EVILSPLOIT - A UNIVERSAL HARDWARE HACKING TOOLKIT

PRESENTED BY TYA

Chui Yew Leong
Wan Ming Ming
WHO ARE WE

• Chui Yew Leong
• System Architect
• Embedded, Hardware
• System Design
• cawan[at]ieee[dot]org

• Wan Ming Ming
• Senior Engineer
• Embedded, Hardware
• usix[at]tya[dot]email
HARDWARE HACKING – AN INTRODUCTION

• To understand the inner working mechanism of a piece of hardware or a group of hardware where connected each other in wired or wireless way

• Reversing is the core
  • Static
    • Hardware connection, interfacing method, and high level communication
    • Binary Disassembly
  • Dynamic
    • Debugging
    • Fuzzing
HARDWARE HACKING – AN INTRODUCTION

• Hardware connection, interfacing, and communication
  • PCB, Multi-meter, Soldering / De-soldering Machine, Magnifier, Oscilloscope, Analyzer

• Binary Disassembly
  • Disassembler

• Debugging
  • JTAG, UART

• Fuzzing
  • Custom Fuzzer
HARDWARE HACKING – AN INTRODUCTION

• You need 3 things
  • Schematic, PCB, Datasheets
  • Firmware
  • Provisioning Ports

• Datasheets and some connection verification are sufficient to understand overall hardware operation
  • PCB manufacturer can reverse the hardware and generate PCB design and schematic files

• Firmware can be obtained in 2 ways
  • Get it from manufacturer
  • Dump it from hardware

• Provisioning ports usually for in-house debugging or dealer servicing, repairing, and support
  • High chance to leak inner working of hardware
  • Can use it to dump firmware without applying chip-off technology

• It is crucial to harness provisioning ports
TODAY’S OBJECTIVES

• To automate the process of finding provisioning ports
• To manipulate the provisioning ports being found directly
• To maximize the automation level of hardware hacking process
PROBLEM STATEMENTS

• You need 2 sets of tools for bus identification (pins finding) and bus manipulation (control and monitoring)
  • Example: You use Jtagulator to identify Jtag and use Shikra to interface with it

• You need to do manual wire connection for 2 times
  • Example: You have to mark down the colors for the first time and reconnect it accordingly
  • How often you connect it wrongly?

• You cannot automate the process of hardware hacking
  • Should blame the gap between bus identification and bus manipulation
A NEW PROPOSAL

• Use a connection matrix to bridge the gap between bus identification and bus manipulation
• Use a bus interfacing chip to enumerate and communicate with the target
• Use a controller to manage the synchronous control between the connection matrix and bus interfacing chip
• Use a computer to automate the process of hardware hacking in high level
  • You can attach a small form factor embedded device to do this job
WHY CONNECTION MATRIX?

- Something seems similar to connection matrix
  - MUX – TOAD, another manufacturer which we cannot disclose its name
  - Voltage translator + FPGA – By Opale
  - Keep doing connect and reconnect manually
- MUX is definitely not a good idea, giant and ugly board, cannot cover all the routing patterns
- Voltage translator + FPGA
  - Need to pre-determine the target voltage and apply voltage setting accordingly
  - You know how messy to deal with such voltage translator, especially to compatible with different forms of buses
  - The full digital way of FPGA based connection matrix is a little bit hard to implement in a small board
- Keep doing connect and reconnect manually
  - Use 2 sets of tools, one for pin finding, and one for bus manipulation
  - Doing all the cable connection manually
  - You know how easy to make mistake, as a human being
WHAT IS CONNECTION MATRIX?

- A device which can provide arbitrary physical connection routing between inputs and outputs
- A well-known technology in large-scale audio distribution system such as public address (PA) system
- The connection is in analog way, not digital way
  - FPAA is something similar, but the voltage range is rather too small
- Can change the state of a particular connection (connected or disconnected) in runtime
- Allow one input being connected to multiple output
- Control via communication bus
WHAT IS CONNECTION MATRIX

• Can be implemented by using AD75019 from ADI
• A general-purpose version of AD8113 which is widely being used in audio visual industry
• Consist of 256 analog switches to implement a 16 x 16 analog array
  • You know how impractical to implement such thing with MUX-alike approach
• All connections are in analog way
  • Any of the X or Y pins may serve as an input or output
• Multiple AD75019 can be cascaded to extend the array scale
  • Yes, it is a little bit ugly
WHAT IS CONNECTION MATRIX
WHAT IS CONNECTION MATRIX

- Let’s study it in matrix form
- Input 2 connected to output 5
- Input 5 connected to output 3
- All the rest are not connected
WHAT IS CONNECTION MATRIX

- Input 2 still connected to output 5
- Input 5 disconnected from output 3, but state remain
- It can be connected again anytime
- You might need it to disable reset pin in runtime
WHAT IS CONNECTION MATRIX

• Both input 3 and input 5 are connected to output 3
• You might need it to ground multiple pins
WHAT IS CONNECTION MATRIX

- Since the connection is in analog way, both of them are no difference, but connected in reverse direction
- You might need it to tap impedance-monitored or EOL protected pins
WHAT IS CONNECTION MATRIX

- It is better to mention the matrix in this way
- It seems simple
- Yes, it is simple for matrix expansion, generally
- But not so simple to cascade multiple of matrixes, particularly
- Let’s see an example
WHAT IS CONNECTION MATRIX

• If the route via Matrix 0 and Matrix 1 detect a suspected impedance-monitored pin, then the connection via Matrix 1 can be disconnected

• Matrix 0 will route to Matrix 2 to take place of Matrix 1 in tapping the impedance-monitored pin

• Since the source-follower provides infinity impedance at the input, then the counter-measure bypassed

• The whole process can be automated

• It is really hard to build this with MUX/DEMUX
WHAT IS CONNECTION MATRIX

- The yellow signal is an input signal
- It is a square wave
- Its voltage level is 1.84 V
- How about its output?
WHAT IS CONNECTION MATRIX?

- The blue signal is an output signal
- It is almost identical to the input signal
- Its voltage level is 1.80 V, very close to the input signal
- What is their frequencies?
WHAT IS CONNECTION MATRIX?

- Their frequencies are 2.06 MHz
- No significant delay found
- No significant jitter found
- Since they are square wave, they comprise a number of odd harmonics (6.18 MHz, 10.3 MHz, ...)
- Since the input square wave maintains its shape at the output, the bandwidth of the connection matrix is even higher
- Let's try with higher frequency
WHAT IS CONNECTION MATRIX
What is Connection Matrix

- The voltage level is slightly reduced from 3.20 V to 3.12 V
- The frequency is 14.7 MHz
- No significant delay or jitter found
- The shape is slightly degraded
- This is in fact the TCK signal of JTAG
- The JTAG bus works normally while connected via connection matrix
WHAT IS CONNECTION MATRIX

• How about 2 MHz and 15 MHz signals being connected to the connection matrix at the same time?
• Yes, the results are exactly the same
• So, connections are nicely isolated each others
WHAT IS CONNECTION MATRIX

- ***This is just a little precaution for beginner
- Output signal is apparently degraded
- While being connected with JTAG, it just not working
- Once detach the probe, it works again
- Why?
WHAT IS CONNECTION MATRIX

- Check the column of input capacitance
- For 1x passive probe, it is 100 pf
- For 10x passive probe, it is 13 pf (P3010)
- For 1x/10x switchable probe, its input capacitance is 110/17 pf
- So, simply switch the probe from 1x to 10x will solve the problem, for whatever probe you use
WHAT IS BUS INTERFACING CHIP?

• Those chips being used in Shikra, Adafruit, Buspirate, and etc
• Playing the role as an agent between a computer and a target with standard communication bus such as UART, SPI, I2C, JTAG, and etc
  • For provisioning purposes, UART and JTAG are the most common choices
  • I2C and SPI are normally for chip to chip communication (ADC, DAC, Flash)
• You no need to create control signal yourself (bit-banging), the chip will generate appropriate control signal with better signal integrity and timing for you
• Normally control by USB interface
WHAT IS MICROCONTROLLER

• We all know what is microcontroller 😊
• We need microcontroller here to create special control signal to control connection matrix (not to bit-bang the target directly)
• The special control signal is SPI-alike, but not SPI
• A set of commands are created to allow control via UART
• And some macro functions
• The computer will determine the routing pattern of the connection matrix via control channel bus
• The connection matrix can be expanded (scalability) or cascaded (complexity)
• The computer will communicate with the target via communication channel bus
• The connection matrix allows arbitrary permutation of routing pattern for the computer to get access to the target
WHY SO SPECIAL? BUS IDENTIFICATION

- Let the bus interfacing chip to do all the signal level task for you
  - Better signal integrity
  - Better accuracy
  - Easier
- Routine of 2 steps task
  - Set a routing pattern of connection matrix
  - Communicate with the target
  - Repeat (you can propose your optimization to minimize the repeat time)
  - Once getting right response, it means your routing pattern is correct
- What next?
  - Taking control from UART – dump firmware, read memory, write memory, analyze bootlog, and etc
  - Analyze the Jtag IDCODE, debug the target with the correct configuration or bsdl file
  - More, more, more, as long as it is scriptable now
WHY SO SPECIAL? UART AND JTAG

- They are the most common provisioning port for embedded system in high level for debugging purposes
  - Have you ever seen anything others? Seldom
- I2C and SPI are mainly for chip to chip communication, in master-slave manner
  - I2C supports multiple master and multiple slave
  - SPI supports single master and multiple slave
  - They let you to read/write flash memory, configure the parameters of chip such as ADC/DAC, communicate each other, and etc
  - They all come to the master
  - Do you really need to work in these signal level? Normally not
- The master hosts all the controls to the slaves (peripherals)
- The master normally run by RTOS or embedded OS
- The master must be debug-able or reflash-able
  - From R&D to production of troubleshooting and QC/QA purposes
  - Dealer level support and servicing purposes
WHY SO SPECIAL? UART AND JTAG

• The master provisioning ports must be simple, not complicated, and easily operated by non-R&D level technical support staffs
• Without special software and hardware tools, UART should be the top choice
• With special software and hardware tools, which is standardized by industry standard, JTAG should be the right choice
• So, in most of the time, UART will offer limited features (depends to the implementation, it is normally powerful enough) for provisioning purposes
• JTAG means in god mode, undoubtedly
• Since producing a marketable embedded system is a chain of processes from R&D to production, and from production to dealers and end users, it is really hard (not technically) to ensure hardware security from the provisioning point of view
• So, the conclusion is UART and JTAG are always be there in most of the embedded systems
• Target both of them
GOING FURTHER - HACKING MACHINE IN YEAR 2005

• Phrack #63 – Hacking with Embedded System
• I need to build a complicated FPGA board
• I need to manage clock recovery process to get correct data, otherwise is garbage
• I need to ensure all the timing logic are appropriate
• I need to deploy a soft core to build a processing unit
• I need to create custom instruction to drive the programmable logic transceiver from processing unit
• I need to implement decision making and analysis modules to parse data properly
GOING FURTHER - HACKING MACHINE IN YEAR 2017

• EDA tools can do almost all the previously mentioned works, in semi-automatically way
  • If someone tell you is full-automatically, then this is either a salesman or a crook
• Soft core might be quite resource intensive, but there are so many FPGAs come with embedded hard core
• It is easier to develop high performance signal level hacking machine in 2017
• But the entry barrier for this route is still quite high
• Any alternative?
Connection matrix will try to identify the bus

For those unidentified buses, it can switch to logic analyzer-assisted mode to visualize the signals

Once identified, launch a signal parser at the computer to dump data
- Oversampled log
- Log parser

For active attack, such as signal injection, FPGA is still needed to create the signals
GOING FURTHER – HACKING MACHINE FOR NON-HARDWARE GEEKS (INTUITIVE)

- Connection matrix will try to identify the bus
- For those unidentified buses, it can switch to logic analyzer-assisted mode to visualize the signals
- Once identified, launch a signal parser at the computer to dump data
  - Oversampled log
  - Log parser
- For active attack, such as signal injection, FPGA is still needed to create the signals
GOING FURTHER – SCA ASSISTIVE DEVICE

- SCA needs to process huge number of samples
- Capacity of sample is limited (proportional to cost)
- You have to spot and trigger the sampling process at the right point
  - Console message via UART
  - Breakpoint via JTAG
  - Pattern recognition via oscilloscope
GOING FURTHER – SCA ASSISTIVE DEVICE (INTUITIVE)

- SCA needs to process huge number of samples
- Capacity of sample is limited (proportional to cost)
- You have to spot and trigger the sampling process at the right point
  - Console message via UART
  - Breakpoint via JTAG
  - Pattern recognition via oscilloscope
GOING FURTHER – FI ASSISTIVE DEVICE

- FI will cause target to make mistake while executing codes
- The mistake can be temporary or permanent
- Permanent can be within a boot cycle or bricked
- Normally in terms of clock or supply voltage
- Clock normally effective to non-self-clock-synthesis device
- Affect the code execution or pipelining of the processor
- PLL has lock-time, how about fault in jitter form
- Supply voltage fault normally in hiccup form, otherwise, high voltage can spoil the target
- Triggering device determines the moment to inject fault
- Feedback in terms of data or signal to determine result of FI
GOING FURTHER – FI ASSISTIVE DEVICE (INTUITIVE)

- FI will cause target to make mistake while executing codes
- The mistake can be temporary or permanent
- Permanent can be within a boot cycle or bricked
- Normally in terms of clock or supply voltage
- Clock normally effective to non-self-clock-synthesis device
- Affect the code execution or pipelining of the processor
- PLL has lock-time, how about fault in jitter form
- Supply voltage fault normally in hiccup form, otherwise, high voltage can spoilt the target
- Triggering device determines the moment to inject fault
- Feedback in terms of data or signal to determine result of FI
GOING FURTHER – QEMU ASSISTIVE DEVICE

• QEMU has its limitation
  • Binary blob RTOS cannot be emulated as simple as ELF file
  • It is unfriendly to build the low level to port a RTOS being dumped and emulate in QEMU
  • You might take times to build one or two, but how about millions? They are unique and seems impossible to automate the build process
  • It is only possible to emulate a piece of codes, but it is not so helpful from system point of view

• Jtag can create context for emulation of code snippet
GOING FURTHER – STATIC ANALYSIS ASSISTIVE DEVICE

• Static analysis to find potential vulnerable points
• JTAG gaining context for static analysis
• Verify potential bug in real device
• Finding a way to automate bug finding process for embedded system
• Today topic is about hardware, will talk about this in the future
EVILSPLOIT – THE PHYSICAL BOARD
SUMMARY

• Connection matrix has bridged the gap between bus identification and bus manipulation

• It allows higher automation level of hardware hacking process

• A universal hardware hacking toolkit has been proposed

• Future works will focus in automating the hardware hacking process to the max

• Expect us in future conference
DEMO 1

• UART pins finding and control
DEMO 2

• JTAG pins finding and control
**DEMO 3**

- SM4 cracking by using EvilsplIoit as SCA’s assistive device
- Target is a soft crypto
- The crypto process takes 0.5 ms
- ChipWhisperer is the SCA tool
- Only used for samples capturing
- We used our highly optimized MATLAB script to crack the secret key
DEMO 4

- 3DES cracking by using Evilsploit as SCA’s assistive device
- Target is a hard crypto
- The crypto process takes 27 us
- We used ChipWhisperer in the same way again
- Messy signal tuning process
- We still manage to crack the secret key finally
ARSENAL SESSION

• Please join our arsenal session for real stuff demo
  • Demo 1
  • Demo 2
  • Demo 3

• Demo 4 will not be included due to time consuming issue, but will show a more detailed video
TAKEAWAYS

Try to imagine, with Evilsplot, you can manipulate JTAG without knowing what’s the hell of TDI, TDO, TMS, and TCK.
FURTHER INFORMATION AND UPDATES

- [www.evilsplot.com](http://www.evilsplot.com)
- [https://github.com/evillabs/EvilSploit](https://github.com/evillabs/EvilSploit)
THANK YOU
Q & A
SPECIAL THANKS TO

GuangZhou WeehourSEC Information Technology Co., Ltd.
The h4rdenedzer0 team