Technical Report

UCAM-CL-TR-630
ISSN 1476-2986

Number 630

UNIVERSITY OF
CAMBRIDGE
Computer Laboratory

Semi-invasive attacks – A new approach to hardware security analysis

Sergei P. Skorobogatov

April 2005

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted September 2004 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Darwin College.

ISSN 1476-2986

Abstract

Semiconductor chips are used today not only to control systems, but also to protect them against security threats. A continuous battle is waged between manufacturers who invent new security solutions, learning their lessons from previous mistakes, and the hacker community, constantly trying to break implemented protections. Some chip manufacturers do not pay enough attention to the proper design and testing of protection mechanisms. Even where they claim their products are highly secure, they do not guarantee this and do not take any responsibility if a device is compromised. In this situation, it is crucial for the design engineer to have a convenient and reliable method of testing secure chips.

This thesis presents a wide range of attacks on hardware security in microcontrollers and smartcards. This includes already known non-invasive attacks, such as power analysis and glitching, and invasive attacks, such as reverse engineering and microprobing. A new class of attacks – semi-invasive attacks – is introduced. Like invasive attacks, they require depackaging the chip to get access to its surface. But the passivation layer remains intact, as these methods do not require electrical contact to internal lines. Semi-invasive attacks stand between non-invasive and invasive attacks. They represent a greater threat to hardware security, as they are almost as effective as invasive attacks but can be low-cost like non-invasive attacks.

This thesis' contribution includes practical fault-injection attacks to modify SRAM and EEPROM content, or change the state of any individual CMOS transistor on a chip. This leads to almost unlimited capabilities to control chip operation and circumvent protection mechanisms. A second contribution consist of experiments on data remanence, which show that it is feasible to extract information from powered-off SRAM and erased EPROM, EEPROM and Flash memory devices.

A brief introduction to copy protection in microcontrollers is given. Hardware security evaluation techniques using semi-invasive methods are introduced. They should help developers to make a proper selection of components according to the required level of security. Various defence technologies are discussed, from low-cost obscurity methods to new approaches in silicon design.

Acknowledgements

I would like to thank my supervisor Ross Anderson for his help throughout my research and for encouraging my initial experiments with Static RAM data remanence which in the end reveal much wider problem with data remanence in a wide range of semiconductor memory devices. Without his help and promotion my initial discovery of fault injection attacks would not have become so widely known. His help in proofreading my papers was invaluable. I would like to thank Markus Kuhn for his helpful discussions and Richard Clayton for his help in proofreading. I am grateful to the Department of Materials Science and Metallurgy for allowing access to a FIB machine and fume cupboard, and in particular to Dae-Joon Kang and Nadia Stelmashenko. I would like to thank Hyun-Jin Choi for his help in FIB work. I am very grateful to Specialised Electronic Systems for their help in modelling the fault injection attacks and laser scanning technique. I would like to thank my father Peter Skorobogatov for his helpful discussions on semiconductor physics. The purchase of some equipment used was made possible through the TAMPER hardware security laboratory support provided by NDS and Hitachi. I would like to thank Radiolinija for allowing me access to their semiconductor testing equipment necessary for some of my research. My research was supported by a European G3Card project I took part in. I would like to thank Simon Moore who was the local coordinator of this project in the Computer Laboratory.

Disclaimer

I do not accept any responsibility or liability for loss or damage occasioned to any person or property through using material, instructions, methods or ideas contained herein, or acting or refraining from acting as a result of such use. The reader must be aware of the danger involved in some operations and refer to health and safety warnings for each particular product used. In case of any doubt please seek for professional advice.

The potential hazard involves:

- Chemicals used for decapsulation and deprocessing. They contain very strong acids and alkalines and could cause severe burns to eyes and skin. Adequate protective goggles and gloves must be worn.
- Class 3B laser products used for depassivation and class 3R laser products used for laser scanning and fault injection (visible and invisible laser radiation). Avoid eye and skin exposure to direct and reflected radiation. Lasers can cause permanent damage to eyes and severe burns to skin. Appropriate protective goggles must be worn.
- UV light used for erasing on-chip memories. Avoid eye and skin exposure, as these can damage eyes and skin. Appropriate protective goggles must be worn.

Contents

Chapter 1

Introduction

Semiconductor chips are everywhere around us – from computers, cars, TV sets and mobile phones to mp3 players, washing machines, microwave ovens and phone cards.

With constantly growing demand for security, silicon chips started to be used not only for control purposes but for protection as well. The last ten years have seen a big boom in this area. From military and bank applications, the technology moved to everyday life: to prevent the use of unbranded batteries in mobile phones and laptops, to block non-genuine and refilled cartridges for printers, and to restrict the servicing of your appliances to manufacturer service centres.

These days we have a continuous battle between manufacturers who invent new security solutions learning their lessons from previous mistakes, and the hacker community which is constantly trying to break the protection in various devices. Both sides are also constantly improving their knowledge and experience. In this endless war, the front line shifts forward and backward regularly. Deep down, the problem concerns both economics and law. On the one hand, when dishonest people try to steal property, there will be a demand to increase security. On the other, reverse engineering was always part of technological progress, helping to design compatible products and improve existing ones. The dividing line between legal (reverse engineering) and illegal (piracy) is difficult.

Unfortunately very little attention is paid to proper selection of microcontrollers for secure applications. Mainly this happens because information about the true level of security protection is not widely available from manufacturers or distorted. Chip manufacturers do not pay much attention to the proper design and testing of protection mechanisms. Even where they claim their products are highly secure, they do not guarantee this and do not take any responsibility if the device is compromised. In this situation it is crucial for the design engineer to have a convenient and reliable method of testing secure chips.

In this thesis I try to cover a wide range of problems with the hardware security of silicon chips. As such research is an endless process due to continuous technological progress, I had to choose a narrow area of security analysis in microcontrollers and only slightly touched evaluation of smartcards. Although smartcards offer a greater level of security protection, they are based on the same core design with extra security features added to protect against various kinds of attack. Some attacks, usually invasive, can still be applied to smartcards, but compared to microcontrollers they will require more effort and therefore would be more expensive and difficult to carry out.

There is no such thing as absolute security. A determined hacker can break any protection provided he has enough time and resources. The question is how practical it would be. If it takes ten years to break a device which in three years time is replaced by a successor with even better security, then the defence has won. On the other hand, the vulnerability could be buried within the design blocks itself. What if your secure system was designed from insecure components? In the end, the overall security of your system is determined by the least secure element. Even if you implement a provably secure protocol, your system could be broken if the key can be easily extracted from the hardware by mechanical or optical probing. Therefore, whenever you design a secure system, proper security evaluation of all the components must be performed. Of course it is impossible to avoid all problems; a reasonable goal is to make the process of breaking your design more expensive and time-consuming. With luck, potential attackers will switch to other products rather than spending money and effort on breaking yours.

It is obvious that one of the first steps in any hardware design is choosing the right components. Despite all the electrical, performance and resource parameters which are widely available from all semiconductor manufacturers, information on security protection and implementation is either limited or totally restricted. That forces the designers to evaluate security-sensitive components themselves or hire other companies for that job. I have tried to give some ideas on hardware security testing throughout my thesis.

A new class of attack – the semi-invasive attack – was recently introduced [1]. Using semi-invasive methods for security evaluation could expose more problems in the hardware design with less effort, and in a shorter time, compared to invasive or non-invasive methods [2]. The archetypal examples of such attacks are fault injection and data remanence, which we are in a process of learning. I would expect more achievements in this area within the next few years.

I have tried to make this thesis acceptable to a wide audience – from embedded designers interested in protection of their intellectual property in microcontrollers, to security analysts in large manufacturing companies. I hope everyone will find something interesting and new in this thesis.

1.1 Previous work and knowledge

Hardware security analysis requires broad knowledge in many areas. Firstly, you have to know the subject of your analysis. That implies, in case of microcontrollers, an ability to write small test programs in assembler and C, and to use development and debugging tools, universal programmers and test equipment like signal generators, oscilloscopes and logic analysers. Performing simple attacks involves using special electronic tools to manipulate the signals applied to the chip. That requires building interface boards and writing programs on a PC for controlling them. More powerful and complex attacks require direct access to the chip surface. For that, some minimal knowledge and experience in chemistry is necessary, as well as the use of a microscope and microprobing station. Once you get down to the chip layout you need some basic knowledge of silicon design. That comes from microelectronics and could be very challenging task, especially for modern chips. There is a lot of literature available on this subject [3][4] and lots of information on the Internet [5][6], but it takes time to assimilate, especially with deep submicron technologies. Also, the many different design approaches and technological processes used by each individual manufacturer make the analysis more difficult. Common engineering work is required for building special tools and adapters necessary for some experiments. And finally, broad knowledge of physics is needed throughout many of the experiments.

I have tried to cover all these aspects in my thesis, giving the basic idea behind each step involved. As many of the tools required for detailed analysis are either too expensive or not available at all, I had to improvise a lot. This work would be different for well equipped laboratories as they could buy everything they wish. On the other hand, one cannot buy something that has not been built yet or is only being designed. In this case the creative researcher is always ahead as he could build such tools for innovative analysis. Unfortunately academic researchers have very limited funding compared to industrial and government funded laboratories. Therefore only a limited number of invasive methods can be taken into consideration. That was the main reason I paid so much attention to semi-invasive methods as they require much less investment in equipment. Meantime, as it will be seen in this thesis, they give very good results and can be very effective. At the same time, low-cost attacks represent the most severe threat to secure systems as they could be repeated by a large number of individuals. If breaking a secure device requires a multimillion dollar investment, then only a very well-funded laboratory would be able to carry out the work. Even if they succeed, they will be unlikely to give away the information and results for free. In other words, there will be not only technological but economic barriers against attacking such highly secure devices. Also such laboratories usually collaborate, or are directly funded by, large semiconductor manufacturers or governments, and do not do any work for untrustworthy customers.

I travelled a very long way to the hardware security subject. At school, electronics was my hobby. In 1987 I learnt my first microcontroller – Intel 8048. As I did not have any access to a computer or even a programmer device, I wrote my first programs in machine code and then programmed the bytes of the code into a 2716 EPROM chip connected as an external memory to the microcontroller. Of course my programs were quite small, the maximum I did was a programmable timer from one to thousand seconds and a simple calculator. But what I learned from doing such projects was invaluable.

In 1989 I got my first computer – a Sinclair ZX Spectrum 48K. A year later I started building computer external hardware modules like EPROM programmers and ROM emulators, and writing different programs in assembler for the Zilog Z80 processor used in it. All those projects were a hobby, but that was probably the first time I started reverse engineering code. There were two reasons for that. One was to learn the assembler language better through understanding disassembled code; another was to modify the programs, for example, to add extra features. Also I used to modify software drivers to get them working with non-standard hardware equipment. After two years, when IBM PC compatible computers became widely used, I switched to them. I started at the bottom end – with an XT based on an Intel 8088 processor. I continued developing external hardware devices, but for programming I started using C++ language.

My first security-related project was started in 1995 with testing the security protection in the Motorola MC68HC05 and MC68HC11 microcontroller families against low cost attacks. Some of these attacks were repeated later as a part of my research into non-invasive attacks.

A year later I was reverse engineering the program from a Microchip PIC16C57 microcontroller used in a security access system. The aim of this work was to evaluate how secure the system was from the point of view of non-invasive attacks. It was a door access control system with only two contacts, for an iButton key and an LED for indication. I found some problems that made timing attacks possible, so in the end the design was improved to withstand such attacks. From that I learnt about PIC controllers and started using them for all my further hardware projects.

My Master's thesis project at university was connected with hardware security as well. It involved designing and debugging a cash control monitor – the device used in a supermarket till to store the information on all purchases. As a part of this project, I built a prototype of a PC card with a microcontroller on it and a memory for data storage. Different levels of data access control were implemented inside the microcontroller's program.

After my graduation in 1997, I worked part-time for an ophthalmology company designing microcontroller-based special hardware devices for eyesight testing, training and correction. As these devices had commercial value, I was asked to make them as hard to clone as possible. That was when I got really interested in learning hardware security. In 1999, playing with power glitch attacks, I found a vulnerability in PIC16F84 microcontroller that in the end turned out to be a much bigger problem than simply for that particular device. Only four years later, I realised that it was possible because of the data remanence effect.

Before I came to Cambridge in 2000, all my research on the hardware security of microcontrollers was on non-invasive attacks. Only here at the Computer Laboratory I started learning about invasive attacks and in 2001 I accidentally discovered a new class of attacks – fault injection. We defined this new class of attacks as semi-invasive. By this we mean that, although the access to the chip die surface is required, the passivation layer and the internal wires are not destroyed or physically contacted. This thesis represents the result of almost four years of my research into hardware security.

1.2 The subject of hardware security

Each system has different levels of design with security features normally implemented at each level. The highest level belongs to communication protocols and human interfaces. It might also involve all sorts of restrictions and access control to the building or room where the equipment is running. The application software level supports all the external interfaces and communication protocols and may also do encryption and authentication. It runs under the control of operating system which has built-in security for authentication, encryption and protection of sensitive information. Highly secure systems, for example, those used in bank applications have hardware tamper resistant modules that keep the secret key inside and perform all critical cryptographic operations. The software communicates with the secure module through the application programming interface (API). This interface is implemented in software running inside the module in a secure environment that prevents anyone downloading or modifying it. The module itself is a hardware device normally implemented as a PCI card for a standard PC board. The program inside this module is stored in battery backed-up SRAM and can be destroyed within seconds if any tampering is detected. The printed circuit board (PCB) with security critical components and sensors against low temperature and ionizing radiation is enclosed within a tamper-sensing mesh and potting material. The whole construction has electromagnetic shielding [7]. In terms of hardware security we will be interested in everything that is located inside this secure module box from the security of the silicon chips buried somewhere on the chip die to embedded software running in the protected environment.

Sometimes it is not quite obvious whether a certain part should be classified as software or hardware. If we have a program running on a PC and providing access to confidential information after entering the right password, then it is obviously software security. If the same sort of access is provided by plugging in a special USB dongle, then the dongle itself is a hardware security device. But what if you have a tamper-resistant secure module with software running inside it? Obviously, there is no firm line between software and hardware, because the first cannot run without the second.

Examples of hardware secure devices are microcontrollers, complex programmable logic devices (CPLDs), field programmable gate arrays (FPGAs), smartcards and secure tamper-resistant modules. They are used in a wide range of applications, from industrial control and wireless communication to bank payment cards, pay-TV applications and building access control.

A good example of hardware security progress is pay-TV [8]. Around 1994 providers started using smartcards for conditional access control. At the same time a large community of hackers was trying to reverse engineer and clone these cards to get free access to the contents of cable and satellite channels. The very first cards were quite simple and had lots of vulnerabilities, like sensitivity to UV light, power fluctuation and reduced clock rates. Service providers learned lessons from this and significantly improved not only the smartcards they used, but the encryption protocols too. Yet after several months determined hackers were again able to find a hole in the security. That again forced the providers into a cycle of hardware security improvements.

Attackers very often are ahead of the defenders for some obvious reasons. Firstly, a single vulnerability could let an attacker succeed, while a defender must protect against all known attacks. Secondly, even if the device can withstand all known attacks, no one can guarantee that a new attack will not be discovered. Initially, security-related bugs are always present in software. The question is who finds one first, and either fixes or exploits it. All these problems force the developer of a secure system to look for better ways of designing and exhaustive testing of his system. If the key element of his system is a standard smartcard, or off-the-shelf secure chip, he should test it properly, to be sure it will not be broken and reverse engineered in a few weeks time. One way might be to hire a specialised security evaluation company, but the better way, especially for large manufacturers, would be to build his own research laboratory. In this case he will not leak any confidential information, and also do it faster.

This thesis gives some idea of what sort of evaluation can be done by a qualified engineer using widely available and not very expensive equipment. Of course, for proper evaluation of smartcards one will need much more sophisticated equipment, but the basic approach could be the same.

With microcontrollers the first question could be: ‘Which chip is the best from the security point of view?’ The question is not practical as it will involve testing hundreds of different microcontrollers and in the end the ‘best’ might not be suitable for the required application. The better approach is to compile a list of microcontrollers suitable for the desired application and then test them against various kinds of attacks. Then you have to find a trade off between the price you will pay for the extra security and the actual protection it will give you. In the end you have to think about who will be likely to attack your device. If they are inexperienced, and the maximum they could do is to try reading the microcontroller used inside your device in a self-made programmer, then most microcontrollers available on the market will provide adequate protection. If you consider your design to be valuable and you think about competitors, then the decision will depend on the amount of money and time attackers could spend trying to reverse-engineer your device. Again, if you worry only about low-cost attacks, then a suitable secure microcontroller could be easily found and you will have to spend some time and money to evaluate it against all known low cost attacks to be sure that nothing will go wrong. If your project is very important and valuable and you are going to invest a lot of effort into algorithm and code development, or you want to provide conditional access control and keep your keys secret, then you must be very careful with your selection. Very likely you will have to choose between different smartcard vendors and not from microcontrollers as they are very unlikely to survive a well equipped and determined attacker.

Another important thing that should be pointed out is that the only practical and reliable way of comparing security in different microcontrollers and smartcards is by estimating the cost of a successful attack which will involve actual breaking into the devices. In other words, a person or a company that carries out the security evaluation should be able to break the security in almost any device within their area of interest. Otherwise the evaluation will be incomplete and theoretical only. In practice it is very difficult to estimate how capable that tester is and how close his report will be to the real situation with the security in a particular device. Another problem that might arise is if he deliberately reports a higher level of security for the tested device in order to prevent you from increasing the security, thus making easier for him to access the information stored in your device later. Unfortunately, very often security evaluation is done in theory based on the information provided by the chip manufacturer and not on proper tedious research. That does not give the real picture of the security and actual cost of any possible attack.

Of course no-one could guarantee that a certain chip will not be cloned or reverse engineered, but proper selection could significantly delay this event. With the speed of technological progress, one year is enough to make an existing product less attractive, and usually it will be either replaced or upgraded. Such upgrade could involve replacement of the security sensitive parts with better versions.

However, as things stand, the security of a standard microcontroller could be significantly improved and some ideas are given in Chapter 8 of this thesis.

1.3 Motivation and overview

This thesis ignores some serious problems in hardware security design of silicon chips. For technical reasons, I had to choose to cover mainly the security analysis of microcontrollers and I only touched lightly on the evaluation of smartcards. I did my best with the equipment I had access to, but it fell far short of what large security analysis laboratories have. Therefore some methods are only mentioned without testing them on actual chips.

I tried to draw a picture of the situation with hardware security in silicon chips from both the design and the evaluation sides. Some low cost defence technologies were suggested together with a possible approach to silicon circuit design that could significantly improve the overall hardware security of chips.

In spite of the poor equipment I have access to, I am very proud of the progress I have done here in the Computer Laboratory during my research. The most important achievement is the discovery of fault injection attacks and defining the new area of attacks – semi-invasive. As such attacks do not require direct mechanical access to the chip surface they are cheaper and easier to apply than invasive ones. Using semi-invasive methods for security evaluation could expose the design problems in days, whereas finding them with conventional invasive methods would take weeks and months, and non-invasive methods might not help at all.

I tried to give as many examples as possible of different security protection schemes in modern microcontrollers. Because the equipment I used for my experiments was too far from modern, I have only very briefly touched hardware security in smartcards, secure ICs, CPLDs and FPGAs.

Chapter 2 of this thesis has some background information about the area of my research. It gives a basic overview of silicon chips' evolution from the technological and hardware security points of view, and information about different types of memory used in microcontrollers. This chapter also discusses the reasons why hardware security is important and popular, and who actually needs it. The last part of the chapter gives an introduction to the crucial part of integrated circuit design – failure analysis. On the one hand, silicon design engineers want to find any design problems that occurred. On the other, if a manufacturer has a tool that allows him to look closely at any point of the chip, then someone else could use it to find a vulnerability in the design or to extract sensitive data.

A general introduction to attack technologies is given in Chapter 3. A classification of attackers and attack scenarios is given, as well as a definition and short introduction into each type of attack. A detailed classification of protection levels is given. This makes the comparison of security in different microcontrollers easier.

Chapter 4 gives examples of non-invasive attacks. These attacks do not require large investments in equipment. It is wrongly assumed that these attacks can be developed by inexperienced hackers with minimal equipment and knowledge. Once found, they can be repeated by almost anyone, but the process of finding the attack could be very difficult and time-consuming. Sometimes one will have to fully reverse engineer the silicon chip to find a vulnerability which could be exploited. This chapter also discusses power analysis techniques, which can sometimes help a lot in understanding the behaviour of a chip. For example, one can easily figure out what command the processor executes, or distinguish between different results of arithmetic operations.

Power and clock glitch attacks are widely used to circumvent many types of protection incorporated in embedded software including smartcards. Also power glitches can be used to break hardware protection and some examples are given in Chapter 4.

Another important area I tried to expose and attract attention to is data remanence in on-chip memories and separate memory devices. This effect could leak a lot of sensitive and secure information after the memory contents are deliberately deleted, by either disconnecting the power supply in case of volatile memories, like DRAM and SRAM, or erasing non-volatile memories, like EPROM, EEPROM and Flash. Finally this chapter presents experiments on data remanence in EPROM, EEPROM and Flash non-volatile memories.

Invasive attacks are discussed in Chapter 5. Starting from an introduction into sample preparation techniques, it gives some idea on what well-equipped and experienced attackers could do with the chip. Although such techniques are well known and straightforward, an attacker will need to have at least basic knowledge about silicon design and be familiar with the different technologies used for semiconductor manufacturing. The main obstacle to these attacks is the cost of the necessary equipment, which could be enormous for attacking smartcards and modern secure chips.

Chapter 6 introduces and discusses the area of semi-invasive attacks. It starts from well known ultraviolet light (UV) attacks, which were counted as invasive attacks before, but actually do not require physical contact to the chip surface or any modifications to the die. Imaging techniques can be counted as semi-invasive as well and they are widely used for partial reverse engineering. Using laser-scanning microscopy one can get significantly more information out of the chip – not only the location of a transistor’s active areas, but even its logic state. Both optical imaging and laser scanning can be done through the silicon substrate from the rear side of the chip. This becomes more important for modern chips where multiple top-metal layers that cover the surface prevent us seeing any structures on a die.

Fault injection is another shining example of semi-invasive attacks. With this technique, an attacker can change the state of any individual transistor within the chip. That gives him almost unlimited capabilities in understanding the functions of any block within the design and circumventing many security protections. Some examples of these attacks applied to microcontrollers are given in Chapter 7. Fault injection can be used to break some encryption algorithms as well. The idea was discussed in the late nineties but no practical means were suggested at the time [9][10][11]. Being synchronised with the system clock and program flow, fault injection can be used as a perfect glitch attack. This is because power and clock glitches usually affect a large number of transistors on the chip whereas fault injection can be focused down to any individual transistor.

Chip manufacturers are constantly acting upon the security problems. Chapter 7 shows some examples of progress in defending against different kind of attacks. Common problems in the security protection design of various chips are also discussed. Technological progress and the continuous shrinking of the chip elements makes invasive attacks very difficult and expensive, and also introduces some problems in implementation of certain semi-invasive attacks. Modern chips with wide data buses, pipelining and instruction caches make power analysis more difficult but not impossible. Different schemes of security implementation are discussed.

Chapter 8 shows various methods of protection against different kind of attacks. There are low budget ways based on obscurity, through to expensive protection techniques requiring quite different approaches to chip design, like asynchronous logic and dual-rail logic.

Finally Chapter 9 summarises the work and achievements of my research. It also proposes directions for further research.

The Appendix contains summarised information on a wide range of microcontrollers and some smartcards.

Chapter 2

Background

With the release of the first microprocessor, the 4004, by Intel in 1971, the era of small computers began [12]. In 1976 Intel introduced the world's first microcontrollers, the 8748 and 8048, which combine a central processor unit (CPU), memory (ROM for instructions and RAM for data), peripherals, and input-output functions on a single piece of silicon [13]. A microcontroller may be described as a 'computer on a chip'. They are often embedded into a device and programmed to perform certain operations. Usually they perform a specific task for their lifetime, for example, the processor of a calculator. Having all the logic on a single piece of silicon reduces the cost and size of the board. Now microcontrollers are at the heart of a huge range of commercial and industrial equipment, including domestic appliances such as microwaves, DVD players and televisions. They are used in automobiles for engine-control and service functions, in medical instruments, and in many other areas.

The widespread availability of microcontrollers is a testament to their flexibility and low unit cost. Usually they have internal memory and a high level of input and output (I/O) device options including UART, I^2C, SPI, CAN, USB and other interfaces. The use of a microcontroller minimises the number of devices used in the system by integrating much of the external interfacing to switches, motors or other devices. Modern microcontrollers can even directly handle analog signals as many of them have built-in analog-to-digital (ADC) and digital-to-analog converters (DAC), comparators and pulse width modulators (PWM).

The internal structure of the microcontroller varies from one family to another. Sometimes it exploits an existing CPU core to make the development process easier and eliminate the need for extra tools. For some microcontrollers, the CPU core was specifically designed to achieve higher performance or easier I/O control. Many chip manufacturers offer both microprocessors and microcontrollers based on the same CPU core, to cover as large market as possible.

CPU performance mainly depends on the operating frequency (or more precisely, number of instructions per second) and the internal data bus width. Early microcontrollers had 4-bit or 8-bit internal data buses, while modern microcontrollers have 16-bit or 32-bit data buses. Usually, an external clock signal is used to synchronise each low-level operation within each instruction, as a result the instruction throughput could be ten times slower than the clock frequency. To compare the performance of two different microcontrollers one should use MIPS (millions of instructions per second) figures. As to the data bus, an 8-bit microcontroller might use a 16-bit bus to increase data throughput between CPU and memory; while a 16-bit microcontroller has an 8-bit data bus to save cost on external buffers and memory chips. Obviously, the wider the data bus, the more difficult it is to microprobe it and reverse engineer the CPU structure.

The CPU cores inside a microcontroller have either Harvard or von Neumann architecture [14]. The first refers to an architecture that uses physically separate storage for instructions and data, and the second uses a single storage structure to hold both instructions and data. From the security point of view, the Harvard architecture could offer better protection against microprobing attacks. In a von Neumann architecture, an attacker could modify the CPU so that it will no longer execute branch instructions or it will not fetch any instructions at all. Then by microprobing the CPU data bus and storing the signals, the contents of the whole memory can be recovered. The same trick applied to a Harvard microcontroller would reveal only the program code, whereas the data memory, which usually contains passwords and decryption keys, will not be available.

Another difference in CPU structure relates to the instruction set. Commonly it belongs to either a CISC (complex instruction set computer) or RISC (reduced/regular instruction set computer) architecture [15]. In CISC each single instruction can invoke several low level operations, such as a load from memory, an arithmetic operation, and a memory store. This is done to support high-level languages and make programs smaller, but at a cost in performance. RISC architecture employs a smaller and simpler set of instructions that all take about the same amount of time to execute. Simpler instructions make CPU design less complex and significantly increase its performance. Modern microcontrollers usually have an instruction set that lies somewhere in between CISC and RISC, balancing the advantages of each architecture. It is very hard to say which architecture gives better security protection. As the RISC CPU is simpler, it might be easier to reverse engineer it, but a CISC CPU leaves more characteristic power traces making identification of each instruction through power analysis much easier. An example of a CISC microcontroller with Harvard architecture is the Intel 8051, and one with von Neumann architecture is the Motorola 68HC05; RISC with Harvard – Microchip PIC, with von Neumann – Texas Instruments MSP430. More information about CPU structures in different microcontrollers can be found in the Appendix.

Some microcontrollers have a CPU with extra features to increase their performance. One is an instruction pipeline [16]. Each CPU instruction is divided into some simple sub-instructions. These smaller instructions are then dispatched to the hardware in step, with the pipeline controller watching the process. Hence, the code that the processor loads will not immediately execute, but as a result the processor executes two or more instructions at a time. Pipelining makes the power analysis more difficult, because during each CPU cycle two or more instructions contribute to the power trace.

Another feature modern microcontrollers might have is a cache memory that stores instructions and data that requires frequent and fast access [17]. For example, if the CPU executes a loop, then the instructions will be fetched from the cache memory rather than from the external memory thus saving time. Instruction and data caches could make microprobing attacks harder, as some information will not appear on the external data bus.

The above comparison can be applied to smartcards as well, because they have a structure very similar to microcontrollers.

2.1 Security evolution in silicon chips

Hardware security begins with embedded systems for industrial controllers. Thirty years ago such systems were built with separate components like CPU, ROM, RAM, I/O buffers, serial interfaces and other communication and control interfaces. Examples include control boards inside industrial controllers (Figure 1), printers, game consoles and home appliances.

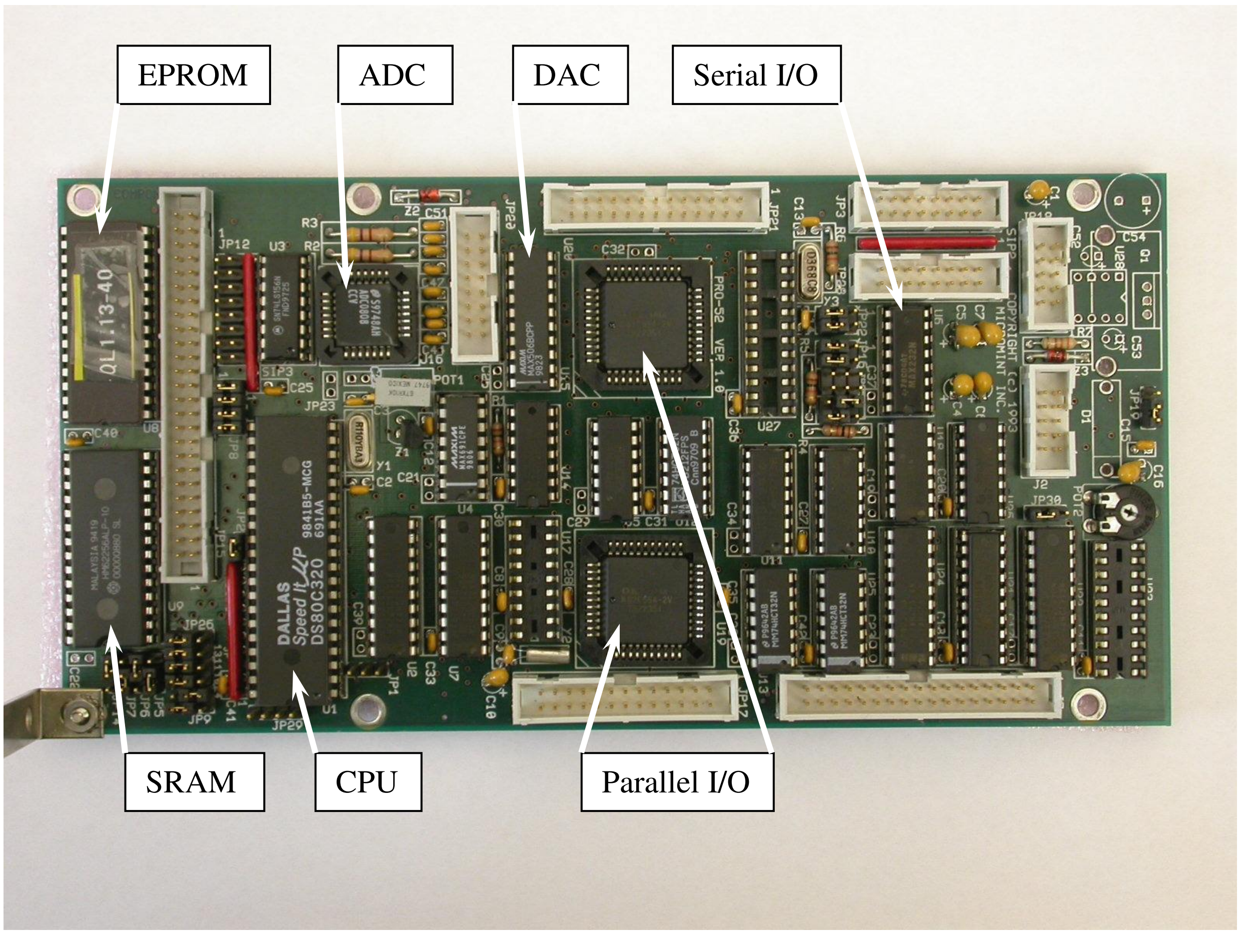


Figure 1. Universal embedded controller from Micromint Inc. [18]. Each component on the PCB is easy to identify and the board could be easily cloned

In the beginning there were almost no protection against cloning of such devices except law and economics. For example, ROMs were made with low-cost mask technology and cloning would involve either replacing them with EPROMs which are usually 3–10 times more expensive, or ordering Mask ROMs which would take time and require large capital investments. Another approach was used in game consoles where simple ASICs (Application-Specific Integrated Circuits) were widely used (Figure 2). Such ASICs were mainly carrying out I/O functions to replace tens of simple logic components, thus reducing the cost of the board and at the same time protecting against competitors who had to use larger and more expensive solutions. In fact these ASICs did not carry much security and their functionality could be understood in a few hours with a simple analysis of the signals using an oscilloscope or doing an exhaustive search over all possible combinations on their pins.

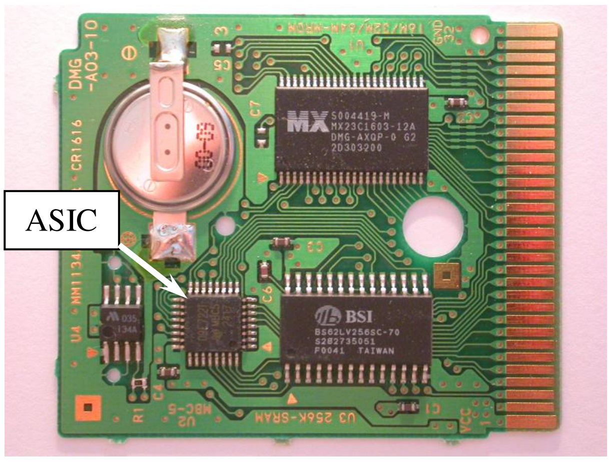


Figure 2. Game cartridge for Nintendo GameBoy game console [20]

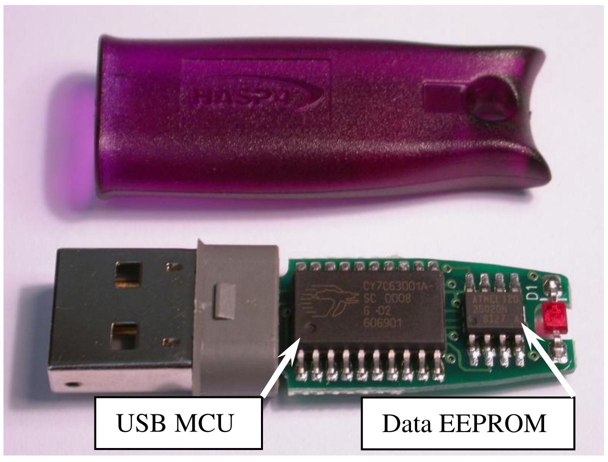


Figure 3. Aladdin HASP4 USB dongle [21]

From the late seventies, microcontrollers offered a very good replacement for CPU-based controller boards. They not only had internal memory and populated I/O interfaces, but some sort of security protection against unauthorised access to the internal memory contents. Unfortunately, early microcontrollers did not offer non-volatile storage facility and important data had to be stored in a separate chip outside the microcontroller (Figure 3) thus allowing the attacker to easily access them. Some low cost attacks on USB dongles used for software protection were published recently [19].

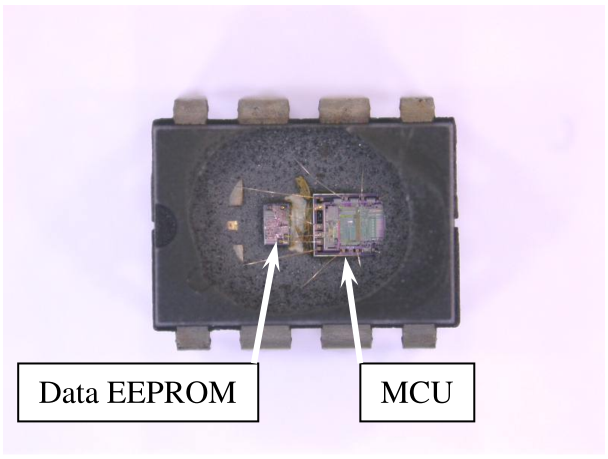


Figure 4. Non-volatile data memory and micro-controller chip are encapsulated into the same package in Microchip PIC12CE518 microcontroller [22]

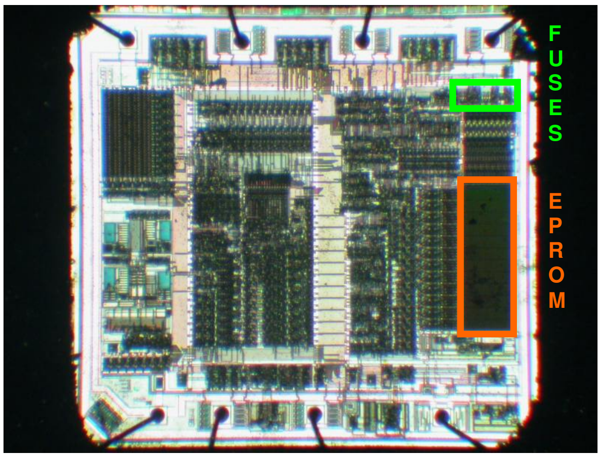


Figure 5. Security fuses are located outside the program memory array in Microchip PIC12C508 microcontroller [22] and can be easily disabled with UV light

The next step in security evolution was to place the EEPROM data storage chip next to the microcontroller inside the same plastic package (Figure 4). To attack such a chip is not easy; a professional would decapsulate the sample and either microprobe the data chip or bond it into a separate test package. Both methods require equipment which cannot be afforded by a low-budget attacker. Such an attacker could try to use homemade microprobers (bonding pads on old chips are relatively large) or exploit a software bug to get access to the data.

Some microcontrollers do not have any special hardware security protection at all. Their protection is based on obscurity of the proprietary programming algorithm. It might be the case that the read-back function was deliberately disguised, or replaced with a verify-only function. Usually such microcontrollers do not offer very good protection and some examples are presented in Chapter 4. In fact, the verify-only approach could be very powerful if implemented properly, as it is in some smartcards.

The next step in increasing the security protection was in adding a hardware security fuse that disables the access to data. The easiest implementation, which does not require the complete redesign of the microcontroller structure, was for the fuse to control the read-back function of the programming interface (Figure 5). The drawback of this approach was in making it easier to locate the security fuse and perform an invasive attack. For example, the state of the fuse could be changed by connecting the output from the fuse cell directly to the power supply or ground line. In some cases it might be enough to just disconnect the sense circuit from the fuse cell by cutting the wire from it with a laser cutter or focused ion beam (FIB) machine. It might be possible to succeed in non-invasive attack as well, because a separate fuse would certainly behave differently from the normal memory array. As a result it might be possible to find such a combination of external signals under which the state of this fuse would not be read correctly thus allowing the access to the information stored in the on-chip memory. Some examples of these attacks are given in Chapter 4. Semi-invasive attacks could bring the attacker to success even faster but will require decapsulation of the chip to get access to the die. A well known example of such attacks is erasing the security fuse under a UV light; these attacks are discussed in Chapter 6.

The next step was to make the security fuse part of the memory access circuit, so that any external access to the data is disabled if the fuse is set (Figure 6). Usually the fuse is located very close to the main memory or even shares some control lines with it. Also it is fabricated with the same technology as the main memory array making it harder to locate and reset. Non-invasive attacks could still exist but would require much time and effort to find. At the same time, semi-invasive attacks might still work. Certainly it would take more time for an attacker to find the security fuse or the part of the control circuit responsible for the security monitoring, but this could be easily automated. Performing invasive attacks could be more difficult as most of the work would need to be done manually, so it will certainly increase the cost and time of the attack.

A further improvement involved using a part of the main memory to control access to the data from outside. This was implemented either by latching the information stored at a certain address at power-up and treating it as a security fuse, or by using passwords to grant access to the memory. For example, in the Texas Instruments MSP430F112 microcontroller, the read-back operation can be called only after the correct 32-bytes password is entered [25]. Without that, only the chip erase operation is available. Although such protection seems to be more effective than previous offerings, it has some drawbacks which could be exploited in low-cost non-invasive attacks such as timing attacks and power analysis. More details on these attacks are presented in Chapter 4. If the state of the security fuse is sampled from the memory during power-up or reset, it could present some room for the attacker to play with power glitches, trying to force the circuit to get the wrong state of the memory.

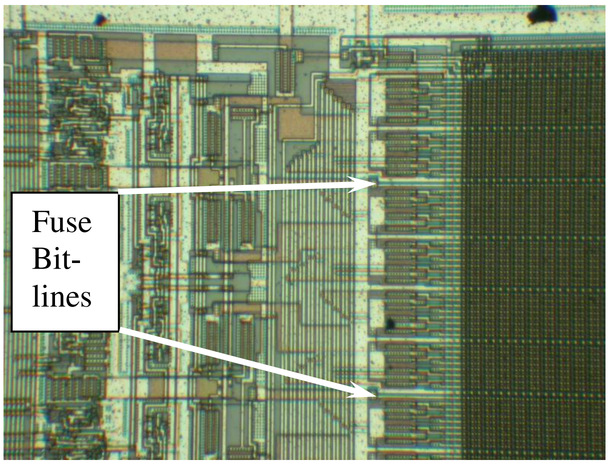


Figure 6. Security fuse is a part of the memory access control logic in Motorola MC68HC705C9A microcontroller [23]. 200× magnification

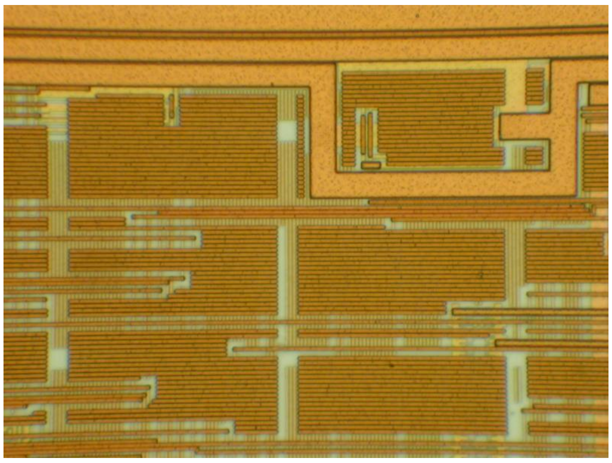


Figure 7. Fake top metal layer pattern on Microchip PIC16F648A microcontroller [24] makes analysis of the chip die and microprobing attacks more difficult. 200× magnification

Other improvements in making invasive attacks more expensive involve using a top metal sensor mesh [8]. All paths in this mesh are continuously monitored for interruptions and short circuits, and cause reset or zeroing of the EEPROM memory if alarmed. Normally such protection is not used in ordinary microcontrollers because, firstly, it increases the design cost and, secondly, it can be triggered unintentionally in abnormal working conditions such as high electromagnetic noise, low or high temperatures, irregular clock signal or power supply interruptions. Instead, ordinary microcontrollers adopt the less expensive approach of placing a fake top layer mesh (Figure 7), but this still remains a very effective annoyance for optical analysis and microprobing attacks. In smartcards such meshes are implemented properly with the sensor wires going between the power supply and ground wires (Figure 8). Some design flaws were found in such implementations making microprobing attacks possible. Also such meshes do not protect against non-invasive attacks, and some semi-invasive attacks are still possible because the mesh has gaps between the wires and light can pass through it down to the active areas of the circuit.

In user programmable smartcards manufacturers went even further and totally removed the standard programming interface. Instead of it, they used a bootstrap loader which either erased or deactivate itself after the user code was uploaded. Thus the card could be programmed only once during an initialisation step and beyond this it becomes the responsibility of the user's embedded software to support any access to the program and data stored inside the card.

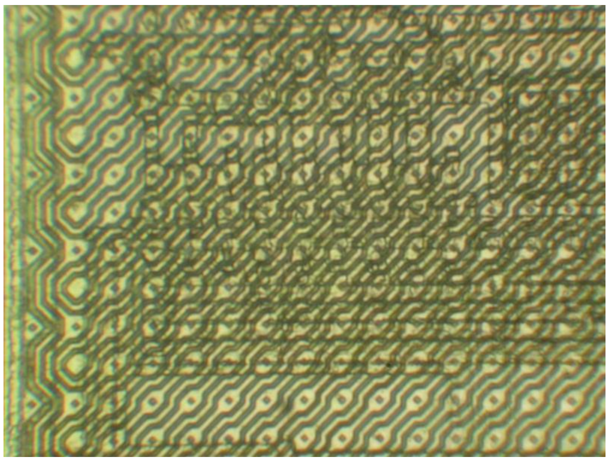


Figure 8. Top metal layer sensor mesh on ST Microelectronics ST16 family smartcard chip [26]. 500× magnification

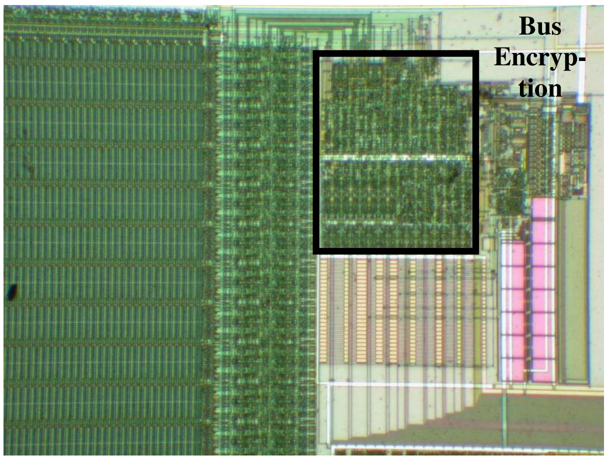


Figure 9. Hardware bus encryption module in Infineon SLE66 family smartcard chip [27] preserves EEPROM data from being microprobed. 100× magnification

In some recent smartcards further protection against microprobing attacks is used such as EEPROM data memory bus encryption (Figure 9). Even if the attacker manages to pick up the signals from the data bus he will not be able to recover the passwords, the secret keys or other sensitive information from it. This protection was aimed at preventing invasive and semi-invasive attacks. At the same time non-invasive attacks could still be successful as the CPU normally has full access control to unencrypted information. The only microcontroller that employs a similar approach to the external program memory is the Dallas Semiconductor DS5002FP [28]. In fact some vulnerabilities were found in the implementation of the data encryption that lead to a relatively low cost attack published several years ago [29].

Another improvement worth mentioning is moving from the standard building-block structures like CPU instruction decoder, register file, ALU and I/O circuits, to a complete ASIC-like logic design. This design approach is called 'glue logic' and it is widely used in smartcards [30]. Glue logic makes it virtually impossible to tap into the card's information by manually finding signals or nodes to attack physically. The glue logic design could be done automatically with using such tools as MILES^{TM} (Managed IC Lifetime Extension System) from InnovASIC Inc. [31]. This technique is widely used for cloning, and improving the security and performance of the popular CPU cores without licensing them [32]. For example, the Ubicom (former Scenix) SX28 microcontroller (Figure 10) is pin-to-pin and program compatible with the Microchip PIC16C57 microcontroller but employs glue logic design, Flash program memory, larger RAM and it has ten times higher performance [33]. In the PIC microcontroller (see Figure 5), an attacker can easily trace the data bus coming from the memory to the CPU while on the SX microcontroller the data lines never form any sort of a physical bus, so tracing them is a very challenging task. Reverse-engineering and microprobing attacks on such microcontrollers are very difficult and time-consuming. They require a well-equipped laboratory and experienced engineers.

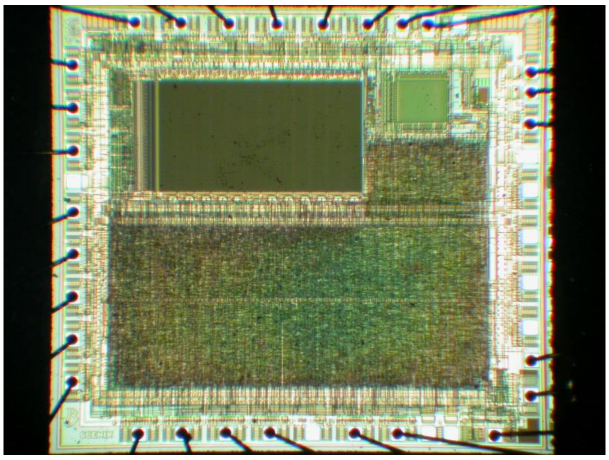


Figure 10. Ubicom SX28 microcontroller exploits ASIC-like glue logic design with improved security and performance

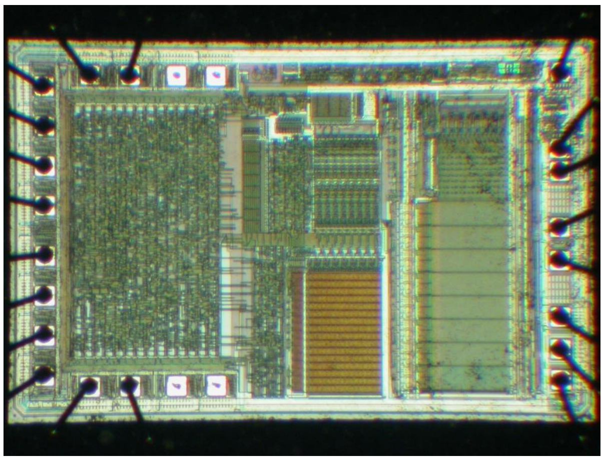


Figure 11. Cypress CY7C63001A microcontroller has partial glue logic design but all internal bus lines are easily accessible

More often an intermediate approach is used when the chip is built from separate blocks but each block uses glue logic design as in, for example, the Cypress CY7C63001A microcontroller (Figure 11) [34]. In this case an attacker could more easily trace the bus and control lines between the blocks, and launch invasive or semi-invasive attacks on the chip. Glue logic design does not eliminate the possibility of non-invasive attacks, but as the performance increases, faster and more expensive equipment is required. Semi-invasive attacks will also face problems due to disguised design blocks. Of course the attacker could automate the process by running an exhaustive search and trying to attack all possible areas. Definitely this approach would take a long time and may in the end not be successful. On the other hand, an attack could be applied directly to the memory itself or its control circuit, because they cannot be implemented in the same glue logic structure and stay visibly separate.

Technological progress on its own is increasing the costs to the attackers. Ten years ago it was possible to use a laser cutter and a simple probing station to get access to any point on the chip surface, but for modern deep submicron semiconductor chips very sophisticated and expensive technologies must be used. That excludes most potential attackers. For example, the structure of the Microchip PIC16F877 microcontroller can be easily observed and reverse engineered under a microscope (Figure 12). The second metal layer and polysilicon layer can still be seen even if buried under the top metal layer. This is possible because each subsequent layer in the fabrication process follows the shape of the previous layer. Under a microscope the observer sees not only the highest layer but also edges that reveal the structure of the deeper layers. In 0.5 μm and smaller technologies, for example in the Microchip PIC16F877A microcontroller, each predecessor layer is planarised using chemical-mechanical planarisation (CMP) process before applying the next layer [35]. As a result the top metal layer does not show the impact of the deeper layers (Figure 13). The only way to reveal the structure of the deeper layers is by removing the top metal layers either mechanically or chemically. Some examples of this process are presented in Chapter 5.

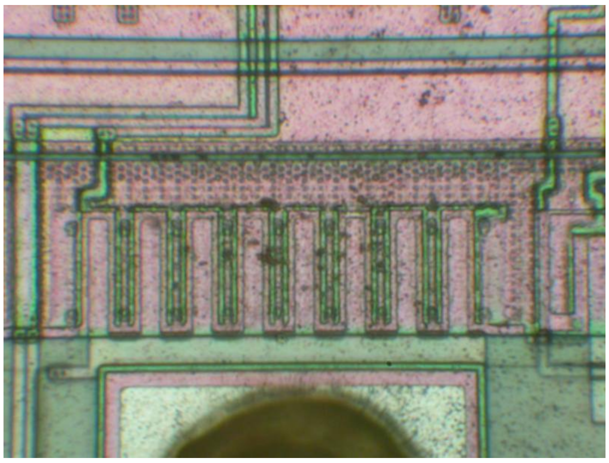


Figure 12. Second metal layer and polysilicon layer can be seen through top metal layer on Microchip PIC16F877 microcontroller [36]. 500× magnification

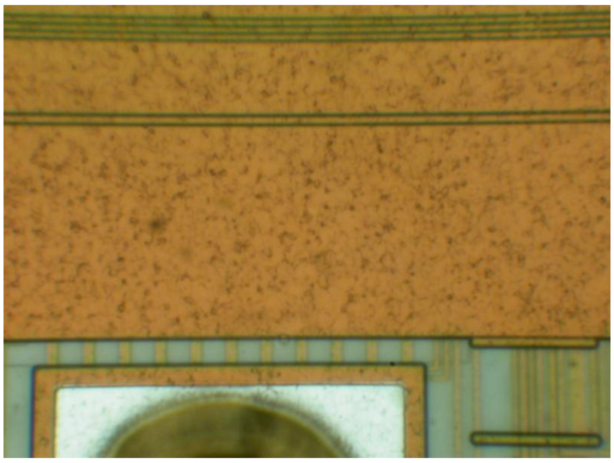


Figure 13. Nothing can be seen under the top metal layer on Microchip PIC16F877A microcontroller [37] as the layers inside the chip were planarised during fabrication. 500× magnification

As can be seen from all the shown examples, hardware security in microcontrollers and smartcards is being constantly improved. Because the tools for reverse engineering are becoming more sophisticated, better and better security protection is required. Rapid co-evolution is driven by this continuous battle between chip manufacturers and attackers.

Another threat that must be considered is that a great deal of second-hand semiconductor manufacturing and testing equipment appears on the market. It cannot be used to attack high-end products, but should be enough to attack chips manufactured with older technology. For example, while 90 nm manufacturing technology is currently leading-edge, most microcontrollers are produced with 0.35 µm technology and smartcards with 0.25 µm technology.

2.1.1 Memory types

A microcontroller operates according to a program located in its memory. There are many different memory types and most of them are used inside microcontrollers. The majority of modern microcontrollers are made with CMOS technology. Therefore their on-chip memories are either CMOS (SRAM and some EEPROM) or MOS. The latter is very likely to be NMOS as n-channel transistors have better parameters and smaller size.

Early microcontrollers used Mask ROM and UV EPROM for program storage and SRAM for data storage. Mask ROM is still used in modern microcontrollers when large-quantity production and low cost are required. Normally such microcontrollers are not marked with their part number on the package and have only manufacturer logo and internal factory ROM revision number (Figure 14). Mask ROM offers very good performance, but cannot be reprogrammed or updated. Microcontrollers with UV EPROM are usually offered in two versions – one for prototyping, in ceramic packages with a quartz window (Figure 15), and another, called OTP (One-Time Programmable), for mass production, in standard plastic packages (Figure 16). UV EPROM has some disadvantages for developers: it requires high voltages for programming; data can be written only one byte or word at a time, so it takes a long time to program the whole chip; even erasable versions cannot be reprogrammed more than a hundred times; and the erase operation takes 20–30 minutes under a very intensive UV light source.

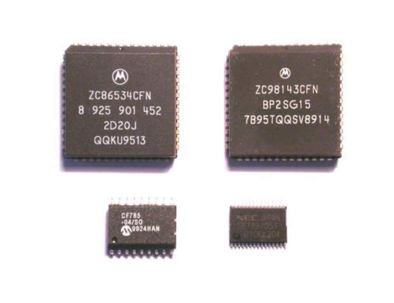


Figure 14. Microcontrollers with Mask ROM. They normally have only internal factory marking

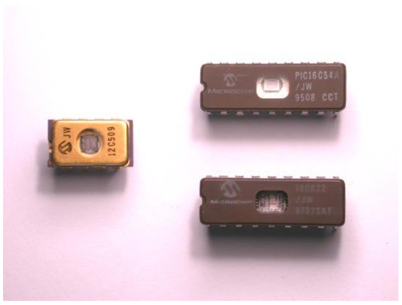


Figure 15. Microcontrollers with UV EPROM and quartz window

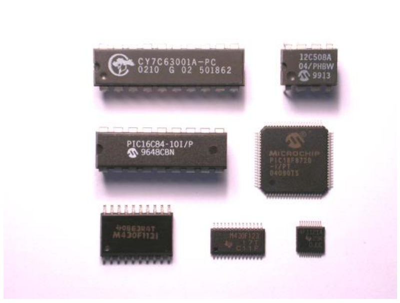


Figure 16. Microcontrollers with OTP EPROM, EEPROM and Flash EEPROM

SRAM is also used in some microcontrollers as a program storage memory when fast access time or frequent memory update is required. One example is the Cypress USB 2.0 microcontroller CY7C68013 [38]. SRAM is also used in Dallas Semiconductor secure microcontrollers, where fast memory erasure is required in tamper-proof applications.

The more modern EEPROM (Electrically Erasable PROM) memory had some advantages over the UV EPROM: it can be reprogrammed electronically, though only a limited number of times – from thousands to hundreds of thousands cycles; high voltages are usually generated by on-chip voltage charge-pump circuits; and programming is much faster. A further improvement of the EEPROM memory, called Flash EEPROM, is becoming the main memory storage in modern microcontrollers and smartcards. It offers much faster programming time and it can be reprogrammed in blocks saving a lot of time. It can be reprogrammed hundreds of thousands of times, and most of the modern microcontrollers with Flash memory offer internal memory programming, thus allowing field code upgrades without expensive programming tools. Flash memory also has high density offering 3–5 times more storage capacity than the same area of EEPROM. The downside of this memory type is that it can only be erased in blocks, which are relatively large. That puts some strain on embedded software design where memory updates are required. Some microcontrollers offer an alternative solution to this problem, having both Flash and EEPROM memories on the same die, so that the Flash is used for program storage and infrequent updates, and the EEPROM is for data that requires frequent or bitwise updates.

One approach to combine SRAM and EEPROM in one structure, thus achieving fast read/write access time and non-volatilaty, was in NVRAM memory, which has a basic SRAM structure, plus an EEPROM cell and control logic to store the memory state either by external signal or when the power supply drops below a certain level [39]. Due to the complexity of this memory and high production cost, it was not used in microcontrollers and it was only used in a very few smartcard and ASIC products.

The recently introduced Ferroelectric memory, or FRAM, was promoted as a very effective replacement for EEPROM and Flash memories [40]. It has very fast write time – in the same order as SRAM – and it does not require internal high voltage generators. Unfortunately this memory has a limited number of read/write cycles, and cannot be used as a direct SRAM replacement. Also the FRAM memory cell size is 3–5 times larger than a Flash cell and its fabrication technology is more complex. There are very few areas where FRAM-based controllers are used, mainly in contact-less smartcards where the low power consumption and instant write operation of the FRAM is a big advantage [41].

Another modern memory type, which is considered to be a very effective replacement for Flash memory, is Magnetoresistive memory or MRAM [42]. At the moment there are only engineering samples of this memory technology but it is proposed to be used in future smartcard products. This memory has almost unlimited endurance and very fast access and write times, it does not require high voltages for operation and it has a high density design [43].

	Static RAM	Mask ROM	OTP EPROM	UV EPROM	EEPROM	Flash EEPROM	NVRAM	FRAM	MRAM
Read time	FAST ~10 ns	FAST ~5 ns	MED ~50 ns	MED ~50 ns	MED ~50 ns	FAST ~20 ns	FAST ~50 ns	SLOW ~150 ns	FAST ~10 ns
Write time	FAST ~10 ns	N/A	SLOW ~10 ms	SLOW ~10 ms	SLOW ~1 ms	MED ~10 µs	FAST ~50 ns	FAST ~150 ns	FAST ~10 ns
Data retention	>5 years (battery)	N/A	>10 years	>10 years	>40 years	>100 years	>40 years	>10 years	>10 years
Cell size	6T	1T	1T	1T	2T	1T	10T	2T/2C	1T
Low voltage	YES	YES	NO	NO	NO	NO	NO	YES	YES
Endurance, rewrites	N/A	N/A	1	100	10^3–10^6	10^4–10^6	N/A	10^9–10^{12}	N/A
Cost	HIGH	LOW	MED	HIGH	MED	LOW	HIGH	MED	LOW

Table 1. Characteristics of different memory types used in microcontrollers. As can be seen, FRAM and MRAM offer very effective replacement to SRAM, EPROM, EEPROM and Flash memories.

Table 1 summarises the characteristics for different types of memories. As can be seen, each memory type has its own advantages and disadvantages and the hardware designer should use appropriate memory according to his requirements. Normally microcontrollers have different memories on the same die, so that developers use the appropriate memory technology. For example, ‘SRAM and EPROM’ in OTP PIC microcontrollers, ‘SRAM and Mask ROM and

EEPROM' in Motorola MC68HC05 microcontrollers, or 'SRAM and EEPROM and Flash' in Microchip Flash PIC microcontrollers.

In CPLDs, EPROM, EEPROM and Flash memories are mainly used. FPGAs are mostly SRAM based; very few manufacturers offer non-volatile FPGAs with Antifuse and Flash memories [44][45]. The Antifuse memory [46] is a different kind of OTP memory that uses programmable interconnection links between metal wires inside the chip. As these links are extremely small (~100 nm wide) it is virtually impossible to identify their state and that gives an extremely high security level to the devices based on antifuse technology.

From the security point of view, the same microcontroller with Mask ROM memory offers better security protection than with EPROM memory, which in its turn has better protection than EEPROM or Flash memory. Normally the Mask ROM version of microcontroller does not provide any form of external access to the information stored inside the chip; the only way to get access to the code is by either reading the memory optically or by microprobing the data bus. Both methods require special equipment and highly skilled attackers. However some manufacturers intentionally leave backdoor access to the code for testing purposes after fabrication. Normally the information on these test protocols is kept secret by the manufacturers, but if an attacker finds out how to use this interface, he will get access to the code.

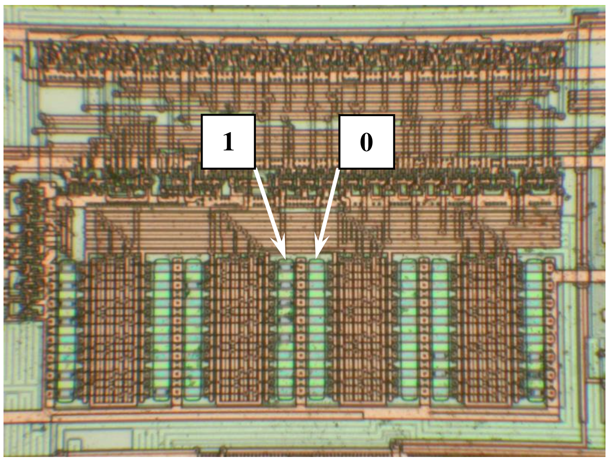


Figure 17. Laser ROM in Dallas DS1961S iButton chip [49]. Information can be read optically and altered with a laser cutter

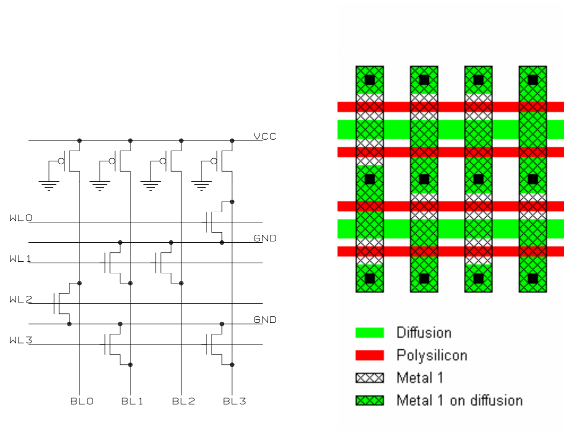


Figure 18. Configuration and layout of MOS NOR ROM with active layer programming. This type of memory can be read optically

Mask ROM usually has NOR or NAND structure according to the way transistors are connected inside the memory array [47][48]. There is an OR structure as well but the only difference between it and the NOR structure is that the transistors are connected to V_{CC} instead of V_{SS}. For each structure the information is encoded in different ways. The information is placed into the ROM during chip fabrication and cannot be changed later. There is another type of ROM memory which is programmed after fabrication but still at a factory – laser ROM (Figure 17). It is used in Dallas Semiconductor iButton products for serialisation and the information is programmed by cutting memory bits with a laser cutter. This memory leaves some freedom to the attacker who could alter the memory contents on a secure product.

In NOR ROM with active layer programming, the logic state is encoded by the presence or absence of a transistor (Figure 18). Information from this type of memory is easily extractable under optical microscope. For technologies smaller than 0.8 μm deprocessing might be required to remove the top metal layers which obstruct observation.

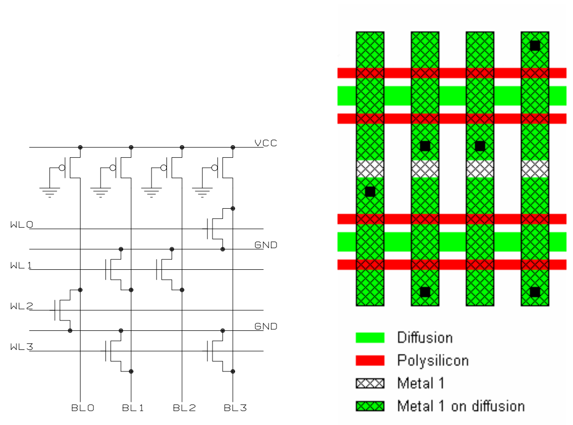


Figure 19. Configuration and layout of MOS NOR ROM with contact layer programming. This type of memory can be read optically but usually requires deprocessing

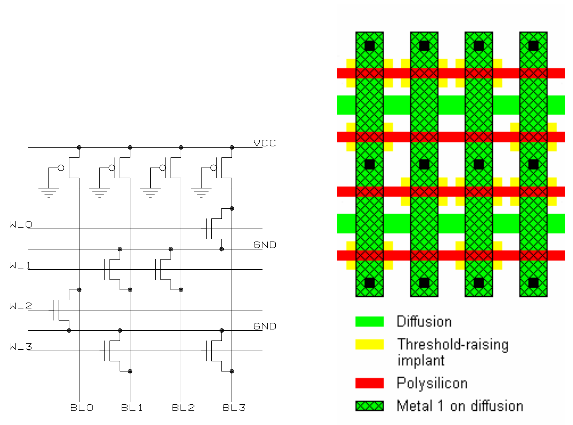


Figure 20. Configuration and layout of MOS NOR ROM with programming using implants. This type of memory offers high level of security protection against optical reading

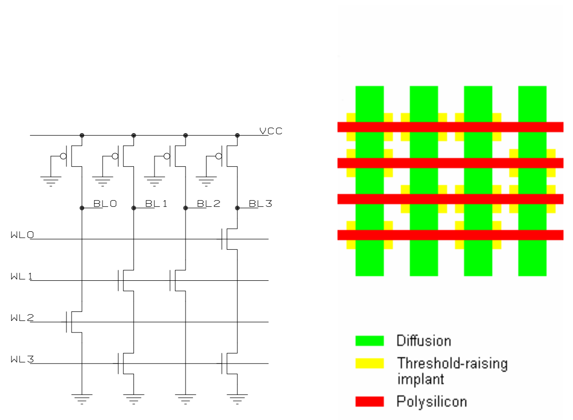


Figure 21. Configuration and layout of MOS NAND ROM with programming using implants. This type of memory offers high level of security protection against optical reading

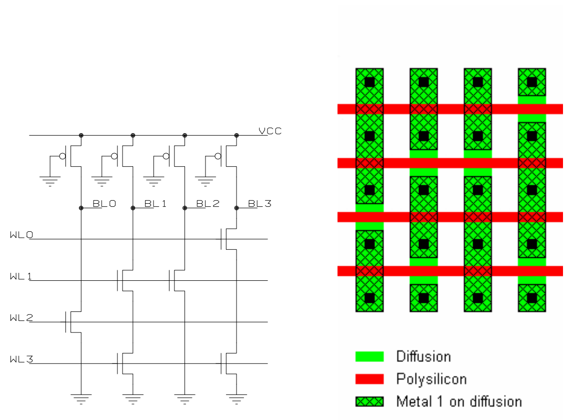


Figure 22. Configuration and layout of MOS NAND ROM with metal layer programming. This type of memory can be read optically

In NOR ROM with contact-layer programming, the information is encoded by the presence or absence of a via plug from bit-line to the active area of a transistor (Figure 19). In old memory technologies, these plugs are visible under a microscope, but in modern memory technologies with CMP planarised layers, deprocessing is required to expose the plugs. This could cause trouble to an inexperienced attacker, so this type of memory could be considered more secure against invasive and semi-invasive attacks.

In NOR ROM with programming using implants, the data is encoded by the threshold level of a transistor (Figure 20). This is achieved by creating transistors with different doping levels during fabrication. This type of memory, also called VTROM (Voltage Threshold ROM), provides a high level of protection against various kinds of attacks because the state of each transistor cannot be observed optically even after deprocessing procedure. This type of memory is very often used in smartcards to prevent code extraction from the memory. Later smartcards use NAND VTROM which offers the same level of security but has more compact design (Figure 21). At the same time there are some relatively low cost methods that make it possible to extract the contents from these memories [8] and some other methods are presented in Chapter 6. To prevent these attacks, modern smartcards also use memory encryption.

In NAND ROM with metal layer programming, the information is encoded by short-circuiting the transistors (Figure 22). This type of memory has a very low level of protection against optical observation, as these metal fuses are clearly visible under a microscope.

Examples of most of the above mentioned memory types in real chips and the associated memory extraction techniques are given in Chapters 5 and 6.

Non-volatile memories, such as EPROM, EEPROM and Flash, use floating-gate transistors to store the information [47]. In OTP and UV EPROM the memory cell consists of a single transistor (Figure 23) and electrons are injected into the floating gate during programming. As a result, the threshold level of the transistor changes, and this is detected by sense amplifiers when the memory is read. The only way to erase the memory and remove the charge from the floating gate is by exposing it to the UV light with a wavelength shorter than 300 nm (normally this memory is erased under a mercury lamp and the wavelength is specified as 253.7 nm). The total dose necessary to properly erase the memory is about 15 W/cm^2 which normally corresponds to 15–20 minutes exposure under a mercury lamp. It makes this type of memory unattractive in applications where frequent code update is required. But it is still used in OTP microcontrollers where erasure is not required.

EEPROM memory was introduced by Intel in 1980 and it offered a great advantage over the EPROM by allowing full electrical control over both write and erase operations (Figure 24). Due to high manufacturing cost and complexity, it was not widely used in microcontrollers until the early nineties. Today most microcontrollers have either EEPROM or its successor, Flash EEPROM memory, on chip. Flash memory has simpler structure (Figure 25), faster write and access time but unfortunately it cannot be reprogrammed in single bytes as it can be erased only in blocks, which is not convenient for small data updates. Flash EEPROM has many different layouts and each semiconductor manufacturer normally has its own memory design. According to the way the transistors are connected inside the array this memory could have either NOR or NAND (Figure 26) structure.

From the security point of view all floating-gate memories offer very good protection against invasive attacks, because the charge injected during programming is very small, and buried deeply inside the memory cell, so it cannot be detected directly. Deprocessing does not reveal any information – only cell structure. The only practical invasive way of extracting the information is by microprobing the internal memory bus. This could be extremely difficult for modern submicron Flash memories which have multiple top metal layers over the data wires.

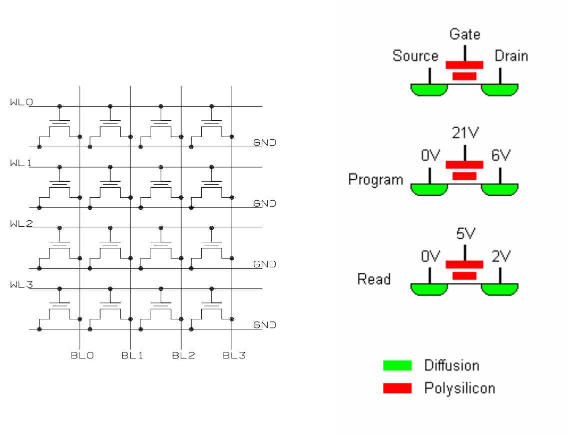


Figure 23. Configuration, cross-section and modes of operation of UV EPROM

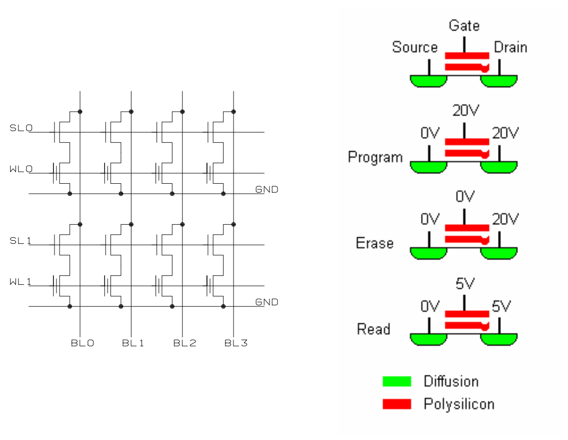


Figure 24. Configuration, cross-section and modes of operation of FLOTOX EEPROM

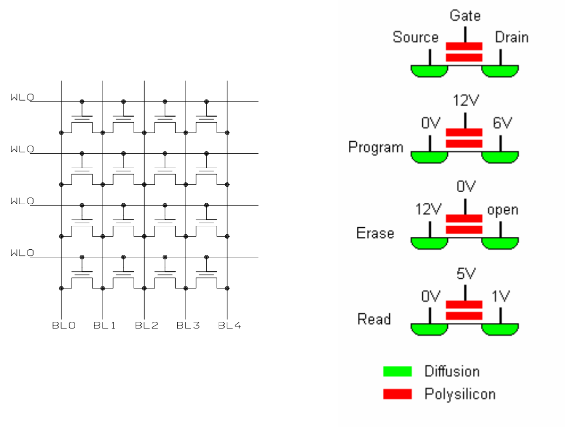


Figure 25. Configuration, cross-section and modes of operation of ETOX Flash EEPROM

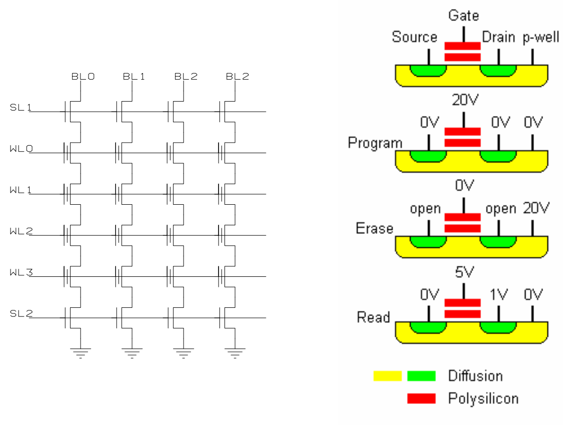


Figure 26. Configuration, cross-section and modes of operation of NAND Flash EEPROM

At the same time, such semi-invasive attacks as selective UV erasure can be easily applied and widely used to breach the security protection in OTP microcontrollers. As EEPROM and Flash memories use very similar floating-gate transistors, it might be possible to erase the security fuses with UV light as well. The problem is that UV attacks can be performed on a relatively small number of microcontrollers, as manufacturers implement different protections against UV attacks. For example, they cover security fuses with a top metal layer that stops the UV light. Another trick is to invert the state of the fuse to make its programmed state indicate ‘disabled protection’ and erased state – ‘enabled protection’. In this case UV light will have no effect on the activated security fuse. Other ways of protecting against UV light attack are discussed in Chapter 7.

In terms of non-invasive attacks, EPROM memory has some advantages over EEPROM and Flash memories as it is more robust against power glitch attacks. This happens because it has a simpler structure, larger cell size, thicker gate oxide and no on-chip high-voltage charge pumps. The sense amplifiers used to distinguish between ‘0’ and ‘1’ logic states are much simpler in EPROM and less sensitive to the power supply voltage. Against semi-invasive attacks, EPROM memory is also better than EEPROM and Flash. For example, the fault injection attacks that will be discussed in Chapter 6 can be used to modify the contents of the cell but for EPROM much higher power is required. That makes OTP microcontrollers more attractive in the applications where high security is required. Unfortunately modern microcontrollers do not use this type of memory any more as it cannot be reprogrammed, has lower density than the Flash memory and is thus more expensive. That forces semiconductor manufacturers to introduce additional protection against unauthorised access to the memory contents. For example, modern smartcards do not have hardware control for access to the on-chip Flash and EERPOM memories, but only a bootstrap loader located in the Flash memory that overwrites itself during first initialisation, eliminating any possible access to the information (unless implemented by the customer). Hardware access to the memory has multi-level security protection ensuring that access will not be granted unless all the requirements are met. In some microcontrollers, very sophisticated access password protection is implemented.

Another big problem for EPROM, EEPROM and Flash memories that affects the hardware security of the semiconductor devices is data remanence [50]. Many microcontrollers with these types of memory have a security fuse which, once activated, cannot be reset until the whole memory content is first erased. Manufacturers put a lot of effort into hardware design to ensure that the security fuse will not be deactivated by manipulation of external signals such as power glitches. They made very good progress, and very few of the modern microcontrollers can be broken using tricks such as applying power glitches during the chip erase operation to terminate the memory erase without affecting the erase of the security fuse, or exposing the chip to UV light for long enough to erase the security fuse but not long enough to destroy the memory contents. Some examples on these attacks are given in Chapters 4 and 6, but recent revisions of microcontrollers are not sensitive to such attacks.

In modern chips, an additional voltage monitoring circuit is usually implemented, causing a reset of the hardware programming interface or preventing any write/erase operations below or above certain voltages. What was wrongly assumed is that information must disappear from the memory after it was erased. In fact some traces of the data are still left after the erase operation, and to get the information back we just have to find the right method to measure the residual charge on a floating gate, or a threshold of a memory transistor. This is not an easy task, but if the security fuse was deactivated during the chip erase operation, the memory can be accessed normally. That allows the attacker to measure the response from each transistor inside the array by sequential reading of each memory location and microprobing the internal memory bus. Of course it is not a trivial task, but a determined and experienced attacker can do this. In some microcontrollers the threshold level of each transistor can be measured in fully non-invasive way by playing with the interface and power supply voltages. This is possible because very often the memory sense circuit uses the power supply voltage as a reference. Some examples on how it can be exploited are given in Chapter 4.

Another memory type used in all microcontrollers (mainly as a register file memory and operational memory) is SRAM [51]. It is also used in secure microcontrollers such as the Dallas DS5002FP and JAVA iButtons [52] where the information should disappear quickly if a tampering attempt is sensed. An SRAM memory cell consists of six transistors (Figure 27), four of which create a flip-flop while the other two are used for accessing the cell inside the array.

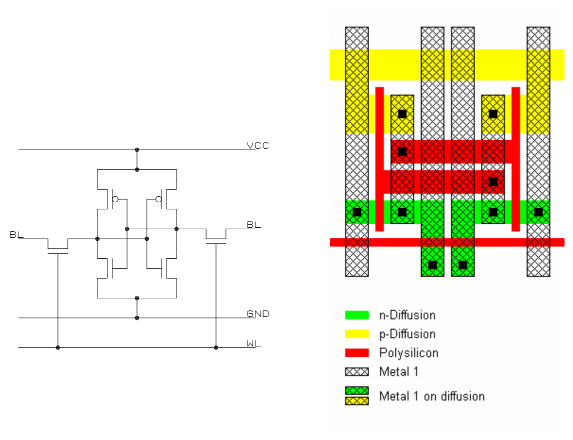


Figure 27. Configuration and layout of SRAM cell

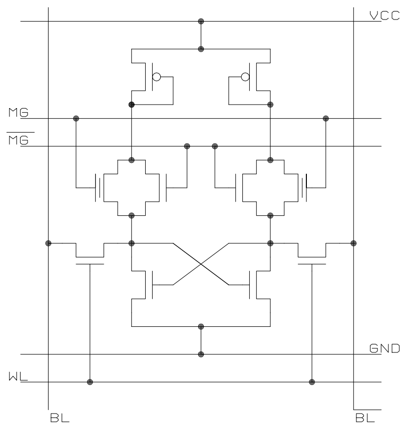


Figure 28. Configuration of NVRAM cell

SRAM memory offers very good security protection, as the information from it can be easily erased by disconnecting the power supply if the alarm is triggered. Performing invasive or semi-invasive attacks is very problematic because any attempt to access the chip surface would very likely destroy the data. For example, decapsulation requires very strong acids to be used which are conductive and cannot be used on a powered up chip. Even if the attacker manages to access the die, the state of its transistors cannot be observed optically. Microprobing is difficult because the internal wires are buried under top metal bit-lines, ground and power supply wires. The only practical way to access the memory is from the rear side of the chip die, but this requires more expensive equipment and a highly skilled attacker. Meantime, there are some semi-invasive techniques that allow observation of the memory state, but require special laser scanning microscopes. More detailed information on such techniques is given in Chapter 6. At the same time non-invasive attacks can be used to exploit any problems that might exist in the memory interface, as happened with the Dallas Semiconductor secure microcontroller [29]. Data remanence could cause some problems to SRAM security as well [53]. At temperatures below 0°C some samples of the SRAM chips retain information for hours [54]. But, in general, SRAM memory offers a very good level of protection and low-temperature attacks can be avoided by placing temperature sensors into the secure module enclosure as in the IBM 4758 cryptoprocessor [7].

The main disadvantage of SRAM storage memory is the requirement for constant battery backup which makes secure systems larger and requires regular servicing.

NVRAM memory combines in one cell the ordinary SRAM cell structure and EEPROM cell (Figure 28) [47]. It is not widely used, and might have lower security compared with SRAM and EEPROM, because the attacker can exploit the weak points of both memory types. In the absence of the power supply, the information is stored in the EEPROM part of the cell and after power-up it is present in both parts of the cell.

FRAM memory uses a two-transistor cell with non-linear capacitors which change their polarisation according to the applied electric field and keep it without the power supply (Figure 29) [55]. The big disadvantage of this memory type is that the read operation is destructive to the contents of the cell so that overwrite refresh is required. At the same time this type of memory offers very good security protection against invasive and non-invasive attacks because its state cannot be observed optically or detected with probes. Microprobing of the memory data bus is of course still possible, unless the information is encrypted.

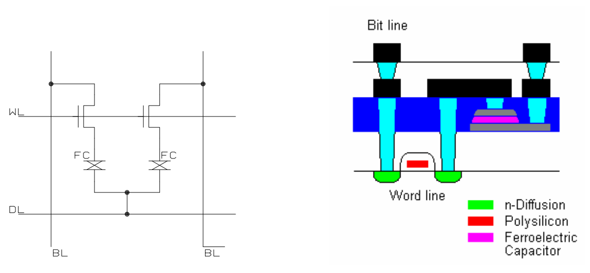


Figure 29. Configuration and cross-section of FRAM cell

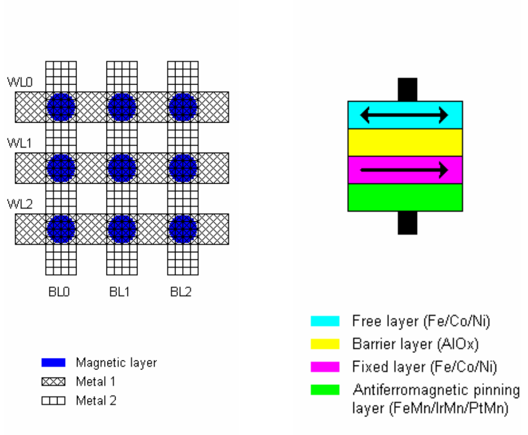


Figure 30. Layout of MRAM and cross-section of its cell

MRAM memory uses magnetic material to store the information and magneto-resistive sensors to read it out (Figure 30) [56]. Due to the very small size of the cell, and its deep location under top metal layers, it would be extremely difficult to probe it directly. Meantime, there exist high resolution magnetic sensors that can be used to scan such memories [57], provided the top metal layers are removed. Microprobing of the memory data bus might be considered as well but could be very difficult due to the small fabrication process of this memory (normally 0.18–0.25 μm). Together with other characteristics of MRAM memory such as fast access and write time, low power consumption and high density, this memory could be a very good alternative to the Flash memory.

2.1.2 *Types of security protection*

The programming interface allows writing, verifying, reading and erasing of data in on-chip memory. It could be implemented either in hardware (JTAG state machine or proprietary interface) or in software (Mask ROM or Flash bootloader). In the hardware interface, security protection is normally associated with security fuses that control operation of the interface, for example, by blocking the data from being sent from the memory to the output buffer. As for the software interface, password protection is normally used, but there could be a hardware security fuse as well whose state is checked in software. Some microcontrollers benefit from both interfaces offering a software bootloader control for in-system programming and a fast hardware interface for mass production programming. Each implementation has its pros and cons. The software approach gives more flexibility and better control over the programming process, but could leak some security information in the form of time delays and power consumption. A hardware implementation is faster, less sensitive to glitch attacks, and does not leak much information through power consumption. In terms of the silicon resources, both variants take similar space, and in modern microcontrollers it is negligible compared to the other large parts such as program memory, CPU and analog interfaces. That allows manufacturers to place two or more different programming interfaces on the same die, such as a hardware in-circuit serial programming via synchronous interface (e.g. SPI, JTAG), fast industrial parallel programming, and a software bootloader via asynchronous serial interface (e.g. RS-232).

Some manufacturers intentionally do not provide any programming specifications for their microcontrollers. That does not give very good protection on its own, and only slightly increases the cost of attack, because this information could be extracted by observing the signals applied to the chip during programming in a development kit or in a universal programmer.

Obviously, for the highest security, the system would not have any programming interface at all, and would not provide any access to stored data. This is normally the case for Mask ROM microcontrollers and smartcards. The only practical ways of attacking in this case would be either to microprobe the data bus to recover the information or use power analysis and glitch attacks to exploit any vulnerability in software. Relatively high security can be obtained when a microcontroller is user programmable but does not provide any read-back facility – only verify and write check, for example in the NEC 78K0S family Flash microcontrollers [58]. Of course this should be implemented properly to avoid the situation where the attacker can force the system to verify one byte at a time. In this case he would need on average 128 attempts per byte ($2^8 \times 0.5$) and assuming the byte access cycle is 5 ms it will take him less than a day to extract the contents of the memory, which is usually between 4 Kb and 64 Kb. Even if the verify operation is applied to large blocks of data, the attacker could try glitch attacks to reduce the cycle to a single byte.

Most microcontrollers on the market have a security fuse (or multiple fuses) that control access to the information stored in on-chip memory. These fuses could be implemented in software or in hardware. Software implementation means that a password is stored in the memory or a certain memory location is assigned as a security fuse. For example, in the Motorola MC68HC908 family, password protection is used, and in the Motorola MC68HC705B family, the security fuse is located in the first byte of the data EEPROM memory. Both variants have relatively high security, because it is extremely difficult to find the physical location of the fuse or password and reset them. At the same time, the attacker can try using glitch attacks to override the security check subroutine, or use power analysis to see whether a password guess is correct or not.

In hardware implementation, security fuses are physically located on the chip die. This could mean a separate memory cell located next to the main memory array, or even far from it. For example, this is the case for all Microchip PIC and Atmel AVR microcontrollers. In both cases, the security is not very high as the fuses can be easily found and disabled by one or another method. Meantime, some methods require very expensive equipment and even if the attacker knows where the fuse is, he will not be able to reset it until he gets access to such equipment and learns how to use it.

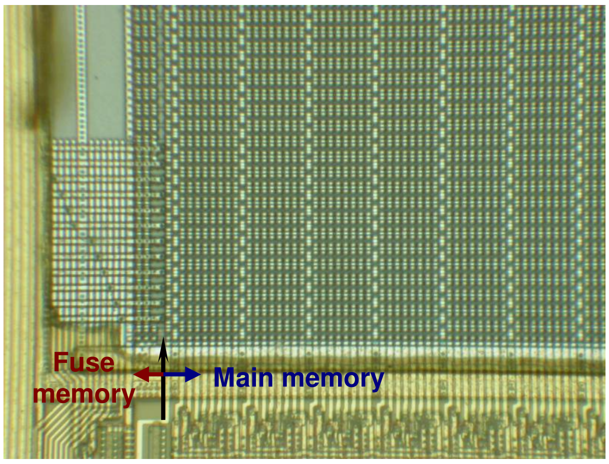


Figure 31. Fuse memory array touches the main memory array along the bit-lines in Z86E33 microcontroller. 200× magnification

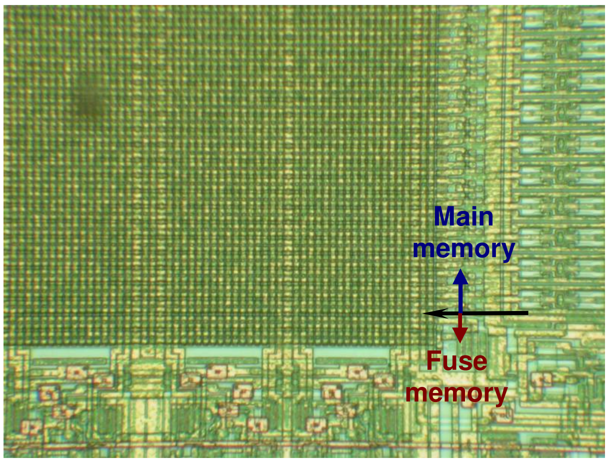


Figure 32. Fuse memory array touches the main memory array along the word-lines in ST62T60 microcontroller. 200× magnification

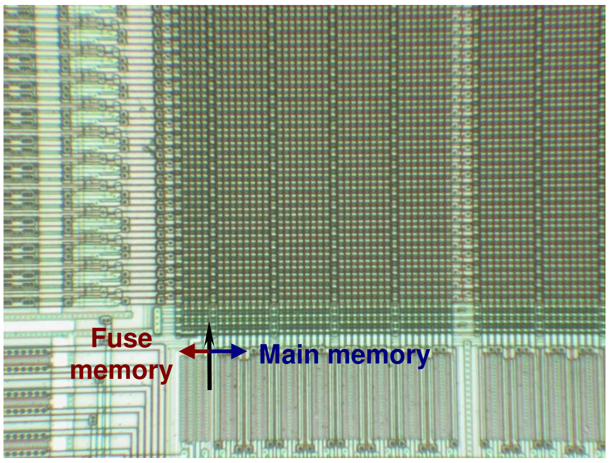


Figure 33. Fuse memory array shares word-lines with the main memory array in HD6473048 microcontroller. 200× magnification

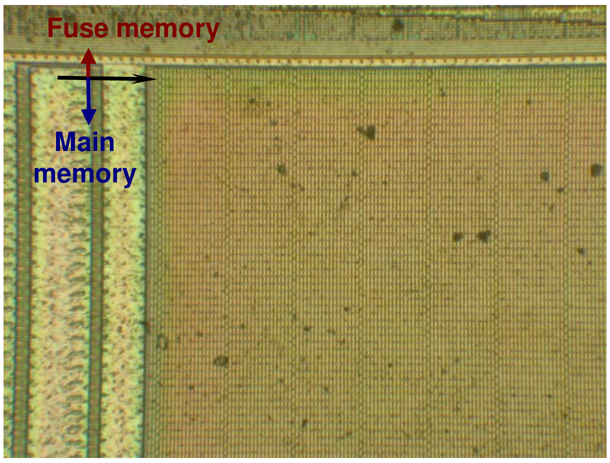


Figure 34. Fuse memory array shares bit-lines with the main memory array in HT48R50A microcontroller. 200× magnification

Better protection can be achieved when the security fuses are located on the same memory array but with separate control and signal lines. In this case it is very difficult to find and disable them. Several different implementations are possible. Main and fuse memory arrays can touch each other with bit-lines, for example, as in the Zilog Z86E33 microcontroller (Figure 31) [59]; or with word-lines, as in the STMicroelectronics ST62T60 microcontroller (Figure 32) [60]. A very interesting, and much more secure solution, was used in the Motorola MC68HC705C9A microcontroller (see Figure 6) where the fuse cells were placed in between the main memory cells so that the bit-lines were mixed together. Even if the fuses could be erased with UV light, very likely the attacker would damage the memory area trying to erase them. Reverse engineering of the memory structure to figure out which part belongs to the memory and which to the fuses is not an easy task and demands very high skills from the attacker. At the same time, semi-invasive methods could work very well because the fuses have a separate control circuit which could be attacked without affecting the main memory.

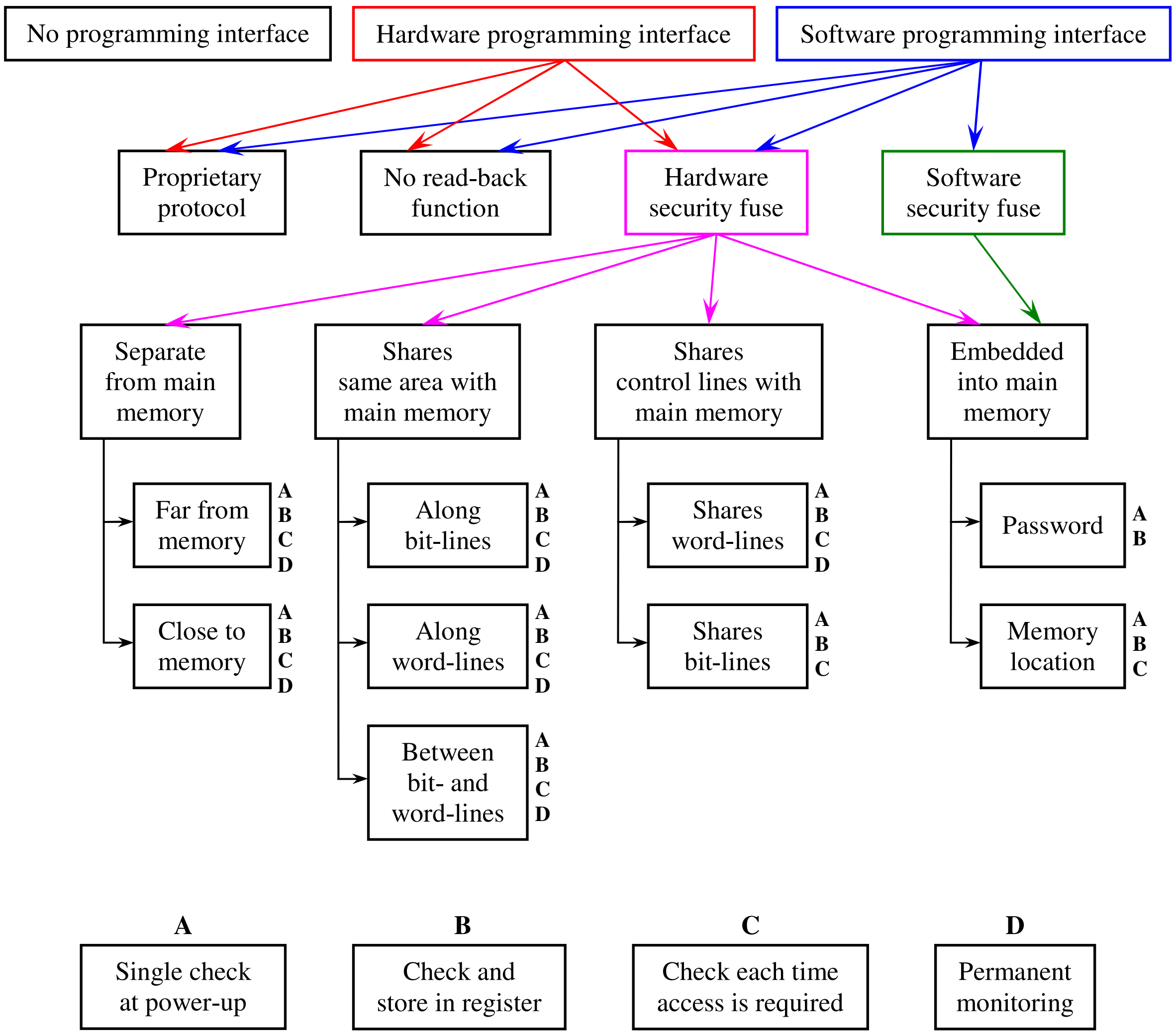


Figure 35. Security protection implementations in different microcontrollers

The next improvement to hardware security protection was done by embedding the fuse area into the main memory array so that it shares some of the control or data lines. This implementation is more secure because the fuses are part of the memory array and their localisation is very difficult and challenging task. Fuses can share word-lines with the main memory, for example, as in the Hitachi HD6473048 microcontroller (Figure 33) [61]; or they can share bit-lines as in the Holtek HT48R50A microcontroller (Figure 34) [62]. In the latter implementation the fuses do not have a separate bit-lines that could be attacked. But that does not mean it will be more secure because the state of the fuses cannot be monitored all the time and usually is sampled at power-up and stored in a separate register.

A high level of security can be achieved if a certain memory location is used as a security fuse. In this case it would be extremely difficult to find this location and reset it without disturbing the contents of other memory cells. That does not mean that other attack methods will not work, such as, non-invasive attacks, but at least this reduces the chance of success with simple semi-invasive attacks.

Apart from different implementations, the security fuse can be monitored in different ways. The simplest way is to check the state of the fuse on power-up, on reset, or on entering the programming mode. This is not good, as the state of the fuse can be changed for a short time by power glitch or laser pulse. Storing the fuse state in a flip-flop or register is not much better, because the fuse state is checked only once and the flip-flop could be changed with a fault injection attack. Having the fuses checked each time access is requested is better. It still gives an opportunity for the attacker to synchronise his attack with the fuse checks, but will require more effort. The highest security can be obtained if the fuse state is constantly monitored and affects memory access. In this case the attacker will have to permanently disable the fuse to access the information.

Figure 35 summarises all the above discussed possibilities for different implementations of security protection.

2.2 Developers and attackers

As can be seen from the discussion so far, modern microcontrollers have some potential problems with security protection. Microcontroller-based systems engineers are often interested in having maximum protection for their intellectual property. One approach is to rely on the information provided by the chip manufacturers. Unfortunately very little information about the actual protection can be obtained from the datasheets. The only information provided is what type of the security was implemented, e.g. password, lock bits or no read-back; and how to program or activate the security protection via programming interface. This is not enough information to estimate which microcontroller is better from the security point of view. It gives the impression that all microcontrollers offer the same, relatively high, level of security, which is not true.

In this situation the only reliable way to evaluate the security in one or another microcontroller is to either find a security evaluation company for this job or do it on your own. Both are quite expensive, because commercial companies charge a lot, and building your own laboratory is not an easy task and requires knowledge and experience. Evaluation should include not only the security protection itself but memory type, internal chip structure, and programming interface, as they all affect the hardware security of the chip.

The best security tester for any product is the open market. Provided a chip is available for long enough, developers can estimate how secure it is by observing the time between the production of any useful device using it and the time its clones are available. Of course this gives only a rough estimate as some products could be cloned without breaking the microcontroller they are based on, and some products might be not so attractive to clone. In addition, developers can use some tricks to increase product security; some of them are discussed in Chapter 8. The developer should also pay attention to the security of his embedded software, and protocols for communication with other devices, as all of them could leak some sensitive information as well.

Chip manufacturers indirectly confirm that security in their products may be limited. For example, phrases like “No security feature is absolutely secure. Our strategy is to make reading or copying the ROM difficult for unauthorized users”, “Due to the nature of this feature it cannot be tested and therefore not guaranteed” and “Code protection does not mean that we are guaranteeing the product as ‘unbreakable’” are typical for the datasheets on microcontrollers [63][64][65]. Even for smartcards the security is not guaranteed and the manufacturer will not accept responsibility if one of their products is broken. What they claim is that protection against certain attack methods is implemented, and they are constantly working on increasing the security level of their products. The security of smartcards is increased by the fact that detailed information, datasheets and development tools are sold under strict non-disclosure agreement, while the smartcards themselves are sold in large quantities and to authorised buyers only.

Smartcard, CPLD and FPGA manufacturers are trying to attract customers who need the highest level of security protection. Of course they put a lot of effort into chip design to avoid any existing attacks but sometime they come up with ridiculous claims about protection against new and not properly understood attacks. For example, when fault injection and laser attacks become known, some manufacturers rushed to claim that their products were designed to withstand these attacks. The strange thing is that these announcements appeared a couple of months after such attacks become known, while it takes over a year to design a new silicon family chip.

More funny stories happened when some CPLD and FPGA chip families were announced to be the most secure products on the market. One manufacturer wrongly implemented secret key protection which resulted in an all-zeros-key programmed all the time. Another manufacturer simply forgot to activate the programming of the security fuse in the programming software so that, although the fuse was reported programmed, in fact it was not. As a result some products based on these chips were successfully hacked. Of course the problems were fixed very quickly, but blackened the reputation of these manufacturers. If the developers had tested the security of these chips and chosen others, their products would not have been cloned. But because all the protocols used for CPLD and FPGA programming are proprietary, it is not an easy task to evaluate these products. If the protocols were publicly available, the developers would be able to check if the security was properly activated on these chips and either inform the manufacturer to update the software or else write their own programming software.

Another problem that affects hardware security is the fact that usually a whole family of chips from one manufacturer has the same implementation of the security protection. It means that once an attacker finds a way to overcome the security in one device, very likely he would be able to break another. Manufacturers do change the security protection from time to time, but again that affects a wide range of products simultaneously.

Nowadays attackers are very clever. They do not believe in what the manufacturers claim about the security of their products. They are constantly looking for new and low-cost attack methods, and they never give up. As a result there is a permanent battle between the manufacturers who are trying to improve the security of their products and the attackers who are constantly breaking these products. There is no real change in this process within the last decade – only temporary shifts of the front line from time to time. For sure modern smartcards are extremely secure, but attackers are not idle and sometimes are very successful. That forces the developers to update their products quite often.

2.3 Failure analysis techniques

Failure analysis involves testing and debugging silicon chips after fabrication. Very often the chip does not function in the required way, so the manufacturer wants to investigate the problem and fix it in the next revision of the die. When a new technological process or memory type is being developed, failure analysis techniques are used to measure all the parameters and make necessary alterations in further designs. Obviously such tools should provide the ability to observe signals at any point of the chip and, if necessary, make modifications to the silicon design. From the attacker's point of view, a perfect failure analysis tool gives the ultimate capability to circumvent security protection. It allows connection to any point on the chip die, and lets him disable the security protection by modifying the security circuit. Fortunately, with constant technological progress resulting in a significant reduction of the transistor feature sizes, failure analysis becomes more and more complicated and expensive. It also forces the attackers to be more and more knowledgeable and experienced. Of course, not all failure analysis techniques are useful for breaking the security of chips. For example, an attacker is not interested in cross sectioning the chip, transistor sizes, thickness of the gate oxide or metallization.

The first operation which is crucial for any invasive or semi-invasive attack is decapsulation of the chip sample to get access to the die surface. There are different techniques for doing this [66] and the most widely known and reliable method involves using hot fuming nitric acid to dissolve the plastic package material. A detailed explanation of this method is given in Chapter 5. Modern chips have multiple metal layers and in order to investigate and analyse the structure of the chip, the attacker must expose each layer, photograph it under a microscope and then combine all the photos together to get a complete picture. Then he could trace the signals from one transistor to another and simulate the whole chip. This process is called reverse engineering and a basic overview of it is given in Chapter 5. For many years microprobing technology was used to observe the signals inside the chip during operation. This is a basic and simple way of extracting the information from semiconductor chips. It is not cheap, as it requires a probing station with micropositioners, which is quite expensive. Also, in order to establish a contact with the internal metal wire, the attacker will need either a laser cutter or a FIB machine which are both very expensive. A short introduction into these tools is given in Chapter 5.

Failure analysis has recently acquired more sophisticated and effective techniques that allow probing any point on the chip without establishing physical contact [67]. Such techniques could be also called semi-invasive, because they require decapsulation of the sample to get access to the die, but do not require any modifications to the chip or establishing physical contact with its surface or internal wires. They involve different types of microscopy and advanced probing techniques. Some of these techniques allow access to the on-chip transistors from the rear side of the die, which is very desirable for modern chips with multiple metal layers covering 99% of the chip surface. For example, thermal imaging can be used to find the active area, which has a higher temperature, then photoemission microscopy can be used to find the active transistor. Infrared microscopy can be used to look through rear side of the chip as silicon is partially transparent to infrared light. If necessary the thickness of the silicon substrate can be locally reduced down to ten micrometers by mechanical milling or chemical etching. Active photon probing can be used to find the transistors and p-n junctions on the chip surface, while light-induced probing can reveal the state of each individual transistor. These techniques are presented in Chapter 6. A scanning electron microscope (SEM) can be used for different purposes from imaging deep submicron structures in modern chips, to reading the voltages directly from the metal wires inside the chip. Modern FIB machines can create test points and modify the chip structure from the rear side thus overcoming sophisticated top metal mesh protections and sensors [68].

Current trends in the miniaturisation of electronic devices demand the ability to understand the structure and properties of the deep submicron level (the latest technology is 90 nm, and 65 nm is proposed). Within the last few years nanotechnologies have become widely available and started to be used in failure analysis. One of them is atomic force microscopy (AFM). It gives deep submicron resolution of the analysed surface [69]. The same idea was used to build a large group of microscopes called scanning probe microscopy (SPM). It allows observation of many of the characteristics of semiconductor chip surface [70]. It has a large number of implementations, one of which (called scanning capacitance microscopy, SCM) has very useful applications for semiconductor failure analysis. It can be used to measure doping concentrations inside each individual transistor [71]. That can be used, for example, to extract the contents from the previously highly secure VTROM memory. Other variants of SPM include, but are not limited to the following measurement techniques: surface resistance with scanning spreading resistance microscopy (SSRM), atomic structure and Fermi level with scanning tunneling microscopy (STM), magnetic field with magnetic force microscopy (MFM), electric field with scanning surface potential microscopy (SSPM) and temperature with scanning thermal microscopy (SThM).

Most of the modern failure analysis techniques require very expensive equipment and highly skilled attackers. Only large laboratories and chip manufacturers can afford them. On the other hand, an attacker does not need all this equipment as he will often succeed with relatively low-cost equipment. Some equipment can be bought second-hand or built from available components. Chip manufacturers get rid of old equipment at auctions. There are also some companies dealing with second-hand equipment [72]. This equipment cannot be used to test and attack modern deep submicron chips but should be good enough for devices over five years old. There was a large breakthrough during the last five years in the design of the specialist microscopes mentioned above. Their prices have dropped significantly and they have become available to attackers with about fifty thousand pounds of funding.

Chapter 3

Attack technologies

Secure microcontrollers and smartcards are designed to protect both the confidentiality and the integrity of sensitive information. It is not sufficient to prevent an attacker from finding out the value of a stored cryptographic key; he must also be unable to set part of the key to a known value, or to induce errors in the computation that enable sensitive information to be deduced. These errors may be data errors, such as an incorrect digital signature that leaks the value of the signing key [9], or errors in the code, such as a missed conditional jump that reduces the number of rounds in a block cipher [10].

3.1 Introduction

Evaluating the level of tamper resistance offered by a given product is a very important problem. Unfortunately, very little attention has been given to hardware security evaluation by the security research community. Even chip manufacturers are trying to avoid discussions about the security protection schemes implemented in their products. If a problem in hardware security design is found, they try to solve it as quietly as possible and release another revision of the chip.

Not much information about security protection can be found in datasheets on secure products. Normally they only list the attacks against which the protection was designed and do not discuss any implementation details. The critical question is always whether an opponent can obtain unsupervised access to the device. If the answer is no, then relatively simple measures may suffice. For example, the VISA security module is vulnerable to people with occasional access: a service engineer could easily disable the tamper protection circuitry on one of his visits, and extract key material on the next. But this is not considered to be a problem by banks, who typically keep security modules under observation in a computer room, and closely supervise service visits.

In an increasing number of applications the opponent can obtain completely unsupervised access not just to a single instance of the cryptographic equipment but to many of them. This is the case that most interests us: it includes microcontrollers for industrial applications, pay-TV smartcards, prepayment meter tokens, protection dongles for software, hardware identification tags, remote locking devices for cars and SIM cards for GSM mobile phones [73]. Many such systems are already the target of well funded attacks.

In a well known article from IBM [74], where the design of a range of IBM products is discussed, attackers can be grouped into three classes, depending on their expected abilities and attack strength:

Class I (clever outsiders):
They are often very intelligent but may have insufficient knowledge of the system. They may have access to only moderately sophisticated equipment. They often try to take advantage of an existing weakness in the system, rather than try to create one.

Class II (knowledgeable insiders):

They have substantial specialised technical education and experience. They have varying degrees of understanding of parts of the system but potential access to most of it. They often have access to highly sophisticated tools and instruments for analysis.

Class III (funded organisations):
They are able to assemble teams of specialists with related and complementary skills backed by great funding resources. They are capable of in-depth analysis of the system, designing sophisticated attacks, and using the most advanced analysis tools. They may use Class II adversaries as part of the attack team.

3.1.1 Protection levels

It is not an easy task to estimate the protection level of a semiconductor chip as so many factors should be taken into consideration from the chip's package and die layout to memory structure, memory type, programming and access interfaces, security fuses or secret key location, protection mechanisms and other security features such as glitch detection, power supply voltage monitors, protection meshes, tamper resistance etc. There is no straightforward way to evaluate the hardware security of a semiconductor device; what normally has to be done is to apply different attack methods and observe the result. The more attacks tested, the more confidence in the result. From the other side, semi-invasive methods are much easier to automate than non-invasive methods and at the same time they require much cheaper equipment than invasive methods. That makes semi-invasive methods very attractive when a quick and relatively inexpensive hardware security evaluation is required. Where any problem is found in the design, the more expensive but accurate invasive methods should be used to locate and eliminate the problem. This will involve more expensive equipment, as well as more knowledgeable engineers to operate it and understand the results.

The same article from IBM [74] discusses the protection levels a secure system can provide against different attacks. Their classification suggests six security levels starting from a zero level corresponding to the system without any security protection to a high level for the virtually unbreakable system. There might, of course, be all sorts of intermediate levels which can be used to compare the devices with each other.

Level ZERO:
No special security features are used in the system. All parts have free access and can be easily investigated. Example: microcontroller or FPGA with external ROM.

Level LOW:
Some security features are used but they can be relatively easy defeated with minimum tools required such as soldering iron and low cost analog oscilloscope. Attack takes time but does not involve more than £1,000 of equipment. Example: microcontroller with unprotected internal memory but proprietary programming algorithm.

Level MODL:
Security used protects against most low cost attacks. More expensive tools are required as well as some special knowledge. Total equipment cost does not exceed £3,000. Examples: microcontrollers sensitive to power analysis and power glitches.

Level MOD:
Special tools and equipment are required for successful attack as well as some special skills and knowledge. Total equipment cost is up to £30,000. Only Class II attackers can afford this. The attack could be time-consuming. Examples: microcontrollers with protection against UV attacks; old smartcard chips.

Level MODH:
Special attention is paid to design of the security protection. Equipment is available but is expensive to buy and operate. Total equipment cost is up to £150,000. Special skills and knowledge are required to utilise the equipment for an attack. A group of Class II attackers may be required with complementary skills to work on the attack sequence. Examples: modern smartcard chips with advanced security protection; complex ASICs; secure FPGAs and CPLDs.

Level HIGH:
All known attacks are defeated and some research by a team of specialists is necessary to find a new attack. Highly specialised equipment is necessary, some of which might have to be designed and built. Total cost of the attack could be over a million pounds. The success of the attack is uncertain. Only large organisations like semiconductor manufacturers or government funded laboratories could afford this. Examples: secure cryptographic modules in certification authority applications.

For applications or devices that include cryptography, U.S. and Canadian federal government agencies are required to use a cryptographic products that has been FIPS 140 (Federal Information Processing Standards) validated [75] or Common Criteria validated [76]. Most Common Criteria protection profiles rely on FIPS validation for cryptographic security. Within the FIPS 140-2 (or 140-1) validations, there are four possible security levels for which a product may receive validation.

Security Level 1 provides the lowest level of security. It specifies basic security requirements for a cryptographic module.

Security Level 2 improves the physical security of a Level 1 cryptographic module by adding the requirement for tamper evident coatings or seals, or for pick-resistant locks.

Security Level 3 requires enhanced physical security, attempting to prevent the intruder from gaining access to critical security parameters held within the module.

Security Level 4 provides the highest level of security. The physical security provides an envelope of protection around the cryptographic module to detect a penetration into the device from any direction.

The security level of a particular device does not last forever. It is possible that a low cost attack will be found in the future when the attack tools become cheaper or available at second-hand.

3.1.2 Attack categories

For security evaluation we will assume that all attackers can obtain several examples of the target equipment. We will concentrate on attacks aimed at recovering security algorithms and crypto key material stored in microcontrollers, smartcards and other chip-level security processors.

We can distinguish five major attack categories:

Microprobing techniques can be used to access the chip surface directly, so we can observe, manipulate, and interfere with the integrated circuit.

Reverse engineering is used to understand the inner structure of semiconductor chip and learn or emulate its functionality. It requires the use of the same technology available to semiconductor manufacturers and gives similar capabilities to the attacker.

Software attacks use the normal communication interface of the processor and exploit security vulnerabilities found in the protocols, cryptographic algorithms, or their implementation.

Eavesdropping techniques allows the attacker to monitor, with high time resolution, the analog characteristics of supply and interface connections and any electromagnetic radiation by the processor during normal operation.

Fault generation techniques use abnormal environmental conditions to generate malfunctions in the processor that provide additional access.

All microprobing and reverse engineering techniques are invasive attacks. They require hours or weeks in specialised laboratory and in the process they destroy the packaging. The other three are non-invasive attacks. The attacked device is not physically harmed during these attacks. The last attack category could also be semi-invasive. It means that the access to the chip's die is required but the attack is not penetrative and the fault is generated with intensive light pulse, radiation, local heating or other means.

Non-invasive attacks are particularly dangerous in some applications for two reasons. Firstly, the owner of the device might not notice that the secret keys or data have been stolen, therefore it is unlikely that the validity of the compromised keys will be revoked before they are abused. Secondly, non-invasive attacks often scale well, as the necessary equipment can usually be reproduced and updated at low cost.

The design of most non-invasive attacks requires detailed knowledge of both the processor and software. On the other hand, invasive microprobing attacks require very little initial knowledge and usually work with a similar set of techniques on a wide range of products. Attacks therefore often start with invasive reverse engineering, the results of which then help to develop cheaper and faster non-invasive attacks. Semi-invasive attacks can be used to learn the device functionality and test its security circuits. As these attacks do not require establishing any physical contact to the internal chip layers, expensive equipment such as laser cutters and FIB machines are not required. The attacker could succeed using a simple off-the-shelf microscope with a photoflash or laser pointer attached to it.

Attacks can be reversible when the device can be put back into the initial state, or irreversible with permanent changes done to the device. For example, power analysis and microprobing could give the attacker a result without harming the device itself. Certainly microprobing will leave tamper evidence but usually that does not affect further device operation. On the contrary, fault injection and UV light attacks could very likely put the device into the state where the internal registers or memory contents are changed and cannot be restored. In addition, UV attacks leave tamper evidence as they require direct access to the chip surface.

3.1.3 Attack scenarios

Attacks can be used for different purposes depending on the goal. Sometimes copying a profitable on-the-market product can give easy money. Larger manufacturers could consider stealing intellectual property (IP) from the device and mixing it with their own IP to disguise the theft. Others could try to steal secrets from the device either to produce a competitive product or to steal service. Product designers should first think about the possible motives for attacking their devices and then concentrate on the protection mechanisms. The following attack scenarios should be considered during the system design.

Cloning is one of the most widely used attack scenarios. It is used by a large variety of attackers from individuals, who want cheaper electronic gadgets, to large companies interested in increasing their sales without large investment in design. For example, dishonest competitors may try to clone existing products to reduce development costs. Of course they will have to spend some effort to disguise the fact of piracy, but compared to honest development cost this is negligible. Normally cloning requires reverse engineering of the device to some extent.

Overbuilding is the easiest form of IP piracy. It takes place when a contract manufacturer builds more than the requested quantity of electronic devices. The extra devices can be then sold on the market. The design could also be sold to third parties.

Theft of service could happen when electronic devices are used to provide access to some information or service. For example, cable and satellite TV companies control the channels a viewer can see. If a pirate can bypass security or simulate the device, the service provider will lose. As the pirates normally work in a large community, any success is distributed among all members of the group, so it incurs huge losses to the service provider.

Denial of service can be used by a competitor to damage a vendor's product. This could happen when the device firmware is updated over a network. If the competitor manages to reverse engineer the device and work out the update protocol, he could launch a malicious update code and then switch off all the devices or even damage them by uploading bad code. For example, it is possible to permanently damage an FPGA device by uploading a bad configuration file. Also, modern microcontrollers and smartcards have Flash memory for the program code. If an erase command is issued for all memory blocks, then the device will stop operating for good. The developer should design firmware update features very carefully to make sure they cannot be used without proper authentication.

3.2 Non-invasive attacks

The most widely used non-invasive attacks include playing around with the supply voltage and clock signal. Under-voltage and over-voltage attacks could be used to disable protection circuit or force a processor to do the wrong operation. For these reasons, some security processors have a voltage detection circuit, but this circuit cannot react to fast transients. Power and clock transients can also be used in some processors to affect the decoding and execution of individual instructions.

Another possible attack uses current analysis. We can measure with an analog-to-digital converter the fluctuations in the current consumed by the device. Drivers on the address and data bus often consist of up to a dozen parallel inverters per bit, each driving a large capacitive load. They cause a significant power-supply short circuit during any transition. Changing a single bus line from '0' to '1' or vice versa can contribute in the order of 0.5–1mA to the drain current right after the clock edge. So a 12-bit ADC is sufficient to estimate the number of bus bits that change at anyone time. SRAM write operations often generate the strongest signals.

Another possible threat to secure devices is data remanence. This is the capability of volatile memory to retain information for some time after power is disconnected. Static RAM storing the same key for a long period of time can reveal it on next power on [77]. Another possibility is to 'freeze' the memory by applying low temperature. In this case, static RAM can retain information for enough time to get access to the memory chip and read its contents. Data remanence can take place in non-volatile memories as well; the residual charge left on a floating gate transistor may be detected. For example, it could affect a threshold level or time-switching characteristics.

The next possible way of attacking a device is playing around with its interface signals and access protocols. Also, if a security protocol is wrongly implemented, that leaves a hole for the attacker to exploit. Some microcontrollers and smartcards have a factory-test interface that provides access to on-chip memory and allows the manufacturer to test the device. If an attacker can find a way of exploiting this interface, he can easily extract the information stored inside the chip. Normally information on test circuits is kept secret by the manufacturer, but an attacker can try applying different voltages and logic levels to the pins in the hope that it will put it into test mode. This sometimes works for microcontrollers but in smartcards such test circuitry is usually destroyed after use. Also, embedded software developers sometimes implement functions that allow downloading from internal memory for test and update purposes. That must be done in a way that prevents any access to the code without proper authentication, or so that the code can be sent out in encrypted form only.

Examples of different non-invasive attacks are presented and discussed in Chapter 4.

3.3 Invasive attacks

Despite the greater complexity of invasive attacks, some of them can be done without expensive laboratory equipment. Low-budget attackers are likely to get a cheap solution on the second-hand market for semiconductor test equipment. With patience and skill, it should not be too difficult to assemble all the required tools for under ten thousand pounds by buying a second-hand microscope and using self-designed micropositioners.

Invasive attacks start with the removal of the chip package. Once the chip is opened it is possible to perform probing or modifying attacks. The most important tool for invasive attacks is a microprobing workstation. Its major component is a special optical microscope with a long working distance objective lens. Micropositioners are installed on a stable platform around the chip test socket and allow the movement of probe arms, with submicron precision, over a chip surface. A probing needle with an elastic hair at the end is installed on each arm and allows electrical contact to on-chip bus lines without damaging them.

On the depackaged chip, the top-layer aluminium interconnect lines are still covered by a passivation layer (usually silicon oxide or nitride), which protects the chip from the environment and ion migration. This passivation layer must be removed before the probes can establish contact. The most convenient depassivation technique is the use of a laser cutter. Carefully dosed laser flashes remove patches of the passivation layer. The resulting hole in the passivation layer can be made so small that only a single bus line is exposed. This prevents accidental contacts with neighbouring lines and the hole also stabilizes the position of the probe, making it less sensitive to vibration and temperature changes.

It is not usually practical to read the information stored on a security processor directly out of each single memory cell, except for ROM. The stored data has to be accessed via the memory bus where all data is available at a single location. Microprobing is used to observe the entire bus and record the values in memory as they are accessed.

In order to read all memory cells without the help of the device software, we have to abuse a CPU component such as an address counter to access memory for us. The program counter is already incremented automatically during every instruction cycle and used to read the next address, which makes it perfectly suited to serve us as an address sequence generator. We only have to prevent the processor from executing jump, call, or return instructions, which would disturb the program counter in its normal read sequence. Tiny modifications of the instruction decoder or program counter circuit, which can easily be performed by opening the right metal interconnect with a laser, often have the desired effect.

Another approach to understanding how a device work is to reverse engineer it. The first step is to create a map of the processor. It could be done by using an optical microscope with a CCD camera to produce several meter large mosaics of high-resolution photographs of the chip surface. Basic architecture structures, such as data and address bus lines, can be identified quite quickly by studying connectivity patterns and by tracing metal lines that cross clearly visible module boundaries (ROM, RAM, EEPROM, ALU, instruction decoder, etc.). All processing modules are usually connected to the main bus via easily recognizable latches and bus drivers. The attacker obviously has to be familiar with CMOS VLSI design techniques and microcontroller architectures, but the necessary knowledge is easily available from numerous textbooks.

Most currently available microcontrollers and smartcard processors have feature sizes of 0.25–0.5 μm and two to four metal layers. These can be reverse-engineered and observed with manual and optical techniques, but require some specific deprocessing operations to remove one metal layer after another. For future generations of microcontrollers with more metal layers and features below the wavelength of visible light, it may be necessary to use more expensive tools such as scanning electron microscopes (SEM).

The most common tool used for failure analysis and to apply any modifications to the chip structure is a focused ion beam (FIB) machine. It consists of a vacuum chamber with a particle gun, comparable to a SEM. With a FIB machine the attacker can cut the metal and polysilicon interconnections and build new ones with a deep submicron precision. Using laser interferometer stages, a FIB operator can navigate blindly on a chip surface. Chips can also be polished from the rear side down to a thickness of just a few tens of micrometers. Using laser interferometer navigation or infrared imaging, it is then possible to locate individual transistors and contact them through the silicon substrate by FIB editing a suitable hole. This rear-access technique has probably not yet been used by pirates so far, but the technique is about to become much more commonly available and therefore has to be taken into account by designers of new security chips. FIBs are primarily used by attackers today to simplify manual probing of deep metal and polysilicon lines. A hole is drilled to the signal line of interest and then filled with platinum to bring the signal to the surface, where a several micrometer large probing pad is created to allow easy access. Modern FIB workstations (for example the FIB 200xP from FEI) cost less than half a million pounds and are available in over a hundred organizations including Universities. Some old FIB models are available on a second-hand market at a price of less than fifty thousand pounds.

Some basic invasive attack techniques are discussed in Chapter 5 together with sample preparation techniques.

3.4 Semi-invasive attacks

There is a large gap between previously discussed non-invasive and invasive types of attack and many attacks fall into this gap, being not so expensive as classical penetrative invasive attacks but as easily repeatable as non-invasive attacks. Therefore we decided to define and introduce a new class of attack called semi-invasive. Like invasive attacks, they require depackaging the chip in order to get access to its surface. However, the passivation layer of the chip remains intact, as semi-invasive methods do not require depassivation or creating contacts to the internal lines. This is because microprobing is not used for this attack technology and thus such expensive tools as laser cutters and FIBs are not required.

Semi-invasive attacks are not entirely new. UV light has been used to disable security fuses in EPROM and OTP microcontrollers for many years. Modern microcontrollers are less susceptible to this attack as they were designed to withstand it. More information on the evolution of defences against UV attacks in microcontrollers is given in Chapter 7.

Advanced imaging techniques can be considered as semi-invasive as well. This includes various kinds of microscopy such as infrared, laser scanning and thermoimaging. Some of them can be applied from the rear side of the chip which is very useful for modern chips with multiple metal layer design. Some of these techniques allow observation of the state of each individual transistor inside the chip.

One of the main contributions of this thesis is fault injection attacks done in a semi-invasive manner which can be used to modify the contents of SRAM and change the state of any individual transistor inside the chip. That gives almost unlimited capabilities to the attacker in getting control over the chip operation and abusing the protection mechanism.

Compared to non-invasive attacks, semi-invasive attacks are harder to implement as they require decapsulation of the chip. However, very much less expensive equipment is needed than for invasive attacks. These attacks can be performed in a reasonably short period of time. Also they are scalable to a certain extent, and the skills and knowledge required to perform them can be easily and quickly acquired. Some of these attacks, such as an exhaustive search for a security fuse, can be automated. If compared to invasive attacks, the semi-invasive kind do not normally require precise positioning for success because they are normally applied to a whole transistor or even a group of transistors rather than to a single wire inside the chip.

The many examples of semi-invasive attacks described in Chapter 6 and Chapter 7 shows how they can be used for hardware security analysis.

Chapter 4

Non-invasive attacks

A non-invasive attack does not require any initial preparations of the device under test. The attacker can either tap the wires to the device, or plug it into a test circuit for the analysis. Once found, these attacks could be easily scaled and their reproduction does not involve very much cost. In addition, no tamper evidence is left after they are applied. Therefore they are considered to be the most serious threat to the hardware security of any device. At the same time it usually takes a lot of time and effort to find an attack on any particular device. This often involves reverse engineering the device in the sense of either disassembling its software or understanding its hardware layout.

Non-invasive attacks can be either passive or active. Passive attacks, also called side-channel attacks, do not involve any interaction with the attacked device but, usually, observation of its signals and electromagnetic emissions. Examples of such attacks are power analysis and timing attacks. Active attacks, like brute force and glitch attacks, involve playing with the signals applied to the device including the power supply line.

One example of a simple non-invasive attack could be cloning a device based on SRAM FPGA as it is configured at a power-up. The attacker could easily connect to the JTAG interface wires used for configuring the chip and, with either an oscilloscope or a logic analyser, grab all the signals. Then he can thoroughly analyse the waveforms and replay the commands in his own design. He could also slightly change the bitstream to disguise the fact of cloning as usually only half of the FPGA resources are used, leaving a room to fiddle with the configuration without harming device operation. Also the JTAG interface itself gives some freedom in the sequence of the signals being applied so that the waveforms used to configure the pirate copy will look different from the original. In addition, the attacker could mix the row addresses during the upload, giving the impression of a completely different design.

Another example is when the attacker invests a huge amount of money to reverse engineer a pay-TV access card. Then he disassembles the internal code from the card, learning everything that happens during authorisation and operation. Very likely he would be able to find vulnerabilities which give unlimited access to the subscription channels, for example, by applying a power glitch at just the right moment to cause a malfunction of the CPU. Once he succeeded he could either offer the subscription service at a very competitive price, or sell equipment for counterfeiting the card to malicious people. Obviously such an attacker needs to invest some capital to do this. But once he launches a pirate device on the market, it will be attacked by others. This time the attack will not be so expensive, because pirate devices are normally based on standard microcontrollers which have much lower security protection than pay-TV smartcards. Very likely the device will be cracked in a few weeks, and the secondary attackers will flood the market with their clones. Fairly soon, the information on how to build pirate devices becomes available on the Internet and anyone can build pirate devices at almost no cost. So the pay-TV service provider loses millions of dollars; sometimes the original attacker is sued or prosecuted. But because the lost profit was distributed among all the pirates and dishonest subscribers, the service provider hardly gets any money back. The only effect of such actions is to threaten the hacker community with punishment. In addition the service provider will have to spend a fortune on redesigning his access control system, choosing and developing software for the new smartcard, and distributing cards to the subscribers.

4.1 Obscurity vs security

Semiconductor manufacturers offer valuable customers an easy way to increase the protection of their products: chips with custom marking on the packages instead of standard chip names. That gives the impression that the final product was designed using ASICs or full custom ICs. 'Everyone knows' that ASICs offer very good protection against different sorts of attacks and only well equipped and highly skilled attackers could succeed with breaking them. This may stop many potential attackers fiddling with the product. However, a determined attacker could try an easy way to check whether this chip was actually an ASIC. The easy way is to note which pins are connected to power supply, ground, clock, reset, serial, and other interfaces, and to compare all this information with the database of suspect microcontrollers or other ICs. This works very reliably, as each microcontroller family has its own characteristic pinout. Once similarities are found the suspected microcontroller could be verified by placing it into a programming device or universal programmer and trying to read it.

Another simple trick many semiconductor manufacturers use is restricting access to information on memory programming. This is normally used for smartcards, but on some microcontrollers such information is not publicly available as well. This is not a reliable and practical way of making the design secure. Of course it works well with smartcards where all the customers are obliged to sign a non-disclosure agreement with the chip manufacturer. But microcontrollers, with very few exceptions, can be programmed with universal programmers that are widely available from different companies around the world. Even if the programming specification is not documented, all the necessary waveforms can be easily extracted in a few hours with using any low cost oscilloscope, because all the signals are normally applied with less than 1 MHz frequency. If the microcontroller is not supported by a particular universal programmer, it is always possible to buy the development kit directly from the manufacturer and obtain all the necessary protocols from it directly.

4.2 Timing attacks

Some security-related operations a semiconductor chip performs can take a different time to compete depending on the values of the input data and the secret key. Careful timing measurement and analysis may allow recovery of the system's secret key. This idea was first published in the scientific literature in 1996 [78]. Then later these attacks were successfully performed on an actual smartcard implementation of the RSA signature [79].

To conduct the attack one needs to collect a set of messages, together with their processing time, e.g. question-answer delay. Many cryptographic algorithms were found to be vulnerable to timing attacks. The main reason why this happens is in the software implementation of each algorithm. That includes performance optimisation to bypass unnecessary branching and conditional operations, cache memory usage, non-fixed time processor instructions such as multiplication and division, and a wide variety of other causes. As a result performance characteristics typically depend on both the encryption key and the input data.

To prevent such attacks the techniques used for blinding signatures can be used [80]. The general idea is to prevent the attacker knowing the input to the modular exponentiation operation by mixing the input with a chosen random value.

Timing attacks can be applied to microcontrollers whose security protection is based on passwords, or to access control systems that use cards or keys with fixed serial numbers, for example, Dallas iButton products [81]. The common mistake in such systems is the way the serial number of the entered key is verified against the database. Very often the system checks each byte of the key against one entry in the database and stops as soon as an incorrect byte is found. Then it switches to the next entry in the database until it reaches the end. So the attacker can easily measure the time between the input of the last key and the request for another key and figure out how many coincidences were found. With a relatively small number of attempts, he will be able to find one of the matching keys.

To prevent these attacks, the designer should carefully calculate the number of CPU cycles that take place when the password is compared and make sure they are the same for correct and incorrect passwords. For example, in the Motorola 68HC08 microcontrollers family the internal ROM bootloader allows access to the Flash memory only if the correct eight-byte password was entered first [82]. To achieve that, extra NOP commands were added to the program making the processing time equal for both correct and incorrect bytes of the password. That gives good protection against timing attacks. Some microcontrollers have an internal RC generator mode of operation in which the CPU running frequency depends upon the power supply voltage and the die temperature. This makes timing analysis more difficult as the attacker has to stabilize the device temperature and reduce any fluctuations and noise on the power supply line. Some smartcards have an internally randomised clock signal to make measurements of the time delays useless for the attack.

4.3 Brute force attacks

'Brute force' has different meanings for cryptography and semiconductor hardware. In cryptography, a brute force attack would be defined as the methodical application of a large set of trials for a key to the system. This is usually done with a computer or an array of FPGAs delivering patterns at high speed and looking for success.

One example could be the password protection scheme used in microcontrollers, such as the Texas Instruments MSP430 family [25]. The password itself is 32 bytes (256 bits) long which is more than enough to withstand direct brute force attack. But the password is allocated at the same memory addresses as the CPU interrupt vectors. That, firstly, reduces the area of search as the vectors always point to even addresses within memory. Secondly, when the software gets updated, only a small part of the password is changed because most of the interrupt subroutines pointed to by the vectors are very likely to stay at the same addresses. As a result, if the attacker knows one of the previous passwords he could easily do a systematic search and find the correct password in a reasonable time.

Brute force can be also applied to a hardware design implemented into an ASIC or a CPLD. In this case the attacker tries to apply all possible logic combinations to the input of the device while observing all its outputs. That kind of attack could be also called black-box analysis because the attacker does not have to know anything about the design of the device under test. He only tries to understand the function of the device by trying all possible combinations of signals. This approach works well only for relatively small logic devices. Another problem the attacker will face is that designs implemented in CPLDs or ASICs have flip-flops, so the output will probably be function of both the previous state and the input. But the search space can be significantly reduced if the signals are observed and analysed beforehand. For example, clock inputs, data buses and some control signals could be easily identified, significantly reducing the area of search.

Another possible brute force attack, applicable to many semiconductor chips, is applying an external high voltage signal (normally twice the power supply) to the chip's pins to find out whether one of them has any transaction like entering into a factory test or programming mode. In fact, such pins can be easily found with a digital multimeter because they do not have a protection diode to the power supply line. Once sensitivity to a high voltage is found for any pin, the attacker can try a systematic search on possible combinations of logic signals applied to other pins to figure out which of them are used for the test/programming mode and exploit this opportunity.

The attack could be also applied to the device communication protocol in order to find any hidden functions embedded by the software developer for testing and upgrade purposes.

Chip manufacturers very often embed hardware test interfaces for postproduction testing of their semiconductor devices. If the security protection for these interfaces is not properly designed, the attacker can exploit it to get access to the on-chip memory. In smartcards such test interfaces are normally located outside the chip circuit and physically removed after the test operation, eliminating any possibility of use by outsiders.

Any security system, either software or hardware, could also have holes in its design and there is always a small chance that an attacker would eventually find one with brute force random testing. Careful design of the security protection, followed by proper evaluation, could help avoid many problems and make such attacks virtually impossible.

4.4 Power analysis

A computing device's power consumption depends on its current activity. The consumption depends on changes of state of its components, rather than on the states themselves, because of the nature of CMOS transistors [83]. When an input voltage is applied to a CMOS inverter, a transient short-circuit is induced. The rise of the current during this transient is much higher than the static dissipation caused by parasitic current leakage. Using a 10–20 Ω resistor in the power supply line, these current fluctuations can be measured. To achieve good results, measurements should be made with at least 12-bit resolution and 50 MHz sampling frequency. Such acquisition parameters allow us to distinguish between different CPU instructions and estimate the number of bus bits changing at a time.

By averaging the current measurements of many repeated identical operations, even smaller signals that are not transmitted over the bus can be identified. Signals such as carry-bit states are of special interest, because many cryptographic key-scheduling algorithms use shift operations that single out individual key bits in the carry flag. Even if the status-bit changes cannot be measured directly, they often cause changes in the instruction sequence or microcode execution, which then cause a clear change in the power consumption.

The various instructions cause different levels of activity in the instruction decoder and arithmetic units, and can often be quite clearly distinguished so that parts of algorithms can be reconstructed. Various units of the processor have their switching transients at different times relative to the clock edges, and can be separated in high-frequency measurements.

There are many publications on different power analysis techniques that can be used to break many cryptographic algorithms [84][85][86]. The whole process of analysis is relatively easy to implement, and only requires standard off-the-shelf measurement equipment costing a few thousand pounds.

There are two major power analysis techniques – simple power analysis (SPA) and differential power analysis (DPA). SPA involves direct observation of the power consumption during cryptographic or other security sensitive operations. SPA can reveal information about the device's operation as well as the key material. If the attacker knows the cryptographic algorithm (and especially its implementation in the tested device) he can easily work out some bits of information by observing the sequences of CPU instructions, especially rotation and conditional branches. If the result of an arithmetic or logic operation can be observed as well, i.e. the state of carry, zero or negative flags, more information can be obtained. DPA is a more powerful technique, because the attacker does not have to know as many details about how the cryptographic algorithm was implemented. It uses statistical analysis to extract hidden information from a large sample of power traces obtained during cryptographic computations with known ciphertexts. The statistical methods identify small differences in power consumption which can be used to recover individual bits in a secret key.

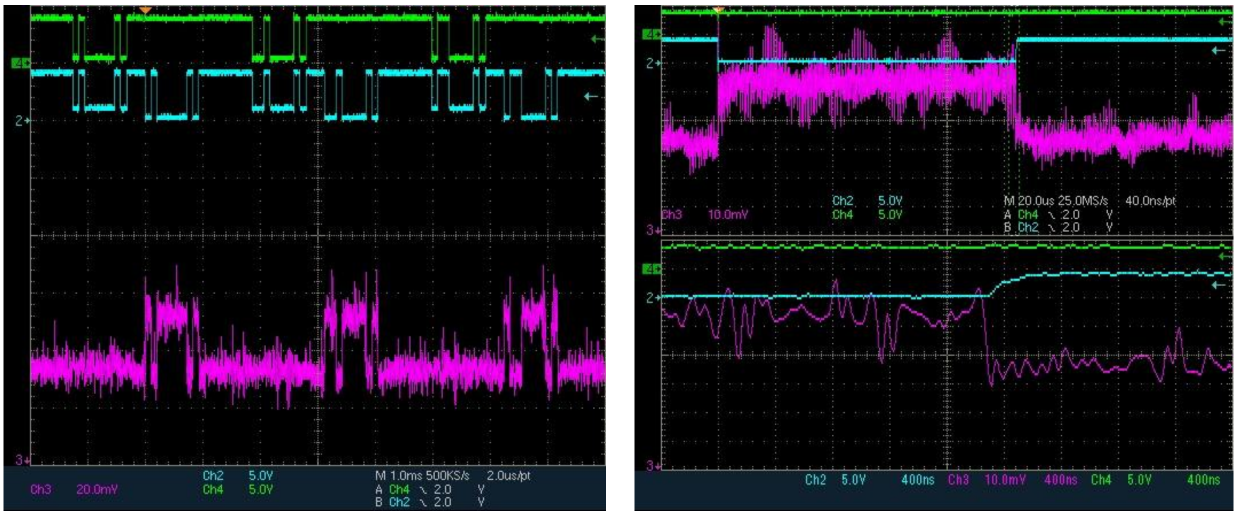


Figure 36. Power trace acquired with a passive probe, 500KS/s and 25MS/s rates

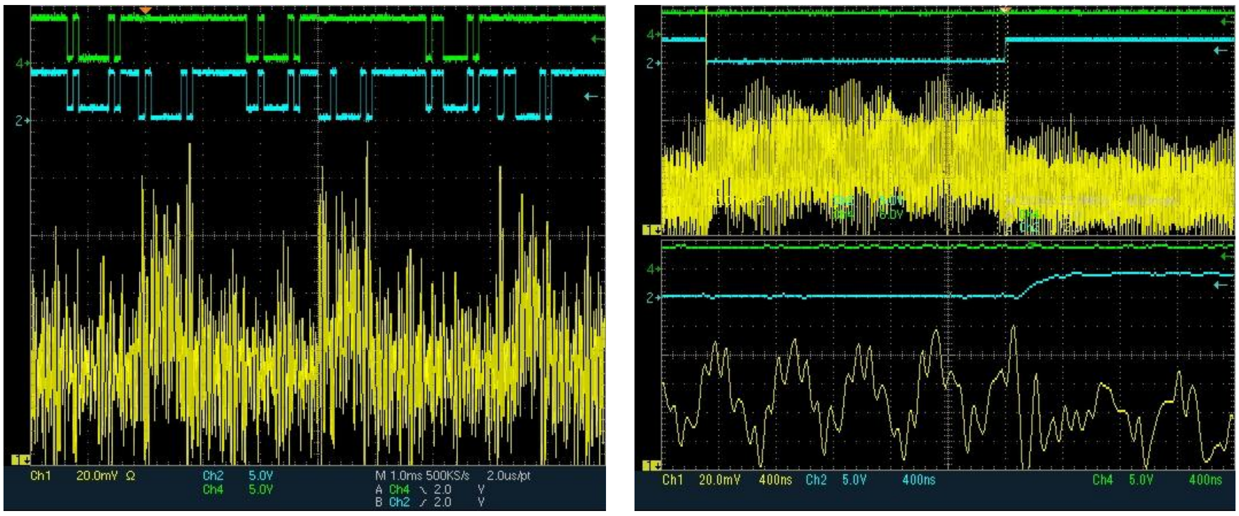


Figure 37. Power trace acquired with an active probe, 500KS/s and 25MS/s rates

Power consumption characteristics always include noise components. The external noise can be reduced by proper design of the signal acquisition path and careful use of the measurement equipment. Measuring the power consumption on the resistor in the ground line has some advantages. Firstly, it reduces the noise level and, secondly, it allows us to measure the signal directly with an oscilloscope probe, because most probes have their common line permanently connected to the main power ground. To increase the signal-to-noise ratio further, the number of averaged samples can be increased.

Figure 36 shows an example of the power trace acquired from Motorola MC68HC908JB8 microcontroller [87] in bootloader mode running at 6 MHz with a standard passive probe (8 pF, 1 KΩ @ 10 MHz). An active probe (1 pF, 10 KΩ @ 10 MHz) helps us reduce the input capacitance and thus increase the bandwidth of the acquired signal (Figure 37). A low-cost alternative to such a probe can be relatively easily built from a high-speed low-noise operational amplifier, available from a local electronic components store for a few pounds. Another possibility is to use a very short coaxial cable connected directly to the oscilloscope input. In this case the input capacitance of the probe is significantly reduced, but the measurements could be inaccurate as modern oscilloscopes use probes with built-in attenuators, automatic detection and self-calibration.

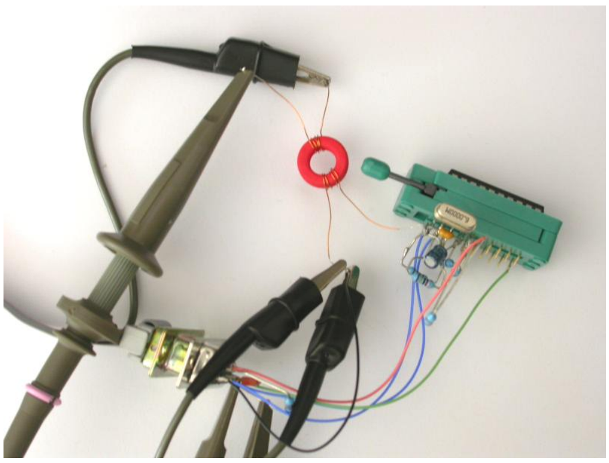


Figure 38. Experiment setup for measuring power consumption via a ferrite transformer

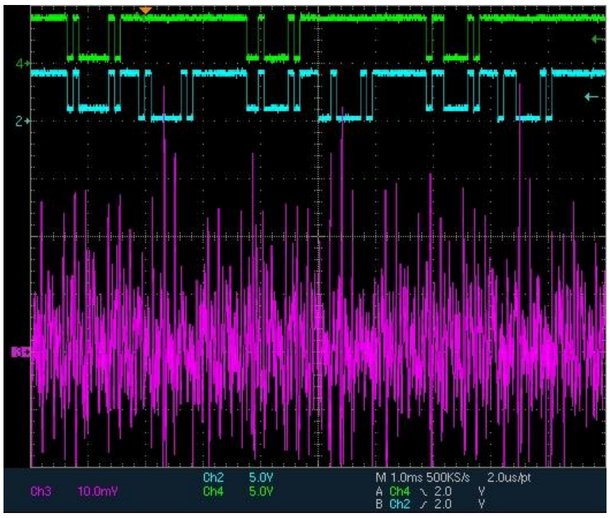


Figure 39. Power trace acquired via a ferrite transformer with a passive probe

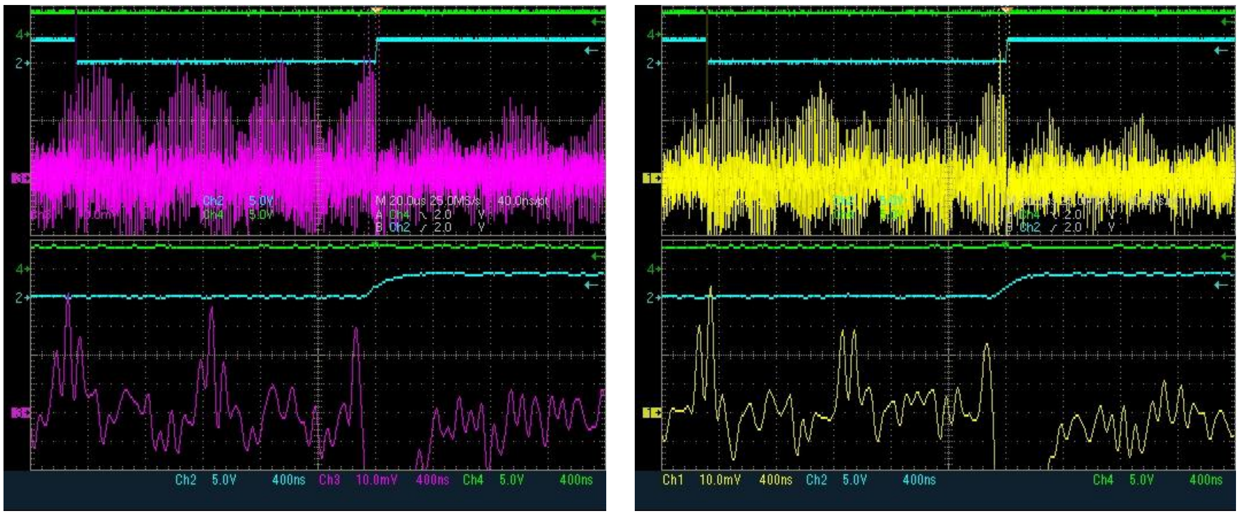


Figure 40. Power trace acquired via transformer with passive (left) and active (right) probes at 25MS/s rate

We made some improvements to the existing power analysis setup. This is a new approach and we have not seen any reference to it before. Instead of using a resistor in the power or ground line we used a ferrite core transformer (Figure 38). That brought some changes to the waveform because the DC component of the signal was lost (Figure 39). At the same time it has some advantages. There is almost no limit on DC current flow where with a 10 Ω resistor a transient increase in the consumption current to 100 mA will cause a 1 V drop, which could disrupt the normal operation of the device. Reducing the resistor value will solve the problem but make it harder to recognise small changes in the power consumption, as needed to perform reliable analysis. With the transformer, there is no need to use an expensive active probe, as the standard passive probe gives almost the same result (Figure 40). If the signal is too small, extra turns in the secondary coil will increase the amplitude. Also the transformer acts as a passive filter itself. As it can be seen from the waveforms in Figures 37 and 40, the same CPU instructions have different influence on the waveform for resister and transformer measurements. That can be used as a form of post-processing of the acquired signal.

To perform a successful attack, thousands of samples may have to be acquired, following which processing and analysis reveal the full secret or private key quickly.

Recently, progress in chip design prevents such attacks or at least makes them more difficult to apply. Some of the protection measures being adopted are discussed in Chapter 8.

4.5 Glitch attacks

Glitch attacks are fast changes in the signals supplied to the device and designed to affect its normal operation. Usually glitches are inserted in power supply and clock signals, but a glitch could be an external electric field transient or an electro-magnetic pulse. One approach was suggested in [8]; two metal needles might be placed on a smartcard within a few hundred micrometers away from the chip surface. Then by applying a spike of a few hundred volts for less than a microsecond on these needles, an electric field in the silicon substrate of sufficient strength to temporarily shift the threshold voltages of nearby transistors will be induced. One modification of the above proposal was suggested recently: using a miniature inductor consisting of several hundred turns of fine wire around the tip of a microprobe needle. A current injected into this coil will create a magnetic field, and the needle will concentrate the field lines [88][89].

Every transistor and its connection paths acts like an RC element with a characteristic time delay. The maximum usable clock frequency of a processor is determined by the maximum delay among its elements. Similarly, every flip-flop has a characteristic time window (of a few picoseconds) during which it samples its input voltage and changes its output accordingly. This window can be anywhere inside the specified setup cycle of the flip-flop, but is quite fixed for an individual device at a given voltage and temperature. So if we apply a clock glitch (a clock pulse much shorter than normal) or a power glitch (a rapid transient in supply voltage) this will affect only some transistors in the chip and cause one or more flip-flops to adopt the wrong state. By varying the parameters, the CPU can be made to execute a number of completely different wrong instructions, sometimes including instructions that are not even supported by the microcode. Although we do not know in advance which glitch will cause which wrong instruction in which chip, it can be fairly simple to conduct a systematic search.

4.5.1 Clock glitches

Clock-signal glitches are currently the simplest and most practical ones. In real application glitches are normally used to replace conditional jump instructions and test instructions preceding them. They create a window of vulnerability in the processing stages of many security cryptographic barriers by simply preventing the execution of the code that detects an unsuccessful authentication attempt. Instruction glitches can also be used to extend the runtime of loops, for example, in serial port output routines to see more of the memory after the output buffer, or to reduce the number of loops in cryptographic operation to transform the cipher into a weak one.

To perform a glitch, the clock frequency should be temporarily increased for one or more half cycles so that some flip-flops sample their input before the new state has reached them. As clock glitches are normally aimed at CPU instruction flow, they are not very effective for devices with hardware implementations of security protection. Therefore it is practical to use clock glitches only when attacking microcontrollers with software programming interfaces or some smartcards.

```
        LDA    #01h
        AND    $0100            ;the contents of the first byte of EEPROM is checked
loop:   BEQ    loop             ;endless loop if bit 0 is zero
        BRCLR 4, $0003, cont    ;test mode of operation
        JMP    $0000            ;direct jump to the preset address
cont:   LDA    #C0h
        STA    $000D            ;initialize the serial asynchronous port
        CLR    $000E
        BSET   2, $000F
        LDX    #50h
wait:   BRCLR 5, $0010, wait    ;upload user code
        LDA    $0011
        STA    , x
        INCX
        DEC    $0050
        BNE    wait
        JMP    $0051            ;jump to the user code
```

Figure 41. Example of the bootloader code responsible for security in MC68HC05B6 microcontroller

For example, the Motorola MC68HC05B6 microcontroller [90] has a Mask ROM bootloader which prevents user code upload if the security bit is set. The part of the code responsible for the security is presented in Figure 41. It checks the contents of the first byte in the EEPROM and if the bit 0, assigned as a security fuse, is programmed then the CPU goes into endless loop. That sort of protection could be relatively easy defeated. As the CPU performs only one instruction in the loop, all the attacker has to do is apply different clock glitches to cause CPU malfunction. He does not even have to carefully synchronise the attack to the CPU clock signal, as doing glitches at a random time will give a success in a short number of attempts. Glitches could be inserted relatively easy without the use of any external generators by short circuiting the crystal resonator for a short time. When the resonator starts it produces oscillations at different harmonics which cause many glitches. In most cases the attack has to be applied at a certain clock cycle to cause the desired result. In this case it is better to use either a signal pattern generator which can supply all the necessary signals to the chip or built such a generator using an FPGA prototyping board.

Applying clock glitches to some microcontrollers could be difficult. For example, the Texas Instruments MSP430 microcontroller family operates from an internal RC generator in bootloader mode and it is difficult to synchronise to the internal clock and estimate the exact time of the attack. Some smartcards benefit from having randomly inserted delays in the CPU instruction flow, which makes applying the attacks even more difficult. Using power analysis could help, but requires very sophisticated and expensive equipment to extract the reference signal in real time.

4.5.2 Power glitches

Power supply voltage fluctuations can shift the threshold level of the transistors. As a result some flip-flops will sample their input at different time or the state of the security fuse will be read incorrectly. This is usually achieved by either increasing the power supply voltage or dropping it for a short period of time, normally from one to ten clock cycles. Power glitches can be applied to a microcontroller with any programming interface as they could affect both the CPU operation and the hardware security circuit. In general, they are harder to find and exploit than clock glitches because in addition to the timing parameters, the amplitude and rising/falling times are variables.

One example is the attack on the MC68HC05B6 microcontroller discussed above. If the power supply voltage is reduced by 50–70% for the period of time that the “AND \$0100” instruction is executed, the CPU fetches an FFh value from the EEPROM memory rather than the actual value and this corresponds to the unsecured state of the fuse. The trick is to carefully calculate the exact time to reduce the supply voltage, otherwise the CPU will stop functioning or go into the reset mode. This is not a difficult task, as the target instruction is executed within the first hundred cycles after the reset. Again, the attacker could use a pattern generator or build his own glitch device.

Another example is an old PIC16F84 microcontroller from Microchip [91]. The chip erase operation removes the security protection but at the same time erases the contents of program and data memories on the chip. The hardware design of the security protection circuit is made such that the memory is always erased before the security fuse is reset to the initial state. However it was found that if during the chip erase operation the power supply voltage is increased to about 10 V for a few milliseconds it causes the memory erase process to terminate but the security fuse reset finishes as usual making it possible to read the contents of the memory. Such a high voltage pulse should be applied carefully as increasing its length could permanently damage the chip. The later revision of this microcontroller, PIC16F84A [92], has protection against under- and over-voltage attacks. Any memory modification operations performed via the programming interface are immediately terminated if the power supply voltage goes below 3 V or above 6 V.

It is not always necessary for the power glitches to be outside the specified power supply voltage range. For example, in the same PIC16F84A microcontroller the protection mechanism can be defeated by applying a mere 50 mV glitch after the chip erase operation has started. That causes termination of the program memory erase operation but not the fuse erase.

All the above examples of glitch attacks show how powerful such attacks can be unless special countermeasures are implemented. These could be voltage and clock monitor circuits which reset the CPU if the voltage or clock frequency go out of range. Clock-monitoring circuits are normally used in smartcards but very few microcontrollers have them.

4.6 Data remanence

Security processors typically store secret key material in Static RAM, from which power is removed if the device is tampered with. It is widely known that, at temperatures below −20°C, the contents of SRAM can be 'frozen'; therefore, many devices treat temperatures below this threshold as tampering events. We have done some experiments to establish the temperature dependency of data retention time in modern SRAM devices. Our experiments show that the conventional wisdom no longer holds and that data remanence can be a problem even at higher temperatures [54].

Data remanence affects not only SRAM but other memory types as well, like DRAM, UV EPROM, EEPROM and Flash [50]. As a result, some information still can be extracted from memory that has been erased. This could create many problems for secure devices which assume that all the sensitive information is gone once the memory is erased.

4.6.1 Low temperature data remanence in SRAM

Security engineers are interested in the period of time for which an SRAM device will retain data once the power has been removed. The reason for this is as follows. Many products do cryptographic and other security-related computations using secret keys or other variables that the equipment's operator must not be able to read out or alter. The usual solution is for the secret data to be kept in volatile memory inside a tamper-sensing enclosure. On detection of a tampering event, the volatile memory chips are powered down or even shorted to ground. If the data retention time exceeds the time required by an opponent to open the device and power up the memory, then the protection mechanisms can be defeated [53][93][94].

In the 1980s, it was realised that low temperatures can increase the data retention time of SRAM to many seconds or even minutes. With the devices available at that time, it was found that increased data retention started about −20°C and increased as temperature fell further [93]. Some devices are therefore designed with temperature sensors; any drop below −20°C is treated as a tampering event and results in immediate memory zeroisation [95][96]. We set out to repeat this work. Our goal was to find whether the memory devices available in the year 2000 exhibit the same behaviour.

Another important thing to keep in mind is that security information could be restored even if part of the memory is corrupted. Suppose an attacker has correctly restored only m = 115 bits of an n = 128 bits long secure key, or 90% of the information. Then he will have to search through $n!/(m!(n{-}m)!) = 128!/(115!13!) = 2.12{\cdot}10^{17} \sim 2^{58}$ possible keys. Having 10,000 computers, each performing 1 billion key-search operations per second, the attacker will spend only 6 hours to search through all possible keys. If only 80% of information or 103 bits of a 128-bit secure key are known, than an attacker will need $2.51{\cdot}10^{26} \sim 2^{88}$ tries. Having even 100 times the capability, the attacker will spend more than a million years searching for the key. So to be sure that symmetric 128-bit keys cannot be retrieved from memory, it should be left without power for the time necessary to corrupt 20% or more of the cells. If error correction for key data is used, this value should be increased correspondingly. In our experiments, we assumed that no error correction was used.

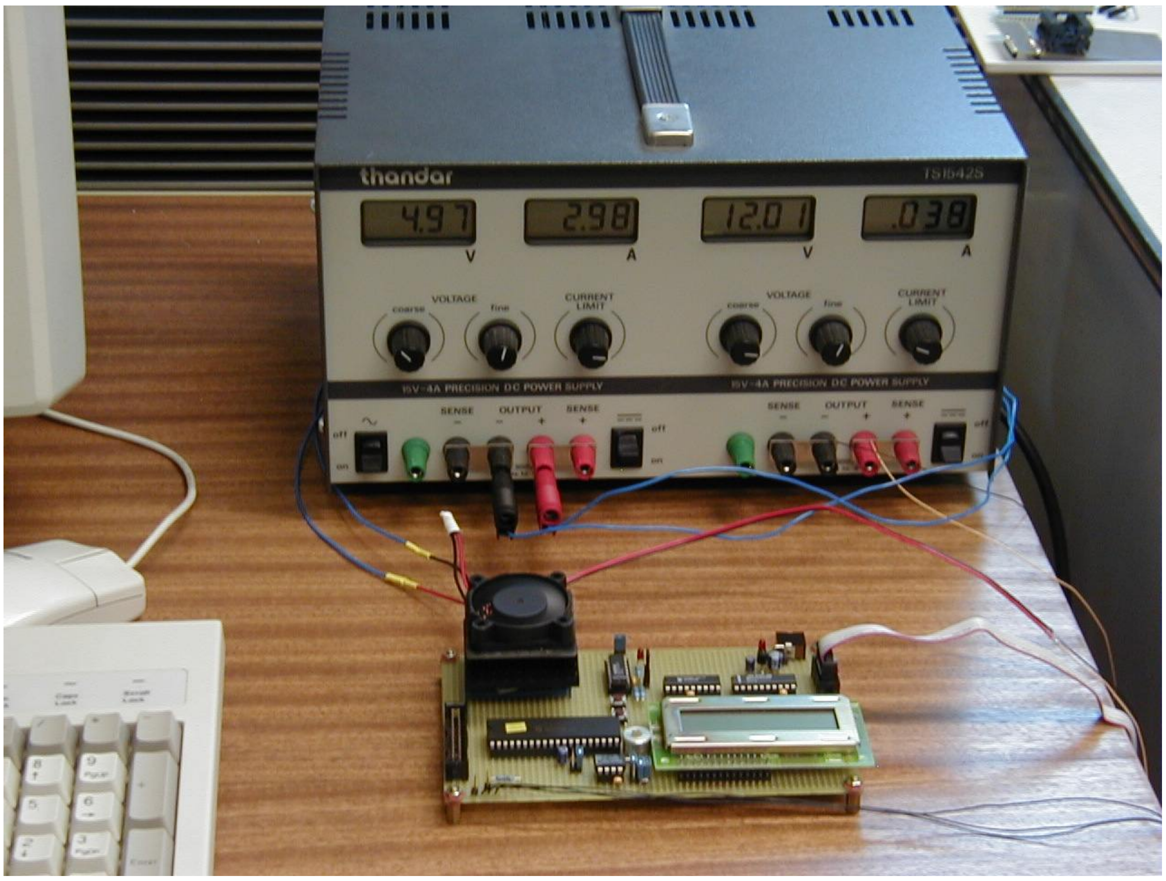


Figure 42. View on the test board

We built a special circuit board for testing SRAM chips. All signals were controlled by a PIC16F877 microcontroller working at 4 MHz, which was connected via an RS-232 interface to a computer for programming the necessary modes and downloading information (Figure 42). The power supply line of the SRAM chip was controlled by a CMOS switch (MAX314). We also had an LCD display and two buttons for hand controlling the experiments. For convenient insertion and extraction of SRAM chips, we put a lock/eject socket on to the board. Also, we put an external connector for testing SRAM chips inside a freezer. In this case, we used a flat cable with an IC socket at the end.

For temperature control, we used an LM135H temperature sensor, which operates from −55°C to +150°C with ±1°C precision, and provides an output voltage directly proportional to the absolute temperature at +10 mV/K. For temperature monitoring, we used a standard digital multimeter.

For temperatures from +25°C down to 0°C, we used Peltier elements. For lower temperatures, we used a domestic freezer in conjunction with Peltier elements.

Each SRAM chip was tested under two conditions – with the power supply pin shorted to the ground after power-off, and with it left floating. Each SRAM chip was tested with all 0's and all 1's test patterns and the number of bits flipped after the test was counted by downloading the memory contents afterwards.

Eight different SRAM samples were tested at different temperatures. All SRAM samples were bought from a semiconductor distributor (Farnell). The list of the SRAM chips we tested included Dallas DS2064-200, GoldStar GM76C88AL-15, Hyundai HY6264AP-10LL and HY62256BLP-70, NEC D4364C-15 and D4364C-15L, Samsung K6T0808C1D-DB70, Toshiba TC5564APL-15.

We also measured the power supply current, in non-active mode, for all SRAM samples at room temperature. Because this current is very small, it is not possible to measure it directly with a digital multimeter. To measure this current, we built a circuit board with a MAX4374H current sense amplifier (100×) and a socket for the SRAM chip. As a sensor we used a 10 kΩ resistor, so the output voltage on the MAX4374H corresponds to the power supply current with a ratio 1 mV per 1 nA. The results of these measurements are represented in Table 2. We defined the data retention time to be the time during which at least 80% of the memory contents are preserved.

Sample	TC5564	DS2064	K6T0808	D4364CL	HY6264	HY62256	GM76C88	D4364C
Current nA	1	2	9	319	357	384	1375	1697
Ret. Time (shorted), ms	3520	2315	1365	65	34	65	20	12
Ret. Time (floating), ms	13100	12200	4610	206	67	206	63	37

Table 2. Power consumption and data retention time of different SRAM chips at a room temperature.

An important observation is that the smaller the power consumption of the chip, the longer is its data retention time. We suspect that this will hold even where chips come from the same batch.

With the power supply pin connected to ground, the data retention time is always less than if the power supply pin is left floating. Once information loss from an SRAM chip begins, it proceeds quickly.

Comparing the two SRAM chips NEC4364C-15 and NEC4364C-15L (the last one is a low power version) we can note that the low power version has a longer retention time at any temperature. The reverse situation holds with the HY6264A-10LL and HY62256BL-70. The first one is an ultra low power version, but, although the second one is the low power version, it has a longer data retention time, because it was produced later and it was designed using smaller transistors. Thus it consumes less power.

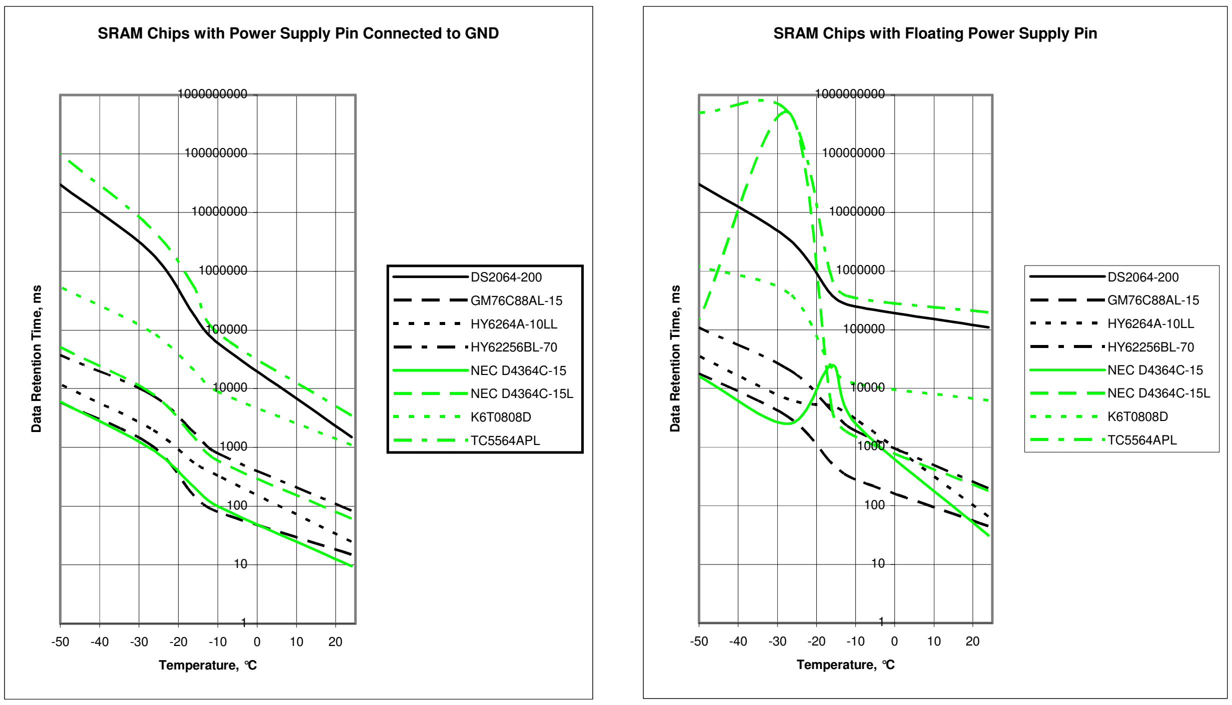


Figure 43. Dependence of data retention time from temperature

We tested and documented the data retention characteristics of a sample of modern SRAM chips, as a function of temperature (Figure 43). Contrary to the established wisdom, there are several chips that retain data for dangerous periods of time at temperatures above −20°C. The temperature at which 80% of the data are retained for one minute varies widely between devices. Some require cooling to at least −50°C, while others retain data for this period at room temperature. Retention times can be significantly reduced by shorting V_{CC} to ground rather than by leaving it floating. Another unexpected observation is that memory retention time varies not just from one device type to another, but also between devices from the same manufacturer and of the same type but of different subtype or series. Presumably this is because chip makers do not control data retention time as part of their manufacturing quality process. Low power versions of the same chip always seem to have longer retention times provided they are implemented in the same process generation. Thus, to build secure processors that reliably erase memory on tampering, it would appear to be vital to test chip samples before use. As this is time-consuming, it will not usually be feasible for each individual device. However, measuring the power consumption of each chip in a batch can give a useful and practical test of inter-device variability.

4.6.2 Data remanence in non-volatile memories

Unlike SRAM which has only two stable logic states, EPROM, EEPROM and Flash cells actually store analog values in the form of a charge on the floating gate of a MOS transistor. The floating-gate charge shifts the threshold voltage of the cell transistor and this is detected with a sense amplifier when the cell is read. The maximum charge the floating gate can accumulate varies from one technology to another and normally is between 10^3 and 10^5 electrons. For standard 5 V EEPROM cell, programming causes about a 3.5 V shift in the threshold level. Some modern Flash memory devices employ multiple level detection, thus increasing the capacity of the memory [97]. There are also memory devices with full analog design which store charge proportional to the input voltage [98].

There are two basic processes that allow placing the electrons on the floating gate – Fowler-Nordheim tunnelling and channel hot electron (CHE) injection [47]. Both processes are destructive to the very thin dielectric between the floating gate and the channel of a transistor. As a result, the number of possible cycles is limited because the floating gate slowly accumulates electrons, causing a gradual increase in the storage transistor's threshold voltage and programming time. After a certain amount of program/erase cycles (typical values were represented in Table 1) it is no longer possible to erase or program the cell. Another negative effect (which is the main failure mode for Flash memory) is negative charge trapping in the gate oxide. It inhibits CHE injection and tunnelling, changes the write and erase times of the cell, and shifts its threshold voltage.

The amount of trapped charge can be detected by measuring the gate-induced drain leakage current of the cell, or its effect can be observed indirectly by measuring the threshold voltage of the cell. In older devices, which had the reference voltage for the sense amplifier tied to the device supply voltage, it was often possible to do this by varying the device supply voltage. In newer devices, it is necessary to change the parameters of the reference cell used in the read process, either by re-wiring portions of the cell circuitry or by using undocumented test modes built into the device by manufacturers.

Another phenomenon which helps with this is overerasing. If the erase cycle is applied to an already-erased cell, it leaves the floating gate positively charged, thus turning the memory transistor into a depletion-mode transistor. To avoid this problem, some devices, for example Intel's original ETOX devices [99], first program all cells to 0's before erasing them to 1's. In later devices this problem was solved by redesigning the cell to avoid excessive overerasing, however even with this protection there is still a noticeable threshold shift when a virgin cell is programmed and erased.

The changes in the cell threshold voltage caused by write/erase cycles are particularly apparent in virgin and freshly-programmed cells. It is possible to differentiate between programmed-and-erased and never-programmed cells, especially if the cells have only been programmed and erased once, since virgin cell characteristics will differ from the erased cell characteristics. The changes become less noticeable after ten program/erase cycles.

Programmed floating-gate memories cannot store information forever. Various processes (such as field-assisted electron emission and ionic contamination) cause the floating gate to lose the charge, and go faster at higher temperatures. Another failure mode in the very thin tunnel oxides used in Flash memories is programming disturb, where unselected erased cells adjacent to selected cells gain charge when the selected cell is written. This is not enough to change the cell threshold sufficiently to upset a normal read operation, but could cause problems to the data retention time and should be considered during measurement of the threshold voltage of the cells for data analysis and information recovery. Typical guaranteed data retention time for EPROM, EEPROM and Flash memories are 10, 40 and 100 years respectively.

Obviously, in a floating gate memory cell, the floating gate itself cannot be accessed. Its voltage is controlled through capacitive coupling with the external nodes of the device. Often, the floating-gate transistor is modelled by a capacitor equivalent circuit called the capacitor model [100]. In practice, write/erase characteristics for many EEPROM/Flash memories are close to that of a charge/discharge of a capacitor. Meanwhile there are some differences in how the charge/discharge process takes place in real memory cells. There is an initial delay between the time the voltages are applied to the cell, and the charge starting to be removed or injected. This delay is caused by the need for very high electric fields to be created inside the floating-gate transistor to start the injection or tunnelling process. Some EEPROM cells have been reported to have nonuniformity during the erase operation [101]. As a result, it might take longer to erase a half-charged cell than a fully-charged cell. In addition, an ideal capacitor discharges exponentially: $q = q_0\, e^{-t/\tau}$. Applied to the floating gate, that would mean that after $t = 10\,\tau$ the charge is totally removed from the cell. In practice this does not happen, because the parameters of the cell's transistor change as the charge is removed from its floating gate. All the above-mentioned problems could seriously affect data remanence in floating-gate memories.

The main difficulty with analysis of the floating-gate memory devices, especially EEPROM and Flash, is the variety of different designs and implementations from many semiconductor manufacturers. There are hundreds of different types of floating-gate transistor, each with its own characteristics and peculiarities. It means that for security applications where data remanence could cause problems, careful testing should be applied to the specific non-volatile memory device used in the system.

We undertook the evaluation of some microcontrollers with different memory types to investigate the possible influence of data remanence on EPROM, EEPROM and Flash memories. For that purpose we built a special test board controlled by a PC via a parallel interface (Figure 44). The board has two programmable power supplies for generating V_{DD} and V_{PP} voltages, a programming interface with bidirectional voltage level converters, and sockets for microcontroller chips. That allowed us to control the voltages applied to the chip under test with 100 µV precision and apply any signals within a 1 µs time frame.

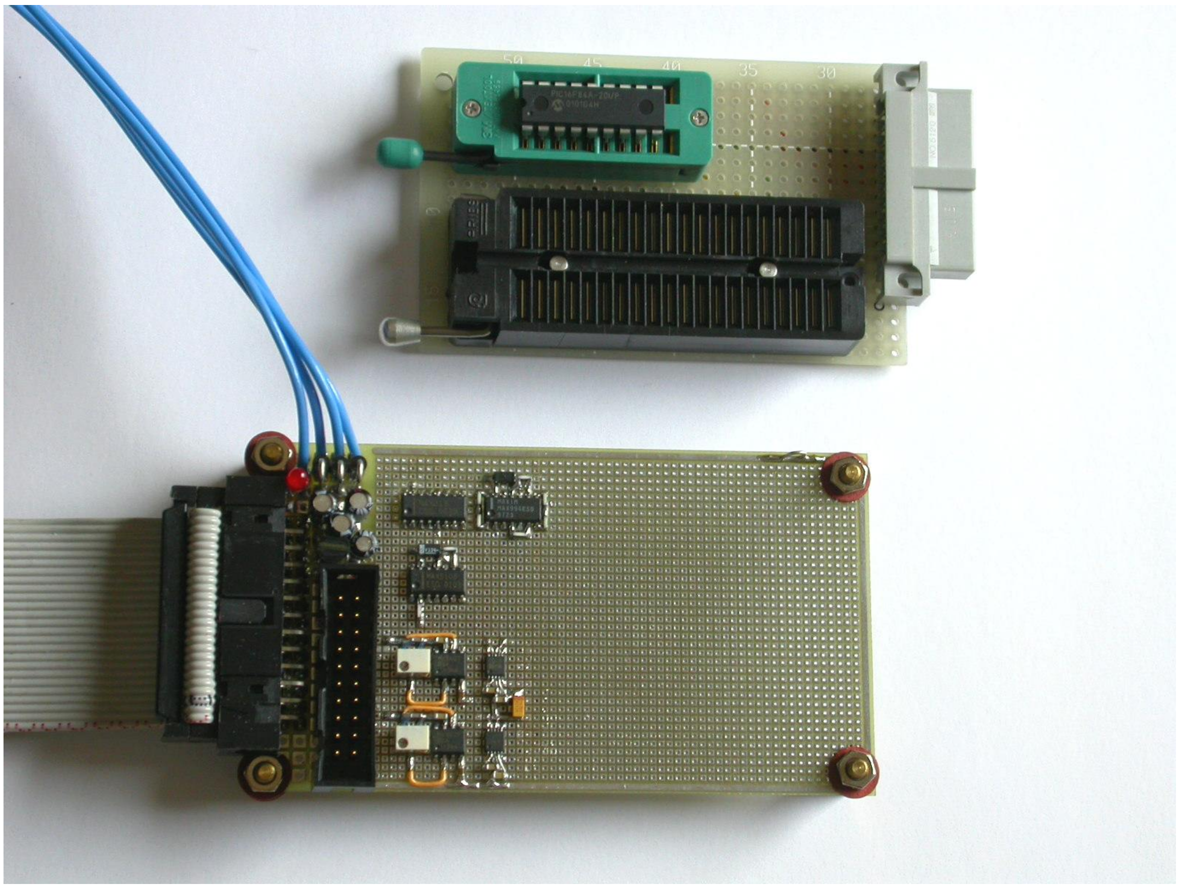


Figure 44. The test board for data remanence evaluation in microcontrollers

The first experiment was performed on the Microchip PIC12C509 microcontroller with UV EPROM [22]. The chip was programmed with all 0's and exposed to UV light for different periods of time. Then it was read in the test board at different power supply voltages to estimate the threshold level for each EPROM cell in the memory array. We assumed that the reference voltage is tied to the power supply line and therefore the threshold level of the transistor $V_{TH} = K\, V_{DD}$, where K is usually close to 0.5. The fact that we do not measure the exact threshold voltage of the transistor does not affect our results because we are interested in the relative erase timing between the memory and the security fuse. The same test was applied to a chip with a programmed security fuse. The results are presented in Figure 45. As can be seen from the graph, the memory gets fully erased before the security fuse is erased. However some security flaws still could exist. Although nothing could be extracted directly by reading the memory when the fuse is erased, power glitch tricks could work. For example, after seven minutes of exposure to the UV light (253 nm peak, 12 mW/cm^2) the memory content can be read non-corrupted at V_{DD} below 2.2 V, but the security fuse remains active up to 4.8 V. If the attacker works out the exact time when the data from memory is latched into the output shift register and the time when the state of the security fuse is checked, he might be able to extract the memory contents by reducing the power supply down to 2 V for the data latching and increasing it to 5 V to make the security fuse inactive.

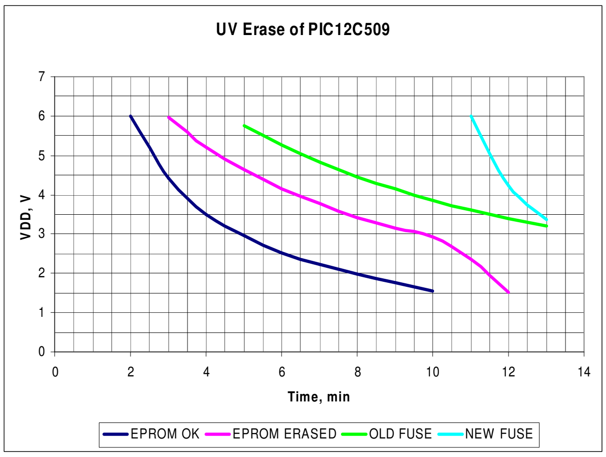


Figure 45. Memory contents of PIC12C509 tested at different power supply voltages after UV erase

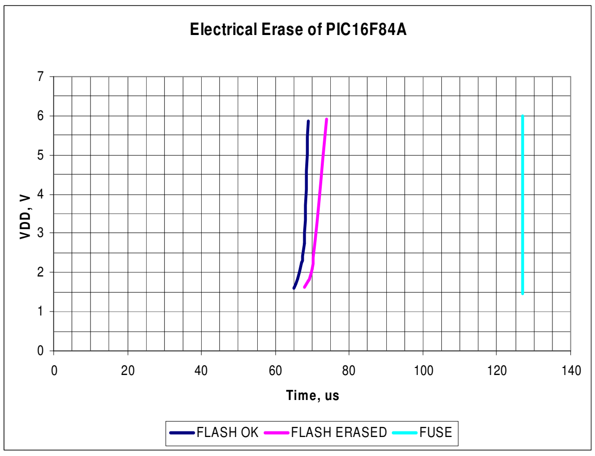


Figure 46. Memory contents of PIC16F84A tested at different power supply voltages after electrical erase

There is another trick that makes recovery of memory contents possible, even when there is no overlap between the erased security fuse and non-corrupted memory content at the time of erasure. For example, we found that newer samples of the same chip will start to corrupt the memory before the security fuse is erased (Figure 45). In this case a power glitch cannot be used to recover information from the memory. What can be done instead is a careful adjustment of the threshold voltage in the cell's transistor. It is possible to inject a certain portion of charge into the floating gate by carefully controlling the memory programming time. Normally, the programming of an EPROM memory is controlled by external signals and all the timings should be supplied by a programmer unit. This gives an opportunity for the attacker to inject charge into the floating gate thus shifting the threshold level enough to read the memory contents when the security fuse is inactive. Such a trick is virtually impossible to apply to modern EEPROM and Flash memory devices for several reasons. First, the programming is fully controlled by the on-chip hardware circuit. Second, the programming of EEPROM and Flash cells is normally performed by using much faster Fowler-Nordheim tunnelling rather than CHE injection. As a result it is very hard to control the exact amount of charge being placed into the cell. Also, the temperature and the supply voltage affect this process making it even harder to control.

Our next experiment was done to the PIC16F84A microcontroller which has Flash program memory and EEPROM data memory. A similar test sequence was applied with the only difference that electrical erasing was used (Figure 46). A huge difference in the memory behaviour can be observed. The memory erase starts 65 µs after the 'chip erase' command was received and by 75 µs the memory is erased. However, this time changes if the temperature or the supply voltage is changed. For example, if the chip is heated to 35°C the memory erase starts at 60 µs and is finished by 70 µs. The security fuse requires at least 125 µs to be erased giving at least five times excess for reliable memory erase. Reducing the power supply voltage increases the erase time for both the memory and the fuse erase, so that the ratio remains practically the same. It should be mentioned that unless terminated by the hardware reset, the chip erase operation lasts for at least 1 ms. Both this fact and the fast erase time give an impression that EEPROM and Flash memories have fewer problems with data remanence and therefore should offer better security protection. We decided to investigate whether this is true or not.

In our early experiments with the security protection in PIC microcontrollers, I noticed that the same PIC16F84 chip behaves differently if it is tested right after the erase operation was completed. As this microcontroller is no longer in use and has been replaced by the PIC16F84A, the testing was applied to the new chip.

We performed an experiment to estimate how much information could be extracted from the PIC16F84A chip after a normal erase operation was applied to it. As can be seen from Figure 46 the memory is completely erased and read as all 1's well before the end of the standard 10 ms erase cycle. The threshold of the cell's transistors becomes very low after the erase and cannot be measured the same way as with UV EPROM because the chip stops functioning if the power supply drops below 1.5 V. With the power glitch technique, it is possible to reduce the supply voltage down to 1 V for a short period of time – enough for the information from memory to be read and latched into the internal buffer. But this is still not enough to shift the reference voltage of the sense amplifier low enough to detect the threshold of the erased cells. To achieve the result another trick was used in addition to the power glitch. The threshold voltage of all the floating gate transistors inside the memory array was shifted temporarily by $V_W = 0.6$–0.9 V, so that $V_{TH} = K\,V_{DD} - V_W$. As a result it became possible to measure the threshold voltage of an erased cell which is close to 0 V. This was achieved by precisely controlling the memory erase operation, thus allowing the substrate and control gates to be precharged and terminating the process before the tunnelling is started. As a result, the excess charge is trapped in the substrate below the floating gate, and shifts the threshold of the transistor. The process of recombination of the trapped excess charge could take up to one second, which is enough to read the whole memory from the device. This can be repeated for different supply voltages combined with power glitches, in order to estimate the threshold of all the transistors in the memory array.

Applying the above test to differently programmed and erased chips we were able to build the diagrams for threshold voltage dependence in the Flash program memory from different factors such as the number of erased cycles (Figure 47) and memory address (Figure 48). As can be seen, the charge is not entirely removed from the floating gate even after one hundred erase cycles thus making it possible for the information to be extracted from the memory. This was measured on a sample after 100 program/erase cycles to eliminate the effect of the threshold shift taking place in a virgin cell. At the same time the memory analysis and extraction is complicated by the fact that the difference in threshold voltages between the memory cells is larger than within the cell itself. The practical way to avoid this problem is to use the same cell as a reference and compare the measured threshold level with itself after the extra erase operation is applied to the chip. Very similar results were received for the EEPROM data memory inside the same PIC16F84A chip. The only difference was that the threshold voltage after ten erase cycles was very close to that of the fully erased cell, thus making it almost impossible to recover the information if the erase operation was applied more than ten times.

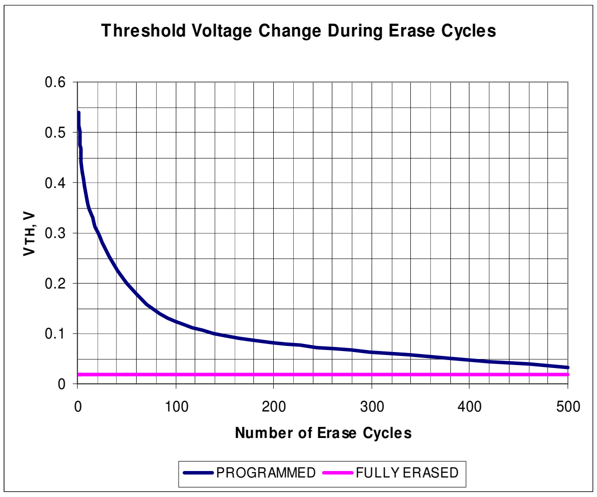


Figure 47. Change of the threshold voltage during erase for programmed and previously erased cells in PIC16F84A

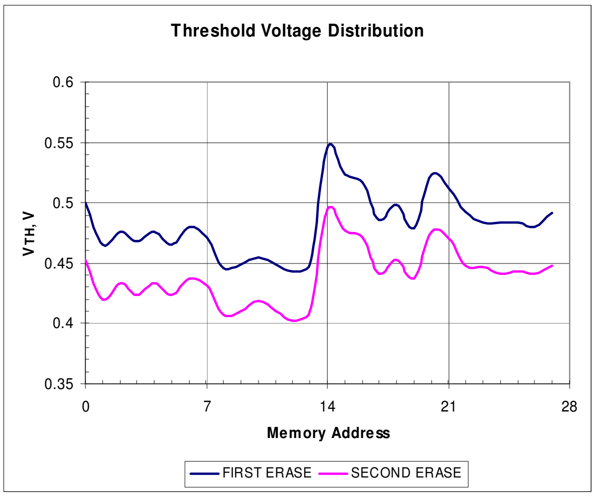


Figure 48. Change of the threshold voltage of previously programmed cells after the second erase cycle was applied to the memory in PIC16F84A

In our next test we programmed the chip with all 0's before applying the erase operation. As a result we were unable to distinguish between previously programmed and non-programmed cells. That means that pre-programming the cells before the erase operation could be a reasonably good solution to increase the security of the on-chip memory.

Fortunately, the extraction from the erased memory technique can only be applied to a very limited number of chips with EEPROM/Flash memory. First, some microcontrollers such as in the Texas Instruments MSP430 family of microcontrollers [102] have an internally stabilized supply voltage for the on-chip memory. Changing the power supply from 1.8 V to 3.6 V does not affect memory read operation from partially erased cells. Second, most microcontrollers fully reset and discharge the memory control circuit if the chip is reset or the programming mode is re-entered. But still, if the memory contents do not disappear completely, this represents a very serious threat to any security based on an assumption that the information is irrecoverable after one memory erase cycle. Even if non-invasive methods do not work, invasive methods could help. For example, the memory control circuit can be modified under a FIB to directly access the reference voltage, the current source or the control gate voltage. Finally, some chips program all the memory locations before applying the erase operation. This makes it almost impossible to extract any useful information from the erased memory.

One more thing should be mentioned in connection with hardware security. Some microcontrollers have an incorrectly designed security protection fuse, which gets erased earlier than the memory. As a result, if the chip erase operation is terminated prematurely, information could be read from the on-chip memory in a normal way. That was the case, for example, for the Atmel AT89C51 microcontroller. When this became known in the late nineties, Atmel redesigned the chip layout and improved security to prevent this attack, so that chips manufactured since 1999 do not have this problem. Nowadays, most microcontroller manufacturers design their products so that the security fuses cannot be erased before the main memory is entirely cleared, thus preventing this low cost attack on their devices.

4.6.3 Requirements for reliable data deleting from memory

To avoid data remanence attacks in secure applications the developer should follow some general design rules that help making data recovery from semiconductor memories harder:

- Do not store cryptographic keys, passwords and other sensitive information for a long period of time in SRAM. Move them to new locations from time to time and zeroise the original storage, or flip the bits if that is feasible.
- To prevent low-temperature data remanence in SRAM, temperature detection circuits should be used in addition to the tamper detection.
- Cycle EEPROM/Flash cells 10–100 times with random data before writing anything sensitive to them to eliminate any noticeable effects arising from the use of fresh cells.
- Program all EEPROM/Flash cells before erasing them to eliminate detectable effects of the residual charge.
- Remember that some non-volatile memories are too intelligent, and may leave copies of sensitive data in mapped-out memory blocks after the active copy has been erased. That also applies to file systems which normally remove the pointer to the file rather than erasing the file itself.
- Use the latest highest-density storage devices as the newest technologies generally make data recovery more difficult.

Using encryption, where applicable, also helps make the data recovery from erased memory more difficult. Ideally, for secure applications, each semiconductor memory device should be evaluated against all possible outcomes of data remanence on its security protection.

Chapter 5

Invasive attacks

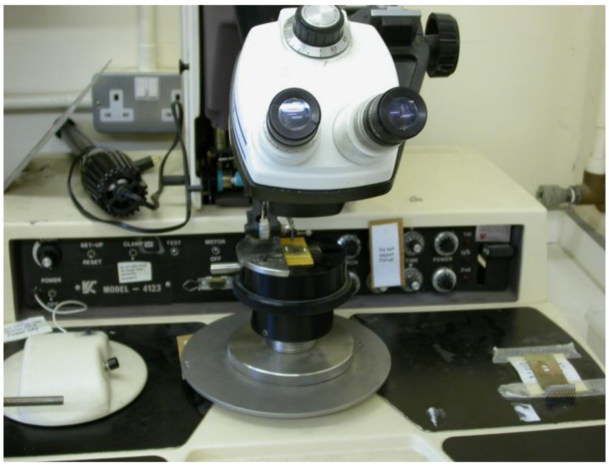
These attacks require direct access to the internal components of the device. If it is a security module or a USB dongle, then it has to be opened to get access to the internal memory chips. In the case of a smartcard or a microcontroller, the packaging should be removed followed by FIB or laser depassivation to get access to the internal wires buried deep under the passivation layer of the chip. Such attacks normally require a well equipped and knowledgeable attacker to succeed. Meanwhile, invasive attacks are becoming constantly more demanding and expensive, as feature sizes shrink and device complexity increases.

Some operations such as depackaging and chemical etching can still be performed by almost anyone with a small investment and minimal knowledge. There are also some attacks, for example optical reading of an old Mask ROM memory, or reverse engineering of a chip built with 1 µm technology and two metal layers, where gaining the access to the chip surface is enough to succeed. The necessary chemicals and tools are relatively cheap, and a suitable optical microscope could be bought second-hand for less than £1,000.

Normally invasive attacks are used as an initial step to understand the chip functionality and then develop cheaper and faster non-invasive attacks.

5.1 Sample preparation

Invasive attacks start with partial or full removal of the chip package in order to expose the silicon die. There are several methods, depending upon the package type and the requirements for further analysis. For microcontrollers, partial decapsulation is normally used, so that the device can be placed in a standard programmer unit and tested. Some devices cannot be decapsulated and still maintain their electrical integrity. In this case the chip die has to be bonded to a chip carrier using a bonding machine which connects to the bonding pads on the die with thin aluminium or gold wire (Figure 49). Such bonding machines are available from different manufacturers and can be bought second-hand for less than £5,000. The contacts to the die can be also established using microprobing needles on a probing station.



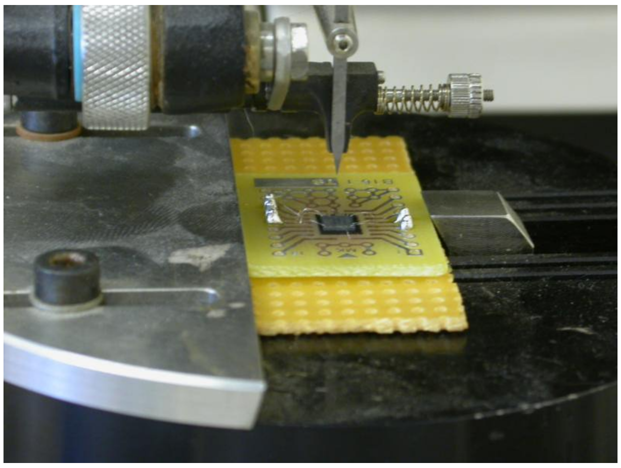


Figure 49. Kulicke & Soffa 4123 wedge wire bonder [103] and a smartcard chip being bonded on the PCB carrier

To undertake further work under a FIB or a SEM the chip surface has to be coated with a thin gold layer making it conductive, otherwise it will very quickly accumulate charge and the picture become dark. We used an Emitech K550 gold sputter coater [104] to coat samples prior to the FIB work. Some modern FIB machines have a built-in video camera for optical navigation, eliminating the need for the special coating.

5.1.1 Decapsulation

It is a common opinion that decapsulation is a complicated process which requires a lot of experience. In fact it is not and anyone capable of carrying out chemical or biological work in the context of a standard high-school program can do this. All the necessary experience could be obtained by decapping a dozen different samples. Some precautions should be taken as the acids used in this process are very corrosive and dangerous; ideally, the work should be performed in a fume cupboard to prevent inhalation of the fumes from acids and solvents. Eyes should be protected with safety goggles and appropriate acid-resistant gloves should be worn as the acid will cause severe burns if it accidentally comes into contact with the skin. Protective clothing should be worn as well.

The process of manual decapsulation usually starts with milling a hole in the package so that the acid will affect only the desired area above the chip die (Figure 50). The tools necessary for this operation are available from any DIY shop for less than £10.

The commonly used etching agent for plastic packages is fuming nitric acid (>95 %), which is a solution of nitrogen dioxide NO_2 in concentrated nitric acid HNO_3. It is very strong nitrifying and oxidizing agent; it causes plastic to carbonise, and it also affects copper and silver in the chip carrier island and pins. Sometime a mixture of fuming nitric acid and concentrated sulphuric acid H_2SO_4 is used. This speeds up the reaction with some types of packages and also prevents the silver used in bonding pads and chip carrier from reacting.

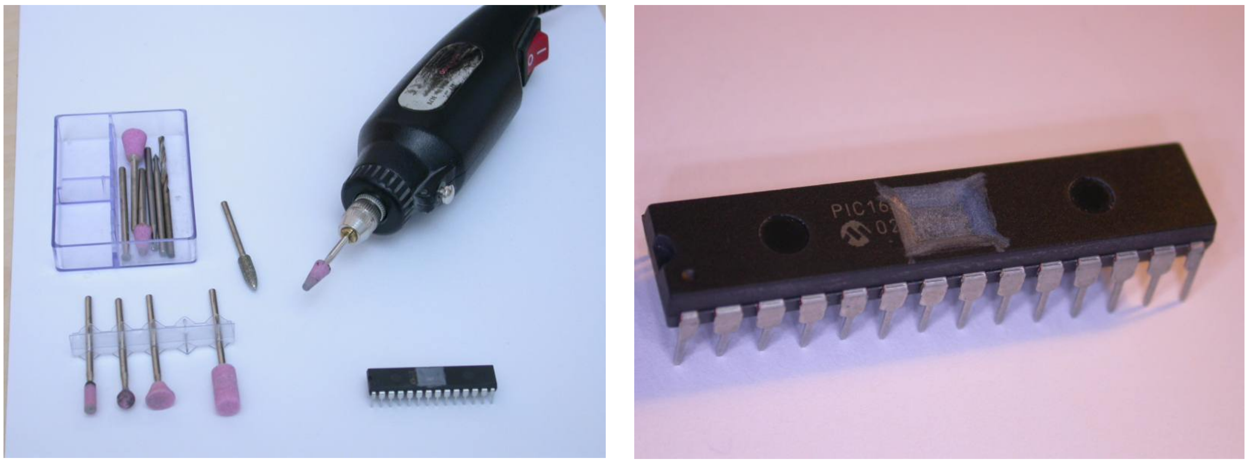


Figure 50. Mechanical grinding tool for plastic packages preparation for manual decapsulation and pre-milled chip in the DIP package

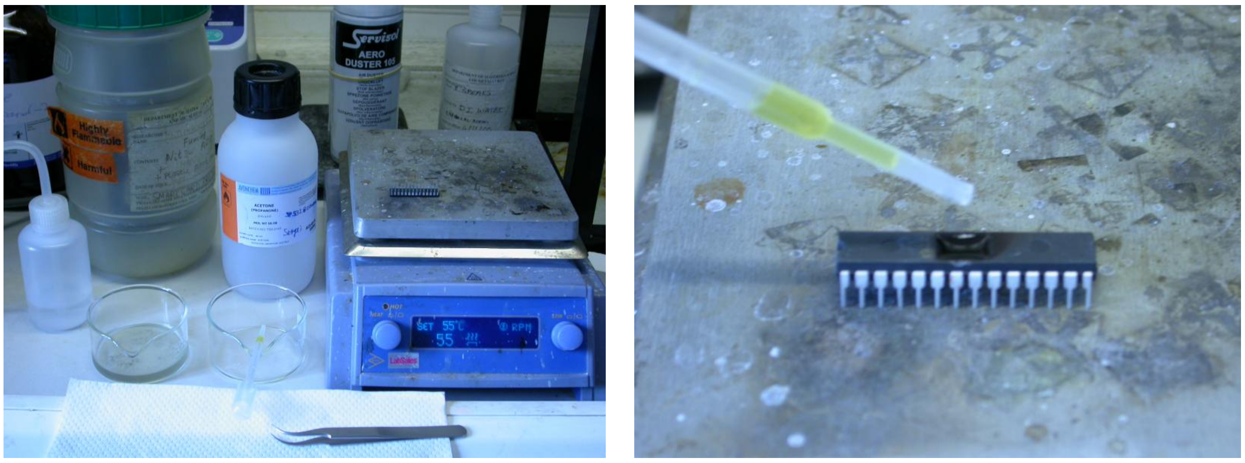


Figure 51. Manual decapsulation tools and dropping the fuming nitric acid into the pre-milled hole

The acid is normally applied in small portions with a pipette into a pre-milled hole in a chip preheated to 50–70˚C (Figure 51). After 10–30 seconds the chip is sprayed with dry acetone from a washing bottle to remove the reaction products. This process has to be repeated several times until the die is sufficiently exposed. To speed up the process, the chip can be placed in a sand bath and the acid can be preheated in a glass beaker.

The acid residues can be removed from the etched plastic and from the chip surface by ultrasonic treatment. For that the chip is placed into a beaker with acetone and then put in an ultrasonic bath for 1–3 minutes (Figure 52). After washing the chip with acetone and drying it in an air jet, we have a clean and fully operational chip (Figure 53).

For decapping chips in large quantities an automatic decapsulation system can be used, for example PA103 from Nippon Scientific [105]. Very little skill and experience is required to operate it, and packages can be decapped easily even by unskilled workers. Such systems cost over £10,000 and are bought by relatively large labs only. They also consume ten times more acid compared to the manual method, and their waste has to be disposed of in a proper way to avoid harm to the environment.

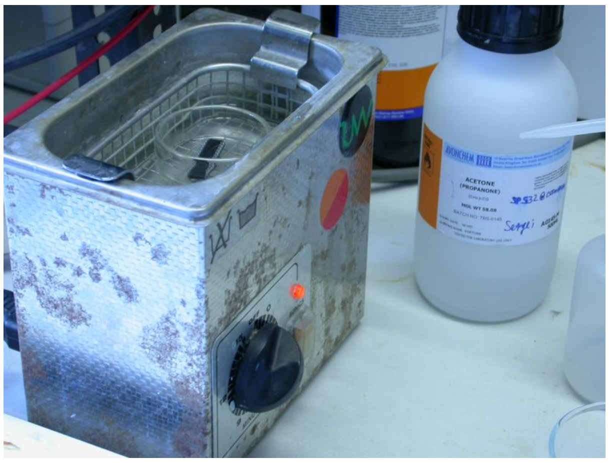


Figure 52. Cleaning the chip in the ultrasonic bath

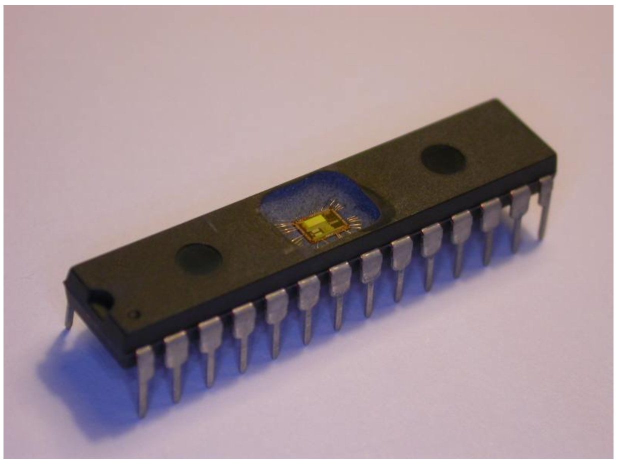


Figure 53. Decapsulated sample in DIP package

A very similar approach can be used for decapsulating chips from the rear side. The only obstacle is the copper plate under the chip die which reacts slowly with the fuming nitric acid. That could create problems if the automatic decapsulator is used because the surrounding plastic will be etched away before this copper plate and the chip leads are very likely to be damaged (Figure 54). However, access to the rear side of the die can be established without using chemical etching. The chip package can be milled down to the copper plate which is then removed mechanically. The residues of the glue used to attach the die to the plate can be removed with solvents or by scraping it off with a wooden toothpick stick.

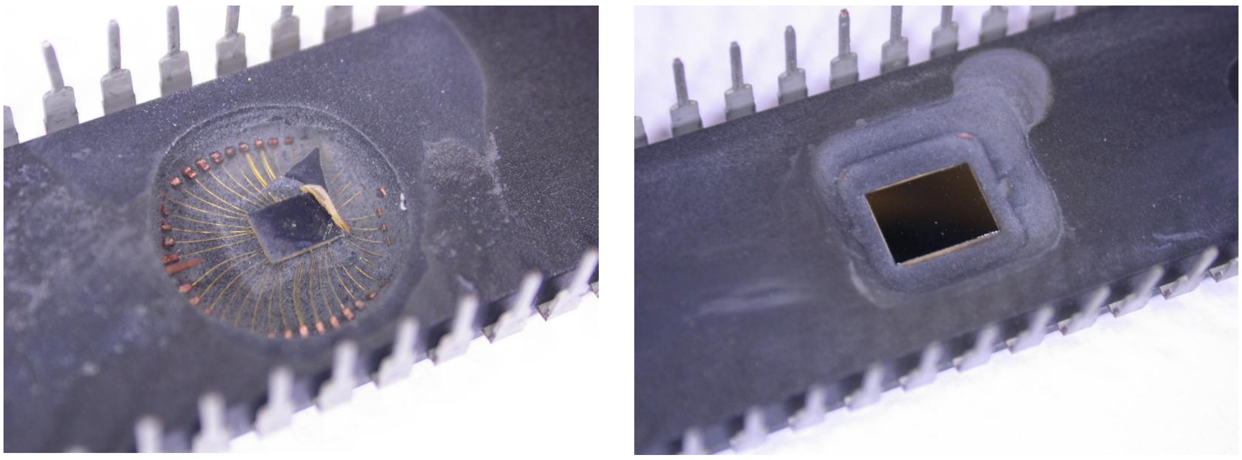


Figure 54. Over-decapped chip from the rear side as a result of automatic decapsulation (sample was provided by Radiolinija) and manually decapped sample

The same partial decapsulation technique can be used for smartcards as well (Figure 55) although not all of them would maintain electrical integrity. Very often the chip has to be decapsulated completely and then bonded onto a chip carrier.

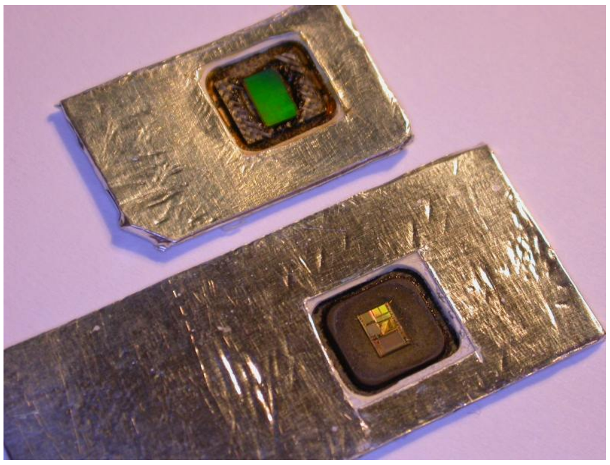
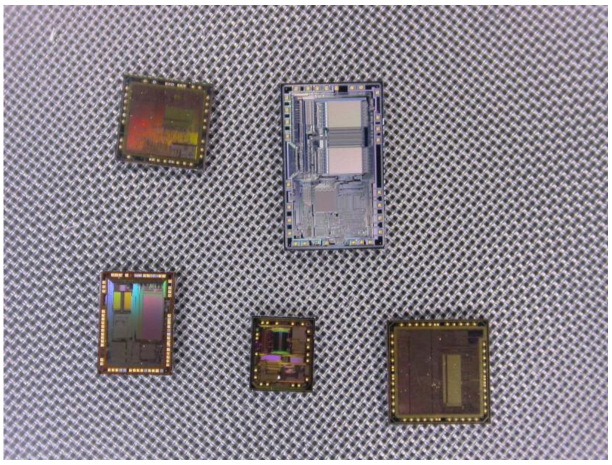



Figure 55. Decapped smartcard samples and fully decapped chips

Other methods used for opening the chip packages are described in the literature and only very few of them require expensive tools [66][106]. One interesting approach suggests using an acid resistant tape to prevent the acid reacting with unwanted parts [107]. The chip is placed on a glass to prevent lead bending and then covered with acid resistant tape. The tape over the area to be etched is cut away and the whole package is then immersed in a chemical solution to etch away the plastic. This method is limited to QFP, SOP, BGA and other thin packages.

5.1.2 Deprocessing

The opposite process to chip fabrication is called deprocessing. A standard CMOS chip has many layers. The deepest doping layers inside the substrate form the transistors. The gate oxide layer isolates the gate from the active area of the transistors. The polysilicon layer on top of it forms the gates and interconnections. The interlayer oxide isolates conducting layers. Metal layers, usually made of aluminium (Al), form the signal wires, and they are connected with other layers through 'via' plugs (Al, W, Ti). Finally, a passivation layer made out of silicon oxide SiO_2 or nitride Si_3N_4 protects the whole structure from moisture and air which could harm the die. In plastic packages the passivation layer is covered with a polymer layer, usually polyimide, to protect against sharp grains in the compound during the package formation.

There are two main applications of deprocessing. One is to remove the passivation layer, exposing the top metal layer for microprobing attacks. Another is to gain access to the deep layers and observe the internal structure of the chip.

Three basic deprocessing methods are used: wet chemical etching, plasma etching, also known as dry etching, and mechanical polishing [67]. In chemical etching each layer is removed by specific chemicals. Its downside is its isotropic nature, i.e. uniformity in all directions. That produces unwanted undercutting. As a result, narrow metal lines will have a tendency to lift off the surface. Isotropic etching also leads to etching through holes such as vias, resulting in unwanted etching of underlaying metallization. Plasma etching uses radicals created from gas inside a special chamber. They react with the material on the sample surface to form volatile products which are pumped out of the chamber. As the ions are accelerated in an electric field they usually hit the surface of the sample perpendicularly. The removal of material is strongly anisotropic (directional). Only the surfaces hit by the ions are removed, sides perpendicular to their paths are not touched. Mechanical polishing is performed with the use of abrasive materials. The process is time-consuming and requires special machines to maintain the planarity of the surface. From the inspection perspective, the advantages of using polishing over wet and dry etching techniques is the ability to remove layer by layer and view features in the area of interest within the same plane. It is especially useful on multilayer interconnect processes fabricated with advanced planarisation techniques.

For wet and dry etching, each type of material requires certain etchants to be used. Some of them have very high selectivity and remove only the desired layer; others affect many layers at a time. For example, silicon and polysilicon can be etched with a mixture of hydrofluoric acid HF and nitric acid HNO_3, but HF etches silicon oxide as well. Table 3 summarises the etchants used for different materials in wet and plasma etching.

Material	**Wet etching chemicals**	**Dry etching gases**
Si	$HF + HNO_3$, KOH	CF_4, C_2F_6, SF_6
Poly Si	$HF + CH_3COOH + HNO_3$	CF_4, SF_6
SiO_2	HF, $HF + NH_4OH$	CF_4, $CF_4 + O_2$, CHF_3
Al	HCl, $H_2O_2 + H_2SO_4$, $HPO_3 + HNO_3 + CH_3COOH$, KOH	CCl_4, BCl_3
W, Ti	$HF + HNO_3$, $H_2O_2 + H_2SO_4$, H_2O_2	CF_4
Si_3N_4	$HF + HNO_3$, HPO_3, Nitrietch	CF_4
Polyimide	H_2O_2, $H_2O_2 + H_2SO_4$	CF_4, $CF_4 + O_2$

Table 3. Etching agents used for wet chemical etching and dry plasma etching [66].

Other etchants are used for specific purposes, such as doping etchants with a doping-dependent etch rate to make visible doping fronts and p-n junctions. Such etchants are used, for example, to make visible the contents of VTROM in modern smartcards [8]. More information about different etchants and etching technology can be found in the literature on failure analysis techniques.

For wet chemical etching we used the Nitrox wet etchant [108] – one of the most effective etching agents for silicon nitride and silicon dioxide passivation layers which selectively removes the passivation layers of integrated circuits while preserving full device functionality. To observe deeper layers, top aluminium layers were etched away with a 20% water solution of hydrochloric acid HCl or 33% water solution of potassium hydroxide KOH. Although wet etching does not provide good uniformity across the die surface, a lot of information about the internal chip structure can be obtained. Examples of such operations are presented in Figures 56 and 57. As can be seen, wet chemical etching does not provide very good uniformity over the surface resulting in some areas where the top metal is not entirely removed and other areas where the underneath layer is starting to be etched. Also, as can be seen in Figure 57, some long metal wires lifted off the surface obstructing the view.

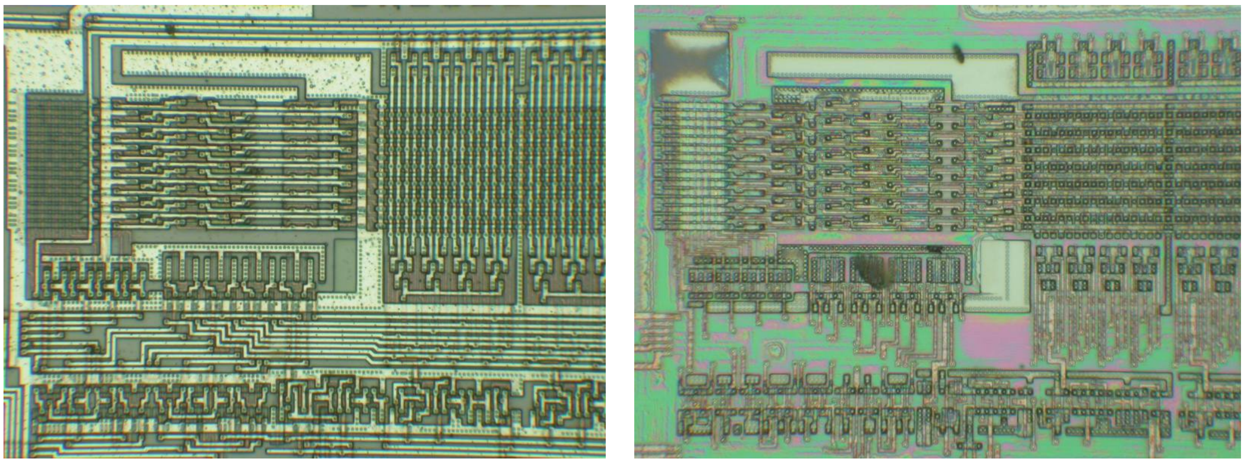


Figure 56. Original and chemically etched die of the Motorola MC68HC705C9A microcontroller [23]. The metal layer is removed exposing the polysilicon and the doping layers. 200× magnification

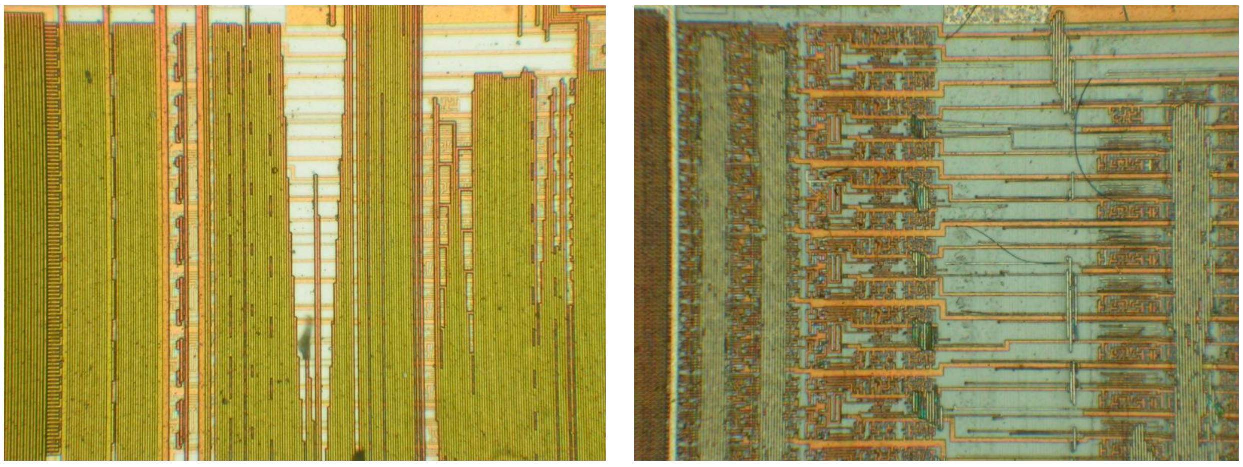


Figure 57. Original and chemically etched die of the Microchip PIC16F76 microcontroller [109]. The top metal layer is removed exposing the second metal layer. 200× magnification

Deprocessing using wet chemical etching does not require much more experience than decapsulation and all the necessary chemicals can be bought for about £100. Care must be taken during the work, as these chemicals are very aggressive and dangerous, especially the ones containing fluorine.

5.2 Reverse engineering

Reverse engineering is a technique aimed at understanding the structure of a semiconductor device and its functions. In case of an ASIC or a custom IC, that means extracting information about the location of all the transistors and interconnections. In order to succeed, a general knowledge of IC and VLSI design is required. All the layers formed during chip fabrication are removed one-by-one in reverse order and photographed to determine the internal structure of the chip. In the end, by processing all the acquired information, a standard netlist file can be created and used to simulate the device. This is a tedious and time-consuming process, but there are some companies, for example Chipworks [110], which do such work as a standard service.

When it comes to reverse engineering smartcards and microcontrollers, both structural and program-code reverse engineering are required to understand how the device works. First, the security protection needs to be understood by partial reverse engineering of the chip area associated with it. Thus if memory bus encryption was used, the hardware responsible for this should be reverse engineered. Then, finally, the internal memory contents have to be extracted and disassembled to understand device functions.

A slightly different approach is required for reverse engineering CPLDs and FPGAs. Even if the security protection is defeated and the attacker manages to extract the configuration bitstream file from the device, he will have to spend a substantial amount of time and effort to convert it into the logic equations and primitive blocks for further simulation and analysis. Meantime, there are some companies on the market, for example Bottom Line Technologies [111], which provide bitstream reverse engineering for CPLDs and FPGAs.

5.2.1 Optical imaging for layout reconstruction

The most important tool for reverse engineering silicon chips down to 0.18 μm feature size is an optical microscope with a CCD camera to produce mosaics of high-resolution photographs of the chip surface. Not every microscope would do. As light cannot pass through the chip, the microscope should have reflected light illumination. The image should be sharp and without geometric distortion and colour aberration, otherwise it will not be possible to stick all the images together. The most important parameters of the microscope are resolution and magnification. The resolution of a microscope mainly depends upon its objective lenses and is defined as the smallest distance between two points on a specimen that can still be distinguished as two separate entities. Resolution is a somewhat subjective value in microscopy because at high magnification an image may appear non-sharp but still be resolved to the maximum ability of the objective [112].

Normally a microscope objective has at least two parameters printed on it – magnification and numerical aperture (NA). Modern optical microscopes provide magnification up to 9,000× and 500× magnification is provided by most modern microscopes. Numerical aperture determines the resolving power of an objective, but the total resolution of a microscope system is also dependent upon the numerical aperture of projection optics [113]. The higher the numerical aperture of the total system the better the resolution. The numerical aperture is related to the angle μ which is one-half of the angular aperture at which the light cone comes to the specimen surface: $NA = n \sin(\mu)$. The relationship between the numerical aperture and the resolution can be expressed with the following equation: $R = 0.61\ \lambda / NA$, where λ is the wavelength of light used for observation.

In practice the maximum resolution which can be achieved with a standard 100× objective (*NA* = 0.9) is about 0.3 µm. In order to obtain higher working NA the refractive index of the medium between the objective and the specimen must be increased. There are objectives that allow imaging in water (*n* = 1.33) and immersion oil (*n* = 1.51). That increases the maximum resolution up to 0.2 µm for 100× objective. Another way of increasing the resolution is using a shorter wavelength. By shifting to near-ultraviolet (NUV) light with 360 nm wavelength, the resolution can be increased to 0.18 µm, but this requires special CCD cameras.

Some microscopes have additional features aimed at increasing the contrast of the image and thereby achieving the highest possible resolution. These are darkfield (DF) illumination, differential interference contrast [114], phase contrast [115] and confocal imaging [116]. All the major microscope manufacturers such as Nikon, Olympus, Carl Zeiss and Leica offer a wide range of models from basic to high-end; the latter have all the features necessary to achieve the highest resolution. There are models specifically designed for semiconductor analysis such as the Nikon Optiphot 200C [117], Olympus MX50 [118], Zeiss Axiotron 2 [119] and Leica INM100 [120].

The main disadvantage of high resolution microscopes is the short working distance between the objective and a specimen, especially at high magnifications (about 0.3 mm with 100× objective). As a result partially decapsulated chips cannot be observed and full decapsulation of the die is required. Using microscopes with a long working distance, for example the Mitutoyo FS70 [121] with 13 mm working distance on 200× objective, helps solve this problem but at a cost: the resolution is at most 0.4 µm because the NA cannot be high.

Another problem of the high-resolution objectives is a very short depth of focus, which makes the out-of-focus planes look blurred, thus reducing the image quality. This is more noticeable on multilayer chips where the distance between the top and the bottom layer is more than 1 µm. Confocal microscopy reduces this effect as all out-of-focus planes become dark or appear in different colours depending from their depth. Such confocal systems are very expensive, especially the ones that use laser scanning, and therefore can be afforded by relatively large labs only. Even second-hand confocal microscopes start from £10,000.

Layout reconstruction requires the images of all the layers inside the chip to be combined. The images are normally taken automatically using a motorised stage to move the sample and special software to combine all the images together [122].

Normally, for semiconductor chips fabricated with 0.13 µm or smaller technology, images are created using a SEM which has a resolution better than 10 nm.

5.2.2 Memory extraction

Direct optical memory extraction is possible only from certain types of Mask ROM devices. For example, NOR Mask ROM with the active layer programming (see Figure 18) used in the old revision of the Motorola MC68HC705P6A microcontroller [123] built with 1.2 µm technology (Figure 58) can be read under a microscope with 500× magnification.

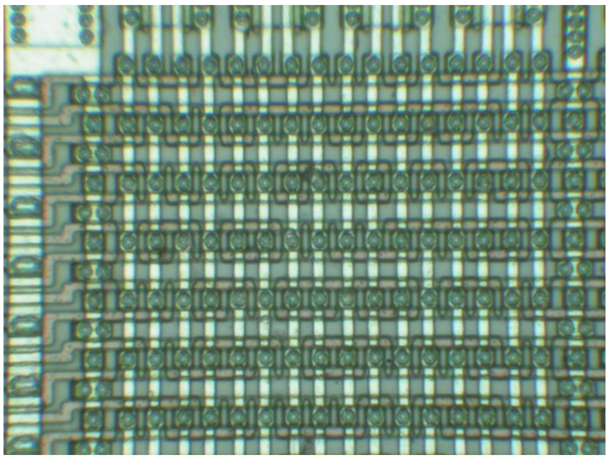


Figure 58. Optical image of the Mask ROM inside MC68HC705P6A microcontroller. 500× magnifi-cation

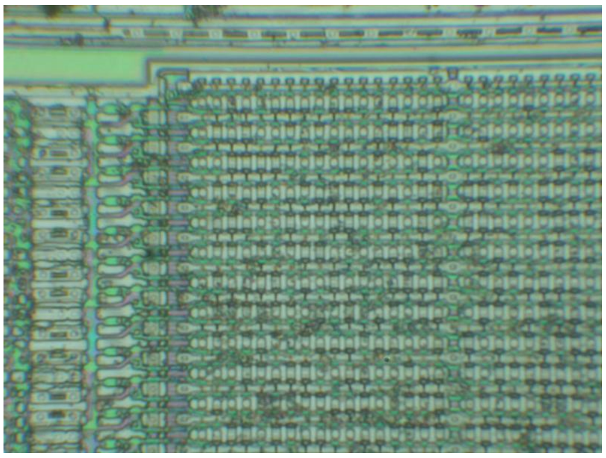


Figure 59. Optical image of the Mask ROM inside PIC16CR57A microcontroller after plasma etching (sample was provided by Radiolinija). 500× magnification

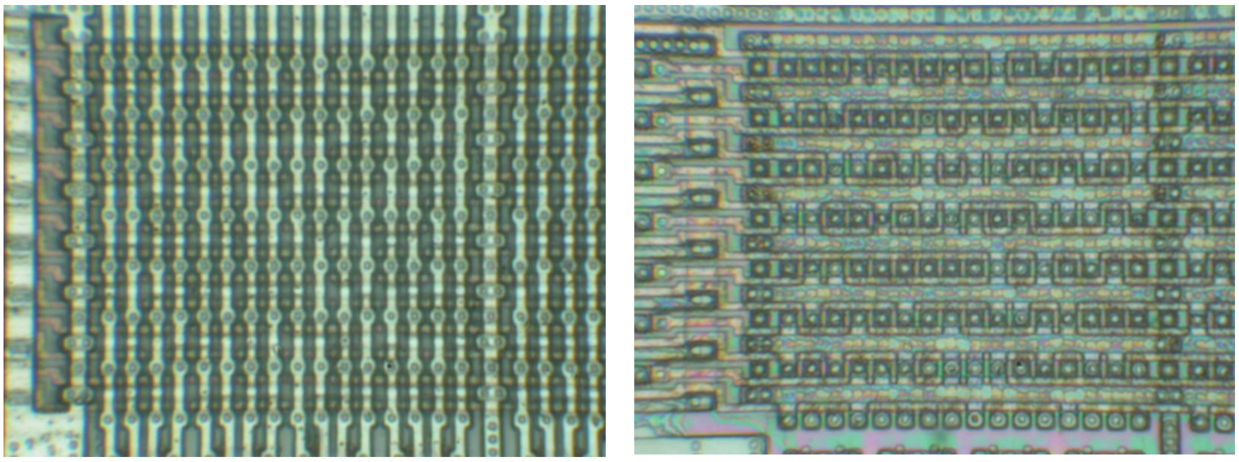


Figure 60. Optical image of the Mask ROM inside MC68HC705C9A microcontroller before and after wet chemical etching. 500× magnification

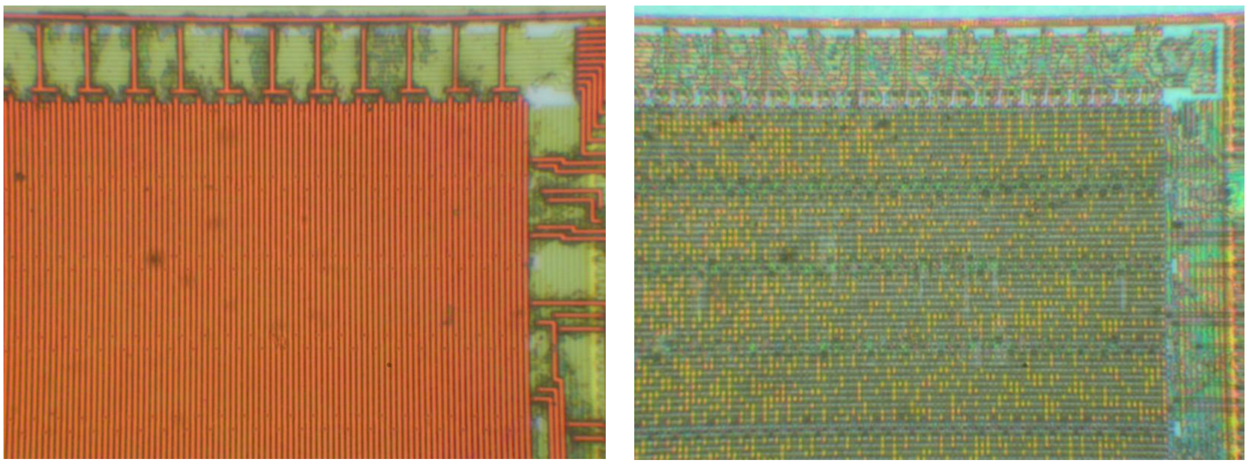


Figure 61. Optical image of the Mask ROM inside μPD78F9116 microcontroller before and after wet chemical etching. 500× magnification

The same memory type but built with newer technologies such as 0.9 µm in the Microchip PIC16CR57 microcontroller [124] and 1.0 µm in the Motorola MC68HC705C9A microcontroller [23] requires deprocessing because the top bit-line metal wires obstruct observation of the transistors (Figures 59 and 60).

NAND Mask ROM memory type with metal layer programming (see Figure 22) was used in the NEC µPD78F9116 microcontroller [125] fabricated with 0.35 µm technology. As all the internal layers were planarised, deeper layers cannot be observed unless the top metal layer is removed (Figure 61). This was accomplished by using Nitrox etching for the passivation layer followed by treatment in a 33% water solution of KOH to etch the top aluminium metal layer but preserving the interconnection layer which is probably made out of tungsten (because when the HCl solution was used to etch the top metal layer, the interconnection layer was etched away as well).

5.3 Microprobing

The most important tool for invasive attacks is a microprobing station (Figure 62). It consists of five elements: a microscope, stage, device test socket, micromanipulators and probe tips. The microscope must have long working distance objectives – sufficient enough to accommodate six to eight probe tips (Figure 63) between the sample and the objective lens. It should also have enough depth of focus to follow the probe tip movement. Usually the microscope has 3–4 objectives to accommodate different magnification and depths of focus. Lower magnification with greater focus depth is used for coarse location of the probe tip and higher magnification for placing the tip on a conductor wire or a test point. The chip is normally placed in a test socket that provides all the necessary signals and is controlled by a computer.

On a stable platform around the test socket, several micropositioners are installed. They allow us to move a probe tip with submicron precision. The probe tip can be either passive or active. The passive tip, for example Picoprobe T-4 [126], can be used for both eavesdropping and injecting signals, but as it is normally connected directly to an oscilloscope, it has low impedance and high capacitance. As a result it cannot be used for probing internal signals on the chip, except for the bus lines which are usually buffered. Another application for the passive tips is making connections to the bonding pads on a fully decapsulated chip. The active tip, for example Picoprobe 12C [127], has a FET amplifier close to the end of the tip. Active tips offer high bandwidth (1 GHz for Picoprobe 28) with low loading capacitance (0.02 pF for Picoprobe 18B) and high input resistance (>100 GΩ for Picoprobe 18B).

The probe tips are made out of a tungsten wire which is sharpened to <0.1 µm at the end for probing small features. The stage under the microscope is used for coarse positioning of the test socket with the chip. There are no specific requirements for it and a 1 µm precision is enough for most applications.

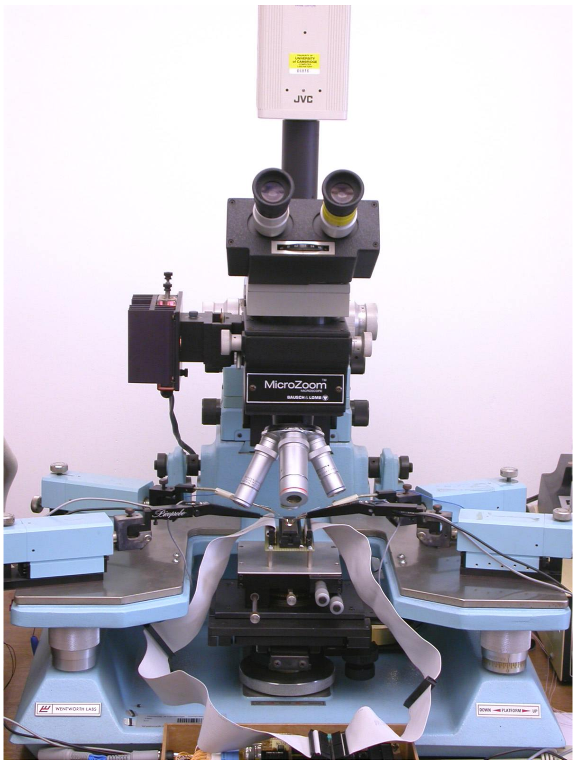


Figure 62. Microprobing a chip on the Wentworth Labs MP-901 manual probing station

Modern probing stations benefit from full automatic control over the microscope, stage and micropositioners. For simple applications, a manually controlled probing station is enough and can be bought second-hand for less than £5,000. Passive probe tips are very cheap (less than £3 each) but active probes are quite expensive – over £40 for the tip plus over £1,000 for the tip holder with amplifier. However, they can easily be built from a £2 operational amplifier, for example MAX4174AB or MAX427, and a passive tip soldered directly to its input (Figure 64).

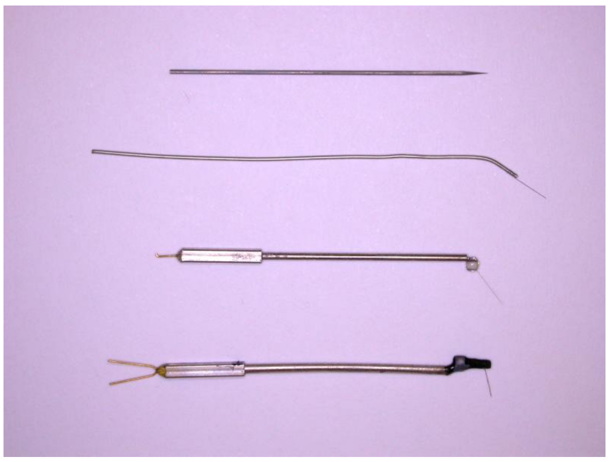


Figure 63. Passive and active probe tips (Picoprobe ST-20, T-4, 12C-4 and 18B-4)

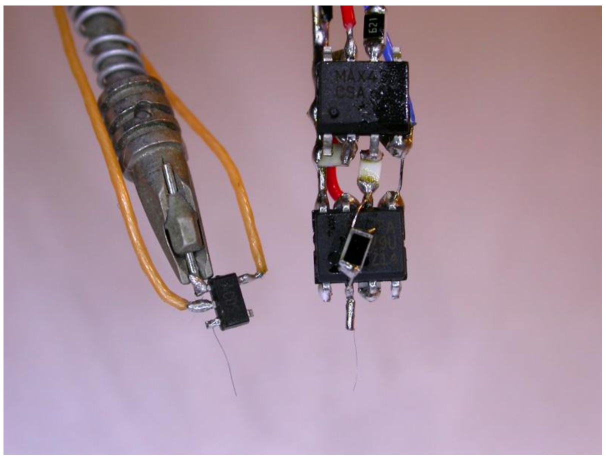


Figure 64. Home-made active probe tips

Usually to extract the information such as memory contents or a secret key, microprobing is applied to the internal CPU data bus. It is difficult to observe the whole bus at a time and various techniques can be used to overcome this. For instance, the same transaction or memory read operation can be repeated many times and then two to four probes are used to observe the signals which then combined into a complete bus trace. Memory extraction from smartcards is more difficult as their software usually does not provide any access to the internal memory. To succeed we have to abuse a CPU component such as an address counter or instruction decoder to access all memory cells for us. The program counter is already incremented automatically for each instruction cycle and used to read the next address. This makes it perfectly suited for memory scanning. All we have to do is to prevent the processor from executing jump, call and return instructions so it cannot disturb the program counter. Tiny modifications to the instruction decoder or program counter usually have the desired effect. This is not an easy task and normally requires partial reverse engineering of the CPU circuit. This could be more challenging for modern smartcards with a top protection mesh and a glue logic design.

5.3.1 Laser cutter

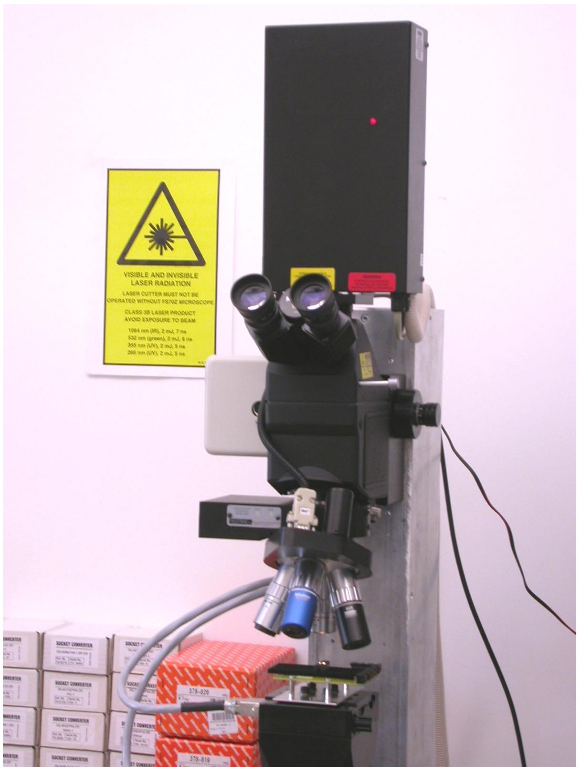


Figure 65. New Wave Research QuikLaze-II laser cutting system [128]

In silicon chips, the top-layer aluminium interconnect lines are covered by a passivation layer. We have to remove the passivation before the probes can establish contact. The most convenient and easy-to-use depassivation technique involves a laser cutting system (Figure 65). The system consists of the laser head mounted on the camera port of a microscope and the submicron-precision stage to move the sample. Very few microscopes are suitable for laser cutting, and usually a Mitutoyo FS60Y or FS70L metallurgical microscope with NUV and NIR laser objectives is used [121]. Carefully dosed laser flashes remove patches of the passivation layer with micrometer precision (Figure 66).

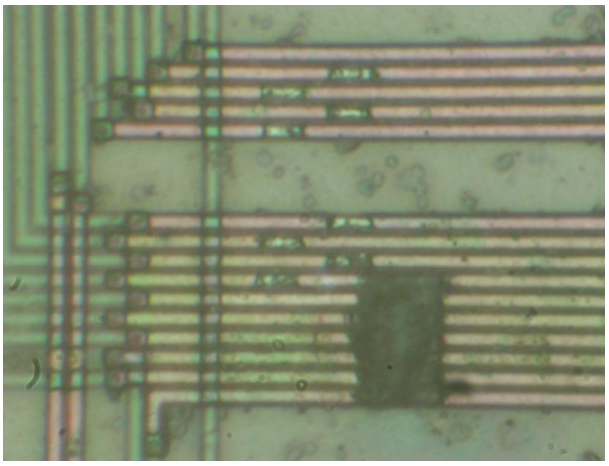


Figure 66. Passivation layer cuts with UV laser and wires cut with the green laser under 100× NUV objective. 1000× magnification

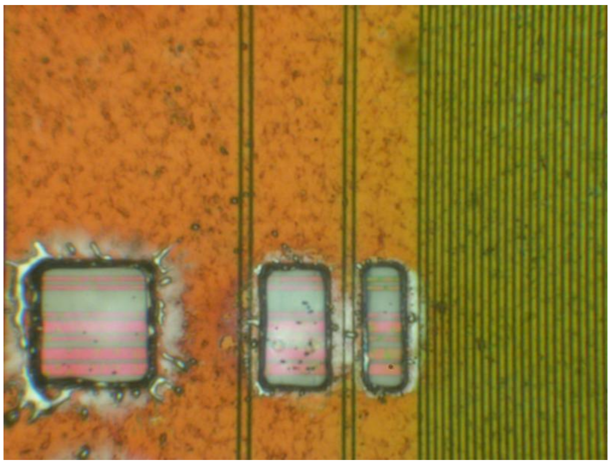


Figure 67. Top metal layer cuts with IR laser under 100× NIR objective exposing the second metal layer. 500× magnification

Normally, UV laser is used for removing a polyimide or other organic layer which is very often present on top of the passivation layer. UV laser is also used to cut the passivation, then green laser is used to cut the metal wires (Figure 66) and IR laser to cut through the top metal layer to access wires in the second metal layer (Figure 67).

Complete laser cutting systems cost over £50,000 and as such systems are not normally used for failure analysis, their second-hand market is very limited. Low-budget attackers could try to use chemical wet etching to carefully remove the passivation layer. Also, vibrations in the probing needle could be used to break holes in the passivation and this was successful for old 1.2 μm chips.

5.3.2 FIB workstation

For semiconductor devices fabricated with 0.5 μm or smaller technology more sophisticated tools are required for establishing contacts with the interconnection wires. The ultimate tool for that purpose is a FIB workstation. It can be used not only to create test points, but also for imaging and repairing as well. In failure analysis it is used for cross section preparation and defect analysis. A FIB provides the improved spatial resolution and precision that are required for probe-point creation on deep submicron technologies.

A FIB workstation consists of a vacuum chamber with a particle gun comparable to a SEM with the difference that, instead of electrons, gallium ions are accelerated and focused from a liquid metal cathode into a beam which can be as small as 5–10 nm in diameter. FIBs can image samples from secondary particles similar to a SEM with down to 10 nm resolution. By increasing the beam current, chip material can be removed with the same resolution at a rate of around 0.1 µm^3 nA^{-1} s^{-1} (Figure 68 and 69). Better etch rates can be achieved by injecting a gas like iodine via a needle that is brought to about 0.5 mm above the chip surface. The gas is absorbed on to the surface and reacts with removed material to form a volatile compound that can be pumped away and is not redeposited. Using this gas-assisted etch technique, holes with aspect ratios up to 15:1 can be created to get access to deep metal layers without damaging nearby structures. By injecting a platinum-based organometallic gas that is broken down on the chip surface by the ion beam, platinum can be deposited to establish new contacts (Figure 70). With other gas chemistries, even insulators can be deposited to establish surface contacts to deep metal without contacting any covering layers.

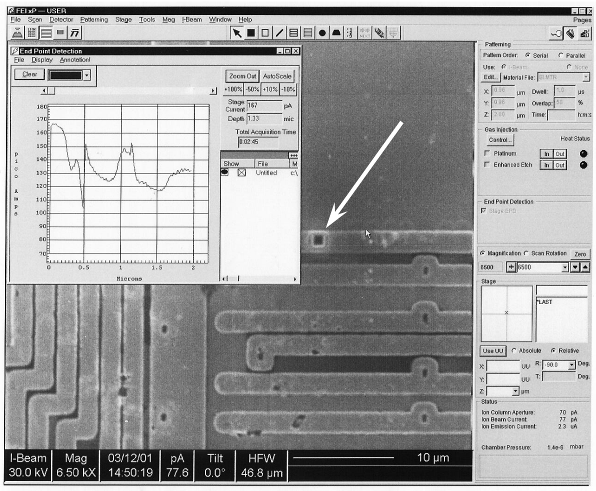

I-Beam 30.0 kV | Mag 6.50 kX | 03/12/01 14:50:19 | pA 77.6 | Tilt 0.0° | HFW 46.8 µm | 10 µm

Figure 68. The process of milling the hole using FIB

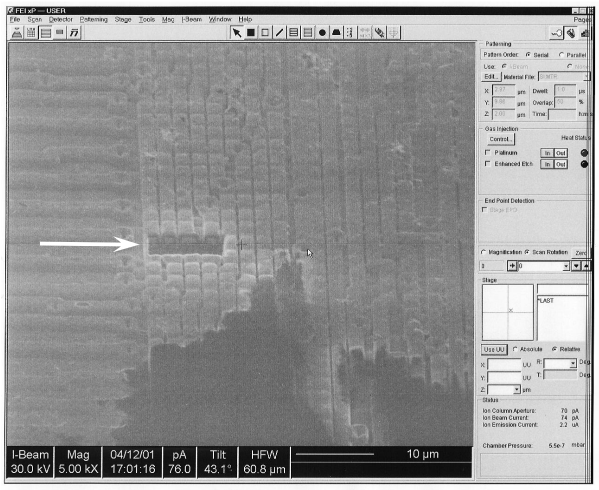

I-Beam 30.0 kV | Mag 5.00 kX | 04/12/01 17:01:16 | pA 76.0 | Tilt 43.1° | HFW 60.8 µm | 10 µm

Figure 69. Cutting the wires using FIB

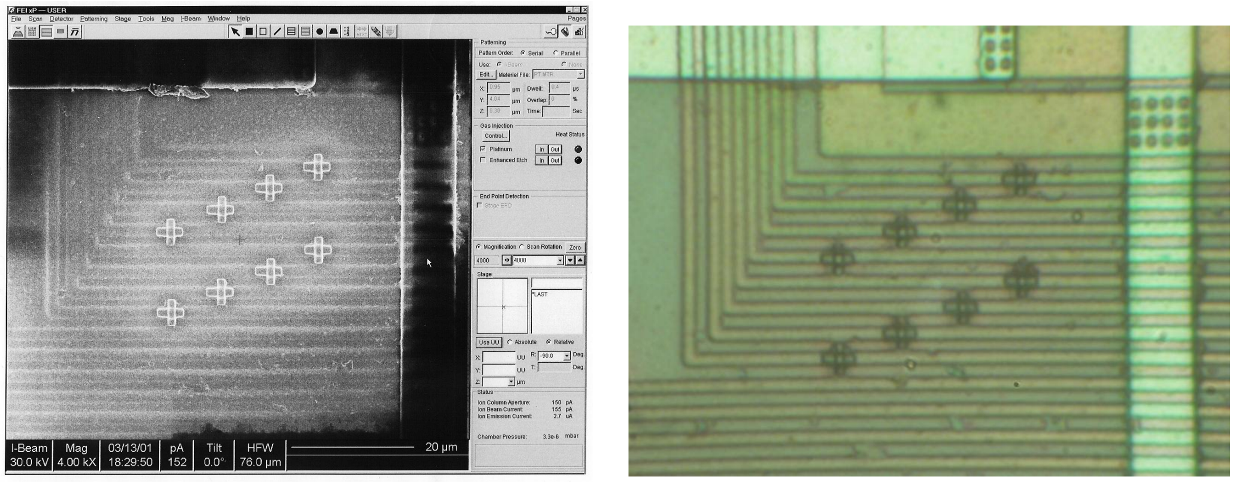

I-Beam 30.0 kV | Mag 4.00 kX | 03/13/01 18:29:50 | pA 152 | Tilt 0.0° | HFW 76.0 µm | 20 µm

Figure 70. Test points created under FIB and optical image of these points

The FIB operator can navigate blindly on a chip surface with down to 0.1 µm precision even if the chip has been planarised and has no recognisable surface structures. Chips can also be polished from the rear side down to a thickness of 10–20 µm. Using infrared imaging, it is possible to locate individual transistors and contact them through the silicon substrate [67].

Although FIB machines are extremely expensive – over half a million pounds new and from £40,000 second-hand, an attacker can rent the time on them from various labs around the world at about £200 per hour.

5.4 Chip modification

It is not always necessary to create test points and extract the information from a chip by microprobing its internal buses. Sometime it is possible, especially for microcontrollers, to disable the security protection circuit by either cutting one of the internal metal interconnection wires (Figure 71) or by completely destroying the circuit associated with the security protection using a laser cutter (Figure 72). A FIB can be used for more sophisticated attacks like connecting the wire that transmits the security state to either the ground or the supply line. A microprobe tip can be used for that as well, but might be hard to achieve if the target wire is not in the top metal layer.

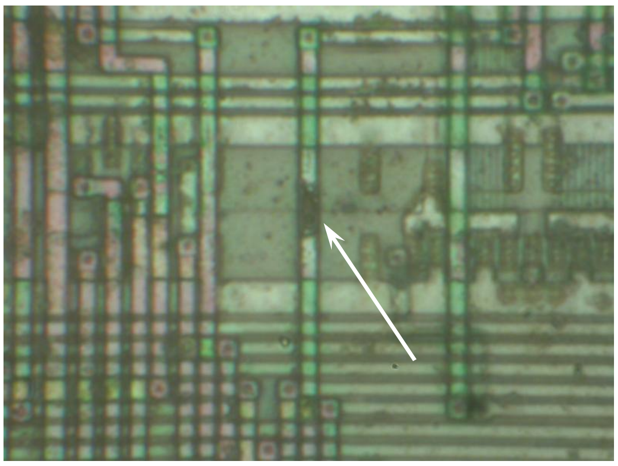


Figure 71. Cutting a single wire in the PIC12C508A microcontroller disables the security. 1000× magnification

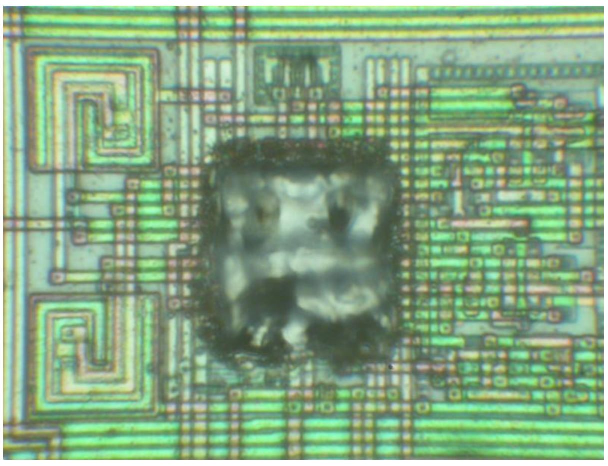


Figure 72. Disabling the security in the PIC16F628 microcontroller by destroying the fuse control circuit with a laser cutter. 500× magnification

Chip modification always requires at least partial reverse engineering of the chip to find the point for possible attack. It means that only a knowledgeable and well equipped attacker could succeed. At the same time, as will be seen in the next chapter, semi-invasive attacks can be used to locate the security protection circuit and find the right point for the attack. They could also be used to directly change the state of a transistor without expensive depassivation techniques.

Chapter 6

Semi-invasive attacks

This is a new class of attacks we described in 2002 when we introduced the optical fault induction attacks to the community [1]. Semi-invasive attacks, like invasive attacks, require depackaging the chip to get access to the chip surface. But the passivation layer of the chip remains intact – semi-invasive methods do not require electrical contact to the metal surface, so there is no mechanical damage to the silicon.

As invasive attacks are becoming constantly more demanding and expensive, with shrinking feature sizes and increasing device complexity, semi-invasive attacks become more attractive as they do not require very expensive tools and give results in a shorter time. Also, being applied to a whole transistor or even a group of transistors they are less critical to the small feature size of modern chips.

Semi-invasive attacks are not entirely new. Electromagnetic analysis is best performed on a naked chip [129], and the old EPROM-hacking trick of exposing the memory protect bit of a microcontroller to UV light usually entails depackaging it. Semi-invasive attacks could in theory be performed using such tools as UV light, X-rays, lasers, electromagnetic fields and local heating. They could be used individually or in conjunction with each other. However, this field has hardly been explored.

Semi-invasive methods are starting to be used for failure analysis. This involves backside imaging and signal probing in flip-chip and modern deep-submicron devices which cannot be accessed in a standard way. Using semi-invasive methods for hardware security analysis of semiconductor devices in postproduction testing could help avoid some security problems and save time and money on expensive and time-consuming invasive methods.

We will now show that extremely powerful attacks can be carried out quickly using very cheap and simple equipment. Such attacks could be easily automated by using a computer-controlled motorised stage and they could be accomplished in a few hours, compared to days or weeks for invasive attacks.

6.1 UV attacks

These are among the oldest attacks used on microcontrollers since their release in the middle seventies. UV attacks were often considered invasive attacks before. But as most of them require only decapsulation of the chip, they certainly belong to the class of semi-invasive attacks. These attacks can be applied to many OTP and UV EPROM microcontrollers as their protection is designed to withstand low cost non-invasive attacks only.

The attack can be divided into two stages – finding the fuse and resetting it to the unprotected state with a UV light. As the security fuse is normally designed such that it cannot be erased earlier than the program memory, the UV light cannot be applied to the whole chip. Either the memory must be protected with opaque material, or the UV light can be applied to the fuse selectively by using a microscope or a UV laser.

6.1.1 Locating the security fuses

The security fuse could be physically separate from the main memory, share the same area with the memory or even be embedded into the memory array. Different implementations were discussed in Chapter 2.

Various methods can be used to find the exact location of the fuse. The universal, but expensive and time-consuming method is reverse engineering the whole chip. Partial reverse engineering could be used to save time. For example, the high voltage used for memory programming is normally supplied from an external pin and could be traced down to all the EPROM memory cells including the fuses. This can be done easily under an optical microscope for chip technologies down to 0.8 µm. Smaller technologies, especially with planarisation, make optical analysis almost impossible and require deprocessing the chip to observe the internal structure. Another method involves causing damage to different parts of the chip and observing the result. This process could take a long time and requires many samples to succeed.

If the fuse is located very close to the memory or even embedded in it, localisation could be very difficult. At the same time, the UV light that is used for erasing the fuse can be used to find it as well. The test sequence is very simple. Half of the surface of programmed and secured chip is covered with any opaque material, for example duct tape, and then the chip is placed in a UV eraser for the standard erase time. If the protection is removed then it is in uncovered part, otherwise – swap covered and uncovered areas. Repeating the process for each subsequent half, the attacker can find the location of the fuse with 10–20 µm precision in about 15 iterations which normally takes no longer than three hours. To achieve better precision, which is required when the fuse is embedded into the main memory, photoresist masks or UV lasers should be used. However, there is an even more effective, and cheaper, method suggested below which uses a marker pen to paint over the area which needs to be protected from the UV light.

Other semi-invasive methods could be used to find the fuse. One is a laser scanning technique which allows us to read the state of transistors directly. Scanning the same chip twice – first with the security bit set and second with it reset and then subtracting one scan from another, will reveal the changed locations. Then the area can be observed optically to identify the fuse. Another method involves using an optical fault injection to change the state of different transistors inside the chip and find the locations that affect the fuse state. Both methods will be discussed later in this chapter and their application for hardware security analysis will be shown in Chapter 7.

6.1.2 Toothpick attack

The term 'toothpick attack' appeared for the first time in the Xilinx paper on CPLD security [130]. It referred to our recently invented optical fault induction attacks as well as other low cost unknown attacks and attack-yet-to-come. To be more precise, by 'toothpick attack' we understand our recently invented improvement to the UV attacks.

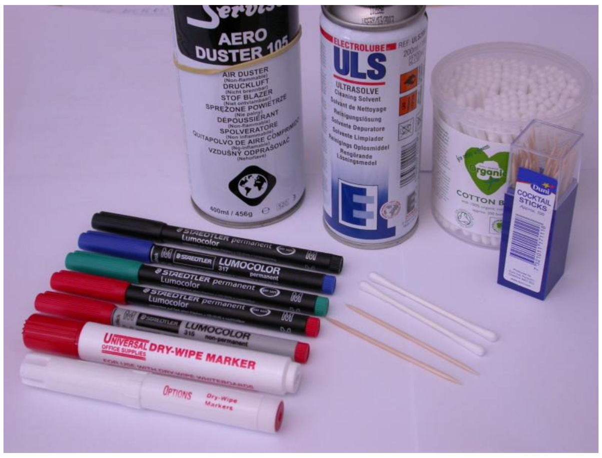


Figure 73. Tools for the toothpick UV attack

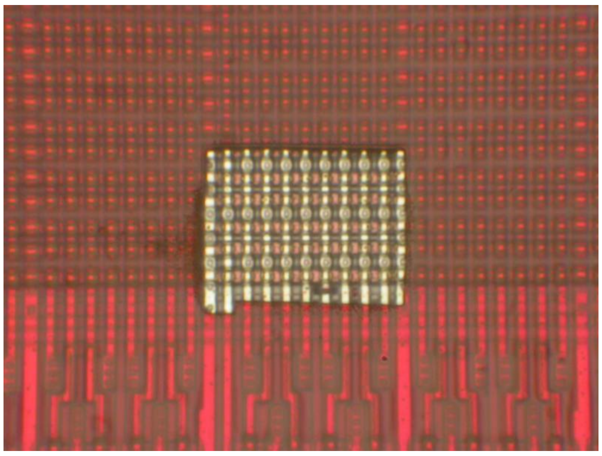


Figure 74. The precise trench in the paint was cut with a laser cutter exposing required EPROM cells. 500× magnification

Instead of using patches of a duct tape to cover different areas on the chip surface we used an ordinary marker pen to paint the chip surface and a wooden toothpick to scratch over the desired area (Figure 73). Several different markers with various colours were tested on an EPROM chip and the difference in the erase time between clear and painted chips was measured. The result is presented in the Table 4. The red marker colour is optimal when the chip surface should be seen for precise positioning and also provides reasonably good attenuation. Dry-wipe markers are easy to scratch without any danger of damaging the underneath passivation layer, but they do not provide very good attenuation, though for most applications it is enough. Some paints are slightly conductive, so care should be taken when applying them to avoid painting over the bonding pads.

The process does not require much experience and the technique can be learned in a very short time. The same trick could be used to find the security fuse location; the paint would be removed by solvent after each paint/erase cycle.

It is almost impossible to get better than 10 µm precision even if the surface is scratched under a microscope, but for some EPROM microcontrollers this should be enough as the security fuses are located separately from the main memory. To achieve a better result, the paint layer can be cut with a laser cutter (Figure 74). Ideally, the UV laser should be used, but a green laser also works well on black and red paint.

Type of marker pen	Colour	Attenuation of a single layer (~3–5 µm thick)	Scratching with toothpick	Transparancy to visible light
Staedtler, Lumocolor, non-permanent (water soluble)	Red	17 dB	Hard	Transparent
Staedtler, Lumocolor, permanent	Red	14 dB	Hard	Transparent
	Green	6 dB	Hard	Very transparent
	Blue	13 dB	Hard	Transparent
	Black	24 dB	Possible	Partially transparent
Universal Office Supplies, Dry-Wipe Marker	Red	10 dB	Easy	Partially transparent
Options, Dry-Wipe Marker	Red	5 dB	Easy	Very transparent

Table 4. UV light attenuation for different marker paints.

So, even an attacker equipped with extremely cheap tools can succeed in applying UV attacks to the required area on the chip. Only some microcontrollers, mainly EPROM-based, are susceptible to these attacks nowadays, as manufacturers started implementing various protections against UV attacks. This may involve covering the chip with a top metal layer that stops the UV light completely, using specifically designed cells that cannot be reset with UV light, and implementing UV sensors which prevent the chip from functioning once it has been exposed to UV. More examples will be discussed in Chapter 7.

6.1.3 EEPROM and Flash issues

As well as EPROM memory, most floating-gate memory devices are also susceptible to UV attack. Meantime, chip designers have more freedom in choosing different protections against such attacks. As the EEPROM and Flash cells can change their state in both directions, the obvious thing to do is to use an erased state of the cell to indicate the alarm state and a programmed state to correspond to disabled security. Minimal changes to the control logic will do the job. This is widely used in Flash microcontrollers from many manufacturers.

There are five possible ways in which the UV light could affect the floating gate memory cell:

- It changes the cell's state from programmed to erased. This affects the security fuse if the erased state corresponds to the disabled security.
- It changes the cell's state from erased to programmed. This affects the security fuse if the programmed state corresponds to the disabled security.

- It changes the cell's state from programmed or erased to intermediate. Could possibly affect the security if the reference voltage in the cell control circuit depends upon the power supply voltage.
- It shifts the threshold of the cell's transistor out of its operating level thus locking up the cell. This provides reasonably good protection against UV attacks but may allow an attacker to locate the fuse.
- It cannot shift the threshold of the cell's transistor enough to change the state.

Toothpick attacks still can be used to locate the fuse, even if it cannot be reset with UV light, unless the fuse is covered with a top metal protection layer or it is not sensitive to the UV light. Still, fault injection attacks can be used to change the fuse state, or invasive attacks to disable it permanently.

6.2 Backside imaging techniques

Visual observation under a microscope is the first step in semiconductor analysis. As feature sizes of transistors shrink each year, structures on the chip surface become more and more difficult to observe. Down to 0.8 µm technology, it was possible to identify all the major elements of microcontrollers – ROM, EEPROM, SRAM, CPU and even instruction decoder and registers within the CPU. On chips built using 0.5 µm or 0.35 µm processes, one can hardly distinguish ROM, Flash and SRAM, whereas on chips with 0.25 µm or smaller transistors, almost nothing can be seen. This is caused not only by the small feature sizes, but most of all by multiple metal layers covering the chip surface (up to eight on modern 0.13 µm chips). In addition, planarisation technology often involves filling blank spaces on metal layers with metal pads which block the optical path as well.

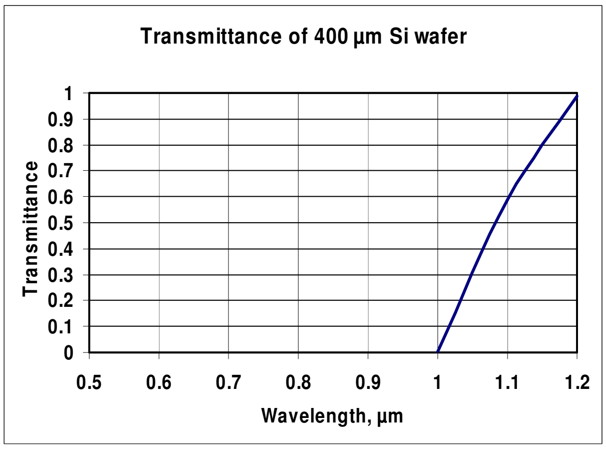


Figure 75. Transmittance of low-doped bulk silicon

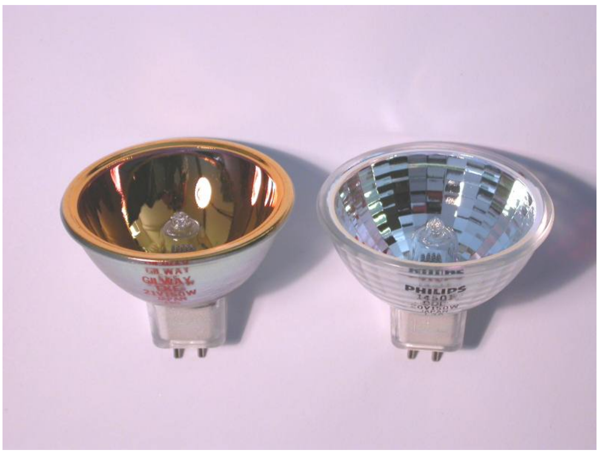


Figure 76. Halogen lamp used for IR illumination and a standard halogen lamp

One approach is to use infrared light, either reflected or transmitted, and observe the chip from its rear side. Silicon is almost transparent to photons with wavelengths >1100 nm (Figure 75).

However, highly doped silicon wafers ($>1 \cdot 10^{19}$ cm^{-3}) used in some modern chips, are less transparent to IR light [131] and more intensive light source or an IR camera with higher sensitivity is required.

For taking images in our experiments, we used a special infrared camera from Hamamatsu C2741-03C with 400–1800 nm sensitivity and 700 lines resolution [132]. A standard halogen lamp was used for illumination with a silicon wafer used as an infrared filter. Using a halogen lamp with a gold reflector (Figure 76), for example the L6409-G from Gilway [133], for reflected light illumination, increased the output of the photons in the desired region by 300%. However, it required a metal holder for the IR-pass filter inside the microscope illuminator, and more intensive cooling to prevent the optics from being damaged by the heat. Using a Mitutoyo FS60Y microscope with NIR objectives improved quality and brightness of the images (Figure 77).

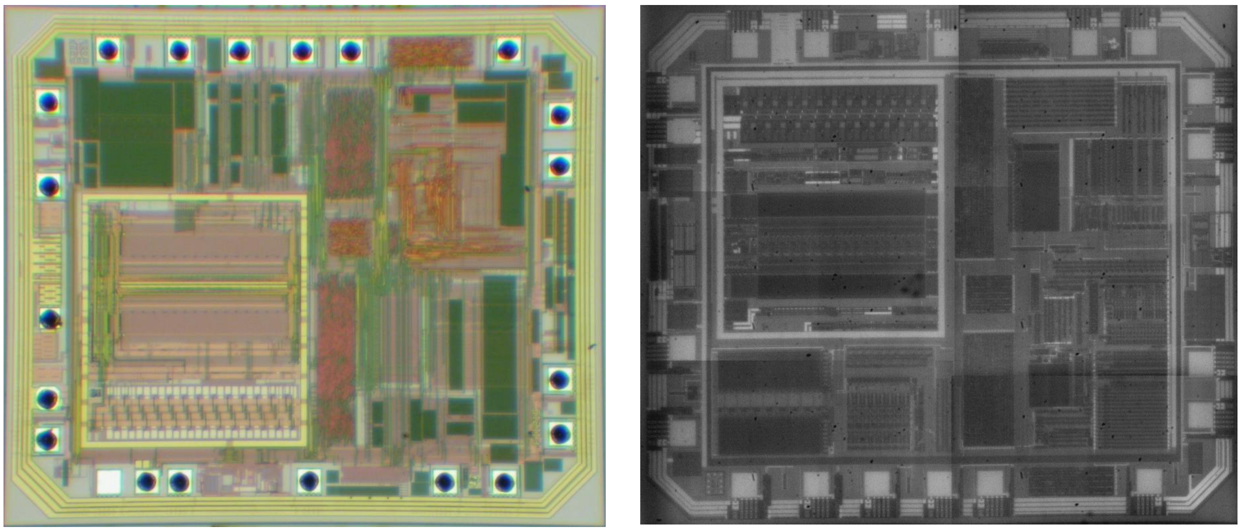


Figure 77. Standard optical image and reflected light backside image of the MSP430F112 microcontroller built with 0.35 μm technology. 50× magnification

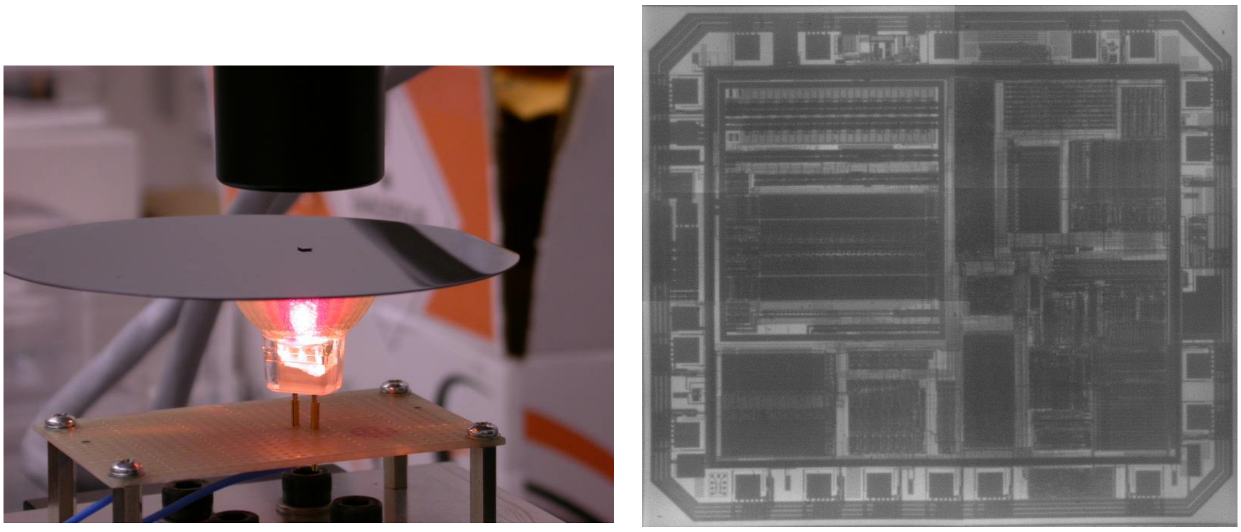


Figure 78. Transmitted light setup and image of the MSP430F112 microcontroller. 50× magnification

The images could be taken with a transmitted light setup as well (Figure 78). In this case a less intensive light source is required and even a standard low-cost monochrome CCD camera can be used. A blank silicon wafer was used to filter out the normal light.

Reflected light gives better contrast, as it does not pass through multiple metal layers. For 0.5 µm and smaller technologies, much more information is revealed than from a normal image. It should be noted that the image from the rear side is a mirror-image of the normal front side image; in our case the chip was flipped horizontally.

Another useful application of backside imaging is ROM content extraction. On the front side, transistors are shielded by the top metal layer, whereas through the rear side they are clearly visible (Figure 79). For example, instead of using an invasive attack such as a chemical etching to extract the contents of the Mask ROM, direct observation from the rear side can be used, which is a semi-invasive technique. There are some limitations to this approach caused by the maximum resolution of the optical system in the NIR region (1000–1100 nm for Si) which cannot be better than 0.6 µm.

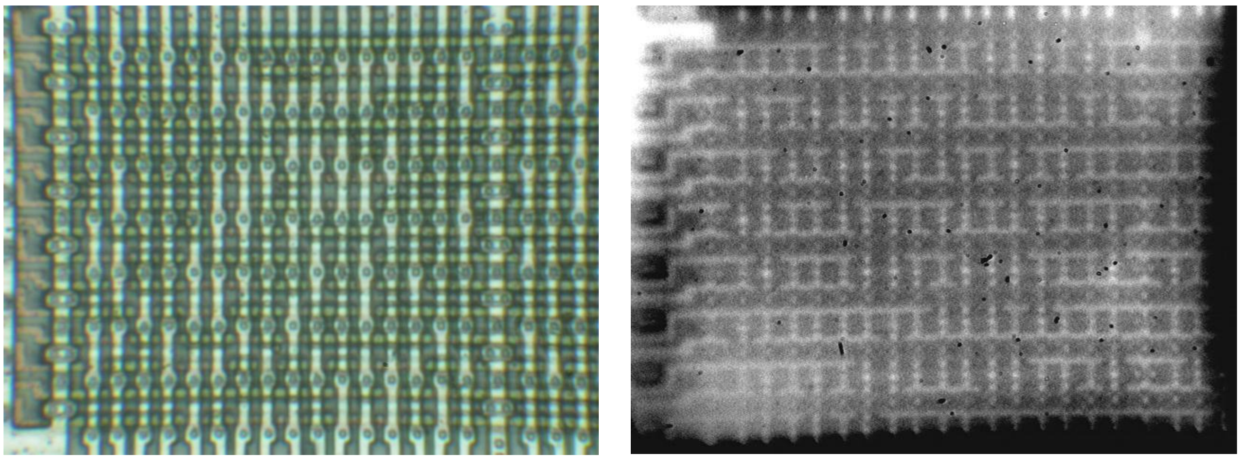


Figure 79. Standard optical image and reflected light backside image of the Mask ROM inside MC68HC705P6A microcontroller built with 1.0 µm technology. 500× magnification

Backside imaging is widely used in failure analysis tasks, from locating the failures in transistors or interconnections to navigation during a FIB work. There are special microscopes designed for such applications, for example the BEAMS V-2000 from Hypervision [134]. Needless to say, such systems cost a tremendous amount of money and can only be afforded by relatively large companies. Yet, low budget laboratories could use NIR extended microscopes with IR-sensitive video cameras, as we used for our experiments.

6.3 Active photon probing

Once the semiconductor transistor had been invented, it was found to be more sensitive to ionizing radiation – whether caused by nuclear explosions, radioactive isotopes, X-rays or cosmic rays – than the thermionic valves (vacuum tubes) used previously. In the middle sixties, during experiments with pulsed lasers, it was found that coherent light causes some similar phenomena. Lasers started to be used to simulate the effects of ionizing radiation on semiconductors [135].

Since then the technology has been improved dramatically. Expensive inert-gas-based lasers and solid-state lasers have been replaced with low-cost semiconductor lasers. As a result, the technology has moved from the laboratory all the way down to consumer electronics.

Laser radiation can ionize an IC’s semiconductor regions if its photon energy exceeds the semiconductor band gap (>1.1 eV or λ<1100 nm). Laser radiation with 1.06 µm wavelength (1.17 eV photon energy) used in [136] has a penetration depth of about 700 µm and provides good spatial ionization uniformity for silicon devices. However, its focusing is restricted by dispersion to several micrometers, and this is not precise enough for modern semiconductor devices. However, when moving from infrared to visible light, photon absorption dramatically increases [137], and it has become possible to use red and green lasers as the transistors in modern chips became thinner. Smaller devices also mean that less energy is required to achieve the same level of ionization.

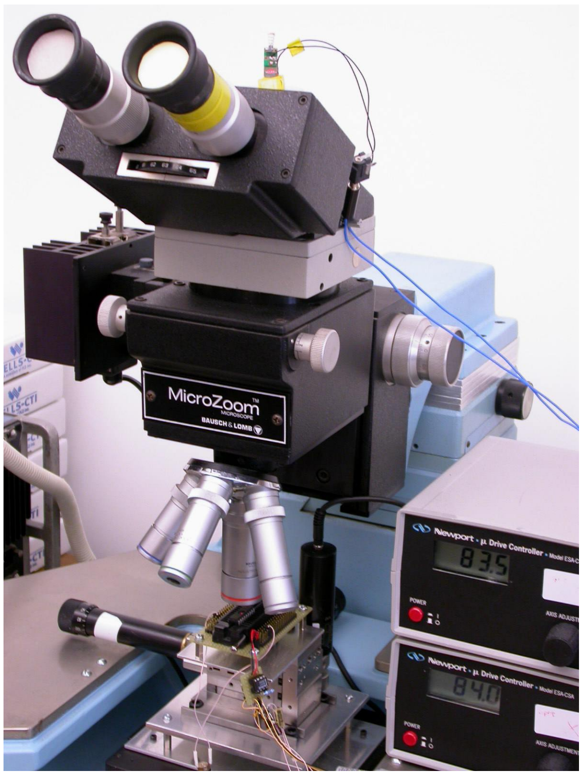


Figure 80. Laser scanning setup with the laser pointer on top of the microscope and a moving sample under it

In the case of CMOS devices, there is a danger of latching up the circuit, causing an open circuit which results in permanent damage. So the use of radiation with CMOS structures must be done with appropriate precautions.

In active photon probing, a scanned photon beam interacts with an IC. Photons with energies greater than the band gap of silicon generate electron-hole pairs in the semiconductor. Photons with lower energies can still interact with p-n junctions, but with only a heating effect taking place, which is significantly weaker than the photovoltaic effect.

There are different scanning techniques used for photon probing in failure analysis [67]. As a photon source they normally use a laser scanning microscope. Although such microscopes have a big advantage of fast scanning – about one frame per second, their price is too high for small research labs. Therefore, for our experiments we used the less expensive but much slower approach of a stationary laser source and a sample moved on an X-Y motorised stage (Figure 80). The laser was attached to the camera port of the probing station's microscope, and the test socket with chip was mounted on the Newport 561D-XYZ stage [138] with AD-100 actuators [139]. Although scanning a 100 µm × 100 µm area takes about 15 minutes, this is still suitable for research purposes.

6.3.1 Laser scanning techniques

There are two major laser scanning techniques which can be used for hardware security analysis. One is called optical beam induced current (OBIC) and is applied to an unbiased chip to find the active doped areas on its surface [140]. Another, called light-induced voltage alteration (LIVA), applied to a chip under operation [141]. In OBIC, photocurrents are used directly to produce the image. For that, the analysed chip is arranged with its power supply pin connected to a current amplifier and the values are registered on the computer via an acquisition board. In LIVA, images are produced by monitoring the voltage changes of the constant current power supply as the optical beam is scanned across the IC surface.

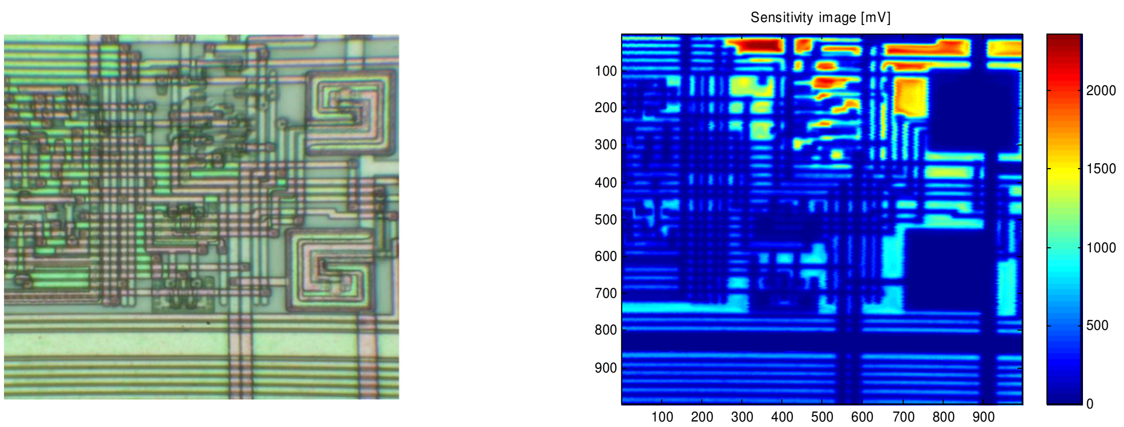


Figure 81. Optical image (500× magnification) and laser scan of the security fuse area in PIC16F84A

The OBIC technique can be used in addition to standard optical imaging, as it allows us to locate the active areas inside a chip. We used Matlab software to draw a graphical representation of the result after scanning. As with optical imaging, the laser can be focused on the active areas of the chip from both front and rear. For front-side scanning we used a red laser pointer (~650 nm wavelength) as a light source and the result of scanning the security fuse area inside the Microchip PIC16F84A microcontroller is shown in Figure 81. For backside operation we used infrared laser module with 1065 nm wavelength. The same area on the chip scanned from the rear side is shown in Figure 82. As can be observed, the image is more informative compared to the front-side scan because metal wires do not stop the laser beam.

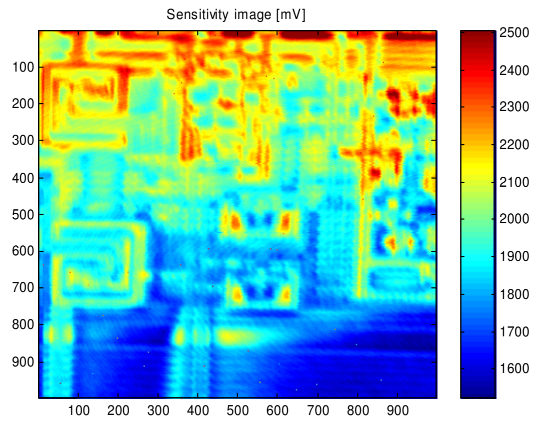


Figure 82. Backside laser scan of the security fuse area in PIC16F84A

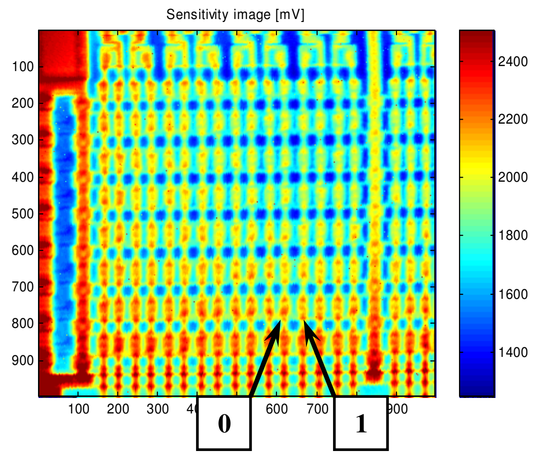


Figure 83. Backside laser scan of Mask ROM inside MC68HC705P6A microcontroller

Laser imaging can be used for Mask ROM extraction in a similar way to the IR backside imaging (see Figure 79). The scan of the same Mask ROM area inside MC68HC705P6A microcontroller is presented in Figure 83.

6.3.2 Reading the logic state of CMOS transistors

The traditional way of reading out data from semiconductor memories involves an invasive attack using mechanical probing, usually of the processor's bus [8]. Such attacks involve making direct electrical connections to internal components using microprobes. This is becoming more difficult for a number of reasons, ranging from shrinking feature sizes to the use of hardware access control circuits for the on-chip memory. We decided to investigate whether semi-invasive techniques could be used to read out the state of a memory cell in a non-destructive way. The technique described below was used to extract the contents of the CMOS SRAM but could have much wider applicability [89].

To analyse the structure of SRAM memory we used a red laser focused on the chip surface using a microscope. As photons from the red laser (650 nm wavelength) have energy larger than the silicon band gap, they will ionize active areas inside the chip. If the photons reach an area near p-n junctions, a photocurrent will be produced due to the photovoltaic effect. When the photons hit the p- or n-channel area, this will decrease the resistance of the channel by injecting free carriers.

The fact that enables us to read a memory cell's state is that the increase in current is noticeable for closed channels, and almost negligible for open channels. Thus, by aiming the laser beam at an appropriate transistor or transistors, we can distinguish between the two possible memory states.

In our first experiment, we built a map of the active areas in a microcontroller by measuring the photocurrent induced by laser scanning the chip surface. The chip was mounted on an X-Y motorized stage with 0.1 µm resolution. The result of the scan is shown in Figure 84. The active areas can be seen as they produce higher current, but most of the chip is covered with metal layers which the laser cannot penetrate, so these areas do not produce any current. We used this picture as a reference for the results obtained from a powered chip.

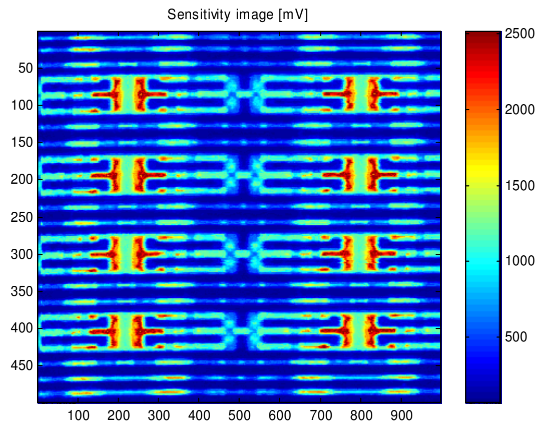


Figure 84. Laser scan of unpowered memory

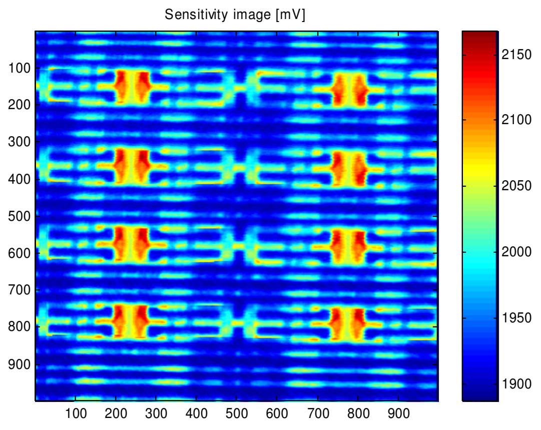


Figure 85. Laser scan of powered-up memory with state

Our next experiment was done with an operating chip. It was programmed to allow us to upload any value into its RAM and then stop the chip operation. The result of the scanning with memory cells loaded with random data is shown in Figure 85. It can be seen that memory cells have different states: where the cell holds a '1' the top is brighter, and where it is a '0' the bottom is.

Our experiments are somewhat similar to results published by Sandia Labs [141], but with a number of differences. They were done without using extremely expensive laser scanning microscopes; we scanned the chip from its top side; and instead of sending constant current through the chip, we used a constant voltage supply and measured current as in standard power analysis [84].

Other techniques which can be used to extract the information from memory are IR modulation-based probing [142] and a recently introduced technique that uses an eddy current attack to read the memory [89].

If valuable data are present in the clear within memory for just one clock cycle in a location that an attacker can deduce, and the state can be frozen (whether physically, using low temperature, or by some other means such as stopping the clock), then it is likely to be possible for an attacker to read this data out using optical or electromagnetic probing techniques. The investment in skills and equipment required to carry out such attacks is significantly lower than that needed for full invasive attacks. Hardware countermeasures will thus be necessary for any processors that are required to resist this capability of attacker.

6.4 Fault injection attacks

Here we describe a new class of attacks on secure microcontrollers and smartcards. Illumination of a target transistor causes it to conduct, thereby inducing a transient fault. Such attacks are practical; they do not even require expensive laser equipment. We have carried them out using a flashgun bought second-hand from a camera store for £20 and with a £5 laser pointer. As an illustration of the power of this attack, we developed techniques to set or reset any individual bit of SRAM in a microcontroller. Unless suitable countermeasures are taken, optical probing may also be used to induce errors in cryptographic computations or protocols, and to disrupt the processor's control flow. It thus provides a powerful extension of existing glitching and fault analysis techniques. This vulnerability may pose a big problem for the industry, similar to those resulting from probing attacks in the middle nineties and power analysis attacks in the late nineties.

6.4.1 Changing SRAM contents

Although there are many publications about using pulsed lasers to simulate ionizing radiation, we could find no published information about using them to control or change the behaviour of integrated circuits. So we decided to apply an intense light source to a semiconductor chip, and particularly to CMOS logic, to see whether it would be possible to change the state of a memory cell and how easy, or difficult, it might be.

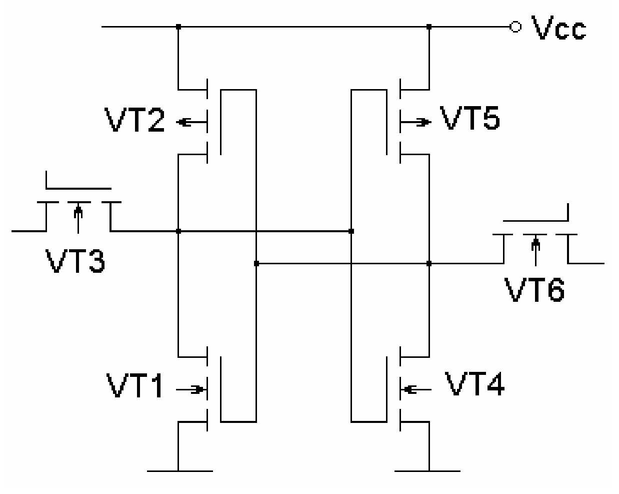


Figure 86. The architecture of an SRAM cell

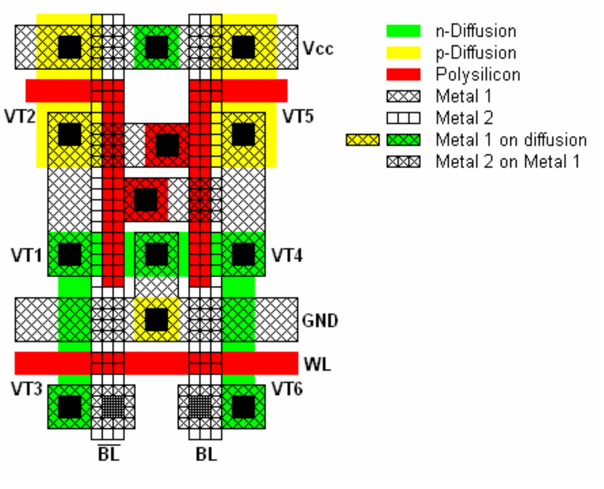


Figure 87. The layout of an SRAM cell

Our first experiments targeted SRAM. The structure of a standard six-transistor SRAM cell is shown on Figure 86 [143]. Two pairs of p- and n-channel transistors create a flip-flop, while two other n-channel transistors are used to read its state and write new values into it. The layout of the cell is shown on Figure 87. The transistors VT1 and VT2 create the CMOS inverter; together with the other similar pair, they create the flip-flop which is controlled by the transistors VT3 and VT6.

If the transistor VT1 could be opened for a very short time by an external stimulus, then it could cause the flip-flop to change state. By exposing the transistor VT4, the state of the cell would be changed to the opposite. The main difficulties we might anticipate are focusing the ionizing radiation down to several μm^2 and choosing the proper intensity.

For our experiments we chose the Microchip PIC16F84 microcontroller [91], which has 68 bytes of on-chip SRAM memory. A standard depackaging procedure was applied to the chip as explained in Chapter 5 and the result of this operation is represented on Figure 88. The SRAM memory array is located in the middle of the bottom of the chip die. This area is shown with 50× magnification on Figure 89.

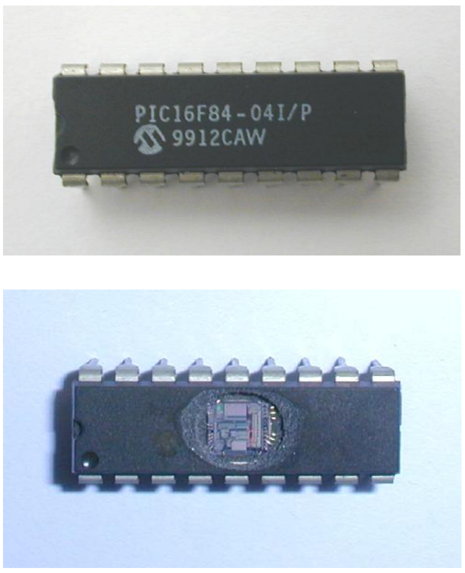


Figure 88. Original and decapsulated PIC16F84 microcontroller

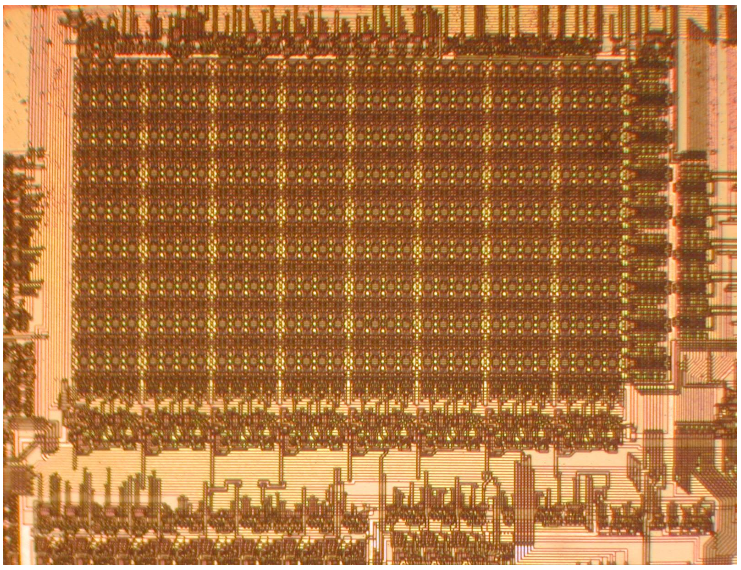


Figure 89. SRAM memory array with 50× magnification

Because we had a very limited equipment budget, and the laser we had appeared unsuitable, we decided to use a cheap photoflash lamp (a Vivitar 550FD, bought secondhand from Campkins' camera shop for £20). Although the luminosity of a flashlamp is much less than that of a pulsed laser, with appropriate magnification the necessary level of ionization might be achieved. We used duct tape to fix the photoflash lamp on the video port of a Wentworth Labs MP-901 manual probing station (Figure 90). Magnification was set to the maximum – 1500×.

The microcontroller was programmed to upload and download its memory. By filling the whole memory with constant values, exposing it to the flash light, and downloading the result, we could observe which cells changed their state.

By shielding the light from the flash with an aperture made from aluminium foil, we succeeded in changing the state of only one cell. The output power of the lamp was set to the maximum. The final state of the cell depended on the area exposed to the flash. This confirmed our intuition that it would be possible to change the contents of SRAM using a low cost semi-invasive attack.

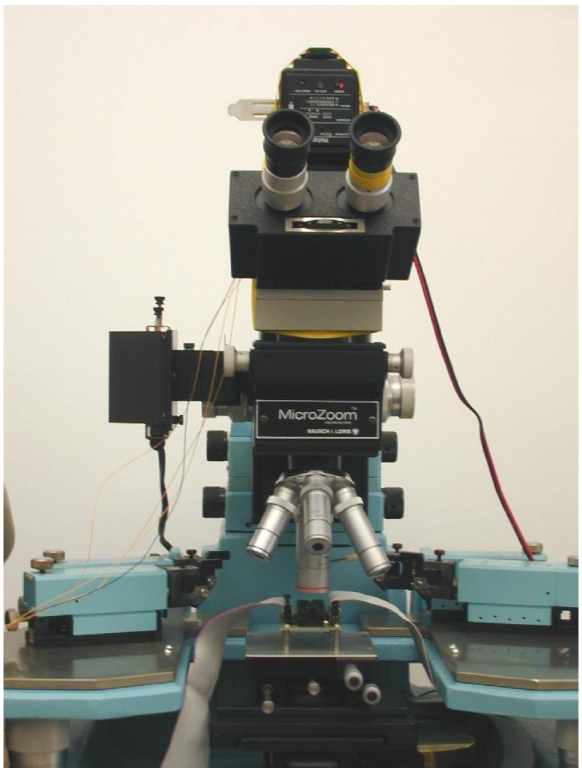


Figure 90. Wentworth Labs MP-901 manual prober with Vivitar 550FD photoflash lamp mounted on top

We found we could change any individual bit of an SRAM array. The array, under maximum magnification, is shown in Figure 91. Focusing the light spot from the lamp on the area shown by the white circle caused the cell to change its state from '1' to '0', with no change if the state was already '0'. By focusing the spot on the area shown by the black circle, the cell changed its state from '0' to '1' or remained in state '1'.

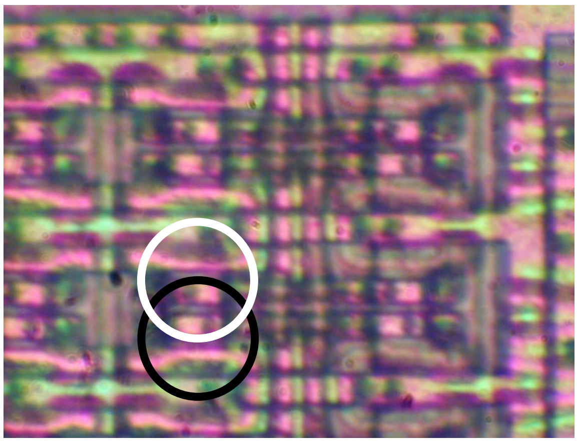


Figure 91. SRAM memory array with maximum magnification (1500×)

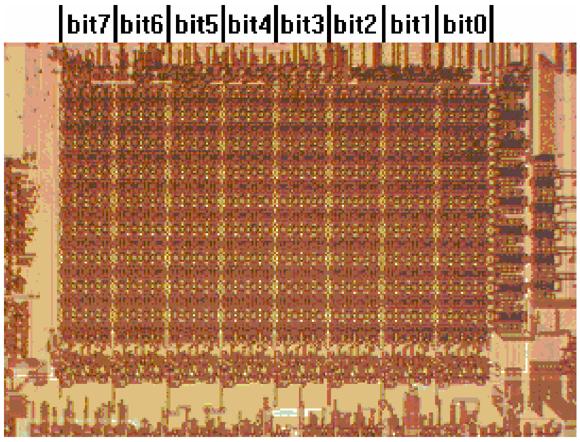


Figure 92. Allocation of data bits in SRAM memory array

It can be seen from Figure 89 that the SRAM array is divided into eight equal blocks. By exposing cells in different blocks, we found that each block corresponds to one bit plane of information. The resulting bitplane map is shown in Figure 92.

We built a full memory map by exposing each cell in turn to the photoflash light. The result is presented in Figure 93, with the left edge corresponding to the bottom side of the block. It can be seen that the addresses are not sequential, but divided into three groups.

30h	34h	38h	3Ch	40h	44h	48h	4Ch	10h	14h	18h	1Ch	20h	24h	28h	2Ch	0Ch
31h	35h	39h	3Dh	41h	45h	49h	4Dh	11h	15h	19h	1Dh	21h	25h	29h	2Dh	0Dh
32h	36h	3Ah	3Eh	42h	46h	4Ah	4Eh	12h	16h	1Ah	1Eh	22h	26h	2Ah	2Eh	0Eh
33h	37h	3Bh	3Fh	43h	47h	4Bh	4Fh	13h	17h	1Bh	1Fh	23h	27h	2Bh	2Fh	0Fh

Figure 93. Allocation of addresses in each bit block of SRAM memory array

This shows how simple semi-invasive attack methods can be used for reverse engineering a memory address map. The only limitation is that the flash does not produce even and monochromatic light, so it is very difficult to control the area where the spot of the light will be applied. This problem can be solved by replacing the flash with a suitable laser.

This work shows that optical probing attacks are possible using low-cost equipment. We have repeated the experiments using a laser pointer, which we bought in Cambridge market for £5, and a motorized stage. The same results were achieved, but there were several practical differences. On the one hand, we could probe the chip surface automatically, and at a rate which we estimate could be driven as high as 100 flashes per second. On the other hand, we had to be more careful with alignment because of the narrower aperture and lower power. (The pointer is described as being eye-safe because it is 'less than 5mW'; we can focus it to about a 1 μm spot on the chip surface and its wavelength is around 650 nm.)

We used our automated probing equipment to implement attacks on a number of semiconductor devices. The best designed of the modern secure microcontrollers are not vulnerable to attacks using single laser flashes, as their protection state depends on a number of bits of physical storage. However, a number of designs can be unprotected by changing the state of the flip-flop that latches the read-protect state. We strongly recommend that designers of ICs should study their designs carefully to ensure that there are no single-transistor failures that can subvert the chip's security policy.

Attack experiments have been conducted on smartcards too. It may be helpful at this point to recall some of the earlier literature on fault analysis. In [9], Boneh, Demillo and Lipton pointed out that the faulty computation of an RSA digital signature leaks the signing key. For example, when doing an RSA signature the secret computation $S = h(m)^d \pmod{pq}$ is carried out mod p, then mod q, and the results are then combined, as this is significantly faster. However, if the card returns a defective signature S_p which is correct modulo p but incorrect modulo q, then we will have $p = \gcd(pq, S_p^e - h(m))$.

In [10], Anderson and Kuhn pointed out that interference with jump instructions is an even more powerful and general attack: an attacker who can cause conditional branches in the smartcard code to be taken wrongly may, for example, reduce the number of rounds in a block cipher to one or two, making key recovery straightforward. The first of these two types of attack has been implemented successfully using our technique, but a non-disclosure agreement prevents us from giving further information.

The optical probing attack described above is a new and powerful technique for attacking smartcards and other security processors. We anticipate that, like the power analysis attacks reported by Kocher in [84], it could have a significant commercial effect on the industry, in that it will force a thorough reappraisal of security claims and the introduction of new defensive technology.

Existing high-end chip-defense techniques, such as top-layer metal shielding and bus encryption, may make an attack using these techniques more complicated, but are not enough. A sophisticated attacker can defeat metal shielding by using infrared light or X-rays, while bus encryption can be defeated by attacking registers directly.

There are many effects that can be used by an attacker. We have described here in detail how the illumination of a certain area of an SRAM cell can be used to set it to either ‘0’ or ‘1’. This is only the beginning. Given only moderately expensive equipment, an attacker may be able to induce a fault in a CMOS integrated circuit, in any targeted transistor, and at precisely the clock cycle of his choice. This could represent a very serious threat unless special countermeasures are used.

6.4.2 Non-volatile memory contents modification

EPROM, EEPROM and Flash memory cells are even more sensitive to fault injection attacks. This happens because the currents flowing inside the floating gate cell are an order of magnitude smaller that inside the SRAM cell.

It should be mentioned that our research on active optical attacks began when I was microprobing an SLE66 smartcard on our probing station and realised that the answer-to-reset (ATR) and user memory contents became changed if the light under microscope was left switched on. Our first assumption was that the chip has built-in light sensors to prevent decapsulation, but after we investigated the problem it appeared to be the light injected faults into the EEPROM on-chip memory.

The light from the 20 W halogen bulb installed inside our probing station microscope’s illuminator is also enough to flip the state of the security fuse in the PIC16F84 microcontroller when the diaphragm is fully opened and magnification is set to the maximum of 1500×. In this case the light can be focused down to the security fuse area without affecting the main memory. No modification to the existing equipment was required. A programmed, secured and decapsulated PIC16F84 chip was placed in the test socket under the microscope and connected to a universal programmer. Once the location of the security fuse was found through the microscope eyepiece, the light was switched to the maximum and the memory content was read in a normal way.

The same operation was successfully performed from the rear of the chip as well. On the one hand, it is not easy to find the location of the fuse from the rear side of the die; on the other, rear-side decapsulation can be done mechanically to avoid use of aggressive chemicals. The location can be found either by using the top-side optical image as a reference or with the help of an infrared camera.

Although the light from the halogen lamp does not have enough IR component in the 1000–1100 nm region to ionize the active areas of transistors directly from the rear side, most of the electron-hole pairs created in the volume of the silicon substrate migrate down to the active areas before they recombine with each other.

Fortunately, this attack does not work on modern chips built with smaller features, for a number of reasons. The top-metal layers prevent the light from reaching the active areas; planarisation reduces the transparency of the oxide layers and also diffuses the light; higher doping concentrations reduce penetration of the light from the rear; and smaller transistors require a higher dose of radiation for switching.

UV EPROM memory cells are less sensitive to light attacks because a higher current is required to switch the state of the memory transistor. Nevertheless, the fact that a photoflash or even a common light bulb can be used to break the security should force developers to design security protection more carefully. For example, security fuses should not be placed far away from the main memory, as this makes it easier to attack them. Also, light and radiation sensors could be used to detect ionizing radiation and reset the circuit before it leaks sensitive information.

6.5 Modelling the attacks

It is very important to understand how specific attacks work, in order to develop effective countermeasures. In the case of optical fault injection and laser scanning attacks, a full understanding of the mechanisms and processes taking place inside the silicon chips is necessary. There are very limited ways in which such attacks can be simulated. As each process requires a complex system of differential equations to be solved, it is very difficult to calculate the effect of a laser beam on the whole chip. But if applied to a small group of transistors, simulation is achievable. One of the software tools which allows emulation of the effect of ionizing radiation on semiconductors is TCAD [144]. As this software tool is very complex and expensive we chose another tool “DIODE-2D” which was specifically developed for simulation of local and bulk ionization in semiconductor devices [145]. The simulation was done in collaboration with Specialised Electronic Systems and “DIODE-2D” is their proprietary product.

6.5.1 Modelling the logic state reading

The state of a CMOS transistor can be read optically using laser scanning, as shown earlier in this chapter. As expected, the highest photocurrent is produced when the laser beam hits the source and drain of p- and n-channel transistors inside the memory (see Figures 84 and 85). The lowest photocurrent corresponds to the opaque metal lines. The results of the scanning show noticeable variation in photocurrent between opened and closed inverters inside the SRAM cell. This can be used to determine the current state of the memory cell.

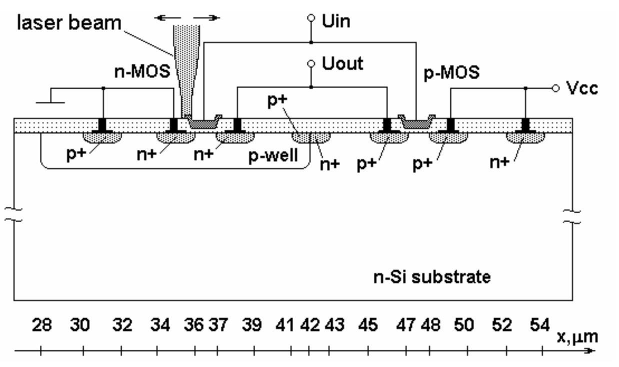


Figure 94. Cross-section of the analysed CMOS inverter

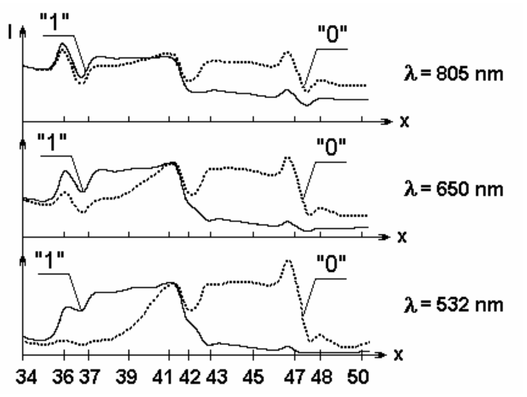


Figure 95. Power supply current dependence from the laser beam position for different wavelengths

To interpret the results and find more suitable parameters for scanning, two-dimensional mathematical modelling of the laser pulse applied to a CMOS inverter was performed using "DIODE-2D". A cross-section of the modelled structure is shown in Figure 94. The length of the channels was assumed to be 1 µm and the intensity of the laser radiation $1{\cdot}10^4$ W/cm^2. Other parameters, such as doping concentrations and depth of p- and n-areas, were taken from standard average for 1 µm bulk CMOS technology with n-type substrate.

We simulated the dependence of the power supply current on the laser position for different wavelengths of the laser and two states of the inverter. The result is shown in Figure 95. It can be seen that exposure of the closed transistor results in higher current than for the opened one. Opening the closed transistor channel increases the total current more than slightly decreasing the effective resistance of the already open channel.

Another noticeable effect is the dependence of the photocurrent on the wavelength of the laser. This is caused by reduced parasitic photocurrent in the substrate-well junction; photons with shorter wavelengths are mostly absorbed near the surface. The higher effectiveness for shorter wavelengths is also confirmed by computations of power-supply current dependence on the laser wavelength for the same CMOS inverter (Figure 96). The shorter the wavelength, the higher the supply current difference between '0' and '1' states. This was confirmed experimentally. With a 650 nm laser, it was possible to recognise the difference in supply current for different states of the inverter, while with an 805 nm laser the difference was close to the noise level. The main noise source is the substrate-well photocurrent, which also depends on the doped regions above it. Shifting to a shorter wavelength reduces its influence.

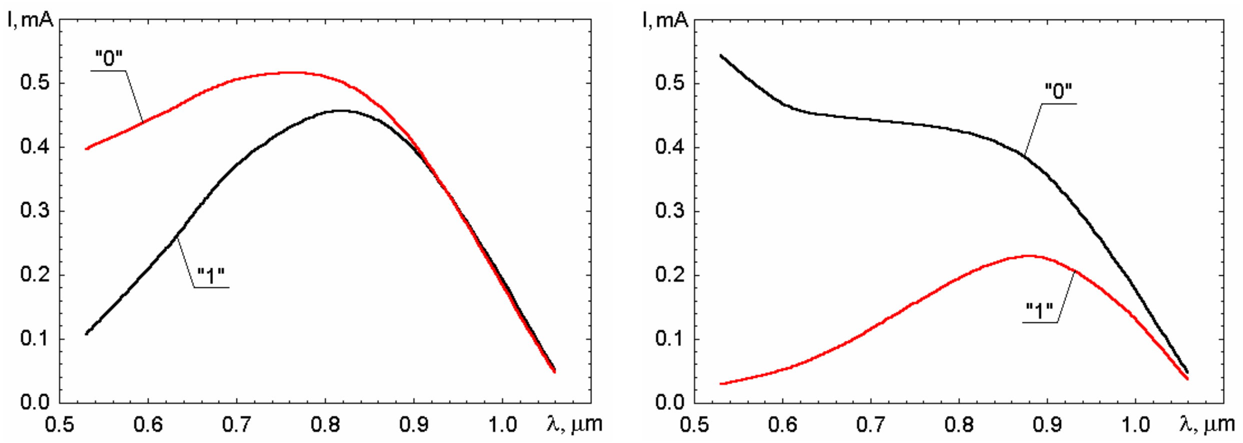


Figure 96. Power supply current dependence from the laser wavelength for two logic states in p-MOS (left) and n-MOS (right) transistors inside the CMOS inverter

The results of the modelling exercise confirm the possibility of using a focused laser beam for reading the state of a CMOS inverter. These results were obtained after we had done the experiments. However, they taught us to choose the appropriate wavelength for the laser.

6.5.2 *Modelling the fault injection attack*

As discussed above, focused laser radiation can also be used to change the state of SRAM cells and CMOS inverters. The process was also simulated using the "DIODE-2D" software tool. The cross-section of the analysed structure is shown in Figure 97. This is a typical CMOS inverter with 1 μm transistor channels and n-type substrate. Doping concentrations were assumed as $1.5{\cdot}10^{15}$ cm^{-3} for the substrate and $5.0{\cdot}10^{15}$ cm^{-3} for the p-well. Modelling was done for +3.3 V power supply and for two wavelengths of the laser – 532 nm and 1064 nm.

The results of previous computations showed that the output voltage has the maximum change when either closed p-MOS (logic '0') or closed n-MOS (logic '1') transistor is exposed to the laser radiation. The areas of interest used for further modelling are shown in Figure 97.

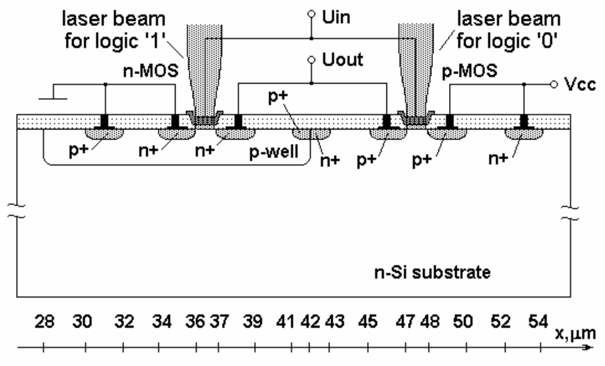


Figure 97. Cross-section of the analysed CMOS inverter

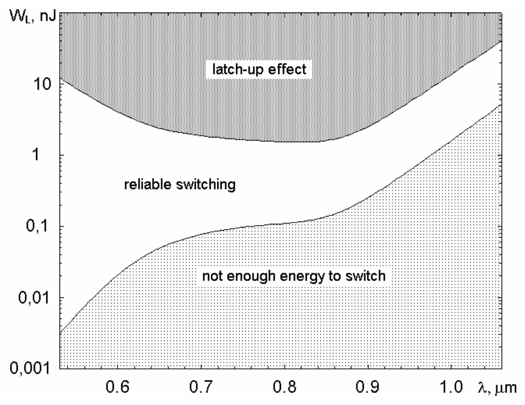


Figure 98. Power supply current dependence from the laser beam position for different wavelengths

Modelling revealed that the major factor that limits the ability of the laser to change the state of a CMOS inverter is a radiation induced latch-up effect caused by ionization of the parasitic four-layer structures in a CMOS circuit. The state of the CMOS inverter can be changed only if the intensity of the laser radiation necessary to switch it is less than the threshold level of any parasitic four-layer structures inside it. Focusing the laser radiation in the channel area of closed transistor is optimal for switching, but cannot eliminate latch-up because electron-hole pairs generated by the laser radiation quickly fill the volume of the inverter and create ionization currents in all the p-n junctions inside it.

The simulation result for an 11 ns laser pulse applied to a CMOS inverter in logic ‘0’ state is presented in Figure 98. The laser pulse was applied to the 10 μm^2 area above the p-MOS transistor gate. It was assumed that the switching takes place if the output voltage become higher than the minimum input voltage for logic ‘1’ which is 2 V for a 3.3 V power supply. This may be a bit excessive as normally the threshold level of a CMOS inverter is around 0.5 V_{CC}, but without the exact information on the threshold voltages inside the IC, our assumption guarantees switching the element next to the inverter, or flipping the CMOS SRAM cell.

As can be observed from the results of the simulation, there is a large area corresponding to laser energies sufficient to switch the inverter without causing the latch-up effect. This difference becomes larger for shorter wavelengths of the laser. This is caused by increased absorption of the shorter wavelength laser in silicon, which increases the number of electron-hole pairs under the transistor’s gate. On the one hand, it increases the modulation of the channel; on the other, it makes switching the parasitic four-layer structures more difficult by reducing the ionization currents in n-MOS transistors.

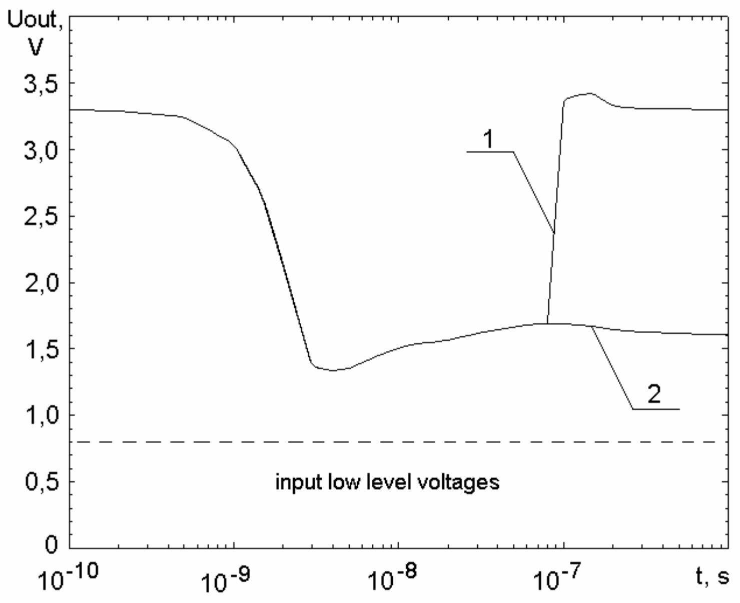


Figure 99. Change of the output voltage of CMOS inverter after 0.35 nJ (1) and 0.36 nJ (2) laser pulses

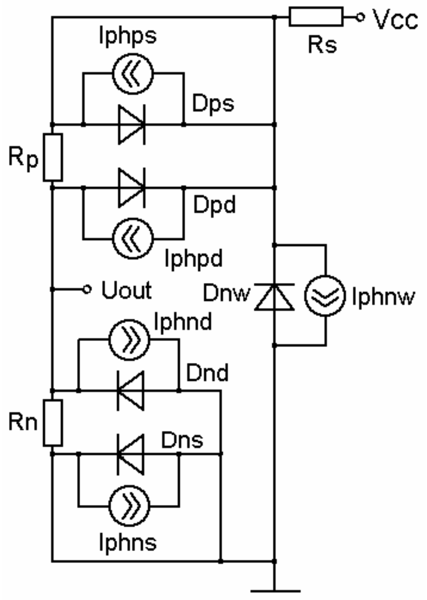


Figure 100. Equivalent circuit of a CMOS inverter during inozation

However, computations made for a CMOS inverter in logic '1' state showed that there is no area of reliable switching for any wavelengths of the laser between 532 nm and 1064 nm. Parasitic four-layer structures can be switched at energies which are not enough to shift the output voltage below the minimum level for logic '0' state (0.8 V for 3.3 V power supply).

As an example, the time-dependence of the output voltage after an 11 ns 532 nm laser pulse for energies below and above the latch-up threshold is shown in Figure 99. It can be noticed that even for energies close to the threshold level, the output voltage does not go below the required minimum necessary to reliably switch the next transistor. Switching is possible only if the threshold voltage of the next cascade is above 1.4 V which cannot be guaranteed.

The reason for the behaviour is that the ionization current of the p-MOS drain area I_{phpd} is not limited by the p-well, and significantly exceeds the ionization current of the n-MOS drain (Figure 100). This leads to an increase of the output potential, and prevents the output voltage reaching the required minimum.

For CMOS technology with a p-substrate, the behaviour is expected to be the opposite. For any CMOS technology, the optimal results are achieved when the gate area of the previously closed transistor is exposed to a short wavelength.

Chapter 7

Hardware security analysis

This chapter shows how the technologies introduced in previous chapters can be used for hardware security evaluation of semiconductor devices. We concentrate on semi-invasive technologies as, on the one hand, they give results faster than non-invasive attacks, and on the other they require less capital investment than invasive attacks. We do not claim that this is an exhaustive methodology as each new device, especially if built with new technology, requires a different approach, and hence it is practically impossible to give a universal evaluation technique for security analysis. Therefore, only some ideas are presented here.

7.1 Evolution against UV attacks

UV attacks were successfully used against many generations of microcontrollers from different manufacturers. We decided to investigate what has been done to different microcontrollers in order to protect against these attacks.

Over twenty years ago, very little attention was paid to protection against invasive attacks in microcontrollers. Then, microcontrollers had either Mask ROM or UV EPROM for program storage. The EPROM-based microcontrollers usually had a security fuse to protect against unauthorised access to the memory. Different implementations for the security protection fuse were discussed in Chapter 2. Here we give some examples and ways of protection against UV attacks in microcontrollers and also estimate the defence level against these attacks.

Typical examples of microcontrollers without any protection against UV attacks are the Microchip PIC16C57 and PIC12C509. The security fuse can be found relatively easy in a few hours time by exposing different areas on the chip to UV light for 10–15 minutes and observing the result. Because the fuse is located far away from the program memory, it can be easily defeated by covering the EPROM with an opaque material like duct tape. Placing the fuse closer to the memory makes finding it harder but the protection can still be defeated relatively easy. Some fuses have a cell structure very similar to that of the main memory as in the Cypress CY7C63001A microcontroller which is widely used in secure USB dongles (see Figures 3 and 11). By preparing a good digital photo of the chip surface and searching for the same pattern as the main memory cells have, it is not very difficult to find the fuse. Although the fuse is located close to the EPROM memory, it is still possible, with some experience, to shield the EPROM from the UV light. Some manufacturers place the security fuses in the same array as the main memory, making it extremely hard to selectively erase the fuse without disturbing the main memory (examples were given in Chapter 2). However, some low cost attacks against such implementations were discussed in the previous chapter.

Once microcontroller manufacturers realised that their products could be easily attacked with UV light, they got smarter. Some EPROM-based microcontrollers use memory encryption. For example, the Philips 87C51 microcontroller [146] has a 64 byte encryption table in addition to the security fuse protection. Even if the security bit is reset (with UV light or somehow else), the program memory value will be read XNOR with the corresponding byte from the table. In the Microchip PIC16C61 and PIC16C71 microcontrollers [147], the memory content is encrypted when the security bit is activated. The encryption involves XNOR of the seven high order bits of the memory location with the seven low order bits. Neither encryption schemes is much good, as an attacker can erase part of the memory using toothpick attacks and then restore the memory or encryption table. For old PIC microcontrollers, the attacker did not have to find and reset the security fuse, as the encrypted memory contents can be read in a standard way using a suitable programmer.

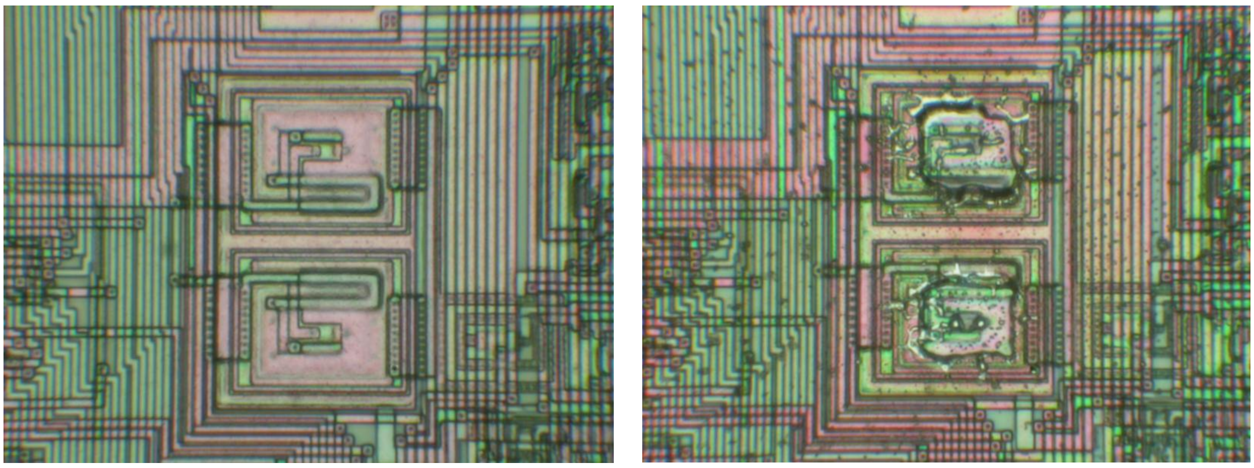


Figure 101. CP fuses in the PIC16C622A microcontroller and the top metal layer removed with a laser cutter

Further improvement to the protection of EPROM-based microcontrollers involved covering the fuses with a top metal layer opaque to UV. For example, most OTP PIC microcontrollers with 14-bit and 16-bit cores benefit from this. This not only prevents the fuses from being reset, but makes finding them more difficult. Until very recently, when we introduced fault injection attacks, there were only two practical ways of defeating the protection in such microcontrollers. One is reverse engineering followed by laser cutter or FIB treatment to remove the top metal protection layer (Figure 101). Another is modifying the CPU and microprobing the data bus. Both ways require substantial investments in equipment and a highly skilled attacker.

When EEPROM memory started to be used in microcontrollers, it allowed more protection possibilities. In addition to the top metal, inverted memory cells were used which are less sensitive to UV light. For example, in the Atmel AT89C51 microcontroller [148], security fuses cannot be reset with UV light because the erased fuse corresponds to active security. However, as the UV can change the fuse from the non-secure state to secure, it can still be used to find the fuse. To prevent this, some microcontrollers benefit from both inverted fuses and top metal protection, for example the Microchip PIC16F628 [149] and PIC16F876 [36] microcontrollers.

Certain implementations have potential security flows. For example, Ubicom SX microcontrollers have a security fuse which cannot be reset with UV; protected memory is read as the XOR of the four high order bits with the four middle and the four low bits. By using the toothpick attacks described here it is possible to recover the memory contents. The attacker has to erase, say, four high bits, then read the memory, then erase four middle bits and read it again to get four low bits.

7.2 Semi-invasive analysis of different security implementations

Since many vendors started implementing protections against UV attacks, it has become hard to locate security fuses and defeat them. Full reverse engineering is extremely expensive and partial reverse engineering, such as following the high voltage programming line, is not practical for modern chips because they have multiple metal layers which are planarised. However, our new semi-invasive attacks such as laser scanning and fault injection could help in finding fuse location in a reasonably short time.

7.2.1 Using laser scanning technique for analysis

Laser scanning of a powered chip can help in finding the security fuse if it is not embedded in main memory. All the attacker has to do is perform two scans – one with active security and another with it disabled. Comparing two scans or subtracting one from another with minimal pre-processing should reveal the information about which transistors have changed state. Discarding the CPU registers and SRAM memory locations will lead us to the security fuse.

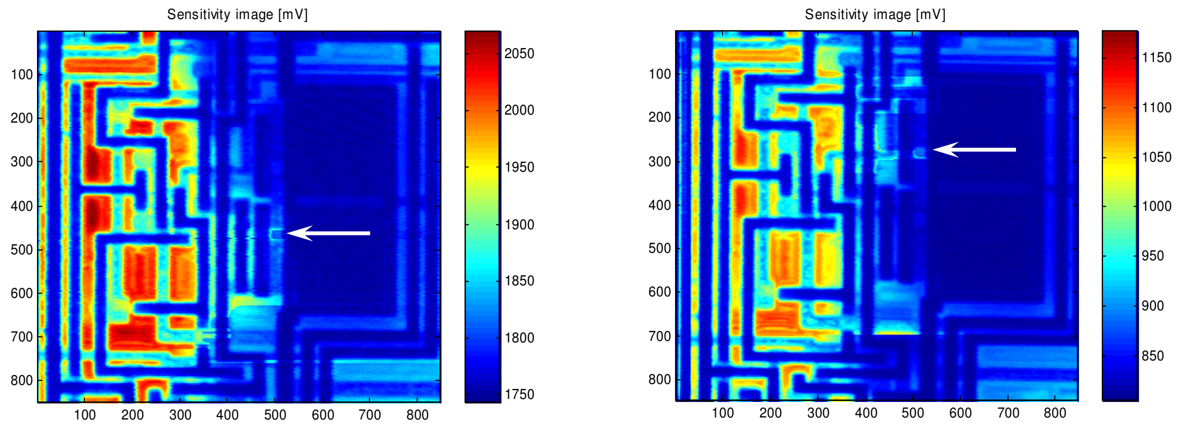


Figure 102. Laser scan of the fuse area in powered PIC16F84 chip with disabled and enabled security

We performed such a search on the Microchip PIC16F84 microcontroller and the result of two scans applied to the security fuse area is presented in Figure 102. As can be observed, the transistors connected to the security fuse and propagating its state can be easily distinguished between scans.

Although this technique gives a very quick result, it should be mentioned that it does not work for modern microcontrollers with small feature sizes and multiple top metal layers. An alternative is backside laser scanning, but this requires more precision and quite expensive equipment, because the signal has a high noise level. Either a more expensive low-noise laser source or signal averaging and post-processing will be required.

7.2.2 Using fault injection technique for analysis

Fault injection can be used to find security fuses. Although it cannot give the exact position of the fuse, especially if a photoflash lamp is used to inject the faults, this might be not necessary because the same attack can be used to disable the security once the weak point is found. The equipment setup is very similar to the one used for laser scanning, with the difference that a more powerful laser or photoflash is attached to the microscope's camera port. The chip has to be placed in a test socket on a motorised stage.

By exposing one area after another on the chip surface to the light flashes and testing the state of the security fuse, we can find photosensitive locations, which can then be examined to find the fuse. Not only the fuse itself, but also the whole chain of the control circuit may be sensitive to the fault injections. Therefore, it might not be easy to find the fuse, but from the attacker's point of view this is not necessary as he could exploit any of the vulnerable points on the chain.

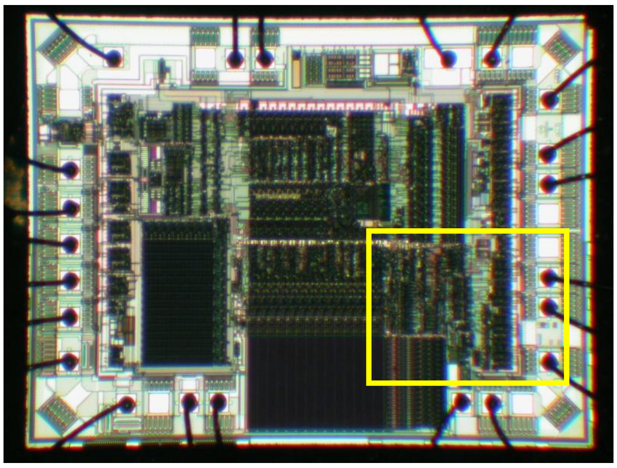


Figure 103. Picture of the PIC16C622A microcontroller die and tested area

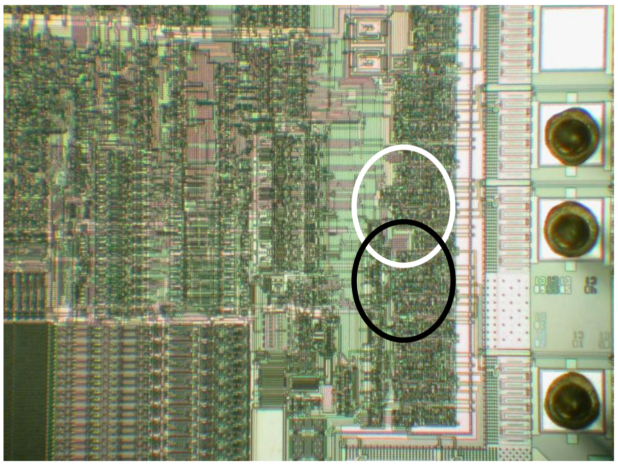


Figure 104. Locations sensitive to the photoflash light: white – CP1 reset, black – CP2 reset

The same setup can be used for security evaluation purposes. In this case each point on the chip surface is exposed to the light flash to make sure that the security is not disabled even for a very short time.

We tested the Microchip PIC16C622A microcontroller [150], which has security fuses protected from UV attacks. The die picture and tested area are shown in Figure 103. The areas which were found to be sensitive to the photoflash are shown in Figure 104. This microcontroller has two security fuses for protecting different parts of the memory and it was possible to attack each of them. In order to get access to the whole memory, both fuses must be deactivated. We can do this by flashing the area where both circles cross.

Chapter 8

Defence technologies

This chapter discusses technologies that can be used to increase the protection of semiconductor devices.

If the project can be divided and each part placed in a different microcontroller, we might arrange that the attacker will have to break all the components to reverse engineer the whole device. However, this is not a good approach for mass-production devices as cost is an important factor.

Shifting from microcontroller-based design to a microcontroller plus CPLD/FPGA could be a good alternative, especially as it is much more expensive to attack and reverse engineer CPLDs or FPGAs. Even placing important algorithms inside a 'non-secure' SRAM-based FPGA makes its reverse engineering very difficult.

Using multilayer PCBs and chips in BGA packages also increases the cost of reverse engineering because interconnections cannot be observed and it is impossible to access BGA pins directly for analysis. In order to get access to the pins, a chip must be desoldered and placed on a special test adapter. This requires special tools and a qualified engineer. Furthermore, BGA packages are hard to decapsulate as they are made from a different type of plastic than DIP, SOIC and QFP packages.

A common and widely-used technology for protecting sensitive information is data encryption. However, more attention should be paid to the secret key storage and management. If the key is programmable or can be changed at any time, it is very likely to be stored in EPROM or EEPROM. As we discussed in previous chapters, keeping plaintext information in reprogrammable memories can pose a threat to the security of the system.

Our semi-invasive attacks, and in particular fault injection attacks, pose a very high threat to hardware security of semiconductor chips, and so we have developed a technology to block such attacks. We use self-timed dual-rail circuit design techniques whereby a logical '1' or '0' is not encoded by a high or low voltage on a single line, but by 'HL' or 'LH' on a pair of lines. The combination 'HH' signals an alarm, which will typically reset the processor. Circuits can be designed so that single-transistor failures do not lead to security failure. This technology also makes power analysis attacks much harder too [151].

Of course, a balance between the cost of protection and the achieved security level is necessary. Therefore we need proper evaluation of attack scenarios and threats.

8.1 Unmarking, remarking and repackaging

Removing marking from semiconductor components is a widely used method of protection against low budget attackers, but it does not stop a determined adversary. In any case, it increases the cost of cloning and reverse engineering.

There are two ways the part number can be found. One is decapsulation followed by optical examination under a microscope. Most chips have their manufacturer name and chip ID written on the die. Another, less expensive way, is to observe the signals such as power supply, ground, clock and reset. As each family of microcontrollers has its specific pinout, it is not difficult to work out what type of microcontroller it might be.

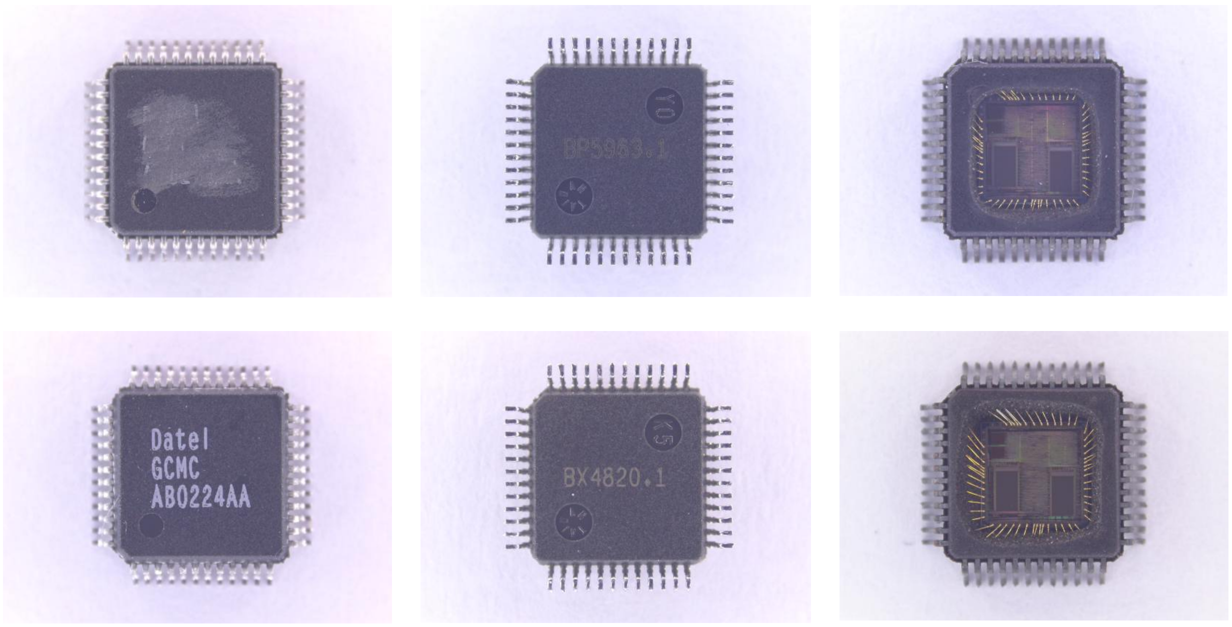


Figure 105. Ubicom SX48 microcontroller found in game consoles, with removed marking (top row) and remarked to look like an ASIC (bottom row) from front side, rear side and decapsulated

When some developers remove the chip marking, they remove it from the front side (Figure 105). Almost every chip manufacturer has a distinguishable style of marking chips from the rear side. Some, such as Atmel, print an internal factory chip number on its rear side. An attacker can easily build a correspondence table between the actual and the factory numbers by ordering samples of all types of microcontrollers from this manufacturer. Also some microcontrollers, like the Microchip Flash PICs, have a built-in chip ID number which can be read in a universal programmer to identify the chip.

Some developers went further and not only removed the original marking, but added their own marking to make the chip look like an ASIC (Figure 105). As everyone knows that reverse engineering of ASIC is extremely expensive, it is very likely that only a small group of potential attackers will try to figure out what the real chip is.

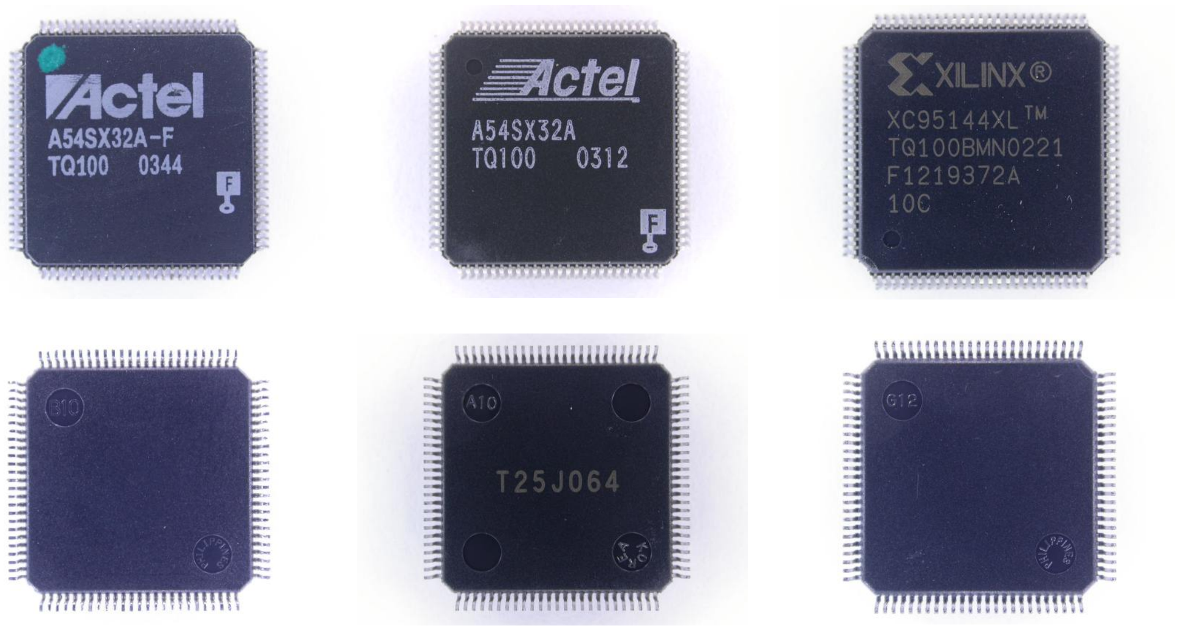


Figure 106. Remarked chip (left column) and the original Actel FPGA (middle column), and Xilinx CPLD (right column) chips from both sides

Other developers remark their chips so they look like more secure chips. For example, we found one of the chips in a game console was marked as a highly secure FPGA chip, while actually it was only a remarked CPLD which has lower security protection (Figure 106). As can be seen, its marking is slightly different from the original Actel device. From the rear side they look completely different, but very similar to the marking on the Xilinx CPLD chip. However, this trick could be illegal as it violates trademark laws. A developer might be prosecuted by the vendor for such remarking.

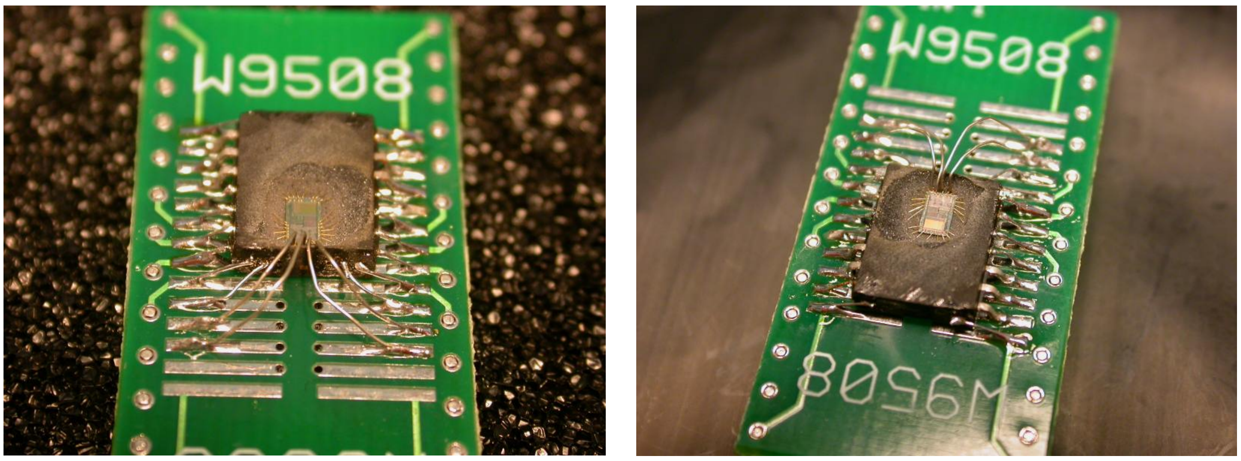


Figure 107. Restoring the cut pins on the Cypress CY7C63001A microcontroller in SOIC package

To achieve better protection, one should not only remove all markings but also turn the chip upside-down. DIP and SOIC packages could be cut from one end to make the chip look like a different one. Even if the attacker depackages the chip and finds its name, he will have to bond the broken pins, which is not easy. At the same time, as the bonding wires are still connected to the pads, it is possible to restore the connections relatively easily by placing the chip on a PCB adapter and using thin electric wires to establish connections with the bonding wires (Figure 107).

If better protection is required, the chip can be bonded and encapsulated in a non-standard package with custom marking. This is suitable for mass-production chips only, but some chip manufacturers offer such a service. Some manufacturers also offer to pre-program the chips and place custom chip ID numbers inside.

8.2 Multilevel and multipoint protection

Relatively good protection can be achieved if different security features are activated during a device's lifetime. This approach is widely used by the smartcard industry, for example, in pay-TV applications, where occasional software updates help to exclude forged cards.

Another example is toner cartridges in modern laser printers [152]. Printer manufacturers do not receive much profit from selling printers as they are complex and expensive to produce. Instead, most of the profit comes from selling toner cartridges and other accessories. But cartridges are much easier to produce and they can also be refilled. That forces printer manufacturers to develop protection mechanisms to prevent refilling and cloning of their cartridges. From the late nineties some manufacturers started placing ID chips in their products, to make refilling and cloning more difficult. The first chips had simple serial numbers which the printer's software checked. Others had a standard serial EEPROM memory with some data. Later chips had both serial ROM and EEPROM. Such protection was defeated relatively easily as these were standard chips; once the correct chip family was identified and the data structure was understood, the chip was easily reprogrammed or replaced using an ordinary microcontroller.

In the latest printers, manufacturers started using much more secure chips, such as one of the Dallas Semiconductor iButton products (a DS2432), featuring a unique serial ID number, secure EEPROM and SHA-1 engine. Then, the printer manufacturer could implement various verification protocols inside printer firmware. The advantage of having different features on a single piece of silicon allows him to activate them when necessary. For example, in the beginning printer cartridges can be shipped with only serial ID numbers used for verification. Once the manufacturer realises that his product is being cloned (for example, by a drop in sales), he can activate the next protection level by starting to use the EEPROM inside the chip. After that he can start using encryption for the EEPROM and later the SHA-1 authentication. Switching to better protection can be done automatically through on-line updates without informing the customers.

By switching to a custom ASIC chip, a further increase in protection can be achieved and some manufacturers have already started using ASICs to protect their products from cloning and refurbishing.

Having multiple layers of protection and activating them one by one lets a vendor extend a device's lifetime and increase the cost of attacking it.

Security in microcontrollers can be increased by using multiple fuses instead of a single one. Some microcontrollers benefit from this. For example, in the Microchip PIC16F84 microcontroller a single security fuse controls access to both the program and the data memories (Figure 108). In the PIC16F628 microcontroller, two fuses have to be deactivated to gain access to the whole program memory, while the data memory has separate security fuse (Figure 109). As the fuses are located far away from each other, it is extremely hard to apply a fault injection attack in a way that it will affect both security fuses without disturbing the contents of main memory.

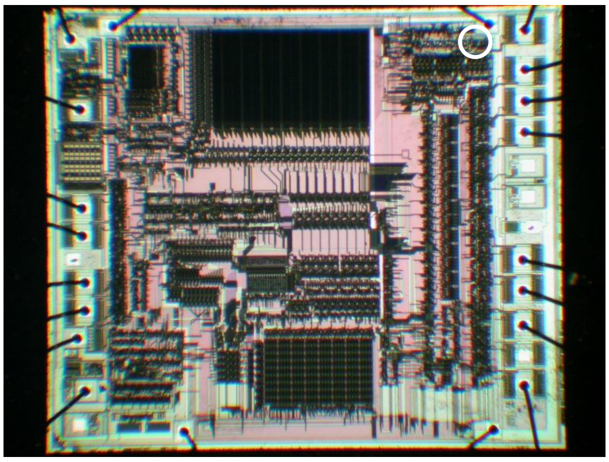


Figure 108. Location of the security fuse in the PIC16F84 microcontroller

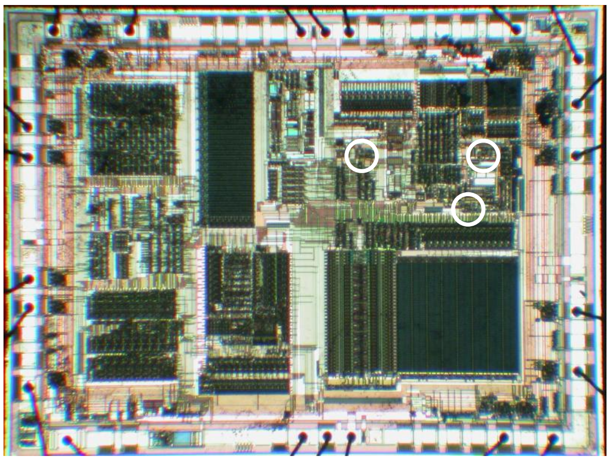


Figure 109. Location of the security fuses in the PIC16F628 microcontroller

In addition to the multiple-fuse protection, each security fuse consists of two floating gate memory cells and the fuse is disabled only if the both cells are in the right state. That increases protection against glitch attacks as well.

8.3 Burning access circuit and destroying test interface

One of the most effective methods of increasing the cost of attacking ordinary microcontrollers is burning some of the pins used for memory programming. A similar approach is used in some smartcards where the test interface dedicated for post-production testing and flash memory programming is physically removed after testing by cutting it off.

Most microcontrollers are attacked by disabling their security fuses and then accessing the memory in a normal way. Burning one of the pins prevents access even if the security fuse was successfully removed. At the same time, memory updates and reprogramming can be done through a bootloader in user code as many Flash-based microcontrollers have a self-programming feature.

The way to burn the pin is quite simple. A voltage, either positive or negative, above the allowed maximum is applied to the required pin and a current above 1 A is sent through it. This causes permanent damage to the transistors connected to the pin – p-MOS in case of high positive voltage and n-MOS in case of negative voltage.

We decided to analyse what happens to the chip's die after burning. We tested a PIC16F76 chip used in a software dongle. Our universal programmer did not recognise the chip and complained that the pin number 28 had poor contact. We decapsulated the chip in order to find out what happened to the die and whether it might be possible to restore the memory programming interface. An optical image of the relevant area is shown in Figure 110. We also chemically etched the chip to see the actual size of the damage. It can be seen that both the output transistors are completely damaged, and probably the gate oxide of the input transistors is damaged as well because it is not designed to survive high voltages.

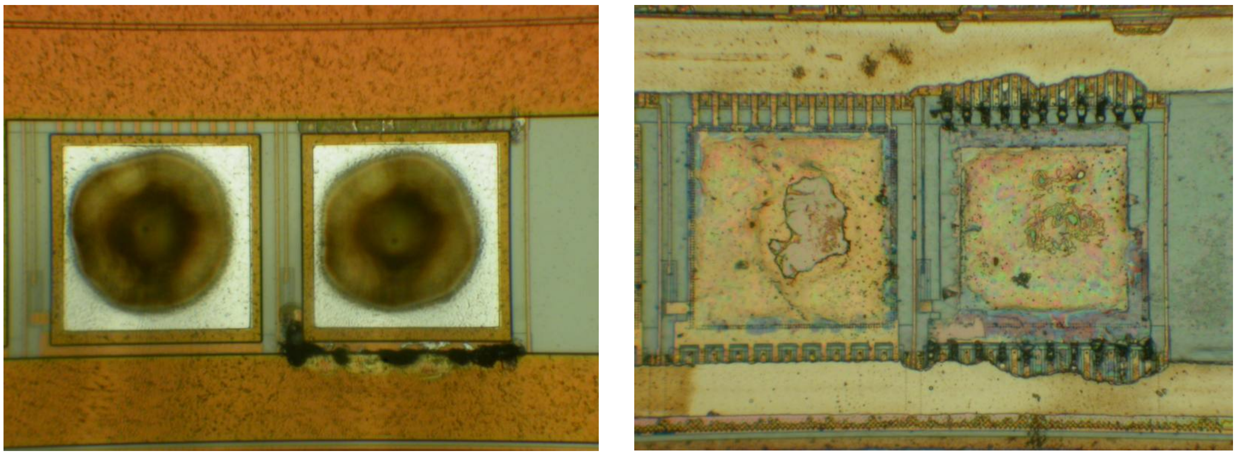


Figure 110. The pin used in programming interface is blown with high voltage. Optical images after decapsulation (left) and chemical etching (right). 200× magnification

One could try to restore the functionality of the pin, but this will very likely require FIB work with full reverse engineering of the I/O port. Another way to recover the information from the memory is by microprobing the internal data bus. These are expensive and time consuming operations demanding a highly skilled attacker.

In any case, by burning one of the pins used in memory programming, the cost of attack can be substantially increased. As this causes damage to the internal structures and passivation layer, it could probably not be used for mass-production devices. The parameters of the chip might change in time, because water and air will slowly penetrate through damage in the passivation layer and cause further degradation of the chip.

Chip manufacturers do not recommend using such technique as this can affect the operation characteristics of the device: "Even though this might seem to disable the programming interface, other functions in the device may also be damaged. Characteristics of the device, like power consumption, life time or ESD structures, may change in the process" [153].

This trick can be used for small quantities where protection is essential and very long life time is not expected from the devices. It might be a very good solution for unique designs where the design cost significantly exceeds the cost of its components. For mass products and reliable devices, more secure microcontrollers and smartcards should be used.

8.4 Smartcards and tamper protection

The obvious choice for developer who wants to achieve better protection is to choose secure microcontrollers or smartcards for his applications. Even smartcards that were designed ten years ago offer better protection against various kinds of attacks than most of the microcontrollers.

However, smartcards have some disadvantages compared to microcontrollers. They are more expensive, and very few of them can be ordered in small quantities. Development tools are expensive, and usually sold under a non-disclosure agreement with the manufacturer – as are the datasheets. Many smartcard manufacturers sell their products in large quantities and to corporate customers only.

Another disadvantage of smartcards is the limited I/O functionality. They normally have only an ISO 7816 interface and very few smartcard chips have separate I/O lines or USB port. This makes using them as a drop-in replacement to microcontrollers impossible in most applications. Smartcards could be used just to store a sensitive part of algorithms or for authentication and copy protection. However, they suit certain applications such as payment and access cards which do not require extra I/O ports. Also smartcards do not contain very powerful processors as their power consumption is an important factor as well.

Modern smartcards provide protection against various attacks. Internal voltage sensors protect against under- and over-voltages used in power glitch attacks. Clock frequency sensors prevent attackers slowing down the clock frequency for static analysis and also from raising it for clock-glitch attacks. On-chip random number generators (RNG) make attacks on encryption harder. Top-metal sensor meshes and internal bus hardware encryption make microprobing attacks very problematic. Light sensors prevent a decapsulated chip from functioning. In addition, software access to the internal memory is restricted by passwords.

Attacking smartcards, especially recently designed ones, is an extremely expensive and time consuming task. Only well equipped laboratories with highly qualified engineers generally succeed.

Better protection is offered by tamper resistant devices and modules. One example is Dallas Semiconductors JAVA iButton [52] in which the secure smartcard chip is placed in a metal can with battery and tamper sensors. If the tampering is detected the internal memory gets erased thus preserving all sensitive information from the attacker. Another example is secure module used in bank applications such as IBM 4758 [7]. It has more sophisticated tamper sensors as well as radiation and low temperature sensors to protect against various kinds of attacks. Such products are much more expensive than smartcards but certainly they provide significantly better protection. However, an attacker could still exploit bugs in software to extract the secret keys [154].

8.5 Asynchronous logic

The defensive technology that we have developed uses self-timed dual-rail logic [151]. Conventional digital logic uses a clock to synchronize activities; but the cost of clocking rises as devices become more complex, and this has led to a surge of interest in design techniques for self-timed, or asynchronous, circuits which do not use clocks. Such circuits need some mechanism whereby functional components in a circuit can signal that they are ready to receive data, or are finished. One way of doing this is to introduce redundancy into the data path.

In dual-rail logic, a '0' or '1' is signaled not by a low or high voltage on a single wire, but by a combination of signals on a pair of wires. For example, '0' may be 'LH' and '1' may be 'HL'. When used in self-timed circuits, 'LL' signals quiescence. The principal drawback of this simple arrangement is fragility: bugs tend to cause the emergence of the unwanted 'HH' state, which propagates rapidly throughout the circuit and locks it.

Our innovation was to turn this fragility to advantage, by making 'HH' into an error signal. This signal can be raised deliberately by tamper sensors, causing the device to lock [155]. Of more interest here is the fact that matters can be so arranged that single device failures are unlikely to cause the output of sensitive information [156]. We believe that such robustness will be a requirement for many high-security devices in future.

Another advantage of dual-rail encoding is reduced data dependent power consumption as all states have the same Hamming weight. Dual-rail encoding is not sufficient to guarantee a data independent power signature. The path taken by each wire could vary resulting in different wire load. This problem can be solved by careful layout control.

Self-timed designs are immune to clock glitch attacks. If the clock is required for the serial interface, it is relatively easy to make it separate from the sensitive circuit. Power glitch attacks are less successful on asynchronous circuits as they naturally adapt to the power supply voltage. However, such components as EEPROM cannot be protected by the design and might be vulnerable especially if they store keys.

The dual-rail design allows us to reliably propagate the alarm signal from the tamper sensor and block the device operation. As a result the sensitive data will be deleted and a global alarm raised. This helps to protect against fault injection attacks as well. In order to succeed, the attacker will have to inject two faults simultaneously to switch the line from 'LH' to 'HL'. With very high probability this will cause the line to go in 'HH' state for a short period of time and immediately trigger the alarm circuit.

Data dependent timing might be a problem for the self-timed circuits as their computation time varies from the computation task. This could be solved by inserting random delays into the data path [157].

The engineering details are non-trivial. For example, an obvious concern is that almost any undetected malfunction could be exploited by the attack of Boneh et al on RSA signatures. Colleagues have therefore developed a modular multiplication unit using our technology. Similarly, although bus encryption can remove the need to protect on-chip memory arrays, there remains the risk of attacks on the program counter and other registers. Other colleagues have therefore developed registers, and a memory management unit, that use our technology.

Chapter 9

Conclusion and further work

The aim of this thesis was to highlight some potential problems of hardware security in microcontrollers and smartcards, and give an introduction to various attack methods and possible protections against such attacks.

Our introduction to attack technologies included already known non-invasive attacks, such as power analysis and glitching, and invasive attacks, such as reverse engineering and microprobing. This thesis introduced a new class of attacks – semi-invasive attacks. Like invasive attacks, they require depackaging the chip to get access to the chip surface, but the passivation layer of the chip remains intact as these methods do not require electrical contact to internal metal wires.

Semi-invasive attacks are becoming more attractive as they do not require very expensive tools and give results more quickly. This is especially important, because with technological progress, invasive attacks are becoming constantly more demanding and expensive, with shrinking feature sizes and increasing device complexity. Being applied to a whole transistor or even a group of transistors, semi-invasive attacks are less sensitive to the small feature size of modern chips.

Semi-invasive attacks are not entirely new. The old EPROM-hacking trick of exposing the memory protect bit of a microcontroller to UV light requires depackaging it. Semi-invasive attacks could in theory be performed using such tools as UV light, X-rays, lasers, electromagnetic fields and local heating. They could be used individually or in conjunction with each other.

Figure 111 summarises various attacks. They are grouped in classes and way of approaching such as passive (eavesdropping, electromagnetic emission), active (interfering with device operation) and exploiting design vulnerabilities such as software bugs, weaknesses in protocols and data remanence.

The main contribution of this thesis is fault injection attacks. Using low-cost equipment, such as a photoflash or a laser pointer mounted on a microscope, the attacker can modify SRAM and EEPROM contents, or change the state of any individual CMOS transistor on a chip. This leads to almost unlimited capabilities to control chip operation and circumvent protection mechanism.

Another important contribution is our data remanence experiments. They showed that information from powered-off SRAM and erased EPROM, EEPROM and Flash memories can be extracted unless special countermeasures are implemented.

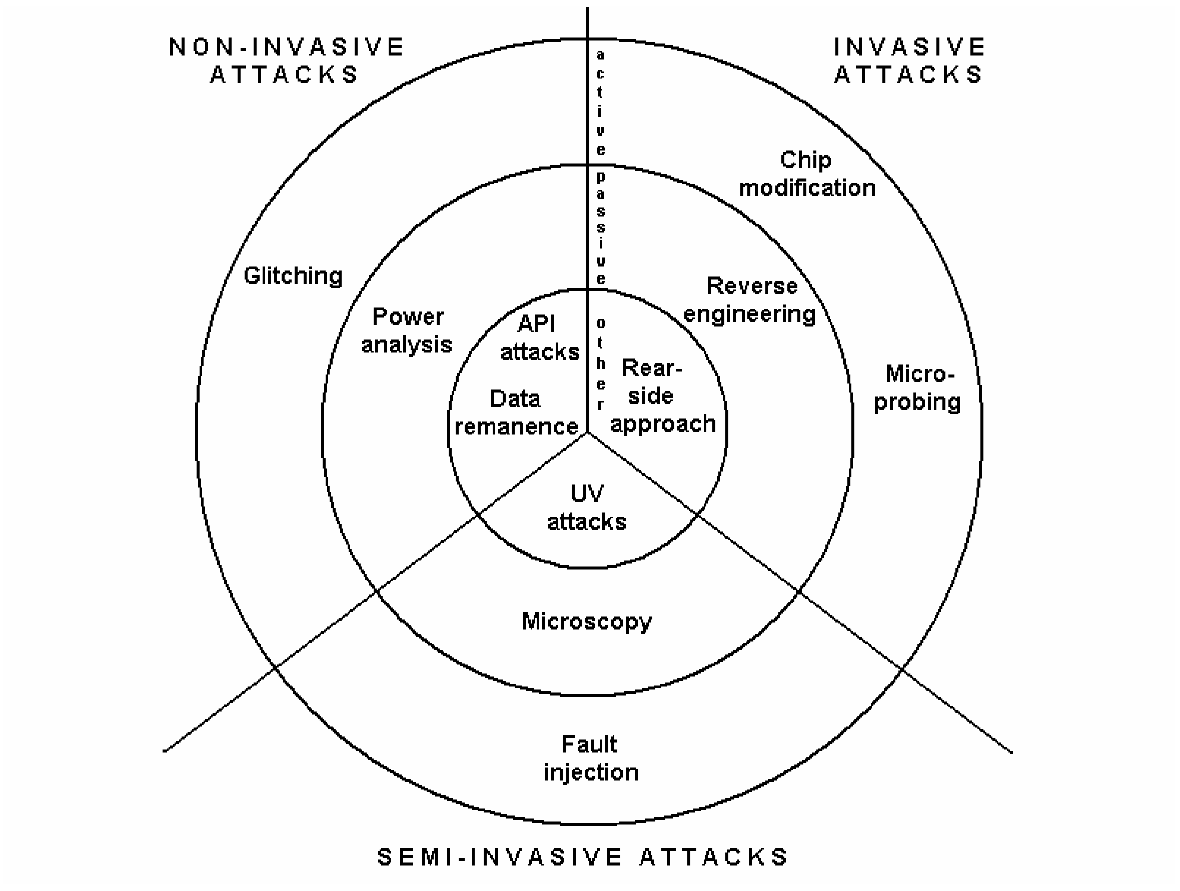


Figure 111. Relationship between different attacks

Figure 112 represents symmetry between non-invasive and semi-invasive attacks. The first is applied to a whole chip where the second to a particular point on the surface. Glitch attacks have a similar influence to fault injection with the difference that one uses electric signals and another uses optical beam, but they are both are active attacks. The same is true for the passive attacks, such as power analysis and special microscopy. Vulnerabilities caused by data remanence can be exploited in both non-invasive and invasive ways.

A general introduction into sample preparation techniques necessary for invasive and semi-invasive attacks was given. It showed how easily an attacker could get access to the chip die without using expensive tools.

Some ideas on possible protection were introduced. They involved low-cost solutions based on obscurity of the design, as well as more expensive solutions that involved a new approach to the design of silicon chips.

There is no such a thing as absolute security. Given enough time and resources, any protection can be broken. The question is how long the device can be on the open market before it is cracked. What a developer can do is increase security protection making attacks to be more expensive.

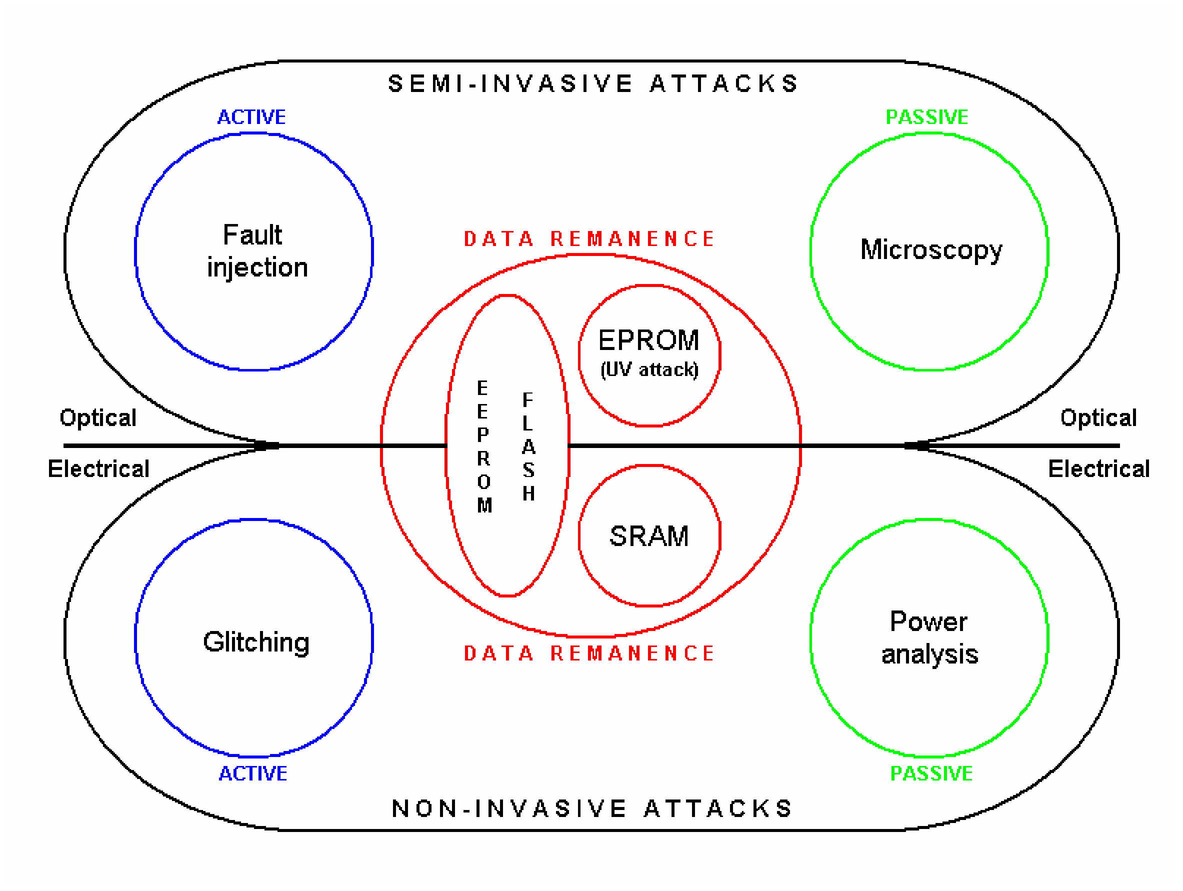


Figure 112. Symmetry between non-invasive and semi-invasive attacks

Technical progress does not stand still. Yesterday's high-end technologies become widely available to everyone including the attackers. That forces manufacturers to come up with better protection to stay at the leading edge. In this race, proper hardware security evaluation methods are required. This can be done faster with less expensive equipment by using semi-invasive techniques, as presented in this thesis.

Very often security is connected with human factors and system design. Many smartcards were broken without sophisticated invasive attacks, but mainly through analysing the protocols used, and with power analysis. A wide variety of examples on how secure systems should not be designed, and ideas on how to do design work properly, are given in Ross Anderson's book "Security Engineering" [11].

The fact that most examples are given on the Microchip PIC microcontrollers does not mean they are less secure than the others. Only proper testing and evaluation of a particular device can estimate its security level.

New devices usually offer better protection, even in the same family, because they have smaller feature sizes which are harder to attack. Also, non-invasive attacks are very sensitive to design.

If the chip is redesigned or shrunk, with very high probability old non-invasive attacks will not work.

Further scientific work in our plan includes a fuller investigation of the potential for attacks by an opponent with a moderately resourced laboratory, by which we mean a modern probing station with a multiple wavelength laser. They might become essential tools for laboratories offering hardware security analysis.

Finally we suggest four major directions for further research in hardware security:

Data remanence in non-volatile memories. Security protection in modern microcontrollers, CPLDs and FPGAs with EEPROM/Flash memories is based on the assumption that information from the memory disappears completely after erasing. Chip manufacturers were very successful in making their hardware security fuses very robust to all sorts of attacks. The common problem though was data remanence in floating gate transistors. The information stored inside a EEPROM/Flash cell in the form of a charge on the floating gate changes some parameters of the storage transistor, so that even after erase operation the transistor does not get back to its initial state, thereby allowing for attacker to somehow distinguish between previously programmed and not programmed transistors and restore the information from the erased memory. In practice the attack can be done in different ways. The cheapest way is to measure the parameters of the transistor non-invasively by observing voltage and time dependent characteristics of each memory cell inside the array. Fortunately for security, this can be applied to a very limited number of chips. Another approach is to use semi-invasive and invasive methods to measure or directly observe the changes inside the memory transistors. This is currently under investigation and we are currently looking for funding to continue this research.

Advanced power analysis techniques. Power analysis has been used for years to monitor the processes taking place inside microcontrollers and smartcards. It is possible to figure out what instruction is currently being executed and number of bits set/reset in arithmetic operations, as well as the states for carry, zero and negative flags. However, as chips become more and more complex with instruction/data caches and pipelining mechanisms used inside their CPUs, it becomes more and more difficult to observe their operation through power consumption. One approach is to use semi-invasive and invasive attacks so that the power consumption of a relatively small area will be monitored thus eliminating the influence of the rest of the chip circuit.

Practical use of fault injection attacks. We introduced these attacks over two years ago. Unfortunately they have still not been properly investigated. Research is needed to estimate the requirements on these attacks for each chip manufacturing technology and possible success rate. We are currently setting up the equipment necessary for this research.

Using nanotechnologies for hardware security analysis. Current trends in the miniaturisation of electronic devices demand the ability to understand the structure and properties on the deep submicron level (the latest technology is 90 nm and 65 nm is already proposed). Recent achievements in scanning probe microscopy allow us to observe many characteristics of semiconductor chip surface such as landscape (with atomic force microscopy), doping concentration (with scanning capacitance microscopy), resistance (with scanning spreading resistance microscopy), magnetic field (with magnetic force microscopy), temperature (with scanning thermal microscopy), and many others. We need research to estimate how much information could be extracted from silicon chips using such technologies. This research might involve designing and building some special microscopes. As such research requires large investments in equipment, it is difficult to predict when it will be started.

Appendix

During our research we tested and evaluated security protection in many microcontrollers and smartcards. The summarised results for some of them are presented in the table below. The information on technology and design is only approximate because many manufacturers do not provide such information and we had to measure the transistors under a microscope. Security implementation was assessed in the way discussed in Chapter 2 (see Figure 35). UV protection concerns whether it is possible to disable the security with a UV light. This table is related to the results of the hardware security analysis, and we have not necessarily succeeded with all of the possible attacks.

Chip name	**Memory type**	**CPU core**	**Techno-logy and design**	**Micro-probing protect.**	**Fuse location**	**Security scheme**	**UV pro-tect.**	**Possible attacks**
MC68HC05B6	ROM E^2PROM	CISC Neu-mann	Not tested	No	Embed-ded in E^2PROM	A Software	Not tested	Glitching, fault injection
MC68HC705C9A	ROM EPROM	CISC Neu-mann	1.0 µm, 1M	No	Embed-ded in EPROM	D Hardware Software	No	UV, fault injection
PIC12C509A	EPROM	RISC Har-vard	0.9 µm, 2M	No	Far from memory	D Hardware	No	UV, fault injection, wire cut
PIC12C672	EPROM	RISC Har-vard	0.9 µm, 2M	No	Far from memory	D Hardware	Yes	Fault inj., UV after laser cut
PIC16F84	Flash	RISC Har-vard	1.2 µm, 2M	No	Far from memory	D Hardware	Yes	Fault inj., glitching, wire cut
PIC16F628	Flash	RISC Har-vard	0.9 µm, 2M	No	Far from memory	D Hardware	Yes	Glitching, wire cut
SX28	Flash	RISC Har-vard	0.6 µm, 2M glue log.	Yes	Far from memory	B Hardware	Yes	Glitching, fault injection
ST62T60	EPROM	CISC Har-vard	0.7 µm, 2M	No	Shares same area	D Hardware	No	UV, fault injection

Chip name	**Memory type**	**CPU core**	**Techno-logy and design**	**Micro-probing protect.**	**Fuse location**	**Security scheme**	**UV pro-tect.**	**Possible attacks**
HD6473048	EPROM	CISC Neu-mann	0.8 µm, 2M	No	Shares wordlines	D Hardware	No	UV, fault injection
CY7C63001A	EPROM	RISC Har-vard	0.6 µm, 2M, glue log.	No	Close to memory	D, Hardware	No	UV, fault injection
AT89C51	Flash	CISC Har-vard	0.7 µm, 2M	No	Far from memory	D Hardware	Yes	Fault inj., micro-probing
AT90S2313	Flash	RISC Har-vard	0.7 µm, 2M glue log.	Yes	Far from memory	D Hardware	Yes	Glitching, fault injection
µPD78F9116	Flash	CISC Neu-mann	0.35 µm, 3M glue log.	No	No readback	Software	Not tested	Glitching
MSP430F112	ROM Flash	RISC Neu-mann	0.35 µm, 3M glue log.	No	Embed-ded in Flash	B Software	Yes	Fault injection
HCS200	E^2PROM	N/A	1.2 µm, 2M	No	No readback	Hardware	No	UV, micro-probing
MC68HC05SC27	ROM E^2PROM	CISC Neu-mann	1.0 µm, 2M	No	Embed-ded in E^2PROM	Software	Not tested	Micro-probing, glitching
SLE44	ROM E^2PROM	CISC Har-vard	1.0 µm, 2M	No	Embed-ded in E^2PROM	Software	Not tested	Micro-probing
ST16	ROM E^2PROM	CISC Neu-mann	1.0 µm, 2M	Yes	Embed-ded in E^2PROM	Software	Not tested	Micro-probing
AT90SC	Flash	RISC Har-vard	0.35 µm, 3M glue log.	Yes	Embed-ded in Flash	Software	Not tested	Glitching

Glossary

AC	Alternating Current
ADC, A/D	Analog-to-Digital Converter
AFM	Atomic Force Microscopy
ALU	Arithmetic Logic Unit. Part of processor
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
ATR	Answer To Reset. The code string a smartcard normally sends to the host after a reset.
AVR	Family of microcontrollers from Atmel.
BGA	Ball Grid Array. Type of chip package
CHE	Channel Hot Electrons.
CISC	Complex Instruction Set Computer
CMOS	Complementary Metal-Oxide-Semiconductor transistor
CMP	Chemical-Mechanical Planarisation
CPLD	Complex Programmable Logic Device
CPU	Central Processor Unit
DAC, D/A	Digital-to-Analog Converter
DC	Direct Current
DIP, DIL	Dual-In-Line Plastic. Type of a chip package
DPA	Differential Power Analysis
DRAM	Dynamic RAM

EEPROM, E^{2}PROM	Electrically Erasable Programmable ROM
EPROM	Electrically Programmable ROM
FET	Field-Effect Transistor
FIB	Focused Ion Beam. Machine used in failure analysis
FIPS	Federal Information Processing Standards. FIPS 140 is a validation standard for security devices
FPGA	Field Programmable Logic Device
FRAM, FeRAM	Ferroelectric RAM
IC	Integrated Circuit
I^{2}C	Inter-IC bus. Serial synchronous master-slave interface
IP	Intellectual Property
IR	InfraRed. Light with wavelength longer than visible light
ISO	International Organisation for Standardisation. ISO 7816 is a smartcard standard
I/O	Input/Output
JTAG	Joint Test Action Group. A test interface used in FPGAs and complex semiconductor devices
LCD	Liquid Crystal Display. Used for thermal analysis of integrated circuits
LED	Light Emitting Diode
LIVA	Light-Induced Voltage Alterations. Failure analysis technique
LWD	Long Working Distance. Microscope objective that has focal length greater than in a standard objective
MCU	MicroController Unit
MOS	Metal-Oxide-Semiconductor transistor
MRAM	Magnetoresistive RAM
NA	Numerical Aperture. The parameter of optical system associated with the angular aperture and which determines the resolution
NAND	Not AND. Boolean-logic function; also, type of memory structure with memory transistors connected into the chain along the bit-line

NIR	Near-Infrared. Region of infrared light close to the visible light
n-MOS, NMOS	N-channel MOS transistor
NOR	Not OR. Boolean-logic function. Also, type of memory structure with memory transistors placed between the ground and the bit-line
NUV	Near-Ultraviolet. Region of ultraviolet light close to the visible light
NVM	Non-Volatile Memory
NVRAM	Non-Volatile RAM
OBIC	Optical Beam Induced Current. Failure analysis technique
OR	Boolean-logic function. Also, type of memory structure with memory transistors placed between the supply and the bit-line
OTP	One-Time Programmable. Devices with memory that can be written but not erased
PC	Personal Computer
PCB	Printed Circuit Board
PIC	Family of microcontrollers from Microchip
p-MOS, PMOS	P-channel MOS transistor
PROM	Programmable ROM
QFP	Quad Flat Pack. Type of chip package
RAM	Random Access Memory
RISC	Reduced/Regular Instruction Set Computer
RNG	Random Number Generator
ROM	Read-Only Memory
RSA	Cryptographic algorithm invented by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1977
RS-232	Serial asynchronous interface used in most PCs and some microcontrollers for communication
SCM	Scanning Capacitance Microscopy
SEM	Scanning Electron Microscope

SHA-1	Secure Hash Algorithm
SOIC	Small Outline Integrated Circuits. Type of chip package
SPA	Simple Power Analysis
SPI	Serial Peripheral Interface
SPM	Scanning Probe Microscopy
SRAM	Static RAM
SX	Family of microcontrollers from Ubicom (former Scenix)
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
UV	UltraViolet. Light with wavelength shorter than visible light
VTROM	Voltage-Threshold ROM
XNOR	Exclusive NOR. Boolean-logic function
XOR	Exclusive OR. Boolean-logic function

Bibliography

[1] Sergei Skorobogatov, Ross Anderson, Optical Fault Induction Attacks, Cryptographic Hardware and Embedded Systems Workshop (CHES-2002), LNCS, Vol. 2523, Springer-Verlag, 2002, pp. 2–12

[2] Ross J. Anderson, Markus G. Kuhn, Tamper Resistance – a Cautionary Note, The Second USENIX Workshop on Electronic Commerce, Oakland, California, November 18–21, 1996

[3] Neil H.E. Weste, Kamran Eshraghian, Principles of CMOS VLSI Design: A Systems Perspective, Addison-Wesley, 1992

[4] Lance A. Glasser, W. Dobberpuhl, The Design and Analysis of VLSI Circuits, Addison Wesley, 1995.

[5] Computer Aids for VLSI Design. *http://www.rulabinsky.com/cavd/*

[6] Design of VLSI Systems. *http://vlsi.wpi.edu/webcourse/toc.html*

[7] IBM PCI Cryptographic Coprocessor.
http://www-3.ibm.com/security/cryptocards/html/overproduct.shtml

[8] Oliver Kömmerling, Markus G. Kuhn, Design Principles for Tamper-Resistant Smartcard Processors, USENIX Workshop on Smartcard Technology, Chicago, Illinois, USA, May 10–11, 1999

[9] Dan Boneh, Richard A. DeMillo, Richard J. Lipton, On the Importance of Checking Cryptographic Protocols for Faults, Advances in Cryptology – Eurocrypt 97, LNCS, Vol. 1233, Springer-Verlag, 1997, pp. 37–51

[10] Ross J. Anderson, Markus G. Kuhn, Low Cost Attacks on Tamper Resistant Devices, Proceedings of the 5th International Workshop on Security Protocols, 1997, pp. 125–136

[11] Ross J. Anderson, Security Engineering – A Guide to Building Dependable Distributed Systems, John Wiley and Sons, 2001

[12] Intel, Microprocessor Hall of Fame.
http://www.intel.com/intel/intelis/museum/online/hist_micro/hof/index.htm

[13] Intel, Thirty-five Years of Innovation.
ftp://download.intel.com/intel/anniversary/35th.pdf

[14] The Free Dictionary Encyclopedia: Harvard Architecture.
http://encyclopedia.thefreedictionary.com/harvard%20architecture

[15] The Free Dictionary Encyclopedia: CISC.
http://encyclopedia.thefreedictionary.com/CISC

[16] The Free Dictionary Encyclopedia: Instruction Pipeline.
http://encyclopedia.thefreedictionary.com/instruction%20pipeline

[17] The Free Dictionary Encyclopedia: Cache.
http://encyclopedia.thefreedictionary.com/cache

[18] Embedded controllers by Micromint. *http://www.micromint.com/products/sbc.htm*

[19] Kingpin, Attacks on and Countermeasures for USB Hardware Token Devices, Proceedings of the Fifth Nordic Workshop on Secure IT Systems Encouraging Co-operation, Reykjavic, Iceland, October 12–13, 2000, pp. 35–57

[20] GameBoy Advance. *http://www.ziegler.desaign.de/GBA/gba.htm*

[21] Secure Software Protection, Licensing & Distribution.
http://www.ealaddin.com/hasp/

[22] Microchip PIC12C5XX Data Sheet, 8-Pin, 8-Bit CMOS Microcontrollers.
http://ww1.microchip.com/downloads/en/DeviceDoc/40139e.pdf

[23] Motorola MC68HC705C9A, Advance Information.
http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC705C9A.pdf

[24] Microchip PIC16F627A/628A/648A Data Sheet, Flash-Based 8-Bit CMOS Microcontrollers with nanoWatt Technology.
http://ww1.microchip.com/downloads/en/DeviceDoc/40044b.pdf

[25] Texas Instruments, Features of the MSP430 Bootstrap loader, Application Report, January 2003. *http://focus.ti.com/lit/an/slaa089b/slaa089b.pdf*

[26] Secure ICs, Smartcard ST16 Platform ICs.
http://www.st.com/stonline/products/families/smartcard/flsc9907.htm

[27] Infineon Technologies. Smart Card Controllers, The 66/11 Family.
http://www.infineon.com/cgi/ecrm.dll/ecrm/scripts/prod_cat.jsp?oid=-9351

[28] Dallas Semiconductor/Maxim DS5002FP Secure Microprocessor Chip.
http://pdfserv.maxim-ic.com/en/ds/DS5002FP.pdf

[29] Markus G. Kuhn, Cipher Instruction Search Attack on the Bus-Encryption Security Microcontroller DS5002FP, IEEE Transactions on Computers, Vol. 47(10), October 1998, pp. 1153–1157.

[30] Philips Semiconductors Leads Industry with Smart Card Security Benchmark, Product News from Philips Semiconductors, October 16, 1998.
http://www.semiconductors.philips.com/news/content/file_354.html

[31] The Replacement IC Process, InnovASIC – The End of Obsolescence.
http://www.innovasic.com/icprocess.html

[32] IA64F3048SEC, Security-Enhanced Microcontroller, Advance Information Sheet.
http://www.innovasic.com/pdfs/IA64F3048SEC_Advance_Info_Sheet.pdf

[33] Ubicom SX20AC/SX28AC, Configurable Communications Controllers with EE/Flash Program Memory, In-System Programming Capability and On-Chip Debug, Datasheet.
http://www.ubicom.com/pdfs/SX-DDS-SX2028AC-17.pdf

[34] Cypress CY7C63000A, CY7C63001A, CY7C63100A, CY7C63101A, Universal Serial Bus Microcontroller.
http://www.cypress.com/cfuploads/img/products/CY7C63101A.pdf

[35] CMP Technology. *http://www.cmptechnology.com/*

[36] Microchip PIC16F87X Data Sheet, 28/40-Pin, 8-Bit CMOS Microcontrollers.
http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf

[37] Microchip 16F87XA Data Sheet, 28/40-Pin, Enhanced Flash Microcontrollers.
http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf

[38] Cypress CY7C68013A, CY7C68015A, EZ-USB FX2LP USB Microcontroller, High-Speed USB Peripheral Controller.
http://www.cypress.com/cfuploads/img/products/cy7c68013a.pdf

[39] Catalyst CAT22C10, 256-Bit Nonvolatile CMOS Static RAM.
http://www.catalyst-semiconductor.com/documents/22C10.pdf

[40] Fujitsu FRAM Ferroelectric RAM Technology.
http://www.fme.fujitsu.com/products/fram/technology.html

[41] Fujitsu FRAM Products.
http://www.fujitsu.com/services/microelectronics/technical/framcard/casestudy_technical-framcard_3.html

[42] James Daughton, Magnetoresistive Random Access Memory (MRAM).
http://www.nve.com/otherbiz/mram.pdf

[43] Infineon Technologies News: Infineon and IBM Present World's First 16 Mbit MRAM – Innovative Chip Design Results in Highest Density Reported to Date.
http://www.infineon.com/cgi/ecrm.dll/jsp/showfrontend.do?lang=EN&news_nav_oid=-9860&content_type=NEWS&content_oid=106372

[44] FuseLock: Security in Actel Antifuse FPGAs.
http://www.actel.com/products/rescenter/security/solutions/antifuse.aspx

[45] Security in QuickLogic Devices.
http://www.quicklogic.com/images/QL_security_WP.pdf

[46] Programmable ASICs. The Antifuse.
http://www.owlnet.rice.edu/~elec522/w3/CH04.pdf

[47] William D. Brown, Joe E. Brewer, Nonvolatile Semiconductor Memory Technology: A Comprehensive Guide to Understanding and Using NVSM Devices, IEEE Press, 1997.

[48] Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic, Digital Integrated Circuits, Second Edition, Prentice-Hall, 2002

[49] Dallas Semiconductor DS1961S, 1kb Protected EEPROM iButton With SHA-1 Engine. *http://pdfserv.maxim-ic.com/en/ds/DS1961S-DS1961S-F5.pdf*

[50] Peter Gutmann, Data Remanence in Semiconductor Devices, 10th USENIX Security Symposium, Washington, D.C., August 13–17, 2001

[51] Kenneth Yun, Memory, UC San Diego, Adapted from EE271 notes, Stanford University

[52] Java-Powered Cryptographic iButton.
http://www.ibutton.com/ibuttons/java.html

[53] Peter Gutmann, Secure Deletion of Data from Magnetic and Solid-State Memory, 6th USENIX Security Symposium Proceedings, San Jose, California, July 22–25, 1996, pp. 77–89

[54] Sergei Skorobogatov, Low Temperature Data Remanence in Static RAM, Technical Report UCAM-CL-TR-536, University of Cambridge, Computer Laboratory, June 2002.

[55] Fujitsu Microelectronics and Electronic Devices, FRAM Cell Structure.
http://www.fujitsu.com/services/microelectronics/technical/fram/casestudy_technical-fram_3.html

[56] Jacob Baker, Sensing Circuits for Resistive Memory.
http://cmosedu.com/jbaker/papers/IEEE_Boise_EDS_2002_2.pdf

[57] Kate Ravilious, Keep an eye on electricity as it flows through circuits, New Scientist, 24 May 2003, p. 21.

[58] NEC Electronics Microcomputer, 8-Bit 78K0S Series.
http://www.necel.com/micro/english/product/sc/78k0s/78k0s.html

[59] Zilog Z86E33/733/E34, Z86E43/743/E44. CMOS Z8 OTP Microcontrollers.
http://www.zilog.com/docs/z8/z86e33.pdf

[60] STMicroelectronics ST62T53C/T60C/T63C, ST62E60C, 8-Bit OTP/EPROM MCUs with A/D Converter. *http://www.st.com/stonline/books/pdf/docs/6215.pdf*

[61] Hitachi Single-Chip Microcomputer, H8/3048 Series.
http://documentation.renesas.com/eng/products/mpumcu/e602073_h83048.pdf

[62] Holtek HT48R50A-1/HT48C50-1, I/O Type 8-Bit MCU.
http://www.holtek.com/pdf/uc/48x50_1v160.pdf

[63] Motorola M68HC05 Microcontrollers, ROM Security Feature, p. 30.
http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC05C8A.pdf

[64] Microchip PIC12F629/675 Data Sheet, Code protection feature notes, p. 2.
http://ww1.microchip.com/downloads/en/DeviceDoc/41190c.pdf

[65] EPROM Security on the H8/300, H8/300L, H8/500, H8/300H and SH7000 families, Hitachi Europe Ltd, Issue APPS/011/3.1, 1996

[66] Friedrich Beck, Integrated Circuits Failure Analysis: A Guide to Preparation Techniques, John Wiley & Sons, 1997

[67] Lawrence C. Wagner, Failure Analysis of Integrated Circuits: Tools and Techniques, Kluwer Academic Publishers, 1999

[68] FEI Company, Tools for Nanotech, VectraVision.
http://www.feicompany.com/systems/product.aspx?id=33

[69] G. Binnig, C.F. Quate, C. Gerber, Atomic Force Microscope, Physical Review Letters, Vol. 56(9), 1986, pp. 930–933

[70] Scanning Probe Microscopy – Summary. MSE603//MSE571, April 2004, pp. 76–83.
http://www.mse.cornell.edu/courses/mse603/Course_Notes/notes04_24.pdf

[71] C.C. Williams, Two-Dimentional Dopant Profiling by Scanning Capacitance Microscopy, Annual Review of Material Science, 1999, Vol. 29, pp. 471–504

[72] Used-Line. *http://www.used-line.com*

[73] Ross J. Anderson, Crypto in Europe - Markets, Law and Policy, Cryptography: Policy and Algorithms, LNCS, Vol. 1029, Springer-Verlag, 1995, pp. 75–89

[74] D.G. Abraham, G.M. Dolan, G.P. Double, J.V. Stevens, Transaction Security System, IBM Systems Journal, Vol. 30(2), 1991, pp. 206–229

[75] TIPS for FIPS 140, Selling Applications with Cryptography to Federal Agencies, RSA Security White Paper.
http://wp.bitpipe.com/resource/org_1039183786_34/FIPS_WP_0603_bitpipe.pdf

[76] Common Criteria for IT Security Evaluation. *http://csrc.nist.gov/cc/*

[77] Ross J. Anderson, Markus G. Kuhn, Low Cost Attacks on Tamper Resistant Devices, in M.Lomas et al. (ed.), Security Protocols, 5th International Workshop, Paris, France, April 7–9, 1997

[78] Paul C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems, Advances in Cryptology, CRYPTO'96, LNCS, Vol. 1109, Springer-Verlag, 1996, pp. 104–113

[79] J.-J. Dhem, F. Koeune, P.-A. Leroux, P. Mestre, J.-J. Quisquater, J.-L. Willems, A Practical Implementation of the Timing Attack, Proceedings of CARDIS'98, Smart Card Research and Advanced Applications, 1998

[80] D. Chaum, Blind Signatures for Untraceable Payments, Advances in Cryptology: Proceedings of Crypto 82, Plenum Press, 1983, pp. 199–203

[81] iButton Products. *http://www.ibutton.com/products/ibuttons.html*

[82] Engineering Technical Laboratory, Security protection in Motorola Microcontrollers. *http://www.etlweb.com/articles_secprotect.html*

[83] S.M. Sze, Semiconductor Devices: Physics and Technology, John Wiley and Sons, 2002

[84] Paul Kocher, Joshua Jaffe, Benjamin Jun, Differential Power Analysis, CRYPTO'99, LNCS, Vol. 1666, Springer-Verlag, 1999, pp. 388–397

[85] John Kelsey, Bruce Schneier, David Wagner, Chris Hall, Side Channel Cryptanalysis of Product Ciphers, Journal of Computer Security, Vol. 8(2–3), 2000, pp. 141–158

[86] Thomas S. Messerges, Ezzy A. Dabbish, Robet H. Sloan, Investigations of Power Analysis Attacks on Smartcards, USENIX Workshop on Smartcard Technology, Chicago, Illinois, USA, May 10–11, 1999

[87] Motorola M68HC08 Microcontrollers, MC68HC908JB8, MC68HC08JB8, MC68HC08JT8, Technical Data.
http://e-www.motorola.com/files/microcontrollers/doc/data_sheet/MC68HC908JB8.pdf

[88] Jean-Jacques Quisquater, David Samyde, Eddy current for Magnetic Analysis with Active Sensor, UCL, Proceedings of Esmart 2002 3rd edition, Nice, France, September 2002.

[89] David Samyde, Sergei Skorobogatov, Ross Anderson, Jean-Jacques Quisquater, On a New Way to Read Data from Memory – SISW2002 First International IEEE Security in Storage Workshop

[90] Motorola MC68HC05B6 Technical Data.
http://e-www.motorola.com/files/microcontrollers/doc/data_sheet/MC68HC05B6.pdf

[91] Microchip PIC16F8X 18-pin Flash/EEPROM 8-Bit Microcontrollers.
http://ww1.microchip.com/downloads/en/DeviceDoc/30430c.pdf

[92] Microchip PIC16F84A Data Sheet, 18-pin Enhanced Flash/EEPROM 8-bit Microcontroller. *http://ww1.microchip.com/downloads/en/DeviceDoc/35007b.pdf*

[93] Sean W. Smith, Steve Weingart, Building a High-Performance, Programmable Secure Coprocessor, Computer Networks, Vol. 31, April 1999, pp. 831–860

[94] Sean W. Smith, Elaine R. Palmer, Steve Weingart, Using a High-Performance, Programmable Secure Coprocessor, Second International Conference on Financial Cryptography, LNCS, Vol. 1465, Springer-Verlag, February 1998

[95] Steve H. Weingart, Physical Security for the μABYSS System, Proceedings of the IEEE Computer Society Conference on Security and Privacy, 1987, pp. 52–58

[96] Steve H. Weingart, Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defenses, Workshop on Cryptographic Hardware and Embedded Systems (CHES 2000), LNCS, Vol. 1965, Springer-Verlag, 2000, pp. 302–317

[97] Intel StrataFlash Memory (J3), 28F256J3, 28F128J3, 28F640J3, 28F320J3.
ftp://download.intel.com/design/flcomp/datashts/29066719.pdf

[98] P.L. Rolandi, R. Canegallo, E. Chioffi, D. Gerna, G. Guaitini, C. Issartel, A. Kramer, F. Lhermet, M. Pasotti, 1M-Cell 6b/Cell Analog Flash Memory for Digital Storage, SGS-Thomson Microelectronics, IEEE International Solid-State Circuits Conference (ISSCC), Agrate Brianza, Italy, 1998

[99] Intel 28F010 and 28F020, 5 Volt Bulk Erase Flash Memory.
http://www.sunmark.com/datasheets/28f010.pdf

[100] Paolo Pavan, Luca Larcher, Massimiliano Cuozzo, Paola Zuliani, Antonino Conte, A Complete Model of E^2PROM Memory Cells for Circuit Simulations, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 22(8), August 2003

[101] H. Kume, H. Yamamoto, T. Adachi, T. Hagiwara, K. Komori, T. Nishimoto, A. Koike, S. Meguro, T. Hayashida, T. Tsukada, A Flash-erase EEPROM cell with an asymmetric source and drain structure, IEEE IEDM Technical Digest, 1987, pp. 560–563

[102] Texas Instruments Microcontrollers, MSP430 MCUs.
http://focus.ti.com/mcu/docs/overview.tsp?familyId=342&templateId=5246&navigationId=11466&path=templatedata/cm/mcuovw/data/msp430_ovw

[103] Kulicke and Soffa Model 4123 Universal Wedge Binder.
http://www.kns.com/prodserv/PDFS/MWB/4123.pdf

[104] Sputter Coating Incorporating Emitech K500, K550, K575 and K675X, Technical Brief.
http://www.emitech.co.uk/PDF/SputterCoating.pdf

[105] NSC IC Decapping System using Fuming Nitric Acid.
http://www.nscnet.co.jp/e/pdt/pa103.html

[106] 12c508a Deprotection Tutorial. *http://www.rampantapathy.co.uk/12c508a.html*

[107] Sony Semiconductor, Quality and Reliability HandBook, October 2000, p. 105
http://www.sony.net/Products/SC-HP/tec/catalog/pdf/qr_all.pdf

[108] Integrated Reliability, Nitrox. *http://www.irsi.com/Site1/NitroxIntro.html*

[109] Microchip PIC16F7X Data Sheet, 28/40-pin 8-bit CMOS Flash Microcontrollers.
http://ww1.microchip.com/downloads/en/DeviceDoc/30325b.pdf

[110] Chipworks. *http://www.chipworks.com*

[111] Bottom Line Technologies, Xilinx Reverse Engineering Services.
http://www.bltinc.com/Services.Xilinx-Reverse-Engineering.htm

[112] Nikon Microscopy, Basic Microscopy Concepts, Resolution.
http://www.microscopyu.com/articles/formulas/formulasresolution.html

[113] Olympus Microscopy Resource Center, Numerical Aperture and Image Resolution.
http://www.olympusmicro.com/primer/java/imageformation/airyna/

[114] Nikon Microscopy, Differential Interference Contrast Microscopy.
http://www.microscopyu.com/articles/dic/dicindex.html

[115] Nikon Microscopy, Principles of Phase Contrast Microscopy.
http://www.microscopyu.com/articles/phasecontrast/phasehome.html

[116] Nikon Microscopy, Introduction to Confocal Microscopy.
http://www.microscopyu.com/articles/confocal/index.html

[117] Nikon Microscopy, Integrated Circuit Inspection.
http://www.microscopyu.com/tutorials/java/confocal/index.html

[118] Olympus Microscopy, MX50 Inspection Microscope.
http://cf.olympus-europa.com/micro/intro.cfm?id=MX50

[119] Carl Zeiss, Wafer Inspection.
http://www.zeiss.de/C1256C1500431210/Inhalt-Frame/4D4FE59087410BB741256A4D003EB1EF

[120] Leica Microsystems, Leica INM100.
http://www.semiconation.com/WebSite/SC_SEMI.nsf?opendatabase&path=/WEBSITE/Products.nsf/(ALLIDs)/E9037898D3B8D85EC1256A1F004C17EC

[121] Mitutoyo FS70 Ergonomic High Power Microscope.
http://www.mitutoyo.com/pdf/1503-378.pdf

[122] S. Blythe, B. Fraboni, S. Lall, H. Ahmed, U. de Riu, Layout reconstruction of complex silicon chips, IEEE Journal of Solid-State Circuits, Vol. 28(2), February 1993, pp. 138–145

[123] Motorola MC68HC705P6A, Advance Information.
http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC705P6A.pdf

[124] Microchip PIC16C5X Data Sheet, EPROM/ROM-based 8-Bit CMOS Micro-controller Series. *http://ww1.microchip.com/downloads/en/DeviceDoc/30453d.pdf*

[125] NEC µPD789104A, 789114A, 789124A, 789134A Subseries, 8-Bit Single-Chip Microcontrollers. *http://www.necel.com/nesdis/image/U14643EJ2V0UD00.pdf*

[126] GGB Industries Tungsten Probe Tips, T-4 Series. *http://www.ggb.com/t-4.html*

[127] GGB Industries, Model 12C. *http://www.ggb.com/12c.html*

[128] New Wave Research, QuikLaze FPD Laser Repair System.
http://www.new-wave.com/1nwrProducts/QuikLaze.htm

[129] Jean-Jacques Quisquater, David Samyde, ElectroMagnetic Analysis (EMA): Measures and Countermeasures for Smart Cards, International Conference on Research in Smart Cards, E-smart 2001, Cannes, France, September 2001, pp. 200–210

[130] Jesse Jenkins, CoolRunner-II CPLDs in Secure Applications, Xilinx White Paper, WP170. *http://www.xilinx.com/bvdocs/whitepapers/wp170.pdf*

[131] R. Aaron Falk, Near IR Absorption in Heavily Doped Silicon – An Empirical Approach, (ISTFA) International Symposium for Testing and Failure Analysis 2000

[132] Hamamatsu Infrared Vidicon Camera C2741-03.
http://sales.hamamatsu.com/assets/pdf/hpspdf/C2741-03HPD.pdf

[133] Gilway Engineering Catalog, High Intensity Infrared Source.
http://www.gilway.com/pdf/gold-coated.pdf

[134] Hypervision Visionary 2000 BEAMS.
http://www.hypervisioninc.com/v2000.html

[135] D.H. Habing, Use of Laser to Simulate Radiation-induced Transients In Semiconductors and Circuits, IEEE Transactions on Nuclear Science, Vol. 12(6), December 1965, pp. 91–100

[136] Allan H. Johnston, Charge Generation and Collection in p-n Junctions Excited with Pulsed Infrared Lasers, IEEE Transactions on Nuclear Science, Vol. 40(6), 1993, pp. 1694–1702

[137] Edward D. Palik, Handbook of Optical Constants of Solids, Orlando: Academic Press, 1985, pp. 547–569

[138] Newport 561/562 Series ULTRAlign Precision Multi-Axis Positioning System.
http://www.newport.com/store/product.asp?id=3159&lang=1

[139] Newport AD Series Ultra-Resolution Electrostrictive Actuators.
http://www.newport.com/store/product.asp?id=2898&lang=1

[140] K.S. Wills, T. Lewis, G. Billus, H. Hoang, Optical Beam Induced Current Applications For Failure Analysis of VLSI Devices, Proceedings International Symposium for Testing and Failure Analysis, 1990, p. 21.

[141] C. Ajluni, Two New Imaging Techniques Promise To Improve IC Defect Identification, Electronic Design, Vol. 43(14), July 1995, pp. 37–38.

[142] H.K. Heinrich, N. Pakdaman, J.L. Prince, G. Jordy, M. Belaidi, R. Franch, D.C. Edelstein, Optical Detection of Multibit Logic Signals at Internal Nodes in a Flip-chip Mounted Silicon Static Random-Access Memory Integrated Circuit, Journal of Vacuum Science & Technology, Microelectronics and Nanometer Structures, Vol. 10(6), November 1992, pp. 3109–3111

[143] Jan M. Rabaey, Digital Integrated Circuits: A Design Perspective, Prentice-Hall, 1995

[144] Silvaco TCAD Tools. *http://www.silvaco.com/products/TCAD.html*

[145] Vladimir V. Belyakov, Alexander I. Chumakov, Alexander Y. Nikiforov, Vyacheslav S. Pershenkov, Peter K. Skorobogatov, A.V. Sogoyan, Prediction of Local and Global Ionization Effects on ICs: The Synergy between Numerical and Physical Simulation, Russian Microelectronics, Vol. 32(2), March 2003, pp. 105–118

[146] Philips Semiconductors 80C51/87C51/80C52/87C52 8-bit Microcontroller Family. *http://www.semiconductors.philips.com/acrobat/datasheets/8XC51_8XC52_6.pdf*

[147] Microchip PIC16C6XX/7XX/9XX Programming Specifications. *http://ww1.microchip.com/downloads/en/DeviceDoc/30228k.pdf*

[148] Atmel AT89C51, 8-bit Microcontroller with 4K Bytes Flash. *http://www.atmel.com/dyn/resources/prod_documents/doc0265.pdf*

[149] Microchip PIC16F62X Data Sheet, Flash-Based 8-Bit CMOS Microcontroller. *http://ww1.microchip.com/downloads/en/DeviceDoc/40300c.pdf*

[150] Microchip PIC16C62X Data Sheet, EPROM-Based 8-Bit CMOS Microcontrollers. *http://ww1.microchip.com/downloads/en/DeviceDoc/30235j.pdf*

[151] 3rd Generation Smartcard Project ‘G3Card’. *http://www.g3card.com/*

[152] Chad Golden, Toner Cartridge Computer Chip Usage and the Impact on the Aftermarket, Static Control Components, Recharger Magazine, December 2002, pp. 36–46.

[153] AVR Freaks Forum. *http://www.avrfreaks.net/phorum/viewtopic.php?t=19790&postdays=0&postorder=asc&start=75*

[154] Richard Clayton, Mike Bond, Experience Using a Low-Cost FPGA Design to Crack DES Keys, Cryptographic Hardware and Embedded Systems Workshop (CHES-2002), LNCS, Vol. 2523, Springer-Verlag, 2002, pp. 579–592

[155] Simon W. Moore, Ross J. Anderson, Markus G. Kuhn, Improving Smartcard Security using Self-Timed Circuit Technology, Fourth AciD-WG Workshop, Grenoble, 2000

[156] Simon W. Moore, Ross J. Anderson, Paul Cunningham, Robert Mullins, George Taylor, Improving Smartcard Security using Self-Timed Circuits, Asynch 2002, Proceedings published by IEEE Computer Society Press

[157] Simon Moore, Ross Anderson, Robert Mullins, George Taylor, Balanced Self-Checking Asynchronous Logic for Smart Card Applications, Journal of Microprocessors and Microsystems. Special Issue on Asynchronous System Design, 2003