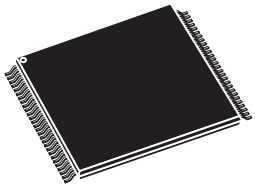
# M29W128FH
# M29W128FL

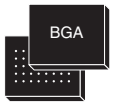
## 128 Mbit (8Mb x 16 or 16Mb x 8, Page, Uniform Block) 3V Supply Flash Memory

## Feature summary

- Supply voltage
  - $V_{CC}$ = 2.7 to 3.6V for Program, Erase and Read
  - $V_{PP}$ =12V for Fast Program (optional)
- Asynchronous Random/Page Read
  - Page Width: 8 Words/16 Bytes
  - Page Access: 25, 30ns
  - Random Access: 60, 70ns
- Programming time
  - 10µs per Byte/Word (typical)
  - 4 Words / 8 Bytes Program
  - 32-Word (64-Bytes) Write Buffer
- 64 KByte (32 KWord) Uniform Blocks
- Program/ Erase Suspend and Resume Modes
  - Read from any Block during Program Suspend
  - Read and Program another Block during Erase Suspend
- Unlock Bypass Program
  - Faster Production/Batch Programming
- Common Flash Interface
  - 64 bit Security Code
- 100,000 Program/Erase cycles per block
- Low power consumption
  - Standby and Automatic Standby
- Hardware Block Protection
  - $V_{PP}/\overline{WP}$ pin for fast program and write protect of the highest (M29W128FH) or lowest block (M29W128FL)
- Extended Memory Block:
  Extra block used as security block or to store additional information



TSOP56 (N)
14 x 20mm




TBGA64 (ZA)
10 x 13mm

- Electronic Signature
  - Manufacturer Code: 0020h
  - Device Code:
    M29W128FH: 227Eh + 2212h + 228Ah
    M29W128FL: 227Eh + 2212h + 228Bh
- ECOPACK® packages

# Contents

# List of tables

# List of figures

# 1 Summary description

The M29W128FH and M29W128FL are 128 Mbit (16Mb x8 or 8Mb x16) non-volatile memories that can be read, erased and reprogrammed. These operations can be performed using a single low voltage (2.7 to 3.6V) supply. At Power-up the memories default to Read mode. The M29W128FH and M29W128FL are divided into 256 thirty-two KWord (sixty-four KByte) uniform blocks.

Program and Erase commands are written to the Command Interface of the memory. An on-chip Program/Erase Controller simplifies the process of programming or erasing the memory by taking care of all of the special operations that are required to update the memory contents. The end of a program or erase operation can be detected and any error conditions identified. The command set required to control the memory is consistent with JEDEC standards.

The Chip Enable, Output Enable and Write Enable signals control the bus operations of the memory. They allow simple connection to most microprocessors, often without additional logic.

The devices support Asynchronous Random Read and Page Read from all blocks of the memory array.

The M29W128FH and M29W128FL have an extra 128 Word (256 Byte) Extended Memory Block that can be accessed using a dedicated command. The Extended Memory Block can be protected and so is useful for storing security information. However the protection is irreversible, once protected the protection cannot be undone.

Each block can be erased independently, so it is possible to preserve valid data while old data is erased.

The devices feature two different levels of hardware block protection to avoid unwanted program or erase (modify):

- The $V_{PP}/\overline{WP}$ pin protects the highest block on the M29W128FH and the lowest block on the M29W128FL.
- The $\overline{RP}$ pin temporarily unprotects all the blocks previously protected using a High Voltage Block Protection technique (see *Appendix D: High Voltage Block Protection*).

The memories are offered in TSOP56 (14 x 20mm) and TBGA64 (10 x 13mm, 1mm pitch) packages.

In order to meet environmental requirements, Numonyx offers the M29W128FH and the M29W128FL in ECOPACK® packages. ECOPACK packages are Lead-free. The category of second Level Interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

The memories are supplied with all the bits erased (set to '1').

The M29W128FH and the M29W128FL will be referred to as M29W128F throughout the document.

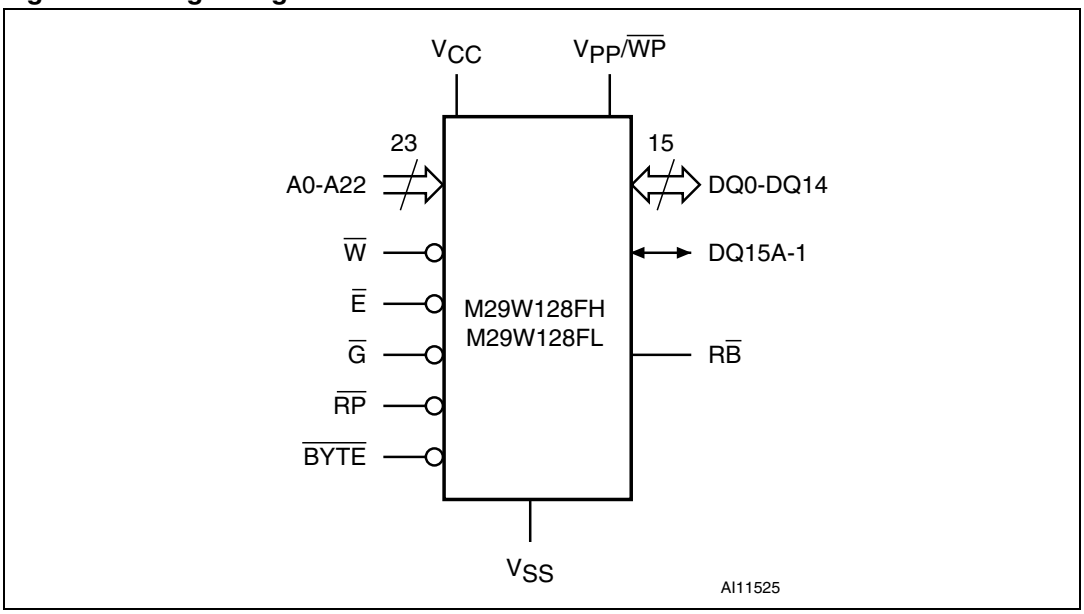
**Table 1. Signal names**

| | |
|---|---|
| A0-A22 | Address Inputs |

**Table 1. Signal names**

| | |
|---|---|
| DQ0-DQ7 | Data Inputs/Outputs |
| DQ8-DQ14 | Data Inputs/Outputs |
| DQ15A-1 | Data Input/Output or Address Input |
| $\overline{E}$ | Chip Enable |
| $\overline{G}$ | Output Enable |
| $\overline{W}$ | Write Enable |
| $\overline{RP}$ | Reset/Block Temporary Unprotect |
| $R\overline{B}$ | Ready/Busy Output |
| $\overline{BYTE}$ | Byte/Word Organization Select |
| $V_{CC}$ | Supply voltage |
| $V_{PP}/\overline{WP}$ | $V_{PP}$/Write Protect |
| $V_{SS}$ | Ground |
| NC | Not Connected Internally |

**Figure 1. Logic diagram**




1. Also see *Appendix A* and *Table 28* for a full listing of the Block Addresses.

**Figure 2. TSOP connections**

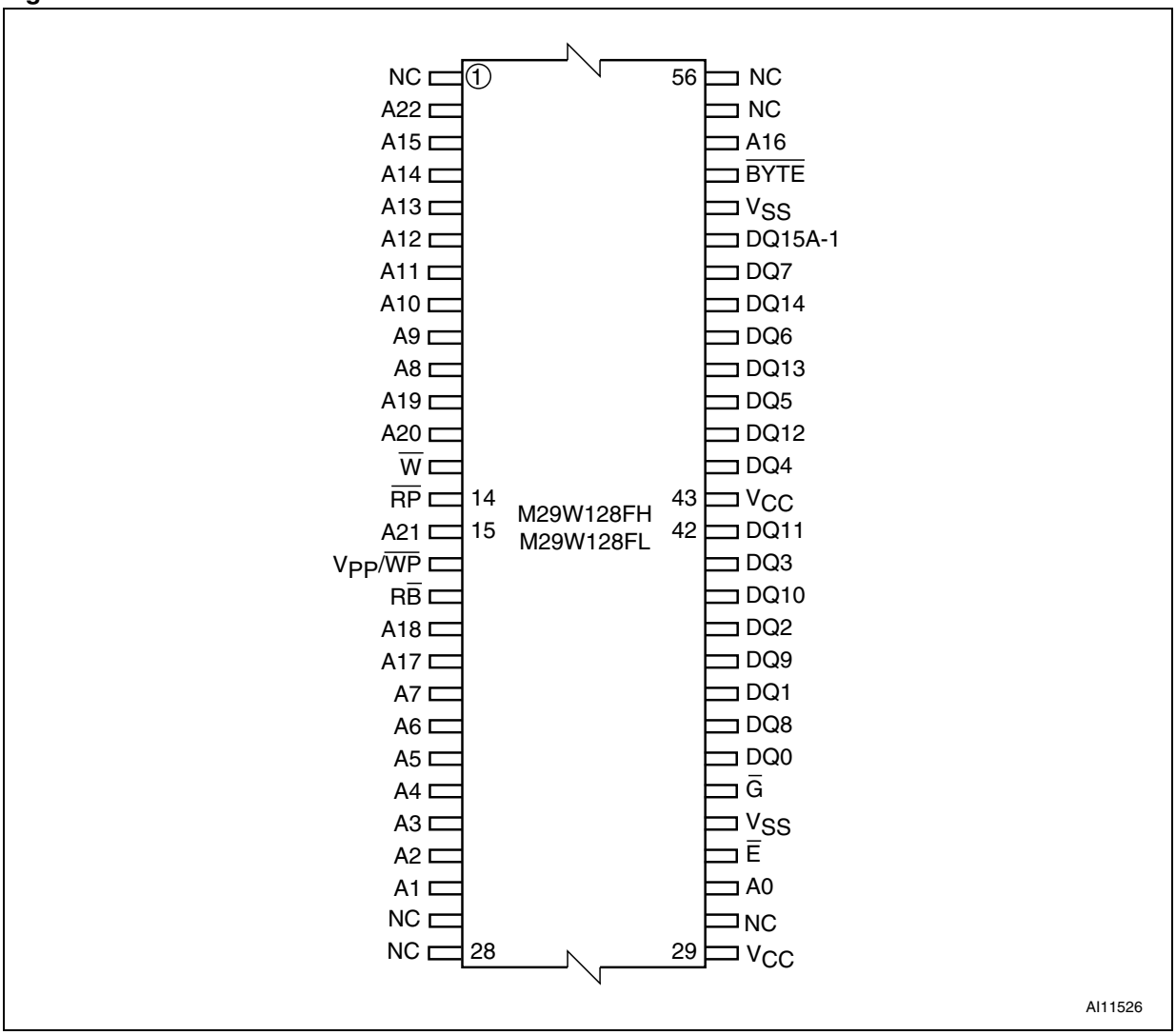


**Figure 3. TBGA connections (top view through package)**

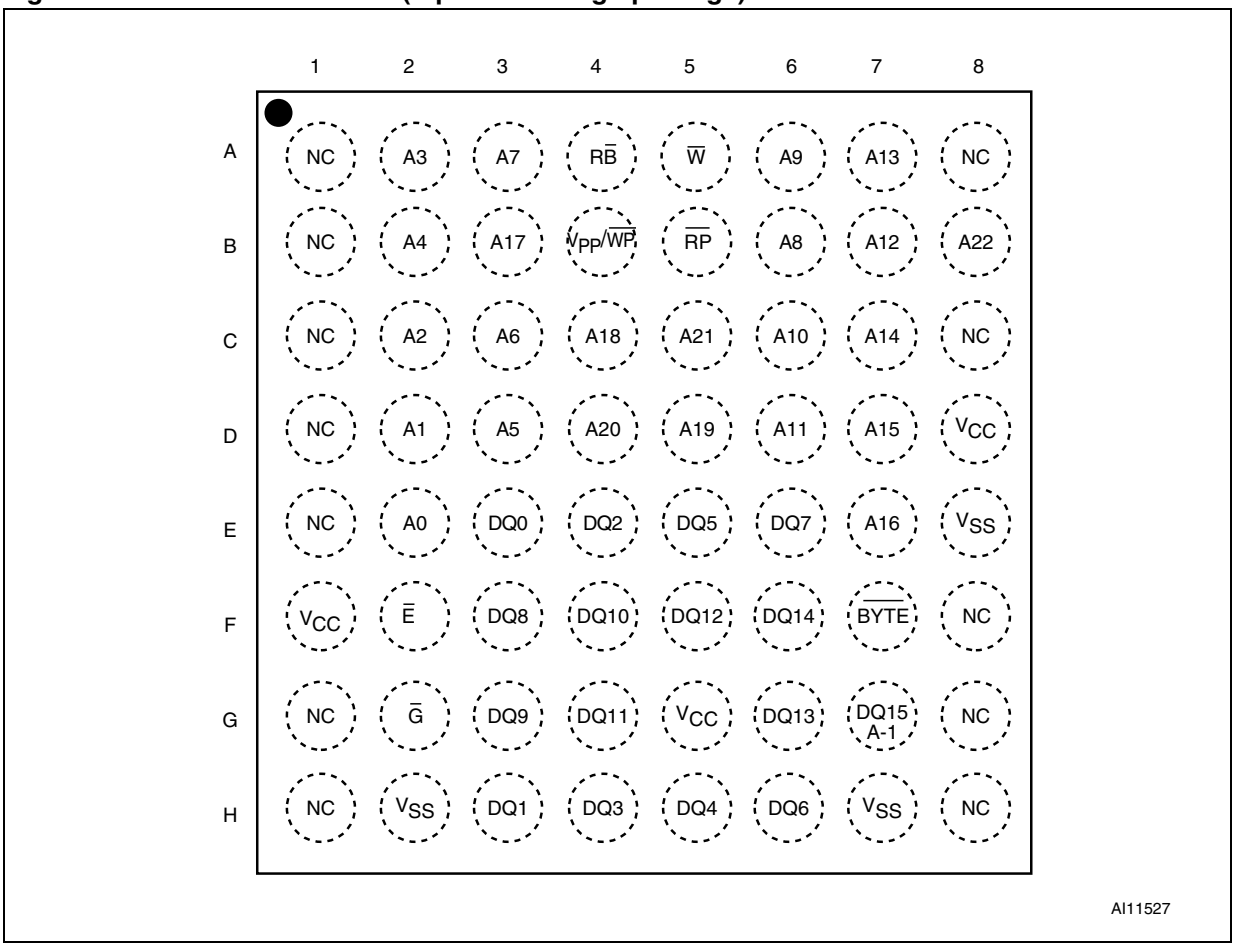
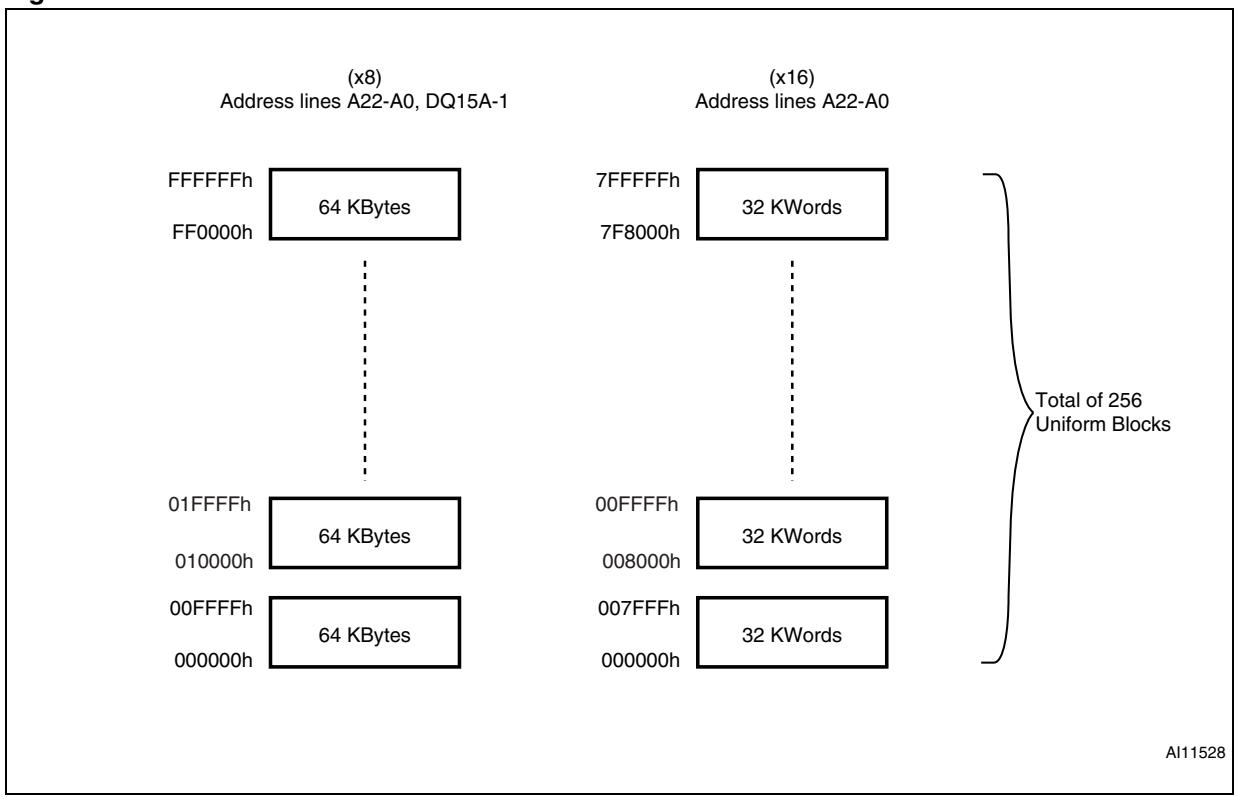


**Figure 4. Block addresses**



# 2 Signal descriptions

See *Figure 1: Logic diagram*, and *Table 1: Signal names*, for a brief overview of the signals connected to this device.

## 2.1 Address Inputs (A0-A22)

The Address Inputs select the cells in the memory array to access during Bus Read operations. During Bus Write operations they control the commands sent to the Command Interface of the Program/Erase Controller.

## 2.2 Data Inputs/Outputs (DQ0-DQ7)

The Data I/O outputs the data stored at the selected address during a Bus Read operation. During Bus Write operations they represent the commands sent to the Command Interface of the internal state machine.

## 2.3 Data Inputs/Outputs (DQ8-DQ14)

The Data I/O outputs the data stored at the selected address during a Bus Read operation when $\overline{BYTE}$ is High, $V_{IH}$. When $\overline{BYTE}$ is Low, $V_{IL}$, these pins are not used and are high impedance. During Bus Write operations the Command Register does not use these bits. When reading the Status Register these bits should be ignored.

## 2.4 Data Input/Output or Address Input (DQ15A–1)

When the device is in x16 Bus mode, this pin behaves as a Data Input/Output pin (as DQ8-DQ14). When the device is in x8 Bus mode, this pin behaves as an address pin; DQ15A–1 Low will select the LSB of the addressed Word, DQ15A–1 High will select the MSB. Throughout the text consider references to the Data Input/Output to include this pin when the device operates in x16 bus mode and references to the Address Inputs to include this pin when the device operates in x8 bus mode except when stated explicitly otherwise.

## 2.5 Chip Enable ($\overline{E}$)

The Chip Enable pin, $\overline{E}$, activates the memory, allowing Bus Read and Bus Write operations to be performed. When Chip Enable is High, $V_{IH}$, all other pins are ignored.

## 2.6 Output Enable ($\overline{G}$)

The Output Enable pin, $\overline{G}$, controls the Bus Read operation of the memory.

## 2.7 Write Enable ($\overline{W}$)

The Write Enable pin, $\overline{W}$, controls the Bus Write operation of the memory's Command Interface.

## 2.8 $V_{PP}$/Write Protect ($V_{PP}/\overline{WP}$)

The $V_{PP}$/Write Protect pin provides two functions. The $V_{PP}$ function allows the memory to use an external high voltage power supply to reduce the time required for Program operations. This is achieved by bypassing the unlock cycles and/or using the multiple Word (2 or 4 at-a-time) or multiple Byte Program (2, 4 or 8 at-a-time) commands.

The Write Protect function provides a hardware method of protecting the highest or lowest block. When $V_{PP}$/Write Protect is Low, $V_{IL}$, the highest or lowest block is protected; Program and Erase operations on this block are ignored while $V_{PP}$/Write Protect is Low, even when $\overline{RP}$ is at $V_{ID}$.

When $V_{PP}$/Write Protect is High, $V_{IH}$, the memory reverts to the previous protection status of the highest or lowest block. Program and Erase operations can now modify the data in this block unless the block is protected using Block Protection.

Applying $V_{PPH}$ to the $V_{PP}/\overline{WP}$ pin will temporarily unprotect any block previously protected (including the highest or lowest block) using a High Voltage Block Protection technique (In-System or Programmer technique). See *Table 8: Hardware Protection* for details.

When $V_{PP}$/Write Protect is raised to $V_{PP}$ the memory automatically enters the Unlock Bypass mode. When $V_{PP}$/Write Protect returns to $V_{IH}$ or $V_{IL}$ normal operation resumes. During Unlock Bypass Program operations the memory draws $I_{PP}$ from the pin to supply the programming circuits. See the description of the Unlock Bypass command in the Command Interface section. The transitions from $V_{IH}$ to $V_{PP}$ and from $V_{PP}$ to $V_{IH}$ must be slower than $t_{VHVPP}$ (see *Figure 15: Accelerated Program Timing waveforms*).

Never raise $V_{PP}$/Write Protect to $V_{PP}$ from any mode except Read mode, otherwise the memory may be left in an indeterminate state.

The $V_{PP}$/Write Protect pin must not be left floating or unconnected or the device may become unreliable. A 0.1µF capacitor should be connected between the $V_{PP}$/Write Protect pin and the $V_{SS}$ Ground pin to decouple the current surges from the power supply. The PCB track widths must be sufficient to carry the currents required during Unlock Bypass Program, $I_{PP}$.

## 2.9 Reset/Block Temporary Unprotect ($\overline{RP}$)

The Reset/Block Temporary Unprotect pin can be used to apply a Hardware Reset to the memory or to temporarily unprotect all the blocks previously protected using a High Voltage Block Protection technique (In-System or Programmer technique). Note that if $V_{PP}/\overline{WP}$ is at $V_{IL}$, then the highest or lowest block will remain protected even if $\overline{RP}$ is at $V_{ID}$.

A Hardware Reset is achieved by holding Reset/Block Temporary Unprotect Low, $V_{IL}$, for at least $t_{PLPX}$. After Reset/Block Temporary Unprotect goes High, $V_{IH}$, the memory will be ready for Bus Read and Bus Write operations after $t_{PHEL}$ or $t_{RHEL}$, whichever occurs last. See *Section 2.10: Ready/Busy Output (RB)*, *Table 24: Reset/Block Temporary Unprotect AC characteristics* and *Figure 13* and *Figure 14* for more details.

Holding $\overline{RP}$ at $V_{ID}$ will temporarily unprotect all the blocks previously protected using a High Voltage Block Protection technique. Program and erase operations on all blocks will be possible. The transition from $V_{IH}$ to $V_{ID}$ must be slower than $t_{PHPHH}$.

## 2.10 Ready/Busy Output ($\overline{RB}$)

The Ready/Busy pin is an open-drain output that can be used to identify when the device is performing a Program or erase operation. During Program or erase operations Ready/Busy is Low, $V_{OL}$. Ready/Busy is high-impedance during Read mode, Auto Select mode and Erase Suspend mode.

After a Hardware Reset, Bus Read and Bus Write operations cannot begin until Ready/Busy becomes high-impedance. See *Table 24: Reset/Block Temporary Unprotect AC characteristics* and *Figure 13* and *Figure 14*.

The use of an open-drain output allows the Ready/Busy pins from several memories to be connected to a single pull-up resistor. A Low will then indicate that one, or more, of the memories is busy.

## 2.11 Byte/Word Organization Select ($\overline{BYTE}$)

It is used to switch between the x8 and x16 Bus modes of the memory. When Byte/Word Organization Select is Low, $V_{IL}$, the memory is in x8 mode, when it is High, $V_{IH}$, the memory is in x16 mode.

## 2.12 $V_{CC}$ supply voltage (2.7V to 3.6V)

$V_{CC}$ provides the power supply for all operations (Read, Program and Erase).

The Command Interface is disabled when the $V_{CC}$ Supply Voltage is less than the Lockout Voltage, $V_{LKO}$. This prevents Bus Write operations from accidentally damaging the data during power up, power down and power surges. If the Program/Erase Controller is programming or erasing during this time then the operation aborts and the memory contents being altered will be invalid.

A 0.1µF capacitor should be connected between the $V_{CC}$ Supply Voltage pin and the $V_{SS}$ Ground pin to decouple the current surges from the power supply. The PCB track widths must be sufficient to carry the currents required during program and erase operations, $I_{cc2}$.

## 2.13 $V_{SS}$ ground

$V_{SS}$ is the reference for all voltage measurements. The device features two $V_{SS}$ pins both of which must be connected to the system ground.

# 3 Bus operations

There are five standard bus operations that control the device. These are Bus Read (Random and Page modes), Bus Write, Output Disable, Standby and Automatic Standby.

See *Table 2* and *Table 5*, Bus Operations, for a summary. Typically glitches of less than 5ns on Chip Enable, Write Enable, and Reset/Block Temporary Unprotect pins are ignored by the memory and do not affect bus operations.

## 3.1 Bus Read

Bus Read operations read from the memory cells, or specific registers in the Command Interface. To speed up the read operation the memory array can be read in Page mode where data is internally read and stored in a page buffer. The Page has a size of 8 Words (or 16 Bytes) and is addressed by the address inputs A2-A0 in x16 mode and A2-DQ15A–1 in Byte mode.

A valid Bus Read operation involves setting the desired address on the Address Inputs, applying a Low signal, $V_{IL}$, to Chip Enable and Output Enable and keeping Write Enable High, $V_{IH}$. The Data Inputs/Outputs will output the value, see *Figure 9: Random Read AC waveforms*, *Figure 10: Page Read AC waveforms (Word mode)*, and *Table 21: Read AC characteristics*, for details of when the output becomes valid.

## 3.2 Bus Write

Bus Write operations write to the Command Interface. A valid Bus Write operation begins by setting the desired address on the Address Inputs. The Address Inputs are latched by the Command Interface on the falling edge of Chip Enable or Write Enable, whichever occurs last. The Data Inputs/Outputs are latched by the Command Interface on the rising edge of Chip Enable or Write Enable, whichever occurs first. Output Enable must remain High, $V_{IH}$, during the whole Bus Write operation. See *Figure 11* and *Figure 12*, Write AC Waveforms, and *Table 22* and *Table 23*, Write AC Characteristics, for details of the timing requirements.

## 3.3 Output Disable

The Data Inputs/Outputs are in the high impedance state when Output Enable is High, $V_{IH}$.

## 3.4 Standby

When Chip Enable is High, $V_{IH}$, the memory enters Standby mode and the Data Inputs/Outputs pins are placed in the high-impedance state. To reduce the Supply Current to the Standby Supply Current, $I_{CC2}$, Chip Enable should be held within $V_{CC}$ ± 0.3V. For the Standby current level see *Table 20: DC characteristics*.

During program or erase operations the memory will continue to use the Program/Erase Supply Current, $I_{CC3}$, for Program or Erase operations until the operation completes.

## 3.5 Automatic Standby

If CMOS levels ($V_{CC}$ ± 0.3V) are used to drive the bus and the bus is inactive for $t_{AVQV}$+ 30ns or more the memory enters Automatic Standby where the internal Supply Current is reduced to the Standby Supply Current, $I_{CC2}$. The Data Inputs/Outputs will still output data if a Bus Read operation is in progress.

## 3.6 Special Bus operations

Additional bus operations can be performed to read the Electronic Signature, verify the Protection Status of the Extended Memory Block, and apply and remove Block Protection. These bus operations are intended for use by programming equipment and are not usually used in applications. They require $V_{ID}$ to be applied to some pins.

### 3.6.1 Read Electronic Signature

The memory has two codes, the Manufacturer code and the Device code used to identify the memory. These codes can accessed by performing read operations with control signals and addresses set as shown in *Table 3* and *Table 5*.

These codes can also be accessed by issuing an Auto Select command (see *Section 5.1.2: Auto Select command*).

### 3.6.2 Verify Extended Memory Block Protection Indicator

The Extended Memory Block is either Factory Locked or Customer Lockable.

The Protection Status of the Extended Memory Block (Factory Locked or Customer Lockable) can be accessed by reading the Extended Memory Block Protection Indicator. This is performed by applying the signals as shown in *Table 4* and *Table 7*. The Protection Status of the Extended Memory Block is then output on bit DQ7 of the Data Input/Outputs. (see *Table 2* and *Table 5*, Bus Operations).

The Protection Status of the Extended Memory Block can also be accessed by issuing an Auto Select command (see *Section 5.1.2: Auto Select command*).

### 3.6.3 Verify Block Protection Status

The Protection Status of a Block can be directly accessed by performing a read operation with control signals and addresses set as shown in *Table 4* and *Table 7*.

If the Block is protected, then 01h (in x8 mode) is output on Data Input/Outputs DQ0-DQ7, otherwise 00h is output.

### 3.6.4 Hardware Block Protect

The $V_{PP}/\overline{WP}$ pin can be used to protect the highest or lowest block. When $V_{PP}/\overline{WP}$ is at $V_{IL}$ the highest or lowest block is protected and remains protected regardless of the Block Protection Status or the Reset/Block Temporary Unprotect pin state.

### 3.6.5 Temporary Unprotect of high voltage Protected Blocks

The $\overline{RP}$ pin can be used to temporarily unprotect all the blocks previously protected using the In-System or the Programmer protection technique (High Voltage techniques).

Refer to *Section 2.9: Reset/Block Temporary Unprotect (RP)*.

**Table 2. Bus operations, 8-bit mode**

| Operation[1] | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | $\overline{RP}$ | $V_{PP}/\overline{WP}$ | Address Inputs | Data Inputs/Outputs | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | A22-A0, DQ15A-1 | DQ14-DQ8 | DQ7-DQ0 |
| Bus Read | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | Cell Address | Hi-Z | Data Output |
| Bus Write | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | Command Address | Hi-Z | Data Input |
| Output Disable | X | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | X | Hi-Z | Hi-Z |
| Standby | $V_{IH}$ | X | X | $V_{IH}$ | $V_{IH}$ | X | Hi-Z | Hi-Z |

1. X = $V_{IL}$ or $V_{IH}$.

**Table 3. Read Electronic Signature, 8-bit mode**

| Read Cycle[1] | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | Address Inputs | | | | | | | | | | Data Inputs/Outputs | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | A22-A10 | A9 | A8-A7 | A6 | A5-A4 | A3 | A2 | A1 | A0 | DQ15A-1 | DQ14-DQ8 | DQ7-DQ0 |
| Manufacturer Code | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | X | $V_{ID}$ | X | $V_{IL}$ | X | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | X | Hi-Z | 20h |
| Device Code (Cycle 1) | | | | | | | | | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | X | Hi-Z | 7Eh (Both Devices) |
| Device Code (Cycle 2) | | | | | | | | | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | $V_{IL}$ | X | Hi-Z | 12h (Both Devices) |
| Device Code (Cycle 3) | | | | | | | | | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | X | Hi-Z | 8Ah (M29W128FH)<br>8Bh (M29W128FL) |

1. X = $V_{IL}$ or $V_{IH}$.

**Table 4. Block Protection, 8-bit mode**

| Operation (1) | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | $\overline{RP}$ | $V_{PP}/\overline{WP}$ | Address Inputs | | | | | | | | | Data Inputs/Outputs | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | A22-A12 | A11-A10 | A9 | A8-A7 | A6 | A5-A4 | A3-A2 | A1 | A0 | DQ15A-1 | DQ14-DQ8 | DQ7-DQ0 |
| Verify Extended Memory Block Protection Indicator (bit DQ7) | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | BA | X | $V_{ID}$ | X | $V_{IL}$ | X | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | X | Hi-Z | M29W128FH 88h (factory locked) 08h (customer lockable) |
| | | | | | | | | | | | | | | | | | M29W128FL 98h (factory locked) 18h (customer lockable) |
| Verify Block Protection Status | | | | | | | | | | | | | | $V_{IL}$ | | | 01h (protected) 00h (unprotected) |
| Temporary Block Unprotect (2) | X | X | X | $V_{ID}$ | X | Valid | | | | | | | | | | | Data Input |

1. X = $V_{IL}$ or $V_{IH}$. BA any Address in the Block.
2. The $\overline{RP}$ pin unprotects all the blocks that have been previously protected using a High Voltage protection Technique.

**Table 5. Bus operations, 16-bit mode**

| Operation(1) | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | $\overline{RP}$ | $V_{PP}/\overline{WP}$ | Address Inputs | Data Inputs/Outputs |
|---|---|---|---|---|---|---|---|
| | | | | | | A22-A0 | DQ15A-1, DQ14-DQ0 |
| Bus Read | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | Cell Address | Data Output |
| Bus Write | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | Command Address | Data Input |
| Output Disable | X | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | X | Hi-Z |
| Standby | $V_{IH}$ | X | X | $V_{IH}$ | $V_{IH}$ | X | Hi-Z |

1. X = $V_{IL}$ or $V_{IH}$.

**Table 6. Read Electronic Signature, 16-bit mode**

| Read Cycle[1] | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | Address Inputs | | | | | | | | | Data Inputs/Outputs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | A22-A10 | A9 | A8-A7 | A6 | A5-A4 | A3 | A2 | A1 | A0 | DQ15A-1, DQ14-DQ0 |
| Manufacturer Code | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | X | $V_{ID}$ | X | $V_{IL}$ | X | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | 0020h |
| Device Code (Cycle 1) | | | | | | | | | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | 227Eh (Both Devices) |
| Device Code (Cycle 2) | | | | | | | | | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | $V_{IL}$ | 2212h (Both Devices) |
| Device Code (Cycle 3) | | | | | | | | | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | 228Ah (M29W128FH)<br>228Bh (M29W128FL) |

1. X = $V_{IL}$ or $V_{IH}$.

**Table 7. Block Protection, 16-bit mode**

| Operation[1] | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | $\overline{RP}$ | $V_{PP}/\overline{WP}$ | Address Inputs | | | | | | | | Data Inputs/Outputs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | A22-A12 | A11-A10 | A9 | A8-A7 | A6 | A5-A4 | A3-A2 | A1 | A0 | DQ15A-1, DQ14-DQ0 |
| Verify Extended Memory Block Indicator (bit DQ7) | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ | BA | X | $V_{ID}$ | X | $V_{IL}$ | X | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | M29W128FH<br>0088h (factory locked)<br>0008h (customer lockable) |
| | | | | | | | | | | | | | | | M29W128FL<br>0098h (factory locked)<br>0018h (customer lockable) |
| Verify Block Protection Status | | | | | | | | | | | | | | $V_{IL}$ | 0001h (protected)<br>0000h (unprotected) |
| Temporary Block Unprotect [2] | X | X | X | $V_{ID}$ | X | Valid | | | | | | | | | Data Input |

1. X = $V_{IL}$ or $V_{IH}$. BA Any Address in the Block.
2. The $\overline{RP}$ pin unprotects all the blocks that have been previously protected using a High Voltage protection Technique.

# 4 Hardware protection

The M29W128F features hardware protection/unprotection. Refer to *Table 8* for details on hardware block protection/unprotection using $V_{PP}/\overline{WP}$ and $\overline{RP}$ pins.

## 4.1 Write Protect

The $V_{PP}/\overline{WP}$ pin protects the highest or lowest block (refer to *Section 2: Signal descriptions* for a detailed description of the signals).

## 4.2 Temporary Block Unprotect

When held at $V_{ID}$, the Reset/Block Temporary Unprotect pin, $\overline{RP}$, will temporarily unprotect all the blocks previously protected using a High Voltage Block Protection technique.

**Table 8. Hardware Protection**

| $V_{PP}/\overline{WP}$ | $\overline{RP}$ | Function |
|---|---|---|
| $V_{IL}$ | $V_{IH}$ | Highest or lowest block protected from Program/Erase operations |
| | $V_{ID}$ | All blocks temporarily unprotected except the highest or lowest block |
| $V_{IH}$ or $V_{ID}$ | $V_{ID}$ | All blocks temporarily unprotected |
| $V_{PPH}$ | $V_{IH}$ or $V_{ID}$ | All blocks temporarily unprotected |

# 5 Command interface

All Bus Write operations to the memory are interpreted by the Command Interface. Commands consist of one or more sequential Bus Write operations. Failure to observe a valid sequence of Bus Write operations will result in the memory returning to Read mode. The long command sequences are imposed to maximize data security.

The address used for the commands changes depending on whether the memory is in 16-bit or 8-bit mode.

## 5.1 Standard commands

See either *Table 9*, or *Table 10*, depending on the configuration that is being used, for a summary of the Standard commands.

### 5.1.1 Read/Reset command

The Read/Reset command returns the memory to Read mode. It also resets the errors in the Status Register. Either one or three Bus Write operations can be used to issue the Read/Reset command.

The Read/Reset command can be issued, between Bus Write cycles before the start of a program or erase operation, to return the device to Read mode. If the Read/Reset command is issued during the time-out of a Block erase operation, the memory will take up to 10µs to abort. During the abort period no valid data can be read from the memory.

The Read/Reset command will not abort an Erase operation when issued while in Erase Suspend.

### 5.1.2 Auto Select command

The Auto Select command is used to read the Manufacturer Code, the Device Code, the Protection Status of each block (Block Protection Status) and the Extended Memory Block Protection Indicator.

Three consecutive Bus Write operations are required to issue the Auto Select command. Once the Auto Select command is issued Bus Read operations to specific addresses output the Manufacturer Code, the Device Code, the Extended Memory Block Protection Indicator and a Block Protection Status (see *Table 9* and *Table 10* in conjunction with *Table 3*, *Table 4*, *Table 6* and *Table 7*). The memory remains in Auto Select mode until a Read/Reset or CFI Query command is issued.

### 5.1.3 Read CFI Query command

The Read CFI Query Command is used to put the memory in Read CFI Query mode. Once in Read CFI Query mode, Bus Read operations to the memory will output data from the Common Flash Interface (CFI) Memory Area. One Bus Write cycle is required to issue the Read CFI Query Command. This command is valid only when the device is in the Read Array or Auto Select mode.

The Read/Reset command must be issued to return the device to the previous mode (the Read Array mode or Auto Select mode). A second Read/Reset command is required to put the device in Read Array mode from Auto Select mode.

See *Appendix B*, *Table 29*, *Table 30*, *Table 31*, *Table 32*, *Table 33* and *Table 34* for details on the information contained in the Common Flash Interface (CFI) memory area.

### 5.1.4 Chip Erase command

The Chip Erase command can be used to erase the entire chip. Six Bus Write operations are required to issue the Chip Erase Command and start the Program/Erase Controller.

If any blocks are protected, then these are ignored and all the other blocks are erased. If all of the blocks are protected the Chip Erase operation appears to start but will terminate within about 100µs, leaving the data unchanged. No error condition is given when protected blocks are ignored.

During the erase operation the memory will ignore all commands, including the Erase Suspend command. It is not possible to issue any command to abort the operation. Typical chip erase times are given in *Table 15*. All Bus Read operations during the Chip Erase operation will output the Status Register on the Data Inputs/Outputs. See the section on the Status Register for more details.

After the Chip Erase operation has completed the memory will return to the Read mode, unless an error has occurred. When an error occurs the memory will continue to output the Status Register. A Read/Reset command must be issued to reset the error condition and return to Read mode.

The Chip Erase Command sets all of the bits in unprotected blocks of the memory to '1'. All previous data is lost.

### 5.1.5 Block Erase command

The Block Erase command can be used to erase a list of one or more blocks. It sets all of the bits in the unprotected selected blocks to '1'. All previous data in the selected blocks is lost.

Six Bus Write operations are required to select the first block in the list. Each additional block in the list can be selected by repeating the sixth Bus Write operation using the address of the additional block. The Block Erase operation starts the Program/Erase Controller after a time-out period of 50µs after the last Bus Write operation. Once the Program/Erase Controller starts it is not possible to select any more blocks. Each additional block must therefore be selected within 50µs of the last block. The 50µs timer restarts when an additional block is selected. After the sixth Bus Write operation, a Bus Read operation outputs the Status Register. See *Section 6: Status register* for details on how to identify if the Program/Erase Controller has started the Block Erase operation.

After the Block Erase operation has completed, the memory returns to the Read mode, unless an error has occurred. When an error occurs, Bus Read operations will continue to output the Status Register. A Read/Reset command must be issued to reset the error condition and return to Read mode.

If any selected blocks are protected then these are ignored and all the other selected blocks are erased. If all of the selected blocks are protected the Block Erase operation appears to start but will terminate within about 100µs, leaving the data unchanged. No error condition is given when protected blocks are ignored.

During the Block Erase operation the memory will ignore all commands except the Erase Suspend command and the Read/Reset command which is only accepted during the 50µs time-out period. Typical block erase times are given in *Table 15*.

### 5.1.6 Erase Suspend command

The Erase Suspend command may be used to temporarily suspend a Block or multiple Block Erase operation. One Bus Write operation is required to issue the command. Issuing the Erase Suspend command returns the whole device to Read mode.

The Program/Erase Controller will suspend within the Erase Suspend Latency time (see *Table 15: Program, Erase Times and Program, Erase Endurance cycles*) of the Erase Suspend Command being issued. Once the Program/Erase Controller has stopped the memory will be set to Read mode and the Erase will be suspended. If the Erase Suspend command is issued during the period when the memory is waiting for an additional block (before the Program/Erase Controller starts) then the Erase is suspended immediately and will start immediately when the Erase Resume Command is issued. It is not possible to select any further blocks to erase after the Erase Resume.

During Erase Suspend it is possible to Read and Program cells in blocks that are not being erased; both Read and Program operations behave as normal on these blocks. If any attempt is made to program in a protected block or in the suspended block then the Program command is ignored and the data remains unchanged. The Status Register is not read and no error condition is given. Reading from blocks that are being erased will output the Status Register.

It is also possible to issue the Auto Select, Read CFI Query and Unlock Bypass commands during an Erase Suspend. The Read/Reset command must be issued to return the device to Read Array mode before the Resume command will be accepted.

During Erase Suspend a Bus Read operation to the Extended Memory Block will output the Extended Memory Block data. Once in the Extended Block mode, the Exit Extended Block command must be issued before the erase operation can be resumed.

### 5.1.7 Erase Resume command

The Erase Resume command is used to restart the Program/Erase Controller after an Erase Suspend.

The device must be in Read Array mode before the Resume command will be accepted. An Erase can be suspended and resumed more than once.

### 5.1.8 Program Suspend command

The Program Suspend command allows the system to interrupt a program operation so that data can be read from any block. When the Program Suspend command is issued during a program operation, the device suspends the program operation within the Program Suspend Latency time (see *Table 15: Program, Erase Times and Program, Erase Endurance cycles*) and updates the Status Register bits.

After the program operation has been suspended, the system can read array data from any address. However, data read from Program-Suspended addresses is not valid.

The Program Suspend command may also be issued during a program operation while an erase is suspended. In this case, data may be read from any addresses not in Erase Suspend or Program Suspend. If a read is needed from the Extended Memory Block area (One-time Program area), the user must use the proper command sequences to enter and exit this region.

The system may also issue the Auto Select command sequence when the device is in the Program Suspend mode. The system can read as many Auto Select codes as required.

When the device exits the Auto Select mode, the device reverts to the Program Suspend mode, and is ready for another valid operation. See Auto Select command sequence for more information.

### 5.1.9 Program Resume command

After the Program Resume command is issued, the device reverts to programming. The controller can determine the status of the program operation using the DQ7 or DQ6 status bits, just as in the standard program operation. Refer to *Section 6: Status register* for details.

The system must issue a Program Resume command, to exit the Program Suspend mode and to continue the programming operation.

Further issuing of the Resume command is ignored. Another Program Suspend command can be written after the device has resumed programming.

### 5.1.10 Program command

The Program command can be used to program a value to one address in the memory array at a time. The command requires four Bus Write operations, the final Write operation latches the address and data in the internal state machine and starts the Program/Erase Controller.

Programming can be suspended and then resumed by issuing a Program Suspend command and a Program Resume command, respectively (see *Section 5.1.8: Program Suspend command* and *Section 5.1.9: Program Resume command*).

If the address falls in a protected block then the Program command is ignored, the data remains unchanged. The Status Register is never read and no error condition is given.

After programming has started, Bus Read operations output the Status Register content. See *Section 6: Status register* for more details. Typical program times are given in *Table 15: Program, Erase Times and Program, Erase Endurance cycles*.

After the program operation has completed the memory will return to the Read mode, unless an error has occurred. When an error occurs, Bus Read operations to the memory continue to output the Status Register. A Read/Reset command must be issued to reset the error condition and return to Read mode.

One of the Erase Commands must be used to set all the bits in a block or in the whole memory from ’0’ to ’1’.

**Table 9. Standard commands, 8-bit mode**

| Command | | Length | Bus operations[1][2] | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | |
| | | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Read/Reset | | 1 | X | F0 | | | | | | | | | | |
| | | 3 | AAA | AA | 555 | 55 | X | F0 | | | | | | |
| Auto Select | Manufacturer Code | 3 | AAA | AA | 555 | 55 | AAA | 90 | (3) | (3) | | | | |
| | Device Code | | | | | | | | | | | | | |
| | Extended Memory Block Protection Indicator | | | | | | | | | | | | | |
| | Block Protection Status | | | | | | | | | | | | | |
| Program | | 4 | AAA | AA | 555 | 55 | AAA | A0 | PA | PD | | | | |
| Chip Erase | | 6 | AAA | AA | 555 | 55 | AAA | 80 | AAA | AA | 555 | 55 | AAA | 10 |
| Block Erase | | 6 + | AAA | AA | 555 | 55 | AAA | 80 | AAA | AA | 555 | 55 | BA | 30 |
| Erase/Program Suspend | | 1 | X | B0 | | | | | | | | | | |
| Erase/Program Resume | | 1 | X | 30 | | | | | | | | | | |
| Read CFI Query | | 1 | AA | 98 | | | | | | | | | | |

1. Grey cells represent Read cycles. The other cells are Write cycles.
2. X Don't Care, PA Program Address, PD Program Data, BA Any address in the Block. All values in the table are in hexadecimal.
3. The Auto Select addresses and data are given in *Table 3: Read Electronic Signature, 8-bit mode*, and *Table 4: Block Protection, 8-bit mode*, except for A9 that is 'Don't Care'.

**Table 10. Standard commands, 16-bit mode**

| Command | | Length | Bus operations[1][2] | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | |
| | | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Read/Reset | | 1 | X | F0 | | | | | | | | | | |
| | | 3 | 555 | AA | 2AA | 55 | X | F0 | | | | | | |
| Auto Select | Manufacturer Code | 3 | 555 | AA | 2AA | 55 | 555 | 90 | (3) | (3) | | | | |
| | Device Code | | | | | | | | | | | | | |
| | Extended Memory Block Protection Indicator | | | | | | | | | | | | | |
| | Block Protection Status | | | | | | | | | | | | | |
| Program | | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | |
| Chip Erase | | 6 | 555 | AA | 2AA | 55 | 555 | 80 | 555 | AA | 2AA | 55 | 555 | 10 |
| Block Erase | | 6+ | 555 | AA | 2AA | 55 | 555 | 80 | 555 | AA | 2AA | 55 | BA | 30 |
| Erase/Program Suspend | | 1 | X | B0 | | | | | | | | | | |
| Erase/Program Resume | | 1 | X | 30 | | | | | | | | | | |
| Read CFI Query | | 1 | 55 | 98 | | | | | | | | | | |

1. Gray cells represent Read cycles. The other cells are Write cycles.
2. X Don't Care, PA Program Address, PD Program Data, BA any address in the Block. All values in the table are in hexadecimal.
3. The Auto Select addresses and data are given in *Table 6: Read Electronic Signature, 16-bit mode*, and *Table 7: Block Protection, 16-bit mode*, except for A9 that is 'Don't Care'.

## 5.2 Fast Program commands

The M29W128F offers a set of Fast Program commands to improve the programming throughput:

- Write to Buffer and Program
- Double and Quadruple Word, Program
- Double, Quadruple and Octuple Byte Program
- Unlock Bypass.

See either *Table 12*, or *Table 11*, depending on the configuration that is being used, for a summary of the Fast Program commands.

When $V_{PPH}$ is applied to the $V_{PP}$/Write Protect pin the memory automatically enters the Fast Program mode. The user can then choose to issue any of the Fast Program commands. Care must be taken because applying a $V_{PPH}$ to the $V_{PP}/\overline{WP}$ pin will temporarily unprotect any protected block.

After programming has started, Bus Read operations in the memory output the Status Register content. Fast program commands can be suspended and then resumed by issuing a Program Suspend command and a Program Resume command, respectively (see *Section 5.1.8: Program Suspend command* and *Section 5.1.9: Program Resume command*)

After the fast program operation has completed, the memory will return to the Read mode, unless an error has occurred. When an error occurs Bus Read operations to the memory will continue to output the Status Register. A Read/Reset command must be issued to reset the error condition and return to Read mode. One of the Erase Commands must be used to set all the bits in a block or in the whole memory from ’0’ to ’1’.

Typical Program times are given in *Table 15: Program, Erase Times and Program, Erase Endurance cycles*.

### 5.2.1 Write to Buffer and Program command

The Write to Buffer and Program Command makes use of the device’s 64-Byte Write Buffer to speed up programming. 32 Words/64 Bytes can be loaded into the Write Buffer. Each Write Buffer has the same A22-A5 addresses.The Write to Buffer and Program command dramatically reduces system programming time compared to the standard non-buffered Program command.

When issuing a Write to Buffer and Program command, the $V_{PP}/\overline{WP}$ pin can be either held High, $V_{IH}$ or raised to $V_{PPH}$.

See *Table 15* for details on typical Write to Buffer and Program times in both cases.

Five successive steps are required to issue the Write to Buffer and Program command:

1. The Write to Buffer and Program command starts with two unlock cycles.
2. The third Bus Write cycle sets up the Write to Buffer and Program command. The setup code can be addressed to any location within the targeted block.
3. The fourth Bus Write cycle sets up the number of Words/Bytes to be programmed. Value N is written to the same block address, where N+1 is the number of Words/Bytes to be programmed. N+1 must not exceed the size of the Write Buffer or the operation will abort.
4. The fifth cycle loads the first address and data to be programmed.

Use N Bus Write cycles to load the address and data for each Word/Byte into the Write Buffer. Addresses must lie within the range from the start address+1 to the start address + N-1. Optimum performance is obtained when the start address corresponds to a 64 Byte boundary. If the start address is not aligned to a 64 Byte boundary, the total programming time is doubled.

All the addresses used in the Write to Buffer and Program operation must lie within the same page.

To program the content of the Write Buffer, this command must be followed by a Write to Buffer and Program Confirm command.

If an address is written several times during a Write to Buffer and Program operation, the address/data counter will be decremented at each data load operation and the data will be programmed to the last word loaded into the Buffer.

Invalid address combinations or failing to follow the correct sequence of Bus Write cycles will abort the Write to Buffer and Program.

The Status Register bits DQ1, DQ5, DQ6, DQ7 can be used to monitor the device status during a Write to Buffer and Program operation.

If is not possible to detect Program operation fails when changing programmed data from ‘0’ to ‘1’, that is when reprogramming data in a portion of memory already programmed. The resulting data will be the logical OR between the previous value and the current value.

A Write to Buffer and Program Abort and Reset command must be issued to abort the Write to Buffer and Program operation and reset the device in Read mode.

See *Appendix E*, *Figure 22: Write to Buffer and Program flowchart and Pseudo Code*, for a suggested flowchart on using the Write to Buffer and Program command.

### 5.2.2 Write to Buffer and Program Confirm command

The Write to Buffer and Program Confirm command is used to confirm a Write to Buffer and Program command and to program the N+1 Words/Bytes loaded in the Write Buffer by this command.

### 5.2.3 Write to Buffer and Program Abort and Reset command

The Write to Buffer and Program Abort and Reset command is used to abort Write to Buffer and Program command.

### 5.2.4 Double Word Program command

This is used to write two adjacent Words in x16 mode, simultaneously. The addresses of the two Words must differ only in A0.

Three bus write cycles are necessary to issue the command:

1. The first bus cycle sets up the command.
2. The second bus cycle latches the Address and the Data of the first Word to be written.
3. The third bus cycle latches the Address and the Data of the second Word to be written and starts the Program/Erase Controller.

### 5.2.5 Quadruple Word Program command

This is used to write a page of four adjacent Words (or 8 adjacent Bytes), in x16 mode, simultaneously. The addresses of the four Words must differ only in A1 and A0.

Five bus write cycles are necessary to issue the command:

1. The first bus cycle sets up the command.
2. The second bus cycle latches the Address and the Data of the first Word to be written.
3. The third bus cycle latches the Address and the Data of the second Word to be written.
4. The fourth bus cycle latches the Address and the Data of the third Word to be written.
5. The fifth bus cycle latches the Address and the Data of the fourth Word to be written and starts the Program/Erase Controller.

### 5.2.6 Double Byte Program command

This is used to write two adjacent Bytes in x8 mode, simultaneously. The addresses of the two Bytes must differ only in DQ15A-1.

Three bus write cycles are necessary to issue the command:

6. The first bus cycle sets up the command.
7. The second bus cycle latches the Address and the Data of the first Byte to be written.
8. The third bus cycle latches the Address and the Data of the second Byte to be written and starts the Program/Erase Controller.

### 5.2.7 Quadruple Byte Program command

This is used to write four adjacent Bytes in x8 mode, simultaneously. The addresses of the four Bytes must differ only in A0, DQ15A-1.

Five bus write cycles are necessary to issue the command.

1. The first bus cycle sets up the command.
2. The second bus cycle latches the Address and the Data of the first Byte to be written.
3. The third bus cycle latches the Address and the Data of the second Byte to be written.
4. The fourth bus cycle latches the Address and the Data of the third Byte to be written.
5. The fifth bus cycle latches the Address and the Data of the fourth Byte to be written and starts the Program/Erase Controller.

### 5.2.8 Octuple Byte Program command

This is used to write eight adjacent Bytes, in x8 mode, simultaneously. The addresses of the eight Bytes must differ only in A1, A0 and DQ15A-1.

Nine bus write cycles are necessary to issue the command:

1. The first bus cycle sets up the command.
2. The second bus cycle latches the Address and the Data of the first Byte to be written.
3. The third bus cycle latches the Address and the Data of the second Byte to be written.
4. The fourth bus cycle latches the Address and the Data of the third Byte to be written.
5. The fifth bus cycle latches the Address and the Data of the fourth Byte to be written.
6. The sixth bus cycle latches the Address and the Data of the fifth Byte to be written.
7. The seventh bus cycle latches the Address and the Data of the sixth Byte to be written.
8. The eighth bus cycle latches the Address and the Data of the seventh Byte to be written.
9. The ninth bus cycle latches the Address and the Data of the eighth Byte to be written and starts the Program/Erase Controller.

### 5.2.9 Unlock Bypass command

The Unlock Bypass command is used in conjunction with the Unlock Bypass Program command to program the memory faster than with the standard program commands. When the cycle time to the device is long, considerable time saving can be made by using these commands. Three Bus Write operations are required to issue the Unlock Bypass command.

Once the Unlock Bypass command has been issued, the memory enters Unlock Bypass mode. When in Unlock Bypass mode, only the Unlock Bypass Program and Unlock Bypass Reset commands are valid. The Unlock Bypass Program command can then be issued to program addresses within the memory, or the Unlock Bypass Reset command can be issued to return the memory to Read mode. In Unlock Bypass mode the memory can be read as if in Read mode.

### 5.2.10 Unlock Bypass Program command

The Unlock Bypass Program command can be used to program one address in the memory array at a time. The command requires two Bus Write operations, the final write operation latches the address and data and starts the Program/Erase Controller.

The Program operation using the Unlock Bypass Program command behaves identically to the Program operation using the Program command. The operation cannot be aborted, a Bus Read operation to the memory outputs the Status Register. See the Program command for details on the behavior.

### 5.2.11 Unlock Bypass Reset command

The Unlock Bypass Reset command can be used to return to Read/Reset mode from Unlock Bypass mode. Two Bus Write operations are required to issue the Unlock Bypass Reset command. Read/Reset command does not exit from Unlock Bypass mode.

**Table 11. Fast Program commands, 8-bit mode**

| Command | Length | Bus Write operations(1) | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | | 7th | | 8th | | 9th | |
| | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Write to Buffer and Program | N +5 | AAA | AA | 555 | 55 | BA | 25 | BA | N(2) | PA(3) | PD | WBL (4) | PD | | | | | | |
| Write to Buffer and Program Abort and Reset | 3 | AAA | AA | 555 | 55 | AAA | F0 | | | | | | | | | | | | |
| Write to Buffer and Program Confirm | 1 | BA(5) | 29 | | | | | | | | | | | | | | | | |
| Double Byte Program | 3 | AAA | 50 | PA0 | PD0 | PA1 | PD1 | | | | | | | | | | | | |
| Quadruple Byte Program | 5 | AAA | 56 | PA0 | PD0 | PA1 | PD1 | PA2 | PD2 | PA3 | PD3 | | | | | | | | |
| Octuple Byte Program | 9 | AAA | 8B | PA0 | PD0 | PA1 | PD1 | PA2 | PD2 | PA3 | PD3 | PA4 | PD4 | PA5 | PD5 | PA6 | PD6 | PA7 | PD7 |
| Unlock Bypass | 3 | AAA | AA | 555 | 55 | AAA | 20 | | | | | | | | | | | | |
| Unlock Bypass Program | 2 | X | A0 | PA | PD | | | | | | | | | | | | | | |
| Unlock Bypass Reset | 2 | X | 90 | X | 00 | | | | | | | | | | | | | | |

1. X Don't Care, PA Program Address, PD Program Data, BA Any address in the Block, WBL Write Buffer Location. All values in the table are in hexadecimal.
2. The maximum number of cycles in the command sequence is 68. N+1 is the number of Bytes to be programmed during the Write to Buffer and Program operation.
3. Each buffer has the same A22-A5 addresses. A0-A4 and A-1 are used to select a Byte within the N+1 Byte page.
4. The 6th cycle has to be issued N time. WBL scans the Word inside the page.
5. BA must be identical to the address loaded during the Write to buffer and Program 3rd and 4th cycles.

**Table 12. Fast Program commands, 16-bit mode**

| Command | Length | Bus Write operations(1) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | |
| | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Write to Buffer and Program | N+5 | 555 | AA | 2AA | 55 | BA | 25 | BA | N(2) | PA(3) | PD | WBL (4) | PD |
| Write to Buffer and Program Abort and Reset | 3 | 555 | AA | 2AA | 55 | 555 | F0 | | | | | | |
| Write to Buffer and Program Confirm | 1 | BA(5) | 29 | | | | | | | | | | |
| Double Word Program | 3 | 555 | 50 | PA0 | PD0 | PA1 | PD1 | | | | | | |
| Quadruple Word Program | 5 | 555 | 56 | PA0 | PD0 | PA1 | PD1 | PA2 | PD2 | PA3 | PD3 | | |
| Unlock Bypass | 3 | 555 | AA | 2AA | 55 | 555 | 20 | | | | | | |
| Unlock Bypass Program | 2 | X | A0 | PA | PD | | | | | | | | |
| Unlock Bypass Reset | 2 | X | 90 | X | 00 | | | | | | | | |

1. X Don't Care, PA Program Address, PD Program Data, BA Any address in the Block, WBL Write Buffer Location. All values in the table are in hexadecimal.
2. The maximum number of cycles in the command sequence is 36. N+1 is the number of Words to be programmed during the Write to Buffer and Program operation.
3. Each buffer has the same A22-A5 addresses. A0-A4 are used to select a Word within the N+1 Word page.
4. The 6th cycle has to be issued N time. WBL scans the Word inside the page.
5. BA must be identical to the address loaded during the Write to buffer and Program 3rd and 4th cycles.

## 5.3 Extended Memory Block Protection commands

The M29W128F offers a set of commands to access the Extended Memory Block and to configure and check its protection mode.

The commands related to the Extended Memory Block Protection are available in both 8 bit and 16 bit memory configuration.

### 5.3.1 Enter Extended Memory Block command

The M29W128F has one extra 128 Word block (Extended Memory Block) that can only be accessed using the Enter Extended Memory Block command.

Three Bus Write cycles are required to issue the Extended Memory Block command. Once the command has been issued the device enters the Extended Memory Block mode where all Bus Read or Program operations are conducted on the Extended Memory Block. Once the device is in the Extended Block mode, the Extended Memory Block is addressed by using the addresses occupied by block 0 in the other operating modes (see *Table 28: Block Addresses and Protection Groups*).

The device remains in Extended Block mode until the Exit Extended Block command is issued or power is removed from the device. After power-up or a hardware reset, the device reverts to the Read mode where commands issued to block 0 address space will address block 0.

Note that when the device is in the Extended Block mode, the $V_{PP}/\overline{WP}$ pin cannot be used for fast programming and the Unlock Bypass mode is not available.

The Extended Memory Block cannot be erased, and can be treated as one-time programmable (OTP) memory.

In Extended Block mode, Erase, Chip Erase, Erase Suspend and Erase resume commands are not allowed.

To exit from the Extended Block mode the Exit Extended Block command must be issued.

The Extended Memory Block can be protected by setting the Extended Memory Block Protection Bit to ‘1’; however once protected the protection cannot be undone.

### 5.3.2 Exit Extended Block command

The Exit Extended Block command is used to exit from the Extended Block mode and return the device to Read mode. Four Bus Write operations are required to issue the command.

**Table 13. Extended Block Protection commands, 8-bit mode**

| Command | Length | Bus operations(1)(2) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | |
| | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Enter Extended Block | 3 | AAA | AA | 555 | 55 | AAA | 88 | | | | | | |
| Exit Extended Block | 4 | AAA | AA | 555 | 55 | AAA | 90 | X | 00 | | | | |

1. X Don't Care. All values in the table are in hexadecimal.
2. Grey cells represent Read cycles. The other cells are Write cycles.

**Table 14. Block Protection commands, 16-bit mode**

| Command | Length | Bus operations(1)(2)(3) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | | 7th | |
| | | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data | Add | Data |
| Enter Extended Block | 3 | 555 | AA | 2AA | 55 | 555 | 88 | | | | | | | | |
| Exit Extended Block | 4 | 555 | AA | 2AA | 55 | 555 | 90 | X | 00 | | | | | | |

1. Grey cells represent Read cycles. The other cells are Write cycles.
2. X Don't Care. All values in the table are in hexadecimal.
3. During Command cycles, if the lower address bits are 555h or 2AAh then the address bits higher than A11 and data bits higher than DQ7 are Don't Care.

**Table 15. Program, Erase Times and Program, Erase Endurance cycles**

| Parameter | | | Min | Typ (1)(2) | Max(2) | Unit |
|---|---|---|---|---|---|---|
| Chip Erase | | | | 80 | 400(3) | s |
| Block Erase (64 KBytes) | | | | 0.8 | 6(4) | s |
| Erase Suspend Latency Time | | | | | 50(4) | µs |
| Byte Program | Single or Multiple Byte Program (1, 2, 4 or 8 Bytes at-a-time) | | | 10 | 200(3) | µs |
| | Write to Buffer and Program (64 Bytes at-a-time) | $V_{PP}/\overline{WP}$ =$V_{PPH}$ | | 90 | | µs |
| | | $V_{PP}/\overline{WP}$=$V_{IH}$ | | 280 | | |
| Word Program | Single or Multiple Word Program (1, 2 or 4 Words at-a-time) | | | 10 | 200(3) | µs |
| | Write to Buffer and Program (32 Words at-a-time) | $V_{PP}/\overline{WP}$= $V_{PPH}$ | | 90 | | µs |
| | | $V_{PP}/\overline{WP}$=$V_{IH}$ | | 280 | | |
| Chip Program (Byte by Byte) | | | | 80 | 400(3) | s |
| Chip Program (Word by Word) | | | | 40 | 200(3) | s |
| Chip Program (Quadruple Byte or Double Word) | | | | 20 | 100(3) | s |
| Chip Program (Octuple Byte or Quadruple Word) | | | | 10 | 50(3) | s |
| Program Suspend Latency Time | | | | 5 | 15 | µs |
| Program/Erase Cycles (per Block) | | | 100,000 | | | cycles |
| Data Retention | | | 20 | | | years |

1. Typical values measured at room temperature and nominal voltages.
2. Sampled, but not 100% tested.
3. Maximum value measured at worst case conditions for both temperature and $V_{CC}$ after 100,00 program/erase cycles.
4. Maximum value measured at worst case conditions for both temperature and $V_{CC}$.

# 6 Status register

The M29W128F has one Status Register. The Status Register provides information on the current or previous Program or Erase operations. The various bits convey information and errors on the operation. Bus Read operations from any address within the memory, always read the Status Register during Program and Erase operations. It is also read during Erase Suspend when an address within a block being erased is accessed.

The bits in the Status Register are summarized in *Table 16: Status register bits*.

### 6.0.1 Data Polling Bit (DQ7)

The Data Polling Bit can be used to identify whether the Program/Erase Controller has successfully completed its operation or if it has responded to an Erase Suspend. The Data Polling Bit is output on DQ7 when the Status Register is read.

During Program operations the Data Polling Bit outputs the complement of the bit being programmed to DQ7. After successful completion of the Program operation the memory returns to Read mode and Bus Read operations from the address just programmed output DQ7, not its complement.

During Erase operations the Data Polling Bit outputs ’0’, the complement of the erased state of DQ7. After successful completion of the Erase operation the memory returns to Read mode.

In Erase Suspend mode the Data Polling Bit will output a ’1’ during a Bus Read operation within a block being erased. The Data Polling Bit will change from a ’0’ to a ’1’ when the Program/Erase Controller has suspended the Erase operation.

*Figure 5: Data Polling flowchart*, gives an example of how to use the Data Polling Bit. A Valid Address is the address being programmed or an address within the block being erased.

### 6.0.2 Toggle Bit (DQ6)

The Toggle Bit can be used to identify whether the Program/Erase Controller has successfully completed its operation or if it has responded to an Erase Suspend. The Toggle Bit is output on DQ6 when the Status Register is read.

During a Program/Erase operation the Toggle Bit changes from ’0’ to ’1’ to ’0’, etc., with successive Bus Read operations at any address. After successful completion of the operation the memory returns to Read mode.

During Erase Suspend mode the Toggle Bit will output when addressing a cell within a block being erased. The Toggle Bit will stop toggling when the Program/Erase Controller has suspended the Erase operation.

*Figure 6: Toggle flowchart*, gives an example of how to use the Data Toggle Bit.

### 6.0.3 Error Bit (DQ5)

The Error Bit can be used to identify errors detected by the Program/Erase Controller. The Error Bit is set to ’1’ when a Program, Block Erase or Chip Erase operation fails to write the correct data to the memory. If the Error Bit is set a Read/Reset command must be issued before other commands are issued. The Error bit is output on DQ5 when the Status Register is read.

Note that the Program command cannot change a bit set to ’0’ back to ’1’ and attempting to do so will set DQ5 to ‘1’. A Bus Read operation to that address will show the bit is still ‘0’. One of the Erase commands must be used to set all the bits in a block or in the whole memory from ’0’ to ’1’.

### 6.0.4 Erase Timer Bit (DQ3)

The Erase Timer Bit can be used to identify the start of Program/Erase Controller operation during a Block Erase command. Once the Program/Erase Controller starts erasing the Erase Timer Bit is set to ’1’. Before the Program/Erase Controller starts the Erase Timer Bit is set to ’0’ and additional blocks to be erased may be written to the Command Interface. The Erase Timer Bit is output on DQ3 when the Status Register is read.

### 6.0.5 Alternative Toggle Bit (DQ2)

The Alternative Toggle Bit can be used to monitor the Program/Erase controller during Erase operations. The Alternative Toggle Bit is output on DQ2 when the Status Register is read.

During Chip Erase and Block Erase operations the Toggle Bit changes from ’0’ to ’1’ to ’0’, etc., with successive Bus Read operations from addresses within the blocks being erased. A protected block is treated the same as a block not being erased. Once the operation completes the memory returns to Read mode.

During Erase Suspend the Alternative Toggle Bit changes from ’0’ to ’1’ to ’0’, etc. with successive Bus Read operations from addresses within the blocks being erased. Bus Read operations to addresses within blocks not being erased will output the memory array data as if in Read mode.

After an Erase operation that causes the Error Bit to be set, the Alternative Toggle Bit can be used to identify which block or blocks have caused the error. The Alternative Toggle Bit changes from ’0’ to ’1’ to ’0’, etc. with successive Bus Read Operations from addresses within blocks that have not erased correctly. The Alternative Toggle Bit does not change if the addressed block has erased correctly.

### 6.0.6 Write to Buffer and Program Abort Bit (DQ1)

The Write to Buffer and Program Abort bit, DQ1, is set to ‘1’ when a Write to Buffer and Program operation aborts. The Write to Buffer and Program Abort and Reset command must be issued to return the device to Read mode (see Write to Buffer and Program in COMMANDS section).

**Table 16. Status register bits(1)**

| Operation | DQ7 | DQ6 | DQ5 | DQ3 | DQ2 | DQ1 | RB |
|---|---|---|---|---|---|---|---|
| Program | $\overline{DQ7}$ | Toggle | 0 | – | – | 0 | 0 |
| Program During Erase Suspend | $\overline{DQ7}$ | Toggle | 0 | – | – | – | 0 |
| Write to Buffer and Program Abort | $\overline{DQ7}$ | Toggle | 0 | – | – | 1 | 0 |
| Program Error | $\overline{DQ7}$ | Toggle | 1 | – | – | – | Hi-Z |
| Chip Erase | 0 | Toggle | 0 | 1 | Toggle | – | 0 |
| Block Erase before timeout | 0 | Toggle | 0 | 0 | Toggle | – | 0 |
| | 0 | Toggle | 0 | 0 | No Toggle | – | 0 |
| Block Erase | 0 | Toggle | 0 | 1 | Toggle | – | 0 |
| | 0 | Toggle | 0 | 1 | No Toggle | – | 0 |
| Erase Suspend | 1 | No Toggle | 0 | – | Toggle | – | Hi-Z |
| | Data read as normal | | | | | – | Hi-Z |
| Erase Error | 0 | Toggle | 1 | 1 | No Toggle | – | Hi-Z |
| | 0 | Toggle | 1 | 1 | Toggle | – | Hi-Z |

1. Unspecified data bits should be ignored.

**Figure 5. Data Polling flowchart**

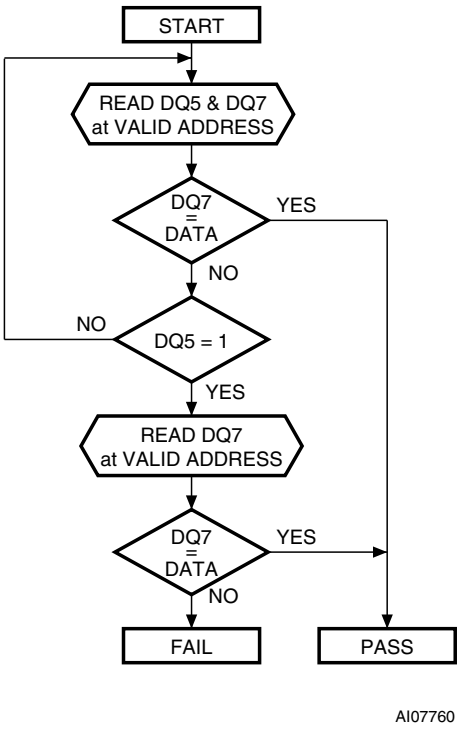
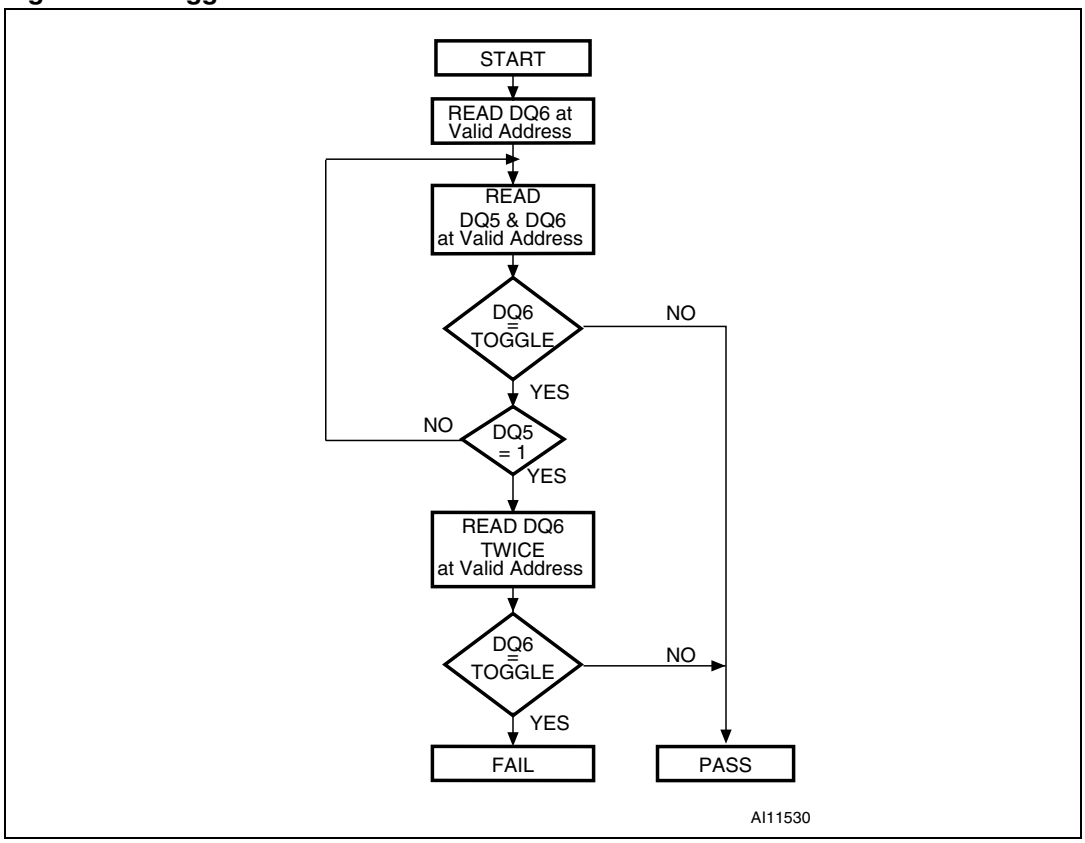


**Figure 6. Toggle flowchart**



# 7 Maximum rating

Stressing the device above the rating listed in the Absolute Maximum Ratings table may cause permanent damage to the device. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Refer also to the Numonyx SURE Program and other relevant quality documents.

**Table 17. Absolute maximum ratings**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $T_{BIAS}$ | Temperature Under Bias | –50 | 125 | °C |
| $T_{STG}$ | Storage Temperature | –65 | 150 | °C |
| $V_{IO}$ | Input or Output voltage(1)(2) | –0.6 | $V_{CC}$ + 0.6 | V |
| $V_{CC}$ | Supply voltage | –0.6 | 4 | V |
| $V_{ID}$ | Identification voltage | –0.6 | 13.5 | V |
| $V_{PP}$(3) | Program voltage | –0.6 | 13.5 | V |

1. Minimum voltage may undershoot to –2V during transition and for less than 20ns during transitions.
2. Maximum voltage may overshoot to $V_{CC}$ + 2V during transition and for less than 20ns during transitions.
3. $V_{PP}$ must not remain at 12V for more than a total of 80hrs.

# 8 DC and AC parameters

This section summarizes the operating measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC characteristics Tables that follow, are derived from tests performed under the Measurement Conditions summarized in *Table 18: Operating and AC measurement conditions*. Designers should check that the operating conditions in their circuit match the operating conditions when relying on the quoted parameters.

**Table 18. Operating and AC measurement conditions**

| Parameter | M29W128FH, M29W128FL | | | | Unit |
|---|---|---|---|---|---|
| | 60 | | 70 | | |
| | Min | Max | Min | Max | |
| $V_{CC}$ supply voltage | 2.7 | 3.6 | 2.7 | 3.6 | V |
| Ambient Operating Temperature | –40 | 85 | –40 | 85 | °C |
| Load capacitance ($C_L$) | 30 | | 30 | | pF |
| Input Rise and Fall Times | | 10 | | 10 | ns |
| Input pulse voltages | 0 to $V_{CC}$ | | 0 to $V_{CC}$ | | V |
| Input and Output Timing Ref. voltages | $V_{CC}/2$ | | $V_{CC}/2$ | | V |

**Figure 7. AC measurement load circuit**

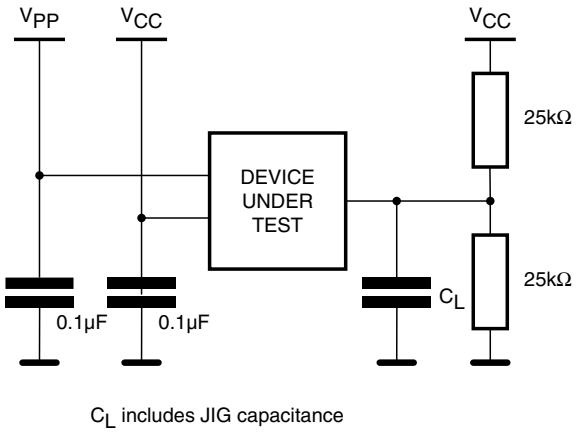


**Figure 8. AC measurement I/O waveform**

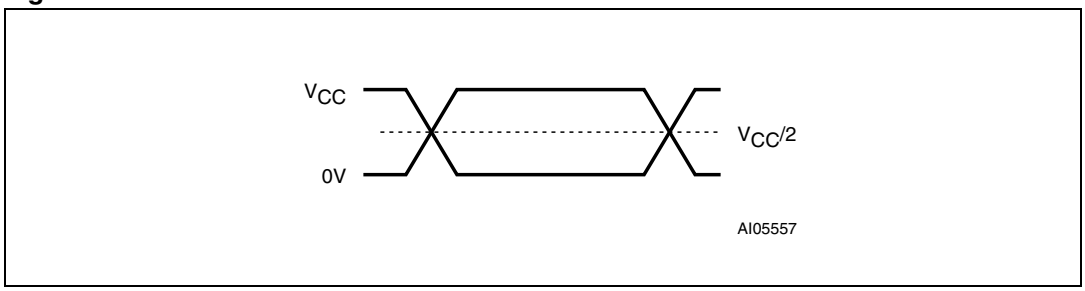



**Table 19. Device capacitance(1)**

| Symbol | Parameter | Test condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 0V | | 6 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 0V | | 12 | pF |

1. Sampled only, not 100% tested.

**Table 20. DC characteristics**

| Symbol | Parameter | Test condition | | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{LI}$ | Input Leakage Current | 0V ≤$V_{IN}$ ≤$V_{CC}$ | | | ±1 | µA |
| $I_{LO}$ | Output Leakage Current | 0V ≤$V_{OUT}$ ≤$V_{CC}$ | | | ±1 | µA |
| $I_{CC1}$ | Supply Current (Read) | $\overline{E}$ = $V_{IL}$, $\overline{G}$ = $V_{IH}$, f = 6MHz | | | 10 | mA |
| $I_{CC2}$ | Supply Current (Standby) | $\overline{E}$ = $V_{CC}$ ± 0.2V, $\overline{RP}$ = $V_{CC}$ ± 0.2V | | | 100 | µA |
| $I_{CC3}$ (1) | Supply Current (Program/Erase) | Program/Erase Controller active | $V_{PP}/\overline{WP}$ = $V_{IL}$ or $V_{IH}$ | | 20 | mA |
| | | | $V_{PP}/\overline{WP}$ = $V_{PPH}$ | | 20 | mA |
| $V_{IL}$ | Input Low voltage | | | –0.5 | 0.8 | V |
| $V_{IH}$ | Input High voltage | | | 0.7$V_{CC}$ | $V_{CC}$ + 0.3 | V |
| $V_{PPH}$ | Voltage for $V_{PP}/\overline{WP}$ Program Acceleration | $V_{CC}$ = 2.7V ±10% | | 11.5 | 12.5 | V |
| $I_{PP}$ | Current for $V_{PP}/\overline{WP}$ Program Acceleration | $V_{CC}$ = 2.7V ±10% | | | 15 | mA |
| $V_{OL}$ | Output Low voltage | $I_{OL}$ = 1.8mA | | | 0.45 | V |
| $V_{OH}$ | Output High voltage | $I_{OH}$ = –100µA | | $V_{CC}$ –0.4 | | V |
| $V_{ID}$ | Identification voltage | | | 11.5 | 12.5 | V |
| $V_{LKO}$ | Program/Erase Lockout Supply voltage | | | 1.8 | 2.3 | V |

1. Sampled only, not 100% tested.

**Figure 9. Random Read AC waveforms**

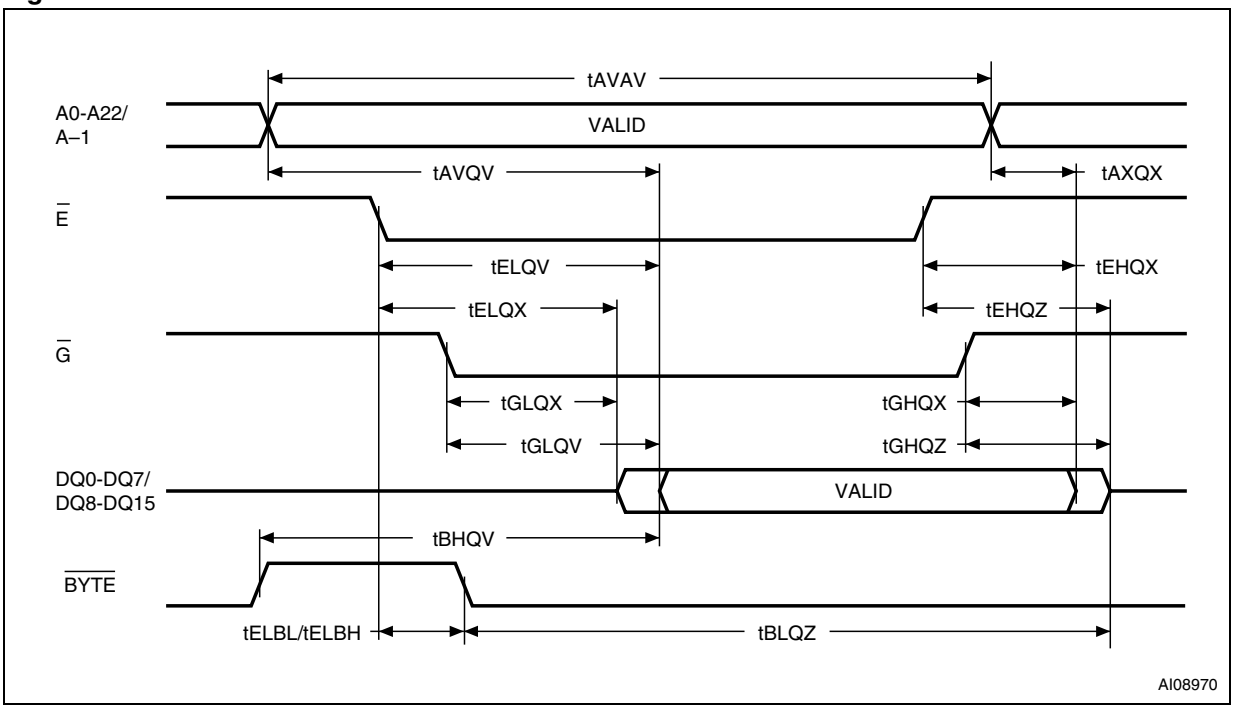


**Figure 10. Page Read AC waveforms (Word mode)**

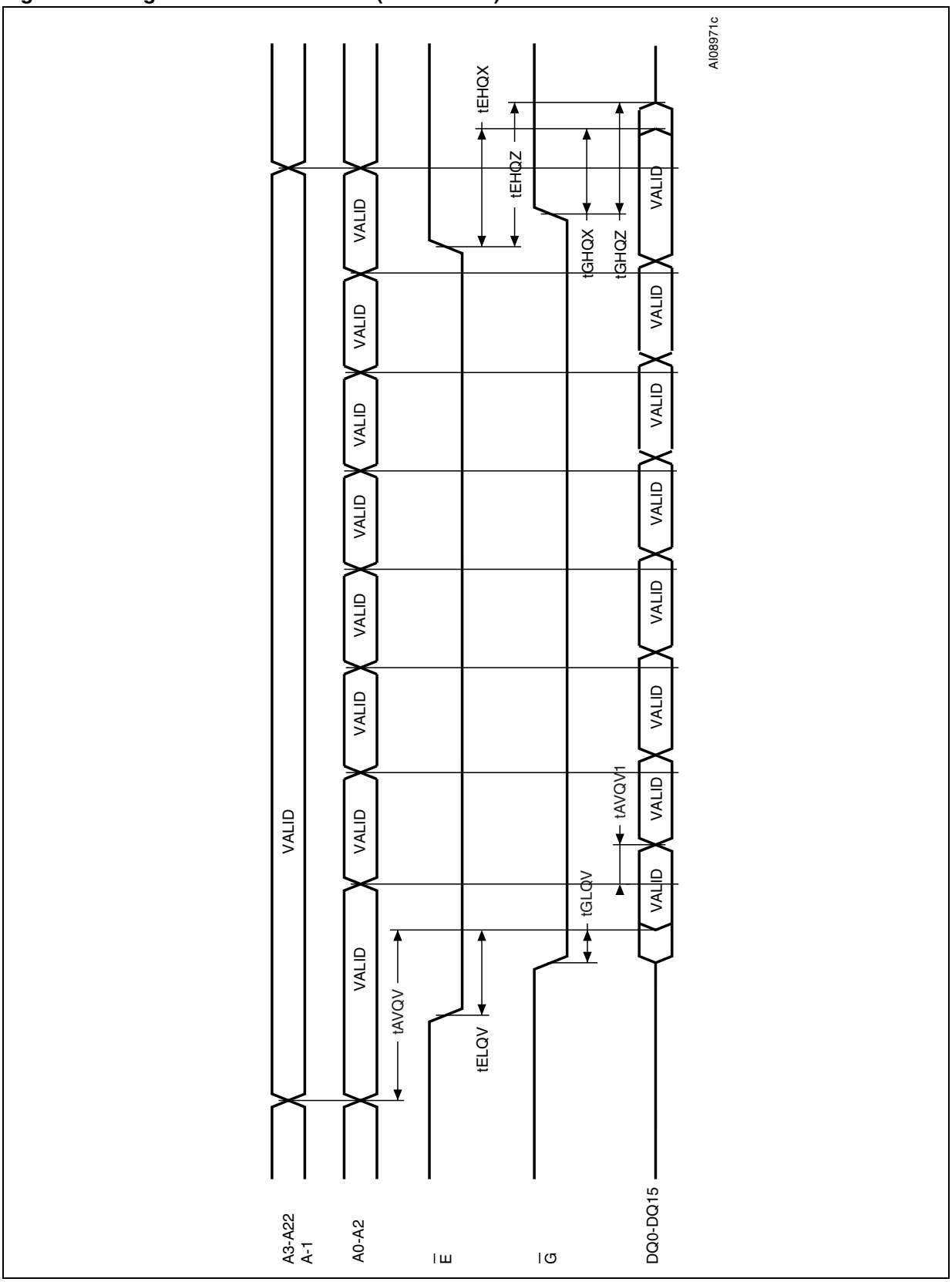


**Table 21. Read AC characteristics**

| Symbol | Alt | Parameter | Test condition | | M29W128FH, M29W128FL | | Unit |
|---|---|---|---|---|---|---|---|
| | | | | | 60 | 70 | |
| $t_{AVAV}$ | $t_{RC}$ | Address Valid to Next Address Valid | $\overline{E} = V_{IL}$, $\overline{G} = V_{IL}$ | Min | 60 | 70 | ns |
| $t_{AVQV}$ | $t_{ACC}$ | Address Valid to Output Valid | $\overline{E} = V_{IL}$, $\overline{G} = V_{IL}$ | Max | 60 | 70 | ns |
| $t_{AVQV1}$ | $t_{PAGE}$ | Address Valid to Output Valid (Page) | $\overline{E} = V_{IL}$, $\overline{G} = V_{IL}$ | Max | 25 | 30 | ns |
| $t_{ELQX}$(1) | $t_{LZ}$ | Chip Enable Low to Output Transition | $\overline{G} = V_{IL}$ | Min | 0 | 0 | ns |
| $t_{ELQV}$ | $t_{CE}$ | Chip Enable Low to Output Valid | $\overline{G} = V_{IL}$ | Max | 60 | 70 | ns |
| $t_{GLQX}$(1) | $t_{OLZ}$ | Output Enable Low to Output Transition | $\overline{E} = V_{IL}$ | Min | 0 | 0 | ns |
| $t_{GLQV}$ | $t_{OE}$ | Output Enable Low to Output Valid | $\overline{E} = V_{IL}$ | Max | 20 | 25 | ns |
| $t_{EHQZ}$(1) | $t_{HZ}$ | Chip Enable High to Output Hi-Z | $\overline{G} = V_{IL}$ | Max | 25 | 25 | ns |
| $t_{GHQZ}$(1) | $t_{DF}$ | Output Enable High to Output Hi-Z | $\overline{E} = V_{IL}$ | Max | 25 | 25 | ns |
| $t_{EHQX}$ $t_{GHQX}$ $t_{AXQX}$ | $t_{OH}$ | Chip Enable, Output Enable or Address Transition to Output Transition | | Min | 0 | 0 | ns |
| $t_{ELBL}$ $t_{ELBH}$ | $t_{ELFL}$ $t_{ELFH}$ | Chip Enable to $\overline{BYTE}$ Low or High | | Max | 5 | 5 | ns |
| $t_{BLQZ}$ | $t_{FLQZ}$ | $\overline{BYTE}$ Low to Output Hi-Z | | Max | 25 | 25 | ns |
| $t_{BHQV}$ | $t_{FHQV}$ | $\overline{BYTE}$ High to Output Valid | | Max | 30 | 30 | ns |

1. Sampled only, not 100% tested.

**Figure 11. Write AC waveforms, Write Enable Controlled**

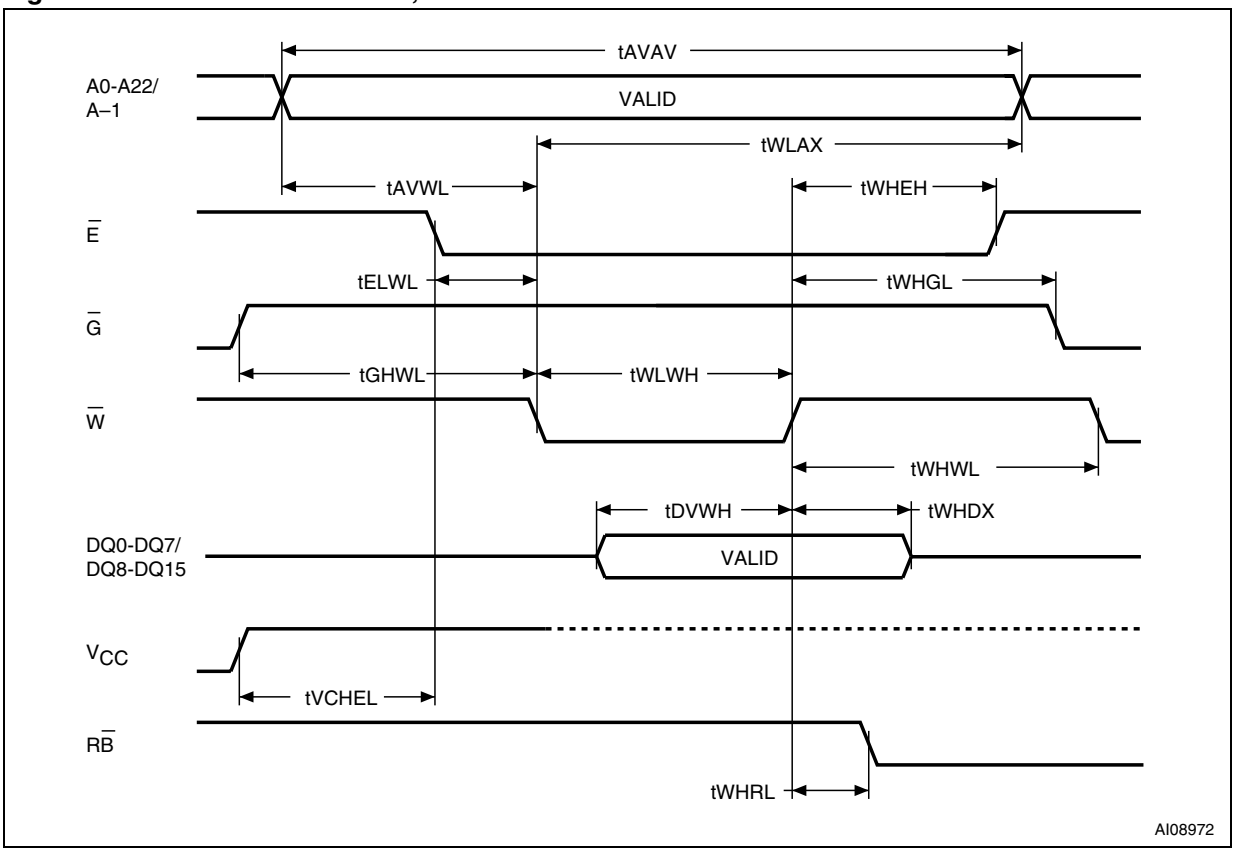



**Table 22. Write AC characteristics, Write Enable Controlled**

| Symbol | Alt | Parameter | | M29W128FH, M29W128FL | | Unit |
|---|---|---|---|---|---|---|
| | | | | 60 | 70 | |
| $t_{AVAV}$ | $t_{WC}$ | Address Valid to Next Address Valid | Min | 60 | 70 | ns |
| $t_{ELWL}$ | $t_{CS}$ | Chip Enable Low to Write Enable Low | Min | 0 | 0 | ns |
| $t_{WLWH}$ | $t_{WP}$ | Write Enable Low to Write Enable High | Min | 45 | 45 | ns |
| $t_{DVWH}$ | $t_{DS}$ | Input Valid to Write Enable High | Min | 45 | 45 | ns |
| $t_{WHDX}$ | $t_{DH}$ | Write Enable High to Input Transition | Min | 0 | 0 | ns |
| $t_{WHEH}$ | $t_{CH}$ | Write Enable High to Chip Enable High | Min | 0 | 0 | ns |
| $t_{WHWL}$ | $t_{WPH}$ | Write Enable High to Write Enable Low | Min | 30 | 30 | ns |
| $t_{AVWL}$ | $t_{AS}$ | Address Valid to Write Enable Low | Min | 0 | 0 | ns |
| $t_{WLAX}$ | $t_{AH}$ | Write Enable Low to Address Transition | Min | 45 | 45 | ns |
| $t_{GHWL}$ | | Output Enable High to Write Enable Low | Min | 0 | 0 | ns |
| $t_{WHGL}$ | $t_{OEH}$ | Write Enable High to Output Enable Low | Min | 0 | 0 | ns |
| $t_{WHRL}$ (1) | $t_{BUSY}$ | Program/Erase Valid to $\overline{RB}$ Low | Max | 30 | 30 | ns |
| $t_{VCHEL}$ | $t_{VCS}$ | $V_{CC}$ High to Chip Enable Low | Min | 50 | 50 | µs |

1. Sampled only, not 100% tested.

**Figure 12. Write AC waveforms, Chip Enable Controlled**

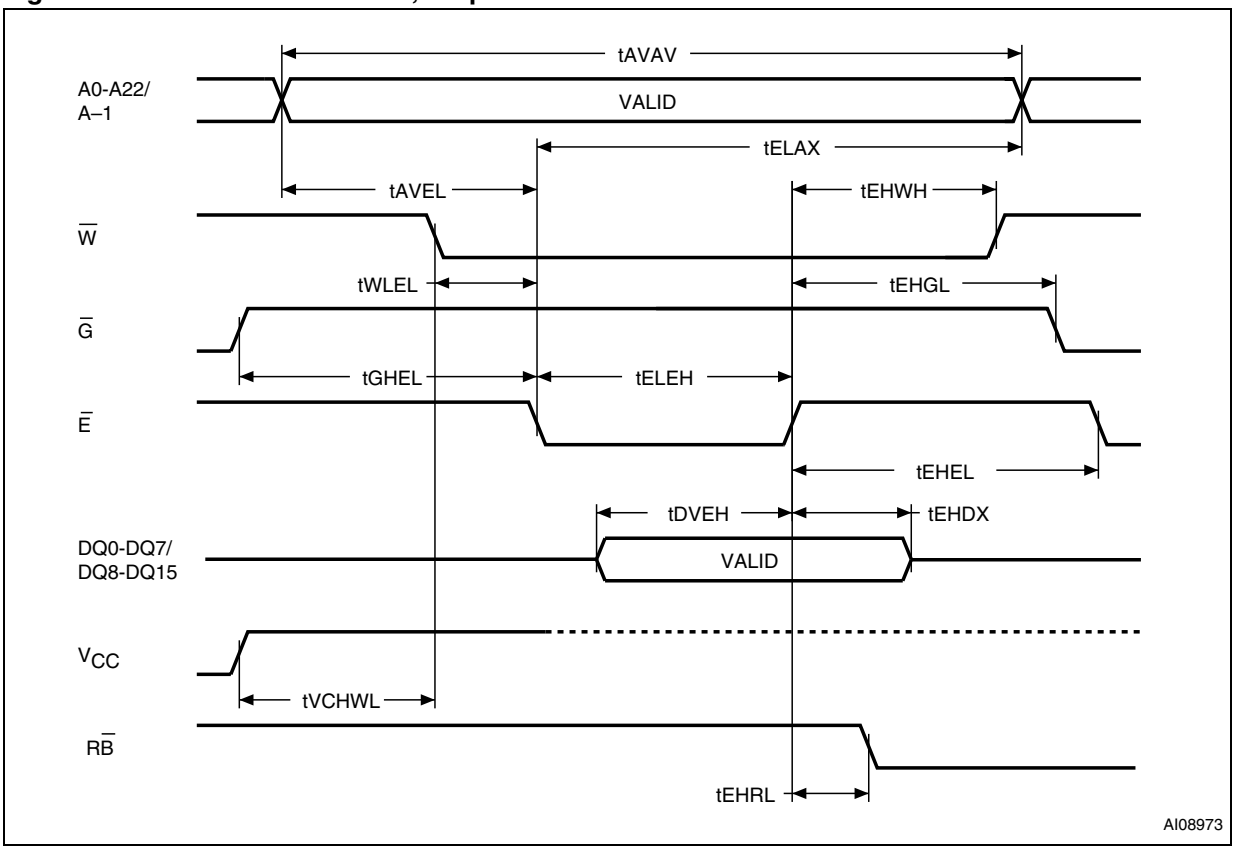



**Table 23. Write AC characteristics, Chip Enable Controlled**

| Symbol | Alt | Parameter | | M29W128FH, M29W128FL | | Unit |
|---|---|---|---|---|---|---|
| | | | | 60 | 70 | |
| $t_{AVAV}$ | $t_{WC}$ | Address Valid to Next Address Valid | Min | 60 | 70 | ns |
| $t_{WLEL}$ | $t_{WS}$ | Write Enable Low to Chip Enable Low | Min | 0 | 0 | ns |
| $t_{ELEH}$ | $t_{CP}$ | Chip Enable Low to Chip Enable High | Min | 45 | 45 | ns |
| $t_{DVEH}$ | $t_{DS}$ | Input Valid to Chip Enable High | Min | 45 | 45 | ns |
| $t_{EHDX}$ | $t_{DH}$ | Chip Enable High to Input Transition | Min | 0 | 0 | ns |
| $t_{EHWH}$ | $t_{WH}$ | Chip Enable High to Write Enable High | Min | 0 | 0 | ns |
| $t_{EHEL}$ | $t_{CPH}$ | Chip Enable High to Chip Enable Low | Min | 30 | 30 | ns |
| $t_{AVEL}$ | $t_{AS}$ | Address Valid to Chip Enable Low | Min | 0 | 0 | ns |
| $t_{ELAX}$ | $t_{AH}$ | Chip Enable Low to Address Transition | Min | 45 | 45 | ns |
| $t_{GHEL}$ | | Output Enable High Chip Enable Low | Min | 0 | 0 | ns |
| $t_{EHGL}$ | $t_{OEH}$ | Chip Enable High to Output Enable Low | Min | 0 | 0 | ns |
| $t_{EHRL}$ (1) | $t_{BUSY}$ | Program/Erase Valid to $\overline{RB}$ Low | Max | 30 | 30 | ns |
| $t_{VCHWL}$ | $t_{VCS}$ | $V_{CC}$ High to Write Enable Low | Min | 50 | 50 | µs |

1. Sampled only, not 100% tested.

**Figure 13. Reset/Block Temporary Unprotect AC waveforms (No Program/Erase ongoing)**

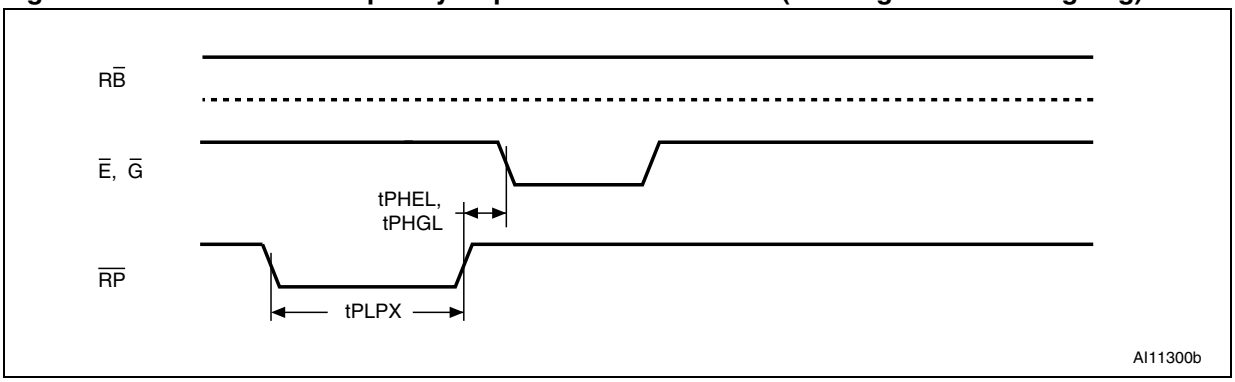



**Figure 14. Reset/Block Temporary Unprotect during Program/Erase operation AC waveforms**

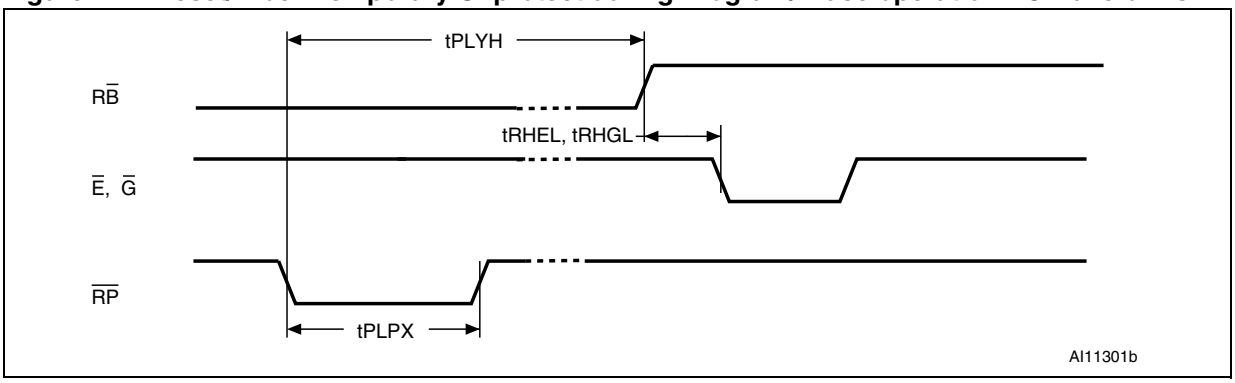
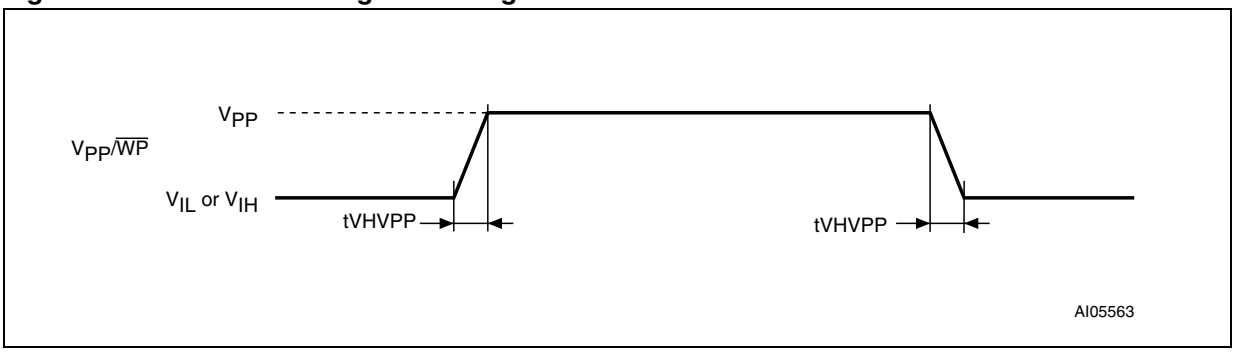



**Table 24. Reset/Block Temporary Unprotect AC characteristics**

| Symbol | Alt | Parameter | | M29W128FH, M29W128FL | | Unit |
|---|---|---|---|---|---|---|
| | | | | 60 | 70 | |
| $t_{PLYH}$ (1) | $t_{READY}$ | $\overline{RP}$ Low to Read mode, during Program or Erase | Max | 20 | | µs |
| $t_{PLPX}$ | $t_{RP}$ | $\overline{RP}$ Pulse Width | Min | 500 | | ns |
| $t_{PHEL}$, $t_{PHGL}$ (1) | $t_{RH}$ | $\overline{RP}$ High to Write Enable Low, Chip Enable Low, Output Enable Low | Min | 50 | | ns |
| | $t_{RPD}$ | $\overline{RP}$ Low to Standby Mode. | Min | 20 | | ns |
| $t_{RHEL}$ $t_{RHGL}$ (1) | $t_{RB}$ | $\overline{RB}$ High to Write Enable Low, Chip Enable Low, Output Enable Low | Min | 0 | | ns |

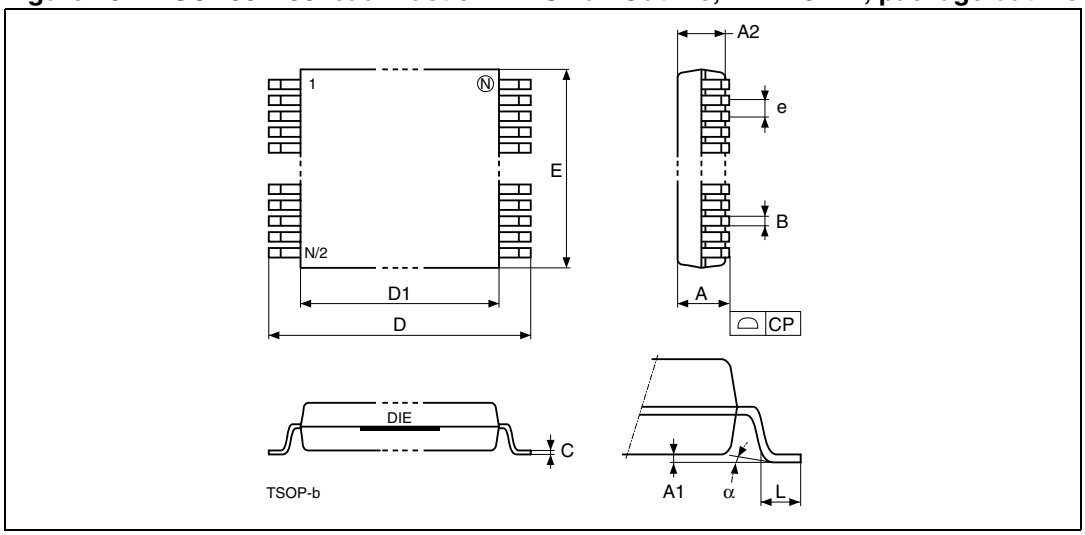
1. Sampled only, not 100% tested.

**Figure 15. Accelerated Program Timing waveforms**



# 9 Package mechanical

**Figure 16. TSOP56 – 56 lead Plastic Thin Small Outline, 14 x 20mm, package outline**

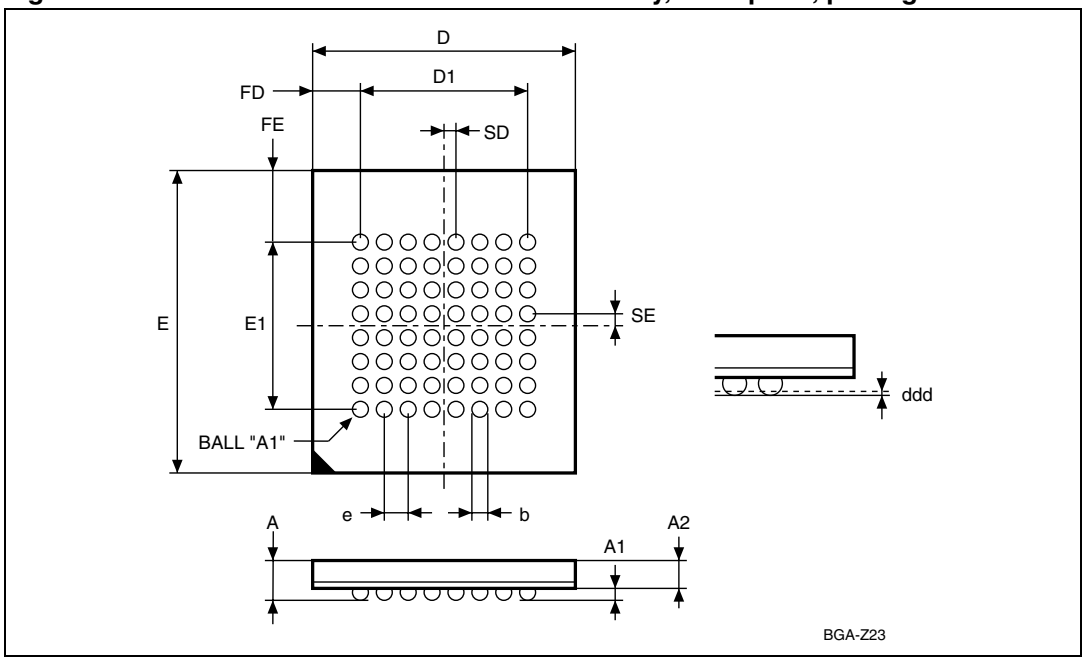



1. Drawing is not to scale.

**Table 25. TSOP56 – 56 lead Plastic Thin Small Outline, 14 x 20mm, package mechanical data**

| Symbol | millimeters | | | inches | | |
|---|---|---|---|---|---|---|
| | Typ | Min | Max | Typ | Min | Max |
| A | | | 1.200 | | | 0.0472 |
| A1 | 0.100 | 0.050 | 0.150 | 0.0039 | 0.0020 | 0.0059 |
| A2 | 1.000 | 0.950 | 1.050 | 0.0394 | 0.0374 | 0.0413 |
| B | 0.220 | 0.170 | 0.270 | 0.0087 | 0.0067 | 0.0106 |
| C | | 0.100 | 0.210 | | 0.0039 | 0.0083 |
| CP | | | 0.100 | | | 0.0039 |
| D | 20.000 | 19.800 | 20.200 | 0.7874 | 0.7795 | 0.7953 |
| D1 | 18.400 | 18.300 | 18.500 | 0.7244 | 0.7205 | 0.7283 |
| e | 0.500 | – | – | 0.0197 | – | – |
| E | 14.000 | 13.900 | 14.100 | 0.5512 | 0.5472 | 0.5551 |
| L | 0.600 | 0.500 | 0.700 | 0.0236 | 0.0197 | 0.0276 |
| α | 3° | 0 | 5° | 3° | 0 | 5° |
| N | 56 | | | 56 | | |

**Figure 17. TBGA64 10x13mm - 8x8 active ball array, 1mm pitch, package outline**




1. Drawing is not to scale.

**Table 26. TBGA64 10x13mm - 8x8 active ball array, 1mm pitch, package mechanical data**

| Symbol | millimeters | | | inches | | |
|---|---|---|---|---|---|---|
| | Typ | Min | Max | Typ | Min | Max |
| A | | | 1.200 | | | 0.0472 |
| A1 | 0.300 | 0.200 | 0.350 | 0.0118 | 0.0079 | 0.0138 |
| A2 | 0.800 | | | 0.0315 | | |
| b | | 0.350 | 0.500 | | 0.0138 | 0.0197 |
| D | 10.000 | 9.900 | 10.100 | 0.3937 | 0.3898 | 0.3976 |
| D1 | 7.000 | – | – | 0.2756 | – | – |
| ddd | | | 0.100 | | | 0.0039 |
| e | 1.000 | – | – | 0.0394 | – | – |
| E | 13.000 | 12.900 | 13.100 | 0.5118 | 0.5079 | 0.5157 |
| E1 | 7.000 | – | – | 0.2756 | – | – |
| FD | 1.500 | – | – | 0.0591 | – | – |
| FE | 3.000 | – | – | 0.1181 | – | – |
| SD | 0.500 | – | – | 0.0197 | – | – |
| SE | 0.500 | – | – | 0.0197 | – | – |

# 10 Part numbering

**Table 27. Ordering information scheme**

Example: M29W128FH 70 N 6 F

**Device type**

M29

**Operating voltage**

W = $V_{CC}$ = 2.7 to 3.6V

**Device function**

128FH = 128 Mbit (x8/x16), Page, Uniform Block, Flash Memory, Highest Block Protected by $V_{PP}/\overline{WP}$

128FL = 128 Mbit (x8/x16), Page, Uniform Block, Flash Memory, Lowest Block Protected by $V_{PP}/\overline{WP}$

**Speed**

60 = 60ns

70 = 70ns

**Package**

N = TSOP56: 14 x 20 mm

ZA = TBGA64: 10 x13mm, 1mm pitch

**Temperature range**

6 = –40 to 85 °C

**Option**

E = ECOPACK Package, Standard Packing

F = ECOPACK Package, Tape & Reel Packing

*Note: This product is also available with the Extended Memory Block factory locked. For further details and ordering information contact your nearest Numonyx sales office.*

Devices are shipped from the factory with the memory content bits erased to '1'.

For a list of available options (Speed, Package, etc.) or for further information on any aspect of this device, please contact your nearest Numonyx Sales Office.

# Appendix A Block addresses and Read/Modify Protection Groups

**Table 28. Block Addresses and Protection Groups**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 0 | 64/32 | Protection Group | 000000h-00FFFFh | 000000h-007FFFh |
| 1 | 64/32 | Protection Group | 010000h-01FFFFh | 008000h-00FFFFh |
| 2 | 64/32 | Protection Group | 020000h-02FFFFh | 010000h-017FFFh |
| 3 | 64/32 | Protection Group | 030000h-03FFFFh | 018000h-01FFFFh |
| 4 | 64/32 | Protection Group | 040000h-04FFFFh | 020000h-027FFFh |
| 5 | 64/32 | | 050000h-05FFFFh | 028000h-02FFFFh |
| 6 | 64/32 | | 060000h-06FFFFh | 030000h-037FFFh |
| 7 | 64/32 | | 070000h-07FFFFh | 038000h-03FFFFh |
| 8 | 64/32 | Protection Group | 080000h-08FFFFh | 040000h-047FFFh |
| 9 | 64/32 | | 090000h-09FFFFh | 048000h-04FFFFh |
| 10 | 64/32 | | 0A0000h-0AFFFFh | 050000h-057FFFh |
| 11 | 64/32 | | 0B0000h-0BFFFFh | 058000h-05FFFFh |
| 12 | 64/32 | Protection Group | 0C0000h-0CFFFFh | 060000h-067FFFh |
| 13 | 64/32 | | 0D0000h-0DFFFFh | 068000h-06FFFFh |
| 14 | 64/32 | | 0E0000h-0EFFFFh | 070000h-077FFFh |
| 15 | 64/32 | | 0F0000h-0FFFFFh | 078000h-07FFFFh |
| 16 | 64/32 | Protection Group | 100000h-10FFFFh | 080000h-087FFFh |
| 17 | 64/32 | | 110000h-11FFFFh | 088000h-08FFFFh |
| 18 | 64/32 | | 120000h-12FFFFh | 090000h-097FFFh |
| 19 | 64/32 | | 130000h-13FFFFh | 098000h-09FFFFh |
| 20 | 64/32 | Protection Group | 140000h-14FFFFh | 0A0000h-0A7FFFh |
| 21 | 64/32 | | 150000h-15FFFFh | 0A8000h-0AFFFFh |
| 22 | 64/32 | | 160000h-16FFFFh | 0B0000h-0B7FFFh |
| 23 | 64/32 | | 170000h-17FFFFh | 0B8000h-0BFFFFh |
| 24 | 64/32 | Protection Group | 180000h-18FFFFh | 0C0000h-0C7FFFh |
| 25 | 64/32 | | 190000h-19FFFFh | 0C8000h-0CFFFFh |
| 26 | 64/32 | | 1A0000h-1AFFFFh | 0D0000h-0D7FFFh |
| 27 | 64/32 | | 1B0000h-1BFFFFh | 0D8000h-0DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 28 | 64/32 | Protection Group | 1C0000h-1CFFFFh | 0E0000h-0E7FFFh |
| 29 | 64/32 | | 1D0000h-1DFFFFh | 0E8000h-0EFFFFh |
| 30 | 64/32 | | 1E0000h-1EFFFFh | 0F0000h-0F7FFFh |
| 31 | 64/32 | | 1F0000h-1FFFFFh | 0F8000h-0FFFFFh |
| 32 | 64/32 | Protection Group | 200000h-20FFFFh | 100000h-107FFFh |
| 33 | 64/32 | | 210000h-21FFFFh | 108000h–10FFFFh |
| 34 | 64/32 | | 220000h-22FFFFh | 110000h–117FFFh |
| 35 | 64/32 | | 230000h-23FFFFh | 118000h–11FFFFh |
| 36 | 64/32 | Protection Group | 240000h-24FFFFh | 120000h–127FFFh |
| 37 | 64/32 | | 250000h-25FFFFh | 128000h–12FFFFh |
| 38 | 64/32 | | 260000h-26FFFFh | 130000h–137FFFh |
| 39 | 64/32 | | 270000h-27FFFFh | 138000h–13FFFFh |
| 40 | 64/32 | Protection Group | 280000h-28FFFFh | 140000h–147FFFh |
| 41 | 64/32 | | 290000h-29FFFFh | 148000h–14FFFFh |
| 42 | 64/32 | | 2A0000h-2AFFFFh | 150000h–157FFFh |
| 43 | 64/32 | | 2B0000h-2BFFFFh | 158000h–15FFFFh |
| 44 | 64/32 | Protection Group | 2C0000h-2CFFFFh | 160000h–167FFFh |
| 45 | 64/32 | | 2D0000h-2DFFFFh | 168000h–16FFFFh |
| 46 | 64/32 | | 2E0000h-2EFFFFh | 170000h–177FFFh |
| 47 | 64/32 | | 2F0000h-2FFFFFh | 178000h–17FFFFh |
| 48 | 64/32 | Protection Group | 300000h-30FFFFh | 180000h–187FFFh |
| 49 | 64/32 | | 310000h-31FFFFh | 188000h–18FFFFh |
| 50 | 64/32 | | 320000h-32FFFFh | 190000h–197FFFh |
| 51 | 64/32 | | 330000h-33FFFFh | 198000h–19FFFFh |
| 52 | 64/32 | Protection Group | 340000h-34FFFFh | 1A0000h–1A7FFFh |
| 53 | 64/32 | | 350000h-35FFFFh | 1A8000h–1AFFFFh |
| 54 | 64/32 | | 360000h-36FFFFh | 1B0000h–1B7FFFh |
| 55 | 64/32 | | 370000h-37FFFFh | 1B8000h–1BFFFFh |
| 56 | 64/32 | Protection Group | 380000h-38FFFFh | 1C0000h–1C7FFFh |
| 57 | 64/32 | | 390000h-39FFFFh | 1C8000h–1CFFFFh |
| 58 | 64/32 | | 3A0000h-3AFFFFh | 1D0000h–1D7FFFh |
| 59 | 64/32 | | 3B0000h-3BFFFFh | 1D8000h–1DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 60 | 64/32 | Protection Group | 3C0000h-3CFFFFh | 1E0000h–1E7FFFh |
| 61 | 64/32 | | 3D0000h-3DFFFFh | 1E8000h–1EFFFFh |
| 62 | 64/32 | | 3E0000h-3EFFFFh | 1F0000h–1F7FFFh |
| 63 | 64/32 | | 3F0000h-3FFFFFh | 1F8000h–1FFFFFh |
| 64 | 64/32 | Protection Group | 400000h-40FFFFh | 200000h–207FFFh |
| 65 | 64/32 | | 410000h-41FFFFh | 208000h–20FFFFh |
| 66 | 64/32 | | 420000h-42FFFFh | 210000h–217FFFh |
| 67 | 64/32 | | 430000h-43FFFFh | 218000h–21FFFFh |
| 68 | 64/32 | Protection Group | 440000h-44FFFFh | 220000h–227FFFh |
| 69 | 64/32 | | 450000h-45FFFFh | 228000h–22FFFFh |
| 70 | 64/32 | | 460000h-46FFFFh | 230000h–237FFFh |
| 71 | 64/32 | | 470000h-47FFFFh | 238000h–23FFFFh |
| 72 | 64/32 | Protection Group | 480000h-48FFFFh | 240000h–247FFFh |
| 73 | 64/32 | | 490000h-49FFFFh | 248000h–24FFFFh |
| 74 | 64/32 | | 4A0000h-4AFFFFh | 250000h–257FFFh |
| 75 | 64/32 | | 4B0000h-4BFFFFh | 258000h–25FFFFh |
| 76 | 64/32 | Protection Group | 4C0000h-4CFFFFh | 260000h–267FFFh |
| 77 | 64/32 | | 4D0000h-4DFFFFh | 268000h–26FFFFh |
| 78 | 64/32 | | 4E0000h-4EFFFFh | 270000h–277FFFh |
| 79 | 64/32 | | 4F0000h-4FFFFFh | 278000h–27FFFFh |
| 80 | 64/32 | Protection Group | 500000h-50FFFFh | 280000h–287FFFh |
| 81 | 64/32 | | 510000h-51FFFFh | 288000h–28FFFFh |
| 82 | 64/32 | | 520000h-52FFFFh | 290000h–297FFFh |
| 83 | 64/32 | | 530000h-53FFFFh | 298000h–29FFFFh |
| 84 | 64/32 | Protection Group | 540000h-54FFFFh | 2A0000h–2A7FFFh |
| 85 | 64/32 | | 550000h-55FFFFh | 2A8000h–2AFFFFh |
| 86 | 64/32 | | 560000h-56FFFFh | 2B0000h–2B7FFFh |
| 87 | 64/32 | | 570000h-57FFFFh | 2B8000h–2BFFFFh |
| 88 | 64/32 | Protection Group | 580000h-58FFFFh | 2C0000h–2C7FFFh |
| 89 | 64/32 | | 590000h-59FFFFh | 2C8000h–2CFFFFh |
| 90 | 64/32 | | 5A0000h-5AFFFFh | 2D0000h–2D7FFFh |
| 91 | 64/32 | | 5B0000h-5BFFFFh | 2D8000h–2DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 92 | 64/32 | Protection Group | 5C0000h-5CFFFFh | 2E0000h–2E7FFFh |
| 93 | 64/32 | | 5D0000h-5DFFFFh | 2E8000h–2EFFFFh |
| 94 | 64/32 | | 5E0000h-5EFFFFh | 2F0000h–2F7FFFh |
| 95 | 64/32 | | 5F0000h-5FFFFFh | 2F8000h–2FFFFFh |
| 96 | 64/32 | Protection Group | 600000h-60FFFFh | 300000h–307FFFh |
| 97 | 64/32 | | 610000h-61FFFFh | 308000h–30FFFFh |
| 98 | 64/32 | | 620000h-62FFFFh | 310000h–317FFFh |
| 99 | 64/32 | | 630000h-63FFFFh | 318000h–31FFFFh |
| 100 | 64/32 | Protection Group | 640000h-64FFFFh | 320000h–327FFFh |
| 101 | 64/32 | | 650000h-65FFFFh | 328000h–32FFFFh |
| 102 | 64/32 | | 660000h-66FFFFh | 330000h–337FFFh |
| 103 | 64/32 | | 670000h-67FFFFh | 338000h–33FFFFh |
| 104 | 64/32 | Protection Group | 680000h-68FFFFh | 340000h–347FFFh |
| 105 | 64/32 | | 690000h-69FFFFh | 348000h–34FFFFh |
| 106 | 64/32 | | 6A0000h-6AFFFFh | 350000h–357FFFh |
| 107 | 64/32 | | 6B0000h-6BFFFFh | 358000h–35FFFFh |
| 108 | 64/32 | Protection Group | 6C0000h-6CFFFFh | 360000h–367FFFh |
| 109 | 64/32 | | 6D0000h-6DFFFFh | 368000h–36FFFFh |
| 110 | 64/32 | | 6E0000h-6EFFFFh | 370000h–377FFFh |
| 111 | 64/32 | | 6F0000h-6FFFFFh | 378000h–37FFFFh |
| 112 | 64/32 | Protection Group | 700000h–70FFFFh | 380000h–387FFFh |
| 113 | 64/32 | | 710000h–71FFFFh | 388000h–38FFFFh |
| 114 | 64/32 | | 720000h–72FFFFh | 390000h–397FFFh |
| 115 | 64/32 | | 730000h–73FFFFh | 398000h–39FFFFh |
| 116 | 64/32 | Protection Group | 740000h–74FFFFh | 3A0000h–3A7FFFh |
| 117 | 64/32 | | 750000h–75FFFFh | 3A8000h–3AFFFFh |
| 118 | 64/32 | | 760000h–76FFFFh | 3B0000h–3B7FFFh |
| 119 | 64/32 | | 770000h–77FFFFh | 3B8000h–3BFFFFh |
| 120 | 64/32 | Protection Group | 780000h–78FFFFh | 3C0000h–3C7FFFh |
| 121 | 64/32 | | 790000h–79FFFFh | 3C8000h–3CFFFFh |
| 122 | 64/32 | | 7A0000h–7AFFFFh | 3D0000h–3D7FFFh |
| 123 | 64/32 | | 7B0000h–7BFFFFh | 3D8000h–3DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 124 | 64/32 | Protection Group | 7C0000h–7CFFFFh | 3E0000h–3E7FFFh |
| 125 | 64/32 | | 7D0000h–7DFFFFh | 3E8000h–3EFFFFh |
| 126 | 64/32 | | 7E0000h–7EFFFFh | 3F0000h–3F7FFFh |
| 127 | 64/32 | | 7F0000h-7FFFFFh | 3F8000h-3FFFFFh |
| 128 | 64/32 | Protection Group | 800000h–80FFFFh | 400000h–407FFFh |
| 129 | 64/32 | | 810000h–81FFFFh | 408000h–40FFFFh |
| 130 | 64/32 | | 820000h–82FFFFh | 410000h–417FFFh |
| 131 | 64/32 | | 830000h–83FFFFh | 418000h–41FFFFh |
| 132 | 64/32 | Protection Group | 840000h–84FFFFh | 420000h–427FFFh |
| 133 | 64/32 | | 850000h–85FFFFh | 428000h–42FFFFh |
| 134 | 64/32 | | 860000h–86FFFFh | 430000h–437FFFh |
| 135 | 64/32 | | 870000h–87FFFFh | 438000h–43FFFFh |
| 136 | 64/32 | Protection Group | 880000h–88FFFFh | 440000h–447FFFh |
| 137 | 64/32 | | 890000h–89FFFFh | 448000h–44FFFFh |
| 138 | 64/32 | | 8A0000h–8AFFFFh | 450000h–457FFFh |
| 139 | 64/32 | | 8B0000h–8BFFFFh | 458000h–45FFFFh |
| 140 | 64/32 | Protection Group | 8C0000h–8CFFFFh | 460000h–467FFFh |
| 141 | 64/32 | | 8D0000h–8DFFFFh | 468000h–46FFFFh |
| 142 | 64/32 | | 8E0000h–8EFFFFh | 470000h–477FFFh |
| 143 | 64/32 | | 8F0000h-8FFFFFh | 478000h–47FFFFh |
| 144 | 64/32 | Protection Group | 900000h-90FFFFh | 480000h–487FFFh |
| 145 | 64/32 | | 910000h–91FFFFh | 488000h–48FFFFh |
| 146 | 64/32 | | 920000h–92FFFFh | 490000h–497FFFh |
| 147 | 64/32 | | 930000h–93FFFFh | 498000h–49FFFFh |
| 148 | 64/32 | Protection Group | 940000h–94FFFFh | 4A0000h–4A7FFFh |
| 149 | 64/32 | | 950000h–95FFFFh | 4A8000h–4AFFFFh |
| 150 | 64/32 | | 960000h–96FFFFh | 4B0000h–4B7FFFh |
| 151 | 64/32 | | 970000h–97FFFFh | 4B8000h–4BFFFFh |
| 152 | 64/32 | Protection Group | 980000h–98FFFFh | 4C0000h–4C7FFFh |
| 153 | 64/32 | | 990000h–99FFFFh | 4C8000h–4CFFFFh |
| 154 | 64/32 | | 9A0000h–9AFFFFh | 4D0000h–4D7FFFh |
| 155 | 64/32 | | 9B0000h–9BFFFFh | 4D8000h–4DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 156 | 64/32 | Protection Group | 9C0000h–9CFFFFh | 4E0000h–4E7FFFh |
| 157 | 64/32 | | 9D0000h–9DFFFFh | 4E8000h–4EFFFFh |
| 158 | 64/32 | | 9E0000h–9EFFFFh | 4F0000h–4F7FFFh |
| 159 | 64/32 | | 9F0000h–9FFFFFh | 4F8000h-4FFFFFh |
| 160 | 64/32 | Protection Group | A00000h–A0FFFFh | 500000h–507FFFh |
| 161 | 64/32 | | A10000h–A1FFFFh | 508000h–50FFFFh |
| 162 | 64/32 | | A20000h–A2FFFFh | 510000h–517FFFh |
| 163 | 64/32 | | A30000h–A3FFFFh | 518000h–51FFFFh |
| 164 | 64/32 | Protection Group | A40000h–A4FFFFh | 520000h–527FFFh |
| 165 | 64/32 | | A50000h–A5FFFFh | 528000h–52FFFFh |
| 166 | 64/32 | | A60000h–A6FFFFh | 530000h–537FFFh |
| 167 | 64/32 | | A70000h–A7FFFFh | 538000h–53FFFFh |
| 168 | 64/32 | Protection Group | A80000h–A8FFFFh | 540000h–547FFFh |
| 169 | 64/32 | | A90000h–A9FFFFh | 548000h–54FFFFh |
| 170 | 64/32 | | AA0000h–AAFFFFh | 550000h–557FFFh |
| 171 | 64/32 | | AB0000h–ABFFFFh | 558000h–55FFFFh |
| 172 | 64/32 | Protection Group | AC0000h–ACFFFFh | 560000h–567FFFh |
| 173 | 64/32 | | AD0000h–ADFFFFh | 568000h–56FFFFh |
| 174 | 64/32 | | AE0000h–AEFFFFh | 570000h–577FFFh |
| 175 | 64/32 | | AF0000h-AFFFFFh | 578000h–57FFFFh |
| 176 | 64/32 | Protection Group | B00000h–B0FFFFh | 580000h–587FFFh |
| 177 | 64/32 | | B10000h–B1FFFFh | 588000h–58FFFFh |
| 178 | 64/32 | | B20000h–B2FFFFh | 590000h–597FFFh |
| 179 | 64/32 | | B30000h-B3FFFFh | 598000h–59FFFFh |
| 180 | 64/32 | Protection Group | B40000h–B4FFFFh | 5A0000h–5A7FFFh |
| 181 | 64/32 | | B50000h–B5FFFFh | 5A8000h–5AFFFFh |
| 182 | 64/32 | | B60000h–B6FFFFh | 5B0000h–5B7FFFh |
| 183 | 64/32 | | B70000h-B7FFFFh | 5B8000h–5BFFFFh |
| 184 | 64/32 | Protection Group | B80000h–B8FFFFh | 5C0000h–5C7FFFh |
| 185 | 64/32 | | B90000h–B9FFFFh | 5C8000h–5CFFFFh |
| 186 | 64/32 | | BA0000h–BAFFFFh | 5D0000h–5D7FFFh |
| 187 | 64/32 | | BB0000h–BBFFFFh | 5D8000h–5DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 188 | 64/32 | Protection Group | BC0000h–BCFFFFh | 5E0000h–5E7FFFh |
| 189 | 64/32 | | BD0000h–BDFFFFh | 5E8000h–5EFFFFh |
| 190 | 64/32 | | BE0000h–BEFFFFh | 5F0000h–5F7FFFh |
| 191 | 64/32 | | BF0000h–BFFFFFh | 5F8000h-5FFFFFh |
| 192 | 64/32 | Protection Group | C00000h–C0FFFFh | 600000h–607FFFh |
| 193 | 64/32 | | C10000h–C1FFFFh | 608000h–60FFFFh |
| 194 | 64/32 | | C20000h–C2FFFFh | 610000h–617FFFh |
| 195 | 64/32 | | C30000h–C3FFFFh | 618000h–61FFFFh |
| 196 | 64/32 | Protection Group | C40000h–C4FFFFh | 620000h–627FFFh |
| 197 | 64/32 | | C50000h–C5FFFFh | 628000h–62FFFFh |
| 198 | 64/32 | | C60000h–C6FFFFh | 630000h–637FFFh |
| 199 | 64/32 | | C70000h-C7FFFFh | 638000h–63FFFFh |
| 200 | 64/32 | Protection Group | C80000h–C8FFFFh | 640000h–647FFFh |
| 201 | 64/32 | | C90000h–C9FFFFh | 648000h–64FFFFh |
| 202 | 64/32 | | CA0000h–CAFFFFh | 650000h–657FFFh |
| 203 | 64/32 | | CB0000h–CBFFFFh | 658000h–65FFFFh |
| 204 | 64/32 | Protection Group | CC0000h–CCFFFFh | 660000h–667FFFh |
| 205 | 64/32 | | CD0000h–CDFFFFh | 668000h–66FFFFh |
| 206 | 64/32 | | CE0000h–CEFFFFh | 670000h–677FFFh |
| 207 | 64/32 | | CF0000h-CFFFFFh | 678000h–67FFFFh |
| 208 | 64/32 | Protection Group | D00000h–D0FFFFh | 680000h–687FFFh |
| 209 | 64/32 | | D10000h–D1FFFFh | 688000h–68FFFFh |
| 210 | 64/32 | | D20000h–D2FFFFh | 690000h–697FFFh |
| 211 | 64/32 | | D30000h–D3FFFFh | 698000h–69FFFFh |
| 212 | 64/32 | Protection Group | D40000h–D4FFFFh | 6A0000h–6A7FFFh |
| 213 | 64/32 | | D50000h–D5FFFFh | 6A8000h–6AFFFFh |
| 214 | 64/32 | | D60000h–D6FFFFh | 6B0000h–6B7FFFh |
| 215 | 64/32 | | D70000h-D7FFFFh | 6B8000h–6BFFFFh |
| 216 | 64/32 | Protection Group | D80000h-D8FFFFh | 6C0000h–6C7FFFh |
| 217 | 64/32 | | D90000h-D9FFFFh | 6C8000h–6CFFFFh |
| 218 | 64/32 | | DA0000h-DAFFFFh | 6D0000h–6D7FFFh |
| 219 | 64/32 | | DB0000h-DBFFFFh | 6D8000h–6DFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 220 | 64/32 | Protection Group | DC0000h-DCFFFFh | 6E0000h–6E7FFFh |
| 221 | 64/32 | | DD0000h-DDFFFFh | 6E8000h–6EFFFFh |
| 222 | 64/32 | | DE0000h-DEFFFFh | 6F0000h–6F7FFFh |
| 223 | 64/32 | | DF0000h-DFFFFFh | 6F8000h-6FFFFFh |
| 224 | 64/32 | Protection Group | E00000h-E0FFFFh | 700000h–707FFFh |
| 225 | 64/32 | | E10000h-E1FFFFh | 708000h–70FFFFh |
| 226 | 64/32 | | E20000h-E2FFFFh | 710000h–717FFFh |
| 227 | 64/32 | | E30000h-E3FFFFh | 718000h–71FFFFh |
| 228 | 64/32 | Protection Group | E40000h-E4FFFFh | 720000h–727FFFh |
| 229 | 64/32 | | E50000h-E5FFFFh | 728000h–72FFFFh |
| 230 | 64/32 | | E60000h-E6FFFFh | 730000h–737FFFh |
| 231 | 64/32 | | E70000h-E7FFFFh | 738000h–73FFFFh |
| 232 | 64/32 | Protection Group | E80000h-E8FFFFh | 740000h–747FFFh |
| 233 | 64/32 | | E90000h-E9FFFFh | 748000h–74FFFFh |
| 234 | 64/32 | | EA0000h-EAFFFFh | 750000h–757FFFh |
| 235 | 64/32 | | EB0000h-EBFFFFh | 758000h–75FFFFh |
| 236 | 64/32 | Protection Group | EC0000h-ECFFFFh | 760000h–767FFFh |
| 237 | 64/32 | | ED0000h-EDFFFFh | 768000h–76FFFFh |
| 238 | 64/32 | | EE0000h-EEFFFFh | 770000h–777FFFh |
| 239 | 64/32 | | EF0000h-EFFFFFh | 778000h–77FFFFh |
| 240 | 64/32 | Protection Group | F00000h-F0FFFFh | 780000h–787FFFh |
| 241 | 64/32 | | F10000h-F1FFFFh | 788000h–78FFFFh |
| 242 | 64/32 | | F20000h-F2FFFFh | 790000h–797FFFh |
| 243 | 64/32 | | F30000h-F3FFFFh | 798000h–79FFFFh |
| 244 | 64/32 | Protection Group | F40000h-F4FFFFh | 7A0000h–7A7FFFh |
| 245 | 64/32 | | F50000h-F5FFFFh | 7A8000h–7AFFFFh |
| 246 | 64/32 | | F60000h-F6FFFFh | 7B0000h–7B7FFFh |
| 247 | 64/32 | | F70000h-F7FFFFh | 7B8000h–7BFFFFh |
| 248 | 64/32 | Protection Group | F80000h-F8FFFFh | 7C0000h–7C7FFFh |
| 249 | 64/32 | | F90000h-F9FFFFh | 7C8000h–7CFFFFh |
| 250 | 64/32 | | FA0000h-FAFFFFh | 7D0000h–7D7FFFh |
| 251 | 64/32 | | FB0000h-FBFFFFh | 7D8000h–7DFFFFh |
| 252 | 64/32 | Protection Group | FC0000h-FCFFFFh | 7E0000h–7E7FFFh |
| 253 | 64/32 | Protection Group | FD0000h-FDFFFFh | 7E8000h–7EFFFFh |

**Table 28. Block Addresses and Protection Groups (continued)**

| Block | Size (KBytes/KWords) | Protection Block Group | (x8) | (x16) |
|---|---|---|---|---|
| 254 | 64/32 | Protection Group | FE0000h-FEFFFFh | 7F0000h–7F7FFFh |
| 255 | 64/32 | Protection Group | FF0000h-FFFFFFh | 7F8000h–7FFFFFh |

# Appendix B Common Flash Interface (CFI)

The Common Flash Interface is a JEDEC approved, standardized data structure that can be read from the Flash memory device. It allows a system software to query the device to determine various electrical and timing parameters, density information and functions supported by the memory. The system can interface easily with the device, enabling the software to upgrade itself when necessary.

When the Read CFI Query command is issued, the memory enters Read CFI Query mode and read operations output the CFI data. *Table 29*, *Table 30*, *Table 31*, *Table 32*, *Table 33* and *Table 34* show the addresses (A-1, A0-A10) used to retrieve the data. The CFI data structure also contains a security area where a 64 bit unique security number is written (see *Table 34: Security Code Area*). This area can be accessed only in Read mode by the final user. It is impossible to change the security number after it has been written by Numonyx.

**Table 29. Query Structure Overview(1)**

| Address | | Sub-section Name | Description |
|---|---|---|---|
| x16 | x8 | | |
| 10h | 20h | CFI Query Identification String | Command set ID and algorithm data offset |
| 1Bh | 36h | System Interface Information | Device timing & voltage information |
| 27h | 4Eh | Device Geometry Definition | Flash device layout |
| 40h | 80h | Primary Algorithm-specific Extended Query table | Additional information specific to the Primary Algorithm (optional) |
| 61h | C2h | Security Code Area | 64 bit unique device number |

1. Query data are always presented on the lowest order data outputs.

**Table 30. CFI Query Identification String(1)**

| Address | | Data | Description | Value |
|---|---|---|---|---|
| x16 | x8 | | | |
| 10h | 20h | 0051h | Query Unique ASCII String "QRY" | “Q” |
| 11h | 22h | 0052h | | "R" |
| 12h | 24h | 0059h | | "Y" |
| 13h | 26h | 0002h | Primary Algorithm Command Set and Control Interface ID code 16 bit ID code defining a specific algorithm | AMD Compatible |
| 14h | 28h | 0000h | | |
| 15h | 2Ah | 0040h | Address for Primary Algorithm extended Query table (see *Table 33*) | P = 40h |
| 16h | 2Ch | 0000h | | |
| 17h | 2Eh | 0000h | Alternate Vendor Command Set and Control Interface ID Code second vendor - specified algorithm supported | NA |
| 18h | 30h | 0000h | | |
| 19h | 32h | 0000h | Address for Alternate Algorithm extended Query table | NA |
| 1Ah | 34h | 0000h | | |

1. Query data are always presented on the lowest order data outputs (DQ7-DQ0) only. DQ8-DQ15 are ‘0’.

**Table 31. CFI Query System Interface Information(1)**

| Address x16 | Address x8 | Data | Description | Value |
|---|---|---|---|---|
| 1Bh | 36h | 0027h | $V_{CC}$ Logic Supply Minimum Program/Erase voltage<br>bit 7 to 4BCD value in volts<br>bit 3 to 0BCD value in 100mV | 3.0V |
| 1Ch | 38h | 0036h | $V_{CC}$ Logic Supply Maximum Program/Erase voltage<br>bit 7 to 4BCD value in volts<br>bit 3 to 0BCD value in 100mV | 3.6V |
| 1Dh | 3Ah | 00B5h | $V_{PP}$ [Programming] Supply Minimum Program/Erase voltage<br>bit 7 to 4HEX value in volts<br>bit 3 to 0BCD value in 100mV | 11.5V |
| 1Eh | 3Ch | 00C5h | $V_{PP}$ [Programming] Supply Maximum Program/Erase voltage<br>bit 7 to 4HEX value in volts<br>bit 3 to 0BCD value in 10mV | 12.5V |
| 1Fh | 3Eh | 0004h | Typical timeout per single Byte/Word program = $2^n$ µs | 16µs |
| 20h | 40h | 0000h | Typical timeout for minimum size write buffer program = $2^n$ µs | NA |
| 21h | 42h | 0009h | Typical timeout per individual block erase = $2^n$ ms | 512ms |
| 22h | 44h | 0000h | Typical timeout for full Chip Erase = $2^n$ ms | NA |
| 23h | 46h | 0005h | Maximum timeout for Byte/Word program = $2^n$ times typical | 512µs |
| 24h | 48h | 0000h | Maximum timeout for write buffer program = $2^n$ times typical | NA |
| 25h | 4Ah | 0004h | Maximum timeout per individual block erase = $2^n$ times typical | 8s |
| 26h | 4Ch | 0000h | Maximum timeout for Chip Erase = $2^n$ times typical | NA |

1. The values given in the above table are valid for both packages.

**Table 32. Device Geometry Definition**

| Address x16 | Address x8 | Data | Description | Value |
|---|---|---|---|---|
| 27h | 4Eh | 0018h | Device Size = $2^n$ in number of Bytes | 16 MBytes |
| 28h<br>29h | 50h<br>52h | 0002h<br>0000h | Flash Device Interface Code description | x8, x16 Async. |
| 2Ah<br>2Bh | 54h<br>56h | 0006h<br>0000h | Maximum number of Bytes in Multiple-Byte program or Page= $2^n$ | 64 |
| 2Ch | 58h | 0001h | Number of Erase Block Regions. It specifies the number of regions containing contiguous Erase Blocks of the same size. | 1 |
| 2Dh<br>2Eh | 5Ah<br>5Ch | 00FFh<br>0000h | Erase Block Region 1 Information<br>Number of Erase Blocks of identical size = 00FFh+1 | 256 |
| 2Fh<br>30h | 5Eh<br>60h | 0000h<br>0001h | Erase Block Region 1 Information<br>Block size in Region 1 = 0100h * 256 Byte | 64 KBytes |

**Table 32. Device Geometry Definition (continued)**

| Address | | Data | Description | Value |
|---|---|---|---|---|
| x16 | x8 | | | |
| 31h<br>32h<br>33h<br>34h | 62h<br>64h<br>66h<br>68h | 0000h<br>0000h<br>0000h<br>0000h | Erase Block Region 2 Information | 0 |
| 35h<br>36h<br>37h<br>38h | 6Ah<br>6Ch<br>6Eh<br>70h | 0000h<br>0000h<br>0000h<br>0000h | Erase Block Region 3 information | 0 |
| 39h<br>3Ah<br>3Bh<br>3Ch | 72h<br>74h<br>76h<br>78h | 0000h<br>0000h<br>0000h<br>0000h | Erase Block Region 4 information | 0 |

**Table 33. Primary Algorithm-Specific Extended Query Table (1)**

| Address | | Data | Description | Value |
|---|---|---|---|---|
| x16 | x8 | | | |
| 40h | 80h | 0050h | Primary Algorithm extended Query table unique ASCII string “PRI” | "P" |
| 41h | 82h | 0052h | | "R" |
| 42h | 84h | 0049h | | "I" |
| 43h | 86h | 0031h | Major version number, ASCII | "1" |
| 44h | 88h | 0033h | Minor version number, ASCII | "3" |
| 45h | 8Ah | 000Ch | Address Sensitive Unlock (bits 1 to 0)<br>00 = required, 01= not required<br>Silicon Revision Number (bits 7 to 2) | Yes |
| 46h | 8Ch | 0002h | Erase Suspend<br>00 = not supported, 01 = Read only, 02 = Read and Write | 2 |
| 47h | 8Eh | 0001h | Block Protection<br>00 = not supported, x = number of sectors in per group | 1 |
| 48h | 90h | 0001h | Temporary Block Unprotect<br>00 = not supported, 01 = supported | Yes |
| 49h | 92h | 0006h | Block Protect /Unprotect<br>06 = M29W128FH/M29W128FL | 6 |
| 4Ah | 94h | 0000h | Simultaneous Operations: Not Supported | NA |
| 4Bh | 96h | 0000h | Burst Mode, 00 = not supported, 01 = supported | No |
| 4Ch | 98h | 0002h | Page Mode, 00 = not supported, 02 = 8-Word page | 02 |
| 4Dh | 9Ah | 00B5h | $V_{PP}$ Supply Minimum Program/Erase voltage<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100mV | 11.5V |

**Table 33. Primary Algorithm-Specific Extended Query Table (continued)(1)**

| Address | | Data | Description | Value |
|---|---|---|---|---|
| x16 | x8 | | | |
| 4Eh | 9Ch | 00C5h | $V_{PP}$ Supply Maximum Program/Erase voltage<br>bit 7 to 4 HEX value in volts<br>bit 3 to 0 BCD value in 100mV | 12.5V |
| 4Fh | 9Eh | 0000h | Top/Bottom Boot Block Flag<br>00h = Uniform device | Uniform + $V_{PP}/\overline{WP}$ Protecting Highest or Lowest Block |
| 50h | A0h | 0001h | Program Suspend, 00 = not supported, 01 = supported | Yes |

1. The values given in the above table are valid for both packages.

**Table 34. Security Code Area**

| Address | | Data | Description |
|---|---|---|---|
| x16 | x8 | | |
| 61h | C3h, C2h | XXXX | 64 bit: unique device number |
| 62h | C5h, C4h | XXXX | |
| 63h | C7h, C6h | XXXX | |
| 64h | C9h, C8h | XXXX | |

# Appendix C Extended Memory Block

The M29W128F has an extra block, the Extended Memory Block, that can be accessed using a dedicated command. This Extended Memory Block is 128 Words in x16 mode and 256 Bytes in x8 mode. It is used as a security block (to provide a permanent security identification number) or to store additional information.

The Extended Memory Block is either Factory Locked or Customer Lockable, its status is indicated by bit DQ7. This bit is permanently set to either ‘1’ or ‘0’ at the factory and cannot be changed. When set to ‘1’, it indicates that the device is factory locked and the Extended Memory Block is protected. When set to ‘0’, it indicates that the device is customer lockable and the Extended Memory Block is unprotected. Bit DQ7 being permanently locked to either ‘1’ or ‘0’ is another security feature which ensures that a customer lockable device cannot be used instead of a factory locked one.

Bit DQ7 is the most significant bit in the Extended Memory Block Verify Indicator and a specific procedure must be followed to read it. See Verify Extended Memory Block Protection Indicator in *Table 4: Block Protection, 8-bit mode* and *Table 7: Block Protection, 16-bit mode*, for details of how to read bit DQ7.

The Extended Memory Block can only be accessed when the device is in Extended Memory Block mode. For details of how the Extended Block mode is entered and exited, refer to the *Section 5.3.1: Enter Extended Memory Block command* and *Section 5.3.2: Exit Extended Block command*, and to *Table 13* and *Table 9*.

## C.1 Factory Locked Extended Memory Block

In devices where the Extended Memory Block is factory locked, the Security Identification Number is written to the Extended Memory Block address space (see *Table 35: Extended Memory Block Address and Data*) in the factory. The DQ7 bit is set to ‘1’ and the Extended Memory Block cannot be unprotected.

## C.2 Customer Lockable Extended Memory Block

A device where the Extended Memory Block is customer lockable is delivered with the DQ7 bit set to '0' and the Extended Memory Block unprotected. It is up to the customer to program and protect the Extended Memory Block but care must be taken because the protection of the Extended Memory Block is not reversible.

There are two ways of protecting the Extended Memory Block:

- Issue the Enter Extended Block command to place the device in Extended Block mode, then use the In-System Technique with $\overline{RP}$ either at $V_{IH}$ or at $V_{ID}$ (refer to *Appendix D: High Voltage Block Protection*, and to the corresponding flowcharts, *Figure 20* and *Figure 21*, for a detailed explanation of the technique).
- Issue the Enter Extended Block command to place the device in Extended Block mode, then use the Programmer Technique (refer to *Appendix D: High Voltage Block Protection*, and to the corresponding flowcharts, *Figure 18* and *Figure 19*, for a detailed explanation of the technique).

Once the Extended Memory Block is programmed and protected, the Exit Extended Block command must be issued to exit the Extended Memory Block mode and return the device to Read mode.

**Table 35. Extended Memory Block Address and Data**

| Address(1) | | Data | |
|---|---|---|---|
| **x8** | **x16** | **Factory Locked** | **Customer Lockable** |
| 000000h-0000FFh | 000000h-00007Fh | Security Identification Number | Determined by Customer |

1. See *Table 28: Block Addresses and Protection Groups*.

# Appendix D High Voltage Block Protection

The High Voltage Block Protection can be used to prevent any operation from modifying the data stored in the memory. The blocks are protected in groups, refer to *Appendix A: Block addresses and Read/Modify Protection Groups*, and *Table 28* for details of the Protection Groups. Once protected, Program and Erase operations within the protected group fail to change the data.

There are three techniques that can be used to control Block Protection, these are the Programmer technique, the In-System technique and Temporary Unprotection. Temporary Unprotection is controlled by the Reset/Block Temporary Unprotection pin, $\overline{RP}$; this is described in the Signal Descriptions section.

To protect the Extended Memory Block issue the Enter Extended Block command and then use either the Programmer or In-System technique. Once protected issue the Exit Extended Block command to return to read mode. The Extended Memory Block protection is irreversible, once protected the protection cannot be undone.

## D.1 Programmer technique

The Programmer technique uses high ($V_{ID}$) voltage levels on some of the bus pins. These cannot be achieved using a standard microprocessor bus, therefore the technique is recommended only for use in Programming Equipment.

To protect a group of blocks follow the flowchart in *Figure 18: Programmer equipment Group Protect flowchart*. To unprotect the whole chip it is necessary to protect all of the groups first, then all groups can be unprotected at the same time. To unprotect the chip follow *Figure 19: Programmer equipment Chip Unprotect flowchart*. *Table 36: Programmer technique Bus operations, 8-bit or 16-bit mode*, gives a summary of each operation.

The timing on these flowcharts is critical. Care should be taken to ensure that, where a pause is specified, it is followed as closely as possible. Do not abort the procedure before reaching the end. Chip Unprotect can take several seconds and a user message should be provided to show that the operation is progressing.

## D.2 In-System technique

The In-System technique requires a high voltage level on the Reset/Blocks Temporary Unprotect pin, $\overline{RP}$ (1). This can be achieved without violating the maximum ratings of the components on the microprocessor bus, therefore this technique is suitable for use after the memory has been fitted to the system.

To protect a group of blocks follow the flowchart in *Figure 20: In-System equipment Group Protect flowchart*. To unprotect the whole chip it is necessary to protect all of the groups first, then all the groups can be unprotected at the same time. To unprotect the chip follow *Figure 21: In-System equipment Chip Unprotect flowchart*.

The timing on these flowcharts is critical. Care should be taken to ensure that, where a pause is specified, it is followed as closely as possible. Do not allow the microprocessor to service interrupts that will upset the timing and do not abort the procedure before reaching the end. Chip Unprotect can take several seconds and a user message should be provided to show that the operation is progressing.

*Note: $\overline{RP}$ can be either at $V_{IH}$ or at $V_{ID}$ when using the In-System Technique to protect the Extended Memory Block.*

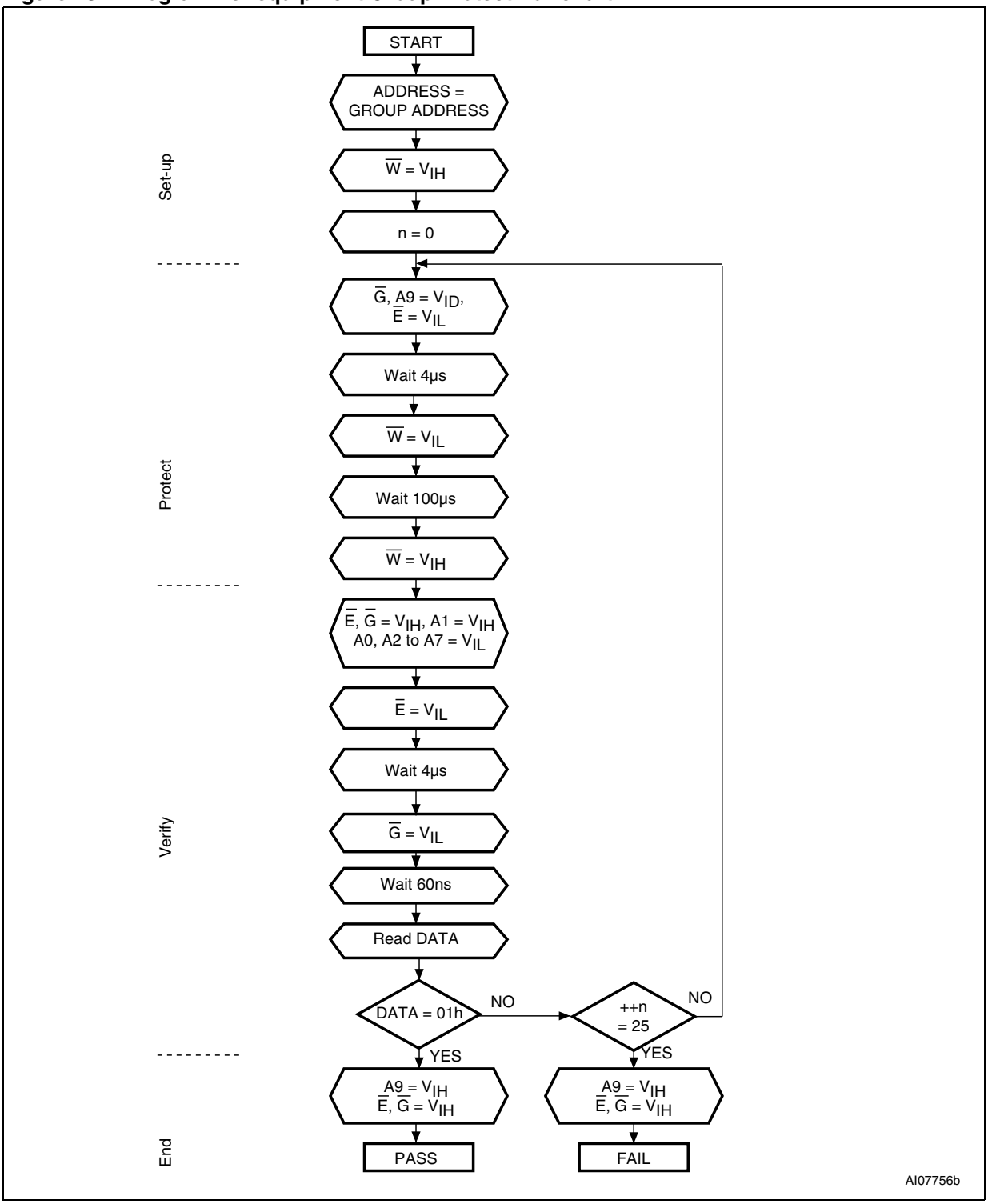
**Table 36. Programmer technique Bus operations, 8-bit or 16-bit mode**

| Operation | $\overline{E}$ | $\overline{G}$ | $\overline{W}$ | Address Inputs A0-A22 | Data Inputs/Outputs DQ15A–1, DQ14-DQ0 |
|---|---|---|---|---|---|
| Block (Group) Protect(1) | $V_{IL}$ | $V_{ID}$ | $V_{IL}$ Pulse | A9 = $V_{ID}$, A12-A22 Block Address<br>Others = X | X |
| Chip Unprotect | $V_{ID}$ | $V_{ID}$ | $V_{IL}$ Pulse | A6 = $V_{IH}$, A9 = $V_{ID}$, A12 = $V_{IH}$,<br>A15 = $V_{IH}$ Others = X | X |
| Block (Group) Protect Verify | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | A0, A2, A3, A6 = $V_{IL}$, A1 = $V_{IH}$<br>A9 = $V_{ID}$, A12-A22= Block Address<br>Others = X | Pass = xx01h<br>Retry = xx00h. |
| Block (Group) Unprotect Verify | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | A0, A2, A3 = $V_{IL}$<br>A1, A6 = $V_{IH}$<br>A9 = $V_{ID}$, A12-A22 = Block Address<br>Others = X | Pass = xx00h<br>Retry = xx01h. |

1. Block Protection Groups are shown in *Appendix D*, *Table 28*.

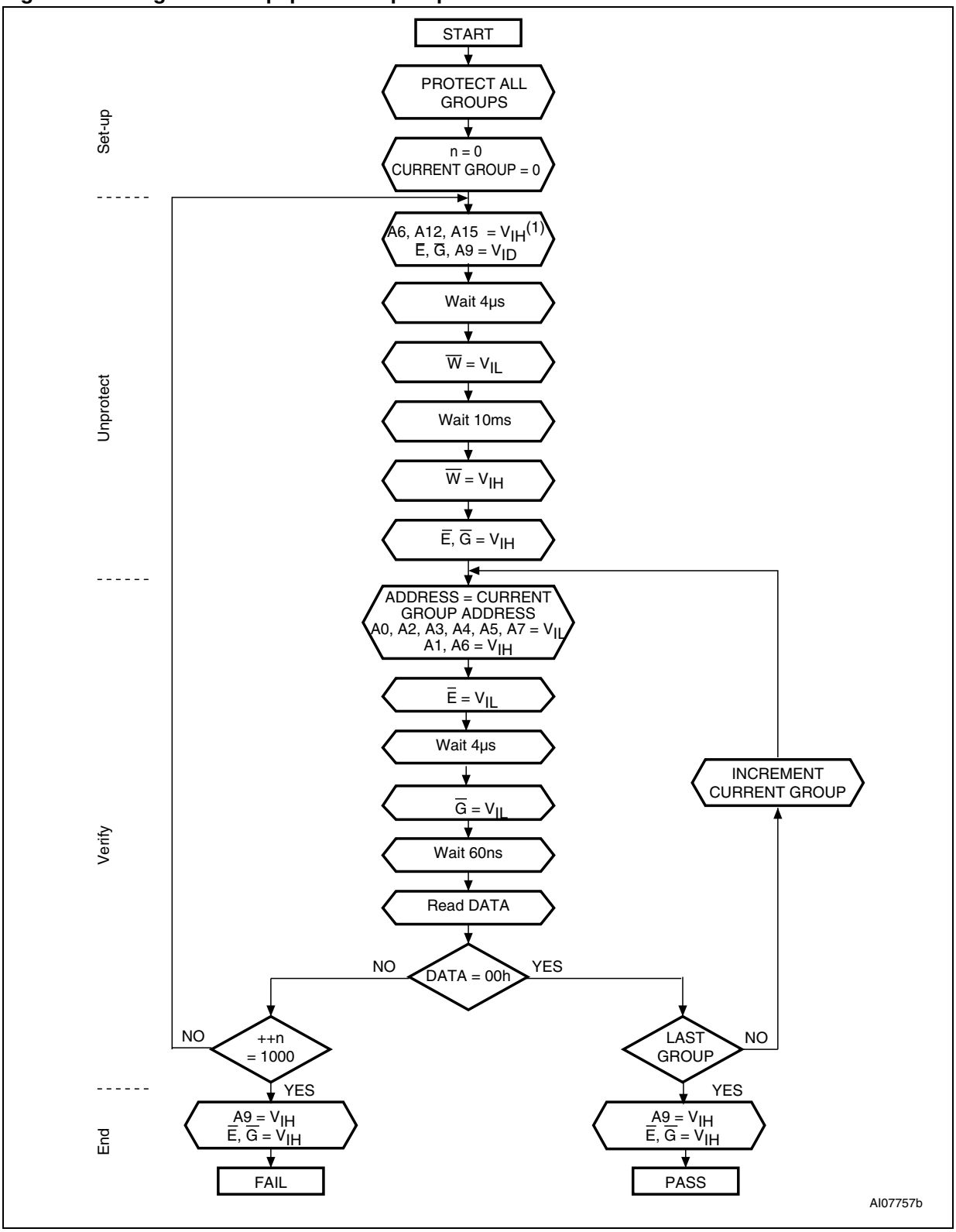
# Appendix E Flowcharts

**Figure 18. Programmer equipment Group Protect flowchart**



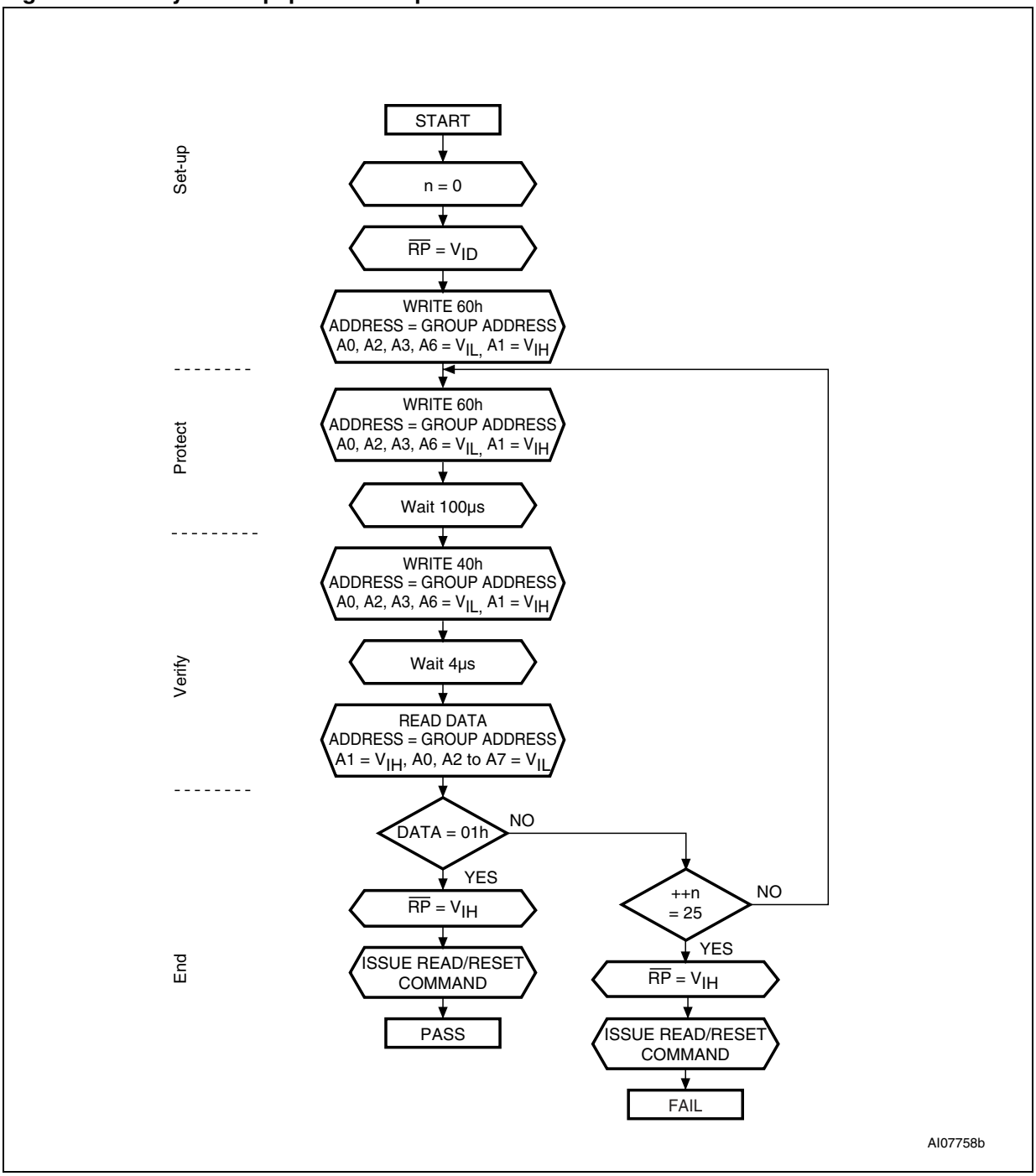

1. Block Protection Groups are shown in *Appendix D: High Voltage Block Protection*, *Table 28*.

**Figure 19. Programmer equipment Chip Unprotect flowchart**



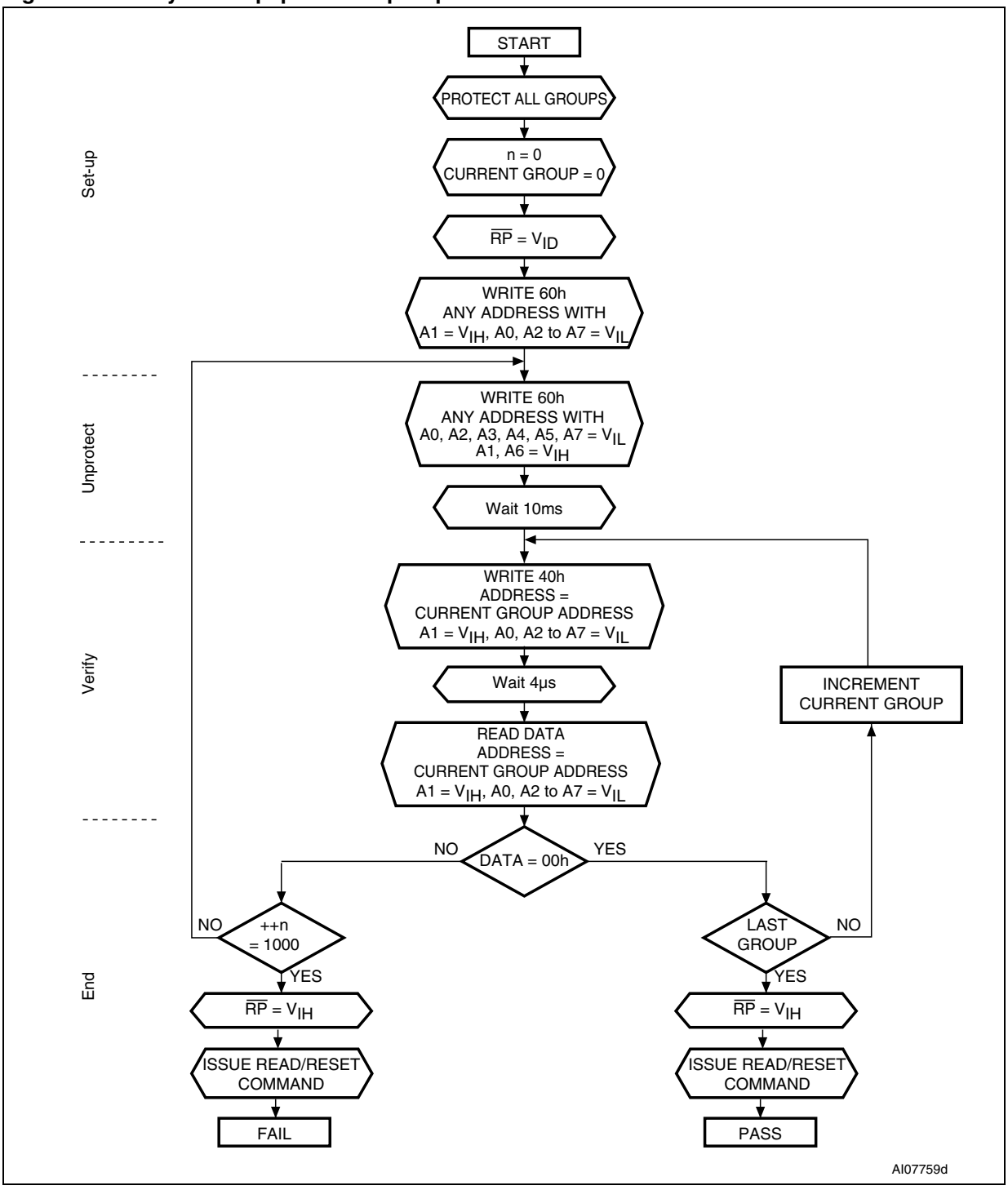

1. Block Protection Groups are shown in *Appendix D: High Voltage Block Protection*, *Table 28*.

**Figure 20. In-System equipment Group Protect flowchart**



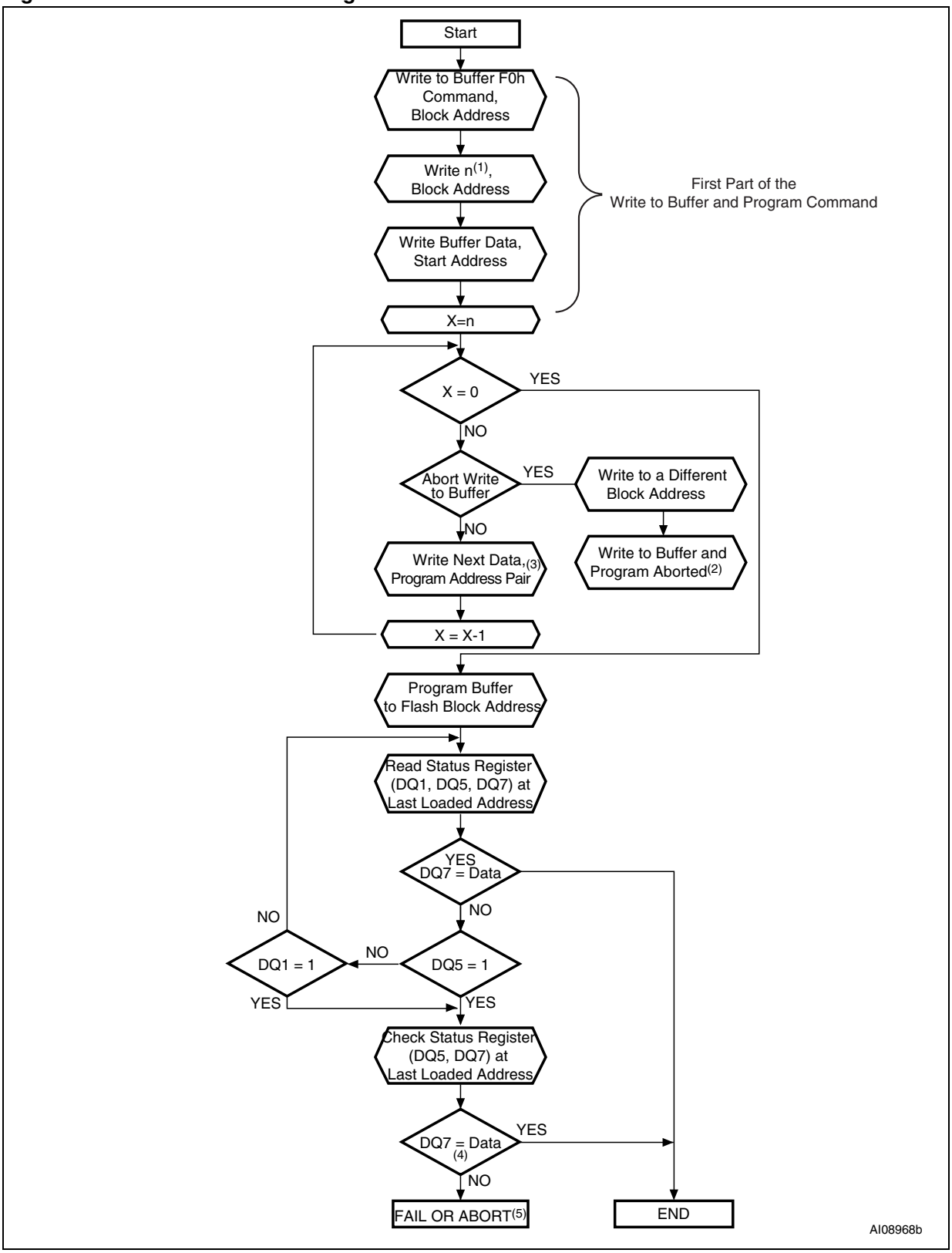

1. Block Protection Groups are shown in *Appendix D: High Voltage Block Protection*, *Table 28*.
2. $\overline{RP}$ can be either at $V_{IH}$ or at $V_{ID}$ when using the In-System Technique to protect the Extended Memory Block.

**Figure 21. In-System equipment Chip Unprotect flowchart**




1. Block Protection Groups are shown in *Appendix D: High Voltage Block Protection*, *Table 28*.

**Figure 22. Write to Buffer and Program flowchart and Pseudo Code**




1. n+1 is the number of addresses to be programmed.

2. A Write to Buffer and Program Abort and Reset must be issued to return the device in Read mode.
3. When the block address is specified, any address in the selected block address space is acceptable. However when loading Write Buffer address with data, all addresses must fall within the selected Write Buffer page.
4. DQ7 must be checked since DQ5 and DQ7 may change simultaneously.
5. If this flowchart location is reached because DQ5='1', then the Write to Buffer and Program command failed. If this flowchart location is reached because DQ1='1', then the Write to Buffer and Program command aborted. In both cases, the appropriate reset command must be issued to return the device in Read mode: a Reset command if the operation failed, a Write to Buffer and Program Abort and Reset command if the operation aborted.
6. See *Table 9* and *Table 10*, for details on Write to Buffer and Program command sequence.

# Revision history

**Table 37. Document revision history**

| Date | Version | Changes |
|---|---|---|
| 29-Sep-2005 | 0.1 | First Issue. |
| 02-Dec-2005 | 1 | Document status changed to “Full Datasheet“.<br>Title updated.<br>Program Suspend Latency time updated in *Table 15: Program, Erase Times and Program, Erase Endurance cycles*. |
| 07-Mar-2006 | 2 | DQ7 changed to $\overline{DQ7}$ for Program, Program During Erase Suspend and Program Error in *Table 16: Status register bits*. |
| 13-Mar-2006 | 3 | *Section 5.2.1: Write to Buffer and Program command*, and *Section 5.2.2: Write to Buffer and Program Confirm command* updated to cover 8-bit mode. *Note 2*, *Note 3*, and *Note 4* updated in *Table 11: Fast Program commands, 8-bit mode*. |
| 06-Apr-2006 | 4 | Verify Extended Memory Block Protection bit command removed. |
| 25-Oct-2006 | 5 | *Table 16: Status register bits* updated. |
| 06-Nov-2006 | 6 | DQ7 was replaced by $\overline{DQ7}$ for ‘Write to Buffer and Program Abort’ in *Table 16: Status register bits*. |
| 10-Dec-2007 | 7 | Applied Numonyx branding. |

**Please Read Carefully:**

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH NUMONYX™ PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN NUMONYX'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, NUMONYX ASSUMES NO LIABILITY WHATSOEVER, AND NUMONYX DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF NUMONYX PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Numonyx products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

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

Numonyx, B.V. may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

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

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

Copies of documents which have an order number and are referenced in this document, or other Numonyx literature may be obtained by visiting Numonyx's website at http://www.numonyx.com.

Numonyx StrataFlash is a trademark or registered trademark of Numonyx or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.