WP455 (v1.2) October 29, 2015

UltraScale Architecture: Highest Device Utilization, Performance, and Scalability

By: Nick Mehta

High-performance architecture and extensive package migration enable designers to build multiple variations of next-generation applications in UltraScale™ devices with optimal design reuse—resulting in product differentiation and a time to market advantage.

ABSTRACT

Devices are forever getting more complex, with increasing density and capability from one generation to the next. There is no letup, however, on the designers' need to get their products to production before their competition to be successful in their market.

The Xilinx® UltraScale architecture adds numerous technical innovations over existing architectures, providing devices that exceed the performance, utilization, and capacity demands of next-generation applications. Offering both architectural migration and package footprint migration, UltraScale devices enable users to build multiple variants of their systems with maximum design reuse and minimal PCB rework.

Market Demands

Many next-generation markets and applications require a tremendous increase in system bandwidth and processing capability. Whether wired or wireless communications, video, or image processing applications, increased data throughput requirements have the same result: increased traffic and demands on all system components. More data arrives on-chip through parallel and serial I/O. The data must then be buffered, again through both parallel I/O in the form of DDR memory and serial I/O in the form of serial memory, before being processed in the logic and DSP then finally transmitted to its next destination back through the parallel and serial I/O.

System processing requirements are becoming more complex for a number of reasons; larger data packets traveling at an increased data rate result in wider parallel data buses at increased frequency. To efficiently process the data, it is often necessary to build an entire system in a single device. This eliminates the latency and power consumption associated with sending large quantities of data between two FPGAs, however, it requires ever increasing density and capability on that single device. It is imperative that, as these high-capability FPGAs are more heavily utilized, they maintain the ability to operate at their maximum possible performance, avoiding performance degradation even at very high device utilization.

To control engineering costs in highly competitive markets, it is increasingly popular to build multiple flavors of systems with maximum possible re-use. A single device can be repurposed to fulfill low-end, mid-range, and high-end applications—but it is more likely and more cost effective that the different variations of the platform will use different devices.

UltraScale Architecture for High-Performance Designs

To address market demands, Xilinx redefined the traditional FPGA architecture, building on a foundation that brought many years of success, with key architectural changes to support the challenges of tomorrow's designs. The need to route ever wider data buses and store and process the data at ever higher clock rates necessitated several changes.

Logic and Interconnect

The primary logic building blocks of the FPGA architecture are the Configurable Logic Blocks (CLBs), which contain multiple registers and look-up tables. To achieve the highest possible performance, it is desirable to tightly pack together the elements of a design. The UltraScale architecture provides an enhanced CLB compared to previous-generation FPGAs to make the most efficient use of the available resources, with the goal of reducing total interconnect (i.e., total wire length). Every aspect of the existing CLB structure, shown in Figure 1, was analyzed to explore how the components can be used more efficiently.

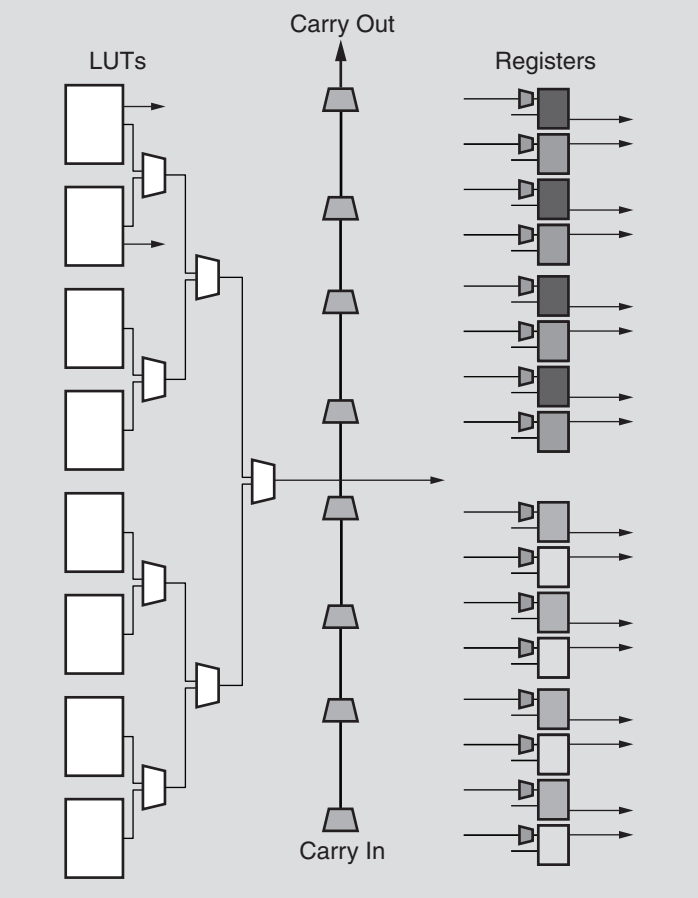

Figure 1: **CLB Architecture**

By merging all the logical resources into a single CLB structure, there is an additional stage of multiplexing for creating wider multiplexers and a longer, 8-bit, carry chain that enables faster arithmetic functions.

At the heart of the CLB are the look-up tables (LUTs) and the registers. See Figure 2.

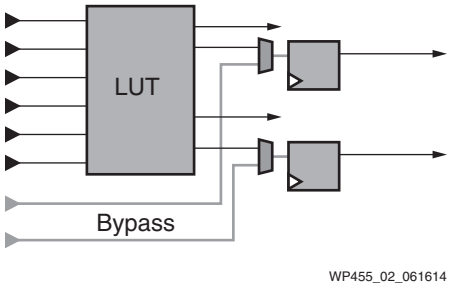

Figure 2: **LUT and Registers**

In the UltraScale architecture, all the elements have their own connectivity—unique inputs and outputs—enabling more efficient packing of unrelated functions and therefore higher performance and more compact designs. This extra connectivity eliminates any need to route through a LUT to gain access to the associated registers. The registers in the UltraScale architecture-based CLB benefit from double the number of clock enable signals compared to the existing architecture as well as several flexibility enhancements such as local ignore and inversion attributes. Having more control signals with increased flexibility provides the software with additional flexibility to use all the resources within each and every CLB in the UltraScale architecture.

When examining the routing architecture in conventional FPGA technology, one of the biggest issues is that as device density increases, the logic grows by a factor of *N*-squared. So moving up from smaller devices to larger devices, this becomes almost exponential growth. At the same time, the interconnect tracks with a conventional architecture only grow by a factor of *N*. So as devices increase in density, a gap develops between the amount of logical resources and the number of interconnect tracks.

Closing the gap between logical resources and interconnect tracks is one of the fundamental challenges that the UltraScale architecture addresses. First, the more traditional interconnect switching architecture has been redesigned, making it smaller, more agile, and more flexible. The UltraScale architecture also doubles the quantity of horizontal and vertical routing tracks and adds more direct routes from point A to point B within the interconnect of the FPGA.

The impact of the architectural enhancements in the CLB and interconnect give the Vivado® Design Suite much more flexibility around design placement, enabling consistently high design performance even when these extremely high density and capability FPGAs reach very high resource utilization.

Traditional and competing software tools use old technology based on simulated annealing, i.e., using a random initial placement seed and random moves that only optimize for a global variable. With the Vivado tools, the placer is capable of mitigating congestion. Its analytical placer is a mathematical solver that finds a solution and optimizes for three variables at the same time: timing, congestion and wire length. Co-optimization of the UltraScale architecture and the Vivado design suite (using the intelligent congestion aware placer to predict where congestion will be encountered in designs) alleviates any congestion bottleneck. Even as these high density UltraScale devices achieve high resource utilization, they deliver consistently high performance and predictable software runtimes.

Co-optimizing the UltraScale architecture with the Vivado Design Suite allows much more logic to be packed into a given device, allowing the tools to shorten total design wire length and achieve consistently high performance as the device utilization increases. See Figure 3.

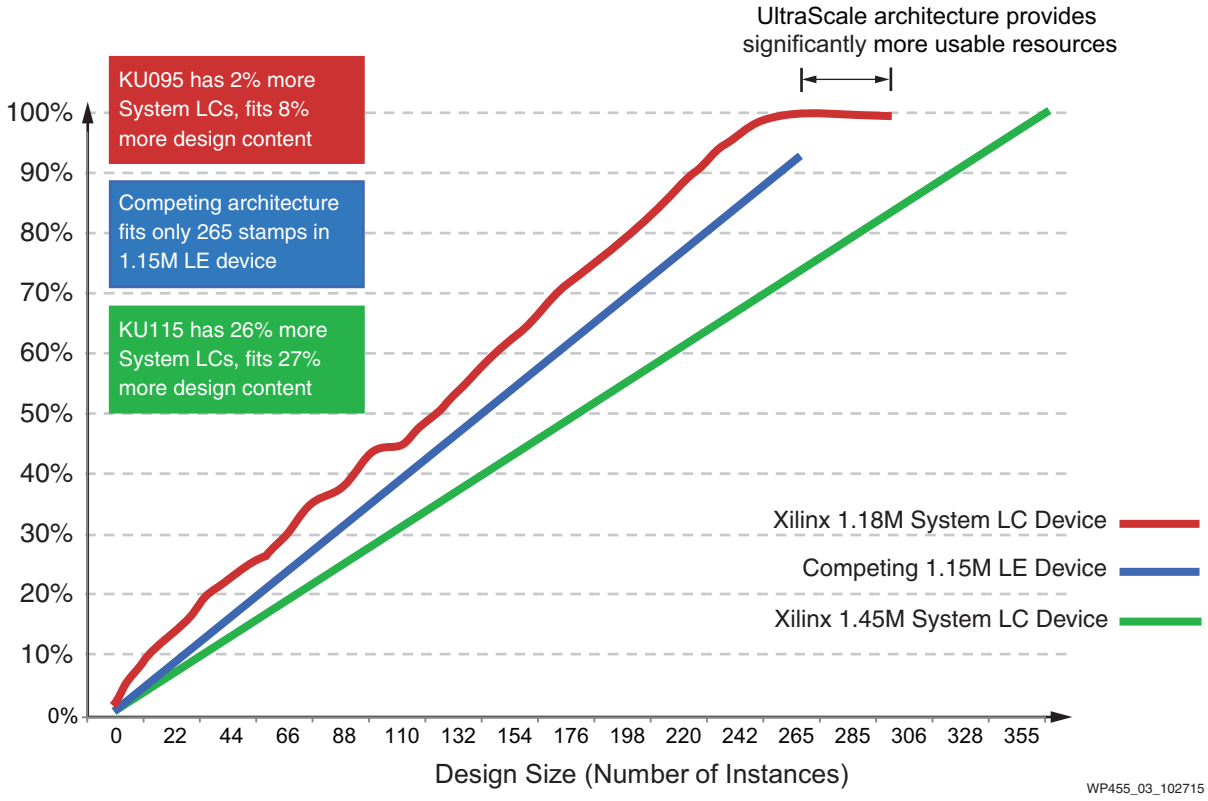

Figure 3: **UltraScale Architecture Density Advantage**

Figure 3 plots the device utilization of the UltraScale architecture and Vivado design suite against a competing product. A design from opencores.org, which was not optimized for either architecture, was placed numerous times in both architectures, with device utilization being monitored. The competing architecture approaches 100% resource utilization when only 265 design instances are placed and can then fit no more instances of the design into the device. The efficient packing enabled by the UltraScale architecture allows the Vivado design suite to fit 8% more design instances into the KU095 device and 27% more design instances in the KU115 device. The result is that the UltraScale architecture enables users to pack significantly more design into a device than the competition.

ASIC-Like Clocking

The UltraScale clocking architecture (Figure 4) has been completely redesigned compared to previous generation FPGAs. There is a uniform matrix of clock routing and clock distribution tracks in both horizontal and vertical directions. The clock routing tracks enable the placement of the center of a clock network in the center of the logic that is driven by that clock signal. The clock distribution tracks then take the clock signal to all of the desired destinations. This structure enables many more clock networks than previous FPGA architectures and dramatically reduces the effect of clock skew on the maximum achievable performance of a design.

All UltraScale FPGAs are divided into clock regions that are of fixed height and width. All regions are 60 rows of CLBs tall with the same geometric width of logic, block RAM, and DSP, resulting in the same time taken for signals to cross each and every clock region. Every clock region has 24 horizontal and vertical routing and 24 horizontal and vertical distribution clock tracks.

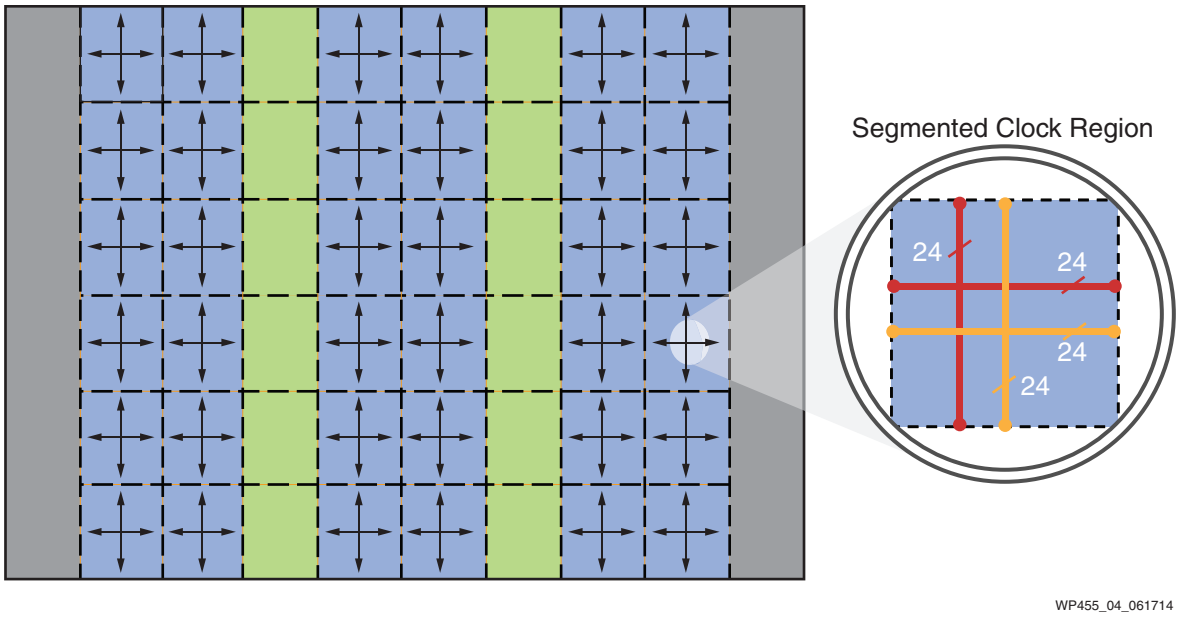

Figure 4: **UltraScale Clocking Architecture**

These clock routing tracks and distribution tracks all connect, so they can be used to drive clocks throughout the entire device but can also be segmented on the clock region boundary. This segmentation means that clock signals are only driven where they are required—just like in an ASIC! An additional benefit to only driving clock signals where they are needed is the reduction of unnecessary transistor switching and, therefore, a reduction in dynamic power consumption.

In addition to the new clock routing scheme, the style and quantity of clock buffers have both been overhauled. The quantity of clock buffers has been greatly increased—up from 32 centrally located global clock buffers to 24 global capable buffers at the junction of every horizontal row and clock management column. This equates to 720 global capable clock buffers in the largest UltraScale device. Coupled with the greater quantity of buffers are the reduced styles of clock buffer. There are fewer buffer types with fewer restrictions than previous architectures, making the task of understanding which buffer to use far simpler than before.

Scaling with UltraScale Architecture

All UltraScale and UltraScale+ devices use the same underlying architecture that provides all the benefits described in this white paper. Whether UltraScale or UltraScale+ devices, the fundamental architecture is the same, allowing any design or IP targeting one UltraScale architecture-based device to be easily re-used on all other UltraScale architecture-based devices. This easy design and IP migration within the UltraScale families is the first step to enabling users to build multiple variants of the same system with the UltraScale architecture.

Equally as important is minimizing the rework to the PCB when switching between programmable devices. The UltraScale families offer a variety of devices in footprint compatible packages. UltraScale and UltraScale+ devices are footprint compatible based on the last letter and number sequence of their package designator. For example, any device in a package ending with A2104 is compatible with all other devices in A2104 packages. This strategy provides package footprint migration between different device families and process nodes.

Optical Transport Network (OTN) transport and muxing applications can illustrate the value of package footprint migration. The demands for intelligent data processing continue to climb at unprecedented levels, fueled by the explosion of social networking and consumer video applications, as well as requirements for highest quality and reliability imposed by enterprise and data center customers. The wired communications infrastructure responsible for delivering data must keep pace with these demands by continuing to multiply resources in a system alone, or by combining more resources with system intelligence in a drive to process data more efficiently.

Xilinx provides numerous OTN SmartCORE™ solutions aimed at enabling Network Equipment Providers (NEPs) to focus their efforts on innovating and differentiating their end product from the competition to best address the increasing performance and reliability demands. The provision of this customizable IP enables users to quickly implement and repeat the building blocks of their system, such as framers, mappers, and forward error correction (FEC) blocks, to create the basic system functionality. The remaining resources within the FPGA can then be used by the NEP to create their desired differentiation. This approach not only saves time developing certain commonly used functions but also allows the NEP to spend their valuable engineering resources on the more challenging and highest return areas of their system.

Figure 5 shows an example of a 2x100G Transponder built in a Virtex UltraScale VU095 FPGA.

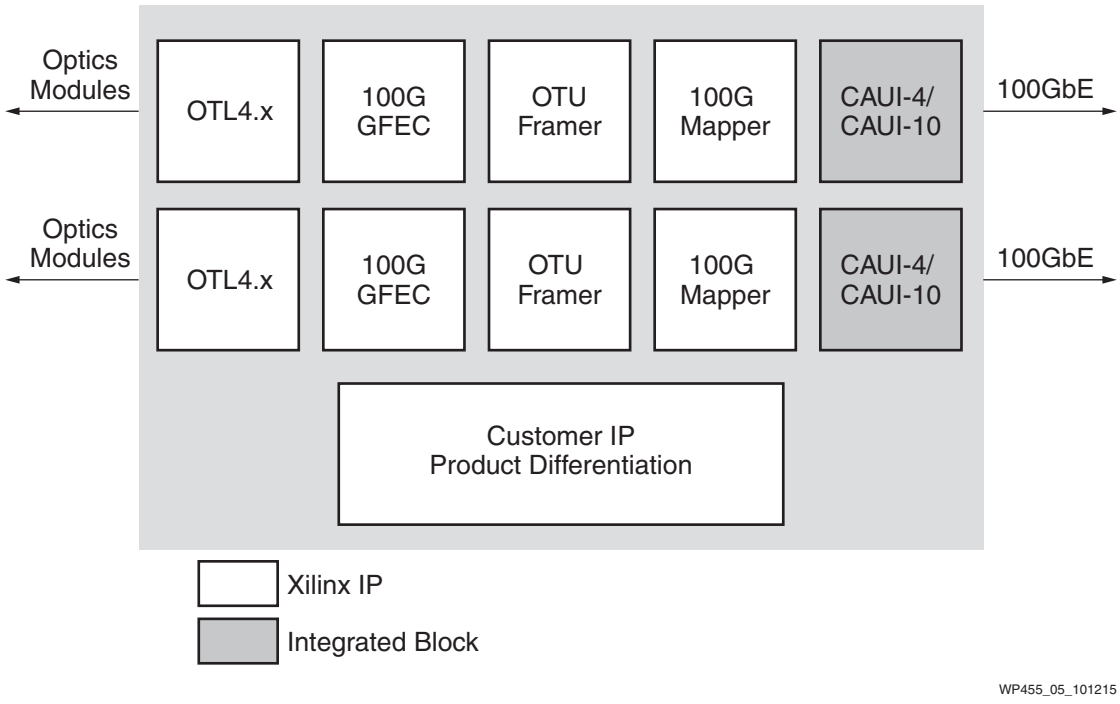

Figure 5: **2x100G Transponder**

As illustrated, many of the system components are Xilinx SmartCORE IP that can easily be implemented and replicated. In addition, this application benefits from the integrated blocks for Ethernet available in the Virtex UltraScale devices, which support 100G Ethernet communication per block, and configurable in CAUI-4 and CAUI-10 modes, depending on the transceiver speed used. CAUI-4 uses four transceivers at 25.78125Gb/s whereas CAUI-10 uses ten transceivers at 10.3125Gb/s to create the 100G Ethernet channel. In addition to the integrated blocks for Ethernet, UltraScale devices include integrated blocks for Interlaken, which can be configured in

transmission speeds from 10Gb/s up to 150Gb/s with various lane and data rate configurations. Both the GTH and GTY transceivers in the UltraScale architecture can support data rates from 0.5Gb/s to 16.3Gb/s, with the GTY transceivers able to run up to 32.75Gb/s. With the same functionality up to 16.3Gb/s, either GTH or GTY transceivers can be used to drive optics modules on OTL4.10 (at 10/11Gb/s) and the GTY transceivers support OTL4.4 (25/28Gb/s).

It is commonplace for equipment vendors to provide multiple variants of a system that scale to offer the end user increased functionality and throughput. Using the same building blocks that were used to build the 2x100G Transponder in the VU095 FPGA, the VU160 FPGA can be used to create a 4 x 100G Transponder, providing a massive 400G ingress and egress in a single device. Not only do all UltraScale architecture devices share the same architectural resources, the package migration within the family enables different devices to be used on the same board with minimal rework. Figure 6 shows the 4 x 100G Transponder built in the VU160 in the same package, with the same footprint, as the VU095.

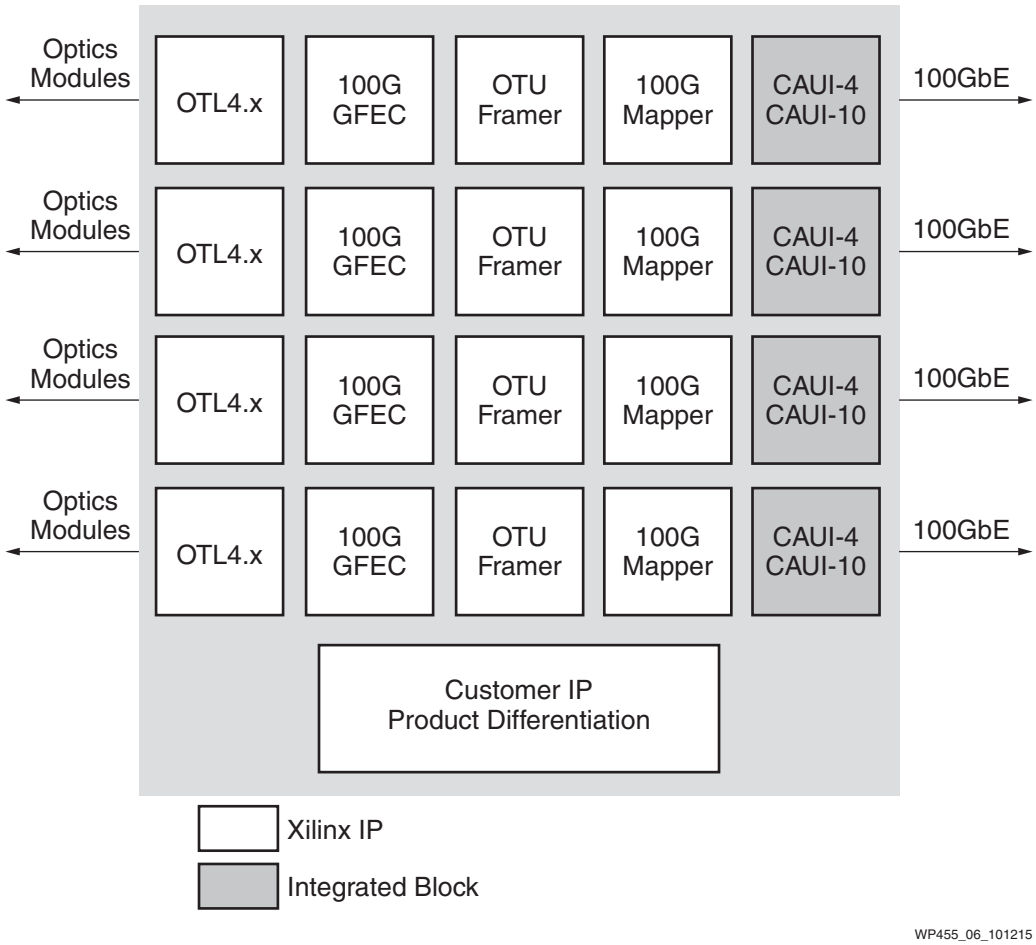

Figure 6: **4 x 100G Transponder**

The implementation of the Xilinx SmartCORE IP consumes approximately 60% of the available logic resources of the VU160 device. Due to the UltraScale architecture enhancements, users can expect logic utilization of 90% or higher, which in this example leaves over 30% of the FPGA available for the NEP to implement their custom IP to differentiate their end product from their competition.

Unprecedented Bandwidth and Connectivity

Virtex UltraScale FPGAs are unrivalled at the high-end, and provide the only solution in the market at 20nm. The Virtex UltraScale family provides unprecedented capability and connectivity with the industry's most transceiver-rich FPGA—the Virtex UltraScale VU190 device, which combines 2.35M System Logic Cells with over 130Mb on-chip RAM, over a thousand parallel I/O pins, and up to 120 serial transceivers, paving the way to 500G systems in a single packaged device. See Figure 7.

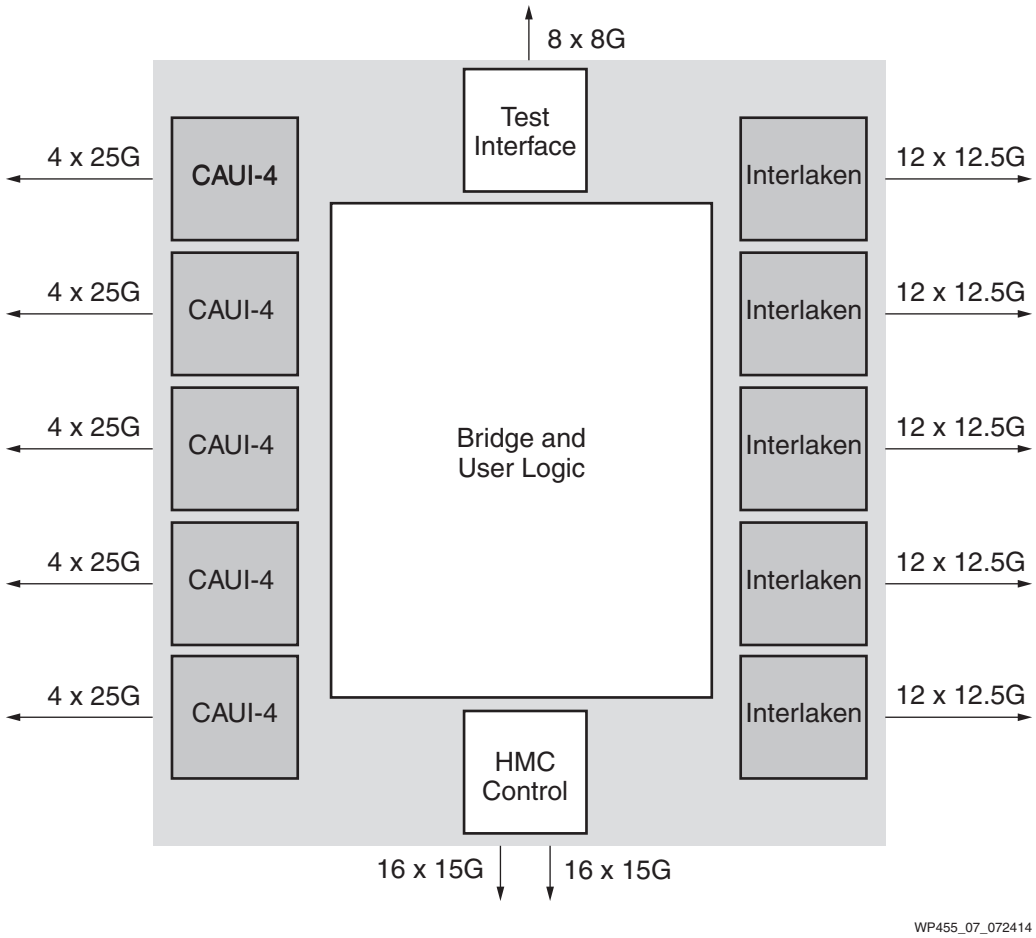

Figure 7: **500G Bridge Application**

Figure 7 shows a 500G bridge application that is enabled by an FPGA with 120 transceivers. Each channel interfaces to the backplane over Interlaken, using twelve transceivers at 12.5Gb/s. On the other side of the bridge, data is transmitted over 100G Ethernet with four 25.78125Gb/s transceivers used per 100G link. In addition to the main bridge application, this bridge must interface to serial memory to buffer data, in this instance connecting two links of 16 channels at the maximum HMC data rate of 15Gb/s. The remaining eight transceivers available on the XCVU190 device are available to the users to implement their desired interface. In this case, the remaining transceivers are used to implement a test interface over PCI Express®, using the integrated blocks for PCI Express running at the Gen3 data rate of 8Gb/s. These eight transceivers could equally have been used to provide further backplane communication over a protocol such as 10G-KR or to communicate with another FPGA in the system over a protocol such as RXAUI.

Due to the ever-present desire to increase bandwidth, there is an inevitable path through 500G to terabit applications. In addition to footprint and architecture compatibility within today's UltraScale FPGAs, there is a clearly defined, simple migration path into the UltraScale FPGAs built on TSMC's 16nm FinFET process, enabling the next generation of highest performance, massive bandwidth applications to start today.

Conclusion

Increasing device size and complexity to meet the demands of next-generation applications results in very large and capable FPGAs. While there are many benefits to doing more in a single device, the challenges of routing wide, fast designs and reusing portions of the design remain. Xilinx addresses these challenges by adopting a new, high-performance architecture across all UltraScale devices, capable of achieving very high resource utilization. Coupled with the architecture and IP migration, package footprint migration enables designers to scale up their application as market needs change.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
10/29/2015	1.2	Updated Logic and Interconnect and Figure 3.
10/15/2015	1.1	Updated Figure 3; Logic and Interconnect; and Scaling with UltraScale Architecture.
08/15/2014	1.0	Initial Xilinx release.

Disclaimer

Automotive Applications Disclaimer