

SHARC Bites Back

The Memory Inside: TigerSHARC Swallows Its DRAM

Extended-range smart missiles, guidance systems and long-range radars need increased accuracy that's only available by more intensive number crunching and signal processing. One DSP vendor, Analog Devices, has solved the problem by embedding large amounts of DRAM on-chip, along with other elegant architectural tricks.

For years, DSP applications have relied on multiple processors working together to execute computationally intensive real-time mathematical functions. Whether the application involved the military, video, communications, medical imaging or industrial instrumentation, DSP systems have need-

ed rapid deterministic execution speeds and fast I/O.

In recent military applications, reliance on signal processing capabilities has taken a sharp upturn as priorities have been redirected to a key goal: minimizing collateral damage to military personnel and to inappropriate targets by increasing

range and accuracy. Military applications demand missile and defense systems with ranges of two to three times today's systems and much more ground-level detail. Distance to targets beyond the line of sight are measured in range gates and traditionally, the number of range gates that could be stored was limited by the lack of internal memory. By extending distance, systems must be able to process a much greater number of range gates. This extension pushes the signal processing envelope well beyond the capabilities of previous generation DSPs.

To gain this new level of performance, DSP vendors are being asked to provide signal processors with on-chip memory capable of handling datasets four to six times larger than current applications, while continuously processing in hard real-time complex mathematical operations such as Fast Fourier Transforms (FFT).

Although clock speed and increased bandwidth offer some of this performance boost, Analog Devices determined that on-chip

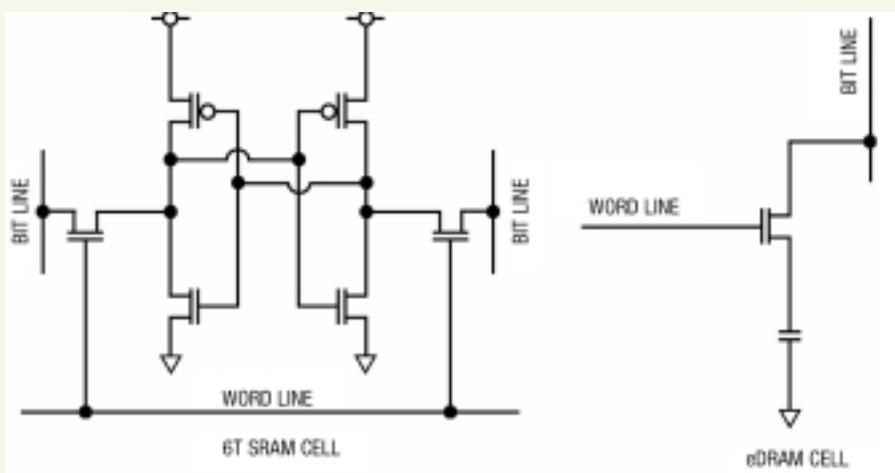


Figure 1

An SRAM cell requires six times more transistors than a DRAM cell. The result: much higher power consumption. A 16 Mbit SRAM consumes 1 A while a 16 Mbit DRAM uses only 10 microA of power.

memory was the only way to gain the additional magnitude of number-crunching capabilities. In the migration from the former ADSP-TS101 to the ADSP-TS2xx TigerSHARC family, ADI evaluated several memory-boosting options before settling on increasing the on-chip memory by using embedded DRAM (see sidebar "What's in the Chip?" for ADSP-TS-20x architectural details).

But accomplishing the goal of more on-chip memory was no easy task. DRAM requires refresh circuitry, and is much slower than SRAM. Nonetheless, a sequence of technical hurdles was crossed using multiple buses, smart buffered memory caches and clever interleaved memory pages. Collectively, the latest TigerSHARC with embedded DRAM can compute a 64K point, 32-bit complex FFT in just 2.3 ms, while still accessing up to 4 gigawords of off-chip SDRAM.

Upping the Memory Ante

Since 1995, the on-chip memory of floating-point DSPs has maxed out at 4-6 Mbits of fast SRAM. Additional memory has been accessed through external low-cost SDRAM or faster, but higher cost SRAM. The speed of SRAM is ideal for performance, while DRAM offers density and the ability to compute larger point FFTs directly in memory.

To gain increased performance, signal processing vendors are moving to more on-chip memory. In doing this, the vendors avoid the slower clock rates and latency issues that come with transferring data off-chip to on-chip. In moving to on-chip memory, developers are also able to reduce the number of potential failure points and avoid the real estate requirements associated with external memory.

Upping the amount of on-chip SRAM, however, has significant drawbacks. To significantly increase the

amount of memory via SRAM, the die size of the signal processor would increase by four to five times—an intolerable real estate requirement for almost all embedded applications.

With the on-chip SRAM running at core clock speed and using six transistors per cell of memory, the

amount of power consumption in a large memory application becomes significantly more than the 2.5 W currently required by the TS201 with its embedded DRAM. To curb excessive power consumption, traditional large memory DSPs incorporate on-chip hierarchical memory.

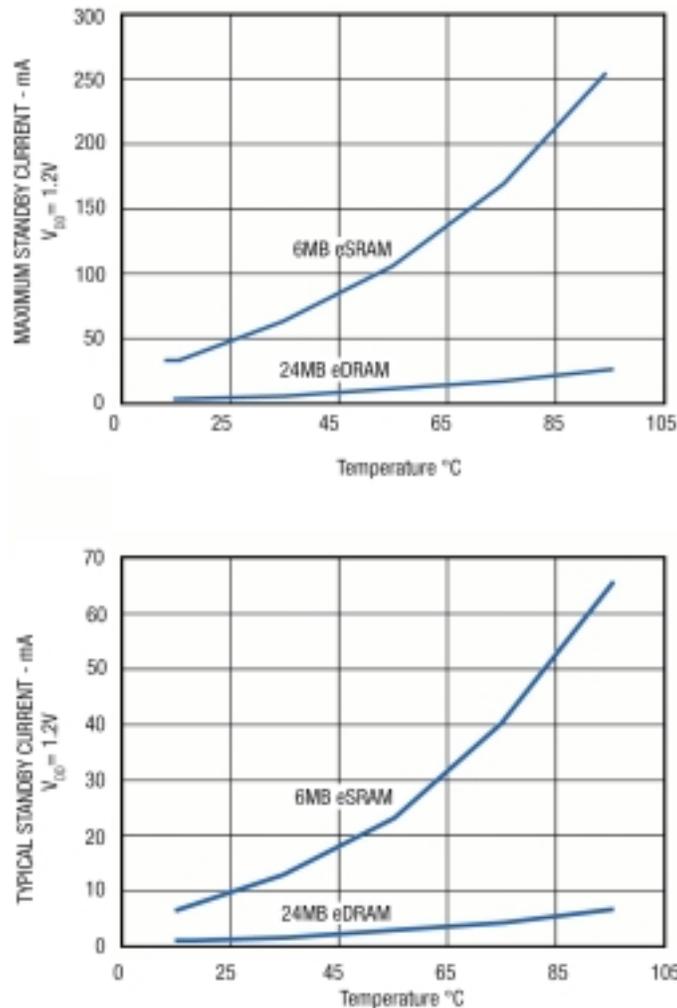


Figure 2

As temperature increases, the power consumption of SRAM becomes exponentially higher than that of DRAM. Compared here are the maximum and typical standby current of 6 Mbits of SRAM and 24 Mbits of DRAM. Notably, four times more DRAM consumes much less power.

Special Feature SHARC Bites Back

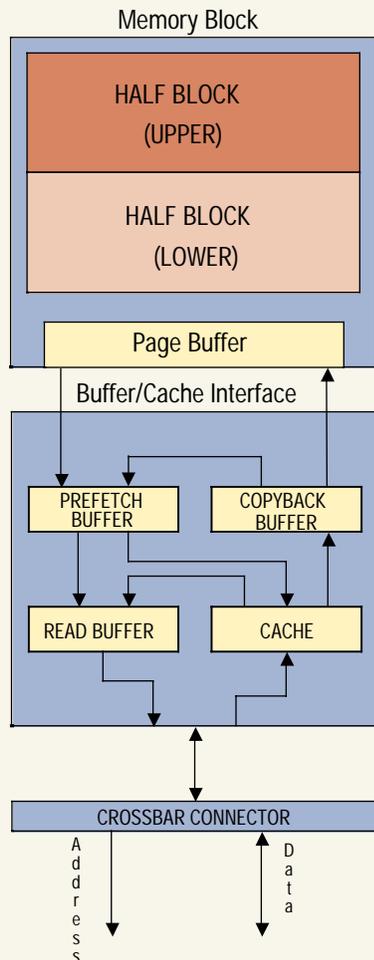


Figure 3

Data is transferred in and out of the embedded DRAM via a pseudo-SRAM buffer/cache interface. Six 4 Mbit memory blocks comprise the 24 Mbits of embedded DRAM.

With this architectural setup, some of the memory runs at the processor's speed while other memory is throttled back, reducing power consumption. The trouble is, to restrict power consumption even further with this methodology, so much of the memory would be running at a lower speed that the benefits gained from additional memory would be lost. The processor would not have fast enough memory resources to meet the

demands of extended range signal processing applications.

As well, large amounts of integrated memory are vulnerable to soft error caused by alpha particles and cosmic rays that can flip bits. The soft error rate (SER) for SRAM is acceptable at about 6-8 Mbits. For the TS-201, this results in an SER of 60 FITs (1 FIT represents a failure every 114,000 years). In a worst case scenario where the amount of SRAM was increased to 24 Mbits and sped up, this same system could potentially incur 12,000 to 36,000 FITs, well over the industry's specified maximum of 500 FITs.

In short, with die size, BER and power constraints, large SRAM arrays are clearly not an optimal way to increase DSP performance. Analog Devices instead chose a clever embedded DRAM approach that balances the advantages of DRAM density with SRAM speed.

Embedded DRAM Alternative

In a digital signal processor, the biggest factor favoring the move from embedded SRAM to DRAM is the memory density. The die size in the older TS101 is 100 mm², approximately the same as that of the TS201, even though the TS201 offers four times more memory (24 Mbits vs. 6 Mbits).

Stated in measurable terms, the TS101's 6 Mbits of SRAM uses 4.49 mm²/Mbit, which means that 27 mm² are devoted to SRAM, a startling 27% of the entire die space. In a 24 Mbit chip, the amount of SRAM die area would be 162 mm², resulting in an overall chip size of 240 mm². In the newer TS201, the DRAM requires 0.915 mm²/Mbit, which results in 22 mm². Overall, embedded DRAM drops the percentage of memory space from 27% to 22% of die size.

Not surprisingly, the smaller amount of die area required for the

TS201's 24 Mbits leads to a significantly smaller overall footprint for the TS201 and represents a substantial cost savings when compared to the same amount of on-chip SRAM. Incidentally, the 24 Mbits total array was chosen for several reasons: it allows whole number increments of 3G cellular WCDMA frames to reside in on-board memory, large FFTs such as 64K points can be executed out of memory, and the proven IP used to create the embedded DRAM was available in 4 Mbit blocks.

Reduced Power Consumption

By embedding DRAM instead of SRAM, the overall transistor count is reduced (Figure 1) since the same quantity of embedded DRAM requires only one transistor per cell (bit line) whereas the SRAM employs six transistors per cell. More transistors means more real estate and current consumption.

As a result, 16 Mbits of SRAM consumes hundreds of milli A while 16 Mbits of DRAM uses only hundreds of microA—three orders of magnitude less. This consumption is further reduced when embedded DRAM's zero-leakage current is compared to that of SRAM over a range of temperatures (Figure 2).

Meeting the Speed Challenge

But DRAM's greatest drawback is its speed and the refresh circuitry required to make it work. Typically, DRAM operates at half the speed of the CPU core, and the complexity of dealing with this proves a significant obstacle—even though DRAM reduces cost, die size and power consumption. To overcome this latency, several steps were taken in the TigerSHARC processor.

To compensate for the slower DRAM speed, the memory was

interleaved and the pipes fattened. The memory allows up to four independent, parallel and simultaneous accesses. The 24 Mbits of total memory is made up of six DRAM arrays (six “blocks” of 4 Mbits), each with four sub-array “banks” that implement a 256-bit wide double-pumped data interface. This doubles the bus size into memory and has the effect of essentially doubling the access speed into memory (refer to sidebar Figure 1 for more details).

Four 128-bit memory buses (J-ALU, K-ALU, program sequencer and SoC interface) provide high internal bandwidth and nonintrusive data transfer between the core and the memory. The SoC bus offers more DMA operations to memory. All of the buses support the core’s Harvard architecture and run at full chip clock rate of 600 MHz, each transferring 9.6 Gbytes of data per second.

In addition, each of the six memory array segments includes a 128 Kbit SRAM unified cache array that operates as a 128-bit wide pseudo-SRAM running at the same speed as the core. The cache functions at full core clock frequency of 600 MHz ensuring access to the memory without latency. Through cache prefetching, data can be read or written at core speed despite the DRAM’s slower rate. The 24 Mbits of overall memory subsystem supports over 57 Gbytes/s sustained throughput *without stalls* on sequential accesses through the whole memory.

Organizing the Memory

Getting the data to memory is only part of the problem. To work around the fact that the embedded DRAM functions at half the core speed, the memory is organized in “strides” of 256 bits (double-

pumped 128 bits). To capitalize on this increased bus width, a buffer/cache interface is introduced between the embedded DRAM and the crossbar that contains one prefetch buffer, two read buffers, one cache and two copyback buffers (Figure 3). The buffer/cache interface optimizes the access between the memory and the buses.

Interestingly, the four memory buses do not have direct access to the embedded DRAM. Access is only available through the crossbar to the buffer/cache interface described above. Because of this, even though the embedded DRAM functions at half core frequency, the data is accessed at full core speed.

The embedded DRAM is organized in memory pages and access to memory is through the page buffer. To access a memory page, it is opened through a process called “memory page activation”—a task that takes two core clock cycles. Its contents are passed into the page buffer in the pseudo-SRAM buffer cache, which can be accessed directly. To access another memory page of the same sub-array, the already opened page must be closed and the new memory page activated.

The prefetch buffer is organized as four prefetch pages, each with 2048 bits (eight 256 bit words) with two prefetch pages allocated to a memory half-block (half of 4 Mbits, or 2 Mbits). The prefetch buffer handles read transactions, providing intermediate storage for read data before the data is transferred to the read buffer and cache.

To save clock cycles, the prefetch buffer anticipates sequential read transactions and queues data from the embedded DRAM prior to the actual transaction request. As well, it also acts as a cache, matching a read transaction with one of the page tags

Inside the Cache

The cache of each memory block serves as a unified data/instruction storage cache. It temporarily stores the data read from the DRAM in order to avoid multiple DRAM accesses to the same data. Each word has its own valid bit to allow read and write transactions using 32-bit word granularity. The cache write policy is to copyback and does not directly write allocate.

The copyback buffer contains two separate 4096-bit buffers, one for each half block of embedded DRAM, which can accommodate sixteen 256-bit entries. The copyback buffer serves as an intermediate holding location for replaced cache lines that must eventually be written back to the embedded DRAM.

The copyback buffer is the only interface to the embedded DRAM for write transactions. When the cache is disabled, the copyback buffer provides a bridge between the crossbar interface and the embedded DRAM for write transactions.

without a prefetch buffer hit. This mechanism enables the embedded DRAM to accommodate sustained sequential read transactions at maximum processor core throughput without incurring cycle penalties, even though the DRAM functions at half the core frequency.

The read buffer is organized as two 256-bit words, one for each half block of memory, and is used for read transactions only. Each read transaction transfers a 256-bit word from the cache or prefetch buffer to the read buffer. From here, 32, 64 or 128 bits are extracted to service the transaction request. When new data enters the read buffer from the prefetch buffer, the data also enters the cache. When a

Special Feature SHARC Bites Back

What's in the Chip?

The next-generation TigerSHARC processor ups the ante on a number of features (Table 1), with the ADSP-TS201 achieving 600 MHz, 24 Mbits of embedded DRAM, a 14-channel zero-overhead DMA engine and I/O processing capable of an aggregate throughput of 4 Gbytes/s (Figure 1). The TigerSHARC uses Very Long Instruction Word (VLIW) load/store architecture and executes as many as four instructions per cycle via an interlocking, ten-stage pipeline and dual computational blocks.

Four 128-bit memory buses (J-ALU, K-ALU, program sequencer and SoC interface) provide high internal bandwidth and non-intrusive data transfer between the core and 24 Mbits of memory. The memory, organized into six segment "blocks" (M0 through M5), is made up of six DRAM arrays each with four sub-array banks that implement a 256-bit wide data interface and 128 Kbit SRAM unified cache array (Figure 2).

Two types of integrated multiprocessing support (four link ports and a cluster bus) enable glueless scalability of up to eight TigerSHARC processors and the addition of other system requirements such as external memory. While other processors require bridge chips for multiprocessor applications, TigerSHARC can manage the dataflow directly with built-in arbitration that determines priorities, obviating the need for bridge chips and minimizing board real estate. And, with each link port capable of 500 Mbytes/s duplex speed, the high-speed streaming data rate between devices totals in aggregate 4 Gbytes/s at 32/64-bits.

Finally, the cluster bus enables logical global memory sharing. Up to 4 Gigawords of external SDRAM can be addressed with the TS201. In a multiprocessor application, the cluster bus enables inter-processor communication and data memory sharing without bridge chips since the arbitration is built in.

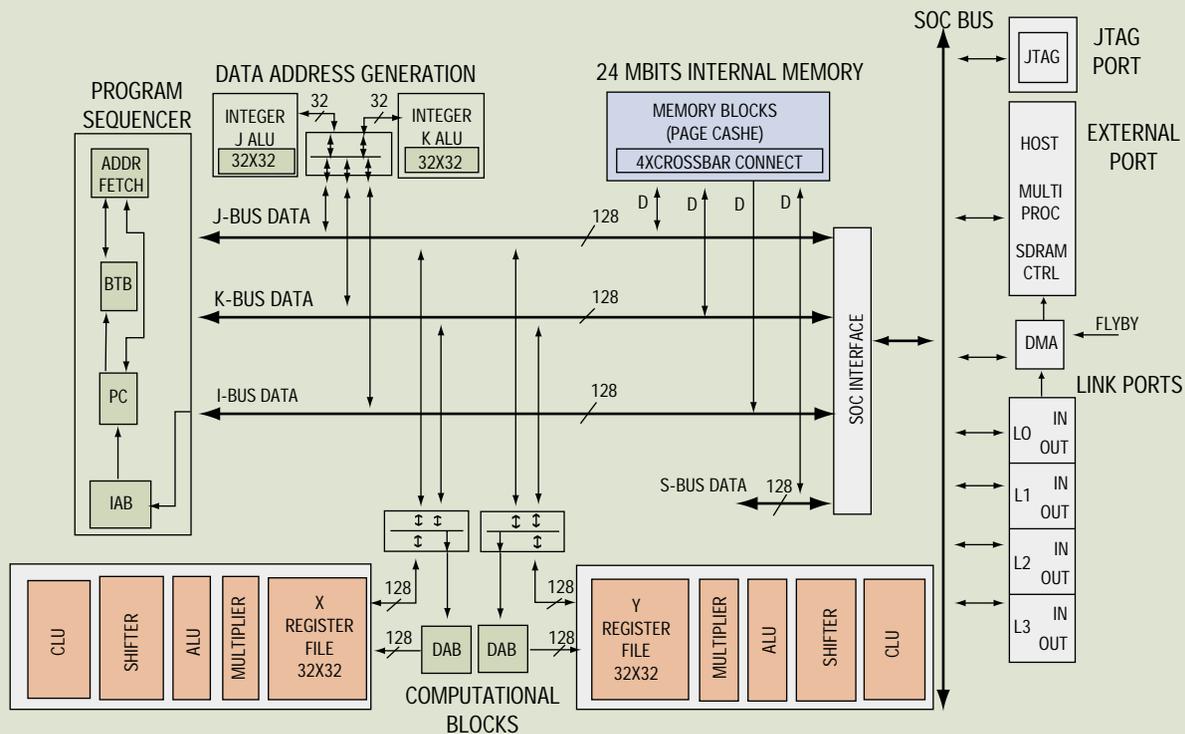


Figure 1

The ADSP-TS201 processor brings an unprecedented amount of on-chip memory to signal processing applications with 24 Mbits of DRAM memory, organized into six memory segments. Four 128-bit buses ensure a high internal bandwidth and nonintrusive data transfer.

Special Feature SHARC Bits Back

	TS201	TS201	TS202	TS203	TS101
Speed	600 MHz	500 MHz	500 MHz	500 MHz	250 MHz
Onchip Memory	24 Mbits	24 Mbits	12 Mbits	4 Mbits	6 Mbits (SRAM)
I/O	4 GB	4 GB	4 GB	500 MB	1 GB
CLU	Yes	Yes	No	No	Yes
Peak Floating Point Performance/Second	3.6 billion operations	3.0 billion operations	3.0 billion operations	3.0 billion operations	1.5 billion operations
Peak Floating Point Multiplies/Second	1.2 billion	1.0 billion	1.0 billion	1.0 billion	500 million
Peak Integer MACs/second	4.8 billion	4 billion	4 billion	4 billion	2 billion
External Bus Width/Speed	64/32 bits @ 125 MHz	64/32 bits @ 125 MHz	64/32 bits @ 125 MHz	32 bits @ 125 MHz	64/32 bits @ 100 MHz
Link Ports & Throughput	4 @ 3.6 GBps, LVDS	4 @ 4 GBps, LVDS	4 @ 4 GBps, LVDS	2 @ 1 GBps, LVDS	4 @ 1 GBps
Bandwidth to Processing Ratio	1B/FLOP	1.33 B/FLOP	1.33B/FLOP	0.5 B/FLOP	1.2 B/FLOP
MFLOP/Watt	1000	1402	1402	1402	1111
1024-Point Complex Floating Point FFT	16 usec	20.1 usec	20.1 usec	20.1 usec	39 usec

Table 1

The ADSP-TS2xx family introduces performance and features, delivering gigaFLOPS of signal processing muscle.

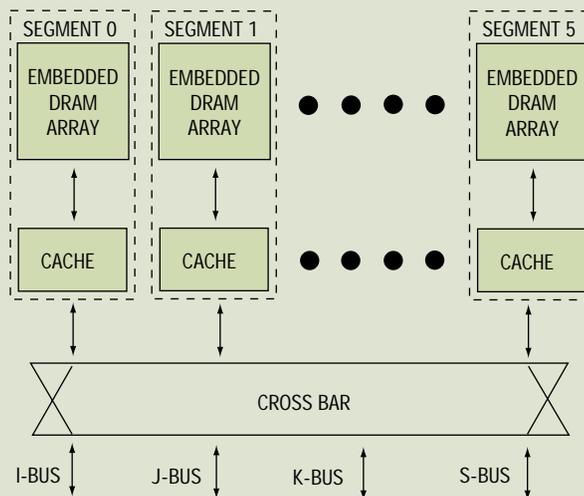


Figure 2

For continuous real-time processing, the processor speed must match memory speed. Analog Devices boosts memory speed by designing into each of six memory array blocks a 128-bit cache, enabling each segment (block) to operate like a 128-bit wide pseudo-static RAM.

Special Feature

SHARC Bites Back

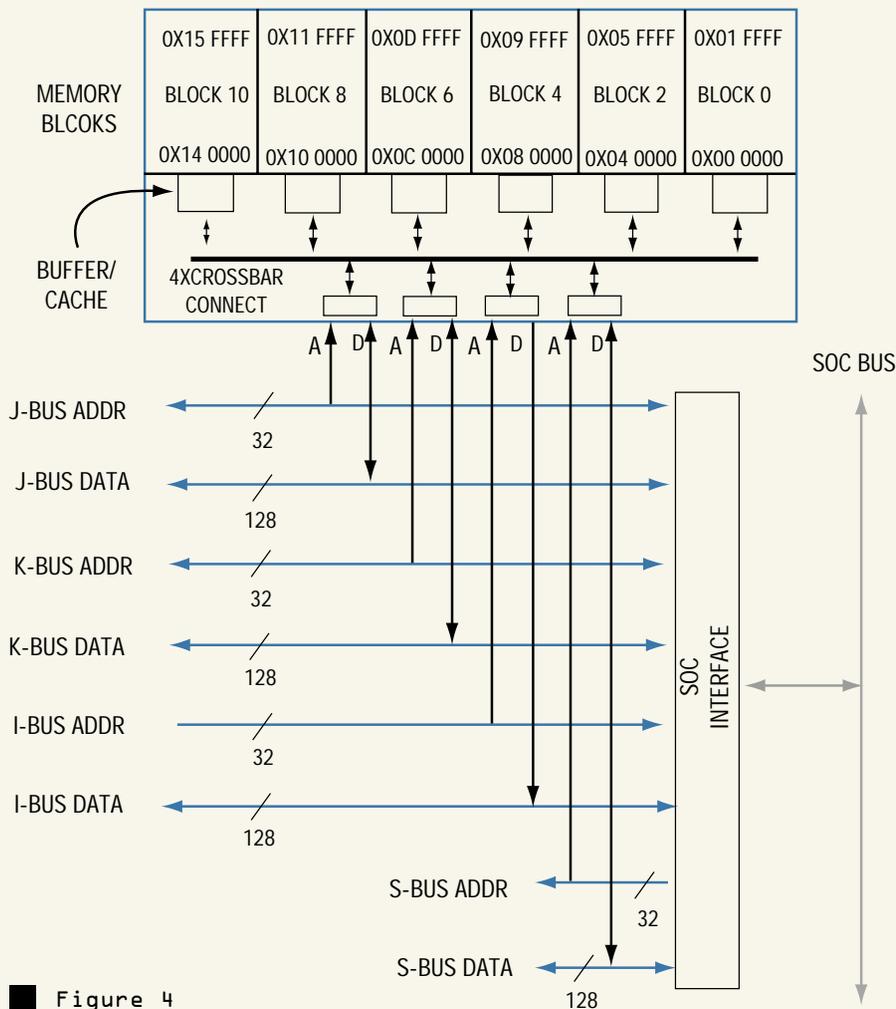


Figure 4

Block processing of 128K-point complex FFT requires six memory blocks, but a 64K-point complex 32-bit FFT can be executed entirely on-chip in 2.3 ms (at 600 MHz).

read transaction matches the tag of a read buffer entry, the transaction occurs with no penalty.

A snooping process is employed in the copyback buffer in the case of a cache miss or when the cache is disabled. The prefetch buffer observes the content of the copyback buffer and if the data exists in the copyback buffer, the data is taken directly from it. Through snooping, the TigerSHARC is able to deliver predictable performance because a cache hit involves no cycle penalty, although a miss results in a loss of one to eight cycles.

Getting Around Refresh

To prevent data loss through leakage, the embedded DRAM cells (bits) must be refreshed every 32 ms. In this process, data is automatically read and written back through the refresh rate select instruction.

The processor treats the refresh process as another peripheral, controlling when the refresh happens such that it is fully predictable and synchronous. To ensure that the refresh happens, the processor may block access to itself from external events, but the memory is still available 98% of the time.

Securing the Next Beachhead

As military applications extend the range—and more importantly, *accuracy*—of missile and defense systems, the need for RADAR systems that can identify meaningful targets with greater granularity is critical. One way of extending these capabilities is through hardware capable of delivering a much larger number of Fast Fourier Transforms (FFT) per second. With the increased memory and the fast I/O transfer via the link ports, the TigerSHARC can process 50,000 1K-point complex (32-bit) FFTs per second, and 128K-point complex FFTs in on-board memory at 600 MHz, a level of performance promising significant breakthroughs in RADAR technology.

With other processors, processing these large point FFTs often requires much more memory and is therefore typically abandoned. However, the separate 4 Mbit blocks of the TigerSHARC processor facilitate continuous processing of a large point size. To do 64K or 128K complex 32-bit floating point FFTs, the input and output data must be double buffered in order to output results and collect new samples while computing the previous result.

Basically, the transaction involves block data processing, which is more efficient, but requires more memory. In a 64K or 128K point FFT, two of the six 4 Mbit memory blocks are used to create overlapping input data files. Two blocks are used for the output, in which one segment performs the computation and stores the input data impulse response while the other stores the twiddle table and data pointers. The TS201 can do this because of its large on-chip embedded memory array, and each memory block is capable of containing 64K samples of complex 32-bit floating point data.

Specifically, Block 0 is used for code and the frequency response of the impulse. Block 1 holds the twiddle factors and miscellaneous data. Block 2 manages the first input buffer while Block 3 provides the first output buffer and Block 4 the second input buffer. Block 5 takes care of the second output buffer (Figure 4). With the six block memory architecture, the data for a 64 x 1024 complex 32-bit floating FFT points fits in one 4 Mbit block, and at 600 MHz the entire transform is completed in approximately 2.3 ms.

Embedded DRAM Helps Signal Processors Enter a New Space

With the dramatic increase in memory size that embedded DRAM offers, signal processors have moved the design community closer to addressing the needs of applications requiring dozens of gigaflops of signal processing muscle—applications once the sole domain of expensive ASICs and power-hungry FPGAs.

Although quadrupling the amount of on-chip memory has required architectural innovation to overcome DRAM's speed and refresh issues, the bandwidth increase and cache interface has compensated to the extent that the embedded DRAM

on the TigerSHARC runs at only 3% less than full clock speed.

With this speed problem solved, embedded DRAM delivers superior performance density, setting a new standard in the amount of performance per watt, per square inch of board or per dollar. Applied to the TigerSHARC TS201, embedded DRAM delivers 3600 MFLOPS per square inch, 1800 MFLOPS per Watt and 75 MFLOPS per dollar—a notable leap in signal processing technology. ■■

Analog Devices
Norwood, MA.
(781) 329-4700.
[www.analog.com].