A BDTI Analysis of the

Texas Instruments TMS320C67x

by the staff of
Berkeley Design Technology, Inc.

Contents of this summary include:

Introduction

The TMS320C67x is a family of 32-bit floating-point DSPs from Texas Instruments. The TMS320C67x family is derived from the TMS320C62x family of 16-bit fixed-point DSP processors, and it can execute all TMS320C62x instructions. In addition, the TMS320C67x adds supports for IEEE-754 32-bit single-precision and 64-bit double-precision floating-point arithmetic and expands support for 32-bit fixed-point arithmetic. The TMS320C67x family targets applications such as home audio, 3D graphics, medical imaging, radar, and speech recognition.

The first TMS320C67x family members were announced in April of 1998. As of mid-2003, the TMS320C67x family includes the TMS320C6701, TMS320C6711, TMS320C6712, and TMS320C6713. The fastest family member, the 225 MHz TMS320C6713, is fabricated in a 0.13-micron process and uses a 1.26-volt core supply. Other TMS320C67x family members operate at speeds ranging from 100 to 200 MHz. As of mid-2003, prices for TMS320C67x parts range from under $15 to over $100 in 10,000-unit quantities, depending on the family member.

The TMS320C67x is backward-compatible with the TMS320C62x: the TMS320C67x can execute TMS320C62x object code unmodified, but the TMS320C62x cannot execute all TMS320C67x instructions. In contrast, the TMS320C67x is not object-code compatible with Texas Instruments' latest generation of the fixed-point TMS320C6000 architecture, the TMS320C64x. The TMS320C64x extends the TMS320C62x instruction set with instructions that are not supported by the TMS320C67x.

Architecture

The TMS320C67x contains two floating-point data paths. Each data path contains two ALUs, a multiplier, and an adder/subtractor for address generation. The ALUs support both integer and floating-point operations, and the multipliers can perform 16 × 16-bit and 32 × 32-bit integer multiplies as well as 32-bit and 64-bit floating-point multiplies.

Each data path contains a register file of sixteen 32-bit general-purpose registers. These registers can be used for storing addresses or data. To support 64-bit floating-point arithmetic, pairs of adjacent registers can be used to hold 64-bit data.

Memory System

The memory system of the TMS320C67x implements a modified Harvard architecture, providing separate address spaces for instruction and data memory. The TMS320C67x fetches instructions using a 32-bit address bus and 256-bit data bus. Each data path accesses data using a 32-bit address bus and a 64-bit data bus. Together, these data buses can perform two 64-bit loads, two 32-bit stores, or a 64-bit load and a 32-bit store to or from on-chip memory per instruction cycle.

The TMS320C67x allows up to two data move instructions to be executed in parallel with other instructions.

The TMS320C6701 contains 64 Kbytes of program RAM and 64 Kbytes of data RAM. The program memory can be configured to act as an instruction cache.

The TMS320C6711 and TMS320C6712 use the same two-level on-chip cache memory architecture as the fixed-point TMS320C6211. The TMS320C6711 and TMS320C6712 each contain two 4 Kbyte level-one caches, one for data and one for instruc-

tions. The level-one caches are fed by a unified 64 Kbyte level-two memory. The level-two memory can be configured as SRAM, as a cache, or as a partitioned combination of the two.

The TMS320C6713 on-chip memory configuration is similar to that of the TMS320C6711 and TMS320C6712, except that it contains a larger 256 Kbyte level-two memory.

Addressing

The TMS320C67x 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 short (5-bit) immediate or by the contents of any general-purpose register, and post-increment/decrement by a short immediate or by the contents of any general-purpose register.

The TMS320C67x supports modulo addressing. Up to eight registers (four from each register file) can be configured to operate under modulo addressing. The TMS320C67x does not support bit-reversed addressing.

Pipeline

The TMS320C67x pipeline consists of 16 stages. The pipeline is non-interlocked and is significantly deeper than those of other commercially available DSP processors.

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 TMS320C67x cannot always execute eight instructions in parallel. The group of instructions to be executed in parallel is called an "execution packet." Because the TMS320C67x 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.

The processor does not check execution packets for resource contention. Consequently, hand-written assembly code may introduce resource conflicts that produce unwanted behavior.

All branches on the TMS320C67x are delayed branches with five delay slots.

Most fixed-point instructions on the TMS320C67x have a latency of one cycle. The branch, fixed-point multiply, and load instructions produce results only after several cycles. Latencies for floating point operations range from one to ten cycles.

Instruction Set

The TMS320C62x 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. In addition, three optional fields can be used to indicate parallel execution, conditional execution, and the targeted execution unit. If the target execution unit field is omitted from the instruction, the assembler attempts to select an appropriate execution unit.

All instructions on the TMS320C67x can be executed conditionally. Five designated general-purpose registers can be used as condition registers.

The TMS320C67x 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.

Due to its simple, RISC-like instructions, 32-bit instruction width, and uniform register sets, the instruction set of the TMS320C67x is extremely regular and straightforward.

Because the TMS320C67x is a highly parallel architecture, obtaining maximum performance often requires the programmer to schedule instructions carefully. This can be a challenge because the TMS320C67x 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 result in optimal code.

Peripherals

TMS320C67x family members include a variety of on-chip peripherals, including a host port, a multi-channel DMA controller, multi-channel buffered serial ports, digital audio interfaces, and 32-bit timers.

Benchmark Performance

The BDTI Benchmarks™ are a set of DSP software functions that BDTI has independently designed to provide an objective basis for comparing processor performance characteristics such as speed and memory use for DSP 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 DSP performance of a set of processors. For each benchmark, BDTI typically reports cycle counts, execution time, a cost-performance metric, an energy-efficiency metric, 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 TMS320C67x and two other floating-point processors: the Analog Devices ADSP-2116x and the Renesas (formerly Hitachi) SH775x, which is based on the SH-4 core.

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 mid-2003.

Sample Execution Time Results

Figure 1 shows execution time results on BDTI's 256-point FFT benchmark for the fastest member of each processor family. The 225 MHz TMS320C6713 is over 30% faster on this benchmark than either the 100 MHz ADSP-21161N or the 240 MHz SH7750R. The TMS320C6713 achieves this speed through a combination of a high clock rate and a high degree of parallelism.

The TMS320C6713 could be even faster if it supported floating-point SIMD instructions like those supported by the ADSP-21161N and the SH7750R. For example, the ADSP-21161N can compute both the sum and the difference of two operands with a single instruction. The TMS320C6713 is also held back by the long latencies of its floating-point arithmetic instructions (for example, four cycles for single-precision additions and multiplications).

As mentioned above, the ADSP-21161N has instructions that accelerate the FFT butterfly. However, the ADSP-21161N has a relatively low clock speed, so it cannot keep up with the TMS320C6713.

The SH7750R is also quite fast at processing the FFT butterfly. However, the SH7750R spends many cycles loading coefficients between FFT butterflies. Consequently, the SH7750R is much slower than the TMS320C6713 on this benchmark.

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 FFT execution time is multiplied by the cost of the processor in 10,000-unit quantities as of mid-2003. Based on this analysis, the 150 MHz TMS320C6712C is the most cost-effective processor considered here. It is over 35% more cost-effective than either the 100 MHz ADSP-21161N or the 200 MHz SH7750R.

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

Figure 1. 256-Point FFT Benchmark Execution Times (lower is better)

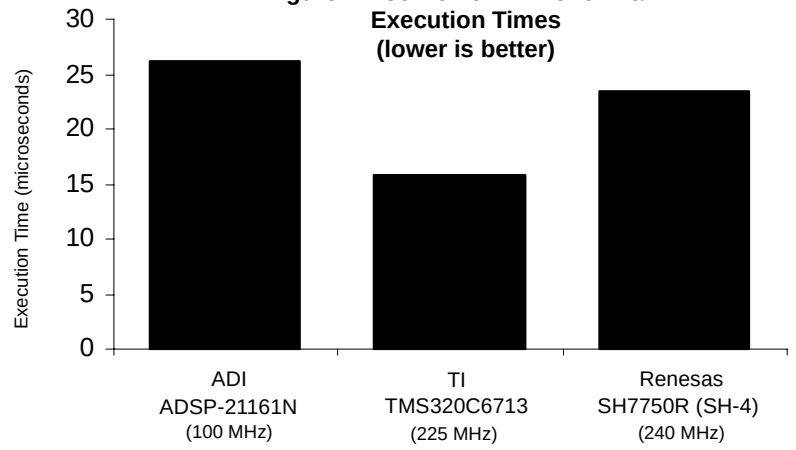


Figure 2. Cost-Execution Time Product for 256-Point FFT (lower is better)

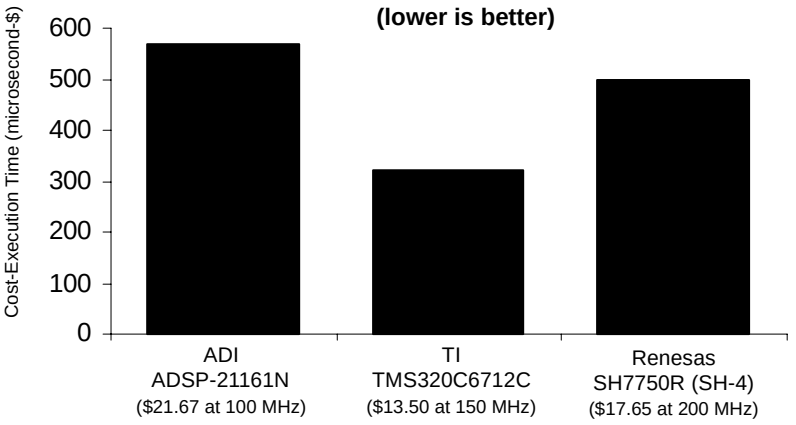
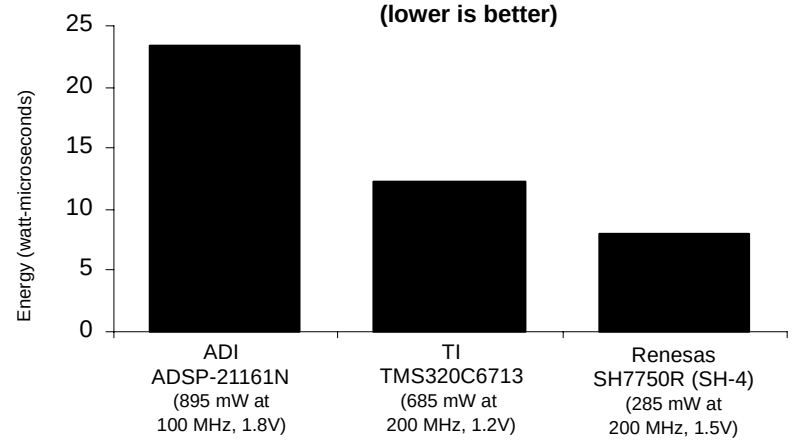


Figure 3. Energy Consumption for 256-Point FFT (lower is better)



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 FFT 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 200 MHz TMS320C6713 is about 30% faster than the 100 MHz ADSP-21161N, and it consumes about 25% less power than the ADSP-21161N. Consequently, it consumes about half as much energy as the ADSP-21161N. The 200 MHz TMS320C6713 is also about

30% faster than the 200 MHz SH7750R, but it consumes over two times more power than the SH7750R. Hence, the TMS320C6713 consumes about 50% more energy than the SH7750R.

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

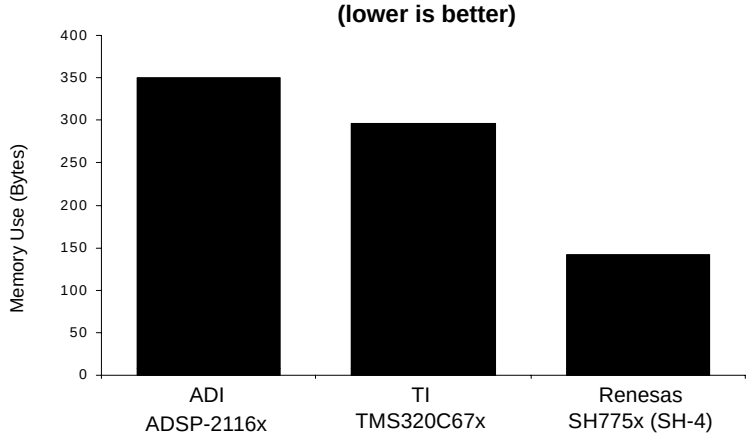
The BDTI Benchmarks™ include one benchmark function specifically designed to evaluate memory use for control-oriented software. Control-oriented tasks usually constitute the bulk of an application's program memory requirements, but only a fraction of the application processing time. Thus, in control-oriented 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.

Sample Control Benchmark Results

Figure 4 shows memory usage results for BDTI's Control benchmark. The large differences in Control benchmark memory usage are primarily due to the differences in instruction widths. The SH775x achieves its low total memory usage result primarily due to its 16-bit instructions, while the TMS320C67x uses 32-bit instructions and the ADSP-2116x uses 48-bit instructions. Furthermore, the TMS320C67x must occasionally use multi-cycle NOP instructions to fill branch delay slots. The TMS320C67x has a total memory usage on the Control benchmark that is more than twice that of the SH775x. Due to its 48-bit instructions, the ADSP-2116x has a total Control benchmark memory usage that is roughly 20% higher than that of the TMS320C67x.

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



Conclusion

The TMS320C67x achieves commendable signal-processing speed, particularly considering its low cost. It is much faster than two of its main competitors, the Analog Devices ADSP-2116x and the Renesas SH775x. It also has better cost-performance results than either of these competitors. However, it is not nearly as fast as high-end floating-point processors such as the Analog Devices ADSP-TS20x and the Motorola MPC74xx. In terms of energy and memory use, the TMS320C67x is more efficient than the ADSP-2116x but less efficient than the SH775x.

The TMS320C67x is the first mainstream DSP to support IEEE-754 double-precision floating-point arithmetic, which is an advantage for applications that require double-precision arithmetic. However, because double-precision operations have much higher latency and lower throughput than single-precision operations, an algorithm that uses double-precision arithmetic will be several times slower than the same algorithm using only single-precision arithmetic.

The TMS320C67x is one of the very few DSP processor families to offer instruction set compatibility between fixed- and floating-point processors. This can be an advantage for algorithms that are initially designed for floating-point arithmetic but are later converted to a fixed-point implementation for volume production; non-numeric code from the floating-point implementation does not need to be re-written for the fixed-point version. However, due to the different latencies of fixed- and floating-point instructions and other restrictions, algorithm kernels will almost always have to be completely re-written when migrating from the TMS320C67x to the TMS320C62x.

The TMS320C67x benefits from TI's mature and well-supported development infrastructure. However, as noted earlier, the TMS320C67x family is relatively difficult to program at the assembly language level.

The TMS320C67x also benefits from extensive third-party support including development boards, emulators, application boards, development tools, and software libraries from a variety of vendors. ■