FPGAs in Anwendungen mit hoher IT-Sicherheit

Halbleiter-Test & Vertriebs-GmbH, Bensheim, Deutschland, 2018

Kurzfassung

FPGAs (Field Programmable Gate Arrays) finden vermehrt Anwendung in nahezu allen Bereichen, in denen elektronische Geräte und Informationsverarbeitungssysteme zum Einsatz kommen. Aufgrund ihrer immer weiter ansteigenden Leistungsfähigkeit und Geschwindigkeit werden sie immer häufiger auch zur Beschleunigung von Berechnungen in Großrechenzentren verwendet (z. B. Amazon Web Services).
Von der einfachen Verknüpfung elektrischer Signale bis hin zur Realisierung komplexer logischer Funktionen, wie z.B. hochsicherer Datenverschlüsselung, ist ein FPGA in der Lage, Aufgaben mit sehr hoher Geschwindigkeit abzuarbeiten. Durch die Vereinigung diverser digitaler Funktionen auf nur einem FPGA-Chip, die bisher in separaten Schaltungsteilen aufgebaut waren, wird die Leistungsfähigkeit einer Schaltung drastisch erhöht und der Platzbedarf minimiert. Auch in sicherheitsrelevanten Anwendungen ist der Einsatz von FPGAs mehr und mehr im Kommen.
Nachfolgend werden die Bedrohungen vorgestellt denen elektronische Schaltungen insbesondere solche mit FPGAs ausgesetzt sind und welche Gegenmaßnahmen dazu vorhanden sind. Es wird ein Konzept vorgestellt mit dem die IT-Sicherheit von FPGAs durch isolierte Zonen weiter erhöht werden kann, wenn partielle Module erst bei Bedarf im Betrieb in isolierte Zonen eines FPGAs geladen werden.
Die dargestellten Ergebnisse und Informationen wurden in Forschungsprojekten der Fa. HTV GmbH für eine deutsche Bundesbehörde erarbeitet (vgl. [001], [002]).

1 Einleitung

Wie Abbildung 1 verdeutlicht gibt es auf elektronische Schaltungen insbesondere solchen mit FPGAs eine Vielzahl von Angriffsmöglichkeiten, die von einem Angreifer ausgenutzt werden könnten.
Die "Common Criteria for Information Technology Security Evaluation" stellen eine Grundlage dar, um Systeme mit hoher IT-Sicherheit entwerfen und sich vor möglichen Angriffen schützen zu können.

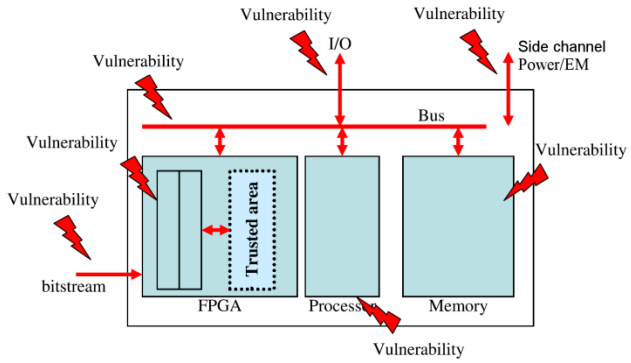


Abbildung 1: Angriffsvektoren auf einen FPGA [003, S. 15].

Um eine System sicher zu gestalten, sollten beim Konzept unter anderem folgende Faktoren beachtet werden [004]:

- Der Wert der Entwicklung (engl. value):
  Der Schutz und die ausgewählten Sicherheitseigenschaften sollten im Verhältnis zu den zu schützenden Werten stehen.
- Der Gegner (engl. adversary):
  Je umfangreicher die technische Ausrüstung, die Erfahrung und die zur Verfügung stehende Zeit des Angreifers sind, umso stärker müssen die Sicherheitseigenschaften des Produktes sein.
- Entwicklungsstand (engl. design stage):
  Es sollte möglichst früh festgelegt werden, welche Sicherheitseigenschaften im späteren Produkt integriert sein sollen.
- FPGA Ressourcen:
  Da besonders aktive AT-Sicherheitseigenschaften Logikelemente eines FPGAs benötigen, muss definiert werden, welcher Bereich dafür zur Verfügung steht.
- Sicherheitskonzept:
  Die einzelnen AT-Sicherheitseigenschaften sollten immer Teil eines schlüssigen Gesamtkonzeptes sein.

Die Sicherheitseigenschafen unterteilen sich dabei in passive Verfahren (z. B.: Bitstromverschlüsselung und -Authentifikation und die Deaktivierung der Auslesefunktion) und aktive Verfahren in den Bereichen Prävention, Erkennung (z. B. Spannungs- und Temperaturüberwachung) und Reaktion.

1.1 Eine Übersicht zu kryptographischen Angriffen

Kryptographische Angriffe können in drei Gruppen unterteilt werden ([005] S. 83):

- Klassische Kryptoanalyse:
  - Mathematische Analyse
  - Brute-Force-Angriff
- Implementationsattacken:
  - Seitenkanalattacken
  - Fehlerinjektion
- Social Engineering:
  - z. B. Passwörter kaufen oder erpressen

In der Vergangenheit wurden besonders die klassische Kryptoanalyse und das Social Engineering zum Umgehen von Sicherheitsmechanismen elektronischer Geräte verwendet.
Begünstigt durch fallende Preise und einen Anstieg der Leistungsfähigkeit im Bereich der Messtechnik und Manipulationstechnik elektronischer Schaltungen gewinnen die Implementationsattacken aber einen immer größeren Einfluss.
Die Implementationsattacken lassen sich weiter in die folgenden Teilbereiche unterteilen (vgl. [005]):

Aktive Verfahren:

- Invasiv
- Semi-invasiv
- Nicht invasiv:
  - Fehlerinduzierung (Störimpulse mit Strom, Spannung, Takt und Laser, Temperaturänderungen)
  - Manipulation mit FIB

Passive Verfahren:

- Seitenkanäle ausnutzen:
  - Timings
  - Stromverbrauch
  - Elektromagnetischen Abstrahlung

1.2 DPA-Analyse

Paul Kocher bewies 1998 erstmals, dass es mit einer sogenannten DPA-Analyse (Differential Power Analysis) möglich ist, vom Energieverbrauch bzw. der elektromagnetischen Abstrahlung eines elektronischen Bausteins (z. B. Mikrocontroller oder FPGA) auf seine inneren Arbeitsabläufe zu schließen (vgl. [006]).
Paul Kocher verfasste nach dieser Entdeckung viele Patente und gründete die Firma Cryptography Research, Inc., die 2011 von der Firma Rambus übernommen wurde. Firmen, die aktuell in ihren elektronischen Schaltkreisen DPA-Gegenmaßnahmen einsetzen, müssen jetzt Rambus Lizenzgebühren zahlen.
Dieser Patentkampf führt dazu, dass Firmen DPA-Gegenmaßnahmen nur dann einsetzen, wenn dies zwingend gefordert wird. Die Erfinder des neuen SHA-3-Hash-Algorithmus, die den Algorithmus ursprünglich unter dem Namen Keccak veröffentlichten, bieten aus diesem Grund als kostenlose Implementation des Algorithmus nur eine Variante ohne DPA-Gegenmaßnahmen auf ihrer Webseite an (vgl. [008]), obwohl eine gegen DPA-Gegenmaßnahmen gehärtete Version intern bereits vorliegt, getestet und in bereits schriftlich veröffentlicht wurde (vgl. [009]).
DPA-Angriffe sind seit 1998 bekannt, dennoch verfügen erst die allerneusten FPGA-Bausteinfamilien über integrierte Gegenmaßnahmen. Beim FPGA-Hersteller Xilinx gibt es die Gegenmaßnahmen erst ab der Ultrascale-Architektur, die seit 2014 auf dem Markt verfügbar ist ([007] S. 154).

1.3 Praktischer DPA-Angriff

Der folgende Abschnitt beschreibt zur praktischen Veranschaulichung der Thematik einen Versuchsaufbau der Kasper & Oswald GmbH für einen DPA-Angriff auf eine AES-Verschlüsselung mit einer Schlüssellänge von 128 Bit.

Für eine einfache DPA-Attacke genügen die folgenden Materialien:

- PC mit MATLAB, Oszilloskopsoftware, ein Konsolenprogramm (z. B. Tera Term VT) und Cyqwin-Umgebung
- Ein Oszilloskop (z. B. PicoScope 5243B) mit zwei Tastköpfen
- Eine EM-Sonde (RF-U 5-2 mit Vorverstärker PA303)

Der AES-Algorithmus hat den Aufbau aus Abbildung 2. Teile des unverschlüsselten Klartestes werden zunächst im Schritt "Key Addition Layer" über eine XOR-Funktion mit einem abgeleiteten Schlüssel k_0 verknüpft. Anschließend folgt in 10 gleich aufgebauten Runden die Verschlüsselung der ursprünglichen Bits.

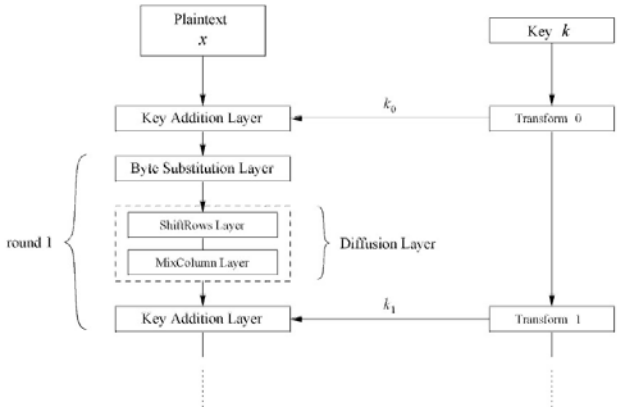


Abbildung 2: Ablauf des AES-Algorithmus mit 1. Runde [010].

Als Angriff bietet sich für die Analyse die Stelle vor und nach der S-Box an (vgl. Abbildung 3).

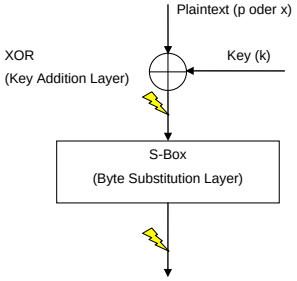


Abbildung 3: Angriffsmöglichkeiten auf den Algorithmus.

Bei der DPA-Analyse wird ausgenutzt, dass einerseits bekannt ist, wie der AES-Algorithmus abläuft und anderseits der Stromverbrauch für den Zustand 0 und 1 innerhalb einer Transistorzelle verschieden ist. Zusammen mit dem bekannten oder abgehörten Klartext (x) und dem Messergebnis (Strom- oder EM-Messung) kann auf den Schlüssel geschlossen werden.
Im ersten Schritt werden mit unterschiedlichen Klartexten Strom- oder EM-Messungen mit einem Oszilloskop aufgenommen (vgl. Abbildung 4). Bei einfachen Mikrocontrollern reichen oft schon weniger als 500 Messkurven aus. Bei FPGAs werden (je nach Implementierung und Baustein) mehrere tausend Messkurven benötigt.

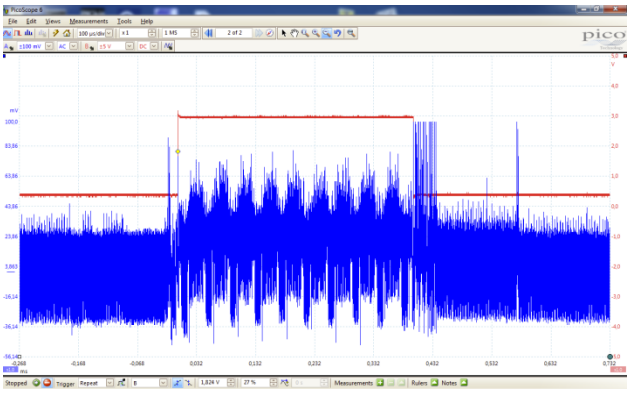


Abbildung 4: Stromverbrauchsmessung einer AES-Verschlüsselung.

Anschließend wird für alle Zustände (00_{Hex} bis FF_{Hex}) eines Schlüsselbytes (k_0) der Wert eines Klartextbytes (x) nach der S-Box berechnet und ermittelt, ob das MSB (Most Significant Bit) eine 0 oder 1 ist. Davon abhängig werden die Messkurven anschließend nach diesem Ergebnis (0 oder 1) sortiert.

Klartext x	k_0=00 $S(x \oplus k_0)$	k_0=01 $S(x \oplus k_0)$
ab	$4A_{Hex}$ $0100101\mathbf{0}_{Bit}$ Hamming = 3	$3E_{Hex}$ $0011111\mathbf{0}_{Bit}$ Hamming = 5

Tabelle 1: Ergebnis nach S-Box des AES-Algorithmus bei bekanntem Klartext und Schlüssel.

In einem letzten Schritt werden für jedes geratene Schlüsselbyte die Messkurven vom Ergebnis 0 von den Messkurven mit Ergebnis 1 abgezogen. Während bei falsch geratenen Schlüsselbytes keine deutlichen Differenzen auftreten, zeigt sich bei einem richtig geratenen Schlüsselbyte ein deutlicher Unterschied zwischen den Messkurven (vgl. Abbildung 5). Eine Tiefpassfilterung der Messergebnisse führt in einigen Fällen zu einer weiteren Optimierung der Messergebnisse.

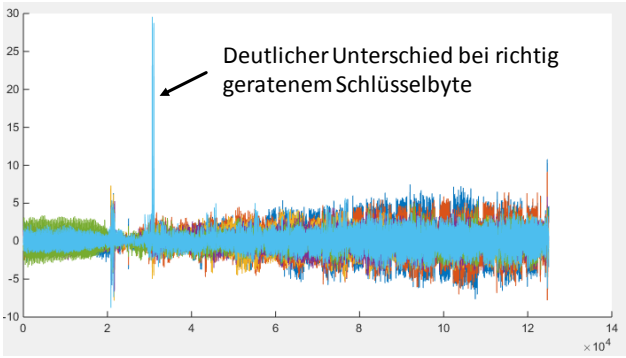


Abbildung 5: DPA-Analyse - Deutlicher Ausschlag im Messergebnis bei richtig geratenem Schlüsselbyte.

Die Aussagekraft der Ergebnisse kann noch weiter verbessert werden, wenn neben der Bewertung des MSB alle Einser mit dem sogenannten Hamming-Gewicht bewertet werden (vgl. Tabelle 1). Dies erfolgt in einer sogenannten CPA-Analyse (Correlation Power Analysis) und bietet sich besonders nach der S-Box an, da diese eine nichtlineare Funktion darstellt.

1.4 Gegenmaßnahmen

Die Quelle [007] enthält auf S. 154 eine Liste zu den aktuellen FPGA-Bausteinfamilien von Xilinx und deren Gegenmaßnahmen. Gegenmaßnahmen zu den genannten Bedrohungen werden ebenfalls in [013] aufgelistet.

1.5 Fazit

DPA-Attacken haben eine hohe Relevanz für die IT-Sicherheit eines Systems, da mathematisch hochsichere Algorithmen in kürzester Zeit mit einfachen Messmitteln gebrochen werden können, wenn der Angreifer in der Lage ist einen Zugang zur Hardware zu bekommen.
Entwickler sollten sich diese Thematik bewusst machen und DPA-Gegenmaßnahmen trotz der Lizenzproblematik anstreben, um eine hohe IT-Sicherheit in ihren Systemen zu erhalten.
Wie lange es dauert, bis Gegenmaßnahmen zu einer Bedrohung flächendeckend in einer Technologie eingesetzt werden, wurde am Beispiel der DPA-Attacken gezeigt.
Die Praxiserfahrung zeigt, dass Entwickler häufig IT-Sicherheitsmechanischmen nur dann einsetzten, wenn diese ausdrücklich gefordert wurden, da diese die Komplexität der zu entwickelnden Geräte in allen Produktphasen erhöhen. Es müssen keine weiteren Algorithmen entwickelt werden, sondern die bestehenden müssen angewandt werden.

2 Isolation Design Flow & Partielle Rekonfiguration

Im Bereich der IT-Sicherheit aber auch Funktionssicherheit wird häufig gefordert, dass gerade bei Ein-Chip-Systemen in einem Bauteil mehrere Funktionen parallel nebeneinander ablaufen sollen, ohne sich gegenseitig zu stören. Eine Kommunikation zwischen den isolierten Bereichen darf entweder gar nicht oder nur über spezielle Verbindungen dem sogenannten "Trusted Routing" erfolgen. Eine lokale Trennung von Funktionen auf dem Chip kann auch gefordert sein, um im Falle eines Angriffs dafür zu garantieren, dass nur ein Teilbereich ausfällt (z. B. durch den Beschuss mit einem Laser).
Da Angreifer häufig versuchen Entscheidungspunkte zu manipulieren (z. B. das Ende einer Passwortprüfung) verwenden Systeme mit erhöhter IT-Sicherheit redundante Strukturen. In der neusten FPGA-Architektur von Xilinx (Zynq Ultrascale+) ist der A-PU-Prozessor z. B. vierfach redundant in der Hardware aufgebaut (vgl. [014]).
Beispielhaft zeigt Abbildung 6 ein von der HTV GmbH realisiertes Floorplanning eines FPGA-Chips mit vier isolierten Zonen.

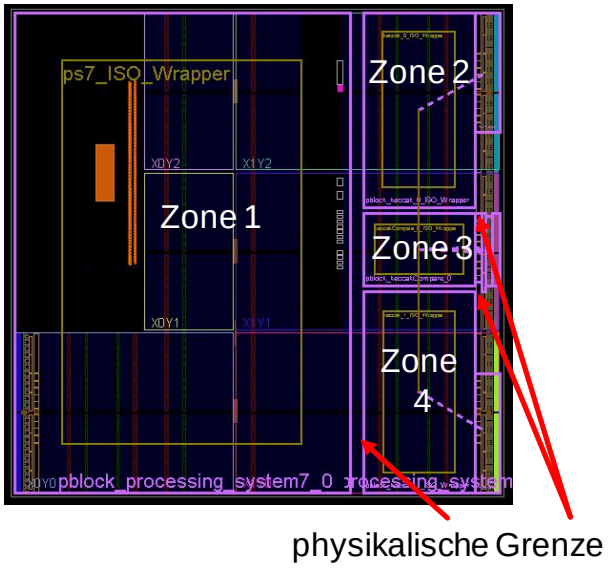


Abbildung 6: Floorplanning-Ansicht eines SoC-FPGA-Chips (Zynq-7000) mit vier isolierten Zonen.

Für die Isolation von Funktionen bietet die Firma Xilinx für FPGAs den "Isolation Design Flow" an, wie er in der XAPP1256 beschrieben ist.
Durch diesen speziellen Ablauf wird sichergestellt, dass die internen Verbindungen bzw. das Routing spezielle Vorgaben einhält und eine physikalische Grenze zwischen benachbarten Zonen verläuft (vgl. Abbildung 7). Die Grenze entsteht dadurch, dass sich in diesem Bereich keine konfigurierte Logik oder Switchboxen befinden, die eine Übertragung von Informationen zwischen den isolierten Zonen ermöglichen.

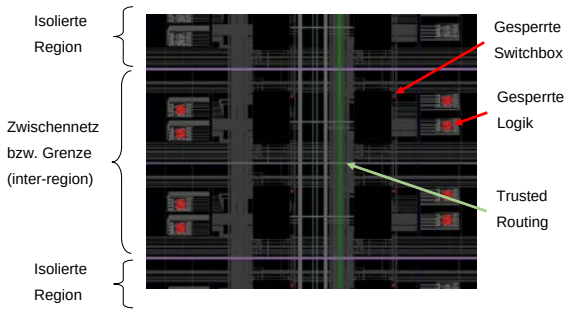
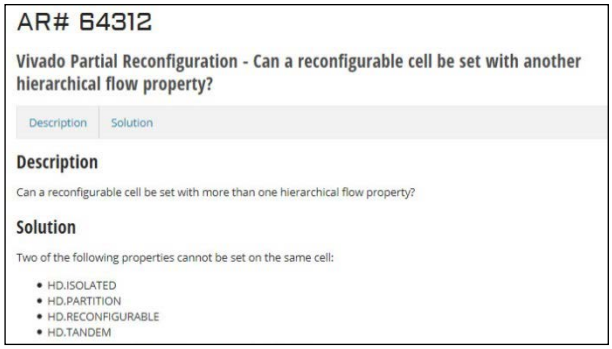


Abbildung 7: Regionen in einem FPGA werden durch physikalische Grenzen voneinander getrennt. Nur Verbindungen, die die Vorgaben des Trusted Routings einhalten, dürfen durch die festgelegten Grenzen verlaufen.

In dem Forschungsprojekt der Fa. HTV GmbH (vgl. [001]) wurde überprüft, ob es möglich ist, die Konfiguration der isolierten Bereiche im laufenden Betrieb zu ändern. Der FPGA-Hersteller Xilinx stellt für Änderungen der Konfiguration im Betrieb einen partiellen Design Flow in seiner Entwicklungsumgebung Vivado zur Verfügung. Eine erste Analyse zeigt aber, dass die beide Design Flows nicht miteinander kombiniert werden können (vgl. Abbildung 8). Aktuell ist die Vereinigung beider Flows für das dritte Quartal von 2018 angekündigt.


AR# 64312

Vivado Partial Reconfiguration - Can a reconfigurable cell be set with another hierarchical flow property?

Description | Solution

Description

Can a reconfigurable cell be set with more than one hierarchical flow property?

Solution

Two of the following properties cannot be set on the same cell:

- HD.ISOLATED
- HD.PARTITION
- HD.RECONFIGURABLE
- HD.TANDEM

Abbildung 8: Der "Isolation Design Flow" und der "partielle Design Flow" lassen sich in Vivado 2017.X nicht vereinen.

Über einen Umweg ist es aber doch möglich beide Flows miteinander zu vereinen. Mit der Software GoAhead (vgl. [012]) kann das Routing der statischen Konfiguration eines FPGAs nach definierten Vorgaben erzeugt werden, um so ein gewünschtes Bussystem in einem vorgegebenen Bereich im FPGA zu realisieren (vgl. Abbildung 9).

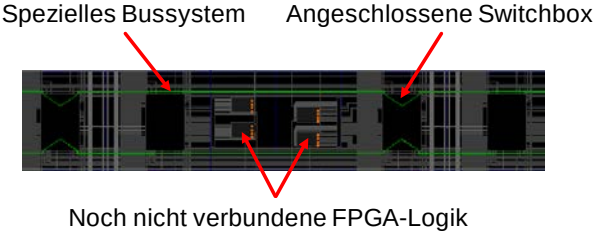


Abbildung 9: Spezielles Bussystem, das an definierte Switchboxen angeschlossen ist.

Im Betrieb kann jetzt in den Bereich in dem sich das erzeugte Bussystem befindet ein anderer Inhalt in die Switchboxen und die damit verbundene FPGA-Logik mit der gewünschten Funktionalität (z. B. ein Verschlüsselungsalgorithmus) geladen werden. Die Konfiguration der Switchboxen und der Logik wird durch den partiell geladenen Bitstrom geändert, das statische Routing wird dadurch aber nicht beeinflusst.
Abbildung 10 zeigt ein zu diesem Vorgehen passendes partielles Modul. Zur Erzeugung dieses Moduls wird das statische Bussystem am äußeren Rand des Moduls nachgebildet. Die im statischen System angeschlossenen Switchboxen stellen jetzt die Schnittstelle zum partiellen Modul her. Die Größe des Moduls ergibt sich aus dessen Funktionalität.

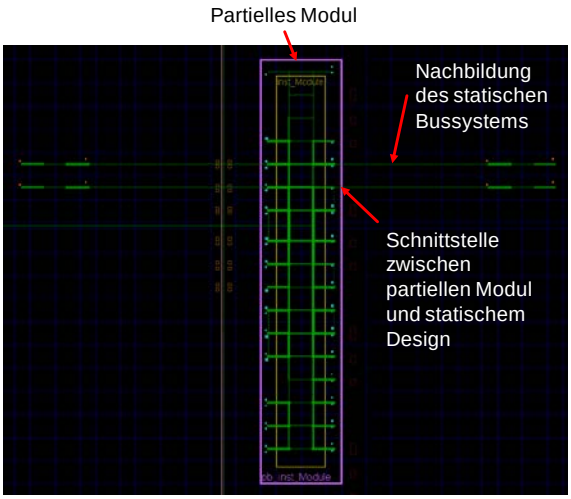


Abbildung 10: Partielles Modul.

Durch diese Schnittstellendefinition, die erst durch das von GoAhead erzeugte Bussystem möglich wird, können partielle Module später in einer beliebigen Vivado Version separat von der statischen Konfiguration entwickelt werden, da sich das Routing zwischen der partiellen und statischen Konfiguration nicht mehr ändert.
In einem ersten Schritt kann die statische Konfiguration in Vivado ohne hohe Sicherheitsauflagen entwickelt und deren Zonen auf Isolation hin überprüft werden. Erst zur Laufzeit werden dann die gewünschten geheimen partiellen Module in die im statischen Design vorhanden Bereiche nachgeladen.

2.1 Fazit

Das demonstrierte Konzept zur partiellen Rekonfiguration von partiellen Modulen in einer isolierten statischen Konfiguration gelang nach Wissen des Autors weltweit erstmals Ende 2017 in Zusammenarbeit mit der University of Manchester.

Die Vorteile der Schnittstellendefinitionen in der statischen Konfiguration sind enorm:

- Die Vorteile des "Isolation Design Flows" und der partiellen Rekonfiguration können vereint werden.
- Geheime Module können separat entwickelt und erst bei Bedarf im Betrieb in die gewünschte isolierte Zone der statische Konfiguration geladen werden.
- Durch eine weitere Manipulation des Bitstroms eines partiellen Moduls ist es zusätzlich möglich dieses an unterschiedlichen Stellen in der statischen Konfiguration zu laden, solange sich dort die passenden Schnittstellen bzw. das passende Bussystem und die passenden Hardwareressourcen befinden.

3 Literatur

[001] HTV GmbH: Forschungsvorhaben - Programmierung von FPGAs. Für eine deutsche Bundesbehörde, 24.11.2017.

[002] HTV GmbH: Studie - Analyse von FPGAs. Für eine deutsche Bundesbehörde, 28.11.2012.

[003] B. Badrinans, J.L. Danger, V. Fischer, G. Gogniat, L. Torres: Security Trends for FPGAs. Springer 2011. ISBN: 978-94-007-1337-6.

[004] E. Peterson: Developing Tamper Resistant Designs with Xilinx Virtex-6 and 7 Series FPGAs. Xilinx, Application Note, XAPP1084 (v1.3) October 15, 2013.

[005] Prof. C. Paar: Implementation of Cryptographic Schemes 1. Ruhr University Bochum, Embedded Security, April 2014.

[006] P. Kocher, J. Jaffe, B. Jun: Differential Power Analysis. Cryptography Research, Inc. 1999.

[007] Xilinx. ultrafast embedded design methodology guide. https://www.xilinx.com/support/documentation/sw_manuals/ug1228-ultrafast-embedded-design-methodology-guide.pdf, UG1228 (v1.0), 31. Maerz, 2017.

[008] TeamKeccak: Keccak. https://keccak.team/keccak.html, 27.03.2018.

[009] Michael Peeters Gilles Van Assche Ronny Van Keer Guido Bertoni, Joan Daemen. Tuturial on Keccak. *Security Pattern - STMicroelectronics - Radboud University, Oktober*, 2017.

[010] David Oswald: Seitenkanal---Angriffe: Theorie und Praxis. KASPER & OSWALD GmbH, November 2014.

[011] Huiyun Li: Security evaluation at design time for cryptographic hardware. University of Cambridge, April 2006.

[012] Dirk Koch: GoAhead - A Partial Design Tool. http://www.mn.uio.no/ifi/english/research/projects/cosrecos/goahead/, 27.03.2018.

[013] Steven McNeil: Solving Today's Design Security Concerns. Xilinx, WP365 (v1.2) July 30, 2012.

[014] Xilinx: Zynq UltraScale+ MPSoC Product Advantages. https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html, 28.03.2018.