ALTERA®

White Paper

FPGAs Provide Reconfigurable DSP Solutions

Introduction

The growing digital signal processing (DSP) market includes rapidly evolving applications such as 3G Wireless, voice over Internet protocol (VoIP), multimedia systems, radar and satellite systems, medical systems, image-processing applications and consumer electronics. These applications cover a broad spectrum of performance and cost requirements.

DSP processors are used for implementing many of these DSP applications. Although DSP processors are programmable through software, the DSP processor hardware architecture is not flexible. Therefore, DSP processors are limited by fixed hardware architecture such as bus performance bottlenecks, a fixed number of multiply accumulate (MAC) blocks, fixed memory, fixed hardware accelerator blocks, and fixed data widths. The DSP processor’s fixed hardware architecture is not suitable for certain applications that might require customized DSP function implementations.

FPGAs provide a reconfigurable solution for implementing DSP applications as well as higher DSP throughput and raw data processing power than DSP processors. Since FPGAs can be reconfigured in hardware, FPGAs offer complete hardware customization while implementing various DSP applications. Therefore, DSP systems implemented in FPGAs can have customized architecture, customized bus structure, customized memory, customized hardware accelerator blocks, and a variable number of MAC blocks.

Despite these benefits, one reason FPGAs have not found wider acceptance in the DSP market is the absence of a viable C code-based design flow that does not require knowledge of FPGA architecture nor hardware description language (HDL). Historically, DSP programmers accustomed to software-based design face a design flow barrier when switching to an FPGA-based solution. However, Altera has introduced new design tools and hardware features that alleviate the design flow problem by incorporating a C code-based design-flow option that mirrors the traditional DSP design flow.

Trends in the DSP Processor Architecture

The fundamental difference between a DSP processor and a generic processor is the DSP processor’s hardware multiply-accumulate (MAC) block and specialized memory and bus structures to facilitate frequent data access commonly found in DSP applications.

The MAC operation is usually the performance bottleneck in most DSP applications. DSP processor vendors incorporate MAC blocks in their architecture to minimize this performance bottleneck. Some DSP processor vendors have also tried adding multiple MAC blocks to their architecture to boost the overall multiplier bandwidth. For example, the TMS320C6411 device from Texas Instruments can calculate up to eight 8 × 8-multiplication results in a single clock cycle.

While adding more MAC units may provide more DSP throughput, the processor falls behind in raw data processing power for certain data-intensive DSP functions such as Viterbi encoder/decoder and FIR filters. To work around this problem, DSP processor vendors have also tried incorporating a hardware accelerator (coprocessor) block such as the Viterbi coprocessor, turbo coprocessor and the enhanced filter coprocessor. While such coprocessor blocks provide high DSP throughput, they do not cater to all DSP applications. Most DSP applications cannot benefit from the DSP vendors' predefined hardware accelerator blocks. Additionally, such hardware accelerator blocks are fixed, do not allow for any level of customization for the specific design needs, and can quickly become obsolete in today's evolving standards. DSP processor vendors also incorporate certain custom instructions that can take advantage of the architectural modifications seen in DSP processors.

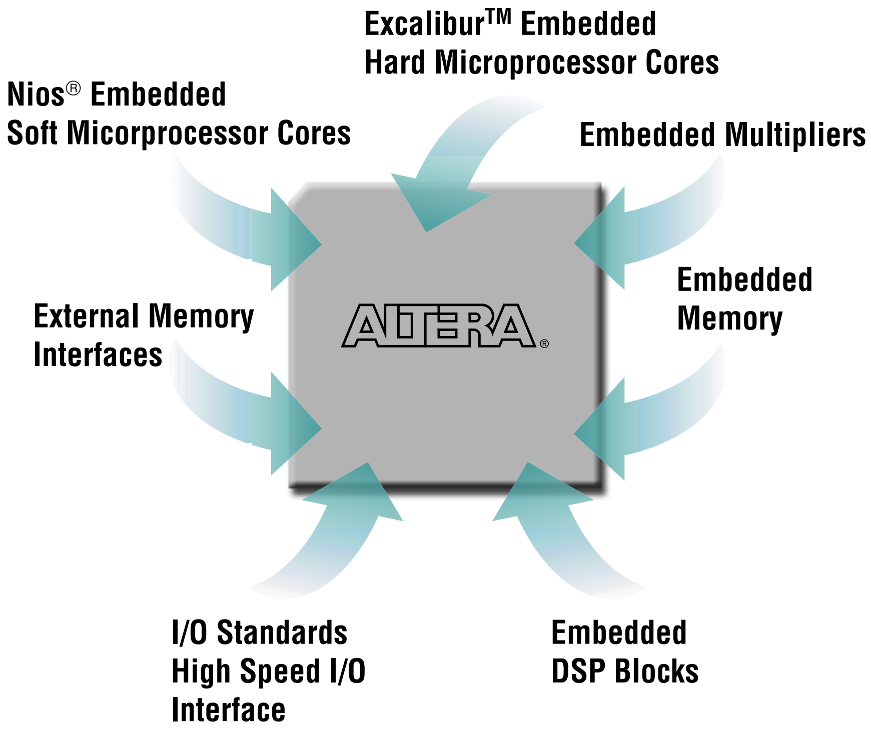
Trends in FPGA Architecture Features

FPGA devices consist of logic elements (LEs) and memory that can be configured to operate in different modes corresponding to a different functionality. This hardware flexibility allows the designer to implement any hardware design described using a suitable hardware description language (HDL) such as VHDL or Verilog HDL. Thus the same FPGA can implement a DSL router, a DSL modem, a JPEG encoder, a digital broadcast system, or a backplane switch fabric interface.

With the introduction of high-density FPGAs, such as Altera's Stratix™ FPGA family, that incorporate various embedded silicon features, designers can now implement complete systems inside an FPGA, creating a system on a programmable chip (SOPC) implementation. FPGA vendors have also started incorporating embedded silicon features that are ideal for DSP applications such as embedded memory, DSP blocks, and embedded processors that are well-suited for implementing DSP functions such as FIR filters, FFTs, correlators, equalizers, encoders, decoders, and arithmetic functions.

Figure 1 highlights the various DSP-related features available in different Altera® FPGA device families.

Figure 1. DSP Related Features in Altera FPGA Devices


Excalibur™ Embedded Hard Microprocessor Cores
Nios® Embedded Soft Micorprocessor Cores
Embedded Multipliers
Embedded Memory
External Memory Interfaces
ALTERA®
I/O Standards High Speed I/O Interface
Embedded DSP Blocks

Embedded DSP Functionality & Memory

FPGA vendors incorporate embedded DSP features in their devices, such as the DSP block in Stratix devices. The embedded DSP blocks also provide other functionality such as accumulation, addition/subtraction, and summation that are common arithmetic operations in DSP functions. For example, Stratix device DSP blocks offer up to 224 multipliers that can perform 224 multiplications in a single clock cycle. Compared to DSP processors that only offer a limited number of multipliers, Altera FPGAs offer much more multiplier bandwidth. Since one determining factor of the overall DSP bandwidth is the multiplier bandwidth, the overall DSP bandwidth of FPGAs can be much higher than the DSP processors. For example, Stratix device DSP blocks can deliver 70 GMACS of DSP throughput while leading DSP processors available today can deliver only up to 4.8 GMACS.

Various DSP applications use external memory devices to manage large amounts of data processing. The embedded memory in FPGAs meets these requirements and also eliminates the need for external memory devices in certain cases. For example, the Stratix device family offers up to 10 Mbits of embedded memory through the TriMatrix™ memory feature.

Embedded Processors

Embedded processors in FPGAs provide overall system integration and flexibility while partitioning the system between hardware and software. Designers can implement the system′s software components in the embedded processors and implement the hardware components in the FPGA's general logic resources. Altera devices provide a choice between embedded soft core processors and embedded hard core processors.

Designers can implement soft core processors such as the Nios embedded processor in FPGAs and add multiple system peripherals. The Nios processor supports a user-determinable multi-master bus architecture that optimizes the bus bandwidth and removes potential bottlenecks found in DSP processors. Designers can use multi-master buses to define as many buses and as much performance as needed for a particular application. Off-the-shelf DSP processors make compromises between size and performance when they choose the number of data buses on the chip, potentially limiting performance.

The ARM embedded processor available in the Excalibur device family features pre-defined, pre-optimized system peripherals such as SDRAM, memory controllers, and UARTs, and allow designers to configure their systems.

Soft embedded processors in FPGAs provide access to custom instructions such as the “MUL” instruction in Nios processors that can perform a multiplication operation in two clock cycles using hardware multipliers.

Hardware Acceleration in FPGAs

FPGA devices provide a flexible platform to accelerate performance-critical functions in hardware because of the configurability of the device’s logic resources. Unlike DSP processors that have predefined hardware accelerator blocks, FPGAs can implement hardware accelerators for each application, allowing the designer to achieve the best performance from hardware acceleration. The designer can implement hardware accelerator blocks by designing such blocks using parametrizable IP functions or from scratch using HDL. Altera and its Altera Megafunction Partner Program (AMPP℠) partners offer the following types of IP cores for hardware acceleration and data path design:

- General cores (e.g., FIR, IIR, NCO)
- Image-processing cores (e.g., JPEG, DCT)
- Modulation cores (e.g., QPSK, Equalizer)
- Encryption cores (e.g., DES, Rjindael)
- Error-correction cores (e.g., Viterbi, Turbo, CRC)

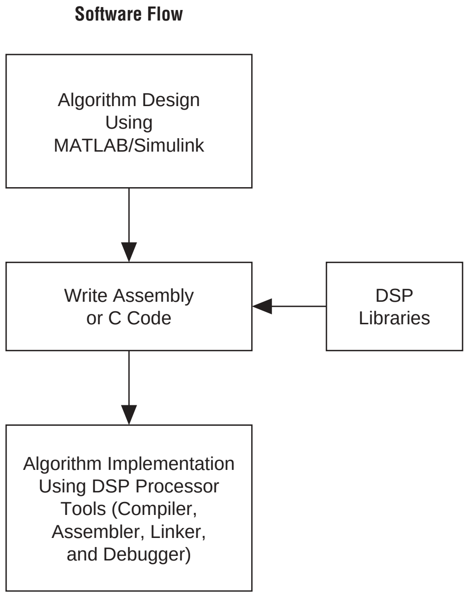
Each of these functions are parameterized (using the MegaWizard® Plug-In Manager) to design the most efficient hardware implementation for a given set of parameters. This provides maximum flexibility, allowing designers to customize IP without changing a design's source code. Designers can integrate a parameterized IP core in any hardware description language (HDL) or netlist file generated using any EDA tool. The designer can also port the IP to new FPGA families, leading to higher performance and lower cost.

The flexibility of programmable logic and soft IP cores allows designers to quickly adapt their designs to new standards such as the Wireless 802.11a, Wireless Broadband Working Group 802.16, and HiperLAN/2 without waiting for long lead times usually associated with DSP processors.

Software Design Flow with DSP Processors

Figure 2 highlights the typical software design flow that DSP programmers follow. DSP designers use algorithm development tools such as MATLAB to optimize DSP algorithms and Simulink for system-level modeling. The algorithms and the system-level models are then implemented in C/C++ or Assembly code using a standard integrated development environment, such as the Code Composer Studio from Texas Instruments, that provides design, simulation, debug, and real-time verification tools. Designers can use standard C-based DSP libraries to shorten design cycles and derive the benefits of design re-use.

Figure 2. Software-Based DSP Design Flow

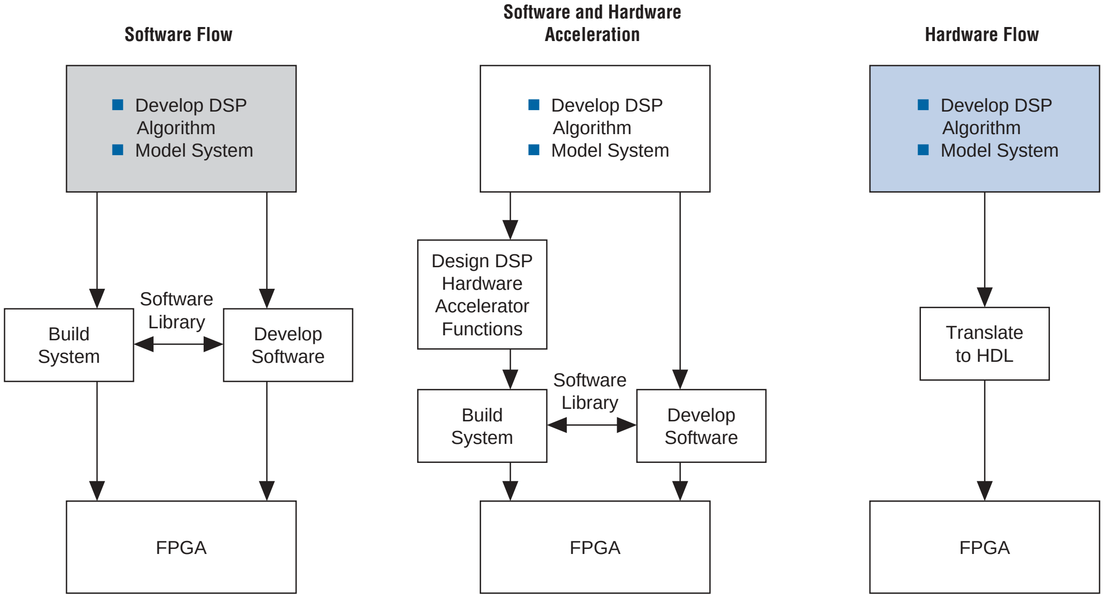

Software Flow
Algorithm Design Using MATLAB/Simulink
Write Assembly or C Code
DSP Libraries
Algorithm Implementation Using DSP Processor Tools (Compiler, Assembler, Linker, and Debugger)

DSP Design Flow in FPGAs

Traditionally, DSP designers had to implement their systems in FPGAs using the hardware flow based on a HDL language such as Verilog HDL and VHDL. New DSP tools such as DSP Builder, SOPC Builder, and a complete software development platform now enable DSP designers to follow a software-based design flow while targeting FPGAs.

Figure 3 outlines the various design-flow options available for FPGAs.

Figure 3. FPGA-Based DSP Design Flow Options



Software Flow in FPGAs

Altera FPGAs with embedded processors support a software-based design flow. Altera provides software development tools including the GNU Pro toolset for compiling, debugging, assembling and linking software designs. These software designs can then be downloaded to an FPGA using either on-chip RAM or an external memory device.

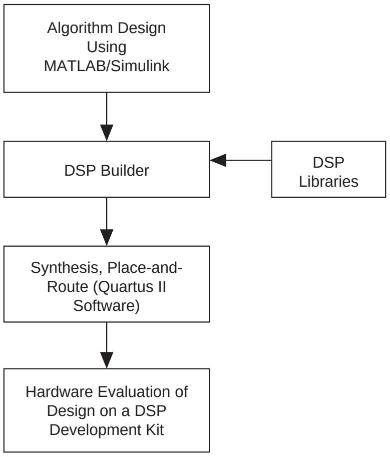
Software Combined with Hardware Acceleration

Embedded processors and hardware acceleration offer the flexibility, performance, and cost effectiveness in a development flow that is familiar to software developers. A DSP designer can combine a software design flow along with hardware acceleration. In this flow, the software developer first profiles C code and identifies the functions that are the most performance-intensive. Then, the developer can use Altera's DSP IP or develop their own custom instruction to accelerate those tasks in the FPGA. The system control code along with the other low-performance DSP algorithms can be run on a Nios embedded processor.

Altera also provides system-level tools such as SoPC Builder for system-level partitioning and integration. Designers can use SOPC Builder to build entire hardware systems by combining the embedded processor, such as a Nios embedded processor, system peripherals, as well as IP MegaCore® functions.

Altera's DSP Builder tool provides an interface from Simulink directly to the FPGA hardware (see Figure 4). The DSP Builder tool simplifies hardware implementation of DSP functions, provides a system-level verification tool to the system engineer who is not necessarily familiar with HDL design flow, and allows the system engineer to implement DSP functions in FPGAs without learning HDL. Additionally, designers can incorporate the designs created by DSP Builder into a SOPC Builder system for a complete DSP system implementation.

Figure 4. DSP Builder-Based Design Flow for Altera FPGAs


Algorithm Design Using MATLAB/Simulink
DSP Builder
DSP Libraries
Synthesis, Place-and-Route (Quartus II Software)
Hardware Evaluation of Design on a DSP Development Kit

Hardware Design Flow

DSP designers can also develop a pure hardware implementation of their DSP system using an HDL-based design flow. Altera provides a complete set of FPGA development tools including the Quartus® II software and interfaces to other EDA tools such as Synopsys, Synplify, and Leonardo Spectrum. These tools enable hardware design, simulation, debug, and in-system verification of the DSP system. The suite of pre-optimized DSP IP MegaCore functions can simplify this development process. DSP engineers can also follow the DSP Builder design flow shown in Figure 4 and implement hardware-only DSP systems in FPGAs without learning HDL.

Benefits of FPGAs in DSP Designs

FPGA devices provide a reconfigurable DSP solution for various DSP applications. FPGA devices incorporate a variety of embedded features such as embedded processors, DSP blocks, and memory blocks. These device features provide very high DSP capability in FPGAs compared to DSP processors. Using FPGAs, DSP designers can customize their hardware for optimal implementation of their applications.

Using embedded processors such as the Nios embedded processor, FPGAs also offer a software-based design flow similar to the traditional DSP software design flow. In addition, FPGAs offer a design flow using the SoPC Builder tool that enables a software design flow to be combined with hardware acceleration. Using this design flow, a DSP designer can implement a complete DSP system in an FPGA and thereby develop a cost-effective, high-performance DSP system.

Altera FPGAs also offer a design flow based on the DSP Builder tool that allows a DSP system engineer designing in MATLAB/Simulink to implement a system on an FPGA without learning HDL. The DSP Builder tool flow can also be combined with the SOPC Builder tool flow for implementing a complete DSP system.

Hence FPGAs from Altera provide the benefits of system integration, flexibility while partitioning the system, lower system costs, and higher performance when compared to pure DSP processor based implementations.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com