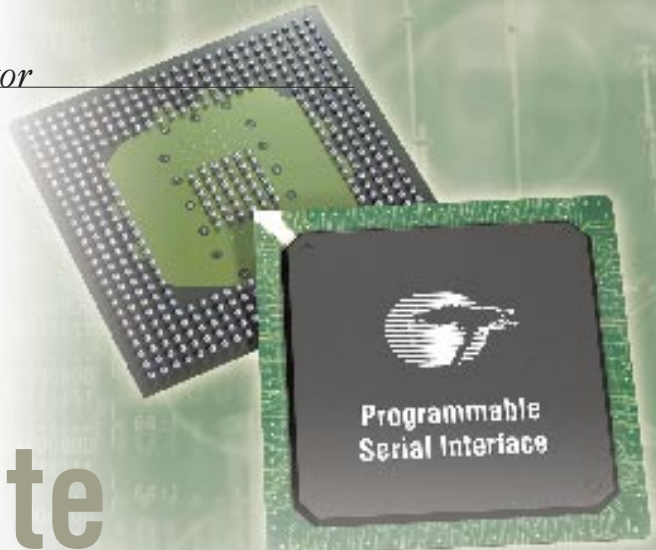


Do combo chips compute (or even compile)?



(Courtesy Cypress Semiconductor)

SINGLE-CHIP ASIC/PROGRAMMABLE LOGIC HYBRIDS ARE MOVING OFF THE DRAWING BOARDS AND INTO SAMPLING AND PRODUCTION AVAILABILITY. YOU REQUIRE A DESIGN AND VALIDATION FLOW THAT ISN'T RADICALLY DIFFERENT FROM THE ONE YOU'RE CURRENTLY USING BEFORE YOU CAN SERIOUSLY CONSIDER THEM. ARE THE VENDORS UP TO THE CHALLENGE?

IN NOVEMBER 1999, when *EDN* explored general-purpose ASIC/programmable-logic hybrid devices, only Lucent Technologies (soon to be Agere Systems), QuickLogic, and Triscend were shipping products, and Adaptive Silicon and LSI Logic's jointly developed embedded-FPGA test chips were still in design (**Reference 1**). The FPGAs with embedded-RAM blocks and PLLs that the other vendors kept pointing to as their highly integrated "system chips"

didn't count as true hybrids, at least in my book. And those other vendors were vigorously asserting that fully programmable logic chips would sooner or later solve all of your design problems, making hybrids unnecessary (**Reference 2**).

Fast-forward to the present and look at how things have changed. Lucent, QuickLogic, and Triscend have all broadened their standard-product portfolios. Atmel has joined them, with Altera, Cypress, and Xilinx set to begin shipping their first hybrid chips this year. And, in the embedded-FPGA arena, Adaptive Silicon's past solo status has now become a multivendor race. The competition now includes Systolix; Actel, which last year pur-

chased start-up Prosys Technology; and ICT (Integrated Circuit Technology), which is attempting to reinvent itself as an embedded-PLD-technology provider.

What's changed? Process technology, for one thing. Many vendors surveyed for this article agree that, at the 0.18-micron-lithography threshold, combining the amounts of on-chip ASIC and programmable logic required to begin credibly touting systems on chips becomes cost-effective. A 0.25- or even 0.35-micron process may be sufficient, though, for embedding simple ASIC-housed logic, such as PCI cores or 8-bit microcontrollers. The smaller programmable-logic vendors have all taken advantage of this

At a glance 102
Co-verification choices 104
Acronyms 106
Pinout and performance promises 106
For more information 118

fact to establish an early lead over today's larger programmable-logic suppliers in the hybrid-standard-product segment, which analysts predict will grow rapidly (Reference 3).

Using the terms "larger" and "smaller" to describe vendors also hints at why the hybrid-chip market has only recently become crowded. Programmable-logic manufacturers have spent years trying to compete against gate-array ASIC suppliers, and to a lesser extent, standard-cell ASIC suppliers. And ASIC manufacturers have focused on beating back the onslaughts of the PLD and FPGA upstarts. Hybrid chips, in some sense, represent both camps' acknowledgment of the alternatives' strengths, along with their own shortcomings. This politically incorrect admission requires a momentum shift in corporate policy that takes longer for larger companies to implement than their smaller, more nimble counterparts. Smaller suppliers also have less to lose and more to gain from the risky mid-course correction that hybrids might represent.

AN ENHANCED FPGA

Traditionally, your system design might have included one or more PLDs or FPGAs containing circuits you design, along with some off-the-shelf RAM, FIFO, and interface-translation chips, and a microcontroller, DSP, CPU, or multiprocessor combination. Now, however, you might want to combine some or all of these functions in one device to eliminate the performance loss, power consumption, and radiated EMI of chip-to-chip interconnect; increase overall reliability; or reduce board space. Still, you don't want to turn your back on the programmable-logic-tool flow that you've already learned and spent lots of money assembling.

Fortunately, you don't have to throw all that design expertise away. The ASIC-housed core appears to your programmable-logic synthesis compiler, and to your fitter or place-and-route tools, as a compiled "black box," with inputs and outputs that you connect to your logic. If you already use the memory-compiler functions offered with FPGAs to construct FIFOs or single- or multi-port RAMs, you are familiar with this concept.

For simple or low-flexibility cores, such as PCI, logic and memory in the

AT A GLANCE

- ▶ The ASIC/programmable-logic single-chip hybrid market has become crowded in recent months; even more competitors are scheduled to appear this year.
- ▶ Core compilers strive to simplify your design process.
- ▶ Validation models offer varying degrees of functional and timing comprehensiveness and tool-set portability.
- ▶ Embedded-FPGA providers strive to smoothly integrate their cores into your ASIC-development flow, but with divergent logic-cell approaches.
- ▶ Different opinions on co-verification leave you with many choices but no easy answers.
- ▶ Run your own evaluations to determine the timing- and pinout-locking robustness of each programmable-logic approach.

PLD or FPGA must adapt only minimally (or not at all) to comprehend on-chip IP. You need to specify configuration bits representing variables such as, for PCI, target-versus-initiator function, bus width, I/O-bus and memory-map addresses, vendor and device identifiers, wait states, burst lengths, and the like. This configuration data resides in the programmable-logic partition and fine-tunes the logic that the ASIC gates store. Ironically, nearly the same situation occurs with ASIC-housed complex microcontrollers and microprocessors and their associated "hard" peripherals. Vendors have already designed them; you just need to hook up your logic to them.

The vast middle ground between simple and complex cores requires a more involved approach. As an analogy, recall that some embedded-memory functions use logic or interconnect resources that the programmable logic—not the memory block itself—houses. You might need to construct FIFO status flags, specify a three-state internal bus for bidirectional capability, create a CAM arrangement, or

(continued on pg 106)

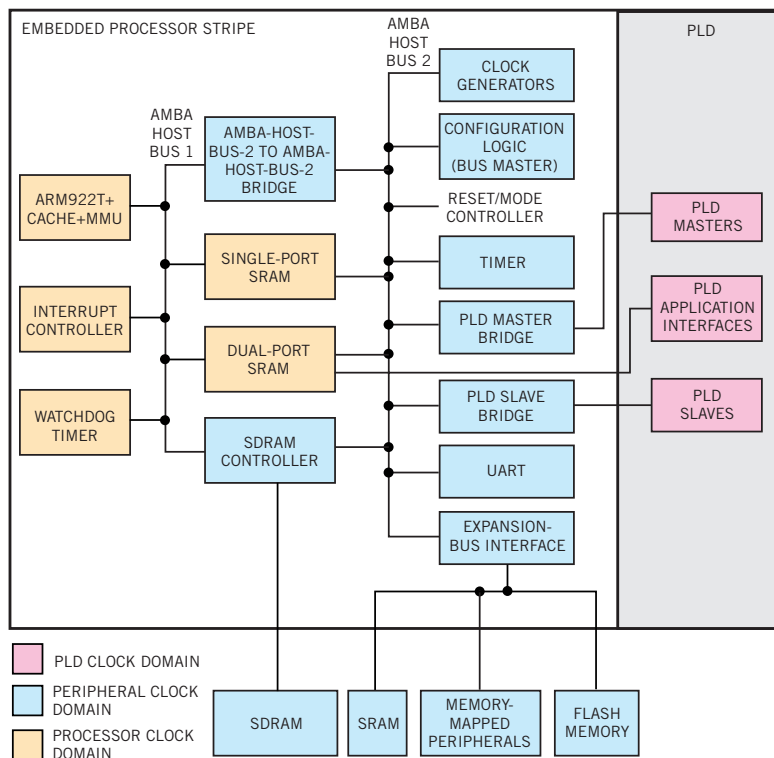


Figure 1

Both Altera's ARM- and MIPS-based hybrid chips incorporate multiple stages of AMBA interconnect.

CO-VERIFICATION CHOICES

Your system on a chip contains a processor running firmware, as well as dedicated-function logic residing in both ASIC and programmable gates. You need to debug them both. Wouldn't it be ideal if you could simultaneously validate both your hardware and software in a coordinated manner (Figure A)?

Atmel's FPSLIC Software SystemDesigner suite, which costs \$995 per year, includes an assortment of Mentor Graphics software: Leonardo for synthesis, ModelSim for verification, and the Seamless co-verification tool. The suite also includes Atmel's own AVR Studio for software debugging and Figaro IDS (Integrated Development System) programmable-logic tool set for macro generation, manual floorplanning, and automated placement and routing. C compilers come from IAR Systems and ImageCraft.

Key to the concept of co-verification is the virtual system prototype, which you can use to integrate your embedded software and hardware before committing to silicon. Atmel and Mentor Graphics believe that many of the design problems exposed during system integration aren't attributable to software or hardware alone, but to the interaction between the two. The companies

believe that you can benefit by simultaneously exercising the boot-ROM code, hardware diagnostics, device drivers, and RTOS to expose hardware/software interface errors, eliminate lengthy hardware-prototype iterations, and reduce overall system-integration time.

Seamless links the software and hardware-verification environments that you're already using. Logic outputs aren't just binary pulses on a screen or binary digits in a text file; they're hex codes that your software debugger can display, interpret, and interact with. Similarly, you need not create test vectors representing what the activity of the CPU bus feeding your logic *should* be; you can validate your logic using bit patterns that the software simulator running your real-life code generates. Seamless comprises four main elements:

- the co-simulation kernel, which controls communications between the software and the hardware portions of a co-simulation session;
- the instruction-set simulator, which performs the software portion of the co-simulation session, executing your target system's exact application code pro-

duced by cross assemblers and compilers;

- the bus-interface model, which provides the interface between the processor and hardware and models the processor's peripherals and pin interface; and
- the debugger interface, which controls execution of the instruction-set simulator and performs single-stepping and other typical debugging functions.

Atmel's competitors don't dispute the conceptual value of co-verification, but they claim its real-life usefulness is significantly diminished when your custom logic resides in programmable logic. Co-verification makes sense when it can help you avoid a costly, time-consuming ASIC-mask tweak necessitated by an error that you overlooked during traditional, separate, hardware and software simulation.

In the programmable-logic-design world, though, new hardware is available for little to no incremental cost, a few hours of compilation, and few minutes of downloading time. So, you need not get "stuck" at the presilicon-validation stage. Instead, Atmel's competitors claim, you should use a JTAG- or other port-based

CPU debugger, along with an FPGA-housed internal-node logic analyzer (such as Actel's Silicon Explorer, Altera's SignalTap, or Xilinx's ChipScope). This combination can analyze thousands or millions of lines of code and associated hardware cycles in the time it would take to cosimulate a few hundred cycles before silicon.

Embedded-logic analyzers, though, consume both logic and routing resources for each monitored internal node, you can monitor only a handful of nodes, and changing the nodes you monitor sometimes requires a design recompilation. Also, there's no guarantee (in spite of the vendors' claims of abundant logic and routing overhead) that your design's performance and pinout characteristics won't change when you remove the embedded-logic analyzer. And you will want to remove it, because, otherwise, you'll be taking to production a system containing a PLD or FPGA that's too big and too expensive.

Most silicon vendors offer families of chips, both 100% programmable and hybrid, with varying sizes of memory and FPGA or PLD resources but in identical packages and pinouts. This compatibility may preclude you from redesigning your board when you move to an analyzer-excluded chip. But still there's no guarantee that the production device will act like the one you debugged. In summary, when you need to monitor hundreds or thousands of internal signals, but only over a few hundred or thousand simulation cycles, presilicon co-verification might make sense. On the other hand, when you need to run millions of validation cycles but monitor only a few dozen signals, consider skipping the presilicon co-verification step and taking the embedded-logic-analyzer gamble.

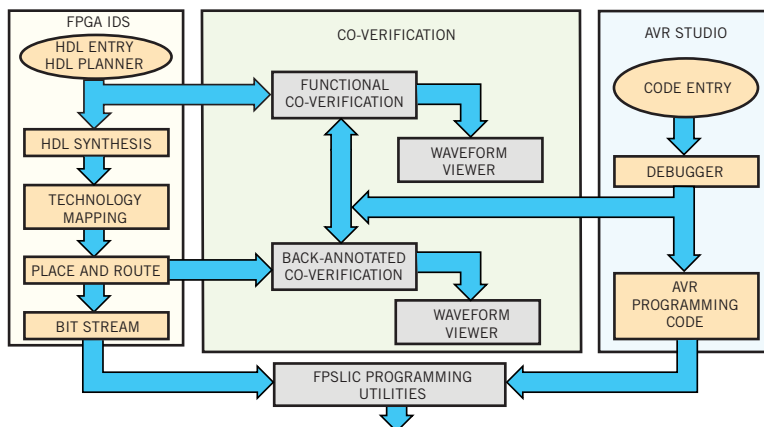


Figure A

Mentor Graphics powers the co-verification capability of Atmel's FPSLIC tool set, enabling you to simultaneously debug hardware and software.

add a register bank to define that the memory operates synchronously. Although you can design this logic yourself, it's often easier and faster to rely on a compiler from a vendor that lets you specify parameters and then automatically creates the configuration for you (**Reference 4**). The result is a combination of memory arrays plus fixed-location circuits in programmable logic. Their presence leaves fewer logic and routing resources available for the rest of your circuitry.

Now consider, for example, the multiply/accumulators in QuickLogic's Quick-DSPs or the multipliers in Xilinx's Virtex-II architecture. For these arithmetic circuits to implement digital-signal-processing functions, you must design a state machine to drive their various control and data inputs and outputs. Again, you could design that state machine yourself, but you might prefer to let the core compiler do the designing for you.

Theoretically, you need only to tell the core compiler that, for example, you'd like to create a fourth-order, lowpass Butterworth filter or, in a more elaborate example, an MP3 decoder. Some of the resulting logic and memory resides in ASIC gates, and the user-customized remain-

der targets programmable logic. Simplifying this process is at the heart of Xilinx's partnership with The Mathworks. ASIC-housed embedded functions aren't necessarily restricted to purely logic circuits, either. Consider, for example, the high analog content in some of Lucent's hybrid chips, or the ultra-high-speed I/O buffers that both QuickLogic and Xilinx have licensed from Conexant.

VALIDATION VARIABLES

Programmable-logic designers have a notorious reputation for following a "blow-and-go" philosophy: Compile the design but don't bother simulating it; stick it into a chip; and, if it doesn't work, scratch your head and repeat the process. That reputation, born in an era of low-density PALS, is becoming increasingly obsolete in the age of multimillion-gate FPGAs, multihundred-macrocell CPLDs and, particularly, programmable-logic/ASIC hybrids. Simulating only the logic you design with test vectors representing how you *think* the ASIC-housed cores will function is insufficient. You need to functionally simulate the entire chip to comprehend how the hard IP *really* works and run static timing analyses to ensure the design meets or exceeds target

ACRONYMS

ALU: arithmetic-logic unit
AMBA: Advanced Microcontroller Bus Architecture
BIST: built-in self test
CAM: content-addressable memory
CDR: clock and data recovery
DLL: digital delay-locked loop
FPSLIC: field-programmable system-level IC
HDL: hardware-description language
IP: intellectual property (cores)
JTAG: Joint Test Action Group
LUT: look-up table
PAL: programmable-array logic
PLA: programmable-logic array
PLD: programmable-logic device (product-term-based)
PLL: phase-locked loop
RTL: register-transfer level
SERDES: serializer/deserializer
SONET: synchronous-optical network
VHDL: very high-speed hardware-description language

performance (see **sidebar** "Co-verification choices").

In attempting to balance your desire for in-depth knowledge of the core's op-

PINOUT AND PERFORMANCE PROMISES

Whenever you combine in one chip functions that formerly resided in separate devices, your ability to perform timing and pinout locks of the portion of the design housed in programmable logic becomes critical. Your design inside the programmable logic may change during system development, but you usually can't alter the design's timing requirements. And you can no longer compensate, via a board-level interconnect redesign, for a CPLD or FPGA pinout alteration that changing circuit-level timing and inflexible chip-level timing force you to make.

If you're designing both the logic housed in the ASIC and the circuits placed in programmable logic, pinout- and performance-locking capability

becomes even more important. One of the reasons you incorporate embedded programmable logic, after all, is because it lets you perform tape-out and ship the chip to your ASIC vendor or foundry before you finish designing the portion of the design in PLD or FPGA gates.

Routing within the programmable logic may be flexible, but the interconnect between ASIC and programmable logic, once frozen in silicon, is unchangeable. And for both custom ASICs and hybrid standard products, you can no longer alter the board design once you ship it to a customer. This fact is important if you want to upgrade the circuits housed in programmable logic to fix bugs or add or alter features after the system's in the field.

All vendors will tell you not to worry, that their architectures have so much extra on-chip logic and routing resources that locking pinout and performance is not a problem. Actel's documentation, for example, claims that experiments show only about a 10% change in delays when pins are fixed randomly over allowing the software to fix pins. It also claims that densities more than 80% of three-input LUTs will eventually be routinely achievable. In the initial phase, Actel claims to be currently achieving 70% three-input LUT usage without difficulty and that this result is true regardless of the nature of the design.

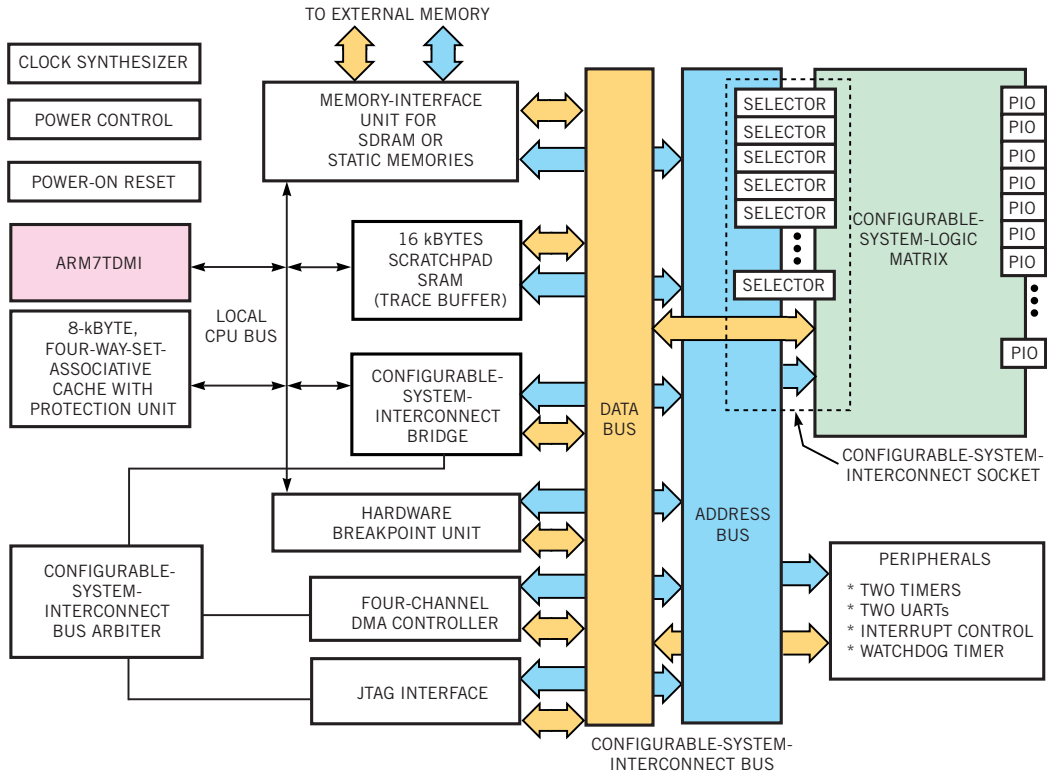
Actel's and other vendors' claims are undoubtedly true but only under their chosen subsets of all possible designs run dur-

ing fine-tuning of both silicon and software. Choose different circuits, a different design-entry method or mix of methods, or different tool flow, and your results might be less positive. The hybrid-standard-product vendors all offer evaluation boards. The embedded-programmable-logic suppliers all plan fully programmable bond-out chips for your assessments. Use them—on your designs or mine (**Reference A**). Good luck! And let me know how your evaluations turn out.

REFERENCE

A. Dipert, Brian, "Lies, damn lies, and benchmarks: The race for the truth is on," *EDN*, May 27, 1999, pg 54.

Figure 2



Triscend's signal-routing approach memory-maps every programmable logic array node into the embedded-processor address space.

eration with vendors' desire to protect their cores from illegal duplication, you'll find that the vendors' simulation models span a range of capabilities and formats (Reference 5). Minimally, you should at least get a bus-functional model, which mimics how the core works but doesn't give away actual circuit-implementation details. The trick, of course, is that the vendor supplying the core must ensure that the simplified model matches the actual core's functions. This task is particularly challenging for complex, superscalar, variable-latency microprocessors, for example. Also, digging into and understanding the circuitry that your logic is interfacing with can sometimes help you debug *your* circuits.

At the opposite end of the spectrum, some companies quietly provide full RTL HDL models. Expect a thick stack of legal documentation to accompany the code to help ensure that you don't misbehave and either incorporate elements of the core into your designs without license or transfer them to others. As a midpoint alternative, some core developers have turned to either encrypted, secure HDL, such as Cadence's NC-Sim, or precompiled models, such as the Synop-

sys-championed SmartModels. Xilinx, for example, won't begin shipping samples of its first 0.13-micron PowerPC 405-D5 and Virtex-II-based hybrid chips until late summer, but the company will start sending out development systems by the end of the first quarter, containing SmartModels for the CPU core, CoreConnect bus, and ASIC-housed peripherals.

Altera plans to use the ARM-developed AMBA bus for both its ARM922- and its MIPS32 4Kc-based hybrid-device families, which are built on an Apex 20K programmable-logic foundation. In this case, though, the vendor plans to provide nothing more elaborate than behavioral models for the CPUs and related hard IP. Figure 1's depiction of the devices' architectural block diagram might shed some light on the reasons behind Altera's actions (aside from ARM and MIPS' understandable paranoia about the potential release of their cores' source code into the public domain).

Neither your logic in the programmable-logic partition nor the Altera-supplied soft IP also potentially stored in that partition directly interfaces to the CPU core. Instead, the hookup is through

master-and-slave bridges and dual-port SRAM (different from the embedded-array-block memory that the Apex 20K array contains). Altera officials feel that accurately modeled bridges and memory, along with accurate bus-functional models of the remainder of the processor stripe, provide sufficient capability for any presilicon-simulation requirement. Triscend has adopted a similar approach for the interconnect, which the company claims is highly predictable, that links the programmable-logic arrays in Triscend's E5 and A7 devices to the respective 8051 and ARM7TMI cores (Figure 2).

In Triscend's approach, most micro-processor peripherals reside in programmable logic for maximum flexibility. To accurately simulate the chip, you must obtain models not only of the ASIC-housed circuitry but also of the FPGA-based peripherals in the vendor's library. Altera and Xilinx plan a more balanced mix of ASIC- and FPGA-housed peripherals; Atmel's current FPSLIC devices reserve the entire programmable-logic array for your use. QuickLogic isn't yet releasing details of its MIPS32 4Kc- and MIPS64 5Kc-based hybrid chips, which

TABLE 1—HYBRID-STANDARD-PRODUCT VENDORS, PRODUCTS, KEY FEATURES, AND REFERENCES

Vendor	Product	Key ASIC-embedded features	Issue dates and URLs for EDN write-ups
Altera	Excalibur	ARM and MIPS CPU cores and peripherals, AMBA interconnect, single- and dual-ported RAM blocks	Nov 30, 2000, pg 30 (www.ednmag.com/ednmag/reg/2000/11232000/leadingedge.htm#24le14)
Atmel	FPSLIC	AVR CPU core and peripherals, program and data SRAM	July 20, 2000, pg 107 (www.ednmag.com/ednmag/reg/2000/07202000/15tt.htm)
Cypress	PSI	SERDES core with SONET, Infiniband, Fibre Channel, and Gigabit Ethernet I/O options	
Lucent	OR3TP12 and OR3LP26B	Embedded 64-bit, 66-MHz PCI core	Aug 17, 2000, pg 68 (www.ednmag.com/ednmag/reg/2000/08172000/17cs_sub04.htm)
	ORLI10G	10-Gbps line interface	
	ORT4622	Four-channel, 622-Mbps, full-duplex synchronous interface with built-in CDR	Aug 17, 2000, pg 68 (www.ednmag.com/ednmag/reg/2000/08172000/17cs_sub04.htm)
	ORT82G5 and ORT8850	Eight-channel; 1.25-, 2.5-, 3.125-Gbps transceiver Eight-channel, 850-Mbps, full-duplex synchronous interface with built-in CDR	Aug 17, 2000, pg 68 (www.ednmag.com/ednmag/reg/2000/08172000/17cs_sub04.htm)
QuickLogic	QuickDSP	User-configurable multiply/accumulators	Jan 20, 2000, pg 28, (www.ednmag.com/ednmag/reg/2000/01202000/leadingedge.htm#02le13)
	QuickFC	Fibre Channel encoder/decoder	Aug 17, 2000, pg 70, (www.ednmag.com/ednmag/reg/2000/08172000/17cs_sub05.htm)
	QuickPCI	32- and 64-bit PCI cores, running as fast as 66 MHz	Aug 17, 2000, pg 70 (www.ednmag.com/ednmag/reg/2000/08172000/17cs_sub05.htm)
	QuickSD	SERDES core	Oct 12, 2000, pg 22, (www.ednmag.com/ednmag/reg/2000/10122000/leadingedge.htm#21le08)
Xilinx	Virtex-II Pro	Embedded PowerPC core and peripherals, Core-Connect interconnect, SkyRail 3.125 Gbps I/O buffers	

are due out in the second half of this year. Similarly, Lucent Technologies hasn't yet revealed its embedded-processor plans, although the AMBA bus in the company's Orca 4-based hybrids and the company's ARM license are telling clues.

All of the aforementioned hybrid chips, plus most of the others in Table 1, use conventional multiplexer- and LUT-based FPGA structures to implement the programmable logic. One exception is Cypress, which plans to use SRAM-based PLD structures (that is, product-term logic feeding registers) from the Delta 39K for its PSI (Programmable Serial Interface) hybrid chips.

These chips place SERDES and SONET, Infiniband, Fibre Channel, or Gigabit Ethernet high-speed interfaces in the ASIC partition. The choice of PLD logic is peculiar, particularly at the several-hundred- to several-thousand-macrocell-count levels that Cypress plans to embed in its PSIs.

Product-term logic is ideal for implementing address decoders and state machines as well as for translating between state machines, data and clock synchronization, handshake protocols and exception handling, control semaphores, and resource management. Register-rich FPGA logic, on the other hand, is ideal for the types of datapath functions you

commonly find in the interface-translation applications that Cypress hopes to crack: arithmetic and logic transforms, synchronous pipelines, and similar circuits. And, employing design techniques such as one-hot encoding, you can use FPGA logic to also implement, though sometimes inefficiently, many of the same circuits that PLDs support.

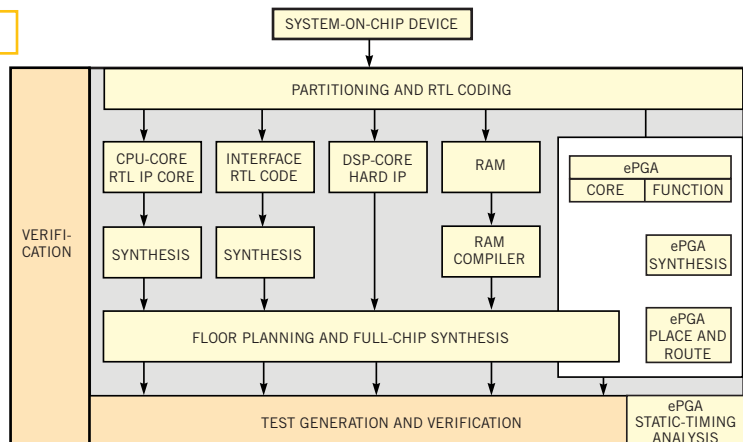
A MORE FLEXIBLE ASIC

What if your silicon-design platform of choice isn't programmable logic but

standard-cell ASIC? Even the largest FPGA might be too small, too expensive, or too high-power for your needs. Hundred-megahertz processor cores are beyond the capabilities of even hand-tuned FPGA layouts, regardless of what the vendors might be telling you. Or you might require embedded DRAM or mixed-signal capability that off-the-shelf programmable-logic devices don't offer.

All news is not good news with ASICs, though. Mess up your design, and you're potentially staring at hundreds of thou-

Figure 3



NOTE: ePGA=EMBEDDED PROGRAMMABLE ARRAY.

Actel's embedded FPGA design-and-simulation flow incorporates a programmable-logic-tuned synthesis compiler.

sands or even millions of dollars in additional mask-set charges. Throughput from design completion to first silicon back from your ASIC partner takes weeks or months, versus minutes or hours for programmable-logic-based equivalents. ASIC-housed designs aren't amenable to in-the-field upgrades to fix bugs or add features. And you can't easily customize their hardware to enable one silicon platform to service multiple applications.

For these reasons, you might want to add embedded-FPGA capability to your ASIC. And vendors such as Actel, Adaptive Silicon, Atmel, Lucent Technologies, and Systolix are happy to help you do it.

The conventional ASIC design-and-validation flow is much more complex than in the previously described standard-product approach targeting FPGAs (Figure 3). The preferred synthesis compiler is an ASIC-optimized product such as Synopsys' Design Compiler. This distinction is important, because this compiler is optimized for the fine-grained ASIC structure, not the coarser programmable-logic cell.

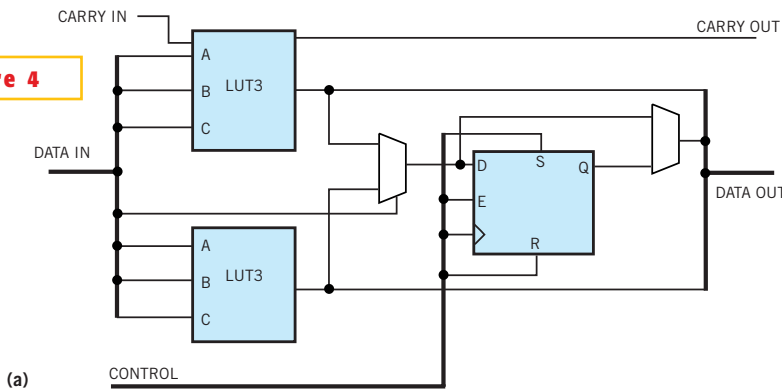
Actel therefore recommends that you obtain and use a separate programmable-logic-optimized synthesis compiler for the portion of your design that goes into its VariCore embedded FPGAs. Ac-

tel has chosen for its technology a conventional logic-cell approach, consisting of dual three-input LUTs feeding a register (Figure 4). Reflecting the fact that FPGA compilers are tuned for the more common four-input LUT structure, the Actel logic cell can optionally but less efficiently group the two three-input LUTs into one four-LUT equivalent.

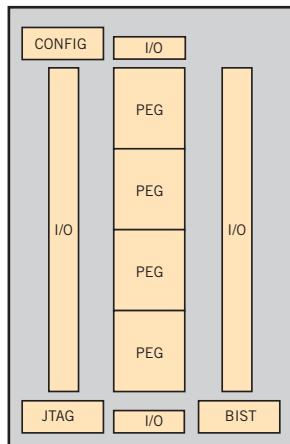
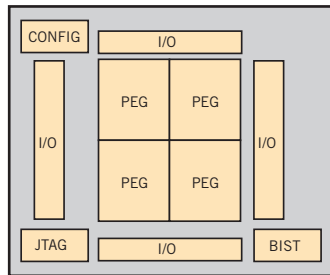
A review of Actel's VariCore data sheet reveals some curious omissions: no hardware multipliers or other arithmetic-optimized structures, no three-state buffers or bidirectional buses, and no partial-reconfiguration capability. The embedded FPGA arrangement does, however, include built-in vertical-carry chains that optionally connect to one of the three LUT inputs. Each FG (functional group) contains four pairs of three-input LUTs and four registers. The registers share common control inputs (enable, preset, reset, and clock).

The next level of hierarchy is the PEG (programmable-embedded-gate) array, an 8x8 matrix of FGs that Actel estimates represents a multiple-design average of 2500 ASIC gates, excluding RAM. Some FPGA vendors use RAM to artificially boost their system-gate counts (Reference 6). (Actel, along with partners Chartered, TSMC, and UMC, initially

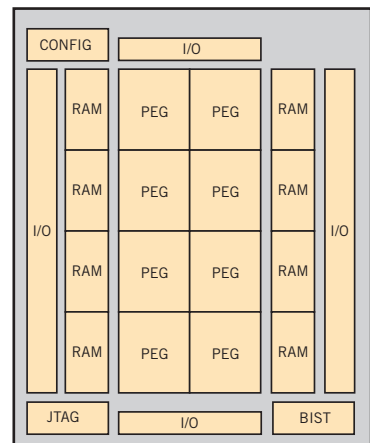
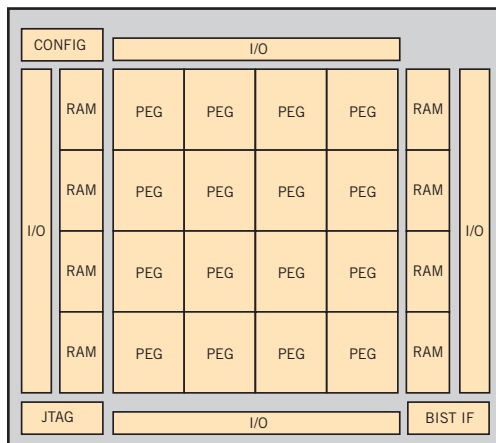
Figure 4



(a)



(b)



The elemental Actel logic cell (a) hierarchically multiplies to form programmable embedded-gate arrays of various shapes and sizes (b).

plans to offer 2×1, 2×2, 4×2, and 4×4 PEG structures on 0.18-micron process technologies (Table 2.) You can orient these multi-PEG arrangements in square, rectangular, and even L-shaped structures.

According to Actel, the ideal VariCore shape is square, which minimizes the internal delays within the FPGA core. However, in some applications, a different-shaped core provides a more efficient implementation at the physical level. Also, each of the PEG blocks with external edges provides 48 inputs and outputs per horizontal edge and 32 inputs and outputs per vertical edge. A 2×2 PEG array offers a maximum of 640 inputs and outputs, whereas a 4×1 equivalent contains 704 inputs and outputs. The place-and-route software is parameterized to support multiple PEG matrix sizes and orientations, thereby not forcing the HDL source code that you develop at the beginning of the design process to explicitly specify the final physical layout.

The architecture arranges both the logic structures and the routing between them in a hierarchical manner. Actel believes that this approach offers flexibility, predictability in timing and logic usage, and faster place-and-route times (see

sidebar “Pinout and performance promises”). In fact, the vendor claims that a design with 70% three-LUT usage on a 4×4 PEG FPGA completely compiles on a 500-MHz PC in about nine minutes and that designs routinely and easily use close to 100% of the FGs. Each multi-PEG array also contains JTAG circuitry and a BIST interface (but not a BIST controller). Because of the fixed-silicon cost of this overhead circuitry, Actel doesn’t anticipate offering single-PEG arrays, at least at the 0.18-micron process generation. PEG arrays can also include optional cascadable RAM blocks of 9 kbits each with both 9- and 18-bit data-interface options and built-in FIFO flag logic. You might use these RAM structures, for example, as bridges between ASIC and FPGA logic running at different bus widths and clock speeds.

DO THE MATH

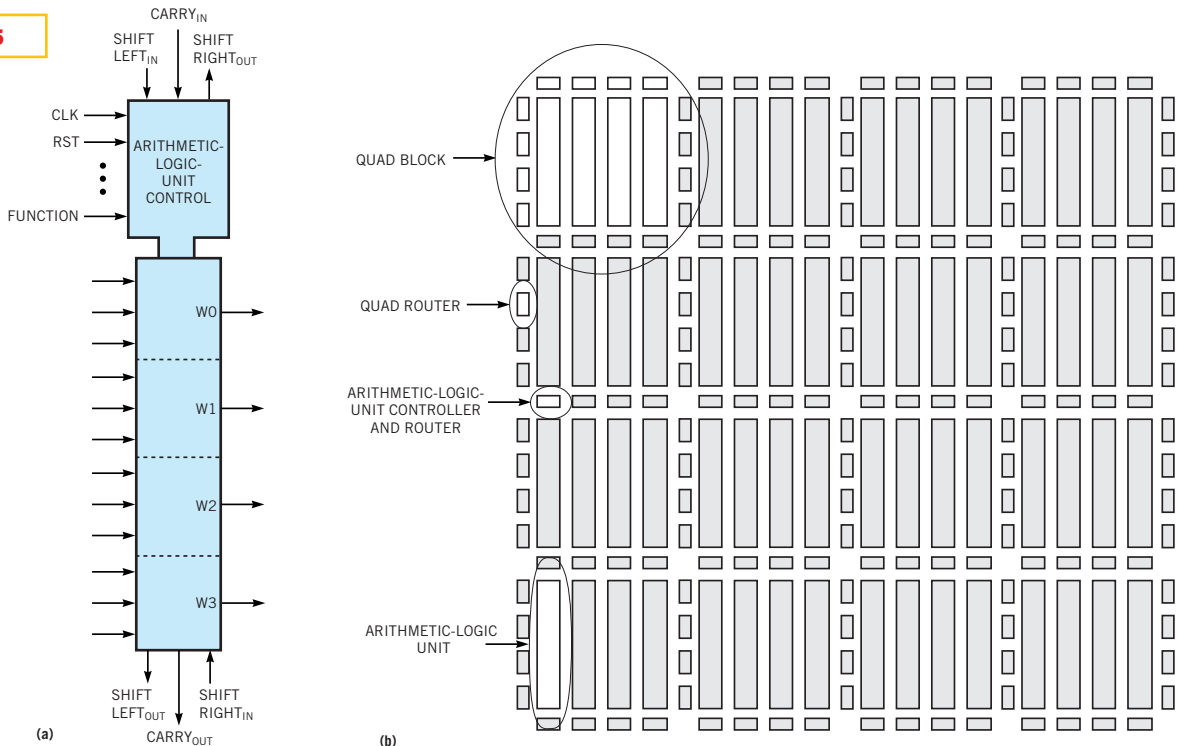
Actel’s primary competitor at the moment, Adaptive Silicon, has chosen a completely different programmable-logic structure-and-synthesis flow approach. The company uses an ALU-based logic cell whose heritage extends back to the days when some of the company’s founders were employed by National

Semiconductor and developed the academically interesting, but economically unprofitable, CLay (Configurable Logic Array) and NAPA (National Adaptive Processing Architecture) FPGAs. Other vendors that have attempted to enter the programmable-logic market with arithmetic-logic-tuned devices include:

- Concurrent Logic, funded and second-sourced by National Semiconductor;
- IBM, a Concurrent licensee;
- Pilkington Microelectronics, acquired by Motorola for its quickly canceled ColdFire-based hybrid-product line; and
- Toshiba, another Pilkington licensee and ultimately another nonplayer in the FPGA business.

As you can see, arithmetic-tuned devices don’t have a stellar track record. National Semiconductor’s architecture became the foundation of Atmel’s AT6000 devices, but Atmel has turned to a more conventional LUT structure for the follow-on AT40K product line, the basis of FPSLIC. Adaptive Silicon officials believe, though, that with an architecture tuned by years of silicon and software experimentation, the company now has a winner. The fundamental building block of

Figure 5



Adaptive Silicon’s ALU-based logic cell (a) is the basis of the vendor’s embedded FPGA arrays (b).

Adaptive's MSA (multiscale array) is an enhanced 74LS181 4-bit ALU with a modular carry-look-ahead scheme and optional output registering (Figure 5). Four ALUs group to form a Quad Block, connected to other Quad Blocks by signal routing and ALU-control resources, and 16 Quad Blocks form a Hex Block.

Adaptive Silicon estimates that each Hex Block represents an average of 1500 ASIC gates. The company points out that the logic cell can also implement more generic LUT-like functions via both gate- and ALU-level mapping capability. The interface between programmable- and standard-cell partitions comprises a PLC (programmable-logic-core) control structure in the programmable partition (along with an application-circuit interface) and a matching PLC adapter in the ASIC partition. The vendor has 0.18-micron test chips already in-house from initial partner and investor LSI Logic, and chips from TSMC are due back from fabrication next month.

Adaptive Silicon's silicon approach may be unconventional, but the company's design flow is ASIC-designer-friendly (Figure 6). The synthesis tool set for the programmable-logic partitions is the same one you use for standard-cell-targeted logic: Synopsys' Design Compiler or an equivalent. Adaptive Silicon relies on DesignWare libraries that it developed to perform the necessary translation and architectural optimization of VHDL or Verilog source code. The registers in each logic cell, with scan chains throughout the array, aid in silicon verification. Adaptive Silicon and its ASIC partners tailor the PLC adapter for each application, with a soft macro that manages the

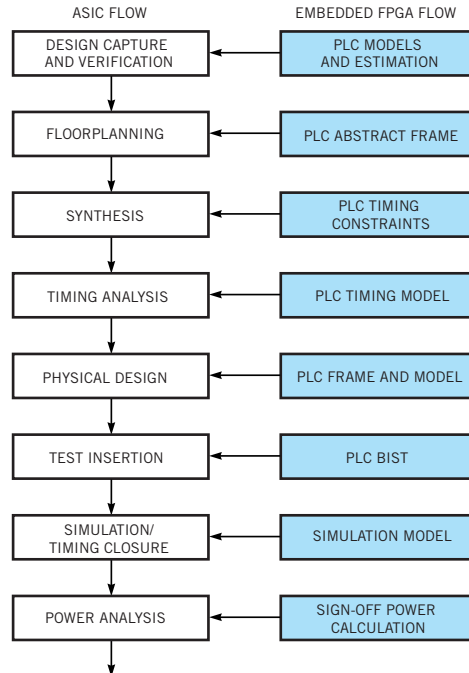


Figure 6 Embedded-FPGA providers strive to support each step in the overall ASIC-design and -validation process (courtesy Adaptive Silicon).

programming and test interfaces, a five-pin test-access port, and multiplexed test and data I/O signals. The programmable-logic core comprehends full-chip scan BIST to ensure programming integrity, quiescent power testing, and full-speed testing.

Although their silicon and synthesis techniques may differ, Actel and Adaptive Silicon agree on a few things. First off, both companies strive to seamlessly integrate their embedded FPGA partitions into your chip's layout, functional verification, and static-timing-analysis flows, with GDS II floorplanning layout frames, SDF timing models, and the like. Adaptive Silicon supports partner LSI Logic's CoreWare design flow, including the layout-versus-schematic, transistor-level check before tape-out, even though a dy-

namically configured programmable-logic partition doesn't use many of the available transistors.

Both vendors also stress the full compatibility of their technologies with standard CMOS processes. Adaptive Silicon prefers to employ a five-layer metal process, although a four-layer version is acceptable. Neither vendor wants its embedded FPGA to be the guilty party that makes the chip too expensive. And, although Actel makes aggressive claims about the superior cost- and power-effectiveness of its VariCores over conventional FPGAs (half the die area, and one-third the power consumption), you should carefully analyze whether those assertions apply to your design.

Actel's claims of die-size savings come from the use of three-input LUTs instead of conventional four-input LUTs. The prediction comes true only if your design and compilation tools efficiently map to the three-input LUTs and inefficiently translate to four-input LUTs. Conventional FPGAs also include peripheral logic (I/O buffers, DLLs, or PLLs, among others) that embedded FPGAs don't include. And speaking of I/O buffers, much of the powers savings that embedded FPGAs claim come from the absence of power-hungry I/O buffers in an embedded configuration relying exclusively on on-chip interconnect.

THE REST OF THE PACK

Actel and Adaptive Silicon may be today's most visible embedded-programmable-logic advocates, but they aren't the only games in town. Tiny UK IP vendor Systolix, comprising ex-Pilkington employees, recently scored a coup when Analog Devices unveiled the AD7725 16-

TABLE 2—ACTEL VARICORE FAMILY MEMBERS

Device	PEG array	ASIC gates	Functional groups	Registers	Logic cells	Available I/O	RAM blocks	RAM bits
V18L2x1	2x1	5000	128	512	1024	448	0	0
V18L2x2	2x2	10,000	256	1024	2048	640	0	0
V18L4x1	4x1	10,000	256	1024	2048	704	0	0
V18L4x2	4x2	20,000	512	2048	4096	896	0	0
V18L4x4	4x4	40,000	1024	4096	8192	1280	0	0
V18L2x2R	2x2	10,000	256	1024	2048	640	Four	36,864
V18L4x2R	4x2	20,000	512	2048	4096	896	Eight	73,728
V18L4x4R	4x4	40,000	1024	4096	8192	1280	Eight	73,728

bit Delta-Sigma ADC containing a Systolix-developed PulseDSP programmable ALU array. The ASIC divisions of both Atmel and Lucent announced their respective embedded-FPGA capabilities months or years ago, although when questioned, neither vendor can provide much tangible information beyond pointing to their hybrid standard products as proof of concept.

PLA vendor ICT hopes to move its programmable AND/OR PAL technology to the world of IP cores. The company points out that the portions of a design most likely to change, such as state machines, control-signal translation, and address decoders, are ideal applications for PLAs. Unlike Cypress, ICT isn't trying to convince customers to embed large chunks of product-term-based logic on their chips. Such an approach would quickly become cost-pro-

hibitive, particularly in a global-routing-matrix structure. Instead, the company wants you to scatter small PLAs across the ASIC.

Like Cypress, however, ICT is moving away from an EEPROM-based technology toward an approach employing SRAM to store configuration-bit values. Embracing SRAM in this manner enables the company to leverage the most advanced processes at multiple ASIC vendors and foundries. ICT, along with partner Faraday, is designing a proof-of-concept hybrid using discrete programmable-logic devices. The first embedded PLA bond-out chips, targeting UMC's 0.15-micron process, are scheduled for tape-out in March. □



You can reach Technical Editor Brian Dipert at 1-916-454-5242, fax 1-916-454-5101, e-mail bdipert@pacbell.net.

REFERENCES

1. Dipert, Brian, "The best (or worst?) of both worlds," *EDN*, Nov 11, 1999, pg 139.

2. Dipert, Brian, "The end of the road?" *EDN*, May 13, 1999, pg 54.

3. Selburn, Jordan, "Application-specific programmable products: the coming revolution in system integration," *Dataquest Perspective*, Sept 21, 1998.

4. Dipert, Brian, "Getting a handle on HDLs," *EDN*, May 7, 1998, pg 71.

5. Dipert, Brian, "Cunning circuits confound crooks," *EDN*, Oct 12, 2000, pg 103.

6. Dipert, Brian, "Counting on gate counts? Don't count on it," *EDN*, Aug 3, 1998, pg 52.

7. Levy, Markus, "Processors drive (or dive) into programmable-logic devices," *EDN*, July 20, 2000, pg 107.

ACKNOWLEDGMENTS

Special thanks to Atmel FPGA Marketing and Applications Director Joel Rosenberg for the arithmetic-tuned-programmable-logic history refresher.

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, go to our information-request page at www.rscahners.ims.ca/ednmag/. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Actel
1-408-739-1010
www.actel.com
Enter No. 396

Integrated Circuit Technology
1-408-434-0678
www.ictpld.com
Enter No. 401

Triscend
1-650-968-8668
www.triscend.com
Enter No. 406

ImageCraft
www.imagecraft.com
Mathworks
www.mathworks.com

Adaptive Silicon
1-408-335-2700
www.adaptivesilicon.com
Enter No. 397

LSI Logic
1-408-433-8000
www.lsil.com
Enter No. 402

Xilinx
1-408-559-7778
www.xilinx.com
Enter No. 407

Mentor Graphics
www.mentor.com
MIPS
www.mips.com

Altera
1-408-544-7000
www.altera.com
Enter No. 398

Lucent Technologies (Agere Systems)
1-908-582-8500
www.lucent.com
Enter No. 403

OTHER COMPANIES MENTIONED IN THIS ARTICLE
Analog Devices
www.analog.com

Motorola
www.motorola.com
National Semiconductor
www.national.com

Atmel
1-408-441-0311
www.atmel.com
Enter No. 399

QuickLogic
1-408-990-4000
www.quicklogic.com
Enter No. 404

ARM
www.arm.com
Cadence
www.cadence.com

Synopsys
www.synopsys.com
Toshiba
www.toshiba.com

Cypress Semiconductor
1-408-943-2600
www.cypress.com
Enter No. 400

Systolix
+ 44-151-242 0600
www.systolix.co.uk
Enter No. 405

Chartered
www.csminc.com
Conexant
www.conexant.com

TSMC
www.tsmc.com
UMC
www.umc.com

Faraday
www.faraday-usa.com
IAR Systems
www.iar.com
IBM
www.ibm.com

SUPER INFO NUMBER
For more information on the products available from all of the vendors listed in this box, enter No. 408 at www.rscahners.ims.ca/ednmag/.