COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Électronique*

Par
May Myat Thu

Exploitation of EM Emanations of a SoC-FPGA Bus to Eavesdrop the Internally Processed Images of a Deep Learning Accelerator

Thèse présentée et soutenue à Rennes, le 26 Février 2025
Unité de recherche : Institut d'Electronique et des Technologies du numéRique (IETR - UMR CNRS 6164)

Rapporteurs avant soutenance :

Alexandre BOYER	Maitre de conférences HDR, INSA Toulouse
Cédric KILLIAN	Professeur des universités, Université Jean Monnet, St-Etienne

Composition du Jury :

Président :	Christophe JEGO	Professeur, ENSEIRB-MATMECA, Bordeaux
Examinateurs :	Cédric KILLIAN	Professeur des universités, Université Jean Monnet, St-Etienne
	Alexandre BOYER	Maitre de conférences HDR, INSA Toulouse
	Nele MENTENS	Full Professor, Universiteit Leiden, Pays-Bas
	Christophe JEGO	Professeur, ENSEIRB-MATMECA, Bordeaux
	Gabriel ZAID	Dr. Ingénieur de recherche, Thales ITSEF, Toulouse, France
Dir. de thèse :	Philippe BESNIER	Directeur de Recherche, CNRS, Rennes
Co-dir. de thèse :	Maxime PELCAT	Professeur des universités, INSA Rennes
Enc. de thèse :	Maria MÉNDEZ REAL	Chaire de Professeur Junior, Université de Bretagne Sud, Lorient

Acknowledgement

I would like to extend my deepest gratitude to my jury members, Christophe Jego, Alexandre Boyer, Cédric Kilian, Nele Mentens, and Gabriel Zaïd, for their time, effort, and commitment to evaluating my work. My sincere thanks go to my supervisors, Philippe Besnier, Maxime Pelcat, and Maria Méndez Real. Their guidance, expertise, and thoughtful feedback have been indispensable in shaping this research. I am truly grateful for their mentorship and encouragement throughout this journey. I am also thankful to INSA Rennes and the IETR laboratory for providing the academic environment and resources that made this work possible.

To my family, thank you for your unconditional love and reassurance — your belief in me has been the foundation of all my achievements, throughout this journey and beyond.

I appreciate my seniors, Quentin and Alexis, for the insightful discussions that helped broaden my perspective. I also acknowledge my colleagues, Baptiste, Damien, Jérôme, Juliette, Roua, Loreine, Fatima, and Safouane for their valuable support and collaboration during these years.

A special mention goes to Soe Nyi, Ye Pyae, and Reo, whose moral support, especially in the final stages of my PhD, meant more than words can express. To Chara, Marieme, Nwe, May Thu, Joshua, and Allen — your kindness, compassion, and shared laughter brought much-needed relief during the most intense periods of study.

Tamar, my dearest friend, you are one of the blessings this PhD journey has given me. Your optimism and encouragement lifted me in times of doubt, and for that, I will forever be grateful.

Finally, heartfelt thanks to Pierro — for standing by me through every moment of uncertainty. Your patience, comforting presence, and unwavering belief in me have been my anchor during this thesis. I thank you for your greatest source of love and support.

To everyone who has contributed to this thesis in ways both big and small — **thank you**.

List of Acronyms

AES	Advanced Encryption Standard
AI	Artificial Intelligence
BNN	Binarized Neural Network
CEMA	Correlation Electromagnetic Analysis
CIFAR	Canadian Institute For Advanced Research
CMOS	Complementary Metal-oxide-semiconductor
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CPA	Correlation Power Analysis
CRT	Cathode-ray Tubes
DAQ	Data Acquisition
DDR	Double Date Rate
DEMA	Differential Electromagnetic Analysis
DES	Data Encryption Standard
DL	Deep Learning
DLL	Delay-Locked Loop
DMA	Direct Memory Access
DNN	Deep Neural Network
DPA	Differential Power Analysis
DSS	Digital Signature Standard
DVI	Digital Visual Interface
ECDSA	Elliptic Curve Digital Signature Algorithm
EM	electromagnetic
EMC	Electromagnetic Compatibility

FPGA Field Programmable Gate Array

GPIO General-Purpose Input/Output

GPU Graphic Processing Unit

HDL Hardware Description Language

HD Hamming Distance

HDMI High Definition Multimedia Interface

HBIR Hysteresis Binary Image Reconstruction

HLS High-Level Synthesis

HW hardware

IC Integrated Circuit

IoT Internet of Things

ISP Image Signal Processor

LCD Liquid Crystal Display

LNA Low Noise Amplifier

LSB Least Significant Bit

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multilayer Perceptron

MNIST Modified National Institute of Standards and Technology

MIPI Mobile Industry Processor Interface

MSB Most Significant Bit

MSSIM Mean Structural Similarity Index Measure

NN Neural Network

NSA National Security Agency

OLED Organic Light-Emitting Diode

PCB Printed Circuit Board

PE Processing Elements

PL Programmable Logic

POI	Point of Interest
PS	Processing System
PSNR	Peak Signal-to-Noise Ratio
QNN	Quantized Neural Network
ReLU	Rectified Linear Unit
RSA	Rivest, Shamir, Adleman
SCA	Side-Channel Attack
SDR	Software Defined Radio
SEMA	Simple Electromagnetic Analysis
SIR	Super-Bits Image Reconstruction
SNR	Signal-to-Noise ratio
SoC	System-on-Chip
SPA	Simple Power Analysis
SRAM	Static Random Access Memory
SQL	Structured Query Language
SVHN	Street View House Numbers
SVM	Support Vector Machine
SW	software
TPU	Tensor Processing Unit
USB	Universal Serial Bus
VDU	Video Display Unit
VGA	Video Graphics Array

List of Figures

1 La précision de reconnaissance de la méthode Hysteresis Binary Image Reconstruction (HBIR) (en rouge) par rapport à la précision du jeu de données original Modified National Institute of Standards and Technology (MNIST) (en vert). 20

2 La précision de reconnaissance de la deuxième méthode (en rouge) par rapport à la première (en jaune) ainsi que la précision des images Fashion-MNIST originales (en vert). 21

3 Analyse électromagnétique du circuit imprimé : cartographie des zones de fuite et positionnement optimal de la sonde de mesure. 22

4 Le montage expérimental pour la collecte des traces électromagnétiques à partir du System-on-Chip (SoC)-Field Programmable Gate Array (FPGA) 23

1.1 Architecture of a SoC-FPGA, illustrating the integration of a Processing System (PS), Programmable Logic (PL), and the bus system. 30

2.1 The taxonomy of hardware attacks. 39

2.2 An eavesdropper illegally intercepts the information transmitted over the communication channel between two devices, a sender and a receiver. . . . 42

2.3 An illustration of basic mechanism behind a Side-Channel Attack 45

2.4 An overview of the working principle of a Side-Channel Attack (SCA) on a cryptographic algorithm. 46

2.5 A block diagram depicting the process of a SCA on a Neural Network (NN). 50

3.1 Organization of Chapter 3 58

3.2 Charge and discharge of a Complementary Metal-oxide-semiconductor (CMOS) Inverter 63

3.3 Measuring principles of Langer E-10 probe 66

3.4 Curve for frequency response of Langer E-10 across its defined bandwidth . 66

3.5 Curve for E-field correction (conversion of induced voltage at the probe terminal to E-field value) of Langer E-10 across its defined bandwidth. . . 66

3.6 Calibration data provided by manufacturer for Langer E-10 66
3.7 Measuring principles (detectable H-field) for Langer RF R-32 67
3.8 Waveform for frequency response of Langer RF R-32 within its eligible frequency range 67
3.9 Waveform for H-field correction (conversion of induced voltage at the probe terminal to actual H-field value) for Langer RF R-32 within its eligible frequency range 67
3.10 Waveform for current correction for Langer RF R-32 between within its eligible frequency range 67
3.11 Calibration data provided by manufacturer for Langer RF R-32 67
3.12 (a) Custom-made near-field probe developed in lab for near-field measurements. (b) Schematic diagram of the probe design, showing its dimensions. 68
3.13 Experimental setup for the measurement of probe response. 68
3.14 Effect of probe orientation: (a) Normal orientation(0°) with positive peak response representing rising edge of clock, (b) Inverted orientation (180°) resulting in negative peak response for the same rising edge of the clock. . 69
3.15 Comparison of square wave clock signals created by wave generator and its probe outputs with varying rise and fall times 71
3.16 Experimental test bench employed for electromagnetic (EM) signal acquisition and mapping. 72
3.17 Location of H-field probe on SoC surface and the Low Noise Amplifier (LNA) used for capturing EM emanations. 73
3.18 (a) Surface of the SoC-FPGA under investigation. (b) Schematic representation of the magnetic field component H_y, generated by current I flowing along the x-direction. (c) Schematic representation of the magnetic field component H_x, generated by current I flowing along the y-direction. 74
3.19 E-field cartography of the SoC-FPGA using Langer E-10. This cartography is mapped by taking time-domain traces at each (x,y) coordinate, where the peak values of the signals are averaged. The averaged peak values are then plotted to create the EM cartography. 75
3.20 Cartography (in both H_x and H_y direction) based on the average of maximum voltage peaks of the EM trace during data transmission on bus SoC-FPGA bus. 76
3.21 H-field cartography of the SoC-FPGA for H_x and H_y using customized probe. 77

3.22 Bit arrangements for 5 controlled cases. $x_1, x_2, \ldots, x_{16}$ on the first line represent the bit position on 32-bit sequence, separated into 4 groups, each consisting of 8-bits. Second to fifth line show how each case put active bits on their sequence. 77
3.23 Frequency of the bus is $525MHz$ and the period is $1.9ns$. It is clearly seen that the first half of time instance $t_{\mathrm{d}} = 0.95ns$ is produced by the EM emanation of the first 16-bits (two bytes) of each bus transfer while the second half $t_{\mathrm{d}+1}$ represents the following 16-bits. 78
3.24 Basic working principles of Software Defined Radio (SDR) and Double Date Rate (DDR) operation. SDR transmits data only on the rising edge of the clock signal, while DDR transmits data on both the rising and falling edges, effectively doubling the data transfer rate. 79
3.25 Voltage levels of EM traces for bit arrangement 1,2,3 and 4 with active states assigned to different sections of the bus wires while inducing the same Hamming Distance (HD). 81
3.26 Observation of difference in level of EM emanations with respect to Most Significant Bit (MSB) and Least Significant Bit (LSB) using special grouping of bits. 83
3.27 Hypothesis of position of bus wires inside SoC-FPGA 84
3.28 Experimental setup for validating the bus wire hypothesis. The probe is systematically moved along the x-axis, from (x,y,z) to (x',y,z) $0 < x < x'$, where $x' = 2.6$, covering the approximate area of bus wires to observe how the distance between the probe loop and individual wires affects the leakage. 85
3.29 Resulting voltage peaks of EM traces plotted for each probe positions along the x-axis ($0 < x < x'$, where $x' = 2.6$) The observed variation in leakage reflects the changing proximity of the probe loop to specific bus wires, with higher peaks corresponding to regions where the probe is closer to active wires. Prior to 160 *ns*, the bus remained low activity, carrying only bit '0's. At 160 *ns*, bit-arrangement groups are transferred on the bus, triggering transitions between '0' and '1' bits. After this period of activity, the bus returned to its low-activity state, transmitting '0' bits once again. . 86
3.30 Comparison of V_{peak} (mV) for BA-5 and BA-6 with respect to x-axis coordinates (mm). 88

4.1 **Left**: Four calibration images sent on the internal bus of SoC-FPGA replica during pre-attack phase (a) Plain Black, (b) Plain White, (c) Black & White Rows and (d) Checker Print. **Right**: Corresponding EM traces generated in response to each calibration image are labeled (i)-(iv) respectively. . . . 96
4.2 Representation of the difference in the EM signature between when the bus is idle (no data transfer) and when the bus initiates the input pixel transfer for inference inside accelerator. 97
4.3 (a) Representation of the EM signal, highlighting the pre-amble, data trace, and post-amble sections respectively. (b) & (c) Zoomed-in views of the pre-amble and post-amble parts of the main EM signal shown in (a). 98
4.4 Linear correlation between trace and preamble: It illustrates the correlation coefficient as a function of lag (sample index). The distinct peak in the time domain indicates a strong match between the trace and preamble at that index, highlighting the temporal alignment. The surrounding noise reflects minimal correlation at other lags, underscoring the uniqueness of the detected peak. 100
4.5 Correlation-based detection of True Start and True End Times for data traces of each MNIST digit. **Note**: In the figure, the terms "Start Time" and "End Time" are used for brevity. These correspond to the "True Start Time" and "True End Time" as defined in section 4.4.3. 101
4.6 The experimental setup of EM trace collection from SoC-FPGA (Pynq-Z1 as target) 103
4.7 Gaussian distribution curve using EM emanation, produced by the bus transmission of plain black image as an input to the inference. 104
4.8 Voltage vs. HD chart, showcasing the steady linear relation between the two elements. 105
4.9 Overview of real-time attack 107
4.10 Randomly chosen examples of retrieved MNIST images with HBIR 109
4.11 Recognition accuracy of HBIR method (in red) with respect to the accuracy of original MNIST dataset (in green). 110
4.12 Targeted victim system, HBIR attack, and validation process of retrieved images with a NN classifier. 111

5.1 Traditional (sequential) grouping. 115
5.2 Super-Bits of 8 grouping 116

5.3 Super-Bits of 4 grouping 117

5.4 Creation of a database template by the adversary involves sending special groups of bits on the SoC bus (of the hardware (HW) replica) to induce specific HD values. The corresponding EM traces are collected to construct the database template. 119

5.5 EM signals corresponding to HD values in the database template for Super-Bits of 8 120

5.6 EM signals corresponding to HD values in the database template for Super-Bits of 4. 122

5.7 An overview of the exploitation phase, including its mandatory stages: preparation, correlation, and reconstruction. 123

5.8 Time shift vs transition plot to analyze the delay in peaks before interpolation. 125

5.9 Alignment of real peaks to the theoretical window **before** interpolation. . . 126

5.10 Alignment of real peaks to the theoretical window **after** interpolation. . . . 128

5.11 Time shift vs transition plot to analyze the delay in peaks after interpolation 129

5.12 HD images for Super-Bits of 4 and 8 when MNIST 0, 1, 2, 3, and 4 images are sent as input. Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4. 133

5.13 HD images for Super-Bits of 4 and 8 when MNIST 5, 6, 7, 8 and 9 images are sent as input.Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4. 134

5.14 HD images for Super-Bits of 4 and 8 when Fashion-MNIST 0, 1, 2, 3 and 4 images are sent as input. Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4. 135

5.15 HD images for Super-Bits of 4 and 8 when Fashion-MNIST 5, 6, 7, 8 and 9 images are sent as input. Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4. 136

5.16 Comparison of original input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8. 138

5.17 Comparison of original input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8. 139
5.18 Comparison of orginal input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for Fashion-MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8. 140
5.19 Comparison of orginal input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for Fashion-MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8. 141
5.20 Recognition accuracy of Super-Bits Image Reconstruction (SIR) method (in red) with respect to the HBIR (in yellow) and the accuracy of original Fashion-MNIST images (in green). This accuracy calculation is based on 1000 Fashion-MNIST images from test dataset. 146
5.21 Overall process of how input image data (transferred on the bus inside victim's hardware system) is reversed by using SIR. 149

List of Tables

3.1 Different of bit arrangements on 16-bits sequence and its configuration . . 80
3.2 V_{peak} (mV) for BA-5 and BA-6 with respect to x-axis values. 87

5.1 Possible HD values and combinations for Super-Bits of 8 grouping. The numeric values in the first column represent the HD values. If a number is accompanied by a letter (e.g., 8a, 8b), it indicates possible alternatives to that specific HD value, reflecting different bit transition patterns 119
5.2 Possible HD values and combinations for Super-Bits of 4 grouping. Similar to Table 5.1 the numeric values in the first column represent the HD values, meanwhile, its accompanied letter tells possible alternatives to that specific HD value 121
5.3 Average of metric values on 10,000 MNIST test images 143
5.4 Average of metric values on 10,000 Fashion-MNIST test images 143
5.5 MNIST Performance Metrics 144
5.6 Fashion-MNIST Performance Metrics 145
5.7 Comparison of arbitrarily selected original labels with predicted labels from SB-4, SB-8, and HBIR models 147

Table of Contents

Résumé en Français **19**

1 Introduction **25**
1.1 Security: Software vs. Hardware . . . 26
1.2 Neural Networks . . . 27
1.3 Hardware Accelerators . . . 29
1.4 Motivation of this thesis . . . 29
1.5 Contributions . . . 31
1.6 Dissertation Organization . . . 33
1.7 Chapter Conclusion . . . 34

2 Related Work **37**
2.1 Taxonomy of Hardware Attacks . . . 39
2.2 Eavesdropping Attacks . . . 41
2.2.1 Eavesdropping of Power Traces . . . 42
2.2.2 Eavesdropping of EM Emanations . . . 43
2.3 Side-Channel Attacks . . . 45
2.3.1 SCAs on Cryptosystems . . . 46
2.3.2 SCAs on Neural Networks . . . 49
2.3.3 Cryptosystems and Neural Networks: Challenges in Information Recovery . . . 54
2.4 Advancements Beyond the State-of-the-Art . . . 55

3 Characterization of NF- Emanations of a SoC **57**
3.1 Notions about EM Emanations and their Sensing in Near-Field . . . 59
3.1.1 Origins of Electromagnetic Emanations . . . 59
3.1.2 Induction Effect of Probe . . . 60
3.2 EM and modern ICs . . . 62
3.3 Probe Characterization . . . 64
3.3.1 Transfer Function . . . 64

3.4 SoC-FPGAs Emanations . . . 71
3.4.1 Localisation of EM Leakage . . . 71
3.4.2 Cartography based on Time-domain Signal . . . 74
3.5 Deducing On-chip Bus from EM Emanations . . . 76
3.5.1 Behaviour of EM Leakage in SoC-FPGA . . . 76
3.5.2 Location of On-chip Bus Wires . . . 80
3.5.3 Hypothesis of Bus Location . . . 82
3.5.4 Validation of Bus Hypothesis . . . 84
3.6 Chapter Conclusion . . . 87

4 Hysteresis Binary Image Reconstruction 89
4.1 Overview of Attack . . . 90
4.1.1 FINN Framework . . . 90
4.1.2 PYNQ-Z1 . . . 92
4.2 Threat Model . . . 93
4.3 Example Use Case of Attack . . . 94
4.4 Pre-attack Phase . . . 94
4.4.1 Calibration Images . . . 95
4.4.2 Pre-amble and Post-amble . . . 97
4.4.3 Detection of True Start Time . . . 98
4.5 Real-time Attack Phase . . . 100
4.5.1 Signal Processing . . . 102
4.5.2 Image Recovery . . . 106
4.6 Results and Evaluation of HBIR Methodology . . . 108
4.6.1 Limitations of HBIR . . . 110
4.7 Chapter Conclusion . . . 110

5 Super-Bits Image Reconstruction 113
5.1 Super-Bits . . . 114
5.1.1 Super-Bits of 8 . . . 115
5.1.2 Super-Bits of 4 . . . 116
5.2 Preliminary Phase . . . 117
5.2.1 Database Template . . . 118
5.3 Exploitation Phase . . . 121
5.4 Preparation Stage . . . 123

5.4.1 Raw EM Trace . . . 123
5.5 Correlation Stage . . . 129
5.6 Reconstruction Stage . . . 137
5.7 Evaluation of Results from Methodology 2 . . . 137
5.7.1 Transition Recovery . . . 137
5.7.2 Limitations of SIR . . . 147
5.8 Comparison Between Methods: HBIR and SIR . . . 147
5.8.1 Limitations . . . 148
5.9 Chapter Conclusion . . . 152

6 Conclusion **153**
6.1 Contributions Summary . . . 154
6.2 Perspectives and Future Work . . . 155

Appendix **159**
6.3 Approximation of Biot-Savart's Law and Hertz Dipole in the Near Field . . 159

Bibliography **161**

Résumé en Français

Les réseaux neuronaux sont aujourd'hui largement utilisés dans divers secteurs, des organisations publiques et entreprises privées jusqu'à l'électronique grand public, tout comme dans les smartphones et les tablettes. Ces réseaux neuronaux fonctionnent au travers de couches cachées complexes et nécessitent généralement des accélérateurs matériels pour répondre à leurs exigences de calcul importantes. Pour répondre à ces exigences, les accélérateurs de réseaux neuronaux basés sur des FPGA sont devenus un choix populaire dans l'actuel état de l'art. Cependant, les vulnérabilités matérielles de tels accélérateurs doivent être soigneusement étudiées afin d'atténuer les menaces de sécurité potentielles et d'empêcher les fuites de données.

La plupart des SoC ne disposent pas de cryptage intégré pour les bus de données entre le système processeur et la logique programmable, ou pour les interfaces de mémoire externe, en raison de contraintes économiques ainsi que de réduction de la consommation d'énergie. Par défaut, les données transmises par ces bus ne sont pas cryptées [1]. En outre, comme la majorité des dispositifs SoC présentent des mécanismes de démarrage sécurisé pour protéger l'intégrité et la confidentialité du processus de démarrage, il est supposé que le cryptage des données sur le bus n'est pas nécessaire tant que la mémoire et les bus sont censés fonctionner dans un environnement de confiance établi par le démarrage sécurisé [2]. Ces bus de données non protégés constituent une faiblesse critique dans le système; c'est pourquoi l'étude de leurs vulnérabilités matérielles est essentielle pour développer des contre-mesures efficaces.

Pendant le transfert des données d'entrée au réseau neuronal pour l'inférence, les bus internes émettent des émanations électromagnétiques, qui peuvent être captées à l'aide d'une sonde magnétique en champ proche. La Fig. 3a illustre la cartographie électromagnétique utilisée pour localiser la zone de fuite principale, ainsi que le positionnement de la sonde permettant une capture efficace des signaux. L'analyse des variations de l'intensité de l'émanation selon la position de la sonde permet de déduire l'acheminement physique des fils de bus. Comme ces bus transfèrent les valeurs des pixels d'entrée pour l'inférence du réseau neuronal, deux nouvelles méthodes sont proposées pour extraire ces valeurs à partir d'une unique trace d'émanation.

Dans cette thèse, nous proposons deux méthodes pour reconstruire des images à partir d'émanations électromagnétiques capturées pendant l'inférence de réseaux neuronaux. La première méthode, appelée « **Reconstruction d'images binaires par hystérésis** », a été testée sur l'ensemble de données standard MNIST. Cette méthode permet d'obtenir une précision de fidélité allant jusqu'à 84% vu dans Fig. 1 pour les images reconstituées. La seconde méthode, appelée « **Reconstruction d'images par Super-Bits** », étend l'applicabilité de notre approche au-delà de l'ensemble de données MNIST, à l'ensemble de données Fashion-MNIST, plus complexe, qui comporte des images d'articles vestimentaires.

Les deux méthodes fonctionnent en deux étapes: Premièrement, elles extraient un résultat intermédiaire des signaux électromagnétiques en analysant le nombre de transitions de bits entre deux cycles d'horloge consécutifs sur le bus.

Pour mesurer la précision de ce résultat intermédiaire, nous utilisons deux métriques:

— La différence au niveau de la distance de Hamming, qui compare les transitions de bits de l'image originale aux transitions obtenues par notre méthode. On obtient une différence moyenne de 0,9, ce qui indique un niveau élevé de similitude.

— Le PSNR, qui mesure à quel point le signal reconstruit est proche de l'image originale. Le PSNR moyen est de 35,72 pour MNIST et de 33,6 pour Fashion-MNIST.

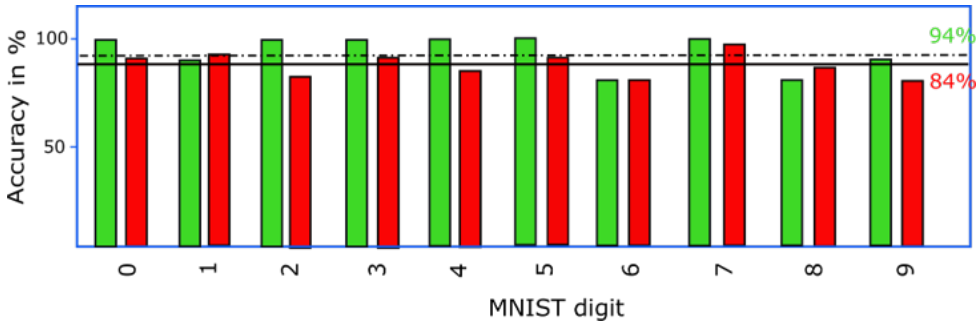


Figure 1 – La précision de reconnaissance de la méthode HBIR (en rouge) par rapport à la précision du jeu de données original MNIST (en vert).

Après avoir obtenu ce résultat intermédiaire, nous reconstruisons les images d'entrée originales qui ont été envoyées au réseau neuronal. Pour évaluer la qualité de ces images reconstituées, nous utilisons un réseau neuronal tiers de classification d'images. Selon la Fig. 2, la reconnaissance des images reconstruites de Fashion-MNIST par la deuxième méthode est de 58%, ce qui démontre que notre méthode permet d'intercepter efficacement les images d'entrée via une fuite électromagnétique.

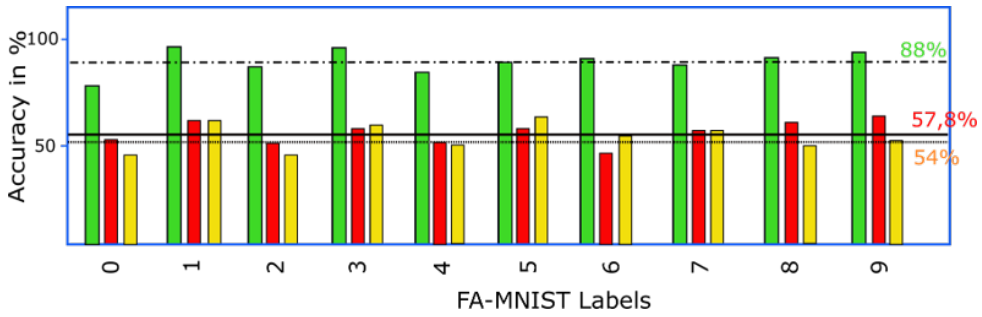
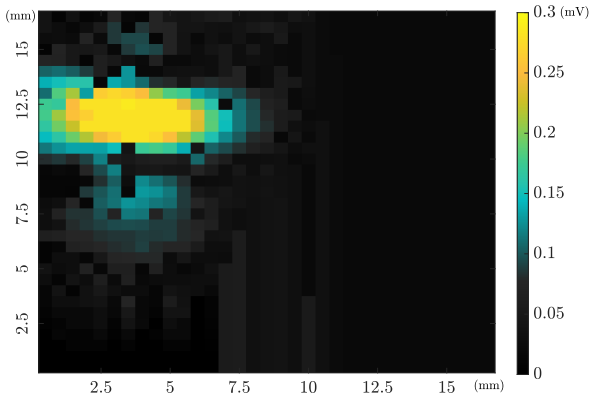


Figure 2 – La précision de reconnaissance de la deuxième méthode (en rouge) par rapport à la première (en jaune) ainsi que la précision des images Fashion-MNIST originales (en vert).

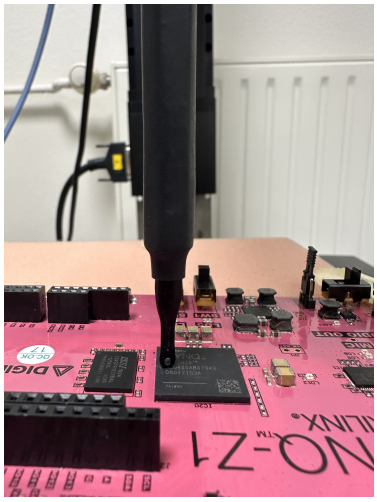
Le reste de cette thèse est organisé comme suit : chaque chapitre développe progressivement l'état de l'art, en passant par le contexte, la méthodologie, les résultats expérimentaux et les conclusions. **Chapitre 1** introduit le contexte général de cette thèse. Il présente les motivations liées à la sécurité des accélérateurs de réseaux neuronaux, en particulier ceux basés sur les FPGA, et expose les problématiques de fuite de données par canaux auxiliaires. Ce chapitre sert de fondement aux méthodologies développées dans les chapitres suivants.

Chapitre 2 présente une revue de la littérature publiée dans le domaine des attaques par canal latéral. Une taxonomie des attaques est proposée pour organiser un vaste ensemble de travaux de manière structurée, afin d'aider les lecteurs à mieux comprendre et catégoriser les recherches existantes. En outre, les attaques hardware sont principalement classées en deux catégories : les attaques par écoute et les attaques par canaux auxiliaires. Cette classification, en conjonction avec la taxonomie, mettent en lumière les lacunes de la littérature existante, que cette thèse vise à aborder. Après avoir présenté l'état de l'art, le chapitre conclut en discutant de la nouveauté de notre travail et de la façon dont celui-ci contribue à combler les lacunes identifiées dans la littérature.

Chapitre 3 explique pas à pas la procédure pour caractériser et capturer les émanations en champ proche électromagnétiques d'un SoC-FPGA. Ce chapitre évoque des principes fondamentaux d'EM et de la nature des émanations EM ainsi que de leur impact sur les circuits intégrés moderne. Cette étape prépare le terrain pour les deux méthodologies d'attaque suivantes. Avant de procéder à l'acquisition régulière des fuites électromagnétiques, il est obligatoire de mesurer la fonction de transfert de la sonde d'intérêt illustré à Fig. 3b. Un signal carré imitant la fréquence de l'horloge du SoC ciblé est introduit dans les fils de cuivre d'un circuit imprimé. La sonde est ensuite mise en place et les



(a) La cartographie électromagnétique réalisée pour l'identification de la zone de fuite maximale.



(b) Positionnement de la sonde de champ magnetique perpendiculairement au plan du circuit imprimé, permettant une mesure efficace des émanations.

Figure 3 – Analyse électromagnétique du circuit imprimé : cartographie des zones de fuite et positionnement optimal de la sonde de mesure.

émanations électromagnétiques provenant de ce fil sont analysées. De plus, des cartographies électromagnétiques sont réalisées pour identifier la zone de fuite maximale sur la surface du SoC-FPGA afin qu'un attaquant puisse se concentrer uniquement sur le point d'intérêt.

Chapitre 4 fournit une description détaillée de la première méthode d'attaque de cette thèse. La méthodologie est basée sur une granularité de quatre pixels, *i.e.* des groupes cibles de quatre pixels (en supposant les données en rapport avec la largeur des données) sont observés à une instance temporelle. La méthode d'exploitation proposée est mise en œuvre dans le dispositif expérimental matériel vu dans Fig. 4 et les résultats obtenus valident l'approche. Ces résultats sont de surcroît réévalués pour vérifier la validité et la fiabilité de ces expériences. Les contributions de ce chapitre ont été publiées dans EMC Europe 2023 [3] et ARES 2023 [4].

Chapitre 5 propose la seconde méthodologie, qui opère à une granularité de deux pixels en manipulant une paire de pixels de chaque groupe de quatre pixels. La méthode présentée dans ce chapitre est conçue pour extraire des informations d'images grayscale plus complexes, caractérisées par des valeurs de pixels diverses. Cette différence de granularité affecte la résolution et la précision de l'attaque. En suivant une approche identique

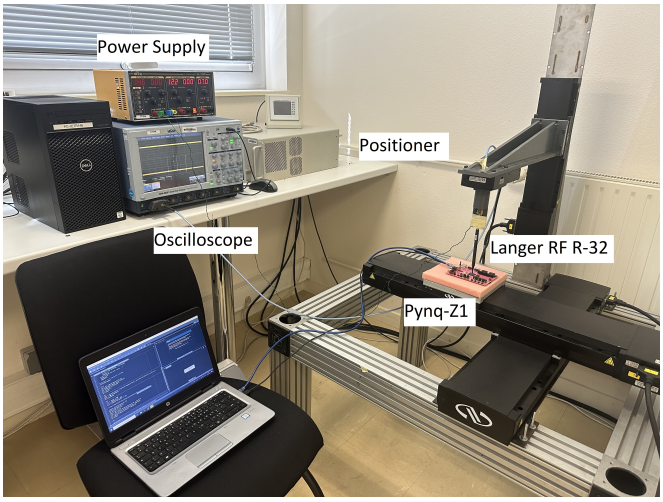


Figure 4 – Le montage expérimental pour la collecte des traces électromagnétiques à partir du SoC-FPGA

à celle du chapitre précédent, les résultats finaux sont évalués et analysés.

Chapitre 6 résume les contributions de la thèse et discute des perspectives. Les travaux futurs possibles sont également explicités, tous sont des sujets ayant pour but d'améliorer la performance de l'attaque et d'ouvrir des voies pour des approches alternatives.

Chapter 1

Introduction

Contents

1.1 Security: Software vs. Hardware 26
1.2 Neural Networks 27
1.3 Hardware Accelerators 29
1.4 Motivation of this thesis 29
1.5 Contributions 31
1.6 Dissertation Organization 33
1.7 Chapter Conclusion 34

The introduction chapter provides the theoretical knowledge necessary for understanding the content of this thesis. The thesis is built upon three fundamental disciplines: Security, Neural Networks, and Hardware Accelerators. Following the discussion of each discipline, the chapter also presents the motivation behind this research and highlights the key contributions of the thesis. The last but not the least, the organization of the dissertation is described to guide the reader through the subsequent chapters.

1.1 Security: Software vs. Hardware

In the current digital era, security has become a top priority for many enterprises, government organizations, and individuals [5]. Pervasive security threats these days are data breaches, phishing, malware, side-channel attacks, denial-of-service attacks [6], etc. In general, this broad spectrum of security-related problems can be categorized into two main areas; software (SW) and HW security.

Software security predominantly involves protecting the vulnerabilities in SW applications such as antivirus, firewalls, and intrusion detection systems. Software is prone to a range of security risks such as buffer overflows, configurations, coding flaws, and other exploitable vulnerabilities. Therefore, the key objective of SW security is to detect viruses, thwart unauthorized data access attempts, and ensure the reliability of SW functions. Software-based vulnerability, such as Structured Query Language (SQL) injection arises from flaws in the logic of the application. In this type of attack, an adversary manipulates input fields to inject malicious SQL queries, gaining unauthorized access to back-end databases and compromising data security. SW security works to maintain data integrity, confidentiality, and availability through encryption, the use of a firewall, access control, and enforcing secure coding practices.

On the other hand, HW security is the process of preventing HW components and devices from data theft and illegal access to maintain data integrity, confidentiality, and availability [7]. Hardware devices encounter various security risks such as tampering [1], reverse engineering, or even the malicious integration of hardware Trojan at the manufacturing phase. The fundamental goal of hardware security is to ensure that the HW components are resistant to attacks, which aim at implementation-level vulnerabilities [8]. Smart cards or cryptographic devices are the classical targets of attackers in which secret keys

1. An unauthorized modification, manipulation, or interference with hardware components to compromise their functionality, integrity, or security.

are stolen by the analysis of side-channels leakage [9]. However, attention is increasingly turning toward more sophisticated devices as targets, which remain under-explored in the literature, highlighting the need to investigate their vulnerabilities. Safety measures for HW protection include HW Encryption Modules [2] [10], Trusted Platform Modules [3] [11], and tamper-resistant Integrated Circuits (ICs). These security mechanisms help to ensure that HW components remain secure even in high-risk environments where they are exposed to physical threats. Although such safety measures exist, they are often costly to implement and introduce significant complexity during the design phase. Moreover, measures like TPMs provide protection for specific functions (e.g., secure boot, cryptographic operations) [12] and may fail to address vulnerabilities outside their intended scope, leaving certain aspects of HW unprotected. This highlights the need to look into the potential vulnerabilities in more complex HW as targets, which remain largely unexplored.

1.2 Neural Networks

Neural Networks originated from the perceptron model in the 1950s [13] and evolved further into multi-layered architectures, fitted to solve even more complex tasks. Current NNs are built upon decades of research and advancements, making use of sizable data and modern computing power. NNs play an extensive role in decision-making processes by mimicking the structure of a human brain [14]. The main goal of Deep Neural Networks (DNNs) is to facilitate systems to learn from training data and to automate decision-making and data generation processes across diverse applications, including image recognition, natural language processing, and predictive analytic [15]. The Multilayer Perceptron (MLP) is considered as the simplest form of NN architecture and, when deepened, it turns a form of complex DNN [16]. In other words, adding extra hidden layers to the MLP increases the depth of the network, transforming it into a DNN. MLPs operate as a feedforward network, made up of an input layer, one or more hidden layers, and an output layer, with each layer fully connected to the next [17].

MLPs consist of multiple computational layers composing a pipeline of processing that generates output labels from input data [18]. Unlike the input and output layers, these intermediate layers do not directly interact with the raw input data or the final

2. A specialized device for encryption, decryption, and key management, offering faster and more secure data protection

3. A small hardware that stores sensitive data like cryptographic keys while supporting system integrity with features such as secure boot and encryption

prediction results. Such computational layers are called hidden layers and contain many neurons, in which each one of them performs a combination of multiplication and addition operations, followed by non-linear activation functions to categorize data and predict outcomes effectively [19]. These layers extract and transform features from the input data, enabling the network to learn complex patterns and relationships. In a MLP, the computation for a single neuron is given by:

$$y = \sigma\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right) \tag{1.1}$$

where y is the output of the neuron, σ is the activation function, w_i is the weight associated with the i-th input, x_i is the i-th input feature to the neuron, b is the bias term added to the weighted sum and n is the number of input features connected to the neuron.

$$\sigma(z) = \max(0, z) \tag{1.2}$$

where: $\sigma(z)$ is the Rectified Linear Unit (ReLU) activation function, z is the input to the activation function, which is the result of the weighted sum and $\max(0, z)$ represents the ReLU operation, outputting z if $z > 0$, otherwise 0. This simplicity makes ReLU [20] a popular choice amongst others (sigmoid, tanh, etc). Besides, the non-saturating nature of ReLU helps mitigate the vanishing gradient issue, which has been proven to enhance both the speed of convergence and the accuracy of DNNs. NNs operate in two distinct phases: training and inference phase. During the training, the model learns from a training dataset, where several labeled samples are used to repeatedly adjust the weights through backward propagation to reduce prediction error [21]. After the completion of the training phase, the inference involves using the trained model to make classifications on new, unseen data. Throughout this work, the MLP serves a dual purpose. Firstly, we target a HW implementation of an MLP, developing an attack strategy specifically designed to exploit its vulnerabilities. Secondly, to assess the effectiveness of our proposed attacks, we also utilize another MLP to classify the resulting images, providing the accuracy of the attacks.

Notable DNNs for image classification include LeNet [22], AlexNet [23], MobileNet [24], SpinalNet [25], VGG [26] etc. Each one of the diverse architectures demonstrates how NNs are adapted for certain tasks. In the case of more complex architectures like Vision Transformer [27], EfficientNet [28], InceptionV3 [29] and ResNet [30] which consist of 20+ layers, they usually demand high computational power. Given these substantial requirements of DNNs, especially for deep architectures, optimized HWs like Graphic Processing

Units (GPUs) or FPGAs are suitable for efficient deployment [31].

1.3 Hardware Accelerators

An accelerator is a distinct architectural component designed with a different set of objectives compared to the based processor and is tailored specifically to meet the unique requirements of a particular applications [32]. However, it is important to recognize that HW accelerator is not the replacement of a traditional Central Processing Unit (CPU) [33]. While the latter is optimized for various general-purpose workloads, the accelerators are dedicated to the execution of computationally intensive operations, such as data encryption, NN inference, etc. HW accelerators are widely adopted in applications namely, cryptography, image and video processing, data analysis, and, most prominently, NNs [18].

HW accelerators are particularly advantageous in the implementation of NNs as they have capabilities of processing large amounts of data in parallel [34], while keeping the costs low. They also ensures the optimal performance in terms of power consumption [35]. A few popular examples of HW accelerators are GPUs [36], Tensor Processing Units (TPUs), and SoC-FPGAs. Out of these widely used accelerators, this thesis particularly focuses on SoC-FPGAs because of their unique advantages. SoC-FPGAs integrate PL with processor, *i.e.* PS, onto a single chip as seen in Fig. 1.1. This gives SoC-FPGAs an upper hand compared to traditional FPGAs which lack general processing unit and are less versatile. On-chip communication between PS and PL is established by a series of complex wires and interconnects.

Given the important role of SoC-FPGAs accelerators, it is crucial to investigate their potential HW-level vulnerabilities. Since these accelerators involve handling sensitive data, any security risk can lead to severe consequences, such as system compromise or data leakage [37]. This underscores how significant it is to examine and address the security concerns of HW accelerators in modern computing systems.

1.4 Motivation of this thesis

Most SoCs do not provide built-in encryption for data buses between PS and PL, or for external memory interfaces due to cost constraints and reduction of power consumption. By default, data transmitted over these buses is not encrypted [1]. In addition, as the

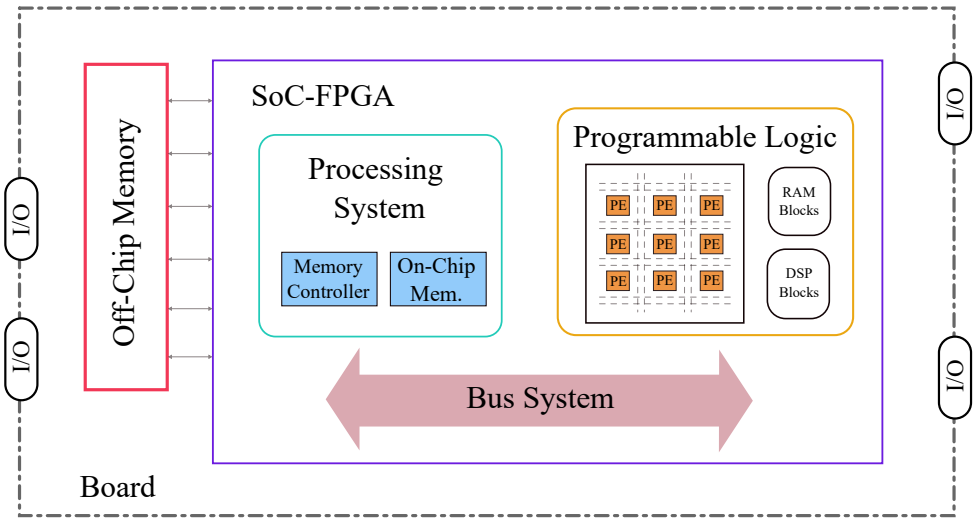


Figure 1.1 – Architecture of a SoC-FPGA, illustrating the integration of a PS, PL, and the bus system.

majority of SoC devices support secure boot mechanisms to protect the integrity and confidentiality of the boot process, it presumes that data encryption on the bus is not required, as the memory and buses are expected to operate in a trusted environment established by secure boot [2]. These unprotected data buses cause a critical weak point in the system, therefore, addressing their HW-level vulnerabilities becomes an interesting area of research to develop effective mitigation.

A Motivating Example in Real-life Scenarios

According to "Big Hack" by Bloomberg in 2018 [38], HW Trojan, with the size of a pencil tip, is planted onto server motherboards, creating stealth doorways for spies to infiltrate into the systems through HW vulnerabilities. This incident highlights the potential risks associated with HW Trojans, especially when sensitive data is transmitted over unprotected buses, a scenario often encountered in SoCs. Because of the ever-increasing cost for manufacturing of semiconductors, the IC foundries outsource their fabrication to a third-party companies [39]. As reported by the survey in [40], a malicious actor working in the design house could introduce HW Trojans or alter existing circuits at the design phase. A HW Trojan is an unauthorized modification embedded into a circuit by an adversary with the intent to exploit HW vulnerabilities [41]. Such modifications are designed to either tamper with the functional operations of HW or covertly monitor sensitive data. In the case of unencrypted information transmitted over the SoC buses, these

Trojans can eavesdrop HW leakage to spy on the data. The absence of encryption on SoC data buses creates a vulnerability, as demonstrated in real-world scenarios like the "Big Hack". Although data encryption prior to the transmission over the SoC bus is technically feasible, it is not widely implemented due to the CPU performance degradation and the high costs [42]. As a result, the risk of non-encrypted data buses remains important, emphasizing the importance of exploring HW vulnerabilities, particularly on data buses.

1.5 Contributions

Thanks to their parallelism and energy-efficient property, SoC-FPGAs are increasingly recognized as computationally adequate solutions to be deployed as DNN accelerators. The term "System-on-Chip" implies the whole system packed into a single chip, and thus, faster processing time. They are hybrid semiconductor devices that integrate both hardwired processors and FPGA programmable logic onto a single chip, providing multitasking functionality [43]. Therefore, securing DNN implementations on SoC-FPGA accelerators is likely to become a *de-facto* standard operation of side-channel analysis. EM side-channel, out of other types of analysis (power, timing, or thermal), is particularly interesting due to its non-invasive nature, giving the privilege to be conducted without decapsulation.

Data communication buses internal to the SoC-FPGAs are composed of conducting wire lines. This thesis investigates the hardware-level weakness of such on-chip buses on the SoC-FPGAs using EM leakage analysis. According to Maxwell's equations, these wires radiate EM fields as a result of the time-varying currents associated with their operations. These EM fields, which correspond to the activities inside the SoC, can be detected by near-field H probes, making them vulnerable to eavesdropping. From this analysis, the position of those conducting wire lines is deduced.

Additionally, these buses carry critical data during operation. For instance, in a cryptographic system, the data might be plaintext or encryption keys, while in an image classification NN, the data could be an image. The state of the bus lines changes dynamically to reflect the input data being transferred, especially at high clock frequencies. By analyzing the captured EM side-channels, we not only exploit information about the number of bits that have changed on the bus during data transmission but also determine the direction of these bit transitions, whether a bit is changing from 0 to 1 or from 1 to 0.

By combining the above two findings, *i.e.* the hypothetical location of the bus wires

and the sensitivity of the bus to bit transitions, we discovered that the probe's sensitivity is also influenced by its physical placement relative to the bus wires. Depending on the probe's distance to the cluster of conducting wires, it exhibits varying levels of sensitivity to specific bits in the input data sequence. This spatial dependence allows us to infer which bits along the input data sequence have changed, which in turn leads to better retrieval of the input data.

Based on the analysis of the exploited bit transitions, two novel methodologies to reverse-engineer the EM trace [4] back to the original pixel values are proposed. These approaches enable the reconstruction of the transmitted input data by leveraging the observed state changes on the bus. To validate the proposed methodologies, they are implemented onto an image classifying MLP pre-trained by one of the recent Xilinx Binarized Neural Network (BNN) frameworks to prove that input images to the accelerator are retrieved from EM traces. During the NN inference, the input image pixels are transferred on the bus, which emits an EM field because of the changing pixel values with each clock cycle. The emitted EM field is captured by a near-field probe and then processed to retrieve the information about the bit changes on the bus. Leveraging this information, the recovered image is reconstructed which showcases the content of the original input image. The results achieved by these methodologies are evaluated with HD-level difference, Peak Signal-to-Noise Ratio (PSNR) and classifying MLP.

To summarize, the main contributions of this work are:

1. Conducting bus wires inside SoC-FPGAs produce EM fields in its vicinity, due to the current changes corresponding to the data sent on the bus. This thesis demonstrates that by measuring the strength of these EM fields with a near-field probe, the hypothetical routing of the bus wires inside the SoC is inferred.
2. The analysis of these emanated EM fields also provides information about the number of bits that have changed during each clock cycle. Besides, it also tells the overall direction of the bit transition, whether from 0 to 1 or from 1 to 0. This information enables a deeper understanding of bus activities and provides valuable insights for reconstructing the transmitted data from the observed EM traces.
3. The spatial sensitivity of the probe relative to the bus wires is also studied. It indicates that the allocation of the near-field probe affects the detection of certain bits in the data sequence. This concept is applied to improve the exploitation of bit changes and optimize the data retrieval process.

4. A set of side-channel measurement, in this case, EM, taken across a computing operation [44]

4. By exploiting the EM traces collected from the targeted internal data bus, two attack methodologies are developed. These methods retrieve the input image of NN implemented on SoC-FPGA without the knowledge of Neural Network architecture or training details. An experimental setup is built using a popular and accessible HW target, on which one of the pre-trained and binarized models by the recent Xilinx framework, is implemented. Two types of datasets, the standard MNIST [45] and Fashion-MNIST [46] developed by Zalando are tested with this MLP to verify if the attacks are feasible on gray-scale images with different pixel-level of complexity.
5. Both methods do not require any form of direct interaction with the victim's hardware platform, in other words, there is no need to trigger NN inference by the adversary. It is assumed that the inference begins right after the input image is sent through the targeted bus; thus, if attackers could trigger the inference process, it would offer them an important advantage. However, in this work, synchronization of each trace for the start and the end of input transmission is obtained by analyzing emanations patterns in the signal processing stage after the trace acquisition (rather than by triggering inference).
6. Both methodologies are one-shot attacks where a single EM trace from one inference is sufficient to perform the attack.

1.6 Dissertation Organization

This dissertation is structured in an organized flow, using related work, methodologies, findings, and conclusions. The remainder of this document is organized as follows:

Chapter 2 presents a review of published literature in the scope of Side-Channel Attacks. A taxonomy of SCAs is proposed to organize a vast body of works in a structured manner, helping readers to better understand and categorize the existing research. Besides, the hardware attacks are categorized mainly into two: eavesdropping attacks and side-channel attacks. This classification, along with the taxonomy, highlights gaps in the existing literature, which this thesis aims to address. After presenting the state-of-the-art, the chapter concludes by discussing the novelty of our work and how it contributes to bridging the gaps identified in the literature.

Chapter 3 explains the step-by-step procedure to characterize and capture the near-field EM emanations from a SoC-FPGA. This chapter discusses the basic notions of EM and the nature of EM emanations along with its impact on modern ICs. This step lays the

groundwork for the following two attack methodologies. Before the steady acquisition of EM leakages, a measurement of the transfer function of the probe is obligatory. A square wave signal that mimics the clock frequency of the targeted SoC is introduced to copper wire lines on a simple Printed Circuit Board (PCB). The probe is then put into place and EM emanations coming from that wire are analyzed. Furthermore, EM cartographies are made to identify the maximal leakage zone on the surface of the SoC-FPGA so that an attacker can solely focus on the spatial Point of Interest (POI).

Chapter 4 provides a detailed description of the first attack method of this thesis. The methodology is based on a four-pixel granularity, *i.e.* targets groups of four pixels (hypothesizing the data with respect to data width) are observed at a time instance. This proposed exploitation method is implemented onto the HW experimental setup and the achieved results validated the approach. These results are further evaluated to verify the validity and reliability of these experiments. The contributions of this chapter have been published in EMC Europe 2023 [3] and ARES 2023 [4].

Chapter 5 proposes the second methodology, which operates at a two-pixel granularity by manipulating a pair of pixels from each four-pixel group. The method presented in this chapter is designed to retrieve information from more complex grayscale images characterized by diverse pixel values. This difference in granularity affects the resolution and precision of the attack. Following the approach of the previous chapter, the final results are evaluated and discussed.

Chapter 6 summarizes the contributions of the thesis and discusses its perspective. Feasible future works are also explained to improve the attack performance and open up paths for alternative approaches.

1.7 Chapter Conclusion

This chapter provides an overview of DNNs on HW accelerators, highlights their associated security risks, and outlines the contributions of this thesis. This introductory chapter has laid the foundation for understanding the importance of HW and SW security in the current digital landscape, especially within the context of Artificial Intelligence (AI)-powered embedded systems. Due to the growing prominence of NN applications, they are more fitted to be implemented on HW accelerators to fulfill their price of computational cost and decrease latency as much as possible. We discussed the vulnerabilities these accelerators face due to the lack of comprehensive security measures.

This thesis addresses these vulnerabilities by exploring how near-field EM leakage from on-chip buses inside SoC-FPGAs can be exploited to recover sensitive input data, specifically focusing on dataset images processed by BNN accelerators. To fill the identified gaps in the literature, we propose two novel attack methodologies that leverage EM eavesdropping techniques to infer input data from SoC buses. The motivations behind this research are grounded in real-world applications, such as potential risks to AI-powered surveillance cameras, and emphasize the importance of secure HW designs in sensitive environments. The contributions of this work include the demonstration of successful image reconstruction through EM eavesdropping, along with a comprehensive assessment of the limitations and potential applications of these methods. The subsequent chapters are organized to build on this foundation. Together, these chapters will provide a thorough investigation into the vulnerabilities of SoC-FPGA systems and propose pathways for enhancing their security in embedded AI contexts. The upcoming chapter explores existing research and methodologies, offering a detailed overview of the state of the art to better inform the reader of the foundational work in the SCAs.

Chapter 2

Related Work

Contents

2.1 Taxonomy of Hardware Attacks 39
2.2 Eavesdropping Attacks 41
2.2.1 Eavesdropping of Power Traces 42
2.2.2 Eavesdropping of EM Emanations 43
2.3 Side-Channel Attacks 45
2.3.1 SCAs on Cryptosystems 46
2.3.2 SCAs on Neural Networks 49
2.3.3 Cryptosystems and Neural Networks: Challenges in Information Recovery 54
2.4 Advancements Beyond the State-of-the-Art 55

Hardware attacks refer to any security compromise which exploit the vulnerabilities in the physical components of an electronic device or a computing system [40]. These attacks are performed through a variety of means, either by spying on the vulnerable HW components to eavesdrop information [47, 48] or by exploiting its side-channel information produced by the HW to compromise the security of the system [49, 44]. Exploitable HW vulnerabilities pose a significant threat to the confidentiality of private data. Although such attacks necessitate for the knowledge and physical access to the HW devices, along with specialized equipment, they are considered effective in bypassing SW-based security mechanisms. There are two categories when it comes to classifying the nature of the HW attack: *active* or *passive* [9]. In an active attack scenario, the targeted HW is tampered with faults to function abnormally. Faults can be referred to as clock/power glitches, magnetic pulses, or sometimes, the injection of laser on silicone die [50]. Passive attacks, on the contrary, do not tackle the system itself. Instead, they monitor non-functional characteristics of the HW or eavesdrop on its inside activity discretely.

Having introduced the concepts of active and passive attacks, this thesis focuses exclusively on passive attacks, as we do not directly interact or tamper with the targeted system. The passive attacks are divided into two main types in this thesis: **Eavesdropping attacks** and **Side-channel attacks**.

The act of illegally listening to a private conversation between two people is called eavesdropping [51]. Eavesdropping attacks are defined as the unauthorized interception of a private communication over a channel and can occur when the connection between two endpoints (sender and receiver), lacks security, *i.e.*, non-encrypted [52]. Although there are various categories of eavesdropping attacks, such as wireless eavesdropping [53], network sniffing [54], man-in-the-middle attacks [55] etc., we focus on physical eavesdropping of HW devices due to its relevance to our work.

Side-channel attacks exploit unintended leakage from the physical implementation of a cryptographic system to retrieve secret encryption keys. They target the side-channel information rather than directly aiming at breaking the underlying execution algorithm. Cryptographic algorithms (*e.g.*, Advanced Encryption Standard (AES), Rivest, Shamir, Adleman (RSA), Elliptic Curve Digital Signature Algorithm (ECDSA)) provide solid security by utilizing complex mathematical algorithms to protect their sensitive data. Since SCAs exploit HW vulnerabilities as side-channels rather than brute-forcing the algorithm itself, they are highly effective in stealing the secret encryption key.

Although initially created to target cryptographic algorithms, the focus of SCAs has

shifted onto NN accelerators in recent years. With NN accelerators being extensively deployed across various domains, they often require processing sensitive user data. In the case of NN accelerators, SCAs exploit HW-level weakness to extract physical side-channels, which contain information about data behavior, such as NN architecture, weight parameters, or input. Different exploitable side-channel information exist, in which the most common ones include, EM emanations, power consumption, timing, sound, temperature, etc. Hence, the implications of HW vulnerabilities on NN-embedded accelerators should be studied carefully.

2.1 Taxonomy of Hardware Attacks

Despite the diversity of hardware attacks in current state-of-the-art, they can be categorized systematically into distinct based on a few axes as proposed in Fig. 2.1. This classification will help clarify the scope of SCAs and aid the reader in comprehending better about their nature.

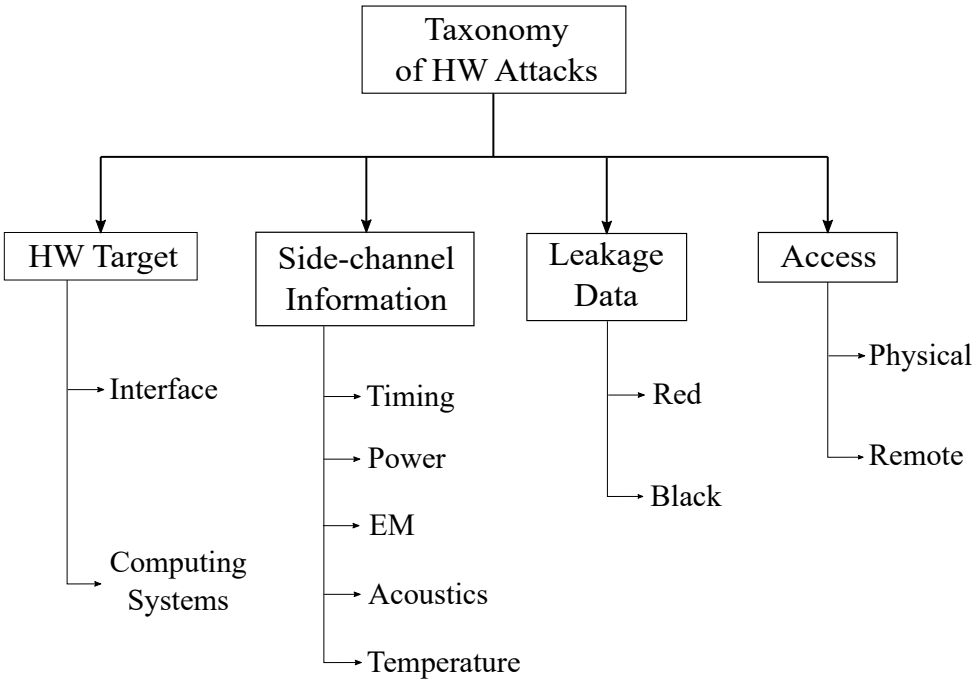


Figure 2.1 – The taxonomy of hardware attacks.

- **Hardware Target**: According to [7], a vulnerability is a weakness inside a HW component of an information system that, if exploited, compromises Confidentiality,

Integrity, or Availability. Any factor that exists within a HW that threatens at least one of these three key aspects is susceptible to being targeted by attackers, as it is not fully secure. These HW devices can either be *interfaces* (e.g., screen, cables) for eavesdropping attacks, or *computing systems* (e.g., microprocessors, FPGAs) that perform sophisticated computational tasks as in cryptographic algorithms, or complex Neural Networks.

- **Side-Channel Information**: Due to the HW vulnerabilities, the HW targets have potential to leak physical side-channels. In side-channel analysis, a variety of physical leakages produced by the vulnerable HW are collected and analyzed to retrieve confidential information from the victim's system. *Timing* information is often observed by the adversary to deduce the duration of certain program execution, which in turn reveals hints about underlying instruction or data [56, 57]. As different operations or data states in a system can produce distinct power signatures, the malicious attacker can exploit *power* usage of the victim's circuit with the final purpose of recovering important data, keys, or architecture [58, 59, 60, 61, 62, 63, 64, 65, 66]. Side-channel information such as the *potential difference* from exposed metal parts on hardware devices (e.g., voltage fluctuations on a laptop's metal chassis) are exploited to eavesdrop on processes running on a laptop or encryption activity [67]. In EM analysis, conducting wires inside a PCB can emit *electromagnetic*, field unintentionally, in which they contain secret details about the operations, taking place inside the targeted HW [68, 69, 70, 71, 72]. Additionally, *acoustic* analysis consists of capturing noise patterns or vibrations generated by HW, keyboards in particular, and exploiting them afterward to obtain keystrokes and data processing patterns of a computer [73, 74, 75]. Differences in thermal outputs of a HW component represent different processing states [76, 77, 78]. Besides, changes in surrounding *temperature* of a system can operationally impact, or in the case of a NN, can even drop its accuracy. Even though all exploitable side-channel information in the current state-of-the-art is described, only power and EM properties will be discussed consistently in this chapter because of their similarity to our work.
- **Leakage Data**: The red/black concept is first introduced in NACSIM report [79] by National Security Agency (NSA). The red signal refers to information that is unprotected and non-encrypted, such as plaintext, and therefore considered as highly sensitive material. The black signal, on the other hand, applies to safe information that is already encrypted by a secret key, like cipher text [80]. Based on this concept,

the terms *red* (for non-encrypted data) or *black* (for encrypted data) are adopted and used regularly throughout this thesis.

- **Access**: Two types of access, *physical* and *remote* are generally required for the SCA to be implemented successfully. For the *physical* access, the attacker must be in the vicinity of HW system so that a near-field probe can be placed to observe side-channel information. In the case of *remote* access, the attacker spies on the target at a moderate to significant distance with the help of SDR for a long-range antenna. Another feasible approach to perform remote access is that the victim and the adversary must share the same HW resources [81]. This scenario prevails in environments such as cloud FPGAs, where multiple tenants share a common FPGA provided by cloud services. Due to the lack of strict HW isolation, the adversary has access to the side-channel leakage on shared HW and further exploits to steal confidential data of the victim.

2.2 Eavesdropping Attacks

An eavesdropping attack is the interception of information from electronic *interfaces* that unintentionally radiate or leak emanations [82]. As illustrated in Fig. 2.2, eavesdropping is done when a malicious person intercept the data transmitted over a communication channel that is established between two devices. Even though eavesdropping can be performed using different types of emissions, such as electrical signals, acoustic and vibrations [83], this thesis predominantly focuses on discussing how power and radiated EM waves are eavesdropped in the current literature. The term "TEMPEST" is a code name given by the U.S. government in the late 1960s [84] and refers to the eavesdropping of non-encrypted leakage, *i.e., red data*. According to the book written by P. Wright [85], the first official occurrence of an eavesdropping (TEMPEST) attack was reported in 1965 by MI5, the British intelligence agency. The microphone bug was secretly deployed in the Embassy of Egypt in London and MI5 was able to spy on the internal communication of the embassy for years just from the clicking sound of the rotor-cipher machine used by the clerk. According to [86], the pioneering works in eavesdropping attacks mainly focused on intercepting analog signals while recent works expand to digital signals on embedded circuits.

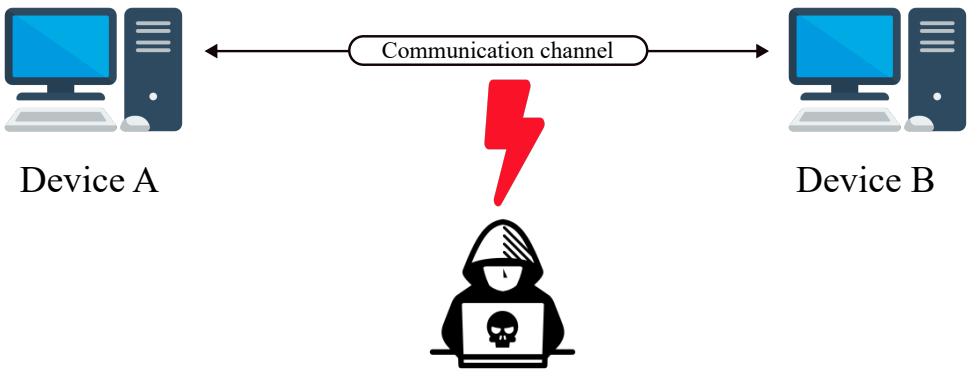


Figure 2.2 – An eavesdropper illegally intercepts the information transmitted over the communication channel between two devices, a sender and a receiver.

2.2.1 Eavesdropping of Power Traces

Power eavesdropping attacks have been used to spy on activities operating on certain electronic interfaces, such as Universal Serial Bus (USB) cables, power outlets, and battery chargers. As stated in [87], the first mention of the eavesdropping method on power traces from USB interfaces was in [88]. The authors observe voltage variations at the USB port of the host computer to expose ongoing activities within the system. From the power analysis, the authors are successful in distinguishing the processes on the computer, such as the execution of OpenSSL RSA decryption. Computers plugged into power outlets leak information through its power side-channels. To enhance user experience, web browsers consume resources, leading to variations in AC power traces depending on the complexity of the web page. Based on this fact, authors of [89] tap into the AC power outlet to which the computer is connected and monitor the power traces using 0.1 Ω resistor and Data Acquisition (DAQ). These traces are used as training data in Support Vector Machine (SVM) classifier and then used to identify the web page that is being browsed on the targeted computer screen.

Modern smartphones use Liquid Crystal Display (LCD) and Organic Light-Emitting Diode (OLED) technologies for the touchscreen. Depending on the color of the pixel on these screens (brighter or darker), the power is consumed differently. When the button is pressed on the screen because of the user input (e.g., typing PIN code), it triggers the power spike due to the dynamic power consumption. Besides, different states of phone, sleep, locked, or receiving input from users, cause different current utilization. The authors of [90] collect such power traces from the USB cables when the phone is being charged

and then train two different classifiers: one for the detection of the smartphone model and another for the detection of the button pressed on the screen. By doing that, they are able to retrieve private data such as the user's password.

With the rise of the popularity of wireless charging standards, the authors of [91] develop a website fingerprinting attack on smartphones to spy on the websites that are being browsed on the victim's phone screen. Wireless power transfer works on the electromagnetic induction between two coils, one inside the charging dock or power bank and the other in the smartphone. If the user is browsing the internet during the charging period, the current fluctuation occurs and is detectable by a current sensor which is planted secretly inside the charging dock. These fluctuations caused by loading different web pages are more evident when the battery percentage is more than 80%. Two classifiers, Convolutional Neural Network (CNN) and Random Forest [92] are trained by the collected current traces to infer the correct website in which the first one gives better recognition accuracy than the second. Similar to [91], the authors of [93] identify the exploitable weakness in contactless charging power banks and propose a framework called "BankSnoop". Once the phone and power bank are attached to each other, it causes coil whine [1] which marks the beginning of the charging process. In addition, if the user continues to use the phone during the charging, it produces fluctuations in the induced current which in turn results in magnetic-field disturbance in its surroundings. These magnetic-fields are examined to reveal users' in-app activities such as play/pause video, commenting, etc. However, the differentiation of [93] from its related work is that they use less current traces as training data for their CNN (which is used to infer the web activity on the phone) and their attacking device is portable with the size of an Airpod, making the attack more feasible in real scenarios.

While these attacks are mostly based on monitoring the power behaviour coming from the peripherals and cables attached to computers and mobile devices, our work highlights the eavesdropping of information inside a specific computing unit, which are SoC-FPGAs.

2.2.2 Eavesdropping of EM Emanations

Although there is a substantial amount of EM eavesdropping research publicly available in current literature, the early works focused on exploiting the EM leakage on the displays and other peripherals. In 1985, a Dutch researcher Van Eck illustrated the eavesdrop-

1. Electromagnetically induced acoustic noise due to the vibrations of the coils.

ping of information from Cathode-ray Tubes (CRT) screens at hundreds of meters away using low cost (around 15$ worth) equipment [47]. Following this work, MG Kuhn [48] proved that this vulnerability is not only limited to CRTs screens but also applicable to flat panel displays, including Video Graphics Array (VGA), High Definition Multimedia Interface (HDMI) and Digital Visual Interface (DVI). Kuhn proves that the eavesdropping of flat displays is feasible because of the digital serial cables that are used to connect the display and the video controller. The compromised signal is successfully captured from a distance of 10 meters (through three intermediate plasterboard walls) without relying on a directional antenna.

In [94], the authors develop Deep-TEMPEST which uses an antenna and a SDR to acquire EM emissions from a monitor connected to the HDMI cable and then successfully reconstruct the video frames of digital displays. They presented two operational attack scenarios: (1) the victim's system and the eavesdropper are separated by a wall, and (2) a small omni-directional antenna is positioned in close proximity to the cable, requiring physical access. An end-to-end CNN is trained with the captured EM radiations and is able to reduce the Character Error Rate of the recovered text by 60% in comparison with other TEMPEST attacks. Besides, authors of [95] proposed an approach for retrieving video images displayed on the ultra high-definition screen by collecting EM radiations from as far as 80 meters away. This demonstrated EM compromise originates from the internal electronic component of a Video Display Unit (VDU) connected to a laptop via HDMI cable. Their work focuses on enhancing the Signal-to-Noise ratio (SNR) of the emitted signal by exploiting the periodicity of the video signal, rather than revisiting the feasibility of TEMPEST, which has already been established in related work.

Recently, authors of [96] demonstrate screen gleaning, which is a type of TEMPEST attack, on the display cable of a mobile phone screen and is able to read sensitive data (*e.g.* bank sign-in codes) on the screens without being in their line of sight. This method involves positioning a near-field probe at a distance of less than 1cm because, in small modern smartphones, the screens consume much less energy thanks to the compact circuitry inside, causing the coupling effect to be lowered. This makes the acquisition of EM leakage signal to be more complicated, thus, close access to the phone is required compared to traditional TEMPEST attacks which operate at a distance. A new security vulnerability, related to embedded cameras, is investigated in [97]. The camera sensor and the Image Signal Processor (ISP) inside the embedded system controller are connected by pixel transmission interface, such as Mobile Industry Processor Interface (MIPI), digital video port (DVP),

etc. By placing a near-field EM probe in the vicinity of that transmission interface, it is possible to capture EM emissions and then use to reconstruct approximate gray-scale images. They also compare three attack scenarios with different distances (10, 100, and 150cm) and found that eavesdropping can still be performed if used with SDR along with a high-quality directional antenna.

According to [97], EM emission produced by small circuits have very low energy and require the positioning of measurement instrument (for e.g., probe, antenna) close to the targeted HW component. As a compact SoC-FPGA is utilized in our experiments, it is mandatory to put the probe close to the leaking component for better acquisition of EM emanations. However, the authors in [97] also describe that by connecting the probe output to the SDR instead of an oscilloscope, the attack becomes implementable at a further distance. Although this approach is not yet implemented in our work, it highlights an intriguing opportunity for further research, enabling remote analysis for our methodology and allowing greater practicality.

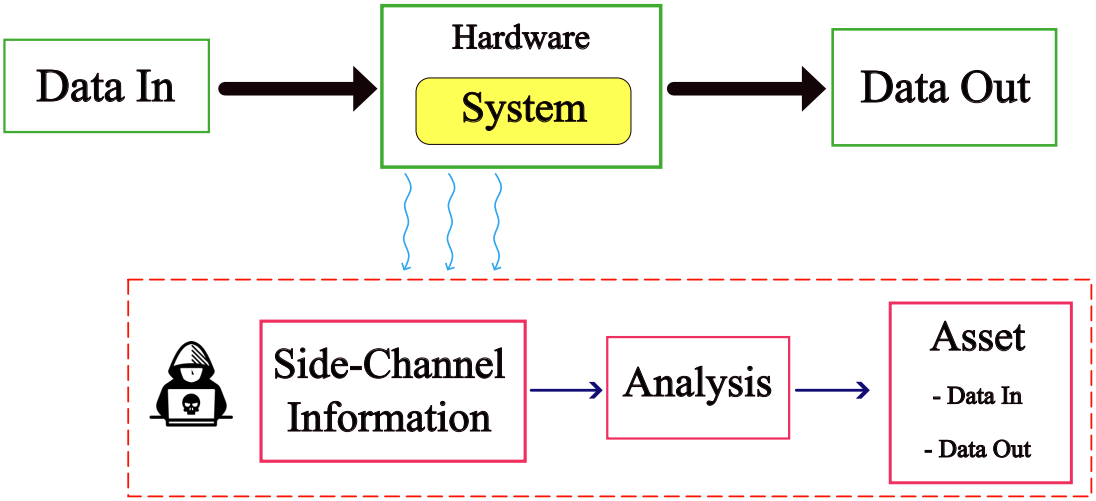


Figure 2.3 – An illustration of basic mechanism behind a Side-Channel Attack

2.3 Side-Channel Attacks

While SCAs were initially developed to compromise the secure cryptographic algorithms by extracting the secret encryption keys [49, 44, 98], their application has recently expanded to DNNs, where input data, architectural details, and parameter values are exploited through physical characteristics of the vulnerable hardware. This is why, in this

thesis, both SCAs on **cryptosystems** and **neural networks** are primarily discussed. Fig. 2.3 illustrates a traditional computing system with its input and output data, showing how an attacker can exploit side-channel information to steal valuable assets from the system.

2.3.1 SCAs on Cryptosystems

As cryptographic algorithms have become more resistant to traditional brute force attacks, SCAs has emerged as an alternative security compromise due to their capability to exploit the system from observable side-channel leakage rather than mathematical vulnerabilities. Cryptographic modules which are implemented on *computing systems*, are targeted occasionally to extract the secret encryption key. Once the secret key is extracted, the adversary can easily decrypt the cipher, hence, this type of leakage is referred to as *black data*. Fig. 2.4 depicts the basic procedure of a cryptographic SCA. The attacker exploits physical side-channels from encryption HW to steal the secret key (asset) of the cryptographic algorithm. Generally, the attacks on cryptosystems can either be accessible physically or remotely.

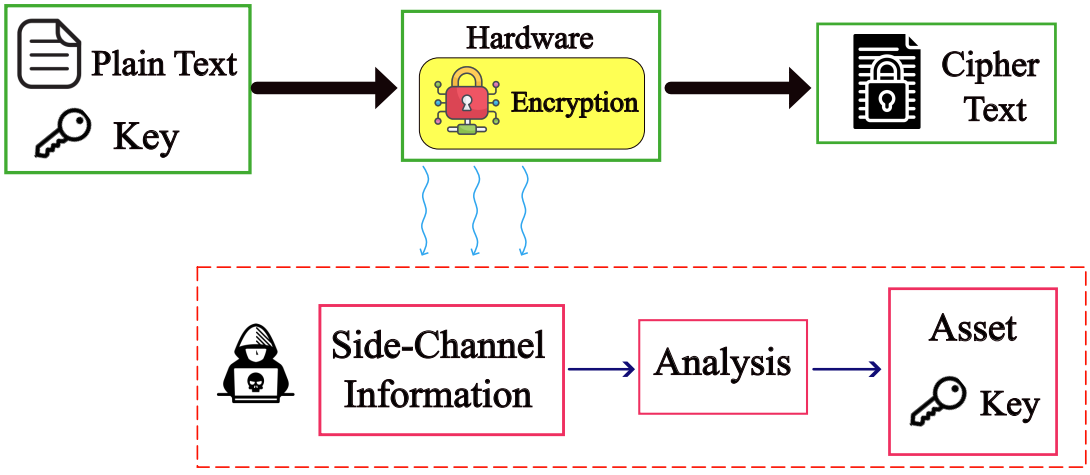


Figure 2.4 – An overview of the working principle of a SCA on a cryptographic algorithm.

Crypto: Power Traces

In the late 90s, Kocher et al. [49] first developed timing analysis to highlight the power activity over time for certain cryptographic algorithms like RSA, Digital Signature Standard (DSS), etc. In 1999, the same author demonstrated the Simple Power Analysis (SPA)

and Differential Power Analysis (DPA) attack to underscore the power dependency of cryptographic operations and was successful in retrieving the encryption key. SPA analyzes power traces to identify patterns, e.g., spikes and variations, directly revealing secret keys of cryptographic algorithm [99]. DPA, however, is an advanced statistical method to uncover sensitive information from multiple power side-channels by identifying the relationship between processed data and its power behaviour [9]. It is worth noting that SPA only works on algorithms which possess distinct patterns, thus, more and more algorithms start to put masking countermeasures to prevent SPA. As for the implementation of DPA, the attacker requires multiple access to the system to collect and analyze numerous traces. To address these limitations, template attacks were first introduced in [100] as an alternative method for extracting cryptographic keys. Even though building a template requires a preliminary phase (*i.e.* profiling stage) to collect samples, this kind of attack is more effective than SPA and DPA as they only necessitate a single trace at the actual attack phase.

Since then, many power SCAs have emerged with the purpose of stealing the key of encryption algorithms. Authors of [101] demonstrate that the registers within pipelined AES architectures are particularly vulnerable to Correlation Power Analysis (CPA) attacks. Pipelining is a technique used to increase the throughput of the AES algorithmn [102]. In [101], the pipeline structure is built by providing 10 AES cores, one for each round of standard AES design, adding up to 10 rounds in total. These cores are connected by registers to store the output of each round. The extraction of AES key is done by CPA and their results show that the 16-byte key is retrieved with only 4,900 traces, making their demonstrated model 4 times more effective.

In traditional FPGAs, the power measurement units like time-to-digital converter (TDC) or ring oscillators are implemented exclusively on PL, taking advantage of their reconfigurable nature. However, on the SoC architecture, these sensors have limitations to be implemented. In [103], the authors aim at using delay-line sensors to exploit voltage fluctuations derived from the propagation delay of digital logic gates inside SoC environment. Using two delay-line measurement techniques, such as Delay-Locked Loop (DLL) and delay block, they demonstrate SCA, called *SideLine*, which perform full key recovery on the encryption key of OpenSSL AES implementation.

Crypto: Electromagnetic Emanations

In 2001, authors of [98] stated that EM side-channel analysis is as effective as power in extracting cryptographic keys from smart cards. Experiments are conducted with Data Encryption Standard (DES), COMP128, and RSA and are able to correlate the processes of the crypto-algorithm. One year afterward, an overview of EM side-channels on CMOS devices is presented in the work of [104]. The authors distinguish the two types of EM side-channel: direct emanation and unintentional emission and discuss methods for the acquisition of each type. These two works have opened up ways to exploit EM information on HW different targets beyond smart cards.

According to [105], EM cartography highlights the active region on the FPGA when it is executing AES encryption. The authors try two placements of ring oscillators on FPGA chip: first, close to the encryption co-processor, and second, farther away to analyze the leakages captured with a near-field probe set. It is found that the precise location of cryptographic module is not mandatory to perform the attack. They also demonstrate that while the EM imaging can tell the approximate leaking zone on the chip, it is not capable of revealing the detailed underlying layout of the FPGA. However, in our work, we have found that by analyzing the strength of EM emanations with respect to the probe's position on the top of the wires, we can deduce the routing of SoC data bus.

Square EM attack is introduced in [106] which leverages the algebraic properties of AES. By crafting a particular set of plaintexts with specific active and passive bytes [2], the attacker analyze the EM traces captured with a probe during MixColumns operation to distinguish the transitioned state based on active byte counts. The best key hypotheses are formed and are then validated using EM signature, enabling key recovery.

In [107], the authors perform the AES key retrieval from CPA. However, for the acquisition of EM emanations, near-field probes with different diameters are deployed at varying distances from the target: (1) 1cm loop placed directly above the target, *i.e.* FPGA, (2) 5cm loop placed 10cm above the target, (3) 5cm loop placed 20cm above the target and (4) 5cm loop placed 30cm above the target. The analysis reveals that the key guesses across the four scenarios are similar when using 1,000 samples. However, the accuracy improves as the number of samples increases. Notably, with 10,000 samples, scenario (1) produces the most accurate key hypothesis among all scenarios, while scenarios (3)

2. Bytes that change during transformations (e.g., SubBytes, MixColumns in AES) are called active bytes while those that stay constant and remain unchanged after certain operations are called passive bytes.

and (4) yield similar results. The authors insert a trigger signal at the initialization of the encryption process to accurately capture EM emission and align the traces perfectly. However, this approach is impractical in real-life scenarios, which is why to address this issue, our work employs pattern matching to align data-carrying EM traces. This alignment technique without using an external trigger will be discussed in detail in Chapter 3.

Additionally, cryptographic operations implemented on the SoC environments are vulnerable to EM side-channel analysis as seen in [108]. For a close-range acquisition of EM emanations, a near-field probe is placed near the main CPU and its co-processor of the SoC and then analyzed using CPA to recover the encryption key. In the work of [109], the authors introduce the STELLAR method, which facilitates the local routing of the AES-128 encryption core within the lower-tier metal layers of an integrated circuit. A novel class of HW Trojans that AES, by leveraging modifications in the lower metal layers of the physical layout without altering the gate-level netlist or inserting additional logic. By manipulating these routing paths, the Trojan leaks encryption keys through path delay variations. Moreover, authors of [110], develop an efficient EM mapping that reduces the size of the scanned chip surface area by comparing the normalized standard deviations. They target the off-chip memory buses in 2.5-D IC packaging technology where different parts of the chips are integrated on a silicon interposer. EM probe made with single turn rectangular coil is placed on top the bus, *i.e.* point-of-interest derived from their proposed EM mapping method. Compared to the similar works in the state-of-the-art, their methodology is proven to be successful with fewer traces (less than 256). According to [111], a deterministic delay is inserted into the interposer-based circuit to mitigate the EM-SCAs, while keeping the SNR at favorable level. The memory interconnects embedded in the interposer's distribution layers are specially targeted by the near-field probe to collect EM traces, which are used to calculate right AES key guesses. By inducing non-random delays as proposed in [111], the correlation between the traces and the correct key is reduced.

2.3.2 SCAs on Neural Networks

Even though SCAs were initially used to extract cryptographic algorithms, their focus has recently shifted to NN implementations. Since NN contains complex calculations, high computational power is essential to be able to execute them smoothly with little to no latency. This makes *computing systems*, like accelerators, to become suitable candidates

for Neural Network applications [36]. In response to the expanding role of Deep Learning applications on HW accelerators, they are required to process large amounts of sensitive data. Because of this reason, NN accelerators are more and more targeted by adversaries to either steal input data or reverse engineer the design [112]. Most NN have their input data, output labels, architecture, and hyperparameters transparent and remain non-encrypted (*i.e., red data*). In Fig. 2.5, the adversary collects the side-channels emitted from HW component and analyzes them to retrieve either the input image to the inference or the NN architecture or the parameters as valuable assets. These attacks can be carried out through both physical and remote access.

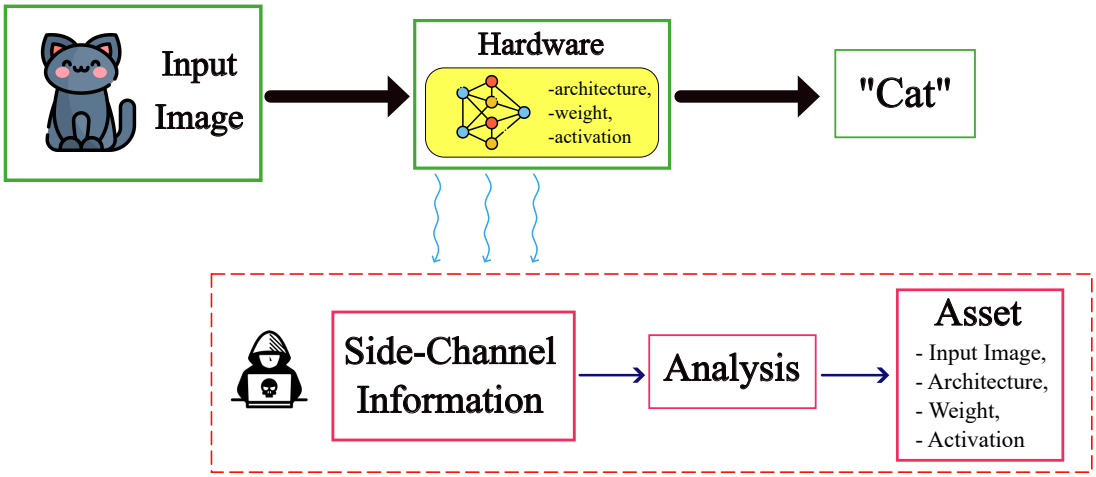


Figure 2.5 – A block diagram depicting the process of a SCA on a NN.

NN: Power Traces

The first mention of power SCAs that compromises the confidentiality on DNN accelerator was in [59]. The authors calculate the power consumption for each layer (convolution, pooling, fully connected) and sparsity values (ratio of zeros) of different NNs implemented on the Raspberry Pi platform. They use these power signatures and model sparsity as features to train the SVM classifier, and then use it to infer the DNN architecture, obtaining up to an average accuracy of 96.50%. Subsequent to this pioneering study, analysis has revealed that the weight parameters of NN can also be exploited. The authors in [66] demonstrate a power SCA on FPGA-based systolic array circuit which mainly consists of a wavefront array of 3 x 3 Processing Elementss (PEs). From the acquisition of 100,000 power traces from PEs, the weights values of implemented MLP are

recovered by using sequentially executed CPA, called chain CPA. In [65], the authors are successful in retrieving the MLP architecture, weights, and even the activation functions. They first use SPA to reveal each layer of LeNet which is implemented on FPGA platform. As for retrieving the weight parameters, they perform multi-stage CPA which is based on chain CPA method described in [66]. The novelty of their work in comparison with its state-of-the-art is the recovery of activation functions. They implement five different activation functions, ReLU, Sigmoid, Tanh, Softmax, and Swish using the piecewise-liner (PWL) function and lookup tables on the same FPGA. Afterward, the corresponding power traces are collected, and a total of 500 traces are used to train an MLP for classifying activation functions. Moreover, power SCAs are capable of retrieving non-traditional NN related parameters. Authors of [61] use the recently developed FINN framework to implement pre-trained BNN on SoC-FPGA. They reverse-engineer the folding factor [3], which is a FINN exclusive parameter, and the number of neurons in BNN, by training Random Forest Trees classifier with the TDC-captured power traces.

In addition, authors in [58] focus on retrieving the secret NN inputs from the line buffer [4] of the FPGA. Since the power traces collected from the oscilloscope are distorted, the authors also present a power extraction technique as a solution to denoise power traces. Firstly, these noise-free power traces are analyzed to differentiate the background and foreground of the MNIST images, which are the input to their DNN. This differentiation is done by the definition of threshold as the power consumption between background, *i.e.* black pixels, and foreground pixels is evident. Secondly, a power-pixel template is generated with different kernel sizes (3x3 and 5x5) and compared with the acquired power traces during the inference process to deduce the individual pixel value of the input image.

Power SCAs are not only limited to FPGAs but also can be applicable on other HW platforms, such as micro-controller, GPUs. In the work of [64], parameters of NNs with different precision (floating point, fixed point and binary) and their input images are exploited using timing information and SPA. Since the timing information is dependent on the multiplication of input and the weight of the first layer of NN, the authors perform step-by-step model recovery by first recovering the mantissa of weight, followed by extraction of three rounds of weights and bias and then continue onto successive layers. As the first layer has to handle the input image directly, they are able to reverse engineer the approximate input pixel values once the necessary parameters of the first layer are

3. The detailed description of folding factor will be presented in section 4.1.1
4. HW structure for the implementation of CNNs.

achieved.

Nowadays, more and more cloud service providers are starting to integrate FPGAs into their infrastructure to offer high-performance computation, scalability, and accessibility for users. In such scenarios, multiple users share the same cloud FPGA platform for resource optimization and cost reduction. This opens up plenty of ways for the adversary to attack the victim's implementation without physical access as long as the former is co-located with the latter on the same FPGA platform. Authors of [113, 60] prove this possibility by demonstrating a remote SCA that mainly compromises TDCs to get traces from the power distribution network of FPGA during the execution of first convolution layer. From this side-channel information, they extract the input image to the BNN and CNN with high accuracy as the first layer directly handles the input pixels. In addition, the authors in [62] take advantage of Generative CNNs [5] to reconstruct original input images (MNIST and Fashion-MNIST) from remote power traces captured with TDCs. As these raw traces are directly fed into generative NN to infer the pixel values, the precise synchronization of traces is not required. Furthermore, the paper [63] studies the side-channel vulnerability of cloud-based FPGAs deployed with DNNs. Ring oscillators, serving as power sensors, are placed on the multi-tenant FPGA to collect power traces, leaked from the victim's inference operation. Depending on where the ring oscillators are placed, either evenly distributed across the whole surface or limited to a specific part of the FPGA, the results differ. Eight Machine Learning (ML) classifiers are trained and then used to distinguish the architecture and hyperparameters of the victim's DNN, evaluating their recognition accuracy afterward.

NN: Electromagnetic Emanations

Similar to power traces, EM SCAs can be applicable on NN accelerators to retrieve important system information. DeepEM [68] proves that attackers can accurately reveal the underlying NN model hyperparameters by exploiting EM emanations from the interconnects on the PL part of SoC-FPGA, with the use of a near-field EM probe. Authors use adversarial active learning to generate informative queries that estimate the parameters of the network, effectively reconstructing the entire model (referred to as the substitute model in their paper) from a black-box setting. In other words, the attacker turns the victim's black-box into a white-box. The drawback of DeepEM is that since they guess

5. A type of CNN able to generate new data samples that resemble the original dataset from the underlying distribution of the input data [114].

plausible configurations to make the substitute model, this methodology is only applicable on a limited set of commonly-known DNNs architectures, e.g. AlexNet, VGG, etc.

In [69], the results prove that the secret BNN weights of the first layer are successfully exploited by Correlation Electromagnetic Analysis (CEMA). They mainly focus on the transition of bits in the register when the multiplication of input and weight value is executed. Subsequently, each layer of BNN is also identified by the visual inspection of timing information from the measured EM trace. According to [70], the architecture of NNs running on HW accelerators are inferred from EM leakages. The two CNN models: one for classifying the convolution layer and the other for identifying the remaining layers, are trained with the collected traces. By using these two profilers, the authors demonstrate the feasibility of revealing CNN architectures from near-field EM emanations.

Similar to how power SCAs are not restricted to FPGAs hardware platforms, EM SCAs have also been conducted on non-FPGA platforms in the literature. The authors of [72] demonstrated the complete parameter recovery of the MLP implemented on the micro-controller. Parameters like the number of layers, neurons per layer, and activation functions are revealed by the timing analysis due to the sequential processing nature of micro-controllers. As for the weights, they guess all possible weight hypotheses and perform CPA to select the highest correlated value (They assume that the attacker has access to input). However, for the HW platforms like FPGAs where parallelism is favored, the timing analysis may fail and requires an alternative solution for the parameter retrieval. Likewise, [115] proposes "Clairvoyance", a methodology to extract layer configuration of DNN implemented on GPU from a remote location. They exploit the vulnerability of GPUs memory clocks which propagate far and carry sensitive information about DNN architecture. The antenna is set up at different distances with varying conditions: first, positioned at 2 meters away from the victim's GPU without any obstacles, and second, at 2 meters away with 16 centimeters thick drywall in between the victim and attacker. Regardless of physical barriers, not only the number of layers but also the significant details about DNN models (kernel, stride, etc.) are exposed to the attacker. The limitation to this work is that the parameter retrieval is not possible as their leakage is small and invisible on the EM trace.

While the existing works in the literature mostly concentrate on retrieving the parameters or reverse-engineering the NN architectures through EM SCAs, limited attention has been given to the recovery of input information to the inference. This poses a significant gap in the current state-of-the-art, therefore, our work addresses this gap by exploiting

vulnerabilities in SoC data buses to recover input images processed by the NN.

2.3.3 Cryptosystems and Neural Networks: Challenges in Information Recovery

While the majority of SCAs have focused on cryptographic algorithms, recent studies have begun to explore HW vulnerabilities in NNs applications. In our thesis, we focus on exploiting EM side-channels to retrieve secret input information from NNs and face a few complications in comparison to key recovery of cryptosystems. The two main challenges include:

— **System Access**: In cryptosystems, the attackers have repeated access to capture the side-channel leakages (e.g., power consumption or EM emissions), enabling them to gather sufficient traces to infer the secret key. On the other hand, the access is often more limited in stealing secret information of the NNs. For instance, if the goal of the attack is to exploit the input to the NN, the attacker may only be able to query the model a few times. This scenario is common in NNs, deployed on an Internet of Things (IoT) device or an embedded camera, making the attacker's interactions with the system restricted.

— **Targeted Asset**: In cryptography, the attacker aims to extract a secret key, which remains unchanged. In NNs, the target is more complex when the attacker seeks to steal model parameters, such as weights, which can change over time as the model is updated. Therefore, the asset is less stable compared to the encryption key.

Considering these challenges, we aim to perform our attack using only a single acquisition of an EM trace. Contrary to existing works in the literature that rely on ML techniques in which several acquisition of traces (power or EM) is required, our approach avoids the need for multiple acquisitions, overcoming the scenarios where multiple access to the system is infeasible.

Although our methodology requires a pre-attack phase for template creation to correlate the captured EM emanations, only a few traces, one for each sample of the template, adding up to 15 samples in total, are necessary. Since the exploited HW vulnerability is the SoC bus, which operates independently of the application implemented on the programmable fabric, this phase can be carried out as long as the attacker has a HW replica of the victim.

2.4 Advancements Beyond the State-of-the-Art

This thesis focuses on exploiting HW vulnerabilities on these non-encrypted data buses, emphasizing the potential risks by the adversaries to extract sensitive information from unprotected data communication channels. EM leaks induced by manipulation data on SoC buses are exploited to retrieve the secret input images of NN accelerator. We demonstrate that such attacks are feasible and present two novel methodologies to interpret EM emanations and transform them into input pixels. Besides, our experimental EM analysis also reveals the routing and position of conducting bus wire lines inside the chip that emit the corresponding leakage. This aspect represents a significant gap in the existing literature, as no prior work has demonstrated the ability to deduce the underlying wire routing within the chip based solely on EM emanations. To the best of our knowledge, this thesis is the first to reveal the correlation between the level of emanation and the physical position of the source of leakage.

Both our work and [58] focus on extracting input images to NN accelerators. The approach in [58] employs a two-step method: (1) background pixel detection using a threshold and (2) a template attack based on the pixel-power templates. Similarly, our work utilizes a threshold in trace to identify background pixels, which are characterized by lower EM emanations (lower peaks). Additionally, we constructed HD-EM template and perform correlation throughout the acquired EM trace. The retrieved HW are subsequently used to reconstruct original pixel values. However, they target the vulnerability of the line buffer inside PL, whereas we exploit the specific vulnerability of the data bus within SoC-FPGA. In other words, although both methods recover input data, the source of the HW leakage is different. Besides, they use power side-channel analysis, while we rely on EM traces. EM analysis is preferred over power in this thesis thanks to the clarity it gives in the identification of bit transitions. While both power and EM side-channels expose information about the direction of data bits flipping (0 to 1 or 1 to 0), the distinction in power traces is subtle as the bit transitions from 0 to 1 result in slightly higher power consumption than transitions from 1 to 0. However, this difference is marginal and challenging to distinguish. On the other hand, EM traces provide a more evident indication of transition direction. The polarity of peaks in EM signature, whether positive or negative, corresponds to the type of bit transition (positive for 0 to 1 and negative for 1 to 0). This makes EM a more robust and better interpretation of the signal for analysis.

In addition, as EM traces are locally acquired, emanations are spatially dependant,

i.e. they correspond to the respective part of the chip where the measurement equipment is located [108]. This geometrical degree of freedom allows for a more precise understanding of the underlying components which leak information [98]. Furthermore, with the deployment of multiple EM probes across different regions of the chip, the side-channel information obtained becomes richer. Therefore, the attacker gain deeper insights into the system's behavior, leading to more effective exploitation.

Moreover, this thesis and the DeepEM [68] both exploit EM emanations from a SoC-FPGA. However, similar to [58], the leakage exploited in DeepEM arises from the interconnects within the PL, whereas our attack targets the data communication bus inside the PS part of the SoC. The objectives of the attacks also differ: DeepEM is a SCA for the recovery of NNs architecture implemented on the FPGA, while our approach focuses on retrieving input data to the NN inference. DeepEM limits its analysis to EM signatures generated during the execution of each layer, dismissing the EM behaviour before inference as "Read/Write noise". Our work aims to exploit the data transfer stage preceding the inference process, which we hypothesize relates to the Read/Write noise phase mentioned in DeepEM. While this hypothesis cannot be confirmed with certainty, our approach focuses on the EM emanations which emit during the input data transfer on the bus.

Chapter Conclusion

This chapter states an overview of the current state-of-the-art related to HW attacks. Two categories of attack, eavesdropping and side-channel attacks are described. SCAs exploit the physical characteristics of computation to gather information or break the security of system, while eavesdropping attacks focus on intercepting and analyzing communication channel to extract data. Taxonomies are discussed to help understand their scope and diversity. Key papers and journals that are closely related to this thesis are presented and provide readers with insights into the domain of SCAs. The chapter concludes by discussing the innovative aspects of our proposed approach and differentiate it from related works. In the following chapter, the characterization of EM side-channels from SoC buses, identification of zone of leakage and present our first contribution.

Chapter 3

Characterization of Near-field Emanations of a SoC for Data Eavesdropping

Contents

3.1 Notions about EM Emanations and their Sensing in Near-Field 59
3.1.1 Origins of Electromagnetic Emanations 59
3.1.2 Induction Effect of Probe 60
3.2 EM and modern ICs 62
3.3 Probe Characterization 64
3.3.1 Transfer Function 64
3.4 SoC-FPGAs Emanations 71
3.4.1 Localisation of EM Leakage 71
3.4.2 Cartography based on Time-domain Signal 74
3.5 Deducing On-chip Bus from EM Emanations 76
3.5.1 Behaviour of EM Leakage in SoC-FPGA 76
3.5.2 Location of On-chip Bus Wires 80
3.5.3 Hypothesis of Bus Location 82
3.5.4 Validation of Bus Hypothesis 84
3.6 Chapter Conclusion 87

This chapter establishes the foundational work on the role of EM characteristics in modern ICs. It demonstrates how they can be exploited to eavesdrop on data, particularly the ones sent through the conducting wire buses. Current-carrying bus wires emit direct EM emanations, which correspond to the internal data activity of the system. Therefore, by analyzing these data-rich emanations, the attacker is able to retrieve the victim's secret information discreetly. The chapter is divided into five parts as highlighted in Fig. 3.1; (1) fundamentals of electromagnetism, (2) EM & modern ICs, (3) probe characterization, (4) SoC-FPGA emanation and (5) deduction of bus geometry from emanations. Part (1) introduces the basic EM theories and the working principles of near-field probes, along with their practical application to this work. Part (2) examines how radiated EM fields, especially in tiny compact ICs, expose the underlying activities, for instance, data, execution of instructions, and operating algorithm on the hardware. Some probes of interest are characterized for a deeper understanding of their response in part (3). Furthermore, part (4) explores the behavior of EM emanations from SoC-FPGA using cartography. Lastly, part (5) provides a deeper analysis of on-chip bus activities from EM traces and demonstrates how the position of bus lines within the chip can be deduced. This chapter provides the insight and foundational understanding necessary for the two upcoming chapters, where the attack methodologies are developed based on the findings and principles established here.

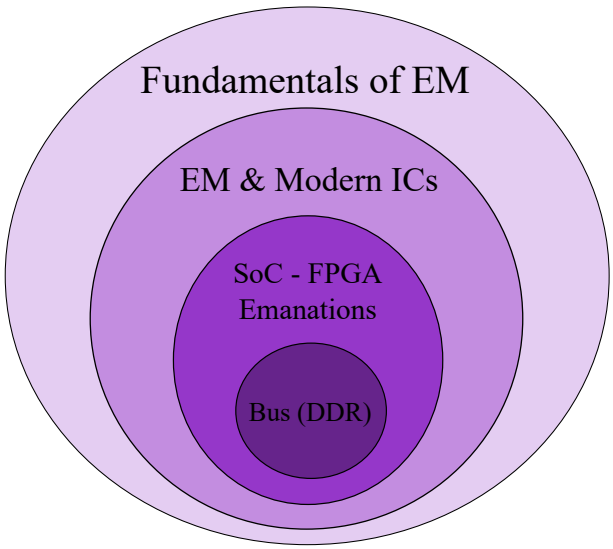


Figure 3.1 – Organization of Chapter 3

3.1 Notions about EM Emanations and their Sensing in Near-Field

3.1.1 Origins of Electromagnetic Emanations

EM emanations arise from charge and current distributions, which act as their sources. According to Maxwell's equations, these sources give birth to electric (E) field and magnetic (H) field. In the immediate proximity of the sources, reactive fields dominate and can be sensed using specialized probes. Specifically, near-field E-field probes detect emanations from charge sources, while near-field H-field probes detect emanations from current sources.

Radiation fields, which occur at further distances, are characterized by intertwined E and H fields that propagate uniformly in the far field, regardless of the source type. However, in cases where radiation is weak or negligible, as in the current study, the sensing is limited to the reactive fields in close vicinity to the sources. Therefore, this work primarily focuses on magnetic field emanations, which are detected using a near-field H-field probe.

In electromagnetism, the magnetic field is generated by constant direct or time-varying electric current. In the case of direct or slowly varying current, Biot-Savart's Law states that the magnetic field at a point along the current-carrying conductor is proportional to the current flowing through that conductor, the length of the wire, the angle between the conductor and position vector, and inversely proportional to the square of the distance from conductor to the point. The mathematical expression of Biot-Savart's Law is expressed as follows:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int \frac{I\, d\mathbf{l} \times (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} \tag{3.1}$$

Where: $\mathbf{B}(\mathbf{r})$ is the magnetic field at point $\mathbf{r}$. μ_0 is the permeability of free space. I is the current. $d\mathbf{l}$ is the differential element of the current-carrying wire. $\mathbf{r}$ is the position vector of the point where the magnetic field is being calculated. $\mathbf{r}'$ is the position vector of the differential element $d\mathbf{l}$. The principles of Biot-Savart's Law are prominently applied in modern technologies such as particle accelerators, antenna design, wireless communication systems, and most importantly in ICs.

Biot-Savart's law is a magneto-static law that describes the magnetic field generated by direct currents. While this law is strictly valid for static currents, it can be extended to

time-varying currents under the condition that the contributions of elementary currents do not exhibit significant propagation (phase) delays at the position where the magnetic field is observed. For fast time-varying currents, the time-dependent behavior of the magnetic field requires the application of full Maxwell's equations for a complete derivation, particularly in the far field where propagation effects become significant. In the near field $l \ll \lambda$, Biot-Savart's law provides an effective approximation, as the reactive components of the field dominate and phase differences are negligible.

Infinitesimal dipole

To understand why Biot-Savart's law is still a valid approximation in the near field for time-varying currents, the situation can be verified using an infinitesimal dipole, also known as a Hertzian dipole. An infinitesimal dipole is an idealized, infinitesimally small antenna element consisting of a short linear conductor carrying an oscillating current. It is the most fundamental model used to describe the radiation of EM waves, characterized by its ability to generate both near- and far-fields, in which the former is dominated by the reactive components. Despite its impracticality, infinitesimal dipoles serve as a building block for more complex antenna geometries and radiating systems. The behavior of infinitesimal dipole in the near-field makes it an ideal system for applying Biot-Savart's law to approximate the magnetic field induced by time-varying currents, as the field's reactive nature and negligible phase differences allow the use of this approximation. (see Appendix 6.3 for a detailed comparison of Biot-Savart's law and the Hertz dipole).

As for the far-field, the EM field produced by an isolated dipole in free space decreases with the inverse of the distance from the dipole. While this behavior is significant for isolated dipoles in free space, it is largely irrelevant in the context of PCBs. In PCBs, the currents flow at the close proximity of ground planes or adjacent wires, strongly attenuating the radiated fields and reinforcing the dominance of near-field effects. Therefore, the radiated field is very small and cannot be detected by the probe.

3.1.2 Induction Effect of Probe

In the (reactive) near-field, the H-field dominates over the E-field for current-carrying loops whereas in the far-field, E and H fields are radiated together as EM wave. An inverse relationship occurs if the source is not a current loop but a charge distribution, the E-field is dominant in the near-field. Here, we focus on magnetic field since the H-field

probe is found to be more sensitive for the particular sources investigated in this work. A near-field magnetic probe (in loop form) interacts with localized H-fields, measuring their strength, frequency, and spatial variations near the source. For an H-field probe, the loop senses the time-varying magnetic flux (the flow of a magnetic field through a surface) passing through its plane. This enables accurate measurement of the H-fields produced by the currents in nearby wires or components on a PCB. In addition, a loop probe also has the ability to isolate specific frequencies of interest as it is sensitive to oscillating H-fields. More importantly, the probe does not require direct contact with the circuit but instead relies on the magnetic coupling effect. It interacts with the H-field formed by the current, inducing a small voltage proportional to the field strength. To fully interpret the electrical signals observed, it is crucial to understand the induction effect of the probe since it determines how the fluctuating H-field is converted into a measurable voltage difference at the the terminal load of the probe.

The induction effect of a using a loop-based probe is the principle by which the probe delivers an electrical signal (voltage, current) in response to the fluctuating H-field detected. This effect is influenced by Faraday's Law of EM Induction, which states that "a time-varying H-field can induce an electric current or voltage in a nearby conductive loop (e.g. a probe)". The probe only reacts to variations in H-field, *i.e.* it functions well with dynamic but not static H-fields. This is because when the H-field close to the probe variates, the field generates a changing flux through the loop, which in turn produces a measurable voltage (emf) according to the equation 3.2.

$$\mathcal{E} = -\frac{d\Phi_B}{dt} \tag{3.2}$$

where $\mathcal{E}$ is the induced electromotive force (emf), Φ_B is the magnetic flux through the loop and $\frac{d}{dt}$ is the derivative of time (change of magnetic flux Φ_B over time). This measurable voltage is directly proportional to the rate of change of the magnetic flux. The faster the changes are, the larger the voltages produced. In high-frequency signals, rapid changes in the H-field are more pronounced (due to the faster oscillations of the clock), resulting in stronger induced voltages proportional to the frequency. For instance, a probe placed near a 500 MHz clock trace is likely to detect a much stronger signal compared to one near a 100 MHz clock trace, simply because the higher clock frequency is associated with faster rising and falling time signals that causes rapidly changing H-fields.

Furthermore, due to the EMF voltage, the current flows through the probe circuit, and the power received at the probe terminal depends on the probe's impedance and

this induced current. The relationship between I_{ind}, P, and emf ($\mathcal{E}$) is defined by the equations 3.3 and 3.4.

$$I_{\text{ind}} = \frac{\mathcal{E}}{Z_{\text{probe}}} \tag{3.3}$$

$$P = I_{\text{ind}}^2 R_{\text{probe}} \tag{3.4}$$

As for the power received at the probe terminal, it depends on the impedance of the probe, which itself varies with frequency. In reference to equation 3.5, the impedance is made up of frequency-independent resistance R_{probe} and inductive reactance $j\omega L_{\text{probe}}$ which relies on the frequency range. This means that the probe's response to different frequencies is influenced by how its impedance interacts with the signal characteristics.

$$Z_{\text{probe}} = R_{\text{probe}} + j\omega L_{\text{probe}} \tag{3.5}$$

3.2 EM and modern ICs

High-frequency operations on ICs produce substantial EM emanations which can affect their overall performance and security. Several electrical and physical phenomena are responsible for the EM leakages on the ICs:

- Switching activity: The multiple simultaneous commutations of logic gate transistors are primarily responsible for EM emissions of IC [116]. During switching events, transistors transition between on- and off-states to perform logical operations or store data. These rapid transitions produce transient currents, leading to voltage variations between the power V_{DD} and ground V_{gnd} planes. In particular, when $pMOS$ and $nMOS$ transistors conduct simultaneously for brief periods (determined by the clock rate), current loops may form in a CMOS inverter gate, depicted in Fig. 3.2, as in the case of a conducting bus wire. These transient currents and current loops are significant contributors to EM emissions.
- Parasitic capacitance: ICs are made up of various wires, metal traces, and components and are segregated by insulating material. The close proximity of these components can bring unwanted capacitance, caused by electric fields formed between them. Besides, the insulating material (dielectric) can contribute to parasitic capacitance as the capacitance is proportional to the dielectric constant. In the

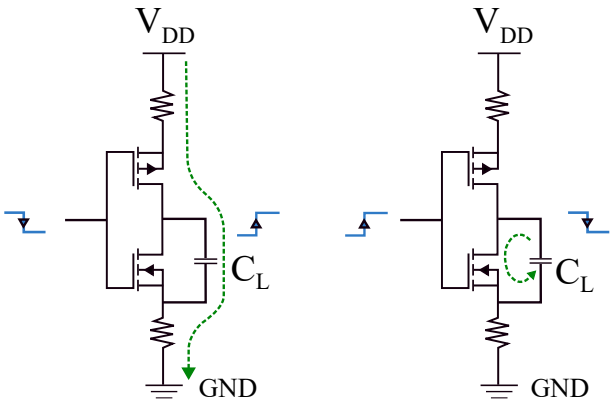


Figure 3.2 – Charge and discharge of a CMOS Inverter

case of bus wire lines, charging or discharging activity can be influenced by current leakage brought on by this parasitic capacitance.

These phenomena provide rich sources of compromising emanations, especially at high clock rates (100 MHz or more). This is because, as stated in section 3.1.2, the higher the frequency, the greater the change in H-fields, hence, resulting in larger induced voltage. Consequently, they increase the likelihood of near-field EM emanations being detected with the use of an adequate H-field probe.

Types of EM Emanations

According to [104], EM leakages, prompted by the ICs can be divided into two types: *direct emanations* and *unintentional emission*. Intentional current flows on the device are the main cause of *direct emanations*. At higher frequencies (100MHz for near-fields), emanations provoked from short bursts of current, are observable at close vicinity. This requires the usage of near-field EM probes, placed in the proximity of the source. Decapsulation of the chip or filtering the acquired signal afterwards can be done as well, in order to improve the leakage energy. *Unintentional emanations* result from the modulation of EM leakage signal and a carrier wave, such as a clock. Current CMOS devices embed complex circuitry, miniaturized wires, and transistors which can promote the risk of coupling between different components if not properly designed. A steady and harmonious clock signal present on the device serves as the carrier signal.

In this thesis, direct emanations, caused by intentional current flows, are specifically exploited while leveraging their observability at higher frequencies for effective EM eavesdropping of target's hardware.

3.3 Probe Characterization

EM probes are essential equipment used to analyze and detect EM fields in many applications, from debugging circuits components on PCBs to performing Electromagnetic Compatibility (EMC) diagnosis. These probes are categorized into two: E-field probes, which detect electric fields, and H-field probes, which are sensitive to magnetic fields. In this work, there are three suitable near-field probes that are designed to measure fields in localized regions with accuracy. Two probes out of three are commercially available on the market, hence, their transfer function information can be found in the documentation from the manufacturer. However, the last remaining one is custom-made in the laboratory by using a tiny loop ($5mm$) printed on a small PCB. Understanding the characteristics of probes, such as their structure (shape), transfer functions and bandwidth is a must for achieving reliable measurements. Therefore, these aspects will be explored in detail in the upcoming sections.

3.3.1 Transfer Function

A transfer function is a mathematical form of a system. It is a critical concept in comprehending the manner in which a probe converts a physical signal (voltage, current, EM fields) to an electrical signal that can be analyzed on measuring equipment such as an oscilloscope or a spectrum analyzer. In the case of a near-field probe, it is the relationship between a local H-field or E-field in the immediate vicinity of the probe and the output voltage of the probe. The transfer function of an H-field probe is mathematically described as:

$$H(f) = \frac{V_{\text{out}}(f)}{B_{\text{in}}(f)} \tag{3.6}$$

where: $H(f)$ is the transfer function of the probe at frequency f. $V_{\text{out}}(f)$ is the output voltage across the loading impedance of the probe at f. $B_{\text{in}}(f)$ is the local magnetic field at f.

Several factors influence this transfer function:

1. **Frequency Response**: Frequency response is the capability of signal acquisition by the probe across various frequencies without inducing distortion or loss. From the perspective of measurement instruments, such as oscilloscopes and probes, it is related to the frequency range of the probe. As the frequency of an harmonic signal source increases,the probe is theoretically more sensitive to the local H-field, but

could be limited by its internal impedance. For the probe to accurately represent the input signal at all frequencies, the transfer function should ideally be flat (constant gain) over the desired frequency range.

2. **Bandwidth**: The bandwidth of the probe defines the range of frequencies over which it performs effectively. At frequencies outside of this bandwidth, the transfer function decreases. At low frequencies, the limitation is due to the low rate of the low time derivative of the flux. In the high-frequency range, it is limited by its impedance and propagation effects when the size of the probe is not too small with respect to the wavelength.
3. **Impedance Matching**: Probes are designed with specific input and output impedance (*e.g.,* 50 ohms). The transfer function is distorted by an impedance mismatch between the probe and the measurement tool or signal source, which results in signal attenuation or reflection.

Langer E-10

For the commercially available Langer E-10 probe, its characteristic can be found in two key waveforms provided by the manufacturer; the frequency response and the E-field correction curve, as shown in Fig. 3.6. The E-field probe senses the time-varying electric field component $E_z(t)$, which is directed upward and perpendicular to the surface as illustrated in Fig. 3.3. According to the data sheet [117], the dimension of the probe head is approximately 0.5 x 2 mm with the resolution of 0.2 mm. The graph in Fig. 3.4 represents the frequency response, illustrating the sensitivity of the probe across its operational bandwidth (30 MHz to 3 GHz). The flat response graph ensures reliable measurements within the specified frequency range. Additionally, the E-field correction curve (in Fig. 3.5) offers a reference for converting the measured output voltage into the corresponding E-field strength at different frequencies. This calibration data is critical for precisely interpreting the captured E-field values. The combination of these two waveforms ensures that the E-10 probe is a decent tool for characterizing E-fields in various electromagnetic environments.

Langer RF R-32

As for the Langer RF R3-2 probe, its characteristic is given by the frequency response graph, H-field correction curve, and current correction curve as shown in Fig. 3.11. According to Fig. 3.7, the probe features a loop at its tip, which is specifically designed to

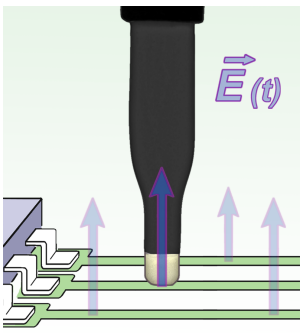


Figure 3.3 – Measuring principles of Langer E-10 probe

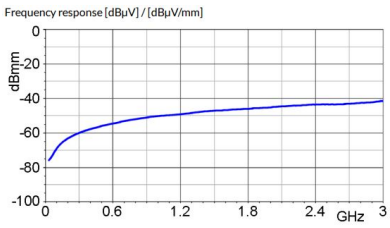


Figure 3.4 – Curve for frequency response of Langer E-10 across its defined bandwidth

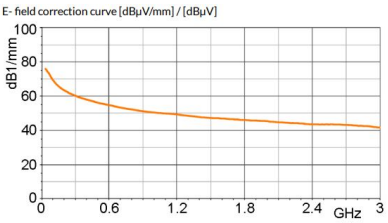


Figure 3.5 – Curve for E-field correction (conversion of induced voltage at the probe terminal to E-field value) of Langer E-10 across its defined bandwidth.

Figure 3.6 – Calibration data provided by manufacturer for Langer E-10

detect respective H-fields. Its sensitivity to the fields is highly dependent on the orientation of the probe, making accurate positioning is crucial for measurements. As stated in the datasheet [118], the probe head measures 3 mm and offers a resolution of 1 mm. The graph in Fig. 3.8 shows the frequency response as a transfer function which indicates the sensitivity of the probe across its defined bandwidth (30 MHz to 3 GHz) and provides the gain of the probe as a function of frequency. The graph is relatively flat over a significant portion of the frequency range, implying a consistent sensitivity across the range. Moreover, the two correction curves for raw H-field and induced current (seen in Fig. 3.9, 3.10) bring additional calibration data to reverse-calculate the measured output voltage into the corresponding H-field or current values across the supported bandwidth. The three waveforms all together offer a comprehensive view of the probe behavior and allow the users to precisely measure both H-fields and current values across the specified frequency range.

Customized Probe

For a custom-made system, the probe response is deduced from physical implementation and computed from the measurements or from a simulation. To study the response of our third probe of interest which is a custom-made probe shown in Fig. 3.12, a few necessary experiments are conducted. The hardware setup for this experiment is described in Fig. 3.13.

A µ-strip wire line printed on a small PCB is used as a HW mockup to imitate the

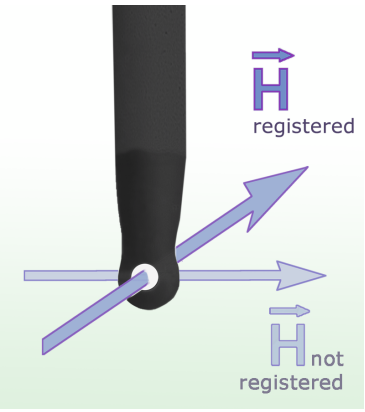


Figure 3.7 – Measuring principles (detectable H-field) for Langer RF R-32

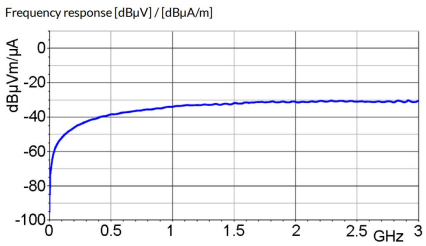


Figure 3.8 – Waveform for frequency response of Langer RF R-32 within its eligible frequency range

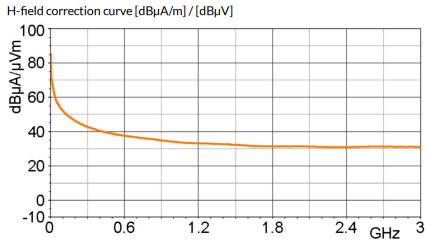


Figure 3.9 – Waveform for H-field correction (conversion of induced voltage at the probe terminal to actual H-field value) for Langer RF R-32 within its eligible frequency range

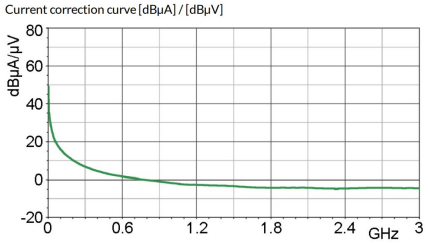


Figure 3.10 – Waveform for current correction for Langer RF R-32 between within its eligible frequency range

Figure 3.11 – Calibration data provided by manufacturer for Langer RF R-32

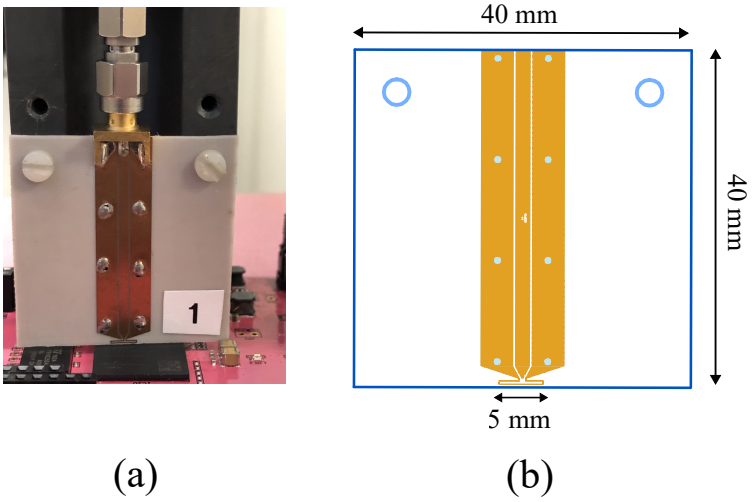
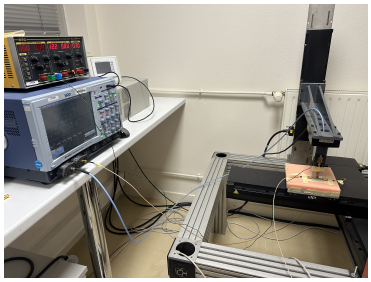
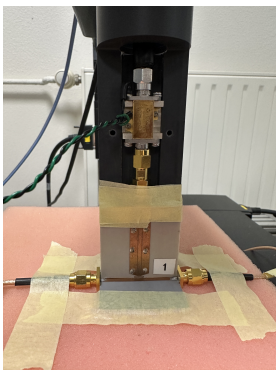


Figure 3.12 – (a) Custom-made near-field probe developed in lab for near-field measurements. (b) Schematic diagram of the probe design, showing its dimensions.

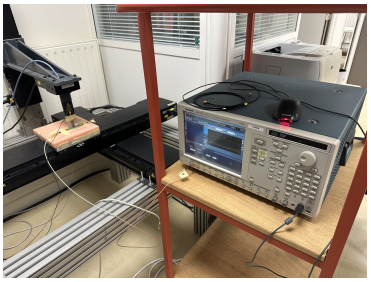
bus subsystem on a SoC. In order to replicate the configuration of a SoC-FPGA bus, a clock wave with a period T of 2 ns (f = 500MHz to mimic high bus frequency) and an amplitude of 0.6V is generated by an Arbitrary Wave Generator AWG7052. The sampling rate of the signal is 5G samples per second. This generated clock signal is sent as an input to the µ-strip while the output of the strip is connected to the oscilloscope. The probe "to be tested" (the customized probe) is placed right on top of the strip line to capture the current derivatives and the probe output is also connected to the oscilloscope as illustrated in Fig. 3.13b.



(a) Part of setup showing the wave generator connected to a thin µ-strip line.



(b) Close-up view of the custom-made probe positioned above the µ-strip line.



(c) Part of setup where the output of µ-strip is attached to oscilloscope.

Figure 3.13 – Experimental setup for the measurement of probe response.

The two resulting waveforms on the oscilloscope, *i.e* one from the output of the µ-strip, and another from the output of probe via LNA, can be seen in Fig. 3.14. It is important to notice that even though the pulse generated by the wave generator is square-shaped, the generator itself has bandwidth limitations that prevent abrupt transitions. This, combined with the attenuation and dispersion in the connecting cable, causes the signal to become slightly deformed by the time it reaches the probe. This deformation affects the sharpness of the transitions, resulting in a less ideal waveform at the oscilloscope. The positive peak of probe output (red) seen in Fig. 3.14a is caused by the rising edge of the square wave clock (green) produced by the wave generator, while the falling edge of the clock induces the negative peak. In Fig. 3.14, it is seen that the delay between the rising edge of the clock and the actual peak of probe output indicates that the two waveforms are not perfectly synchronized. This misalignment is primarily caused by the propagation delay introduced by the difference in the length of the connection cable (1.5m and 2m), which affects the timing of the signal reaching the oscilloscope. These delays are not adjusted, as the experiment is mainly intended to observe the probe's response.

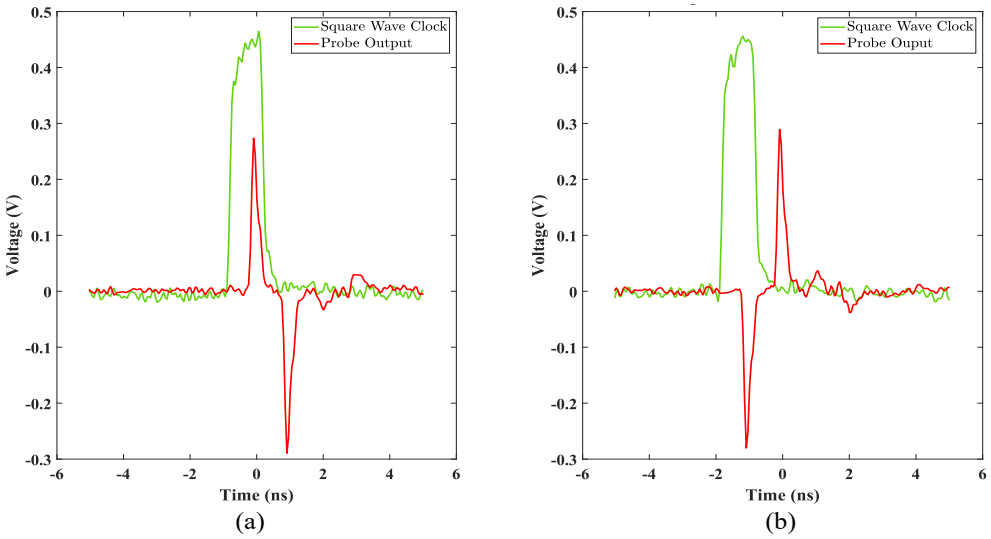


Figure 3.14 – Effect of probe orientation: (a) Normal orientation(0°) with positive peak response representing rising edge of clock, (b) Inverted orientation (180°) resulting in negative peak response for the same rising edge of the clock.

When the probe is turned 180° counterclockwise from the original orientation, the output signal collected at the probe is inverted as well. The signal inversion when rotating the probe to 180° degrees is directly related to the probe's sensitivity to the H-field generated by the microstrip line. In both orientations, loop of the probe remains parallel to the line,

primarily detecting the magnetic field component perpendicular to it. The H-field with respect to different orientations of the probe loop is illustrated in Fig. 3.18. Effectively, rotating the probe changes the direction from which it senses this field component, resulting in an inverted signal. This is why in Fig. 3.14(b), the rising edge of the clock triggers a negative peak instead of a positive (and *vice versa*) which is inverse to what has been seen with the original probe orientation in Fig. 3.14a. From this experiment, it is concluded that the edges of the clock, regardless of the direction (rising or falling), can trigger the voltage peaks on EM signals captured by an H-field probe.

The rising and falling times of a clock signal play a crucial role in shaping the characteristics of the H-field and its induced voltage. Faster rise and fall times cause sharper transitions in the signal, leading to a higher rate of current change ($\frac{dI}{dt}$). According to equation 3.2, a higher $\frac{dI}{dt}$ generates stronger transients in the H-field and thus, greater voltage peaks. On the other hand, slower rise and fall times result in smoother transitions, reducing $\frac{dI}{dt}$. This decline in $\frac{dI}{dt}$ results in weaker transients in the H-field and lower induced voltage peaks. In our experiment, an Arbitrary Waveform Generator (AWG) with a 5 GSamples/second sampling rate is used to produce a clock-like square-shaped signal. Due to this sampling rate, the generated 500 MHz signal exhibited a staircase pattern, inherent to the discretization process. We investigated the impact of varying the rising time (0.2 *ns*, 0.4 *ns*, and 0.6 *ns*) while maintaining a constant falling edge (0.2 *ns*). This resulted in different time derivatives during the signal transitions. Interestingly, despite variations in signal amplitude, the probe's output remained similar for signals with comparable time derivatives, indicating its sensitivity to the rate of change. Furthermore, we observed a decrease in the probe's output response as the rising edge duration increased. Meanwhile, for the falling edge, with its consistent 0.2 *ns* fall time, the probe output remains consistent.

The results from Fig. 3.15 show that when the rise time is 0.2 *ns* it creates an abrupt current change which, in turn, results in a sharp slope with high voltage peak on the output signal. However, with a rise time of 0.4 *ns*, both the peak value and the slope are reduced. When the rise time is further increased to 0.6 *ns*, the peak value decreases even more, and the slope of the output signal becomes noticeably smoother. Due to the underlying discretization of AWG, the rising edges of the generated square waveforms exhibit a staircase-like pattern, even when attempting to create slower rise times. This discretization is a consequence of the finite sampling rate of AWG, which limits its ability to produce perfectly smooth transitions. This limitation is observed in Figure. 3.15, where

the three square wave (SW) clock, despite having different rise times, do not appear drastically different due to the stepped nature of the rising edges.

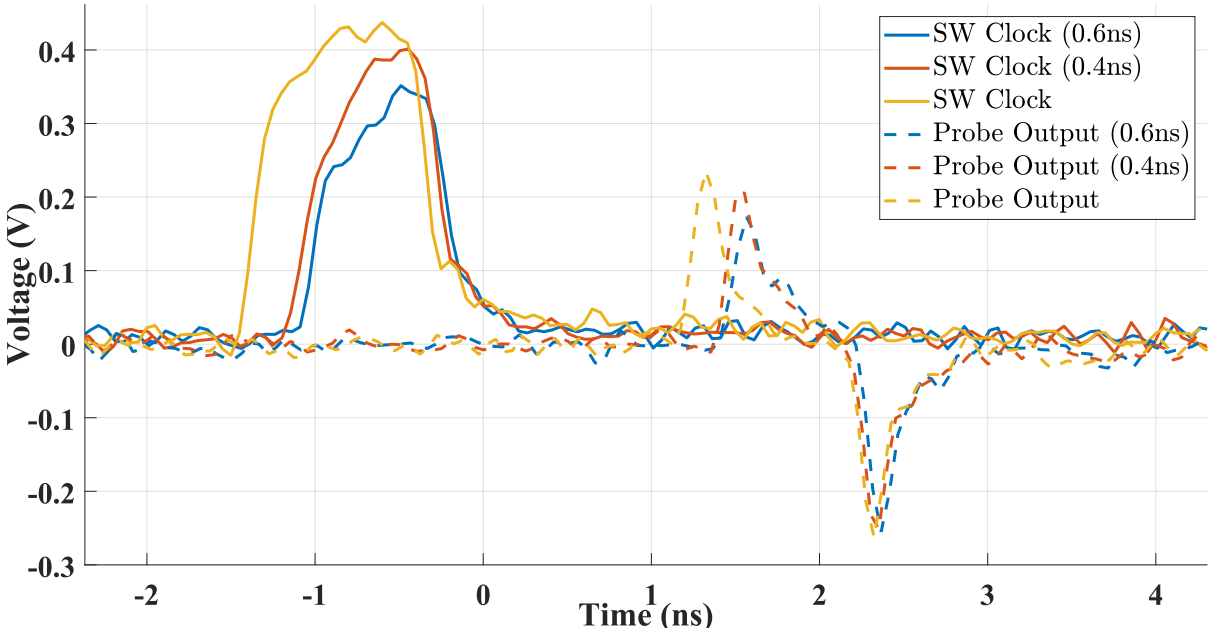


Figure 3.15 – Comparison of square wave clock signals created by wave generator and its probe outputs with varying rise and fall times

3.4 SoC-FPGAs Emanations

The previous section verified that the time-varying current (caused by a high-frequency clock) flowing through a copper strip of wire produces an H-field detectable by a near-field probe. Given that the SoC-FPGAs contains a network of interconnects and generates significant transient currents due to their sophisticated circuitry, we are intrigued to investigate whether exploitable EM emanations can be observed from the interconnects wires inside the SoC. By utilizing near-field probes, characterization of the EM emanations generated by these interconnects is done, opening up ways to exploit the system itself.

3.4.1 Localisation of EM Leakage

The first step in the exploitation of secret information from SoC-FPGA is to capture EM traces from the HW and to locate the zones of significant leakage, also referred to as spatial POI [105]. In pursuance of POI and to better understand the behavior of

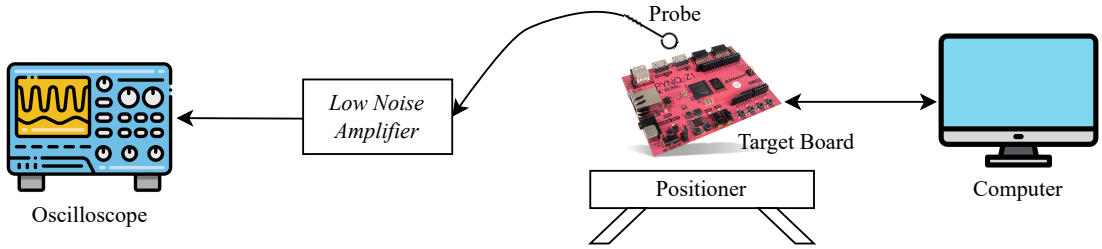
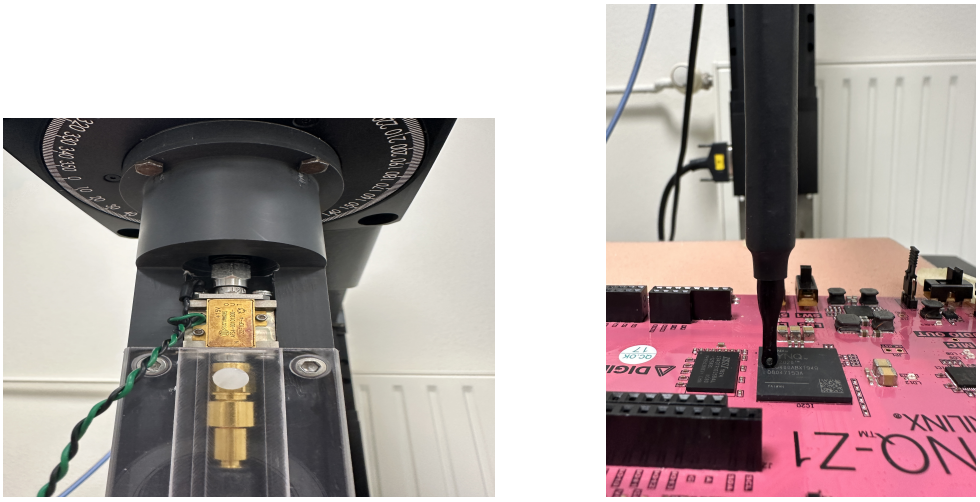


Figure 3.16 – Experimental test bench employed for EM signal acquisition and mapping.

emanations, EM cartographies are made upon the targeted SoC-FPGA, using three near-field probes mentioned before. As explained in section 4.2, it is crucial that the attacker must own a HW replica of the victim so that he can locate the leakage zone on HW level, irrespective of the software application implemented inside FPGA. The designed test bench for capturing and mapping EM emanations on the cartography, is showcased in Fig. 3.16.

The near-field H-probe is placed perpendicularly and immediately above the surface of the SoC-FPGA as depicted in Fig. 3.17b. The probe output is connected to the oscilloscope for signal analysis through the LNA with a bandwidth of 100 MHz - 12 GHz and 30 dB gain as seen in Fig. 3.17a. The field components H_y and H_x are measured for both orientations of the probe (0° and 180°), *i.e.*, when the plane of the probe is aligned with O_x and O_y, respectively. A diagram illustrating the relationship between H_y and H_x with respect to O_x and O_y is seen in Fig. 3.18b,c. For the purpose of mapping the EM activity, the probe is systematically moved across the entire surface of the SoC-FPGA, capturing traces at each (x, y) coordinate. As for the O_z, the probe is fixed at 0.1 mm above the surface of the SoC-FPGA. As the probe's sensing loop is moved vertically away from the surface, the emanations become progressively weaker. The maximum distance at which emanations can still be detected is 1.8 mm. Using a spatial resolution of 0.5 mm, three EM cartographies for each probe of interest are made.

If the data transition on the high-frequency bus (e.g., $525MHz$ for this SoC-FPGA) flips from the lowest possible value to the highest possible value, a strong H-field will be generated, resulting in a higher induced voltage detectable by the probe. In our setup, this concept is utilized by toggling the data value from '0' to '1' across the entire data width. This drastic contrast from all '0's to all '1's across the whole data width ensures maximum data flipping, thereby triggering stronger EM emanations. The concept of maximizing data transitions by flipping bits from the lowest to the highest value is also employed



(a) Low noise amplifier attached at the output of the probe, secured by the arm of the positioner.

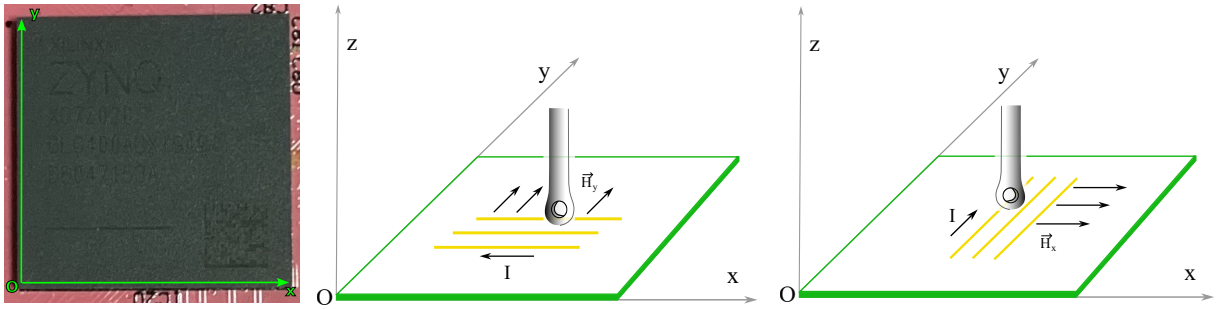
(b) Positioning of the H-field probe perpendicular to the PCB plane, enabling efficient measurement of the emanations.

Figure 3.17 – Location of H-field probe on SoC surface and the LNA used for capturing EM emanations.

in calibration images in Fig. 4.1, as in our application on the FPGA is a MLP, (thus, its input type is an image). This approach is particularly designed to study emanation behavior and will be further elaborated in section 4.4.1 of the next chapter.

While the data transitions between '0' and '1' alternately per clock cycle and is transmitted across the targeted bus, EM traces are collected at each coordinate (x,y) across the surface of the SoC-FPGA (dimension of 1.6 x 1.6 cm). In order to trigger the trace acquisition, the interval trigger is employed on the oscilloscope to detect two consecutive pulses with a time interval of 12.8 ns between them. This data-independent EM signature, characterized by a fixed time interval, is used as a trigger since it corresponds to the data transfer occurring on the bus. The details of this pattern, as well as its role in achieving precise alignment of data traces, will be explained in detail in the next chapter 4.4.2.

The H-field cartography shown hereafter are based on the extraction the maximum amplitude of positive peaks over the time interval of data transmission More precisely, the average of the maximum voltage differences out of all of the positive peaks found in the time-domain signal collected at each coordinate are plotted as a colored heat map. H_x which is the H-field in the direction of O_x, is much weaker compared to H_y. This phenomenon arises from the fact that the highest potential difference values are located along the horizontal axis which resembles the internal bus position oriented in the direction



(a) (b) (c)

Figure 3.18 – (a) Surface of the SoC-FPGA under investigation. (b) Schematic representation of the magnetic field component H_y, generated by current I flowing along the x-direction. (c) Schematic representation of the magnetic field component H_x, generated by current I flowing along the y-direction.

of O_y. H_y also shows more localized POI since the voltage induced at the probe terminal is significant. This can be attributed to the current predominantly flowing along the O_x direction, with additional vertical current flows occurring at the extremities of the bus. These vertical currents, possibly influenced by bonding wires at the edges of the chip, contribute to the observed H_y field. After a series of explorations, the on-chip bus is indeed identified as the vulnerable location where the input-related data activates H-fields, hence, making it an attractive target for potential attackers. The three cartographies are based on each probe and will be discussed in the following section.

3.4.2 Cartography based on Time-domain Signal

Langer E-10

The E-field probe deployed in this experiment is designed to measure the vertical component of the E-field, denoted as E_z. Langer E-10 is used to assess whether the output voltage recorded from an E-field probe could be stronger than that of an H-Field probe. As seen in Fig. 3.19, the cartography reveals that the output voltages detected by the E-field probe are significantly weak. While a small localized zone, represented in dark blue, suggests some leakage that may be associated with the bus, the overall emanations remain minimal. Given this limitation, the focus shifts to using an H-field probe, to better characterize the leakage and investigate more about the H-field emanations produced by the bus wires.

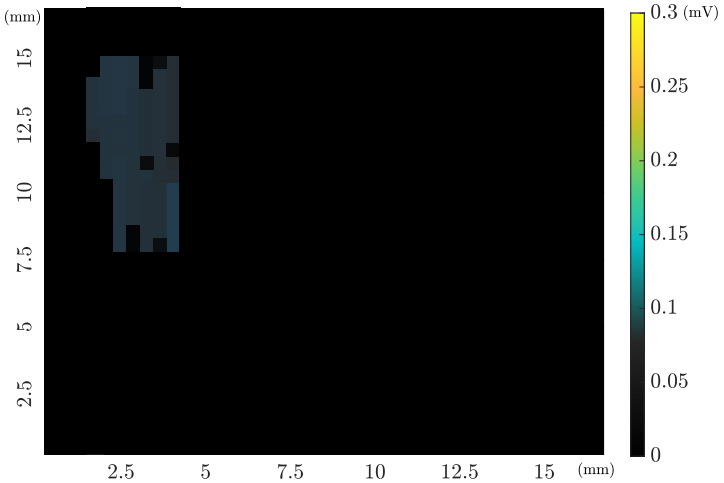
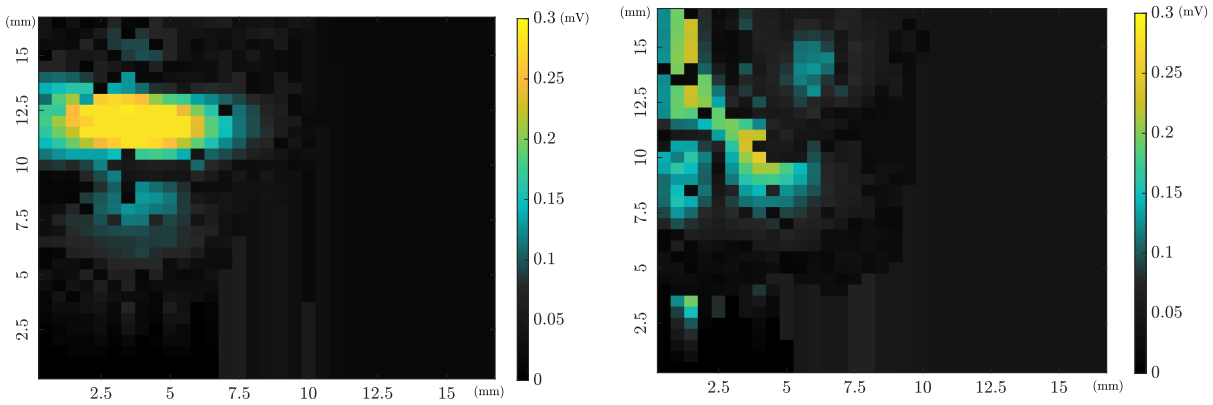


Figure 3.19 – E-field cartography of the SoC-FPGA using Langer E-10. This cartography is mapped by taking time-domain traces at each (x,y) coordinate, where the peak values of the signals are averaged. The averaged peak values are then plotted to create the EM cartography.

Langer RF R-32

Fig 3.20 illustrates the H-field cartographies of the SoC-FPGA, showing the H-field distributions in both the H_y and H_x directions, captured using the Langer RF R-32 probe. In the H_y-field cartography (Fig. 3.20a), a prominent POI is observed as a yellow zone, indicating strong H-field emanations, which is likely associated with the bus wires responsible for bus data transmission. This high-intensity region suggests the presence of significant current activity, with the strongest emanations aligning with specific locations along the bus. Conversely, in the H_x-field cartography in Fig. 3.20b, distinct areas of concentrated emanations are visible, though they appear more scattered and weaker compared to the H_y-field. This is because the probe is more sensitive to H-fields aligned perpendicularly to its loop structure, making it more effective at detecting the H_y field generated by currents flowing in the x-axis direction. The weaker detection of H_x emanations is due to the probe's reduced sensitivity to fields parallel to its loop, resulting in a less pronounced field distribution in the H_x cartography. This dependency on the probe's orientation highlights the importance of probe alignment when characterizing H-fields around the conducting wires. It is also concluded that the sensitivity of the H-field probe is higher than that of the E-field probe.



(a) H-field cartography of the SoC-FPGA for H_y using Langer RF R-32

(b) H-field cartography of the SoC-FPGA for H_x using Langer RF R-32

Figure 3.20 – Cartography (in both H_x and H_y direction) based on the average of maximum voltage peaks of the EM trace during data transmission on bus SoC-FPGA bus.

Customized Probe

The cartography in Fig. 3.21 is generated using the customized H-probe. Similar to the Langer RF R-32 probe, this custom probe captures the H-field emanations from the SoC-FPGA bus wire lines, showing a comparable distribution pattern of EM activity. However, differences in the sensitivity and design of the two probes result in variations in the measured field strength. While the overall shape and location of the high-intensity regions remain consistent, the custom probe exhibits slightly lower signal magnitudes compared to the Langer RF R-32. This highlights the impact of probe characteristics on the detection and measurement of EM emanations.

Based on the analysis of the three cartographies, the Langer RF R-32 probe is selected for further experiments. Its high sensitivity and reliable detection of EM emanations make it the ideal choice for accurately analyzing the emitted H-field generated by the bus wires in the SoC-FPGA.

3.5 Deducing On-chip Bus from EM Emanations

3.5.1 Behaviour of EM Leakage in SoC-FPGA

Evidence suggests that the POI which has been identified in previous cartographies corresponds to data-carrying bus wires. With the aim of verifying the type of bus on

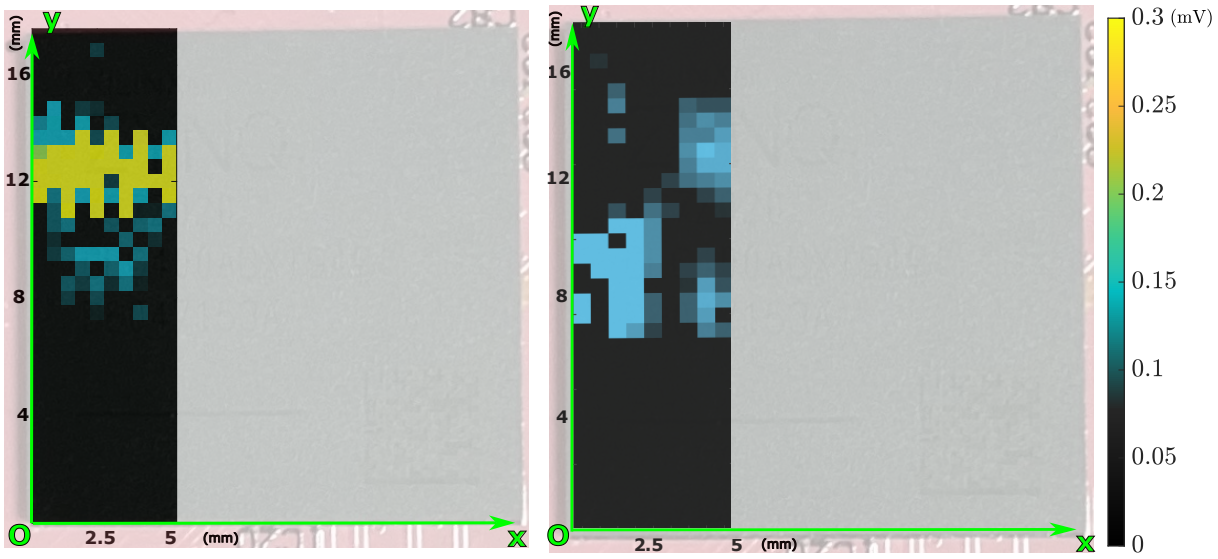


Figure 3.21 – H-field cartography of the SoC-FPGA for H_x and H_y using customized probe.

the SoC-FPGA, a controlled experiment is done. It is known to us that the targeted bus has a data width of 32 bits. The 32-bit data is divided into four blocks of 8-bit data. In this controlled experiment, five cases are considered. In each case, one block (of 8 bits) is set with an "active" state (binary '1'), while the remaining three blocks are all set to "inactive" state (binary '0'). For instance, in case 1, the first block will have an active bit '1' value, while the other three blocks remain '0'. This pattern is repeated in the first four cases, while in the last one (case 5), all four blocks of 8-bit data are set to binary '1'. The illustration of 8-bit blocks and the cases are shown in Fig. 3.22.

	$X_1\ X_2\ X_3\ X_4\ X_5\ X_6\ X_7\ X_8$	$X_9\ X_{10}\ X_{11}\ X_{12}\ X_{13}\ X_{14}\ X_{15}\ X_{16}$	$X_{17}\ X_{18}\ X_{19}\ X_{20}\ X_{21}\ X_{22}\ X_{23}\ X_{24}$	$X_{25}\ X_{26}\ X_{27}\ X_{28}\ X_{29}\ X_{30}\ X_{31}\ X_{32}$
Case 1 →	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Case 2 →	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Case 3 →	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0
Case 4 →	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1
Case 5 →	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

Figure 3.22 – Bit arrangements for 5 controlled cases. $x_1, x_2, \ldots, x_{16}$ on the first line represent the bit position on 32-bit sequence, separated into 4 groups, each consisting of 8-bits. Second to fifth line show how each case put active bits on their sequence.

In Fig.3.23, it is found that case 1 and 2 trigger a positive peak at the same time instance, $t_1 = 160.808$ ns, due to the bit flip from '0' to '1' and ends 0.95 ns later, at $t_2 = 161.758$ ns, where bits in each case are turned from '1' back to '0'. The positive and negative peaks of cases 3 and 4 behave similarly to the two prior cases but with a time shift of 0.95 ns. (Thus, positive peak at $t_2 = 161.758$ ns and negative peak at $t_3 = 162.708$ ns) This 0.95 ns time shift, labeled as t_d in Fig. 3.23, is equivalent to half the clock period of 1.95 ns which matches with $525MHz$ frequency. Finally, in case 5, its positive peak occurs at $t_1 = 160.808$ ns and ends with a negative peak at $t_3 = 162.708$ ns. This means that case 5 (red curve) is the result of the superposition of the first four cases (blue, black, green, and magenta). Notably, case 5 gives the highest voltage amongst all cases, as all 32 bits are flipped simultaneously, whereas in cases 1, 2, 3, and 4, only 8 bits out of 32 bits are transitioned to the active state '1'.

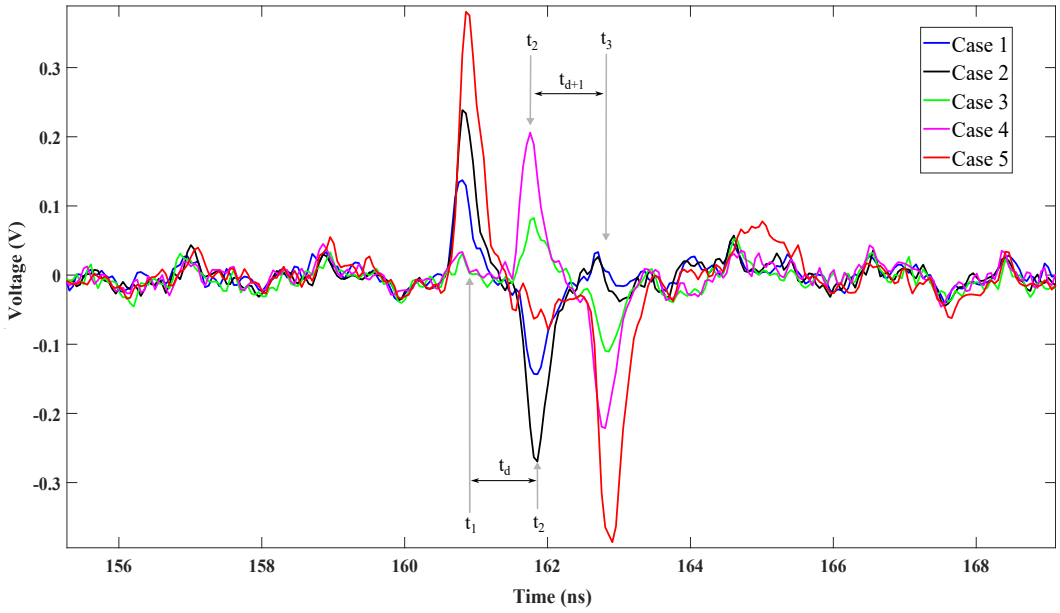


Figure 3.23 – Frequency of the bus is $525MHz$ and the period is $1.9ns$. It is clearly seen that the first half of time instance $t_{\mathrm{d}} = 0.95ns$ is produced by the EM emanation of the first 16-bits (two bytes) of each bus transfer while the second half $t_{\mathrm{d+1}}$ represents the following 16-bits.

Indeed, it is confirmed that the data is sent on each rising and falling edge of a clock which complies with the working principles of a DDR bus. This information not only becomes valuable for figuring out the position of physical bus wires inside SoC but is also important for the image reconstruction algorithm in Chapter 5. Furthermore, while it is expected that case 5 produces the highest voltage peak due to the maximum number of

bit transitions (32), it is intriguing to observe that cases 1, 2, 3, and 4, despite having the same number of bit transitions (8), also shows variations in their peak voltage levels. This variation requires further investigation.

Double Data Rate Transmission

Before diving further into investigating this variation, it is important to first provide an explanation of the DDR protocol to establish a comprehensive understanding. For this reason, the essential foundation of DDR operation will be briefly outlined in this section, providing essential background context. A DDR bus is sampling data not only on rising edges of the clock as a normal bus, but rather on both rising and falling edges. Such buses are widely used in memory communications to transport more data without raising clock frequency. The working principles of DDR in comparison to SDR is highlighted in Fig. 3.24. DDR memories are mostly applied in combination with SoC-FPGA to offer off-chip dynamic random access memory (DRAM) storage, large bandwidth, and efficient data transmission for fast processing tasks. While internal Static Random Access Memory (SRAM) in an FPGA is limited to a few MBytes by nature, DRAM can store GBytes of memory thanks to their dedicated capacitor-based cells and vertical structure. Besides, DDR DRAM memory controller on the FPGA facilitates concurrent access from multiple masters such as Direct Memory Access (DMA) engines and processors. For the architecture of SoCs, DDR controllers are integrated inside the FPGA to interface to the external DDR memory. This DDR controller within SoC manages burst transfers, minimizing latency and optimizing throughput. Various standards of DDR (DDR2, DDR3, DDR4) are supported subject to the model of each SoC platform.

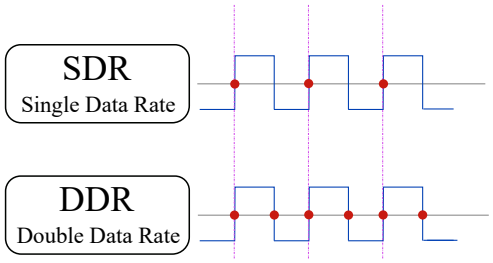


Figure 3.24 – Basic working principles of SDR and DDR operation. SDR transmits data only on the rising edge of the clock signal, while DDR transmits data on both the rising and falling edges, effectively doubling the data transfer rate.

3.5.2 Location of On-chip Bus Wires

From the previous analysis in section 3.5.1, there is variation about the difference in voltage peak even though the number of bit transitions is the same across the total data width. Therefore, additional experiments are required to gain deeper insights and explore this anomaly. To undertake these experiments, first, we retain the concept of separating the 32-bit data width into four 8-bit blocks like in section 3.5.1. Since data is transmitted on both the rising and falling edges of the clock, transitions happen at every 16 bits even though the total data width is 32 bits, hence, we focus on manipulating 16 bits of data per clock cycle. These carefully designed bit patterns, called "bit arrangements", allow us to observe how the positioning of active bits along the bus influences the emanation characteristics. Bit arrangements sent through the targeted bus system are devised and presented in Table 3.1. Not only that, this experiment may also offer a more detailed understanding of the spatial distribution and organization of the bus wires as in modern SoC design, data buses are composed of metal interconnect and traces, carrying currents to transmit signals between different components.

Bit Arrangement	**Configuration**
1	1111 0000 0000 0000
2	0000 1111 0000 0000
3	0000 0000 1111 0000
4	0000 0000 0000 1111
5	1111 0000 0000 1111
6	0000 1111 1111 0000
7	1111 1111 0000 0000
8	0000 0000 1111 1111

Table 3.1 – Different of bit arrangements on 16-bits sequence and its configuration

The 16-bit sequence is separated into four distinct configurations, each designed to activate a specific section of the bus wires. In arrangement 1, the active state (binary ‘1’) occupies the first four MSB, while the remaining 12 bits remain in the inactive state (binary ‘0’). In arrangement 2, the active state is shifted to the next block of four bits, starting from the fifth bit and continuing to the eighth bit, with all other bits set to inactive. Similarly, in arrangement 3, the active state is positioned within the next four-bit block, spanning the ninth to the twelfth bits. Finally, in arrangement 4, the active state occupies the last four LSB, with the preceding 12 bits remaining inactive. According to the resulting Fig. 3.25, it is seen that the orange and yellow curve, which represents

bit arrangement 1 and 4, exhibit the highest and second highest emanation between $t = 160\,\text{ns}$ and $t = 161\,\text{ns}$. On the other hand, for bit arrangements 2 and 3 which are represented by indigo and sky blue curves in Fig. 3.25, the emanation is visibly weaker. This indicates that even when the number of bit transitions between two clock cycles remains the same— four transitions across all four arrangements —there are noticeable differences in the voltage levels observed on the EM trace.

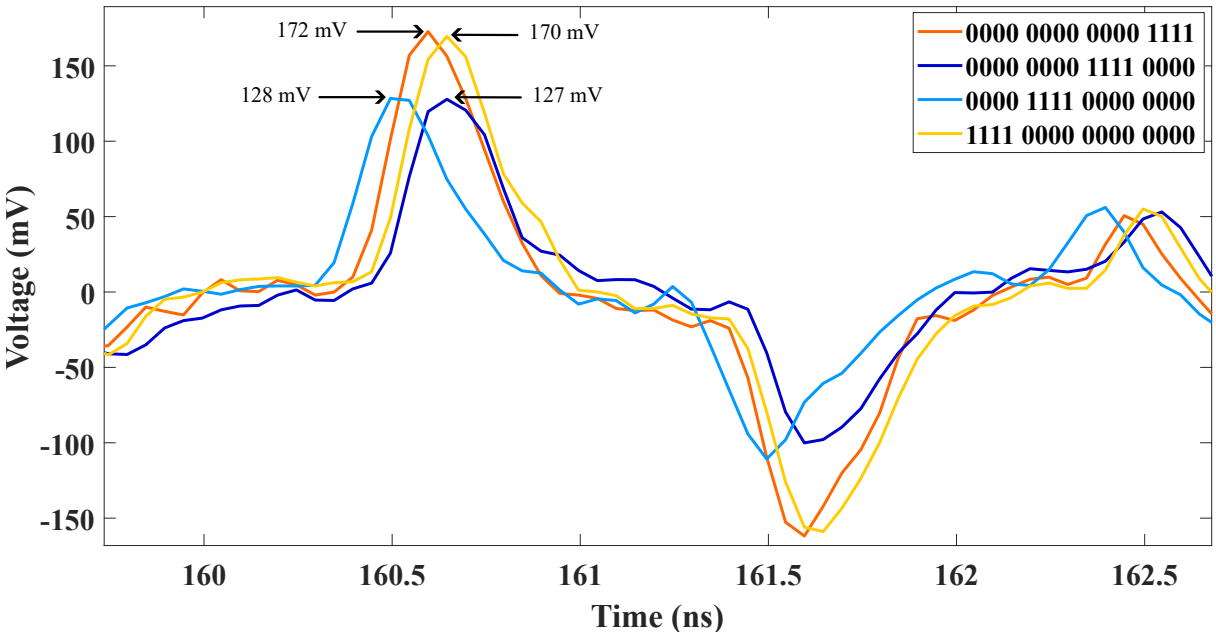


Figure 3.25 – Voltage levels of EM traces for bit arrangement 1,2,3 and 4 with active states assigned to different sections of the bus wires while inducing the same HD.

Compared to the previous 4-bit arrangements, where 4-bits were simultaneously set to the active state, the bit arrangements 5 and 6 increase the number of active bits to 8 within the 16-bits sequence. In arrangement 5, the active bits occupy the first four MSB and the last four LSB, resulting in the sequence [1111 0000 0000 1111]. In this case, the active states are positioned at the extremities of the sequence, leaving the central bits inactive. In arrangement 6, the active bits are shifted inward, occupying the second and third 4-bit blocks of the sequence, resulting in [0000 1111 1111 0000]. Here, the active states are positioned closer to the center of the sequence, while the outer bits remain inactive. The resulting graphs for arrangements 5 and 6 are indicated in Fig. 3.26a. When the active state '1' is placed at the extremities of the 16-bit sequence, which is the case for arrangement 5, the strength of the emanation increases significantly (higher voltage). On the contrary, it is evident that in arrangement 6 the corresponding emanation imposes

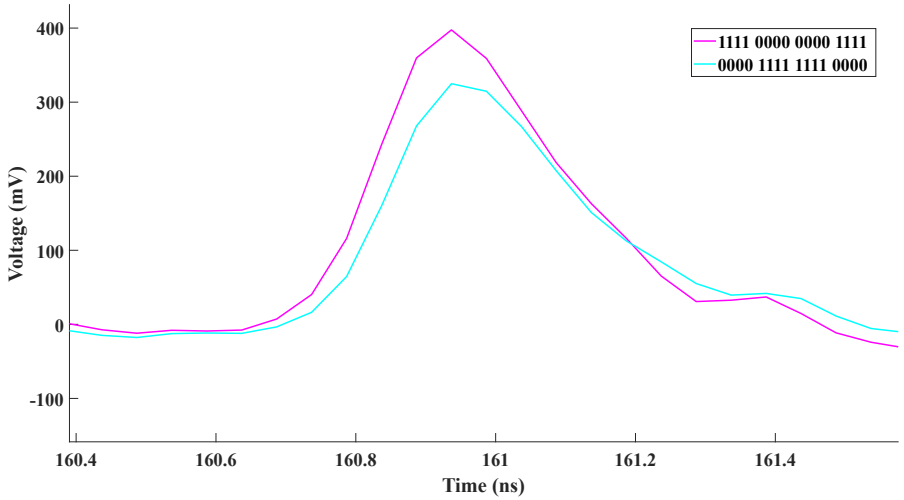
a lower voltage level on the output of the probe. Once again, it is confirmed that the emanation peaks differ based on the arrangement of bits, even when the number of bit transitions remains the same (8 transitions in both arrangements 5 and 6) as previously noted.

In arrangement 7, the active bits are positioned in the first two blocks of the 16-bit sequence, forming [1111 1111 0000 0000], while in arrangement 8, they are shifted to the last two blocks, resulting in [0000 0000 1111 1111]. These configurations feature 8 active bits, akin to arrangements 5 and 6, but cluster them within distinct halves of the 16-bit sequence, granting the ability to examine how the spatial grouping of active bits affects EM signatures. For better understanding, we compare arrangements 5,6,7, and 8 as they all have the same bit transitions of 8. According to Fig. 3.26b, among all four configurations, arrangement 5, where the active bits are positioned at the two extremities of the sequence, induces the highest voltage peak. In contrast, arrangement 6 generates the lowest peak, even lower than arrangements 7 and 8. Arrangements 7 and 8, on the other hand, produce moderate voltage peaks on the EM traces, falling between those of arrangements 5 and 6. Based on this evidence, the hypothetical geometry of the physical bus wires inside the SoC-FPGA is deduced in the following section.

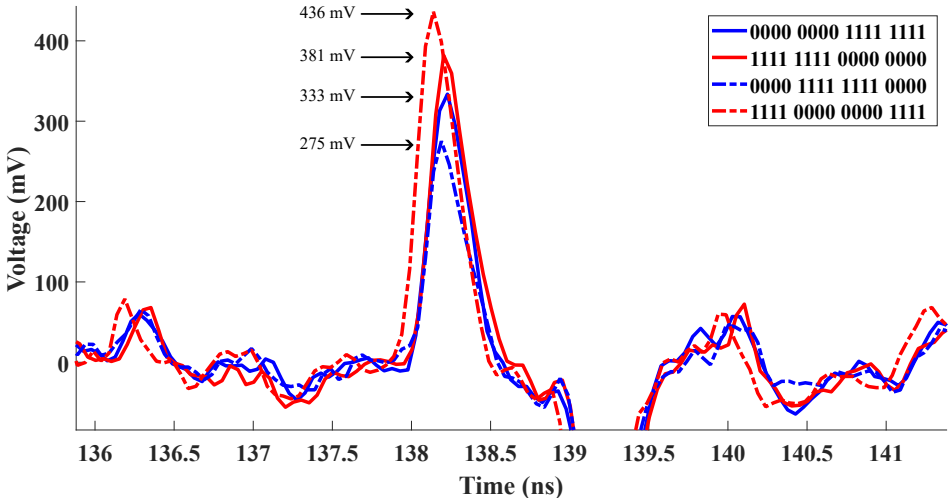
It is worth noting that the voltage peaks are higher (250–300 mV) in bit arrangements 5, 6, 7, and 8 as the number of bit transitions across the entire 16-bit sequence is 8 whereas in arrangements 1, 2, 3, and 4 cause fewer transitions (only 4), resulting in lower voltage peaks (120–170 mV).

3.5.3 Hypothesis of Bus Location

The assessment of results from the first six bit-arrangements out of 8 arrangements suggest that the 16 bus wires are distributed across two layers, with 8 lines on each. The relative physical location of bus wire lines inside the SoC-FPGA is illustrated in Fig. 3.27. The strength of the emanations depends on the physical location of the transitioning wires relative to the probe. By looking at the emanation strength of arrangements 1 and 4 from Fig. 3.25, it is concluded that wire numbers 1,2,3,4,13,14,15,16 (carrying bits X_1, X_2, X_3, X_4, X_{13}, X_{14}, X_{15}, X_{16}) are closer to the loop of the probe compared to the remaining wires. As the emanation of arrangement 4 is higher than arrangement 1, it is well-supported to say that wire 13,14,15,16 is situated on the upper layer which is even closer to the probe. Meanwhile, wires 1,2,3,4 are located on the lower layer which accounts for their slightly lower, but still significant, voltage levels on the output signal. Emanations



(a) Observation of difference in level of EM emanations with respect to MSB and LSB using special grouping of bits. Note that all the traces have the same HD, which is 8.



(b) Results of voltage peaks caused by bit arrangements 5,6,7 and 8, showcasing that the position of bit flip plays an important effect on EM trace signal.

Figure 3.26 – Observation of difference in level of EM emanations with respect to MSB and LSB using special grouping of bits.

from arrangements 2 and 3 prove that the physical wires carrying their respective data bits are located further to the probe in comparison with the first two. Among the two, the emanations induced by arrangement 3 generate lower emanation when compared to arrangement 2, suggesting that it is positioned on the second plane.

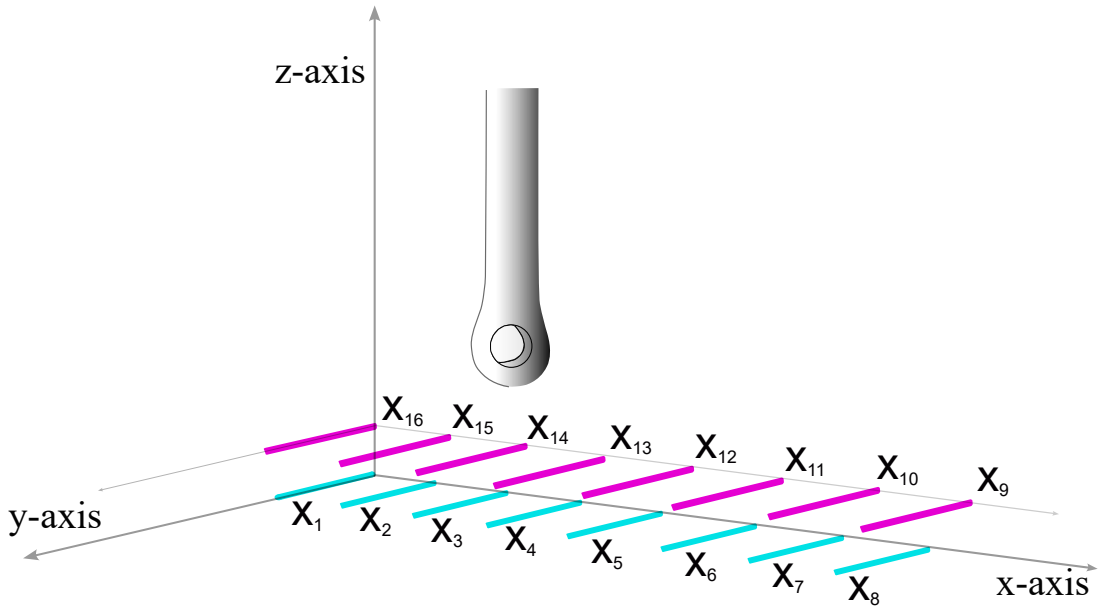


Figure 3.27 – Hypothesis of position of bus wires inside SoC-FPGA

Moreover, the result curves from arrangements 5, 6, 7, and 8 offer additional verification to the hypothetical bus wires located inside the SoC. By switching wires 1, 2, 3, 4, 13, 14, 15, and 16 simultaneously, which are positioned closest to the probe (equal to bit arrangement 5), the emanations are the strongest due to their proximity. In contrast, transitions on wires 1, 2, 3, 4, 5, 6, 7, and 8 are distributed evenly across the lower plane, resulting in weaker peaks. When transitions occur on wires 9, 10, 11, 12, 13, 14, 15, and 16, they still span the entire upper plane; however, since the upper plane is closer to the probe, the emanations are stronger than those from the lower plane. Finally, transitions on wires 5, 6, 7, 8, 9, 10, 11, and 12 are presumed to be physically distant from the probe loop, which is why they produce the weakest emanations out of all.

3.5.4 Validation of Bus Hypothesis

The observations from different bit arrangements propose the hypothetical placement of the physical bus wires within the SoC-FPGA. However, to further reinforce this hypothesis, an additional experiment is conducted. Based on the cartography shown in Fig. 3.20,

the approximate zone of leakage, referred to as the POI is assumed to correspond to a group of bus wires routed within the SoC. As the purpose of this experiment is to double-confirm the bus placement hypothesis, the focus is narrowed to this specific region. The probe was systematically moved along the x-axis while keeping the y- and z- value fixed, covering the entire span of the 16 wires in 0.2 mm steps in this experiment as mentioned in Fig. 3.28.

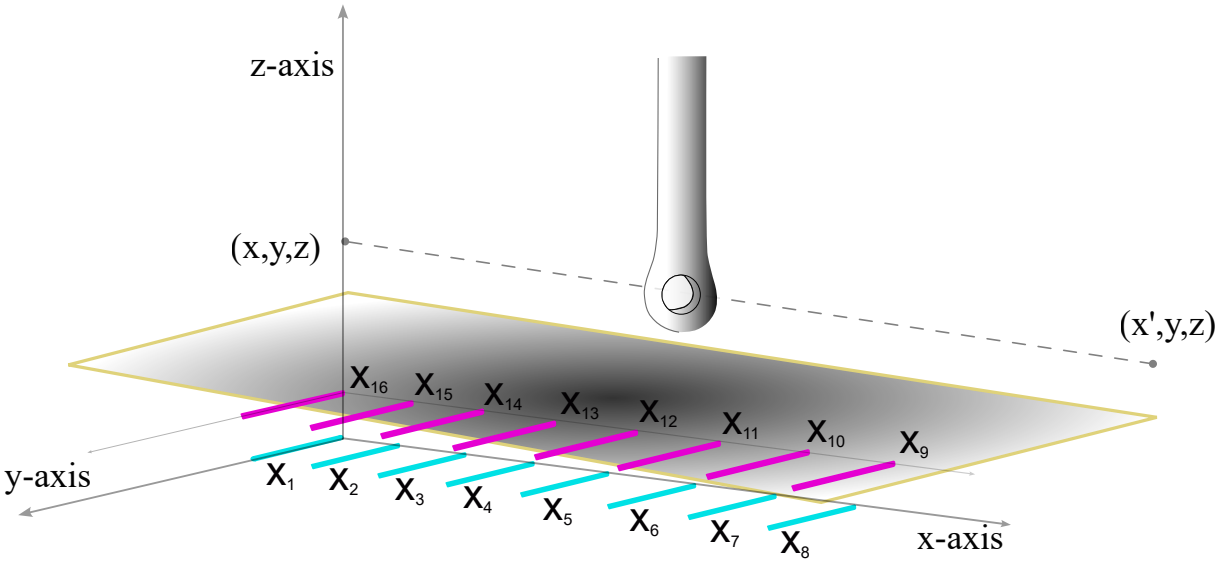


Figure 3.28 – Experimental setup for validating the bus wire hypothesis. The probe is systematically moved along the x-axis, from (x,y,z) to (x',y,z) $0 < x < x'$, where $x' = 2.6$, covering the approximate area of bus wires to observe how the distance between the probe loop and individual wires affects the leakage.

Bit arrangements 5 [1111 0000 0000 1111] and 6 [0000 1111 1111 0000] are selected as input for this experiment. This is because arrangement 6 activates wires 5, 6, 7, 8, 9, 10, 11, and 12, which are clustered on one side of the 16 wires (adjacent to (x',y,z)), while the rest of the wires remain inactive. On the contrary, arrangement 5 induces transitions on wires 1, 2, 3, 4, 13, 14, 15, and 16, which are positioned on the opposite side of the 16 wire buses (located closer to (x,y,z)), with no changes occurring on the remaining lines.

EM traces collected at each position is illustrated in Fig. 3.29. For ease of interpretation, the peak voltage values of each trace are presented in Table 3.2. An analysis of the traces produced by bit arrangement 5 reveals that the induced voltage is highest when the probe hovers directly above its active wires, located between x = 0.6 mm and 0.8 mm. As the probe moves further away from this region along the x-axis, the strength of the emanation diminishes progressively, eventually nearing the noise floor. A similar trend

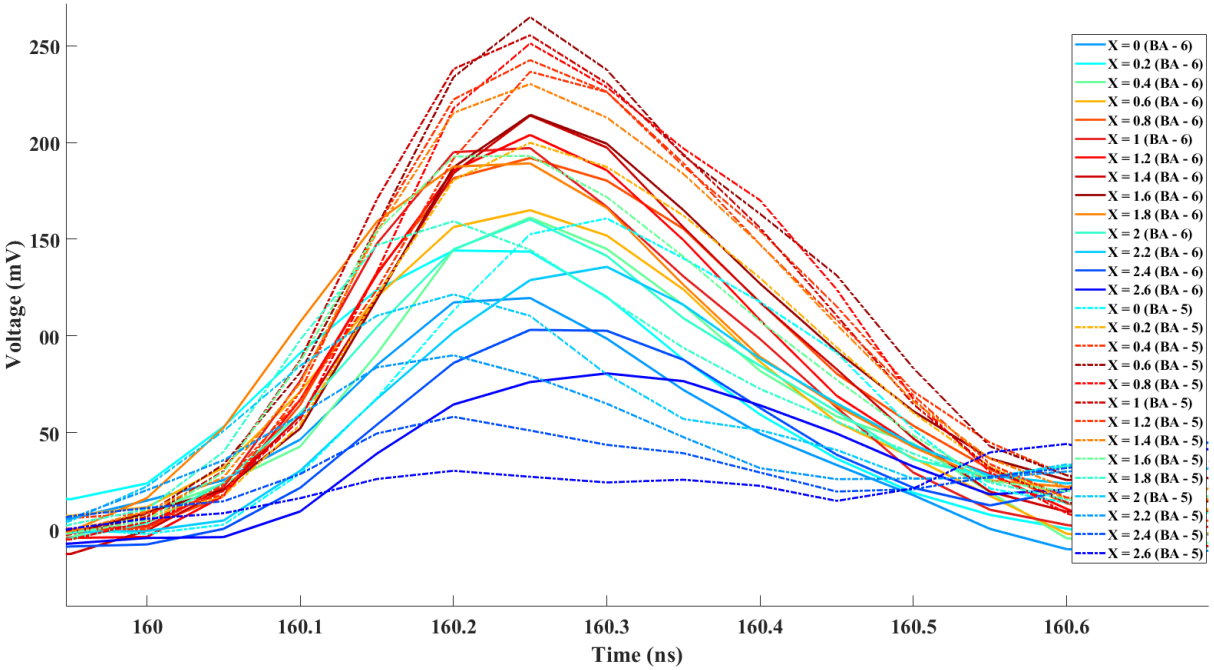


Figure 3.29 – Resulting voltage peaks of EM traces plotted for each probe positions along the x-axis ($0 < x < x'$, where $x' = 2.6$) The observed variation in leakage reflects the changing proximity of the probe loop to specific bus wires, with higher peaks corresponding to regions where the probe is closer to active wires. Prior to 160 *ns*, the bus remained low activity, carrying only bit '0's. At 160 *ns*, bit-arrangement groups are transferred on the bus, triggering transitions between '0' and '1' bits. After this period of activity, the bus returned to its low-activity state, transmitting '0' bits once again.

is observed for bit arrangement 6; however, its active wires are located between x = 1.4 mm and 1.6 mm, as evidenced by the peak voltage observed in this range. Beyond these coordinates, the voltage peaks gradually decrease as the probe moves outward, indicating weaker emanations.

x coordinates (mm)	$V_{\mathbf{peak}}$ **for BA-5 (mV)**	$V_{\mathbf{peak}}$ **for BA-6 (mV)**
0	160	119
0.2	199	144
0.4	236	161
0.6	264	165
0.8	255	193
1	251	197
1.2	242	203
1.4	230	213.9
1.6	193	214.2
1.8	159	189
2	121	160
2.2	90	135
2.4	58	103
2.6	30	80

Table 3.2 – V_{peak} (mV) for BA-5 and BA-6 with respect to x-axis values.

Consequently, this experiment offers a second layer of confirmation for the proposed wire geometry, further supporting the hypothesis of the location of the bus layout depicted in Fig. 3.27. In Fig 3.30, it is clearly seen that the distance between the two ends of the wire cluster is 0.8 mm. Based on our hypothesis, these 16 bus wires occupy approximately 1 mm of space within the SoC. Despite this compact routing, the difference between the two extremes of the wire cluster is detectable using a highly sensitive near-field H-probe, demonstrating the precision of the measurement. Information such as the physical routing of bus wires is a critical and confidential asset for chip manufacturers. This thesis demonstrates that, with the correct choice of instrumentation and setup, valuable insights into the internal geometry of the wires within the chip can be derived from EM emanations.

3.6 Chapter Conclusion

In this chapter, EM emanations caused by modern SoC architectures as side-channels are explored, with their potential vulnerabilities investigated, and their exploitation for

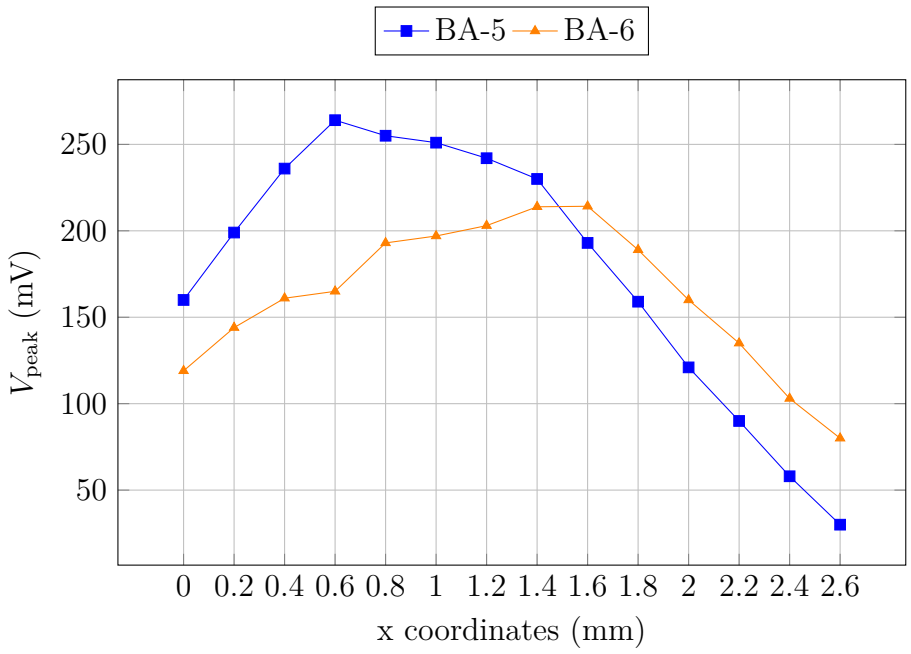


Figure 3.30 – Comparison of V_{peak} (mV) for BA-5 and BA-6 with respect to x-axis coordinates (mm).

data eavesdropping is examined. With the knowledge of fundamental EM principles, the behavior of near-field probes, and the design of on-chip buses, the groundwork has been laid for understanding how EM leakages can expose valuable system information. Key insights of this chapter are;

- **EM as hardware vulnerability**: Distinct EM signatures emitted by data-carrying bus wires contain sensitive information that can be accessed without being invasive.
- **Identification of leakage zone**: With the use of near-field H-probes, the rough area of EM emission on the surface of the SoC is spotted.
- **Findings of bus geometry**: The physical layout of bus wires inside the SoC-FPGA is deduced from captured EM traces of leakage zone, with its strength heavily influenced by wire proximity and internal layer positioning.

Building on these findings, two methodologies are developed using simple image classifying MLP to retrieve important information about the input image sent through the SoC bus. The first methodology, based on the relation between the bus data flipping and the strength of the H-field emitted by the bus, will be implemented and evaluated in the next chapter.

Chapter 4

Hysteresis Binary Image Reconstruction

Contents

4.1 Overview of Attack 90
4.1.1 FINN Framework 90
4.1.2 PYNQ-Z1 92
4.2 Threat Model 93
4.3 Example Use Case of Attack 94
4.4 Pre-attack Phase 94
4.4.1 Calibration Images 95
4.4.2 Pre-amble and Post-amble 97
4.4.3 Detection of True Start Time 98
4.5 Real-time Attack Phase 100
4.5.1 Signal Processing 102
4.5.2 Image Recovery 106
4.6 Results and Evaluation of HBIR Methodology 108
4.6.1 Limitations of HBIR 110
4.7 Chapter Conclusion 110

4.1 Overview of Attack

This chapter introduces a simple and efficient methodology called HBIR for performing SCAs through EM eavesdropping. This newly developed approach enables the extraction of input-related information from SoC-FPGAs by capturing leaked EM emissions from POI. In particular, it targets the conducting bus wires inside the SoC that transfer data bits. By synchronizing the collected EM traces, interpreting them, and reconstructing pixel data, this method demonstrates how EM emanations can be leveraged to compromise input image data from a target SoC platform. The following sections provide a profound examination of the technique, hardware setup, and results that showcase the accuracy of HBIR.

In general, the attack is divided into two main phases: *Pre-attack* phase and *Real-time* phase. In the *Pre-attack* phase, the goal is to identify the EM leaking zones on the surface of the SoC-FPGA and analyze the behavior of the emanations in relation to the data transferred on the bus. The *Real-time* phase then applies these findings, using the principles of the attack on uniquely collected traces to recover the input pixel data in real-time. In the upcoming section, we will introduce the FINN framework and the Pynq-Z1 board to provide a foundational understanding for the readers. With this foundation in place, we will then move on to a step-by-step breakdown of the attack methods.

4.1.1 FINN Framework

The FINN framework, developed by Xilinx, is designed to create fast and flexible binary and quantized neural networks (BNN and Quantized Neural Network (QNN)) accelerators on FPGAs. FINN enables low-latency, high-throughput inference, making it particularly valuable for embedded applications that require efficient, real-time processing without the power or space requirements of traditional GPUs. A key advantage of FINN lies in its dataflow-oriented approach, which customizes both the architecture and precision to maximize efficiency across various FPGA configurations. Using the Brevitas library, FINN facilitates quantization-aware training, allowing neural network parameters to be reduced to fewer bits, often as low as 1-bit for weights and activations, without significant loss in accuracy.

A key feature of FINN is its dataflow architecture, where each neural network layer is mapped to dedicated processing elements, allowing operations to be performed in parallel and stream data directly from one layer to the next. This setup enables sub-microsecond

latency and avoids bottlenecks associated with off-chip memory access by storing model parameters directly on-chip. To maximize efficiency, FINN uses a parameter known as the neuron folding factor, which customizes the degree of parallelism within each layer. These folding parameters dictate the number of parallel PE and single instruction, multiple data (SIMD) lanes, allowing users to optimize resource usage for their target FPGA device. This factor can be adjusted to meet specific hardware constraints, balancing throughput against resource utilization.

The FINN workflow begins with quantization-aware training in PyTorch using Brevitas, a library specifically designed to optimize models for hardware deployment by introducing quantization adjustments. The trained model is then exported in the ONNX format, a widely compatible intermediate representation that FINN imports to build an FPGA-friendly dataflow architecture. In this process, each NN layer is synthesized into a separate IP block, interconnected by streaming interfaces that enable low-latency data transfer. This setup allows FINN to support various NN topologies, from fully connected to convolutional networks, while accommodating a range of quantization precisions for weights and activations. Finally, deploying on FPGAs, especially on PYNQ boards, enables Python-based control and seamless integration into existing software stacks, making it adaptable to diverse application requirements.

Several pre-trained prototypes for classifying benchmark datasets, such as MNIST, Street View House Numbers (SVHN), and Canadian Institute For Advanced Research (CIFAR)-10, are available publicly [119]. The MNIST [45] dataset is a widely used collection of handwritten digit images (0–9), while the Fashion-MNIST dataset, which will be used specifically in this work, is a variation featuring images of clothing items for a more challenging classification task. The SVHN dataset [120] contains real-world images of house numbers captured from Google Street View, making it more complex due to varied lighting and backgrounds. CIFAR-10 [121] consists of low-resolution images across ten categories, including animals and vehicles, providing a more general object classification challenge. These prototypes can be efficiently implemented on popular Xilinx HW, such as Pynq boards, Ultra96, and ZCU104.

For the two methodologies which will be presented in this thesis, MNIST and Fashion-MNIST datasets are used as input images to the NN accelerator. This is because FINN comes with a MLP model with 1-bit quantized weight and activation functions, pre-trained on MNIST dataset. This MLP gives image recognition accuracy up to 93.17% [122]. Fashion-MNIST is a dataset provided by Zalando, consisting of article images organized

into a training set of 60,000 samples and a test set of 10,000 samples. Each sample is a 28x28 grayscale image labeled with one of 10 predefined classes. Designed as a modern alternative to the original MNIST dataset, Fashion-MNIST is intended for benchmarking ML algorithms. Although Fashion-MNIST is not pre-trained with FINN, it can still be provided as input to the existing MLP due to its identical dimensions (28x28) with MNIST. Since Fashion-MNIST has not been trained, it is expected to result in lower recognition accuracy compared to the traditional MNIST. However, as the objective of this work is to eavesdrop on the data bus transmitting inputs to the accelerator, the use of Fashion-MNIST is deemed acceptable, regardless of the final accuracy produced by the accelerator.

The FINN framework is open source, encouraging community contributions and making it accessible for various applications. Future versions are expected to include automated folding and multi-FPGA dataflow, further optimizing performance and flexibility across different hardware setups.

4.1.2 PYNQ-Z1

PYNQ-Z1, developed by Xilinx, is a highly versatile platform that combines SW processing with HW acceleration, making it an ideal choice for applications demanding efficient computation and customization. The core of PYNQ-Z1 is Zynq-7000 SoC, which integrates a dual-core ARM Cortex-A9 processor with programmable FPGA logic. This dual-processing architecture permits coherent interaction between the software-controlled PS and the HW-configurable PL, hence, creating a platform that excels in real-time processing and performance-intensive tasks. The PS and PL, are connected by high-speed communication interconnects available on Zynq-7000 SoC, allowing the two components to exchange data and control. The FPGA fabric in the PL section can be configured to perform HW acceleration for computation-heavy tasks such as signal processing, cryptography, image processing, and machine learning inference. This capability is particularly beneficial for SCA research, where precise control and real-time data handling are critical.

The development environment of PYNQ-Z1 is further enhanced by the PYNQ framework, which provides a Python-based interface for managing FPGA functions. This open-source framework allows developers to load pre-built FPGA overlays or create custom HW configurations directly from Python, removing the need for in-depth Hardware Description Language (HDL) expertise. Additionally, the board supports a wide range of input/output interfaces, including HDMI, USB, Ethernet, and General-Purpose Input/Output (GPIO),

facilitating real-time data acquisition and interaction with external devices such as sensors and cameras. To put it simply, the PYNQ-Z1 merges the flexibility of Linux-based SW processing with the parallelism of custom FPGA logic, making it an excellent platform for research and development in domains such as side-channel analysis, embedded AI, and high-performance computing.

4.2 Threat Model

The threat model refers to a structured approach for identifying and comprehending potential security threats in conjunction with developing mitigation to ensure the protection of valuable assets within the system [123]. In the context of our work, the threat model can be divided into three main branches; *Knowledge*, *Capability*, and *Goal* of the attacker.

Knowledge: The attacker can have a general idea of DNN input image encoding scheme (e.g., grayscale in the case of MNIST and Fashion-MNIST datasets which is used primarily for the proof of concept in Chapter 4 & 5). However, the content of the input image is secret and obviously unknown to the attacker as the objective is to retrieve it. Information on the dimension of the image is not required by the attacker as it can be deduced from the acquired EM signal length over time. Additionally, the architecture, and parameters, such as weights, layers, and neurons of the DNN are unknown to the attacker and are not mandatory to perform our attack.

Capability: The attack is divided into two phases: *pre-attack* and *real-time attack*, and the attack capabilities differ with respect to each phase. During the pre-attack phase, the attacker is required to have access to the same hardware platform as the victim, but the attacker does not have any knowledge of the NN hardware implementation. This phase makes it possible to profile the EM behavior of the device. During the real-time attack, the only requirement of the attacker is to have physical access to the targeted victim device implementing the NN inference in order to collect the EM traces. This prerequisite can be fulfilled in the case of IoT devices, where sensors and smart cameras are usually physically accessible. In addition, the access with a near-field probe can also be conformed by the insertion of HW Trojan in the device package as explained before. Nevertheless, the attacker can neither control the NN inputs, nor manipulate the BNN architecture, nor trigger the inference process at his will.

Goal: The proposed attack aims at deriving sensitive information carried by the input of the BNN inference implemented on the FPGA accelerator, without the knowledge of DNN configuration or the implemented design.

4.3 Example Use Case of Attack

Such attacks can be employed in both, the public and the private sectors, in real-life situations. As a potential critical use case, AI-powered surveillance cameras capture real-time video footage that is subsequently sent to a camera-embedded accelerator where the deep learning algorithm is implemented. The algorithm analyzes the video frames and then, classifies them according to what is being observed by an attacker in each frame. By obtaining physical access to the system, a near-field probe can be installed close to the camera device to capture EM traces. A tiny loop probe, comparable in size to a grain of rice [124], could be planted as a HW Trojan during production at the factory. These tapped surveillance cameras if installed in a secure areas, such as a laboratory or warehouse of a factory, can be used to steal sensitive data processed locally. For medical and military-related businesses, this type of security threat is considered serious as the private data of patients, important manufacturing data or new research findings can be leaked. These use cases open up possibilities for our methodology to be applicable in real-life scenarios at a certain scale.

Now that we have a clear understanding of the HW chosen for our setup, threat model and the example use case, we proceed to examine the actual attack methodology, starting from the pre-attack phase.

4.4 Pre-attack Phase

The primary goal of the *pre-attack* phase is to study how EM prints vary according to the diversity of the information transmitted from the PS to the PL. In an effort to set up this phase, HW replica of the victim device is required by the attacker as discussed in section 4.2. However, it is not necessary for this replica to use the exact same NN model or implementation. In our experiments, the FINN framework by Xilinx is arbitrarily chosen to implement MLP on victim SoC, which is PYNQ-Z1 also manufactured by Xilinx.

4.4.1 Calibration Images

In order to study the behavioral changes of EM print with respect to data transmitted on the bus, four different images carrying different pixel patterns are sent to the NN through the targeted internal bus. These images are named *calibration images* and consist of (a) plain black, (b) plain white, (c) alternating black & white lines, and (d) checker print, as indicated in Fig. 4.1. It is important to emphasize that this procedure is achievable for the attacker as he possesses the same platform as the victim as mentioned in section 4.2.

The purpose of the first two images in Fig.4.1(a) and (b) is to examine the changes in EM activity on the bus under two conditions: *Idle*, when no data is being transmitted, and *Data Transfer*. When the bus is idle, EM activity remains near the noise floor as illustrated in Fig. 4.2. However, when image pixels (shown in Fig.4.1(a) and (b)) are sent across the bus, a distinct pattern, different from the noise floor, emerges as depicted in Fig. 4.1 (i) and (ii). Although the pixel values (0 for white and 255 for black) are present in these two images, there are no transitions between pixels, as each image consists entirely of the same pixel value. As a result, plain black and plain white images produce a constant pattern without any large peaks; however, this signal is still slightly higher than the noise floor. To study how EM emanation behaves in response to pixel changes, the two remaining calibration images are tested.

Fig. 4.1(c) shows an image with alternating black & white lines. At the end of each line, the pixel transitions from black to white (or vice versa on the following line), resulting in a full-range pixel change. When this image is transmitted over the targeted on-chip bus, a large peak appears at the end of each line on the EM signature as represented in Fig. 4.1(iii). This corresponds to the location of pixel color transitions. The interval between successive positive and negative peaks indicates the time required to transfer one row of pixels (approximately equivalent to 13.65 ns), which will be valuable for the image reconstruction process discussed in future section 4.5.2.

The checker print image in Fig. 4.1(d) features an alternating black-and-white square pattern, similar to the arrangement on a chessboard. This checker pattern is designed with a granularity of four pixels per group, alternating black & white values within each group to increase the number of transitions. The internal bus, structured as a 32-bit bus with 32 parallel data lines (though later determined to have 16 lines), processes four pixels at a time, enhancing the pattern's visibility in EM signals by triggering maximum bit transitions as the image is raster scanned and transmitted line by line from left to right. According to Fig. 4.1 (iv), the peaks produced by these maximum transitions are very

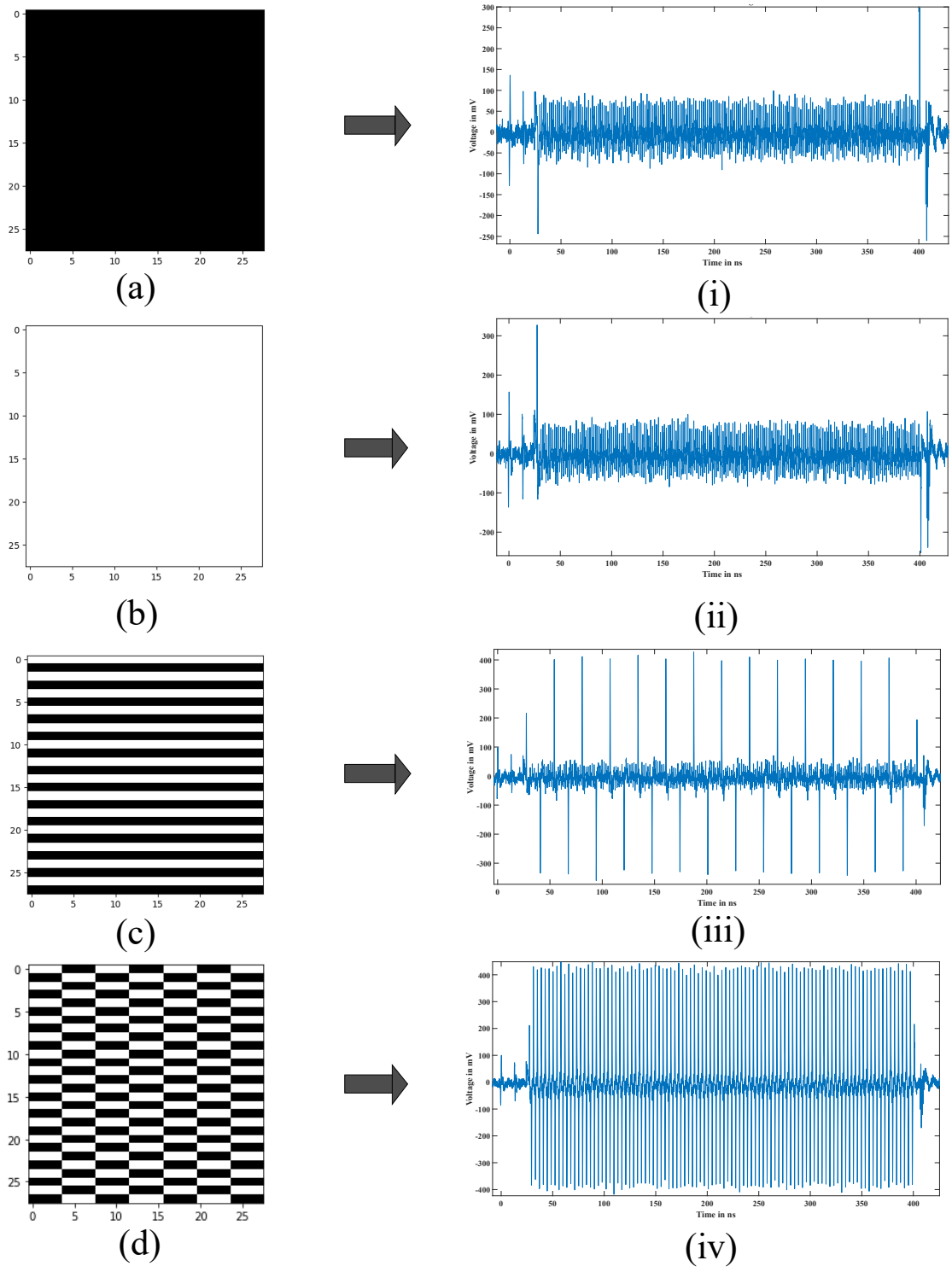


Figure 4.1 – **Left**: Four calibration images sent on the internal bus of SoC-FPGA replica during pre-attack phase (a) Plain Black, (b) Plain White, (c) Black & White Rows and (d) Checker Print. **Right**: Corresponding EM traces generated in response to each calibration image are labeled (i)-(iv) respectively.

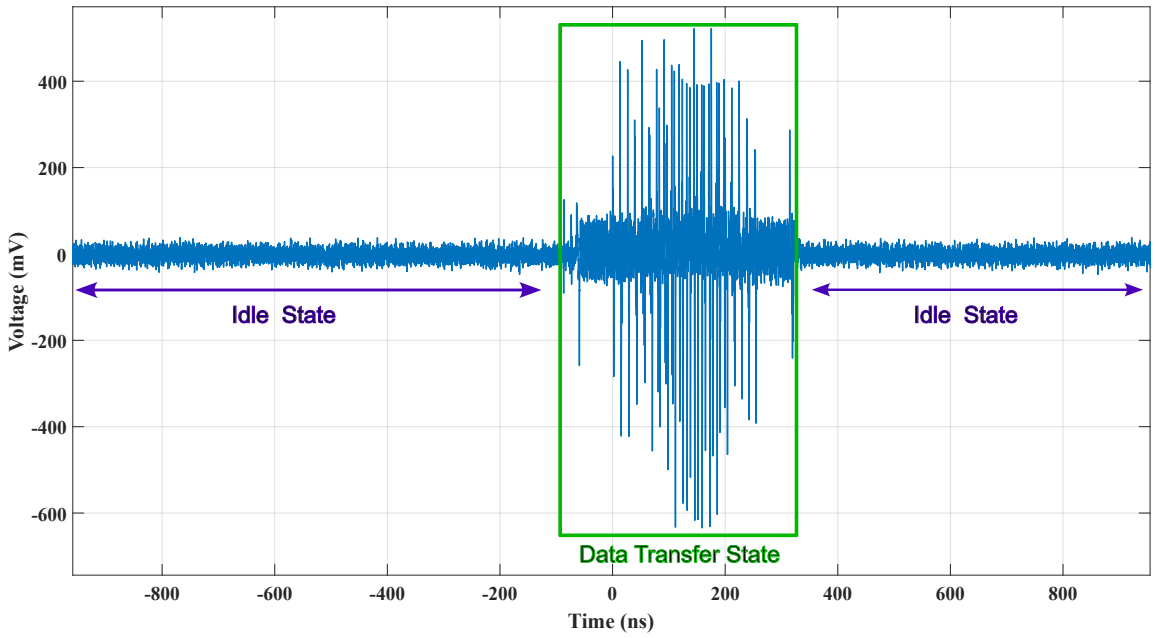


Figure 4.2 – Representation of the difference in the EM signature between when the bus is idle (no data transfer) and when the bus initiates the input pixel transfer for inference inside accelerator.

high, reaching up to 400 mV. The checker print image is primarily used in generating EM cartography as discussed in section 3.4.2.

4.4.2 Pre-amble and Post-amble

To ensure accurate reconstruction of the input image, EM traces emitted from the POI must be captured in a synchronized manner. This is especially crucial for one-shot attacks, where the attacker must perform a precise acquisition of traces that are tightly correlated to the input image rather than to noise. Unlike in most attacks, synchronization in this work is achieved *without any interaction with the inference system*, thus eliminating the need for an auxiliary trigger signal to initiate trace acquisition on an oscilloscope.

In all signals acquired during image transfers over the bus, two distinctive and reproducible signal patterns, termed *Pre-amble* and *Post-amble*, mark the beginning and end of the data transfer as demonstrated in Fig. 4.3. Interestingly, these two patterns, now described in detail, are also be observed in the traces generated by four calibration images shown in Fig. 4.1 (i,ii,iii,iv) from the section 4.4.1, where they were present but not yet explained. These data-independent pre- and post-amble patterns help in synchronizing

traces for each image transfer and are essential for accurate image reconstruction. To detect the positions of the pre- and post-ambles within a trace, correlation is used.

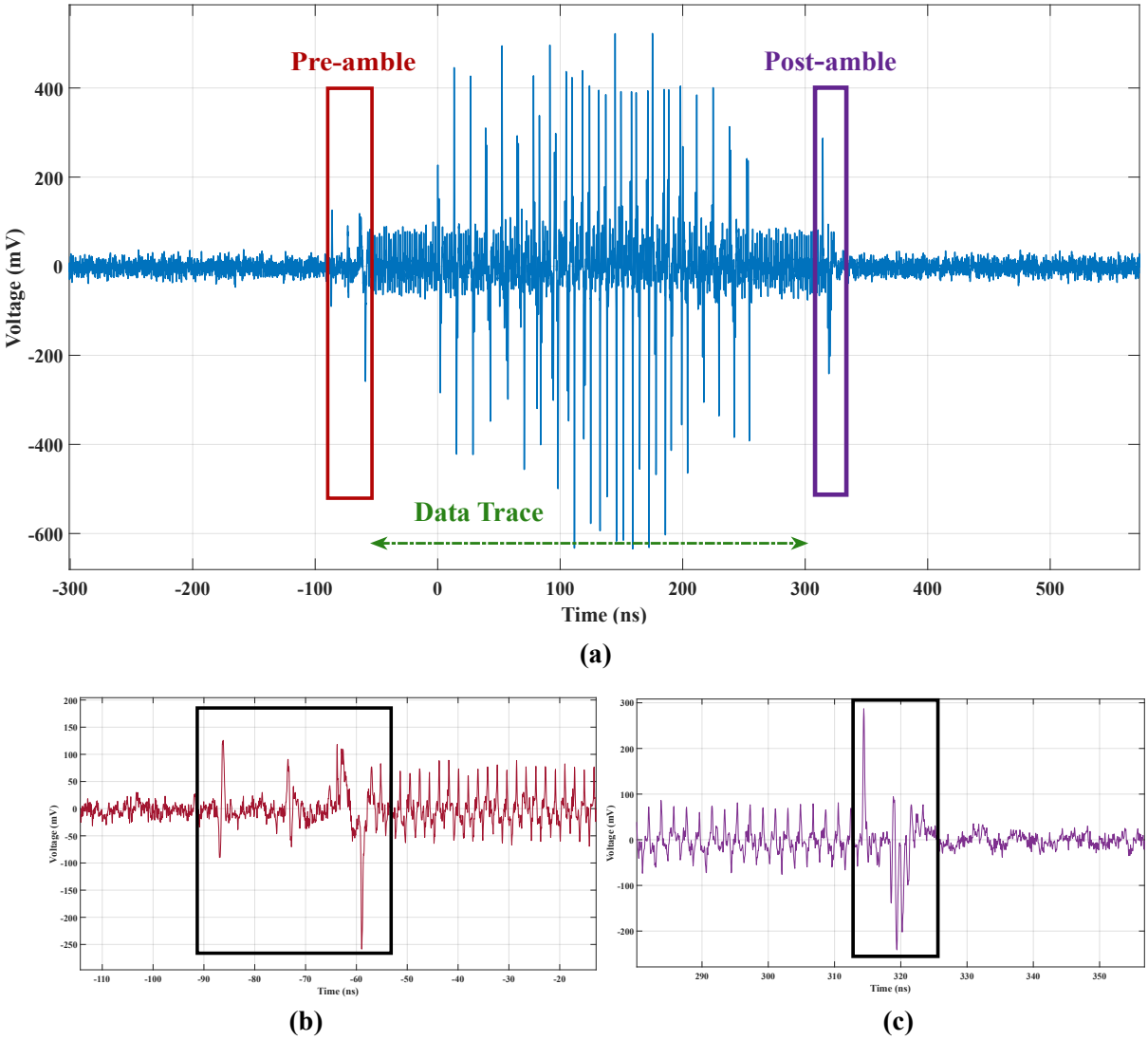


Figure 4.3 – (a) Representation of the EM signal, highlighting the pre-amble, data trace, and post-amble sections respectively. (b) & (c) Zoomed-in views of the pre-amble and post-amble parts of the main EM signal shown in (a).

4.4.3 Detection of True Start Time

In this experiment, the goal is to determine the start time of the data trace captured from the oscilloscope with high precision. Due to the nature of one-shot acquisition (no averaging of traces), an accurate starting time is required to be fulfilled to align perfectly

with the data. Consequently, the exact starting point of the trace had to be distinguished using *preamble* embedded at the beginning of bus data transmission. The *preamble* is utilized as a known reference to locate the precise starting point through a correlation-based process.

Correlation, which is a technique to find a statistical relationship between two signals, is applied by sliding the *preamble* pattern across the collected trace. A similarity score is computed at each time index using the Pearson correlation coefficient. This coefficient quantifies the linear relationship between the *preamble* and a segment of the trace of equal length, with values ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation). The process involved calculating the correlation coefficient for every index offset, resulting in a correlation coefficient versus index offset plot as seen in Fig. 4.4. The index offset with the highest coefficient tells the almost exact time in the trace where the *preamble* achieves the closest match, marked by the red circle in Fig. 4.4. This time instance, called as **peak correlation time** in Fig. 4.5 marks the beginning of the *preamble* within the trace.

However, **peak correlation time** only determines the start of the *preamble*, not the actual beginning of the data transmission. Since the duration of the *preamble* is known (27 ns), the actual starting time of the data trace was determined by adding the *preamble* time duration to **peak correlation time**. The resulting value, named as **true start time**, represents the precise starting time of the data transmission. The signals are illustrated, with both peak correlation time and true start time marked with respective colors in Fig. 4.5 to indicate the actual beginning of the data transfer. Once the true start time is identified, the raw EM trace is sliced into the data-only portion, which is then used for further processing.

At the end of the data transmission, a *postamble* is utilized to indicate the conclusion of the trace. Similar to the *preamble*, the *postamble* is a predefined, data-independent pattern embedded at the end of the transfer. Its purpose is to provide a reliable way to pinpoint the end of the bus data transfer, even in the presence of timing uncertainties or noise. *Postamble* pattern is correlated with the captured trace, sliding across the signal to compute a correlation coefficient at each index. This sliding correlation quantifies the similarity between the *postamble* and segments of the trace. The highest correlation coefficient indicates the position where the *postamble* matches most closely with the signal, identifying the **true end time** of the trace. In contrast to the true start time, where the duration of the *preamble* ($t_{\text{preamble}} = 27\ ns$) has to be added to the correlation peak time,

no such adjustment is required for the *postamble*. The position of the highest *postamble* correlation coefficient directly corresponds to the true end time of the transmission. An alternative way to determine the *postamble* is by using the known total duration of the bus data transfer ($t_{\text{total transfer}} = 374\ ns$). By adding this total time to the true start time, the end of the data trace can be estimated without relying on *postamble* correlation.

This approach provided a consistent and precise method for the synchronisation of one-shot traces and is particularly effective for cases where averaging is not applied, ensuring accurate alignment for subsequent analysis. While the *preamble* helps define the beginning of the trace by accounting for its duration, the *postamble* directly marks the end, making the correlation process a robust solution for synchronizing and analyzing oscilloscope-acquired traces.

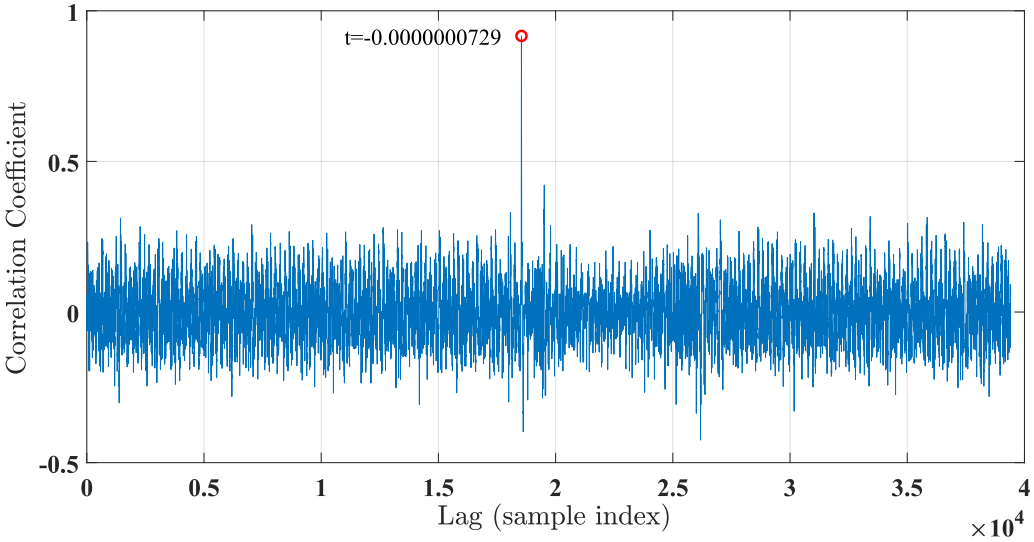


Figure 4.4 – Linear correlation between trace and preamble: It illustrates the correlation coefficient as a function of lag (sample index). The distinct peak in the time domain indicates a strong match between the trace and preamble at that index, highlighting the temporal alignment. The surrounding noise reflects minimal correlation at other lags, underscoring the uniqueness of the detected peak.

4.5 Real-time Attack Phase

Thanks to the information gathered from the pre-attack phase, the attack is now applicable to the actual victim system in at real-time scenario. By putting the probe in

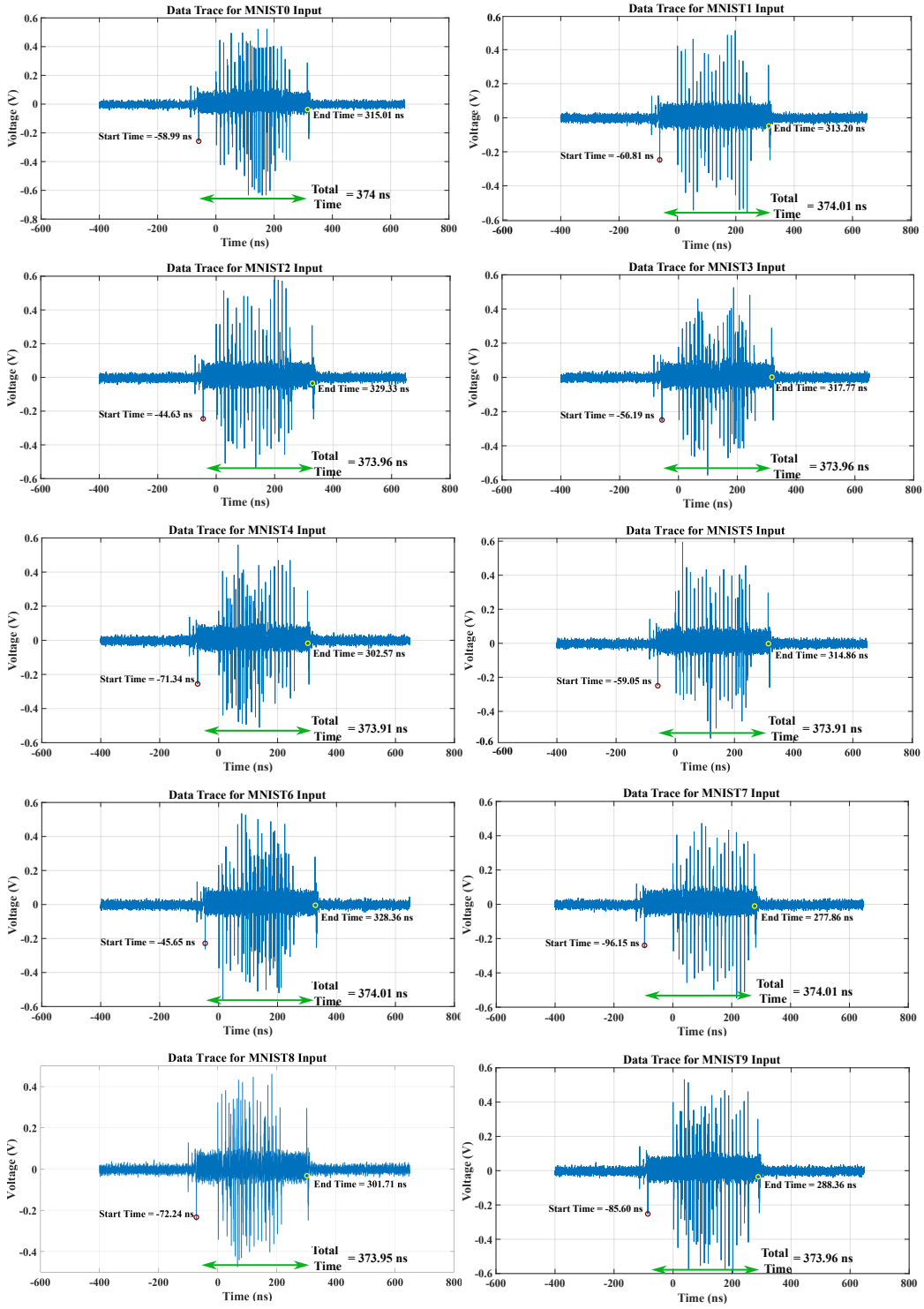


Figure 4.5 – Correlation-based detection of True Start and True End Times for data traces of each MNIST digit. **Note**: In the figure, the terms "Start Time" and "End Time" are used for brevity. These correspond to the "True Start Time" and "True End Time" as defined in section 4.4.3.

POI, EM emanations are collected using the HW setup shown in Fig. 4.6. The chosen target system, Pynq-Z1, is placed on the positioner, while the near-field RF R-32 probe manufactured by Langer (30 MHz to 3 GHz bandwidth) is attached to its mechanical arm. This allows the stable and precise acquisition of emanations. Similar to the setup mentioned in section 3.4.1, the output of the probe is connected to the oscilloscope through a LNA. A power supply unit is required to operate the LNA, which features a bandwidth of 100 MHz - 12 GHz and 30 dB gain. Finally, the purpose of the laptop is to control both the positioner and the oscilloscope, while accessing Pynq-Z1 via Jupyter Notebook. Trace collection is conducted in strict synchronization, facilitated by a *pre-amble* signal pattern as described in the previous section 4.4.2. With the collected EM signals, the attacker can attempt to reconstruct the target image, exposing part of the content of the original input image. To achieve this, signal processing and an image reconstruction algorithm are necessary.

4.5.1 Signal Processing

Noise Characterization

Several peaks appear in the data-activated EM traces, playing a crucial role in interpreting the trace. The nature of these peaks, whether positive or negative, contains exploitable information that reveals input data from the victim. Therefore, distinguishing the true peaks from the noise floor is essential to the success of this attack. These thresholds ensure precise data discrimination from noise and decrease the probability of false detection.

To determine appropriate thresholds for peak identification, a Gaussian distribution curve is plotted using the trace obtained from the transmission of a plain black image. It is well-understood that high peaks in the trace are caused by transitions in pixel values, *i.e.*, changes from one color to another within the image. Since a plain black image contains no transitions (all pixel values are zero), its trace remains constant and does not generate such peaks. Instead, the resulting trace reflects the baseline behavior of the system, mimicking the black background in the absence of active (non-zero) pixel values. This makes the plain black image an ideal reference for analyzing the noise floor and setting thresholds to distinguish meaningful peaks from noise. This approach provides a statistical model for separating significant, information-carrying peaks from background noise.

Before applying a Gaussian fit, the empirical probability distribution of the trace

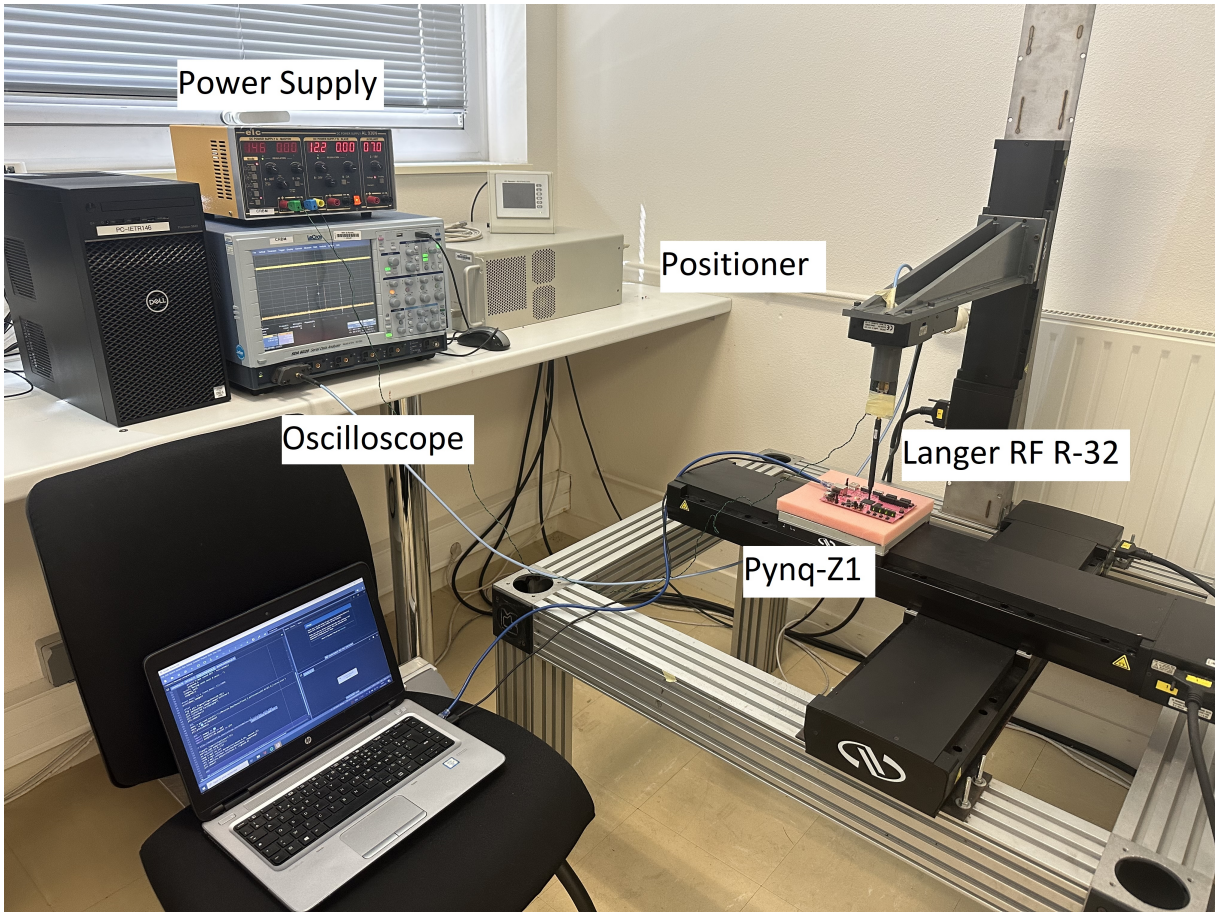


Figure 4.6 – The experimental setup of EM trace collection from SoC-FPGA (Pynq-Z1 as target)

is analyzed to understand its behavior and is done by charting a histogram, as shown in Fig. 4.7. This histogram exhibits that, while the central portion of the distribution features a Gaussian shape slightly not centered, the tails deviate from symmetry. Despite such deviations, the Gaussian fit is chosen to define the noise threshold, as it provides a practical and effective approach for separating true peaks from the noise floor. The underlying reason for the heavier lower tail remains unexplained, but the Gaussian fit still offered valuable insights into threshold determination. The thresholds are empirically set by taking three times the standard deviation (corresponding to the 99th percentile), resulting in a positive threshold of 64 mV and a negative threshold of -72 mV. Thus, any signal with voltage lying beyond the positive or negative threshold will not be interpreted as noise with a 99% confidence interval. This noise floor remains constant across the entire SoC surface area, ensuring a reliable baseline measurement regardless of spatial variation.

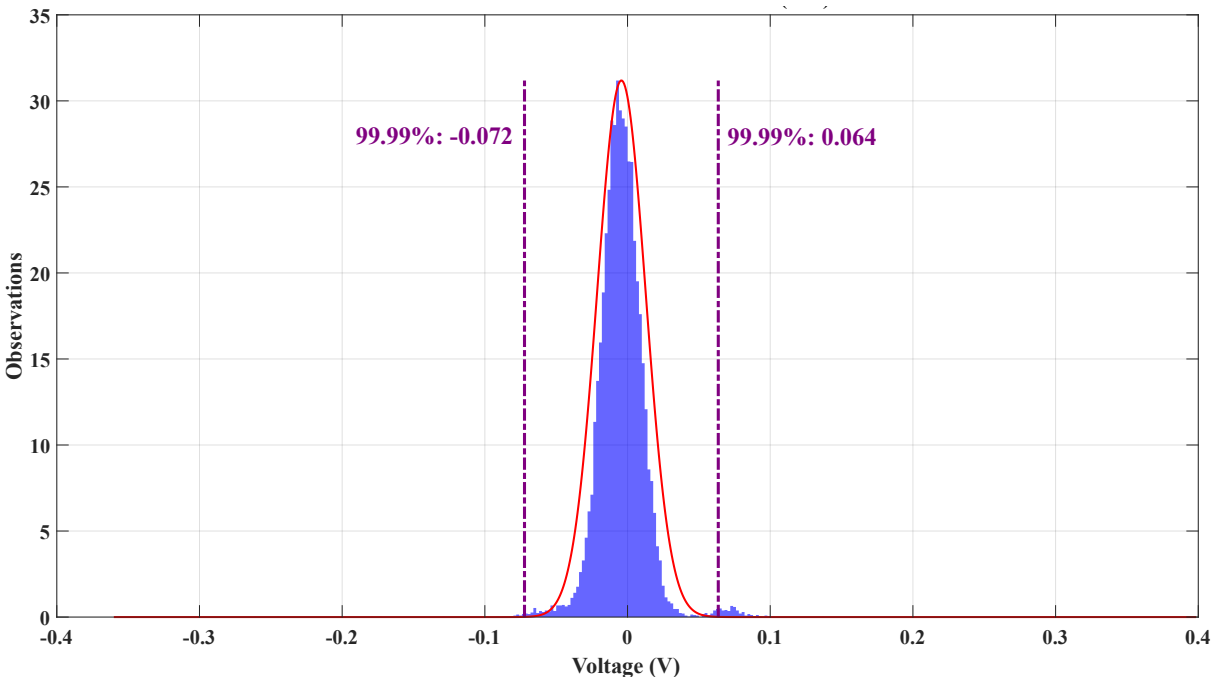


Figure 4.7 – Gaussian distribution curve using EM emanation, produced by the bus transmission of plain black image as an input to the inference.

Hysteresis Binary Image Reconstruction

The peaks in EM trace, positive or negative, represent the HD between two continuous groups of P pixels sent on the bus. Fig. 4.8 illustrates the relationship between induced HD and the resulting EM emanation. To generate this table, we adjust the input bit patterns to trigger specific HD values on the bus. Each data point represents a distinct experiment where the bus transitioned from all '0' bits (at *t-1*) to an active state with a specific bit pattern designed to induce the desired HD (at t). The corresponding EM emanation is measured, resulting in the voltage values shown on the y-axis. This experiment allows us to map the variation in EM emanation as a function of the HD. It is seen that the greater the peak magnitude, the larger the HD value. Additionally, the sign of each peak—whether positive or negative—indicates the nature of the bit transition between the pixels group ($0 \rightarrow 1$ for *ascending* and $1 \rightarrow 0$ for *descending*). Therefore, if the bit values of the current pixel group sent in clock cycle t are higher than the bit values of the previous pixel group at previous clock cycle, *t-1*, it will cause a positive peak representing the *ascending* transition among the P bits sent. Likewise, if the bit values of the previous pixel group sent at *t-1* are higher than the bit values of the current pixel group at t, a negative peak

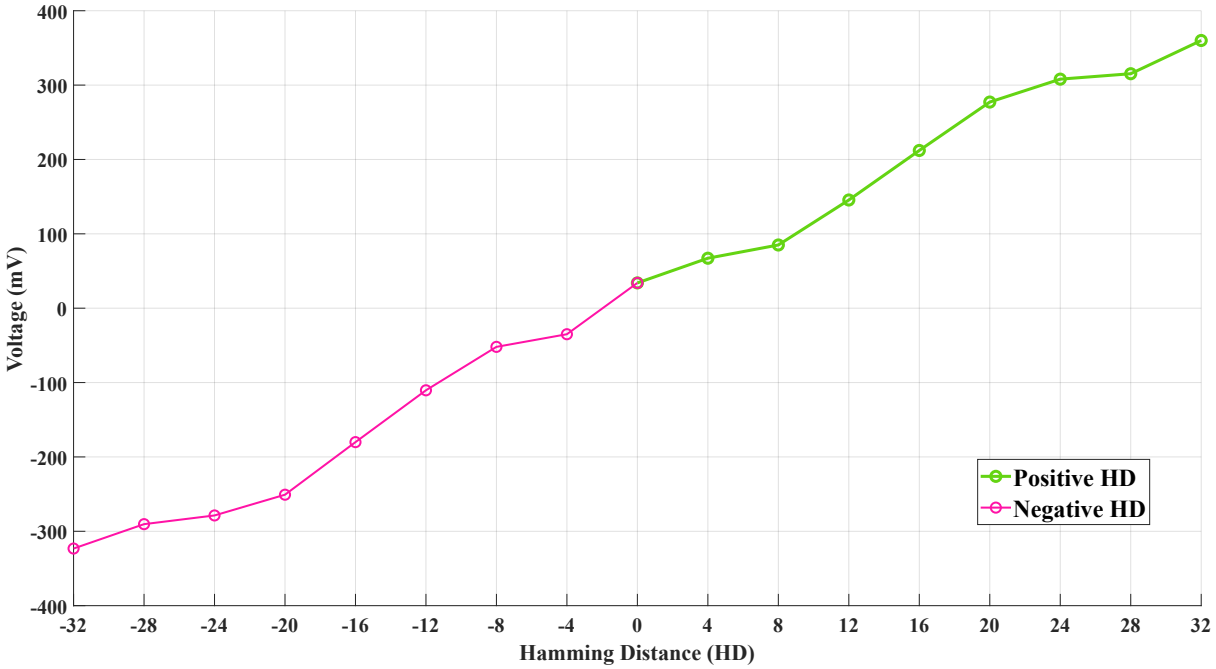


Figure 4.8 – Voltage vs. HD chart, showcasing the steady linear relation between the two elements.

will be formed in the trace which can be translated as a *descending* transition.

Note that a positive and negative bit transition on a bus corresponds to the charging and discharging of output gate capacitance or the parasitic capacitance of the bus line. For grayscale images used in this work, this ascending or descending nature of the bit transition determines whether the upcoming group of P pixels is likely to become lighter, or darker, than the current group. EM emanations induced by these transitions, $0 \rightarrow 1$ and $1 \rightarrow 0$, are additive and signed, leading to a form of *signed HD*. This signed HD is sensitive to variations between two successive pixel groups and not to actual pixel values directly, thus, information is lost in the interception process. This analysis does not segregate transitions on the least significant bits from those on the most significant bits. This method of analyzing the peak's nature of the EM footprint to infer the input pixels, which are sent through the internal bus for the inference process inside the FPGA is called Hysteresis Binary Image Reconstruction.

Mathematical Expression for HBIR Let P be the number of pixels transmitted simultaneously on the bus. Let G_t and G_{t-1} denote groups of P pixels transmitted at clock cycles t and $t-1$, respectively.

$$\mathrm{HD}(G_t, G_{t-1}) = \sum_{i=1}^{P} \delta(g_t^i, g_{t-1}^i)$$

where g_t^i and g_{t-1}^i represent the i-th pixel in the groups G_t and G_{t-1}, and

$$\delta(x, y) = \begin{cases} 1, & \text{if } x \neq y, \\ 0, & \text{if } x = y \end{cases}$$

indicates bit-wise difference between pixels.

The signed HD, $\mathrm{SHD}(G_t, G_{t-1})$, accounts for the direction of bit transitions (ascending or descending):

$$\mathrm{SHD}(G_t, G_{t-1}) = \mathrm{sign}(G_t - G_{t-1}) \cdot \mathrm{HD}(G_t, G_{t-1}),$$

$$\text{where } \mathrm{sign}(x) = \begin{cases} +1, & \text{if } x > 0 \quad \text{(ascending transition)} \\ -1, & \text{if } x < 0 \quad \text{(descending transition)} \\ 0, & \text{if } x = 0 \quad \text{(no transition)} \end{cases}$$

The EM trace value $T(t)$ at clock cycle t is proportional to the signed HD:

$$T(t) \propto \mathrm{sign}(G_t - G_{t-1}) \cdot \sum_{i=1}^{P} \delta(g_t^i, g_{t-1}^i)$$

Thus, the magnitude of $T(t)$ reflects the HD between the two pixel groups, and the sign of $T(t)$ represents whether the transition is ascending (positive) or descending (negative).

4.5.2 Image Recovery

Hamming Distance

The number of input pixels P, which can be sent concurrently at a single clock cycle, is determined by the data width of the internal bus. HD between two groups of P pixels, transferred between two consecutive clock cycles, corresponds to the number of bit positions at which the corresponding bit value has flipped (from $0 \rightarrow 1$, or $1 \rightarrow 0$). The HD provides crucial knowledge regarding the transmission of data, which can be exploited in this case. If the victim bus data width is 32 bits, a group of pixels comprises $P = 4$ pixels.

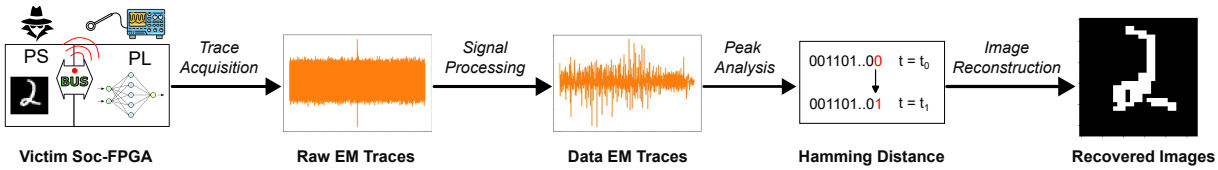


Figure 4.9 – Overview of real-time attack

Image Reconstruction Methodology

In this section, we propose a method to reconstruct a binarized version of the original image from the EM traces acquired. Since the transitions on the least significant bits are not distinguished from transitions on the most significant bits, binarization is a good solution to extract a decent amount of information with limited complexity. In our use case, this method allows us to recover the digit of the secret MNIST input image without needing to recover the exact values of the pixels of the image. To start reconstructing the binary version of the secret input image, an image matrix is set to the background color (*e.g.*, black in our experiments). The dimension of this matrix is deduced by the attacker during the pre-attack phase with the help of calibration images in Fig. 4.1. As the peaks are induced by bit transitions, in the case of black & white row, there is always a significant peak at the end of each line (at every 13.65 *ns*). By using this information, the placement of pixels in each row and the whole dimension of the image (approximately 374 *ns*) is recovered. The regeneration of the image is sequentially done until the whole image matrix is obtained. The general principle of the attack is demonstrated in Fig. 4.9.

Let P be the number of data pixels that can be transmitted simultaneously on the internal bus. For *e.g.*, the value of P is 4 for 32-bit bus and 8 for 64-bit bus. Starting off, the voltage values of the trace at each individual time instance (= 1.95 *ns*) are steadily scanned. When the scanned voltage value reaches a positive threshold, the pixel at that certain time instance, as well as the upcoming *P-1* pixels in the scanning order, will be assigned to a different color (*e.g.*, white in our case as MNIST is grayscale). Now the current white pixel value is maintained for the rest of the following pixels until a negative threshold is triggered. Similarly, once the negative threshold is reached during the scan, the corresponding pixel at the detected time instance and its following *P-1* pixels are assigned to black until the detection of another positive peak. This process is repeated continuously for the entire length of EM trace signal while every pixel value inside the image matrix is assigned to either black or white.

Image Reconstruction Algorithm Description

Define the following parameters:

- $T(t)$: EM trace signal as a function of time t.
- P: Number of pixels transmitted simultaneously on the bus (e.g., $P = 4$ for 32-bit, $P = 8$ for 64-bit).
- Threshold_+ and Threshold_-: Positive and negative voltage thresholds, respectively.
- $I(i, j)$: Binary image matrix initialized to black ($I(i, j) = 0$), where i and j are the row and column indices.

For each time instance t_k, compare $T(t_k)$ to the thresholds and update pixel values in the image matrix:

$$\begin{aligned} &\text{If } T(t_k) \geq \text{Threshold}_+, \qquad I(i, j) = 1 \quad \text{for } j = k, k+1, \ldots, k+(P-1) \\ &\text{If } T(t_k) \leq \text{Threshold}_-, \qquad I(i, j) = 0 \quad \text{for } j = k, k+1, \ldots, k+(P-1) \end{aligned}$$

Initialize the **Image** matrix:

$I(i, j) = 0$ (black for all pixels initially)

For each time instance t_k :

If $T(t_k) \geq \text{Threshold}_+$:

Set the next P pixels to white:

$I(i, j) = 1 \quad \text{for} \quad j = k, k+1, \ldots, k+(P-1)$

If $T(t_k) \leq \text{Threshold}_-$:

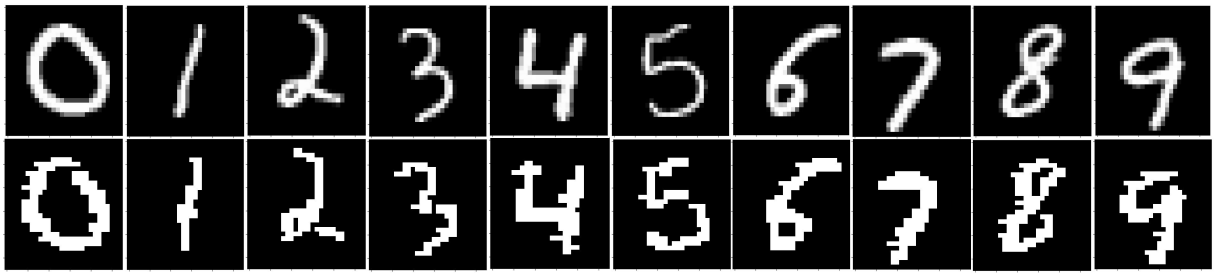
Set the next P pixels to black:

$I(i, j) = 0 \quad \text{for} \quad j = k, k+1, \ldots, k+(P-1)$

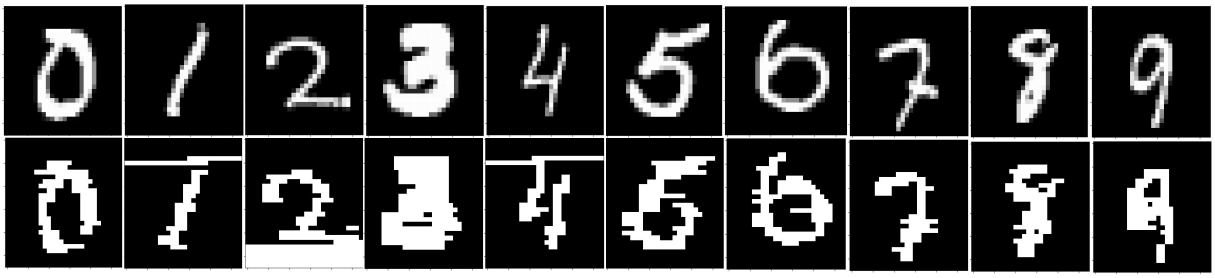
4.6 Results and Evaluation of HBIR Methodology

The acquisition of EM traces is performed as described in section 4.5, followed by the step-by-step application of the proposed methodology (noise characterization, HBIR, etc.) onto 10,000 MNIST test images. A few randomly picked images and their respective results are shown in Fig. 4.10. Even with visual inspection, the excellent condition of the recovered images makes them easy to identify the shape of the original input image to the accelerator.

Despite the algorithm being able to recover distinguishable images from the EM traces (for instance in Fig. 4.10a), a third-party Neural Network classifier (trained on MNIST) allows to categorize them systematically. The classifier used is a three-layer MLP, structurally similar to the one implemented on the Pynq Z-1 platform, though not identical. Additionally, this verification with third-party NN classifier gives a fair evaluation of the precision of this attack and computes the recognition accuracy. The overall process of validation of the resulting images to calculate the accuracy of the attack is described in Fig. 4.12. To perform this, the proposed methodology is applied to 10,000 MNIST images from test dataset [125] and attain the average recognition accuracy of 83.69% from the recovered images by HBIR as evidenced by Fig. 4.11. These results highlight the performance of thHBIR, demonstrating its potential for accurately reconstructing input pixel values from EM side-channel information. Furthermore, a comparative analysis of this method with the second methodology will be provided in detail in Chapter 5 to evaluate and contrast their respective performances.



(a) Example of original MNIST images (first line) with highly accurate recovered images through HBIR (second line)



(b) Example of original MNIST images (first line) with less accurate recovered images through HBIR (second line)

Figure 4.10 – Randomly chosen examples of retrieved MNIST images with HBIR

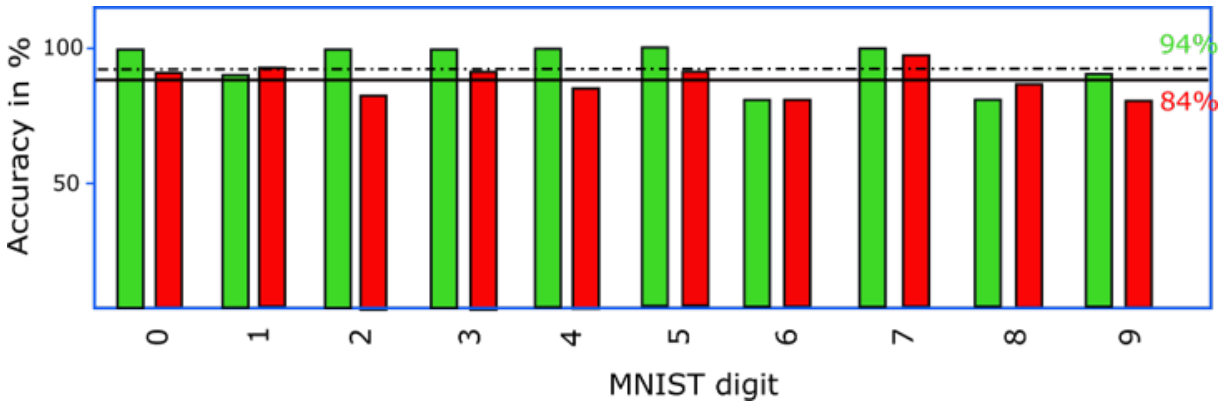


Figure 4.11 – Recognition accuracy of HBIR method (in red) with respect to the accuracy of original MNIST dataset (in green).

4.6.1 Limitations of HBIR

The presented study is investigated on signed HD values between two consecutive 32-bit data transitions. HD in this case, gives information about the number of bit changes between two groups (each containing four pixels), instead of the exact value of a pixel. Despite the consequent loss of information, HBIR can reconstruct images close enough to the secret original images, from the collected EM emanations (seen in Fig. 4.10). On the other hand, increasing the data bus width (*e.g.*, from 32 to 64 bits) is expected to result in greater information loss. This is because HBIR primarily relies on HD information between consecutive pixel groups to reconstruct images, meaning larger pixel groups (per time instance) in larger buses may lead to higher loss.

In addition, certain handwritten digit can result in lower recognition ability compared to the others. For instance, in the case of *9*, the classifier may misclassify it as *4* depending on the shape it is written (in Fig. 4.10b). Another cause would be the false detection of peaks in the EM trace. Since our experiments are done in an open space and not within a noise-free environment, such as a Faraday's cage, unwanted noises may interfere during the acquisition. Despite the noise, our HBIR method successfully retrieves the input MNIST digits with high accuracy.

4.7 Chapter Conclusion

This chapter highlights the HBIR methodology, a novel and efficient approach for conducting EM eavesdropping on SoC-FPGA platforms. The methodology successfully

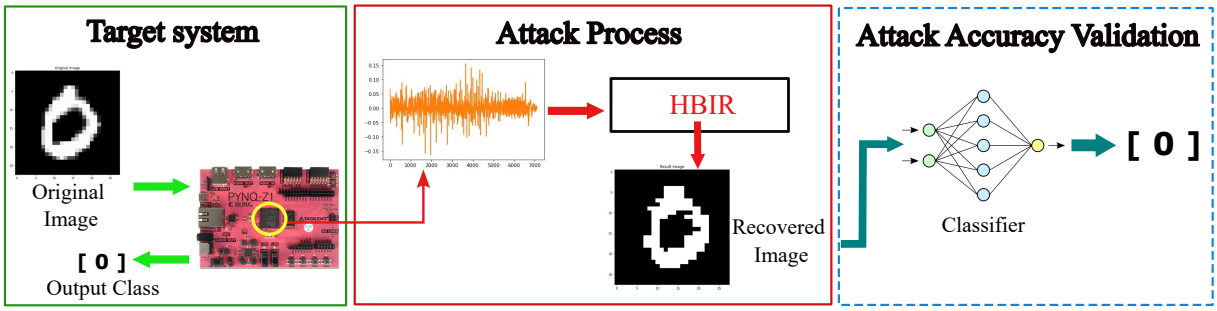


Figure 4.12 – Targeted victim system, HBIR attack, and validation process of retrieved images with a NN classifier.

demonstrated how EM emanations from internal data buses can be exploited to reconstruct input image data with minimal system interaction. HBIR mainly consists of two parts to execute; pre-attack and real-time attack phases. Pre-attack phase not only allows the attacker to understand the EM signature concerning pixel changes but also helps with the detection and synchronization of data transfer on the bus— an essential step given the one-shot acquisition process, where signals are captured without averaging. Real-time attack is composed of noise characterization and preparation of HBIR, the image reconstruction framework for the first methodology (main content of this Chapter). By applying this framework, we achieved an average recognition accuracy of 83.69% for recovered MNIST images from the training dataset, demonstrating the feasibility of the attack. Regardless of the inherent information loss due to reliance on HD between pixel groups, the approach proved effective in extracting recognizable patterns.

HBIR, however, suffers from information loss due to the aggregated analysis at the 16-bit level. This limitation constrains the retrieval of more complex images or the analysis of finer-grained data types. To address this limitations, the next chapter will introduce the second attack methodology, which offers finer granularity and improved precision in reconstructing input data. Unlike the HBIR, this advanced technique is equipped to handle not only MNIST images but also more complex grayscale images, showcasing its broader applicability and enhanced capabilities.

Chapter 5

Super-Bits Image Reconstruction

Contents

5.1 Super-Bits . . . 114
5.1.1 Super-Bits of 8 . . . 115
5.1.2 Super-Bits of 4 . . . 116
5.2 Preliminary Phase . . . 117
5.2.1 Database Template . . . 118
5.3 Exploitation Phase . . . 121
5.4 Preparation Stage . . . 123
5.4.1 Raw EM Trace . . . 123
5.5 Correlation Stage . . . 129
5.6 Reconstruction Stage . . . 137
5.7 Evaluation of Results from Methodology 2 . . . 137
5.7.1 Transition Recovery . . . 137
5.7.2 Limitations of SIR . . . 147
5.8 Comparison Between Methods: HBIR and SIR . . . 147
5.8.1 Limitations . . . 148
5.9 Chapter Conclusion . . . 152

While HBIR demonstrates a reasonably high attack accuracy for image recovery on the standard MNIST test dataset, which consists of 10,000 images, its limitations becomes evident when examining the retrieved HD. Specifically, these HD values obtained through HBIR are restricted to -16, 0, and 16, representing descending transitions, no transitions, and ascending transitions, respectively. This limited range fails to capture intermediate HD such as 4, 8, and 12, which could provide a more detailed and granular representation of bit transitions. Therefore, the focus of this chapter is to explore and develop an improved methodology, called SIR, that enables the retrieval of a wider range of HD values (0, 4, 8, 12, and 16), offering finer insights into the transition behavior and enhancing the accuracy of image reconstruction. Although the total range of HD values spans from 0 to 32, certain limitations make it impractical to consider every single HD. Firstly, the differences in voltage between consecutive HD are minimal. This lack of distinction makes the recovery of precise transitions for every HD value challenging. Hence, restricting the focus to small sets of HD values allows better differentiation.

Besides, while HBIR is effective for synthetic datasets such as MNIST, for which white pixels appear on a perfectly black background, it exhibit limitations when applied to more complex images with more diverse range of grayscale pixel values (or even colorful images). HBIR is primarily based on the integration of peaks across the EM trace to differentiate between lighter and darker pixel changes. SIR helps identify the edges of the shapes in the original image, as most pixel transitions occur along the shape's boundaries. In the case of MNIST, which is a type of synthetic grayscale image, HBIR is sufficient to identify the edges due to the high contrast between the foreground shape and the background. However, for more naturalistic grayscale images, where pixel intensity variations are more significant across the entire image, additional analysis is required to retrieve and reconstruct images. The main comparison between the two methodologies is that, the first one integrates the positive and negative peaks along the length of the time-domain trace to distinguish (lighter or darker) pixel changes, while this methodology directly correlates the HD on a similar trace signal using a template.

5.1 Super-Bits

In section 3.5.1, it is found that there are a total of 16 physical bus wires internal to SoC-FPGA, responsible for transferring 32-bit input data on both rising and falling edge of the clock for rapid data rate. It is also found that the emanation strength varies depending

on where the HD is located along the 16-bit sequence from that analysis (2-bytes when the input is in image format). The proposed layout of bus wires is deduced as illustrated in Fig. 3.27. The induced voltage on the EM signature is directly influenced by the distance between the physical position of the wires and the probe loop. This observation from the bit arrangement experiments (in section 3.5.2) suggests that grouping the bits sequentially, as traditionally done in Fig. 5.1, may not be the most effective approach. Instead, we propose to group the bits based on their relative positions to the probe —either at the extremities or centralized— allowing for a clearer and more consistent characterization of data from its EM emanations. We call this particular way of grouping the bits as **Super-Bits** in the remainder of this manuscript. The expression for the super-bits grouping is as follows:

$$\mathbf{X} = (X_1, X_2, X_3, \ldots, X_{31}, X_{32})$$

Here, $\mathbf{X}$ represents the sequence of 32 bits transmitted in a single clock cycle. However, it is already known that the targeted buses run on DDR, therefore, the 32-bit data is split into two halves of 16 bits each, with the first half transmitted on the rising edge of the clock and the second half on the falling edge. Considering that the transitions occur at every clock edge (rise and fall), this method aims at analyzing the bit transitions within each 16-bit segment, rather than treating the entire 32-bit data width as a single unit. Therefore by focusing on the changes between bits in every half-clock cycle, this methodology naturally emphasizes the super-bits grouping within every 16-bit sequence. Two types of Super-Bits groupings, super-bits of 8 and 4 are implemented in this methodology.

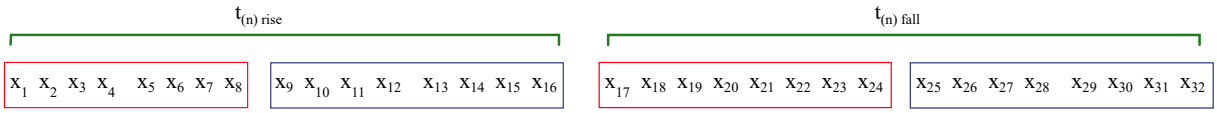


Figure 5.1 – Traditional (sequential) grouping.

5.1.1 Super-Bits of 8

Super-Bits of 8 consist of building groups of 8 bits out of 16-bit sequences, thus, resulting in two special groups per 16-bit sequence. For a full 32-bit sequence, this produces four groups in total—two groups corresponding to the rising edge and two groups

corresponding to the falling edge of the clock cycle as illustrated in Fig. 5.2.

$$X_a = (X_1, X_2, X_3, X_4, X_{13}, X_{14}, X_{15}, X_{16}) \quad \in \{0,1\} \tag{5.1}$$
$$X_b = (X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}) \quad \in \{0,1\} \tag{5.2}$$
$$X'_a = (X_{17}, X_{18}, X_{19}, X_{20}, X_{29}, X_{30}, X_{31}, X_{32}) \quad \in \{0,1\} \tag{5.3}$$
$$X'_b = (X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}) \quad \in \{0,1\} \tag{5.4}$$

Group X'_a, seen in equation (5.1), contains 8 bits selected from the first 16 bits of the 32-bit sequence. Specifically, it comprises bits $X_1, X_2, X_3, X_4, X_{13}, X_{14}, X_{15}, X_{16}$, presenting the first 4 bits and the last 4 bits of the that sequence. As for the group X_b in equation (5.2), it consists of 8 bits, which are taken from the middle section of the same 16-bit sequence. It covers from the 5th bit to 12th bit in order, which represent $X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}$. Group X'_a is similar to X_a but is formed from the second 16-bit segment (the remaining 16 bits out of the 32-bit sequence which occurs on the falling edge of clock) according to equation (5.3). It includes bits X_{17}, X_{18}, X_{19}, X_{20}, X_{29}, X_{30}, X_{31}, X_{32} which are the first 4 and last 4 bits of the second 16-bit sequence. Likewise, group X'_b focuses on the second 16-bit segment as shown in equation (5.4). It gathers bits X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}, X_{27} and X_{28} corresponding to the middle 8 bits of the second 16-bit sequence. This grouping is motivated by the results of section 3.5.2.

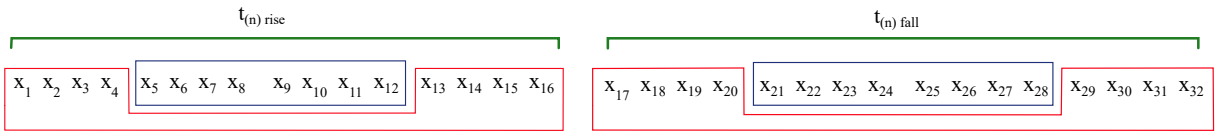


Figure 5.2 – Super-Bits of 8 grouping

5.1.2 Super-Bits of 4

Similar to the Super-Bit of 8, the Super-Bit of 4 also organizes bits by merging those from the extremities, progressively shifting toward the center. However, the key difference lies in the size per group, which consists of 4 bits instead of 8, allowing for a finer-grained

analysis of the data sequence. The expression of Super-Bits of 4 is shown as follows:

$$X_a = (X_1, X_2, X_{15}, X_{16}) \quad \in \{0, 1\} \tag{5.5}$$
$$X_b = (X_3, X_4, X_{13}, X_{14}) \quad \in \{0, 1\} \tag{5.6}$$
$$X_c = (X_5, X_6, X_{11}, X_{12}) \quad \in \{0, 1\} \tag{5.7}$$
$$X_d = (X_7, X_8, X_9, X_{10}) \quad \in \{0, 1\} \tag{5.8}$$
$$X'_a = (X_{17}, X_{18}, X_{31}, X_{32}) \quad \in \{0, 1\} \tag{5.9}$$
$$X'_b = (X_{19}, X_{20}, X_{29}, X_{30}) \quad \in \{0, 1\} \tag{5.10}$$
$$X'_c = (X_{21}, X_{22}, X_{27}, X_{28}) \quad \in \{0, 1\} \tag{5.11}$$
$$X'_d = (X_{23}, X_{24}, X_{25}, X_{26}) \quad \in \{0, 1\} \tag{5.12}$$

The 32-bit data sequence is divided into two 16-bit segments, with each segment further grouped into four Super-Bit groups of 4 bits each. Explicitly, each group combines bits from opposite ends of the sequence and works inward as depicted in Fig. 5.3.

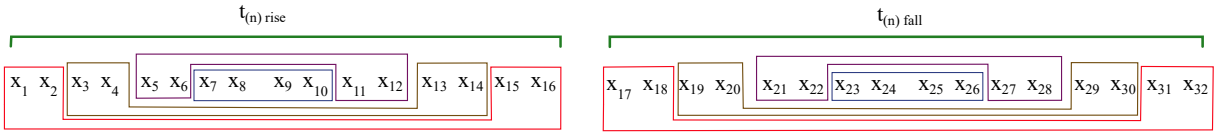


Figure 5.3 – Super-Bits of 4 grouping

The use of two categories of Super-Bits, 4 and 8, in this methodology is to evaluate the performance of the attack under different levels of granularity. Super-Bit 4 is expected to provide a finer granularity, allowing for more detailed analysis of the transitions, while Super-Bits of 8 offers a coarser granularity.

5.2 Preliminary Phase

This phase primarily focuses on making a database template for the reference to reverse engineer data from the collected EM traces. HD voltage graph in Fig. 4.8 suggests that there is a linear relationship between signed HD and its induced voltage captured at the probe terminal. Based on this evidence, a HD template (referred to as "database template" in this thesis) is created to apply in the correlation stage. This template serves as a reference for detecting the most correlated signed HD pattern throughout the captured data trace.

5.2.1 Database Template

It is well-established that the transition of data bits on bus lines causes current switching at the HW level, which in turn induces a voltage (generated by the H-field according to Faraday's Law, as explained in 3.2). The EM traces obtained in this work are composed of these induced voltages, which are a result of bit flipping, or in other terms, the HD. In this methodology, we aim to reverse engineer the HD from the targeted bus with higher precision by observing the leakage trace. It is already known to us that different HD values produce distinct shapes of peaks in the trace according to Fig. 4.8. By creating a database template that maps HD values to their corresponding EM peak patterns, a strong base is established for the identification of HD values throughout the acquired trace.

It is also important to consider that the distance from the probe to the actual bus wires inside the SoC-FPGA impacts the strength of the emanation. Using the probe placement described in Fig. 3.27, the concept of Super Bits is taken into account when making database template. With this approach, the bits in the database are not grouped in a traditional sequential order but are instead organized based on the Super Bits grouping schemes.

Finally, it is crucial to recognize that the attacker is assumed to possess the HW replica identical to the victim system, as explained in section 4.2. This allows the attacker to manipulate the targeted bus system by sending bits in specific arrangements to generate the desired HD values and their corresponding induced voltages as reported in Fig. 5.4. Additionally, since the attacker knows the preamble and postamble patterns from the calibration experiment described in section 4.4.2, it becomes straightforward for them to determine the start of the data transfer. This knowledge enables the attacker to construct the database in advance, prior to launching the attack.

To conclude, database template in this context, is a collection of different HD values with their reciprocal EM signature. It serves as a reference value for the correlation which is required in the next stage. To compare and observe different levels of granularity, two database templates are generated, one with a Super-Bits size of 8 and the other with a Super-Bits size of 4.

Database with Super-Bits of 8

It is important to emphasize that, although the total data width is 32 bits, the analysis focuses solely on super-bit transitions, which are evaluated within 16-bit sequences. For

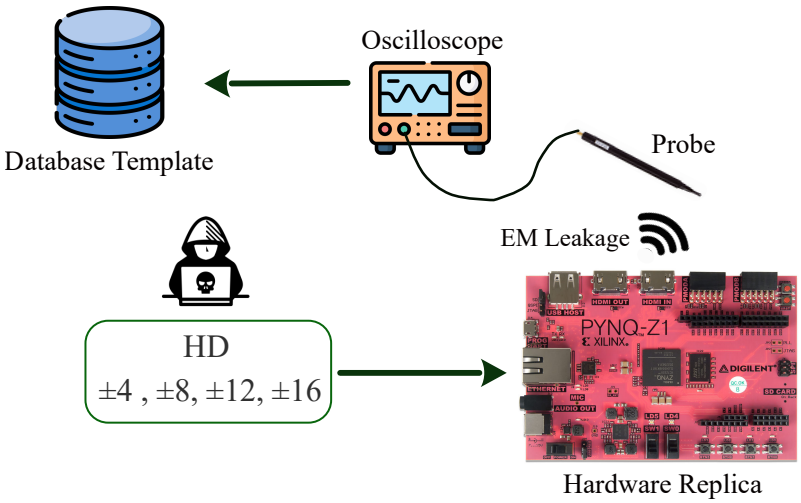


Figure 5.4 – Creation of a database template by the adversary involves sending special groups of bits on the SoC bus (of the HW replica) to induce specific HD values. The corresponding EM traces are collected to construct the database template.

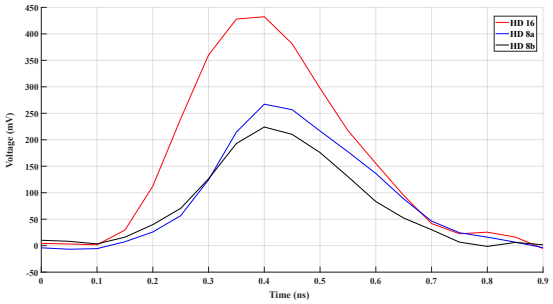
Super-Bits of 8, each group consists of 8 bits, and the total 16-bit sequence is divided into two non-overlapping groups. This results in only 2 groups of 8 bits as seen in Table. 5.1, and the number of possible Super-Bits of 8 combinations is $2^2 = 4$, as each bit can independently take the value 0 or 1.

HD	Binary representation	Possible Combinations
0	0000 0000 0000 0000	(0, 0)
8a	1111 0000 0000 1111	(1, 0)
8b	0000 1111 1111 0000	(0, 1)
16	1111 1111 1111 1111	(1, 1)

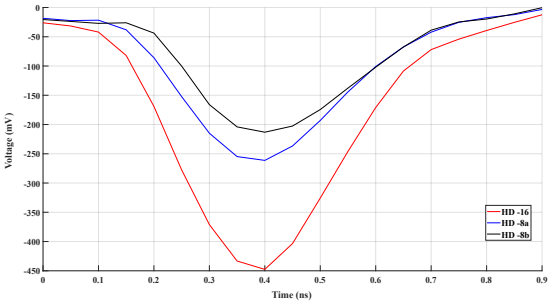
Table 5.1 – Possible HD values and combinations for Super-Bits of 8 grouping. The numeric values in the first column represent the HD values. If a number is accompanied by a letter (e.g., 8a, 8b), it indicates possible alternatives to that specific HD value, reflecting different bit transition patterns

Prior to transmitting the special bit groups, the bus carries only zero bits. For instance, if the special group of bits is sent at time t, the bus state at $t-1$ is 0000 0000 0000 0000, representing the inactive state. Once the special group is sent at t, the state of the bus changes to reflect the transmitted data. Afterward, at $t+1$, this state reverts back to its inactive state (0000 0000 0000 0000). This EM peak pattern induced by change of bus state, allows for both ascending and descending transitions of that HD to be ob-

served. The ascending transition occurs as the bus transitions from the inactive state (0000 0000 0000 0000) at $t-1$ to the active state (the special bit group) at t. The descending transition is obtained when the bus returns from the active state at t back to the inactive state (0000 0000 0000 0000) at $t+1$. Through this process, the EM signature for both + and - of that same HD values are captured, taking a step forward to the correlation stage. Based on Super-bits of 8 grouping, the database template illustrated in Fig. 5.5 is built. Fig. 5.5a illustrates the distinctive EM signatures induced by each positive HD, while Fig. 5.5b captures the EM emanations generated by their negative counterparts, offering the reference templates for the subsequent correlation phase.



(a) Each curve corresponds to a positive HD value, showcasing the distinct signal shapes associated with different transitions on the bit sequence.



(b) Each curve corresponds to a negative HD value, showcasing the distinct signal shapes associated with different transitions on the bit sequence.

Figure 5.5 – EM signals corresponding to HD values in the database template for Super-Bits of 8

Database with Super-Bits of 4

For Super-Bits of 4, each special group consists of 4 bits, and the 16-bit sequence is divided into four non-overlapping groups. This results in 4 groups of 4 bits each, with 2^4=16 possible combinations per group as shown in Table 5.2. To recall, although the data width is 32 bits, the focus is solely on bit transitions, leading the analysis to be based on 16-bit sequence, instead of full 32-bit. Although HD of 0 is part of the possible bit combinations, the corresponding EM waveform is not included in the database. This is because the waveform for HD of 0 is very close to the noise floor, thus, by applying the same threshold introduced in section 4.5.1, correlation with EM waveform by HD of 0 is automatically excluded from the set of possible candidates.

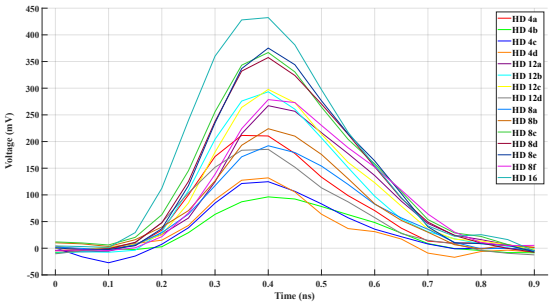
HD	Binary representation	Possible Combinations
0	0000 0000 0000 0000	(0, 0, 0, 0)
4a	1100 0000 0000 0011	(1, 0, 0, 0)
4b	0011 0000 0000 1100	(0, 1, 0, 0)
4c	0000 1100 0011 0000	(0, 0, 1, 0)
4d	0000 0011 1100 0000	(0, 0, 0, 1)
8a	1111 0000 0000 1111	(1, 1, 0, 0)
8b	1100 1100 0011 0011	(1, 0, 1, 0)
8c	1100 0011 1100 0011	(1, 0, 0, 1)
8d	0011 1100 0011 1100	(0, 1, 1, 0)
8e	0011 0011 1100 1100	(0, 1, 0, 1)
8f	0000 1111 1111 0000	(0, 0, 1, 1)
12a	1111 1100 0011 1111	(1, 1, 1, 0)
12b	1111 0011 1100 1111	(1, 1, 0, 1)
12c	1100 1111 1111 0011	(1, 0, 1, 1)
12d	0011 1111 1111 1100	(0, 1, 1, 1)
16	1111 1111 1111 1111	(1, 1, 1, 1)

Table 5.2 – Possible HD values and combinations for Super-Bits of 4 grouping. Similar to Table 5.1 the numeric values in the first column represent the HD values, meanwhile, its accompanied letter tells possible alternatives to that specific HD value

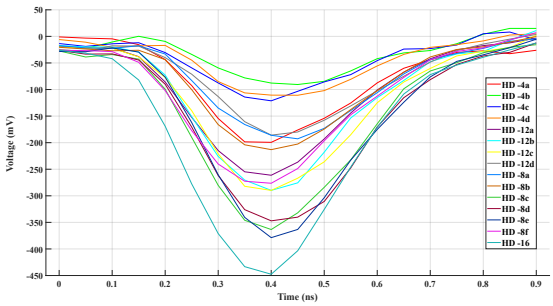
Just as in Super-Bits of 8, in this case, the special bit group is transferred at a certain time instance. Before this instance, the bus remains in an inactive state and it returns to this same inactive state afterwards, ensuring no other transitions occur outside this instance. This process produces corresponding peak patterns on the EM trace, capturing both ascending (+HD) and descending (-HD) transitions with one pulse. Database template constructed using Super-Bits of 4 is shown in Fig. 5.6.

5.3 Exploitation Phase

There are three obligatory stages to perform this attack, called **PCR**; **P**reparation, **C**orrelation and **R**econstruction. Preparation stage involves arranging and processing the collected raw EM traces so that it can be put into the correlation operation. It is important



(a) EM peak pattern induced by positive HD values, highlighting the unique signal patterns generated by different transitions within the bit sequence.



(b) EM peak pattern induced by negative HD values, highlighting the unique signal patterns generated by different transitions within the bit sequence.

Figure 5.6 – EM signals corresponding to HD values in the database template for Super-Bits of 4.

to notice that synchronization of raw trace to detect the true start and the end of the actual data transmission is identical to the one mentioned in Chapter 4 (section 4.4.3). Once the precisely aligned data trace is ready, interpolation, directly followed by decimation [1], are applied onto the data trace to make sure that peaks in the signal stays in the center of each clock cycle window.

After the preparation is complete, the next stage is to use the correlation operator to identify the signed HD among the elements of template that matches at each half clock cycle of across the whole time-domain data trace. The goal of using correlation is to extract the HD value from the EM trace by recognizing the similarity between the template and each window of the trace. A few extra conditions (such as, putting a voltage threshold) are added in the correlation stage for better selection of correct HD. At the end of this stage, an HD image that tells the individual signed HD at each half cycle is achieved. This HD image is considered as an intermediate result and plays an important role in reconstructing the final image.

In the reconstruction stage, the retrieved HD image from correlation stage acts as the reference for recovering the original image. It is important to note that the information obtained from the attack represents the HD (*i.e.*, the differences in bit transitions throughout the 16-bit sequence), rather than the original pixel values. As a result, a large amount of information loss is inherent in the process. The original pixel values can only be approximated based on the numbers of bits that have flipped, with the reconstruction

1. The process of reducing the sampling rate of a signal.

heavily relying on the prior state for accurate estimation. The overall procedure of **PCR** is illustrated in Fig. 5.7.

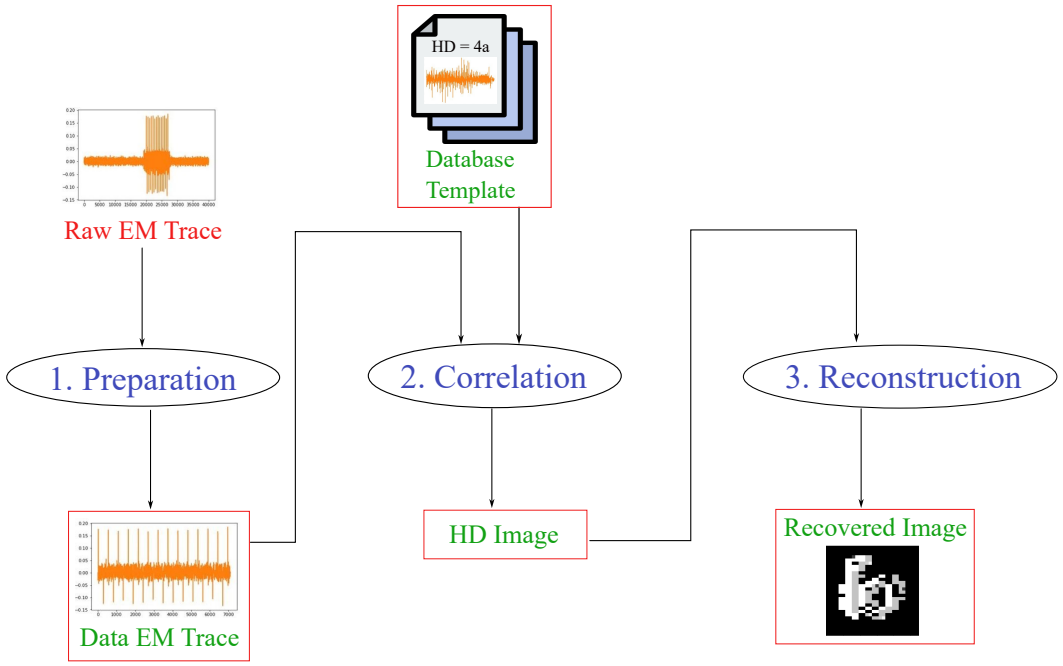


Figure 5.7 – An overview of the exploitation phase, including its mandatory stages: preparation, correlation, and reconstruction.

5.4 Preparation Stage

According to **PCR**, the first stage to do in this methodology is the preparation. It involves preparing and processing the raw EM trace captured at the output of the probe to be suitable for further stages. As a reminder, the experimental setup for the acquisition of raw EM trace is the same as in Fig. 4.6 Here, a thorough, step-by-step explanation of how to carry out preparations is provided in the following sections.

5.4.1 Raw EM Trace

Since the traces will be used in the correlation process and the template consists of peak patterns, it is essential that the peaks must be aligned at the center of each scanning window (corresponding to half a clock cycle). This requirement makes the preparation of the raw trace a critical requirement. Prior to this, it is mandatory to first convert the

raw traces into data traces like in Chapter 4. This step is done by the precise detection of preamble and postamble which marks the duration of bus data transfer as described in section 4.4.3. Once true start and end time is identified, the raw traces are segmented to actual data traces. After obtaining the perfectly synchronized data trace, the next step is to ensure that the peaks are properly aligned within each scanning window. This alignment is achieved through interpolation, which increases the resolution of the signal, and followed by decimation to reduce the sampling rate back to the original rate.

Interpolation and Decimation of Data Traces

Interpolation is an estimation method in signal processing, used to construct new data points (samples) within the range of existing ones in a signal. This technique allows us to increase sampling rate of the signal, align signals with different sampling rates, or estimate missing data points in the original signal under some hypothesis. The main objective of interpolation is to create a smoother representation of the signal by filling in the gaps between known samples. This technique is used for the enhancement of digital signal resolution, synchronize signals sampled at different rates, and recover signals when data points are sparse or missing. A few types of commonly used interpolation in signal processing are linear, polynomial, and spline interpolation. In this method, linear interpolation is employed due to its simplicity for aligning the peaks in the data trace with minimal computational complexity. The linear interpolation is done by:

$$y = y_1 + \frac{(x - x_1)(y_2 - y_1)}{x_2 - x_1}$$

Where: x_1 and x_2 are the known data points on the x-axis, y_1 and y_2 are the corresponding values on the y-axis. For a given point x within ($x_1 \leq x \leq x_2$), the value y is interpolated.

Interpolation is essential in our methodology to achieve better correlation results. It is applied to align the peak of the signal within the center of each scanning window. To demonstrate the necessity of interpolation, the time shift versus transition plot is discussed. In Fig. 5.8, x-axis represents the transitions, where a total of 392 transitions are observed in a 784-pixel (28 x 28) image. This is because, in DDR, two pixels are transmitted per half clock cycle. The y-axis indicates the time shift between the theoretical peak (expected position of the peak at each window, which is the center) and the actual peak (observed position on the trace). Moreover, the red bars denote positive peaks, while the blue bars represent negative peaks. Given that the input image is a checker-

board calibration image (as shown in Fig. 4.1d), the positive and negative peaks alternate consistently along the signal. Additionally, a positive time shift indicates that the actual peak precedes the theoretical peak, whereas a negative time shift signifies that the actual peak lags behind the theoretical peak. According to Fig. 5.8, it is clear that the time shifts are minimal during the initial transitions, indicating close alignment between the actual and theoretical peaks. However, as the transitions progress, the shifts gradually accumulate, resulting in increasingly larger deviations. In the few final transitions, these shifts become significantly pronounced, reaching up to 0.9 ns. Notably, the shifts predominantly show a delay, with the actual peaks occurring after the theoretical peaks. This is because, the oscilloscope in this experiment setup runs at a sampling rate of 20 GS/s and each sample represents a time interval of 50 ps. However, the actual window size is 0.95238 ns, calculated as $\frac{1}{525\,\text{MHz}} = \frac{1.9047\,\text{ns}}{525} \div 2$. Because the sampling interval is fixed at 50 ps, a misalignment of 0.00238 ns *i.e.*, the difference between the theoretical window and the sampling period, occurs at every transition. Over 392 transitions, this small shift aggregates, resulting in a total misalignment of up to 0.9 ns as seen in Fig. 5.8. In other words, the misalignment arises because the sampling resolution of $50ps$ is not fine enough to precisely capture the exact position of the peaks.

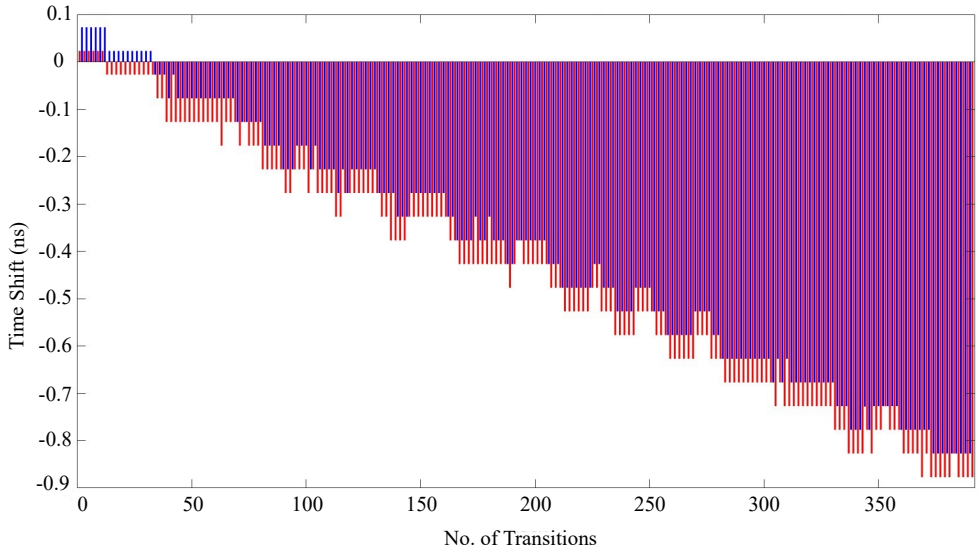
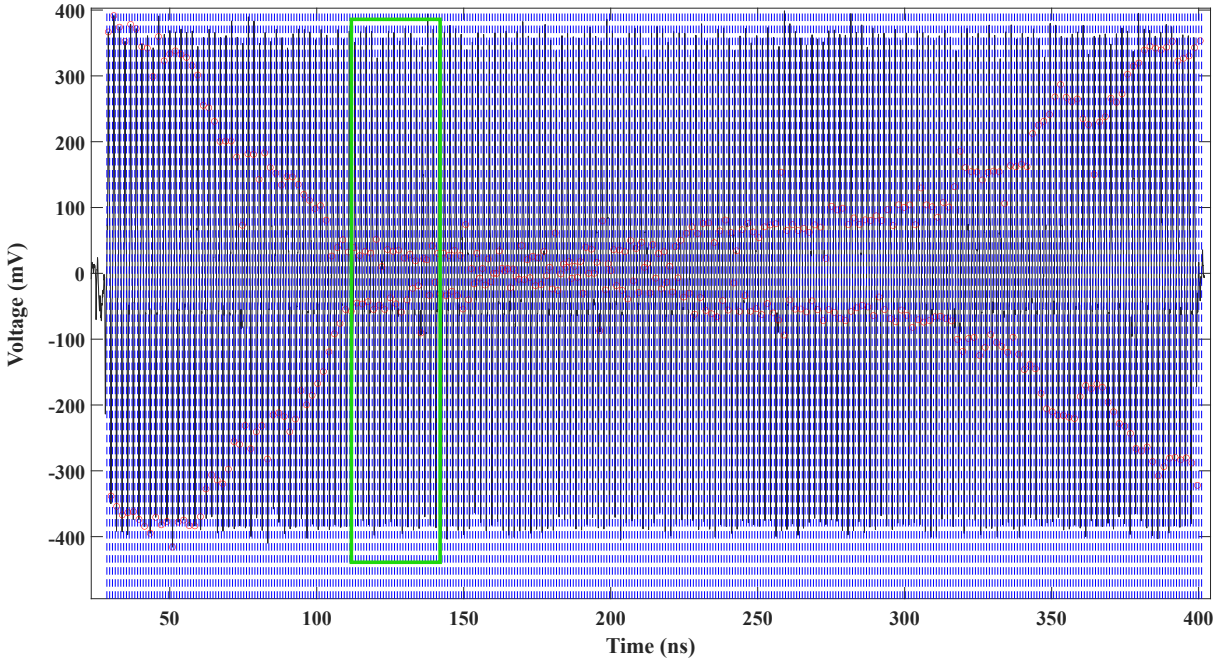
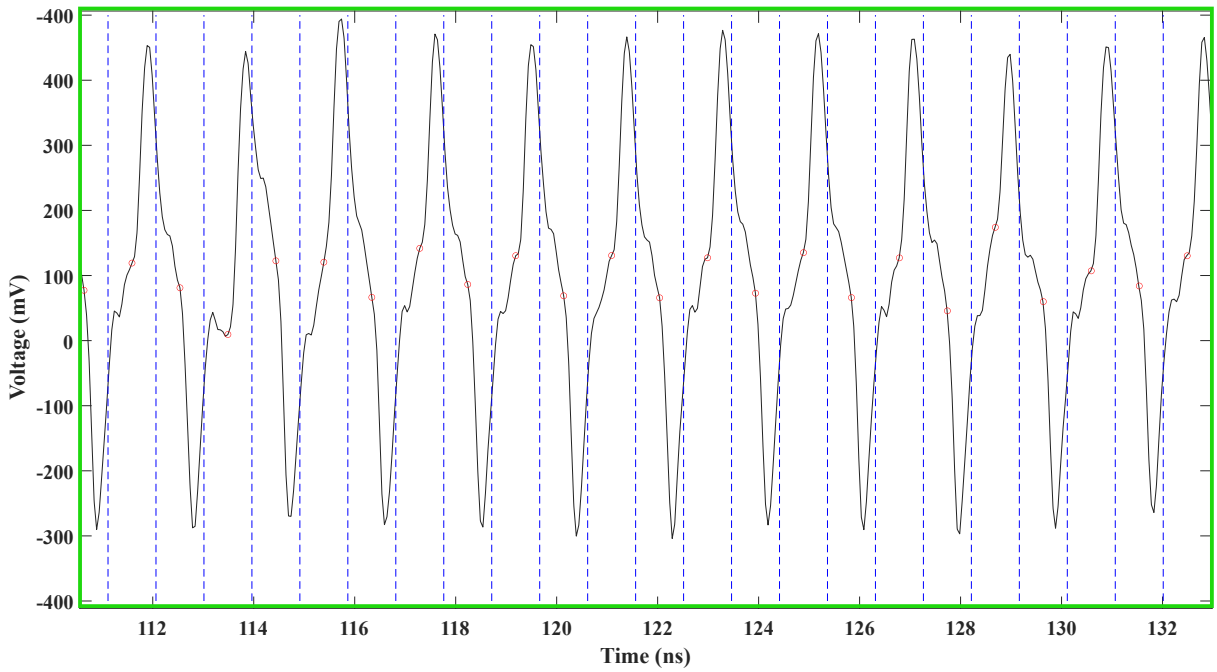


Figure 5.8 – Time shift vs transition plot to analyze the delay in peaks before interpolation.

Another evidence to prove the shift in the signal with original sampling rate is seen in Fig. 5.9. Fig. 5.9a reveals that the EM trace collected at the probe terminal when a checkerboard input image is sent (in black). The vertical dotted blue lines indicate the



(a) EM trace showing the captured signal with vertical dotted blue lines representing the scanning windows and red circles indicating the theoretical peak positions.



(b) Zoomed-in view of (a) for visualization of shift in theoretical peak position and real peak.

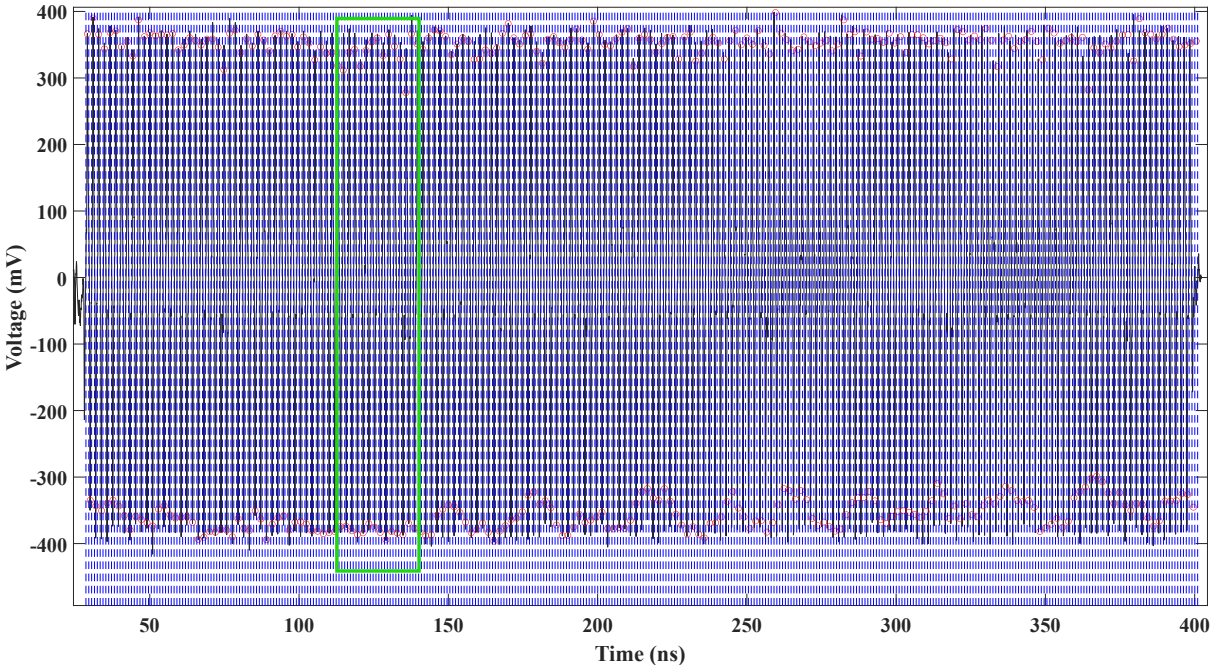
Figure 5.9 – Alignment of real peaks to the theoretical window **before** interpolation.

scanning window intervals at 0.95 ns while the red circles mark the theoretical positions where the peaks should occur. However, as displayed in Fig. 5.9b, the actual peaks in reality do not perfectly align with the center of the intervals. Therefore, it clearly highlights the peak misalignment due to signal imperfections, necessitating interpolation to correct these shifts and adjust the peaks accurately for subsequent correlation analysis. Without interpolation, the correlation process would suffer from inaccuracies caused by this misalignment.

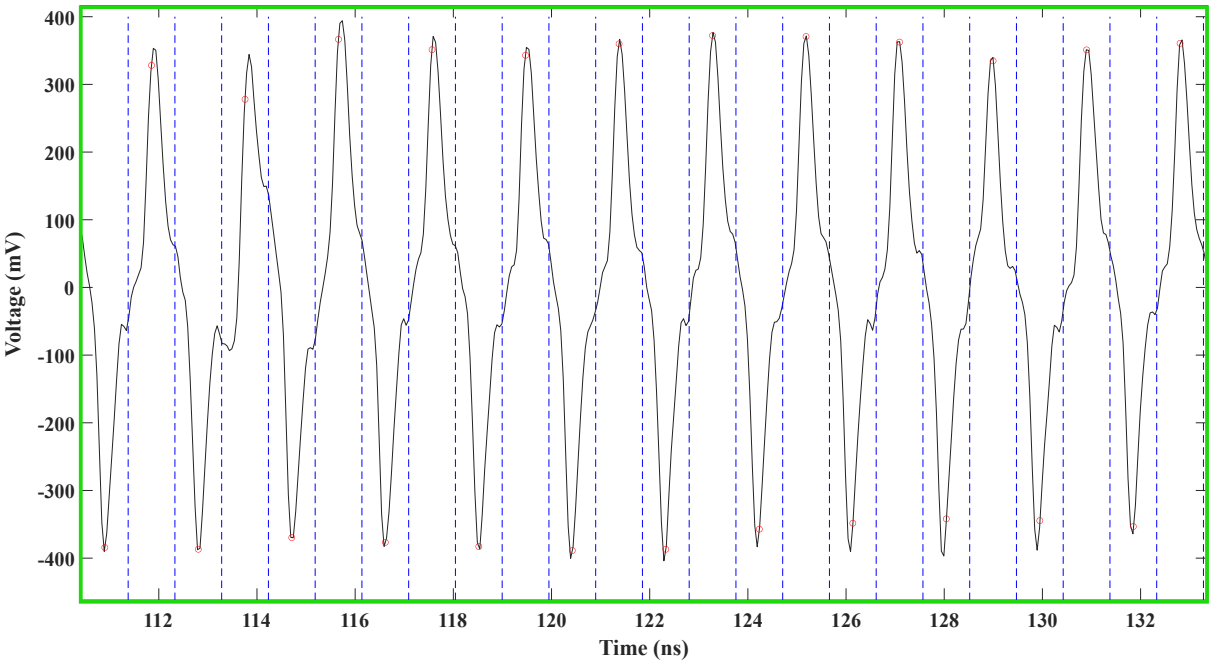
Having established the importance of interpolation, a comprehensive, step-by-step guide to the interpolation process is provided below. As a reminder, the preamble that exists before the start of data transfer, has time duration of approximately 27.3 *ns*. At the end of this 27.3 *ns* preamble pattern, a negative peak appears, marking the boundary between the preamble and the data transfer. This negative peak signifies the true start time of the data transfer. It is measured that the total duration for sending the whole image (784 pixels) is approximately 372.4 *ns*. By adding 372.4 *ns* to the true start time, the full length of the data trace is determined. Subsequently, interpolation is performed on the full length data trace signal. The initial sampling rate at the oscilloscope output is 50 *ps*, so the signal is first interpolated down to 1 *ps* sampling rate using interpolation formula. Upsampling the rate to 1 *ps* allows precise adjustment of the peak position to fit the theoretical window of 0.95238 ns with good exceptional accuracy. Since the information to be retrieved from the signal is the bit transitions (*i.e.*, HD), the time window for scanning during correlation is set to 0.95238 ns. Based on 1 *ps* sampling resolution, the peaks are then aligned to the center of each window interval, which is 0.47619 ns. Once this alignment procedure is done, the trace is downsampled back to 50 *ps*, making it convenient for correlation.

With the alignment procedure completed, the structure of the newly adjusted signal is examined in Fig. 5.10 to see if the shift still exist or not. The same signal is shown again in Fig. 5.10a after decimation, demonstrating the better outcome from this alignment process. It is observed that there is minimal difference or shift between the theoretical peaks and the actual peaks, as revealed in Fig. 5.10b.

Furthermore, the peak versus transition relationship is plotted again using the well-adjusted signal, as shown in Fig. 5.11. It is clear that while a small shift still exists along the trace, it is significantly reduced compared to the previous case, now being only the average of 0.1ns. Besides, the nature of the shift alternates over time, switching between earlier and later than theoretical peaks. This alternating pattern effectively compensates



(a) EM trace showing the captured signal with vertical dotted blue lines representing the scanning windows and red circles indicating the theoretical peak positions.



(b) Zoomed-in view of (a) for clear visualization of shift in theoretical peak position and real peak.

Figure 5.10 – Alignment of real peaks to the theoretical window **after** interpolation.

for the shifts overall, ensuring improved alignment and stability in the adjusted signal.

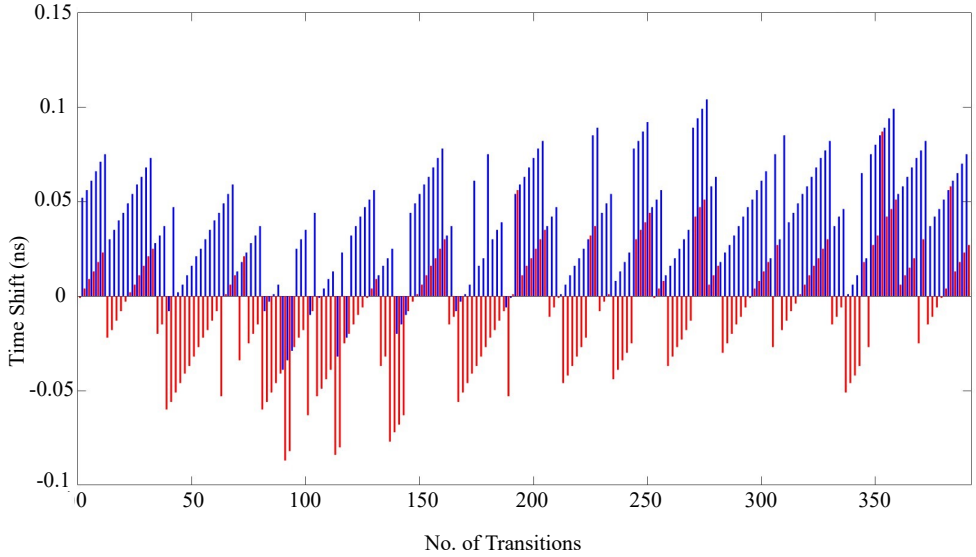


Figure 5.11 – Time shift vs transition plot to analyze the delay in peaks after interpolation

In conclusion, it is critical to ensure that the peaks in the data trace fall at the center of each scanning window. However, in some cases, the peaks on the data trace do not naturally align at the center of the scanning window due to the original sampling rate. Such misalignment can lead to a cumulative time shift across the time-domain signals, resulting in inaccuracies during correlation and ultimately degrading the quality of bit transitions retrieval. Due to this reason, the preparation stage is important as it serves as a foundation for efficient correlation performance in the following stage. After completing these two preparations, the data is ready and primed to proceed to the correlation operation.

5.5 Correlation Stage

Correlation is a mathematical method for comparing two signals or datasets to determine how similar they are. It is performed by assessing how well one signal or dataset aligns with another when one is shifted across the other. This process is notably useful for checking patterns or specific features, even in the case of noisy signals or complex datasets. In this methodology, correlation works by measuring how similar the derived HD template is to the preprocessed data trace at every half a clock cycle (scanning window) throughout the entirety of data trace, selecting the most correlated pattern out of all elements of database template. This similarity is quantified using the pearson correlation

coefficient, ρ, which is calculated with the following formula:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where: x_i represents the values from the HD template, y_i represents the values from the data trace, $\bar{x}$ and $\bar{y}$ are the mean values of x and y, respectively. The ρ ranges from -1 to 1, where:

$\rho = 1$ (Perfect positive linear correlation indicates that the two datasets are highly similar.)

$\rho = -1$ (Perfect negative linear correlation indicates that the two datasets are inversely related.)

$\rho = 0$ (No linear correlation indicates that there is no linear relationship between the datasets.)

By systematically scanning through the data trace using correlation and choosing the maximum ρ value from all element of the template data base, the HD estimates at each half cycle are retrieved from the induced voltage in the data EM trace. However, additional conditions are put into place in the correlation stage to optimize performance. A threshold on the level of peak in each window is defined to filter out unnecessary correlations. Since the noise floor of the data trace is already known through the Gaussian analysis described in section 4.5.1, the correlation operation is skipped if the peak within a window interval falls below the defined threshold, *i.e.*, between 64 mV and -72 mV. In other words, correlation is performed only when the peak is significant enough to not be classified as noise. Another condition to consider is the polarity of the peak. The polarity (whether positive or negative) is examined prior to the actual correlation, to filter out which elements in the database are correlated. For example, if the polarity of the peak is positive, the correlation will only be performed with the positive HD possibilities in the database template as seen in Fig. 5.6a. Similarly, if the peak is negative, only the negative HD possibilities, illustrated in Fig. 5.6b will be considered for correlation. This approach ensures that irrelevant database entries are excluded, improving the efficiency of the correlation process.

Since the threshold and polarity of peaks are discussed in the HBIR method, it is

important to clarify the differences between the two approaches. While both methodologies share the same threshold (as they are based on the same EM trace) and both examine the polarity of peaks, the distinction lies in what follows afterwards. In HBIR, the focus is restricted to three possibilities: no flipping of bits (HD = 0), positive peaks representing ascending bit transitions (HD = 16), and negative peaks representing descending bit transitions (HD = -16). On the contrary, for SIR method, the threshold and polarity verification serve as supporting factor to optimize the correlation and refine the results. However, the primary operator for retrieving bit transition information from the EM data trace in this method, is the correlation itself, which provides a more robust analysis.

After correlation is applied to the full length of data trace, the database element that yields the highest coefficient ρ is selected for each time window of 0.95238 ns. These recovered HD values form the HD image, which is then used in the image reconstruction stage.

HD Image

The HD represents the transition between the previous clock cycle, occurring at either the rising or falling edge of the clock, and the current clock cycle during the transmission of a 16-bit packet of data. It is characterized in terms of the net number of positive or negative bit-flips, which quantify the changes in bit states. The HD transition image serves as an intermediate representation prior to the final reconstruction of the input images from the EM traces. It also constitutes the core information required for reconstruction stage. The ground truth serves as the reference HD image, derived directly from the original input image. To construct the ground truth [2], the following process is undertaken. The original 28x28 input image is processed pixel by pixel, where each pixel value is converted into its binary representation. Next, the HD is calculated for every consecutive pair of two pixels. Each pair of pixels represents a 16-bit sequence, and the HD is computed as the bitwise difference between these two binary values. In addition to the magnitude of HD, the ground truth also includes the signs for HD, which provides additional information about the direction of bit transitions. The direction of the transition refers to the order that the bit flips, *i.e.* from 0 to 1 (ascending) or 1 to 0 (descending). After HD is calculated, the sign of the HD is determined based on the relative count of the direction of transitions. If the number of ascending transitions exceeds the number of descending transitions, the

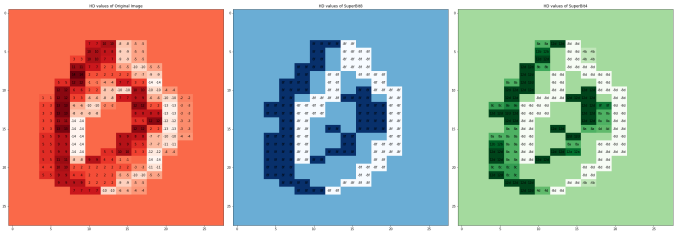
2. A reference data that is confirmed to be true based on measurements or direct observations [126]

HD is assigned a positive sign. On the other hand, if the number of descending transitions is greater, the HD is assigned a negative sign. Once all signed HD values are calculated, they are mapped to a 28x28 image again to form HD image of the ground truth. This image represents the ideal HD values for the input image and serves as the standard for comparison during evaluation.

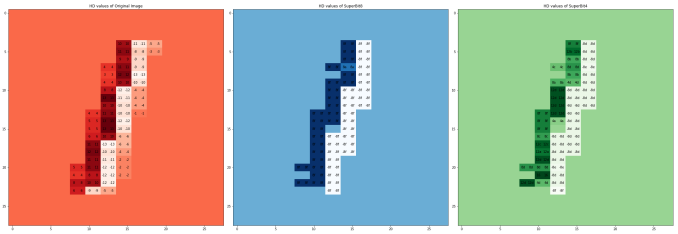
Following the correlation process, the most correlated HD value at each window is retrieved in accordance with maximum coefficient. Nevertheless, it is important to note that the total number of pixels for a single input image sent through the bus is 784, corresponding to 28x28 original dimension. Given that the retrieved information represents the bit transition, *i.e.* the difference of the two consecutive bytes at each half clock cycle, the total number of transitions retrieved by the correlation is only 392. However, to make a side-by-side comparison with the ground truth and the retrieved HD, these 392 transition values are doubled to fit into the 28x28 image dimension. Fig. 5.12 and 5.13 reveals the HD image for 10 different images in MNIST while Fig. 5.14 and 5.15 represents 10 different images in fashion-MNIST dataset.

Fig. 5.12 5.13 5.14 and 5.15 compare the ground truth HD image to the HD images recovered using Super-Bits of 8 and Super-Bits of 4. The orange image represents the ground truth HD values derived directly from the original image and serves as a reference to evaluate the accuracy of the recovered HD values. The blue image shows the HD obtained through the correlation operator using Super-Bits of 8. These values are less detailed due to the coarser granularity of Super-Bits of 8, resulting in a more approximate representation of the actual HD transitions. In contrast, the green image represents the HD values recovered using Super-Bits of 4. The regions in the all three images where no numbers are displayed correspond to default HD values of 0, indicating no bit transitions occurred in those locations.

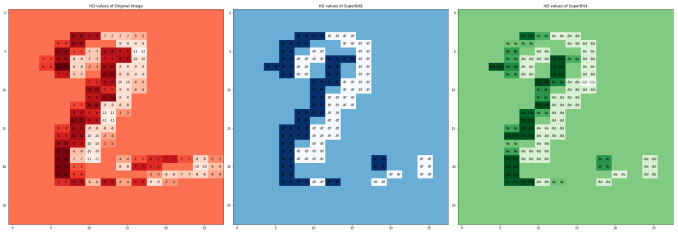
Furthermore, it is observed that for images with distinct shapes, such as shoes (heels, sneakers, or boots), the HD image alone provides sufficient information to infer the content of the image. However, in cases involving objects with similar shapes, such as t-shirts and pullovers, the HD representation becomes less distinguishable with bare eyes, leading to potential confusion in identifying the content. This limitation underscores the difficulty of reconstructing the HD images back into original pixel-based images.



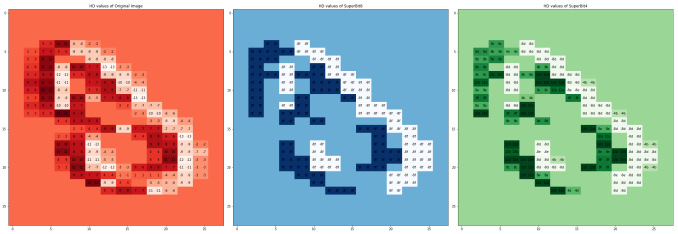
(a) MNIST 0



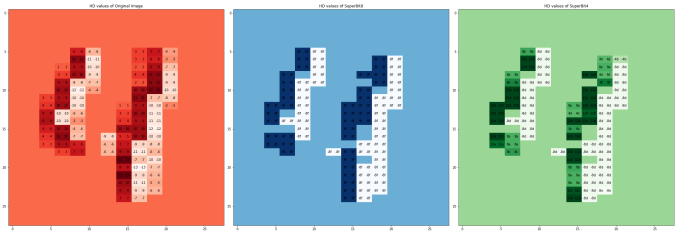
(b) MNIST 1



(c) MNIST 2

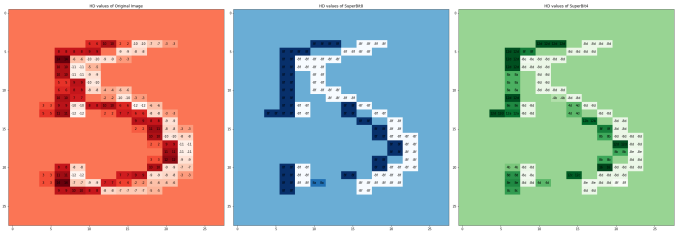


(d) MNIST 3

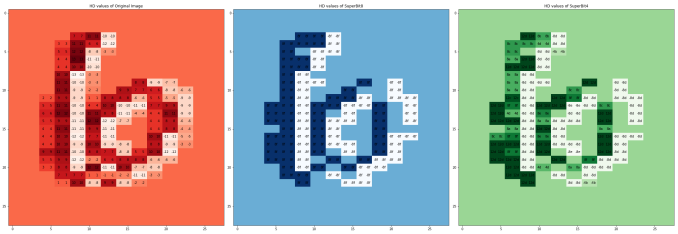


(e) MNIST 4

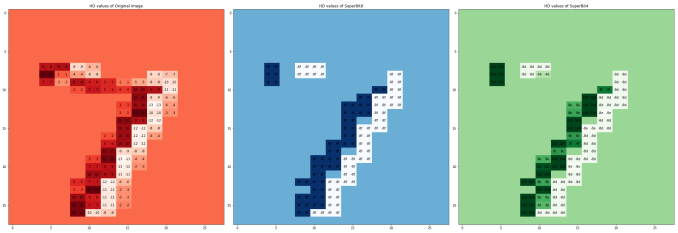
Figure 5.12 – HD images for Super-Bits of 4 and 8 when MNIST 0, 1, 2, 3, and 4 images are sent as input. Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4.



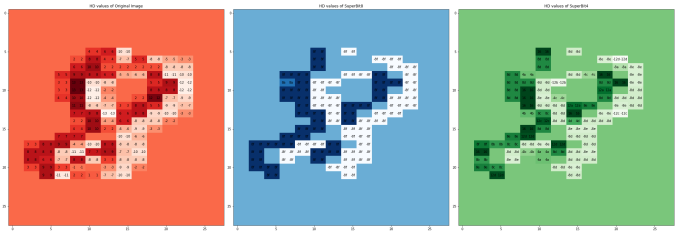
(a) MNIST 5



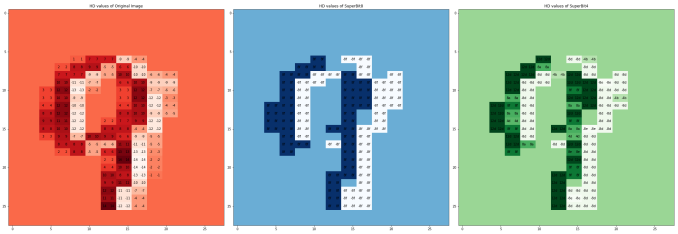
(b) MNIST 6



(c) MNIST 7

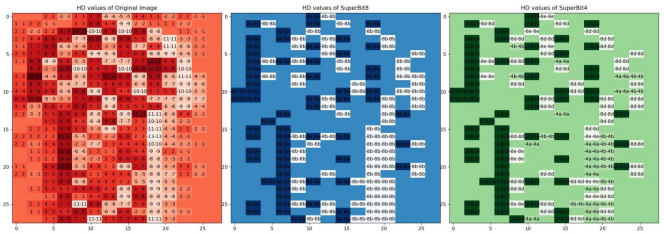


(d) MNIST 8

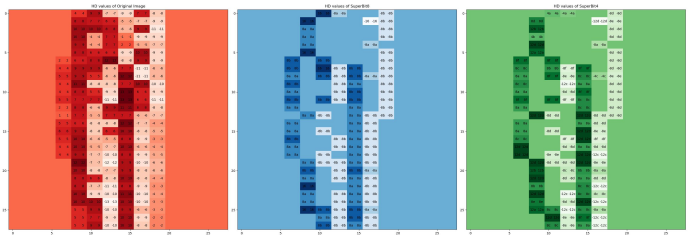


(e) MNIST 9

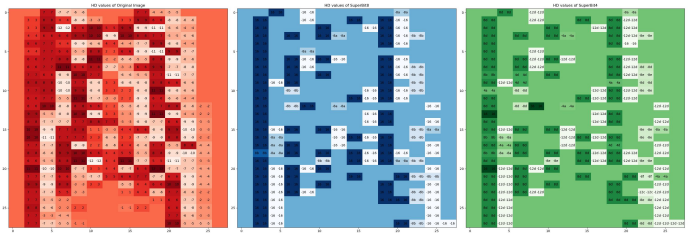
Figure 5.13 – HD images for Super-Bits of 4 and 8 when MNIST 5, 6, 7, 8 and 9 images are sent as input.Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4.



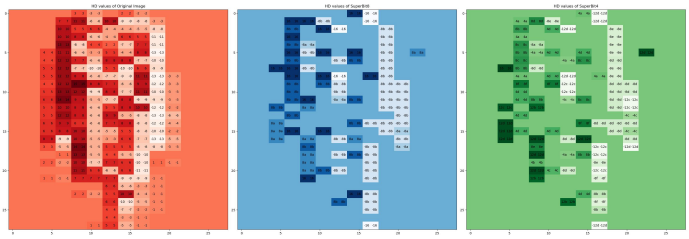
(a) Fashion MNIST label 0 (T-shirt) as input



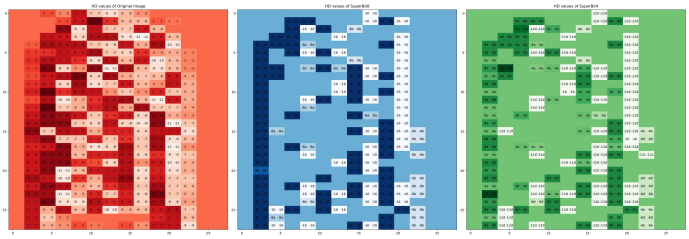
(b) Fashion MNIST label 1 (pants) as input



(c) Fashion MNIST label 2 (pullover) as input

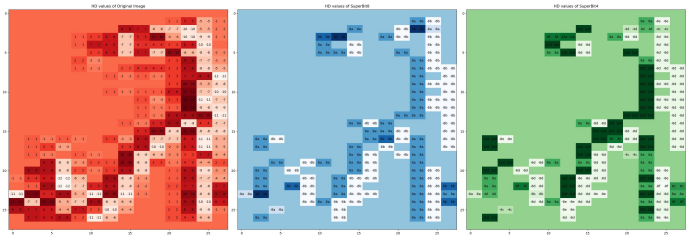


(d) Fashion MNIST label 3 (dress) as input

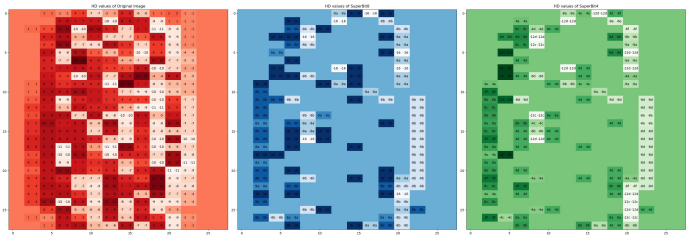


(e) Fashion MNIST label 4 (coat) as input

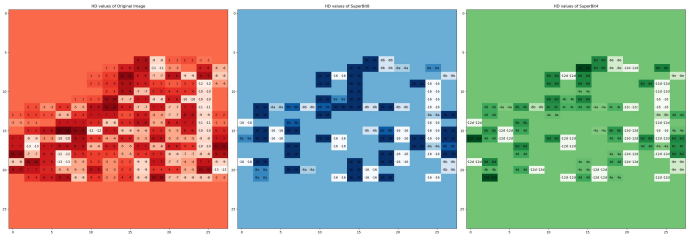
Figure 5.14 – HD images for Super-Bits of 4 and 8 when Fashion-MNIST 0, 1, 2, 3 and 4 images are sent as input. Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4.



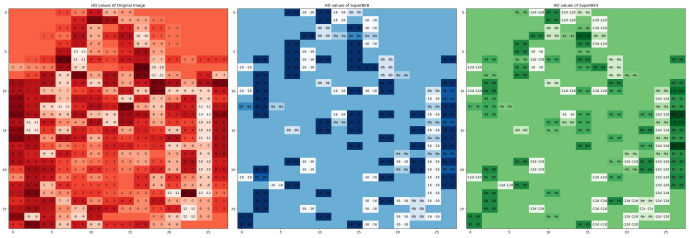
(a) Fashion MNIST label 5 (sandals) as input



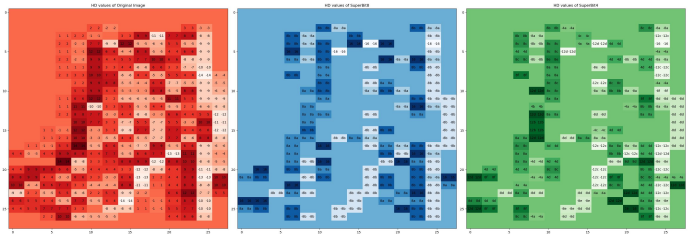
(b) Fashion MNIST label 6 (shirt) as input



(c) Fashion MNIST label 7 (sneakers) as input



(d) Fashion MNIST label 8 (bag) as input



(e) Fashion MNIST label 9 (ankle boots) as input

Figure 5.15 – HD images for Super-Bits of 4 and 8 when Fashion-MNIST 5, 6, 7, 8 and 9 images are sent as input. Left: Ground Truth, Middle: Super-Bits of 8, Right: Super-Bits of 4.

5.6 Reconstruction Stage

After the bit transitions are retrieved in the form of the HD image, the reconstruction of the input image intended for the accelerator is performed. The image reconstruction phase begins by initializing a base image, made of 28x28 pixels, filled entirely with black color, serving as a background to adapt the specificity of the datasets. This assumption is valid because the attacker knows that the type of input image is grayscale and the datasets have black pixels on borders. The image reconstruction works by first checking the HD from the image, and then depending on its value, assign the pixel value at the base image. A raster scan is then performed along the full length of HD image. During this scan, if no transition is detected, *i.e.* $HD = 0$, the pixel value is kept at 0 (black). However, if any non-zero HD is detected, two key factors need to be considered:

— The direction of the transition —whether the transition is ascending (0 to 1) or descending (1 to 0). This is determined by the sign of the HD with the + represents the bit flip from 0 to 1 and - refers to as the flip from 1 to 0.

— The specific bits within the 16-bit sequence that are responsible for the transition (*i.e.*, the location of the bits flipped). It is decided by the selected element from the database template that is highly correlated.

This process ensures that the captured traces are reconstructed onto the 28x28 image, effectively reversing the intermediate results, (retrieved HD information) back to the original input image. The recovered images for both MNIST and Fashion-MNIST using this reconstruction algorithm, based on the results of their corresponding HD images are presented in Fig. 5.16, 5.17, 5.18 and 5.19.

5.7 Evaluation of Results from Methodology 2

5.7.1 Transition Recovery

In order to evaluate the quality of the HD retrieved, PSNR and HD-level difference are employed as metrics for comparing ground truth to the computed HD image.

HD-level difference refers to the average of absolute difference between the HD image of original image (ground truth) and the HD image generated by selecting the highest coefficient value from correlation process. In this case, the metric evaluates the HD retrieval process by quantifying the deviation between ground truth and the achieved HD values.

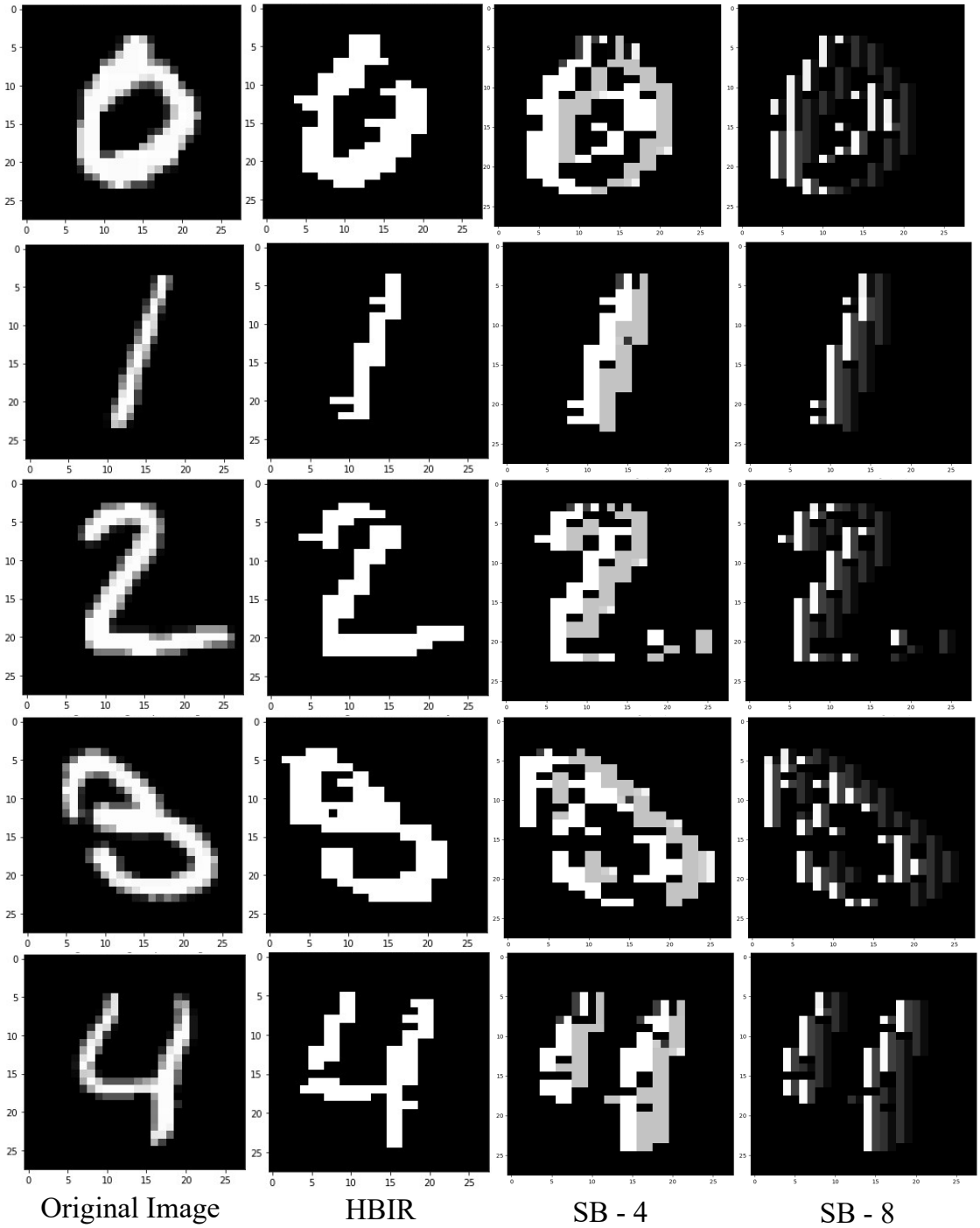


Figure 5.16 – Comparison of original input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8.

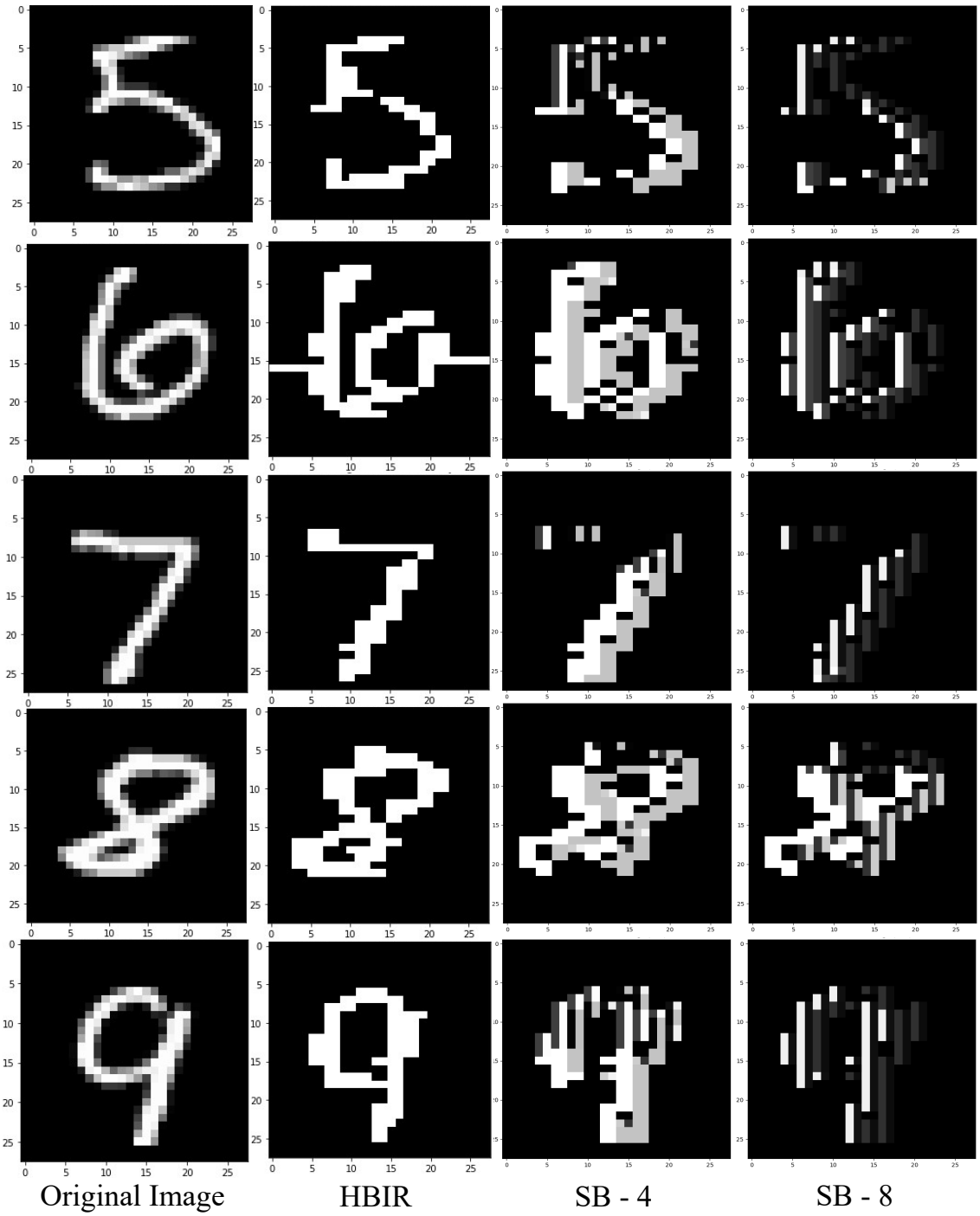


Figure 5.17 – Comparison of original input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8.

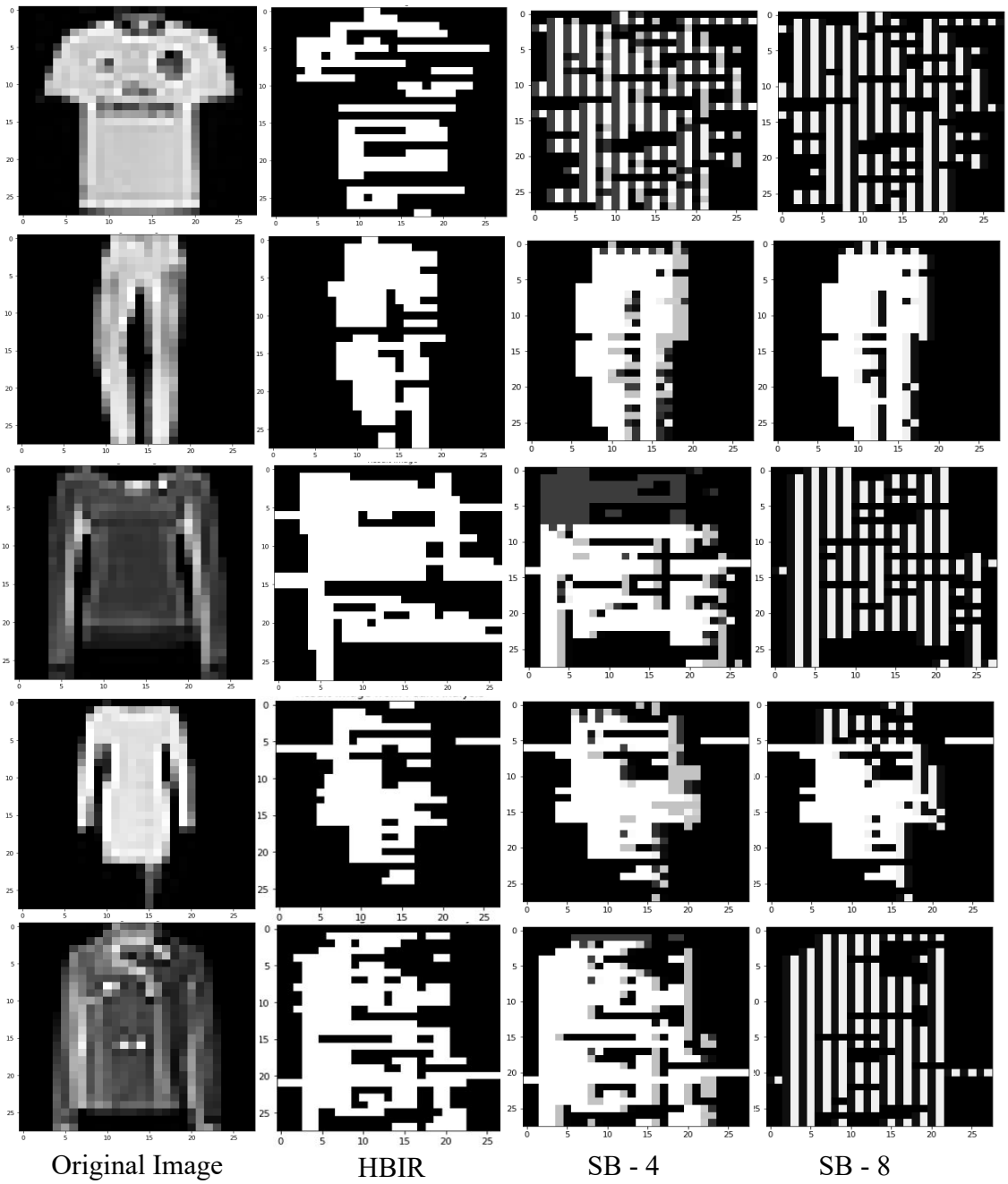


Figure 5.18 – Comparison of orginal input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for Fashion-MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8.

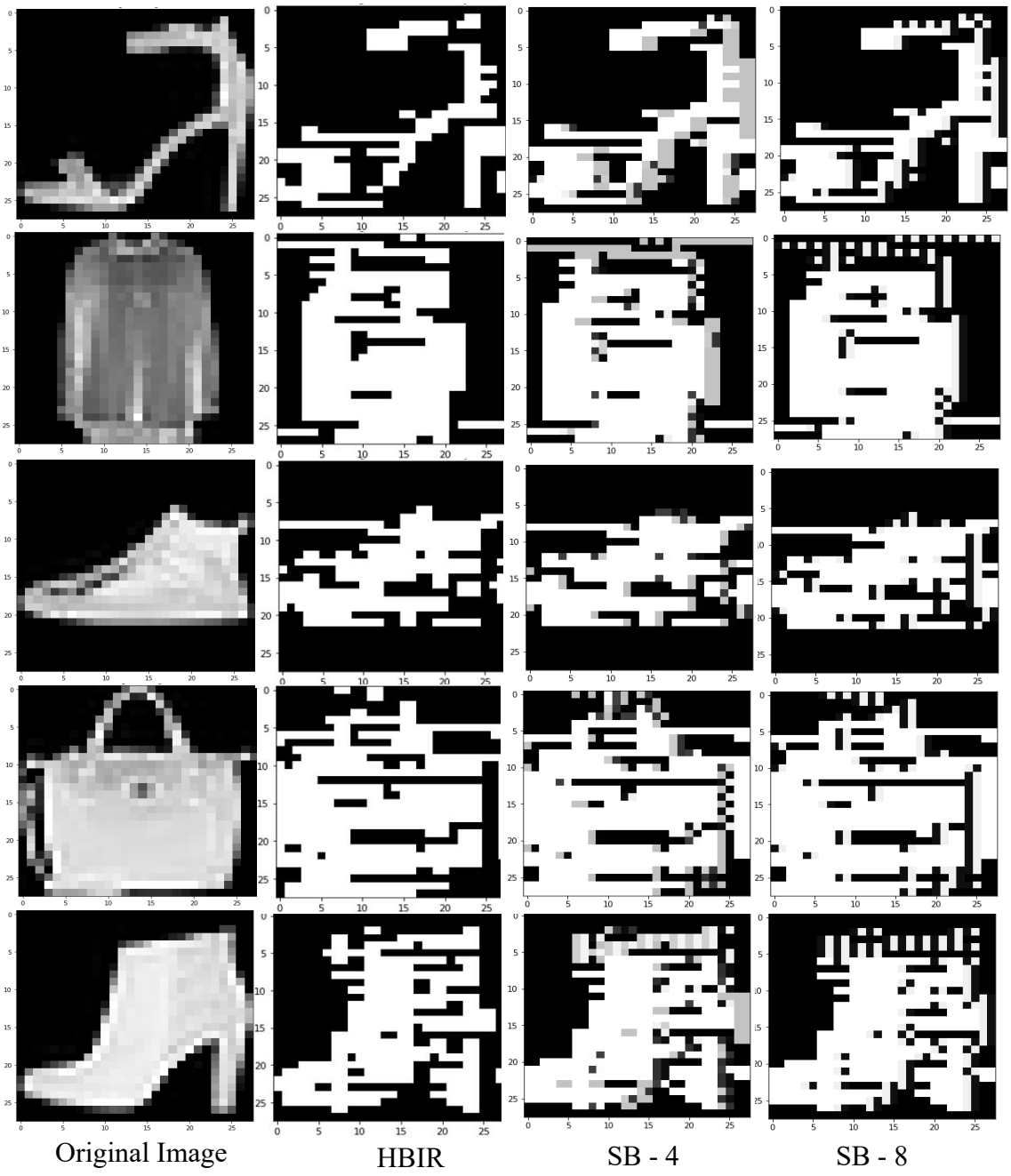


Figure 5.19 – Comparison of orginal input image with the recovered images from different methodologies, HBIR, Super-Bits of 4 and Super-Bits of 8 for Fashion-MNIST images. Left to Right: Input image to inference, recovered image by HBIR, recovered image by Super-Bits of 4 and recovered image by Super-Bits of 8.

Lower HD-level difference value indicates that the retrieved HD from the correlation are closer with the ground truth. The equation for HD-level difference is as follows:

$$\frac{1}{N}\sum_{i=1}^{N}|HD_o - HD_r|$$

On the other hand, PSNR calculates the pixel-wise differences between the original and the new images. Primarily, it quantifies the error introduced during processing. The measurement unit of PSNR is in decibels (dB). Higher dB values stands for better quality of the image while lower dB values signify noticeable distortion. The reference for PSNR is as follows:

- PSNR $> 40\,\text{dB}$: Excellent
- $30\,\text{dB} <$ PSNR $< 40\,\text{dB}$: Good
- $20\,\text{dB} <$ PSNR $< 30\,\text{dB}$: Moderate
- PSNR $< 20\,\text{dB}$: Poor

According to metrics results from Table. 5.5 and 5.6, Super-Bits of 4 consistently achieves the highest in both HD-level difference and PSNR for both datasets. Its finer granularity enables better image detail and structure compared to Super-Bits of 8 and HBIR.

Super-Bits of 4: Table. 5.5 shows that out of all three methods, Super-Bits of 4 provides least difference between ground truth and retrieved HD. This indicates that the HD values retrieved using Super-Bits of 4 are the closest to the original HD image. The same observation holds for the metric results with Fashion-MNIST as reported in Table. 5.6. The performance difference between Super-Bits of 4 and the rest of the method is more obvious in Fashion-MNIST images which proves that Super-Bits of 4 is a better choice for recovering complex grayscale images with various pixel intensity across the whole image. As for PSNR metric, the performance gap between Super-Bits of 4 and other methods is more pronounced in MNIST, where it benefits more from the simple shape of the digits. For Fashion-MNIST, superiority of Super-Bits of 4 is less prominent due to the increased complexity of transitions. In particular, the very dark backgrounds of Fashion-MNIST images are not composed solely of zero-value pixels. Instead, they often contain very low-intensity values (below 10), which appear black in the image. However, as discussed in Chapter 4, HBIR method is already proven to function well on standard MNIST images. The objective here is to deploy a method which performs effectively on more naturalistic grayscale images like Fashion-MNIST. The results in the Table. indicate

that the performance gap between Super-Bits of 4 and HBIR is by 1. Meanwhile, for Fashion-MNIST, this difference increases to 3, highlighting the superior performance of Super-Bits of 4. Under all circumstances, Super-Bits of 4 consistently yields the best results for both metrics.

Super-Bits of 8: Super-Bits of 8 prioritizes efficiency over granularity, therefore, with its smaller granularity, it struggles to capture the finer details in Fashion-MNIST images, resulting in greater loss compared to MNIST. As for the comparison between Super-Bits of 8 and HBIR, they both give nearly identical results with PSNR while Super-Bits of 8 shows a slight improvement in terms of HD-level difference for both datasets. This is because the HBIR simplifies the transition by mapping each peak directly to either +16 or -16, which reduces the variety of possible HD values. On the other hand, the Super-Bits of 8 allows for sightly more granular distinction by incorporating extra possible transitions such as ±8 and ±16. This marginally broader range of HD in Super-Bits of 8 contributes a more detailed representation of the retrieved HD image, potentially capturing HD values that HBIR overlook.

Table 5.3 – Average of metric values on 10,000 MNIST test images

Metric	**HBIR**	**SB4**	**SB8**
HD-level Difference	2	0.85	2
PSNR (dB)	35.72	35.72	35.72

Table 5.4 – Average of metric values on 10,000 Fashion-MNIST test images

Metric	**HBIR**	**SB4**	**SB8**
HD-level Difference	5.2	3.2	4.5
PSNR (dB)	33.60	33.60	33.60

From the Table. 5.3, the average HD-level difference for the 10,000 MNIST test set is 0.85 for Super-Bits 4, while for Super-Bits 8 and HBIR, the average is 2. For the 10,000 Fashion-MNIST test data, the average HD-level difference is 3.2 for Super-Bits of 4, 4.5 for Super-Bits of 8, and 5.2 for HBIR. As indicated in Table. 5.4, the average PSNR for MNIST for all three methods remains consistent at approximately 35.72, while for Fashion-MNIST, it is around 33.60. It is concluded that the reason for the difference in performance between three methods lies in the limitation in provided HD value that each method can represent. Super-Bits of 8 can only offer HD of ±8 and ±16, while HBIR exclusively represents ±16. Since the range of HD available in both methods is very

similar, it is reasonable that their performance metrics also show minimal differences. This highlights that the finer granularity provided by Super-Bits of 4 (with HD values of ±4, ±8, ±12, ±16) is necessary to achieve a noticeable improvement in recovery of HD from the peaks in the EM traces.

Table 5.5 – MNIST Performance Metrics

Digit	**Method**	**HD-Level Difference**	**PSNR**
0	HBIR	2.21	35.51
	SB4	1.21	40.09
	SB8	2.21	35.51
1	HBIR	1.09	38.12
	SB4	0.55	42.81
	SB8	1.09	38.12
2	HBIR	1.82	36.27
	SB4	0.81	41.43
	SB8	1.82	36.27
3	HBIR	2.67	34.36
	SB4	1.41	38.67
	SB8	2.67	34.36
4	HBIR	1.97	35.78
	SB4	0.96	40.53
	SB8	1.97	35.78
5	HBIR	1.78	36.25
	SB4	0.91	40.77
	SB8	1.78	36.25
6	HBIR	2.49	34.93
	SB4	1.29	39.54
	SB8	2.49	34.93
7	HBIR	1.35	37.88
	SB4	0.75	41.97
	SB8	1.35	37.88
8	HBIR	2.19	35.25
	SB4	1.02	40.50
	SB8	2.19	35.25
9	HBIR	1.90	36.31
	SB4	1	40.98
	SB8	1.90	36.31

Table 5.6 – Fashion-MNIST Performance Metrics

Label	**Method**	**HD-Level Difference**	**PSNR**
0 T-shirt/top	HBIR	5.96	30.60
	SB4	3.93	33.87
	SB8	5.96	30.60
1 Trouser	HBIR	3.40	33.67
	SB4	1.97	37.14
	SB8	3.38	33.70
2 Pullover	HBIR	6.22	32.52
	SB4	3.96	36.41
	SB8	4.45	30.12
3 Dress	HBIR	3.80	32.75
	SB4	2.74	34.93
	SB8	3.58	32.24
4 Coat	HBIR	6.37	31.87
	SB4	4.18	32.82
	SB8	4.78	29.92
5 Sandal	HBIR	4.01	32.20
	SB4	2.19	36.54
	SB8	4.01	32.20
6 Shirt	HBIR	6.11	30.63
	SB4	4.35	32.85
	SB8	5.75	30.63
7 Sneaker	HBIR	3.70	33.58
	SB4	4.06	35.31
	SB8	3.11	32.18
8 Bag	HBIR	6.02	32.08
	SB4	4.06	33.52
	SB8	4.95	30.45
9 Ankle Boots	HBIR	5.16	31.31
	SB4	3.10	35.13
	SB8	4.98	30.99

Pixel Recovery

In order to evaluate the recovered MNIST images, the same classifier described in section 4.6, is deployed. The classifier successfully predicted all the labels of the images (in Fig. 5.16, 5.17) recovered by SIR. It is important to recall that the motivation for developing the SIR method is to handle more sophisticated grayscale images, such as those in the Fashion-MNIST dataset. Therefore, the focus of the subsequent discussion will primarily be on the results for Fashion-MNIST.

To check the recognition ability of the 1000 recovered images, the NN model [127] for classifying Fashion-MNIST images is used to predict the labels. The model architecture consists of flattening a layer to convert the input image into a 1D array, followed by a dense layer with 128 neurons and ReLU activation. The output layer includes 10 neurons, each representing a label in Fashion-MNIST while using softmax activation for classification. This evaluation model is initially trained to be achieve recognition accuracy up to 84% on 10,000 Fashion-MNIST testing dataset.

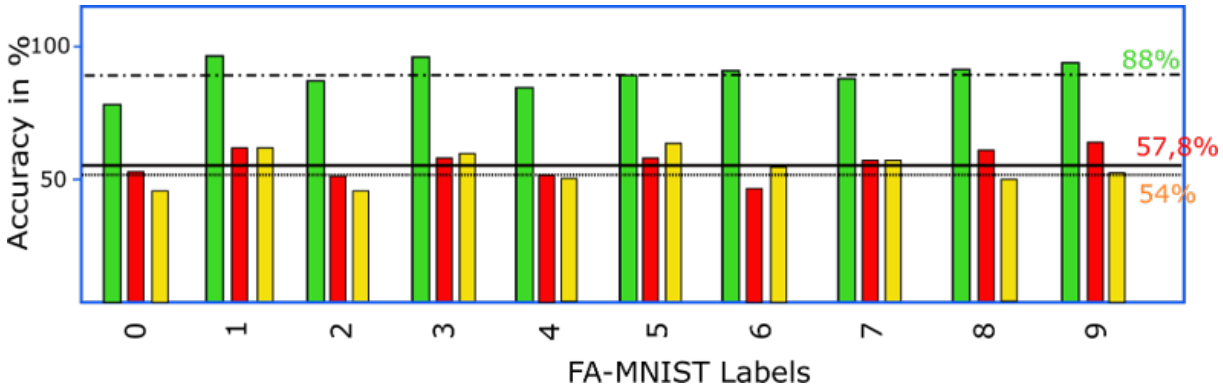


Figure 5.20 – Recognition accuracy of SIR method (in red) with respect to the HBIR (in yellow) and the accuracy of original Fashion-MNIST images (in green). This accuracy calculation is based on 1000 Fashion-MNIST images from test dataset.

For the purpose of validating the effectiveness of image reconstruction, the validation is conducted to calculate the recognition accuracy using the same procedure introduced in Chapter 4. First, 1000 images from testing dataset of Fashion-MNIST are chosen. The average recognition accuracy of these selected original images reached 88% as portrayed in Fig. 5.20. When applying the same evaluation model to the recovered images by the SIR methodology, the accuracy of 58% is achieved. In comparison, the images reconstructed via the HBIR method obtain a slightly lower accuracy of 54%. These results provide quantitative evidence that while HBIR demonstrates strong performance on simpler grayscale

images, SIR is more effective in handling more complex grayscale images such as those found in the Fashion-MNIST dataset.

5.7.2 Limitations of SIR

It is seen in Table 5.7 that Fashion-MNIST images, recovered by Super-Bits of 4 and 8, such as shirt, sneakers, ankle boots, and bag, are correctly classified by the NN. Falsely classifications primarily occur due to the high similarity in the structure of certain items. For instance, garments such as t-shirts, shirts, and pullovers are prone to being incorrectly labeled due to overlapping shape characteristics inherent to these categories of clothing. Similarly, items such as trousers and dresses are prone to misclassification, as both garments produce elongated shapes in the images, leading to ambiguity in recognition. The classification results also expose that the NN perceives minimal difference between images recovered by Super-Bits 4 and 8. In majority of the cases, both sets of images are assigned the same labels, suggesting that the NN predictions remain consistent regardless of whether the images are recovered using Super-Bits 4 or 8.

Original Label	**Predicted Label (HBIR)**	**Predicted Label (SB4)**	**Predicted Label (SB8)**
0 T-shirt	Shirt	Shirt	Pullover
1 Trousers	Trousers	Dress	Dress
2 Pullover	Shirt	Coat	T-Shirt
3 Dress	Pullover	Shirt	Shirt
4 Coat	Pullover	Shirt	Pullover
5 Sandals	Ankle Boots	Ankle Boots	Ankle Boots
6 Shirt	Pullover	Shirt	Shirt
7 Sneakers	Bag	Sneakers	Sneakers
8 Bag	Coat	Bag	Bag
9 Ankle Boots	Ankle Boots	Ankle Boots	Ankle Boots

Table 5.7 – Comparison of arbitrarily selected original labels with predicted labels from SB-4, SB-8, and HBIR models

5.8 Comparison Between Methods: HBIR and SIR

When comparing SIR with HBIR, it is seen that the images reconstructed using the former approach achieve slightly better recognition performance. This improvement is attributed to the way each method handles edge detection, which plays an essential role in image classification NN.

In HBIR, detection of the edges is performed by identifying the margin between background (dark pixels) and foreground (light pixels) using thresholds values as described in section 4.5.1. However, this approach is sensitive to noise and outliers, which cause the threshold to misinterpret the edges, leading to classification errors. It is worth noting that the HD values change most significantly at these edges, as the transitions of pixels occur the most in these regions. On the other hand, SIR detects edges more robustly by deploying correlation rather than thresholding. This ensures that even minimal differences between the background and foreground are captured more reliably. Even though the reconstructed images by SIR appear incomplete with visual inspection, as illustrated in Fig. 5.18 and 5.19, they retain enough information about the shape of the item for the classifying NN to recognize precisely.

It is important to note that the reconstructed images produced by SIR utilize a simple construction of the images. While they may not appear visually promising, their edges are sufficient for NN classification. Furthermore, the quality of these reconstructed images in Fig. 5.18 and 5.19 could be enhanced with additional image processing techniques, potentially improving their visual appearance. For instance, the authors of [60] use denoising algorithm to their initially recovered images to improve its quality. While our work does not employ various image processing techniques, since it falls outside the scope of the study, incorporating such methods could be an interesting direction for future research, with the potential to optimize the visual quality of reconstructed images.

5.8.1 Limitations

Before discussing the limitations, it is crucial to understand the current process of the work and identify potential future opportunities. As depicted in Fig. 5.21, the left side shows the victim's bus system, which functions normally, transmitting data through a SoC-FPGA buses. On the right-hand side, the attacker's process is illustrated, where the trace captured from the victim SoC-FPGA is reverse-engineered to reconstruct the original image. This diagram provides an overview of how the attacker spies on the victim bus and processes its EM side-channels to retrieve valuable input data. On the victim side, the original input image is transmitted through on-chip bus, resulting in a data stream. The stream of data changes per half clock cycle which leads to the transitions of bits, also known as HD. These transitions cause the current derivatives on the conducting wires, that can be detected with the use of H-field probe. The probe circuit transforms the H-fields into electrical signals which are visualized as time-domain traces on the oscilloscope.

The right side of the Fig. 5.21, represents the reverse engineering the the acquired EM trace. The raw trace is first synchronized to get precise data transfer time and then interpolated. This processed data trace is correlated with database template to get HD image, which is also an intermediate result. Based on that, the recovered image which resembles the original image of the victim is reconstructed.

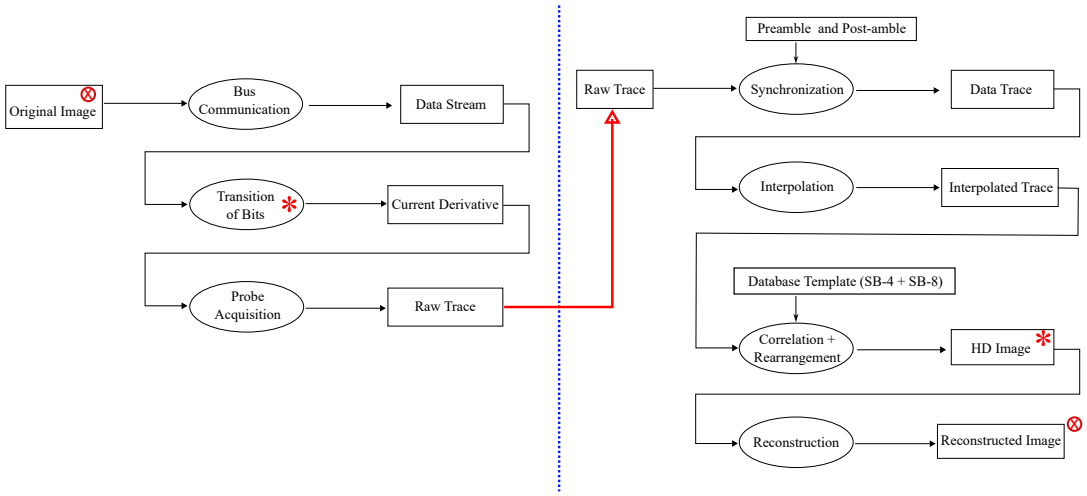


Figure 5.21 – Overall process of how input image data (transferred on the bus inside victim's hardware system) is reversed by using SIR.

There are a few challenge and limitation that need to be addressed in this methodology. It is important to note that the retrieved information obtained from the trace represents transitions, indicating the number of bits that have changed from one state to another, rather than the original pixel values. This limitation introduces ambiguity, making it impossible to retrieve the exact original image. To address this, during the reconstruction stage, the initial pixel value is assumed to be 0 (black), given the grayscale nature of the image. The subsequent pixel values are calculated by iteratively applying the transitions encoded in the HD image and integrating transitions. Although this approach compensates for the absence of absolute information, some degree of information loss is inherent to the methodology.

Another information loss arises from the limitations of the database template used during the correlation process. The templates contain only a specific set of HD values: 4, 8, 12, and 16 for Super-Bits of 4, and 8 and 16 for Super-Bits of 8. Thus, any HD that does not match these predefined templates are approximated to the closest available value in the database. This approximation introduces additional inaccuracies, as it replaces the

true HD with the closely related representation from the template. Moreover, MSB and LSB are mixed in Super-Bits. Confusion of MSB and LSB transitions results in strong grayscale value changes in the reconstructed images that do not appear in ground truth images.

The third source of information loss comes from the algorithm assigning pixel values in pairs rather than individually. This occurs due to the fact that the retrieved HD values reflect the transitions between pairs of two pixels, rather than a single pixel. This pairing is a direct consequence of the bus structure, which transmits two bytes at each edge of the clock. As a result, the algorithm cannot resolve transitions to the level of individual pixels, making a pixel pair the finest level of granularity achievable in the reconstruction process. This structural limitation leads to a loss of detail and adds to the overall information loss in the methodology.

A final consideration is that while the intermediate HD images MNIST and Fashion-MNIST, the final image recovery process is validated on a smaller representative sample of 10 images from each dataset. This smaller test set is chosen as a proof of concept to demonstrate the feasibility of the proposed recovery method. Expanding this evaluation to larger test dataset of 10,0000 images is left as future work.

Comparison with "I Know what You See" by L. Wei et al.

A comparison is drawn between our proposed methodologies and a closely related state-of-the-art approach, "I Know What You See" by Wei et al. [58], higlighting both similarities and differences in the underlying approaches.

- **Type of Side-channel**: A key difference between the work of Wei et al. and ours lies in the type of the exploited side-channel, which, in their case, are power traces, whereas our approach is based on EM emanations. As discussed in section 2.4, EM analysis is preferred over power in this thesis due to its clearer identification of bit transitions. Unlike power traces, where distinguishing between transitions is subtle between ascending and descending bit transitions, EM traces provide a more evident distinction through the polarity of peaks, making them more robust for signal interpretation.
- **Used Dataset**: The two works also differ in terms of the dataset used to apply the methodology. Wei et al. conduct their experiments only on the standard MNIST dataset, which consists of simple, grayscale handwritten digits. On the other hand,

we apply our method to exploit more complex Fashion-MNIST dataset, which features grayscale images of clothing items and presents a more challenging classification task due to their greater pixel variability.

- **Template Building**: The process of building templates from the side-channel signals also differs between the two approaches. In the work of Wei et al., a total of 300 power traces are used to build their template which facilitates the estimation of pixel from the consumed power. However, SIR method necessitate only 15 EM traces to build database template which serves as reference for correlation. This reduction in the number of traces required to build template makes our method more efficient in terms of time and data collection.
- **Evaluation Scale**: The scale of evaluation is different between the two works. Wei et al. evaluate their methodology on a test set of 200 images, whereas our current evaluation has been conducted on a smaller test set of 10 images.
- **Information Retrieval Process**: Wei et al. directly retrieve pixel values from the collected power traces by selecting the closest pixel candidate in comparison with the template. With our methodologies, an intermediate results *i.e.* the HD between two groups of two-pixel sets are first achieved. After that, the final pixel values are derived from these HD values. This additional extraction of intermediate results allow our method to adapt to different types of data leakage, without being restricted to pixel data.
- **Accuracy**: Finally, the reported accuracy of the two methods differs significantly. Wei et al. report an accuracy ranging from 79% to 89% for their image recovery and classification. In comparison, our SIR achieves an accuracy ranging from 20% to 40%. However, it is important to note that our accuracy is based on a smaller test set of 10 images, and our method operates on a more complex dataset (Fashion-MNIST). These factors suggest that while our reported accuracy is lower, our method has the potential to improve with further optimization and larger-scale evaluations.

Both approaches share a common objective of recovering images from side-channel, but they differ significantly in the methodology and evaluation. While Wei et al. achieve higher accuracy on a simpler dataset, SIR demonstrates broader applicability by extending beyond MNIST to complex grayscale datasets, such as Fashion-MNIST. In terms of template building, the two methods vary considerably in the number of traces needed to construct a reference template. The substantial reduction in the number of traces re-

quired for SIR not only increases the efficiency of the method but also makes it practical for real-world applications where data collection time is limited.

5.9 Chapter Conclusion

This chapter has introduced a novel methodology, called SIR to enhance the bus-transferred data retrieval from EM traces. By incorporating correlation analysis and the concept of Super-Bits grouping, the approach improves both granularity and precision in recovering bit transitions. While HBIR and its reliance on integrating peaks along the trace makes it convenient in a high-contrast datasets, such as MNIST, it faces challenges in complex datasets like Fashion-MNIST, where pixel intensity variations are more subtle. Super-Bits address these challenges by reorganizing 16-bit sequences to optimize EM signal characterization.

Critical steps to perform this proposed methodology, including synchronization, interpolation, along with the creation of database template, are described in this chapter. These processes ensure detailed alignment of EM trace signal and accurate correlation at each window to extract HD. From the HD image, the pixels are recovered using reconstruction algorithm, completing the reverse engineering procedure. SIR is evaluated on MNIST and Fashion-MNIST datasets. Super-Bits of 4, with its finer granularity, delivers superior results on metrics evaluated on HD image. Super-Bits of 8, although computationally efficient, perform similarly to HBIR due to its coarser representation and limited range of HD values. As for the evaluation of final recovered images, two image classification NNs, one for MNIST and the other for Fashion-MNIST are used and the results are discussed afterwards. Despite inherent information losses, the reconstructed images demonstrate the method's potential for reverse engineering EM leakage signals. This chapter concludes by acknowledging the limitations such as information loss. In the next chapter, the conclusion of this work is presented by summarizing the contributions and proposing future directions.

Chapter 6

Conclusion and Perspectives

Contents

6.1 Contributions Summary 154
6.2 Perspectives and Future Work 155

This thesis explores the EM side-channel vulnerabilities of HW accelerators, such as SoC-FPGAs and exploit information from them. Nowadays, as HW accelerators are widely used in domains that require high computational efficiency and low latency, the security risks associated with these systems have become a critical concern and need to be addressed. In this work, we show that such HW vulnerabilities reveal two main valuable assets: the physical position of conducting bus wires inside the SoC and the data being transferred on the bus. Instead of directly obtaining the actual pixel values transmitted on the bus, we observe the transitions of bits, *i.e.* the number of bits that change at each clock cycle, known as the HD from the EM signature. Depending on the location of the probe loop, certain bits are more sensitive than the others, allowing us to pinpoint which group of X (group of 4 bits for Super-Bits of 4 and group of 8 bits for Super-Bits of 8) bits have flipped in the data sequence.

We implement these findings on an image classification application, particularly on a basic image classification MLP. In this case, the input data type to the MLP is the pixel, hence, we retrieve HD between consecutive pixels over clock cycles. From the analysis of these recovered HD, we reconstruct the EM traces back to original image using HBIR and super-bits methodologies. By addressing these hardware-level vulnerabilities, this work emphasizes the need for more secure implementations in embedded systems.

6.1 Contributions Summary

Our work underscores the HW-level vulnerabilities of SoC-FPGA systems by demonstrating that their data communication buses emit detectable EM traces due to time-varying currents on the conducting wires. In Chapter 3, we illustrate that localized EM traces generated by these buses are detected using a near-field H-probe, as detailed in the cartographies of Section 3.4.2. This approach not only identifies the exploitable leakage zone (POI) but also reveals the physical arrangement of bus wires within the chip by analyzing the variation in strength of the emitted H-field with respect to the location of the probe loop. In other words, we captured EM traces along the bus wires within the POI and observed how these traces varied with the probe's position. Based on these observations, we proposed a hypothetical layout of the wires, as illustrated in Fig. 3.27.

Furthermore, by analyzing the same EM traces collected with the probe, we extract HD values. Notably, EM side-channels provide insights not only into the number of bits that have changed (HD) but also the direction of the transitions—whether from an inactive

'0' to an active '1' or vice versa. While power side-channels can also hint at the direction of transitions through slight differences in power consumption (e.g., the transition from 0 to 1 consumes slightly more power than from 1 to 0 [1]), EM traces reveal this information more clearly due to the polarity of the signal peaks, which can be positive or negative.

We characterize how internal buses within SoC-FPGA emit EM leakage and proposed two novel attacks to recover input data from these emissions. Our experiments demonstrate that sensitive information, such as input images, are reconstructed without prior knowledge of the underlying NN architecture. Using a controlled experimental setup, we implemented these attacks on a commonly used Pynq-Z1 board and evaluated them on two grayscale datasets: MNIST and Fashion-MNIST. It was observed that HBIR is sufficiently effective for the complete recovery of standard grayscale MNIST images, primarily due to the high contrast between the background and the foreground. However, for more complex grayscale images, such as Fashion-MNIST, which exhibit varied pixel intensities throughout the entire image, HBIR may not be a suitable approach. This limitation is addressed by the Super-Bits method, which is specifically designed for estimating lower contrasts between consecutive group of bytes. The Super-Bits method focuses on retrieving finer scale HD values, rather than coarse approximations such $\pm$ 16, making it slightly more suited for handling the intricate details in Fashion-MNIST images. Depending on the granularity of Super-Bits, even finer HD values can be recovered, as demonstrated with Super-Bits of 8 and 4. Besides, the success of these methodologies as one-shot attacks highlights the efficiency of exploiting EM emanations as a non-invasive means of compromising security. Our results validate the feasibility of recovering input data, such as image pixels, through EM leakage analysis, emphasizing a critical HW-level vulnerability in on-chip buses.

6.2 Perspectives and Future Work

The findings discussed in this work underscore the need for a new framework in the HW designs of modern accelerators, especially for SoC-FPGA implementations. The ability to retrieve bit transitions on the bus wires, and further extend these intermediate results to recover image pixels, through observed EM leakages, highlights the imperative for incorporating adequate security mechanisms in HW designs.

The vulnerability explored in this thesis arises from the HW, specifically the SoC bus,

1. The transition from 0 to 1 consumes slightly more power compared to the one from 1 to 0.

making this weakness inherent and independent of the application. In our case, the application is an image classification NN, which is why our objective is first on retrieving the intermediate result, represented as HD image, and then reconstructing back to the original images from it. However, in other applications, the exploitable asset may differ. Any type of information transmitted through this vulnerable SoC bus remains at risk of being exploited via EM side-channels. This opens up the possibility of applying our methodology to other types of applications beyond NNs. Another possibility is to investigate bus vulnerabilities on other HW platforms which in turn, broaden the scope of this research.

EM traces are localized, *i.e.* the emanation originates directly from the conducting bus wires and varies depending on the relative position of the bus to the loop of the probe. In our experiment, even with a single probe, we are able to detect the position of the bits on the bus lines and identify the physical routing of the wires. It indicates that placing two probes, one at the edge of the wire cluster and another on the opposite side, can provide supplementary information for more accurate exploitation of HD. For instance, when a probe is positioned on the right side of a wire group, bits closer to the right side exhibit greater leakage compared to those on the left side. By putting a second probe on the left side, complementary leakage patterns can be measured, thereby enriching the information available for further analysis. This multiple probe setup allows for a more comprehensive database template for correlation, enabling the retrieval of finer and more precise HD values from the EM traces. This approach has not been tested in the current thesis, but it represents a potential avenue for future work that could further enhance the precision and accuracy of the EM trace analysis.

Moreover, another future work involves expanding the range of possible HD in the database templates. We only utilize Super-Bits of 8 and Super-Bits of 4 in this work, however, for future research, we can investigate the performance of Super-Bits of 2, where bits are grouped in special pairs. Additionally, the approach can be extended further by creating template elements for each individual HD. By including both positive and negative signs for each HD, this would result in 32 template elements in the database. If two probes are implemented in the experiment, this allows for the achievement of a finer granularity in HD correlation, This potential approach represents another direction for future research.

While we successfully reconstruct the images from the HD data using a basic and straightforward reconstruction approach, the quality of the recovered images are reason-

ably good, especially around the edges of the shapes. However, to improve the overall quality of the reconstructed images, applying advanced image processing techniques becomes a possibility. For instance, denoising algorithms or low-pass filters can be used to reduce noise and improve the clarity of the reconstructed images. In addition, morphological operations such as opening and closing can be explored to further enhance the results.

There are two approaches for mitigating the risks associated with FPGAs: either by preventing the leakage of sensitive information through side-channels or by disrupting the link between the leaked information and the sensitive data. Future research could explore improving existing techniques for masking information and developing new strategies to disrupt the link between leaked data and sensitive information. This work belongs to the field of side-channel analysis, demonstrating the potential of leveraging EM emissions for information retrieval and the importance of developing robust countermeasures to safeguard against such vulnerabilities.

APPENDIX

6.3 Approximation of Biot-Savart's Law and Hertz Dipole in the Near Field

1. Biot-Savart's Law

Biot-Savart's law describes the magnetic field produced by a small current element $I\,d\mathbf{L}$:

$$d\mathbf{B} = \frac{\mu_0}{4\pi}\frac{I\,d\mathbf{L}\times\hat{\mathbf{r}}}{R^2}$$

The magnitude of the magnetic field is:

$$dB = \frac{\mu_0}{4\pi}\frac{I\,dL\,\sin\theta}{R^2}$$

where: μ_0: Permeability of free space (H/m), I: Current in the conductor (A), dL: Length of the current element (m), $\sin\theta$: Angle between $d\mathbf{L}$ and $\hat{\mathbf{r}}$ and R: Distance from the current element (m).

2. Hertz Dipole

For a time-varying current I in a small dipole, the magnetic field H_ϕ is:

$$H_\phi = \frac{I\Delta L\sin\theta}{4\pi r^2}(1+jkr)e^{-jkr}$$

where: $\omega = 2\pi f$: Angular frequency (rad/s), L: Length of the dipole (m), $k = \frac{2\pi}{\lambda}$: Wavenumber (rad/m) and λ: Wavelength (m). .

3. Near-Field Approximation ($R \ll \lambda$)

In the near field ($kR \ll 1$), the dominant term in the magnetic field is:

$$B_\phi \approx \frac{\mu_0 I L\omega\sin\theta}{4\pi R^2}$$

4. Comparison of Biot-Savart and Hertz Dipole

For quasi-static conditions ($\omega \to 0$), the Hertz dipole simplifies to:

$$B_\phi \approx \frac{\mu_0 I L \sin\theta}{4\pi R^2}$$

which matches the form of the Biot-Savart law:

$$B \propto \frac{I\, dL\, \sin\theta}{R^2}$$

In the near field ($R \ll \lambda$), Biot-Savart's law and the Hertz dipole are equivalent for the magnetic field, as both describe the $\frac{\sin\theta}{R^2}$-dependent behavior of the field. This equivalence holds when radiative effects are negligible and the reactive component dominates.

BIBLIOGRAPHY

[1] João Silva, Valery Sklyarov, and Iouliia Skliarova, « Comparison of On-chip Communications in Zynq-7000 All Programmable Systems-on-Chip », *in*: *IEEE Embedded Systems Letters* 7.*1* (2015), pp. 31–34, DOI: 10.1109/LES.2015.2399656.

[2] Berardino Carnevale et al., « A flexible system-on-a-chip implementation of the Advanced Encryption Standard », *in*: *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, 2016, pp. 156–161, DOI: 10.1109/ICSTCC.2016.7790658.

[3] May Myat Thu et al., « Bus Electrocardiogram: Vulnerability of SoC-FPGA Internal AXI Buses to Electromagnetic Side-Channel Analysis », *in*: *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe 2023)*, Kraków, Poland, Sept. 2023, Paper ID: 249.

[4] May Myat Thu et al., « You Only Get One-Shot: Eavesdropping Input Images to Neural Network by Spying SoC-FPGA Internal Bus », *in*: *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, Benevento, Italy: Association for Computing Machinery, 2023, ISBN: 9798400707728.

[5] Keman Huang, Michael Siegel, and Stuart Madnick, « Systematically Understanding the Cyber Attack Business: A Survey », *in*: *ACM Comput. Surv.* 51.*4* (July 2018), ISSN: 0360-0300, DOI: 10.1145/3199674.

[6] Zhuolin Li et al., « Denial of Service (DoS) Attack Detection: Performance Comparison of Supervised Machine Learning Algorithms », *in*: *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 469–474, DOI: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00088.

[7] Paolo Prinetto and Gianluca Roascio, « Hardware Security, Vulnerabilities, and Attacks: A Comprehensive Taxonomy », *in*: *Italian Conference on Cybersecurity*, 2020, URL: https://api.semanticscholar.org/CorpusID:216554981.

[8] Sonia Akter, Kasem Khalil, and Magdy Bayoumi, « A Survey on Hardware Security: Current Trends and Challenges », *in*: *IEEE Access* 11 (2023), pp. 77543–77565, DOI: 10.1109/ACCESS.2023.3288696.

[9] Stefan Mangard, Elisabeth Oswald, and Thomas Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, 1st, Springer Publishing Company, Incorporated, 2010.

[10] Maria Sommerhalder, « Hardware Security Module », *in*: *Trends in Data Protection and Encryption Technologies*, ed. by Valentin Mulder et al., Cham: Springer Nature Switzerland, 2023, pp. 83–87, ISBN: 978-3-031-33386-6, DOI: 10.1007/978-3-031-33386-6_16.

[11] *In*: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2024.*2* (Mar. 2024), pp. 714–734, DOI: 10.46586/tches.v2024.i2.714-734.

[12] Siddharth Prakash Rao, Gabriela Limonta, and Janne Lindqvist, « Usability and security of trusted platform module (TPM) library APIs », *in*: *Proceedings of the Eighteenth USENIX Conference on Usable Privacy and Security*, SOUPS'22, Boston, MA, USA: USENIX Association, 2022, ISBN: 978-1-939133-30-4.

[13] F. Rosenblatt, *The perceptron - A perceiving and recognizing automaton*, tech. rep. 85-460-1, Ithaca, New York: Cornell Aeronautical Laboratory, Jan. 1957.

[14] Kevin Gurney, *An Introduction to Neural Networks*, USA: Taylor & Francis, Inc., 1997, ISBN: 1857286731.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, http://www.deeplearningbook.org, MIT Press, 2016.

[16] Christian Szegedy et al., « Going deeper with convolutions », *in*: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9, DOI: 10.1109/CVPR.2015.7298594.

[17] Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

[18] Tamador Mohaidat and Kasem Khalil, « A Survey on Neural Network Hardware Accelerators », *in*: *IEEE Transactions on Artificial Intelligence* 5.*8* (2024), pp. 3801–3822, DOI: 10.1109/TAI.2024.3377147.

[19] Charu C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, 1st, Springer Publishing Company, Incorporated, 2018.

[20] Vinod Nair and Geoffrey E. Hinton, « Rectified linear units improve restricted boltzmann machines », *in*: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, Haifa, Israel: Omnipress, 2010, pp. 807–814, ISBN: 9781605589077.

[21] Alex Smola and S.V.N. Vishwanathan, *Introduction to Machine Learning*, Also affiliated with the Departments of Statistics and Computer Science, Purdue University, and the College of Engineering and Computer Science, Australian National University, Santa Clara: the Press Syndicate of the University of Cambridge, 2008.

[22] Y. Lecun et al., « Gradient-based learning applied to document recognition », *in*: *Proceedings of the IEEE* 86.*11* (1998), pp. 2278–2324, DOI: 10.1109/5.726791.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, « ImageNet Classification with Deep Convolutional Neural Networks », *in*: *Advances in Neural Information Processing Systems*, ed. by F. Pereira et al., vol. 25, Curran Associates, Inc., 2012.

[24] Andrew G. Howard et al., « MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications », *in*: *CoRR* abs/1704.04861 (2017), arXiv: 1704.04861.

[25] Hussain Mohammed Dipu Kabir et al., « SpinalNet: Deep Neural Network with Gradual Input », *in*: *CoRR* abs/2007.03347 (2020), arXiv: 2007.03347.

[26] Karen Simonyan and Andrew Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », *in*: *CoRR* abs/1409.1556 (2014).

[27] Alexey Dosovitskiy et al., « An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale », *in*: *ArXiv* abs/2010.11929 (2020).

[28] Mingxing Tan and Quoc V. Le, « EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks », *in*: *ArXiv* abs/1905.11946 (2019).

[29] Christian Szegedy et al., « Rethinking the Inception Architecture for Computer Vision », *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, DOI: 10.1109/CVPR.2016.308.

[30] Kaiming He et al., « Deep Residual Learning for Image Recognition », *in*: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, DOI: 10.1109/CVPR.2016.90.

[31] Farhan Shafiq et al., « Automated flow for compressing convolution neural networks for efficient edge-computation with FPGA », *in*: (Dec. 2017), DOI: 10.48550/arXiv.1712.06272.

[32] Sanjay Patel and Wen-mei W. Hwu, « Accelerator Architectures », *in*: *IEEE Micro* 28.*4* (2008), ISSN: 0272-1732, DOI: 10.1109/MM.2008.50.

[33] Tony Nowatzki et al., « Stream-Dataflow Acceleration », *in*: *SIGARCH Comput. Archit. News* 45.*2* (June 2017), pp. 416–429, ISSN: 0163-5964, DOI: 10.1145/3140659.3080255.

[34] William J. Dally, Yatish Turakhia, and Song Han, « Domain-specific hardware accelerators », *in*: *Commun. ACM* 63.*7* (June 2020), pp. 48–57, ISSN: 0001-0782, DOI: 10.1145/3361682.

[35] Biagio Peccerillo et al., « A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives », *in*: *Journal of Systems Architecture* 129 (2022), p. 102561, ISSN: 1383-7621, DOI: https://doi.org/10.1016/j.sysarc.2022.102561.

[36] Li Du and Yuan Du, « Hardware Accelerator Design for Machine Learning », *in*: Sept. 2018, ISBN: 978-1-78923-752-8, DOI: 10.5772/intechopen.72845.

[37] Massimo Alioto, « Trends in Hardware Security: From Basics to ASICs », *in*: *IEEE Solid-State Circuits Magazine* 11.*3* (2019), pp. 56–74, DOI: 10.1109/MSSC.2019.2923503.

[38] Jordan Robertson and Michael Riley, *The Big Hack: How China Used a Tiny Chip to Infiltrate America's Top Companies*, Accessed: 2023-12-27, Oct. 2018, URL: https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies.

[39] Tiago Perez et al., « Side-Channel Trojan Insertion - a Practical Foundry-Side Attack via ECO », *in*: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5, DOI: 10.1109/ISCAS51556.2021.9401481.

[40] Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri, « A Primer on Hardware Security: Models, Methods, and Metrics », *in*: *Proceedings of the IEEE* (2014).

[41] Mohammad Tehranipoor and Farinaz Koushanfar, « A Survey of Hardware Trojan Taxonomy and Detection », *in*: *IEEE Design & Test of Computers* (2010).

[42] R. Elbaz et al., « Hardware Engines for Bus Encryption: A Survey of Existing Techniques », *in*: *Designer's Forum of IEEE DATE'05* (Oct. 2007), DOI: 10.1109/DATE.2005.170.

[43] Kamel Abdelouahab et al., « Accelerating CNN inference on FPGAs: A Survey », *in*: *CoRR* (2018).

[44] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, « Differential Power Analysis », *in*: *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, Lecture Notes in Computer Science, Springer, 1999.

[45] Li Deng, « The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web] », *in*: *IEEE Signal Processing Magazine* (2012).

[46] Han Xiao, Kashif Rasul, and Roland Vollgraf, *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*, Aug. 28, 2017, arXiv: cs.LG/1708.07747 [cs.LG].

[47] Wim van Eck, « Electromagnetic radiation from video display units: An eavesdropping risk? », *in*: *Computers & Security* (1985).

[48] Markus G. Kuhn, « Electromagnetic Eavesdropping Risks of Flat-Panel Displays », *in*: PET'04, Toronto, Canada: Springer-Verlag, 2004.

[49] Paul C. Kocher, « Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems », *in*: *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, vol. 1109, Lecture Notes in Computer Science, Springer, 1996.

[50] Vincent Meyers, Dennis Gnad, and Mehdi Tahoori, « Active and Passive Physical Attacks on Neural Network Accelerators », *in*: *IEEE Design & Test* (2023).

[51] Umesh Sehgal and Kamaljeet Singh, « Eavesdropping Attack Problem Threat in Cyber Network Security and Wireless Sensor Network », *in*: *International Journal of Development Research* 07.*01* (Jan. 2017), pp. 11177–11180.

[52] Verizon Business, *What Are Eavesdropping Attacks?*, https://www.verizon.com/business/resources/articles/s/what-are-eavesdropping-attacks/, Accessed: 2025-01-05, 2025.

[53] Qinghua Wang, « Defending wireless communication against eavesdropping attacks using secret spreading codes and artificial interference », *in*: *Computers & Security* 103 (2021), p. 102175, ISSN: 0167-4048, DOI: https://doi.org/10.1016/j.cose.2020.102175.

[54] Marcin Gregorczyk et al., « Sniffing Detection Based on Network Traffic Probing and Machine Learning », *in*: *IEEE Access* 8 (2020), pp. 149255–149269, DOI: 10.1109/ACCESS.2020.3016076.

[55] « A Survey of Man in the Middle Attacks », English, *in*: *I E E E Communications Surveys and Tutorials* 18.*3* (2016), pp. 2027–2051, ISSN: 1553-877X, DOI: 10.1109/COMST.2016.2548426.

[56] Cezar Reinbrecht et al., « Gossip NoC – Avoiding Timing Side-Channel Attacks through Traffic Management », *in*: *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2016.

[57] Ayomide Akinsanya and Tegan Brennan, « Timing Channels in Adaptive Neural Networks. », *in*: *Proceedings of the 2024 Network and Distributed System Security Symposium, San Diego, California, USA* (2024).

[58] Lingxiao Wei et al., « I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators », *in*: 2018.

[59] Yun Xiang et al., « Open DNN Box by Power Side-Channel Attack », *in*: *CoRR* (2019).

[60] Shayan Moini et al., « Remote Power Side-Channel Attacks on CNN Accelerators in FPGAs », *in*: 2020.

[61] Vincent Meyers, Dennis Gnad, and Mehdi Tahoori, « Reverse Engineering Neural Network Folding with Remote FPGA Power Analysis », *in*: *2022 IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2022.

[62] Lukas Huegle et al., « Power2Picture: Using Generative CNNs for Input Recovery of Neural Network Accelerators through Power Side-Channels on FPGAs », *in*: *2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2023.

[63] Yicheng Zhang et al., « Stealing Neural Network Structure Through Remote FPGA Side-Channel Analysis », *in*: *IEEE Transactions on Information Forensics and Security* (2021).

[64] Saurav Maji, Utsav Banerjee, and Anantha P. Chandrakasan, « Leaky Nets: Recovering Embedded Neural Network Models and Inputs Through Simple Power and Timing Side-Channels—Attacks and Defenses », *in*: *IEEE Internet of Things Journal* (2021).

[65] Ya Gao et al., « NNLeak: An AI-Oriented DNN Model Extraction Attack through Multi-Stage Side Channel Analysis », *in*: *2023 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, 2023.

[66] Kota Yoshida et al., « Model Reverse-Engineering Attack using Correlation Power Analysis against Systolic Array Based Neural Network Accelerator », *in*: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[67] Daniel Genkin, Itamar Pipman, and Eran Tromer, « Get your hands off my laptop: physical side-channel key-extraction attacks on PCs: Extended version », *in*: *Journal of Cryptographic Engineering* (2015).

[68] Honggang Yu et al., « DeepEM: Deep Neural Networks Model Recovery through EM Side-Channel Information Leakage », *in*: *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2020.

[69] Ville Yli-Mäyry et al., « Extraction of Binarized Neural Network Architecture and Secret Parameters Using Side-Channel Information », *in*: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[70] Sudarshan Sharma et al., « SNATCH: Stealing Neural Network Architecture from ML Accelerator in Intelligent Sensors », *in*: *2023 IEEE SENSORS*, 2023.

[71] Raphaël Joud et al., « Like an Open Book? Read Neural Network Architecture with Simple Power Analysis on 32-Bit Microcontrollers », *in*: *Smart Card Research and Advanced Applications*, ed. by Shivam Bhasin and Thomas Roche, Cham: Springer Nature Switzerland, 2024, pp. 256–276, ISBN: 978-3-031-54409-5.

[72] Lejla Batina et al., « CSI NN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel », *in*: *28th USENIX Security Symposium (USENIX Security 19)*, USENIX Association, 2019.

[73] Jian Liu et al., « Snooping Keystrokes with mm-level Audio Ranging on a Single Phone », *in*: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, New York, NY, USA: Association for Computing Machinery, 2015.

[74] Sourav Panda et al., « Behavioral Acoustic Emanations: Attack and Verification of PIN Entry Using Keypress Sounds », *in*: *Sensors* (2020).

[75] Zhen Xiao et al., « Keystroke Recognition with the Tapping Sound Recorded by Mobile Phone Microphones », *in*: *IEEE Transactions on Mobile Computing* (2021).

[76] Ognjen Glamočanin et al., « Temperature Impact on Remote Power Side-Channel Attacks on Shared FPGAs », *in*: *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023.

[77] Michael Hutter and Jörn-Marc Schmidt, « The Temperature Side Channel and Heating Fault Attacks », *in*: *Smart Card Research and Advanced Applications: 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, Berlin, Heidelberg: Springer-Verlag, 2014.

[78] Taehun Kim and Youngjoo Shin, « ThermalBleed: A Practical Thermal Side-Channel Attack », *in*: *IEEE Access* (2022).

[79] National Security Agency, *NACSIM 5000: Tempest Fundamentals*, Partially declassified transcript, `http://cryptome.org/jya/nacsim-5000/nacsim-5000.htm`, Feb. 1982.

[80] Corentin Lavaud et al., « Whispering Devices: A Survey on How Side-channels Lead to Compromised Information », *in*: *Journal of Hardware and Systems Security* (2021).

[81] Maria Méndez Real and Rubén Salvador, « Physical Side-Channel Attacks on Embedded Neural Networks: A Survey », *in*: *Applied Sciences* (2021).

[82] Ireneusz Kubiak, « Electromagnetic Eavesdropping », *in*: Aug. 2019, ISBN: 978-1-83880-506-7, DOI: `10.5772/intechopen.86478`.

[83] Jiadi Yu et al., « An Indirect Eavesdropping Attack of Keystrokes on Touch Screen through Acoustic Sensing », *in*: *IEEE Transactions on Mobile Computing* 20.*2* (2021), pp. 337–351, DOI: `10.1109/TMC.2019.2947468`.

[84] Frank Leferink, « An Introduction to TEMPEST (Classified), using ChatGPT », *in*: *International Symposium and Exhibition on Electromagnetic Compatibility (EMC Europe 2023)*, IEEE, Sept. 2023, pp. 1–6.

[85] P. Wright, *Spycatcher: The Candid Autobiography of Senior Intelligence Officer*, Stoddart, 1987, ISBN: 9780773721685.

[86] Florian Lemarchand et al., « Electro-Magnetic Side-Channel Attack Through Learned Denoising and Classification », *in*: *CoRR* abs/1910.07201 (2019), arXiv: 1910.07201.

[87] Hao Liu et al., « USB powered devices: A survey of side-channel threats and countermeasures », *in*: *High-Confidence Computing* (2021).

[88] Y. Oren and A. Shamir, « How Not to Protect PCs From Power Analysis », *in*: *Proceedings of the Rump Session at Crypto 2006*, Rump Session, 2006.

[89] Shane Clark et al., « Current Events: Identifying Webpages by Tapping the Electrical Outlet », *in*: Sept. 2013, pp. 700–717, ISBN: 978-3-642-40202-9, DOI: 10.1007/978-3-642-40203-6_39.

[90] Patrick Cronin et al., « Charger-Surfing: Exploiting a Power Line Side-Channel for Smartphone Information Leakage », *in*: *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[91] Alexander S. La Cour, Khurram K. Afridi, and G. Edward Suh, « Wireless Charging Power Side-Channel Attacks », *in*: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2021.

[92] Leo Breiman, « Random Forests », *in*: *Mach. Learn.* 45.*1* (Oct. 2001), pp. 5–32, ISSN: 0885-6125, DOI: 10.1023/A:1010933404324.

[93] Tao Ni et al., « Exploiting Contactless Side Channels in Wireless Charging Power Banks for User Privacy Inference via Few-shot Learning », *in*: *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, Madrid, Spain: Association for Computing Machinery, 2023.

[94] Santiago Fernández et al., *Deep-TEMPEST: Using Deep Learning to Eavesdrop on HDMI from its Unintended Electromagnetic Emanations*, 2024.

[95] Pieterjan de Meulemeester, Bart Scheers, and Guy A.E. Vandenbosch, « Eavesdropping a (Ultra-)High-Definition Video Display from an 80 Meter Distance Under Realistic Circumstances », *in*: *2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)*, 2020, pp. 517–522.

[96] Zhuoran Liu et al., « Screen Gleaning: A Screen Reading TEMPEST Attack on Mobile Devices Exploiting an Electromagnetic Side Channel », *in*: *CoRR* (2020).

[97] Yan Long et al., « EM Eye: Characterizing Electromagnetic Side-channel Eavesdropping on Embedded Cameras », *in*: 2024.

[98] Karine Gandolfi, Christophe Mourtel, and Francis Olivier, « Electromagnetic Analysis: Concrete Results », *in*: *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, Berlin, Heidelberg: Springer-Verlag, 2001.

[99] Mark Zhao and G. Edward Suh, « FPGA-Based Remote Power Side-Channel Attacks », *in*: *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 229–244, DOI: 10.1109/SP.2018.00049.

[100] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi, « Template Attacks », *in*: *Cryptographic Hardware and Embedded Systems - CHES 2002*, ed. by Burton S. Kaliski, çetin K. Koç, and Christof Paar, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[101] Jun-Sheng Ng et al., « A Highly Efficient Power Model for Correlation Power Analysis (CPA) of Pipelined Advanced Encryption Standard (AES) », *in*: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[102] A. Hodjat and I. Verbauwhede, « A 21.54 Gbits/s fully pipelined AES processor on FPGA », *in*: *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004, pp. 308–309, DOI: 10.1109/FCCM.2004.1.

[103] Joseph Gravellier et al., « SideLine: How Delay-Lines (May) Leak Secrets from Your SoC », *in*: *Constructive Side-Channel Analysis and Secure Design*, Springer International Publishing, 2021.

[104] Dakshi Agrawal et al., « The EM Side-Channel(s) », *in*: *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, Lecture Notes in Computer Science, Springer, 2002.

[105] Laurent Sauvage, Sylvain Guilley, and Yves Mathieu, « Electromagnetic Radiations of FPGAs: High Spatial Resolution Cartography and Attack on a Cryptographic Module », *in*: *ACM Trans. Reconfigurable Technol. Syst.* (2009).

[106] Vincent Carlier et al., « Electromagnetic Side Channels of an FPGA Implementation of AES. », *in*: *IACR Cryptology ePrint Archive* (2004).

[107] Martin Hell and Oskar Westman, « Electromagnetic Side-Channel Attack on AES using Low-end Equipment », *in*: *ECTI Transactions on Computer and Information Technology (ECTI-CIT)* (2020), DOI: 10.37936/ecti-cit.2020142.239925.

[108] J. Longo et al., « SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip », *in*: *Cryptographic Hardware and Embedded Systems – CHES 2015*, Springer-Verlag, 2022.

[109] Debayan Das et al., « STELLAR: A Generic EM Side-Channel Attack Protection through Ground-Up Root-cause Analysis », *in*: *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2019, pp. 11–20, DOI: 10.1109/HST.2019.8740839.

[110] Minmin Jiang and Vasilis F. Pavlidis, « A Probe Placement Method for Efficient Electromagnetic Attacks », *in*: *SMACD / PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, 2021, pp. 1–4.

[111] Minmin Jiang and Vasilis F. Pavlidis, « Performance-Aware Interconnect Delay Insertion Against EM Side-Channel Attacks », *in*: *2021 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*, 2021, pp. 25–32, DOI: 10.1109/SLIP52707.2021.00013.

[112] Yuntao Liu et al., « A Survey on Side-Channel-based Reverse Engineering Attacks on Deep Neural Networks », *in*: *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022, pp. 312–315, DOI: 10.1109/AICAS54282.2022.9869995.

[113] Shayan Moini et al., « Power Side-Channel Attacks on BNN Accelerators in Remote FPGAs », *in*: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* (2021).

[114] Ian Goodfellow et al., « Generative Adversarial Nets », *in*: *Advances in Neural Information Processing Systems*, ed. by Z. Ghahramani et al., vol. 27, Curran Associates, Inc., 2014.

[115] Sisheng Liang et al., « Clairvoyance: Exploiting Far-field EM Emanations of GPU to "See" Your DNN Models through Obstacles at a Distance », *in*: *2022 IEEE Security and Privacy Workshops (SPW)*, 2022.

[116] Sonia Ben Dhia, Mohamed Ramdani, and Etienne Sicard, « Case Studies », *in*: *Electromagnetic Compatibility of Integrated Circuits: Techniques for low emission and susceptibility*, ed. by Sonia Ben Dhia, Mohamed Ramdani, and Etienne Sicard, Boston, MA: Springer US, 2006, pp. 311–394.

[117] Langer EMV, *RF-E 10 E-Field Probe (30 MHz to 3 GHz)*, Accessed: 2025-05-15, Langer EMV, 2025, URL: https://www.langer-emv.de/en/product/rf-passive-30-mhz-up-to-3-ghz/35/rf-e-10-e-field-probe-30-mhz-up-to-3-ghz/10.

[118] Langer EMV, *RF-R 3-2 H-Field Probe (30 MHz to 3 GHz)*, Accessed: 2025-05-15, Langer EMV, 2025, URL: https://www.langer-emv.de/en/product/rf-passive-30-mhz-up-to-3-ghz/35/rf-r-3-2-h-field-probe-30-mhz-up-to-3-ghz/12.

[119] Yaman Umuroglu, *Xilinx finn*, https://github.com/Xilinx/finn, 2020.

[120] Ian Goodfellow et al., « Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks », *in*: (Dec. 2013).

[121] S. P. Costa, *Data Science and Artificial Intelligence: Second Edition*, Includes CIFAR-10 dataset and discusses the impact of resource sharing on performance., Springer, 2023.

[122] Alessandro Pappalardo, *Xilinx/brevitas*, 2023, DOI: 10.5281/zenodo.3333552, URL: https://doi.org/10.5281/zenodo.3333552.

[123] Zhenpeng Shi et al., « Threat Modeling Tools: A Taxonomy », *in*: *IEEE Security & Privacy* (2022).

[124] DigiNews Desk, *A tiny spy chip the size of a rice grain could have compromised servers of Amazon, Apple, CIA and many other US federal agencies*, https://www.digit.in/news/general/a-tiny-spy-chip-the-size-of-a-rice-grain-could-have-compromised-servers-of-amazon-apple-cia-and-many-43945.html, 5 October 2018.

[125] Git-DISL, *GTDLBench: Benchmark suite for graph-based deep learning*, https://github.com/git-disl/GTDLBench, Accessed: 2024-11-01.

[126] Giles M. Foody, « Ground Truth in Classification Accuracy Assessment: Myth and Reality », *in*: *Geomatics* 4.*1* (2024), pp. 81–90, ISSN: 2673-7418, DOI: 10.3390/geomatics4010005.

[127] TensorFlow Authors, *Classification with Keras and TensorFlow*, Accessed: January 7, 2025, n.d. URL: https://www.tensorflow.org/tutorials/keras/classification?hl=fr.

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Titre : Exploitation des émanations EM d'un bus SoC-FPGA pour espionner les images traitées en interne par un accélérateur de Deep Learning

Mot clés : Compatibilité Électromagnétique, Attaques par Canal Auxiliaire, Accélérateur de Réseau Neuronal, Bus de Données

Résumé : Cette thèse étudie l'exploitation des fuites EM des bus de données pour récupérer des images d'entrée secrètes traitées par des accélérateurs NN mis en œuvre sur des SoC-FPGA. Pendant l'inférence du NN, les pixels de l'image d'entrée transmis à travers les bus internes émettent des émanations EM qui sont ensuite analysées pour reconstruire les images d'entrée originales du NN. En tirant parti d'un accès physique au système, un adversaire peut déployer une sonde en champ proche, telle qu'un cheval de Troie HW pour capturer ces traces EM émises par le bus SoC pendant la classification des images. Deux nouvelles méthodologies sont proposées pour interpréter les émanations EM en pixels d'entrée, tout en exposant les vulnérabilités matérielles critiques, y compris l'acheminement et le positionnement des fils de bus internes dans la puce. Pour valider les approches proposées, un classificateur d'images tiers, HD-level difference et PSNR sont utilisés. Cette recherche met en évidence les risques de sécurité posés par les fuites de EM dans SoC-FPGAs. Les résultats soulignent la nécessité de développer des contre-mesures solides pour atténuer les risques de SCAs et protéger les informations sensibles contre les accès non autorisés.

Title: Exploitation of EM Emanations of a SoC-FPGA Bus to Eavesdrop the Internally Processed Images of a Deep Learning Accelerator

Keywords: Electromagnetic Compatibility, Side-Channel Attacks, System-on-chip, Neural Network Accelerator, On-chip Data Buses

Abstract: This thesis investigates the exploitation of EM leaks from data buses to recover secret input images processed by NN accelerators implemented on SoC-FPGAs. During NN inference, input image pixels transmitted across internal buses emit EM emanations which are then analyzed to reconstruct the original input images to the NN. By leveraging physical access to the system, an adversary can deploy a near-field probe, such as a HW Trojan, to capture these EM traces emitted by SoC bus during image classification. Two novel methodologies are proposed to interpret EM emanations into input pixels, while exposing critical hardware-level vulnerabilities, including the routing and positioning of internal bus wires within the chip. To validate the proposed approaches, two metrics: HD-level difference and PSNR are used. A third-party image classifier is also used to validate the accuracy of attack. This research highlights the security risks posed by EM leaks in SoC-FPGAs. The findings underscore the need for developing robust countermeasures to mitigate the risks of SCAs and protect sensitive information from unauthorized access.