White Paper

Automating DSP Simulation and Implementation of Military Sensor Systems

Military sensor-driven systems normally use FPGAs to interface with the ADCs that digitize sensor inputs. Because ADCs operate at rates of up to 3 MSPS, they require very high-performance DSP circuitry. In most cases, this functionality is implemented using FPGAs optimized for DSP applications.

Introduction

Military sensor-driven systems normally use FPGAs to interface with the analog to digital converters (ADCs) that digitize sensor inputs. The ADCs operate at rates of up to 3 MSPS, which requires very high-performance digital signal processing (DSP) circuitry, often implemented by FPGAs optimized for such DSP applications. The Altera® 40-nm FPGA families, capable of unprecedented density and performance, provide the highest DSP performance now available in a programmable platform. However, this is only part of the story, as optimizing a military sensor DSP algorithm in an FPGA can be a major engineering effort with multiple steps:

1. Port the sensor algorithm, normally modeled in either "C" or The MathWorks MATLAB and Simulink tools, to HDL.

2. Verify RTL functional simulation to match the high-level simulation test vectors: Typically a manual process, this requires a transfer of testbench data to and from the simulation to the HDL testbench.

3. Perform timing verification: The higher the required clock rate, the more pipelining and circuit optimization is required by the designer. Therefore the timing closure process can be very iterative, with multiple compilations required.

4. Repeat functional and timing verification: If multiple channels must be processed simultaneously, the whole process is complicated by the need for control logic and memory to provide the proper time division multiplexing of the data flow, while maintaining the timing margins using the desired clock rate.

Happily, this whole process can now be automated, allowing high-level design entry, multiple channels, and optimized performance. The simulation environment and implementation environments are common, which gives the system designer assurance that the algorithms simulated exactly match the hardware implemented. This design methodology reliably delivers well over 300 MHz in large designs, ideal for high-performance sensor applications.

Design Entry Using DSP Builder Advanced Blockset

Building military sensor suite-based systems using FPGAs can be challenging. Unexpected delays occur in schedule, design, integration, verification, feature requirement changes, and transition to production. Often, design teams feel there is no time or acceptable risk to adopt alternate design methodologies due to aggressive project schedules. There is a trade-off between investment in better design and verification methods, and expected reductions in product development delays in design, integration, and verification efforts. There is also often skepticism that a new design methodology can actually deliver its promised improvement.

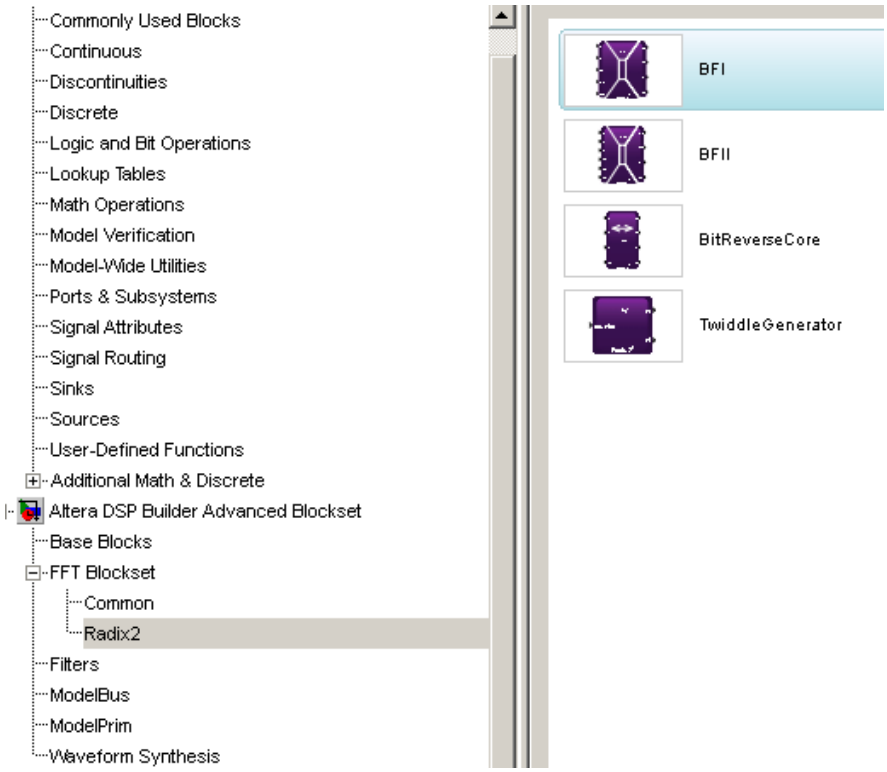
The risk of adopting a design methodology can be reduced dramatically if it is compatible with current design flows, and can be introduced incrementally. This allows specific areas of a design to be implemented and verified using the new methodology, and integrated with the rest of the design build using traditional HDL methods. Risk is reduced by a gradual and systematic adoption, which can be influenced by both the actual productivity advantages and the associated learning curve. The design team also retains the freedom to adopt the new flow only in areas where the benefits clearly outweigh the initial cost of adoption.

Fortunately, an FPGA development tool available today simplifies the product development cycle and eliminates many of the common bottlenecks that occur throughout the product life cycle. The tool controls the design process

via top-level design parameters, so changing the clock rate, increasing or decreasing the number of channels, or changing FPGA family involves a just few clicks and a recompile, thus facilitating design reuse or late product specification changes. This capability is now available through Altera's Advanced Blockset within the DSP Builder tool in Quartus® II design software. The Advanced Blockset takes a high-level behavioral description of the DSP algorithm using the Simulink tool and allows an algorithm's signal flow to be described in a model form, similar to what might seen in a DSP or RADAR algorithm textbook.

This capability of having a higher level of design abstraction and productivity and yet delivering the performance of traditional optimized HDL is unique to the DSP Builder tool's Advanced Blockset. The design can be both simulated and instantiated into FPGA hardware, using the Simulink environment of block-diagram circuit descriptions. Within the Simulink environment, it is simple to recreate the textbook diagram using elements such as registers, multipliers, and adders, in a very intuitive and graphical form. For example, a common algorithm in military sensor designs systems is the fast Fourier transform (FFT). Advanced Blockset makes it very easy to construct a textbook-like FFT implementation intuitively using a library of common primitive blocks (Figure 1) such as butterflies and twiddle factors.

Figure 1. FFT Building Blocks



Advanced Blockset makes it very easy to construct intuitively a textbook-like FFT implementation using a library of common ready built blocks such as butterflies and twiddle factors shown in Figure 2, An FFT circuit butterfly stage (Figure 3) appears just like the flow diagrams in DSP texts. Example FFT designs are included in the Advanced Blockset demo library.

Figure 2. Radix-4 FFT Butterfly

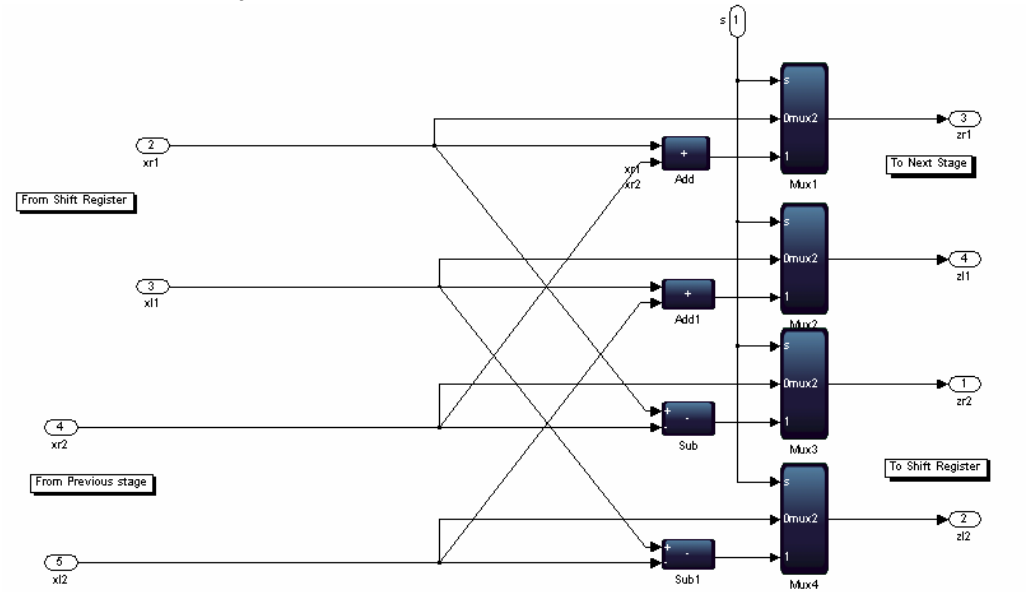
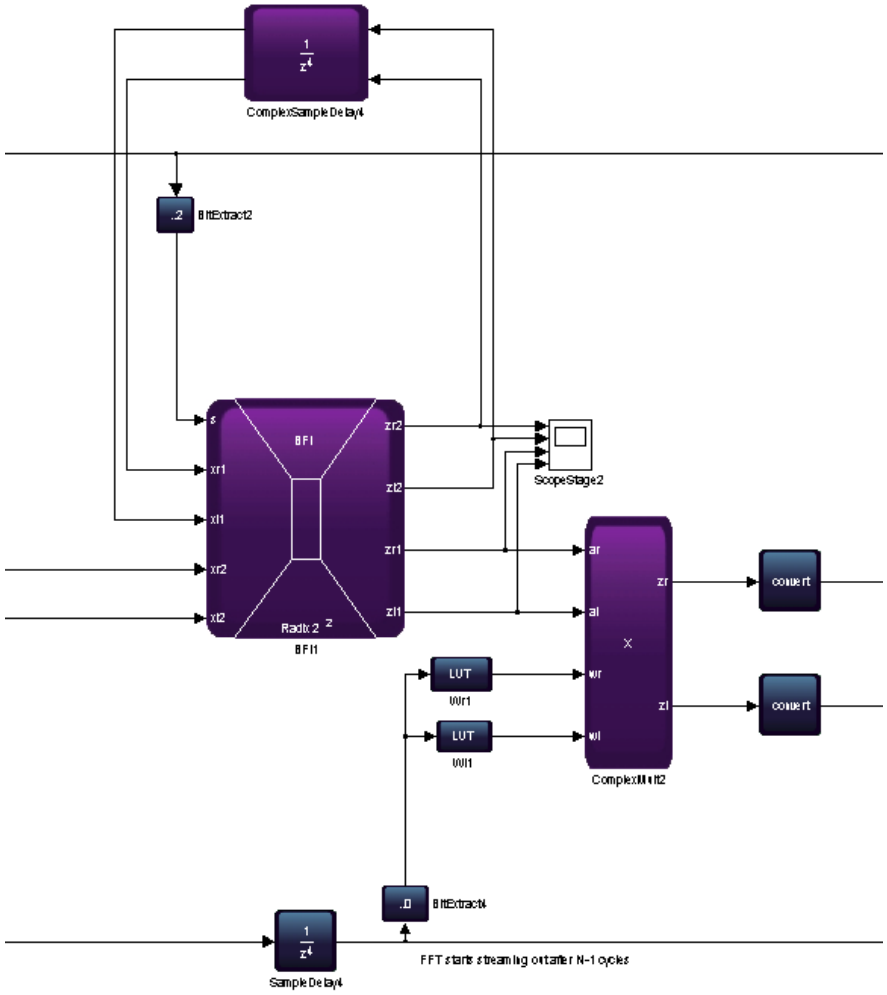


Figure 3. FFT Butterfly Stage



Testbenches and Debugging in Simulink

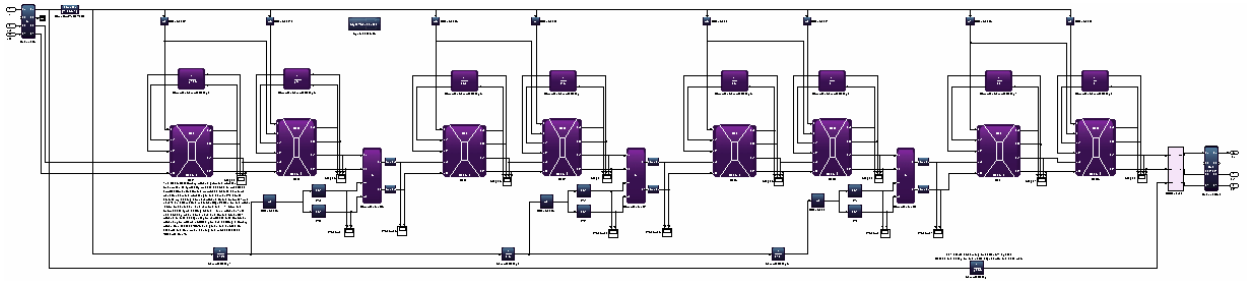
The Simulink environment makes it easy to add virtual sources, such as sinusoidal sources, modulated sources, or even noise-like signals, thus simplifying the production of testbenches. Output signals can be terminated with virtual oscilloscopes and spectrum analyzers to view both time and frequency domain responses. In this way, the FFT or finite impulse response (FIR) filter response can be seen easily. Because the testbench circuits are not implemented in the FPGA, they can be implemented using any Simulink block.

The Advanced Blockset is also integrated with the Mentor Graphics® ModelSim® tool, allowing a comparison of the outputs of the Simulink design to that of the generated RTL. This gives the designer assurance that the RTL produces a bit-exact match to the high-level Simulink output.

Multichannel Support

Many, if not most, DSP datapaths use multiple channels with parallel processing of the I and Q quadrature signals. In order to make this a multichannel design, simply add “ChannelIn” and “ChannelOut” blocks to the design diagram. There is no need to design logic to time division multiplex the DSP circuits among multiple channels; this is done automatically. An example of this is the 256-point FFT shown in Figure 4, where the input and output are delineated by the ChannelIn and ChannelOut blocks, respectively.

Figure 4. 256-Point Radix 4-FFT Dataflow



To perform design verification and HDL generation, simulate the design within the Simulink tool. Each time the Simulink file is run, the Advanced Blockset is activated in the background. It takes the Simulink behavior model, along with parameters provided, and performs a timing-driven synthesis of RTL using the built-in timing models for any FPGA and speed grade built into the Quartus II software. The Advanced Blockset adds pipeline registers and control logic as necessary to achieve the clock rate provided. By adding channelization blocks, the Advanced Blockset automatically configures the FPGA registers and control logic to perform time-division multiplexing, so the high clock-rate filter can process multiple lower speed data-rate channels.

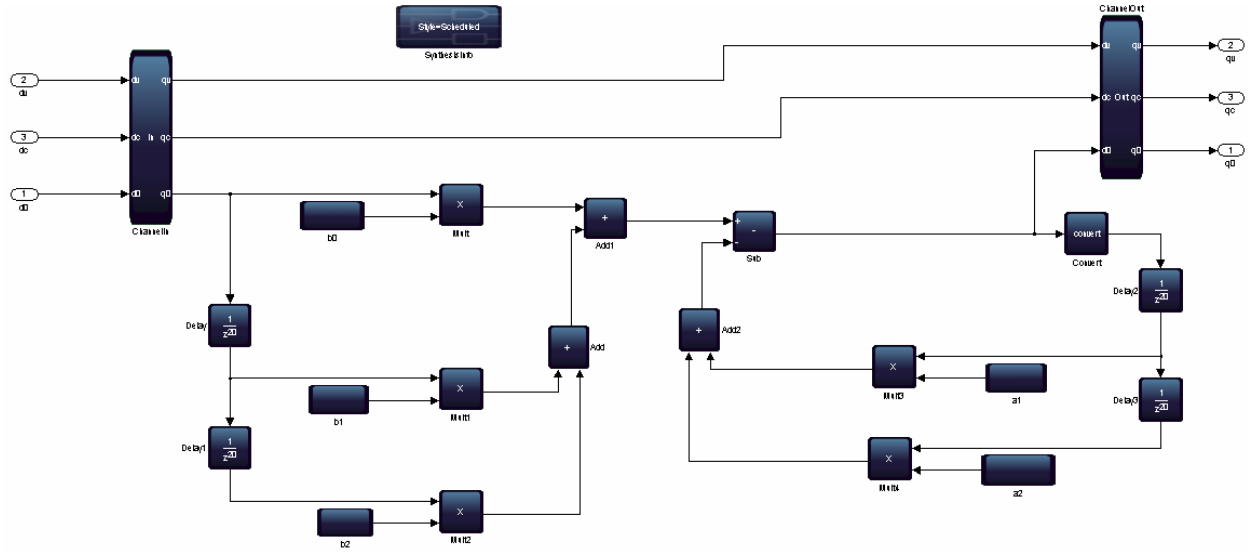
In all cases, the Advanced Blockset uses multiplier and logic resources as necessary to provide the aggregate number of MMACS to perform DSP over the specified number of channels using the specified clock rate. Performed automatically, the compile results are reported to the TimeQuest timing analyzer in the Quartus development software. Using the Advanced Blockset results in high-performance FPGA implementations that can exceed 400 MHz.

Latency Constraint

Latency is not inherently deterministic, as the Advanced Blockset adjusts it to accommodate the requested clock rates and channel counts. However, features are available to manage or constrain latency on a design-block subset basis, allowing the use of perform-design exploration and trade-offs with unconstrained latency. Once a design configuration and performance level is selected, the latency is constrained to allow for predictable interfacing to the remainder of the surrounding implementation. If excess latency is selected for future design flexibility, the Advanced Blockset adds output register stages to match.

This capability allows a design to be reused easily by updating to add channels, change system clock rate, and porting to other FPGA devices. Not just limited to FFTs and FIR filters, the Advanced Blockset provides IP blocks to allow rapid and highly optimized design of interpolating and decimating FIR filters, cascaded-integrator comb (CIC) filters, IIR filters, radix-2 FFTs, non-radix-2 FFTs, and many other DSP functions. It can also be used to build complex subsystems, such as a multichannel FFTs, digital upconversion or downconversion chains, correlators, and crest factor reduction (CFR) algorithms. The high-performance, multichannel datapaths used in military radar, broadcast edge QAM, wireless basestations, and medical imaging are just a few applications where the Advanced Blockset delivers significant design team productivity and flexibility gains. There is a choice of using numerous and varied IP blocks in the Advanced Blockset library or to build algorithms out of primitive blocks, as shown in the multi-channel IIR filter example in Figure 5.

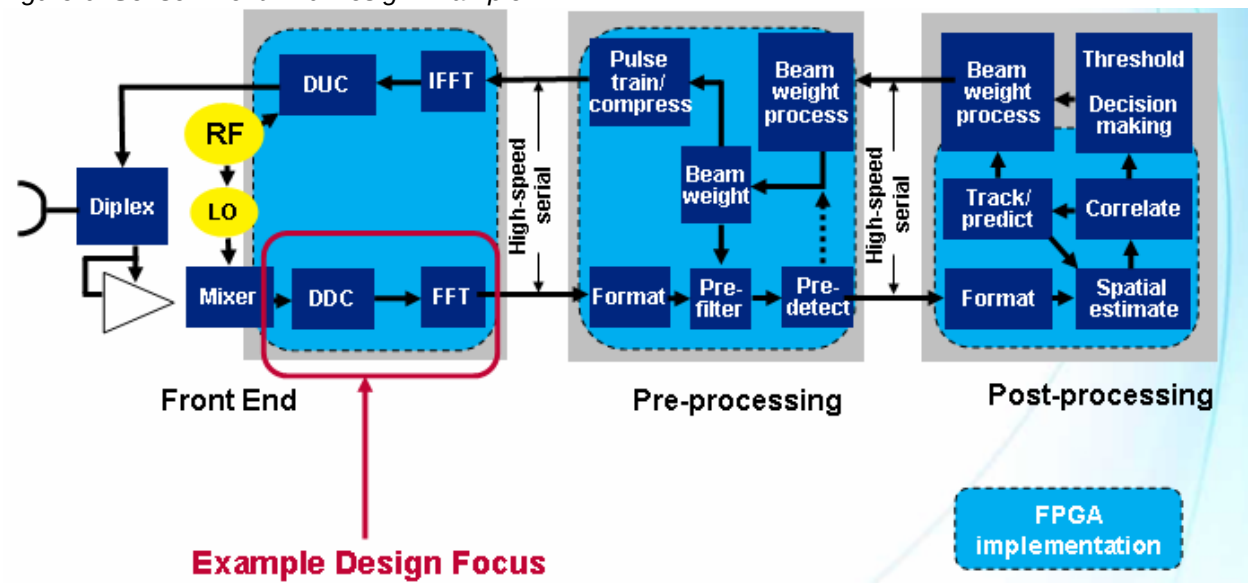
Figure 5. Multi-Channel IIR Filter Design Example



Conclusion

When building a system composed of high-performance FIR filter chains, FFTs, beamforming antennas, time, frequency, and spatial estimations, and many other complex algorithms, implementing with the Advanced Blockset results in reduced design time and performance. As it is possible to insert Advanced Blockset implemented subsections within a larger FPGA design, customers can experiment with this design methodology while minimizing risk due to disruption of existing design flow. The capabilities are such that even if the Simulink tool is not a part of the current design simulation, consideration should be given to access this highly optimized FPGA synthesis technology. Many design examples, including a military sensor front end (Figure 6) that interfaces to multiple 2.8-GHz ADCS with a system clock of 350 MHz, are contained within the Advanced Blockset. This is an ideal section of a military sensor design to begin adopting the Simulink/Advanced Blockset design flow. Available today, an engineering evaluation of the Advanced Blockset can be easily arranged by any Altera FAE.

Figure 6. Sensor Front-End Design Example



Further Information

- DSP System Design With DSP Builder (ODSP1110), 1-Hour Online Course: www.altera.com/education/training/courses/ODSP1110
- DSP System Design With DSP Builder Using the Advanced Block Set (ODSP1115), 1-Hour Online Course: www.altera.com/education/training/courses/ODSP1115
- Designing With DSP Builder Advanced Blockset (IDSP220), 8-Hour Instructor-Led Course: www.altera.com/education/training/courses/IDSP220

Acknowledgements

- Michael Parker, Sr. Technical Marketing Manager, Altera Corporation

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