White Paper

Agilex™ FPGAs
5G Beamforming Solutions

Proven 5G Beamforming System Level Benefits: Agilex™ 7 SoC FPGAs vs. Competing ACAP

Agilex™ 7 SoC FPGAs deliver 25% power savings at logic parity or 60% more fabric functionality at power parity compared to AMD's Versal* ACAP FPGAs.

Authors

Michael Wu
FPGA Core Architect

Maanasa Mohanambal Sathianarayanan
Technical Marketing Manager

Ilya Ganusov
Fellow

Bret Gustafson
Strategic Business Developer

Tolga Ayhan
System Solutions Engineer
Altera Corporation

Table of Contents

Introduction

Today's FPGAs are very different from their predecessors. For example, in addition to high-capacity, high-performance programmable fabric, large numbers of variable-precision digital signal processing (DSP) blocks, and large amounts of on-chip static random access memory (SRAM), Altera's Agilex™ SoC FPGAs contain a variety of hardened functions that are implemented in silicon. These hardened functions include a network-on-chip (NoC), a secure device manager (SDM), a hard processor subsystem (HPS) featuring a cluster of high-performance processors, and transceiver (XCVR) chiplets or tiles. Similarly, AMD's Versal* Adaptive Compute Acceleration Platform (ACAP) devices are FPGAs with programmable fabric augmented with a NoC, processors, and other hardened functions. One of the ACAP-hardened functions is known as an Artificial Intelligence Engine (AIE).

Agilex 7 SoC FPGAs are built on Altera's 10 nm and 7 nm process nodes and feature next-generation core fabric architecture, which delivers over 50% higher performance or 40% lower power, on average, over the prior generation Stratix® 10 FPGAs [1,2]. These improvements were achieved through a significant overhaul of the fabric architecture to enable performance and power scaling beyond the 7 nm process node as well as tight design technology co-optimization (DTCO) with Altera's process node recipe and software-hardware co-design with the Quartus® Prime Pro Edition Design Software.

AMD's Versal FPGAs (officially named ACAPs rather than FPGAs) are built on TSMC's 7 nm process node. AMD's marketing material highlights that the Versal fabric architecture experienced issues scaling to the 7 nm process node, and this served as a motivation to introduce more acceleration intellectual property (IP) cores, such as artificial intelligence (AI) engines for DSP and AI workloads [3,4]. AMD also claimed that AI engines deliver 2.14X higher performance per watt as compared to the Agilex FPGA when performing complex multiply-accumulate (MAC) operations as part of a massive multiple input, multiple output (mMIMO) 5G wireless beamforming application [4]. However, the justification and details of FPGA resources used behind these projections were not disclosed to verify the veracity of the claims.

The goal of this paper is to propose a clear and reproducible methodology for evaluating the power efficiency of 5G beamformer workloads on Agilex FPGAs vs AMD's Versal ACAP devices and present the power projections generated on the latest IP and their respective power estimation tools from both vendors. First, we present a high-level introduction to 5G mMIMO beamforming technology. Then, we describe our methodology for comparing a real-world, 64-antenna implementation on an Agilex SoC FPGA relative to an equivalent AMD Versal ACAP

device. Finally, we show that when a full 5G beamformer workload is evaluated, Agilex 7 FPGAs deliver 25% lower power than comparable Versal ACAP devices utilizing AI engines for the same workload.

5G mMIMO Beamformer Solution Overview

Traditional cellular radio transmitters use a single antenna to transmit their power omnidirectionally – that is, equally in all directions – with the result that most of the power is wasted in locations where pieces of user equipment (UE), such as mobile phones, are not present. The use of mMIMO in 5G New Radio (NR) radio access networks (RANs) allows radio towers, or base stations, to increase their capacity by employing multiple antennas and the ability to transmit and receive data simultaneously.

Furthermore, in the case of signals being transmitted from the radio tower by using channel state information (CSI) received from the UEs, it's possible to control the timing and phase difference of the signals from the antennas in such a way as to achieve constructive interference (i.e., boosting the signal) in a desired direction along with destructive interference (i.e., rendering the signal undetectable) in other directions. This process, known as beamforming, allows the tower to direct multiple beams simultaneously at multiple UEs. Beamforming techniques can also be employed to isolate signals being received from multiple users simultaneously (Figure 1).

At the heart of a 5G beamformer solution are three computationally intensive functions that we will refer to as the multiple input, multiple output (MIMO) beamformer module, the Physical Random Access Channel (PRACH) module, and the Fast Fourier transform (FFT)/Inverse FFT (IFFT) module (Figure 2). In practice, these modules are supported by other functions, such as interleavers, de-interleavers, cyclic prefix addition/subtraction, sounding reference signal (SRS) processing, and more.

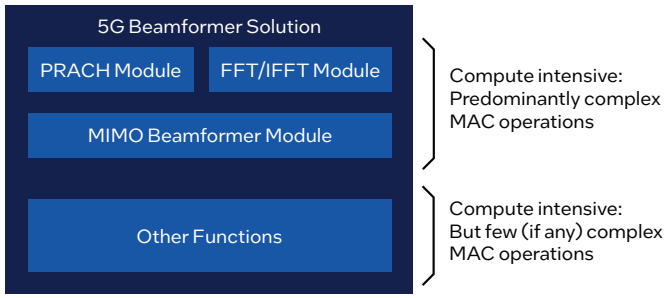


Figure 2. A high-level view of the main functions forming a 5G beamforming solution.

In reality, most functions forming a 5G beamforming solution require hardware acceleration. For this discussion, the main differentiator is that the MIMO beamformer, PRACH, and FFT/IFFT modules predominantly perform complex MAC operations.

One point to note about the 5G beamformer example in AMD's System-Level Benefits of the Versal Platform white paper is that they based their results on a 200 MHz bandwidth implementation. The benchmark in this paper is based on a real-world configuration with a 100 MHz bandwidth and a 30 kHz subcarrier spacing commonly deployed by carriers. Two instantiations of this 100 MHz implementation can be aggregated to deploy a 200 MHz beamforming solution.

The mMIMO Beamformer Module

In the context of a 5G cellular network, from a user's perspective, downlink (DL) refers to the communication link from the radio tower to the user's device (for example, a cell phone) and, uplink (UL) refers to the communication channel from the user's device to the radio tower.

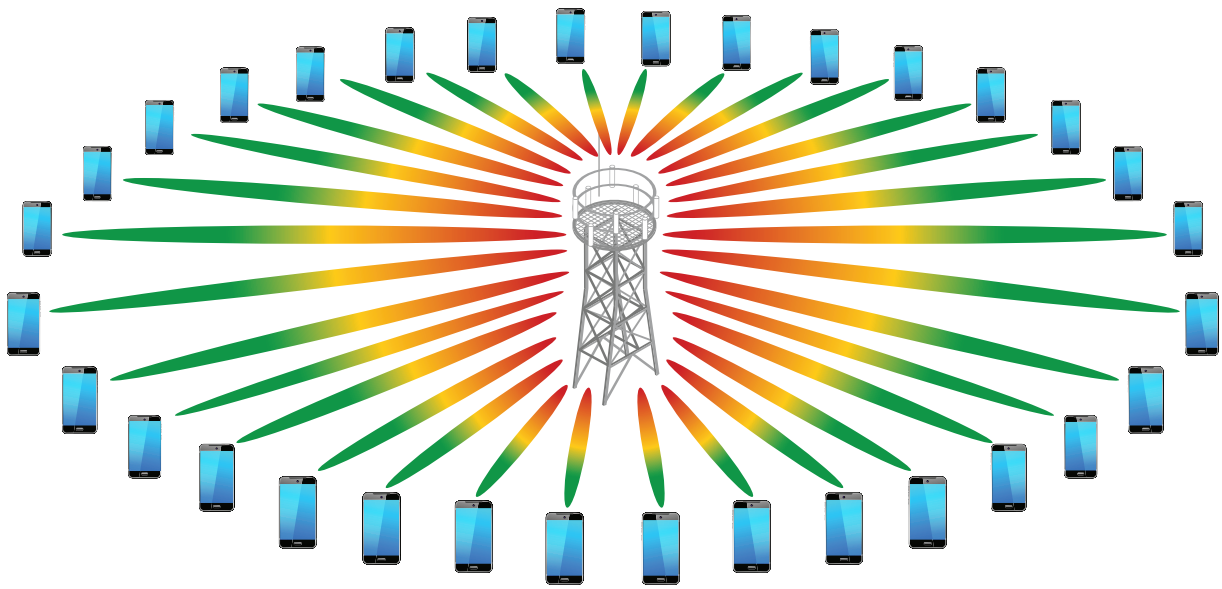


Figure 1. mMIMO allows a base station to communicate with many more users simultaneously.

The main task of the mMIMO beamformer module is to calculate the signals transmitted from the base station to users in the DL and reconstruct the transmitted signals from users to the base station in the UL. In the DL, the beamformer module constructs the transmitted signals, or beams, by multiplying beam weights with the data signals. In the UL, the beamformer module reconstructs the transmitted data from users by multiplying beam weights with the received signals from the antennas. A mMIMO beamformer is computationally expensive and requires many MAC operations.

In 5G, resource block (RB) refers to the smallest resource allocation unit in the form of a time-frequency resource that can be allocated to a user or service. RBs are used for both the DL and UL paths.

The degree of mMIMO is indicated by the number of antennas and layers. We use a commonly deployed 64-antenna, 16-layer mMIMO configuration for this benchmark. Each component is programmed to perform matrix multiplication per RB and to process 273 RBs every ~35.7 µs for the DL and UL paths.

The PRACH Module

The PRACH module is employed in the UL channel to facilitate the UE establishing a connection with the base station. This is used for initial access and to restore access when the UE loses its UL synchronization.

PRACH preamble processing is computationally expensive and requires finite impulse response (FIR) filters. In this benchmark, the PRACH module processes all 64 antennas, performing digital down conversion (DDC) on 128 mega-sample-per-second (Msps) signals for each antenna.

The 4K FFT/IFFT Module

An efficient modulation technique in many modern wireless communication systems, including 5G NR, is orthogonal frequency division multiplexing (OFDM). In the case of OFDM, the signals used inside the MIMO beamformer are represented in the frequency domain. However, the signals transmitted and received at the antenna are in the time domain.

In the case of this benchmark, the 4K FFT/IFFT module is used to convert symbols from the time domain to the frequency domain on the UL path and from the frequency domain to the time domain on the DL path.

The 4K FFT/IFFT module is computationally expensive and requires many MAC operations. In this benchmark, the module processes 64 4K FFT/IFFTs every ~35.7 µs.

5G mMIMO Beamformer Benchmark

The benchmark has been carefully designed to closely match the experimental setup documented in the aforementioned AMD white paper. As previously discussed, this paper's benchmark is based on a commonly deployed real-world configuration with a 100 MHz carrier bandwidth and a 30 kHz subcarrier spacing. It's important to note that this 100 MHz value refers to the beamformer system bandwidth; the clock frequencies associated with the various functions forming the beamformer may be much higher.

The primary goal of this analysis is to provide a fair comparison between an Agilex SoC FPGA and an AMD Versal ACAP device equipped with an AIE when implementing the 5G beamformer solution discussed above. As part of this, we have done our best to maximize the performance of both devices.

Experimental Setup

The Quartus Prime Design Software[2] version 23.1 and Xilinx Vivado* Design Suite[3] version 2023.2 is used in this evaluation, along with their respective power estimator tools – the FPGA Power and Thermal Calculator (PTC) and AMD's Power Design Manager (PDM) version 2023.2. The CAD flows of these tools can be customized to trade off design performance, logic resource consumption, compile time, and memory utilization. The customized settings that produce the best results for one design are not necessarily the best for others. As such, the analysis was done using the default compilation settings for both tools.

To conduct these experiments, we use an Agilex 7 device with a similar speed grade and comparable logic density to AMD's Versal VC1902 device. More specifically, the devices used in our experiments are as follows:

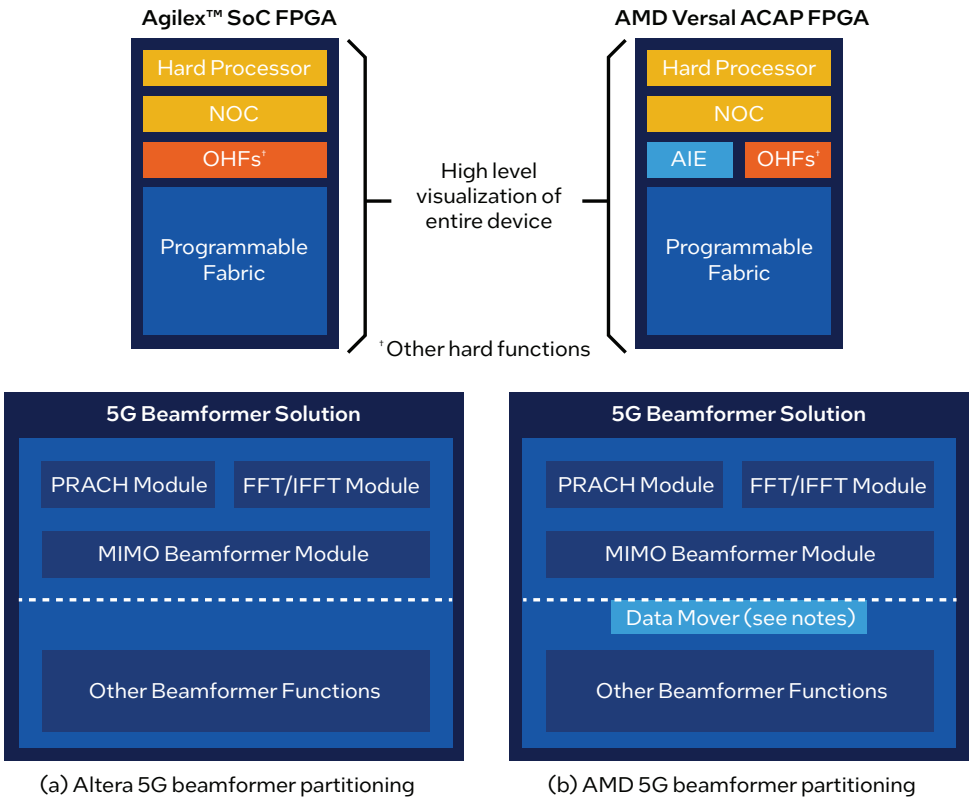
- Agilex 7 device AGFA027R25A3I3E
- AMD Versal AIE device XCVC1902-VSVA2197-1LP-I-L

The designs are implemented on the targeted devices using their respective tool chains, and the collective performance, power, and resource utilization are obtained and compared with a toggle rate of 20%. The 5G beamformer solution has been designed to meet the throughput requirements of 64 antennas with 16 layers capable of simultaneously supporting up to 16 users.

Design Partitioning

For this benchmark, in the Agilex 7 SoC FPGA case, there was no design partitioning per se because all the functions used to implement the 5G MIMO beamformer were implemented in the device's programmable fabric (Figure 3a). This programmable fabric includes thousands of variable-precision DSP blocks and M20K SRAM blocks.

By comparison, in the case of the AMD Versal ACAP FPGA, the three most compute-intensive functions—the MIMO beamformer, PRACH, and FFT/IFFT modules—were implemented in the AIE. The remaining functions required to realize a working 5G beamformer were implemented in the programmable fabric (Figure 3b).



(a) Altera 5G beamformer partitioning

(b) AMD 5G beamformer partitioning

Figure 3. The 5G beamformer design partitioning in Altera and AMD devices.

The 2.14X performance per watt power advantage claimed by AMD was based only on the complex MAC operations associated with the MIMO beamformer, PRACH, and FFT/IFFT modules, all of which were implemented solely in the Versal ACAP's AIE.

AMD's claim does not consider the power associated with moving data back and forth between the programmable fabric and the AIE (see the Data Mover block in Figure 3b and the discussions on this function later in this paper). Also, the claim does not consider any power consumed by any remaining functions required to realize a working 5G beamformer, where these functions are implemented in the programmable fabric as well.

As discussed in the results below, the AMD solution consumes more power than the Altera solution, even when only considering the MIMO beamformer, PRACH, and FFT/IFFT modules.

Results: Resource Utilization

First, we will consider the resources used to implement the compute-intensive MIMO beamformer, PRACH, and FFT/IFFT modules. As we've previously discussed, in the case of the Agilex 7 SoC FPGA solution, all three of these modules are implemented in programmable fabric. By comparison, in the AMD Versal ACAP FPGA case, all three modules are implemented in the AIE. The results of these alternative implementations are as follows.

The MIMO Beamformer Module

To fulfill the performance criteria of the MIMO 5G beamforming configuration, the MIMO beamformer module must process 273 RBs within ~35.7 µs.

In the case of AMD, each AIE performs [12,8]*[8,8] complex matrix multiplications every 120 clock cycles. The design necessitates 16 AIE cores for UL and 16 AIE cores for DL, totaling 32 AIE cores to meet the specified performance requirements. We assumed that no additional fabric is required to enable this functionality.

In the case of Altera, we implemented an equivalent beamformer for both DL and UL in the programmable fabric. The resource utilization for both implementations is detailed in Table 1. Note that M20K and BRAM18 refer to the Altera and AMD SRAM blocks embedded in the programmable fabric, respectively.

Configuration	Agilex 7 FPGA	AMD Versal ACAP
Frequency	491.52 MHz	1 GHz
Adaptive logic module (ALM) / look-up table (LUT)	30,900	0
DSP	816	0
AIE	N/A	32
M20K/BRAM18	0	0

Table 1. Resource utilization for the MIMO beamformer module.

The PRACH Module

The 5G standard specifies the throughput requirements for the PRACH preamble module, which involves performing DDC on sixty-four 122.88 MSps signals (one signal for each of the base station's 64 antennas). Unlike the AMD solution, where users would need to familiarize themselves with and optimize AMD's AIEs, the Altera FPGA fabric offers the option to use the DSP Builder[4] for a streamlined implementation.

The resource utilization for both designs is detailed in Table 2. In the AMD case, the resource utilization aligns with their white paper claims; 4 AIE cores are necessary for 8 antenna paths, so 32 AIE cores are required for 64 antennas. By comparison, DSP Builder can be used to create a PRACH IP that can address the needs of 8 signal paths, so 8 copies of this IP are required to achieve the required throughput.

The resource utilization for both implementations is detailed in Table 2.

Configuration	Agilex 7 FPGA	AMD Versal ACAP
Frequency	491.52 MHz	1 GHz
ALM/LUT	25,632	0
DSP	136	0
AIE	-	32
M20K/BRAM18	104	0

Table 2. Resource utilization for the PRACH module.

The 4K FFT/IFFT Module

The 4K FFT/IFFT module is required to perform 64 FFT/IFFTs every 35.7 µs. Both AMD and Altera meet the specified throughput. In AMD's design, five AIEs are dedicated per 4,096-length FFT to achieve the throughput specification. In this design, every fifth AIE runs only 1,024 4-point FFTs, resulting in lower efficiency than the other four AIEs, where each performs a 1,024-point FFT. This creates a performance bottleneck, which leads to efficiency loss. For a detailed analysis of FIR/FFTs and to learn more about the Agilex 7 FPGA capabilities for this module, please refer to the Power and Performance Analysis of Finite Impulse Response (FIR) Filters and Fast Fourier Transforms (FFT) on Agilex™ 7 FPGAs[5] white paper.

Like the PRACH preamble processing module, Altera's implementation of the 4K FFT/IFFT model utilizes the built-in IP from the DSP Builder, significantly enhancing the user experience. In this scenario, one FFT IP is required for the DL, and another FFT IP is needed for the UL.

The resource utilization for both implementations is detailed in Table 3.

Configuration	Agilex 7 FPGA	AMD Versal ACAP
Frequency	491.52 MHz	1 GHz
ALM/LUT	42,200	0
DSP	472	0
AIE	-	40
M20K/BRAM18	96	0

Table 3. Resource utilization for the 4K FFT/IFFT module.

The Data Mover Function

In the case of AMD's implementation of the 5G beamformer, they strategically integrated the computationally intensive portions of the design onto the AIE. However, certain functionalities, such as data movement, less computationally intensive tasks, and buffering (for example, buffers for the beamforming weights), would be implemented in the device's programmable fabric.

The handling of AXI streams to and from the AIE is managed by programmable logic interface (PLIO) tiles, facilitating the transfer of AXI streams between the AIE and the programmable fabric. However, it must be noted that the handling of AXI streams from programmable fabric to AIE and vice versa necessitates fabric intervention. In this design, we have identified the requirement for 67 streams. Assuming 500 LUTs per stream, the data mover will demand 33,500 LUTs. It's also worth noting that since the Altera design is implemented entirely in the FPGA's programmable fabric, there is no need for an equivalent data mover function.

Results: Power

The total power consumption for the competing beamforming solutions is illustrated in Figure 4. In keeping with the AMD white paper, we consider only the power consumed by the three most computationally intensive functions: the MIMO beamformer, PRACH, and FFT/IFFT modules. In the case of the AMD device, we also include the power consumed by the data mover function, which is implemented in programmable fabric, because the modules in the AIE simply cannot be used without this function.

Based on this, our starting point – 0 on the horizontal axis – is 27.4 watts for the Altera implementation and 28.4 watts for the AMD solution. Apart from anything else, the fact that Altera performs at lower power as showed in the charts in Figure 4 clarifies that AMD's claim of 2.14x performance per watt advantage doesnt hold true.

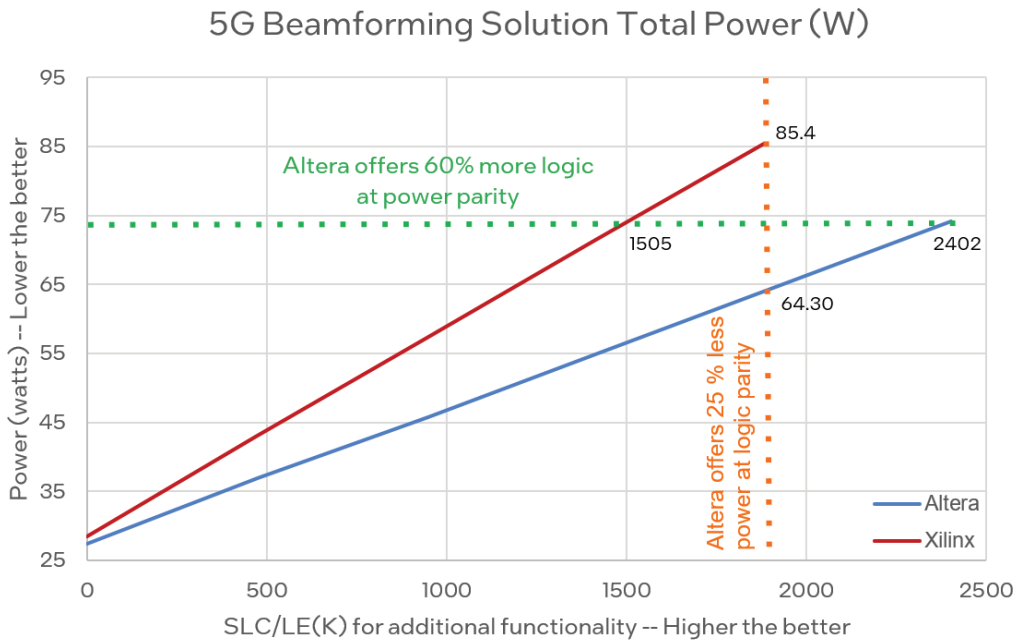


Figure 4. Total power consumption for the competing beamforming solutions.

Remember that this is only a starting point. As we previously noted, a fully functional 5G beamforming solution also requires a suite of additional functions to be implemented in the FPGA's programmable fabric. Altera's logic element (LE) and AMD's system logic cell (SLC) are functionally equivalent, but LEs consumes less power.

As one example of what this means, if we specify a maximum power of 74.22 W for our 5G beamformer implementation, then—in addition to the MIMO beamformer, PRACH, and FFT/IFFT modules—designers working on the AMD part can only employ 1,505,000 SLCs. In contrast, designers working with the Altera device have 2,402,000 LEs at their disposal, which equates to Altera providing 60% more logic at power parity.

Alternatively, if we were to say that our design required 1,900,000 LEs/SLCs, then--including the MIMO beamformer, PRACH, and FFT/IFFT modules—the design would consume 85.46 W if implemented in the AMD part, or only 64.30 W if realized in the Altera device, which equates to Altera consuming 25% less power at logic parity.

Latency Considerations

We observe the use of double buffering facilitated by a window-based Application Programming Interface (API) (API) in AMD's implementation, contributing to increased latency in computational tasks on the AIE. This latency arises from filling the input buffer before kernel execution and draining the output buffer afterward, both operating at 2x32-bit input/output widths.

Adopting a window-based API is essential for maximizing the achieved MACs per cycle. An AIE core in AMD's design executes vector loads from adjacent data memory, performs vector operations, and stores results back into data memory. However, the narrow stream into data memory (2x32-bit input/output width) contrasts with the 256-bit width of vector load/stores. AMD employs double buffering to optimize MAC efficiency by leveraging Xilinx AIE Windows* API between kernels executing on AIE cores. This strategy gradually fills the buffer for the next kernel while the current one is in execution, thereby enabling 256-bit vector load/stores.

However, windows, in this context, are typically large. During the computation phase of a kernel, most MAC operations take place in a loop body, where an input buffer is consumed. Due to the overhead associated with functional calls, preamble, and postamble in a typical kernel function, achieving high MAC efficiency requires large windows. This ensures staying in the loop body for many cycles, thereby reducing overhead costs. The trade-off is an increase in latency due to the necessity of large windows.

In Altera's design, the primary focus is minimizing latency to ensure that samples produced are consumed as soon as possible. No limitations are imposed on the width in and out of each block, reducing latency.

In addition, for typical applications, computationally intensive functions align with AIEs, whereas data movement or rearrangement functions may not. This partitioning implies that compute tasks may need to be distributed across the AIE and Programmable Logic (PL) leading to data transfers between the AIE and PL. For Altera's design, all functions are mapped onto the PL. This means there are no necessary data transfers, which reduces the latency of the overall design.

Conclusion

In this paper, we have carefully analyzed solution briefs and blogs from AMD and implemented a benchmark on closely matched Agilex and Versal AIE devices using default-optimized tool settings. The initial claims from the AMD white paper are presented at a very high level. Altera attempted to understand and diagnose this paper to provide a thorough analysis with transparent results. These results reveal that the Agilex 7 device utilizes 25% less power than the competition at logic parity and offers 60% more logic functionality at power parity. Hence, we can conclude that Altera offers the most cost-effective and efficient 5G mMIMO beamforming implementations.

References

[1] https://xilinx.eetrend.com/files/2021-08/wen_zhang_/100553221-218344-wp539-versal-system-level-benefits.pdf

[2] "Intel Quartus Prime Design Software," [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/development-tools/quartus-prime.html.

[3] "Vivado Design Suite," [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html.

[4] DSP Builder for Intel FPGA," [Online]. Available: https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/dsp-builder.html.

[5] Power and Performance Analysis of Finite Impulse Response (FIR) Filters and Fast Fourier Transforms (FFT) on Agilex 7 FPGAs https://cdrdv2-public.intel.com/792842/power-and-performance-analysis-white-paper.pdf.

WP-01335-1.1