AP-364

APPLICATION NOTE

28F008SA Automation and Algorithms

BRIAN DIPERT
MCD MARKETING APPLICATIONS

January 1996

Order Number: 292099-003

Information in this document is provided in connection with Intel products. Intel assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of Intel products except as provided in Intel's Terms and Conditions of Sale for such products.

Intel retains the right to make changes to these specifications at any time, without notice. Microcomputer Products may have minor variations to this specification known as errata.

*Other brands and names are the property of their respective owners.

†Since publication of documents referenced in this document, registration of the Pentium, OverDrive and iCOMP trademarks has been issued to Intel Corporation.

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

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

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

28F008SA AUTOMATION AND ALGORITHMS

CONTENTS

PAGE

CONTENTS

PAGE

1.0 INTRODUCTION

The Intel 28F008SA FlashFile™ Memory is today's optimum solution for high density solid state storage. Flash memory, exemplified by the 28F008SA, is an enabling technology for today's powerful system designs that are higher performance, more compact, lighter, more rugged and have longer battery life.

Features of the 28F008SA include:

- High-Density Symmetrically Blocked Architecture:
  - Sixteen 64-Kbyte Blocks
- Extended Cycling Capability
  - 100,000 Block Erase Cycles
  - 1.6 Million Block Erase Cycles per Chip
- Automated Byte Write and Block Erase
  - Command User Interface
  - Status Register
- System Performance Enhancements
  - RY/BY# Status Output
  - Erase Suspend Capability
- Deep Powerdown Mode
  - 0.20 μA I_{CC} Typical
- Very High Performance Read
  - 85 ns Maximum Access Time
- SRAM-Compatible Write Interface
- Hardware Data Protection Features
  - Erase/Write Lockout during Power Transitions
- Industry Standard Packaging
  - 40 Lead TSOP, 44 Lead PSOP
- ETOX III Nonvolatile Flash Memory Technology
  - 12V Byte Write/Block Erase

The 28F008SA's automation is a significant enhancement to the manual algorithms of first-generation flash memory devices. System software and hardware designs that fully understand and exploit this automation will greatly benefit from its versatility and capabilities. The concepts presented in this document are applicable to such designs.

This application note discusses in-depth operation of the 28F008SA FlashFile memory Write State Machine and internal algorithms, emphasizing how they interface to system hardware and software. The 28F008SA datasheet (order number 290429) is a valuable reference document, providing in-depth device technical specifications, package pinouts and timing waveforms. Companion application note AP-359, "28F008SA Hardware Interfacing" (order number 292094) describes supply voltage derivation and filtering, control input/output implementation, high density layout and high speed design techniques, as well as providing example system interfaces to common microprocessor buses. AP-360, "28F008SA Software Drivers" (order number 292095) provides example ASM-86 and "C" routines for controlling the 28F008SA. AP-359 and AP-360 should be reviewed in conjunction with this application note and the 28F008SA datasheet for a complete understanding of this device.

2.0 AUTOMATION AND ALGORITHMS

Figure 1 shows a block diagram of the 28F008SA and its internal contents. Although a main subject of this application note is software interface to read and alter memory contents, it is useful to begin with an overview of the 28F008SA hardware subsections that are directly manipulated by the system. In particular, this application note will first discuss the Write State Machine (WSM) and Command User Interface/Status Register, and then explain the software routines that control this hardware.

2.1 28F008SA Automation and the Write State Machine

When the system microprocessor reads flash memory data from the 28F008SA, it uses control lines CE# and OE#, along with address inputs, to select a byte of data directly from the memory cell array. However, the system does not directly access the array when it writes to the 28F008SA; instead it writes to the Command User Interface, whose register contents are interpreted and translated into WSM actions. The WSM can be thought of as a dedicated "processor", along with companion clock-generation circuitry, integrated into the flash memory. After receiving proper commands or command sequences, it controls byte write and block erase algorithms internally. The status of the WSM is not invisible to the system; the WSM interfaces to the outside world through a full-featured Status Register and dedicated RY/BY# (Ready/Busy#) output. Automation has significant benefits, some of which are more obvious than others.

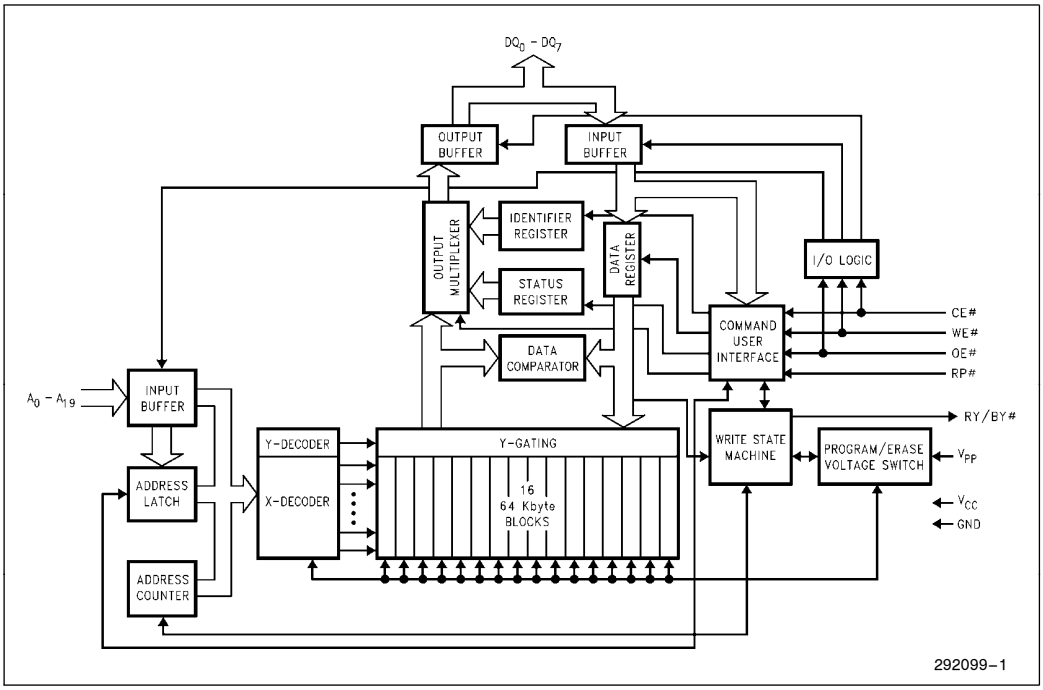


Figure 1. 28F008SA Block Diagram

The WSM architecture dramatically simplifies the program and erase algorithms of first-generation flash memory devices. Hardware/software timers, erase preprogramming, byte-by-byte verification and margining, pulse repetition and limited microprocessor multitasking capability throughout data update have been eliminated, replaced by a simple two-command write for both block erase and byte write. The 28F008SA WSM halts itself when its internal algorithms are complete, and can alert the system to this completion by a hardware interrupt (using RY/BY#) or via software polling of the 28F008SA Status Register.

Internal automation frees the system to execute higher-priority tasks while a 28F008SA is being block erased or byte written, and inherent in this capability is the most powerful advantage of the WSM. Operating systems prioritize file operations in the following order:

- Read
- Write
- Erase

When an array of 28F008SA components is used as solid-state storage (in a memory card, integrated in a flash-based "hard drive" form factor or resident on the system motherboard), system software can initiate slower block erase (0.3 sec minimum) of one or several 28F008SAs and, by not being "tied" to the erase algorithm, execute higher priority reads (85 ns minimum) or writes (6 μs minimum) of other 28F008SAs as operating system requests dictate. Additionally, erase suspend/resume capability allows data retrieval from a 28F008SA currently being block erased, again enabling "read" as the highest priority task. Block erase as a background task is discussed in Section 2.6 of this document.

Command User Interface

Table 2 shows the various command sequences that are accepted and interpreted by the 28F008SA Command User Interface and WSM. Writes to the CUI enable reading of device data and intelligent identifiers, reading and clearing of the Status Register, and commencement of internal byte write, block erase and erase suspend/resume algorithms. The CUI itself does not occupy a specifically addressable memory location, and contains a latch used to store the command and address/data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Byte Write Setup command requires appropriate command data and the address of the location to be written, while the Byte Write command consists of the data to be written and its address location.

Table 1. Status Register Definitions

WSMS	ESS	ES	BWS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
1 = Ready
0 = Busy

SR.6 = ERASE SUSPEND STATUS
1 = Erase Suspended
0 = Erase in Progress/Completed

SR.5 = ERASE STATUS
1 = Error in Block Erasure
0 = Successful Block Write

SR.4 = BYTE WRITE STATUS
1 = Error in Byte Write
0 = Successful Byte Write

SR.3 = V_{PP} STATUS
1 = V_{PP} Low Detect; Operation Abort
0 = V_{PP} OK

SR.2–SR.0 = RESERVED FOR FUTURE ENHANCEMENTS
These bits are reserved for future use and should be masked out when polling the Status Register.

NOTES:
RY/BY# or the Write State Machine Status bit must first be checked to determine byte write or block erase completion, before the Byte Write or Erase Status bits are checked for success.
If the Byte Write AND Erase Status bits are set to "1"s during a block erase attempt, an improper command sequence was entered. Attempt the operation again.
If V_{PP} low status is detected, the Status Register must be cleared before another byte write or block erase operation is attempted.
The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the byte write or block erase command sequences have been entered and informs the system if V_{PP} has not been switched on. The V_{PP} Status bit is not guaranteed to report accurate feedback between V_{PPL} and V_{PPH}.

Table 2. Command Definitions

Command	Bus Cycles Req'd	Notes	First Bus Cycle			Second Bus Cycle		
			Operation	Address	Data	Operation	Address	Data
Read Array/Reset	1		Write	X	FFH			
Intelligent Identifier	3	1, 2, 3	Write	X	90H	Read	IA	IID
Read Status Register	2	2	Write	X	70H	Read	X	SRD
Clear Status Register	1		Write	X	50H			
Erase Setup/Erase Confirm	2	1	Write	BA	20H	Write	BA	D0H
Erase Suspend/Erase Resume	2		Write	X	B0H	Write	X	D0H
Byte Write Setup/Write	2	1, 2, 4	Write	WA	40H	Write	WA	WD
Alternate Byte Write Setup/Write	2	1, 2, 4	Write	WA	10H	Write	WA	WD

NOTES:
1. IA = Identifier Address: 00H for manufacturer code, 01H for device code.
   BA = Address within the block being erased.
   WA = Address of memory location to be written.
2. SRD = Data read from Status Register. See Table 4 for a description of the Status Register bits.
   WD = Data to be written at location WA. Data is latched on the rising edge of WE.
   IID = Data read from intelligent identifiers.
3. Following the intelligent identifier command, two read operations access manufacturer and device codes.
4. Either 40H or 10H are recognized by the WSM as the Byte Write Setup command.
5. Commands other than those shown above are reserved by Intel for future device implementations and should not be used.

Status Register

Table 1 shows the 28F008SA Status Register and defines its various bits. Like the Command User Interface, it does not occupy a specific memory location within the device. It functions as an output of the WSM, informing the system when internal byte write or block erase algorithms have completed, if these algorithms completed successfully, and whether the 28F008SA is currently in Erase Suspend mode. Bit 7 (Write State Machine Status) is replicated in the device RY/BY# hardware output. The default state of the upper 5 bits of the Status Register after powerup and return from deep powerdown mode is 10000 (binary).

A separate Clear Status Register command allows reinitialization of Status Register data after analysis. The Status Register is not cleared until this command is written to the 28F008SA.

Bits 5 and 4 of the Status Register, if set by the WSM via a byte write or block erase attempt, do not block subsequent attempts (they need not be cleared before another byte write/block erase command sequence is written to the device). However, if the WSM detects a "low V_{PP}" condition and subsequently sets bit 3 of the Status Register, the Status Register MUST be cleared before another algorithm command sequence will be recognized by the 28F008SA.

It is important to note that the V_{PP} Status bit of the Status Register DOES NOT act like an always-functional A/D converter; its normal state, even with V_{PP} below 6.5V, is "0". The WSM only analyzes the V_{PP} level after a byte write or block erase command sequence has been written to the device, and if it detects that V_{PP} is "low" it will cancel the impending byte write or block erase operation and set the V_{PP} Status bit to "1". Therefore, the V_{PP} Status bit cannot be used by the system as an indication of proper V_{PP} level, before a byte write or block erase sequence is initiated. The system should instead insert an appropriate software delay between turning on V_{PP} and writing an initial command sequence, or use external hardware as a V_{PP} feedback mechanism.

2.2 Byte Write Algorithm

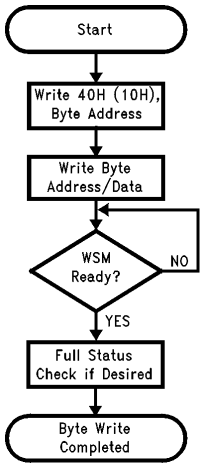
Figure 2 provides a graphical representation of the 28F008SA byte write algorithm. As can be seen, this consists solely of a two-command write sequence, followed by a periodic poll of the device RY/BY# output or Status Register. The 28F008SA automatically outputs Status Register data when read after the two-command byte write sequence (see Section 2.5). Byte write typically completes in 9 μs.

The byte write algorithm requires high voltage V_{PPH} (12V $\pm$5%) on the device V_{PP} input until internal algorithm completion is reported by the WSM. If byte write is attempted while $V_{PP} = V_{PPL}$ ($\leq$6.5V), the V_{PP} Status bit of the Status Register will be set to "1", and array data will not be altered. Byte write attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

The Status Register will only report errors for "1"s that do not write to "0"s during a byte write attempt. Erasure (see Section 2.3) is the method used to change data "0"s to "1"s using flash technology. If the system software attempts to write "1"s to a byte at bit locations already at value "0", no Status Register error will be reported for those specific bits.

It is often desired to write multiple bytes of data at one time to memory. Since the Status Register is only cleared after the Clear Status Register command is written to the 28F008SA, a string of bytes can be sequentially written to the device before the "full status check procedure" examines Status Register bits other than SR.7.

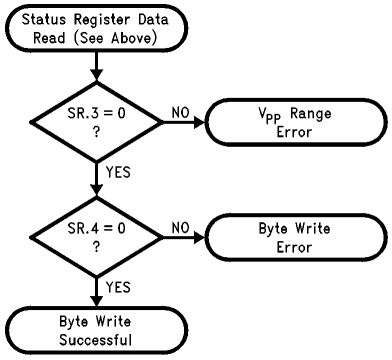
Byte write abort occurs when the 28F008SA RP# (Reset/Powerdown) input drops to V_{IL} (deep powerdown mode is entered), or V_{PP} drops to V_{PPL}. Although the WSM is halted in either case, byte data is partially written at the location where aborted. A repeat byte write sequence after system integrity is restored will complete the desired operation, or data can be initialized to a known value of "FF" thru block erasure.



292099–2

Bus Operation	Command	Comments
Write	Byte Write Setup	Data = 40H (10H) Address = Byte to be written
Write	Byte Write	Data to be written Address = Byte to be written
Standby/ Read		Check RY/BY# V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle OE# or CE# to update Status Register
Repeat for subsequent bytes Full status check can be done after each byte or after a sequence of bytes Write FFH after the last byte write operation to reset the device to Read Array Mode		

FULL STATUS CHECK PROCEDURE



292099–3

Bus Operation	Command	Comments
Optional Read		CPU may already have read Status Register data in WSM Ready polling above
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4 1 = Byte Write Error
SR.3 MUST be cleared, if set during a byte write attempt, before further attempts are allowed by the Write State Machine SR.4 is only cleared by the Clear Status Register Command, in cases where multiple bytes are written before full status is checked If error is detected, clear the Status Register before attempting retry or other error recovery.		

Figure 2. Automated Byte Write Flowchart

2.3 Block Erase Algorithm

Figure 3 provides a graphical representation of the 28F008SA block erase algorithm, similar in its two-command write sequence to the byte write algorithm discussed earlier. Both the Erase Setup and Erase Confirm commands must be accompanied by an address within the desired block to be erased to FFH. The 28F008SA automatically outputs Status Register data when read after the two-command block erase sequence (see Section 2.5). Block erase typically completes in 1.6 sec.

Again similar to byte write, the block erase algorithm requires high voltage V_{PPH} (12V $\pm$5%) on the device V_{PP} input until internal algorithm completion is reported by the WSM. If block erase is attempted while $V_{PP} = V_{PPL}$ ($\leq$6.5V), the V_{PP} Status bit of the Status Register will be set to "1", and array data will not be altered. Block erase attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and should not be attempted.

If write of the Erase Setup command is followed by write of any other command but Erase Confirm, the WSM will decode this as an illegal sequence. It will not attempt to erase the specified block, and will report error back to the system by setting both the Erase Status and Byte Write Status bits of the Status Register to "1". Since the Status Register is only cleared after the Clear Status Register command is written to the 28F008SA, a string of blocks within a 28F008SA can be sequentially erased before the "full status check procedure" examines Status Register bits other than SR.7.

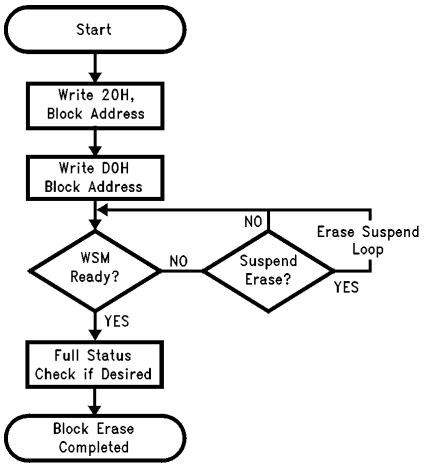
Block erase abort occurs when the 28F008SA RP# (Reset/Powerdown) input drops to V_{IL} (deep power-down mode is entered) or V_{PP} drops to V_{PPL}. A repeat block erase sequence after system integrity is restored will complete the desired operation.

2.4 Erase Suspend/Resume Algorithm

Figure 4 gives a software flowchart for implementing erase suspend/resume using the 28F008SA. As mentioned in Section 2.1, operating systems prioritize data reads highest, and consequently the 28F008SA has been designed with read as its highest performance function. Erase suspend allows system software to postpone WSM-controlled block erase if the system requests read of data from **a different block** of the same device. Although any block of the 28F008SA can be read, the block being erased when suspended will contain unknown data.

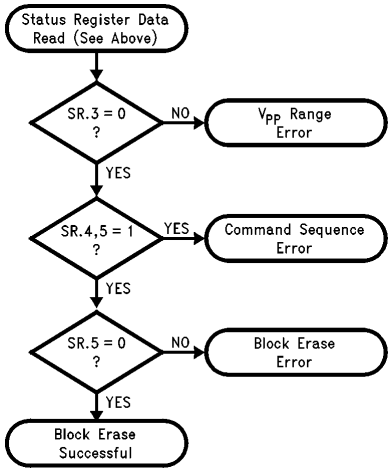
The 28F008SA is suspended by writing the Erase Suspend command (B0H) to it while the WSM is executing an erase algorithm. The WSM will halt block erase, set bits 7 and 6 of the Status Register to "1" and transition RY/BY# to V_{OH}, after which time system software can read data from either the array or Status Register. Issuing the Erase Resume command (D0H) signals the WSM to resume block erase.

V_{PP} must remain at V_{PPH} throughout the erase suspend interval, even when reading from the flash memory array. The 28F008SA will detect a V_{PP} transition to V_{PPL} while suspended, and report this error via Status Register bit 3 (set to "1") after the Erase Resume command is written to it.



292099-4

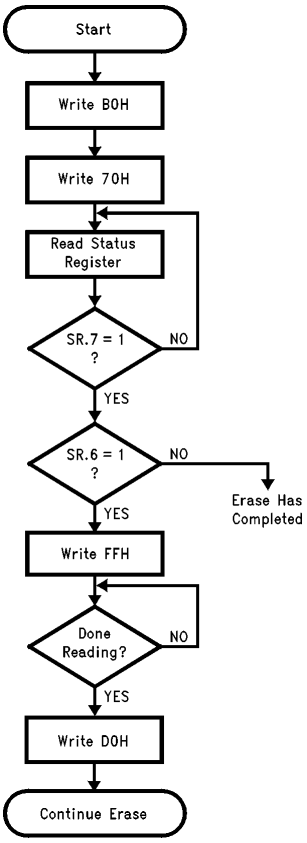
Bus Operation	Command	Comments
Write	Erase Setup	Data = 20H Address = Within block to be erased
Write	Erase Confirm	Data = D0H Address = Within block to be erased
Standby/ Read		Check RY/BY# V_{OH} = Ready, V_{OL} = Busy or Read Status Register Check SR.7 1 = Ready, 0 = Busy Toggle OE# or CE# to update Status Register
Repeat for subsequent blocks Full status check can be done after each block or after a sequence of blocks Write FFH after the last block erase operation to reset the device to Read Array Mode		



292099-5

Bus Operation	Command	Comments
Optional Read		CPU may already have read Status Register data in WSM Ready polling above
Standby		Check SR.3 1 = V_{PP} Low Detect
Standby		Check SR.4,5 Both 1 = Command Sequence Error
Standby		Check SR.5 1 = Block Erase Error
SR.3 MUST be cleared, if set during a block erase attempt, before further attempts are allowed by the Write State Machine SR.5 is 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		

Figure 3. Automated Block Erase Flowchart



Bus Operation	Command	Comments
Write	Erase Suspend	Data = B0H
Write	Read Status Register	Data = 70H
Standby Read		Check RY/BY# V_{OH} = Ready, V_{OL} = Busy or Read Status Register
		Check SR.7 1 = Ready, 0 = Busy Toggle OE# or CE# to Update Status Register
Standby		Check SR.6 1 = Suspended
Write	Read Array	Data = FFH
Read		Read array data from block other than that being erased.
Write	Erase Resume	Data = D0H

292099–6

Figure 4. Erase Suspend/Resume Flowchart

Since the WSM is driven by its own oscillator, internal to the 28F008SA, it operates asynchronously to the system CPU and its clock. Therefore, the possibility exists that the WSM could complete erase, returning to "ready", **between** when the system reads "busy" from the Status Register and writes the Erase Suspend command. Analyzing both the WSM Status and Erase Suspend Status bits of the Status Register, as shown in the flowchart, will alert the system to such an occurrence.

2.5 Write State Machine Current/Next State Overview

Byte write and block erase automation equate to tremendous power and capability in system implementations of the 28F008SA, if fully exploited. An in-depth understanding of the WSM, its states and its responses to inputs, will assist the software engineer in developing optimized driver routines for flash memory-based file storage and other high-performance applications. Table 3 lists all possible WSM "current states", command inputs and resultant "next states".

Non-shaded boxes highlight those state transitions which will most commonly occur when reading from and modifying 28F008SA contents, and these transitions should be understood in most depth. Shaded boxes, on the other hand, represent lesser-used or nonsensical transitions, such as improper erase command sequences.

Before reading the 28F008SA, if the current WSM mode is not known (if, for example, an interrupt service routine has potentially interacted with the device), first write the desired output command (i.e. Read Status Register, Read Array or Intelligent Identifier). This ensures that the 28F008SA will be in a known state when read and will output expected data.

Read Array

The 28F008SA automatically defaults to Read Array mode when powered up, or when it returns from Deep Powerdown mode. As the name implies, the 28F008SA outputs array data when read in Read Array mode. Read Array is also the default mode after the Clear Status Register command is written in most other modes.

Byte Write Setup

The 28F008SA transitions to Byte Write Setup mode after it receives the Byte Write Setup command. If the 28F008SA is read in Byte Write Setup mode, it outputs Status Register data.

Byte Write

After the 28F008SA is placed in Byte Write Setup mode, the next address/data combination written to it transitions the WSM to Byte Write mode, where the "Byte Write Command" is latched as desired data to write to the array at the specified address location. Immediately, the WSM examines V_{PP}, and if it detects an invalid level, it halts with V_{PP} error indication in the Status Register (bit 3 = "1"). Bit 7 of the Status Register is "0", and the RY/BY# output is driven to V_{OL}, while the WSM is executing the internal byte write algorithm in Byte Write mode. The 28F008SA automatically outputs Status Register data when in Byte Write mode.

Erase Setup

The 28F008SA transitions to Erase Setup mode after it receives the Erase Setup command. If the 28F008SA is read in Erase Setup mode, it outputs Status Register data.

Table 3. Write State Machine Current/Next States

Current State	RY/BY# Status	Data When Read	Command Input (and Next State)								
			Read Array (FFH)	Byte Write Setup (10/40H)	Erase Setup (20H)	Erase Confirm (D0H)	Erase Suspend (B0H)	Erase Resume (D0H)	Read Status (70H)	Clear Status (50H)	Read ID (90H)
Read Array	V_{OH}	Array	Read Array	Byte Write Setup	Erase Setup	Read Array	Read Array	Read Array	Read Status	Read Array	Read ID
Byte Write Setup	V_{OH}	Status	Byte Write (Command Input = Data to be Byte Written)								
Byte Write (Not Complete)	V_{OL}	Status	Byte Write								
Byte Write (Complete)	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Erase Setup	V_{OH}	Status	Erase Command Error			Erase	Erase Command Error	Erase	Erase Command Error		
Erase Command Error	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Erase (Not Complete)	V_{OL}	Status	Erase				Erase Suspend to Status	Erase			
Erase (Complete)	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Erase Suspend to Status	V_{OH}	Status	Erase Suspend to Array	Reserved	Erase Suspend to Array	Erase	Erase Suspend to Array	Erase	Erase Suspend to Status	Erase Suspend to Array	Reserved
Erase Suspend to Array	V_{OH}	Array	Erase Suspend to Array	Reserved	Erase Suspend to Array	Erase	Erase Suspend to Array	Erase	Erase Suspend to Status	Erase Suspend to Array	Reserved
Read Status	V_{OH}	Status	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID
Read Identifier	V_{OH}	ID	Read Array	Byte Write Setup	Erase Setup	Read Array			Read Status	Read Array	Read ID

NOTE:
1. State transitions labeled ‘‘Reserved’’ are set aside by Intel Corporation for potential future device implementations. Command sequences to access these states should not be attempted.

Erase

After the 28F008SA is placed in Erase Setup mode, write of the Erase Confirm command transitions the WSM to Erase mode, where the specified address is decoded into one of 16 blocks to be erased. Immediately, the WSM examines V_{PP}, and if it detects an invalid level, it halts with V_{PP} error indication in the Status Register (bit 3 = "1"). Bit 7 of the Status Register is "0", and the RY/BY# output is driven to V_{OL}, while the WSM is executing the internal block erase algorithm in Erase mode. The 28F008SA automatically outputs Status Register data when in Erase mode.

Erase Command Error

This is the other possible transition mode after Erase Setup, and occurs when an invalid command (anything but Erase Confirm/Resume) is written to the 28F008SA as the second in the two-command block erase sequence. In this mode, the WSM does not attempt a block erase, and it returns an error indication to the system by setting both bits 4 and 5 of the Status Register to "1". The 28F008SA automatically outputs Status Register data when in Erase Command Error mode.

Erase Suspend to Status/Array

While the WSM is busy executing an internal block erase algorithm, it can be placed in erase suspend by writing the Erase Suspend command. After receiving and decoding this command, the WSM suspends block erase, drives the RY/BY# output to V_{OH}, sets bits 6 and 7 of the Status Register to "1" and transitions to "Erase Suspend to Status" mode. The 28F008SA automatically outputs Status Register data when in "Erase Suspend to Status" mode.

The only valid command other than Read Status and Erase Resume at this time is Read Array, which transitions the WSM to "Erase Suspend to Array" mode. As the name implies, the 28F008SA outputs array data, not Status Register contents, in this mode. While in both Erase Suspend modes, V_{PP} must remain at V_{PPH} for erase to complete successfully when resumed.

Writing the Erase Resume (same as Erase Confirm) command to the 28F008SA transitions the WSM out of Erase Suspend and back to Erase. In conjunction with this, the WSM returns RY/BY# to V_{OL} and resets bits 6 and 7 of the WSM to "0".

Read Status

As the name implies, the 28F008SA automatically outputs Status Register contents when read in Read Status mode. If system software writes the Clear Status command at this point, the WSM resets the Status Register to its default value and transitions to Read Array mode.

Read Identifier

The 28F008SA outputs its manufacturer identifier of 89H when read from address 00000H when in Read Identifier mode. Similarly, a read from address 00001H returns the device identifier A2H. Using this information, the system can automatically match the device with its proper block erase and byte write algorithms. Reads from addresses other than 00000H and 00001H are not supported by Intel, and consistent results of such reads are not documented, guaranteed or recommended.

2.6 Block Erase as a Background Task

As mentioned earlier, the internal WSM block erase algorithm typically takes 1.6 seconds to complete. Proper implementation of block erase from a hardware and software standpoint, however, can mask this delay, by taking advantage of the 28F008SA's internal automation and full-featured system interface. Execution of block erase as a background task, with higher priority read and write functions in the foreground, is the key.

The recommended scenario includes an "intelligent" operating system routine which can keep track of "busy" devices in the 28F008SA array. After initiating block erase on these components, the operating system is free to concentrate on reads and writes, or any other pending requests that demand its attention. The 28F008SA RY/BY# output alerts the system when block erase completes, and the operating system acts on this completion in the resulting interrupt service routine.

Hardware interrupt via the RY/BY# output is a recommended technique for block erase. However, this method should be evaluated closely for alerting the system to byte write completion. The WSM typically completes a byte write attempt in 9 μs, a much shorter time than that consumed in many CPU interrupt latencies. In such cases, software polling of the 28F008SA Status Register to detect WSM "ready", versus hardware interrupt, will result in highest byte write performance. Reference AP-359, "28F008SA Hardware Interfacing", for circuit implementations that not only combine RY/BY#s into a common INT, but also allow RY/BY# masking if desired.

ADDITIONAL INFORMATION

		Order Number
28F008SA Datasheet		290429
28F008SA-L Datasheet		290435
AP-359	''28F008SA Hardware Interfacing''	292094
AP-360	''28F008SA Software Drivers''	292095
ER-27	''The Intel 28F008SA Flash Memory''	294011
ER-28	''ETOX-III Flash Memory Technology''	294012

REVISION HISTORY

Number	Description
003	PWD# pin renamed RP# to match JEDEC standards.

INTEL CORPORATION, 2200 Mission College Blvd., Santa Clara, CA 95052; Tel. (408) 765-8080

INTEL CORPORATION (U.K.) Ltd., Swindon, United Kingdom; Tel. (0793) 696 000

INTEL JAPAN k.k., Ibaraki-ken; Tel. 029747-8511