ARM TrustZone in an SOC Environment

Stephen Donchez, *Member, IEEE*

***Abstract*—ARM's TrustZone platform is a well-known platform for the implementation of security functionality in embedded systems. Although the platform is widely used in the larger embedded development industry, it has been less widely studied with regards to its use in Field Programmable Gate Array (FPGA) based embedded systems, which have widespread popularity. This paper analyzes current research work being done in this field, which has demonstrated that there are numerous potential vulnerabilities exposed by such a system. It then analyzes the countermeasures recommended by those conducting said research, as well as proposes avenues for future work.**

***Index Terms*—ARM TrustZone, Embedded System, System on a Chip, FPGA Security**

I. Introduction

IT has rapidly become common knowledge within the technology industry, and to some extent in society at large, that the phrase "secure IoT device" is an oxymoron. Recent advances in computer technology have led to a massive surge of smart devices, with a particularly rapid growth in the consumer electronics sector. However, this rapid proliferation of such devices has led to an unfortunate discovery – many of them fail to adequately address security concerns, leading to vulnerabilities that are often harnessed by malicious actors.

As a result, the industry has begun to take a second look at security in embedded systems. To this end, ARM ltd., the organization that oversees the development of the ARM processor family, has introduced the ARM TrustZone platform, which seeks to offer "an efficient, system-wide approach to security with hardware-enforced isolation built into the CPU." [1] This platform effectively partitions the processor into two discrete "worlds", one for secure operations and one for nonsecure (or normal) operations.

TrustZone has enjoyed tremendous success since its inception, and forms the basis for the security of many common devices, including Android based smartphones. However, the concept of Field Programmable Gate Array (FPGA) based System-on-a-Chip (SoC) devices introduces a host of complexities into the implementation of a TrustZone enabled system. The presence of in-situ reprogrammable hardware in such a system creates drastically increased potential for malicious actors to attempt to compromise the integrity of said system.

Beyond the vulnerability created by having logic that can be altered present in such a system, the FPGA development process introduces several additional concerns. First and foremost, the industry as a rule relies heavily on third party logic designs, known as intellectual property (IP), for abstracting much of the intricacies of these complex systems. The presence of this IP brings with it a host of potential security concerns, both as a result of maliciously compromised IP and also defects in otherwise genuine IP that may expose the larger system to exploitation. Furthermore, the nature of the FPGA development process is heavily automated by a complex buildchain not dissimilar to a compiler. This poses another avenue for attack - compromised buildchain software could result in a device that performs as expected but presents additional avenues for exploitation.

S. Donchez is with Villanova University, Villanova, PA 19085 USA e-mail:sdonchez@villanova.edu

Manuscript received May 6, 2020

A. Structure of the Paper

This paper is structured into 6 major sections. The introduction in Section I seeks to provide essential context as to the purpose of the ARM TrustZone platform, as well as the additional complexities introduced by using the platform in combination with a FPGA based SoC. Section II provides an explanation of the principles governing the operation of the TrustZone platform itself. Section III and IV discusses the state of current research on the effective implementation of the ARM TrustZone into FPGA-based Embedded Systems, while sections V and VI discuss potential avenues for further research. Section VII concludes the paper.

II. Details of the TrustZone platform

ARM's TrustZone technology exists to facilitate a Trusted Excecution Environment (TEE), in order to enable the execution of "sensitive" tasks in such a way that maintains their integrity without imposing massive constraints on system design. It facilitates this TEE by means of partitioning the Processing System (PS) into two distinct "worlds" - the secure world and the normal (interchangably referred to as non-secure) world. The technology then polices access to secure world resources to ensure that non-secure tasks aren't making erroneous (or malicious) accesses to secure world resources. This is acomplished by the system's "Monitor Mode", which performs context switches between worlds while also verifying access permissions and the validity of secure-world directed requests.

A. Securing the AXI Bus

TrustZone extends the concept of worlds beyond the PS and into peripherals, via an extension to the AMBA3 Advanced eXtensible Interface (AXI) bus protocol. This ensures that non-secure bus masters don't have access to secure slaves, while enabling secure masters to access both secure and non-secure slave devices. Furthermore, this is accomplished without requiring any modifications to the peripherals themselves, enabling compatability with pre-existing devices.

The principle means of enforcing isolation on the AXI (system) bus is through a pair of bits contained in the extended

bus protocol. These bits are termed the Non-Secure (or NS) bits, and one is dedicated to indicating the status of read operations, while the other is reserved for write operations. [2] In the case of a non-secure bus master, these bits are permanently hardwired high, thereby preventing access to any secure world resources. Meanwhile, secure world bus masters drive these lines using internal logic, allowing them access to both secure and nonsecure resources.

Attempts to access secure world resources by a non-secure master result in a failure mode that is implementation dependent. The sytem can either allow the request to fail silently or can raise one of two errors: a slave error or a decode error. These errors are raised on the BRESP (write) and RRESP (read) response signals, another crucial part of the TrustZone's AXI controls. [3]

B. Securing the Advanced Peripheral Bus

The AXI bus serves to connect the ARM processor with a variety of other components internal to the system, much like the PCI family of busses commonly found in laptops and desktops. However, most peripheral devices do not require the high rates of data transfer (and the correspondingly stringent constraints) of the AXI specification. Instead, these peripherals reside on the Advanced Peripheral Bus (APB), and are interfaced with the rest of the system via the AXI-APB bridge. [4]

As peripherals are often outside the scope of the designer's control, backwards compatability is an impoerant factor in system design. Therefore, the APB specifications do not allow for the implementation of an equivalent mechanism to the NS bit, as any such implementation would not be compatible with prior versions of the AMBA protocol. To mitigate this, the TrustZone architecture allocates responsibility for security enforcement of APB peripherals to the AXI-APB bridge itself.

The bridge features a set of one-bit input signals, one per peripheral, used to indicate their respective security states. These signals can either be tied statically to limit the peripheral to one world or the other, or can be driven by some other logic to allow for dynamic reassignment. One example of such logic is ARM's TrustZone Protection Controller (TZPC) [4], although developers are also free to develop their own control system.

C. Securing the AXI Switch

The nature of the AXI protocol is such that any non-trivial set of bus devices will require the implementation of a routing mechanism to facilitate traffic flow. For Xilinx based systems, this mechanism is the AXI Interconnect, a Xilinx provided piece of IP. [5] The interconnect provides a crossbar switch to enable routing between multiple masters, as well as a host of converters to handle different implementations of the protocol (data widths, clock speed, protocol versions, etc.). Altera offers a similar piece of IP, the Platform Designer (formerly Qsys) Interconnect, which offers identical capabilities as well as an interface with their Avalon streaming protocol. [6]

The TrustZone documentation does not outline specific treatment of these interconnect devices, however research explored in sections III and IV below demonstrate that substantial vulnerabilities exist in this logic. As all bus traffic in the system must flow through this IP, it is imperative that it be adequately secured.

III. Related Research in Embedded Systems

IV. Related Research in FPGA-Based Embedded Systems

V. Avenues for Future Research in TrustZone Enabled Systems

VI. Conclusion

The conclusion goes here.

Acknowledgment

The author would like to thank Dr. Xiaofang Wang, Villanova University, for her support in the authoring of this paper.

References

[1] "TrustZone," library Catalog: developer.arm.com. [Online]. Available: https://developer.arm.com/ip-products/security-ip/trustzone

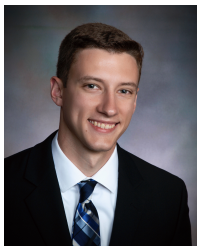
[2] E. M. Benhani, L. Bossuet, and A. Aubert, "The Security of ARM TrustZone in a FPGA-Based SoC," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1238–1248, Aug. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8643768/

[3] E. M. Benhani, C. Marchand, A. Aubert, and L. Bossuet, "On the security evaluation of the ARM TrustZone extension in a heterogeneous SoC," in *2017 30th IEEE International System-on-Chip Conference (SOCC)*. Munich: IEEE, Sep. 2017, pp. 108–113. [Online]. Available: http://ieeexplore.ieee.org/document/8226018/

[4] "ARM Security Technology Building a Secure System using TrustZone Technology," p. 108.

[5] "AXI Interconnect v2.1 LogiCORE IP Product Guide," Dec. 2017.

[6] "Intel Quartus Prime Standard Edition User Guide: Platform Designer."



Stephen Donchez (M'20) was born in Bethlehem, PA, USA in 1998. He anticipates receiving his B.S. in computer engineering from Villanova University, Villanova, PA in 2020.

In the summer of 2018 he was a software engineering intern at Harris Coroporation (now L3Harris Technologies, Inc.). He returned to L3Harris for the summer of 2019 as a systems engineering intern, and anticpates returning to the same position for the summer of 2020 at their facility in Clifton, NJ. His research interests include FPGAs, embedded software development, and embedded systems with a focus on System-on-a-Chip technologies.

Mr. Donchez is a student member of the IEEE.