Breakthrough silicon scanning discovers backdoor in military chip (DRAFT of 05 March 2012)

Sergei Skorobogatov
University of Cambridge
Cambridge, UK
sps32@cam.ac.uk

Christopher Woods
Quo Vadis Labs
London, UK
chris@quovadislabs.com

Abstract. This paper is a short summary of the first real world detection of a backdoor in a military grade FPGA. Using an innovative patented technique we were able to detect and analyse in the first documented case of its kind, a backdoor inserted into the Actel/Microsemi ProASIC3 chips. The backdoor was found to exist on the silicon itself, it was not present in any firmware loaded onto the chip. Using Pipeline Emission Analysis (PEA), a technique pioneered by our sponsor, we were able to extract the secret key to activate the backdoor. This way an attacker can disable all the security on the chip, reprogram crypto and access keys, modify low-level silicon features, access unencrypted configuration bitstream or permanently damage the device. Clearly this means the device is wide open to intellectual property theft, fraud, re-programming as well as reverse engineering of the design which allows the introduction of a new backdoor or Trojan. Most concerning, it is not possible to patch the backdoor in chips already deployed, meaning those using this family of chips have to accept the fact it can be easily compromised or it will have to be physically replaced after a redesign of the silicon itself.

Keywords: Hardware Assurance; silicon scanning; side-channel analysis; hardware Trojans and backdoors

1 Introduction

With the globalisation of semiconductor manufacturing, integrated circuits become vulnerable to malevolent activities in the form of Trojan and backdoor insertion [1]. An adversary can introduce Trojans into the design during a stage of fabrication by modifying the mask at a foundry or fab. It can also be present inside third parties' modules or blocks used in the design. Backdoors could be implemented by malicious insiders at the design house. In this paper we demonstrate how a deliberately inserted backdoor and additional functionalities can be found in a highly secure FPGA (field-programmable gate array) chip used in both military and sensitive industrial applications. Trojans can be found using a similar approach, altering the way the scanning is performed slightly. To our knowledge, this is the first documented case of finding a deliberately inserted backdoor in a real world chip.

Several Trojan detection approaches have been proposed in recent years. These can be divided into three major categories. One is full reverse engineering of the chip

which gives an in-depth analysis of the chip [2]. However, this has some drawbacks – it is an extremely expensive and time consuming operation, and it will not work for cases where the Trojan is present only in a small fraction of chips. A second category is an attempt to activate the Trojan by applying test vectors and comparing the responses with expected responses [3][4][5]. This might not work in situations where the Trojan is activated under rare conditions. For modern complex circuits it is close to impossible to enumerate all states. In addition, this approach will not detect Trojans designed to leak the information rather than take control of the hardware [6]. The final category uses side-channel analysis to detect Trojans by measuring circuit parameters such as power consumption, electro-magnetic emissions and timing analysis. These methods can be used against golden samples [7][8] or within the same integrated circuit (IC) to minimise the variations between samples [9]. However, the effectiveness of side-channel analysis methods greatly depends on the sensitivity of the measuring equipment [10].

One of the most widely used approaches in Trojan and backdoor detection is to employ differential power analysis (DPA) techniques [11] to detect any abnormalities in the device operation. However, due to the latency introduced by the setup and substantial noise of the acquisition equipment, it normally takes a very long time to scan silicon chips. With modern devices such as FPGAs, it could be unfeasible to detect any Trojans or backdoors with DPA techniques. We used a new sensing technique which detects tiny variations in the device operation and is thus able to detect small variations which are well below the noise level in a standard DPA setup.

If a bug is found in firmware programmed into an FPGA then it can be rectified by a firmware update. However, if the Trojan or backdoor is present in the silicon itself, then there is no way to remove the bugs other than replacing all the affected silicon chips, as has happened several times with bugs found in Intel CPUs. The cost of such an operation is enormous and can seriously affect an organisation's revenue.

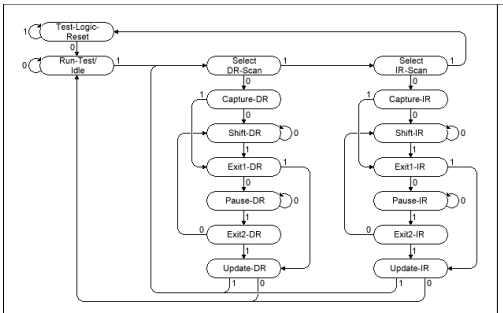
If a potential attacker takes control of the FPGA device, he can cause a lot of damage to the device. For example, he can erase or even physically destroy the FPGA by uploading a malicious bitstream that will cause a high current to pass through the device and burn it out. Using the backdoor, an attacker can extract the intellectual property (IP) from the device and make some changes to the firmware, inserting new Trojans into its configuration. That would provide a wide range of capabilities in carrying out more sophisticated attacks at a later stage.

In a search of the ideal target we decided to test the Actel/Microsemi ProASIC3 (PA3) A3P250 device because of its high security specifications and wide use in military and industrial applications. Actel, who developed PA3 devices, market them as chips which *'offer one of the highest levels of design security in the industry*' [12].

This paper is organised as follows. Section 2 gives a brief introduction into chip access and scanning approaches. Section 3 introduces the experimental setup, while Section 4 sets out our results. Section 5 discusses limitations and possible improvements. The impact of the research is discussed in the concluding section.

2 Background

With the growing complexity of integrated circuits the importance of post production testing and functional verification is growing. This is necessary to address the issues in failure analysis and to perform design verification for correctness, and to eliminate inevitable bugs [13]. The majority of chip manufacturers use the JTAG (Joint Test Action Group) interface as a standard port for IC testing [14]. However, until recently it was primarily used for boundary scan testing rather than internal IC testing. In the early 2000s the JTAG specification was expanded with programming abilities and security features to meet the FPGA market demands [15]. However, even before then chip manufacturers were using the expanded JTAG usually referred as IEEE 1149.x. This expansion was not standardised and for most chips was kept confidential. In that respect, the knowledge of the test interface being a JTAG did not give any advantage to the outsider over a proprietary test interface. However, this allowed chip manufacturers to use standard JTAG implementation libraries without compromising on the security of their chips. It was important for manufacturers to use undocumented or disguised commands for granting access to the JTAG or test interface, because in some chips it provided access to the internal memory, usually holding the end user IP and secret data [16].




```
PROCEDURE IS_SECOK USES GV,DO_EXIT;
    IF ( ! (SECKEY_OK==0) ) THEN GOTO SECOK;
    STATUS = -35;
    PRINT "Error, pass key match failure";
    CALL DO_EXIT;
    SECOK:
    LABEL_SEPARATOR = 0;
ENDPROC;

PROCEDURE DO_CHECK_R USES GV,DO_EXIT,DO_READ_SECURITY;
    CALL DO_READ_SECURITY;
    IF ( ! (ULARE==0) ) THEN GOTO Label_70;
    STATUS = -37;
    PRINT "FPGA Array Encryption is not enforced.";
    PRINT "Cannot guarantee valid AES key present in target device.";
    PRINT "Unable to proceed with Encrypted FPGA Array verification.";
    CALL DO_EXIT;
    Label_70:
    IF ( ! (ULARD==1) ) THEN GOTO SKIPRCHK1;
    STATUS = -30;
    PRINT "FPGA Array Verification is protected by pass key.";
    PRINT "A valid pass key needs to be provided.";
    CALL DO_EXIT;
    SKIPRCHK1:
    IF ( ! (ULARD==0) ) THEN GOTO Label_71;
    CHKSEC = 0;
    Label_71:
    LABEL_SEPARATOR = 0;
ENDPROC;
```

Fig. 1: (a) JTAG TAP state machine, (b) example of STAPL code subroutines

The JTAG interface is operated via test access port (TAP) pins which control the state machine (Figure 1a). It has two registers – IR (instruction register) and DR (data register) into which the serial data can be shifted and then executed. The IR registers must be selected first and then, depending on the command, DR data shifted in. The length of the IR register varies from chip to chip and normally lies

between 4 and 32 bits. Some commands do not involve the DR register, for others its length could be many thousands of bits.

For many chips, and especially for secure microcontrollers and secure FPGAs, the commands and data fields of JTAG registers are not documented. However, an inquisitive attacker can gain most of this information from development kits supplied by a particular device's chip manufacturer. Even when the availability of such kits is restricted by the manufacturer, their clones can be found in the third world. For FPGA chips the task of gathering more information about JTAG commands was simplified by the introduction of a special high level test language called STAPL (Standard Test and Programming Language) [17]. All the commands and data fields in the programming file compiled by design tools are easily identifiable with both subroutines and meaningfully named variables (Figure 1b).

Knowing all the JTAG commands is not sufficient to search for backdoors. Firstly, the obtained list could be incomplete because the STAPL file is compiled only with commands which serve only a particular task. Secondly, although subroutines, functions and variables are meaningfully named, the IR level commands are not explained and usually remain as numbers. That complicates the reverse engineering of the JTAG functionality. What adds to the complexity is the sequence of commands. For complex devices it will not be just one command executed for a particular function, but a series of commands mixed with data. Each command could be not solely IR or IR+DR, but an endless list of possible combinations such as IR+IR, IR+DR+DR, IR+DR+IR+DR and so forth.

Searching for Trojans could represent an easier task, because in that case the design is known as well as its likely implementation in silicon. This operation is usually performed by the chip manufacturer or its subcontractors. However, from an attacker's point of view, there is not much difference between Trojans and backdoors as he is looking for any potential vulnerability within the silicon chip.

The following sections describe how we first approached the problem of finding all the active commands, and then how we performed an efficient scanning over the large field of possible data.

3 Experimental method

As a target for our experiments we chose the Actel/Microsemi ProASIC3 (PA3) A3P250 device [18] for many reasons. Firstly, it has high security specifications and is positioned as the device with highest security protection in the industry. Actel who developed PA3 chips market them as devices which *'provide the most impenetrable security for programmable logic designs'* [19][20]. Secondly, PA3 chips are widely used in military and industrial applications especially in critical systems. Therefore, without doubt PA3 devices posed suitable challenges for this research. Any outcome occurring from analysing this device will have a greater impact and will be more useful compared to low-end security chips such as normal microcontrollers or standard FPGAs.

Initially, we analysed the chip with standard design tools from Actel – Libero IDE and FlashPro. The sample of A3P250 device was connected to a standard Actel FlashPro3 programmer. All of the JTAG operations are undocumented for PA3, however, using Actel development software we were able to generate series of STAPL files which we analysed for the commands used for different operations. Once we learned the JTAG communication we moved onto exploring the field of undocumented features. For that we built a special test board with master JTAG interface and simple functions controlled by PC software via an RS-232 interface for convenience (Figure 2a). The PA3 chip was placed into a ZIF socket for easier handling. During that stage we gathered information about the command field and data registers.

The next step was to determine which commands have data fields and measure the size of DR registers. Then we used a classic DPA setup to analyse the side-channel emission from the PA3 devices during decryption and to access operations as well as other undocumented commands. We constructed a simple prototype board with a ZIF socket for the A3P250 device (Figure 2b) and connected it to our test board which was providing some additional triggering functions for the oscilloscope. The power consumption was measured via a 20 Ω resistor in V_{CC} core supply line with the Agilent 1130A differential probe and acquired with the Agilent MSO8104A digital storage oscilloscope. Then the waveforms were analysed using MatLab software with our own proprietary program code.

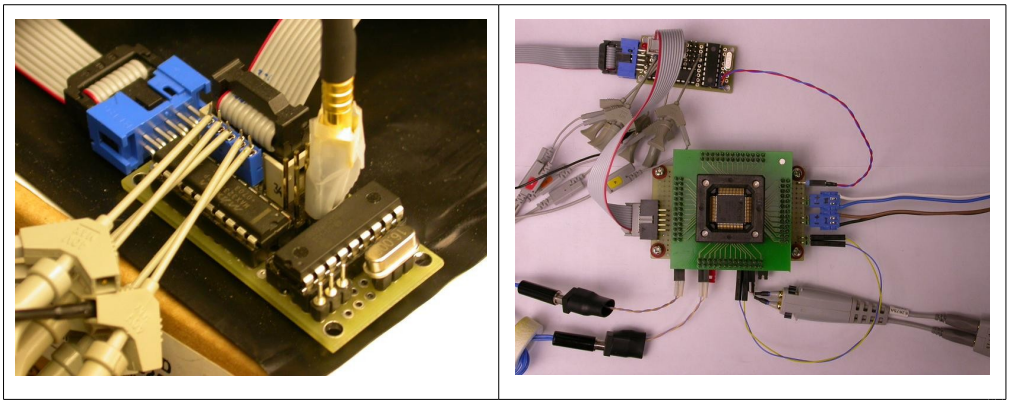


Fig. 2: Test setup: (a) control board, (b) test board with DPA setup

We tried all available JTAG command fields in different combinations and observed all the traces scanned with DPA. In this way we were able to separate commands with different functions. Then the unknown commands were tested with different data fields, while we observed the response and tried to understand their function. DPA is a good approach to find normal commands; however it cannot calculate their functionality because of high noise and the number of traces required.

In the next set of experiments we used PEA technology (described in our paper [21]) to achieve better signal-to-noise ratio (SNR) in an attempt to better understand the functionality of each unknown command. Some operations were found to have robust silicon level DPA countermeasures. For example, the Passkey is documented as another layer of security protection on top of the AES encryption in PA3 to prevent IP cloning. Some DPA countermeasures found in the Passkey protection include very good compensation of any EM leakage and broadband spectrum spreading of side-channel emissions for the bit comparison leakage; internal unstable clock; high noise resulting in SNR of at the best –20 dB. The first generation of the sensor is presented in Figure 3a while the second generation is in Figure 3b. In the end we used a silicon scanning technique based on PEA pioneered by our project sponsor, combined with a classic DPA setup (resistor in power line, differential probe, oscilloscope, PC with MatLab). Nevertheless scanning for a backdoor was not a simple process.

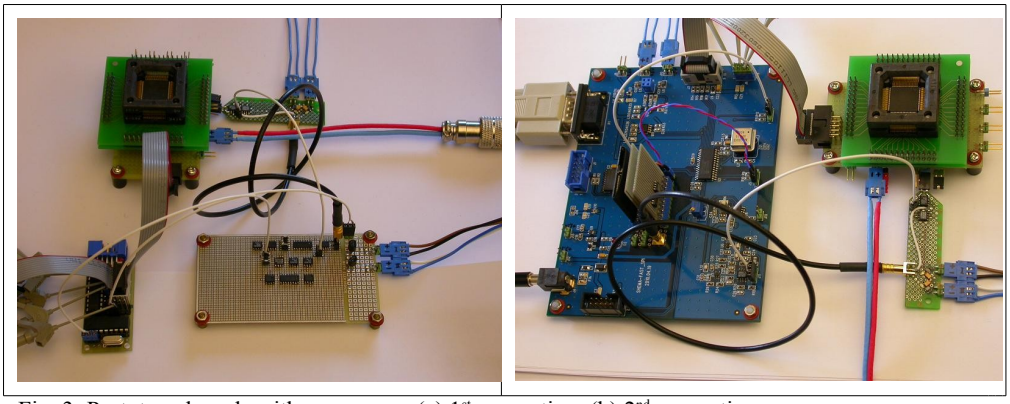


Fig. 3: Prototype boards with our sensor: (a) 1st generation, (b) 2nd generation

4 Results

Scanning the JTAG command field for any unknown commands by checking the length of the associated DR register revealed an interesting picture. There were plenty of commands for which the associated DR register has a length different from one, hence, used by the JTAG engine. Figure 4a shows some of these registers with the light ones being known from STAPL file analysis, and the dark ones showing newly discovered registers. Not only that, but some registers were impossible to update with a new data suggesting that these registers were representing a ROM (Read-Only Memory) (Figure 4b). This did make some sense as we learned about FROW memory from the STAPL file, from which only one row was actually read, but three address bits allowed eight rows to be accessed. All those hidden and non-updatable registers were found to be imprinted into certain locations in FROW memory. However, every single PA3 chip has unique values stored in FROW and, hence, in hidden registers suggesting that this memory was

initialised at a factory and then locked against overwriting. Now we knew for sure that there is some hidden functionality in the PA3 chips.

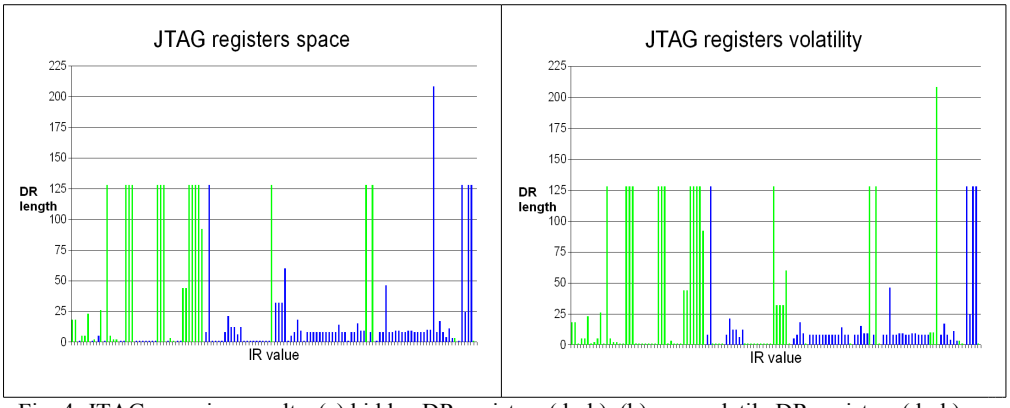


Fig. 4: JTAG scanning results: (a) hidden DR registers (dark), (b) non-volatile DR registers (dark)

Although they do not have any specialised DPA countermeasures, the PA3 devices are at least 100 times harder to attack using DPA than non-protected conventional microcontrollers such as PIC, AVR, MC68HC, MSP430 etc. The robust hardware design features are complemented with the total lack of information about JTAG engine operation, hardware implementation and commands. That makes any attacks on the PA3 chips quite a challenging task. Figure 5a shows the result obtained by comparing single traces for different input data. Averaging over 4096 traces gives a pretty nice result but takes a couple of minutes to acquire (Figure 5b). As can be seen, for single traces the noise overshadows any useful signal with SNR being at the best –20 dB. The FFT spectrum of the power trace does not have any characteristic peaks (Figure 6a) and filtering will not be very effective for substantially improving DPA results.

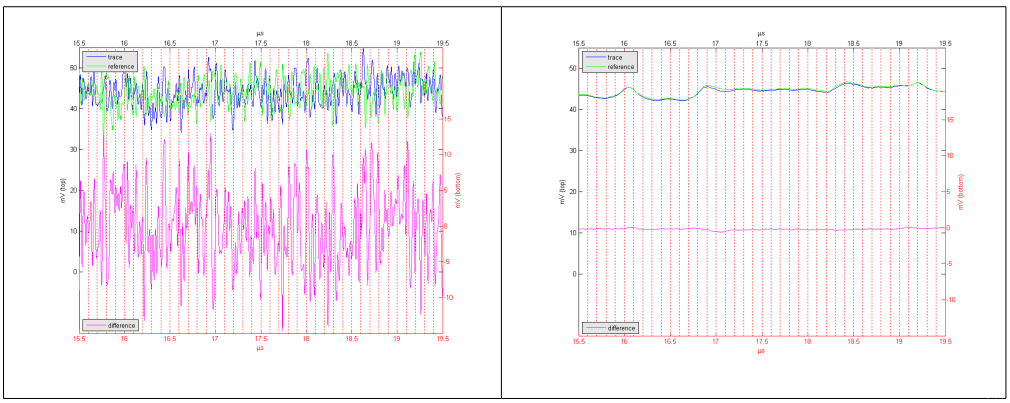


Fig. 5: Power analysis results on PA3: (a) single trace difference, (b) averaging over 4096 traces

The noise can be reduced by using frequency locking technique. There are publications on the successful use of these techniques on FPGAs [22]. That way the

timing jitter between traces can be reduced to approximately ten degrees of the phase shift at 19.7 MHz (Figure 6b). However, on the other hand this injects a strong carrier frequency which needs to be filtered out to avoid any influence on the power analysis results. Despite good synchronisation and triggering results we did not observe any improvements compared to the standard DPA setup because of a very strong presence of a 19.7 MHz signal in the power trace which we were unable to eliminate of completely.

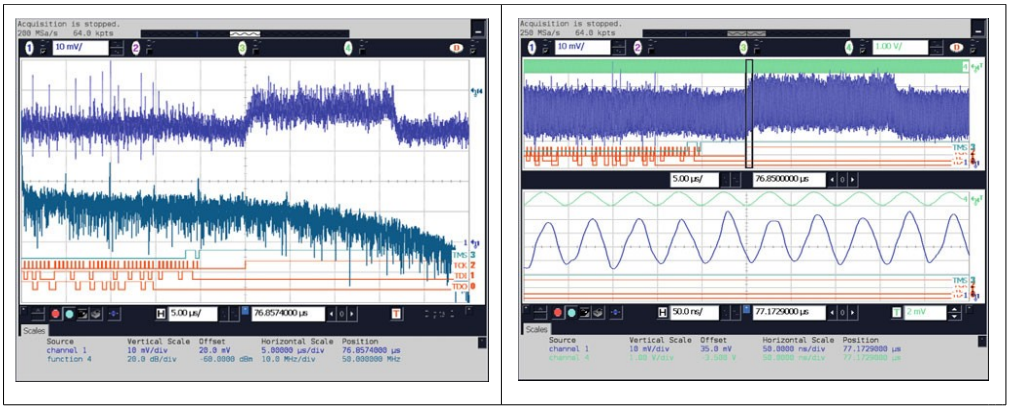


Fig. 6: Power traces from PA3: (a) FFT spectrum, (b) frequency locking of internal oscillator

Table 1 summarises security protection levels in the PA3 devices according to the findings from our research. The Passkey offers the highest level of reversible protection mechanism while the Permanent lock should be used as the last resort and will turn the device into a one-time programmable (OTP) chip, so that in the event of a bug in the design, the Permanent lock bricks the chip and renders it non-usable, meaning it has to be physically replaced. However, despite it being a seemingly ultimate protection mechanism, the Permanent lock has some physical security flaws. We found it vulnerable to some fault injection attacks, but this lies outside of this paper scope as it has no relation to the backdoor.

Table 1. Security protection levels in PA3

Secure object	Read Access	Write Access	Secure Lock/Fuse	Encryption Option	Security Level
FROM (User Flash)	Y	Y	Y	Y	Medium
FPGA Array	N	Y	Y	Y	High
User Row	Y	Y	N	N	Low
AES key	Indirectly	Y	Y	N	Medium
Passkey	N	Y	Y	N	Very High
Permanent Lock	N	Y	N	N	Medium

Various DPA techniques were attempted to extract the Passkey, however, we were unable to get even a single bit in two weeks time using our off-the-shelf DPA equipment (oscilloscope with differential probe and PC with MatLab). The traces that appeared using DPA accounted for many functions including memory access, AES, Passkey and other yet to be learned functions. Even for poorly protected against DPA attacks AES encryption it would require many traces to be averaged to get reliable correlation with key bits (Figure 5). PEA approach allowed the key bits to be guessed in real time and with a very good correlation with the key bits. The outstanding sensitivity of the PEA is owed to many factors. One of which is the bandwidth of the analysed signal, which for DPA, stands at 200 MHz while in PEA at only 20 kHz. This not only results in much lower noise, which is proportional to the square root of the bandwidth, but the cost of the acquisition hardware becomes several orders of magnitude lower. This also impacts on the latency thus allowing real-time analysis, because the signal produced for the analysis has almost 100% correlation with the key bits (Figure 7a, Ch3 – power trace, Ch1 – PEA signal, Ch2 – demodulated signal). This makes extraction time extremely fast. All that needs to be done in the end for the key extraction is to demodulate the signal and compare it with the reference peak. This can be easily performed by a simple one-dollar microcontroller with on-chip ADC.

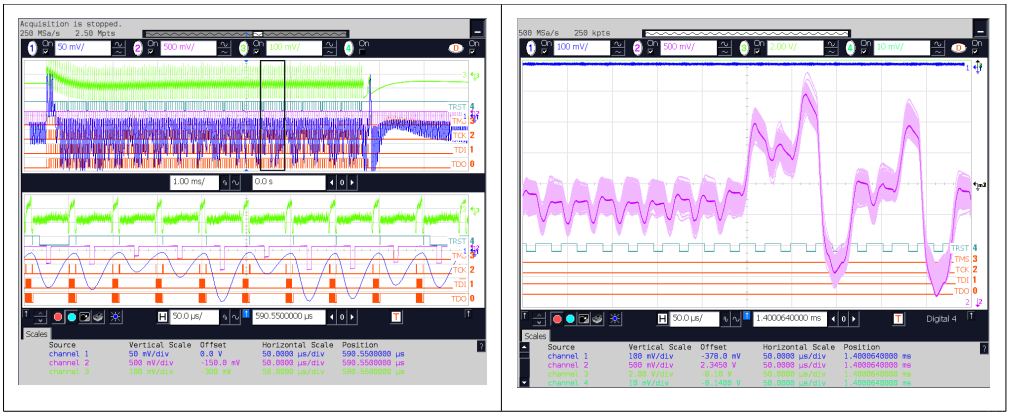


Fig. 7: Analysis of PA3 using PEA: (a) scan for AES key, (b) scan for passkey

With the analysis of JTAG commands, one particular function was requesting a 128-bit key with the similar low-leakage DPA resistance property as the Passkey. It also had robust countermeasures that proved to be DPA resistant. In addition to an unstable internal clock and high noise from other parts of the circuit, the Passkey and backdoor access verification had their side-channel leakage reduced by a factor of 100. This was likely to be achieved through using a well compensated silicon design together with ultra-low-power transistors instead of standard CMOS library components. In addition, the useful leakage signal has a spread spectrum with no characteristic peaks in frequency domain, thus making narrow band filtering useless. We used the similar PEA approach to extract both the Passkey and the Backdoor key by looking for any notable changes in the response from our sensor

for correct and incorrect guesses (Figure 7b). However, due to much more robust DPA countermeasures it took us approximately one day to achieve this using simple PEA hardware. According to our calculations and experiments, finding the key using a classic DPA setup would take approximately 2,000 years to complete. Further investigation revealed that this is a backdoor function with the key capable of unlocking many of the undocumented functions, including IP access and reprogramming of secure memory.

At this point we went back to those JTAG registers which were non-updatable as well as FROW to check whether we could change their values. Once the backdoor feature was unlocked, many of these registers became volatile and the FROW was reprogrammable as a normal Flash memory. Actel has a strong claim that *'configuration files cannot be read back via JTAG or any other method'* in the PA3 and in their other latest generation Flash FPGAs [18]. Hence, they claim, they are extremely secure because the readback access is not implemented. We discovered that in fact Actel did implement such an access, with a special key used for activation.

Alongside this backdoor there is another layer of security in the guise of data permutation to obscure information and make IP extraction less feasible. This can also be dealt with using a simple brute force attack, because permutation functions do not withstand differential cryptanalysis as every single bit change at the input results in a single-bit change at the output. Our experiments showed how some information can be found via systematic testing of device operations. Through this method, for example, we found the correspondence between bits in the 832-bit verification data and bits in the data bus.

5 Implications and further improvements

Many countermeasures are designed to defeat high end oscilloscopes and their known noise, latency and signal issues. These countermeasures prevent themselves from being broken in an affordable time through suppressing the signal or by bringing it to a higher noise level. Our approach through the use of bespoke hardware and the removal of the oscilloscope from the testing process, is designed to have the sensitivity to detect even the smallest variation in signal, which then allows more detailed analysis. The setup with which we achieved these eye-opening results is in its most basic form, employing a single pipeline (one channel).

Having taken this technology to proof of concept, we would like to develop it by building a multi-pipeline system consisting of 100 channels as well as new, more efficient hardware for our probes, with the aim of further improving sensitivity and speed by a factor of 10. We firmly believe that defeating these more secure DPA countermeasures is a very real and achievable expectation with this increase in capability planned for the next generation of our technology. Using a low-noise side-channel measurement setup with a carefully designed probe a 10× further improvement can be achieved. Further improvements can be done to the scanning

algorithm itself thus improving the effectiveness by a further 10×. All these improvements can bring the analysis time down to hours or even minutes.

We noticed that FPGA security relies heavily on obscurity. This ranges from the lack of any documentation on the JTAG access interface, and absence of information on the internal operations, down to the data formats. This works well unless an attacker is determined to discover all this information on their own. Alternatively, more information can be gained through the analysis of development tools and programming files for some chips. That certainly raises a concern about the amount of information a potential attacker can gain through development kits.

Some DPA and design cloning countermeasures might be ineffective in light of efficient silicon scanning techniques. For example, Intrinsic ID offers a software level solution for secure storage of crypto keys [23]. However, for an attacker who has full access to the chip through a backdoor and is capable of extracting the bitstream, localising and defeating the protection mechanism will be trivial. He will still have to understand the proprietary bitstream encoding, however, this can be achieved in several ways from reverse engineering the development software, through active attacks on chips, to reverse engineering the FPGA chip itself. Therefore, solutions with silicon-level fingerprinting using physical unclonable functions (PUF) will be ineffective in the presence of backdoors.

One could possibly argue that the backdoor we discovered is a bug or something overlooked by the developers. However, this is not the case as we performed intensive investigation into this problem and found proof that the backdoor was deliberately inserted and even used as a part of the overall security scheme. We cannot disclose all of these findings at present due to a confidentiality agreement.

6 Conclusion

Our experiments had achieved the affordable time for scanning of two weeks. As a result we were able to locate and exploit undocumented backdoor in the Actel ProASIC3 chip positioned as industry's highest security device. To our knowledge this is the first documented case of a backdoor inserted in real world device with critical applications. Not only can a poorly protected AES key be extracted from the PA3 chips in no time and with minimal effort, but the Passkey which was believed to be unbreakable and which was robust against DPA attacks can also be extracted.

The discovery of a backdoor in a military grade chip raises some serious questions about hardware assurance in the semiconductor industry. When you use and buy an embedded system or computer it is assumed, wrongly in our opinion, that the hardware is completely devoid of any vulnerabilities. We investigated the PA3 backdoor problem through Internet searches, software and hardware analysis and found that this particular backdoor is not a result of any mistake or an innocent bug, but is instead a deliberately inserted and well thought-through backdoor that is crafted into, and part of, the PA3 security system. We analysed other Microsemi/Actel products and found they all have the same deliberate backdoor.

Those products include, but are not limited to: Igloo, Fusion and Smartfusion. The PA3 is heavily marketed to the military and industry and resides in some very sensitive and critical products. From Google searches alone we have found that the PA3 is used in military products such as weapons, guidance, flight control, networking and communications. In industry it is used in nuclear power plants, power distribution, aerospace, aviation, public transport and automotive products. This permits a new and disturbing possibility of a large scale Stuxnet-type attack via a network or the Internet on the silicon itself. If the key is known, commands can be embedded into a worm to scan for JTAG, then to attack and reprogram the firmware remotely. The backdoor is close to impossible to fix on chips already deployed because, unlike software bugs in a PC Operating System, you cannot issue a patch to fix this. Instead one has to replace all the hardware which could be extremely expensive. It may simply be a matter of time before this backdoor opportunity, which has the potential to impact on many critical systems, is exploited.

Having a security related backdoor on a silicon chip jeopardises any efforts of adding software level protection. This is because an attacker can use the underlying hardware to circumvent the software countermeasures. Using PUFs is not likely to offer much help as the firmware that calculates them could be extracted and then reverse engineered to defeat the protection layer. Using encryption as an additional protection layer does not always help. Moreover, it could make things worse, as in the PA3, where the AES key can be extracted in less than a second's time [21] compared to hours required for Passkey extraction.

A debug port, factory test interface or JTAG can all potentially be used as points to scan the silicon chip for backdoors or Trojans. Most chips manufactured these days have at least one of these features present. By abusing the PEA technology to understand functionality and to extract keys, a new and inviting area of cyber warfare may be started.

Until the development of the techniques pioneered by our research sponsor, it has been unfeasible to test real silicon chips for Trojans or backdoors. Using a low cost system it becomes possible to independently test silicon for backdoors and Trojans in a matter of weeks. It would take many years to perform the same task using standard DPA. Most silicon chips are now designed and made abroad by third parties. Is there any independent way to evaluate these products that are used in critical systems?

References

[1] M. Tehranipoor, F. Koushanfar: A survey of hardware Trojan taxonomy and detection. IEEE Design and Test of Computers, 2010

[2] R. Torrance, D. James: The State-of-the-Art in IC Reverse Engineering. Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2009, pp. 363-381

[3] S. Jha and S. K. Jha: Randomization Based Probabilistic Approach to Detect Trojan Circuits. Proc. 11th IEEE High Assurance System Engineering Symp., 2008, pp. 117-124

[4] M. Banga and M. Hsiao: A Region based Approach for the Identification of Hardware Trojans. IEEE Int. Workshop on Hardware-Oriented Security and Trust (HOST), 2008, pp. 40-47

[5] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty: Towards Trojan-free Trusted ICs: Problem Analysis and Detection Scheme. In: Design, Automation and Test in Europe, 2008. DATE'08, pp. 1362-1365, 10-14 March 2008

[6] X. Wang, M. Tehranipoor, and J. Plusquellic: Detecting Malicious Inclusions in Secure Hareware: Challenges and Solutions. IEEE Int. Hardware-Oriented Security and Trust (HOST), 2008

[7] D. Agrawal, S. Baktir, and D. Karakoyunlu, P. Rohatgi, and B. Sunar: Trojan Detection using IC Fingerprinting. IEEE Symp. on Security and Privacy (SP), pp. 296-310, 2007

[8] Y. Jin and Y. Makris: Hardware Trojan Detection using Path Delay Fingerprint. IEEE Int. Workshop on Hardware-Oriented Security and Trust (HOST), 2008

[9] D. Du, S. Narasimhan, R. Chakraborty and S. Bhunia: Self-Referencing: A Scalable Side-Channel Approach for Hardware Trojan Detection. Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2010

[10] R. Rad, M. Tehranipoor, J. Plusquellic: A Sensitivity Analysis of Power Signal Methods for Detecting Hardware Trojans under Real Process and Environmental Conditions. IEEE. Trans. in VLSI, vol 18, pp. 1735 - 1744, 2009

[11] P. Kocher, J. Jaffe, B. Jun: Differential Power Analysis. CRYPTO'99, LNCS, Vol. 1666, Springer-Verlag, 1999, pp. 388–397

[12] Design Security in Nonvolatile Flash and Antifuse FPGAs, Security Backgrounder
http://www.actel.com/documents/DesignSecurity_WP.pdf

[13] Introduction to Hardware Security and Trust, Eds: Mohammad Tehranipoor and Cliff Wang, Springer, September 2011

[14] JTAG Boundary scan. IEEE Std 1149.1-2001

[15] JTAG Programming specification. IEEE 1532-2002

[16] Jean DaRolt, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre: New security threats against chips containing scan chain structures. HOST 2011, pp. 110

[17] Actel, ISP and STAPL, Application Note AC171
http://www.actel.com/documents/ISP_STAPL_AN.pdf

[18] Actel ProASIC3/E Production FPGAs, Features and Advantages
http://www.actel.com/documents/PA3_E_Tech_WP.pdf

[19] Actel ProASIC3 Datasheet, ProASIC3 Flash Family FPGAs
http://www.actel.com/documents/PA3_DS.pdf

[20] ProASIC3 Frequently Asked Questions, Actel Corporation, Mountain View, CA 94043-4655 USA. http://www.actel.com/documents/pa3_faq.html

[21] Removed to comply with anonymity requirement for submission

[22] S. Skorobogatov: Synchronization method for SCA and fault attacks. Journal of Cryptographic Engineering (JCEN), Vol.1, No.1, Springer, 2011, pp.71-77

[23] Intrinsic ID, Quiddikey on ProASIC3 FPGAs
http://www.intrinsic-id.com/quiddikey_on_Actel_FPGA.htm