WHITE PAPER

FPGA

Enabling Impactful DSP Designs on FPGAs with Hardened Floating-Point Implementation

Hardened floating-point DSP implementations enable algorithmic performance and faster time to market compared to traditional implementations.

Authors

Udayan Sinha
DSP Product Marketing Specialist
Intel Programmable Solutions Group

Introduction

Intel changes the game for floating-point digital signal processing (DSP) implementation in FPGAs with the Intel® Arria® 10 and Intel® Stratix® 10 families. The dedicated hardened floating-point blocks in these devices are capable of high-performance floating-point operations that are compliant with the IEEE 754 single-precision floating-point standard. This technology enables up to 1.5 TeraFLOPS performance in Arria 10 devices and up to 10 TeraFLOPS performance in Stratix 10 devices, delivering floating-point DSP performance that is unrivalled by any other FPGA vendor to date.

With hardened floating-point DSP implementation, FPGAs can now be used in an expanding range of computationally intensive applications, such as high-performance computing (HPC), radar, and medical imaging. Examples of such complex applications are shown in Table 1.

We take MIMO processing in wireless systems as an example of a complex DSP application that becomes realizable on an FPGA with hardened floating-point operators. MIMO processing involves multiple transmit and receive paths where copious amounts of matrix processing, specifically matrix multiplication and vector dot products, can ultimately result in a high dynamic range for the numerical outputs. While a large percentage of MIMO processing involves a 2 x 2 configuration (i.e., 2 transmit and 2 receive paths) and is easily represented in fixed point, the requirements are expanding to 8 x 8 or greater for smart antennas. Once the design requirements exceed the 2 x 2 configuration, predicting the numerical dynamic range becomes difficult. Therefore, floating point becomes the only option to accurately and reliably implement complex MIMO systems.

Application	Example
Radar	Doppler
Space-time adaptive processing (STAP)	
Seismic modeling	Reverse time migration (RTM)
Medical imaging	Back projection
Financial	Black Scholes
Monte Carlo	
Wireless	Multiple input, multiple output (MIMO) processing

Table 1. Example Complex Applications

Table of Contents

Evolution of floating-point implementation in Intel FPGAs

Let's take a step back and look at the significant advancements Intel has made to ease floating-point DSP implementations and costs. These advancements span different FPGA generations.

First, consider a basic floating-point operation, such as an add function, implemented in FPGAs using a generic approach. Adding two floating-point numbers involves a preliminary denormalization step which shifts the smaller of the two numbers until both exponents match. Figure 1 shows the preliminary steps for floating-point addition that includes the denormalization step.

The result of the addition will be normalized and rounded to return it to its form of $1.xx \times 10^x$. To comply with the IEEE 754 standard, the denormalization and normalization steps occur after every operation.

In prior FPGA families, this is where a tremendous performance bottleneck occured. These normalization and denormalization steps are traditionally implemented in hardware via large barrel shifter circuits within the FPGA fabric that are up to 48 bits wide for a single-precision floating-point number. A barrel shifter requires several layers of small multiplexers, and separate shifters are required for both the denormalization and normalization steps. It is common to see a single-precision floating-point adder requiring upwards of 500 lookup tables (LUTs), with the barrel shifters consuming almost 30 percent of this total LUT count. More complex mathematical functions, such as exponents and natural logarithms, can require approximately 1,000 LUTs.

As these barrel shifters are built outside of the DSP block of an FPGA, they will also consume valuable routing resources. Due to these factors, FPGA performance degradation will become glaringly visible as the complexity of the DSP algorithm grows. FPGA designs that approach 80 to 90 percent logic utilization will suffer from tremendous routing congestions or limited access to the fastest FPGA interconnects. Ultimately, timing closure would be negatively impacted.

Fused datapath design flow

In 2010, Intel introduced the fused datapath design flow in DSP Builder for Intel FPGAs. This behind-the-scenes backend optimization identified the bit growth generated by an algorithm and automatically reduced the number of denormalizations and normalizations required. Instead of treating the datapath as an aggregate of elementary IEEE 754 operators (each with their respective denormalization and

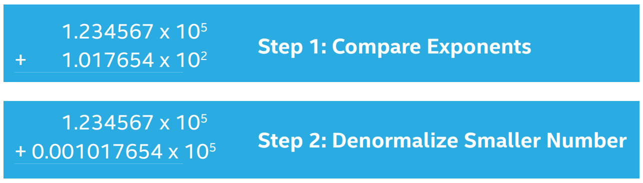
1.234567×10^5
$+ \ 1.017654 \times 10^2$
Step 1: Compare Exponents
1.234567×10^5
$+ \ 0.001017654 \times 10^5$
Step 2: Denormalize Smaller Number

Figure 1. Preliminary Steps for Floating-Point Addition

normalization functions), the fused datapath approach reduced portions of the algorithm into individual floating-point operators and expanded the mantissa widths to better accommodate bit growths. A combination of fixed-point DSP blocks and programmable soft logic was the platform for this optimization. This effectively minimized barrel shifting requirements through the various stages of the datapath and eliminated 50 percent of logic and latency, compared with a datapath using several elementary IEEE 754 operators. The approach also led to IEEE 754 representation at the boundaries of the datapath that were overall more accurate than standard IEEE 754 operator libraries.

The fused datapath design flow enabled performance and energy efficiency that were, prior to hardened floating-point DSP blocks in Intel Arria 10 and Stratix 10 devices, unprecedented from any other FPGA vendors. Table 2 shows the results from a Cholesky-based matrix equation solver of the form A*x* = B using the fused datapath design flow in DSP Builder for Intel FPGAs. These results can be replicated by using the Cholesky-based solver reference design that is available with DSP Builder for Intel FPGAs. The performance and power measurements were taken from the design running on the DSP Development Kit, Stratix V Edition that features a Stratix V GS device.

To note in this particular Cholesky-based solver design example is the use of complex data types for input matrices and how the algorithm itself calls upon high-order floating-point mathematical functions, namely square root and inverse square root. Historically, these mathematical functions have been difficult to implement in FPGA hardware due to their large sizes and long latencies, but the DSP Builder for Intel FPGAs with fused datapath was able to provide outputs at only 3 to 4 times the logic cost of a basic floating-point multiplier, while also producing one result per clock cycle.

With these advancements leveraging Intel's DSP block architecture and software tool optimizations, the groundwork for high-performance floating-point DSP implementation on an FPGA is in place. Intel's Generation 10 hardened floating-point DSP blocks in Arria 10 and Stratix 10 devices build upon this industry-leading floating-point solution.

Input Matrix Size (Complex)	Number of Channels	Vector Size	Throughput per Core (Matrices/Second)	Power per Core (W)	% ALUTs	% Memory Blocks	% 27 x 27 Blocks
30 x 30	64	30	472K at 250 MHz	7.7	22	39	9
60 x 60	20	60	119K at 235 MHz	13.6	41	47	17
240 x 240	1	60	3.3K at 200 MHz	14.0	45	90	17

Table 2. Results for Cholesky-based Solver Design Example Using Fused Datapath

Benefits of hardened floating-point implementation

The most significant benefits of hardened floating-point implementation on the Arria 10 and Stratix 10 devices are as follows:

- Improved algorithmic performance
- Faster time to market

Improved algorithmic performance

For the following discussion, the floating-point algorithmic performance will be made up of the following metrics: the drastic reduction of logic costs resulting in improved algorithm f_{MAX}, the numerical accuracy of the results, and energy efficiency.

Lower logic utilization

With the hardened floating-point DSP blocks in Arria 10 and Stratix 10 devices, FPGA systems can overcome many of the performance-restricting challenges mentioned earlier. The new architecture eliminates 100 percent of the IEEE 754 floating-point single-precision logic usually implemented using FPGA resources. First, all necessary barrel shifting requirements are captured within the hardened DSP blocks. This eliminates the need to create denormalization and normalization logic using valuable FPGA resources and allows the datapath to close timing with an f_{MAX} ranging up to 450 MHz in Arria 10 devices.

Note: Minimal logic usage is still required for the floating-point implementation to account for the algorithm control path and required memory addressing.

Second, the hardened floating-point DSP blocks internalize and eliminate the circuitry required for IEEE 754 single-precision processing, such as:

- Exponent comparison, to compare exponents of the inputs
- "Sticky" bit handling, which helps reduce rounding errors

Furthermore, to comply with the IEEE 754 standard, the additional circuitry required to account for exceptions is also implemented by the hardened floating-point DSP blocks. The exceptions include situations such as positive/negative infinity resulting from a divide by zero, or asymptotic functions such as tan(π/2). Also, inputs or outputs that produce an undefined number, or NaN, which are encountered for operations such as the square root of a negative value are handled. In addition, any operations involving all zeros in the exponent and mantissa fields can also be automatically accounted for.

By minimizing this overall floating-point logic usage, timing closure or f_{MAX} requirements are no longer restricted to sub-optimal routing. When in use, the blocks are optimized and fully characterized for predictable performance, power, and timing closure.

With this novel floating-point architecture, designers can ensure FPGA devices running at 80 to 90 percent logic utilization will maintain the high f_{MAX} performance that would be evident at lower logic utilizations.

Numerical accuracy

The hardened single-precision DSP mode, shown in Figure 2, supports many complex floating-point arithmetic operations. Some of the key floating-point arithmetic modes are shown in Table 3.

Mode Name	Mathematical Function
Multiplication Mode	X ´ Y
Adder or Subtract Mode	(X + Y) or (X-Y)
Multiply-Add/Subtract	(X ´ Y) + Z or (X ´ Y) - Z
Multiply Accumulate Mode	(X ´ Y) + Acc or (X ´ Y) – Acc
Vector One Mode	(X ´ Y) + Chain In

Table 3. Key Floating-Point Arithmetic Modes

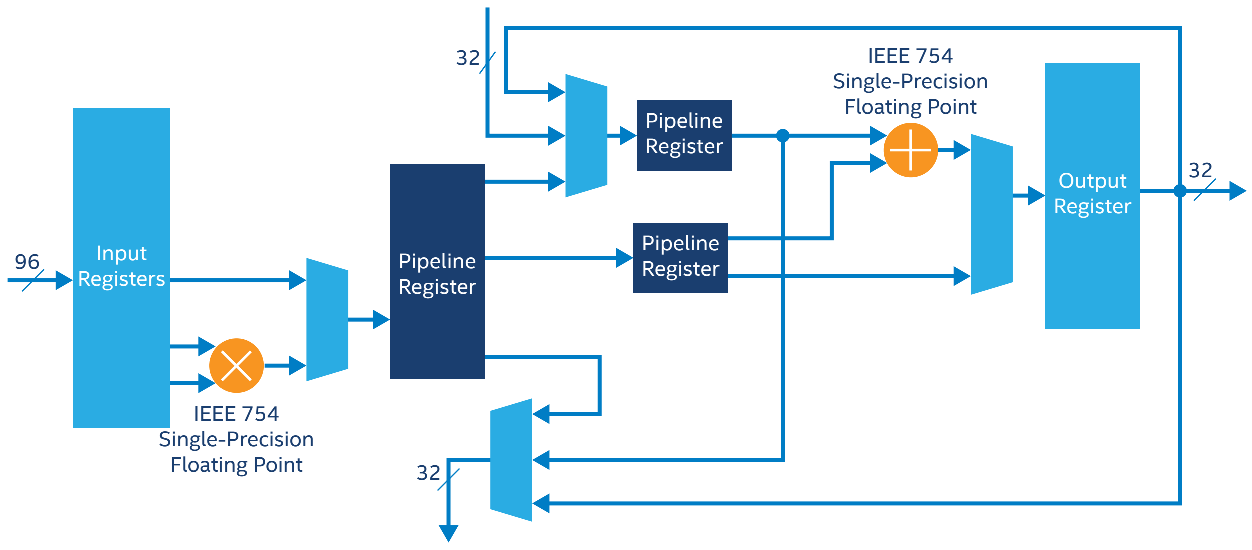

Figure 2. IEEE 754 Single-Precision Floating-Point Mode

As the hardened DSP blocks and the associated modes conform to the IEEE 754 single-precision specification, the floating-point outputs from the blocks match the IEEE 754 standard at every stage of the designer's datapath. This compliance guarantees numerical consistency for applications with the strictest numerical resolution requirements.

Past FPGA implementations used an internal two's complement representation within the datapath. This internal two's complement representation gets translated to and from IEEE 754 format at the boundaries of the algorithm. While this methodology was crucial in reducing the bottlenecks from the barrel shifter logic, it also introduced small amounts of deviation from the golden-standard MATLAB/Simulink models. However, designers can expect hardware outputs with zero deviation from a single-precision Simulink model when utilizing the hardened floating-point blocks in Arria 10 and Stratix 10 devices.

Highest energy efficiency

The Arria 10 and Stratix 10 devices also provide the FPGA industry's highest floating-point energy efficiency, with 50 GFLOPS and 100 GFLOPS per watt respectively. Drastically reducing the logic and routing required for previous floating-point implementations drastically reduces the core dynamic power consumption for DSP algorithms.

Faster time to market

As many common DSP modeling and simulation environments are natively in floating point, an ideal design flow would consist of seamless hardware targeting of floating-point design models without requiring added engineering time for creating an optimized implementation. Arria 10 and Stratix 10 devices give designers this efficient design option. Intel has seen an average of 6 to 12 months of reduced design time due to the elimination of the extra implementation steps with end applications ranging from radar to communication systems in the military space.

No conversion process required

In previous-generation FPGAs, high-performance floating-point hardware implementation was not instantly possible and often involved the need to convert the algorithm to a fixed-point implementation that was optimal for FPGA architectures. This conversion process typically begins with several tedious steps, such as creating a testbench for identifying the discrepancies between the two models, identifying the maximum and minimum data values through the path, identifying the points of possible overflow and underflow conditions, and choosing the appropriate fixed-point fraction lengths to most accurately capture the floating-point data. This conversion process to fixed point can introduce artifacts in the data, which is common when real-world floating-point algorithms are forcibly mapped to finite, fixed-point word lengths. For algorithms particularly sensitive to these miniscule errors, it becomes essential to validate the stability and other characteristics of the system after the process is complete.

In this process, an experienced engineer must manually convert the simulation model while maintaining the functionality and numerical integrity of the algorithm. Any subsequent changes, such as tweaks to the algorithm late in development or errors during the original conversion, required subsequent iterations. Debugging this conversion process had to be accounted for at all times in the design cycle.

Easy-to-use design tools

Intel's premier DSP design tools include DSP Builder for Intel FPGAs for hardware and model-based engineers, and the SDK for OpenCL™ § for software programmers. These tools allow designers to eliminate the floating-to-fixed conversion process entirely, and consequently, the need to debug this step in the implementation process. Intel's hardened floating-point blocks are mapped automatically when designing algorithms using DSP Builder for Intel FPGAs or the SDK for OpenCL and instantiating floating-point data types in the software. For lower-level designs, individual FPGA IP functions and megafunctions are also available. Intel's design tools automate the optimization and usage of the floating-point blocks while abstracting away hardware-centric design concerns, such as the conversion to fixed point and the knowledge of block topologies, pipelining, and time-division multiplexing.

Conclusion

The hardened floating-point DSP implementation in the Arria 10 and Stratix 10 devices builds on Intel's fused datapath design flow, providing a new realm of processing capabilities for floating-point designs. The Arria 10 and Stratix 10 devices provide the highest floating-point performance, energy efficiency, and accuracy in the industry, while reducing an average of 6 to 12 months of development time by eliminating the need for designers to convert their floating-point designs to fixed point. In addition, Intel's provides multiple tool flows that allow hardware engineers, model-based engineers, and software programmers alike to easily target the high-performance floating-point DSP blocks in the devices.

§ OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

Where to get more information

For more information about Intel and Arria 10 FPGAs, visit **https://www.altera.com/products/fpga/arria-series/arria-10/overview.html**

[1] http://www.altera.com/b/wp-fp-performance-claims.html
[2] http://www.altera.com/literature/po/bg-floating-point-fpga.pdf