Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Kevin-Pascal Kumeth

Seitenkanalanalyse einer FPGA AES Implementierung

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Kevin-Pascal Kumeth

Seitenkanalanalyse einer FPGA AES Implementierung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüferin : Prof. Dr. rer. nat. Heike Neumann
Zweitgutachter : Prof. Dr.-Ing. Robert Fitz

Abgegeben am 7. März 2016

Kevin-Pascal Kumeth

Thema der Bachelorarbeit

Seitenkanalanalyse einer FPGA AES Implementierung

Stichworte

Kryptographie, VHDL, FPGA, Seitenkanalanalyse, SPA, AES, Hamming-Distanz, Hamming-Gewicht

Kurzzusammenfassung

In dieser Arbeit wird mittels Seitenkanalanalyse die Hamming-Distanz von verschiedenen Daten gemessen. Hierdurch kann der geheime Schlüssel der Verschlüsselung bestimmt werden. Die Voraussetzung für eine solche Messung wird besprochen, auftretende Probleme benannt und Lösungsansätze diskutiert.

Kevin-Pascal Kumeth

Title of the paper

Side-Channel Analysis of an FPGA AES Implementation

Keywords

Cryptography, VHDL, FPGA, Side-Channel Analysis, SPA, AES, Hamming distance, Hamming weight

Abstract

Inside this report the Hamming distance of various data is analysed via side-channel analysis for the purpose of revealing the secret key of the encription. The modification of a development FPGA board is explained and issues are discussed with possible solutions.

Inhaltsverzeichnis

I. Einführung **6**

1. Einleitung **7**
1.1. Aufgabenstellung 7

2. AES **9**
2.1. Endliche Körper 9
2.2. Verschlüsselungsalgorithmus 11
2.2.1. Substitutions-Layer 13
2.2.2. Permutations-Layer 15
2.2.3. Schlüssel-Layer 16
2.3. Implementierungen 16

3. Hardware-Angriffe **17**
3.1. Invasive Angriffe 17
3.2. Nichtinvasive Angriffe 17
3.2.1. Simple Power Analysis (SPA) 17
3.2.2. Differential Power Analysis (DPA) 18
3.3. Semi-Invasive Angriffe 18

4. Leistungsaufnahme CMOS **19**
4.1. Statische Leistungsaufnahme 19
4.2. Dynamische Leistungsaufnahme 19

II. Hardware Angriff **20**

5. Analyse **21**
5.1. Zielgerät - Basys3 21
5.1.1. Versorgung 22
5.1.2. Leistungsaufnahme und Ressourcen 23
5.2. Versuchsaufbau 24
5.2.1. Modifikation am Board 24

5.2.2. Messequipment 27

6. Attacke **28**
6.1. Voruntersuchung des Angriffspunktes 28
6.2. Angriff auf den AES 31
6.2.1. Angriffspunkt im Algorithmus 31
6.2.2. Testumgebung 31
6.2.3. Messung 32
6.3. Auswertung 33

7. Störungen **34**
7.1. EMV-Einstrahlung 34
7.2. Temperatur 35
7.3. Triggersignal 37
7.4. Weißes Rauschen 38

III. Ausblick **39**

8. Fazit **40**
8.1. Verbesserungsmöglichkeiten 41
8.1.1. Umwelteinflüsse minimieren 41
8.1.2. Versuchsaufbau verbessern 41
8.1.3. Angriffspunkt wechseln 41

9. Anhang **42**

Literaturverzeichnis **43**

Tabellenverzeichnis **44**

Abbildungsverzeichnis **45**

Teil I.

Einführung

1. Einleitung

In einer zunehmend digitalisierten Welt ist die Sicherheit ein wichtiger Aspekt. Hierzu werden in immer mehr Systemen die Daten verschlüsselt um sie gegen Angriffe zu schützen. Neben mathematischen Angriffen und dem ausnutzen von Softwarefehlern, gibt es das Feld der Seitenkanalanalysen.
Bei einer Seitenkanalanalyse handelt es sich um einen nicht-invasiven beziehungsweise semi-invasiven Angriff. Das heißt das anzugreifende Objekt wird nicht aktiv beeinflusst und beschädigt. Im Bezug auf digitale Systeme geht es zum Beispiel um die Analyse der Leistungsaufnahme, die ein System zu bestimmten Zeitpunkten aufweist.
Eine Verschlüsselung stellt eine Verknüpfung von Daten mit einem geheimen Schlüssel da. Es finden arithmetische Operationen statt. Dabei werden die Daten mit dem Schlüssel verbunden werden. Dies führt zu einer unterschiedlichen Leistungsaufnahme bei verschiedenen Daten.
Im Falle des *Advanced Encryption Standard*, welcher in dieser Arbeit angegriffen wird, werden Schlüssel und Daten mittels einer XOR-Operation miteinander verknüpft.

1.1. Aufgabenstellung

Ziel dieser Arbeit ist es die Durchführung einer Seitenkanalanalyse einer AES Implementierung an einem FPGA-Development-Board neuerer Generation.
Hierzu wird ein Board modifiziert und mit einem Messpunkt vorbereitet. Anhand eines selbst gewählten Angriffspunktes wird das Leckverhalten des Algorithmus untersucht. Hierfür soll die Hamming-Distanz von Daten und Schlüssel bestimmt werden. Eine frei erhältliche AES Implementierung wird für diese Untersuchung genutzt.
Abschließend wird ein Ausblick gegeben, wie die Seitenkanalanalyse verbessert werden kann und welche Problemstellungen bestehen.

Zur Durchführung dieser Arbeit werden, neben einem handelsüblichen PC, folgende Geräte und Software verwendet:

- Geräte:
  - PicoScope 5444B
  - Basys3 Board - Xilinx Artix7-FPGA
  - Rohde & Schwarz - DC Power Supply
  - Grundig - Regel-Trenn-Transformator
- Software:
  - Xilinx Vivado 2014.4
  - ModelSim PE Student Edition 10.4a
  - Octave 4.0.0
  - TeXstudio
  - yEd

2. AES

Beim *Advanced Encryption Standard (AES)* handelt es sich um eine Blockchiffre, welche im Jahr 2001 nach einem Ausschreibungsverfahren als offizieller Standard mit der FIPS-197[1] NIST (2001) veröffentlicht wurde. Der AES gilt als Nachfolger des *Data Encryption Standard (DES)*, welcher Aufgrund der kurzen Schlüssellänge (56 Bit) inzwischen anfällig gegen vollständige Schlüsselsuche ist.
Die Blocklänge ist beim AES auf 128 Bit beschränkt mit Schlüssellängen von 128, 192 oder 256 Bit. Dieser ist Abgeleitet vom Rijndael-Algorithmus mit welchem auch 128, 160, 192, 224 und 256 Bit als Block- und Schlüssellängen möglich sind.
Als Grundlage für dieses Kapitel werden die Bücher von Paar und Pelzl (2010) und Ertel (2012) verwendet. Wenn nicht anders Beschrieben wird von einer Daten- und Schlüssellänge von 128 Bit ausgegangen.

2.1. Endliche Körper

Eine mathematische Gruppe beschreibt eine Menge von Elementen G für die eine Operation $\circ$ mit zwei Elementen definiert ist. Sie besitzt folgende Eigenschaften:

- Die Operation ist abgeschlossen. Es gilt $a \circ b = c \in G$ für alle $a, b \in G$.
- Die Operation ist assoziativ. Es gilt $a \circ (b \circ c) = (a \circ b) \circ c$ für alle $a, b, c \in G$.
- Es gibt ein neutrales Element $e \in G$. Es gilt $a \circ e = e \circ a = a$ für alle $a \in G$.
- Für jedes Element $a \in G$ existiert ein inverses Element $a^{-1} \in G$ für das gilt $a \circ a^{-1} = a^{-1} \circ a = 1$.
- Eine Gruppe ist Kommutativ, auch Abelsche Gruppe genannt, wenn gilt $a \circ b = b \circ a$ für alle $a, b \in G$.

[1] Federal Information Processing Standards Publication

Bei einem Körper handelt es sich um eine Menge mit zwei Operationen. Ein endlicher Körper, auch Galoiskörper genannt, besitzt dabei eine endliche Menge von Elementen im Gegensatz zu Körpern wie den reellen Zahlen oder imaginären Zahlen. Aufgrund der Abgeschlossenheit kann dieser Wertebereich per Definition nicht verlassen werden. Dies ist in digitalen Systemen essenziell wichtig um korrekte Berechnungen zu ermöglichen. Der AES verwendet den Körper GF(2^8). Dies entspricht genau der Größe eines Bytes und ist daher gut für eine 8 Bit Mikrocontroller Implementierung geeignet. In Tabelle 2.1 sind exemplarisch die Addition und Multiplikation für den Körper GF(2) aufgeführt.

+	0	1
0	0	1
1	1	0

x	0	1
0	0	0
1	0	1

Tabelle 2.1.: Operationen im Körper GF(2)

Ein Element ($b_7\,b_6\,b_5\,b_4\,b_3\,b_2\,b_1\,b_0$) mit b $\in$ 0,1 des Körpers GF(2^8) wird dabei als Polynom betrachtet:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 \tag{2.1}$$

Für arithmetische Operationen[2] gilt die Polynomrechnung mit anschließender Reduzierung (modulo) mit dem charakteristischen Polynom:

$$x^8 + x^4 + x^3 + x + 1 \tag{2.2}$$

Ausführlichere Erläuterungen zu diesem Thema werden im Buch *Endliche Körper* von Kurzweil (2008) behandelt.

[2] Finite field arithmetic

2.2. Verschlüsselungsalgorithmus

Die Verschlüsselung des Klartext-Blockes erfolgt in Abhängigkeit der Schlüssellänge über mehrere Runden. Es handelt sich hierbei um ein Substitutions-Permutations-Netzwerk. Im Gegensatz zu einer Feistelchiffre ist eine Entschlüsselung nicht durch einfaches umstellen möglich, stattdessen müssen die Transformationen durch ihre Inversen ersetzt werden. Der gesamte Ablauf des AES ist in Abbildung 2.1 dargestellt.

Schlüssellänge	Rundenanzahl
128 bit	10
192 bit	12
256 bit	14

Tabelle 2.2.: AES - Rundenanzahl bei verschiedenen Schlüssellängen

Es wird Byte-Weise gearbeitet, 4 Bytes bilden dabei ein Wort. Dadurch ergeben sich 4 (128 Bit), 6 (192 Bit) bzw. 8 (256 Bit) Wörter. Die Daten und der Schlüssel werden wie in Tabelle 2.3 als Matrix dargestellt. Diese Darstellung erleichtert das Verständnis für den Permutations-Layer.

A0	A4	A8	A12
A1	A5	A9	A13
A2	A6	A10	A14
A3	A7	A11	A15

Tabelle 2.3.: AES - Darstellung von 128 Bit als Matrix

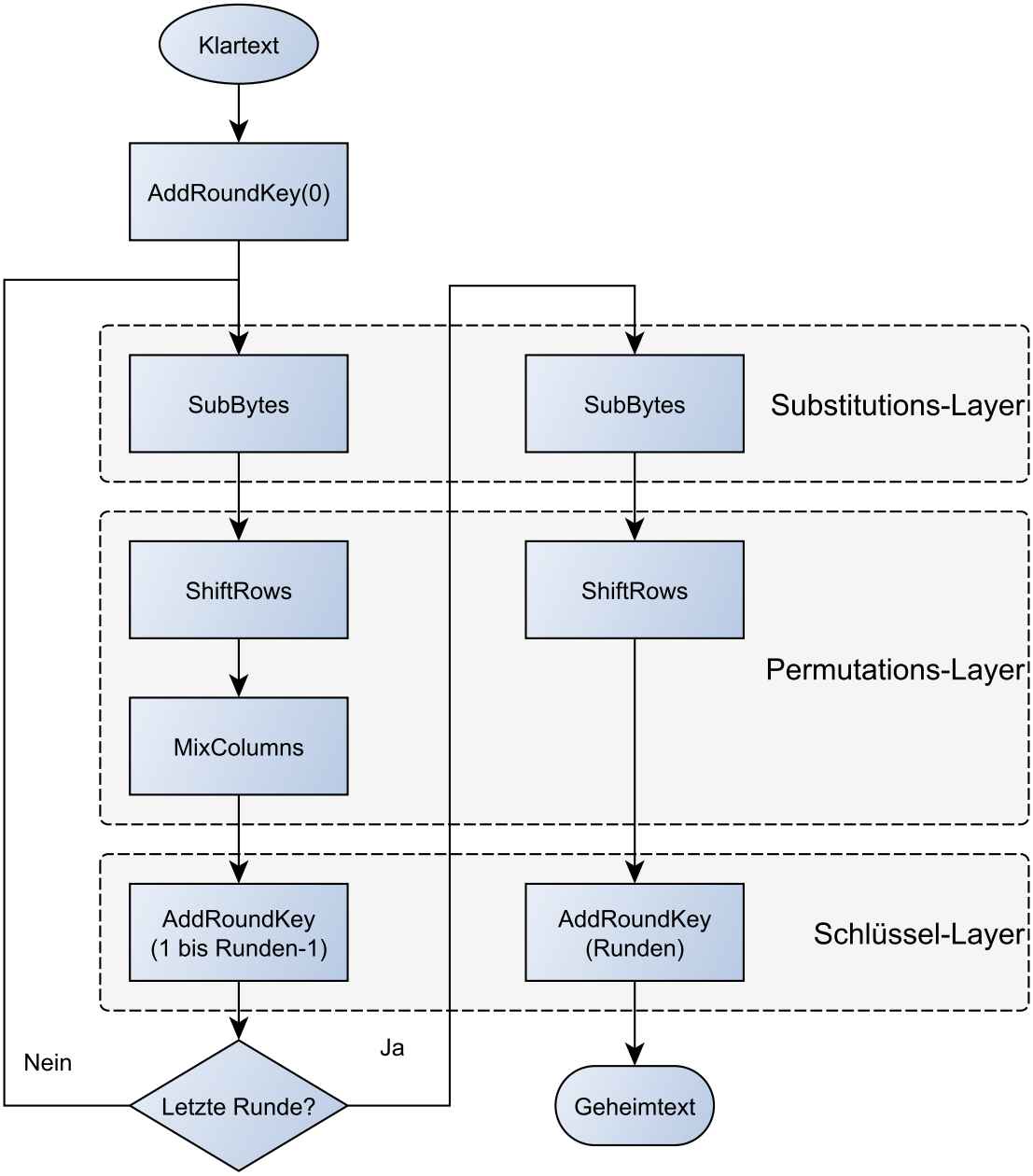


Abbildung 2.1.: AES - Ablauf der Verschlüsselung

2.2.1. Substitutions-Layer

Im Substitutions-Layer wird Konfusion erreicht, diese ist hochgradig nicht-linear. Dadurch lassen sich aus den statistischen Eigenschaften des Geheimtextes keine statistischen Eigenschaften des Klartextes ableiten.

Byte Substitution (S-Box)

Bei der Byte Substitution werden einzelne Bytes ersetzt. Dies kann über eine Substitutionsbox (S-Box) wie in Tabelle 2.4 dargestellt erfolgen oder dynamisch mittels der Formel 2.3 berechnet werden. Es handelt sich um eine monoalphabetische Substitution[3]. Für die Entschlüsselung existiert eine inverse S-Box.

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Tabelle 2.4.: AES - S-Box (Verschlüsselung) für das Byte *xy*

[3] Im Gegensatz zur polyalphabetische Substitution wird nur ein Alphabet benutzt. Wie zum Beispiel bei der Caesar-Verschlüsselung.

Für die dynamische Berechnung wird Formel 2.3 verwendet. In Vektor *b* werden die inversen der Bytes eingesetzt.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{2.3}$$

2.2.2. Permutations-Layer

Im Permutations-Layer wird Diffusion erreicht, bei Änderung weniger Eingangbits ändern sich viele Ausgangsbits. Diese Eigenschaft nennt sich Lawineneffekt[4]. Die Operationen sind linear, somit gilt das Assoziativgesetz. Sie können auch ohne Kenntnis des Schlüssels zurückgerechnet werden.

Shift Rows

Wie in Tabelle 2.3 dargestellt liegen die Daten in Form einer zweidimensionalen Tabelle vor. Im Schritt *Shift Rows* wird die erste Zeile nicht, die zweite einmal, die dritte zweimal und die vierte Zeile dreimal nach links verschoben. Dargestellt in Abbildung 2.2.

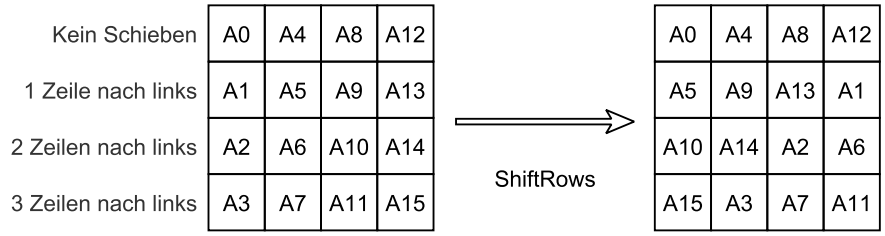


Abbildung 2.2.: AES - ShiftRows

Mix Columns

Im Schritt *Mix Columns* wird jede Spalte mit einer MDS[5]-Matrix multipliziert. Dadurch beeinflusst jedes Eingangsbyte alle Ausgabebytes.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} A0 \\ A1 \\ A2 \\ A3 \end{bmatrix}$$

[4] English: Avalanche Criterion
[5] *Maximum Distance Separable Code* - Codierung mit maximalen Hamming-Abstand

2.2.3. Schlüssel-Layer

AddRoundKey

Der geheime Schlüssel wird bitweise mit den Daten XOR verknüpft. Für jeden Durchgang *AddRoundKey* wird ein anderer Schlüssel verwendet. In der Vorrunde wird der gewählte geheime Schlüssel verwendet und bildet in dieser Arbeit den Angriffspunkt im Algorithmus.

Schlüsselexpansion

Um für jeden Aufruf von *AddRoundKey* einen anderen Rundenschlüssel zu verwenden, muss der ursprüngliche Schlüssel erweitert beziehungsweise expandiert werden.
Es werden insgesamt *Blockgröße ·(Runden +1)* Schlüsselbits benötigt. Im Fall eines AES mit der Schlüssel- und Blocklänge von 128 Bit: $128bit \cdot (10+1) = 1408Bit$ dies entspricht 44 Wörtern.
Die Expansion wird mittels XOR-Verknüpfungen der einzelnen Wörter erreicht. Ein neues Wort entsteht durch die Verknüpfung vom vorherigen Wort und dem Wort, welches eine Schlüssellänge vorher beginnt. Also das fünfte Wort ist das Ergebnis aus der Verknüpfung vom ersten und vierten Wort. Das sechste entsteht aus dem zweiten und fünften.
Wörter, die bei einem vielfachen der Schlüssellänge beginnen (Wörter 8,12,16,...), werden vor der XOR-Verknüpfung eine nichtlineare Substitution (S-Box) auf das vorherige Wort angewendet.

2.3. Implementierungen

Eine Anforderung des AES ist die hohe Performanz. Daher wird in der Einreichung des Algorithmus (siehe Daemen und Rijmen (1998)) auf verschiedene Implementierungsmöglichkeiten eingegangen. Eine detaillierte Betrachtung ist im dazugehörigen Buch von Daemen und Rijmen (2002) zu finden.
Für den Angriff dieser Arbeit ist die Breite von *AddRoundKey* entscheidend. Diese kann zwischen 8 Bit und einer kompletten Blocklänge liegen.

3. Hardware-Angriffe

In der Kryptographie wird zwischen drei Arten von Hardware Angriffen unterschieden. Diese können auch miteinander Kombiniert werden.

3.1. Invasive Angriffe

Bei invasiven Angriffen handelt es sich um solche durch die das anzugreifende Gerät physikalisch beeinflusst wird. Oft führt dies zur Beschädigung beziehungsweise zur Zerstörung des Gerätes.
Ein invasiver Angriff ist zum Beispiel das *Microprobing*. Hierbei wird mit Nadeln auf dem Halbleiter direkt gemessen.

3.2. Nichtinvasive Angriffe

Bei nichtinvasiven Angriffen wird das anzugreifende Gerät nicht beeinflusst. Ein Angriff ist somit später nicht erkennbar. Das Einfügen von Messwiderständen wird in dieser Arbeit nicht als Beeinflussung betrachtet.
Im folgenden wird näher auf Energieverbrauchsanalysen eingegangen.

3.2.1. Simple Power Analysis (SPA)

Bei der SPA werden Stromverläufe über die Zeit gemessen. Hierbei können mehrere Messungen vorgenommen werden, zum Beispiel für eine Mittelwertbildung. Es können auch verschiedene Eingangssignale mit ihrem dazugehörigen Stromprofil verglichen werden. Diese Art von Angriff wird in dieser Arbeit verwendet, um das Hamming-Gewicht von Daten und Schlüssel zu bestimmen.

3.2.2. Differential Power Analysis (DPA)

Wenn zusätzlich zur SPA statistische Methoden zur Auswertung der Messdaten genutzt werden so ist von einer DPA die Rede. Für diesen Angriff wird ein Leistungsaufnahmemodell erstellt welches auf die Messdaten angewendet wird. Hierbei muss der genaue Zeitpunkt des Angriffspunktes nicht bekannt sein, sondern sich nur innerhalb der Messdaten befinden.

3.3. Semi-Invasive Angriffe

Bei Semi-Invasiven Angriffen wird das zu testende Gerät physikalisch manipuliert jedoch nicht zerstört. Hierzu zählen Glitch-/Lichtattacken.

4. Leistungsaufnahme CMOS

Die Leistungsaufnahme von CMOS-Halbleitern teilt sich in statische und dynamische Leistungsaufnahme. Die spezifischen Werte für das Testsystem sind in Kapitel 5.1.2 aufgeführt. Eine detaillierte Betrachtung wird im Buch von Mangard, Oswald und Popp (2007) im Kapitel 3 behandelt.

4.1. Statische Leistungsaufnahme

Die Leistungsaufnahme, die ein Halbleiter ohne Schalten aufweist wird als statische Leistungsaufnahme bezeichnet. Diese entsteht durch physikalisch bedingte Leckströme, welche je nach Art des Halbleiters variieren und ist Temperaturabhängig (siehe Kapitel 7.2).
Bei der Messung stellt die statische Leistungsaufnahme einen Offset dar, der durch Wechselspannungsmessung herausgefiltert wird.

4.2. Dynamische Leistungsaufnahme

Als dynamische Leistungsaufnahme wird jeglicher Strom beim Schalten bezeichnet. Darunter fallen getaktete und kombinatorische Logik, welche ihren Zustand von 1 auf 0 oder von 0 auf 1 ändern.
Aufgrund des CMOS Aufbaus bei dem es sich um komplementär geschaltete p-Kanal- und n-Kanal-Feldeffekttransistoren handelt, besteht im Umschaltmoment eine Verbindung zwischen Versorgungsspannung und Masse.
Damit ist die dynamische Leistungsaufnahme von der Versorgungsspannung, dem Takt und den Daten abhängig. In Kapitel 6.1 wird mittels Messung anschaulich auf die Datenabhängigkeit eingegangen.

Teil II.

Hardware Angriff

5. Analyse

5.1. Zielgerät - Basys3

Das Basys3 (Abbildung 5.1) ist ein Low-Cost-FPGA-Developement-Board. Als FPGA wird ein Artix-7 (XC7A35T-1CPG236C) verwendet.
Für die Kommunikation stehen 16 Schalter und LEDs, fünf Taster, eine Siebensegmentanzeige mit vier Stellen, eine USB-UART Brücke (FTDI FT2232HQ), ein USB-Host, ein VGA-Anschluss und vier PMod Anschlüsse zur Verfügung.
Die Programmierung kann über JTAG, USB oder SPI mittels Flash Modul (S25FL032) erfolgen. Die Taktversorgung erfolgt über einen 100MHz Oszillator.
Weiterführende Informationen zum Board sind im Basys3 Resource Center zusammengefasst.

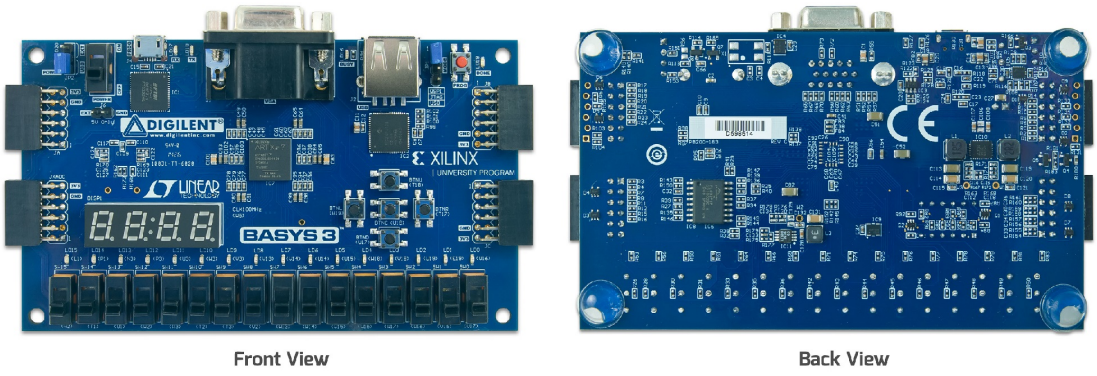


Abbildung 5.1.: Board Vorder- und Rückseite (Quelle: Basys3 Reference Manual)

5.1.1. Versorgung

Das Basys3 wird mit drei verschiedenen Spannungen versorgt. Die Leistungsaufnahme der Logik kann über die Masseleitung und über die 1.0V Versorgung gemessen werden. Bei einer Messung mittels Masse addieren sich die Strome der einzelnen Spannungen und es wird nicht nur die Leistungsaufnahme der Logik gemessen.

Spannung	Funktion	IC	Strom (typisch)
1.0V	Kernspannung	IC10	0.2 - 1.3 A
1.8V	AUX, ADC	IC11	0.05 - 0.15 A
3.3V	I/O, sonstige Periferie	IC10	0.1 - 1,5 A

Tabelle 5.1.: Basys 3: Versorgungsspannungen (Quelle: Basys3 Reference Manual)

Wie in Abbildung 5.2 dargestellt wird zuerst die 1.0V Versorgungsspannung aktiviert. Nachfolgend 1.8V und 3.3V.

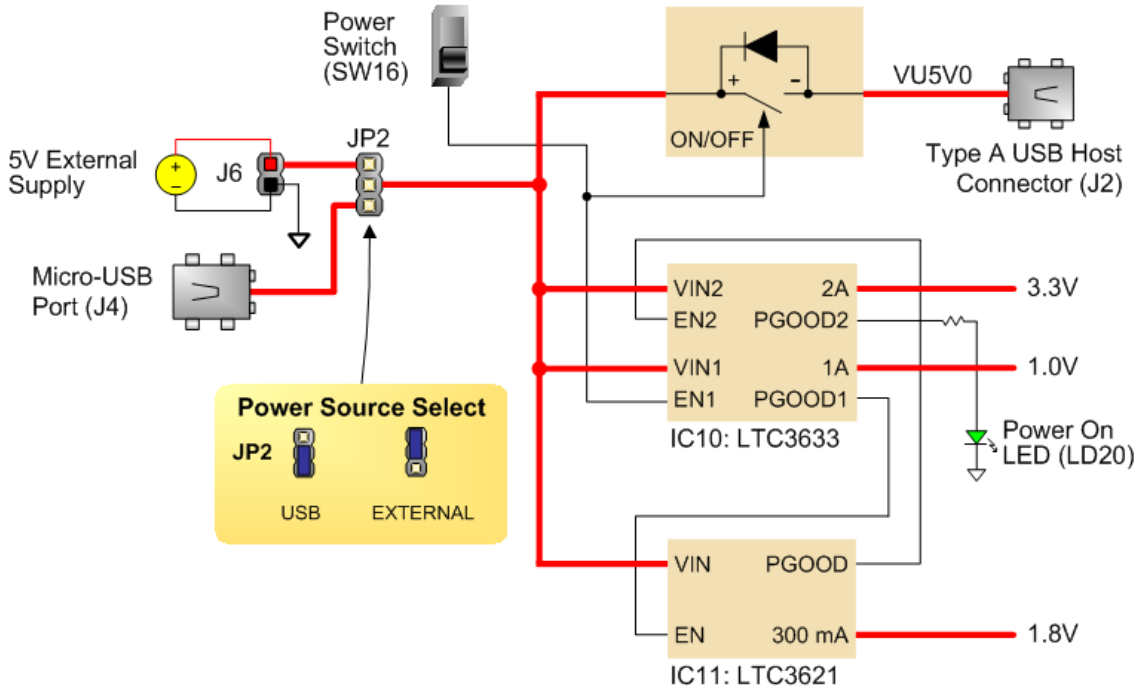


Abbildung 5.2.: Blockschaltbild Versorgung (Quelle: Basys3 Reference Manual)

5.1.2. Leistungsaufnahme und Ressourcen

In Tabelle 5.2 sind der Ressourcenverbrauch der Test-FSM und des AES-Cores aufgeführt. Der Verbrauch des AES-Cores entspricht circa 1200 Slices[1] und ist damit vergleichbar mit kommerziellen Standard-IP-Cores[2] von Firmen wie Cast oder Helion.

	Artix-7 (XC7A35T)	Test-FSM	AES (Gbur)
FF	41600 (100%)	39 (0,09%)	327 (0,79%)
LUT	20800 (100%)	38 (0,18%)	782 (3,76%)
Memory LUT	9600 (100%)	2 (0,02%)	24 (0,25%)
BRAM	50 (100%)		2 (4,00%)

Tabelle 5.2.: Ressourcennutzung der Implementierung

Hierdurch ergibt sich bei einer einzelnen AES Implementierung, gemessen an der 1.0V Versorgung, eine statische Leistungsaufnahme von 45mW. Die dynamische Leistungsaufnahme ist mit 1μW durch nur 8MHz Takt sehr gering. Bei einer höheren Taktrate ist die dynamische Leistungsaufnahme deutlich höher. Dies führt aufgrund von parasitären Kapazitäten zu Glättungen der Messung und verschlechtert die Messung dadurch insgesamt.
Der Xilinx Power Estimator (XPE) schätzt die Leistungsaufnahme auf 67mW (20°C Umgebungstemperatur) bei 99% statischem Verbrauch, hierbei werden sämtliche Verbräuche geschätzt, inklusive I/Os und sonstige Peripherie. Ein übliches Beispieldesign mit DDR-Speicherinterface, DSP-Slices und moderater Ausnutzung kommt auf mehrere Watt.

Da der Artix-7 zur neusten Generation von Xilinx FPGAs gehört ist die Fertigungsgröße und damit auch die Leistungsaufnahme bei selben Takt geringer als bei älteren FPGAs.

Chip	Fertigungsgröße
Coolrunner II	180nm
Spartan III	90nm
Virtex 5	65nm
Spartan 6	45nm
Artix 7	28nm

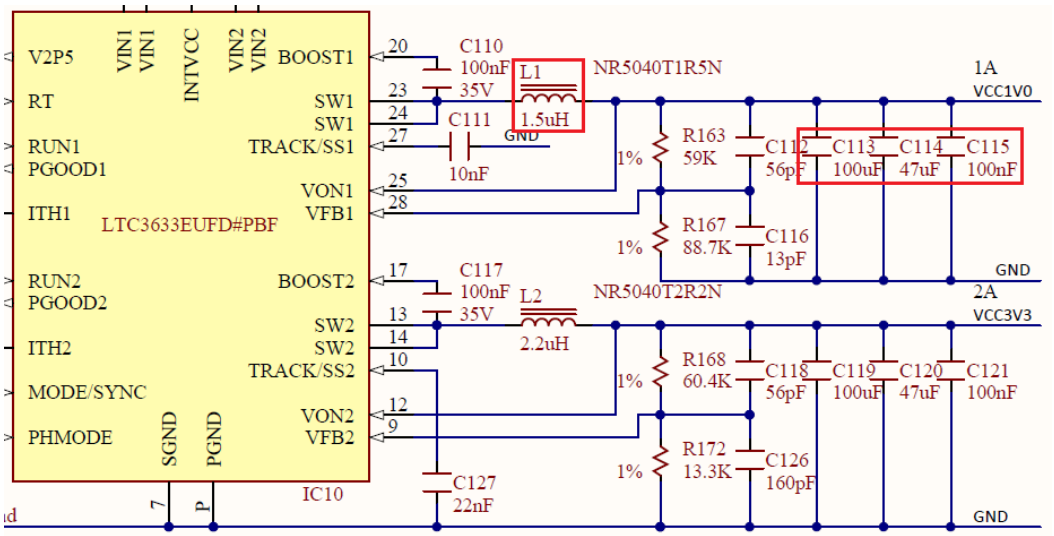
Tabelle 5.3.: Fertigungsgrößen verschiedener Xilinx FPGAs

[1] Die Vergleichbarkeit dieser Angaben ist durch verschiedene Architekturen erschwert.
[2] intellectual property core

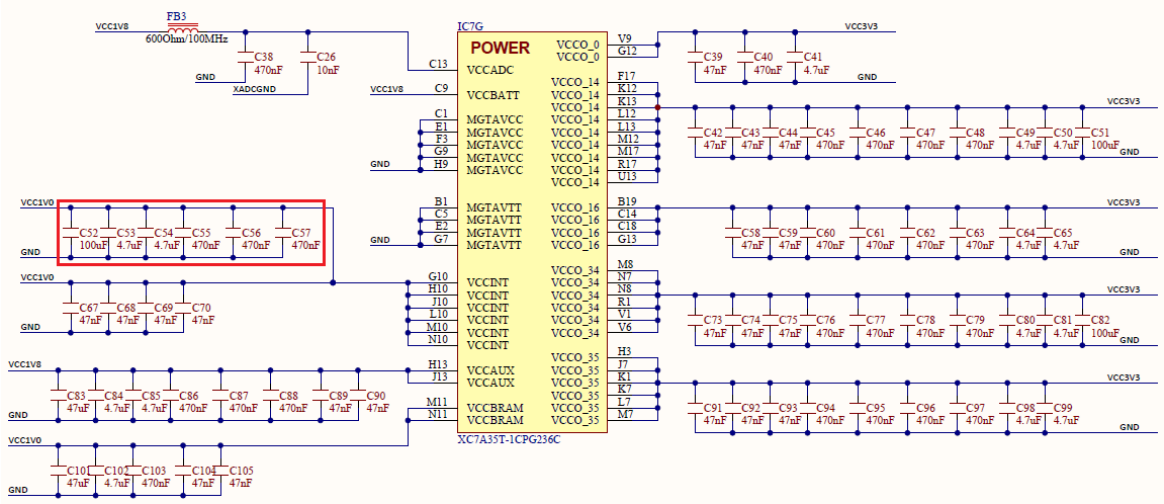
5.2. Versuchsaufbau

5.2.1. Modifikation am Board

Um nur die Leistungsaufnahme der Logik zu messen wird die 1.0V Versorgung aufgetrennt und extern versorgt. In Abbildung 5.3 ist die Versorgung aus dem Schaltplan aufgeführt. Rot markierte Bauteile sind zu entfernen um die Messung nicht zu glätten.



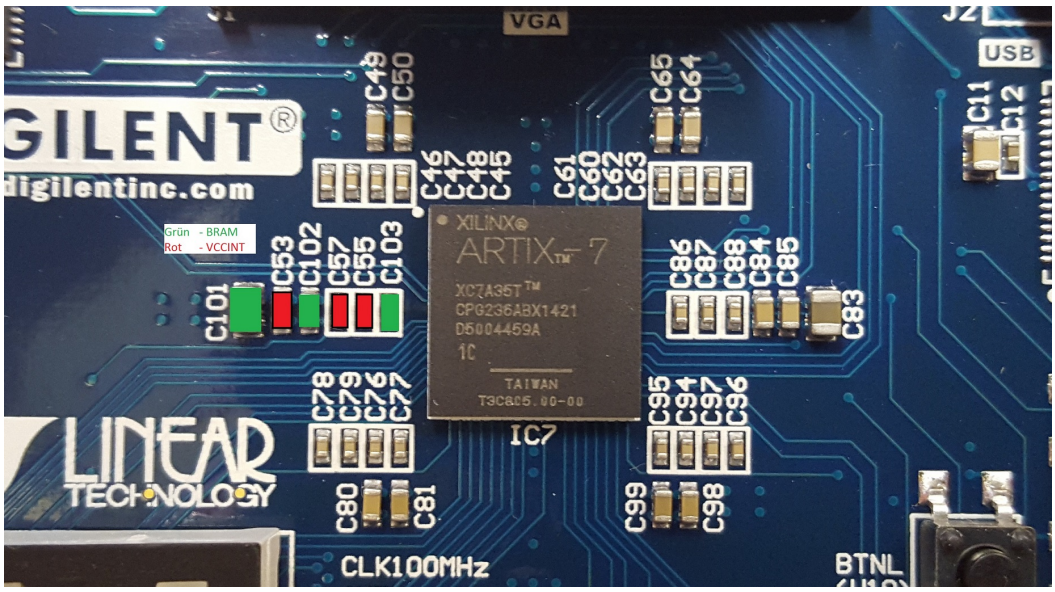
(a) DCDC-Wandler



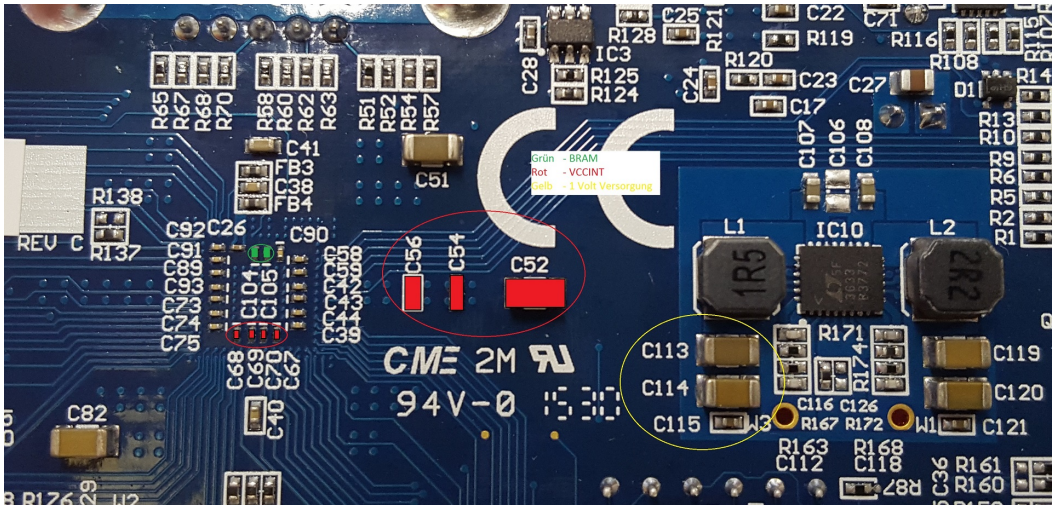
(b) FPGA

Abbildung 5.3.: Versorgung (Schaltplan)

Zur Auftrennung der bisherigen Versorgung muss die Spule L1 entfernt werden. An den Pads von C113 und C114 wird eine Steckverbindung für die Messung angebracht.
Die Kapazitäten vom Block RAM (BRAM) werden bei dieser Arbeit für die Betriebssicherheit nicht entfernt. Die Messung wird dadurch verschlechtert.



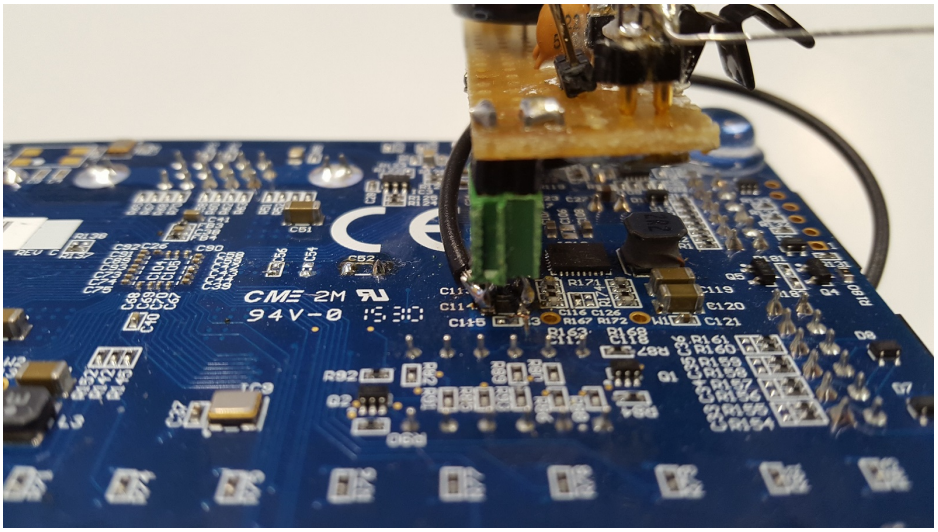
(a) Vorderseite



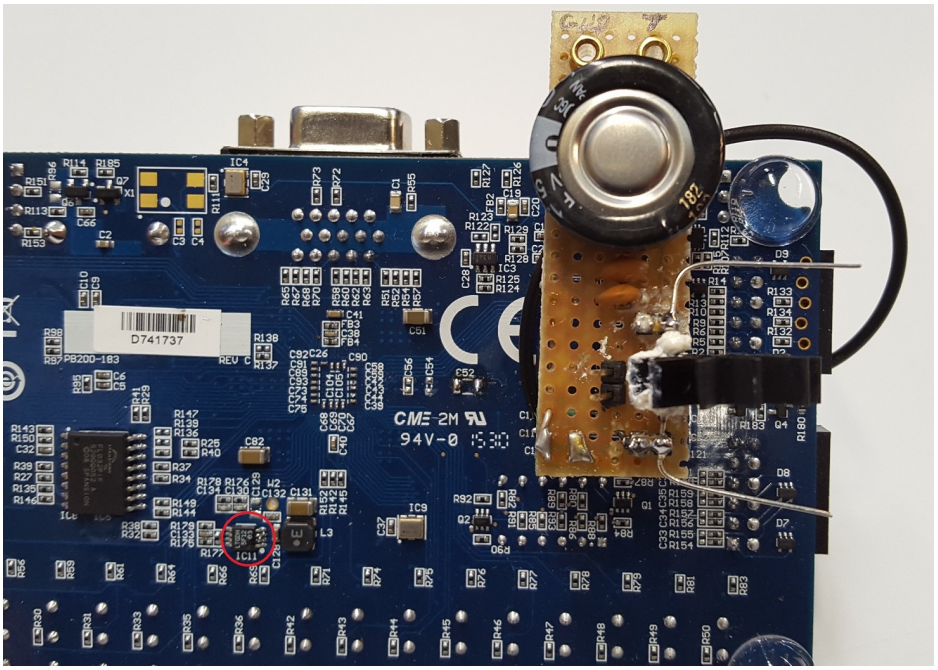
(b) Rückseite

Abbildung 5.4.: Versorgung (Boardlayout)

In Abbildung 5.5 ist das fertig modifizierte Board abgebildet. Beim IC11 ist das Enable-Signal mit VCC verlötet um die Spannung 1.8V unabhängig von 1.0V zu aktivieren. Der Power-Schalter *SW16* schaltet dadurch nur noch den USB-Stecker *J2* und ist unabhängig von der Versorgung.



(a) Steckverbindung der neuen Versorgung



(b) Draufsicht

Abbildung 5.5.: Modifizierte Versorgung (Boardlayout)

Messwiderstand

Für den Messwiderstand sollte ein vergleichsweise hoher Wert für Shunt-Widerstände gewählt werden, jedoch klein genug um die Funktion der Schaltung nicht zu stören. In der Fachliteratur (siehe Mangard u. a. (2007) Kapitel 3.4.2) werden Werte zwischen 1Ohm und 50Ohm, vornehmlich für Mikrocontroller, vorgeschlagen.
Reguläre Shunt-Widerstände, wie sie auf Development-Boards häufig vorkommen sind ungeeignet. Mit Widerstandswerten von unter 1Ohm dienen sie zur Bestimmung der absoluten gemittelten Leistungsaufnahme. Eine Messung der dynamischen Leistungsaufnahme ist zum einen Aufgrund von Störeinflüssen wie EMV-Einstahlung (siehe Kapitel 7.1) und der Positionierung vor DC-DC-Wandlern und Kapazitäten nahezu unmöglich.
Ein weiterer Aspekt ist die Temperaturabhängigkeit des Widerstandes (siehe Kapitel 7.2).
Für die Messungen wird, aufgrund der sehr geringen dynamischen Leistungsaufnahme im Vergleich zur statischen, ein 68Ohm Metallschichtwiderstand verwendet.

5.2.2. Messequipment

Um eine Masse-unabhängige Messung durchzuführen, wird der PC mit Picoscope mittels eines Trenntrafos betrieben. Es ist darauf zu achten das sämtliche externe Verbindungen mit Masseanschluss, zum Beispiel Ethernet-Kabel, unterbrochen werden. Die Programmierung des Board erfolgt über SPI mittels eines Flash Moduls.
Alternativ kann auch ein differenzieller Tastkopf verwendet werden, welcher typischerweise als aktives Bauteil eine geringere Bandbreite aufweist als passive Tastköpfe. Hierdurch kann der PC mit dem Board verbunden sein und direkt über JTAG programmiert werden.
Ein Multimeter misst die Spannung am FPGA um die Betriebsspannung mit einem Labornetzteil einzustellen. Die restliche Versorgung des Boards erfolgt über ein 5.0V Steckernetzteil.
Das Picoscope ist mit einem Kanal am Pin des Triggersignals und mit einem zweiten Kanal am Shuntwiderstand mit Masse angeschlossen.

6. Attacke

6.1. Voruntersuchung des Angriffspunktes

Zur Bestätigung der Theorie des Angriffes wird der Angriffspunkt, die Speicherung eines XOR-Ergebnisses in ein 32bit Register, nachgebaut, wie in Abbildung 6.1 zu sehen.

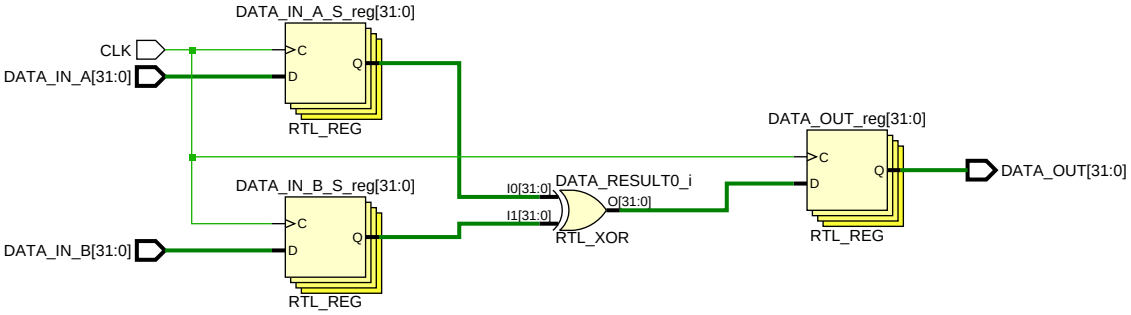


Abbildung 6.1.: Schaltbild des nachgestellten Angriffspunktes

In Abbildung 6.2 wird ein Fall mit dem höchstem Hamming-Gewicht simuliert. Zum Zeitpunkt *27 ns* werden zueinander antivalente Eingangssignale angelegt und zur steigenden Taktflanke bei *35 ns* gespeichert. Zeitlich versetzt liegt das XOR-Ergebnis am Data-Out-Register an und wird bei der nächsten steigenden Taktflanke übernommen.

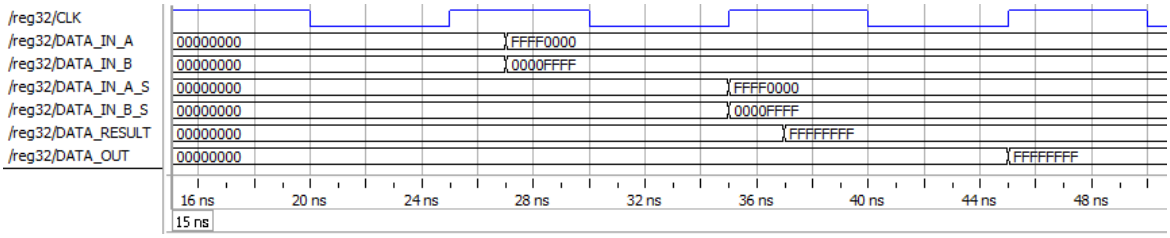


Abbildung 6.2.: Simulation des nachgestellten Angriffspunktes

Die Schaltung ist mehrfach parallel auf dem FPGA implementiert. Hierzu werden sämtliche Optimierungen abgeschaltet, damit die Register nicht zusammengelegt werden. (siehe Kapitel 6.2.2)
Im ersten Versuch wird die Schaltung zweihundert mal implementiert, wodurch es zu $200 \cdot 32 = 6400$ Übergängen von Null auf Eins kommt. Das Ergebnis Data-Out wird ebenfalls in einem Register gespeichert, sodass diese Leistungsaufnahme zweimal hintereinander auftreten sollte.
Um nur die dynamische Leistungsaufnahme zu messen wird die Wechselspannung (AC) am Messwiderstand gemessen. Es werden jeweils 0.8 Millionen Messungen durchgeführt.

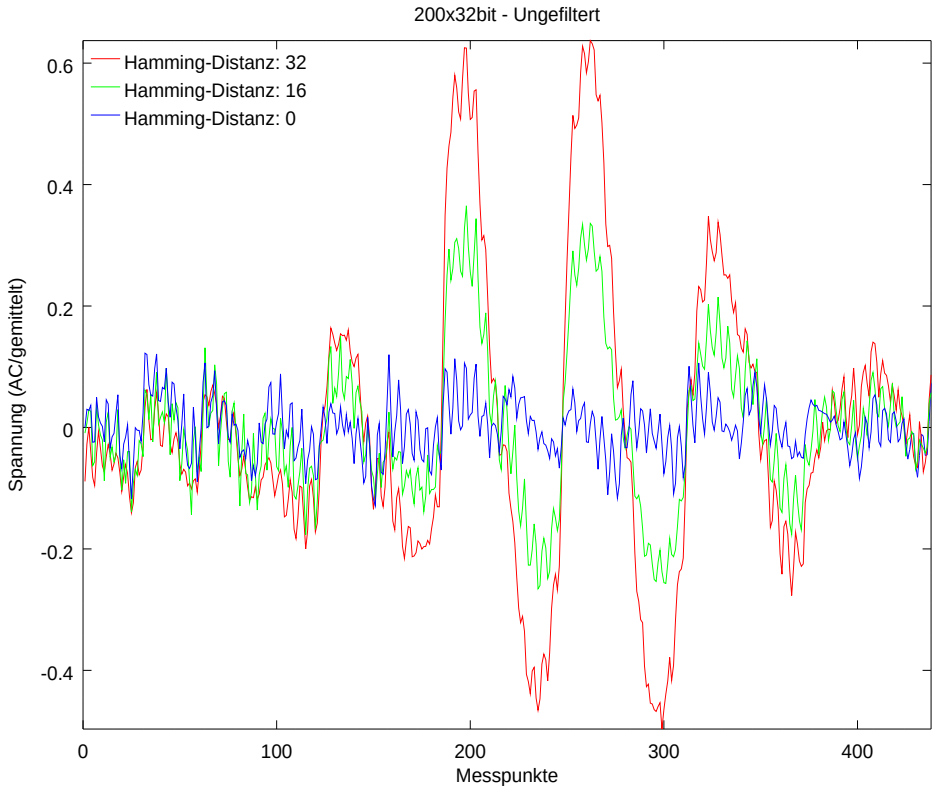


Abbildung 6.3.: Messung des Angriffspunktes (200)

Wie in Abbildung 6.3 ersichtlich, unterscheidet sich die Leistungsaufnahme bei verschiedenen Daten erheblich. Für den Fall der Äquivalenz der Daten gibt es nur eine geringe dynamische Leistungsaufnahme. Mit zunehmender Hamming-Distanz nimmt die Leistungsaufnahme zu und ist auch ohne weitere Bearbeitung der Messdaten sichtbar.

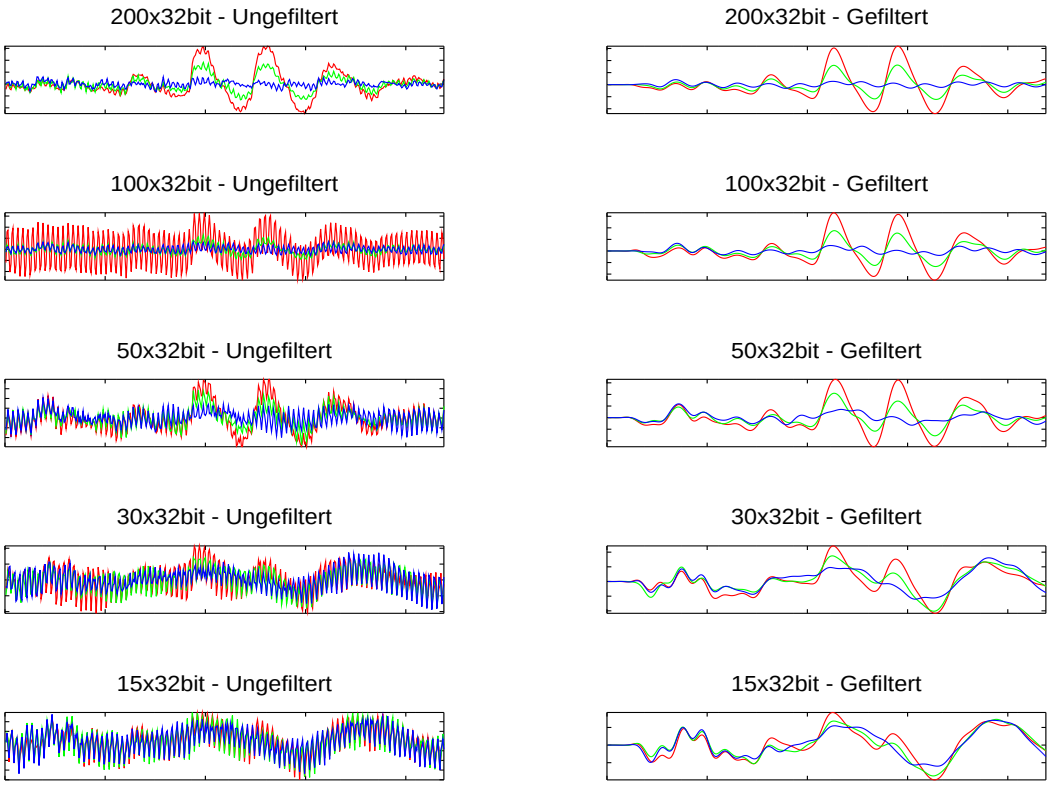


Abbildung 6.4.: Messung des Angriffspunktes mit unterschiedlich vielen Registern
Achsenbeschriftung und Legende entsprechen Abbildung 6.3

Nachfolgend wird die Anzahl der Register reduziert. In Abbildung 6.4 sind die Messdaten sowohl unbearbeitet als auch gefiltert abgebildet.
Die dynamische Leistungsaufnahme reduziert sich mit sinkender Registeranzahl. Dies führt dazu, dass der Signal-Stör-Abstand geringer wird und die Hamming-Distanz nicht mehr eindeutig bestimmbar ist. Mit anwenden eines FIR-Filters[1] können die Störeinflüsse minimiert werden und die Graphen sind weiterhin unterscheidbar.

[1] Diskretes Filter - finite impulse response filter

6.2. Angriff auf den AES

Als Grundlage wird eine öffentlich zugängliche AES-Implementierung auf OpenCores.org von Jerzy Gbur genommen. Diese besitzt weder Abwehrmaßnahmen noch gewollte Schwachstellen.

6.2.1. Angriffspunkt im Algorithmus

Das Ziel dieser Arbeit ist die Bestimmung des Hamming-Abstandes von Daten und Schlüssel. Hierfür wird als Angriffspunkt das Speichern des XOR-Ergebnisses in der Vorrunde gemessen. In Abbildung 6.5 mit rot markiert.

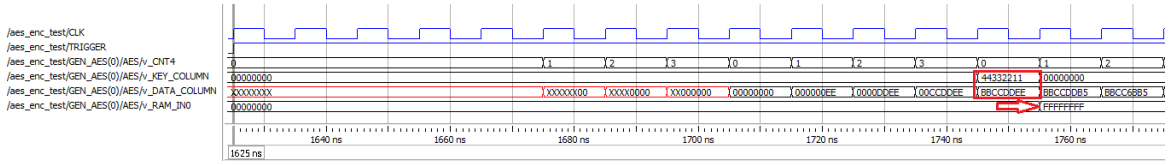


Abbildung 6.5.: Angriffspunkt des AES simuliert in ModelSim

6.2.2. Testumgebung

Für die Testumgebung wird die AES-Implementierung mit einem Gray-codierten Zustandsautomat verbunden. In diesem sind der Schlüssel und verschiedene Daten gespeichert. Durch Stellung von Schaltern am Board kann die Hamming-Distanz gewählt werden. Dies verhindert Einflüsse auf die Leistungsaufnahme durch externe Kommunikation.
Als weitere Vereinfachung sind die ersten 32 Schlüssel- und Datenbits null. Dadurch entspricht das Hamming-Gewicht der Hamming-Distanz.
Die Ausgänge des AES sind offen, so dass das Triggersignal das einzige I/O-Signal der Implementierung ist. Um Optimierung, wie das komplette nicht-implementieren von Schaltungen ohne Ausgänge, zu verhindern werden die in Listing 6.1 aufgeführten Xilinx-Befehle verwendet.

Listing 6.1: *don't touch* Befehle

```
attribute dont_touch: string;
attribute dont_touch of aes_enc_test : entity is "true|yes";
attribute dont_touch of arch: architecture is "yes";
attribute dont_touch of aes_enc: component is "yes";
```

6.2.3. Messung

Für die Messung des Angriffspunktes wird mit 8 Bit bei einer Abtastrate von 500 MHz gemessen. Die Schaltung taktet mit 8 MHz (125ns pro Takt). Aufgrund von Störungen (Kapitel 7.3, Seite 37) beginnt die Messung 8 Takte nach auslösen des Triggersignals und speichert 10 Takte. In Abbildung 6.6 ist die gesamte Messung mit zehn AES Instanzen zu sehen, der Angriffspunkt ist der sechste Takt. Es wird ein 68 Ohm Widerstand verwendet.

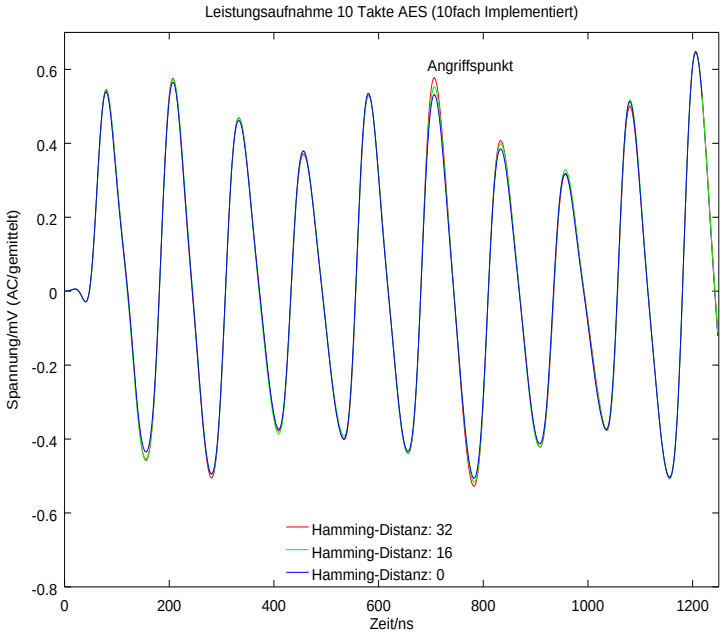


Abbildung 6.6.: Messung um den Angriffspunkt

Abbildung 6.7 zeigt den Angriffspunkt bei jeweils reduzierter Anzahl an parallel implementierten AES Instanzen. Anhand der Spannung ist recht eingängig eine Linearität der dynamischen Leistungsaufnahme ersichtlich.
Da Störungen bei geringeren Nutzsignalamplituden einen höheren Einfluss nehmen musste die Anzahl der Messung, bei weniger Instanzen, erhöht werden.

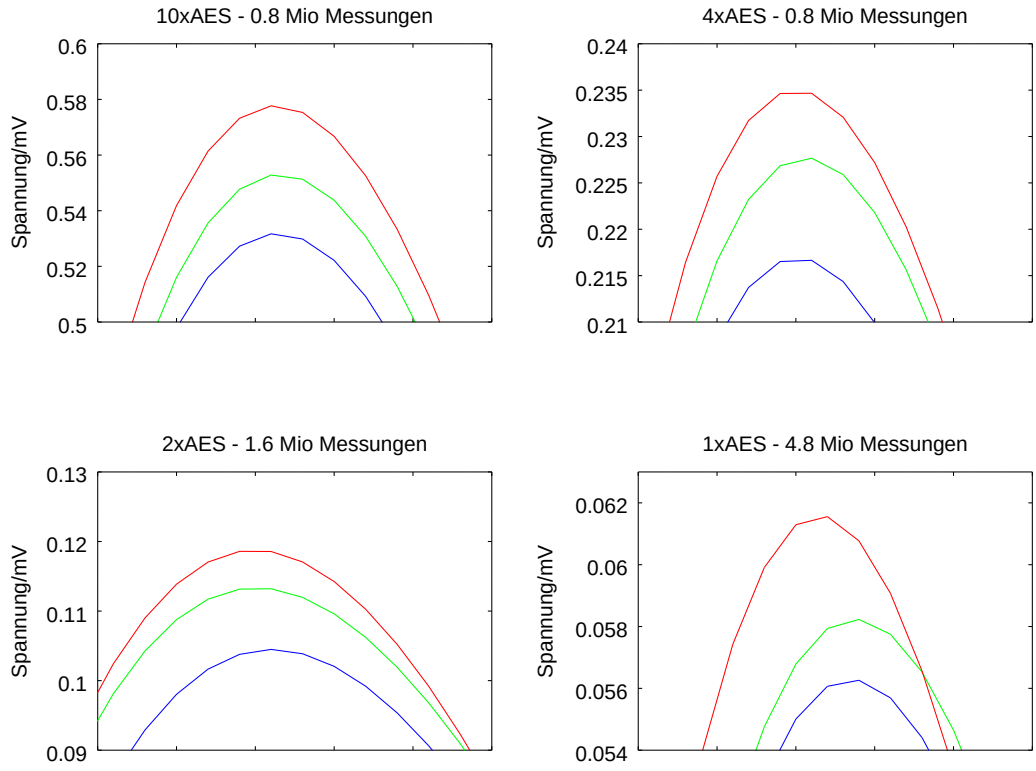


Abbildung 6.7.: Leistungsaufnahme im Angriffspunkt, Legende entspricht Abbildung 6.6

6.3. Auswertung

Eine Verbesserung des Angriffs indem die Hamming-Distanz von nur 16 Bit bestimmt wird ist möglich, indem mittels Zufallszahlengenerator, zum Beispiel LFSR[2], die nicht anzugreifenden 16 Bit bei jeder Messung verändert werden. Dadurch können diese als Rauschen betrachtet werden, welches mittels Mittlung ausgelöscht wird. Hierbei muss die Anzahl der Messung weiter erhöht werden oder der Messaufbau/Messequipment verbessert werden. Durch Bildung des Betrags der Messdaten kann die Hamming-Distanz bei den durchgeführten Messungen abgelesen werden. Dies ist aber nicht hinreichend empirisch überprüft und bedarf weiterer Untersuchungen.

[2]linear feedback shift register

7. Störungen

7.1. EMV-Einstrahlung

Aufgrund des, aus Hochfrequenz Gesichtspunkten, suboptimalen Versuchsaufbaus wird elektromagnetische Strahlung in die Schaltung gestreut. In Abbildung 7.1 wurde das Spektrum bei nicht aktiver Schaltung aufgenommen. Zu sehen sind Einstreuung um 100 MHz vom UKW-Rundfunk und DVB-T, welches in Hamburg von 490 MHz bis 738 MHz gesendet wird. Einstrahlungen vom Mobilfunk sind in dieser Messung nicht enthalten. Diese sind messbar bei UMTS Benutzung in der Nähe des Messaufbaus.
Die Signalstärken sind sehr gering und wirken sich bei einem kleinem Messwiderstand auf die Messung aus.

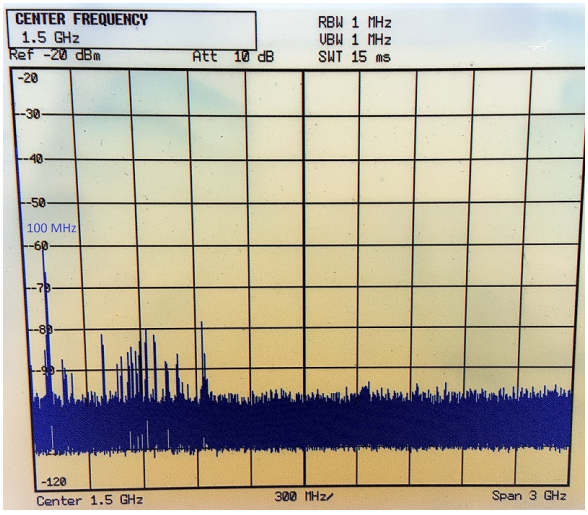


Abbildung 7.1.: Messung des Spektrums am Messwiderstand (Schaltung nicht aktiv)

7.2. Temperatur

Die Raumtemperatur des Bachelorarbeitsplatzes, BT7 Raum 06.83A (Süd-West-Seite), hat zu erheblichen Störungen geführt.

Die Wahl des Widerstandes hat erheblichen Einfluss auf die Temperaturabhängigkeit der Schaltung, so weisen Kohleschicht-Widerstände typischerweise einen Temperaturkoeffizienten von $\pm$250ppm/K auf. Dies führt zu einem Unterschied von 1% pro vier Grad Celsius beziehungsweise Kelvin, beim Versuchsaufbau steigert sich der Widerstand bei steigender Temperatur. Verbessert werden kann dies durch Nutzung von Metallschicht-Widerständen mit einem Temperaturkoeffizienten von $\pm$50ppm/K. Ein noch niedrigerer Koeffizient ist mit SMD-Bauteilen oder Spezialwiderständen möglich.

Ein weiterer Effekt ist die Temperaturabhängigkeit des Halbleiters. Die statische Leistungsaufnahme nimmt bei steigender Temperatur zu. In Rechnung 7.1 ist die temperaturabhängige Steigung zwischen 15 °C und 28 °C aufgeführt. Die Werte ergeben sich aus dem *Xilinx Power Estimator (XPE)*.

$$m = \frac{(71 - 66)mW}{(28 - 15)^\circ C} = 0,39\frac{mW}{^\circ C} \tag{7.1}$$

Dadurch Verstärken sich beide Effekte, es fällt mehr Spannung am Widerstand ab und weniger am Halbleiter, wodurch die dynamische Leistungsaufnahme sinkt. Hierdurch sind Messreihen nicht vergleichbar.

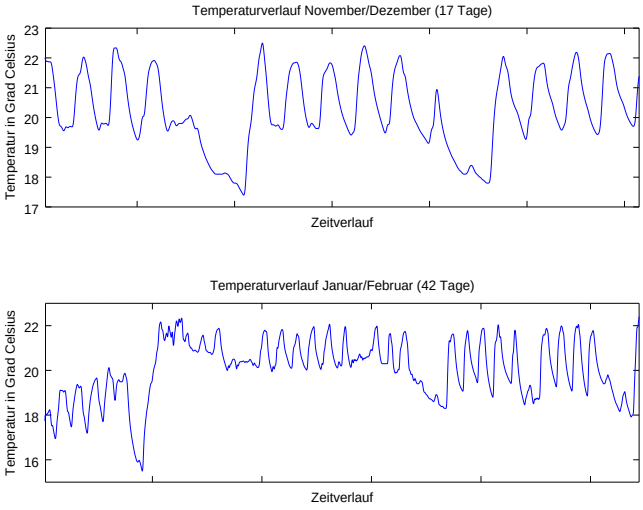


Abbildung 7.2.: Temperaturverlauf BT7, Nord-Ost-Seite (Quelle: Herr Neugebauer)

In Abbildung 7.2 ist die Raumtemperatur des Büros von Herrn Neugebauer dargestellt. Dieser Raum liegt auf der Nord-Ost-Seite des Gebäudes. Da der Bachelorarbeitsraum auf der Süd-West-Seite liegt und damit einer höheren Sonnenbelastung ausgesetzt ist, sind die Einwirkungen auf den Versuchsaufbau größer.
Typischerweise lag der Spannungsunterschied am FPGA bei -5mV bis -10mW von 10 Uhr zu 14 Uhr (Kohleschicht). Auch beim Metallschichtwiderstand werden lange Messung, zum Beispiel die Bestimmung der Hamming-Distanz von 16 Bit, bei diesem Messaufbau verfälscht.

7.3. Triggersignal

Zusätzlich zur 3.3V Versorgung der I/O-Banken müssen die IOBs[1] über die 1.0V Schiene versorgt werden. In Abbildung 7.3 wird die Leistungsaufnahme bei Schalten des Triggersignals (200ns) gemessen.
Diese Störung lässt sich mittels Floorplanning[2] verkleinern. Durch das Einfügen von D-FFs wird die Strecke weiter verkürzt. Die Frequenz der Schaltung hat keinen Einfluss auf die Länge der Schwingung.
Durch frühes Auslösen des Triggersignals ist dieses zum Zeitpunkt der relevanten Operation abgeklungen und hat somit keinen Einfluss auf die Messung.

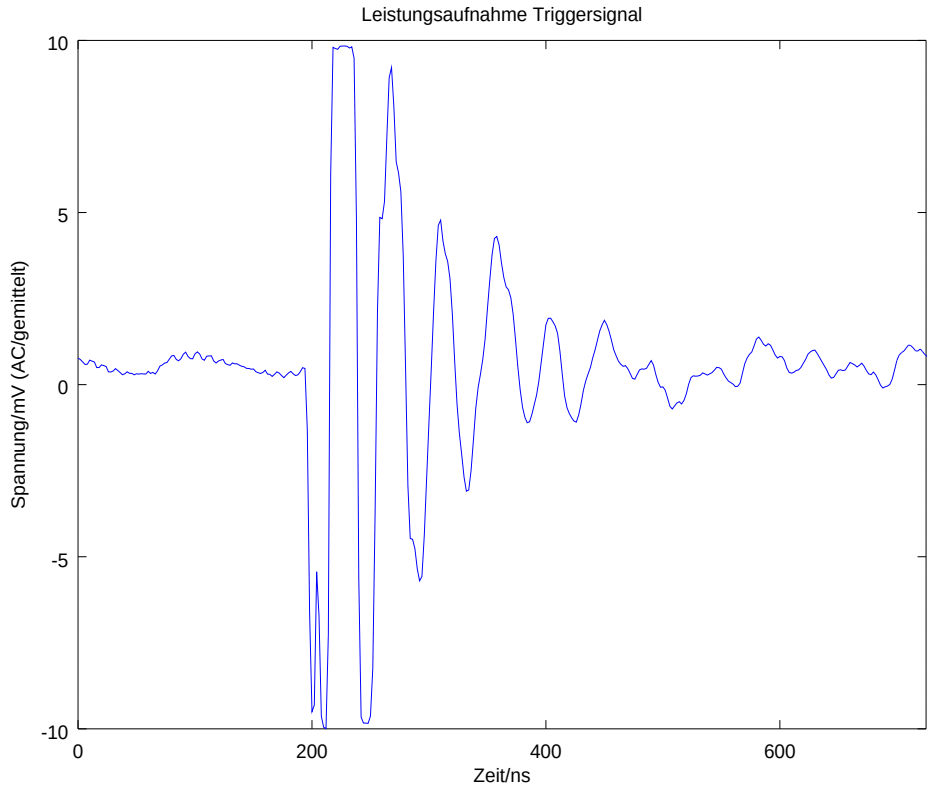


Abbildung 7.3.: Leistungsaufnahme des Triggersignals auf der 1.0V Schiene

[1] Input/Output Blocks
[2] Anordnung der Schaltung für möglichst kurze Signalwege

7.4. Weißes Rauschen

Weißes Rauschen weißt ein konstantes Leistungsdichtespektrum auf. Es tritt in Form von thermischen, elektrischen und weiterem Rauschen auf.
Auf einen Zeitpunkt gesehen sind die Einflüsse normalverteilt, wie in Abbildung 7.4 als Gaußsche Glockenkurve dargestellt. Durch die Mittlung vieler Messungen hebt sich dieses Rauschen auf.

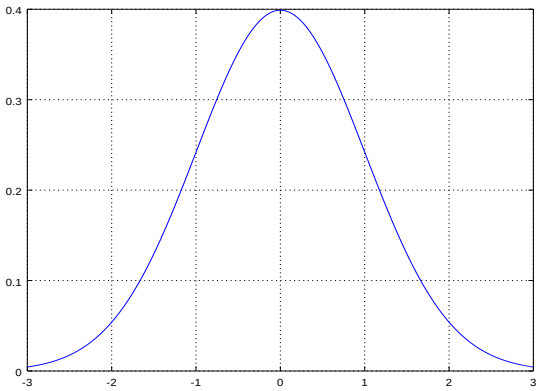


Abbildung 7.4.: Darstellung der Gaußschen Glockenkurve ($\mu = 0$, $\sigma = 1$)

Genauere Untersuchung zu diesem Thema sind im Buch von Mangard, Oswald und Popp (2007) Kapitel 4.2 aufgeführt.

Teil III.

Ausblick

8. Fazit

Diese Bachelorarbeit zeigt, dass es selbst bei modernen, in kleiner Fertigungsgröße hergestellten, Halbleitern möglich ist, die Daten-abhängige dynamische Leistungsaufnahme zu messen. Da hierfür kein spezielles Equipment benötigt wird sind Gegenmaßnahmen auch bei Systemen mit sehr geringer Leistungsaufnahme nötig.
Anfänglich wurde im Zuge dieser Arbeit versucht die Leistungsaufnahme vor den DC-DC-Wandlern zu messen. Dies war komplett erfolglos und es wurde beschlossen das Board zu modifizieren. Da eine Seitenkanalanalyse seitens des Herstellers nicht vorgesehen ist hat die Modifikation einige Zeit in Anspruch genommen. Die Anzahl der Layer und der Verlauf der Leiterbahnen sind nicht dokumentiert und mussten mittels Mikroskop nachempfunden werden.
Auch wurde mit einer einzelnen AES Implementierung begonnen, ohne das bekannt war wie der Verlauf der dynamischen Leistungsausnahme sich darstellt. Hierdurch kam es zu keinem erfolgreichen Ergebnis.
Der Ansatz mit einem simplen Testmodell anzufangen, mit vielen parallel implementierten Registern, war dagegen sehr erfolgreich. Von da an konnte die Schwierigkeit schrittweise erhöht werden. Hierdurch konnten bei jedem Schritt Methoden zur Signalverbesserung geprüft und angewendet werden.

8.1. Verbesserungsmöglichkeiten

8.1.1. Umwelteinflüsse minimieren

Zur Minimierung von Einflüssen von außen kann die Messungen im Faraday-Raum der HAW durchgeführt werden.
Des Weiteren kann eine Kommunikation mit PC und Board implementiert und dadurch die Messungen vermischt werden, sodass bei jeder Messung im Rotationsverfahren der Datensatz wechselt, dadurch sind durch Temperaturschwankungen oder andere Einflüsse alle Messreihen betroffen und dadurch relativ gesehen nicht relevant.Im Gegenzug kann es zu Störungen durch die Kommunikation kommen.
Alternativ kann die Messung auch in einer wärmeren Umgebung wie einem Ofen durchgeführt werden. Hierbei wird es leichter sein eine konstante Temperatur zu halten.

8.1.2. Versuchsaufbau verbessern

Durch einen Messaufbau mit SMD[1] Bauteilen kann die Temperatur- und Störanfälligkeit weiter gesenkt werden. Außerdem können zusätzlich weitere Kapazitäten auf dem Board entfernt werden.

8.1.3. Angriffspunkt wechseln

Als weitere Möglichkeit bietet sich an, den Angriffspunkt zu verlegen. Beim jetzigen Angriff finden zeitgleich zum Speichern des Ergebnisses von *AddRoundKey* noch Aktivitäten in der Test-FSM statt. Außerdem sind die Signalwege von XOR-Verknüpfung zu den dazugehörigen Flip-Flops vergleichsweise kurz. Eine Messung nach *ByteSub* hat den Vorteil das die Daten aus dem BRAM des FPGAs gelesen werden. Aufgrund der Menge von BRAM und Slices für Flip-Flops und Look-Up-Tables führt dies tendenziell zu längeren Signalwegen.
Bei einem Angriff auf einen Mikrocontroller wird meist dieser Angriffspunkt verwendet, da das Laden des Bussystemes eine höhere Leistungsaufnahme aufweist als das Speichern in ein CPU Register.

[1] surface-mount device

9. Anhang

Der Anhang enthält:

- Code
  - AES
  - HammingTest
  - aes_128_192_256_latest.tar.gz
- Dokumente
  - 7_Series_XPE_2016_1_EA1.xlsm
  - AES_Proposal_Rijndael.pdf
  - Basys3_rm.pdf
  - Basys3_sch.pdf
  - DPA_Kocher.pdf
  - fips-197.pdf
- Messung
  - Messdaten
    - AES_SPA
    - HammingTest
    - TemperaturDaten
  - PicoScopeCommunication

Die CD mit Anhang ist bei Prof. Dr. rer. nat. Heike Neumann einsehbar.

Literaturverzeichnis

[Daemen und Rijmen 1998] DAEMEN, Joan ; RIJMEN, Vincent: AES Proposal. (1998)

[Daemen und Rijmen 2002] DAEMEN, Joan ; RIJMEN, Vincent: *The Design of Rijndael*. Springer, 2002. – ISBN 3-540-42580-2

[Ertel 2012] ERTEL, Wolfgang: *Angewandte Kryptographie*. Hanser, 2012. – ISBN 978-3-446-42756-3

[Kurzweil 2008] KURZWEIL, Hans: *Endliche Körper*. Springer, 2008. – ISBN 978-3-540-79597-1

[Mangard u. a. 2007] MANGARD, Stefan ; OSWALD, Elisabeth ; POPP, Thomas: *Power Analysis Attacks*. Springer, 2007. – ISBN 978-0-387-30857-9

[NIST 2001] NIST: FIPS PUBS 197 - Announcing the AES. (2001)

[Paar und Pelzl 2010] PAAR, Christof ; PELZL, Jan: *Understanding Cryptography*. Springer, 2010. – ISBN 978-3-642-04100-6

[Reichardt und Schwarz 2009] REICHARDT, Jürgen ; SCHWARZ, Bernd: *VHDL-Synthese*. Oldenbourg, 2009. – ISBN 978-3-486-58987-0

Tabellenverzeichnis

2.1. Operationen im Körper GF(2) 10
2.2. AES - Rundenanzahl bei verschiedenen Schlüssellängen 11
2.3. AES - Darstellung von 128 Bit als Matrix 11
2.4. AES - S-Box (Verschlüsselung) für das Byte *xy* 13

5.1. Basys 3: Versorgungsspannungen 22
5.2. Ressourcennutzung der Implementierung 23
5.3. Fertigungsgrößen verschiedener Xilinx FPGAs 23

Abbildungsverzeichnis

2.1. AES - Ablaufplan 12
2.2. AES - ShiftRows 15

5.1. Basys3: Board Ansichten (Quelle: Basys3 Reference Manual) 21
5.2. Basys3: Blockschaltbild Versorgung (Quelle: Basys3 Reference Manual) . . . 22
5.3. Basys3: Versorgung (Schaltplan) 24
5.4. Basys3: Versorgung (Boardlayout) 25
5.5. Basys3: Modifizierte Versorgung (Boardlayout) 26

6.1. Schaltbild: 32-Bit Register 28
6.2. ModelSim: Simulation 32-Bit Register 28
6.3. Messung: 200 mal 32-Bit Register 29
6.4. Messung: 32-Bit Register (Zusammenfassend) 30
6.5. ModelSim: AES Angriffspunkt 31
6.6. Messung: AES Angriffspunkt (10 Takte) 32
6.7. Messung: AES Angriffspunkt 33

7.1. Messung: Spektrum der EMV-Einstahlungen 34
7.2. Messung: Temperaturverlauf HAW 36
7.3. Messung: Leistungsaufnahme Triggersignal 37
7.4. Plot: Gaußsche Glockenkurve 38

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 7. März 2016	
Ort, Datum	Unterschrift