WHITE PAPER

HPC
AI Inference

Accelerating Memory Bound AI Inference Workloads using Intel Stratix 10 MX Devices with Samsung HBM2

SAMSUNG

Authors

Ramakrishna Madhava
Hardware Engineer
Intel® Corporation

Manish Deo
Senior Product Marketing Manager
Intel Programmable Solutions Group

Martin S. Won
Senior Member of Technical Staff
Intel Network and Custom Logic Group

Table of Contents

Introduction 1
Recurrent Neural Networks: Applications and Challenges 1
Intel Stratix 10 MX Device Architecture. 2
Building an AI Accelerator with Intel Stratix 10 MX Devices. 3
PIE – A High Performance AI Inference Accelerator for Intel FPGAs 3
Summary 5
Where to get more information. 5

Introduction

Current-generation artificial intelligence (AI) systems are increasingly constrained by conventional memory solutions that impose limits on available memory bandwidth. This white paper describes an AI hardware accelerator intellectual property (IP) that leverages the Intel® Stratix® 10 MX FPGA to overcome memory-bandwidth limitations and achieve industry-leading performance.

Recurrent Neural Networks: Applications and Challenges

Recurrent Neural Networks (RNNs) are a class of neural networks used in applications that model the sequential nature of some types of data. In short, RNNs capture the influence of past data on current output. They are used in finance, genome mapping, and speech AI applications, such as Automatic Speech Recognition (ASR), and Natural Language Processing/Understanding (NLP/NLU). Two common traits of almost all these applications is that they are memory intensive and demand very low latency.

An illustration of why RNNs are memory intensive is shown in Figure 1; since each output depends on the previous output, the entire kernel (or weight matrix) is accessed for each 'time-step' of input. With conventional memory technologies, it will be almost impossible to efficiently deliver real-time latency on such workloads.

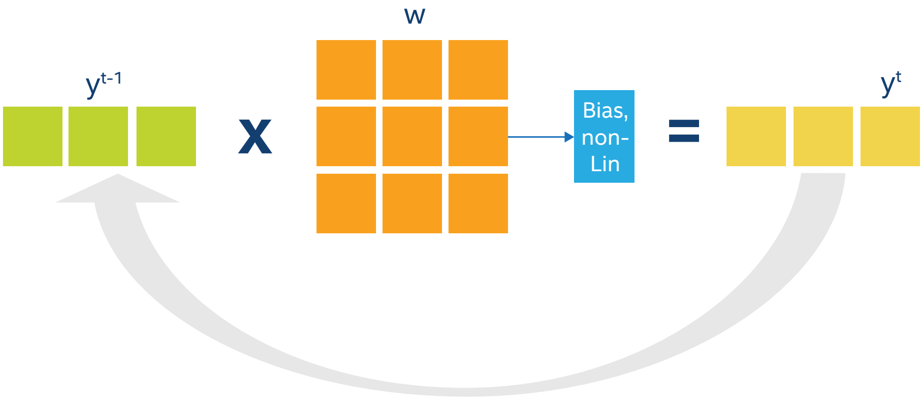

Figure 1. Recurrent Kernels Require High Bandwidth

An FPGA-based AI Inference accelerator using a device like the Intel Stratix 10 MX FPGA can deliver the real-time performance needed by RNN applications. The next sections in this white paper provide an introduction to the Intel Stratix 10 MX FPGA, followed by the description of a novel AI architecture that achieves the lowest latency for RNN-based workloads, while maintaining software flexibility to support a wide range of AI topologies.

Intel Stratix 10 MX Device Architecture

The Intel Stratix 10 MX FPGA is a multi-die, System-in-Package (SiP) device family that combines a high-performance, FPGA fabric, state-of-the-art Intel Embedded Multi-die Interconnect Bridge (EMIB) technology, and Samsung* High Bandwidth Memory 2 (HBM2), all in a single package. Intel Stratix 10 MX devices are specifically designed to meet high-performance system demands where bandwidth is paramount. These devices provide 10X more memory bandwidth with the highest performance per watt compared to conventional memory solutions such as DDR SDRAM†. Figure 2 shows the basic construction of an Intel Stratix 10 MX device.

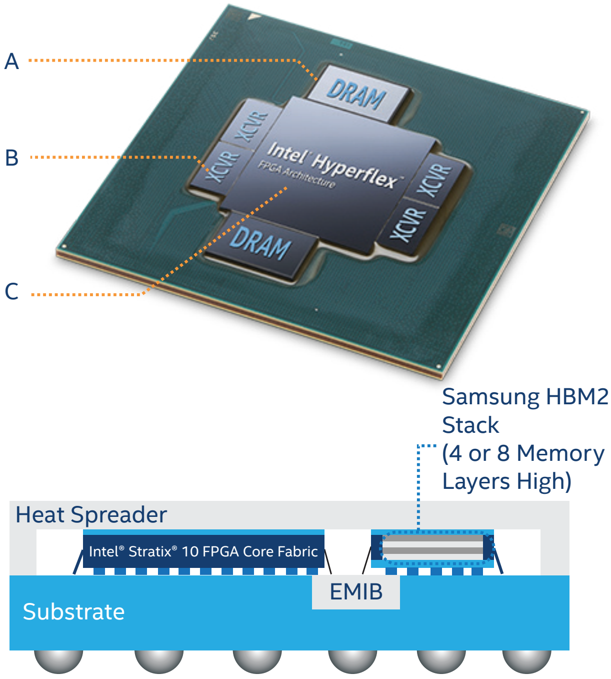

Figure 2. Intel Stratix 10 MX Device

Figure 2 (A), the DRAM boxes, show the Samsung HBM2 memory stacks that the device integrates in package. Each Samsung HBM2 memory stack is either 4 or 8 memory layers high and supports up to 8 physical channels (128 bits each) divided into 16 independent pseudo- channels (64 bits each). Each channel can run at data rates of up to 2 GBps and provide up to 16 GBps of aggregate bandwidth per channel. Figure 3 shows a logical representation of the memory channels and base die.

Figure 2 (B), represent high-performance transceiver tiles that connect to the monolithic core fabric using Intel's EMIB technology.

Figure 2 (C) represents a high-performance monolithic core fabric built using the Intel Hyperflex™ FPGA Architecture. This core fabric can run up to 1 GHz and provide up to 2X performance gains compared to previous generation high-end FPGAs†. The high-performance monolithic core fabric ensures efficient processing of the in-package memory bandwidth and enables a viable system-level solution.

Intel EMIB technology enables effective in-package integration of different tiles alongside a high-performance monolithic core fabric. The EMIB interface supports the required interface data rates between the core fabric and the Samsung HBM2 memory stack tile. This interface is compatible with standard JEDEC and IEEE 1500 specifications. Intel's EMIB provides an elegant way of integrating multiple tiles in a single package. This unprecedented bandwidth enables multiple applications across AI, machine learning, data analytics, image recognition, workload acceleration, 8K video processing, and high-performance computing.

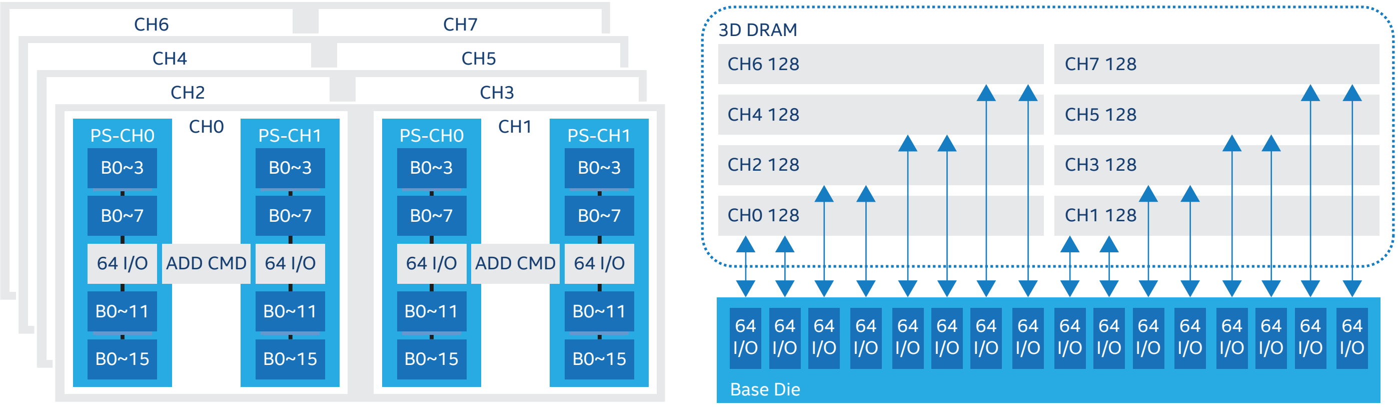

Figure 3. Logical Representation of Samsung HBM2 Device with Four Memory Layers and 16 Pseudo-Channels (PS-CH0, PS-CH1, etc.)

Building an AI Accelerator with Intel Stratix 10 MX Devices

Programming FPGAs to accelerate AI workloads can be a complex task. 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. This approach may require developing a separate design for each new function and topology. Building multiple register transfer level (RTL)-based designs and optimizing them can be time consuming and resource intensive.

The Universal Multifunction Accelerator (UMA) IP developed by Manjeera Digital Systems solves this problem: It is a programmable datapath processor, which provides the high performance of a hardware datapath while retaining the flexibility of software. Implementing UMA in Intel FPGAs results in the Programmable Inference Engine (PIE). PIE is a scalable, plug-and-play AI inference core that is capable of accelerating a wide variety of deep neural network (DNN) topologies. When taking advantage of the considerable digital signal processing (DSP) and memory bandwidth provided by Intel Stratix 10 MX FPGAs, PIE delivers industry-leading performance for low-latency RNN applications like ASR.

PIE – A High Performance AI Inference Accelerator for Intel FPGAs

Figure 4 shows a simplified block diagram of PIE on the Intel Stratix 10 MX FPGA Development Kit (DK-DEV-1SMX-H-A). The development kit's card interfaces to the host via its PCI Express* (PCIe*) interface.

The various functional units instantiated within the Intel Stratix 10 MX FPGA are interconnected via an Avalon® system bus. A Nios® processor serves as the control-flow engine for the PIE and performs the role of executing the workload on the computing core, which consists of a UMA IP core developed by Manjeera Digital Systems, and a Multiply Accumulate (MAC) Unit.

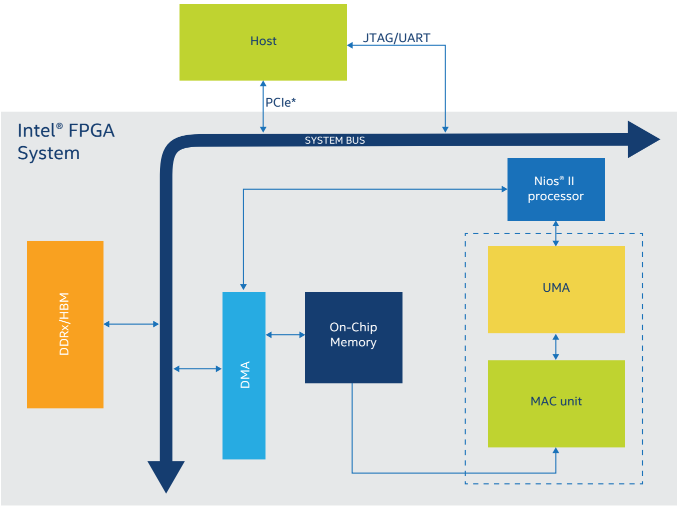

Figure 4. PIE Architecture

PIE connectivity to Samsung HBM2: A diagram showing the connectivity of the on-chip memory to HBM2 stacks is shown in Figure 5. To maximize HBM2 bandwidth utilization, all 16 pseudo-channels of one HBM2 stacks are connected to the on-chip memory, which is partitioned into 16 blocks. With this implementation, a real data transfer rate of 170 GBps on one tile was achieved while accounting for all interface overheads/latencies. This high memory bandwidth has proven key to achieving low-latency performance.

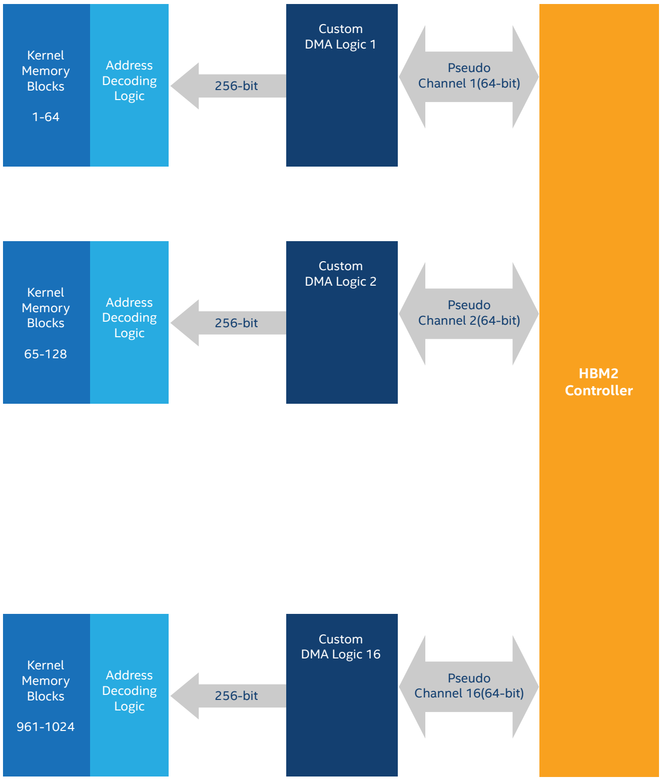

Figure 5. PIE Architecture

The Mozilla* DeepSpeech algorithm for ASR was accelerated using an Intel Stratix 10 MX 2100 FPGA (speed grade -2). The algorithm is an RNN-based topology. A large, memory-intensive, bi-directional long short-term memory (LSTM) layer consumes more than 80% of the overall compute and memory footprint, as illustrated by Fig. 6†.

To highlight the benefit of HBM2 vs. DDR4 SDRAM, a comparison of the ASR latency for a single audio stream of varying lengths on the Mozilla DeepSpeech topology between an Intel Stratix 10 MX device (with integrated HBM2) and an Intel Stratix 10 GX FPGA (with interface to DDR4 SDRAM) is shown in Table 1. Both devices have the same logic and DSP architecture; the main difference is the memory architecture (integrated HBM2 vs. interface to DDR4 SDRAM). Also, both devices have PIE running at 400 MHz and the latency shown is for one audio stream.

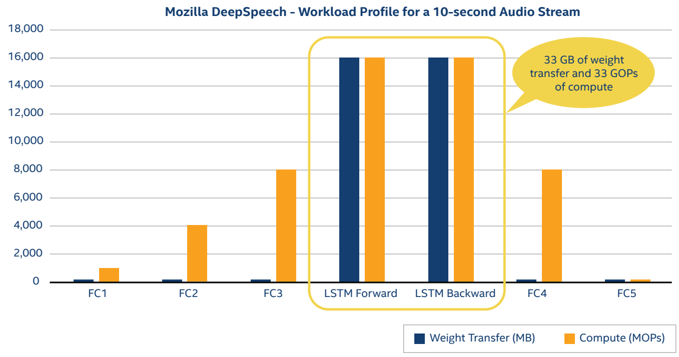

Figure 6. Mozilla DeepSpeech Workload Profile

AUDIO STREAM LENGTH (s)	PROCESSING LATENCY WITH INTEGRATED HBM2 (USING INTEL® STRATIX® 10 MX FPGA) (ms)	PROCESSING LATENCY WITH DDR4 SDRAM(USING INTEL STRATIX 10 GX FPGA) (ms)	LATENCY REDUCTION (%)
1	28	485	94
6	171	2790	94

1. Intel Stratix 10 MX configuration: Intel Stratix 10 MX 2100 FPGA Development Kit Speed Grade -2 with PIE at 400 MHz, INT16 Precision
2. Intel Stratix 10 GX configuration: Intel Stratix 10 GX 280 FPGA Development Kit Speed Grade -2, with PIE at 400 MHz, INT16 precision

Table 1. Comparison of Measured Processing Latency Advantage of HBM vs. DDR for Mozilla DeepSpeech on the PIE IP

Figure 7 is a graph showing the measured latency of the Mozilla DeepSpeech process in the Intel Stratix 10 MX device using PIE vs. an NVIDIA* P4 device, a GPU without HBM memory. The graph shows that the Intel Stratix 10 MX has ~4X lower latency than the NVIDIA P4 device when performing the DeepSpeech inferencing†.

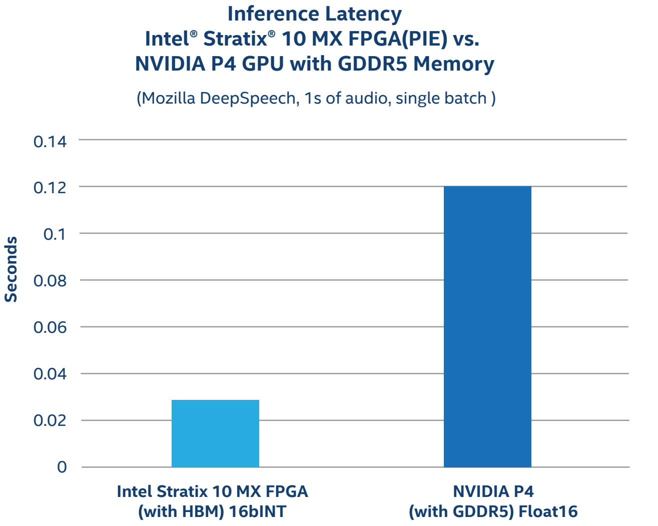

Figure 7. Comparison of Measured Inference Latency Between Intel Stratix 10 MX(PIE) Device and GPU with GDDR5 (NVIDIA P4)

Even when comparing to a GPU with HBM memory, the Intel Stratix 10 MX device delivers a latency advantage. Figure 8 is a graph showing the measured latency of the Mozilla DeepSpeech process in the Intel Stratix 10 MX device using PIE vs. an NVIDIA V100 device, a GPU with integrated HBM memory. The graph shows that the Intel Stratix 10 MX has 20% lower latency than the NVIDIA V100 device when performing the DeepSpeech inferencing†.

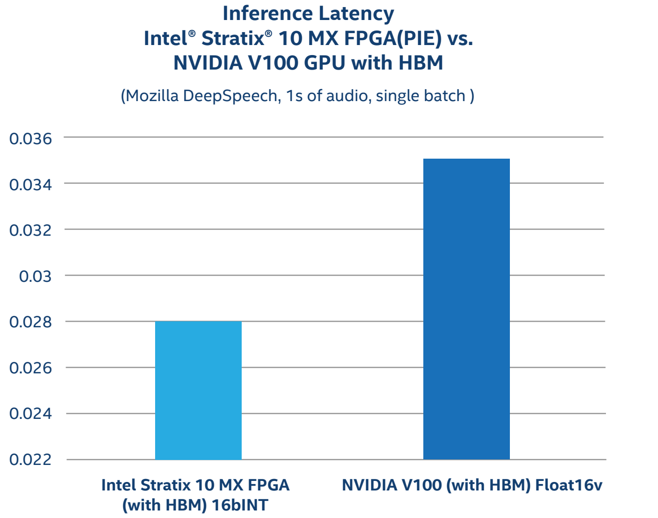

Figure 8. Comparison of Measured Inference Latency Between Intel Stratix 10 MX(PIE) Device and GPU with HBM (NVIDIA V100)

Figure 9 depicts a comparison across multiple batch sizes showing an up to 6X improvement in latency using PIE on the Intel Stratix 10 MX FPGA card versus an NVIDIA P4 GPU with similar compute and bandwidth specifications.

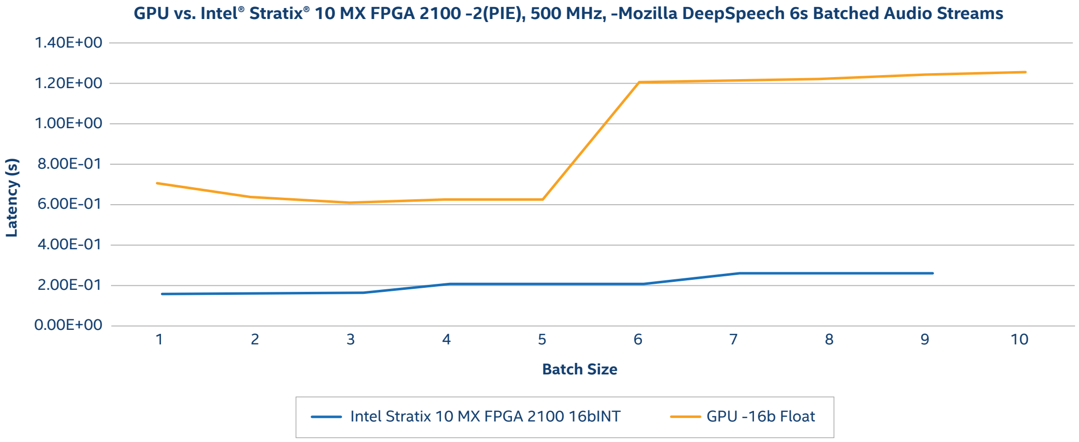

Figure 9. Latency comparison (lower is better) for batched 6s audio stream

The PIE IP also offers the following additional benefits:

- Programmability: The key to the PIE's programmability is the UMA IP, which has a software programmable datapath, unlike conventional fixed datapaths in RTL.
- Scalability: The PIE IP core is modular in nature. At the core of this modularity is the "slice," which is the instantiation of the PIE IP. Designers can parameterize the size of compute and memory based on compute/resource considerations. A common software API permits the same PIE topology to run on many different Intel FPGAs.

Summary

With integrated Samsung HBM2 memory, Intel Stratix 10 MX devices can effectively address memory bandwidth challenges that cannot be solved with traditional DDR-based memory architectures. This paper has demonstrated an example in the RNN space, where the combination of PIE and Intel Stratix 10 MX FPGAs delivers industry-leading low-latency for memory-intensive AI workload with as much as 6X improvement vs. GPUs†. For more details on the PIE IP, refer to the **Solution Brief.**

Where to Get More Information

- 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/content/www/us/en/products/programmable.html

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 May 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure.

§ Configurations:
Intel Stratix 10 MX configuration: Stratix 10 MX 2100 FPGA develpment kit Speed Grade -2 w/PIE@500MHz, INT16 Precision, Quartus Prime version 19.1

NVIDIA P4 configuration: TensorFlow 1.8.0, CUDA 9.0.176, CuDNN 7.0.5
NVIDIA V100 configuration: TensorFlow 1.10.0, 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 www.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.