intel®

AP-629

APPLICATION NOTE

Simplify Manufacturing by Using Automatic-Test-Equipment for On-Board Programming

PETER T. LARSEN
TECHNICAL MARKETING
ENGINEER

June 1996

Order Number: 292185-001

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 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683 CG-041493

CONTENTS

FIGURES

REVISION HISTORY

Number	Description
-001	Original version

1.0 INTRODUCTION

Multiple programming solutions exist to write data to Intel Flash memory components. This document explores the following solution: Using Automatic-Test-Equipment (ATE) to program Intel Flash memory components on a printed circuit board (PCB).

Specifically, we will investigate on-board programming (OBP) with ATE. OBP saves manufacturing costs and reduces susceptibility to problems associated with manual programming alternatives.

In this document we'll review the following topics:

- ATE overview
- OBP benefits and limitations
- PCB hardware design requirements
- ATE considerations
- Programming algorithm optimization ideas
- ATE benchmark data for 28F002BC flash memory

After reading this application note, you will have a clear understanding of how to program Intel Flash memory components on ATE systems. For detailed ATE system information please contact the appropriate manufacturer. You can find ATE vendor references in Appendix A.

If you seek introductory OBP information please order Intel Ap-Note 624, *Introduction to On-Board Programming with Intel Flash Memory*, order number 292179, from the Intel Literature Center. This document provides a high-level overview of OBP methods and design rules.

2.0 WHAT IS AUTOMATIC-TEST-EQUIPMENT?

Manufacturing engineers use ATE to perform in-circuit testing on assembled PCBs. ATE validates PCB construction by seeking assembly faults, open traces, shorted traces, misaligned components, and missing components. You can obtain additional value from ATE by integrating flash memory programming routines in the test flow.

Assembled PCBs connect to ATE systems via a bed-of-nails interface, see Figure 1. Bed-of-nails refers to multiple spring loaded probes that connect to special test land pads strategically located on PCB traces. The ATE pin drivers send test signals through the probes which then get routed to the PCB components via test land pads.

Test engineers develop unique test programs for each variation of PCB tested. They use automatic test generator tools, provided by ATE vendors, to reduce software development time. Automatic test generators create the majority of test sequences. The engineer must perform some manual code development for a complete test vector portfolio.

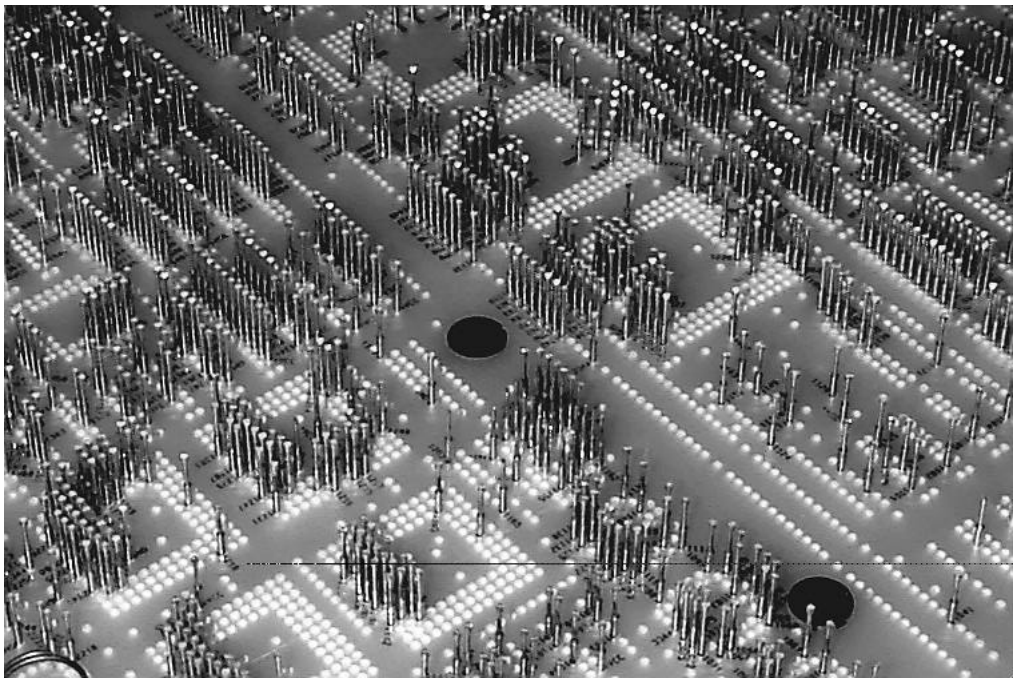

Figure 1. Probe Contacts Link the Board under Test to the ATE System

3.0 USING AUTOMATIC-TEST-EQUIPMENT TO PERFORM OBP

Any programming solution, including OBP, provides both strengths and limitations. Please analyze your project requirements and consider the following OBP capabilities and shortcomings before implementing OBP for your project.

Strengths:

- Allows Just-In-Time (JIT) programming
- Eliminates off-line programming
- Reduces manual handling of components
- Eliminates assigning separate part numbers to programmed parts
- No individual device labeling necessary
- Enables use of economical tape and reel shipping media for flash memory components
- No inventory storage required
- Additional handling equipment not required for Small Outline Packages (SOP)
- Reduces susceptibility to bent component leads
- Eliminates flash memory sockets
- Reduces manufacturing expense

Limitations:

- Design for test to enable ATE to program the flash memory, without causing signal conflicts or bus contention, may result in additional circuitry on the PCB
- Project cost may increase due to additional hardware
- manufacturing line through-put time (beat-rate) will increase by programming time
- PCBs with limited available space may not accommodate test land pads which will prevent using your ATE equipment for programming

You can streamline flash memory programming by integrating the process in your ATE test flow. By doing this you avoid the costs of maintaining programming equipment, personnel, dedicated floor space and outsourcing programming functions to third-party vendors.

Some two-sided PCBs with limited available space are difficult to interface to ATE systems. Because of space constraints engineers sometimes exclude test land pads from the board design. Test engineers use alternative test and programming methods for these boards. Flash memory programming via the JTAG Test Access Port (TAP) can be an option in these cases. Intel will provide specific information about programming flash memory with the JTAG TAP in Ap-Note 630, *Designing for On-Board Programming Using the JTAG Test Access Port*, order number 292186,

3.1 Manufacturing Requirements for ATE Programming

To allow flash memory programming on ATE systems the design engineer must provide bed-of-nails probe access to all component pins. An ATE-compliant PCB design provides adequate-sized test land pads leading to the component, for the chosen type of test fixture.

Do not pull-up/down certain enable pins like V_{PP} and RP# directly to V_{CC} or GND. If you want these pins pulled-up/down, use series resistors to isolate the pins from V_{CC} or GND. This allows the ATE system, during programming operations, to set the appropriate voltages. For normal system operation the resistors connected to V_{CC} or GND pull the pins to the desired voltage level.

Design engineers who create PCB applications for ATE systems must take special consideration for some flash memory pins. During programming you must isolate device pins driven to +12V from other PCB circuits to not damage the pin drivers. Examples of these types of pins are V_{PP} and RP# pins. V_{PP} provides +12V for program and erase operations otherwise it is held at GND. RP#, the reset/deep power-down pin may also be biased with up to +12V.

For these special considerations you can incorporate resistors in the PCB application, see Figure 2. In this example the V_{PP} voltage for the 28F002BC-T is +12V during programming. If desired, for SmartVoltage devices, you can use the ATE variable power supply to set V_{PP} to +5V. For normal board operation, you can switch the system power supply out of the circuit and V_{PP} will be held at GND to protect the flash memory against inadvertent programming.

Design engineers also must address power and GND noise and signal integrity issues in some OBP environments. ATE pin drivers often send signals through lengthy single-strand wire, through bed-of-nails probes, onto the PCB, then through PCB traces before the signal reaches the flash memory. Routing the signal long distances adds capacitance and inductance to the transmission line and, at high speeds, it degrades signal integrity. Power and GND transient noise spikes are another possible source of problems. For solutions to these types of problems please review the Intel Technical Paper *Designing for Successful Flash Memory Read and Verify Operations*, order number 297691. This document suggests some basic design rules that help reduce power and GND transient voltage spikes.

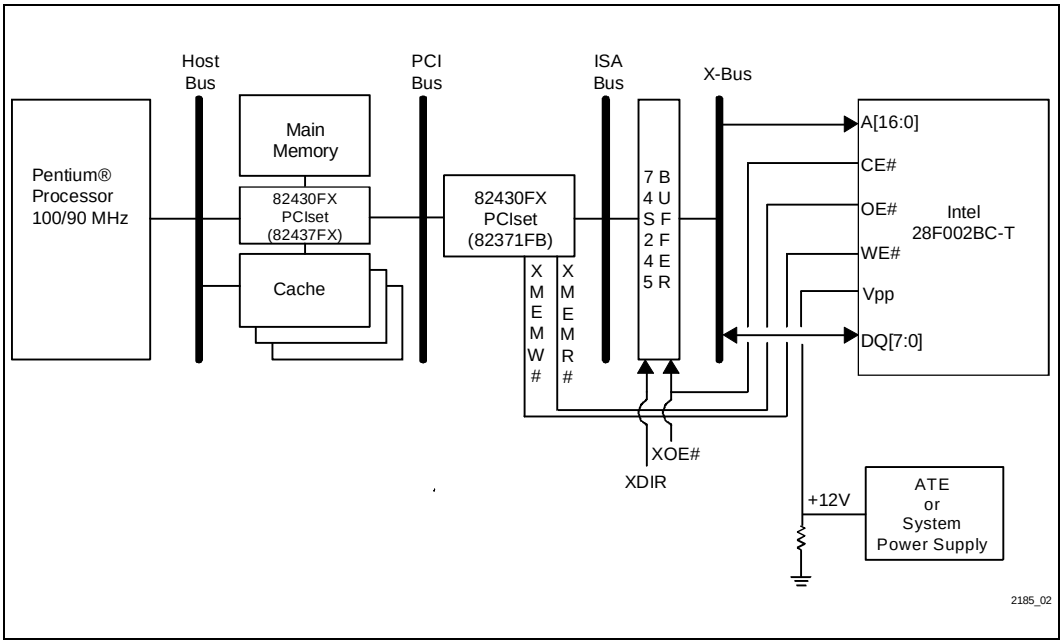

Figure 2. Notice the V_{PP} Pin on the 28F002BC-T Can Be Set to +12V for ATE Programming or GND for Normal System Operation

3.2 Design for Test with ATE Programming

Test engineers who create ATE test programs must tri-state all peripheral signals connected to flash memories before starting the flash memory programming operation. These include address, data and control signals from CPU, memory controllers, and other memory components. It is important to tri-state signals to avoid bus contention. ATE systems are capable of overdriving signals, but this is not recommended for MOS components (it is safe to overdrive signals for only a few milliseconds). Flash memories may take many seconds to program, therefore you cannot overdrive signals without potentially damaging the interconnecting devices.

If your application includes a microprocessor you can tri-state its bus with the RESET, HOLD or ONCE modes. Any of these modes places the microprocessor control signals and local bus in a high-impedance state. It is important to understand how to tri-state peripheral devices connected to the flash memory. To do this, you need to review the manufacturers' product datasheets and implement the tri-state mode as suggested.

In addition, be careful not to forward bias pins on peripheral components around the flash memory, see Figure 3. Forward biasing a peripheral component occurs when input voltages exceed V_{CC} of the peripheral component. You can avoid forward biasing components by maintaining stringent voltage tolerances supplied by the ATE.

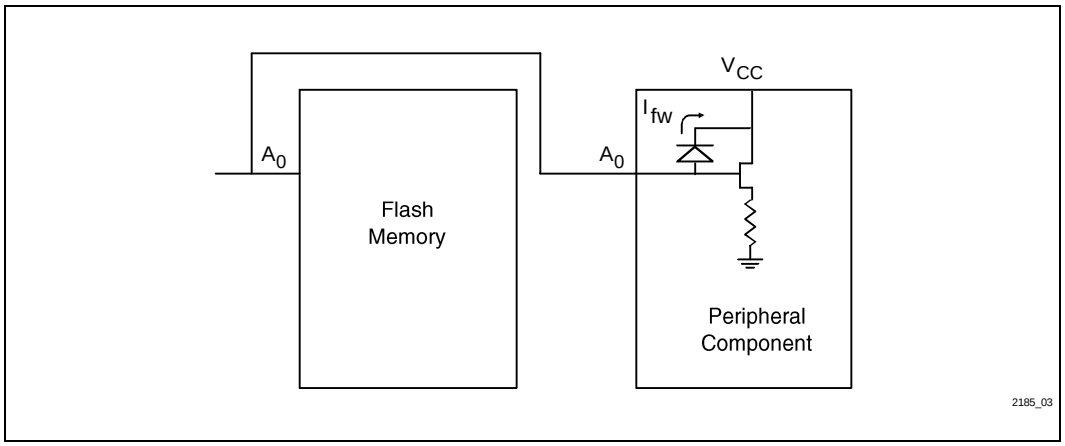

Figure 3. Forward Biasing Occurs When Input Voltages Exceed V_{CC}. In This Case, the Peripheral Component ESD Protection Diode Sinks Current When Forward Biased.

4.0 PROGRAMMING ALGORITHM OPTIMIZATIONS

An optimized programming algorithm will make the difference between slow programming and fast programming on ATE systems. This section provides some ideas about how you can improve programming times and in turn speed up your test process manufacturing beat-rate.

4.1 Test Flow Optimization

To understand test flow optimizations refer to the ATE programming flowchart, Figure 4. To reduce programming time and increase manufacturing beat-rate (allow more boards per hour to be processed) consider modifying the flow to perform only necessary operations. For instance, Intel ships blank (erased) flash memory components from the factory. You can reduce test time by eliminating erase check for new devices.

Eliminating redundant program verify checks are another optimization opportunity. The flash memory internal Write State Machine (WSM) automatically verifies data written to the memory. Program verify operations initiated by the ATE are redundant with flash memory internal program verify operations. You can save time by not performing program verify operations with the ATE.

Consider programming in word-wide mode (x16) when possible. Some devices are configurable for x8 or x16 and can be programmed in either mode. You can program these devices in half the time if you select word programming over byte programming.

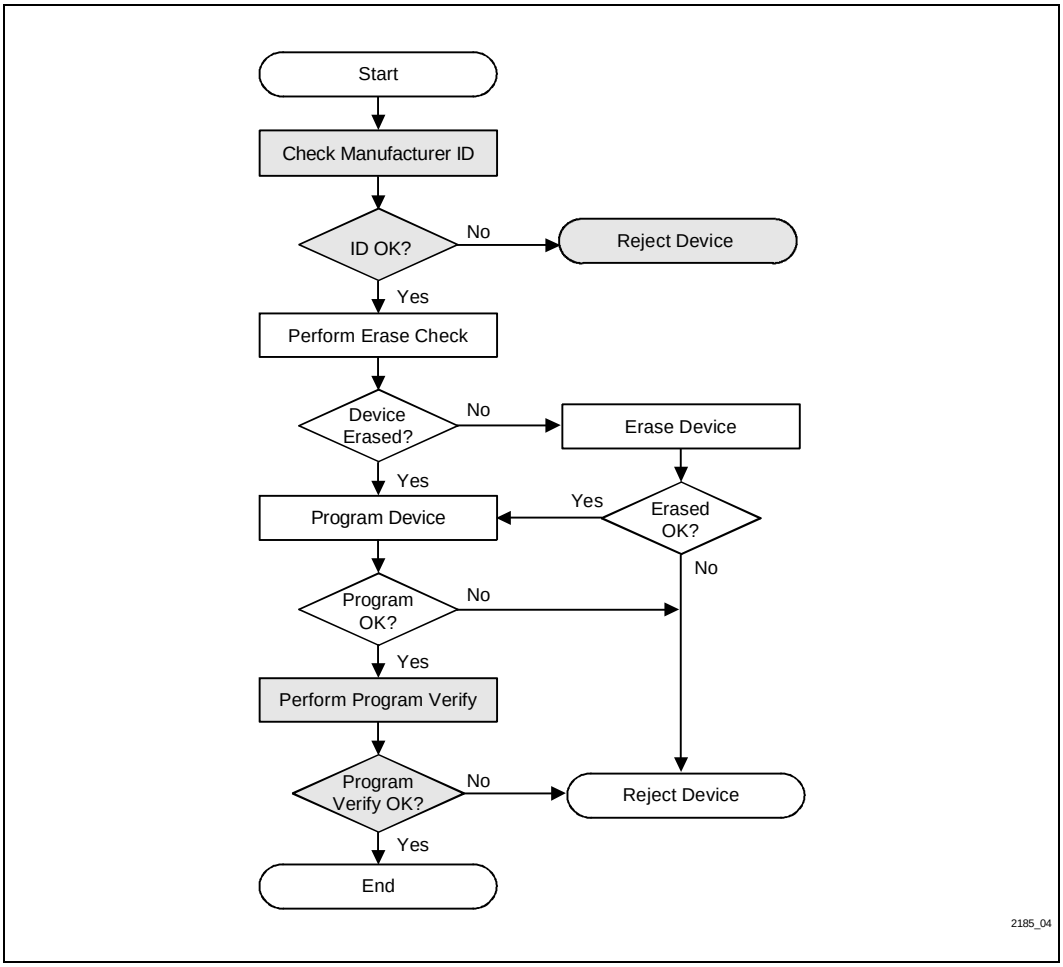

Figure 4. Darkened Areas in This ATE Programming Flowchart Can Be Omitted to Optimize Programming Time

4.2 Flash Memory Algorithm Optimization

The flash memory WSM controls programming operations. The WSM executes a sequence of internally timed events to:

1. Program the desired bits of the addressed memory byte/word
2. Verify that the desired bits are sufficiently programmed

You can optimize byte/word programming on all flash memories except for the 28F256A, 28F512, 28F010 and 28F020 with the following suggestions: The status register bit SR.7 (write state machine status) indicates programming status. While the programming sequence is executing, SR.7 is “0” (busy). When programming is complete, SR.7 is “1” (ready). You can poll SR.7 to determine when the byte/word programming is complete then immediately begin the next programming operation. This effort reduces overall device programming time.

Please review product specifications in the *Intel Flash Memory Databook* for more information on status register operation.

Some flash memory devices have a RY/BY# pin. This pin also indicates the status of the internal WSM. It functions similar to the status register bit SR.7, the only exception is that you do not need to pulse OE# low to read RY/BY#. Use the RY/BY# pin to determine when the WSM is finished performing a write operation then immediately start the next byte/word programming operation. The RY/BY# pin on the 28F016 devices are open drain, thus a pull-up resistor is necessary to see a signal transition on this pin.

4.3 Read/Write Cycle Optimization

You should optimize ATE code to reduce software overhead. Each signal transition performed by the ATE requires a certain amount of time to execute. For instance, if you pulse CE# low at each address location you consume a certain amount of additional system overhead for each CE# transition. To reduce the system overhead switch CE# low at the first address location and leave CE# low while programming the entire memory array.

NOTE

If this software optimization is implemented be certain to adhere to databook specifications that typically restrict input rise and fall times to < 10 ns, (10%–90%).

4.4 Method for Checking Byte/Word Programming Times

You can determine byte/word programming times by connecting oscilloscope probes to your flash memory component and capturing plots while programming. Use a digital signal oscilloscope with single acquisition mode. Perform this experiment to determine byte/word programming time:

Oscilloscope setup:

Ch. 1 = A_0

Ch. 2 = WE#

Ch. 3 = OE#

Ch. 4 = DQ_7

(DQ_7 is SR.7 when reading the Status register)

horizontal timebase to 5 µs per division

vertical amplitude to 1V per division

Look at programming cycle time from A_0 falling to A_0 rising. Within this time period you will see WE# pulse low twice, once to write the program setup command and once to write the address location and data. You will also see OE# pulse low to read the SR.7 bit. Optimized programming conditions exist when the time between SR.7 going high (WSM ready) and the next address location is minimized. Any time delay between SR.7 switching high and the beginning of the next program cycle is system overhead. Reduction in this time will improve overall device programming time.

Other ATE system factors influence programming times but to a lesser degree than byte/word programming time. For instance, some ATE architectures with less memory than is needed to program the entire flash memory require system overhead to periodically download more data from the ATE computer to main memory. These memory updates can occur many times per programming session, thus increasing the programming time. In contrast, byte/word programming time occurs up to four million times for the 28F032SA x8 mode, or less times depending on which Intel Flash memory you use. When totaled, byte/word programming time substantially influences overall device programming time.

5.0 ATE BENCHMARK DATA FOR E28F002BC-T FLASH MEMORY COMPONENT

ATE users frequently ask, “How fast can my ATE system program flash memory?” This is important because programming flash memory decreases the manufacturing beat-rate which directly impacts the quantity of products coming off the manufacturing line. Please review the E28F002BC-T benchmark data collected from several industry leading ATE systems. For specific information about how fast your ATE programs a particular device you must perform the same benchmark experiment with your system.

ATE vendors continuously improve their flash memory programming algorithms for quicker programming times. The benchmark data contained in this document are reflective of data collected in May, 1996. Programming times may be faster when you read this application note due to improved algorithms. Please refer to your ATE vendor for updated benchmark information.

Intel E28F002BC-T Flash Memory Benchmark Data

	Minimum (seconds)	Maximum (seconds)
Check Manufacturer ID	0.04	0.10
Erase Check	0.22	1.00
Erase Device	3.21	3.55
Program Device	3.00	6.60
Program Verify Device	0.13	1.50

NOTE:
We programmed a data pattern of 00, 00, 00... into the entire 256 Kbytes of the E28F002BC memory array.

6.0 CONCLUSION

Programming flash memory with ATE systems both streamlines the programming process and reduces manufacturing costs. OBP particularly fits the needs of today's small outline IC packages. It eliminates manual programming operations and enables flash memory users to move to economical tape and reel shipping media.

You can program Intel Flash memories on an ATE system very fast, as shown in the E28F002BC benchmark section, if you incorporate the optimization ideas suggested in this document. The Intel databook specifies worst-case programming times. In some instances, the specified time is longer than actual optimized programming time on an ATE system.

If you already own an ATE system and perform off-line programming you should consider integrating the two processes to take advantage of manufacturing cost savings. Please review your project requirements, understand the programming concepts outlined in this document, and work with your ATE vendor to obtain the best programming results from your ATE system.

APPENDIX A
ATE VENDOR REFERENCES

NOTE

OBP 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.

Hewlett Packard

Australia	(61/3) 890-0326	Italy	02.92.122.241
Austria	0660/8004	Japan	81-4-2648-0722
Belgium	02/778 3417	Malaysia	603-291-0213
Canada	(800) 387-3154	Netherlands	020-547 6222
Czech Republic	42/2/2435 5808	New Zealand	(64/4) 802-6800
Finland	(90) 8872 2100	Norway	22 73 57 59
France	(1) 69.82.60.20	South Africa	27/11/8061193
Germany	(0 70 31) 14-63 33	Spain	(91) 631.13.23
Greece	30/1/726 40 45	Sweden	08-444 2277
Hungary	36/52/415580	Switzerland	01/735 72 00
Israel	972/3/53 80 378	United Kingdom	0344-366666
Ireland	01/2844 633	United States	(800)-452-4844 xINTL

GenRad

Canada	1-(905)-898-4181
France	33.1.49442200
Germany	49.89.431990
Italy	39.2.580701
Singapore	65.278.4400
Switzerland	41.1.422420
United Kingdom	44.1.628.544200
United States	1-(800)-4-GENRAD
All Other Countries	1-(508)-287-7000

Teradyne, Inc.

USA:
Teradyne Inc.
2625 Shadelands Drive
Walnut Creek, CA 94598
Phone 510-932-6900
Fax 510-934-0540
Worldwide web URL = http://www.teradyne.com
Internet mail = teranet@teradyne.com

Other Countries:

Australia	61) 46 251378
Austria (via Germany)	(49) 89 4 18 61 0
Benelux (via France)	(33) 1 46 13 15 00
Brazil	(55) 11 273 5677
Bulgaria	(359) 2 277455
China PRC	(86) 1 407 2200
Denmark (via UK)	(44) 1344 426899
Eastern Europe (via Bulgaria)	(359) 2 277455
France	(33) 1 46 13 15 00
Germany	(49) 89 4 18 61 0
Hong Kong, PRC	(852) 730 3131
India	(91) 44 4926902
Indonesia (via Singapore)	(65) 773 0788
Ireland (via UK	(44) 1344 426899
Israel	(972) 3 751 2929
Italy	(39) 2 213 4601
Japan	(81) 3 3719 0151
Korea	(82) 2 548 9888
Malaysia	(60) 4 6579584
New Zealand (via Singapore)	(65) 773 0788
Norway (via Sweden)	(46) 8 59092000
Pakistan (via Singapore)	(65) 773 0788
Philippines (via Singapore)	(65) 773 0788
Portugal (via Spain)	(34) 1 373 91 76
Singapore	(65) 773 0788
Spain	(34) 1 373 91 76
Sweden	(46) 8 59092000
Switzerland (via Germany)	(49) 89 4 18 61 0
Taiwan ROC	(886) 3 3268300
Thailand (via Singapore)	(65) 773 0788
United Kingdom	(44) 1344 426899

APPENDIX B
ADDITIONAL INFORMATION

RELATED INFORMATION (1,2)

Order Number	Document/Tool
210830	Intel *Flash Memory Databook*
292166	*AP-612 Improving Programming Throughput of Automated Flash Memories*
292179	*AP-624 Introduction to On-Board Programming with Intel Flash Memory*
292186	*AP-630 Designing for On-Board Programming Using the JTAG Test Access Port*
297691	*Designing for Successful Flash Memory Read and Verify Operations*

NOTES:

1. Please call the Intel Literature Center at (800) 548-4725 to request Intel documentation. International customers should contact their local Intel or distribution sales office.
2. Visit Intel's World Wide Web home page at http://www.Intel.com for technical documentation and tools.