

2Mx64 3.3V Simultaneous Operation Flash Multi-Chip Package

GENERAL DESCRIPTION

The W72M64V-XBX consists of 4-32 megabit, 3.0 volt-only flash memory devices, organized as 2,097,152 words of 16 bits each. The device is designed to be programmed in-system with the standard 3.0 volt V_{CC} supply, and can also be programmed in standard EPROM programmers.

Standard control pins—chip enable CS#, write enable WE#, and output enable OE#—control normal read and write operations, and avoid bus contention issues.

The devices requires only a single 3.0 volt power supply for both read and write functions. Internally generated and regulated voltages are provided for the program and erase operations.

SIMULTANEOUS READ/WRITE OPERATIONS WITH ZERO LATENCY

The Simultaneous Read/Write architecture provides simultaneous operation by dividing the memory space into two banks. The device can improve overall system performance by allowing a host system to program or erase in one bank, then immediately and simultaneously read from the other bank, with zero latency. This releases the system from waiting for the completion of program or erase operations.

The W72M64V-XBX device uses the following bank architecture.

Bank 1	Bank 2
8	24

FEATURES

The SecSi. (Secured Silicon) Sector is an extra sector capable of being permanently locked by customers. The SecSi Indicator Bit (DQ7) is set to a 0 if customer lockable. Current device has 64 Kbytes per 2Mx 16 Flash chip.

The device offers complete compatibility with the JEDEC single-power-supply Flash command set standard. Commands are written to the command register using standard microprocessor write timings. Reading data out of the device is similar to reading from other Flash or EPROM devices.

The host system can detect whether a program or erase operation is complete by using the device status bits: RY/ BY# pin, DQ7 (Data# Polling) and DQ6/DQ2 (toggle bits). After a program or erase cycle has been completed, the device automatically returns to the read mode.

The sector erase architecture allows memory sectors to be erased and reprogrammed without affecting the data contents of other sectors. The device is fully erased when shipped from the factory.

Hardware data protection measures include a low V_{CC} detector that automatically inhibits write operations during power transitions. The hardware sector protection feature disables both program and erase operations in any combination of the sectors of memory. This can be achieved in-system or via programming equipment.

The device offers two power-saving features. When addresses have been stable for a specified amount of time, the device enters the automatic sleep mode. The system can also place the device into the standby mode. Power consumption is greatly reduced in both modes.

DEVICE BUS OPERATIONS

This section describes the requirements and use of the device bus operations, which are initiated through the internal command register. The command register itself does not occupy any addressable memory location. The register is a latch used to store the commands, along with the address and data information needed to execute the command. The contents of the register serve as inputs to the internal state machine. The state machine outputs dictate the function of the device. Table 1 lists the device bus operations, the inputs and control levels they require, and the resulting output. The following subsections describe each of these operations in further detail.

* This product is subject to change without notice.

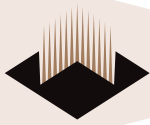


TABLE 1. DEVICE BUS OPERATIONS

Operation	CS#	OE#	WE#	RESET#	WP#/ACC	Addresses (2)	DQ0-DQ7	DQ8-DQ15
Read	L	L	H	H	L/H	AIN	D _{OUT}	D _{OUT}
Write	L	H	L	H	(3)	AIN	D _{IN}	D _{IN}
Standby	V _{CC} ± 0.3 V	X	X	V _{CC} ± 0.3 V	H	X	High-Z	High-Z
Output Disable	L	H	H	H	L/H	X	High-Z	High-Z
Reset	X	X	X	L	L/H	X	High-Z	High-Z
Sector Protect (1)	L	H	L	V _{ID}	L/H	SA, A6 = L, A1 = H, A0=L	D _{IN}	X
Sector Unprotect (1)	L	H	L	V _{ID}	(2)	SA, A6 = H, A1 = H, A0=L	D _{IN}	X
Temporary Sector Unprotect	X	X	X	V _{ID}	(2)	AIN	D _{IN}	D _{IN}

Legend:

L = Logic Low = V_A,
H = Logic High = V_{IH},
V_{ID} = 8.5-12.5 V,
V_{IH} = 9.0 ± 0.5 V,
X = Don't Care,
SA = Sector Address,
AIN = Address In,
D_{IN} = Data In,
D_{OUT} = Data Out

Notes:

1. The sector protect and sector unprotect functions may also be implemented via programming equipment. See the "Sector/Sector Block Protection and Unprotection" section.
2. If WP#/ACC = V_{IL}, the two outermost boot sectors remain protected. If WP#/ACC = V_{IH}, the two outermost boot sector protection depends on whether they were last protected or unprotected using the method described in "Sector/Sector Block Protection and Unprotection" If WP#/ACC = V_{IH}, all sectors will be unprotected.

REQUIREMENTS FOR READING ARRAY DATA

To read array data from the outputs, the system must drive the CS# and OE# pins to V_{IL}. CS# is the power control and selects the device. OE# is the output control and gates array data to the output pins. WE# should remain at V_{IH}.

The internal state machine is set for reading array data upon device power-up, or after a hardware reset. This ensures that no spurious alteration of the memory content occurs during the power transition. No command is necessary in this mode to obtain array data. Standard microprocessor read cycles that assert valid addresses on the device address inputs produce valid data on the device data outputs. Each bank remains enabled for read access until the command register contents are altered.

Refer to the AC Read-Only Operations table for timing specifications and to Figure 3 of the data sheet for the timing diagram. I_{CC1} in the DC Characteristics table represents the active current specification for reading array data.

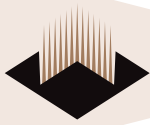
WRITING COMMANDS/COMMAND SEQUENCES

To write a command or command sequence (which includes programming data to the device and erasing sectors of memory), the system must drive WE# and CS# to V_{IL}, and OE# to V_{IH}.

The device features an Unlock Bypass mode to facilitate faster programming. Once a bank enters the Unlock Bypass mode, only two write cycles are required to program a word, instead of four.

An erase operation can erase one sector, multiple sectors, or the entire device. The device address space is divided into two banks: Bank 1 contains the boot/parameter sectors, and Bank 2 contains the larger, code sectors of uniform size. A "bank address" is the address bits required to uniquely select a bank. Similarly, a "sector address" is the address bits required to uniquely select a sector.

I_{CC2} in the DC Characteristics table represents the active current specification for the write mode. The AC Characteristics section contains timing specification tables and timing diagrams for write operations.



Accelerated Program Operation

The device offers accelerated program operations through the ACC function, provided by the WP#/ACC pin.

If the system asserts V_{HH} on this pin, the device automatically enters the aforementioned Unlock Bypass mode, temporarily unprotects any protected sectors, and uses the higher voltage on the pin to reduce the time required for program operations. The system would use a two-cycle program command sequence as required by the Unlock Bypass mode. Removing V_{HH} from the WP#/ACC pin returns the device to normal operation. Note that the WP#/ACC pin must not be at V_{HH} for operations other than accelerated programming, or device damage may result. In addition, the WP#/ACC pin must not be left floating or unconnected; inconsistent behavior of the device may result.

Autoselect Functions

If the system writes the autoselect command sequence, the device enters the autoselect mode. The system can then read autoselect codes from the internal register (which is separate from the memory array) on DQ7-DQ0. Standard read cycle timings apply in this mode. Refer to the Autoselect Mode and Autoselect Command Sequence sections for more information.

SIMULTANEOUS READ/WRITE OPERATIONS WITH ZERO LATENCY

This device is capable of reading data from one bank of memory while programming or erasing in the other bank of memory. An erase operation may also be suspended to read from or program to another location within the same bank (except the sector being erased). Figure 9 of the data sheet shows how read and write cycles may be initiated for simultaneous operation with zero latency. I_{CC6} and I_{CC7} in the DC Characteristics table represent the current specifications for read-while-program and read-while-erase, respectively.

STANDBY MODE

When the system is not reading or writing to the device, it can place the device in the standby mode. In this mode, current consumption is greatly reduced, and the outputs are placed in the high impedance state, independent of the OE# input.

The device enters the CMOS standby mode when the CS# and RESET# pins are both held at $V_{CC} \pm 0.3$ V. (Note that

this is a more restricted voltage range than V_{IH} .) If CS# and RESET# are held at V_{IH} , but not within $V_{CC} \pm 0.3$ V, the device will be in the standby mode, but the standby current will be greater. The device requires standard access time (t_{CE}) for read access when the device is in either of these standby modes, before it is ready to read data.

If the device is deselected during erasure or programming, the device draws active current until the operation is completed.

I_{CC3} in the DC Characteristics table of the datasheet represents the standby current specification.

AUTOMATIC SLEEP MODE

The automatic sleep mode minimizes Flash device energy consumption. The device automatically enables this mode when addresses remain stable for $t_{ACC} + 30$ ns. The automatic sleep mode is independent of the CS#, WE#, and OE# control signals. Standard address access timings provide new data when addresses are changed. While in sleep mode, output data is latched and always available to the system. I_{CC5} in the DC Characteristics table represents the automatic sleep mode current specification.

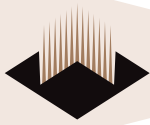
RESET#: HARDWARE RESET PIN

The RESET# pin provides a hardware method of resetting the device to reading array data. When the RESET# pin is driven low for at least a period of t_{RP} , the device immediately terminates any operation in progress, tristates all output pins, and ignores all read/write commands for the duration of the RESET# pulse. The device also resets the internal state machine to reading array data. The operation that was interrupted should be reinitiated once the device is ready to accept another command sequence, to ensure data integrity.

Current is reduced for the duration of the RESET# pulse. When RESET# is held at $V_{SS} \pm 0.3$ V, the device draws CMOS standby current (I_{CC4}). If RESET# is held at V_{IL} but not within $V_{SS} \pm 0.3$ V, the standby current will be greater.

The RESET# pin may be tied to the system reset circuitry. A system reset would thus also reset the Flash memory, enabling the system to read the boot-up firmware from the Flash memory.

If RESET# is asserted during a program or erase operation, the RY# pin remains a "0" (busy) until the internal reset operation is complete, which requires a time of t_{READY} (during Embedded Algorithms). The system can thus monitor RY#/BY to determine whether the reset operation



is complete. If RESET is asserted when a program or erase operation is not executing (RY/BY pin is “1”), the reset operation is completed within a time of t_{READY} (not during Embedded Algorithms). The system can read data t_{RH} after the RESET pin returns to V_{IH} .

Also refer to AC Characteristics tables for RESET timing parameters and to Figures 4 and 5 of the data sheet for the timing diagrams.

OUTPUT DISABLE MODE

When the OE input is at V_{IH} , output from the device is disabled. The output pins are placed in the high impedance state.

TABLE 2. DEVICE BANK DIVISIONS

Bank 1		Bank 2	
Megabits	Sector Sizes	Megabits	Sector Sizes
8 Mbit	Eight /4 Kword fifteen /32 Kword	24 Mbit	Forty-eight 32 Kword

TABLE 3 – BOTTOM BOOT SECTOR ADDRESSES

	Sector	Sector Address A20-A12	Sector Size (Kwords)	Address Range
Bank 1	SA0	00000000	4	000000h-000FFFh
	SA1	00000001	4	001000h-001FFFh
	SA2	00000010	4	002000h-002FFFh
	SA3	00000011	4	003000h-003FFFh
	SA4	00000100	4	004000h-004FFFh
	SA5	00000101	4	005000h-005FFFh
	SA6	00000110	4	006000h-006FFFh
	SA7	00000111	4	007000h-007FFFh
	SA8	00001xxx	32	008000h-00FFFFh
	SA9	000010xxx	32	010000h-017FFFh
	SA10	000011xxx	32	018000h-01FFFFh
	SA11	000100xxx	32	020000h-027FFFh
	SA12	000101xxx	32	028000h-02FFFFh
	SA13	000110xxx	32	030000h-037FFFh
	SA14	000111xxx	32	038000h-03FFFFh
	SA15	001000xxx	32	040000h-047FFFh
	SA16	001001xxx	32	048000h-04FFFFh
	SA17	001010xxx	32	050000h-057FFFh
	SA18	001011xxx	32	058000h-05FFFFh
	SA19	001100xxx	32	060000h-067FFFh
	SA20	001101xxx	32	068000h-06FFFFh
	SA21	001110xxx	32	070000h-077FFFh
SA22	001111xxx	32	078000h-07FFFFh	
Bank 2	SA23	010000xxx	32	080000h-087FFFh
	SA24	010001xxx	32	088000h-08FFFFh
	SA25	010010xxx	32	090000h-097FFFh
	SA26	010011xxx	32	098000h-09FFFFh
	SA27	010100xxx	32	0A0000h-0A7FFFh
	SA28	010101xxx	32	0A8000h-0AFFFFh
	SA29	010110xxx	32	0B0000h-0B7FFFh

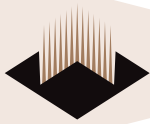


TABLE 3 – BOTTOM BOOT SECTOR ADDRESSES (cont'd)

	Sector	Sector Address A20-A12	Sector Size (Kwords)	Address Range
Bank 2	SA30	01011xxx	32	0B8000h-0BFFFFh
	SA31	011000xxx	32	0C0000h-0C7FFFh
	SA32	011001xxx	32	0C8000h-0CFFFFh
	SA33	011010xxx	32	0D0000h-0D7FFFh
	SA34	011011xxx	32	008000h-0DFFFFh
	SA35	011100xxx	32	0E0000h-0E7FFFh
	SA36	011101xxx	32	0E8000h-0EFFFFh
	SA37	011110xxx	32	0F0000h-0F7FFFh
	SA38	011111xxx	32	0F8000h-0FFFFFh
	SA39	100000xxx	32	100000h-107FFFh
	SA40	100001xxx	32	108000h-10FFFFh
	SA41	100010xxx	32	110000h-117FFFh
	SA42	100011xxx	32	118000h-11FFFFh
	SA43	100100xxx	32	120000h-127FFFh
	SA44	100101xxx	32	128000h-12FFFFh
	SA45	100110xxx	32	130000h-137FFFh
	SA46	100111xxx	32	138000h-13FFFFh
	SA47	101000xxx	32	140000h-147FFFh
	SA48	101001xxx	32	148000h-14FFFFh
	SA49	101010xxx	32	150000h-157FFFh
	SA50	101011xxx	32	158000h-15FFFFh
	SA51	101100xxx	32	160000h-167FFFh
	SA52	101101xxx	32	168000h-16FFFFh
	SA53	101110xxx	32	170000h-177FFFh
	SA54	101111xxx	32	178000h-17FFFFh
	SA55	111000xxx	32	180000h-187FFFh
	SA56	110001xxx	32	188000h-18FFFFh
	SA57	110010m(32	190000h-197FFFh
	SA58	110011xxx	32	198000h-19FFFFh
	SA59	110100xxx	32	1A0000h-1A7FFFh
SA60	110101xxx	32	1A8000h-1AFFFFh	
SA61	110110xxx	32	1B0000h-1B7FFFh	
SA62	110111xxx	32	1B8000h-1BFFFFh	
SA63	111000xxx	32	1C0000h-1C7FFFh	
SAM	111001xxx	32	1C8000h-1CFFFFh	
SA65	111010xxx	32	1D0000h-1D7FFFh	
SA66	111011xxx	32	1D8000h-1DFFFFh	
SA67	111100xxx	32	1E0000h-1E7FFFh	
SA68	111101xxx	32	1E8000h-1EFFFFh	
SA69	111110xxx	32	1F0000h-1F7FFFh	
SA70	111111xxx	32	1F8000h-1FFFFFh	

Note: The address range is A20:A0. The bank address bits are A20 and A19.

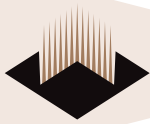


TABLE 4 – BOTTOM BOOT SECSITM SECTOR ADDRESSES

Sector Address A20-A12	Sector Size	Address Range
000000xxx	32	00000h-07FFFh

AUTOSELECT MODE

The autoselect mode provides manufacturer and device identification, and sector protection verification, through identifier codes output on eight least significant data pins for each die, (ie: die 1-DQ7-DQ0). This mode is primarily intended for programming equipment to automatically match a device to be programmed with its corresponding programming algorithm. However, the autoselect codes can also be accessed in-system through the command register.

When using programming equipment, the autoselect mode requires V_{ID} (8.5 V to 12.5 V) on address pin As. Address pins A6, A1, and A0 must be as shown in Table 5. In addition, when verifying sector protection, the sector

address must appear on the appropriate highest order address bits (see Table 3). Table 5 shows the remaining address bits that are don't care. When all necessary bits have been set as required, the programming equipment may then read the corresponding identifier code on eight least significant data pins for each die, (ie: die 1-DQ7-DQ0).

To access the autoselect codes in-system, the host system can issue the autoselect command via the command register, as shown in Table 7. This method does not require V_{ID}. Refer to the Autoselect Command Sequence section for more information.

TABLE 5 – AUTOSELECT CODES, (HIGH VOLTAGE METHOD)

Description	CS#	OE#	WE#	A20 to A12	A11 to A10	A9	A8 to A7	A6	A5 to A2	A1	A0	DQ8 to DQ15	DQ7 to DQ0
Sector Protection Verification	L	L	H	SA	X	V _{ID}	X	L	X	H	L	X	01h (protected), 00h (unprotected)
SecSl. Indicator Bit (DQ7)	L	L	H	BA	X	V _{ID}	X	L	X	H	H	X	81h (factory locked), 01h (not factory locked)

Legend: L = Logic Low = V_{IL}, H = Logic High = V_{IH}, BA = Bank Address, SA = Sector Address, X = Don't care.

SECTOR/SECTOR BLOCK PROTECTION AND UNPROTECTION

(Note: For the following discussion, the term “sector” applies to both sectors and sector blocks. A sector block consists of two or more adjacent sectors that are protected or unprotected at the same time (see Table 6).

The hardware sector protection feature disables both program and erase operations in any sector. The hardware sector unprotection feature re-enables both program and erase operations in previously protected sectors. Sector protection/unprotection can be implemented via two methods.

The primary method requires V_{ID} on the RESET# pin only, and can be implemented either in-system or via programming equipment. Figure 2 shows the algorithms and Figure 13 in the data sheet shows the timing diagram. This method uses standard microprocessor bus cycle timing.

For sector unprotect, all unprotected sectors must first be protected prior to the first sector unprotect write cycle.

The sector unprotect algorithm unprotects all sectors in parallel. All previously protected sectors must be individually re-protected. To change data in protected sectors efficiently, the temporary sector unprotect function is available. See “Temporary Sector Unprotect”.

The alternate method intended only for programming equipment requires V_{ID} on address pin A9 and OE#. This method is compatible with programmer routines written for earlier 3.0 volt-only flash devices.

The device is shipped with all sectors unprotected.

It is possible to determine whether a sector is protected or unprotected. See the Autoselect Mode section for details.



TABLE 6 – BOTTOM BOOT SECTOR/SECTOR BLOCK ADDRESSES FOR PROTECTION/UNPROTECTION

Sector	A20-A12	Sector/Sector Block Size
SA70	111111XXX	64 Kbytes
SA69-SA67	11111 0XXX, 111101XXX, 111100XXX	192 (3x64) Kbytes
SA66-SA63	1110XXXXX	256 (4x64) Kbytes
SA62-SA59	1101XXXXX	256 (4x64) Kbytes
SA58-SA55	1100XXXXX	256 (4x64) Kbytes
SA54-SA51	1011XXXXX	256 (4x64) Kbytes
SA50-SA47	1010XXXXX	256 (4x64) Kbytes
SA46-SA43	1001XXXXX	256 (4x64) Kbytes
SA42-SA39	1000XXXXX	256 (4x64) Kbytes
SA38-SA35	0111XXXXX	256 (4x64) Kbytes
SA34-SA31	0110XXXXX	256 (4x64) Kbytes
SA30-SA27	0101XXXXX	256 (4x64) Kbytes
SA26-SA23	0100XXXXX	256 (4x64) Kbytes
SA22-SA19	0011XXXXX	256 (4x64) Kbytes
SA18-SA15	0010XXXXX	256 (4x64) Kbytes
SA14-SA11	0001XXXXX	256 (4x64) Kbytes
SA10-SA8	000011XXX, 000010XXX, 000001XXX	192 (3x64) Kbytes
SA7	000000111	8 Kbytes
SA6	000000110	8 Kbytes
SA5	000000101	8 Kbytes
SA4	000000100	8 Kbytes
SA3	000000011	8 Kbytes
SA2	000000010	8 Kbytes
SA1	000000001	8 Kbytes
SA0	000000000	8 Kbytes

WRITE PROTECT (WP)

The Write Protect function provides a hardware method of protecting certain boot sectors without using V_{ID}. This function is one of two provided by the WP#/ACC pin.

If the system asserts V_{IL} on the WP#/ACC pin, the device disables program and erase functions in the two “outermost” 8 Kbyte boot sectors independently of whether those sectors were protected or unprotected using the method described in “Sector/Sector Block Protection and Unprotection”. The two outermost 8 Kbyte boot sectors are the two sectors containing the lowest addresses in a bottom-boot-configured device, or the two sectors containing the highest addresses in a topboot-configured device.

If the system asserts V_{IH} on the WP#/ACC pin, the device reverts to whether the two outermost 8K Byte boot sectors were last set to be protected or unprotected. That is, sector protection or unprotection for these two sectors depends on whether they were last protected or unprotected using the method described in “Sector/ Sector Block Protection and Unprotection”.

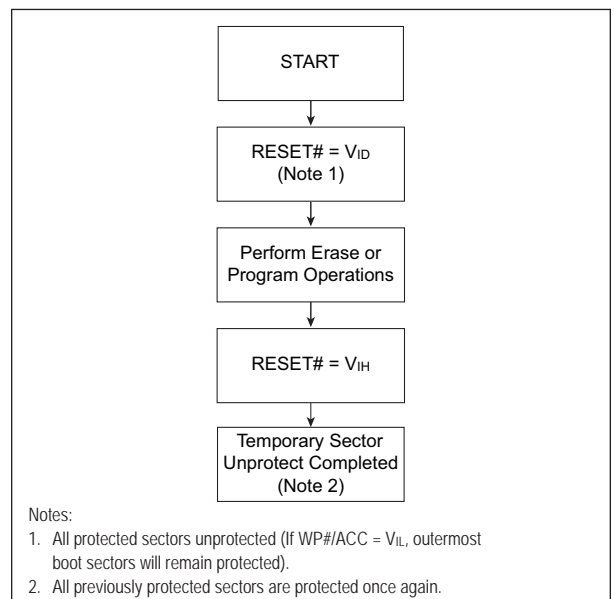
Note that the WP#/ACC pin must not be left floating or unconnected; inconsistent behavior of the device may result.

TEMPORARY SECTOR UNPROTECT

(Note: For the following discussion, the term “sector” applies to both sectors and sector blocks. A sector block consists of two or more adjacent sectors that are protected or unprotected at the same time (see Table 6).

This feature allows temporary unprotection of previously protected sectors to change data in-system. The Sector Unprotect mode is activated by setting the RESET# pin to V_{ID} (8.5 V - 12.5 V). During this mode, formerly protected sectors can be programmed or erased by selecting the sector addresses. Once V_{ID} is removed from the RESET# pin, all the previously protected sectors are protected again. Figure 1 shows the algorithm, and Figure 13 of the data sheet shows the timing diagrams, for this feature.

FIGURE 1 – TEMPORARY SECTOR UNPROTECT OPERATION



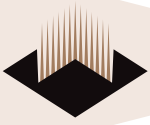
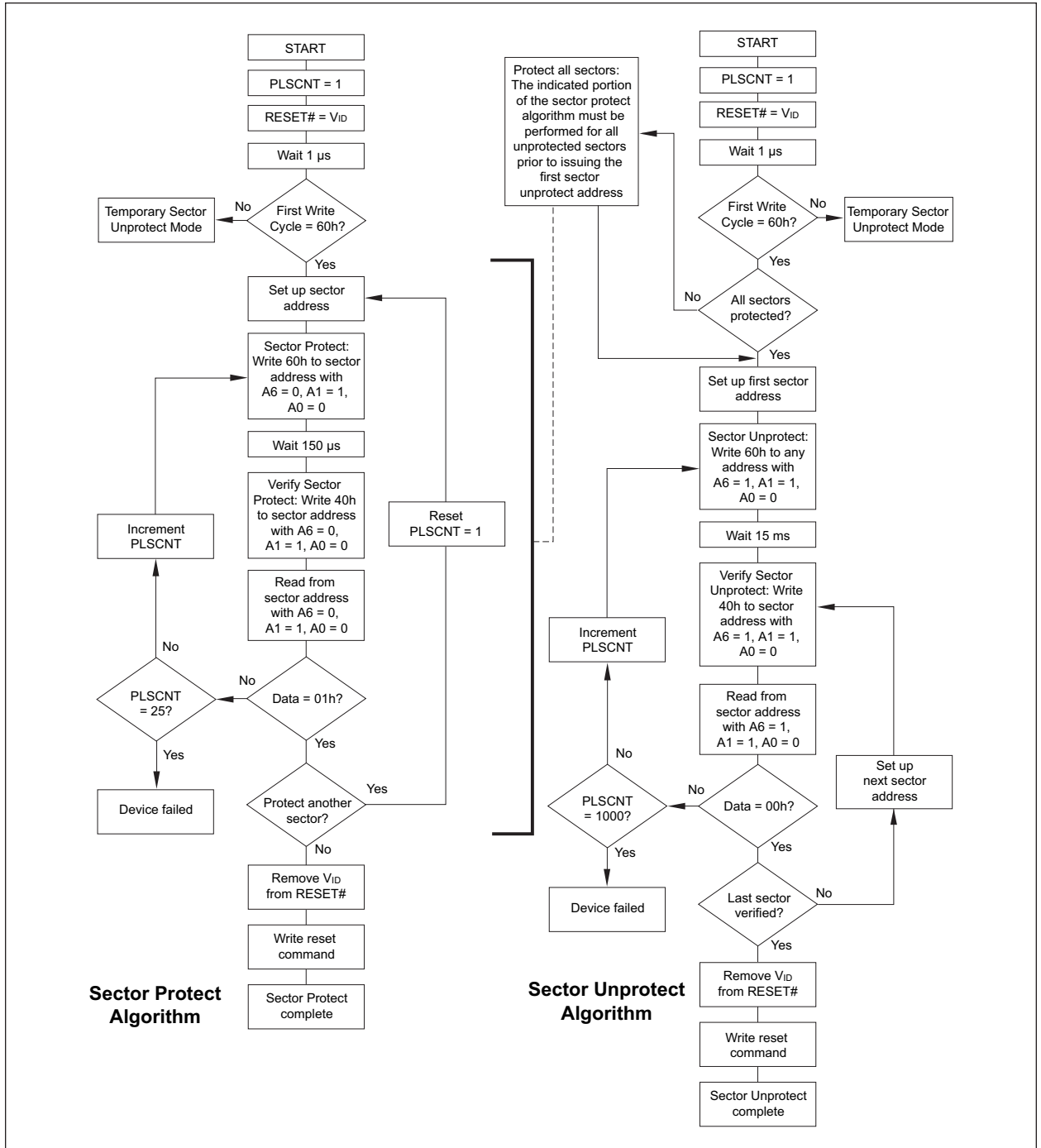
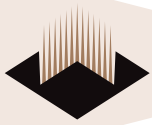


FIGURE 2 – IN-SYSTEM SECTOR PROTECTION/SECTOR UNPROTECTION ALGORITHMS





HARDWARE DATA PROTECTION

The command sequence requirement of unlock cycles for programming or erasing provides data protection against inadvertent writes (refer to Table 7 for command definitions). In addition, the following hardware data protection measures prevent accidental erasure or programming, which might otherwise be caused by spurious system level signals during V_{CC} power-up and power-down transitions, or from system noise.

Low V_{CC} Write Inhibit

When V_{CC} is less than V_{LKO} , the device does not accept any write cycles. This protects data during V_{CC} power-up and power-down. The command register and all internal program/erase circuits are disabled, and the device resets to the read mode. Subsequent writes are ignored until V_{CC} is greater than V_{LKO} . The system must provide the proper signals to the control pins to prevent unintentional writes when V_{CC} is greater than V_{LKO} .

Write Pulse “Glitch” Protection

Noise pulses of less than 5 ns (typical) on $OE\#$, $CS\#$ or $WE\#$ do not initiate a write cycle.

Logical Inhibit

Write cycles are inhibited by holding any one of $OE\# = V_{IL}$, $CS\# = V_{IH}$ or $WE\# = V_{IH}$. To initiate a write cycle, $CS\#$ and $WE\#$ must be a logical zero while $OE\#$ is logical one.

Power-Up Write Inhibit

If $WE\# = CS\# = V_{IL}$ and $OE\# = V_{IH}$ during power up, the device does not accept commands on the rising edge of $WE\#$. The internal state machine is automatically reset to the read mode on power-up.

COMMAND DEFINITIONS

Writing specific address and data commands or sequences into the command register initiates device operations. Table 7 defines the valid register command sequences. Writing incorrect address and data values or writing them in the improper sequence resets the device to reading array data.

All addresses are latched on the falling edge of $WE\#$ or $CS\#$ whichever happens later. All data is latched on the rising edge of $WE\#$ or $CS\#$, whichever happens first. Refer to the AC Characteristics section for timing diagrams.

Reading Array Data

The device is automatically set to reading array data after device power-up. No commands are required to

retrieve data. Each bank is ready to read array data after completing an Embedded Program or Embedded Erase algorithm.

After the device accepts an Erase Suspend command, the corresponding bank enters the erase-suspend-read mode, after which the system can read data from any non-erase-suspended sector within the same bank. After completing a programming operation in the Erase Suspend mode, the system may once again read array data with the same exception. See the Erase Suspend/ Erase Resume Commands section for more information.

The system *must* issue the reset command to return a bank to the read (or erase-suspend-read) mode if $DQ5$ goes high during an active program or erase operation, or if the bank is in the autoselect mode. See the next section, Reset Command, for more information.

See also Requirements for Reading Array Data in the Device Bus Operations section for more information. The Read-Only Operations table provides the read parameters, and Figure 3 of the data sheet shows the timing diagram.

Reset Command

Writing the reset command resets the banks to the read or erase-suspend-read mode. Address bits are don't cares for this command.

The reset command may be written between the sequence cycles in an erase command sequence before erasing begins. This resets the bank to which the system was writing to the read mode. Once erasure begins, however, the device ignores reset commands until the operation is complete.

The reset command may be written between the sequence cycles in a program command sequence before programming begins. This resets the bank to which the system was writing to the read mode. If the program command sequence is written to a bank that is in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode. Once programming begins, however, the device ignores reset commands until the operation is complete.

The reset command may be written between the sequence cycles in an autoselect command sequence. Once in the autoselect mode, the reset command must be written to return to the read mode. If a bank entered the autoselect mode while in the Erase Suspend mode, writing the reset command returns that bank to the erase-suspend-read mode.



If DQ5 goes high during a program or erase operation, writing the reset command returns the banks to the read mode (or erase-suspend-read mode if that bank was in Erase Suspend).

Autoselect Command Sequence

The autoselect command sequence allows the host system to access the manufacturer and device codes, and determine whether or not a sector is protected. Table 7 shows the address and data requirements. This method is an alternative to that shown in Table 5, which is intended for PROM programmers and requires V_{ID} on address pin A9. The autoselect command sequence may be written to an address within a bank that is either in the read or erase-suspend-read mode. The autoselect command may not be written while the device is actively programming or erasing in the other bank.

The autoselect command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle that contains the bank address and the autoselect command. The bank then enters the autoselect mode. The system may read at any address within the same bank any number of times without initiating another autoselect command sequence:

- A read cycle at address (BA)XX00h (where BA is the bank address) returns the manufacturer code.
- A read cycle at address (BA)XX01h returns the device code.
- A read cycle to an address containing a sector address (SA) within the same bank, and the address 02h on A7-A0 returns 01h if the sector is protected, or 00h if it is unprotected. (Refer to Table 5 for valid sector addresses).

The system must write the reset command to return to the read mode (or erase-suspend-read mode if the bank was previously in Erase Suspend).

PROGRAM COMMAND SEQUENCE

Programming is a four-bus-cycle operation. The program command sequence is initiated by writing two unlock write cycles, followed by the program set-up command. The program address and data are written next, which in turn initiate the Embedded Program algorithm. The system is not required to provide further controls or

timings. The device automatically provides internally generated program pulses and verifies the programmed cell margin. Table 7 shows the address and data

requirements for the byte program command sequence.

When the Embedded Program algorithm is complete, that bank then returns to the read mode and addresses are no longer latched. The system can determine the status of the program operation by using DQ7, DQ6, or RY/BY#. Refer to the Write Operation Status section for information on these status bits.

Any commands written to the device during the Embedded Program Algorithm are ignored. Note that a hardware reset immediately terminates the program operation. The program command sequence should be reinitiated once that bank has returned to the read mode, to ensure data integrity.

Programming is allowed in any sequence and across sector boundaries. A bit cannot be programmed from “0” back to a “1.” Attempting to do so may cause that bank to set DQ5 = 1, or cause the DQ7 and DQ6 status bits to indicate the operation was successful. However, a succeeding read will show that the data is still “0.” Only erase operations can convert a “0” to a “1.”

Unlock Bypass Command Sequence

The unlock bypass feature allows the system to program bytes or words to a bank faster than using the standard program command sequence. The unlock bypass command sequence is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the unlock bypass command, 20h. That bank then enters the unlock bypass mode. A two-cycle unlock bypass program command sequence is all that is required to program in this mode. The first cycle in this sequence contains the unlock bypass program command, A0h; the second cycle contains the program address and data. Additional data is programmed in the same manner. This mode dispenses with the initial two unlock cycles required in the standard program command sequence, resulting in faster total programming time. Table 7 shows the requirements for the command sequence.

During the unlock bypass mode, only the Unlock Bypass Program and Unlock Bypass Reset commands are valid. To exit the unlock bypass mode, the system must issue the two-cycle unlock bypass reset command sequence. See Table 7 for address and data requirements.

The device offers accelerated program operations through the WP#/ACC pin. When the system asserts V_{HH} on the WP#/ACC pin, the device automatically enters the Unlock Bypass mode. The system may then write the two-cycle Unlock Bypass program command sequence. The device



uses the higher voltage on the WP#/ACC pin to accelerate the operation. Note that the WP#/ACC pin must not be at V_{HH} any operation other than accelerated programming, or device damage may result. In addition, the WP#/ACC pin must not be left floating or unconnected; inconsistent behavior of the device may result.

Figure 3 illustrates the algorithm for the program operation. Refer to the Erase and Program Operations table in the AC Characteristics section for parameters, and Figure 6 of the data sheet for timing diagrams.

CHIP ERASE COMMAND SEQUENCE

Chip erase is a six bus cycle operation. The chip erase command sequence is initiated by writing two unlock cycles, followed by a set-up command. Two additional unlock write cycles are then followed by the chip erase command, which in turn invokes the Embedded Erase algorithm. The device does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically preprograms and verifies the entire memory for an all zero data pattern prior to electrical erase. The system is not required to provide any controls or timings during these operations. Table 7 shows the address and data requirements for the chip erase command sequence.

When the Embedded Erase algorithm is complete, that bank returns to the read mode and addresses are no longer latched. The system can determine the status of the erase operation by using DQ7, DQ6, DQ2, or RY/BY#. Refer to the Write Operation Status section for information on these status bits.

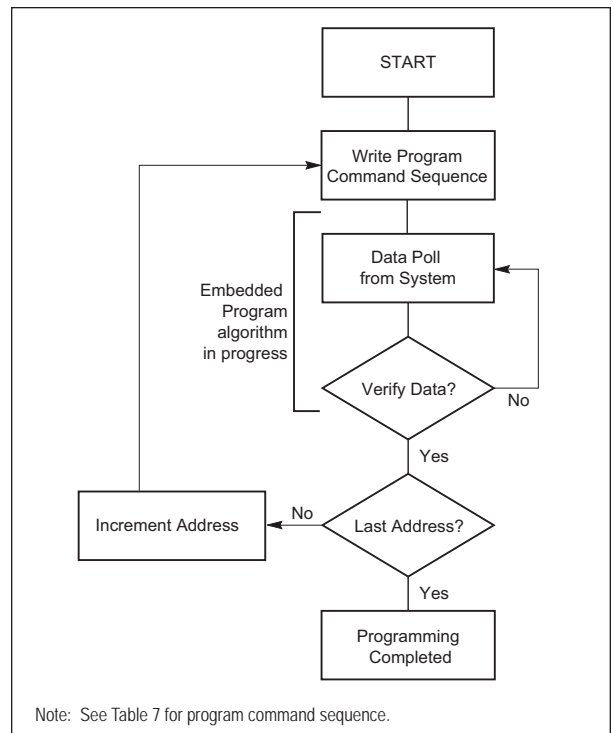
Any commands written during the chip erase operation are ignored. However, note that a hardware reset immediately terminates the erase operation. If that occurs, the chip erase command sequence should be reinitiated once that bank has returned to reading array data, to ensure data integrity.

Figure 4 illustrates the algorithm for the erase operation. Refer to the Erase and Program Operations tables in the AC Characteristics section for parameters, and Figure 8 of the data sheet for timing diagrams.

SECTOR ERASE COMMAND SEQUENCE

Sector erase is a six bus cycle operation. The sector erase command sequence is initiated by writing two unlock cycles, followed by a set-up command. Two additional

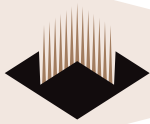
FIGURE 3 – PROGRAM OPERATION



unlock cycles are written, and are then followed by the address of the sector to be erased, and the sector erase command. Table 7 shows the address and data requirements for the sector erase command sequence.

The device does not require the system to preprogram prior to erase. The Embedded Erase algorithm automatically programs and verifies the entire memory for an all zero data pattern prior to electrical erase. The system is not required to provide any controls or timings during these operations.

After the command sequence is written, a sector erase time-out of 50 μ s occurs. During the time-out period, additional sector addresses and sector erase commands (for sectors within the same bank) may be written. Loading the sector erase buffer may be done in any sequence, and the number of sectors may be from one sector to all sectors. The time between these additional cycles must be less than 50 μ s, otherwise erasure may begin. Any sector erase address and command following the exceeded time-out may or may not be accepted. It is recommended



that processor interrupts be disabled during this time to ensure all commands are accepted. The interrupts can be re-enabled after the last Sector Erase command is written. Any command other than Sector Erase or Erase Suspend during the time-out period resets that bank to the read mode. The system must rewrite the command sequence and any additional addresses and commands.

The system can monitor DQ3 to determine if the sector erase timer has timed out (See the section on DQ3: Sector Erase Timer.). The time-out begins from the rising edge of the final WE# pulse in the command sequence.

When the Embedded Erase algorithm is complete, the bank returns to reading array data and addresses are no longer latched. Note that while the Embedded Erase operation is in progress, the system can read data from the non-erasing bank. The system can determine the status of the erase operation by reading DQ7, DQ6, DQ2, or RY/BY# in the erasing bank. Refer to the Write Operation Status section for information on these status bits.

Once the sector erase operation has begun, only the Erase Suspend command is valid. All other commands are ignored. However, note that a hardware reset immediately terminates the erase operation. If that occurs, the sector erase command sequence should be reinitiated once that bank has returned to reading array data, to ensure data integrity.

Figure 4 illustrates the algorithm for the erase operation. Refer to the Erase and Program Operations tables in the AC Characteristics section for parameters, and Figure 8 of the data sheet for timing diagrams.

ERASE SUSPEND/ERASE RESUME COMMANDS

The Erase Suspend command, B0h, allows the system to interrupt a sector erase operation and then read data from, or program data to, any sector not selected for erasure. The bank address is required when writing this command. This command is valid only during the sector erase operation, including the 50 μ s time-out period during the sector erase command sequence.

The Erase Suspend command is ignored if written during the chip erase operation or Embedded Program algorithm.

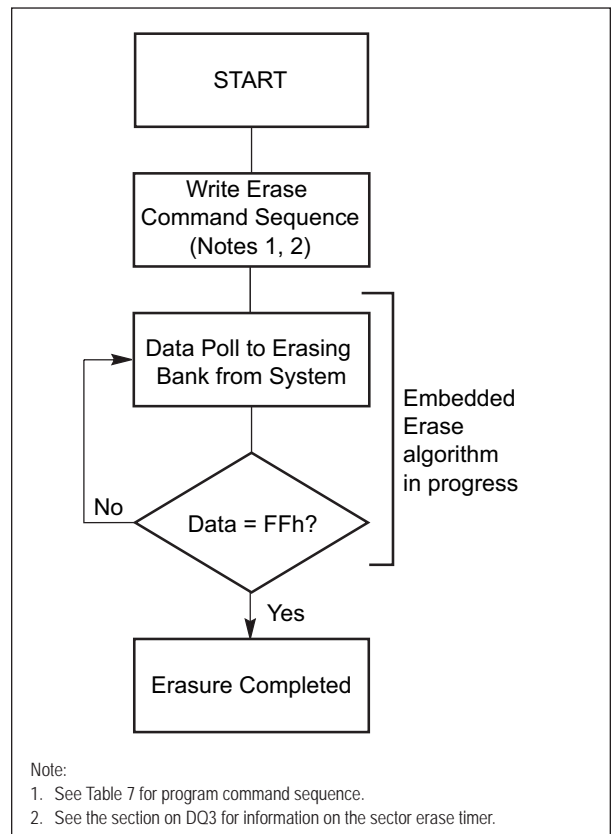
When the Erase Suspend command is written during the sector erase operation, the device requires a maximum of 20 μ s to suspend the erase operation. However, when the Erase Suspend command is written during the sector erase

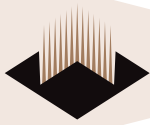
time-out, the device immediately terminates the time-out period and suspends the erase operation.

After the erase operation has been suspended, the bank enters the erase-suspend-read mode. The system can read data from or program data to any sector not selected for erasure. (The device “erase suspends” all sectors selected for erasure.) Reading at any address within erase-suspended sectors produces status information on DQ7-DQ0. The system can use DQ7, or DQ6 and DQ2 together, to determine if a sector is actively erasing or is erase-suspended. Refer to the Write Operation Status section for information on these status bits.

After an erase-suspended program operation is complete, the bank returns to the erase-suspend-read mode. The system can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard Byte Program operation. Refer to the Write Operation

FIGURE 4 – ERASE OPERATION





Status section for more information.

In the erase-suspend-read mode, the system can also issue the autoselect command sequence. Refer to the Autoselect Mode and Autoselect Command Sequence sections for details.

To resume the sector erase operation, the system must write the Erase Resume command. The bank address of the erase-suspended bank is required when writing this command. Further writes of the Resume command are ignored. Another Erase Suspend command can be written after the chip has resumed erasing.

TABLE 7 – COMMAND DEFINITIONS

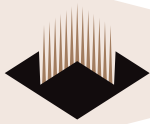
Command Sequence (1)	Cycles	Bus Cycles (2-5)												
		First		Second		Third		Fourth		Fifth		Sixth		
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	
Read (6)	1	RA	RD											
Reset (7)	1	XXX	F0											
Autoselect	Sector/Sector Block Protect Verify (8)	4	555	AA	2AA	55	(BA)555	90	(SA)X02	00/01				
Program		4	555	AA	2AA	55	555	AO	PA	PD				
Unlock Bypass		3	555	AA	2AA	55	555	20						
Unlock Bypass Program (9)		2	XXX	A0	PA	PD								
Unlock Bypass Reset M (10)		2	XXX	90	XXX	00								
Chip Erase		6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10
Sector Erase		6	555	AA	2AA	55	555	80	555	AA	2AA	55	SA	30
Erase Suspend (11)		1	BA	B0										
Erase Resume (12)		1	BA	30										

Legend:

- X = Don't care
- RA = Address of the memory location to be read.
- RD = Data read from location RA during read operation.
- PA = Address of the memory location to be programmed. Addresses latch on the falling edge of the WE# or CS# pulse, whichever happens later.
- PD = Data to be programmed at location PA. Data latches on the rising edge of WE# or CS# pulse, whichever happens first.
- SA = Address of the sector to be verified (in toselect mode) or erased. Address bits A20-A12 uniquely select any sector.
- BA = Address of the bank that is being switched to autoselect mode, is in bypass mode, or is being erased.

Notes:

1. See Table 1 for description of bus operations.
2. All values are in hexadecimal.
3. Except for the read cycle and the fourth cycle of the autoselect command sequence, all bus cycles are write cycles.
4. Data bits DQ15-DQ8 are don't care in command sequences, except for RD and PD.
5. Unless otherwise noted, address bits A20-A11 are don't cares.
6. No unlock or command cycles required when bank is reading array data.
7. The Reset command is required to return to the read mode (or to the erase-suspend-read mode if previously in Erase Suspend) when a bank is in the autoselect mode, or if DQ5 goes high (while the bank is providing status information).
8. The data is 00h for an unprotected sector/sector block and 01h for a protected sector/sector block.
9. The Unlock Bypass command is required prior to the Unlock Bypass Program command.
10. The Unlock Bypass Reset command is required to return to the read mode when the bank is in the unlock bypass mode.
11. The system may read and program in non-erasing sectors, or enter the autoselect mode, when in the Erase Suspend mode. The Erase Suspend command is valid only during a sector erase operation, and requires the bank address.
12. The Erase Resume command is valid only during the Erase Suspend mode, and requires the bank address.



WRITE OPERATION STATUS

The device provides several bits to determine the status of a program or erase operation: DQ2, DQ3, DQ5, DQ6, and DQ7. Table 8 and the following subsections describe the function of these bits. DQ7 and DQ6 each offer a method for determining whether a program or erase operation is complete or in progress. The device also provides a hardware-based output signal, RY/BY#, to determine whether an Embedded Program or Erase operation is in progress or has been completed.

DQ7: DATA# POLLING

The Data# Polling bit, DQ7, indicates to the host system whether an Embedded Program or Erase algorithm is in progress or completed, or whether a bank is in Erase Suspend. Data# Polling is valid after the rising edge of the final WE# pulse in the command sequence.

During the Embedded Program algorithm, the device outputs on DQ7 the complement of the datum programmed to DQ7. This DQ7 status also applies to programming during Erase Suspend. When the Embedded Program algorithm is complete, the device outputs the datum programmed to DQ7. The system must provide the program address to read valid status information on DQ7. If a program address falls within a protected sector, Data# Polling on DQ7 is active for approximately 1 ps, then that bank returns to the read mode.

During the Embedded Erase algorithm, Data# Polling produces a “0” on DQ7. When the Embedded Erase algorithm is complete, or if the bank enters the Erase Suspend mode, Data Polling produces a “1” on DQ7. The system must provide an address within any of the sectors selected for erasure to read valid status information on DQ7.

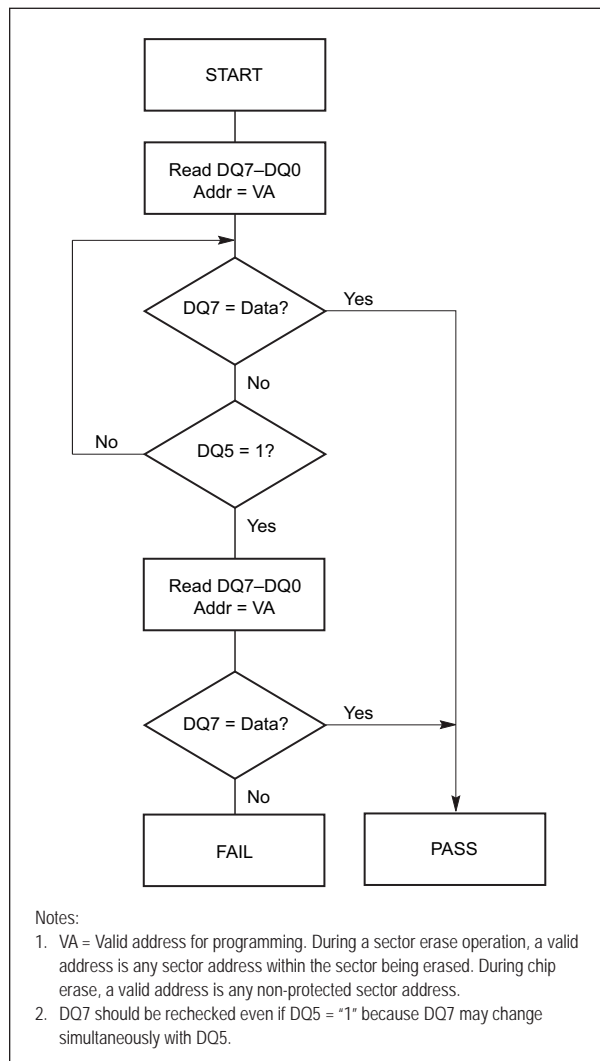
After an erase command sequence is written, if all sectors selected for erasing are protected, Data# Polling on DQ7 is active for approximately 100 μ s, then the bank returns to the read mode. If not all selected sectors are protected, the Embedded Erase algorithm erases the unprotected sectors, and ignores the selected sectors that are protected. However, if the system reads DQ7 at an address within a protected sector, the status may not be valid.

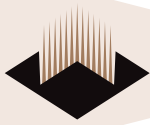
Just prior to the completion of an Embedded Program or Erase operation, DQ7 may change asynchronously with DQ0-DQ6 while Output Enable (OE#) is asserted low. That is, the device may change from providing status

information to valid data on DQ7. Depending on when the system samples the DQ7 output, it may read the status or valid data. Even if the device has completed the program or erase operation and DQ7 has valid data, the data outputs on DQ0-DQ6 may be still invalid. Valid data on DQ0-DQ7 will appear on successive read cycles.

Table 8 shows the outputs for Data# Polling on DQ7. Figure 5 shows the Data# Polling algorithm. Figure 10 in the data sheet shows the Data# Polling timing diagram.

FIGURE 5 – DATA# POLLING ALGORITHM





RY/BY#: READY/BUSY#

The RY/BY# is a dedicated, open-drain output pin which indicates whether an Embedded Algorithm is in progress or complete. The RY/BY# status is valid after the rising edge of the final WE# pulse in the command sequence. Since RY/BY# is an open-drain output, several RY/BY# pins can be tied together in parallel with a pull-up resistor to V_{CC}.

If the output is low (Busy), the device is actively erasing or programming. (This includes programming in the Erase Suspend mode.) If the output is high (Ready), the device is in the read mode, the standby mode, or one of the banks is in the erase-suspend- read mode.

Table 8 shows the outputs for RY/BY#.

DQ6: TOGGLE BIT I

Toggle Bit I on DQ6 indicates whether an Embedded Program or Erase algorithm is in progress or complete, or whether the device has entered the Erase Suspend mode. Toggle Bit I may be read at any address, and is valid after the rising edge of the final WE# pulse in the command sequence (prior to the program or erase operation), and during the sector erase time-out.

During an Embedded Program or Erase algorithm operation, successive read cycles to any address cause DQ6 to toggle. The system may use either OE# or CS# to control the read cycles. When the operation is complete, DQ6 stops toggling.

After an erase command sequence is written, if all sectors selected for erasing are protected, DQ6 toggles for approximately 100 μs, then returns to reading array data. If not all selected sectors are protected, the Embedded Erase algorithm erases the unprotected sectors, and ignores the selected sectors that are protected.

The system can use DQ6 and DQ2 together to determine whether a sector is actively erasing or is erase-suspended. When the device is actively erasing (that is, the Embedded Erase algorithm is in progress), DQ6 toggles. When the device enters the Erase Suspend mode, DQ6 stops toggling. However, the system must also use DQ2 to determine which sectors are erasing or erase-suspended. Alternatively, the system can use DQ7 (see the subsection on DQ7: Data# Polling).

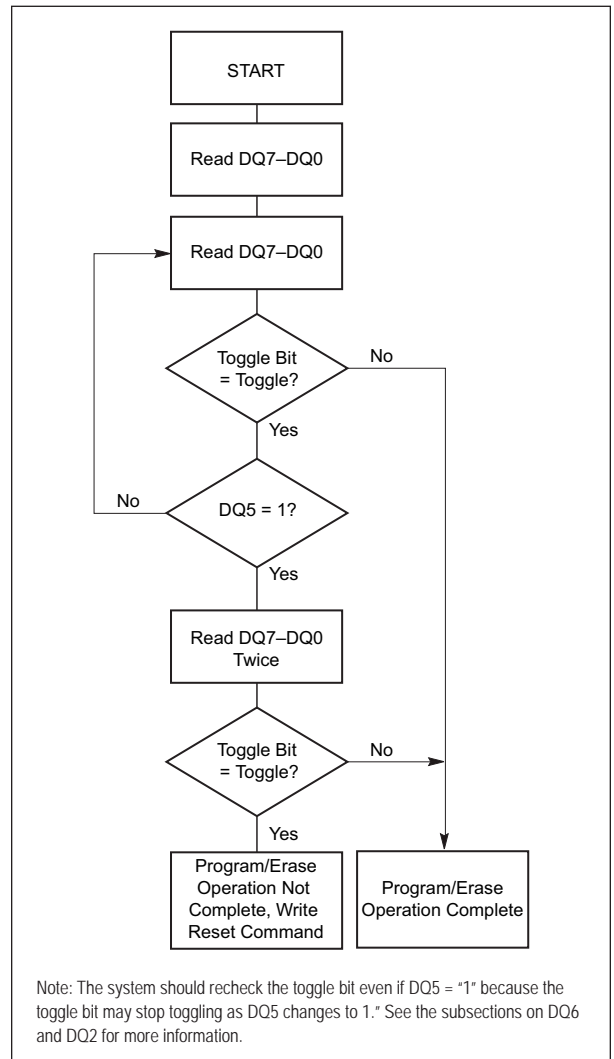
If a program address falls within a protected sector, DQ6 toggles for approximately 1 μs after the program command sequence is written, then returns to reading array data.

DQ6 also toggles during the erase-suspend-program

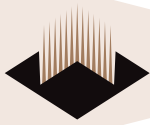
mode, and stops toggling once the Embedded Program algorithm is complete.

Table 8 shows the outputs for Toggle Bit I on DQs. Figure 6 shows the toggle bit algorithm. Figure 11 in the data sheet shows the toggle bit timing diagrams. Figure 12 shows the differences between DQ2 and DQ6 in graphical form. See also the subsection on DQ2: Toggle Bit II.

FIGURE 6 – TOGGLE BIT ALGORITHM



Note: The system should recheck the toggle bit even if DQ5 = "1" because the toggle bit may stop toggling as DQ5 changes to 1." See the subsections on DQ6 and DQ2 for more information.



DQ2: TOGGLE BIT II

The “Toggle Bit II” on DQ2, when used with DQ6, indicates whether a particular sector is actively erasing (that is, the Embedded Erase algorithm is in progress), or whether that sector is erase-suspended. Toggle Bit II is valid after the rising edge of the final WE pulse in the command sequence.

DQ2 toggles when the system reads at a address within those sectors that have been selected for erasure. (The system may use either OE# or CS# to control the read cycles.) But DQ2 cannot distinguish whether the sector is actively erasing or is erase-suspended. DQ6, by comparison, indicates whether the device is actively erasing, or is in Erase Suspend, but cannot distinguish which sectors are selected for erasure. Thus, both status bits are required for sector and mode information. Refer to Table 8 to compare outputs for DQ2 and DQ6.

Figure 6 shows the toggle bit algorithm in flowchart form, and the section “DQ2: Toggle Bit II” explains the algorithm. See also the DQ6: Toggle Bit I subsection. Figure 11 of the data sheet shows the toggle bit timing diagram. Figure 12 of the data sheet shows the differences between DQ2 and DQ6 in graphical form.

READING TOGGLE BITS DQ6/DQ2

Refer to Figure 6 for the following discussion. Whenever the system initially begins reading toggle bit status, it must read DQ7-DQ0 at least twice in a row to determine whether a toggle bit is toggling. Typically, the system would note and store the value of the toggle bit after the first read. After the second read, the system would compare the new value of the toggle bit with the first. If the toggle bit is not toggling, the device has completed the program or erase operation. The system can read array data on DQ7-DQ0 on the following read cycle.

However, if after the initial two read cycles, the system determines that the toggle bit is still toggling, the system also should note whether the value of DQ5 is high (see the section on DQ5). If it is, the system should then determine again whether the toggle bit is toggling, since the toggle bit may have stopped toggling just as DQ5 went high. If the toggle bit is no longer toggling, the device has successfully completed the program or erase operation. If it is still toggling, the device did not complete the operation successfully, and the system must write the reset command to return to reading array data.

The remaining scenario is that the system initially

determines that the toggle bit is toggling and DQ5 has not gone high. The system may continue to monitor the toggle bit and DQ5 through successive read cycles, determining the status as described in the previous paragraph. Alternatively, it may choose to perform other system tasks. In this case, the system must start at the beginning of the algorithm when it returns to determine the status of the operation (top of Figure 6).

DQ5: EXCEEDED TIMING LIMITS

DQ5 indicates whether the program or erase time has exceeded a specified internal pulse count limit. Under these conditions DQ5 produces a “1,” indicating that the program or erase cycle was not successfully completed.

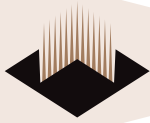
The device may output a “1” on DQ5 if the system tries to program a “1” to a location that previously programmed to “0.” **Only an erase operation can change a “0” back to a “1.”** Under this condition, the device halts the operation, and when the timing limit has been exceeded, DQ5 produces a “1.”

Under both these conditions, the system must write the reset command to return to the read mode (or to the erase-suspend-read mode if a bank was previously in the erase-suspend-program mode).

DQ3: SECTOR ERASE TIMER

After writing a sector erase command sequence, the system may read DQ3 to determine whether or not erasure has begun. (The sector erase timer does not apply to the chip erase command.) If additional sectors are selected for erasure, the entire time-out also applies after each additional sector erase command. When the time-out period is complete, DQ3 switches from a “0” to a “1.” If the time between additional sector erase commands from the system can be assumed to be less than 50 μ s, the system need not monitor DQ3. See also the Sector Erase Command Sequence section.

After the sector erase command is written, the system should read the status of DQ7 (Data# Polling) or DQ6 (Toggle Bit I) to ensure that the device has accepted the command sequence, and then read DQ3. If DQ3 is “1,” the Embedded Erase algorithm has begun; all further commands (except Erase Suspend) are ignored until the erase operation is complete. If DQ3 is “0,” the device will accept additional sector erase commands. To ensure the command has been accepted, the system software should check the status of DQ3 prior to and following



each subsequent sector erase command. If DQ3 is high on the second status check, the last command might not have been accepted.

Table 8 shows the status of DQ3 relative to the other status bits.

TABLE 8 – WRITE OPERATION STATUS

Status		DQ7# (2)	DQ6	DQ5 (1)	DQ3	DQ2 (2)	RY/BY#	
Standard Mode	Embedded Program Algorithm	DQ7	Toggle	0	N/A	No toggle	0	
	Embedded Erase Algorithm	0	Toggle	0	1	Toggle	0	
Erase Suspend Mode	Erase-Suspend-	Erase Suspended Sector	1	No toggle	0	N/A	Toggle	1
	Read	Non-Erase Suspended Sector	Data	Data	Data	Data	Data	1
	Erase-Suspend-Program		DQ7#	Toggle	0	N/A	N/A	0

Notes:

1. DQ5 switches to '1' when an Embedded Program or Embedded Erase operation has exceeded the maximum timing limits. Refer to the section on DQ5 for more information.
2. DQ7 and DQ2 require a valid address when reading status information. Refer to the appropriate subsection for further details.
3. When reading write operation status bits, the system must always provide the bank address where the Embedded Algorithm is in progress. The device outputs array data if the system addresses a non-busy bank.