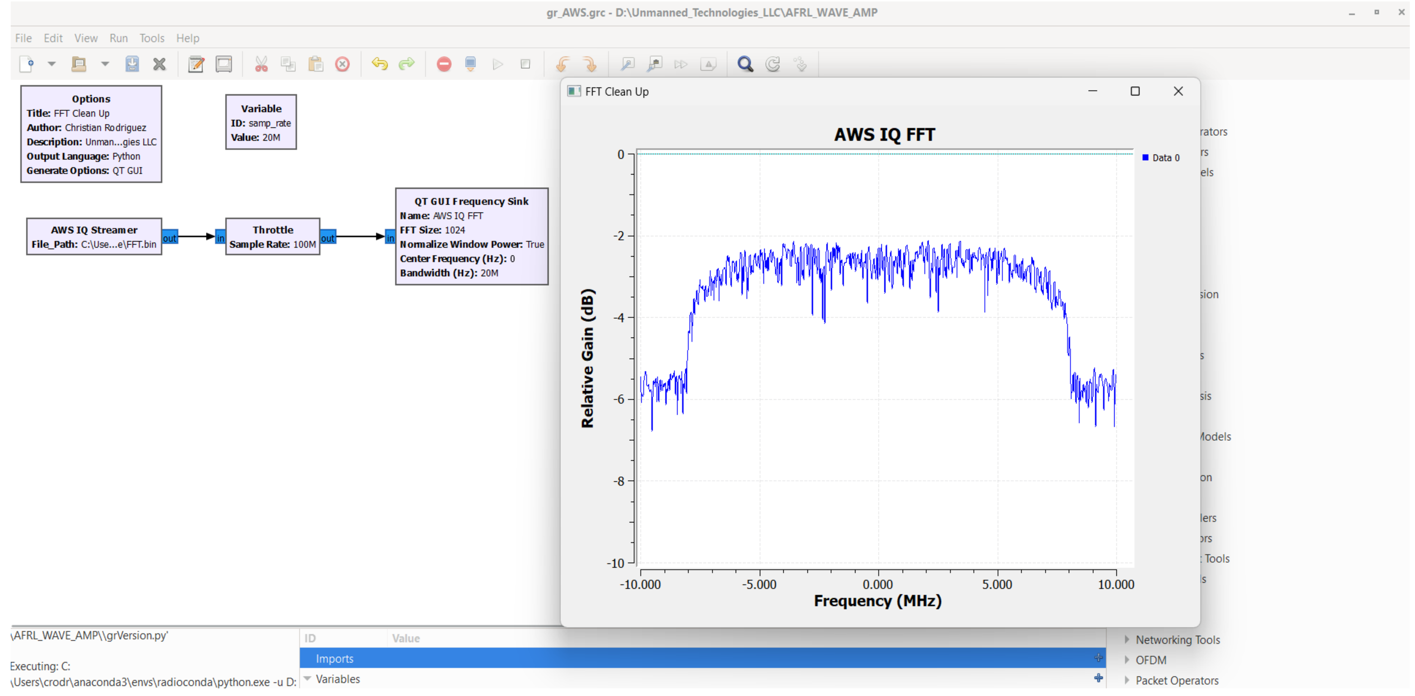


Verification Software

The output of the FPGA accelerated demodulator stored on EFS storage is verified using custom Python scripts. The FFT python script reads the IQ samples and performs a Fast Fourier Transform (FFT) on the data. The DC spike seen in the data is typical in software-defined radio FFTs and can be removed post-processing. The constellation diagram script reads constellation information and graphs it. Having control over the scripts allows the user to perform verification in batches as data is collected.

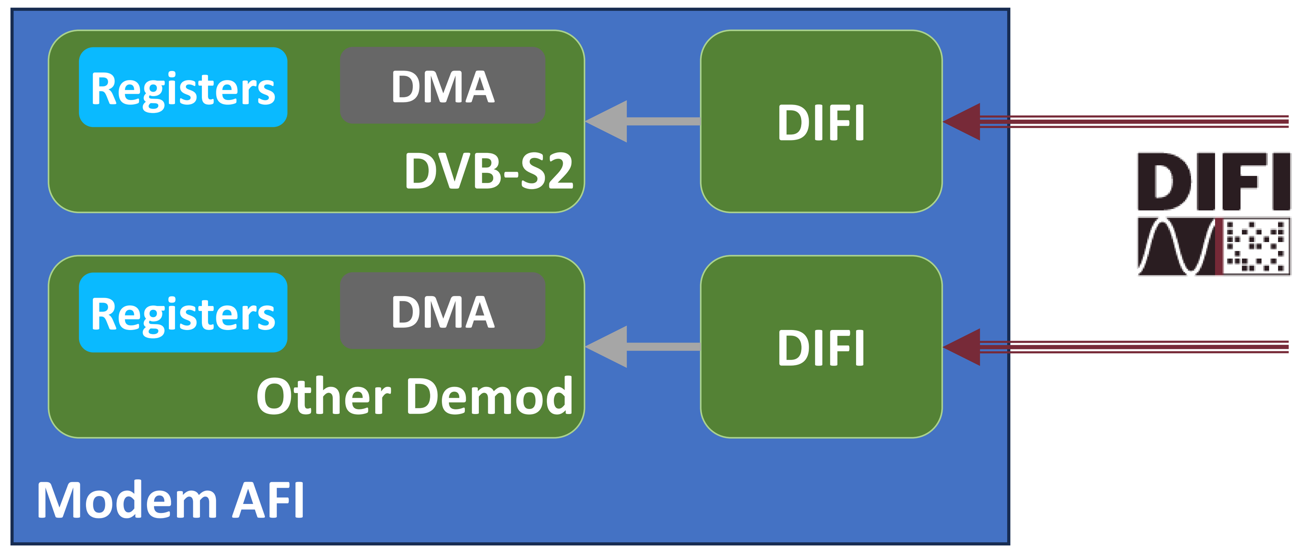


In addition, a streaming service was created for GNURadio to take the IQ output in AWS and convert it into a GNURadio friendly format. This capability empowers the user to leverage GNURadio for further signal processing. A normalization and windowing function is applied to the signal in order to remove the DC spike.



Conclusions/Future Work

Future SBIR efforts will integrate a DIFI core into the network interface of F1 or future AWS FPGA instances to enable real-time SATCOM demodulation. Additionally, we plan to incorporate a third-party waveform to demonstrate multi-tenancy, allowing two different waveforms to operate concurrently on the same platform.



Sponsors/Thanks

Funding for this project was provided by Air Force Research Lab under the supervision of Dr. Khanh Pham. Special thanks to AWS for providing credits to support experimentation.