Enabling TinyML Inference on Resource Constraint Edge Devices

Uttej Kallakuri (ukalla1@umbc.edu)

Energy Efficient and High Performance Systems Lab

Department of Computer Science and Electrical Engineering

University of Maryland Baltimore County

Overview

- Intersection between machine learning (deep learning) and edge devices called TinyML.
- TinyML enables deployment of small DL models into a tiny edge devices.
- Analysis and interpretation of data locally on the devices and acts in real time.
- Compress model for resource constrained deployment.
  - Quantization techniques
  - Pruning techniques
- Led to inventions and is leading to the rapid growth of IoT fields
  - e.g., smart manufacturing, smart health, autonomous driving, etc.
- Thus, it allows for real-time analysis and interpretation of data
  - massive advantages in terms of latency, privacy, and cost [1,2]

- The primary goal of TinyML is to improve the adequacy of DL systems
  - Requiring less computation and less data
  - This facilitates the giant market of edge AI and the IoT [17].
- According to ABI Research, a total of 2.5 billion devices are expected to be shipped with a TinyML chipset by 2030.
- Device focus on,
  - advanced automation
  - low cost
  - low latency in transmitting data
  - and ultra-power-efficient Artificial Intelligence (AI) chipsets.
- These chipsets are known as edge AI or embedded AI
  - they perform AI inference almost fully on the board.
- The training phase, for these devices, depends on servers or cloud.

ML Deployment on MCUs

	Model	Model Result in Desktop			Inference in Devices			Result after Deployment		
		ACC	Model Size	Platform	Name	Platform	Metrics	Latency	Ram	Flash Memory
[9]	SVM	84%	-	-	All devices	STM X-Cube-AI expansion package, and C language platform	All 84%	<1 ms	-	-
	ANN1	99%—<1 m	-	-	F746ZG H743ZI2		Both 99%	1 ms	-	-
	KNN	99%—<1 ms	-	-	F746ZG H743ZI2		Both 92%	Both 10 ms	-	-
	ANN2	99%—<1 ms	-	-	F746ZG H743ZI2		Both 99%	Both <1 ms	-	-
	DT	99%—<1 ms	-	-	F746ZG H743ZI2		Both 99%	Both <1 ms	-	-
	ANN3	0.86	-	-	F401RE F746ZG H743ZI2 L452RE		0.86 R2	<1 ms	-	-
[10]	NN CNN	97.25% 99%	15 MB 7172 KB	TFLite and TFliteConver	F746ZG	X-CUBE-AI tool	100%	330 ms	135.68	668.97
[11]	CNN1 CNN2	98.53% 99.02%	185 KB	TF Lite	OpenMV H7 board STM32H743VI.	TF-Convert	95.28% 98.84%	20 FPS	-	-
[12]	CNN SqueezeNet SqueezeNet2	99.83% 98.50% 98.93%	1.5 MB 8.0 MB 3.8 MB	-	OpenMV H7 STM32H743VI.	-	99.83% 98.53% 98.99%	30 FPS	-	-
[13]	Keras	19%	-	TensorFlow	Taiyo Yuden EYSH-SNZWZ NRF52	-	-	-	-	-
	LSTM	93%	2.8 MB	TensorFlow	-	Tensor Flow Lite Micro-Not Support it	-	-	-	-
[14]	RNN FFNN	61%	-	-	ATMega4809	TensorFlow.	84% 93%	Both 40 Hz	Both 2 KB	Both 32 KB
[15]	NN	-	-	-	ESP32	Arduino-LMIC software	99.33% indoor 97.5%. outdoor	2 min per activity. 0.5 per gesture.	-	-
[16]	TinySpeech-X. TinySpeech-Y TinySpeech-Z TinySpeech-M	96.4% 93.6% 92.4% 91.9%	- 48.8 KB 21.6 KB -	TensorFlow Lite for Microcontroller	-	-	-	-	-	-
[17]	LetNet5 model Vehicle Neural Networks (VNN1,2)	99.53% 79.62% 81.27%	-	PyTorch	STM32 L476 board	X-Cube-AI (float32 operations)	-	14.15 ms	80 MHz	-
					NXP k64f	ARM CMSIS-NN.	-	0.97	120 MHz	-
					GAP8	PULP-NN	-	1000 fps with 1 ms	-	-

Table source [3]

Summary of TinyML MCU Devices

Processor	Flash Memory	RAM	Processor Speed (MHz)
STM32-L476RG	1 MB	128 KB	80 MHz
STM32-H743VI	2 MB	1 MbB	480 MHz
STM32 Nucleo-64 F091RC	256 KB	32 KB	48 (max: 48)
STM32 Nucleo-64 F303RE	512 KB	80 KB	72 (max: 72)
STM32 Nucleo-64 F401RE	512 KB	96 KB	84 (max: 84)
STM32 Nucleo-144 F746ZG	1 MB	340 KB	96 (max: 216)
STM32 Nucleo-144 H743ZI2	2 MB	1 MB	96 (max: 480)
STM32 Nucleo-64 L452RE	512 KB	160 KB	80 (max: 80)
STM32H747I-Disco_CPU (ARM Cortex M4+ ARM Cortex M7)	1 MB	2 MB	240 MHz (M4) + 480 MHz (M7)
STM32H743VI	2 MB	1 MB	400 MHz
GAP 8 based PULP architecture	512 kB	80 KB	22.65 Giga Operations Per Secon (GOPS)
NXP Semiconductors FRDM-K64F	1 MB	256 KB	120 MHz
ATMEGA4809	48 KB	6 KB	20 MHz
Arm CPU Cortex-M4	0.38 MB	1 MB	96 MHz
Xtensa DSP HiFi Mini	1 MB	1 MB	10 MHz
STM32H743 SoC- ARM Cortex- M7	2 MB	512 KB	480 MHz
Sparkfun Edge (Ambiq Apollo3), Arm CPU Cortex-M4	1 MB	0.38 MB	96 MHz
Tensilica HiFi, Xtensa DSP HiFi Mini processor	1 MB	1 MB	10 MHz
ESP32	448 KB	520 KiB SRAM	160 MHz–240 MHz
Taiyo Yuden EYSHSNZWZ NRF52	512 KB	64 KB	2402 MHz–2480 MHz
OpenMV Cam H7—Processor (ARM Cortex M7 480 MHz)	2 MB	1 MB	480 MHz
STM32H747I-Disco_CPU (ARM Cortex M4 + ARM Cortex M7)	1 MB	2 MB	240 MHz (M4) + 480 MHz (M7)
STM32H743VI	2 MB	1 MB	400 MHz

Table source [3]

Domain Specific Manycore

- In our previous work we present a domain-specific manycore platform
- Consists of 64 clusters. Each cluster comprises of a cluster memory of 3072 words (6KB)
  - Memory is shared between 3 processing cores with a RISC-like ISA and a 6-stage pipeline
- The cores have simplified data and instruction memory.
- A single cluster was fully placed and routed in 65nm TSMC CMOS technology using Cadence SoC Encounter.

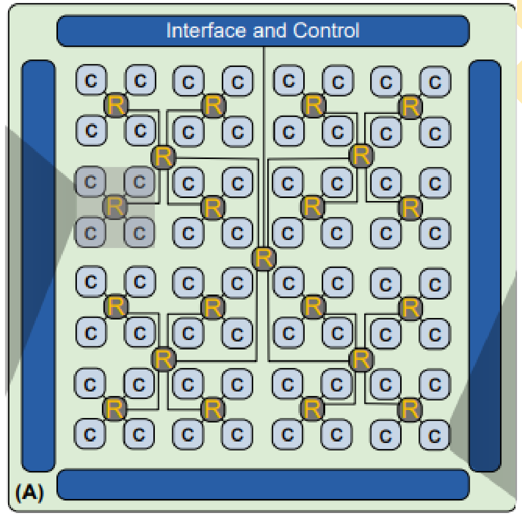


Figure source [4]

Fig: Manycore Architecture with 64 Clusters

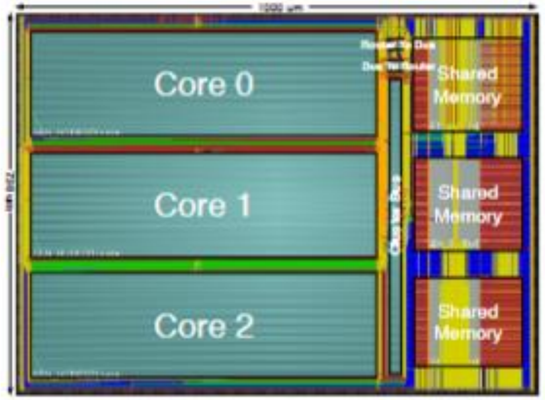


Figure source [5]

Implementation Results

Technology	TSMC 65 nm, 1 V
Logic Utilization	76.60%
Area (μm^2)	730,000
Max Freq (GHz)	1.00
Total Power (mW)	228.47

Table source [5]

Fig: Layout view and post-layout implementation results of the Cluster

Domain Specific Instructions:

- BiNMAC [4] developed specialized instructions such as
  - XNOR (1cycle)
  - PCNT (population-count) (1cycle)
  - PXNR (fused pop-count and xnor) (1cycle)
  - ACCB (bit-based accumulation) (1cycle)
  - and STT (store transpose of a block)

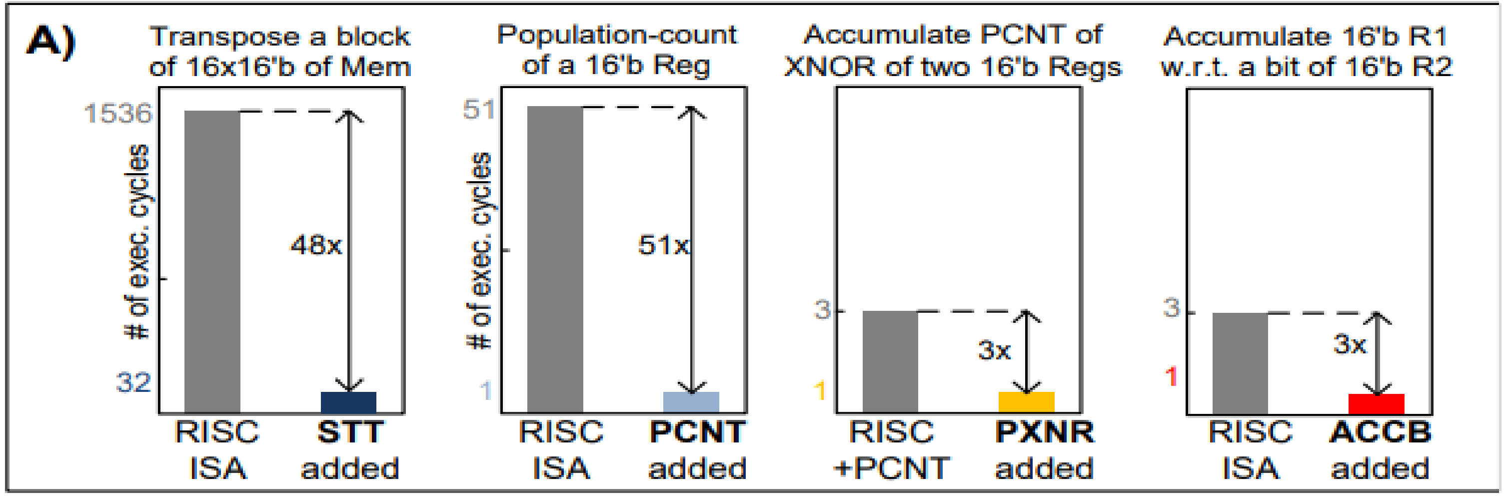


Figure source [4]

Fig: The optimization improvement factor for the new instructions as compared to an equivalent function implemented using basic RISC instructions

TinyML on Tiny FPGAs [6]

- ML on embedded edge devices is gaining increased attention.
- CFU Playground is a full-stack opensource framework for iteratively (deploy→profile→optimize) exploring the design space of lightweight accelerators.
- Open-source toolchain bundles together
  - opensource software (TensorFlow Lite Micro, GCC)
  - open-source RTL generation IP and toolkits (LiteX, VexRiscv, Migen, nMigen)
  - open-source FPGA tools for synthesis, PnR (yosys and nextpnr)

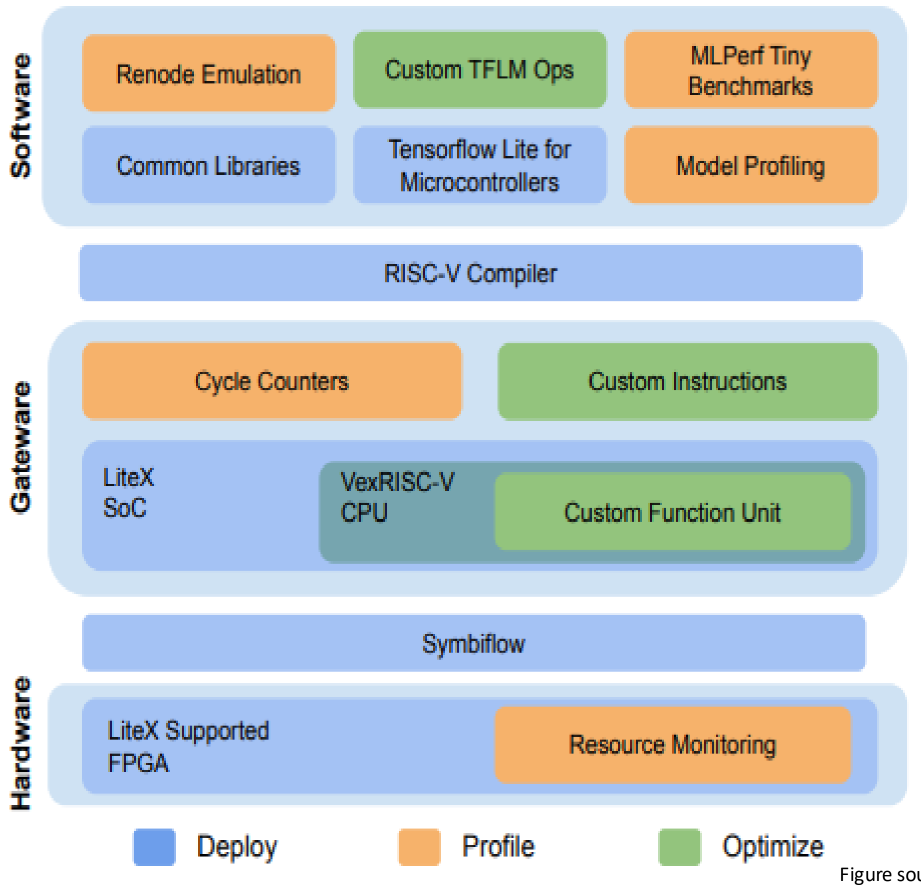


Figure source [6]

Why Tiny FPGAs

- Processors on FPGA platform are customizable
- ML computations, tend to be regular and repetitive.
- Targeted improvement (custom instructions)
  - Custom Hardware for custom instructions (CFU)
- A CFU can specialize operations.
- Flexible, configurable storage allows data to be stored and reused locally.
- An accelerator can be tightly coupled into the CPU pipeline.
  - This can be invoked by adding new custom instructions that complement the CPU’s standard functions.

Figure source [7]

Fig: TinyFPGAs A1 (left), AX2 (right)

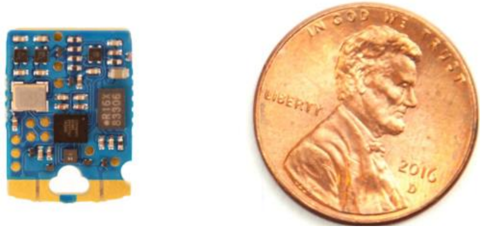


Figure source [8]

Fig: FOMU FPGA

- FPGAs, furthermore, allows bit-level flexibility.
- CFU Playground runs a complete System-on-Chip (SoC).
- Adaptable to a wide range of FPGA platforms.
- The minimum requirements for the board and its FPGA include,
    - Some means of creating a TTY / UART connection to interact with the software on the board.
    - The FPGA must have enough resources to build variations of VexRiscv CPU cores.
    - The system must have enough RAM to provide working memory for the software.
    - There must be sufficient RAM and/or ROM to hold the code and any constant data such as the TensorFlow Lite model.

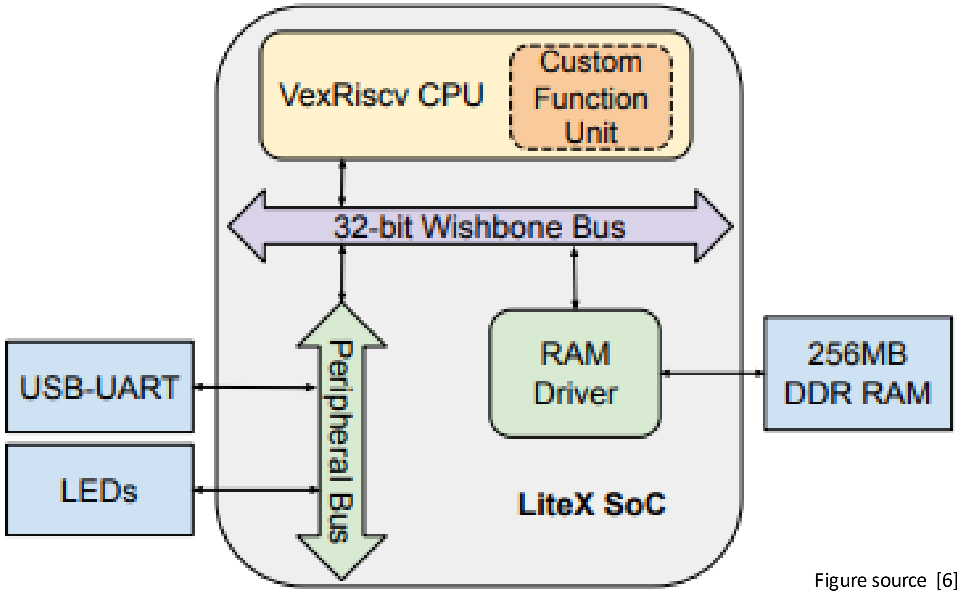


Figure source [6]

Fig: LiteX SoC with the CFU

Evaluation of CFU-Playground:

- [6]utilized CFU Playground to accelerate quantized (int8) inference of the MLPerf Tiny [18] Key-Word-Spotting model
- Tiny Fomu FPGA board,
  - It combines an iCE40UP5k FPGA
  - 5280 logic cells and 128 kB of on-chip RAM) with a 2 MB flash memory.

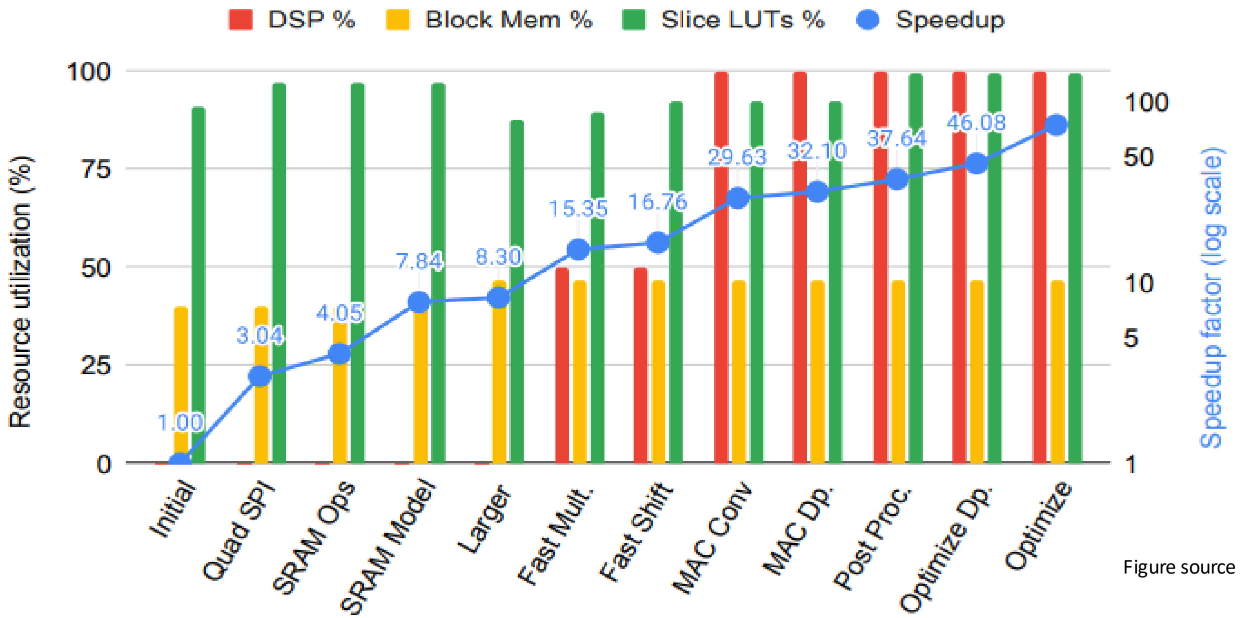
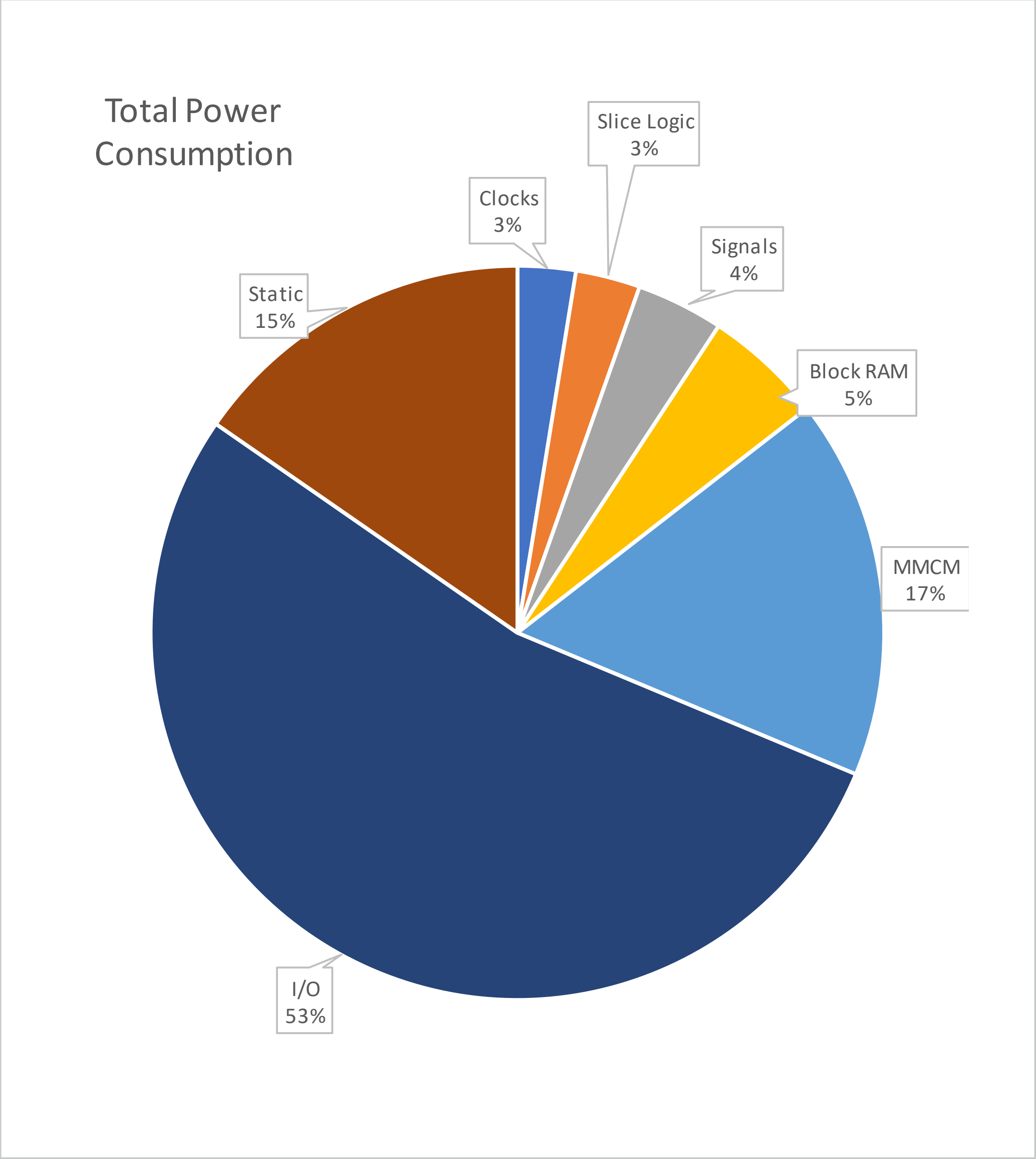


Figure source [6]

Fig: Speedup and resource usage on FOMU FPGA

Preliminary Results

- As a base line study, we are targeting a MobileNetV2 (MNV2) model.
- We aim to accelerate this model on a Digilent Nexys-4 DDR Artix-7 FPGA.
- The model is quantized down to 8-bit integers (int8).
- Profiling the (MNV2) on the Artix-7 FPGA, the unaccelerated baseline application takes abut 220M clock cycles.



Potential Plan Ahead

- To optimize for Latency
  - Profile for latency; obtain specific functions to accelerate (layerwise profiling).
  - Accelerate function using a CFU (eg: Depthwise Conv Layers).
  - Measure the improvement due to the new instruction
- To optimize for power
  - Identify major power-hungry components(eg: MMCM, I/O).
  - Replace these components with equivalent simpler ones
  - Measure the improvement in power
- To introduce structured pruning techniques to minimize storage and number of computations.

References

- [1] Lin, J.; Chen, W.M.; Lin, Y.; Cohn, J.; Gan, C.; Han, S. MCUNet: Tiny Deep Learning on IoT Devices. arXiv 2020, arXiv:2007.10319v2 neural network quantization. In International Conference on Machine Learning, pages 11875–11886. PMLR, 2021
- [2] Banbury, C.R.; Reddi, V.J.; Lam, M.; Fu, W.; Fazel, A.; Holleman, J.; Huang, X.; Hurtado, R.; Kanter, D.; Lokhmotov, A.; et al. Benchmarking TinyML Systems: Challenges and Direction. arXiv 2020, arXiv:2003.04821
- [3] Norah N. Alajlan, Dina M. Ibrahim; et al. TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications
- [4] Morteza, H.; Tinoosh, M.; et al. Binary Precision Neural Network Manycore Accelerator
- [5] Varun, S.; Tinoosh, M.; et al. The design and Implementation of a Scalable Bus-Based Cluster with Shared Memory for a Programmable Manycore Platform
- [6] Shvetank, P.; Tim, C.; Joseph, B.; Colby, B.; Alan, V. G.; Pete, W.; Tim, A.; Vijay, J.R.; et al. CFU Playground: Full-Stack Open-Source Framework for Tiny Machine Learning (tinyML) Acceleration on FPGAs
- [7] https://tinyfpga.com/
- [8] https://www.crowdsupply.com/sutajio-kosagi/fomu/
- [9] Sakr, F.; Bellotti, F.; Berta, R.; De Gloria, A. Machine Learning on Mainstream Microcontrollers. Sensors 2020, 20, 2638
- [10] Merenda, M.; Porcaro, C.; Iero, D. Edge Machine Learning for Ai-Enabled IoT Devices: A Review. Sensors 2020, 20, 2533
- [11] Paul, A.J.; Mohan, P.; Sehgal, S. Rethinking Generalization in American Sign Language Prediction for Edge Devices with Extremely Low Memory Footprint. In Proceedings of the 2020 IEEE Recent Advances in Intelligent Computational Systems, RAICS, Thiruvananthapuram, India, 3–5 December 2020; pp. 147–152
- [12] Mohan, P.; Paul, A.J.; Chirania, A. A Tiny Cnn Architecture for Medical Face Mask Detection for Resource-Constrained Endpoints. In Innovations in Electrical and Electronic Engineering; Springer: Singapore, 2020; pp. 657–670
- [13] Coffen, B.; Mahmud, M.S. TinyDL: Edge Computing and Deep Learning Based Real-Time Hand Gesture Recognition Using Wearable Sensor. In Proceedings of the 2020 IEEE International Conference on E-Health Networking, Application & Services (HEALTHCOM), Shenzhen, China, 1–2 March 2021; pp. 1–6
- [14]. Venzke, M.; Klisch, D.; Kubik, P.; Ali, A.; Missier, J.D.; Turau, V. Artificial Neural Networks for Sensor Data Classification on Small Embedded Systems. arXiv 2020, arXiv:2012.08403v1
- [15] Orfanidis, C.; Hassen, R.B.H.; Kwiek, A.; Fafoutis, X.; Jacobsson, M. A Discreet Wearable Long-Range Emergency System Based on Embedded Machine Learning. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), Kassel, Germany, 22–26 March 2021; pp. 182–187
- [16] Wong, A.; Famouri, M.; Pavlova, M.; Surana, S. TinySpeech: Attention Condensers for Deep Speech Recognition Neural Networks on Edge Devices. arXiv 2020, arXiv:2008.04245v6
- [17] De Prado, M.; Rusci, M.; Donze, R.; Capotondi, A.; Monnerat, S.; Benini, L.; Pazos, N. Robustifying the Deployment of TinyML Models for Autonomous Mini-Vehicles. Sensors 2021, 21, 1339.