

Due to recent technological developments, high-performance floating-point signal processing can, for the first time, be easily achieved using FPGAs. To date, virtually all FPGA-based signal processing has been implemented using fixed-point operations. This white paper describes how floating-point technology on FPGAs is not only practical now, but that processing rates of one trillion floating-point operations per second (teraFLOPS) are feasible—and on a single FPGA die.

Introduction

Altera's 28-nm Stratix® V FPGAs family enables much higher levels of both fixed- and floating-point digital signal processing (DSP) than ever before. A key aspect of the FPGAs is Altera's new variable-precision DSP architecture, which efficiently supports both fixed- and floating-point implementation.

However, FPGA resources and architecture are not enough. Verilog and VHDL have poor to basically non-existent support for floating-point representation, and there are no synthesis tools available today which support floating-point operations. Additionally, the traditional approach that is used for floating-point processors will not work with FPGAs. Instead, Altera has developed a new "fused-datapath" toolflow designed specifically to build floating-point datapaths while taking into account the hardware implementation issues inherent in FPGAs. This design tool allows designers, for the first time, to create high-performance floating-point implementations of large FPGA designs.

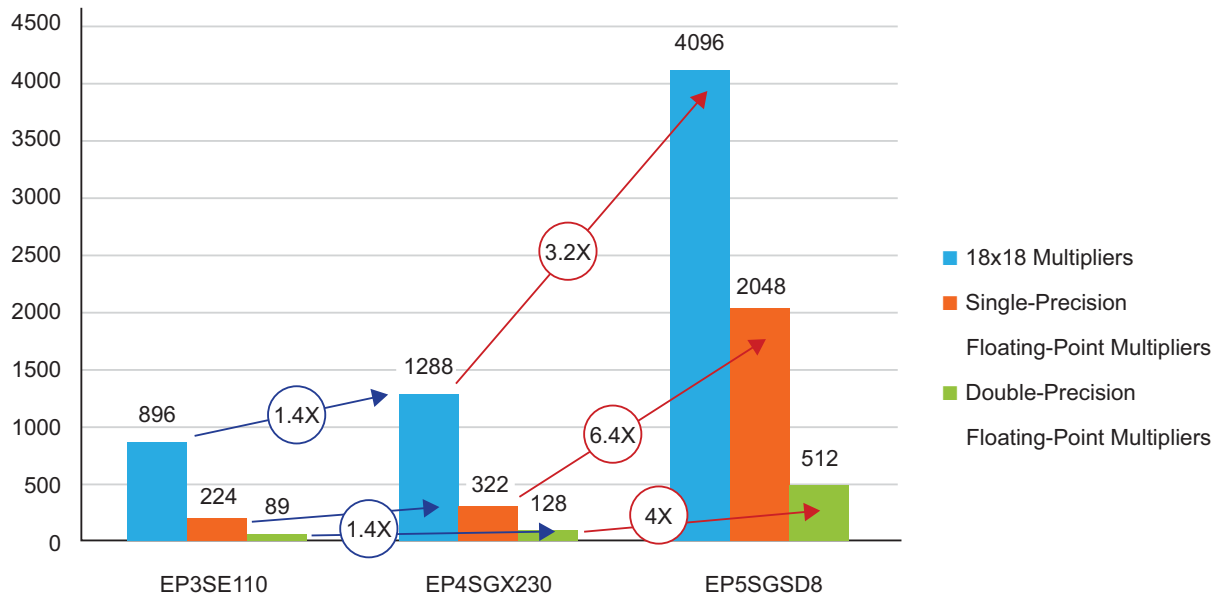
The combination of Stratix V FPGAs and the fused-datapath toolflow can now support 1-teraFLOPS processing rates. No competing FPGA vendor can benchmark this level of performance. The fused-datapath toolflow also works well on other Altera® FPGA families, such as Stratix II, Stratix III, and Stratix IV FPGAs, and Arria® and Arria II FPGAs. Altera has been using this toolflow internally to build floating-point IP and reference designs for several years, and the performance benchmarks (1) for these reference designs can be easily replicated. In addition, the IP for Stratix IV floating-point performance is already available to designers.

To maximize fixed- and floating-performance, Altera has developed a new variable-precision DSP architecture for 28-nm FPGAs. By giving the designer the option to "dial" the DSP block for the required precision, the variable-precision architecture can efficiently support the existing 18x18- and 18x25-bit fixed-point applications, while at the same time offer the higher precision required for floating-point applications. The 27x27- and 54x54-bit modes in particular are designed to support single- and double-precision floating-point applications. The efficiency of the new variable-precision DSP block is critical to provide 1-teraFLOPS performance on a single FPGA die.

New Levels of DSP Resources

The floating-point processing rates are limited by multiplier resources. Altera multiplier densities have progressively increased with each successive family. [Figure 1](#) illustrates the progression of single-precision multiplier density, which has increased over six times from Stratix IV FPGAs to Stratix V FPGAs, and now offers the highest single-precision FPGA multiplier density per die.

Figure 1. Multipliers vs. Stratix III, Stratix IV, and Stratix V FPGAs



The density and architecture of Stratix V FPGAs are optimized for floating-point applications. But a floating-point design flow, such as the fused-datapath toolflow, is also required.

For conventional floating-point implementation in a microprocessor, the input and output data structure for each floating-point instruction conforms to the 754-2008 IEEE Standard for Floating-point Arithmetic (2). This representation of floating-point numbers is very inefficient to implement within an FPGA, as the “twos complement” representation, which is usually well suited to digital hardware implementation, is not used. Instead, the sign bit is separated, and there is an implicit “one” which must be added to each mantissa value.

Specially designed circuitry is necessary to accommodate this representation, which is why microprocessors or DSP blocks typically are optimized for either fixed- or floating-point operations, but usually not both. Furthermore, in a microprocessor, there is no knowledge of the floating-point operations before or after the current instruction, so no optimization can be performed in this regard. This means the circuit implementation must assume that the logic-intensive normalization or denormalization must be performed on each instruction data input and output. Because of the inefficiency resulting from these issues, virtually all FPGA-based designs today are performed in fixed-point operations, even when the algorithm being implemented would work better with the high dynamic range of floating-point operations.

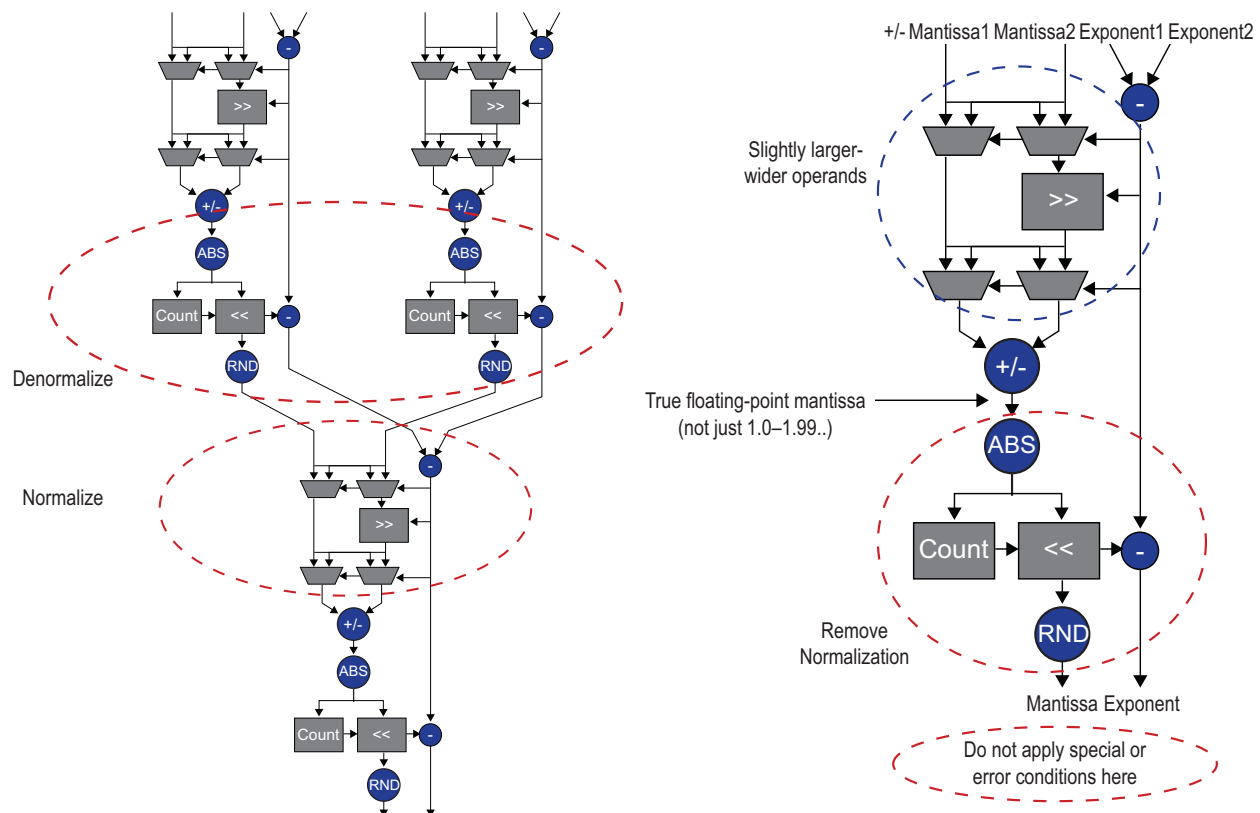
FPGA-specific Optimizations for Floating-point Operations

FPGAs have specific characteristics lacking in microprocessors, and these features can be leveraged to produce a more optimal floating-point flow. FPGAs, unlike microprocessors, have thousands of hardened multiplier circuits. These can be used for both mantissa multiplication, and used as shifters. Shifting of the data is required to perform the normalization to set a mantissa decimal point, and denormalization of mantissas as needed to align exponents. Using a barrel shifter structure would require very high fan-in multiplexers for each bit location, and the routing to connect each of the possible bit inputs. This leads to very poor fitting, slow clock rates, and excessive logic usage, which have discouraged use of floating-point operations in FPGAs in the past.

In addition, an FPGA has the ability to use larger mantissas than IEEE 754 representation. Because the variable-precision DSP blocks support 27x27 and 36x36 multiplier sizes, they can be used for 23-bit single-precision floating-point datapaths. Using configurable logic, the rest of the circuits can by definition be made whatever mantissa size is desired. Using a mantissa size of a few extra bits, such as 27 bits instead of 23 bits, allows for extra precision to be carried from one operation to the next, significantly reducing normalization and denormalization.

As shown in Figure 2, the fused-datapath tool analyzes the need for normalization in the design, and inserts these stages only where necessary. This analysis leads to a dramatic reduction in logic, routing, and multiplier-based shifting resources. It also results in much higher f_{MAX} , or achievable clock rates, even in very large floating-point designs.

Figure 2. Fused-datapath Optimizations



Because IEEE 754 representation is still necessary to comply with the floating-point world, all of the Altera 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 specified by customers. But whether the fused-datapath toolflow gives the same results as the IEEE 754 approach used by microprocessors and how the verification is performed are still under question. Even microprocessors have different floating-point results, depending on how they are implemented.

The main reason for these differences is that floating-point operations are not associative, which can be proved easily by writing a program in C or MATLAB to sum up a selection of floating-point numbers. Summing the same set of numbers in the opposite order will result in a few different LSBs. To verify the fused-datapath method, the designer must discard the bit-by-bit matching of results typically used in fixed-point data processing. The tools allow the designer to declare a tolerance, and to compare the hardware results output from the fused-datapath toolflow to the simulation model results.

A large single-precision floating-point matrix inversion function was implemented using the fused-datapath toolflow (3), and tested across different-size input matrices. These results were also computed using an IEEE 754-based Pentium processor. Next, the reference result was computed on the processor, using double-precision floating-point operations, which provided near-perfect results compared to single-precision architecture. By comparing both the IEEE 754 single-precision results and the single-precision fused-datapath results, and computing the Frobenious norm of the differences, it can be shown that the fused-datapath toolflow gives more precise results than the IEEE 754 approach, due to the extra mantissa precision used in the intermediate calculations.

Table 1 lists the mean and the standard deviation and Frobenious norm where the SD subscript refers to IEEE 754-based single-precision architecture in comparison with the reference double-precision architecture, and the HD subscript refers to the hardware-based fused-datapath single-precision architecture in comparison with the reference double-precision architecture.

Table 1. Fused-datapath Precision Results

Frobenious Norm/ Standard Deviation	8x8	32x32	64x64	128x128
$\ E_{SD}\ _F$	57.60	9.40	5.33	2.29
σ	459.18	44.30	36.94	7.60
$\ E_{HD}\ _F$	10.38	2.73	1.65	1.27
σ	47.10	10.36	7.36	5.33

Fused-datapath Example

The fused-datapath toolflow has been integrated into Altera's DSP Builder Advanced Blockset toolflow. The design environment for the fused-datapath toolflow, as part of DSP Builder, is The MathWorks' Simulink. This environment allows for easy fixed- and floating-point simulation as well as FPGA implementation of customer designs.

- ASIN
- CEIL
- ACOS
- FABS
- ATAN
- SQRT
- EXP
- DIVIDE
- LOG
- 1/SQRT
- LOG10

The implementation is multiplier based because Altera FPGAs have an abundance of high-precision multipliers. Traditionally, CORDIC implementations have been used for such operations, but the rationale for this choice is based on the outdated ASIC idea that logic is very cheap and multipliers relatively expensive. For floating-point operations on an FPGA, multiplications are now cheap and have the advantages of predictable performance, low power, and low latency. Contrary to accepted wisdom, the multistage CORDIC approach is a highly inefficient method to build these functions.

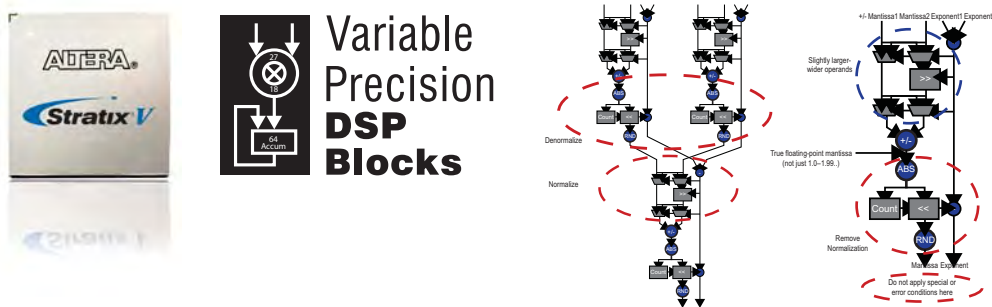
One of the most common functions requiring high dynamic range is matrix inversion. To support this, the fused-datapath library includes a linear algebra library, which includes the following functions:

- Matrix multiply
- Cholesky decomposition
- LU decomposition
- QR decomposition
- More linear algebra and vector functions will be added over time.

The comprehensive library support of the fused-datapath toolflow within the DSP Builder Advanced Blockset allows customers to build large, complex, and highly optimized floating-point datapaths. In addition, fixed- and floating-point operations can be easily mixed within the same design.

Achieving 1-TeraFLOPS Performance

Building high-performance floating-point systems requires both the right FPGA hardware resources and the fused-datapath design flow. This requires DSP blocks suited to floating-point operations—and lots of them. Stratix V FPGAs offer more single-precision floating-point multipliers per die than any other competing 28-nm FPGA. They also require careful mapping of floating-point datapaths to FPGA architecture, which is performed by Altera's exclusive software and IP. In addition, major changes in both hardware and software (Figure 4) were required to achieve 1-teraFLOPS processing rates on Stratix V FPGAs.

Figure 4. Hardware and Software Innovations for Floating Point

18x18, 27x27, 36x36 seamless trade-off High f_{MAX} with logic and routing reductions
Greatly increased multiplier density

On the hardware side, there is greatly increased multiplier, or DSP block, density. The DSP blocks are the new variable-precision DSP architecture, which allows 18x18-, 27x27-, and 36x36-bit modes, and simple construction of a 54x54-bit mode with a seamless trade-off between the configurations. There are also new fused datapath algorithms that better combine multipliers and logic, and use vector datapaths. FPGAs are nominally fixed-point devices with the right mix of logic and DSP balanced for fixed-point operations. This usually results in a deficiency of logic to efficiently implement floating-point operations. But with fused datapath technology, logic, latency, and routing, all can be significantly reduced by more than 50% to bring the balance needed for floating-point operations in line with that for fixed-point operations.

On the software side, Altera offers the industry's most comprehensive set of floating-point megafunctions, including FFT, matrix, and trigonometric functions. Best of all, the fused-datapath technology is currently being integrated into DSP Builder Advanced Blockset. This provides the capability for large, customized, and mixed fixed- and floating-point designs in a toolflow which that analyzes, optimizes, and compiles datapaths that may fill an entire FPGA.

The Stratix V EP5SGSD8 FPGA offers the highest level of DSP performance on a single die of any 28-nm FPGA. It contains the following resources:

- FPGA logic resources
 - 703K logic elements (LEs) or
 - 282K adaptive logic modules (ALMs)
- Organized as
 - 574K adaptive lookup tables (ALUTs)
 - 1128K registers
- Hardened blocks
 - 4096 multipliers (using 18x18 precision) or
 - 2048 multipliers (using 27x27 precision)
 - 55-Mb internal RAM (in 20k blocks)

Although the Stratix V floating-point benchmark design is not yet available, the floating-point performance can be conservatively estimated using Stratix IV benchmark designs. Normally, peak floating-point performance benchmarks are made using matrix multiplication. This is because matrix multiplication is formed from vector dot products, which is simply cross-multiplying two vectors, and summing the products using an adder tree.

The Altera matrix multiply megafunction, available in today in Quartus® II design software, has been benchmarked as shown in Table 2. For example, the 64x64-bit matrix multiply operates at 388 MHz in a Stratix IV FPGA.

Table 2. Floating-Point Performance for Matrix Multiply with Single Precision

Dimension	Vector Size	Logic Usage					f _{MAX}	Latency (Cycles)	GFLOPS
		ALMs	DSP	M9K	M144K	MemBits			
8x8 × 8x8	8	3,367	32	26		14,986	420	209	6.30
16x16 × 16x16	8	3,585	32	27		55,562	421	611	6.32
32x32 × 32x32	16	6,301	64	76		339,718	419	2,172	13.00
64x64 × 64x64	32	11,822	128	80	16	2,382,318	388	8,353	24.45

This megafunction benchmark uses a fairly large 12K ALM, or 30K-LE floating-point core, yet it is able to close timing at almost 400 MHz. Without incorporating the fused-datapath technology in the core, it would be impossible to achieve this level of performance on a large floating-point design. Now this core uses a vector size of 32, meaning that it operates by cross-multiplying vectors of length 32 each clock cycle. There are 32 floating-point multipliers, and an adder tree of 16 in the first level, 8 in the second, 4 in the third, 2 in the fourth, and finally 1 adder in the fifth level, for a total of 31 adders. Along with the 32 multipliers, there are a total of 63 floating-point operations per clock cycle at 388 MHz, giving 24.4 GFLOPS. Using the first-generation fused-datapath technology in Stratix IV FPGAs, the 12K ALMs and 128 multipliers of 18x18 size are used to achieve 24 GFLOPS.

Benchmarking the Stratix V FPGA with its second-generation fused-datapath tools, and the improved architecture shows a marked improvement over the Stratix IV FPGA. The Stratix V logic has been enhanced with twice as many registers, four, per ALM. This is important, because floating-point operations tend to be very register intensive. Doubling the register-to-lookup-table ratio enhances the FPGA's capability in these applications. Another major innovation is the variable-precision DSP block and the new 27x27-bit mode with 64-bit accumulators. Previous Stratix generations use the 36x26-bit mode instead. The variable-precision DSP block results in a doubling of DSP resource efficiency, relative to Stratix IV FPGAs.

Table 3 shows how the second-generation fused-datapath toolflow achieved significant reduction in logic usage relative to the first generation. A 64-vector dot product uses just a bit more logic than the 32-vector product in the first-generation fused-datapath technology. Second-generation fused-datapath technology is being integrated into DSP Builder.

Table 3. Vector Dot-product Logic Resources

64-Vector Dot-product Sum (Single-precision Floating-point Operations)	Second-generation Fused Datapath	First-generation Fused Datapath
ALUTs	13.4K	21.6K
Registers	16.4K	28.9K

The 64-bit vector product sum calculation is 127 floating-point operations, resulting in 49 GFLOPS. The benchmarking conservatively assumes that Stratix V FPGAs operate at the same rate as Stratix IV FPGAs.

GFLOPs Limitations

Limitations for the Stratix V EP5SGSD8 FPGA are imposed by three resources: logic, registers, and multipliers:

- Logic limited
 - 13.4K ALUTs = 127 floating-point operations = 49 GFLOPS
 - $574 / 13.4 = 43$ vectors
 - 43×49 GFLOPS = 2107 GFLOPS
- Register limited
 - 16.4K registers = 127 floating-point operations = 49 GLOPS
 - $1128 / 16.4 = 69$ vectors
 - 69×49 GFLOPS = 3381 GFLOPS
- Multiplier limited
 - Need 64 multipliers (27x27) per vector
 - $2048 / 64 = 32$ vectors
 - 32×49 GFLOPS = 1568 GFLOPS

The limiting factor is multiplier resources. This is excellent news for the designer, because it is much more feasible to use 100% of DSP block resources than 100% of the logic resources. As the logic usage approaches 100%, closing timing can get more difficult. In addition, additional logic is needed to get data in and out of the FPGA, to build memory buffer interfaces and other functions.

The basis for the f_{MAX} is the Stratix IV large matrix-multiply benchmark, which runs at 388 MHz. But to be on the safe side, the benchmarking has derated the f_{MAX} by 20% to about 310 MHz. This reduces the GFLOPS per vector product sum by 20%, to about 39 GFLOPS. For 32-vector products, this results in 1.25 teraFLOPS, and the following resource usage:

- Multiplier resource usage = 100%
- Logic resource usage = 75%
- Register resource usage = 46%

This analysis, based upon existing Stratix IV benchmarks and fused-datapath technology, shows that an over 1-teraFLOPS floating-point benchmark is easily achievable. The actual benchmark will be measured against power consumption when Stratix V FPGAs are available, and power efficiency is expected to be in the range of 10 to 12 GFLOPS/W.

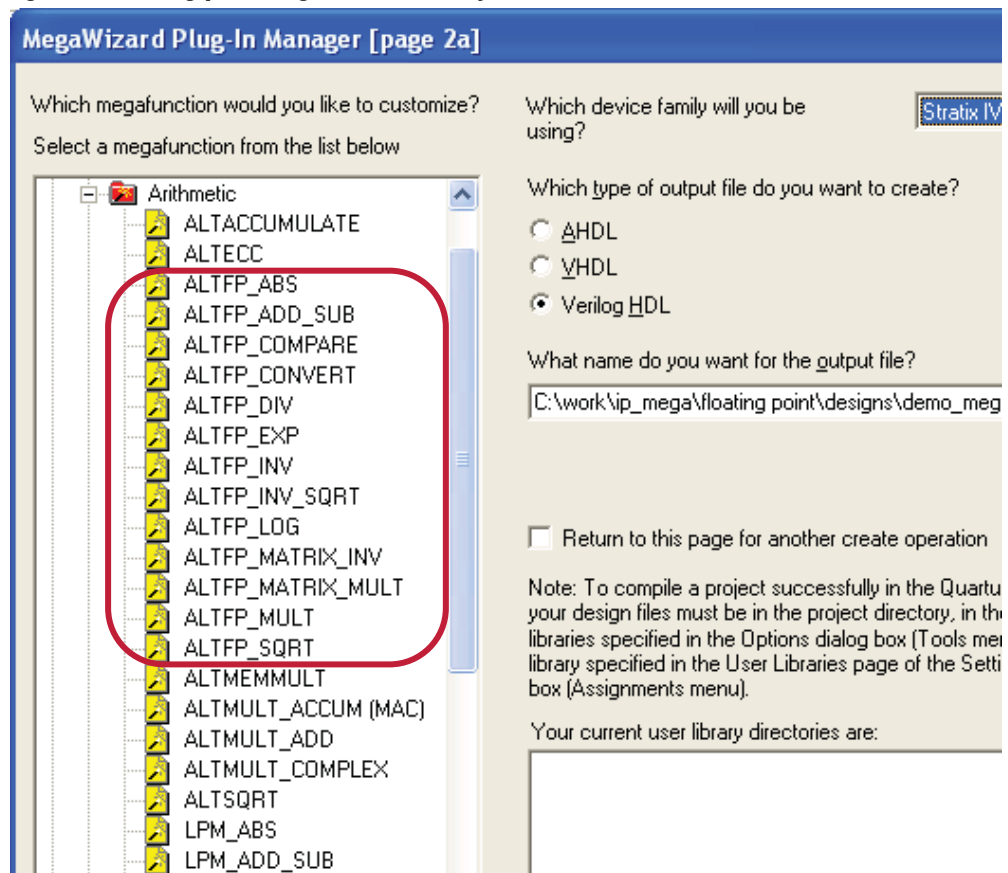
Floating-point IP Cores

While floating-point IP was first released by Altera in 2009, the associated benchmarks demonstrate that this robust fused-datapath technology has been working and used internally by Altera IP development teams for several years.

Altera has an extensive floating-point library of megafunctions. All of these functions support single-precision floating-point architectures, most support double- and single-extended-precision architectures, and all have IEEE 754 interfaces. The library also includes advanced functions, such as matrix inversion and FFTs, which no other FPGA vendor offers. Many of the floating-point megafunctions incorporate the fused-datapath technology. In fact, most of these functions are not practical without the fused-datapath approach. This explains why competing FPGA vendors do not offer these floating-point functions.

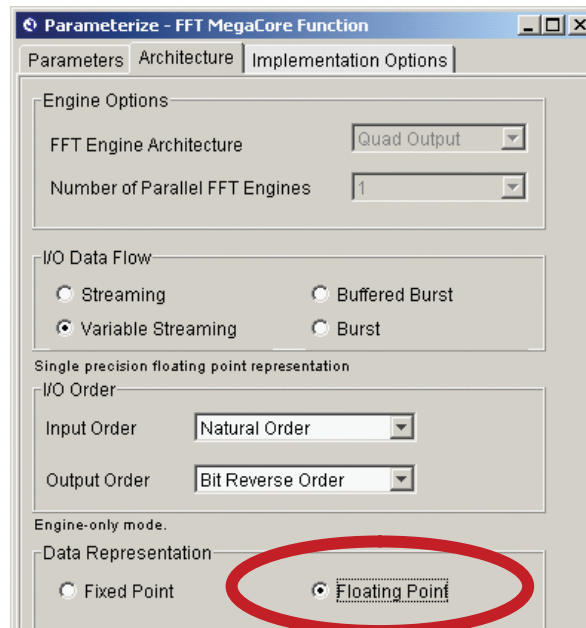
The Altera floating-point megafunctions are easily accessible within Quartus II software, with no additional license fees. They can be found under the arithmetic function library, shown in Figure 5, with the exception of the FFT, which is part of the standard Altera FFT MegaCore[®] function.

Figure 5. Floating-point Megafunction Library



The FFT MegaCore function incorporates an option for floating-point implementation, as shown in [Figure 6](#).

Figure 6. FFT MegaCore GUI



The higher precision of Altera DSP blocks also allows the implementation of floating-point FFTs. These FFTs use a 36x36-bit complex multiply in Stratix IV FPGAs, or the new, more efficient 27x27-bit complex multiply available with the Stratix V FPGA's variable-precision DSP block. Not only does the architecture efficiently implement such a structure, but the new FFT compiler enables the designer to choose the floating-point option from an easy-to-use GUI, thus making the code generation of such a complex structure as easy as pushing a button. This FFT core is useful in next-generation radar designs, which are increasingly migrating from fixed- to floating-point implementations, in order to provide greater detection ranges or detection of ever smaller targets. Achieving the required level of system performance needs the dynamic range of floating-point operations.

Using the FFT MegaCore function incorporating the first-generation fused-datapath toolflow, a study compared Pentium processor-based FFT throughput to FPGA-based FFT throughput. A total of 14 single-precision FFTs were implemented in a Stratix IV FPGA. The design closed timing at over 220 MHz, with a 96% full device. As shown in [Table 4](#), the FPGA's timing was very impressive, effectively demonstrating the ability to fill a device full of floating-point circuits and still efficiently route the design. The dynamic power consumption was about 1W per FFT, or 14W in this case.

Table 4. Performance of 14 FFT Floating-point MegaCores, First-Generation Fused-Datapath Toolflow, 1,024 pt

Stratix IV EP4SGX530	Usage	Max	Percentage
Logic utilization	406,465	424,960	96
ALUTs	308,521	424,960	73
Reg	294,579	424,960	69
M9K	1,280	1,280	100
M144K	64	64	100
DSP block 18-bit	896	1,024	88
f_{MAX}	222.72 MHz		
Transform time per core	4.5977 μ s (normalized: 0.3284 μ s)		

The second-generation fused-datapath toolflow (Table 5) does even better. Note that the clock rate is now over 300 MHz, and the logic utilization has dropped to 70%.

Table 5. Performance of 14 FFT Floating-point MegaCores, Second-Generation Fused-Datapath Toolflow, 1,024 pt

Stratix IV EP4SGX530	Usage	Max	Percentage
Logic utilization	300,000	424,960	70
ALUTs	224,000	424,960	53
Reg	210,000	424,960	49
M9K	1,280	1,280	100
M144K	64	64	100
DSP block 18-bit	896	1,024	88
f_{MAX}	300+ MHz		
Transform time per core	3.4 μ s (normalized: 0.24 μ s)		

Conclusion

To conclude, the fused-datapath toolflow is a major innovation in FPGA-based DSP technology. It is the only known synthesis tool able to generate floating-point datapaths in FPGAs, and provides another arrow in the quiver of available design tools and capabilities. Coupled with the 28-nm Stratix V FPGAs, floating-point processing rates in excess of 1 teraFLOPS are now feasible. In addition, the full complement of `math.h` and linear algebra functions provide the designer with full support for complex designs. This functions allow designers the abilities to meet the dynamic range, precision, and processing loads required in next-generation DSP applications and products.

Further Information

1. Altera Floating-point Megafunctions:
www.altera.com/products/ip/dsp/arithmetics/m-alt-float-point.html
2. 754-2008 IEEE Standard for Floating-point Arithmetic:
<http://ieeexplore.ieee.org>
3. Suleyman S. Demirsoy and Martin Langhammer, "Fused Datapath Floating Point Implementation of Cholesky Decomposition," *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, February 22 - 24, 2009:
<http://portal.acm.org/dl.cfm>

Acknowledgements

- Michael Parker, Senior Technical Marketing Manager, DSP IP and Technology Product Marketing, Altera Corporation

Document Revision History

Table 6 shows the revision history for this document.

Table 6. Document Revision History

Date	Version	Changes
September 2010	1.0	Initial release.