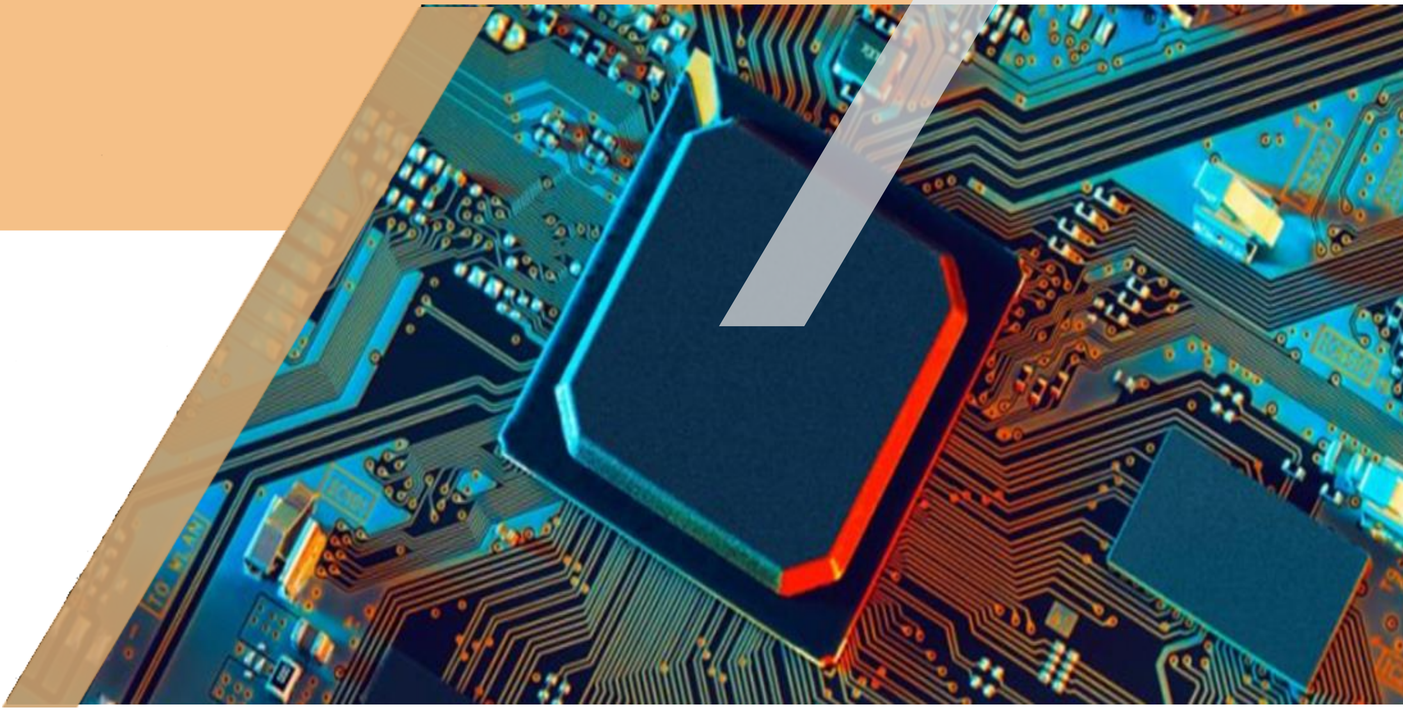
CruxML
REAL-TIME COMPUTING & MACHINE LEARNING

Email: info@cruxml.com

Website: www.cruxml.com

CruxML Capability: FPGA-based Machine Learning Whitepaper



Prof. Philip Leong, CTO

Email phwl@cruxml.com

Dr. Barry Flower, CEO

Email barry@cruxml.com

This page intentionally left blank.

FPGA-based Machine Learning

Philip Leong and Barry Flower
{phwl,barry}@cruxml.com

2020/12/11

Executive Summary

Field-programmable gate array (FPGA) technology is eminently applicable to a large class of computing problems in general and machine learning in particular. Compared with other technologies including central processing units, digital signal processors, graphics processing units and application specific integrated circuits, FPGAs offer unique Enhancement, Parallelism, Integration and Customisation (EPIC) opportunities, leading to improved latency, Size, Weight, Power and Cost (SWaP-C). This white paper describes the benefits of FPGAs for machine learning at the edge and is intended for non-specialists.

Keywords: FPGA, field programmable gate array, real-time, machine learning, edge

1 Introduction

While central processing unit (CPUs) and digital signal processors (DSPs) offer an easy-to-use, powerful and flexible implementation medium for digital systems, limitations in achievable parallelism, inter-chip bandwidth, and high power consumption have limited their performance. As highlighted in the 2018 Turing Award Lecture, this inefficiency stems from the fact that many optimisations in modern processors are unnecessary for a range of computationally demanding problems. Indeed, caches, virtual memory, speculative execution and branch prediction incur large area and power overheads but do not perform any computing [1]. Domain-specific architectures, such as graphics processor units (GPUs), have achieved substantial improvements for specialised workloads by devoting more silicon area to computation, allowing them to achieve greater levels of parallelism for many problems such as machine learning.

Compared to these technologies, field programmable gate arrays (FPGAs) allow arbitrary computer architectures to be implemented. As will be described, this has significant performance benefits in terms of throughput, latency, size, weight, power and cost. Furthermore, FPGAs are commercial off-the-shelf parts with mature design tools, an ability to accommodate legacy applications, and defence-grade and radiation hardened offerings.

2 FPGA Technology

2.1 Enhancement, Parallelism, Integration, Customisation (EPIC)

We use the acronym, EPIC, to describe the four unique features which summarise the significant benefits of FPGA Technology:

- *Enhancement* – easily explore the solution space to arrive at a good solution.
- *Parallelism* – highly flexible FPGA architectures allow increased parallelism to be achieved.

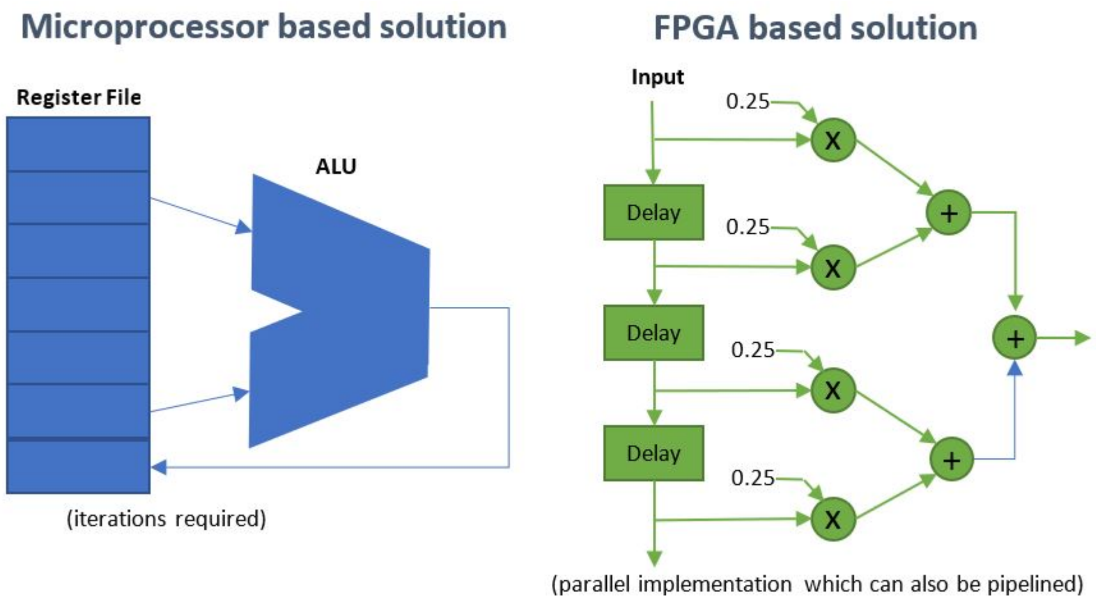


Figure 1: Illustration of a microprocessor-based FIR filter vs. an FPGA solution. In the microprocessor, operations are performed in the ALU sequentially. Furthermore, instruction decoding, caching, speculative execution, control generation and so on are required. For the FPGA approach spatial composition is used to increase the degree of parallelism. The FPGA implementation can be further parallelized through pipelining.

- *Integration* – processors and GPUs are designed to interface to off-chip memories and peripherals, whereas FPGAs provide the opportunity to include most of the system on the same device.
- *Customisation* – FPGAs enable problem-specific designs to improve efficiency (power, speed, density).

These combine to produce significant performance benefits as well as accelerate the hardware design cycle through the adoption of an agile methodology.

The key to achieving high performance is to start with a good algorithm. FPGA design changes can be implemented in a short period of time by synthesising a new bitstream and downloading it to the FPGA. It is thus easy to start with a simple design and achieve *Enhancement* in an incremental fashion. It is also feasible to compare implementations of a number of different techniques and choose the one with the best performance in terms of power and latency. In the context of machine learning, the flexibility of FPGAs are an excellent substrate for implementing the frequent advances in the field.

In most FPGA-based solutions, computations are arranged spatially rather than temporally to improve *Parallelism*. The absence of caches and instruction decoding enables the same amount of work to be done with less chip area, lower power consumption and lower heat generation compared with CPUs and GPUs [2, 3]. Figure 1 illustrates the concept of spatial versus temporal computation with the implementation of a finite impulse response (FIR) filter. Where the microprocessor-based solution requires iterations in time, the FPGA solution performs the computations in parallel without the overheads associated with interpreting programming. Spatially parallel architectures provide freedom in the design space not present in GPUs. Coarse and fine grained parallelism, pipelining, and explicit buffering can all be combined to maximise Parallelism and hence minimise latency.

Since FPGA SoCs (System on a Chip) combine high-speed memory controllers and network

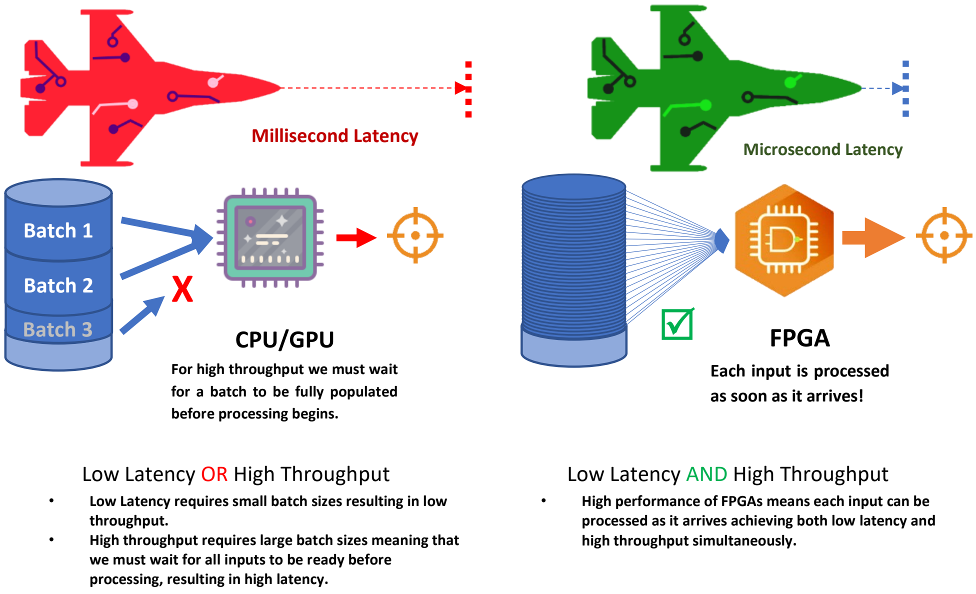

Millisecond Latency
Batch 1
Batch 2
Batch 3
X
CPU/GPU
For high throughput we must wait for a batch to be fully populated before processing begins.
Microsecond Latency
FPGA
Each input is processed as soon as it arrives!
Low Latency OR High Throughput
Low Latency requires small batch sizes resulting in low throughput.
High throughput requires large batch sizes meaning that we must wait for all inputs to be ready before processing, resulting in high latency.
Low Latency AND High Throughput
High performance of FPGAs means each input can be processed as it arrives achieving both low latency and high throughput simultaneously.

Figure 2: Comparison of the Low Latency and High Throughput advantage of FPGAs over CPUs and GPUs.

interfaces with low-level logic, entire or large parts of the system can be *Integrated*. For example, in contrast to a typical software defined radio implementation with separate data acquisition, signal processing and neural network blocks implemented on an acquisition card and GPU with processor host, it is possible to combine all of these blocks in a single FPGA device. Since FPGAs offer massive on-chip bandwidth, chip-to-chip transfers can be avoided and as a result large reductions in power, latency, weight and size are possible.

Finally, it is possible to *Customise* FPGA designs in a manner not possible with processors and GPUs. One example is binarised neural networks which achieve massive parallelism using binary weights. CPUs and GPUs do not provide support for such datatypes but FPGAs achieve unprecedented performance for these types of deep networks [3].

2.1.1 Latency, Size, Weight and Power

To achieve the absolute minimum latency, data should be processed as soon as it is received. However, in a typical GPU implementation, input data is acquired and buffered, transferred to a host CPU and then to the GPU for processing, and then back to the host before any action can be taken. Buffering is necessary as initiating a direct memory access (DMA) transfer is expensive so must be amortised over a large amount of data to be efficient. In an FPGA each input can processed as it arrives, and all operations can be completed in a single chip. Hence an *Integrated* and *Customised* single chip solution to this problem will achieve both low latency and high throughput as illustrated in Figure 2.

Size and weight are a significant factor for portable edge devices that may be carried by a person or placed on a drone to perform smart-sensing or even run an AI based inference engine. With recent SoC offerings from the major FPGA vendors it is now possible to implementing complete systems that sense, analyse, decide and report/act in real-time on stimulus gathered in the field all while running only on battery power. The lower energy demands of FPGAs mean smaller and lighter batteries can be utilised.

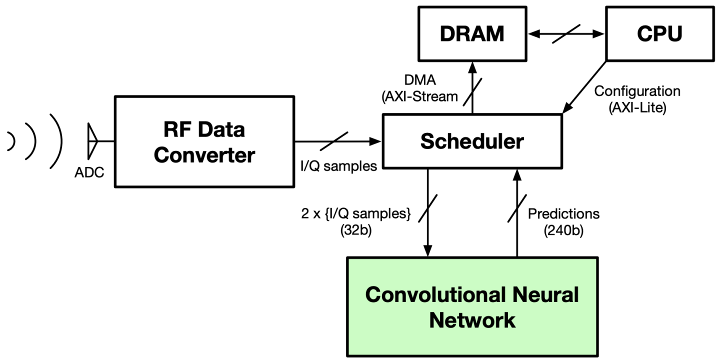


Figure 3: An example of a highly integrated automatic modulation classification system implementation. All components were implemented on a single system-on-chip FPGA.

2.2 Harsh Environments

Defence grade FPGAs are available with ruggedized packaging and higher resistance to temperature extremes. Even more challenging are space applications which combine extremely tight size, weight and power requirements with high levels of hostile, ionizing radiation. Radiation hardened FPGAs are available off-the-shelf which mitigate single-event upsets, single-event functional interrupts, single-event latchups, single-event transients and total ionizing dose effects. Moreover, system-level tools to implement not only triple-modular redundancy (TMR), but also triplicate clocks and interfaces are available to further enhance reliability.

3 Applications

In this section we highlight a number of applications which utilise the advantages of FPGA technology.

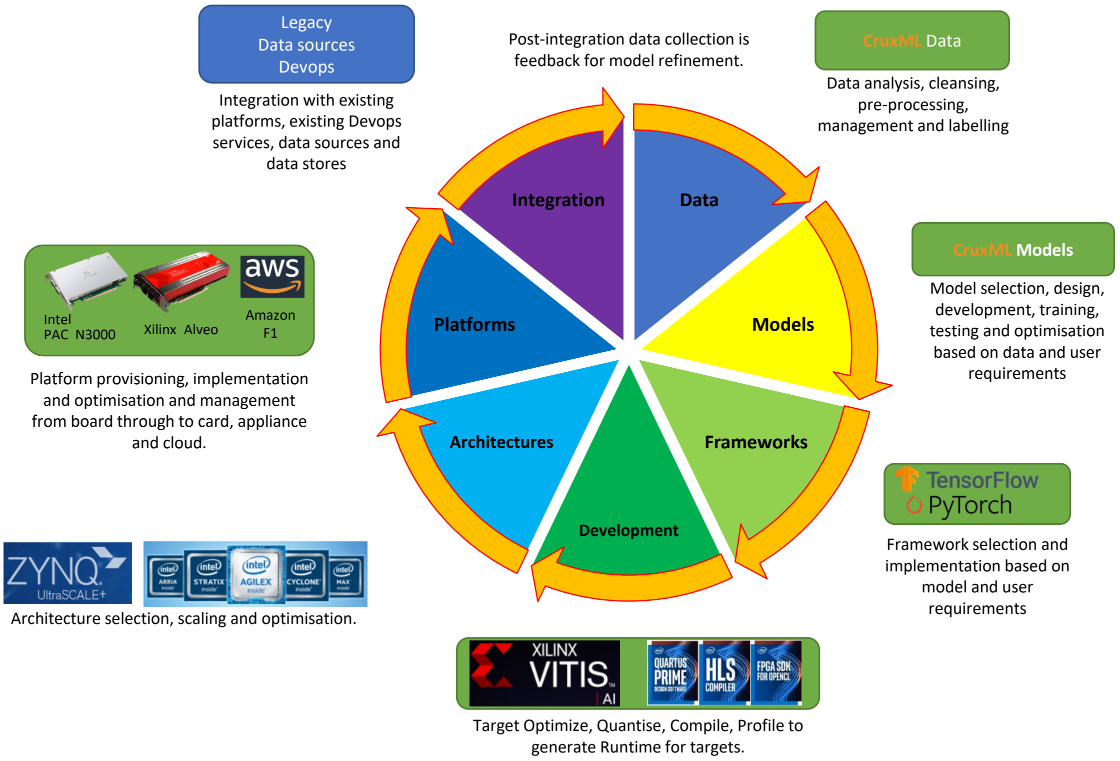
3.1 Hyperspectral Image Understanding at the Edge

Hyperspectral imaging (HSI) is important in defence, space and agriculture. It can be used to identify and analyse objects and regions, enabling better segmentation of scenes and classification of objects of interest. Although it already has widespread usage from airborne and space vehicles, HSI sensors produce vast amounts of data. Even using the CCSDS 123 HSI compression standard, transmission of the images requires high bandwidth transmitters.

A better approach for many applications is to perform machine learning (ML) at the sensor to first interpret the data, and then limit data transmission to interesting images or higher level information about the image e.g. the type of tank and its location. There is an opportunity to combine recent advances in ML image processing algorithms with improvements in FPGA technology. Next generation high-performance, low power HSI platforms will enable higher quality information to be obtained from more compact HSI platforms.

3.2 Radio Frequency Machine Learning

The understanding of uncooperative radio signals for cognitive radio and defence applications is an extremely challenging task due to the high data rates involved and the requirement for low latency. FPGAs enable the integration of radio and machine learning on a single device, allowing latency to be minimised and making them an excellent platform for physical layer radio frequency



Post-integration data collection is feedback for model refinement.

Integration with existing platforms, existing Devops services, data sources and data stores

Data analysis, cleansing, pre-processing, management and labelling

Model selection, design, development, training, testing and optimisation based on data and user requirements

Platform provisioning, implementation and optimisation and management from board through to card, appliance and cloud.

Framework selection and implementation based on model and user requirements

Architecture selection, scaling and optimisation.

Target Optimize, Quantise, Compile, Profile to generate Runtime for targets.

Figure 4: The CruxML value proposition.

(RF) applications. An example is a high performance automatic modulation classification (AMC) system, implemented using the Xilinx RFSoC platform illustrated in Figure 3. This design utilises a fully parallel ternary neural network to achieve throughput of 488K classifications per second [4]. A similar architecture could be used for applications in specific emitter identification, RF fingerprinting, radar signal understanding and RF jamming.

3.3 Cybersecurity

A fundamental challenge with cyber security systems is the need to perform sophisticated data analysis at high speed, e.g. to support 100 Gigabit Ethernet. While ML is effective at addressing many Cyber problems, its computational complexity often makes its implementation infeasible at line rates on CPUs or GPUs. FPGAs not only provide a unique, high-speed platform, they also have minimal attack surface. FPGA technology can combine misuse and anomaly based intrusion detection, and provide a general secure-by-design framework for scalable ML in cyber applications.

4 CruxML Capability

The significant advantages of FPGA technology have been describe in this whitepaper. However, developing systems which achieve the maximum performance from these devices is extremely challenging and requires expertise in computer architecture, computer arithmetic, machine learning and embedded systems. The founders of CruxML have a proven and deeply relevant track record, the company being a spinoff from the Computer Engineering Lab at the University of Sydney (CEL)[1].

[1] See `http://phwl.org/assets/papers/overview` for an overview of our research.

CruxML's mission is to enable real-time machine learning. As illustrated in Figure 4, development of machine learning systems begins with the data. Its preliminary analysis and preparation is crucial since the ultimate success of the system are dependent on the data. Model selection is extremely challenging since the ML field is changing rapidly and improved techniques are constantly being reported. CruxML's heritage in academia is beneficial in this regard. The next step, framework selection, is influenced by legacy concerns, translation paths to FPGA hardware and other factors. Utilising the appropriate FPGA platform and libraries can save man-years of development effort, however an inappropriate choice could slow development and hamper upgradability for years to come. Issues such as sovereignty, performance and design productivity also influence choices. The architecture of the resulting FPGA design needs to consider the vendor, performance requirements, target platform, cost and maintenance. Degrees of freedom include choice of datapath and control structures, arithmetic, and software/hardware partitioning. Finally, integration with existing systems needs to be managed with consideration of all its implications.

CruxML provide full-stack expertise, including collection of training data, development of machine learning models, off-the-shelf and bespoke FPGA acceleration, hardware/software integration, device drivers, printed circuit board design and training. Our goal is to supplement our clients' existing expertise with deep ML and FPGA hardware design knowledge so that novel products can be developed with an agile hardware development methodology.

References

[1] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Commun. ACM*, vol. 62, no. 2, p. 48–60, Jan. 2019. [Online]. Available: https://doi.org/10.1145/3282307

[2] Y. Li, Z. Liu, K. Xu, H. Yu, and F. Ren, "A GPU-outperforming FPGA accelerator architecture for binary convolutional neural networks," *J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 2, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3154839

[3] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. H. Leong, M. Jahre, and K. Vissers, "FINN: A framework for fast, scalable binarized neural network inference," in *Proc. ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, 2017, pp. 65–74, source code available from https://github.com/Xilinx/BNN-PYNQ. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3020078.3021744

[4] S. Tridgell, D. Boland, P. H. Leong, R. Kastner, A. Khodamoradi, and Siddhartha, "Real-time automatic modulation classification using RFSoC," in *2020 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2020, New Orleans, LA, USA, May 18-22, 2020*. IEEE, 2020, pp. 82–89. [Online]. Available: https://doi.org/10.1109/IPDPSW50202.2020.00021