JTAG Security and Trust

M. Tehranipoor

Introduction to Hardware Security & Trust
University of Florida

Agenda

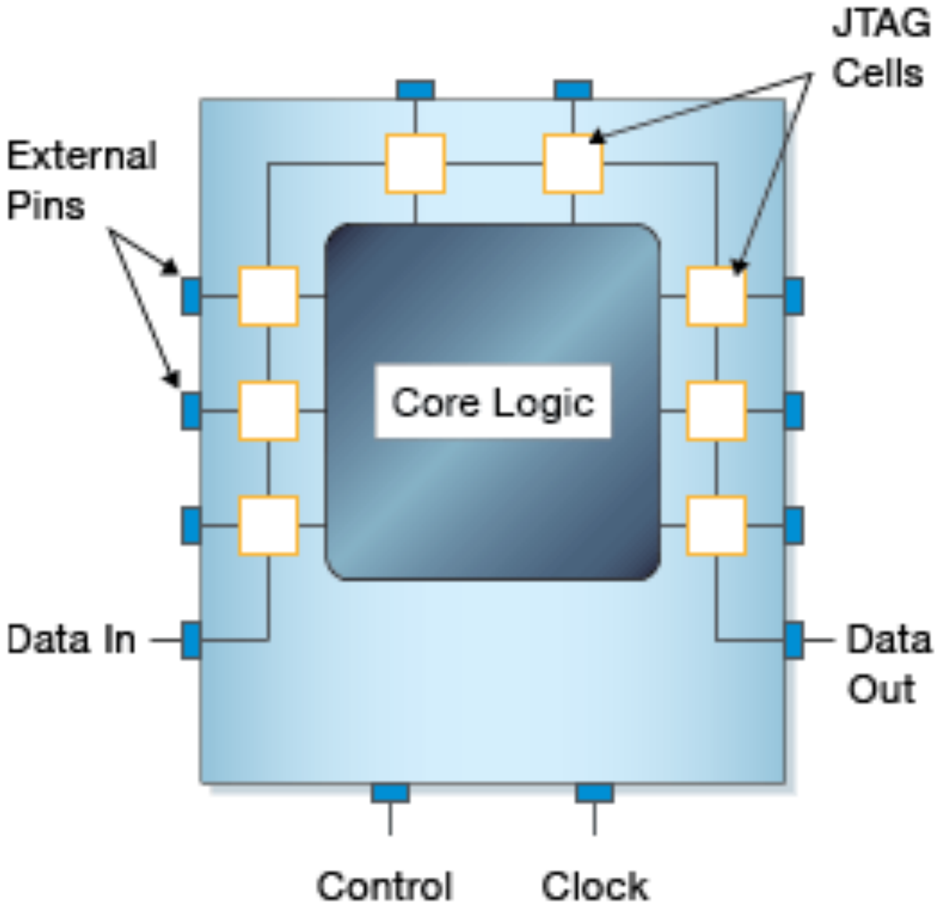
- Introduction to JTAG
- High-Level JTAG Exploits
- Popular JTAG Exploits
- Security Options

JTAG Introduction

- ‘‘JTAG’’ refers to IEEE Std. 1149.1, Standard Test Access Port and Boundary Scan Architecture
- IEEE Std. 1532, Boundary-Scan-Based In-System Configuration of Programmable Devices

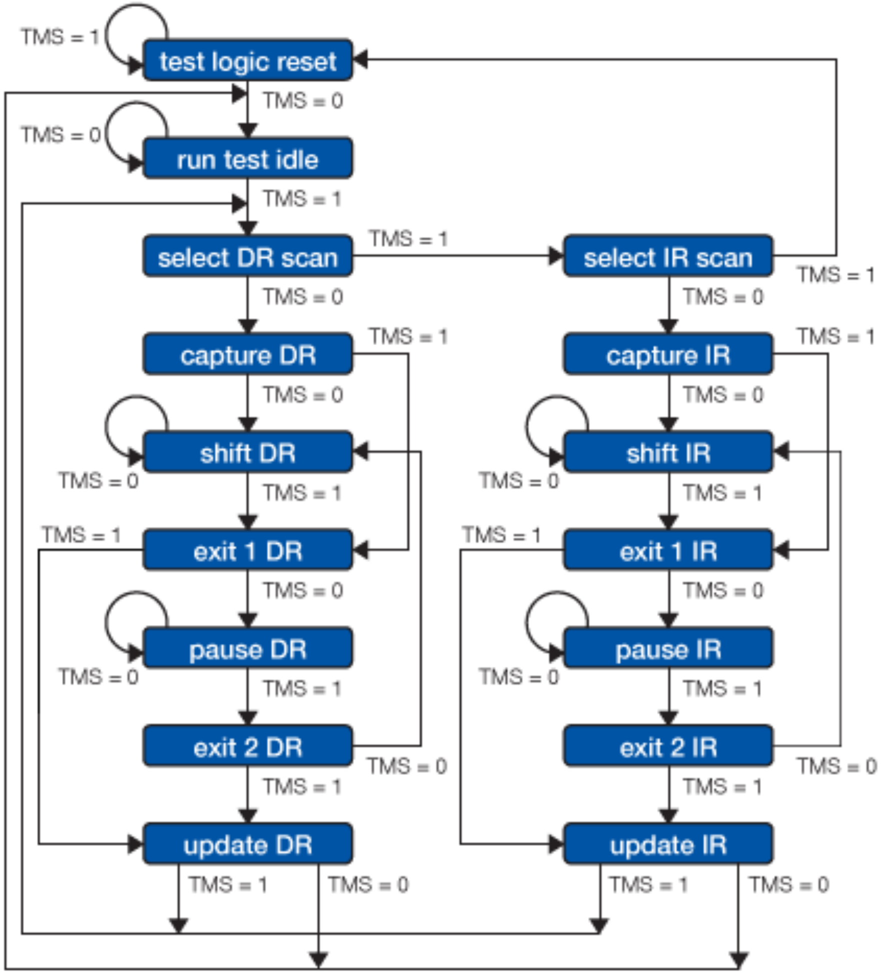
Goals/Benefits of JTAG

- Low Cost
- Inter-circuit testing without need of physical test-probes
- Increased fault detection coverage
- Lower test time



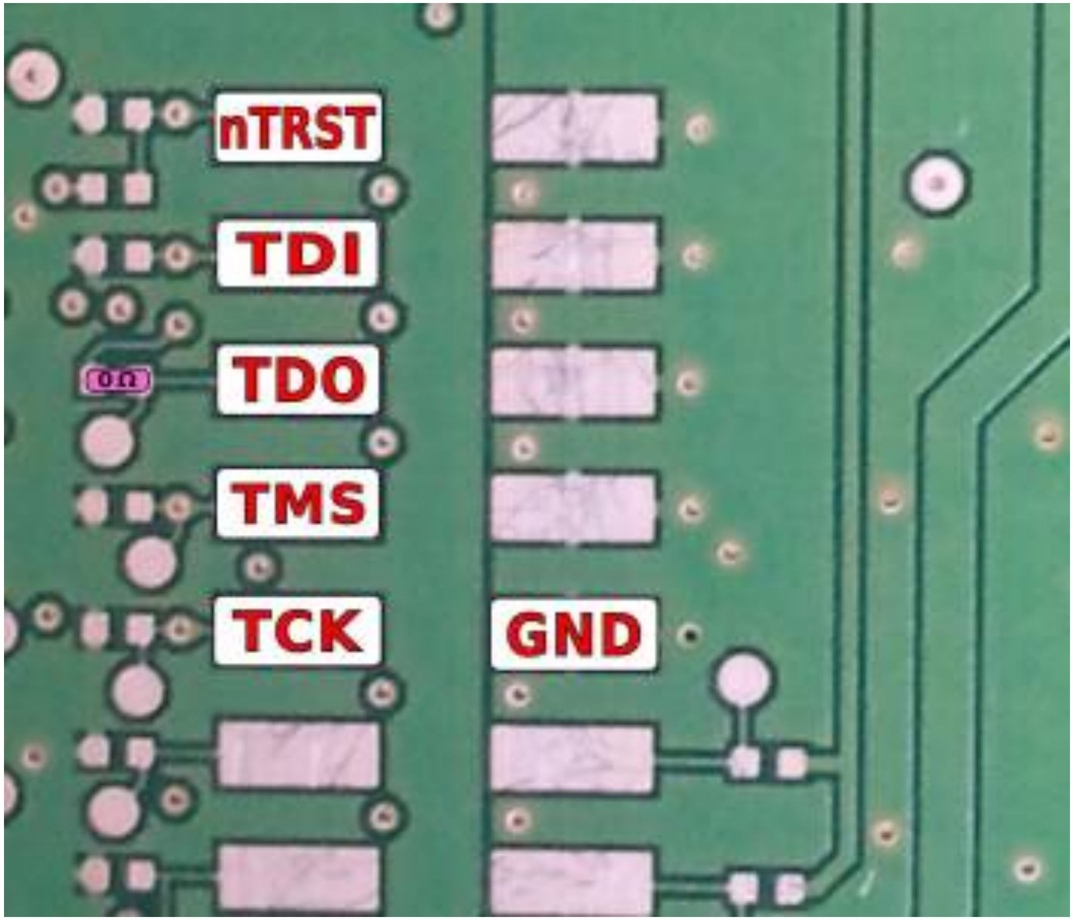
Physical Components

- TAP
  - Test Access Port
  - Interprets JTAG protocol
  - Controlled by TMS signal
- BSR
  - Boundary Scan Registers
  - Between module and TAP



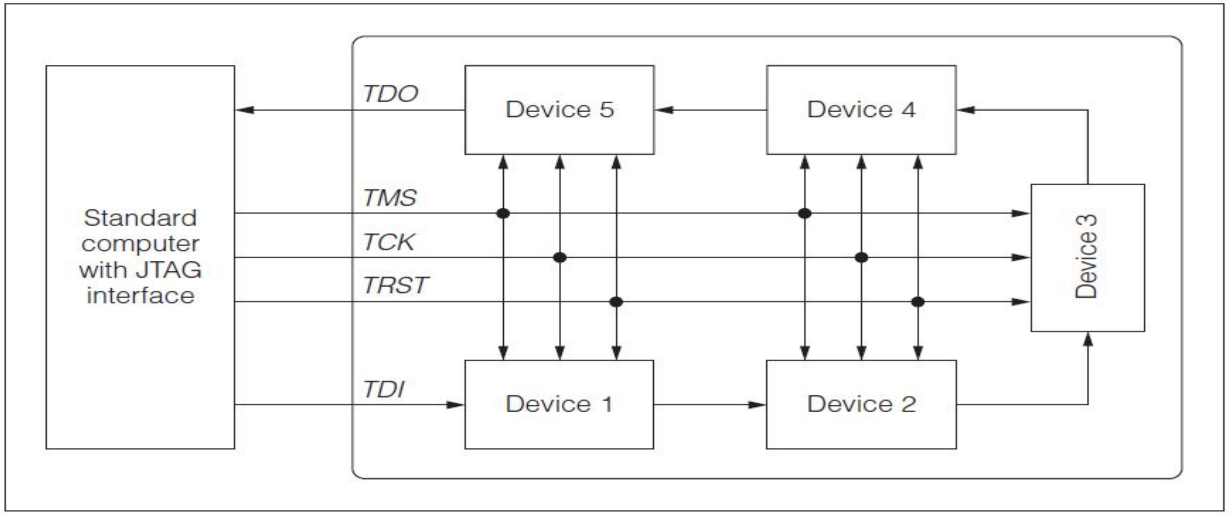
JTAG Control

- TDO
  - Test Data Output
- TDI
  - Test Data Input
- TMS
  - Test Mode Select
- TCK
  - Test Clock
- TRST
  - Resets TAP Controller



JTAG Modes

- Bypass
  - Connects TDI to TDO
  - One cycle delay
- ExTest
  - Asserts data on output pins
  - Reads data from input pins
- InTest
  - Asserts data on input pins
  - Reads data from output pins



JTAG Overview

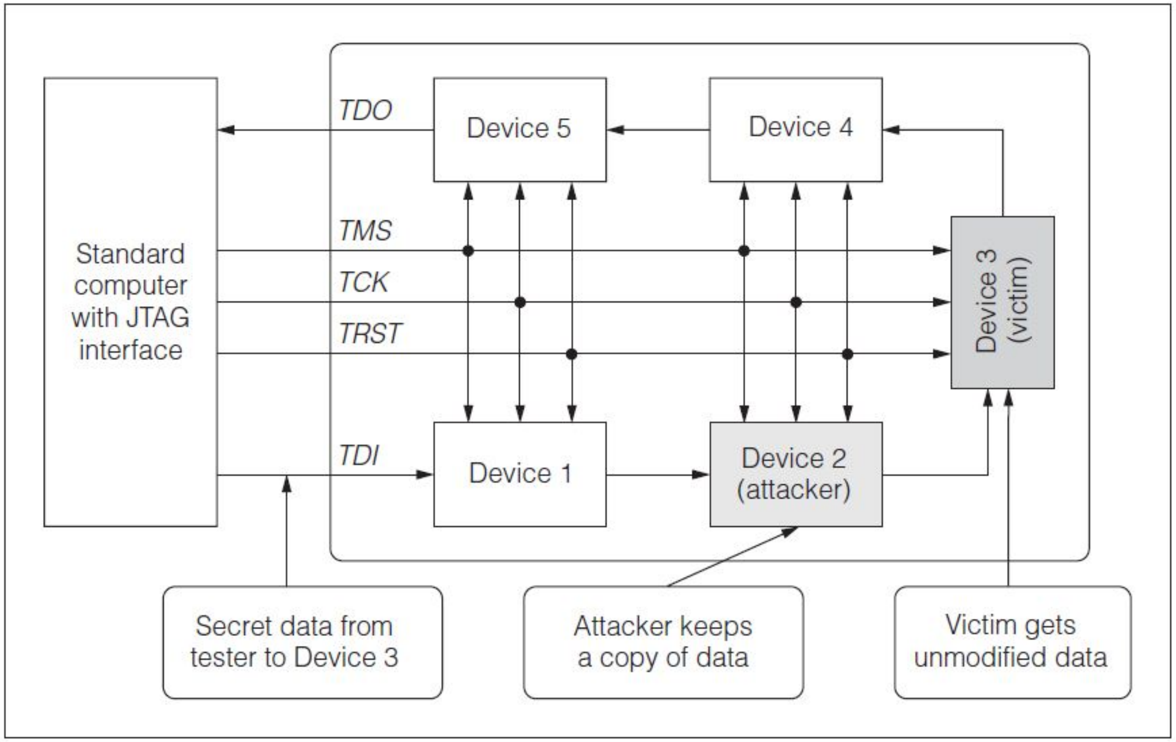
- JTAG Benefits
  - Low Cost
  - Ease of testing
- Physical Components
  - TAP, BSR
- JTAG Pins
  - TDI, TDO, TMS, TCK, TRST
- JTAG Modes
  - Bypass, ExTest, InTest

High-Level JTAG Exploits

- Sniff TDI/TDO signals
- Modify TDI/TDO signals
- Control TMS and TCK signals

Sniff TDI/TDO Signals

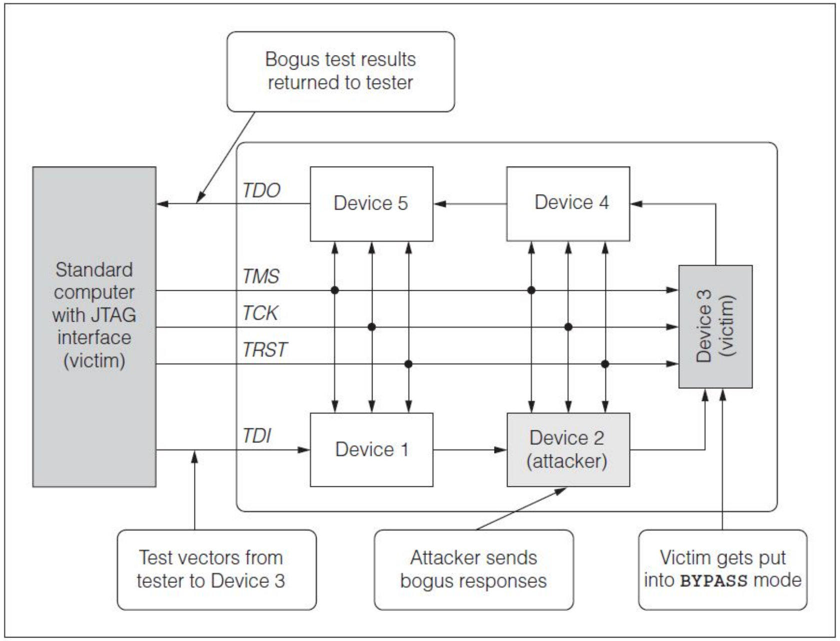
- Used to intercept secrets being sent to or from a chip
- Preceding or chip after victim chip behaves differently during bypass to intercept message



TDO
Device 5
Device 4
Standard computer with JTAG interface
TMS
TCK
TRST
Device 3 (victim)
TDI
Device 1
Device 2 (attacker)
Secret data from tester to Device 3
Attacker keeps a copy of data
Victim gets unmodified data

Modify TDI/TDO Signals

- Can modify Test Vectors and Test Responses
- Can be used to fake correct or false tests
- Attacker can either be upstream or downstream of victim based on attack

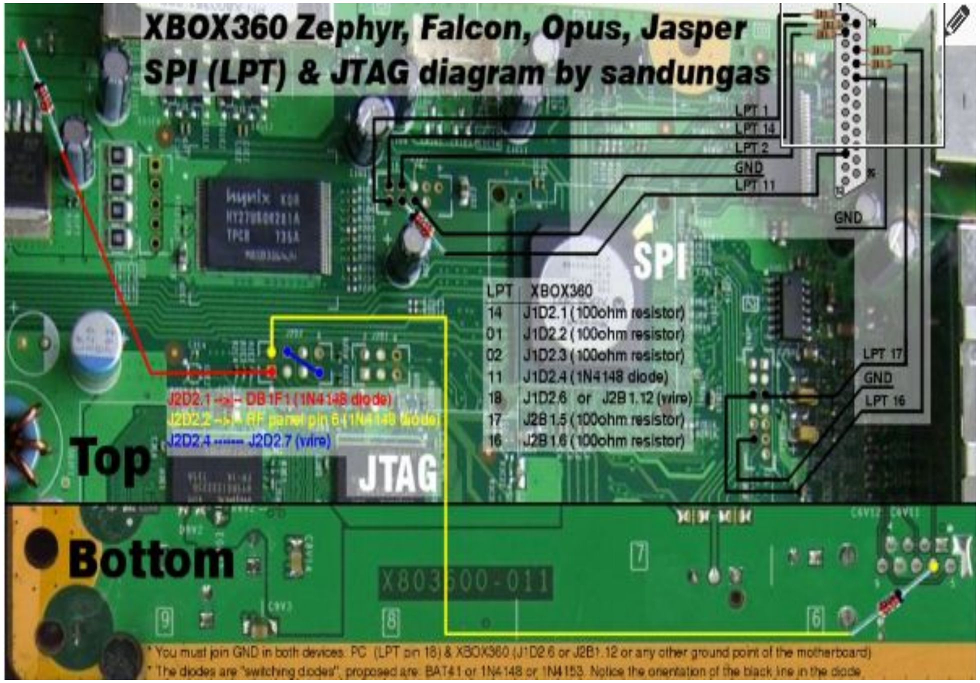

Bogus test results returned to tester
TDO
Device 5
Device 4
Standard computer with JTAG interface (victim)
TMS
TCK
TRST
Device 3 (victim)
TDI
Device 1
Device 2 (attacker)
Test vectors from tester to Device 3
Attacker sends bogus responses
Victim gets put into BYPASS mode

Control TMS and TCK Signals

- For many exploits, TMS and TCK signals need to be controlled
- Attacker needs to be able to overpower The signal sent by TAP
- Attacking device needs to be able to force TMS and TCK above or below logic threshold voltage
- Can be done by combining lines to make a more powerful driver or using multiple attackers to overcome TMS and TCK signals

Xbox 360 Exploit

- Used to override Microsoft security features
- Allows homebrew code to be run, installation of HD, game modification, ripping of games
- JTAG is used to extract secret keys needed to perform exploits and to change programming



XBOX360 Zephyr, Falcon, Opus, Jasper
SPI (LPT) & JTAG diagram by sandungas
LPT 1
LPT 14
LPT 2
GND
LPT 11
GND
SPI
LPT 17
GND
LPT 16
J2D2.1 --»-- DB1F1 (1N4148 diode)
J2D2.2 --»-- RF panel pin 6 (1N4148 diode)
J2D2.4 ------ J2D2.7 (wire)
Top
JTAG
Bottom
X803600-011
9
8
7
6

LPT	XBOX360
14	J1D2.1 (100ohm resistor)
01	J1D2.2 (100ohm resistor)
02	J1D2.3 (100ohm resistor)
11	J1D2.4 (1N4148 diode)
18	J1D2.6 or J2B1.12 (wire)
17	J2B1.5 (100ohm resistor)
16	J2B1.6 (100ohm resistor)

* You must join GND in both devices: PC (LPT pin 18) & XBOX360 (J1D2.6 or J2B1.12 or any other ground point of the motherboard)

* The diodes are "switching diodes", proposed are: BAT41 or 1N4148 or 1N4153. Notice the orientation of the black line in the diode

Security Options

- Buffers in the JTAG Chain

- JTAG system connected in “Star” pattern instead of being chained(Separate TMS and TCK)

- Encryption/Authentication for JTAG use
  - Most of the research in JTAG security would be classified under this
  - Although it would provide much better protection, like all security hardware, increases cost and space.

Challenge, Response

- Requires PUF or randomly burned fuses
- Requires Set_Challenge and Get_Response instructions in JTAG implementation
- A Challenge input is given to the JTAG module, and the module will hash this with the value of it' s fuses to create the response
- Only a known, trusted module will give a correct response
- So, can be determined if modules are trusted or not

Public/Private Key Authentication

- Tester/Updater is required to have a certificate of authentication signed by a designated third party.
- Authenticators public key is known to JTAG system
- Using the known public key, the JTAG system can decrypt the certificate and determine whether the tester/updater is trusted
- Trusted testers/updaters are allowed access to JTAG system, un-trusted are blocked

User Permissions

- A user permission level, i, allows them access to instructions with a level less than i
- Requires extra hardware to authenticate user and set permission level, and to save settings for what each permission level can and cannot do
- Ex. In memory, a permission level is saved for each module in the JTAG system. When that module is trying to be accessed, the saved level is compared to the current permission level

Removal/Destruction of JTAG

- To completely defend against JTAG attacks, one thought is to remove the JTAG hardware all together
  - Does not leave a way to in-field test
  - Can use BIST for testing
- Similarly to removal of JTAG, some companies use security fuses to disable JTAG before the hardware leaves the factory
  - Can implement different levels of disabled JTAG use

Acknowledgments

- Kumar, P. A., Kumar, P. S., & Patwa, A. (2012). *Jtag architecture with multi level security*. Manuscript submitted for publication, Department of Electronics and Communication, Amrita School of Engineering, Bangalore, India.
- Pierce, L., & Tragoudas, S. (2011). *Multi-level secure jtag architecture*. Manuscript submitted for publication, Department of Electrical and Computer Engineering, Southern Illinois University, Carbondale, .
- Rosenfeld, K., & Karri, R. (2010). *Attacks and defenses for jtag*. Manuscript submitted for publication, Polytechnic Institute of New York University, New York, NY, .
- The IEEE Std 1149.1-1990 - Test Access Port and JTAG Architecture, and the Std 1149.1-1994b - East 47th Street, New York, NY.
- Corelis (An EWA Company), (n.d.). *Boundary-scan for pcb interconnect testing and in-system programming of cplds and flash memories*. Retrieved from website: http://www.corelis.com/whitepapers/Boundary-Scan_Whitepaper.pdf
- Rosenfeld, K., & Karri, R. (2012). Chapter 17: Security and testing. In M. Tehranipoor & C. Wang (Eds.),*Introduction to Hardware Security and Trust* doi:www.springer.com
- *How to jtag your xbox 360 and run homebrew*. (n.d.). Retrieved from http://www.instructables.com/id/How-to-JTAG-your-Xbox-360-and-run-homebrew/