AP-654

APPLICATION NOTE

Intel Flash Memory Manufacturing Programming Solutions

February 1998

Order Number: 292212-001

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 5937
Denver, CO 80217-9808

or call 1-800-858-4725
or visit Intel's Website at http://www.intel.com

CONTENTS

REVISION HISTORY

Date of Revision	Version	Description
02/17/98	-001	Original version

1.0 INTRODUCTION

Today's manufacturing environment affords many opportunities for innovative engineers to program flash memory components. Engineers program code into consumer electronic products, during the manufacturing process, and save money doing so. This document investigates various programming methods, compares the benefits and limitations of each, and helps you determine which method best suits your manufacturing requirements.

Programming speed is an important factor in any manufacturing process. The quicker you load code into the flash memory component the more money you save. Different programming methods operate at different speeds. We investigate various methods and speed associated with each type. This helps you determine the programming time budget you need to allow in your manufacturing process. Programming time affects product costs. Be wise about the programming method you select to ensure that you are contributing to a cost-effective product.

Ultimately, the programming method you select should result from close work between the design engineer, product engineer, manufacturing engineer and the third-party equipment vendor. If you analyze everyone's needs and incorporate suggestions from your equipment supplier, you can efficiently program flash memories in your manufacturing process.

Intel works with many third-party vendors to ensure problem-free programming exists on a variety of systems. Programmer support for Intel Flash memory components is available at product introduction on many systems. If the system you use does not support Intel Flash memory components, please contact the equipment vendor to acquire device support.

2.0 PROGRAMMING TERMINOLOGY

Let us begin with a review of programming terminology. Please be aware that different terms are used world-wide for various programming alternatives. We highlight some widely used terms in this document.

- **On-Board Programming**
  - Automatic-Test-Equipment
  - IEEE 1149.1 JTAG Test Access Port
  - In-Circuit Programming
  - In-System Write
- **Off-Board Programming**
  - Engineering Programmer
  - Production Programmer
  - Automated Programming System

2.1 On-Board Programming (OBP)

On-board programming loads code into the flash memory component after it is mounted on the PCB. There are multiple methods to perform OBP. We investigate some of those methods in this document.

2.1.1 AUTOMATIC-TEST-EQUIPMENT (ATE)

Operators functionally test assembled Printed-Circuit Boards (PCBs) with Automatic-Test-Equipment. ATE seeks board faults such as: opens, shorts, missing components, mis-aligned components, etc. Once the PCB passes functional testing, ATE performs OBP to load code into the flash memory component.

2.1.2 IEEE 1149.1 JTAG TEST ACCESS PORT (JTAG)

JTAG (Joint Test Action Group) is an IEEE 1149.1 standard which specifies methods of testing integrated circuits. The IEEE 1149.1 JTAG Test Access Port provides the capability to program flash memory mounted on the PCB. JTAG is an ideal solution for space-constrained PCBs. The JTAG Test Access Port requires a four signal interface to the JTAG compliant component. For instance, think of a cellular telephone. Cellular telephone PCBs are so small that other OBP methods which require access to all the flash memory pins would be difficult to accomplish. Because JTAG uses a serial interface, this OBP method utilizes little PCB area and best fits the needs of space-constrained products.

2.1.3 IN-CIRCUIT PROGRAMMING (ICP)

In-circuit programming is similar to ATE programming. With ICP the engineer connects external equipment to the PCB to drive programming signals to the flash memory component. This external equipment can be a device programmer, tester, custom hardware configuration, etc.

2.1.4 IN-SYSTEM WRITE (ISW)

In-system write utilizes the system CPU to execute flash memory erase and program algorithms. One of several sources ports data to the flash memory: serial or parallel port, floppy diskette or hard disk, modem, etc. An engineer creates erase, program and verify algorithms then downloads these algorithms into the CPU RAM. The CPU executes the algorithms and ports code to the flash memory for updates. ISW is a popular method to perform PC BIOS code updates.

2.2 Off-Board Programming

Off-board programming means that operators load code into individual components away from the product assembly and test processes. Manufacturers mount pre-programmed components on PCBs. Off-board programming equipment ranges from device programmers to automatic handling systems. Typically, equipment operator(s) manage product flow through off-board programmers.

2.2.1 ENGINEERING PROGRAMMER

Engineering programmers drive erase, program and verify signals to individual flash memory components. Programmer manufacturers create and maintain these algorithms for many different programmable components.

2.2.2 PRODUCTION PROGRAMMER

Production programmers drive erase, program and verify signals to multiple flash memory components. Programmer manufacturers create and maintain these algorithms for many different programmable components.

2.2.3 AUTOMATED PROGRAMMING SYSTEM

Automated programming systems typically load individual flash memory components into device programmer sockets. The automated system segregates successfully programmed components from unsuccessfully programmed components.

3.0 OFF-BOARD PROGRAMMERS

There are three categories of off-board programmers: engineering, gang and concurrent. Engineering programmers, as the name implies, most often support engineering work. Their single socket configuration make it easy for an engineer to quickly program a single device. There are two types of production programmers: gang and concurrent. Production programmers provide capability to load code into many components simultaneously.

Manufacturers use off-board programmers to load code into individual flash memory components before mounting the components on the PCB.

3.1 Engineering Programmer

Engineers use device programmers during the prototype stage to load code into the flash memory component, see Figure 1. In this stage code frequently changes. Because of these changes engineers need easy access to the flash memory component. Engineers often mount the flash memory component in a socket on the prototype PCB. They can then remove the flash memory component from the socket, reprogram it in the engineering programmer, then reinstall it into the socket to test the latest revision of code.

Because engineering programmers are low-cost, each engineer typically keeps a programmer at their desk for personal use.

Figure 1. Engineering Programmers Provide Capability to Program One Component at a Time

Engineering programmers are versatile, allowing the user to program many different types of components. Because of the versatility, programming times are slower than production programmers. In an engineering environment slower programming times make little difference since engineers program only a few devices at a time.

Most engineering programmers utilize socket adapters to accommodate different package types. Socket adapters typically convert from a DIP pinout, which most programmers support, to any available package type (see Figure 2).

Socket adapters provide versatility to program different packages. They are used for a particular device configuration. Be aware that a socket adapter may not support every component in a particular device family. Someone programming many different component configurations is likely to have many different socket adapters.

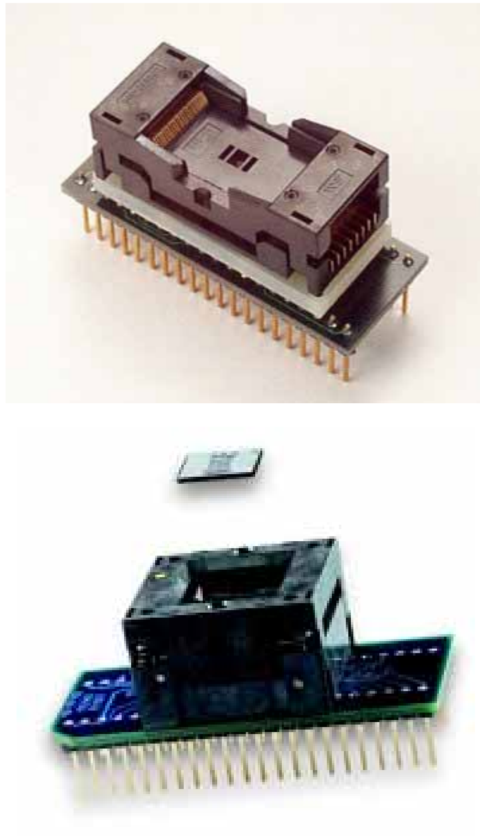

Figure 2. Programming Socket Adapters Convert DIP, a Standard Interface for Most Programmers, to Any Package Type

3.2 Gang Programmer

Gang programmers (Figure 3) load code into 8, 10, 16 or 20 flash memory components at one time. The number of components depends on the number of sockets the programmer supports. Some programmers support a fixed number of sockets. Other programmers daisy-chain together to support many sockets. Gang programmers often support manufacturing environments where high volume programming is required.

Using a gang programmer, the operator typically loads components into all sockets. After filling the sockets the operator depresses the start button to begin loading code into all flash memory components simultaneously. LEDs associated with each socket indicates pass or fail status of each device. Operators manually separate passing and failing components into their respective bins.

Figure 3. Gang Programmers Load Code into All Devices Simultaneously

Some gang programmers use dedicated socket modules for dedicated package types. This means that when an operator programs a new flash memory package they must change socket modules. Other gang programmers use the socket adapter method. In these cases the operator switches socket adapters to accommodate another package.

Gang programmers are a cost effective tool to manually program components. One limitation is the possibility of damage to component leads. Anytime you use manual labor to insert or remove components from sockets, the probability exists to damage device pins.

Because these programmers support manufacturing operations, manufacturers pay close attention to programming times. These programmers load code into flash memory components faster than engineering programmers.

3.3 Concurrent Programmer

Concurrent programmers are similar to gang programmers. They support multiple sockets. One difference between a concurrent programmer and a gang programmer is that the concurrent programmer programs each socket individually. For instance, when an operator loads the flash memory component into the first socket on a concurrent programmer, then closes the socket, the programming process begins. This occurs for all sockets. By the time the operator finishes loading the last socket, the first socket is ready to accept the next unprogrammed device.

Concurrent programmer speed is about the same as gang programmer speed. The concurrent programmer in some respects is more efficient than gang programmers since the operator does not wait for the parts to finish programming before loading the next set of components into the sockets. This continual loading and unloading of components utilizes time more efficiently allowing operators to program more components per hour with a concurrent programmer.

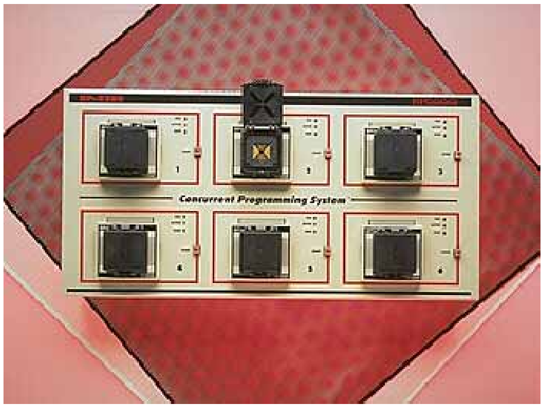

Figure 4. Concurrent Programmers Load Code into Devices on a Socket-by-Socket Basis

4.0 OFF-BOARD PROGRAMMERS

Automated programming systems integrate gang or concurrent programmers with a device handling system. By integrating these two functions, you eliminate the human factor from the programming process. This reduces the probability to damage components by automatically socketing or removing the devices from sockets.

Automated programming systems provide the highest programming throughput. These systems continually place components into sockets or remove programmed components from sockets. No lunch breaks, no coffee breaks—just continual programming. Operators load the delivery bay with unprogrammed parts then the machine processes the parts.

Actual programming times are comparable to gang programmers. Automated programming systems reduce device loading times thus increasing throughput. These systems are more expensive than other device programmers but if you plan to program high volumes of components the return on investment will be quick.

Figure 5. Automated Programming Systems Provide High Throughput, Hands-Off Programming, for High-Volume Manufacturing

5.0 ON-BOARD PROGRAMMING

In response to consumer electronic demands, Intel's Flash memory package sizes keep getting smaller, see Figure 6. With the shrinking flash memory package, OBP is a more appealing method to load code into the flash memory component.

OBP allows hands-free programming. Manufacturers use surface mount assembly equipment to mount blank flash memory components on the PCB. Once mounted, OBP provides hands-free programming.

To perform OBP, some types of test equipment must be able to access all flash memory component pins. This method works well if there is sufficient area on the PCB to connect to all flash memory pins. In other cases, where PCB space is not available to access all flash memory component pins, serial OBP is an option. We discuss both OBP options in this section.

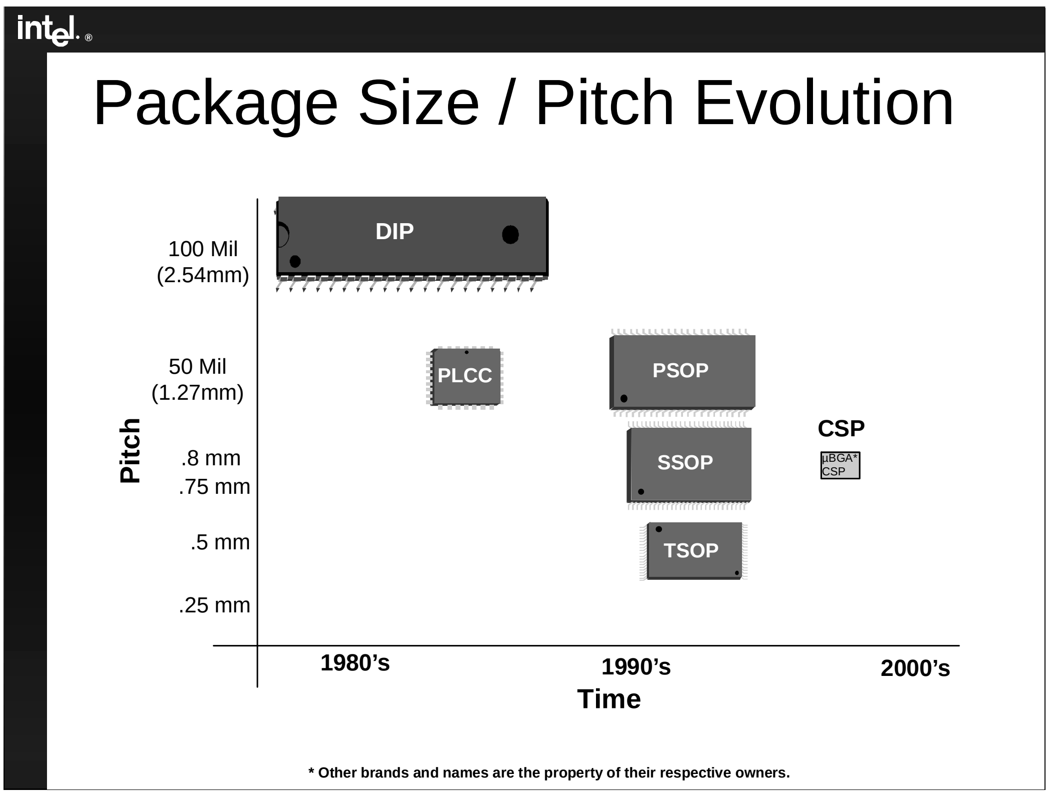

Figure 6. Intel Flash Memory Package Sizes Continually Get Smaller to Meet Consumer Product Requirements

6.0 AUTOMATIC-TEST-EQUIPMENT

Automatic-Test-Equipment performs functional testing on assembled PCBs. ATE seeks opens, shorts, misaligned components and other functional problems. Additionally, ATE can load code into the flash memory component. ATE provides the opportunity to streamline your manufacturing process. For instance, if you perform off-board programming and use ATE to functionally test the PCB why not integrate the two processes? You can avoid off-board programming costs if you integrate OBP into the ATE test flow. One limitation to integrating the two processes is that the manufacturing beat-rate slows down by the time it takes to load code into the flash memory component. Beat-rate refers to the number of products produced on the assembly line. Higher beat-rate means you are producing more products per hour.

ATE loads code into the flash memory component quicker than most programming methods. But there is a trade off, ATE is more expensive than other programming equipment. It does not make sense to purchase ATE specifically for programming, but if you already use ATE it does make sense to integrate the programming process into your test flow.

Figure 7. Automatic-Test-Equipment Programs Flash Memory On-Board

7.0 IEEE 1149.1 JTAG TEST ACCESS PORT PROGRAMMING

JTAG programming is a type of OBP. JTAG is an excellent solution for products that do not have enough space available on the PCB to connect a tester interface to every flash memory pin. JTAG uses a four wire serial interface to program the flash memory component (see Figure 8). Be aware that the flash memory component does not have JTAG compliant circuitry in it. You cannot perform JTAG programming with only the flash memory component. To perform JTAG programming you must connect a JTAG compliant device to ALL flash memory pins.

Many cellular telephone manufacturers use JTAG programming to load code into the flash memory component. Think about how small cellular telephone PCBs are. With small PCBs there is no room available to connect an ATE bed-of-nails interface to the PCB test-land-pads. Therefore, to perform OBP, cellular telephone manufacturers use the four-wire JTAG interface. These manufacturers load code into the flash memory component on millions of telephones per year.

JTAG programming is fast if you appropriately setup all hardware and software conditions. To obtain the best programming times run TCK, test clock input, at the fastest rate the JTAG compliant device can accept. Be sure the equipment you use to send the programming signals to the JTAG compliant device operates quick enough that the flash memory is not waiting for data. Ideally, the programming equipment should deliver the programming signals and data to the flash memory component quicker than the flash device can accept data. With most equipment today, the flash memory component waits for data and programming signals from the programming equipment.

The JTAG programming equipment shown in Figure 9 is used world-wide by many manufacturers to program flash memories.

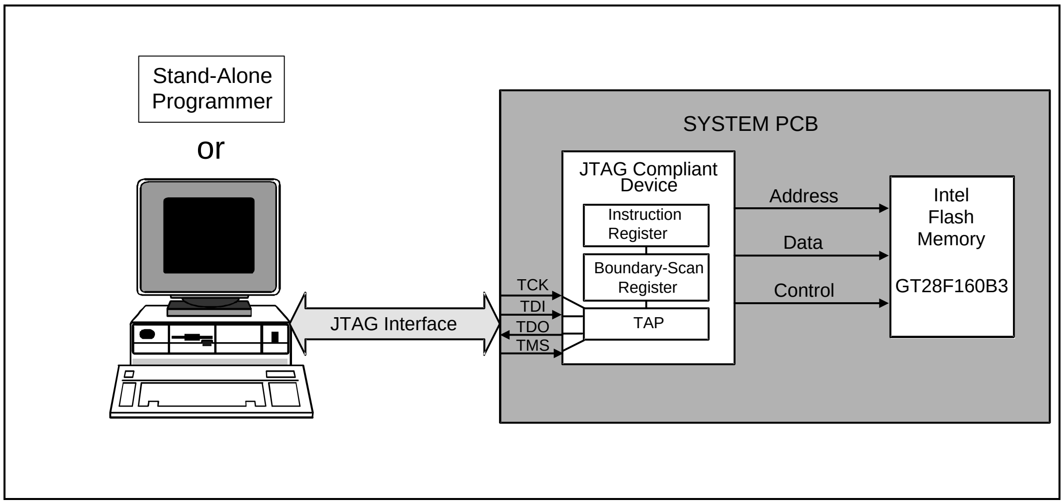

Figure 8. A JTAG-Compliant Device Connected to the Flash Memory Component Allows Serial Programming Capability

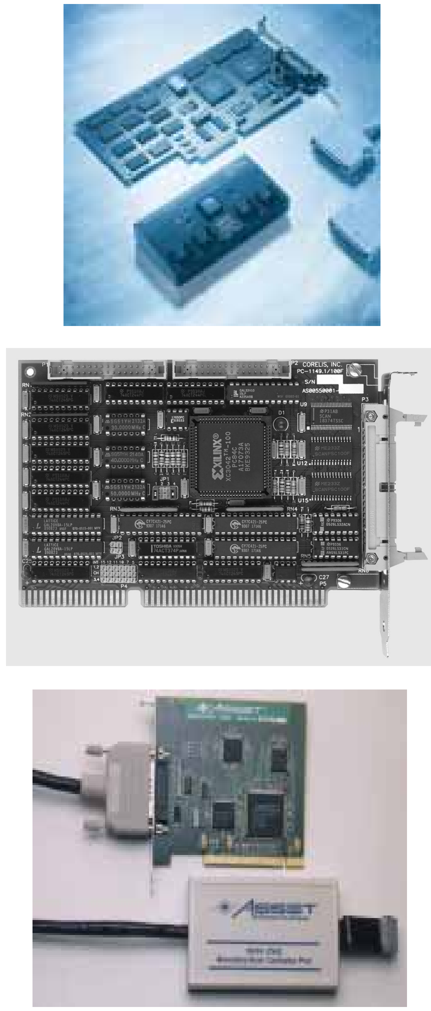

Figure 9. JTAG Programming Equipment Enables OBP for Space-Constrained PCBs

8.0 PROGRAMMING TIME

Programming time directly affects product costs. The longer it takes to program devices the greater manufacturing costs are. Intel works directly with many third-party vendors to reduce programming times. We created a technical paper that describes methods to improve programming times: *Improving Programming Throughput of Automated Flash Memories*, order number 297769.

In order to share programming times, as they relate to different equipment, we performed programming benchmark comparisons with the 28F160B3 device using different platforms (Figure 10). Use this chart to better understand how programming times affect your manufacturing beat-rate. Be aware that the times shown in the chart are general for a wide range of equipment which falls into the particular category represented. Your programming times depend heavily on the specific manufacturers' equipment and software revision you use. To best understand programming times for specific equipment you should perform a benchmark comparison using that equipment.

Intel Flash memories will always provide the best programming times obtainable. Intel's two write-cycle programming command sequence, combined with the extra work we do to improve programming times, provides the quickest available programming throughput. This is true for our entire product line. Intel Flash memories will save time and money in your manufacturing flow.

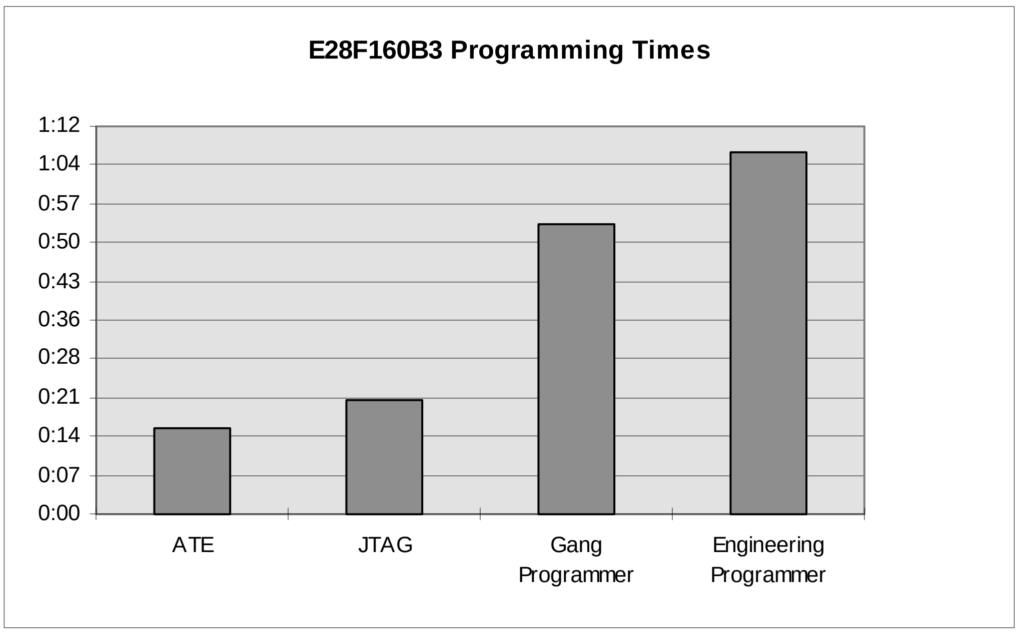

Figure 10. Programming Times Are Different Depending on Which Method You Use

9.0 SUMMARY

Flash memory programming methods are abundant in today's manufacturing environment. To select the best programming method you need to understand the various options.

Device programmers provide capability to manually or automatically program components off-board. These programmers offer a variety of capabilities and programming speeds to best fit specific requirements.

OBP is quickly becoming popular. With flash memory component package sizes shrinking, OBP provides capability to load code into the flash memory components without human intervention. Different OBP methods best fit different needs. ATE most frequently supports PCBs with enough area to allow test-land-pad connections for all flash memory component pins. JTAG programming supports loading code into flash memories on PCBs where little space is available for interface connections.

Use this application note to better understand different programming alternatives. To determine the best programming method for your application work internally with your design and manufacturing engineers and externally with the third-party programming vendor. This will help you understand how specific equipment affects your manufacturing line beat-rate.

10.0 INTEL REFERENCES

Order Number	Document
210830	*Intel Flash Memory Databook*
292179	*AP-624 Introduction to On-Board Programming With Intel Flash Memory*
292185	*AP-629 Simplify Manufacturing by Using Automatic-Test-Equipment for On-Board Programming*
292186	*AP-630 Designing for On-Board Programming Using the IEEE 1149.1 (JTAG) Access Port*
297691	*Designing For Successful Flash Memory Read and Verify Operations* Technical Paper
297769	*Improving Programming Throughput of Automated Flash Memories* Technical Paper
295112	*Advanced Flash Memory Packages Drive OBP Innovation* SMT Article Reprint

11.0 OTHER REFERENCES

IEEE Std 1149.1 Standard Test Access Port and Boundary-Scan Architecture (available from IEEE, Inc,. 345 East 47th Street, New York, NY 100167, USA)

The Boundary-Scan Handbook, author: Kenneth P. Parker, publisher: Kluwer Academic Publishers

Boundary-Scan Test, authors: Harry Bleeker, Peter van den Eijnden, Frans de Jong, publisher: Kluwer Academic Publishers

AP-Note JTAG-102, *JTAG In-Circuit Programming of Flash Memory* available from Corelis (see Appendix for vendor contact information)

APPENDIX A
VENDOR REFERENCES

Intel provides a variety of tools and vendor information to support the product definition, design, prototyping and production phases using our Flash Memory components. To access these tools please visit the Intel WWW at: http://developer.intel.com/design/flcomp/toolbrfs/

NOTE

Programming options were selected from products offered by a variety of vendors. Since this industry develops many new solutions each year, Intel recommends that designers contact vendors for their latest products. Intel will continue to work with the industry to develop optimum solutions for programming flash memories. The hardware vendor remains solely responsible for the design, sale, and functionality of its product, including liability arising from product infringement or product warranty.

Vendor Name	Telephone	WWW URL
Programmer Vendors		
Advantech Instrument Corp., Taiwan	886-2-218-4567	http://www.aec.com.tw
Advantest Corp., Japan	81-33-342-7500	http://207.240.44.26/index-e.html
Advin Systems Inc., USA	(800) 627-2456	http://www.advin.com
Aval Data Corp., Japan	81-44-952-1322	http://www.avaldata.com
Data I/O Corp., USA	(800) 247-5700	http://www.data-io.com
B&C Microsystems Inc., USA	(408) 730-5511	http://www.bcmicro.com
BP Microsystem Inc, USA	(800) 225-2102	http://www.bpmicro.com
Bytek Corp, USA	(407) 994-3520	http://www.bytek.com
Elan Digital Systems Ltd, United Kingdom	44-1489-579799	http://www.elan-digital.demon.co.uk/
Hi-Lo System Research Co. Ltd, Taiwan	886-2-764-0215	http://www.hilosystems.com.tw
ICE Technology Ltd., United Kingdom	44-1226-767404	http://www.icetech.com
Logical Devices Inc., USA	(800) 331-7766	http://www.logicaldevices.com/
Needhams Electronics, USA	(916) 924-8037	http://www.needhams.com
Minato Electronics Inc., Japan	81-45-591-5605	not available
SMS Mikrocomputer Systeme GmbH, Germany	49-7522-97280	http://www.sms-sprint.com
Stag Programmers Ltd., United Kingdom	44-1707-332148	http://www.stag.co.uk/
System General Corp., USA	(800) 967-4776	http://www.sg.com.tw/

Vendor Name	Telephone	WWW URL
ATE Vendors		
GenRad Corp., USA	(800) 443-6723	http://www.genrad.com
Hewlett Packard Co., USA	(800) 452-4844	http://www.hp.com
Teradyne Inc., USA	(510) 932-6900	http://www.teradyne.com
JTAG Vendors		
Asset-InterTech, USA	(214) 437-2800	http://www.asset-intertech.com/
Corelis, USA	(310) 926-6727	http://www.corelis.com/
Data I/O Corp., USA	(800) 247-5700	http://www.data-io.com
Needhams Electronics, USA	(916) 924-8037	http://www.needhams.com
JTAG Technologies B.V., Netherlands	31-40-295-08-70	http://www.jtag.com/
Socket Adapter Vendors		
California Integration Coordinators, USA	(916) 626-6168	http://www.cic-inc.com/
EDI Corp., USA	(702) 735-4997	http://www.edi-adapters.com/
Emulation Technology, USA	(408) 982-0660	http://www.emulation.com
Logical Systems Corp., USA	(315) 478-0722	http://www.logicalsys.com/