White Paper

Broadcast and ProAV Video and Vision

Performance Advantages on OpenCores with Agilex™ 7 FPGAs

Agilex™ 7 FPGAs deliver more than a speed grade faster core performance for publicly available designs from opencores compared to competing fpgas.

Authors

Zidong Weng
FPGA Development tools Engineer
Altera Corporation

Kiran Tondehal
Technical Marketing Manager
Altera Corporation

Table of Contents

Introduction

This paper presents a methodology for benchmarking the core performance of the Agilex™ 7 FPGA product family, with the goal of transparently presenting the methods and data so that any interested party can reproduce and analyze the results. To achieve this, eight publicly available designs from OpenCores representing a variety of functions were implemented in a device from the Agilex 7 FPGA family and two devices from the closest competitor: the AMD Versal* and Virtex* Ultrascale+* family.

The performance benchmark results show that Agilex 7 FPGAs deliver

- 17% to 27% faster core performance[1] against Versal devices
- 13% to 25% faster core performance[1] against Virtex Ultrascale+ devices
- Stable performance across high utilization compared with competing FPGAs.

Background: Benchmarking of Altera® FPGAs and SoCs

The programmable logic industry does not have a standard benchmarking methodology. Therefore, Altera employs rigorous internal analysis using a broad combination of customer and internally generated designs to understand and quantify the performance of its programmable logic products relative to prior generation Altera products and competing products. The designs are collected from a variety of market segments, such as high-performance computing, image and video processing, wired and wireless communications, and consumer products. Additionally, the designs use a variety of implementation technologies, including ASICs and FPGAs from other vendors. By using a broad suite of designs, Altera ensures that the results are accurate and representative of the complex interaction of customer designs and FPGA design tools such as the Quartus® Prime Software Suite. To use customer designs, Altera invests significant resources in converting designs to work with various synthesis tools and electronic design automation (EDA) vendors. Altera also ensures that functionality is preserved and appropriate code optimizations for the specific FPGA vendor are made, which is necessary because designs are often developed such that they are optimized for a specific FPGA.

For performance comparisons, Altera employs the best effort method – the purpose is to indicate the best possible result achievable. The experiments for the best effort compilation method require longer individual compilation times than in a default push-button compile and more than one compile per design.

Using this methodology, Altera has determined that Agilex 7 FPGAs and SoCs deliver a core performance advantage over competitive 7 nm FPGA products, as measured by the maximum f_{MAX} achievable for the speed-critical clock paths in each of the designs in the design suite. The performance advantage ranges from around 13% to 25% depending on the design, which equates roughly to an advantage

[1] As measured by maximum clock frequencies achieved across lowest and highest utilization respectively on the eight OpenCores designs listed in the paper.

of one or two speed grades, where speed grades are typically defined as a performance difference of 10% to 15%†. These results help validate the position of the Agilex 7 FPGAs as the highest performance 10 nm SuperFin FPGA family. However, because these results were obtained using customer and Altera proprietary designs, Altera can share only limited details of the analysis, which ultimately limits the usefulness of this information to programmable logic users.

Increasing Transparency via OpenCores-based Performance Comparisons

To address this challenge to understanding programmable logic performance, Altera has undertaken a benchmarking effort using publicly available designs from OpenCores (www.opencores.org), an organization that offers open-source hardware intellectual property (IP) cores. The goal of this benchmarking effort is to help programmable logic users:

- Understand the exact designs used in the performance evaluation, including the specific details of those designs down to the register transfer level (RTL) description
- Reproduce the results of the analysis themselves
- Scrutinize the results of the analysis to better understand the applicability of the Agilex 7 FPGA performance and device utilization advantage to their specific design

The scope of this OpenCores-based analysis is narrower than the internal analysis that Altera employs because it focuses specifically on timing-constrained compilations. This analysis is not a comprehensive conclusion, but the results provide insight into the relative performance of Agilex 7 FPGAs compared with competitive devices when implementing similar designs or designs that are composed of functions similar to the ones used in the design example suite.

Target Device Families for Performance Comparison

Altera chose its Agilex 7 FPGA family and the Versal and Virtex Ultrascale+ family for the performance analysis.

- Agilex 7 device: AGFB014R24A2E2V
- Versal device: XCVM18022MSEVSV2197
- Virtex Ultrascale+: XCVU7P2FLVA2104E

Note: Based on internal tests, smaller devices within the families, Versal Prime VM1402 of the same speed grade exhibit similar performance levels. We observed that 2/8 designs couldn't fit into the smaller devices at larger stamp instances, hence chose the above devices for showing results.

Opencores Designs Used in the Analysis

Altera selected OpenCores designs based on design size and complexity, with the intent of representing a wide variety of function types that use a mix of different device resource types, such as logic, RAM, and DSP. Table 1 lists the OpenCores designs used in the performance comparison and links to the OpenCores web page for each design where users can learn more about the design and download it. The table also shows the average amount of logic utilized by each of the OpenCores designs, measured using:

- Adaptive logic modules (ALMs) for Agilex 7 FPGAs
- Configurable logic block (CLB) look-up tables (LUTs) for Versal and Virtex Ultrascale+ FPGAs

Note: ALMs and CLB LUTs have the following architectural differences – ALMs use LUTs with eight inputs and CLBs use LUTs with six inputs. Therefore, the devices are expected to have different utilization numbers for a given design.

OpenCores Stamping Methodology

The OpenCores designs use only a small fraction of the resources in the target devices. Utilizing only a small fraction of the total device resources is not a common practice or desired goal among programmable logic users. Also, increasing utilization often has a negative impact on the highest achievable f_{MAX} as device resources become exhausted and the design becomes harder to place and route. To simulate the impact of device utilization on programmable logic performance, Altera performed a large number of compilations, each one incrementally adding more OpenCores instances compared to the prior compilation. To increase the design size in the programmable logic device, each OpenCores design was instantiated repeatedly (multiple stamps of the same core) in the FPGA such that:

- Each stamp was implemented in parallel
- An I/O wrapper logic was added to reduce the number of I/O pins required for the larger design
- No timing critical paths between the cores and the wrapper logic existed
- The wrapper logic provided as little overhead as possible

Figure 1 illustrates the stamping process.

OpenCores Stamping and Benchmarking Methodology

As the number of stamps of the OpenCores design increases (and thus design size increases), resources such as I/O pins and global clocks become limited. To avoid running out of pins, each OpenCores design was wrapped in a shift register, such that one physical pin would feed all input pins of a core and all output pins of a core would feed into a loadable shift register. Figures 2 and 3 show the input and output shift registers respectively. The shift register size is dependent on the number of I/O pins, and the number of shift registers is dependent on the number of OpenCores designs implemented in the FPGA.

No	OpenCores Design	Design Function	URL	Number of Stamp instances	Device Utilization				
					Agilex™ 7 Device ALMs	Versal* Device CLB LUTs	Virtex* Ultrascale+* Device CLB LUTs	Agilex™ 7 Device DSP	Agilex™ 7 Device M20K
1	oc_warp_tmu	Image Processing	https://opencores.org/project,warp	50	129,352	194,021	139,720	100	150
2	oc_reed_solomon_decoder	Error Correction Code	https://opencores.org/project,reed_solomon_decode	50	118,076	203,025	176,916	0	750
3	oc_usbhostslave	USB 1.1 Controller	https://opencores.org/project,usbhostlave	200	256,540	357,694	316,813	0	0
4	oc_dma_axi64	Single-channel 64-bit AXI Master DMA	https://opencores.org/project,dma_axi	50	101,433	141,096	127,557	0	0
5	oc_256_aes	AES	Note 2	12X12	140,350	368,516	328,514	0	0
6	oc_m1_core	32-bit RISC Processor	http://opencores.org/project,m1_core,overview	50	101,684	162,387	152,269	0	0
7	oc_fpu100	32-bit Floating Point Unit	https://opencores.org/project,fpu100	50	102,047	158,007	134,301	0	0
8	oc_xge_mac	Ethernet MAC Controller	https://opencores.org/project,xge_mac	50	96,336	143,298	140,197	0	400

Table 1. Eight OpenCores Designs Used in the Performance Comparison

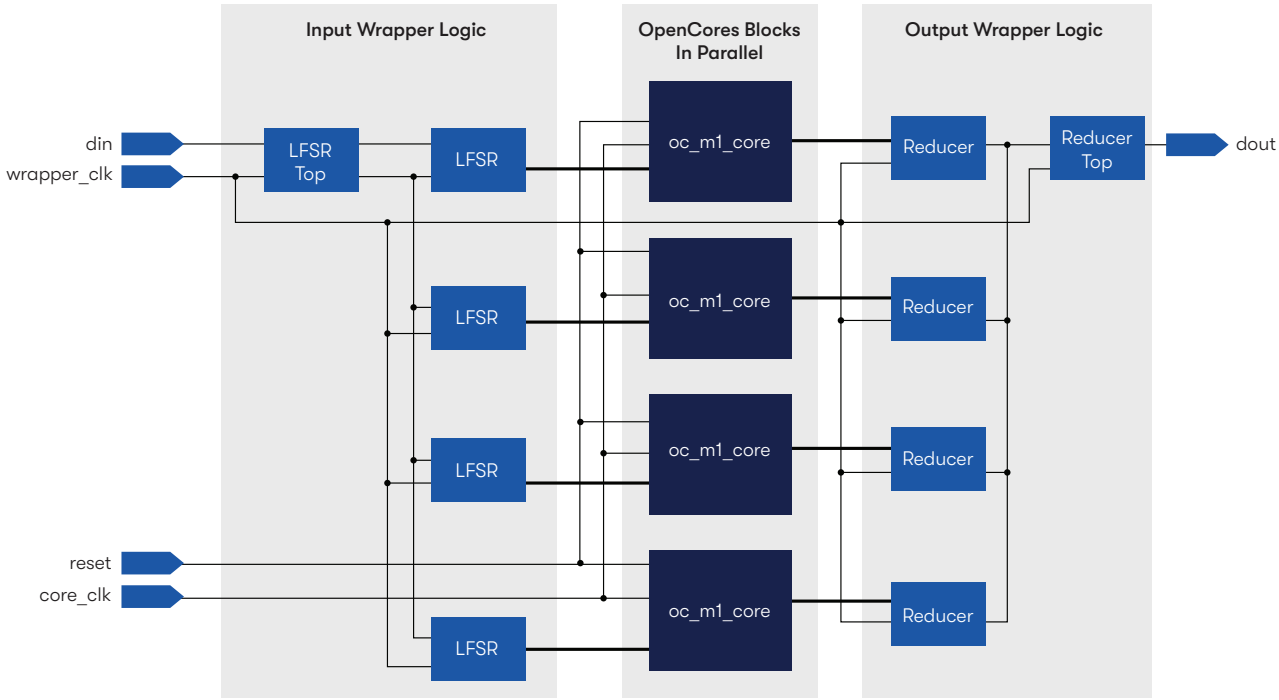

Figure 1. oc_m1_core Design Instantiated Four Times in the FPGA

[2] The oc_256_aes was available from OpenCores when the performance comparison project was started. However, it is now unavailable. The design is still available under its open-source license from Altera at the links included at the end of this paper.

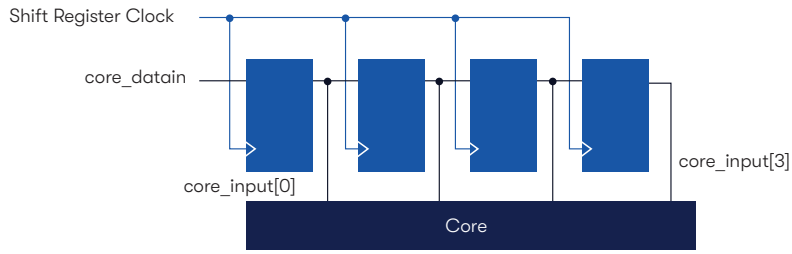

Figure 2. Input Shift Register Implementation

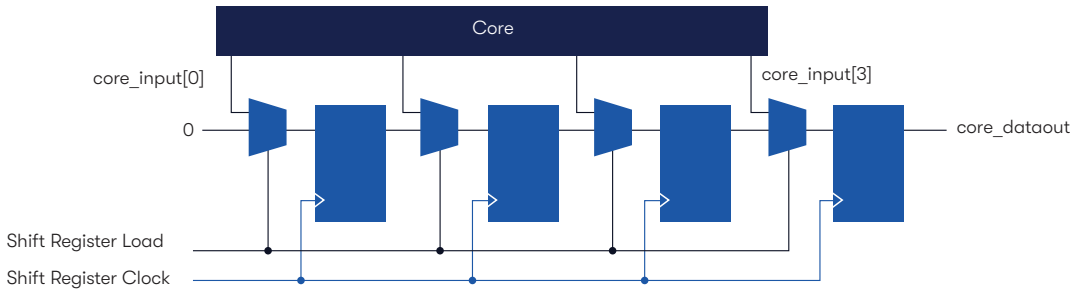

Figure 3. Output Shift Register Implementation

To avoid running out of global clock resources, a pin directly fed the global clock and reset signals for all OpenCores designs. For example, if a core required two clocks (i.e.; core clock 1 and core clock 2) and one reset signal, all instances of core clock 1 were fed by one pin, all instances of core clock 2 were fed by a different pin, and all instances of the reset signal were fed by a third pin. With this method, all OpenCores designs were fed by the same clock and reset signals (see Figure 4).

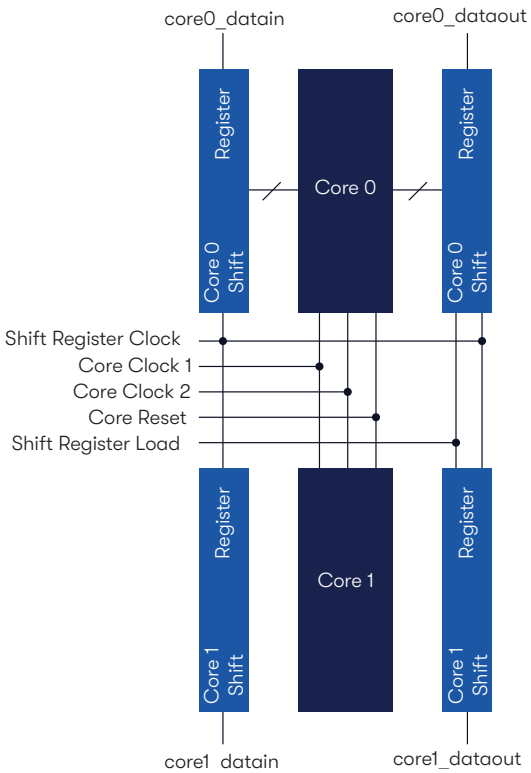

Figure 4. Two-core implementation with shared clock and reset signals

Once the wrapper logic tied up all of the OpenCores designs in the FPGA, Altera ensured that no critical paths existed between the wrapper logic (i.e.; shift registers) and the OpenCore designs. To achieve this goal, false paths were created and, by making the core clock(s) and wrapper logic clock on different unrelated clock domains, no timing paths existed. The design tools could then optimize the cores separately from the shift registers. Altera instantiated the OpenCores designs as many times as the device and design tools would allow without compilation errors.

Software Tools, Settings, and Constraints

To perform this study, Altera used the latest version of the required FPGA development tools that were available at the time of the analysis:

- Quartus Prime Software Suite version 22.4
- AMD Vivado* software version 2022.2

Both tools were installed and operated on Linux64 machines. These programmable logic tools offer settings that provide a trade-off among design performance, logic resource consumption, compile time, and memory usage. The settings that produce the best results for one design are likely not the best for another. Additionally, user constraints that guide the EDA tool can improve the results. Even with a design set that is representative of customer designs, the benchmarking outcome varies significantly with software settings and applied constraints. For the comparisons presented in this paper, Altera used the best effort compilation mode and set aggressive timing constraints. To determine aggressive timing constraints for each design, Altera applied a frequency constraint to each OpenCores design clock such that the constraint is just beyond what is achievable for each clock. Altera determined a base constraint value by increasing the constraint until it could not be met. Then, Altera determined the aggressive constraint by multiplying the base constraint value by a factor of at least 1.3. The following sections describe the constraints applied to each design.

Individual OpenCores Design Compilation Results – Performance

This section provides the detailed compilation results for each OpenCores design. In each case, a graph is provided of the fMAX achieved for each design compilation.

- The vertical axis measures the fMAX of the compilations
- The horizontal axis measures the core fabric utilization, as measured by logic utilization (Agilex 7 FPGA ALMs and Versal and Virtex Ultrascale+ device CLB LUTs)

The data points at the leftmost edge are for the compilations corresponding to lowest device utilization starting at 10% to 20% having same stamp sizes and going up to when the device gets almost full with larger stamp sizes towards the right. As indicated in most of the graphs, the fMAX values are quite stable for the majority of the device utilization and only tends to fall off when one of the resources, such as logic, memory, DSP, routing, and so on, becomes limited.

OC_Warp_TMU Core

Figure 5 shows the fMAX results for the OC _Warp_TMU image processing design compilations. The Agilex 7 FPGA compilations are successful and stable until 93% utilization, producing a geometric mean of 362MHz. The Versal fMAX begins to fall off by ~12% at 75% utilization producing a geometric mean of 346MHz. The Virtex Ultrascale+ fMAX values begin to fall off by ~17% at about 81% utilization producing a geometric mean fMAX of 336 MHz.

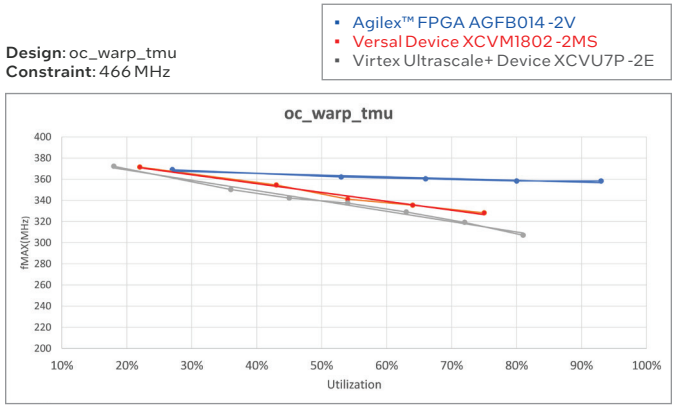

Figure 5. OC_Warp_TMU Results

OC_Reed-Solomon-Decoder Core

Figure 6 shows the fMAX results for the compilations of the OC_Reed-Solomon_Decoder, a function commonly used for error correction. The Agilex 7 FPGA fMAXvalues are stable and within 6% until 94% utilization, producing a geometric mean fMAX of 476 MHz. The Versal device fMAX values see a drop off of ~14% at 92% utilization and producing a geometric mean of 415 MHz. The Virtex Ultrascale+ device fMAX values also see a drop off of ~14% at 90% utilization and producing a geometric mean fMAX of 425 MHz. The last successful compilations are at about 94% utilization, and fail afterwards due to insufficient logic.

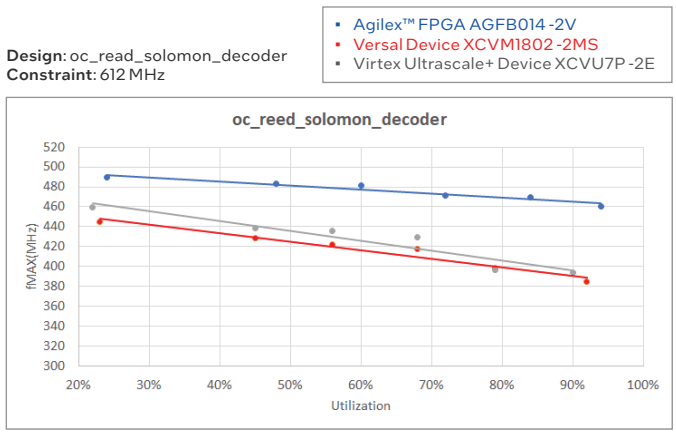

Figure 6. Reed-Solomon Decoder Results

OC_DMA_AXI64 Core

Figure 7 shows the fMAX results for the compilations of the OC_DMA_AXI64, a single-channel 64-bit AXI master direct-memory access function. The Agilex 7 FPGA fMAX values are stable and within 5% until 97%+ utilization, producing a geometric mean fMAX of 461 MHz. The Versal device fMAX is stable until it sees a 10% drop off at 85% utilization producing a geometric mean fMAX of 376 MHz. The Virtex Ultrascale+ device fMAX values fall off by 19% at 81% utilization producing a geometric mean fMAX of 397 MHz.

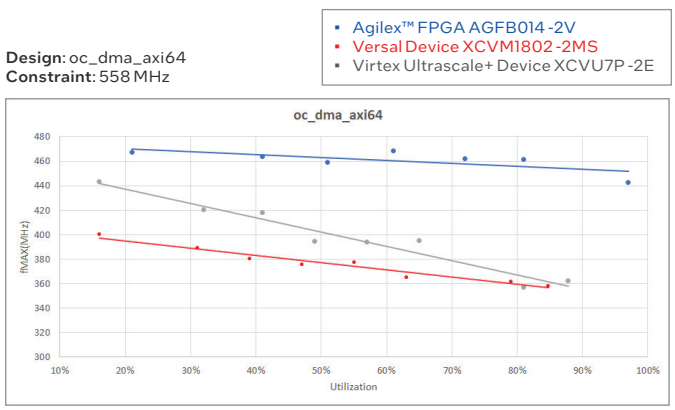

Figure 7. DMA AXI64 Results

OC_XGE_MAC

Figure 8 shows the fMAX results for the compilations of the OC_XGE_MAC, implements media access control (MAC) function for the 10 Gbps operation. The Agilex 7 FPGA fMAX values are stable and produce a geometric mean fMAX of 546 MHz. The Versal device fMAX starts to drop off from 74% utilization and provides a geometric mean of 415 MHz. The Virtex Ultrascale+ device fMAX values show ~20% drop off at 92% utilization, producing a geometric mean fMAX of 419 MHz.

Design: oc_xge_mac
Constraint: 638 MHz

- Agilex™ FPGA AGFB014 -2V
- Versal Device XCVM1802 -2MS
- Virtex Ultrascale+ Device XCVU7P -2E

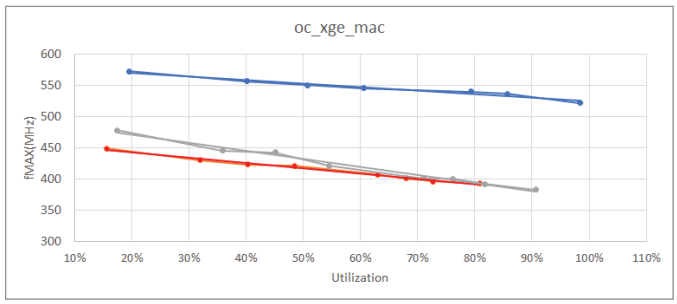

Figure 8. XGE MAC Results

OC_USBHostSlave Core

Figure 9 shows the fMAX results for the compilations of the OC_USBHostSlave, a USB 1.1 controller. The Agilex 7 FPGA fMAX values are stable and produce a geometric mean fMAX of 776 MHz. The Versal device fMAX starts to drop off from 80% utilization and provides a geometric mean of 549 MHz. The Virtex Ultrascale+ device fMAX values are faster than the Versal device and stable producing a geometric mean fMAX of 622 MHz.

Design: oc_usbhostslave
Constraint: 859.1MHz

- Agilex™ FPGA AGFB014 -2V
- Versal Device XCVM1802 -2MS
- Virtex Ultrascale+ Device XCVU7P -2E

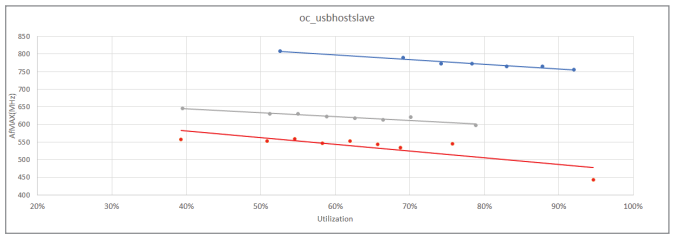

Figure 9. USB Host Slave Results

OC_256_AES Core

Figure 10 shows the fMAX results for the compilations of the OC_256_AES, a 256-bit Advanced Encryption Standard (AES) function. The Agilex 7 FPGA fMAX values are stable and start to fall off by 9% starting at 86% utilization, producing a geometric mean fMAX of 367 MHz. The Versal device fMAX values drop off by ~9% at 91% utilization, producing a geometric mean of 277 MHz. The Virtex Ultrascale+ device fMAX values are within 7%, producing a geometric mean fMAX of 269 MHz.

Design: oc_256_aes
Constraint: 406.50 MHz

- Agilex™ FPGA AGFB014 -2V
- Versal Device XCVM1802 -2MS
- Virtex Ultrascale+ Device XCVU7P -2E

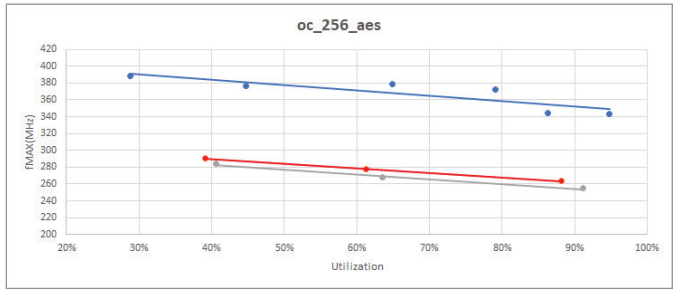

Figure 10. Advanced Encryption (AES) Results

OC_M1 Core

Figure 11 shows the fMAX results for the compilations of OC_M1, a 32-bit processor core. The Agilex 7 FPGA fMAX values are stable producing a geometric mean fMAX of 419 MHz. The Versal device fMAX values drop off by ~17% at 90% utilization, producing a geometric mean of 343 MHz. The Virtex Ultrascale+ device values also fall off by 18% at about 88% utilization, producing a geometric mean fMAX of 336 MHz.

Design: oc_m1_core
Constraint: 465.3 MHz

- Agilex™ FPGA AGFB014 -2V
- Versal Device XCVM1802 -2MS
- Virtex Ultrascale+ Device XCVU7P -2E

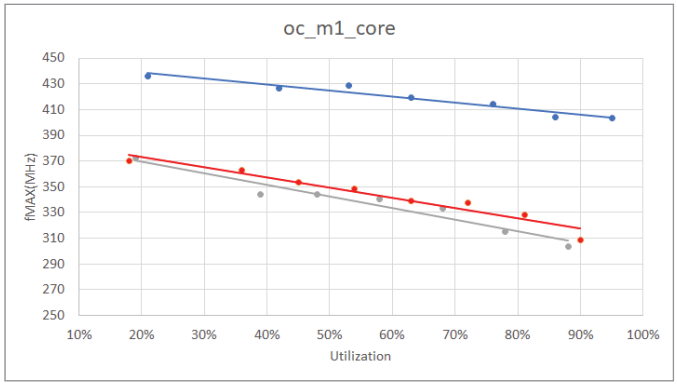

Figure 11. M1 (32 bit Processor) Results

OC_FPU100 Core

Figure 12 shows the fMAX results for the compilations of the OC_ FPU100, a floating-point unit function. The Agilex 7 FPGA fMAX values are stable and within 5% across utilization, producing a geometric mean fMAX of 367 MHz. The Versal device fMAX values fall off by ~16% at 94% utilization, producing a geometric mean of 357 MHz. The Virtex UltraScale+ device fMAX values tend to drop by 16% at 85% utilization – producing a geometric mean fMAX of 341 MHz.

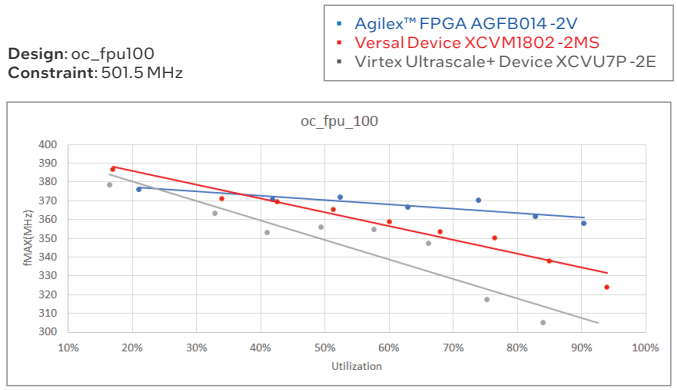

Figure 12. FPU100 Results

No	OpenCores Design Name	Design Function	Performance	Positive % is Better
			(Geometric mean) Agilex™ 7 FPGA fMAX/ Versal* Device fMAX	(Geometric mean) Agilex™ 7 FPGA fMAX/ Virtex* Ultrascale+* Device fMAX
1	oc_warp_tmu	Image Processing	+5%	+8%
2	oc_reed_solomon_decoder	Error Correction Code	+15%	+12%
3	oc_usbhostslave	USB 1.1 Controller	+41%	+25%
4	oc_dma_axi64	Single-channel 64-bit AXI Master DMA	+23%	+16%
5	oc_256_aes	AES	+32%	+37%
6	oc_m1_core	32-bit RISC Processor	+22%	+25%
7	oc_fpu100	32-bit Floating Point Unit	+3%	+8%
8	oc_xge_mac	Ethernet MAC Controller	+32%	+30%
	Geometric mean across all utilization		+21%	+20%
	Geometric mean across low ultilization		+17%	+13%
	Geometric mean across high utilization		+27%	+25%

Table 2. OpenCore Results Summary

Conclusion

Table 2 summarizes the relative performance and the device utilization used by the Agilex 7 FPGA family relative to the Versal and Virtex Ultrascale+ device families.

Across the eight benchmark designs, all designs achieve higher fMAX values in the Agilex 7 FPGA family vs. the competing devices, in the range of 3% to 41% with a geometric mean of 20%. This advantage represents a one to two speed grades better performance. Additionally, the results show better performance stability across higher utilization. The Hyperflex® Architecture delivers higher performance even with higher utilization, allowing you to use more of the available logic.

Altera provides this data and the designs upon which the analysis is based with the intent of increasing transparency and understanding among programmable logic users of the performance capabilities of Agilex 7 FPGAs. Agilex 7 FPGAs and SoCs were designed to be the highest performance products in their class, and the comparisons described in this analysis using publicly available designs help to reinforce that they deliver industry-leading speeds across wide range of applications.

References

www.opencores.org

Learn More:

- For more information about Altera and Agilex 7 FPGAs, visit the Agilex 7 FPGA and SoC FPGA web page
- For more information about the high-performance architecture of Agilex 7 devices, download the Agilex FPGA Architecture White Paper

WP-01328-1.2