Using External Memory Interfaces to Achieve Efficient High-Speed Memory Solutions

Achieving Lowest System Cost with Midrange 28-nm FPGAs

Meeting the Low Power Imperative at 28 nm

Using Floating-Point FPGAs for DSP in Radar

FPGA Configuration via Protocol

Enabling High-Performance DSP Applications with Arria V or Cyclone V Variable-Precision DSP Blocks

Optimize Power and Cost with Altera’s Diversified 28-nm Device Portfolio

Document Revision History

Table 1 shows the revision history for this document.

Table 1. Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2011</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
This white paper describes some of the components involved in an external memory interface. Altera’s portfolio of 28-nm FPGAs was developed to provide both the highest overall bandwidth of 921 Gbps and a highly efficient solution that allows a designer to get the most effective bandwidth possible.

Introduction

Many FPGA-based systems require an external memory interface. This memory interface often serves as a buffer between the external memory data path, which is often faster than the internal FPGA fabric, and the internal FPGA processing blocks. With the advent of transceiver-based FPGAs, the memory interface has become increasingly important. In order to ensure peak system performance, the memory must be able to store up to hundreds of gigabits of data as fast as the transceivers can provide that data to the FPGA. To keep pace, these data streams require a wide and fast memory interface.

Although I/O performance is important, it is not the complete story on bandwidth. The efficiency of the memory controller can be critically important in determining the actual system bandwidth that can be achieved. This “effective” bandwidth is a critical factor in determining the actual performance of a system.

The Memory Interface

Altera’s external memory interface controller consists of three blocks, as shown in Figure 1. The multiported front end allows multiple processes inside the FPGA to share a common bank of memory, the memory controller implements all of the DDR3 command and addressing, and the physical layer interface (PHY) handles the timing on the data path itself. All three blocks are critical to the design and use of the memory interface block.
The PHY

In the Quartus® II development software version 11.0, Altera introduced a new type of controller to address the increasing speeds of DDR3 memories. The quarter rate controller allows the interface between the UniPHY version of PHY and the FPGA core fabric to run at a quarter of the rate of the DDR3 memory clock speeds. So a 2133-Mbps data interface with a 1066-MHz clock can be captured by the UniPHY and presented to the FPGA core fabric at a clock rate of only 266-MHz. This means that four bits of data would be presented to the core fabric on every 266-MHz clock edge. The quarter-rate controller supports 2T command timing as well. This means that a command is issued every two DDR3 clock cycles. Figure 2 shows a timing diagram of the new quarter-rate controller with 2T command timing.

Figure 2. Quarter Rate Controller with 2T Command Timing

Controller Clock

Controller Command 0

Controller Command 1

Memory Clock

Memory Command

DQ Bus Utilization

In the Quartus® II development software version 11.0, Altera introduced a new type of controller to address the increasing speeds of DDR3 memories. The quarter rate controller allows the interface between the UniPHY version of PHY and the FPGA core fabric to run at a quarter of the rate of the DDR3 memory clock speeds. So a 2133-Mbps data interface with a 1066-MHz clock can be captured by the UniPHY and presented to the FPGA core fabric at a clock rate of only 266-MHz. This means that four bits of data would be presented to the core fabric on every 266-MHz clock edge. The quarter-rate controller supports 2T command timing as well. This means that a command is issued every two DDR3 clock cycles. Figure 2 shows a timing diagram of the new quarter-rate controller with 2T command timing.
In addition, Altera supports full-rate and half-rate controllers, and which controller to use is dependent upon the speed of the interface. Table 2 shows the different DDR3 clock rates and the controller required.

**Table 2. DDR3 Clock Speeds vs. Controller Configuration**

<table>
<thead>
<tr>
<th>DDR3 Clock</th>
<th>Quarter Rate</th>
<th>Half Rate</th>
<th>Full Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1066</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>933</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>667</td>
<td>✔ ✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>533</td>
<td>✔ ✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>✔ ✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>333</td>
<td>✔ ✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Altera is a pioneer in the area of low-latency memory controllers. Altera’s portfolio of 28-nm FPGAs implements a balanced clocked network in the periphery to reduce switching noise, while a hardened read-data FIFO buffer guarantees timing and makes it easier for the fitter to place the controller. These design changes have led to dramatic reduction in latency with the latest release of UniPHY. Table 3 shows the latency based on different command types for both the quarter-rate and half-rate controllers. These latencies are 60% better than the previous generations of UniPHY.

**Table 3. UniPHY DDR3 Memory Latency**

<table>
<thead>
<tr>
<th>Controller Rate</th>
<th>Latency Type</th>
<th>Latency (Memory Clock Cycles)</th>
<th>Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Controller</td>
<td>PHY</td>
</tr>
<tr>
<td>Half</td>
<td>Write command</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Read command</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Read data</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Quarter</td>
<td>Write command</td>
<td>20</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Read command</td>
<td>20</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Read data</td>
<td>0</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 4 shows that these latencies are also dramatically better than those of Altera’s closest competition.

**Table 4. DDR3 Memory Latency—Altera vs. Competition**

<table>
<thead>
<tr>
<th>Controller Rate</th>
<th>Latency Type</th>
<th>Latency (Memory Clock Cycles)</th>
<th>Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Competitor</td>
<td>Altera</td>
</tr>
<tr>
<td>Quarter</td>
<td>Write command</td>
<td>46*</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Read command</td>
<td>46*</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Read data</td>
<td>31*</td>
<td>29</td>
</tr>
</tbody>
</table>

Board skew caused by variation in the traces can be significant when designing a 72-bit or larger DQ bus and can significantly impact the performance of the system. Figure 3 shows how board skew can significantly degrade the data valid window and cause errors on the memory bus. This is especially critical at higher data speeds.
Altera’s UniPHY has configurable delay chains that can adjust the delays of each DQ pin. This adjustment widens the data eye by lining up each of the individual DQ signals, then sending a PBRS pattern to memory and doing both a gross and fine calibration of the bus. This iterative pattern is performed by an embedded soft processor, part of the UniPHY PHY, which continues to skew the lines until it achieves the largest data eye without error. Figure 4 is the resulting eye after deskew.

**Figure 3. Reduced Data Valid Window Due to Board Skew**

<table>
<thead>
<tr>
<th>DQs</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
<th>75</th>
<th>90</th>
<th>105</th>
<th>120</th>
<th>135</th>
<th>150</th>
<th>165</th>
<th>180</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4. Increased Data Valid After Calibration**

<table>
<thead>
<tr>
<th>DQs</th>
<th>0</th>
<th>15</th>
<th>30</th>
<th>45</th>
<th>60</th>
<th>75</th>
<th>90</th>
<th>105</th>
<th>120</th>
<th>135</th>
<th>150</th>
<th>165</th>
<th>180</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**The Controller**

Investing in higher speed memory interfaces means an investment in memory bandwidth. Unfortunately, DDR3 memory, although lower in cost, is not designed to achieve maximum bandwidth, so the efficiency of the controller at managing the required SDRAM commands is critical in achieving maximum bandwidth through the interface. Efficiency is the amount of occupied DQ cycles (non-idle) that occur on the bus divided by the total number of cycles on the bus, or

\[
\text{Efficiency} = \frac{\text{# of clock cycles that DQ bus is occupied}}{\text{# of clock cycles in the period}}
\]
Increased efficiency on the bus can be achieved in two ways. The first is by reordering commands to take advantage of idle or dead cycles. Altera’s advanced bank management recognizes that precharges are required when the system switches banks and places those precharges to take effect during idle cycles. This minimizes the impact of bank switching on the DQ bus by eliminating the dead cycles that can occur between operations. Figure 5 shows the impact of reordering two precharges. In this example the number of idle cycles was reduced by four.

**Figure 5. Advance Bank Management**

<table>
<thead>
<tr>
<th>Command</th>
<th>Address</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Bank 0</td>
<td>Activate Required</td>
</tr>
<tr>
<td>Read</td>
<td>Bank 1</td>
<td>Precharge Required</td>
</tr>
<tr>
<td>Read</td>
<td>Bank 2</td>
<td>Precharge Required</td>
</tr>
</tbody>
</table>

In the case of a page hit, the Quartus II controller also has the ability to automatically cancel the precharge, so there is no need to change banks.

The second way to improve efficiency is to change the data order or commands. Bus turnaround time and the bank cycle time (tRC) can result in large idle cycles that reduce the overall efficiency of the bus. To minimize the bus turnaround time, it is important to group similar commands together. In other words, to make sure read and write commands are issued in groups to minimize the number of bus turnaround events that occur. Figure 6 shows that the 4-cycle time hit caused by bus turnaround time only needs to occur once in the transaction if the system can group the writes and reads together. The overall impact is to remove five dead cycles from the bus.
Figure 6. Data Reordering to Minimize Dead Cycles

Minimize bus turnaround time by grouping read and write transactions

<table>
<thead>
<tr>
<th>Command Address</th>
<th>WR B0</th>
<th>RD B1</th>
<th>WR B0</th>
<th>RD B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Address</td>
<td>WR B0</td>
<td>WR B0</td>
<td>RD B1</td>
<td>RD B1</td>
</tr>
</tbody>
</table>

Minimize tRC impact by reordering transactions with bank conflicts

<table>
<thead>
<tr>
<th>Command Address</th>
<th>WR B0R0</th>
<th>WR B0R1</th>
<th>WR B0R0</th>
<th>WR B0R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Address</td>
<td>WR B0R0</td>
<td>WR B0R0</td>
<td>WR B0R1</td>
<td>WR B0R1</td>
</tr>
</tbody>
</table>

In order to minimize tRC, commands to the same bank are gathered together to remove the four idle cycles it takes to change banks on SDRAM memory. This type of operation results in the removal of eight idle cycles from these transactions. With these types of operation, Altera has been able to gain significant improvements in efficiency in the 28-nm portfolio of devices, as shown in Figure 7.

Figure 7. Memory Efficiency of Altera’s New High-Performance Memory Controller

Hard Memory IP vs. Soft Memory IP

Altera’s 28-nm portfolio of FPGAs provides two types of memory solutions: soft IP, which is provided in the Stratix® V, Arria® V, and Cyclone® V FPGA families, and a hard IP solution, which is provided in the Arria V and Cyclone V FPGA families.

Soft IP consists of the UniPHY and the high-performance memory controller. The hard read/write data paths ensure timing is meet at the highest speeds. Figure 8 shows the hard paths, including the I/Os, the PLLs, the DLL, and the read/write FIFO buffers, in the soft IP.
Altera offers soft IP instead of hard IP to allow the designer to choose where they can place the memory controller and the ability to size the memory controller based on the system requirements, especially in Stratix V FPGAs.

The hard memory controller IP consists of a hard UniPHY, a high-performance memory controller, and a multiported front end. The hard IP has a fixed location on the die and has a fixed maximum width: x16 in the case of Cyclone V FPGAs and x32 in the case of Arria V FPGAs. Furthermore, the hard IP runs at full rate, which allows for decreased latency and minimizes the required bus width of signals to the core of the device. This simplifies the overall memory design in Arria V and Cyclone V FPGAs and provides a truly out-of-the-box experience for the designer. Simply put, it just works.

The MPFE

The hard multiported front end (MPFE) is a new feature in the Arria V and Cyclone V FPGAs. It allows multiple internal processes in the core to access the same hard memory interface. As shown in Figure 9, the MPFE has two types of ports: four bidirectional data ports and six address and command ports. The address and command port supports allow access for up to four bidirectional processes or up to six unidirectional processes. Each of these processes is considered either a read or a write transaction, and the interface to each port is the Avalon® Memory-Mapped (Avalon-MM) system interconnect.
The complete memory controller provides two levels of scheduling for the six address and command ports. Per-port scheduling takes place in the MPFE and DRAM burst scheduling takes place in the high-performance memory controller.

The per-port scheduler in the MPFE decides which of the active addresses and command ports to service next by first dividing the transactions on each port into a series of individually scheduled DRAM bursts. Then these bursts are serviced based on a deficit round robin (DRR) arbitration algorithm. The algorithm uses a per-port absolute priority and weight, which can be updated dynamically, and allows for temporary over and under service so the controller can smooth out memory traffic. A port’s priority is automatically increased if its maximum latency is greater than the worst-case latency allowed in the controller. A port may also be configured for priority to allow multiple-burst transactions to keep a DRAM page open.

**Conclusion**

Altera provides the fastest, most efficient, and lowest latency memory controllers, which allow designers to work with today’s higher speed memories quickly and easily. Soft IP, provided in the Stratix V, Arria V, and Cyclone V FPGAs, gives the designer the flexibility to create interfaces that meet system requirements while benefitting from Altera’s industry-leading performance. The hard IP, provided in Arria V and Cyclone V FPGAs, gives the designer a complete out-of-the-box experience when developing a memory controller, allowing them to get the design up and running quickly without having to worry about the memory interface. Altera understands that a fast and robust memory interface is crucial for many designers. Altera is committed to not only making those designs successful but ensuring that the implementation is both fast and easy.
Further Information

- Arria V FPGAs: Balance of Cost, Performance, and Power:
  www.altera.com/devices/fpga/arria-fpgas/arria-v/arrv-index.jsp

- Video: “Arria V FPGA Sneak Peek: Transceiver Operation at 6.375 Gbps and 10.3125 Gbps”:
  www.altera.com/b/arria-v-fpga.html

- Webcast: “Achieving 1066-MHz DDR3 Performance With Advanced Silicon and Memory IP”:

- White Paper: *Achieving Lowest System Cost with Midrange 28-nm FPGAs*:

Acknowledgements

- Trung Tran, Staff Product Marketing Manager, High-Density Products, Altera Corporation

Document Revision History

Table 5 shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2011</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
As electronic products increase in complexity and capability, the cost to develop and support them also increases from a design and debug perspective, a manufacturing perspective, and an operation and maintenance perspective. This white paper discusses some of the ways in which Altera® Arria® V FPGAs are designed to reduce costs from all of these perspectives.

Introduction

As electronic products increase in complexity and capability, the cost to develop and support them also increases. Longer development time, increased need for design expertise, all costly debug infrastructure all increase the cost of system development. Furthermore, increased complexity adds increased cost in supply chain management and manufacturing, not to mention the increase in operating costs once the system is deployed. To truly save costs, one must evaluate all of these areas early in the design cycle. Altera has focused on this larger cost arena to develop the 28-nm portfolio of FPGAs, specifically in the development of Arria V FPGAs. This white paper discusses some of the ways in which Arria V FPGAs are designed to reduce costs from all of these perspectives.

Reducing Design and Debug Costs

Arria V devices provide several ways for designers to reduce their design and debug costs, including hard IP, intelligent pin placement, proprietary tools and resources, and Altera’s unique Virtual Target.

Hard IP

Arria V devices include a large amount of circuitry dedicated to performing the most common functions in FPGA designs. Arria V devices offer the highest number of these “hard” intellectual property (IP) blocks of any midrange FPGA family, and they serve to significantly simplify the overall design process. One of the most time-consuming parts of FPGA design is achieving timing closure, and the hard IP blocks in Arria V devices ease this task by operating consistently and reliably at the specified maximums—particularly the memory controllers, multiport front end, and memory PHY. These specifications can cause timing closure challenges in FPGAs that do not feature hard IP blocks. Figure 1 shows a block diagram of the hard IP blocks in an Arria V FPGA.
Intelligent Pin Placement

Another way that Arria V devices reduce development time is with their intelligent pin placement, which is designed to ease board layout and maximize signal integrity, and thereby minimize PCB design and debug effort. Figure 2 shows a representative pin layout of an Arria V FPGA on the left side, and expanded detail of the transceiver pin layout on the right side.

The power and ground pins are placed in a regular, repeating checkerboard pattern, which eases PCB layout. As shown on the right side of Figure 2, the transceiver pin placement is regular and repeating along the outside edges of the device, with a minimal number of TX-pair columns, which also eases layout and reduces PCB layers. The RX pins are placed on the outside of the device, which maximizes their signal...
integrity by ensuring that they receive as little interference as possible from possible aggressor signals routed to the device. Finally, the memory interface pins are shielded and placed away from the transceiver pins to minimize the possibility of crosstalk. In these ways, Arria V pin placement reduces PCB layout effort and cost, and minimizes development time.

**Tools and Resources**

Another way in which Arria V devices minimize design and debug effort is with their comprehensive support tools and resources for designing power distribution networks (PDNs). Unlike other off-the-shelf semiconductors, FPGAs have different switching supply current requirements, depending on their application. As a result, the PDN must be designed to meet the needs of the specific application to supply clean power that avoids violating device specifications and other negative impacts like ground bounce.

To aid in PDN development, Altera offers PDN tools and a Board Design Resource Center. The purpose of the Altera PDN tools is to help the design of a robust PDN for the device in the targeted device family by determining an optimum number, type, and value of decoupling capacitors needed for selected device/power rails to meet the desired impedance targets. This spreadsheet tool is useful for exploring the various what-if scenarios during the early design phase, without extensive and time-consuming pre-layout analysis.

For more information, refer to the Board Design Resource Center on Altera’s website.

Altera offers a PDN tool that features each density-package combination of the Arria V FPGA family, ensuring that the exact values for each specific device are included in the calculations to determine the exact PDN characteristics desired by the user. The PDN tool allows the user to select a variety of device characteristics from drop-down menus, and to select the number and values of the capacitors for each power rail. The tool calculates the composite impedance of the PDN along with the impedance characteristics of the voltage regulator module VRM, the decoupling capacitors and their mounting inductance, the PCB, and the FPGA with on-package and on-chip capacitors. By determining the optimal set of decoupling capacitors for a given design in a fast, accurate, and interactive way without requiring a SPICE simulation, the PDN tool eases the board layout process and potentially saves board space, while enabling the user to easily evaluate cost and performance trade-offs. Figure 3 shows a screenshot of an Altera PDN tool.
Virtual Target

Users of Arria V system-on-chip (SoC) devices benefit from a development option unique in the FPGA industry. Software developers can use the SoC FPGA Virtual Target, a PC-based functional simulation of an Altera SoC FPGA development board. The Virtual Target is a binary- and register-compatible functional equivalent of an SoC FPGA development board, that enables embedded software engineers to develop using familiar tools and maximize legacy code reuse, then move their application to the SoC FPGA with minimal effort.

The Virtual Target offers ecosystem tools compatibility and additional debugging capabilities unique to a simulation environment. As a simulation model, the Virtual Target offers more visibility into the system under debug, allows users greater control of the application execution (especially in a multicore system), and performs many debugging tasks that are hard or impossible on hardware. These features and capabilities enable significant productivity gains during embedded software development, which is often the longest and most resource-intensive part of embedded systems projects. Figure 4 shows the Virtual Target with the optional FPGA-in-the-loop extension that allows embedded developers to test their software with Altera FPGA hardware such as custom peripherals and hardware accelerators.1
Lower Manufacturing Costs

Arria V devices reduce manufacturing costs in a number of ways, including hard IP, fractional phase-locked loops (FPLLs), voltage rails, core fabric, and new packaging, all of which mainly stem from the highest level of integration offered in a midrange FPGA.

**Hard IP**

Arria V FPGAs include the most hard IP in a midrange FPGA. The hard IP in Arria V FPGAs was selected to ease the implementation of the most commonly used functions in midrange FPGA applications, maximizing the usefulness of the silicon area they occupy. Using hard IP for these functions reduces their cost of implementation and their power consumption. Table 2 lists the hard IP functions in Arria V devices, and shows the amount of device resources that are saved via a hard implementation versus implementation in FPGA fabric resources.

<table>
<thead>
<tr>
<th>Hard IP Block</th>
<th>FPGA Resources Saved per Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit DDR3/DDR2 memory controller with ECC, command/data reordering, and multiport front end</td>
<td>&gt;40K LEs and 45 M10K blocks</td>
</tr>
<tr>
<td>PCIe Gen1 and Gen2</td>
<td>&gt;10K LEs</td>
</tr>
<tr>
<td>PCIe Multifunction</td>
<td>&gt;20K LEs</td>
</tr>
<tr>
<td>ARM® Cortex™-A9 MPCore™ processor and peripherals</td>
<td>&gt;40K LEs</td>
</tr>
</tbody>
</table>
Figure 5 shows an example FPGA design that uses four memory controllers, two PCIe blocks with multifunctions, and 60K LEs of user logic. The left side of Figure 5 shows that an Arria II device with 260K LEs of user logic capacity (the EP2AGX260) is needed to accommodate all of these functions if they were implemented in the FPGA fabric in “soft” logic. The right side of Figure 5 shows the same design implemented in an Arria V FPGA (5AGXA1) with lower user logic capacity (75K LEs), demonstrating how the hard IP-based architecture of Arria V can reduce overall implementation cost.

Figure 14. Hard IP Blocks Reduce FPGA User Logic Requirements and Lower Implementation Costs

Note: Hard IP block sizes not to scale

**FPLLs**

FPLLs enable Arria V devices to synthesize a wide range of frequencies with high precision. FPLLs accomplish this by leveraging 32-bit M and N values in their feedback path, as well as a δ-σ modulator. With this capability, FPLLs can potentially replace voltage-controlled oscillators (VCXOs) on the board, reducing both board costs and board space.

**Voltage Rails**

As shown in Figure 6, Arria V devices require the fewest number of voltage rails of any midrange FPGA, as few as three in the simplest configuration. Additional voltage rails are required to support I/O standards that are different voltages from the ones shown in the diagram, as is standard with other FPGAs. These minimal voltage rail requirements can reduce voltage regulator requirements, as well as simplify board design and reduce board layers.
Core Fabric

Arria V FPGAs utilize an innovative core fabric to efficiently implement both logic and digital signal processing (DSP) functions. The basic building block of the Arria V is the adaptive logic module (ALM). In this generation of Altera devices, the ALM has two more registers than to the prior generation, increasing the register-to-logic ratio and providing a better fit for the register-rich designs that are common to higher density FPGAs. In addition, Arria V FPGAs feature a new embedded memory block, the M10K. This memory block is smaller than embedded memory blocks in competing architectures, resulting in higher granularity, more memory ports, and fewer wasted blocks overall. Finally, Arria V FPGAs have variable-precision DSP blocks that can implement multipliers of varying precision. This capability enables Arria V FPGAs to deliver the precision multiplier required by the customer’s application, rather than imposing a single predetermined precision upon all users.

New Packaging

Another way in which Arria V devices reduce manufacturing costs is with their innovative new packaging. Arria V devices are the first FPGAs to be offered in thermal composite flip chip ball-grid array (BGA) packaging. Thermal composite packages, shown in Figure 7, provide several cost-saving features compared to the lidless flip chip packages used with other midrange FPGAs.
First, thermal composite packages are designed to use the same familiar handling and heat-sink procedures as metal-lidded packages. The thermal composite material provides a large surface area for ease of adhesion and greater thermal dissipation than lidless packages. A designer can mount heat sinks easily to the top of thermal composite packages in the same way as with metal-lidded packages, without costly handling procedures and PCB attachments. Finally, thermal composite packages are also lower profile than lidded packages, enabling their use in more space-constrained environments.

**Keep Operation and Maintenance Costs to a Minimum**

Arria V FPGAs keep operation and maintenance costs to a minimum via two main aspects: low power consumption and in-field reconfiguration.

**Low Power Consumption**

Arria V FPGAs are the lowest power-consuming midrange FPGAs, with the lowest static power and transceiver power. Power consumption is increasingly a key decision criterion for today’s electronic products and systems, and low power consumption is an attractive way to keep operational costs to a minimum. Figure 8 shows the static power consumption of Arria V FPGAs (blue lines) compared to the power consumption of competing 28-nm midrange FPGAs (red lines). The solid lines show the typical static power consumption of commercial-temperature devices under typical conditions at 85°C, while the dotted lines show the worst-case static power consumption of commercial-temperature devices at 85°C. As the graph shows, Arria V devices consume significantly lower static power, up to 60% less power than competing devices.

**Figure 17. Static Power Consumption of Arria V FPGAs Compared to Other 28-nm Midrange FPGAs**
Arria V devices also consume the lowest amount of transceiver power of any midrange FPGA. Altera’s extensive transceiver design expertise is unmatched in the industry, and this unique advantage is reflected in the low dynamic power consumption of its transceivers. For example, at 6 Gbps, Arria V transceivers consume less than 100 mW of power, significantly lower than transceivers in competing 28-nm midrange FPGAs, as shown in Figure 9. For designs that utilize the up to 36 transceivers available in Arria V devices, the power savings is over 5 W.

**Figure 18. Power-Per-Channel of Arria V FPGAs Compared to Other 28-nm Midrange FPGAs**

Arria V SoC FPGAs offer software-controlled power-down modes, enabling them to operate while the FPGA fabric is powered down. Other modes enable the processor to run in single-core (versus dual-core) operation, or at slower clock speeds. With these and other low-power features in place, Arria V devices consume the least amount of power for midrange applications, as demonstrated by recently published power benchmarks.

**In-Field Reconfiguration**

Another way in which Arria V devices lower maintenance costs is with remote upgrade capability. With this capability, product developers can upgrade their products in the field without resorting to costly “forklift” upgrades, and without requiring maintenance personnel to perform the upgrades locally or manually. Entirely new functionality can be delivered in the field using this feature, which enables product developers to use a platform-based approach to offer multiple products of various capabilities based on a single hardware platform. Altera FPGAs provide a unique feature to facilitate this process: Configuration via Protocol (CvP), which enables configuration of the FPGA over an industry-standard protocol. The first CvP protocol to be supported is PCIe. Using CvP, developers can take control of the FPGA configuration process using a host processor connected to the PCIe bus, and source the configuration information anywhere in their system. In this way, the memory costs of storing the FPGA configuration file can be also reduced, since they can be combined with the host processor’s stored instructions and data.
Integration Example

The application example of an high-definition (HD) IP camera with H.265 encoding illustrates how an application can take advantage of some of the Arria V FPGA’s features to achieve the highest level of system integration and corresponding integration cost benefits. The block diagram on the left side of Figure 19 shows a current implementation of an HD IP camera using a digital signal processor to perform image and signal processing, as well as the H.264 video encoding. This design also includes an FPGA to perform custom HD video analytics. The right side of Figure 19 shows an implementation based on an Arria SoC FPGA, which combines the image and signal processing, an H.265 video coder/decoder (CODEC), wide dynamic range (WDR) processing, and custom HD video analytics in a single device.

Figure 19. HD IP Camera Developmental Evolution

Table 3 shows a breakdown of the FPGA device resources utilized by this design in the top rows, and for comparison shows the device resources available in an Arria V 5ASXB3 SoC FPGA in the last row.

Table 7. FPGA Design Resources Used in the Example

<table>
<thead>
<tr>
<th>Design Module</th>
<th>Logic Resource Requirement (LEs)</th>
<th>Memory Resource Requirement (M10K)</th>
<th>DSP Resource Requirement (18x18 multipliers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD Video Analytics</td>
<td>74K</td>
<td>295</td>
<td>38</td>
</tr>
<tr>
<td>Wide Dynamic Range Signal Processing</td>
<td>95K</td>
<td>335</td>
<td>180</td>
</tr>
<tr>
<td>H.265 CODEC(1)</td>
<td>40K</td>
<td>250</td>
<td>200</td>
</tr>
<tr>
<td>Total</td>
<td>209K</td>
<td>880</td>
<td>481</td>
</tr>
<tr>
<td>Devices Resources in Arria V 5ASXB3</td>
<td>350K</td>
<td>1,729</td>
<td>1,618</td>
</tr>
</tbody>
</table>

(1) Estimated
This level of integration is only possible using an Arria V SoC FPGA, resulting in a smaller form factor, lower power consumption, and fewer support components—all of which contribute to lower overall product cost, as well as lower cost of operation. In addition, all of the image processing or video analytic algorithms can be updated dynamically and remotely while the product is deployed in the field, resulting in lower maintenance costs and longer product lifetimes.

Conclusion

This paper discusses just some of the ways in which Arria V FPGAs are uniquely qualified to reduce overall product development costs from several different perspectives, including design and debug, manufacturing, and operation/maintenance. Altera has invested in lowering the total system cost for designs by integrating features that decrease the need for support components such as oscillators, capacitors and power regulators. Design costs are reduced due to higher integration and increased hard IP with the memory controllers, PCIe blocks, and embedded processors. Finally, by offering a drastic 40% reduction in total power consumption, Arria V FPGAs save significant operating costs, making the devices a cost-effective solution not only at the time of design, but for the lifetime of the product.

Further Resources

- Board Design Resource Center, including the PDN Design Tool: www.altera.com/technology/signal/board-design-guidelines/sgl-bdg-index.html
Acknowledgements

- Martin Won, Senior Member of Technical Staff, Product Marketing, Altera Corporation

Document Revision History

Table 4 shows the revision history for this document.

Table 8. Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2011</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
This document describes how Altera’s 28-nm devices enable product developers to control power consumption in today’s increasingly power sensitive applications.

Introduction

Reducing power consumption in electronic products is no longer just a good idea; for many product developers and manufacturers, it is an essential strategy for gaining competitive advantage in an increasingly power-aware and power-hungry world. Lower power consumption delivers the following clear advantages to designers and end users:

- Enables developers to address more power-constrained or thermally-restrictive markets
- Provides developers more freedom to increase capabilities within the same thermal and power budget
- Lowers operational and material costs and results in more compact products
- Reduces stringent cooling requirements
- Provides social responsibility benefits

Component suppliers must provide developers and manufacturers with the best options to reduce energy consumption and address this power imperative—or quickly find themselves at significant disadvantage. Altera’s latest generation 28-nm devices empower product developers and manufacturers to directly address the power imperative.

Low Power Imperative

The US Department of Energy predicts that global electricity generation will exceed 20 trillion kilowatt hours (kWh) by 2015, and 35 trillion kWh by 2035, as illustrated in Figure 20. (1) Electronic equipment is one of the fastest-growing segments of that increase, driven by elements such as data centers and communications networks. For example, power consumption for servers and other internet infrastructure in the US doubled, from over 20 billion kWh in 2000, to over 40 billion kWh in 2005. (2) Likewise, worldwide power consumption for the same functions has similarly doubled in that same time period, going from over 60 billion kWh in 2000, to over 120 billion kWh in 2005. (3)
Looking forward, data centers in the US are projected to consume up to 100 billion kWh in 2011, and double that by 2020 if historical trends are followed. This rapid increase has real economic impact. By 2015 the energy costs for server operation will exceed the costs of the server hardware—an impact that could significantly impact the economics of the global communications and data center infrastructure.

Data centers account for only part of the rapid increase in electronic product power consumption. Other power consumption increases include the communication networks that transport the data, and the PCs and monitors that often serve as the endpoints for the data streams. Taken together, these elements comprise the information and communications technology (ICT) sector. Multiple industry groups measure the energy consumption of the ICT sector in terms of CO2 equivalent emissions, reflecting the industry’s concern over its role in producing greenhouse gasses. The ICT sector generates over 500 megatons of CO2 equivalents annually—with over 30% of that attributed to wired and wireless communications—and the total is estimated to exceed 1.4 gigatons by 2020 if these trends continue, as shown in Figure 21. (5)

**Figure 20. Forecasted Growth in Global Electricity Generation**

![Forecasted Growth in Global Electricity Generation](image)

**Figure 21. Estimated Distribution of Global ICT CO2 Emissions**

- PCs and monitors (excluding embodied energy) 40%
- Servers (including cooling) 23%
- Fixed-line telecommunications 15%
- Mobile telecommunications 9%
- LAN and office telecommunications 7%
- Printers 6%

Note — This analysis does not include radio-broadcasting equipment or television sets. It is based on a global estimate of 0.9 Gigatonne of CO2 equivalent.

These rapid rises in energy demand have prompted governments and industry to increase energy efficiency. For example, the US government—estimated to be the largest consumer of energy in the USA at an annual cost of over $20 billion—is under direction to reduce power consumption in a number of ways, including only purchasing products that consume 1 watt or less of standby power. (6) Similarly, the European Union has adopted regulations requiring products to draw only 1W to 2 W of standby power, with a reduction to 0.5 W or 1 W by 2013. Similarly, California has adopted regulations requiring televisions to consume 49% less electricity by 2013. (8)

Private companies and industry bodies are also acting to reduce power consumption. For example, Verizon mandates that all new equipment must be at least 20% more energy efficient than its predecessor. (9) According to the industry group GreenTouch™, the global communications network is currently responsible for 300 million tons of CO₂ equivalents, (10) and according to the GSMA consortium for mobile communications, the worldwide mobile communications infrastructure including portable devices produces 245 megatons of CO₂ equivalents. (11)

Both of these organizations propose minimizing the generation of greenhouse gasses by setting specific targets. For example, GreenTouch proposes to increase the energy efficiency of the Internet and other communications networks by 1,000 times and specify the exact means to accomplish this improvement by 2015. (12) Similarly, the GSMA recommends efforts to maintain the current levels of global greenhouse gas emissions from the mobile industry, despite anticipated growth in the number of mobile connections by 70% to 8 billion by the year 2020. This goal requires decreasing the amount of greenhouse gas emissions per connection by 40% by the year 2020, compared to 2009. The same GSMA document predicts that “energy efficiency could save 15% of global emissions in 2020. It is one of the lowest-cost quick return options for cutting emissions.” (11)

### Ideal Low Power Components

With these power reduction goals in mind, product developers must deliver greater functionality while reducing the energy consumption of their products. In addition, power reduction also provides attractive economic benefits for end users. For example, in the case of a service provider who relies on electronic products as part of their business model, reducing power consumption lowers the operational expenditures associated with powering and cooling the electronic equipment. Also, lower overall power consumption can reduce the physical power supply requirements. All of these factors can also minimize the actual footprint of the physical plant, which reduces capital and operational costs. As shown in Figure 22, these end user requirements translate to requirements for the equipment provider, and ultimately for the component supplier.
FPGAs and programmable logic devices (PLDs) are uniquely suited to enable product developers to cost-effectively control power consumption in the following ways:

- Provides the rapid integration of many on-board logic, memory, and processor components into fewer devices or even a single device
- Reduces the total support components, power rails, board space, and associated power required to implement complex electronic systems
- Allows exploration of different implementation approaches and algorithms to fine-tune power consumption

Programmable logic’s flexibility and power advantages make it an attractive choice when balanced against the cost and time-to-market disadvantages of full custom silicon.

**Tailored Power at 28 nm**

At the 28-nm node, Altera devices enable designers to tailor power consumption to specific target markets and applications. Altera’s approach leverages multiple semiconductor processes across its 28-nm product portfolio, as well as product or family-specific architecture optimizations, and hardened IP. As a result, Altera’s 28-nm FPGAs consume up to 40% less power compared to their prior generation counterparts.

**Figure 23** shows three 28-nm process options available from Taiwan Semiconductor Corporation (TSMC), which is the semiconductor foundry chosen by programmable logic vendors at the 28-nm node. Within each of these processes are available a number of transistors with a range of static power characteristics. Those transistors on the left side of the band use less static power, and those on the right use more. There is
also a relationship between the static power consumption and the performance of these transistors. In general, the higher the performance of the transistors, the higher their static power consumption. Altera uses both the 28LP and 28HP processes to provide the widest range of performance and power consumption options for its 28-nm products. The third process option, 28HPL, provides some transistors that draw less static power as indicated by the section labeled “HPL Option,” but significant use of these transistors would result in FPGAs that are correspondingly slower—unacceptably slow for many designers. Accordingly, the 28HPL process for FPGAs requires use of its faster and leakier transistors, reducing or eliminating any static power advantage.

Figure 23. 28-nm Process Options from TSMC

Altera’s devices consume the lowest total power of any FPGAs at the 28-nm node. The exceptional power characteristics of these devices result from a focused effort to reduce power consumption at all levels of product development. This effort begins with the 28HP and 28LP semiconductor processes.

For more information about Altera’s efforts to reduce power in its high-performance 28HP-based Stratix® V device family, refer to the Reducing Power Consumption and Increasing Bandwidth on 28-nm FPGAs white paper.

Unlike the Stratix V family, Altera’s other 28-nm FPGA products—the Cyclone® V and Arria® V families—are designed for applications that do not require the absolute highest performance and bandwidth. As a result, they are based on the 28LP process, which is designed to provide the lowest total power, as described by TSMC:

“The SiON-based 28LP process, the family’s lowest total power and cost-effective technology, is expected to provide twice the gate density, up to 50% more speed or 30-50% lower power consumption than TSMCs’ 40LP technology.”
Other major semiconductor vendors seeking the absolute lowest power at the 28-nm node also choose the 28LP process, as Qualcomm demonstrated when they announced, “Qualcomm’s work with TSMC yielded our Snapdragon™ S4 class of processors, including the Snapdragon S4 MSM8960™, a highly-integrated, dual-core SoC designed to reduce power in cutting-edge smartphones and tablets. The Snapdragon S4 class of processors are manufactured in TSMC’s highly sophisticated 28LP process, enabling Qualcomm to deliver the breakthrough combination of high performance and ultra low power to mobile devices.”

Building on this low-power foundation, Altera took additional steps to reduce static power in its 28LP devices by maximizing the use of transistors that are less “leaky” and therefore draw less static current. In addition, the Cyclone V and Arria V families offer a number of device features that can be disabled, including transceivers, I/O banks, PCI Express® blocks, memory blocks and fractional PLLs. These combined efforts result in devices that consume 70% less static power compared to prior generation FPGAs. For example, the Arria V family offers devices that consume less than 750 mW at 500K LEs—much lower than the static power consumed by the current generation of midrange and high-end 40-nm FPGAs. Even competitive 28-nm FPGAs consume up to 2.6X the static power of Arria V FPGAs. Figure 24 shows the typical static power of the Arria V GX devices in the solid blue line and the worst-case power in the dotted blue line. Similarly, the solid red line indicates the typical static power of competitive midrange 28-nm FPGAs, and the dotted line shows the worst-case power. With these characteristics, Arria V devices have the lowest static power consumption of any FPGAs in their class.

**Figure 24. Arria V Static Power Comparison**

---

**Low Dynamic Power Architecture**

In addition to low static power, Altera’s Cyclone V and Arria V devices deliver low dynamic power to achieve the lowest total power. Altera’s approach to achieving low dynamic power begins with the 28LP process, which targets power-sensitive applications including portable consumer, wireless connectivity, and cellular baseband. Describing their motivation for offering an advanced process designed for lowest total power rather than just static or dynamic power alone, TSMC states:
“(Our) decision to build on proven SiON technology for the 28LPT process is driven by changing wireless and portable consumer application dynamics under unrelenting pressure for products to hit market windows. Consumers a few years ago wanted low-leakage handsets that supported long battery life. Today’s consumers increasingly rely on their wireless devices for Internet browsing, video streaming, music, mobile TV, GPS navigation, along with traditional phone and texting services. Active usage power consumption is now a much larger factor in battery life. SiON gate technology, because of its smaller gate capacitance and therefore lower active power than HKMG (high-K metal gate), provides a solution with lower total power, cost, and risk for power-limited applications.”

The 28LP process features low gate capacitances to reduce active power gate capacitances that are 30% lower than 28HPL. In Cyclone V and Arria V devices, Altera also adopted other strategies to reduce device capacitance, including high reliance on hard IP for memory controllers, PCI Express, and transceiver protocol support to reduce die area and its associated capacitance. Finally, Altera has also made significant optimizations to the basic architectural blocks of the Cyclone V and Arria V devices, as compared to Stratix V devices. These optimizations reduce silicon area and associated capacitance, and tune the 28LP-based device families for the performance required by their target applications. For example, a Logic Array Block (LAB, a collection of 10 Adaptive Logic Modules) in an Arria V device is 40% smaller in die area than the Stratix V LAB. The hard memory controllers featured in Cyclone V and Arria V devices similarly reduce the die area and associated capacitance of the external memory interface function. All of these reductions in device capacitance translate to lower dynamic power via the familiar power equation below, where $C$ represents the capacitance of the switching circuit:

$$\text{Dynamic Power} = \frac{1}{2} CV^2 \times f$$

Altera has also reduced dynamic power in device transceivers. Altera’s extensive transceiver design expertise is unmatched in the industry, and this unique advantage is reflected in the low dynamic power consumption of its transceivers. For example, at 6 Gbps, Arria V transceivers consume less than 100mW of power, significantly lower than transceivers in 28-nm competitive FPGAs, as shown in Figure 25. For designs that utilize the up to 36 transceivers available in Arria V devices, the power savings is over 5 W.

**Figure 25. Transceiver Total Power per Channel at 6 Gbps at 85°C Junction**
Altera’s low transceiver power at the 28-nm node is the result of over a decade of honing and enhancing proprietary architecture. This continuous, extensive experience with advanced transceiver technology is unmatched in the programmable logic industry and results in a historical trend of steady power reduction over time. Competitive solutions have a record of increasing transceiver power with each product generation, as shown in Figure 26, which graphs the power of the transceiver physical media attachment (PMA) across multiple generations of FPGAs.

Figure 26. Historical Trend in Transceiver Power

These static and dynamic power optimizations result in Altera 28LP-based FPGAs that consume up to 40% less total power than the prior generation of devices, with reductions across all areas of power consumption, as shown in Figure 27.

Figure 27. Cyclone V Power Reductions
Figure 28 shows similar results for Arria V devices.

**Figure 28. Arria V Power Reductions**

![Arria V Power Reductions](image)

**Power Optimization through Software Innovation**

In addition to process and architectural innovations, Altera has made many investments in software power optimization in Quartus II software. Power-driven compilation focuses on reducing the design’s total power consumption using power-driven synthesis and power-driven place-and-route. This power reduction method is transparent to designers and enabled through simple compilation settings. The design engineer simply sets the timing constraints as part of the design entry process and synthesizes the design to meet performance. The Quartus II software automatically selects the required performance for each functional block as well as minimizes power through power-aware placement, routing, and clocking, as illustrated in Figure 29.

**Figure 29. Quartus II Design Flow Including Automatic Power Optimization**

![Quartus II Design Flow](image)
The Quartus II software performs various compilation stages to minimize total power of designs. At the synthesis stage, the Quartus II software extracts clock-enable signals for clock gating, minimizes RAM blocks accessed, and restructures logic to eliminate high-toggling nets. At the fitter stage, the Quartus II software localizes high-toggling nets to reduce dynamic power, optimizes logic placement to reduce clock power, and implements power-efficient DSP and RAM block configurations. Finally, at the assembler level, the Quartus II software programs unused circuitry to minimize toggling or power down when possible. The resulting design meets the designer’s timing requirements with the minimum power.

As shown in Table 9, the designer has the option to select different levels of power optimization to meet the design constraints. Selecting the Extra Effort setting offers the greatest power savings at the expense of longer compilation times. Results vary based on design and effort level selected. This feature reduces power without designer intervention, while having minimal impact on design performance. In addition, the power optimization is guided by detailed models of the circuitry and by advanced statistical techniques that estimate which signals are toggling the most often. This information allows Altera to determine power-efficient implementations without extra input from the designer (such as a time-consuming simulation of the design to determine switching rates).

### Table 9. Power Optimization Settings in Quartus II Software

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Not netlist, routing, or performance optimizations are performed to minimize power.</td>
</tr>
<tr>
<td>Normal</td>
<td>Low compute effort algorithms are applied to minimize power through netlist optimizations, as long as they do not reduce design performance.</td>
</tr>
<tr>
<td>Extra Effort</td>
<td>High compute effort algorithms are applied to minimize power through netlist optimizations. Maximum performance may be affected.</td>
</tr>
</tbody>
</table>

### Benchmarks

Altera provides the most advanced, lowest power-consuming FPGAs available in the industry. Benchmarks estimating total power for various representative applications further demonstrate the low-power advantage that Arria V devices deliver over their 28-nm competition. Figure 30 shows the result of some of these benchmarks, all of which are documented on Altera’s wiki site for further study.

**Figure 30. Arria V FPGAs vs. 28-nm Competition on Total Power**
Conclusion

The benefits of Altera’s low-power 28-nm devices include lower product costs, lower or relaxed power budgets, fewer thermal restrictions, the ability to address more markets, and options to increase capability within the same thermal/power budget. With this comprehensive approach to power reduction in 28-nm products, Altera enables designers to address the stringent demands of the power imperative.

Further Information

- Reducing Power Consumption and Increasing Bandwidth on 28-nm FPGAs
- Arria V Power—Altera Wiki:
  www.alterawiki.com/wiki/Arria_V_Power

References

   www.eia.doe.gov/oiaf/ieo/highlights.html
3. US Environmental Protection Agency, *Report to Congress on Server and Data Center Efficiency*, 2 August 2007:
   www.efficient-server.eu
5. SMART 2020, *Enabling the Low Carbon Economy in the Information Age*:
   www1.eere.energy.gov/femp/regulations/EO13221.html.
   www.planetark.org/dailynewsstory.cfm/newsid/49257/story.htm
   www.verizonnebs.com/energy.html
10. Judge, Peter. *Alcatel Lucent Vows To Boost Network Efficiency 1000-Fold*, Eweek Europe, 11 January 2010:
11. Global System for Mobile Communications (GSM) Association, *Green Manifesto*, 1 November 2009:  


Acknowledgements

- Martin S. Won, Senior Member of Technical Staff, Altera Corporation.

Document Revision History

Table 2 shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2011</td>
<td>2.0</td>
<td>- Minor text edits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated Figure 24, Figure 25, and Figure 26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added Benchmarks section and Figure 30</td>
</tr>
<tr>
<td>April 2011</td>
<td>1.1</td>
<td>Corrected minor typographical errors.</td>
</tr>
<tr>
<td>March 2011</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
This document describes the advantages of using floating-point processing in FPGAs for digital signal processing (DSP) in radar applications.

**Introduction**

Modern radar systems process high-frequency signals at over 100 GHz. Modern array radar systems have various modes enabled by digital signal processing, including modes for searching, identification, tracking, targeting, and surveillance. The majority of these radar systems, whether steered mechanically or electronically, now process signals digitally to improve system flexibility with multiple modes using software-driven waveforms.

Many military applications have physical limitations of board space and power consumption. FPGAs offer the best performance and size, weight, and power (SWaP) characteristics for these DSP-driven applications. In particular, Altera® FPGAs provide the following signal processing advantages:

- Efficient floating-point DSP for superior system range and observability
- Parallel DSP processing with expanded memory capacity and I/O bandwidth
- Variable-precision DSP enabling efficient integration near the antenna
- A complete DSP solution unprecedented in the industry
- Low static and dynamic power

In addition to the above list, Altera floating-point FPGAs enable higher-precision processing nearer to the antenna, which improves system dynamic range and reduces losses. Floating-point processing also scales numbers by tracking and moving the binary decimal point in the mantissa with minimal risk of overflow. (1)

**Radar Digital Processing Requirements**

Modern radars calculate an enormous amount of information in real-time. That means they process the information without delays. Real-time processing demands stringent requirements of the signal processing device. These systems also have substantial space, power, and heat requirements. The following technologies support real-time DSP processing:

- FPGAs
- Standalone DSPs
- General-purpose Graphical Processing Units (GPUs)
- Multi-core processors
While all of these devices offer flexible, software-defined digital processing, only Altera FPGAs offer superior SWaP, true floating-point, and parallel signal processing.

A critical element of signal processing is the measure of performance versus power, often measured in GFLOPs per Watt for floating-point operation. Table 11 shows the range of GFLOPs per Watt for various products:

<table>
<thead>
<tr>
<th>Product Type</th>
<th>GFLOPs per Watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-end CPU</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>General-purpose GPU</td>
<td>&lt; 5</td>
</tr>
<tr>
<td>High-end DSP</td>
<td>&lt; 8</td>
</tr>
<tr>
<td>Stratix® IV FPGA</td>
<td>5 – 7</td>
</tr>
<tr>
<td>Stratix V FPGA</td>
<td>12 - 15</td>
</tr>
</tbody>
</table>

Table 11 shows that Altera FPGAs can perform floating-point operations more than 10 times as efficiently as many CPUs. Altera’s 28-nm Stratix V device family is expected to double the GFLOP performance compared to previous generation FPGAs. Designers can extend this efficiency by combining the capabilities of Stratix IV or Stratix V FPGAs with Bittware Anemone DSPs. The combination of Stratix series FPGAs with Anemone DSP is expected to achieve even higher performance, and can use standard ANSI C software code for the digital signal processor with acceleration in the FPGA. Floating-point FPGAs provide the most space and power efficient solution for a given performance requirement.

Real-time DSP solutions must perform thousands of calculations in parallel. Only FPGAs can provide this capability because they provide superior customized memory access (a topic known as data locality) and extremely wide internal bandwidth, unlike hard-coded processors. Modern FPGAs contain 1000s of DSP elements operating in parallel, over 50 Mb of memory on chip, and over 500 Gbps of I/O bandwidth. The DSP elements also include highly efficient features such as pre-adder elements that reduce the elements needed for processing digital filters.

Radar systems must operate with significant dynamic range. The power of the transmitted signal decays with the square of the distance on the way to the target and again on the return, resulting in decay with the 4th power of the distance. Discriminating distant and low-observable targets, without being “blinded” by high intensity, results in exorbitant data widths if implemented in fixed point. Typically single or even double precision floating-point processing is used in at least part of the overall radar processing chain to solve this problem. A very limited number of DSP and multi-core processors have the ability to support these elements of floating-point processing. Only Altera offers true single- and double-precision floating-point methodology in an FPGA.

**Radar Digital Processing Architecture**

Modern radars have an analog interface to the antenna or antenna elements, but the analog signals are converted to digital signals for processing. The receiver typically includes downconversion and beamforming elements, as shown in Figure 31.
The emitter includes pulse generation, beamforming, and digital up conversion. The radar processing element processes information to enhance the signal, remove environmental effects, detect target location and velocity, and perform other tasks such as system control. The following sections describe these elements in detail.

**Figure 31. Typical Digital Radar Architecture**

![Typical Digital Radar Architecture Diagram]

**Beamform Filtering**

In radar systems, beamforming is the forming of beams in a physical direction when radiating or receiving energy, also known as “steering.” Beamforming is a spatial filtering method of gaining higher sensitivity in a direction. On receiving, the beamforming element locates the arrival angle of the signal. This is particularly useful with phased array antennas that have multiple elements, or large antenna arrays, as shown in **Figure 32**.

**Figure 32. Array Antenna**
When radars receive energy at an antenna or array element (that is, antennas), interference energy from unintended targets and the environment can have a detrimental effect on the capabilities of the radar system. To resolve these effects, radar designers create a system that does spatial and temporal filtering. Spatial filtering focuses on the distance and direction in space, while temporal filtering focuses on the time element (or frequency) of the signal.

Beamforming looks at multiple antennas in a multi-channel environment and aligns signal delays along one direction by adjusting the phase and gain, such that signals from the different antennas constructively add in the region of interest, and cancel out in unwanted directions, as shown in Figure 33.

**Figure 33. Beamforming Radar Array Model**

Weights are also applied to control the beam shape, as represented in Figure 34 and Figure 35.

**Figure 34. Beamforming Map for Narrowband Phased Array**

\[
\sum \frac{\alpha_m}{\sin \theta_0} \text{ for sensor } m
\]

- \(\alpha_m\) = amplitude weight for sensor \(m\)
- \(f_0\) = bandpass center frequency, Hz
- \(\theta_0\) = direction of max response
Mathematically, narrowband beamforming corresponds with a FIR filter method. Traditionally, beamforming is a fixed-point processing effort. The use of floating-point FPGAs can simplify the task of converting MATLAB system-level models from floating point to fixed point. Figure 34 and Figure 35 show beamforming for narrowband and broadband applications, respectively.

**Figure 35. Beamforming Map for Broadband Delay-Sum Array**

![Beamforming Map for Broadband Delay-Sum Array](image)

When \( T = \tau \), channels are all time aligned for signal from direction \( \theta \). Gain in direction \( \theta = \Sigma w_m \).

Less in other directions due to incoherent addition.

**Figure 36. Filtered Beam Response**

![Filtered Beam Response](image)

Figure 36 shows an FIR filtered beam response that controls sidelobe levels. In this example, the weights are applied to the beam to determine a 20 degree peak angle.
The time domain analysis works well with a single beam filter, but many arrays filter multiple beams at once. For multiple beams in broadband applications, Fourier transform analysis is shown in Figure 37.

**Figure 37. Fourier Transform Analysis in Broadband Beamformer**

Beamforming in Altera FPGAs

Figure 38 shows a traditional downconversion system, where an incoming signal is shifted to a lower frequency. The incoming signal is mixed with another locally generated frequency creating a heterodyne output that is filtered and down sampled. A heterodyne is the generation of new frequencies by mixing (multiplying) two oscillating waveforms. The baseband data is then ready for signal and data processing at later stages in the system.

**Figure 38. Traditional Downconversion**

![Diagram of a traditional downconversion system]
To work well, digital beamformers should have the following characteristics:

- High-bandwidth input connections that can be targeted at a number of interface types
- A simple architecture that is well suited for real-time filtering in parallel
- Customizable and sufficient memory to locate beam weight data near the processor element in time and/or space
- Customizable time-domain or frequency-domain processing at a data rate sufficient to process all signals
- High-performance in a platform that uses low-power and size
- Wide-bandwidth for continued on-chip processing or high-bandwidth output to migrate data for further processing in the system

The following section demonstrates how the Altera beamformer performs against these needs.

Altera provides a radar front end beamforming design example that helps designers get started with beamform processing. This design example uses the aliased polyphase digital downconversion (DDC) shown in Figure 39, due to its efficient use of resources. This polyphase decomposition is computationally efficient and allows analysis of multiple phases in a signal stream. This method uses aliasing to reduce resources while having the same effect as shifting to lower frequencies before sampling. Finally, the mixer operates at the output sample rate rather than the input rate, again saving resources and power.

The signal flow of the radar front end design example is shown in Figure 39 and Figure 40. In this example the 2.8-GSPS ADC input is converted to 8 phases at 350MHz by the ALTLVDS megafunction. The design first performs 8 to 1 downsampling with the polyphase filters. Then, the spinner and adder allow selection of the desired Nyquist zone. The complex baseband signal is then ready for further processing. FFT analysis is used for conversion to spectral representation. The high speed SERDES can...
be used to channel the large amount of data to another FPGA or a backplane. The polyphase filters, multi-phase NCO, band selection, complex adder, and 1K complex FFT blocks are available in the DSP Builder advanced blockset. This example design can also for the basis of a more complex design on the same chip, perhaps followed by pulse-compression, Doppler formation, or space-time adaptive processing (STAP).

Altera Stratix and Arria® series devices provide the following superior support for high-performance beamformer applications:

- High-bandwidth input connections, including LVDS and high-speed SERDES
- A simple architecture optimized for real-time parallel filtering, including pre-adders for efficient symmetric filtering, the only true floating-point capable DSP solution, and a 64-bit accumulation path for higher precision processing
- High memory and DSP density, including the Stratix V GS device, with up to 55-Mbit of on-chip memory, and over 4000 DSP elements on a single FPGA die
- For applications that require off chip data analysis, these devices offer a number of high-speed memory interface types, including DDR3, QDRII+ and RLDRAM II
- User-configurable logic to perform filtering in the time or frequency domain
- The highest performance to power ratio of any digital processing device available
- High-bandwidth SERDES, LVDS, and general purpose I/O to move data on and off-chip without delays

Figure 40. Radar Front End Beamformer Design Example

Space-Time Adaptive Processing

Radar systems use increasingly complex and high processing rate techniques, such as Space-Time Adaptive Processing (STAP). STAP is an advanced signal processing technique that is used in radar applications to suppress interference by working in both the spatial and time domains. This method improves the detection of slow moving targets that are obscured by clutter or jamming, making it particularly suitable for airborne surveillance, where the search for slow moving targets in severe clutter is a common scenario. The challenge with STAP is that it is difficult to process with low latency. With STAP algorithms, FPGAs reduce system size, weight, and power while reducing calculation latency.

Key components in the STAP algorithm are QR decomposition, as well as forward and backward substitution. QR decomposition is a floating-point matrix inversion operation. Both QR decomposition and substitution are highly iterative and sensitive to numerical effects. With the wide dynamic range requirements in radar, and rounding noise introduced in fixed point processing by other SRAM FPGAs and multi-core offerings, the use of floating-point processing is preferred.
An effective radar must discriminate targets against noise. Figure 41 illustrates three different types of noise. Receiver noise acts as a noise floor, and is indicated in light blue in the diagram. This noise level is determined by the quality of the receiver chain, including the antenna, analog processing, and digital downconversion (DDC). A second source of noise is the clutter, shown in green. Ground clutter is based on reflections off the ground from stationary or slow moving elements. Their Doppler component is thus largely defined by the platform speed. The third source of noise is from jammers. Jammers typically transmit across all frequencies. However, since any one jammer has only one specific location, only one particular angle is affected, as shown in tan color.

Figure 41. Radar with Noise from Clutter and Jamming

STAP processing filters and suppresses clutter and jammers, so that targets can be more easily identified.

There are a number of possible algorithms to perform STAP processing. In selecting a STAP processing algorithm, designers must decide whether to work in the power domain or voltage domain. Both methods involve deriving a noise estimate from surrounding radar cells, and applying the inverse to the cell of interest. Calculating the inverse of the noise estimate requires matrix inversion and back substitution. Highly iterative computations like these are only possible using floating-point processing. In addition, the high number of mathematical operations required can exceed the data processing capabilities of many DSPs. These limitations lead to practical constraints in the design of modern radar applications, limiting the noise suppression performance and sensitivity of a radar system. A parallel processing floating-point FPGA can achieve superior noise suppression and sensitivity than comparable systems.

When applied correctly, STAP is a very challenging algorithm to perform. The benefit of using STAP is an order-of-magnitude sensitivity improvement in target detection. To accomplish this, a developer needs very high processing requirements, low latency, fast adaptation, and very high dynamic range. The following section describes how Altera meets or exceeds these requirements.
STAP Processing in Altera FPGAs

Altera has developed a STAP radar floating-point design example that illustrates how to implement this algorithm in Stratix series FPGAs. This design example demonstrates how high performance floating-point and vector processing are implemented. Altera provides the following support for efficient STAP floating-point processing:

- Floating-point operations supported by the underlying silicon structure
- A library of efficient floating-point elements available to designers
- A design entry tool that allows efficient mapping of algorithms to the silicon structures

The Altera STAP design example demonstrates Altera's floating-point support. This design example is comprised of a realistic set of parameters, including 16 antennas, 16 Doppler bins, 64 target steering vectors, and a pulse repetition frequency of 1 kHz. This translates to a processing speed of 80 GFLOP/s. The entire design example can be implemented on the EP4SGX230 medium density Stratix IV FPGA.

The design example was created using MATLAB and Altera’s DSP Builder advanced blockset. This is a standard Altera design flow that includes the following steps:

1. The entire STAP processing chain is implemented in MATLAB, including stimuli generation and plotting facilities for the results.
2. The data processing chain is implemented using the DSP Builder advanced Blockset.
3. MATLAB/ DSP Builder co-simulation verifies correct operation.

Figure 42 shows a plot generated using the example design. The upper plot shows the signals collected by the uniform linear array (ULA), before STAP is applied. The blue line indicates the location of the target. This plot shows that the target would not be recognized, as the presence of a jammer completely overcomes the system.

Figure 42. Results of STAP Processing on Range
Figure 43 shows the data snapshot, which is dominated by a jammer at 60 degrees. The second subplot displays the weights that are calculated in STAP. The diagram shows that a weight of -80dB is applied at 60 degrees, suppressing the jammer. The yellow line along the clutter ridge indicates a weight of around -30dB to -40dB suppresses clutter. The jammer suppression in Figure 43 is located exactly where the jammer is indicated in Figure 41. The clutter, which was shown as a green diagonal in Figure 41, corresponds to the yellow clutter suppression line in subplot 2 of Figure 43, also a straight line along the diagonal with appropriate scaling.

Figure 43 shows an order-of-magnitude sensitivity improvement in target detection with an Altera Stratix IV FPGA driven by the high dynamic range enabled by true floating-point processing. Radar systems built with Stratix series devices have low latency and can quickly adapt to environmental changes, with increased embedded memory and DSP element density for parallel processing.

In summary, Altera’s STAP design example is a good example of how to move from a complex algorithm to a real hardware implementation. The example uses the Simulink design entry method to unlock the full potential of the underlying silicon structure. This method gives radar system designers access to hundreds of GFLOPs on a single FPGA, which raises radar system performance to new levels.

Other Processing Algorithms

There are a number of other algorithms that are of interest to the radar developer. Constant False Alarm Rate (CFAR) processing is often the first detection decision made in processing. This algorithm uses an adaptive measurement of the noise in the neighboring cells and adaptively adjusts the detection threshold. CFAR maintains the probability of a false alarm at a constant level, even in the presence of noise. Designers can use floating point in conjunction with CFAR algorithms to detect targets surrounded by background clutter, such as a submarine periscope surrounded by a rough sea.
Alternatively, pulse compression is another method that reduces transmitter power while maintaining the desired range resolution. Designers can use floating-point FFTs to improve the filtering capability of the system.

Doppler filtering uses the Doppler Effect to compare the frequency shift of the return pulse with the outgoing pulse. FFT filters sort the target velocity vector toward the radar into bins. Again, floating point helps with the sensitivity of the calculation.

Table 12 and Table 13 show benchmarks for Altera’s floating-point FFT IP core. This IP core implements a true floating-point format that scales each individual number without scaling blocks of numbers or causing rounding errors. Table 12 shows the resource and performance results for a single 1024-point floating-point FFT core in a Stratix IV 4SGX70 device using the Quartus II software version 10.1. Table 13 provides the resource and performance results of fourteen 1024-point FFT IP cores in a Stratix IV 4SGX530 device. Results show that even a dense, large, floating-point design clocks at over 300 MHz. These results are easily replicated using the FFT IP core, or the benchmark design is available from Altera upon request.

In summary, Altera Stratix IV devices can process floating-point operations at a similar frequency to competitive FPGA fixed-point processing.

### Table 12. Resource and Performance Results: One FFT IP Core in Stratix IV 4SGX70 Device

<table>
<thead>
<tr>
<th></th>
<th>Logic Elements</th>
<th>M9K Blocks</th>
<th>DSP blocks</th>
<th>f_{MAX} (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 FFT IP Core</td>
<td>23,722</td>
<td>89</td>
<td>64</td>
<td>315</td>
</tr>
<tr>
<td>58,080</td>
<td>462</td>
<td>384</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41%</td>
<td>19%</td>
<td>17%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 13. Resource and Performance Results: 14 FFT IP Cores in Stratix IV 4SGX70 device

<table>
<thead>
<tr>
<th></th>
<th>Logic Elements</th>
<th>M9K Blocks</th>
<th>DSP blocks</th>
<th>f_{MAX} (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 FFT IP Cores</td>
<td>301,308</td>
<td>1280</td>
<td>896</td>
<td>302</td>
</tr>
<tr>
<td>424,960</td>
<td>1280</td>
<td>1024</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71%</td>
<td>100%</td>
<td>88%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Altera’s Floating-Point FPGAs

The following items are required to implement floating point in a system:

- A silicon structure that supports full floating-point processing
- Floating-point capable tools
- A complete library of efficient floating-point functions

Until recently, most silicon, tools, and IP are not integrated and designers are required to piece all of the elements together. Altera FPGAs are superior to the market leading FPGA because they process true floating-point calculations instead of block truncated floating-point calculations. Additionally, Altera has designed a tool flow and IP library which complements the superior silicon architecture. Altera FPGAs are also better than microprocessors and digital signal processors for floating point since they leverage the natural parallelism of FPGAs.
Altera FPGAs, unlike microprocessors, have thousands of high precision, hardened multiplier circuits that can be used for mantissa multiplication, and also used as high speed barrel shifters. Data shifting is required to perform the normalization to set the mantissa decimal point, and denormalization of mantissas as needed to align exponents. Use of a simple barrel shifter structure to perform this task requires very high fan-in multiplexers for each bit location, as well as the routing to connect each of the possible bit inputs. Altera devices are optimized to solve the high fan-in and routing problems that lead to device resource constraints, slow clock rates, and excessive logic usage in competitive FPGAs.

Altera FPGAs can use larger mantissas than an IEEE 754 representation. This is possible because the variable-precision DSP blocks support 27x27 and 36x36 multiplier sizes, which can be used for 23-bit single-precision floating-point datapaths. Using configurable logic, floating-point mantissa precision can be extended as necessary while still maintaining IEEE754 compliant interfaces. Using a mantissa size of a few extra bits, such as 27 bits instead of 23 bits, allows for extra precision from one operation to the next, and allows for more efficient hardware implementations. For example, a fully parallel vector dot product operation requires a bank of floating-point multipliers followed by an adder tree of floating-point adders. By carrying extra mantissa precision, the logic intensive denormalization and normalization functions associated with floating-point adders are eliminated except for the entrance and exit stage of the adder tree.

### 28-nm Variable-Precision Architecture

The DSP blocks in 28-nm Stratix V and Arria V FPGAs are specifically designed to meet the requirements of next generation radar and electronic warfare systems. Altera’s new variable-precision DSP architecture allows designers to specify the required precision for each part of the design. This results in more efficient utilization of logic and DSP resources, and lower power consumption, while providing higher-precision DSP where it is needed.

In 18-bit precision mode, variable-precision architecture incorporates dual 18x18 multipliers, with optional hard pre-adders. The pre-adders are useful in applications like symmetric filtering, as they can add samples to be multiplied with the same coefficients. In 18 x 18 mode, variable precision supports dual integrated coefficient register banks, and the ability to efficiently implement either direct form or systolic form FIR filters. Efficient complex multiplication, essential for FFT implementation, is also supported.

An asymmetrically sized multiplier can be useful for the complex multiplications used in FFT processing because it provides for fixed-precision coefficients for the complex twiddle factors, while allowing data growth that occurs during processing. In FFTs, data growth occurs at a rate of 1 bit per each radix2 stage.

The Stratix V Variable Precision DSP block has been designed for FFT processing. Two DSP blocks can perform an 18x18 complex multiplier, three DSP blocks can perform an 18x25 complex multiplier, and four DSP blocks can perform an 18x36 complex multiplier. This allows the DSP resources to increase in proportion to the bit precision growth on the data side of the multiplier, and use fixed precision 18-bit twiddle
factors. The result is a highly efficient use of DSP resources which allows designers to trade precision for usage of DSP block resources and associated power consumption at each radix stage of the FFT. Using these modes, the Variable Precision DSP architecture is well suited to perform parallel frequency-domain processing of data from large antenna arrays.

In addition, the Variable Precision DSP block is the first to incorporate internal coefficient storage banks, in either 18-bit or 27-bit modes. This reduces usage of external memory blocks and the required routing of coefficients. It also improves timing closure at high clock rates.

The variable-precision DSP also supports native 27x27 multipliers with 64-bit accumulators, the largest in industry. This provides for higher precision and higher dynamic range signal processing, reducing fixed point numerical processing effects. Hard pre-adders, integrated coefficient register banks, and direct form or systolic form FIR filters are also supported in 27-bit mode.

Larger multiplier sizes are also supported by combining 18x18 and 27x27 multipliers in the variable-precision DSP blocks. This technique allows high performance implementation of 36x36 and 54x54 multiplier sizes. The 27x27, 36x36, and 54x54 multiplier sizes allow efficient implementation of single-precision, single-extended, and double-precision floating point.

**Fused Datapath Tool Flow**

Altera’s high-performance, low-latency, floating-point tool flow is known as “fused datapath” technology, as described in Figure 44. This tool flow allows the designer to build mixed fixed- and floating-point FPGA vector signal processing datapaths. This tool analyzes normalization requirements, and inserts these stages only where necessary. This technique leads to a dramatic reduction in logic, routing, and multiplier-based shifting resources. It also results in much higher f\textsubscript{MAX}, or achievable clock rates, even in the very large floating-point designs.

Because an IEEE 754 representation is required to comply with floating-point standards, all of the floating-point functions support this interface at the boundaries of each function, whether a fast Fourier transform (FFT), a matrix inversion, sine function, or a custom datapath.

A fused-datapath tool flow is likely to produce results different from the IEEE 754 microprocessors approach. The main reason for these differences is that floating-point operations are not associative. Summing the same set of numbers in the opposite order results in various least significant bits (LSBs). To verify the fused-datapath method, the fused-datapath tools allow the designer to declare a tolerance, and to compare the hardware results output from the fused-datapath tool flow to the simulation model results. Altera analyzed the numerical precision of fused-datapath methodology and determined that it is statistically more accurate than IEEE754.
The fused-datapath tool flow is integrated in Altera’s DSP Builder advanced blockset, supported by MathWorks’ MATLAB and Simulink. This method allows easy simulation as well as FPGA implementation of fixed and floating-point designs. Figure 45 illustrates how floating-point complex types—both single- and double-precision architecture—are used in conjunction with fixed-point types.

Figure 45. Floating-Point Design Entry Example
DSP Builder offers a single environment for building mixed floating-point and fixed-point designs. The tool also supports abstraction of complex numbers and vectors, making design description clean and easy to change. Complexity associated with mantissas, exponents, normalizations, and special conditions are abstracted away, similar to a floating-point software flow.

**Floating-Point Function Library**

Math.h functions are simple functions expected in a simple C library—trigonometric, log, exponent, and inverse square root, as well as basic operators such as divide. These functions are supported in the fused-datapath flow, as a floating-point library available for designers.

One of the most common functions requiring high dynamic range is matrix inversion. To accommodate this, the fused-datapath library includes linear algebra support, including the following reference designs:

- Matrix multiply
- Cholesky decomposition (used in matrix inversion algorithms)
- LU decomposition (used in matrix inversion algorithms)
- QR decomposition (used in matrix inversion algorithms)

The DSP Builder tool flow supports complex and vector representation. In addition, fixed- and floating-point operations can be easily mixed within the same design. This is essential for efficiently implementing many of the linear algebra operators in many algorithms used in the next generation radar systems. This also allows rapid design reuse and re-parameterization of vector and matrix sizes. Finally, the comprehensive library support of the fused-datapath tool flow allows customers to build large, complex, and highly optimized floating-point datapaths.

**Summary**

Altera’s floating-point FPGAs provide a superior solution for DSP in radar applications. FPGAs offer better size, weight and power characteristics as a function of performance. This method can help reduce system latency while improving dynamic range and reducing losses. By combining highly optimized silicon features and patented library functions with a floating-point DSP methodology, radar systems developed with Altera FPGAs can achieve the new level of performance required by modern military applications.
Further Information

- Achieving One TeraFLOPs with 28-nm FPGAs

- Implementing FIR Filters and FFTs with 28-nm Variable-Precision DSP Architecture

Acknowledgements

- Ian Land, Senior Manager, Military Business Unit, Altera Corporation
- Michael Parker, Senior Manager, Product Marketing, Altera Corporation
- Volker Mauer, Senior Manager, SSG Engineering, Altera Corporation

References


FPGA Configuration via Protocol

Altera’s new device configuration mode—configuration via protocol (CvP)—can be used with PCI Express® to configure the core fabric of Altera’s 28-nm Arria® V, Cyclone® V, and Stratix® V FPGAs. CvP can reduce product cost and board size, while simplifying the software usage model, and providing robust in-field system upgrade capability. In addition, the autonomous, embedded PCIe IP core helps ensure that designs meet PCIe power-up time requirements, irrespective of the FPGA core fabric configuration time, guaranteeing a wide range of interoperability with various PCIe-based computer platforms.

Introduction

PCIe technology has replaced PCI as the standard control plane interface between processors and the devices that they monitor. Since its introduction in 2005, FPGA designers have made PCIe one of the most widely used interfaces between FPGAs and processors. Today’s FPGAs include embedded PCIe cores that serve as endpoints or root ports.

Until recently the embedded PCIe core could not begin link training and bus enumeration until the FPGA was fully configured. As FPGA configuration times increase with rising device densities, it’s becoming difficult to fully configure the FPGAs within the initialization time required by the PCIe specification.

With the announcement of its 28-nm device portfolio, Altera solves this problem by allowing configuration of the PCIe hard IP separate from the FPGA core logic. This technology also allows designers to configure the core fabric of Altera Arria V, Cyclone V, and Stratix V FPGAs via PCIe. The new CvP device programming method can reduce product cost and board size, while simplifying the software usage model, and providing robust in-field system upgrade capability, as described below:

■ Lower system cost—CvP can eliminate one or more parallel flash devices and possibly an external programming controller device. In addition, CvP allows designers to store FPGA programming files in a CPU memory system attached to the FPGA via a PCIe link. Using this technique, only the FPGA I/O programming and PCIe core parameters are stored in the flash device, requiring a smaller and cheaper flash device.

■ Reduced FPGA resources—Stratix series devices typically require wide, data-path flash devices to store the FPGA programming file. In contrast, EPCS and EPCQ devices supported by CvP require fewer dedicated pins.

■ Power savings—Low-power, temporary FPGA images can be loaded via software control based on the user application profile. This feature is useful in battery powered computers.
Figure 46 provides a simplified representation of the PCIe power-up timing sequence as described in the PCI Express Base Specification 1.0a or 1.1 for Gen1 and PCI Express Base Specification 2.0 for Gen2. The minimum time allocated to device initialization and device training is 200 ms (equivalent to the difference between point 5 and point 1 in Figure 46). The minimum amount of time allocated to device initialization is depicted by the time difference between point 3 and point 2 in Figure 46, or about 95 ms.

**Figure 46. PCIe Power-Up Timing Waveform**

Since FPGA devices pack more logic at smaller geometries, more time is required to program large FPGA core fabric with application-specific content. The total configuration time can exceed 95 ms in large devices. When an endpoint device does not reach L0 within the time allocated to it by the PCIe specifications, the endpoint may not respond to the software configuration access transactions (point 5), and the host CPU may fail to recognize this endpoint. In that case, the host CPU may ignore the endpoint and the system operates without it.

**Autonomous PCIe Hard IP**

In order to circumvent this failure discovery mechanism, Altera’s 28-nm FPGAs support operation of embedded PCIe cores, prior to fully configuring the FPGA core fabric. The 28-nm FPGA-embedded PCIe core always meets the PCIe power-up timing requirements by initializing the embedded PCIe cores and the device I/O ring in less than 95 ms. This separately configurable embedded PCIe IP core is referred to as “autonomous.”

The following sequence defines the CvP initialization period for PCIe:

1. The embedded PCIe core is held in reset by PERST#, and is released shortly before point 3 to start PCIe link discovery and training.
2. The rest of the FPGA core fabric begins programming after the PCIe link completes the training phase and reaches the L0 state.
3. After the embedded PCIe endpoint core reaches the L0 state, the host operating system (OS) starts accessing the PCIe core’s configuration space registers (CSR) to perform configuration write access cycles that are part of the system initialization and discovery process (point 5).

4. In case the FPGA core fabric is not fully programmed with the designer’s application content (in other words, it has not yet reached “user mode”), the autonomous PCIe core responds with configuration retry status (CRS) transactions until the FPGA core fabric is fully loaded.

5. The OS correctly identifies this endpoint and attempts to poll it again until it becomes fully functional.

6. The endpoint is allowed to respond with CRS for one second before the OS determines that the endpoint is faulty. In other words, the time allocated for the FPGA core fabric programming cannot exceed one second in this device initialization mode.

FPGA Core Fabric Programming Across the PCIe Bus

The following section provides details about the use of CvP with PCIe. Designers can use CvP to load the initial FPGA core fabric image via PCIe, and then later modify that core fabric image during run-time to meet application needs. After the PCIe IP core periphery is programmed, the link trains to the corresponding PCIe operating mode.

After the PCIe link finishes training and reaches the L0 state, the host CPU can program the FPGA core fabric image via PCIe.

Only one of the embedded PCIe cores in each 28-nm FPGA is capable of performing CvP, and only when used as an endpoint.

During the FPGA core fabric configuration via PCIe, all non-serializer/deserializer (SERDES) I/O pins are held high by internal weak pull-up resistors. All other high-speed SERDES pins are essentially held in reset during CvP core fabric image loading. These I/O assignments are designed to freeze I/O operations while the FPGA core fabric configuration is updated. When CvP is enabled, the autonomous PCIe core does not respond with CRS transactions, but rather accepts and responds to PCIe configuration and data transactions in order to perform the FPGA core fabric configuration.

Table 14. 28-nm FPGA Configuration Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Status</th>
<th>Data Widths (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active serial (AS)</td>
<td>Existing</td>
<td>1</td>
</tr>
<tr>
<td>Active quad (AQ)</td>
<td>New</td>
<td>4</td>
</tr>
<tr>
<td>Passive Serial (PS)</td>
<td>Existing</td>
<td>1</td>
</tr>
<tr>
<td>8-bit fast passive parallel (FPP) using flash loader</td>
<td>Existing</td>
<td>8</td>
</tr>
<tr>
<td>4-, 16-, or 32-bit FPP using flash loader</td>
<td>New</td>
<td>4, 16, 32(1)</td>
</tr>
<tr>
<td>JTAG-based</td>
<td>Existing</td>
<td>Dedicated JTAG port</td>
</tr>
<tr>
<td>CvP using PCIe</td>
<td>New</td>
<td>1, 2, 4, 8(2)</td>
</tr>
</tbody>
</table>
Notes:
(1) Stratix V devices only.
(2) Number of lanes in the PCIe link (Gen1 x1, x2, x4, or x8; Gen2 x1, x2, x4, or x8)

Figure 47 provides a high-level representation of the Stratix V configuration modes and flash programming methods. To simplify the diagram, all flash modes are combined in the block diagram.

Figure 47. Device Configuration and Flash Programming Modes

- Notes:
  (1) Four possible FPGA configuration modes: (a) Active serial (x1, x4). (b) Passive serial/parallel (x1, x4, x8, x32) using an Altera MAX® CPLD or other logic to read from flash memory and configure the FPGA. (c) JTAG configuration for debug purposes (no need for external flash memory to configure the FPGA). (d) CvP of the FPGA core fabric only. The PCIe hard IP (HIP) and I/O ring are first configured through another method.
  (2) Altera EPCS (serial) or EPCQ (quad) can be directly programmed via the download cable.
  (3) In cases where a MAX CPLD is used to program the flash memory, the MAX CPLD reads from the flash memory and configures the FPGA.

When using CvP with PCIe, the various PCIe IP core parameters, as well as the functionality of the respective high-speed transceivers (SERDES), are initially programmed through one of the existing device initialization modes shown in Table 14. Lower-cost production solutions are possible (but not mandated) through the use of an Altera serial- or quad-flash device that stores only the initialization bits associated with the I/O ring and the PCIe hard IP. Table 15 shows how two bits loaded via one of the device configuration modes (listed in Table 14) determine CvP functionality. The bits are cvp_enabled and full_chip_initialization.

<table>
<thead>
<tr>
<th>CvP Mode Number</th>
<th>CvP Mode Bits</th>
<th>FPGA Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1</td>
<td>CvP is off. Full chip initialization through a standard configuration mode. CvP can not be used to update the FPGA core fabric image.</td>
</tr>
</tbody>
</table>

Table 15. CvP Operating Modes
Table 15. CvP Operating Modes

<table>
<thead>
<tr>
<th>CvP Mode Number</th>
<th>CvP Mode Bits</th>
<th>FPGA Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1 0</td>
<td>CvP is on. Only PCIe hard IP, FPGA I/Os, and transceivers are initialized through a standard configuration mode. CvP initially configures the FPGA core fabric. CvP may also be used to update the FPGA core fabric image.</td>
</tr>
<tr>
<td>3</td>
<td>1 1</td>
<td>CvP is on. Full chip initialization through a standard configuration mode. CvP can be used to update the FPGA core fabric image.</td>
</tr>
</tbody>
</table>

Figure 48 depicts a possible CvP operating mode with a low-cost Altera flash device, and illustrates Mode 2 of Table 15.

**Figure 48. Autonomous PCIe Hard IP Core and CvP with an Altera Flash Device**

Regardless of the bit settings, the embedded PCIe core is always autonomous. In other words, it wakes up and starts link training before the FPGA core fabric is configured. When `cvp_enabled` is on, CvP is enabled after the embedded PCIe core reaches the L0 state. The host CPU then can configure the FPGA core fabric image when instructed by the software application. Figure 49 depicts Mode 3 in Table 15.

**Figure 49. CvP Mode with an Altera Flash Loader**
The PCIe endpoint link operating mode used for device programming is the same mode used by the FPGA target application after CvP is complete. For example, designers can program a CvP PCIe core to operate in Gen1 x4 mode. In that case, the core fabric image is loaded through a Gen1 x4 link. The user application logic that subsequently operates in the FPGA after CvP also uses a Gen1 x4 link. The I/O ring configuration (including SERDES) and the embedded PCIe CSRs content remain unchanged during and after the CvP FPGA image updates. Designs modified to replace the initial core fabric image must maintain the same I/O and PCIe core parameters and functionality used in the initial design image.

The CvP topology is not limited to the basic operating modes depicted in Figure 48 and Figure 49. Figure 50 describes a mixed FPGA programming mode that is also feasible in Stratix V FPGAs. FPGA #1 is programmed via CvP. The embedded PCIe endpoint core and the I/O ring of FPGA #1 are programmed via an Altera serial configuration device (EPCS) or quad configuration device (EPCQ). The FPGA core fabric is programmed via CvP, similar to Figure 48. Subsequent FPGA devices are programmed through the fast passive parallel mode, described in Table 14. A user-designed IP core in FPGA #1 controls the programming of the other cascaded FPGAs (FPGA #2 – FPGA #N).

Figure 50. Mixed-Mode CvP Programming
Another possible CvP topology is shown in Figure 51. Multiple FPGAs are programmed via CvP. All FPGAs interface with the root port behind a PCIe switch, thereby using the PCIe topology to program all the FPGA devices that are attached to the switch.

**Figure 51. CvP with PCIe Switch**

CvP support is also feasible in other PCIe topologies, such as the daisy-chain FPGA topology shown in Figure 52. In this mode, all embedded PCIe cores in the FPGAs are initially programmed by their respective Altera EPCS or EPCQ device. The FPGA core fabrics of the daisy-chained devices are programmed via CvP, and all endpoints and root ports come up in parallel. Each FPGA (except FPGA #N) has a designer-developed IP core in its core fabric that controls the programming of the next FPGA.

**Figure 52. Daisy-Chain Format FPGA Programming with CvP**
The FPGA designs shown in Figure 50, Figure 51, and Figure 52 may have different user application content, and hence different configuration file images. In all cases, the programming files are initially retrieved from the host CPU memory or its file system.

**CvP Benefits**

Programming the FPGA core fabric via PCIe leverages the capabilities of the autonomous PCIe core. Combined together, these features can deliver any or all of the following key benefits to 28-nm designers:

- Lower system cost—CvP can eliminate one or more parallel flash devices and possibly an external programming controller device. In addition, CvP allows designers to store FPGA programming files in a CPU memory system attached to the FPGA via a PCIe link. Using this technique, only the FPGA I/O programming and PCIe core parameters are stored in the flash device, requiring a smaller and cheaper flash device.

- Smaller board space—Parallel flash devices can be replaced by a single Altera EPCS or EPCQ flash device.

- Reduction of dedicated FPGA configuration pins—Stratix series devices typically require wide, data-path flash devices to store the FPGA programming file. In contrast, EPCS and EPCQ devices require fewer dedicated pins.

- No host-CPU downtime during core fabric updates—No need for a host-CPU stall or reboot following core fabric image updates when the FPGA operates in user mode. CvP is just another software application that the CPU can execute.

- User application image protection—Core fabric image copies are accessible only to the host CPU and can be encrypted and/or compressed.

- Simple user software model—This model uses the PCIe protocol and the user application PCIe topology to initialize single or multiple FPGAs.

- Power saving—Low-power, temporary images can be loaded via software control based on the user application profile. This feature can be useful in portable computers that are battery powered.

**CvP Operation**

Figure 53 provides an overview of the main building blocks that support CvP and related interfaces in Altera 28-nm FPGAs.
CvP operation includes the following sequence:

1. I/O pin configurations, including transceiver-block electrical and logical parameters and the embedded PCIe core functionality, are programmed by the FPGA control block.

2. The FPGA control block reads programming data from the serial, quad, or parallel flash devices.

3. The FPGA core fabric is programmed through CvP.

4. The embedded PCIe endpoint buffers the data and sends it to the control block to program the FPGA core fabric.

5. The host CPU views the CvP system as a collection of PCIe CSRs and data registers.

6. The host CPU transmits FPGA core fabric programming data to the embedded PCIe endpoint and the PCIe device passes these data onto the control block, which in turn programs the FPGA core fabric.

7. CvP software monitors the CvP status register to determine if the control block detects any errors, and reacts accordingly.

8. After CvP completion, the PCIe core switches to the functionality assigned by the application logic in the FPGA. As the FPGA operates in user mode, the host CPU software can switch the PCIe core back to the CvP mode via CSR write transactions.
In PCIe terms, CvP support is a Vendor-Specific Extended Capability (VSEC). New Altera-added registers reside in the CSR. CvP writes in these registers and poll status bits to communicate with the FPGA control block. The new set of VSEC registers includes:

- VSEC capability header.
- VSEC length, revision, and ID.
- 16-bit CvP status register—The host CPU monitors this register to know when to start/stop sending data, or when there has been a programming error that should be treated as an uncorrectable error. There are also bits to reflect whether Encryption (AES) and/or Compression (DC) are enabled.
- 32-bit CvP control register—This register provides mode and programming control. The host CPU software driver can set these bits to initiate CvP events.
- 32-bit CvP data register—This register holds the programming data coming from the embedded PCIe core receiver buffer before sending it on the control block.
- 32-bit JTAG Silicon ID—This read-only register returns the FPGA Silicon ID, which can be used by an Altera programming software to make sure it is using the correct programming file.
- 16-bit user device/board type ID—Provides a user-settable value to distinguish between the different FPGAs that need to be programmed in the PCIe topologies, such as those shown in Figure 50, Figure 51, and Figure 52.

The host CPU programs the core fabric image using the PCIe technology’s standard 32-bit memory-mapped I/O (MMIO) or configuration write transactions. As mentioned in a previous section, the I/O ring configuration (including SERDES) and the embedded PCIe CSR content remain unchanged during and after the FPGA image updates. During CvP events, all PCIe base address registers (BARs) are intercepted by the CvP. In normal operating mode, all BARs are available for application use.

PCIe core cold-reset events bring down the PCIe link, but they do not start CvP image-loading events and they do not alter the FPGA core fabric image.

**Software Support**

The CvP functionality is supported by Altera’s Quartus® II development software, by the CvP design flow, and by design examples that demonstrate CvP operation.

**Quartus II Software Support**

The Quartus II software provides CvP support across all supported platforms and operating systems. The Quartus II software generates the respective programming files required to initialize the embedded PCIe cores or program the external flash devices for the CvP system initialization of the FPGAs that reside in the designer’s PCIe topology. The content of the files varies based on the CvP operating modes, as illustrated in Table 15. The Quartus II software also generates one or more separate FPGA core fabric programming files for each FPGA that participates in the PCIe CvP hierarchy.
For example, if a designer uses CvP to configure four FPGAs populated below a PCIe switch, similar to the topology depicted in Figure 51, the Quartus II software generates two types of configuration files: raw binary files and FPGA core images. Four raw binary files are created to program the four EPCS or EPCQ devices of this topology. Each file contains the programming information necessary to configure the embedded PCIe hard IP and I/O rings of its respective FPGA. The EPCS or EPCQ devices are programmed via one of the regular FPGA programming methods.

The Quartus II software also creates one or more FPGA fabric core images per device, depending on the number of different user application variants per FPGA. In total, there are at least four such files (one per FPGA). In theory, there is no upper limit to the number of FPGA core images per device. The FPGA core image files can use raw binary, encrypted, or compressed format. One of the images (called ‘initial’ image) is used to initialize the FPGA upon power-up. The initial image can be loaded via CvP or through one of the other FPGA core fabric programming methods depicted in Table 14. All other FPGA core images can be used to update the FPGA core fabric through CvP. Each of the FPGAs in the PCIe topology has a unique 16-bit user device/board type ID value to help direct the host CPU to program the right device through CvP.

**CvP Design Flow Support**

Designers who want to update the FPGA content through CvP must ensure that the I/O ring and embedded PCIe core parameters and functionality remain unchanged. The FPGA image content update through CvP can be viewed as a simple partial reconfiguration with two partitioned regions, where one region (the embedded PCIe hard IP and the I/O ring) remains unchanged, while the other region (FPGA core fabric) can be updated multiple times. This process entails ensuring that the embedded PCIe parameters remain fixed (including the PCIe hard IP that are not used for CvP), as well as maintaining the same I/O functionality.

Some timing and clocking constraints are introduced in the multiple designs that target a single FPGA to ensure migration from the initial design to the other designs. Additionally, in multi-PCIe core applications, only the CvP-capable PCIe core is guaranteed to remain operational during and after the CvP core fabric image update events.

By combining partial reconfiguration with CvP, designers can keep all other PCIe links operational during and after CvP partial core fabric image update events. Partial reconfiguration via CvP is feasible in Altera FPGAs, but it is outside the scope of this white paper.

Altera recommends a design flow that logically locks the I/O ring and the embedded PCIe hard IP cores, and allows designers to attach them to one or multiple FPGA core fabric design images subsequently loaded via the CvP. Figure 54 depicts the partition between the various building blocks, which are configured just once following device power-up (I/O ring including SERDES and PCIe cores), and the FPGA core fabric, which may be updated by CvP as many times as required based on the user application code that runs on the host CPU.
Design Example Support

The ultimate goal of CvP designers is to initialize and periodically update the FPGA core fabric image in the target systems. From an embedded system designer’s viewpoint, CvP is a software application that runs on top of a PCIe device driver that accesses CvP-capable endpoints in their native OS environment.

Altera provides a clear-text C application program design example that targets PCIe development boards under Windows 7 and Linux. The C program initializes the FPGA core fabric through CvP. Designers can use this C program as a starting point for their own code development.

The C program demonstrates the steps needed to implement the CvP algorithm. The C program design example is applicable in both CvP operating modes (Table 15, Modes 2 and 3), and can be used as a software design example for FPGA core fabric image initialization as well as subsequent core fabric image updates.

Conclusion

The 28-nm device portfolio’s autonomous embedded PCIe core ensures designers that their FPGA meets the power-up time requirements of the PCIe Base, as well as the PCIe CEM specifications irrespective of the FPGA core fabric size and link-operating mode. This feature guarantees wide-range interoperability with various PCIe-based computer platforms.

Altera’s 28-nm FPGA portfolio includes the major CvP feature enhancement that benefits most PCIe-based customer applications. CvP can reduce the product cost, lower the board size, simplify the software usage model, and provide a robust in-field system upgrade capability. CvP enables designers to initialize the FPGA core fabric image and later update it in run time as many times as needed by their applications.
Further Information

- **PCIe Base Specification and PCIe Card Electromechanical Specification:**
  [www.pcisig.com/specifications/pciexpress/base](http://www.pcisig.com/specifications/pciexpress/base)
- “PCI Express bridging options enable FPGA-based configurable computing,” *Programmable Logic Designline*, Mike Alford, Genum Corp., September 8, 2008:
  [www.pldesignline.com/howto/210300269](http://www.pldesignline.com/howto/210300269)
- Stratix V FPGAs: Built for Bandwidth:
- Literature: Stratix V Devices:
  [www.altera.com/literature/lit-stratix-v.jsp](http://www.altera.com/literature/lit-stratix-v.jsp)

Acknowledgements

- Arye Ziklik, Senior Manager, Product Planning, Altera Corporation
- Mario Khalaf, Principal Investigator, Office of the CTO, Altera Corporation
Enabling High-Performance DSP Applications with Arria V or Cyclone V Variable-Precision DSP Blocks

This document highlights the benefits of variable-precision digital signal processing (DSP) architecture in Altera’s new Arria® V and Cyclone® V FPGAs. Altera’s variable-precision DSP block allows designers to tailor the precision on a block-by-block basis, thereby saving resources and power while increasing performance.

Introduction

DSP designs use hundreds or thousands of multipliers as basic building blocks to implement filters, fast Fourier transforms (FFTs), and encoders that digitally process signals. Depending on the specific type of filter required, varying precision levels may be required within a design at each stage of FIR filters, FFTs, detection processing, adaptive algorithms, or other functions. In addition, DSP algorithms with varying precision levels often require precision higher than 18 bits. The following sections discuss the benefits of Altera’s variable-precision DSP architecture available in Arria V and Cyclone V devices.

Key DSP Design Trends

The range of DSP precision requirements varies by application, as shown in Figure 55. Video applications use multipliers ranging from 9x9 to 18x18. Wireless and medical applications push precision requirements even further when implementing complex, multi-channel filters that must maintain data precision after each filter stage. Military, test, and high-performance computing also push the performance and precision requirements, sometimes requiring single- and double-precision floating-point calculations for implementing complex matrix operations and signal transforms.

Figure 55. Applications and Precision Range

Applications Moving to Variable and Higher Precisions
The DSP architecture of the 28-nm Arria V and Cyclone V FPGAs is optimized to support both high-performance and variable data precision that enables area and power efficient implementation of both fixed and floating-point operations.

**High-Precision DSP Applications**

Many cutting-edge applications require high-performance DSP designs that support higher than 18-bit precision, as shown in Figure 56. “Precision” in this context means the size of a multiplier, for example 9x9, 12x12, 18x18, 27x27, and other sizes. More specifically, precision refers to the width of each operand applied to a multiplier.

![Figure 56. High-Performance Applications](image)

Many traditional DSP functions such as FIR filters, FFTs, and custom signal processing datapaths have high-precision requirements. These functions are commonly implemented in military, medical, and wireless systems. When designs require precision higher than 18-bit, designers may implement floating-point signal processing to reach this precision level in high-end designs, such as military space-time adaptive radars and MIMO processing on LTE channel cards. Altera’s 28-nm silicon architecture introduces the industry’s first variable-precision DSP architecture that allows designers to tailor the precision of each DSP block to perfectly suit the application.

**Variable-Precision DSP at 28nm**

The variable-precision DSP block in Arria V and Cyclone V FPGAs allow designers to select from 9x9 precision to implement a video processing design, all the way up to floating-point precision required for advanced radar designs. Designers can individually set each DSP block precision to efficiently accommodate bit growth and required precision increases within the DSP datapath. In addition, the Arria V and Cyclone V DSP block is backward-compatible with all modes supported by Altera’s previous generation 65-nm and 40-nm device families. Figure 57 illustrates the precision ranges supported by a single Arria V or Cyclone V DSP block.
Variable-Precision DSP Blocks

Figure 58 maps the multiplier precision required by various FPGA markets to the supported multiplier precisions in Arria V DSP blocks. The Arria V DSP block natively supports nearly all of precision levels required by these applications. The following sections describe the full-precision, 18x18 with pre-adder mode that is effective in the wireless market.

Figure 58. Precision Requirements and Arria V Precisions

<table>
<thead>
<tr>
<th>Precision Requirements</th>
<th>Industrial Video</th>
<th>Broadcast Systems</th>
<th>Wireless Systems</th>
<th>Medical Imaging</th>
<th>Military Radar</th>
<th>High-Performance Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>9x9</td>
<td>12x12</td>
<td>16x16</td>
<td>18x18</td>
<td>18x18</td>
<td>18x25</td>
<td>18x25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18x36</td>
<td>27x27</td>
<td>27x27</td>
<td>27x27</td>
</tr>
</tbody>
</table>

*Requires additional logic outside of the DSP block to implement*
Variable-Precision Modes
The Arria V, Cyclone V, and Stratix V DSP block are the first to offer two native precision modes, as shown in Figure 59.

Figure 59. Arria V and Cyclone V DSP Modes

The available modes are 18-bit mode, and high-precision mode for 27x27 multiplications. Figure 60 shows the various multiplier precision modes available in the Arria V (and Cyclone V) DSP block. Designers can implement an 18x36 multiplier by using one DSP block plus additional logic outside the DSP block. Similarly, designers can implement a 36x36 multiplier by using two DSP blocks and additional logic outside the DSP block, or a 54x54 multiplier by using 4 DSP blocks and additional logic outside the DSP block.

Figure 60. Precisions Available in Arria V and Cyclone V FPGAs

<table>
<thead>
<tr>
<th>Within 1 DSP Block</th>
<th>Within 2 DSP Blocks</th>
<th>Within 4 DSP Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>Multiplier Mode</td>
<td>Quantity</td>
</tr>
<tr>
<td>3</td>
<td>9x9</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>12x12</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>16x16</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>18x19</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>18x25</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>27x27</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>18x36*</td>
<td></td>
</tr>
</tbody>
</table>

Variable-Precision Efficiency
While the key advantage of variable precision is the ability to take advantage of block-by-block implementation efficiencies, the Arria V variable-precision DSP block also provides the highest number of multipliers of different precisions compared to competing architectures, as shown in Figure 61.
Variable-precision DSP blocks provide significant advantages when implementing multipliers of varying precision. Figure 62 compares an Arria V device of 363 KLEs and 1045 variable-precision DSP blocks, against a Kintex-7 device of 356 KLCs and 1440 DSP blocks. When compared with the Kintex-7 XC7K355T device, the Arria V 5AGXB3 device variable-precision DSP blocks provide a clear advantage when implementing multipliers of different precisions. Nearly across the board, variable-precision DSP blocks provide more multipliers per device.

Although competing solutions may offer a few more multipliers in the 18x25 mode, this mode accounts for only a small portion of actual user configurations. Figure 63 provides a comparison of Cyclone V FPGA multipliers against competitive solutions. In general, the Cyclone V device offers more multipliers of different precisions than the Artix-7. The only exception is in the case of 18x25 precision.
Altera’s DSP block architecture has evolved at each process node over time, as illustrated in Figure 64. The fundamental theme of this evolution is backwards-compatibility and new features that support the next generation of DSP system designs.

Historically, the Arria device DSP block implemented four independent 18x18 multipliers. The Arria II device DSP block continues to support this mode and adds more efficient implementation of eight 18x18 multipliers in “sum” mode via a 44-bit cascade bus. Designers can effectively use this mode to implement common FIR filter structures.

The latest 28-nm, variable-precision DSP blocks in Arria V and Cyclone V devices maintain compatibility with previous generation devices, while increasing capability for higher precision signal processing. The Arria V and Cyclone V DSP block architecture fabric is enhanced to implement the highest performance and highest precision DSP application data paths.
Key DSP Enhancements

Arria V and Cyclone V DSP blocks include the following enhancements:

- Pre-adders
- 18x19 Multipliers
- Coefficient Banks
- Feedback Registers
- Independent Multipliers

The following sections discuss these enhancements in greater detail.

Pre-adders

The Arria V and Cyclone V DSP block is enhanced to include pre-adders to reduce multiplier count in symmetric FIR filters, as shown in Figure 65. These pre-adders accept full 18-bit operands, including sign bits. These pre-adders are referred to as “hard” pre-adders because they are implemented in dedicated hardware resources, rather than as FPGA logic gates.

Figure 66 provides a more detailed view of the hard pre-adders. The next section provides an example application that uses pre-adders in a FIR filter design.
Figure 67 illustrates the use of pre-adders in a FIR filter. Typically designers use pre-adders for building symmetric FIR filters. As the filter data is shifted across the coefficient set, two data samples can be multiplied by a common coefficient due to the symmetry. The pre-adder adds two samples prior to multiplication, which allows the use of one, rather than two, multipliers for every two data samples. Pre-adders reduce by half the number of required multipliers for symmetric FIR filters, and eliminate the need to implement such adders using the logic gates in the FPGA. This technique increases logic efficiency and performance. Designers can use this hard pre-adder as either a dual 18-bit pre-adder, or as a single 27-bit pre-adder, depending on the required precision.

Figure 67. Usage of Pre-adders in Symmetric FIR Filter
18x19 Multipliers
The Arria V and Cyclone V DSP block is enhanced to include an 18x19 multiplier, as shown in Figure 68.

Figure 68. 18x19 Multipliers

Previous generation devices included only an 18x18 multiplier. The 18x19 multiplier accepts 19-bit results from the output of the 18+18 pre-adder. Designers can use the extra bit in each pre-adder operand to represent the + or - sign of each operand. Figure 69 shows a close-up view of the 18x19 multiplier.

Figure 69. Close-up View of 18x19 Multiplier
Figure 70 illustrates an example application of the 18x19 multiplier.

**Figure 70. Usage of 18x19 Multiplier**

Benefit: 18x19 multiplier accepts 18+18 data => full 18-bit add with 19-bit result

* Use memory logic array block (MLAB) for cases where number of coefficients per multiplier > 8

**Coefficient Banks**

Arria V and Cyclone V DSP blocks include a coefficient storage bank that is dynamically selectable on each clock cycle, as illustrated by Figure 71.

**Figure 71. Coefficient Blocks within the DSP Block**

This feature is especially helpful in DSP designs that include FIR filters implemented in hardware using a parallel or partially parallel structure, which often require only a small number of coefficients per multiplier. Altera’s variable-precision DSP architecture provides an internal coefficient bank that designers can set to support 18-bit and higher precision signal processing. In 18-bit mode, the coefficient bank is
configured as two, 18-bit wide register banks, each capable of storing eight coefficients per multiplier. In the high-precision mode, the coefficient bank is configured as a single, 27-bit wide register bank capable of storing eight coefficients per multiplier. The coefficient banks allow designers to select which of the eight registers should be used as a coefficient source for the multiplier for every clock cycle.

Use of the internal coefficient bank eases timing closure complexity and reduces on-chip memory and register resource usage, both of which are critical in DSP designs. Figure 72 shows the coefficient bank in the 18-bit mode and in the 27-bit mode.

**Figure 72. Structure of Coefficient Bank**

![Coefficient Bank Diagram]

Figure 73 shows how a serial filter is implemented, making use of the two 18-bit coefficient banks. The DSP architecture of the Arria V and Cyclone V FPGA effectively supports this type of filter because the coefficient banks, the 18x19 multipliers, and the output register are all contained in one DSP block. In addition, the output can be cascaded to the next block in a sequential chain. Having the coefficient bank inside the DSP block reduces logic and routing utilization, thus improving filter performance.

**Figure 73. Usage of Coefficient Bank in Filter**

![Filter Diagram]
Feedback Registers

Arria V and Cyclone V DSP blocks include feedback registers that can serve as the second stage in a two-stage accumulator comprised of the output register and feedback registers. The relative position of the feedback register in the DSP block is illustrated in Figure 74.

Figure 74. Feedback Register

Figure 75 shows how a polyphase serial filter is implemented, with the feedback register enabled to provide a feedback path. This structure enables two independent serial-filter channels in one single DSP block. Each channel has its own set of input. The feedback path is time multiplexed, allowing processing of the real part and the imaginary part of a complex signal in alternating clock cycles. Only $N/2$ adders are needed because the Arria V and Cyclone V DSP block in 18-bit mode has two 18x19 multipliers per DSP block. This implementation is efficient and saves resources.

Figure 75. Feedback Register Usage
Independent Multipliers

Arria V and Cyclone V DSP blocks include independent multipliers. This means that the output(s) of the multiplier(s) can be routed to the output port of the DSP block directly, without going through any adder. Figure 76 shows two 18x19 multipliers which can be configured to work in the sum mode or independent mode.

Figure 76. Input/Output Ports

Each DSP Block contains two 18x19 multipliers. These blocks can be used as two completely independent multipliers with inputs fed from outside the DSP block, as shown on the left-hand side of Figure 77, or each multiplier having one operand fed from a coefficient bank, and the outputs of the multipliers delivered independently, as shown on the right-hand side of Figure 77.

The output port of the DSP block in Arria V and Cyclone V is 74-bits wide and therefore can accommodate the output of 37 bits of the two independent 18x19 multipliers. This means that all 37 bits from each multiplier are directly accessible on the output port.

Figure 77. Application Example
Altera Floating-Point Precision

Depending on the application, the precision requirement may require that multiplications are performed with single-precision, floating-point multiplications, or double-precision, floating-point multiplications. The Arria V and Cyclone V DSP block is capable of both levels of precision, as described in the following sections.

IEEE Standard 754 floating point is the most common representation of floating-point numbers. In this format, single-precision floating point is 32-bits wide with a 24-bit mantissa, while double-precision floating point is 64-bits wide and has a 53-bit mantissa.

Floating-point computations involve mantissa multiplication and exponent addition. The Altera variable-precision DSP architecture can implement mantissa multiplication for a single-precision, floating-point number using one block OR mantissa multiplication for a double-precision, floating-point number.

Single-Precision Floating-Point Multiplication

Using the high-precision mode, the variable-precision block is uniquely suited for implementing single-precision, floating-point operations. Mantissa multiplication can be implemented using only one variable-precision block configured in the high-precision mode. This resource efficiency is an FPGA industry first. Traditionally designers had to cascade multiple blocks to implement this operation. The coefficients may be applied externally as shown on the left-hand side or internally as shown on the right-hand side in Figure 78. Competing DSP architectures with 18x25 bit resolution require multiple blocks, as well as external logic to implement a floating-point mantissa multiplication, resulting in a lower performance and higher power implementation.

Figure 78. Single-Precision Floating-Point Multiplication
Double-Precision Floating-Point Multiplication

Double-precision mantissa multiplication requires four DSP blocks all cascaded by using the dedicated 64-bit cascade bus in the DSP block, as shown in Figure 79.

Figure 79. Floating-Point Modes

This technique is an FPGA industry first, because competing architectures require cascading two 18x25 blocks for single-precision, floating-point mantissa multiplication and up to nine blocks (with extra logic) to implement a 54x54 double-precision mantissa multiplier.

Competitive Summary

With the introduction of the variable-precision DSP architecture, Altera has opened a DSP technology gap against competing architectures, as summarized in Figure 80. Altera’s latest 28-nm devices can natively, and within a single block, implement a 27x27 multiplier useful for high-precision, fixed-point DSP, or for emerging floating-point DSP applications. Variable precision means that designers set the DSP architecture precision to match the algorithm, not the other way around. Also with a 64-bit cascade bus and accumulator, designers don’t have to forgo precision when the algorithm implementation requires multiple DSP blocks.
Conclusion

Altera’s variable-precision DSP block allows the designer to tailor the precision on a block-by-block basis. For symmetric filters, hard pre-adders in the DSP block reduce the required multiplier count by 50%, thus saving resources and power. The 18x19 multipliers accommodate full 18+18 addition, including sign bits. Internal coefficient banks enable higher multiplier performance and save logic resources.

The Arria V and Cyclone V DSP block is optimized for FIR filters, and the feedback register allows implementation of two independent serial-filter channels per DSP block. The independent multipliers allow operands to be applied directly to the multipliers and allow the multiplier outputs to be observed directly on the DSP block output port. Finally, Altera offers the industry’s first floating-point function in an FPGA architecture.

Further Information

- Arria V FPGA Family Overview
- Arria V Device Family Advance Information Brief
- Cyclone V FPGA Family Overview
- Cyclone V Device Family Advance Information Brief

Acknowledgements

- Pat Fasang, Senior Member of Technical Staff, DSP Marketing, Altera Corporation
This white paper describes the power and cost advantages of Altera’s diversified 28-nm FPGA portfolio that is tailored for a wide range of applications. This portfolio includes the Stratix® V, Cyclone® V, Arria® V, and HardCopy® V devices.

Introduction

Altera’s 28-nm devices provide a diversified product portfolio tailored to support a broad spectrum of applications, ranging from low cost and power, to high-end applications. This diversification extends beyond simple size and density options, offering developers devices with the right combination of price, performance, functionality, and power consumption required by their designs. FPGAs are the technology of choice for many development teams, and Altera continually evolves FPGA technology to reduce cost and power, while increasing performance. Altera’s versatile FPGAs are supplanting ASIC and ASSP approaches as a major component in system designs, and as a multi-function adjunct to the system CPU.

Today’s system application designers face increasing requirements for higher integration and performance, along with lower power and cost. Simply designing FPGAs to the next incremental progression may satisfy mid-range application needs, but would compromise performance at the high end and cost at the low end. To meet the needs of the widest application range, the FPGA must take greater incremental leaps. To meet the widest application requirements, Altera’s 28-nm portfolio leverages significant advancements in the following areas:

- Process Technology
- Transceiver Design
- Product Architecture
- System IP

Leaping Beyond Incremental Progression

Altera has taken an evolutionary leap by creating a diversified portfolio of product options at the 28-nm process node. The 28-nm process brings inherent improvements to Altera’s FPGA performance.

Greater circuit density reduces the cost for each functionality, while smaller transistors reduce power. But unlike competitors “one type fits all” process technology, Altera offers two distinct 28-nm process options. Altera’s 28-nm high-performance (HP) process optimizes for performance, while the low-power (LP) process minimizes power for a given functionality or application.
Altera’s 28-nm HP process is for designs requiring the fastest possible logic speeds. High-K metal gates produce logic blocks and DSP functions that are as much as 35% faster than other process options. High-performance transistors use a strained-silicon structure in which a cap covers the channel that induces mechanical strain in the silicon lattice, as shown in Figure 81. This architectural feature increases carrier mobility and thus switching speed, without increasing leakage current. This results in delivery of 28-Gbps transceivers that operate at 200-mW total power, the highest performance at the lowest power available.

Conversely, the LP process targets applications that require minimal device power consumption. The LP process provides minimum leakage and reduced dynamic current through the use of longer gate channels and other techniques. The LP process also uses more conventional metallization than the HP process to further minimize cost and power. As a result, the LP process achieves 50% lower power than the previous process node.

**Tailored Transceivers Provide Performance Options**

Altera’s versatile approach also provides performance options to tailor the serial transceivers. If an application such as I/O expansion requires only 5-Gbps transceivers, the design does not require the power and cost of larger transistors required for operation at 28 Gbps. Rather, the design requires a transceiver that simply meets performance requirements at the lowest power and cost.

Altera’s 28-nm portfolio introduces a modular transceiver that enables designers to match device performance with the application. This transceiver design uses the same base architecture to produce multiple variants targeting from 3-, 6-, 10-, 14.1-, and 28-Gbps operation. Further, the transceivers allow dynamic selection from among several speed settings within their ranges, drawing less power with reduced speed. This selectability provides a method to reduce average system power consumption by operating transceivers at minimal speed when idle, ramping to higher speed only as needed for time-critical data transfers.
These transceivers support a wide variety of protocols, including 3G SDI, 1-, 10-, and 100-Gigabit Ethernet, Fibre Channel, Infiniband®, Interlaken®, PCI Express® (PCIe®), SATA, Serial RapidIO®, and SONET.

For more details about Altera’s 28-nm transceiver technology, refer to the *Extending Transceiver Leadership at 28nm* white paper.

**Versatile Architecture Reduces Cost and Power**

Altera incorporates a number of architectural enhancements into its 28-nm devices to increase versatility and reduce power. These enhancements include support for smart power options, dynamic reconfiguration, variable-precision DSP, and support for various memory protocols and I/O standards.

**Take Control of Power**

Smart power features can reduce average device power consumption by controlling FPGA operation in specific blocks. To completely eliminate power consumption in inactive logic blocks, designers can use multiple power planes to ‘turn off’ individual logic blocks when not needed. When circuits must remain powered even when idle, the available block-oriented clock gating can help reduce dynamic power consumption. The power controller can use the gating to slow or stop the clock to specific blocks when their function does not need high performance or can be suspended.

For more information, refer to *Reducing Power Consumption and Increasing Bandwidth on 28-nm FPGAs*.

**Minimize Downtime with Dynamic and Partial Reconfiguration**

The partial and dynamic reconfiguration capability in Altera’s 28-nm FPGAs increases versatility while reducing cost. Unlike previous FPGAs which must be fully configured before running, Altera’s 28-nm devices allow in-circuit reconfiguration of some FPGA blocks, while others continue operating normally. Such reconfiguration permits an FPGA’s functionality to change in response to system needs without stopping system operation. Rather than requiring a large FPGA to store all functions and then switch among them, designers can target a smaller FPGA and load functions as needed.

For more information, refer to *Increasing Design Functionality with Partial and Dynamic Reconfiguration in 28-nm FPGAs*. 
Leverage Variable-Precision DSP

Altera’s innovative DSP design allows designers to independently configure the precision of each DSP block in a device—ranging from 9x9 to 27x27 bits—to implement required higher packing or precision. For example, simple video processing may require only 9-bit precision, while a high-end color system may require 24 bits. In the case of 9-bit video, a single block can fracture to support three, 9-bit multipliers, tripling the DSP block efficiency. A single variable-precision block can efficiently address this full range, thus allowing designers to adapt FPGA resources to their algorithms rather than adapting the algorithm to fixed resources.

Variable-precision DSP also supports bit growth of FIR filters and FFTs in the design, by allowing successive stages to increase precision as needed, thereby maximizing resource utilization. Altera’s DSP blocks provide the industry’s only 64-bit cascade bus and adder for combining individual blocks to achieve even higher precision.

For more information about Altera’s variable-precision DSP, refer to the *Accelerating DSP Designs with the Total 28-nm DSP Portfolio* white paper.

Altera’s 28-nm portfolio supports three different types of on-chip memory for various applications. For example, the M20K memory block is optimized for high-performance operation and high bit density required by 100G and packet processing applications that require large memories with high, raw bandwidth. The M10K block has a lower bit density but offers more ports per silicon area, making it suitable for DSP-intensive applications such as motor control, studio equipment, and 3D TV that would not fully utilize an M20K block. Most applications also require small buffers which would waste 90% of the M10K block’s capacity. For efficient and low-cost handling of shallow buffers and delay elements, Altera offers the 640-bit MLAB block.

Interface with Application-Specific I/O

Altera’s high-performance I/O blocks for memory interfaces and LVDS signaling are ideal for high-end wireless, radar, and 100G systems. However, this high performance I/O is unnecessary for mid-range applications, such as remote radio units and broadcast equipment. Instead, designers can use Altera’s various I/O architectures tailored for distinct application classes. The high-end I/O supports 1066-MHz DDR3 DIMM memory and 1.4-Gbps LVDS, for applications such as 100GbE switches. The mid-range I/O block handles 533-MHz DDR3 memory and 1.25 Gbps. The low-cost I/O block is well suited for 400-MHz DDR3 and offers 3.3V-I/O with 16-mA drive for industrial applications.

Hard IP Increases Efficiency

Hard IP implementations first appeared in Altera’s 40-nm devices as PHY layer elements that eliminate the need for external devices in high-performance serial I/O. In Altera 28-nm devices, Embedded HardCopy blocks provide a measure of ASIC cost, performance, and power characteristics without compromising design flexibility, as shown in Figure 82.
Incorporating HardCopy blocks in the FPGA frees the device’s programmable resources for custom circuits while reducing cost. For example, a PCIe protocol stack requires 150 K logic elements as a soft implementation, but requires as little as one-third the die area in HardCopy blocks. The hard IP also improves performance. Compared to soft implementations in the device’s LEs, hard IP for the same functionality uses up to 65% less power while offering up to 50% higher performance.

The HardCopy blocks include support for common functions—such as memory controllers, PCIe stacks, and Ethernet interfaces—to address the broadest range of application requirements. Furthermore, these blocks support a degree of configurability. For example, designers can configure a PCIe HardCopy block for Gen1 or Gen2, and an Ethernet block for 40G or 100G operation, as required.

Altera can also use Embedded HardCopy blocks to rapidly implement emerging, application-targeted device variants. The ability to expand support in current-generation devices reduces the need for device migration, thus reducing designer’s time-to-market.

**Chose Application-Tailored Solutions**

To provide the most versatility, Altera’s 28-nm portfolio includes devices with various combinations of process, architecture, transceiver, and system IP features. Within each family are variations for different transceiver speed grades and hard IP block resources that allow designers to target devices that closely match their application requirements. The following sections describe these combinations of features.
Cyclone V Lowest Cost and Power

Many applications have only modest performance requirements but face severe cost, power (typically less than 5 W), and space constraints. For example, handheld projectors must draw the lowest possible power to maximize battery life. Motor controllers, displays, and software-defined radios have both power and space constraints. Each application also has other specific requirements. For example, devices such as a WDR surveillance cameras require low data-rate transceivers to send signals off-unit, while night vision goggles require internal video processing and buffering.

Altera’s Cyclone V family—the lowest cost and power option available in Altera’s 28-nm portfolio—targets such applications. Cyclone V devices are built on the LP process and save cost by using wire bond packaging. The Cyclone V family includes the following variants:

- Cyclone V GX devices offer 3-Gbps transceivers, a hard IP PCIe Gen 1 x 4 interface, M10K and MLAB memory blocks, and variable-precision DSP. A hard IP external memory controller supports low-cost, low-power memories such as mobile DDR, LPDDR2, and 400-MHz DDR3.

- The Cyclone V GT devices offer the same features as Cyclone V GX, with the addition of 5-Gbps transceivers and two PCIe Gen 2 x 1 hard IP blocks.

- The Cyclone V E series includes no hard IP blocks, but is composed solely of LEs for maximum design flexibility.

Arria V Balanced Performance, Power, and Cost

For mid-range applications, the ability to balance performance, power, and cost is critical. Devices such as remote radio units, broadcast video cameras, 10G/40G line cards, and video switchers need higher performance than those targeting low-cost devices, with 10-Gbps transceiver requirements common. Nevertheless, such devices must remain as inexpensive as possible, while using less than 10 W of power. Common functions for this application space include video processing and buffering, FIR filters, and higher-performance external memory.

Altera’s Arria V family offers the balanced performance, power, and cost required by mid-range applications. Arria V devices use the LP process technology to minimize power. These devices include M10K/MLAB memory blocks, while offering faster transceivers to handle more demanding performance requirements. Arria V devices also use flip-chip packaging to maximize off-chip signaling speeds. The Arria V family offers the following variants:

- Arria V GX devices include 6-Gbps transceivers, a hard IP PCIe Gen 2x4 interface with multi-function support, variable-precision DSPs optimized for FIR filter applications, and a hard IP memory controller that supports DDR3 SDRAM at 533 MHz.

- Arria V GT devices offer 10-Gbps transceivers for applications requiring slightly more speed. Logic capacities range from 75,000 to 500,000 LEs.
Stratix V Highest Performance and Bandwidth

Once considered outside the reach of FPGAs, high-performance applications such as 40/100GbE switches, military radar, and advanced LTE base stations are among the most demanding. These systems require transceivers that operate at the highest possible performance for backplane and chip-to-chip communications and dense, high-speed external and internal memory. These applications often call for high-precision DSP, along with logic in excess of 350-MHz clock speed. While power may not be the primary concern in these applications, reduced power carries significant benefits in terms of lower system cooling costs and greater system reliability.

Altera’s Stratix V devices meet the demands of these high-end applications—offering the highest performance at the lowest power in their class. Stratix V devices use the HP process technology to offer transceiver speeds up to 28 Gbps, requiring only 200 mW. A soft memory controller configurable for a wide range of memory types (including DDR3, RLDRAM II, and QDRII+) up to 72-bits wide handles speeds up to 1066 MHz.

Available in logic capacities up to 1.1 million LEs, the Stratix V family offers the following variants:

- The Stratix V GX devices incorporate 14.1-Gbps transceivers that support serial backplane and optical communications and include hard IP PCIe Gen 3 x8 and 40/100G Ethernet IP options.
- The Stratix V GT series is similar to Stratix V GX, but also offers 28.2-Gbps transceivers for highest performance applications. The GT series includes 14.1-Gbps transceivers, but also incorporates variable-precision DSP blocks for up to 54x54 operation.
- The Stratix V E series offers only LEs to provide the highest capacity of configurable logic available. All Stratix V devices include M20K memory blocks for the highest performance and density.

Quick HardCopy Path to ASICs

In addition to the three FPGA families, Altera offers the 28-nm HardCopy V ASIC family that provides a low-risk, low-power, and low-cost path for taking FPGA designs to volume production. The resulting ASIC uses the same process technology and has the same hard IP blocks as the FPGA, including transceivers, thus reducing performance differences in migration. The HardCopy ASIC is package- and pin-compatible with the original Stratix V design, and offers the same signal integrity. The use of HardCopy V ASICs enables developers to deliver products significantly earlier than other ASIC methodologies.

Conclusion

Altera’s versatile 28-nm device portfolio precisely matches designer’s requirements, while minimizing cost and power. In addition, Altera offers an integrated set of design tools that help empower all device features. Altera’s Quartus® II design software includes system integration and power analysis tools to help achieve the right balance of performance and cost. Altera offers a variety of IP cores for quick and easy implementation of standard functions. Designers can use the Quartus II software to rapidly migrate to a HardCopy V ASIC implementation.
Altera’s 28-nm device portfolio represents a significant leap forward, making FPGAs the ideal solution in an ever-widening application range. By embracing process, architectural, and IP diversity, Altera provides an integrated set of FPGA options that accurately match the various cost, performance, and power requirements. This offering provides an unprecedented array of application developers with a rapid, low-risk approach to designing and producing next-generation products.

Further Information

- White paper: *Accelerating DSP Designs with the Total 28-nm DSP Portfolio*  

- White paper: *Extending Transceiver Leadership at 28 nm*  

- White paper: *Increasing Design Functionality with Partial and Dynamic Reconfiguration in 28-nm FPGAs*  

- White paper: *Reducing Power Consumption and Increasing Bandwidth on 28-nm FPGAs*  

Acknowledgements

- Juwayriyah Hussain, Sr. Product Marketing Engineer, Altera Corporation
- James Adams, Corporate Marketing, Altera Corporation
- Umar Mughal, Product Marketing Manager, Altera Corporation

Document Revision History

Table 1 shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2011</td>
<td>1.1</td>
<td>Minor text edits</td>
</tr>
<tr>
<td>May 2011</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>