

# TI Aims for Floating-Point DSP Lead

## DSP Giant Ups the Ante on VLIW With Powerful 'C6701

By Amit Shoham and Jeff Bier,  
Berkeley Design Technology, Inc. (BDTI)

Next month, Texas Instruments plans to reassert itself in the floating-point DSP market by sampling the TMS-320C6701, the first member of its new high-performance floating-point product family. With the introduction of the 'C6701, TI has put another egg in its VLIW basket, strengthening its commitment to the VLIW approach for high-performance DSPs.

Floating-point devices have historically accounted for a small share of the market for DSPs. Indeed, in recent years, two of the four major DSP vendors—Lucent and Motorola—have withdrawn from the floating-point market (formally or otherwise), leaving Analog Devices and TI as the only contenders. While TI's earlier TMS320C3x and TMS320C4x lines met with success, in recent years these families have stagnated as TI has focused on much higher volume fixed-point products. In the resulting competitive void, Analog Devices (ADI) has made significant inroads with its ADSP-2106x SHARC devices. (ADI recently announced a second generation of SHARC processors with SIMD enhancements and higher clock speeds.) Making matters worse, high-end CPUs have pulled ahead of DSP chips on floating-point DSP application performance.

With the 'C6701, TI is back in the floating-point DSP game. Using the same underlying VLIW-like architecture as

the fixed-point 'C6201 (see MPR 2/17/97, p. 14), the 'C6701 promises to make TI's floating-point DSP performance competitive with that of SHARC and of high-end general-purpose processors. In doing so, however, the new device makes the same sacrifices as the 'C6201: voracious program-memory usage, software-development complexity, high power consumption, and system-integration challenges. Users of floating-point DSPs are accustomed to sacrificing memory usage and power consumption compared with fixed-point devices, but the 'C67xx is even more resource-hungry than other floating-point DSPs.

### A First in Compatibility

Notably, the 'C67xx and 'C62xx families are the first floating-point and fixed-point DSP families to share the same underlying architecture: the instruction set of the 'C67xx is a superset of the 'C62xx instruction set, allowing the 'C67xx to execute 'C62xx object code. Additionally, the 'C6701 will be pin-compatible with the fixed-point 'C6201. Because of the differing priorities of fixed-point and floating-point DSP users, all previous generations of floating-point DSPs have been incompatible with their fixed-point siblings.

Developers of DSP-intensive applications often begin with a floating-point simulation of their application and can spend months porting it to a fixed-point implementation to minimize production costs. For applications where the 'C6xxx devices fit, users may create an initial floating-point product or prototype based on the 'C67xx and later migrate to the 'C62xx without having to switch to a very different type of device.

The pin-compatibility between the 'C6701 and 'C6201 will allow developers to use the same hardware design for an initial floating-point product and for a later fixed-point implementation. Unfortunately, the software compatibility between the 'C67xx and the 'C62xx will prove less advantageous than one might expect. Converting optimized 'C67xx assembly code to optimized 'C62xx assembly code will require almost complete rewriting and will be extremely challenging, due to the programming complexity of these devices. If programmers are willing to give up much of the performance advantage of the devices and write applications in C, however, the similarity between the two families will simplify porting to a modest degree.

### Up to One Billion FLOPS

In many respects, the 'C6xxx families resemble a high-end RISC architecture with an unusual

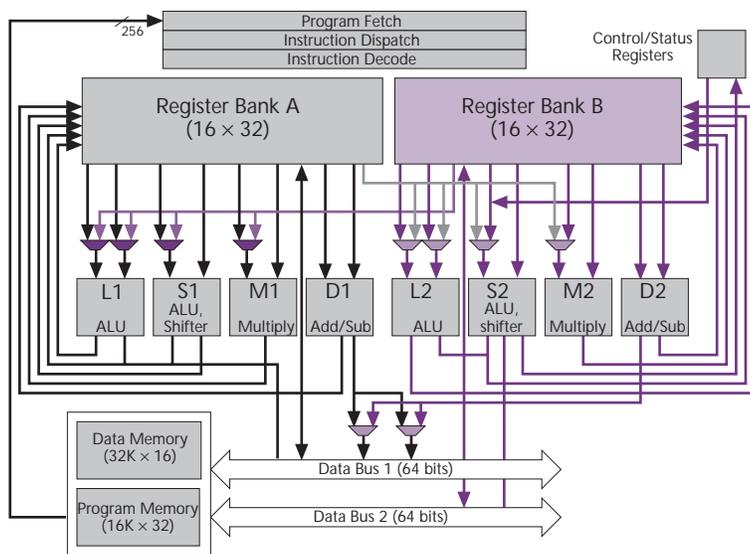


Figure 1. The block diagram of the 'C67xx core illustrates the eight execution units, arranged in two sets of four. The 256-bit-wide instruction bus allows the core to fetch eight 32-bit instructions per cycle.

instruction set more than a conventional DSP architecture. Like its fixed-point counterpart, the 'C67xx core is eight-way VLIW, has a very deep pipeline, and sports two sets of four execution units, as Figure 1 shows. The 'C67xx extends the 11-stage pipeline of the 'C62xx to a breathtaking 16 stages.

Like the 'C62xx, the 'C67xx is nominally divided in half, with 16 registers and four execution units on each side (A and B). The 'C6xxx families have a register-oriented architecture, providing 32 general-purpose 32-bit registers. In contrast, more traditional DSPs typically provide dedicated address registers, operand registers, and accumulators. Each of the execution units of the 'C67xx has unlimited access to the registers on its side of the device. Additionally, during each clock cycle, any one execution unit on each side may access one register on the opposite side of the device.

The two groups of execution units are nearly identical, and they perform the same functions as on the 'C62xx. Three of the execution units on each side, however, have been extended to support 32-bit floating-point arithmetic. Each group contains a floating-point multiplier (M), a floating-point adder (L), a floating-point unit for comparisons and other miscellaneous operations (S), and a load/store unit (D).

With three FP execution units per side, the 'C67xx is capable of up to six FP operations per clock cycle, or one billion FP operations per second at its projected clock speed of 167 MHz. But keeping all six FP execution units busy during every clock cycle requires an even mix of multiply, add, and miscellaneous operations, such as compare or absolute value. This instruction mix is uncommon in DSP applications, so the chip's peak performance will rarely be achieved in applications.

With one FP multiplier and one FP adder per side, the 'C67xx is capable of performing up to 334 million multiply-accumulates per second at 167 MHz—assuming software pipelining or loop unrolling is used to cover the latencies of the multiply, add, and load operations. By this somewhat more realistic performance measure, the 'C67xx is still faster than any other floating-point DSP.

The two FP adders also perform 40-bit fixed-point arithmetic as well as logical compares, normalization, bit-count operations, and integer/FP conversions. Multiplication is handled by the M units. In addition to FP multiplication, the M units can perform both signed and unsigned  $16 \times 16 \rightarrow 32$ -bit multiplication and  $32 \times 32 \rightarrow 32$ -bit or  $32 \times 32 \rightarrow 64$ -bit integer multiplication.

The S units perform FP comparison, absolute value, and reciprocal or reciprocal-square-root estimate operations. The S units also have a 32-bit fixed-point ALU and a 40-bit shifter. These units can perform some of the same 32-bit fixed-point arithmetic operations as the L units, along with 32-bit and 40-bit shifts. One S unit is also responsible for branching and branch-address generation.

A 32-bit adder allows the D units to perform simple fixed-point arithmetic operations, but their primary purpose is address generation.

Most fixed-point operations execute in a single cycle (multiplies, loads, and branches are exceptions), but floating-point operations have longer latencies (typically three to five cycles), complicating software development.

### Double-Precision Arithmetic Takes Longer

The 'C67xx is the only DSP to support both single-precision and double-precision IEEE-754 floating-point arithmetic. All floating-point instructions include both single- and double-precision variants, and instructions to convert between the single- and double-precision formats are also provided. Double-precision values are stored in adjacent pairs of registers, as are 40-bit fixed-point operands.

While all single-precision arithmetic instructions have single-cycle throughput, most double-precision arithmetic instructions stall the corresponding execution unit for one or three cycles and have longer latencies than their single-precision variants.

These stalls and other restrictions considerably reduce the performance of the 'C67xx on double-precision arithmetic compared with single-precision arithmetic. For example, the 'C67xx is capable of up to 83.5 million double-precision multiply-accumulates per second at 167 MHz compared with 334 million single-precision multiply-accumulates per second at the same clock rate.

For the vast majority of signal-processing applications, single-precision arithmetic is sufficient. However, for those rare applications that require double precision, the 'C67xx will have a strong performance advantage over other floating-point DSPs, which must emulate double-precision operations in software.

### Up to 256 Bits of Instructions Per Cycle

The 'C67xx has a VLIW-like architecture like that of its fixed-point predecessor, most accurately described as a statically scheduled superscalar machine. As on the 'C62xx, the 'C6701 core consumes eight 32-bit instructions at once from its on-chip 256-bit instruction bus. The eight-instruction group, known as a fetch packet in TI's nomenclature, must be 32-byte aligned. The 'C67xx always fetches a complete fetch packet at once. However, not all eight instructions in the fetch packet are necessarily executed simultaneously.

Independent of the 256-bit fetch packet, the 'C67xx defines an execute packet, which can be 1–8 instructions long. All instructions in an execute packet are dispatched together. A bit in each instruction indicates whether that instruction is the last one in its execution packet. It is the programmer's (or compiler's) responsibility to guarantee that all instructions in the execute packet can, indeed, be dispatched simultaneously. The 'C67xx hardware does no dependency checking among instructions.

Although there are eight instructions in each fetch packet, and eight execution units, each instruction does not necessarily correspond to one execution unit; the instructions are not position-dependent within the fetch packet,

which is the traditional VLIW method. Instead, each instruction is encoded for a specific execution unit.

Some fixed-point instructions can be encoded for multiple types of execution units, but the encoding is fixed prior to run-time. The fixed-point ADD instruction, for example, can be encoded for the L1, L2, S1, S2, D1, or D2 units. Floating-point instructions must be encoded for a particular type of execution unit; the floating-point ADDSP instruction, for example, can be encoded only for the L1 or L2 units. Programmers can explicitly dictate the binding of instructions or leave it to the assembler or compiler.

Under ideal circumstances, all eight of the 'C67xx's execution units can be kept busy on every cycle. In practice, data dependencies, resource conflicts, multicycle operations, and other realities of programming will force less than total utilization of the core's resources. Rather than waste space in the fetch packet by padding with NOPs, TI allows multiple execution packets in a single fetch packet.

### RISC-Like Instructions With Predication

Table 1 lists the new instructions found in the 'C67xx but not in the fixed-point 'C62xx. Every 'C67xx instruction (including branches) can be predicated, or executed conditionally, based on the zero/nonzero status of the five condition registers. Theoretically, all eight instructions in a packet could each be predicated on a different condition. This type of predicated execution is also used in the Philips Trimedia architecture (see MPR 11/13/95, p. 22), a VLIW media processor.

The 'C67xx has none of the moderately complex multi-operation instructions most DSP chips have. Multiply-accumulate, for example, is handled as a multiply followed

by a separate add. Fetching a memory-resident coefficient requires a third, independent, operation. Loops must also be explicitly coded in software; there is no intrinsic zero-overhead loop feature in the 'C67xx. Loop counters must be explicitly decremented, with a conditional branch used to return to the top of the loop.

Because the 'C67xx divides common DSP operations into separate instructions, performance comparisons with conventional DSPs are tricky. A single multiply-accumulate becomes three or four different instructions on the 'C67xx compared with a single instruction on a conventional DSP. This makes MIPS a poor performance metric when comparing the 'C67xx with other DSPs. In promoting the fixed-point 'C6201, TI has focused heavily on its "1,600 MIPS" performance claim, leaving many users to discover for themselves that 'C6201 MIPS is defined quite differently from traditional DSP MIPS. The 'C6701 presents the same hazard.

### More Memory, Please!

TI's initial implementation of the 'C67xx architecture is the TMS320C6701. The part has 128 Kbytes of on-chip memory, evenly divided between program and data space.

The on-chip program memory has a 256-bit path into the 'C67xx core, allowing it to transfer an entire eight-word fetch packet in one cycle. In contrast, off-chip memory accesses occur over a 32-bit external bus, requiring at least eight cycles to transfer an eight-word fetch packet. The 'C67xx must therefore execute from on-chip program memory for good performance. At the user's option, the program memory can be configured as a 64K direct-mapped cache.

The 'C6701's on-chip data memory is divided into eight 8K banks, each with a 16-bit bus to the execution units. All eight banks can be accessed simultaneously, but simultaneous accesses to the same bank are not allowed. This configuration allows an on-chip access rate of four 32-bit words per cycle, enough bandwidth to support the core's peak processing rate of two multiply-accumulate operations per cycle. Off-chip data memory accesses incur severe penalties, so avoiding them is paramount.

VLIW processors traditionally suffer from very high program-memory usage. TI has taken steps to mitigate this problem in the 'C6xxx architectures. Although these steps alleviate the problem to some degree, the 'C67xx's memory usage will still be significantly higher than that of a more traditional DSP. Thus, the 64 Kbytes of on-chip program memory are equivalent to about 16–32 Kbytes of program memory on a traditional floating-point DSP.

The amounts of on-chip program and data memory available on the 'C67xx are sufficient for many of the applications that

Mnemonic	Description	Ex Unit			
		L	M	S	D
<b>Floating-Point Arithmetic</b>					
ABS(SP/DP)	Absolute value (single-/double-precision)			•	
ADD(SP/DP)	Add (single-/double-precision)	•			
MPY(SP/DP)	Multiply (single-/double-precision)		•		
SUB(SP/DP)	Subtract (single-/double-precision)	•			
RCP(SP/DP)	Reciprocal estimate (single-/double-precision)			•	
RSQR(SP/DP)	Reciprocal square root estimate (single/double)			•	
<b>Fixed-Point Arithmetic</b>					
ADDAD	Add doubleword address nonsaturating				•
<b>Conversion</b>					
DPSP	Convert double to float	•			
SPDP	Convert float to double			•	
(SP/DP)INT	Convert (float/double) to integer	•			
SPTRUNC	Convert float to integer, and truncate instead of round	•			
INT(SP/DP)(U)	Convert integer to (float/double) (unsigned)	•			
<b>Load/Store</b>					
LDDW	Load doubleword				•
<b>Floating-Point Comparison</b>					
CMPEQ(SP/DP)	Compare for equality (single-/double-precision)			•	
CMPGT(SP/DP)	Compare for greater-than (single-/double-precision)			•	
CMPLT(SP/DP)	Compare for less-than (single-/double-precision)			•	

Table 1. New 'C6xxx instructions introduced in the 'C67xx. The 'C67xx instruction set is a superset of the 'C62xx instruction set (see MPR 2/17/97, p. 14), adding support for floating-point arithmetic and doubleword loads and address calculations.

utilize fixed-point or older floating-point DSPs, but they are too small for many of the applications that merit the cost of a high-performance floating-point processor. In contrast, Analog Devices' new ADSP-21160 provides 512 Kbytes of on-chip memory and generally consumes less program space than the 'C67xx.

Floating-point DSPs are often used in multiprocessor configurations. Devices such as TI's '320C4x and ADI's ADSP-2106x have included extensive features to facilitate integration in multiprocessor environments, such as specialized external memory interfaces and multiple inter-processor communication ports. Surprisingly, the 'C6701 omits these features, making do with a bare-bones complement of interfaces consisting of its external memory interface, two serial ports, and a host port. Of course, TI may add multiprocessor features later, but, for now, their absence will limit the 'C6701's appeal for many of the applications that have gobbled up large numbers of floating-point DSPs in the past.

### An Assembly Programmer's Worst Nightmare

As with the 'C62xx, crafting carefully arranged object code is crucial to extracting performance from the 'C67xx. This will be no easy task.

As mentioned previously, the chip does no dependency checking and incorporates no interlocks; multiple writes to the same destination register give undefined results. Avoiding this condition can be harder than it sounds, because not all instructions have the same latency. For example, issuing an FP add instruction one cycle after a load with the same destination will cause a failure because of their different latencies.

Packing two mutually exclusive conditional instructions in the same execute packet is not a programming error and, in fact, can be a good idea. Programmers can create their own conditional moves, adds, or other functions simply by combining conflicting instructions that are predicated on opposite states of the same condition. Again, there is an opportunity for mischief here, as the 'C6xxx software tools cannot check for unintended conflicting instructions that are predicated on the contents of unrelated registers.

Branches introduce further programming complexity. Since the chip has no branch prediction, all taken branches introduce a five-cycle delay before the pipeline refills from the branch target, as Figure 2 illustrates. The 'C67xx executes instructions in the branch-delay slot, which in this case has space for 40 instructions (5 cycles  $\times$  8 instructions).

The processor is not interruptible while any execute packet in the pipeline contains a branch or is in the delay slot of a branch. Given the long branch latency, this fact means the processor is rarely interruptible, rendering interrupts useless in many applications.

Manual scheduling on the 'C67xx is extremely complicated. Figure 3 shows the kernel of a two-tap FIR filter

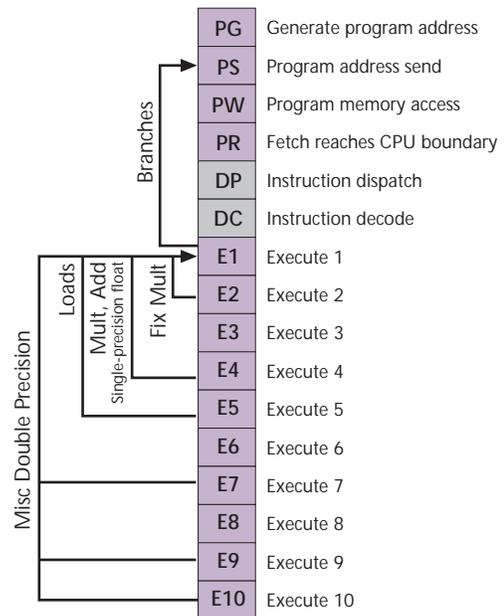


Figure 2. The 'C67xx has a 16-stage pipeline, but few operations use more than 11 stages. Floating-point adds and multiplies complete in stage E4, and loads complete in stage E5, for example.

implemented in a single repeating execute packet. It performs two multiplies, two adds, and two loads while it decrements the loop counter and branches back to itself.

The effects of this packet are difficult to deduce from a cursory reading of the source code, complicated by the fact that adds, multiplies, loads, and branches have different latencies (four, four, five, and six clocks, respectively).

On any given iteration of this loop,  $n$ , the 'C67xx resolves the multiplies and additions executed on iteration  $n-4$ , the data loaded on iteration  $n-5$ , and the branch encountered on iteration  $n-6$ . Once under way, this loop executes two taps per cycle, better than most DSPs and, at 167 MHz, faster than all but the 200-MHz 'C62xx.

The deep software pipeline surrounding this loop (not shown in the figure), however, takes 5 clock cycles and 21 instructions to fill, and an additional 13 clock cycles and 16 instructions to flush. The extra cycles can significantly detract from the chip's peak performance when loops are nested or when iteration counts are low. Moreover, program-memory usage can be more than an order of magnitude greater than for traditional DSPs.

```

LOOP:
[B0] LDDW .D1 *A4++[1],A7:A6 ; if(B0!=0) load A7:A6
[B0] LDDW .D2 *B4++[1],B7:B6 ; if(B0!=0) load B7:B6
      MPYSP .M1X A6,B6,A5      ; A5 = A6 * B6
      MPYSP .M2X A7,B7,B5      ; B5 = A7 * B7
      ADDSP .L1 A5,A8,A8        ; A8 = A8 + A5
      ADDSP .L2 B5,B8,B8        ; B8 = B8 + B5
[B0] B .S1 LOOP ; if B0 not zero, loop
[B0] SUB .S2 B0,2,B0 ; decrement counter
    
```

Figure 3. In this example of an FIR filter, eight instructions fit in a single execute packet, executing in parallel and calculating two taps per iteration.

Processor	Vendor	Execution Speed (MHz)	Program Time ( $\mu$ S)	Size (bytes)
'C6701*	TI	150	12.1	444
'C44	TI	30	99.0	96
ADSP-21160*	Analog Devices	100	15.4	156
ADSP-21065L	Analog Devices	60	46.9	180
PowerPC 604e	Motorola/IBM	350	16.9	192

Table 2. The results of BDTI's complex FIR-filter benchmark show the 'C6701 to be the highest performing floating-point processor of those evaluated. \*These processors are not yet available. Clock speeds for these processors are BDTI's projections, and benchmark results are preliminary. (Source: *Buyer's Guide to DSP Processors, 1999 Edition*, BDTI)

Manually scheduling the processor's eight execution units for optimum performance is a daunting task. As with the 'C62xx, TI will provide a C compiler and assembly optimizer to ease software development. The assembly optimizer accepts "linear" meta-assembly code that is not parallelized and that assumes single-cycle latency for all instructions. The optimizer attempts to transform this code into a scheduled, optimized form.

Although the optimizer reduces the difficulty of writing assembly code for the 'C6xxx families, debugging code generated by the optimizer is much more difficult than debugging assembly code for more traditional DSP processors, and the performance of the optimizer for the 'C62xx has so far been uneven. Most programmers will find the 'C67xx too complex for assembly-level coding, forcing them to program in C and give up some of the part's impressive performance.

### Promising Floating-Point DSP Performance

The final question is whether the performance of the 'C67xx is as daunting as its programming model. Although no hardware-verified numbers are available at this time, BDTI has run some preliminary benchmarks on a cycle-accurate simulator of the 'C6701.

Although TI announced the 'C6701 at 167 MHz, our analysis uses a more conservative speed of 150 MHz to evaluate the likely performance of initial 'C6701 devices. The 'C6701's fixed-point cousin, the 'C6201, was announced by TI at 200 MHz, but initial samples ran at approximately 120 MHz; and 200-MHz samples did not become generally available for approximately a year after the first samples were provided. It isn't clear whether this unusually long time lag was part of a deliberate strategy by TI to gain an apparent advantage by preannouncing, or whether TI simply had difficulty obtaining full-speed silicon. Either way, some skepticism seems warranted until 'C6701 devices become available.

BDTI's complex-FIR filter benchmark is an FIR filter that operates on blocks of complex data. Such filters are commonly used in modem channel-equalization applications, for example. A simulated 'C6701 finished the BDTI complex FIR filter benchmark in 12.1 microseconds, 20% less time than our projected result for the Analog Devices new ADSP-21160. As Table 2 shows, the 'C6701s are much

### Price & Availability

According to Texas Instruments, initial samples of the 'C6701 are expected in October with production quantities to ship in the second quarter of 1999. Initial pricing is \$196 (quantity 10,000). The device will be fabricated in TI's 0.18-micron process and packaged in a 352-pin BGA. Power consumption is projected to be 1.9 W at 1.8 V and 167 MHz. Further details are available at [www.ti.com/sc/docs/dsp/products/c6000/c67x/index.html](http://www.ti.com/sc/docs/dsp/products/c6000/c67x/index.html).

faster than older floating-point DSPs: roughly four times faster than the ADSP-21065L and more than eight times faster than TI's own 'C44 on this benchmark.

In the past few years, high-end general-purpose processors have outpaced floating-point DSPs, providing better FP performance. The new 'C67xx and ADSP-21160 can compete head-to-head with the FP performance of general-purpose processors. The 'C6701 and ADSP-21160 execution times on BDTI's complex-block FIR filter benchmark are faster than that of the 350-MHz PowerPC 604e, as Table 2 shows. As general-purpose processors begin to provide floating-point SIMD capabilities in the form of multimedia extensions such as AltiVec and KNI, however, they may regain the FP performance lead.

As expected, code density leaves much to be desired on the 'C67xx. The 'C67xx binary for the BDTI complex-FIR filter benchmark was almost three times larger than that for the ADSP-21160 and more than four times larger than that for the 'C44.

### Back in the Game

With the 'C6701, TI is making a big splash in the floating-point DSP business after several quiet years. While floating-point devices represent only a small share of the total market, it appears that the DSP giant is not content to cede leadership of any segment of the market to its competitors.

If TI takes as long to reach its projected 167-MHz clock speed for the 'C6701 as it did for the 'C6201, however, the Analog Devices ADSP-21160 may deliver similar performance without the 'C6701's shortcomings in on-chip memory and multiprocessor support, and with a simpler programming model. Because of its pin-compatibility with the 'C6201, the 'C6701 is likely to find a niche as a rapid prototyping and algorithm-development tool for users who plan to build products based on the 'C6201 but want prototypes running quickly, without first having to grapple with fixed-point considerations. □

*Authors Amit Shoham and Jeff Bier are with Berkeley Design Technology, Inc., the DSP technology analysis and software development firm. Shoham and Bier are co-authors of Buyer's Guide to DSP Processors, the 1999 edition of which will be available from MDR in October.*