

Microprocessors Outperform DSPs 2:1

Unpredictable Execution, Poor Tools Complicate Use in Real-Time Applications

by Garrick Blalock, Berkeley Design Technology

An independent benchmarking study recently completed by Berkeley Design Technology (BDT) reveals that some general-purpose 32-bit microprocessors outperform dedicated digital signal processors on DSP tasks by factors of more than 2:1. The results of the study caution, however, that many general-purpose microprocessors suffer from problems that can complicate their use in real-time DSP applications.

Digital signal processing is becoming ubiquitous in both desktop PCs and embedded applications. Many engineers are deciding how best to implement signal-processing functions in their systems, and the choices often include using a microprocessor or microcontroller already present in the design. The study's benchmark results show that this may often be a viable and attractive approach. Rapid increases in performance enable general-purpose microprocessors to handle tasks that only DSPs could do a few years ago. Furthermore, recognizing the importance of DSP, many general-purpose CPU vendors have begun adding DSP features that further boost signal-processing performance.

Economics Favor Microprocessors

Using a system microcontroller or microprocessor for signal processing, as opposed to using a dedicated DSP chip, has the obvious benefits that often come with increased integration: reduced cost and parts count, lower power consumption, smaller PCB size, and fewer software platforms. With desktop PCs, which already contain a high-performance CPU, implementing signal processing on the existing processor allows the designer to add DSP applications like modems or sound cards with little or no additional hardware cost.

General-purpose microprocessors, however, typically have several shortcomings that complicate implementation of DSP tasks. In particular, general-purpose processors often have poor execution-time predictability, a problem in applications with real-time constraints. Software development is also a concern. High-end general-purpose processors feature

Processor	Application	Fixed Point	Floating Point
PowerPC 604e	Desktop PCs	Excellent	Excellent
Pentium	Desktop PCs	Poor	Excellent
P55C	Desktop PCs	Excellent	Excellent
ARM7 TDMI	Embedded	Poor	n/a
SH-DSP	Embedded	Excellent	n/a
R4650	Embedded	Good	Poor

Table 1. The processors covered in Berkeley Design Technology's study and their suitability for fixed-point and floating-point DSP.

complicated architectures that make it very challenging to optimize DSP inner loops in assembly language—which is often necessary for top performance. The problem is intensified by general-purpose processors' overall lack of strong DSP-oriented development tools and applications-engineering support. Furthermore, in desktop PC applications, the operating system's lack of support for real-time tasks presents another hurdle.

BDT's study examined the signal-processing features of several general-purpose microprocessors and benchmarked their performance using the BDT Benchmarks, a suite of synthetic DSP benchmarks. The six processors examined in the study are summarized in Table 1. Figure 1 shows the execution time on BDT's FFT benchmark of these general-purpose processors and some popular DSPs.

PowerPC 604e Takes the Lead in Floating Point

The IBM/Motorola PowerPC 604e demonstrates exceptional floating-point DSP performance. For example, the 604e at 200 MHz outperforms every floating-point DSP in execution speed on the BDT FFT benchmark. This strong perfor-

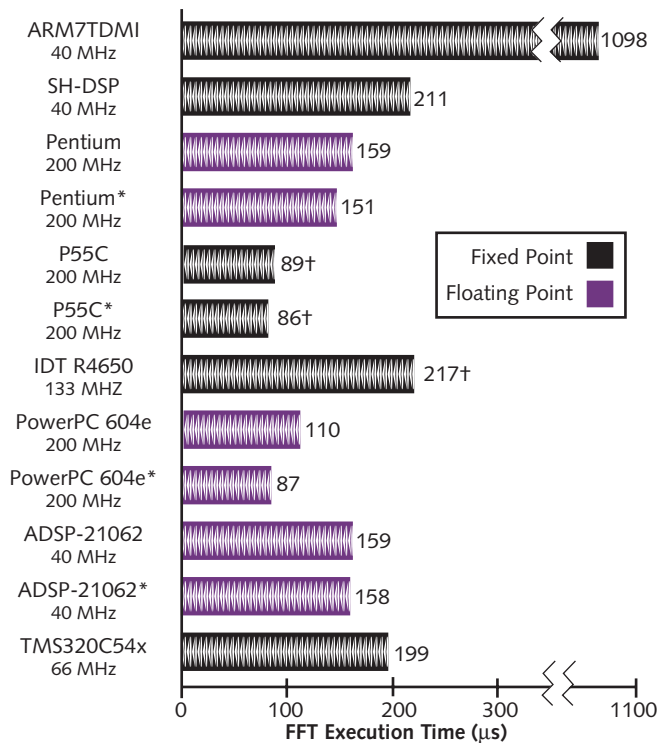


Figure 1. Execution time on BDT's 256-point complex, radix-2 FFT benchmark. *code, data preloaded in cache. †performance estimated. (Source: Berkeley Design Technology)

mance comes despite the 604e's lack of significant DSP-specific features. For example, although the 604e's floating-point unit can theoretically support multiply-accumulate operations with single-cycle throughput, it is impossible to sustain this performance for the majority of DSP applications because of limitations in memory bandwidth. Unlike most DSPs, which have two independent on-chip data buses to load operands, the 604e has only one on-chip data bus, limiting it to one operand load per clock cycle and forcing the FPU to wait for operands in typical DSP algorithms. Even with frequent stalls, however, the 604e still exhibits excellent performance because of its fast 200-MHz clock speed, four-way superscalar architecture, and branch prediction.

Although the PowerPC 604e displays excellent DSP performance potential, obtaining this performance can be difficult. Several factors complicate writing optimal DSP code and predicting its execution time. The 604e's superscalar architecture, for example, creates difficulties in determining the optimal instruction ordering and the time required to execute inner loops. Furthermore, the 604e employs branch prediction, which creates variations in execution times for some applications. These factors hinder the programmer's ability to ensure optimal, dependable real-time execution of code—a key concern in almost any DSP application.

Unpredictable Performance Could Be Trouble

More troubling, data-dependent instruction timing can create situations where the worst-case execution time can be an order of magnitude longer than in the typical case. For example, when storing a single-precision IEEE floating-point number to memory, a penalty of up to 23 clock cycles may be incurred if the number is nonzero but small enough to need denormalization to fit into a single-precision representation. Although such denormalization occurs infrequently in most applications, this penalty can make the worst-case execution time of an algorithm much longer than its average execution time. However unlikely, the possibility of exceeding critical real-time constraints is a potentially fatal flaw for many DSP applications. Programmers must include appropriate safeguards in their software. For example, turning off IEEE floating-point compliance forces the 604e to simply truncate extremely small numbers to zero and bypass the time-consuming denormalization process.

The 604e's excellent benchmark results are achieved in its floating-point data path, which is better suited to some DSP applications than the integer data path. Unfortunately, in many DSP systems, particularly those that process real-world data via A/D and D/A converters, data is likely to enter and exit in a fixed-point format and require conversion to or from floating point. The 604e performs conversion in either three or seven instructions, depending on the direction of the conversion. Both conversions include memory accesses that may cause pipeline stalls, however, even if no cache misses occur. The potential for pipeline stalls adds still more

variability to program execution time and complicates the programming of real-time applications.

Development tools for the 604e also fall short when developing DSP applications that run under real-time constraints. Although the PowerPC has more development tools available than do many other processors, the lack of a generally available cycle-accurate simulator or other tool to enable detailed observation of the execution of instructions often leaves DSP programmers unable to explain longer-than-expected execution times in DSP inner loops. This uncertainty can make optimizing DSP assembly code for the 604e quite difficult. In contrast, cycle-accurate simulators are generally available for nearly all DSPs.

Pentium Pulls Alongside Floating-Point DSPs

Pentium also demonstrates strong floating-point performance in some DSP applications. Like the 604e, Pentium lacks many DSP-specific features but compensates with high clock speeds and a full-featured floating-point unit. Its complexity, however, makes it difficult to predict execution time with clock-cycle accuracy. Furthermore, as with the 604e, Pentium achieves peak DSP performance in its floating-point data path, so designers must consider the time required for conversions between floating-point and integer formats.

Although Pentium, like the 604e, suffers from difficulties in predicting code-execution time and optimizing DSP code, Intel has taken some steps to address the needs of real-time application development. The company offers Vtune, a tool suite for profiling and optimizing 32-bit Pentium code. Vtune first collects a trace of a program's execution by running the program on an actual processor. It then uses an approximate, timing-only model of the processor to predict the performance of the traced program, to identify places where performance penalties are incurred, and to suggest possible optimizations. In addition, Intel offers several libraries of NSP (native signal processing) routines, optimized DSP functions that programmers may call from within their C programs.

MMX Helps P55C in Fixed-Point Calculations

Intel's formal announcement of the Pentium processor with MMX Technology (P55C) will come in early January. MMX (see [100301.PDF](#)) adds a number of DSP-oriented features to Pentium. Several MMX instructions are SIMD (single-instruction, multiple-data). For example, there is an instruction that performs four 16-bit integer multiplies and then adds the 32-bit products in pairs. This instruction can be used for complex multiplication or vector products.

The eight 64-bit MMX registers are aliased onto the existing eight 80-bit floating-point registers. Switching back and forth between MMX mode and floating-point mode is slow, requiring up to 50 clock cycles. The penalty for this switch, however, is unlikely to significantly affect many DSP applications. Because the MMX data path is fixed-point, DSP programmers working on the P55C will typically use fixed-

point arithmetic instead of floating-point arithmetic. Thus, the slow switch between floating-point and MMX will occur infrequently in most DSP applications.

Projected BDT Benchmark results for the P55C suggest significant improvements over the performance of a Pentium without MMX. For example, the P55C executes the BDT FFT benchmark using 16-bit fixed-point arithmetic in about half the time (estimated) of a standard Pentium using floating-point arithmetic at the same clock speed. Of course, since the MMX version is calculating the FFT in 16-bit fixed-point arithmetic while the standard Pentium is using full floating-point precision, one must be careful in making comparisons.

SH-DSP Excels at Only 40 MHz

Hitachi's SH-DSP (see [091603.PDF](#)) is one of three general-purpose processors in the study that are designed for embedded applications. The SH-DSP adds a complete fixed-point DSP data path and instruction set to Hitachi's successful SH-2 microcontroller architecture. This hybrid approach allows programmers to add DSP functions while protecting their investment in SH-2 code, which will run unaltered on the SH-DSP. Programmers can access the chip's DSP data path by adding DSP instructions to an SH-2 program. The SH-DSP issues the DSP and CPU instructions to the appropriate execution unit.

The DSP capabilities of the SH-DSP are very similar to those of many 16-bit DSPs and enable strong fixed-point DSP performance at clock speeds much lower than those of Pentium and the 604e. The SH-DSP's compatibility with the SH-2 provides a natural migration path for SH-2 customers contemplating DSP-intensive designs. Of course, there is a price for this compatibility. Although the SH-DSP is a single processor, it has two personalities: two instruction sets, two data paths, two sets of registers, and so on. This duality complicates the programming model and hinders performance in some instances.

In contrast to several high-performance microprocessors, the SH-DSP executes programs in a very predictable manner. This predictability is attributable to the processor's use of on-chip SRAM instead of caches, its single-issue architecture, and its lack of branch prediction and data-dependent instruction execution times. These factors are advantages for programmers seeking to optimize DSP code and ensure real-time performance. In this sense, the SH-DSP is very similar to typical DSPs.

The SH-DSP is also remarkable for turning in competitive times at a clock speed of just 40 MHz, compared with the 200-MHz clock rates of the 604e and Pentium.

ARM7TDMI Sacrifices Performance for Size

The ARM7TDMI core is the simplest general-purpose processor covered in the study. Its only DSP-oriented feature is an integer multiply-accumulate unit. The ARM7 uses a single external data bus, limiting its memory bandwidth but

reducing die size and production cost. Thus, ARM7 single-word load instructions require a minimum of three cycles. The ARM7 has a multiple-word load instruction that allows additional data to be loaded at one word per cycle. The multiple-word load instruction mitigates some of the ARM7's memory-bandwidth limitation.

Because the ARM multiplier employs an early-termination algorithm that processes multiplicands in 8-bit stages, an ARM7TDMI multiply instruction can take as few as two cycles (one to load the instruction and one for the first-stage multiply) or as many as five cycles (one load and four multiply). The benchmark results in Figure 1 assume that three instruction cycles are required for multiplies and multiplicands are limited to 16 bits, with a 32-bit result. Depending on the data used, typical ARM7TDMI code may execute significantly faster. Since real-time constraints are common in DSP applications, however, programmers may be unable to assume performance beyond the ARM7TDMI's worst-case instruction cycle counts.

ARM is aiming the ARM7TDMI at embedded applications requiring a very low-cost, low-power processor with high code density. The code density is provided by the Thumb extension (see [090401.PDF](#)), a mechanism designed to reduce code size by mapping the most crucial ARM7 instructions into a 16-bit instruction set. Thumb should be useful in reducing the size of the supervisory control code that surrounds many DSP algorithms. In fact, the ARM7TDMI has the smallest memory footprint of the processors tested in BDT's finite-state-machine benchmark, a measure of chips' code density in supervisory control code.

In many DSP applications, control code consumes only a small portion of processing time but accounts for the majority of the program memory requirements. In these cases, the Thumb instruction set may significantly reduce program memory requirements, supporting ARM's strategy of targeting cost-sensitive applications.

The ARM7TDMI's slow multiplier, limited memory bandwidth, modest clock speed, and lack of branch prediction or a hardware loop construct all conspire to give it by far the longest execution time of the benchmarked processors. Thus, the ARM7TDMI core is most appropriate for applications requiring only modest DSP performance.

For applications that need higher DSP performance, ARM has announced the Piccolo DSP coprocessor (see [MPR 101504.PDF](#)), which the company plans to integrate with the ARM7TDMI core. Piccolo will process DSP-oriented instructions while the ARM core performs other instructions, such as loading data, and should vastly improve the processor's DSP performance.

IDT R4650 Benefits from Locking Cache

The Integrated Device Technology (IDT) R4650 is a general-purpose 64-bit RISC processor family aimed at embedded applications. The R4650 (see [081504.PDF](#)) is derived from the R4600 MIPS processor, adding minor enhancements to

improve performance and reduce cost in DSP and other real-time applications. The DSP-oriented enhancements are modest, consisting mostly of a multiply-accumulate instruction and support for cache locking. Like the ARM7TDMI, the R4650 is very much a general-purpose processor, designed for moderately low cost.

The execution times of R4650 programs are very predictable, in contrast to those of many other general-purpose processors. This predictability is attributable to the processor's lockable cache, its single-issue architecture, and its lack of branch prediction or data-dependent execution times. These factors are an advantage for programmers seeking to optimize code and ensure deterministic performance.

Price/Performance Fairly Constant Among CPUs

Figure 2 shows the price/performance ratios for the general-purpose processors and some dedicated DSPs on BDT's complex block-FIR filter benchmark. Because the fastest versions of many chips, especially desktop processors, command a price premium, we chose the most cost-effective speed for each of the microprocessors. The ARM7TDMI is a core (as opposed to a packaged processor), so pricing is not available, and it is not included in this analysis.

The Texas Instruments TMS320C54x, a fixed-point DSP, and the Hitachi SH-DSP, which has a dedicated DSP data path, demonstrate the best price/performance by a wide margin. This is not surprising, since these processors contain fixed-point data paths, which generally require less silicon area than floating-point data paths, and memory architectures designed specifically for signal processing. One must remember, however, that a 16-bit fixed-point architecture delivers less-precise results than a floating-point architecture, so direct comparisons of fixed- and floating-point processors are tricky. The results for the PowerPC 604e, Pentium, and Analog Device's ADSP-21062 (SHARC) were calculated for floating-point arithmetic.

Of the two desktop processors, the 604e has the most competitive price/performance rating. In fact, the 604e's price/performance is substantially better than that of the Analog Devices ADSP-21062, a popular floating-point DSP. The 604e performs best when its on-chip cache is preloaded with the DSP program and data, a condition that may not be realistic for many applications. Even without the preloaded cache, however, the 604e scored extremely well, and better than competing floating-point DSPs. Pentium did not score quite as well in price/performance but was still competitive with floating-point DSPs overall.

In many cases, price/performance may not be a critical factor for Pentium or the 604e. Because these processors are central to the desktop PCs built around them, any DSP capability provided adds little additional cost to the system designer. Of course, designers may have trouble freeing enough of the host processor's time to ensure real-time execution of important DSP applications. Operating-system support for real-time applications is essential. This is espe-

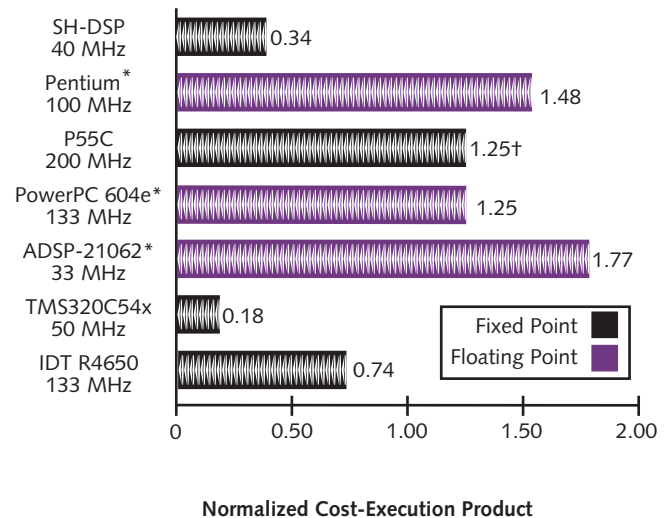


Figure 2. Price/performance ratios are indicated by the normalized price-execution product (chip price multiplied by execution time) for the BDT complex block-FIR filter (smaller is better). All prices are for quantity 1,000, as reported by the manufacturer. *code, data preloaded in cache. †performance and price estimated. (Source: Berkeley Design Technology)

cially true for Pentium and PowerPC, which have execution times that can vary widely from the typical case.

Microprocessors Begin Supplanting DSPs

General-purpose microprocessors have strong performance potential for DSP applications. Although high-end general-purpose microprocessors usually require more instruction cycles than DSPs to accomplish the same work, their designs allow for much faster clock speeds. These fast clock rates more than compensate for their high instruction-cycle counts and make high-end general-purpose processors top performers in DSP applications.

Unfortunately, DSP development tools for general-purpose processors have not kept pace with the rapid growth in raw performance, and designers must navigate past many potential pitfalls, such as execution-time unpredictability. Nonetheless, the incentives to integrate DSP capabilities are strong, and designers will increasingly want to use general-purpose processors for DSP applications.

Many vendors have already added, or announced plans to add, DSP capabilities to their general-purpose processors. In many cases, design changes may be possible to improve execution predictability. With the addition of better tools, DSP-oriented software libraries, and more applications support, along with continued improvement in performance, we expect the role of general-purpose processors in DSP applications to grow rapidly. □

Garrick Blalock is a DSP engineer at Berkeley Design Technology, specializing in analysis and evaluation of DSP technology. He is an author of BDT's new report, DSP on General-Purpose Processors, available through MicroDesign Resources by calling 707.824.4001.