NSA Playset:
JTAG Implants

Introductory Rites

Today's Clergy

- Electrical Engineering education with focus on CS and Infosec
- 10 years of fun with hardware
  - silicon debug
  - security research
  - pen testing of CPUs
  - security training
- Hardware Security Training:
  - Secure RTL design
  - Low-cost physical attacks
  - "Applied Physical Attacks on x86 Systems"

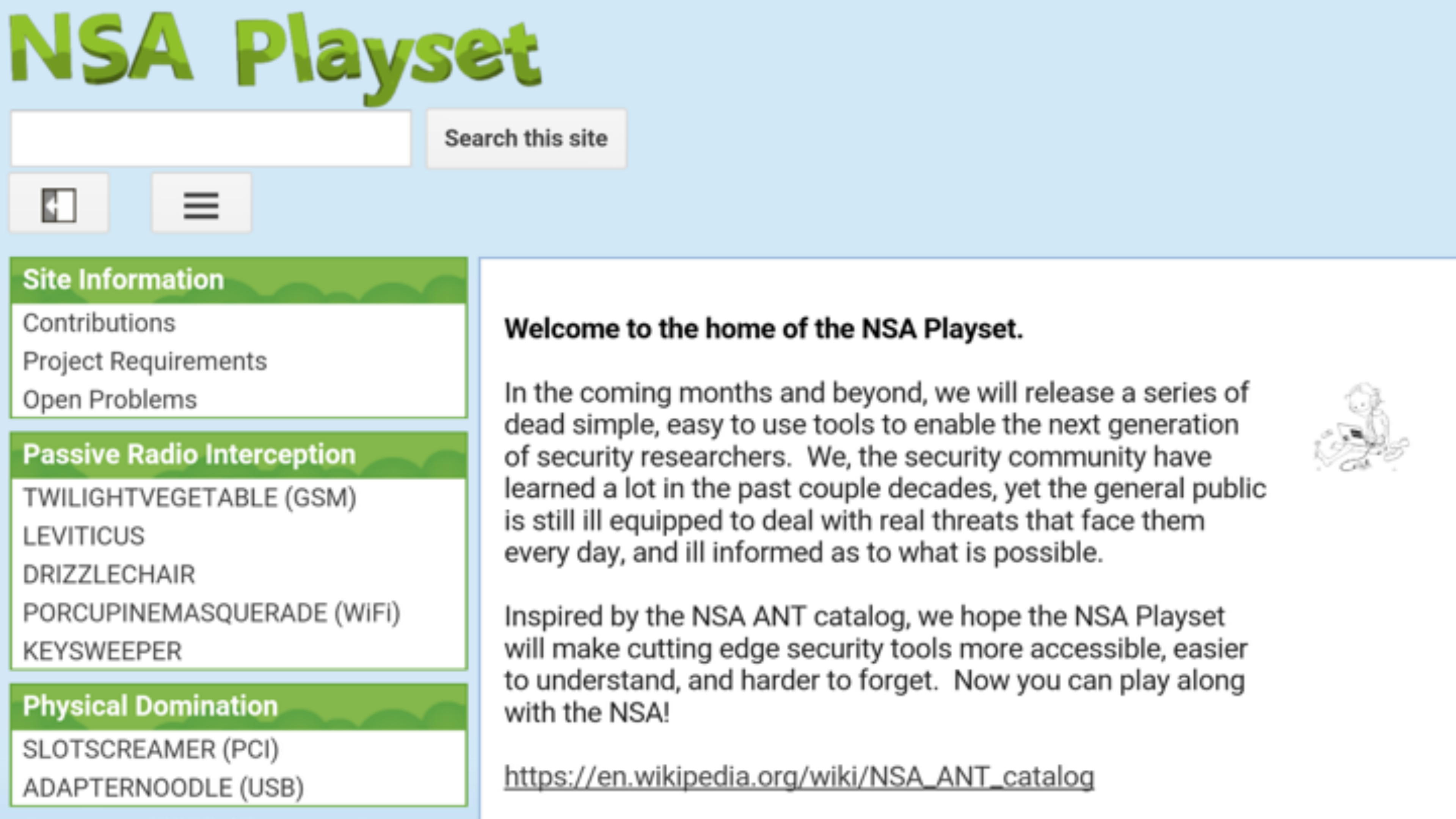
Joe FitzPatrick
@securelyfitz
joefitz@securinghardware.com

SECURINGHARDWARE.COM

Today's Clergy

- Degrees in Electrical and Computer Engineering
- 10+ years designing, implementing, and testing SoC silicon debug features
- Hardware and firmware pentesting

Matt King
@syncsrc
jtag@syncsrc.org

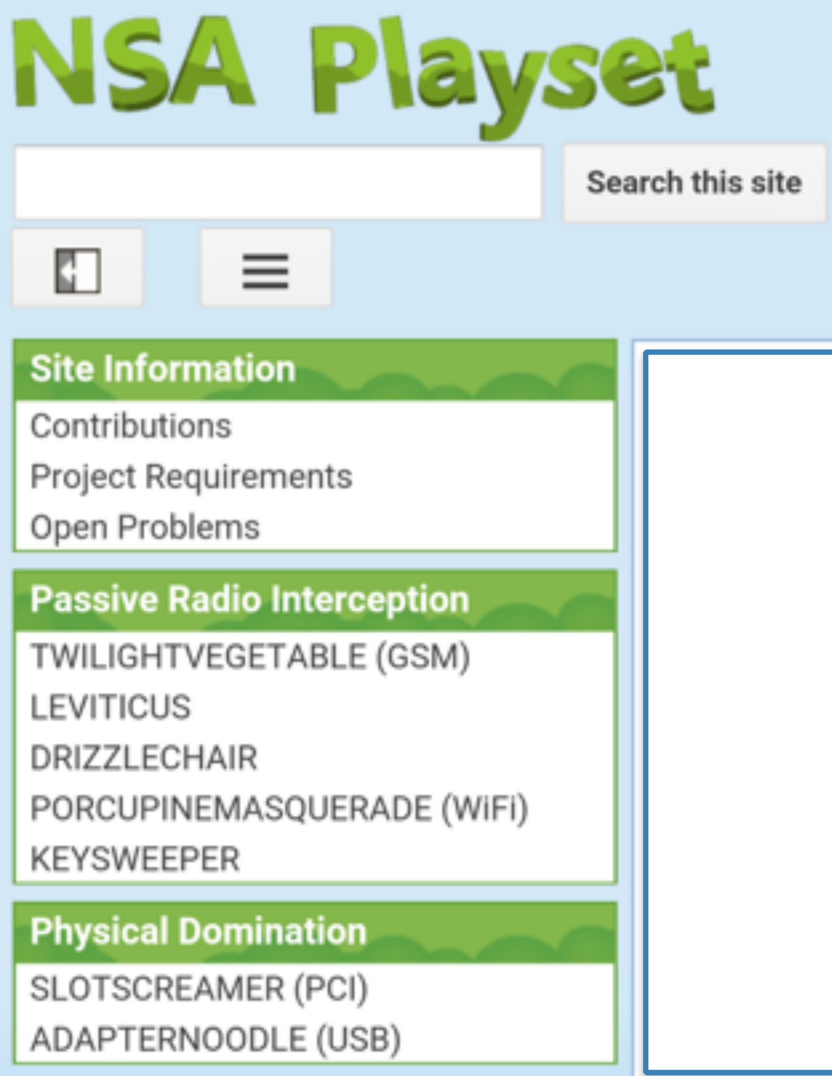


Welcome to the home of the NSA Playset.

In the coming months and beyond, we will release a series of dead simple, easy to use tools to enable the next generation of security researchers. We, the security community have learned a lot in the past couple decades, yet the general public is still ill equipped to deal with real threats that face them every day, and ill informed as to what is possible.

Inspired by the NSA ANT catalog, we hope the NSA Playset will make cutting edge security tools more accessible, easier to understand, and harder to forget. Now you can play along with the NSA!

https://en.wikipedia.org/wiki/NSA_ANT_catalog



NSA Playset

Site Information
- Contributions
- Project Requirements
- Open Problems

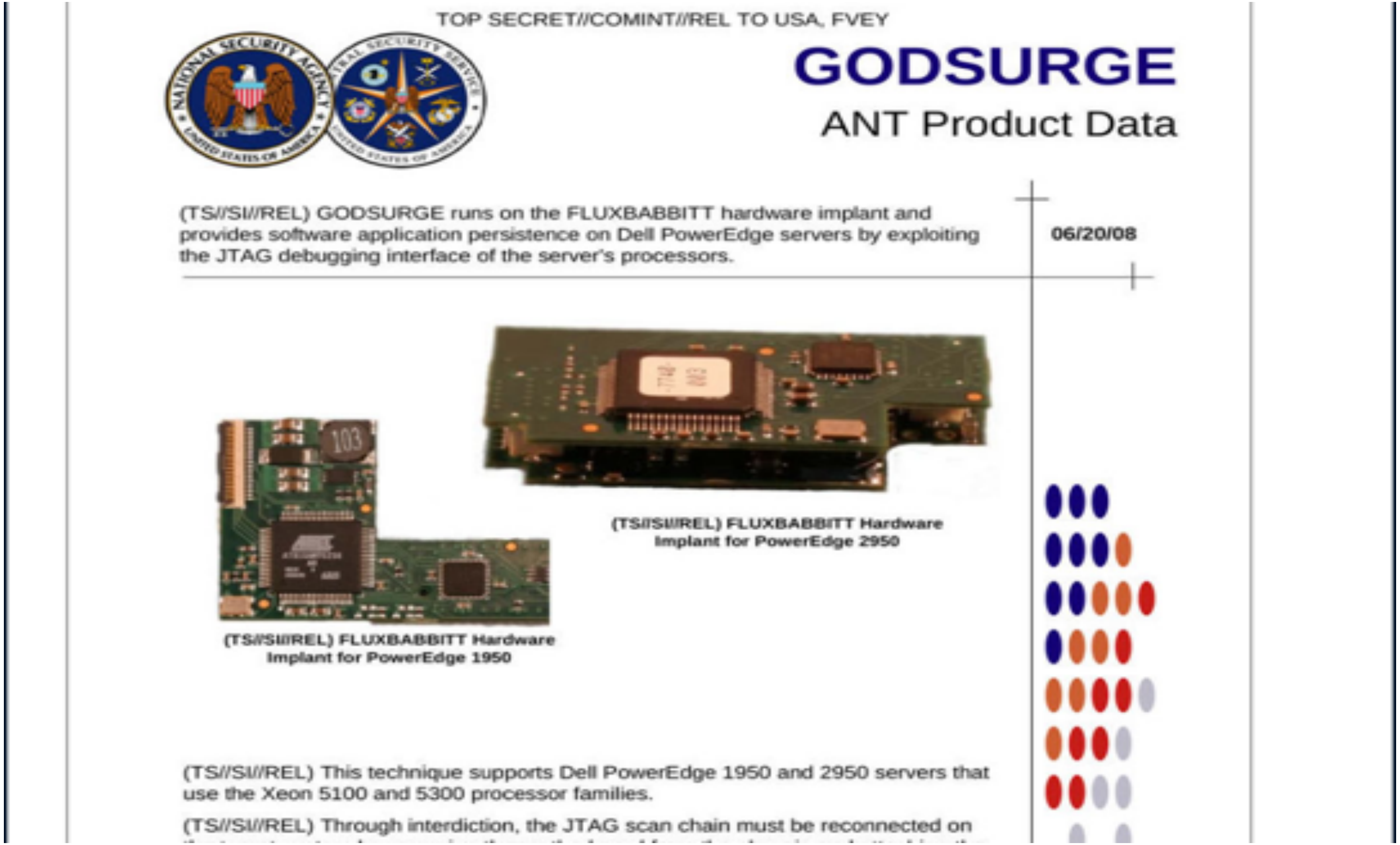
Passive Radio Interception
- TWILIGHTVEGETABLE (GSM)
- LEVITICUS
- DRIZZLECHAIR
- PORCUPINEMASQUERADE (WiFi)
- KEYSWEEPER

Physical Domination
- SLOTSCREAMER (PCI)
- ADAPTERNOODLE (USB)

More toys for sale!

Sunday at
Hacker Warehouse
in the vendor area!

The Penitence of Godsurge & Fluxbabbit



TOP SECRET//COMINT//REL TO USA, FVEY

NATIONAL SECURITY AGENCY · UNITED STATES OF AMERICA

CENTRAL SECURITY SERVICE · UNITED STATES OF AMERICA

GODSURGE

ANT Product Data

(TS//SI//REL) GODSURGE runs on the FLUXBABBITT hardware implant and provides software application persistence on Dell PowerEdge servers by exploiting the JTAG debugging interface of the server's processors.

06/20/08

(TS//SI//REL) FLUXBABBITT Hardware Implant for PowerEdge 2950

(TS//SI//REL) FLUXBABBITT Hardware Implant for PowerEdge 1950

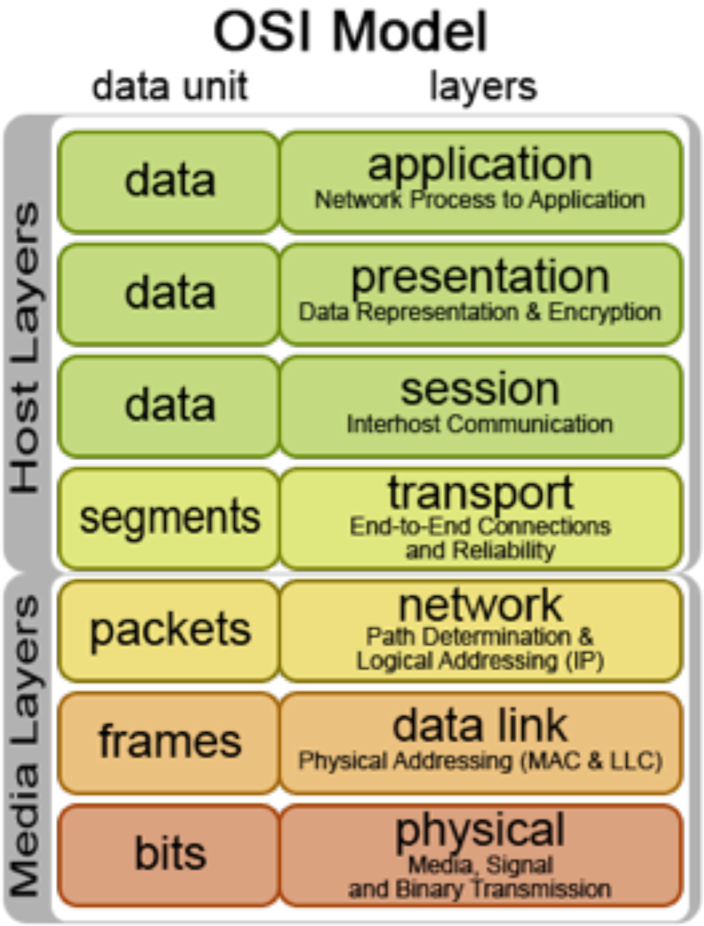
(TS//SI//REL) This technique supports Dell PowerEdge 1950 and 2950 servers that use the Xeon 5100 and 5300 processor families.

(TS//SI//REL) Through interdiction, the JTAG scan chain must be reconnected on

Liturgy of the DWORD: JTAG

Joint
Test
Action
Group

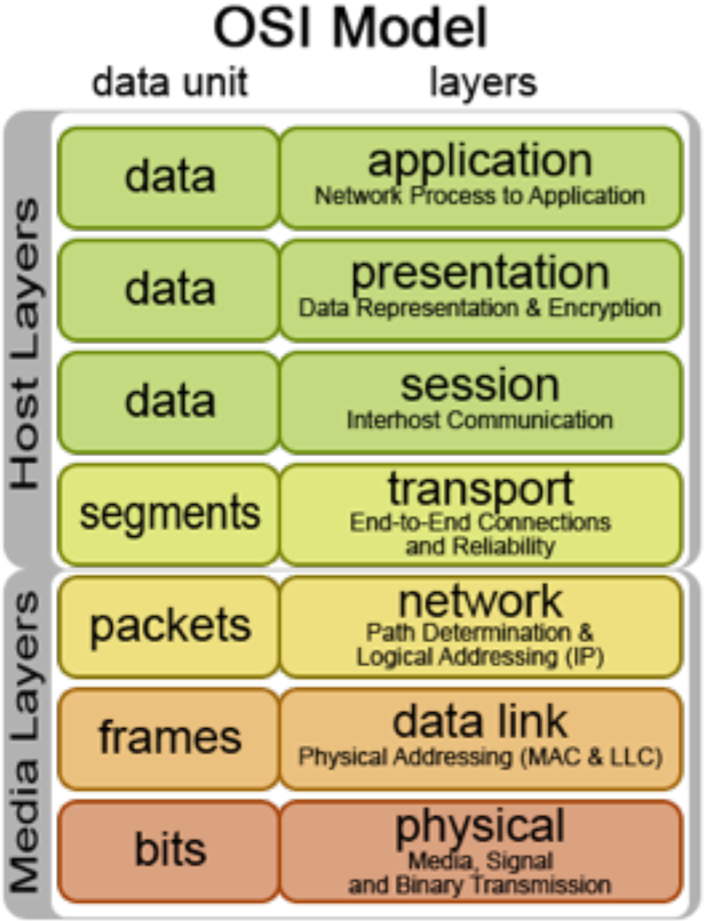
A reading from IEEE 1149



OSI Model

	data unit	layers
Host Layers	data	application Network Process to Application
	data	presentation Data Representation & Encryption
	data	session Interhost Communication
	segments	transport End-to-End Connections and Reliability
Media Layers	packets	network Path Determination & Logical Addressing (IP)
	frames	data link Physical Addressing (MAC & LLC)
	bits	physical Media, Signal and Binary Transmission

Remember This?



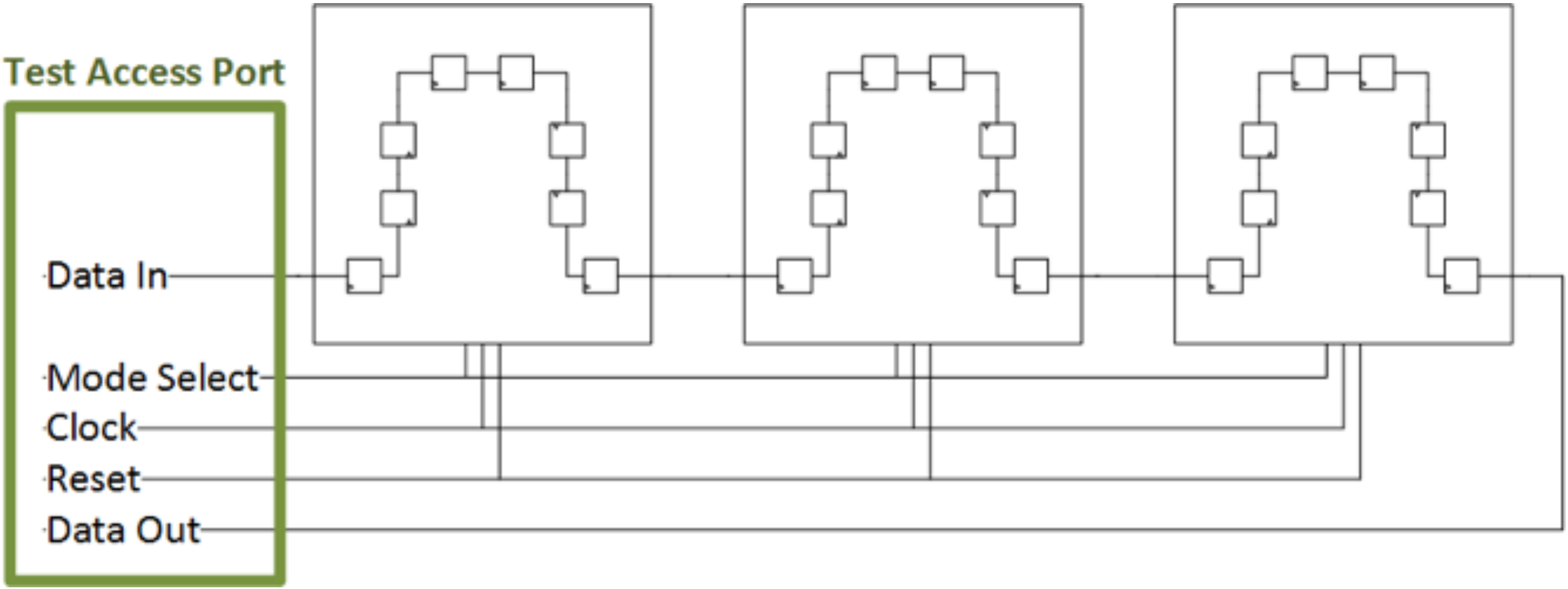
OSI Model

	data unit	layers
Host Layers	data	application Network Process to Application
	data	presentation Data Representation & Encryption
	data	session Interhost Communication
	segments	transport End-to-End Connections and Reliability
Media Layers	packets	network Path Determination & Logical Addressing (IP)
	frames	data link Physical Addressing (MAC & LLC)
	bits	physical Media, Signal and Binary Transmission

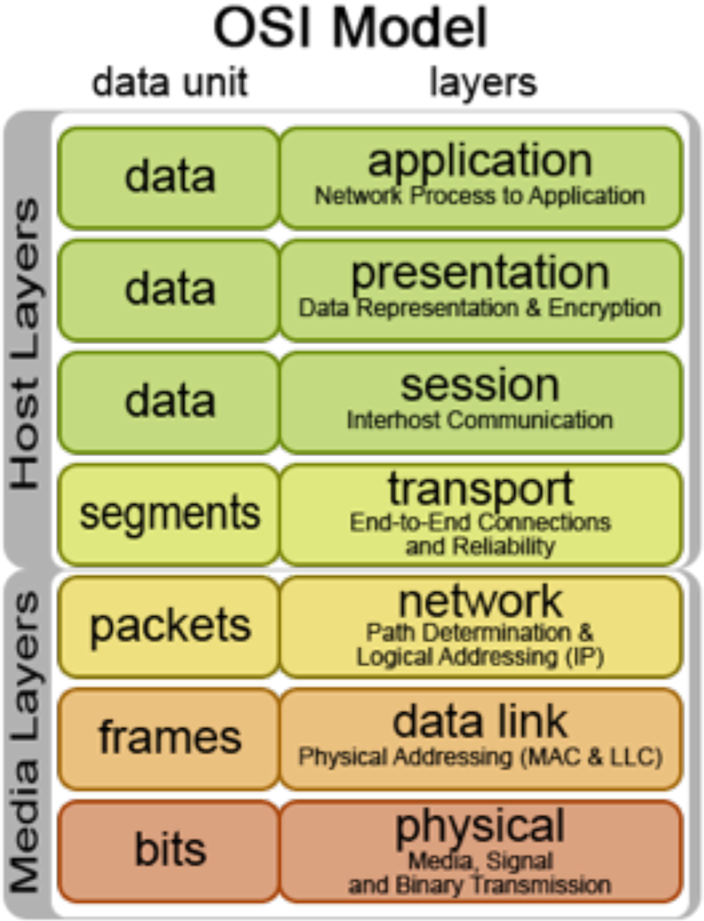
JTAG Model

TDI, TDO, TMS, TCK, TRST

Physical Layer: Test Access Port



TDO unto others
As others TDI unto you



OSI Model

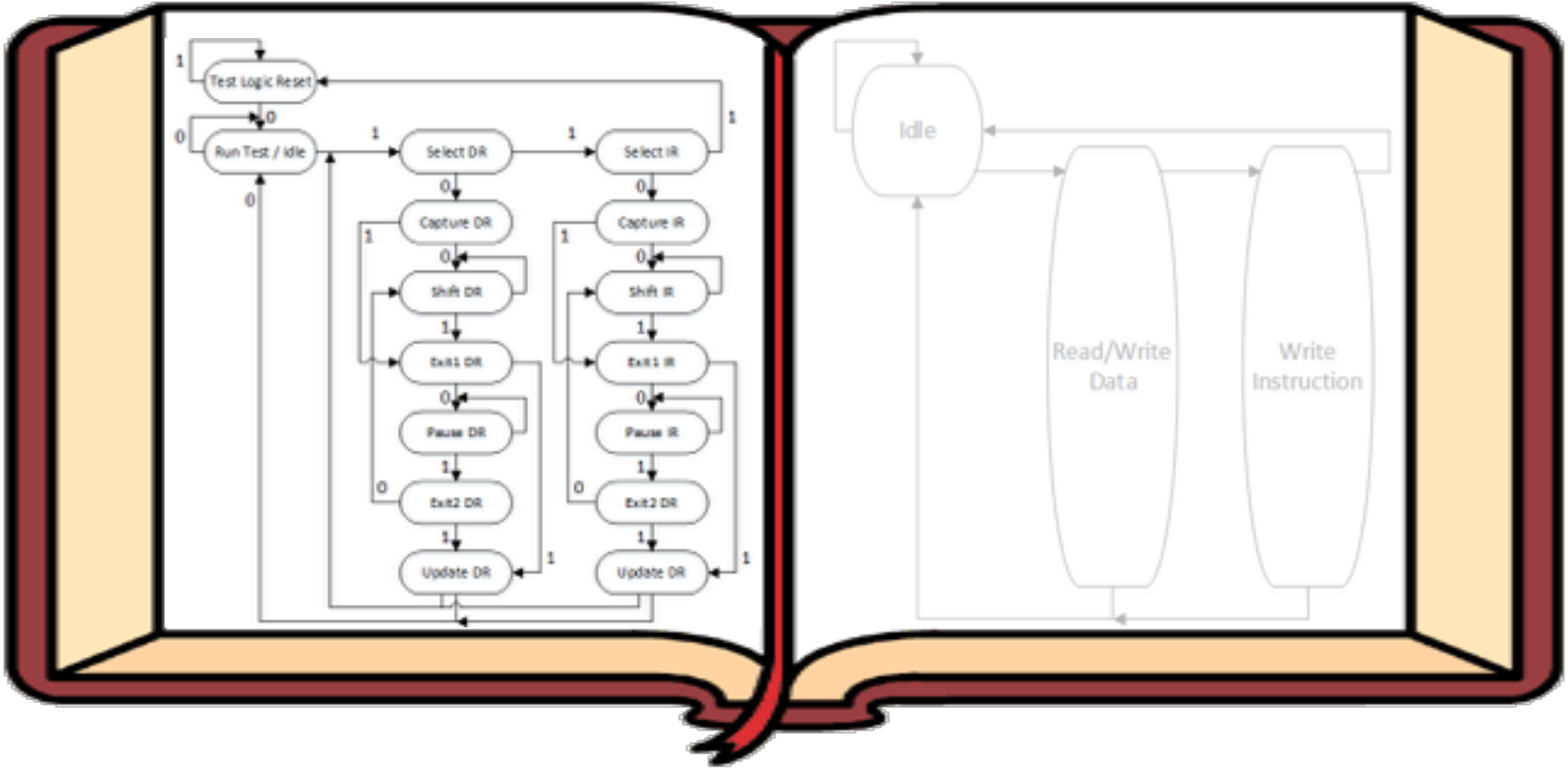
	data unit	layers
Host Layers	data	application Network Process to Application
	data	presentation Data Representation & Encryption
	data	session Interhost Communication
	segments	transport End-to-End Connections and Reliability
Media Layers	packets	network Path Determination & Logical Addressing (IP)
	frames	data link Physical Addressing (MAC & LLC)
	bits	physical Media, Signal and Binary Transmission

JTAG Model

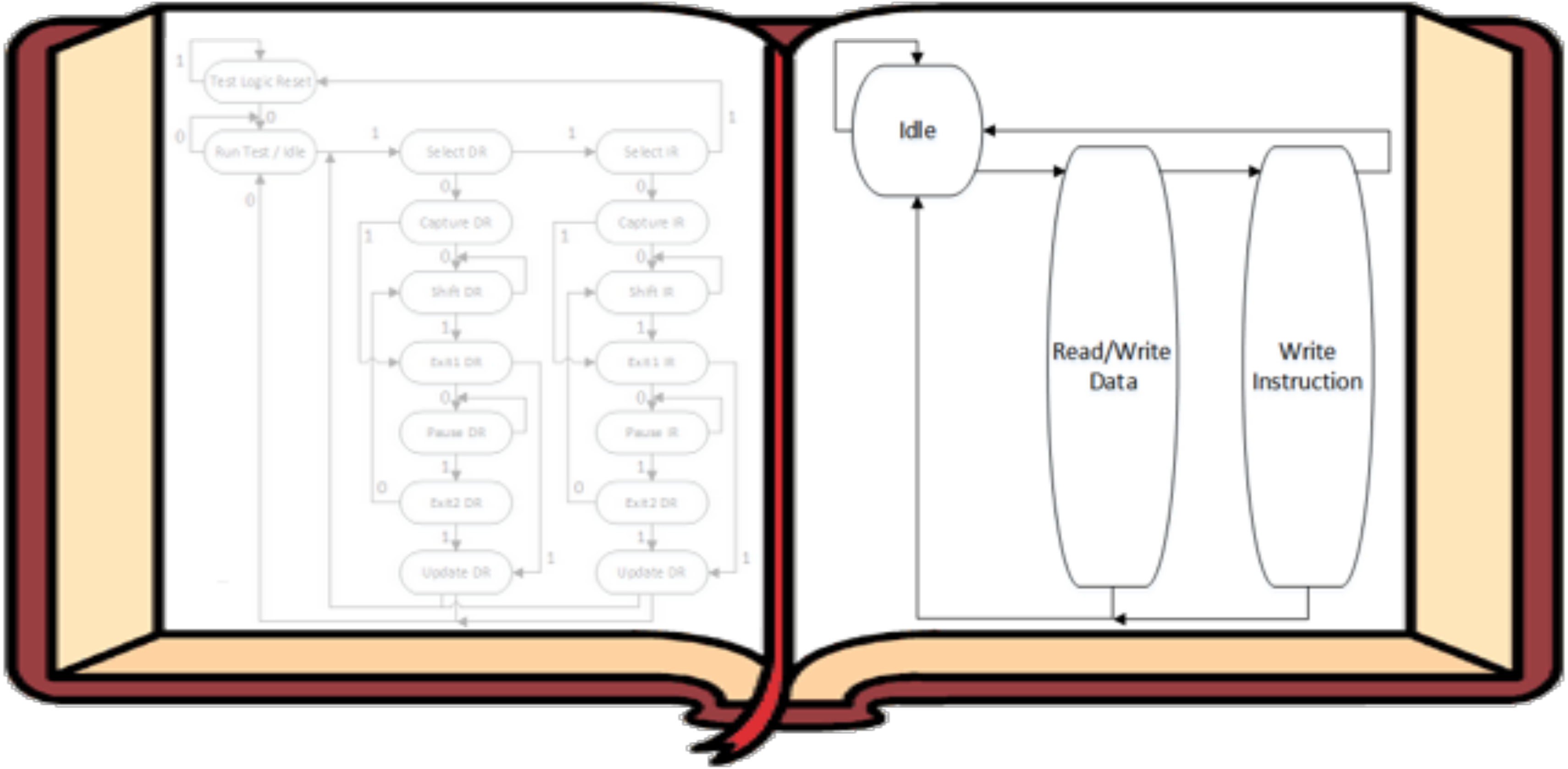
TAP FSM

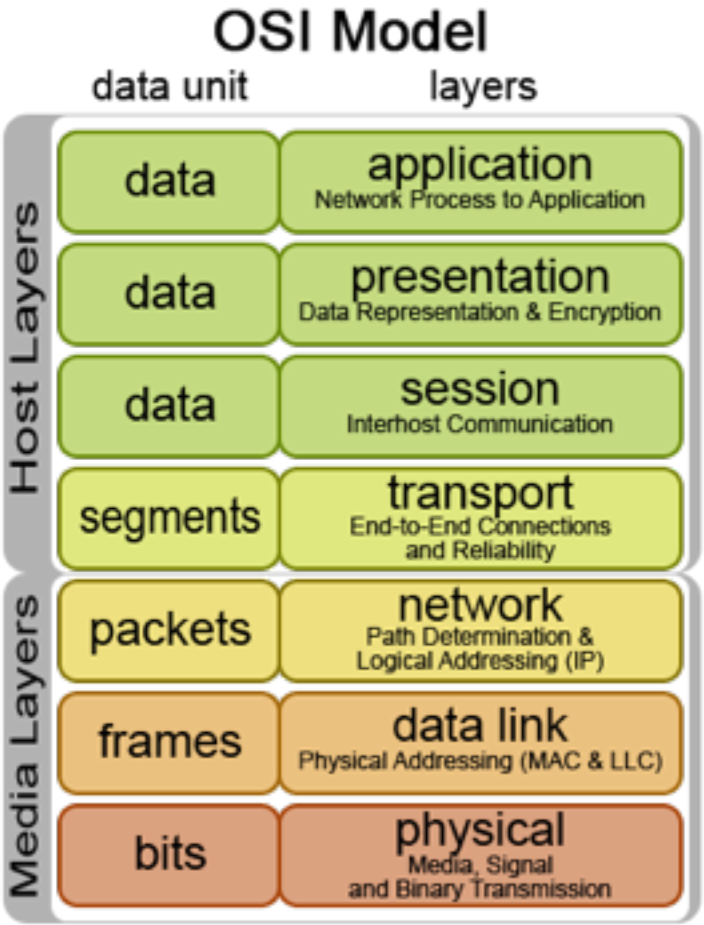
TDI, TDO, TMS, TCK, TRST

Data Link: TAP FSM



Data Link: TAP FSM





OSI Model

	data unit	layers
Host Layers	data	application Network Process to Application
	data	presentation Data Representation & Encryption
	data	session Interhost Communication
	segments	transport End-to-End Connections and Reliability
Media Layers	packets	network Path Determination & Logical Addressing (IP)
	frames	data link Physical Addressing (MAC & LLC)
	bits	physical Media, Signal and Binary Transmission

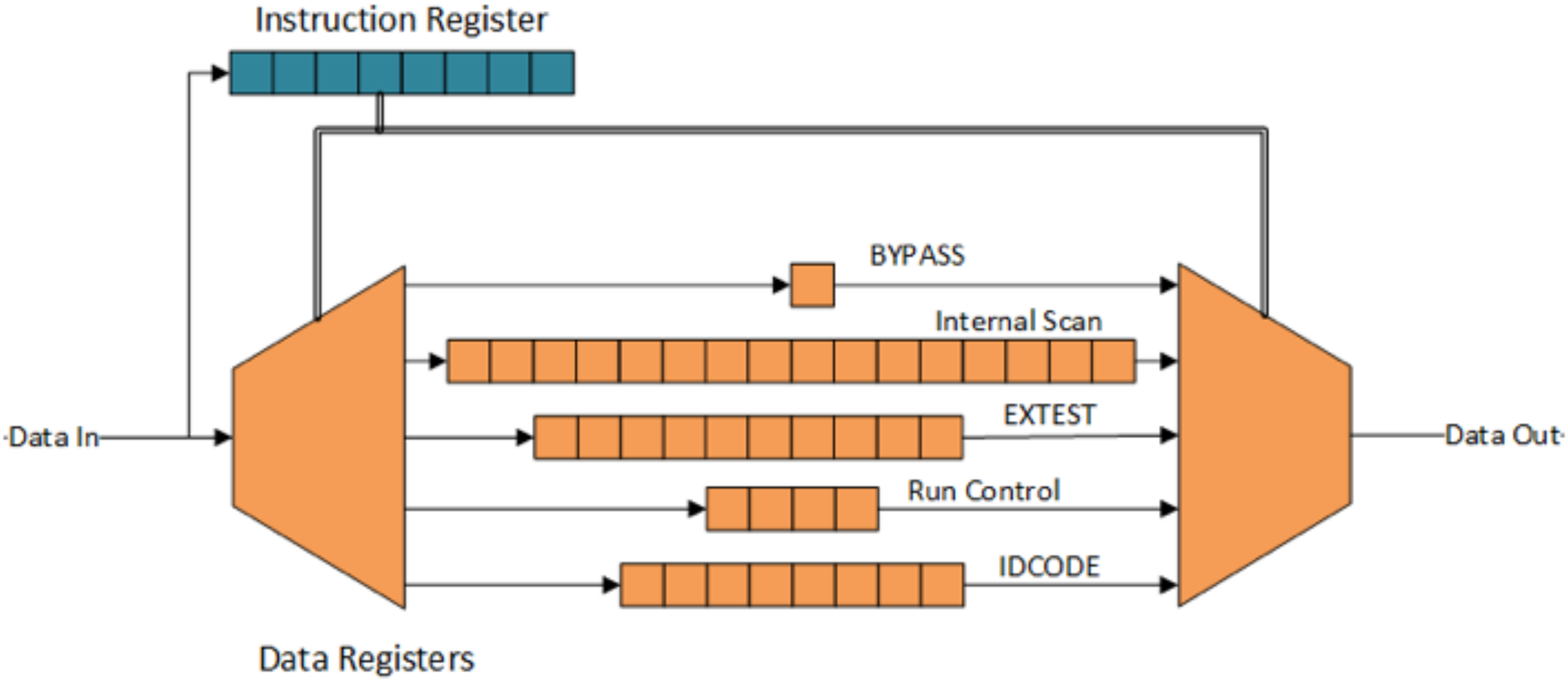
JTAG Model

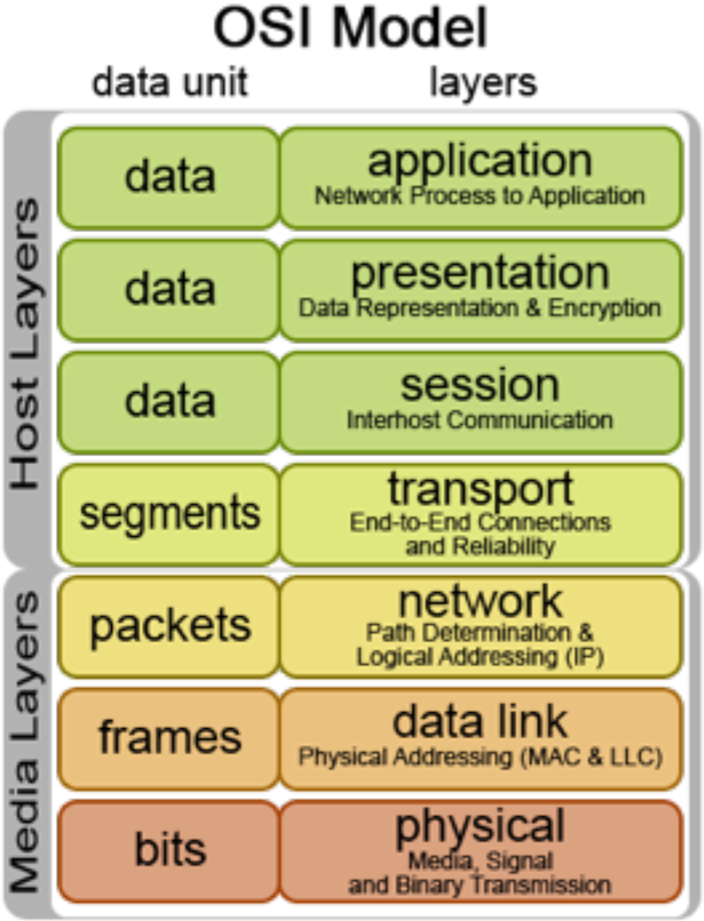
IR/DR access

TAP FSM

TDI, TDO, TMS, TCK, TRST

Network Layer: IRs & DRs





OSI Model

	data unit	layers
Host Layers	data	application Network Process to Application
	data	presentation Data Representation & Encryption
	data	session Interhost Communication
	segments	transport End-to-End Connections and Reliability
Media Layers	packets	network Path Determination & Logical Addressing (IP)
	frames	data link Physical Addressing (MAC & LLC)
	bits	physical Media, Signal and Binary Transmission

JTAG Model

Target-specific configuration

IR/DR access

TAP FSM

TDI, TDO, TMS, TCK, TRST

Transport Layer: Target-Specific

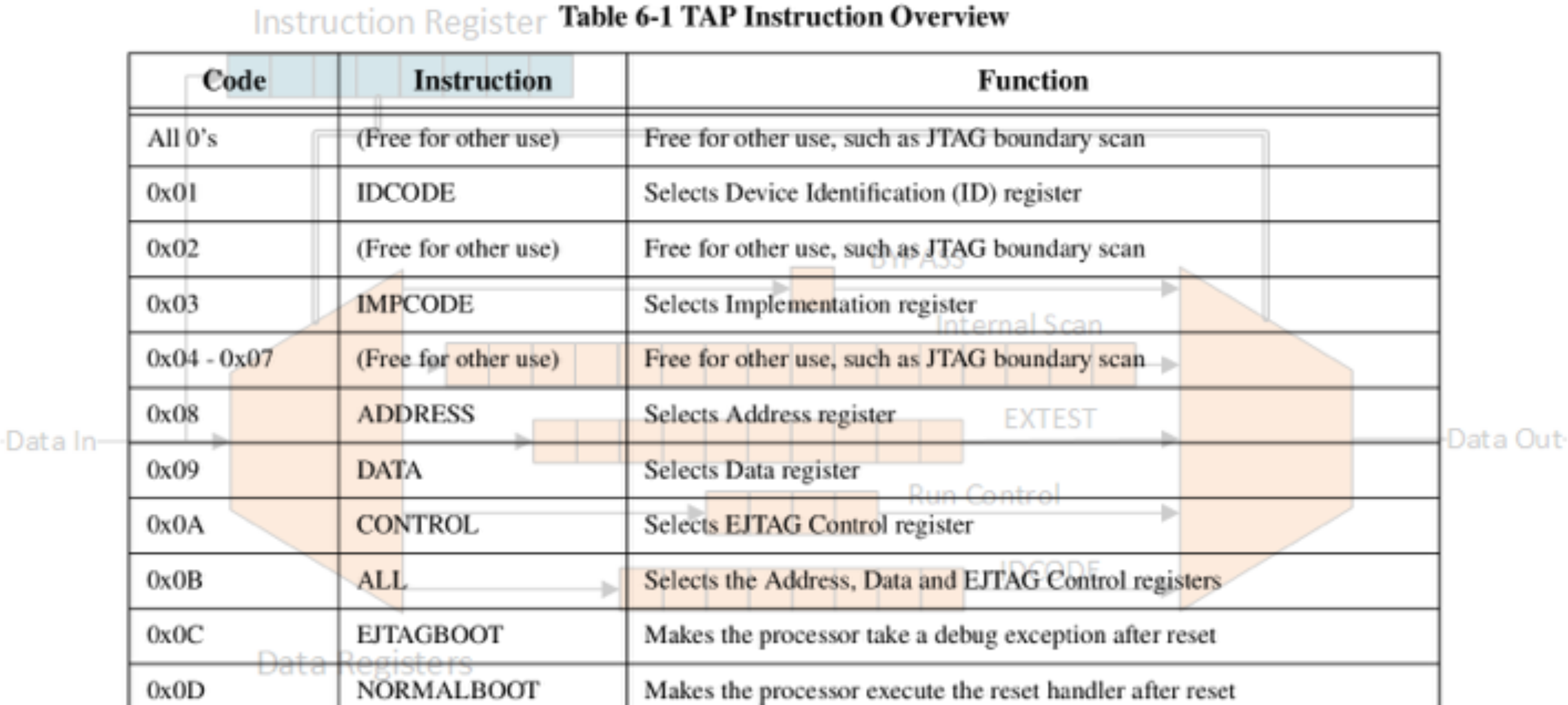


Table 6-1 TAP Instruction Overview

Code	Instruction	Function
All 0's	(Free for other use)	Free for other use, such as JTAG boundary scan
0x01	IDCODE	Selects Device Identification (ID) register
0x02	(Free for other use)	Free for other use, such as JTAG boundary scan
0x03	IMPCODE	Selects Implementation register
0x04 - 0x07	(Free for other use)	Free for other use, such as JTAG boundary scan
0x08	ADDRESS	Selects Address register
0x09	DATA	Selects Data register
0x0A	CONTROL	Selects EJTAG Control register
0x0B	ALL	Selects the Address, Data and EJTAG Control registers
0x0C	EJTAGBOOT	Makes the processor take a debug exception after reset
0x0D	NORMALBOOT	Makes the processor execute the reset handler after reset

That’s just MIPS.

That’s just MIPS.

X86 is different
ARM is different
Each SOC is different

That’s just MIPS.

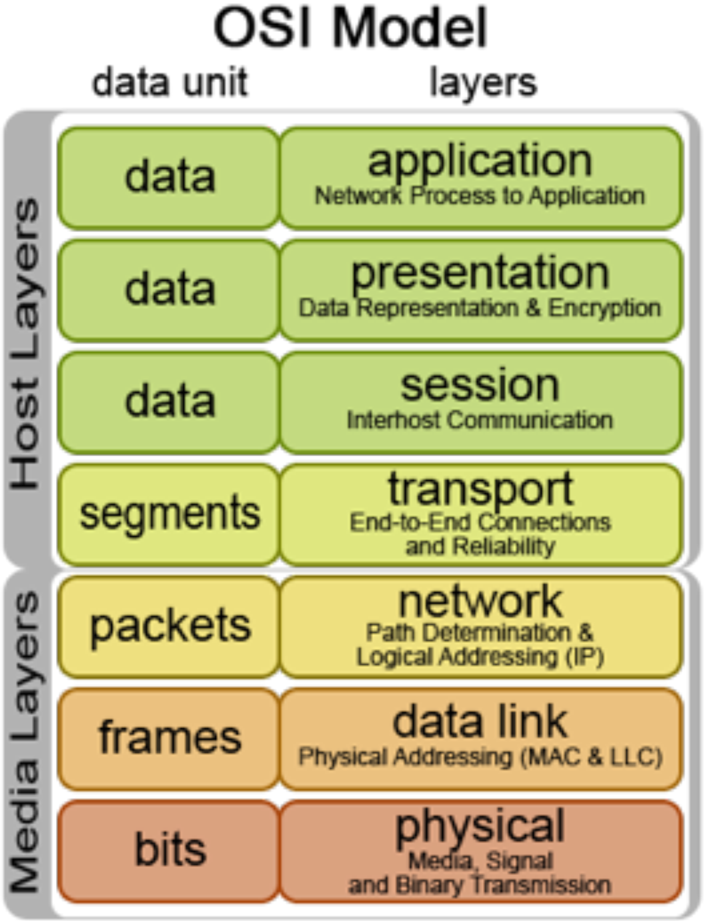
X86 is different
ARM is different
Each SOC is different

Romans 12:2 (NIV)
Do not conform to the pattern of this world

That’s just MIPS.

X86 is different
ARM is different
Each SOC is different

Romans 12:2 (~~NIV~~) NIH
Do not conform to the pattern of this world


OSI Model
data unit
layers
Host Layers
data
application
Network Process to Application
data
presentation
Data Representation & Encryption
data
session
Interhost Communication
segments
transport
End-to-End Connections and Reliability
Media Layers
packets
network
Path Determination & Logical Addressing (IP)
frames
data link
Physical Addressing (MAC & LLC)
bits
physical
Media, Signal and Binary Transmission

JTAG Model

--- (no one uses this crap)

--- N/A - sessionless...

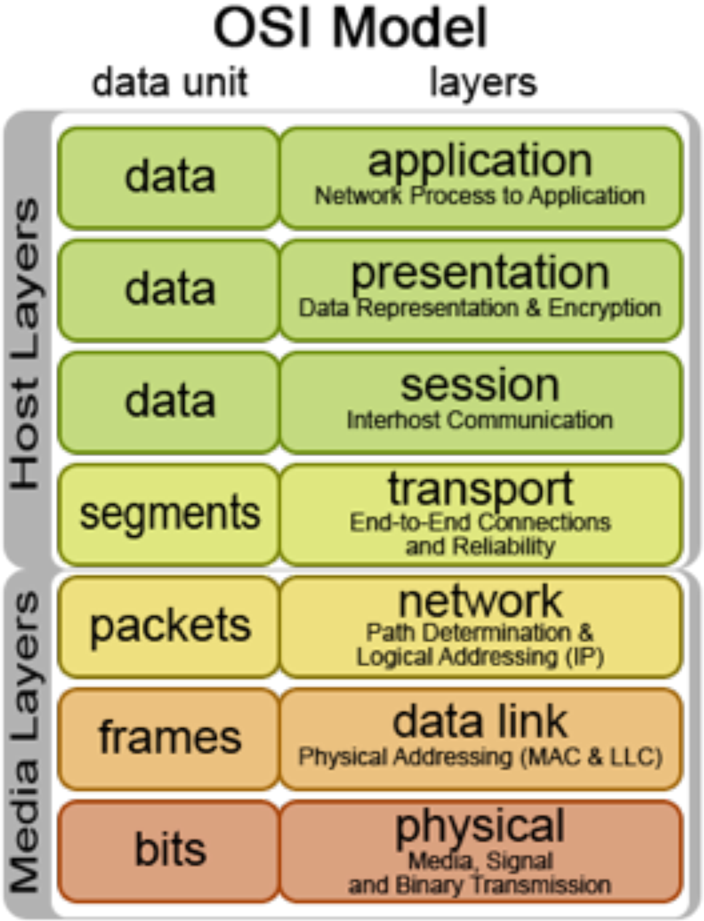
Target-specific configuration

IR/DR access

TAP FSM

TDI, TDO, TMS, TCK, TRST

A Reading from The second email from Joe to people with JTAG questions


OSI Model
data unit
layers
Host Layers
Media Layers
data
application
Network Process to Application
data
presentation
Data Representation & Encryption
data
session
Interhost Communication
segments
transport
End-to-End Connections and Reliability
packets
network
Path Determination & Logical Addressing (IP)
frames
data link
Physical Addressing (MAC & LLC)
bits
physical
Media, Signal and Binary Transmission

JTAG Model

Boundary Scan, Run Control, Memory Access

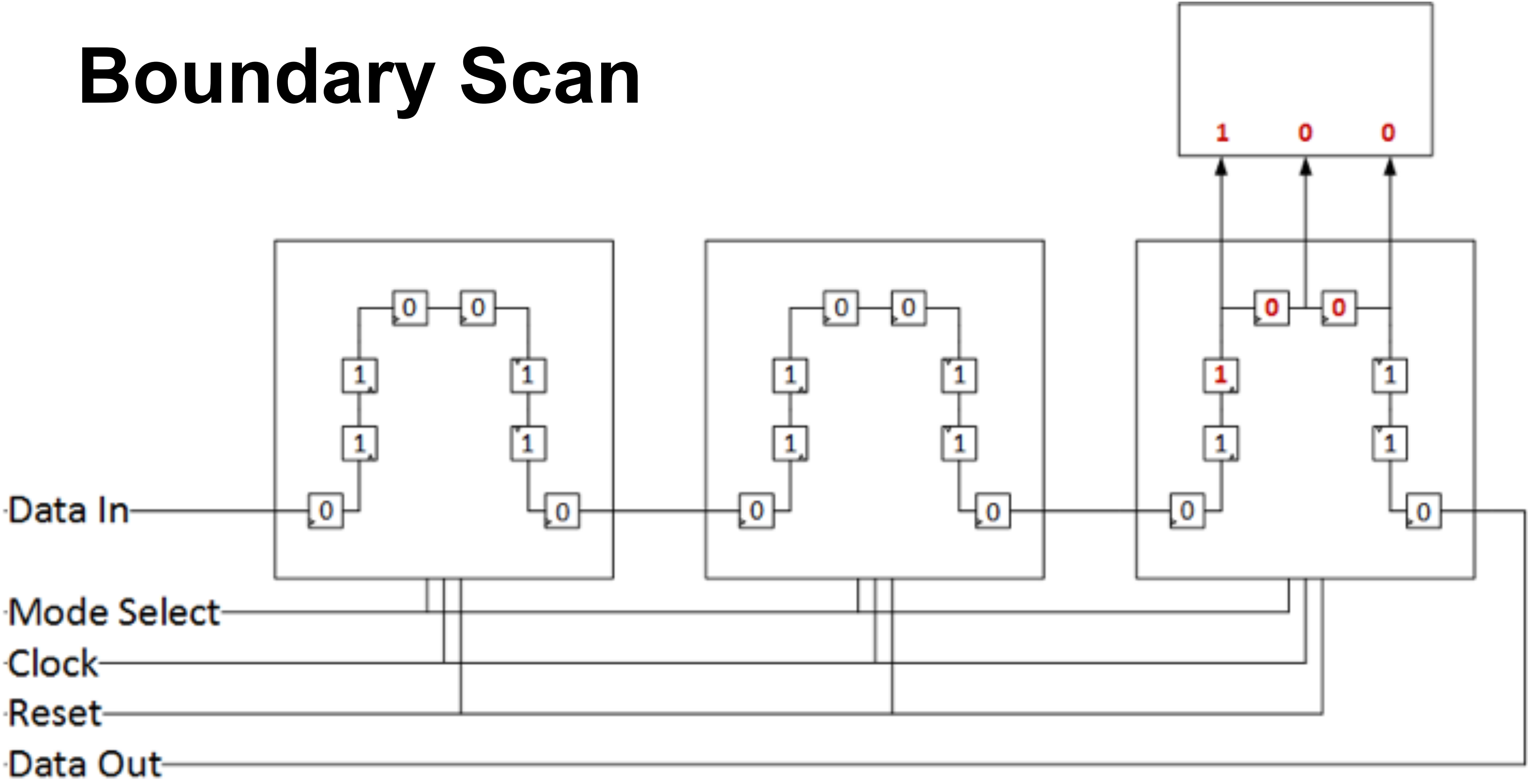
Target-specific configuration

IR/DR access

TAP FSM

TDI, TDO, TMS, TCK, TRST

Boundary Scan



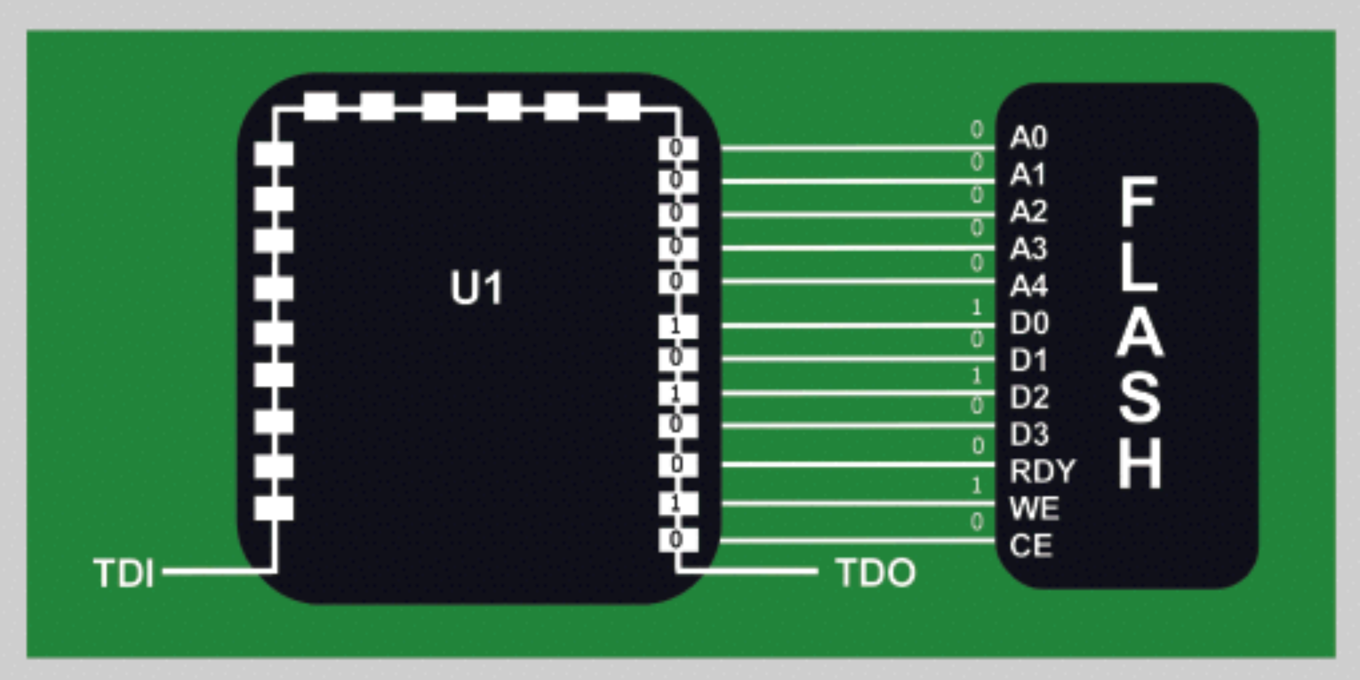


image from intelletech.com, they make stuff to read flash like this

Run Control

~~Run~~ Stop Control

The Debugger’s Gospel

Homily

1149.1 Section 8.3: Private Instructions

c) If private instructions are utilized in a component, the vendor shall clearly identify any instruction binary codes that, if selected, would cause hazardous operation of the component.

JTAG
https://securinghardware.com
intel
Galileo
GEN 2

Liturgy of the PCB

SAVIORBURST Payload

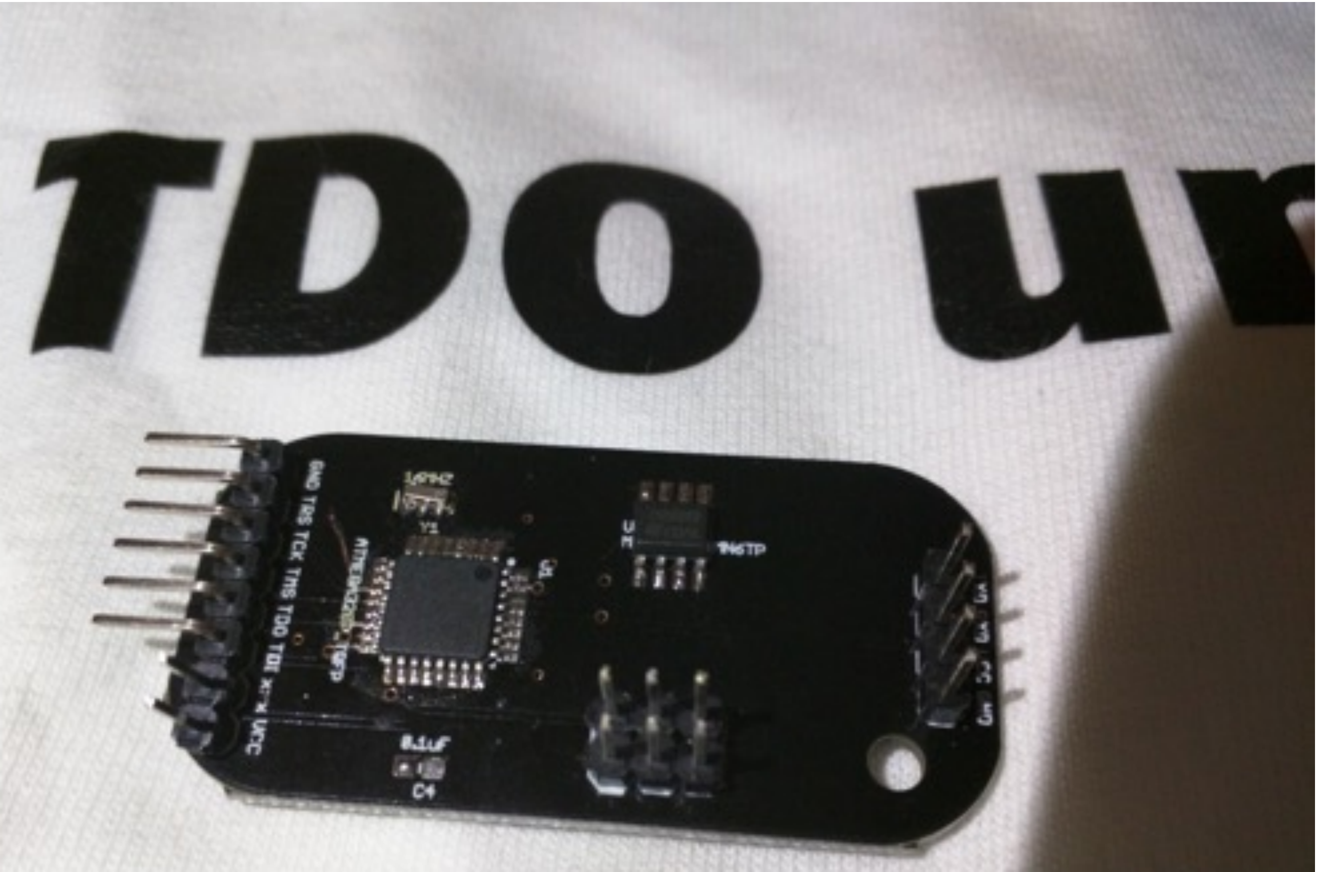
Replay of debug performed in OpenOCD

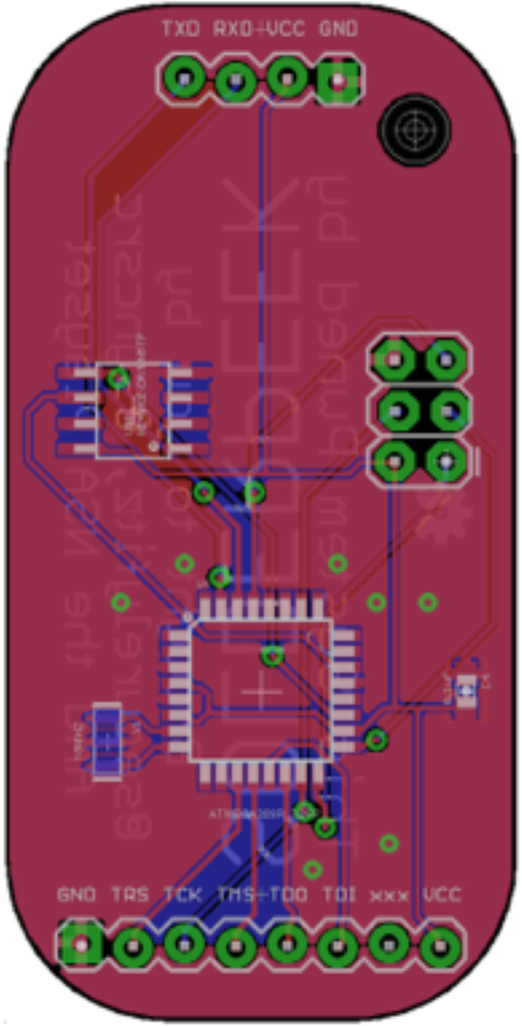
- Target (potentially kernel) specific

Commands are converted into a standard format (SVF/XSVF)

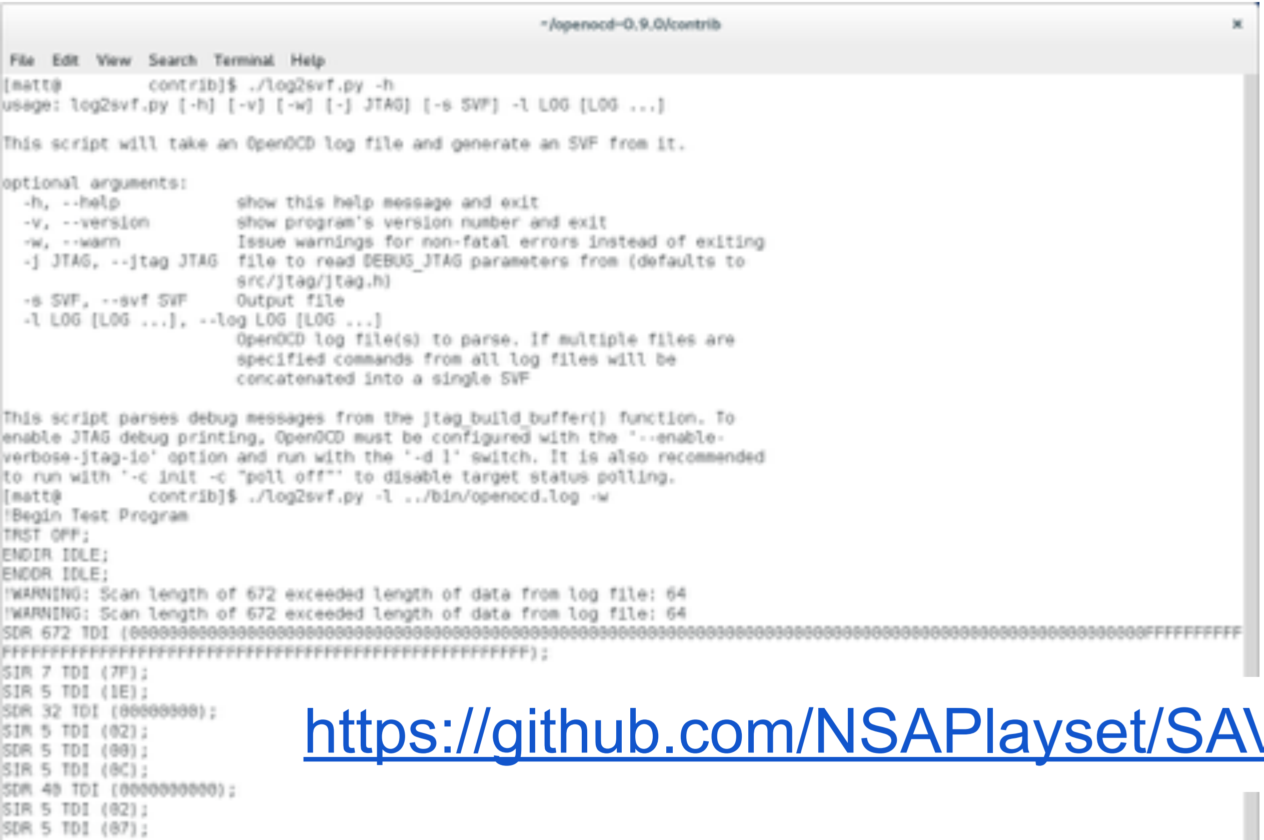
```
!Begin Test Progr
TRST OFF;
ENDIR IDLE;
ENDDR IDLE;
HIR 8 TDI (00);
HDR 16 TDI (FFFF
TIR 16 TDI (0000
TDR 8 TDI (12);
SIR 8 TDI (41);
SDR 32 TDI (ABCD
STATE DRPAUSE;
RUNTEST 100 TCK
```

SOLDERPEEK Implant





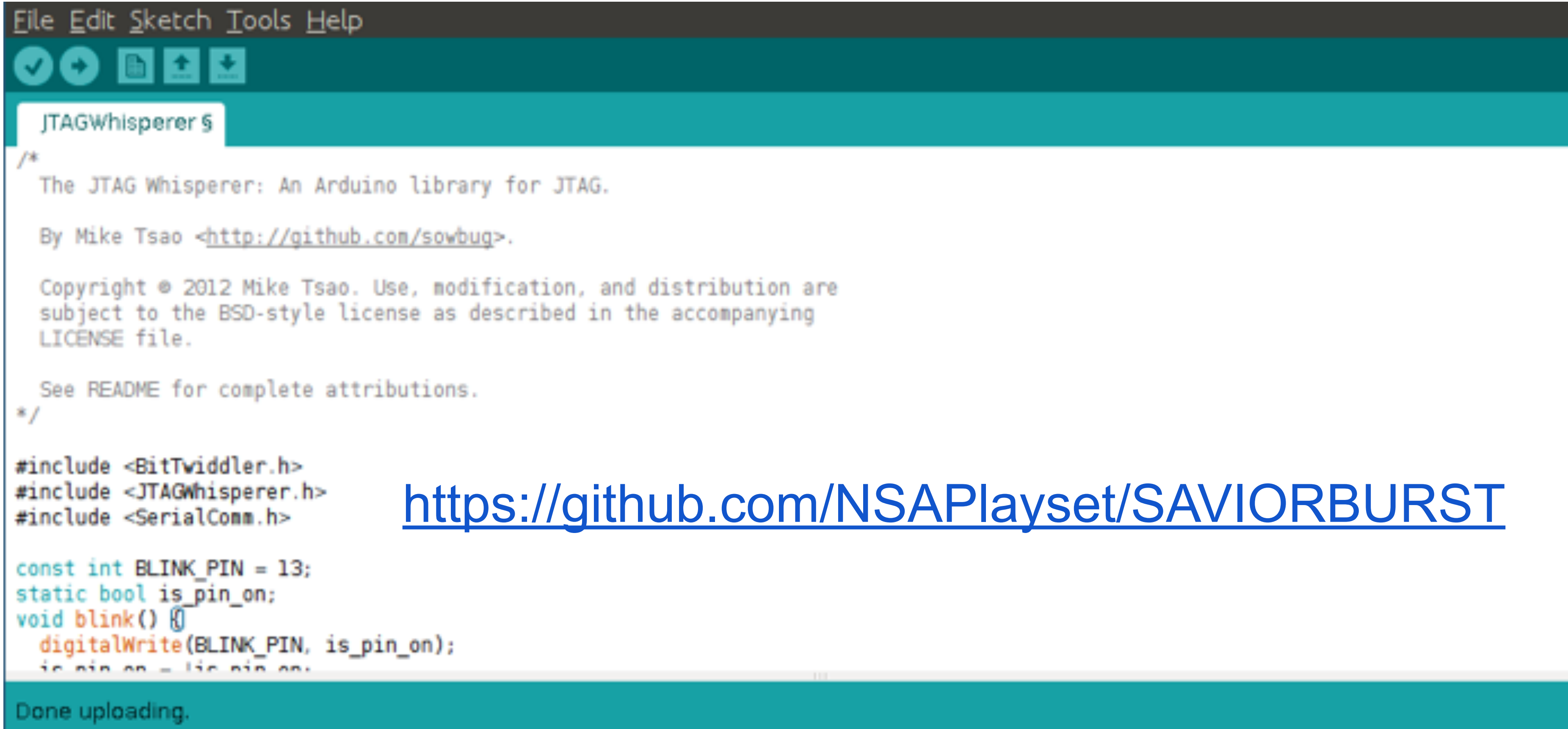
Transubstantiation



```
[matt@          contrib]$ ./log2svf.py -h
usage: log2svf.py [-h] [-v] [-w] [-j JTAG] [-s SVF] -l LOG [LOG ...]

This script will take an OpenOCD log file and generate an SVF from it.

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -w, --warn            Issue warnings for non-fatal errors instead of exiting
  -j JTAG, --jtag JTAG  file to read DEBUG_JTAG parameters from (defaults to
                        src/jtag/jtag.h)
  -s SVF, --svf SVF     Output file
  -l LOG [LOG ...], --log LOG [LOG ...]
                        OpenOCD log file(s) to parse. If multiple files are
                        specified commands from all log files will be
                        concatenated into a single SVF

This script parses debug messages from the jtag_build_buffer() function. To
enable JTAG debug printing, OpenOCD must be configured with the '--enable-
verbose-jtag-io' option and run with the '-d 1' switch. It is also recommended
to run with '-c init -c "poll off"' to disable target status polling.
[matt@          contrib]$ ./log2svf.py -l ../bin/openocd.log -w
!Begin Test Program
TRST OFF;
ENDIR IDLE;
ENDDR IDLE;
!WARNING: Scan length of 672 exceeded length of data from log file: 64
!WARNING: Scan length of 672 exceeded length of data from log file: 64
SDR 672 TDI (00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000FFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF);
SIR 7 TDI (7F);
SIR 5 TDI (1E);
SDR 32 TDI (00000000);
SIR 5 TDI (02);
SDR 5 TDI (00);
SIR 5 TDI (0C);
SDR 40 TDI (0000000000);
SIR 5 TDI (02);
SDR 5 TDI (07);
```

https://github.com/NSAPlayset/SAVIORBURST

Transubstantiation

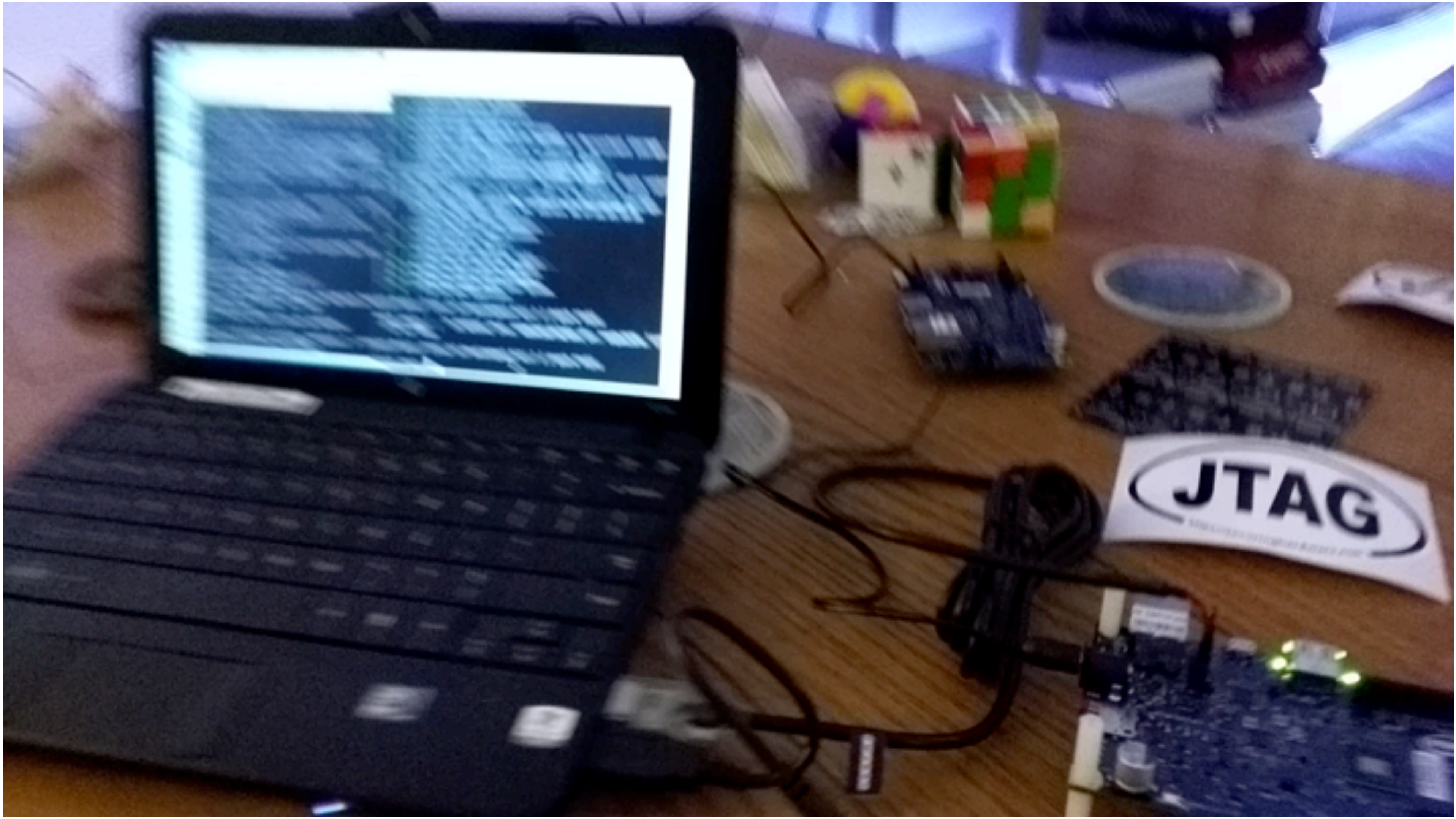


```
/*
  The JTAG Whisperer: An Arduino library for JTAG.

  By Mike Tsao <http://github.com/sowbug>.

  Copyright © 2012 Mike Tsao. Use, modification, and distribution are
  subject to the BSD-style license as described in the accompanying
  LICENSE file.

  See README for complete attributions.
*/

#include <BitTwiddler.h>
#include <JTAGWhisperer.h>
#include <SerialComm.h>

const int BLINK_PIN = 13;
static bool is_pin_on;
void blink() {
  digitalWrite(BLINK_PIN, is_pin_on);
```

Done uploading.

https://github.com/NSAPlayset/SAVIORBURST

Communion


JTAG

Concluding Rites

Solemn Invocation

Not all devices can rely on physical security

Protecting user data requires user control over hardware debug capabilities

Dismissal

I don't want to talk to you no more, you empty-headed animal food trough wiper! I fart in your general direction! Your mother was a hamster and your father smelt of elderberries!

Q & A