FINN Radio ML on FPGA

Dataset and Equipment

- The dataset can be obtained through this link

Inside the Dataset
• Value is stored in int8 with range [-128, 127] • 27 modulations • 26 SNRs per modulation (-20 dB through +30 dB in steps of 2) • 4096 frames per modulation-SNR combination • 1024 complex time-series samples per frame • Samples as floating point in-phase and quadrature (I/Q) components, resulting in a (1024,2) frame shape • 2.875.392 frames in total (90% for training, 10% for both testing/validation)

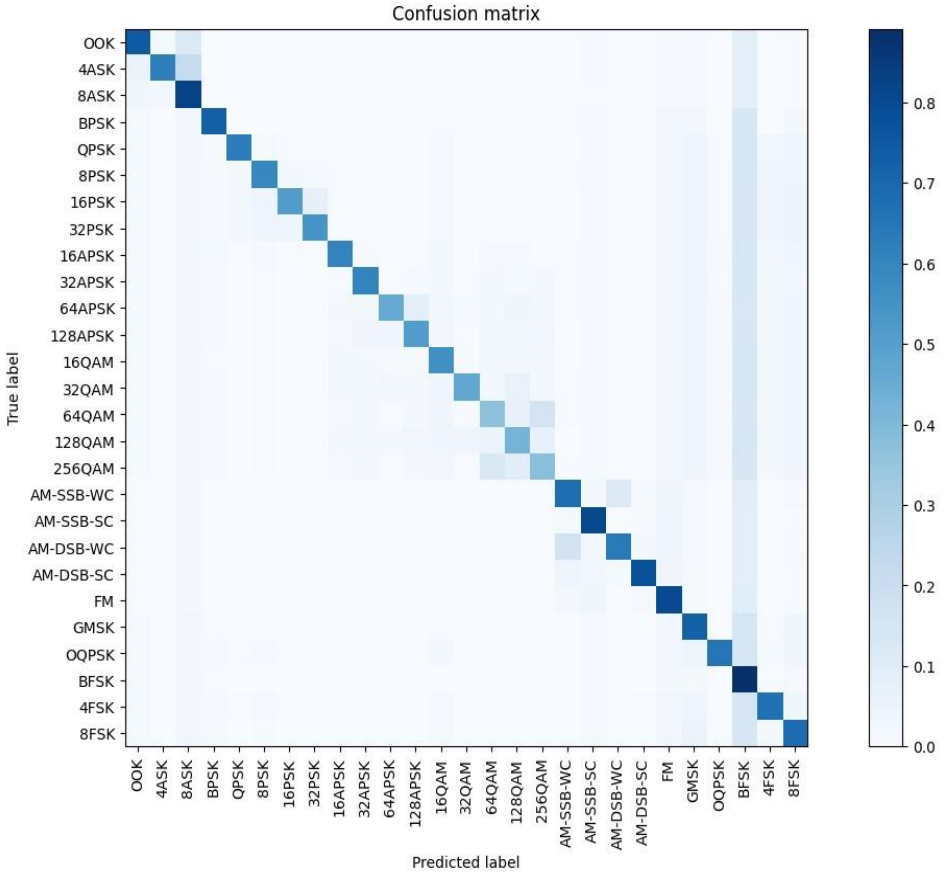
- The model was trained and transformed on the Titan I7 machine using an NVIDIA GPU
- The FGPA used in this research: ZCU104

Report from building and training model (VGG10 - 27modulations)

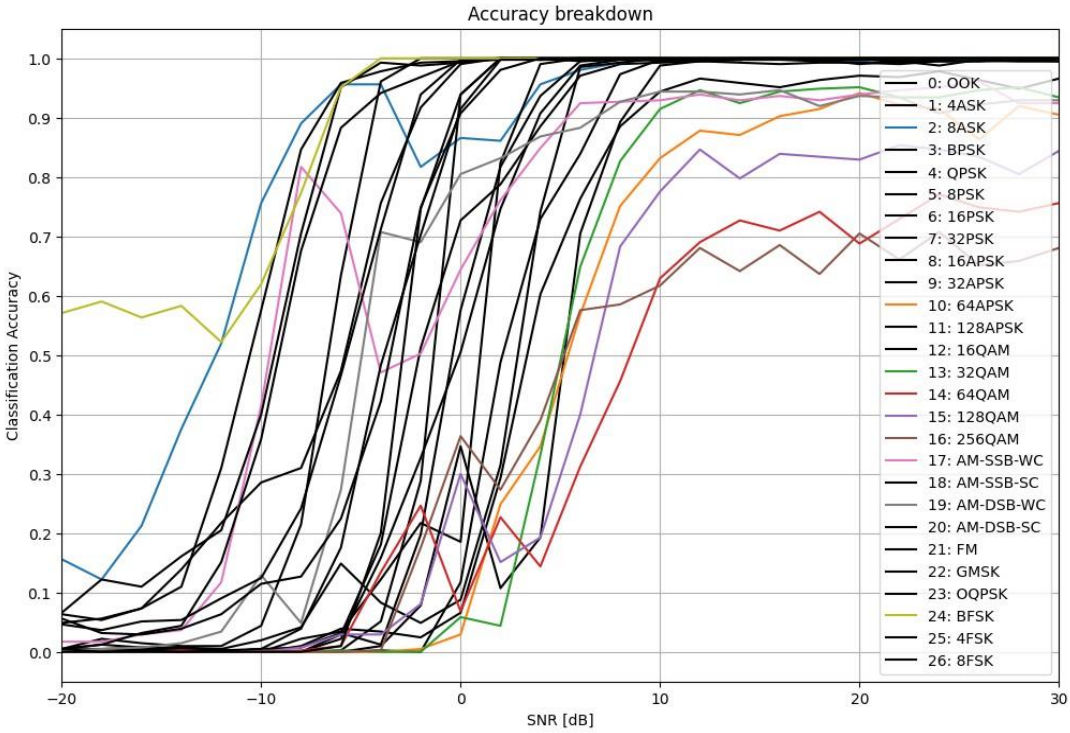
- The model followed the architecture (VGG10) mentioned in *RadioML Meets FINN: Enabling Future RF Applications With FPGA Streaming Architectures*

Building Model
• Seven 1D-Convolutional layers, kernel size of 3, each followed by Batch Normalization, ReLU activation, and 1D Maxpool that reduces the layer by half • One Flatten Layer • Two Linear Layers, each follwed by Batch Normalization and ReLU activation • One Final Linear Layers performing modulation classification
Total parameters (all trainable): 160932

- The model ran for 15 epochs (stopped early due to no improvements), and has accuracy of 61.5746% for all SNRs. At the highest SNR (30dB), the model has an accuracy of 95.4291%



Confusion Matrix (across all SNRs)



Accuracy Breakdown of Each Classification (across all SNRs)

Transforming from model through FINN framework

- FINN provided a custom build transformation that generate an accelerator from a VGG10 model. The steps can be reproduced following this github repository (contact Dr. Boyang if you do not have access).
- Inside the notebook, the transformation steps include:
    - Exporting a Brevitas model to a FINN-ONNX model.
    - Performing network surgery to remove the initial quantization node.
    - Generate the accelerator from FINN's build flow.
    - Transfer the accelerator onto FPGA and run verification.

Report from running throughput and verification on FPGA

- *Throughput or FPS is measured in [images/s] or frames per second.
- For runtime, the model is tested on 1 batch with size 1024 frames with 27 modulations.

<table>
<tr><th colspan="2">VGG10 (27 modulations) – 8 bits for weights and activations (w8a8)</th></tr>
<tr><td colspan="2">runtime[ms]: 420.3033447265625
*throughput[images/s]: 2436.3355962969686
DRAM_in_bandwidth[MB/s]: 4.989615301216191
DRAM_out_bandwidth[MB/s]: 0.0024363355962969684
fclk[mhz]: 249.9975
batch_size: 1024
fold_input[ms]: 0.0896453857421875
pack_input[ms]: 0.0667572021484375
copy_input_data_to_device[ms]: 2.5424957275390625
copy_output_data_from_device[ms]: 0.1270771026611328
unpack_output[ms]: 0.5090236663818359
unfold_output[ms]: 0.064849853515625</td></tr>
<tr><th>Accuracy on FPGA</th><th>Accuracy before FINN transformation</th></tr>
<tr><td>Accuracy at 30db on FPGA:
10564 / 11070 = 95.42908%</td><td>Accuracy at 30db before FINN transformation:
95.4291%</td></tr>
<tr><td>Accuracy at all SNRS on FPGA:
177222 / 287820= 61.57390%</td><td>Accuracy at all SNRS before FINN transformation: 61.5746%</td></tr>
</table>

FINN estimated report of resources usage from post synthesis (before the final stage of transformation)

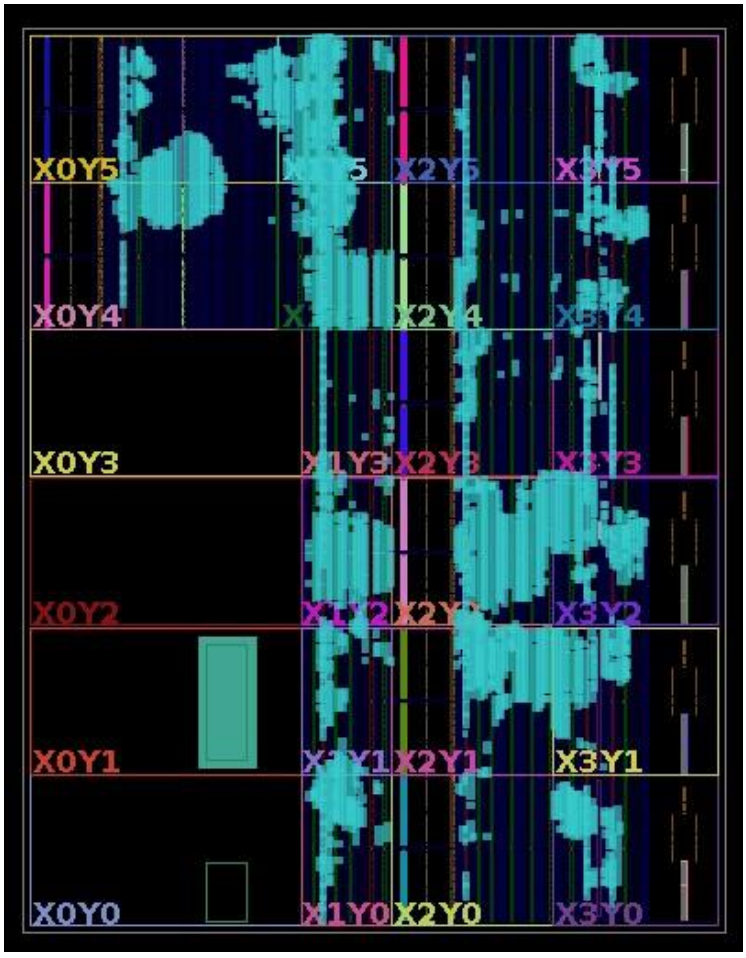
- Report generated directly from running FINN's build flow, which give the estimated resource usage of the model after running synthesis.

Post synthesis report
Total Flip flops: 56656 Total Look-up Tables: 33630 Total BRAM_36K: 177 Total BRAM_18K: 61

Report from running implementation of the final stitch project on Vivado

- After running FINN's build flow, the final stitch project is generated along with the .xpr file. The report below is from running implementation of that file on Vivado.
- More information about the final report can be found in this github discussion

Running implementation on Vivado
Total Flip flops: 453 Total Look-up Tables: 2680 Total BRAM: 205



Hardware Design Layout

Contact:

- Boyang Wang, boyang.wang@uc.edu
- Ryan Evans, evans2ra@mail.uc.edu
- Phu Le, lepq@mail.uc.edu