



**Advanced  
Micro  
Devices**

# Introduction to JTAG and Five-Volt Programming with In-circuit Programmable MACH Devices

*Application Note*

---

# Introduction to JTAG and Five-Volt Programming with In-circuit Programmable MACH Devices



Advanced  
Micro  
Devices

## Application Note

An important feature of some MACH devices is: JTAG testing capabilities and five-volt programming. All MACH 3 and 4 devices with more than 84 pins will contain a fully JTAG compliant interface with in-system programming capabilities. Devices which have boundary-scan cells and support all JTAG instructions are called JTAG compliant. These devices currently include the MACH355, MACH445, MACH465 and the entire MACH 5 Family. In addition, MACH Performance Plus devices such as the MACH111SP, MACH211SP, and the MACH231SP are available with five-volt programming and JTAG compatibility. JTAG compatible means that the device may be placed in any JTAG-compliant chain, but only the IDCODE and BYPASS JTAG instructions are available; boundary scan testing is not supported. Table 1 lists the differences between the MACHXX1SP and the MACH 3, 4, 5 families with respect to JTAG and 5 volt in-system programming.

**Table 1. Summary of AMD's MACH ISP Devices**

	MACH 5	MACHXX1SP	MACH 3 & 4
JTAG Compliant	YES	NO, Compatible Only	YES
ENABLE* Pin	NO	NO	YES
TRST* Pin	NO	NO	YES
Boundary Scan Cells	YES	NO	YES
Supports All Mandatory JTAG Instructions	YES	Only BYPASS and IDCODE	YES
Preload/Observe Security	NO	NO	YES
Program/Verify Security	YES	YES	YES
5V Programming	YES	YES	YES

## A BRIEF HISTORY OF JTAG

JTAG is the commonly used acronym for the IEEE Standard 1149.1-1990. The concept for this standard was proposed by a group of European companies known as the Joint European Test Action Group, or JETAG, in 1985. A year later, this group was expanded to include companies from North America and the name was changed to Joint Test Action Group, or JTAG. This group developed a standard so that circuit connectivity could be checked using a boundary-scan register approach. This standard was adopted by the Institute of Electrical and Electronics Engineers, or IEEE, in 1990. An amendment, Supplement "A", was passed in 1993 and included corrections, clarifications, and additions to the original standard.

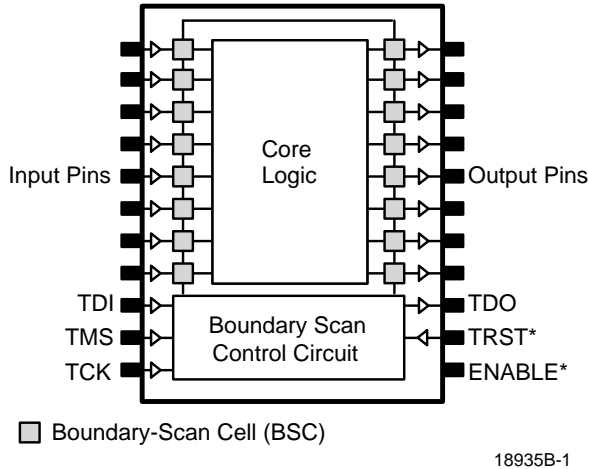
Before the standard for boundary scan was developed, many companies were implementing their own serial or parallel scan testing methods. The JTAG standard was developed to provide both board-level and device-level testing. The JTAG standard defines input and output pins, logic control functions, and instructions. To use JTAG for testing, all that is required is a four- or five-wire interface to accommodate a serial data stream and the software to drive that interface. JTAG allows access and control of each node in each JTAG-compatible integrated circuit (IC), to test board connections and board functionality.

The JTAG approach has many benefits. As IC manufacturers reduce the size and increase the complexity of their devices, testing becomes more difficult with conventional methods. Printed Circuit Board (PCB), traces have decreased in width by a factor of ten and IC package pin sizes have decreased by a factor of eight, increasing the probability of damage to the leads during conventional testing with bulky test probes. The internal array of Boundary-Scan Cells (BSCs) become a virtual "bed-of-nails" multi-point test set-up, making it possible to test PCB connections and circuit logic functionality independently. The structured approach of JTAG enables the testing done while prototyping to be easily transferred to manufacturing on conventional bed-of-nails testers.

## JTAG Boundary - Scan Architecture

Figure 1 illustrates an example of a JTAG compliant device containing the JTAG circuitry. The core logic remains intact, but a separate boundary scan control circuit is added to perform the JTAG functions. The five JTAG pins are used to access the state machine, instruction register, and data registers. These five pins are known as the Test Access Port, or TAP. The TMS

and TCK pins drive the state-machine-based TAP controller. A boundary-scan cell is paired with every important node in the device. These nodes include all inputs, all outputs, and enable control lines. For the JTAG compliant MACH 3 and 4 devices, the ENABLE\* pin is a dedicated non-JTAG programming pin that is utilized for programming security.



**Figure 1. Boundary-Scan Architecture**

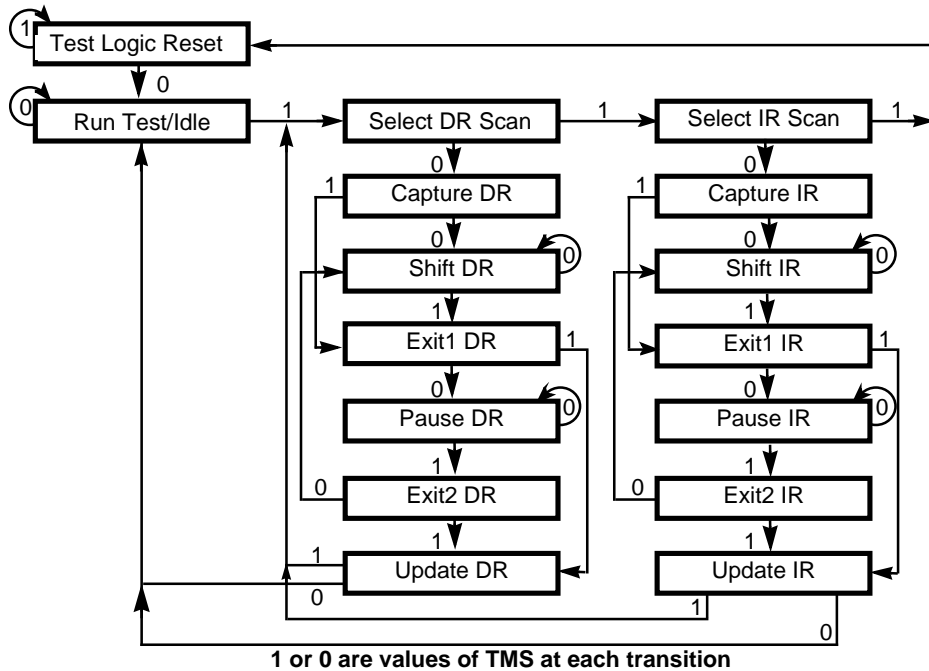
The five JTAG pins and associated functions are listed in Table 2. The ENABLE\* programming pin is included in the definitions, but is not considered part of the JTAG TAP. Details on programming appear later in this application note. The TRST\* and ENABLE\* pins are active LOW inputs as denoted by the asterisk.

**Table 2. TAP Pin Descriptions**

TCK (Test Clock)	This pin controls the state machine and data transfer operations
TMS (Test Mode Select)	Selects the boundary scan test mode, which controls the state machine test operations
TDI (Test Data Input)	Receives serial instruction codes and data on the rising edge of the edge of the TCK signal
TDO (Test Data Output)	Shifts serial output data on the falling edge of the TCK signal
TRST* (Test Reset)	Optional JTAG pin used to reset the state machine
ENABLE* (Program)	Non-JTAG pin used for programming some MACH devices

**JTAG TAP Controller**

All implementations of JTAG are required to contain three key elements for the TAP controller. These elements include the state machine, the instruction register, and the data registers. The first requirement is the synchronous finite state machine which controls the function of the various JTAG registers. The state machine determines whether the device is in reset mode, receiving an instruction, receiving data, or idle. The state machine, as illustrated in Figure 2, is completely controlled by the TCK, TMS, and TRST\* pins. The value of TMS on the edge of TCK is located next to each transition in Figure 2.



**Figure 2. TAP Controller State Machine**

The TAP controller shifts and updates the individual JTAG registers. The data enters on the TDI pin and exits through the TDO pin. After the data is shifted into place, the TAP controller updates the register to make the data current. As stated earlier, for a JTAG compliant MACH device there are boundary-scan cells on all of the inputs into the core logic and on all of the outputs from the core logic. The TAP controller manipulates both the input and the output BSCs. This allows the capability to drive inputs and capture outputs or vice versa. The pause states are included so that the shifting of data can be temporarily stopped.

When power is applied to the device, the TAP controller is forced into the Test Logic Reset state and the ID-CODE register is initialized. Note that from any state position in the state machine, five consecutive ones on the TMS pin will reset the state machine without the use of the optional TRST\* pin. If standard four-pin JTAG is desired, tie the TRST\* pin high. To disable the JTAG circuitry entirely, tie the TRST\* pin low. Note that the MACHXX1SP and MACH 5 devices do not have the TRST\* pin nor the ENABLE\* pin.

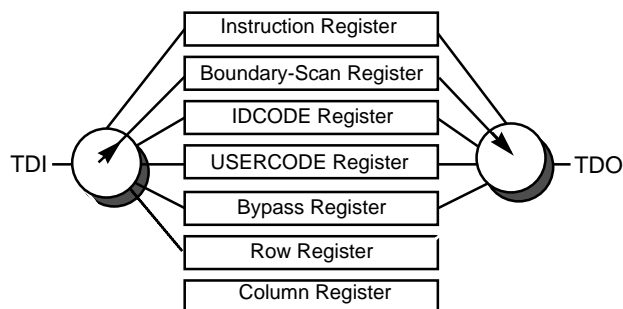
The second required element for a TAP controller is an instruction register (IR) which holds the instruction word. The instruction word length in a MACH device is six bits. The instruction decoder interprets the instruction word held in the IR. The instruction in the instruction register dictates which JTAG register acts as the data register being shifted and updated.

The third element is the data registers (DR) which could be the boundary-scan register, the IDCODE register, the USERCODE register, the row register, the column register, or the bypass register, depending on the individual instruction. Please note, the MACHXX1SP devices only support the IDCODE and BYPASS register modes.

The registers illustrated in Figure 3 are part of the JTAG circuitry, with the exception of the row and column registers. The MACH device instruction register, as defined earlier, holds the six-bit instruction words. The Boundary-Scan Register is a term for the sum of all of the BSCs, so the size of this register varies with the size of the device. The IDCODE register is an optional register that contains a 32-bit word with three components: the manufacturer identification, the device identification, and the revision number. The USERCODE register is a user-specified, 32-bit word. The bypass register is a single shift register stage which provides a serial path between the TDI and TDO pins.

The row and column registers implemented by AMD are not actually part of the JTAG standard but have been included here for educational purposes. The row register corresponds to a particular row within the fuse array of a device. The column register corresponds to

the number of bits that will be sent via the TDI pin to program the appropriate row.



18935B-3

**Figure 3. JTAG Instruction and Data Registers**

## JTAG Testing Procedure

The actual JTAG testing procedure will not be addressed in this application note. The purpose of this note is to introduce the concept of JTAG and the various components that comprise the feature. AMD does not provide software support for JTAG testing, however, there are a number of third-party vendors that specialize in JTAG testing hardware and software packages. These vendors include bed-of-nails manufactures such as Hewlett-Packard, GenRad and Teradyne, as well as exclusive JTAG testers manufactured by Texas Instruments and Corelis. A listing of presently available JTAG support contacts is provided at the end of this document.

## FIVE-VOLT PROGRAMMING

Another benefit from the JTAG circuitry that AMD has derived is the ability to use the JTAG TAP for five-volt programming. This allows the device to be soldered to the board before programming. Once the device is attached, the delicate Plastic Quad Flat Pack, or PQFP, leads are protected from programming and testing operations that could potentially damage them. Programming and verification of the device is done serially which is ideal for on-board programming.

The ENABLE\* pin is used for extra programming security and is only found in the MACH355, 445, and 465. If desired, the ENABLE\* pin can be connected to ground, and the device programmed with only the TAP. If TRST\* is not used in the system, it too, can be removed by connecting it to  $V_{CC}$ .

Programming is the process where MACH devices are loaded with a pattern defined in a JEDEC file obtained from MACHXL software or third-party software. Programming the MACH device after it has been placed on a circuit board is easily accomplished. Programming is initiated by placing the device into programming mode, using the MACHPRO programming

software provided by AMD. The device is bulk erased and the JEDEC file is then loaded. After the data is transferred into the device, the PROGRAM instruction is loaded and the ENABLE\* pin is pulsed low if it is used to complete the programming sequence. Further programming details can be found in application note, “Advanced In-circuit Programming Guidelines.”

## On-Board Programming Options

Since the MACHPRO software performs these steps automatically, the following programming options are published for reference.

The configuration file, which is also known as the chain file, defines the MACH device scan path. The file contains the information concerning which JEDEC file is to be placed into which device, the state which the outputs should be placed, and whether the security fuses should be programmed. The configuration file is discussed in detail in the MACHPRO software manual.

For the MACH 3, 4, and 5, the state of the I/Os during programming can be controlled by preloading the boundary-scan cells with a known state or by disabling the Output Enable for each I/O, placing them in a high-impedance state. This preload value is obtained from the configuration file.

The MACHXX1SP devices tristate the outputs during programming. There are two optional security bits which can be used in all MACH 3 and 4 devices. The first one is the program and verify security bit. Once this bit has been programmed, all of the programming and verification options are disabled until the device is erased. Programming the second bit also prohibits the option to preload and observe the macrocell registers. The In-system Programmable MACH Performance Plus devices and MACH 5 devices do not support register preload and register observe. They have one security bit which inhibits program and verify. This allows the user to protect proprietary patterns and designs.

Program verification of a MACH device involves reading back the programmed pattern and comparing it with the original JEDEC file. The AMD method of program verification performed on the MACH devices permits the verification of only one device at a time.

## Accidental Programming or Erasure Protection

It is virtually impossible to program or erase a MACH device inadvertently. The following conditions must be met before programming actually takes place:

- The device must be in the password protected program mode
- The programming or bulk erase instruction must be in the instruction register
- For the MACH 3 and 4 devices, the ENABLE\* pin must be low

If the above conditions are not met, the programming circuitry cannot be activated. Even if the device is in program mode with a programming instruction in the register, an internal pull-up resistor on the ENABLE\* pin prevents any accidental pulses from occurring.

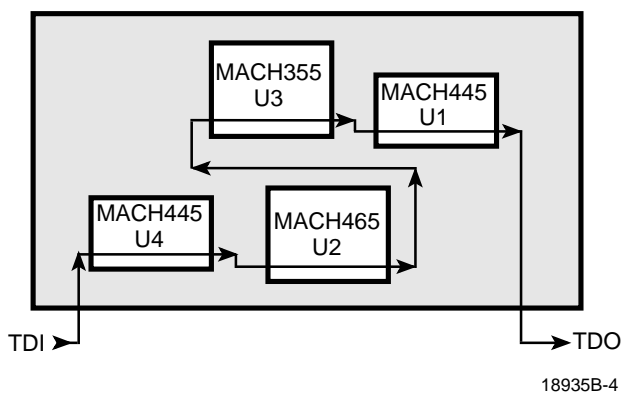
To ensure that the AMD ten year device data retention guarantee applies, 100 program/erase cycle limit should not be exceeded.

## Programming Multiple MACH Devices

There will often be more than one JTAG-compatible MACH device in a circuit and these devices will be connected to form a scan path. The simplest scan path is when all JTAG-compatible devices have their TMS, TCK and TRST\* pins connected in parallel. The TDI and TDO pins are connected serially, where the TDO of one device is connected to the TDI of the next device. If the ENABLE\* pins are used, they may be connected in parallel or remain separate for individual MACH device programming access. More complex scan paths could involve multiple TDI and TDO paths or multiple TMS paths. Multiple device programming does not imply multiple device verification since verification is always performed one device at a time. The user specifies which devices in the scan path to program, and the MACHPRO software will program them.

Figure 4 illustrates a map of the scan path which contains each device and its position in the chain. Any device in a scan path which has not been selected for programming must be bypassed for the entire programming and program verification cycle. This is done by loading the appropriate instruction into the device anytime new instructions are loaded. Additionally, the bypass register must be taken into account any time data is being shifted through the path.

The order in which program and program verification is completed is from the first device in the scan path to the last device in the chain.



**Figure 4. JTAG Scan-Path Map**

### Programming Hardware

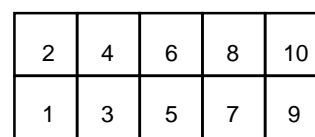
The hardware interface for supporting both the JTAG and five-volt programming features requires four pins (TAP and ENABLE\* pins are optional on MACH 3 and 4). Additional pins to accommodate power and ground, as well as specialized functions such as in-circuit reconfigurability and multiple scan path configurations, may be desirable.

A parallel printer port of any IBM-compatible Personal Computer, or PC, is sufficient to interface with the target board containing the MACH devices. AMD supplies a cable that could be used to adequately program and test the MACH devices.

Some PC systems have software keys connected to the parallel port. Please remove these keys before using the supplied software.

The AMD cable contains the four required interface signals.

The target board cable connector is a 10 (5x2) pin female connector. The connector chosen by AMD is manufactured by Dupont and their connector number is 71602-010. The suggested target board male connector is Dupont number 71918-110 which is a 10-pin header with latches that allow for a quick disconnect. Figure 5 illustrates the cable header connector pinout locations. There is a locator key opposite of pin 5 for safety purposes.

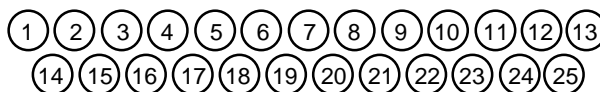


- Pin 1 - TCK
- Pin 2 - NC
- Pin 3 - TMS
- Pin 4 - GND
- Pin 5 - TDI
- Pin 6 - V<sub>CC</sub> or NC
- Pin 7 - TDO
- Pin 8 - GND
- Pin 9 - TRST\*
- Pin 10- ENABLE\*

**Figure 5. Board Header Cable Connector Pin Locations**

The pin locations of the board header were chosen to minimize crosstalk between the wires in the ribbon cable. This header-pin configuration is supported by AMD, but individual applications may require an alternate header-pin arrangement. The AMD cable is six feet in length. If the distance from the user's computer to the target application is more than six feet, proper cable design rules should be followed in order to reliably program the MACH devices.

The connector for the computer end of the cable is a standard DB-25 male connector. The connections are illustrated in Figure 6.



- Pin 1 - TRST\* (STROBEB)
- Pin 2 - TCK (D0)
- Pin 3 - TDI (D1)
- Pin 4 - TMS (D2)
- Pin 11 - TDO (PRINTER\_BUSY)
- Pin 23 - GND
- Pin 17 - ENABLE\* (ACT\_SELB)
- Pin 24 - GND
- Pin 25 - ZCTL (GND)

**Figure 6. Computer Port Cable Connector Pin Locations**

A third-party programmer could also provide programming support for the MACH devices. The programmer also uses the AMD cable which plugs into a 28-pin socket on the programmer instead of the PC parallel port. An assortment of programmer manufacturers are listed at the end of this document.

There are a number of additional programming scenarios that could also be considered. Programming practices used while prototyping could vary significantly from final production methods. A number of bed-of-nails tester vendors, including Hewlett-Packard, GenRad and Teradyne, support MACH device programming. Contact individual vendors listed at the end of this application note for more information.

### Software Support Tools

AMD's MACHXL software includes support for programming MACH devices. This software is an easy to use tool that supports the design entry, fitting, and simulation of MACH device applications. Once the design is completed, it may be downloaded to the MACH device through the programming procedure.

The five-volt programming software in MACHXL software supports multiple MACH devices and allows the user to combine AMD MACH devices with other JTAG devices in a chain. A listing of third-party software support appears at the end of this note. The FusionPLD<sup>SM</sup> Catalog contains a complete list of available third-party software support.

### Programming Procedure Overview

This section provides an overview of a typical programming procedure using the MACHPRO software. This assumes that the MACH devices are already placed on the board and are linked in a serial chain with the other JTAG devices. Additional details of the programming process are available in the MACHXL software documentation as well as in the advanced application note.

Initially, the JEDEC files for the MACH devices to be programmed are generated using MACHXL or other third-party tools. Next, a serial chain description file, listing the JTAG devices in the chain and the actions to be performed on each device, is written. Then, the target board is connected to a power supply and the JTAG programming cable is attached to both the PC parallel port and the board. Finally, after the target board supply is switched on, the MACHPRO software with the associated chain file is used to program the MACH devices.

Please refer to the MACHPRO software manual for more information on writing a chain description file.

### Boundary Scan Description Language File

The Boundary Scan Description Language, or BSDL, file describes the pinout of an IC. The file also describes the instruction codes and layout of the boundary-scan cells. The file does not describe the JTAG scan path. A generic BSDL file is supplied for every device type.

The BSDL file declares TAP pin locations, instruction codes, register length/structure, device ID code, and whether there is a TRST\* pin. To test a JTAG board, a collection of all of the BSDL files for the JTAG ICs is required along with a net list describing how these ICs are connected.

Additional sources of information on BSDL files are listed in the Suggested Reading section.

### SUGGESTED READING

*IEEE Standard 1149.1-1990*

*IEEE Standard 1149.1-1990 Supplement A*

*The Boundary-Scan Handbook* by Kenneth Parker

*Meeting the Challenge of Boundary Scan* by GenRad

## Preliminary JTAG and/or Programming Support for MACH Devices

Manufacturer	Bed-of-Nails Tester Tools
GenRad 300 Baker Avenue Concord, MA 01742 (508) 369-4400	Model 2272
Teradyne 179 Lincoln Street Boston, MA (617) 422-3567	Victory Software
Hewlett-Packard Loveland, CO 80539	3070 Series

Manufacturer	JTAG Analyzer
Corelis 12607 Hidden Creek Way Suite H Cerritos, CA 90701 (310) 926-6727	SCANIO/280 SCANTEST Softwares JTAG PROG
Texas Instruments P.O. Box 869305 Plano, TX 75086 (214) 575-6396	Asset

Manufacturer	Software Development System
Advanced Micro Devices P.O. Box 3453 Sunnyvale, CA 94086 (800) 222-9323	MACHXL
Data I/O 10525 Willows Road N.E. P.O. Box 97046 Redmond, WA 98073-9746 (800) 332-8246 or (206) 881-6444	ABEL5
MINC Incorporated 6755 Earl Drive, Suite 200 Colorado Springs, CO 80918 (719) 590-1155	PLDesigner

Manufacturer	Programmer Support
BP Microsystems 1000 N. Post Rd. Houston, TX 77055-7237 (800) 225-2102 or (713) 688-4600	BP1200
Data I/O 10525 Willows Road N.E. P.O. Box 97046 Redmond, WA 98073-9746 (800) 332-8246 or (206) 881-6444	UniSite Model 3900 AutoSite
System General 510 S. Park Victoria Dr. Milpitas, CA 95035 (408) 263-6667	Turpro-1

### Trademarks

Copyright © 1996 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD logo, and MACH are registered trademarks of Advanced Micro Devices, Inc.

MACHXL and MACHPRO are trademarks of Advanced Micro Devices, Inc.

FusionPLD is a service mark of Advanced Micro Devices, Inc.

Product names used in this publication are for identification purposes only and may be trademarks of their respective companies.