JTAG TECHNOLOGIES®

IN-SYSTEM DEVICE PROGRAMMING GUIDE

- fast and convenient
- program flash & µprocessors
- configure PLDs & FPGAs

www.jtag.com


JTAG
TECHNOLOGIES®
We are boundary-scan.®

PREFACE

JTAG/Boundary-Scan Technology for PCB Testing and In-System Configuration is an essential technique widely used in the production of electronic assemblies in the 21st century.

This guide details the benefits of in-system (device) programming via JTAG/boundary-scan and investigates also how it operates within various device types such as microprocessors & DPSs, programmable logic devices and also flash memories.

May 2017

CONTENTS

INTRODUCTION

The popular JTAG/boundary-scan test and programming interface was first introduced in the early 90s when the vast majority of parts were programmed 'off board' using either simple bench programmers or more highly automated production programmers. At this time device programming as a service was also a booming market in its own right.

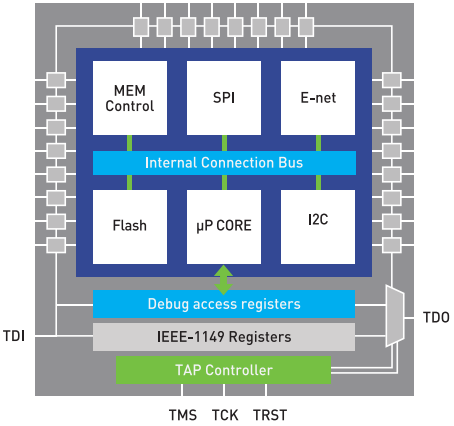
With the advent of JTAG-programmable devices, focus switched to 'In-System', aka 'In-Circuit' or 'On-board', programming of devices (ISP). ISP offers several advantages in terms of the reduced handling of parts (leading to less likely mechanical or static damage), easier field updates, and more flexible production processes (specific code can be provided at assembly time leading to a lower inventory of pre-programmed parts).

With ISP established as standard working practice, JTAG/boundary-scan test companies such as JTAG Technologies set-out to supply a wide range of programming solutions to complement their testing products in manufacturing and streamline the board production process.

DEVICES WITH EMBEDDED MEMORIES

1

Many microprocessors, microcontrollers and systems-on-chip (SoCs) feature embedded flash memory to store both boot code and applications. The most common method of programming this memory is via the JTAG interface. In most cases this means using the micro's debug feature to take over the MCU core allowing fast writing to memory. However, since each device family uses a unique internal structure (bus system, memory controller etc), automating the programming set-up can be difficult.



JTAG Technologies can supply a wide range of programming support options for micros with embedded flash not only through classic JTAG (IEEE std 1149.1) but also through other debug interfaces such as SWD (single-wire debug) and BDM(background debug mode).
A facility is also available to configure the applications to fit within the JTAG topology of your design, thus given a number of daisy-chained micros you can target each with specific code.

A summary of currently available device support options can be found in appendix A to this document. However, due to the dynamic nature of semiconductor developments, new products may not be listed on printed publications. Please visit our web-site for the latest overview or to request a bespoke solution.

2 FLASH MEMORY (NOR, NAND AND SERIAL)

While some devices such as microprocessors, CPLDs and FPGAs can be directly programmed by the JTAG interface, flash memory devices do not typically include any JTAG/IEEE 1149.1 capability or provision for in system programming (ISP). Instead programming is achieved by accessing the signal pins of the device (address, data and control lines) in order to create memory writes and reads and so issue programming instructions and data.

This method of programming the device can be considered indirect since it is a secondary JTAG-compliant device that accesses the flash pins via a boundary-scan register. Traditional parallel NOR flash, NAND flash or serial flash devices can all be programmed indirectly, but there are several variations of the technique that can speed-up the process and reduce programming times.

2.1) Techniques that affect programming times

a) Using AutoWrite™ - AutoWrite (AW) is a signal deployed in addition to the standard JTAG signals (see figure 2a). When programming flash via JTAG/boundary-scan there can be a significant overhead involved just in producing a WE_ pulse. In a standard set-up, a write cycle can be initiated by shifting a data-stream with address information and valid data in a pattern that also holds the flash WE_ line high. The same address and data is shifted again with WE_ low, and a third time with WE_ high again. A JTAG controller that includes AW produces a supplementary WE_ pulse and reduces the boundary-scan shifts per write cycle from three to just one.

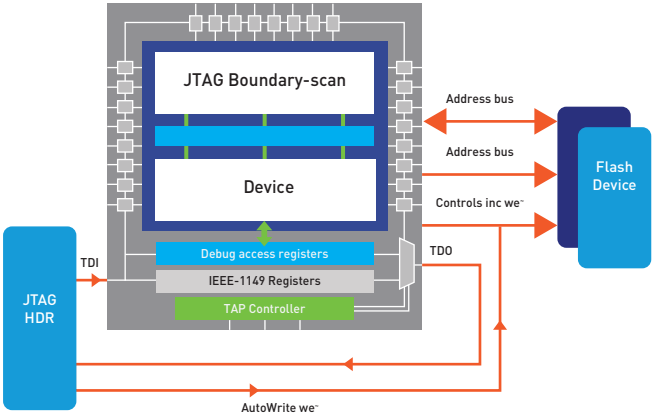


Figure 2a) Flash Programming via JTAG and use of optional AutoWrite

b) Using a shortened chain - shortening the boundary-scan register chain by reducing the number of shift register 'bits' will increase the data throughput. Chains can be shortened by bypassing devices not required to access the flash, and also by implementing an alternative

'pseudo' boundary-scan register in a programmable logic part. This system is often used when programming configuration PROMs for Field Programmable Gate Arrays (FPGAs) e.g. Altera's Active Serial mode (see JTAG Technologies Application Note 21 for details).

c) Using the core - In some devices, such as microprocessors or microcontrollers, it is possible to harness the power of the embedded core to program external flash. As most cores feature a debug mode that is accessed via the JTAG port this can be used to access the Write State Machines (WSMs) of embedded device memory controllers which in turn access external flash devices. It is possible to perform writes at full system speed using this technique, leading to extremely high programming speeds.

d) Adding special logic to FPGA to minimize data transfer between controller and target eg autoaddress increment.

Similarly, JTAG Technologies also offers a system that can program a translator core into FPGAs. The translator acts as a bridge between the JTAG interface and an internal bus (e.g. CoreConnect) within the FPGA which connects to a high-speed memory controller. Accessing the memory controller through JTAG, the translator can enable high-speed flash programming via the FPGA. The following two examples show the speed increases possible using the FPGA translator and embedded programming logic that can be a temporary or permanent part of the FPGA configuration.

A) Altera Cyclone II with JTAG Translator IP

- Device EP2C35F672
- Bscan register length 1449 bits
- Config time for set-up - 4 sec
- TCK rate 10 MHz.
- Target - SPI ROM EPCS64

B) Xilinx Kintex 7 with JTAG Translator

- Device XC7K410T
- Bscan register length 1649 bits
- Config time for set-up – 30 sec
- TCK rate 10 MHz
- Target – QSPI Flash S25FL128

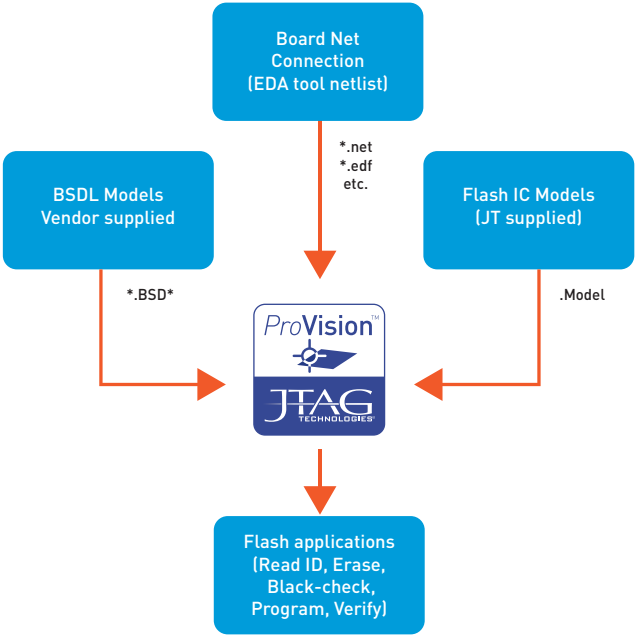
Time for PCS64 - 8MByte	Using Bscan Register	Using JTAG translator
Erase	58s	58s
Write	20288s	120s
Verify	29504s	226s

Time for S25FL128 – 16 Mbyte	Using Bscan Register	Using JTAG translator
Erase	58s	58s
Write	12724s	44s
Verify	17098s	35s

2.2) Software solutions for flash programming application

To set up flash memory programming using JTAG-compliant devices and their boundary-scan registers to perform memory writes, you need the following information i) a model detailing how the boundary-scan device works; ii) a model detailing how the flash device works and iii) design data showing how the two devices are connected,

In an automated application generator, such as JTAG ProVision, **i)** is provided by the programming device's Boundary-Scan Description Language (BDSL) model, part **ii)** will be part of the developer tool's library of programmable devices. In the case of JTAG Provision this will be a, device_name, .model file) and part **iii)** will be a netlist export, typically from an EDA schematic entry or layout system. ProVision models currently support over 3,000 flash devices including serial, I2C, Serial Peripheral Interface (SPI), parallel and NAND types. As well as offering rapid development, the automatically generated applications can be compiled into an optimized format that is executed directly on the controller hardware and provides ultra-fast programming of devices.



SPECIALIST PARTS (PMBUS, MDOC)

3

In addition to programmable logic parts, discrete flash memories and micros/SoCs, with embedded memory, there are a number of other devices that benefit from ISP.

Power management devices are becoming more prevalent on multi-rail designs that might also specify power-up sequences and shutdown modes. Often these parts are programmed by the proprietary Power Management Bus (PMBus), which is based on I2C. JTAG Technologies offers PMBusProg, which harnesses the JTAG capability of a device to mimic the bus transfers and program the parts. Other specialist devices include block flash construct memories such as Disk-On-Chip from M-systems (MDoC).

4 CPLDs AND FPGAs

Although early devices used a variety of proprietary interfaces, by the early 1990s IEEE 1149.1 JTAG emerged as the interface of choice for configuring non-volatile CPLDs (Complex Programmable Logic Devices). However, while IEEE 1149.1 defined the hardware interface widely used for ISP, there was no consensus among device manufacturers for a unified set of data formats or programming instructions. The CPLD vendor's design tools would export a programming data file in Serial Vector Format (SVF), Xilinx Serial Vector Format (XSVF), JEDEC or Virtual Machine (VM) format that would only work with basic PC+ JTAG programmer hardware.

Generic JTAG test and programmer tools suppliers, such as JTAG Technologies, needed to develop support options that would parse these outputs and create a secondary format compatible with their system. Later formats, such as Altera's JAM, and its spin-off, Standard Test And Programming Language (STAPL), received accreditation from JESD (Jedec Standards) and were used by a number of vendors. It was not until 2002 that a new IEEE standard (1532) was ratified, introducing an agreed file format and extended JTAG instruction set dedicated for device configuration. IEEE 1532 now exists as a superset of the base level IEEE 1149.1 and uses the same interface pins, state machine and so on.

One of the main benefits of IEEE 1532 is interoperability. It allows devices from different vendors to be connected in the same JTAG 'chain' and programmed concurrently, using a merged set of data files.

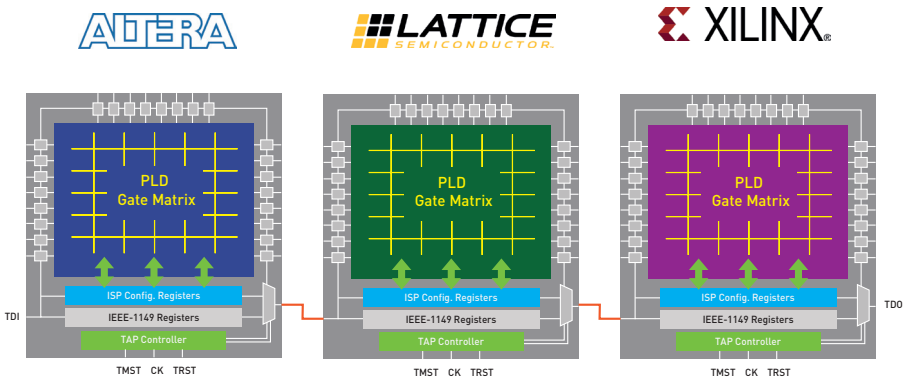


Diagram showing chain of devices from different vendors, programmable using IEEE 1532

Actel	Altera	Lattice	Xilinx
IGLOO	Stratix II- 10	Mach X0-X3	CoolRunnerII
IGLOO	Arria II-10	Mach 4000	XC95xxXL
ProAsic	Cyclone II-V	XP2	Virtex
	Max II	ECP2-5	Kintex
	Max 10	iCE40	Artix
			Spartan
			Virtex_UltraScale
			Kintex_UltraScale

The above devices can be supported by JTAG Technologies tools using SVF, JAM, STAPL or IEEE 1532 (ISC) formats check manufacturer's data for compatible options.

5 DfP (DESIGN FOR PROGRAMMING)

When preparing your design for in-system programmability, there are several design considerations that could increase throughput and improve the convenience during the manufacturing process.

AutoWrite - this feature is provided by JTAG Technologies JTAG/IEEE Std 1149 controller hardware and is used to pulse the we~ line in order to reduce scan register shifts (see section 2.1a). To benefit from AutoWrite (AW), the signal must be incorporated into the board design by connecting back to a JTAG interface header, or made available to a test point. In the latter case, users must ensure that the AW/we~ pad is on the same side of the PCB as the JTAG point signals, to greatly reduce the test fixture complexity.

Access holes for JTAG signals - to allow closed-case on "boxed" programming and re-programming of on-board devices. It can be convenient to allow test pins to probe JTAG signal pads through holes in the case. These could be specially designed or existing ventilation holes.

Access for mode switching JTAG/debug - for devices with dual operating modes such as JTAG/boundary-scan and JTAG/debug, it is important to make a provision to switch between the two modes. This is normally just one signal changing, and can be made by deploying a dedicated cable that either grounds or powers the switch pin to the desired state.

Gang programming - some high-end JTAG systems can support gang program and verification of up to four targets simultaneously. If the target features multiple devices and multiple Test Access Points (TAPs) it may be worth reconsidering the TAP layout, and direct all JTAG devices through a single TAP. For higher target counts, multiple controllers can be operated through a single software interface.

Shortening the chain - another technique to improve data throughput for flash programming is to shorten the scan chain (boundary-scan (shift) register)that is used to access the flash's signal pins (address/data/control) - see also 2.1b). You can shorten a chain by setting any unused (for programming) parts into IEEE 1149.1 BYPASS or HIGHZ modes (HIGHZ is preferred as it bypass a device and tri-states all outputs). Alternatively you can deploy additional parts such as scan buffer swith shorter chains that are used exclusively for on-board programming, however this will add to the BOM cost and may not be desirable. The scheme most often used, if the accessing part is a FPGA, is to program the fabric of the FPGA with an 'artificial' short scan chain that can be used just for the duration of the flash programming stage.

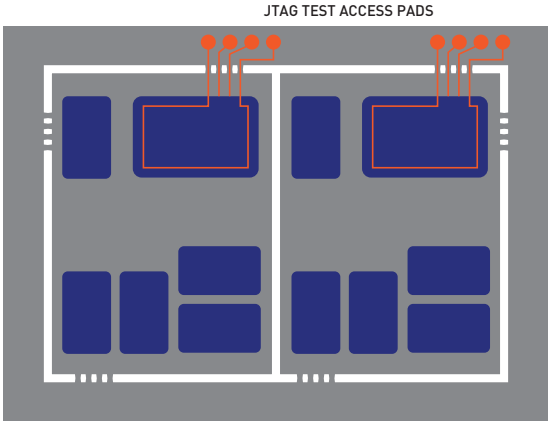
SECURING DATA AND PROGRAMMING

Secure programming protects Intellectual Property (IP) and/or prevents hacking. JTAG programming and re-programming, however, is sometimes seen as a loophole in this process. Although it is relatively simple to read and modify contents of flash memory, it is not a trivial procedure to reverse engineer the contents of a PLD. A degree of design data, preferably schematic diagrams together with access to the JTAG signals, is needed to make both operations possible.

A simple way to enable a basic level of security is to disguise or remove JTAG access. Traditionally JTAG-enabled designs will feature one or more connectors. Removing the silk-screening from these may deter a hacker for a short while. Removing the connector altogether and/or replacing it with test pads that are only accessible via a spring pin fixture, is better still. A further refinement would be the physical isolation of the JTAG signals, with fusible tracks that can permanently remove access from the PCB, or a 'break-off' section that locates the JTAG test pads for the duration of the manufacturing process only. A further security measure would be to underfill programmed BGA components. In most cases, this will make them impossible to 'lift' and analyse on a device programmer.

A disadvantage of removing the access to JTAG pins/ports is that it might also make it unavailable to field service repair. Therefor a better solution to use embedded security feature in devices.

Embedded security now also features on a number of high-end FPGAs (such as Altera's Stratix, Arria and some Cyclone devices). The security measures are based upon the encryption of data from the FPGA's configuration source which must be loaded each time on power-up. Without encryption, the data-stream could be intercepted or recorded on a logic analyser and reverse engineered.



JTAG signals are routed through the break-off section for security

In the case of Altera FPGAs, a security key can be permanently 'blown' (via efuses) into the device, or stored in non-volatile (or battery-backed) Random Access Memory (RAM). Keys are usually 256-bit and can be produced from an algorithm that imports two 256-bit strings. Both volatile and non-volatile (efuse) keys can be used within a device, with the option set in the configuring data stream. Essentially, most devices feature a tamper protection mode that prevents the FPGA from being loaded with an unencrypted configuration file. With tamper protection enabled, the FPGA can only be loaded with a configuration that has been encrypted with your key. Unencrypted configurations, and configurations encrypted with the wrong key, will not work. Tamper protection is also enabled by setting a fuse within the device.

HARDWARE AND SOFTWARE SELECTION GUIDES

7

Hardware Selection Guide

JTAG Technologies supplies a selection of hardware interfaces that support not only JTAG IEEE 1149, but can, in some cases, be re-configured to support allied interfaces such as BDM and SWD. Lower cost and less sophisticated hardware can still support the majority of programming applications via JTAG, although there will be some compromise in programming speed and versatility. The table below illustrates price-performance of JTAG Technologies hardware.

	Number of TAPs	Speed grade	PLDP Prog	FlashProg	Embedded support	Recon-figurable
JT-Live	1	✓	✓	✓		
JT 3705	2	✓	✓	✓		
JT 5705	2+	✓✓	✓	✓	✓	✓
JT 37x7	4	✓✓✓✓	✓	✓ inc. NAND	✓	✓

Speed grade 1 controllers operate at a max TCK speed of 6MHz and their throughput (mean programming speed) is also governed by the host PC. Speed grade 4 controllers operate an autonomous state machine that allows them to operate at continuous clock speed of up to 40MHz.'

JT37x7 QuadPod

JT 5705/USB

JT 3705/USB

Software Selection Guide

JTAG Technologies offers two software options for device programming: JTAGLive and JTAG ProVision.

	PLDs via SVF	PLDs via JAM & STPL	PLDs via IEEE 1532	Flash (NOR/ Serial)	Embedded support (µPs etc.)	NAND Flash
Studio	✓	✓		✓*		
ProVision Flash				✓	✓**	✓
Provision PLD	✓	✓	✓			

* *Programming applications available as Python module examples*

** *Most µP support options are ready to run applications in an optimised format for the JT 37x7 series*

CONTACT INFORMATION

8

For more information

If you want to apply boundary-scan for testing or in-system programming, and need more help, or need product information, please contact:

JTAG Technologies' Sales and Customer Support Offices

To contact JTAG Technologies' local sales representatives, visit
www.jtag.com/en/About/How_to_contact_us

Europe and ROW

T +31 (0) 40-2950870
F +31 (0) 40-2468471
E info@jtag.nl

United Kingdom & Ireland

T +44 (0) 1234-831212
F +44 (0) 1234-831616
E sales@jtag.co.uk

USA, Canada and Mexico

T (Toll Free) 877-FOR-JTAG F: 410-604-2109
E info@jtag.com

China (including Malaysia, Singapore, Taiwan, Thailand)

T +86 (021) 5831-1577
F +86 (021) 5831-2167
E info@jtag.com.cn

IEEE Standards

- IEEE Std 1149.1-2001 - IEEE Standard Test Access Port and Boundary-Scan Architecture (Supersedes former issues IEEE 1149.1-1990 (Including 1149.1a-1993) and IEEE 1149.1b-1994 and errata)
- IEEE Std 1532-2001 - IEEE Standard for In-System Configuration of Programma- ble Devices (Supersedes IEEE 1532-2000)

For more information on the IEEE Standards

IEEE Customer Service, 445 Hoes Lane, PO Box 1331 Piscataway NJ 08855-1331 USA

T (800) 701 4333 (within the US and Canada) F: (732) 981 9667
T (732) 981 0060 (outside the US and Canada) E: customer.service@ieee.org
W www.ieee.org

9 APPENDIX

Manufacturer	Device Family	Option Name
Analog Devices	Blackfin	-
	ADuC7xxx	ADuC7xxxProg
	Blackfin	ADSP-BF538Prog
	Blackfin	ADSP-BF539Prog
	Blackfin	ADSP-BF51xProg
ATMEL	AT91SAM7	-
	AT91SAM7SEProg	-
	ATMega64	-
	ATMega8	-
	ATtiny	-
Cypress	Psoc3	PSoc3Prog
	Traveo	TraveoProg
Freescale	Coldfire	MCF52xxx -
	Qorivva	MPC55xx MPC5500Prog
	Qorivva	MPC56xx MPC5600Prog
	MPC5xx	MPC500Prog
	HC08	HC08Prog
	HCS08	HCS08Prog
	HCS12	HCS12Prog
	Kinetis	KinetisProg
	MC56F8000	MC56F8000Prog
Infineon	XC166	XC16xProg
	XE166	XC16xProg
	XC27xx	XC16xProg
	XC23xx	XC16xProg
	XC22xx	XC16xProg
Microchip	PIC32MX	PIC32MXProg
	dsPIC33	-
	PIC 10F*	-
	PIC 12F*	-
	PIC 16F*	-
	PIC 18F*	-
Nordic	NRF51822	NRF51822Prog

NXP	LPC2xxx	LPC2xxxProg
	LPC17xx	LPC17xxProg
	LPC12xx	LPC12xxProg
	SJA2020	SJA2020Prog
Philips	SAA56xx	SAAProg
	TDA95xx	SAAProg
Renesas	SH7K	SH7KProg
ST	DSM 2xxx	PSDProg
	PSD 4xxx	PSDProg
	PSD 8xx	PSDProg
	PSD 9xx	PSDProg
	SMM 1xxx	PSDProg
	uPSD3200 PSD	Prog uPSD3300
	PSDProg uPSD	3400 PSDProg
	SPC560x	SPC560xProg
	STM32F1	STM32F10Prog
	STR91xFxxx	STR91XProg
	STM32F3	STM32F30Prog
	STM32F4	STM32F4Prog
	STM32L05	STM32L05Prog
SiliconLabs	C8051	8051Prog
	SiM3C1xx	SiM3Prog
	SiM3U1xx	SiM3Prog
TI	MSP430F1xx	MSP430Prog
	MSP430F2xx	MSP430Prog
	MSP430F4xx	MSP430Prog
	MSP430FE4xx	MSP430Prog
	MSP430F5xxx	MSP430Prog
	MSP430FR5xxx	MSP430Prog
	MSP430F6xxx	MSP430Prog
	MSP430G2xxx	MSP430Prog
	CC430F5xxx	MSP430Prog
	CC430F6xxx	MSP430Prog
	Stellaris LM3Sxxxx	StellarisProg
	TMS320F28xx	TMS320Prog
	UCD9240	UCD9xxxProg
	TMS570	TMS570Prog
	Tiva	TM4C12x

JTAG TECHNOLOGIES®

www.jtag.com