Machine Learning Accelerator for Speech Recognition

Final Demo Presentation

Group 33
Sherman Lin
Zhenze Zhao
Xinyu Chen
Xinyue Chen

Project Description

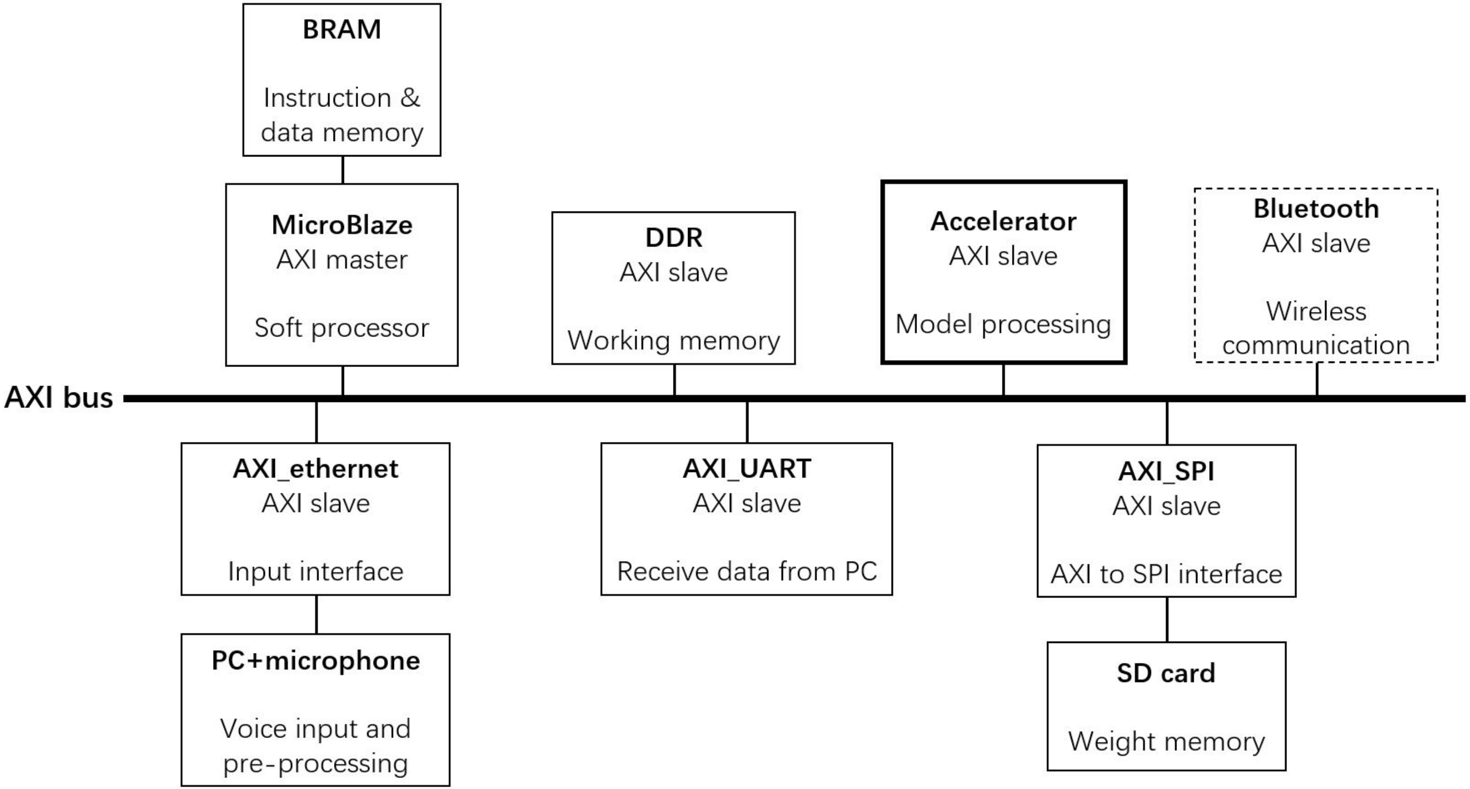
How does it work?

- Model Weights/Biases
  - Trained model weights and biases are transferred from PC to DRAM via IP/MAC packets.
- Speech Input
  - A microphone captures voice input (triggered by FPGA’s push button).
  - Audio signal is preprocessed on PC and sent to FPGA accelerator via IP/MAC packets.
- Hardware Implementation
  - MicroBlaze handles system control & data management.
  - MicroBlaze loads weights and biases from DRAM and inputs to accelerator.
  - FPGA accelerator processes input/filter data for model inference.
  - Model prediction sent to PC using Bluetooth
  - Prediction confidence probability sent to 7-segment display

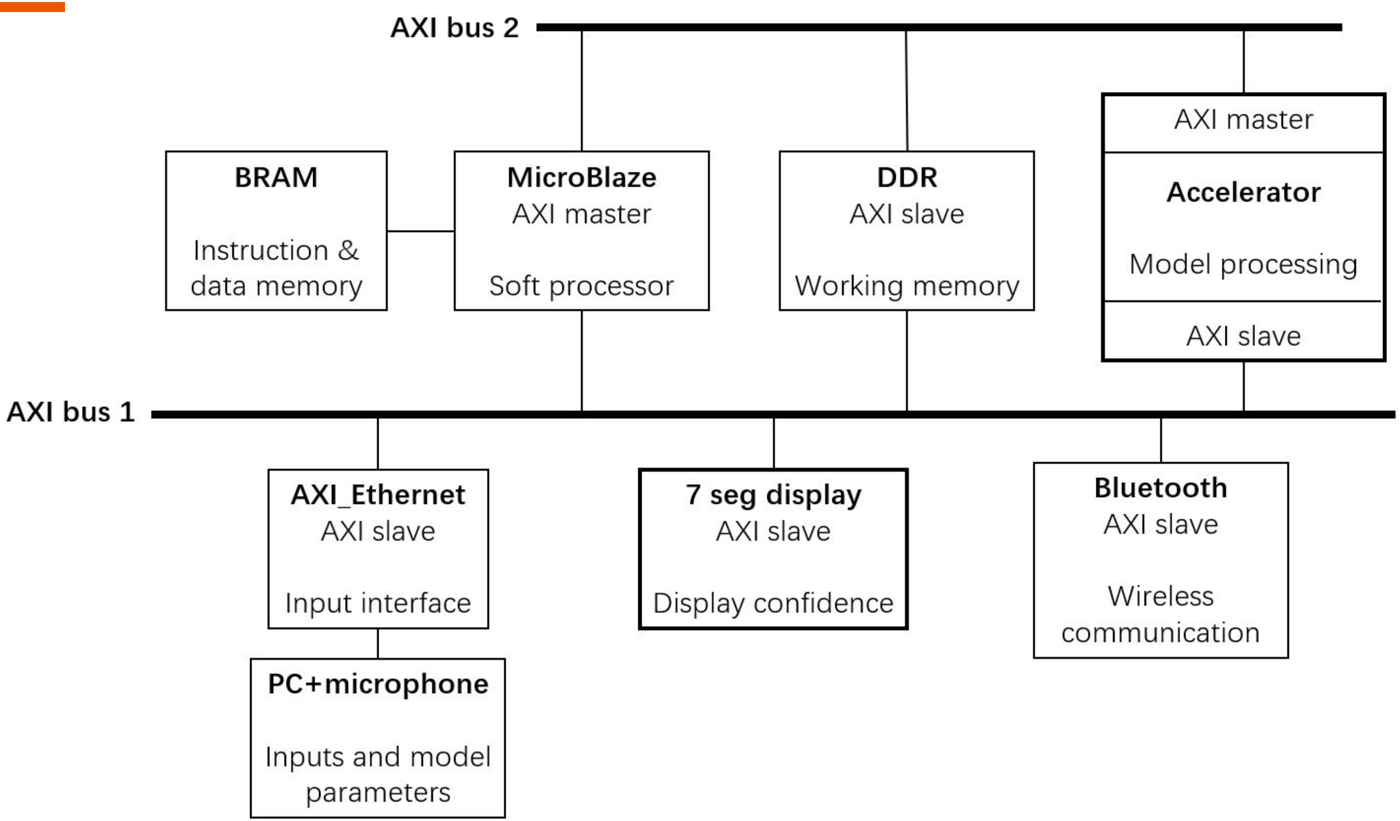
Initial Goals & Current Implementation Differences

- SD Card to store model parameters binary file
  - Could not resolve issue with required code library
  - Use DRAM to store instead
- On-board microphone integration
  - Input audio data needs to be preprocessed on PC
  - Inefficient and unnecessary back and forth data transfers
  - Use PC microphone instead
- Flash memory access without Microblaze involvement
  - Underestimated final size of model
  - Flash memory capacity not sufficient to store entire model
- UART transfer of input audio data from PC to FPGA
  - Bluetooth module also uses UART transmission
  - Use Ethernet instead to avoid conflict with Bluetooth transmission

Previous Block Diagram



Final Block Diagram



Ethernet Data Trasnfer

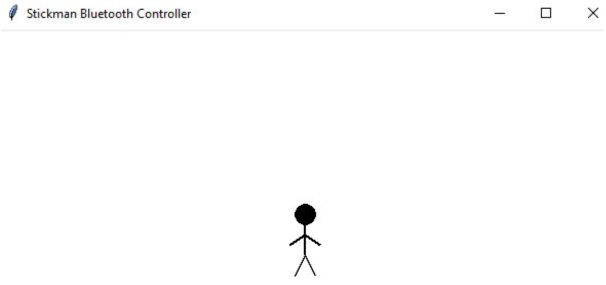
- Model Parameters Binary File
  - Tensor data encoded using Python script on PC
  - Python script divides data into Ethernet allowable sized packets
  - ACK/NACK system resolves dropped/out of order packets
  - Microblaze receives binary file and parses/decodes data
- Input Audio Data
  - FPGA pushbutton triggers a request sent to PC
  - PC records 1 second of audio using microphone
  - Audio input is preprocssed into a spectrogram, a 124 x 129 matrix
  - data is divided into packets and sent back to FPGA
  - Microblaze receives input data and stores into DRAM

Custom Acclereator IP

- 8 independent Processing Elements (PEs)
- Implemented multiplication and accumulation function in each PE
- Ping-Pong local buffer to speed up loading data
- Independent DDR access without Microblaze to alleviate bus congestion
- 0.78 GOPS peak performance (1 operation = 1 multiplication + 1 accumulation)

Bluetooth Radio PMOD

- The PMOD BT2 is a Bluetooth serial module used for wireless communication between the FPGA and a PC or mobile device. Acts as a wireless UART bridge, operating at 115200 baud rate.
- AXIuartlite sends the model prediction results from fpga to bluetooth module
- PC receives keywords then pass it to custom python GUI
- GUI displays a “stickman” that moves based on the command:
  - “left” “right” “up” “down” → the stickman moves
  - “stop” “go” “ yes” “no” → a dialog bubble appears



Software Inference Flow

- Reception of Model Binary File
    - Parse tensor data and store into DRAM
- Main Loop
    - Poll for FPGA push button
        - send request to PC for audio input
    - Reception of input audio data from PC
    - Model Inference
        - Load required tensor data from DRAM during each model layer
    - Model prediction result sent via Bluetooth to PC
    - Output prediction confidence probability to 7-segment display

Project Complexity Score

Component	**Complexity Score**
raw IP/MAC packets from Python	0.25
Use of 7-Seg Display	0.20
IP Core implemented in FPGA	0.25 - 2.0 (estimated:1.5)
Software Algorithm on Microblaze	0.10 - 1.0 (estimated:1.0)
Meaningful visualization of program run/statistics/results	0.25
Bluetooth Radio PMOD	1
Total	4.20

Potential Improvements

- Investigate techniques to reduce data transfer time
  - Implement compression algorithm during data transfer
- Explore solutions to further reduce model size
- Explore solutions to improve quantized model accuracy
- Enable Datacache
- Fully utilized PE array
  - data transfer (data bus width, fetch strategy, DMA)

Video Demo

project_demo.mp4

Thank You For Your Time!

Q&A