A BDTI Analysis of the

Texas Instruments TMS320C64x

by the staff of
Berkeley Design Technology, Inc.

Contents of this summary include:

Introduction

The TMS320C64x, announced in February 2000, is a family of 16-bit fixed-point VLIW DSPs from Texas Instruments. The TMS320C64x is the successor to the earlier TMS320C62x family, and its instruction set is a superset of that of the TMS320C62x. The TMS320C64x expands upon the capabilities of the TMS320C62x by adding significant SIMD (single-instruction, multiple-data) processing capabilities and a number of specialized telecom-oriented instructions, among other enhancements. The TMS320C64x can execute TMS320C62x object code unmodified, but the TMS320C62x cannot execute all TMS320C64x instructions.

The TMS320C64x family targets high-performance applications such as wireless base stations, DSL central office equipment, multi-line modems, ISDN modems, imaging equipment, video and audio applications, and radar and sonar systems.

As of late 2003, the fastest family members, the TMS320C6414, TMS320C6415, and TMS320C6416, execute at up to 720 MHz at 1.4 volts.

Three TMS320C64x family members—the TMS320DM640, TMS320DM641, and TMS320DM642—specifically target digital media applications. These chips provide on-chip video ports and audio serial ports. The TMS320DM642 began sampling in June 2003, and the TMS320DM640 and TMS320DM641 are expected to begin sampling in the first quarter of 2004.

As of late 2003, prices for TMS320C64x family members range from $35 to $132 in 10,000-unit quantities.

Architecture

Like the TMS320C62x, the TMS320C64x is a VLIW architecture that is based on two identical fixed-point data paths. Each data path contains four execution units: an ALU, a shifter, a multiplier, and an adder/subtractor used for address generation. Each data path also contains a register file with thirty-two 32-bit general-purpose registers, twice as many as on the TMS320C62x. The eight execution units are capable of executing up to eight 32-bit instructions in parallel using the two register files.

The TMS320C64x can operate on 8-, 16-, 32-, and 40-bit ("long") data (as can the TMS320C62x), and can also operate on 64-bit ("double word") data when loading or storing data from or to memory. The TMS320C64x handles 64-bit data by using a pair of registers.

Typically the ALUs, shifters, and address generation units operate on 32-bit operands, but the ALU and shifter units can also operate on 40-bit operands. The multipliers perform 16 × 16-bit or 8 × 8-bit multiplication.

Compared to the TMS320C62x, the TMS320C64x adds quad 8-bit and dual 16-bit SIMD arithmetic and logical instructions to most of the execution units, and introduces new quad 8-bit and dual 16-bit vector dot product instructions to the two multiply units. With this enhancement, the TMS320C64x is able to perform up to four 16-bit or eight 8-bit multiplications in parallel, for example.

Memory System

The TMS320C64x implements a modified Harvard memory architecture, providing separate address spaces for instruction and data memory. The TMS320C64x fetches instructions using a 32-bit address bus and 256-bit data bus. Each data path accesses data memory using a 32-bit address bus and a 64-bit data bus. The TMS320C64x supports 16-bit, 32-bit, and 64-bit data transfers. Up to two 64-bit data move instructions can be executed in parallel with other instructions. Thus, up to four 32-bit registers can be loaded or stored in every cycle.

The TMS320C64x uses a two-level cache-based memory organization similar to that of the fixed-point TMS320C6211.

The TMS320C64x level-one (L1) memory is organized as separate instruction and data caches. L1 memory for all current TMS320C64x family members is comprised of a 512×256 program cache and 4K×32 data cache.

The processor's on-chip level-two (L2) memory is unified; i.e., it is used for both instructions and data. The L2 memory of TMS320C64x devices serves as both a second-level cache and as regular

RAM: the size of the L2 cache is configurable, and any remaining portion of L2 memory serves as direct-mapped RAM. (The two L1 caches cannot be configured as regular RAM.) L2 memory ranges in size from 2K×256 to 16K×256 depending on the family member.

Addressing

Like the TMS320C62x, the TMS320C64x supports register-direct and register-indirect addressing modes and immediate data. In register-indirect addressing mode, the address register modification options include pre-increment/decrement by a 5-bit immediate value or by the contents of any general-purpose register, and post-increment/decrement by a 5-bit immediate value or by the contents of any general-purpose register.

Compared to the TMS320C62x, the TMS320C64x adds support for 64-bit data addressing and non-aligned loads and stores.

The TMS320C64x also supports modulo addressing. Up to eight registers (four from each register file) can be configured to operate under modulo addressing. The TMS320C64x does not support bit-reversed addressing, but does have a bit-reversal instruction.

Pipeline

The TMS320C64x uses an 11-stage pipeline. The pipeline is non-interlocked, meaning that the processor does not resolve pipeline-related resource conflicts or data dependencies. Consequently, hand-written assembly code must be carefully crafted to avoid unwanted behavior.

Many arithmetic and logical instructions on the TMS320C64x have a latency of one cycle, as on the TMS320C62x. The new application-specific instructions and SIMD instructions have latencies ranging from one to five cycles. Multiplies have a two-cycle latency; loads have a five-cycle latency. Stores have a single-cycle latency. All branches on the TMS320C64x are delayed branches with five delay slots.

Instructions are always fetched eight at a time via the 256-bit instruction bus. This group of eight instructions is called a "fetch packet." However, the TMS320C64x cannot always execute eight instructions in parallel. The group of instructions to be executed in parallel is called an "execution packet." Because the TMS320C64x supports variable-length execution packets (and thus can execute from one to eight instructions in parallel), a single fetch packet may contain several execution packets.

On the TMS320C62x, a single execution packet cannot span two fetch packets. Consequently, some TMS320C62x fetch packets must be padded with NOPs. The TMS320C64x improves code density by allowing execution packets to span fetch packets.

Instruction Set

The TMS320C64x uses an opcode-operand assembly language format where each instruction has an opcode field for the operation and an operand field for one to four operands.

All instructions on the TMS320C64x can be executed conditionally. Six designated general-purpose registers can be used as condition registers.

Many TMS320C64x instructions are simple and RISC-like, as on the TMS320C62x. However, the addition of extensive SIMD capabilities and application-specific instructions means that some TMS320C64x instructions support multiple parallel operations within a single instruction.

The TMS320C64x does not support hardware looping, so all loops must be implemented in software. However, the parallel architecture of the processor allows the implementation of software loops with virtually no overhead.

Because the TMS320C64x is a highly parallel architecture, obtaining maximum performance requires the programmer or code-generation tools to schedule instructions carefully. This can be a challenge because the TMS320C64x has a complex architecture and long, variable instruction latencies. Texas Instruments' assembly optimizer tools and C compiler simplify code development by automating the scheduling and parallelization processes, but these tools do not always yield optimal code.

Peripherals

TMS320C64x family members include a variety of on-chip peripherals, including a host port, a multi-channel DMA controller, multi-channel buffered serial ports, and 32-bit timers. The TMS320C6416, which targets high-performance communications applications, also includes on-chip coprocessors that perform Viterbi and turbo decoding operations. The TMS320DM64x family members, which target audio and video-oriented applications, include an audio serial port and one or two video ports.

Benchmark Performance

The BDTI Benchmarks™ are a set of signal processing software 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. The BDTI Benchmark functions are implemented in optimized assembly language to allow a realistic assessment of processors' 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 time, cost-performance, energy efficiency, and memory usage.

In this section, we present sample execution time, cost-performance, energy consumption, and memory usage results taken from BDTI's library of benchmark results for the TMS320C64x and two other processors: the Analog Devices ADSP-TS20x "TigerSHARC" and the Motorola MSC810x (which is

based on the StarCore SC140 core). All three processors are benchmarked using 16-bit fixed-point arithmetic, although the ADSP-TS20x also provides native support for floating-point arithmetic.

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 October 2003.

Sample Execution Time Results

Figure 1 shows execution-time results on BDTI's Real Block FIR Filter benchmark for the fastest member of each processor family. The 720 MHz TMS320C6414 has roughly the same speed on this benchmark as the 600 MHz ADSP-TS201S, and is about twice as fast as the 300 MHz MSC8103.

The TMS320C6414 achieves this speed through a combination of a high clock rate and a high degree of parallelism. For example, the MSC8103 and TMS320C6414 require a similar number of cycles to execute this benchmark, but the TMS320C6414 clock speed is more than twice as fast—thus yielding an execution time that is much faster than that of the MSC8103.

The Real Block FIR Filter benchmark places a strong emphasis on multiply-accumulate (MAC) throughput. As shown by the results here, however, other factors can influence the relative results. For example, although the ADSP-TS201S can execute up to eight 16-bit fixed-point multiplies per cycle—twice as many as the TMS320C6414 and MSC8103—the ADSP-TS201S must spend a number of cycles combining results of the multiplies. Thus, its execution time on this benchmark is not as fast as one might expect given the three processors' relative MAC throughputs and clock speeds.

Figure 1. Real Block FIR Filter Execution Times (lower is better)

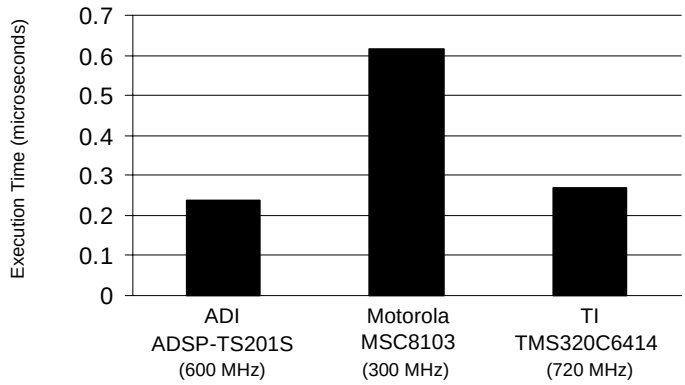


Figure 2. Cost-Execution Time Product for Real Block FIR Filter (lower is better)

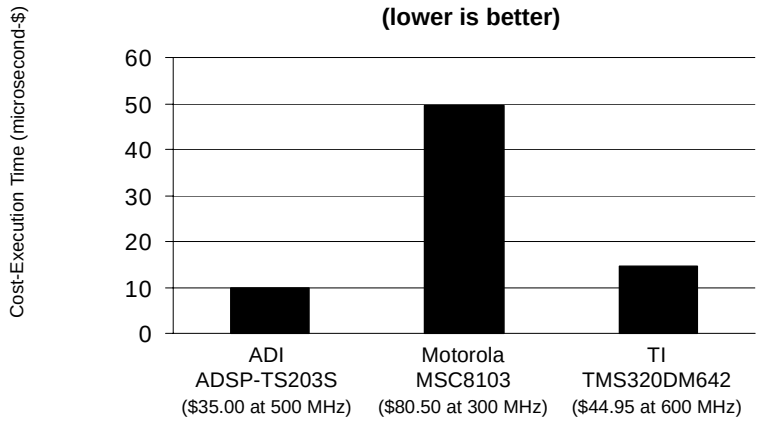
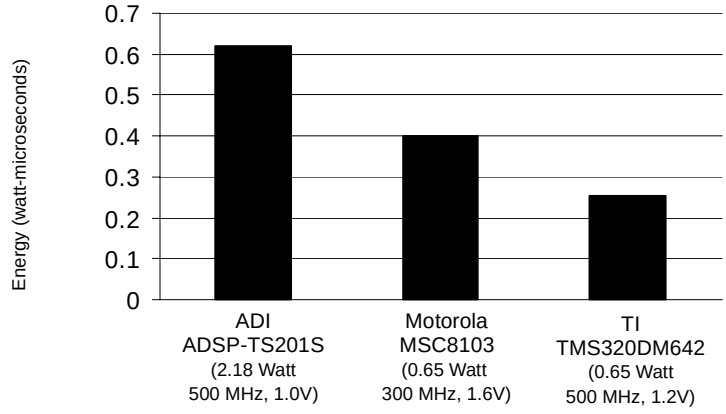


Figure 3. Energy Consumption for Real Block FIR Filter (lower is better)



Cost-Performance

Figure 2 shows cost-performance results for the most cost-effective member of each processor family. To create the cost-performance metric, the Real Block FIR Filter execution time is multiplied by the cost of the processor in 10,000-unit quantities as of October 2003. Based on this analysis, the 500 MHz ADSP-TS203S, at $35.00, is the most cost-effective processor considered here. It is about a third more cost-effective than the 600 MHz TMS320DM642, and five times as cost-effective as the 300 MHz MSC8103.

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

Energy Efficiency

Figure 3 shows energy consumption results for the most energy-efficient member of each processor family. To estimate the energy consumption metric,

the Real Block FIR Filter execution time is multiplied by the typical power consumption of the processor. Figure 3 lists the typical power consumption for each processor below the processor name.

The 500 MHz TMS320DM642 has the best result of the three processors on this metric. It is about 35% faster than the 300 MHz Motorola MSC8103, and it consumes roughly the same amount of power. Consequently, the TMS320DM642 consumes about a third less energy than the MSC8103. The TMS320DM642 is not as fast as the 500 MHz ADSP-TS201S on this benchmark, but it consumes only about a third of the power. Thus, the TMS320DM642 energy consumption is about 2.5 times lower than that of the ADSP-TS201S.

Memory Use

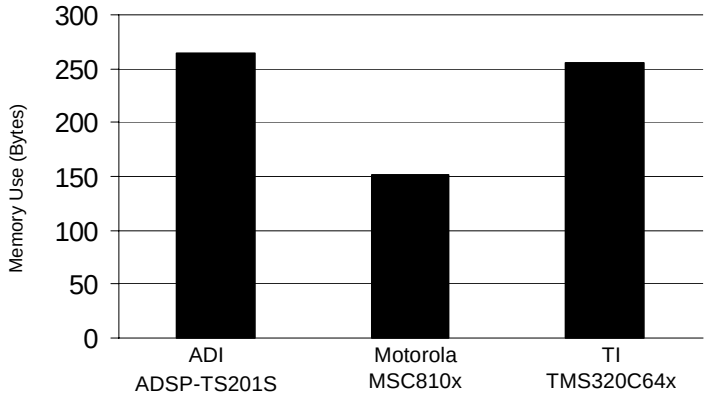
Execution speed is often the primary metric used to compare processors. However, a processor's memory usage is also important. For example, the memory requirements of an application can have a significant impact on overall system cost. In addition, processors may experience significant performance degradation 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 one benchmark function specifically designed to evaluate memory use for control-oriented software. 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.

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

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



Sample Control Benchmark Results

Figure 4 shows memory usage results for BDTI's Control benchmark. The differences in Control benchmark memory usage shown here are primarily due to differences in instruction widths. The MSC810x (SC140) achieves its low total memory usage result because of its support for a mixed-width 16/32-bit instruction set. On this benchmark, the MSC810x is able to use mostly 16-bit instructions. Both the ADSP-TS201S and TMS320C64x support only 32-bit instructions. As a result, the Control benchmark memory usage results for these two processors are similar, and are significantly higher than that of the MSC810x.

Conclusion

The TMS320C64x offers extremely fast fixed-point signal-processing speed. It is much faster than the Motorola MSC810x, and offers speed similar to that of the Analog Devices ADSP-TS201S. The TMS320C64x also has better energy efficiency results than either of these competitors. However, the most cost-effective TMS320C64x family member is somewhat less cost-effective than the corresponding ADSP-TS20x family member. The TMS320C64x memory usage result (like that of the ADSP-TS201S) is fairly high.

The extensive support for eight-bit SIMD operations on the TMS320C64x makes the processor very well suited for image processing and video compression/decompression algorithms, such as MPEG-2 video decoding. The processor's new application-specific instructions also improve its performance in some traditional audio- and telecom-oriented algorithms.

The TMS320C64x benefits from TI's mature and well-supported development infrastructure. However, as noted earlier, the TMS320C64x family is relatively difficult to program at the assembly language level. The TMS320C64x also enjoys extensive third-party support including development boards, emulators, and software libraries from a variety of vendors. ■

About BDTI

Berkeley Design Technology, Inc. (BDTI) was founded in 1991 to assist companies in creating, selecting, and using DSP technology. BDTI offers a variety of technical products and services, including:

- *Published reports on DSP processors and technology*
- *DSP software development services*
- *Technical advisory services*
- *Training*

■