Digital Signal Processing

in FLEX Devices

January 1996, ver. 1 Product Information Bulletin 23

Introduction

Designers of digital signal processing (DSP) applications are often forced to choose between flexibility and performance due to the limited solutions available. At one extreme, DSP processors are cheap, flexible, and can be used in a variety of applications. However, these processors only deliver moderate real-time performance due to sequential processing through the multiplier-accumulator (MAC) bottleneck. Designers who require high-throughput, real-time processing must use multiple DSP processors at a considerable cost. At the other extreme, fixed-function DSP devices and ASICs offer improved performance, but are inflexible. Only a limited variety of fixed-function DSP devices are available, and ASICs require long lead times for development, prototyping, and production.

Industrial imaging applications—such as in-line, high-resolution inspection equipment—demand real-time filtering and template matching, which requires a performance of thousands of MIPS. DSP functions are used increasingly in RF processing for applications such as spread-spectrum LANs, RF spectrum analysis, and radar. These applications require powerful, flexible DSP solutions.

Complex programmable logic devices (CPLDs) provide the performance and flexibility required for DSP applications. Because DSP algorithms are optimally mapped to the device architecture, CPLD performance can significantly exceed DSP processor and ASIC performance. CPLDs also allow for function modification and instant prototyping. For example, a DSP processor can implement an 8-tap finite impulse response (FIR) filter at 5 million samples per second (MSPS). FLEX devices can implement the same FIR filter at over 100 MSPS.

This product information bulletin compares DSP solutions using FLEX devices and traditional DSP design approaches.

Traditional DSP Solutions

DSP involves performing mathematical computations on a data stream that represents a signal. The traditional DSP solutions are summarized below.

- *Microprocessors*—Historically, general-purpose microprocessors performed DSP algorithms. Microprocessors are still used in some less demanding applications such as providing sound for PCs.

- *DSP processors*—As designers demanded higher data rates and more complex algorithms, the DSP processor was developed. The DSP processor usually has a RISC-type architecture and a fast MAC function, and performs at 3 to 50 MIPS. DSP processors can perform in one cycle an operation that requires many cycles in a general-purpose microprocessor.

  DSP processors have complete algorithm flexibility, but have limited performance because algorithms are implemented by sequential MAC operations. More recent DSP processors have several processors on a single die. However, these processors are expensive and lack real-time processing power for growing RF and video applications.

- *Application-specific standard products (ASSPs)*—Performance requirements led to the development of dedicated, function- or algorithm-specific devices, such as FIR filters and fast Fourier transform (FFT) processors. ASSPs permit a function to be optimized in hardware, at the cost of flexibility. Due to a narrow range of applicability—and therefore low volume—many ASSPs are not available in state-of-the-art process technology and often require long lead times.

- *ASICs*—High performance usually requires a custom ASIC. For example, critical DSP functions can be custom-tailored in hardware or performed in parallel to increase performance. Core modules for common or standardized DSP functions such as MPEG decoding simplify the ASIC design task. However, non-recurring engineering (NRE) costs, non-interactive prototyping, and the cost of inflexibility throughout the lifetime of the product make ASICs desirable only for stable, high-volume designs.

FLEX DSP

FLEX devices provide the high performance of dedicated or custom hardware while maintaining the full flexibility of a DSP processor. Because FLEX devices can be reconfigured, the entire DSP implementation flow can be rearranged. Algorithms that were traditionally regarded as a sequence of single instructions can be analyzed to determine possible parallelism. In-circuit reconfigurability (ICR) permits algorithm-specific minimization and adaptation.

Implementing DSP functions in FLEX devices provides the following advantages:

- *Parallelism*—Implementing DSP algorithms in hardware permits parallelism, leading to significantly higher performance than typical DSP processors. The scale of a function can often be increased (e.g., by adding taps to a FIR filter) with minimal performance degradation. In contrast, the sequential processing of traditional DSP processors causes performance to degrade linearly as the size of the computation increases.

- *Efficiency*—Many DSP algorithms such as FIR filters and video color space converters involve multiplication by a constant. The FLEX architecture contains look-up tables (LUTs), which can simplify multiplication. The LUT stores the precomputed values, eliminating the area and performance cost of a full multiplier. In contrast, ASICs and ASSPs require a full multiplier.

- *Pipelining*—Adding pipelined registers to a design permits latency to be traded for higher throughput. Because FLEX devices have registers at the output of every logic cell, adding pipelined registers does not require additional device resources.

- *In-circuit reconfigurability (ICR)*—ICR permits the algorithm or function to be changed while operating in-circuit. This feature enables algorithm updates in the field and allows functions to vary according to control panel input.

- *Adaptability*—A device that can implement large internal RAM blocks, such as an Altera FLEX 10K device, can be used to implement real-time adaptive functions at a throughput that cannot be matched by conventional DSP solutions. FLEX devices can also be reconfigured to accommodate differing filter characteristics.

- *Specialized algorithms*—FLEX 10K devices have embedded array blocks (EABs) that can implement specialized algorithms, including those that require the storage and rapid retrieval of intermediate values. Examples include transform functions such as FFT, two-dimensional FFT, and discrete cosine transform (DCT) functions.

Figure 1 compares the performance and flexibility of traditional and FLEX DSP solutions.

Figure 1. Performance & Flexibility of DSP Solutions

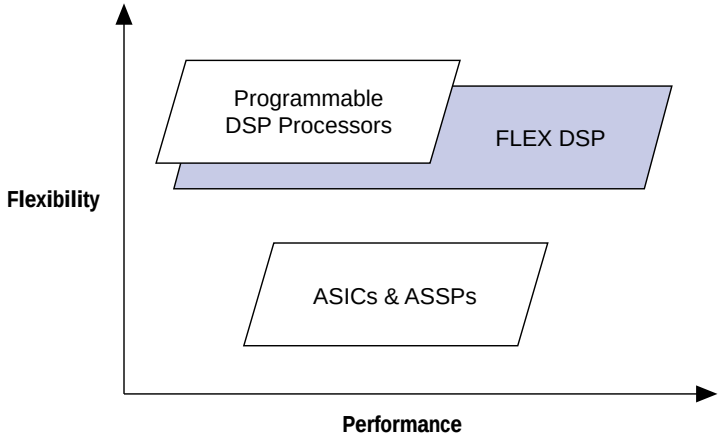
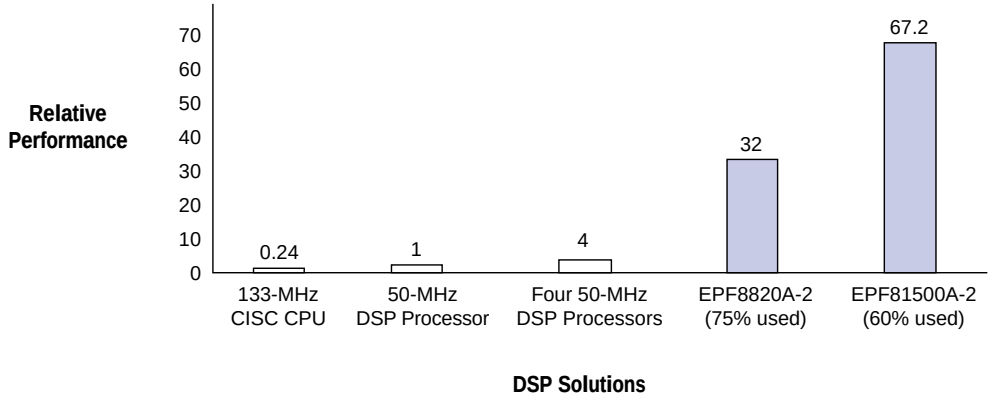


Figure 2 compares the relative performance for several implementations of a 16-tap, 8-bit FIR filter.

Figure 2. Performance Comparison for 16-Tap, 8-Bit FIR Filter



DSP Applications

FLEX devices can perform a variety of DSP functions, including FIR filters, infinite impulse response (IIR) filters, and arithmetic functions.

FIR Filters

Altera has developed parameterized, high-speed FIR filter functions that are optimized for the FLEX architecture. The LUT-based, register-intensive FLEX architecture combined with vector-processing algorithms results in extremely fast, efficient FIR filters.

Altera FIR filter megafuctions are available in the DSP Design Kit. Contact Altera Literature Services for more information.

FLEX devices can implement parallel and serial filters, allowing a tradeoff between silicon resources and performance. Parallel filters perform at a rate of up to 100 MSPS, enabling digital processing of RF-intermediate frequency (RF-IF) data. Serial filters require considerably less logic, and still perform at 5 to 6 MSPS—higher performance than a comparably priced DSP processor.

You can add more taps to the parallel filter with only a small performance tradeoff with additional parallel silicon resources. In contrast, DSP processors exhibit a linear decrease in performance as the number of taps increases. Table 1 shows the performance for an 8-bit FIR filter with increasing taps. The performance of the fully parallel implementation is nearly constant as the number of taps is increased.

Table 1. Fully Parallel 8-Bit FIR Filter in FLEX 8000A (-2 Speed Grade Device)

Number of Taps	FLEX 8000A Performance -2 Speed Grade (MSPS)	Equivalent MIPS (DSP Processor)
8	104	832
16	101	1,616
24	103	2,472
32	105	3,360

An 8-tap, 8-bit FIR filter in a FLEX device needs only 80% more silicon resources than one 8-bit × 9-bit fixed multiplier (see Table 2). This ratio is substantially lower than that of a fixed-hardware, full-multiplier implementation. This efficiency is due to FLEX reconfigurability, which allows designers to exploit the advantages offered by a fixed-coefficient implementation. Because FLEX devices can be reconfigured, filter coefficients or even topologies can be modified in-system.

Table 2. Silicon Resource Comparison

Function	Inputs & Outputs	FLEX 8000A Logic Cells
FIR filter	8-bit data, coefficients, 17-bit output	296
Fixed-point multiplier	8-bit × 9-bit data, 17-bit output	164

Refer to *Application Note 73 (Implementing FIR Filters in FLEX Devices)* and the *FIR Filters Functional Specification* for more information on creating FIR filters for FLEX devices.

Infinite Impulse Response (IIR) Filters

FLEX devices can efficiently implement IIR filters. For example, a LUT-based vector multiplier can be used to create a complete second-order section of an all-pole analog filter. The vector multiplier requires the same resources and operates at the same speed as a fixed-point multiplier. A Butterworth Chebychev-I filter can run at a rate of 25 MSPS and require only 139 logic cells.

Refer to "References" on page 8 of this product information bulletin for more information on IIR filters.

Arithmetic Functions

DSP algorithms can be implemented as a series of discrete mathematical operations. Entire algorithms can be implemented in an optimized structure, such as filtering algorithms. However, some applications require a simple, extremely fast design.

For example, an application may integrate gamma correction and color space conversion (e.g., YUV to RGB), which essentially requires a multiply and add algorithm. DSP processors use sequential processing and have an overhead for reading in the operands and writing the result through a single data port. Thus, a DSP processor may require four cycles (i.e., read, multiply, add, and write) to perform the algorithm, resulting in a 10-MIPS performance from a 40-MIPS DSP processor. The same algorithm would execute at 60 to 90 MIPS in a fully pipelined FLEX design.

To implement fixed-point multiplication in MAX+PLUS II, use `lpm_mult`, a function from the library of parameterized modules (LPM).

Table 3 shows examples of arithmetic functions for a non-pipelined design.

Table 3. Examples of Arithmetic Functions (Non-Pipelined) for FLEX 8000A Devices

Number	Function	Input *Note (1)*	Output *Note (1)*	Logic Cells	Performance (MHz)	
					A-2 Speed Grade	A-4 Speed Grade
Integer	Multiplier	8-bit	16-bit	164	20.4	13.0
	Adder/Subtractor	8-bit	8 plus carry	34	64.5	46.9
Floating-point	Multiplier	8M7E1S	16M7E1S	246	14.2	8.6
	Adder/Subtractor	8M7E1S	8M7E1S	162	14.3	9.2

Note:
(1) *M* is the mantissa, *E* is the exponent, and *S* is the sign.

FLEX vs. Other DSP Solutions

This section compares the DSP solutions for several applications.

FLEX Devices vs. DSP Processors

A DSP processor usually provides acceptable performance at a reasonable cost for complex algorithms on low-sample-rate data. However, many DSP functions, such as filtering, require the repetition of simple algorithms. FLEX devices can handle repetitive DSP functions with extremely high performance. In addition, when serialization and a slower device speed grade are used, FLEX devices can offer better performance than DSP processors at a lower cost. This advantage allows devices to replace DSP processors entirely, or to be used as DSP coprocessors.

FLEX Devices as DSP Coprocessors

A FLEX device can be used to free the DSP processor from implementing functions such as high-speed filters, allowing designers to use a less-expensive DSP processor to perform any remaining functions. For example, in a spread-spectrum, direct-sequence RF modem application, a FLEX 8000A device with a -4 speed grade can implement the receiver's correlation filter function at a chip rate over 60 MHz. A DSP processor can perform the remaining tasks, such as quadrature-phase shift key (QPSK) demodulation. The resulting DSP application can deliver six times the data rate as the DSP processor alone.

Go to *Technical Brief 4 (Using FLEX Devices as DSP Coprocessors)* for more information.

FLEX Devices vs. ASSPs & ASICs

Efficient vector multipliers coupled with the fast, predictable FastTrack Interconnect enables FLEX devices to perform substantially better than ASSPs or ASICs. Table 4 compares the performance for a 3 × 3 video convolver function. The ASSP suffers from the overhead of a full multiplier implementation; the lowest-speed-grade, least-expensive FLEX 8000A device offers over twice the performance. A typical DSP processor requires one cycle for each of the 9 MAC operations in addition to any overhead for reading data. Thus, a 50-MHz (25-MIPS) DSP processor performs at less than 2 MSPS.

Go to *Technical Brief 3 (FLEX Devices as Alternatives to ASSPs & ASICs)* for more information.

Table 4. Performance Comparison for 3 × 3 Video Convolver

Solution	Device/ Speed Grade	Performance (MSPS)	Relative Price *Note (1)*
ASSP	HSP48901	30	3.0
DSP processor	50 MHz	< 2	2.8
FLEX 8000 (139 logic cells)	A-4	62	1.0
	A-2	98	1.7

Note:
(1) The relative price refers to quantities of approximately 100 units.

Conclusion

The advent of CPLDs with densities of up to 100,000 gates and large, embedded memory blocks opens up new possibilities for cost-effective DSP. With system-level integration, FLEX devices can fully implement complex algorithms—for example, high-resolution discrete cosine transformations (DCT), proprietary compression, and adaptive algorithms—allowing them to move beyond high-speed, pre-filtering processor implementations. ICR permits the same silicon resources to perform multiple functions, such as compression/decompression, encoding/decoding, and error generation/checking. Likewise, as digital techniques continue to pervade the RF spectrum, FLEX devices provide flexibility and high performance.

References

Langhammer, Martin. ***Efficient Logic Synthesis Techniques for IIR Filters***. Boston: Proceedings of International Conference on Signal Processing Applications & Technology (ICSPAT) '95, October 1995.

Langhammer, Martin. ***Optimal Architecture for 1-D and 2-D DCT via Vector Processing Algorithm***. Boston: Proceedings of International Conference on Signal Processing Applications & Technology (ICSPAT) '95, October 1995.

2610 Orchard Parkway
San Jose, CA 95134-2020
(408) 894-7000
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 894-7104
Literature Services:
(408) 894-7144