Introduction

This paper presents a methodology for benchmarking the core performance of the Intel 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 Intel Agilex 7 FPGA family and two devices from the closest competitor: the AMD Versal® and Virtex® Ultrascale+ family.

The performance benchmark results show that Intel Agilex 7 FPGAs deliver:

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

Background: Benchmarking of Intel® FPGAs and SoCs

The programmable logic industry does not have a standard benchmarking methodology. Therefore, Intel 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 Intel 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, Intel ensures that the results are accurate and representative of the complex interaction of customer designs and FPGA design tools such as the Intel® Quartus® Prime Software Suite. To use customer designs, Intel invests significant resources in converting designs to work with various synthesis tools and electronic design automation (EDA) vendors. Intel 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, Intel 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, Intel has determined that Intel Agilex 7 FPGAs and SoCs deliver a core performance advantage over competitive 7 nm FPGA products, as measured by the maximum f_{\text{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 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 Intel Agilex 7 FPGAs as the highest performance 10 nm SuperFin FPGA family. However, because these results were obtained using customer and Intel proprietary designs, Intel 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, Intel 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 Intel 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 Intel 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 Intel 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.

**OpenCores designs used in the analysis**

Intel 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 Intel 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.

**Target device families for performance comparison**

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

- Intel Agilex 7 device: AGFB014R24A2E2V
- Versal device: XCVM18022MSESV2197
- 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 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 $f_{\text{MAX}}$ 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, Intel 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.
Table 1. Eight OpenCores Designs Used in the Performance Comparison

<table>
<thead>
<tr>
<th>No</th>
<th>OpenCores Design</th>
<th>Design Function</th>
<th>URL</th>
<th>Number of Stamp instances</th>
<th>Device Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>oc_warp_tmu</td>
<td>Image Processing</td>
<td><a href="https://opencores.org/project.warp">https://opencores.org/project.warp</a></td>
<td>50</td>
<td>Intel Agilex® 7 Device ALMs 129,352</td>
</tr>
<tr>
<td>2</td>
<td>oc_reed-solomon_decoder</td>
<td>Error Correction Code</td>
<td><a href="https://opencores.org/project.reed_solo">https://opencores.org/project.reed_solo</a></td>
<td>50</td>
<td>Intel Agilex® 7 Device CLB LU Ts 118,076</td>
</tr>
<tr>
<td>3</td>
<td>oc_usbhostslave</td>
<td>USB 1.1 Controller</td>
<td><a href="http://opencores.org/project.usbhostslave">http://opencores.org/project.usbhostslave</a></td>
<td>200</td>
<td>Intel Agilex® 7 Device CLB LU Ts 256,540</td>
</tr>
<tr>
<td>4</td>
<td>oc_dma_axi64</td>
<td>Single-Channel 64-bit AXI Master DMA</td>
<td><a href="https://opencores.org/project.dma_axi">https://opencores.org/project.dma_axi</a></td>
<td>50</td>
<td>Intel Agilex® 7 Device DSP 101,433</td>
</tr>
<tr>
<td>5</td>
<td>oc_256_aes</td>
<td>Advanced Encryption Standard (AES)</td>
<td>Note 2</td>
<td>12X12</td>
<td>Intel Agilex® 7 Device DSP 140,350</td>
</tr>
<tr>
<td>6</td>
<td>oc_m1_core</td>
<td>32-bit RISC Processor</td>
<td><a href="http://opencores.org/project.ml_core,overview">http://opencores.org/project.ml_core,overview</a></td>
<td>50</td>
<td>Intel Agilex® 7 Device DSP 101,684</td>
</tr>
<tr>
<td>7</td>
<td>oc_fpu100</td>
<td>32-bit Floating Point Unit</td>
<td><a href="https://opencores.org/project.fpu100">https://opencores.org/project.fpu100</a></td>
<td>50</td>
<td>Intel Agilex® 7 Device DSP 102,047</td>
</tr>
<tr>
<td>8</td>
<td>oc_xge_mac</td>
<td>Ethernet MAC controller</td>
<td><a href="https://opencores.org/project.xge_mac">https://opencores.org/project.xge_mac</a></td>
<td>50</td>
<td>Intel Agilex® 7 Device DSP 96,336</td>
</tr>
</tbody>
</table>

Figure 1. oc_m1_core Design Instantiated Four Times in the FPGA

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 Intel at the links included at the end of this paper.
Software tools, settings, and constraints

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

- Intel 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, Intel used the best effort compilation mode and set aggressive timing constraints. To determine aggressive timing constraints for each design, Intel applied a frequency constraint to each OpenCores design clock such that the constraint is just beyond what is achievable for each clock. Intel determined a base constraint value by increasing the constraint until it could not be met. Then, Intel 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 $f_{\text{MAX}}$ achieved for each design compilation.

- The vertical axis measures the $f_{\text{MAX}}$ of the compilations
- The horizontal axis measures the core fabric utilization, as measured by logic utilization (Intel 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 $f_{\text{MAX}}$ 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 $f_{\text{MAX}}$ results for the OC_Warp_TMU image processing design compilations. The Intel Agilex® 7 FPGA compilations are successful and stable until 93% utilization, producing a geometric mean of 362MHz. The Versal $f_{\text{MAX}}$ begins to fall off by ~12% at 75% utilization producing a geometric mean of 346MHz. The Virtex Ultrascale+ $f_{\text{MAX}}$ values begin to fall off by ~17% at about 81% utilization producing a geometric mean $f_{\text{MAX}}$ of 336 MHz.

**OC_Reed-Solomon-Decoder core**

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

**OC_DMA_AXI64 core**

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

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

OC_USBHostSlave core

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

OC_256_AES core

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

OC_M1 core

Figure 11 shows the $f_{\text{MAX}}$ results for the compilations of OC_M1, a 32-bit processor core. The Intel Agilex 7 FPGA $f_{\text{MAX}}$ values are stable producing a geometric mean $f_{\text{MAX}}$ of 419 MHz. The Versal device $f_{\text{MAX}}$ 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 $f_{\text{MAX}}$ of 336 MHz.
**OC_FPU100 core**

Figure 12 shows the $f_{\text{MAX}}$ results for the compilations of the OC_FPU100, a floating-point unit function. The Intel Agilex 7 FPGA $f_{\text{MAX}}$ values are stable and within 5% across utilization, producing a geometric mean $f_{\text{MAX}}$ of 367 MHz. The Versal device $f_{\text{MAX}}$ values fall off by ~16% at 94% utilization, producing a geometric mean of 357 MHz. The Virtex UltraScale+ device $f_{\text{MAX}}$ values tend to drop by 16% at 85% utilization – producing a geometric mean $f_{\text{MAX}}$ of 341 MHz.

![Figure 12. FPU100 Results](image-url)

<table>
<thead>
<tr>
<th>No</th>
<th>OpenCores Design Name</th>
<th>Design Function</th>
<th>Performance</th>
<th>Positive % is Better</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>oc_warp_tmu</td>
<td>Image Processing</td>
<td>+5%</td>
<td>+8%</td>
</tr>
<tr>
<td>2</td>
<td>oc_reed_solomon_decoder</td>
<td>Error Correction Code</td>
<td>+15%</td>
<td>+12%</td>
</tr>
<tr>
<td>3</td>
<td>oc_usbhostslave</td>
<td>USB 1.1 Controller</td>
<td>+41%</td>
<td>+25%</td>
</tr>
<tr>
<td>4</td>
<td>oc_dma_axi64</td>
<td>Single-channel 64-bit AXI Master DMA</td>
<td>+23%</td>
<td>+16%</td>
</tr>
<tr>
<td>5</td>
<td>oc_256_aes</td>
<td>AES</td>
<td>+32%</td>
<td>+37%</td>
</tr>
<tr>
<td>6</td>
<td>oc_m1_core</td>
<td>32-bit RISC Processor</td>
<td>+22%</td>
<td>+25%</td>
</tr>
<tr>
<td>7</td>
<td>oc_fpu100</td>
<td>32-bit Floating Point Unit</td>
<td>+3%</td>
<td>+8%</td>
</tr>
<tr>
<td>8</td>
<td>oc_xge_mac</td>
<td>Ethernet MAC Controller</td>
<td>+32%</td>
<td>+30%</td>
</tr>
</tbody>
</table>

| Geometric mean across all utilization | +21% | +20% |
| Geometric mean across low utilization | +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 Intel Agilex 7 FPGA family relative to the Versal and Virtex Ultrascale+ device families.

Across the eight benchmark designs, all designs achieve higher $f_{\text{MAX}}$ values in the Intel 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 Intel Hyperflex™ Architecture delivers higher performance even with higher utilization, allowing you to use more of the available logic.

Intel 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 Intel Agilex 7 FPGAs. Intel 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

Where to get more information

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

1 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 https://edc.intel.com/content/www/us/en/products/performance/benchmarks/fpga/

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.

Intel reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

No product or component can be absolutely secure.

Your costs and results may vary.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. *Other names and brands may be claimed as the property of others.