



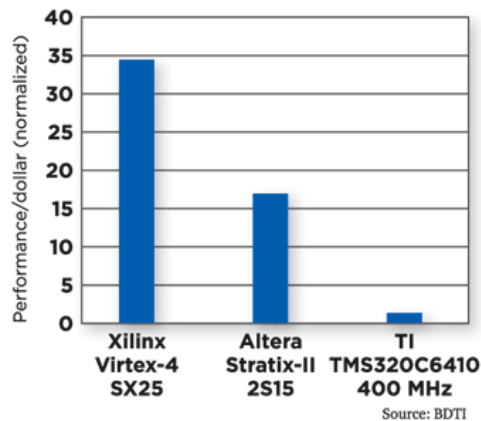
[Editor's Note: This article contains selected highlights from BDTI's new report, [FPGAs for DSP, Second Edition](#).]

In recent years, FPGAs (field-programmable gate arrays) have become increasingly attractive as signal processing engines, sometimes used alone and sometimes in conjunction with a processor chip. The largest FPGA vendors, Altera and Xilinx, have invested heavily in developing DSP-oriented chips and development tools. BDTI has just completed an in-depth study of these DSP-oriented FPGAs, including benchmarking using an OFDM receiver benchmark representative of telecommunications infrastructure applications. In this article we present insights gained from our benchmarking and analysis, focusing on the evolving role of FPGAs and other implementation technologies targeting digital-signal-processing-intensive applications.

If the advantages of FPGAs for signal processing can be boiled down to a single word, that word is flexibility. The flexibility of the FPGA compute fabric is key to how FPGAs achieve high throughput and cost-effectiveness: the FPGA designer can use the reconfigurable logic in an FPGA to form computation structures that are well matched to the needs of the application. In contrast, the user of a DSP or general-purpose processor has much less flexibility. The advantages of FPGA flexibility can be seen in the BDTI Communications Benchmark (OFDM)[™] results. In the FPGA implementations of the benchmark, highly parallel architectures were designed specifically for the application. These specialized architectures, in conjunction with the massive computational resources offered by high-end FPGAs, resulted in much higher throughput than that achieved by high-performance DSP processors. Due to their large throughput advantages the FPGAs were also able to deliver significantly lower cost per channel than the DSPs.

FPGAs outperform DSP rivals

BDTI benchmark performance ratings



Looking forward, it is interesting to consider how advances in IC fabrication technology benefit different kinds of chips. As the industry moves to 90 nm processes, 65 nm processes and beyond, the most obvious benefit is the ability to pack more circuits into a given silicon die. Processor designers have made use of this gain by designing more-complex processors that achieve higher performance through sophisticated instruction sets and microarchitectures, incorporating features such as deeper pipelines and multiple execution units.

But such architectural techniques quickly reach a point of diminishing returns. For example, adding execution units to a processor results in smaller and smaller performance and efficiency gains as the number of execution units increases. This is due to difficulties in finding and extracting suitable parallelism from applications (especially when the applications are expressed in inherently serial languages, like C), and to bottlenecks elsewhere in the processor, such as in data memory bandwidth. This is why we don't routinely see, for example, processors with 16 or 32 execution units. As a result, processor designers have largely used their expanded gate budgets to add increasing amounts of on-chip memory. Increased on-chip memory

can boost performance and efficiency, but again these increases quickly reach the point of diminishing returns.

The current interest in multi-core processors is largely driven by the recognition that the processor industry has reached the point of diminishing returns in scaling single-core processors. Unfortunately, with limited exceptions, the development process for mapping an application onto a multi-core processor differs in important ways from that associated with single-core processors, and tools and techniques for software development on multi-core processors are not nearly as rich, mature, and widely understood as those for single-core processors. This is the key obstacle to the rapid, widespread adoption of multi-core processors, and creates an opportunity for FPGAs. Because FPGAs use silicon in a relatively homogenous way, designers can take advantage of additional logic resources without changing their process of mapping applications onto the chips.

While the future of FPGAs looks bright, there are several major challenges hindering the penetration of FPGAs into DSP applications. One challenge is the complexity in mapping applications to FPGAs. Using an FPGA requires devising an architectural design that harnesses the overall resources of the FPGA in a way that matches the needs of the application. It then requires creation of an implementation of the architecture, which is typically done with hardware description languages at the register-transfer level (RTL). As a result, mapping a digital signal processing application onto an FPGA in an optimized manner requires substantially more effort than mapping the same application onto a DSP processor, assuming appropriately skilled engineers in both cases. In addition, iterations through the typical design-test-debug cycle tend to be more time-consuming with FPGAs, because of longer tool run times and a wider range of design possibilities.

FPGA vendors have responded to this challenge by investing heavily in high-level development tools and libraries of off-the-shelf design elements, which allow designers to work at a higher level of abstraction. These tools and libraries have certainly helped boost FPGA user productivity, but they have a long way to go before mapping an application to an FPGA will be as easy as mapping it to a DSP. Another major challenge for FPGA vendors is expanding the user base of FPGAs. While FPGAs have been in widespread use for many years, it remains true that engineers skilled in digital signal processing algorithm and system development are much more likely to be comfortable with the software-centric development paradigm associated with DSP processors than with the hardware-centric paradigm associated with FPGAs.

Today FPGAs play an increasing role in a wide range of DSP applications. We expect this trend to continue over the next several years. While we do not expect FPGAs to replace other types of chips altogether, we do expect FPGAs to replace DSPs and ASICs in a significant number of applications. In many applications, FPGAs will continue to be used alongside DSPs, general-purpose processors (GPPs), ASICs, and application-specific standard products (ASSPs).

More in-depth analysis of FPGAs and benchmark results for Altera and Xilinx FPGAs as well as high-end DSPs can be found in BDTI's new report, [FPGAs for DSP, Second Edition](#).

Related article:

- [FPGAs can outperform DSPs, says study](#)