N° d’ordre NNT : 2025STET0011

THÈSE de DOCTORAT
DE L’UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Membre de l’Université de Lyon

Ecole Doctorale N°488
SIS - Sciences Ingénierie Santé

Spécialité de doctorat : Microélectronique

Soutenue publiquement le 3 février 2025, par :

Paul GRANDAMME

Techniques avancées d'attaques par injection de fautes sur circuits intégrés

---

Devant le jury composé de :

Sylvain GIRARD, Professeur des universités, Université Jean Monnet Saint-Etienne IUF	Président
Giorgio DI NATALE, Directeur de recherche, CNRS	Rapporteur
Jérémy POSTEL-PELLERIN, Maître de conférences HDR, Aix-Marseille Université	Rapporteur
Vincent POUGET, Chargé de recherche, CNRS	Examinateur
Stéphanie ANCEAU, Ingénieure de recherche, CEA	Examinatrice
Lilian BOSSUET, Professeur des universités, Université Jean Monnet Saint-Etienne	Directeur de thèse
Jean-Max DUTERTRE, Professeur des universités, Ecole des Mines de Saint-Étienne	Directeur de thèse
Michel AGOYAN, Ingénieur, STMicroelectronics	Invité

Remerciements

Ce manuscrit est la finalité de trois années de thèse et plus largement d'un parcours scolaire que je n'aurais pas pu mener à bien sans l'accompagnement de nombreuses personnes. À vous, ces premières pages sont dédiées.

Tout d'abord, je tiens à remercier mes directeurs de thèse, Lilian et Jean-Max pour la confiance qu'ils m'ont accordée afin de mener à bien ce projet. Jean-Max est le professeur qui a su me faire apprécier l'électronique dès mes premiers pas en école d'ingé jusqu'à ce qu'il devienne un choix de carrière. Lilian m'a accueilli dans son équipe au laboratoire pendant le COVID et m'a fait découvrir le monde de la recherche académique. Ils ont tous les deux joué un rôle déterminant dans mon choix d'orientation professionnelle. Je ne les remercierai jamais assez pour leur accompagnement ainsi que la confiance, la sympathie et la pédagogie qu'ils ont témoignées envers moi durant ces quelques années.

Plus généralement, je tiens à remercier toute l'équipe SESAM qui m'a accueillie dans un premier temps en CDD puis en thèse : Florent avec qui j'ai apprécié les nombreuses discussions cinématographiques et qui m'a fait découvrir la loi de poisson-binomiale, Pierre-Louis qui m'a donné le goût du trail et avec qui j'ai pris plaisir à donner mes enseignements, Viktor et ses anecdotes partagées la plupart du temps autour d'une bière, Vincent et son humour souvent *piquant*, Brice pour toute l'aide et les conseils qu'il m'a apportés mais aussi pour les séances de montée d'escaliers du Crêt de Roc un peu douloureuses, Nathalie qui répond toujours présente pour résoudre les problèmes, Alain qui m'a beaucoup appris sur la conception d'ASIC lors de mon CDD, Cédric avec qui j'ai pris plaisir à débriefer les matchs de l'ASSE, Damien qui m'a communiqué son expérience de doctorant dès mon arrivée dans l'équipe, Pierre-Antoine et Éloïse les rouchons : j'ai eu la chance de vous rencontrer et vous compte désormais parmis mes proches amis, Arturo et sa bonne humeur permanente, Jorge qui m'a permis d'atteindre le Top 8 (pour l'instant) du tour de Montaud, Mathieu *a.k.a* le magicien flutiste, Matéus, Raphaële, Nicolas, Anis, Simon, Alexandre, William, Viet-Sang, Justine. Je n'oublie pas les stagiaires que j'ai eu la chance d'encadrer lors de ma thèse : Émilie et Bilel.

Je remercie aussi l'ensemble des membres du département SAS de l'École des Mines de Saint-Étienne à Gardanne pour leur accueil et leurs nombreux conseils sur les parties

expérimentales de cette thèse. Je pense notamment à Raphael V, Simon, Pierre-Alain, Jean-Baptiste, Olivier, Anne-Lise ainsi qu'aux aux doctorants, Élise, Kévin, Raphaël J, Roukoz, Rodrigo et le meilleur pour la fin, Théophile.

Cette thèse a été réalisée dans le cadre du projet POP dont je tiens à remercier l'ensemble des membres : Giorgio Di Natale, Ioana Vatajelu, Paolo Maistri, David Hely, Vincent Beroulle et Aghiles Douadi. J'ai ainsi pleinement pu profiter de leurs diverses expériences lors des nombreuses réunions de projet.

Je tiens également à avoir un mot pour l'ensemble des personnels techniques et administratifs du laboratoire Hubert Curien et du campus Georges Charpak sans qui le bon déroulement de cette thèse n'aurait pas été possible.

Un grand merci à Sylvain Girard, Adriana Morana, Arnaud, Martin, Hugo de l'équipe MOPERE du laboratoire Hubert Curien, Stéphanie Anceau, Sophie Bouat, Laurent Maingault du CEA et Luc Salvo du SIMAP pour leur collaboration scientifique dans ces travaux.

Je remercie aussi l'ensemble de la communauté de la sécurité matérielle que j'ai pu rencontrer lors de nombreux événements. Les remarques, questionnements et discussions issus de ces rencontres ont permis de faire évoluer ma recherche.

Merci également à l'ensemble de mes amis. Je vais procéder par ordre chronologique. Tout d'abord, Amélie, qui m'accompagne depuis le début du lycée. D'une rencontre un peu hasardeuse à la Base Nature de Fréjus en seconde, s'est construit une amitié qui m'est très chère aujourd'hui. Gaby et Matthieu, qui ont toujours été présents malgré la distance et nos parcours différents. Maxime, qui m'a énormément aidé, scolairement et moralement, pendant la prépa, et avec qui je prends un grand plaisir à courir la nuit entre Sainté et Lyon ou à parcourir les terrasses de bars à Lille en plein hiver. Je pense également à Louise *a.k.a* mon soleil musical, Hugo qui tient à nous recruter à Cambridge, Zicca que je n'ai jamais manqué de voir à Marseille lors de mes séjours à Gardanne, David et son jeu du lieu, Paul, Loïc et sa bonne humeur légendaire et Alex qui avait des cheveux à l'époque. Je remercie également mes amis stéphanois, Jérémy, Kévin, Clément, Juju, Alex, Perrine, Amandine et Léonie. C'est avec un grand plaisir que j'ai passé ces quelques années à vos côtés. J'ai aussi une pensée pour Mariane, sans qui je ne me serai sûrement pas lancé dans l'aventure d'une thèse.

Je tiens, enfin, à remercier l'ensemble de ma famille, et tout particulièrement, Pierre et Valérie. Vous m'avez montré un soutien sans faille pendant toutes ces années, depuis la première réunion parents-profs (transformée en réunion *tata-prof*) de classe prépa à ma soutenance de thèse. Je n'oublierai jamais les nombreux week-ends où vous m'avez accueilli pendant la prépa, les carnavals de Dunkerque, les sorties à Calais-Nord ou

les vacances d'été passées avec vous. Pour tout ces moments riches, je vous remercie. Je n'oublie pas Thomas, Dominique et Margot, qui m'ont réservé un accueil privilégié pendant la prépa. Vous avez, vous aussi, largement contribué à ma réussite lors de ces années.

Un dernier remerciement à mes parents, Yves et Sylvie, et à mes frères, Gauthier et Jérôme. Vous m'avez encouragé tout au long de ce parcours scolaire et donné les moyens de le mener à bien sereinement, votre soutien sans faille m'a été précieux.

Résumé

La sécurité physique des circuits intégrés est souvent évaluée en menant des attaques qui exploitent leurs vulnérabilités matérielles. Les attaques par injection de fautes sont une technique couramment utilisée dans cet objectif d'évaluation. Elles permettent à un attaquant d'altérer le fonctionnement nominal du composant afin d'obtenir des informations confidentielles. Les techniques principales d'injection de fautes localisées sont les injections laser et électromagnétique. Plus récemment, des travaux pionniers ont montrés que les rayons X pouvaient également modifier le comportement d'un circuit. L'objectif de cette thèse est d'évaluer le potentiel des attaques en fautes. Cela se fait en améliorant l'état de l'art existant, notamment sur les attaques par injection de fautes laser et rayons X, sur des mémoires Flash de microcontrôleurs dédiés à des applications IoT. La finalité de cette étude est de contribuer à la prise en compte de la menace que constituent ces attaques mais également de comprendre les phénomènes associés. Ces points constituent les premiers pas en vue de la conception de contre-mesures adaptées. Premièrement, après une description des limitations des bancs laser monospot, nous caractérisons les avantages significatifs, notamment d'un point de vue spatial et temporel, apportés par un nouveau banc laser multispot. Des exemples concrets de scénarios désormais atteignables sont décrits et une exploration théorique des nouvelles possibilités offertes par le banc laser est également réalisée. Deuxièmement, nous mettons en lumière la possibilité d'utiliser l'effet thermique d'un banc laser infrarouge afin d'injecter des fautes persistantes au sein de mémoires Flash de composants non alimentés. Ce nouveau vecteur d'attaque aboutit à la description d'un nouveau modèle de faute complet allant du niveau physique au niveau applicatif. Les résultats obtenus nous permettent d'une part, de réaliser l'ingénierie inverse de la mémoire Flash du composant ciblé et d'autre part, de retrouver la clé de chiffrement d'une implémentation logicielle de l'algorithme de chiffrement AES. Pour finir, l'utilisation de sources non focalisées de rayons X est décrite dans le but d'injecter des fautes dans des mémoires Flash de composants alimentés et non alimentés. Les phénomènes de récupération thermique et temporelle sont également caractérisés. La conception et la caractérisation de masques permettant, dans une certaine mesure, de focaliser l'injection de fautes est mise en pratique.

Abstract

The security of integrated circuits is evaluated through the implementation of attacks that exploit their inherent hardware vulnerabilities. Fault injection attacks represent a technique that is commonly employed for this purpose. These techniques permit an attacker to alter the nominal operation of the component in order to obtain confidential information. The principal techniques for localised fault injection are laser and electromagnetic injection. Recently, pioneering work has demonstrated that X-rays can also modify the behaviour of a circuit. The objective of this thesis is to assess the potential of fault attacks. This is achieved by advancing the existing state of the art, with a particular focus on laser and X-ray fault injection attacks on Flash memories of microcontrollers dedicated to the IoT. The aim of this study is twofold : firstly, to highlight the threat posed by these attacks and secondly, to gain insight into the associated mechanisms. These steps are crucial for the development of effective countermeasures. Firstly, after outlining the limitations of single-spot laser benches, we present a detailed analysis of the significant advantages offered by a new multispot laser bench, particularly in terms of spatial and temporal capacity. The study goes on to describe a number of concrete examples of scenarios that are now achievable, and also carries out a theoretical exploration of the new possibilities offered by the laser bench. Secondly, we propose the utilisation of the thermal effect of an infrared laser bench for the injection of persistent faults into the Flash memory of unpowered components. This novel attack vector gives rise to the delineation of a comprehensive new fault model, encompassing both the physical and application levels. The outcomes obtained facilitate the reverse engineering of the Flash memory of the targeted component and the extraction of the encryption key for a software implementation of the AES encryption algorithm. The final section of the thesis describes the use of unfocused X-ray sources for the injection of faults into the Flash memories of both powered and unpowered components. Furthermore, the thermal and temporal recovery phenomena are also characterised. The design and characterisation of masks that enable the focused injection of faults are demonstrated.

Table des matières

Liste des figures

Liste des tableaux

Liste des sigles

AES Advanced Encryption Standard. vii, 3–5, 8, 16, 17, 23, 40, 92, 93, 95, 98, 100, 134

AMFoRS Architectures and Methods for Resilient Systems. 6

ANR Agence Nationale de la Recherche. 5

BBRAM Battery Backed RAM. 34

BGA Ball Grid Array. 44

BTI Bias Temperature Instability. 54, 56

CARDIS Smart Card Research and Advanced Application Conference. 60

CC Critères Communs. 65

CEA Commissariat à l'Énergie Atomique et aux énergies alternatives. 5

CFI Control Flow Integrity. 4, 27

CME Coronal Mass Ejection. 41, 43

CMOS Complementary Metal-Oxide-Semiconductors. xiii, xiv, 8, 32, 35, 36, 53, 54, 67

CNRS Centre National de la Recherche Scientifique. 5

CTSYS Sûreté et Sécurité des Systèmes embarqués et distribués. 6

DDD Displacement Damage Dose. 48

DES Data Encryption Standard. 16, 23

DFA Differential Fault Analysis. 23, 121

DMD Digital Micromirror Device. 61

DOE Diffractive Optical Element. 61

DPA Differential Power Analysis. 2

DRAM Dynamic Random Access Memory. 53

ECC Error Correcting Code. 27

EDAC Error Detection and Correction. 4

EEPROM Electrically Erasable Programmable Read-Only Memory. xvi, 4, 10, 11, 15, 51, 112

EMI Injection électromagnétique. 27

EMP Impulsion électromagnétique. 27

EPROM Erasable Programmable Read-Only Memory. 10, 11

FIB Focused Ion Beam. 3, 4, 116

FPGA Field Programmable Gate Arrays. 23, 34, 134, 135

FSA Fault Sensitivity Analysis. 36, 37

FWHM Full-Width at Half Maximum. 61, 82, 83, 87

GCR Galactic Cosmic Rays. 41, 42

HCI Hot Carrier Injection. 56

IoT Internet of Things. vii, viii, 1, 65, 85, 93, 99, 133, 134

ISA Instruction Set Architecture. 22

LabHC Laboratoire Hubert Curien. 5, 129

LCIS Laboratoire de Conception et d'Intégration des Systèmes. 6

LET Linear Energy Transfer. 46

MCU MicroController Unit. 37

MLE Maximum Likelihood Estimation. 24

MOPERE Materials for Optics and Photonics in Extreme Radiation Environments. 104

MOS Metal Oxide Semiconductor. 8, 11, 54, 111, 114, 130, 134

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor. 8

MSE Mines de Saint-Étienne. 5, 6

NBTI Negative-BTI. xiv, 54, 55

NIST National Institute of Standards and Technology. 16

NMOS Negative MOS. xiii, xiv, 8, 9, 34–36, 49, 50

PAINE Physical Assurance and Inspection of Electronics. 102

PBTI Positive-BTI. 54

PCB Printed Circuit Board. 44

PFA Persistent Fault Analysis. 5, 23–25, 85, 90, 92, 94–96, 98–100, 133, 135

PMOS Positive MOS. xiii, 8, 9, 35, 36, 49

POP Power-Off laser attacks on security Primitives. 5

PUF Physically Unclonable Functions. 135

RSA Rivest-Shamir-Adleman. 22

SAA South Atlantic Anomaly. 43
SAS Systèmes et Architectures Sécurisés. 5
SEE Single Event Effect. 20, 48, 52
SEFI Single Event Functional Interrupt. 20, 53
SEL Single Event Latch-up. 53
SESAM Systèmes Embarqués Sécurisés et Architectures Matérielles. 5
SET Single Event Transient. 20, 36, 52, 53
SEU Single Event Upset. 20, 36, 53
SFA Statistical Fault Analysis. 23
SIFA Statistical Ineffective Fault Attack. 23
SIMaP Science et Ingénierie des Matériaux et des Procédés. 115, 121
SLM Spatial Light Modulator. 61
SoC System On Chip. 134
SOI Silicon On Insulator. 53
SPA Simple Power Analysis. 2
SRAM Static Random Access Memory. xiii, 9, 10, 36, 37, 49

TCHES Transactions on Cryptographic Hardware and Embedded Systems. 80
TDDB Time-Dependant-Dielectric Breakdown. xiv, 56, 57
TID Total Ionizing Dose. xiv, 48–51
TIMA Techniques of Informatics and Microelectronics for integrated systems Architecture. 6
TRNG True Random Number Generator. 135

UGA Université Grenoble-Alpes. 6, 121
UJM Université Jean-Monnet de Saint-Étienne. 5

ZCE Zone de charge d'espace. 32

Chapitre 1

Introduction générale

1.1 Positionnement du problème

Les progrès réalisés dans le domaine de la microélectronique ces dernières décennies ont permis la démocratisation des circuits intégrés. Ces progrès permettent la fabrication à bas coût, une forte augmentation de la puissance de calcul et une nette diminution de la consommation énergétique des composants. De nombreuses applications, notamment dans le cadre de l'*Internet of Things* (IoT), sont devenues possibles grâce à l'existence de ces nouveaux circuits intégrés communicants. Aujourd'hui, les objets connectés à base de ces circuits intégrés sont omniprésents dans de nombreux secteurs tels que celui de la domotique, des télécommunications, ou encore les secteurs bancaire, médical ou militaire.

Avec la démocratisation de ces objets connectés s'est posée la question de leur sécurité. En effet, les objets connectés peuvent contenir des informations sensibles concernant des biens, des infrastructures ou encore des personnes. La sécurité des données stockées et/ou manipulées est assurée mathématiquement par l'implémentation de primitives de sécurité comme les algorithmes cryptographiques. Ceux-ci doivent garantir notamment trois propriétés : l'*intégrité*, la *confidentialité* et l'*authenticité*. L'intégrité désigne la non-altération malicieuse d'une donnée après sa création, la confidentialité assure qu'uniquement les parties dans le secret ont accès à l'information et l'authenticité que l'origine de la donnée est certifiée et identifiée.

En revanche, l'implémentation matérielle de ces primitives peut contenir des failles exploitables par un attaquant. Ces failles ne sont pas nécessairement dues à une implémentation incorrecte des algorithmes, mais à des propriétés intrinsèques liées à la fabrication ou à la technologie des composants. Lorsqu'un circuit est en fonctionnement, des informations directement liées à son activité peuvent être extraites par le biais de diverses grandeurs physiques mesurables comme ses émanations électromagnétiques ou sa consommation

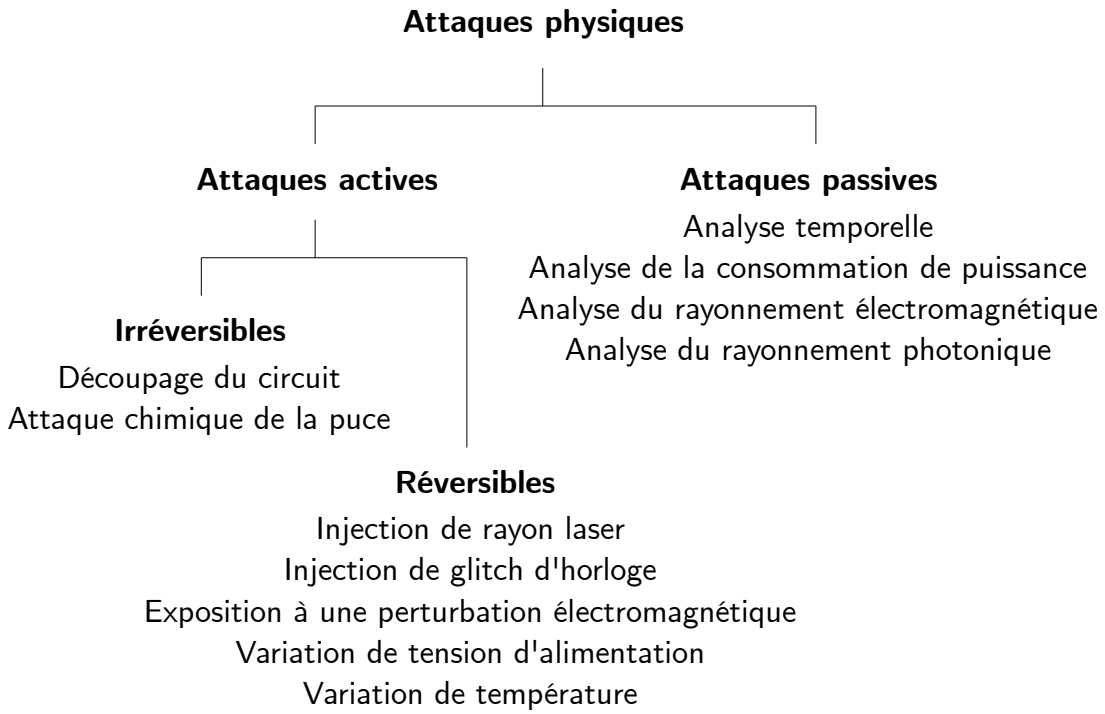


Figure 1.1 – Classification des attaques physiques [Bos18].

de puissance. Symétriquement, le fonctionnement d'un circuit peut être perturbé par des facteurs extérieurs comme une tension d'alimentation du circuit ou une température en dehors de leurs plages nominales ou des particules qui peuvent être radiatives.

Les phénomènes décrits précédemment peuvent être exploités par un adversaire pour attaquer physiquement un système cryptographique. On peut distinguer différents types d'attaques physiques. D'une part, les attaques passives sont basées sur la mesure de grandeurs physiques émanant du circuit pour retrouver de l'information sensible sur le système. D'autre part, les attaques actives visent à obtenir un fonctionnement détourné du circuit qui permet de retrouver de l'information secrète. Dans les deux cas, un attaquant cherche à induire ou à exploiter une fuite d'information. La Figure 1.1 classifie ces différents types d'attaques.

Les attaques passives sont souvent plus simples à mettre en place. Parmi ces dernières, on peut citer l'analyse temporelle (étude du temps d'exécution d'un programme ciblé), de la consommation de puissance, du rayonnement électromagnétique ou photonique. La corrélation entre les données secrètes manipulées par le circuit et la présence d'information contenue dans les canaux auxiliaires a été démontré par `Kocher` en 1996 [Koc96]. Il existe plusieurs schémas permettant d'extraire une clé de chiffrement d'un algorithme cryptographique tels que l'analyse simple de consommation (SPA pour *Simple Power Analysis*) ou l'analyse différentielle de la consommation (ou DPA pour *Differential Power*

Analysis) [Koc96 ; KJJ99]. Une très faible quantité d'informations qui fuite du système peut être exploitée pour attaquer un algorithme de cryptographie mathématiquement très robuste.

Les attaques actives quant à elles peuvent encore être scindées en deux sous-catégories : les attaques irréversibles et les attaques réversibles. Dans le cas des attaques irréversibles le système attaqué est altéré de façon permanente, voire détruit. On peut citer l'abrasion chimique, la découpe laser ou l'édition avec un faisceau d'ions focalisés (FIB pour *Focused Ion Beam*) du circuit. Il est également possible de sonder les signaux internes d'un circuit altéré au FIB pour extraire des informations [Gou+23]. Ces attaques sont principalement utilisées dans le but d'effectuer de l'ingénierie inverse, c'est-à-dire d'obtenir des informations sur la conception et l'architecture du circuit ou pour permettre d'autres attaques. Par exemple, une attaque par sondage grâce à un FIB est alors une attaque par observation (passive). Les attaques réversibles nécessitent une préparation de l'échantillon, comme l'ouverture du boîtier par exemple, qui n'est pas nécessairement destructive. On y trouve classiquement les injections laser [Col+19 ; Men+20b ; Lim+22] ou électromagnétique [Men+19 ; Men+20a], les *glitchs* d'horloge [BGV11] ou de tension d'alimentation [Zus+14] ou encore les variations de température [Sko09]. Elles sont souvent regroupées sous l'appellation d'attaques en fautes. Les attaques actives (réversibles ou non) nécessitent un temps, des moyens, des connaissances techniques et des informations sur le système, cela rend ces attaques plus complexes à mettre en œuvre.

Les attaques par injections de fautes peuvent aussi être classées en deux catégories : les attaques globales et locales. Les attaques globales perturbent le fonctionnement de l'entièreté du composant. C'est le cas des *glitchs* de tension d'alimentation [Zus+14 ; O'F16 ; BFP19] et d'horloge [Ago+10c ; BGV11]. Les attaques locales visent à modifier le comportement du système en altérant une des parties spécifiques du composant. C'est notamment le cas pour l'injection de fautes par exposition à une impulsion laser ou à une perturbation électromagnétique.

La majorité des études publiées sur les attaques par injection de fautes laser ont été menées à l'aide d'un banc laser monospot, c'est-à-dire ne comportant qu'une seule source laser [Ago+10a ; Ago+10b ; Dut+18 ; Col+19 ; Dut+19 ; Men+20b]. Ces attaques ont permis de mettre en défaut des algorithmes de chiffrement comme AES. Les bancs laser utilisés dans ces travaux possèdent les mêmes limitations. En effet, les temps de déplacement mécanique du spot laser empêchent un attaquant de cibler des positions différentes à des instants proches ou de fauter deux positions différentes en même temps. Cette limitation a été partiellement levée avec l'apparition des bancs laser bispot [Dum+21 ; VDL22 ; Vie+24]. Cette évolution des bancs laser a permis d'envisager des scénarios d'attaques plus évolués.

À quelques exceptions près, la majeure partie des travaux existants dans le domaine de l'injection de fautes ont été réalisés sur des circuits en fonctionnement et donc alimentés en énergie. Dans ce cas, des capteurs sont capables de détecter l'injection de fautes et permettent aux circuits de réagir en conséquence. Par exemple, en 2006, `Neto` *et al.* ont proposé un capteur permettant de détecter des courants transitoires anormaux dans le substrat pendant un tir laser [Net+06]. D'autre part, dans [Ber+14], une solution utilisant des capteurs numériques et analogiques, tirant ainsi parti des avantages de chaque méthode, est proposée pour détecter les *glitchs* de tensions. Pour finir, en 2016 `El-Baze` *et al.* [ERM16] ont décrit une solution entièrement numérique qui permet la détection d'injection d'ondes électromagnétiques basée sur l'étude des violations des temps de propagation. Il existe aussi d'autres méthodes qui ne sont pas basées sur l'utilisation de capteurs. On peut citer les mécanismes de *Control Flow Integrity* (CFI) qui détectent des comportements anormaux dans l'exécution d'un programme, ou les principes de redondances et les codes détecteurs et correcteurs d'erreurs (*Error Detection and Correction* (EDAC)) qui peuvent également détecter de mauvaises exécutions du code ou des erreurs dans ce dernier. Bien qu'efficaces, ces contre-mesures sont dites *actives* car elles ne fonctionnent que sur des composants alimentés en énergie. En revanche, lors de l'exécution d'un code source corrompu, c'est-à-dire lorsque le résultat d'une injection de faute au sein de la mémoire Flash d'un circuit non alimenté sera exploité, ces contremesures conservent leur efficacité.

En 2009, `Schmidt` *et al.* [SHP09] ont réussi à retrouver une clé AES en modifiant la boîte de substitution (S-Box) stockée en mémoire EEPROM d'un composant non alimenté par une exposition prolongé à une lumière UV. La méthode proposée repose sur une analyse différentielle qui nécessite d'avoir des chiffrés corrects et des chiffrés fautés. La cryptanalyse développée dans ces travaux n'exploite pas la persistance des fautes. D'autres travaux basés sur la modification physique d'un circuit non alimenté à l'aide d'un FIB (*Focused Ion Beam*) ont permis la désactivation de fonctions de sécurité et de contremesures. Cette méthode possède l'inconvénient d'être irréversible, très coûteuse et complexe à mettre en place.

Force est de constater que malgré les nombreuses études sur les attaques par injections de fautes, il en existe très peu qui abordent les injections multiples de fautes [Vie+23] et les attaques de circuits éteints [Sko09].

1.2 Plan et contributions

L'objectif de ces travaux est donc d'évaluer les possibilités d'attaques par injection de fautes plus évoluées qui aillent au-delà de l'état de l'art. L'utilisation d'un banc laser

multispot sur des circuits alimentés, ainsi que des attaques par injection de fautes laser ou rayons X sur circuits non alimentés peuvent prendre en défaut les contremesures existantes. Ces études seront détaillées dans ce manuscrit.

Le Chapitre 2 décrit quelques notions préliminaires nécessaires à la compréhension des travaux décrits dans ce manuscrit. La structure et le fonctionnement de la technologie MOS, des mémoires volatiles et non volatiles sont abordés dans ce chapitre. L'algorithme de chiffrement symétrique AES est également décrit dans ce chapitre. Pour finir, une revue des différents types d'attaques en fautes et de leur exploitation est réalisée.

Le Chapitre 3 synthétise l'état de l'art des effets du laser et des radiations sur les circuits électroniques. Les effets du vieillissement sont aussi décrits dans ce chapitre.

Le Chapitre 4 décrit l'utilisation d'un banc laser multispot. Les avantages, spatiaux et temporels, qui permettent de s'affranchir des limites d'un banc laser monospot sont démontrés dans ce chapitre [Col+22]. Les nouvelles possibilités d'attaques offertes par ce nouveau banc laser sont également explorées.

Le Chapitre 5 démontre la possibilité d'injecter des fautes permanentes au sein d'une mémoire Flash non alimentée en utilisant une source laser classiquement utilisée en injection laser. Une caractérisation des fautes obtenues du niveau physique au niveau logique est réalisée. Les résultats obtenus sont appliqués dans le cadre de l'Analyse de Faute Persistante (PFA pour *Persistent Fault Analysis*) [Gra+24].

Le Chapitre 6 analyse l'utilisation d'une source diffuse de rayons X pour corrompre le contenu de mémoires Flash d'un circuit non alimenté [GBD23]. La réalisation et l'exploitation d'un masque permettant de focaliser les injections de fautes sont également abordées.

1.3 Contexte

Cette thèse est réalisée dans le cadre du projet Power-Off laser attacks on security Primitives (POP) ou *Attaques laser de primitives de sécurité non alimentées* financé par l'Agence Nationale de la Recherche (ANR) et est réalisé en collaboration avec quatre équipes de recherche :

— Systèmes Embarqués Sécurisés et Architectures Matérielles (SESAM) du Laboratoire Hubert Curien (LabHC) de l'Université Jean-Monnet de Saint-Étienne (UJM) et le Centre National de la Recherche Scientifique (CNRS)
— Systèmes et Architectures Sécurisés (SAS) de Mines de Saint-Étienne (MSE) et du Commissariat à l'Énergie Atomique et aux énergies alternatives (CEA)

— Sûreté et Sécurité des Systèmes embarqués et distribués (CTSYS) du Laboratoire de Conception et d'Intégration des Systèmes (LCIS) de l'Université Grenoble-Alpes (UGA)
— Architectures and Methods for Resilient Systems (AMFoRS) du laboratoire Techniques of Informatics and Microelectronics for integrated systems Architecture (TIMA) de l'Université Grenoble-Alpes (UGA)

Cette thèse s'est déroulée de janvier 2021 à décembre 2024 au sein du laboratoire Hubert Curien de l'Université Jean-Monnet de Saint-Étienne et du Centre Microélectronique de Provence de Mines de Saint-Étienne à Gardanne.

Chapitre 2

Notions préliminaires

Table des matières

2.1 Introduction

Ce chapitre décrit quelques notions préliminaires nécessaires à la compréhension des travaux décrit dans cette thèse. Le fonctionnement de la technologie MOS, des mémoires volatiles et non-volatiles, du chiffrement AES ainsi que les notions d'attaque en fautes, de schéma d'attaque et les mécanismes de protection associés seront abordés. Un lecteur familier de ces notions peut s'affranchir de la lecture de ce chapitre.

2.2 Technologie MOS

Le transistor *Metal Oxide Semiconductor* (MOS) ou MOSFET est le composant élémentaire des circuits électroniques en technologie Complementary Metal-Oxide-Semiconductors (CMOS). Il est composé d'un empilement Métal | Oxyde | Semi-conducteur et de trois contacts dits de source, de drain et de grille. Il existe deux types de transistors en technologie CMOS [Abb20].

Transistor NMOS Il est constitué de deux zones enrichies en électrons (dopage n+) dans un substrat appauvri en électrons (substrat p). Le substrat est généralement polarisé à la masse.

Transistor PMOS Il est constitué de deux zones appauvries en électrons (dopage p+) dans un puits enrichi en électron (dopage n). Le puits est habituellement polarisé à la tension d'alimentation. Ce puits est implanté dans le même substrat que les transistors NMOS.

Une vue en coupe de ces deux types de transistors est visible sur la Figure 2.1. On y retrouve le substrat p ① avec son point de polarisation ②, la source du NMOS ③, la grille du NMOS ④, l'oxyde de grille du NMOS ⑤, le drain du NMOS ⑥, le puits n ⑦ et son point de polarisation ⑧.

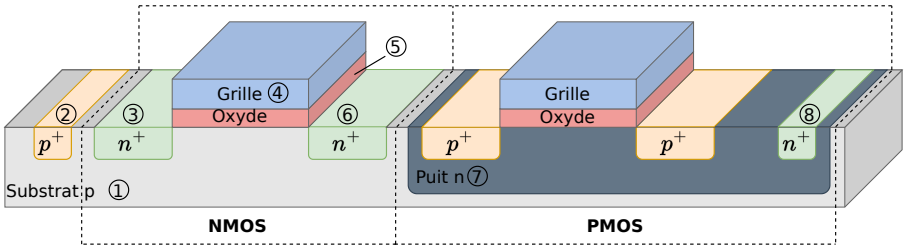
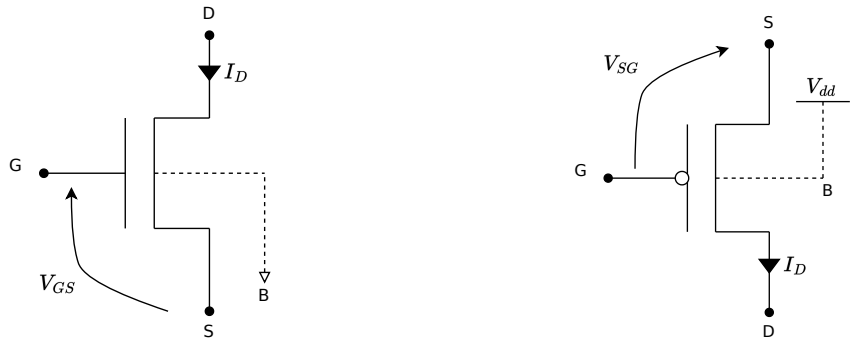


Figure 2.1 – Vue en coupe de transistors NMOS et PMOS en technologie CMOS.

Ces deux types de composants ont quatre terminaux : la grille G, la source S, le drain D et le substrat B (*bulk* en anglais). La figure Figure 2.2 montre la représentation électrique de ces composants.



(a) Transistor NMOS. **(b)** Transistor PMOS.

Figure 2.2 – Schéma électrique des transistors NMOS et PMOS.

Ces composants possèdent deux principaux modes de fonctionnement en saturation : bloqué ou passant.

Mode bloqué Le transistor est dit *bloqué* lorsque, en l'absence de polarisation, les différences de dopage entre les zones de diffusion et le substrat sont telles que la source et le drain sont isolés électriquement. Dans ce cas, aucun courant ne peut circuler entre ces terminaux.

Mode passant Le transistor est dit *passant* lorsqu'une tension supérieure à la tension de seuil V_{th} est appliquée sur la grille du transistor. Dans ce cas, un canal de conduction se forme entre la source et le drain à l'interface entre le substrat et l'oxyde de grille. Un courant peut alors circuler entre la source et le drain.

2.3 Mémoires volatiles

Les mémoires volatiles sont utilisées pour stocker les données, le contexte et les résultats de l'exécution d'un programme. Elles sont donc des cibles privilégiées pour un attaquant. Les données stockées sont conservées tant que le circuit est alimenté. Les mémoires à base de cellules SRAM (*Static Random Access Memory*) sont un type de mémoires volatiles.

La plus commune des cellules SRAM est constituée d'un circuit bistable, c'est-à-dire qui possède deux états d'équilibre qui représentent respectivement l'état `'1'` et l'état `'0'`. Il faut appliquer une stimulation électrique pour forcer la cellule à entrer dans l'un des

deux états. L'équilibre de la cellule est maintenu en l'absence de stimulation extérieure. Chaque cellule stocke un bit d'information.

La Figure 2.3 représente une cellule SRAM standard à six transistors. Elle est constituée de deux inverseurs (M_{n1}/M_{p1} et M_{n2}/M_{p2}) connectés tête-bêche et de deux transistors d'accès (T_1 et T_2). Ces derniers sont activés par le signal `word` et permettent alors d'accéder au bit d'information via les signaux `bit` et son complémentaire $\overline{\texttt{bit}}$.

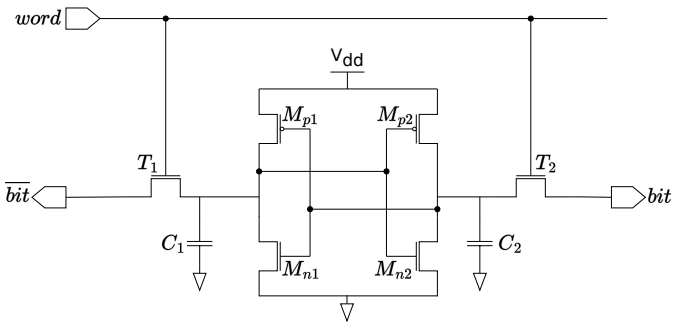


Figure 2.3 – Cellule SRAM standard à 6 transistors.

Si l'entrée de l'inverseur M_{n1}/M_{p1} est à l'état haut (`'1'`), la capacité de sortie C_1 est maintenue déchargée (`'0'`) par le transistor M_{n1} qui est passant. À l'inverse, si l'entrée de l'inverseur M_{n2}/M_{p2} est à l'état bas (`'0'`), la capacité de sortie C_2 est maintenue chargée (`'1'`) par le transistor M_{p2} qui est passant.

La lecture du point mémoire se fait en activant la ligne `word`, cela rend les transistors d'accès T_1 et T_2 passants. La donnée est donc disponible sur le signal `bit` et son complémentaire $\overline{\texttt{bit}}$. L'écriture d'un `'0'` (respectivement d'un `'1'`) dans la cellule se fait en activant les signaux `word` et `bit` et en désactivant le signal $\overline{\texttt{bit}}$ (respectivement en activant les signaux `word` et $\overline{\texttt{bit}}$ et en désactivant le signal `bit`).

2.4 Mémoires non volatiles

Une mémoire non-volatile est une mémoire qui conserve les données stockées en l'absence d'alimentation électrique. Il en existe trois principaux types : EPROM (*Erasable Programmable Read-Only Memory*), EEPROM (*Electrically Erasable Programmable Read-Only Memory*) et Flash. Historiquement, les premières mémoires non-volatiles à voir le jour sont les mémoires EPROM. Elles peuvent être lues et programmées, mais ne sont pas effaçables électriquement. L'effacement des mémoires EPROM est effectué en exposant la puce à une lumière UV au travers d'une fenêtre en quartz. Les mémoires EEPROM sont une évolution des mémoires EPROM. Elles sont effaçables électriquement sans qu'il soit nécessaire de retirer la mémoire de la puce de l'appareil. Les mémoires Flash sont un type de mémoire EEPROM qui se distingue par leur rapidité et leur effaçabilité par

secteur, et non par adresse individuelle. Les technologies EPROM et EEPROM sont aujourd'hui obsolètes.

Elles sont toutes construites autour d'un composant élémentaire : le transistor à grille flottante.

2.4.1 Transistor à grille flottante

Un transistor à grille flottante en technologie Flash possède la même structure qu'un transistor MOS à l'exception de la grille flottante qui est ajoutée dans l'oxyde entre le substrat et la grille de contrôle. Cette grille flottante est isolée électriquement du reste de la structure. Une représentation schématique d'un transistor à grille flottante est visible en Figure 2.4.

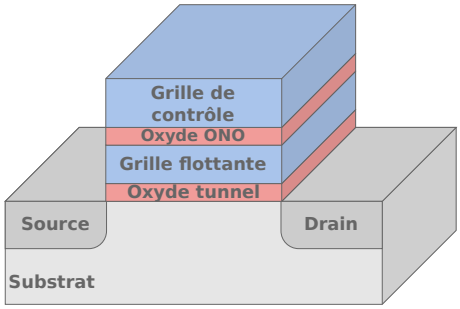


Figure 2.4 – Vue en coupe d'un transistor à grille flottante.

La grille flottante est utilisée pour stocker une charge électrique. La présence de cette charge électrique traduit l'état, effacé ou programmé, du transistor à grille flottante. Cet état permet de représenter un bit d'information. Selon la convention usuelle adoptée par une majorité des fabricants de semi-conducteurs, l'état effacé (absence d'électrons dans la grille flottante) correspond à un '1' logique et l'état programmé (présence d'électrons dans la grille flottante) correspond à un '0' logique.

Il existe trois types d'opérations au sein des mémoires Flash : *(i)* la programmation, *(ii)* l'effacement et *(iii)* la lecture. La Figure 2.5 synthétise les différentes tensions aux bornes d'un transistor à grille flottante pour les opérations d'écriture, d'effacement ou de lecture. Les valeurs sont données à titre d'exemple et peuvent varier d'un composant à l'autre en fonction de la technologie employée. La programmation et l'effacement consistent en l'ajout ou le retrait de charges électriques, des électrons, dans la grille flottante. Deux principaux mécanismes permettent ces opérations :

1. **L'injection d'électrons chauds** permet la programmation des transistors à grille flottante. Un potentiel est appliqué sur la grille de contrôle et sur le drain alors que

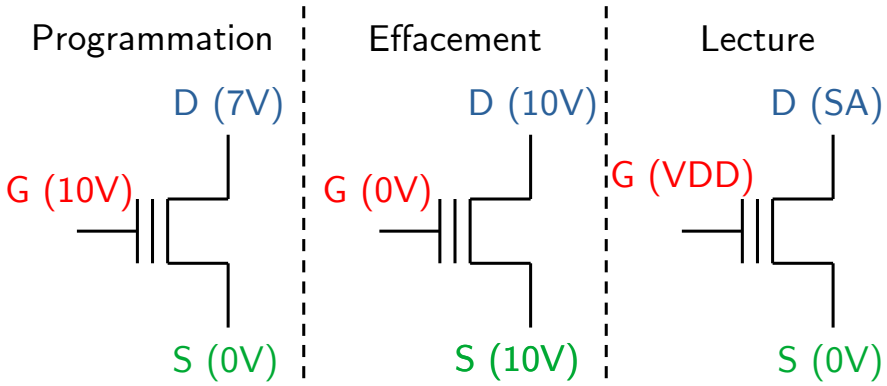
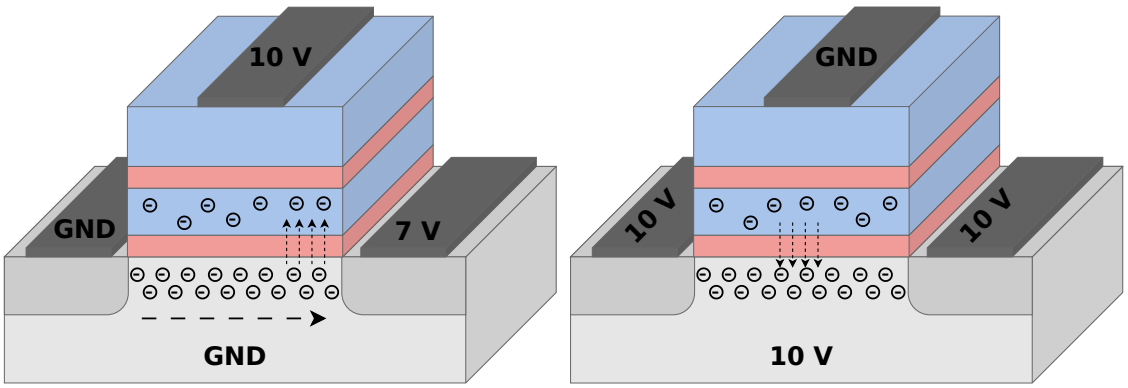


Figure 2.5 – Tensions appliqués aux bornes d'un transistor à grille flottante pendant les différentes opérations [Vie+21].

la source est maintenue à la masse. Un champ électrique important apparaît donc dans la zone de charges d'espace, un courant électrique circule entre la source et le drain, les électrons constituant ce courant ont assez d'énergie pour passer la barrière de potentiel représentée par l'oxyde tunnel et sont injectés dans la grille flottante. La Figure 2.6a illustre ce phénomène.

2. **L'effet tunnel Fowler-Nordheim** est utilisé pour l'effacement des transistors à grille flottante. Un champ électrique important est appliqué entre le substrat et la grille de contrôle alors que la source et le drain sont en haute impédance. Les électrons stockés dans la grille sont ainsi capables de traverser l'oxyde tunnel par effet tunnel pour rejoindre le substrat. La Figure 2.6b illustre ce phénomène.



(a) Mécanisme d'injection d'électrons chauds. **(b)** Effet tunnel Fowler-Nordheim.

Figure 2.6 – Programmation (a) et Effacement (b) d'un transistor à grille flottante.

La présence des charges électriques dans la grille flottante génère une augmentation de la tension de seuil V_{th} du transistor à grille flottante comme le montre la Figure 2.7. Pour une certaine tension V_{read} appliquée sur la grille de contrôle, le transistor à grille flottante va créer un appel de courant différent selon la charge stockée dans la grille

flottante, c'est-à-dire selon V_{th}. Ainsi, l'état logique du transistor à grille flottante est déterminé par la présence de charges dans celle-ci, ce qui affecte la tension de seuil.

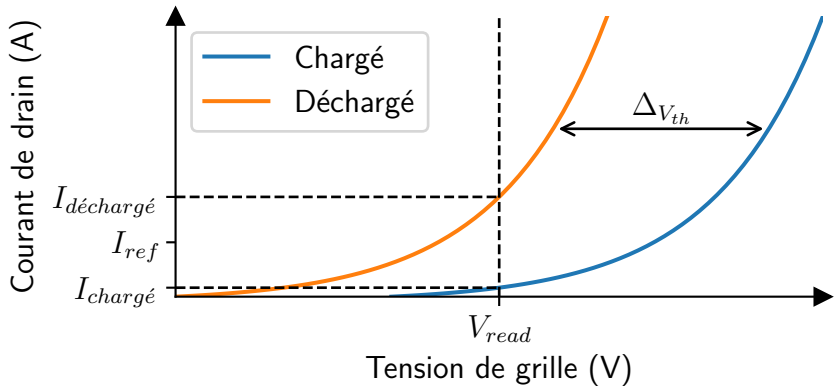


Figure 2.7 – Caractéristique I-V de transistors à grille flottante chargé et déchargé.

La lecture d'un transistor à grille flottante est effectuée en appliquant la tension de lecture V_{read} sur la grille de contrôle, le courant appelé par le transistor est ensuite comparé à un courant de référence I_{ref} par un comparateur de courants (ou *sense amplifier* en anglais). Ce dernier génère un `'1'` ou un `'0'` selon la différence de courant mesurée.

Par convention, un transistor chargé, c'est-à-dire avec des électrons dans sa grille flottante, est dit *programmé* et représente l'état logique `'0'` alors qu'un transistor ne contenant pas d'électrons dans sa grille flottante est dit *effacé* et représente l'état logique `'1'`. La convention inverse peut aussi être rencontrée.

La Figure 2.8 schématise la lecture de transistors à grille flottante programmés et effacés. Ce schéma est valable pour une mémoire Flash respectant l'architecture NOR. L'agencement des transistors est différents pour une mémoire NAND Flash.

Pour lire la valeur stockée dans le transistor T_0 qui est chargé, on applique V_{read} sur la *wordline* WL_p. Le transistor à grille flottante T_0 va ainsi appeler un courant $I_{chargé}$ qui est inférieur à I_{ref} ce qui produira un `'0'` en sortie du comparateur de courant. C'est le scénario bleu sur la Figure 2.8.

Pour lire la valeur stockée dans le transistor T_1 qui est déchargé, on applique V_{read} sur la *wordline* WL_{p+1}. Le transistor à grille flottante T_1 va ainsi appeler un courant $I_{déchargé}$ qui est supérieur à I_{ref} ce qui produira un `'1'` en sortie du comparateur de courant. C'est le scénario orange sur la Figure 2.8.

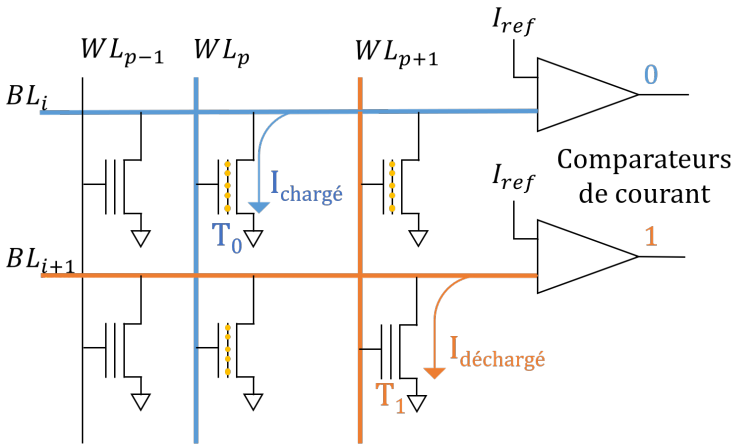


Figure 2.8 – Lecture de transistors à grille flottante (architecture NOR).

La Figure 2.9 montre une vue en coupe d'une mémoire NOR Flash lors de l'opération de lecture. On y retrouve : ① la grille de contrôle, ② l'isolant interpoly, ③ la grille flottante, ④ l'oxyde tunnel, ⑤ la source, ⑥ le drain, ⑦ le substrat, ⑧ les vias et ⑨ la *bitline*. Le transistor sélectionné par l'activation de sa *wordline* (présence du 5 V sur sa grille de contrôle) est encadré en noir sur la figure. On peut remarquer que la *bitline* est sélectionnée en étant polarisée à une tension de 1 V et que la *wordline* est sélectionnée en étant polarisée à une tension de 5 V. Ici encore, les valeurs numériques sont données à titre d'exemple et peuvent varier d'une technologie à une autre. Les *wordlines* des transistors non sélectionnés sont polarisées à la masse. Dans ces conditions, le courant I_{read} dépend de la charge stockée dans la grille flottante du transistor sélectionné.

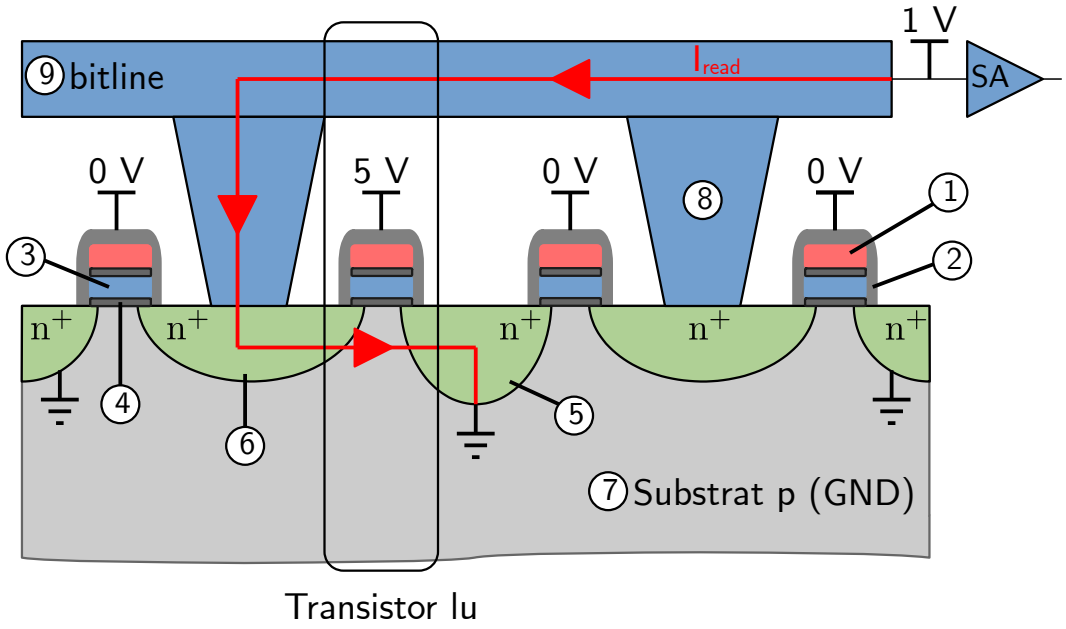


Figure 2.9 – Vue en coupe d'une mémoire NOR Flash pendant l'opération de lecture.

2.4.2 Mémoires Flash

Les mémoires non-volatiles, qu'elles soient de type Flash ou EEPROM, possèdent les mêmes structures. Elles sont composées d'une matrice de transistors à grille flottante, de décodeurs X et Y, de pompes de charges, d'une logique de contrôle, de bascules et de comparateurs de courants. La Figure 2.10 représente l'architecture de ces mémoires.

La logique de contrôle fait le lien entre les adresses logiques et les adresses physiques dans la mémoire. Les décodeurs X et Y sélectionnent les transistors à grille flottante selon les sorties de la logique de contrôle. Les comparateurs de courants sont utilisés pour comparer le courant appelé par un transistor à grille flottante lors de la lecture au courant de référence. Une mémoire contient autant de comparateurs de courant que la largeur en bits du bus de données. Les pompes de charges génèrent les hautes tensions nécessaires à la programmation et l'effacement des cellules mémoires. Les bascules servent à échantillonner les données stockées dans la mémoire lors de la lecture.

Ainsi, pour lire une donnée de 32 bits, il est nécessaire de sélectionner une *wordline* et 32 *bitlines*.

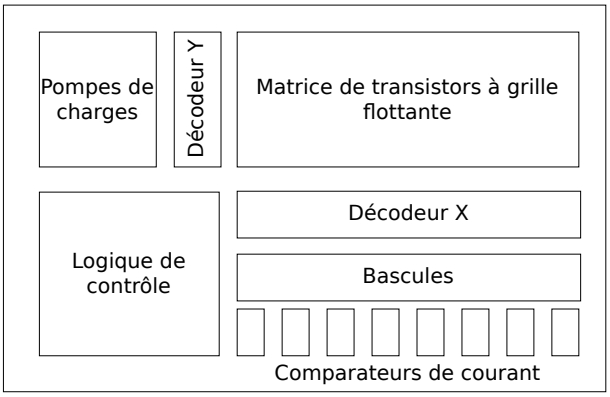
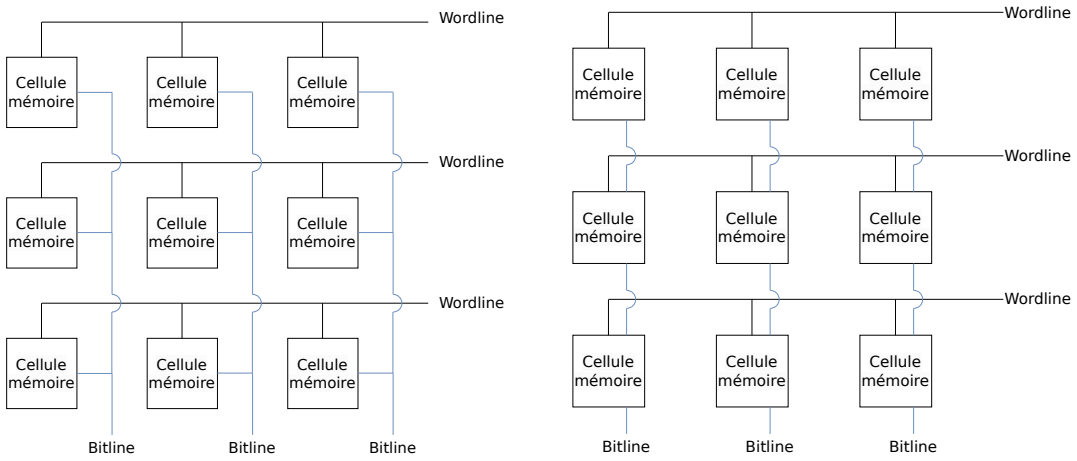


Figure 2.10 – Organisation usuelle des mémoires Flash.

Il existe deux configurations de mémoire Flash : NOR et NAND, selon la ressemblance de la ligne de transistors d'état avec une cellule de transistors NOR ou NAND. Les mémoires Flash NOR possède l'avantage d'avoir un accès rapide aux données, mais ont une plus faible densité. Elles sont essentiellement utilisées dans les cas où la lecture est l'opération la plus fréquente. L'architecture simplifiée d'une mémoire Flash NOR est visible en Figure 2.11a. Les mémoires Flash NAND ont une très forte densité, mais uniquement un accès séquentiel aux données, c'est-à-dire que les données sont lues dans un ordre prédéfini. Ce mode de lecture n'est donc pas adapté aux situations où un accès aléatoire aux données est requis, ce qui est typiquement le cas lors de l'exécution d'un programme. Elles sont utilisées pour le stockage de données. L'architecture simplifiée d'une mémoire Flash NAND est visible en Figure 2.11b. Dans le cas des systèmes embarqués, ce sont

les mémoires NOR qui sont principalement utilisées. Les mémoires Flash NAND sont utilisées par exemple pour les clés USB.



(a) NOR Flash. **(b)** NAND Flash.

Figure 2.11 – Architectures des mémoires Flash.

2.5 Chiffrement AES

L'AES (Advanced Encryption Standard) [DR02] est un algorithme de chiffrement symétrique par blocs. Il est issu d'un concours lancé en 1997 par le National Institute of Standards and Technology (NIST) afin de devenir le nouveau standard de chiffrement symétrique, remplaçant le DES. Il a été proposé par `Daemen` et `Rijmen` en 2000 et remporta le concours. Il devient la référence lors de sa publication par le NIST [ST01].

Les données sont traitées par blocs de 128 bits pour le texte clair et le chiffré. La clé secrète de chiffrement a une longueur de 128 bits dans le cas de l'AES128. Il existe aussi deux autres variantes pour lesquelles la clé a une taille de 192 ou 256 bits.

L'algorithme est appliqué sur un bloc organisé en matrice de 4×4 éléments, qui sont des octets, appelé *état*. Soit S la matrice représentant l'état et $S_{i \in [\![0;15]\!]}$ l'élément d'indice i de S.

$$S = \begin{pmatrix} S_0 & S_4 & S_8 & S_{12} \\ S_1 & S_5 & S_9 & S_{13} \\ S_2 & S_6 & S_{10} & S_{14} \\ S_3 & S_7 & S_{11} & S_{15} \end{pmatrix}$$

L'algorithme est composé de plusieurs rondes, elles-mêmes divisées en plusieurs transformations : *AddRoundKey*, *SubBytes*, *ShiftRows* et *MixColumns*. Selon la longueur de la clé, le nombre de rondes diffère. Le Tableau 2.1 représente les différentes versions de l'algorithme AES.

Taille de la clé en bits	N_r
128	10
192	12
256	14

Tableau 2.1 – Différentes versions de l'algorithme AES avec N_r le nombre de rondes.

Le déroulement global de l'algorithme de chiffrement AES est décrit en Figure 2.12.

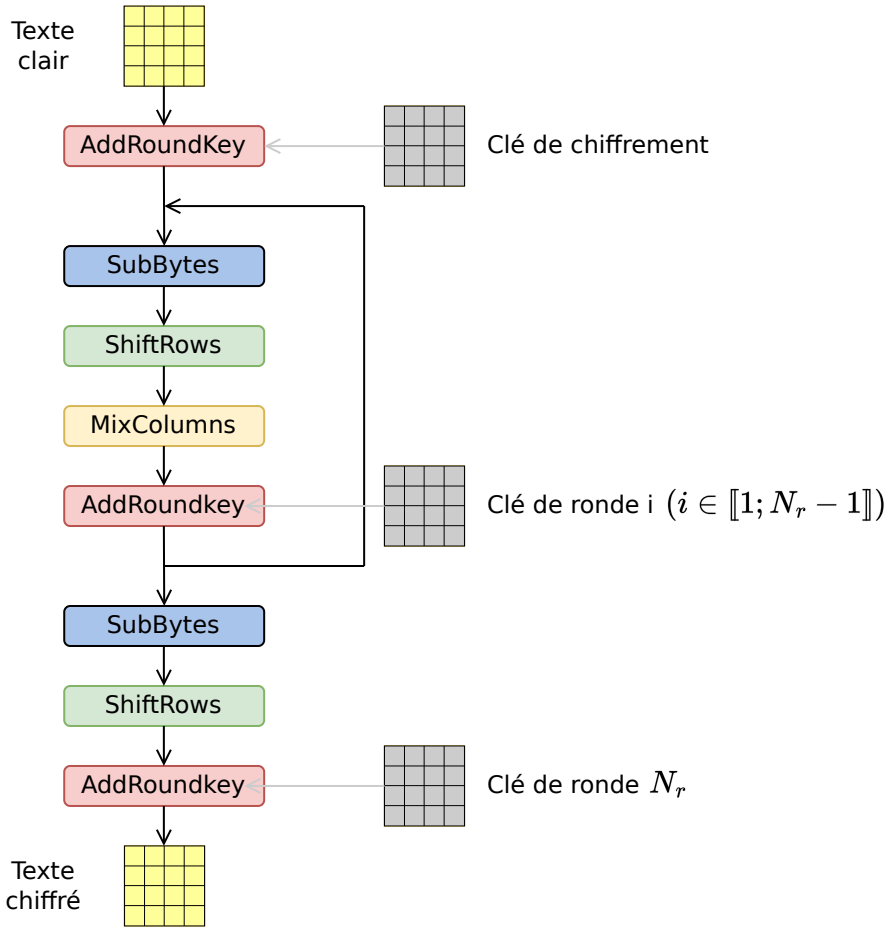


Figure 2.12 – Déroulement du chiffrement AES.

AddRoundKey Cette transformation ajoute la clé de ronde[1] à l'état. Un OU exclusif bit à bit est appliqué entre l'état et la clé. Elle est la seule opération faisant intervenir la clé. Les clés de rondes sont obtenues avec l'opération *KeySchedule*.

SubBytes Cette étape est une transformation non linéaire appliquée à chacun des octets de l'état en utilisant une boîte de substitution : la S-box. Elle est représentée en Tableau 2.2. Par exemple, $\text{S-Box}[0x73] = 0x8f$. Cette fonction de substitution remplit

1. ou la clé de chiffrement pour la ronde intiale

le rôle de couche de *confusion* dans le chiffrement. La *confusion* a pour objectif de rendre la relation entre la clé de chiffrement et le texte chiffré la plus complexe possible.

ShiftRows Cette étape est une permutation cyclique des octets de l'état. La valeur du décalage est fixée par l'indice de la ligne considérée. Cette étape amène de la *diffusion* dans l'information. Une représentation graphique de cette transformation est visible ci-dessous.

$$\begin{pmatrix} S_0 & S_4 & S_8 & S_{12} \\ S_1 & S_5 & S_9 & S_{13} \\ S_2 & S_6 & S_{10} & S_{14} \\ S_3 & S_7 & S_{11} & S_{15} \end{pmatrix} \Rightarrow \begin{pmatrix} S_0 & S_4 & S_8 & S_{12} \\ S_5 & S_9 & S_{13} & S_1 \\ S_{10} & S_{14} & S_2 & S_6 \\ S_{15} & S_3 & S_7 & S_{11} \end{pmatrix}$$

MixColumns Cette transformation linéaire est appliquée à l'état colonne par colonne. Les colonnes sont considérées comme des polynômes dans $GF(2^8)$ et sont multipliées modulo $X^4 + 1$. Cette étape apporte également de la *diffusion* dans l'information. Une représentation graphique de cette transformation est visible ci-dessous.

$$\begin{pmatrix} S_i \\ S_{i+4} \\ S_{i+8} \\ S_{i+12} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} S_i \\ S_{i+4} \\ S_{i+8} \\ S_{i+12} \end{pmatrix}$$

On peut remarquer que, dans $GF(2^8)$:

— La multiplication par 1 est neutre.
— La multiplication par 2 est un décalage vers la gauche des bits suivi d'une réduction par le polynôme caractéristique s'il y a dépassement.
— La multiplication par 3 est une multiplication par 2 suivie d'une addition avec la valeur initiale.

Ces propriétés sont particulièrement intéressantes pour les implémentations de l'algorithmes.

KeySchedule Chaque clé de ronde est calculée à partir de la clé de ronde précédente. La première colonne de la clé de ronde $i (i \in [\![1; 10]\!])$ est calculée en suivant les étapes suivantes :

1. Une permutation circulaire de la dernière colonne de la clé de ronde $i - 1$.

2. Une opération *SubBytes* sur chaque octet de la colonne.
3. Un XOR bit à bit avec la première colonne de la clé de ronde $i-1$ et avec la $i^{ème}$ colonne de la matrice RCON telle que définie dans Équation 2.1.

$$RCON = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x08 & 0x10 & 0x20 & 0x40 & 0x80 & 0x1b & 0x36 \end{pmatrix} \tag{2.1}$$

Les trois colonnes suivantes sont obtenues en effectuant un XOR bit à bit entre la colonne précédemment calculée et la colonne de même indice dans la clé de ronde précédente.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tableau 2.2 – Table de substitution d'AES.

2.6 Attaques par injection de fautes

La conception d'un circuit intégré se base sur un ensemble de spécifications matérielles. Ces dernières définissent le fonctionnement et les caractéristiques du composant. On appelle **faute** toute erreur causée par une perturbation physique ou une perturbation des conditions nominales de fonctionnement du composant volontaire et **attaque en faute** l'exploitation de ces perturbations pour modifier les fonctionnalités du composant à des fins d'attaque.

Par ailleurs, on appelle **faute naturelle** tout dysfonctionnement involontaire du composant. Les fautes naturelles peuvent avoir des origines diverses comme le bruit électromagnétique, la diaphonie (ou *crosstalk*) ou un environnement radiatif.

Les fautes peuvent être **transitoires**, c'est-à-dire temporaires ou **persistantes**, c'est-à-dire qu'elles persistent pendant un certain temps (jusqu'à une réécriture, un *reset* ou un cycle d'alimentation) ou **permanentes**, c'est-à-dire qu'elles ne peuvent être corrigées.

2.6.1 Classification des fautes matérielles

Les effets singuliers (Single Event Effect (SEE)) désignent la perturbation d'un circuit électronique par l'impact d'une particule. Ce sont des phénomènes instantanés, de l'ordre de la nanoseconde, et qui peuvent intervenir dans des circuits neufs.

Les effets singuliers peuvent être catégorisés selon une classification standardisée des fautes [Esa]. Cette classification est décrite ci-dessous.

Single Event Transient (SET) : Apparition d'un pic parasite (ou d'une chute) de courant ou de tension en sortie d'une porte logique. Cette altération du signal peut ne pas avoir d'impact sur le fonctionnement du circuit. Une faute intervient dans le cas contraire.

Single Event Upset (SEU) : Changement de l'état logique d'un élément mémoire (une bascule par exemple). Peut intervenir suite à un SET.

Single Event Functional Interrupt (SEFI) : Apparition d'une faute transitoire à l'origine d'un dysfonctionnement du système (*reset*, verrouillage ou défaillance). Un cycle d'alimentation ou une remise à zéro du système corrige ce dysfonctionnement.

2.6.2 Modèle de fautes

Un **modèle de fautes** définit un cadre d'abstraction permettant à un attaquant de construire une attaque.

Les trois principaux niveaux d'abstractions sont les suivants :

— **Niveau physique :** C'est le plus bas niveau d'abstraction, au plus près des transistors et des charges électriques. Ce niveau a pour rôle d'étudier les interactions physiques intervenant lors d'une perturbation extérieure et aboutissant à la création ou l'altération d'un courant ou d'une tension.

— **Niveau logique :** Ce niveau intermédiaire définit l'impact d'une faute au niveau binaire (en termes de `'0'` ou de `'1'`).
— **Niveau applicatif :** C'est le plus haut niveau d'abstraction. C'est à ce niveau que l'impact d'une faute sur un programme ou un algorithme cryptographique est analysé.

Pour une bonne compréhension des phénomènes intervenant lors d'une attaque, il est primordial de comprendre le modèle de fautes à tous les niveaux d'abstractions.

Au niveau logique, il existe trois modèles de fautes usuels, indépendamment de la technique d'injection utilisée :

— **bitset :** Passage d'un `'0'` logique à un `'1'` logique suite à l'injection d'une faute. Si la donnée est déjà à `'1'`, elle reste à `'1'` et il n'y a pas de véritable faute.
— **bitreset :** Passage d'un `'1'` logique à un `'0'` logique suite à l'injection d'une faute. Si la donnée est déjà à `'0'`, elle reste à `'0'` et il n'y a pas de véritable faute.
— **bitflip :** Passage d'un `'1'` (respectivement `'0'`) logique à un `'0'` (respectivement `'1'`) logique suite à l'injection d'une faute indépendamment de la valeur initiale.

Les modèles *bitset* et *bitreset* sont dépendants de la valeur initiale de la donnée.

2.6.3 Conséquences au niveau de la mémoire

Au niveau de la mémoire, une injection de faute peut avoir différentes conséquences. C'est sur ce niveau d'abstraction qu'un attaquant peut construire un scénario d'attaque permettant de mettre en défaut un algorithme cryptographique ou un mécanisme de protection. Les principales conséquences d'une faute au niveau de la mémoire sont décrits ci-dessous.

Corruption de données Une injection de faute peut altérer les données manipulées et aboutir à un comportement dégradé de l'algorithme. `Boneh`, `DeMillo` et `Lipton` [BDL97 ; BDL01] ont été les premiers à montrer que la corruption de données pouvaient aboutir à la construction de scénarios d'attaques permettant de retrouver de l'information secrète. Il a aussi été démontré que la corruption de données peut modifier l'exécution d'un programme [Dut+12] ou désactiver des mécanismes de protection [Vas+17].

Corruption d'instructions Les instructions d'un programme sont définies selon une norme appelée *Instruction Set Architecture (ISA)*. Cette norme dépend du type de cœur considéré (ARM, RISC-V, AVR, etc). Elles sont principalement composées d'un opcode traduisant l'opération à effectuer, d'opérandes qui sont les variables nécessaires à l'opération et éventuellement d'une constante numérique ou de conditions optionnelles. Ainsi, il a été démontré qu'il est possible de modifier le comportement d'un algorithme en modifiant l'opcode ou les opérandes [Dut+19 ; LG19 ; Cay+21] mais aussi de modifier le résultat d'un test conditionnel [SFM20]. Ces corruptions d'instructions sont souvent modélisées par un remplacement de l'instruction ciblée par une autre instruction [BGV11 ; Mor+13 ; SFM20 ; Als+22].

Saut ou rejeu d'instruction Un saut d'instruction correspond à la non-exécution d'une instruction d'un programme. Cela peut être utilisé pour éviter un branchement [BJC15] ou modifier une variable [KH14]. Il est également possible de modifier le flot d'exécution du programme [SH08] ou l'état intermédiaire d'un calcul [Deh+12]. La corruption d'une instruction peut aboutir à un saut d'instruction si l'instruction corrompue correspond à une instruction "`nop`" , si l'instruction sautée modifie le contenu d'une variable stockée en dehors de la mémoire du programme [Dut+19] ou si l'instruction d'arrivée est invalide. Le rejeu d'instruction correspond au remplacement d'instructions par le bloc d'instructions précédent. Il y a alors *duplication* ou *n-plication* du bloc d'instructions [CPT17 ; KDD21]. Un rejeu d'instruction qui n'a pas d'effet sur l'exécution du programme est dit *idempotent*. Le rejeu est donc équivalent à un saut d'instruction car tout se passe comme si la seconde exécution n'avait pas eu lieu et donc qu'elle a été sautée [Mor+14].

2.7 Scénarios d'attaque

Les attaques en fautes décrites précédemment permettent d'élaborer des schémas d'attaques sur des algorithmes cryptographiques. Ces derniers exploitent un modèle de faute afin d'affaiblir le niveau de sécurité de l'algorithme ciblé.

Les premiers travaux mettant en défaut un algorithme cryptographique sont connus sous le nom d'*attaque Bellcore* [BDL97 ; BDL01]. Cette attaque permet la factorisation des nombres premiers de la clé privée de l'algorithme Rivest-Shamir-Adleman (RSA) en injectant une faute pendant la génération de la signature et nécessite également une génération de la signature non fautée du même message. Cela a ouvert la voie à de nombreux autres schémas d'attaques dont les principaux sont décrits ci-dessous.

Un autre schéma d'attaque permettant de retrouver la clé secrète d'un algorithme de chiffrement est la *Differential Fault Analysis* (DFA). Proposée en 1997 par Biham et Shamir [BS97], elle nécessite d'avoir accès à des chiffrés fautés et à des chiffrés non fautés. Elle permet d'extraire la dernière clé de ronde de l'algorithme *Data Encryption Standard* (DES) et par extension la clé secrète de l'algorithme. D'autres travaux ont étendu cette attaque à l'algorithme AES [PQ03 ; BS03 ; CY03 ; DLV03 ; Gir04 ; KQ08 ; Muk09 ; SHP09 ; Ago+10a ; Ago+10b ; RDT13]. Les premières propositions de DFA sur la dernière ronde de l'AES ont été proposés par Giraud en 2004 [Gir04]. Le premier scénario nécessite la modification d'un seul bit de l'état avant l'opération *SubBytes* de la dernière ronde. Pour obtenir un octet de la clé de la dernière ronde, l'attaquant doit injecter des fautes monobit lors de trois chiffrements au minimum. Une des plus performante de ces attaques est celle de Piret et Quisquater [PQ03] dans laquelle ils parviennent à retrouver la clé de l'AES avec uniquement une paire de chiffré fauté/chiffré correct avec une complexité calculatoire d'environ 2^{40} si un seul octet de l'état est fauté avant l'opération *MixColumns* de la 9$^{\text{ème}}$ ronde.

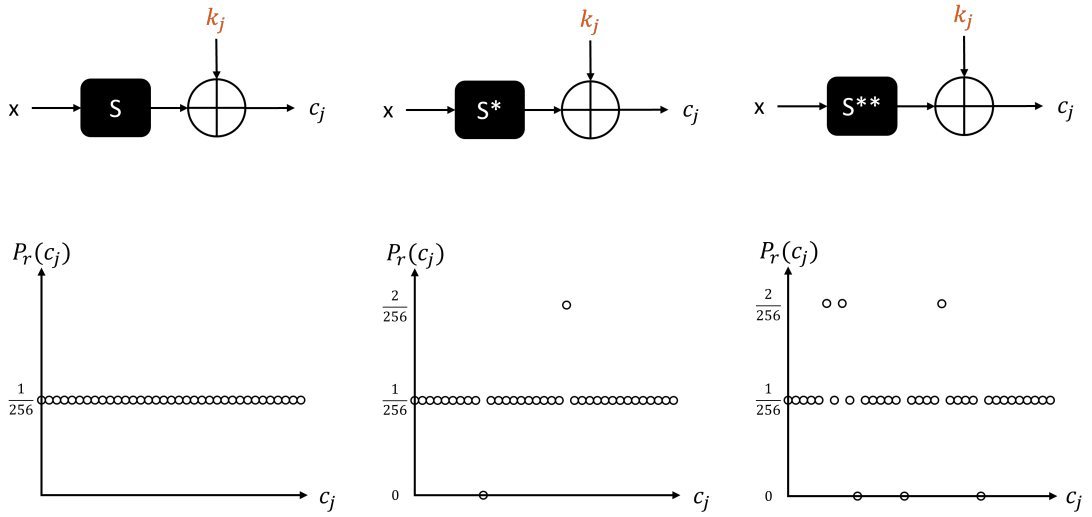
La majorité des schémas d'attaques développés jusqu'aujourd'hui sont basés sur des modèles de fautes transitoires. Comme nous l'avons vu en introduction de cette section, une faute **transitoire** est une faute ayant une durée de vie limitée. Ce sont des attaques qui sont difficiles à mettre en place car elles nécessitent que la faute soit injectée à un moment précis de l'exécution de l'algorithme. Une connaissance précise de l'exécution ou une recherche exhaustive de l'instant d'injection est donc nécessaire. Au contraire, on qualifie de **persistante** une faute qui est présente jusqu'à une réécriture de la donnée fautée. Dans le cas d'attaques d'algorithmes cryptographiques, ces fautes sont donc présentes pendant plusieurs chiffrements successifs avec des textes clairs différents.

Il existe une autre famille d'attaques qui exploitent les propriétés statistiques des fautes et des états intermédiaires d'un algorithme en présence de faute (SFA, SIFA, etc.). L'analyse décrite ci-après en fait partie, mais en considérant une faute persistante. Une analyse récente, proposée par Zhang *et al.* en 2018, nommée Persistent Fault Analysis (PFA), utilise l'injection de fautes persistantes au sein de la S-Box d'AES stockée en mémoire Flash d'un FPGA [Zha+18]. Un intérêt particulier est porté à cette analyse car elle sera mise en œuvre dans le Chapitre 5. Cette attaque diffère d'une DFA car elle ne nécessite pas d'exécution non fautée de l'algorithme de chiffrement. Le principe de cette attaque est d'injecter une ou plusieurs fautes dans les octets de la S-Box utilisée dans la transformation SubBytes. Une analyse statistique des octets du message chiffré est alors possible.

La distribution de probabilité des octets du chiffré $c_{j\in[\![0;15]\!]}$ considérant une S-Box non fautée S et incorporant la clé k_j est présentée en Figure 2.13a. On remarque bien une

équiprobabilité, égale à $\frac{1}{256}$, dans la distribution de probabilité d'apparition des valeurs de c_j.

À l'inverse, la Figure 2.13b montre la distribution de probabilité obtenue dans le cas où un octet de la S-Box, notée S^*, est fauté. Cette faute aboutit à une valeur présente deux fois plus souvent que la normale dans les octets du message de sortie notée c_j^{max} (correspondant a la valeur de S-box qui est présente deux fois v^*) et une valeur n'apparaissant plus notée c_j^{min} (correspondant a la valeur de S-box qui n'est plus présente v).



(a) Sans faute. **(b)** Avec un octet fauté. **(c)** Avec plusieurs octets fautés.

Figure 2.13 – Distribution de probabilité des octets du chiffré en fonction du nombre de fautes sur la S-Box (S).

Dans la première proposition de PFA [Zha+18], l'hypothèse est faite que l'attaquant connaît la position et la valeur de la faute injectée. Ainsi il connaît les valeurs v et v^*. Par élimination des c_j apparaissant au moins une fois dans les octets du chiffré, il peut retrouver directement la valeur de l'octet de clé k_j en utilisant la relation :

$$\mathbb{P}(C_j = c_j) = 0 \Rightarrow c_j = c_j^{min} = v \oplus k_j \Rightarrow k_j = c_j^{min} \oplus v \tag{2.2}$$

Afin d'obtenir la valeur c_j^{min} avec certitude, c'est-à-dire de savoir quelle valeur d'octet n'apparaît plus en sortie de l'algorithme de chiffrement, il est nécessaire d'effectuer un grand nombre de chiffrements. Dans la publication originale de la PFA, 2 272,9 chiffrements sont nécessaires en moyenne pour distinguer l'octet qui n'apparaît plus. Une version améliorée de la PFA a été proposée en 2020 par `Zhang` *et al.* dans laquelle, en utilisant l'estimation par maximum de vraisemblance (ou MLE pour *Maximum Likelihood Estimation*), seuls 1 640,7 chiffrements sont nécessaires en moyenne. Cette amélioration

n'est possible que dans le cas où un unique octet de la S-Box est fauté. La première version de l'analyse n'utilise que la connaissance de c_j^{min} pour retrouver k_j, l'amélioration proposée utilise aussi la connaissance de c_j^{max} et la relation qui les lie :

$$c_j^{max} = S^*[i] \oplus k_j = (S^*[i] \oplus S[i]) \oplus (S[i] \ \oplus k_j) = f \oplus c_j^{min} \tag{2.3}$$

Si un seul octet de la S-Box est fauté et que la valeur de la faute n'est pas connue l'analyse reste possible. En effet, Zhang *et al.* ont proposé en 2020 [Zha+20] une amélioration de leurs précédents travaux dans laquelle la valeur de la faute f peut-être retrouvée en utilisant 623 messages chiffrés en moyenne. Pour calculer la valeur de la faute, on utilise la relation suivante :

$$f = c_j^{min} \oplus c_j^{max} \tag{2.4}$$

Le calcul de f requiert moins de chiffrements que le calcul de c_j^{min} car la valeur de la faute est la même pour les 16 octets des messages chiffrés alors que c_j^{min} doit être calculé pour chaque octet.

De même, il est possible de retrouver l'indice de l'octet fauté de la S-box en analysant environ 300 messages chiffrés. Ainsi, la Tableau 2.3 synthétise le nombre moyen de chiffrements nécessaires pour retrouver les différentes valeurs utiles à l'analyse. Pour retrouver la position de la faute, on effectue une hypothèse sur `i` puis la dernière clé de ronde k_{10} est calculée. À partir de cette clé, on retrouve la sortie de la transformation *SubBytes* de la 9ème ronde pour vérifier si une sortie impossible de la S-Box est présente. Dans ce cas, l'hypothèse i est éliminée. Statistiquement, la bonne hypothèse est l'unique valeur pour laquelle S_{min} n'apparaît pas avec S_{min} la valeur n'apparaissant plus dans la S-Box.

Information	Valeur moyenne
Position de la faute (i)	300
Valeur de la faute (i)	623
c_j^{min}	2272.9

Tableau 2.3 – Nombre moyen de chiffrements nécessaires pour retrouver une information utile à la PFA.

L'analyse proposée en 2018 est également possible si plusieurs octets de la S-Box, notée S^{**}, sont fautés. En effet, la distribution de probabilité n'est pas uniforme donc la PFA est applicable. La Figure 2.13c illustre le cas où plusieurs valeurs n'apparaissent plus et

plusieurs valeurs apparaissent avec une probabilité de $\frac{2}{256}$. Dans l'exemple illustré trois octets sont fautés et trois valeurs de S-Box n'apparaissent plus. Par ailleurs, il est possible que deux octets différents soient fautés de façon différente pour aboutir à une seule et unique valeur.

En revanche, même si l'analyse est possible avec plusieurs octets fautés elle est beaucoup plus complexe à mettre en place et nettement moins efficace. La Figure 2.14 montre l'évolution du nombre de clés candidates selon le nombre d'octets de la S-Box fautés et le nombre de chiffrements [Zha+18]. En effet, dans ce scénario, plusieurs c_j peuvent être associés à c_j^{min} et c_j^{max}.

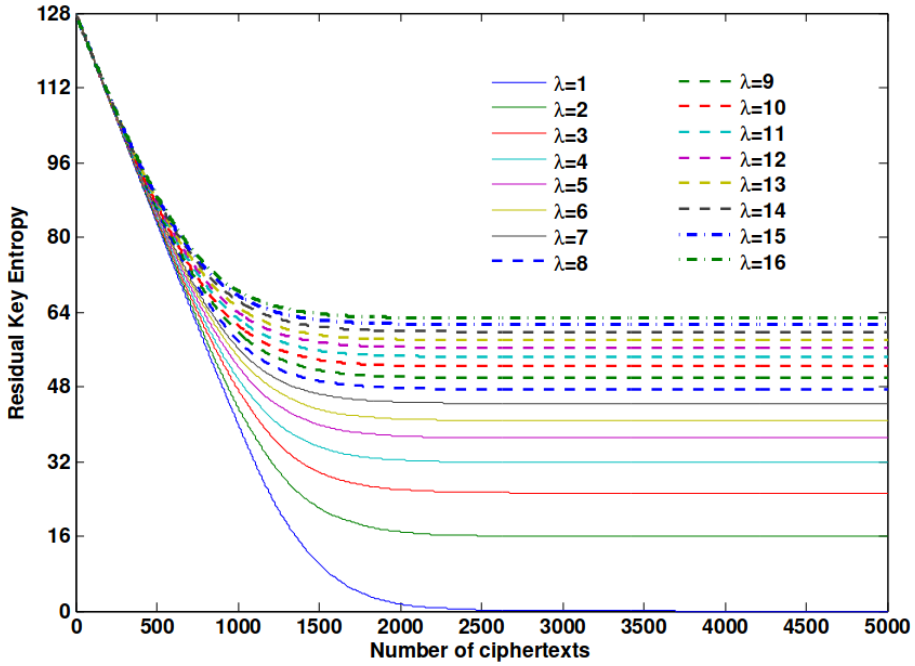


Figure 2.14 – Entropie résiduelle de la clé en fonction du nombre d'octets fautés et du nombre de textes chiffrés [Zha+18].

2.8 Mécanisme de protection

Les découvertes des failles matérielles ont abouti à l'élaboration de mécanismes de protection, également appelés contremesures, permettant de sécuriser les composants électroniques notamment dans le cadre d'applications sensibles (bancaires ou historiquement la télévision payante) [AK96 ; Wag12].

On appelle durcissement l'ensemble des mesures prises visant à améliorer la sécurité des composants électroniques. Ces techniques s'étendent de la conception, à l'implémentation en passant par la fabrication des systèmes [Ker+88]. Les principales techniques de durcissement utilisées sont décrites ci-après.

La redondance est le fait de répéter plusieurs fois la même opération et de comparer les résultats obtenus. Ce principe est basé sur l'hypothèse qu'un attaquant ne peut pas répéter plusieurs fois la même attaque. La redondance peut être spatiale, lorsque plusieurs instances effectuent la même opération en parallèle, ou temporelle, lorsqu'une seule instance effectue la même opération plusieurs fois [Bar+04]. Il est également possible d'effectuer de la redondance d'information. Dans ce cas, une information supplémentaire est ajoutée à la donnée sensible. Elle permet de vérifier l'intégrité des données. Les bits de parité et les *Error Correcting Code* (ECC) sont des applications courantes de ce principe. Les différents types de redondance sont utilisés pour détecter, mais aussi pour corriger une faute.

Les mécanismes de *Control Flow Integrity* (CFI) vérifient le flot d'exécution d'un programme. Cela concerne principalement le bon déroulement des branchements et des retours de fonction. Ces vérifications peuvent être réalisées de façon logicielle [LHB14 ; Pro+17] ou matérielle [Cle+16 ; Dan+18] ou les deux [CCH22]. Ces mécanismes empêchent un attaquant d'exécuter une série d'instructions non désirée ou de "sauter" une ou plusieurs instructions.

L'obfuscation désigne l'ensemble des techniques utilisées pour dissimuler la manière de fonctionner du système. Il est possible de modifier l'ordre d'exécution des instructions [RPD09], d'effectuer des opérations de masquage des données [RPD09], d'insérer des instructions factices [GST12] ou encore d'utiliser une horloge irrégulière [Moo+02] afin d'introduire de l'aléa dans l'exécution temporelle d'un algorithme. Le *scrambling* est également une méthode d'obfuscation des mémoires et des bus de données qui permet de décorréler les données de leur représentation [JMR07]. L'obfuscation peut également être réalisée lors de la fabrication du circuit ou dans les étapes de placement et de routage afin de cacher les fonctions logiques et leur implémentation [Vij+17].

La présence de capteurs matériels peut également permettre la détection d'intrusion ou de faute. Ces derniers peuvent détecter la perturbation du composant par de la lumière [Cha+15], des variations du signal d'horloge et de la tension d'alimentation [Des+16], d'une Injection électromagnétique (EMI) [Hom+14] ou d'une Impulsion électromagnétique (EMP) [ERM16] ou d'une violation de contraintes temporelles [Zus+14]. La détection d'une injection de faute permet alors au composant de réagir. Ces capteurs sont dits *actifs* car ne fonctionnent que lorsque le circuit est alimenté. Ce type de protection n'est donc pas efficace dans le cadre d'attaques sur circuits non alimentés, ce qui renforce l'intérêt de ces attaques.

Il existe d'autres mécanismes de protection, logiciels ou matériels, spécifiques à certaines attaques. C'est notamment le cas de `BALoo` [TBG23], proposé en 2023 par `Tissot` *et*

al., qui vérifie l'intégrité des S-Box dans le cadre des attaques par analyse de fautes persistantes. Il est également possible d'empêcher l'accès physique et la manipulation du circuit en plaçant des protections spécifiques à certaines attaques matérielles. On peut citer par exemple les boucliers actifs ou passifs contre l'injection laser de fautes, ou l'utilisation d'horloge et d'alimentation internes contre les attaques par *glitchs*.

Chapitre 3

État de l'art

Table des matières

3.1 Introduction

Le troisième chapitre de ce manuscrit est consacré à l'état de l'art nécessaire à la bonne compréhension des chapitres suivants. Les différents vecteurs pouvant porter atteinte à l'intégrité des circuits électroniques explorés dans cette étude sont les injections laser, les radiations et le vieillissement.

Dans un premier temps, les effets du laser sur les circuits électroniques sont abordés. Ces derniers seront traités d'un point de vue hiérarchique en allant du plus bas niveau, le niveau physique, au plus haut niveau, le niveau logiciel. Dans un second temps, les effets des radiations sont étudiés. Cette étude comporte une description des différents environnements radiatifs, les interactions entre les radiations et la matière et, pour finir, les effets des radiations sur l'électronique. Dans un dernier temps, les effets du vieillissement, qui interviennent naturellement au cours de la vie du composant, seront décrits.

3.2 Effet du laser sur les circuits intégrés

La description de l'effet du laser sur les circuits intégrés a fait l'objet de nombreuses études. Cette section synthétise les principaux résultats sur l'injection laser présents dans l'état de l'art et servants de base au Chapitre 4 et au Chapitre 5. Cette description sera d'abord abordée au niveau physique, puis au niveau logique et pour finir au niveau logiciel pour une cible programmable. Ces trois niveaux ne sont pas indépendants mais complémentaires. Un intérêt particulier sera porté aux attaques sur les mémoires Flash car ce sont les cibles visées dans les travaux décrits dans les chapitres suivants.

3.2.1 Niveau physique

3.2.1.a Structure électronique

Le comportement électrique des métaux, des isolants et des semi-conducteurs est décrit par la théorie des bandes [YC96]. Ce modèle stipule que les électrons d'un matériau ne peuvent prendre que certaines valeurs d'énergie comprises dans des intervalles, les bandes permises, qui sont séparées par des bandes interdites.

Il existe deux bandes notables :

— la bande de valence : dernière bande entièrement remplie,
— la bande de conduction : bande d'énergie permise suivante responsable de la conduction.

Dans les conducteurs, les bandes de valence et de conduction se chevauchent. Cela permet aux électrons de se déplacer librement dans les deux bandes et de circuler dans le matériau.

Dans les isolants, la bande interdite, souvent appelée *gap*, séparant les bandes de valence et de conduction, est grande (environ 6 eV). Il n'y a donc pas de circulation d'électrons et par extension pas de courant. Le matériau est donc isolant.

Dans les semi-conducteurs, le gap est de l'ordre de 1 eV (1,12 eV pour le silicium). Les électrons peuvent passer de la bande de valence à la bande de conduction si une énergie extérieure (élévation de température, champ électrique, illumination) est apportée aux électrons. Dans ces conditions, les électrons peuvent circuler dans le matériau. En l'absence d'excitation extérieure, le matériau se comporte comme un isolant. Le dopage peut également rendre ces matériaux conducteurs. Dans le cas contraire, ils sont dits "intrinsèques" et sont isolants.

La Figure 3.1 illustre la théorie des bandes pour les métaux isolants, semi-conducteurs et conducteurs

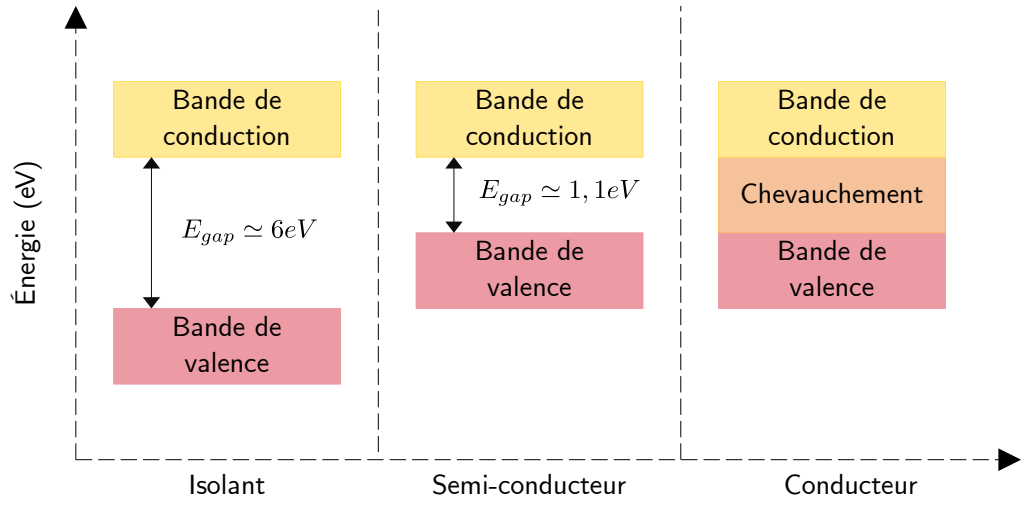


Figure 3.1 – Position des différentes bandes d'énergie.

3.2.1.b Effet photoélectrique

L'effet photoélectrique, décrit en 1 905 par `Albert Einstein`, désigne l'émission d'électrons par un matériau sous une exposition à la lumière. Ce phénomène introduit la notion de photon en tant que particule élémentaire de lumière et s'applique dans le cas d'une exposition d'un matériau semi-conducteur à un faisceau laser.

L'exposition d'un matériau semi-conducteur à un rayon laser provoque un transfert d'énergie des photons du rayon aux électrons du matériau. L'énergie d'un photon dépend

de sa longueur d'onde λ, de la vitesse de la lumière c et de la constante de Planck h selon l'Équation 3.1.

$$E_{photon} = \frac{hc}{\lambda} \tag{3.1}$$

Si un photon possède une énergie suffisante, c'est-à-dire supérieure à l'énergie du gap, il peut être absorbé par le matériau par interaction photoélectrique et transférer un électron de la bande de valence vers la bande de conduction. En appliquant l'Équation 3.1 au cas du silicium, on obtient (Équation 3.2) que l'effet photoélectrique intervient dans le silicium pour des longueurs d'ondes inférieures à 1 100 nm. Il y a ainsi création de paires électron/trou dans le matériau semi-conducteur. Ce phénomène est représenté sur la Figure 3.2.

$$E_{photon} > E_{gap} \Rightarrow \lambda < \frac{h \times c}{E_{gap}} \simeq 1\,100\,\text{nm} \tag{3.2}$$

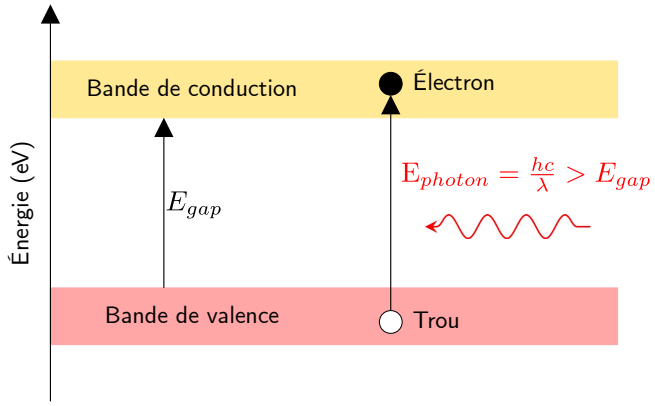
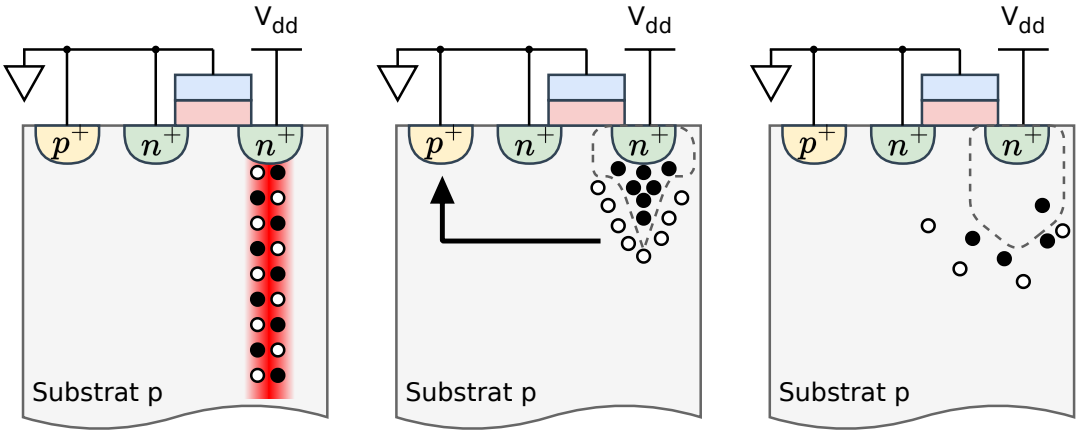


Figure 3.2 – Absorption d'un photon par effet photoélectrique.

En l'absence de champ électrique, les paires électron-trou générées dans le substrat se recombinent naturellement. En revanche, lorsqu'un champ électrique est présent, comme dans les jonctions PN polarisées en inverse, les charges sont séparées et un photocourant apparaît [Hab65]. En technologie CMOS, on trouve généralement ces zones dans la Zone de charge d'espace (ZCE) à l'interface entre le substrat et le drain des transistors bloqués.

L'apparition d'un photocourant se fait en trois étapes comme le montre la Figure 3.3 [Bau05]. Dans un premier temps, le faisceau laser génère des paires électron-trou tout au long de son passage dans le substrat (Figure 3.3a). Dans un deuxième temps, ces paires sont séparées et un courant important apparaît au niveau des jonctions PN polarisées en inverse (Figure 3.3b). Les électrons sont collectés au niveau des ZCE et les trous

au point de polarisation du substrat. Dans un dernier temps, les porteurs de charges diffusent dans le substrat (Figure 3.3c).



(a) Tir laser. **(b)** Courant de conduction. **(c)** Courant de diffusion.

Figure 3.3 – Mécanisme d'apparition d'un photocourant.

Le photocourant ainsi produit est représenté sur la Figure 3.4. Il atteint sa valeur maximale pendant la phase de collection causée par le courant de conduction. On remarque l'aspect transitoire du photocourant généré.

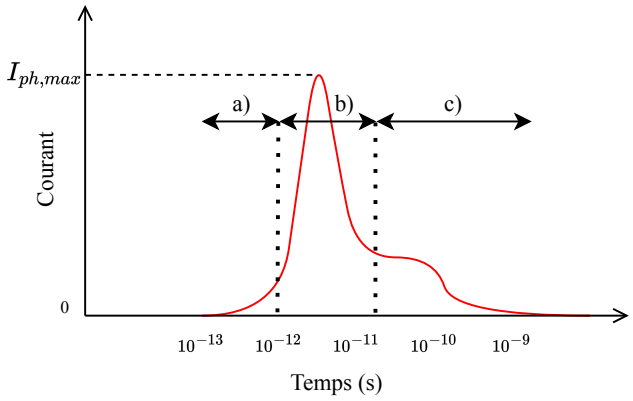


Figure 3.4 – Évolution temporelle du photocourant [Hab65].

3.2.1.c Modélisation électrique

Les effets d'un tir laser sur un transistor peuvent être modélisés par une source de courant entre le drain et le substrat du transistor [Dou+05]. Le schéma électrique représentatif de cette modélisation est présenté sur la Figure 3.5. Les résistances R_0, R_1 et la capacité C représentent le chemin de conduction dans le substrat.

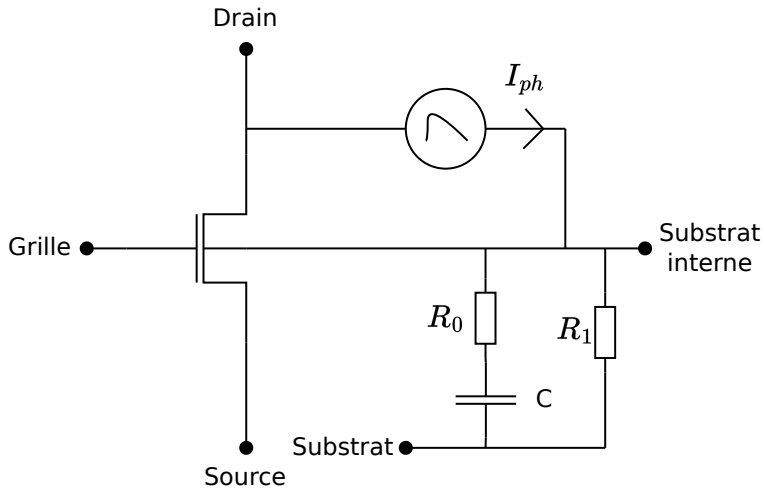


Figure 3.5 – Modélisation électrique d'un tir laser sur un transistor NMOS [Dou+05].

Soient V_{DB} la tension appliquée sur la jonction Drain-Substrat du transistor et S la surface de la jonction, la valeur maximale du photocourant $I_{ph,max}$ peut être approximée par l'Équation 3.3 [Sar+13a].

$$I_{ph,max} = (a \times V_{DB} + b) \times \alpha_{gauss}(x, y) \times S \tag{3.3}$$

avec :

- a, b des polynômes dépendants de la puissance de la source laser
- α_{gauss} représentant le profil gaussien de l'intensité du faisceau laser
- S la surface de la jonction PN considérée

3.2.1.d Effet thermique

En parallèle de l'effet photoélectrique, un rayon laser apporte également de l'énergie au circuit causant une élévation locale de température [San11]. Un effet thermique est présent pour des sources laser ayant une longueur d'onde permettant à l'effet photoélectrique d'intervenir. En revanche, pour des sources laser avec une longueur d'onde de l'ordre de 1 300 nm-1 400 nm, l'effet photoélectrique n'intervient plus mais l'effet thermique est encore présent. C'est notamment l'effet thermique qui a permis de modifier l'état de cellule `OxRAM` résistive à base de HfO_2 [Kra+16 ; Kra+17]. L'effet thermique a également permis de retrouver une clé de chiffrement stockée dans la BBRAM d'un FPGA [Loh+18].

3.2.1.e Effet physique sur les transistors à grille flottante

Les effets du laser sur les transistors à grille flottante peuvent aussi être exploités au niveau physique. En 2009, `Skorobogatov` [Sko09] relie l'augmentation de température liée à l'injection laser à la décharge des transistors à grille flottante d'une mémoire. Une seconde étude [Cha+17] décrit une ionisation à deux photons qui implique la décharge des grilles flottantes vers le substrat en utilisant un laser femtoseconde.

3.2.1.f Effet physique sur une mémoire Flash pendant la lecture

Le mécanisme de lecture des mémoires Flash est décrit dans la sous-section 2.4.1. Dans une mémoire NOR Flash les drains de tous les transistors à grille flottante de la colonne sont connectés à une même *bitline*. Les jonctions PN entre les drains et le substrat de ces transistors sont alors polarisées en inverse. L'effet photoélectrique décrit dans la sous-sous-section 3.2.1.b peut alors s'appliquer. On a donc l'apparition d'un courant I_{ph} comme représenté sur la Figure 3.6. Le mécanisme sera détaillé dans la sous-sous-section 3.2.2.d.

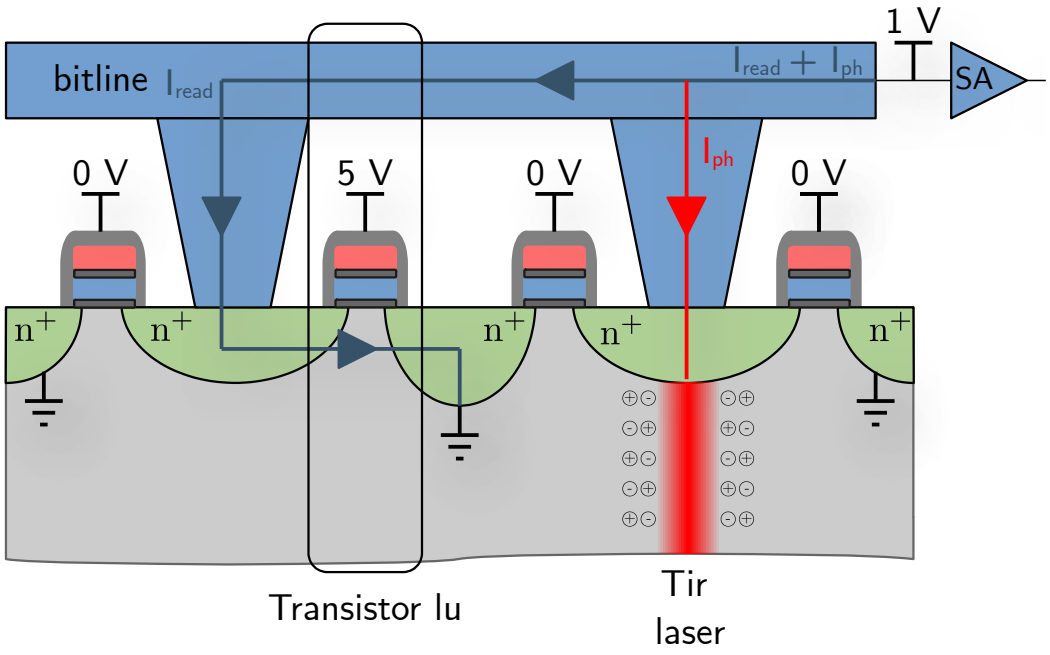


Figure 3.6 – Mécanisme d'apparition d'un photocourant dans une colonne d'une mémoire NOR Flash.

3.2.2 Niveau logique

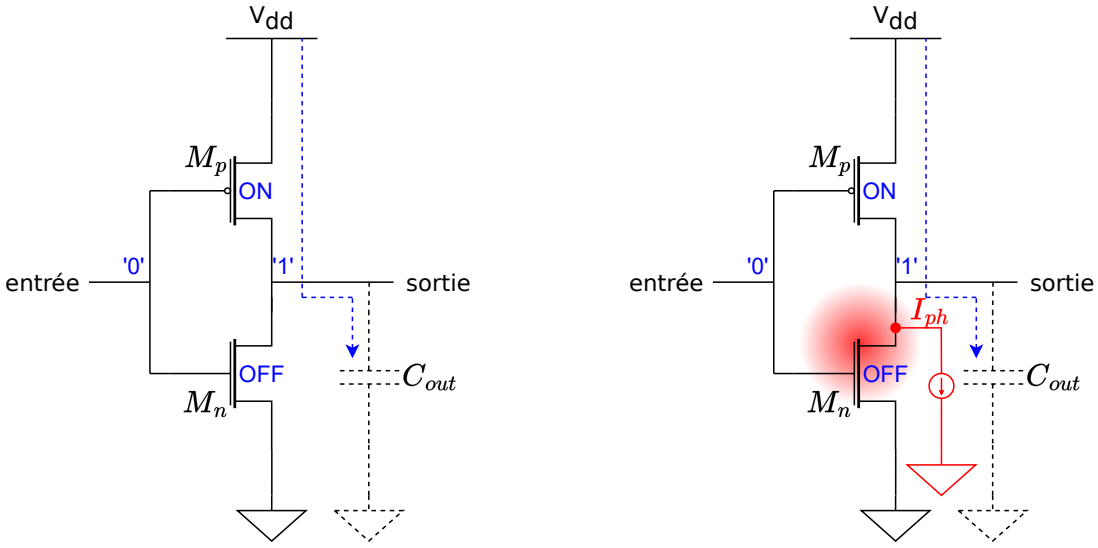
3.2.2.a Effet logique sur un inverseur CMOS

L'inverseur CMOS est une porte logique élémentaire constituée d'un transistor NMOS et d'un transistor PMOS qui fonctionnent de façon complémentaire. Le signal de sortie

est l'inverse du signal d'entrée. Ils sont notamment présents dans les circuits logiques, les *buffers* et les sources de courant pilotées en tension.

Le fonctionnement d'un inverseur en conditions normales est illustré sur la Figure 3.7a. Dans la configuration illustrée l'entrée est polarisée à la masse (‘0’), le transistor M_n NMOS est bloqué et la capacité de sortie C_{out} se charge via le transistor PMOS M_p. Le potentiel de sortie augmente alors jusqu'à atteindre la tension d'alimentation V_{dd} (‘1’). En statique, une fois la capacité C_{out} chargée, il n'y a plus de courant de charge.

Dans cette configuration, c'est la jonction PN à l'interface drain-substrat du transistor M_n qui est polarisée en inverse. Un photocourant I_{ph} peut donc y être injecté par illumination laser. Ce cas est représenté sur la Figure 3.7b. La capacité C_{out} peut ainsi se décharger temporairement dans le substrat du transistor M_n avant d'être chargée à nouveau par le transistor M_p. Un *glitch* (c'est-à-dire une chute brève) de tension, ou transitoire de tension, sera donc observé en sortie de l'inverseur. Ce *glitch* pourra se propager et insérer une erreur transitoire dans les données qui transitent ou une erreur de calcul.



(a) Fonctionnement d'un inverseur en conditions normales.

(b) Fonctionnement d'un inverseur sous illumination laser.

Figure 3.7 – Effet d'un tir laser sur un inverseur CMOS.

Le *glitch* ainsi généré (SET) peut se transformer en une faute de mémorisation (SEU) s'il a lieu directement dans une cellule SRAM ou une bascule D. La largeur du *glitch* augmente avec la durée de l'impulsion laser. Il est donc possible de réaliser une augmentation progressive de la largeur de l'impulsion laser afin d'induire une violation progressive des contraintes temporelles et de transformer les fautes obtenues d'aléatoires en déterministes. Cette méthodologie a été utilisée en 2016 [Sch+16] pour réaliser une FSA

(Fault Sensitivity Analysis [Li+10]) sur une implémentation matérielle d'un algorithme de chiffrement.

3.2.2.b Effet logique sur une cellule SRAM

Le mécanisme d'effet photoélectrique décrit précédemment s'applique aux jonctions PN des transistors bloqués de la cellule SRAM. Si la cellule est à l'état haut (`'1'`), cela concerne les transistors M_{p1} et M_{n2} de la Figure 3.8. La Figure 3.8 illustre le cas où le transistor M_{n2} est visé. Dans ce cas, le photocourant généré au niveau du drain du transistor M_{n2} permet à la capacité C_2 de se décharger. Si ce photocourant dépasse une valeur seuil, une faute du type *bitreset* est constatée. Une faute similaire est obtenue si le transistor M_{p1} est illuminé. À l'inverse, si la cellule est à l'état bas (`'0'`), cela concerne les transistors M_{n1} et M_{p2} de la Figure 3.8 et la faute obtenue est du type *bitset*. Dans les deux cas, le courant transitoire induit par l'illumination laser entraîne le basculement de la cellule SRAM dans l'état inverse.

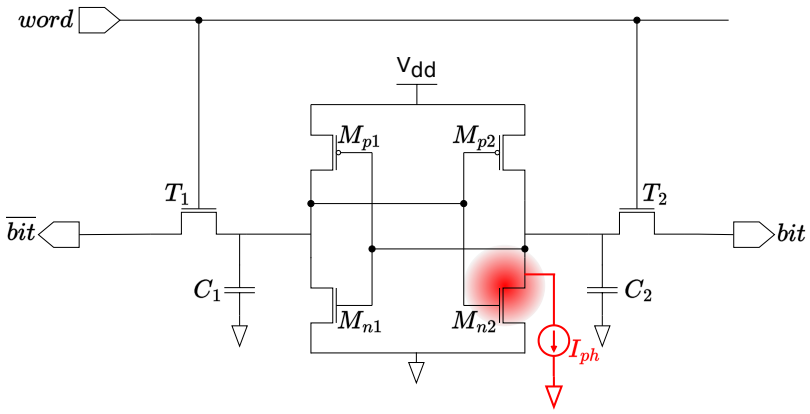


Figure 3.8 – Injection d'une faute sur une cellule SRAM. Le cercle rouge représente le spot laser.

L'illumination laser d'une cellule SRAM résulte en l'injection d'une faute du type *bitset* ou *bitreset* selon la position du spot laser [Sar+13b ; Ros+13]. Ces résultats ont permis d'altérer de façon très précise des variables, constantes ou instructions d'algorithmes stockées en SRAM d'un microcontrôleur afin d'en extraire une clé de chiffrement [Ago+10a ; Ago+10b ; Dut+12 ; Zha+20] ou de modifier l'exécution d'un programme [Dut+12].

3.2.2.c Effet logique sur les registres généraux

Il a également été démontré qu'il est possible de modifier l'état des registres généraux d'un MCU en injectant des fautes laser. Ces attaques permettent de modifier l'état ou l'exécution d'un algorithme [TK10 ; BJ15 ; KMW17 ; Vas+17].

3.2.2.d Effet logique sur la mémoire Flash pendant la lecture

Le modèle de fautes en mémoire Flash pendant l'opération de lecture décrit au niveau physique dans la sous-sous-section 3.2.1.f a un impact au niveau logique comme illustré sur la Figure 3.6. Le courant I_{ph} vient s'ajouter au courant I_{read} usuel et modifie la valeur de sortie du comparateur de courant. Les transistors d'une même colonne de la mémoire NOR Flash étant connectés au même comparateur de courant, tous les transistors à grille flottante de la colonne seront lus fautés. Il est uniquement possible de créer un photocourant, pas d'en atténuer un existant, ce qui justifie l'unidirectionnalité des fautes observées [Col+19 ; Men+20b]. Afin d'obtenir un modèle de faute bidirectionnel, il faudrait être capable de réduire le courant appelé sur la *bitline*, ce qui est impossible d'après la description du modèle de faute au niveau physique.

Ce modèle de faute permet de retrouver partiellement l'organisation physique d'une mémoire Flash. En effet, `Menu` [Men21] a observé la présence de 32 colonnes distinctes dans la mémoire Flash d'un microcontrôleur 32 bits (STM32F100RB). Pour ce composant, les données sont stockées sous la forme de mots de 32 bits et chacune des 32 colonnes observées correspond à un bit d'un mot de 32 bits. Cette dépendance spatiale est représenté sur la Figure 3.9. On peut remarquer que la $i^{ème}$ colonne contient les bits d'indice i des mots de 32 bits. On observe également que les fautes obtenues sont unidirectionnelles de type *bitset*.

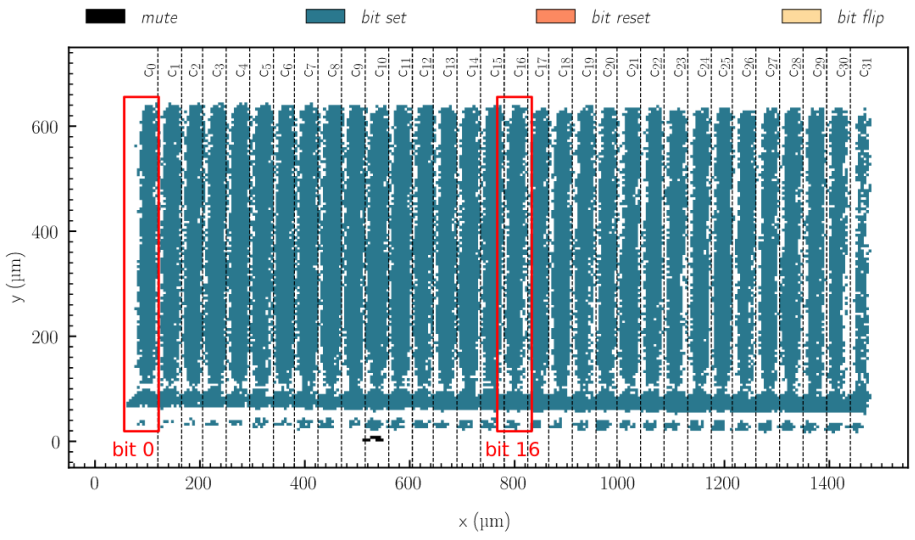


Figure 3.9 – Cartographie spatiale des fautes sur la mémoire Flash d'un STM32F100RB [Men+19].

3.2.3 Niveau logiciel

3.2.3.a Corruption d'instructions

Skorobogatov a démontré en 2005 qu'il est possible de corrompre une instruction lue en mémoire Flash en injectant un photocourant dans le décodeur de colonnes ou dans les entrées d'un *Sense Amplifier* [Sko05]. Ces travaux ont été améliorés par Sakamoto *et al.* [SFM20] dans laquelle chacune des entrées des *Sense Amplifiers* sont ciblées pour obtenir des fautes du type *bitset* ou *bitreset*. La précision spatiale et temporelle du laser permet de cibler précisément l'instruction et l'indice du bit au sein de l'instruction. Une durée d'impulsion laser plus longue permet de cibler plusieurs instructions successives.

Il est également possible de corrompre les instructions lues en mémoire Flash en ciblant les références de courant utilisées en entrée des *Sense Amplifiers*. Kumar *et al.* [Kum+18] ont observé des *bitresets* en effectuant des injections lasers sur les références de courant de la mémoire Flash d'un microcontrôleur 8 bits.

D'autres travaux de Skorobogatov [Sko10a] ont montré qu'il était également possible de mettre à zéro les données lues en mémoires Flash en ciblant la logique de contrôle. Il est également possible d'empêcher l'écriture d'un mot en mémoire en ciblant la logique de contrôle [Sko10b]. Dans ces travaux, la taille du bloc de données fautées est choisie en ajustant l'instant et la durée de l'injection laser.

Les travaux de Colombier *et al.* [Col+19] et Menu *et al.* [Men+20b] utilisent l'injection laser lors de l'opération de lecture d'instructions en mémoire Flash afin de les corrompre. Les instructions stockées en mémoire sont constituées d'un opcode et potentiellement d'un registre source, d'un registre destination et d'une valeur immédiate. En fautant l'opcode, on modifie le type d'instruction réalisée. En fautant les autres éléments, le résultat de l'opération peut être fauté. Il est ainsi possible de modifier l'instruction réalisée ou les données manipulées. C'est cette attaque qui a initié les avancées récentes en injection de fautes par laser. La Figure 3.10 illustre un exemple de corruption d'instructions obtenu dans [Col+19]. L'exemple choisi est l'instruction `MOVW R0, 0`. Dans le premier cas, c'est la valeur immédiate stockée sur 16 bits qui est fautée afin de transformer l'instruction en `MOVW, R0, 4`. Dans le deuxième cas, c'est le registre de destination qui est fauté afin de changer l'instruction en `MOVW, R1, 0`. Pour finir, dans le troisième cas, c'est l'opcode de l'instruction qui est fauté pour passer à `MOVT, R0, 0`.

Il est ainsi possible de constater qu'un banc laser monospot ne permet de fauter qu'un seul bit (ou plus si le spot est assez large et placé sur plusieurs bits adjacents). Le Chapitre 4 décrit comment il est possible de s'affranchir de cette limite en utilisant un banc laser multispot.

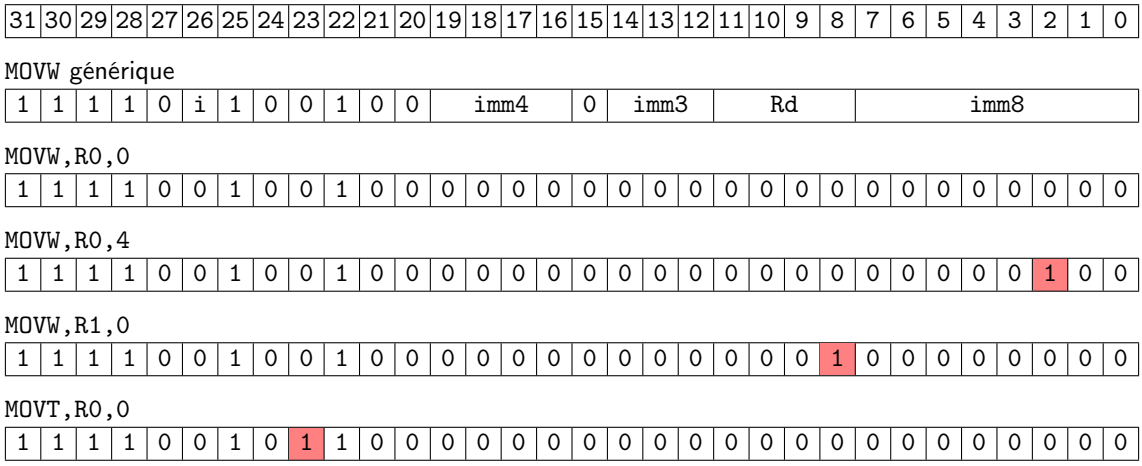


Figure 3.10 – Exemples de corruption d'instructions lors de l'opération de lecture [Col+19].

3.2.3.b Saut d'instructions

En 2015, `Breier` *et al.* [BJC15] ont réussi à réaliser des sauts d'instructions en synchronisant l'injection laser sur la logique de contrôle de la mémoire Flash avec l'exécution du programme ciblé. Ces travaux leur permettent de sauter la fonction `AddRoundKey` de la dernière ronde de l'AES afin de réaliser une attaque différentielle pour retrouver la clé de chiffrement de l'algorithme. Ces travaux sont améliorés en 2019 par `Dutertre` *et al.* [Dut+19] lors d'une étude dans laquelle la position et la taille du bloc d'instructions sautées sont choisies en ajustant la durée et l'instant du tir laser. Cette attaque leur permet de corrompre plusieurs tests conditionnels d'un algorithme de vérification d'un code PIN en temps constant.

3.3 Effets des radiations

On appelle *radiation* l'émission ou la transmission d'énergie sous forme d'onde ou de particule. Dans cette étude, on s'intéresse exclusivement aux radiations électromagnétiques et aux radiations particulaires.

Les radiations électromagnétiques désignent la propagation d'ondes électromagnétiques. Cela inclut les ondes radio et micro-ondes, infrarouges, visibles et UV, les rayons X et les rayons γ. La Figure 3.11 représente la répartition des différentes radiations électromagnétiques selon leur fréquence. Les radiations particulaires désignent la propagation de particules énergétiques c'est-à-dire à grande vitesse. Cela inclut les électrons, les neutrons, les protons et les ions lourds. Ces particules peuvent porter des charges électriques ou non.

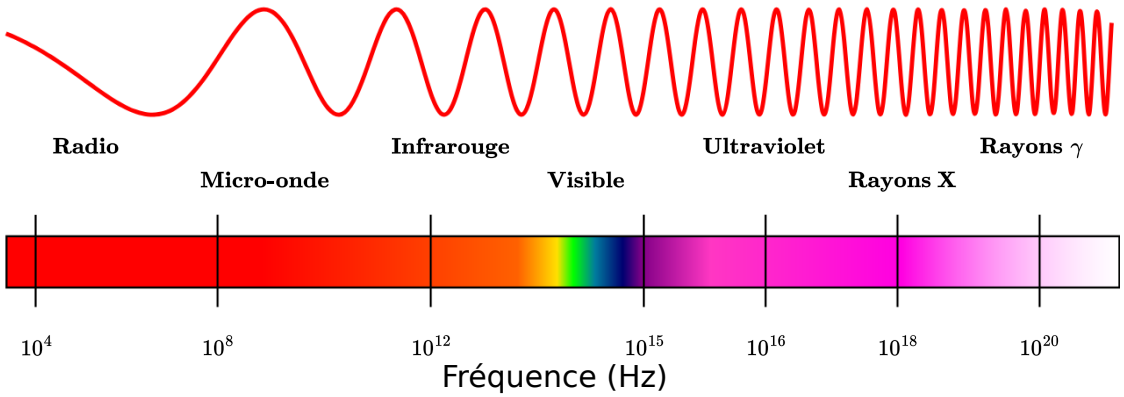


Figure 3.11 – Spectre électromagnétique.

Une distinction est faite entre les radiations non ionisantes et les radiations ionisantes. Ces dernières sont suffisamment énergétiques pour arracher des électrons à la matière et donc de ioniser le matériau traversé. La frontière entre ces deux types de radiations n'est pas strictement définie car des matériaux différents sont ionisés à des énergies différentes [Fou20 ; Mey23].

3.3.1 Environnements radiatifs

3.3.1.a Radiations spatiales

Radiations interplanétaires

La principale source de radiations interplanétaires est le Soleil. Ce dernier émet des radiations couvrant la majeure partie du spectre électromagnétique allant des rayons γ aux rayons X, aux ondes radios et micro-ondes. Les radiations qu'il émet majoritairement possèdent des longueurs d'ondes comprises entre 100 nm et 1 mm (ce qui inclut l'ultraviolet, le visible et l'infrarouge). Le Soleil émet également un flux continu de particules composé principalement d'électrons et de protons. C'est ce flux que l'on appelle *vent solaire*.

En plus du vent solaire, le Soleil émet ponctuellement des rafales massives de plasmas. Ces évènements peuvent être des *éruptions solaires* ou des *éjections coronales de masse* (ou CME pour Coronal Mass Ejection). Le plasma éjecté est majoritairement composé de protons et d'électrons.

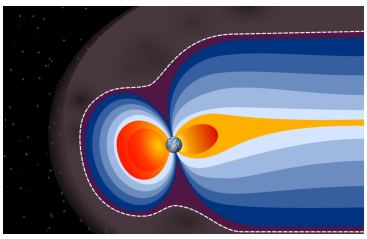
Une autre source de radiation interplanétaire sont les *rayons cosmiques galactiques* (ou GCR pour Galactic Cosmic Rays) qui sont composés de noyaux à haute énergie allant de l'hydrogène (environ 89% du flux) à l'uranium (seulement quelques traces). Ces rayons ne proviennent pas du système solaire mais de supernovas très éloignées.

Environnement proche de la Terre

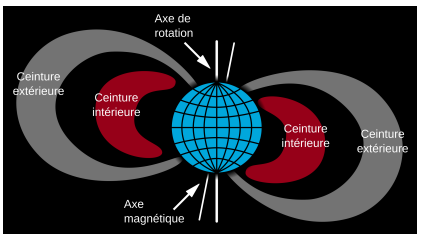
La Terre génère un champ magnétique de type dipôle incliné de 11° par rapport à son axe de rotation. Ce champ magnétique est causé par des mouvements de métaux conducteurs à l'état liquide (essentiellement du fer et du nickel) dans le noyau externe de la Terre. Ces mouvements apparaissent sous l'effet de la convection thermique et des forces de Coriolis provoquées par la rotation de la Terre. Lorsque les particules chargées (générées par les vents solaires ou les GCR) rencontrent le champ magnétique terrestre, elles sont déviées par la force de `Lorentz` qui est proportionnelle à leur vitesse et à l'amplitude du champ magnétique. Cette interaction définit la *magnétosphère* visible en Figure 3.12a. Ce phénomène a deux impacts sur l'environnement radiatif proche de la Terre.

Les particules qui possèdent une faible rigidité magnétique [1] (notamment celles issues des GCR) peuvent être déviées loin de la Terre. Un effet de blindage géomagnétique les empêche d'atteindre les zones où le champ géomagnétique est le plus intense.

Les particules les moins énergétiques et les moins rigides (les protons et les électrons) ont des trajectoires tellement courbées qu'elles peuvent être piégées dans une zone (allant de quelques centaines de kilomètres à environ 60000 kilomètres d'altitude) que l'on appelle la *ceinture de Van Allen* (voir Figure 3.12b). Les particules piégées dérivent autour de la Terre selon la charge qu'elles portent (vers l'est pour les électrons et l'ouest pour les protons). Elles se déplacent donc de façon hélicoïdale autour des lignes de champ magnétique en "rebondissant" entre les deux pôles magnétiques. Lorsque les lignes de champ se rapprochent de la Terre à proximité des pôles, ces particules interagissent avec les atomes de la haute atmosphère et génèrent la formation d'un plasma "froid". Ce phénomène est à l'origine des aurores boréales.



(a) Illustation de la magnétosphère. Les ceintures de Van Allen sont situées dans la zone orange.



(b) Illustration des ceintures de Van Allen.

Figure 3.12 – Magnétosphère et ceintures de Van Allen. Source NASA/Wikimedia Commons.

Les particules piégées dans le champ magnétique terrestre se concentrent à différentes altitudes selon leur masse et leur vitesse. Les ceintures de Van Allen sont constituées

1. Quantité représentative du moment de la particule rapporté à sa charge électrique.

d'une petite ceinture interne et d'une grande ceinture externe. La ceinture interne est constituée de protons (avec des énergies allant jusqu'à 400 MeV) et d'électrons (avec des énergies allant jusqu'à 5 MeV). La ceinture externe est exclusivement constituée d'électrons ayant des énergies allant jusqu'à 7 MeV. Les structures de ces ceintures peuvent être altérées lors d'évènements solaires comme les éruptions ou les CME.

Comme le centre de la magnétosphère est légèrement décalé par rapport au centre terrestre, il existe une zone au-dessus de l'Amérique du Sud et de l'océan Atlantique où la frontière intérieure de la ceinture interne est proche de la haute atmosphère (environ 200 km au lieu de 1 000 km). Cette région (nommée SAA pour South Atlantic Anomaly et représentée sur la Figure 3.13) possède de hauts niveaux de radiations à des altitudes relativement basses et représente une menace pour les objets électroniques.

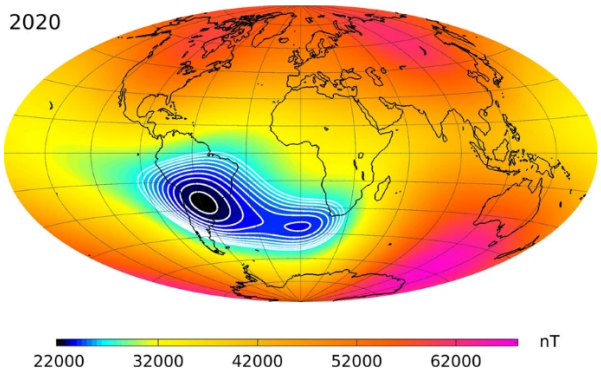


Figure 3.13 – Champ magnétique terrestre et anomalie de l'Atlantique sud. Source [Fin+20].

3.3.1.b Radiations atmosphériques

Comme indiqué précédemment, le champ géomagnétique terrestre peut dévier certaines particules chargées. En revanche, les particules chargées ayant le bon angle d'incidence et une énergie suffisante peuvent pénétrer dans la haute atmosphère. Ces ions vont générer des particules à haute énergie (muons/antimuons, pions, photons gamma, etc.) par plusieurs interactions nucléaires successives appelées *pluie de rayons cosmiques*. Ce sont ces particules et notamment les neutrons et les muons/antimuons qui représentent une menace pour la fiabilité des composants électroniques des équipements œuvrant à haute altitude.

3.3.1.c Sources de radiation artificielles

Il existe également de nombreux cas d'usage qui mettent en œuvre des circuits électroniques dans des milieux radiatifs. Les sources de radiations peuvent être des antennes,

des lasers, des accélérateurs de particules, des réacteurs nucléaires, etc. Toutes ces applications couvrent un large spectre de particules et d'énergie avec de nombreuses conséquences possibles pour les objets électroniques.

On peut prendre pour exemple la stérilisation (produits pharmaceutiques, agricoles ou alimentaires) qui est souvent effectuée en utilisant des rayons γ afin de désinfecter ou d'augmenter la durée de conservation. On peut également citer l'utilisation des rayons γ ou des rayons X pour l'inspection de bagages ou de personnes dans les aéroports. Dans l'industrie des semi-conducteurs, la fabrication des circuits imprimés (ou PCB pour Printed Circuit Board) est contrôlée à l'aide de rayons X. Cela permet de vérifier l'intégrité des soudures qui ne sont pas visibles sur les boîtiers BGA (*Ball Grid Array*). Cela pose la question de la sensibilité des circuits électroniques face à une exposition prolongée aux radiations.

Les accélérateurs de particules sont des installations dans lesquelles des particules chargées sont accélérées en utilisant des champs électriques et magnétiques. Ils sont utilisés dans de nombreux domaines de la recherche scientifique et ont de nombreuses applications. Ces installations complexes nécessitent des circuits électroniques avancés pour être utilisées et contrôlées. Ces circuits peuvent être endommagés s'ils sont atteints par les radiations produites. Les mêmes enjeux interviennent dans les centrales nucléaires.

3.3.2 Interactions radiation-matière

Lorsque des particules rencontrent de la matière, plusieurs processus peuvent intervenir. Ces processus dépendent du matériau, du type et de l'énergie des radiations. Les radiations peuvent être classées en deux grandes catégories : les radiations ionisantes et les radiations non ionisantes. Dans cette partie, on s'intéresse principalement aux radiations ionisantes pour lesquelles les interactions radiation-matière causées peuvent être distinguées selon la charge portée par la particule. Pour finir, les effets de ce type de radiation sur l'électronique seront abordés. La Figure 3.14 synthétise ces mécanismes.

La quantité de particules irradiant un matériau peut être définie de plusieurs manières et celles principalement utilisées dans la littérature sont définies dans le Tableau 3.1. La grandeur N définit le nombre de particules incidentes, E leur énergie, t le temps et α la section transversale à travers laquelle les particules sont comptées.

3.3.2.a Cas d'une particule chargée

Les particules chargées, comme les protons et les électrons, peuvent provoquer des interactions coulombiennes avec les atomes du matériau où elles cèdent leur énergie. Ainsi, un électron de l'atome peut recevoir une énergie supérieure à son énergie de liaison, cassant

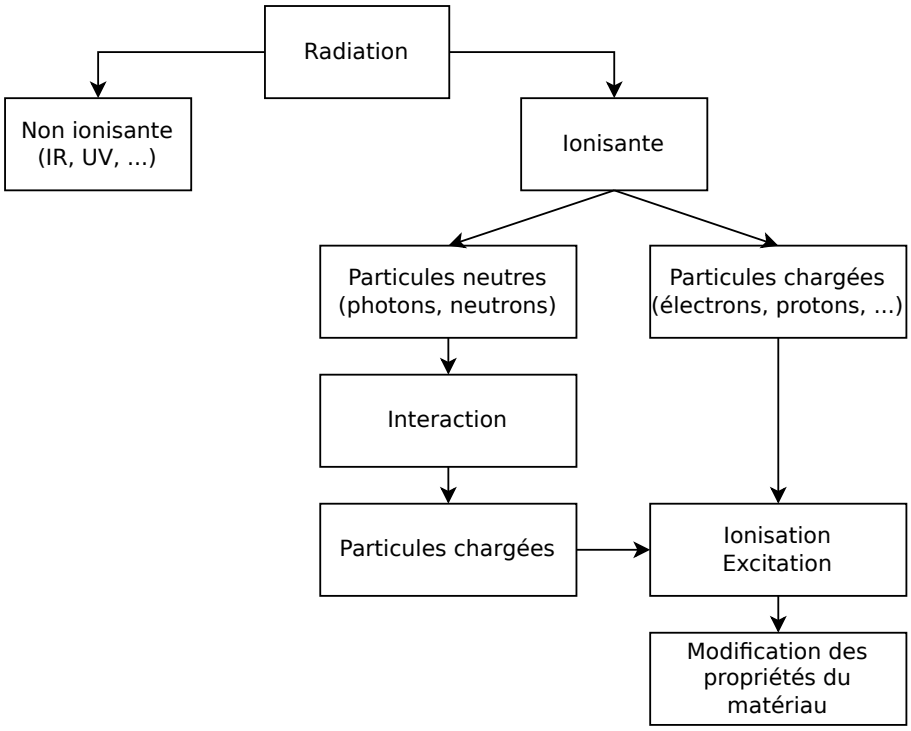


Figure 3.14 – Interaction radiation-matière.

Grandeur	Flux	Fluence	Énergie-Fluence	Taux de fluence
Symbole	$\dot{N}$	ϕ	ψ	$\dot{\phi}$
Définition	$\frac{dN}{dt}$	$\frac{dN}{d\alpha}$	$E_\phi(E)$	$\frac{d\phi}{dt}$
Unité SI	s^{-1}	m^{-2}	$J.m^{-2}$	$m^{-2}.s^{-1}$
Unité usuelle	s^{-1}	cm^{-2}	$J.cm^{-2}$	$cm^{-2}.s^{-1}$

Tableau 3.1 – Définitions de la quantité de particules irradiant un matériau. [Sel+11].

ainsi la liaison nucléaire et devenant un électron secondaire. L'atome du matériau est donc devenu un ion, c'est que l'on appelle l'*ionisation*. Il existe également deux autres processus non ionisants issus de l'interaction entre une particule chargée et un atome [Val00] :

— le *bremsstrahlung* qui produit un photon secondaire,
— le *recul atomique* qui engendre un déplacement atomique.

À l'échelle microscopique, une particule chargée se déplaçant dans la matière peut engendrer de nombreuses interactions avec une certaine probabilité. Chaque interaction peut avoir des résultats différents en termes d'énergie et de direction de la particule créée. À l'échelle macroscopique, l'ensemble de ces interactions peuvent être résumées

comme une perte d'énergie de la particule incidente par unité de longueur traversée (ou perte d'énergie linéique) dans le matériau. Cette approximation est appelée *Continuous Slowing-Down Approximation* qui implique le paramètre S appelé *Pouvoir d'arrêt*.

Une autre grandeur utilisée pour définir cette perte d'énergie est le LET pour *Linear Energy Transfer* qui exprime l'énergie transférée par une particule au matériau par unité de longueur parcourue. Le LET peut être associé à une variation d'énergie Δ afin de ne considérer que les électrons secondaires ayant une énergie $E \leq \Delta$.

Ces deux grandeurs sont définies dans la Tableau 3.2. dE est l'énergie perdue par une particule parcourant une longueur dl dans le matériau.

Grandeur	Pouvoir d'arrêt	LET
Symbole	S	L_Δ
Définition	$\frac{dE}{dl}$	$\frac{dE_\Delta}{dl}$
Unité SI	$J.m^{-1}$	

Tableau 3.2 – Grandeurs usuelles pour quantifier la perte d'énergie linéaire d'une particule chargée.

L'énergie transférée par une particule chargée peut altérer le matériau ciblé, soit par les multiples ionisations produites sur le chemin de la particule soit par des déplacements atomiques. La grandeur physique qui mesure l'énergie transférée au matériau par une particule ionisante normalisée par unité de masse du matériau est appelée la *dose absorbée*. Elle s'exprime en Gray (Gy) qui équivaut au J/kg. Une autre unité, plus ancienne, est le rad pour *radiation-absorbed dose* qui vaut 0,01 Gy et est encore utilisée dans la littérature.

La dose absorbée dans un environnement radiatif dépend grandement du matériau ciblé, c'est pourquoi il est important de spécifier dans l'unité le matériau irradié, on parle alors de Gy(matériau) [Rav18].

3.3.2.b Cas d'une particule non-chargée

Les photons qui composent des rayons X et γ ne provoquent pas directement d'ionisation de la matière mais engendrent une réaction en chaîne aboutissant à l'ionisation de la matière. Le phénomène est appelé l'*ionisation indirecte*. Il existe trois processus pouvant intervenir lors de l'interaction entre des photons et des atomes.

Le premier processus est l'*effet photoélectrique* décrit dans la sous-sous-section 3.2.1.b. Ce processus peut également s'accompagner de l'émission d'un photon fluorescent ou d'un électron d'Auger [MH19].

Le deuxième processus est la *diffusion Compton* dans laquelle le photon incident cède une partie de son énergie à un électron de l'atome. L'électron est alors libéré de l'atome et le photon change de direction [Com23a ; Com23b].

Le troisième processus est la *production de paires* dans laquelle le photon interagit avec le champ de Coulomb et une partie de son énergie pour produire une paire électron/positron. Si l'interaction a lieu dans le champ d'un électron de l'atome, cet électron est aussi libéré, on a alors production d'un triplet électron/électron/positron. Ces effets ne peuvent intervenir que pour des photons hautement énergétiques ($E > 1{,}022\,\text{MeV}$) [Rav18].

Ces trois effets dépendent du matériau ciblé et de l'énergie du photon incident. L'effet photoélectrique est prédominant pour des photons de faibles énergies ($E < 100\,\text{keV}$) et la production de paires pour des photons de hautes énergies ($E > 10\,\text{MeV}$). Entre ces bornes, c'est la diffusion Compton qui prédomine. Une illustration de la prédominance de ces processus selon l'énergie du photon incident est visible en Figure 3.15.

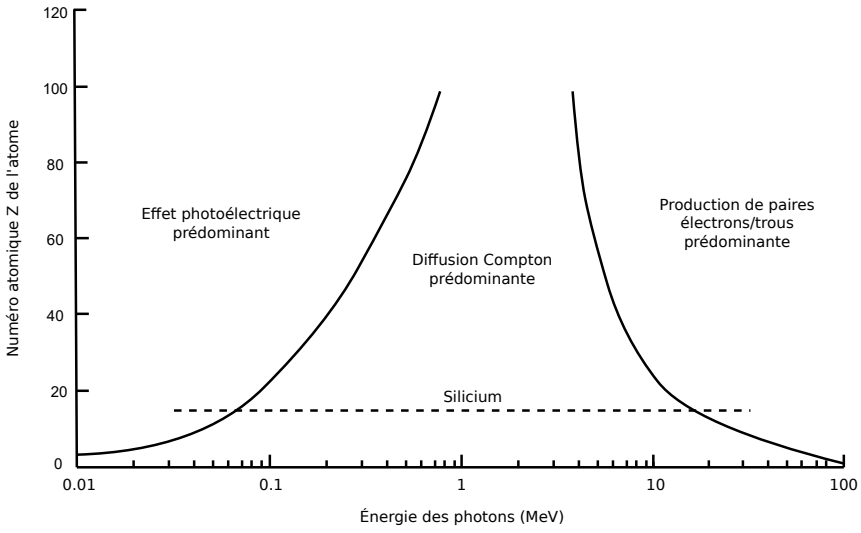
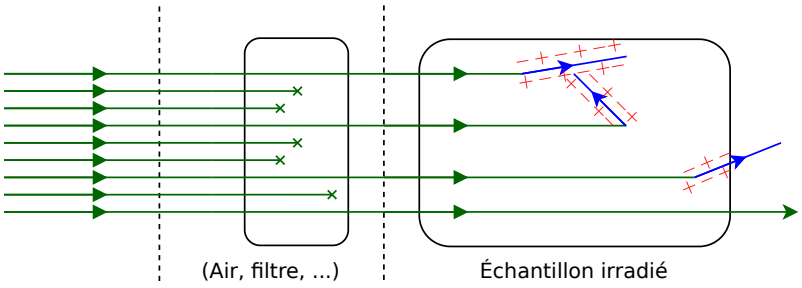


Figure 3.15 – Domaine de prédominance des différents effets selon l'énergie du photon incident. Adapté de [Sch94].

Les électrons générés par ces trois processus interagissent avec la matière comme décrit dans la partie précédente en entraînant des ionisations à partir des électrons secondaires. C'est pourquoi il est important de quantifier la génération d'électrons secondaires par des photons incidents. Cette conversion d'énergie des photons en électrons secondaires est définie par la grandeur physique *kerma*, qui exprime une énergie cinétique par unité de masse. Elle représente la somme des énergies cinétiques de toutes les particules chargées secondaires libérées par le photon incident par unité de masse. Le concept de kerma peut s'étendre à d'autres particules non chargées comme les neutrons.

L'ensemble de ces effets est résumé sur la Figure 3.16. Le nombre de photons incidents par unité de surface est représenté par la fluence Φ qui décroit en $\frac{1}{d^2}$ avec d la distance de la source. Une partie de ces photons sont absorbés pendant leur déplacement, ce phénomène est traduit par le *coefficient d'atténuation massique* $\frac{\mu}{\rho}$. Dans le matériau, les photons vont générer des électrons secondaires caractérisés par le kerma K. À ce processus est associé le *coefficient d'absorption massique d'énergie* $\frac{\mu_{en}}{\rho}$ qui est défini dans une table pour la plupart des éléments [HS04]. Pour finir, ces électrons secondaires vont causer des ionisations dans le matériau, résultant en un *dépôt de dose* D. Cette grandeur est toujours inférieure ou égale au kerma K car certains électrons peuvent quitter le matériau avant d'avoir déposé toute leur énergie.



Photon incident	Atténuation	Libération d'électrons	Dose déposée
Fluence Φ	Fluence Φ	Kerma K	Dose déposée D
$\Phi_B = \Phi_A \frac{d_A^2}{d_B^2}$	$\Phi(z) = \Phi_0 exp(-\frac{\mu}{\rho} z\rho)$	$K = \frac{\mu_{tr}}{\rho} \Phi E$	Simulation (Monte Carlo)

Figure 3.16 – Schéma de synthèse des différentes étapes avec les grandeurs physiques représentatives. [Mey23].

3.3.3 Effets des radiations sur l'électronique

Les effets des environnements radiatifs sur l'électronique, et plus spécifiquement les mémoires Flash, sont souvent classés en trois catégories : l'effet TID (Total Ionizing Dose ou *Effet de dose totale*), l'effet DDD (Displacement Damage Dose) et les effets singuliers (SEE pour Single Event Effect).

3.3.3.a Effet de dose totale

Piégeage de charges pour les transistors MOS

Le mécanisme de piégeage de charge est plus connu sous le nom Total Ionizing Dose (TID). Le processus qui engendre des défauts d'ionisation à partir d'une irradiation ionisante est constitué de plusieurs étapes :

1. la génération de paires électron/trou

2. la recombinaison rapide d'une partie des paires créées
3. le transport des porteurs libres de charges restants dans l'oxyde
4. le piégeage de charges positives à l'interface Si/SiO_2.

Ces étapes sont représentées sur la Figure 3.17. Après la génération des paires électron/trou, une partie de ces paires va se recombiner. Celles qui ne se recombinent pas sont séparées sous l'effet du champ électrique présent. Si une polarisation positive est appliquée, les électrons sont évacués par la grille en un temps très court (de l'ordre de la picoseconde) et les trous s'accumulent à l'interface Si/SiO_2 et sont piégés. Les charges positives piégées dégradent l'état de l'interface.

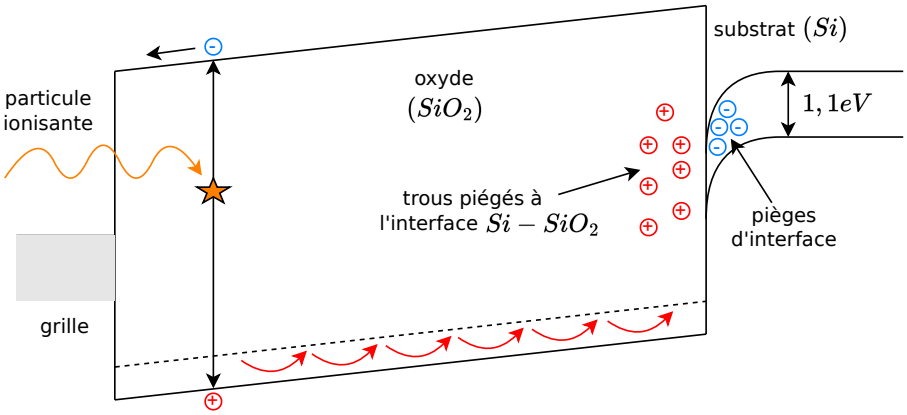


Figure 3.17 – Principales étapes de l'effet TID. Adaptée de [Bar06].

L'accumulation de cette charge positive à l'interface entre l'oxyde et le substrat cause un décalage de la courbe $I_D = f(V_{GS})$ vers les faibles tensions de grille comme illustré sur la Figure 3.18. Cela correspond à une diminution (respectivement augmentation) de la tension de seuil V_{th} pour les transistors NMOS (respectivement PMOS).

Au niveau transistor :

— les NMOS deviennent passants plus facilement, voire de façon permanente
— les PMOS deviennent bloquants plus facilement, voire de façon permanente.

Ces effets ont notamment permis aux auteurs de [Anc+17 ; Bou+23] de forcer l'état d'une cellule SRAM en ciblant certains transistors de la cellule avec une source focalisée de rayons X. Ils parviennent également à fauter des colonnes entières d'une mémoire Flash [Anc+17 ; Mai+21].

Les charges piégées peuvent être évacuées par recuit thermique. En effet, une récupération thermique est possible en plaçant le composant à environ 150 °C pendant quelques

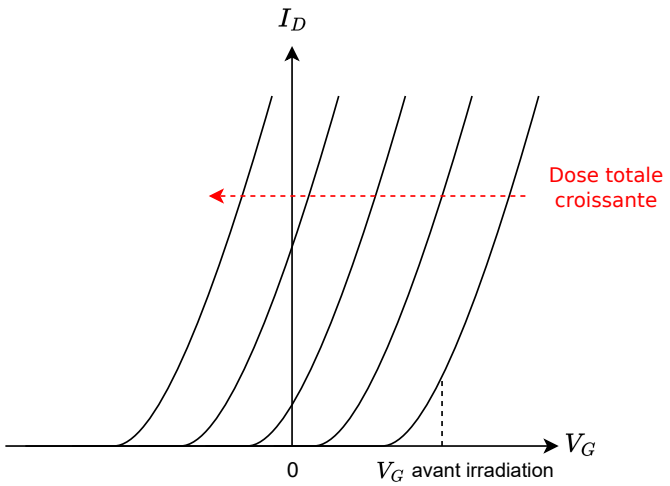


Figure 3.18 – Effet de la totale sur la caractéristique $I_D = f(V_{GS})$ d'un transistor NMOS. Adaptée de [Sha02].

heures. Dans cette configuration, les charges piégées obtiennent assez d'énergie pour quitter les pièges où elles sont stockées et le circuit récupère son fonctionnement nominal.

Transistor à grille flottante

Trois processus différents ont été identifiés dans la littérature comme étant responsables des changements de tension de seuil des transistors à grille flottante induits par l'effet TID. Ils sont représentés sur la Figure 3.19 et détaillés ci-dessous :

(1) les paires électron/trou créées par les radiations dans les oxydes sont séparées par le champ électrique. L'une des charges peut s'échapper par la grille de contrôle alors que l'autre est injectée dans la grille flottante. Cette dernière pourra alors se recombiner avec les charges stockées dans la grille flottante. La charge globale stockée sera donc diminuée.

(2) les charges peuvent être piégées dans les oxydes.

(3) les charges stockées dans la grille flottante obtiennent assez d'énergie des radiations pour s'échapper du puits de potentiel. Ce phénomène est la *photoémission*.

Les mécanismes (1) et (3) contribuent à la diminution du nombre de charges stockées dans la grille flottante.

Le mécanisme (1) injecte des trous (respectivement des électrons) depuis les oxydes dans la grille flottante où sont stockés des électrons (respectivement des trous). Le premier

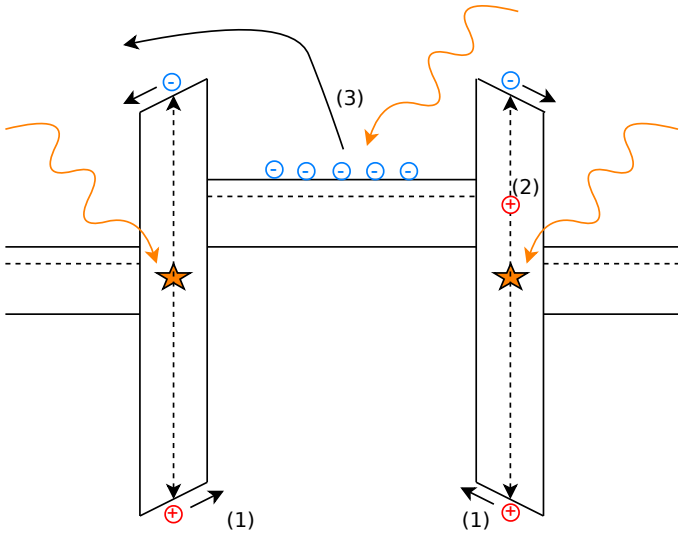


Figure 3.19 – Mécanismes TID dans un transistor à grille flottante. Adaptée de [Ger+13].

mécanisme nécessite la présence d'un champ dans l'oxyde afin de séparer les charges électriques. Il y a alors recombinaison d'une partie des paires électron/trou. Les charges restantes s'accumulent dans l'oxyde ou à proximité des interfaces $Si - SiO_2$. Les charges piégées engendrent un décalage de la tension de seuil V_{th} et une augmentation du courant de fuite du transistor.

Le mécanisme (2) est peu significatif car les oxydes ont des épaisseurs très faibles.

Le mécanisme (3) permet aux charges stockées (indépendamment desdites charges) de franchir la barrière de potentiel créée par les oxydes et de s'évacuer. Il est observé dans les travaux de Anceau *et al.* [Anc+17 ; Mai+21 ; Bou+23] dans lesquels ils parviennent à effacer le contenu de transistors à grille flottante d'une mémoire Flash avec des sources de rayons X. Ils obtiennent des résultats similaires sur une mémoire EEPROM.

Pour les transistors à grille flottante, une faute intervient lorsqu'un décalage suffisamment important de la tension de seuil V_{th}. Les charges stockées dans les grilles flottantes peuvent être évacuées par l'effet des radiations et réduire la tension de seuil du point mémoire. Une faute apparaît si la tension de seuil est suffisamment réduite pour devenir inférieure à une valeur de référence V_{read}. La Figure 3.20 illustre la distribution des tensions de seuil des transistors à grille flottante d'une mémoire Flash avant et après irradiation [Ger+13 ; BGP17].

3.3.3.b Displacement Damage

L'effet *Displacement Damage* résulte d'interactions coulombiennes ou nucléaires avec les noyaux du matériau. Il intervient lorsqu'une particule incidente est assez énergétique pour

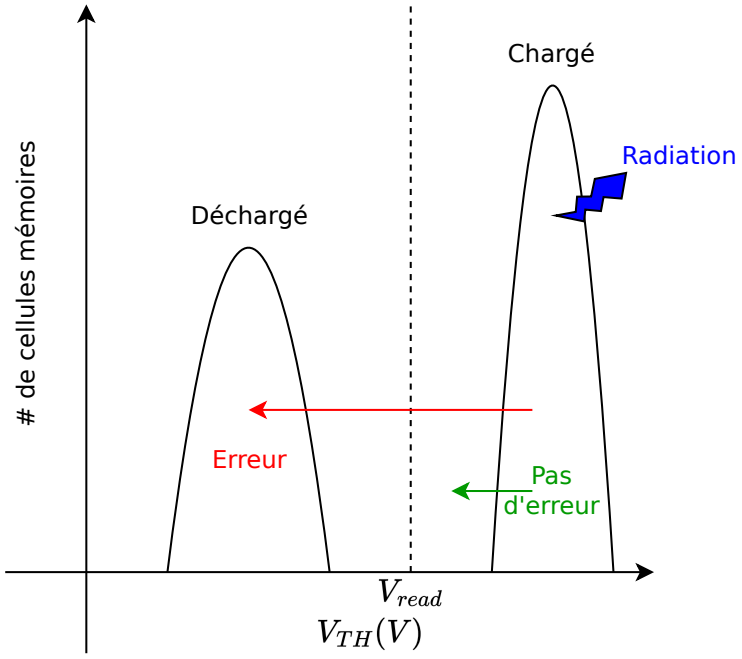


Figure 3.20 – Impact des radiations sur la distribution des tensions de seuil des transistors à grille flottante d'une mémoire Flash. Adaptée de [Ger+13].

déplacer un atome de la maille cristallographique, créant ainsi un défaut. Ces défauts sont l'absence de l'atome à sa position initiale et la présence de cet atome à sa nouvelle position [SP13 ; Mol18].

Cet effet a pour conséquence de faire apparaître des niveaux d'énergie dans la bande interdite du semi-conducteur. Une accumulation de ces apparitions dégrade les comportements électroniques et optiques des semi-conducteurs. D'un point de vue électrique, on constate notamment une augmentation des courants de fuite des transistors.

3.3.3.c Effets singuliers

Des SEE sont également possibles lors de la présence d'un composant électronique dans un environnement radiatif. À l'inverse d'un effet de dose totale, ils sont causés par une unique particule énergétique ionisant le matériau lors de son passage, ce n'est pas un effet cumulatif. L'utilisation du laser comme moyen de perturbation d'un composant électronique a été introduite dans un premier temps pour émuler les SEE radiatifs [Hab65].

Single Event Transient

Les paires électron/trou créées par les radiations peuvent générer un pic de tension ou de courant sur un nœud sensible du circuit et provoquer un comportement imprévu. Ce phénomène est notamment présent dans les circuits analogiques et la partie combinatoire

des circuits numériques. Si le pic est assez long pour être échantillonné, il peut se propager dans la logique du circuit. Les circuits fonctionnant à haute fréquence sont donc plus sensibles [BG16].

Single Event Upset

Un SEU apparaît lorsqu'un SET intervient et que le signal impacté est l'entrée d'un point mémoire. Par exemple, les mémoires DRAM (Dynamic Random Access Memory) stockent les données en chargeant des capacités. Si un SET intervient à proximité de la capacité ou des transistors d'accès, un chargement de la capacité est alors possible et l'information stockée est altérée [MW79 ; DM03 ; Bou+11].

Single Event Functional Interrupt

Lors de l'impact d'une particule, il est possible que le composant électronique perde sa fonctionnalité. Cela peut être dû à un changement d'une valeur de registre qui fait entrer le composant dans un état imprévu ou indéfini. Il existe certaines positions précises au sein du composant où ce genre d'erreur peuvent intervenir. Ces effets dépendent grandement de la technologie du composant et de sa conception [Gai11].

Single Event Latch-up

Le passage d'une particule ionisante et la phase de collection de charges qui en découle peut activer une structure bipolaire parasite dans le substrat des transistors CMOS. Cette structure est un thyristor normalement présent mais non actif en fonctionnement nominal. La Figure 3.21 représente cette structure sur un inverseur CMOS classique. En condition normale, la structure parasite est en haute impédance. Une particule ionisante peut forcer l'un des transistors dans un état passant, un courant important apparaît alors entre l'alimentation V_{dd} et la masse Gnd [Gai11 ; BP96]. Si un cycle d'alimentation n'est pas réalisé rapidement, des dommages permanents peuvent intervenir par effet thermique [Puc+06]. La technologie SOI (Silicon On Insulator) permet d'éliminer ce thyristor parasite [Sch+03]. Le phénomène de *Latch-up* peut également intervenir lors d'injections laser mais il est très rare en pratique.

3.4 Effet du vieillissement

Des effets intrinsèques aux circuits électroniques peuvent également dégrader le fonctionnement de ces derniers au cours du temps. Ces effets apparaissent progressivement au cours de la vie des composants électroniques. C'est pourquoi ils sont regroupés sous

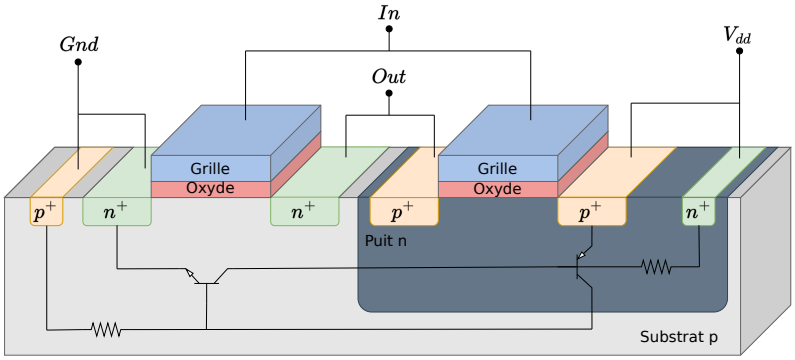


Figure 3.21 – Vue en coupe d'un inverseur CMOS avec la structure *pnpn* parasite.

l'appellation *mécanismes de vieillissement*. Ils peuvent affecter des propriétés des transistors comme la tension de seuil ou le courant de fuite. Il est possible d'accélérer ces mécanismes de vieillissement en laboratoire afin d'altérer le fonctionnement du composant. Avec une maitrise de ce vieillissement, il est possible d'en exploiter les effets à des fins d'attaque. Les quatre principaux mécanismes de vieillissement qui sont décrits dans cette section sont :

— l'instabilité de la température de polarisation,
— l'injection de porteurs chauds,
— le *Time-Dependant-Dielectric Breakdown*,
— l'électromigration.

3.4.1 Instabilité de la température de polarisation

L'effet Bias Temperature Instability (BTI) est une dérive des paramètres électriques des transistors MOS polarisés dans un milieu à haute température. Cette dégradation intervient en l'absence de courant de porteurs dans le canal du transistor, ce qui en fait une dégradation statique. Ce phénomène intervient de façon plus marquée pour les transistors PMOS (effet Negative-BTI (NBTI)) que pour les transistors NMOS (effet Positive-BTI (PBTI))[Sch07 ; Gra+14].

La contrainte NBTI se résume en l'application d'une tension de grille V_{GS} négative dans un milieu à haute température [HDP06]. Un exemple de configuration électrique causant cette dégradation est visible en Figure 3.22. La contrainte PBTI, quant à elle, consiste en l'application d'une tension de grille V_{GS} positive dans un milieu à haute température.

Plusieurs modèles ont été proposés pour expliquer l'effet NBTI, néanmoins, c'est le modèle de *Réaction-Diffusion* qui domine. Dans ce modèle, des pièges d'interface sont

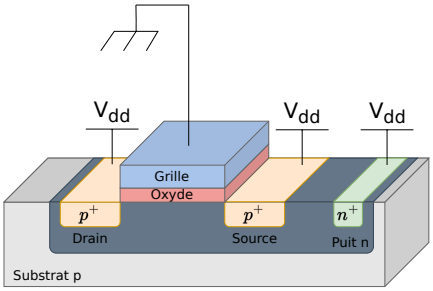
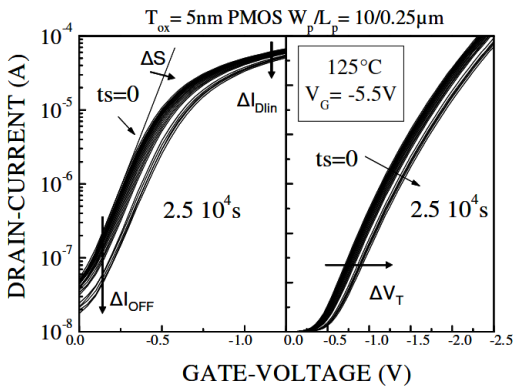
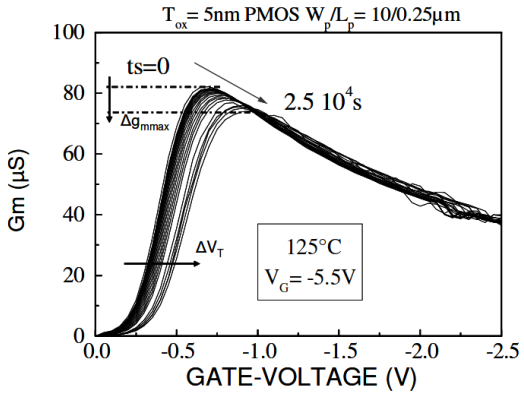


Figure 3.22 – Configuration électrique d'une contrainte NBTI.

générés à la frontière SiO_2/Si selon une dépendance linéaire avec le temps de stress. Pendant la phase de *Réaction* des charges sont générées à l'interface et de l'hydrogène est libéré. Pendant la phase de *Diffusion* l'hydrogène se diffuse dans l'oxyde. D'autres défauts sont générés dans le diélectrique. Les charges d'interface sont des défauts électriquement actifs avec une distribution d'énergie dans toute la bande interdite du silicium. Ils jouent le rôle de centre de génération/recombinaison et contribuent à l'apparition d'un courant de fuite, à la réduction de la mobilité des porteurs, du courant de drain et de la transconductance. De plus, les porteurs de charges, occupant les pièges à l'interface, génèrent un potentiel de surface qui va contribuer à un décalage de la tension de seuil du transistor. La Figure 3.23a représente l'évolution de la caractéristique I_d-V_g d'un transistor PMOS pendant une contrainte NBTI. On peut observer une augmentation de la tension de seuil V_t et une diminution du courant linéaire du transistor. La Figure 3.23b représente l'évolution de la transconductance d'un PMOS en fonction de la tension appliquée sur la grille. On peut remarquer une diminution de la transconductance maximale $g_{m,max}$ ce qui traduit une diminution de la mobilité des trous à l'interface S_i/S_iO_2.



(a) Caractéristique $I_d - V_g$.



(b) Évolution de la transconductance.

Figure 3.23 – Dérive des paramètres électriques d'un transistor PMOS après une contrainte NBTI. [Den05].

La rupture des liaisons $Si-H$ à la frontière entre le substrat et l'oxyde génèrent des états d'interfaces (liaisons pendantes) et des charges positives dans l'oxyde. Ces deux effets

constituent la composante permanente de la dégradation NBTI. La partie recouvrable est associée à un mécanisme de piégeage (pendant la contrainte) et de dépiégeage (pendant la relaxation) de charges dans l'oxyde. Les dépendances temporelles des parties recouvrable et permanente sont illustrées sur la Figure 3.24.

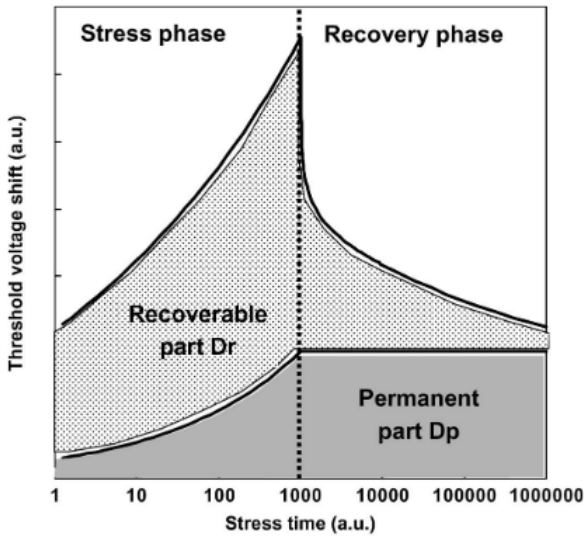


Figure 3.24 – Illustration des effets permanents et recouvrables de la contrainte NBTI sur la tension de seuil. [Hua+07].

3.4.2 Injection de porteurs chauds

L'injection de porteurs chauds (Hot Carrier Injection (HCI)) est une dégradation locale des paramètres électriques d'un transistor soumis à un champ électrique important. À l'inverse de l'effet BTI, ce phénomène est plus présent dans les transistors NMOS que PMOS. Les électrons peuvent atteindre une vitesse importante dans la zone de charge d'espace à proximité du drain à cause du champ électrique présent entre la source et le drain. Les impacts de ces derniers avec la matière génèrent des paires électron-trou. Les électrons peuvent être injectés dans l'oxyde de grille grâce au champ électrique vertical. Les trous sont collectés dans le courant de substrat. Cet effet a pour impact une augmentation de la tension de seuil, une réduction du courant linéaire et de saturation du transistor et une diminution de la transconductance. La Figure 3.25 schématise ce phénomène. Cet effet est utilisé pour programmer les transistors à grille flottante des mémoires Flash.

3.4.3 Time-Dependant-Dielectric Breakdown

L'effet Time-Dependant-Dielectric Breakdown (TDDB) est la rupture d'un diélectrique au sein d'un composant exposé à un stress électrique. Il peut intervenir entre deux

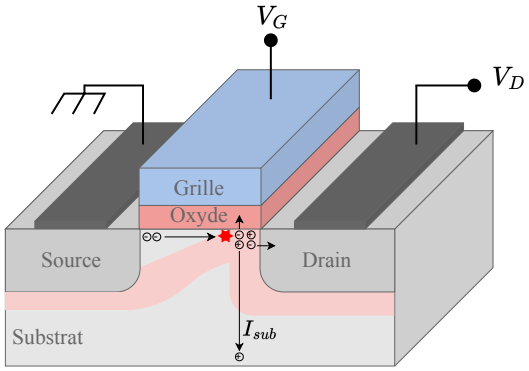


Figure 3.25 – Injection d'électrons chauds dans un NMOS.

niveaux de métallisation, entre une grille de contrôle d'un transistor et un contact ou encore entre la grille de contrôle et la source ou le drain du transistor comme illustré dans la Figure 3.26 [Lop20].

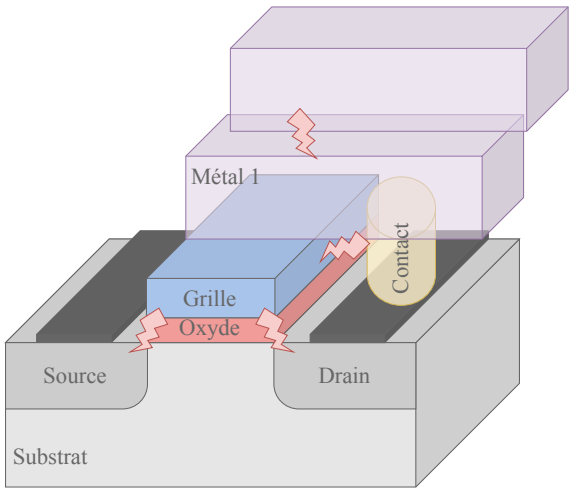


Figure 3.26 – Différents types de TDDB au sein d'un composant.

Lors d'un stress, la présence de champs électriques importants dans les diélectriques peut générer des pièges qui forment un chemin de conduction s'ils sont assez proches les uns des autres. Si la densité de ces pièges est assez importante, un claquage du diélectrique peut intervenir.

3.4.4 Électromigration

L'électromigration est un phénomène de déplacement de matière dans un conducteur induit par la présence d'un fort courant électrique dans ce dernier. Dans les circuits intégrés, l'électromigration n'intervient pas dans les semi-conducteurs mais dans les pistes métalliques. Ce phénomène peut générer des courts-circuits ou des coupes-circuits [HK89].

3.5 Conclusion

L'état de l'art ainsi réalisé nous permet d'obtenir les clés essentielles à la compréhension des travaux décrits dans ce manuscrit. Les effets du laser, thermique et photoélectrique, servent de socle aux contributions décrites dans le Chapitre 4 et Chapitre 5. Les effets des radiations permettent de comprendre les interactions physiques utilisées dans les travaux du Chapitre 6.

3.5.1 Objectifs de ces travaux

Dans un premier temps, l'objectif de ces travaux est d'évaluer les nouvelles possibilités offertes par l'exploitation d'un banc laser multispot. Ces travaux se basent sur les résultats obtenus lors d'attaques précédemment réalisées avec des bancs laser monospot et bispot.

Dans un second temps, les travaux décrits dans ce manuscrit évaluent la possibilité de réaliser des injections de fautes sur des circuits non alimentés. Dans ce contexte, des sources laser et rayons X seront utilisés. Dans les deux cas, une caractérisation des phénomènes en jeu est réalisée.

3.5.2 Contributions

Ces travaux ont fait l'objet de plusieurs publications internationales dans des revues avec comité de relecture. Ces publications regroupent notamment :

- la description d'un banc laser multispot et l'étude des nouvelles possibilités offertes par ce banc [Col+22],
- l'utilisation d'une source de rayons X pour injecter des fautes dans des mémoires non-volatiles de microcontrôleur 32 bits non alimentés [GBD23],
- l'utilisation d'une source laser pour injecter des fautes localisées dans une mémoire Flash d'un microcontrôleur 32 bits non alimenté [Gra+24].

Ces travaux sont détaillés dans les chapitres suivants.

Chapitre 4

Injection laser de fautes avec un banc laser multispot

Table des matières

4.1 Introduction

L'utilisation d'une source laser pour corrompre la lecture des données en mémoires Flash est une technique d'attaque couramment utilisée. Le modèle de faute, du niveau physique au niveau logiciel, est bien maîtrisé. Dans l'industrie comme dans le monde académique, la majeure partie des attaques réalisées est effectuée avec des bancs laser monospot. Ces derniers possèdent certaines limites, notamment temporelle et spatiale, dont un attaquant peut s'affranchir en utilisant un banc laser multispot. Dans ce chapitre, ces limites sont abordées et un nouveau dispositif d'injection laser de fautes à quatre spots est décrit.

Ce chapitre a fait l'objet d'une publication à la conférence internationale CARDIS en 2022 [Col+22].

4.2 Généralités

Dans ce chapitre, on se place dans un contexte d'injection laser de fautes ciblant des données stockées en mémoire Flash. Si ces données sont des instructions, l'injection d'une faute peut aboutir à l'exécution par le processeur d'instructions corrompues.

Le modèle de faute associé à la corruption de données possède trois caractéristiques. La première est la *direction* de la faute pour laquelle il existe trois possibilités :

— *bitset :* passage de `'0'` à `'1'`,
— *bitreset :* passage de `'1'` à `'0'`,
— *bitflip :* inversion de la donnée indépendamment de sa valeur initiale.

Dans le cas d'un *bitset* (respectivement *bitreset*), si la donnée est déjà à `'1'` (respectivement `'0'`) il n'y a pas de faute injectée en pratique. Dans ce cas, le modèle de faute est dit *data-dependent*.

La deuxième caractéristique est la *cardinalité*, *c.-à-d.* le nombre de bits impactés par la faute. À titre d'exemple, on définit les trois suivants :

— *single-bit :* un seul bit est fauté,
— *multi-bit :* plusieurs bits sont fautés,
— *octet :* 8 bits sont fautés.

La dernière caractéristique est la *répétabilité* de la faute c'est-à-dire la probabilité que la faute intervienne pour un ensemble de paramètres d'injection de fautes donné.

L'ensemble de ces trois paramètres ne permet pas de prendre en compte deux caractéristiques d'une faute : la *contiguïté* de la faute et l'aspect temporel du modèle de faute. Ces deux notions sont définies plus précisément ci-après.

4.3 Limites d'un banc laser monospot

4.3.1 Limite spatiale

Dans le cas d'injection de fautes multiples, le modèle existant ne prend en pas compte la *contiguïté* des fautes.

La création de charges électriques dans le silicium lors d'une exposition laser suit une distribution gaussienne radiale. La dispersion de cette distribution dépend de l'énergie du laser et est souvent caractérisé par la grandeur Full-Width at Half Maximum (FWHM). Si la puissance du laser est assez importante, la surface où la génération de charges intervient peut être assez grande pour recouvrir plusieurs transistors et ainsi induire de multiples fautes. La taille du spot laser peut également influer en ce sens.

Certains équipements optiques permettant de diviser un rayon lumineux en plusieurs, comme les Diffractive Optical Element (DOE), les Spatial Light Modulator (SLM) ou les Digital Micromirror Device (DMD). Ces derniers pourraient permettent de cibler différentes zones avec une seule source laser. En revanche, ils sont coûteux et complexes à mettre en place. De plus, les spots obtenus ne sont pas totalement indépendants, ni spatiallement ni temporellement, les uns des autres et leur forme finale est très peu maîtrisable. De fait, dans ces configurations, les différents spots laser obtenus ont la même durée et le même instant d'injection. Cette solution n'est donc pas viable pour réaliser des injections de fautes non contiguës.

La limite spatiale des bancs laser monospot est illustrée sur la Figure 4.1. Cette figure illustre l'injection de fautes de type *bitset* dans un mot de 32 bits initialisé à `'0'`.

Une autre solution pour réaliser des injections de fautes non contiguës est d'exploiter la non-linéarité du *layout* de la cible. En effet, deux éléments disposés côte à côte sur la cible ne sont pas nécessairement liés à des données côte-à-côte en mémoire. Ce scénario dépend de la cible et n'est donc pas généralisable.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Faute multi-bit contiguë : réalisable avec un seul spot.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

(b) Faute multi-bit non contiguë : irréalisable avec un seul spot mais réalisable avec plusieurs spots.

Figure 4.1 – Faisabilité de fautes multi-bits contiguë et non contiguë avec un seul spot laser.

4.3.2 Limite temporelle

Un autre point qui n'est pas pris en compte par le modèle de faute existant est la possibilité d'injecter des fautes à des positions différentes à des instants proches dans le temps. En effet, avec un banc laser monospot, il est nécessaire d'éteindre la source laser après la première injection laser, de déplacer mécaniquement la cible ou le laser puis d'effectuer la seconde injection laser. Il est nécessaire d'éteindre la source car pendant le déplacement la cible poursuit l'exécution de son code. Cette méthode ne peut pas être mise en pratique sur des cibles ayant des fréquences d'horloge allant du mégahertz (MHz) au gigahertz (GHz).

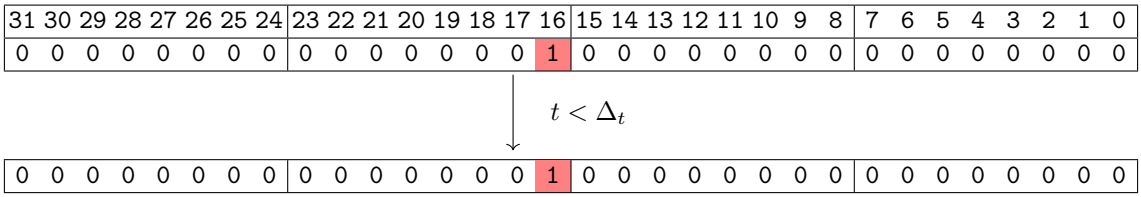
Si le délai entre deux fautes prévues est trop court, il est ainsi impossible de les réaliser avec un seul spot laser. Soit Δ_t l'intervalle de temps entre deux fautes et v_{max} la vitesse maximale de déplacement linéaire du laser ou de la cible et d_{pos} la distance entre les deux positions visées sur la cible. On obtient alors l'Équation 4.1 pour que les injections de ces deux fautes soient possibles.

$$\Delta_t > \frac{d_{pos}}{v_{max}} \tag{4.1}$$

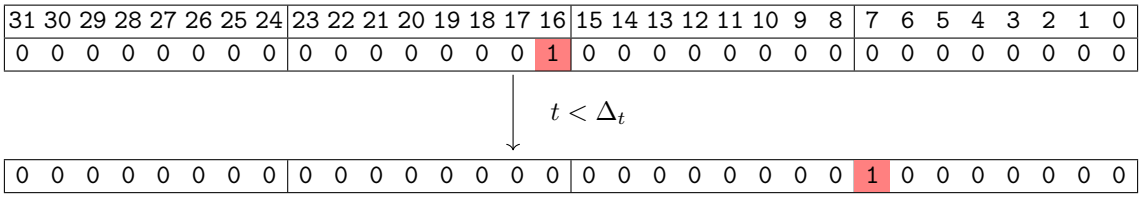
Par simplification, on suppose que le système mécanique se déplace à vitesse constante. En réalité, les phases d'accélération et de décélération ont des profils sinusoïdaux afin d'éviter un désalignement lors des changements brusques de vitesse. On suppose une vitesse de déplacement linéaire de $20\,\mathrm{mm\,s^{-1}}$ et deux positions distances de 10 % du côté d'une cible de 2 mm de côté. Ainsi, l'intervalle de temps minimum $\Delta_{t_{min}}$ entre ces deux fautes est donné par l'Équation 4.2

$$\Delta_{t_{min}} = \frac{d_{pos}}{v_{max}} = \frac{2 \times \frac{10}{100}}{20} = 0.01s \tag{4.2}$$

Soit un composant électronique fonctionnant à une fréquence relativement faible de 10 MHz, soit une période d'horloge de 100 ns, $\Delta_{t_{min}}$ équivaut à 10^5 périodes d'horloge. Cela représente une contrainte extrêmement forte sur l'intervalle de temps compris entre deux instructions d'un programme qu'un attaquant souhaiterait fauter. La Figure 4.2 illustre cette limitation.



(a) Fautes consécutives sur un même bit : réalisable avec un seul spot.



(b) Fautes consécutives sur des bits différents : irréalisable avec un seul spot mais réalisable avec plusieurs spots.

Figure 4.2 – Faisabilité de fautes consécutives avec un seul spot ou plusieurs spots laser sur un mot de 32 bits.

Par ailleurs, le déplacement du système mécanique entre deux positions rend difficile la synchronisation entre les fautes. En effet, alors que les tirs lasers sont très précis et déclenchés par un signal électrique généré par la cible puis transmis à la source laser, le système mécanique n'est généralement pas conçu pour être synchronisé de façon précise, ce qui ajoute une grande incertitude sur le délai entre le positionnement et le tir laser. Pour finir, il est nécessaire d'avoir deux signaux de déclenchement pour synchroniser les deux fautes entre elles, ajoutant une nouvelle contrainte aux scénarios d'attaques.

La section suivante décrit un banc laser multispot qui permet à un attaquant de s'affranchir de ces contraintes. Il est ainsi possible de réaliser de multiple injection de fautes arbitrairement proches dans le temps sans avoir plusieurs signaux de déclenchement ou des injections simultanées mais avec des durées différentes.

4.4 Présentation du dispositif expérimental

4.4.1 Banc laser ALPhANOV

Le nouveau banc laser de la société `ALPhANOV` [ALP19] est présenté en Figure 4.3. Il comprend quatre sources laser monomodes, deux d'une longueur d'onde de 980 nm

et deux d'une longueur d'onde de 1 064 nm. Les sources laser monomodes permettent d'avoir des spots plus petits que les sources multimodes et ainsi de cibler de plus petits éléments sur le composant. Les faisceaux des couples de sources laser de même longueur d'onde sont linéairement polarisés mais perpendiculaires. Ils sont combinés par les cubes séparateurs de faisceaux de polarisation (CP) qui sont réfléchissants pour une direction de polarisation mais transparents pour l'autre direction. Le miroir dichroïque (MD) permet de combiner des sources laser de longueur d'onde différente, ici réfléchissant pour la longueur d'onde 980 nm et transparent pour la longueur d'onde 1 064 nm. Les quatre faisceaux lumineux sont focalisés sur la cible par une unique lentille de focalisation (LF). Cette lentille est partagée avec la caméra infrarouge, qui est confocale. Cela permet de faire la mise au point grâce à la caméra infrarouge. Plusieurs lentilles de focalisation sont disponibles sur ce montage : $\times 2.5$, $\times 20$, $\times 50$.

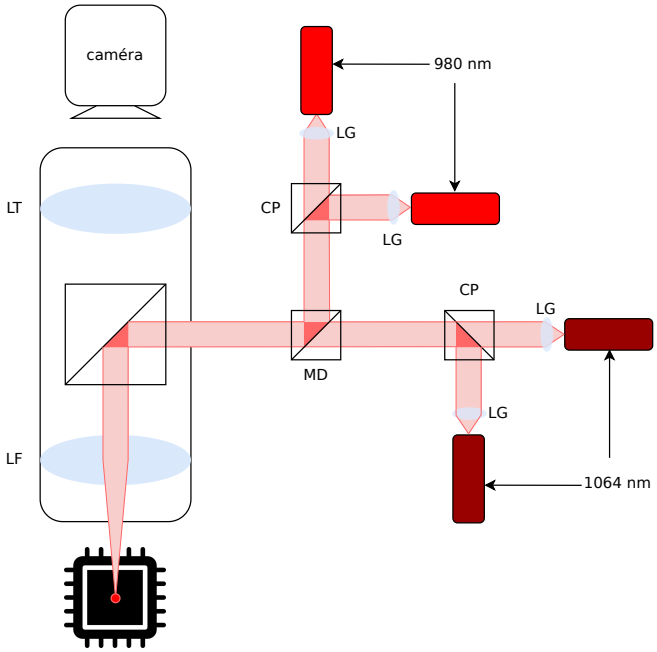


Figure 4.3 – Schéma du banc laser à quatre spots. (CP : Cube séparateur de faisceau de Polarisation, MD : Mirroir Dichroïque, LF : Lentille de Focalisation, LG : Lentille Grossissante, LT : Lentille Tubulaire).

4.4.1.a Avantages

Chacune des quatre sources laser peut être contrôlée indépendamment en temps, durée et puissance car chacune des sources possède sa propre électronique de commande et spatialement afin d'être déplacée dans le plan focal de la lentille de focalisation afin d'obtenir quatre spots distincts. L'ingénierie permettant de déplacer les spots laser n'est pas connue. Cela permet de placer les quatre spots à des positions distinctes et également de déclencher les tirs lasers indépendamment. Ainsi, les fautes injectées peuvent être aussi proche dans le temps que requis par l'application ciblée. Il est également possible

de synchroniser les sources entres elles (signal de *trigger* partagé entre plusieurs sources) afin de fauter simultanément plusieurs positions différentes.

4.4.1.b Limites

Le montage à quatre spots décrit n'est pas équivalent à quatre montages monospots. En effet, les quatre faisceaux lumineux traversent la même lentille de focalisation (LF sur la Figure 4.3), ce qui impose une distance maximale entre les spots sur la cible. Cette distance dépend du grossissement de la lentille de focalisation et modifie la taille minimale atteignable du spot laser. Ces caractéristiques sont présentées dans le Tableau 4.1.

Grossissement	Champ visuel	Diamètre minimal du spot
$\times 2.5$	4 mm	25 µm
$\times 20$	500 µm	2,2 µm
$\times 50$	200 µm	1,3 µm

Tableau 4.1 – Champ visuel et diamètre minimal du spot pour les différents grossissements disponibles.

Ainsi, avec un grossissement de $\times 20$, la distance maximale entre les spots est de 500 µm. Si la distance entre deux points cibles est supérieure à cette dernière valeur, il n'est pas possible de les atteindre en même temps et il est nécessaire de déplacer la cible, ce qui n'est pas réaliste dans un scénario d'attaque comme décrit dans la sous-section 4.3.2.

Un autre point à prendre en compte, lié aux positions des spots, est que lorsqu'ils s'éloignent du centre du champ visuel, ils perdent progressivement de leur puissance comme illustré sur la Figure 4.4. Pour les grossissements $\times 2.5$ et $\times 20$ l'effet n'est quasiment pas visible. En revanche, cet effet de bord est conséquent pour le grossissement $\times 50$. En effet, dans cette configuration, un spot placé sur le bord du champ visuel n'a qu'une infime puissance optique. Ce n'est pas un problème pour les expériences décrites dans la section 4.5 car c'est le grossissement $\times 20$ qui est utilisé.

4.4.2 Cible

Pour des applications nécessitant un certain niveau de sécurité, il est nécessaire d'utiliser des circuits dits "sécurisés". La sécurité de ces derniers est assurée par leur conformité à des standards de sécurité définis par des organismes gouvernementaux et évaluée par des instances étatiques ou indépendantes. Les cartes à puces (carte bancaire par exemple) doivent être certifiés Critères Communs (CC) [ANS22] pour être commercialisées. Dans le cadre de l'IoT, la majeure partie des circuits ne sont pas sécurisés et ne sont donc pas protégés contre les attaques par injection de fautes.

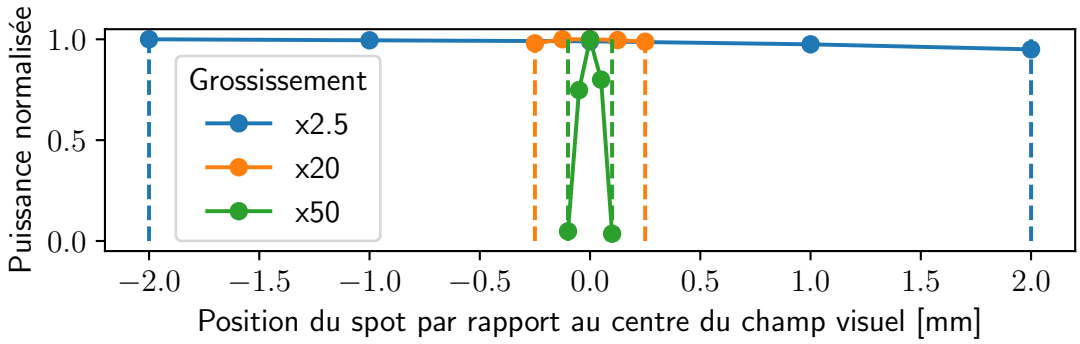


Figure 4.4 – Puissance relative du laser en fonction de la distance entre le spot et le centre de l'objectif.

Les expériences décrites dans ce manuscrit ont été réalisées sur des microcontrôleurs `STM32F100RBT6B` fabriqué par la société franco-italienne `STMicroelectronics`.

4.4.2.a Préparation

Les injections laser peuvent être réalisées soit par la face avant car les sources ont une longueur d'onde dans le proche infrarouge soit par la face arrière du microcontrôleur. En revanche, les nombreuses pistes métalliques réparties sur les différents niveaux de métallisation rendent difficile l'accès aux zones actives des transistors par le faisceau lumineux via la face avant. De plus, le faisceau lumineux peut être réfléchi par les métallisation. C'est pourquoi c'est un accès par la face arrière qui est choisi.

Indépendamment de la face choisie, plusieurs méthodes existent afin de réaliser la décapsulation, c'est-à-dire l'ouverture, des composants [Lim+22]. Elles sont réalisées par la plateforme `MicroPackS` [MP19].

Décapsulation chimique : Un mélange chimique est utilisé pour retirer la résine du composant. Ce mélange est composé d'acide nitrique (HNO_3), d'acide sulfurique (H_2SO_4) ou d'un mélange des deux selon la composition de la résine et des fils de *bonding* comme l'illustre le Tableau 4.2. Les paramètres dovient être judicieusement choisis afin de ne pas détériorer le *bonding*, ce qui rendrait le circuit inutilisable.

Matériau des fils de *bonding*	Or	Cuivre
Solution	$100\%HNO_3$	$\frac{5}{6}HNO_3 + \frac{1}{6}H_2SO_4$
Température	80 °C	44 °C

Tableau 4.2 – Solutions acides utilisées pour la décapsulation. Adapté de [Lim+22].

Décapsulation mécanique : Une fraiseuse peut également être utilisée afin de retirer la résine du composant. Il est nécessaire d'être prudent lorsque la décapsulation est effectuée sur la face avant car le circuit peut être détruit si les pistes métalliques sont atteintes. Ce procédé est souvent complété par une décapsulation chimique. Lors d'un fraisage par la face arrière, l'attaquant peut aller jusqu'au silicium. Dans ce cas, il est nécessaire de polir la surface obtenue afin de permettre l'observation et les attaques par la face arrière.

Décapsulation laser : Une source laser peut être utilisée afin de fondre et d'enlever la résine. Le circuit peut être endommagé s'il est atteint par le laser, c'est pourquoi cette méthode est utilisée pour enlever la majeure partie de la résine et cette étape est achevée par une décapsulation chimique. Ce procédé est plus rapide et permet une plus grande précision que la décapsulation mécanique.

Ainsi, une ouverture en face avant requiert principalement d'enlever la résine alors qu'une ouverture en face arrière nécessite d'enlever la résine ainsi que la couche de cuivre servant de plan de masse et de dissipateur thermique et de polir la surface obtenue.

4.4.2.b Caractéristiques

La cible matérielle est un microcontrôleur 32 bits (`STM32F100RBT6B`) fabriqué en technologie CMOS 80 nm. Il est monté sur une carte conçue pour la plateforme d'évaluation `ChipWhisperer` [OC14]. Cette carte est spécifiquement conçue avec une ouverture dédiée afin de permettre l'accès à la face arrière du composant. La cible possède un cœur `ARM Cortex-M3`, une mémoire Flash de 128 kB et fonctionne avec une horloge interne à 7,4 MHz. Le cœur utilise le jeu d'instruction `ARM Thumb-2`. La carte de test comportant la cible et l'image infrarouge de la cible sont visibles en Figure 4.5.

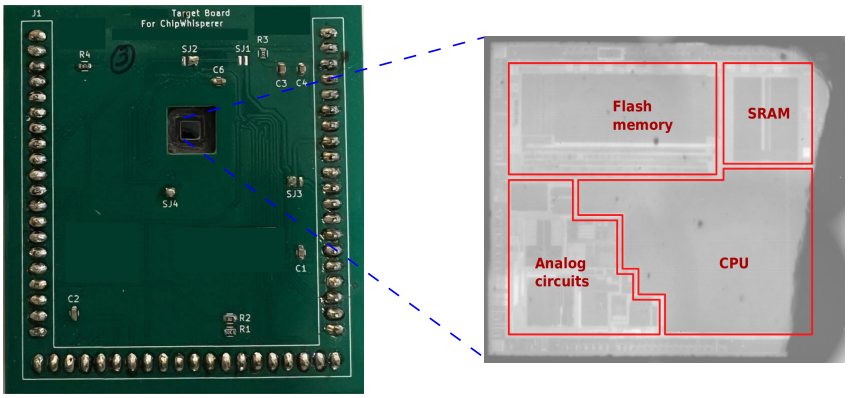


Figure 4.5 – Carte de test (gauche) et image infrarouge (droite) de la cible.

4.4.2.c Description du jeu d'instructions

Pour comprendre les attaques réalisées dans ce chapitre, il est nécessaire d'étudier la représentation binaire des instructions exécutées par le microcontrôleur. Le jeu d'instruction `ARM Thumb-2` utilise des instructions définies sur 16 bits. Les instructions contiennent un ou plusieurs opcodes et des données. L'opcode principal est défini sur 6 bits et représente l'opération à réaliser, ce qui permet au composant de traiter $2^6 = 64$ instructions différentes.

On s'intéresse notamment à deux configurations spécifiques : des instructions dites "basiques" et des instructions de traitement de données. Ces instructions seront les cibles de nos attaques dans les expériences décrites dans la section 4.5. Elles sont décrites dans le Tableau 4.3. Les deux configurations se distinguent par les deux bits de poids forts. Dans le cas d'instructions de traitement de données les quatre bits suivants sont à `'0'` alors que dans le cas d'instructions "basiques" ils sont utilisées pour définir spécifiquement l'instruction à réaliser.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Commentaires
opcode1																
0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	*Instructions basiques*
0	1	0	0	0	0	*opcode2*				x	x	x	x	x	x	*Instructions de traitement de données*

Tableau 4.3 – Encodage des instructions dans le jeu `ARM Thumb-2`. Les "x" peuvent prendre les valeurs '0' ou '1'. Adapté de [Arm].

Traitement des données

Les instructions traitant des données sont spécifiées avec un second opcode de 4 bits. Leur syntaxe globale est donnée dans le Tableau 4.4. Elles manipulent les données entre les différents registres et sont encodées sous la forme décrite dans le Tableau 4.5.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	*opcode2*				x	x	x	x	x	x

Tableau 4.4 – Encodage des instructions de traitement de données dans le jeu `ARMv7 Thumb`. Les "x" peuvent prendre les valeurs '0' ou '1'. Adapté de [Arm].

Instructions dites "basiques"

Il existe un grand nombre d'instructions différentes de celles décrites précédemment. On ne s'intéresse qu'à certaines d'entre elles dans ces travaux. Cela concerne les instructions de *décalage d'une valeur immédiate*, d'*addition*, de *soustraction*, de *comparaison* et de *déplacement* de données. La Tableau 4.6 décrit la syntaxe globale de ces instructions et ces dernières sont listés dans le Tableau 4.7.

opcode	Instruction	Description
0000	AND	ET logique
0001	EOR	OU-exclusif
0010	LSL	Décalage à gauche logique
0011	LSR	Décalage à droite logique
0100	ASR	Décalage à droite arithmétique
0101	ADC	Addition avec retenue
0110	SBC	Soustraction avec retenue
0111	ROR	Rotation à droite
1000	TST	Test d'égalité sans affectation
1001	RSB	Soustraction inversée
1010	CMP	Soustraction sans affectation
1011	CMN	Addition sans affectation
1100	ORR	OU logique
1101	MUL	Multiplication de registres
1110	BIC	Mise à zéro du bit
1111	MVN	Complémentaire à 1 logique

Tableau 4.5 – Instructions de traitement des données [Arm].

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	*opcode*													

Tableau 4.6 – Encodage des instructions de *décalage d'une valeur immédiate*, d'*addition*, de *soustraction*, de *comparaison* et de *déplacement* de données dans le jeu `ARMv7 Thumb`. Extrait de [Arm].

opcode	Instruction	Description
000xx	LSL (*immediate*)	Décalage à gauche logique de *immediate* bits
001xx	LSR (*immediate*)	Décalage à droite logique de *immediate* bits
010xx	ASR (*immediate*)	Décalage à droite arithmétique de *immediate* bits
01100	ADD (*register*)	Addition de registres
01101	SUB (*register*)	Soustraction de registres
01110	ADD (*immediate*)	Addition d'une valeur immédiate sur 3 bits
01111	SUB (*immediate*)	Soustraction d'une valeur immédiate sur 3 bits
100xx	MOV (*immediate*)	Déplacement d'une valeur immédiate
101xx	CMP (*immediate*)	Comparaison avec une valeur immédiate
110xx	ADD (*immediate*)	Addition d'une valeur immédiate sur 8 bits
111xx	SUB (*immediate*)	Soustraction d'une valeur immédiate sur 8 bits

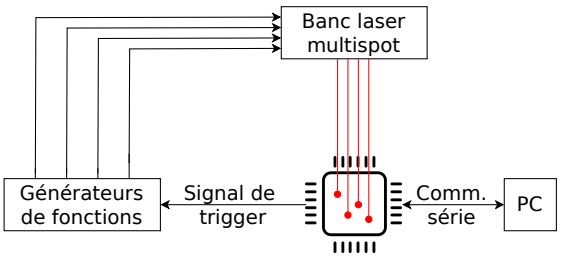
Tableau 4.7 – Instructions dites "basiques". [Arm].

4.5 Caractérisation

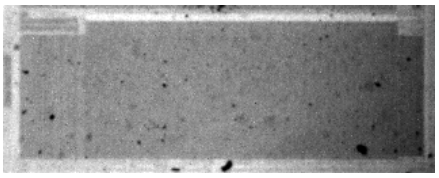
Après une description du montage expérimental et des différents paramètres configurés, cette section s'intéresse à deux scénarios permettant de mettre en avant les nouvelles possibilités offertes par ce banc laser.

4.5.1 Montage expérimental

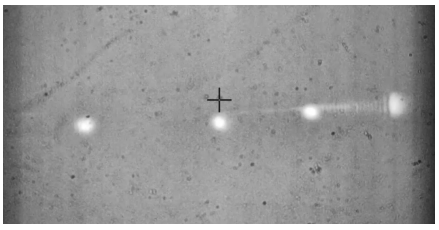
Afin de réaliser une caractérisation du banc laser, le dispositif illustré en Figure 4.6a est mis en place. La cible matérielle communique avec un ordinateur par une liaison série. Les sources laser sont pilotées par quatre signaux distincts générés par un générateur de fonctions. Ainsi chaque source peut être contrôlée indépendamment, cela permet de tirer à des instants différents et pour des durées différentes. Le tir est déclenché par un signal de *trigger* provenant de la cible. Ce signal permet la synchronisation entre l'activité de la cible, l'exécution du code source, et la génération des signaux de contrôle des sources laser. La mémoire Flash du microcontrôleur est visible sur la Figure 4.6b alors que la Figure 4.6c montre les spots laser sur cette mémoire.



(a) Dispositif expérimental utilisé pour la caractérisation du banc laser.



(b) Image infrarouge de la mémoire Flash.



(c) Quatre spots laser sur une zone réduite de la mémoire Flash.

Figure 4.6 – Injection laser à quatre spots sur une mémoire Flash.

Une photographie du banc laser est visible en Figure 4.7. On peut y voir :

— en bleu les quatre sources laser,
— en rouge la cible matérielle,
— en vert le générateur de fonctions,
— en jaune les différents objectifs,
— en orange la caméra infrarouge.

4.5.2 Réglage des sources laser

Dans un premier temps, une caractérisation est réalisée en suivant la procédure décrite dans [Men+20b]. Cela consiste à explorer cinq paramètres : l'instant d'injection, la durée

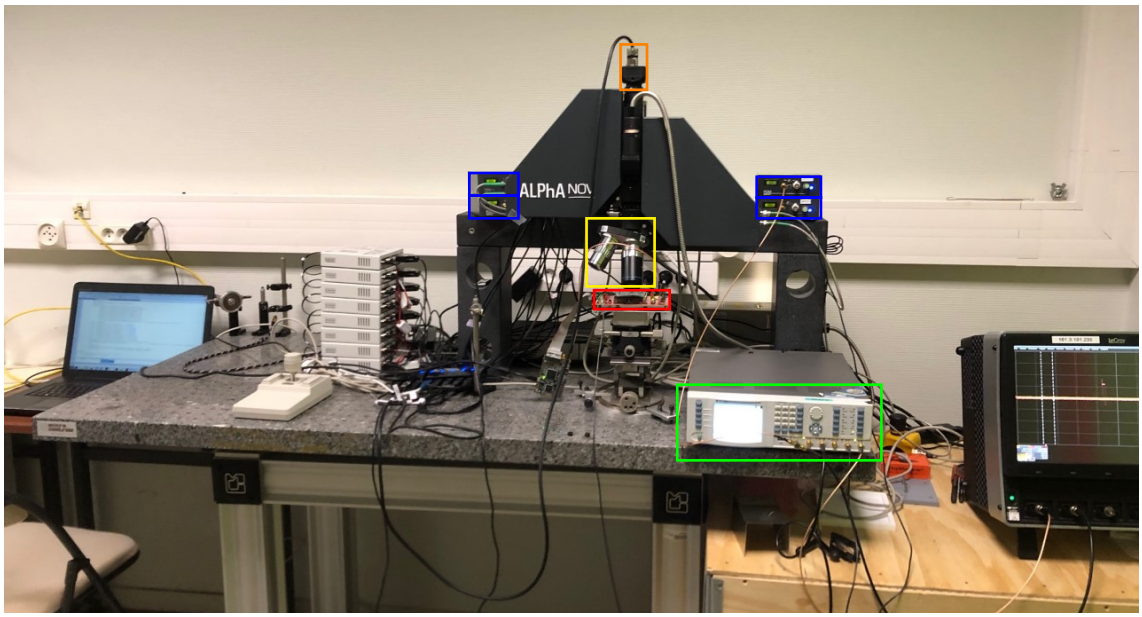


Figure 4.7 – Photographie du montage expérimentale de caractérisation du banc laser multispot.

du tir laser, les coordonnées X et Y du spot laser et la puissance de la source laser. Pour chaque position (X,Y), on augmente progressivement les autres paramètres jusqu'à obtenir les valeurs permettant d'avoir une répétabilité satisfaisante.

Ainsi la puissance du laser est réglée à 1,5 W dans le but d'obtenir des fautes monobits lors de l'opération de lecture d'instructions en mémoire Flash. La source 980 nm et le grossissement $\times 20$ sont utilisés. La durée des tirs lasers est fixée à 135 ns, ce qui correspond à une période de l'horloge de la cible.

4.5.3 Programmes de test

Afin d'évaluer les nouvelles possibilités offertes par ce banc d'injection laser multispot des programmes de tests sont mis en place. Le premier, décrit dans la sous-section 4.5.4, met en avant l'avantage spatial alors que le second, décrit dans la sous-section 4.5.5, met en avant l'avantage temporel.

4.5.4 Avantage spatial

L'objectif de cette première caractérisation est d'évaluer la possibilité d'injecter plusieurs fautes non contiguës simultanément. Dans ce but, une instruction `MOV` qui charge une donnée 8 bits dans un registre est ciblée. La Figure 4.8a illustre le code source utilisé afin de charger la valeur `0x00` dans le registre `R0`. Un signal de *trigger* est déclenché avant l'exécution de l'instruction et désactivé après son exécution. Pour finir, la valeur chargée

dans le registre R0 est relue. L'intérêt de cette instruction est quelle est quasiement entièrement constitué de '0' , elle est donc idéale pour tester un modèle de *bitset*.

```
1 // Raise trigger signal
2 MOV R0, 0x00
3 // Lower trigger signal
4 // Read back R0
```

(a) Code source Assembleur.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Instruction correcte (MOV) : Rd = imm8

0	0	1	0	0	Rd	imm8

Instruction correcte (MOV) : R0 = 0x00

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Instruction fautée (MOV) : R0 = 0x55

0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	1

(b) Chargement de donnée fautée.

Figure 4.8 – Code assembleur de caractérisation pour quatre fautes simultanées sur des bits non contiguës.

Ce code source est compilé en utilisant le jeu d'instruction *Thumb* avec aucune optimisation. La Figure 4.8b montre l'encodage de l'instruction. Dans cet exemple, on vient stocker la valeur immédiate imm8 dans le registre R0. On cible la valeur immédiate imm8 de l'instruction, on souhaite forcer à 0x55 la valeur écrite dans le registre R0 à la place de 0x00 afin de démontrer la possibilité d'injecter des fautes non contiguës simultanément.

Une vidéo de démonstration illustrant l'avantage spatial de ce nouveau banc laser est disponible sur YouTube en français [Cha+21a] ou en anglais [Cha+21b].

Résultats expérimentaux

Conformément aux résultats obtenus par Menu *et al.* présentés en Figure 3.9, il est nécessaire de placer les spots laser à 90 nm les uns des autres car il y a un pas de 45 nm entre les *bitlines*. Afin de trouver le bon instant d'injection laser, on parcourt tous les temps disponibles entre le déclenchement du signal de *trigger* avec un pas de 135 ns. L'instant d'injection permettant de fauter l'instruction ciblée est obtenu pour un délai de 1 113 ns. Dans cette configuration, on parvient à forcer la valeur 0x55 dans le registre R0 en fautant la lecture de l'instruction depuis la mémoire Flash. Le chronogramme visible en Figure 4.9 illustre cette première caractérisation.

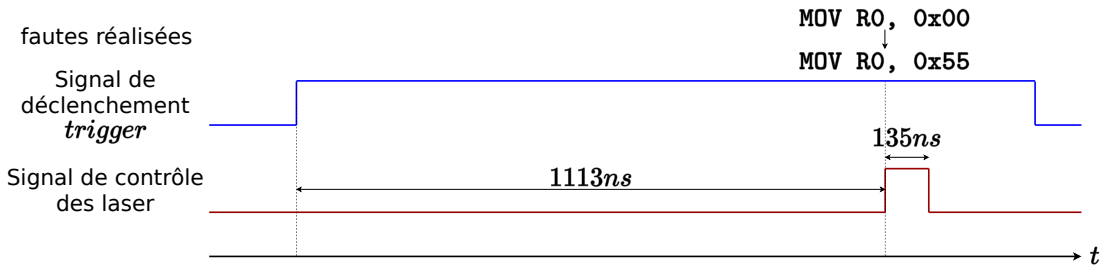


Figure 4.9 – Chronogramme de la caractérisation de l'avantage spatial

4.5.5 Avantage temporel

Cette seconde caractérisation a pour but d'évaluer la possibilité d'injecter des fautes sur des bits différents de deux instructions proches dans le temps. Dans ce but, le code présent dans la Figure 4.10a est mis en place.

Dans ce programme, on parcourt toutes les valeurs entières allant de 1 à `N_ITER` et pour chaque valeur `iter` :

— un compteur `ref_count` est incrémenté dans le but de compter le nombre de passages effectifs dans la boucle

— les opérations $iter \oplus iter$ et $iter + iter$ sont effectuées et les résultats sont respectivement stockées dans les variables `XOR` et `ADD`.

```c
#define N_ITER 1000
void charac_func(void) {
    volatile uint32_t ref_count = 0;
    uint32_t results[2] = {0, 0};
    uint32_t XOR, ADD = 0;
    trigger_high();
    for (volatile uint32_t iter = 1; iter <= N_ITER; iter++) {
        ref_count++;
        XOR = iter ^ iter;
        ADD = iter + iter;
        results[1] += (XOR == ADD);
        }
    results[0] = N_ITER - ref_count;
    trigger_low();
    // Read back results
}
```

(a) Code C de caractérisation avec chaque instruction ciblée est en rouge et en gras.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Instruction correcte (ADD) : Rdn = Rdn + imm8

0	0	1	1	0	Rdn	imm8

Instruction correcte (ADD) : Rdn = Rdn +1

0	0	1	1	0	Rdn	0	0	0	0	0	0	0	1

Instruction fautée (ADD) : Rdn = Rdn + 5

0	0	1	1	0	Rdn	0	0	0	0	0	1	0	1

(b) Faute sur l'incrément de la boucle.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Instruction correcte (EORS) : Rdn = Rdn ⊕ Rm

0	1	0	0	0	0	0	0	0	1	Rm	Rdn

Instruction fautée (ADCS) : Rdn = Rdn + Rm

0	1	0	0	0	0	0	1	0	1	Rm	Rdn

(c) Faute sur l'opération OU-exclusif.

Figure 4.10 – Code de caractérisation pour deux fautes proches dans le temps sur des bits d'indice différent.

L'objectif ici est de cibler les instructions associées aux deux opérations suivantes :

— l'incrément du compteur de boucle. À chaque passage dans la boucle la valeur `iter` est incrémenté de `'1'` soit $iter = iter + 1$. Cette instruction est compilée en une instruction `ADD` comme illustré dans la Figure 4.10b. La valeur de l'incrément est stocké dans la valeur immédiate `imm8` et est la cible de l'injection. Ainsi, on modifie la valeur de l'incrément de la boucle pour le changer en `N` au lieu de `1`. Avec un modèle de faute mono-bitset, `N` sera de la forme $2^i + 1$ avec $i \in [\![1\ ;\ 7]\!]$. Dans ce cas, il faut injecter un bitset sur le bit d'index i. Par exemple, l'incrément peut devenir `5` si le bit d'index 2 est fauté, comme montré sur la Figure 4.10b

— un OU-exclusif dans la boucle. Cette instruction est compilée en une instruction `EORS` comme illustré dans la Figure 4.10c. L'opcode de l'instruction est la cible de l'injection. Ainsi, on modifie l'instruction réalisée pour la passer de `EORS` à `ADCS` (`ADd Carry S`, le `S` spécifie que l'instruction mettra à jour un *flag* pouvant indiquer que le résultat de l'opération est nul, négatif, contient une retenue ou présente un *overflow*) dans le but de réaliser une addition avec retenue. Dans ce cas, il faut injecter un bitset sur le $8^{ème}$ bit de l'instruction.

Un signal de *trigger* est déclenché avant l'exécution des instructions ciblées (voir Figure 4.10a, ligne 6) et désactivé à la fin (voir Figure 4.10a, ligne 17). Les résultats de l'expérience sont stockés dans un tableau à deux éléments `results`. Le premier élément contient la différence entre le nombre d'exécutions prévues et le nombre d'exécutions réalisée de la boucle `for`. Le second élément du tableau contient le nombre de fois où l'opération OU-exclusif (`EORS`) est transformé en une addition(`ADCS`). Il est ainsi possible d'isoler les deux fautes et d'observer leur impact distinctement.

Résultats expérimentaux

Plusieurs expériences ont été réalisées en modifiant la valeur de l'incrément de la boucle pour différentes valeurs tout en transformant l'instruction `EORS` du corps de la boucle `for`. Comme spécifié précédemment, la cible matérielle envoie un signal de *trigger* unique. Le défi principal est de trouver les paramètres des signaux de contrôle des deux sources lasers. Ces paramètres sont au nombre de quatre :

— le délai initial pour la première source laser t_{init_1}. Il correspond au temps entre le déclenchement du signal de *trigger* et l'exécution de la première instruction à fauter.

— le délai initial pour la seconde source laser t_{init_2}. Il correspond au temps entre le déclenchement du signal de *trigger* et l'exécution de la seconde instruction à fauter.

— la période t_{laser} qui correspond au temps d'exécution d'une itération de la boucle `for` afin de fauter l'opération OU-exclusif à chaque itération de la boucle. Les deux signaux de contrôle ont la même période.
— le rapport cyclique α qui définit la durée de l'impulsion laser. Comme on désire que chaque laser ne faute qu'une instruction par exécution de la boucle `for`, cette grandeur doit être réglée en conséquence.

La première étape est de trouver les deux délais initiaux. Cela est effectué en augmentant progressivement ces valeurs l'une après l'autre en regardant les résultats de l'exécution du programme. Dès qu'une faute est observée, le délai initial est obtenu. Nous avons obtenu les valeurs suivantes : $t_{init_1} = 2\,070$ ns et $t_{init_2} = 3\,825$ ns.

La seconde étape est de trouver la période des signaux de contrôles des sources laser. Cette étape est réalisée est effectuant deux tirs lasers successifs en augmentant progressivement la période jusqu'à l'obtention de deux fautes. Nous avons obtenu $t_{laser} = 5\,535$ ns. Cette valeur correspond à 41 cycles d'horloge ($5535 = 41 \times 135$) pour notre cible possédant une horloge interne avec une période de 135 ns.

La dernière étape est de trouver le rapport cyclique. Pour cela, on sait qu'une exécution de la boucle `for` nécessite 41 cycles d'horloge. On peut ainsi configurer le rapport cyclique à $\alpha = \frac{1}{41} = 2.5\%$ afin de cibler une instruction sur 41 dans le corps de la boucle.

La Figure 4.11 montre l'évolution temporelle des différents signaux de contrôle des sources laser. Avec cette configuration, il est effectivement réalisable de changer l'incrément de la boucle à `5` au lieu de `1` et d'altérer l'opération OU-exclusif dans le corps de la boucle pour la transformer en addition.

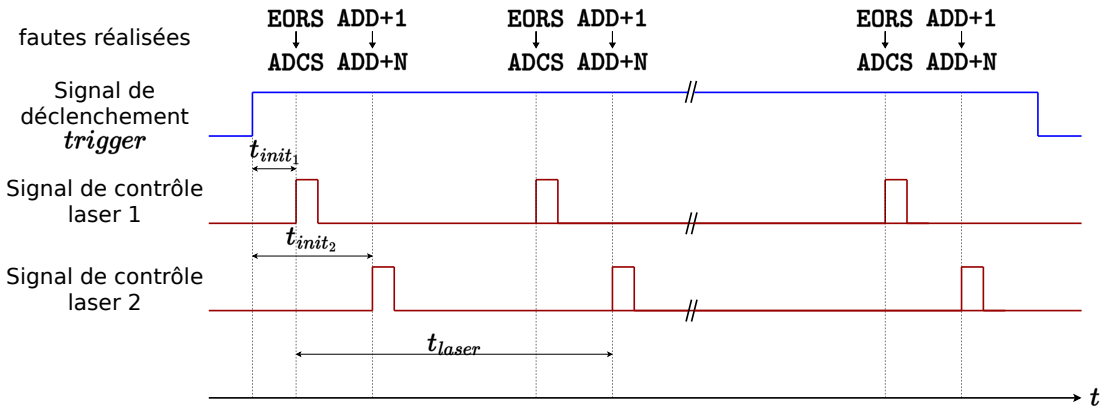


Figure 4.11 – Évolution temporelle des différents signaux de contrôle.

Ainsi, dans cette section, nous avons démontré les nouvelles possibilités offertes par ce banc laser multispot. D'une part, il est en effet possible de fauter plusieurs bits non

contiguës d'une même instruction, c'est à dire à un instant donné. D'autre part, il est également réalisable de fauter des bits différents d'instructions très proches temporellement. Dans chaque configuration, nous avons obtenu une répétabilité de 100%.

4.6 Nouvelles possibilités d'attaques

Cette partie s'intéresse, d'un point de vue théorique, aux nouvelles possibilités d'attaques offertes par ce banc laser concernant la corruption d'opcode d'instructions de traitement de données. Le Tableau 4.8 synthétise l'ensemble des instructions atteignables en injectant jusqu'à quatre bitsets dans l'opcode des instructions de traitement de données. Par exemple, en injectant 3 bitsets dans l'opcode d'une instruction `AND`, il est possible d'obtenir les instructions `ROR`, `CMN`, `MUL` ou `BIC` selon la position des fautes. On peut observer qu'il existe de nombreuses options qui ne sont réalisables qu'avec un moyen d'injection multiple de fautes.

La pertinence de ces corruptions dépend grandement du code source exécuté par le microcontrôleur. Dans un contexte d'attaque d'algorithme cryptographique, il est raisonnable de penser que de nombreuses fautes décrites dans ce tableau n'aboutissent pas à un affaiblissement de la sécurité de l'algorithme.

En revanche, il est également possible de fauter l'opcode et les identifiants des registres, ou l'opcode et les valeurs immédiates, etc. Il est ainsi impossible de réaliser une étude exhaustive des scénarios envisageables.

Instructions \ # bitsets	1				2						3				4
AND 0000	EOR 0001	LSL 0010	ASR 0100	TST 1000	LSR 1001	ADC 0101	SRC 0110	NEG 1001	CMP 1010	ORR 1100	ROR 0111	CMN 1011	MUL 1101	BIC 1110	MVN 1111
EOR 0001	LSR 0011	ADC 0101	NEG 1001		ROR 0111	CMN 1011	MUL 1101				MVN 1111				
LSL 0010	LSR 0011	SBC 0110	CMP 1010		ROR 0111	CMN 1011	BIC 1110				MVN 1111				
ASR 0100	ADC 0101	SBC 0110	ORR 1100		ROR 0111	MUL 1101	BIC 1110				MVN 1111				
TST 1000	NEG 1001	CMP 1010	ORR 1100		CMN 1011	MUL 1101	BIC 1110				MVN 1111				
LSR 0011	ROR 0111	CMN 1011			MVN 1111										
ADC 0101	ROR 0111	MUL 1101			MVN 1111										
SBC 0110	ROR 0111	BIC 1110			MVN 1111										
NEG 1001	CMN 1011	MUL 1101			MVN 1111										
CMP 1010	CMN 1011	BIC 1110			MVN 1111										
ORR 1100	MUL 1101	BIC 1110			MVN 1111										
ROR 0111	MVN 1111														
MUL 1101	MVN 1111														
CMN 1011	MVN 1111														
BIC 1110	MVN 1111														

Tableau 4.8 – Nouvelles possibilités de corruption d'opcode pour le jeu d'instruction ARMv7.

4.7 Conclusion

Dans ce chapitre, nous avons tout d'abord montré les limites des attaques par injection laser avec un banc laser mono-spot. En effet, le modèle de faute existant ne prend pas en compte deux caractéristiques : la *contiguïté* des fautes et l'aspect temporel du modèle de faute associé.

Nous avons ensuite décrit un nouveau banc laser multispot permettant de s'affranchir de ces limites et d'explorer de nouvelles possibilités d'attaques. Dans ce but, deux expériences ont été réalisées afin de démontrer que de nouveaux types de fautes sont possibles. D'une part, il a été possible d'injecter quatre fautes non contiguës simultanément. D'autre part, il a été également possible d'injecter deux fautes très proches dans le temps mais à des positions différentes.

Pour finir, une exploration non exhaustive des nouvelles possibilités de corruptions d'*opcodes* dans le cadre du jeu d'instruction `ARMv7` est réalisée. Cette dernière montre qu'un banc laser à quatre spots offre un très nombre de fautes possibles par rapport à un banc laser monospot.

Ce chapitre a fait l'objet d'une publication en conférence internationale :

- `B. COLOMBIER, P. GRANDAMME, J. VERNAY` *et al.*, "Multi-spot Laser Fault Injection Setup : New Possibilities for Fault Injection Attacks", CARDIS 2021. [Col+22]

Chapitre 5

Injection laser de fautes sur circuit non alimenté

Table des matières

5.1 Introduction

Le chapitre précédent (Chapitre 4) a décrit l'utilisation d'un banc laser multispot dans le but d'injecter de multiples fautes au sein d'un composant alimenté. Les résultats obtenus reposent sur le modèle de faute accepté par l'ensemble de la communauté.

Ce modèle de faute n'est valable que pour des attaques réalisées sur des circuits alimentés. Ces derniers peuvent embarquer des contremesures, logicielles ou matérielles, permettant aux composants de détecter et de réagir à ces attaques. La majeure partie de ces protections ne sont efficaces que lorsqu'elles sont alimentées. Elles sont alors qualifiées d'*actives*.

Se pose alors la question du contournement de ces dispositifs de protection. Dans cet objectif, plusieurs scénarios sont envisageables. D'une part, les capteurs d'injection laser constituent l'une des protections qui peuvent être mises en place. Un attaquant peut alors songer à dégrader voire détruire physiquement ces capteurs. Il doit en connaître le fonctionnement et la localisation, ce qui limite le réalisme de ce scénario. D'autre part, il est également possible d'attaquer certaines sous-parties du composant lorsque le composant est éteint (les capteurs étant inactifs). L'altération de sous-parties du composant aura un impact sur le fonctionnement de ce dernier lorsqu'il sera alimenté à nouveau. Les mémoires non-volatiles, et plus spécifiquement les mémoires Flash, contiennent encore les données stockées de manière permanente (code embarqué, clé cryptographique, etc.) lorsque le composant est éteint. Une altération de leur contenu impactera le comportement du composant, elles représentent ainsi des cibles privilégiées.

C'est ce second scénario qui est abordé dans cette étude. En effet, ce chapitre s'intéresse aux injections laser de fautes au sein de mémoires Flash de microcontrôleurs non alimentés. Une description complète du nouveau modèle de faute, une caractérisation de ce nouveau vecteur d'attaque ainsi qu'une mise en pratique visant une application cryptographique seront présentées dans ce chapitre.

Ce chapitre a fait l'objet d'une publication dans la revue internationale TCHES en 2024 [Gra+24].

5.2 Modèle de faute

Pour comprendre et décrire l'effet d'une injection laser de faute sur une cible matérielle, il est nécessaire de définir un modèle de faute. Cette description peut être faite à plusieurs niveaux d'abstraction. Ces niveaux s'étendent du niveau *physique*, le plus bas niveau, au niveau *applicatif*, le plus haut niveau, en passant les niveaux *logique* et *mémoire*.

La Figure 5.1 illustre les liens entre les différents niveaux d'abstractions d'un modèle de faute. Il est important d'avoir une bonne compréhension du modèle de faute à tous ces niveaux afin de proposer des contremesures efficaces. Dans le cas contraire, les protections proposées peuvent être trop coûteuses ou dans le pire cas inefficaces.

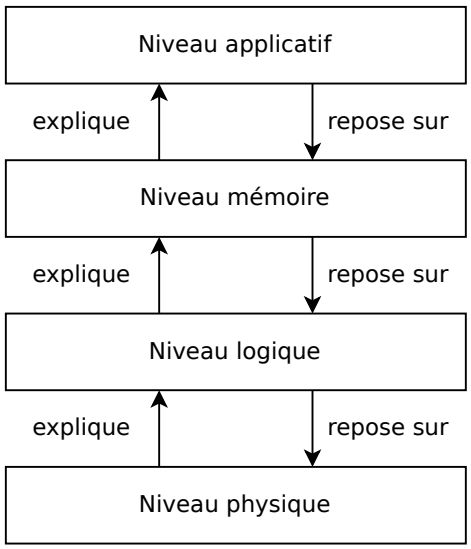


Figure 5.1 – Représentation schématique des liens entre les différents niveaux d'abstraction d'un modèle de faute.

Le modèle de faute lié aux injections laser sur des composants en fonctionnement donc alimentés est maîtrisé à tous les niveaux d'abstraction. En revanche, il ne peut s'appliquer dans le cadre d'attaques sur des circuits non alimentés. En effet, il n'existe aucun champ électrique dans les oxydes des transistors, ce qui ne permet pas la séparation des paires électron/trou. Ces dernières se recombinent donc très vite et aucun effet n'est visible. Ainsi il n'y a pas de fautes liées à l'apparition de courants transitoires par effet photoélectrique. Il est donc nécessaire de revoir le modèle de faute comme nous le proposons dans la suite.

5.2.1 Modèle de faute au niveau physique en mémoire Flash

Le niveau physique représente le plus bas niveau d'abstraction que nous considérons, au plus près des charges électriques et des transistors. Il illustre les mécanismes physiques conduisant à l'apparition d'une faute.

D'après [San11], l'énergie fournie par le faisceau lumineux génère une élévation de température locale au sein du composant électronique. L'énergie lumineuse est convertie en énergie thermique et cette dernière est transmise aux charges électriques, les électrons, stockées dans les grilles flottantes des transistors à grille flottante qui composent

la mémoire Flash. Ainsi, les électrons obtiennent une énergie suffisante pour franchir la barrière de potentiel et s'échapper de la grille flottante.

L'intensité du rayon laser dans le plan focal de la lentille suit une distribution gaussienne radiale comme décrit dans l'Équation 5.1 [Buc+13], avec ω_0 la dispersion de la distribution gaussienne dans le plan focal et r la distance radiale du centre du spot laser.

$$I(r) = I_0 \cdot e^{-\frac{2r^2}{\omega_0^2}} \tag{5.1}$$

Le paramètre ω_0 dépend de la longueur d'onde λ du laser et de l'ouverture numérique NA de l'objectif avec la relation définie par l'Équation 5.2.

$$\omega_0 = \frac{2\lambda}{\pi \cdot NA} \tag{5.2}$$

La taille du spot laser est définie par le critère Full-Width at Half Maximum (FWHM), c'est-à-dire le diamètre de la zone où l'intensité du rayon laser est égale à la moitié de son maximum. L'Équation 5.3 donne la relation entre la taille du spot d_0 et le paramètre ω_0.

$$d_0 = \omega_0 \cdot \sqrt{\frac{\ln 2}{2}} \tag{5.3}$$

Le banc laser utilisé dans les expériences réalisées dans ce chapitre est dans la sous-section 5.3.1. Le banc possède une longueur d'onde λ de 1 064 nm et l'objectif utilisé une ouverture numérique NA de 0,16. Ce banc laser possède également une source dite "purement thermique" avec une longueur d'onde $\lambda_{th} = 1\,300$ nm. Ici, le choix a été fait de réaliser les attaques avec une source classiquement utilisée dans les injections laser de fautes, c'est-à-dire la source à 1 064 nm.

Les cibles matérielles sur lesquelles sont réalisées les expériences sont les mêmes que dans le Chapitre 4 et embarquent 128 kB de mémoire Flash. L'image infrarouge visible en Figure 5.2 nous permet d'estimer la surface de la mémoire Flash. On ne considère que la partie centrale contenant les transistors à grille flottante pour le calcul de la surface d'une cellule mémoire unitaire. Cette dernière s'étend sur environ 1 400 µm de long et 600 µm de large. Les travaux de `Menu` [Men21] permettent de réaliser une partie de l'ingénierie inverse de l'organisation de la mémoire Flash de ce composant. Elle est ainsi organisée en 2 048 *bitlines* et 512 *wordlines*.

Ces informations nous permettent d'estimer la taille d'une cellule Flash de ce composant :

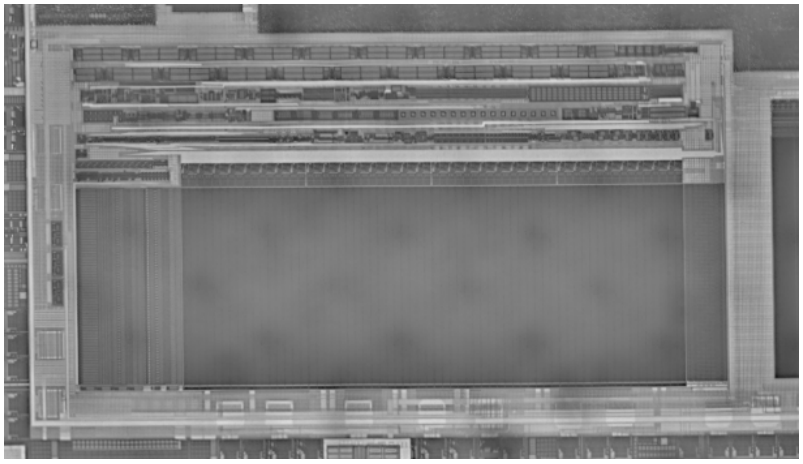


Figure 5.2 – Image infrarouge de la mémoire Flash.

$$width = \frac{1400}{2048} \approx 0{,}68\,\mu\text{m} \qquad length = \frac{600}{512} \approx 1{,}17\,\mu\text{m}. \tag{5.4}$$

Une simulation numérique, réalisée en `Python` et représentée en Figure 5.3, permet de générer la *carte thermique* causée par l'exposition au laser du circuit. Sur cette représentation, chaque rectangle noir illustre la surface occupée par un transistor à grille flottante de la mémoire Flash. Le cercle blanc schématise le spot laser conformément au critère FWHM pour un objectif $\times 20$. Cela n'augure rien de la zone dans laquelle des fautes seront injectées. L'échelle est exprimée en pourcentage de I_0 qui est l'intensité maximale présente au centre du spot laser. Cette grandeur n'est pas connue.

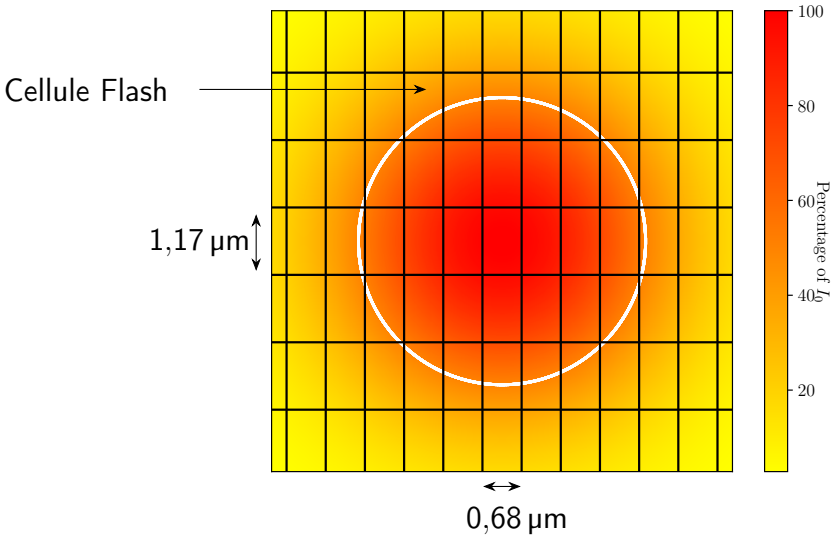
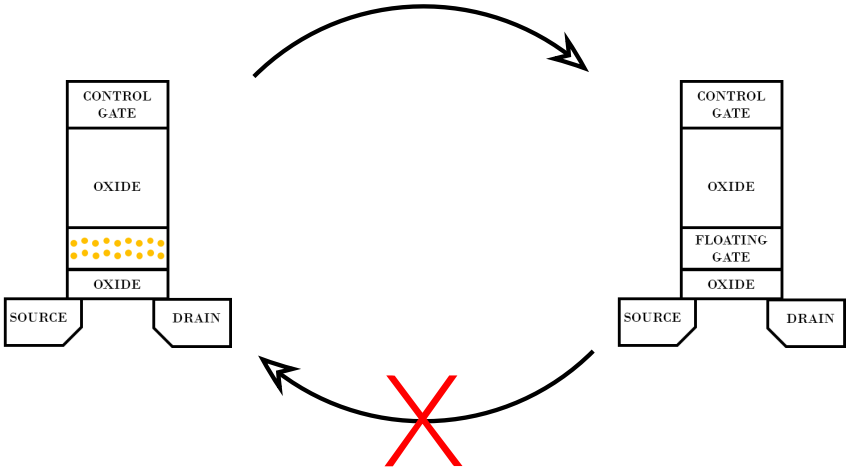


Figure 5.3 – Carte thermique générée par l'exposition laser avec l'objectif $\times 20$ (Simulation numérique avec $\lambda = 1\,064$ nm et $NA = 0{,}16$).

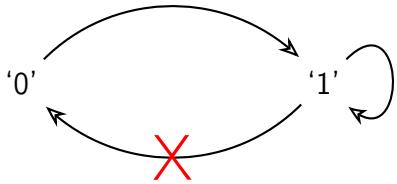
D'une part, on peut observer que le spot laser recouvre de nombreux transistors à grille

flottante. Cela indique que plusieurs cellules se retrouvent chauffés par l'exposition laser. D'autre part, l'énergie du laser est concentrée au centre du spot et décroît exponentiellement avec le carré de la distance du centre du spot laser. Le nombre de fautes est déterminé par le nombre de transistors suffisamment chauffés. On ne peut pas prédire cette valeur.

Une exposition laser peut ainsi conduire à la décharge des transistors à grille flottante de la mémoire Flash au sein de composants non alimentés comme illustré en Figure 5.4a. Sur cette figure, on peut voir deux transistors à grille flottante, l'un chargé à gauche avec les électrons représentés par les points jaunes présents dans la grille flottante et l'autre déchargé à droite sans électrons dans sa grille flottante.



(a) Illustration du modèle de faute au niveau physique.



(b) Illustation du modèle de faute au niveau logique.

Figure 5.4 – Modèle de faute au niveau physique et logique.

5.2.2 Modèle de faute au niveau logique en mémoire Flash

L'étude au niveau physique, décrite ci-dessus, a démontré qu'il était possible de chauffer localement les transistors à grille flottante qui composent la mémoire Flash en les exposant avec une source laser de longueur d'onde $\lambda = 1\,064\,\text{nm}$. En nous appuyant sur les travaux de `Skorobogatov` [Sko09], nous faisons l'hypothèse que l'élévation de température sera à l'origine d'une décharge des grilles flottantes de ces transistors. Dans ce cadre, il sera possible de décharger un transistor chargé mais impossible de charger

un transistor déchargé, le mécanisme ne sera pas destructif et sera persistant jusqu'à la prochaine programmation du transistor. Nous obtiendrons donc un modèle de faute unidirectionnel. Nous verrons dans la section 5.3 que notre hypothèse est vérifiée, et elle est appliquée dans un contexte d'attaque dans la section 5.4.

D'après la documentation du composant, l'état effacé, c'est-à-dire déchargé, de la mémoire correspond à un `'1'` logique. Ainsi, avec ce modèle de faute, il est possible de fauter un `'0'` logique stocké en mémoire Flash pour le transformer en un `'1'` logique si suffisamment de charges électriques sont évacuées de la grille flottante. En revanche, un `'1'` stocké en mémoire sera toujours lu comme un `'1'` comme il est impossible d'injecter des électrons dans la grille flottante. Nous obtenons donc un modèle de faute dit "*data-dependant*" et unidirectionnel comme illustré en Figure 5.4b.

Ainsi, l'injection de faute laser au sein d'une mémoire Flash de ce composant non alimenté aboutit à un modèle de faute de *bitset*. Il est important de noter qu'un modèle de type *bitreset* peut être obtenu si la convention inverse est choisie par le fabricant du composant. La Figure 5.4 illustre ce modèle de faute au niveau physique et sa conséquence au niveau logique.

5.2.3 Modèle de faute au niveau mémoire Flash et niveau applicatif

Dans les circuits intégrés dédiés aux applications IoT, les mémoires Flash sont utilisées pour stocker l'ensemble des données nécessaires de façon permanente. Ces données peuvent être le code exécuté par le composant, des constantes, des valeurs de configuration du fonctionnement du composant, des clés cryptographiques, etc. Le modèle de faute, décrit au niveau physique puis logique, montre qu'il est possible de corrompre chacune de ces données afin de modifier le fonctionnement global du circuit. Concernant la corruption du code, on peut retrouver l'exemple décrit en Figure 3.10. On peut également prendre pour exemple la corruption des constantes utilisées dans des algorithmes cryptographiques. C'est le cas de la PFA, une cryptanalyse basée sur un biais dans la distribution des octets en sortie d'algorithmes de chiffrement symétrique, qui repose sur la corruption de la S-Box de ces algorithmes. Ce scénario sera mis en place et décrit dans la section 5.4.

5.3 Validation expérimentale du modèle de faute au niveau logique

5.3.1 Matériel

Le banc laser utilisé dans ces travaux est constitué d'une source laser et d'un système optique. Ces deux éléments sont reliés par une fibre optique monomode. Une représentation schématique de ce banc laser est disponible en Figure 5.5. Ce banc monospot est couramment utilisé pour réaliser des injections de fautes en suivant le modèle de faute usuel en mémoire Flash, c'est-à-dire pendant la lecture des données en mémoire Flash.

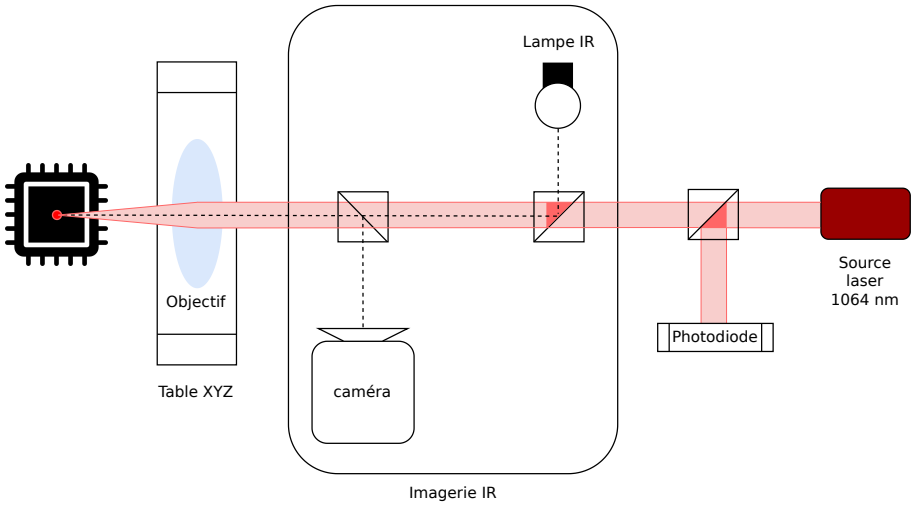


Figure 5.5 – Schéma du banc laser Pulscan [Pul24].

Le banc laser est construit sur une architecture client-serveur permettant à l'utilisateur de contrôler le banc laser et le système de positionnement de façon logicielle.

La source laser permet de générer des impulsions laser comprises entre 50 ns et 1 s. La puissance de la source laser peut être configurée entre 0 W et 3 W. La source laser génère un flux continu de photons. La durée de l'impulsion est contrôlé par le pilotage d'un interrupteur optique commandé électriquement. Lors de nos essais la source laser est configurée en mode ‘‘`IntCLK`’’ qui permet de déclencher les tirs laser selon les fronts montants d'un signal d'horloge interne dont la fréquence f peut être configurée entre 0,025 Hz et 1 MHz. Ce mode de fonctionnement permet au banc laser d'avoir un comportement entièrement asynchrone de la cible. En effet, cette dernière étant éteinte, il n'y a donc pas besoin de synchroniser le banc avec la cible.

Comme dans le chapitre précédent, les injections laser sont réalisées avec une source de

longueur d'onde 1 064 nm. Les photons de cette longueur d'onde peuvent pénétrer le silicium sur une distance d'environ 650 µm [Men21]. Cela permet au rayon lumineux de traverser le substrat et d'atteindre les zones actives des transistors [Buc+13]. L'Équation 5.3 et l'Équation 5.2 nous permettent de calculer la taille du spot d_0 dans le plan focal de l'objectif conformément au critère FWHM. Ce banc laser possède un montage optique avec trois objectifs présentés dans le Tableau 5.1. Ainsi, selon le scénario, un attaquant peut choisir l'objectif qui convient.

Objectif	x5	x20	x100
Grossissement	x5	x20	x100
Ouverture numérique (NA)	0,04	0,16	0,8
Coefficient de transmission	67 %	57 %	26 %
Diamètre du spot d_0 (μm)	20	5	1

Tableau 5.1 – Caractéristiques des objectifs du banc laser Pulscan.

La cible matérielle est la même que celle décrite dans la sous-section 4.4.2.

5.3.2 Protocole expérimental

D'après le modèle de faute décrit en section 5.2, il est possible de décharger les transistors à grille flottante d'une mémoire Flash non alimentée en les chauffant par une exposition prolongée à une source laser infrarouge. Comme indiqué précédemment, pour la cible choisie, l'état *chargé* des transistors à grille flottante est représenté par un `'0'` et l'état déchargé par un `'1'` logique.

La mémoire Flash est donc, dans un premier temps, complètement initialisée à zéro afin de caractériser la possibilité d'injecter des fautes de type *bitset*.

Les coordonnées X et Y sont alors parcourues afin de cartographier des fautes injectées pour X compris entre 200 µm et 1 200 µm avec un pas de 100 µm et Y compris entre 100 µm et 500 µm avec un pas de 100 µm. La cible est éteinte avant chaque série de tirs laser et allumée ensuite afin de relire le contenu de la mémoire. La décharge des grilles flottantes est un effet cumulatif, c'est pourquoi des séries de 1 000 tirs consécutifs sont réalisées pour chaque position. La fréquence des tirs laser est fixée à 1 Hz et la durée des impulsions à 0,9 s. Au vu des temps longs nécessaires pour chaque position (environ 17 minutes par position), il n'est pas raisonnable d'avoir des pas fins pour les coordonnées X et Y. L'Algorithme 1 décrit la procédure expérimentale suivie pour cette caractérisation. La puissance du laser est également un paramètre à explorer. Nous avons choisi de commencer à une puissance de 0,5 W et de l'incrémenter avec un pas de 0,1 W. Pour chaque valeur de puissance et pour chaque position la mémoire est initialisée et le composant est éteint. Les séries de 1 000 tirs laser consécutifs sont ensuite réalisées

jusqu'à l'obtention d'une faute dans la mémoire. L'entièreté du contenu de la mémoire est ensuite récupérée.

Algorithme 1 Cartographie des fautes injectées.

Require: $X_{min}, X_{max}, X_{step}, Y_{min}, Y_{max}, Y_{step}, P_{min}, P_{max}, P_{step}$
  for $p \in \texttt{range}(P_{min}, P_{max}, P_{step})$ **do**
    set laser power to p
    for $x \in \texttt{range}(X_{min}, X_{max}, X_{step})$ **do**
      for $y \in \texttt{range}(Y_{min}, Y_{max}, Y_{step})$ **do**
        reset target memory
        do
          move laser to (x,y)
          power target off
          for $i \in [0, \ldots, 999]$ **do**
            laser shot
          power target on
          dump target memory
        while #faults == 0
        mapping[x][y] = #faults
  return mapping[x][y]

5.3.3 Résultats

Les premiers effets du laser sont obtenus pour une puissance de 1 W en sortie de la fibre optique et avant son passage au travers du microscope. Des fautes de type *bitset* sont obtenues pour l'ensemble des positions testées. Cela correspond bien à un effacement, c'est-à-dire une décharge, des transistors à grille flottante qui composent la mémoire Flash. De plus, les fautes sont présentes de façon persistante jusqu'à une reprogrammation de la mémoire et aucun effet destructif n'est observé. Ces résultats confirment le modèle de faute proposé en section 5.2.

En effectuant une lecture complète du contenu de la mémoire Flash et en analysant les fichiers au format *.hex*, il est possible d'obtenir les adresses et valeurs des fautes injectées.

Les résultats sont représentés en Figure 5.6 sur laquelle les points bleus représentent les positions fautées après 1 000 tirs, les points rouges les positions fautées après 2 000 tirs et les points oranges les positions fautées après 3 000 tirs. On peut ainsi observer que le nombre de tirs laser nécessaires pour injecter une faute n'est pas le même pour chaque position. Les premières positions sont fautées avec 1 000 tirs seulement. En revanche, un maximum de 3 000 tirs est nécessaire afin de fauter toutes les positions.

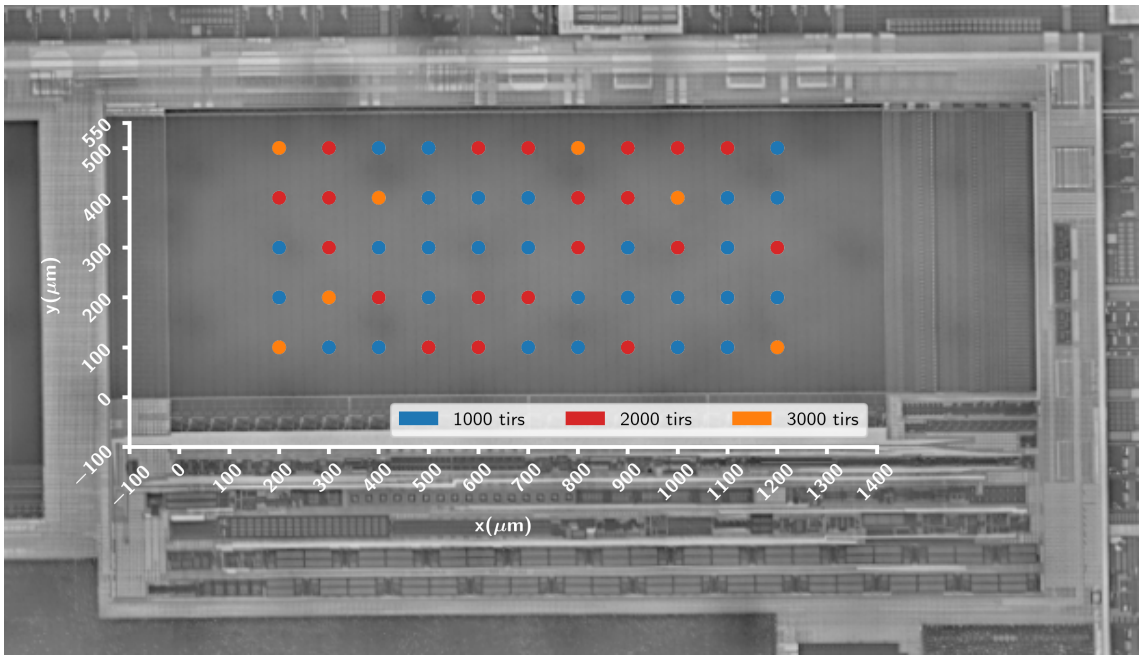


Figure 5.6 – Cartographie des fautes injectées. $P_{laser} = 1\,\text{W}, f_{laser} = 1\,\text{Hz}, T_{pulse} = 0{,}9\,\text{s}$

Une étude statistique du nombre de fautes injectées pour chaque position nous permet d'observer la distribution du nombre de fautes en fonction du nombre de tirs et est visible en Figure 5.7. On remarque tout d'abord que dans de nombreux cas, les fautes sont multiples avec un maximum de 6 fautes. Lorsque plusieurs bits sont fautés, ils sont toujours situés à des positions physiques adjacentes en mémoire Flash. On peut ainsi observer que, pour toutes positions confondues, une moyenne de 2,2 fautes sont obtenues. Cette valeur est illustrée par la ligne rouge sur la figure. Cela s'explique par le fait que le spot laser recouvre plusieurs transistors à grille flottante (environ 25) comme détaillé dans la Figure 5.4a. Des fautes monobits sont obtenues dans un tiers des cas. Par ailleurs, aucune corrélation entre le nombre de tirs nécessaires pour injecter une faute et le nombre de fautes obtenues n'a été observée.

La valeur moyenne de 2,2 bits fautés peut paraître limitante dans l'objectif d'obtenir des fautes monobits, mais ces résultats sont obtenus avec une mémoire Flash complètement initialisée à `'0'`. Dans le cas d'un *firmware réel*, il est raisonnable de penser que la mémoire Flash contient à peu près autant de bits à `'0'` que de bits à `'1'`. Ainsi, un modèle de faute monobit peut être obtenu comme démontré dans la section 5.4.

Le nombre de tirs nécessaires pour injecter une faute peut varier sensiblement d'un composant à l'autre. Cela peut être dû à des effets de vieillissement, de précédentes attaques ou des usages différents des composants. Les transistors à grille flottante sont en effet

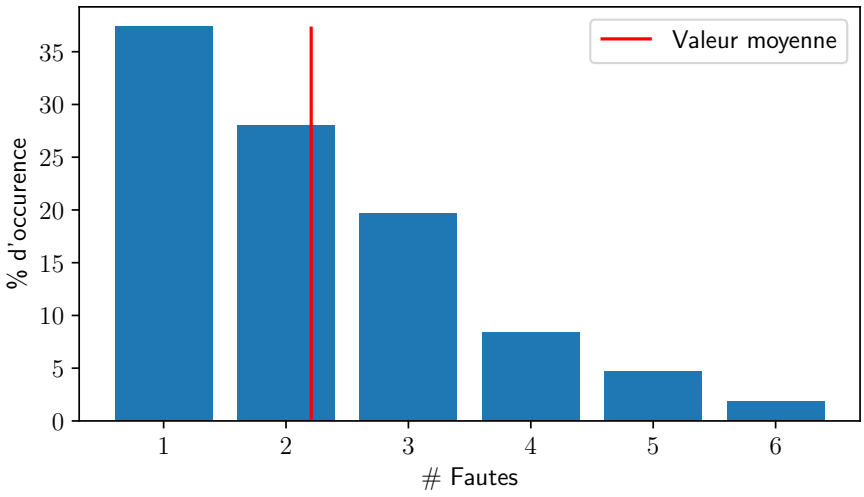


Figure 5.7 – Distribution expérimentale du nombre de fautes injectées pour l'ensemble des positions.

connus pour avoir une durée de vie limitée en terme d'effacement et de programmation. De plus, nous n'avons pas observé d'effet destructif lors de nos essais. Du point de vue d'un attaquant, il est possible d'effectuer le nombre maximal de tirs sans aucun effet parasite.

L'ensemble des fautes observées sont persistantes, c'est-à-dire qu'elles sont présentes jusqu'à une réécriture de la mémoire. Les fautes ne sont pas destructives, aucune dégradation de la mémoire n'a été observée et la mémoire a conservé sa fonctionnalité. De plus, des essais sur des cellules déchargées ont également été réalisés et aucune faute n'est apparue. L'ensemble de ces résultats permettent de valider le modèle de faute proposé dans la section 5.2.

5.3.3.a Ingénierie inverse

Les résultats décrits ci-dessus nous permettent de réaliser l'ingénierie inverse complète du *mapping* de la mémoire Flash, c'est-à-dire de retrouver la correspondance entre l'adresse logique d'une donnée et sa position physique au sein de la mémoire. C'est une étape nécessaire préalable à tout scénario d'attaque comme la PFA presentée plus loin.

Dans un premier temps, les données en mémoire sont stockées sous la forme de mots de 32 bits. La Figure 5.8 illustre l'organisation des bits au sein des mots de 32 bits dans la mémoire Flash de ce composant. On peut ainsi observer que la mémoire est organisée en 32 colonnes. La première colonne contient tous les bits d'indice 31 des mots de 32 bits et la dernière contient tous les bits d'indice 0 des mots de 32 bits. Cette organisation peut

être retrouvée également par l'injection laser de fautes pendant la lecture des données comme décrit dans la sous-sous-section 3.2.2.d.

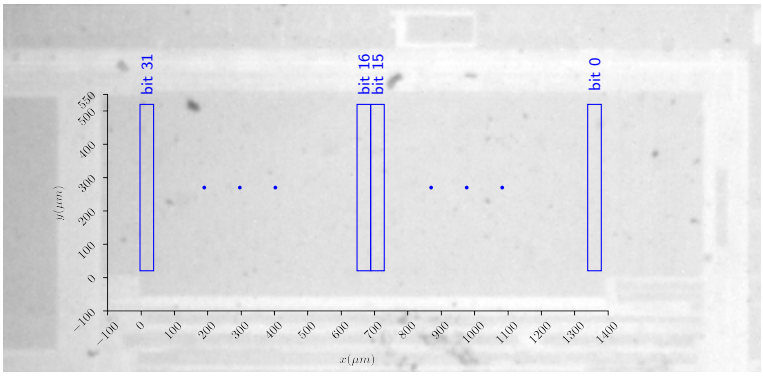


Figure 5.8 – Ingéniérie inverse de l'organisation de la mémoire Flash au niveau bit.

Dans un second temps, la documentation du composant nous informe que la mémoire Flash, d'une taille totale de 128 kB, est organisée en 128 pages de 1 kB chacune. La Figure 5.9 illustre la disposition des pages au sein de la mémoire. Ainsi, la première page d'indice 0, contenant l'ensemble des données ayant une adresse comprise entre 0x08000000 et 0x080003FF, se trouve en bas de la mémoire alors que la dernière page d'indice 127 se trouve en haut de la mémoire. L'injection laser de fautes pendant la lecture des données ne permet pas de retrouver cette organisation. En effet, dans ce scénario la faute injectée est indépendante de la position du laser selon la largeur de la mémoire Flash car c'est la même *bitline* qui est fautée.

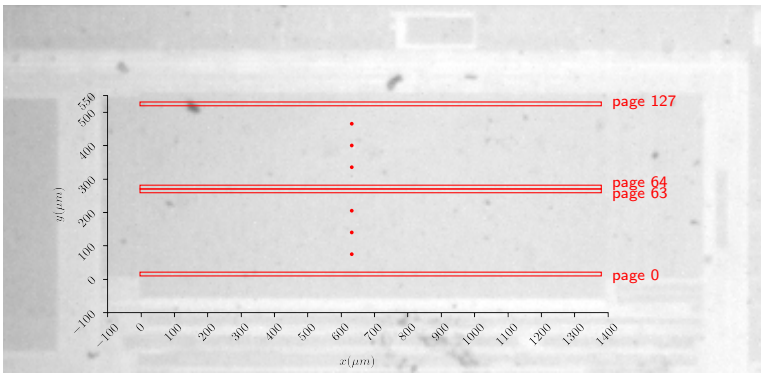


Figure 5.9 – Ingéniérie inverse de l'organisation de la mémoire Flash au niveau page.

Dans un dernier temps, l'organisation des mots et des bits au sein d'une page peut également être retrouvée. On note $w_{k\in[\![0,255]\!]}$ les 256 mots de 32 bits stockés dans une

page de la mémoire et $b_{i,j}$ le $i^{ème}$ bit du $j^{ème}$ mot de la page. Cette organisation est illustrée en Figure 5.10. Spatialement, pour un indice de bit soit une colonne, il y a 64 transistors à grille flottante par ligne soit 64 sous-*bitlines* en parallèle, et une page comporte 4 lignes horizontales de mots.

bit 31				bit 30					bit 0			
$b_{31,0}$	$b_{31,1}$	⋯	$b_{31,64}$	$b_{30,0}$	$b_{30,1}$	⋯	$b_{30,64}$		$b_{0,0}$	$b_{0,1}$	⋯	$b_{0,64}$
$b_{31,65}$	⋯	⋯	$b_{31,127}$	$b_{30,65}$	⋯	⋯	$b_{30,127}$	⋯	$b_{0,65}$	⋯	⋯	$b_{0,127}$
$b_{31,128}$	⋯	⋯	$b_{31,191}$	$b_{30,128}$	⋯	⋯	$b_{30,191}$		$b_{0,128}$	⋯	⋯	$b_{0,191}$
$b_{31,192}$	⋯	⋯	$b_{31,255}$	$b_{30,192}$	⋯	⋯	$b_{30,255}$		$b_{0,192}$	⋯	⋯	$b_{0,255}$

Figure 5.10 – Position physique des mots et des bits au sein d'une page de la mémoire. $b_{i,j}$ étant le $i^{ème}$ bit du $j^{ème}$ mot de la page.

Ainsi, nous avons montré que l'injection laser de fautes au sein de mémoires Flash non alimentées permet de réaliser l'ingénierie inverse de l'organisation de cette mémoire. Un attaquant peut ainsi, connaissant l'adresse d'une donnée stockée en mémoire, connaître la localisation exacte de cette donnée en mémoire jusqu'au niveau bit.

5.3.3.b Synthèse

Dans cette partie, nous avons démontré que des tirs laser successifs pouvaient corrompre le contenu d'une mémoire Flash d'un microcontrôleur non alimenté. Les résultats obtenus sont conformes au modèle de faute proposé : une décharge des transistors à grille flottante due à l'élévation de température causée par le laser. En effet, nous avons bien observé des fautes persistantes, c'est-à-dire présentes jusqu'à une nouvelle programmation du composant, de type *bitsets*. Ces résultats nous permettent également de réaliser l'ingénierie inverse de la mémoire Flash afin de retrouver la position physique de chaque bit d'information stocké. Nous allons maintenant voir dans la suite de ce chapitre comment il est possible d'exploiter ces résultats dans un scénario d'attaque de type PFA afin de retrouver la clé d'un AES 128 bits.

5.4 Application

Cette section s'intéresse à l'application du modèle de faute décrit précédemment afin de réaliser une Analyse de Faute Persistante (PFA, voir Chapitre 2). Cette analyse peut s'appliquer sur tous les algorithmes de chiffrement symétrique qui utilisent une S-Box bijective pour réaliser la couche de substitution. La démonstration de cette attaque sera faite sur l'algorithme AES.

Comme expliqué dans le Chapitre 2, cette analyse repose sur l'injection d'une faute persistante dans la S-Box de l'algorithme afin d'en soustraire l'aspect bijectif et d'insérer

un biais dans la distribution des octets des messages chiffrés. L'attaquant peut ainsi extraire de l'information sur la clé de chiffrement utilisée.

5.4.1 Implémentation de l'attaque

La cible matérielle utilisée dans cette attaque est la même que celle décrite dans la sous-section 4.4.2. Nous avons choisi d'utiliser une implémentation logicielle d'AES : *tiny-AES* avec une clé sur 128 bits[1]. Cette version n'occupe que très peu de place en mémoire et est donc intéressante pour les applications IoT auxquelles la cible matérielle est dédiée. En suivant les mêmes hypothèses d'attaque que `Zhang` *et al.* [Zha+18], on considère que la S-Box n'est pas fautée lors de l'expansion de clé.

La S-Box de l'AES est composée de 256 octets. Une page de la mémoire Flash du composant choisi contient 1 kB. Ainsi la S-Box occupe un quart d'une page. Ce quart de page est une *wordline* si les données sont alignées en mémoire. Sinon, on peut retrouver la moitié de la S-Box sur une *wordline* et la seconde moitié sur la *wordline* suivante. Ainsi, par simplification, on s'assure de l'alignement de la S-Box en mémoire lors de la compilation du *firmware* en ajoutant un attribut à la déclaration de la variable contenant la S-Box dans le code source `C` comme illustré dans la Figure 5.11. Une *worldline* est divisée en 32 colonnes et chaque colonne contient 64 bits. L'implémentation physique en mémoire Flash de la S-Box est visible en Figure 5.12.

```
1 AES_CONST_VAR uint8_t __attribute__((aligned)) sbox[256] =   {
2   0x63, 0x7c, 0x77, 0x7b, ... , 0x54, 0xbb, 0x16 };
```

Figure 5.11 – Définition de la S-Box alignée en mémoire dans le code source `C`.

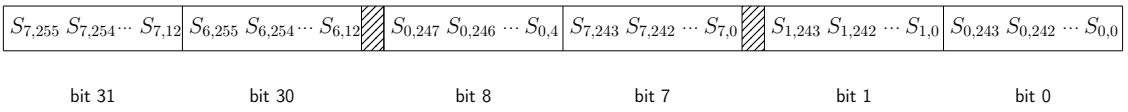


Figure 5.12 – Implémentation physique de la S-Box en mémoire Flash. $S_{i,j}$ est le $i^{ème}$ bit du $j^{ème}$ octet de la S-box.

On peut ainsi constater que les 256 octets de la S-Box sont organisés sous forme de mots de 32 bits correspondant à la concaténation de 4 octets en suivant le schéma décrit en Figure 5.13.

Plus spécifiquement, les cellules mémoire les plus à droite de chaque colonne stockent les 32 bits du premier mot w_0, les secondes cellules mémoires les plus à droite de chaque colonne stockent les 32 bits du mot suivant w_1 et ce processus est répété jusqu'au stockage des 32 bits du dernier mot de la S-Box w_{63}. Cette caractéristique nous permet de

1. https ://github.com/kokke/tiny-AES-c

$$\underbrace{(S_{*,255}S_{*,251}S_{*,247}S_{*,243})}_{w_{63}}\ldots S_{*,3}S_{*,254}S_{*,250}\ldots S_{*,2}S_{*,253}\ldots S_{*,1}S_{*,252}\ldots\underbrace{(S_{*,12}S_{*,8}S_{*,4}S_{*,0})}_{w_0}.$$

Figure 5.13 – Organisation sous forme de mots de 32 bits de la S-Box.

savoir que si de multiples bits sont fautés lors d'une injection, ces bits sont nécessairement stockés de façon adjacente horizontalement les uns des autres et ces fautes concernent des octets différents de la S-Box.

Dans le but d'injecter une faute dans la S-Box, un attaquant doit connaître la position physique de la S-Box en mémoire Flash. En suivant la même démarche d'ingénierie inverse que celle décrite dans la sous-sous-section 5.3.3.a sur un composant *clone*[2], cet attaquant est capable de retrouver la position physique à partir de l'adresse logique. Dans cette étude, nous faisons l'hypothèse que le *firmware* est connu de l'attaquant, il connaît donc l'adresse logique de la S-Box en mémoire Flash. Dans le cas contraire, il existe des méthodes d'*extraction de firmware* [VOC18 ; Gao+19 ; BH22].

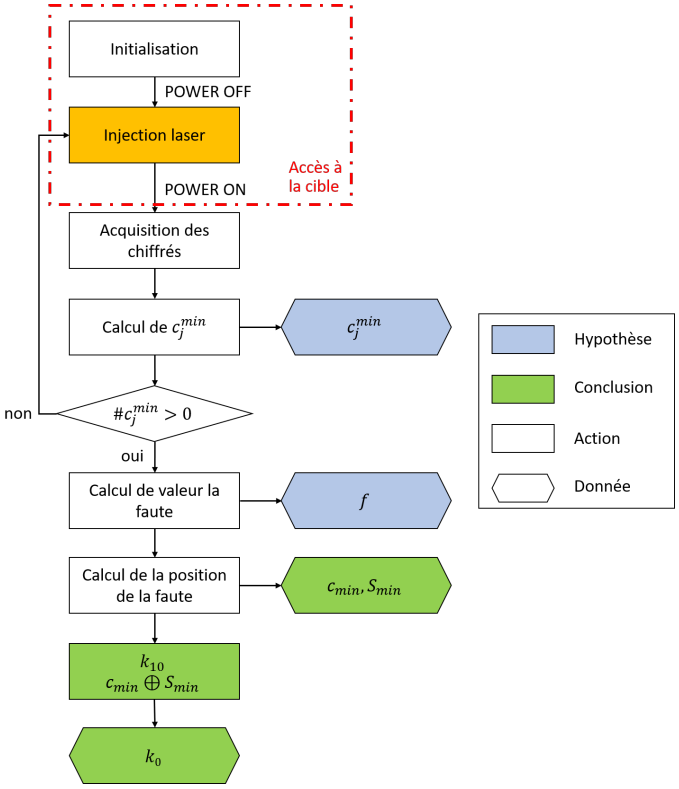


Figure 5.14 – Protocole expérimental de l'attaque.

Le protocole expérimental suivi pour réaliser la PFA est décrit en Figure 5.14. Après une phase initiale (programmation du composant, initialisation de la communication série

2. On appelle composant clone, un composant similaire sur lequel un attaquant possède les droits de lecture et d'écriture.

nécessaire à l'échange des messages clairs et des messages chiffrés avec la cible, etc.), l'alimentation de la cible est éteinte. Une série de 1 000 tirs laser est alors effectuée sur la position physique où se trouve la S-Box en mémoire Flash. La cible est ensuite alimentée de nouveau et 3 000 chiffrements sont effectués. Ce processus est répété jusqu'à trouver une ou plusieurs valeurs d'octet c_j^{min} n'apparaissant plus dans les textes chiffrés indiquant la corruption de la S-Box. Une fois que cette étape est atteinte, on peut calculer la valeur de la faute f ainsi que sa position i. Ces données sont nécessaires pour estimer la valeur S_{min}. Les combinaisons des c_j^{min} nous donne le chiffré c_{min} et, pour finir, la dernière clé de ronde peut être calculée à partir de S_{min} et c_{min}. La clé principale de l'algorithme de chiffrement AES est retrouvée en réalisant l'inverse de l'expansion de clé à partir de la dernière clé de ronde. Une description complète du fonctionnement de la PFA est disponible en section 2.7.

5.4.2 Modèle d'attaquant en pratique

La PFA peut être réalisée de façon *hors ligne*, c'est-à-dire qu'elle nécessite d'avoir un accès physique uniquement pendant l'injection de faute et non pendant les opérations de chiffrement. L'attaquant doit uniquement connaître les messages chiffrés, qui sont des données publiques, c'est un avantage significatif de cette attaque.

Tout d'abord, les messages à chiffrer sont choisis de façon aléatoire. Ce choix nous place au plus près d'un modèle d'attaquant réaliste. En effet, ce scénario traduit le cas où l'attaquant n'a ni contrôle ni accès aux textes clairs. Il existe d'autres versions de la PFA dans lesquels l'attaquant peut choisir les textes clairs [Sol+22]. Dans ce cas, en fixant tous les octets du texte clair sauf le premier et en faisant varier ce premier octet de 0 à 255, un octet de clé peut être retrouvé en, au plus, 256 chiffrements [Zha+23]. Ainsi, il est nécessaire d'effectuer $16 \times 256 = 4096$ chiffrements pour retrouver l'entièreté de la clé.

Le protocole décrit en Figure 5.14 illustre une analyse divisée en trois étapes.

La première étape est d'identifier c_j^{min}. Pour cette étape, présentée par l'Algorithme 2, les 3 000 messages chiffrés sont analysés et le nombre d'apparitions de chaque valeur pour l'ensemble des 16 octets sont mémorisés. Cet algorithme fournit une liste, notée c_j^{min}, contenant pour chacun des 16 octets la ou les valeurs n'apparaissant pas dans les messages chiffrés. Si cette liste est vide, cela signifie qu'aucune faute n'a été injectée dans la S-Box (les 3 000 chiffrements nous garantissent ce postulat). Il faut donc répéter l'étape d'injection de faute. Sinon, le nombre de valeurs contenues dans la liste pour chaque octet nous indique le nombre d'octets fautés dans la S-Box.

Algorithme 2 Identification de c_j^{min}.

Require: 3000 messages chiffrés (C_{LIST})
Ensure: Liste des valeurs impossibles pour chaque octet (c_j^{min})

```
c_j ← [[0,...,0],...,[0,...,0]]          ▷ 16 listes de 256 valeurs nulles
for c ∈ C_LIST do
    for j ∈ [0,...,15] do
        c_j[j][c[j]] ← c_j[j][c[j]] + 1   ▷ +1 au # d'apparition pour la valeur d'octet
c_j^min ← [[ ],...,[ ]]                  ▷ 16 listes vides
for j ∈ [0,...,15] do
    for i ∈ [0,...,256] do
        if c_j[j][i] = 0 then
            c_j^min[j].append[i]
return c_j^min
```

La seconde étape est le calcul de la valeur de la faute. Pour chaque octet des messages chiffrés, on calcule les valeurs les plus probables. Dans le cas d'une S-Box contenant n fautes, cela représente n valeurs pour c_j^{min} et n valeurs pour c_j^{max} pour chaque octet. Les candidats pour la faute sont les résultats des opérations OU-exclusif pour chaque pair (c_j^{min},c_j^{max}). En revanche, comme les textes clairs sont aléatoires, les valeurs apparaissant le plus souvent ne sont pas forcément liées à la valeur fautée dans la S-Box (c_j^{max} n'est pas toujours en relation avec S_{max} si le nombre de chiffrements réalisés n'est pas suffisant). Ce problème est résolu par le fait que la faute est la même pour l'ensemble des octets des messages chiffrés. En pratique, il y a $16 \times n^2$ candidats pour la valeur de la faute (n^2 pour chaque octet) et la valeur réelle de la faute est le candidat apparaissant le plus souvent.

La dernière étape est le calcul de la position de la faute. L'objectif ici est de retrouver la valeur en sortie de S-Box qui n'apparait jamais et le message chiffré associé à cette valeur fautée. En calculant ces deux informations, un attaquant peut retrouver la dernière clé de ronde k_{10}. Cette étape est illustrée par l'Algorithme 3.

5.4.3 Amélioration de la PFA

La connaissance du modèle de faute décrit dans en section 5.2 et section 5.3 nous permet d'appliquer la PFA de façon plus efficace en y apportant quelques modifications.

La première amélioration exploite le caractère unidirectionnel du modèle de faute. En effet, on sait que seules des fautes de type *bitset* peuvent être injectées. L'hypothèse sur l'index de l'octet fauté est initialement faite parmi les 256 valeurs candidates. Avec le modèle de faute décrit en section 5.2, nous obtenons la condition sur s_{max} donnée dans l'Équation 5.5 et la définition de s_{max} est rappelée dans l'Équation 5.6. De plus,

Algorithme 3 Calcul de S_{min}.

Require: Liste des c_j^{min}, liste des candidats S_{min}, liste des messages chiffrés $C_{\texttt{LIST}}$
Ensure: La position de la faute s_{min} et le message chiffré impossible associé c_{min}

```
for s_min ∈ S_min do                          ▷ Hypothèse sur la position de la faute
    for c_min ∈ c_j^min do                    ▷ Hypothèse sur les combinaisons de c_j^min
        k_10 ← s_min ⊕ c_min
        k_9 ← reverse_key_schedule_k9(k_10)
        for c_0 ∈ C_LIST do
            c_1 ← SR^-1(MC^-1(ADR(SB^-1(SR^-1(ADR(c_0, k_10))), k_9)))
            if s_min ∈ c_1 then
                next s_min
        return k_10      ▷ La bonne paire (c_min,s_min) nous donne la clé de la dernière ronde k_10
```

cette amélioration est liée au nombre de fautes injectées. Ainsi, le nombre de valeurs impossibles pour chaque octet des messages chiffrés est égal au nombre d'octets fautés dans la S-Box. L'organisation de la mémoire Flash implique que si plusieurs octets sont fautés, ces fautes concernent obligatoirement des bits de même indice dans des octets différents de la S-Box. Cette dernière est stockée de telle manière que deux transistors à grille flottante consécutifs stockent le même bit de deux valeurs de S-Box distantes de 4 octets (voir Figure 5.12). À cet instant de l'analyse, la valeur de la faute et la position de faute sont connues. Ainsi, certains candidats pour S_{min}, notés s_{min}, peuvent être éliminés. En effet, pour n fautes injectées, si un candidat est valide, les $n-1$ candidats restants doivent être dans le voisinage, au sens de la position physique en mémoire, du candidat valide. Sinon, le candidat est éliminé.

$$s_{max} = s_{min} \vee f \tag{5.5}$$

$$s_{max} = s_{min} \oplus f \tag{5.6}$$

En pratique, on compare les résultats des opérations OU et OU-exclusif entre la valeur de la faute et le candidat pour S_{min} (grâce à l'Équation 5.5 et à l'Équation 5.6). Si les deux opérations donnent le même résultat, le candidat est considéré comme valide. Cette méthode est appliquée à l'ensemble des valeurs de la S-Box afin d'éliminer tous les candidats invalides. Ensuite, avec le nombre connu n de fautes injectées, on vérifie qu'un candidat est entouré de $n-1$ candidats valides. Sinon, le candidat est éliminé. La Tableau 5.2 présente le nombre de candidats éliminés par cette méthode en fonction du nombre de fautes injectées. Cette amélioration nous permet d'appliquer l'analyse sur

moins de 50 % des valeurs de S-Box, résultant en une durée d'exécution de l'analyse au moins deux fois plus courte.

Nombres de fautes injectées	Candidats restants
1	50 %
2	36 %
3	25 %
4	14 %

Tableau 5.2 – Nombre de candidats restants en fonction du nombre de fautes injectées.

La seconde amélioration est basée sur l'élimination de paires de candidats (c_{min},S_{min}). Dans la version initiale de la PFA proposée dans [Zha+18], les auteurs éliminent une valeur S_{min} lorsque 256 valeurs d'octets différentes sont obtenues en sortie de l'opération `SubBytes` de l'avant-dernière ronde de l'AES. Nous proposons de chercher l'apparition du candidat S_{min} dans les octets de sortie de l'opération `SubBytes` de l'avant-dernière ronde. Ainsi, nous n'attendons plus 256 valeurs différentes mais une seule valeur précise. Cette amélioration permet d'éliminer un candidat en moyenne environ 6 fois plus rapidement que dans la version initiale.

5.4.4 Résultats expérimentaux

En étudiant les fichiers issus de la compilation (*.map* ou *.hex*) ou de l'extraction du *firmware*, on retrouve l'adresse logique de la S-Box. Ainsi cette dernière est stockée dans la plage d'adresses suivante : de `0x080012F4` à `0x080013F3`. Cette plage d'adresse se trouve dans page d'indice 4 de la mémoire Flash. D'après l'ingénierie inverse précédemment réalisée, la position physique de cette plage d'adresse est connue. Pour trouver la position exacte, on fixe l'origine de notre repère XY dans le coin supérieur droit de la mémoire Flash. On réalise alors une petite cartographie d'injection de faute laser autour de la position sur un composant clone afin de préciser la position du laser. Nous avons obtenu la faute `0x00000002` à l'adresse `0x08001310` pour la position $(x, y) = (44{,}3; 300)$. Cette adresse est bel et bien dans la plage d'adresse de la S-Box.

Le composant cible peut désormais être attaqué en plaçant le spot laser à la même position que sur le composant clone. L'injection de faute dans la S-Box est réalisée par deux séries de 1 000 tirs laser. Le composant est ensuite alimenté de nouveau et 3 000 chiffrements sont effectués. Dans cet exemple, la $32^{ème}$ valeur de la S-Box est fautée. En effet, la valeur `0xC2` remplace la valeur `0xC0`, ce qui signifie que la valeur de la faute est `0x02`. Le calcul de k_{10} et l'expansion inverse de la clé pour retrouver la clé principale sont instantanés car un seul octet de la S-Box est fauté.

Si deux octets de la S-Box sont fautés, le déroulement complet de l'attaque est réalisé en moins d'une heure dans le pire cas. Le pire cas est celui où les octets fautés de la S-Box sont à la fin de celle-ci. Le temps d'exécution de l'algorithme complet est obtenu par simulation. En effet, lors des attaques expérimentales, nous n'avons obtenu que des fautes sur un unique octet de la S-Box.

5.5 Discussion

Dans cette étude, nous avons présenté un nouveau vecteur d'attaque. L'arrivée d'un nouveau vecteur d'attaque pose inévitablement la question des contremesures associées.

Nous nous sommes concentrés sur la mémoire Flash d'un microcontrôleur du marché conçu pour des applications IoT. Ainsi, ce composant n'embarque aucun dispositif de protection de la mémoire. Dans le cas d'une attaque d'un composant dit *sécurisé*, il est courant de trouver des dispositifs de protection tels que des *codes détecteurs ou correcteurs d'erreurs*, du *scrambling* ou un *chiffrement* de la mémoire. Ces protections sont efficaces contre l'attaque proposée. En revanche, dans le contexte de l'IoT, où les coûts de conception et fabrication sont un enjeu important, ces dispositifs de protection ne sont pas forcément implémentés en plus des algorithmes de chiffrement. Par exemple, le composant que nous avons choisi, le `STM32F1` du fabricant `STMicroelectronics`, est conçu pour l'IoT et ne contient aucune protection. Il est ainsi recommandé que l'étude réalisée dans ce chapitre soit améliorée dans le cas d'un composant embarquant certains mécanismes de protection de la mémoire. Par conséquent, par ce nouveau vecteur d'attaque, nous incitons fortement les fabricants de semi-conducteurs et les concepteurs de *firmware* à inclure des protections.

Jusqu'aujourd'hui très peu de contremesures spécifiques à la PFA ont été proposées. L'une d'entre elles [Zha+20] consiste à stocker des paires (message clair, message chiffré) de référence et à vérifier l'exactitude des messages chiffrés périodiquement. Cette méthode permet de vérifier l'intégrité de l'algorithme et donc de la S-Box. Il est également possible d'implémenter d'autres contremesures propres à cette analyse. Par exemple, `Tissot` *et al.* [TBG23] proposent une méthode de vérification de l'intégrité de la S-Box en analysant les cycles qu'elle contient. Pour conclure, cette étude renforce l'intérêt de la PFA en donnant un exemple de cas d'attaque pratique et encourage la communauté de la sécurité matérielle à proposer des contremesures efficaces et peu coûteuses.

5.6 Conclusion

Dans ce chapitre, nous avons démontré pour la première fois la possibilité d'injecter des fautes par laser au sein de composants non alimentés. Il a été montré que l'utilisation d'un spot laser de 5 µm de diamètre permet d'injecter des fautes en mémoires Flash avec une précision au niveau du bit. Nous avons obtenu un modèle de faute de type *bitset* et avons proposé un modèle de faute complet, allant du niveau physique au niveau applicatif en passant par les niveaux logique et mémoire. Un effet thermique, dû à l'exposition laser, est à l'origine de ce modèle de faute. En effet, en chauffant localement les transistors à grille flottante qui composent la mémoire Flash, les électrons stockés obtiennent assez d'énergie pour s'échapper de la grille flottante. Ainsi, l'injection laser de fautes au sein de la mémoire Flash d'un composant non alimenté est caractérisée par une décharge des transistors à grille flottante et un modèle de faute unidirectionnel. Les fautes obtenues sont persistantes et non destructives.

Ce chapitre démontre également qu'une attaque, la PFA, peut être réalisée avec ce modèle de faute et un modèle d'attaquant très réaliste. Cette attaque nous a permis de retrouver la clé 128 bits de l'algorithme de chiffrement AES.

Il existe d'autres scénarios d'attaques potentiels à explorer. En utilisant cette nouvelle technique d'injection de faute, la corruption d'un *firmware* ou la modification de droits d'accès peuvent représenter des challenges intéressants à relever. De plus, d'autres algorithmes de chiffrement ou d'autres cibles matérielles peuvent également être visés.

Ce chapitre a fait l'objet d'une publication en revue internationale :

- `P. GRANDAMME, P.-A. TISSOT, L. BOSSUET` *et al.*, "Switching Off your Device Does Not Protect Against Fault Attacks", TCHES 2024. [Gra+24]

Chapitre 6

Injection par rayons X de fautes sur circuit non alimenté

Table des matières

6.1 Introduction

Les chapitres précédents ont décrit l'utilisation de sources laser dans le but d'injecter des fautes au sein de mémoires Flash embarquées dans des microcontrôleurs.

De précédents travaux, décrits dans la section 3.3 du Chapitre 3, ont déjà étudiés l'utilisation des rayons X dans le but d'injecter des fautes au sein de circuits intégrés. L'ensemble de ces résultats ont été obtenu sur des circuits alimentés, c'est pourquoi dans ce chapitre on s'intéresse à l'utilisation d'une source de rayons X pour corrompre le contenu d'une mémoire Flash non alimentée.

Certains travaux décrits dans ce chapitre ont fait l'objet d'une contribution à la conférence internationale PAINE en 2023 [GBD23].

6.2 Présentation du dispositif expérimental d'irradiation X

6.2.1 Généralités sur les sources de rayons X

Les tubes à rayons X représentent le moyen de production de rayons X le plus courant. Ils possèdent de nombreuses applications comme l'imagerie médicale, la caractérisation de matériaux, la stérilisation et l'évaluation de la fiabilité des composants électroniques en environnement radiatif.

Une illustration d'un tube à rayons X est visible en Figure 6.1. Ces tubes sont composés d'une cathode, d'un filament au travers duquel circule un courant électrique I, et d'une anode, typiquement un matériau avec un haut numéro atomique comme le tungstène (W). L'ensemble est placé dans un environnement sous vide. Une forte tension électrique V (de l'ordre de la dizaine ou de la centaine de kilovolts) est appliquée entre la cathode et l'anode alors qu'un courant I (de l'ordre de l'ampère) traverse la cathode. Ce courant cause une élévation de température du filament et l'émission d'électrons par effet thermoélectrique. Le courant électrique i ainsi créé est appelé *courant de tube* et est de l'ordre du milliampère (avec 1 mA $\approx 6.24 \times 10^{15}$ électrons/s).

La grande différence de potentiel V appliquée entre l'anode et la cathode génère un champ électrique important. Ce champ électrique accélère les électrons libérés avec une énergie $E = qV$ (avec $q = 1.602176634 \times 10^{-19}$ C la charge élémentaire). En exprimant cette énergie E en électronvolt, cette dernière possède la même valeur numérique que V. Ainsi, une différence de potentiel de 100 kV équivaut à une énergie de 100 keV pour chaque électron.

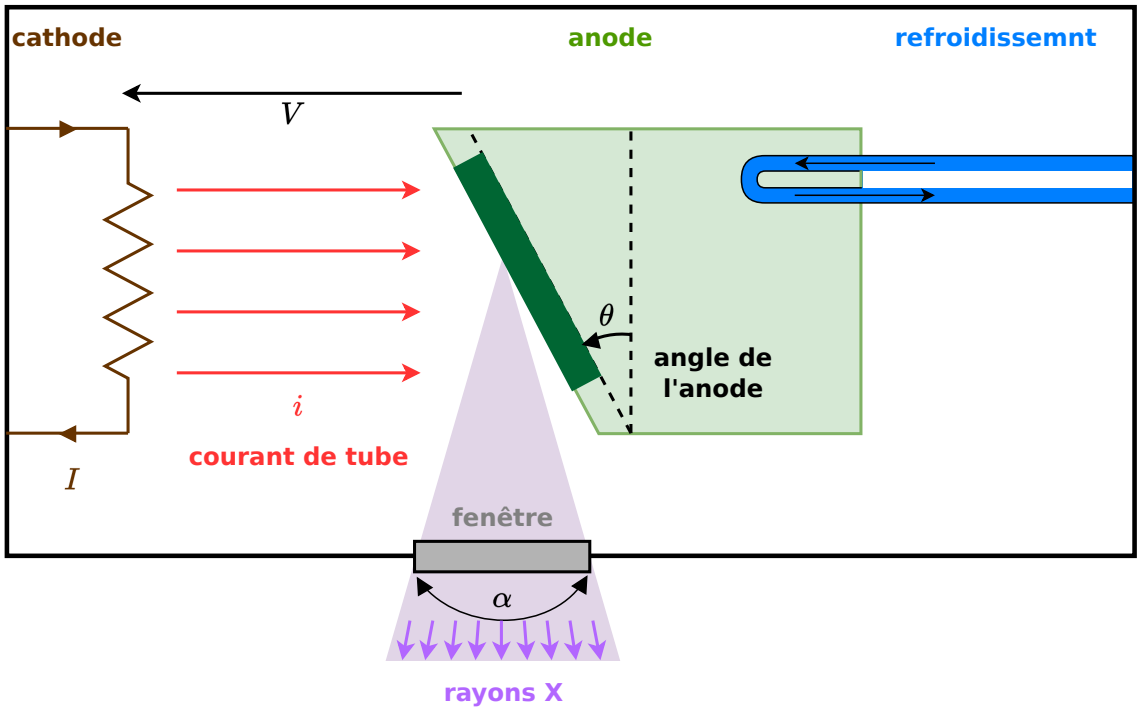


Figure 6.1 – Illustration de la composition d'un tube X. Adaptée de [Mey23].

Ces électrons hautement énergétiques interagissent avec le matériau constituant l'anode et produisent des rayons X secondaires au travers deux processus : le *bremsstrahlung* et l'émission caractéristique qui sont détaillés ci-après.

Bremsstrahlung ou radiation de freinage

Le *bremsstrahlung* est issu de la déviation de la trajectoire de l'électron incident causée par une interaction de Coulomb avec le champ électrique du noyau atomique du matériau constituant l'anode. L'électron dévié est décéléré par le champ électrique et la différence d'énergie est émise sous la forme d'un flux de photons [Nak94]. Le spectre d'énergie produit est continu avec un maximum équivalent à l'énergie de l'électron incident. La probabilité d'occurrence de cet effet est proportionnel au carré du numéro atomique Z de la cible, c'est pourquoi des matériaux avec un haut numéro atomique sont souvent choisi pour l'anode [Sei04].

Émission caractéristique

L'émission caractéristique est un phénomène de fluorescence causé par la libération d'électrons de cœur du matériau de l'anode. Les électrons de cœur de l'anode sont remplacés par des électrons du faisceau. Ces derniers possèdent une énergie de liaison plus faible que les électrons initiaux, ainsi la différence d'énergie est compensée par l'émission de photons avec une énergie égale à la différence des énergies de l'électron chassé

et de l'électron de remplacement. Le spectre d'énergie de l'émission caractéristique est composé de pics spécifiques au matériau de l'anode.

Ces deux processus contribuent à la conversion d'électrons en photons de rayons X. Leur efficacité globale est très faible à cause de la faible section transversale de ces interactions, ainsi l'efficacité des tubes à rayons X est souvent considérée inférieure à 1 % [Sei04].

Les rayons X produits possèdent une certaine particularité angulaire, c'est-à-dire qu'ils ne sont pas émis uniformément selon toutes les directions de l'espace. Alors qu'ils sont théoriquement émis dans toutes les directions, comme les interactions ont lieu dans l'anode qui possèdent une forte densité (matériau à haute atténuation), les rayons X s'échappent selon un certain angle pour lequel la distance à parcourir dans le matériau de l'anode est minimale. Ainsi, les rayons X sont émis selon un petit angle qui dépend de l'angle θ de l'anode [FG14].

L'environnement sous vide, généralement construit avec un matériau possédant une forte densité, est clos par une fenêtre qui possède à l'inverse un numéro atomique faible comme le béryllium (Be) positionnée sous l'anode permettant aux rayons X de s'échapper de l'environnement sous vide en limitant l'atténuation [Rog47].

À cause de la grande quantité de chaleur générée par l'anode lors de la production de rayons X, il est nécessaire d'ajouter un système de refroidissement afin de ne pas endommager l'anode et de prolonger sa durée de fonctionnement.

6.2.2 Matériel d'irradiation et cible

6.2.2.a Irradiateur IDfix

Le premier irradiateur, nommé IDfix, utilisé dans ces travaux, appartient à l'équipe MOPERE (Materials for Optics and Photonics in Extreme Radiation Environments) du laboratoire Hubert Curien de l'université Jean Monnet de Saint-Étienne. Ses caractéristiques sont présentées dans le Tableau 6.1. La Figure 6.2 montre l'enceinte en plomb (Pb) qui apporte une protection permettant de garantir des niveaux d'irradiations suffisamment faibles pour l'opérateur lorsque le tube fonctionne avec un courant et une tension maximums. Le passage de fils est possible grâce à deux ouvertures en forme de labyrinthe conçues pour limiter les fuites.

Tube à rayons X	COMET MXR-165 [COM21]
Tension maximale	160 kV
Courant maximal	45 mA
Matériau de l'anode	Tungstène (W)
Angle de l'anode	30°
Couverture du faisceau	50°
Filtrage du faisceau	4 mm béryllium (Be)

Tableau 6.1 – Caractéristiques de l'irradiateur IDfix.

Figure 6.2 – Irradiateur IDfix.

On appelle *débit de dose* la dose absorbée par un matériau par unité de temps. Cette grandeur dépend du matériau considéré. Dans cette étude, le matériau considéré est le dioxyde de silicium (SiO_2).

L'irradiateur offre une grande plage de débits de dose, typiquement de 500 µGy$_{(SiO_2)}$/s à 20 Gy$_{(SiO_2)}$/s. Le débit peut être ajusté en réglant plusieurs paramètres comme la distance entre la cible et l'irradiateur, le courant et la tension du tube et la position d'un éventuel bouclier. Plus spécifiquement, le réglage de la tension du tube permet de modifier le spectre d'énergie et donc l'énergie moyenne des photons générés. Le courant du tube permet de contrôler l'intensité des radiations produites.

Avant chaque irradiation, une dosimétrie est réalisée à position, courant et tension fixés afin d'assurer la bonne reproductibilité des expériences. Cette dosimétrie est faite en utilisant une chambre d'ionisation `PTW 23344` qui possède une zone de sensibilité de 3 cm de diamètre. Une chambre d'ionisation est un détecteur de particule qui mesure la charge totale des électrons et ions produits lors de l'ionisation du milieu.

Le spectre caractéristique du tube, obtenu par simulation `Python` avec la librairie `SpekPy`, est visible en Figure 6.3. Ce spectre prend en compte la fenêtre en béryllium. L'aspect continu de la courbe est dû au *bremsstrahlung* alors que les pics sont causés par l'émission caractéristique. Cette librairie nous permet également de calculer l'énergie-fluence (produit de l'énergie et de la fluence) moyenne des photons générées. Dans notre configuration, nous obtenons une énergie-fluence E_ϕ de 40 keV.

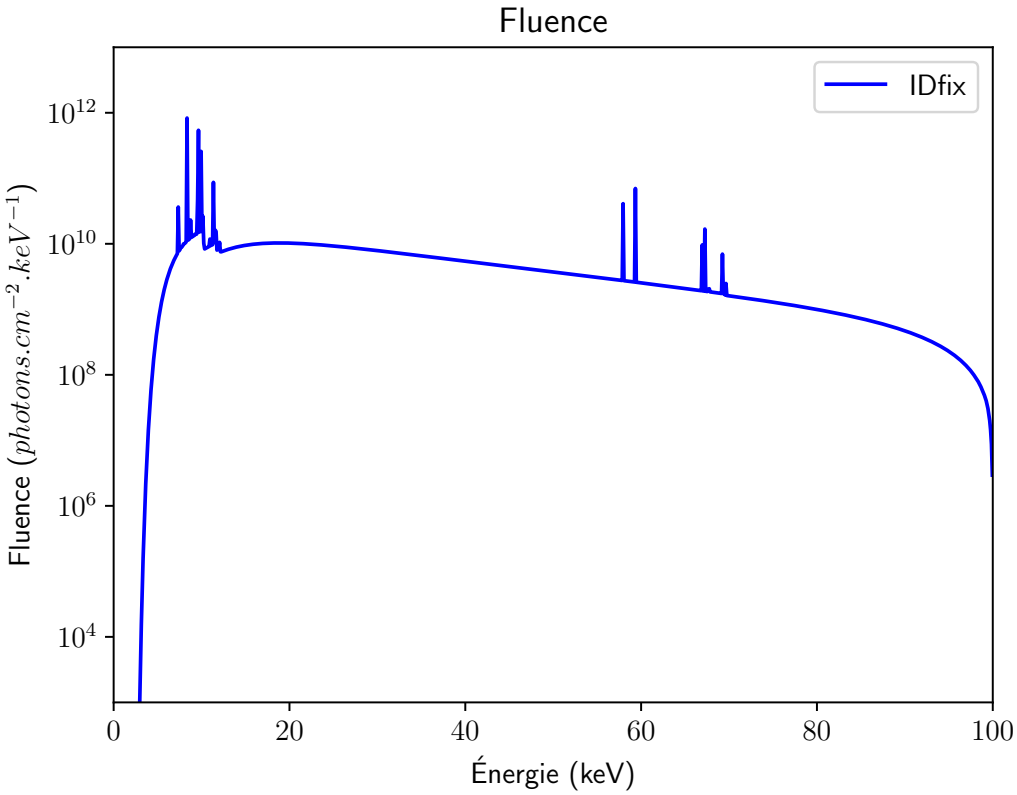


Figure 6.3 – Spectre simulé de l'irradiateur IDfix obtenu avec une tension de tube de 100 kV, un courant de 45 mA et une distance verticale de 25 cm.

6.2.2.b Cible matérielle

La cible matérielle est la même que décrit dans la sous-section 4.4.2, un microcontrôleur `STM32F100` embarquant 128 kB de mémoire Flash. Dans cette étude, il est important de noter la présence de bits de sécurité contrôlant les droits d'accès en lecture au contenu de la mémoire Flash [Stm]. Ce dispositif permet de protéger le code source ou les données stockées en mémoire. Pour ce composant, il existe deux niveaux de protection :

— Niveau 0 : Aucune restriction,
— Niveau 1 : Lecture de la mémoire Flash impossible lorsque le *debugger* est connecté.

Ces deux niveaux de protections sont contrôlés par la valeur des registres `RDP` et son complémentaire `nRDP`. Le Tableau 6.2 illustre le niveau de protection en lecture de la mémoire Flash selon les valeurs de ces registres. Avant chaque expérience, la mémoire Flash n'est pas protégée (RDP=`0xA5` et nRDP=`0x5A`). On peut ainsi voir que si un bit quelconque de l'un des deux registres est fauté, la protection passe en niveau 1.

RDP	nRDP	Statut	Niveau
0xFF	0xFF	Protégé	1
0xA5	0x5A	Non protégé	0
0xXY	$\neq \overline{\texttt{0xXY}}$	Protégé	1
0xXY	$\overline{\texttt{0xXY}}$	Non précisé dans la documentation	?

Tableau 6.2 – Statut de la protection de la mémoire Flash selon les valeurs de RDP et nRDP. 0xXY représente n'importe quelle valeur différente de 0xFF et 0xA5. [Stm]

Une diminution de la protection en lecture de la mémoire Flash de façon logicielle et légitime (c'est-à-dire un passage du niveau 1 vers le niveau 0) engendre un effacement complet du contenu de la mémoire Flash.

6.2.3 Protocole expérimental

Des campagnes d'irradiation préliminaires sur ces mémoires Flash non alimentées et initialisées complètement à 0x55555555 (pour un mot de 32 bit), permettant de tester deux modèles de faute (*bitsets* pour les bits initialement à '0' et *bitresets* pour les bits initialement à '1'), ont montré qu'uniquement des fautes de type *bitset* sont obtenues. Ces premiers résultats sont conformes à la décharge attendue par effet radiatif des grilles flottantes des transistors à grille flottante de la mémoire Flash.

C'est pourquoi nous avons choisi de suivre le protocole décrit en Figure 6.4. Dans un premier temps, la mémoire Flash est complètement remplie avec la valeur 0x00000000 au niveau d'un mot de 32 bits avant d'éteindre le composant. La cible est ensuite exposée aux radiations. La source de rayons X n'étant pas focalisée, la totalité du composant est irradiée. Le composant est ensuite alimenté de nouveau hors irradiation afin de relire le contenu de la mémoire Flash. Cette dernière est relue 5 fois après chaque irradiation pour évaluer l'instabilité de certaines fautes injectées si les registres RDP et nRDP le permettent. En effet, au fur et à mesure que des fautes sont injectées il devient de plus en plus probable de fauter l'un des transistors à grille flottante mémorisant les registres RDP et nRDP. Dès qu'une telle faute est injectée la mémoire Flash passe en mode sécurisé et il n'est plus possible de relire son contenu. En passant à nouveau dans le mode non sécurisé un effacement complet de la mémoire intervient, il n'est donc pas utile de poursuivre les expériences. Ces étapes sont répétées jusqu'à l'injection de fautes dans ces registres, forçant la fin de l'expérience.

La Figure 6.5 montre l'intérieur de l'enceinte en plomb dans laquelle se trouvent la source de rayons X et la cible. On peut y voir la source en haut et la cible située en dessous dont la face arrière est exposée aux radiations.

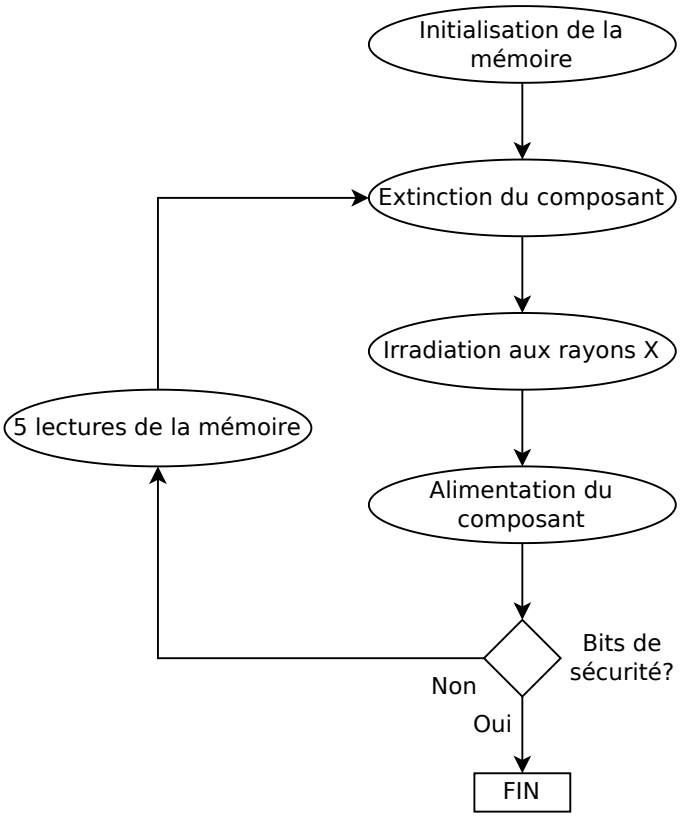


Figure 6.4 – Protocole expérimental suivi lors des campagnes d'irradiation aux rayons X.

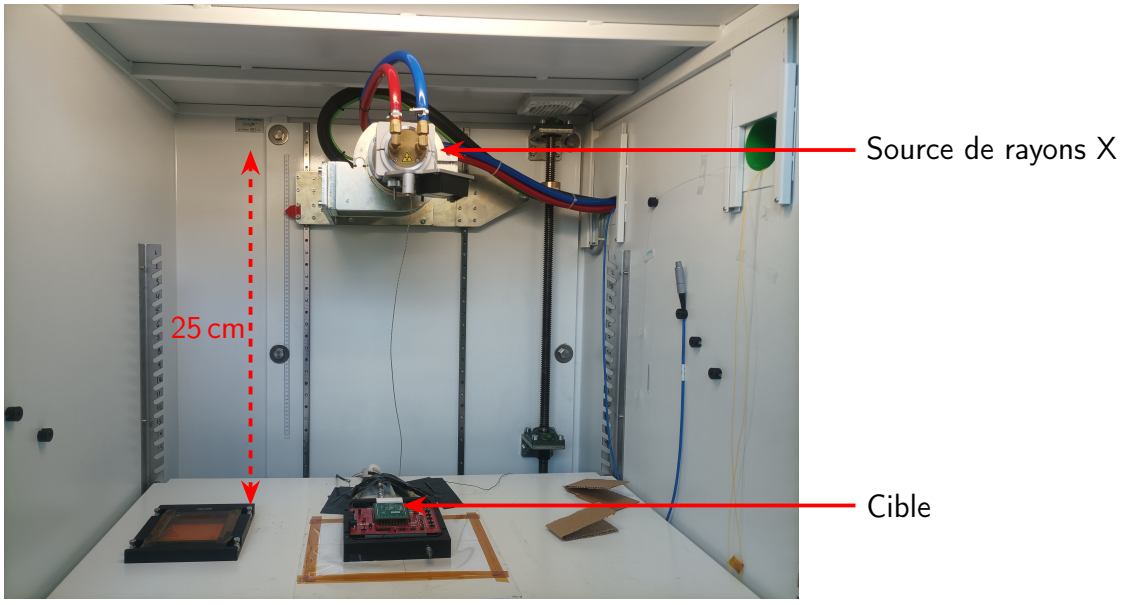


Figure 6.5 – Photographie de l'intérieur de l'enceinte en plomb avec la source de rayons X et la cible.

6.3 Caractérisation de l'exposition aux rayons X d'un microcontrôleur

6.3.1 Résultats expérimentaux des campagnes d'irradiation

Pour ces expériences, la tension du tube à rayons X est fixée à 100 kV et le courant à 45 mA.

L'ensemble des résultats décrit ci-après ont été obtenus en irradiant quatre composants du même type (`STM32F100`).

La distance verticale entre la source de rayons X et la cible est fixée à 25 cm afin d'avoir un débit de dose équivalent à 1 $Gy_{(SiO_2)}$/s. Dans un premier temps, les irradiations sont réalisées avec un pas de dose d'irradiation de 100 $Gy_{(SiO_2)}$ et lorsque les premières fautes apparaissent avec un pas de 25 $Gy_{(SiO_2)}$.

La Figure 6.6 montre l'évolution du nombre de fautes dans la mémoire Flash du composant en fonction de la dose totale. Chaque point bleu correspond à une relecture du contenu de la mémoire.

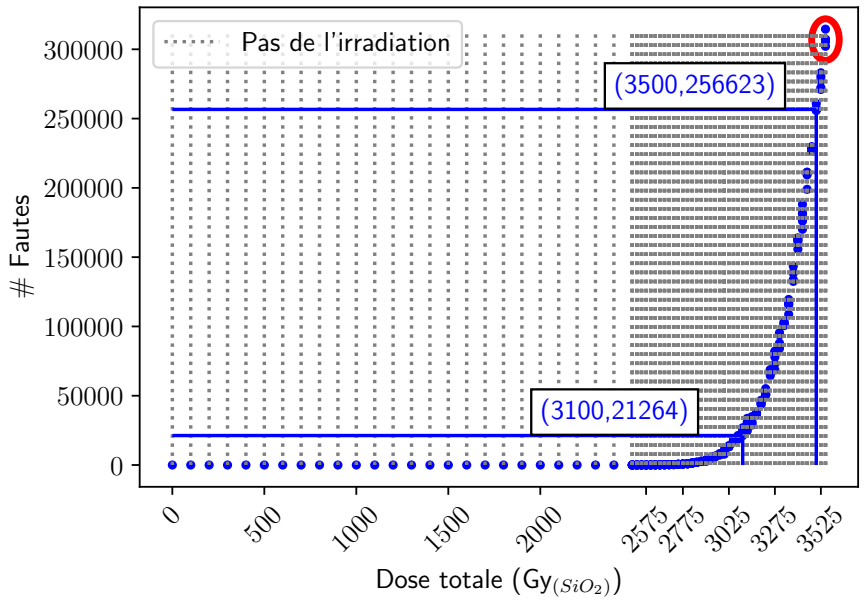


Figure 6.6 – Évolution du nombre de fautes en mémoire Flash pendant les irradiations aux rayons X. Chaque point bleu correspond à une lecture de la mémoire.

On peut observer que les premières fautes apparaissent aux alentours de 2 525 $Gy_{(SiO_2)}$. Une augmentation exponentielle du nombre de fautes en fonction de la dose déposée est ensuite observée. À la fin de l'expérience, environ 300 000 bits sont fautés, ce qui

correspond environ au tiers de la mémoire Flash. Les expériences sont arrêtées lorsqu'une faute est injectée dans l'un des registres `RDP` ou `nRDP` et donc que la relecture de la mémoire n'est plus possible.

Nous avons observé que pour une dose donnée, le nombre de fautes injectées n'est pas constant pour chacune des relectures de la mémoire Flash, il est cependant du même ordre de grandeur. Par exemple, sur la Figure 6.6, le cercle rouge entoure les 5 lectures qui correspondent à une dose de 3 525 $Gy_{(SiO_2)}$. Cela indique une instabilité des bits fautés.

Il y a en effet des fautes dites *permanentes*, c'est-à-dire qu'une fois que la faute est injectée elle est toujours présente lors de lectures successives, et des fautes dites *non permanentes*, c'est-à-dire que la faute est présente par intermittence lors de lectures successives de la mémoire. La Figure 6.7 illustre l'évolution du nombre de fautes permanentes et non permanentes. On peut observer que les fautes permanentes semblent converger vers une valeur maximale aux alentours de 46 000 bits fautés.

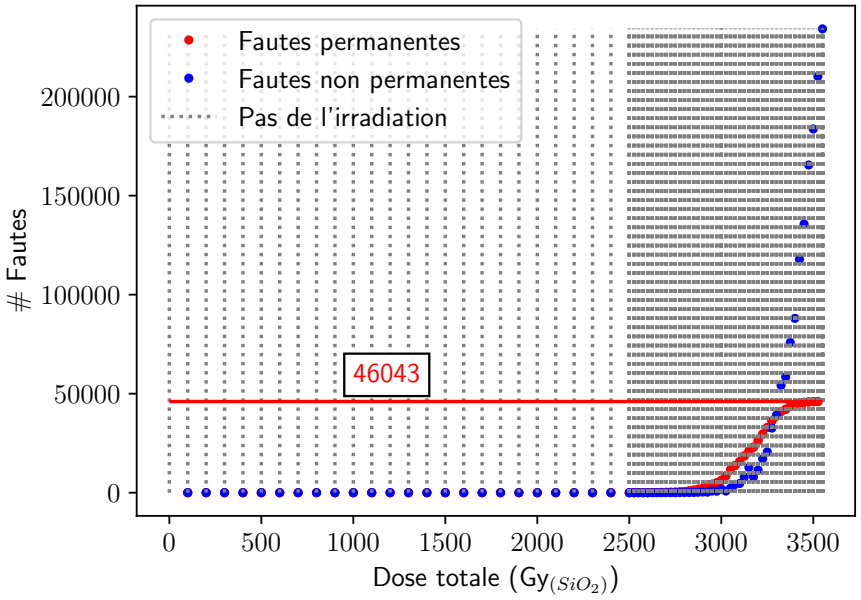


Figure 6.7 – Évolution du nombre de fautes permanentes en rouge et non permanentes en bleu en mémoire Flash pendant les irradiations aux rayons X.

La Figure 6.8 représente l'état de la mémoire Flash après le dépôt d'une dose de 3 525 $Gy_{(SiO_2)}$ par exposition aux rayons X. Chaque point représente un bit de la mémoire. Les bits fautés sont les points noirs alors que les bits non fautés sont les points blancs. La mémoire y est représentée selon les 2 048 *bitlines* en abscisse et les 512 *wordlines* en ordonnée. On observe bien un modèle de faute de type *bitset*.

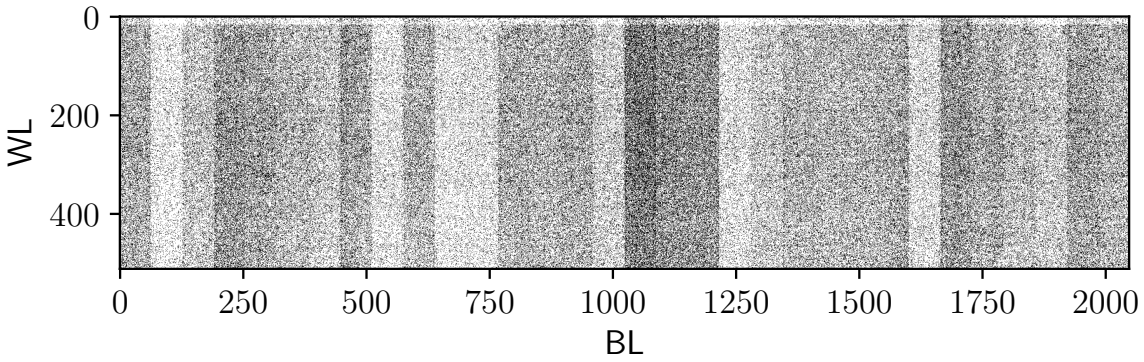


Figure 6.8 – État de la mémoire Flash à la fin des irradiations aux rayons X. Chaque point représente un bit d'information.

À l'issu des irradiations, on peut distinguer 32 bandes verticales. Ces bandes correspondent à l'organisation selon 32 bits des mots contenus dans la mémoire Flash. Ainsi, la première colonne (à gauche) contient l'ensemble des bits d'indice 31 des mots de 32 bits stockés en mémoire Flash et la dernière colonne (à droite) contient l'ensemble des bits d'indice 0 des mots de 32 bits. On peut également observer que les colonnes ne sont pas uniformément fautées. En effet, certaines bandes sont plus claires que d'autres, c'est-à-dire qu'elles contiennent plus de bits fautés que d'autres. L'irradiation est uniforme sur l'ensemble du composant, il devrait donc y avoir une densité de faute similaire sur l'ensemble de la mémoire Flash. Si l'apparition des fautes était uniquement dûe à la décharge des transistors à grille flottante, les fautes seraient uniformément réparties sur toutes les colonnes. Ce point nous indique que la logique de contrôle de la mémoire Flash, comme les décodeurs de ligne ou de colonne, ou la logique de lecture, comme les *sense amplifiers*, est également impactée par les irradiations. L'hypothèse la plus probable est qu'une dérive des tensions de seuil des transistors MOS des composants analogiques est responsable de ce phénomène. On observe bien une répartition uniforme des fautes au sein des colonnes de la mémoire.

Pour résumer, nous avons mis en évidence la possibilité d'injecter deux types de fautes en mémoire Flash de composants non alimentés. D'une part, les fautes non permanentes sont causées par un décalage de tension de seuil des transistors MOS des logiques d'accès et d'adressage aux points mémoire dû à un piégeage de charges dans les oxydes. D'autre part, les fautes permanentes sont causées par la décharge des électrons contenus dans la grille flottante des transistors à grille flottante qui composent la mémoire Flash. Quand l'effet cumulé entraîne une dérive importante de la tension de seuil des transistors, on obtient une faute permanente. À l'inverse, lorsqu'on est proche de la tension de lecture les fautes sont non permanentes. Ces effets constituent les effets de dose totale décrits en sous-sous-section 3.3.3.a.

Des résultats similaires ont été obtenus sur des mémoires EEPROM d'un autre microcontrôleur du même fabricant : le `STM32L151` embarquant 16 kB de mémoire EEPROM. L'évolution du nombre de fautes dans cette mémoire est visible en Figure 6.9.

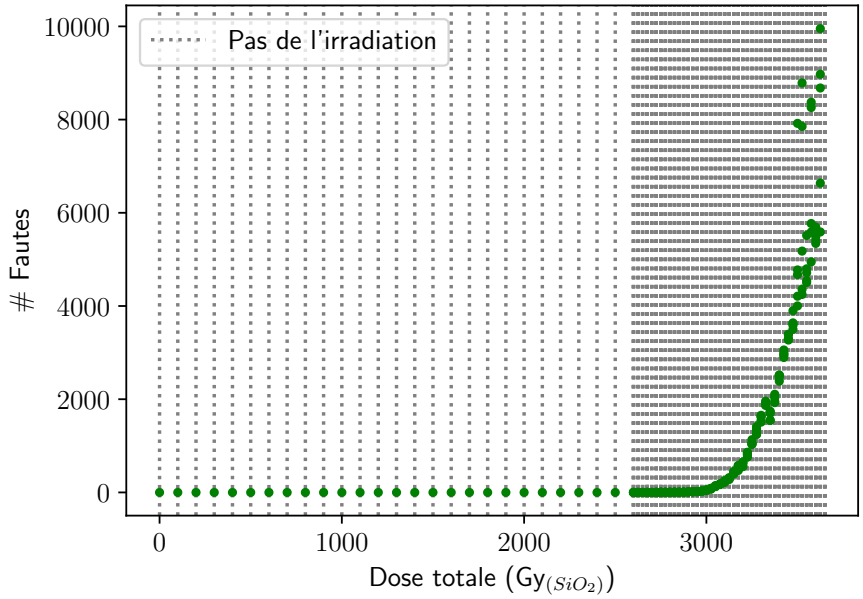


Figure 6.9 – Évolution du nombre de fautes en mémoire EEPROM pendant les irradiations aux rayons X. Chaque point vert correspond à une lecture de la mémoire.

6.3.2 Récupération temporelle et thermique

Certains effets des irradiations aux rayons X, notamment les dérives de tension de seuil des transistors dues aux charges piégées dans les oxydes, sont connus pour être récupérables avec le temps et la température. Pour évaluer cette récupération, l'un des circuits irradiés a été laissé au repos à température ambiante pendant plusieurs jours.

La Figure 6.10 montre l'état de la mémoire Flash après une semaine de récupération à température ambiante. On peut observer que l'image est globalement plus claire que celle générée juste après les irradiations, ce qui indique la disparition de nombreuses fautes. On peut également observer l'apparition de certaines lignes horizontales indiquant que la logique de contrôle des *wordlines* ou les *sense amplifiers* n'ont pas tous récupéré de manière équivalente.

Pour finir, une récupération thermique, ou un recuit, a également été réalisée en plaçant le composant dans une enceinte thermique pendant 2 h à 150 °C.

La Figure 6.11 montre l'état de la mémoire Flash après la récupération temporelle et la récupération thermique. On peut y observer une chute significative du nombre de fautes présentes dans la mémoire Flash ainsi qu'une accentuation de l'apparition des lignes

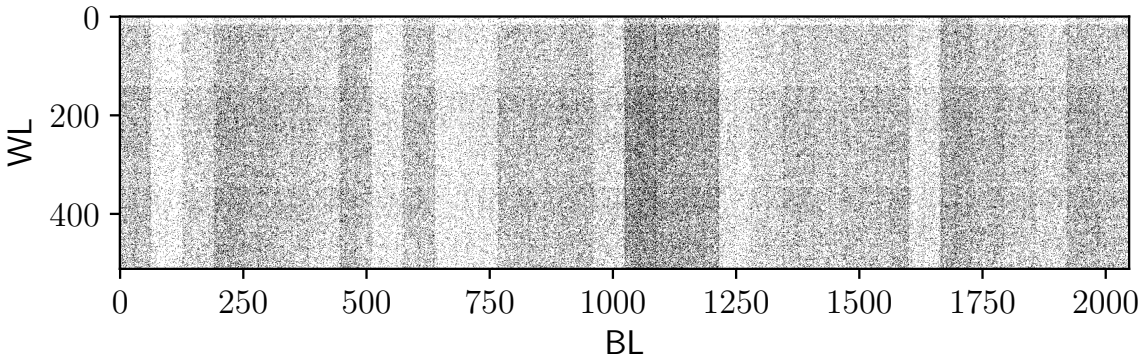


Figure 6.10 – État de la mémoire Flash après une semaine de récupération à température ambiante.

horizontales. On peut supposer que ces lignes horizontales sont dues à l'irradiation des transistors d'accès aux sous-*bitlines* connectés à un même *sense amplifier*.

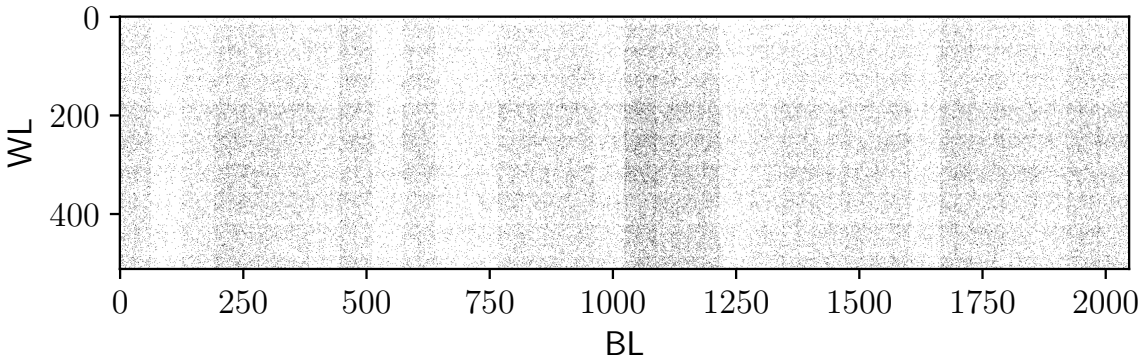


Figure 6.11 – État de la mémoire Flash après récupération temporelle et thermique.

La Figure 6.12 illustre l'évolution du nombre de fautes en mémoire Flash pendant une irradiation par étapes conduisant à une dose totale d'environ 3 500 $Gy_{(SiO_2)}$ ainsi que l'évolution du nombre de fautes après une récupération temporelle d'une semaine à température ambiante ou d'un recuit thermique de 2 h à 150 °C.

On peut ainsi remarquer qu'une récupération thermique courte à haute température est beaucoup plus efficace qu'une récupération temporelle longue à température ambiante comme indiqué dans le Tableau 6.3. En effet, la récupération thermique élimine 70 % des fautes alors que la récupération temporelle n'en élimine que 25 %.

6.3.3 Synthèse des résultats obtenus

Les résultats précédemment décrits mettent en évidence qu'il est possible d'injecter des fautes au sein de mémoires Flash de composants non alimentés en irradiant le composant

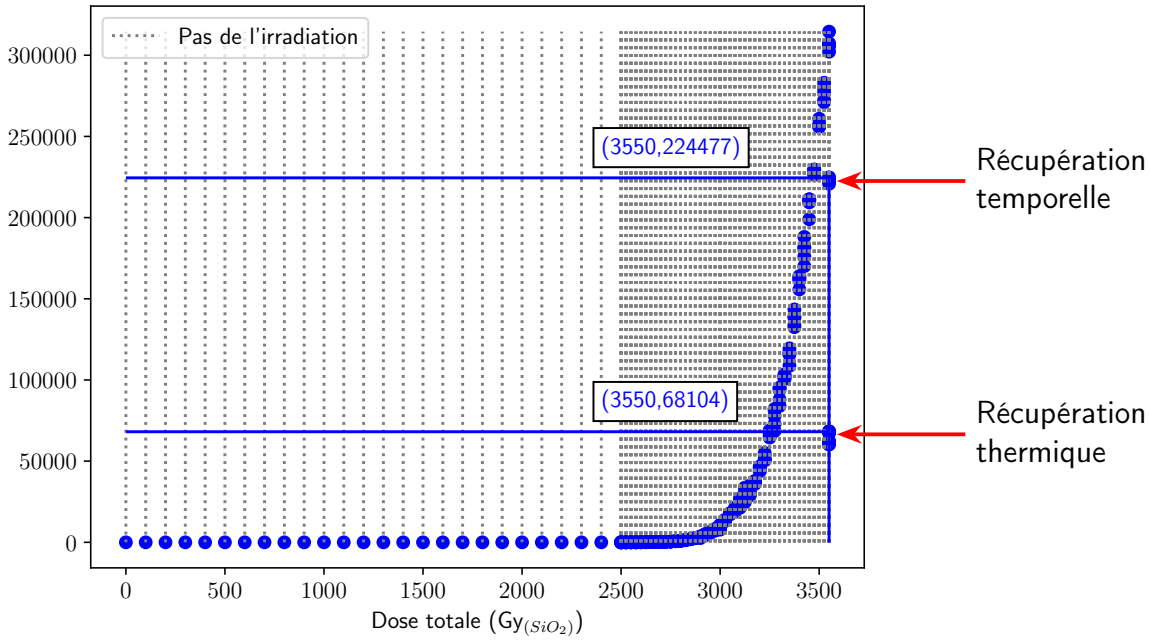


Figure 6.12 – Évolution du nombre de fautes en mémoire Flash pendant les irradiations aux rayons X avec récupération thermique et temporelle. Chaque point bleu correspond à une lecture de la mémoire.

	# Fautes (en bits)	Diminution
Après les irradiations	≃ 300 000	-
Après récupération temporelle (environ une semaine)	≃ 225 000	-25%
Après récupération thermique (2 h à 150 °C)	≃ 70 000	-69%

Tableau 6.3 – Récupération temporelle et thermique.

aux rayons X. Les fautes sont de deux types :

1. Des fautes permanentes dues à la décharge des transistors à grille flottante qui composent la mémoire Flash. L'énergie apportée par les photons X aux électrons leur permet de s'échapper de puit potentiel formé par la grille flottante et les oxydes. On a donc une diminution progressive de la charge stockée.
2. Des fautes non permanentes dues à une dérive des tensions de seuil des transistors MOS qui composent la logique de contrôle et de lecture de la mémoire. Ces fautes peuvent être partiellement corrigées après une récupération temporelle et une récupération thermique. Il est en effet impossible que les processus de récupération injecte des électrons dans la grille flottante et restaure la charge électrique stockée initialement.

De plus, dans cette configuration, l'entièreté de la mémoire Flash est irradiée. Ainsi, les fautes injectées sont réparties sur l'ensemble de la mémoire Flash. Il apparait difficilement envisageable qu'un scénario d'attaque puisse exploiter ce type d'injection lorsqu'il affecte la totalité du circuit. C'est pourquoi la section 6.4 s'intéresse à la conception et à la réalisation d'un masque afin de focaliser l'injection de fautes.

6.4 Réalisation d'un masque de focalisation

Afin de focaliser les rayons X sur une petite surface de la mémoire Flash, un masque a été réalisé [1]. Ce dernier permet de limiter, voire bloquer, les rayons X selon son épaisseur et sa composition. De plus, il est percé en son centre afin de laisser passer les radiations en un point donné et donc de focaliser les rayons X. Il existe notamment deux matériaux possédant les caractéristiques adéquates à une focalisation de l'irradiation : le tungstène (W) et le plomb (Pb).

6.4.1 Simulations numériques de l'efficacité des masques

Des simulations numériques en `Python` avec la librairie `SpekPy` ont été réalisées afin d'estimer les capacités de filtrage des rayons X des deux matériaux précédemment évoqués. Afin de comparer les efficacités des deux matériaux, les épaisseurs des masques sont fixées à 25 µm. La Figure 6.13 montre les spectres d'énergie après filtration par les masques.

On peut ainsi observer que ces masques ne sont efficaces que pour les basses énergies, s'étalant de 0 keV à environ 50 keV. Ce phénomène est particulièrement intéressant car ce sont les photons de basse énergie qui interagissent le plus avec la matière.

De plus, on peut également noter que le masque en tungstène est plus efficace pour filtrer les rayons X que le masque en plomb. Ce phénomène s'explique par le fait que les rayons X sont produits avec une cible en tungstène.

La Figure 6.14 montre le facteur d'atténuation des différents masques selon l'énergie des photons générés. Ici encore, on remarque que le masque en tungstène est plus efficace que le masque en plomb, notamment pour les basses énergies.

Ces simulations numériques nous permettent de calculer l'atténuation globale des deux masques. Le Tableau 6.4 synthétise ces grandeurs.

1. Merci à Stéphanie Anceau, Laurent Maingault, Sophie Bouat et Luc Salvo, membres du `CEA-LETI` et du laboratoire SIMaP pour la réalisation du masque.

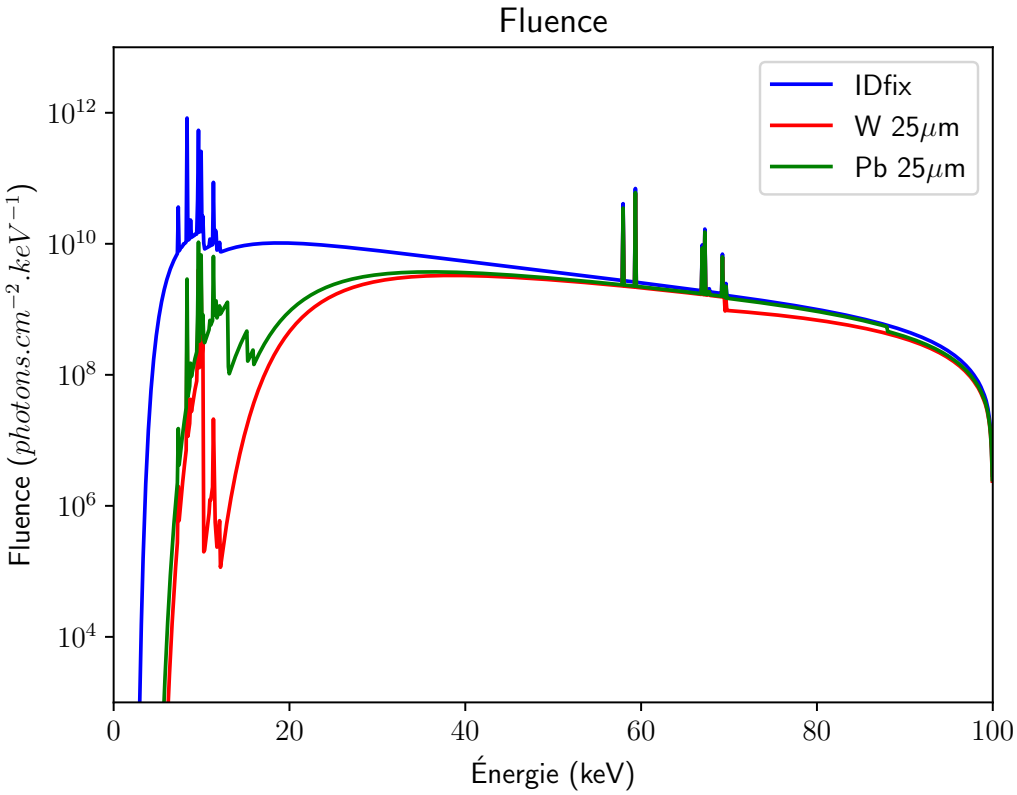


Figure 6.13 – Simulation du spectre d'énergie avec filtration par un masque en tungstène (W) ou en plomb (Pb) d'épaisseur 25 µm.

Masque	Atténuation globale
Tungstène (25 µm)	≃ 75%
Plomb (25 µm)	≃ 70%

Tableau 6.4 – Synthèse des atténuations des deux masques obtenues en simulation pour les énergies comprises entre 0 keV et 100 keV.

Ces résultats nous indiquent que le masque en tungstène semble plus adapté pour focaliser les rayons X dans la configuration expérimentale utilisée.

6.4.2 Caractérisation expérimentale du masque

À l'issu des simulations précédemment présentées, un masque en tungstène d'une épaisseur de 25 µm a été réalisé. Dans ce but, une feuille carrée de 2,5 cm de côté de tungstène de cette épaisseur a été percée à l'aide d'un FIB (*Focused Ion Beam*).

La Figure 6.15 montre le trou percé dans cette feuille de tungstène vu au microscope optique. Le trou n'est pas parfait à l'échelle micrométrique, il mesure 9 µm d'un côté et 18,5 µm de l'autre.

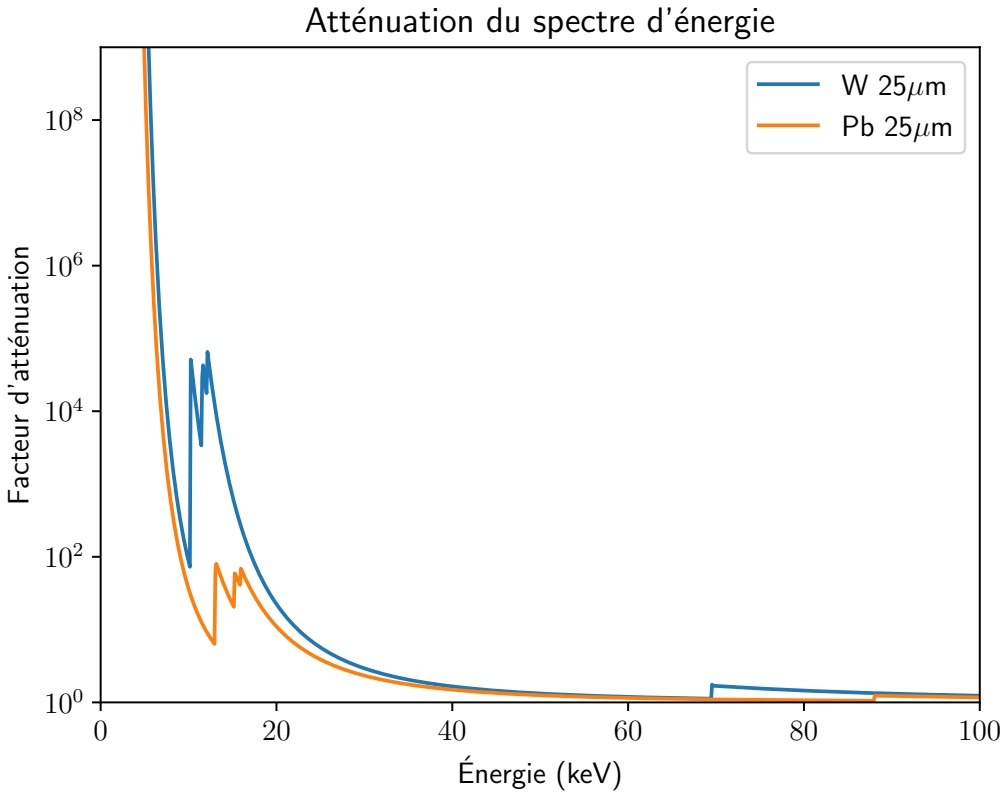
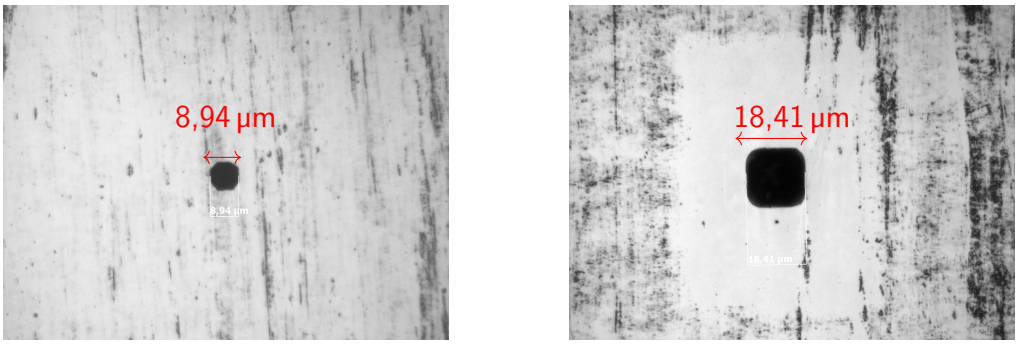


Figure 6.14 – Atténuation des masques en plomb et tungstène pour une épaisseur de 25 µm.



(a) Face avant. **(b)** Face arrière.

Figure 6.15 – Images au microscope optique du masque en tungstène.

Afin de caractériser expérimentalement l'efficacité du masque, une dosimétrie dite "à vide", à l'aide de la chambre d'ionisation PTW 23344, est réalisée. L'irradiateur est configuré pour obtenir un débit de dose de 1 $Gy_{(SiO_2)}$/s. Le masque est ensuite placé sur la chambre d'ionisation et une dosimétrie est à nouveau effectuée.

Le Tableau 6.5 indique les débits de dose obtenus avec le masque.

Positionnement du masque	Débit de dose ($Gy_{(SiO_2)}/s$)	Atténuation
Face avant au dessus	0,135	$\backsimeq$ 87 %
Face arrière au dessus	0,060	$\backsimeq$ 94 %

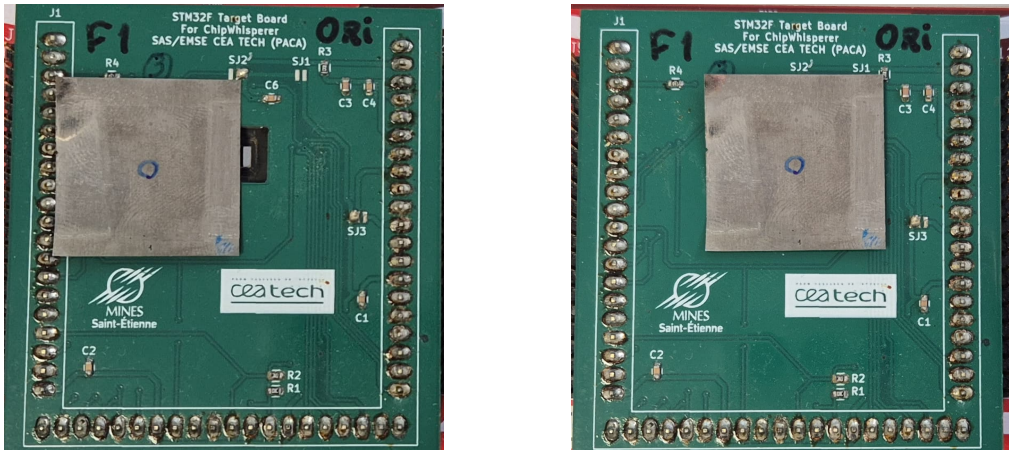
Tableau 6.5 – Mesure expérimentale de l'efficacité du masque.

Ainsi, on peut observer que le masque offre une meilleure protection en pratique (environ 90 %) qu'en simulation (environ 75 %).

Des campagnes d'irradiations, suivant le même protocole que décrit en Figure 6.4, avec le masque placé au-dessus du composant ont été réalisées. Deux configurations différentes ont été mises en place :

— configuration ⓐ : le masque couvre partiellement le composant (voir Figure 6.16a). Dans cette configuration, on ne cherche pas à focaliser l'irradiation mais à caractériser l'effet du masquage (mémoire Flash partiellement protégée et partiellement exposée aux radiations),

— configuration ⓑ : le masque couvre la totalité de la mémoire Flash à l'exception du trou présent dans le masque (voir Figure 6.16b).



(a) Configuration ⓐ **(b)** Configuration ⓑ

Figure 6.16 – Positionnements du masque pour les essais d'irradiations aux rayons X.

Configuration ⓐ

La Figure 6.17 montre l'évolution du nombre de fautes en mémoires Flash pendant les irradiations successives. On observe que la courbe suit une tendance similaire aux irradiations sans le masque. En revanche, les premières fautes apparaissent pour une dose plus faible que lors des irradiations sans le masque. En effet, la première faute apparaît pour une dose de 1 500 $Gy_{(SiO_2)}$ et plus de 8 000 fautes sont obtenues pour une

dose de 2 000 $Gy_{(SiO_2)}$ alors que sans le masque la première faute apparaît pour une dose de 2 500 $Gy_{(SiO_2)}$.

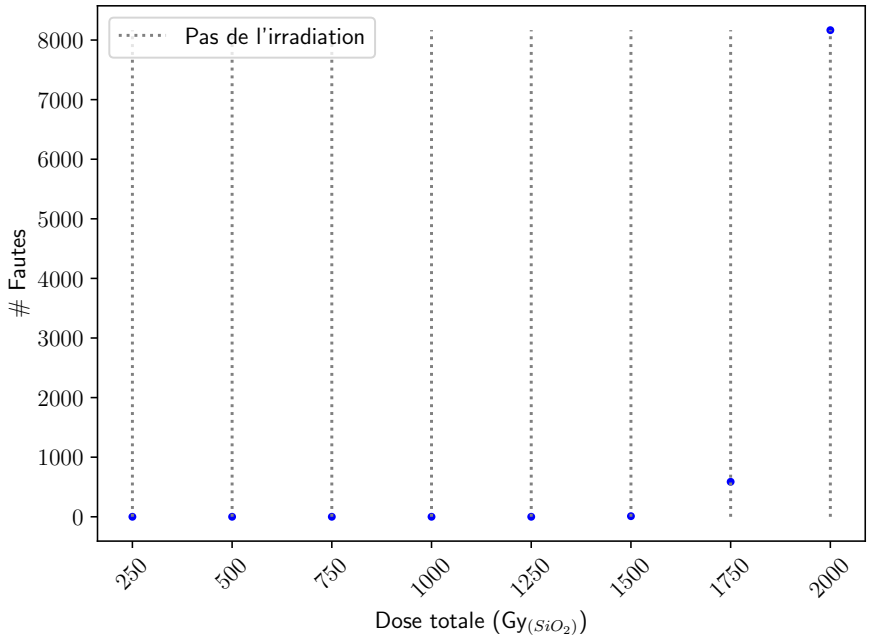


Figure 6.17 – Évolution du nombre de fautes en mémoires Flash pour la configuration ⓐ.

La Figure 6.18 illustre l'état de la mémoire Flash à la fin de la campagne d'irradiation. On peut observer que les fautes sont majoritairement présentes dans la moitié gauche de la mémoire à l'exception d'un ensemble de fautes aux alentours de la *bitline* d'indice 1 500 et des *wordlines* d'indices 0 à 200. L'expérience s'arrête pour une dose totale de 2 250 $Gy_{(SiO_2)}$ à cause de fautes injectées dans les registres `RDP` ou `nRDP` du microcontrôleur.

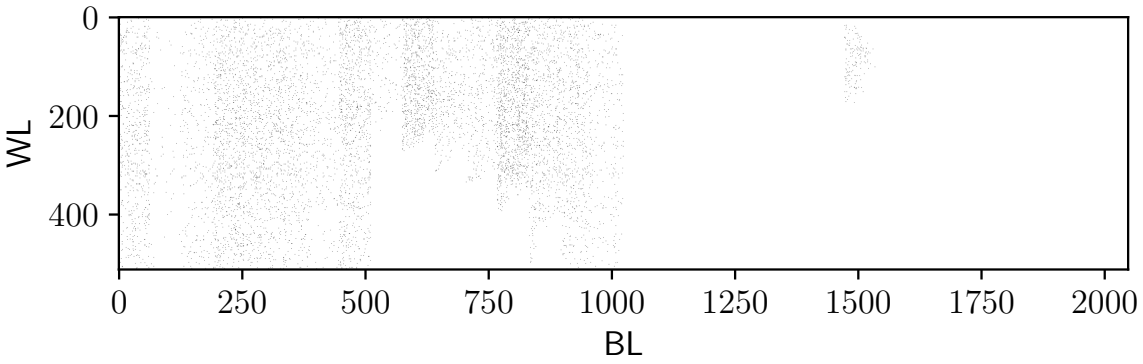


Figure 6.18 – État de la mémoire Flash après les irradiations pour la configuration ⓐ et une dose totale de 2 000 $Gy_{(SiO_2)}$.

Il existe deux causes possibles pouvant expliquer ce phénomène. Soit le masque ne filtre pas les rayons X, soit les fautes sont dues à une irradiation des logiques de contrôle ou de lecture.

Configuration ⓑ

La Figure 6.19 représente l'évolution du nombre de fautes en mémoire Flash en configuration ⓑ.

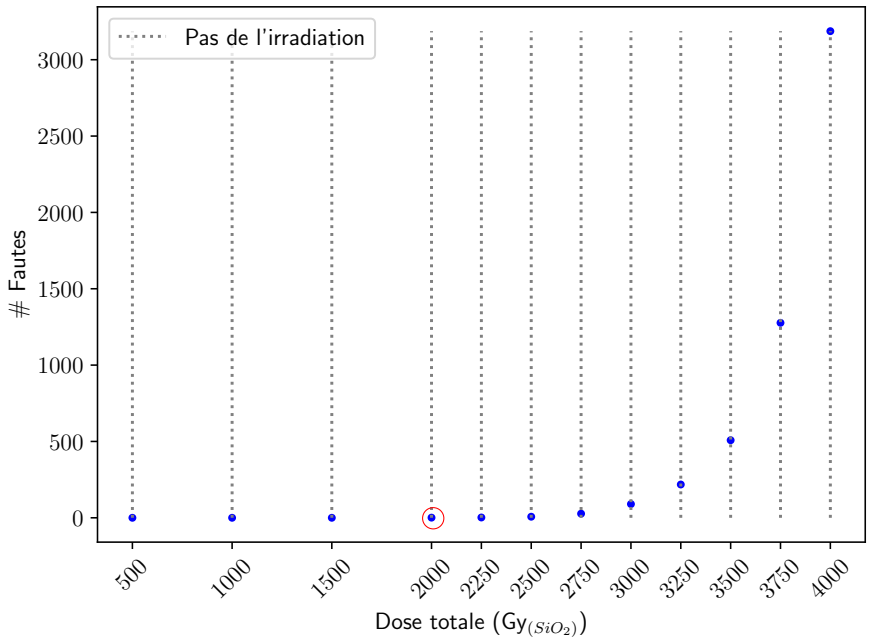


Figure 6.19 – Évolution du nombre de fautes en mémoires Flash pour la configuration ⓑ. Entouré en rouge l'apparition d'une faute monobit pour une dose de 2 000 Gy$_{(SiO_2)}$.

Dans cette configuration, on peut observer que la croissance du nombre de fautes injectées dans la mémoire Flash est nettement ralentie par la présence du masque. Ainsi, il est possible d'appliquer une dose totale plus importante que sur les précédentes expériences avant d'avoir un impact sur les registres `RDP` ou `nRDP`. On obtient environ 3 100 fautes pour une dose de 4 000 Gy$_{(SiO_2)}$ alors que dans l'expérience décrite précedemment nous avions obtenu environ 8 000 fautes pour une dose de 2 000 Gy$_{(SiO_2)}$.

De plus, il existe un point particulièrement intéressant sur la Figure 6.19. Pour une dose de 2 000 Gy$_{(SiO_2)}$, une seule faute est présente en mémoire Flash. Ainsi, nous sommes parvenus à injecter une faute monobit dans la mémoire Flash à l'adresse `0x080099F0`, soit dans la $38^{ème}$ page et plus précisèment à l'intersection de la *bitline* d'indice 1 087 et de la *wordline* d'indice 153. Cet aspect peut présenter un avantage significatif dans

le cadre d'attaques d'algorithmes cryptographiques. En effet, il existe des scénarios d'attaques qui requièrent que la faute injectée soit monobit comme la DFA.

La Figure 6.20 illustre l'état de la mémoire Flash à la fin des irradiations après le dépôt d'une dose de 4 000 $Gy_{(SiO_2)}$. On peut observer que les fautes ne sont pas concentrées mais réparties sur l'entièreté de la mémoire Flash. Ainsi, ce masque ne permet pas de focaliser l'injection de fautes aux rayons X malgré les résultats obtenus par les simulations numériques. En revanche, il permet de ralentir l'apparition des fautes et d'obtenir une faute monobit pour un certain niveau de dose.

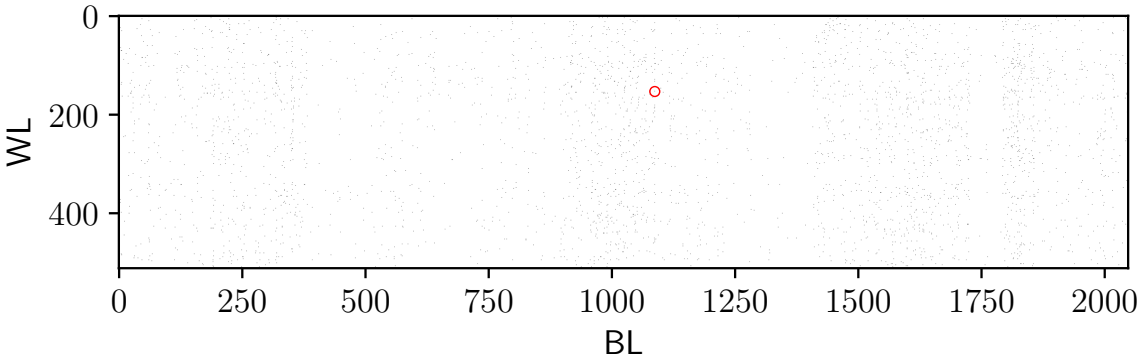


Figure 6.20 – État de la mémoire Flash après les irradiations pour la configuration ⓑ. Entouré en rouge la première faute monobit apparue pour une dose de 2 000 $Gy_{(SiO_2)}$.

6.5 Utilisation d'un tomographe comme irradiateur

D'autres campagnes d'irradiations ont été réalisées au sein du laboratoire SIMaP (Science et Ingénierie des Matériaux et des Procédés) de l'Université Grenoble-Alpes. Ces expériences n'ont pas été réalisées avec un irradiateur mais à l'aide d'un *tomographe*. La tomographie par absorption de rayons X est une technique non destructive qui permet la reconstruction d'images en coupe d'un objet [THI13].

6.5.1 Description du tomographe

Le fonctionnement général d'un tomographe est décrit en Figure 6.21. Un tomographe est constitué de plusieurs éléments distincts :

— une source de rayons X avec un flux de photons assez faible,
— un plateau tournant permettant le positionnement et la rotation de l'objet à analyser,

— une caméra CCD permettant la conversion des photons X en image visible.

Pour chaque valeur angulaire du plateau tournant, une radiographie de l'objet est obtenu par le détecteur caméra CDD et un logiciel d'imagerie permet la reconstruction 3D de l'objet.

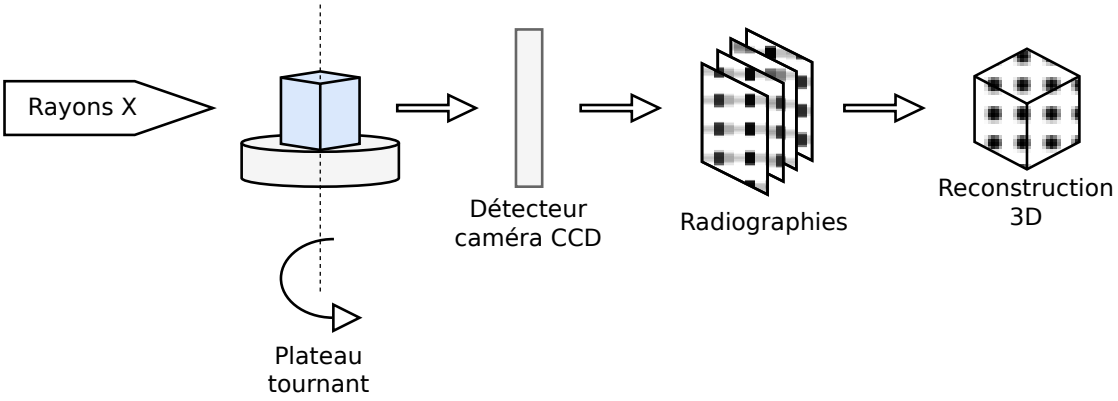


Figure 6.21 – Description du fonctionnement d'un tomographe. Adapté de [Fal08].

Une photographie du montage est visible en Figure 6.22. On y retrouve, de droite à gauche, la source de rayons X, le masque sur son support, la cible et la caméra CDD qui sert de détecteur. On peut également voir le plateau tournant et le programmateur.

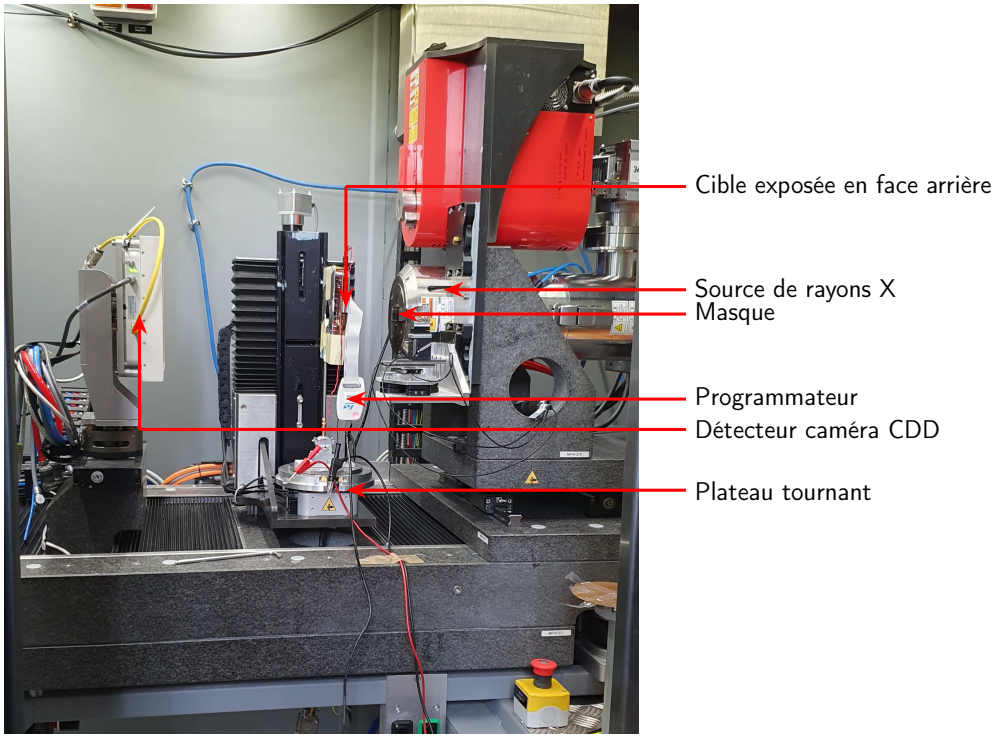
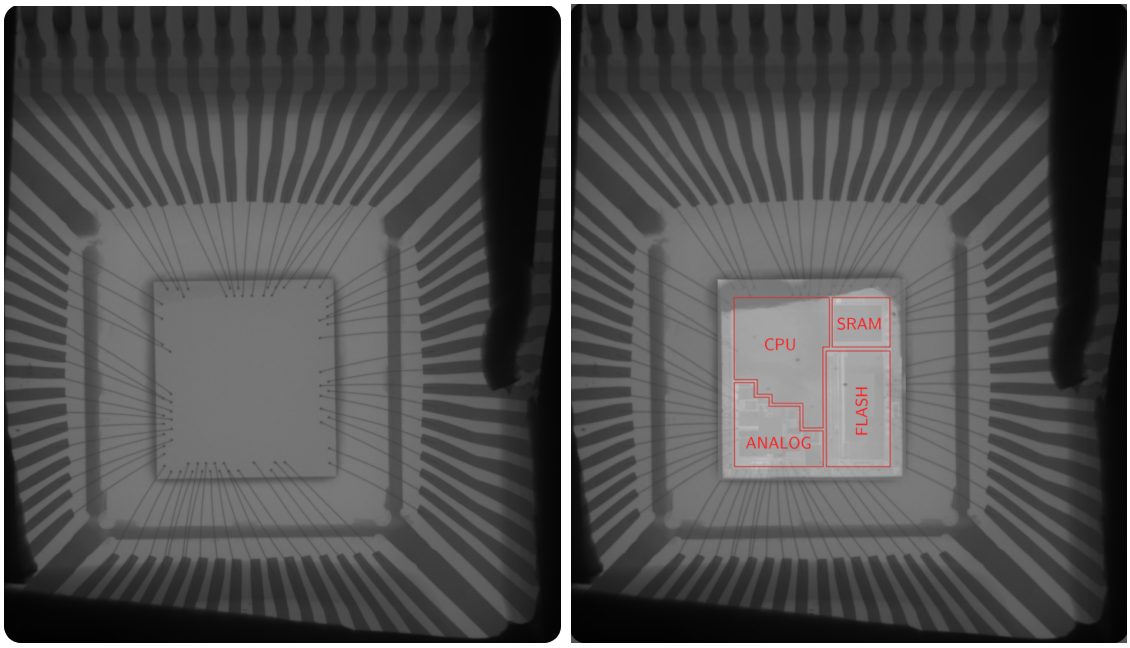


Figure 6.22 – Photographie du montage expérimental.

Une image de la cible obtenue avec le tomographe est visible en Figure 6.23a. On peut y voir les fils de *bonding*. Une superposition de l'image précédente avec une image infrarouge est visible en Figure 6.23b.



(a) **(b)**

Figure 6.23 – Images obtenues au tomographe du `STM32F100` sans **(a)** et avec **(b)** la superposition de l'image IR.

Les caractéristiques de l'irradiateur utilisé dans les expériences décrites ci-après sont fournies dans le Tableau 6.6.

Tube à rayons X	Hamamatsu L10711-03 [Ham]
Tension maximale	160 kV
Courant maximal	50 µA
Matériau de l'anode	Tungstène (W)
Couverture du faisceau	140°
Filtrage du faisceau	diamant (C)

Tableau 6.6 – Caractéristiques du tomographe.

Dans nos expériences, la tension du tube est fixée à 50 kV et le courant du tube à 50 µA.

Le spectre du tomographe est visible en Figure 6.24 en bleu. La simulation numérique réalisée en `Python` avec la librairie `SpekPy` prend en compte le filtre constitué de la fenêtre en diamant (voir Tableau 6.6).

Dans certaines des expériences décrites ci-après, un nouveau masque a été utilisé. C'est un disque de tungstène (W) d'une épaisseur de 1 mm et comportant un trou au centre d'un diamètre de 25 µm. Une illustration du masque est visible en Figure 6.25.

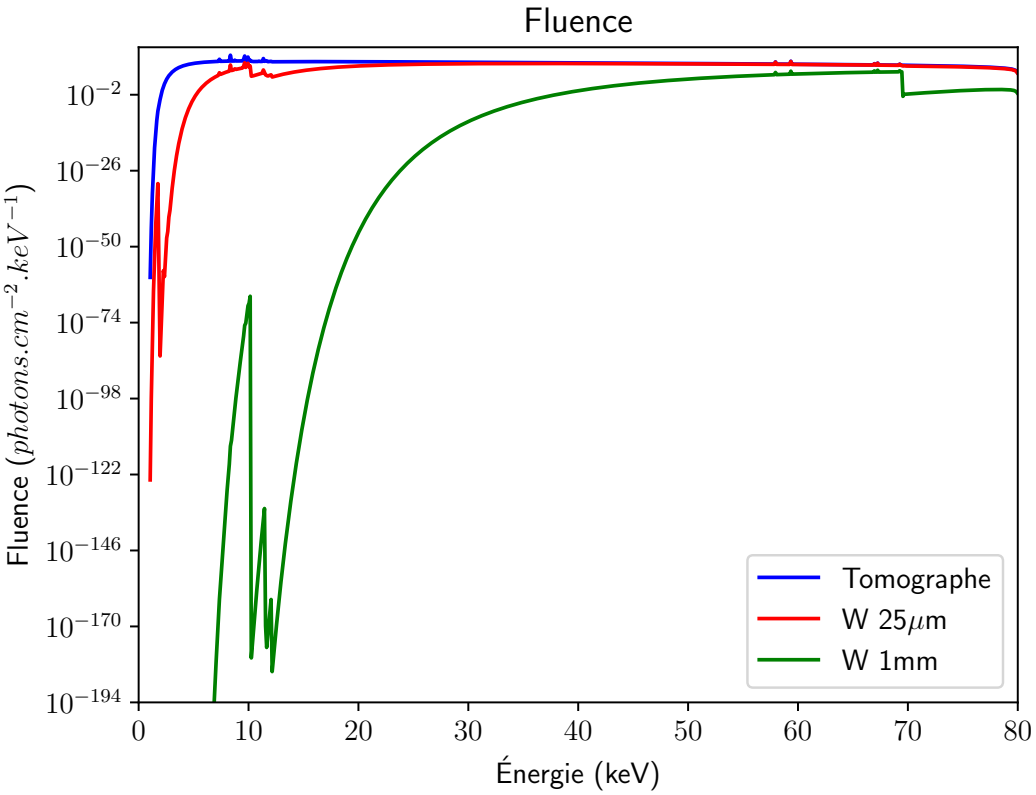


Figure 6.24 – Spectre simulé du tomographe obtenu avec une tension de tube de 80 kV, un courant de 50 µA et une distance de 5 cm avec ou sans un filtrage par masque (W) d'épaisseur 25 µm ou 1 mm.

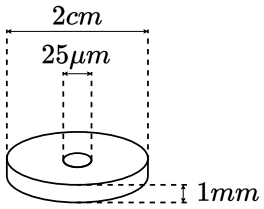


Figure 6.25 – Schéma du nouveau masque.

La Figure 6.24 contient également le spectre simulé avec la présence d'un masque en tungstène d'une épaisseur de 25 µm en rouge et un masque en tungstène d'une épaisseur de 1 mm en vert. On peut remarquer une nette diminution dans la fluence différentielle lors de la présence du masque épais d'un millimètre. En simulation, on obtient une atténuation d'environ 80 % pour le masque avec une épaisseur de 25 µm alors que l'atténuation atteint 99,99 % pour le masque avec une épaisseur de 1 mm.

6.5.2 Expériences réalisées

Lors de ces expériences, trois configurations différentes ont été mises en place :

— configuration ⓐ : circuit alimenté avec le masque en tungtène d'épaisseur 25 µm décrit en section 6.4,
— configuration ⓑ : circuit alimenté avec le nouveau masque en tungstène d'épaisseur 1 mm,
— configuration ⓒ : circuit non alimenté avec le nouveau masque en tungstène d'épaisseur 1 mm.

De précédents travaux de l'état de l'art ont déjà étudiés l'effet des rayons X produits par ce tomographe sur des circuits alimentés [Mai+21], c'est pourquoi c'est le point de départ choisi dans nos expériences. Nous avons ensuite souhaité investiguer la possibilité d'injecter des fautes avec le même dispositif expérimental sur des circuits non alimentés.

Pour l'ensemble des expériences, l'origine des axes est prise à l'intersection des fils de *bonding* du bord droit et du bord inférieur à l'image.

6.5.2.a Configuration ⓐ

Description

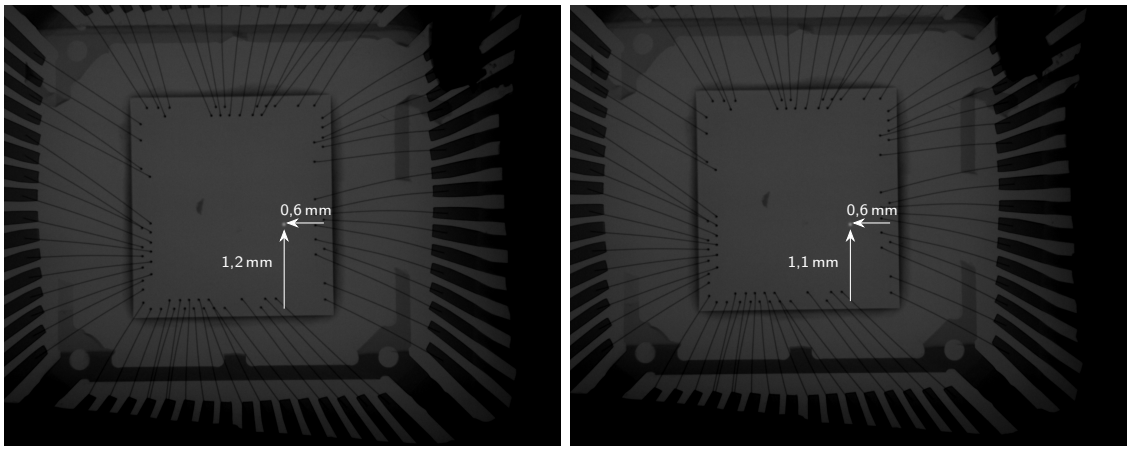
Pour cette configuration, le composant est alimenté pendant les irradiations et des lectures complètes de la mémoire Flash sont réalisées toutes les 5 minutes. Ici, c'est le masque avec une épaisseur de 25 µm écrit en section 6.4 qui est utilisé. Ce dernier est placé à deux positions différentes au-dessus de la mémoire Flash. Les deux positions sont les suivantes :

— position ① : (x,y)=(0,6 mm,1,2 mm),
— position ② : (x,y)=(0,6 mm,1,1 mm).

Les deux positions du masque sont visibles sur les Figure 6.26a et Figure 6.26b. Le point blanc pointé par les flèches correspond au trou dans le masque qui laisse passer la totalité des rayons X. Cela constitue un avantage significatif dans l'utilisation du tomographe. En effet, le système d'imagerie permet d'observer la position du masque, et notamment du trou dans le masque, par rapport au circuit ciblé.

Une lecture complète du contenu de la mémoire Flash est réalisée toutes les 5 minutes. Après 35 minutes d'irradiations, le masque est déplacé de la position ① vers la position ② et une irradiation de 25 minutes est effectuée.

Résultats



(a) Position ①. **(b)** Position ②.

Figure 6.26 – Images obtenues au tomographe de la cible avec le masque pour la configuration ⓐ.

À l'issu de la campagne d'irradiation à la position ① une dizaine de fautes sont obtenues. En revanche, leur position ne correspond pas à la position du masque au-dessus de la mémoire Flash. De plus, lors de l'irradiation de la position ②, la communication avec le composant est impossible après une quinzaine de minutes ce qui nous indique que d'autres parties du composant que la mémoire sont impactées par les irradiations.

Ces résultats nous montrent que le masque n'est pas efficace. C'est pourquoi un second masque est testé dans les configurations décrites ci-après.

6.5.2.b Configuration ⓑ

Description

Pour cette configuration, le composant est alimenté pendant les irradiations mais contrairement à la configuration ⓐ c'est le masque décrit en sous-section 6.5.1 qui est utilisé. Pour rappel, ce masque est un disque de 2 cm de diamètre avec une épaisseur de 1 mm et possédant un trou de 25 µm au centre comme illustré sur la Figure 6.25. Trois positions différentes ont été testé dans cette configuration :

— position ① : (0,6 mm,1,1 mm),
— position ② : (0,6 mm,1,2 mm),
— position ③ : (0,7 mm,1,2 mm).

L'image obtenue au tomographe lors de la position ② est visible en Figure 6.27. On peut observer que l'image obtenue du composant est nettement moins clair que lors de

la configuration ⓐ car le masque est par contraste beaucoup plus épais. En revanche, le trou dans le masque est beaucoup plus visible.

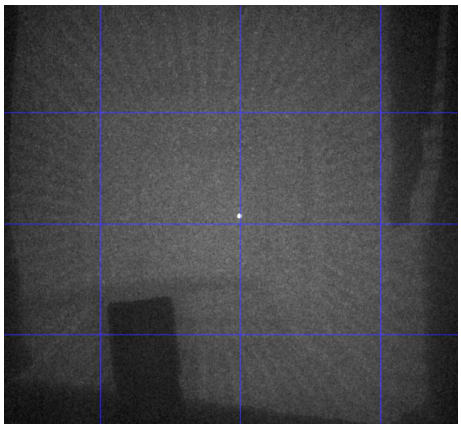


Figure 6.27 – Image obtenue au tomographe de la cible avec le masque pour la configuration ⓑ lors de la position ②.

Résultats

L'état de la mémoire après 1h20 d'exposition aux radiations est visible en Figure 6.28. On observe la présence de 37 fautes localisées à la position du trou dans le masque.

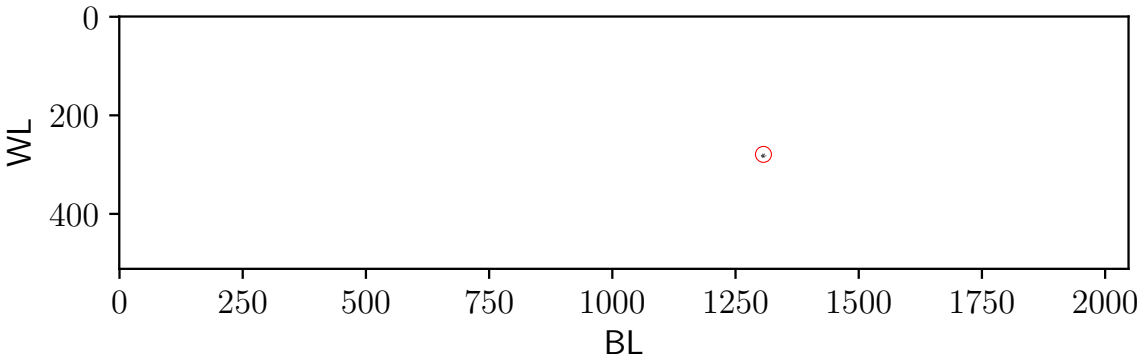


Figure 6.28 – État de la mémoire Flash après 80 min d'exposition à la première position.

Après 1 heure d'irradiations à la position ②, une dizaine de fautes apparaissent à une nouvelle position comme visible en Figure 6.29.

Pour finir, le masque est placé à la troisième position et une irradiation est effectuée pour une durée de 3 heures. L'état de la mémoire est visible en Figure 6.30. À l'issu de l'irradiation, environ 300 fautes sont présentes à la nouvelle position du masque.

Nous pouvons conclure de cette configuration que le masque épais de 1 mm permet effectivement de focaliser l'injection de fautes.

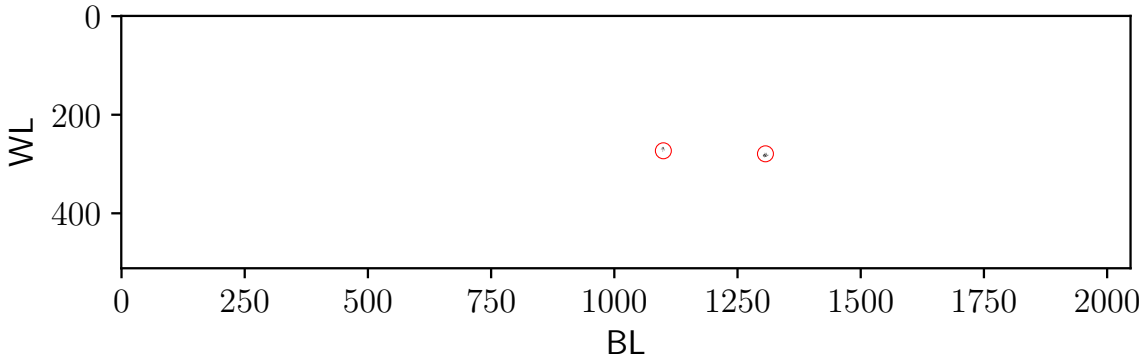


Figure 6.29 – État de la mémoire Flash après 60 min d'exposition à la deuxième position.

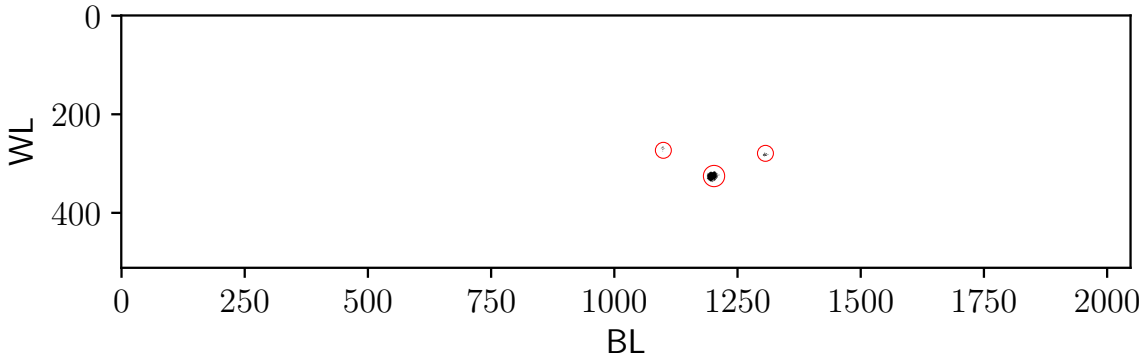


Figure 6.30 – État de la mémoire Flash après 180 min d'exposition à la troisième position.

6.5.2.c Configuration ©

Description

Pour cette configuration, le composant est éteint pendant les irradiations et c'est le masque épais de 1 mm qui est utilisé. Deux positions différentes sont exposées aux radiations :

— position ① : (0,6 mm,1,2 mm),
— position ② : (0,6 mm,1,1 mm).

Pour la position ①, deux irradiations longues d'environ 1 800 s sont réalisées alors que pour la position ② une première irradiation de 1 800 s est effectuée et est suivie de nombreuses irradiations de 60 s.

Résultats

L'état de la mémoire Flash à l'issue des irradiations aux deux positions définies ci-dessus est visible en Figure 6.31. Le groupe de fautes de gauche correspond aux irradiations de la position ① alors que le groupe de fautes de droite correspond aux irradiations de la position ②.

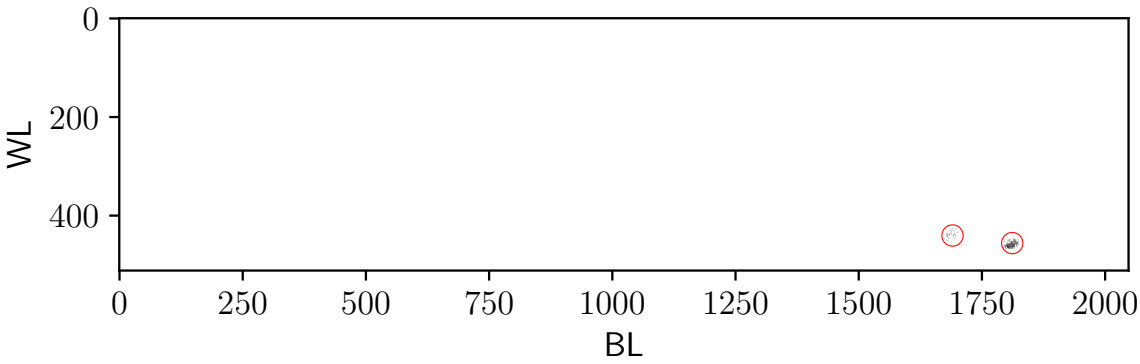


Figure 6.31 – État de la mémoire Flash après les irradiations des deux positions.

L'évolution du nombre de fautes présentes en fonction du temps d'exposition en mémoire Flash est illustrée sur la Figure 6.32. Aucune dosimétrie n'a été réalisée sur le tomographe, ainsi il nous est impossible de représenter l'évolution du nombre de fautes en fonction de la dose déposée. On peut toutefois observer une dépendance exponentielle entre le nombre de fautes injectées et le temps d'exposition aux radiations. Ainsi nous observons environ 70 fautes groupées à la position ① après une exposition totale de 3 600 s, soit 1 h, et environ 240 fautes groupées à la position ② après une seconde exposition totale de 8 100 s soit 2 h15 min.

6.5.3 Synthèse des résultats

Les résultats précédemment présentés nous montrent qu'il est possible de focaliser les injections de fautes aux rayons X en utilisant un masque avec une épaisseur de 1 mm. En revanche, les contraintes de fabrication de ces masques limitent la taille du trou réalisable. En effet, la capacité à bloquer les radiations du masque est directement liée à son épaisseur et plus le masque est épais moins il est possible de réaliser un trou de petit diamètre.

Les fautes injectées avec le tomographe sont, comme avec l'irradiateur du LabHC, composées de fautes permanentes et de fautes non permanentes.

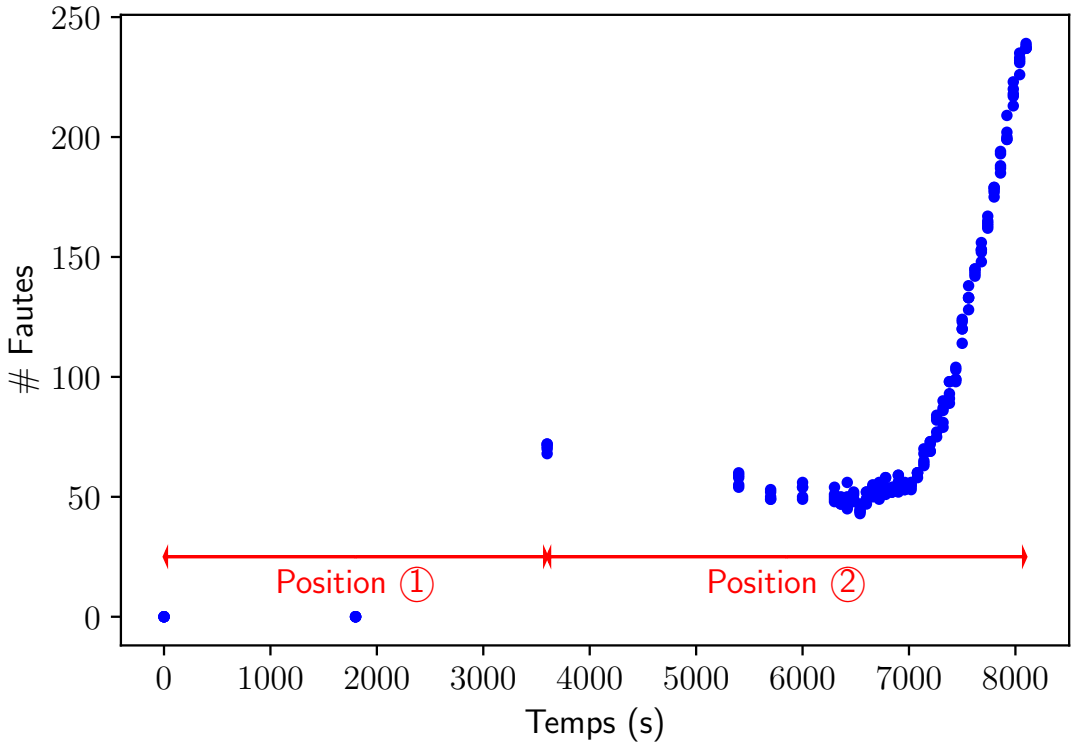


Figure 6.32 – Évolution du nombre de fautes en mémoire Flash avec la configuration ⓒ.

6.6 Conclusion

Dans ce chapitre, nous avons démontré la possibilité d'injecter des fautes en exposant un composant non alimenté aux rayons X. Les fautes obtenues peuvent être scindées en deux catégories : des fautes permanentes et des fautes non permanentes.

D'une part, les fautes permanentes sont causées par la décharge des transistors à grille flottante. En effet, les électrons stockés dans la grille flottante peuvent obtenir suffisamment d'énergie des radiations pour s'échapper de la grille flottante.

D'autre part, les fautes non permanentes sont dues à des dérives de tension de seuil de transistors MOS dans les logiques de contrôle ou de lecture. Ces fautes peuvent être partiellement récupérées par recuit thermique en chauffant le composant pendant 2 h à 150 °C.

Nous avons également montré qu'il est possible de focaliser l'injection de fautes en utilisant un masque en tungstène d'une épaisseur de 1 mm. En revanche, les contraintes mécaniques de fabrication des masques nous empêchent d'avoir une résolution fine sur le diamètre du trou. Ainsi, il n'est pas possible à l'heure actuelle d'envisager l'attaque

d'algorithmes cryptographiques.

Ce chapitre a fait l'objet d'une publication dans une conférence internationale :

- P. GRANDAMME, L. BOSSUET, J.M. DUTERTRE, "X-Ray Fault Injection in non-volatile memories on Power OFF devices ", IEEE PAINE 2023 (Physical Assurance and Inspection of Electronics). [GBD23]

Chapitre 7

Conclusion générale

7.1 Conclusion

Cette thèse avait pour but d'apporter des contributions à l'état de l'art existant sur les attaques par injection de fautes sur des circuits intégrés dédiés aux applications IoT. Ces circuits, souvent non sécurisés, sont omniprésents dans nos usages et peuvent contenir des données sensibles. Les attaques par injection de fautes peuvent aboutir à la compromission de la sécurité de ces composants.

La première contribution de ce manuscrit, développée dans le Chapitre 4, décrit l'utilisation d'un banc laser multispot dans le but d'injecter des fautes multiples dans l'exécution d'un code source stocké en mémoire Flash d'un microcontrôleur. Après avoir réalisé une description des limites des bancs laser monospot, les avantages d'un banc laser multispot ont été mis en évidence. À des fins de caractérisation, deux exemples expérimentaux de nouvelles possibilités d'attaques sont décrits dans ce chapitre. Ce nouveau banc laser permet ainsi d'injecter des fautes à des positions différentes à des instants très proches dans le temps ou d'injecter de multiples fautes simultanément. Il permet ainsi d'envisager de nouveaux scénarios d'attaque qui n'étaient pas atteignables jusqu'à ce jour avec les bancs laser existants.

La deuxième contribution de ce manuscrit, apportée dans le Chapitre 5, détaille l'utilisation d'un banc laser infrarouge monospot dans le but d'injecter des fautes persistantes au sein de la mémoire Flash de microcontrôleurs non alimentés. Une précision au niveau du bit est atteinte en utilisant un spot laser de 5 µm de diamètre. Un nouveau modèle de faute complet, allant du niveau physique au niveau applicatif en passant par les niveaux logique et mémoire, est developpé. Une décharge des grilles flottantes qui composent la mémoire Flash, causée par l'élévation de température apportée par le laser, est à l'origine des fautes. Ces dernières sont persistantes et non destructives. Une application cryptographique est également décrite dans un contexte de PFA permettant de retrouver la clé

128 bits d'une implémentation logicielle de l'algorithme de chiffrement AES. Pour finir, cette attaque étant menée sur un circuit non alimenté, elle empêche toute détection ou réaction du circuit pendant l'attaque et aucune synchronisation entre la cible et le laser n'est nécessaire. Cela permet donc à l'attaquant de disposer du temps qu'il souhaite pour injecter de multiples fautes sans nécessiter un banc laser multispot.

La dernière contribution de ce manuscrit, exposée dans le Chapitre 6, expose la possibilité d'injecter des fautes au sein de mémoire Flash de composants alimentés et non alimentés. Deux types de fautes distincts sont obtenus : des fautes permanentes et des fautes non permanentes. Les fautes permanentes sont causées par la décharge des transistors à grille flottante qui composent la mémoire alors que les fautes non permanentes sont causées par une dérive des tensions de seuil due à un piégeage de charges dans les oxydes des transistors MOS des logiques de contrôle et de lecture de la mémoire. Nous avons également démontré la possibilité de focaliser l'injection de fautes en utilisant un masque en tungstène d'une épaisseur de 1 mm.

Pour conclure, les travaux réalisés dans cette thèse ont permis de contribuer à un élargissement de l'état de l'art par l'utilisation d'un nouveau banc laser multispot, par l'utilisation des rayons X comme moyen d'injection de fautes ou encore par l'attaque de circuits non alimentés. Avec très peu de travaux existants sur les attaques de circuits non alimentés, cette étude constitue une preuve de concept complète pour ce type d'attaque original. Enfin, nous avons considéré des circuits dédiés à applications IoT mais ces travaux sont suffisamment généraux pour être applicables à d'autres circuits (sécurisé, SoC, etc.).

7.2 Perspectives

Le Chapitre 4 a apporté une caractérisation du banc laser multispot dans le cadre d'injection multiple de fautes. Une première piste de recherche serait d'utiliser ce banc pour des attaques d'algorithmes de cryptographie. En effet, l'injection de multiple fautes peut aboutir à la découverte de nouvelles failles dans ces algorithmes. Une seconde piste pourrait être d'investiguer l'utilisation de ce banc laser pour mener des attaques sur des circuits reconfigurables comme des FPGA ou des SoC-FPGA. Par ailleurs, le fait d'avoir de multiple spots laser nous permet également d'envisager d'attaquer des composants comportant des contre-mesures. Par exemple, il devient possible d'attaquer un bloc d'un circuit et la contre-mesure protégeant ce bloc.

Le Chapitre 5 a décrit l'utilisation d'une source laser infrarouge pour injecter des fautes persistantes au sein de mémoire Flash de microcontrôleur. Ici encore, il est envisageable de mener ce type d'attaque sur des circuits comportant des dispotifs de protection de la

mémoire comme des codes détecteur ou correcteur d'erreurs. De plus, il est également possible d'imaginer un scénario combinant les contributions des Chapitre 4 et Chapitre 5. En effet, les codes détecteur et correcteur d'erreurs possèdent une capacité maximale de détection ou de correction. Il est en effet raisonnable que cette capacité puisse être dépassée en injectant de multiple fautes sur un composant non alimenté. Par ailleurs, il est également possible que le code correcteur d'erreur soit sensible à ces types d'injections de fautes. Il serait intéressant de réfléchir à des scénarios d'attaques et le cas échéant de réfléchir à leur durcissement contre ce nouveau modèle de menace.

Le Chapitre 6 a exposé l'utilisation de sources de rayons X afin d'injecter des fautes dans des mémoires Flash de microcontrôleur. La limite des résultats obtenus est liée a l'aspect non focalisé des sources de radiations. Il a été démontré que l'utilisation d'un masque permet de focaliser les injections mais les résultats ne sont pas suffisants pour envisager les attaques connues dans l'état de l'art de la sécurité matérielle. Une direction de recherche serait de réfléchir à la conception de masques ou des systèmes plus complexes afin d'améliorer la focalisation. De plus, il est également probable que les rayons X aient un effet sur les circuits reconfigurables. C'est pourquoi une autre piste serait l'étude des effets des radiations sur des FPGA, notamment dans le cadre de la génération de nombres aléatoires. Par exemple, les oscillateurs en anneaux, notamment utilisés dans les TRNG et les PUF, semblent être des cibles privilègiées pour ces attaques.

Globalement, nous avons montré que des attaques menées sur des circuits non alimentées pouvaient modifier des propriétés physiques des transistors qui composent ces circuits. Ainsi se pose la question de la conception de contre-mesures dédiées aux attaques sur circuits éteints. On peut, d'une part, imaginer des capteurs conçus autour de ce composant élémentaire, le transistor, dont les propriétés électriques seraient altérées par une attaque sur circuit éteint à l'image de la décharge des transistors à grille flottante des mémoires Flash démontrée dans les Chapitre 5 et Chapitre 6. D'autre part, il est également nécessaire de réfléchir à la conception de contre-mesures propres aux analyses qui exploitent ces attaques, comme cela a été réalisé par `Tissot` *et al.* [TBG23], étude dans laquelle un dispositif efficace de protection contre la PFA est proposé.

Comme nous l'avons démontré, les attaques sur circuits non alimentés peuvent mettre en défaut les implémentations logicielles des algorithmes cryptographiques mais également d'autres applications telles que les circuits embarquant de l'intelligence artificielle ou des générateurs de nombres aléatoires par exemple. Il est donc primordial que la communauté de la sécurité matérielle s'empare de ce sujet et propose des contre-mesures contre ce nouveau vecteur d'attaque.

7.3 Publications

7.3.1 Publication dans un journal international

[Gra+24] Paul Grandamme, Pierre-Antoine Tissot, Lilian Bossuet, Jean-Max Dutertre, Brice Colombier, Vincent Grosso. **"Switching Off your Device Does Not Protect Against Fault Attacks"**. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES 2024)*, Septembre 2024.

7.3.2 Conférence internationale avec comité de lecture

[Col+22] Brice Colombier, Paul Grandamme, Julien Vernay, Émilie Chanavat, Lilian Bossuet, Lucie de Laulanié, Bruno Chassagne. **"Multi-spot laser fault injection setup : New possibilities for fault injection attacks"**. *20th Smart Card Research and Advanced Application Conference (CARDIS 2021)*, Novembre 2021.

[GBD23] Paul Grandamme, Lilian Bossuet, Jean-Max Dutertre. **"X-Ray Fault Injection in Non-Volatile Memories on Power OFF Devices"**. *2023 IEEE Physical Assurance and Inspection of Electronics (PAINE 2023)*, Octobre 2023.

7.4 Communications

7.4.1 Présentation à un congrès international sans acte

[Gra23b] Paul Grandamme. **"X-Ray Fault Injection on Power OFF devices"**. *Cryptographic architectures embedded in logic devices (CryptArchi 2023)*, Cantabria, Espagne, Juin 2023.

[Gra24b] Paul Grandamme. **"Laser Fault Injection on power off devices"**. *RADLAS 2024 : 6th Workshop on Laser Testing of Radiation Effects on Components and Systems*, Noordwijk, Pays-Bas, Septembre 2024.

7.4.2 Présentation à un congrès national sans acte

[Gra23c] Paul Grandamme. **"X-Ray Fault Injection on Power OFF Devices."**. *Journée des doctorants de l'équipe SAS*, Gardanne, France, Juin 2023.

[Gra23d] Paul Grandamme. **"X-Ray Fault Injection on Power OFF devices"**. *Journée thématique sur les Attaques par Injection de Fautes (JAIF 2023)*, Gardanne, France, Septembre 2023.

[Gra24a] Paul Grandamme. **"Éteindre votre composant électronique ne le protège pas !"**. *Journée thématique sur les Attaques par Injection de Fautes (JAIF 2024)*, Rennes, France, Octobre 2024.

7.4.3 Posters

[Gra22] Paul Grandamme. **"Attaque laser de primitives de sécurité non alimentées"**. *Journée thématique sur les Attaques par Injection de Fautes (JAIF 2022)*, Valence, France, Novembre 2022.

[Gra23a] Paul Grandamme. **"Injection de fautes dans les circuits électroniques non alimentés"**. *Journée de la recherche de l'école doctorale EDSIS*, Saint-Étienne, France, Juin 2023.

[Gra24a] Paul Grandamme. **"Éteindre votre composant électronique ne le protège pas !"**. *Journée thématique sur les Attaques par Injection de Fautes (JAIF 2024)*, Rennes, France, Octobre 2024.

Bibliographie

[Abb20] Karim ABBAS. *Handbook of Digital CMOS Technology, Circuits, and Systems*. Springer Cham, 2020. ISBN : 978-3-030-37195-1.

[Ago+10a] Michel AGOYAN et al. « How to flip a bit ? » Dans : *16th IEEE International On-Line Testing Symposium (IOLTS 2010), 5-7 July, 2010, Corfu, Greece*. IEEE Computer Society, 2010, p. 235-239.

[Ago+10b] Michel AGOYAN et al. « Single-bit DFA using multiple-byte laser fault injection ». Dans : *2010 IEEE International Conference on Technologies for Homeland Security (HST)*. 2010, p. 113-119.

[Ago+10c] Michel AGOYAN et al. « When Clocks Fail : On Critical Paths and Clock Faults ». Dans : *Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings*. Sous la dir. de Dieter GOLLMANN, Jean-Louis LANET et Julien IGUCHI-CARTIGNY. T. 6035. Lecture Notes in Computer Science. Springer, 2010, p. 182-193.

[AK96] Ross ANDERSON et Markus KUHN. « Tamper Resistance – a Cautionary Note new ». Dans : *2nd USENIX Workshop on Electronic Commerce (EC 96)*. Oakland, CA : USENIX Association, nov. 1996.

[ALP19] ALPHANOV. *ALPhANOV a conçu un banc laser quatre spots pour l'injection de fautes sur circuits intégrés*. https://www.alphanov.com/actualites/alphanov-concu-un-banc-laser-quatre-spots-pour-linjection-de-fautes-sur-circuits. 2019.

[Als+22] Ihab ALSHAER et al. « Variable-Length Instruction Set : Feature or Bug ? » Dans : *2022 25th Euromicro Conference on Digital System Design (DSD)*. 2022, p. 464-471.

[Anc+17] Stéphanie ANCEAU et al. « Nanofocused X-Ray Beam to Reprogram Secure Circuits ». Dans : *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*. Sous la dir. de Wieland FISCHER et Naofumi

HOMMA. T. 10529. Lecture Notes in Computer Science. Springer, 2017, p. 175-188.

[ANS22] ANSSI. *Comprendre la certification*. 2022. URL : https://cyber.gouv.fr/comprendre-la-certification.

[Arm] *Armv7-M Architecture Reference Manual*. Manual ID021621. Arm Limited, 2021.

[Bar+04] Hagai BAR-EL et al. « The Sorcerer's Apprentice Guide to Fault Attacks ». Dans : *IACR Cryptol. ePrint Arch.* (2004), p. 100.

[Bar06] H. J. BARNABY. « Total-Ionizing-Dose Effects in Modern CMOS Technologies ». Dans : *IEEE Transactions on Nuclear Science* 53.6 (2006), p. 3103-3121.

[Bau05] Robert BAUMANN. « Soft Errors in Advanced Computer Systems ». Dans : *IEEE Des. Test Comput.* 22.3 (2005), p. 258-266.

[BDL01] Dan BONEH, Richard A. DEMILLO et Richard J. LIPTON. « On the Importance of Eliminating Errors in Cryptographic Computations ». Dans : *J. Cryptol.* 14.2 (2001), p. 101-119.

[BDL97] Dan BONEH, Richard A. DEMILLO et Richard J. LIPTON. « On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract) ». Dans : *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*. Sous la dir. de Walter FUMY. T. 1233. Lecture Notes in Computer Science. Springer, 1997, p. 37-51.

[Ber+14] Noemie BERINGUIER-BOHER et al. « Voltage Glitch Attacks on Mixed-Signal Systems ». Dans : *17th Euromicro Conference on Digital System Design, DSD 2014, Verona, Italy, August 27-29, 2014*. IEEE Computer Society, 2014, p. 379-386.

[BFP19] Claudio BOZZATO, Riccardo FOCARDI et Francesco PALMARINI. « Shaping the Glitch : Optimizing Voltage Fault Injection Attacks ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2019 (2019), p. 199-224.

[BG16] M. BAGATIN et S. GERARDIN. « Introduction to the effects of radiation on electronic devices ». Dans : *Ionizing Radiation Effects in Electronics : From Memories to Imagers*. 2016. Chap. 1, p. 1-22.

[BGP17] Marta BAGATIN, Simone GERARDIN et Alessandro PACCAGNELLA. « Space and terrestrial radiation effects in flash memories ». Dans : *Semiconductor Science and Technology* 32.3 (fév. 2017), p. 033003.

[BGV11] Josep BALASCH, Benedikt GIERLICHS et Ingrid VERBAUWHEDE. « An In-depth and Black-box Characterization of the Effects of Clock Glitches on 8-bit MCUs ». Dans : *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2011, Tokyo, Japan, September 29, 2011*. Sous la dir. de Luca BREVEGLIERI et al. IEEE Computer Society, 2011, p. 105-114.

[BH22] Daehyeon BAE et JaeCheol HA. « Implementation of Disassembler on Microcontroller Using Side-Channel Power Consumption Leakage ». Dans : *Sensors* 22.15 (2022), p. 5900.

[BJ15] Jakub BREIER et Dirmanto JAP. « Testing Feasibility of Back-Side Laser Fault Injection on a Microcontroller ». Dans : *Proceedings of the 10th Workshop on Embedded Systems Security, WESS 2015, Amsterdam, The Netherlands, October 8, 2015*. Sous la dir. de Stavros A. KOUBIAS et Thilo SAUTER. ACM, 2015, p. 5.

[BJC15] Jakub BREIER, Dirmanto JAP et Chien-Ning CHEN. « Laser Profiling for the Back-Side Fault Attacks : With a Practical Laser Skip Instruction Attack on AES ». Dans : *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, CPSS 2015, Singapore, Republic of Singapore, April 14 - March 14, 2015*. Sous la dir. de Jianying ZHOU et Douglas JONES. ACM, 2015, p. 99-103.

[Bos18] Lilian BOSSUET. « Sécurité des systèmes embarqués ». Dans : *Techniques de l'ingénieur Sécurité des systèmes d'information* ref. article : h8280 (2018).

[Bou+11] A. BOUGEROL et al. « Experimental Demonstration of Pattern Influence on DRAM SEU and SEFI Radiation Sensitivities ». Dans : *IEEE Transactions on Nuclear Science* 58.3 (2011), p. 1032-1039.

[Bou+23] S. BOUAT et al. « X ray nanoprobe for fault attacks and circuit edits on 28-nm integrated circuits ». Dans : *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT 2023, Juan-Les-Pins, France, October 3-5, 2023*. Sous la dir. de Luca CASSANO et al. IEEE, 2023, p. 1-6.

[BP96] G. BRUGUIER et J.-M. PALAU. « Single particle-induced latchup ». Dans : *IEEE Transactions on Nuclear Science* 43.2 (1996), p. 522-532.

[BS03] Johannes BLÖMER et Jean-Pierre SEIFERT. « Fault Based Cryptanalysis of the Advanced Encryption Standard (AES) ». Dans : *Financial Cryptography, 7th International Conference, FC 2003, Guadeloupe, French West Indies, January 27-30, 2003, Revised Papers*. Sous la dir. de Rebecca N.

WRIGHT. T. 2742. Lecture Notes in Computer Science. Springer, 2003, p. 162-181.

[BS97] Eli BIHAM et Adi SHAMIR. « Differential Fault Analysis of Secret Key Cryptosystems ». Dans : *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*. Sous la dir. de Burton S. Kaliski JR. T. 1294. Lecture Notes in Computer Science. Springer, 1997, p. 513-525.

[Buc+13] Stephen P. BUCHNER et al. « Pulsed-Laser Testing for Single-Event Effects Investigations ». Dans : *IEEE Transactions on Nuclear Science* 60.3 (2013), p. 1852-1875.

[Cay+21] Pierre-Louis CAYREL et al. « Message-Recovery Laser Fault Injection Attack on the Classic McEliece Cryptosystem ». Dans : *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*. Sous la dir. d'Anne CANTEAUT et François-Xavier STANDAERT. T. 12697. Lecture Notes in Computer Science. Springer, 2021, p. 438-467.

[CCH22] Thomas CHAMELOT, Damien COUROUSSÉ et Karine HEYDEMANN. « SCI-FI : Control Signal, Code, and Control Flow Integrity against Fault Injection Attacks ». Dans : *2022 Design, Automation & Test in Europe Conference & Exhibition, DATE 2022, Antwerp, Belgium, March 14-23, 2022*. Sous la dir. de Cristiana BOLCHINI, Ingrid VERBAUWHEDE et Ioana VATAJELU. IEEE, 2022, p. 556-559.

[Cha+15] Clement CHAMPEIX et al. « Experimental validation of a Bulk Built-In Current Sensor for detecting laser-induced currents ». Dans : *21st IEEE International On-Line Testing Symposium, IOLTS 2015, Halkidiki, Greece, July 6-8, 2015*. IEEE, 2015, p. 150-155.

[Cha+17] Maxime CHAMBONNEAU et al. « Suppressing the memory state of floating gate transistors with repeated femtosecond laser backside irradiations ». Dans : *Applied Physics Letters* 110.16 (2017), p. 161112.

[Cha+21a] Émilie CHANAVAT et al. *Attaques d'un microcontrôleur par un banc d'injection LASER à 4 spots*. Youtube. 2021. URL : https://www.youtube.com/watch?v=9dgH57r9exU&list=PLWR7ZHocfRYYV_G4rAwB937XUolfFOzpH&index=6&t=44s.

[Cha+21b] Émilie CHANAVAT et al. *Microtroller attack with a 4-spot LASER*. Youtube. 2021. URL : https://www.youtube.com/watch?v=QY2N2B1fR3Q&list=PLWR7ZHocfRYYV_G4rAwB937XUolfFOzpH&index=7&t=1s.

[Cle+16] Ruan de CLERCQ et al. « SOFIA : Software and control flow integrity architecture ». Dans : *2016 Design, Automation & Test in Europe Conference & Exhibition, DATE 2016, Dresden, Germany, March 14-18, 2016*. Sous la dir. de Luca FANUCCI et Jürgen TEICH. IEEE, 2016, p. 1172-1177.

[Col+19] Brice COLOMBIER et al. « Laser-induced Single-bit Faults in Flash Memory : Instructions Corruption on a 32-bit Microcontroller ». Dans : *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2019, McLean, VA, USA, May 5-10, 2019*. IEEE, 2019, p. 1-10.

[Col+22] Brice COLOMBIER et al. « Multi-Spot Laser Fault Injection Setup : New Possibilities for Fault Injection Attacks ». Dans : *Smart Card Research and Advanced Applications*. Sous la dir. de Vincent GROSSO et Thomas PÖPPELMANN. Cham : Springer International Publishing, 2022, p. 151-166. ISBN : 978-3-030-97348-3.

[COM21] COMET. *MXR-165 datasheet*. 2021. URL : https://xray.comet.tech/getmedia/1ebeab27-fb14-42cf-9407-d8f49fb53347/mxr-165_single_sheet_en_v12.pdf?ext=.pdf&disposition=attachment.

[Com23a] Arthur H. COMPTON. « A Quantum Theory of the Scattering of X-rays by Light Elements ». Dans : *Phys. Rev.* 21 (5 mai 1923), p. 483-502.

[Com23b] Arthur H. COMPTON. « The Spectrum of Scattered X-Rays ». Dans : *Phys. Rev.* 22 (5 nov. 1923), p. 409-413.

[CPT17] Lucian COJOCAR, Kostas PAPAGIANNOPOULOS et Niek TIMMERS. « Instruction Duplication : Leaky and Not Too Fault-Tolerant ! » Dans : *Smart Card Research and Advanced Applications - 16th International Conference, CARDIS 2017, Lugano, Switzerland, November 13-15, 2017, Revised Selected Papers*. Sous la dir. de Thomas EISENBARTH et Yannick TEGLIA. T. 10728. Lecture Notes in Computer Science. Springer, 2017, p. 160-179.

[CY03] Chien-Ning CHEN et Sung-Ming YEN. « Differential Fault Analysis on AES Key Schedule and Some Coutnermeasures ». Dans : *Information Security and Privacy, 8th Australasian Conference, ACISP 2003, Wollongong, Australia, July 9-11, 2003, Proceedings*. Sous la dir. de Reihaneh SAFAVI-NAINI et Jennifer SEBERRY. T. 2727. Lecture Notes in Computer Science. Springer, 2003, p. 118-129.

[Dan+18] Jean-Luc DANGER et al. « CCFI-Cache : A Transparent and Flexible Hardware Protection for Code and Control-Flow Integrity ». Dans : *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018*. Sous la dir. de Martin NOVOTNÝ, Nikos KONOFAOS et Amund SKAVHAUG. IEEE Computer Society, 2018, p. 529-536.

[Deh+12] Amine DEHBAOUI et al. « Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES ». Dans : *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, Leuven, Belgium, September 9, 2012*. Sous la dir. de Guido BERTONI et Benedikt GIERLICHS. IEEE Computer Society, 2012, p. 7-15.

[Den05] Mickael DENAIS. « Étude des phénomènes de dégradation de type Negative Bias Temperature Instability (NBTI) dans les transistors MOS submicroniques des filières CMOS avancées ». Theses. Université de Provence - Aix-Marseille I, sept. 2005.

[Des+16] Chinmay DESHPANDE et al. « A Configurable and Lightweight Timing Monitor for Fault Attack Detection ». Dans : *IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2016, Pittsburgh, PA, USA, July 11-13, 2016*. IEEE Computer Society, 2016, p. 461-466.

[DLV03] Pierre DUSART, Gilles LETOURNEUX et Olivier VIVOLO. « Differential Fault Analysis on A.E.S ». Dans : *Applied Cryptography and Network Security, First International Conference, ACNS 2003. Kunming, China, October 16-19, 2003, Proceedings*. Sous la dir. de Jianying ZHOU, Moti YUNG et Yongfei HAN. T. 2846. Lecture Notes in Computer Science. Springer, 2003, p. 293-306.

[DM03] P.E. DODD et L.W. MASSENGILL. « Basic mechanisms and modeling of single-event upset in digital microelectronics ». Dans : *IEEE Transactions on Nuclear Science* 50.3 (2003), p. 583-602.

[Dou+05] Alexandre DOUIN et al. « Electrical Modeling for Laser Testing with Different Pulse Durations ». Dans : *11th IEEE International On-Line Testing Symposium (IOLTS 2005), 6-8 July 2005, Saint Raphael, France*. IEEE Computer Society, 2005, p. 9-13.

[DR02] Joan DAEMEN et Vincent RIJMEN. *The Design of Rijndael : AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[Dum+21] Mathieu DUMONT et al. « An Overview of Laser Injection against Embedded Neural Network Models ». Dans : *7th IEEE World Forum on Internet of Things, WF-IoT 2021, New Orleans, LA, USA, June 14 - July 31, 2021*. IEEE, 2021, p. 616-621.

[Dut+12] Jean-Max DUTERTRE et al. « Fault Round Modification Analysis of the advanced encryption standard ». Dans : *2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2012, San Francisco, CA, USA, June 3-4, 2012*. IEEE Computer Society, 2012, p. 140-145.

[Dut+18] Jean-Max DUTERTRE et al. « Laser Fault Injection at the CMOS 28 nm Technology Node : an Analysis of the Fault Model ». Dans : *2018 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2018, Amsterdam, The Netherlands, September 13, 2018*. IEEE Computer Society, 2018, p. 1-6.

[Dut+19] Jean-Max DUTERTRE et al. « Experimental Analysis of the Laser-Induced Instruction Skip Fault Model ». Dans : *Secure IT Systems - 24th Nordic Conference, NordSec 2019, Aalborg, Denmark, November 18-20, 2019, Proceedings*. Sous la dir. d'Aslan ASKAROV, René Rydhof HANSEN et Willard RAFNSSON. T. 11875. Lecture Notes in Computer Science. Springer, 2019, p. 221-237.

[ERM16] David EL-BAZE, Jean-Baptiste RIGAUD et Philippe MAURINE. « A fully-digital EM pulse detector ». Dans : *2016 Design, Automation & Test in Europe Conference & Exhibition, DATE 2016, Dresden, Germany, March 14-18, 2016*. Sous la dir. de Luca FANUCCI et Jürgen TEICH. IEEE, 2016, p. 439-444.

[Esa] *Single event effects test method and guidelines*. ESCC Basic Specification No. 25100. ESA. URL : https://escies.org/download/specdraftapppub?id=3095.

[Fal08] Alexandre FALLET. « Structure et propriétés mécaniques d'empilements aléatoires de sphères creuses : caractérisation et modélisation ». Theses. Institut Polytechnique de Grenoble, 2008.

[FG14] K. K. FUNG et W. B. GILBOY. « "Anode heel effect" on patient dose in lumbar spine radiography. » Dans : *British Journal of Radiology* 73.869 (mai 2014), p. 531-536. ISSN : 0007-1285.

[Fin+20] Christopher C FINLAY et al. « The CHAOS-7 geomagnetic field model and observed changes in the South Atlantic Anomaly ». Dans : *Earth, Planets and Space* 72.1 (2020), p. 156.

[Fou20] Denis FOURGERON. « Etude et mise en oeuvre de cellules résistantes aux radiations dans le cadre de l'évolution du détecteur à pixels d'Atlas technologie CMOS 65 nm ». Theses. Université de Toulon, 2020.

[Gai11] Rémi GAILLARD. « Single Event Effects : Mechanisms and Classification ». Dans : *Soft Errors in Modern Electronic Systems*. Sous la dir. de Michael NICOLAIDIS. Boston, MA : Springer US, 2011, p. 27-54. ISBN : 978-1-4419-6993-4.

[Gao+19] Chao GAO et al. « Microcontroller Based IoT System Firmware Security : Case Studies ». Dans : *IEEE International Conference on Industrial Internet, ICII 2019, Orlando, FL, USA, November 11-12, 2019*. IEEE, 2019, p. 200-209.

[GBD23] Paul GRANDAMME, Lilian BOSSUET et Jean-Max DUTERTRE. « X-Ray Fault Injection in Non-Volatile Memories on Power OFF Devices ». Dans : *2023 IEEE Physical Assurance and Inspection of Electronics (PAINE)*. 2023, p. 1-7.

[Ger+13] S. GERARDIN et al. « Radiation Effects in Flash Memories ». Dans : *IEEE Transactions on Nuclear Science* 60.3 (2013), p. 1953-1969.

[Gir04] Christophe GIRAUD. « DFA on AES ». Dans : *Advanced Encryption Standard - AES, 4th International Conference, AES 2004, Bonn, Germany, May 10-12, 2004, Revised Selected and Invited Papers*. Sous la dir. d'Hans DOBBERTIN, Vincent RIJMEN et Aleksandra SOWA. T. 3373. Lecture Notes in Computer Science. Springer, 2004, p. 27-41.

[Gou+23] Théophile GOUSSELOT et al. « Lightweight Countermeasures Against Original Linear Code Extraction Attacks on a RISC-V Core ». Dans : *2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2023, p. 89-99.

[Gra+14] Tibor GRASSER et al. « NBTI in Nanoscale MOSFETs—The Ultimate Modeling Benchmark ». Dans : *IEEE Transactions on Electron Devices* 61.11 (2014), p. 3586-3593.

[Gra22] Paul GRANDAMME. *Attaque laser de primitives de sécurité non alimentées*. Journée thématique sur les attaques par injection de fautes (JAIF 2022), Valence, France, Novembre 2022. 2022.

[Gra23a] Paul GRANDAMME. *Injection de fautes dans les circuits électroniques non alimentés*. Journée de la recherche de l'école doctorale EDSIS, Saint-Étienne, France, Juin 2023. 2023.

[Gra23b] Paul GRANDAMME. *X-Ray Fault Injection on Power OFF devices*. Cryptographic architectures embedded in logic devices (CryptArchi 2023), Cantabria, Espagne, Juin 2023. 2023.

[Gra23c] Paul GRANDAMME. *X-Ray Fault Injection on Power OFF Devices*. Journée des doctorants de l'équipe SAS, Gardanne, France, Juin 2023. 2023.

[Gra23d] Paul GRANDAMME. *X-Ray Fault Injection on Power OFF devices*. Journée thématique sur les attaques par injection de fautes (JAIF 2023), Gardanne, France, Septembre 2023. 2023.

[Gra+24] Paul GRANDAMME et al. « Switching Off your Device Does Not Protect Against Fault Attacks ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 4 (2024), p. 425-450.

[Gra24a] Paul GRANDAMME. *Éteindre votre composant électronique ne le protège pas !* Journée thématique sur les attaques par injection de fautes (JAIF 2024), Rennes, France, Octobre 2024. 2024.

[Gra24b] Paul GRANDAMME. *Laser Fault Injection on power off devices*. RADLAS 2024 : 6^{th} Workshop on Laser Testing of Radiation Effects on Components and Systems, Noordwijk, Pays-Bas, Septembre 2024. 2024.

[GST12] Benedikt GIERLICHS, Jörn-Marc SCHMIDT et Michael TUNSTALL. « Infective Computation and Dummy Rounds : Fault Protection for Block Ciphers without Check-before-Output ». Dans : *Progress in Cryptology - LATINCRYPT 2012 - 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings*. Sous la dir. d'Alejandro HEVIA et Gregory NEVEN. T. 7533. Lecture Notes in Computer Science. Springer, 2012, p. 305-321.

[Hab65] D. H. HABING. « The Use of Lasers to Simulate Radiation-Induced Transients in Semiconductor Devices and Circuits ». Dans : *IEEE Transactions on Nuclear Science* 12.5 (1965), p. 91-100.

[Ham] HAMAMATSU. *Hamamatsu L10711-03 datasheet*. URL : https://www.hamamatsu.com/eu/en/product/light-and-radiation-sources/microfocus-x-ray-source/L10711-03.html.

[HDP06] Vincent HUARD, M. DENAIS et C. R. PARTHASARATHY. « NBTI degradation : From physical mechanisms to modelling ». Dans : *Microelectron. Reliab.* 46.1 (2006), p. 1-23.

[HK89] Paul S HO et Thomas KWOK. « Electromigration in metals ». Dans : *Reports on Progress in Physics* 52.3 (1989), p. 301.

[Hom+14] Naofumi HOMMA et al. « EM Attack Is Non-invasive ? - Design Methodology and Validity Verification of EM Attack Sensor ». Dans : *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*. Sous la dir. de Lejla BATINA et Matthew ROBSHAW. T. 8731. Lecture Notes in Computer Science. Springer, 2014, p. 1-16.

[HS04] J. H. HUBBELL et S. M. SELTZER. *X-Ray Mass Attenuation Coefficients*. Rapp. tech. NIST Standard Reference Database 126. Gaithersburg, MD : National Institute of Standards et Technology, 2004. DOI : 10.18434/T4D01F.

[Hua+07] Vincent HUARD et al. « Design-in-Reliability Approach for NBTI and Hot-Carrier Degradations in Advanced Nodes ». Dans : *IEEE Transactions on Device and Materials Reliability* 7.4 (2007), p. 558-570.

[JMR07] Marc JOYE, Pascal MANET et Jean-Baptiste RIGAUD. « Strengthening hardware AES implementations against fault attacks ». Dans : *IET Inf. Secur.* 1.3 (2007), p. 106-110.

[KDD21] Vanthanh KHUAT, Jean-Luc DANGER et Jean-Max DUTERTRE. « Laser Fault Injection in a 32-bit Microcontroller : from the Flash Interface to the Execution Pipeline ». Dans : *2021 Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*. 2021, p. 74-85.

[Ker+88] Sherra E. KERNS et al. « The design of radiation-hardened ICs for space : a compendium of approaches ». Dans : *Proc. IEEE* 76.11 (1988), p. 1470-1509.

[KH14] Thomas KORAK et Michael HOEFLER. « On the Effects of Clock and Power Supply Tampering on Two Microcontroller Platforms ». Dans : *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2014, Busan, South Korea, September 23, 2014*. Sous la dir. d'Assia TRIA et Dooho CHOI. IEEE Computer Society, 2014, p. 8-17.

[KJJ99] Paul C. KOCHER, Joshua JAFFE et Benjamin JUN. « Differential Power Analysis ». Dans : *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. Sous la dir. de Michael J. WIENER. T. 1666. Lecture Notes in Computer Science. Springer, 1999, p. 388-397.

[KMW17] Martin S. KELLY, Keith MAYES et John F. WALKER. « Characterising a CPU fault attack model via run-time data analysis ». Dans : *2017 IEEE International Symposium on Hardware Oriented Security and Trust, HOST*

2017, McLean, VA, USA, May 1-5, 2017. IEEE Computer Society, 2017, p. 79-84.

[Koc96] Paul C. KOCHER. « Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems ». Dans : *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*. Sous la dir. de Neal KOBLITZ. T. 1109. Lecture Notes in Computer Science. Springer, 1996, p. 104-113.

[KQ08] Chong Hee KIM et Jean-Jacques QUISQUATER. « New Differential Fault Analysis on AES Key Schedule : Two Faults Are Enough ». Dans : *Smart Card Research and Advanced Applications, 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, London, UK, September 8-11, 2008. Proceedings*. Sous la dir. de Gilles GRIMAUD et François-Xavier STANDAERT. T. 5189. Lecture Notes in Computer Science. Springer, 2008, p. 48-60.

[Kra+16] Alexis KRAKOVINSKY et al. « Impact of a laser pulse on HfO2-based RRAM cells reliability and integrity ». Dans : *2016 International Conference on Microelectronic Test Structures (ICMTS)*. 2016, p. 152-156.

[Kra+17] Alexis KRAKOVINSKY et al. « Thermal laser attack and high temperature heating on HfO2-based OxRAM cells ». Dans : *23rd IEEE International Symposium on On-Line Testing and Robust System Design, IOLTS 2017, Thessaloniki, Greece, July 3-5, 2017*. IEEE, 2017, p. 85-89.

[Kum+18] Dilip S. V. KUMAR et al. « An In-Depth and Black-Box Characterization of the Effects of Laser Pulses on ATmega328P ». Dans : *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*. Sous la dir. de Begül BILGIN et Jean-Bernard FISCHER. T. 11389. Lecture Notes in Computer Science. Springer, 2018, p. 156-170.

[LG19] Haohao LIAO et Catherine H. GEBOTYS. « Methodology for EM Fault Injection : Charge-based Fault Model ». Dans : *Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019*. Sous la dir. de Jürgen TEICH et Franco FUMMI. IEEE, 2019, p. 256-259.

[LHB14] Jean-François LALANDE, Karine HEYDEMANN et Pascal BERTHOMÉ. « Software Countermeasures for Control Flow Integrity of Smart Card C Codes ». Dans : *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September*

7-11, 2014. Proceedings, Part II. Sous la dir. de Miroslaw KUTYLOWSKI et Jaideep VAIDYA. T. 8713. Lecture Notes in Computer Science. Springer, 2014, p. 200-218.

[Li+10] Yang LI et al. « Fault Sensitivity Analysis ». Dans : *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*. Sous la dir. de Stefan MANGARD et François-Xavier STANDAERT. T. 6225. Lecture Notes in Computer Science. Springer, 2010, p. 320-334.

[Lim+22] Rodrigo Silva LIMA et al. « Target Preparation Methodology for Semi-Invasive Attacks on Microcontrollers ». Dans : *2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)*. 2022, p. 1-7.

[Loh+18] Heiko LOHRKE et al. « Key Extraction Using Thermal Laser Stimulation A Case Study on Xilinx Ultrascale FPGAs ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.3 (2018), p. 573-595.

[Lop20] Israel Da Costa LOPES. « Méthodologie d'évaluation d'effets des radiations dans les systèmes numériques : du niveau composant au niveau système ». Theses. Université de Montpellier, 2020.

[Mai+21] Laurent MAINGAULT et al. « Laboratory X-rays Operando Single Bit Attacks on Flash Memory Cells ». Dans : *Smart Card Research and Advanced Applications - 20th International Conference, CARDIS 2021, Lübeck, Germany, November 11-12, 2021, Revised Selected Papers*. Sous la dir. de Vincent GROSSO et Thomas PÖPPELMANN. T. 13173. Lecture Notes in Computer Science. Springer, 2021, p. 139-150.

[Men+19] Alexandre MENU et al. « Precise Spatio-Temporal Electromagnetic Fault Injections on Data Transfers ». Dans : *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2019, Atlanta, GA, USA, August 24, 2019*. IEEE, 2019, p. 1-8.

[Men+20a] Alexandre MENU et al. « Experimental Analysis of the Electromagnetic Instruction Skip Fault Model ». Dans : *15th Design & Technology of Integrated Systems in Nanoscale Era, DTIS 2020, Marrakech, Morocco, April 1-3, 2020*. IEEE, 2020, p. 1-7.

[Men+20b] Alexandre MENU et al. « Single-bit Laser Fault Model in NOR Flash Memories : Analysis and Exploitation ». Dans : *17th Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2020, Milan, Italy, September 13, 2020*. IEEE, 2020, p. 41-48.

[Men21] Alexandre MENU. « Sécurité matérielle des objets connectés ». Thèse de doctorat. École des Mines de Saint-Étienne, nov. 2021.

[Mey23] Arnaud MEYER. « Optical-fiber-based distributed dosimetry for space applications ». Theses. Université Jean-Monnet - Saint-Étienne, 2023.

[MH19] Raymond L. MURRAY et Keith E. HOLBERT. *Nuclear energy : An introduction to the concepts, systems, and applications of nuclear processes*. English (US). Elsevier, jan. 2019. ISBN : 9780128128824.

[Mol18] Michael MOLL. « Displacement Damage in Silicon Detectors for High Energy Physics ». Dans : *IEEE Transactions on Nuclear Science* 65.8 (2018), p. 1561-1582.

[Moo+02] Simon W. MOORE et al. « Improving Smart Card Security Using Self-Timed Circuits ». Dans : *8th International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC 2002), 9-11 April 2002, Manchester, UK*. IEEE Computer Society, 2002, p. 211-218.

[Mor+13] Nicolas MORO et al. « Electromagnetic Fault Injection : Towards a Fault Model on a 32-bit Microcontroller ». Dans : *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*. Sous la dir. de Wieland FISCHER et Jörn-Marc SCHMIDT. IEEE Computer Society, 2013, p. 77-88.

[Mor+14] Nicolas MORO et al. « Formal verification of a software countermeasure against instruction skip attacks ». Dans : *J. Cryptogr. Eng.* 4.3 (2014), p. 145-156.

[MP19] MICRO-PACKS. *Micro-PackS : Votre plateforme technologique*. https://www.pf-micropacks.org/fr/micro-packs/la-plate-forme/. 2019.

[Muk09] Debdeep MUKHOPADHYAY. « An Improved Fault Based Attack of the Advanced Encryption Standard ». Dans : *Progress in Cryptology - AFRICACRYPT 2009, Second International Conference on Cryptology in Africa, Gammarth, Tunisia, June 21-25, 2009. Proceedings*. Sous la dir. de Bart PRENEEL. T. 5580. Lecture Notes in Computer Science. Springer, 2009, p. 421-434.

[MW79] T.C. MAY et M.H. WOODS. « Alpha-particle-induced soft errors in dynamic memories ». Dans : *IEEE Transactions on Electron Devices* 26.1 (1979), p. 2-9.

[Nak94] Werner NAKEL. « The elementary process of bremsstrahlung ». Dans : *Physics Reports* 243.6 (1994), p. 317-353.

[Net+06] Egas Henes NETO et al. « Using Bulk Built-in Current Sensors to Detect Soft Errors ». Dans : *IEEE Micro* 26.5 (2006), p. 10-18.

[OC14] Colin O'FLYNN et Zhizhang (David) CHEN. « ChipWhisperer : An Open-Source Platform for Hardware Embedded Security Research ». Dans : *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*. Sous la dir. d'Emmanuel PROUFF. T. 8622. Lecture Notes in Computer Science. Springer, 2014, p. 243-260.

[O'F16] Colin O'FLYNN. « Fault Injection using Crowbars on Embedded Systems ». Dans : *IACR Cryptol. ePrint Arch.* (2016), p. 810.

[PQ03] Gilles PIRET et Jean-Jacques QUISQUATER. « A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD ». Dans : *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*. Sous la dir. de Colin D. WALTER, Çetin Kaya KOÇ et Christof PAAR. T. 2779. Lecture Notes in Computer Science. Springer, 2003, p. 77-88.

[Pro+17] Julien PROY et al. « Compiler-Assisted Loop Hardening Against Fault Attacks ». Dans : *ACM Trans. Archit. Code Optim.* 14.4 (2017), 36 :1-36 :25.

[Puc+06] H. PUCHNER et al. « Elimination of Single Event Latchup in 90nm SRAM Technologies ». Dans : *2006 IEEE International Reliability Physics Symposium Proceedings*. 2006, p. 721-722.

[Pul24] PULSCAN. *Pulsys*. 2024. URL : https://www.pulscan.com/pages/pulsys.php.

[Rav18] Federico RAVOTTI. « Dosimetry Techniques and Radiation Test Facilities for Total Ionizing Dose Testing ». Dans : *IEEE Transactions on Nuclear Science* 65.8 (2018), p. 1440-1464.

[RDT13] Cyril ROSCIAN, Jean-Max DUTERTRE et Assia TRIA. « Frontside laser fault injection on cryptosystems - Application to the AES' last round - ». Dans : *2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013, Austin, TX, USA, June 2-3, 2013*. IEEE Computer Society, 2013, p. 119-124.

[Rog47] T. H. ROGERS. « High-Intensity Radiation from Beryllium-Window X-Ray Tubes ». Dans : *Radiology* 48.6 (1947), p. 594-603. DOI : 10.1148/48.6.594.

[Ros+13] Cyril ROSCIAN et al. « Fault Model Analysis of Laser-Induced Faults in SRAM Memory Cells ». Dans : *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*. Sous la dir. de Wieland FISCHER et Jörn-Marc SCHMIDT. IEEE Computer Society, 2013, p. 89-98.

[RPD09] Matthieu RIVAIN, Emmanuel PROUFF et Julien DOGET. « Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers ». Dans : *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Sous la dir. de Christophe CLAVIER et Kris GAJ. T. 5747. Lecture Notes in Computer Science. Springer, 2009, p. 171-188.

[San11] David SANDS. « Pulsed Laser Heating and Melting ». Dans : *Heat Transfer*. Sous la dir. de Vyacheslav S. VIKHRENKO. Rijeka : IntechOpen, 2011. Chap. 3.

[Sar+13a] Alexandre SARAFIANOS et al. « Building the electrical model of the pulsed photoelectric laser stimulation of an NMOS transistor in 90nm technology ». Dans : *2013 IEEE International Reliability Physics Symposium (IRPS)*. 2013.

[Sar+13b] Alexandre SARAFIANOS et al. « Electrical modeling of the photoelectric effect induced by a pulsed laser applied to an SRAM cell ». Dans : *Microelectron. Reliab.* 53.9-11 (2013), p. 1300-1305.

[Sch+03] J.R. SCHWANK et al. « Radiation effects in SOI technologies ». Dans : *IEEE Transactions on Nuclear Science* 50.3 (2003), p. 522-538.

[Sch07] Dieter K. SCHRODER. « Negative bias temperature instability : What do we understand ? » Dans : *Microelectron. Reliab.* 47.6 (2007), p. 841-852.

[Sch+16] Falk SCHELLENBERG et al. « Large laser spots and fault sensitivity analysis ». Dans : *2016 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2016, McLean, VA, USA, May 3-5, 2016*. Sous la dir. de William H. ROBINSON, Swarup BHUNIA et Ryan KASTNER. IEEE Computer Society, 2016, p. 203-208.

[Sch94] J R SCHWANK. « Basic mechanisms of radiation effects in the natural space radiation environment ». Dans : (juin 1994).

[Sei04] J. Anthony SEIBERT. « X-Ray Imaging Physics for Nuclear Medicine Technologists. Part 1 : Basic Principles of X-Ray Production ». Dans : *Journal of Nuclear Medicine Technology* 32.3 (2004), p. 139-147. ISSN : 0091-4916.

[Sel+11] S. SELTZER et al. « Fundamental Quantities and Units for Ionizing Radiation ». Dans : *Journal of the ICRU* 11 (avr. 2011).

[SFM20] Junichi SAKAMOTO, Daisuke FUJIMOTO et Tsutomu MATSUMOTO. « Laser-Induced Controllable Instruction Replacement Fault Attack ». Dans : *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* 103-A.1 (2020), p. 11-20.

[SH08] Jörn-Marc SCHMIDT et Christoph HERBST. « A Practical Fault Attack on Square and Multiply ». Dans : *Fifth International Workshop on Fault Diagnosis and Tolerance in Cryptography, 2008, FDTC 2008, Washington, DC, USA, 10 August 2008*. Sous la dir. de Luca BREVEGLIERI et al. IEEE Computer Society, 2008, p. 53-58.

[Sha02] Ashok K. SHARMA. *Semiconductor Memories : Technology, Testing and Reliability*. Wiley-IEEE Press, sept. 2002. ISBN : 978-0-780-31000-1.

[SHP09] Jörn-Marc SCHMIDT, Michael HUTTER et Thomas PLOS. « Optical Fault Attacks on AES : A Threat in Violet ». Dans : *Sixth International Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2009, Lausanne, Switzerland, 6 September 2009*. Sous la dir. de Luca BREVEGLIERI et al. IEEE Computer Society, 2009, p. 13-22.

[Sko05] Sergei P. SKOROBOGATOV. « Data Remanence in Flash Memory Devices ». Dans : *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*. Sous la dir. de Josyula R. RAO et Berk SUNAR. T. 3659. Lecture Notes in Computer Science. Springer, 2005, p. 339-353.

[Sko09] Sergei P. SKOROBOGATOV. « Local Heating Attacks on Flash Memory Devices ». Dans : *IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2009, San Francisco, CA, USA, July 27, 2009. Proceedings*. Sous la dir. de Mohammad TEHRANIPOOR et Jim PLUSQUELLIC. IEEE Computer Society, 2009, p. 1-6.

[Sko10a] Sergei SKOROBOGATOV. « Flash Memory 'Bumping' Attacks ». Dans : *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*. Sous la dir. de Stefan MANGARD et François-Xavier STANDAERT. T. 6225. Lecture Notes in Computer Science. Springer, 2010, p. 158-172.

[Sko10b] Sergei SKOROBOGATOV. « Optical Fault Masking Attacks ». Dans : *2010 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2010,*

Santa Barbara, California, USA, 21 August 2010. Sous la dir. de Luca BREVEGLIERI et al. IEEE Computer Society, 2010, p. 23-29.

[Sol+22] Hadi SOLEIMANY et al. « Practical Multiple Persistent Faults Analysis ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2022.1 (2022), p. 367-390.

[SP13] J. R. SROUR et J. W. PALKO. « Displacement Damage Effects in Irradiated Semiconductor Devices ». Dans : *IEEE Transactions on Nuclear Science* 60.3 (2013), p. 1740-1766.

[ST01] National Institute of STANDARDS et TECHNOLOGY. « Advanced Encryption Standard ». Dans : *NIST FIPS PUB 197* (2001).

[Stm] *Programming manuel : STM32F100xx value line Flash programming*. Manual PM0063. STMicroelectronics, 2010.

[TBG23] Pierre-Antoine TISSOT, Lilian BOSSUET et Vincent GROSSO. « BALoo : First and Efficient Countermeasure dedicated to Persistent Fault Attacks ». Dans : *IACR Cryptol. ePrint Arch.* (2023), p. 944.

[THI13] Christian THIERY. « Tomographie à rayons X ». Dans : *Techniques de l'ingénieur Techniques d'analyse* base documentaire : TIP630WEB.ref. article : p950 (2013).

[TK10] Elena TRICHINA et Roman KORKIKYAN. « Multi Fault Laser Attacks on Protected CRT-RSA ». Dans : *2010 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2010, Santa Barbara, California, USA, 21 August 2010*. Sous la dir. de Luca BREVEGLIERI et al. IEEE Computer Society, 2010, p. 75-86.

[Val00] Vladivoj VALKOVIC. « Measurements of Radioactivity ». Dans : déc. 2000, p. 117-258. ISBN : 9780444829542.

[Vas+17] Aurélien VASSELLE et al. « Laser-Induced Fault Injection on Smartphone Bypassing the Secure Boot ». Dans : *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2017, Taipei, Taiwan, September 25, 2017*. IEEE Computer Society, 2017, p. 41-48.

[VDL22] Raphael VIERA, Jean-Max DUTERTRE et Rodrigo Silva LIMA. « Injecting Permanent Faults into the Flash Memory of a Microcontroller with Laser Illumination During Read Operations ». Dans : *Proceedings of the 2022 Workshop on Attacks and Solutions in Hardware Security, ASHES 2022, Los Angeles, CA, USA, 11 November 2022*. Sous la dir. de Chip-Hong CHANG et al. ACM, 2022, p. 75-84.

[Vie+21] Raphael Andreoni Camponogara VIERA et al. « Permanent Laser Fault Injection into the Flash Memory of a Microcontroller ». Dans : *19th IEEE International New Circuits and Systems Conference, NEWCAS 2021, Toulon, France, June 13-16, 2021*. IEEE, 2021, p. 1-4.

[Vie+23] Raphael VIERA et al. « Tampering with the flash memory of microcontrollers : permanent fault injection via laser illumination during read operations ». Dans : *Journal of Cryptographic Engineering* 14 (2023), p. 207-221.

[Vie+24] Raphael VIERA et al. « Tampering with the flash memory of microcontrollers : permanent fault injection via laser illumination during read operations ». Dans : *Journal of Cryptographic Engineering* 14 (juin 2024), p. 207-221. ISSN : 2190-8516.

[Vij+17] Arunkumar VIJAYAKUMAR et al. « Physical Design Obfuscation of Hardware : A Comprehensive Investigation of Device and Logic-Level Techniques ». Dans : *IEEE Trans. Inf. Forensics Secur.* 12.1 (2017), p. 64-77.

[VOC18] Sebastian VASILE, David F. OSWALD et Tom CHOTHIA. « Breaking All the Things - A Systematic Survey of Firmware Extraction Techniques for IoT Devices ». Dans : *Smart Card Research and Advanced Applications, 17th International Conference, CARDIS 2018, Montpellier, France, November 12-14, 2018, Revised Selected Papers*. Sous la dir. de Begül BILGIN et Jean-Bernard FISCHER. T. 11389. Lecture Notes in Computer Science. Springer, 2018, p. 171-185.

[Wag12] Mathias WAGNER. « 700+ Attacks Published on Smart Cards : The Need for a Systematic Counter Strategy ». Dans : *Constructive Side-Channel Analysis and Secure Design - Third International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings*. Sous la dir. de Werner SCHINDLER et Sorin A. HUSS. T. 7275. Lecture Notes in Computer Science. Springer, 2012, p. 33-38.

[YC96] Peter Y. YU et Manuel CARDONA. *Fundamentals of Semiconductors*. Graduate Texts in Physics. Springer Berlin, Heidelberg, 1996.

[Zha+18] Fan ZHANG et al. « Persistent Fault Analysis on Block Ciphers ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.3 (2018), p. 150-172.

[Zha+20] Fan ZHANG et al. « Persistent Fault Attack in Practice ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2020.2 (2020), p. 172-195.

[Zha+23] Fan ZHANG et al. « Efficient Persistent Fault Analysis with Small Number of Chosen Plaintexts ». Dans : *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2023.2 (2023), p. 519-542.

[Zus+14] Loïc ZUSSA et al. « Analysis of the fault injection mechanism related to negative and positive power supply glitches using an on-chip voltmeter ». Dans : *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014*. IEEE Computer Society, 2014, p. 130-135.