A BDTI Analysis of the

Analog Devices ADSP-BF5xx

by the staff of
Berkeley Design Technology, Inc.

Contents of this summary include:

Introduction

The ADSP-BF5xx (Blackfin) is a family of 16-bit fixed-point dual-MAC processors from Analog Devices. The ADSP-BF5xx combines features typical of low-power DSPs with features traditionally associated with general-purpose processors, such as privilege modes and memory protection. The ADSP-BF5xx targets power-sensitive applications, such as cell phones; applications that require the functionality of both a DSP and a general-purpose processor, such as automotive applications; and computationally intensive applications, such as consumer video equipment.

The ADSP-BF5xx uses a mixed-width 16-/32-bit instruction set. These instructions can be combined to form 64-bit VLIW-style "multi-issue" instructions. Multi-issue instructions can include up to one 32-bit arithmetic instruction and up to two 16-bit move instructions. The ADSP-BF5xx provides single-instruction, multiple-data (SIMD) instructions, including a dual 16 × 16 multiply-accumulate and a variety of video-oriented, eight-bit ALU operations.

The ADSP-BF5xx is based on the Micro Signal Architecture (MSA) instruction set architecture jointly developed by Analog Devices and Intel. There are two generations of Blackfin processors which have slightly different instruction sets and microarchitectures. Due to these architectural differences, the two generations are only partly assembly-code compatible.

As of late 2004, the ADSP-BF5xx family included five family members. The sole first-generation Blackfin processor, the ADSP-BF535, achieves a clock speed of 350 MHz at 1.6 volts. Second-generation Blackfin processors include the ADSP-BF531, ADSP-BF532, ADSP-BF533, and ADSP-BF561. The second-generation parts operate at up to 750 MHz at 1.45 volts.

ADSP-BF5xx family members are designed to operate over a range of clock speeds and operating voltages and are able to switch dynamically between speeds via software. For example, a single ADSP-BF533 chip can operate at speeds and voltages ranging from 750 MHz at 1.45 volts to 100 MHz at 0.8 volts.

For the purposes of this analysis, the Blackfin processor family is referred to as the ADSP-BF5xx throughout this text. However, because the ADSP-BF561 contains two processor cores, some of the following analysis does not apply to this dual-core family member.

As of late 2004, prices for ADSP-BF5xx family members range from $5 to $40 in 10,000-unit quantities. All five ADSP-BF5xx family members are currently in full production.

Architecture

The ADSP-BF5xx contains two fixed-point data paths, two address generation units, and a program sequencer. The ADSP-BF5xx also includes a data register file of eight 32-bit registers, as well as two 40-bit accumulators and two address register files. The ADSP-BF5xx uses a load/store architecture: the data paths generally take inputs from and return results to the data register file or the accumulators.

Collectively, the two ADSP-BF5xx data paths include two multipliers (MAC0 and MAC1), two ALUs (ALU0 and ALU1), and a single barrel shifter. One data path includes MAC0, ALU0, and the shifter; the other data path contains MAC1 and ALU1. The ADSP-BF5xx can issue one SIMD instruction that uses the two ALUs or the two MAC units in parallel, but it does not support instructions that use two dissimilar execution units; for example, one ALU and one MAC unit.

In addition to instructions that use both ALUs or both MAC units in parallel, the ADSP-BF5xx supports SIMD operations within each ALU and within the shifter (but not within the MAC units). These SIMD operations allow an execution unit to perform two operations per cycle. Thus, it is possible to perform

four operations per clock cycle using both ALUs or two operations per clock cycle using the shifter. However, the SIMD operations within the ALUs are supported only as part of a SIMD operation across the data paths, so that both ALUs must perform the same type of SIMD computation (e.g., a 16-bit add/subtract).

The ADSP-BF5xx generally processes data as 16-bit values. Many instructions support 32-bit data, however, and some support 40-bit data. For example, a five-cycle $32 \times 32 \rightarrow 32$ multiply is supported. The ADSP-BF5xx also supports a set of video-oriented operations that operate exclusively on 8-bit data.

Memory System

All ADSP-BF5xx memory is organized into a single unified 32-bit address space. However, the ADSP-BF5xx organizes its level-one (L1) on-chip memory into separate instruction and data banks.

The organization of the L1 memory is a primary differentiator among ADSP-BF5xx family members. Depending on the family member, the L1 memory is comprised of up to five separate banks of memory organized as a modified Harvard architecture. Up to three of the memory banks are data SRAM banks, some of which can be optionally configured as cache. The remaining two banks contain instruction ROM and instruction SRAM/cache.

In addition to level-one memory, the ADSP-BF535 includes an on-chip level-two (L2) memory system. The L2 memory of the ADSP-BF535 consists of 256 Kbytes of unified program and data RAM. The dual-core ADSP-BF561 provides four banks of private L1 memory for each core as well as 128 Kbytes of shared L2 memory.

The ADSP-BF5xx core can perform one instruction read and two data transfers in each cycle. Each data transfer can be 8, 16, or 32 bits wide. Both data transfers can access the same data bank if they use different sub-banks, but only one of the transfers can be a store. If data is arranged as 16-bit pairs in memory, the ADSP-BF5xx can transfer four 16-bit values each cycle. Thus, the maximum sustainable on-chip data bandwidth is 3 billion 16-bit words per second at 750 MHz for reads, or 1.5 billion 16-bit words per second for writes.

The ADSP-BF5xx address space is byte addressable. However, instructions must be aligned on 16-bit boundaries, and loads and stores must maintain the appropriate alignment for the transfer: 8-, 16-, and 32-bit transfers must be aligned on 8-, 16-, and 32-bit boundaries, respectively.

Addressing

The ADSP-BF5xx has two address generation units that can each generate an independent address in each cycle. The address generation units access two 32-bit register files that contain general-purpose address registers, stack pointers, and special registers for modulo addressing.

The ADSP-BF5xx supports a variety of addressing modes, including: register-indirect, register-indirect with post-increment or post-decrement, register-indirect indexed addressing with a short or long immediate offset, register-indirect addressing with pre-decrement for stack pushes, and register-indirect addressing with post-increment for stack pops. The register-indirect mode supports bit-reversed addressing.

The address generation units can also perform some addition, subtraction, and shifting operations on the address registers.

Pipeline

The first-generation ADSP-BF535 pipeline has eight stages, while the second-generation ADSP-BF5xx pipeline has ten stages. Both generations feature fully interlocked pipelines, so that data hazards do not cause unexpected results. However, some data hazards are resolved by stalling the processor. Data forwarding has been improved in the second-generation microarchitecture; some stalls that result from data hazards on the ADSP-BF535 have been removed by the improved forwarding mechanisms.

The ADSP-BF5xx uses static branch prediction for all conditional branches. The programmer (or compiler) must specify the prediction for each branch.

Instructions generally execute in one cycle in the absence of memory access-related delays and pipeline stalls. The most significant exceptions are the jump, call, and most return instructions, which require four or more cycles.

Instruction Set

The ADSP-BF5xx assembly language syntax is algebraic. The ADSP-BF5xx assembly language is not compatible with that of earlier Analog Devices processors.

The ADSP-BF5xx uses both 16- and 32-bit instructions. Arithmetic instructions are generally 32 bits wide, but some have 16-bit variants; 16-bit-wide arithmetic instructions offer less flexibility than their 32-bit counterparts. Load, store, and branch instructions are typically 16 bits wide, but most have 32-bit variants that support long immediate operands.

The ADSP-BF5xx instruction set is highly orthogonal. Most arithmetic instructions can take operands from the data registers, the pointer registers, the accumulators, or immediate operands. Most arithmetic instructions support both 16-bit and 32-bit data. Most arithmetic instructions also support SIMD operations.

Peripherals

ADSP-BF5xx family members offer a variety of peripherals, including DMA controllers, general-purpose I/O pins, timers with pulse width modulation (PWM) and pulse measurement capability, real-time clocks, watchdog timers, serial ports, UART ports, and Serial Peripheral Interface (SPI) ports. The ADSP-BF531, ADSP-BF532, ADSP-BF533, and ADSP-BF561 each have an internal voltage regulator and one or more parallel ports that support ITU-R 656 video modes. The ADSP-BF535 has PCI and USB interfaces.

Benchmark Performance

The BDTI Benchmarks™ are a set of digital signal processing functions that

BDTI has independently designed to provide an objective basis for comparing processor performance characteristics such as speed and memory use for signal processing applications. Implementations of the BDTI Benchmark functions are carefully optimized for each processor to allow a realistic assessment of signal processing performance. The resulting software is then verified for functional correctness, optimality, and adherence to the BDTI Benchmark specifications. Benchmark performance results are obtained either through manual analysis and careful, detailed simulation, or by measurement on sample devices.

BDTI's reports such as *Buyer's Guide to DSP Processors* and the *Inside* series of reports include extensive BDTI Benchmark results used to evaluate the signal processing performance of a set of processors. For each benchmark, BDTI typically reports cycle counts, execution times, a cost-performance metric, an energy-efficiency metric, and memory use.

In this section, we present sample execution time, energy consumption, and memory use results taken from BDTI's library of benchmark results for the ADSP-BF5xx and two other fixed-point DSPs: the Texas Instruments TMS320C55x and TMS320C64x.

Execution Time

Execution time results in this report were obtained assuming instructions and data are preloaded in caches where applicable. Processor speeds are for the fastest available chips as of late 2004.

Sample Execution Time Results

The execution time results for BDTI's LMS Adaptive FIR Filter benchmark are shown in Figure 1. The LMS Adaptive FIR Filter benchmark consists of an FIR filter, an error calculation, and a filter coefficient update. As shown in Figure 1, the ADSP-BF533 is much faster than the TMS320C5501 on this benchmark. Part of this difference in speed is due to a difference in cycle efficiency: the TMS320C5501 requires about 25% more cycles than the ADSP-BF533 to complete this benchmark. Although the ADSP-BF533 and the TMS320C5501 are both dual-MAC DSPs, only the ADSP-BF533 uses its dual-MAC capability on this benchmark. In the TMS320C5501, the two MAC units share an input—a restriction not imposed by most architectures. The TMS320C5501 is therefore limited to single-MAC operation on the filtering section of this benchmark. In contrast, the ADSP-BF533 uses both of its MAC units on this benchmark. As a result, the ADSP-BF53xx is significantly more efficient than the TMS320C5501 on this benchmark.

Figure 1. LMS Adaptive FIR Execution Times (lower is better)

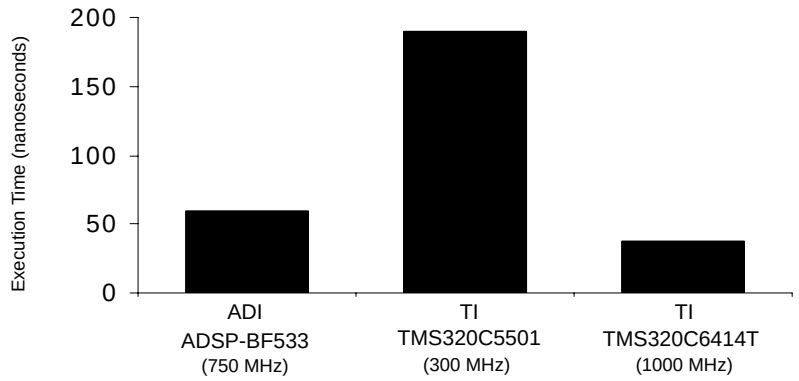
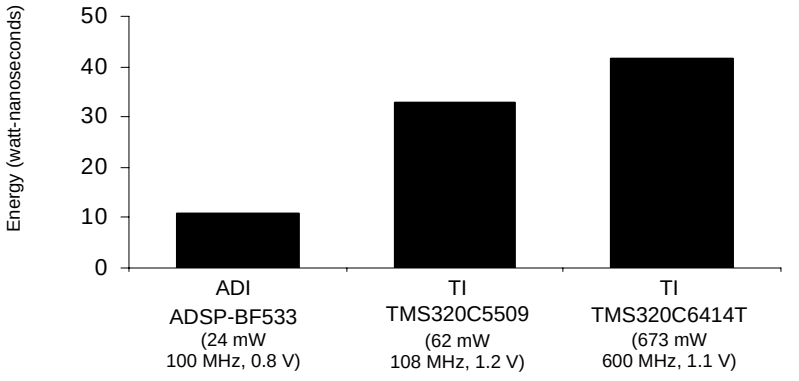


Figure 2. Energy Consumption for LMS Adaptive FIR (lower is better)



The TMS320C6414T is a high-performance 8-issue VLIW DSP, but it does not make full use of it parallelism on this benchmark. The LMS Adaptive FIR benchmark is a short benchmark in which setup and housekeeping tasks are an important factor in overall performance. These tasks present little opportunity for performing operations in parallel. Thus, the TMS320C6414T is only about 20% more cycle-efficient than the ADSP-BF533 on this benchmark.

Differences in clock rates also play a large role on this benchmark. The 750 MHz ADSP-BF533 executes instructions at a much faster rate than the 300 MHz TMS320C5501, but at a much slower rate than the 1000 MHz the TMS320C6414T. Due to its higher cycle efficiency and higher clock rate, the ADSP-BF533 is over three times faster than the TMS320C55x on this benchmark. Similarly, the combination of high cycle efficiency and high clock rate makes the TMS320C6414T roughly 1.5 times faster than the ADSP-BF533 on this benchmark.

Energy Efficiency

Figure 2 shows energy efficiency results. To estimate the energy required for a processor to complete a given benchmark, the benchmark execution time is multiplied by the estimated typical power consumption for that processor. For each professor family we select

the family member with the best energy efficiency; the chosen family member and speed grade may differ from the one used in the execution time comparison.

Based on processor power consumption and the LMS Adaptive FIR Filter benchmark execution times, the ADSP-BF533 is roughly three times more energy-efficient than the TMS320C5501 and nearly four times more efficient than the TMS320C6414T.

Cost-Performance

To create a cost-performance metric, the execution time is multiplied by the cost of the processor family member with the best speed-to-price ratio. This is not necessarily the same processor used in the execution time or energy efficiency metrics. The cost-performance results are shown in Figure 3.

Based on this metric, the ADSP-BF531 is the most cost-effective processor considered here, over 1.5 times more cost effective than the TMS320C5501 and nearly 3 times more cost effective than the TMS320C6410.

It should be noted that included on-chip memory and peripherals can be have a significant impact on overall system cost. These factors are not considered in the cost-performance metric used here.

Memory Use

Execution speed is often the primary metric used to compare processors. However, a processor's memory use is also important. For example, the memory requirements of an application can have a significant impact on overall system cost. In addition, processors may experience significant performance degradation when instructions and data do not fit in on-chip memory. Because of these and other factors, memory efficiency is an important metric in processor selection. For each of the BDTI Benchmarks™, BDTI measures each processor's program, constant data, non-constant data, and total memory use.

Control Benchmark

The BDTI Benchmarks™ include the Control benchmark, a benchmark specifically designed to evaluate memory use for control-oriented software.

Figure 3. Cost-Execution Time Product for LMS Adaptive FIR (lower is better)

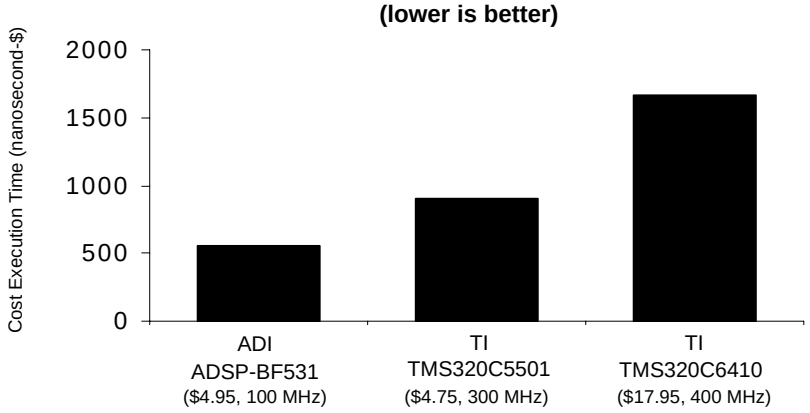
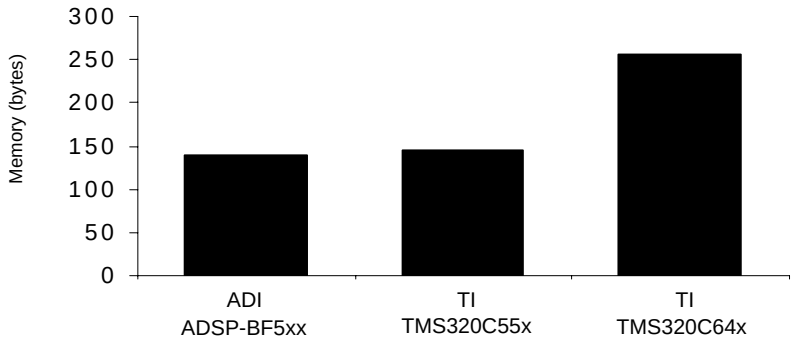


Figure 4. Control Benchmark Total Memory Use (lower is better)



Control-oriented tasks usually constitute the bulk of an application's program memory requirements, but only a fraction of the application processing time. Thus, in control-oriented tasks, minimizing memory use is usually a more serious concern than maximizing execution speed.

While most of the BDTI Benchmarks™ are optimized primarily for maximum speed, BDTI's Control benchmark is optimized for minimum memory use. This optimization hierarchy mirrors the approach generally followed by application programmers. Note that memory use results on the Control benchmark are not necessarily indicative of processor memory use in signal-processing-intensive code.

Sample Control Benchmark Results

The memory use results for BDTI's Control benchmark are shown in Figure 4. Large differences in Control benchmark memory use are typically due to differences in instruction widths. ADSP-BF5xx instruction widths range from 16 to 64 bits, but the Control benchmark implementation for the ADSP-BF5xx only uses 16-bit instructions. Similarly, TMS320C55x instruction widths range from 8 to 48 bits, but its Control benchmark implementation uses mostly 16-bit instructions. Thus, the ADSP-BF5xx and TMS320C55x have similar Control benchmark memory use. The TMS320C64x, on the other hand, uses 32-bit instruction words, increasing its memory use.

Conclusions

The ADSP-BF5xx competes with an unusually large number of processors, ranging from general-purpose embedded processors to high-performance DSPs. It is unusual to find a DSP processor family that offers variants ranging from a low-end 400 MHz part for less than $5 to a high-performance 750 MHz dual-core part.

Although the ADSP-BF5xx family spans a wide range of price and performance points, the family contains relatively few members: As of late 2004, only five family members were available. However, Analog Devices recently announced six new ADSP-BF5xx family members that are expected to be released in the near future.

Blackfin processors are not the fastest, nor the least expensive, nor the most energy efficient DSPs on the market. However, the ADSP-BF5xx does offer an excellent balance of these three factors.

The ADSP-BF5xx is particularly notable for its ability to dynamically switch between a wide range of voltages and clock speeds. This feature is no longer as rare as it once was, but it is still unusual—and the ADSP-BF5xx offers more flexibility than most other DSPs in this regard. For example, some of the latest TMS320C55x family members offer dynamic voltage and frequency adjustments, but these parts support only a narrow range of operating voltages.

The ADSP-BF5xx is quite straightforward to program in assembly language due to its orthogonal instruction set, algebraic assembly language, relatively benign pipeline, and flexible addressing modes. The orthogonal instruction set also makes the ADSP-BF5xx an excellent compiler target.

The ADSP-BF5xx has excellent assembly language tools. The tools make it very easy to detect memory conflicts and other stalls, which greatly aids code optimization. The ADSP-BF5xx also has good third-party software support. It should be noted, however, that software tools from Analog Devices are not as comprehensive and feature-rich as those from Texas Instruments.

The ADSP-BF5xx combines features typical of low-power DSPs with features traditionally associated with general-purpose processors. More and more DSPs are adding general-purpose features, but the ADSP-BF5xx includes a more complete set of features than some of its competitors. For example, the ADSP-BF5xx includes privileged execution modes that are useful for operating systems and multitasking software. There is also a version of the Linux operating system available for the ADSP-BF5xx.

Just as Analog Devices has incorporated many general-purpose features into the Blackfin processors, many of the latest general-purpose processors now include signal-processing-oriented features. The motives for combining DSP and general-purpose processor features is clear: Many signal processing applications currently employ both a DSP and a general-purpose processor. If these two processors could be replaced with a single processor, it could lead to significant cost savings. Hence, manufacturers are keenly interested in processors that combine the capabilities of a DSP and a general-purpose processor. Although DSPs such as the ADSP-BF5xx still enjoy some performance advantages over general-purpose processors, they cannot yet match the abundant application software, tools, and operating system support available for popular general-purpose processors. Thus, Analog Devices faces a tough battle as the line between DSPs and general-purpose processors continues to blur.

The ADSP-BF5xx faces a difficult market that is filled with strong competitors. However, Analog Devices is poised to strengthen the ADSP-BF5xx family with the introduction of several new family members in 2005. This processor family is a compelling choice overall for its excellent balance of price, features, and performance. ■