White Paper

Hardware/Software Co-Verification Using FPGA Platforms

Introduction

The problem of hardware and software co-design is as old as systems design and the integration of systems composed of multiple elements. Systems built using electrical and electronic subsystems, mechanical subsystems, software, and embedded software and firmware have always been difficult to design. The problem was less complex when technology for mechanical, electronic, electrical, and software did not allow paper designs to be realized easily in a tangible system, but things have changed with the increasing complexity of systems.

Not doing staged integration and co-development of hardware plus software is extremely costly because of the added complexity. Parallel development efforts must be run for the subsystems and the cost of errors along the way minimized. This has become important due to:

- *Time-to-market pressures*, such as launching the product before competitors do, before the next shareholders call, or because customers are asking for it.
- *High cost of custom semiconductors:* ASIC development is extremely expensive. For lower power, the latest geometry node and process are necessary, while high performance requirements dictate the building of custom solutions for area and power savings and for maximum performance.
- *Cost of re-design and sustaining:* Product design cycles are typically shorter than re-design and test phases, so sustaining becomes a costly exercise for poorly designed products.

These factors have spawned many technologies and tools for process management at the project and design level, as well as integrated design flows at the element or component level such as ASIC design. The flows and technology have matured much faster for software development and ASIC development, as there are numerous instances, corporations, and groups involved. Complex system design requires staying power to manage an entire product design cycle involving hundreds if not thousands of constituent elements.

In contrast, FPGA-based prototyping platforms enable both a wider adoption of development methodologies and risk and cost reduction in integrated product development.

Quantifying the Problem

Figure 1 shows a development flow for a hypothetical product that has a mix of hardware and software with custom elements. The development flow goes through eight stages:

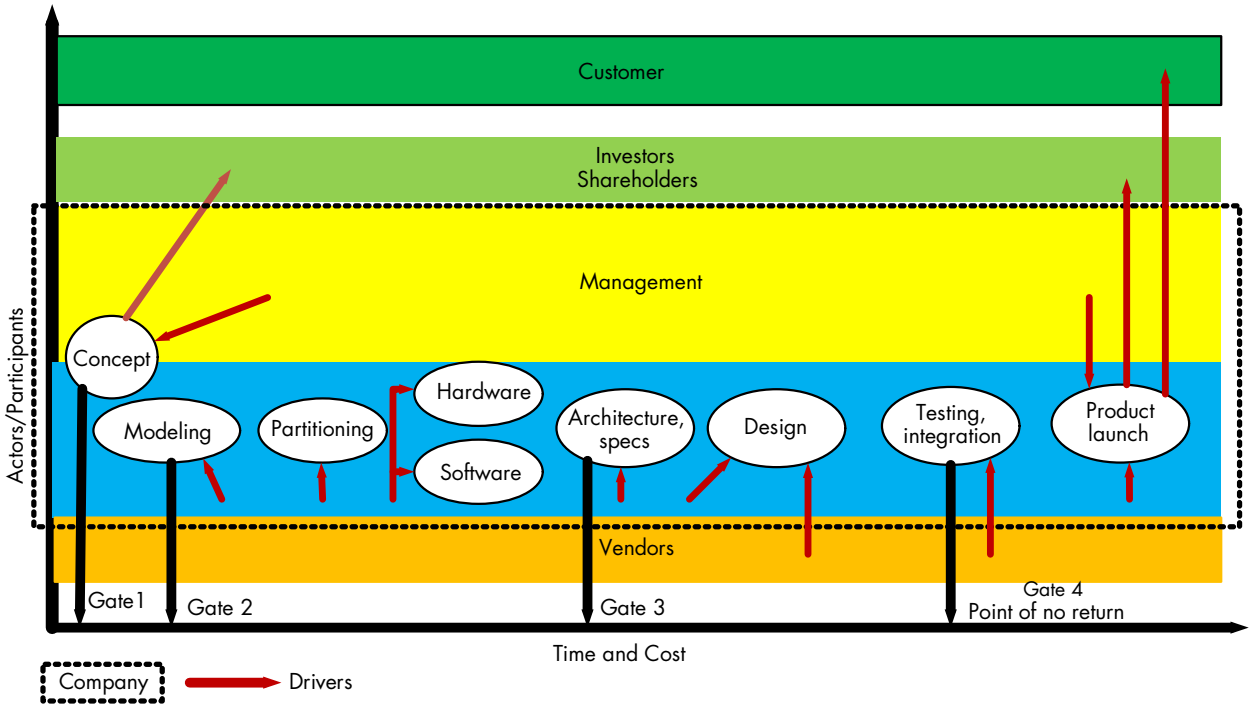
1. The concept for a product is developed jointly by management and engineering, and is communicated to the shareholders and investors as a preview of future business plans or a roadmap. This concept stage serves as Gate 1 for a go/no-go decision.

2. Engineering and marketing validate assumptions in the concept by modeling, which leads to Gate 2 for a go/no-go decision. If the feasibility of the concept is found to be impractical because of the premise or assumptions, a refinement of the concept (return to the Stage 1) or no-go decision are likely. The output of this stage is an equivalent to a high-level requirements document (HRD) coupled with a market requirements document (MRD). These serve as the feature and function descriptions for the engineering and execution team.

3. After a go-ahead decision at Gate 2, the systems engineers and architects begin to partition the concept and model into hardware and software components. They also set the top-level performance constraints. The typical output of this stage is the systems design document (SDD).

4. Part of the process at this stage is the design and architecture exploration.

Figure 1. Actors and Drivers in an Integrated Product Development Flow



5. Gate 3 comes after the team leads, individual contributors, and hardware and software systems engineers generate a detailed architecture of what is to be implemented. The output of this stage can be thought of as an equivalent to a high-level design document (HLD). In addition, trade-offs are made about what can be done within budget, with state-of-the-art technology, where all the elements come from, and within project plan and timeline restrictions.

6. After a go/no-go decision at Gate 3, the active design process begins. The internal teams and vendors are engaged for a variety of services (providing software services, parts, testing, etc). At this stage, the company still can decide to cancel product development without going through a detailed engineering and execution phase.

7. There must be a final or factory test and integration before product launch. Typically, once a project has reached this stage, enough dollars have been sunk into it that it is beyond the point of no return. In rare cases, projects are cancelled at this late stage; despite the money that has been spent, other factors influence the decision.

8. The product launch is where the engineering team, management, and all aspects of the value chain are validated by customer and market acceptance.

Note that even though cost is shown on a linear scale, it goes up exponentially as number of people involved and the budget outlay for operations and capital increase as time passes. The tangible and intangible costs of failures, errors, and delays scale exponentially with time.

Potential Solutions

The consequences and the cost of discovering errors and performing modifications associated with feature functionality required in product go up as time progresses. These include:

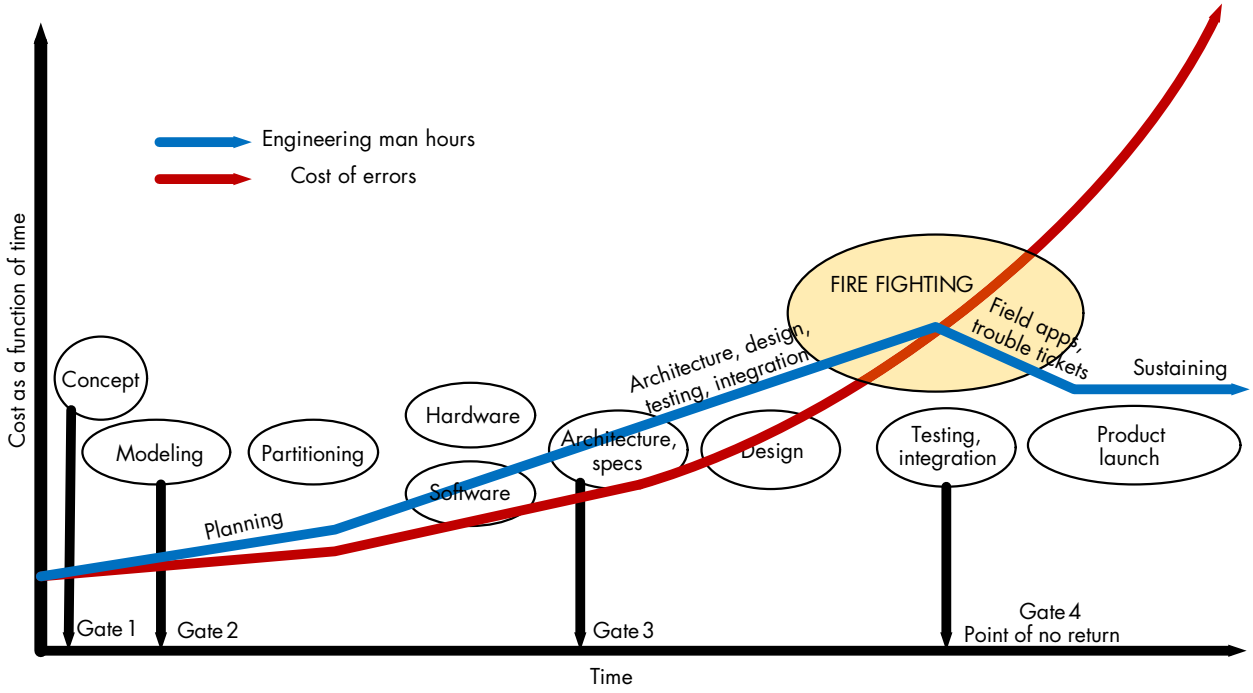
- *High launch and sustaining cost:* The high cost of discovering functional or feature-related errors in silicon or system operation leads to a much higher sustaining cost for the product (recalls, re-design, support, maintenance, loss of market share, etc.).

- *Increased planning complexity:* Planning integrated product development can be difficult where hardware, software (applications), and embedded software development are run in parallel tracks. Staged integration and co-verification with robust feedback does not take place. The hardware takes longer to produce and modify, as the whole process is serialized no matter that it is an ASIC, system, or subsystem. The software and firmware (embedded software) teams wait for the hardware to test the drivers, feature sets, hardware abstraction layer (HAL) and other features and functions. This reduces the probability of first-pass success, increases risk, wastes development dollars, and delays the schedule.

FPGAs, especially Altera® Stratix® III FPGAs, and design software such as Quartus® software, provide a clear risk reduction solution. IRIS Technologies incorporates multiple Stratix III EP3SL340 devices in its S3 card to provide a standards complaint (form factor and I/O), scalable prototyping platform for ASIC and systems prototyping, and simulation and emulation. The S3 cards and IRIS systems use standard design flows and tools from Altera and its partners. The "building block" method of putting together systems and platforms enables rugged, robust, scalable solutions. This paper examines several use models by application, the key features of the S3 card, and the advantages they provide for each application for ASIC and systems prototyping, simulation, and emulation.

Figure 2 shows cost as a function of time overlaid graphically with the product flow.

Figure 2. Cost of Errors and Modifications as a Function of Time



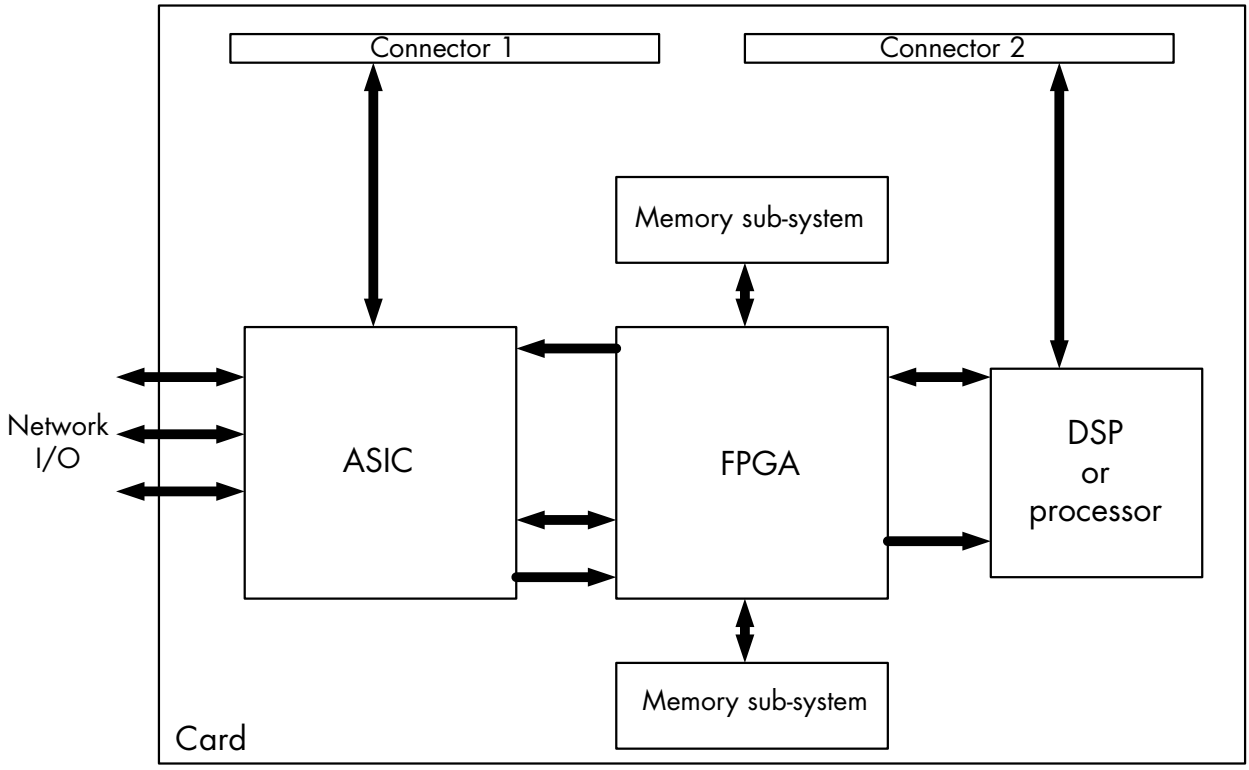
Every project manager, engineering manager, and executive tries their best to avoid the "fire-fighting" zone. The solution to avoid this is multi-faceted, but a key enabler of any risk reduction strategy is the availability of a reconfigurable prototyping platform that can be used for hardware and software co-verification. Good rules to follow in an integrated product development flow are:

- Form strong integrated product teams with systems engineering skills that know the application well.
- Test each software and hardware module in-system as part of the flow to minimize risk. Provide hardware-in-loop (HIL) testing, a good QA- and module-level test and verification plan.
- Perform incremental integration, verification, and testing of hardware and software modules using emulation and prototyping platforms.

An Example Target Design

Figure 3 shows an example design-possibly a line card, a set-top box, or a generic motherboard-that involves hardware and software components. It has software at the application layer, embedded software (firmware), perhaps a modified operating system kernel with a protocol stack, and hardware that consists of an ASIC, an FPGA coprocessor and an off-the-shelf digital signal processor or standard processor. The operating system and the protocol stack run on the ASIC, the FPGA serves as a memory interconnect and bridge, and the processor runs embedded software (firmware).

Figure 3. Example Design (1)

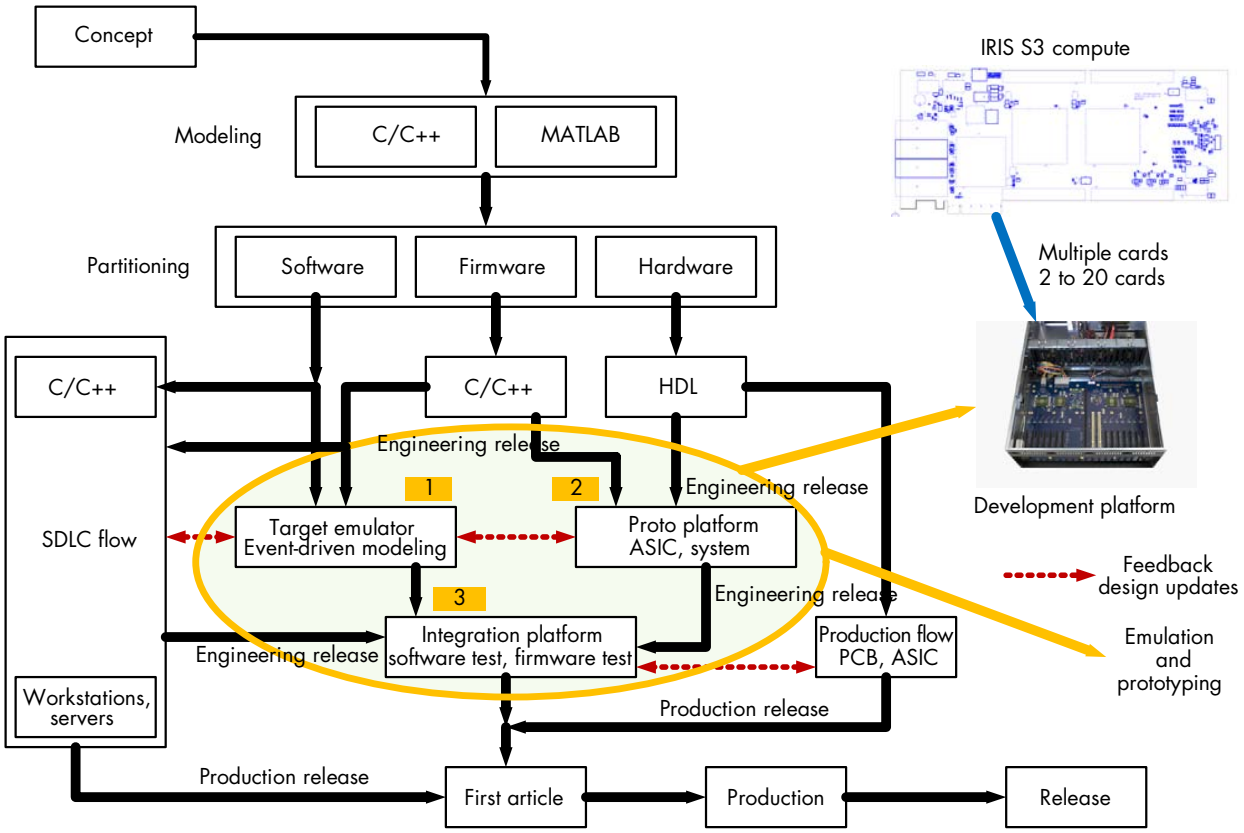


Note:
(1) This design is not a representation of the limits of the FPGAs and the platforms discussed

The block diagram in Figure 3 is a result of the partitioning stage of the engineering design flow in Figure 4. The block diagram is part of a high-level block diagram created for the MRD and HRD. Stage 1, target emulator event-driven modeling, Stage 2, proto-platform ASIC and system, and Stage 3, integration platform software test and firmware test are critical for risk reduction and staged integration of all the modules in the product development flow. They also help in the smooth integration and testing of the first article and moving it to the production and release stages.

To run parallel hardware, software, and firmware efforts, each requires a verification and validation test bed. Test benches and test environments help in functional testing and generation of golden test vectors for actual hardware testing. The integration of software and firmware cannot wait for the ASIC to be ready and verified, or for the PCB and components to be designed and tested. If any errors or modifications are required, then the process is in the fire-fighting zone.

Figure 4. High-Level Integrated Product Design and Development Flow



Spending engineering cycles on designing an incremental staged integration platform using a test bed that emulates hardware is a compelling risk reduction strategy. The platform must be easy to use, reconfigurable, and must be common between Stages 1, 2, and 3 to provide consistency, feedback, and handshaking between each of the parallel development efforts.

A flexible reconfigurable prototyping and emulation platform can address the requirements of Stages 1, 2, and 3. Stratix III FPGAs with their gate count, flexible features, and high-performance intellectual property (IP) cores fit the bill. Such a FPGA-based modular platform architecture provides many benefits:

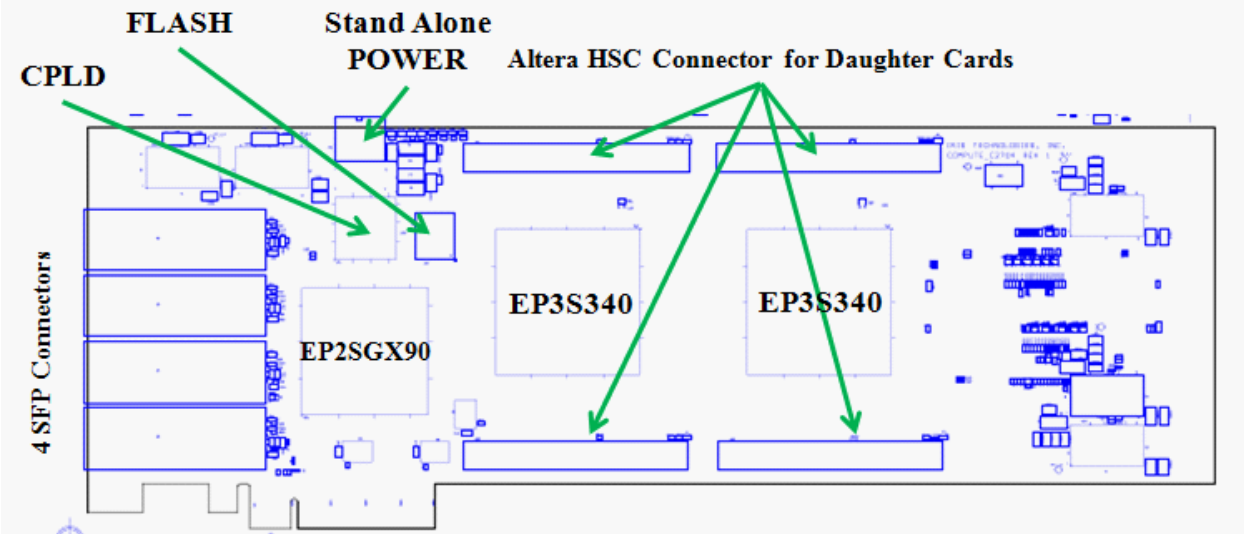
- Amortization of NRE across several projects
- Saving of internal development dollars for a one-time prototype
- No hardware updates
- Catching of bugs in software, firmware and hardware design early
- Scalable solution with building blocks
- Path to ASIC available for smaller designs that fit onto one FPGA

IRIS Technologies Inc. developed such a platform with the advantages listed above in mind. The S3 compute platform is based on the 65-nm Stratix III FPGAs and provides flexible interconnect options for mapping various I/O and multiple-use modes.

IRIS S3 Compute Board

The IRIS S3 compute card silkscreen is shown in Figure 5. The card has two Stratix III EP3SL340 FPGAs and one Stratix II GX EP2SGX90 FPGA for driving I/O and PCI Express (PCIe) interfaces.

Figure 5. S3 Compute Card Top Silkscreen



The Stratix II EP2SGX90E FPGA has 12 transceiver channels that can run up to speeds of 6.375 Gbps. Four of these transceivers are connected to small form-factor pluggable (SFP) cages that can host modules for Fiber Channel, Gigabit Ethernet, and SONET. The board has a x8 PCIe interface that runs at PCIe 1.0 speeds (providing 20 Gbps total) and PCIe 2.0 speeds (providing 40 Gbps total). The interface is an Altera PCIe core embedded as an IP block that IRIS Technologies supplies preconfigured on the board. (The core is also available from Altera by purchasing a PCIe core license or downloading an evaluation versions). The devices also provide eight designer-accessible phase-locked loops (PLLs).

The board also has two 1-Gbyte DDR2 DIMMs standard and can host 2-Gbyte or 4-Gbyte DIMMs. The HSC connectors can host third-party or IRIS daughtercards with processors, I/O for video, analog-to-digital converters (ADCs), digital-to-analog converters (DACs), radio frequency (RF), and additional volatile memory (SRAM) and non-volatile memory (flash, NAND). Custom daughtercards can also be provided or built by designers.

The Stratix III devices each contain 338,000 equivalent logic elements (LEs), which, coupled with the Stratix II GX FPGA, provide a total of 750,000 equivalent LEs. The Stratix III devices also provide 12 PLLs each, which, coupled with eight PLLs in the Stratix II GX devices, provide a total of 32 PLLs for local clock generation. The PLLs have between six and ten outputs each (six for the Stratix II GX PLL and ten for the Stratix III PLL) to give the designer the ability to generate a total of 250 local clocks.

Most of these clocks are accessible and can be forwarded through a SMA connector on the board to other boards or through the HSC connectors. IRIS Technologies provides cables that are compatible with the HSC connectors for connecting boards in a stand-alone use mode or in a system. Cables can be provided for connecting a set of boards to a desktop server or workstation through a PCIe connector in a x8 slot on the motherboard.

The IRIS S3 compute board provides features (shown in Table 1) for putting together prototyping platforms for system prototyping, ASIC prototyping, and emulation. This provides customers a flexible platform for hardware/software co-development and -verification.

Table 1. Feature Set of the IRIS S3 Compute Card (1)

Features	Number	Scalability
2 Stratix III EP3SL340 FPGAs	670,000 LEs	Interconnect boards using 4 HSMC connectors
1 Stratix II GX EP2GX90 FPGA	90,000 LEs	PCIe to host (Gen1 or Gen2 end-point or root cores)
Clocks (oscillators)	4 global single ended	Can be forwarded through HSMC connectors
Clocks (oscillators)	3 global differential pairs	Can be forwarded through HSMC connectors
Clocks (oscillators)	3 global user-selectable frequency	Can be forwarded through SMA connectors to other boards
PLLs per FPGA	12 per Stratix III FPGA, 8 per Stratix II GX FPGA, TOTAL of 32 PLLs	TOTAL of 288 clocks per board
Embedded FPGA memory per board	36.5 Million bits	Can be increased by daisy-chaining boards
DDR2 memory	4 Gbytes per board standard	Can be increased by daisy-chaining boards
Connectivity	336 pins inter board (168 LVDS), 4 SFP (optical or copper), multi-standard (SONET, GbE, WDM, FC)	Can extend chain to multiple boards

Note:

(1) Inter-FPGA communication > 5 Gbps LVDS > 10 Gbps, inter-board communication HSMC connectors: 168 LVDS channels each at 1 Gbps, PLL clocks up to 700 MHz and internal FPGA clocks > 250 MHz

Design Flow and Tools

Figure 6 shows an ASIC prototyping flow, which, with respect to Figure 4, is the HDL design flow.

Figure 6. Typical ASIC Design Flow

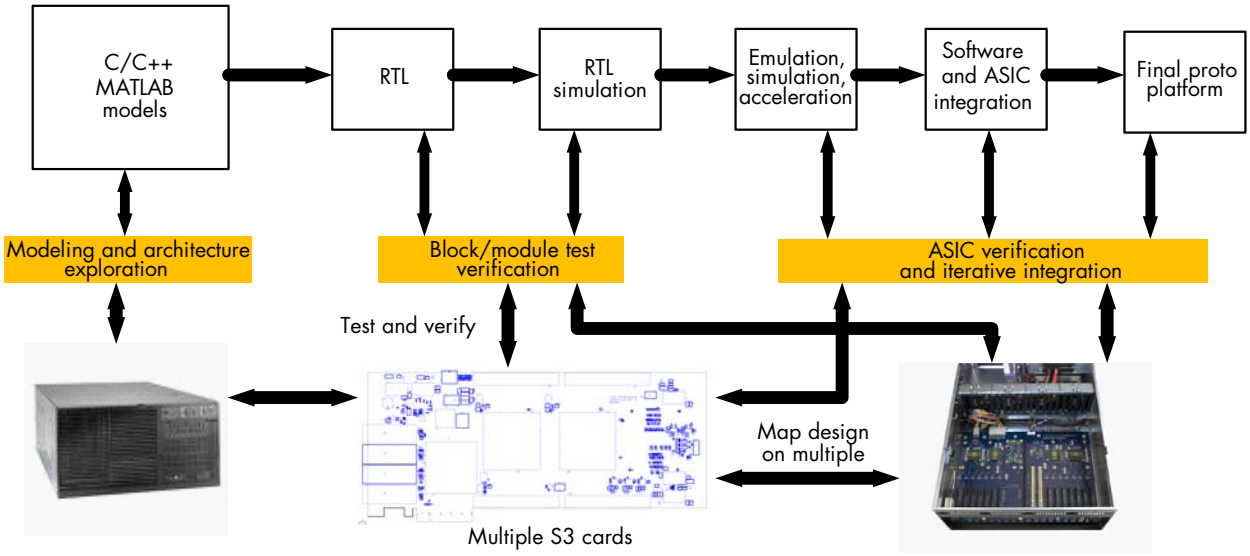
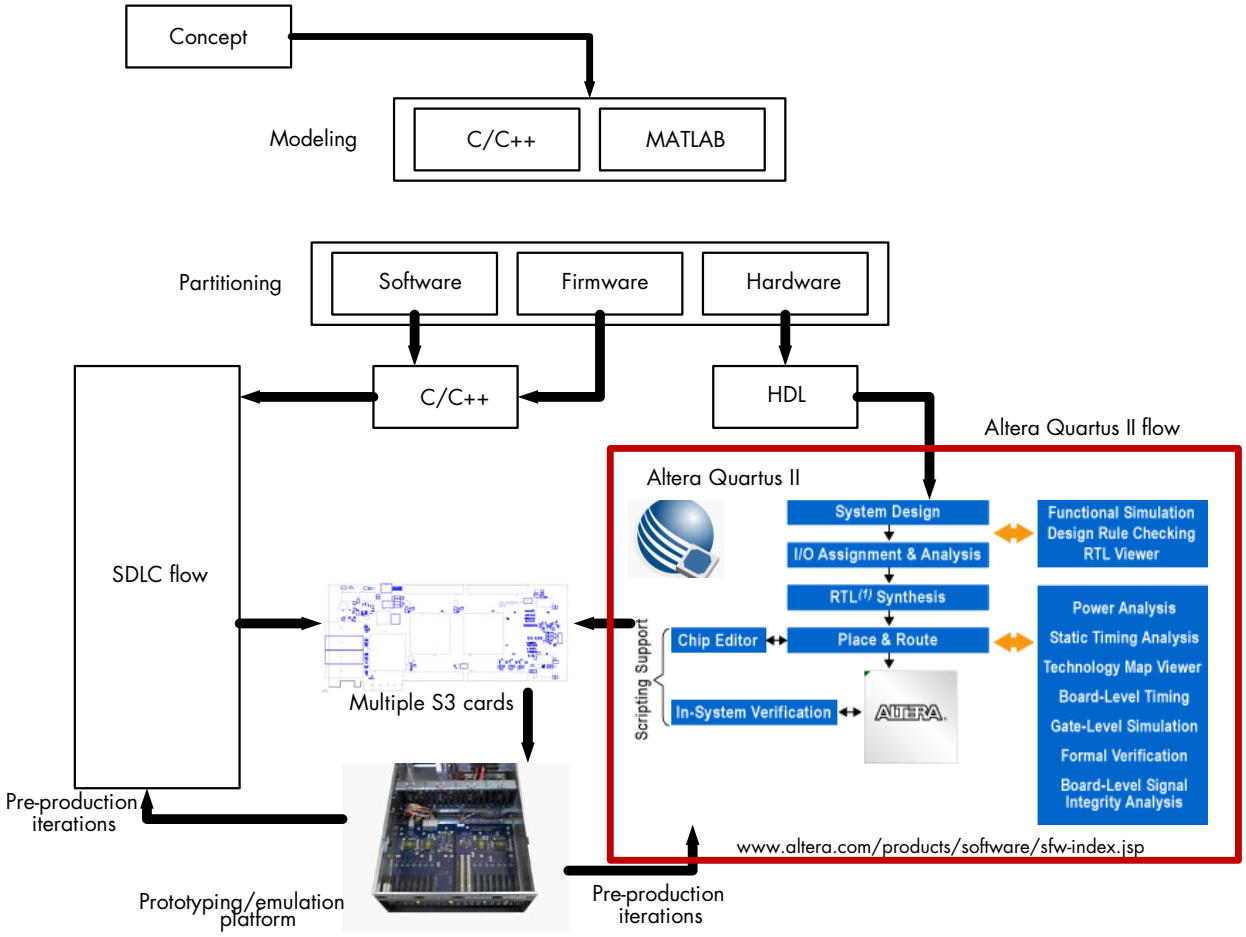


Figure 7 shows a more generic flow for an ASIC that includes system prototyping, simulation, and emulation.

Figure 7. Integrated Prototyping and Emulation Flow



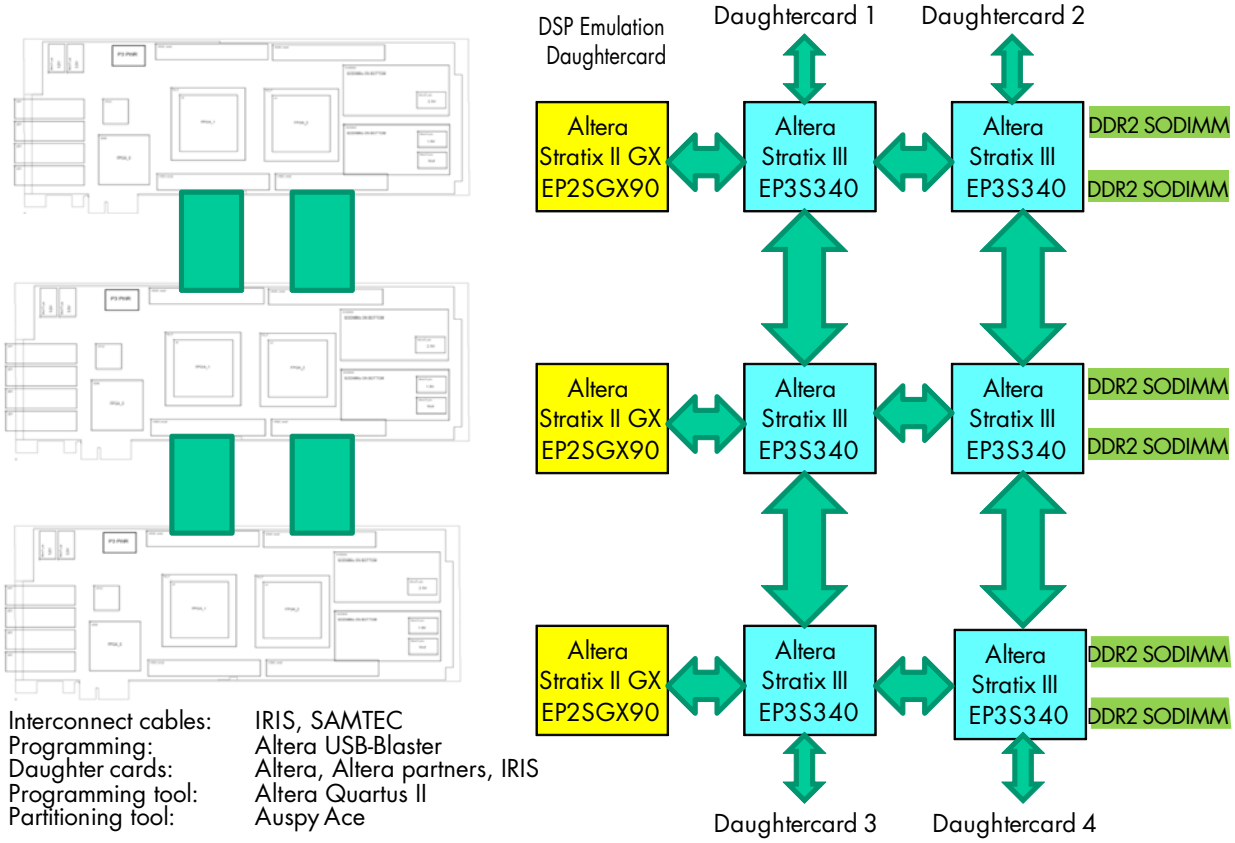
Use Modes

S3 compute cards can be used in several modes to provide the prototyping platform. For ASIC prototyping, examples are shown in Figure 8, Figure 9, Figure 10, and Figure 11. For an integrated development system, examples are shown where the cards are embedded in systems with an operating system, drivers, and cables for card interconnections.

For hardware and software co-verification, the S3 compute platform can be used to prototype the ASIC component of the example design shown in Figure 3. In addition, it can be used to prototype the DSP function as well as the FPGA function in hardware. Once a flexible platform such as the one built from multi-FPGA boards is available to software developers, they can port embedded software, as well as the application software, to incrementally test releases or builds.

This process of incrementally testing the software reduces risk of integration with the first article that comes out of the production process. The staged integration and testing also allows for anticipating problems with the hardware as well.

Figure 8. ASIC Prototyping and Emulation Stand-Alone Example 1



The most common use of FPGA-based prototyping platforms is for prototyping ASICs such as the one shown in Figure 3. For increasing the number of desired gates, boards can be daisy-chained or cross-connected using interconnect cables provided by IRIS Technologies as shown in Figure 8, Figure 9, and Figure 10, or in a system as shown in Figure 11.

The use of standard tool flows such as Quartus II software and the Nios® II design flow from Altera, along with MATLAB from The MathWorks and other C/C++ development flows, makes the process easy to adapt to and negates the necessity of learning new tools. For designers who desire to map the C code directly to HDL running on an FPGA, ESL tools such as Impulse C can be used.

Figure 9. ASIC Prototyping and Emulation Example 2

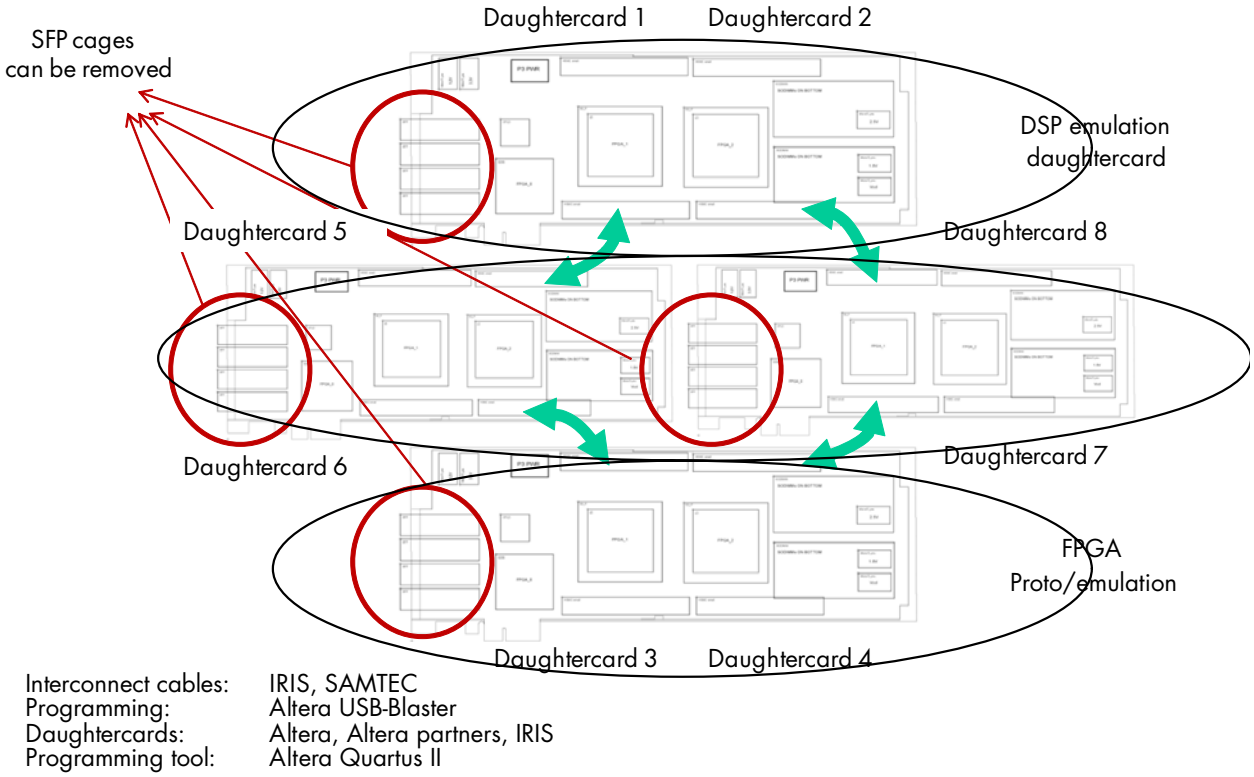
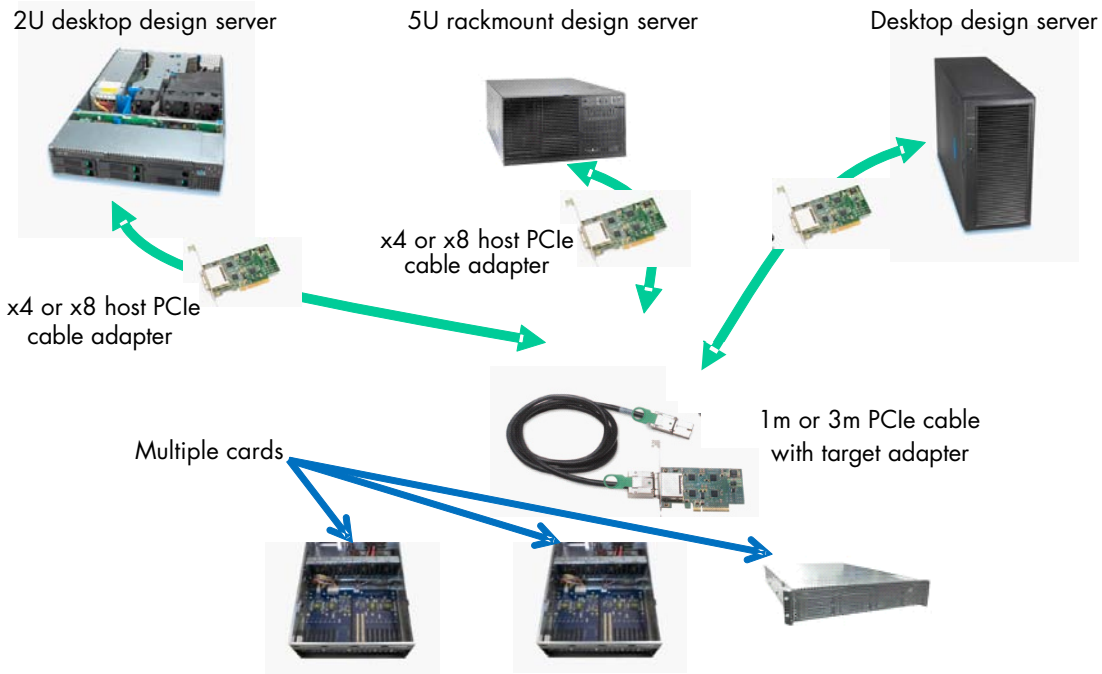


Figure 10. Enterprise-Level ASIC and Systems Prototyping, Simulation and Emulation Platform



Benefits of the S3 Card-Based Platforms

When designing a module or multi-FPGA card such as the S3 card and the platforms around it, one must consider several features. These features aid a variety of users, from small companies with limited budgets to large corporations that build multiple systems or tape out several ASICs per year. The key to this versatility is a scalable, flexible, rugged platform.

Scalability

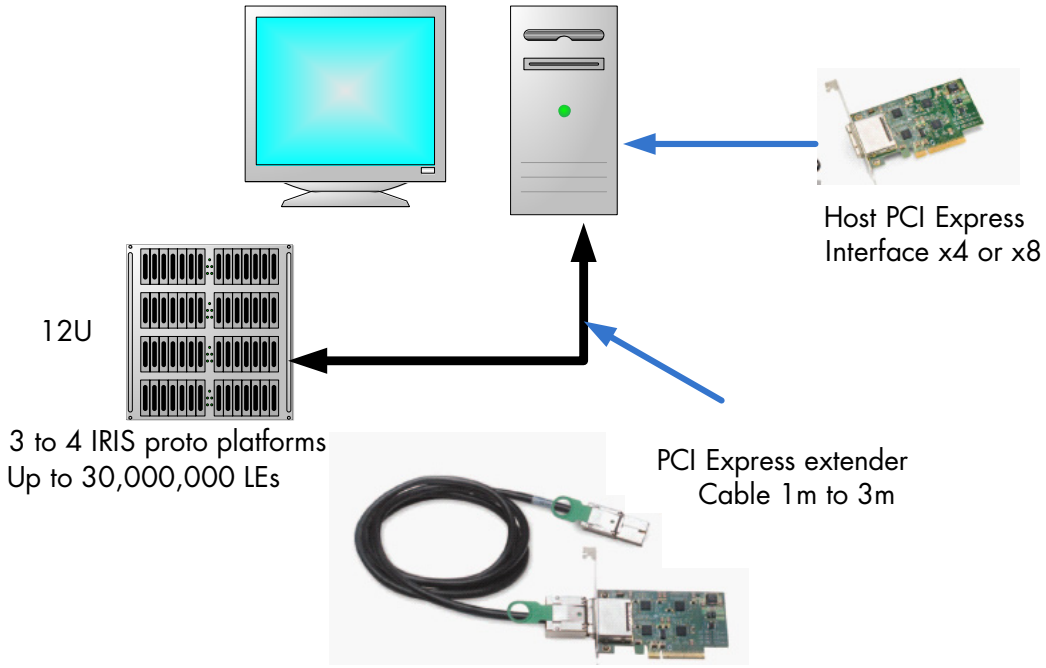
The connection flexibility provides many options to scale the design. On a single board, the cables can be crossed to provide maximum inter-FPGA connectivity to provide 680,000 LEs. The complete system-on-a-chip (SOC) ASIC can be mapped onto this fabric. Alternatively, a configuration such as those shown in Figure 10 and Figure 11 can be used.

Another scalable option is to plug a number of the cards into a standard IRIS system with inter-card communication through the HSC connectors. This configuration allows for a large number of gates to model the SOC design logic and embedded processor hardware components, while at the same time providing a clear interface to the system software.

The 4U ruggedized IRIS chassis provides a fabric capacity of 7.5 million LEs, an interface speed of 20 Gbps to any one card, and an inter-card rate of up to 25 Gbps. This gives the designer many choices on how best to partition and partition the design. The scalable aspects provide these advantages:

- Interconnect multiple cards using cables to provide more than 100-Gbps interconnect I/Os
- Plug cards into chassis for base fabric interconnectivity of 10 Gbps per card and inter-card using cables of 20 Gbps
- Cross-connect cards as for inter-card connectivity of 20 Gbps
- Choose maximum daughtercard flexibility
- Scale and stack multiple chassis using extender cables, as shown in Figure 11
- External power supplies, clock forwarding, and many local clocks per board available

Figure 11. Scalable Rugged Prototyping, Simulation, and Emulation Platform



Ruggedized and Robust

It is critical that hardware platform errors-due to loose cables, signal integrity on high-frequency designs, high-speed signals, or scalable solutions-are not introduced into rugged interconnects and configuration. Robust features include:

- No cables to trip over
- Secure interconnect options provide non-cluttered configuration
- Provides the highest gate density per chassis and PCIe-based secure cabling to chassis from host computer
- Chassis-based configurations are tested for EMI and RFI
- Chassis-based configurations have a high degree of vibration immunity as cards are mounted using the front card faceplates and card guides.
- Temperature sensors and voltage sensors
- Additional encryption key via a daughtercard
- Insertion of encryption key in CPLD for a unique ID per board

High Performance

All the Altera devices used on the S3 card can be clocked internally at 200 MHz or above for logic, and the I/O provided on the HSC connectors can be configured as LVDS or single ended. If SSL noise is a problem for large parallel busses on the HSC connectors, an LVDS configuration can be used.

All S3 cards are designed with critical I/O traces matched in length, within impedance tolerance limits, and that reference the appropriate signal planes. This provides for greater noise immunity and high-speed signals to be run intra-FPGA, board-to-board, and through the SFP modules to another board or network I/O.

Standards Based

The S3 cards are designed to PCIe specifications. The card edge connector can be used for plugging into standard servers with PCIe x8 slots, or can be left alone when cards are daisy-chained or connected through a PCIe extender cable to a workstation or server.

Expansion HSC connectors are designed to Altera's HSC connector specifications. This provides customers with access to a large system of daughtercards through Altera, third parties, and IRIS Technologies. IRIS Technologies has a roadmap for daughtercards that include DSP and data acquisition cards. Customers can also attach custom daughtercards.

I/O and Programming Flexibility

For maximum flexibility, multiple I/Os are provided for the designer, including:

- PCIe x8 connector for Gen1 throughput of 20 Gbps and Gen2 throughput of 40 Gbps
- SFP connectors for optics or copper modules for SONET, Fiber Channel, and Ethernet I/O
- Altera HSC connectors with multiple parallel I/O or LVDS pairs for use with daughtercards or board-to-board interconnect
- JTAG programmable, on-board CPLD, and flash device for programming
- Optional encryption key in CPLD for a unique ID per board

Memory

Ample memory is provided on the card itself. It can be scaled by the use of daughtercards with both volatile memory (DDRII, DDRIII, SRAM) and non-volatile (NAND and NOR flash). Each board comes standard with 1- or 2-Gbyte DDRII standard laptop SODIMMS.

Clocks and Distribution

The board has multiple onboard clocks that are distributed to each of the individual FPGAs, including an optional designer-populated clock, which by default is populated to 100 MHz. The clocks drive all the PLLs in the FPGA.

This gives the designer the ability to generate up to 288 local clocks on a single board. The clocks can also be forwarded from board to board through the HSC connectors, or through the SMA connector on the boards.

In the chassis, the parallel cabling between the HSC connectors can be used to forward the clocks. This provides the maximum flexibility of zoning clock domains for the system and subsystems such as the ASIC or the FPGA for prototyping and emulation requirements. The clocking scheme includes:

- 6 clock sources per board
- Driver output totaling 18 clocks
- 6 of the 18 clocks from a designer-selectable frequency and source
- SMA connector for clock forwarding
- 6 to 10 outputs from PLLs in the FPGA to generate local clocks
- Total of 32 PLLs

Partitioning Is Key

Partitioning the design well is key to successful design development and potential first pass integration without major changes. The partitioning problem can be complex for large systems, but generally includes:

- Data path implemented using hardware
- Control via a mix of embedded firmware for non-real time state maintenance
- Hardware-based state machines used for real-time machine state maintenance
- Event triggers provided by driving interrupts up to software, writing to registers in hardware, and setting flags
- Set flags with polling mechanisms to provide a deterministic state to update trapping mechanism and handle large asynchronous designs

Multi-FPGA Boards and Platforms

Tools such as Auspy Ace can be used to partition RTL designs for multi-FPGA boards. IRIS boards can be programmed using Quartus II design software along with Auspy ACE. Cables and interconnect options that provide flexibility in building the reconfigurable fabric are key. For larger deployments, a robust and rugged solution is necessary that provides both gate density and low power for a given footprint.

Standard Multi-Processor Systems

For emulating and simulating large machines on standard multiprocessor systems, each module can be modeled as a thread. Inter-process communication (IPC) between threads is a creative scheduling technique for thread states to be exchanged. By contrast, a robust re-configurable FPGA platform provides for easier synchronization between hardware and firmware modules through registers, and global or local busses between and within the FPGA.

Conclusion

A flexible, scalable, rugged platform that can be re-purposed for multiple projects is the key to risk-free ASIC and system development projects. FPGAs such as the Stratix III FPGAs from Altera are an ideal element on which to build. The solution from IRIS Technologies builds on the strong foundation provided by Altera tools and devices such as Stratix III FPGAs and Quartus II software. The number of equivalent logic elements, PLLs for local clock generation, and I/O options are ideally suited for building ASIC and system prototyping cards.

A robust platform such as the one presented in this paper is an example of a platform that can be built for hardware and software co-verification and co-development. The incremental integration, testing, and verification gives users the ability to anticipate problems ahead of time, a critical element in planning for a successful product release, be it an ASIC, system, or a subsystem.

Further Information

- Altera Stratix III FPGAs:
  www.altera.com/products/devices/stratix-fpgas/stratix-iii/st3-index.jsp
- Altera Stratix II GX FPGAs:
  www.altera.com/products/devices/stratix-fpgas/stratix-ii/stratix-ii-gx/overview/s2gx-overview.html
- IRIS Technologies S3 Compute:
  www.iris-technologies.net/s3_compute.php

Acknowledgements

- Gautam Kavipurapu, Founder, IRIS Technologies Inc.
- Jeff Holmbeck, Senior Strategic Alliances Manager, Altera Corporation

IRIS Technologies

ALTERA®

101 Innovation Drive
San Jose, CA 95134
www.altera.com