

Massana Unveils DSP Coprocessor Core

VLIW Design Adds DSP Horsepower to Host Processors



by Rick Grehan, Jeff Bier, and Jennifer Eyre, Berkeley Design Technology, Inc.

At Microprocessor Forum last month, Irish startup Massana presented its FILU-200, a new, licensable, VLIW-based DSP coprocessor core. The compact, fixed-point core is designed to bolt onto a general-purpose host core and provide significant DSP horsepower without requiring changes to the host processor. The FILU-200 (whose name, according to Massana, doesn't stand for anything) is specifically targeted at applications that make heavy use of fast-Fourier-transform (FFT) algorithms, such as DSL modems. For these applications, Massana aims to provide speed and numeric fidelity approaching that of custom hardware, while easing time to market for system-on-chip designs by offering an off-the-shelf core in synthesizable form, along with a library of prewritten DSP functions.

While DSP-enhanced microcontrollers and CPUs have become commonplace over the past few years, a combination of two factors differentiates Massana's approach from those of other vendors. First, the FILU-200 will work with any processor; although it is designed to work in conjunction with a 32-bit MCU such as an ARM or MIPS core, it could alternatively be connected to a proprietary MCU, a high-end CPU, or even a DSP core to boost performance. Second, Massana intends the FILU-200 to be viewed as a hardware-assisted DSP software library rather than as a programmable processor. Hence, the FILU-200 comes with a set of common DSP subroutines in ROM and a host-side high-level API for accessing those routines. Massana hopes that many application developers will be able to implement the key DSP functionality of their applications by using these built-in functions, avoiding the forbidding task of hand-crafting optimized DSP routines.

The best-known—and perhaps only—prior example of a licensable DSP coprocessor core was ARM's ill-fated Piccolo, designed to augment the DSP capabilities of the ARM7. Introduced in 1996, Piccolo was hindered by an unusual and complex host/coprocessor interface. Despite the vast number of licensees of the ARM7, Piccolo never gained a significant foothold in the market—arguably, because few potential licensees were willing to puzzle out its complicated interface. ARM has subsequently adopted a more conventional approach to DSP enhancements with its ARM9E (see MPR 6/21/99, p. 11). In contrast to Piccolo, the FILU-200 employs a straightforward host interface, as Figure 1 shows, and has the ability to work with any host processor core.

High-Level Programming Is Simple

Consistent with the paradigm of the coprocessor as a hardware-assisted library of DSP routines, Massana provides a set of host-side C functions comprising an API that wraps the nitty-gritty of host-FILU-200 communication within high-level function calls.

Executing a coprocessor function from within a host program is straightforward. The host application first calls the ConstructFILU function to initialize the host-to-FILU-200 interface and build the necessary static data structures on the host. A ResetFILU call initializes the FILU-200. Next, the host application issues a series of WriteFILU function calls to transfer data and coefficients to the FILU-200's RAMs. Finally, the host issues a StartFILU call, which invokes a specified FILU-200 DSP function, passing the address of a parameter list for the invoked function. The host can proceed with other tasks while the FILU-200 executes the selected DSP function. When the FILU-200 has completed the specified DSP function (determined via polling or interrupt), the host reads back the resulting data via one or more ReadFILU calls.

The FILU-200's built-in program ROM stores its library of prewritten functions, in the form of up to 32K of 156-bit very long instruction words, which Massana calls "macro-instructions." As Table 1 details, the FILU-200's ROM library consists of six fundamental DSP functions: radix-4 FFT and inverse FFT, FIR and IIR filters, vector multiply, and phase

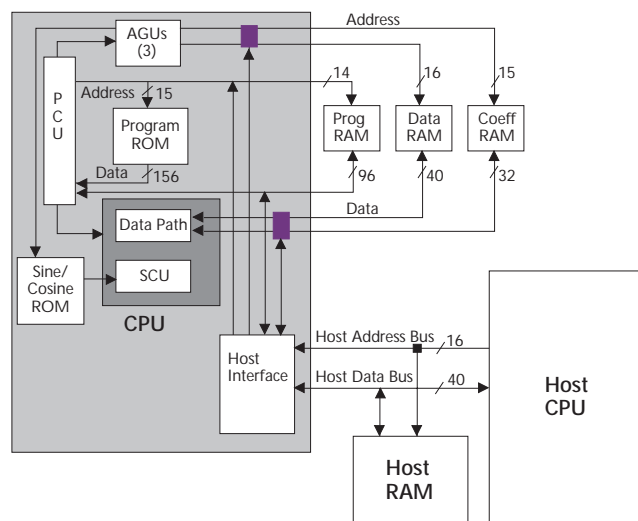


Figure 1. The FILU-200 coprocessor is meant to be used as an add-on DSP engine for a host processor. The coprocessor includes program ROM containing preprogrammed DSP functions, a program control unit (PCU), and a sine/cosine table ROM controlled by the sine/cosine unit (SCU). External data RAM is required; external program and coefficient RAMs are optional.

rotation. In addition, the FILU-200 contains rounding and normalization functions for converting blocks of data to either the FILU-200's internal 20-bit format or a 16-bit format for data returning to the host. Massana is currently developing additional ROM routines for other DSP algorithms, including adaptive filters, correlation, Taylor series, real and complex matrix multiplication, and lattice filters.

Architecture Specialized for DSP Algorithms

In contrast to some newer VLIW-based DSPs that have adopted more regular, RISC-like designs, the FILU-200 displays a high degree of specialization internally. Like the designers of previous-generation DSPs, Massana has chosen to forgo regularity in exchange for algorithm-specific features, with the goal of achieving strong DSP performance with a small core.

For example, the FILU-200 data path provides an unusual mixture of 16-, 20-, and 22-bit fixed-point data widths. Massana refers to the device as a 16-bit design, however, since data transferred to and from the host will generally be in 16-bit fractional form. The 20- and 22-bit data widths, as well as 44-bit accumulators, are tuned to the needs of specific DSP algorithms. The 22-bit format, for example, consists of a sign bit, two integer bits, and 19 fraction bits; this format is useful for some algorithms (e.g., the FFT) that benefit from more than 16 bits of precision for intermediate results. Using these data widths, Massana aims to achieve better numeric fidelity with FILU-200—without resorting to performance-sapping multiprecision operations—than other vendors achieve with typical 16-bit DSPs.

Like the device's data widths, the FILU-200's data path reflects a design tailored for fundamental DSP algorithms. As



VP Brian Murray presents Massana's approach for adding DSP capabilities to a host processor at the Forum.

MICHAEL MUSTACCHI

Figure 2 shows, the FILU-200 data path contains two data register sets (X and Y), each of which comprises five 22-bit registers and three 20-bit registers. The 20-bit registers can be loaded from RAM only and cannot be used as destination registers for any operations; hence, they are mainly useful for holding coefficient values. The register sets anchor a pair of independent data buses that bracket a near-symmetrical arrangement of function units featuring dual multiply-accumulate (MAC) units. Other function units include an ALU, an adder, and dual 44-bit barrel shifters.

The FILU-200's MAC units perform 22×16 -bit single-cycle multiplies. This asymmetry is another reflection of optimizations for specific algorithms, specifically FFTs, where maintaining 22-bit intermediate results helps the FILU-200 achieve fast implementations with good numeric fidelity. DSP algorithms that use the MACs must be designed so that constant coefficients—which have fixed precision—are fed to the 16-bit side of the MACs, while data values—whose precision may grow through the course of the algorithm—pass into the MAC on the 22-bit side. In this case, coefficients stored in 20-bit memory are truncated to 16 bits (by discarding the LSBs) before multiplication.

The FILU-200 supports block-floating-point data representation by tracking the minimum number of leading sign bits of blocks of data passing out of the two MAC units. (Block floating-point is a technique commonly used in fixed-point processors to increase numeric fidelity by associating a single exponent with a group of mantissas.) Normalization is then performed in the barrel shifters, which also handle rounding. The FILU-200 also supports five levels of nested zero-overhead hardware looping.

The FILU-200's sine/cosine unit (SCU) works in conjunction with a dedicated on-chip lookup ROM and specific address registers to provide rapid calculation of sine and cosine values (useful for FFT and phase-rotation algorithms) with 16-bit accuracy.

Ins and Outs

As Figure 1 shows, the FILU-200 DSP core is surrounded by four interfaces: three for external RAM (one each for data, coefficients, and program) and a fourth for host/FILU-200 communication. This host/FILU-200 interface gives the host direct access to the FILU-200's RAMs as well as control of the FILU-200's operation. Control takes place through a dual-ported control register that appears (from the host's point of view) at the lowermost address of FILU-200 RAM.

Because the control register is dual ported, the host can access it without slowing the FILU-200's operation. For example, the host can poll the control word's "busy" bit to determine whether the FILU-200 has completed its task.

Function	Description
FFT	N-point radix-4 FFT. (Both real and complex FFTs are provided, and there are 8 variations, depending on output ordering desired and FFT size.)
IFFT	Radix-4 inverse FFT, real and complex. As with the FFT, there are 8 variations.
FIR	Arbitrary-order finite-impulse response filter. Supports block filtering for input data sets larger than available FILU-200 RAM.
IIR	Second-order infinite-impulse response filter.
Vector Multiply	Real, complex, or mixed real-complex multiplication of arbitrary length vectors.
Rotor Function	Performs complex phase rotation of a vector. (Useful in V.90 and DSL systems.)
Normalization	Two functions for rounding and normalizing blocks of data.

Table 1. The FILU-200 on-core ROM includes preprogrammed DSP functions that can be called from the host via an API.

Alternatively, chip designers can utilize the FILU-200's "busy" signal to drive an interrupt on the host.

A minimal FILU-200 configuration requires only the data RAM. Program RAM is optional, allowing the host to download user-written FILU-200 routines. Coefficient RAM is also optional, allowing increased memory bandwidth where needed.

All FILU-200 RAMs are single ported by default. Although the host can access the FILU-200 data and coefficient RAMs while the FILU-200 is running, each host/FILU-200 access stalls the FILU-200 for one clock cycle—regardless of whether the two processors are accessing the same RAM or not. The host cannot access FILU-200 program RAM while the FILU-200 is running. Given the fine-to-medium granularity of the FILU-200's built-in DSP functions, the overhead of host-coprocessor communications will be significant—in some cases largely eliminating the benefit of using the coprocessor in the first place. According to Massana, the FILU-200 can also be used with dual-ported or multibank RAMs, which allow simultaneous access by the host and the FILU-200 at the cost of increased silicon area. But making use of this arrangement to reduce communication overhead will require application programmers to schedule their code carefully, so that data transfer for one function can occur in parallel with computation for another function.

Each of the FILU-200's three RAMs serves a different function. The FILU-200's data RAM consists of up to 64K 40-bit locations. This RAM holds data inputs (from the host) and results of FILU-200 processing, and it provides intermediate storage for executing algorithms. Data is accessed as pairs of 20-bit words; a single memory access can load or store two data registers. Thus, maximum bandwidth is achieved only when data can be accessed as pairs of 20-bit words in adjacent memory locations, with the pair aligned on a 40-bit boundary. This restriction is similar to limitations found in many recently introduced high-performance DSPs; it is a compromise that sacrifices flexibility for high peak bandwidth with low cost—avoiding the need for multi-ported or banked memories and an additional address generator and address bus.

The FILU-200's optional coefficient RAM consists of up to 32K of 32-bit words. The host views the coefficient RAM as 32-bit words; the FILU-200 views it as 16-bit words and can access one word per cycle. With this additional RAM in place, three values can be read or written in a single cycle, up from the two 20-bit data values per cycle if data RAM alone is used.

The FILU-200's optional program RAM stores user-written routines. Consistent with the spirit of specialization embodied in the FILU-200 architecture, software in program RAM uses different instruction encodings than software in program ROM. Program ROM uses 156-bit macroinstructions, each of which can contain up to 19 subinstructions; program RAM holds up to 16K 96-bit macroinstructions, each of which can contain up to 18 subinstructions.

The FILU-200 supports some of the register-indirect addressing modes common among DSP processors, such as pre-/post-increment and bit-reversed addressing. Circular addressing is not supported, however, nor are immediate data and absolute addressing. Direct addressing of registers is supported, with the caveat that register-to-register move operations tie up specific function units. For example, a move between X data registers requires the ALU, a move between Y data registers requires the adder, and a move between address registers requires the loop control unit.

Reading RAM is a pipelined process, with two-cycle latency and single-cycle throughput regardless of the addressing mode. In the first cycle, the data is fetched into a buffer register (in the RAM itself); in the second cycle, the data is transferred to the destination register. Writing data to RAM requires only one cycle. (Note that the FILU-200 cannot write to coefficient RAM; only the host can.) Accesses from ROM (such as the sine/cosine ROM) require only one cycle.

Internal Programming Model is Complex

While the FILU-200/host interface is simple, the internal programming model is quite complex. For users who can make do with the FILU-200's built-in DSP function library, this won't be an issue. Massana does not currently provide a compiler, however, so developers whose applications demand custom functions will have to wade in and grapple directly with a complicated VLIW machine. (Massana says that a compiler is planned.)

As Table 2 shows, the FILU-200 has 19 independent function units. A subinstruction can be issued to each unit during each clock cycle.

The FILU-200's instruction set reflects the core's deliberate nonorthogonal design. For example, because of the asymmetrical nature of the ALU/adder pair, most mathematical

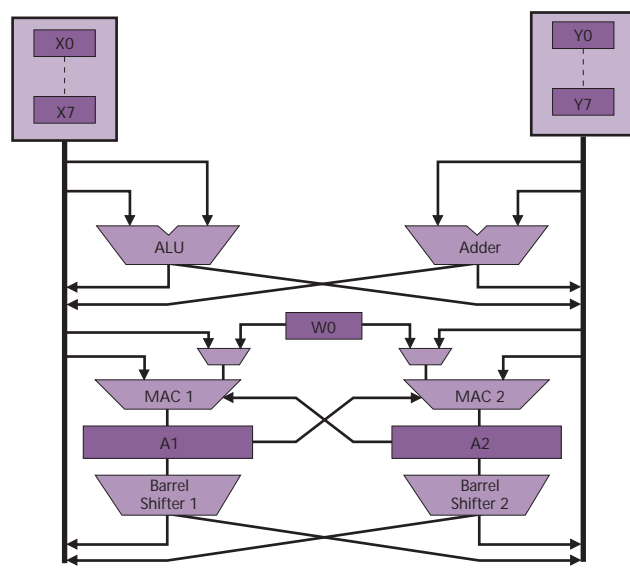


Figure 2. The FILU-200 data path includes two MAC units, an ALU, an adder, and two 44-bit shifters.

and logical operations—such as negate, absolute value, bitwise AND and OR—can be performed only on X data registers. In addition, though the FILU-200 has three address-generation units, one of the units is devoted exclusively to managing address register R2.

Fortunately, assembly-language programming won't be further complicated by the FILU-200 pipeline; it's only three stages deep. This depth is comparable to that of many older, conventional DSP processors and much shallower than that of TI's VLIW-based TMS320C62xx, whose 11-stage pipeline is the stuff of assembly programmers' nightmares. Programming the Massana design, however, is complicated by the two-cycle latency of memory loads.

It seems unlikely that users will be able or willing to program the FILU-200, given its wide instruction word and extremely nonorthogonal design. Even when a compiler does become available, it is doubtful that the efficiency of the generated code will be satisfactory for performance- and efficiency-minded users.

Multiple Instruction Encodings Supported

The FILU-200 supports multiple macroinstruction encodings. As mentioned earlier, the program ROM stores 156-bit macroinstructions, each of which can carry 19 subinstructions. This encoding allows a subinstruction to be issued to each of the 19 function units, with four of the subinstructions explicitly encoding the two-step SRAM load process.

The program RAM recognizes three encodings. Two encodings offer a 15-subinstruction density; the third offers a density of 18 instructions, with some restrictions on the operands involved. The encodings represent a tradeoff between the number of function units that can be accessed versus the flexibility of the subinstructions that can be issued to

a function unit. For example, neither of the 15-subinstruction encodings permits access to the FILU-200's status register access unit or to the control-word access unit. The 18-subinstruction encoding does, but limits the shift operations that can be performed, restricts the address registers that can be used by the AGUs, and imposes a relative jump distance limit of 128 macroinstructions.

The assembler determines the particular encoding used for a given subinstruction mix; the presence of certain subinstructions will force a particular encoding. The programmer must ensure that multiple subinstructions don't collide on the same function unit in a single macroinstruction (the FILU-200 assembler will flag such collisions).

Plug and Play

Massana provides the FILU-200 as fully synthesizable Verilog HDL. To ease integration of the core into a chip, the FILU-200 uses a single clock, avoids the use of latches, and supports scan testing. The current design targets 100 MHz in a 0.25-micron process; a 0.18-micron version is projected to reach 150 MHz.

The FILU-200 development kit includes the HDL design and a number of software tools, including an instruction-set simulator and a tool for verifying the syntactical correctness of user-written code. The kit also includes a Verilog test-bench tool that allows the developer to test new functions using the FILU-200's Verilog design and a Verilog simulator, and compare their results with those obtained from the instruction-set simulator.

Solid DSP Performance

Independently verified benchmark results are not yet available for the FILU-200. In his Microprocessor Forum presentation, Brian Murray, Massana's VP of engineering, highlighted the projected performance of the FILU-200 on a 256-point FFT against that of several prospective competitors. At its projected 100-MHz clock speed, Massana's results suggest that on the FFT the FILU-200 will be about half as fast as TI's high-end 'C62xx running at 300 MHz, and roughly twice as fast as Lexra's 200-MHz LX5280 and DSP Group's 130-MHz TeakDSPCore. The FILU-200 achieves this performance using only 0.7 mm² in a 0.25-micron process; the TeakDSPCore, by way of comparison, requires 1.7 mm² in a 0.25-micron process.

Unit	Operations and Operands
Adder	Y registers only: move, add, subtract, increment, decrement
ALU	X registers only: move, add, subtract, increment, decrement, logical, single-bit arithmetic shift, negate, absolute value
MAC 1	MAC, multibit shifts (X register inputs)
MAC 2	MAC, multibit shifts (Y register inputs)
Barrel Shifter 1	Arithmetic and logical shifts of A1 accumulator
Barrel Shifter 2	Arithmetic and logical shifts of A2 accumulator
Address Generation Unit 1	Indirect address w/post-increment or post-decrement (all addr registers)
Address Generation Unit 2	Special modes for R2 register only
Loop Counter Generation Unit	Counter register support; also address register increment/decrement
Jump Unit	Conditional, unconditional, and do {...} while support
Block Exponent Unit	Block floating-point support
ROM Address Lookup Unit (AGU)	Special addressing for sine/cosine and FFT coefficient calculations
Data RAM Access Units (2)	Composed of two function units (one fetches from RAM to buffer register; the other fetches from buffer register to FILU-200 register)
Coefficient RAM Access Units (2)	Same as data RAM access units
ROM Access Unit/WO Load Unit	Supports high-precision sine/cosine calculations
Status Register Access Unit	Fetch/set status register contents
FILU Control Word Access Unit	Set rounding mode (control word also accessed by host)

Table 2. The FILU-200 contains 19 parallel function units, each of which can execute a subinstruction in every clock cycle. The number of subinstructions that can be encoded in a single VLIW (macroinstruction) varies from 19 in program ROM to 15–18 in program RAM.

These figures, however, don't include the time required for the host to transfer the input and output data to and from the FILU-200 RAMs, nor the five cycles required to initialize the coprocessor. This overhead could have a significant impact on performance—particularly for small functions such as FIR filters.

In addition, extrapolation from the FFT benchmark is hazardous, since the FILU-200 architecture is particularly well suited to FFTs. Nevertheless, the FILU-200 should compete competently in terms of speed against current dual-multiplier DSPs like the 'C62xx and TeakDSPCore, particularly if Massana can crank up its clock speed. Unfortunately, the shallow pipeline and requirements of synthesizability may limit the clock speeds attainable.

New? Not Really ...

Massana's "hardware-assisted DSP software library" paradigm is unusual, but not unique. Somewhat surprisingly, Motorola's MPC82xx embedded PowerPC devices have long used a similar approach. These devices, introduced in 1995, incorporate an elaborate "communications processor module" that, among other things, provides a similar library of about a dozen microcoded DSP routines accessed via a high-level API. In the case of the Motorola devices, the DSP function library is specialized for V.32 and V.34 voice-band modems.

DSP applications are quite varied and use very specialized algorithms, making it unlikely that a single small function library will meet most of the signal-processing needs of a range of applications. Hence, specializing the library for a particular application or narrow range of applications is key to the success of Massana's approach. The FILU-200 itself serves as a generic template, from which the company is creating a family of application-specific derivatives. For example, the FILU-DMT is tailored to the G.Lite flavor of DSL modems, and it includes a specialized function library in ROM, as well as additional interface logic, to support this application. According to Massana, other coprocessors are under development for full-rate DSL, voice-over-IP, digital audio broadcast, and other applications. Some of these will be based on the FILU-200, while others will be based on variants of the architecture tuned to the relevant algorithms.

A Black Box Looking for a Niche

The FILU-200 bucks the trend of more-general-purpose, less-specialized DSP architectures found in recent high-end VLIW-based DSPs such as TI's TMS320C62xx and StarCore's SC140. While users accustomed to more orthogonal architectures may be baffled by such features as a 16/20/22-bit data path and addressing modes specialized for radix-4 FFTs, specialization like this has served DSP applications well for the past two decades—allowing conventional DSP architectures to meet challenging performance and numeric-fidelity demands while conserving silicon area and power.

As 0.18-micron technology approaches the mainstream, though, the justification for such a highly non-

Price & Availability

The FILU-200 is a synthesizable core, provided in Verilog HDL. Initial licensees are currently designing chips with the FILU-200, according to Massana, with first silicon expected in early 2000. Massana has offices in Campbell, Calif., and Dublin, Ireland. Contact Massana for licensing fees and terms.

For more information, call 408.871.1415 or visit www.massana.com.

orthogonal architecture is debatable. In today's system-on-chip designs, a conventional DSP core (such as DSP Group's OakDSPCore) might occupy 10% of the die area. As densities increase, the area penalty paid for a more regular architecture decreases.

Typically, highly nonorthogonal architectures sacrifice some performance for compactness, but programming difficulty is increasingly seen as the most critical weakness of such architectures, as indeed it is in the FILU-200 design. This compromise may work for Massana, though; the company is not positioning the core as a general-purpose device but rather as a quick solution—with software built in—for system-on-chip designs requiring strong FFT performance. By focusing on specific applications and providing well-tuned software libraries for those applications, Massana may be able to sidestep programmability difficulties and deliver the best of both worlds: a powerful, economical processor that presents a simple programming model to application developers.

Given the vast array of DSP-enhanced general-purpose processors currently being offered, one may reasonably question whether there is room for yet another approach to combining host and DSP capabilities. Due to the complexity of developing new DSP routines for the device, the FILU-200 has very limited appeal as a general-purpose DSP engine.

Massana's approach, however, is likely to appeal to developers of system-on-chip devices who want a "black-box" solution to their application-specific DSP needs, and to those who value the flexibility of a DSP coprocessor that isn't bound to a particular host architecture. Massana's reliance on DSP subroutine libraries tuned for specific applications is both a key strength and a vulnerability of the FILU-200. Such libraries will indeed provide strong appeal to system developers suffering increasingly painful time-to-market constraints, but nothing prevents other vendors of licensable DSP cores from offering equivalent libraries, potentially eliminating one of the FILU-200's key advantages. ■

The authors are with Berkeley Design Technology, Inc. (BDTI), the DSP technology analysis and software development firm (www.BDTI.com). BDTI's flagship industry report, Buyer's Guide to DSP Processors, is available from MDR.