Weekly Progress Report

Project: TinyML-Based Project on FPGA Board with RISC-V Core

Name: Aman Chauhan
Branch: Btech. Computer Science and Engineering (CSE Core)
Roll Number: 22BCE0476
College: Vellore Institute of Technology, Vellore
Mentor: Dr. Sudip Roy

Week: *1 report*
Date: *23rd May 2025*

Introduction

- This report summarizes my progress during the first week of research and practical learning for the TinyML-based project on an FPGA platform using a RISC-V core. The focus was to build a strong foundational understanding of the three core technologies involved: FPGA hardware, RISC-V architecture, and Tiny Machine Learning (TinyML). Additionally, I started working on practical aspects such as FPGA programming and dumping techniques which are critical for developing and debugging hardware designs.

Learning and Research Activities

2.1 Basics of FPGA, RISC-V, and TinyML

- Divided my initial study into learning the fundamentals of FPGA, RISC-V, and TinyML to build a solid base for project development.
- Explored the architecture and working principles of FPGA boards and RISC-V processors.
- Understood the concepts of TinyML for deploying machine learning on resource-constrained devices.

2.2 Software Tools and Setup

- Investigated Vivado software as the primary development environment for FPGA programming.
- Studied the installation requirements for Vivado Standard ML Edition (approx. 22 GB download, 80 GB free disk space needed).
- Planned software setup to facilitate efficient FPGA design and testing.

Xilinx Unified Installer 2022.2: Windows Self Extracting Web Installer (EXE - 209.61 MB)

MD5 SUM Value : 985168f6920c5ee2111c5d16573330e1

Download Verification

Digests | Signature | Public Key

Xilinx Unified Installer 2022.2: Linux Self Extracting Web Installer (BIN - 271.02 MB)

MD5 SUM Value : 9bf473b6be0b8531e70fd3d5c0fe4817

Download Verification

Digests | Signature | Public Key

VIVADO ML Editions

Learning and Research Activities

2.3 Hands-On FPGA Programming and Dumping

- Actively followed detailed tutorial playlists on FPGA programming to learn design flow and hardware description languages.
- Focused on FPGA dumping techniques which involve extracting bitstreams and debugging data from the FPGA board, a vital skill for verification and troubleshooting.
- Video tutorials used for reference:
  - FPGA programming playlists:
    - https://www.youtube.com/playlist?list=PLO89phzZmnHiqLIfMWk9fGK8-ZtpO4O-j
    - https://www.youtube.com/playlist?list=PLbFgDf51ZkCEJb9MvxKIs-0q2obouwf-f
  - FPGA dumping tutorials:
    - https://www.youtube.com/playlist?list=PL5AmAh9QoSK7Fwk9vOJu-3VqBng_HjGFc
    - https://www.youtube.com/watch?v=QR1cvB9-xcM

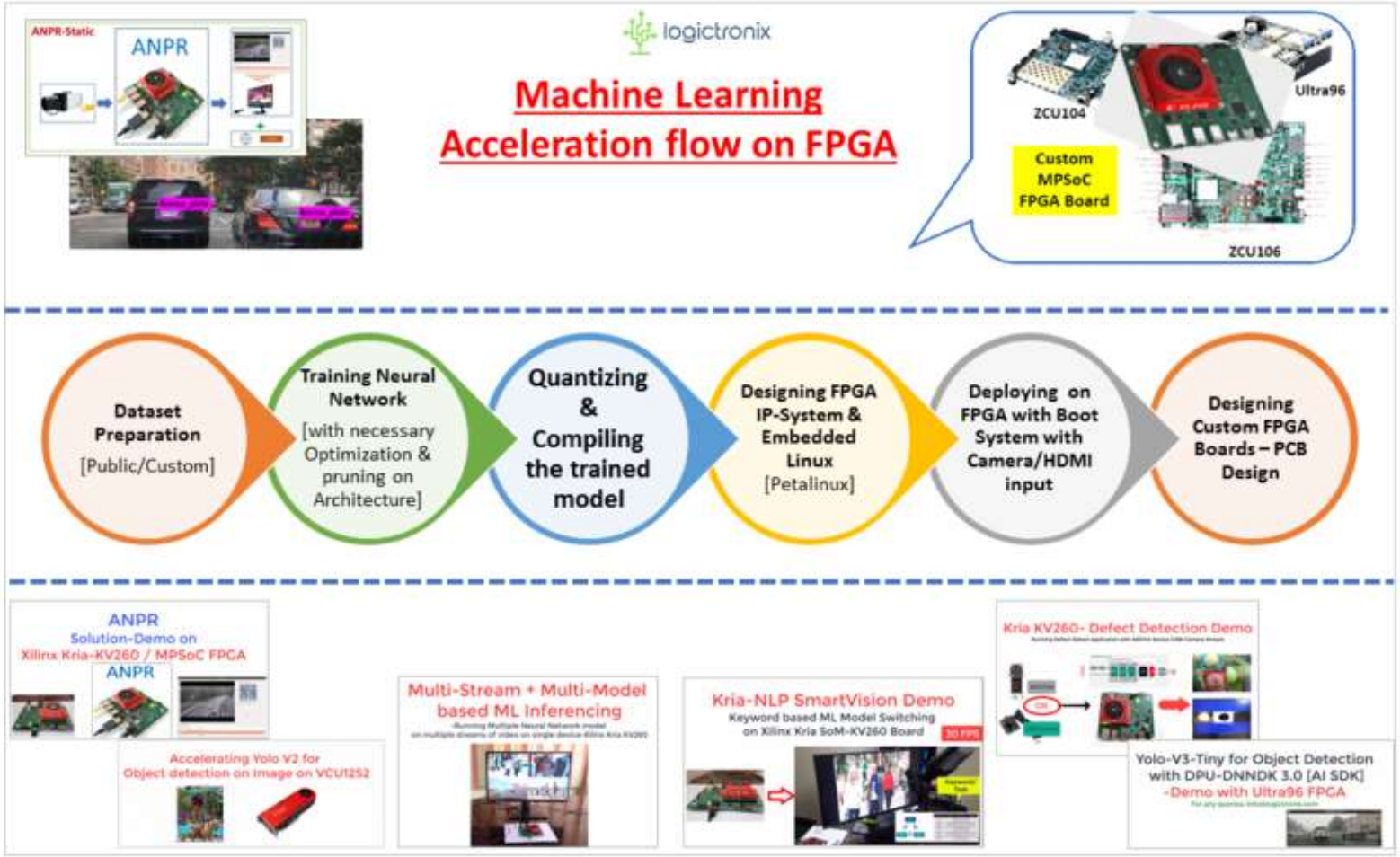
Learning and Research Activities

2.4 TinyML Learning Resources

- Studied Arduino-based TinyML projects and foundational tutorials to understand deploying lightweight ML models on edge devices.
- Explored TinyML GitHub repositories to access code, papers, and real-world projects:
  - https://github.com/mit-han-lab/tinyml
  - https://github.com/gigwegbe/tinyml-papers-and-projects
  - https://github.com/PacktPublishing/TinyML-Cookbook
  - https://github.com/Efinix-Inc/tinyml

2.5 RISC-V Toolchain Research and Setup

- Researched installation and integration of the RISC-V GCC toolchain compatible with Vivado to support development on FPGA:
  - https://github.com/riscv-collab/riscv-gnu-toolchain
  - https://gnutoolchains.com/risc-v/
- Planned to install and configure the toolchain to compile and run RISC-V based programs on FPGA.


ANPR-Static
ANPR
logictronix
Machine Learning
Acceleration flow on FPGA
ZCU104
Ultra96
Custom
MPSoC
FPGA Board
ZCU106
Dataset
Preparation
[Public/Custom]
Training Neural
Network
[with necessary
Optimization &
pruning on
Architecture]
Quantizing
&
Compiling
the trained
model
Designing FPGA
IP-System &
Embedded
Linux
[Petalinux]
Deploying on
FPGA with Boot
System with
Camera/HDMI
input
Designing
Custom FPGA
Boards – PCB
Design
ANPR
Solution-Demo on
Xilinx Kria-KV260 / MPSoC FPGA
ANPR
Accelerating Yolo V2 for
Object detection on image on VCU1252
Multi-Stream + Multi-Model
based ML Inferencing
Kria-NLP SmartVision Demo
Keyword based ML Model Switching
on Xilinx Kria SoM-KV260 Board
30 FPS
Kria KV260- Defect Detection Demo
Yolo-V3-Tiny for Object Detection
with DPU-DNNDK 3.0 [AI SDK]
-Demo with Ultra96 FPGA

Learning and Research Activities

2.6 TinyML on FPGA Specific Resources

- Reviewed specialized academic materials and presentations focusing on TinyML implementations on FPGA boards with RISC-V cores to understand the system-level integration challenges:
  - https://cms.tinyml.org/wp-content/uploads/talks2021/tinyML_Talks_Altaf_Khan_and_Martin_Kellermann_-210512.pdf

2.7 Reference Books for Deep Understanding

- Reading and referencing key books to deepen technical knowledge:
  - *songml_tiny devices*
  - *Rohan_TinyML*
  - Monte Dalrymple's book on RISC-V FPGA development

Summary Table of Progress and Next Steps

Area	Current Progress	Next Steps
FPGA Fundamentals	Completed basic tutorials and explored Vivado software	Setup Vivado environment; begin hardware design and synthesis practice
RISC-V Toolchain	Studied repositories and installation guides	Install RISC-V GCC toolchain and perform test compilations and runs
TinyML Basics	Reviewed Arduino-based TinyML projects and GitHub repos	Start developing and training TinyML models for FPGA deployment
FPGA Dumping	Learned FPGA bitstream extraction and debugging techniques	Practice dumping on physical FPGA board; troubleshoot design issues
Software Environment	Assessed Vivado installation needs and planned setup	Install and configure Vivado Standard ML Edition
TinyML on FPGA Papers	Analyzed academic presentations on TinyML FPGA systems	Investigate hardware-software co-design approaches in depth

Conclusion

- During this first week, I laid the groundwork for my TinyML FPGA project by thoroughly researching the fundamentals and essential tools. My work included both theoretical learning and practical preparation, particularly focusing on FPGA programming and dumping techniques necessary for debugging hardware designs. The upcoming weeks will emphasize hands-on implementation, including software environment setup, toolchain installation, and the start of TinyML model integration on the FPGA platform.