ELECTRONIC DESIGN EXCLUSIVE

Cascadable DSP chip attacks tasks with parallel punches

Tony King-Smith and Hossein Yassaie
Inmos Ltd., 1000 Aztec West, Almondsbury, Bristol, BS12 4SQ England; (+44) 454-616616.

A 16-bit digital transversal filter cracks the bottleneck that bogs down other single and multichip solutions. It cascades to handle most tasks.

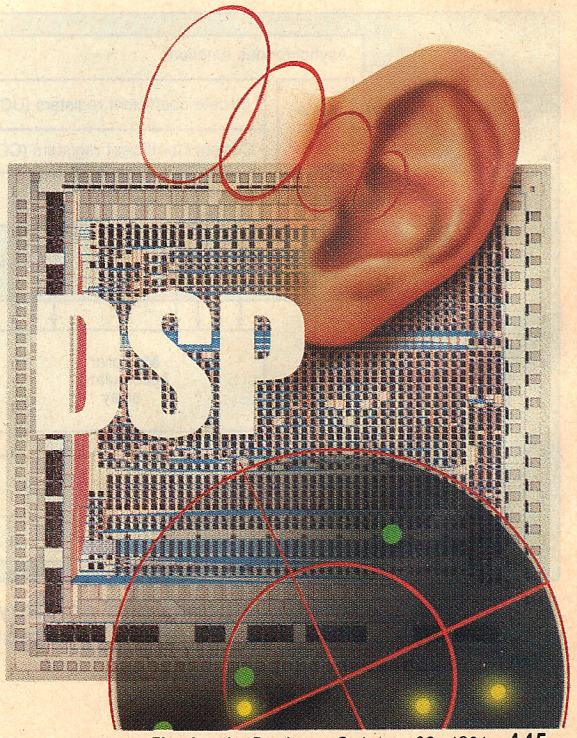
The recent spate of digital signal-processing chips has yielded only two approaches to their task. One approach stresses high throughput and requires multiple chips for the job, drawing on very fast multipliers, multiplier-accumulators (MACs), and microprogrammed bit-slice machines. The alternative approach is relatively slow, but combines in one chip all the elements of a general-purpose DSP system. Usually, the chip's instruction set aims for the highest possible memory bandwidth in order to make maximum use of the multiplier-accumulator, a crucial player in executing most DSP algorithms.

Both approaches are general-purpose solutions, and each has pitfalls. High complexity haunts the bit-slice designs, while low data rates plague the one-chip choices. Furthermore, neither averts a classic bottleneck at the multiplier-accumulator, which stems from the fact that it typically requires three memory operations: Two to read and one to write. As a result, a processor's speed is ultimately determined by the memory's cycle time (assuming all memory addresses must be accessed). Neither multiple buses nor even an infinitely fast multiplier-accumulator can entirely solve the dilemma.

By giving designers the resources to perform many operations in parallel, the IMS A100 cascadable signal-processor chip opens a route around the multiplier-accumulator bottleneck. The chip takes advantage of concurrent operations possible in running many DSP algorithms. Specifically, the CMOS chip is a 32-stage transversal filter, able to repeatedly multiply and accumulate values. It is organized so that each node, or tap, within the data path contains a multiplier-accumulator and some local storage.

Although such a filter might seem suitable only for finite impulse response (FIR) applications, its transversal structure can handle most common DSP algorithms, as well as general operations such as matrix multiplication and encryption (see "Transversal Filter Basics", p. 148). The combination of a reasonably powerful microprocessor and one or more of the filter chips results in a system that has a throughput previously possible only with bit-slice architectures, but at a fraction of the size, cost and complexity. Moreover, a 1.5-μm CMOS process keeps power drain under 1 W when pulling in data at 10 MHz.

The chip supplies a highly flexible building block for a wide range of signal processing tasks. Multiple devices can be cascaded without degrading throughput, making possible a transversal filter of several thousand stages. DSP techniques can then be applied to a wide range of communication applications, such as fast echo cancelling in satellite links, as well as radar, sonar, and image processing,



applications that demand performance orders of magnitude greater than DSP for modems and speech synthesis.

With one or more circuits, a designer can make many one- and two-dimensional filters. Users can also build systems that perform high-speed correlations; convolutions; or, by applying prime-radix and decomposition techniques, Fourier transforms. For example, two chips controlled by a general-purpose host can perform a 1000-point complex discrete Fourier transform (DFT) in less than 10 ms—a speed that matches the best bipolar logic boards. Moreover, performance improves almost in proportion to the number of filter chips that bear on a DSP algorithm.

Data and coefficients are represented in the A100 as two's complement numbers. It takes in 16-bit-wide data words as well as constants, or coefficients, whose widths can be programmed as either 4, 8, 12 or 16 bits. With 16-bit coefficients and 16-bit data words, the unit delivers continuous data rates of up to 2.5 Msamples per second. With the coefficient reduced to 4 bits, the throughput soars to 10 Msamples/s, or effectively 80 to 320 million operations per second (MOPS). For a single device, that speed corresponds to between 3 and 12 ns per tap for an FIR filter—several times faster than most other FIR filter chips.

To reach those speeds, the circuit exploits high-speed RAM structures and advanced microprocessor architectures combining the equivalent of 32 16-by-16-bit multiplier-accumulators, two sets of coefficient registers (a total of 64), assorted control and I/O registers, a barrel shifter, and control logic (Fig. 1).

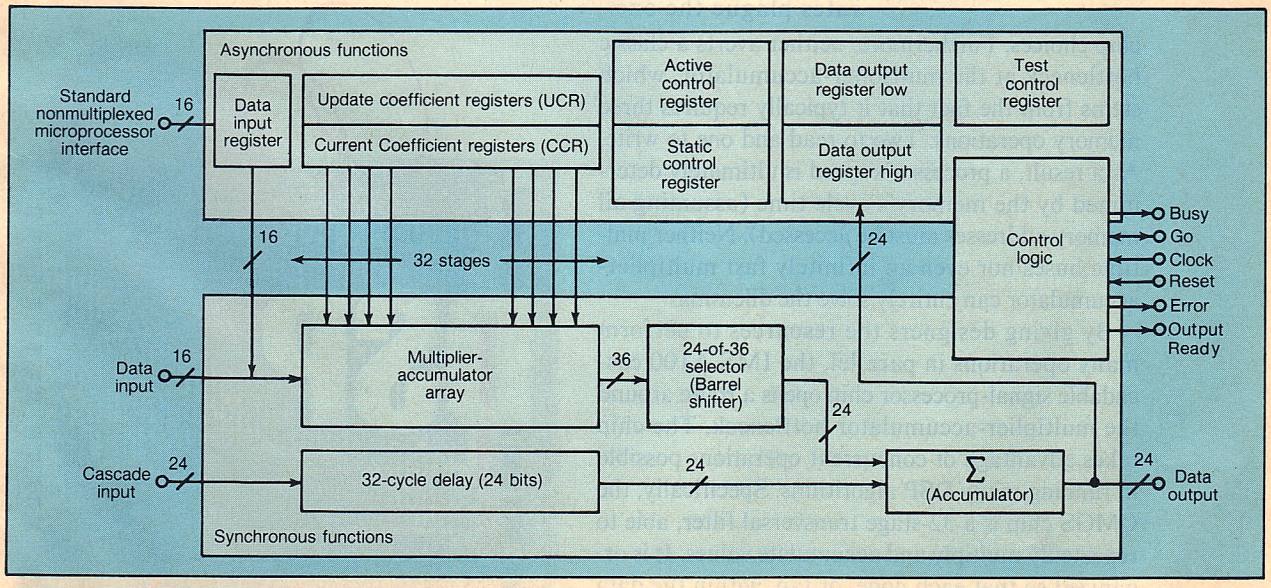
The dual coefficient registers are especially useful, holding present and updated values to let the user alter or load in new values while the remaining circuit processes data. They also allow the user to rapidly switch between different application programs or easily respond to adaptive algorithms. Each coefficient can be updated in less than 100 ns asynchronously to the system clock.

Once a new set of coefficients has been loaded, all 32 coefficients can be interchanged with the previous set without interrupting the input data flow. That exchange is done by a bank swap operation. The swap feature can be automatic, permitting the two register sets to alternately feed their coefficients to the array of multiplier-accumulators for each data input, a handy feature for computing complex numbers.

MINIMIZING I/O BANDWIDTH

The chip's coefficient registers, in conjunction with its delay-and-add chain, also cut I/O bandwidth requirements. For example, a 32-tap FIR filter with a conventional multiplier-accumulator operating on one data sample would do 32 read operations for the coefficients, another 32 reads for data, and a write cycle to deliver the result.

As a result, even if separate buses are used for the address and data, and the write cycle is overlapped with the next read cycle, the process requires a minimum of 32 memory cycles for every data sample. The chip cuts the number of operations to just one read and one write cycle per sample. Still more dramatic results are derived from two or more chips, since no matter how many stages are built, the I/O requirement holds to one read and one write cycle per data sample. In contrast, a multiplier-accumulator-based solution's I/O bandwidth, at best, in-



1. Dual coefficient register banks and an array of 32 16-by-16-bit multiplier-accumulators gives the IMS A100 transversal filter chip the ability to compute complex numbers and implement filters with speeds of 3 to 12 ns for each tap.

creases linearly with the number of stages.

Another advantage of the A100 concerns long transversal filters. Programmers can automatically adjust the word growth of the accumulated result after each 32 stages of evaluation. The result is scaled by the chip's barrel shifter, located near the output port. To preserve accuracy, the partial products are not truncated or rounded in the multiplier-accumulator array. The array output can grow up to a width (precision) of 36 bits, sufficient to guarantee that no overflow occurs. The barrel shifter sends to the bus one of four possible 24-bit fields (starting at bits 7, 11, 15, or 20) within the 36-bit result. Those bits are correctly rounded and sign-extended.

Multiplier-accumulator-based systems, in contrast, would have to periodically read the accumulator, and then shift and round the result to avoid exceeding the accumulator's maximum word size. In a 1024-stage transversal filter with 16-bit data and coefficients, for example, results can be as wide as 42 bits, exceeding the capacity of most single multiplier-accumulators. That width would make it hard to control the logic that surrounds the multiplier-accumulator, as well as add memory cycles.

MICROPROCESSOR INTERFACE EASES DESIGN

To a host and programmer, the A100 looks like a memory-mapped peripheral, specifically a 128-word-by-16-bit block of fast static RAM (Fig. 2). The bottom 64 words of this address space can hold two complete sets of coefficients, with the remaining locations mapping the control and status registers, and the data input and output registers. Only half of the 16 bits in the static control register are used. Two select coefficient size, two pick the output range from the shifter, and four handle operations like selecting the coefficient registers, the source of incoming data, and the master or slave mode.

The chip's standard nonmultiplexed interface easily ties it to any microprocessor or memory-bus-based architecture. Coefficients and status registers are accessed through the interface, which can also carry data to the chip. Maximum performance, however, requires use of the chip's dedicated input and output data ports. In addition to the basic 16-bit interface, control signals provide interrupt requests to the host when an error is detected, and synchronize data transfers between other components.

Input data comes through the 16-bit microprocessor bus interface via the data input register (DIR), the 16-bit data input port, or from the multiplexed 12-bit cascade input port, which is internally demultiplexed to 24 bits. Data leaving the chip can supply a 24-bit result directly through the output accumulator over the multiplexed 12-bit data output port or through the two 16-bit high and low data output registers, DOH and DOL.

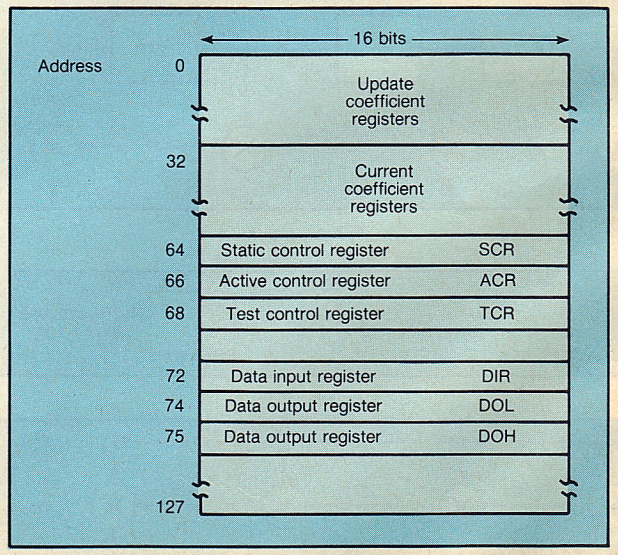
The two registers can send their contents to the host over the interface port. DOL contains the least-significant 16 bits correctly rounded, while DOH contains the most-significant 8 bits, which are sign-extended to 16 bits. Note that by selecting the appropriate 24 bits from the 36-bit internal result, the designer can arrange matters so that the 16 most-significant bits will always appear, correctly rounded, in the DOL register. This permits the host to read the result in a single memory cycle.

To ease the construction of cascaded systems, the chip has a 32-stage, 24-bit-wide shift register and a 24-bit adder. Moreover, the chip combines the adder with a programmable barrel shifter, enabling it to perform block floating-point operations. Those operations permit several chips to be cascaded without encountering the problem of word growth. The data output of one chip simply connects directly to the cascade input of the next, with the input data reaching all cascaded devices at once. Output from the shift register is added to the output of the programmable barrel shifter to produce the final 24-bit output, which is then multiplexed across a 12-bit wide interface. Without such a scheme, the speed of the cascaded devices would be severely limited by the speed of data transfers from device to device.

Many signal processing algorithms can be mapped onto the filter's architecture. For example, when multiplying two complex numbers represented by $(a+ib)$ and $(x+iy)$, the result becomes:

$$(a+ib) \times (x+iy) = (ax\text{-}by) + i(bx+ay) \ldots\ldots$$

Substituting that into the general equation for a transversal filter and taking advantage of the chip's ability to swap between two sets of coefficients reveals a simple path to complex multiplication: Load the complex coefficients into both sets of registers, then feed the real part of the



2. To the programmer, the A100 looks like a 128-word block of fast memory. The first 64 words of the block hold the two sets of filter coefficients while others store the various control-register bits.

complex input data, followed by its imaginary part, into the transversal filter. In that way the chip can execute most DSP algorithms in their complex form.

For applications in the frequency domain, the chip can run a discrete Fourier transform algorithm, based on Rader's prime radix transform and a decomposition technique that involves index mapping. The array of multiplier-accumulators on the chip overcomes computational drawbacks, paving a new way to perform very-high-performance DFTs. Essentially, the algorithm reduces the DFT to a series of circular convolutions, for which the chip is ideal. Without the chip, the algorithm's appeal is limited because of the amount of multiplication required compared to the conventional radix-2 FFT.

Rader's prime number algorithm gives two basic ways to implement DFT in light of the chip's ability to handle long data blocks. One approach is to cascade enough filters to deal with an entire data block. That approach, however, becomes impractical for large amounts of data. The alternative approach breaks a large DFT into several short ones—a scheme that is particularly attractive since it allows a tradeoff between cost and speed.

Correlation and convolution operations are also ideally suited to the circuit's architecture. One way to evaluate these functions is in the frequency domain, where the prime-number algorithm and related decomposition techniques lend themselves to DFTs and inverse DFTs. Alternatively, the chip's advanced ability to multiply and accumulate means it can also perform these functions in the time domain. The A100 is effectively a 32-tap correlator (convolver) in which the samples of the two signals to be correlated can be expressed in words up to 16 bits long—a length corresponding to a 96-dB dynamic range.

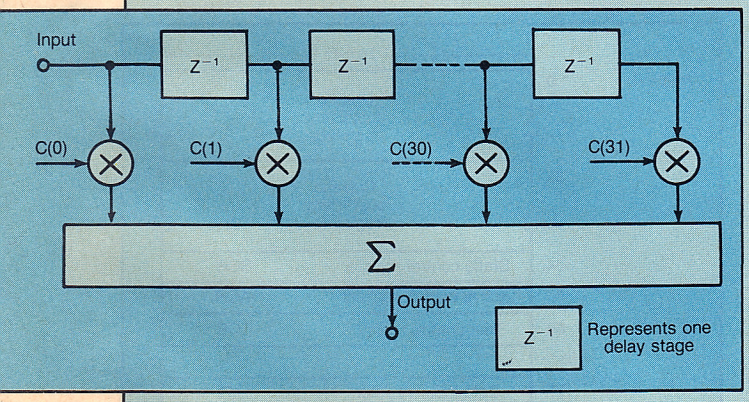
To extend the correlation beyond 32 taps, multiple circuits can easily be cascaded. As noted, with 16-bit coefficients the data rate can go as high as 2.5 Msamples/s, and as high as 10 Msamples/s with 4-bit coefficients, regardless of the number of stages required. Several other techniques allow a long correlation and convolution to be de-

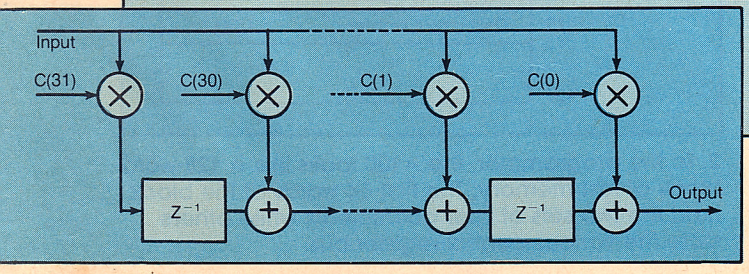
Price and availability

The IMS A100 cascadable signal-processor comes in an 84-lead pin-grid-array package. It is available in sample quantities and costs about $420 each in lots of 100. For more information in North America, contact Inmos Corp., P.O. Box 16000, Colorado Springs, CO. 80935; (303) 630-4000. CIRCLE 502

Transversal filter basics

The basic structure of a standard transversal filter consists of multiplication and delay stages (see figure, top). Each time the input is sampled, all the values sitting in the delay chain are multiplied by the coefficients located at their respective locations, and the results are added together to produce a result. The input data is then shifted down the delay chain by one position, ready for the next input sample. Thus the input data effectively traverses the filter over a period of time proportional to the number of stages, or the number delay locations, in the transversal filter.





Mathematically, the filter does nothing more than a series of multiplications and additions in a regular sequence, given by the following equation:

$$y(kT) = C(0) \times (kT) + C(1) \times ((k-1)T) + \ldots + C(N-1) \times ((k-N+1)T)$$

which can also be expressed as

$$y(kT) = \sum_{i=0}^{N-1} C(i) \times ((k-i)T)$$

where x(kT) represents the kth input data sample, and C(0) to C(N−1) are the coefficients for the N stages.

By modifying the architecture to do individual summations rather than a single addition, the complex adder can be decomposed into a sequence of simpler adders (see figure, bottom). The structure performs the same function as the first architecture, but each filter stage is self-contained, and that makes the structure seem like a systolic array. This form lends itself to multichip partitioning and, therefore, long transversal filters.

composed into several short ones, which can then be carried out by one or more devices. The host machine would be responsible for combining the results to obtain the overall solution.

Several approaches can also be taken to multiply matrices. One scheme sequences through the data that makes up a matrix equation (Fig. 3). Extending the matrix multiplication is simply a matter of considering the second matrix as a collection of vectors. One device can multiply matrices of dimensions N-by-M, where N+M is 32 or less. Larger matrices can be handled by partitioning techniques or by cascading several devices.

TOOLS, TOO

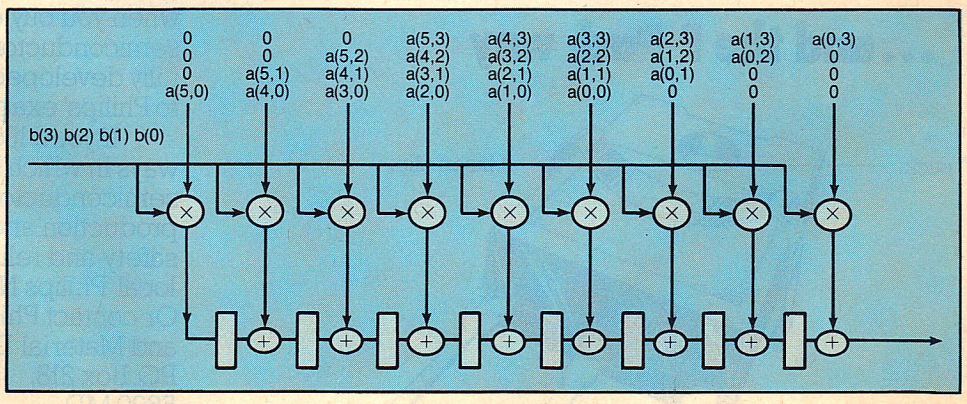
Software development time and cost requirements for the A100 are minimized by virtue of its memory-mapping, allowing it to be programmed using any conventional language with access to the host machine's memory. Design time takes weeks, not months. In addition, designers can draw from a range of support and programming tools, as well as application notes, that guide them through the creation of hardware and software.

One software tool is a system simulator written in Occam, a concurrent programming language. The simulator runs in any Occam 2 environment, such as the IBM PC with a B004 Transputer board. It lets designers investigate the performance of algorithms, model address decoding schemes, and observe the effects of cascading multiple devices. A still more important purpose is served by the software: As an executable system-level specification of the chip, it evaluates the circuit's behavior in a system. Users, for example, could capture image data on disk, then feed it through the simulator to observe the processed result. Once the desired algorithm is simulated, the software controlling the simulation can control hardware.

For the hardware engineer, an IBM PC-compatible board, the IMS B009, combines four filter chips, 16-bit IMS T212 and 32-bit IMS T414 Transputers, 1 Mbyte of dynamic RAM, and 64 kbytes of fast static RAM. The board serves as a complete Occam development environment, and software developed on it can run on any or all of its filter chips. Several boards can also be cascaded thanks to eight built-in Transputer serial links, each of which transfers data at rates of up to 0.5 Mbytes per second. Demonstration software that comes with the board shows how to implement primary algorithms, like correlation and discrete Fourier transforms, and is a way to evaluate popular FIR designs. □

Tony King-Smith is the design manager for Inmos' silicon systems group, which defines, builds, and helps customers with application specific ICs. He has a BEE from Melbourne University in Australia.

Hossein Yassaie is a senior technologist for the same group at Inmos and the architect of the company's signal-processing products. He has BS and PhD degrees in electronics, both from the University of Birmingham in England.



$$\begin{bmatrix} c(0) \\ c(1) \\ c(2) \\ c(3) \\ c(4) \\ c(5) \end{bmatrix} = \begin{bmatrix} a(0,0) & a(0,1) & a(0,2) & a(0,3) \\ a(1,0) & a(1,1) & a(1,2) & a(1,3) \\ a(2,0) & a(2,1) & a(2,2) & a(2,3) \\ a(3,0) & a(3,1) & a(3,2) & a(3,3) \\ a(4,0) & a(4,1) & a(4,2) & a(4,3) \\ a(5,0) & a(5,1) & a(5,2) & a(5,3) \end{bmatrix} \times \begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ b(3) \end{bmatrix}$$

3. Ordering the vectors in a matrix multiplication so that the values feed into the transversal filter (top) allows the chip to very rapidly run through a set of matrix to matrix calculations (left) in no time.