FPGAs Dress Up for Expanded System Roles

At the chip, board and system levels, designers are making innovative use of FPGAs for complex signal processing, security and interfacing apps.

by Jeff Child, Senior Editor

Once relegated to mundane roles as glue logic—linking the CPUs, memory and I/O in computer-based systems—field-programmable gate arrays are moving into prime time. Today's leading FPGA products boast multi-million gate counts, I/O pins in the high hundreds and true system chip-level functionality including embedded CPUs. As they continue to bulk up in all these directions, they're evolving into subsystems in their own right. Bringing that trend to its full potential involves more than just the innovations of the FPGA vendors themselves. OEM board and system-level companies also play a part in placing FPGAs into new system roles.

System-Level Chips

At the chip level, the two leading FPGA vendors, Xilinx and Altera continue to leapfrog one another in gate counts and new features. Exemplifying the complete system capabilities of today's FPGA technology, the latest Xilinx offering is its Virtex-II Pro Platform FPGA (Figure 1), which embeds up to four IBM PowerPC 405 processors on a chip. Each PowerPC runs at 300+ MHz, delivering 420 Dhrystone Mips. The device supports 3.125 Gbit/s transceivers on chip, enabling it to implement 10 Gigabit Ethernet, PCI Express, RapidIO and SerialATA interfaces.

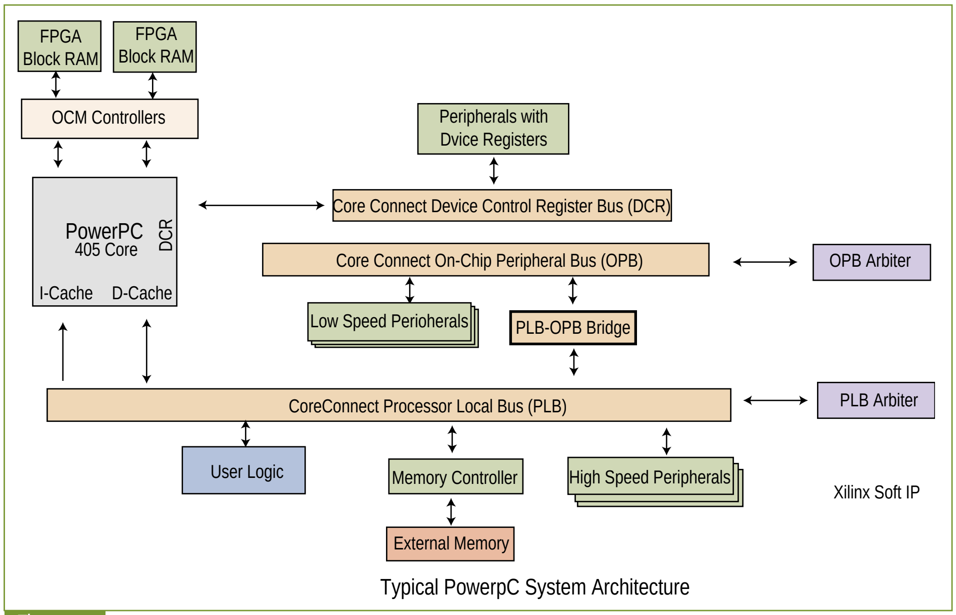


Figure 1 The Xilinx Virtex-II Pro Platform FPGA embeds up to four IBM PowerPC 405 processors on chip. Each PowerPC runs at 300+ MHz delivering 420 Dhrystone Mips.

Designers can add a variety of Xilinx Soft IP to implement various functions, including IBM's CoreConnect bus technology.

For its part, Altera brought its Stratix family into full production at the end of last year. Stratix devices are specifically designed to address bandwidth-intensive designs. They offer up to 114,140 logic elements, 10 Mbits of embedded memory and optimized DSP blocks and high performance. The devices feature LVDS I/O support capable of 840 Mbit/s performance. The Stratix architecture has also been designed to work with Altera's Nios embedded processor. The architectural features of Stratix devices boost the performance of the Nios soft embedded core processor to over 125 MHz.

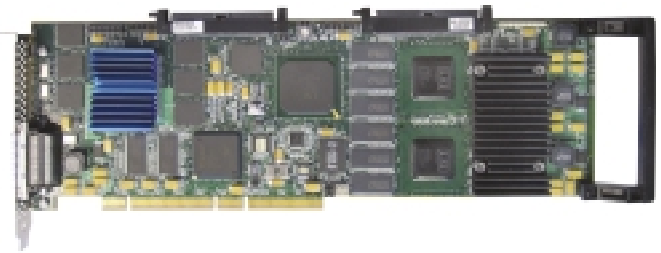


Figure 2

Mercury's VantageRT PCI board consists of a PCI form-factor module with one FPGA compute node and two PowerPC processors connected to a RACE++ crossbar. Using RACE++ Interlink modules, the modules can be configured with other members of the VantageRT family. With the board, Mercury was able to accelerate the algorithm in a FPGA to run 50 times faster than it could run on a G4 PowerPC processor.

Meanwhile other FPGA vendors are finding other avenues of innovation beyond speed and density. Quicklogic's recent Eclipse-II FPGA product family aims at ultra-low power. The architecture features dedicated SRAM blocks, flexible clock architecture and ultra-low power consumption of 250 µA standby current. Based on the company's patented ViaLink interconnect scheme the family is aimed at developers of mobile, portable, wireless and hand-held systems with a feature-rich alternative to CPLDs and ASICs.

With demand for security on the rise, it's conceivable that every mobile communicator, server and Internet-enabled appliance in the worldwide communications infrastructure will embed high-performance encryption technology in some form or other. Leveraging its experience in design-secure FPGAs, Actel made available last November new Advanced Encryption Standard (AES), Data Encryption Standard (DES) and triple DES intellectual property (IP) cores optimized for its nonvolatile Axcelerator, ProASIC, ProASICPLUS, RTSX-S and SX-A field-programmable gate array (FPGA) architectures. The cores are certified by the National Institute of Standards and Technology (NIST).

In the board-level realm, companies are finding expanded roles for FPGAs. According to Rodger Hosking, vice president at Pentek, his company began using FPGAs for more than simple interface logic three years ago when FPGAs with 300 to 600-million gate capacities emerged. "Looking at those larger devices, we realized that the FPGAs were sitting right in the middle of the dataflow path on our boards, in many cases handling the data. It only made sense then to do something besides just formatting and handling the data," said Hosking.

With that in mind, Pentek board designers began incorporating specific DSP functions on its boards using the Xilinx Virtex II family of FPGAs, which had hardware multipliers on-chip. Pentek then took that strategy a step further by crafting its own Fast Fourier Transform (FFT) engine implemented on an FPGA. This success led Pentek to start offering the FFT factory-installed intellectual property inside the FPGA as an option on Pentek's mezzanine card. Last month Pentek added a new set of five libraries to its GateFlow IP Core offering.

FPGAs Aim at Medical Imaging

Moving up to the system level, Mercury Computer Systems has had a history of using FPGAs to accelerate specialized applications. The company offers daughter cards that use FPGAs to accelerate 2D convolutions for medical imaging applications. Mercury also took another look at ways to use FPGA-implemented algorithms. In those efforts Mercury designers decided to implement a back-projection algorithm.

Back-projection takes 2D images and turns them into a three-dimensional dataset. Although back-projection is most applicable to the medical imaging market, Mercury says there's been interest from the military side in back-projection techniques applied in synthetic aperture radar. Mercury was able to accelerate the algorithm in an FPGA to run 50 times faster than it could run on a G4 PowerPC processor. The company has showed the technology in medical imaging circles where it's received much interest.

The idea was implemented into a product earlier this year in the VantageRT FCN (Figure 2), a new design for Mercury's VantageRT PCI product family. The system consists of a PCI form-factor module with one FPGA compute node and two PowerPC processors connected to a RACE++ crossbar. Using Mercury's RACE++ Interlink modules, VantageRT FCN modules can be configured with other members of the VantageRT family, including VantageRT 7410 dual-G4 and VantageRT HCD quad-G4 modules.

FPGAs Enable Custom Mezzanines

Even mezzanine products are making use of FPGA advances. There, the most interesting new trend is toward customizable I/O functions. Many industrial control applications inevitably have certain I/O requirements that are unique to their specific needs. In the past such function required users to either design a module themselves or have a vendor custom design

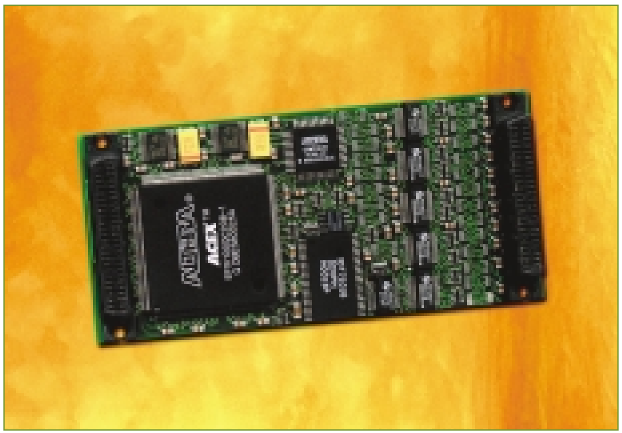


Figure 3

Acromag's IP 1K100 series of IndustryPack mezzanine modules let users develop and store their own instruction sets in the Altera EP1K100 FPGA for interfacing to VME and other form-factors. The FPGA can control up to 48 TTL or 24 EIA-485 I/O signals or a mix of both types. Application programs are downloaded through the IP bus directly into the FPGA.

a module for them. Thanks to advances in FPGAs, there's now a third option: system developers can buy mezzanine boards that marry a digital I/O link—TTL or differential—with an onboard programmable FPGA.

That trend isn't new, but it's been traditionally relegated to the high end. Decreasing FPGA costs have now brought that functionality into the $700 to $800 realm. That's starting to make the approach attractive to a mainstream segment of industrial automation system designers.

Along those lines, Acromag offers its IP 1K100 series (Figure 3). These IndustryPack mezzanine modules allow users to develop and store their own instruction sets in the Altera EP1K100 FPGA for interfacing to VME, CompactPCI and PCI computer systems. The EP1K100 FPGA can control up to 48 TTL or 24 EIA-485 I/O signals or a mix of both types. User application programs are downloaded through the IP bus directly into the FPGA.

A pre-programmed internal CPLD on the IP 1K100 facilitates initialization by acting as the bus controller during power-up and while the program is downloading. This bus controller is limited to functions necessary for power-up and downloading. After the program downloads, the FPGA takes control of the IP bus and the CPLD is disabled. Local static RAM (64K x 16) is controlled by the FPGA. Other features include a user-programmable PLL-based clock synthesizer and interval timer. ◢

Acromag
Wixom, MI.
(248) 624-1541.
[www.acromag.com].

Actel
Sunnyvale, CA.
(408) 739-1010.
[www.actel.com].

Altera
San Jose, CA.
(408) 544-7000.
[www.altera.com].

Mercury Computer Systems
Chelmsford, MA.
(978) 256-1300.
[www.mc.com].

Pentek
Upper Saddle River, NJ.
(201) 818-5900.
[www.pentek.com].

QuickLogic
Sunnyvale, CA.
(408) 990-4000
[www.quicklogic.com].

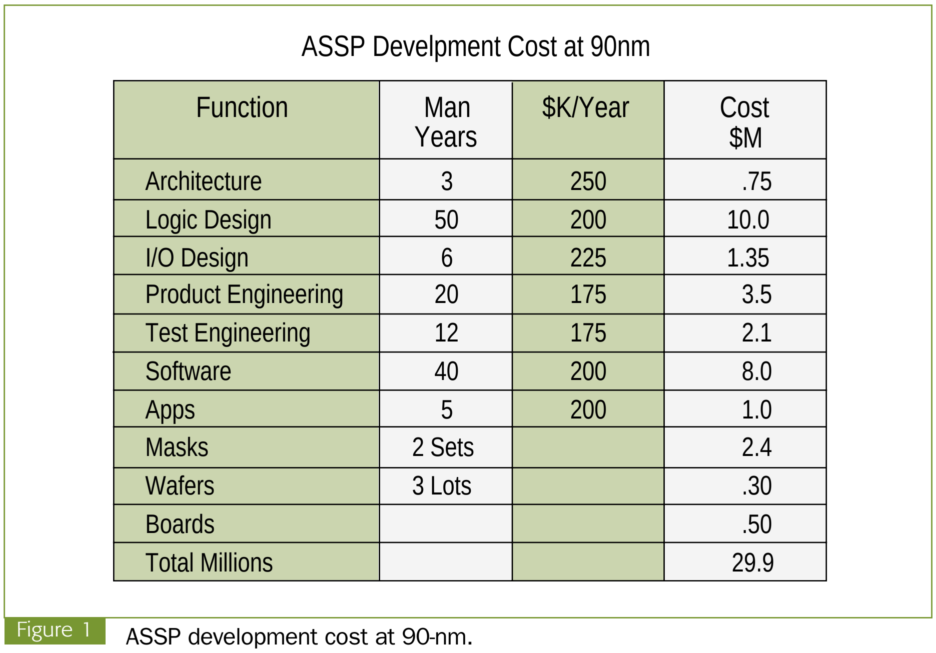
Xilinx
San Jose, CA.
(408) 559-7778.
[www.xilinx.com].

The Changing Economics of FPGAs, ASICs and ASSPs

There are still high-volume, high-performance markets where ASIC unit cost and performance clearly outweigh high development costs and risks. However, the number of those markets is steadily shrinking thanks to advances in FPGA technology.

by Jordan Plofsky, Altera

The economic and technological complexities of semiconductor manufacturing are increasingly becoming a threat to the dominant position held by application-specific integrated circuits (ASICs) and application-specific standard products (ASSPs). The industry's shift to the 90-nm process node is only serving to accelerate this trend as development costs skyrocket in step with the sophisticated and complex manufacturing requirements of the next generation of devices. Contrary to the increasing costs and risks of designing ASICs, field-programmable gate arrays (FPGAs), with their growing densities and on-board system functionality, are rapidly proving to be a cost-effective, flexible and lower risk alternative. FPGAs have evolved into an enabling technology that allows system designers to minimize the time and risk involved in developing a new product. Most importantly, FPGAs, with their in-field programmability, extend the time a product is *in the market*, thereby decreasing its threat of obsolescence by new generations of the same product.



ASSP Develpment Cost at 90nm

Function	Man Years	$K/Year	Cost $M
Architecture	3	250	.75
Logic Design	50	200	10.0
I/O Design	6	225	1.35
Product Engineering	20	175	3.5
Test Engineering	12	175	2.1
Software	40	200	8.0
Apps	5	200	1.0
Masks	2 Sets		2.4
Wafers	3 Lots		.30
Boards			.50
Total Millions			29.9

Figure 1 ASSP development cost at 90-nm.

The High Risk and Development Costs of ASICs

Development costs to first silicon have been rising with each new process node—today they can be as much as $20 million to produce first silicon, with estimates of increasing development costs at the 90-nm node approaching $30 million or more (Figure 1). If first silicon doesn't perform to specification—a likely possibility given today's highly complex designs incorporating hundreds of millions of transistors—

development costs can easily rise significantly before the product is ready to go into volume production. The impact on time-to-market and *time-in-market*—that period when a product's pricing, profitability and market dominance can be maximized—could be severe in terms of lost market share and revenue.

While the per-unit cost of an ASIC or ASSP may appear to improve with each new process node, one must take into account all the peripheral costs that go into a single chip's development. As integrated circuits shrink in size and increase in complexity, non-recurring engineering (NRE) costs have risen in kind. There is little reason to assume that NRE costs will not continue to rise as the industry pursues its technology roadmap down to the 90 and 65-nm nodes, which require cutting-edge processes such as deep ultraviolet (DUV) lithography and strained silicon.

In addition, the number of metal layers on advanced ICs has increased as the industry has migrated to smaller design rules, while at the same time significantly increasing device functionality. Each added metal layer requires an additional photomask and contact. Not only does development time and cost rise with each additional mask, but the risk increases as well. The complexities inherent in deep sub-wavelength lithography have significantly increased the likelihood that mask reworks will be required to correct yield-killing defects such as critical dimension or overlay errors. The time-to-market impact of these reworks can be extremely costly. These risks will only increase as the industry transitions from 193 to 157-nm lithography processes.

Time-To-Market, Time-In-Market

In today's fast-paced electronics markets, the time-in-market window is rapidly shrinking, putting considerable pressure on companies to reduce the time it takes to bring a product to market. This is in conflict with the fact that increasingly complex ASIC designs require longer validation processes that further extend the time it takes to bring an ASIC product to first silicon. The greater number of process steps required to build these advanced devices also increases both the cost and time of bringing a new product to market.

Already, typical ASIC development time to first silicon is generally around 18 months (Figure 2). Even a relatively small design fix to bring the device up to desired performance specifications can lead to a catastrophic delay in time-to-market of as much as six to nine months. Today, 24 to 27 months is often the equivalent to an entire product or process generation. Not only does each such delay erode a manufacturer's credibility in the market, it provides a dangerous opportunity for the competition to seize dominant market share for a product generation, or more, all the while reaping the benefits associated with greater time-in-market.

Being second to market often means steeply discounting product pricing in order to capture any market share. Market share gained at the expense of revenue is a

Pyrrhic victory at best. By contrast, opting for an FPGA solution may save as much as six months in bringing a product to volume manufacturing (Figure 3). That translates directly into an additional six months of time-in-market commanding premium prices, while competitors using an ASIC approach are still struggling to bring their new devices into production.

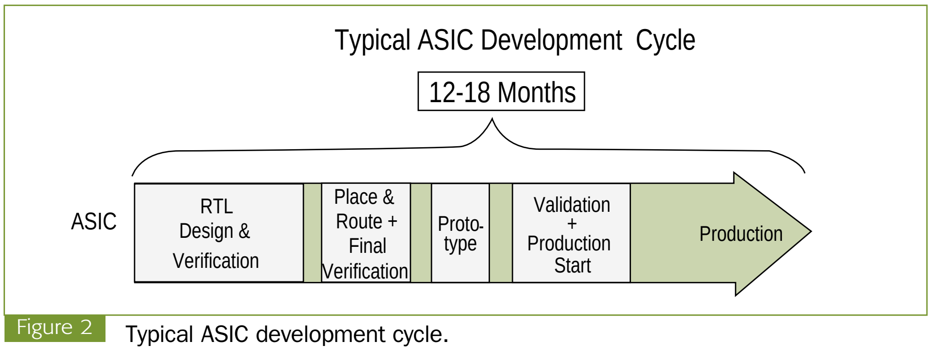

Typical ASIC Development Cycle
12-18 Months
ASIC
RTL Design & Verification
Place & Route + Final Verification
Proto-type
Validation + Production Start
Production

Figure 2 Typical ASIC development cycle.

In the highly competitive electronics markets, even an ASIC flawlessly brought to first silicon may not provide the optimal solution for system designers. Because of their high development costs, today ASICs have come to be a cost-effective approach for only those relatively few high-volume applications that, over a product lifecycle, will require millions of devices—such as microprocessors, graphics chips or cell phone chipsets. The time-to-market and time-in-market issues associated with ASICs only exacerbate this fact. For those applications with lower unit volumes, the alternative FPGA approach offers increased levels of flexibility, shortened development cycles and improved time-in-market windows at a significantly reduced level of risk.

Flexibility and Time-to-Market

FPGAs have come a long way from the days when they primarily served as glue logic or prototyping tools. In today's FPGAs, logic shares a proportion of the die area, with a variety of new functions, such as transceivers, specialized memory, embedded processors, embedded DSP accelerators and clock data recovery circuitry. These functions can be built into an FPGA just as cost-effectively, and achieve the same high performance, as in many ASICs.

More and more system designers are flocking to FPGAs for volume applications because of the advantages they offer in terms of increased flexibility, reduced risk, better time-to- and time-in-market and lower overall costs. High-performance FPGA families are being used in demanding applications such as high-end switches and routers for networking applications, cellular phone base station processing applications and high-end professional video services such as video conferencing. Low-cost FPGA families are within a hair's breadth of ASIC prices and are increasingly being used for cost-sensitive, volume applications—especially for consumer electronics products such as digital video players and set-top boxes—where overall system cost and time-to-market are major considerations.

Flexibility remains one of the major advantages provided by FPGAs because of their re-programmability. Systems using programmable devices can be easily upgraded or have bugs repaired in the field. In addition, system manufacturers can use the same FPGA device to differentiate system performance and cost with minimal redesign since most changes in performance functionality can be programmed, and hardware redesign, a supply chain headache, becomes a thing of the past. Manufacturers of TV set-top boxes, for example, are leveraging low-cost FPGAs to differentiate standards and protocols by geography, using the same box for millions of customers in one country or several countries.

An FPGA solution also provides significant cost savings in terms of design and support tools. A suite of the tools required to design a new FPGA costs approximately 85 percent less than those required for new ASIC development. That can amount to a savings of nearly $400,000 in development costs. Meanwhile, FPGA development tools are increasing in value as their costs have remained constant over the last few years, even though the devices they target are becoming more complex. In addition, the increasing availability of a broad range of pre-verified intellectual property to design various functions into FPGAs further reduces design cycle times.

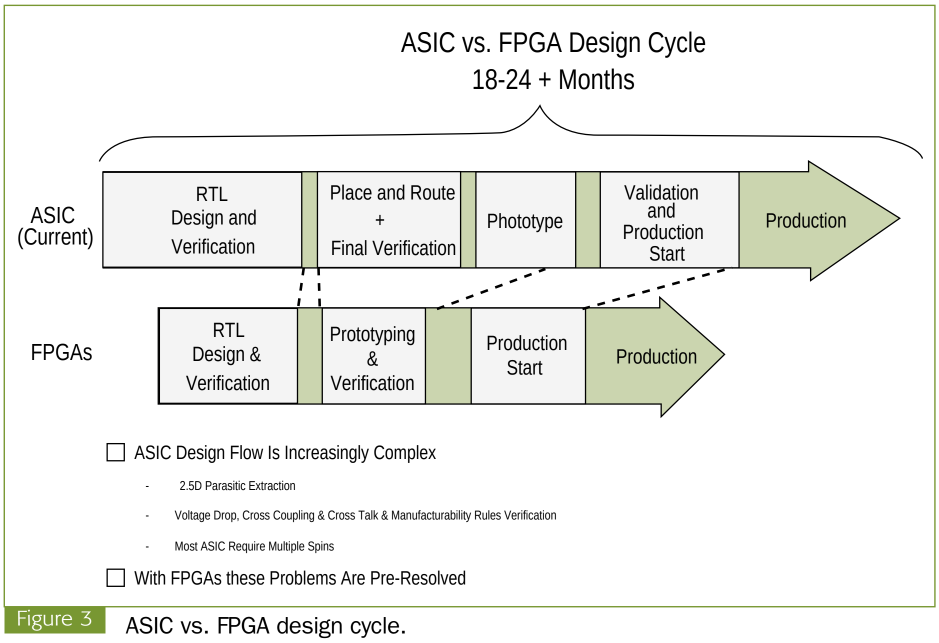

ASIC vs. FPGA Design Cycle
18-24 + Months
ASIC (Current)
RTL Design and Verification
Place and Route + Final Verification
Phototype
Validation and Production Start
Production
FPGAs
RTL Design & Verification
Prototyping & Verification
Production Start
Production
□ ASIC Design Flow Is Increasingly Complex
- 2.5D Parasitic Extraction
- Voltage Drop, Cross Coupling & Cross Talk & Manufacturability Rules Verification
- Most ASIC Require Multiple Spins
□ With FPGAs these Problems Are Pre-Resolved

Figure 3 ASIC vs. FPGA design cycle.

As FPGA vendors continue to turn their focus on capturing more of the market occupied by ASIC manufacturers, technology exists today that enables an FPGA design to migrate seamlessly into a hard-mask solution, similar to an ASIC, with the same FPGA architecture. ASIC and ASSP designers frequently use FPGAs in prototyping. Once the prototype has proved out, the design is transferred to an ASIC architecture, a process that may take up to 18 months or more. Even with a successful FPGA prototype, there is never any guarantee that the ASIC intended for volume production will not have to go through some last minute redesigns to meet end-market specifications.

With this migration approach, development is also done using an FPGA. Once the design has been proved and the device is ready to go into production, however, the unused programmability functions are stripped out of the device. The rest of the design remains the same, without the requirement for redesign inherent in a conversion to an ASIC for volume production. This approach helps lower volume unit price, enables a die shrink (though the packaging remains the same) and avoids the risks inherent in converting to an ASIC for volume production, all while dramatically speeding time-to market.

The original FPGA cost is then dramatically reduced by as much as 70 percent, bringing the volume cost to parity with an equivalent ASIC. Meanwhile, the finished device can be brought to full volume production in approximately six to eight weeks, a time-to-market improvement of nearly 80 percent as compared to that of using an ASIC. Furthermore, rather than spending 18 months or more managing the conversion to an ASIC design for volume production, a company's design team can focus on developing its next-generation product—potentially putting it one product generation ahead of competitors who did use ASICs.

Converging Worlds

Perhaps the best evidence that FPGAs are successfully challenging ASICs and ASSPs as a viable volume production alternative is the way the two worlds are slowly merging. ASIC vendors are developing embedded FPGA-style programmability on their devices as a means to reduce design time. Others are developing semi-custom devices that incorporate IP blocks and limited levels of programmability on the metal layers. At the same time, FPGAs are embedding ASIC cores for specific applications, such as digital signal processing. While at first glance all these approaches may appear to be equally valid, the approaches that attempt to embed programmability onto an ASIC have only limited value. Efforts to embed programmability into an ASIC have so far seen no success, as no products are commercially available today.

The semi-custom, or gate array, approach also has some serious drawbacks when compared to FPGAs. Essentially, it offers neither the time-to-market advantages nor the flexibility inherent in true programmable devices. It also lacks the dynamic reconfigurability available with FPGAs. Further, while such devices may have shorter development cycles than traditional ASICs or ASSPs, they still do not offer the time-to-market and flexibility advantages of FPGAs. ◢

Altera
San Jose, CA.
(408) 544-7000.
[www.altera.com].

Using Embedded FPGA Cores in DSP Applications

With the advent of adaptive signal conditioning and rapidly changing infrastructure "standard" specifications, programmable logic is poised to complement or even replace general-purpose DSP engines for much of the heavy lifting in the computing, consumer and communication markets.

by Dan Pugh, Leopard Logic

One of the key issues in designing DSP algorithms is to select the right implementation platform from a number of different choices such as general-purpose digital signal processors (DSPs) or programmable logic devices like FPGAs. Primary selection criteria usually include cost, performance and power consumption, followed by no less important issues like ease-of-use, availability of third-party IP and integration with development and analysis tools in the DSP domain. Recently, an innovative approach of creating a new breed of configurable application platforms to facilitate the efficient implementation of high-performance DSP algorithms is emerging. Until now, there have been three alternatives: general-purpose DSPs, discrete FPGAs or custom ASICs.

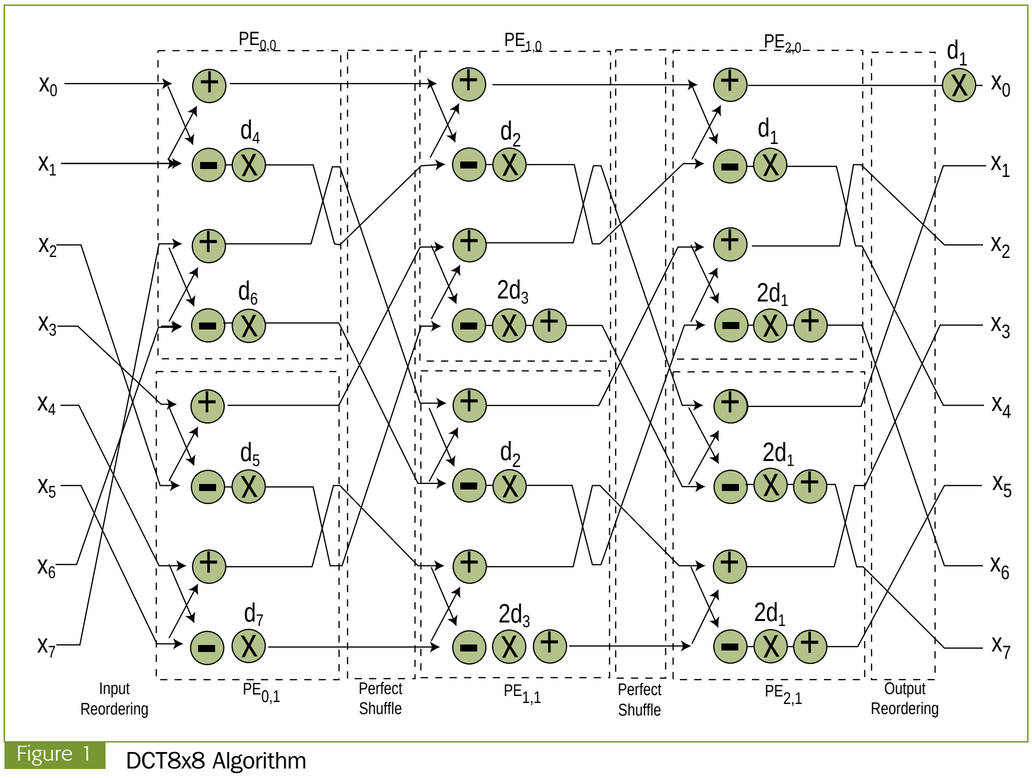


Figure 1 DCT8x8 Algorithm

Off-the-shelf general-purpose DSPs are typically inexpensive and are supported by a wealth of third-party IP in the form of optimized assembly language routines. Due to their fixed generalized architecture they are usually not ideally suited for any specific application. System designers regularly use multi-DSP core solutions or multi-execution unit engines to meet computing and throughput needs.

These multi-core engines are rapidly converging with current CPU-based approaches but suffer from the additional programming complexities introduced by non-uniform dedicated computing ele-

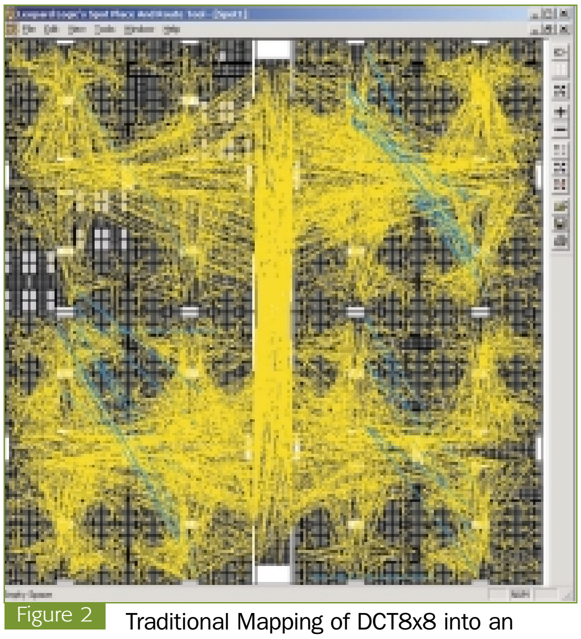


Figure 2 Traditional Mapping of DCT8x8 into an Embedded FPGA.

ments such as multiply/accumulate units (MACs) or special indexing units. These architectures are strongly dependent on compiler efficiency and the compiler's ability to utilize the parallel computation elements effectively.

The implementation of DSP algorithms in software carries a large overhead compared to optimized hardware implementations and thus results in a performance hit of at least an order of magnitude. Multi-execution oriented architectures are more RISC-like in nature and have an associated register transfer overhead (code bloat phenomenon) of at least 40%, which leads to high power consumption, especially at higher clock frequencies.

Current FPGA products can offer superior performance for dedicated task functions but are not optimally placed in terms of space and partially for that reason suffer from high overhead in terms of hardware interconnections. Other current limitations include high cost, high power consumption and specialized mapping tools and design methodologies.

FPGAs offer a generalized pool of resources consisting of memories, multipliers and logic that is replicated in a two-dimensional tiled structure. Thus the resources are organized according to physical considerations rather than the requirements of a specific application domain. The ability to use these devices is a hardware driven exercise requiring specific hardware architecture knowledge and tool application.

The efficiency of a mapped algorithm, although a hardware exercise, still requires specific knowledge of the chosen FPGA target in order to achieve high performance with decent utilization of the arrayed resources. It is not possible to take advantage of existing higher layer functionality in an integrated form by using software application code decks. That would be a requirement for layer 3-7 communication protocol handling or for system-level application control layer functionality.

ASICs are best suited for high-volume applications. The architecture is defined specifically for the application at hand, yielding a low-cost, low-power solution. An ASIC has the precise mix of processing elements, memory and interconnect to fit the target application, but it yields a fixed solution. ECOs and algorithm updates common in today's products can render an existing ASIC obsolete. Although ASICs can lead to an ideal solution for a single application, the multimillion dollars costs associated with tools and mask sets limits their use to high-volume, fixed function designs, which average six months to a year to design. Modifications to the ASIC require costly mask set changes that are approaching one million dollars for 0.13-micron CMOS designs.

In order to provide flexibility, ASICs are commonly paired with discrete FPGAs, but this flexibility comes at a price. The addition of discrete FPGAs is a costly solution, especially when only a small amount of flexibility may be needed for the design. The discrete FPGAs also require the ASIC to provide additional pins for interconnect. Not only does this drive up the ASIC package cost, but the higher voltage at the pins required to maintain noise mar-

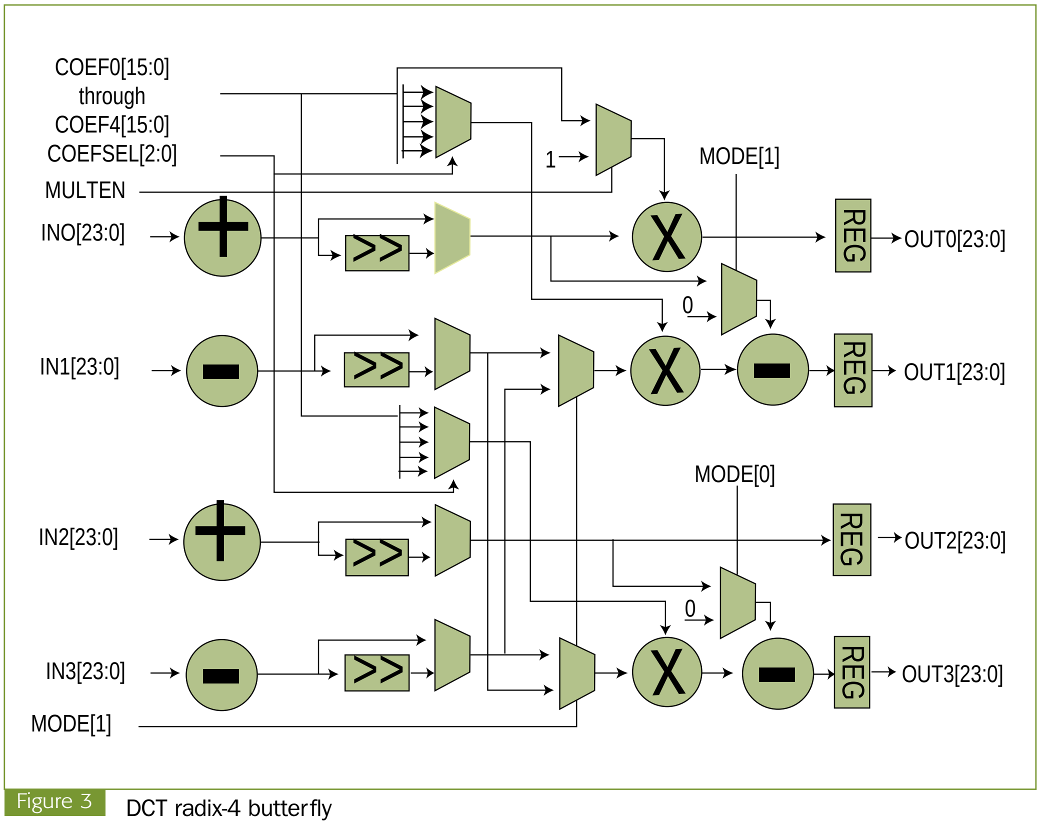


Figure 3 DCT radix-4 butterfly

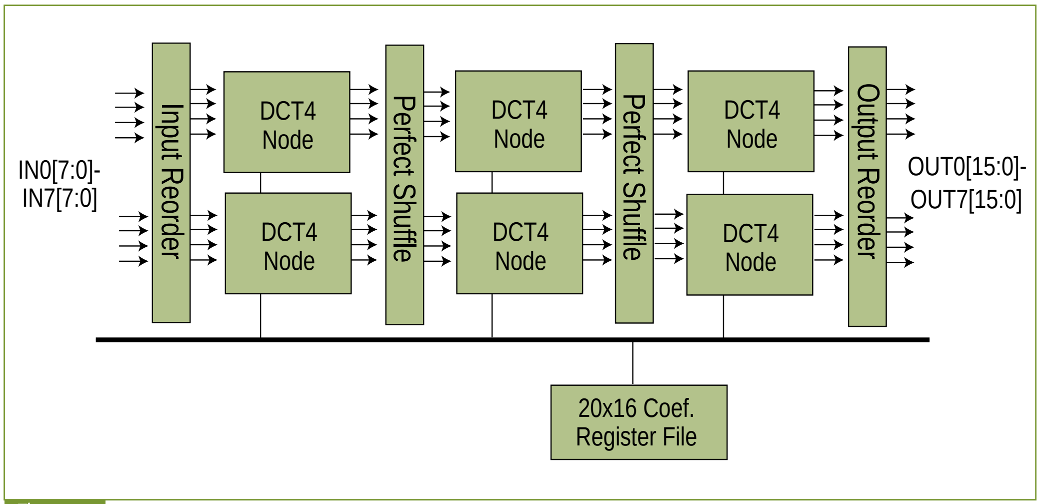


Figure 4 DCT8x8 Architecture implemented as standard-cell ASIC technology.

gins on a circuit board also drives up the power requirements.

A New Hybrid Solution

A new approach to the traditional DSP solutions addresses the limitations of the technologies outlined above. This technology combines the best aspects of ASICs and FPGAs into a "best-of-breed" solution, using embedded FPGA cores integrated in an ASIC fashion along with mostly commercially available IP elements such as processor cores and memory. As a result, ASIC designs are no longer limited to fixed functionality, but can now have the advantages of embedded FPGA—e.g., flexibility and custom algorithms—without the limitations of discrete FPGAs.

The best-of-breed solution uses hard-wired logic for well-defined, fixed functions such as processors, memories and select data path elements. The FPGA core is used in areas that require flexibility such as processor accelerators, data path control, state machines, IO blocks or high risk areas in a design. By tailoring the proper mix of fixed and programmable elements to address an area of applications, this approach delivers flexible platform ASICs.

DCT Design Example

The Discrete Cosine Transform (DCT) algorithm is a common application kernel used in the image processing domain. It is generally widely understood and as such will be used to illustrate concepts of interest from the perspective of using an embedded FPGA core.

This example implements an 8x8 DCT that is most typically used in video encoders with 8-bit data inputs and 16-bit data outputs. This algorithm has been implemented in a DSP, a discrete FPGA, an embedded FPGA core and as a hybrid platform that uses the best-of-breed method, which divides the design into hard-wired data path components and flexible control components. The design contains thirteen multipliers in hard-wired form with the remainder of the logic in more traditional FPGA-like logic. The DCT8x8 algorithm is defined in Figure 1.

Obviously this algorithm can be performed even on a generic class 16-bit DSP, but it will incur a large latency and require many DSP clock cycles for execution. The relative inefficiency of logic activity per clock cycle versus a direct parallel hardware implementation also incurs a significant power penalty. In general, it is advisable in DSP computation to calculate at the highest possible throughput for the least amount of time.

Another design alternative is to map the DCT8x8 of Figure 1 into hardware using a conventional FPGA technology. The design challenge in this case is to maximize the use of the tiled resources that have been pre-selected for the general case. This becomes an issue when tables need to match available memory segment sizing, or when arithmetic processing needs to be matched to the availability of multiplier units and their associated bit widths. Independent of the mapping process, the final form design realization in a commercial FPGA will likely consume a substantial amount of power and have a significant cost premium as compared to a full custom, standard cell or gate array version.

For this example, the DCT8x8 has been mapped into a platform FPGA costing over one hundred dollars in small quantities from a leading provider of programmable logic solutions. The design uses the hard-wired multipliers that are distributed through the FPGA logic array. The resulting design runs at a maximum operating frequency of 103 MHz under worst-case commercial conditions.

When the complete DCT design is mapped into the embedded FPGA the design requires 975 Core Cells, utilizing over 95% of the 1024 Core Cells, shown in Figure 2. Whereas speed degradation is common in highly utilized discrete FPGAs that are on the market today, the hierarchical interconnect of the embedded FPGA

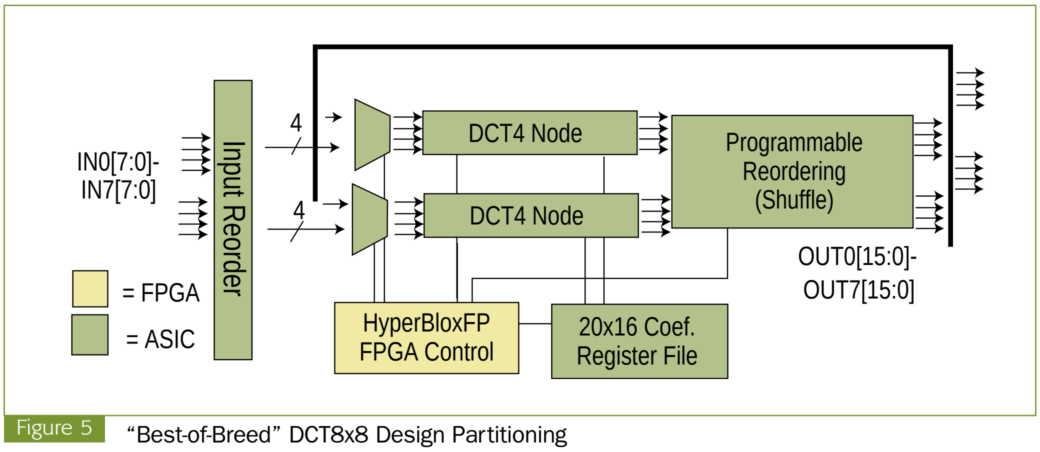


Figure 5 "Best-of-Breed" DCT8x8 Design Partitioning array allows this design to operate at a maximum operating frequency of 288 MHz under worst-case commercial conditions on a 0.13 micron CMOS process. Although this mapping shows impressive performance and is good for comparison purposes, it is not the best method to implement designs using embedded FPGA.

"Best-of-Breed" Mapping to an Embedded FPGA

The final implementation of the DCT8x8 illustrates the design approach that represents a best-of-breed solution by leveraging the best technology from the ASIC and FPGA worlds. The designer is no longer forced to accept a specific configured platform solution. As seen in Figure 1, the DCT algorithm may be decomposed into three columns, each of which contains two radix-4 DCT butterfly operations. The six butterfly components each take on a slightly different form, but a single butterfly can be designed that encompasses all of the required modes with a single design, as shown in Figure 3.

Note that in Figure 3 additional shifters were added to the basic structure in order to control bit growth. Six of these butterfly nodes are then combined into the structure required for the DCT8x8 as shown in Figure 4. This structure could be implemented in standard cells.

To maximize efficiency, the radix-4 butterfly of Figure 3 can also be implemented in low-power and area-efficient standard-cell ASIC circuitry because the butterfly contains well-defined arithmetic blocks that have fixed functions. Note the control lines in Figure 3. Although the ASIC components are fixed in function, the control lines allow the desired mode of operation to be selected externally—in this case by adding FPGA circuitry.

The embedded FPGA is used to provide flexible, reprogrammable control to the ASIC data path components. For even better flexibility, two of the radix-4 butterfly nodes are combined to form one of the three columns of the DCT8x8. When allowing the data three cycles for processing, a single column can be used to circulate the data through the pair of radix-4 butterfly nodes three times. On each pass the controller implemented in the embedded FPGA selects the proper mode of operation for each of the butterfly nodes.

The resulting architecture in Figure 5 shows the partitioning between the ASIC and FPGA circuitry. This best-of-breed partitioning results in a circuit with a maximum operating frequency of 769 MHz under worst-case commercial conditions on a 0.13-micron CMOS process. Since this architecture requires three passes through the circuit, the net processing rate of this circuit is 300 MHz. If the design requires the full 769 MHz, all three DCT columns would be instantiated. This partitioning of the DCT8x8, with a simple reprogramming of the embedded FPGA, can also be used to implement a 32-point DCT such as is used in audio compression.

Hybrid solutions using embedded FPGA technology in combination with

Implementation	Performance	Area	Recurring Cost	Flexibility	Power
DSP Processor	Up to 1GHz, many cycles req.	High	Med	Med	Med
ASIC	1GHz +	Low	Low	None	Low
Discrete FPGA	103 MHz	High	High	High	High
Mapping into embedded FPGA Architecture	288 MHz	High	Low	High	Med
"Best-of-Breed" Hybrid Architecture	**769 MHz**	**Low**	**Low**	**High**	**Low**

Table 1 DCT8x8 Implementation Comparison

hard-wired logic, memories and multipliers allow designers to achieve maximum DSP performance with low power and at attractive price points (Table 1). Leveraging this capability, designers can architect system-on-chip (SoC) devices with the optimum tradeoff between performance, area and costs for implementing the desired target application. Using the described best-of-breed approach, memory bandwidth is flexible and bus allocation is again under user control for optimal results.

Embedded FPGA technology enables this new class of semiconductor device that offers new choices to designers as compared to conventional approaches used today. Hardware is the only implementation that is capable of meeting market needs for high-performance compute algorithms and high-bandwidth signal processing. However, flexibility and adaptability are requirements for most DSP applications. Economics coupled with time-to-market needs will replace growing banks of conventional DSP processors with computationally efficient, cost optimized and power efficient solutions.

Leopard Logic
Cupertino, CA.
(408) 777-0905.
[www.leopardlogic.com].

New FPGA Price Points and Density Range Revolutionize System Design

ASICs and processors are not the only kinds of silicon that have been shaped by Moore's Law.

by Rob Schreck, Xilinx

The industry has ridden the Moore's Law technology wave to higher integration. The 18-month doubling of density has impacted memories, microprocessors, ASICs and even programmable logic. Now, with more capability on a smaller and smaller die, the industry has a new programmable development platform enabling a new class of systems development.

Moore's Law enables designers to get incredible capability at a very low cost, because standard products can take advantage of high production volumes to drive down unit costs. These new standard programmable logic devices now deliver a true alternative to ASIC design. With up to 5M system gates and almost 2M bits of block RAM, they compete on both performance and integration levels with both gate array and standard cell technology. With abundant logic and memory resources, designers also have access to flexible I/O, embedded high-speed multipliers and even soft-core processors, to get a flexible, low-cost programmable platform for a wide range of high volume applications.

Looking Back at FPGAs and Tools

Field Programmable Gate Arrays (FPGAs) and Programmable Logic Devices (PLDs) have been part of the mainstream electronic design community since the early 1980s and initially offered hundreds of programmable gates. Many of the early PLDs were fields of "And" and "Or" arrays, so that sum-of-products logic design could be created with programmable interconnects. These devices replaced various discrete digital logic components with a single family of off-the-shelf units that could be used for any type of design. Early devices were generic, which simplified inventory but required specific manual design tools of the schematic-capture and Boolean-equation compiler variety.

Over the last two decades, FPGAs and PLDs have swelled in size and functionality at an astounding rate (Moore's Law!). As such, design techniques have had to evolve to keep up with these advances. When programmable devices grew to tens-of-thousands and hundreds-of-thousands of logic gates, manual design at the gate level became less efficient. Similar to the growth in popularity of software programming languages—such as C/C++, Java, Ada and Pascal—over assembly language, a higher level of abstraction was quickly required for larger, more complex hardware designs.

Hardware Description Languages

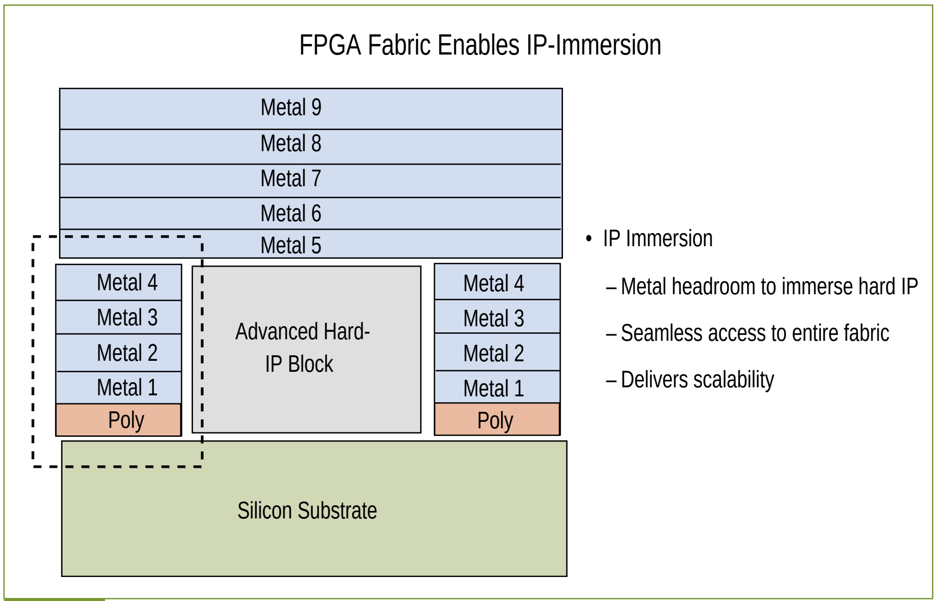


Figure 1 Hard IP-immersion embeds low- and medium-complexity hard IP blocks into FPGAs, such as Block RAM and multipliers. The cross section of the silicon die shows metal headroom over the immersed hard IP block. By contrast, Soft IP-immersion enables users to integrate soft IP into any location of the FPGA and move it around without any performance or access penalties.

(HDLs) like Verilog and VHDL were quickly adopted for hardware design because they were more efficient at creating higher level designs, yet did not sacrifice low-level support for hardware development. Now that the latest devices supply millions of gates of available user logic and introduce integrated system components within the silicon, the commensurate design tools must support all aspects of "system" design, including the addition of IP generation and embedded software tools, such as compilers and debuggers.

Meet the New Platform FPGAs

One of the big differences between a traditional logic device and an ASIC is that the term "system on a chip" historically has been reserved just for the ASIC. That position has changed radically now that IP-Immersion processes are so successful for embedding system component IP into fully programmable FPGA devices. For example, the Xilinx Spartan-3 FPGA integrates block memory, soft microprocessor cores, customizable IP and high-performance DSP functionality into the logic to provide a flexible platform.

IP-immersion technology allows for integration of system components onto a single device, reducing the total bill of materials parts count, shrinking board space and improving reliability. With the IP-immersion technology, the hard IP is embedded into the layers of the device, and, combined with abundant routing over the IP blocks, allows high performance and high integration (Figure 1).

Some FPGA vendors are making both soft and hard microprocessors available, so users can craft the solution best suited for their embedded application. Partnering with industry leaders in the microprocessor arena provides popular and mature processor hard cores in a variety of high-performance configurations. You can choose devices that include hard IP processors and can utilize the on-chip memory to guarantee a fixed latency of execution for a higher level of determinism.

For example, today, FPGAs offer a platform for programmable system design along with 32-bit processor cores, such as the PowerPC, MIPS and ARM, and multi-gigabit transceivers immersed into the device. Straight FPGAs along with those integrating processor cores address many aspects of reducing system costs. They have similar logic structures, and use the same design tools and IP, but are tuned with different capability/performance trade-offs. This flexibility and scalability are the direct result of driving Moore's Law to a new level.

FPGA suppliers have also introduced 32-bit soft microprocessor cores that can stand alone or complement hard-core processor applications. You can add soft-core computing applications to your platform FPGA designs or, using the same bus standard, introduce soft micro-engines to off-load time consuming functions from the main hard-core processor. The soft-core processors do not have the performance of the hard-core types, but they are small and limited only by the size of the FPGA. By being able to choose between both hard- and soft-core processors, an engineering team can create the ideal platform for its specific application.

Ideally, peripherals and other system intellectual property are soft in these programmable platforms, so you can choose exactly what you want and not worry about "running out" of that IP. Gone are the days of placing additional microprocessor packages on a board—not to use the processor cores themselves, but to supply the required amount of discrete hard IP.

FPGA providers should supply a standard library of IP, including such cores as arbiters, bridges and UARTs with the device, along with additional high-end cores separately. An additional desired feature is the support necessary for customizing the IP to such a detailed level that designers can trade-off features, performance and size for every individual piece of IP to enable fine-tuning the entire design

In addition, digital signal processing functionality is exploding in these new platform FPGA systems, in the form of hardwired multipliers that yield hundreds of billions of multiply/accumulates per second. This capability greatly exceeds even the fastest of sequential DSP processors available on the market today. The sweet spot for FPGAs in DSP applications is in the region of 1 to 300 megasamples per second (MSPS), where customers are most concerned with high performance and high flexibility.

One of the other key advantages of this new breed of platform FPGAs is flexible connectivity. These FPGAs have to interface to a wide range of products in order to provide a complete design platform. HSTL and SSTL allow efficient memory interfaces, while LVDS provides a high-speed link that avoids cross-talk and other interference. Platform FPGAs offer support for I/O standards and designers can use soft IP building blocks for a wide range of interface protocols such as PCI 32/33 and PCI 64/33, RapidI/O, POS PHY Level 4, Flexbus 4, SPI-4 and HyperTransport.

The challenge in developing a low-cost Platform FPGA is to develop a small die to save costs, yet have enough I/O pads around the periphery of the smaller die to offer adequate I/O. Logic IC designers have found a solution with staggered pad technology that implements two rings of I/O pads around the periphery of the die to maintain I/O counts with ever-decreasing die size (Figure 2).

Platform FPGA Development Tools

Platform FPGAs need a full spectrum of development tools, both for design synthesis and for FPGA design compilation, floor-planning and place-and-route. Vendors need to provide a wide range of partnerships with EDA vendors to support a design engineer's development environment. Furthermore, FPGA design tools need to provide high productivity to minimize design time and design engineering costs.

Incremental design capability slashes design re-compile times by limiting the re-implementation to only the design modules that need to change, the rest of the design is frozen and intact, preserving previous performance results. Modular Design delivers a "divide-and-conquer" team-based approach to high-density design, allowing teams of engineers to complete their modules in parallel, focusing on individual module performance rather than overall design completion, and speeding the design flow through to faster completion.

The area mapping capabilities of a floor planner allow for quick-and-easy logic grouping, leading to better timing results, and faster design performance. Relationally placed macros (RPM) allow design teams to save floor-planned HDL designs for later design reuse, further increasing productivity.

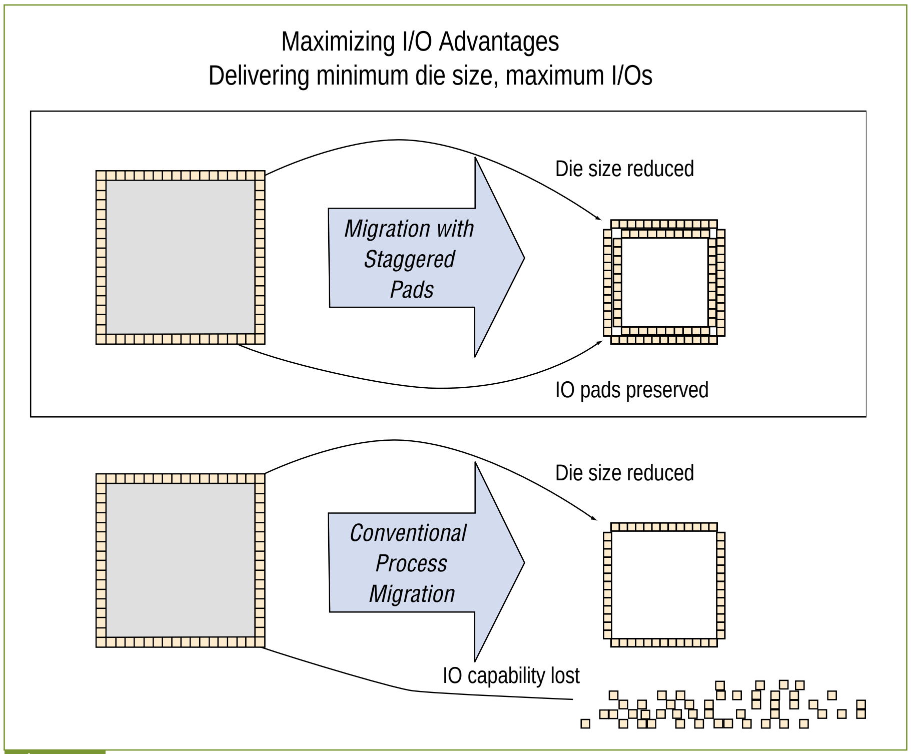

Maximizing I/O Advantages
Delivering minimum die size, maximum I/Os
Migration with Staggered Pads
Die size reduced
IO pads preserved
Conventional Process Migration
Die size reduced
IO capability lost

Figure 2 Staggered Pad Technology allows reduction in die size while preserving the number of I/O pads.

With platform FPGAs, engineers will be able to target design modules for silicon hardware in FPGA logic gates or as software applications run on process engines, implemented as soft processors. Because many kinds of engineers—hardware, software, firmware, system architects and others—may target these platform FPGAs, look for a tools strategy aimed at appealing to these different camps. Top suppliers not only should produce tools for IP generation, DSP design and logic implementation, but partner with leading EDA and embedded companies to provide best-of-class support for logic synthesis, simulation, co-verification and embedded software development.

Moore's Law has driven the industry to adopt more and more advanced technologies to deliver higher integration and lower costs. Today companies can take advantage of low-cost, off-the-shelf solutions that provide high integration to get a wide range of products to market quickly. These new low-cost Platform FPGAs embed hard IP cores, memory and high-speed I/O with large logic density to offer a flexible solution that increases design productivity. Imagine how you can use these for the next 4 decades of system development. ◢

Xilinx
San Jose, CA
(408) 559-7778.
[www.xilinx.com].