FPGAs in Anwendungen mit hoher IT-Sicherheit

FPGAs bieten ein hohes Maß an Security, doch Power Analysis und ungeschicktes Layout machen diese Bausteine angreifbar. Warum ist DPA-Schutz wichtig, und wie erreicht man einen sicheren Design Flow?

THOMAS KUHN

FPGAs werden besonders durch ihre Eigenschaft, Aufgaben mit sehr hoher Geschwindigkeit abzuarbeiten, geschätzt. Auch ihre Flexibilität und Reprogrammierbarkeit erfreuen sich zunehmender Beliebtheit: 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. Kein Wunder also, dass FPGAs auch in sicherheitsrelevanten Anwendungen im Kommen sind.

Doch leider gibt es eine Vielzahl von Angriffsmöglichkeiten, die auf elektronische Schaltungen, die einen FPGA verwenden, anwendbar sind. Die folgenden Erkenntnisse und Analysen stammen aus einer Studie, welche die HTV GmbH für eine deutsche Bundesbehörde erarbeitet hat. Die dabei angewandten „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.

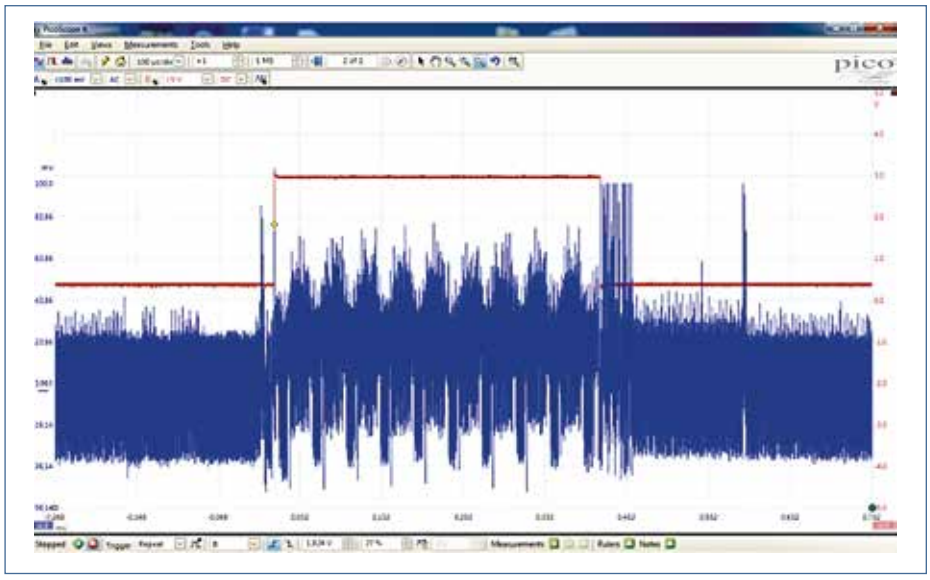
*** Dipl.-Ing. Thomas Kuhn**
ist Assistent der Geschäftsleitung der HTV Halbleiter-Test & Vertriebs-GmbH in Bernsheim.

Sicherheitsüberlegungen für integrierte Schaltkreise

Um ein System sicher zu gestalten, sollten beim Konzept unter anderem folgende Faktoren beachtet werden:

- Der Wert der Entwicklung (engl. value): Der Schutz und die ausgewählten Sicherheitseigenschaften sollten im Verhältnis zu den zu schützenden Werten stehen.
- Der Angreifer: Je umfangreicher technische Ausrüstung, Erfahrung und 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 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.



Bilder: HTV

Bild 1: *Stromverbrauchsmessung einer AES-Verschlüsselung. Der Algorithmus arbeitet in mehreren Runden, was im Stromverbrauch deutlich zu erkennen ist.*

Eine Übersicht zu kryptographischen Angriffen

Kryptographische Angriffe können in drei Gruppen unterteilt werden:

- Klassische Kryptoanalyse: z.B. Mathematische Analyse, Brute-Force-Angriffe
- Implementationsattacken: z.B. Seitenkanalattacken, Fault Injection
- 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.

Seitenkanal-Angriffe durch DPA (Differential Power Analysis)

Paul Kocher demonstrierte 1998 erstmals einen Angriff via DPA-Analyse (Differential Power Analysis). Damit ist es möglich, vom Energieverbrauch bzw. der elektromagneti-

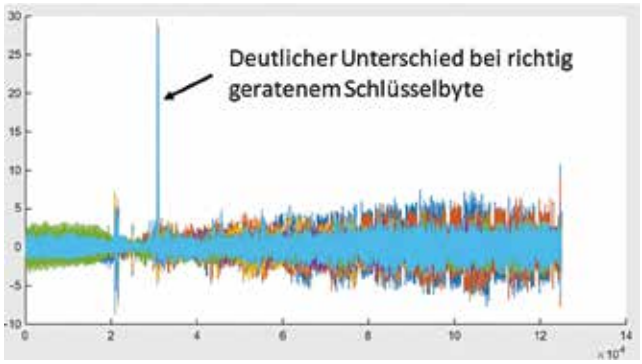


Bild 2:
DPA-Analyse - Deutlicher Ausschlag im Messergebnis bei richtig geratenem Schlüsselbyte.

Klartext x	k_0=00 S(x ⊕ k_0)	k_0=01 S(x ⊕ k_0)
ab	$4A_{Hex}$ 01001010_{Bit} Hamming = 3	$3E_{Hex}$ 00111110_{Bit} Hamming = 5

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

schen Abstrahlung eines elektronischen Bausteins (z. B. MCU oder FPGA) auf seine inneren Arbeitsabläufe zu schließen.

Paul Kocher ließ sich Schutzlösungen zu dieser Form von Angriffen patentieren und gründete Cryptography Research, das 2011 von der Firma Rambus übernommen wurde. Firmen, die aktuell in ihren elektronischen Schaltkreisen DPA-Gegenmaßnahmen einsetzen, müssen in der Regel Lizenzgebühren an Rambus zahlen. Aus Kostengründen verzichten daher einige Hardwarehersteller darauf, entsprechende Algorithmen in ihren Bausteinen zu implementieren.

DPA-Attacken besitzen allerdings 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.

Ein anschauliches Beispiel eines DPA-Angriffes

Warum ein Schutz vor DPA-Angriff so wichtig ist, soll der folgende Abschnitt veranschaulichen. Dieser zeigt 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:

- Ein 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)

Bei einem Angriff auf einen in der Softwareimplementation sicheren Algorithmus nutzt der Angreifer bei einer Hardwareimplementation aus, dass er die Daten bzw. den Klartext kennt, die er dem Algorithmus schickt. Ihm ist auch der innere Aufbau des Algorithmus bekannt, da dieser öffentlich standardisiert ist. So ist er in der Lage, zu jeder Stelle im Algorithmus die aktuellen Werte der Ursprungsnachricht zu berechnen.

Der Angriff erfolgt dann typischerweise an Stellen im Algorithmus an denen eine Berechnung mit dem geheimen Schlüssel durchgeführt wird (z. B. nach einer XOR-Funktion oder S-Box). Für den Angriff wird der unterschiedliche Stromverbrauch in den Transistorzellen der Hardware bei der Verarbeitung von 0 und 1 ausgenutzt.

Zuerst nimmt der Angreifer mit unterschiedlichen Klartexten Messkurven vom Stromverbrauch oder der elektromagnetischen Abstrahlung mit einem Oszilloskop auf (vgl. Bild 1). Bei MCUs reichen oft weniger als 500 Messkurven für die spätere Analyse aus, bei FPGAs sind je nach Implementierung und Baustein mehrere tausend Messkurven nötig. Anschließend wird für alle Zustände (00_{Hex} bis FF_{Hex}) eines Schlüsselbytes der Wert eines Klartextbytes an einer bestimmten Stelle im Algorithmus berechnet und ermittelt, ob das MSB (Most Significant Bit) eine 0 oder 1 ist. Davon abhängig sortiert der Angreifer die Messkurven (nach 0 oder 1).

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. Bild 2). Eine Tiefpassfilterung der Messergebnisse führt in einigen Fällen zu einer weiteren Optimierung der Messergebnisse.

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

Wie zu sehen ist, lässt sich ein DPA-Angriff mit relativ leichten Mitteln umsetzen. Entwickler sollten daher bei der Auswahl ihrer Hardware prüfen, ob Gegenmaßnahmen gegen DPA-Angriffe vorhanden sind. Nicht in allen auf dem Markt verfügbaren Bausteinen sind diese ab Werk vorhanden: FPGA-Hersteller Xilinx bietet etwa aktuell nur in den neusten Bausteinfamilien UltraScale und UltraScale+ standardmäßig integrierte DPA-Gegenmaßnahmen. Anbieter Microsemi hat zum Schutz seiner Low- und Mid-Range-FPGA-Familien eine entsprechende Lizenz von Rambus erworben und implementiert.

Isolation Design Flow und Partielle Rekonfiguration

Gerade bei Ein-Chip-Systemen ist häufig gefordert, dass in einem Bauteil mehrere Funktionen parallel nebeneinander ablaufen sollen, ohne sich gegenseitig zu stören. Das ist sowohl aus Security-Aspekten als auch aus Sicht der funktionalen Sicherheit relevant. 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.

Hier ein Beispiel anhand eines Xilinx-Bausteins: Bild 3 zeigt ein von der HTV GmbH realisiertes Floorplanning, bei dem die Fläche eines Zynq 7000 MPSoC-FPGA-Chips in vier isolierte Zonen aufgeteilt wurde. Für die Isolation von einzelnen Bereichen im FPGA bietet die Firma Xilinx den „Isolation Design Flow“ XAPP1256. 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. Bild 4). Die Grenze entsteht dadurch, dass sich in diesem Bereich keine konfigurierte Logik oder Switchboxen befinden, die Informationen vom Trusted Routing abhören oder beeinflussen können.

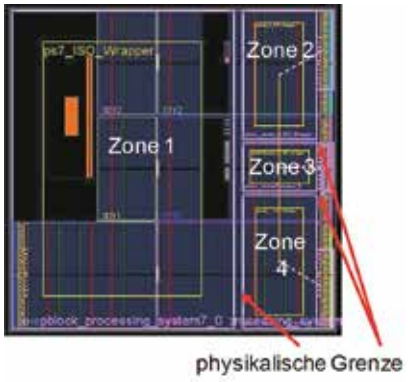


Bild 3: *Floorplanning-Ansicht eines MPSoC-FPGA-Chips (Zynq-7000) mit vier isolierten Zonen.*

Die HTV GmbH überprüfte in einer Studie, 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 per se nicht miteinander kombiniert werden können. Die Vereinigung beider Flows ist für Zynq-7000-Bausteine erst ab Vivado 2018.2 verfügbar; für UltraScale ist dies erst ab 2019 geplant. Über einen Umweg war es aber bereits 2017 möglich, beide Flows zu vereinen. Das Designtool GoAhead (https://bit.ly/2uY2Nzt) erlaubt, das Routing der statischen Konfiguration eines FPGAs nach definierten Vorgaben zu erzeugen, um so ein gewünschtes Bussystem in einem vorgegebenen Bereich im FPGA zu realisieren.

Vorteile der Kombination beider Design Flows

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.

Wenn partielle Module erst bei Bedarf im Betrieb in isolierte Zonen eines FPGAs geladen werden, lässt sich die Sicherheit von FPGAs deutlich verbessern. Wer dies berücksichtigt, und sein Design auch gegen Seitenkanal-Attacken wie DPA sichert, macht sein System erst robust - sowohl in Sachen funktionaler Sicherheit als auch Security. // **SG**

HTV Halbleiter-Test & Vertriebs-GmbH

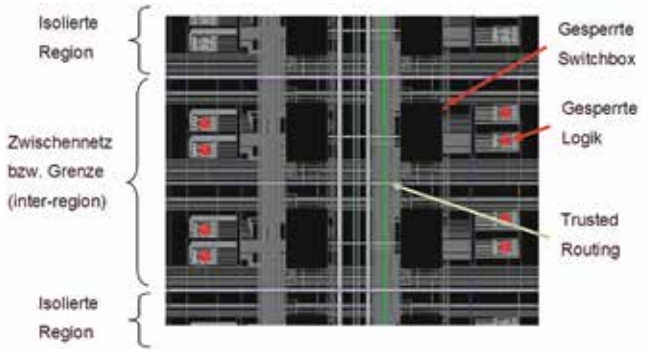


Bild 4: *Regionen in einem FPGA werden durch physikalische Grenzen voneinander getrennt. Nur Verbindungen, die die Vorgaben des Trusted Routings einhalten, dürfen durch die festgelegten Grenzen verlaufen.*

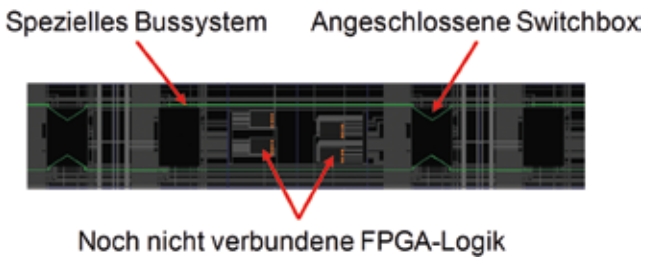


Bild 5: *Spezielles Bussystem, das an definierte Switchboxen angeschlossen ist.*