Breakthrough silicon scanning discovers backdoor in military chip

Sergei Skorobogatov[1] and Christopher Woods[2]

[1] University of Cambridge, Computer Laboratory, Cambridge, UK
sps32@cam.ac.uk
[2] 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 for accessing FPGA configuration. The backdoor was found amongst additional JTAG functionality and exists on the silicon itself, it was not present in any firmware loaded onto the chip. Using Pipeline Emission Analysis (PEA), our pioneered technique, we were able to extract the secret key to activate the backdoor, as well as other security keys such as the AES and the Passkey. This way an attacker can extract all the configuration data from 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 (IP) 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 they can be easily compromised or will have to be physically replaced after a redesign of the silicon itself.

Keywords: Hardware Assurance, silicon scanning, side-channel analysis, silicon Trojans and backdoors, PEA

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. From the attacker's point of view there is not much difference between Trojans and backdoors, because in most cases the device would be analysed as a black box with very limited information provided by the manufacturer. Neither would it be known who inserted the undocumented features of additional capabilities or at what stage of the process this occured.

In a search for the ideal target we decided to test the Actel/Microsemi ProASIC3 A3P250 device because of its high security specifications and wide use in military and industrial applications. According to the chip manufacturer: *"Low power flash devices are unique in being reprogrammable and having inherent resistance to both invasive and noninvasive attacks on valuable IP"* [2].

In this paper we demonstrate how a deliberately inserted backdoor and additional functionalities can be found in the 'highly secure' Actel/Microsemi ProASIC3 Flash FPGA (field-programmable gate array) chip used in both military and sensitive industrial applications. Actel, who developed ProASIC3 devices, market them as chips which *"offer one of the highest levels of design security in the industry"* [3]. These FPGAs are unique by being low-power, live on power-up and inherently secure as no configuration data is stored outside the device: *"In contrast to SRAM-based FPGAs, ProASIC3 configuration files cannot be read back via JTAG or any other method"* [4].

All our experiments were carried out on the Actel/Microsemi ProASIC3 A3P250 Flash FPGA device [2]. Fabricated with a 0.13 μm process with 7 metal layers, this chip incorporates 1,913,600 bits of bitstream configuration data. According to the manufacturer's documentation on this chip: *"Even without any security measures (such as FlashLock with AES), it is not possible to read back the programming data from a programmed device. Upon programming completion, the programming algorithm will reload the programming data into the device. The device will then use built-in circuitry to determine if it was programmed correctly"* [2]. Our research revealed that there is some hidden functionality inside the JTAG controller of this chip and one of the functions is the covert access to the configuration data. The JTAG controller itself is a part of the silicon design as in all FPGA chips and cannot be changed after the chip is manufactured.

To our knowledge, this is the first documented case of finding a backdoor inserted in a real world chip. When we talk about backdoors we treat them differently from other undocumented features most chip manufacturers insert in their devices for factory testing and debug purposes. The difference between a backdoor and an undocumented command is the ability to gain access to the programmed user IP when it is not supposed to be possible. Undocumented commands are known to exist in JTAG for failure analysis or for debugging but they are not designed to circumnavigate the security scheme of the device. The dictionary gives the following definition: *"backdoor – an undocumented way to get access to a computer system or the data it contains"* [5]. This is exactly what we found in the third generation of Actel/Microsemi Flash FPGA chips. The same approach can be used to find Trojans, altering the way the scanning is performed slightly.

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 [6]. 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. The second category is an attempt to activate the Trojan by applying test vectors and comparing the responses with expected responses [7–9]. This might not work in situations where the Trojan is activated under rare conditions. For modern complex circuits it is close to impossible to verify all states. In addition, this approach will not detect Trojans designed to leak the information rather than take control of the hardware [10]. 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 [11, 12] or within the same integrated circuit (IC) to minimise the variations between samples [13]. However, the effectiveness of side-channel analysis methods greatly depends on the sensitivity of the measuring equipment [14].

One of the most widely used approaches in Trojan and backdoor detection is to employ differential power analysis (DPA) techniques [15] to detect any abnormalities in the device operation. However, due to the latency introduced by the setup and the 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 infeasible 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 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.

Greater danger could come from a new and disturbing possibility of a large scale remote 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. This cannot be excluded since the manufacturer specifically designed such remote access feature for ProASIC3 and other Flash FPGA devices: *"Military ProASIC3/EL devices with AES-based security allow for secure, remote field updates over public networks such as the Internet, and ensure that valuable IP remains out of the hands of system over-*

builders, system cloners, and IP thieves. The contents of a programmed device cannot be read back, although secure design verification is possible" [16].

This paper is organised as follows. Section 2 gives a brief introduction into chip access and scanning approaches, and FPGA security. 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 [17]. The majority of chip manufacturers use the JTAG (Joint Test Action Group) interface as a standard port for IC testing [18]. 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 [19]. 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 [20].

The JTAG interface is operated via test access port (TAP) pins which control the state machine (Fig. 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) [21]. All the commands and data fields in the programming file compiled by design tools

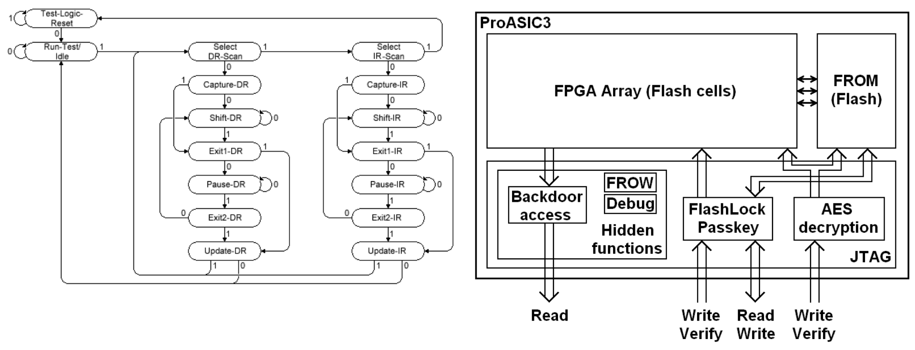

Fig. 1. (a)JTAG TAP state machine, (b)Simplified ProASIC3 security

are easily identifiable with both subroutines and meaningfully named variables. An example of security state verification procedure is given in the Appendix.

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 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.

As a target for our experiments we chose the Actel/Microsemi ProASIC3 A3P250 device [4] 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 ProASIC3 chips market them as devices which *"provide the most impenetrable security for programmable logic designs"* [16, 22]. Secondly, ProASIC3 chips are widely used in military and industrial applications especially in critical systems. Therefore, without doubt, ProASIC3 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 the results obtained from low-end security chips such as normal microcontrollers or standard FPGAs.

ProASIC3 devices have several levels of security protection. When the new part is shipped to the customer there is no security activated. Even at that level the security protection is high. This is because *"Even without any security measures (such as FlashLock with AES), it is not possible to read back the programming data from a programmed device"* [2]. The manufacturer claims that the readback function for the FPGA Array is not physically implemented thus making these devices inherently secure: *"Low power flash devices do not support read-back of FPGA core programmed data; however, the FlashROM contents can selectively be read back (or disabled) via the JTAG port based on the security settings established by the Microsemi Designer software"* [2]. Higher security level offers activation of a special user key that protects rewriting of any security settings: *"Designers have the ability to use a FlashLock Pass Key to prohibit any write or verification operations on the device"* [4].

For remote updates of the device both the configuration bitstream and internal Flash memory can be encrypted with the AES device master key. This function can only be used to decrypt the data being sent to the chip for writing and verification. There is no way to pass the internal data back to the outside world even in an encrypted state.

The highest level of protection turns the device into no-longer-programmable chip and should be considered with caution as in case of any bug found, the chip will have to be physically replaced: *"The purpose of the permanent lock feature is to provide the benefits of the highest level of security to IGLOO and ProASIC3 devices. If selected, the permanent FlashLock feature will create a permanent barrier, preventing any access to the contents of the device. This is achieved by permanently disabling Write and Verify access to the array, and Write and Read access to the FlashROM. After permanently locking the device, it has been effectively rendered one-time-programmable"* [2].

The backdoor we found allows readback access to the configuration data. There are some other hidden JTAG functions which give low-level control over the internal shadow memories and allow modification of hidden registers. However, we do not consider them a backdoor because they are not directly associated with undocumented access to the protected data. The simplified outlook of the ProASIC3 security is presented in Fig. 1b.

We evaluated all levels of protection in ProASIC3 devices and were able to circumvent the security at each level. Table 1 summarises the security protection levels in the ProASIC3 devices according to our research findings. The Passkey offers the highest level of security for reprogrammable chip, while the Permanent Lock should be used as the last resort and will turn the device into a one-time programmable (OTP) chip. 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 as this does not fall within the scope of this paper, we have not gone into further details.

Although with the backdoor we found it is possible to extract the IP blocks – Array configuration from the FPGA, there are other ways to extract the configuration bitstream IP. One was published in 2010 and uses special type of optical fault injection attacks called 'bumping attacks' [23]. Another method uses the vulnerability of the AES implementation and in particular the message authenti-

Table 1. Security protection levels in ProASIC3

Secure area	Read Access	Verify Access	Write Access	Secure Lock	AES Encryption	Expected Security	Attack Time
FROM (Flash)	Yes	Yes	Yes	Yes	Yes	Medium	Seconds
FPGA Array	No	Yes	Yes	Yes	Yes	High	Days
AES Key	No	Yes	Yes	Yes	No	Medium	Seconds
FlashLock Passkey	No	Yes	Yes	Yes	No	Very High	Hours
Backdoor Key	No	Yes	Yes	Yes	No	Very High	Hours
Permanent Lock	No	No	Yes	No	No	Ultra High	Minutes

cation code (MAC) used to protect the integrity of the encrypted bitstream [24]. By compromising the AES key in ProASIC3 the IP could be extracted even without access to the encrypted bitstream. Although there is no readback access for AES-encrypted bitstream configuration, verification is allowed and can be brute forced given small number of unknown bits. An attacker can pass authentication, then write a mask configuration file containing all zeros but a small number of ones per each row, e.g. 16 bits in a 832-bit row. Writing '1' over '0' into the Flash configuration memory has no effect, while writing '0' over '1' changes its state. Since each row can be verified independently in 2 ms time, he can brute force unknown bits row by row. With 50 samples we successfully extracted full IP from A3P250 in 1 week. There is a MAC security feature to prevent arbitrary writing in AES mode through validation of data. We broke it by figuring out that it uses feedback-shift register with just 4 bits of uncertainty per AES CBC (cipher-block chaining) block and easily bruteforceable off-line. Moreover, we managed to disable the MAC verification by modifying few lines in the controlling STAPL file, thus making any arbitrary writing seamless.

3 Experimental Method

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 the ProASIC3, however, using Actel development software we were able to generate a series of STAPL files which we analysed to determine 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 a master JTAG interface and simple functions controlled by PC software via an RS-232 interface for convenience (Fig. 2a). The ProASIC3 chip was placed into a ZIF socket for easier handling. During this 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 the DR registers. We then used a classic DPA setup to analyse the side-channel emission from the ProASIC3 devices during decryption and to

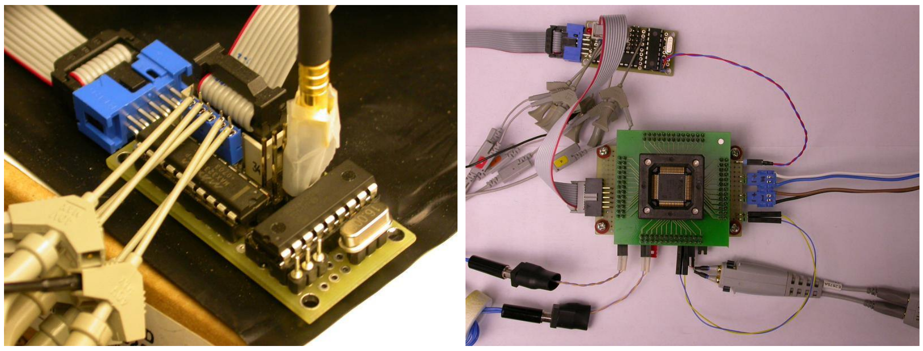

Fig. 2. Test setup: (a)control board, (b)DPA analysis

access operations as well as other undocumented commands. We constructed a simple prototype board with a ZIF socket for the A3P250 device (Fig. 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 the 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.

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. The unknown commands were then 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 can hardly help in understanding 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 [24]) to achieve an improved 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 the ProASIC3 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 below −20 dB. The prototype sensor setup we used is presented in Fig. 3a.

The system consists of a control interface that can be represented by a personal computer, a remote control with embedded processor or other human interface (Fig. 3b). The test algorithm is either present inside the test generator or it is supplied via the control interface. Each device under test (DUT) requires its own test algorithm, which is a part of a standard device operation and consists of a list of commands to run the DUT in the way required by the tester; for

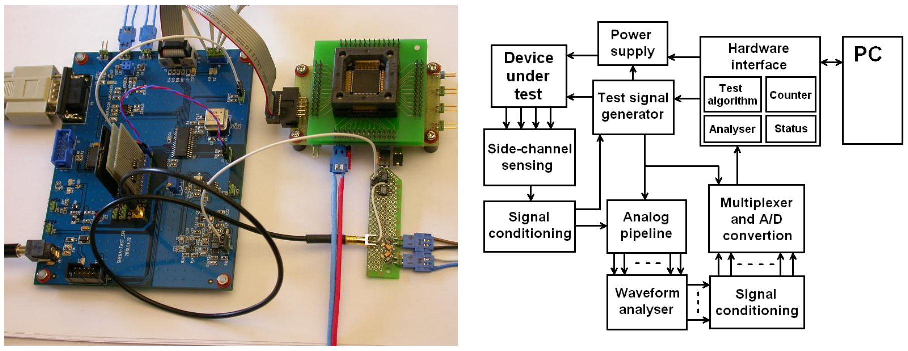

Fig. 3. (a)Prototype board with our sensor, (b)Block diagram of PEA setup

example, to establish an authentication or to decrypt the data. The test signal generator produces sets of test patterns according to the programmed algorithm. One part of the algorithm is fixed while the other is changing. The power supply of the DUT is provided by programmable power supply. The clock of the AC source can be synchronised to the external clock provided by the signal generator, which in turn can be synchronised to the device's internal clock. This could be done by injecting the clock signal from the generator into the DUT power supply line. That allows a significant improvement over the existing measurement equipment setup by significantly reducing the jitter influence on the measurement results.

As the DUT performs a requested operation it leaks some information via side channels as a side effect of the device operation. Those side-channel responses are measured with dedicated sensors specific for each type of side-channel emission, in our case the resistor in the core supply line. The sensors output the signals in analog form which are then put through a signal conditioning circuit to amplify the signal and reduce the noise by applying various filters. The signal is then delayed by one clock period, determined by the test signal generator. The purpose of the delay is to be able to compare the device side-channel response to different input test data. The pipeline delivers its delayed output to a waveform analyser which compares the new signal with the delayed signal for the determined number of points and provides an output, which is the difference between them. The signal from the analyser is conditioned using amplifiers and filters to meet the requirement of the acquisition system, which then converts it in a multiplexed way into a digital form. The output from the multiplexer is then transferred to the hardware interface. The response analyser makes the decision on the reply based on the predetermined decision making patterns and updates the status register, which is checked by the control system. Our invention of this new analysis technique is covered by a patent which is available to the public [25]. Our improvement comes from: real-time attack with no latency associated with an oscilloscope hardware/software, network and memory; lower noise with better probe design, analog signal processing and efficient filtering.

In the end we used a silicon scanning technique based on PEA combined with a classic DPA setup (resistor in power line, differential probe, oscilloscope, PC with MatLab). Communication was analysed with DPA, while the functions of control registers were tested with both DPA and PEA. This is because for some operations DPA is not sensitive enough.

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. Fig. 4a shows some of these registers learned from the STAPL file analysis as well as newly discovered ones. Not only that, but some registers were impossible to update with a new data (Fig. 4b). Most of these registers were representing a Read-Only memory, referred to as FROW in the STAPL file. 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 ProASIC3 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 ProASIC3 chips.

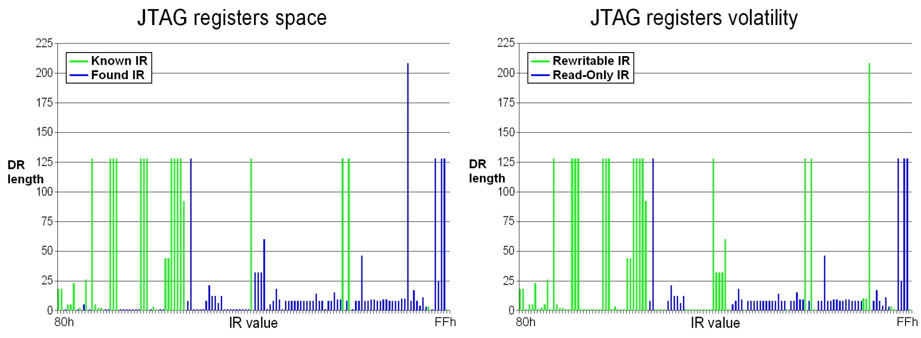

Fig. 4. JTAG scanning results: (a)hidden DR registers, (b)non-volatile DR registers

Although they do not have any specialised DPA countermeasures, the ProASIC3 devices are at least 100 times longer 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 ProASIC3 chips quite a challenging task. Fig. 5a shows the result obtained by comparing single traces for different input data. Averaging over 4,096 traces gives a low-noise result, but takes a couple of minutes to acquire (Fig. 5b). As can be seen, for single traces the noise overshadows any useful signal with SNR being below -20 dB. The FFT spectrum of the power trace does not have any characteristic peaks (Fig. 6a) and filtering will not be very effective for substantially improving DPA results.

The noise can be reduced by using a frequency locking technique. There are publications on the successful use of these techniques on FPGAs [26]. That way the timing jitter between traces can be reduced to approximately ten degrees of the phase shift at 19.7 MHz. 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.

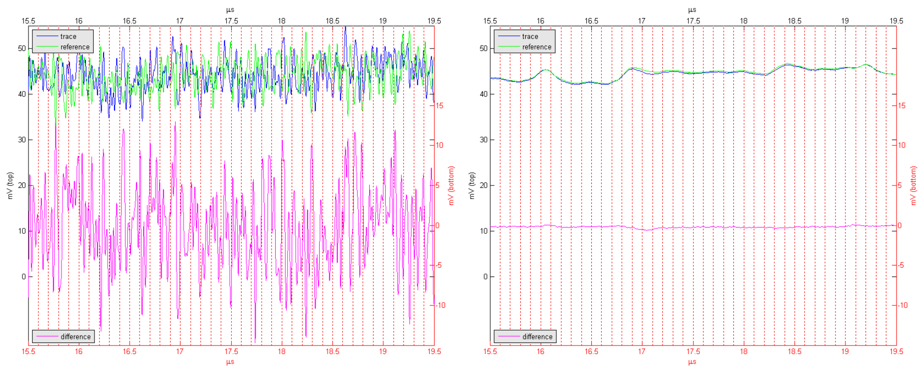

Fig. 5. Power analysis results: (a)single trace difference, (b)average of 4,096 traces

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 an unprotected implementation of AES encryption it would require at least 256 traces to be averaged to reduce the noise and get a reliable correlation with key bits (Fig. 5). The PEA approach allowed the key bits to be guessed at 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 electronic 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 guesses (Fig. 6b). 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.

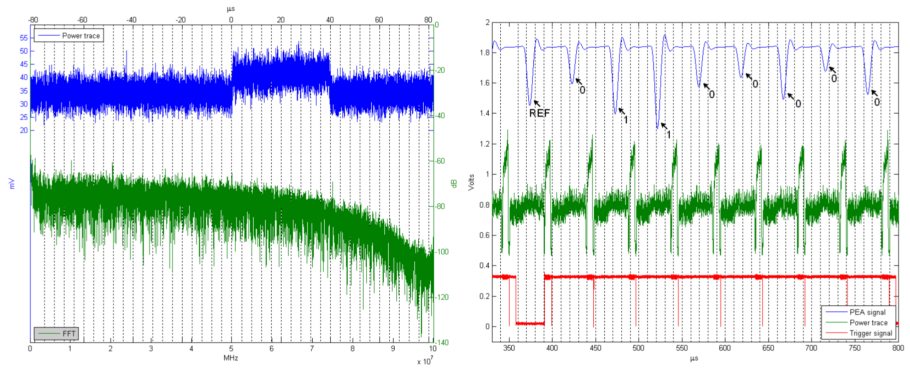

Fig. 6. Power analysis of ProASIC3: (a)FFT spectrum, (b)PEA scan for AES key

Initially we analysed all the active JTAG commands using power analysis. Fig. 7a shows how AES authentication and Passkey verification traces look, while Fig. 7b shows traces of Array verification and Flash FROM reading commands. 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 considerably reduced compared to AES operation. 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. However, due to much more robust DPA countermeasures it took us approximately one day to achieve this using simple PEA hardware. For the classic DPA setup, in order to achieve at least 0.1 mV difference detectable by an oscilloscope, at least 32 consecutive bits of the key must match. Given the input noise of probe+oscilloscope of 1mV at least 64 synchronous or 1024 asynchronous averages must be performed. It takes about 15 seconds to average the signal on MSO8104A to get a positive SNR. Finding all the unknown bits of the key with DPA would take 2^{32} times longer or approximately 2,000 years. Further investigation of the backdoor key operation revealed that it unlocks many of the undocumented functions, including reprogramming of secure memory areas and IP access.

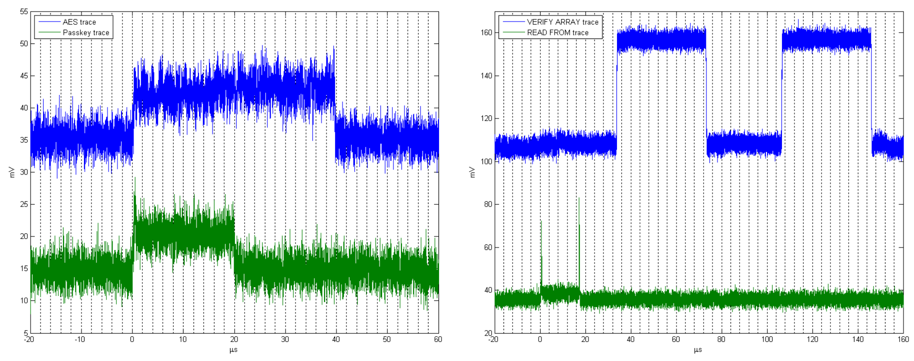

Fig. 7. Examples of JTAG power traces: (a)AES vs Passkey, (b)Array vs FROM

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 ProASIC3 and in their other latest generation Flash FPGAs [4]. 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 with the increase in capability planned for the next generation of our technology, defeating more secure DPA countermeasures is a very real and achievable expectation. 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 the 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 [27]. 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. The backdoor feature was designed as a part of the JTAG security protection mechanism and traces can be found in the Actel's Libero FPGA design software. Anyone with this free software installed on their Microsoft Windows machine can go to the Search option in the Start menu and search for one of the fuse names taken from Actel generated STAPL file. For example, search for the word ULUWE in all files. This will return all STAPL files together with templates and algorithm description files. Inside some of those files there is a proof of the designed backdoor feature.

At the same time, other hidden features could be used to assist even more deeper recovery of the information from the chip. For example, if someone over-produced pre-programmed ProASIC3 chips for some important design and then decided on using another product; they can erase the chips and sell them on the grey market. No one would expect any security flaw because the chips are erased and no longer hold any useful data. This is not quite true and in fact there are still some tiny traces of the information left deep inside the Flash cells [28]. With the help of some hidden registers we found during the JTAG scan we were able to adjust the reference voltage of the read-sense amplifiers used by the backdoor and successfully extracted the bitstream configuration from the erased ProASIC3 chip. If you use ProASIC3 in a sensitive military system that had a destructive wipe as a security feature in case of the systems capture, erasing the chip would not be sufficient to prevent the readback of the user IP. This was a good example of combining the backdoor with hidden and undocumented JTAG debug functions.

6 Conclusion

Our experiments had achieved the affordable time for scanning of two weeks. As a result we were able to locate and exploit an undocumented backdoor in the Actel/Microsemi ProASIC3 chip positioned as the industry's highest security device. To our knowledge this is the first documented case of a backdoor inserted in a real world device with critical applications. Not only can a poorly protected AES key be extracted from the ProASIC3 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 ProASIC3 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 ProASIC3 security system. We analysed other Actel/Microsemi products and found they all have the same backdoor. Those products include, but are not limited to: Igloo, Fusion and SmartFusion. The ProASIC3 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 ProASIC3 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.

The chip manufacturer suggests the possibility of performing remote upgrade of the firmware in ProASIC3 devices via TCP/IP: *"The system containing the low power flash device can be assigned an IP address when deployed in the field. When the device requires an update (core or FlashROM), the programming instructions along with the new programming data (AES-encrypted cipher text) can be sent over the Internet to the target system via the TCP/IP protocol. Once the MCU receives the instruction and data, it can proceed with the FPGA update. Low power flash devices support Message Authentication Code (MAC), which can be used to validate data for the target device"* [2]. However, as we have already mentioned, the implementation of the MAC is very insecure and trivial to break. That only increases concerns about the remote updates and taking over the control of systems based on the ProASIC3 and other Actel/Microsemi Flash FPGAs.

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 ProASIC3, where the AES key can be extracted in less than a second's time [24] compared to hours required for Passkey extraction.

Most of the current DPA-based research into silicon chips is centred on looking for Trojans. There has been little research conducted into comparing legitimate chips with counterfeits, using DPA. This is primarily because standard DPA equipment generates many terabytes of data which requires complex analysis, resulting in a long lead time and a high cost. As PEA offers superior sensitivity and high speed analysis, it would be possible to check all chips to be used in assembly, not just a sample batch. Currently this is not possible with any other comparison methods whether by DPA or reverse engineering.

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. Until the development of the efficient silicon scanning techniques, 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. Military ProASIC3/EL FPGA Fabric User's Guide. Microsemi, 2011. `http://www.actel.com/documents/Mil_PA3_EL_UG.pdf`
3. Design Security in Nonvolatile Flash and Antifuse FPGAs, Security Backgrounder. `http://www.actel.com/documents/DesignSecurity_WP.pdf`
4. Actel ProASIC3/E Production FPGAs, Features and Advantages, 2007. `http://www.actel.com/documents/PA3_E_Tech_WP.pdf`
5. The Free Dictionary. Backdoor. `http://www.thefreedictionary.com/backdoor`
6. 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
7. 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
8. 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
9. 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, 10-14 March 2008, pp 1362–1365
10. 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
11. D. Agrawal, S. Baktir, and D. Karakoyunlu, P. Rohatgi, and B. Sunar: Trojan Detection using IC Fingerprinting. IEEE Symp. on Security and Privacy (SP), 2007, pp 296–310
12. Y. Jin and Y. Makris: Hardware Trojan Detection using Path Delay Fingerprint. IEEE Int. Workshop on Hardware-Oriented Security and Trust (HOST), 2008
13. 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
14. 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, 2009, pp 1735–1744
15. P. Kocher, J. Jaffe, B. Jun: Differential Power Analysis. CRYPTO'99, LNCS, Vol. 1666, Springer-Verlag, 1999, pp 388–397
16. Military ProASIC3/EL Low Power Flash FPGAs Datasheet. Microsemi, 2012. `http://www.actel.com/documents/Mil_PA3_EL_DS.pdf`
17. Introduction to Hardware Security and Trust, Eds: Mohammad Tehranipoor and Cliff Wang, Springer, September 2011
18. JTAG Boundary scan. IEEE Std 1149.1-2001
19. JTAG Programming specification. IEEE 1532-2002
20. Jean DaRolt, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre: New security threats against chips containing scan chain structures. HOST 2011, pp 110-115
21. Actel, ISP and STAPL, Application Note AC171. `http://www.actel.com/documents/ISP_STAPL_AN.pdf`
22. ProASIC3 Frequently Asked Questions, Actel Corporation, Mountain View, CA 94043-4655 USA. `http://www.actel.com/documents/pa3_faq.html`
23. S. Skorobogatov: Flash memory 'bumping' attacks. Cryptographic Hardware and Embedded Systems Workshop (CHES 2010), August 2010, LNCS, Vol. 6225, Springer, pp 158–172

24. S. Skorobogatov, C. Woods: In the blink of an eye: There goes your AES key. IACR Cryptology ePrint Archive, Report 2012/296, 2012.
http://eprint.iacr.org/2012/296
25. Integrated Circuit Investigation Method and Apparatus. Patent number WO2012/046029 A1
26. S. Skorobogatov: Synchronization method for SCA and fault attacks. Journal of Cryptographic Engineering (JCEN), Vol. 1, No. 1, Springer, 2011, pp 71–77
27. Intrinsic ID, Quiddikey on ProASIC3 FPGAs
http://www.intrinsic-id.com/quiddikey_on_Actel_FPGA.htm
28. S. Skorobogatov: Data Remanence in Flash Memory Devices. Cryptographic Hardware and Embedded Systems Workshop (CHES 2005), August 2005, LNCS, Vol. 3659, Springer-Verlag, pp 339–353

Appendix

A STAPL example code

```
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;
```