Intel® 82802AB/82802AC Firmware Hub (FWH)

Datasheet

April 1999

Order Number: 290658-001

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® 82802AB/AC (FWH) Firmware Hub may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

I^2C is a 2-wire communications bus/protocol developed by Philips. SMBus is a subset of the I^2C bus/protocol and was developed by Intel. Implementations of the I^2C bus/protocol may require licenses from various entities, including Philips Electronics N.V. and North American Philips Corporation.

Alert on LAN is a result of the Intel-IBM Advanced Manageability Alliance and a trademark of IBM

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

Intel Corporation
www.intel.com
or call 1-800-548-4725

*Third-party brands and names are the property of their respective owners.

Contents

Figures

Tables

Revision History

Rev.	Draft/Changes	Date
-001	- Initial Release	April 1999

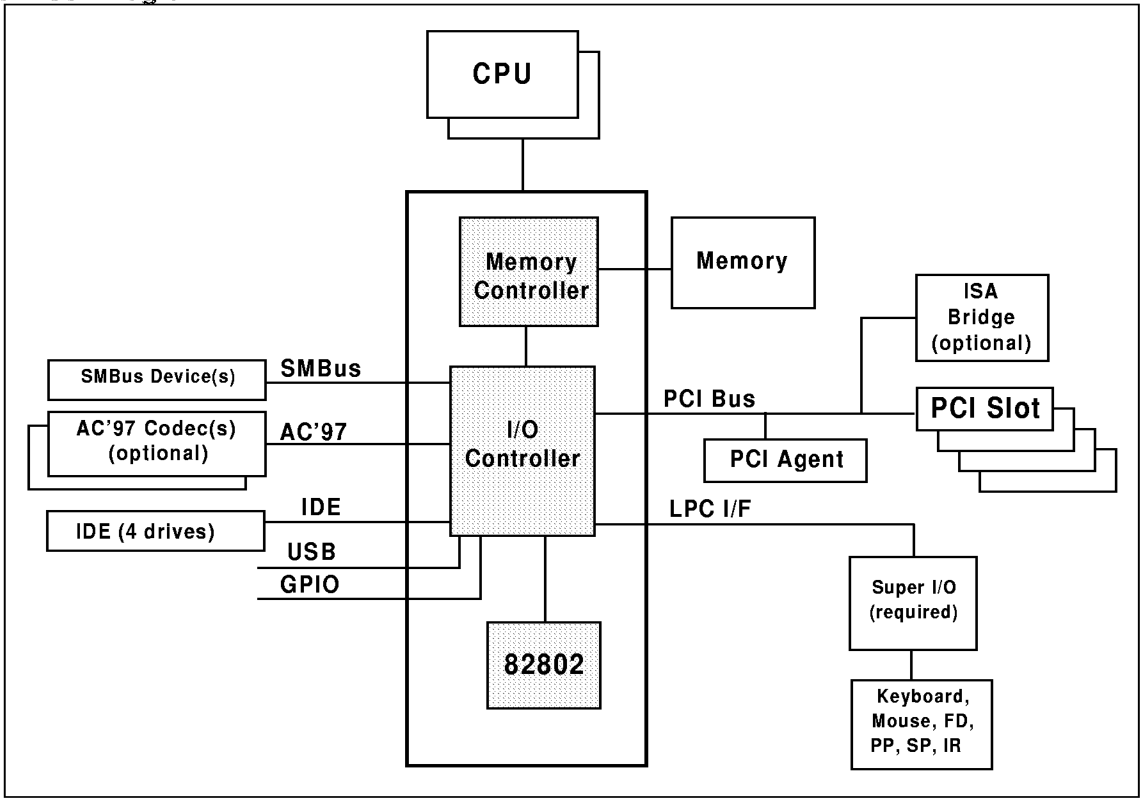
Intel® 82802AB/AC (FWH) Firmware Hub

Product Features

- Intel Platform Compatability
  - Enables security-enhanced platform infrastructure, facilitates option to remove ISA
  - Part of the integrated Intel® 810 Chipset
- Firmware Hub Hardware Interface Mode
  - 5 Signal communication interface supporting x8 reads and writes
  - Register-based read and write protection for each code/data storage block
  - Hardware write protect pins for the top boot block and the remaining code/data storage blocks
  - 5 Additional GPIs for platform design flexibility
  - Supports a hardware Random Number Generator (RNG) for enabling enhanced platform security
  - Integrated Command User Interface (CUI) for requesting access to locking, programming and erasing options. The CUI will also handle requests for data residing in status, ID and block lock registers
  - Operates with 33Mhz PCI clock and 3.3V I/O
- Industry Standard Packages (40L TSOP or 32L PLCC)
- Two Configurable Interfaces
  - Firmware Hub interface for platform operation
  - Address/Address Multiplexed (A/A Mux) interface for programming during manufacturing
- 4M or 8M-bits of Flash Memory for platform code/data nonvolatile storage
  - Symmetrically blocked, 64K-byte memory sections
  - Available in 8Mbit (82802AC) and 4Mbit(82802AB)
  - Automated byte program and block erase through an integrated Write State Machine (WSM)
- Address/Address Multiplexed (A/A Mux) Interface/Mode
  - 11 Pin multiplexed address and 8 pin data I/O interface
  - Supports fast on-board or out-of-system programming for manufacturing
- Case Temprature Operating Range
- Power Supply Specifications
  - Vcc: 3.3v +/- 0.3v
  - Vpp: 3.3v and 12v for fast programming, 80ns

The Intel® 82802 (FWH) Firmware Hub may contain design defects or errors known as errata which may cause the products to deviate from published specifications. Current characterized errata are available on request.

Simplified Block Diagram



1. Architectural Overview

The Intel® 82802 (FWH) Firmware Hub component is part of several integrated Intel® chipsets. The FWH is key to enabling future security and manageability infrastructures for the PC platform. The device operates under the FWH interface/protocol. The hardware features of this device include a Random Number Generator (RNG), five General Purpose Inputs (GPIs), register-based block locking, and hardware-based locking. An integrated combination of logic features and non-volatile memory enables better protection for the storage and update of platform code and data, adds platform flexibility through additional GPIs, and allows for quicker introduction of new security/manageability features into current and future Intel® architecture-based platforms. See "Product Features," above, for a list of key features that the FWH provides.

Intel will provide a Windows* driver to give third party software access to our RNG for use as a security feature. Also provided will be a device driver developer kit (DDK) for operating system vendors who wish to design security drivers for their platform.

1.1. Interface Overview

This device is equipped with two hardware interfaces. The "IC" (Interface Configuration) pin of the device provides the control between the interfaces. The interface mode needs to be selected prior to power-up or before return from reset (RST# or INIT# low to high transition). The FWH interface is designed to work with the I/O Controller Hub (ICH) during platform operation. The A/A Mux interface is designed as a programming interface for OEMs to use during motherboard manufacturing or component pre-programming. The A/A Mux interface is not intended for use during regular personal computer operation – doing so will cause the system boot sequence to fail upon power-up.

An internal Command User Interface (CUI) serves as the control center between the two device interfaces (FWH and A/A Mux) and internal operation of the nonvolatile memory. A valid command sequence written to the CUI initiates device automation. An internal Write State Machine (WSM) automatically executes the algorithms and timings necessary for block erase and program operations.

Driving RST# or INIT# low resets the device, which resets the block lock registers to their default (write-locked) condition, and clears the status register. A reset time (t_{PHQV} A/A Mux) is required from RST# or INIT# switching high until outputs are valid. Likewise, the device has a wake time (t_{PHRH} A/A Mux) from RST# or INIT# high until writes to the CUI are recognized. A reset latency will occur if a reset procedure is performed during a programming or erase operation.

Note: There is no chip enable (like CE#) in either interface. Standby current control in the FWH interface is enabled automatically, if FWH4 is high and the device is not working to complete a requested activity.

1.1.1. Firmware Hub Interface

The Firmware Hub (FWH) interface consists primarily of a 5 signal communication interface used to control the operation of the device in a system environment. The buffers for this interface were designed to be PCI compliant. To ensure the effective delivery of security and manageability features, the FWH interface is the only way to get access to the full feature set of the device. The FWH interface is equipped to operate at 33Mhz, synchronous with the PCI bus.

1.1.2. Address/Address Multiplexed Interface

For information regarding the A/A Mux interface, please consult the Intel® 82802 Firmware Hub PROM Programmer Architecture Specification. The A/A Mux refers to the multiplexed row and column addresses in this interface. This approach is required so the device can be tested and programmed quickly with automated test equipment (ATE) and PROM programmers in the OEM's manufacturing flow. This interface also allows the device to have an efficient programming interface with potentially large future densities, while still fitting into a 32-pin package. Only basic reads, programming, and erase of the nonvolatile memory blocks can be performed through the A/A Mux interface. In this mode FWH features, security features and registers are unavailable. A row/column (R/C#) pin determines which set of addresses "rows or columns" are latched.

1.2. Nonvolatile Flash Memory Core

A feature of the FWH component is nonvolatile memory core based on Intel® flash technology. The high-performance memory is arranged in eight (4-Mbit device), or sixteen (8-Mbit device) 64-Kbyte blocks.

Intel flash technology enables fast factory programming and low power designs. Specifically designed for 3v systems, this component supports read operations at 3.3v V_{CC} and block erase and program operations at 3.3v and 12v V_{PP}. The 12v V_{PP} option renders the fastest program performance which will increase your factory throughput, but is not recommended for standard in-system FWH operation in the platform, due to an 80hr limit for 12v on the V_{PP} pin over the lifetime of the device programming or not. With the 3.3v V_{PP} option, V_{CC} and V_{PP} should be tied together for a simple, low-power 3v design. In addition to the voltage flexibility, the dedicated V_{PP} pin gives complete data protection when $V_{PP} \leq V_{PPLK}$. Internal V_{PP} detection circuitry automatically configures the device for block erase and program operations. Note that, while current for 12v programming will be drawn from V_{PP}, 3.3v programming board solutions should design such that V_{PP} draws from the same supply as V_{CC}, and should assume that full programming current may be drawn from either pin.

Figure 1-1. Device Memory Map with FWH Hardware Lock Architecture

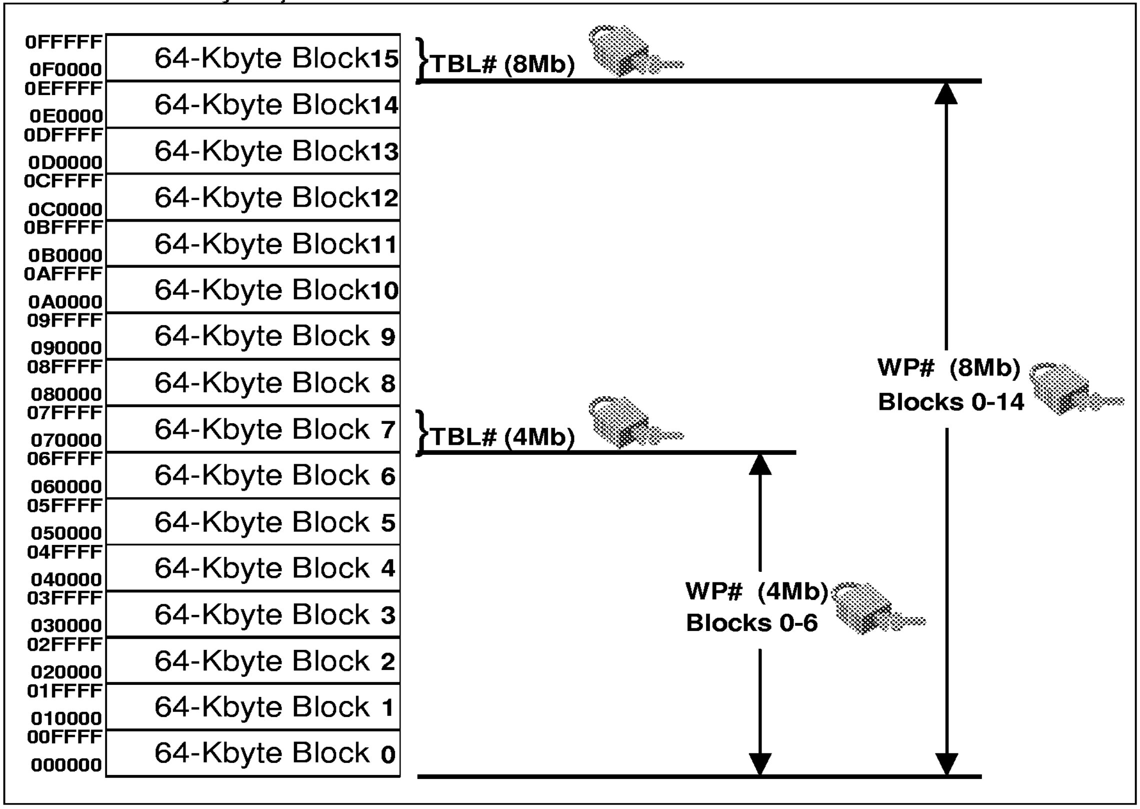
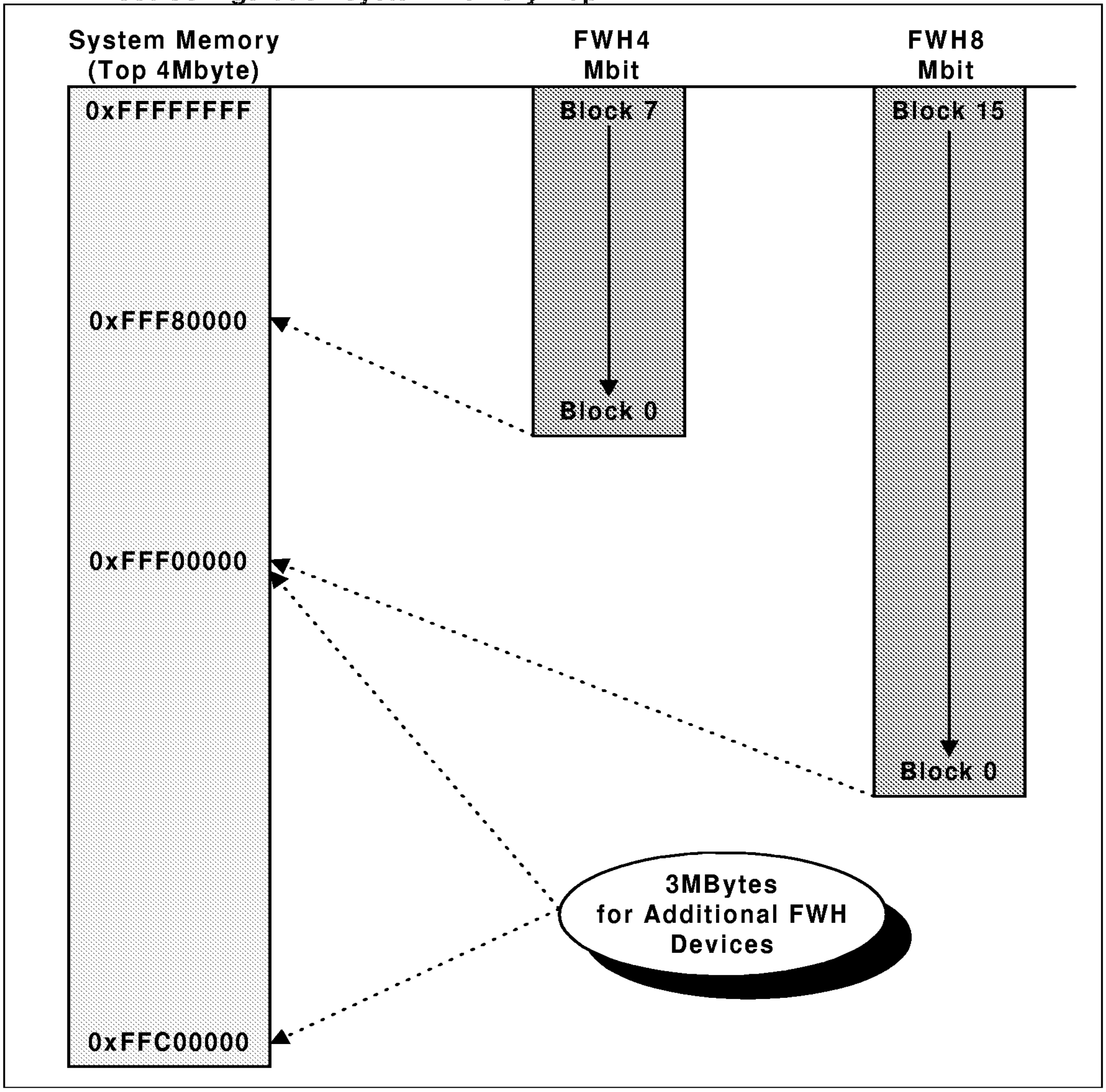


Figure 1-2. FWH Boot-Configuration System Memory Map



2. Pinout Configurations

Figure 2-1. 32-Lead PLCC Firmware Hub Pin-out

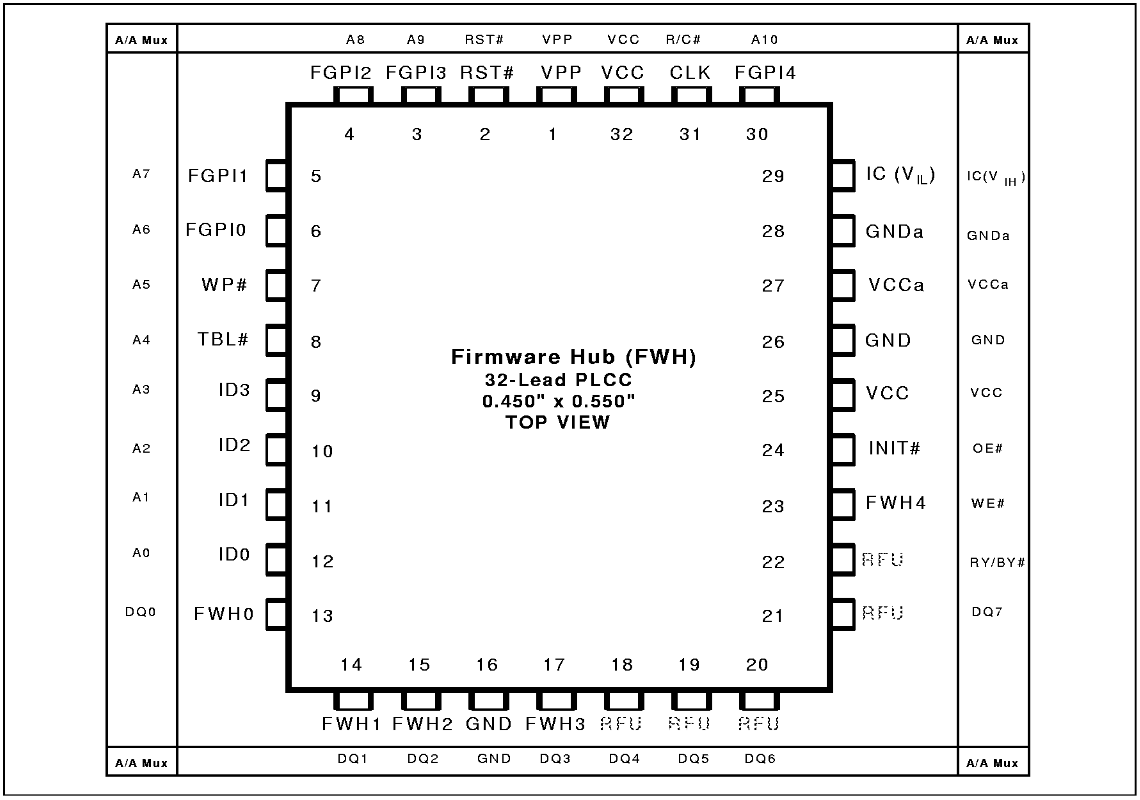
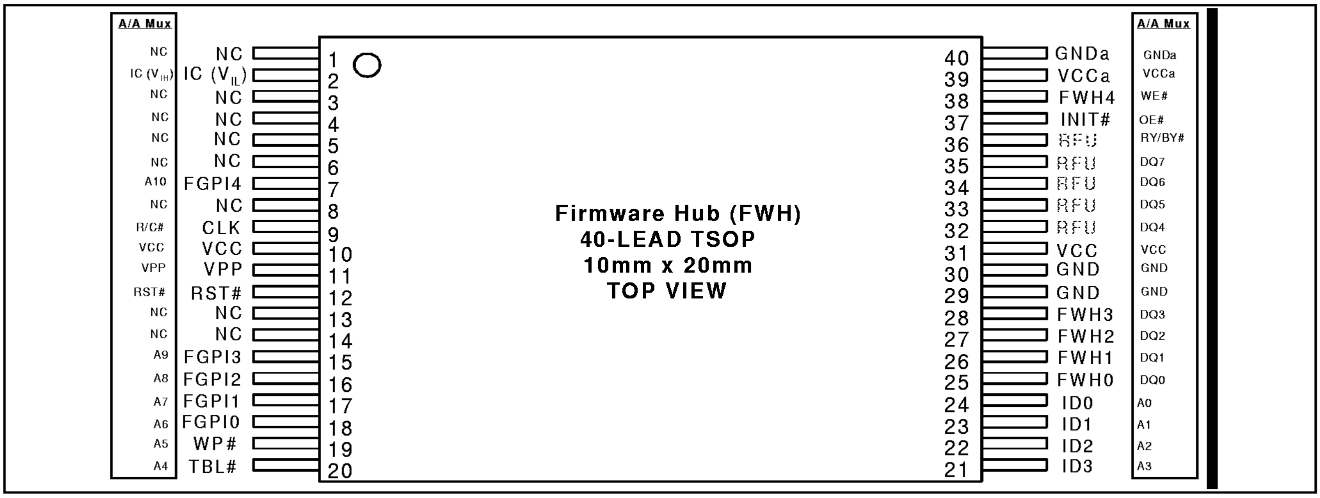


Figure 2-2. 40-Lead TSOP Firmware Hub Pin-out



2.1. Pin Descriptions

The pin descriptions table details the usage of each of the device pins. Most of the pins have dual functionality, with functions in both the Firmware Hub and A/A Mux interfaces. A/A Mux functionality for pins is shown **bold** in the description box for that pin. All pins are designed to be compliant with max of VCC +0.3v, unless otherwise noted.

Table 2-1. Pin Descriptions

Symbol	Type	Interface		Name and Function
		FWH	A/A Mux	
IC	INPUT	X	X	**INTERFACE CONFIGURATION PIN:** This pin determines which interface is operational. This pin is held high to enable the A/A Mux interface. This pin is held low to enable the FWH interface. This pin must be set at power-up or before return from reset and not changed during device operation. This pin is pulled down with an internal resistor of between 20 and 100 KΩ. With IC high (A/A Mux mode), this pin will exhibit a leakage current of approximately 200µA. This pin may be floated, which will select FWH mode.
RST#	INPUT	X	X	**INTERFACE RESET:** Valid for both A/A Mux and FWH to interface operations. When driven low, RST# inhibits write operations to provide data protection during power transitions, resets internal automation, and tri-states pins FWH[3:0] (only in FWH interface). RST#-high enables normal operation. When exiting from reset, the device defaults to read array mode.
INIT#	INPUT	X		**CPU RESET:** This is a second reset pin for in-system use. This pin is internally combined with the RST# pin; If this pin or RST# is driven low, identical operation is exhibited. This signal is designed to be connected to the chipset INIT signal (max=2.5v). **A/A Mux = OE#**
CLK	INPUT	X		**33MHz CLOCK for FWH INTERFACE:** This input is the same as the PCI clock and adheres to the PCI specification. **A/A Mux = R/C#**
FWH[3:0]	I/O	X		**FWH I/Os:** I/O Communication. **A/A Mux = DQ[3:0]**
FWH4	INPUT	X		**FWH Input:** Input Communication. **A/A Mux = WE#**
ID[3:0]	INPUT	X		**IDENTIFICATION INPUTS:** These four pins are part of the mechanism that allows multiple parts to be attached to the same bus. The strapping of these pins is used to identify the component. The boot device must have ID[3:0]=0000 and it is recommended that all subsequent devices should use a sequential up-count strapping (i.e.0001, 0010,0011, etc.). These pins are pulled down with internal resistors, with values between 20 and 100 KΩ, when in FWH mode. Any ID pins that are pulled high will exhibit a leakage current of approximately 200µA. Any pins intended to be low may be left to float. In a single FWH system, all may be left floated. **A/A Mux = A[3:0]**

Symbol	Type	Interface		Name and Function
		FWH	A/A Mux	
FGPI[4:0]	INPUT	X		**FWH GENERAL PURPOSE INPUTS:** These individual inputs can be used for additional board flexibility. The state of these pins can be read through FWH registers. These inputs should be at their desired state before the start of the PCI clock cycle during which the read is attempted, and should remain in place until the end of the read cycle. They may *only* be used for 3.3v signals. Unused FGPI pins must not be floated. **A/A Mux = A[10:6]**
TBL#	INPUT	X		**TOP BLOCK LOCK:** When low, prevents programming or block erase to the highest addressable block 7 in a 4M, 15 in an 8M), regardless of the state of the lock register. TBL#-high disables hardware write protection for the top block, though register-based protection still applies. The status of TBL# does not affect the status of block-locking registers. **A/A Mux = A4**
WP#	INPUT	X		**WRITE PROTECT:** When low, prevents programming or block erase to all but the highest addressable blocks 0-6 in a 4M, 0-14 in an 8M), regardless of the state of the lock register. WP#-high disables hardware write protection for these blocks, though register-based protection still applies. The status of TBL# does not affect the status of block-locking registers. **A/A Mux = A5**
A_0–A_{10}	INPUT		X	**LOW-ORDER ADDRESS INPUTS:** Inputs for low-order addresses during read and write operations. Addresses are internally latched during a write cycle. For the A/A Mux interface these addresses are latched by R/C# and share the same pins as the high-order address inputs.
DQ_0–DQ_7	I/O		X	**DATA INPUT/OUTPUTS:** These pins receive data and commands during CUI write cycles and transmit data during memory array, status register, and identifier code read cycles. Data pins float to high-impedance when the chip is deselected or outputs are disabled. Data is internally latched during a write cycle.
OE#	INPUT		X	**OUTPUT ENABLE:** Gates the device's outputs during a read cycle
R/C#	INPUT		X	**ROW-COLUMN ADDRESS SELECT:** For the A/A Mux interface, this pin determines whether the address pins are pointing to the row addresses, A0-A10, or to the column addresses, A11-A19.
WE#	INPUT		X	**WRITE ENABLE:** Controls writes to the CUI and array blocks. Addresses and data are latched on the rising edge of the WE# pulse.
V_{PP}	SUPPLY	X	X	**BLOCK ERASE PROGRAM POWER SUPPLY:** For erasing array blocks or programming data. V_{PP} = 3.3V or 12V V_{PP} . With $V_{PP} \leq V_{PPLK}$, memory contents cannot be altered. Block erase and program with an invalid V_{PP} (see DC Characteristics) produce spurious results and should not be attempted. V_{PP} may only be held at 12v for 80 hours over the lifetime of the device.

Symbol	Type	Interface		Name and Function
		FWH	A/A Mux	
V_{CC}	SUPPLY	X	X	**DEVICE POWER SUPPLY:** Internal detection automatically configures the device for optimized read performance. Do not float any power pins. With $V_{CC} \leq V_{LKO}$, all write attempts to the flash memory are inhibited. Device operations at invalid V_{CC} voltages (see DC Characteristics) produce spurious results and should not be attempted.
GND	SUPPLY	X	X	**GROUND:** Do not float any ground pins.
V_{CCa}	SUPPLY	X	X	**ANALOG POWER SUPPLY:** This supply does not need to be isolated from Vcc which supplies the core digital voltage. These two lines can, and should, share the same system supply.
GND_a	SUPPLY	X	X	**ANALOG GROUND:** Should be tied to same plane as GND.
RFU		X		**RESERVED for FUTURE USE:** These pins are reserved for future generations of this product and should be connected accordingly. These pins may be left disconnected or driven. If driven, voltage levels should meet V_{IH} and V_{IL} requirements. A/A Mux = DQ[7:4]
NC		X	X	**NO CONNECT:** Pin may be driven or floated. If driven, voltage levels should meet V_{IH} and V_{IL}. No connects appear only on the 40ld TSOP package.

3. *Interface Operation Description*

3.1. Read

Memory information, identifier codes, GPI registers, or the status register can be read independent of the V_{PP} voltage. Commands using the read mode include the following functions: reading memory from the array, reading the identifier codes, reading the status register, reading the lock bit registers, reading the random number generator, reading the GPI registers and reading the random number status register. Upon initial device power-up or after exit from reset mode, the device automatically resets to read array mode.

3.2. Write

The ICH writes to the CUI, by initiating a write through the FWH interface. The CUI does not occupy a specific memory location.

3.3. Output Disable

When the FWH is not selected through a FWH read or write cycle, the FWH interface outputs (FWH[3:0]) are disabled and will be placed in a high-impedance state.

3.4. Reset

RST# or INIT# at V_{IL} initiates a device reset. In read mode, RST# or INIT# low deselects the memory, places output drivers in a high-impedance state, and turns off all internal circuits. RST# or INIT# must be held low for time t_{PLPH} (A/A Mux and FWH operation). The FWH resets to read array mode upon return from reset, and all blocks are set to default (locked) status (see **Table 4-5**) regardless of their locked state prior to reset.

During block erase or program, driving RST# or INIT# low will abort the operation underway, in addition to causing a reset latency. Memory contents being altered are no longer valid, since the data may be partially erased or programmed.

It is important to assert RST# or INIT# during system reset. When the system comes out of reset, it will expect to read from the memory array of the device. If a system reset occurs with no FWH reset (this will be hardware dependent), it is possible that proper CPU initialization will not occur (the FWH memory may be providing status information instead of memory array data).

3.5. Hardware Write-Protect Pins TBL# and WP# Operational Effect

Two pins are available with the FWH to provide hardware write protect capabilities.

The Top Block Lock (TBL#) pin is a signal, when held low (active), prevents program or block erase operations in the top-most block of the device where critical code can be stored. When TBL# is high, hardware write protection of the top block is disabled. The Write Protect (WP#) pin serves the same function for all remaining blocks. All blocks except the top block. WP# operates independently from TBL# and does not affect the lock status of the top block. .

The TBL# and WP# pins must be set to the desired protection state prior to starting a program or erase operation since they are sampled at the beginning of the operation. Changing the state of TBL# or WP# during a program or erase operation may cause unpredictable results.

If the state of TBL# or WP# changes during a program suspend or erase suspend state, the changes to the device's locking status do not take place immediately. The suspended operation may be resumed to successfully complete the program or erase operation. The new lock status will take place after the program or erase operation completes.

These pins function in combination with the register-based block locking described in Section 4.9. These pins, when active, will write-protect the appropriate block(s), regardless of the associated block locking registers. (Example: when TBL# is active, writing to the top block is prevented, regardless of the state of the write-lock bit for the top block's locking register. In such a case, clearing the write-protect bit in the register will have no functional effect, even though the register may indicate that the block is no longer locked. The register may still be set to read-lock the block, if desired.) See Section 4.9 for further information.

4. Functional Descriptions

When the V_{PP} voltage $\leq V_{PPLK}$, read operations from the status register, identifier codes, or memory are enabled, but programming and erase functions are disabled. Placing $V_{PPH1/2}$ on V_{PP} enables successful block erase and program operations.

Table 4-1. Command Definitions

Command	Bus Cycles Req'd.	Notes	First Bus Cycle			Second Bus Cycle		
			Oper	Addr(1)	Data(2)	Oper	Addr(1)	Data(2)
Read Array/Reset	1		Write	X	FFH			
Read Identifier Codes	≥ 2	2	Write	X	90H	Read	IA	ID
Read Status Register	2		Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Block Erase	2	3	Write	BA	20H	Write	BA	D0H
Program	2	3,4	Write	WA	40H or 10H	Write	WA	WD
Block Erase and Program Suspend	1	3	Write	X	B0H			
Block Erase and Program Resume	1	3	Write	X	D0H			

NOTES:

1.

X = Any valid address within the device.
IA = Identifier Code Address.
BA = Address within the block being erased.
WA = Address of memory location to be written.
SRD = Data read from status register.
WD = Data to be written at location WA.
ID = Data read from identifier codes.

2. Following the Read Identifier Codes command, read operations access manufacturer and device. See Table 4-3. Identifier Codes for read identifier code data.
3. The block must be not be write locked when attempting block erase or program operations. Attempts to issue a block erase or program to a write locked block will fail.
4. Either 40H or 10H are recognized by the WSM as the program setup.

Note: Commands other than those shown above are reserved by Intel for future device implementations and should not be used.

Table 4-2. Status Register Definition

WSMS	ESS	ES	PS	VPPS	PSS	DPS	R
7	6	5	4	3	2	1	0

	NOTES:
SR.7 = WRITE STATE MACHINE STATUS 1 = Ready 0 = Busy	Check SR.7 to determine block erase or program completion. SR.6–0 are invalid while SR.7 = "0."
SR.6 = ERASE SUSPEND STATUS 1 = Block Erase Suspended 0 = Block Erase in Progress/Completed	
SR.5 = ERASE STATUS 1 = Error in Block Erasure 0 = Successful Block Erase	If both SR.5 and SR.4 are "1"s after a block erase attempt, an improper command sequence was entered.
SR.4 = PROGRAM STATUS 1 = Error in Program 0 = Successful Program	
SR.3 = V_{PP} STATUS 1 = V_{PP} Low Detect, Operation Abort 0 = V_{PP} OK	SR.3 does not provide a continuous indication of V_{PP} level. The WSM interrogates and indicates the V_{PP} level only after a block erase or program operation. SR.3 is not guaranteed to reports accurate feedback only when $V_{PP} \neq V_{PPH1/2}$.
SR.2 = PROGRAM SUSPEND STATUS 1 = Program Suspended 0 = Program in Progress/Completed	
SR.1 = DEVICE PROTECT STATUS 1 = Write-lock bit, TBL# pin, or WP# pin Detected, Operation Abort 0 = Unlock	SR.1 does not provide a continuous indication of write-lock bit, TBL# pin, or WP# pin values. The WSM interrogates the write-lock bit, TBL# pin, or WP# pin only after a block erase or program operation. It informs the system, depending on the attempted operation, if the selected block is locked.
SR.0 = RESERVED FOR FUTURE ENHANCEMENTS	SR.0 is reserved for future use and should be masked out when polling the status register.

4.1. Read Array Command

Upon initial device power-up and after exit from reset, the device defaults to read array mode. This operation is also initiated by writing the Read Array command. The device remains enabled for reads until another command is written. Once the internal write state machine (WSM) has started a block erase or program, the device will not recognize the Read Array command until the WSM completes its operation unless the WSM is suspended via an Erase Suspend or Program Suspend command. The Read Array command functions independently of the V_{PP} voltage.

4.2. Read Identifier Codes Command

The identifier code operation is initiated by writing the Read Identifier Codes command. Following the command write, read cycles from addresses shown in Table 4-3. Identifier Codes, retrieve the manufacturer and device codes (identifier code values are also shown in Table 4-3. Identifier Codes). To terminate the read identifier code operation, write another valid command to the FWH. The Read Identifier Codes command functions independently of the V_{PP} voltage.

Table 4-3. Identifier Codes

Code		Address	Data
Manufacturer Code		000000	89
Device Code	4-Mbit	000001	AD
Device Code	8-Mbit	000001	AC

4.3. Read Status Register Command

The status register may be read to determine when a block erase or program completes and whether the operation completed successfully. The status register may be read at any time by writing the Read Status Register command. After writing this command, all subsequent read operations will return data from the status register until another valid command is written. The Read Status Register command functions independently of the V_{PP} voltage.

4.4. Clear Status Register Command

Error flags in the status register can only be set to “1”s by the WSM and can only be reset by the Clear Status Register command. These bits indicate various failure conditions. The Clear Status Register command functions independently of the applied V_{PP} voltage.

4.5. Block Erase Command

The erase command operates on one block at a time. This command requires an (arbitrary) address within the block to be erased. Recall that erase changes all block data to OXFF. Block preconditioning, erase, and verify are handled internally by the WSM, which is invisible to the system. After issuing the erase command, the device automatically outputs status register data when read. When the block erase completes, the status register can be checked. If the FWH detects a block erase error, the status register should be cleared before system software attempts corrective actions. The CUI remains in read status register mode until a new command is issued.

Successful block erase requires that the corresponding block’s write-lock-bit be cleared, and the corresponding write-protect pin (TBL# or WP#) be inactive. If block erase is attempted when the block is locked, the block erase will fail.

Successful block erase only occurs when $V_{PP} = V_{PPH1}$ or V_{PPH2}. If the erase operation is attempted at $V_{PP} \neq V_{PPH1}$ or V_{PPH2} erratic results may occur.

4.6. Program Command

Program command operates on one byte at a time. This command specifies the address and data to be programmed. After the CUI receives the command, the WSM takes over, controlling the program and verify algorithms internally. After the program command is written, the device automatically outputs status register data when read. When programming is complete, the status register can be checked. If a program error is detected, the status register should be cleared. The internal WSM verify only detects errors for "1"s that do not successfully program to "0"s. The CUI remains in read status register mode until it receives another command.

Reliable programming only occurs when $V_{PP} = V_{PPH1}$ or V_{PPH2}. If the program operation is attempted at $V_{PP} \neq V_{PPH1}$ or V_{PPH2} erratic results may occur.

A successful program operation also requires that the corresponding block's write-lock-bit be cleared, and the corresponding write-protect pin (TBL# or WP#) be inactive. If a program operation is attempted when the block is locked, the operation will fail.

4.7. Block Erase Suspend Command

The Block Erase Suspend command allows block-erase interruption to read or program data in another block of memory. Once the block erase process starts, writing the Block Erase Suspend command requests that the WSM suspend the block erase sequence at a predetermined point in the algorithm. The device outputs status register data when read after the Block Erase Suspend command is written. Polling status register can help determine when the block erase operation has been suspended.

After a successful suspend, a Read Array command can be written to read data from blocks other than that which is suspended. A Program command sequence can also be issued during erase suspend to program data in blocks other that the block currently in erase suspend mode.

The other valid commands while block erase is suspended include Read Status Register and Block Erase Resume. After a Block Erase Resume command is written, the WSM will continue the block erase process. V_{PP} must remain at $V_{PPH1/2}$ (the same V_{PP} level used for block erase) while block erase is suspended. RST# or INIT# must also remain at V_{IH}. Block erase cannot resume until program operations initiated during block erase suspend have completed.

4.8. Program Suspend Comand

The Program Suspend command allows program interruption to read data in other memory locations. Once the program process starts, writing the Program Suspend command requests that the WSM suspend the program sequence at a predetermined point in the algorithm. The device continues to output status register data when read after the Program Suspend command is written. Polling status register bits will help determine when the program operation has been suspended.

After a successful suspend, a Read Array command can be written to read data from locations other than that which is suspended. The only other valid commands while program is suspended are Read Status Register and Program Resume. After Program Resume command is written, the WSM will continue the programming process. V_{PP} must remain at $V_{PPH1/2}$ (the same V_{PP} level used for program) while in program suspend mode. RST# or INIT# must also remain at V_{IH}.

4.9. Register Based Locking and General Purpose Input Registers

A series of registers are available in the FWH to provide software read- and write-locking and GPI feedback. These registers are accessible through standard addressable memory space (see Table 4-4). Each block has three dedicated locking bits described in

It is recommended that the GPI pins be in the desired state before FWH4 is brought low for the beginning of the next bus cycle, and remain in that state until the end of the cycle.

Table 4-4. Firmware Hub Register Configuration Map

Memory Address	Mnemonic	Register Name	Default	Type
0xFFBF0002	T_BLOCK_LK	Top Block Lock Register (4-8Mbit FWH)	01h	R/W
0xFFBE0002	T_MINUS01_LK	Top Block [-1] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFBD0002	T_MINUS02_LK	Top Block [-2] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFBC0002	T_MINUS03_LK	Top Block [-3] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFBB0002	T_MINUS04_LK	Top Block [-4] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFBA0002	T_MINUS05_LK	Top Block [-5] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFB90002	T_MINUS06_LK	Top Block [-6] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFB80002	T_MINUS07_LK	Top Block [-7] Lock Register (4-8Mbit FWH)	01h	R/W
0xFFB70002	T_MINUS08_LK	Top Block [-8] Lock Register (8Mbit FWH)	01h	R/W
0xFFB60002	T_MINUS09_LK	Top Block [-9] Lock Register (8Mbit FWH)	01h	R/W
0xFFB50002	T_MINUS10_LK	Top Block [-10] Lock Register (8Mbit FWH)	01h	R/W
0xFFB40002	T_MINUS11_LK	Top Block [-11] Lock Register (8Mbit FWH)	01h	R/W
0xFFB30002	T_MINUS12_LK	Top Block [-12] Lock Register (8Mbit FWH)	01h	R/W
0xFFB20002	T_MINUS13_LK	Top Block [-13] Lock Register (8Mbit FWH)	01h	R/W
0xFFB10002	T_MINUS14_LK	Top Block [-14] Lock Register (8Mbit FWH)	01h	R/W
0xFFB00002	T_MINUS15_LK	Top Block [-15] Lock Register (8Mbit FWH)	01h	R/W
0xFFBC0100	FGPI_REG	FWH General Purpose Input Register	N/A	RO

Table 4-5. T_BLOCK_LK and T_MINUSxx_LK (Block Locking Registers)

Memory Address:	FFBx0002h (x = f-0h)
Default Value:	01h
Access:	Read/Write
Size:	8 bits (each)

Bit	Function
7:3	**Reserved**
2	**Read-Lock** 1 = Prevents read operations in the block where set. 0 = Normal operation for reads in the block where clear. This is the default state.
1	**Lock-Down** 1 = Prevents further set or clear operations to the Write Lock and Read Lock bits. Lock-Down can only be set, but not cleared. The block will remain locked-down until reset (with RST# or INIT#), or until the device is power-cycled. 0 = Normal operation for Write Lock and Read Lock bits altering in the block where clear. This is the default state.
0	**Write-Lock** 1 = Prevents program or erase operations in the block where set. This is the default state. 0 = Normal operation for programming and erase in the block where clear.

Table 4-6. Register Based Locking Value Definitions

Data	Reserved Data 7-3	Read Lock, Data 2	Lock-Down, Data 1	Write Lock, Data 0	Resulting block state (1).
0x00	00000	0	0	0	Full access
0x01	00000	0	0	1	Write locked. **Default state at power-up**
0x02	00000	0	1	0	Locked Open (Full access Locked down)
0x03	00000	0	1	1	Write Locked down
0x04	00000	1	0	0	read locked
0x05	00000	1	0	1	read and write locked
0x06	00000	1	1	0	read locked down
0x07	00000	1	1	1	read and write locked down.

Note: The write-lock bit must be set to the desired protection state prior to starting a program or erase operation since it is sampled at the beginning of the operation. Changing the state of the write-lock bit during a program or erase operation may cause unpredictable results. If the state of the write-lock bit changes during a program suspend or erase suspend state, the changes to the block's locking status do not take place immediately. The suspended operation may be resumed successfully. The new lock status will take place after the program or erase operation completes. The individual bit functions are described in the following sections.

4.9.1. Write Lock

The default write status of all blocks upon power-up is write-locked. Any program or erase operations attempted on a locked block will return an error in the status register (indicating block lock.) The status of the locked block can be changed to unlocked by clearing the write-lock bit, provided the lock-down bit is not also set. Current write-lock status of a particular block can be determined by reading the corresponding write-lock bit. The write-lock functions in conjunction with the hardware write-lock pins, TBL# and WP#. When active, these pins take precedence over the register locking function and write-lock the top block or remaining blocks respectively. Reading this register will not read the state of the TBL# or WP# pins.

4.9.2. Read Lock

The default read status of all blocks upon power-up is read-unlocked. When a block's read-lock bit is set, data can not be read from that block. An attempted read from a read-locked block will result in data 0x00. The read-lock status can be unlocked by clearing the read-lock bit, provided the lock-down bit has not been set. Current read-lock status of a particular block can be determined by reading the corresponding read-lock bit.

4.9.3. Lock-Down

When in the FWH interface mode, the default lock-down status of all blocks upon power-up is not-locked-down. The lock-down bit for any block may be set, but only once, as future attempted changes to that block locking register will be ignored. The lock-down bit is only cleared upon a device reset with RST# or INIT#. Current lock-down status of a particular block can be determined by reading the corresponding lock-down bit. Once a block's lock-down bit is set, the read- and write-lock bits for that block can no longer be modified, and the block is locked-down in its current state of read and write accessibility.

4.9.4. General Purpose Inputs Register

This register reads the status of the FGPI[4:0] pins on the FWH. Since this is a pass-through register, there is no default value, only the state of the pins at power-up.

Table 4-7. GPI_REG (General Purpose Inputs Register)

Memory Address:	FFBC0100h
Default Value:	N/A
Access:	R0
Size:	8 bits

Bit	Function
7:5	**Reserved**
4	**FGPI[4]** Reads status of general purpose input pin (PLCC-30/TSOP-7)
3	**FGPI[3]** Reads status of general purpose input pin (PLCC-3/TSOP-15)
2	**FGPI[2]** Reads status of general purpose input pin (PLCC-4/TSOP-16)
1	**FGPI[1]** Reads status of general purpose input pin (PLCC-5/TSOP-17)
0	**FGPI[0]** Reads status of general purpose input pin (PLCC-6/TSOP-18)

4.10. Random Number Generator

The Firmware Hub integrates a Random Number Generator (RNG) using thermal noise generated from inherently random quantum mechanical properties of silicon. When not generating new random bits the RNG circuitry will enter a low power state. Intel will provide a binary software driver to give third party software access to our RNG for use as a security feature. At this time, the RNG is only to be used with a system in an OS-present state.

4.11. Using Multiple FWH Components

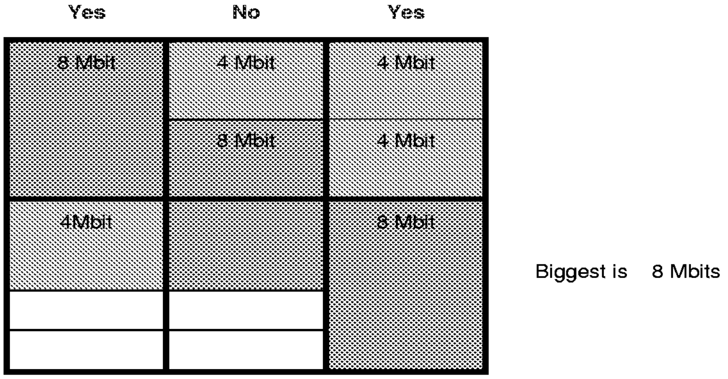
4.11.1. Selecting Appropriate FWH IDs and Densities

It is possible, using different ID strapping, to use multiple FWH components in a system. While the FWH protocol supports up to 16 FWH devices, BIOS support, bus loading, or the attaching bridge (the ICH) may limit this number. Note that regardless of the number of FWH components, the maximum "window" of FWH array that can be visible at one time is 4 MBytes. The boot device is required to have an ID (as determined by ID[0:3]) of 0. It is recommended for clarity that subsequent devices use incremental numbering.

The most straightforward method for using multiple FWH components is to use devices of equal density. This is the recommended technique.

In special applications, when use of multiple FWH components of different densities is desired (if multiple RNGs, or more GPIs are required, for instance, without the need for greater array space), IDs must be chosen such that component memory array spaces do not cross boundaries delimited by the largest-sized device as illustrated below.

For example, in a design with 8-, and 4-Mbit components, the 8-Mbit part must either be first, or must be after enough 4-Mbit parts to add up to a multiple of 8 Mbits.



4.11.2. Mapping FWH Devices into Memory Map

There is an available 4 MBytes of available memory space devoted to the FWH. The ICH, therefore, has the ability to select which FWH device maps into each region of the system address space.

In the existing ICH, the address map is broken up into eight 512K segments. The BIOS Select Register in the ICH is a 32-bit register that contains the needed mapping information, determining which FWH receives requests from which portion of the address map. As an example, in a system with four 8Mbit devices, this register would be 0x00112233. This is the default power-up state for this register. In a system with eight 4Mbit devices, the register would need to be changed to 0x01234567.

Note: The FWH indicated in the most-significant nibble of the register may be shadowed elsewhere in the system memory map. The FWH with ID 0 may not be re-mapped.

4.11.3. Paging FWH Devices for Greater Than 4 Mbytes of FWH Memory

In certain applications, even a 4MByte window of flash memory is not adequate. It is possible to exceed this amount by using a paging scheme. In such a scheme, individual FWH devices may be "swapped" in and out of system memory space. This must be implemented at the BIOS level, to permit modification of the ICH BIOS Select Register. A number of paging algorithms may successfully be used with the FWH memory space, using the ICH BIOS Select Register. This register, then, determines which FWH device gets mapped into each 512K "slice" of the system memory map. The 0th FWH (ID=0) may not be remapped. Reference the Intel® 82801AA (ICH) and Intel® 82801AB (ICH0) I/O Controller Hub Datasheet, (Order number 290655) for information on these components, and the BIOS Select Register.

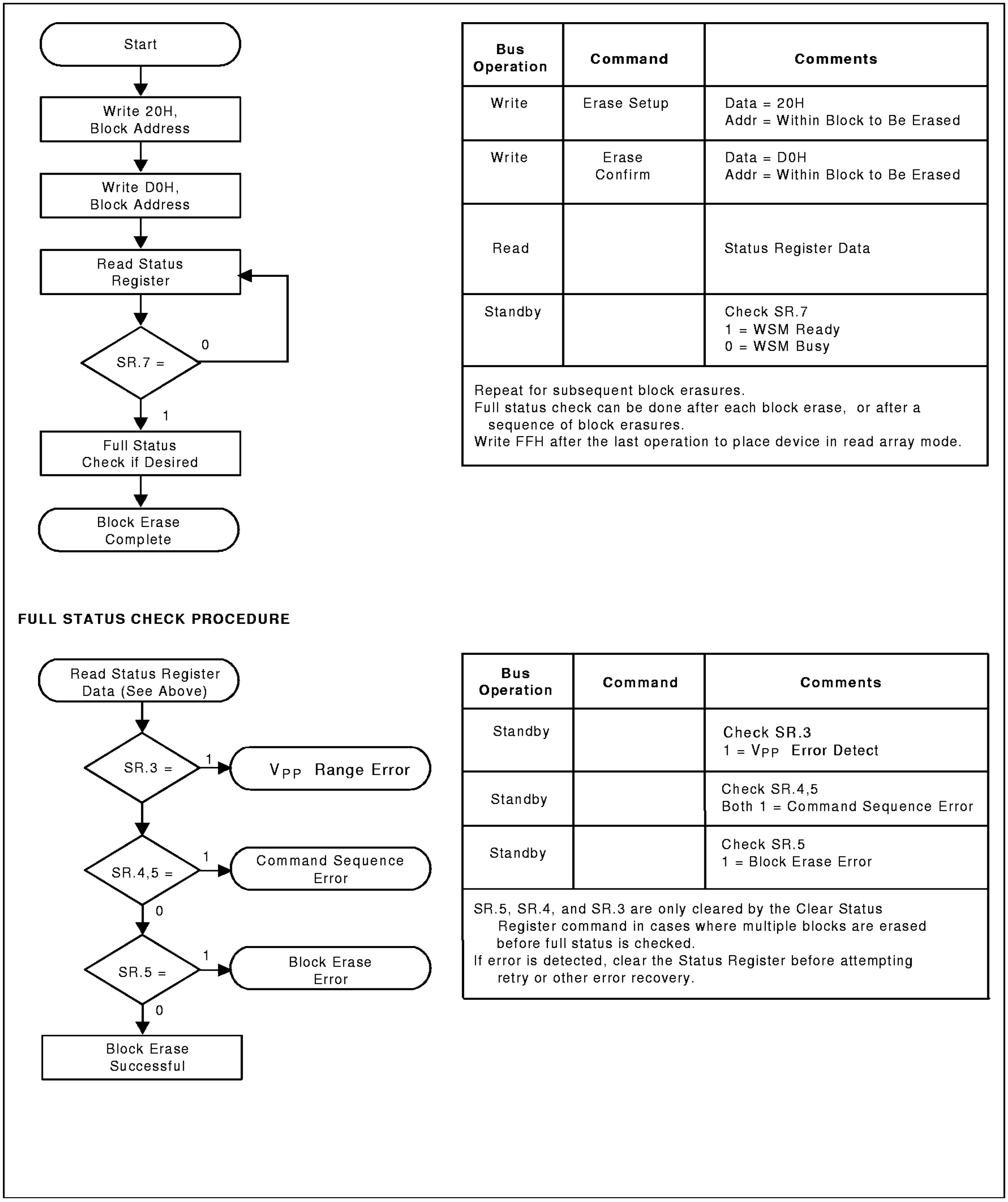
Note: The paging of FWH devices will also "page" features, potentially affecting the presence of, or location of, the FGPI register (Table 4-7). When a paging scheme is used, it is recommended that critical FPGIs be used only on the ID 0 FWH device, which must remain mapped at the top of memory.

4.11.4. Programming Multiple FWH Devices

Special considerations must be taken into account when programming multiple FWH devices in-system. Since there is no ID support in A/A Mux mode, out-of-system programming with standalone PROM programmers, or in-system programming using FWH mode is the recommended means for programming multiple devices. In cases where programming time is critical, or ATE programming is required, provisions should be made to isolate the component from its neighboring devices during A/A Mux programming, or the other devices should be held in a reset (or otherwise disabled) state until programming of the intended device is complete. Do not switch one component into A/A Mux mode, leaving the others in FWH mode.

4.12. CUI Automation Flowcharts

Figure 4-4. Automated Block Erase Flowchart



Bus Operation	Command	Comments

Write	Erase Setup	Data = 20H
		Addr = Within Block to Be Erased

Write	Erase Confirm	Data = D0H
		Addr = Within Block to Be Erased

Read		Status Register Data
Standby		Check SR.7 1 = WSM Ready 0 = WSM Busy

Repeat for subsequent block erasures.
Full status check can be done after each block erase, or after a sequence of block erasures.
Write FFH after the last operation to place device in read array mode.

FULL STATUS CHECK PROCEDURE

Bus Operation	Command	Comments
Standby		Check SR.3 1 = V_{PP} Error Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error

SR.5, SR.4, and SR.3 are only cleared by the Clear Status Register command in cases where multiple blocks are erased before full status is checked.
If error is detected, clear the Status Register before attempting retry or other error recovery.

5. Electrical Specifications

5.1. Absolute Maximum Ratings

Case Temperature under Bias............. –10°C to +85°C

Storage Temperature.......................... –65°C to +125°C

Supply Voltage with Respect to VSS........ -0.2V to 4.1V

Voltage On Any Pin
(except V_{PP}) .... –0.5V to +VCC + 0.5V[1,2]

V_{PP} Voltage ....–0.5V to +14.0V[1,2,4]

Output Short Circuit Current .... 100 mA[3]

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

Note:

1. All specified voltages are with respect to GND. Minimum DC voltage on the V_{PP} pin is –0.5V. During transitions, this level may undershoot to –2.0V for periods <20 ns. During transitions, this level may overshoot to V_{CC} +2.0V for periods <20 ns.
2. Maximum DC voltage on V_{pp} may overshoot to +14.0V for periods <20 ns.
3. Output shorted for no more than one second. No more than one output shorted at a time. This note applies to non-PCI outputs only.
4. Connection to supply of V_{HH} is allowed for a maximum cumulative period of 80 hours.

5.2. Operating Conditions

Table 5-1. Temperature and VCC

Symbol	Parameter	Notes	Min	Max	Unit	Test Condition
T_C	Operating Temperature	1	0	+85	°C	Case Temperature
V_{CC}	V_{CC} Supply Voltage (3.3V ± 0.3V)		3.0	3.6	V	

Note:

1. This temperature requirement is different from the normal commercial operating condition of Flash memories.

5.2.1. Interface DC Input/Output Specifications

Table 5-2. FWH Interface DC Input/Output Specifications

Symbol	Parameter	Conditions	Min	Max	Units	Notes
V_{IH}	Input High Voltage		0.5 V_{CC}	V_{CC} +0.5	V	3
V_{IL}	Input Low Voltage		-0.5	0.3 V_{CC}	V	3
I_{IL}	Input Leakage Current	0 < V_{in} < V_{CC}		±10	µA	1,4
V_{OH}	Output High Voltage	Iout = -500 µA	0.9 V_{CC}		V	
V_{OL}	Output Low Voltage	Iout = 1500 µA		0.1 V_{CC}	V	
C_{IN}	Input Pin Capacitance			13	pF	
C_{CLK}	CLK Pin Capacitance		3	12	pF	
L_{pin}	Recommended Pin Inductance			20	nH	2

Note:

1. Input leakage currents include hi-Z output leakage for all bi-directional buffers with tri-state outputs.
2. Refer to PCI spec.
3. Inputs are *not* "5 volt safe."
4. I_{IL} may be changed on IC and ID pins (up to 200µA) if pulled against internal pull-downs. See pin descriptions (Table 2-1).

Table 5-3. Power Supply Specifications - All Interfaces

Symbol	Parameter	Conditions	Min	Max	Units	Notes
VPPH1	VPP Voltage		3.0	3.6	V	
VPPH2	VPP Voltage		11.4	12.6	V	
VPPLK	VPP Lockout Voltage		1.5		V	
VLKO	VCC Lockout Voltage		1.5		V	
ICCSL1	VCC Standby Current (FWH Interface)	Voltage range of all inputs is VIH to VIL, FWH4 = VIH, VCC = 3.6V, CLK f = 33MHz No internal operations in progress		100	µA	2,3,4
ICCSL2	VCC Standby Current (FWH Interface)	FWH4 = VIL VCC = 3.6V, CLK f = 33MHz No internal operations in progress		10	mA	2,3,4
ICCA	VCC Active Current	VCC = VCC Max, CLK f = 33MHz Any internal operation in progress, IOUT = 0mA		67	mA	2,3,5
IPPR	VPP Read Current	VPP ≥ VCC		200	uA	2
IPPWE	VPP Program or Erase Current	VPP = 3.0-3.6V		40	mA	2
		V_{PP} = 11.4-12.6V		15	mA	2

Note:

1. All currents are in RMS unless otherwise noted. These currents are valid for all packages.
2. $V_{PP} = V_{CC}$.
3. $V_{IH} = 0.9V_{CC}$, $V_{IL} = 0.1V_{CC}$ per the PCI output V_{OH} and V_{OL} spec of Table 5-2.
4. This number is the worst case of I_{PP} + I_{CC} Memory Core + I_{CC} FWH Interface.

5.2.2. Interface AC Input/Output Specifications

Table 5-4. FWH Interface AC Input/Output Specifications

Symbol	Parameter	Condition	Min	Maz	Units	Notes
Ioh(AC)	Switching Current High	$0 < V_{OUT} \leq 0.3V_{CC}$	$-12\ V_{CC}$		mA	
		$0.3V_{CC} < V_{OUT} < 0.9V_{CC}$	$-17.1(V_{CC} - V_{OUT})$		mA	
		$0.7V_{CC} < V_{OUT} < V_{CC}$		Equation C		
	(Test Point)	$V_{OUT} = 0.7V_{CC}$		$-32\ V_{CC}$	mA	
Iol(AC)	Switching Current Low	$V_{CC} > V_{OUT} \geq 0.6V_{CC}$	$16\ V_{CC}$		mA	
		$0.6V_{CC} > V_{OUT} > 0.1V_{CC}$	$-17.1(V_{CC} - V_{OUT})$		mA	
		$0.18V_{CC} > V_{OUT} > 0$		Equation D		
	(Test Point)	$V_{OUT} = 0.18V_{CC}$		$38\ V_{CC}$	mA	
Icl	Low Clamp Current	$-3 < V_{IN} \leq -1$	$-25+(V_{IN}+1)/0.015$		mA	
Ich	High Clamp Current	$V_{CC} +4> V_{IN} \geq V_{CC}+1$	$25+(V_{IN}-V_{CC}-1)/0.015$		mA	
slewr	Output Rise Slew Rate	$0.2V_{CC} - 0.6V_{CC}$ load	1	4	V/ns	1
slewf	Output Fall Slew Rate	$0.6V_{CC} - 0.2V_{CC}$ load	1	4	V/ns	1

Note:

1. PCI specification output load is used.

5.2.3. FWH Interface AC Timing Specifications

5.2.3.1. Clock Specification

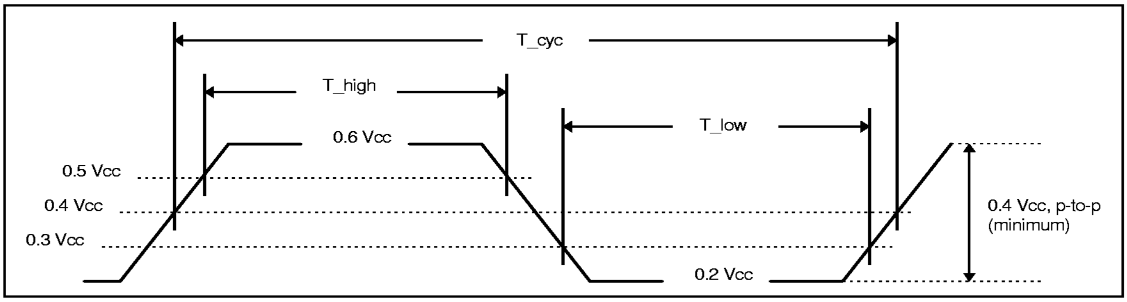
Table 5-5. Clock Specification

Symbol	Parameter	Condition	Min	Max	Units	Notes
tcyc	CLK Cycle Time		30	∞	ns	1
thigh	CLK High Time		11		ns	
tlow	CLK Low Time		11		ns	
-	CLK Slew Rate	peak-to-peak	1	4	V/ns	
-	RST# or INIT# Slew Rate		50		mV/ns	2

Note:

1. PCI components must work with any clock frequency between nominal DC and 33 MHz. Frequencies less than 16 MHz may be guaranteed by design rather than testing. Refer to PCI spec.
2. Applies only to rising edge of signal. See Chapter 4 of the PCI electrical specification.

Figure 5-1. Clock Waveform



5.2.3.2. Signal Timing Parameters

Table 5-6. Signal Timing Parameters

Symbol	PCI Symbol	Parameter	Condition	Min	Max	Units	Notes
TCHQV	t_{val}	CLK to Data Out		2	11	ns	1
TCHQX	t_{on}	CLK to Active (Float to Active Delay)		2		ns	2
TCHQZ	t_{off}	CLK to Inactive (Active to Float Delay)			28	ns	2
TAVCH TDVCH	t_{su}	Input Set-up Time		7		ns	3
TCHAX TCHDX	t_h	Input Hold Time		0		ns	3
TVSPL	t_{rst}	Reset Active Time after Power Stable		1		mS	
TCSPL	$t_{rst\text{-}clk}$	Reset Active Time after CLK Stable		100		µS	
TPLQZ	$t_{rst\text{-}off}$	Reset Active to Output Float Delay			48	ns	2

Note:

1. Minimum and maximum times have different loads. See PCI spec.
2. For purposes of Active/Float timing measurements, the Hi-Z or “off” state is defined to be when the total current delivered through the component pin is less than or equal to the leakage current specification.
3. This parameter applies to any input type (excluding CLK) .

Figure 5-2. Output Timing Parameters

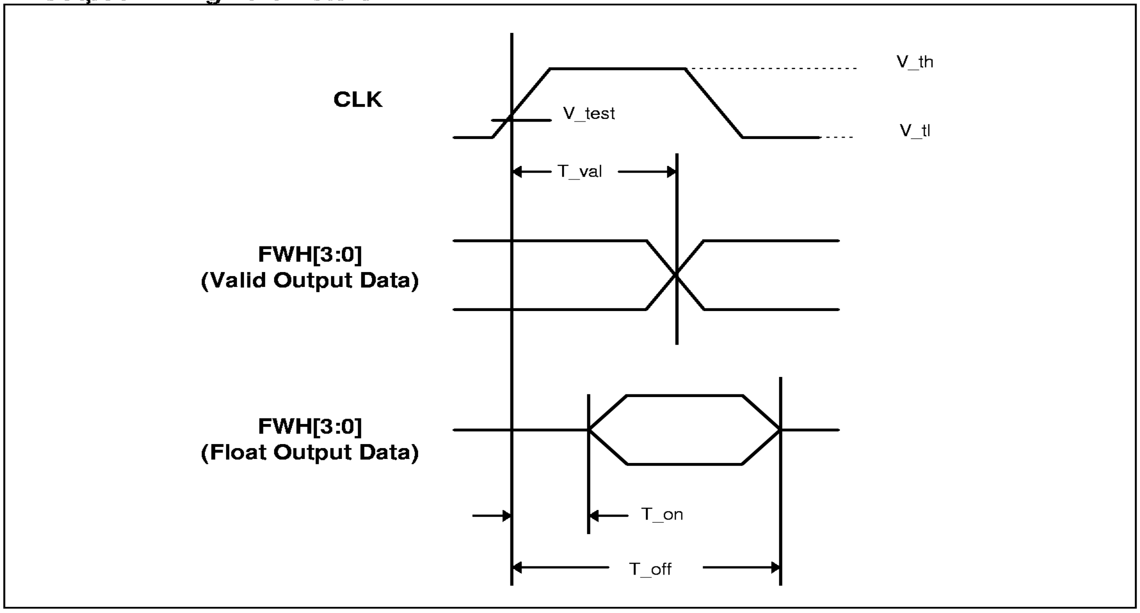


Figure 5-3. Input Timing Parameters

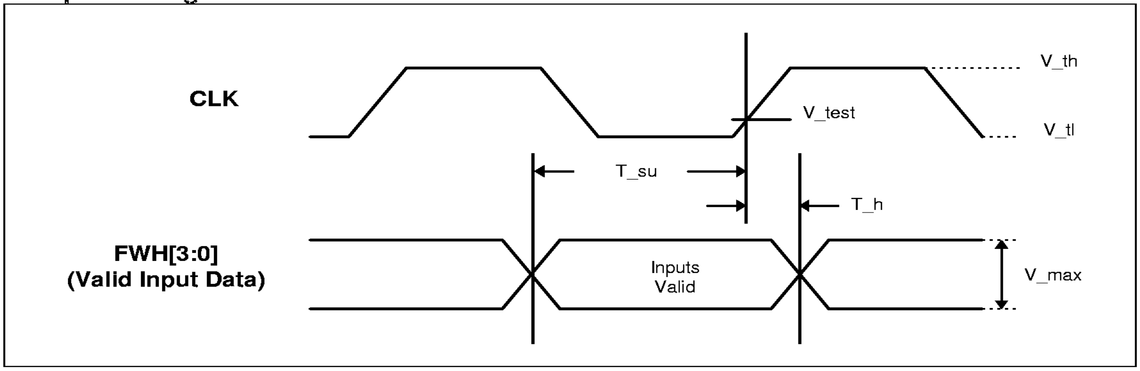


Table 5-7. Interface Measurement Condition Parameters

Symbol	Value	Units	Notes
V_{th}	0.6 V_{CC}	V	1
V_{tl}	0.2 V_{CC}		1
V_{test}	0.4 V_{CC}		
V_{max}	0.4 V_{CC}		1
Input Signal Edge Rate	1 V/ns		

Note:

1. The input test environment is done with 0.1V_{CC} of overdrive over V_{IH} and V_{IL}. Timing parameters must be met with no more overdrive than this. Vmax specifies the maximum peak-to-peak waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

5.2.4. Reset Operations

Figure 5-4. AC Waveform for Reset Operation

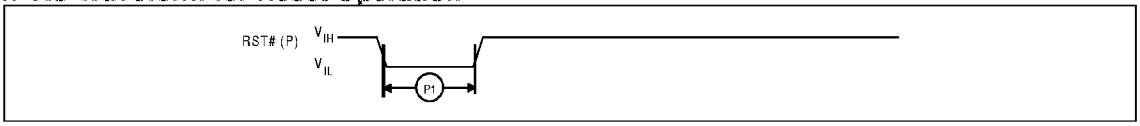


Table 5-8. AC Waveform for Reset Operation

#	Symbol	Parameter	Min	Max	Unit	Notes
P1[1]	t_{PLPH}	RST# or INIT# Pulse Low Time (If RST# or INIT# is tied to V_{CC}, this specification is not applicable)	100		ns	1

Note: A reset latency of 20us will occur if a reset procedure is performed during a programming or erase operation.

5.3. Block Programming Times

Table 5-9. Programming Times

Parameter	Notes	3.3 V V_{PP}		12 V V_{PP}		Unit
		Typ[1]	Max	Typ[1]	Max	
Byte Program Time	2	17	300	7.0	125	µs
Block Program Time	2	1.1	4.0	0.5	1.5	sec
Block Erase Time	2	0.8	6.0	0.3	4.0	sec

Note:

1. Typical values measured at T_A = +25°C and nominal voltages.
2. Excludes system-level overhead.