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

Datasheet

November 2000

Document Number: 290658-004

Contents

Figures

Tables

Revision History

Rev.	Draft/Changes	Date
-001	• Initial Release	April 1999
-002	• Added Chapter 6 • Updated programmer vendor/service provider information.	May 1999
-003	• Changed V_{IH} min. spec to reflect actual value. • Updated programmer vendor/service provider information. • Clarification of part numbering. • Spec now includes all known issues from all densities/lithographies. • Included FWH memory cycle and RNG information.	May 2000
-004	• Removed All references to multi-byte read cycles • Added DC Characteristics for A/A Mux mode	November 2000

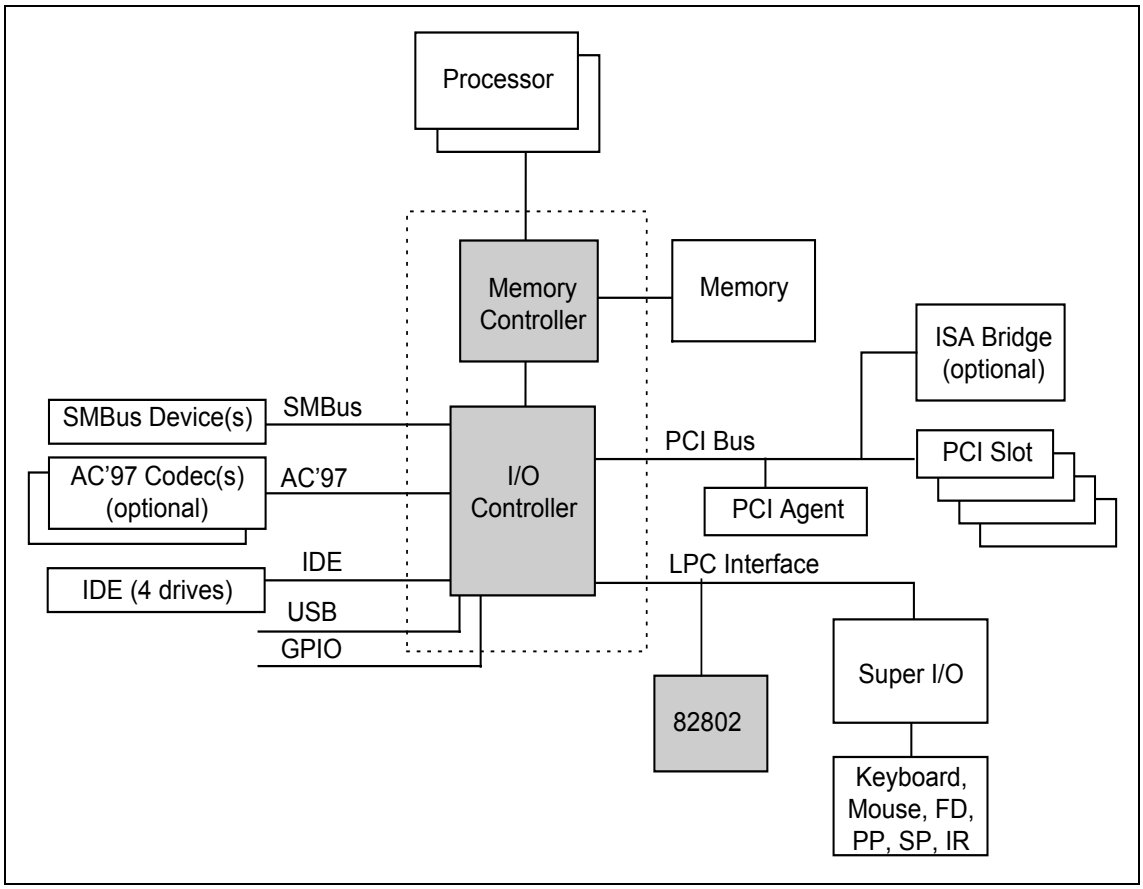
Intel® 82802AB/AC Firmware Hub (FWH)

Product Features

- Intel platform compatability
  - Enables security-enhanced platform infrastructure; facilitates option to remove ISA.
- Firmware hub hardware interface mode
  - 5-Signal communication interface supporting byte-at-a-time 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
  - Contains a hardware Random Number Generator (RNG) for enhancing platform security
  - Integrated Command User Interface (CUI) for requesting access to locking, programming, and erasing options. The CUI also handles requests for data residing in status, ID, and block-lock registers.
  - Operates with 33-MHz PCI clock and 3.3 V 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
- 4 or 8 Mbits of flash memory for platform code/data nonvolatile storage
  - Symmetrically blocked, 64-KB memory sections
  - Available in 8-Mbit (Intel® 82802AC) and 4-Mbit (Intel® 82802AB) densities
  - Automated byte program and block erase via 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: 3.3 V ± 0.3 V
  - Vpp: 3.3 V and 12 V for fast programming, (80 hours maximum)

The Intel® 82802 (FWH) firmware hub may contain design defects or errors known as errata that may cause the products to deviate from published specifications. Current characterized errata are available upon request.

Figure 1. Simplified Block Diagram



1. Architectural Overview

The Intel® 82802 Firmware Hub (FWH) discrete component is compatible with several Intel chipset platforms and a variety of applications. The device operates under the LPC/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. This 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 platforms. The platform RNG, accessed through the Intel® Security Driver and third-party software, enables security features for the PC platform. See the product features listed previously for a list of more key features that the Intel FWH provides.

1.1. Interface Overview

This device is equipped with two hardware interfaces. The state of the device's "IC" (InterfaceConfiguration) pin determines which interface is in use. The interface mode must be selected prior topower-up or before return from reset (RST# or INIT# low-to-high transition). The Intel FWH interface isdesigned to work with the Intel family of I/O Controller Hubs (ICH) during platform operation. The A/A Mux interface is designed as a programming interface for OEMs, for use during motherboard manufacturing or component pre-programming. The A/A Mux interface is not intended for use during regular personal computer operation. Such a configuration would cause the expected (Intel FWH) interface to be disabled, and the system boot sequence would fail upon power-up.

An internal Command User Interface (CUI) serves as the internal control center for the nonvolatilememory core in either of the two device interfaces (Intel FWH or A/A Mux). A single valid commandsequence written to the CUI initiates an automated sequence of internal events to complete various tasks. 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 (tPHQV A/A Mux) is required from RST# or INIT# switching high until outputs are valid. Likewise, the device has a wake time (tPHRH 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. Resetting the component will put the component back into read-array mode.

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

1.1.1. Intel Firmware Hub Interface

The Intel Firmware Hub (Intel 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 Intel FWH interface is the only way access the full feature set of the device. The Intel FWH interface is equipped to operate at 33 MHz, synchronous with the PCI bus.

1.1.2. Address/Address-Multiplexed Interface

The A/A Mux refers to the multiplexed row and column addresses in this interface. This approach is required so that the device can be tested and programmed quickly with automated test equipment (ATE) or off-board 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 erasure of the nonvolatile memory blocks can be performed through the A/A Mux interface. In this mode, the Intel FWH features, security features, and registers are unavailable. A row/column (R/C#) pin determines which set of addresses (rows or columns) is latched. See the A/A Mux pin description table for more information.

1.2. Nonvolatile Flash Memory Core

The primary feature of the Intel FWH component is a nonvolatile memory core based on Intel® Flash Technology. This high-performance memory array is arranged in eight (4-Mbit device) or sixteen (8-Mbit device) 64-KB blocks.

Intel® Flash Technology enables fast factory programming and low-power designs. Specifically designed for 3-V systems, this component supports read operations at 3.3 V V_{CC} and block erase and program operations at 3.3 V and 12 V V_{PP}. The 12 V V_{PP} option yields the fastest program performance, which will increase factory throughput, but is not recommended for standard in-system FWH operation in the platform, due to an **80-hr limit for 12 V** on the V_{PP} pin over the lifetime of the device, whether or not programming is taking place. With the 3.3-V V_{PP} option (recommended for in-system operation), V_{CC} and V_{PP} may be tied together for a simple, low-power 3-V design. In addition to the voltage flexibility, the dedicated V_{PP} pin provides complete data protection when $V_{PP} \leq V_{PPLK}$. Internal V_{PP} detection circuitry automatically configures the device for block erase and program operations. While current for 12-V programming will be drawn from V_{PP}, 3.3-V programming solutions should design their board 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 2. Device Memory Map with Intel FWH Hardware Lock Architecture

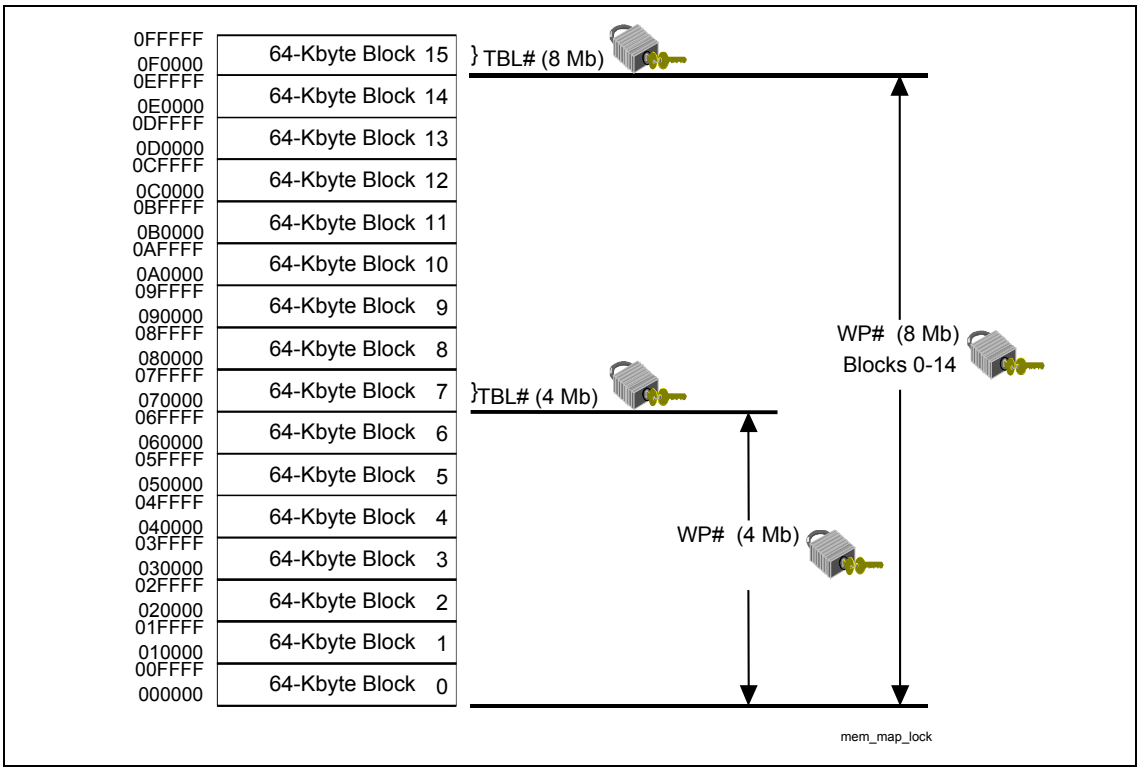
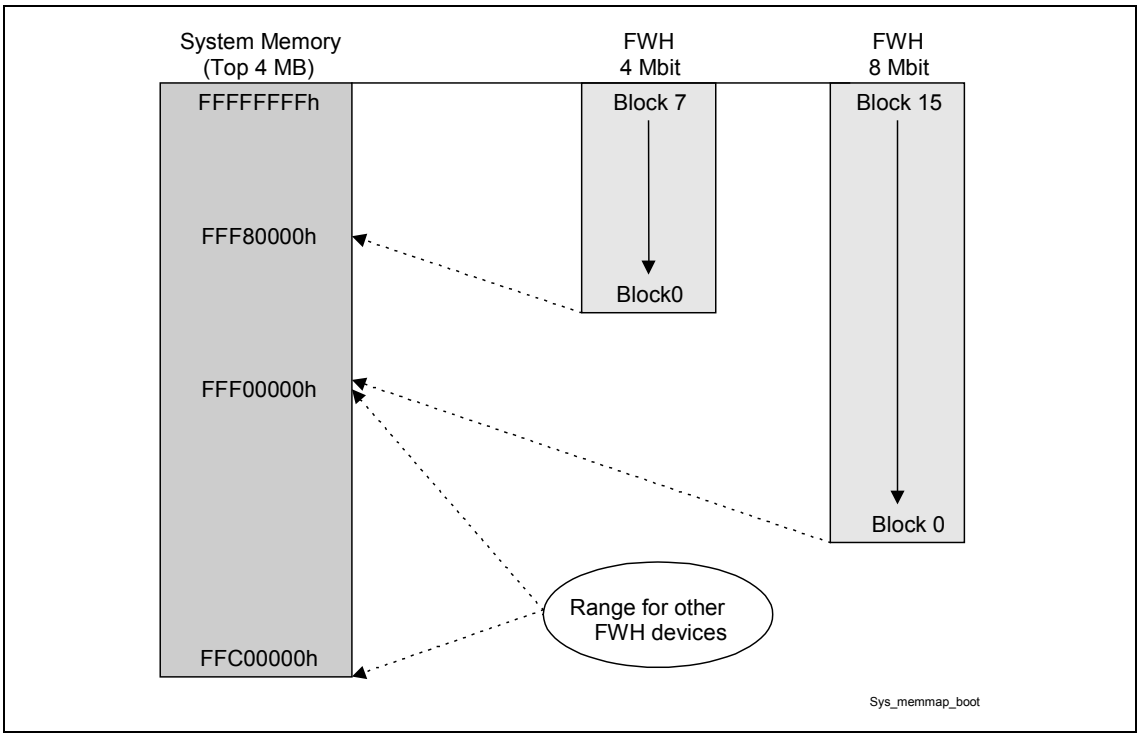


Figure 3. Intel FWH Boot-Configuration System Memory Map



This page is intentionally left blank.

2. Pinout Configurations

Figure 4. 32-Lead PLCC Intel Firmware Hub Pinout

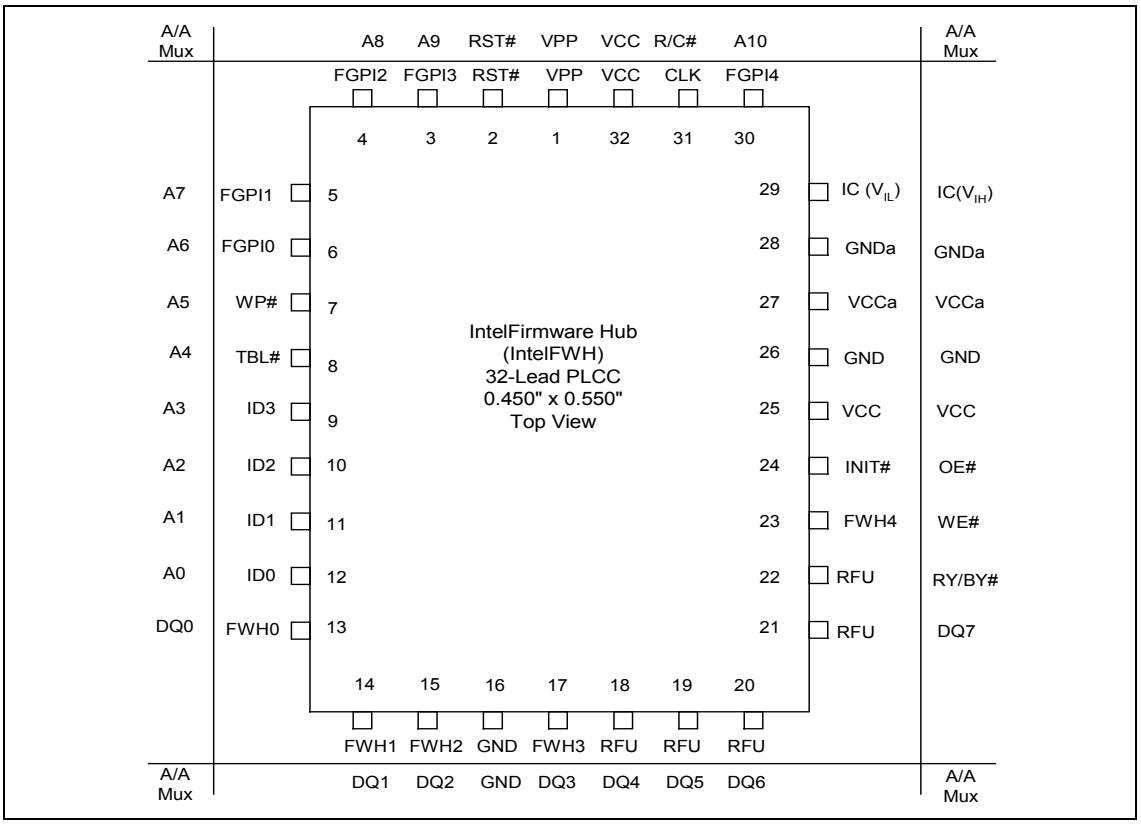
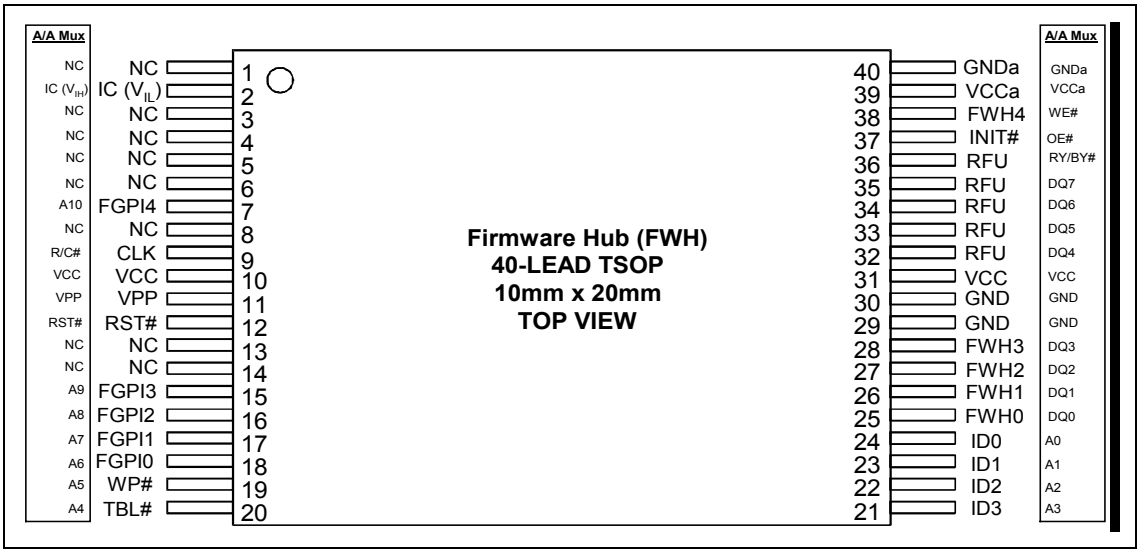


Figure 5. 40-Lead TSOP Intel Firmware Hub Pinout



2.1. Pin Descriptions

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

Table 1. Pin Descriptions

Symbol	Type	Interface		Name and Function
		Intel FWH	A/A Mux	
IC	I	X	X	**Interface Configuration Pin.** This pin determines which interface is used to communicate with the device. When it is held low, the Intel FWH interface is enabled. When it is held High, the A/A Mux interface is enabled. This pin must be set at power-up or before return from reset, and must not be changed during device operation. This pin is pulled down with an internal resistor of between 20 and 100 kΩ. When the IC is High (A/A Mux mode), this pin will exhibit a leakage current of approximately 200 µA. This pin may be floated, which will select the Intel FWH mode.
RST#	I	X	X	**Interface Reset.** Valid for both A/A Mux and Intel FWH interface operation. When driven low, RST# inhibits write operations to provide data protection during power transitions, resets internal automation, and tri-states pins FWH[3:0] (in Intel FWH interface mode). RST#-high enables normal operation. When exiting from reset, the device defaults to read array mode.
INIT#	I	X		**Processor 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. voltage depends on the processor. Do **not** use 3.3 V). ***A/A Mux = OE#***
CLK	I	X		**33-MHz Clock for** Intel **FWH Interface.** This input is the same as that for the PCI clock and adheres to the PCI specification. ***A/A Mux = R/C#***
FWH[3:0]	I/O	X		**Intel FWH I/Os.** I/O communication ***A/A Mux = DQ[3:0]***
FWH4	I	X		**Intel FWH Input.** Input communication ***A/A Mux = WE#***

Symbol	Type	Interface		Name and Function
		Intel FWH	A/A Mux	
ID[3:0]	I	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 use sequential up-count strapping (0001, 0010,0011,...). These pins are pulled down with internal resistors, with values between 20 and 100 kΩ, when in the Intel FWH mode. Any ID pins pulled high will exhibit a leakage current of approximately 200 µA. Any pins intended to be low may be left to float. In a single Intel FWH system, all may be left floating. ***A/A Mux = A[3:0]***
FGPI[4:0]	I	X		**Intel FWH General Purpose Inputs.** These individual inputs can be used for additional board flexibility. The state of these pins can be read immediately at boot, through Intel 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 they should remain at the same level until the end of the read cycle. They may *only* be used for **3.3-V** signals. Unused FGPI pins must **not** be floated. ***A/A Mux = A[10:6]***
TBL#	I	X		**Top Block Lock.** When low, it prevents programming or block erase to the highest addressable block (7 in a 4-Mbit, 15 in an 8-Mbit component), 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#	I	X		**Write Protect.** When low, prevents programming or block erase to all but the highest addressable block (0-6 in a 4-Mbit, 0-14 in an 8-Mbit component), regardless of the state of the corresponding lock registers. 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[0:10]	I		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[0:7]	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 outputs are disabled. Data is internally latched during a write cycle.
OE#	I		X	**Output Enable.** Gates the device's outputs during a read cycle
R/C#	I		X	**Row-Column Address Select.** For the A/A Mux interface, this pin determines whether the address pins are pointing to the row addresses (A[0:10]) or the column addresses (A[11:19]).
WE#	I		X	**Write Enable.** Controls writes to the CUI and array blocks. Addresses and data are latched on the rising edge of the WE# pulse.

Symbol	Type	Interface		Name and Function
		Intel FWH	A/A Mux	
V_{PP}	PWR	X	X	**Block Erase/Program Power Supply.** For erasing array blocks or programming data. V_{PP} = 3.3 V or 12 V V_{PP}. With $V_{PP} \leq V_{PPLK}$, memory contents cannot be altered. Attempting a block erase or program with an invalid V_{PP} (see DC Characteristics) will produce spurious results and should not be attempted. V_{PP} may only be held at 12 V for 80 hours over the lifetime of the device.
V_{CC}	PWR	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 attempts to write to flash memory are inhibited. Device operations at invalid V_{CC} voltages (see DC Characteristics) produce spurious results and should not be attempted.
GND	PWR	X	X	**Ground.** Do not float any ground pins.
V_{CCa}	PWR	X	X	**Analog Power Supply.** This supply should share the same system supply as V_{CC}.
GND_a	PWR	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. They may be left disconnected or driven. If they are driven, the voltage levels should satisfy V_{IH} and V_{IL} requirements. **A/A Mux = DQ[7:4]**
NC		X	X	**No Connect.** Pin may be driven or floated. If it is driven, the voltage levels should satisfy V_{IH} and V_{IL}. No connects appear only on the 40ld TSOP package.
Ry/By#	0		X	**Ready/Busy.** Valid only in A/A Mux Mode. This output pin is a reflection of bit 7 in the Status Register. This pin is used to determine block erase or program completion.

3. Interface Operation Description

3.1. Read

Memory information, identifier codes, GPI registers or the status register can be read, regardless of the V_{PP} voltage. Commands using the read mode include: 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 RNG status register. Upon initial device power-up or after exit from reset, the device automatically resets to read array mode.

3.2. Write

Writes to the memory array's CUI are initiating by issuing a write through the Intel FWH interface. (See the following information on timing and Intel FWH cycle write protocol specifics.) The CUI does not occupy a single, specific memory location—any valid address may be given. However, certain commands, such as block erase, require the address be within the range of the desired address block.

3.3. Output Disable

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

3.4. Reset

RST# or INIT# at V_{IL} initiates a device reset. In the 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 Intel FWH resets to read array mode upon return from reset, and all blocks are set to default (locked) status (see 4.9.1), 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 is hardware dependent—it is possible that proper processor initialization will not occur. (The Intel FWH memory may be providing status information instead of memory array data.)

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

The TBL# and WP# pins on the Intel FWH provide hardware write protect capabilities. The Top Block Lock (TBL#) pin, 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 has a function similar to TBL#, but affects all remaining blocks. 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. When active, these pins write-protect the appropriate block(s), regardless of the associated block-locking registers. (For 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 ≤ 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 2. Command Definitions

Command	Bus Cycles Required	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			

Note:

1. Key:
   - 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 for the read identifier code data.
3. The block must 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 previously are reserved by Intel for future device implementations and should not be used.

Table 3. Status Register Definition

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

Bit	Description
7	**Write State Machine Status (SR.7).** Check SR.7 to determine block erase or program completion. SR.6–0 are invalid while SR.7 = 0. 1 = Ready 0 = Busy
6	**Erase Suspend Status (SR.6).** 1 = Block erase suspended 0 = Block erase in progress/completed
5	**Erase Status (SR.5).** If both SR.5 and SR.4 are 1s after a block erase attempt, an improper command sequence was entered. 1 = Error in block erasure 0 = Successful block erase
4	**Program Status (SR.4).** 1 = Error in program 0 = Successful program
3	**V_{PP} Status (SR.3).** 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 $V_{PP} \neq V_{PPH1/2}$. 1 = V_{PP} low detect, operation abort 0 = V_{PP} OK
2	**Program Suspend Status (SR.2).** 1 = Program suspended 0 = Program in progress/completed
1	**Device Protect Status (SR.1).** 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. Depending on the attempted operation, it informs the system whether or not the selected block is locked. 1 = Write-lock bit, TBL# pin, or WP# pin Detected, operation abort 0 = Unlock
0	**Reserved for future enhancements (SR.0).** 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 the read array mode. This operation can also be initiated by writing the Read Array command. The device remains available for array 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 operation is completed, unless the operation 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 write of the command, the device will read back the (manufacturer and device) ID data from the addresses shown in the following table. To terminate the read identifier code operation, write another valid command to the Intel FWH. The Read Identifier Codes command functions independently of the V_{PP} voltage.

Table 4. 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 1s by the WSM and can only be reset by the Clear Status Register command. These bits indicate various conditions that may cause failure. 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 erasure changes all block data to FFh. Block preconditioning, erase, and erase verify are handled internally by the WSM, which is transparent 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 may be checked. If the FWH detects a block erase error, the status register should be cleared before system software attempts corrective actions. After a block erase, **the CUI remains in read status register mode until a new command is issued.**

Successful block erasure requires that the corresponding block's write-lock-bit is cleared, and the corresponding write-protect pin (TBL# or WP#) is inactive. If a block erase is attempted when the block is locked, the block erase will fail, with the reason for failure in the status register.

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 the status register data when read. When programming is complete, the status register may be checked. If a program error is detected, the status register should be cleared before corrective action is taken by the software. The internal WSM verification error checking only detects 1s that does not successfully program to 0s. 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 programming is attempted at $V_{PP} \neq V_{PPH1}$ or V_{PPH2}, erratic results may occur.

Successful program operation also requires that the corresponding block's write-lock bit be cleared and that the corresponding write-protect pin (TBL# or WP#) be inactive. If 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 the status register can help determine when the block erase operation was suspended.

After a successful suspend, a Read Array command may be written to read data from a block other than the suspended block. A Program command sequence may also be issued during erase suspend to program data in blocks other that the block currently in the 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 initially used for the 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 was 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, General-Purpose Input, and Random Number Generator Registers

A series of registers are available in the Intel FWH to provide software read- and write-locking and GPI feedback. Also available are the set of control registers for controlling and gathering random numbers. These registers are accessible through standard addressable memory space (see the following table).

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 read.

Table 5. Intel Firmware Hub Register Configuration Map

Memory Address	Mnemonic	Register Name	Default	Type
FFBF0002h	T_BLOCK_LK	Top Block Lock Register (4-8-Mbit FWH)	01h	R/W
FFBE0002h	T_MINUS01_LK	Top Block [-1] Lock Register (4-8-Mbit FWH)	01h	R/W
FFBD0002h	T_MINUS02_LK	Top Block [-2] Lock Register (4-8-Mbit FWH)	01h	R/W
FFBC0002h	T_MINUS03_LK	Top Block [-3] Lock Register (4-8-Mbit FWH)	01h	R/W
FFBB0002h	T_MINUS04_LK	Top Block [-4] Lock Register (4-8-Mbit FWH)	01h	R/W
FFBA0002h	T_MINUS05_LK	Top Block [-5] Lock Register (4-8-Mbit FWH)	01h	R/W
FFB90002h	T_MINUS06_LK	Top Block [-6] Lock Register (4-8-Mbit FWH)	01h	R/W
FFB80002h	T_MINUS07_LK	Top Block [-7] Lock Register (4-8-Mbit FWH)	01h	R/W
FFB70002h	T_MINUS08_LK	Top Block [-8] Lock Register (8-Mbit FWH)	01h	R/W
FFB60002h	T_MINUS09_LK	Top Block [-9] Lock Register (8-Mbit FWH)	01h	R/W
FFB50002h	T_MINUS10_LK	Top Block [-10] Lock Register (8-Mbit FWH)	01h	R/W
FFB40002h	T_MINUS11_LK	Top Block [-11] Lock Register (8-Mbit FWH)	01h	R/W
FFB30002h	T_MINUS12_LK	Top Block [-12] Lock Register (8-Mbit FWH)	01h	R/W
FFB20002h	T_MINUS13_LK	Top Block [-13] Lock Register (8-Mbit FWH)	01h	R/W
FFB10002h	T_MINUS14_LK	Top Block [-14] Lock Register (8-Mbit FWH)	01h	R/W
FFB00002h	T_MINUS15_LK	Top Block [-15] Lock Register (8-Mbit FWH)	01h	R/W
FFBC0100h	FGPI_REG	FWH General-Purpose Input Register	N/A	RO
FFBC015Fh		RNG Hardware Status Register	40h*	R/W
FFBC0160h		RNG Data Status Register	0	RO
FFBC0161h		RNG Data Register	N/A	RO

* Assumes RNG is present and not disabled.

4.9.1. T_BLOCK_LK and T_MINUSxx_LK — Block-Locking Registers

Memory Address: FFBx0002h (x = F-0h)
Default Value: 01h
Access: R/W
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 only can 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 bit 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 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).
00h	00000	0	0	0	Full access
01h	00000	0	0	1	Write locked. **Default state at power-up**
02h	00000	0	1	0	Locked open (full access locked down).
03h	00000	0	1	1	Write-locked down.
04h	00000	1	0	0	Read locked.
05h	00000	1	0	1	Read and write locked.
06h	00000	1	1	0	Read-locked down.
07h	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, changes in the block's locking status do not occur 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.

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 also is not set. The 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# pin.

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 cannot be read from that block. An attempted read from a read-locked block will result in the data 00h. (Note that failure is not reflected in the status register.) The read-lock status can be unlocked by clearing the read-lock bit, provided the lock-down bit has not been set. The current read-lock status of a particular block can be determined by reading the corresponding read-lock bit.

Lock-Down

In the Intel 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, because future attempts to change that block-locking register will be ignored. The lock-down bit is cleared only upon a device reset with RST# or INIT#. The 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.

4.9.2. General-Purpose Input Register

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

4.9.2.1. GPI_REG — General-Purpose Input 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).

4.9.3. Random Number Generator Registers

When enabled and active, the Random Number Generator (RNG) is designed to fill an 8-bit register, a bit at a time, with hardware-generated random numbers. When this register is full, a flag bit in the RNG data status register transitions to a 1, indicating that a valid random number is ready to be read. This bit will immediately reset to 0 upon reading the RNG data register.

The advantages of random numbers over pseudo-random numbers as well as a brief overview of the simple mathematics of testing RNGs are discussed superficially in the companion document, *The Intel® Platform RNG Tech Brief*, which is available online.

4.9.3.1. RNG Hardware Status Register

Memory Address: FFBC015Fh
Default Value: 40h, for typical component out of reset
Access: RO
Size: 8 bits

Bit	Function
7	**Reserved**
6	**RNG Present—RO.** Determines whether or not an RNG is present on this component, or if it has been disabled. 1 = RNG Present 0 = RNG not present
5:1	**Reserved**
0	**RNG Enabled—R/W.** Determines whether the RNG is generating a random number. 1 = RNG enabled 0 = RNG disabled

4.9.3.2. RNG Data Status Register

Memory Address: FFBC0160h
Default Value: 00h
Access: RO
Size: 8 bits

Bit	Function
7:1	**Reserved**
0	**RNG Output Valid.** Determines whether the RNG data register contains a valid random number. 1 = RNG data register contians valid random data 0 = RNG data register contents not valid

4.9.3.3. RNG Data Register

Memory Address: FFBC0161h
Default Value: 40h, for typical component out of reset
Access: RO
Size: 8 bits

Bit	Function
7:0	**RNG Output**: (Should only be used if RNG Data Status Register indicates valid output.)

4.10. Using the Random Number Generator

The Intel Firmware Hub integrates a Random Number Generator that utilizes thermal noise generated as a result of the inherently random quantum mechanical properties of silicon, in order to modulate a proven hardware RNG design. Internal circuitry is included to enhance the entropy of the output. Since the output of the RNG is non-deterministic, it is an excellent choice for cryptography applications, but it also is a convenient source of random numbers for mathematics, modeling, graphics algorithms, artificial intelligence, entertainment, and many other applications. The fact that it is a component of the platform and may be utilized remotely on a locked-away server makes it an ideal (and much more reliable) source of entropy for applications that, in the past, have relied exclusively on a key press or other environmental input. Several Intel Firmware Hub components may be used in tandem (see the following section) when additional RNG bandwidth is required. When not generating new random bits, the RNG circuitry will enter a low power state.

4.11. Detecting and Initializing the RNG Device

Before any process attempts to read random data directly from the Intel Firmware Hub RNG device, it should execute a process to verify that a supported RNG device is available for use, enable the device, and verify the correct functionality. This initialization process is described in a following subsection.

4.11.1. Detecting the RNG Device

The Manufacturer Code and Hardware Status registers are used to determine whether a supported RNG device is available on the system.

Step 1: From the system BIOS or using the Read Identifier Codes command, as specified in the Intel® 82802AB/82802AC Firmware Hub (FWH) datasheet, verify the Intel® 82802 manufacturer code.

Step 2: If a valid Intel® 82802 FWH is found, then the RNG Present bit (bit 6) of the Hardware Status register should be checked in order to verify that an RNG device is available.

Note: There is a chance that, even if no RNG device is present, the physical memory locations described above may coincidentally match the values expected for an RNG device. For this reason, before random data is sent to an application, the device should be exercised to verify that it is indeed an RNG. This can be accomplished by enabling the device and running an initial test (e.g., FIPS (Federal Information Processing Standard) 140-1) before use.

4.11.2. Initializing the RNG Device

Once the RNG device is detected, it must be enabled and should be tested before use.

Step 1: The RNG Enabled bit (bit 0) of the Hardware Status register must be set to enable the RNG device.

Step 2: Once the RNG is enabled, an initial read of the RNG Data register should be made to clear any preexisting data from the register.

Step 3: A test (e.g., FIPS 140-1) should be run on the RNG Device. This test will ensure that there was no error in detecting the device and that the device is functioning properly.

4.11.3. 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, the BIOS support, bus loading or the attaching bridge may limit this number. Note that, regardless of the number of FWH components, the maximum "window" of the FWH array visible at one time is 4 MB (for Intel® ICH1) and 8MB for Intel® ICH2. The boot device must have an ID (as determined by ID [0:3]) of 0. For clarity, it is advisable that subsequent devices use incremental numbering.

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

In special applications, when it is desirable to use multiple FWH components of different densities—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 the boundaries delimited by the highest-capacity device, as illustrated in the following table.

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.

Yes	No	Yes
8 Mbits	4 Mbits	4 Mbits
	8 Mbits	4 Mbits
4 Mbits		8 Mbits

Biggest is 8 Mbits.

4.11.4. Mapping FWH Devices onto Memory Map

There is 4 MB of available memory space devoted to the FWH. Therefore, the Intel ICH has the ability to select which FWH device maps into each region of the system address space.

In the existing Intel ICH, the address map is broken up into eight 512-KB segments. The BIOS Select Register in the Intel ICH is a 32-bit register that contains the needed mapping information, thereby determining which FWH receives requests from which portion of the address map. For example, in a system with four 8-Mbit devices, this register would be 00112233h, which is the default power-up state for this register. In a system with eight 4-Mbit devices, the register must be changed to 01234567h.

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.

4.11.5. Paging FWH Devices for Greater Than 4 MB of FWH Memory

In certain applications, even a 4-MB window of flash memory is inadequate. It is possible to exceed this amount by using a paging scheme. Individual FWH devices may then be "swapped" in and out of system memory space. This must be implemented at the BIOS level, to permit modification of the Intel ICH BIOS Select Register. A number of paging algorithms may be used successfully with the FWH memory space, using the Intel ICH BIOS Select Register. This register, then, determines which FWH device gets mapped into each 512 KB "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 regarding these components and the BIOS Select Register.

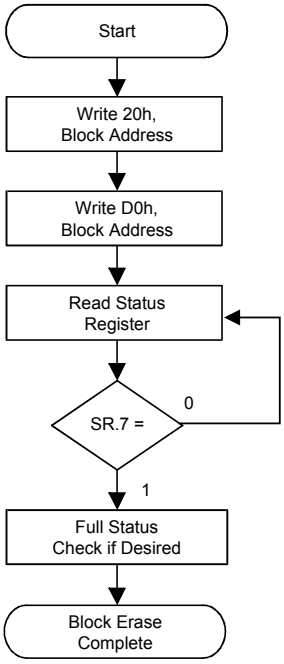
Note: The paging of FWH devices will also "page" features, potentially affecting the visibility or location of the FGPI register (see Section 4.9.2.1) or of an active/ready RNG. When a paging scheme is used, it is recommended that critical FPGIs be used only on the ID 0 FWH device, which must remain mapped at the top of memory. Ideally, the RNG driver in a system with more than four FWHs should verify the mapping of FWHs in order to keep track of which RNGs are active and which are present in the memory map. There is no convenient way, aside from checking the select register, to determine which IDed FWH is in which location in the memory map.

4.11.6. 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 the A/A Mux mode, the recommended means of programming multiple devices is either out-of-system programming with standalone PROM programmers or in-system programming using the FWH mode. 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 the A/A Mux mode, thereby leaving the others in the FWH mode.

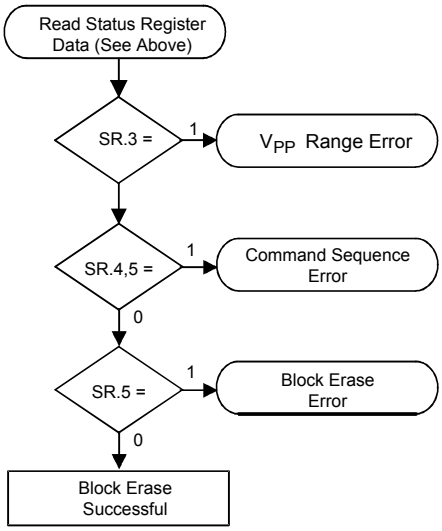
4.12. CUI Automation Flowcharts

Figure 6. 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.		

This page left intentionally blank

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 .....-0.2 V to 4.1 V

Voltage On Any Pin (except V_{PP}):–0.5 V to +VCC + 0.5 V(1,2,5)

V_{PP} voltage: ..... –0.5 V to +14.0 V(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. The minimum DC voltage on the V_{PP} pin is –0.5 V. During transitions, this level may undershoot to –2.0 V for periods of <20 ns. During transitions, this level may overshoot to V_{CC} + 2.0 V for periods of <20 ns.
2. The maximum DC voltage on V_{PP} may overshoot to +14.0 V for periods of <20 ns.
3. Output shorted for no more than one second. No more than one output is shorted at a time. This note applies only to non-PCI outputs.
4. Connection to supply of V_{HH} is allowed for a maximum cumulative period of 80 hours.
5. Do not violate processor or chipset limitations on the INIT# pin.

5.2. Operating Conditions

Table 7. 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 (3.3 V ± 0.3 V)		3.0	3.6	V	

Note:

1. This temperature requirement differs from the normal commercial operating condition of flash memories.

5.2.1. Interface DC Input/Output Specifications

Table 8. Intel 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
$V_{IH\ (INIT\#)}$	INIT# input high voltage		1.35	V_{CC} +0.5	V	5
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. IIL may be changed on IC and ID pins (up to 200µA), if pulled against internal pull-downs. Refer to the pin descriptions (Table 1).
5. Do not violate processor or chipset specifications regarding the INIT# pin voltage.

Table 9. 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 stand-by current (FWH interface)	Voltage range of all inputs is VIH to VIL, FWH4 = VIH, VCC = 3.6 V, CLK f = 33 MHz No internal operations in progress.		100	µA	2,3,4
ICCSL2	VCC stand-by current (FWH interface)	FWH4 = VIL VCC = 3.6 V, CLK f = 33 MHz No internal operations in progress.		10	mA	2,3,4
ICCA	VCC active current	VCC = VCC Max, CLK f = 33 MHz Any internal operation in progress, IOUT = 0mA		67	mA	2,3,5
IPPR	VPP read current	$V_{PP} \geq V_{CC}$		200	µA	2
IPPWE	VPP program or erase current	V_{PP} = 3.0-3.6 V		40	mA	2
		V_{PP} = 11.4-12.6 V		15	mA	2

Note:

1. All currents are RMS, unless otherwise noted. These currents are valid for all packages.
2. $V_{PP} = V_{CC}$
3. $V_{IH} = 0.9\ V_{CC}$, $V_{IL} = 0.1\ V_{CC}$ per the PCI output V_{OH} and V_{OL} specifications of Table 8.
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 10. Intel FWH Interface AC Input/Output Specifications

Symbol	Parameter	Condition	Min.	Max.	Units	Notes
Ioh(AC)	Switching current High	$0 < V_{OUT} \leq 0.3\ V_{CC}$	$-12\ V_{CC}$		mA	
		$0.3\ V_{CC} < V_{OUT} < 0.9\ V_{CC}$	$-17.1\ (V_{CC} - V_{OUT})$		mA	
		$0.7\ V_{CC} < V_{OUT} < V_{CC}$		Equation C		
	(Test point)	$V_{OUT} = 0.7\ V_{CC}$		$-32\ V_{CC}$	mA	
Iol(AC)	Switching current Low	$V_{CC} > V_{OUT} \geq 0.6\ V_{CC}$	$16\ V_{CC}$		mA	
		$0.6\ V_{CC} > V_{OUT} > 0.1\ V_{CC}$	$-17.1\ (V_{CC} - V_{OUT})$		mA	
		$0.18\ V_{CC} > V_{OUT} > 0$		Equation D		
	(Test point)	$V_{OUT} = 0.18\ V_{CC}$		$38\ V_{CC}$	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	$0.2\ V_{CC}$ - $0.6\ V_{CC}$ load	1	4	V/ns	1
slewf	Output fall slew rate	$0.6\ V_{CC}$ - $0.2\ V_{CC}$ load	1	4	V/ns	1

Note:

1. PCI specification output load is used.

5.2.3. Intel FWH Interface AC Timing Specifications

5.2.3.1. Clock Specification

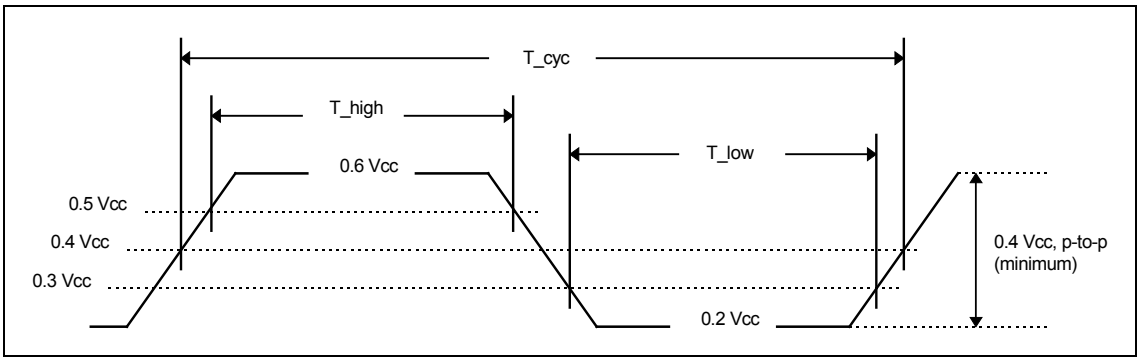
Table 11. Clock Specification

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 the PCI specificaiton.
2. Applies only to the rising edge of the signal. See Chapter 4 of the PCI electrical specification.

Figure 7. Clock Waveform



5.2.3.2. Signal Timing Parameters

Table 12. 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 setup 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	$t_{rst-clk}$	Reset active time after CLK stable		100		µs	
TPLQZ	$t_{rst-off}$	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 as that in which 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 8. Output Timing Parameters

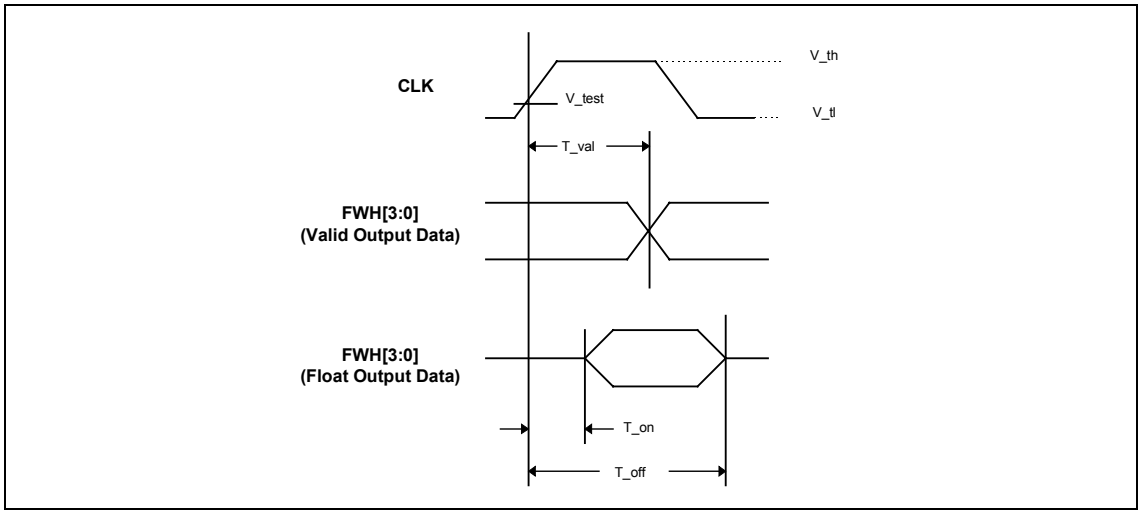


Figure 9. Input Timing Parameters

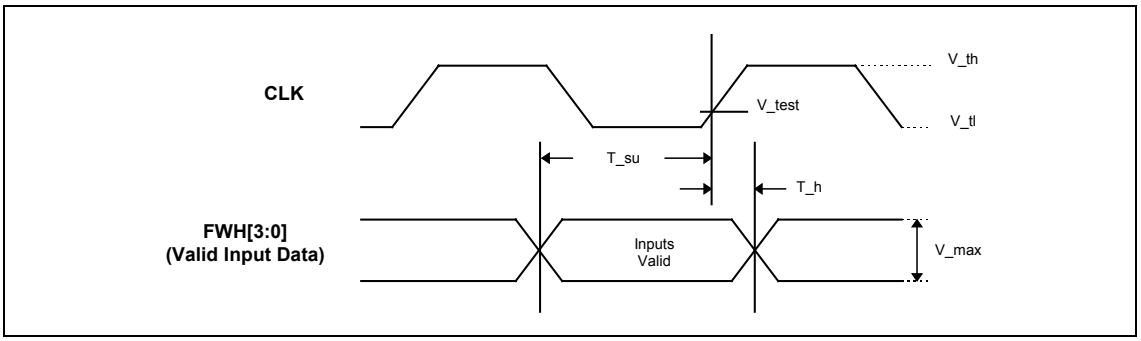


Table 13. Interface Measurement Condition Parameters

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 uses 0.1 VCC of overdrive over VIH and VIL. Timing parameters must be met with no more overdrive than this. Vmax specifies the maximum peak-to-peak waveform allowed for measuring the input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

Reset Operations

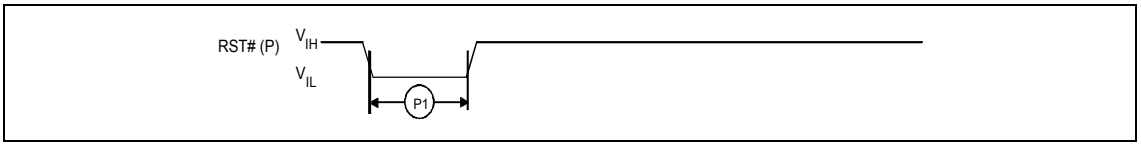


Table 14. AC Waveform for Reset Operation

#	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:

1. There will be a 20-µs reset latency if a reset procedure is performed during a programming or erase operation.

5.3. Block Programming Times

Table 15. 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.

5.4. Intel Firmware Hub Interface

The firmware hub relies on the Intel Firmware Hub interface to communicate with the outside world. This interface consists of four bi-directional signals and one "control" input. The timing and electrical parameters of the FWH interface are similar to those of the LPC interface, to provide compatibility between the interfaces, but differ in cases mentioned earlier in this section (clock pin capacitance), as well as in certain timing parameters. The Intel ICH has been engineered to accommodate both interfaces, which allows the Intel FWH interface signals to be communicated over the same set of pins as LPC. The Intel FWH interface is designed to use an LPC-compatible start cycle, with a reserved cycle type code. This ensures that all LPC devices present on the shared interface will ignore cycles destined for the FWH, without becoming "confused" by the different protocol.

This section contains timing and protocol information for the Intel FWH interface. Note that the Intel FWH interface is a licensed interface, so the appropriate license must be obtained from Intel for components supporting the Intel FWH interface (e.g., ASICs, PLDs).

5.4.1. Intel FWH Interface Cycles

When the Intel FWH interface is active, information is transferred to and from the FWH by a series of "fields," where each field contains 4 bits of data. Many fields are one clock cycle in length but can be of variable length, depending upon the nature of the field. Field sequences and contents are strictly defined for read and write operations. The following tables list the field sequences for read and write cycles.

Addresses in this section refer to addresses as seen from the FWH's "point of view," so some calculation will be required to translate these to the actual locations in the memory map (and vice versa).

5.4.1.1. Read Cycle Sequence

The firmware hub supports single-byte or multibyte reads. The logic waveforms for these cycles are shown in Table 16 and Figure 11

Table 16. FWH Read Cycle

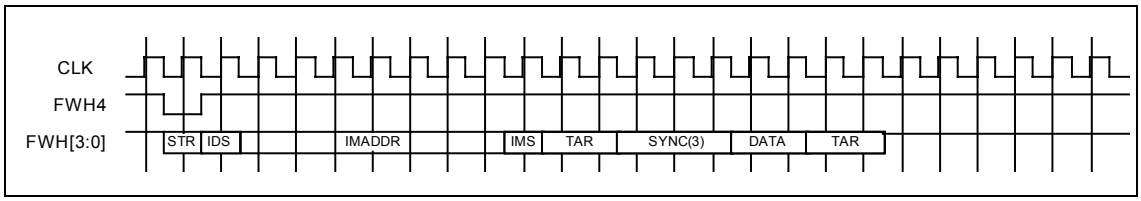
Clock Cycle	Field Name	Field Contents[1] FWH[3:0]	FWH[3:0] Direction	Comments
1	START	1101	IN	FWH4 must be active (low) for the part to respond. Only the last start field (before FWH4 transitioning high) should be recognized. The START field contents indicate an FWH memory read cycle.
2	IDSEL	0000 to 1111	IN	Indicates which FWH device should respond. If the IDSEL (ID select) field matches the value ID[3:0], then that particular device will respond to subsequent commands.
3-9	IMADDR	YYYY	IN	These seven clock cycles make up the 28-bit memory address. YYYY is one nibble of the entire address. Addresses are transferred most-significant nibble first. On multibyte data transfers, lower-order addresses will be zero, depending on page size.
10	IMSIZE	0000 (1 byte)	IN	A field of this size indicates how many bytes will be transferred during multibyte operations. The FWH will only support single-byte transfers.
11	TAR0	1111	IN then float	In this clock cycle, the master (Intel ICH) has driven the bus to all 1s and then floats the bus, prior to the next clock cycle. This is the first part of the bus "turnaround cycle."
12	TAR1	1111 (float)	Float then OUT	The FWH takes control of the bus during this cycle. During the next clock cycle, it will be driving "sync data."
13-14	WSYNC	0101 (WAIT)	OUT	The FWH outputs the value 0101, a wait-sync (WSYNC, a.k.a. "short-sync"), for two clock cycles. This value indicates to the master (Intel ICH) that data is not yet available from the part. This number of wait-syncs is a function of the device's access time.
15	RSYNC	0000 (READY)	OUT	During this clock cycle, the FWH will generate a "ready-sync" (RSYNC) indicating that the least-significant nibble of the least-significant byte will be available during the next clock cycle.
16	DATA	YYYY	OUT	YYYY is the least-significant nibble of the least-significant data byte.
17	DATA	YYYY	OUT	YYYY is the most-significant nibble of the least-significant data byte.
17+ 3 x 2^{n-1} + 2^n	"DATA"	2 WSYNCS + 1 RSYNC + 2 DATA	OUT	n = IMSIZE. Each subsequent byte of data requires 2 wait-syncs + 1 ready-sync + 2 data nibbles. The FWH supports only n=0000 (single-byte) reads.
Previous + 1	TAR0	1111	OUT then float	In this clock cycle, the Inel FWH has driven the bus to all ones and then floats the bus prior to the next clock cycle. This is the first part of the bus "turnaround cycle."
Previous + 1	TAR1	1111 (float)	Float then IN	The master (Intel ICH) resumes control of the bus during this cycle.

Note:

1. Field contents are valid on the rising edge of the present clock cycle.

5.4.1.2. Single-Byte Read Waveforms

Figure 10. FWH Single-Byte Read Waveforms



5.4.1.3. Write Cycle Sequence

The firmware hub only supports single-byte writes. Each byte represents either the data to be written or a valid flash command. Refer to the waveforms in Figure 11.

Table 17. FWH Write Cycle

Clock Cycle	Field Name	Field Contents[1] FWH[3:0]	FWH[3:0] Direction	Comments
1	START	1110	IN	FWH4 must be active (low) for the part to respond. Only the last start field (before FWH4 transitioning high) should be recognized. The START field contents indicate an FWH memory write cycle.
2	IDSEL	0000 to 1111	IN	Indicates which FWH device should respond. If the IDSEL (ID select) field matches the value ID[3:0], then that particular device will respond to subsequent commands.
3-9	IMADDR	YYYY	IN	These seven clock cycles make up the 28-bit memory address. YYYY is one nibble of the entire address. Addresses are transferred most-significant nibble first.
10	IMSIZE	0000 (1 byte)	IN	This size field indicates how many bytes will be transferred during read/write operations. The FWH only supports single-byte writes.
11	DATA	YYYY	IN	This field is the least-significant nibble of the data byte. This data is either the data to be programmed into the flash memory or any valid flash command.
12	DATA	YYYY	IN	This field is the most-significant nibble of the data byte.
13	TAR0	1111	IN then float	In this clock cycle, the master (Intel ICH) has driven the bus to all 1s and then floats the bus prior to the next clock cycle. This is the first part of the bus "turnaround cycle."
14	TAR1	1111 (float)	Float then OUT	The FWH takes control of the bus during this cycle. During the next clock cycle it will be driving the "sync" data.
15	RSYNC	0000	OUT	The FWH outputs the values 0000, indicating that it has received data or a flash command.

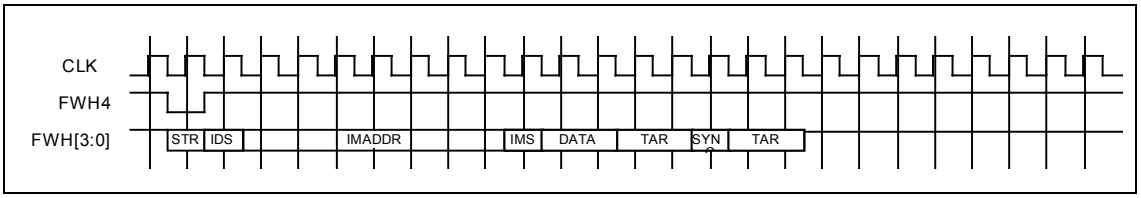
Clock Cycle	Field Name	Field Contents[1] FWH[3:0]	FWH[3:0] Direction	Comments
16	TAR0	1111	OUT then float	In this clock cycle, the FWH has driven the bus to all 1s and then floats the bus prior to the next clock cycle. This is the first part of the bus "turnaround cycle."
17	TAR1	1111 (float)	Float then IN	The master (Intel ICH) resumes control of the bus during this cycle.

Note:

1. Field contents are valid on the rising edge of the present clock cycle.

5.4.1.4. Write Waveforms

Figure 11. Write Waveforms



5.4.1.5. Response To Invalid Fields

During FWH operations, the Intel FWH will not explicitly indicate that it has received invalid field sequences. The response to specific invalid fields or sequences is as follows:

- **Address out of range:** The Intel FWH address sequence is 7 fields long (28 bits), but only the last five address fields (20 bits) will be decoded by an 8-Mbit FWH. (For a 4-Mbit density, the most-significant bit (FWH3) in the third address field also will be ignored.) The Intel FWH will respond to these lower addresses, regardless of the value of the more-significant address bits. Address A22 has the special function of directing reads and writes to the flash core (A22 = 1) or to the register space (A22 = 0).
- **Invalid IMSIZE field:** If the Intel FWH receives an invalid size field during a read or write operation, the internal state machine will reset and no operation will be attempted. The Intel FWH will generate no response of any kind in this situation. Invalid-size fields for a read cycle are anything but 0000. Invalid-size fields for a write cycle are anything but 0000. When accessing register space, invalid field sizes are anything but 0000.
- **Non-page-aligned address:** The Intel FWH assumes that multibyte read addresses are page aligned (i.e., for a 32-byte access, the lower 5 address bits will be zero). If they are not zero, the first byte of data returned by the Intel FWH will correspond to that explicit address, and subsequent data will be as if the first address was indeed page aligned.

Once valid START, IDSEL, and IMSIZE fields are received, the Intel FWH always will respond to subsequent inputs as if they were valid. As long as the states of FWH [3:0] and FWH4 are known, the response of the Intel FWH to signals received during the FWH cycle should be predictable. The Intel FWH will make no attempt to check the validity of incoming flash operation commands.

5.4.1.6. Abort Operations

FWH4 active (low) indicates either that a START cycle will eventually occur or that an abort is in progress. In either case, if FWH4 is asserted, the Intel FWH will "immediately" tri-state its outputs and the FWH state machine will reset.

During a write cycle, there is a possibility that an internal flash write or erase operation is in progress (or has just been initiated). If FWH4 is asserted during this time frame, the internal operation will *not* abort. The software must send an explicit flash command to terminate or suspend the operation.

The internal FWH state machine will not initiate a flash write or erase operation until it has received the last data nibble from the chipset. This means that FWH4 can be asserted as late as this cycle ("cycle 12") and no internal flash operation will be attempted. However, since the Intel FWH will start "processing" incoming data before it generates its SYNC field, it should be considered a *non-buffered peripheral* device.

5.4.1.7. Intel FWH Cycle Timing Information

Refer to Figure Figure 12 and Figure 13.

Table 18. 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 setup 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	$t_{rst-clk}$	Reset active time after CLK stable		100		µs	
TPLQZ	$t_{rst-off}$	Reset active to output float delay			48	ns	2

Note:

1. Minimum and maximum times have different loads. See the PCI specification.
2. For purposes of active/float timing measurements, the Hi-Z or "off" state is defined as the state where 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).

5.5. RNG Parameters

Table 19. RNG Timing Characteristics

#	Sym	Parameter	Notes	Typ.	Max.	Unit
		Write RE = 1 to DWord ready or read DWord to new DWord ready	1	450	1500	µs
		Average sustained throughput	1	13	50	µs/bit

Note:

1. Sampled, not 100% tested.

Table 20. RNG Statistical Characteristics

#	Sym	Parameter	Notes	Min.	Typ.	Max.	Unit
	B2	Fractional probability of excess 1s	1,2,3			±316	10-6
	AC	Auto correlation coefficient	3			± 632	10-6
	FOM	Figure of merit		3	7.5	17	

Note:

1. Sampled, not 100% tested.

Figure 12. Intel FWH Output Timing Parameters

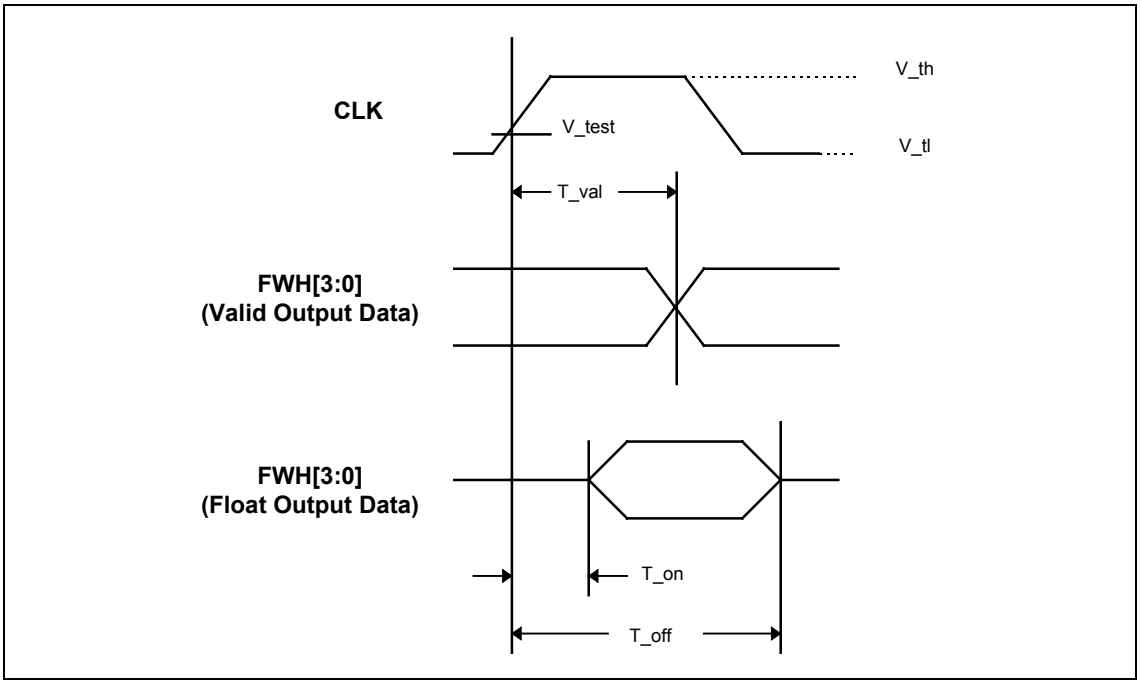
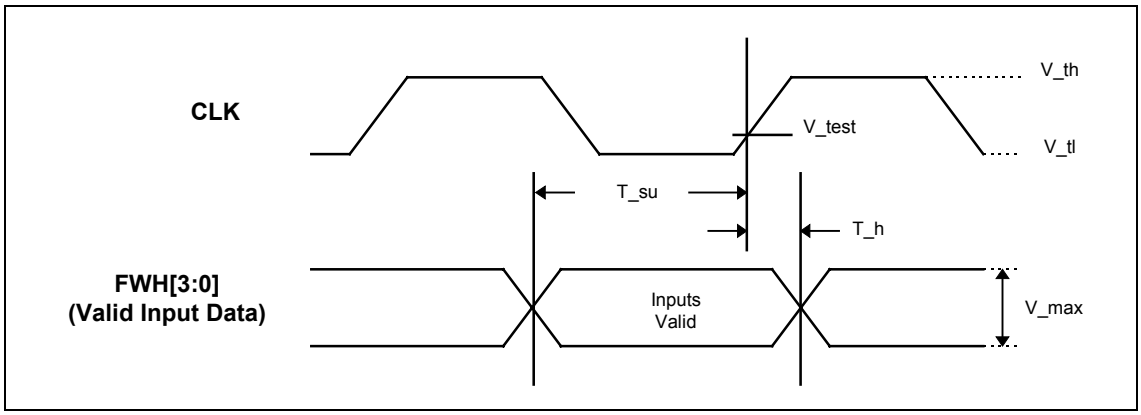


Figure 13. Intel FWH Input Timing Parameters



6. PROM Programming Specifications

6.1. Programming ("A/A Mux") Mode Operation

The Intel® 82802 is designed to offer a parallel programming mode for faster factory programming. This mode, called the A/A Mux mode, is selected by IC high. The IC pin is pulled down internally in the Intel® 82802, so it should be expected that a modest current will be drawn. (See the pin descriptions in Table 1 for further information.)

The following information applies **only** to the Intel® 82802 when in the A/A Mux mode. Information regarding the FWH mode (i.e., the standard operating mode) is provided in earlier chapters of this document

6.2. Bus Operation

All A/A mux bus cycles can be conformed to operate on most automated test equipment and PROM programmers.

6.2.1. Output Disable/Enable

With OE# at the logic-high level (V_{IH}), the device outputs are disabled. Output pins DQ0–DQ7 are placed in the high-impedance state. With OE# at the logic-low level (V_{IL}), the device outputs are enabled. Output pins DQ0–DQ7 are placed in the output-drive state.

6.2.2. Row/Column Addresses

R/C# is the A/A mux control pin used to latch row (A0–A10) and column addresses (A11–A18/4 Mbits, or A[11:19] /8 Mbits). R/C# latches row addresses on the falling edge and column addresses on the rising edge.

6.2.3. Read Operation

Block information, identifier codes or status register data can be read independently of the V_{PP} voltage. The first task is to write the appropriate read-mode command (Read Array, Read Identifier Codes or Read Status Register) to the CUI. Upon initial device power-up or after exit from reset, the device defaults to the read array mode. Four control pins dictate the data flow into and out of the component: R/C#, OE#, WE#, and RST#. R/C# is the A/A mux control pin used to latch row and column addresses. OE#, the data output control pin (DQ0–DQ7), drives the selected memory data onto the I/O bus, when active. WE# and RST# must be at V_{IH}.

6.2.4. Read Identifier Codes Operation

The read identifier codes operation outputs the manufacturer and device codes (see Table 4). Using the manufacturer and device codes, automated test equipment (ATE) or PROM programmer software can confirm the proper device ID.

6.2.5. Write Operation

The CUI does not occupy a specific addressable memory location. It is written to when WE# is active and OE# = V_{IH}. The address previously captured by R/C# transitions and the data needed to execute a command are latched on the WE# rising edge.

Table 21. Bus Operations

Mode	Notes	RST#	OE#	WE#	Address	V_{PP}	DQ[0:7]
Read	1,2,6	V_{IH}	V_{IL}	V_{IH}	X	X	D_{OUT}
Output Disable	6	V_{IH}	V_{IH}	V_{IH}	X	X	High Z
Read Identifier Codes	3,6	V_{IH}	V_{IL}	V_{IH}	*Note 3*	X	*Note 3*
Write	4,5,6	V_{IH}	V_{IH}	V_{IL}	X	X	D_{IN}

Note:

1. When $V_{PP} \le V_{PPLK}$, the memory contents can be read, but not altered.
2. X can be V_{IL} or V_{IH} for the control and address input pins and V_{PPLK} or $V_{PPH1/2}$ for the V_{PP} supply pin. See the DC characteristics for the V_{PPLK} and $V_{PPH1/2}$ voltages.
3. See Table 4 for the read identifier code data and addresses.
4. Command writes involving block erase or program are reliably executed when $V_{PP} = V_{PPH1/2}$ and $V_{CC} = V_{CC} \pm 0.3$ V.
5. Refer to Table 2 for the valid D_{IN} during a write operation.
6. V_{IH} and V_{IL} refer to the DC Characteristics associated flash memory output buffers: V_{IL} min = -0.5V, V_{IL} max = 0.8V and V_{IH} min = 2.0V, V_{IH} max = V_{CC} + 0.5V.

6.3. Command Definitions

Flash core programming commands in A/A Mux mode are identical to commands for the FWH mode. Refer to Section 4 of this document.

6.4. Electrical Characteristics in A/A Mux Mode

Certain specifications differ from the previous sections, when programming in the A/A Mux mode. The following subsections provide this data. Any information not provided here is not specific to the A/A Mux mode. Refer to Section 5 of this document and use the Intel FWH mode specifications.

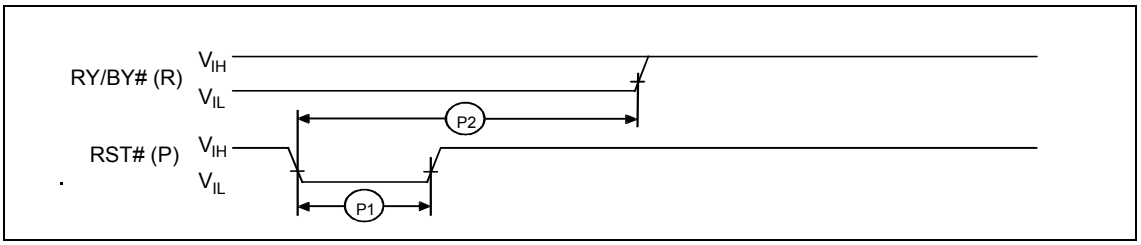
6.4.1. Reset Operations

#	Symbol	Parameter	Notes	Min.	Max.	Unit
P1	t_{PLPH}	RST# pulse low time (If RST# is tied to V_{CC}, this specification is not applicable.)		100		ns
P2	t_{PLRH}	RST# low to reset during block erase or program	1, 2		20	µs

Note:

1. If RST# is asserted when the WSM is not busy (RY/BY# = '1'), the reset will complete within 100 ns.
2. A reset time, t_{PHAV}, is required from the latter of RY/BY# or RST# going high until outputs are valid.

6.4.2. AC Waveforms for Reset Operations



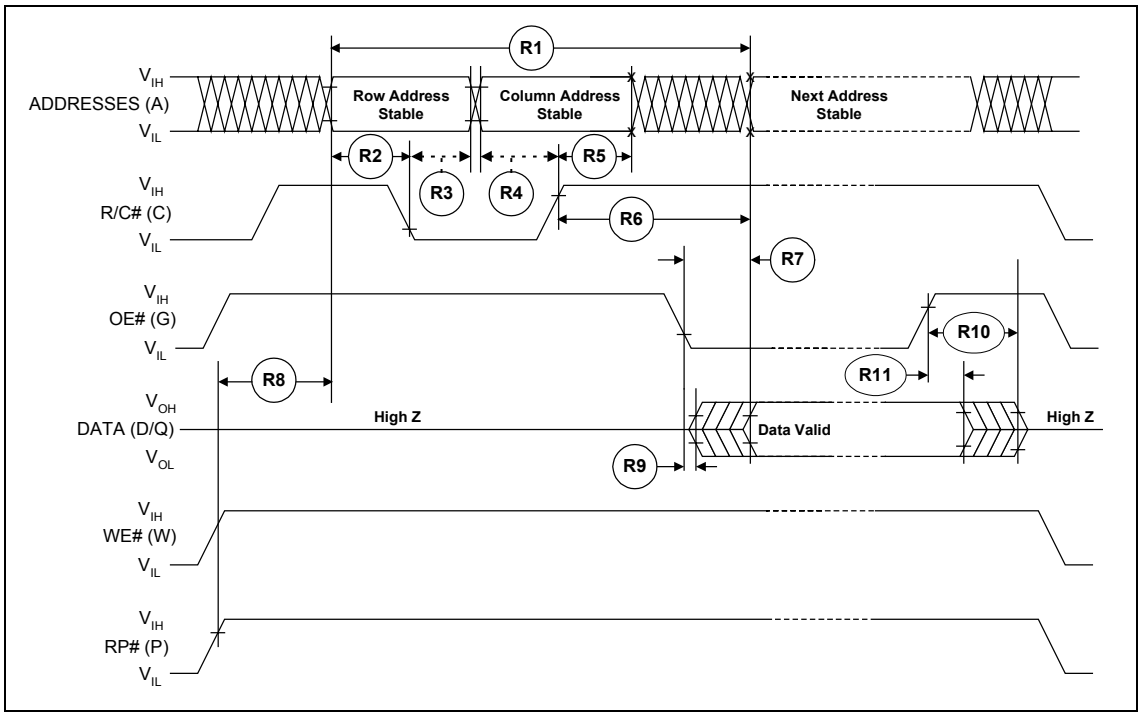
6.4.3. A/A Mux Read-Only Operations (1,3)

#	Symbol	Parameter	Notes	Min.	Max.	Unit
R1	t_{AVAV}	Read cycle time		250		ns
R2	t_{AVCL}	Row address setup to R/C# low		50		ns
R3	t_{CLAX}	Row address hold from R/C# low		50		ns
R4	t_{AVCH}	Column address setup to R/C# high		50		ns
R5	t_{CHAX}	Column address hold from R/C# high		50		ns
R6	t_{CHQV}	R/C# high to output delay	2		150	ns
R7	t_{GLQV}	OE# low to output delay	2		50	ns
R8	t_{PHAV}	RST# high to row address setup		1		µs
R9	t_{GLQX}	OE# low to output in low Z		0		ns
R10	t_{GHQZ}	OE# high to output in high Z			50	ns
R11	t_{QXGH}	Output hold from OE# high		0		ns

Note:

1. See the AC input/output reference waveform for the maximum allowable input slew rate.
2. OE# may be delayed up to $t_{CHQV} - t_{GLQV}$ after the rising edge of R/C# without affecting t_{CHQV}.
3. T_C = 0 °C to + 85 °C, 3.3 V ± 0.3 V V_{CC}

Figure 14. A/A Mux Read Timing Diagram



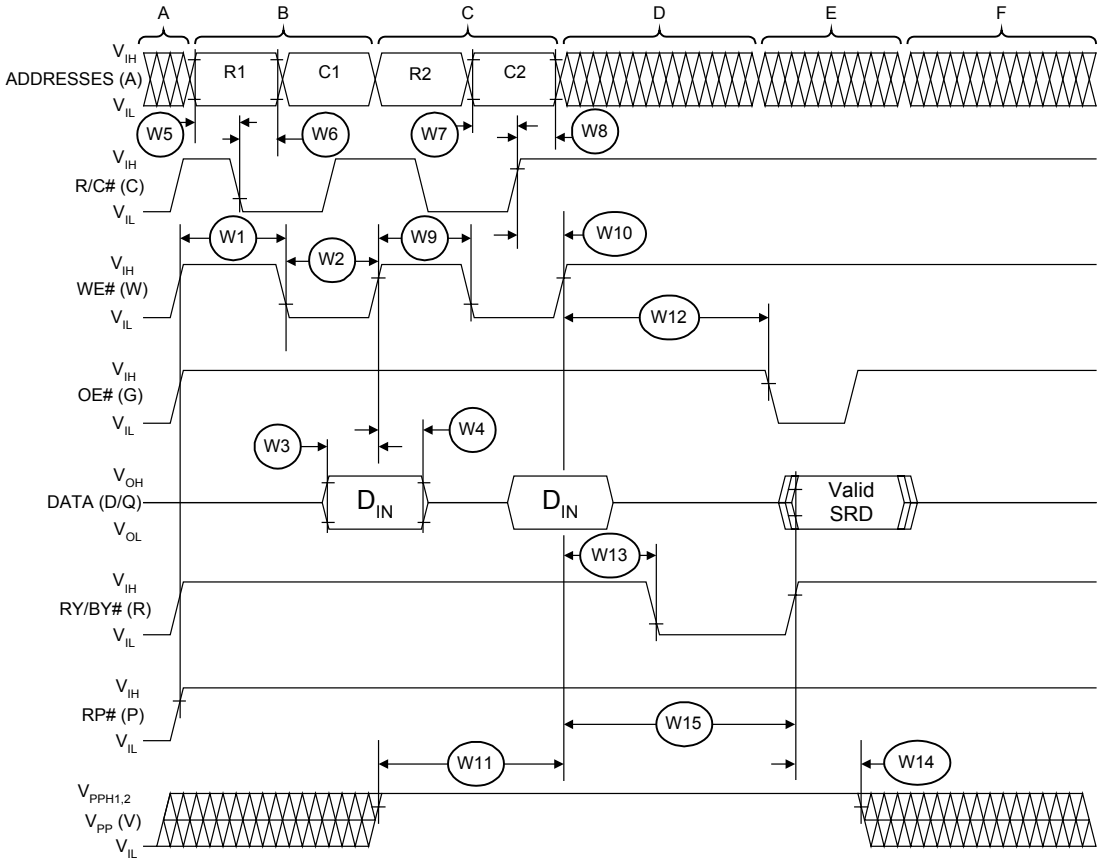
6.4.4. A/A Mux Write Operations (1,2)

#	Symbol	Parameter	Notes	Min.	Max.	Units
W1	t_{PHWL}	RP# high recovery to WE# low		1		µs
W2	t_{WLWH}	Write pulse width low		100		ns
W3	t_{DVWH}	Data setup to WE# high	1	50		ns
W4	t_{WHDX}	Data hold from WE# high	1	5		ns
W5	t_{AVCL}	Row address setup to R/C# low	1	50		ns
W6	t_{CLAX}	Row address hold from R/C# low	1	50		ns
W7	t_{AVCH}	Column address setup to R/C# high	1	50		ns
W8	t_{CHAX}	Column address hold from R/C# high	1	50		ns
W9	t_{WHWL}	Write pulse width high		100		ns
W10	t_{CHWH}	R/C# high setup to WE# high		50		ns
W11	t_{VPWH}	$V_{PP1,2}$ setup to WE# high		100		ns
W12	t_{WHGL}	Write recovery before read			150	ns
W13	t_{WHRL}	WE# high to RY/BY# going low		0		ns
W14	t_{QVVL}	$V_{PP1,2}$ hold from valid SRD, RY/BY# high		0		ns

Note:

1. Refer to Table 6-28 [?] for valid A_{IN} and D_{IN} for block erase or program or other commands.
2. Tc = 0 °C to + 85 °C, 3.3 V ± 0.3 V V_{CC}

Figure 15. A/A Mux Write Timing Diagram



Note:

A V_{CC} power-up and stand-by

B Write block erase or program setup

C Write block erase confirm or valid address and data

D Automated erase or program delay

E Read status register data

F Ready to write another command

Intel around the world

United States and Canada
Intel Corporation
Robert Noyce Building
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119
USA
Phone: (800) 628-8686

Europe
Intel Corporation (UK) Ltd.
Pipers Way
Swindon
Wiltshire SN3 1RJ
UK

Phone:

England	(44) 1793 403 000
Germany	(49) 89 99143 0
France	(33) 1 4571 7171
Italy	(39) 2 575 441
Israel	(972) 2 589 7111
Netherlands	(31) 10 286 6111
Sweden	(46) 8 705 5600

Asia-Pacific
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway, Central
Hong Kong, SAR
Phone: (852) 2844 4555

Japan
Intel Kabushiki Kaisha
P.O. Box 115 Tsukuba-gakuen
5-6 Tokodai, Tsukuba-shi
Ibaraki-ken 305
Japan
Phone: (81) 298 47 8522

South America
Intel Semicondutores do Brazil
Rue Florida, 1703-2 and CJ22
CEP 04565-001 Sao Paulo-SP
Brazil
Phone: (55) 11 5505 2296

For more information
To learn more about Intel Corporation, visit our site on the World Wide Web at www.intel.com