Holistic FPGA Configuration

CJ Clark, Intellitech Corporation

Abstract *What's complex about FPGA configuration? You take your bitstream and load it into a FLASH memory or PROM and you're done, right? Not exactly. Encrypt your FPGA bitstream with AES and the product is protected, right? Not exactly.*
Today's products require protection not just from reverse engineering your bitstream, but also from gray market product builds, trojan bitstreams, hackers, corrupted in-the-field updates, incorrect version updates and unauthorized in-the-field updates (read 'unpaid for'). This tutorial presents current techniques used for FPGA configuration, the disadvantages and then presents a possible solution. The holistic approach looks not just at design security but authentication, update safety, ease-of-use at the CM, ease-of-use during PCB test and ease-of-use for in-the-field updates. The approach considers FPGA configuration during all stages of the product's life, the design, prototype, manufacturing, test and field service.

The complexity in designing the ecosystem needed for multi-FPGA based PCBs continues to increase however engineering time budgets continue to decrease. FPGA based PCBs need programmable DC/DC converters, watch dog timer ICs, power-on reset ICs, programmable clocks, multiple flash memories, serial eeproms and FPGA configuration devices. FPGA based products need security from hacking and cloning, not just bitstream reverse engineering. The security solution preferably is in a product that is plug and play throughout the lifecycle of the product and not just an application note. Downstream manufacturing costs and field service costs can be significantly reduced by including embedded JTAG based test, especially embedded test coupled with loading FPGA test bitstreams. Our position statement is that using an off the shelf 'board manager' can reduce the parts costs, and reduce many other downstream costs to the company that may traditionally not be considered. The SystemBIST IC described in the presentation allows user-defined FPGA configuration sequences, DC/DC voltage margining, user programmable resets, watch-dog activity, security, embedded PCB self test, built-in field updating with version control and hacker security. A unique pre-programmed at the factory on-chip memory enables SystemBIST to be a physically un-clone-able feature on a PCB.

FPGA vendors have done a great job in providing many different ways to program a RAM based FPGA. There exist a large number of application notes and ideas on how it can be done. FPGAs support parallel configuration, serial configuration, JTAG, FPGA as master, FPGA as slave, direct NOR flash, direct serial FLASH, CPLD and FLASH, CPU and FLASH, PROMs, configuration devices and compact flash. These design choices have further ramifications in that some methods will support an encrypted bitstream, some methods will not. Even differences in the vendor's pod matters. Not to pick on Altera, it's a great company, but as an example, their FPGA security key can't be programmed with their standard USB pod you may already be using.

Figure 1. Easy to clone products

In some cases logistics and cost have made adding AES security for bitstreams more challenging, especially for high volume products or products with multiple contract manufacturers in multiple countries. AES security keys are generated by the FPGA tools and delivered as plain-text SVF programming files. Cloned products and product over builds are done by the unscrupulous contract manufacturer in receipt of the files needed for manufacturing or in some cases by ex-employees of those CM's with access to the gerber board files, bitstreams and security keys. Security keys can be programmed in by a trusted third party, but this has added cost and logistics. If the security keys are battery backed up keys they must be programmed after the PCB is assembled. This is not challenging with a single PCB that you can program yourself in the lab. However for a global product with volume production, or regional manufacturing it can be logistically challenging.

Hacker Gets Linux to Run on Xbox; Lays Claim to $100,000 Prize

Authored by Mark Hefflinger on March 31, 2003 - 3:45am.

San Francisco -- A hacker has successfully been able to enable Linux software to run on an unmodified Microsoft Xbox video game console, making him eligible for a $100,000 prize offered by MP3.com founder and current Lindows CEO Michael Robertson, CNET News.com reported. A group of programmers calling itself the Xbox Linux project organized the challenge, which was met by a hacker using the name "Habibi-Xbox." The hacker discovered a bug in the popular game "007: Agent Under Fire" that allows the Linux operating system to be uploaded onto the Xbox. Microsoft has targeted companies selling "mod chips," or aftermarket devices that allow Linux, or possibly pirated games, to run on Xbox by altering the device's hardware; the contest's winner was successful in finding a way to run Linux on the Xbox without altering any hardware. http://news.com.com/2100-1043-994794.html?tag=cd_mh

add new comment | *read more*

tags: Xbox | Hacker | Linux |

Figure 1.

AES decryption in the FPGA does not always prevent hackers from programming in a non-encrypted bitstream. Battery backed key methods which prevent non-encrypted bitstreams are easy to defeat by removing the battery or shorting the battery temporarily until the battery is dead. Non-volatile key storage that allows non-encrypted bitstreams to be programmed also allows non-trusted bitstreams to be programmed in. Key storage that does not allow non encrypted bitstreams to be programmed also creates challenges during production test when FPGAs need to be configured in different ways to increase test coverage. Test engineers may not be privy to the security keys especially when they are working in a company separate from the one responsible for the design. Currently only 'high end' FPGAs support AES encrypted bitstreams, another method must be used for the other FPGAs. Application notes from Altera and Xilinx show alternative methods for protecting bitstreams from copying using an external security device from Maxim. This method helps but requires either providing the key to the contract assembler, logistics for pre-programming the device or programming through a trusted third party after assembly. After these efforts and costs are expended, it remains easy to defeat by a hacker who programs the bitstream storage with an unauthorized bitstream which intentionally doesn't interact with the security device. The hacker looking to install different bitstreams, potential look-alike bitstreams which may be Trojans or for other reasons is not interested in deciphering your bitstream but to make use of the platform you have developed for nefarious purposes. Open, remote in-the-field updates which send bitstreams over the internet or use an ad-hoc updating mechanism to re-program bitstreams into well understood commodity flash also adds obvious security holes in making your product hacker resistant.

iPhone unlocked by 17-year-old hacker

Posted on 25.08.2007 at 10:32 in Tech News by Martin

George Hotz remembers taking apart his first computer, an Apple II, when he was 4 or 5 years old. He cracked open an answering machine, remote control, vacuum cleaner and more computers. He scavenged for more products to tinker with on trash night in his neighborhood. Now the 17-year-old from Glen Rock, N.J., has reached the big leagues of hacking. He says he has "unlocked" the iPhone, finding a way to get around the device's restrictions and allow it to be used not only on AT&T's cell phone network but also on T-Mobile's network and overseas. Until now, however, the iPhone has come with a catch. Because of a revenue-sharing agreement between Apple and AT&T, the iPhone operates only on AT&T's network and requires a two-year subscription.

Figure 2.

Hackers cost the company more than just embarrassment. Some engineers may believe it is OK if the product is hacked, as one said, "We're

still making money, they have to buy the product first to hack it". However, many business models are designed such that the hardware is sold at very low margins and become the platform that a company can make higher margins with the 'consumables' that go with it. Consumables could be games, songs, printer cartridges etc. Security from modifying the software or hardware in the product has to be better. Even the best of designers working in very recognizable companies have created products that a seventeen year old can hack.

Other types of hacks may circumvent controls on revenue generating features, enabling use without paying for them. Freely available software downloadable from the internet enables average users to turn on features within a product that they didn't pay for, thus creating lost revenue for the company. Bitstreams stored in commodity memory are particularly vulnerable as the formats for programming the memory and the ability to re-program it with JTAG is not particularly challenging.


ROUTER

Hack Attack: Turn your $60 router into a $600 router

by Adam Pash

Of all the great DIY projects at this year's Maker Faire, the one project that really caught my eye involved converting a regular old $60 router into a powerful, highly configurable $600 router. The router has an interesting history, but all you really need to know is that the special sauce lies in embedding Linux in your router. I found this project especially attractive because: 1) It's easy, and 2) it's totally free.

So when I got the chance, I dove into converting my own router. After a relatively simple firmware upgrade, you can boost your wireless signal, prioritize what programs get your precious bandwidth, and do lots of other simple or potentially much more complicated things to improve your computing experience. Today I'm going to walk you through upgrading your router's firmware to the powerful open source DD-WRT

Figure 3. Lost Profits when products are hacked

Other types of hacks can provide access to security passwords within a product. The company experiences revenue loss when customers feel uncomfortable about the security of the product or customers have a negative experience with the product when the product is compromised.

Figure 4. Security is compromised

Test is also one of the costs that may not be completely understood during the design phase of a product. Companies don't ship products which are not tested. A product that is delayed shipping by a week due to a test engineering problem cost the company the same amount of money as any other problem; an extra week of company wide expenses against the prior product's profits.

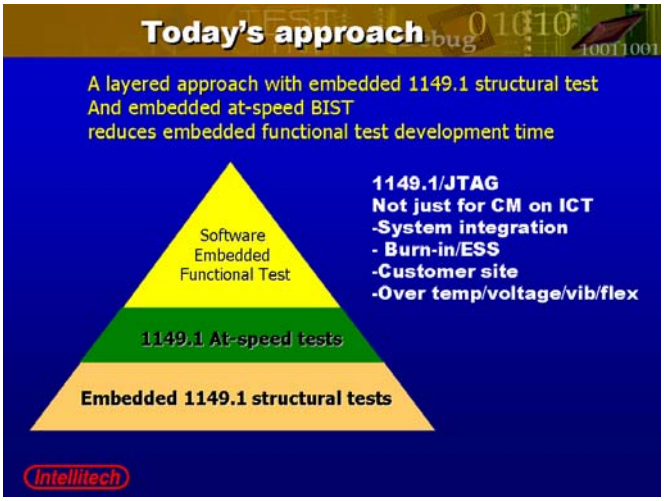

Figure 5. Reduced Functional Test Effort

Embedded functional test development using the mission mode CPU continues to grow in complexity. The mission mode function of the PCB is not well known outside of the originating company. CM engineers are not trained on developing the tests or debugging the functional test failures. It takes in-house resources to develop the tests, high level engineers which could be put towards higher value functions if a structured approach was taken. In order to standardize development and possibly use third parties for

development, it becomes necessary to separate out the mission mode firmware from the embedded test strategy. Mission mode software based test should start at a higher level, layered on top of 1149.1 structural tests and 1149.1 at-speed tests as shown in Figure 5. At-speed tests are performed by downloading test instruments into the FPGAs such as a Bit-Error-Ratio test for SERDES channels or Memory BIST for at-speed testing of DDR2/DDR3 memories. CPU based tests which are controlled by JTAG/1149.1 called 'emulation functional test' can also be added. Emulation functional test focuses on at-speed testing between the CPU and ICs based on datasheets rather than system functional operation ASICS also contain JTAG executable BIST functions. The newly emerging IJTAG (IEEE P1687) standard will proliferate more on-chip instruments accessible by JTAG.

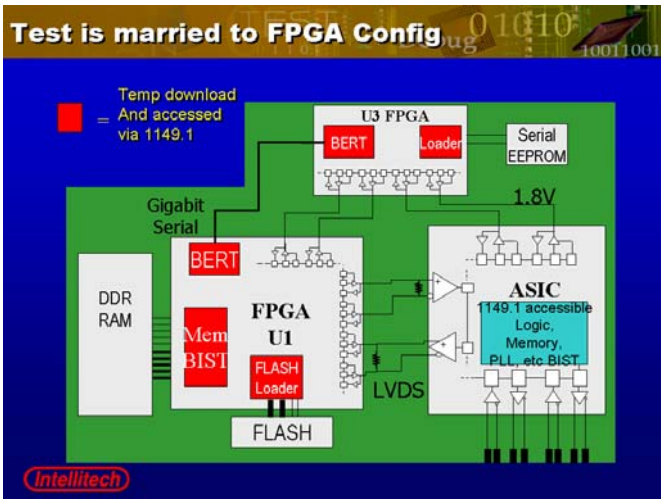

Figure 6 At-speed tests controlled by JTAG

Using the mission mode CPU to execute all possible tests creates the commonly found problem of having a single data point, not knowing whether the failure is in the software or the hardware. Separating out the embedded test and FPGA configuration infrastructure from the mission mode allows not only outsourcing of the development but also a system of checks and balances when failures in the field are encountered. A system that can store the failures in the field for later analysis eliminates the NFF, No Fault Found, enabling feedback to improve the product.

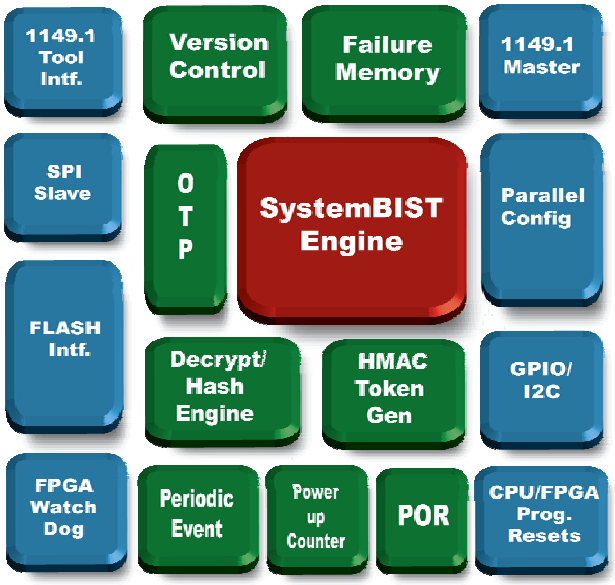

Figure 7. SystemBIST IC Block Diagram

A proposed solution is the SystemBIST IC which provides much of the on-PCB ecosystem needed for complex FPGA systems. PC based software is used to develop the ecosystem operation and strategy, then the binary representation of that operation can be downloaded to the device. SystemBIST does not contain a general purpose CPU, it doesn't have the delays or infrastructure associated with software for configuring FPGAs or running embedded test. SystemBIST provides parallel configuration of Altera and Xilinx FPGAs and JTAG based FPGA configuration. The designer has GUI access to describe how the IC configures the FPGAs and with what bitstreams to configure them. Manufacturing tests based on JTAG can easily be imported and re-used during manufacturing or in the field. When failures are found they are stored in non-volatile memory for later retrieval.

SystemBIST can operate autonomously at power-up or can accept commands and image updates over SPI from a CPU. Updates in the field are done via the SPI and the on-chip version control checks for valid update images which can contain bitstreams, updated JTAG tests or possibly new CPLD designs.

The device supports a built in power-on-reset and programmable control of the board level resets.

It's I2C and GPIO can be used for power sequencing DC/DC converters and programming adjustable DC/DC converters for voltage margining. The device supports a user programmable watch dog for FPGAs or the CPU. Rather than simply toggling reset the user can define a sequence of events to perform when the watch dog kicks such as saving FPGA registers, re-programming FPGAs, or toggling CPU resets.
SystemBIST contains a unique serial number and customer code in its one time programmable memory. Non-authorized parties cannot obtain a SystemBIST IC with a customer's code. FPGA bitstreams and JTAG operations which are in the binary image are encrypted with two 128 bit keys and tied to the customer code. Anyone with the Intellitech software tools cannot generate compatible bitstreams without being authorized to do so. This makes SystemBIST a physically unclonable device. The CPU software can check and access these variables and others over the SPI bus.
SystemBIST is compatible with AES encrypted bitstreams from Xilinx and Altera. Those methods can still be used. However, SystemBIST takes a more active role in checking for FPGA bitstream authenticity. SystemBIST passes tokens via JTAG or I2C to FPGAs that include a small design for hashing a unique response preventing non-authorized bitstreams from being present and protecting bitstreams from copy/reuse. The user can program via the software GUI the operation that occurs in the case of an incorrect response. This could be something as simple as resetting the FPGAs and re-programming or more complex.
The period engine is designed to allow for this periodic checking, it can also be programmed to perform I2C functions such as voltage margining or FPGA configuration checks. For instance, the CRC_CHECK pin of Altera devices could be periodically scanned via JTAG SAMPLE to check that an SEU has not occurred, without using mission mode CPU resources.

The IC has more benefit than just reducing the parts on the PCB, but also the advantages of the integration of normally disparate mission mode functions.

Tutorial attendees can contact the author for access to the power point slides that go with this position paper.

Bibliography:

Using the Design Security Feature in Stratix II and Stratix II GX Devices, Altera Corporation, July 2008.
http://www.altera.com/literature/an/an341.pdf

Trusted Design in FPGAs, Steve Trimberger, Xilinx, Design Automation Conference, 2007
http://videos.dac.com/44th/papers/1_2.pdf

Authentication of FPGA Bitstreams: Why and How, Saar Drimer, ARC 2007
http://www.springerlink.com/content/t71pqn4g7565w806/

A Code-less BIST Processor for Embedded Test and in-system configuration of Boards and Systems, CJ Clark, Intellitech Corp, Mike Ricchetti, ATI Research, ITC 2004,
http://www.intellitech.com/pdf/itc04sb.pdf

Design Security in Stratix III FPGAs, Altera Corporation
http://www.altera.com/products/devices/stratix-fpgas/stratix-iii/overview/architecture/st3-design-security.html

Secure Update Mechanism for Remote Update of FPGA-Based System, Benoît Badrignans1,2, Reouven Elbaz3 and Lionel Torres. SEIS 2008,
http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4569831/4577669/04577703.pdf?temp=x

Physical Unclonable Functions for Device Authentication and Secret Key Generation
G. Edward Suh, Srinivas Devadas
http://videos.dac.com/44th/papers/1_3.pdf

Xilinx® FPGA IFF Copy Protection with 1-Wire SHA-1 Secure Memories, Maxim,
http://www.maxim-ic.com/appnotes.cfm/an_pk/3826

An FPGA Design Security Solution Using a Secure Memory Device, Altera,
http://www.altera.com/literature/wp/wp-01033.pdf

Altera Configuration Handbook
http://www.altera.com/literature/lit-config.jsp

Xilinx Virtex-5 FPGA User Guide
http://www.xilinx.com/support/documentation/user_guides/ug190.pdf

Author Bio:

CJ Clark is the President and CEO of Intellitech Corporation.

His first job was with Plantronics/Wilcom in 1978 working in telecom test. He was the elected chairperson of the IEEE 1149.1 JTAG working group since 1996. He has been active in other IEEE working groups such as IEEE 1149.4, 1149.6, 1532 and 1581. He has presented at International Test Conference, TECS (Testing Embedded Cores-Based Systems) Workshop, the Board Test Workshop, Ottawa Test Workshop, Design Automation and Test Europe and VLSI Test Symposium. He is a recurring program committee member of VTS and guest lecturer on the IEEE lecture series "Mission-Critical FPGA-based Embedded Systems."

CJ serves on the University of New Hampshire College of Engineering and Physical Science (CEPS) Advisory Board. He also serves on the UNH Department of Electrical Engineering Advisory Board and is a guest lecturer. He is co-inventor on four US patents and two Canadian patents, two Taiwanese, two Indian, one European patent with others pending world-wide.

Organization

Intellitech has developed revolutionary patented technology for use by electronic product manufacturers and the semiconductor industry. Using a unique business model, Intellitech Corporation develops and licenses advanced Intellectual Property (IP) for efficient configuration, debug and test of electronic products including SoC (System-on-a-Chip), ICs, PCBs and Systems. The proprietary IP provides a scalable configuration, debug, and test infrastructure that enables customers to build high quality self-testable and in-the-field re-configurable products using a standard and consistent architecture from one product generation to the next. Intellitech's unified approach to test and configuration enables customers to provide field adaptable products, lower their manufacturing test costs, lower their field support costs, and extend their products' useful life with field upgrade-able logic.