

REALIZATION OF A HIGH-ASSURANCE MULTIPLEXER USING FPGA-BASED SINGLE-CHIP CRYPTOGRAPHY

Duncan G. Harris, MSCS; Christopher D. Mackey, BSEE, BA Physics; Brian C. Boorman, MSEE
Harris Corporation, RF Communications Division
Rochester, NY

ABSTRACT

A High Assurance Multiplexer can be used to combine all channels of a MSLS system into a single stream that accommodates all of the channels to be transported at each end of the multiplexed link. The multiplexer/demultiplexer is termed "High Assurance" because it guarantees the integrity of the channel separation process such that, even under multiple failure conditions, the design assures that data from one channel is not inadvertently mixed with or sent to another channel.

Advances in FPGA-based Single-Chip Cryptography design flows and certification agency endorsement policies allow for a whole-new class of information assurance design mechanisms. These design flows verify that separation requirements are met and that additional security qualifications are included as part of the design implementation process.

Utilizing the benefits of this new design methodology, an FPGA-based High-Assurance Multiplexer implementation is realizable.

INTRODUCTION

Tactical communications systems are often required to be able to support independent operations on several channels simultaneously where each channel may be operating at different levels of security classifications. Maintaining the separation of these channels, from a security integrity perspective, is possible using various approaches, including a Multiple Single Levels of Security (MSLS) approach. Using a MSLS approach, all communication paths are confined to a set of closely connected independent resources. This represents a straight forward design approach since each interface is isolated both physically and electrically from each other, as required, to satisfy the security design criteria [1].

In the commercial space, others are investigating and implementing such solutions to solve the needs of the banking industry [2]. While these solutions are available,

they lack some key features required for high-assurance and Type-1 applications. These features include (but are not limited to) items such as algorithm redundancy, physical isolation, tamper detection and zeroization.

Historically, NSA has deemed the internal workings of an FPGA as "intractable". Due to the Fail Safe Design and Analysis (FSDA) requirements levied on Type-1 cryptographic equipment, the necessary physical isolation was accomplished via the use of redundant, physically separate chips. This approach came with the attendant increase in cost, size, weight and power [3, 6, 8].

In order to address these issues, the NSA and industry have since partnered to develop a design methodology and evaluation criteria that allows the implementation and certification of Type-1 and MSLS systems within a single FPGA device. This initiative is termed FPGA-based Single-Chip Cryptography.

FPGA-BASED SINGLE-CHIP CRYPTOGRAPHY

NSA has long realized that FPGAs are becoming a key enabling technology for implementation of digital cryptographic solutions. This is made even more relevant by the NSA policy that all new Type-1 Cryptographic equipment be designed to be programmable and upgradeable to support future cryptographic technology (Crypto Mod) [4].

However, industry vendors have not been able to maximize use of FPGAs due to security concerns of the various certifying organizations. This has often led to the use of multiple devices to provide redundancy, and external discrete components to provide security functions such as comparison, zeroization, and power crowbar control.

The NSA found that certain FPGAs, when used in conjunction with a security monitor, can provide a robust architecture capable of processing classified information while at the same time maintaining a very high level of security [5]. Specific design

methodologies, and a suite of tools, will allow for Multiple Independent Levels of Security (MILS) in a single programmable logic device. The toolset is used to implement the physical and logical separation of algorithms and data and to verify that separation at the end of the chip design.

NSA has performed evaluations of the Xilinx Virtex-4 (V4) family of FPGAs. As part of that evaluation, NSA determined that, with the appropriate development techniques, designs can be created on a single V4 FPGA and still pass a full FSDA evaluation. An additional tool-suite and security IP, when used correctly, can substantially increase the security of the V4 [6].

HAMUX BACKGROUND

The focus of this paper is using SCC methodology to implement a High-Assurance Multiplexer. In order to gain an in-depth understanding of the internal workings of the HAMUX, the reader is referred to the 2008 MILCOM paper titled "HIGH ASSURANCE MULTIPLEXER TECHNIQUES FOR USE WITH SECURE DIGITAL COMMUNICATIONS" [1].

The HAMUX's primary purpose is to provide a transparent communications path element with channel to channel separation by employing data integrity mechanisms whose primary purpose is verification of the channel separation. It should be noted that the HAMUX does not inherently provide for nor require data encryption and does not verify "correctness" of the data being transported.

KEY HAMUX DESIGN ELEMENTS

Given the fundamentals of a HAMUX discussed prior, this paper will now discuss the high assurance design mechanisms that are employed in this example design. These elements implement the required mechanisms indicated above and can be generalized to meet various needs of different systems. The packetized nature of the design can support many communication scenarios, including burst or continuous channel data sources. The key element of the high assurance multiplexer is the "hybrid frame". This hybrid-frame will be composed of the raw channel data and the various channel separation elements.

The HAMUX employs two fundamental elements: the channel adapter(s) and the hybrid-frame builder. The function of the channel adapter is to provide buffering

and to provide time domain separation for the individual discrete input channels. The hybrid-frame builder takes each individual channel's data from the adapter and uses it to build the composite hybrid-frame. The hybrid-frame builder will then transmit the hybrid-frame on a completely deterministic basis based on a digital clock and a set time interval. In the example presented in this paper it will be assumed that the time between transmitting hybrid-frames is a constant. The periodic frame rate needs to be chosen to meet the overall system requirements including: bandwidth required for each independent channel and what degree of channel latency is allowable.

The high assurance multiplexer must employ both redundancy (on the demultiplexer) and data channel isolation on the input and output to the high assurance multiplexer. The requirement to provide redundancy for the receive demultiplexer section of the design is accomplished by implementing a receive demultiplexer (Figure 1) that is completely redundant and provides guard logic to ensure that both demultiplexers are in

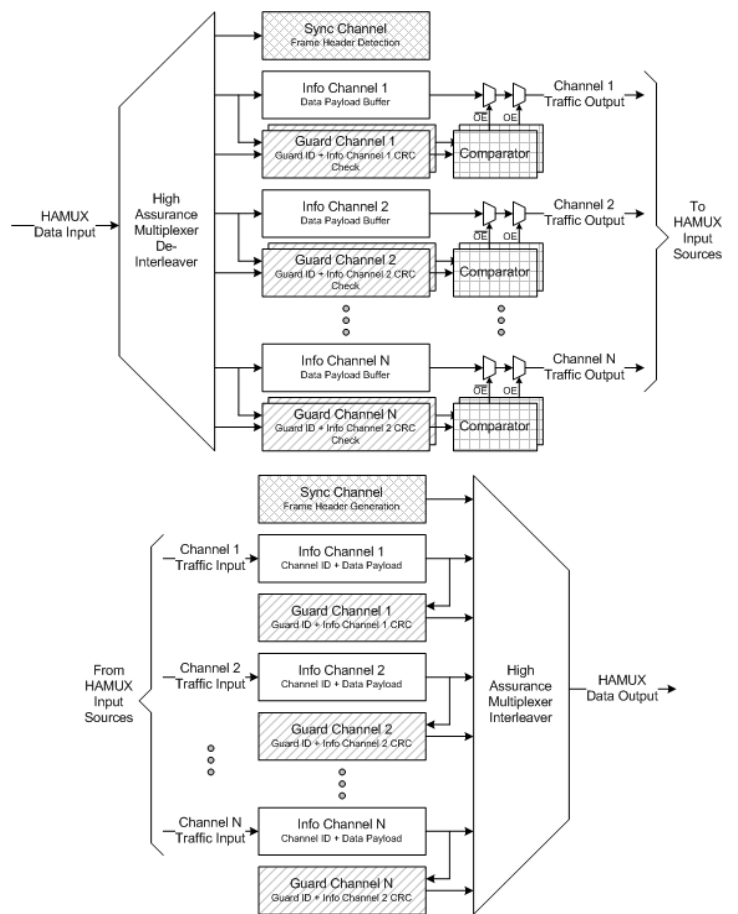


Figure 1 (U) HAMUX Design Overview

synchronization. The output of each receive section is compared on a channel by channel basis redundantly. If the result of the comparison is not identical, then the output of the channel is truncated prior to any output of this data.

FPGA-based Single-Chip Cryptography is key technology required for a FPGA implementation of the high assurance multiplexer. Given the security requirements levied by NSA on High Assurance designs must supply isolation of data streams. This isolation must be provided even if adjacent channels are at the same security level. Channel to channel isolation is paramount in a high assurance design. The typical means of providing this isolation is done by using isolation in the form of distance. In a MSLS design this isolation or distance can be provided by using completely separate signal paths and semiconductors. This approach requires more PWB space and impacts layout and power consumption. In an ASIC implementation isolation/distance can be achieved by using IC features to in essence create guard zones made up of unused gates or physical separation within the silicon to provide this spatial isolation. An ASIC implementation although effective is often a very expensive and requires significant schedule time. In designs that are not mature and changing an ASIC implementation may not be feasible.

REQUIREMENTS FOR HIGH-ASSURANCE

A high assurance multiplexer design, in general, must provide for the following characteristics:

- mechanisms to enforce channel to channel separation at all times
- “spatial” separation of channels
- the detection of any failures which compromise channel separation
- the means to verify all required mechanisms are operative.
- maintain the channel separation under conditions of multiple failures consistent with Fail-Safe Design Analysis (FSDA) requirements

A high assurance multiplexer’s primary purpose is to provide channel to channel separation. Although it will often employ data integrity mechanisms, these mechanisms are provided for the primary purpose of verifying channel separation. A high assurance multiplexer does not provide for nor require data encryption and should not be confused with data

encryption concepts or schemes. The channel data transported by the high assurance multiplexer could be separately encrypted if the user desired. The output of the high assurance multiplexer itself could also be bulk encrypted to provide efficient encryption in applications where a more cost effective means of encrypting multiple channels is desired. The high assurance multiplexer does not verify “correctness” of the data being transported. The overall purpose of the high assurance multiplexer is to provide a transparent communications path element that actively maintains channel to channel separation.

The SCC design flow helps designs meet the NSA high-assurance requirements. Routing restrictions imposed as part of the SCC design flow allow for meeting the critical High Assurance criteria of separation and isolation. The SCC design flow enforces spatial separation of identified design elements. This spatial separation enforced by the SCC design flow allows designs to meet isolation and FSDA requirements dealing with multiple failures. FPGA design techniques in general are very effective and flexible in meeting any redundancy requirements imposed by High Assurance requirements.

IMPLEMENTATION

Actual implementation of a target design using the SCC methodology requires the adherence to a stringent set of design rules during all phases of the design process. The design phases that are affected include: FPGA pin assignment; PWB layout; design partitioning; logic resource estimation; implementation; and verification. NSA guidelines [3, 6] and Xilinx application notes provide start-to-finish guidance on implementation design flows. [7]. Application notes from Altera are also in place to allow this methodology to be applied when using that vendor’s parts [8].

In this research, an SCC implementation of the HAMUX design was undertaken using the Xilinx Virtex-4 series of FPGAs. Specifically, the XC4VSX35 device using the ML402 development kit was utilized. SCC tool-flow was based on the Xilinx ISE 9.1 version of tools with the partial-reconfiguration (PR) overlay for SCC.

Design implementation started by decomposing the design components into separable modules based on traffic type (possible classification levels) and areas requiring redundancy. In the implementation of the SCC HAMUX, these modules included each of the different

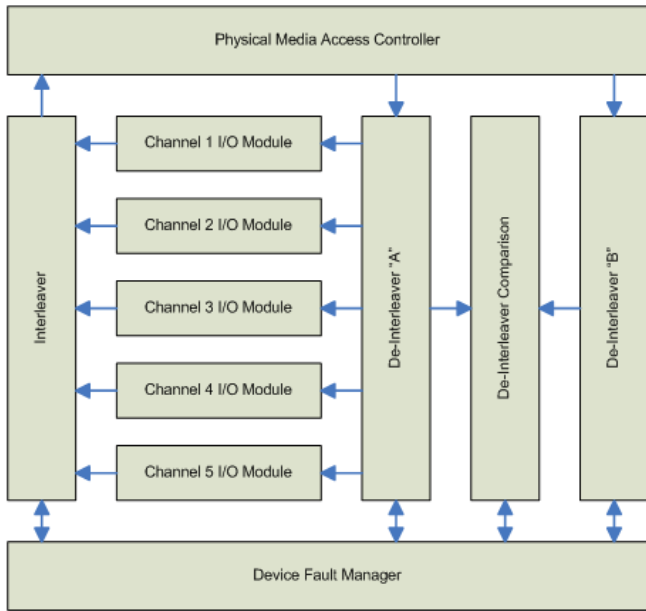


Figure 2 (U) HAMUX SCC Logical Partitioning

traffic processing interface logic blocks, the interleaver, redundant de-interleavers, and control and test logic. All of these elements were modularized, following SCC rules for low-level input/output (I/O) instantiation, module interconnects (trusted bus macros), and global routing rules.

After functional partitioning was completed, the next step was to separate the design into separately synthesizable modules. This is necessary in order to obtain the isolated netlists that are placed into the isolated fabric regions. Each module is treated as if it were its own FPGA, with a few exceptions. At each module, any I/O that would normally go on or off chip, uninhibited by a higher-level hierarchy module, need to be inferred by the synthesis tool. Any I/O that needs to communicate between modules must not have I/O buffers inferred, as these will pass through trusted bus tunnels at the next level of hierarchy.

Another important consideration in the layout of the design at the FPGA physical level is the structure of the FPGA itself and where on-chip resources are located on the die. Items such as Block RAMs, DSP blocks, I/O cells, and PLLs/DLLs are typical items used within an FPGA design, and the HAMUX uses these types of resources as well. Each of the Channel Interface modules, for instance, requires Block RAM FIFOs and I/O cells.

Figure 3 shows physical isolation at the FPGA die level. The Xilinx PlanAhead floorplanning tool is used to

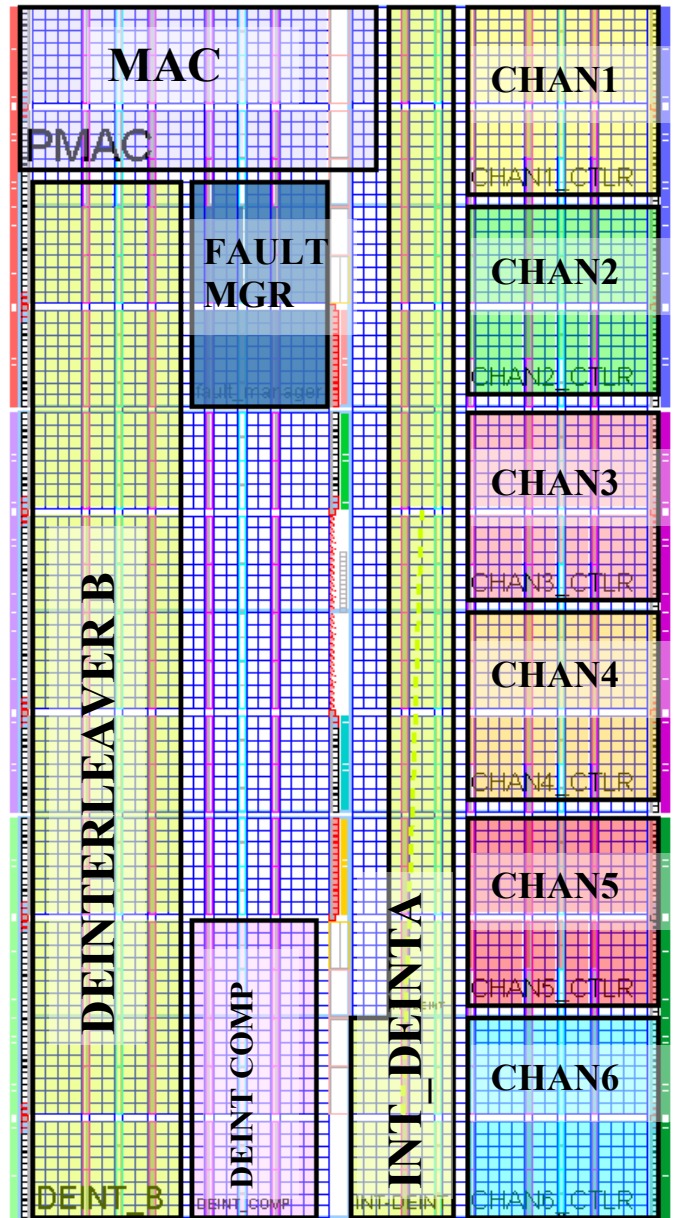


Figure 3 (U) HAMUX SCC Physical Isolation

create the different logic regions on the die, and then save these definitions into a constraints file. Note the apparent opening in the center of the chip. This area is reserved for extra security-related logic that gets added to the chip during place & route activities.

Due to the columnar architecture design of the Virtex-4 series, the I/O elements are in three columns, on the left, center, and right portions of the die. This presented challenges with the layout of the HAMUX modules and connectivity requirements of the design. Portions of the design had to be re-architected in order to route required

signals between the channel modules and the redundant de-interleavers.

Xilinx also provide an “Isolation Verification Tool” (IVT), a software tool that is used to verify that all the current FSDA security rules have been met. Specifically, these rules ensure that there is I/O banking isolation between modules, that the isolation fences are wide enough, and that module level routing is contained within the region. IVT provides an agency-approved method for verifying that the software FPGA development tools have adhered to all the agency guidelines for isolation and FSDA rules.

The base HAMUX design, when allowed “free” place and route (no location constraints) in an XC4VSX35 device, uses 20% of the slice resources and 14% of the Block RAM resources. As can be seen from the physical isolation view, when implemented in the SCC flow, this results in a large percentage of “wasted” space. This is the result of the I/O banking requirements, and the width of the isolation regions. In other words, this design required almost 100% of the resources to be allocated, with only 20% actually being used.

Each logic region is, in actuality, sparsely populated and routed. This does leave room for additional logic in each of the partitions, as long as the partitioning structure does not need to be changed. Again, these are design-dependent parameters.

CONCLUSIONS

In this paper, the authors have outlined the implementation of a high-assurance MSLS design suitable for use in Type-1 bulk cryptographic systems. Details of the HAMUX have been presented, as well as the criteria for high assurance and an SCC-based FPGA implementation.

The SCC technology that is currently available allowed for an implementation of a MILS HAMUX design in a single FPGA. During the initial partitioning process, several issues with resource locations on the die were encountered that required the functional partitioning to be adjusted. Most notable of these is the location of I/O on the V4 die in columns at the edge.

The columnar arrangement does not allow the design to have I/O at the lower modules and for those modules to have bridges to two or more other modules. In this case, the functions of the interleaver and de-interleaver “A”

were combined into one module. This was an acceptable alternative for this design that still allowed the requirements to be met. Users can expect that unless designs were specifically architected for SCC, that design changes will have to be made to use the SCC flow.

While the maturation of the vendor-provided tool flows has increased significantly in the past two years, there is still a long way to go to make this flow viable for the majority of designs in this space. Updates to the FPGA architecture and fabric resources continue to be made by the vendors in their product roadmaps to ease the adoption of SCC. In addition, there are still NSA caveats in place that require a trusted host to provide additional protection mechanisms on FPGA designs that incorporate classified algorithms. [3, 6].

It should also be noted that while the authors believe that the design approach and methods employed here would result in a certification for the handling of classified information, this work has not yet been included in a product in the current queue for certification.

Future work in this area would include porting of the design for other FPGA technologies in the SCC arena, such as upcoming Xilinx and Altera offerings. Additionally, the addition of redundant bulk encryption algorithms on the PMAC interface would provide protection of the bulk HAMUX link.

REFERENCES

1. Harris, Duncan, et. al. "High Assurance Multiplexer Techniques for use with Secure Digital Communications". 11/17/2008, [Proceedings of the Military Communications Conference \(MILCOM\)](#). 2008
2. Crowe, F. Daly, A. Kerins, T. Marnane, W. "Single-Chip FPGA Implementation of a Cryptographic Co-processor", 2004. [Proceedings of the IEEE International Conference on Field-Programmable Technology](#), 6-8 Dec. 2004 ISBN: 0-7803-8651-5
3. United States. National Security Agency. “Using the Virtex-4 Family in Secure Equipment.” [NSA I853-001D-2009](#). February 27, 2009.
4. United States. National Security Agency. “NSA/CSS Policy 3-9, Cryptographic Modernization Initiative Requirements for Type 1 Cryptographic Products”.

National Security Agency, Information Assurance Directorate. 28 Mar 2003

5. McLean, Mark and Moore, Jason. "FPGA-Based Single Chip Cryptographic Solution." Technical Session MILCOM 2006; Military Embedded Systems. March 2007. <http://www.mil-embedded.com/PDFs/NSA.Mar07.pdf>

6. United States. National Security Agency. "Using the Virtex-5 Family in Secure Equipment." NSA I853-002D-2009. February 27, 2009.

7. Moore, Jason and McNeil, Steven. "Developing Secure Designs Using Virtex-4 FPGAs". Xilinx Corporation Application Note XAPP984. 11/17/2008. <http://www.xilinx.com/member/crypto/xapp984.pdf>

8. Quintana, Paul. "Fail-Safe FPGA Design Features for High-Reliability Systems" Altera Corporation Document ESC-443. April 2009. <http://www.altera.com/literature/cp/cp-01053-fail-safe.pdf>