

Engineers' Guide to FPGA & PLD Solutions

Automotive Electronics Fuels Need for High-Reliability Devices



**FPGAs: Good Company in
Consumer Mobile Devices Seeking
Differentiation—Fast**

**Practical Tips For External
Logic Analyzer FPGA Debug**

www.eecatalog.com/fpga

EECatalog

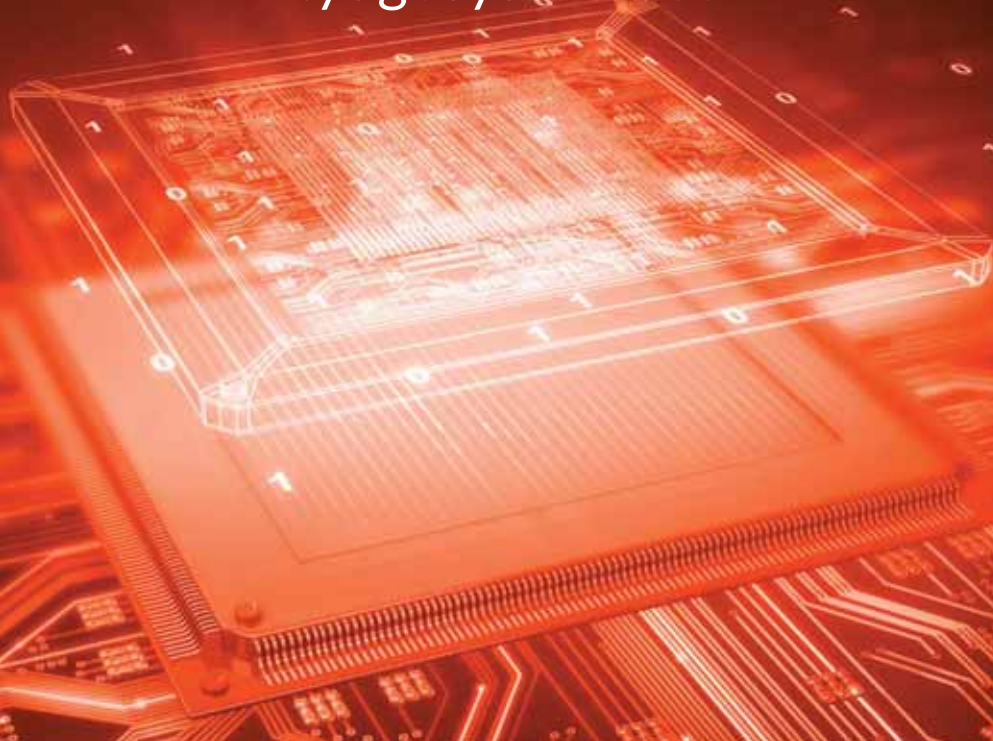
Market Leader Sponsor

Gold Sponsors



World First MGT*^{*}-based FPGA Debug

www.yugosystems.com



EXOSTIV™
APPLICATION

USB 3.0
Super Speed



EXOSTIV™
PROBE

Up to
4 x 6.6 Gb/s



EXOSTIV™
IP

Observability Reloaded

Up to 8GB storage

Up to 32.768 nodes

FPGA DEBUG RELOADED

* Multi-Gigabit Transceiver

SUPERIOR **EMBEDDED** SOLUTIONS



DESIGN YOUR SOLUTION TODAY
CALL 480-837-5200

www.embeddedARM.com

NEW! TS-4740 Computer Module
High Capacity 25 KLut FPGA & GbEth



Pricing Starts At
\$176
Qty 100
\$222
Qty 1

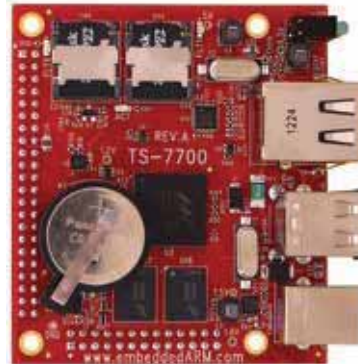
- 1 GHz Marvell PXA168 ARM CPU
- 512 MB DDR3 RAM and 4 GB eMMC Flash
- 25 KLut FPGA, 1x Gigabit Ethernet, 1x 10/100 Ethernet
- 1x microSD card, 1x USB Host, 1x USB OTG
- 121x DIO, 1x I2C, 1x SPI, 1x CAN, 6x UARTs
- -40 °C to 85 °C Industrial Temperature Range
- Supports video acceleration up to 720p
- Boots Debian Wheezy with Linux Kernel 2.6.34

Computer-on-Modules

State of the Art Embedded Design

- TS-4600: 450 MHz low cost w/ 2 Ethernets
- TS-4710: Up to 1 GHz PXA168 w/ video
- TS-4720: Like TS-4710 + 2 GB eMMC Flash & 2 Ethernets
- TS-4200: Atmel ARM9 w/ super low power
- TS-4800: 800 MHz FreeScale iMX515 w/ video

NEW! TS-7700 Single Board Computer
High Performance Compact Computer



Pricing Starts At
\$119
Qty 100
\$165
Qty 1



Available with TS-ENC750 enclosure
and TS-752 interface board

Features:

- Up to 1 GHz ARM CPU
- 512 MB RAM
- 8 KLut FPGA
- 2x microSD Socket
- 1x 10/100 Ethernet
- 2x USB Host
- 128 MB NVRAM
- 6x Serial Ports
- 55x DIO, 1x CAN
- 1x SPI, 1x I2C, 1x I2S

Benefits:

- Hardware flexibility with on-board FPGA
- Several control I/O interfaces
- Launches your application in under a second
- Easy development w/ Debian and Linux 2.6.34
- High data reliability with 2x microSD RAID storage
- -40 °C to 85 °C industrial temperature range



We've never discontinued a product in 30 years



Embedded systems that are built to endure



Support every step of the way with open source vision



Unique embedded solutions add value for our customers

www.embeddedARM.com

Welcome to the 2015 Engineers' Guide to FPGA and PLD Solutions

It's always a good thing when answers to questions don't just set out the facts but go further, pointing out connections or setting details into an historical context. Synopsys's Troy Scott does just that in this issue. (See "A Few Questions on...FPGA-based Prototyping Software Tools.") For example, he compares the culture change he believes must take place as development teams adopt Design for Productivity (DFP) best practices with the earlier shift from "an emphasis on ASIC design productivity to ASIC verification productivity."

Anything that's large and unable to say no to hefty power meals doesn't sound right for mobile designs. However, you'll read in an article by Mauri Delostrinos of Lattice Semiconductor that the emergence of low-gate-count devices as small as 1.4 x 1.48mm that use power sparingly and are inexpensive has opened the way for FPGAs to have a companion chip role in mobile devices. Delostrinos explains why "Mobile Industry Processor Interface (MIPI) displays are a major application area for FPGAs."

On the topic of mobile as in automobile, Dr. Raik Brinkmann of OneSpin Solutions argues that for automotive safety critical design, the "traditional approach of running an FPGA design on the actual hardware to provide a functional testing environment cannot satisfy the verification needs of the standards...." His very thorough article also addresses such topics as the relationship between aggressive design optimizations and error introduction.

One relationship 4DSP's Jason Cella discusses for us is that involving increased mobile broadband traffic and the limited amount of available wireless spectrum—an unhealthy relationship when the result is call interference. But adaptive beamforming—and FPGAs—can help.

Also in this issue, WIN Enterprises looks at what's helpful to high-performance computing projects, Keysight Technologies acknowledges that "probing the FPGA external I/O signals that are part of the real design" can be challenging and BittWare's Jeff Milrod shares his perspective on prototyping.



Anne Fisher

Anne Fisher

Managing Editor

P.S. To subscribe to our series of Engineers' Guides for embedded developers and engineers, visit: www.eecatalog.com/subscribe

Engineers' Guide to FPGA and PLD Solutions 2015

www.eecatalog.com/fpga

Vice President & Publisher

Clair Bright
cbright@extensionmedia.com
(415) 255-0390 ext. 15

Editorial

Editor-in-Chief

Chris A. Ciuffo
cciufo@extensionmedia.com

Managing Editor

Anne Fisher
afisher@extensionmedia.com

Creative/Production Production Manager

Spryde Heithecker

Media Coordinator

Yishian Yao

Graphic Designers

Nicky Jacobson
Caldin Seides

Senior Web Developer

Mariam Moattari

Advertising/Reprint Sales

Vice President & Publisher

Embedded Electronics Media Group
Clair Bright
cbright@extensionmedia.com
(415) 255-0390 ext. 15

Sales Manager

Michael Cloward
mcloward@extensionmedia.com
(415) 255-0390 ext. 17

Marketing/Circulation

Jenna Johnson

To Subscribe

www.eecatalog.com/subscribe

Extension

MEDIA

Extension Media, LLC

Corporate Office

(415)255-0390

President and Group Publisher

Vince Ridley
vridley@extensionmedia.com

Vice President, Sales

Embedded Electronics Media Group
Clair Bright
cbright@extensionmedia.com

Vice President, Business Development

Melissa Sterling
msterling@extensionmedia.com
(415)-970-19100

Special Thanks to Our Sponsors



The Engineers' Guide to FPGA and PLD Solutions is published by Extension Media LLC. Extension Media makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this Catalog nor does it make a commitment to update the information contained herein. The Engineers' Guide to FPGA and PLD Solutions is Copyright ©2014 Extension Media LLC. No information in this Catalog may be reproduced without expressed written permission from Extension Media @ 1786 18th Street, San Francisco, CA 94107-2343.

All registered trademarks and trademarks included in this Catalog are held by their respective companies. Every attempt was made to include all trademarks and registered trademarks where indicated by their companies.

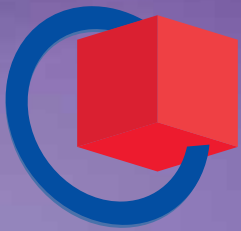
EECatalog

**Extension
M E D I A**

Register and
get your tickets now!
embedded-world.de

Nürnberg, Germany

February 24 – 26, 2015



embedded world 2015

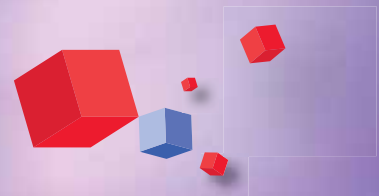
Exhibition & Conference

... it's a smarter world

THE gathering for the embedded community!

The world's largest event for embedded technologies brings together all the players in the embedded sector for a productive conversation.

Make sure you are there when contacts and networks are being cemented on the international level and new trends are being set.



Media partners

elektroniknet.de

computer-automation.de

energie-und-technik.de

MEDIZIN-UND-elektronik.DE

Markt & Technik
DIE UNABHÄNGIGE WOCHENZEITUNG FÜR ELEKTRONIK

**DESIGN &
ELEKTRONIK**
KNOW-HOW FÜR ENTWICKLER

elektroniknet.de
Elektronik
Fachmedium für industrielle Anwender und Entwickler

Elektronik
automotive
Fachmedium für professionelle Automobilindustrie

**ENERGIE
& TECHNIK**
Fachmedium für Energieeffizienz

Computers
AUTOMATION
Fachmedium für Automatisierungstechnik

MEDIZIN & elektronik
Fachmedium für Elektronik in der Medizin

Trade fair organizer

NürnbergMesse GmbH

Tel +49 (0) 9 11.86 06-49 12

visitorservice@nuernbergmesse.de

Congress organizer

WEKA FACHMEDIEN GmbH

Tel +49 (0) 89.2 55 56-13 49

info@embedded-world.eu

NÜRNBERG MESSE

CONTENTS

Automotive Electronics Fuels Need for High-Reliability Devices <i>By Dr. Raik Brinkmann, OneSpin Solutions</i>	5
FPGAs: Good Company in Consumer Mobile Devices Seeking Differentiation—Fast <i>By Mauri Delostrinos, Lattice Semiconductor</i>	8
Practical Tips For External Logic Analyzer FPGA Debug <i>By Brad Frieden, Keysight Technologies</i>	11
A Few Questions on...FPGA-based Prototyping Software Tools <i>By Anne Fisher, Managing Editor</i>	15
Don't Swallow the Camel and Other FPGA Wisdom <i>By Anne Fisher, Managing Editor</i>	18
The FPGA Advantage for Beamforming <i>By Jason Cella, 4DSP, LLC</i>	20
Once Earmarked for FPGA Always Earmarked for FPGA? <i>By Todd Sirois, WIN Enterprises</i>	22

PRODUCT SERVICES

FPGAs & PLDs

FPGA Chips

Xilinx, Inc.

<i>7 series FPGA Portfolio</i>	25
<i>20 nm UltraScale Devices</i>	26

FPGA and SoC Chips

Xilinx, Inc.

<i>Xilinx Low-End Portfolio</i>	27
---------------------------------------	----

SoC

Xilinx, Inc.

<i>Zynq-7000 All Programmable SoC</i>	28
---	----

Boards & Kits

Design Kits

Xilinx, Inc.

<i>Kintex UltraScale KCU105 Evaluation Kit</i>	29
<i>Zynq-7000 All Programmable</i>	
<i>SoC ZC702 & ZC706 Evaluation Kits</i>	30

Development Tools

Design Tools

Xilinx, Inc.

<i>Vivado Design Suite</i>	31
----------------------------------	----

Verification Tools

Development Tools

Yugo Systems

<i>EXOSTIV™</i>	32
-----------------------	----



Cover image courtesy of Daimler Mercedes-Benz

Automotive Electronics Fuels Need for High-Reliability Devices

Already working on behalf of custom automotive devices, verification tools based on formal methods are now helping put FPGAs in the driver's seat—can mil/aero, transportation, power generation and other safety-critical areas be far behind?

By Dr. Raik Brinkmann, OneSpin Solutions

Safety critical design of automotive electronics, including those using FPGAs, falls under the new ISO 26262 standard. The need for more complex functions and high performance in an ultra-reliable environment plays a substantial role in automotive embedded system design.

Field Programmable Gate Arrays (FPGAs) offer flexibility and density at affordable implementation cost, so it is not surprising that the use of FPGAs in automotive systems is expanding. With custom devices being expensive to produce, many design teams resorted to Micro Controller Units (MCUs) for many functions. FPGAs offer an attractive alternative to a software only functional model, while retaining the design cost benefits of the MCU. In addition, modern FPGAs contain convenient hardware functions useful in automotive applications, and may also be updated in the field. Another area where FPGAs shine is in boosting performance for compute-intensive automotive applications such as Advanced Driver Assistance Systems (ADAS).

The traditional approach of running an FPGA design on the actual hardware to provide a functional testing environment cannot satisfy the verification needs of the standards, and using simulation only improves the situation slightly. Developers have been using verification tools based on formal methods for custom automotive devices. It's an approach that can meet FPGA needs as well.

The Impact of Automotive Failsafe Requirements

Today the simplest of modern vehicles will contain a number of processors, and this runs into the hundreds of compute elements for high-end vehicles. Electronics are present throughout the safety critical components in the car, aid the driver in its safe operation, and introduce a new level of comfort. However, if one of these critical systems fails during operation, the result is catastrophic. As such, standards have been ratified, such as ISO 26262, which lay down design and verification metrics that must be followed for these devices to be employed by automakers.

For example, the highest safety standard (or "Automotive Safety Integrity Level") defined in ISO 26262 is ASIL-D, and this sets the required likelihood of malfunction to a

Triple Modular Redundancy

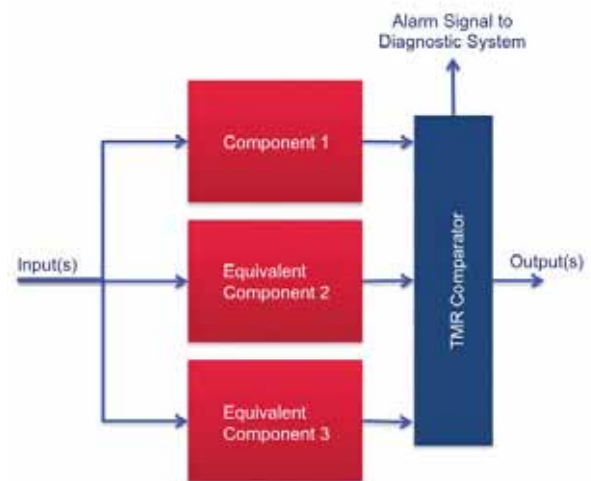


Figure 1. For critical areas of logic, Triple Modular Redundancy (TMR) is sometimes leveraged.

statistically defined failure rate of 10⁻⁹ per hour, a staggering 1 in every 114,155 years. Furthermore, these requirements must be measured on the final gate level representation of the device, not the Register Transfer Level (RTL) used for design and simulation, and the failure introduced during testing without any additional hardware being incorporated on the device. To ensure that these metrics may be met, engineers add additional failsafe structures into their designs.

To take another case, in the event of rare, spurious memory data errors, error correction code mechanisms are used where data is encoded when written into memory, and decoded on a read. Any errors caused by external factors are picked up and corrected using this method. Although it's a significant overhead to added error correcting codes on key RAMs, this method guarantees against memory defects.

FPGA Verification Flow

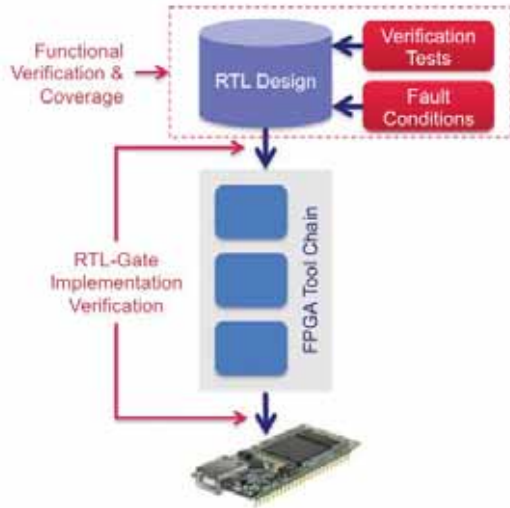


Figure 2. Test methods must be even more rigorous than existing verification practices.

Similarly for critical areas of logic, Triple Modular Redundancy (TMR) is sometimes leveraged. See Figure 1. Instead of just one logic block, three are employed to perform the same function. The output of two of the blocks is continuously compared and if there is a difference, the third block is used to arbitrate between the other two. An alarm bit is also raised on a difference, and if this alarm bit occurs frequently, the overall safety diagnostic system will flag the device for future replacement.

The Verification of Safety Critical Designs

The verification of overall device design and implementation, together with the validation that these safety systems operate correctly, of course requires test methods even more rigorous than existing verification practices. (See Figure 2.) The exhaustive and rigorous nature of the technology makes formal verification techniques a natural choice for these devices.

Bugs introduced through design error, or by the tool chain during implementation, must be eliminated by thorough verification. To ensure that the verification environment is properly qualified for this purpose, it's essential to employ high quality verification coverage. The rate of verification coverage, that is the proportion of the design proven to have been tested, must be as close to 100% as possible, and this metric must be produced somewhat independently of the verification tools employed.

Techniques for testing coverage by analyzing the ability of the verification environment to detect errors in any part of the design code have become established as a mechanism for producing this metric. A number of automotive electronics companies now use Observation Coverage that employs an exhaustive formal-based approach to understand if a design change will trigger verification checks.

Observation Coverage uses a mechanism where the design code is temporarily altered to see if the checks within the verification environment flag these changes. By manipulating these alterations and using the exhaustive nature of the formal techniques, it is possible to establish a precise test metric for the entire design, without the overhead of some similar methods.

Aggressive Design Optimizations

While Observation Coverage assists in proving that design verification has been properly carried out, in an FPGA design additional verification is required to ensure that the design described at the RTL level has properly passed through the synthesis and place and route tool chain without the introduction of additional bugs. For FPGAs, this is particularly important due to the advanced nature of FPGA synthesis. FPGA synthesis targets fixed device fabrics. To produce the highest quality design, they employ a range of aggressive design optimizations. On occasion these optimizations may introduce errors of their own.

Equivalency Checking (EC) tools that use formal verification to exhaustively compare RTL descriptions against resulting gate level code are commonplace in the ASIC world, but are new to FPGAs. However, they are rapidly being employed on large FPGA designs to counter the time it takes to weed out tool chain errors. For FPGAs, specialized EC tools are required

Inserting Faults In Memory Test

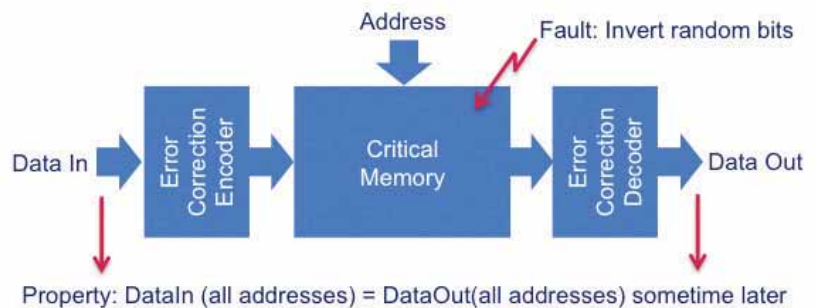


Figure 3. Formal methods offer a mechanism for introducing faults without changing the design code.

that can support the complex sequential nature of the synthesis optimizations. By employing FPGA EC, engineers can safely leverage these optimizations to produce the highest quality design possible, with confidence that bugs will not be introduced. For Safety Critical designs EC usage goes further by proving that the tool chain has not introduced errors after the RTL has been fully validated.

Understanding Forward and Reverse Mapping

For ISO 26262 there is another reason why these tools are required. The verification of faulty device scenarios must be carried out on the final gate level design, not the RTL code. As such, understanding the forward and reverse mapping of the RTL to gate design such that test faults may be properly inserted and results interpreted is key, and this may also be accomplished with these EC Tools.

To test for the system’s ability to recover or absorb faults, a methodology must be leveraged that allows these faults to be introduced without changing the design code, and the correct operation observed. Formal methods provide an easy mechanism for this, (Figure 3). Properties may be written that specify the correct operation of the system. Faults are then injected at various intervals and locations using formal

constraints during verification, and the properties examined to ensure they still hold true. If they do, the system has been proven to respond correctly to these faults.

Automotive FPGAs Here to Stay

Custom hardware devices clearly have advantages over MCUs for many automotive applications but to produce an ASIC often cannot be justified given the expected volumes and design effort required. FPGAs fit this need perfectly and enable a number of unique capabilities especially useful in this environment. However, the testing of these devices has to meet safety critical hardware standards, introducing design and verification overheads. Formal Methods provide an effective way to meet the requirements of the standards for FPGAs, driving the use of the tools in this application, as well as other safety critical areas such as the aerospace, power generation, and defense industries. Using a formal tool based flow improves design quality, return on investment and time-to-market.

Dr. Raik Brinkmann is President and Chief Executive Officer of OneSpin Solutions.



FPGA and PLD Solutions ONLINE

www.eecatalog.com/fpga



Explore...

- Top Stories and News
- White Papers
- Expert Opinions (Blogs)
- Exclusive Videos
- Valuable Articles

Sign up for the FPGA Quarterly Report email newsletter

→ www.eecatalog.com/fpga

FPGAs: Good Company in Consumer Mobile Devices Seeking Differentiation—Fast

Why a companion chip role for today's smaller, less power hungry FPGAs serves smartphones, smart watches, wearables and to-be-imagined mobile consumer devices well.

By Mauri Delostrinos, Lattice Semiconductor

Apple released its first generation iPhone on June 29, 2007; now we are using seventh-generation iPhone 5C and iPhone 5S products that have been available since last year. As I write this, news is breaking that Apple has sent out invites to an event that the company is hosting in Cupertino on September 9, with the tagline “Wish we could say more” to discuss the iPhone 6. The “i” brand—starting with iPods and then iPads before the emergence of the iPhone—has become so successful, first as an aspirational product, then as an ubiquitous item, that 100s of many different “i-__” products have been produced.

Similarly, a quick Internet search identifies that since 2009, Samsung has launched more than 100 smartphones, tablets, phablets, cameras and watches that bear the Galaxy brand alone. Products from other big name players such as LG, Sony and HTC also proliferate in the market, and then there are emerging makers in Asia looking to provide similar products, usually at a lower price premium.

What these well-known market research snippets ably demonstrate is the short lifecycle of any one product generation. Also in evidence: the reduced profitability window that any manufacturer of such consumer items has available to exploit and the essential speed of new product development. Also, with the huge choice on offer to consumers, manufacturers must constantly evolve new features and functions to differentiate themselves in a very competitive marketplace.

The Re-spin Conundrum

Early generation products in this sector used an architecture based on a processor and an application-specific signal processor (ASSP). This is a valid approach, especially in a cost-sensitive market. However, there are two major problems given the need for fast development and product differentiation. Developing any form of ASIC or ASSP-based solution requires a great deal of effort based on early marketing decisions, which can be costly and time-consuming to change in response to evolving competitor and consumer influences. Second, processors have limited ability to handle different I/O, memory types, display and sensor interfaces. Therefore, if the design calls for, say, a different type of sensor, either you'll need to change your processor or manage some form of bridging solution in an ASIC re-spin.

For these reasons—chiefly centering on speed of development and flexibility—manufacturers are now increasingly implementing mobile designs that use FPGAs in a companion chip role. Traditionally, FPGAs would have been considered too big, too expensive and too power-hungry for mobile consumer applications. However, with the advent of low gate count devices that are as small as 1.4 x 1.48mm, consume as little as 21 μ W and cost only 50 cents the picture has changed. For example, a Chipworks teardown identifies a Lattice FPGA inside Samsung's Galaxy S5.



Figure 1: The dramatic shift from FPGA (28 nm Virtex-7) to programmable ASIC-like SoC (20 nm Virtex Ultrascale).

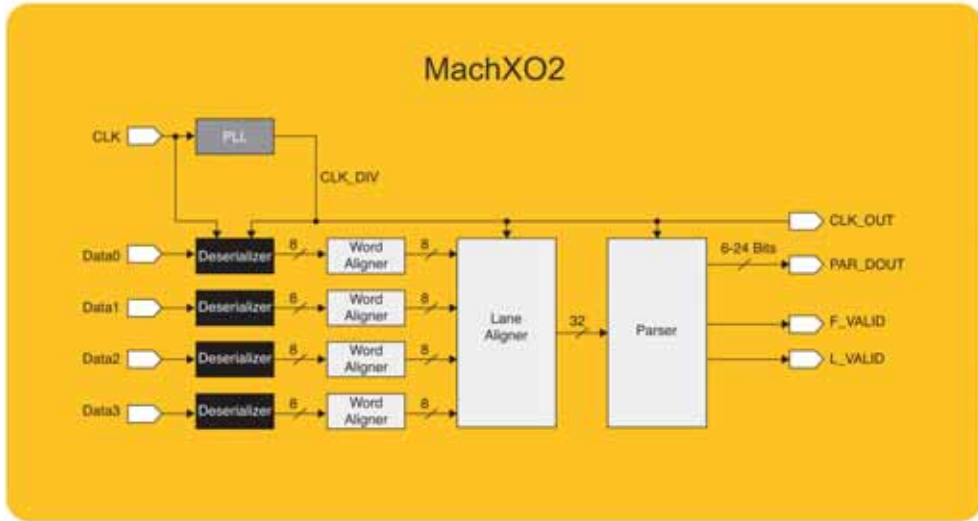


Figure 2. CSI2 Bridge System Block Diagram

Easily Switching Display Types

Within this context of the arrival of smaller and less expensive FPGAs, Mobile Industry Processor Interface (MIPI) displays are a major application area for FPGAs. The majority of image sensors in the consumer market use the MIPI CSI2 interface. The MIPI has become the interface standard for the majority of components in consumer mobile devices. Camera Serial Interface 2 (CSI2) is the MIPI interface specification focused specifically on cameras. Often the ASSP used in smartphone and especially wearable electronics does not have a CSI2 interface. FPGAs can perform a bridging function to convert from CSI2 to parallel CMOS, enabling the manufacturer to easily switch display types and suppliers as is commercially advantageous.

Figure 1 shows this CSI2 to CMOS parallel function being performed by a member of Lattice’s MachXO2 FPGA family.

The CSI2 Bridge converts the CSI2 interface to a parallel sensor interface for an ISP. True LVDS input pads on the MachXO2 device handle the 200 mV common mode voltage of the MIPI DPHY high-speed interface. The CSI2 interface from the image sensor can be 1, 2 or 4 data lanes. To keep the FPGA density small, the CSI2 bridge is typically synthesized for a single CSI2 format. In most embedded applications the image sensor is typically configured for a single CSI2 output format at all times. However, multiple CSI2 formats can be supported for “on-the-fly” switching by adding multiple instantiations of the mipi_csi2_serial2parallel NGO in each

desired format. Figure 2 depicts the system block diagram.

Support for Custom Functionality Efforts

It is tempting to be skeptical about how much can usefully be achieved using small—both in terms of actual size and gate count—FPGAs. In fact, many features such as IR Tx/Rx Control, Bar Code Emulation, Pedometer, Activity Monitoring, Sensor Pre-Processing and LED Control can be successfully implemented in devices such as Lattice’s newly introduced iCE40 Ultra FPGA family. Integration of high current sink LED drivers, multipliers and accumulators

optimizes custom function implementation, standard serial interfaces such as SPI & I2C and a whole host more of hardened IP. This ASSP-like integration reduces system power and speeds implementation so designers can spend more time on implementing their custom functionality.

FPGAs can take on many other functions in the consumer space. For example, they can be used to capture LVDS video data at high speed and process it using the on-chip sysDSP

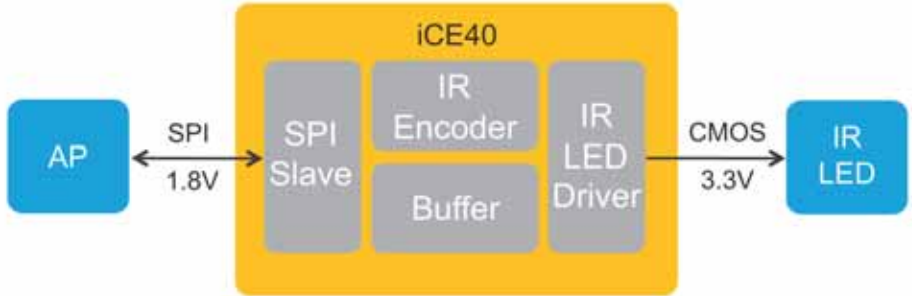


Figure 3. FPGA enables timing-critical IR-LED control.

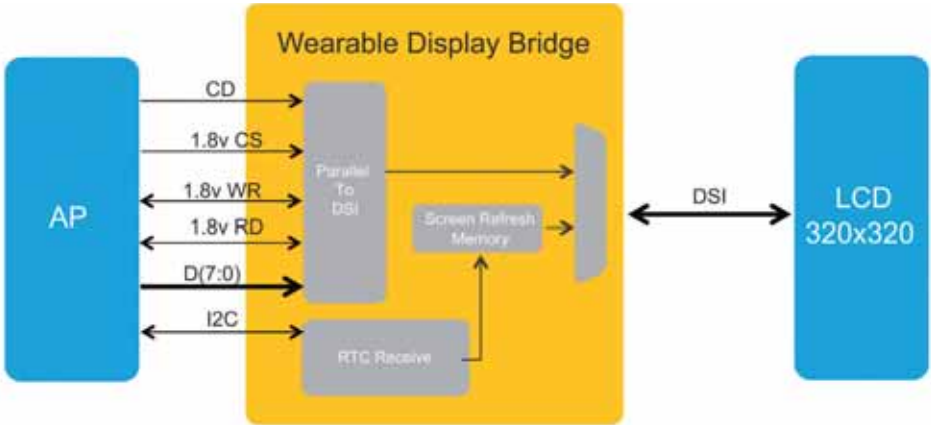


Figure 4. Wearable Display Bridge – Standard Parallel Bus to single to MIPI DSI

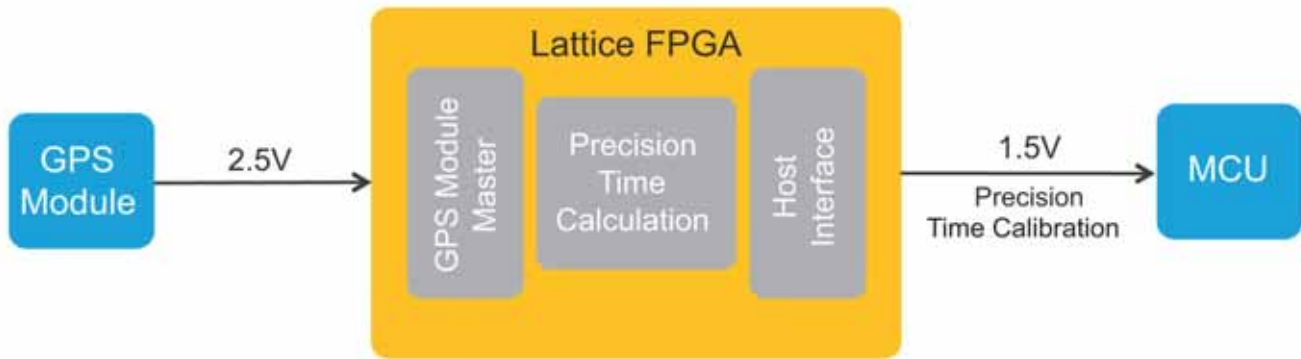


Figure 5: Smart Watch—Auto Time Calibration + IO Bridge

block and embedded RAM. Another application adds a universal remote function within a smartphone using just a tiny iCE40 FPGA as shown in Figure 3.

Figure 4 details a standard parallel bus to single to MIPI DSI bridge that improves battery life in wearable displays by enabling the applications processor to remain in sleep mode for longer.

Figure 5 shows a design for a smart watch. In this application the FPGA is delivering an auto time calibration + IO bridge function. By so doing the FPGA overcomes the problem that some MCUs do not support 2.5V IOs, needed to interface with GPS modules enabling the watch to automatically reset time when the user travels to a new time zone.

FPGA flexibility enables the imagination to run riot. Smart glasses can include branding held in simple code, and intelligent lighting effects—whether user, context or sensor stimulated—can be simply realized using LEDs.

Designers working in the consumer market who have not considered using field programmable devices before may have concerns about the design methodology. Support is available in the form of tools for the following:

- Design Entry
- Synthesis
- Implementation
- Analysis
- On-chip Debug Hardware Analysis
- Simulation
- Programming
- Deployment

Support for third-party tools is also available.

Therefore, designers can develop, run and simulate their RTL code, then run the code and validate it in hardware. Software developers like this design methodology too as it enables C/C++ to be supported with soft-processor IP doing way with the need for developers to learn a new hardware language in order to implement a solution.

Lattice Diamond supports VHDL, Verilog, EDIF, schematics and multiple implementations. It also features an easy to use GUI, but as a script is sometimes the fastest way to do a task, full Tcl scripting support is also provided.

Conclusion

New generation FPGAs that have been architected for low power and small size, targeting mobile consumer applications, are currently being used in high-volume applications and are proving beneficial in adding flexibility to the development of products such as smartphones, tablets, eBooks and wearable electronics. Alongside devices that are fully uncommitted, FPGAs are also starting to be introduced that incorporate hardened features—memory, I/O, display and sensor interfaces, SERDES—which combine the flexibility of programmability with the efficiency of commonly demanded features. Either way, designers who have previously avoided taking the FPGA path may wish to reconsider.

Mauri Delostrinos is currently Consumer Account Manager at Lattice Semiconductor based in San Jose, California. He is responsible for developing strategic planning for selected Consumer Accounts and for supporting major consumer device manufacturers in the Bay Area and beyond. A graduate of the Stanford University School of Business, Delostrinos has also held Applications and Field Applications Engineering roles with Lattice, during which time he participated in Product Definitions for next generation FPGA families and supported key accounts and leading consumer companies in the Silicon Valley.



Practical Tips For External Logic Analyzer FPGA Debug

Already working on behalf of custom automotive devices, verification tools based on formal methods are now helping put FPGAs in the driver's seat—can mil/aero, transportation, power generation and other safety-critical areas be far behind?

By Brad Frieden, Keysight Technologies

Today's design tools provided by FPGA vendors offer powerful debug capabilities through a variety of debug cores. Integrated logic analyzers allow for a designer to probe many internal FPGA nets and store captured signal traces in block RAM for analysis and viewing. Although these "inside the chip" tools are very useful, there are times when the deep memory capture available in an external logic analyzer or mixed-signal oscilloscope can be very helpful to further discover the root cause of the failure in a prototype, or to view a particular operation in a design, such as capturing one complete video frame in a video processing design. Powerful debug-oriented trigger techniques are also helpful to trap bugs. This article will explore techniques for all of these topics.

Routing Signals Out of an FPGA Design

Consider an example Xilinx Zynq 7020 system on chip (SoC) Vivado demo design where a five-bit "one-hot" state machine (shown in Figure 1) represents a process of moving data. The

state machine should go through the "idle" state, the "start" state, "address" state, "data" state where data should be moved, and then reach the "waiting for acknowledge" state. Once the data is received, an acknowledge signal should come back. But in our demo scenario, something's going wrong, and only part of the data is getting transferred.

Understandably, some key signals of interest include the "state" five-bit parallel bus, the acknowledge bit and a timeout bit that gets set if something goes wrong in the design. Sometimes a prototype system hasn't been laid out on a PC board so that there is easy access to the real I/O signals, but there may be some FPGA pins dedicated for debug and routed to a header or connector. How does one get nets of interest, in this case, inside the FPGA I/O ring, routed to that debug connection point? Those desired nets are shown in Figure 1 with blue circles and an oval.

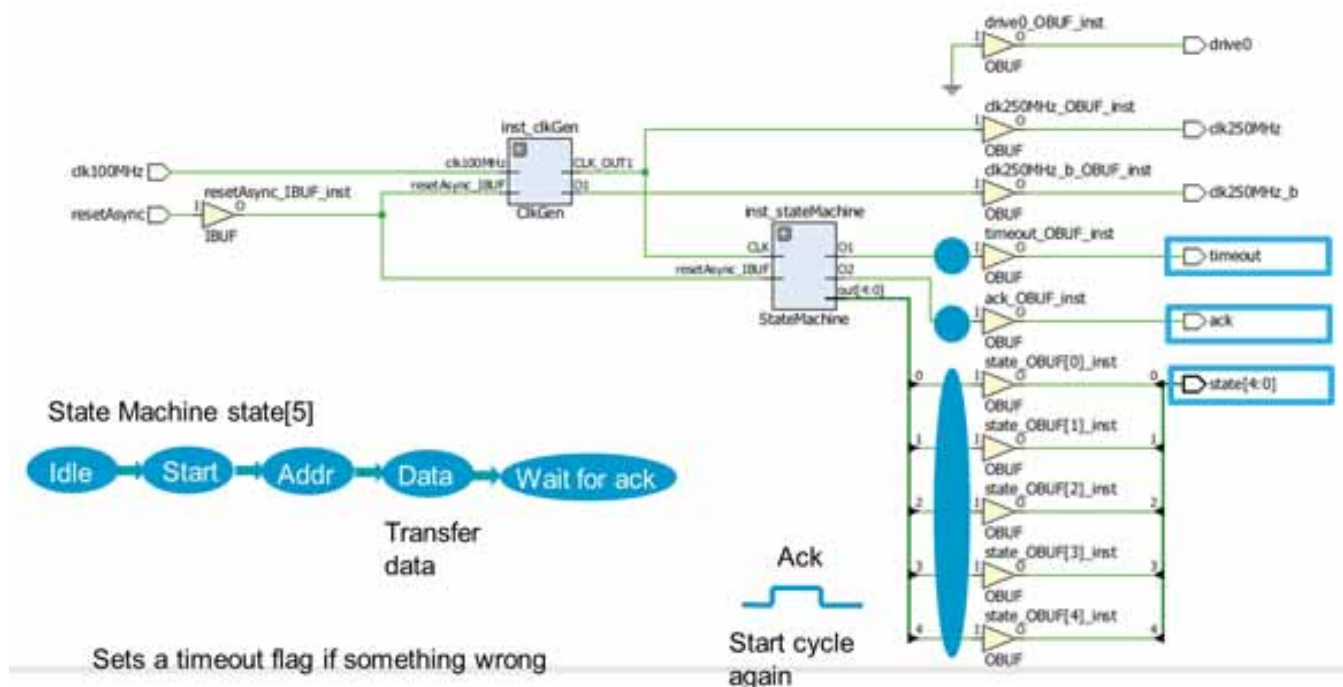


Figure 1. Expected data transfer state machine operation and schematic for the implemented design

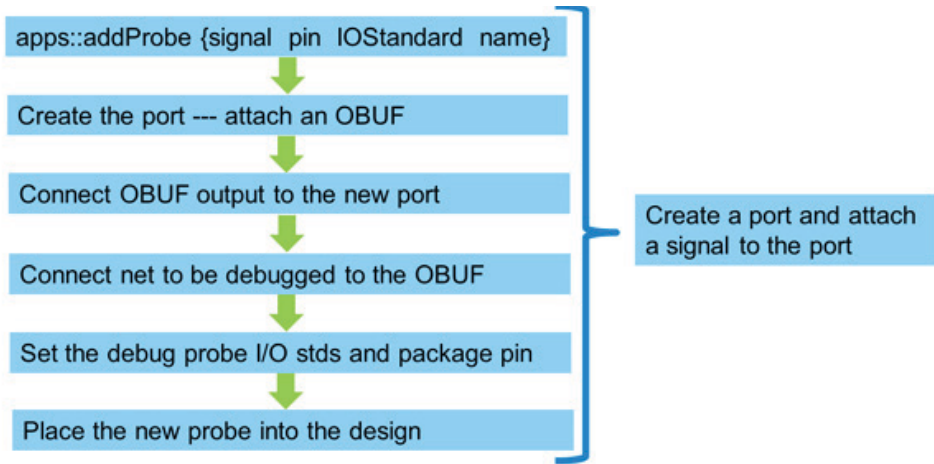


Figure 2. Tcl script syntax and steps to route out a net to a desired FPGA debug pin.

Step 1 From Xilinx Vivado, attach “Mark_debug” property to the wires of interest.

A first step to routing out the “state”, “ack” and “timeout” signals is to attach a “mark debug” property to the wires of interest. This first step ensures that nets of interest do not get optimized away during synthesis. It also ensures that the signal names are preserved through synthesis so that they can still be “found” after synthesis.

In this case, the “state[5]” five-bit bus, the “ack” one-bit signal and “timeout” one-bit signal are all of interest.

In the case of the “ack” signal, one needs to place in the source:

```

wire ack;

assign ack = ack;

(* mark_debug = "true" *) wire ack;
  
```

This must also be done for “state[5]” and “timeout”.

Step 2 Implement the design and uncheck “MARK_DEBUG” and “DON’T_TOUCH”.

The second step is to implement the design and then uncheck the “MARK_DEBUG” and “DON’T_TOUCH” attributes for each of the nets desired for probing. This is done in the “Net Properties” window in Vivado. This allows the freedom to route out signals from those nets to debug ports. Specifically, the way this is done, from the schematic view of the post-implemented design, one clicks on and highlights each net of interest, and then unchecks these two attributes.

It’s important to note that if a signal of interest is already being routed to an FPGA pin as part of the real design, as in this case, then one must select the net just prior to the

final OBUF and route that net out for debug. The name of such signals should have been preserved because of the “MARK_DEBUG” statement that was placed in the source for those signals.

For example, by clicking on the net just prior to the final OBUF before the “ack” signal output in the design, one would see in a Vivado net description field that the net name is called “ack_orig”. This is the preserved net name that will be used in the next step for signal route out.

Step 3 Run a Tcl script to route out each desired signal.

A Tcl script, provided by Xilinx, can now be run to make an incremental compile to route out each net of interest to a pin specified, with the name chosen by the designer for that routed out net. This script is located at <http://www.xilinx.com/support/answers/53266.html>. The syntax and basic steps of this script are shown in Figure 2.

Now the script can be run with these four parameters, resulting in the desired incremental compile and route out of this signal. This is repeated for the five bits of the state machine and for the acknowledge signal as follows:

Net prior to OBUF	Pin	I/O Std	Chosen output name
ack_orig	L21	LVC MOS18	ack_routeout
State0_orig	P22	LVC MOS18	state0_routeout
State1_orig	M21	LVC MOS18	state1_routeout
State2_orig	M22	LVC MOS18	state2_routeout
State3_orig	J18	LVC MOS18	state3_routeout
State4_orig	K18	LVC MOS18	state4_routeout
timeout_orig	L22	LVC MOS18	timeout_routeout

Logic Analyzer Setup

The logic analyzer is then set up for measurements. The first step is to define “labels” for the five bit state machine, the acknowledge signal, and the timeout signal, where logic analyzer channels are assigned to each of these labels. The threshold level is set to a value at the midpoint of the LVC MOS18 logic level. And sampling is chosen for “State” mode capture so that sample points are taken, synchronous with the clock of the DUT. This provides a functional capture with the very deep memory of the logic analyzer. A trigger is set for the rising edge of the timeout flag, which is suspected to be asserted since the system is failing. The logic analyzer is run, it does trigger on the timeout flag, and by right clicking on the screen, and selecting “Zoom Out

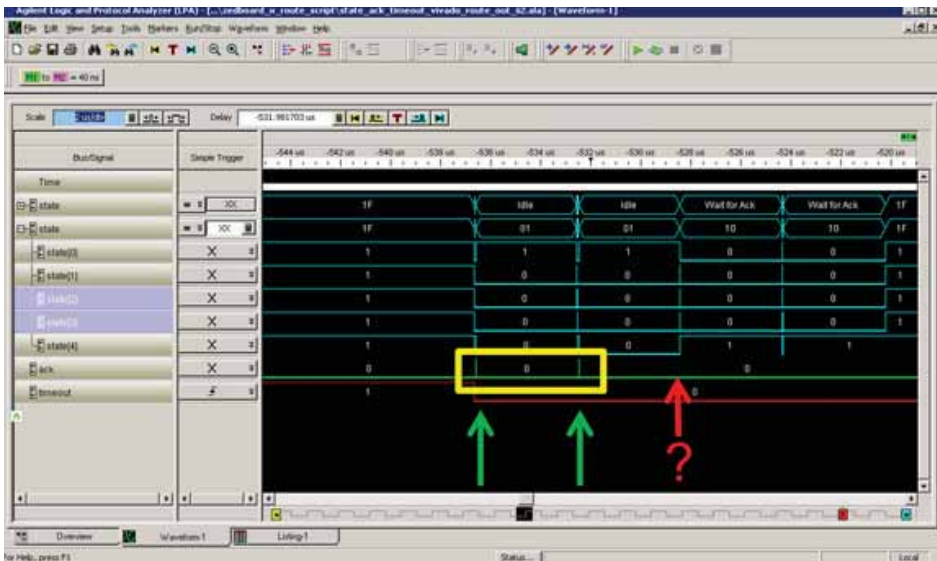


Figure 3. View of missing acknowledge after two cycles of the state machine

Full”, a flurry of activity is seen around 500 μ s prior to the timeout flag rising edge trigger point.

By zooming into this flurry of activity, it can be seen in Figure 3 that for a couple cycles of the state machine, an acknowledge bit was returned, but then on a third round no acknowledge ever comes.

By zooming in once again in the vicinity of the missing acknowledge signal, the detailed steps of the one-hot state machine can be seen in Figure 4, highlighted with a red box.

The sequence is correct for the state machine leading up to the “Wait for Ack” state.

Idle -> Start -> Addr -> Data -> Wait for Ack

The external logic analyzer, with its state capture of the routed out internal FPGA nets, validated the correct basic state machine functionality. But it hasn’t yet revealed a root cause of failure. So the next step toward finding the root cause of failure is to move the probe points to the external FPGA signals that are part of the real design, and observe them with a deep memory, high-speed timing capture.

Probing Critical, “Real” FPGA External Signals in the Design

Probing the FPGA external I/O signals that are part of the real design can be a challenge. There has to be some kind of access to the signals in order to probe them. One approach, when laying out the board, is to route the most critical signals through a probing footprint just in case there is a need to probe the signals later. It can also

be helpful to make sure there is an access to vias under the FPGA for such critical signals. Logic analyzer probes can be soldered onto these vias if necessary.

Low-cost high performance portable logic analyzers now offer 5 GHz timing capture with deep memory, so it is possible to look at signals, like those in this design, and sort out functional versus timing issues. Such timing measurements don’t replace the timing accuracy, or signal integrity view of signals provided by an oscilloscope measurement, but they do allow access to and viewing of many signals at one time. The logic analyzer set up is similar to that used before, however one major difference is that it is set to the “Timing” capture mode, and different logic analyzer channels are assigned to

the bus and signal names --- this time with the channels that are probing the real, external FPGA external signals, “ack”, “state[5]” and “timeout”. Also, with a 5 GHz timing capture, very deep memory is required to look back in time for the root cause of failure. In this case, 4M samples depth is chosen.

Through a similar process as before, and ultimately zooming in to that state machine burst where an acknowledge never comes back, a different kind of view is seen in Figure 5 where a basic timing issue between clock and state machine data is identified. The clock rising edge is not centered on the state machine data.

Alternate Trigger Technique for When There Is No “Timeout” Signal Available

Finally, it is important to know how to set up complex triggers when debugging and validating such logic designs. Here, conveniently, there was a timeout flag that could be probed, either internally or externally, and with deep

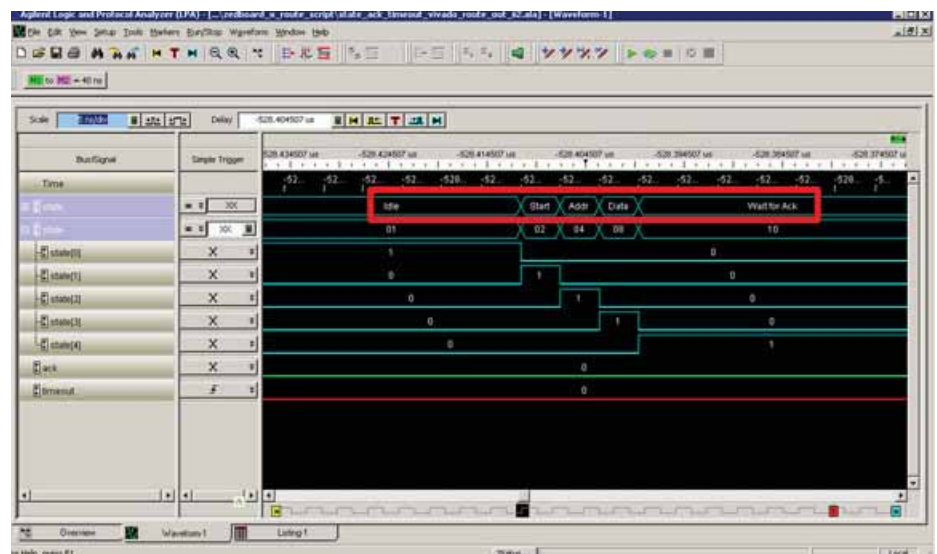


Figure 4. View of state machine functional sequence just prior to missing “ack”

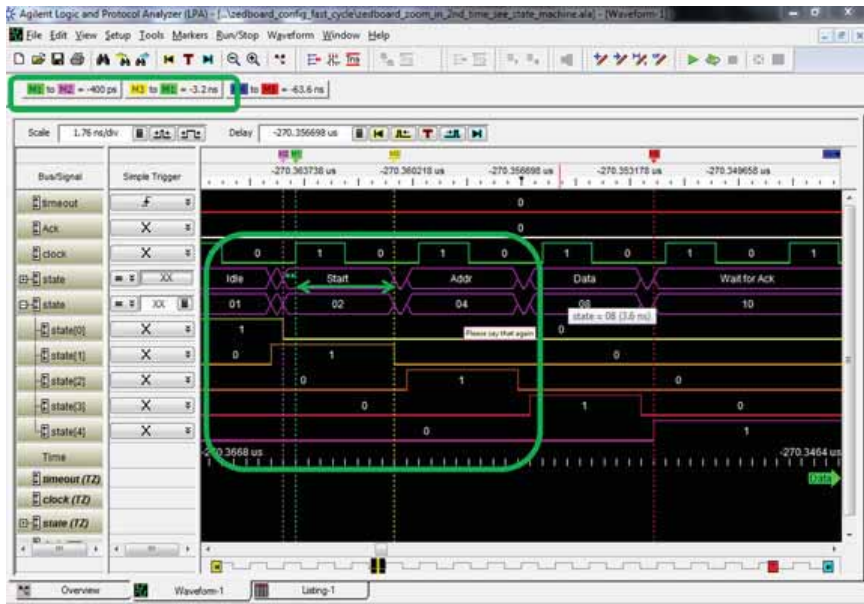


Figure 5. Identifying root cause of failure where clock rising edge not centered on state machine data

memory in the external logic analyzer, one could look far back in time to observe system operation and failure. But what about when there is no timeout signal? The answer is to create a timeout trigger, perhaps one of the most powerful debug triggers possible.

In this example, it would have been helpful to trigger the external logic analyzer if it ever saw the “Wait for Ack” state begin, but then not see the “ack” acknowledge signal returned within an expected time. Such a trigger can be set up as shown in Figure 6. The acknowledge should come within approximately 60 ns after the “Wait for Ack”, so the timeout trigger is defined to trigger the external logic analyzer if more than 100 ns passes after “Wait for Ack” without an acknowledge signal coming.

Summary

Sometimes it’s necessary to examine internal, “real” FPGA signals that are part of the system and find a way to probe those signals with an external logic analyzer. Through some basic steps, important internal nets in an FPGA can be routed out as debug signals for an external logic analyzer. State capture to view functionality of internal logic, combined with deep memory, can be very helpful to validate the design, or identify the functional problem. Timing capture, with high-speed sampling and deep memory, can also help in the search for the root cause of failure. In this case, 5 GHz timing capture, with 4M samples of depth, was able to view failing activity 500 μs prior to an external timeout flag being set. And then, the application of a user defined “timeout trigger”, defined in the logic analyzer trigger interface, can be an important tool to zero in on the root cause of a failure.

Brad Frieden is a Product Planner for the Digital Debug Solutions Product Line at Keysight Technologies. He has been with Hewlett-Packard/Agilent Technologies/Keysight Technologies for 27 years, working in fiber optic test, pulse and data generator measurements, oscilloscopes, and logic/protocol analyzer applications. Brad graduated from Texas Tech University with a BSEE and from the University of Texas at Austin with a MSEE.

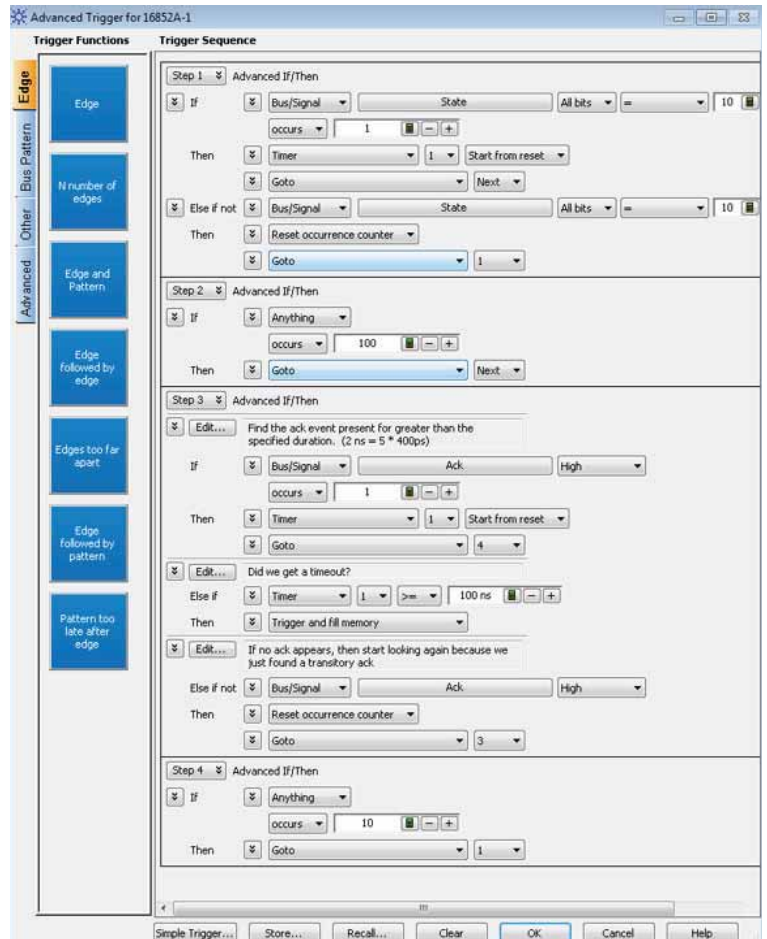


Figure 6. Set up of a “Timeout” trigger to catch a missing acknowledge signal

A Few Questions on...FPGA-based Prototyping Software Tools

Did we say “a few”? Who’s counting? We just know the topic range here spans everything from what designers ready to work on PCIe 4.0 projects should know to SoCs and design partitioning to whether FPGAs’ reputation for being difficult to program is warranted.

By Anne Fisher, Managing Editor

Editor’s note: Our thanks to Troy Scott, product marketing manager at Synopsys, who recently offered his insights on a number of questions. Scott is responsible for FPGA-based prototyping software tools at Synopsys. He has 20 years of experience in the EDA and semiconductor industries. His background includes HDL synthesis and simulation, SoC prototyping and IP evaluation and marketing.

EECatalog: What practices do you recommend for capitalizing on FPGAs’ ability to address the challenges posed by concurrent hardware/software design?



Troy Scott, Synopsys: From an ASIC design perspective FPGAs are absolutely applied to help create a more parallel design process for hardware and software. High-performance FPGA-based prototypes make it feasible to boot an OS, develop drivers

and run a software stack. And, perhaps more important, do so with a minimum number of high-capacity FPGAs to keep prototype cost as low as possible, which is a very important consideration if the prototype will be duplicated throughout the organization.

Best practices to maximize the prototype ROI typically result from collaboration between ASIC designers, the FPGA-based prototyping specialists and the software teams who use the prototype. “Over-the-wall” RTL drops are a recipe for failure and schedule delays. Design For Prototyping (DFP) RTL coding standards maximize prototype performance and speed the schedule at which it can be brought-up. At the deployment phase the most efficient prototyping teams will work closely with and treat the software team as the internal customer. Tracking software alongside RTL changes will minimize confusion as the system is integrated. Some software routines may need to account for slower operation of the prototype versus ASIC silicon. Internal probe points relevant to the software team may need to be designed in to help during the debug phase. Rapid reset cycles designed into the prototype help improve turnaround time. All of these examples require

close collaboration between the prototyping team and the embedded software development team.

EECatalog: Your thoughts on some of the challenges to FPGA-based prototyping, beginning with design partitioning?

Scott, Synopsys: To achieve both rapid time to solution and highest performance you need super-fast partition software and a platform that can be tailored to the SoC needs. One example of this is what Synopsys calls the abstract partition flow with ProtoCompiler and HAPS-70. The combination of ProtoCompiler and HAPS-70 enables prototypers to quickly create an abstracted interconnect architecture representation, generate a partition solution, then incrementally customize the partition and the hardware based on the needs of the SoC.

A smart partition automation tool allows the prototyper to create an abstract representation of the interconnect between FPGAs. At this level of abstraction, there are no fixed traces between FPGAs nor exact connections, but rather a representation of possible I/O interconnections. From this vantage point, the prototyper can very quickly see the expected FPGA utilization and secondly, and most important, the signal-to-multiplexing ratio. A prototyping rule of thumb is: the higher the mux ratio the lower the system performance, and it’s this performance that is the gating factor of overall performance in a prototype. Quick identification of bottlenecks and where to apply more physical I/O between FPGAs enables the prototyper to not only design an ideal partition scheme, but also accomplish the task quickly.

EECatalog: Long bring-up is another challenge.

Scott, Synopsys: The challenge of how to accelerate prototype bring-up remains a focus for commercial vendors of FPGA-based prototypes. FPGA logic synthesis tailored for the prototyping task, partitioning and sophisticated signal sharing schemes to maximize performance will help shorten project schedules. But even with strong

product roadmaps and innovations in prototyping EDA software, the most successful design teams have embraced Design for Prototyping (DFP) best practices throughout the ASIC development process. DFP adoption may require a culture change as profound to development teams as was the industry shift from an emphasis on ASIC design productivity to ASIC verification productivity. In the next two years those ASIC design teams that anticipate design best practices for both ASIC and FPGA targets will benefit most from FPGA-based prototyping methods.

EECatalog: What's the latest on avoiding debugging complications?

Scott, Synopsys: The reason debug of an FPGA-based prototype is complicated is due to at least two conditions prototyping specialists face. One, design modifications to some extent are required to fit into the architectural constraints of a multi-FPGA system where ASIC signal interconnect, reset and clocking, memory blocks, ASIC I/Os, DFT circuits, etc. may require careful replacement and budgeting by prototyping specialists. Functional equivalence checks either through a simulation regimen or formal methods help to confirm that these changes have not changed the logic of the initial RTL drop. Innovations in prototyping automation tools help accelerate this process with schemes to model these changes.

The second reason is that once the prototyping system is operational the very nature of an FPGA-based prototype may expose flaws that simulation and emulation will not expose during the RTL and IP verification phases. Because the prototype is running at multi-megahertz it makes software-driven test and real-world interface testing feasible. These tests are going to uncover problems, or perhaps better stated as “incompatibilities” that require driver and/or RTL changes. In 2015 debugging features tailored for prototyping systems provide high-capacity storage options to allow for long periods of evaluation with schemes to adjust instrumentation that minimizes disruption to the prototype implementation.

Success in prototyping is largely measured by how soon an operational prototype can be deployed. The faster the debug phase can be accomplished, the lower it will be on the prototyping community's list of priorities.

EECatalog: What are the top 5 things designers ready to work on the architecture of PCIe 4.0 projects should know?

Scott, Synopsys:

1. Keep your eye on the ball! Or in this case, the specification. The PCIe 4.0 specification draft 0.3 is out, draft 0.5 is expected around the end of 2014, and while most of the expected changes are electrical, there are some protocol changes too. Make sure your

FPGA vendor's SERDES is going to be able to meet PCIe 4.0 electrical requirements—many are extremely configurable, but the devil may be in the details as the spec finalizes.

2. Go big or go home! In order to keep to FPGA-friendly clock frequencies, datapaths will get very wide. That means things like 128-bit SERDES interfaces, internal datapaths, etc. Make sure your FPGAs have the capacity to handle the increased routing resources, which come with such large internal busses.
3. Feed the beast! Probably obvious, but if you're going to feed PCIe 4.0's 16GT/s data rate, you'll need more bandwidth in whatever your applications are doing.
4. Bring friends! Plan for early interoperability testing with other implementers. Even though FPGAs make logic changes “easy” it will be important to make sure your application will work with upcoming chipsets and other PCIe 4.0 devices until PCI-SIG compliance testing becomes available. Keep an eye out for early opportunities to participate in PCI-SIG “FYI” testing.
5. Never walk alone! Work with an IP vendor who is committed to closely tracking the PCIe 4.0 specification, is involved in the development of both the specification itself and the associated compliance tests and has PCIe 4.0 code available.

EECatalog: Name 5 factors that have to be involved in winning business in a case where the customer had previously designed FPGA boards in-house for an ultra-low latency application, e.g., time-sensitive financial trades.

Scott, Synopsys:

1. Shorter lead time for the prototype availability
2. Superior prototype flexibility across validation scenarios
3. Superior quality compared to low-volume system builds
4. Software tools tailored for system with a deep feature set for bring-up automation and debug
5. Support for advanced prototyping scenarios via workstation connectivity

EECatalog: As FPGAs get used more frequently as co-processors, accelerators, or offload engines, what are the design challenges of writing optimized code to take advantage of this/these capability (ies)?

Scott, Synopsys: FPGAs are becoming more powerful and power efficient over time. This is enabling more widespread use in new applications. In addition FPGAs bring to the table a very high level of processing power that can significantly speed up algorithms. In the past FPGA designers would have needed to work at optimizing their code for a specific device. However, today, synthesis tools are on the market that enable techniques developers can utilize to gain a high level of optimization for area and performance in co-processing/acceleration applications

EECatalog: Please comment on OpenCL and other high-level languages applied to FPGA designs.

Scott, Synopsys: Over the years there have been many tools developed to help make the transition from higher-level languages to the lower-level RTL, but there still remains a need for some FPGA understanding. OpenCL has been a standard in the industry for a while, and recently

both Altera and Xilinx have made announcements around OpenCL. The goal is to abstract away the traditional FPGA development flows and lower the barrier to entry for the masses. It is a difficult question to answer so broadly, but FPGAs are already being adopted more widely and in part due to the higher-level languages and abstraction they provide.

Anne Fisher is managing editor of EECatalog.com. Her experience includes being managing editor, Communications Group, at OpenSystems Media, where she had the opportunity to cover a wide range of embedded solutions in the PICMG ecosystem as well as other technologies. Anne enjoys bringing embedded designers and developers solutions to technology challenges as described by their peers as well as insight and analysis from industry leaders. She can be reached at afisher@extensionmedia.com



For the Advanced SoC Design Community



- ➔ News
- ➔ Technology Trends
- ➔ Blogs
- ➔ System Design Engineering Online Community
- ➔ Semiconductor Manufacturing and Design Community
- ➔ Videos
- ➔ Expert Insight
- ➔ Technical Papers
- ➔ RF & Microwave Systems
- ➔ Resource Guides
- ➔ Valuable Newsletters

ChipDesignMag.com

Don't Swallow the Camel and Other FPGA Wisdom

For FPGAs, higher gate counts and fabric speeds are not all that is new.

By Anne Fisher, Managing Editor

Jeff Milrod, BittWare president and CEO, does not want FPGA developers to get indigestion. To prevent this problem, he recommends the three-step process you'll read about here.

EECatalog: Design partitioning, long bring-up, debug difficulties, performance and reusability have been named as challenges to FPGA-based prototyping. Which of these do you think has the best chance of being addressed successfully in the next couple of years, and how do you define success?

Jeff Milrod, BittWare: For those who either need to use, or insist on using, HDL coding, all of those are quite



challenging and will continue to be—that is simply the nature of any low-level coding. However, great advancements have been made with respect to the tools, and these advancements will continue, albeit incrementally. Maybe the biggest thing that has helped these challenges, and will continue to help, is the fact that gate counts and fabric speeds have increased tremendously, thus requiring much less code optimization and tuning.

Previous efforts at abstracting away these challenges have been underwhelming at best and have generally only been helpful for developing code components rather than complete projects. However, there are some major paradigm shifts occurring now involving high-level coding abstractions that look like they could blow this problem away. For example, Altera's SDK for OpenCL enables high-level coding practices to implement complete complex FPGA algorithms. This does require a one-time, low-level 'board-support package' (BSP) to be built. But once that is done, the coding of the FPGA becomes essentially a software task with the associated ease-of-use and reusability.

EECatalog: Is there really any practical difference between volatile and nonvolatile FPGAs, in real practice?

Milrod, BittWare: The only conceptual difference I'm aware of would be boot time. Nonvolatile FPGAs are instant-on, which can be important for some applications. In practice, volatile FPGAs are so much bigger and faster that they are highly advantageous for all other applications.

On many of our boards, we use an instant-on nonvolatile FPGA to boot the board and the bigger volatile FPGA for all other FPGA-based tasks.

EECatalog: What's the focus of your buy versus build discussions?

Milrod, BittWare: We focus our buy vs. build discussions on low-level board support IP, SW and drivers, technical support, life cycle management, design and manufacturing quality, and technology refresh (i.e., the fact that we will continue to design similar COTS boards with future generations of FPGAs that customers will be able to easily upgrade to). The "build" option isn't simply a matter of designing and building a board (no easy task on its own). It also includes everything mentioned prior—high- and low-level software, technical difficulties, possibly manufacturing issues, as well as having to deal with part sourcing, part EOL and next-generation designs. Unless a company has the capabilities to deal with all of these tasks, the decision to buy is many times the best decision not only from a cost standpoint, but also when looking at time-to-market.

EECatalog: As FPGAs get used more frequently as co-processors, accelerators, or offload engines, what are the design challenges of writing optimized code to take advantage of this/these capability (ies)?

Milrod, BittWare: The concept of FPGA optimization is rather complicated. Unlike SW, it is not about number of cycles, nor can it be thought of as a single threaded problem. While maximum toggle rate frequency (often called Fmax) can be important for some applications, it alone doesn't necessarily indicate the FPGA performance as it neglects the inherent parallelism of FPGAs. Bus widths can be doubled, quadrupled, etc... and whole algorithmic streams can be run simultaneously, thereby resulting in far greater performance increases than improving Fmax alone. Improving performance via parallelisms often requires focusing on the FPGA's resource allocation and optimization so that buses can be widened and streams can run in parallel without overflowing the part.

However, it is often most important to focus on optimizing for human resources – time-to-market can often be more important than that last 10% optimization. Similarly, designing modular and structured function blocks might not yield the most optimal implementation, but can result in huge performance advantages down the road by allowing code reuse and reducing debug times. In fact, I now believe that for most applications the performance advantages of using FPGAs are compelling enough that optimization of the FPGA implementation is often not essential – but getting the system deployed is. To that end, we focus a great deal of our efforts towards getting our customers up and running quickly using our FPGA Developers Kit (FDK), board support packages, example projects, and pro-active support. For any given application, our function blocks and interfaces might not be completely optimal – but they work out of the box, and enable much quicker development and deployment.

Developers often “strain at gnats and swallow camels,” which can cause huge and often unnecessary delays. FPGA development and optimization is better served by first getting the basic project functioning, then identifying bottlenecks and targeting any required optimizations with a more focused approach.

EECatalog: Despite their widespread use, FPGAs are inherently difficult to program due to their low-level (RTL/HDL) languages. C-to-gates, OpenCL, and other high-level languages seek to simplify the job. Do they work? Will FPGAs ever be “easy” and become a tool of the design “masses”? Please comment on OpenCL and other high-level languages applied to FPGA designs.

Milrod, BittWare: Except for what I call the “last micron” problem, for the most part the “guts” of FPGAs have actually been fairly easy to program for quite a while. Both major FPGA vendors, and even 3rd parties, have long had pretty good tools for high-level coding of algorithms. For example, it doesn’t get much easier to implement complex processing than clicking a button to compile a Matlab/Simulink model to RTL/HDL, and that’s been widely available for many years now.

However, like the “last mile” problem of getting high-speed network backbones connected to every house, the key challenge in FPGA design is getting that algorithm developed in Matlab to connect to off-chip peripherals such as off-chip memory, network interfaces, and PCIe ports. Since those don’t exist on an FPGA, they need

to be developed. Even with those in place, it becomes an iterative problem as the peripheral interfaces need to then be integrated with the algorithm and, since its actually programming hardware rather than coding software, the timing must be “closed.” I believe that this ‘last micron’ problem—the peripheral implementation and integration—generally takes at least as long as the algorithm development, and often far longer.

Great strides have been, and are being made, to generally simplify these peripheral implementation and integration challenges with better tools such as Xilinx’s IP Integrator and Altera’s Qsys. Altera’s SDK for OpenCL takes a more extreme approach: it first requires the development of a locked down, low-level hardware peripheral implementation, called a Board Support Package (BSP), that interfaces to the compiler. The development of the BSP requires special expertise and is quite hardware centric, thus we provide several standard versions to our customers along with optional customization services. Once the BSP is implemented, the OpenCL compiler then completely abstracts the FPGA, enabling it to be coded like a processor and allowing the “masses” to have “easy” access to performance of FPGAs.

Anne Fisher is managing editor of EECatalog.com. Her experience includes being managing editor, Communications Group, at OpenSystems Media, where she had the opportunity to cover a wide range of embedded solutions in the PICMG ecosystem as well as other technologies. Anne enjoys bringing embedded designers and developers solutions to technology challenges as described by their peers as well as insight and analysis from industry leaders. She can be reached at afisher@extensionmedia.com



The FPGA Advantage for Beamforming

The argument for adaptive beamforming that relies on FPGAs to make mil aero and 4G applications possible—realizing “look ma!” power reductions in the process.

By Jason Cella, 4DSP, LLC

Beamforming can be switched or adaptive. In switched beamforming, a mobile telecommunications base station, for instance, chooses from a predefined selection of beams that each target a specific direction based on the strength of the received signal. As a user moves in relation to the array of antennas, the signal is switched to other elements in the array that are better positioned to provide a stronger signal in a particular direction. Adaptive beamforming differs from switched beamforming in that adaptive beamforming employs real-time computations that make it possible for the base station to transmit more focused beams in the direction of target users—at the same time reducing output in other directions to substantially lower interference between elements.

A Means to Avoid Overburdening DSPs

Adaptive beamforming designs call for very high processing bandwidth—billions of multiply and accumulate operations must be performed each second—a pace that makes it essential for receiving systems to suppress noise sources and interference. Meanwhile, real-time directional control of each element in the antenna array must be maintained. To accomplish this, it is necessary to digitally process the signal received by each antenna element individually and simultaneously using element-level processing. Because of the heavy computational load required, traditional CPUs and DSPs can be rapidly overburdened in adaptive beamforming applications. FPGAs, however, bring to this heavy computational task the needed features and tools, including embedded DSP blocks, parallel processing architecture, and enhanced memory capabilities.

The ever-growing global demand for mobile broadband data and voice services continually drives wireless network operators to expand and upgrade their networks in order to deliver more capacity. Operators are simultaneously trying to maximize the number of users that each wireless base station can support to lower their infrastructure costs while maintaining an attractive price point for subscribers.

Complicating this effort is the limit on the amount of available wireless spectrum. The limit means increased traffic generates more interference, and call quality suffers partly because of the limitations of antenna technology. One approach is to use omnidirectional antennas to transmit and

receive on cellular towers. However, this traditional method, in which the antennas act as transducers, converting electromagnetic energy into electrical energy, is not efficient. Another drawback to this method is that it suffers from a high degree of interference that diminishes overall connectivity due to the multiplicity of signals present at a single tower.

Brainier Antennas

Using directional sector antennas grouped together on the same tower can mitigate this interference. Telecommunications networks are increasingly using these adaptive array antennas, or smart antennas, to improve wireless connection quality and boost overall capacity. This is achieved with beamforming techniques that direct beams from the base station at individual users through the use of advanced digital signal processing. Beamforming adjusts the power and phase of every incoming and outgoing signal to create a beam that travels in a specific direction while decreasing non-essential output. This reduces the amount of



Figure 1. One example of an FPGA card that designers can now opt to employ for applications that benefit from a close integration of FPGA cards and high-performance analog transceivers is a Xilinx Virtex-7 based card from 4DSP, the FM788.

interference that individual signals inflict upon one another and increases the quality of all connections.

The emergence of innovative adaptive beamforming algorithms has led to an increase in the use of floating-point arithmetic in signal processing to minimize interference and boosted radar capacity by enabling real-time tracking of targets. This is achieved by creating multiple spot beams simultaneously using algorithms such as QR decomposition (QRD) and weight back substitution (WBS). These algorithms facilitate the adaptive formation of beams while reducing noise and interference, but they require a large number of floating point operations per second.

Because of the size, weight and power limitations of many radar systems, the use of legacy CPU or GPU options is not the best approach because of the increased amount of hardware required to perform the floating-point calculations. Demanding more memory, power and space, the use of multiple CPUs has a major impact on radar system design. Higher cost, more complex system design, and extended integration time also come into play. A CPU-based design is further constrained by a limited choice of memory and interface options.

As compared to CPU and GPU options in radar systems employing advanced digital beamforming techniques, FPGAs lower cost and complexity, while shrinking power consumption and shortening time to market. With their ability to process highly parallel floating-point operations in adaptive beamforming applications,

FPGAs can increase algorithmic performance while dramatically reducing power consumption.

Single Device Efficiency

An FPGA is also a much more efficient choice because a single device is used to receive and process large amounts of data over such I/O standards as PCIe and Serial RapidIO from signals captured by each element in an antenna array. Able to carry out higher performance processing, such a system also eliminates the need for numerous, power-hungry, multicore CPU boards housed in a VPX chassis requiring more than 1000 watts of power. A streamlined, single-FPGA design also benefits from external memory and other additional functions present on a single board that draws less than 80 watts.

Although adaptive beamforming and smart antennas have figured in defense applications for a number of years, it's only recently that commercial cellular networks have begun to use this technology extensively. At the time of the emergence of low-cost FPGAs and DSPs, 3G mobile networks began capitalizing on the advantages of adaptive beamforming, a technology that is now seeing broad acceptance in growing 4G networks. The use of high-performance analog transceivers that are tightly coupled with FPGA cards such as 4DSP's Xilinx Virtex-7-based FM788 (Figure 1) opens up new opportunities for hardware and firmware designers to hone beamforming techniques for in commercial as well as defense use.

Jason Cella is a technical writer for 4DSP, LLC.

FPGA and PLD Solutions ONLINE

www.eecatalog.com/fpga



Explore...

- Top Stories and News
- White Papers
- Expert Opinions (Blogs)
- Exclusive Videos
- Valuable Articles

Sign up for the FPGA Quarterly Report email newsletter

→ www.eecatalog.com/fpga

Once Earmarked for FPGA Always Earmarked for FPGA?

The board would have used an FPGA approach to speed processing and I/O. Then it became possible to leverage Intel® Next Generation Network and IA advances to meet IT, cloud and call center demands.

By Todd Sirois, WIN Enterprises

Once marked to rely on FPGAs and ASICs, high-performance computing projects have another option. OEMs are turning to Intel® Architecture (IA) to save time with easier implementation in a less complex development environment.

However, in areas where real-time response is a critical requirement, FPGAs are a good choice. And that is also true for cases where, no matter the size of the datastream, monster I/O is required. Certain networking, financial or military/aerospace applications for which near instantaneous timing is critical need FPGAs. Moreover, specific I/O requirements might need the fine-tuning inherently available in FPGAs. On the other hand, for applications that do not have a critical need for real-time data disposition, IA can offer a less burdensome development approach.

FPGA Design Considerations

Due to both out-of-pocket and opportunity costs, it can make business sense to avoid FPGAs. First, the expertise required to successfully develop a relatively complex FPGA is expensive, as it represents a highly specialized skill set. Second, longer development times mean slower time-to-market resulting in lost revenue or opportunity costs. Serious development projects are generally done with high-end FPGAs, so there is that element of up-front expense as well. In addition, development boards and

software module licenses may be required, further adding to development costs before the project even begins.

The standard languages of FPGA, such as HDL, VHDL, and Verilog, are essentially hardware/firmware design languages. They are great for designing microprocessors, but hard to use and not at all intuitive. They sometimes use similar terms, but with markedly different effects. The same inconsistency is true regarding the basic architectural features of the FPGA between the major manufacturers. This is true of both the feature naming and configuring their internals. Naming and feature inconsistencies often make FPGAs hard to compare to one-to-the-other or to size for a particular project.

An FPGA's three basic features, Logic Blocks, I/O Blocks and the metallic matrix that surrounds them, are all configurable. Think of it—every gate in this “sea of gates” is configurable. This generally makes for a long and complex development process. In the absence of optimal routing and/or timing, attempts to resolve compute efficiency, space, or power management issues can prolong the development process.

Speeding New Product Development

Both FPGA and IA only cover the processor part of the new product equation. Both processor types must be designed into an embedded computing device (board or boards) and chassis of some sort. Concurrent with the processor design and embedded computing development task is software development for the FPGA.

However, OEMs working in IA get to truncate the process because their chosen processor is already designed, with power, heat, I/O, and performance factors both known and well documented. A pool of resources, the Intel ecosystem of users and designers, helps keep IA mature, stable,



Figure 1. An FPGA approach essentially adds a separate fork of development cycle for the FPGA circuit (which can be viewed as a product in and of itself).

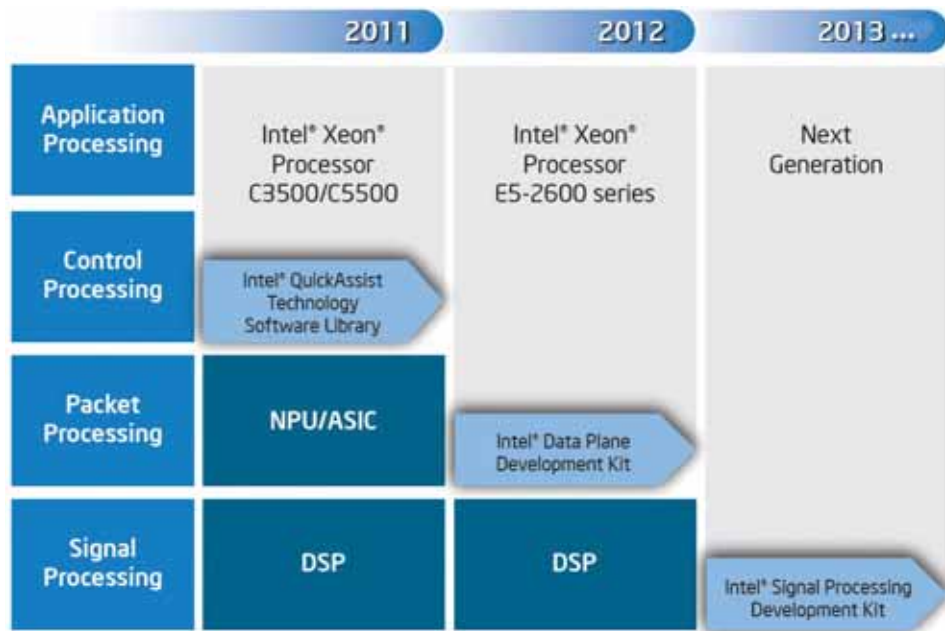


Figure 2. Intel's Next Generation Networking, as expressed by IA, enables the integration of the various IT workload types.

documented and—most importantly—debugged. Leveraging the GbE of an Intel Atom C2000 can take place by downloading a driver, freely available, without having to build it from gate and block arrays.

Figure 1 shows the steps for an IA approach and for an FPGA approach for a typical hardware design cycle. With IA, much of the specification, design and testing is done by Intel during CPU/SoC design and manufacturing. Integrators then select the chip based on clearly defined performance, I/O and power parameters. Further testing of the final product can be as simple as using industry standard, or even open source testing tools that are already designed for use in IA.

Other factors helping to shorten development time when opting for IA are the supporting software tools available from the Intel Internet of Things (IoT) Alliance ecosystem. In addition, the object oriented programming languages used, such as VisualBasic, C, C++, and php, are far more common and have a huge pool of users already available.

Consider too the ever increasing performance of Intel processors and other developments, such as the Intel® Data Plane Development Kit (DPDK*) and integrated QuickAssist acceleration technology.

Intel DPDK is open-source software for developers. It consists of a set of data plane libraries and network interface

controller drivers to enable fast packet processing on IA platforms. Its scalable programming framework for processors spans the Intel® Atom™ to Xeon® processors. Faster development and easier porting of existing high-speed data packet networking applications is possible and uses existing Intel hardware.

Software Enhancements Raise Performance Levels

Certain networking functions, especially those that are security related, are major performance sinkholes in today's networks. Integrated into the new IA SoCs like the Atom C2000 family and DH8900 series PCH, Intel QuickAssist speeds up computational operations, such as cryptography, data compression and pattern matching.

Figure 2 shows one approach to accomplishing the seamless convergence of communications and computer networking. Dedicated tools like the DPDK and QuickAssist are major components in the strategy. Additionally, Intel has the System Design Studio** which is a software suite designed to analyze and better utilize additional offload functions (e.g.: AES-NI, AVX Extensions, etc.) previously requiring FPGAs or DSPs.

Application Focus

WIN's design of a preprocessing board, System on Network Interface Card (SoNIC), shown in Figure 3, is one example of using Intel Architecture for what might have otherwise been accomplished with an FPGA. The board is an advanced networking preprocessing platform featuring a 2/4 core Intel® Xeon® BGA Processor, Next Generation PCH, integrated Intel® QuickAssist Technology and dual 10GbE on a single 3/4 length PCIe Card. Subsequent single



Figure 3. The WIN SoNIC (IP-90380), an instantiation of the Next Generation Communications Platform from Intel®.

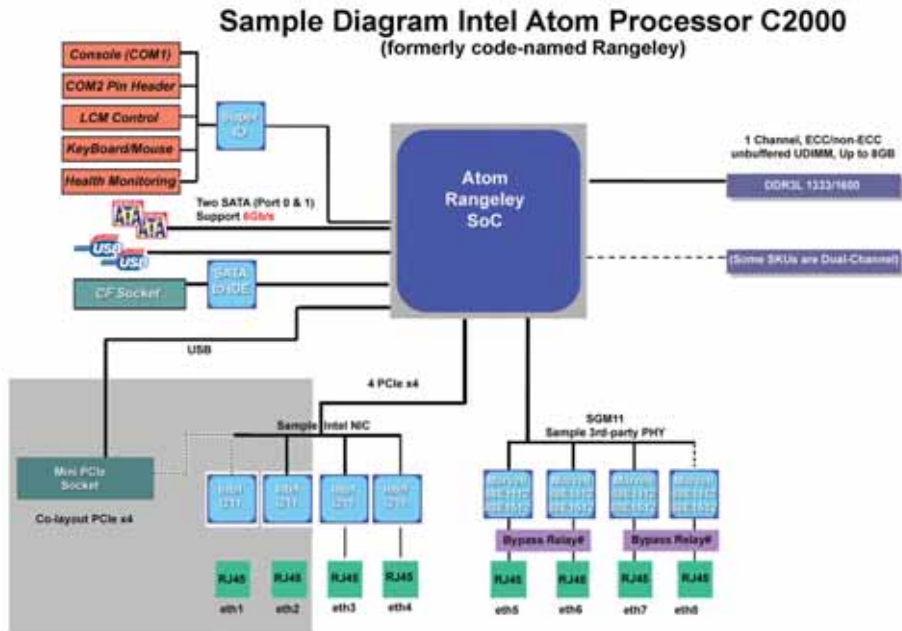


Figure 4. A sample layout using Intel NICs and 3rd-party (Marvell) Quad PHY Ethernet components.

board and blade models in this family are powered with Intel Atom C2000 processors.

Board development took place in the context of WIN support the Next Generation Network, first by developing with a low power Xeon®(Gladden) and DH8900-series PCH (Cave Creek) and then with the Intel Atom C2000 family of processors.

The board was able to support advanced networking software using pre-processing techniques to speed processing and I/O by leveraging Intel’s architectural improvements, integrations, QA and DPDK. The ultimate goal is to accelerate the various networking tasks to meet the demands of today’s IT departments, the cloud, and call centers. Software for Next Generation Networking is open source, however integration and design services are also available from other ecosystem members such as Wind River, 6WIND, Radisys and Tieto.

The reason that Intel Atom C2000 processors were chosen over a similar FPGA setup was that they had the features our customer wanted, and Intel provided proof that the performance the customer required could be met.

Because IA seemed to be the easier road than FPGA development, the Intel Atom C2000 processor approach was selected. Rather than spending development time on custom FPGA software to do the same thing, the customer optimized an operating system and bootable firmware that runs on Intel Atom C2000 processors to perform the required functions, as quickly as necessary for their solution.

When it is not necessary to develop the FPGA driving software, let alone the FPGA itself, it’s possible to focus squarely on the application software. The availability of the DPDK was an important deciding factor in selecting an IA approach for our customer’s new product development, which will span several variant solutions.

The IA nature of the project meant the OEM had the ready option to outsource the hardware aspects. By contrast, FPGA designs are done either in-house or outsourced by specialists who don’t always handle the entire design. Utilizing IA allowed both WIN and our client to limit the number of designers and fabrication plants involved, making for easier and more controlled project management. The customer’s solution can run to their needs using standard EFI firmware and open source software with relatively simple driver development from their existing pool of resources. This saved time compared to their previous FPGA design processes.

Many of WIN Enterprises’ current crop of platforms, custom boards and blades are powered by the Intel® Atom™ processor C2000 (Figure 4). These SoCs feature up to eight cores. The diagram also utilizes popular storage and I/O options.

Todd Sirois has held the position of Technical Project Manager/FAE at WIN Enterprises since 2006. This position incorporates Project Management, Sales/Sales Support and Technical Concept Development. In addition, Sirois helps manage the end-of-life (EOL) process for WIN embedded products. He has been employed in the PC hardware and software industry since 1998. Previous experience includes QA for AAA software titles at the Vivendi/Universal subsidiary Papyrus Design Group, Irrational Games (now 2K Boston).



*DPDK is an enhanced set of drivers and software libraries that are licensed under BSD and GPL where applicable (see dpdk.org).

** For more information on DPDK see: <http://www.intel.com/content/www/en/intelligent-systems/intel-technology/signal-processing-simplify-with-signal-processing-development-kit.html>, and <http://software.intel.com/en-us/intel-system-studio>

7 series FPGA Portfolio

Based on the 28 nm process node, the Xilinx 7 series is comprised of three FPGA families that address the complete range of system requirements, from cost-sensitive, high-volume applications to more demanding, performance-driven applications that need high-end bandwidth, logic capacity, and signal-processing capability.

The Artix®-7 FPGA family is optimized for lowest cost and power, with small form factor packaging for the highest volume applications, including portable ultrasound machines, digital camera control and software-defined radio.

The Kintex®-7 FPGA family is optimized for best price/performance. Kintex-7 provides a perfect balance of features and capabilities, making it ideal for wireless LTE infrastructure equipment, LED backlit and 3D digital video displays, medical imaging and avionics imaging systems.

The Virtex®-7 FPGA family is optimized for high system performance and 2x improvement in capacity over previous-generation FPGAs, making it a perfect fit for Nx100G wired applications, ASIC prototyping and emulation, and portable radar.

FEATURES & BENEFITS

- ◆ Artix-7 devices feature highest I/O, DSP, memory, and fabric performance in their class
- ◆ Kintex-7 devices deliver 2x price/performance with high DSP-to-logic ratios and 12.5G transceivers
- ◆ Virtex-7 devices feature high transceiver and signal processing bandwidth, with up to 2M logic cell capacity using 3D ICs
- ◆ Up To 50% lower power than previous generation
- ◆ Scalable, optimized architecture for rapid design migration across all families



TECHNICAL SPECS

- ◆ Up to 5.3 TeraMACs of DSP bandwidth
- ◆ Up to 2M logic cells using 3D IC technology
- ◆ Up to 96 backplane-capable 13.1 Gb/s transceivers
- ◆ Up to 1,866 Mb/s memory interface performance
- ◆ Integrated analog mixed signal

AVAILABILITY

www.xilinx.com/artix7
www.xilinx.com/kintex7
www.xilinx.com/virtex7

APPLICATION AREAS

Wired Communication, Wireless Communication, Data Center and Storage, Aerospace and Defense, Broadcast, Consumer, Industrial Automation, Medical Imaging, Automotive

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
408-559-7778 Telephone
www.xilinx.com

20 nm UltraScale Devices

The Xilinx UltraScale™ devices extend the company's market leading Virtex® and Kintex® FPGA and 3D IC families to enable next generation smarter systems with new high-performance architectural requirements.

Based on the 20 nm process node from TSMC, the UltraScale architecture addresses the industry's number one bottleneck at advanced nodes—the interconnect—to improve performance and device utilization while accelerating design closure. Core technologies include next generation routing, ASIC-like clocking, and logic infrastructure enhancements to eliminate interconnect bottlenecks while supporting consistent device utilization of more than 90% without performance degradation. Based on the ASIC-class advantage of the UltraScale architecture, these families are co-optimized with the Vivado® Design Suite and leverage the UltraFast™ design methodology to accelerate time to market.

Kintex UltraScale FPGAs deliver ASIC-class system-level performance, clock management and power management for next generation performance-per-watt-per-dollar. These devices expand the mid-range by delivering the highest throughput with lowest latency for medium-to-high volume applications that include: 100G networking, wireless infrastructure, and other DSP-intensive applications.

Virtex UltraScale is the industry's most capable high-end FPGA family, ideal for a wide range of high-performance applications requiring massive bandwidth, including: 400+ Gb/s systems, large-scale emulation and high performance computing. These devices deliver a step-function in increased bandwidth and reduced latency for systems demanding massive data flow and packet processing.

FEATURES & BENEFITS

- ◆ Next generation routing for next generation design complexities
- ◆ ASIC-like clocking for performance, scalability, and power reduction
- ◆ Enhanced logic cell packing improves performance and utilization while reducing dynamic power
- ◆ Up to 45% lower power vs. previous generation
- ◆ Seamless footprint migration between Kintex UltraScale and Virtex UltraScale devices for scalability



TECHNICAL SPECS

- ◆ Up to 4.4M logic cells using next generation 3D IC
- ◆ Up to 8.2 TeraMACs of DSP compute performance
- ◆ 30.5G transceivers for chip-to-chip and -optics and 28.2G backplanes; 16.3G backplane capable transceivers at ½ the power
- ◆ Integrated 100G Ethernet MAC and 150G Interlaken cores
- ◆ DDR4 and HMC serial memory support

AVAILABILITY

www.xilinx.com/ultrascale
www.xilinx.com/kintex-ultrascale
www.xilinx.com/virtex-ultrascale

APPLICATION AREAS

Aerospace and Defense, Automotive, Broadcast, Consumer, Data Processing and Storage, Industrial Automation, Medical Imaging, Wired Communications, Wireless Communications

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
408-559-7778 Telephone
www.xilinx.com

Xilinx Low-End Portfolio

The Xilinx Low-End Portfolio delivers optimized value to a diverse set of cost-sensitive applications and markets, over a broad range of densities.

The Spartan®-6 FPGA family is I/O optimized, and an ideal fit for simple to moderately complex glue logic designs. Because it is biased towards raw connectivity, the Spartan-6 family has best-in-class I/O and form-factor related features.

The Artix®-7 FPGA family is transceiver-optimized and is the low-end leader in nearly every dimension of performance and bandwidth. This family is the perfect fit for low-end applications needing high-end capabilities.

Low-end Zynq®-7000 AP SoC devices provide the highest level of system integration and optimization. Featuring a dual-core ARM® Cortex™-A9 processor fused with Artix-class programmable logic, the Zynq-7000 AP SoC family addresses applications that demand intelligent processing and analytics.

FEATURES & BENEFITS

- ◆ Spartan-6 devices deliver optimal I/O to logic cell ratios over a broad range of densities
- ◆ Artix-7 devices feature the highest transceiver, logic fabric, DSP, and I/O performance in their class
- ◆ Low-end Zynq-7000 devices deliver the highest system processing performance, on-chip memory, and memory bandwidth in a 28 nm based SoC
- ◆ Xilinx and their partners offer a broad portfolio of development boards for rapid prototyping and development

TECHNICAL SPECS

- ◆ Density range of ~4,000 to ~200,000 logic cells
- ◆ Up to sixteen 6.6 Gb/s transceivers
- ◆ Up to 929 GMACs of DSP bandwidth
- ◆ Up to 1,066 Mb/s DDR3 support
- ◆ 866 MHz Dual ARM® Cortex™-A9 (Low-end Zynq-7000 AP SoC devices)



AVAILABILITY

- www.xilinx.com/lowendportfolio
- www.xilinx.com/spartan6
- www.xilinx.com/artix7
- www.xilinx.com/zynq

APPLICATION AREAS

Wired Communications, Wireless Communication, Aerospace and Defense, Consumer, Industrial Automation, Medical Imaging, Automotive, Broadcast

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
408-559-7778 Telephone
www.xilinx.com

Zynq-7000 All Programmable SoC

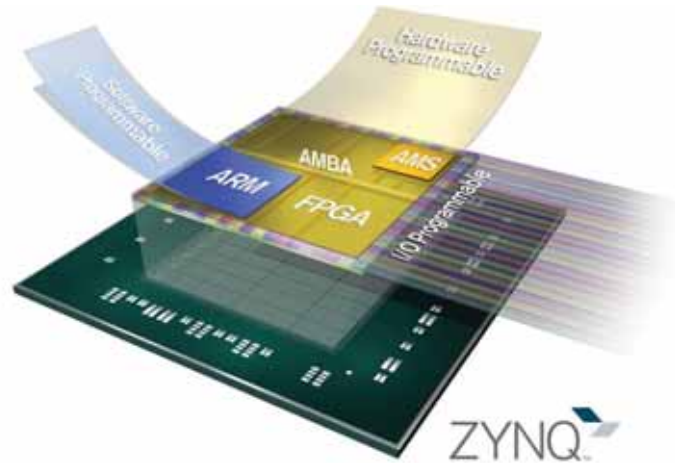
The Zynq®-7000 family is the leading All Programmable SoC with hardware, software and I/O programmability. This innovative class of product combines an industry-standard ARM® dual-core Cortex™-A9 MPCore™ processing system with Xilinx 28nm programmable logic architecture in a single device. This processor-centric architecture delivers a complete embedded processing platform that offers developers ASIC levels of performance and power consumption, the flexibility of an FPGA and the ease of programmability of a microprocessor.

The devices of the Zynq-7000 All Programmable SoC family allow designers to target cost sensitive as well as high-performance applications from a single platform using industry-standard tools. The tight integration of the processing system with programmable logic allows designers to build accelerators and peripherals to speed key functions by up to 10x. ARM architecture and ecosystem maximizes productivity and eases development for software and hardware developers.

Unlike ASICs and ASSPs, Zynq-7000 devices allow designers to modify their design throughout the development phase and after the system is in production. In addition, the Zynq-7000 All Programmable SoC family, with over 3000 interconnections between its processing system and the programmable logic, offers levels of performance that two-chip solutions (ASSP+FPGA) cannot match due to limited I/O bandwidth and limited power budgets.

FEATURES & BENEFITS

- ◆ Dual ARM® Cortex™-A9 MPCore™
 - Up to 1GHz performance
 - Enhanced with NEON Extension and Single & Double Precision Floating point unit
 - 32Kb Instruction & 32Kb Data L1 Cache
- ◆ Unified 512kB L2 Cache 256Kb on-chip Memory
- ◆ DDR3, DDR3L, DDR2 and LPDDR2 Dynamic Memory Controller
- ◆ 2x QSPI, NAND Flash and NOR Flash Memory Controller
- ◆ 2x USB2.0 (OTG), 2x GbE, 2x CAN2.0B 2x SD/SDIO, 2x UART, 2x SPI, 2x I2C, 4x 32b GPIO
- ◆ AES & SHA 256b encryption engine for secure boot and secure configuration, and RSA-2048 for asymmetric authentication of boot loaders, Programmable Logic bitstream, and runtime Software



- ◆ Dual 12bit 1Msps Analog-to-Digital converter
 - Up to 17 Differential Inputs
- ◆ Advanced Low Power 28nm Programmable Logic:
 - 28k to 444k Logic Cells (approximately 430k to 6.6M of equivalent ASIC Gates)
 - 240Kb to 3020Kb of Extensible Block RAM
 - 80 to 2020 18x25 DSP Slices (58 to 1080 GMACS peak DSP performance)
- ◆ PCI Express® Gen2x8 (in some devices)
- ◆ 154 to 454 User IOs (Multiplexed + SelectIO™)
- ◆ 4 to 16 Transceivers (up to 12.5Gb/s, depending on device)

AVAILABILITY

www.xilinx.com/zynq

APPLICATION AREAS

Automotive, Broadcast, Medical Imaging, Industrial, Aerospace & Defense, Wired and Wireless Communications, Consumer

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
408-559-7778 Telephone
www.xilinx.com

Kintex UltraScale KCU105 Evaluation Kit

The Kintex® UltraScale™ FPGA KCU105 Evaluation Kit is the perfect development environment for evaluating the cutting edge Kintex UltraScale All Programmable FPGAs. The Kintex UltraScale family delivers ASIC-class system-level performance, clock management, and power management for next generation systems at the right balance of price, performance and power. This kit provides a vehicle to evaluate cutting edge features that dramatically decrease the time required in prototyping and accelerating your design cycle. This kit is ideal for those prototyping for medium- to high-volume applications, such as Data Center, wireless infrastructure, and other DSP-intensive applications.

FEATURES & BENEFITS

- ◆ Enables evaluation of the Kintex UltraScale family that supports high capability systems at a price that helps reduce your end BOM cost
- ◆ Features rich base board that enables shorter prototyping phase by providing key features such as:
 - 64-bit DDR4 Component Memory
 - Dual SFP+ Cages for Ethernet
 - PCIe Gen3 x8
 - 2x FPGA Mezzanine Card (FMC) Interface for I/O Expansion
- ◆ Comprehensive evaluation kit boosts developer, productivity with a combination of silicon, board hardware tools, IP and reference designsh



TECHNICAL SPECS

- ◆ KCU105 evaluation board featuring the Kintex UltraScale XCKU040-2FFVA1156E FPGA
- ◆ 2x 10Gbps SFP+ modules
- ◆ 1x Fiber optic patch cable
- ◆ 1x FMC loopback card
- ◆ License voucher for a full seat of Vivado® Design Suite: Design Edition
 - Device-locked to the XCKU040
- ◆ Access to targeted reference designs for Ethernet, PCIe and design examples for testing all major interfaces on the board

AVAILABILITY

www.xilinx.com/kcu105

APPLICATION AREAS

Wireless Communications, Data Center, Video Processing

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
408-559-7778 Telephone
www.xilinx.com

Xilinx, Inc.

Zynq-7000 All Programmable SoC ZC702 & ZC706 Evaluation Kits

Xilinx and Xilinx Partners provide a comprehensive offering of embedded processing development kits. These kits feature the Xilinx Zynq®-7000 All Programmable SoC with ARM® dual-core Cortex™-A9 + 28 nm programmable logic and provide out-of-the-box design solutions that significantly cut embedded development time and enhance productivity.

The Xilinx ZC702 features a 1080p60 real-time processing video stream design example. The Xilinx ZC706 kit features include a reference design demonstrating high speed multichannel DMA interfacing to PCIe Endpoint (x4 GEN2), Video DMA (VDMA) and Sobel filtering with a HDMI based display controller and GTX serial transceivers.

The Avnet Software Defined Radio Kits combine Xilinx Zynq AP SoC with the latest generation of Analog Devices high-speed data converters and frequency-agile RF components. These kits target wireless communications from baseband to RF.

The Xilinx ZVIK kit and the Omnitek OZ745 kits are video development platforms for rapid development of video and image processing designs.

Zedboard.org offers a number of low cost Zynq AP SoC solutions including ZedBoard and MicroZed. These are a community-based Zynq-7000 SoC development kits. Digilent also offers ZyBo, a low cost Zynq kit targeted at students, academics and other who want a low cost entry point to learn about the Xilinx Zynq AP SoC products and the Vivado development tools.

FEATURES & BENEFITS

- ◆ Providing customizable, flexible SoC to meet exact project needs
- ◆ Tightly integrated processor and programmable logic and high throughput standard interconnects
- ◆ Integration of multiple system components into a single device and fewer NRE costs through programmability and standard interconnects support
- ◆ Reduced components and interconnection counts, partial reconfiguration ability and programmable processor speed



- ◆ Industry standard processor, peripherals and flexible accelerators; mix of parallel and serial I/Os supporting wide range of interface standards; ecosystem of design tools, operating systems and IP

TECHNICAL SPECS

- ◆ Xilinx ZC702 Evaluation kit for basic embedded processing development and the Xilinx ZC706 for PCIe and Transceiver based embedded development
- ◆ Avnet Zynq SDR kits for wireless communication systems development
- ◆ Xilinx ZVIK kit or the Omnitek OZ745 kit for video, imaging, broadcast or surveillance video development
- ◆ Zedboard, MicroZed or ZyBo for low cost evaluation, development and learning platforms

AVAILABILITY

www.xilinx.com/kits

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
(408) 559-7778 Telephone
www.xilinx.com

Vivado Design Suite

Programmable devices are at the heart of most systems today, enabling not only programmable logic design, but programmable systems integration. Xilinx has transformed from an FPGA company to an 'All Programmable' company, offering technology from logic and IO to SW programmable ARM® processing systems and beyond.

With the next decade of programmable platforms, comes the next generation design environment that meets the aggressive pace and the need for enhanced productivity. The Vivado® Design Suite delivers an IP and system-centric design environment built from the ground up to accelerate productivity for the next generation of all programmable devices. The Vivado Design Suite is already proven to accelerate integration and implementation by 4x over traditional design flows, reducing cost by simplifying design and automating, not dictating, a flexible design environment.

The Vivado Design Suite provides a highly integrated design environment with a completely new generation of system-to-IC level tools, all built on the backbone of a shared scalable data model and a common debug environment. It is also an open environment based on industry standards such as AMBA® AXI4 interconnect, IP-XACT IP packaging metadata, the Tool Command Language (Tcl), Synopsys® Design Constraints (SDC) and others that facilitates customized design flows.

Vivado was architected to enable the combination of all types of programmable technologies and scale up to 100M ASIC equivalent gate designs.

FEATURES & BENEFITS

- ◆ Next generation of system-to-IC level tools, built on the backbone of a shared scalable data model and a common debug environment
- ◆ 4x productivity advantage drives beyond programmable logic to programmable systems integration
- ◆ All programmable device support including 3D stacked silicon interconnect technology, ARM processing systems and Analog Mixed Signal (AMS)



TECHNICAL SPECS

- ◆ Vivado Design Suite: WebPACK Edition - The no cost, device-limited version of the Vivado Design Suite: Design Edition
- ◆ Vivado Design Suite: Design Edition - Simplified design integration and implementation. The Design Edition enables new levels of flexibility for designing, integrating, implementing and reusing modules.
- ◆ Vivado Design Suite: System Edition – The same front-to-back support as Design Edition plus Vivado High-Level Synthesis and System Generator for DSP

AVAILABILITY

Download today www.xilinx.com/download

CONTACT INFORMATION



Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
USA
408-559-7778 Telephone
www.xilinx.com

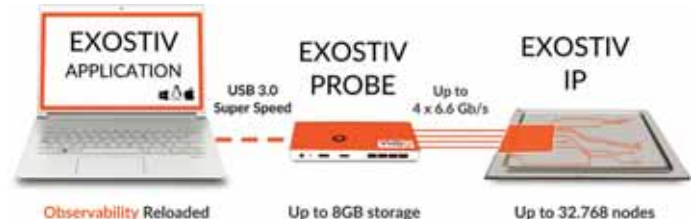
EXOSTIV™

Supported FPGA/CLPDs: Xilinx FPGA from Series 7; Altera FPGA in roadmaps

EXOSTIV™ is the first FPGA debug solution that provides Gigabyte-range observability with a minimal footprint on the target chip resources. This innovative software uses a low profile IP directly inserted into the design. It combines large hardware bandwidth and external storage capacity to reach, collect and analyze FPGA signals at the speed of operation. EXOSTIV™ Probe uses the FPGA's multi-gigabit transceivers (MGT) to flow captured data to an external memory, providing up to 8 Gigabyte of debug data storage. EXOSTIV™ Application includes MYRIAD™ waveform viewer, the industry's first waveform viewer capable of handling terabytes of digital and analog waveform data. EXOSTIV™ IP supports repeating captures of up to 32.768 internal nodes simultaneously at the FPGA's speed of operation (16 data sets x 2.048 bits). EXOSTIV™ IP provides dynamic multiplexer control to capture even more data sets without the need to recompile. Dynamic ON/OFF controls of data sets preserve the MGT's bandwidth for deeper captures.

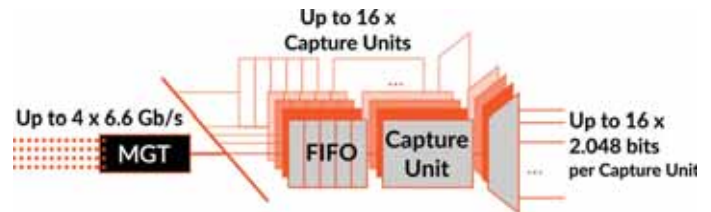
FEATURES & BENEFITS

- ◆ Dramatically increases observability: - Extended reach over the system logic. - Extended reach in time. - Signals are captured at the speed of operation.
- ◆ Preserves the FPGA I/O and memory resources.
- ◆ Seamlessly processes large debug databases.
- ◆ Includes MYRIAD™, the industry's first Terabyte-capable waveform viewer.
- ◆ Adapts to your target FPGA board with scalable connection solutions and optional adapters.



TECHNICAL SPECS

- ◆ Xilinx devices support from Series 7. (Other vendors and devices in roadmap).
- ◆ Requires Vivado software flow.
- ◆ Up to 8 GB external storage in USB 3.0 EXOSTIV probe.
- ◆ Up to 4 x 6.6 Gb/s MGT connections (12.5 Gb/S in roadmap) SFP/SFP+ or HDMI connector. SDI, PCIe, SATA, FMC support with adapters.
- ◆ Highly configurable embedded instrumentation IP with complex triggering options, dynamic data sets multiplexing and data qualification.



AVAILABILITY

Q1 2015

CONTACT INFORMATION



Yugo Systems
Avenue Moliere 18
Wavre B-1300
Belgium
sales@yugosystems.com
www.yugosystems.com

SUPERIOR **EMBEDDED** SOLUTIONS



DESIGN YOUR SOLUTION TODAY
CALL 480-837-5200

www.embeddedARM.com

NEW! TS-7250-V2 Embedded Board
High Performance and Industrial Grade



Pricing Starts At

\$165
Qty 100

\$199
Qty 1



Available with TS-ENC720 enclosure
(Shown with optional microSD card)

Features:

- Up to 1 GHz ARM CPU
- 512 MB RAM
- 8 or 17 KLut FPGA
- 2x SD Card Socket
- 2x 10/100 Ethernet
- 2x USB Host
- 1x USB Device
- 6x Serial Ports
- 75x DIO, 1x CAN
- 1x PC/104 Connector

Benefits:

- Hardware flexibility with on-board FPGA
- Several control I/O interfaces
- Launches your application in half a second
- Easy development w/ Debian and Linux 3.14
- High data reliability with 2 GB SLC eMMC flash
- -40 °C to 85 °C industrial temperature range

NEW! TS-4900 Computer Module
1 GHz i.MX6 with WiFi & Bluetooth



Pricing Starts At

\$99
Qty 100

\$134
Qty 1

- 1 GHz Single or Quad Core Cortex A9 ARM CPU
- Up to 2 GB DDR3 RAM and 4 GB eMMC Flash
- Wireless 802.11 b/g/n and Bluetooth 4.0
- 4 KLut FPGA, 1x Gigabit Ethernet, 1x PCI Express Bus
- 1x mSD slot, 1x SATA II, 1x USB Host, 1x USB OTG
- 70x DIO, 4x I2C, 1x I2S, 2x SPI, 2x CAN
- -40 °C to 85 °C Industrial Temperature Range
- Runs Linux 3.10, Debian, Ubuntu, Yocto, QT, OpenGL
- **Coming Soon:** QNX, Android and Windows Support



Available w/ TS-TPC-8390
Touch Panel Computer



Available w/ TS-8150
PC/104 Development Kit



We've never discontinued a product in 30 years



Embedded systems that are built to endure



Support every step of the way with open source vision



Unique embedded solutions add value for our customers

www.embeddedARM.com

Delivering A Generation Ahead

UltraScale — Doing More for Less at 20nm



KINTEX^U
UltraSCALE

VIRTEX^U
UltraSCALE

- High-Performance Computing
- Image and Video Processing
- Waveform Processing
- Packet Processing

www.xilinx.com

 **XILINX**
ALL PROGRAMMABLE™