White Paper

Anti-Tamper Capabilities in FPGA Designs

Introduction

The mission of Anti-Tamper (AT) is to deter reverse engineering of critical military technology to impede technology transfer, alteration of system capability, and prevent development of countermeasures. AT is important because governments are investing billions of dollars to develop sophisticated weapon systems and technologies. However, weapons that are vulnerable to tampering can weaken a military advantage, shorten the expected combat life of a system, and erode technological competitiveness. Opportunities for the exploitation of weapon systems are increasing due to the large export quantities of weapons and military systems, a diverse battlefield, the uncertainties of current national alliances, and loss of weapons on the battlefield. This is creating an increase in the importance of AT in military systems.

AT is required in all new military programs per the 5000-series directives from the U.S. Department of Defense (DoD). The mission of AT in electronic design is to deter (or delay) reverse engineering of critical program information (CPI), defined as "information, technologies, or systems, which, if compromised, would degrade combat effectiveness, shorten the expected combat-effective life of a system, or alter program direction." (DoD Directive 5200.39)

Components of a Complete AT Solution

There are four components to AT: tamper resistance, tamper detection, tamper response, and tamper evidence.

Tamper resistance is the ability to resist tamper attempts, and is achieved by specialized features such as the design security feature offered in Altera's Stratix® series FPGAs. The design security feature enables encryption of an FPGA's configuration bitstream using an Advanced Encryption Standard (AES) algorithm. (Note: This design security feature also is available in Stratix II, Stratix III, and Stratix IV FPGAs). In addition, Altera's partner White Electronic Designs Corporation (WEDC) provides tamper-resistant coating solutions.

Tamper detection is the ability to make the system or user aware of the tamper event. In Stratix FPGAs, programming failures caused by not using the correct AES key may indicate a tamper event. On a system level, an attempt to open up a system box or casing may indicate a tamper event.

Once tampering is detected, the system must respond by taking countermeasures. For example, Stratix FPGAs respond by not allowing programming when the key entered is incorrect. At a system level, a user can also implement "zeroization" as a response to tamper. Zeroization erases any critical technology information that is stored in the system.

Lastly, there must be evidence of tampering, so that authorized personnel inspecting the system can identify whether the system has been tampered with. In Stratix FPGAs with design security, multiple unsuccessful programming attempts may be evidence of tampering. Tamper-resistant coating solutions available from WEDC cause permanent silicon damage during physical tamper attempts on the FPGA silicon. The components of AT and Altera® solutions for each component are summarized in Table 1.

Table 1. AT Components and Altera Solutions

AT Components	Description	Altera Solution	WEDC Solution
Tamper resistance	Specialized features	Encryption key	Passivation protection
Tamper detection	Awareness	Programming failures	
Tamper response	Countermeasures	Non-programmable device	
Tamper evidence	Visible evidence	Multiple unsuccessful programming	Passivation distortion

AT Requirements by Application

System requirements vary from one application to another, so these applications need different AT solutions. For example, missiles and munitions sometimes have very limited space as well as long shelf lives. A Stratix FPGA's non-volatile AES key solution does not require a battery and therefore meets the long shelf-life requirement. The non-volatile key is also optimal for military systems where batteries are not feasible due to limitations in chemical content.

However, in secure communications applications, a key erase and reprogram capability is required, which is met by the volatile key solution. The volatile key is also optimal for remote sensors applications where key erase is required due to hostile environment exposure. These AT requirements are summarized in Table 2.

Table 2. AT Requirements by Application

Applications	System Requirements	Altera Solution
Missiles and munitions	Very limited space Need long shelf-life	Non-volatile security key No need to continuously monitor battery life
Electronics warfare	Limited space Chemical content	Non-volatile security key Limit additional chemical composition
Secure communications	Small form factor Low power Low maintenance	Volatile security key Provides Type 3 protection in Hi-Assurance NSA applications
Remote sensors	Small form factor Hostile exposure No maintenance	Volatile security key Built-in encryption for in-field protection and key erase capabilities

Altera AT Solutions

FPGAs play two different roles in a military system with anti-tamper requirements: they are devices which themselves need protecting because of bitstream configuration files with critical program information (CPI), also in addition to being flexible and important parts of board- and system-level anti-tamper systems.

Altera's AT solutions include Stratix series FPGAs, HardCopy® ASICs, Nios® II soft processors, and partner solutions from WEDC and LIT.

Stratix Series FPGAs

Stratix series FPGAs are Altera's high-end FPGAs that offer the highest densities, highest performance, and lowest power to enable system-on-a-chip solutions. The Stratix series' GX variants also offer high-speed transceivers.

Design Security Features

The Stratix III and Stratix IV design security feature enables encryption of the FPGAs configuration bitstream using a 256-bit AES, the longest key length supported by the most advanced encryption algorithm widely in use today. The key can be stored on the FPGA in either a volatile (battery-powered) or non-volatile (fuse) key storage, but not both at the same time. Table 3 summarizes differences between the two key storage options.

Table 3. Flexible Key Storage Options

Features	Volatile Key	Non-Volatile Key
Key programmability	Reprogrammable and erasable	One-time programmable
External battery	Required	Not required
Key programming method	On-board	On-board and off-board
Design protection	Secure against copying and reverse engineering	Secure against copying, reverse engineering, and tampering

While the volatile key is reprogrammable and erasable, the non-volatile key is one-time programmable. Once programmed, the non-volatile key cannot be erased or modified, thus preventing key tampering. When using the non-volatile key, the user has the option to place the FPGA into “tamper-protection” mode, which allows only encrypted bitstreams encrypted with the correct key to be configured into the FPGA successfully.

The design security feature in Stratix FPGAs protects against security concerns including copying, reverse engineering, and tampering. To protect against copying, the configuration bitstream is encrypted. Once an FPGA is configured, it is not possible to read back the unencrypted bitstream, as Stratix FPGAs do not support readback. In addition, several steps have been taken to ensure security of the key storage within the FPGA:

- The key storage is placed under layers of metal to resist physical attacks.
- Before the key is stored, it is scrambled, so the stored key is not the actual key.
- The key bits are distributed among other logic.
- The volatile key can be erased via JTAG if a tamper event is detected.

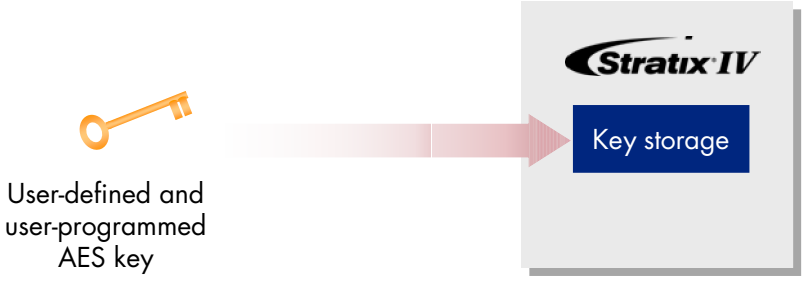
To reverse-engineer an FPGA design protected by design security, the key must first be obtained to decrypt the configuration file. However, the key is stored securely within the FPGA, which makes it extremely difficult to obtain the key. With the volatile key, the user can clear the key when a tamper event is detected. If a tampering agent somehow obtained the key and decrypted the configuration file, the next step would be to map that configuration file to the device resource level, which tells what ALMs, interconnects, memory blocks, and I/Os are used. However, Altera devices’ configuration bitstream uses a confidential format, making it extremely difficult to understand the configuration file information. In addition, the configuration file contains up to more than 250 million bits. Even if the device’s resource-level information is obtained, reverse engineering the design requires mapping to RTL (or schematic) level. This would require an in-depth physical reverse engineering of the FPGA architecture, which would be particularly difficult on an advanced process technology such as the 40-nm Stratix IV FPGAs because the necessary tools are extremely expensive.

Stratix FPGAs provide tamper protection in non-volatile key mode since it is one-time programmable. There is an additional tamper-protection bit, which, once it is set, allows the FPGA to be configured only with an encrypted file. Unencrypted files and files encrypted with wrong keys are rejected. This ensures that only those who know the key can change the FPGA functionality.

Design Security Process

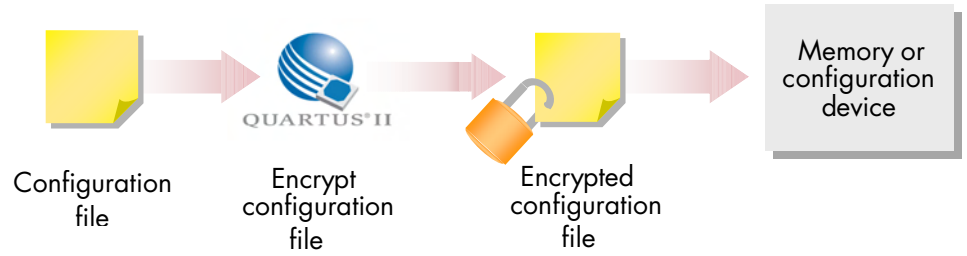
The three steps of Altera’s design security process are shown in Figure 1, Figure 2, and Figure 3. The first step (Figure 1) is for the user to choose a 256-bit key and program it into the Stratix FPGA via the JTAG interface. During the key program process, the user can choose to store the key in either the volatile or the non-volatile key storage. Note that the key is chosen and programmed by the user, and that Altera is not involved in this process.

Figure 1. Step1: Program Stratix IV Device With Key



The second step (Figure 2) is to take the configuration bitstream file (also called the POF file) and encrypt it with the same key used in Step 1, using Altera’s Quartus® design software. The Quartus software generates the encrypted POF, which user will then store in the configuration flash device—this can be an EPC or EPCS device, or an industry-standard flash device.

Figure 2. Step 2: Encrypt Configuration File and Store in Memory



In Step 3, during configuration, the encrypted POF file is downloaded into the Stratix FPGA. The AES decryptor inside the Stratix FPGA will decrypt the POF using the stored key, and configure the FPGA. If an adversary copies the encrypted POF from the flash or while it is being transferred into the FPGA, he or she cannot program another Stratix FPGA using that POF, since the AES key is unknown.

Figure 3. Step 3: Receive and Decrypt Encrypted Configuration File

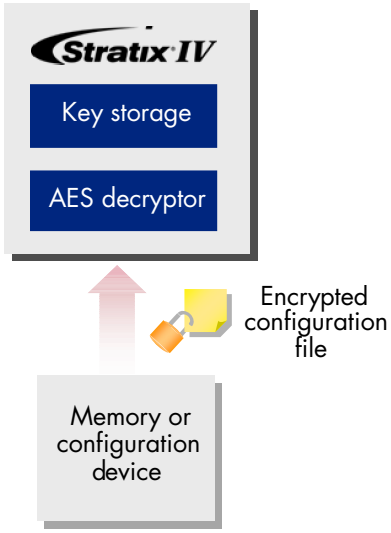
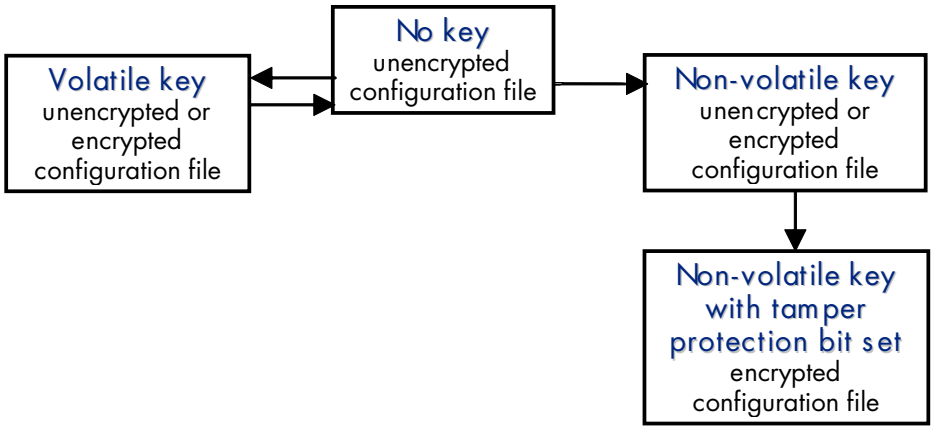


Figure 4 provides an overview of the different security modes available in Stratix III and Stratix IV FPGAs. The “no-key” mode is the default prior to programming a key. In this mode, the FPGA can be configured only using an unencrypted configuration file. Once a volatile key is programmed, the FPGA is in the “volatile key” mode, in which it can accept both encrypted and unencrypted configuration files. If the volatile key is erased, the FPGA returns to no-key mode.

When a non-volatile key is programmed, FPGA can receive both encrypted and unencrypted files. This mode is useful if the user wants the end customer to be able to modify the FPGA function without disclosing the AES key. However, once the FPGA is in non-volatile key mode, it cannot revert to the no-key mode. In the “non-volatile key” mode, once the tamper protection bit is set, the Stratix FPGA will only accept an encrypted configuration file, thus providing FPGA-level tamper protection.

Figure 4. Security Modes in Stratix III and Stratix IV FPGAs



Chip Zeroization

Zeroization entails clearing sensitive contents and verifying that the clearance took place. Stratix FPGAs support chip clearance from both the I/O pins and core logic. From the I/O pins, pulsing the nCONFIG pin clears all configuration RAM cells (CRAM), D-flipflops (DFFs), and look-up tables (LUTs). From the core logic, Stratix FPGAs enable the user's design to issue a reset command, which has the same effect as pulsing nCONFIG pin. This ability is supported in the remote update configuration mode.

The user memory blocks—MLAB, M9K, and M144K—are not cleared by pulsing the nCONFIG pin. Instead, these must be cleared by a user intellectual property (IP) or a "zero POF." Prior to pulsing the nCONFIG pin (or issuing a reset command from core logic), a user IP can clear all contents of the embedded memory blocks. While the user IP consumes additional logic resources, it may be the quickest method to clear the memory blocks. Another option is to reconfigure the FPGA with a zero-POF configuration bitstream in which all memory blocks are initialized to zero. This method takes more time—anywhere from less than 100 milliseconds to a few seconds on the largest Stratix FPGAs.

When using the zero-POF bitstream, the chip clearance is verified by enabling the error detection feature. This feature internally reads back the CRAM content and continuously performs CRC calculation in the background while the user design is running on the FPGA. Once the zero-POF is fully loaded and the FPGA is in user mode, it takes at least one error detection cycle to finish. (See the error detection section of the SEU Mitigation in Stratix IV Devices chapter of the Stratix IV Handbook for details on computing how long this will take.) Once this is complete, a user IP in the zero-POF can read the signature register of the error detection block to ensure there is no cyclical redundancy check (CRC) error. No error indicates that the CRAM cells were configured properly with the configuration pattern in the zero-POF.

Key Zeroization

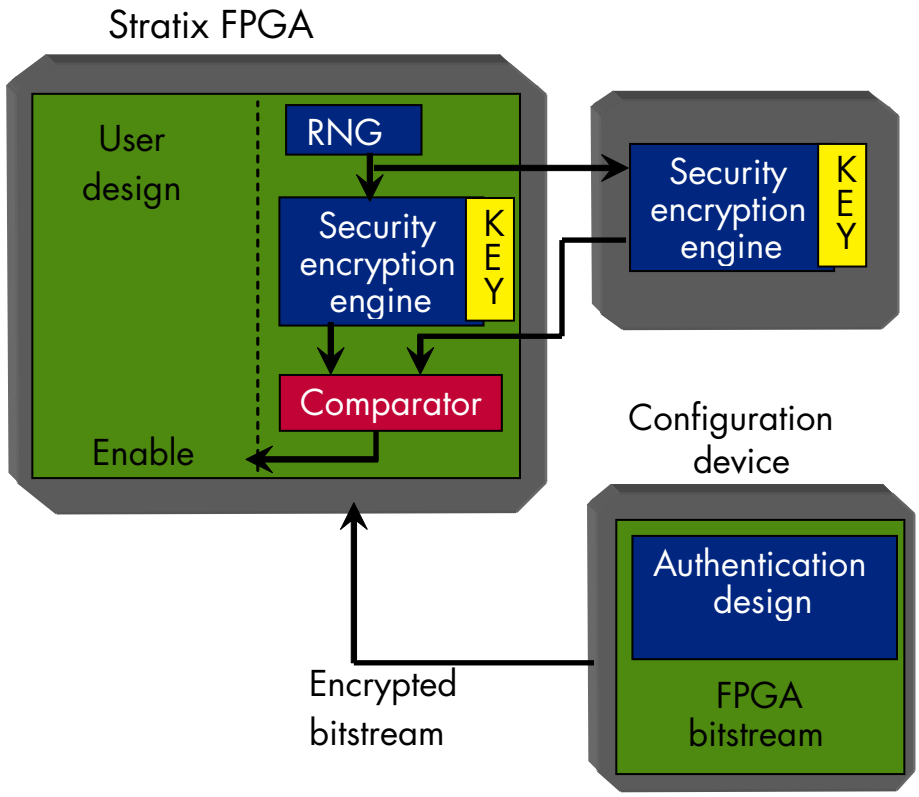
The volatile key can be cleared by issuing the "key clear" command via the JTAG interface. Verification of the clear is possible by programming a zero key (where all bits are zeros, or a zeroization pattern), and checking the verify bit. During key clear, the verify bit will be cleared and can be read back to confirm. The zero key should be programmed next. If it is programmed successfully, the verify bit will be set and can be read back to confirm.

Authentication Solution

The authentication solution originally was developed as a low-cost design security solution for Altera's Cyclone® FPGAs, which do not have a built-in design security feature. The authentication solution by itself may not be secure for military applications. However when used with the Stratix design security feature, the authentication solution can be used to add an additional layer of tamper protection, or a method of confusing a tamper agent.

The authentication solution (Figure 3) is part of the user design (soft IP) and part of the Stratix FPGA bitstream stored in the configuration flash. This FPGA design can be protected using Stratix series AES design security feature. The authentication design consists of an encryption engine with a user-selected key, a random number generator that generates the initial seed for the security engine, and a comparator that enables the user design. The FPGA and authentication device continuously exchange encrypted handshaking tokens. The comparator in the design enables the rest of the user design only if the handshaking tokens match. While the authentication device shown in Cyclone design security solution is a MAX® II CPLD, the authentication device also can be another Stratix series FPGA secured by AES design security. So if the system already has two or more Stratix FPGAs, this scheme can be implemented without adding any additional components.

Figure 5. Authentication Solution: an Added Layer of Tamper Protection



It is possible to have a network of FPGAs within the system authenticating each other, so that if an adversary tampers with one FPGA in the system, another FPGA (or all FPGAs) in the system stops functioning. Since the authentication design is a soft IP designed by the user, the complexity of the authentication design is up to the user entirely. The user can choose any encryption standard and any key length, and can make the handshaking interface as complex (or simple) as needed. The I/O pins for the handshaking interface can also be scattered among other critical data-interface I/O pins, so any tamper attempt in the system will also cause the handshaking interface to fail and the comparator to disable the FPGA design.

Configuration RAM Error Detection

Error detection feature is available in all Stratix, Cyclone, and Arria® GX FPGAs. It is an optional feature, which, if enabled, will continuously perform CRC calculation on the configuration memory inside the FPGA, and can detect any bits that may have flipped since the FPGA was configured. This feature is intended primarily for flagging configuration memory bit-flips due to soft-error (single event upsets) caused by atmospheric particle radiation. However, this feature can also act as an additional layer of AT protection, alerting the user if the configuration memory contents have changed since the FPGA was last configured.

HardCopy ASICs

HardCopy ASICs provide a seamless migration path to an ASIC device for Stratix FPGAs. Using same base arrays with customized top metal layers, HardCopy ASICs offer lower non-recurring engineering (NRE) costs compared to other ASIC solutions. Tools for migrating to HardCopy ASICs are integrated in the Quartus II design suite, thus avoiding expensive tool licensing costs required for other ASIC solutions.

HardCopy ASICs provide ASIC-equivalent security against copying, reverse engineering, and tampering, as there is no configuration bitstream, no readback, and it is not modifiable or programmable. (Making a device non-modifiable is a potential benefit for foreign military sales).

Reverse engineering a 40-nm HardCopy III or HardCopy IV ASIC is complex due to its advanced process geometry. Moreover, standard-cell ASIC reverse engineering tools, which automatically translate layer-by-layer shots to RTL, will not work with the proprietary HardCopy standard-cell architecture.

Nios II Soft Processors

The processor is often the focus of any reverse engineering effort since the processor touches many critical and sensitive functions. In an ASIC system on chip (SOC) with an embedded processor, the processor core can be identified easily by die examination, while a Nios processor design is embedded in the FPGA configuration bitstream. It is very difficult to know a processor even exists in the FPGA design. The Nios II processor also is obfuscated in HardCopy ASICs, since it exists only as HCells and interconnects, and not as an identifiable embedded processor core.

The FPGA bitstream pattern can be changed by recompiling the design using different pinouts or logic placement. This enables the user to generate nearly identical FPGA designs that have very different bitstream patterns. Therefore, even if the system has multiple FPGAs with identical designs, each FPGA can have a different bistream pattern, thus requiring a separate reverse engineering effort for each FPGA. This is true for all FPGA designs, and not just the ones with a Nios II processor.

In addition, multiple Nios II processor instances are possible within same design. Using custom instructions and hardware acceleration features, the user can make each Nios II processor instance unique. A unique processor provides added AT protection, since the documentation is not publicly available.

Custom Instructions

Custom instructions enable the user to define a unique set of instructions. The Nios II processor supports up to 256 custom instructions, which are executed by a custom logic unit added to the Nios II arithmetic logic unit (ALU). The user develops RTL for the custom logic unit and writes custom codes. Only way to reverse engineer these is by reverse engineering the RTL for the custom logic unit. Copying the custom code will not help, since the adversary will not know what the custom code means.

Hardware Acceleration

Hardware acceleration is the other method to make Nios II processors unique. The Nios II C-to-hardware acceleration (C2H) compiler automatically generates RTL for the code the user wants to accelerate. Both the custom instructions and the hardware accelerator help boost Nios II performance. With hardware acceleration, the CPU and hardware accelerator can run in parallel on separate clock domains. Thus, hardware acceleration block is not limited by the CPU clock and enables a higher performance boost compared to custom instructions.

Partner Solutions

Altera's strategy is to develop an AT solutions ecosystem by developing partner solutions that complements the chip-level solutions. Currently, partner solutions from WEDC and LIT are offered.

WEDC

WEDC offers a tamper-resistant packaging solution that involves spraying a tamper coating on the die. In the event of a physical die exam, the tamper coating causes damage to the die. WEDC also offers small form-factor packaging solutions for military applications.

LIT

LIT has developed FPGA-based AT technologies for the Office of the Secretary of Defense under two Small Business Innovative Research (SBIR) projects. LIT's AT solutions include FPGA sanitization and encryption cores.

Conclusion

Altera offers solutions that meet the AT requirements of the military market, and continues to enhance AT features in current and future products. Altera is actively working with government agencies and military contractors to further understand AT requirements, and to define future AT features. Altera is committed to providing robust AT solutions to the military market, including Stratix series FPGAs with both non-volatile and volatile design security solutions, and HardCopy ASICs, Nios soft processors, and partner solutions from WEDC and LIT to provide additional AT value.

Further Information

Stratix IV FPGAs

- *Configuration, Design, Security, and Remote System Upgrades in Stratix VI Devices* chapter of *Stratix IV Handbook*:
  www.altera.com/literature/hb/stratix-iv/stx4_siv51010.pdf
- *SEU Mitigation in Stratix IV Devices chapter* (error detection feature section) of *Stratix IV Handbook*:
  www.altera.com/literature/hb/stratix-iv/stx4_siv51011.pdf

Stratix III FPGAs

- Design Security in Stratix III FPGAs:
  www.altera.com/products/devices/stratix3/overview/architecture/st3-design-security.html
- *Design Security in Stratix III Devices*:
  www.altera.com/literature/wp/wp-01010.pdf
- *Design Security in Stratix III Devices* chapter of *Stratix III Handbook*:
  www.altera.com/literature/hb/stx3/stx3_siii51014.pdf
- *AN 512: Using the Design Security Feature in Stratix III Devices*:
  www.altera.com/literature/an/an512.pdf
- *FPGA Design Security Solution Using MAX II Devices* (authentication reference design):
  www.altera.com/literature/wp/wp_m2dsgn.pdf
- *SEU Mitigation in Stratix III Devices* chapter (error detection feature) of *Stratix III Handbook*:
  www.altera.com/literature/hb/stx3/stx3_siii51015.pdf

HardCopy IV ASICs

- HardCopy IV ASICS: Think AND, not OR:
  www.altera.com/products/devices/hardcopy-asics/hardcopy-iv/hciv-index.jsp
- ITAR Program at Altera:
  www.altera.com/end-markets/military-aerospace/itar/mil-itar.html

Nios II Embedded Processors

- Literature: Nios II Processor
  www.altera.com/literature/lit-nio2.jsp
- *Nios II Custom Instruction User Guide*:
  www.altera.com/literature/ug/ug_nios2_custom_instruction.pdf
- *Using Nios II Floating-Point Custom Instructions Tutorial*:
  www.altera.com/literature/tt/tt_floating_point_custom_instructions.pdf
- *Nios II C2H Compiler User Guide*:
  www.altera.com/literature/ug/ug_nios2_c2h_compiler.pdf
- *Accelerating Nios II Systems With the C2H Compiler Tutorial*:
  www.altera.com/literature/tt/tt_nios2_c2h_accelerating_tutorial.pdf

Acknowledgements

- Juju Joyce, Sr. Strategic Marketing Engineer, Military Business Unit, Altera Corporation
- J. Ryan Kenny, Technical Marketing Manager, Military and Aerospace Business Unit, Altera Corporation

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

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.