SOLUTION BRIEF

High-Performance Computing
Hardware Acceleration of AI Workloads with Special Focus on Recurrent Neural Networks (RNNs)

Programmable Inference Engine

A Truly Programmable AI Megacore for FPGAs

Manjeera Digital Systems
Streaming Innovations

Introduction

FPGAs play a vital role in accelerating massively parallel workloads for multiple industry segments and will continue to do so for current and emerging applications such as Deep Neural Networks (DNNs). In case of DNN acceleration, a particular problem faced by FPGA users is the wide variety of topologies involved. There are multiple mathematical operations over and above the matrix/vector multiplications, and each of these functions will be best accelerated with a dedicated datapath in hardware. However, this would mean developing a new register transfer level (RTL) for each new function and topology. Building multiple RTLs and optimizing them is time consuming and resource intensive.

In this solution brief, we introduce Programmable Inference Engine (PIE) for Intel® FPGAs, an efficient, pre-optimized, artificial intelligence (AI) intellectual property (IP) core that balances the high performance of a hardware datapath while retaining the flexibility of software programmability. PIE can be easily parameterized for the target AI workload and can accelerate a wide variety of DNN topologies with just software updates. This document describes how PIE IP, developed by Manjeera Digital Systems can be used to speed up DNN research, development, and deployment while simultaneously delivering excellent performance by leveraging the unique abilities and advantages of FPGAs.

PIE – A High-Performance AI Inference Accelerator for FPGA

The key to high-performance AI Inference acceleration on FPGAs is to balance the superior performance of a hardware datapath while retaining the flexibility of software programmability. Manjeera Digital Systems' patented datapath accelerator, Universal Multifunction Accelerator (UMA), achieves exactly this requirement. PIE brings together UMA's patented datapath architecture that performs in-memory computing along with the high performance of Intel FPGAs to create a FPGA-based AI inference accelerator that offers RTL-like performance for a wide range of AI topologies.

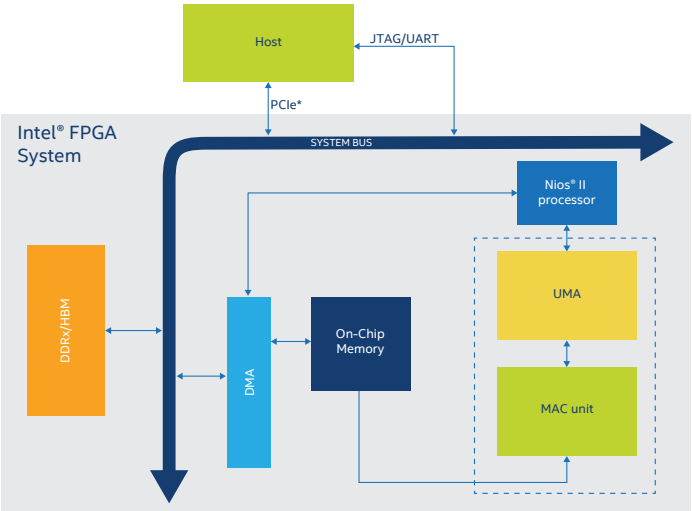

Figure 1. Simplified block diagram of PIE on the Intel® Stratix® 10 MX FPGA Development Kit.

Authors

Ramakrishna Madhava
Hardware Engineer
Intel® Corporation

Juby Jose
Engineering Manager
Intel Corporation

Dr. Venu Kandadai
CEO and Co-Founder
Manjeera Digital Systems

What is a Datapath Accelerator?

One of computing's fundamental challenges is retaining software flexibility while achieving high performance. Computing algorithms have a control path, which determines the sequence of operations, and a datapath, where all of the actual computations occur.

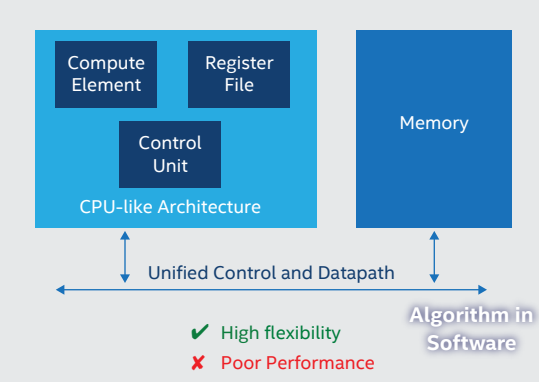

Figure 2. Traditional computing - Unified control and datapaths

Traditional CPUs combine the two paths. The algorithm takes the form of software, which runs on the processor. The software-based approach offers good flexibility but compromises on performance due to the CPU's sequential-processing nature.

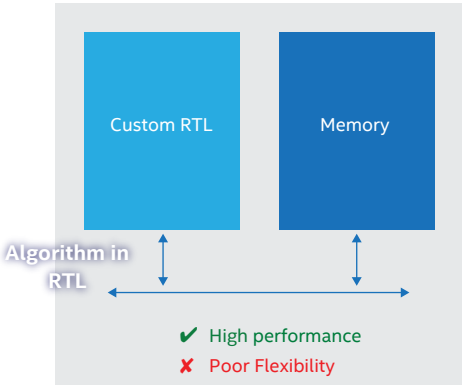

Figure 3. Hardware acceleration by encoding the datapath in RTL

On the other hand, a custom-RTL design implements the entire algorithm in hardware, as defined by an RTL description that is compiled into hardware circuits. While the hardware approach delivers high performance, it offers far less flexibility than the software-centric approach. An FPGA has the ability to implement RTL designs while providing some flexibility but customer must still develop multiple RTL designs to accelerate multiple algorithms, which increases design time and cost.

Manjeera offers a new computing approach that provides both flexibility and near-hardware-level performance. This is achieved through a programmable, lightweight, datapath processor— the UMA —that implements a unique set of datapath instructions called Middle Stratum Operations (MSOs). The MSOs are the UMA's instruction set and this higher-level instruction set achieves better performance when compared to a processor designed around a more conventional instruction set architecture.

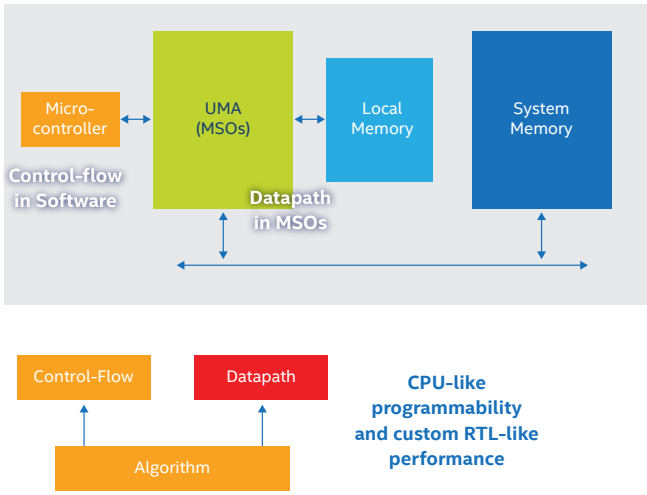

Figure 4. UMA's Programmable datapath: Software flexibility and hardware performance

MSOs directly operate on the FPGA's local memory, which is directly interfaced to the UMA. This approach eliminates data access bottlenecks providing very high performance.

Three Core Benefits of PIE

The UMA implements a datapath for an MSO-based computing machine while a microcontroller external to the UMA handles the control path. Manjeera has developed a pairing—the PIE, which offers a three-fold advantage for DNN hardware acceleration— programmability, scalability, and high performance.

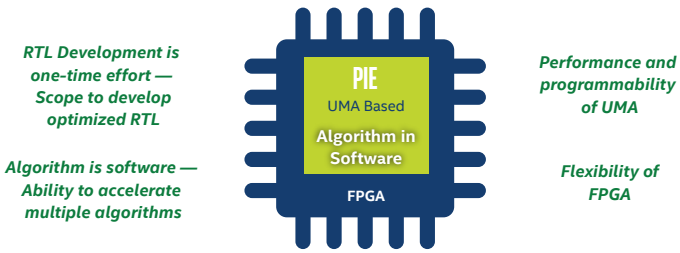

Figure 5. PIE based on UMA delivers truly programmable AI acceleration and high performance on Intel FPGAs

Programmability

The key to PIE programmability is the UMA, which addresses high-performance computing with its innovative MSO-based computing approach and pure datapath architecture. The UMA provides the acceleration of custom hardware while being fully software programmable. The software programmability makes it possible to accelerate multiple DNN algorithms without changing the RTL instantiated in the FPGA. The PIE's software primitives and APIs facilitate easy development of different AI topologies using Intel's OpenVINO™ toolkit.

Scalability

The PIE core is modular; each instance of the PIE is called a "slice." Designers can choose the number of PIE slices to incorporate into their design, trading off performance and resource utilization. Furthermore, the size of each PIE slice can be parameterized based on performance requirements and target device resources. Adding the software API stack creates a hardware abstraction layer, allowing the same PIE topology to run on any size implementation of the PIE core without software modifications.

The PIE IP can be used with a variety of Intel FPGAs as shown in Figure 6. Evaluation drops of the PIE IP are available for Intel Stratix® 10 MX FPGA and Intel Stratix 10 GX FPGA development kits upon request.

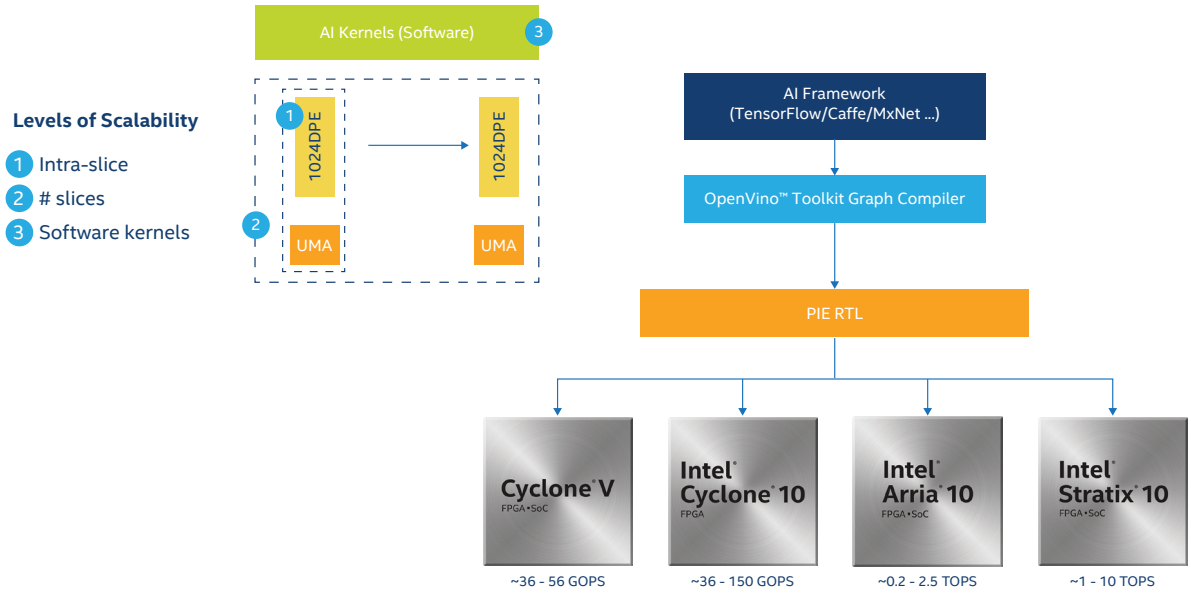

Figure 6. Scalability of PIE across Intel's FPGA portfolio

Performance

A good example of a large RNN algorithm is the Mozilla DeepSpeech algorithm for automated speech recognition (ASR). This algorithm requires a DNN topology with a large memory-intensive LSTM layer, which consumes more than 80% of the workload's total compute and memory footprint†. This is a memory-intensive and compute-intensive algorithm that also requires extremely low latency for real-time speech transcription.

Mozilla DeepSpeech was accelerated on an Intel Stratix 10 MX 2100 FPGA (speed grade -2t). The PIE IP efficiently utilizes FPGA computing resources and delivers a high level of acceleration.

The latency time required to process the DNN portion of the DeepSpeech workload on the Intel Stratix 10 MX FPGA card is four times faster than the time required by a graphics processing unit (GPU) with similar computing performance and bandwidth specifications†. This showcases the PIE IP's efficiency that achieves a high f_{MAX} with high resource utilization.

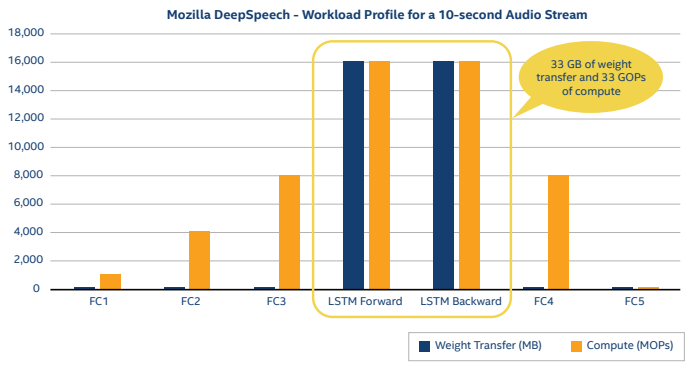

Figure 7. Mozilla DeepSpeech Compute and Memory profile for a 10-second audio stream

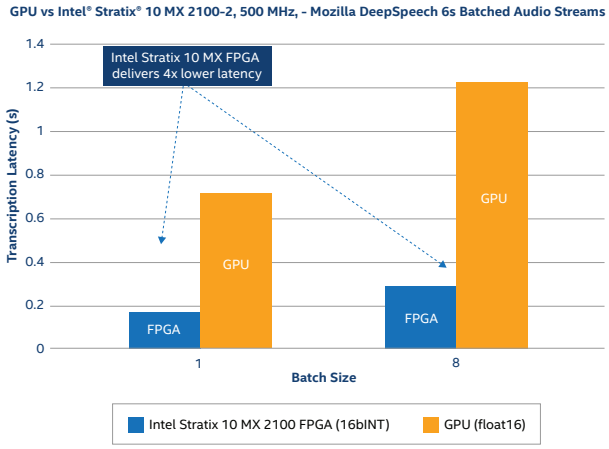

Figure 8. Performance comparison showing 4x drop in latency of FPGA over GPU

Using PIE

Figure 9 shows how the PIE is connected to a host CPU to accelerate the AI inference workloads. Here the application runs on the host and AI computations are off-loaded to the PIE for acceleration.

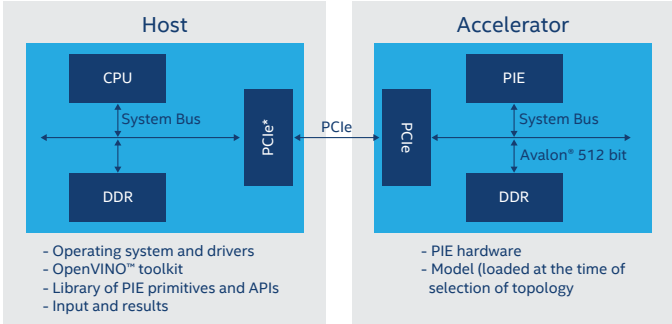

Figure 9. PCIe FPGA card hardware accelerator with PIE

Proposed Software Architecture

The topology to be accelerated, typically developed in AI frameworks such as Caffe or TensorFlow, is expressed as an xml file, which is passed through the OpenVINO model optimizer. The resulting intermediate representation (IR) file is compiled into a binary object file for the PIE IP's integrated Nios® II processor and downloaded onto the Intel FPGA with the inference engine running the host application.

Any network can be quickly coded using API's exposed by the PIE software. To further simplify the process, Manjeera is developing an automated tool-chain, which will generate the Nios binary file directly from the output of the model optimizer.

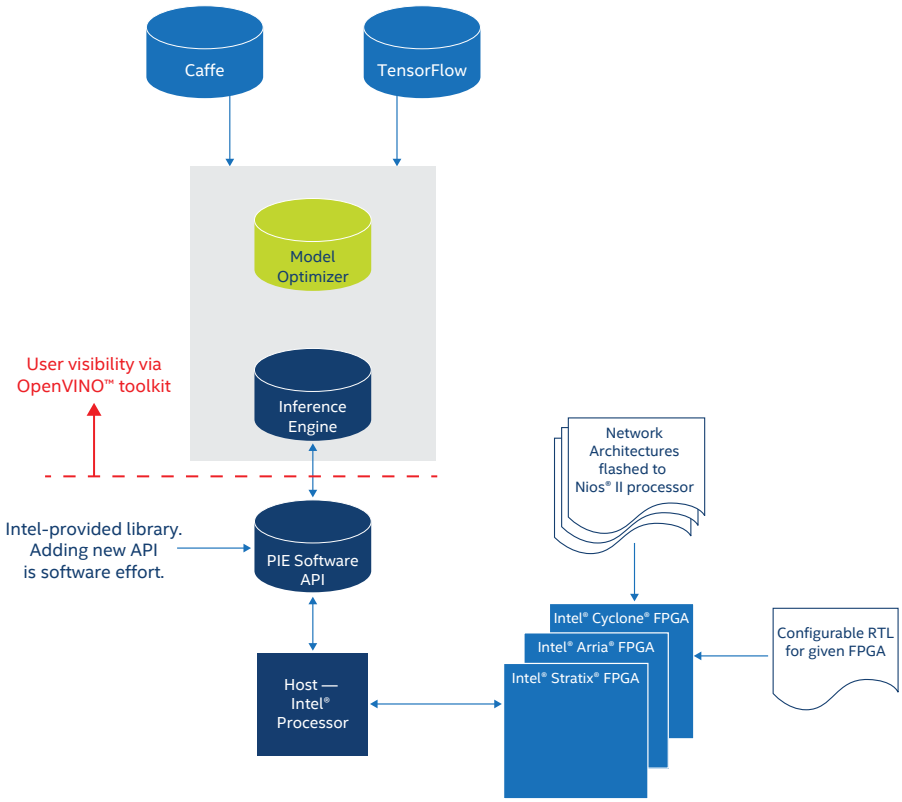

Figure 10. High-level design flow

Summary

Intel FPGAs with their wide and fast datapaths, large on-chip memories and efficient compute make for very compelling AI accelerators. Manjeera's PIE IP core unlocks the goodness of Intel FPGAs, enabling developers to intuitively design and deploy high-performance FPGA AI accelerators for their target use-cases.

Learn More

- To contact Manjeera or to get more information, visit www.manjeerads.com/
- To download and evaluate the PIE IP on your Intel FPGA development kit, contact info@manjeerads.com
- To learn more about Intel FPGAs, visit www.intel.com/fpga-ai

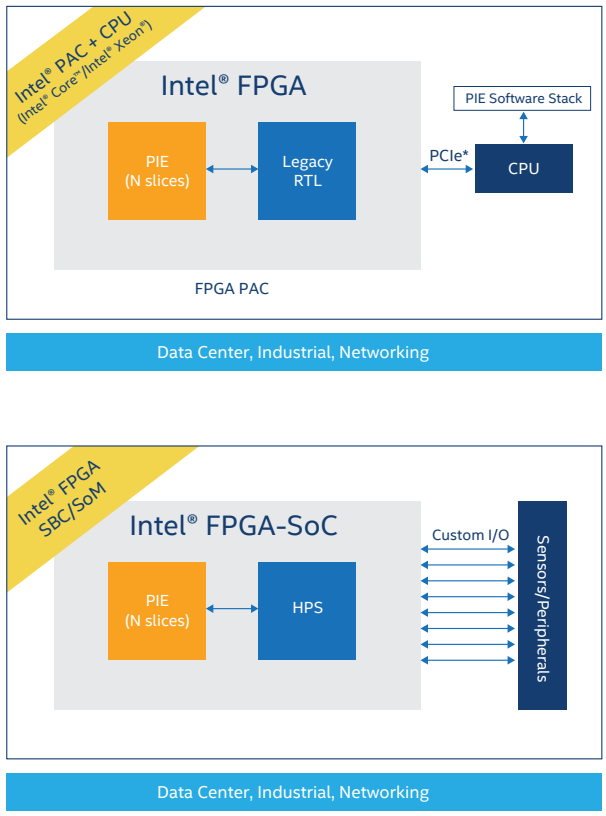

Figure 11. Different implementation methods of PIE

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

For more information go to www.intel.com/benchmarks.

Performance results are based on testing as of [June 2019] and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure.

§ Configurations: [1.] Intel Stratix 10 MX FPGA: PIE 3-core design with dual channel HBM2 interface. 5TOPs/sec (Int16), 205 GBps kernel bandwidth {16 of 16 HBM2 'Bottom' tile pseudochannels}, 38 GBps data bandwidth {3 of 16 HBM2 'Top' tile pseudochannels}

[2.] GPU: ~5.5TFLOPs/sec (16b float), 192 GBps, 810MHz/192 GBps; TensorFlow1.8.0, CUDA 9.0.176, CuDNN 7.0.5

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at [intel.com].

† Tests measure performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

SS-1108-1.0