Cahners
MICROPROCESSOR REPORT
www.MPRonline.com
THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

TI Cores Accelerate DSP Arms Race

New 'C64x and 'C55x DSPs Battle Analog Devices, StarCore, Intel

By Tom R. Halfhill {3/6/00-01}

Everything is bigger in Texas, including the DSPs. The Texas Instruments TMS320C62x-series DSP core, already the T. Rex of digital-signal processing, is about to be surpassed by an even more powerful beast. TI says its new TMS320C64x core offers about 10 times the performance of the existing core—plus greater code density and full compatibility with 'C62x software.

TI isn't ignoring the opposite end of the market either. A second new core, the 'C55x, supplements the popular 'C54x and brings higher performance, lower power consumption, and greater code density to low-power DSPs. While the 'C64x is suitable for wireless base stations and other communications-infrastructure applications, the 'C55x is TI's first DSP core to meet the performance and power-consumption requirements of third-generation (3G) cellular phones. The 'C55x is scheduled to enter production in 2H00 and the 'C64x in 1H01.

If TI meets its deadlines and performance targets—the company has slipped on deliveries of new DSPs in the past—it will be just in time to counter the threat of new competition from Lucent, Motorola, Analog Devices (ADI), and Intel. The Lucent-Motorola StarCore alliance unveiled an impressive DSP core last year that is strong competition for TI's high-end cores (see *MPR 5/10/99-03*, "StarCore Reveals Its First DSP"). Motorola is the first partner to announce a StarCore SC140-based chip (see *MPR 10/6/99-03*, "First StarCore DSP Targets Networking"). In November, ADI announced its first TigerSHARC chip, the ADSP-TS001, another high-end DSP (see *MPR 12/6/99-en*, "ADI's First TigerSHARC DSP Has Sharp Teeth"). And this summer, ADI and Intel will reveal the fruit of their alliance to jointly develop a low-power DSP that will compete against TI's 'C55x (see *MPR 2/15/99-en*, "Intel, Analog Devices Team for DSPs").

All these new DSPs should enter production within the next year or so. That's not a coincidence—the rapid growth of wired and wireless communications is creating huge demand for DSPs, both on the client and the infrastructure sides of the networks. Other fast-growing applications are digital imaging, audio/video processing, automotive control, and medical equipment. Market researchers at Forward Concepts predict that sales of programmable DSPs will more than triple over the next three years, from $4 billion in 1999 to $13 billion in 2003. Last year, TI had a 48% share of that market—greater than ADI, Lucent, and Motorola combined, according to Forward Concepts.

When Overkill Isn't Enough

TI created a sensation when it announced the 'C62x series in 1997 (see *MPR 2/17/97-04*, "TI's New 'C6x DSP Screams at 1,600 MIPS"). That core was five times more powerful than any other DSP on the market. Only three years have gone by since then, but already TI needs to boost performance by the same degree just to keep pace with the latest competition, such as StarCore's SC140. The new 'C64x tries to get there with higher clock frequencies and numerous architectural improvements.

This summer, TI plans to sample the first 'C64x chips at 600MHz or better on its 15C05 CMOS process, which is a five-layer-metal 0.15-micron (L-effective) process. The chip's core voltage will be 1.5V, with 3.3V I/O. Volume production is scheduled to begin early in 2001. TI says clock speeds will ramp almost immediately to about 800MHz. The company's roadmap anticipates frequencies in the 1.1GHz range in 1H01, after it moves the core to an improved version of the 15C05 process with some 0.13-micron elements. So even without architectural improvements, the higher frequencies will give the 'C64x a significant performance advantage over existing 'C62x-series chips. The fastest DSP shipping from TI is the 'C6203, which runs at 250MHz. A newer version of that chip, built on the 15C05 process, is currently sampling at 300MHz. More aggressive circuit design with dynamic logic allows the 'C64x to run at higher clock rates than the older core, even when they're built on the same IC process.

Architectural improvements in the 'C64x core include more-powerful function units, twice as many registers, additional read/write ports to the register files, double the data bandwidth, provisions for larger on-chip caches, more support for packed-data processing (including single-instruction multiple-data [SIMD] arithmetic), and a new ability to arrange instructions across multiple VLIW bundles for parallel execution. The 'C64x is still a 32-bit, fixed-point, eight-issue machine like the 'C62x, but these improvements allow the new core to keep its function units busy more often, thereby wringing even more instruction-level parallelism (ILP) and performance out of code.

Figure 1 is a block diagram of the 'C64x core that shows some of the enhancements. Like its predecessor, the 'C64x has two mirror-image banks of registers and function units served by independent data paths. Together, these banks can execute as many as eight instructions per cycle. But whereas the older core had a pair of 32-bit load/store buses and 16 registers in each bank, the 'C64x has dual 64-bit load/store buses and 32 registers in each bank. Doubling the data bandwidth should relieve any bottlenecks that would have resulted from increasing the ILP. Likewise, doubling the number of registers makes it easier for the compiler to keep data at hand when the instructions are ready to execute.

The register files are even more massively ported than those in the older core. Each register bank in the 'C64x has eleven 32-bit read ports and eight 32-bit write ports—that's one more read port and two more write ports than the 'C62x. The extra read port and one of the extra write ports are for double-word (64-bit) stores. The second new write port allows the MU0/MU1 multiplier units to store the results of two 32-bit multiplies per cycle. All four function units in each bank can access the same register during the same clock cycle if necessary.

Two crossover buses allow each bank of function units to access registers in the other bank. In the 'C62x, only one function unit per cycle could access a register in the opposite bank. That limitation is relieved somewhat in the 'C64x, which allows all the function units in a bank to access the same register in the opposite bank during each cycle. Still, it's faster for operations that need multiple operands to confine themselves to a single bank. It's up to the compiler or the programmer to schedule instructions in a way that minimizes data crossovers. With twice as many registers in each bank, the 'C64x should make minimization of crossovers easier than the 'C62x does, because each bank can hold twice as many operands.

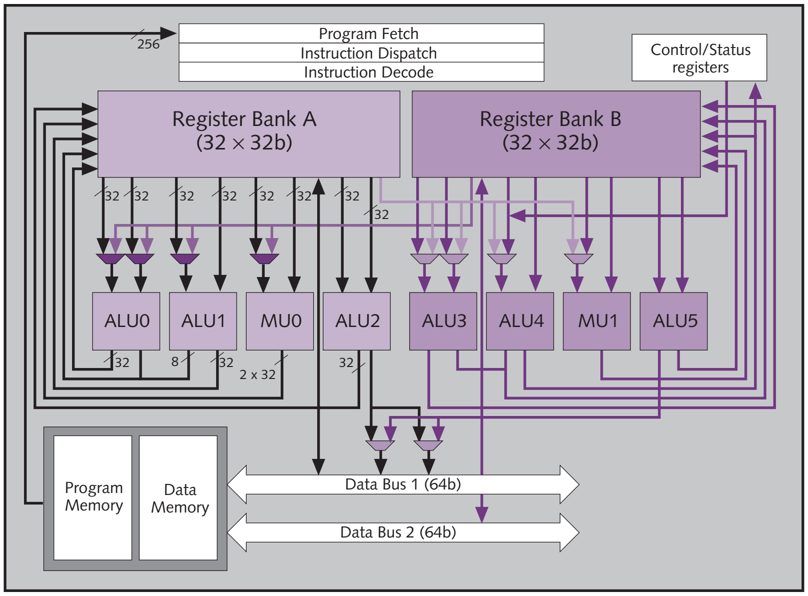

Figure 1. The eight-way 'C64x has two banks of registers and function units, like the 'C62x, but it also has twice as many registers in each bank and data buses that are twice as wide.

Shuffle and Deal

Figure 2 shows the resources in each of the function units that appear in smaller scale in Figure 1. (The function units in both execution paths are identical.) The 'C64x can issue as many as eight instructions in parallel to these units, as can the 'C62x, which has similar resources. But TI significantly increased the amount of logic in the new function units. This allows the 'C64x to execute more types of operations in parallel with each other, so it can achieve its ideal of eight-way parallelism more often. The enhanced function units also support several new instructions and SIMD operations.

In the MU0/MU1 multiplier units, for example, TI added a second 16-bit multiplier, a shifter/rotator, some new logic for bit interleaving/deinterleaving, and a Galois-field multiplier, which speeds up Reed-Solomon encoding (important for noise reduction in telephony applications). As a result, each of these units can now perform two 16-bit multiply-accumulate instructions (MACs) per cycle, or four 8-bit MACs per cycle, in parallel. They can also execute some of the new instructions seen in Table 1, including SHFL, DEAL, and GMPY4, which are useful for convolutional and Reed-Solomon encoding.

In addition to the SIMD arithmetic instructions, TI added some new support for packed-data operations. In these operations, the 'C64x can convert shorter data types to longer types, and vice versa, to preserve greater mathematical precision during intermediate calculations. For example, the processor can load two 8-bit values and expand them to a pair of 16-bit values packed in a 32-bit register. After performing some parallel arithmetic operations on the 16-bit values, the processor can saturate and pack the 16-bit results into 8-bit values before storing them in memory. Special unaligned load/store instructions allow the processor to access memory locations that aren't aligned on 32-bit word boundaries. Using these instructions, the 'C64x can access up to 64 bits of data per cycle at any byte boundary.

Packed-data processing is especially useful for programs that manipulate 8-bit pixels. The DSP can unpack the 8-bit pixel values into 16-bit integers for more precise calculations, then pack them into saturated 8-bit values before storing the final results in memory. This operation goes hand in hand with new SIMD instructions such as AVGx, which finds the average of four 8-bit or two 16-bit integers, and SUBABS4, which calculates the sum of absolute differences among two pairs of 8-bit integers. Such instructions are valuable for the motion-estimation and motion-compensation algorithms in MPEG video software.

NOPs Begone!

Although the 'C64x instruction set is larger, typical application code will be about 25% denser, according to TI. To pull off this trick, TI cleverly modified the VLIW format to eliminate almost all NOPs, (null operations) from the code.

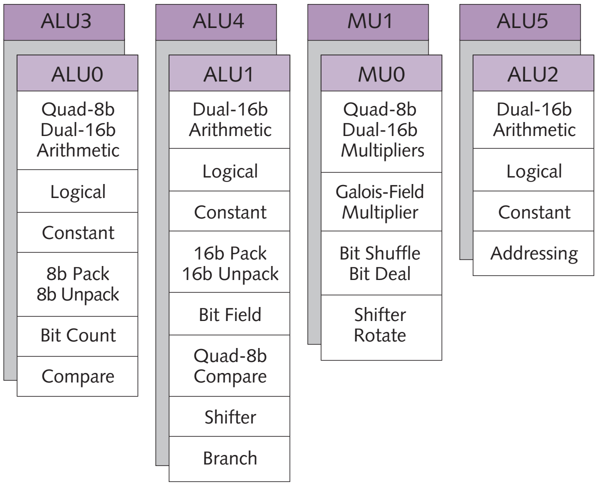

Figure 2. All eight function units in the 'C64x (four in each execution path) have additional logic over those in the 'C62x, so they can execute new instructions and more operations in parallel.

In the 'C6000 architecture, the processor fetches 256 bits per cycle, and those 256 bits contain eight 32-bit subinstructions. TI refers to this bundle as the "fetch packet" to distinguish it from an "execution packet," which is a group of instructions the processor can execute in parallel. Ideally, execution packets would always be the same as fetch packets, so all eight function units could execute an instruction every cycle. In practice, there's not always enough ILP in a program to issue eight instructions in parallel, so most execution packets are shorter than fetch packets. An execution packet can be as short as one subinstruction or as long as eight subinstructions.

One bit at the end of each subinstruction determines the length of an execution packet: 1 means the next subinstruction is part of the same execution packet, while 0 indicates the end of an execution packet. In the original 'C6000 architecture, an execution packet cannot span multiple fetch packets, so fetch packets always end with 0. This causes a problem if an execution packet is longer than the remaining number of subinstruction slots in a fetch packet. In that case, the assembler or compiler usually pads the remaining slots with NOP instructions to push the longer execution packet to the beginning of the next fetch packet. NOPs are the bane of VLIW architectures, because they wastefully inflate the code size and consume instruction-fetch bandwidth. TI's variable-size execution packets greatly reduced the incidence of NOPs compared with their occurrence in traditional VLIW architectures, but they didn't eliminate NOPs entirely.

For the 'C64x, TI modified the VLIW format to allow execution packets to span multiple fetch packets. As Figure 3 shows, a fetch packet can now end with 1 to indicate that an execution packet continues in the next fetch packet. As a result, assemblers or compilers no longer have to pad unused slots in a fetch packet with do-nothing NOPs.

TI estimates that squeezing out the NOPs will reduce the code size up to 25%. If that estimate proves to be accurate, TI has made a major improvement that brings the 'C64x more in line with the code density offered by newer VLIW-based DSP architectures, such as StarCore's SC140

Instruction	Description	Sample Applications
BITC4	Bit count	Machine vision
GMPY4	Galois-Field multiply	Reed-Solomon coding
SHFL	Bit interleaving	Convolutional encoder
DEAL	Bit deinterleaving	Cable modems
SWAP4	Byte swap	Endian swap
XPNDx	Bit expansion	Graphics processing
MPYHIx	Extended-prec 16x32 MPY	Audio processing
MPYLIx	Extended-prec 16x32 MPY	Audio processing
AVGx	Quad 8b, dual 16b average	Motion compensation
SUBABS4	Quad 8b absolute diffs	Motion estimation
SSHVL	Signed-variable shift left	Telephony vocoders
SSHVR	Signed-variable shift right	Telephony vocoders

Table 1. The 'C64x has several new instructions that target communications-related applications and increase parallelism.

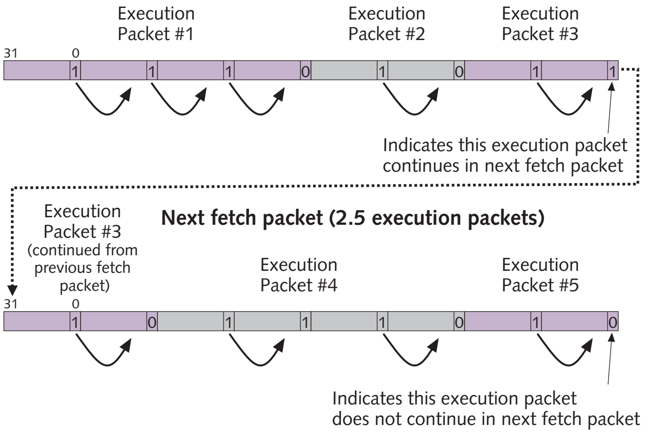
Execution Packet #1
Execution Packet #2
Execution Packet #3
31
0
1
1
1
0
1
0
1
1
Indicates this execution packet continues in next fetch packet
Execution Packet #3 (continued from previous fetch packet)
Next fetch packet (2.5 execution packets)
Execution Packet #4
Execution Packet #5
31
0
1
0
1
1
1
0
1
0
Indicates this execution packet does not continue in next fetch packet

Figure 3. By allowing execution packets to span multiple fetch packets, the 'C64x eliminates unnecessary NOPs, resulting in greater code density.

and ADI's TigerSHARC. Those architectures use variable-size VLIW bundles to accomplish the same thing. TI says it designed the 'C6000 architecture with this extension in mind but just didn't have time to implement it earlier. Because the extension was planned, software written for existing 'C62x-series DSPs will run unmodified on new 'C64x chips, although the reverse is not true. To reap the code-density benefits of the new format, developers will have to reassemble or recompile their code to purge the NOPs.

Running a Tight Race

The higher clock speeds and architectural enhancements will push the 'C64x well beyond the performance of the 'C62x, but it's the new competition that TI has to worry about. In 1997, the 'C62x had no credible competitors; that's not true today. Within the next year, ADI and StarCore partners Lucent and Motorola will ship DSPs that the 'C64x will be hard-pressed to beat. (Interestingly, all of these high-end DSPs are based on VLIW-type architectures.)

Clock speeds will be the key factor. StarCore's SC140 can execute as many 16-bit MACs per cycle as the 'C64x (four), and it's sampling now at 300MHz. To beat that, TI will have to deliver the first 'C64x chips at 600–800MHz as promised. If TI also meets its goal of migrating to an improved IC process with 0.13-micron geometries within a year, 'C64x DSPs could be running at 1GHz in 1H01.

ADI's TS001 TigerSHARC can execute eight 16-bit MACs per cycle—twice as many as the 'C64x—but it will debut at only 150MHz, so the 'C64x has a two-to-one advantage on that benchmark even at TI's conservative target of 600MHz. If ADI and StarCore ramp up their clock speeds at a faster pace, this will be a very tight race indeed.

A close look at Table 2 explains why the 150MHz TigerSHARC appears to be lagging so far behind the pack. TigerSHARC will debut on an older IC process (0.25 micron instead of 0.15/0.18 micron) and has a more complex architecture. No other DSP can handle both 8-, 16-, and 32-bit fixed-point data and 32-bit floating-point data. TigerSHARC can execute two 32-bit floating-point MACs per cycle, while the 'C64x and StarCore DSPs can execute none. It might be fairer to compare TigerSHARC with TI's 'C67x, a 32-bit floating-point DSP that's code compatible with the 'C62x. The 'C67 is shipping now at 167MHz, while the 150MHz TigerSHARC won't be available until 2H00.

If TI follows the pattern it established with 'C62x and 'C67x DSPs, it will introduce a floating-point version of the 'C64x at some point in the future. For most communications and media applications, however, fixed-point math is sufficient.

Looking further ahead to new competition, Lucent's Bell Labs is working on a new internal bus architecture that links numerous DSP cores on a single die and allows them to share on-chip memories. Lucent previewed the bus, known as Daytona, at the recent International Solid-State Circuits Conference (ISSCC) in San Francisco.

A prototype Daytona chip integrates four 100MHz DSP cores on a die to achieve 3.2 billion operations per second. Lucent is also developing a 200MHz chip with 32 DSP cores that will exceed 51 billion operations per second. But Lucent hasn't announced when Daytona chips will be available. They are unlikely to appear before 1H01, when all of these other DSPs should be shipping.

	C64x	C62x	C67x	SC140	ADSP-TS001
Feature	**TI**	**TI**	**TI**	**StarCore**	**ADI**
Suppliers	TI	TI	TI	Lucent, Motorola	ADI
Data Type	32b fixed point	32b fixed point	32b floating point	16b fixed point	32b fixed & float
Architecture	VLIW	VLIW	VLIW	VLIW	VLIW
VLIW Length	32–256b	32–256b	32–256b	96–128b	32–128b
Subinstruction Length	32b	32b	32b	16–48b	32b
Register File	64 x 32b	32 x 32b	32 x 32b	16 x 40b	128 x 32b
Instructions Per Cycle	8	8	8	5	6
16 x 16 MACs (Fixed)	4 per cycle	2 per cycle	2 per cycle	4 per cycle	8 per cycle
32 x 32 MACs (Float)	—	—	2 per cycle	—	2 per cycle
Core Frequency	600MHz	300MHz	167MHz	300MHz	150MHz
IC Process	0.15µ	0.15µ	0.18µ	0.18µ	0.25µ
Production	1H01	3Q00*	Now	2001	2H00

Table 2. New competition is nipping at the heels of the 'C64x, but TI hopes that higher clock speeds will give its DSPs a clear advantage in 2001. (*TI is currently shipping a 250MHz version of the 'C62x.)

’C55x Targets Mobile Devices

Wireless phones outsold PCs by more than two to one last year (283 million versus 113 million), according to Dataquest. With sales expected to climb 44% to 410 million next year—far outstripping growth in the PC market—the demand for microcontrollers and DSPs is skyrocketing. TI is a major player in this business, counting Nokia and Ericsson among its customers. TI’s existing ’C54x-based DSPs are found in more than 60% of digital cell phones. By 2003, the industry will start selling 3G wireless phones that can handle video as well as audio and text. That means embedded chips have to deliver higher performance while still meeting consumer expectations of smaller phones and longer battery life.

Some other fast-selling mobile products are digital cameras and Internet-audio players, which also contain microcontrollers and DSPs. (’C54x chips are the most popular DSPs in MP3 players, according to TI.) To meet the challenges of these hot growth markets, TI’s new 16-bit fixed-point ’C55x core has significantly better performance, lower power consumption, and greater code density than the ’C54x.

Figure 4 is a block diagram of the ’C55x core, with new features highlighted in purple. Whereas the old ’C54x is a relatively simple DSP with a single ALU, one MAC unit, a barrel shifter, two address-generation units, and two 40-bit accumulators, the new ’C55x is a much more powerful device; two ALUs, dual MAC units, three address-generation units, four 40-bit accumulators, four new 16-bit data registers, a primary instruction cache (size dependent on implementation), a 128-entry instruction buffer, a third 16-bit data-input bus, a wider program bus (32 bits instead of 16 bits), a wider address bus (24 bits instead of 16 bits), a power-management unit, and a real-time debug unit

Two of those new features—the wider address bus and the dual 17-bit MAC units—are especially important for wireless phones. The 16-bit address bus on existing ’C54x DSPs can access only 64K of memory, which is woefully inadequate for the software requirements of 3G wireless-telephony standards. With a 24-bit bus, the ’C55x can address a much roomier 16M. Likewise, the 17-bit multipliers in the dual MAC units are crucial for noise-reduction and compression algorithms in vocoders that need an extra bit of precision when multiplying signed and unsigned values.

Such a vast expansion of resources might seem to conflict with TI’s goal of reducing power consumption. But TI says ’C55x-based chips will consume less power than DSPs built on the ’C54x core, despite their higher performance. Some ’C55x DSPs will have 1.5V cores, the same as the latest ’C54x DSPs, but others will have 0.9V cores, which helps to reduce power consumption to 0.05mW per mips or 0.10mW per megahertz (not counting on-chip memories and peripherals), according to TI.

Lower voltage isn’t the only way the ’C55x conserves power. Two other methods are automatic power management and user-programmable “idle domains.” The new power manager automatically switches on-chip memory arrays and peripherals into a low-power mode when they’re not needed, then wakes them up when they’re accessed. Neither operation adds latency to program execution, and the operations are completely transparent to programmers.

The second power-conservation technique gives programmers control over which parts of a chip are idle or fully active at any moment. There are six idle domains: the CPU, the DMA, on-chip peripherals, the external memory interface, the instruction cache, and the clock-generation circuitry. Programmers can idle any of those blocks or nearly any combination of them.

In addition to those measures, the wider program bus and the new instruction buffer and primary cache help reduce bus activity, which saves a little more power. The instruction buffer is like a “Level 0” cache—it stores tight loops and other basic blocks of code that might be flushed out of the instruction cache by a subroutine call.

Taking Advantage of CISC

To boost performance and code density, TI added several dozen new instructions to the ’C55x and modified the instruction format. Table 3 shows some of the most important new instructions. Many of them take advantage of the dual MAC multipliers and ALUs by implementing

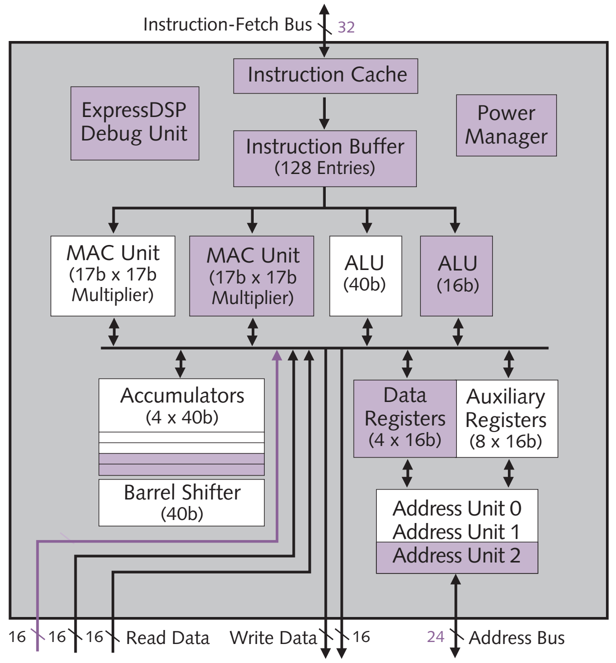

Figure 4. The ’C55x has many more resources than the ’C54x, as indicated here by the blocks and arrows highlighted in purple. Note the extra data-input bus, the wider program bus, and the wider address bus.

Instruction	Description	Sample Applications
ADDSUBCC	Conditional add/subtract	Viterbi butterfly
MAXDIFF	Compare & select extremum	Viterbi butterfly
SFTCC	Conditional shift	General DSP
ADDSUB	Dual 16b add/subtract	Viterbi butterfly
SUBADD	Dual 16b subtract/add	Viterbi butterfly
MAC I MAC	Dual multiply-accumulate	Block FIR, autocorrelation
FIRSADD	Symmetric/Asymmetric FIR	Finite infinite response filter
MPYMR I MOV HI	Parallel multiply and store	General DSP
CMPAND	Register comparison	Control code
SWAP	Register swap	Control code

Table 3. Several new instructions take advantage of the dual MAC units and ALUs in the 'C55x to increase parallelism and code density. Due to the new variable-length format, simple instructions require less memory than complex instructions.

parallel arithmetic operations, while others (such as SWAP) allow programmers to replace multiple instructions with a single instruction.

The new instruction format takes a classic CISC approach. Instead of the fixed-length 16-bit instructions used by the 'C54x, the new core has variable-length instructions that can be 8, 16, 24, 32, 40, or 48 bits long. This arrangement allows simple instructions to occupy less memory than complex instructions. Because the width of the 'C55x program bus has been doubled to 32 bits, the processor can fetch as many as four instructions in a single bus cycle—or, in the case of a 40- or 48-bit instruction, fewer than one instruction per cycle.

Another important modification to the instruction format is conditional execution, also called guarded or predicated execution. A new optional instruction prefix tells the processor to execute the instruction only if a particular condition flag in a status register is true. Almost all the instructions in the 'C55x instruction set can use the optional prefix, which is eight bits long and can be omitted if conditional execution is less important than maximum code density. The only instructions that can't use the prefix are RESET and IDLE.

Unfortunately, changing the instruction format has the side effect of making the 'C55x incompatible with existing 'C54x binaries. The new core is source-code compatible with the 'C54x, however, so developers can simply reassemble or recompile their existing programs. In most cases, developers will want to rewrite at least some of their code anyway, to take advantage of the new instructions and resources in the 'C55x.

According to TI, the new instruction format reduces code size by an average of 30%, with some algorithms for GSM wireless phones shrinking by about 50%. Greater code density is really a power-saving feature in embedded systems, because it reduces memory requirements and the number of bus cycles needed to fetch instructions. It also increases the effective sizes of the instruction cache and instruction buffer, thus minimizing references to off-core and off-chip memory.

One disadvantage of variable-length instructions is the extra time required to scan the instruction stream, find the instruction boundaries, and decode the instructions. But this situation doesn't seem to impair the performance of the 'C55x, because the instruction buffer has some extra logic that marks instruction boundaries and predecodes the instructions. TI says the 'C55x is about five times faster than the 'C54x, with some algorithms (such as convolutional encoding) showing a 7.5 times improvement, thanks to greater parallelism. Those comparisons are based on a hypothetical 300MHz 'C55x and a 120MHz 'C54x, but even after normalizing for clock frequency, the 'C55x is faster than its predecessor. The addition of a 16-bit ALU in the 'C55x should speed up control-code and bit-manipulation operations. TI claims the new core is 6 times faster than the old core for those tasks.

Power consumption is equally impressive. TI says a 300MHz 'C55x consumes about 0.05mW per mips at 0.9V, compared with 0.32mW per mips for a 120MHz 'C54x at 1.8V—a sixfold improvement. Decreasing the power consumption by that degree while simultaneously increasing the clock frequency, raw performance, and code density is a major accomplishment.

	C55x	C54x	DSP16000	DSP56600	SC140	ADSP-219x
Feature	TI	TI	Lucent	Motorola	StarCore	ADI
Data Type	16b fixed point	16b fixed point	16b fixed point	16b fixed point	16b fixed point	16b fixed point
Instr Length	8–48b	16b	32b	24–48b	16–48b	24b
Core Frequency	70–400MHz	30–120MHz	100–200MHz	58–104MHz	300MHz	160–300MHz
16 x 16b MACs	2 per cycle [1]	1 per cycle	2 per cycle	1 per cycle	4 per cycle	1 per cycle
MACs/s (max freq)	800 million	120 million	400 million	104 million	1,200 million	300 million
Core Voltage	0.9V[2]	1.8V[3]	2.5V	1.8–2.4V	1.5V	1.2–2.5V[4]
Power/Freq (est)	0.10mW/MHZ	0.32mW/MHz	NA	NA	0.66mW/MHz	0.18–1mW/MHz
Power/Perf (est)	0.05mW/MMACs	0.32mW/MMACs	NA	NA	0.16mW/MMACs	0.18mW/MMACs
IC Process	0.15µ	0.18µ	NA	NA	0.18µ	0.18–0.25µ
Production	2H00	Now	Now	Now	2001	2000

Table 4. The 'C55x easily surpasses TI's popular 'C54x and is highly competitive with DSP cores from Lucent, Motorola, ADI, and the StarCore alliance. ([1]The dual MAC units in the 'C55x operate on 17-bit integers. [2]TI will offer versions of 'C55x DSPs with 1.5V and 0.9V cores. [3]Only the fastest versions of 'C54x DSPs have 1.8V cores. [4]The ADSP-219x core voltage is 2.5V for initial parts built on a 0.25-micron IC process and will drop to 1.2V on future 0.18-micron parts. NA: data not available.)

More Competition to Come

Table 4 compares the ’C55x with the ’C54x and DSP cores from ADI, Lucent, Motorola, and StarCore. The ’C55x holds up well in terms of performance and power consumption, and its variable-length instructions probably give it an edge in code density too, because the other DSPs have longer instruction formats.

An important performance comparison is the number of MACs these DSPs can execute per cycle or per second. MACs are critical for the data-intensive algorithms used in wireless telephony. Among the low-power DSPs listed in Table 4, only the StarCore SC140 (four MACs per cycle and 1.2 billion MACs/s) exceeds the ’C55x in this regard. The SC140, which will appear in future DSPs from StarCore partners Lucent and Motorola, is extraordinary in that it is fast enough to compete against high-performance DSPs like TI’s ’C64x, yet is so power-stingy that it’s not out of place in the company of DSPs designed for low-power applications, like TI’s ’C55x. However, the low-end TI core still consumes less power than the SC140.

StarCore could compete more effectively with TI by introducing a budget version of the lavishly appointed SC140. By removing half of the MAC units, ALUs, and bitfield units, StarCore could offer a DSP that’s still competitive with the ’C55x in terms of performance but consumes much less power and silicon than the SC140. That action would also give StarCore a pair of source-code-compatible DSP cores that span the range from low power to high performance—a feat TI cannot match with the incompatible ’C55x and ’C64x.

Conspicuously missing from Table 4 is another new DSP core that’s almost certain to alter the balance of power: the outcome of ADI’s strategic alliance with Intel. The companies have not yet revealed any technical details about their new core. Their first volume shipments will have to begin in 2001 or 2002 to catch the first wave of 3G wireless phones.

At this point we can only guess how competitive the ADI/Intel core will be. Even if it compares favorably with the ’C55x on performance, power consumption, and code density, TI’s core still has some advantages—mature tools, compatibility with existing source code, and hundreds of customers who are already using the ’C54x. If the ADI/Intel DSP is a completely new architecture, it will need a few years to gain a foothold in the marketplace.

TI appears to be fully aware of the new threats. The company has been focusing more attention on DSPs lately, girding itself for stiffer competition. TI is investing more heavily in development tools and is broadening its line of complementary analog parts. By rolling out two new DSP cores that address both ends of the market, TI is making a strong effort to defend its turf against old rivals and newcomers alike. ◇

Price & Availability

TI plans to sample the first ’C64x-based DSPs this summer at 600–800MHz. Production is scheduled to begin early in 2001. The first ’C55x DSPs will sample this spring. They are scheduled to enter production in 2H00 at 200MHz and ramp to 400MHz in 1H01. TI has not yet announced prices. For more information, go to *www.ti.com/sc/*.