WP508 (v1.1.2) July 15, 2019

Supercharge Your AI and Database Applications with Xilinx's HBM-Enabled UltraScale+ Devices Featuring Samsung HBM2

High bandwidth memory (HBM) enables more effective data movement and access, realizing the benefits of greater compute capacity afforded by Xilinx UltraScale+ devices.

ABSTRACT

Many application domains that were bottlenecked in the past by insufficient compute power have been greatly advanced by the development of heterogeneous computing accelerators like the Xilinx® UltraScale™ and UltraScale+™ devices and Versal™ ACAPs.

The accelerator solution, however, reveals the next speed-limiting factor: data movement, typically from DDR memory to the compute units. High-bandwidth memory (HBM) helps to ease the data movement and access bottleneck. HBM-enabled devices from Xilinx open up new potentials for the Data Center and raise compute acceleration to the next level**.**

This white paper contains two use cases as examples—deep learning and database acceleration—and explains how a balanced Xilinx compute accelerator system with Samsung's HBM2 delivers high-impact compute acceleration solutions with optimal flexibility, efficiency, and performance.

Background

The rise of heterogeneous computing in recent years broadened the innovations on accelerating compute-intensive workloads in the post-Moore's Law era. Examples of popular heterogeneous compute-accelerated workloads in today's Data Centers are artificial intelligence, live video transcoding, and genomic analytics, to name just a few. Xilinx® UltraScale+™ and next-generation Versal™ devices provide unparalleled adaptability and compute acceleration to modern data-center workloads.

For a long period of time, however, DDR memory architecture did not evolve quickly enough to keep pace with the innovations in compute acceleration. Over the last ten years, the bandwidth capabilities of parallel memory interfaces have improved—but slowly; the maximum supported DDR4 data rate supported in today's FPGAs is still only about 2X what DDR3 provided in 2008. In contrast, FPGA computation capabilities increased about 8X since 2008, and a further magnitude increase can be expected within the next two years as Versal devices with AI cores roll out [Ref 1]. Memory bandwidth and capacity therefore become prime limiting factors in the development of many compute- and memory bandwidth-intensive workloads in the Data Center [Ref 2]. See Figure 1.

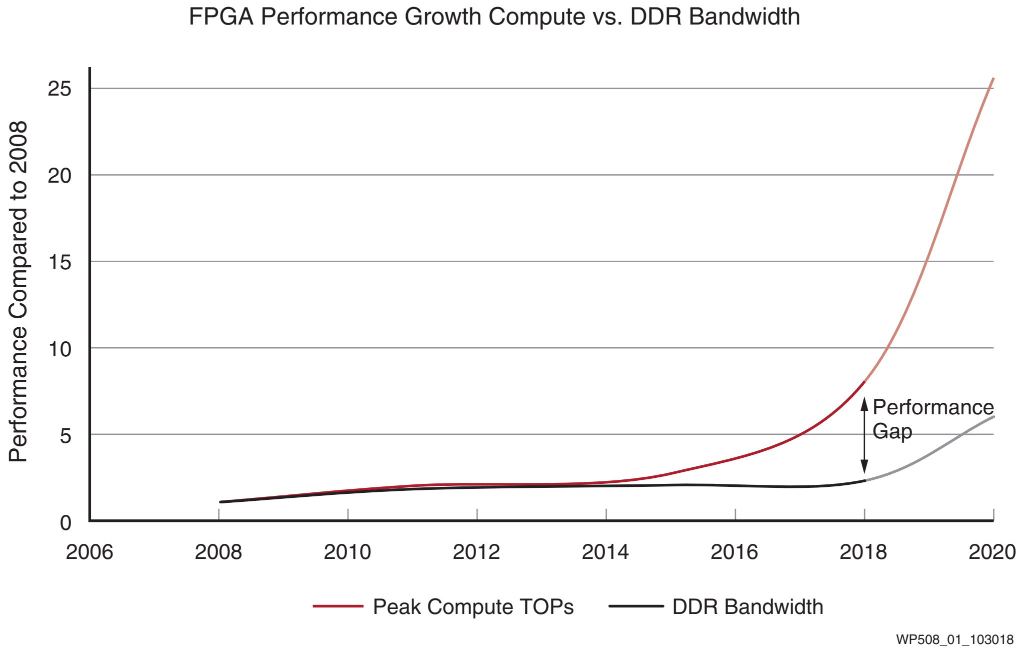


Figure 1: **Compute Capacity Improvement vs. DDR Bandwidth Improvement**

The latest members of the UltraScale+ families feature high-bandwidth memory (HBM) enabled in the same device packaging, finally bridging the gap between memory and computation.

Samsung HBM

HBM is a high-speed, system in package (SiP) memory technology that uses stacks of vertically interconnected DRAM chips and a wide (1024-bit) interface to enable more storage capacity and data bandwidth than memory products using conventional wire bonding-based packages. The Joint Electron Device Engineering Council (JEDEC) adopted the original HBM standard in 2013, and approved its second-generation HBM2 version as an industry standard in January 2016. See Figure 2.

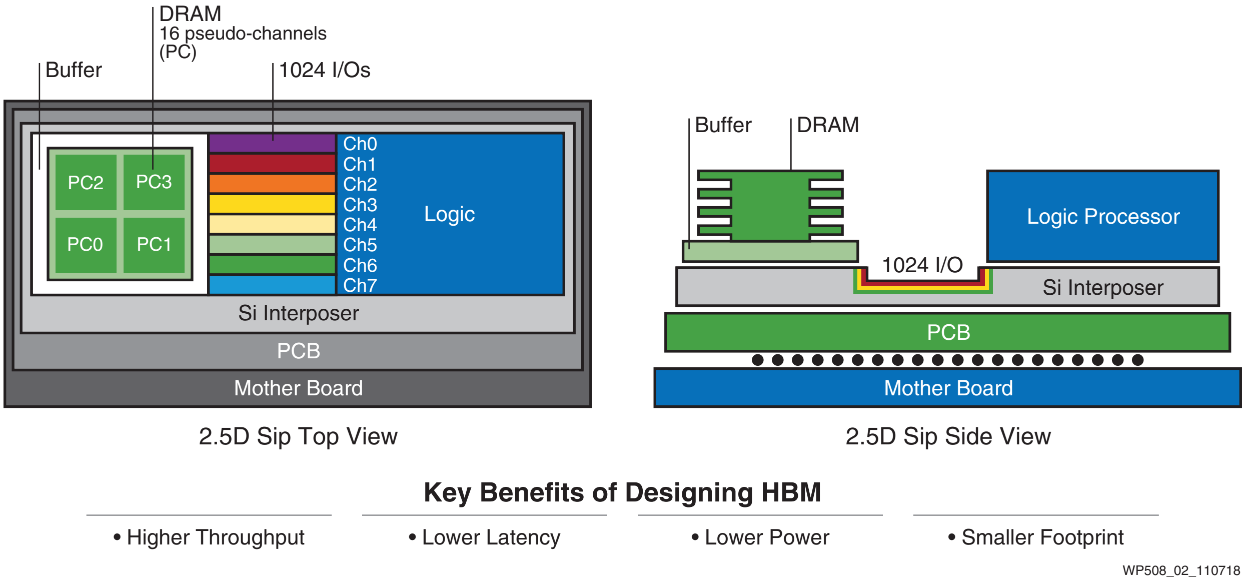
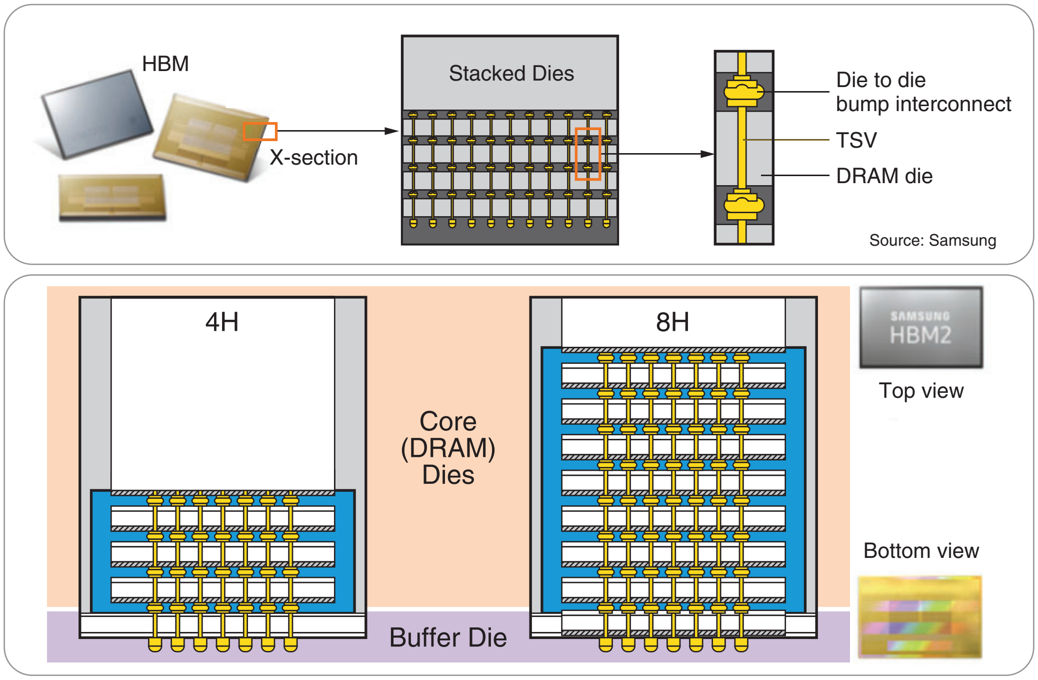


Figure 2: **HBM and 2.5D Structure (Source: Samsung)**

The HBM2 standard allows either four or eight 8Gb DRAM dies to be stacked on top of each other, supported by a 2.5D silicon "interposer," which connects the memory stack with the underlying circuit board. Multi-chip packages that stack some dies (usually memory) but not others (usually a processor) are referred to as 2.5D devices.

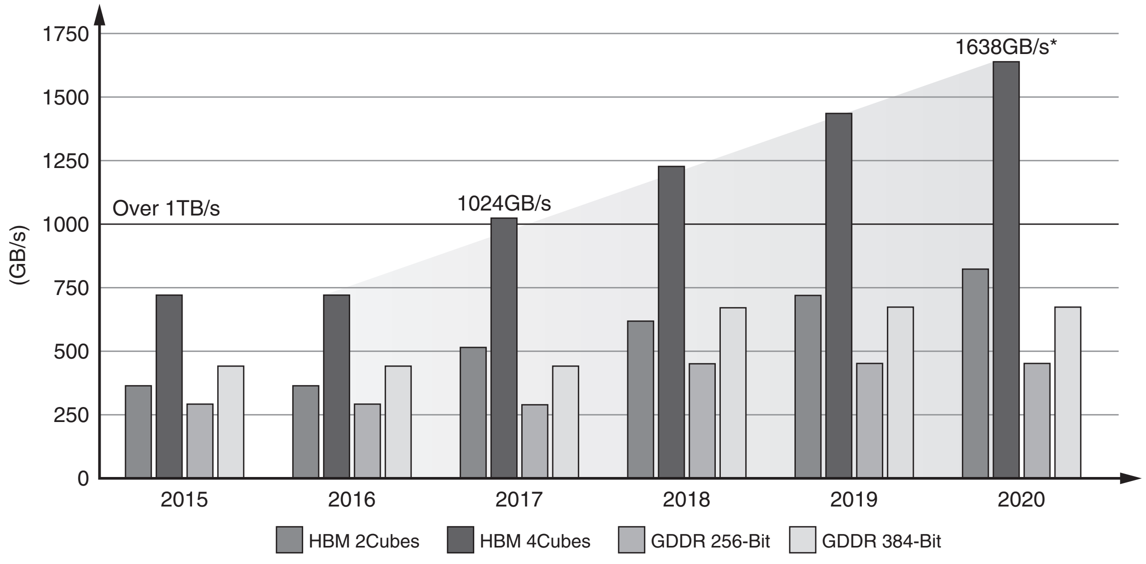
Multiple stacks can be added to the same package. By stacking multiple dies and moving those stacks closer together on a circuit board, the horizontal footprint of HBM memory packages can be greatly reduced, compared with conventional multi-chip DRAM products. HBM technology also improves system performance because signals have a shorter distance to travel between devices. That shorter distance also reduces the energy needed to deliver a given amount of data. See Figure 3.



WP508_11_110718

Figure 3: **HBM Stacked Die Technology (Source: Samsung)**

HBM, with its advanced TSV technology, micro-scale interconnections, and extreme I/O count to increase the memory bandwidth, clearly offers better performance than its nearest competitor, the graphics double data rate (GDDR) memory used in graphics cards. At a component level, a single Samsung HBM cube can offer up to 307GB/s data bandwidth, achieving almost 10X faster data transmission than a GDDR5 chip. At a system level, HBM provides almost 3X better throughput and uses 80% less power compared to a GDDR-based solution while saving valuable circuit real-estate. See Figure 4.



Source: Samsung

* Projection

WP508_12_110218

Figure 4: **HBM vs. GDDR Bandwidth Comparison**

Improved Memory Bandwidth with Virtex UltraScale+ Devices

The Virtex® UltraScale+ FPGAs with HBM [Ref 2] have greatly improved memory bandwidth—up to 460GB/s delivered by two stacks of Samsung HBM2 memory. Pairing one stack or two stacks of HBM2 with various sizes of FPGA logic (up to 2.85 million logic cells) and DSPs (up to 9,024 DSP48E2 slices delivering 28.1 peak INT8 TOPs) allows the user to choose the new HBM-enabled UltraScale+ device family, enabling selection of an optimal combination of compute power and memory bandwidth/capacity for their applications.

In addition to HBM-enabled Virtex UltraScale+ FPGAs, Xilinx is introducing a new addition to its Alveo™ Data Center acceleration card family, Xilinx® Alveo U280 Data Center Accelerator Card [Ref 3].

The Alveo U280 card is built on the Xilinx 16nm UltraScale+ architecture, and features 8GB of Samsung High Bandwidth Memory (HBM2) to provide the highest level of acceleration for database search and analytics, machine learning inference, and other memory-bound applications. The U280 acceleration card includes CCIX support to leverage existing server interconnect infrastructure for high bandwidth, low latency cache coherent shared memory access with upcoming CCIX-enabled processors.

Many memory-bottlenecked applications can benefit from HBM-enabled UltraScale+ devices. In this white paper, case studies in deep neural networks and database acceleration are presented as examples, showing the benefits of HBM-enabled UltraScale+ devices.

Case Study 1

Accelerating AI for Language Translation

Various cloud applications are now providing automated real-time language translation services that can translate sentences between two languages (known as machine translation) using neural network-based machine learning methods.[1]

The encoder-decoder architecture powers today's commercial automated translation services. While using the machine to perform this translation task, words of both languages are represented as high-dimensional vectors through a process known as word embedding; thus, the relationships among words can be quantitatively modeled and reflected by vectors. Structures such as recurrent neural networks, convolutional neural networks, and attention-based models are used to perform encoding and decoding functions.

[1] Amazon Translate:
https://aws.amazon.com/blogs/aws/introducing-amazon-translate-real-time-text-language-translation/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+AmazonWebServicesBlog+%28Amazon+Web+Services+Blog%29.

Microsoft Azure: https://www.microsoft.com/en-us/translator/business/machine-translation/.

Google Machine Translation Service:
https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+blogspot/gJZg+(Google+AI+Blog).

Recent research showed that state-of-the-art accuracy in language translation can be achieved using only attention-based networks [Ref 4]. The attention mechanism described in the research paper—scaled dot-product attention—is constructed by two matrix multiplications and other functions (scale, mask and softmax).

Multiples of parallel scaled dot-product attention with different projections of inputs are used to form a multi-head attention structure. This structure, along with feed-forward networks, is used to construct the decoder and encoder for the overall language translation model [Ref 4]. See Figure 5.

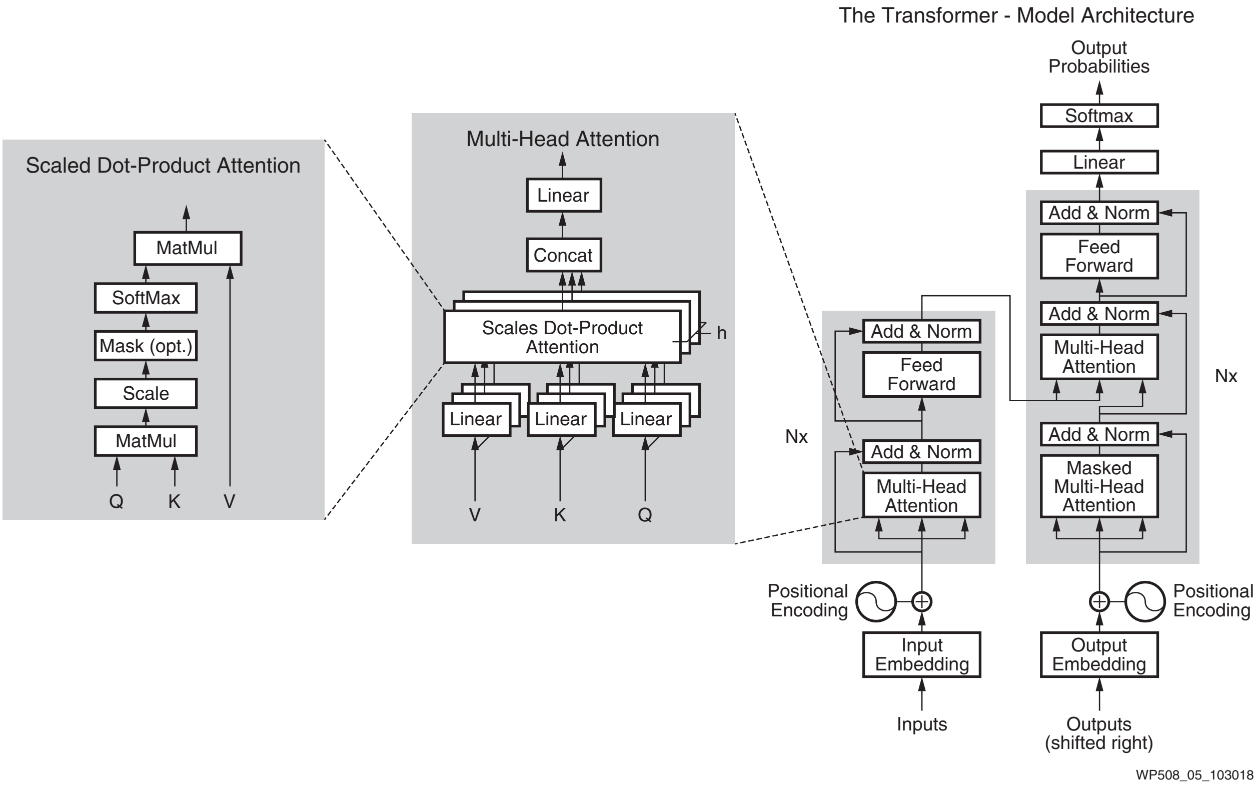
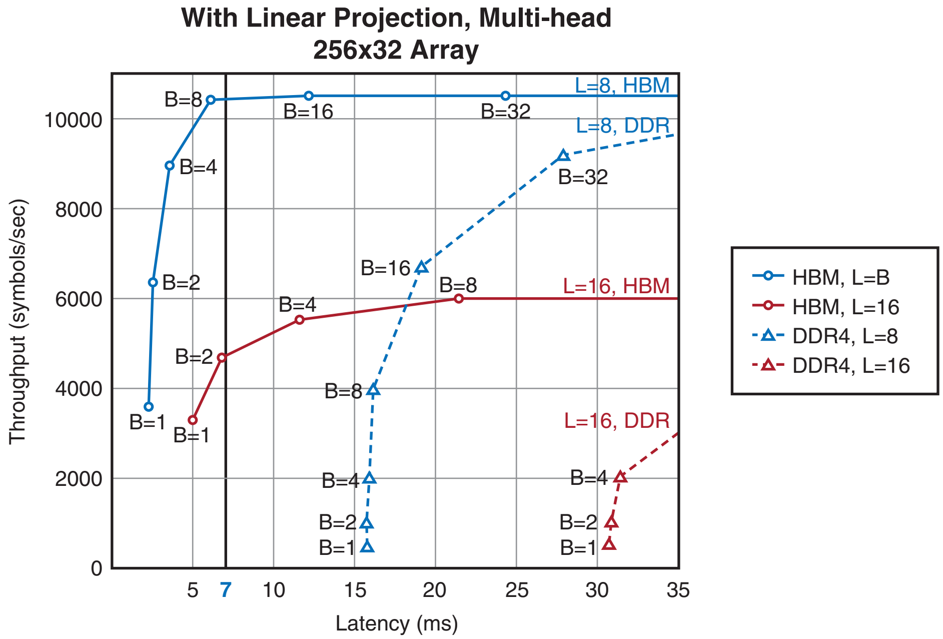


Figure 5: **Transformer: The Attention-Based Neural Network Model for Language Translation [Ref 4]**

As shown in Figure 5, the major compute intensity of the attention-based language translation model comes from computing the dot products for the scaled dot-product attention and feed-forward networks. And for more efficient computations, these dot-products are usually grouped into matrix multiplication computations. However, unlike traditional convolutional neural networks, where weight parameters can be greatly reused across the space to reduce the data movement footprint, this attention-based model barely reuses parameters for the input space, and results in a much higher memory bandwidth requirement to feed the matrix multiplication computations.

An analytical model of the Transformer, the state-of-the-art attention-based neural network used for language translation, has been built and mapped for implementation on UltraScale+ devices. The architecture of the FPGA implementation uses a DSP systolic array to perform matrix multiplication. The intermediate activation data are stored in on-chip URAM to eliminate the frequent data movement between processor and off-chip memory due to activations. HBM or DDR are used to store all the embedding dictionary word embeddings and weight parameters.

English-German translation tasks with various sentence lengths (L) and number of sentences, also known as batch size (B), are analyzed to understand the latency and throughput trade-offs on DDR-attached UltraScale+ devices and HBM-enabled UltraScale+ devices. This study assumes use of a systolic array with 256 16-bit inputs and 32 16-bit outputs running on an UltraScale+ device at 710MHz for both DDR and HBM implementations. A detailed throughput vs. latency graph is shown in Figure 6.



WP508_06_120718

Figure 6: **Performance Analysis of Language Translation Using UltraScale+ Devices with HBM and DDR4**

The lowest latency of a 4-channel implementation is about 15.7ms for a sentence length of 8, with a throughput of 508 symbols/sec. One DDR channel is used to access word embedding data, and the remaining three DDR channels are used for weight upload. With HBM-enabled devices, the minimum latency for a sentence length of 8 is 2.2ms, which is over 7X lower than with a DDR interface. While the DDR interface cannot achieve a latency below 7ms for either sentence length, the HBM-enabled device achieves a latency of 6.1ms for a sentence length of 8 at a throughput of 10,419 symbols/sec and a latency of 6.8ms for a sentence length of 16 at a throughput of 4,682 symbols/sec.

Accelerating AI with General Matrix Operation Library

Like the machine translation example discussed previously, the major computation in almost all the modern deep neural networks (deep learning) are in forms of matrix multiplication.

In addition to machine translation, other typical deep-learning applications widely deployed in Data Centers are image/video analysis, search ranking systems used in web searches, recommendation systems used in ad placement, content/feed recommendations, speech recognitions, and natural language processing.

To better support a wider variety of deep learning applications, Xilinx developed the General Matrix Operation (GEMX) library for accelerating matrix operations on Xilinx devices supported by the

SDAccel™ development environment. This library includes three components: an engine library, a host code compiler, and an application- or system- building environment. The engine library consists of a set of C++ templates with BLAS-like function interfaces for realizing matrix operations on FPGAs. The host code compiler compiles the host code matrix function calls into a sequence of instructions for triggering matrix operations on FPGAs. The building environment uses a GNU make flow to automate the FPGA and host code image generation process. It also allows users to configure different aspects of the system—e.g., FPGA platform, number of engines implemented in the FPGA image, etc. For detailed information about the GEMX engine design, refer to https://github.com/Xilinx/gemx.

While both of the input matrices of the GEMX engine are feeds from DDR memory, the throughput of GEMX is determined by the bandwidth of the DDR interface. The following analysis compares the performance of GEMX using DDR4-attached UltraScale+ devices versus GEMX using HBM-enabled UltraScale+ devices. This analysis models fully utilized memory bandwidths and assumes a matrix of 32x32x128 is used as input to GEMX. The result (illustrated in Figure 7) shows that the HBM-enabled device can boost GEMX performance about 3.6X compared to using four channels of DDR.

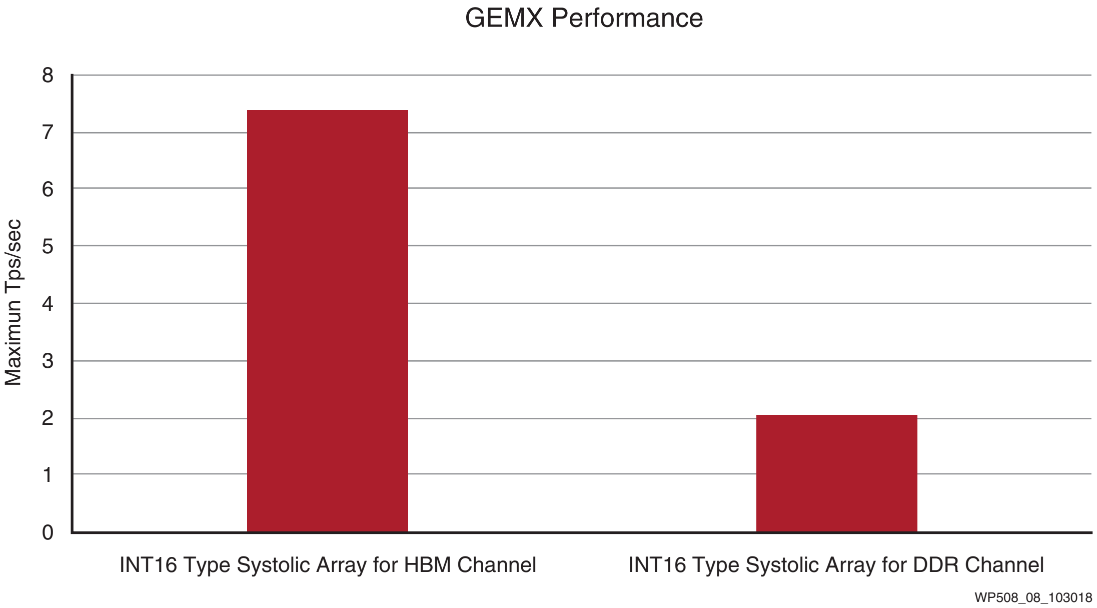


Figure 7: **GEMX Performance**

Case Study 2

Boosting Database Performance

Modern database systems must handle large amounts of unstructured data with low latency. The latest x86 solution can support large internal memory (cache) and even parallel processing units (AVX) inside the processor. However, CPU-based implementations have yet to meet growing performance requirements, and cost effectiveness has not been attained in most cases. GPU-based implementation is also growing in popularity, but it requires large batch operations and power to achieve high performance.

Xilinx devices with high-capacity Samsung HBM allow users to choose the desired memory data width, the number of channels, and the amount of memory needed for each kernel. Programmable logic can use parallel processing engines and, paired with high-speed access to the HBM memory, to accelerate key database operations such as Gather, Group, Indirect Gather, Fused Gather Group, Segmented Gather, etc.

Figure 8 shows a typical use case in which the host sends a database operation request to the FPGA base for the query it is receiving; the FPGA compute engine then performs database operations with high-bandwidth, high-speed access to the attached HBM with accelerated performance. The results are returned to the host to finalize the query requests.

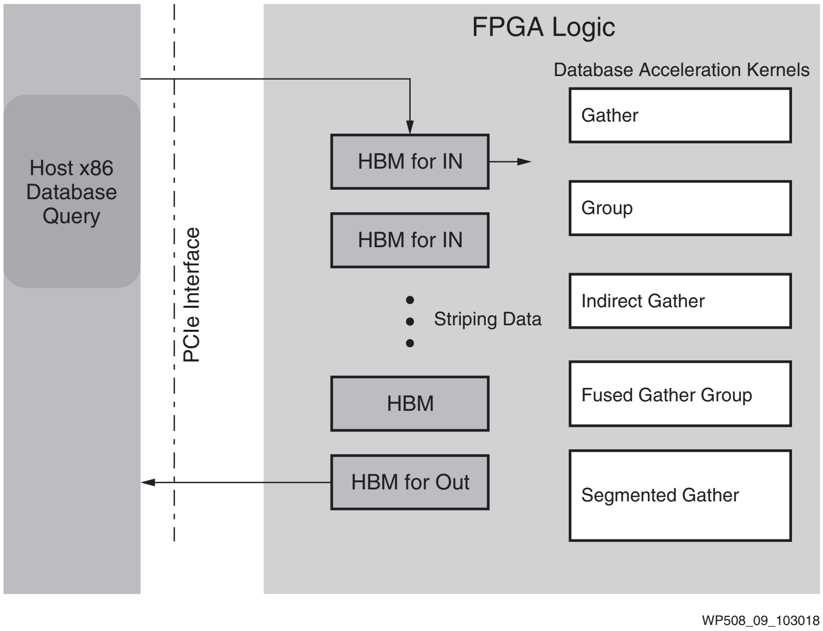


Figure 8: **Example of Database Use Case on an HBM-Enabled UltraScale+ Device**

Typical database operation requires a lot of memory access—mainly random memory read and write operations. To analyze database operation on HBM, one of the database primitive operations called Gather is used to measure performance. The Gather operation involves a large number of random small data accesses. Figure 9 shows the results of performance gain by varying number of compute units (parallel processing engines) and number of pseudo-channels for the Gather operation.

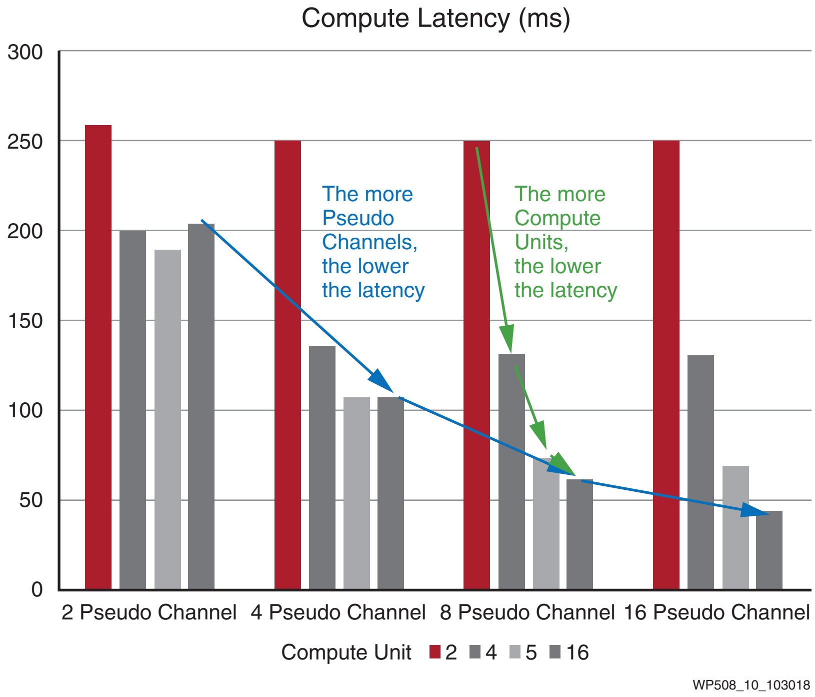


Figure 9: **Database Operation Latency Analysis**

As illustrated in Figure 9, adding more compute units and/or pseudo channels improves latency. For instance, when eight pseudo channels are used, adding more compute units lowers the latency (the green line). When the same 16 compute units are used, adding more pseudo channels also lowers the latency (yellow line).

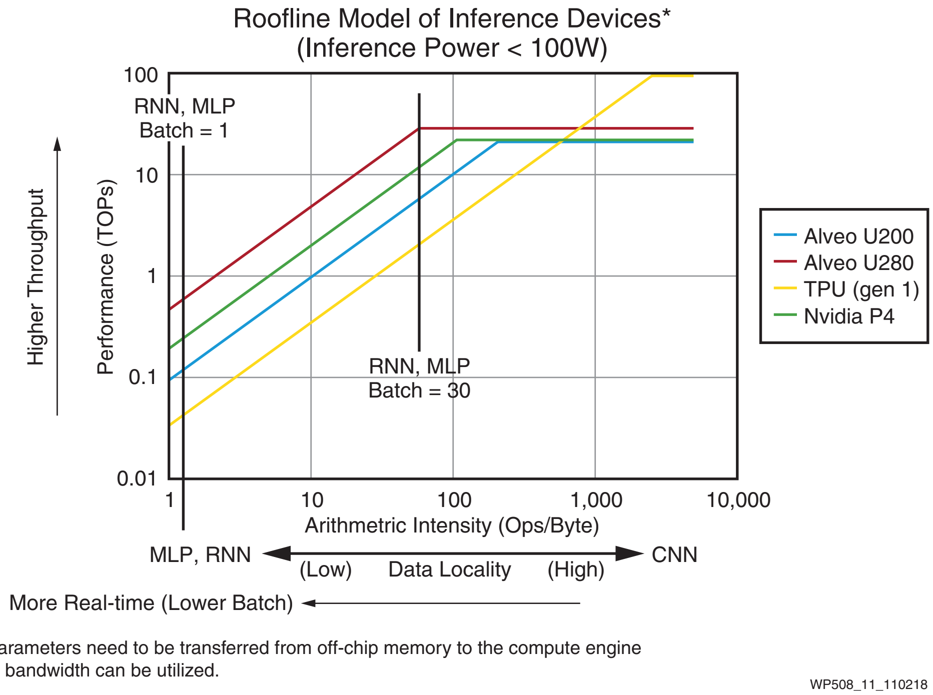
Discussion

High Bandwidth Unlocks Compute Capacities for Real-Time Applications

In Case Study 1, examples of neural networks that require high memory bandwidth are described.

RNN (recurrent neural network) or MLP (multi-layer perceptron) require only two operations (multiplication and addition) to consume one byte of parameters (assume INT8 operations). This is in stark contrast to a CNN (convolutional neural network), which theoretically allows wide reuse of weight parameters across an entire 1D, 2D, or even 3D space in each layer and results in a very high arithmetic intensity (the arithmetic operations needed to consume a byte of data). In this case, the data rate or the bandwidth of memory that feeds the data to compute kernels should match the compute capacity to reach the highest compute efficiency.

Today's best compute accelerators can easily deliver tens to hundreds of INT8 TOPs, such as Xilinx's UltraScale+ devices and Versal ACAPs. In theory, an exactly matching amount of memory bandwidth should be at least tens of terabytes/sec, which is only feasible in very limited on-chip SRAMs. The ideal approach to ease this memory bandwidth requirement is to use batching. While batching tasks can be carried out together, the same parameters can be shared among a batch of tasks. That increases the arithmetic intensity, thereby reducing the intensity of the data movement per compute operation. The trade-off with batching is an increase in latency, as shown in Figure 6 of Case Study 1. So, for many latency critical tasks, like real-time translation, very high batch numbers might not be an option.



*Assumes that the parameters need to be transferred from off-chip memory to the compute engine and that full memory bandwidth can be utilized.

Figure 10: **Roofline Model of Xilinx UltraScale+ and Other Devices**

Figure 10 shows a theoretical roofline model for various devices under 100W power that can target machine learning inference workloads. In this figure, a memory roofline indicates the theoretical peak performance of a device at each given arithmetic intensity point. It is clear that Xilinx's HBM-enabled Alveo U280 accelerator card is capable of delivering the highest performance among all these inference devices for real-time (low-batch, low-latency) RNN and MLP use cases.

Flexible Parallel Memory Channel Access for More Effective Use of Memory and Compute Resources

Thanks to the flexible addressing included in the HBM memory controller [Ref 2], one of the unique advantages of HBM-enabled UltraScale+ device is the flexible and parallel usage of HBM channels. The access and use of HBM channels is completely reconfigurable and reprogrammable, and can be easily accessed by the FPGA logic. Extending the database use case described in Case Study 2, as shown in Figure 11, the user can strip the original database file into the multiple HBM channels, and let parallel-processing compute units perform different database operations to access different parts of the database independently and in parallel, without need for any synchronization. This feature could help to improve operation latency and effective usage of HBM in real-time use cases, as all the compute units can react to the database queries in real time with no need for batching or synchronization.

In the machine translation example of Case Study 1, this feature is also utilized by separating two channels for word embedding access, and the rest of the channels for high-bandwidth weight parameter transfer.

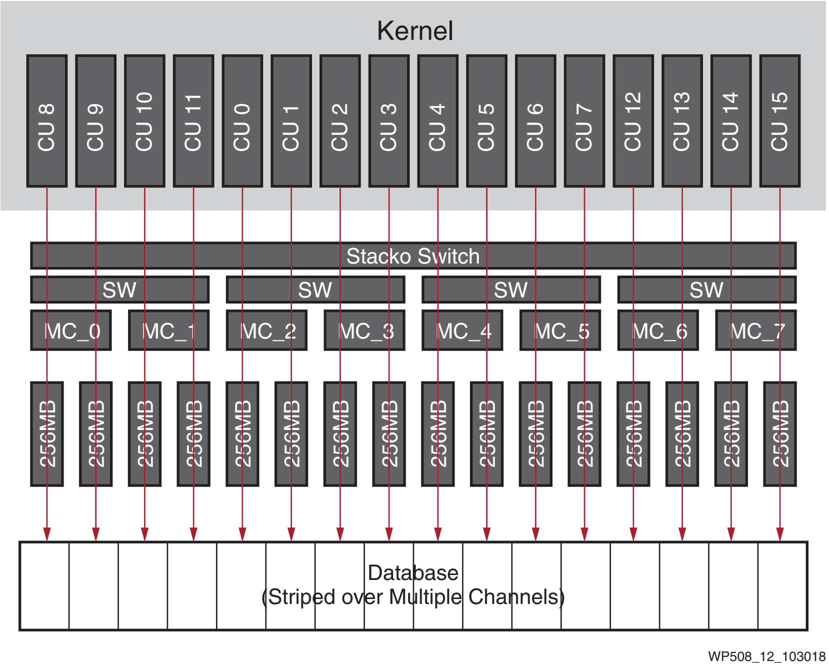


Figure 11: **Flexible Parallel Memory Channel Access**

Conclusion

Although the importance of Samsung HBM in use cases such as AI and database acceleration has been shown, one should note that there are many other Data Center workloads that can also benefit significantly from HBM. For example, a SmartNIC use case was introduced in a Xilinx white paper last year [Ref 2], and Xilinx's partner also published their use case using an HBM-enabled device-acceleration FFT [Ref 5].

With the trends of heterogeneous computing acceleration moving forward in Data Centers together with innovations in computing device technology, there are urgent needs for high performance memory systems like HBM attached close to the compute units.

There are unique values of HBM enabled UltraScale+ devices. Taking deep learning and database acceleration as examples, the HBM-enabled UltraScale+ devices from Xilinx open new potentials and raise computing acceleration to the next level.

References

1. Xilinx White Paper WP505, *Versal: The First Adaptive Compute Acceleration Platform (ACAP).*
2. Xilinx White Paper WP485, *Virtex UltraScale+ HBM FPGA: A Revolutionary Increase in Memory Performance.*
3. Xilinx *Alveo Product Selection Guide*.
4. Vaswani, A., et al., *Attention Is All You Need*, v0.4, 18 July 2018. Google, Inc. Retrieved from Cornell University Library, arXiv.org.
5. McCormick, A. C., *Breaking Memory Bandwidth Barriers Using High Bandwidth Memory FPGA*, v1.0, 15 May 2018. Retrieved from Alpha Data Parallel Systems Ltd., ftp.alpha-data.com.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
07/15/2019	1.1.2	Editorial update.
02/01/2019	1.1.1	Typographical edit.
01/15/2019	1.1	Updated Improved Memory Bandwidth with Virtex UltraScale+ Devices, Accelerating AI for Language Translation, and Figure 6.
11/12/2018	1.0	Initial Xilinx release.

Disclaimer

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available “AS IS” and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx’s limited warranty, please refer to Xilinx’s Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx’s Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos.

Automotive Applications Disclaimer

AUTOMOTIVE PRODUCTS (IDENTIFIED AS “XA” IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE (“SAFETY APPLICATION”) UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD (“SAFETY DESIGN”). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.