
Taking Advantage of Advances in FPGA Floating-Point IP Cores

Recently available FPGA design tools and IP provide a substantial reduction in computational resources, as well as greatly easing the implementation effort in a floating-point datapath. Moreover, unlike digital signal processors, an FPGA can support a DSP datapath with mixed floating- and fixed-point operations, and achieve performance in excess of 100 GFLOPS. This is an important advantage, for many high-performance DSP applications only require the dynamic-range floating-point arithmetic in a subset of the total signal processing. The choice of FPGA implementation coupled with floating-point tools and IP allows the designer flexibility in a mix of fixed-point data width, floating-point data precision, and performance levels unattainable by a processor-based architecture.

Introduction

Many complex systems in communications, military, medical, and other applications are first simulated or modeled using floating-point data processing, using C or MATLAB software. However, final implementation is nearly always performed using fixed-point or integer arithmetic. The algorithms are carefully mapped into a limited dynamic range, and scaled through each function in the datapath. This requires numerous rounding and saturation steps, and if done improperly, can adversely affect the algorithm performance. This usually also requires extensive verification during integration to ensure system operation matches simulation results and has not been unduly compromised.

Previously, the lack of support with FPGA tool suites made floating-point arithmetic an unattractive option for the FPGA designer. Another drawback was poor performance when using many floating-point FPGA operators, due to the dense logic and routing resources required. The key to efficient FPGA implementation of complex floating-point functions is to use multiplier-based algorithms, as this leverages the large amount of hardened multiplier resources integrated into the FPGA devices. The multipliers used to implement these often non-linear functions must have extra precision in the multipliers to keep the required precision through the multiply iterations. Additionally, the high-precision multipliers remove the requirement for normalization and denormalization at every single multiply iteration, which can significantly reduce logic and routing requirements.

FPGAs incorporate hardened digital signal processing (DSP) blocks capable of implementing efficient 36-bit x 36-bit multipliers, which provide ample extra bits above the normal single-precision 24-bit mantissa requirement for single-precision floating-point arithmetic. These multipliers also can be used to build larger multipliers for double-precision floating-point applications, up to a 72-bit x 72-bit size.

To the designer, use of floating-point arithmetic often provides enhanced performance due to large dynamic range and greatly simplifies the task of system performance verification against a floating-point simulation. And in some applications, fixed-point arithmetic is just not feasible. One common example where the dynamic range requirements virtually dictate the use for floating-point arithmetic is matrix inversion.

Floating Point IP Cores

Altera now offers the most comprehensive set of single- and double-precision floating-point IP cores in the industry, operating at very high performance. Floating-point IP cores available currently include:

- Add/subtract
- Multiply
- Divide
- Inverse
- Exponent
- Logarithm
- Square root
- Inverse square root
- Matrix multiply

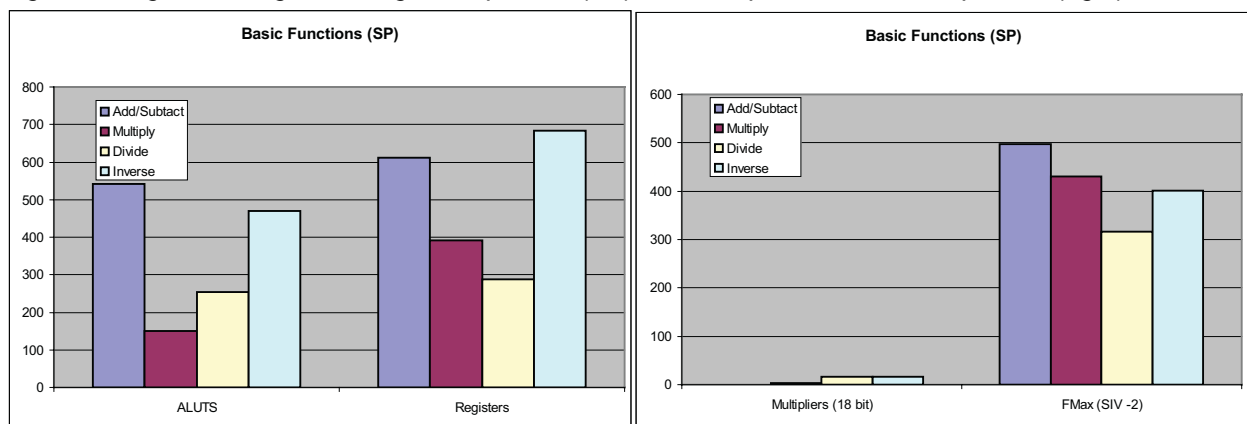
- Matrix inversion
- Fast Fourier Transform (FFT)
- Compare
- Integer and fractional conversion

Only single-precision figures are provided in this white paper. For double-precision figures, refer to the *Floating-Point Megafunctions User Guide*.

Basic Functions

The basic floating-point functions and their performance are detailed in Figure 1. The resources and performance necessary for floating-point division are comparable to addition and subtraction, which means that system designers are not required to avoid division operations in their algorithms to ease hardware implementation.

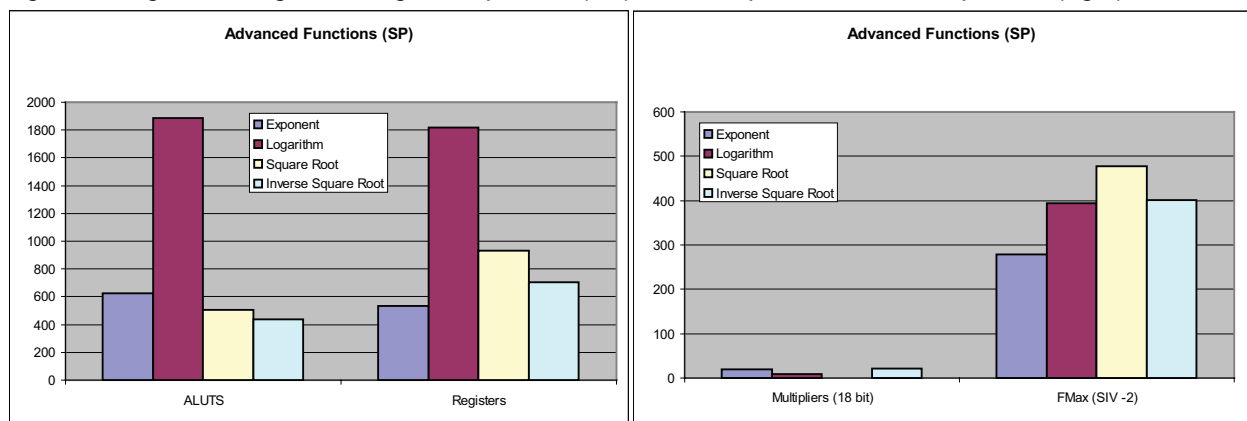
Figure 1. Logic and Register Usage Comparison (left) and Multiplier and f_{MAX} Comparison (right)



Advanced Functions

Altera also offers more advanced floating-point functions. Due to use of multiplier-based algorithms, performance is comparable to the more basic functions, as shown in Figure 2.

Figure 2. Logic and Register Usage Comparison (left) and Multiplier and f_{MAX} Comparison (right)



Matrix-Multiply Function

Altera is unique in offering the only FPGA-based parameterizable floating-point matrix IP cores. These operators integrate dozens or hundreds of floating-point operators, yet maintain high performance. The matrix-multiply core

can also be used to easily perform standard benchmarking or GFLOP/S and GFLOP/W. Performance results for the SGEMM matrix-multiply core, shown in Table 1, are actual post-compiled timing-closed results, unlike the pencil and paper floating-point calculations normally used to determine GFLOP/S. This type of benchmarking is not available from any other FPGA vendor, and can be easily reproduced by customers using the parameterizable matrix-multiply IP cores available in Altera's Quartus® II software.

Table 1. Single-Precision Matrix-Multiply Performance Results

MatrixAA Size	MatrixBB Size	Vectorsize	Logic Usage					GFLOPS	f _{MAX} (MHz)	Power (mW)			
			ALMs (1)	DSP Usage (2)	M9K	M144K	Memory (bits)			Static	Dynamic	I/O	Total
36x112	112x36	8	4,604	32	43	2	576,200	4	291	2,008	1,063	300	3,334
36x224	224x36	16	7,882	64	77	4	1,101,920	9	291	2,045	1,821	300	4,165
36x448	448x36	32	14,257	128	137	8	2,153,040	18	291	2,110	3,448	300	5,858
64x64	64x64	32	13,154	128	41	8	1,333,233	18	292	2,112	2,604	306	5,023
128x128	128x128	64	25,636	256	141	16	3,173,189	37	293	2,244	5,384	306	7,934

Notes:

- (1) Adaptive logic modules
- (2) 18x18 DSP blocks

Using the Quartus II power estimator, real-world giga floating-point operations per second per watt (GFLOPS/W) can be easily calculated. The results reach 5 GFLOPS/W when using a partially full Altera® Stratix® IV EP4SE230 FPGA. Using larger matrix-multiply cores on the Stratix IV EP4SE530 device results in approximately 7 GFLOPS/W at a computation density of 200 GFLOPS. The highest efficiency results when the FPGA's static power consumption is amortized over large floating-point implementations, utilizing the entire device.

Altera has developed a floating-point technology that results in a dramatic reduction in the logic and routing required to implement large floating-point datapaths. Use of this floating-point datapath optimization tool is critical, as the reduction pushes the logic/routing-per-floating-point operator ratio into a range provided by high-end FPGAs. This is reflected by the remarkable ability of the tool to deliver consistent f_{MAX} performance of nearly 300 MHz, independently of the matrix-multiply size instantiated. In this way, customers can reliably fill FPGAs to over 80% capacity and achieve >200-MHz f_{MAX} performance in large floating-point designs.

Matrix-Inversion Function

One of the most common applications for floating-point arithmetic in FPGAs is matrix inversion. The matrix-inversion function is required for most wireless multiple-input, multiple output (MIMO) algorithms, radar STAP systems, medical-imaging beamforming, and many high-performance computing applications. The sample performance of the parameterizable matrix-inversion floating-point IP core (Table 2) shows extremely high matrix throughput. A 4x4 matrix-inversion core is able to operate at 20 million matrix inversions per second, fast enough for LTE wireless MIMO applications.

Table 2. Single-Precision Floating-Point Matrix-Inversion (Cholesky Algorithm) Performance

Dimension	Logic Usage					f _{MAX}	Latency (cycles)	GFLOPS
	ALMs	DSP	M9K	M144K	MemBits			
8x8 * 8x8	5,538	63	49	—	53,736	332	2,501	15.26
16x16 * 16x16	8,865	95	80	—	138,051	329	11,057	30.93
32x32 * 32x32	15,655	159	193	—	699,164	290	52,625	55.12
64x64 * 64x64	29,940	287	386	22	4,770,369	218	281,505	83.16

Fast Fourier Transform Function

FFTs are another high-dynamic-range application. Due to the nature of the FFT algorithm, the bit precision naturally increases as the FFT size grows. Some applications use cascaded FFTs, which can require even higher dynamic range. Many radar implementations use a FFT to bin the range values, using fixed-point arithmetic. This is often followed by a second FFT, to bin the Doppler values, and the dynamic range can be high enough to require floating-point arithmetic. As Figure 3 and Figure 4 show, increased logic is need to implement single-precision floating-point arithmetic compared to fixed-point arithmetic, but the circuit f_{MAX} , memory, and multipliers are similar.

Figure 3. FFT Logic and Register Usage Comparison

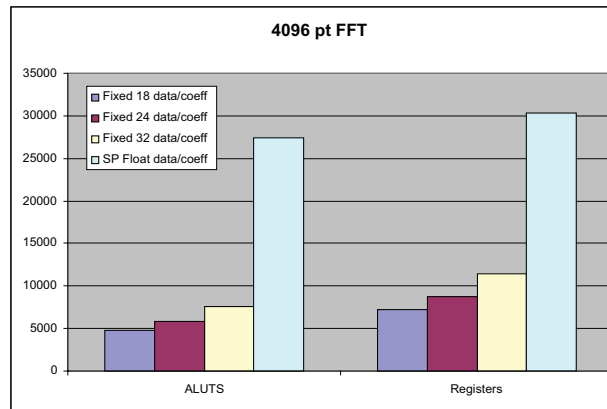
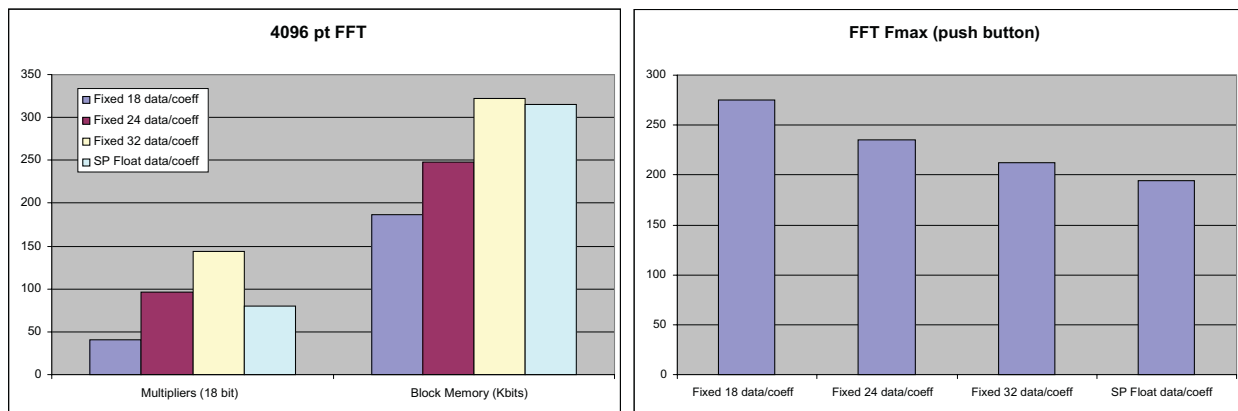


Figure 4. FFT Multiplier and Memory Usage Comparison (left) and FFT f_{MAX} Comparison (right)



Conclusion

Altera’s new techniques of the floating-point circuit optimization, integrated into the floating-point IP cores and coupled with the growth in density and logic resources, provide exceptional FPGA floating-point performance. Other vendors offer specific floating-point processor solutions, but most do not match the high-GFLOPS performance of Altera’s FPGA solutions, and none of them can match the GFLOP/W available with Stratix IV FPGA-based solutions. This is borne out by independent benchmarking conducted by the National Science Foundation (NSF)’s Center for High-Performance Reconfigurable Computing (CHREC), which has rated the Stratix IV EP4SE530 as the overall leader in double-precision floating-point processing.

Additional Altera FPGA advantages are industry-leading external memory bandwidth and SERDES transceiver performance of up to 12.5 Gbps. The FPGA platform also provides the highest performance fixed-point datapaths, with ultimate flexibility in I/O and memory interfaces. With these capabilities, Stratix IV FPGAs offer an ideal platform to build high-performance floating-point datapaths, which can be leveraged on a wide variety of

applications, from high-performance computing to radar and electronic warfare to MIMO-based SDR/wireless systems to medical beam-forming applications.

Further Information

Floating-Point Megafunctions User Guide:
www.altera.com/literature/ug/ug_altfp_mfug.pdf

Acknowledgements

Michael Parker, Sr. Technical Marketing Manager, IP and Technology Product Marketing, Altera Corporation



101 Innovation Drive
San Jose, CA 95134
www.altera.com

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.