Evilsploit – A Universal Hardware Hacking Toolkit

By,

Chui Yew Leong, Wan Ming Ming

Introduction

In general, hardware hacking is about to understand the inner working mechanism of hardware and defeat its security protection in order to harness as many as the resources of hardware together with its collaborative software. In most of the time, the hardware hacking process is started from reversing process. From hardware point of view, reversing in static way includes uncover the schematic and disassemble the binary. On the other hand, reversing in dynamic way includes to find a way to debug the hardware and to fuzz it accordingly.

In practical, it is almost a standard operating procedure to obtain the binary of the hardware and reverse it subsequently. As a supplementary technique for static binary reversing, debugging allows the real hardware operation process being demystified in run time. In fact, the binary itself is possible to be obtained by using debugging technique when it is not available or provided by manufacturer. So, it is crucial to figure out the provisioning ports of the hardware in order to start performing hardware hacking.

The conventional approach to identify provisioning ports is by using pin finder toolkit such as Jtagulator. However, it is impractical and inefficient once provisioning port being found, another toolkit such as Shikra has to be used to manipulate the provisioning port. It is not only prone to error, but not hacker-friendly. So, it is important to find a way to bridge the gap between provisioning port identification and manipulation processes. With this, it allows the hardware hacking process to be automated by making it scriptable in high level.

The authors propose a new method to allow provisioning port identification and manipulation in a single toolkit by using connection matrix. With this, it is possible to construct arbitrary analog-alike connection in array form to implement all pattern of interconnect between bus interfacing chip and the target. Hence, once the appropriate provisioning port being figured out, on the spot, it is ready to be used for debugging or firmware dumping purposes. Besides, it is also an ideal assistive toolkit for unknown signal analysis, side channel analysis (SCA), and fault injection (FI).

Problem Statements

In order to figure out provisioning ports, the current approach is by using pin finder hardware. Once the pins getting identified, another bus interfacing hardware has to be connected to the right pins to manipulate the target bus. The manual pins re-connection process is an

obstacle to prevent the automation of full hardware hacking process, from low level to high level, from electronics to static analysis.

On the other hand, there are some implementations are attempting to build a universal hardware to combine the function of bus identification and manipulation. However, those implementations are not an ideal solution for this purpose due to the limitation of high complexity, huge board size, limited flexibility to probe signal, and stability issues of impedance variation.

Besides, it is not a good idea to fully rely the use of microcontroller to generate probing signal in identifying pins of provisioning port. Due to the unknown nature of the target, it is not a simple issue to ensure the signal integrity of the probing signal in a broad range of physical conditions.

Literature Review

In general, there are 2 kinds of implementation method being found so far for this purpose. First, is by using array of analog multiplexers. With this, a huge number of analog multiplexers in highly complex combination has to be adopted to implement a limited permutation and pattern of interconnect. It is almost impossible to build an arbitrary interconnect by using this method. In this case, if the target provisioning port is as simple as UART or its variation, it shouldn't have any issue to detect Tx pin, following by Rx pin. However, if the target provisioning port is getting more and more complicated as what JTAG does, it is extremely hard to fully cover all the signal routing pattern of TCK, TMS, TDO, and TDI for mutual interconnects. A typical example of this implementation method is TOAD [1].

Second, is by using the combination of FPGA and level translator. With this, it is possible to construct an arbitrary interconnect in digital approach. However, with digital approach, it is not friendly to interface with an unknown voltage range of signal. To overcome this, extra circuitry such as ADC has to be added but it will increase the complexity of the board. Besides, there are still having some potential stability issues when level translator being used to connect with target pin. Those impedance related problems are detrimental to the proper function of the hardware. A typical example of this implementation method is Hardsploit [2].

Regarding the argument of bit banging with microcontroller in generating probing signal to identify pins of provisioning port, it is rather a controversial topic. The full process of JTAG finding is hardly to be implemented in a resource constrained microcontroller. In addition, the physical characteristic of GPIO in driving signal can prone to consequences of signal integrity. Frequency and impedance are among the 2 main issues of such kind of implementation. While being used in fully customized embedded design for development purposes, microcontroller is undoubtedly the best choice to achieve the mission. However, for hacking purposes, a huge variety of unknown targets have to be tolerated in working condition. Hence, it is not as simple as bit banging can achieve. There are a broad range of pin finder hardware can be found in the market.

Methodology

A new approach by applying analog-alike connection matrix to implement perfect arbitrary interconnect is proposed in this paper. With connection matrix, it is possible to construct whatever pattern of signal routing in runtime, control-able from high level, in order to implement a true universal hardware hacking toolkit, which is named as Evilsploit. Since the connection matrix is analog-alike, it has no restriction to the signal direction, the connection from input to output is exactly the same as from output to input. With such a distinct characteristic, the processes of bus identification and manipulation can be combined and implemented in a single piece of hardware toolkit. The universal hardware toolkit allows high level logical control to the signal routing pattern of connection matrix, which enable hardware hacking process to be fully automated. In other words, by connecting cables to those suspicious pins of provisioning port, the toolkit will find out the exact pins and route the connection accordingly, and finally get ready to be controlled from a computer.

The connection matrix can be implemented by using AD75019 [1] from Analog Device. The AD75019 is a general-purpose version of analog switch array, as compared to its audio/video version, AD8113 [2], which is widely used in audio visual industry. According to the specification, the AD75019 contains 256 analog switches in a 16 x 16 array, and any of the X or Y pins may serve as an input or output. Figure 1 shows a routing pattern of an 8 x 8 matrix which connecting input 2 and input 5 to output 3 and output 5, respectively. However, it is also valid to use output 3 and output 5 as input and make input 2 and input 5 as output, it makes no difference at all.

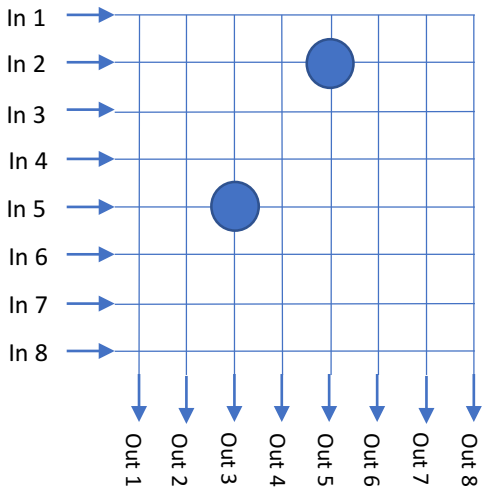

Figure 1: A Routing Pattern of an 8 x 8 matrix

In the section of problem statement, we mentioned it is complicated to implement a full connection matrix to support arbitrary routing pattern in runtime, so, this is the first reason. Multiple AD75019 can be cascaded to extend the number of switches in the array. As a result, by cascading two AD75019, it can build a 32 x 32 array. Figure 2 shows eight AD75019 are cascaded to implement a 128 x 128 array.

Figure 2: Eight AD75019 are cascaded to implement a 128 x 128 array

The analog switch array is control-able via a SPI-alike interface. Control data can be shifted into the internal shift register to deploy new routing pattern or disconnect/reconnect a particular connection in runtime. Figure 3 shows the connection from input 5 to output 3 can be disconnected on the fly once a routing pattern as shown in Figure 1 is deployed to the matrix.

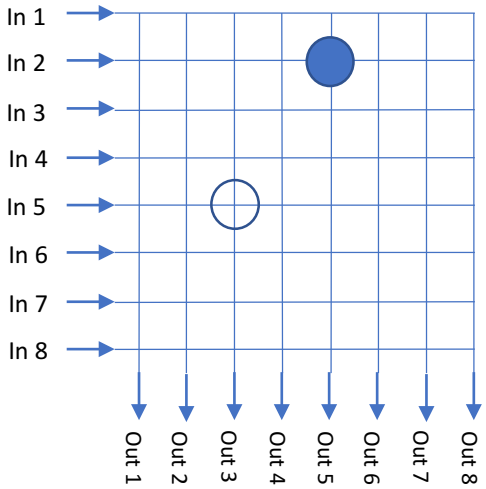

Figure 3: Connection from Input 5 to Output 3 is Disconnected

A microcontroller is added to provide control to the analog switch array from high level interface. A series of command sets are defined to allow a computer to send command in deploying new routing pattern or disconnect a specific connection. In higher abstraction layer, macros are defined in generating probe signal together with sets of routing patterns to brute-force the pins combination in a smart way. The bit banging technique being applied in microcontroller only manage to generate the simplest probing signal to instruct the target to respond immediately. From the perspective of microcontroller, it will not concern the actual data of response from the target. Instead, it only concern is there any response from the target or not. In other words, it will try to complete its job in as few rising and falling edges as

possible. Once a set of routing pattern is realized to be responsive from the target, a new routing pattern will deploy to forward the pins connection to a bus interfacing chip. In Evilsploit, a FT4232H is being used to support multiple bus communication methods in a single chip. From now on, the FT4232H will take over the microcontroller to make control to the target. So, for those debugging tools such as UrJTAG, OpenOCD, or PuTTY can drive the FT4232H as usual. The issues of signal integrity are fully managed by FT4232H with a special interfacing engine named as MPSSE. Hence, the signal degradation problems due to the impact of physical characteristics from signal driver, cable, or grounding can be minimized. Figure 4 shows the system block diagram of Evilsploit.

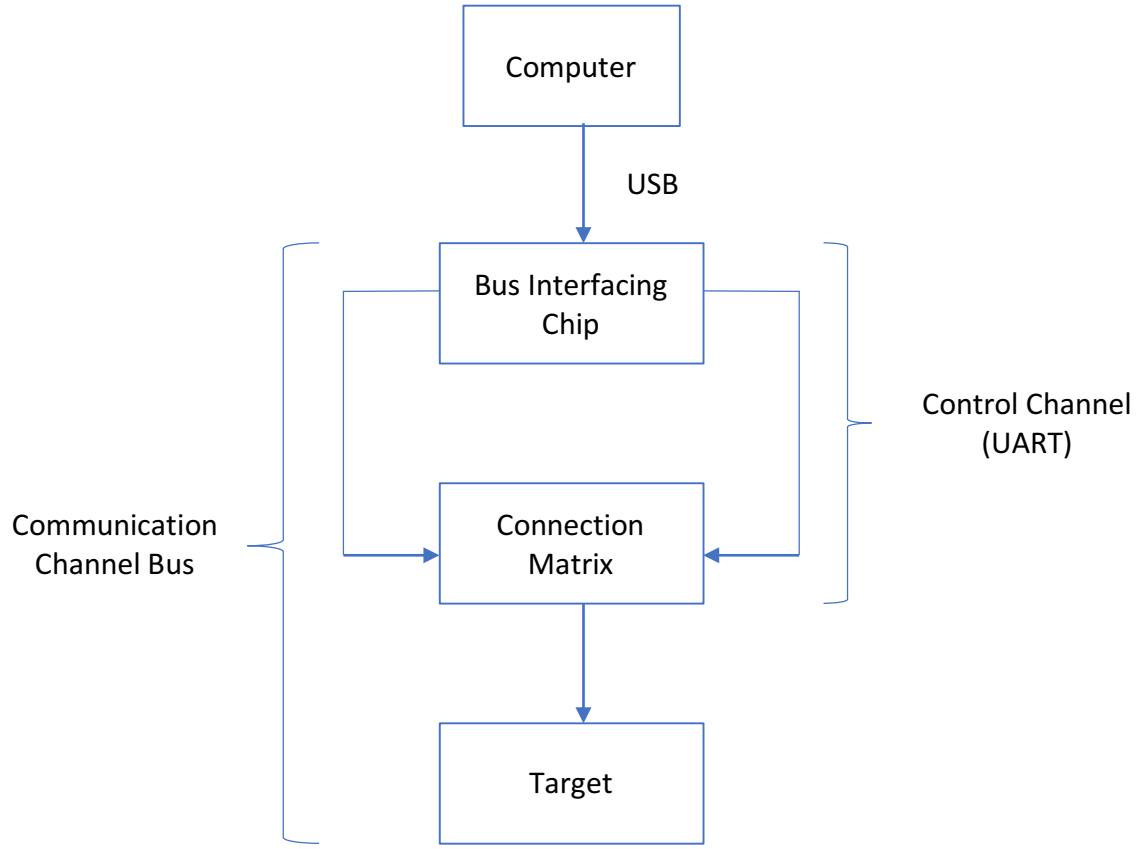

Figure 4: System Block Diagram of Evilsploit

It is crucial to note that the definition of connection matrix comprises analog switch array and the control interface provided by microcontroller. The connection matrix is sitting behind the bus interfacing chip, FT4232H, to communication with computer. So, the computer will only interact with FT4232H to complete the whole hardware hacking process. It provides maximum level of compatibility to the operation of existing debugging tools.

In some special cases, the pins identification process might be affected by connecting to the Vcc or Gnd accidentally. In some even special cases, some pins might apply End of Line (EOL) technique which supervise the continuity of impedance by using a voltage divider. This

technique is widely used in building subsystems such as fire alarm, intruder alarm, building automation or early evacuation public address. For any form of probe being attached to the pin, it will stop functioning or malfunctioning immediately. The theory behind the magic is measurement probe or device has input impedance that will apparently change the voltage in the divider circuit, provided the resister value is closed enough to the input impedance. So, with such a protection approach, it is possible to stop certain level of hardware hacking from the early stage. However, with Evilsploit, it is possible to perform pins classification first, right before the pins identification. By stacking advanced function module in piggyback form on top of the basic module, Evilsploit can be transformed as a hardware hacking machine with built-in circuit instrumentation features. Besides, special circuitry designed by using op-amp is implemented to bypass the EOL protection. The idea of op-amp is because it has infinity input impedance which is ideal to prevent voltage drop in the divider circuit. As a result, the EOL based detection circuitry will not able to detect any attempt of pins probing. It is important to note that the op-amp based bypassing technique is only valid to the output pins from the target. Figure 5 shows an over simplified design block diagram to bypass EOL based pin protection.

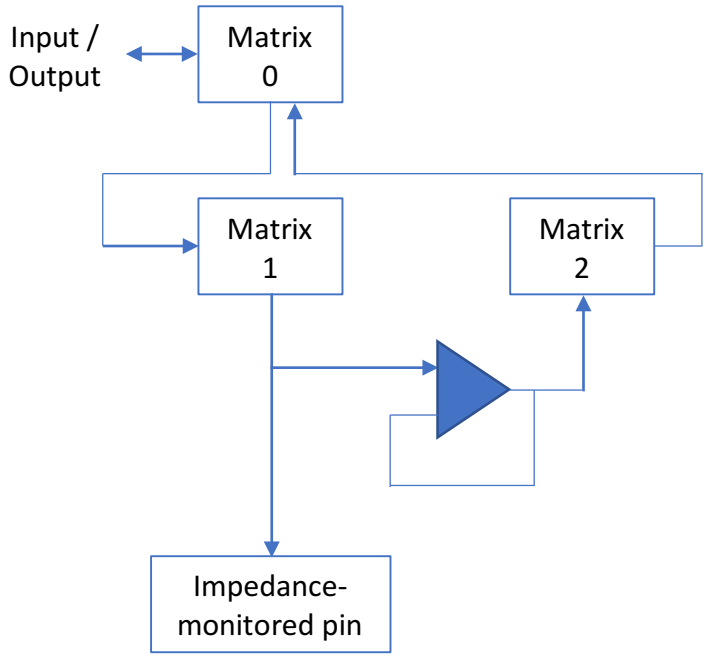

Figure 5: An Over Simplified Design Block Diagram to Bypass EOL Based Pin Protection

Sometimes, digital interconnect being implemented by FPGA would have high similarity to the connection matrix, or more precisely, switch array. However, the analog-alike nature of switch array has shown its advantages over the digital based interconnect. The massive adoption of level translator for voltage compatibility purpose is a potential tradeoff to signal integrity. Besides, it is rather hard to implement mutually transparent bi-directional connection with FPGA. Hence, for such a niche purpose to implement a connection enumerator with high tolerance to variety of electrical characteristics, the analog-alike

connection matrix would have overwhelming advantages over the digital interconnect. In addition, for balanced or differential signal, connection matrix is the only qualified candidate in the field. At the front-end, an ADC is used to detect incoming voltage level in order to switch level translator into a correct setting which is compatible to the bus interfacing chip. So, for each routing path from bus interfacing chip to level translator can be conditionally assumed as fully analog based, in other words, it is bi-directional, support positive and negative voltage signaling, and better signal integrity. With connection matrix, the physical dimension of Evilsploit board can be minimized without affecting the high complexity of the features it can support in good reliability and stability.

In high level application, Evilsploit can be used as a simplified version of hardware hacking machine as what the author published in Phrack #63 by year 2005 [3]. The main objective of hardware hacking machine is to extract data from arbitrary signal especially the asynchronous form in runtime. For asynchronous signal, it is crucial to implement clock data recovery (CDR) unit for synchronous clock re-generation. The quality of synchronous clock will determine the accuracy of data going to be extracted. By integrating Evilsploit with a logic analyzer, it is possible to implement a simplified and low-cost version of hardware hacking machine. The intervention of logic analyzer is to implement a trigger-based oversampled CDR with certain amount of buffer, as opposed to high complexity design with FPGA or CPLD. In fact, for majority of arbitrary signal analysis purpose, it is sufficient to use logic analyzer as a wide-range receiver. Unless signal injection is necessary in the scope of hardware hacking, otherwise, it is a little bit overkill to use FPGA or CPLD as CDR in recovering field bus communication data. Figure 6 shows the system block diagram of simplified hacking machine implementation.

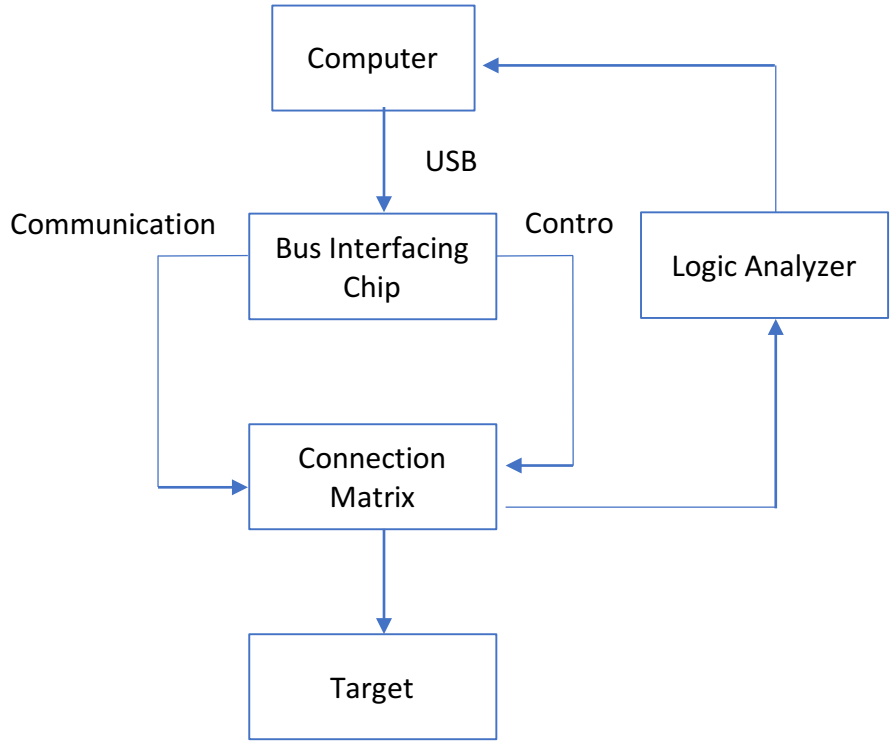

Figure 6: System Block Diagram of Simplified Hacking Machine Implementation

For advanced application, Evilsploit can be used as an assistive tool for Side Channel Analysis (SCA) purposes. Due to the limited buffer space, it is sometimes impossible to sample the correct instant of crypto process. However, it is possible to identify the occurrence of some special pattern before the crypto process. By taking the special pattern as the trigger instant of sampling, it would have high chance to sample the whole crypto process in good condition. We will present an example to recover the secret key of a SM4 crypto by using the combo of Evilsploit and ChipWhisperer. Besides, it is also possible to be used as an assistive tool for Fault Injection (FI) by generating fault signal at the right instant to bypass some conditional protection. Figure 7 and Figure 8 show the system block diagrams of using Evilsploit as assistive tool for SCA and FI purposes, respectively. Finally, the physical view of Evilsploit board is shown in Figure 9.

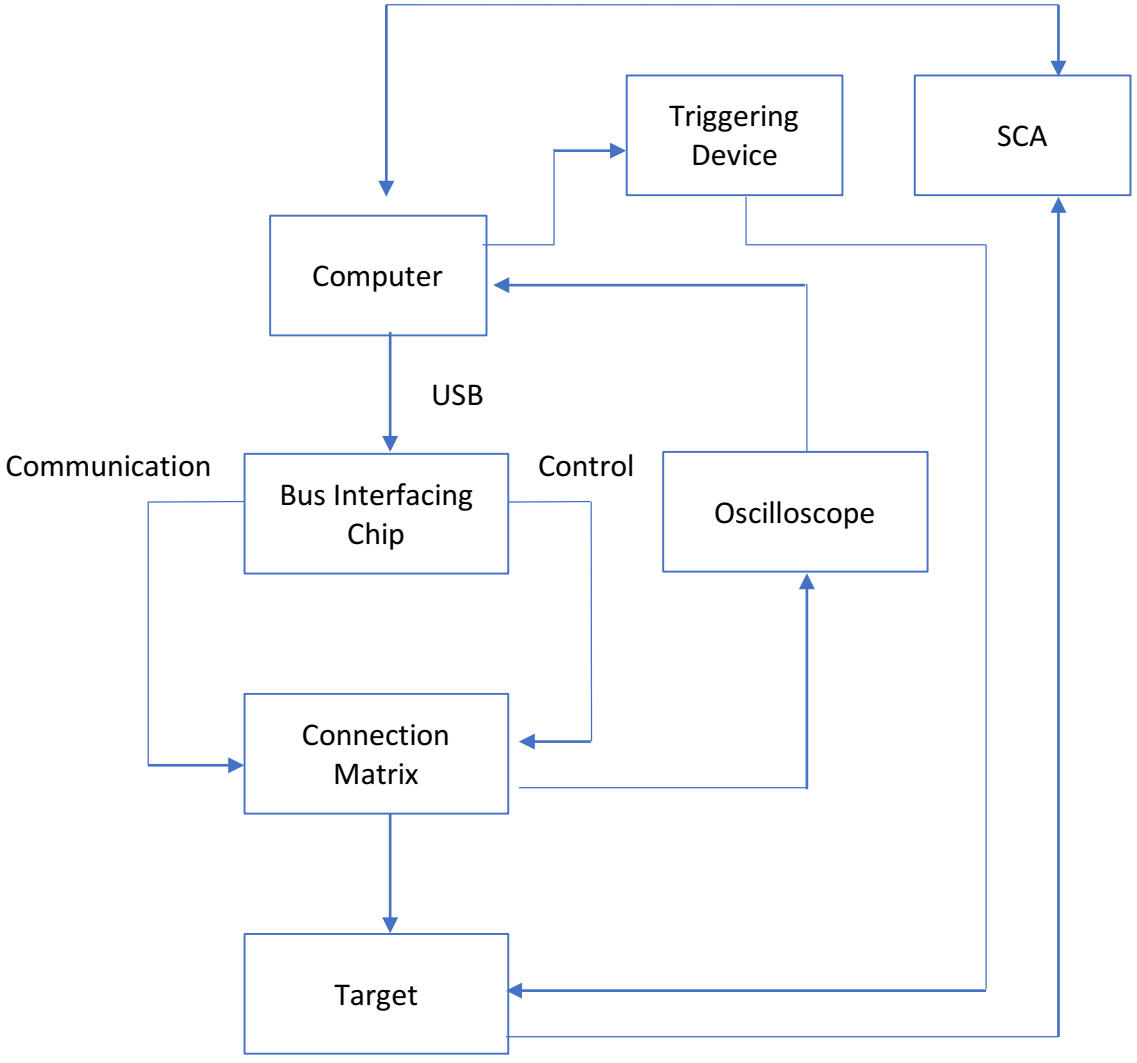

Figure 7: System Block Diagram of Using Evilsploit as Assistive Tool for SCA

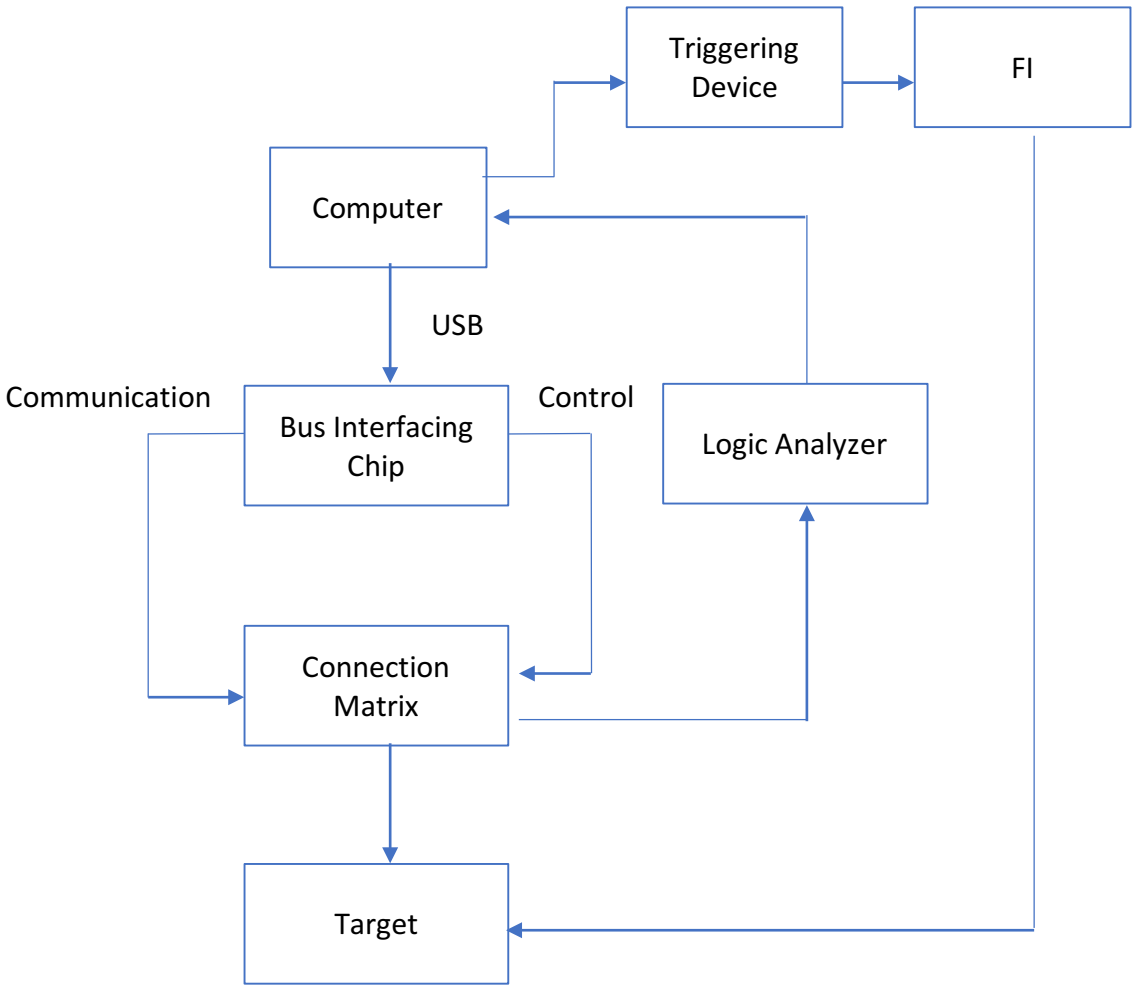

Figure 8: System Block Diagram of Using Evilsploit as Assistive Tool for FI

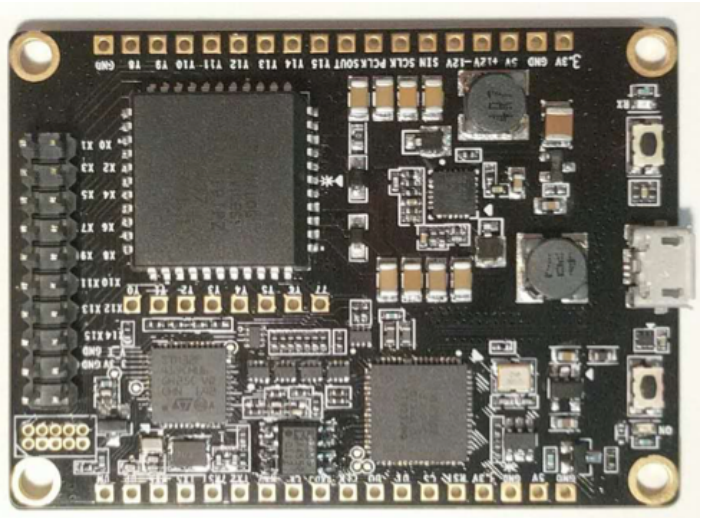

Figure 9: Physical view of Evilsploit Board

Results

In order to identify 4 JTAG pins (TMS, TDI, TDO, TCK) from a total of 16 unknown pins, Evilsploit will take about 100 seconds to complete the whole process. Once the process is completed, the connection matrix will automatically forward the signal routing pattern to the bus interfacing chip. Hence, it is ready to be controlled or manipulated by any well-known JTAG debugging tools such as UrJtag or OpenOCD. For UART, it takes around 90 seconds to identify Tx and Rx pins from a total of 16 unknown pins. Again, it is ready to be controlled by any well-known UART tools such Minicom or PuTTY.

As an assistive tool for SCA attack, Evilsploit has been used together with ChipWhisperer to recover the secret key of 3DES and SM4 crypto, which are implemented in embedded form factor. In both of the cases, Evilsploit has generated trigger signals for ChipWhisperer to start sampling process at the correct timing. The collected samples are going to be processed by a highly optimized Matlab script to recover the secret key of the crypto, individually. Both the secret keys of 3DES and SM4 crypto are successfully recovered.
Regarding the issue of bypassing EOL based detection circuitry, Evilsploit has successfully convinced the detection circuitry that nothing is attached on the controlled pins. Thus, the signal level manipulation is completely committed in such a protected mechanism.

Conclusion and Future Works

The current version of Evilsploit is not hacker-friendly as assistive tool for either SCA or FI. However, we will try to make it possible to compatible with some well-known SCA/FI tools in near future. Other than that, we are also working hard for a fully or semi-fully automated embedded system vulnerability discovery platform. Since Evilsploit has bridged the physical gap of provisioning port identification and manipulation, it is now the turn of emulated, symbolic, and concrete execution to be interacted each other to precisely analyze the security of embedded system with lesser human intervention.

Takeaways for Blackhat 2017

Try to imagine, with Evilsploit, you can manipulate JTAG without knowing what's the hell of TDI, TDO, TMS, and TCK.

References

[1] http://www.analog.com/media/en/technical-documentation/data-sheets/AD75019.pdf
[2] http://www.analog.com/media/en/technical-documentation/data-sheets/AD8113.pdf
[3] http://phrack.org/issues/63/17.html