APPLICATION NOTE

# FLASH MEMORY OPERATION 4M5

**FOR THE FOLLOWING PARTS:**
WMF512K8-XXX5 • WF512K32-XXX5 • WF512K64-XXX5 • WSF512K16-XXX • WSF41632-22XX • WSF512K32-XXX

## PRINCIPLES OF OPERATION

For example, Chip 1 is distinguished by $\overline{\text{CS}}$1 and I/O0-7, Chip 2 by $\overline{\text{CS}}$2 and I/O8-15, Chip 3 by $\overline{\text{CS}}$3 and I/O16-23, and Chip 4 by $\overline{\text{CS}}$4 and I/O24-31 in an MCM built with four die and with a 32 bit wide data bus.

**Programming** is accomplished by executing the program command sequence. The program algorithm, which is an internal algorithm, automatically times the program pulse widths and verifies proper cell status. Sectors can be programmed and verified in less than one second. Erase is accomplished by executing the erase command sequence. The internal erase algorithm automatically preprograms the array if it is not already programmed before executing the erase operation. During erase, the device automatically times the erase pulse widths and verifies proper cell status. This device also features a sector erase architecture. The sector mode allows for 64K byte blocks of memory to be erased and reprogrammed without affecting ofher blocks. The devices are erased prior to shipment.

## BUS OPERATIONS

### Read

Chip-Select ($\overline{\text{CS}}$) and Output Enable ($\overline{\text{OE}}$) must be logically active, to obtain data at the outputs. $\overline{\text{CS}}$ is the power control and should be used for device selection. $\overline{\text{OE}}$ is the output control and should be used to gate data to the output pins of the chip selected.

### Output Disable

With Output-Enable at a logic-high level ($V_{IH}$), output from the device is disabled. Output pins are placed in a high impedance state.

### Standby Mode

In the standby mode the outputs are in a high impedance state, independent of the $\overline{\text{OE}}$ input.

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

### Write

Device erasure and programming are accomplished via the command register. The contents of the register serve as input to the internal state machine. The state machine outputs dictate the function of the device.

The command register itself does not occupy an addressable memory location. The register is a latch used to store the command, along with address and data information needed to execute the command. The command register is written by bringing $\overline{\text{WE}}$ to a logic-low level ($V_{IL}$), while $\overline{\text{CS}}$ is low and $\overline{\text{OE}}$ is at $V_{IH}$. Addresses are latched on the falling edge of $\overline{\text{WE}}$ or $\overline{\text{CS}}$, whichever happens later. Data is latched on the rising edge of the $\overline{\text{WE}}$ or $\overline{\text{CS}}$ whichever occurs first. Standard microprocessor write timings are used.

## COMMAND DEFINITIONS

Device operations are selected by writing specific address and data sequences into the command register. Table 3 defines these register command sequences.

### Read/Reset Command

The read or reset operation is initiated by writing the read/reset command sequence into the command register. Microprocessor read cycles retrieve array data from the memory. The device remains enabled for reads until the command register contents are altered.

The device will automatically power-up in the read/reset state. In this case, a command sequence is not required to read data. Standard microprocessor read cycles will retrieve array data. This default value ensures that no spurious alteration of the memory content occurs during the power transition. Refer to the AC Read Characteristics and Waveforms for the specific timing parameters.

### TABLE 1 - BUS OPERATIONS

| Operation | $\overline{CS}$ | $\overline{OE}$ | $\overline{WE}$ | A0 | A1 | A6 | A9 | I/O |
|---|---|---|---|---|---|---|---|---|
| Read | L | L | H | $A_0$ | $A_1$ | $A_6$ | $A_9$ | DOUT |
| Standby | H | X | X | X | X | X | X | HIGH Z |
| Output Disable | L | H | H | X | X | X | X | HIGH Z |
| Write | L | H | L | $A_0$ | $A_1$ | $A_6$ | $A_9$ | $D_{IN}$ |

### TABLE 2 - SECTOR ADDRESSES TABLE

| | A18 | A17 | A16 | Addr Range |
|---|---|---|---|---|
| SA0 | 0 | 0 | 0 | 00000h-0FFFFh |
| SA1 | 0 | 0 | 1 | 10000h-1FFFFh |
| SA2 | 0 | 1 | 0 | 20000h-2FFFFh |
| SA3 | 0 | 1 | 1 | 30000h-3FFFFh |
| SA4 | 1 | 0 | 0 | 40000h-4FFFFh |
| SA5 | 1 | 0 | 1 | 50000h-5FFFFh |
| SA6 | 1 | 1 | 0 | 60000h-6FFFFh |
| SA7 | 1 | 1 | 1 | 70000h-7FFFFh |

### TABLE 3 - COMMAND DEFINITIONS

| Command Sequence | Bus Write Cycles Req'd | First Bus Write Cycle | | Second Bus Write Cycle | | Third Bus Write Cycle | | Fourth Bus Read/Write Cycle | | Fifth Bus Write Cycle | | Sixth Bus Write Cycle | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| Read/Reset | 1 | XXXH | F0H | | | | | | | | | | |
| Read/Reset | 4 | 5555H | AAH | 2AAAH | 55H | 5555H | F0H | RA | RD | | | | |
| Autoselect | 4 | 5555H | AAH | 2AAAH | 55H | 5555H | 90H | 00H | 01H | | | | |
| | | | | | | | | 01H | A4H | | | | |
| Byte Program | 4 | 5555H | AAH | 2AAAH | 55H | 5555H | A0H | PA | PD | | | | |
| Chip Erase | 6 | 5555H | AAH | 2AAAH | 55H | 5555H | 80H | 5555H | AAH | 2AAAH | 55H | 5555H | 10H |
| Sector Erase | 6 | 5555H | AAH | 2AAAH | 55H | 5555H | 80H | 5555H | AAH | 2AAAH | 55H | SA | 30H |
| Sector Erase Suspended | Erase can be suspended during sector erase with Addr (don't care), Data (B0H) | | | | | | | | | | | | |
| Sector Erase Resume | Erase can be resumed after suspended with Addr (don't care), Data (30H) | | | | | | | | | | | | |

**NOTES:**
1. Address bit $A_{15}$, $A_{16}$, $A_{17}$, and $A_{18}$ = X = Don't Care. Write Sequences may be initiated with $A_{15}$ in either state.
2. Address bit $A_{15}$, $A_{16}$, $A_{17}$, and $A_{18}$ = X = Don't Care for all address commands except for Program Address (PA) and Sector Address (SA).
3. RA = Address of the memory location to be read.
   PA = Address of the memory location to be programmed. Addresses are latched on the falling edge of the $\overline{WE}$ pulse.
   SA = Address of the sector to be erased. The combination of $A_{18}$, $A_{17}$, $A_{16}$ will uniquely select any sector.
4. RD = Data read from location RA during read operation.
   PD = Data to be programmed at location PA. Data is latched on the rising edge of $\overline{WE}$.

## Autoselect Command

The operation is initiated by writing the autoselect command sequence into the command register. Following the command write, a read cycle from address XX01H returns the device code A4H (see Table 4). The device codes will exhibit odd parity with the MSB (I/O7) defined as the parity bit.

Scanning the sector addresses (A16, A17, A18) while (A6, A1, A0) - (0, 1, 0) will produce a logical "1" at device output I/O0 for a protected sector.

To terminate the operation, it is necessary to write the read/reset command sequence into the register.

## Byte Programing

The device is programmed on a byte-by-byte basis. Programming is a four bus cycle operation. There are two "unlock" write cycles. These are followed by the program set-up command and data write cycles. Addresses are latched on the falling edge of $\overline{CS}$ or $\overline{WE}$, whichever happens later and the data is latched on the rising edge of $\overline{CS}$ or $\overline{WE}$, whichever happens first. The rising edge of $\overline{CS}$ or $\overline{WE}$ begins programming. Upon executing the Program Algorithm command sequence the system is *not* required to provide further controls or timings. The device will automatically provide internally generated program pulses and verify the programmed cell margin.

The automatic programming operation is completed when the data on D7 is equivalent to data written to this bit (see Write Operation Status section) at which time the device returns to the read mode and addresses are no longer latched. Therefore, the device requires that a valid address to the device be supplied by the system at this particular time. $\overline{\text{Data}}$ Polling must be performed at the memory location which is being programmed.

Programming is allowed in any sequence and across sector boundaries. Beware that a data "0" cannot be programmed back to a "1". Attempting to do so will hang up the device, or result in an apparent success according to the data polling algorithm. However, a read from reset/read mode will show that the data is still "0". Only erase operations can convert "0"s to "1"s.

Figure 2 illustrates the Programming Algorithm.

## Chip Erase

Chip erase is a six bus cycle operation. There are two “unlock” write cycles. These are followed by writing the “set-up” command. Two more “unlock” write cycles are then followed by the chip erase command.

Chip erase does *not* require the user to program the device prior to erase. Upon executing the erase algorithm sequence the device automatically will program and verify 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.

The automatic erase begins on the rising edge of the last $\overline{WE}$ pulse in the command sequence and terminates when the data in D7 is “1” (see Write Operation Status section) at which time the device returns to read the mode. See figure 3, Erase algorithm.

## Sector Erase

Sector erase is a six bus cycle operation. There are two “unlock” write cycles. These are followed by writing the “set-up” command. Two more “unlock” write cycles are then followed by the sector erase command. The sector address (any address location within the desired sector) is latched on the falling edge of $\overline{WE}$, while the command (data) is latched on the rising edge of $\overline{WE}$. A time-out of 80µs from the rising edge of the last sector erase command will initiate the sector erase command(s).

*Note: Do not attempt to write an invalid command sequence during the sector erase operation. Otherwise, it will terminate the sector erase operation and the device will reset back into the read mode.*

Multiple sectors may be erased concurrently by writing the six bus cycle operations as described above. This sequence is followed with writes of the Sector Erase command to addresses in other sectors desired to be concurrently erased. A time-out of 80µs from the rising edge of the $\overline{WE}$ pulse for the last sector erase command will initiate the sector erase. If another sector erase command is written within the 80µs time-out window the timer is reset. Any command other than Sector Erase within the time-out window will reset the device to the read mode, ignoring the previous command string. Loading the sector erase buffer may be done in any sequence and with any number of sectors (1 to 8).

Sector erase does not require the user to program the device prior to erase. The device automatically programs all memory locations in the sector(s) to be erased prior to electrical erase. When erasing a sector or sectors the remaining unselected sectors are not affected. The system is *not* required to provide any controls or timings during these operations.

### TABLE 4 - SECTOR PROTECTION VERIFY AUTOSELECT CODES

| Type | A18 | A17 | A16 | A9 | A6 | A1 | A0 | Code (HEX) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4M5 Device Code | X | X | X | $V_{ID}$(1) | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | A4H | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Sector Group Protection | Sector Group Addr. | | $V_{ID}$(1) | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | 01H* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |

* Outputs 01H at protected sector addresses.

1. $V_{ID}$ = 12V

## DATA PROTECTION

The device is designed to offer protection against accidental erasure or programming caused by spurious system level signals that may exist during power transitions. During power up the device automatically resets the internal state machine in the read mode. Also, with its control register architecture, alteration of the memory content only occurs after successful completion of specific multi-bus cycle command sequences.

The device also incorporates several features described below to prevent inadvertent write cycles resulting from Vcc power-up and power-down transitions or system noise.

### Low Vcc Write Inhibit

To avoid initiation of a write cycle during Vcc power-up and power-down, a write cycle is locked out for Vcc less than 3.2V (typically 3.7V). If Vcc < $V_{LKO}$, the command register is disabled and all internal program/erase circuits are disabled. Under this condition the device will reset to read mode. Subsequent writes will be ignored until the Vcc level is greater than $V_{LKO}$. It is the users responsibility to ensure that the control pins are logically correct to prevent unintentional writes when Vcc is above 3.2V.

### Write Pulse Glitch Protection

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

### Logical Inhibit

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

### Power-Up Write Inhibit

Power-up of the device with $\overline{WE}$ = $\overline{CS}$ = $V_{IL}$ and $\overline{OE}$ = $V_{IH}$ will not accept commands on the rising edge of $\overline{WE}$. The internal state machine is automatically reset to the read mode on power-up.

### Sector Protection

The device features hardware sector protection. This feature will disable both program and erase operations in any number of sectors (0 through 8). The sector protect feature is enabled using programming equipment at the user's site. The device is shipped with all sectors unprotected.

It is possible to determine if a sector is protected in the system by writing an Autoselect command. Performing a read operation at the address location XX02H, where the higher order addresses (A16, A17, and A18) are the sector address, will produce a logical "1" at I/O0 for protected sector. See Table 3 for Autoselect codes.

It is also possible to use the programming equipment to force $V_{ID}$ (11.5V to 12.5V) on address pin A9. Two identifier bytes may then be sequenced from the device outputs by toggling address A0 from $V_{IL}$ to $V_{IH}$. All addresses are don't cares except A0, A1, and A6 (See Table 4).

### Sector Unprotect

The device also features a sector unprotect mode so that a protected sector may be unprotected to incorportate any changes in the code. The sector unprotect is enabled using programming equipment at the user's site.

It is possible to determine if the sector is unprotected in the system by writing the autoselect command. Performing a read operation at address location XXX2H, where the higher order addresses (A16, A17, and A18) define a particular sector address, will produce 00H at data outputs (I/O0-7) for an unprotected sector. Refer to previous Sector Protection section for additional information.

## WRITE OPERATION STATUS

### D7
### Data Polling

The device features $\overline{\text{Data}}$ Polling as a method to indicate to the host that the internal algorithms are in progress or completed. During the program algorithm, an attempt to read the device will produce compliment data of the data last written to D7. When complete D7 will read true data. During the erase algorithm, an attempt to read the device will produce a "0" at the D7 output. Upon completion of the erase algorithm an attempt to read the device will produce a "1" at the D7 output.

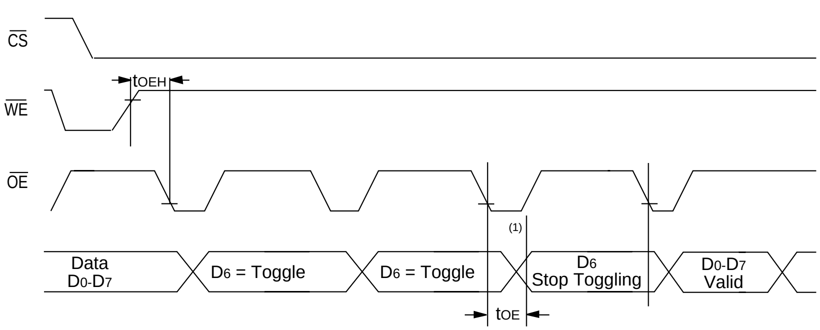
For chip erase, the Data Polling is valid after the rising edge of the sixth $\overline{\text{WE}}$ pulse in the six write pule sequence. For sector erase, the $\overline{\text{Data}}$ Polling is valid after the last rising edge of the sector erase $\overline{\text{WE}}$ pulse. $\overline{\text{Data}}$ Polling must be performed at a sector address within any of the sectors being erased and not a protected sector. Otherwise, the status may not be valid. Once the algorithm operation is close to being completed, data pins (D7) change asynchronously while the ouput enable ($\overline{\text{OE}}$) is asserted low. This means that the device is driving status information on D7 at one instance of time and then that byte's valid data at the next instant of time. Depending on when the system samples the D7 output, it may read the status or valid data. Even if the device has completed internal algorithm operation and D7 has a valid data, the data ouputs on D0 - D6 may be still invalid. The valid data on D0 - D7 will be read on the successive read attempts. The $\overline{\text{Data}}$ Polling feature is only active during the programming algorithm, erase algoithm, or sector erase time-out. See Figure 5, $\overline{\text{Data}}$ Polling algorithm.

**TABLE 4 - WRITE OPERATION STATUS**

| | Status | $D_7$ | $D_6$ | $D_5$ | $D_3$ | $D_2$-$D_0$ |
|---|---|---|---|---|---|---|
| **In Progress** | Byte Program in Program Algorithm | $\overline{D_7}$ | Toggle | 0 | 0 | Reserved for future use |
| | Erase Algorithm | 0 | Toggle | 0 | 1 | |
| **Exceeded Time Limits** | Byte Program in Program Algorithm | $\overline{D_7}$ | Toggle | 1 | 0 | Reserved for future use |
| | Program/Erase in Erase Algorithm | 0 | Toggle | 1 | 1 | |

Note: **Performing successive read operations from any address will cause D6 to toggle.**

**FIG. 1**
**AC WAVEFORMS FOR TOGGLE BIT DURING EMBEDDED ALGORITHM OPERATIONS**




**NOTES:**
1. D6 stops toggling (the device has completed the Embedded operation).

## D6
## Toggle Bit

The device also features the “Toggle Bit” as a method to indicate to the host system that algorithms are in progress or completed.

During a program or erase algorithm cycle, successive attempts to read data from the device will result in D6 toggling between one and zero. Once the program or erase algorithm cycle is completed, D6 will stop toggling and valid data will be read on successive attempts. During programming, the Toggle Bit is valid after the rising edge of the fourth $\overline{WE}$ pulse in the four write pulse sequence. For chip erase the Toggle Bit is valid after the rising edge of the sixth $\overline{WE}$ pulse in the six write pulse sequence. For Sector erase, the Toggle Bit is valid after the last rising edge of the sector erase $\overline{WE}$ pulse. The Toggle Bit is active during the sector time out. See Figure 1, Toggle Bit algorithm.

## D5
## Exceeded Timing Limits

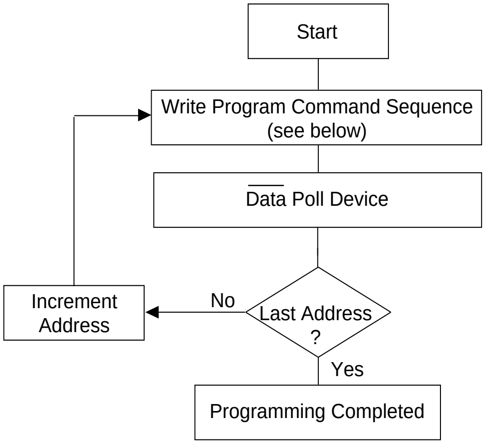
D5 will indicate if the program or erase time has exceeded the specified limits. Under these conditions D5 will produce a “1”, which indicates that the program or erase cycle was not successfully completed. $\overline{\text{Data}}$ Polling is the only operating function of the device under this condition. The $\overline{CS}$ circuit will partially power down the device under these conditions to approximately 2 mA per chip. The $\overline{OE}$ and $\overline{WE}$ pins will control the output disable functions as shown in Table 1. To reset the device, write the reset command sequence to the device. This allows the system to continue to use the other active sectors in the device.

## D3
## Sector Erase Timer

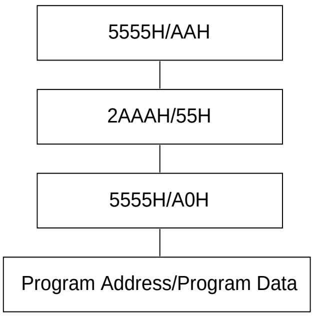
After the completion of the initial sector erase command sequence the sector erase time-out will begin. D3 will remain low until the time-out is complete. $\overline{\text{Data}}$ Polling and Toggle Bit are valid after the initial sector erase command sequence.

If $\overline{\text{Data}}$ Polling or the Toggle Bit indicates the device has been written with a valid erase command, D3 may be used to determine if the sector erase timer window is still open. If D3 is high (“1”) the internally controlled erase cycle has begun; attempts to write subsequent commands to the device will be ignored until the erase operation is completed as indicated by $\overline{\text{Data}}$ Polling or Toggle Bit. If D3 is low (“0”), the device will accept additional sector erase commands. To ensure the command has been accepted, the software should check the status of D3 prior to and following each subsequent sector erase command. If D3 were high on the second status check, the command may not have been accepted. See Table 5.

## FIG. 2
## PROGRAMMING ALGORITHM




**Program Command Sequence (Address/Command):**

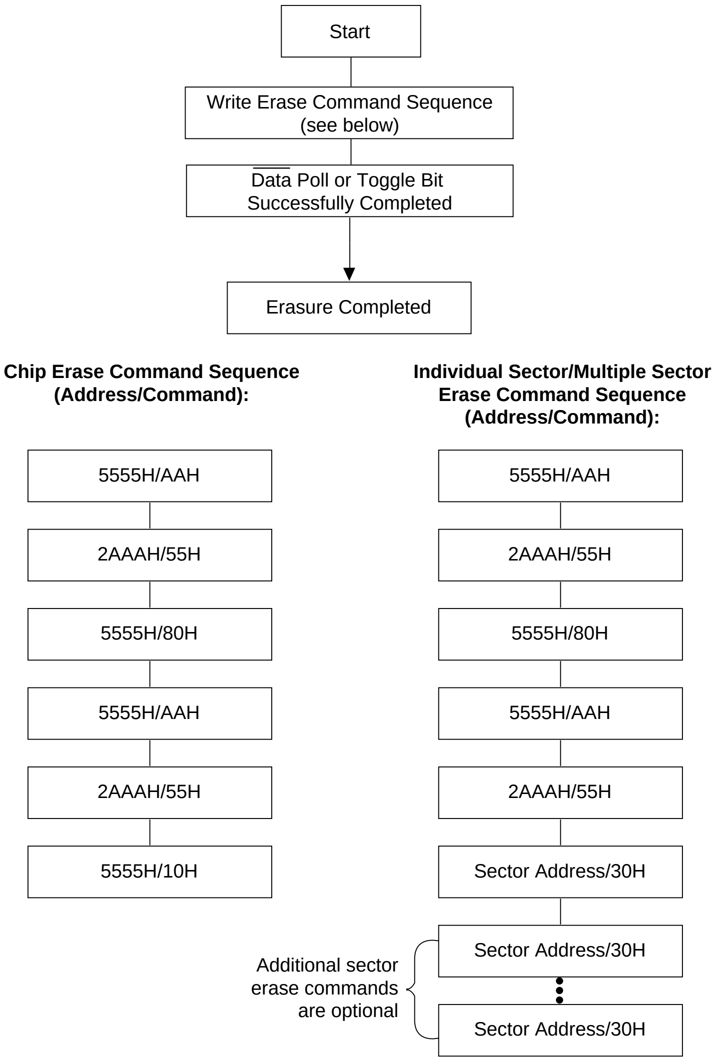



| Bus Operations | Command Sequence | Comments |
|---|---|---|
| Standby * | | |
| Write | Program | Valid Address/Data Sequence |
| Read | | $\overline{\text{Data}}$ Polling to Verify Programming |
| Standby * | | Compare Data Output to Data Expected |

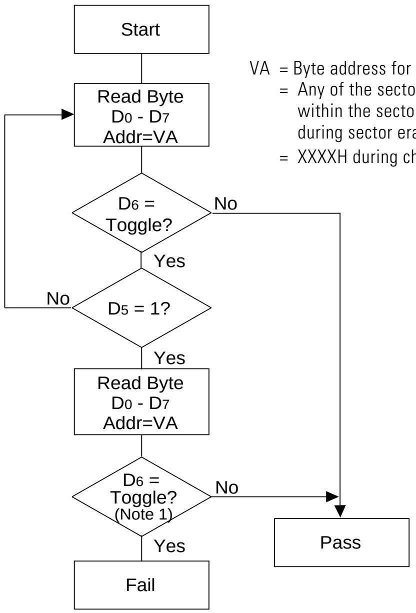
* Device is either powered-down, erase inhibit or program inhibit.

## FIG. 3
### ERASE ALGORITHM

| Bus Operations | Command Sequence | Comments |
|---|---|---|
| Standby | | |
| Write | Erase | |
| Read | | $\overline{\text{Data}}$ Polling to Verify Erasure |
| Standby | | Compare Output to FFH |



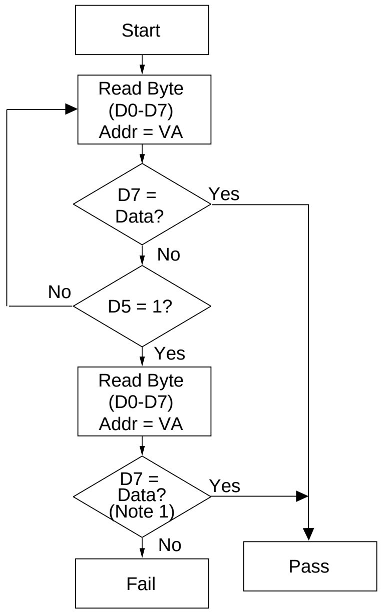
## FIG. 4
### TOGGLE BIT ALGORITHM




VA = Byte address for programming
= Any of the sector addresses within the sector being erased during sector erase operation
= XXXXH during chip erase

1. $D_6$ is rechecked even if $D_5 = 1$ because $D_6$ may stop toggling at the same time as $D_5$ changes to 1.

## FIG. 5
### DATA POLLING ALGORITHM




1. $D_7$ is rechecked even if $D_5 = 1$ because $D_6$ may change simultaneously with $D_5$.