White Paper

Using PLDs for High-Performance DSP Applications

Introduction

Design engineers face the challenge of designing increasingly high performance communications systems in less time with fewer resources. Additionally, these designers must consider rapidly emerging/changing technologies. The wide range of performance needs for today's applications requires a solution that is flexible and easy to implement. One solution is programmable logic devices (PLDs), which provide flexibility, high performance, and fast time-to-market.

Designers are using PLDs more frequently for digital signal processing (DSP) applications. PLDs are found in a wide range of products, from high-performance wired applications such as digital subscriber line access multipliers (DSLAMs) to emerging 3G wireless applications. Designers use PLDs in DSP applications because of their flexibility and hardware-based performance.

To harness the true capabilities of PLDs, designers need a complete design environment with which they can go from system architecture to hardware implementation in a short time. A seamless design flow, like the one provided with the Altera® DSP Builder and intellectual property (IP) MegaCore® functions, extends the performance and cost benefits of programmable logic.

Comparing DSP Processors, ASSPs, ASICs & PLDs

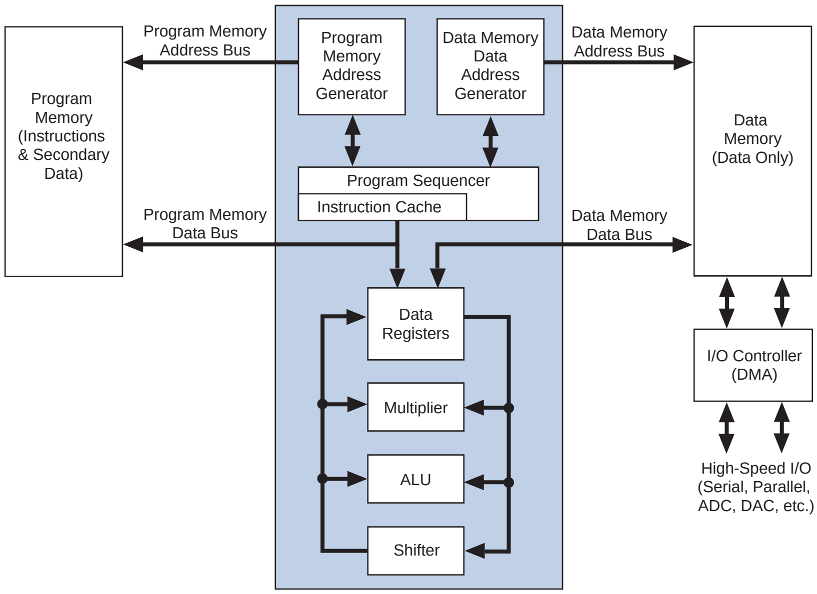
Engineers have various options when implementing DSP applications, including:

- DSP processors
- Application-specific integrated circuits (ASICs)
- Application-specific standard products (ASSPs)
- PLDs

Traditionally, designers chose DSP processors for digital signal processing applications (DSP). DSP processors have a general-purpose architecture, shown in Figure 1, that makes them flexible for a variety of applications. However, their flexibility ultimately limits their system performance.

WP-STXDSP-1.0

Figure 1. General DSP Processor Architecture



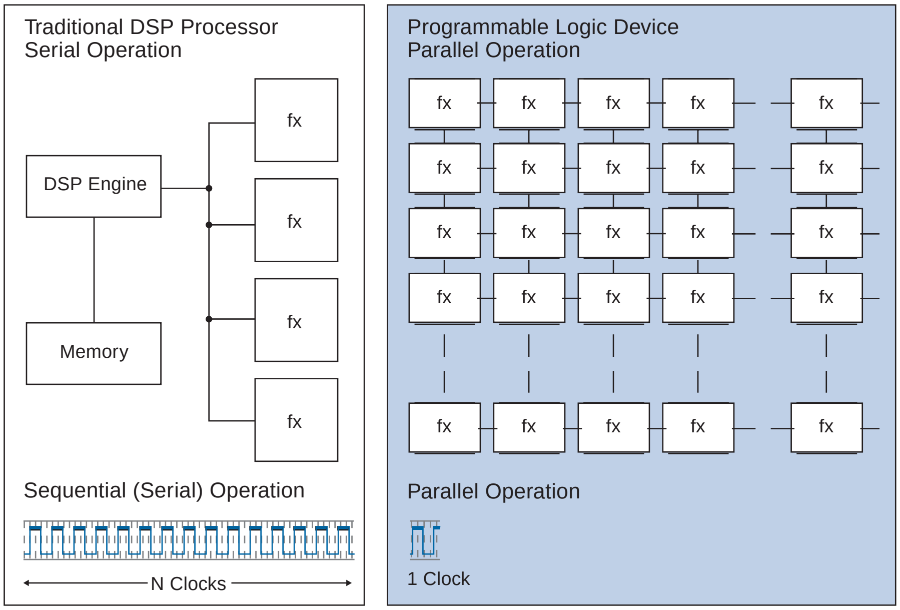
Source: The Scientist and Engineer's Guide to Digital Signal Processing

Historically, DSP processors have had only one multiplier, but today, some have up to 8 multipliers. They use iterative looping for more than 2 to 8 multiplications and need additional clock cycles to calculate the result. Therefore, DSP processors are most suited for back-end signal processing at low data rates. Many DSP processors have multipliers with special instructions to speed up the math calculations, however, they lack real-time performance. DSP processors are flexible and can be used for filtering or modulating applications by changing the processor's software code.

ASSPs and ASICs, which are designed to implement a specific function, have better performance than DSP processors for a low cost. These qualities make them attractive for designers. Because ASSPs are semi-custom integrated circuits that perform specific functions, such as finite impulse response (FIR) and infinite impulse response (IIR) filters, their performance is better than other hardware solutions on a similar process technology. However, ASSPs are inflexible and must be redesigned if the DSP application changes. ASICs provide a customizable, low-cost solution. However, they have long lead times—typical design cycles are 1 to 1.5 years—and require a minimum purchase quantity. Small design changes incur additional non-recurring engineering costs and result in a longer design cycle.

PLDs offer compelling advantages over DSP processors, ASSPs and ASICs. Designers can configure PLD logic to process complex routines in parallel or in serial like DSP processors (see Figure 2). In parallel, they offer greater performance than DSP processors by executing the equivalent of hundreds of instructions at once. Unlike ASSPs and ASICs, PLDs provide the flexibility to make design changes without sacrificing time-to-market.

Figure 2. Architectural Differences between PLDs & DSP Processors



Innovative Support for DSP Algorithms

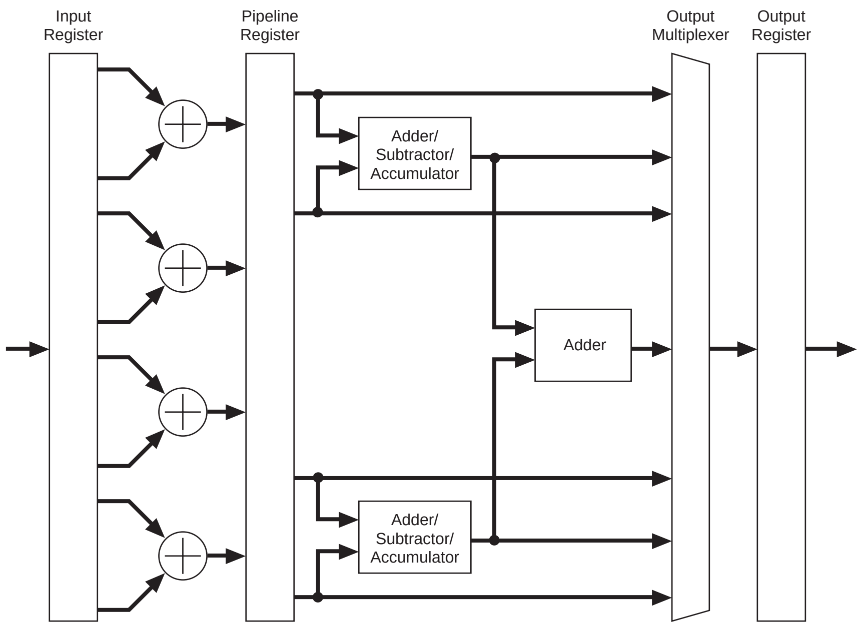
DSP applications use algorithms such as fast Fourier transform (FFT), FIR, IIR, matrix multiplication, correlation, etc. Most of these algorithms are mathematical calculations that combine multiplication followed by an addition (e.g., $y = a \times b + c \times d$), which is called a multiply-accumulate (MAC) operation.

Altera Stratix™ PLDs combine the flexibility of programmable logic with the raw performance and capabilities of dedicated functional blocks, making the devices an ideal choice for next-generation wireless infrastructure systems. Stratix DSP blocks have hardware multipliers, adders/subtractors, accumulators, and pipeline registers. Flexible, efficient, and optimized for DSP applications requiring high data throughput, DSP blocks are ideal for the wireless communication, telecommunication, video, and image processing markets.

Stratix hardware multipliers offer high performance and low resource utilization when implementing mathematical calculations. The DSP blocks contain dedicated multipliers of varying widths and the multiplier outputs can feed an adder/subtractor or an accumulator. Figure 3 shows the Stratix DSP block architecture. Additionally, Stratix embedded memory blocks can store data for DSP applications such as FIR filter coefficients.

For more information on the DSP block, refer to *AN 214: Using DSP Blocks in Stratix Devices*.

Figure 3. Stratix DSP Block Architecture



Design Flow

DSP system design in Altera PLDs requires both high-level algorithms and hardware description language (HDL) development tools. The Altera DSP Builder integrates these tools by combining the algorithm development, simulation, and verification capabilities of The MathWorks MATLAB and Simulink system-level design tools with Altera development tools, HDL synthesis, and simulation.

To win more DSP designs, PLD vendors need to win over DSP engineers by providing a familiar design flow. Altera provides one such tool, the DSP Builder, which interfaces The MathWorks industry-leading, system-level DSP tool Simulink with Altera's Quartus II development software. DSP Builder provides a seamless design flow in which designers perform algorithmic design in the MATLAB software, system integration in the Simulink software, and port the design to hardware description language files for use in the Quartus II software.

Using the DSP Builder, designers can generate an RTL design and RTL testbench from Simulink automatically. These pre-verified RTL output files are optimized for use in Altera's Quartus II design software so that the designer can perform quick timing and simulation comparisons. The design flow also enables floating-point to fixed-point analysis. With this simple and intuitive development flow, designers do not need experience using programmable logic design software.

Designers can use the signal processing IP blocks provided with the DSP Builder to create a hardware implementation of a Simulink system model. The DSP Builder contains bit- and cycle- accurate Simulink fixed-point blocks, which cover basic operations such as arithmetic or storage functions as well as complex functions such as forward error correction, filtering, and modulation.

Refer to the *DSP Builder Quartus II & MATLAB/Simulink Interface User Guide* for more information.

Altera DSP IP Portfolio

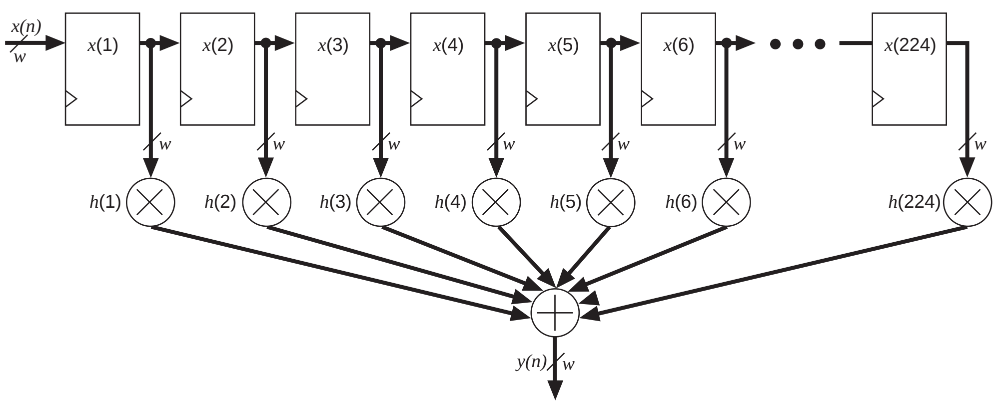
Algorithms that can benefit from PLD performance include filtering, forward error correction, modulation/demodulation, and encryption. Altera has a comprehensive portfolio of standard DSP functions, optimized for Altera PLDs, that designers can license to build system-on-a-programmable-chip (SOPC) designs. Alternatively, designers can use DSP IP cores individually to improve systems that require focused performance enhancements. For systems that require higher throughput, designers can instantiate Altera DSP IP cores using dedicated hardware in parallel.

With the performance advantages of parallel processing and the traditional flexibility of PLDs, DSP cores such as the FIR Compiler and Reed-Solomon Compiler are ideal for emerging applications such as multi-channel multipoint distribution services (MMDS) and orthogonal frequency division multiplexing (OFDM) systems. For example, the Altera Reed-Solomon MegaCore function decodes at rates up to 1 Gbps for 8-bit symbols. With nominal buffering and control overhead, a Reed-Solomon solution decodes at a rate of over 10 Gbps. In contrast, preliminary Texas Instruments benchmarks show that the C64xx DSP processor requires approximately 1,095 cycles to decode one Reed-Solomon codeword (source: Texas Instruments *TMS320C6414 Fixed-Point Digital Signal Processor* data sheet and TMS320C64x DSP Benchmarks page on the Texas Instruments web site). At 300 MHz, the C64xx processor decodes approximately at a rate of 450 Mbps using 100% of the device's available processing power.

Design Example: FIR Filter Implemented in Stratix Devices

The following example—an 8-bit 224-tap FIR—illustrates the performance advantages of Stratix devices for DSP applications. Each register provides the unit sample delay. The delayed inputs are multiplied with their filter coefficients and added together to produce the output. See Figure 4.

Figure 4. Typical FIR Filter



The FIR filter operation can be represented by the equation:

$$y(n) = \sum_{1}^{n} nx(n)h(n)$$

where $x(n)$ and $h(n)$ are the nth sample values of the input signal and the filter coefficient.

This 224-tap filter can be implemented in Stratix DSP blocks and the design fits into a single Stratix PLD. The design uses only a single clock cycle for the data output. In contrast, the same filter design requires multiple DSP processors because a single DSP processor has only 2 to 8 multipliers. Table 1 shows the performance results of these two implementations. The Stratix device improves overall system performance almost twelve times.

Table 1. Stratix Device & DSP Processor Performance Comparison

Requirement	Stratix Device (EP1S120)	TI 320C64x Device *(1)*
Number of Taps	224	224
Number Of Multipliers	224 (28 DSP Blocks)	Maximum of 8
Internal Clock Speed	250 MHz	600 MHz
Clock Cycles Needed to Compute the Result	1	28
GMACs per second	56 GMACs	4.8 GMACs

Note:
(1) Source: Texas Instruments *TMS320C6414 Fixed-Point Digital Signal Processor* data sheet.

Conclusion

Altera PLDs, such as Stratix devices, are very flexible and provide an efficient solution for hardware implementation of DSP application. Stratix devices have flexible, dedicated DSP blocks that support a wide variety of configurations, which makes it easy to implement DSP applications. Additionally, the DSP blocks have efficient routing that provides fast performance. The high-performance DSP blocks and on-chip memory allows designers to maximize system throughput. PLDs also support parallel processing, making them a powerful tool for increasing overall system performance.

To support a seamless design flow, Altera provides valuable tools, such as MegaCore functions and the DSP Builder, that designers can use to target a system design to PLDs. The Altera signal processing portfolio includes proven, high-performance, standard functions created to help engineers meet these design challenges and to implement a solution on a single Altera PLD. The DSP Builder provides a seamless design flow in which designers can perform algorithmic design in MATLAB, carry out the system integration in Simulink, and port the result to an HDL design for use with the Quartus II software.

More Information

For more information, refer to the following documents:

- *AN 214: Using the DSP Blocks in Stratix Devices*
- *AN 215: Implementing High-Performance DSP Functions in Stratix Devices*
- *Stratix Programmable Logic Device Family Data Sheet*

References

Smith, Steven W. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. California Technical Publishing. San Diego, CA. 1999.

Texas Instruments. *TMS320C6414 Fixed-Point Digital Signal Processor.* Houston, TX. September 2001.

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