

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

EMBEDDED ARRAYS VENTURE FORTH

IntellaSys 24-Core SEAforth Chips Target Low-Power Multimedia

By Chris Bailey {8/21/06-03}

After years of neglect from the mainstream computer industry, will Forth—the “fourth-generation” programming language invented by Chuck Moore in the 1960s—make a comeback? It might, if Moore’s new company, IntellaSys, proves successful.

At In-Stat’s Spring Processor Forum in May, Moore introduced a novel embedded-processor architecture for low-power media applications. The first two implementations of the SEAforth (Scalable Embedded Array) architecture cram 24 processor cores on a single chip, have 18-bit words, use asynchronous logic, and have a simple 30-operation instruction set based on the high-level Forth language. Almost every aspect of these devices rebels against mainstream design principles.

Moore says the time is right for large arrays of low-cost Forth-based embedded processors that can tackle media-processing tasks in next-generation consumer electronics. A single SEAforth chip has enough compute power to handle the multiple media-processing algorithms required for audio, video, and voice applications in portable electronics products. Although some other companies make the same claims for their media processors; Moore says the IntellaSys breakthroughs provide stunning performance (up to 24 billion operations per second); ultralow power consumption (about 150mW); low cost (under \$10 in very high volume); and concise programmability (the native machine language is a derivative of Forth).

Target applications include hearing aids, consumer audio, media processing, and smart phones. One aspect of the IntellaSys strategy is to provide large arrays of identical high-performance processor cores that programmers can easily configure into ad-hoc groups to accomplish specific tasks. Another is to dramatically reduce power consumption

in the multicore array. IntellaSys has achieved both objectives by implementing a radical design in asynchronous (clockless) logic.

No Shortage of Development Funds

Unlike most startups, privately held IntellaSys can fuel its efforts with a healthy revenue stream estimated in the tens of millions of dollars per year—even before shipping the first SEAforth chips. One source of revenue comes from two families of consumer-media processors that IntellaSys acquired this year from Indigita and OnSpec. These chips give IntellaSys major customers in Taiwan, China, and Japan and also provide insight into next-generation consumer products, such as high-definition digital video recorders.

A bigger source of revenue is the Moore Microprocessor Patent (MMP) portfolio. In the 1970s and 1980s, Moore patented key technology related to integrated processors, I/O clock circuitry, instruction fetching, on-chip oscillators, and embedded memory. The privately held TPL Group and the publicly traded Patriot Scientific are co-owners of the MMP portfolio, which is managed exclusively by Alliacense—which, like IntellaSys, is owned by the TPL Group. Alliacense has licensed the portfolio to marquee microprocessor manufacturers such as Intel and AMD, as well as to major system manufacturers, including Casio, Fujitsu, Hewlett-Packard, Nikon, Pentax, Seiko Epson, and Sony. By some estimates, more than 100 other system manufacturers are

using Moore's patented technologies and are in line to become licensees.

These twin revenue streams have enabled Moore and his design team to create what is perhaps his ultimate Forth machine—the array of embedded Forth processing cores in the SEAforth architecture. It's not the first time Moore has attempted to implement Forth as the native machine language in a microprocessor. It is, in fact, his seventh attempt. Moore says Forth-native hardware has a significant execution advantage when performing the types of recursive algorithms needed for advanced audio and video processing.

First SEAforth Devices Have Different I/O

The first IntellaSys products are the SEAforth-24A and SEAforth-24B. Each chip has a 24-element array of 18-bit C18 processor cores, a real-time clock, two 18-bit A/D converters, two 9-bit D/A converters, and interfaces for external SRAM and DRAM. Each chip typically consumes about 150mW. The primary difference between the two devices is that the SEAforth-24A has one SPI I/O port, 10 serial I/O lines, and an 18-bit parallel I/O port, whereas the SEAforth-24B has 11 SPI I/O ports and 32 parallel I/O lines. Figure 1 is a block diagram of the SEAforth-24A.

Each C18 processor core can execute as many as one billion operations per second (BOPS), so the maximum theoretical throughput of a 24-core chip is 24 BOPS. This performance makes it possible to combine standard I/O functions with embedded algorithms running on internal cores. Programmers can configure the cores to perform multiple tasks in parallel.

For example, one set of cores could run 18-bit fast Fourier transforms (FFT) and discrete Fourier transforms (DFT), with another set handling audio processing, and another group performing wireless communications tasks. IntellaSys anticipates targeting embedded applications in

which the combination of high performance, low power, and low cost can make a significant difference for audio processing, wireless communications, home automation, remote data collection, and security.

To achieve high throughput and low power, IntellaSys implemented the SEAforth array in asynchronous logic. There is no system-wide clock, so each processor core can run at full speed, independently of the rest of the array. When not executing code, the cores stay in a low-power (less than 1mW) quiescent mode. No central system clock means no clock-tree circuitry throughout the chip, greatly reducing power dissipation. A second benefit of asynchronous logic is fewer coincident clock signals, greatly reducing on-chip signal noise. Two years ago, those same benefits convinced ARM to form a partnership with Handshake Solutions to design the new ARM996HS, the first commercially available 32-bit processor core implemented in asynchronous logic. (See *MPR* 2/21/06-01, "Can ARM Beat the Clock?")

Unlike traditional multicore processors or parallel-processing arrays, the SEAforth architecture needs no complex and power-hungry interprocessor communication mechanisms. Cores communicate directly by reading and writing each others' communication registers. This simplifies the array design and greatly reduces power consumption, because complex handshake signals require power to manage the communication links, even when the processor cores aren't communicating. Instead of using an on-chip bus or token-ring message system, a SEAforth core simply talks to its nearest neighbors via direct reads and writes to internal registers. For dedicated core-to-core communication, it's all that is needed.

When multiple cores need to communicate to one or more cores, the read/write registers have two transaction bits that indicate transaction status. For chip-to-chip communication,

a dedicated core on each side of each chip sends and receives signals through a SPI port. Because the cores can run almost any communications protocol, chip-to-chip signaling can use the hard-wired SPI port (programmable for any two-pad serial protocol, up to 50Mb/s) or a wireless link.

Self-Contained Cores Run Autonomous

The SEAforth C18 processor cores use 18-bit-wide instruction and data words. IntellaSys chose that width to take advantage of low-cost 18-bit-wide memory. In the SEAforth-24A, each core has 64 words of BIOS ROM and 64 words of user RAM. In the SEAforth-24B, each core has 512 words of ROM and RAM. The BIOS includes all the interprocessor communications routines and defines the default I/O functionality.

As Figure 2 shows, each core also has five 18-bit registers and a separate return stack and data stack, which are 18 bits wide. The data stack is 10 entries deep; the return stack is 9 entries deep. These stacks are well

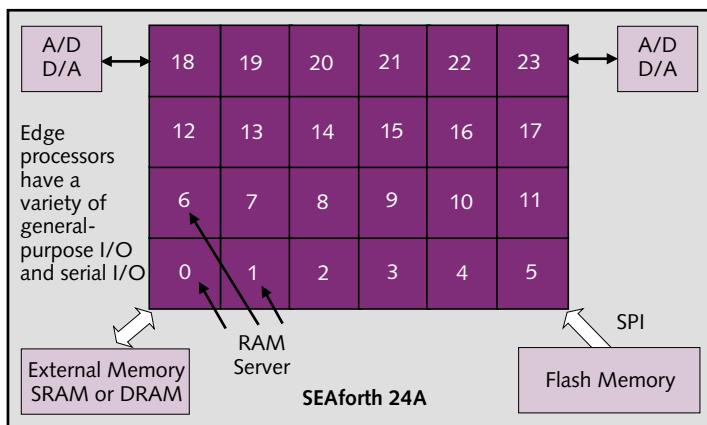


Figure 1. Both of the first two SEAforth chips have an array of 24 processor cores. All the 18-bit processor cores are functionally identical, but some perform different tasks. In particular, cores at the edges of the chip handle various I/O duties.

suites for executing the recursive algorithms used in audio and video processing.

Instructions are three or five bits long. An 18-bit word may contain up to four instructions: three five-bit instructions and one three-bit instruction. Those are just enough instructions to pack a four-instruction loop into a single 18-bit word, which can eliminate repetitive instruction fetches. This technique is particularly useful when moving data from memory to memory or from core to core, because the tight loop works like an efficient DMA mechanism. (SEAforth processors don't have a DMA controller.)

IntellaSys says that four C18 cores working in concert can execute digital-signal processing routines as fast as a typical DSP processor can. Therefore, IntellaSys found no reason to develop specialized DSP functions. Instead, SEAforth relies on fast DSP execution in the general-purpose C18 cores. Some other features considered but rejected were additional on-chip memory (deemed too expensive in power and silicon real estate) and a single-cycle multiply instruction (deemed unnecessary, given the chip's potential for parallelism).

Untethered to a master system clock, each processor core can run at the fastest speed possible—up to 1.0GHz in current versions. The cores execute code in their local ROM and RAM while communicating and sharing the computing load with their neighbors. They can load user code into local RAM through the SPI ports from external flash memory or another external source. With each processor executing its own code from local memory, there is usually no memory bottleneck—unless multiple cores must fetch code from external memory at the same time. Cores can pass data, status bits, or even blocks of code among themselves, using the interprocessor communication routines built into the local ROM BIOS.

A core waiting for data from its neighbor automatically goes into sleep mode. In fact, sleep mode is the default mode. If a core needs to send data to another core that isn't ready, it goes into sleep mode until the transaction partner is ready to accept the data. When the transaction is complete, the receiving core sets a status bit in the sending core. This “sleep when not working” approach, while deadly to the careers of most of us, works well on a SEAforth chip. It keeps power consumption to a minimum, despite the large number of high-speed processors at work. And because everything is implemented in static CMOS logic, moving between sleep mode and active mode requires no startup or shutdown overhead. If necessary, a processor can stop executing, even in the middle of an instruction, to wait for data to arrive.

Given Moore's long history of developing Forth processors, it makes sense that the communications mechanisms took longer to design than the processor cores. Moore says it took a while to figure out how to get all the cores talking efficiently. These mechanisms are crucial for complex media processing and particularly for acoustical processing algorithms.

Instruction Set Is Simple and Extendable

At a time when instruction sets with several hundred operations are commonplace, it's startling to find a microprocessor architecture with fewer than three dozen instructions. The SEAforth architecture's native machine language is called VentureForth, and it consists of just 30 instructions. However, the architecture also supports Forthlets—code objects that programmers can use to extend the instruction set.

As noted above, VentureForth is extremely compact, squeezing as many as four instructions into an 18-bit instruction register. Table 1 shows the complete VentureForth instruction set with its unusual mnemonics. Fast instructions operate on the ALU, stack, and address registers and execute in 1ns. Memory-access instructions take longer, from 2ns to 4ns. The most common memory access is an instruction fetch, which can be overlapped with a fast instruction, netting an execution time that may vary from 0ns to 3ns.

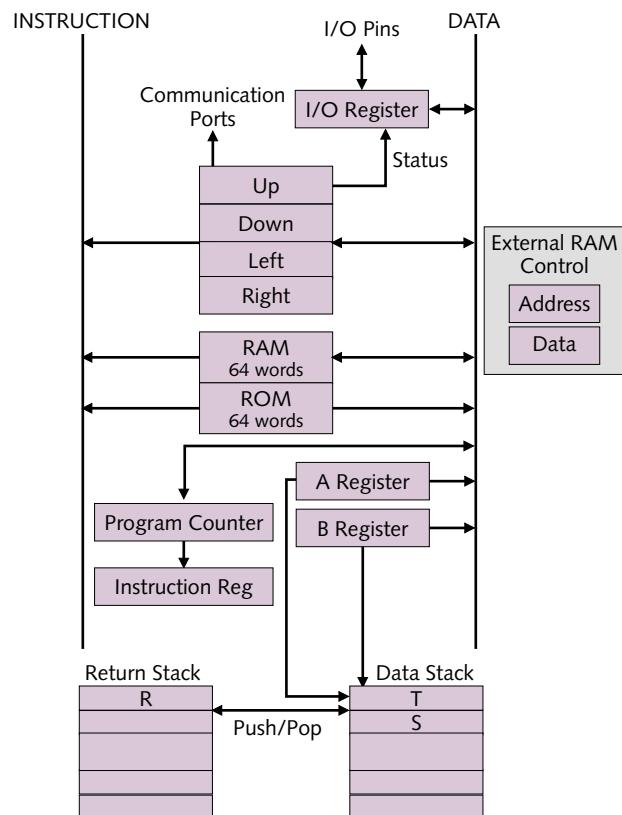


Figure 2. Each SEAforth C18 processor core has a simple five-entry register set (A, B, instruction register, I/O register, and program counter), dual stacks (return and data), and local RAM and ROM. Typically, one stack holds return addresses for subroutines, and the other passes parameters. This dual-stack architecture makes the C18 well suited for the recursive algorithms used in many audio- and video-processing routines. In the first two SEAforth chips, only one C18 core in the 24-core array has an external memory interface.

Price & Availability

The SEAforth Scalable Embedded Array chips will be sampling in 4Q06, with volume availability in 1Q07. They are priced at \$19.95 each in quantities of 1,000, with steeper discounts available in higher volumes. For more information about IntellaSys, visit www.intellasyss.net.

For an interesting treatise on stack-oriented programming, see *Stack Computers* by Philip J. Koopman, Jr.: www.ece.cmu.edu/~koopman/stack_computers/index.html.

IntellaSys offers a Forthlet library that so far includes nearly 100 independent code objects. These objects can be rapidly called from user code. Included are many media-processing Forthlets, such as MP3 (MPEG-1 Audio Layer 3) and H.264/MPEG-4 Advanced Video Coding. H.264 compression can provide DVD-quality video over moderately fast

Instruction	Description
Memory-Access Instructions	
;	Return from subroutine
call	Call subroutine
jump	Jump to address
next	Jump if return stack +; decrement
if	Jump if top of stack = 0
-if	Jump if top of stack = + or 0
@p+	Read in-line number
@+	Read using address in register a, then increment a
@b	Read using register b
@	Read using register a
!p+	Store to address in register p, then increment
!+	Store using register a, then increment
!b	Store using register b
!	Store using register a
Fast Instructions	
+*	Multiply step
2*	Shift left
2/	Shift right
-	Ones complement
+	Add
and	Logical and
xor	Exclusive-or
drop	Discard
dup	Duplicate top of stack
pop	Pop from return stack
over	Read from second stack register
a	Read from address register
.	No operation (NOP)
push	Push onto return stack
b!	Store into address register
a!	Store into address register

Table 1. VentureForth—the native machine language of the SEAforth architecture—is a streamlined 30-operation instruction set. As with any Forth language, programmers can easily extend VentureForth with new instructions, called words. In addition, IntellaSys provides an extended library of code routines called Forthlets.

(1Mb/s) broadband connections. Forthlet objects can move from core to core to perform specialized processing tasks.

IntellaSys supports VentureForth with development tools based on its own T18 compiler and simulator, running on Windows, Linux, and the Mac OS. Initial versions have simple command-line interfaces and support basic debugging functions, including register setup, single stepping, and breakpoints, as well as the ability to simulate all SEAforth 24-core processors. The quality of the development tools may be secondary to the question of whether programmers are willing to adopt VentureForth for all software development on SEAforth processors. C isn't an option, and the assembly language is VentureForth. (See the sidebar, "Forth: Honed by the Rock of Experience or Too Rocky to Deploy?")

Enhancing Cellphones and Home Theaters

Because SEAforth processors are fast enough to process RF signals on chip, IntellaSys is focusing initially on audio and video applications. The existing SEAforth C18 cores, built in a 0.18-micron digital CMOS process, can support a 100MHz A/D sampling rate, making possible the support of direct RF signals in the 10–20MHz range.

The A/D sampling frequency is a function of the software's ability to read and process data from the on-chip A/D and D/A converters, so this capability will improve in subsequent versions, as speeds increase. Future SEAforth processors built in more-advanced manufacturing processes, such as 130nm, will be even smaller and faster, so IntellaSys plans to reduce the number of cores required to support 1.0GHz conversion rates, from 10 cores today to 3.5 cores by 2009. At the same time, smaller design rules will allow IntellaSys to fabricate chips with hundreds of cores in their arrays.

Figure 3 shows a die photo of the SEAforth-24A, implemented in standard 0.18-micron CMOS. When migrated to 130nm, the chip's operating voltage will fall to 1.3V.

With a growing in-house library of Forthlets for MP3, H.264, MPEG-2, and other media codecs, IntellaSys is presenting SEAforth processors as nearly out-of-the-box solutions for audio and video processing. IntellaSys claims that H.264 protocol processing requires only six cores, and that playing cellphone-quality video requires only 12 cores. Seeing the close fit between its devices and the requirements for next-generation smartphones, IntellaSys views cellphones as a vast potential market—although the entrenched competition (primarily ARM and Texas Instruments) will be extremely difficult to dislodge.

An easier market to penetrate may be home theater. IntellaSys envisions a SEAforth-based wireless media server that eliminates speaker cables; shrinks the size and power consumption of the audio/video receiver; provides lossless audio transmission from the receiver to "smart" 5.1-channel surround-sound speakers; and automatically optimizes the audio for any room or listening environment.

Forth: Honed by the Rock of Experience or Too Rocky to Deploy?

The Forth language has both passionate enthusiasts and loud detractors. Enthusiasts praise its concise power, flexibility, extensibility, and efficient use of memory and other system resources. Detractors abhor its unusual attributes: it is stack-based, performs no type-checking, uses postfix arithmetic (reverse Polish notation), and consists of a small set of initial instructions that programmers must adapt to specific applications.

The predefined instructions or "words" are only a starting point. Forth programming consists of defining new routines (also called words) by combining existing words. Programmers can add new application-specific words, constructs, and data structures at will. Although this approach is flexible, it can also result in seemingly unreadable code filled with odd symbols and instructions that make no sense to other programmers. One critic labels Forth programmers as "brain damaged."

For embedded programmers, however, Forth's stack-oriented approach delivers advantages that can make it worth mastering in the appropriate hardware and application environment. Forth is hierarchical, simple, and extensible. Although simple in form and structure, it is also very powerful, because programmers can build new features

into the language as needed. In general, Forth enables rapid code development with a very economical use of memory.

Chuck Moore, who has spent more than 30 years developing Forth in various hardware and software forms, says "Forth is the first language which has been honed against the rock of experience before being cast into bronze." The new SEAforth architecture is his seventh attempt to implement a version of Forth in hardware. Previously, Moore created several versions of hardware Forth processors, including the Novix NC4000, the Harris RTX-2000, and the 32-bit ShBoom. (See *MPR 4/15/96-01, "New Embedded CPU Goes ShBoom."*) His SEAforth strategy is to use many powerful, very fast cores to execute concise Forth code. This architecture keeps power consumption to a minimum while giving programmers maximum flexibility to tackle multiple complex processing tasks simultaneously.

Moore admits that Forth has never won the corporate backing that propelled C and Java to success. At IntellaSys, his goal is to promote VentureForth as a rebirth of the Forth language, at least for a segment of the embedded-systems market.

SEAforth chips are inexpensive enough to be placed in each component of the receiver and speaker system, yet they are powerful enough to perform all the audio processing, wireless communications, and audio calibrations. Power amplifiers could be relocated from the receiver to the individual speakers. The streamlined receiver will contain only the RF front-end and switching functions plus user controls and displays. This design reduces the receiver's power dissipation, allows the unit to draw power from a small AC-DC converter, and enables the receiver to fit into smaller, more esthetically pleasing cabinets. The wireless receiver could even be built into a wall or piece of furniture.

SEAforth chips in the speakers could not only process the wireless 5.1-channel sound stream but also use audio feedback to automatically calibrate the sound envelope. Feedback-driven calibration could allow listeners to move the audio "sweet spot" around the room with a remote control. This adds a new feature that enhances the consumer appeal of the product.

Even though the SEAforth chip in each component of this hypothetical home-theater system performs a different function, the difference is the software, not the hardware, which reduces the component cost. For high-volume consumer-electronics manufacturers that might use SEAforth chips in many different products, the economy-of-scale advantages could be significant.

Is Forth the Next Big Thing? We'll SEA.

In the 1980s, Forth found a few niches, but it never won widespread support. SEAforth may lead embedded-system

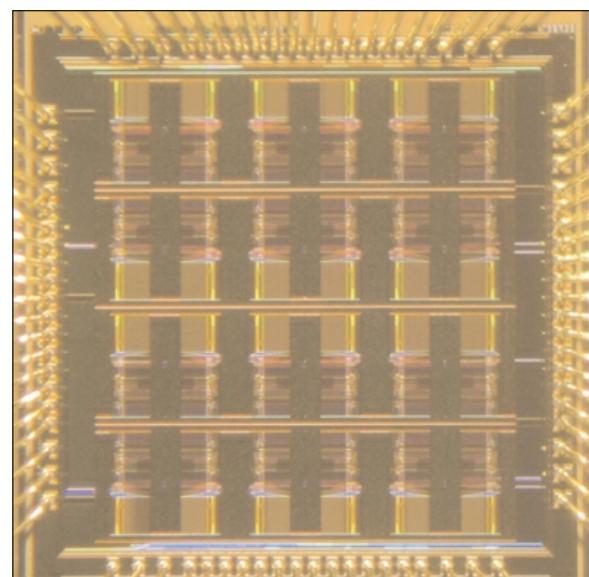


Figure 3. The SEAforth-24A embedded array integrates 24 identical processor cores with on-board serial and parallel I/O ports plus A/D and D/A converters.

developers to take another look at the language (or, in the case of younger developers, perhaps their first look).

IntellaSys, fueled by a healthy royalty stream, seems intent on delivering powerful embedded processors with system-level advantages. If developers aren't deterred by the odd SEAforth architecture and quirky VentureForth programming language, they may find genuine benefits in the technology. ♦

Chris Bailey has participated in and covered the embedded-processor market for more than 20 years. He started his career as

a field applications engineer with Motorola Semiconductor, where he helped customer engineers apply advanced microprocessor technology. He also built a homebrew video computer from spare chips and wrote and debugged the BIOS in assembly language. Later, he held marketing and executive positions at several processor and processor-IP firms, acquiring C, C++, and SQL programming skills. Along the way, he spent several years covering Silicon Valley as a technology editor, first with Electronic Design magazine and then with Systems Integration. For the past decade, he has specialized in helping semiconductor and networking startups and working with industry consortiums

To subscribe to Microprocessor Report, phone 480.483.4441 or visit www.MPRonline.com