

Parallel NOR Flash Embedded Memory

M29W160ET, M29W160EB

Features

- Supply voltage
 - $V_{CC} = 2.7\text{--}3.6\text{V}$ (program, erase, read)
- Access times
 - 70, 90ns
- Program time
 - 10 μs per byte/word (TYP)
- Memory organization
 - 3 parameter and 31 main blocks
 - 1 boot block (top or bottom location)
- Program/erase controller
 - Embedded byte/word program algorithms
- Erase suspend and resume capability
 - Read or program another block during an ERASE SUSPEND operation
- UNLOCK BYPASS PROGRAM COMMAND
 - Fast buffered/batch programming
- Temporary block unprotect mode
- Common Flash interface
 - 64-bit security code
- Low power consumption: Standby and automatic mode
- 100,000 PROGRAM/ERASE cycles per block
- Electronic signature
 - Manufacturer code: 0020h
 - Top device code M29W160ET: 22C4h
 - Bottom device code M29W160EB: 2249h
- Packages
 - 48-pin TSOP (N) 12mm x 20mm
 - 48-ball TFBGA (ZA) 6mm x 8mm
 - 64-ball FBGA (ZS) 11mm x 13mm
- Automotive grade parts available

Part Numbering Information

Devices are shipped from the factory with memory content bits erased to 1. For available options, such as packages or speed, or for further information, contact your Micron sales representative. Part numbers can be verified at www.micron.com. Feature and specification comparison by device type is available at www.micron.com/products. Contact the factory for devices not found.

Table 1: Part Number Information

| Part Number Category | Category Details | Notes |
|----------------------|--|-------|
| Device Type | M29 = Parallel Flash memory | |
| Operating Voltage | W = 2.7 to 3.6V | |
| Device function | 160E = 16Mb memory array | |
| Configuration | T = Top boot B = Bottom boot | |
| Speed | 7A = 70ns | 1 |
| | 70 = 70 ns | 2 |
| | 80 = 80ns | 3 |
| | 90 = 90ns | 4 |
| Package | N = 48-pin TSOP, 12mm x 20mm | |
| | ZA = 48-ball TFBGA, 6mm x 8mm, 0.80mm pitch | |
| | ZS = 64-ball Fortified BGA, 11mm x 13mm, 1mm pitch | |
| Temperature Range | 6 = -40° to 85°C | |
| | 3 = -40°C to 125°C | |
| Voltage Extension | Blank = Standard option | |
| | S = $V_{CC,min}$ extension to 2.5V of V_{CC} and available only with 80ns speed class option | |
| Shipping Options | Blank = Standard packing | |
| | T = Tape and reel packing | |
| | E = RoHS-compliant package, standard packing | |
| | F = RoHS-compliant package, tape and reel packing | |

- Notes:
1. Device speed in conjunction with temperature range = 6 to denote automotive grade (-40° to 85°C) parts.
 2. Device speed in conjunction with temperature range = 6 to denote industrial grade (-40° to 85°C) parts, or in conjunction with temperature range = 3 to denote automotive grade (-40° to 125°C) parts.
 3. Access time, automotive device, in conjunction with temperature range = 3 and voltage extension = S.
 4. Device speed in conjunction with temperature range = 6 to denote industrial grade (-40° to 85°C) parts.



Contents

| | |
|---|----|
| General Description | 6 |
| Signal Assignments | 8 |
| Signal Descriptions | 11 |
| Memory Organization | 13 |
| Memory Configuration | 13 |
| Memory Map, x8 – 16Mb Density | 13 |
| Memory Map, x16 – 16Mb Density | 14 |
| Bus Operations | 15 |
| Read | 15 |
| Write | 15 |
| Standby and Automatic Standby | 15 |
| Output Disable | 16 |
| Commands | 16 |
| READ Operations | 17 |
| READ/RESET Command | 17 |
| READ CFI Command | 17 |
| AUTO SELECT Operations | 18 |
| AUTO SELECT Command | 18 |
| Read Device ID | 18 |
| Block and Chip Protection | 19 |
| BLOCK PROTECT Command | 19 |
| Block Protection Using Programmer Equipment | 20 |
| In-System Block Protection | 22 |
| BYPASS Operations | 24 |
| UNLOCK BYPASS Command | 24 |
| UNLOCK BYPASS RESET Command | 24 |
| PROGRAM Operations | 24 |
| PROGRAM Command | 24 |
| UNLOCK BYPASS PROGRAM Command | 24 |
| ERASE Operations | 25 |
| CHIP ERASE Command | 25 |
| BLOCK ERASE Command | 25 |
| ERASE SUSPEND Command | 26 |
| ERASE RESUME Command | 26 |
| Status Register | 27 |
| Data Polling Bit (DQ7) | 27 |
| Toggle Bit (DQ6) | 27 |
| Error Bit (DQ5) | 27 |
| Erase Timer Bit (DQ3) | 28 |
| Alternative Toggle Bit (DQ2) | 28 |
| Absolute Ratings and Operating Conditions | 31 |
| DC Characteristics | 33 |
| Read AC Characteristics | 34 |
| Write AC Characteristics | 36 |
| Program/Erase Characteristics | 40 |
| Reset Characteristics | 41 |
| Package Dimensions | 42 |
| Revision History | 45 |
| Rev. B – 06/13 | 45 |
| Rev. A – 07/12 | 45 |

List of Figures

| | |
|---|----|
| Figure 1: Logic Diagram | 7 |
| Figure 2: 48-Pin TSOP 160ET/B | 8 |
| Figure 3: 48-Ball TFBGA 160ET/B | 9 |
| Figure 4: 64-Ball FBGA 160ET/B | 10 |
| Figure 5: Block Protect Flowchart – Programmer Equipment | 20 |
| Figure 6: Chip Unprotect Flowchart – Programmer Equipment | 21 |
| Figure 7: Block Protect Flowchart – In-System Equipment | 22 |
| Figure 8: Chip Protection Flowchart – In-System Equipment | 23 |
| Figure 9: Data Polling Flowchart | 29 |
| Figure 10: Data Toggle Flowchart | 30 |
| Figure 11: AC Measurement Load Circuit | 32 |
| Figure 12: AC Measurement I/O Waveform | 32 |
| Figure 13: Random AC Timing | 35 |
| Figure 14: WE#-Controlled AC Timing | 37 |
| Figure 15: CE#-Controlled AC Timing | 39 |
| Figure 16: Reset/Block Temporary Unprotect AC Waveforms | 41 |
| Figure 17: 48-Pin TSOP – 12mm x 20mm | 42 |
| Figure 18: 48-Ball TFBGA – 6mm x 8mm | 43 |
| Figure 19: 64-Ball FBGA – 11mm x 13mm | 44 |



List of Tables

| | |
|--|----|
| Table 1: Part Number Information | 2 |
| Table 2: Signal Names | 7 |
| Table 3: Signal Descriptions | 11 |
| Table 4: x8 Top Boot, Blocks [34:0] | 13 |
| Table 5: x8 Bottom Boot, Blocks [34:0] | 13 |
| Table 6: x16 Top Boot, Blocks [34:0] | 14 |
| Table 7: x16 Bottom Boot, Blocks [34:0] | 14 |
| Table 8: Bus Operations | 15 |
| Table 9: Commands – 16-Bit Mode (BYTE# = V _{IL}) | 16 |
| Table 10: Commands – 8-Bit Mode (BYTE# = V _{IL}) | 17 |
| Table 11: Read Electronic Signature | 18 |
| Table 12: Block and Chip Protection Signal Settings | 19 |
| Table 13: Status Register Bits | 28 |
| Table 14: Absolute Maximum/Minimum Ratings | 31 |
| Table 15: Operating Conditions | 31 |
| Table 16: Input/Output Capacitance | 32 |
| Table 17: DC Current Characteristics | 33 |
| Table 18: DC Voltage Characteristics | 33 |
| Table 19: Read AC Characteristics | 34 |
| Table 20: WE#-Controlled Write AC Characteristics | 36 |
| Table 21: CE#-Controlled Write AC Characteristics | 38 |
| Table 22: Program/Erase Times and Endurance Cycles | 40 |
| Table 23: Reset/Block Temporary Unprotect AC Characteristics | 41 |

General Description

The M29W160ET/B (2Mb x8 or 1Mb x16) is a nonvolatile device that can be read, erased and reprogrammed. These operations can be performed using a single low voltage (2.7–3.6V) supply. On power-up the memory defaults to read mode where it can be read in the same way as a ROM or EPROM.

The device is divided into blocks that can be erased independently to preserve valid data while old data is erased. Each block can be protected independently to prevent accidental PROGRAM or ERASE commands from modifying the memory. 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 blocks in the memory are asymmetrically arranged. The first or last 64KB have been divided into four additional blocks. The 16KB boot block can be used for a small initialization code to start the microprocessor, the two 8 KB parameter blocks can be used for parameter storage, and the remaining 32KB is a small main block where the application may be stored.

CE#, OE#, and WE# signals control the bus operation. They enable simple connection to most microprocessors, often without additional logic.

The device supplied with all the bits erased (set to 1).

Figure 1: Logic Diagram

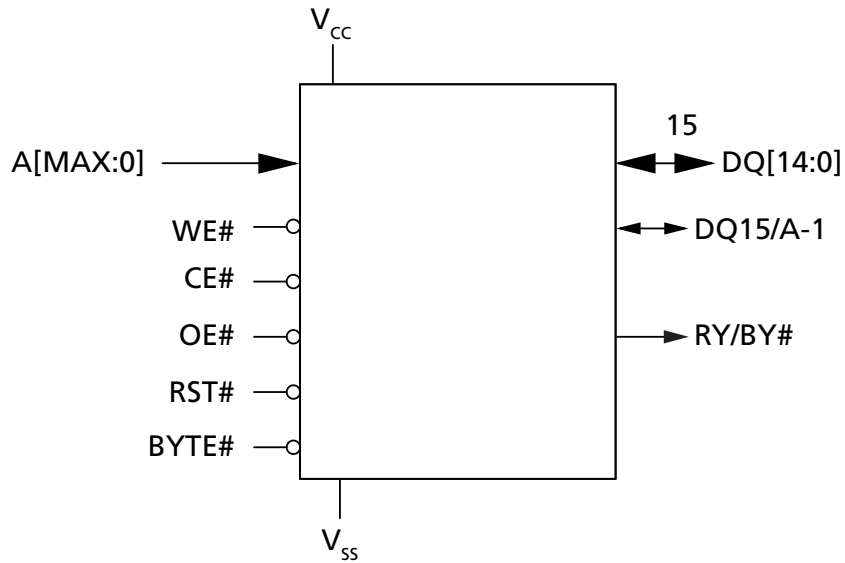
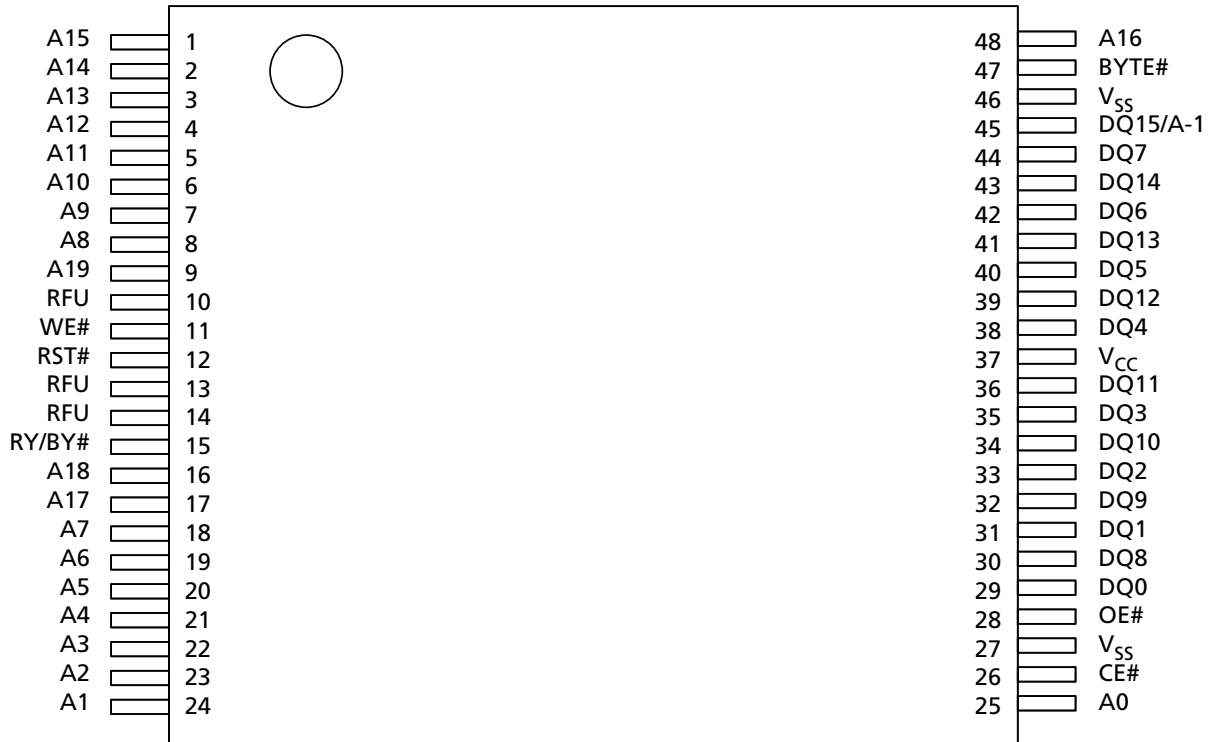


Table 2: Signal Names

| Name | Description | Type |
|-----------------|---------------------------------|--------|
| A[19:0] | Address inputs | Input |
| CE# | Chip enable | Input |
| OE# | Output enable | Input |
| WE# | Write enable | Input |
| BYTE# | Byte/word organization select | Input |
| RST# | Reset/block temporary unprotect | Input |
| DQ[7:0] | Data I/O | I/O |
| DQ[14:8] | Data I/O | I/O |
| DQ15/A-1 | Data I/O or address input | I/O |
| RY/BY# | Ready/busy output | Output |
| V _{CC} | Core power supply | Supply |
| V _{SS} | Ground | Supply |
| NC | Not connected internally | - |

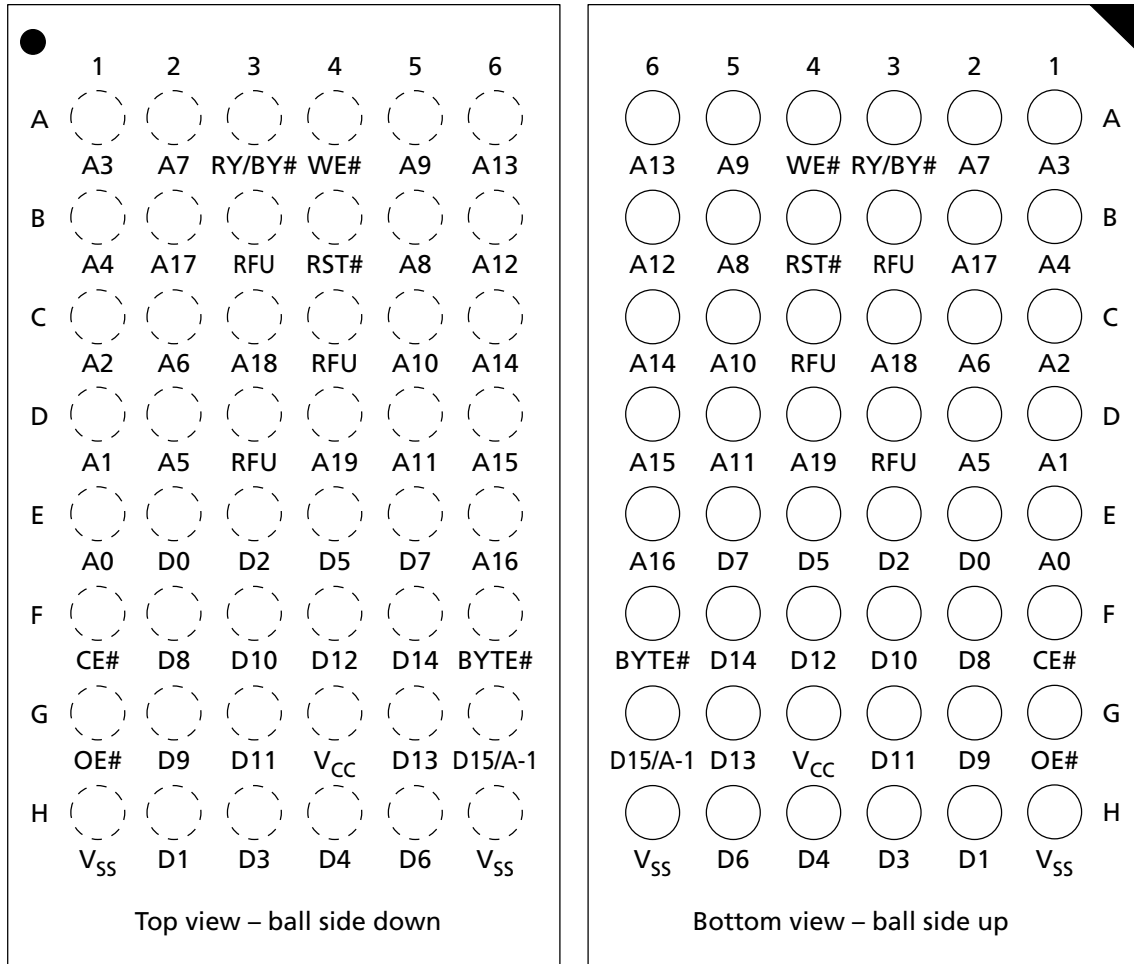
Signal Assignments

Figure 2: 48-Pin TSOP 160ET/B



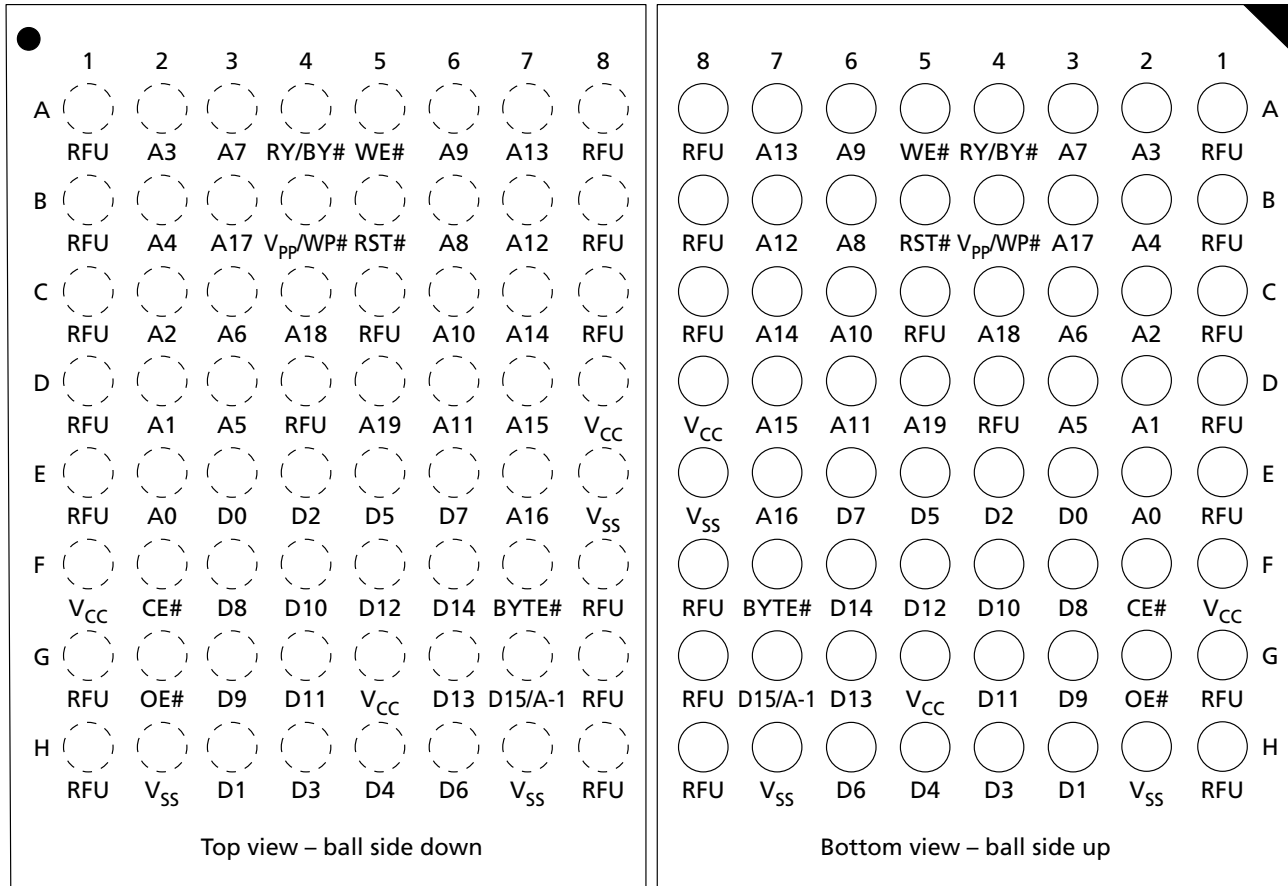
Note: 1. RFU = reserved for future use.

Figure 3: 48-Ball TFBGA 160ET/B



Note: 1. RFU = reserved for future use.

Figure 4: 64-Ball FBGA 160ET/B



Note: 1. RFU = reserved for future use.

Signal Descriptions

The table below is a comprehensive list of signals for this device family. All signals listed may not be supported on this device. See Signal Assignments for information specific to this device.

Table 3: Signal Descriptions

| Name | Type | Description |
|----------|--------|--|
| A[MAX:0] | Input | Address: Selects the cells in the memory array to access during READ operations. During WRITE operations, controls the commands sent to the command interface of the program/erase controller. |
| CE# | Input | Chip enable: Activates the memory, enabling READ and WRITE operations. When CE# is HIGH, all other pins are ignored. |
| OE# | Input | Output enable: Controls the bus READ operation of the memory. |
| WE# | Input | Write enable: Controls the bus WRITE operation of the command interface. |
| BYTE# | Input | Byte/word organization select: Switches between x8 and x16 bus modes. When BYTE# is LOW, the device is in x8 mode; when HIGH, the device is in x16 mode. |
| RST# | Input | Reset/block temporary unprotect: Applies a hardware reset to the memory or temporarily removes protection from all blocks that have been protected. A hardware reset is achieved by holding RST# LOW for at least t_{PLPX} . When RST# goes HIGH, the memory is ready for READ and WRITE operations after t_{PHEL} or t_{RHEL} , whichever occurs last. Holding RST# at V_{ID} temporarily unprotects the protected blocks so that PROGRAM and ERASE operations are possible on all blocks. The transition from HIGH to V_{ID} must be slower than t_{PHPHH} . |
| DQ[7:0] | I/O | Data I/O: Outputs the data stored at the selected address during READ operations. During WRITE operations, represents the commands sent to the command interface of the program/erase controller. |
| DQ[14:8] | I/O | Data I/O: Outputs data stored at the selected address during a READ operation when BYTE# is HIGH. When BYTE# is LOW, these pins are not used and are High-Z. During a WRITE operation, the command register does not use these bits. When reading the status register, these bits should be ignored. |
| DQ15/A-1 | I/O | Data I/O or address input: When BYTE# is HIGH, this pin behaves as a data I/O pin, DQ[14:8]. When BYTE# is LOW, this pin behaves as an address pin; DQ15/A-1 LOW selects the LSB of the word on the other addresses, DQ15/A-1 HIGH selects the MSB. Throughout the text, consider references to data I/O to include this pin when BYTE# is HIGH and consider references to address inputs to include this pin when BYTE# is LOW, except when stated explicitly otherwise. |
| RY/BY# | Output | Ready/busy: Open-drain output that can be used to identify when the device can be read. RY/BY# is High-Z during read, auto select, and erase suspend modes. After a hardware reset, a READ or WRITE operation cannot begin until RY/BY# becomes High-Z. During a PROGRAM or ERASE operation, RY/BY# is LOW and remains LOW during READ/RESET commands or hardware resets until the memory is ready to enter read mode. The use of an open-drain output enables the RY/BY# pins from several memory devices to be connected to a single pull-up resistor. A LOW indicates that one or more of the devices is busy. |

Table 3: Signal Descriptions (Continued)

| Name | Type | Description |
|-----------------|--------|--|
| V _{CC} | Supply | Supply voltage: Provides the power supply for device operations. The command interface is disabled when V _{CC} ≤ V _{LKO} . This prevents a WRITE operation 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, the operation aborts, and the contents being altered will be invalid. See Note 1. |
| V _{SS} | Supply | Ground: All V _{SS} pins must be connected to the system ground. |
| RFU | – | Reserved for future use: RFUs should be not connected. |

- Note: 1. A 0.1μF capacitor should be connected between V_{CC} and V_{SS} 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.

Memory Organization

Memory Configuration

The main memory array is divided into 64KB blocks.

The blocks in the memory are asymmetrically arranged. The first or last 64KB of memory has been divided into four additional blocks. The 16KB boot block can be used for small initialization code to start the microprocessor, the two 8KB parameter blocks can be used for parameter storage and the remaining 32KB small main block can be used for application storage.

Memory Map, x8 – 16Mb Density

Table 4: x8 Top Boot, Blocks [34:0]

| Block | Block Size | Address Range | |
|-------|------------|---------------|-----------|
| | | Start | End |
| 34 | 16KB | 001F C000 | 001F FFFF |
| 33 | 8KB | 001F A000 | 001F BFFF |
| 32 | 8KB | 001F 8000 | 001F 9FFF |
| 31 | 32KB | 001F 0000 | 001F 7FFF |
| 30 | 64KB | 001E 0000 | 001E FFFF |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 2 | 64KB | 0002 0000 | 0002 FFFF |
| 1 | 64KB | 0001 0000 | 0001 FFFF |
| 0 | 64KB | 0000 0000 | 0000 FFFF |

Table 5: x8 Bottom Boot, Blocks [34:0]

| Block | Block Size | Address Range | |
|-------|------------|---------------|-----------|
| | | Start | End |
| 34 | 64KB | 001F 0000 | 001F FFFF |
| 33 | 64KB | 001E 0000 | 001E FFFF |
| 32 | 64KB | 001D 0000 | 001D FFFF |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 4 | 64KB | 0001 0000 | 0001 FFFF |
| 3 | 32KB | 0000 8000 | 0000 FFFF |
| 2 | 8KB | 0000 6000 | 0000 7FFF |
| 1 | 8KB | 0000 4000 | 0000 5FFF |
| 0 | 16KB | 0000 0000 | 0000 3FFF |



Memory Map, x16 – 16Mb Density

Table 6: x16 Top Boot, Blocks [34:0]

| Block | Block Size | Address Range | |
|-------|------------|---------------|-----------|
| | | Start | End |
| 34 | 8KW | 000F E000 | 000F FFFF |
| 33 | 4KW | 000F D000 | 000F DFFF |
| 32 | 4KW | 000F C000 | 000F CFFF |
| 31 | 16KW | 000F 8000 | 000F BFFF |
| 30 | 32KW | 001E 0000 | 001E FFFF |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 2 | 32KW | 0001 0000 | 0001 7FFF |
| 1 | 32KW | 0008 0000 | 0000 FFFF |
| 0 | 32KW | 0000 0000 | 0000 7FFF |

Table 7: x16 Bottom Boot, Blocks [34:0]

| Block | Block Size | Address Range | |
|-------|------------|---------------|-----------|
| | | Start | End |
| 34 | 32KW | 000F 8000 | 000F FFFF |
| 33 | 32KW | 000F 0000 | 000F 7FFF |
| 32 | 32KW | 000E 8000 | 000E FFFF |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 4 | 32KW | 0000 8000 | 0000 FFFF |
| 3 | 16KW | 0000 4000 | 0000 7FFF |
| 2 | 4KW | 0000 3000 | 0000 3FFF |
| 1 | 4KW | 0000 2000 | 0000 2FFF |
| 0 | 8KW | 0000 0000 | 0000 1FFF |

Bus Operations

Table 8: Bus Operations

Notes 1 and 2 apply to entire table

| Operation | CE# | OE# | WE# | 8-Bit Mode | | | 16-Bit Mode | |
|-------------------|-----|-----|-----|-----------------------|----------|-------------------------|-----------------|-------------------------|
| | | | | A[MAX:0], DQ15/A-1 | DQ[14:8] | DQ[7:0] | A[MAX:0] | DQ15/A-1, DQ[14:0] |
| READ | L | L | H | Cell address | High-Z | Data output | Cell address | Data output |
| WRITE | L | H | L | Command address | High-Z | Data input ⁴ | Command address | Data input ⁴ |
| STANDBY | H | X | X | X | High-Z | High-Z | X | High-Z |
| OUTPUT DISABLE | X | H | H | X | High-Z | High-Z | X | High-Z |

- Notes:
1. Typical glitches of less than 5ns on CE# and WE# are ignored by the device and do not affect bus operations.
 2. H = Logic level HIGH (V_{IH}); L = Logic level LOW (V_{IL}); X = HIGH or LOW.
 3. If WP# is LOW, the highest or lowest block remains protected, depending on the line item.
 4. Data input is required when issuing a command sequence or performing data polling or block protection.

Read

Bus READ operations read from the memory cells, registers, or CFI space. A valid READ operation requires setting the appropriate address on the address inputs, taking CE# and OE# LOW and holding WE# HIGH. Data I/O signals output the value.

Write

Bus WRITE operations write to the command interface. A valid WRITE operation requires setting the appropriate address on the address inputs. These are latched by the command interface on the falling edge of CE# or WE#, whichever occurs last. Values on data I/O signals are latched by the command interface on the rising edge of CE# or WE#, whichever occurs first. OE# must remain HIGH during the entire operation.

Standby and Automatic Standby

When the device is in read mode, driving CE# HIGH places the device in standby mode and drives data I/Os to High-Z. Supply current is reduced to standby (I_{CC2}), by holding CE# within $V_{CC} \pm 0.2V$.

During PROGRAM or ERASE operations, the device continues to use the program/erase supply current (I_{CC3}) until the operation completes.

Automatic standby enables low power consumption during read mode. When CMOS levels ($V_{CC} \pm 0.2V$) drive the bus, and following a READ operation and a period of inactivity specified in DC Characteristics, the memory enters automatic standby as internal supply current is reduced to I_{CC2} . Data I/O signals still output data if a READ operation is in progress.

Output Disable

Data I/Os are High-Z when OE# is HIGH.

Commands

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. See the x8 and x16 command tables, depending on the configuration that is being used, for a summary of the commands.

Table 9: Commands – 16-Bit Mode (BYTE# = V_{IL})

| Command | Length | Bus WRITE Operations | | | | | | | | | | | |
|------------------------|--------|----------------------|------|------|------|------|------|------|------|------|------|------|------|
| | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | |
| | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data |
| READ/RESET | 1 | X | F0 | | | | | | | | | | |
| | 3 | 555 | AA | 2AA | 55 | X | F0 | | | | | | |
| AUTO SELECT | 3 | 555 | AA | 2AA | 55 | 555 | 90 | | | | | | |
| PROGRAM | 4 | 555 | AA | 2AA | 55 | 555 | A0 | PA | PD | | | | |
| UNLOCK BY-PASS | 3 | 555 | AA | 2AA | 55 | 555 | 20 | | | | | | |
| UNLOCK BY-PASS PROGRAM | 2 | X | A0 | PA | PD | | | | | | | | |
| UNLOCK BY-PASS RESET | 2 | X | 90 | X | 00 | | | | | | | | |
| 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 SUSPEND | 1 | X | B0 | | | | | | | | | | |
| ERASE RESUME | 1 | X | 30 | | | | | | | | | | |
| READ CFI QUERY | 1 | 55 | 98 | | | | | | | | | | |

Note: 1. X = "Don't Care;" PA = Program address; PD = Program data; BA = Any address in the block. All values in the table are in hexadecimal. The command interface only uses A-1, A0-A10, and DQ0-DQ7 to verify the commands; A11-A20, DQ8-DQ14, and DQ15 are "Don't Care." DQ15A-1 is A-1 when BYTE# is V_{IL} or DQ15 when BYTE# is V_{IH}.

Table 10: Commands – 8-Bit Mode (BYTE# = V_{IL})

| Command | Length | Bus WRITE Operations | | | | | | | | | | | | |
|------------------------|--------|----------------------|------|------|------|------|------|------|------|------|------|------|------|--|
| | | 1st | | 2nd | | 3rd | | 4th | | 5th | | 6th | | |
| | | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | Addr | Data | |
| READ/RESET | 1 | X | F0 | | | | | | | | | | | |
| | 3 | AAA | AA | 555 | 55 | X | F0 | | | | | | | |
| AUTO SELECT | 3 | AAA | AA | 555 | 55 | AAA | 90 | | | | | | | |
| PROGRAM | 4 | AAA | AA | 555 | 55 | AAA | A0 | PA | PD | | | | | |
| UNLOCK BY-PASS | 3 | AAA | AA | 555 | 55 | AAA | 20 | | | | | | | |
| UNLOCK BY-PASS PROGRAM | 2 | X | A0 | PA | PD | | | | | | | | | |
| UNLOCK BY-PASS RESET | 2 | X | 90 | X | 00 | | | | | | | | | |
| 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 SUSPEND | 1 | X | B0 | | | | | | | | | | | |
| ERASE RESUME | 1 | X | 30 | | | | | | | | | | | |
| READ CFI QUERY | 1 | 55 | 98 | | | | | | | | | | | |

Note: 1. X = " Don't Care;" PA = Program address; PD = Program data; BA = Any address in the block. All values in the table are in hexadecimal. The command interface only uses A-1, A0-A10, and DQ0-DQ7 to verify the commands; A11-A20, DQ8-DQ14, and DQ15 are "Don't Care." DQ15A-1 is A-1 when BYTE# is V_{IL} or DQ15 when BYTE# is V_{IH}.

READ Operations

READ/RESET Command

The READ/RESET (F0h) command returns the device to read mode and resets the errors in the status register. One or three bus WRITE operations can be used to issue the READ/RESET command.

To return the device to read mode, this command can be issued between bus WRITE cycles before the start of a PROGRAM or ERASE operation. If the READ/RESET command is issued during the timeout of a BLOCK ERASE operation, the device requires up to 10μs to abort, during which time no valid data can be read.

READ CFI Command

The READ CFI (98h) command puts the device in read CFI mode and is valid only when the device is in read array or auto select mode. One bus WRITE cycle is required to issue the command.

Once in read CFI mode, bus READ operations will output data from the CFI memory area. A READ/RESET command must be issued to return the device to the previous mode (read array or auto select). A second READ/RESET command is required to put the device in read array mode from auto select mode.

AUTO SELECT Operations

AUTO SELECT Command

At power-up or after a hardware reset, the device is in read mode. It can then be put in auto select mode by issuing an AUTO SELECT (90h) command or by applying V_{ID} to A9. Auto select mode enables the following device information to be read:

- Electronic signature, which includes manufacturer and device code information, as shown in the Read Electronic Signature table.
- Block protection, which includes the block protection status and extended memory block protection indicator, as shown in the Block Protection table.

Electronic signature or block protection information is read by executing a READ operation with control signals and addresses set.

Auto select mode can be used by the programming equipment to automatically match a device with the application code to be programmed.

Three consecutive bus WRITE operations are required to issue an AUTO SELECT command. The device remains in auto select mode until a READ/RESET or READ CFI command is issued.

The device cannot enter auto select mode when a PROGRAM or ERASE operation is in progress (RY/BY# LOW). However, auto select mode can be entered if the PROGRAM or ERASE operation has been suspended by issuing a PROGRAM SUSPEND or ERASE SUSPEND command.

Auto select mode is exited by performing a reset. The device returns to read mode unless it entered auto select mode after an ERASE SUSPEND or PROGRAM SUSPEND command, in which case it returns to erase or program suspend mode.

Read Device ID

Table 11: Read Electronic Signature

Note 1 applies to entire table

| READ Cycle | CE# | OE# | WE# | Address Input | | | | | Data I/O | | |
|-------------------|-----|-----|-----|--------------------|----------|-----------------|----|----|-------------------|--------------------------------------|---|
| | | | | 8-Bit/16-Bit | | | | | 8-Bit Only | | 16-Bit Only |
| | | | | A[MAX:10] | A9 | A[8:2] | A1 | A0 | DQ[14:8] | DQ[7:0] | DQ[15]/A-1, DQ[14:0] |
| Manufacturer code | L | L | H | X | V_{ID} | X | L | L | High-Z | 20h | 0020h |
| Device code | L | L | H | X | V_{ID} | X | L | H | High-Z | C4h ² 49h ³ | 22C4h ² 2249h ³ |

- Notes:
1. H = Logic level HIGH (V_{IH}); L = Logic level LOW (V_{IL}); X = HIGH or LOW.
 2. M29W160ET.
 3. M29W160EB.

Block and Chip Protection

Block protection can be used to prevent any operation from modifying the data stored in the Flash. Each block can be protected individually. Once protected, PROGRAM and ERASE operations on the block fail to change the data.

Do not allow microprocessor service interrupts to interfere with timing, and do not abort an operation before its completion. The CHIP UNPROTECT operation can take several seconds, and a user message should be provided to show progression. (Refer to the following flowcharts for details.)

Unlike the command interface of the program/erase controller, techniques for protecting and unprotecting blocks change from one Flash memory supplier to another. Care should be taken when changing drivers for one part to work on another.

BLOCK PROTECT Command

There are three techniques that can be used to control block protection. These are programmer technique, in-system technique, and temporary unprotect. Temporary unprotect is controlled by RST#.

Unlike the command interface of the program/erase controller, the techniques for protecting and unprotecting blocks change between different Flash memory suppliers.

Table 12: Block and Chip Protection Signal Settings

| Signals | Block Protect | Chip Unprotect | Verify Block Protection | Verify Block Unprotect |
|--|--------------------|-----------------|-------------------------|------------------------|
| CE# | L | V _{ID} | L | L |
| OE# | V _{ID} | V _{ID} | L | L |
| WE# | L pulse | L pulse | H | H |
| Address Input, 8-Bit and 16-Bit | | | | |
| A[MAX:16] | Block base address | X | Block base address | Block base address |
| A15 | | H | | |
| A14 | | X | | |
| A13 | | X | | |
| A12 | | H | | |
| A11 | X | X | X | X |
| A10 | X | X | X | X |
| A9 | V _{ID} | V _{ID} | V _{ID} | V _{ID} |
| A8 | X | X | X | X |
| A7 | X | X | X | X |
| A6 | X | X | L | H |
| A5 | X | X | X | X |
| A4 | X | X | X | X |
| A3 | X | X | X | X |
| A2 | X | X | X | X |
| A1 | X | X | H | H |

Table 12: Block and Chip Protection Signal Settings (Continued)

| Signals | Block Protect | Chip Unprotect | Verify Block Protection | Verify Block Unprotect |
|-----------------------------------|---------------|----------------|-------------------------|------------------------|
| A0 | X | X | L | L |
| Data I/O, 8-Bit and 16-Bit | | | | |
| DQ[15]/A-1, and DQ[14:0] | X | X | Pass = XX01h | Retry = XX01h |
| | X | X | Retry = XX00h | Pass = XX00h |

Note: 1. H = Logic level HIGH (V_{IH}); L = Logic level LOW (V_{IL}); X = HIGH or LOW.

Block Protection Using Programmer Equipment

The programmer technique uses high voltage levels (V^{ID}) 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 block, follow the steps in the following figure. To unprotect the whole chip, it is necessary to protect all of the blocks first, then all blocks can be unprotected at the same time.

Figure 5: Block Protect Flowchart – Programmer Equipment

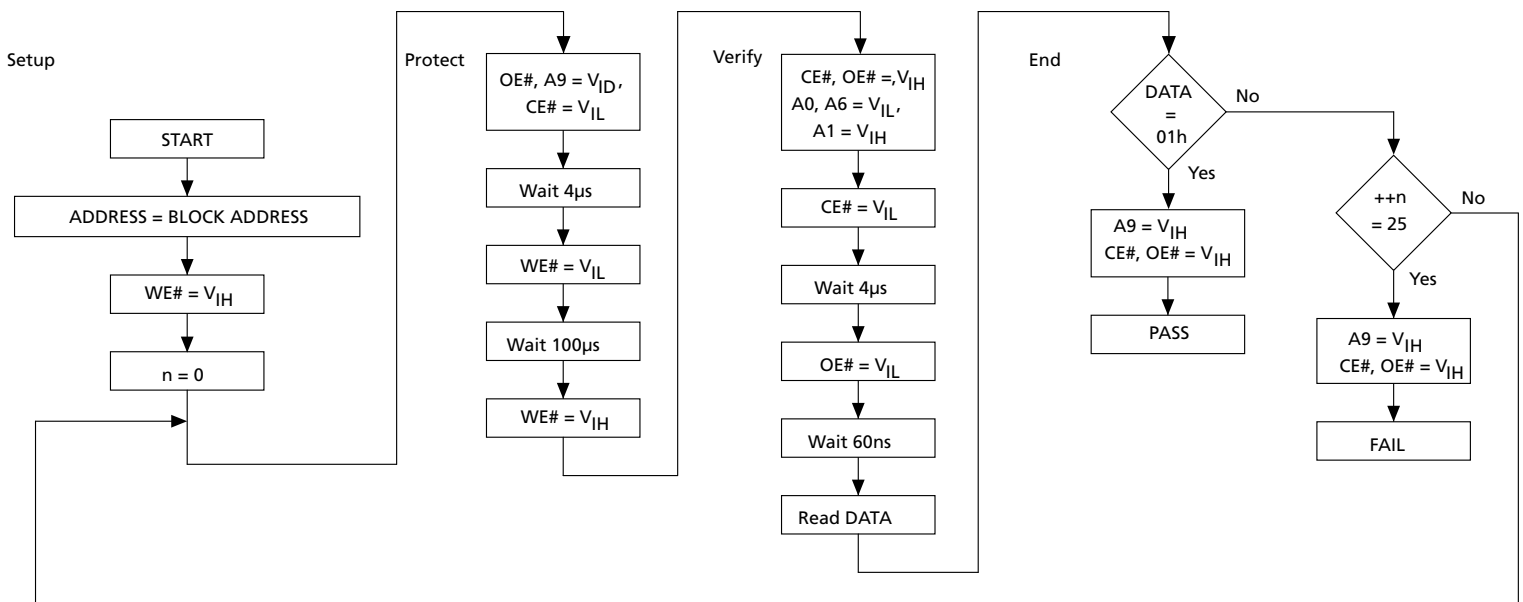
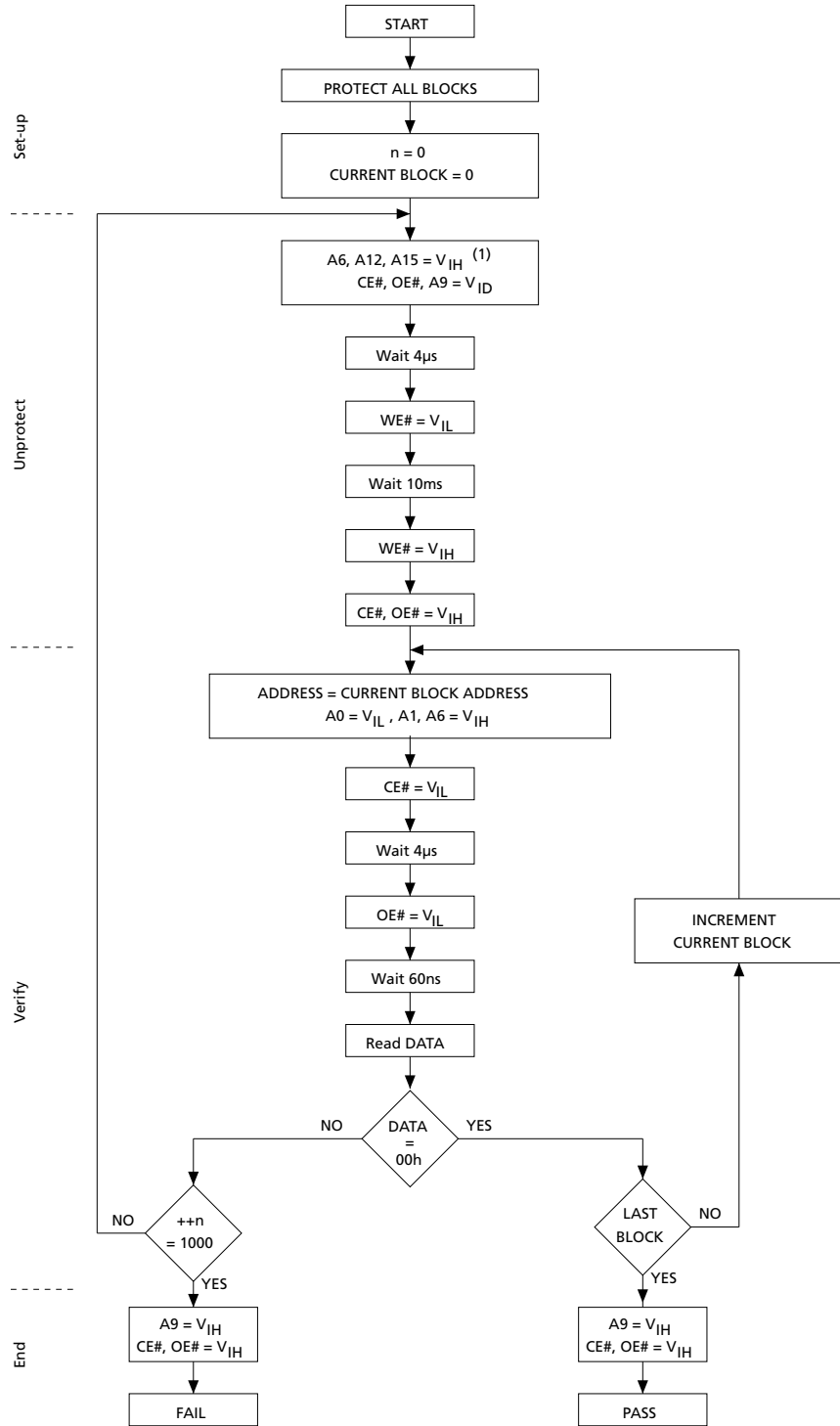


Figure 6: Chip Unprotect Flowchart – Programmer Equipment



- Notes:
1. Address Inputs A[9:12] give the address of the block that is to be protected. It is imperative that they remain stable during the operation.
 2. During the protect and verify phases of the algorithm, CE# must be kept LOW.

In-System Block Protection

The in-system technique requires a high-voltage level on RST#. 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 Flash has been fitted to the system.

To protect a block, follow the steps in the following figure. To unprotect the whole chip, it is necessary to protect all of the blocks first, then all the blocks can be unprotected at the same time.

Figure 7: Block Protect Flowchart – In-System Equipment

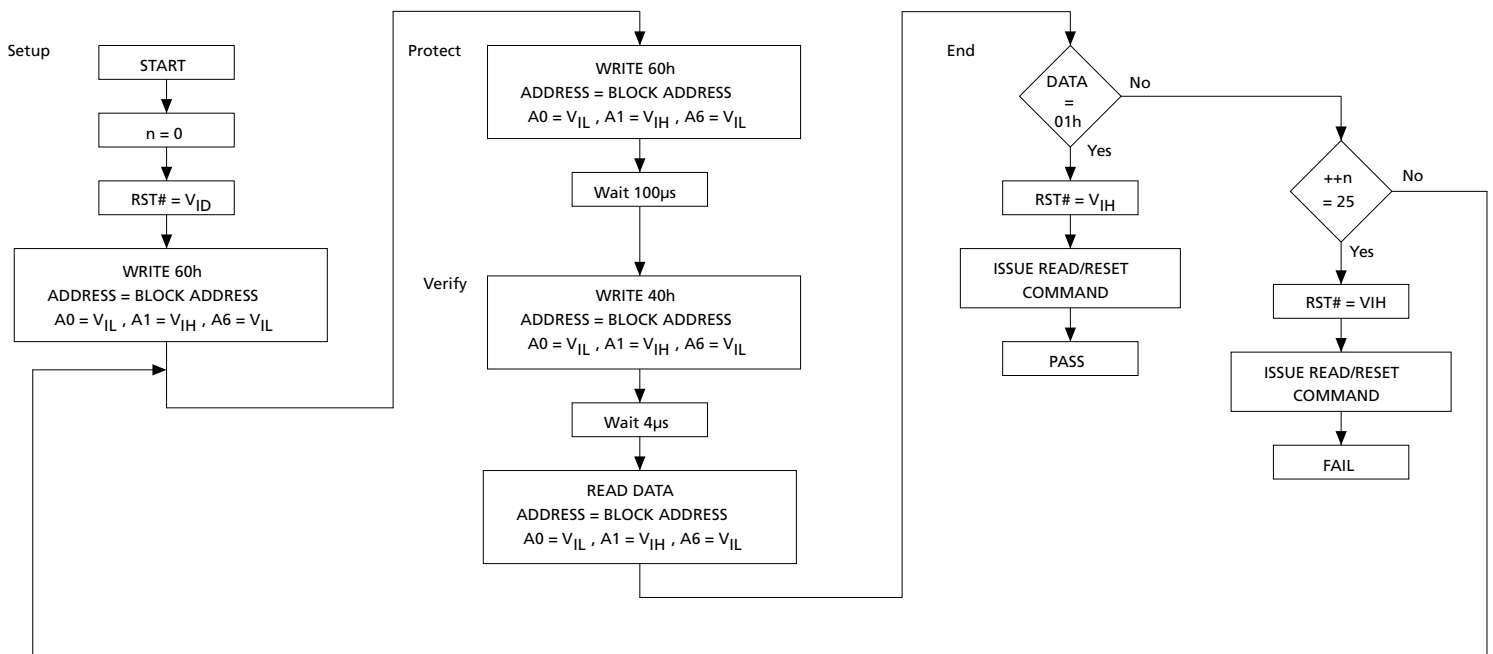
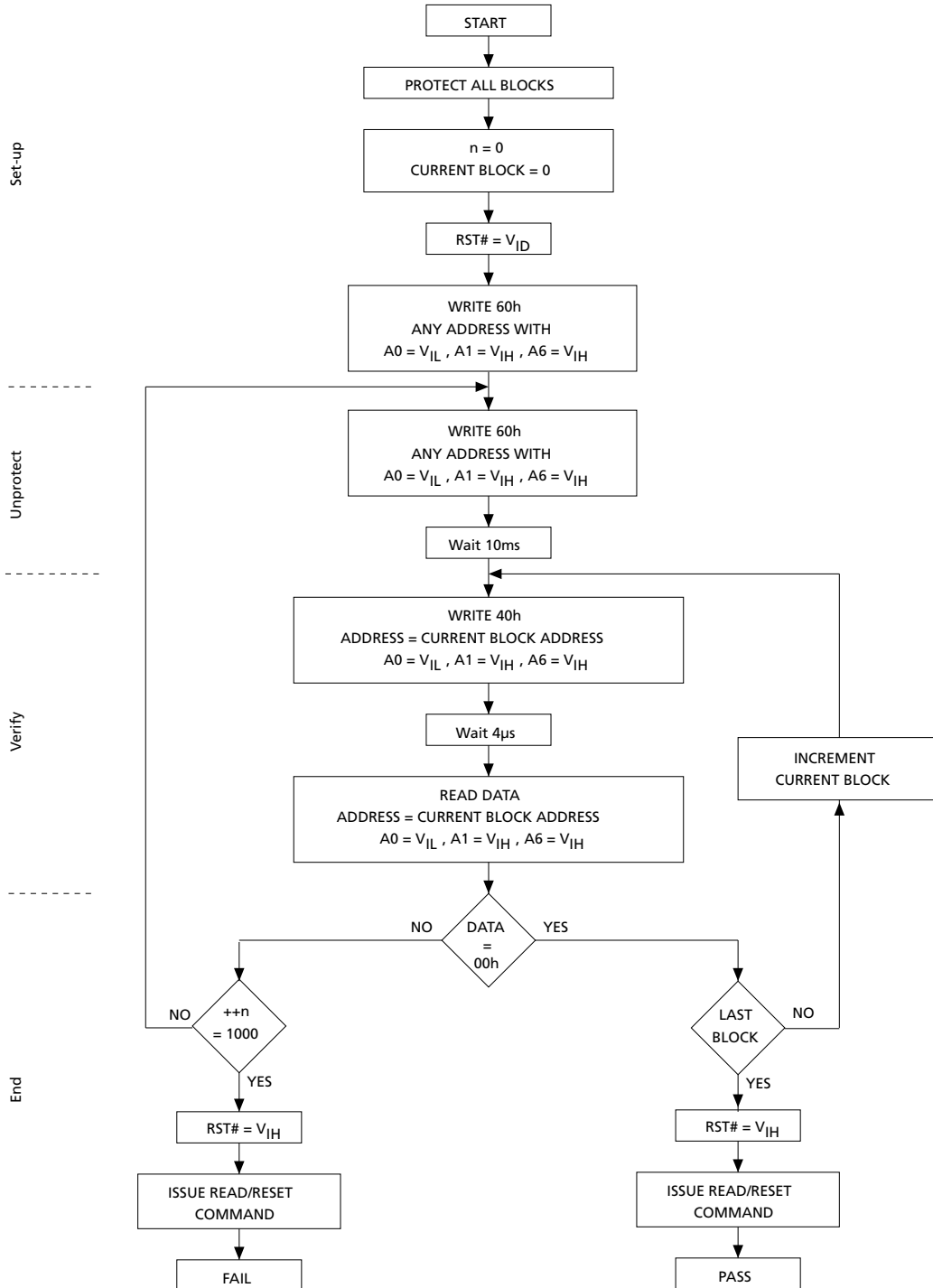


Figure 8: Chip Protection Flowchart – In-System Equipment



BYPASS Operations

UNLOCK BYPASS Command

The UNLOCK BYPASS command is used with the UNLOCK BYPASS PROGRAM command to program the memory. When device access time is slow, as with some EEPROM programmers, considerable time can be saved using BYPASS operations. Three bus WRITE operations are required to issue the UNLOCK BYPASS command.

After the UNLOCK BYPASS command is issued, the memory will accept only the UNLOCK BYPASS PROGRAM and the UNLOCK BYPASS RESET commands. The memory can be read as if in read mode.

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 this command. The READ/RESET command does not exit the device from unlock bypass mode.

PROGRAM Operations

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 and starts the program/erase controller.

If the address falls in a protected block, the command is ignored, the data remains unchanged, the status register is never read, and no error condition is given.

During the PROGRAM operation, the memory ignores all commands. It is not possible to issue any command to abort or pause the operation. A READ operation during the PROGRAM operation will output the status register on the data I/O.

When the PROGRAM operation completes, the memory returns to the read mode unless an error has occurred. When an error occurs, the memory continues to output the status register. A READ/RESET command must be issued to reset the error condition and return to read mode.

Note: The PROGRAM command cannot change a bit set at 0 back to 1. An ERASE command must be used to set all the bits in a block or in the whole memory from 0 to 1.

UNLOCK BYPASS PROGRAM Command

When the device is in unlock bypass mode, the UNLOCK BYPASS PROGRAM (A0h) command can be used to program one address in the memory array. The command requires two bus WRITE operations instead of the four required by a standard PROGRAM command; the final WRITE operation latches the address and data and starts the program/erase controller. (The standard PROGRAM command requires four bus WRITE operations.) 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.

ERASE Operations

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 100 μ s, leaving the data unchanged. No error condition is given when protected blocks are ignored.

During the ERASE operation, the memory will ignore all commands. It is not possible to issue any command to abort the operation. Typical chip erase times are given in the Program/Erase Times and Program/Erase Endurance Cycles table. All bus READ operations during the CHIP ERASE operation will output the status register on the data I/O. (See the Status Register section 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.

BLOCK ERASE Command

The BLOCK ERASE command can be used to erase a list of one or more blocks. 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 about 50 μ s after the last bus WRITE operation. After the program/erase controller starts, it is not possible to select any more blocks. Each additional block must therefore be selected within 50 μ s of the last block. The 50 μ s timer restarts when an additional block is selected. The status register can be read after the sixth bus WRITE operation. (See the Status Register section for details on how to identify if the program/erase controller has started the BLOCK ERASE operation.)

If any selected blocks are protected, 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 100 μ s, 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. Typical block erase times are given in the Program/Erase Times and Program/Erase Endurance Cycles table. All bus READ operations during the BLOCK ERASE operation will output the status register on the data inputs/outputs. (See the Status Register section for more details.)

After the BLOCK 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 BLOCK ERASE command sets all of the bits in the unprotected selected blocks to 1. All previous data in the selected blocks is lost.

ERASE SUSPEND Command

The ERASE SUSPEND command may be used to temporarily suspend a BLOCK ERASE operation and return the memory to read mode. The command requires one bus WRITE operation.

The program/erase controller will suspend within the erase suspend latency time after the ERASE SUSPEND command is issued. (See the Program/Erase Times and Program/Erase Endurance Cycles table for numerical values.) After the program/erase controller has stopped, the memory will be set to read mode, and the ERASE operation 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), the ERASE operation 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, 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 and UNLOCK BYPASS commands during an ERASE SUSPEND operation. The READ/RESET command must be issued to return the device to read array mode before the RESUME command will be accepted.

ERASE RESUME Command

The ERASE RESUME command must be used to restart the program/erase controller from ERASE SUSPEND. An ERASE operation can be suspended and resumed more than once.

Status Register

Bus READ operations from any address always read the status register during PROGRAM and ERASE operations. It is also read during ERASE SUSPEND operations when an address within a block being erased is accessed. The bits in the status register are summarized in the Status Register Bits table.

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 operation. 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. The 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.

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 command. The toggle bit is output on DQ6 when the status register is read.

During PROGRAM and ERASE operations, 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.

If any attempt is made to erase a protected block, the operation is aborted, no error is signaled, and DQ6 toggles for approximately 100 μ s. If any attempt is made to program a protected block or a suspended block, the operation is aborted, no error is signaled, and DQ6 toggles for approximately 1 μ s. The Data Toggle Flowchart gives an example of how to use the data toggle bit.

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.

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. When 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.

Alternative Toggle Bit (DQ2)

The alternative toggle bit can be used to monitor the program/erase controller during ERASE operations. It 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. After the operation completes, the memory returns to read mode.

During an ERASE SUSPEND operation, 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 cell 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.

Table 13: Status Register Bits

| Operation | Address | DQ7 | DQ6 | DQ5 | DQ3 | DQ2 | RY/BY# |
|------------------------------|-------------------|---------------------|-----------|-----|-----|-----------|--------|
| PROGRAM | Any address | DQ7# | Toggle | 0 | – | – | 0 |
| PROGRAM DURING ERASE SUSPEND | Any address | DQ7# | Toggle | 0 | – | – | 0 |
| PROGRAM ERROR | Any address | DQ7# | Toggle | 1 | – | – | 0 |
| CHIP ERASE | Any address | 0 | Toggle | 0 | 1 | Toggle | 0 |
| BLOCK ERASE BEFORE TIMEOUT | Erasing block | 0 | Toggle | 0 | 0 | Toggle | 0 |
| | Non-erasing block | 0 | Toggle | 0 | 0 | No Toggle | 0 |
| BLOCK ERASE | Erasing block | 0 | Toggle | 0 | 1 | Toggle | 0 |
| | Non-erasing block | 0 | Toggle | 0 | 1 | No Toggle | 0 |
| ERASE SUSPEND | Erasing block | 1 | No Toggle | 0 | – | Toggle | 1 |
| | Non-erasing block | Data read as normal | | | | | |

Table 13: Status Register Bits (Continued)

| Operation | Address | DQ7 | DQ6 | DQ5 | DQ3 | DQ2 | RY/BY# |
|-------------|----------------------|-----|--------|-----|-----|-----------|--------|
| ERASE ERROR | Good block address | 0 | Toggle | 1 | 1 | No Toggle | 0 |
| | Faulty block address | 0 | Toggle | 1 | 1 | Toggle | 0 |

Note: 1. Unspecified data bits should be ignored.

Figure 9: Data Polling Flowchart

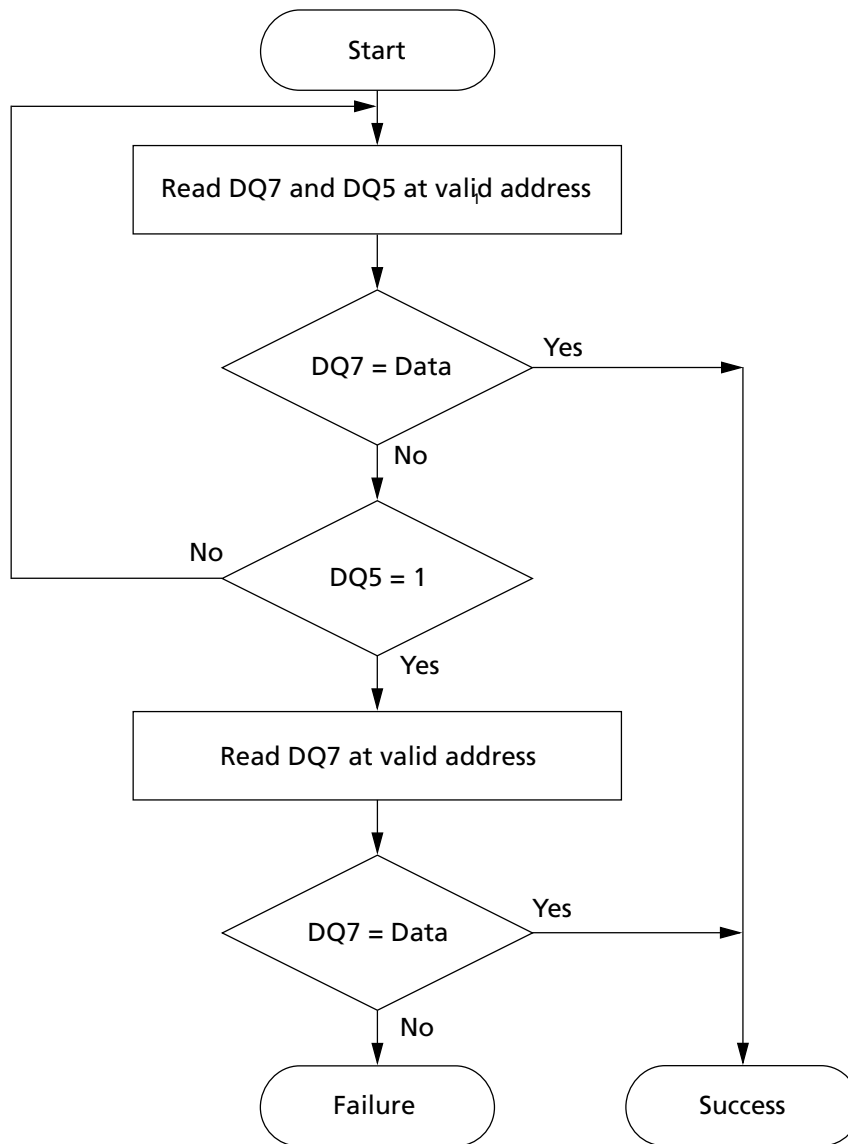
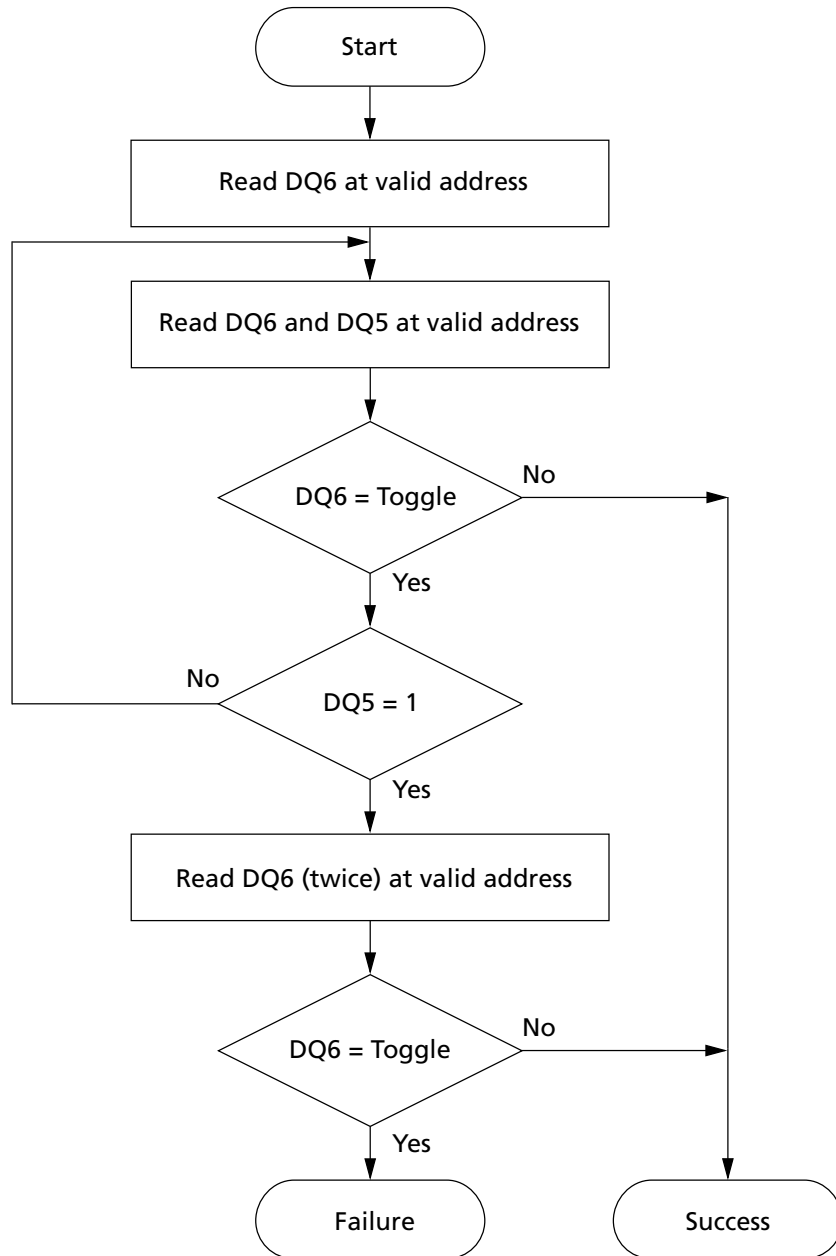


Figure 10: Data Toggle Flowchart



Absolute Ratings and Operating Conditions

Stresses greater than those listed may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions outside those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may adversely affect reliability.

Table 14: Absolute Maximum/Minimum Ratings

| Parameter | Symbol | Min | Max | Unit | Notes |
|------------------------|------------|------|----------------|------|-------|
| Temperature under bias | T_{BIAS} | -50 | 125 | °C | |
| Storage temperature | T_{STG} | -65 | 150 | °C | |
| Input/output voltage | V_{IO} | -0.6 | $V_{CC} + 0.6$ | V | 1, 2 |
| Supply voltage | V_{CC} | -0.6 | 4 | V | |
| Identification voltage | V_{ID} | -0.6 | 13.5 | V | |

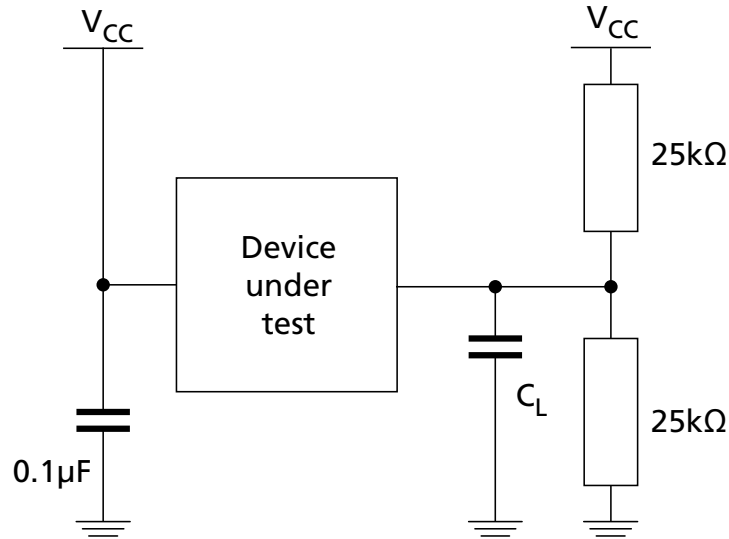
- Notes:
1. During signal transitions, minimum voltage may undershoot to $-2V$ for periods less than 20ns.
 2. During signal transitions, maximum voltage may overshoot to $V_{CC} + 2V$ for periods less than 20ns.

Table 15: Operating Conditions

| Parameter | Symbol | M29W160ET/B | | | | | | | | Unit |
|--|----------|---------------|---------------------|---------------|-----|-------------------|-----|---------------|-----|------|
| | | 70ns | | 7Ans | | 80ns ¹ | | 90ns | | |
| | | Min | Max | Min | Max | Min | Max | Min | Max | |
| Supply voltage | V_{CC} | 2.7 | 3.6 | 2.7 | 3.6 | 2.5 | 3.6 | 2.7 | 3.6 | V |
| Ambient operating temperature (device grade 6) | T_A | -40 | 85/125 ² | -40 | 85 | -40 | 125 | -40 | 85 | °C |
| Load capacitance | C_L | 30 | | 30 | | 30 | | 30 | | pF |
| Input rise and fall times | - | - | 10 | - | 10 | - | 10 | - | 10 | ns |
| Input pulse voltages | - | 0 to V_{CC} | | 0 to V_{CC} | | 0 to V_{CC} | | 0 to V_{CC} | | V |
| Input and output timing reference voltages | - | $V_{CC}/2$ | | $V_{CC}/2$ | | $V_{CC}/2$ | | $V_{CC}/2$ | | V |

- Notes:
1. 80ns option supported only on -40°C to 125°C devices.
 2. 85°C = Industrial part; 125°C = Automotive grade part.

Figure 11: AC Measurement Load Circuit



Note: 1. C_L includes jig capacitance.

Figure 12: AC Measurement I/O Waveform

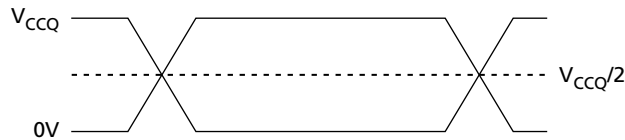


Table 16: Input/Output Capacitance

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

Note: 1. Sampled only, not 100% tested.

DC Characteristics

Table 17: DC Current Characteristics

| Parameter | Symbol | Conditions | Typ | Max | Unit | Notes |
|------------------------------|-----------|--|-----|---------|---------|-------|
| Input leakage current | I_{LI} | $0V \leq V_{IN} \leq V_{CC}$ | – | ± 1 | μA | |
| Output leakage current | I_{LO} | $0V \leq V_{OUT} \leq V_{CC}$ | – | ± 1 | μA | |
| Supply read current | I_{CC1} | $CE\# = V_{IL}$, $OE\# = V_{IH}$, $f = 6 \text{ MHz}$ | 4.5 | 10 | mA | |
| Supply standby current | I_{CC2} | $CE\# = V_{CC} \pm 0.2V$ $RP\# = V_{CC} \pm 0.2V$ | 35 | 100 | μA | 1 |
| Supply program/erase current | I_{CC3} | Program/erase controller active | – | 20 | mA | 2 |
| Identification current | I_{ID} | $A9 = V_{ID}$ | – | 100 | μA | |

- Notes: 1. When the bus is inactive for 150ns or more, the memory enters automatic standby.
2. Sampled only; not 100% tested.

Table 18: DC Voltage Characteristics

| Parameter | Symbol | Conditions | Min | Max | Unit |
|--------------------------------------|-----------|-------------------------|----------------|----------------|------|
| Input LOW voltage | V_{IL} | – | –0.5 | 0.8 | V |
| Input HIGH voltage | V_{IH} | – | $0.7 V_{CC}$ | $V_{CC} + 0.3$ | V |
| Output LOW voltage | V_{OL} | $I_{OL} = 1.8\text{mA}$ | – | 0.45 | V |
| Output HIGH voltage | V_{OH} | $I_{OH} = -100\mu A$ | $V_{CC} - 0.4$ | – | V |
| Identification voltage | V_{ID} | – | 11.5 | 12.5 | V |
| Program/erase lockout supply voltage | V_{LKO} | – | 1.8 | 2.3 | V |

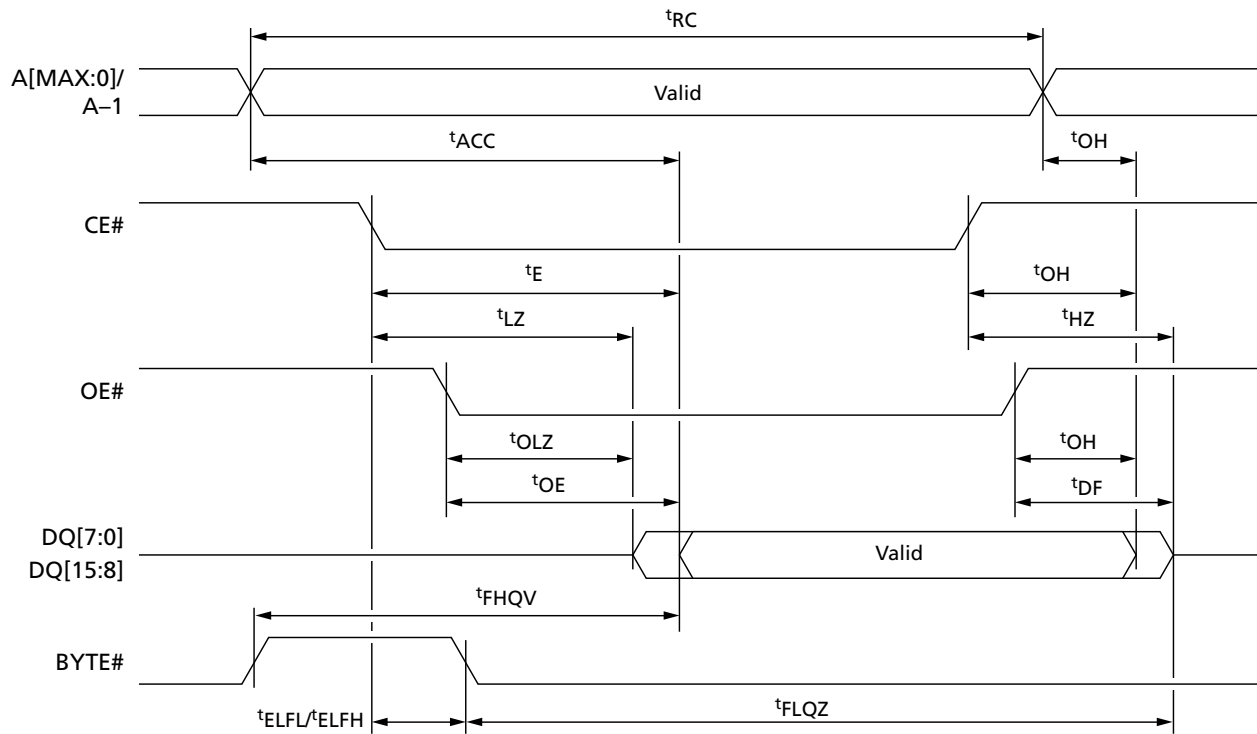
Read AC Characteristics

Table 19: Read AC Characteristics

| Parameter | Symbol | | Condition | 7A/70/80 ¹ | | 90 | | Unit | Notes |
|--|-------------------|---|--|-----------------------|-----|-----|-----|------|-------|
| | Legacy | JEDEC | | Min | Max | Min | Max | | |
| Address valid to next address valid | ^t RC | ^t AVAV | CE# = V _{IL} , OE# = V _{IL} | 70 | – | 90 | – | ns | |
| Address valid to output valid | ^t ACC | ^t AVQV | CE# = V _{IL} , OE# = V _{IL} | – | 70 | – | 90 | ns | |
| CE# LOW to output transition | ^t LZ | ^t ELQX | OE# = V _{IL} | 0 | – | 0 | – | ns | 2 |
| CE# LOW to output valid | ^t E | ^t ELQV | OE# = V _{IL} | – | 70 | – | 90 | ns | |
| OE# LOW to output transition | ^t OLZ | ^t GLQX | CE# = V _{IL} | 0 | – | 0 | – | ns | 2 |
| OE# LOW to output valid | ^t OE | ^t GLQV | CE# = V _{IL} | – | 25 | – | 35 | ns | |
| CE# HIGH to output High-Z | ^t HZ | ^t EHQZ | OE# = V _{IL} | – | 25 | – | 30 | ns | 2 |
| OE# HIGH to output High-Z | ^t DF | ^t GHQZ | CE# = V _{IL} | – | 25 | – | 30 | ns | 2 |
| CE#, OE#, or address transition to output transition | ^t OH | ^t EHQX, ^t GHQX, ^t AXQX | – | 0 | – | 0 | – | ns | |
| CE# to BYTE# LOW | ^t ELFL | ^t ELBL | – | – | 5 | – | 5 | ns | |
| CE# to BYTE# HIGH | ^t ELFH | ^t ELBH | – | – | 5 | – | 5 | ns | |
| BYTE# LOW to output High-Z | ^t FLQZ | ^t BLQZ | – | – | 25 | – | 30 | ns | |
| BYTE# HIGH to output valid | ^t FHQV | ^t BHQV | – | – | 30 | – | 40 | ns | |

- Notes: 1. 70ns becomes 80ns if the 80ns device code is used.
2. Sampled only; not 100% tested.

Figure 13: Random AC Timing



Write AC Characteristics

Table 20: WE#-Controlled Write AC Characteristics

| Parameter | Symbol | | 7A/70/80 ¹ | | 90 | | Unit | Notes |
|-------------------------------------|-------------------|--------------------|-----------------------|-----|-----|-----|------|-------|
| | Legacy | JEDEC | Min | Max | Min | Max | | |
| Address valid to next address valid | t ^{WC} | t ^{AVAV} | 70 | – | 90 | – | ns | |
| CE# LOW to WE# LOW | t ^{CS} | t ^{ELWL} | 0 | – | 0 | – | ns | |
| WE# LOW to WE# HIGH | t ^{WP} | t ^{WLWH} | 45 | – | 50 | – | ns | |
| Input valid to WE# HIGH | t ^{DS} | t ^{DVWH} | 45 | – | 50 | – | ns | 2 |
| WE# HIGH to input transition | t ^{DH} | t ^{WHDX} | 0 | – | 0 | – | ns | |
| WE# HIGH to CE# HIGH | t ^{CH} | t ^{WHEH} | 0 | – | 0 | – | ns | |
| WE# HIGH to WE# LOW | t ^{WPH} | t ^{WHWL} | 30 | – | 30 | – | ns | |
| Address valid to WE# LOW | t ^{AS} | t ^{AVWL} | 0 | – | 0 | – | ns | |
| WE# LOW to address transition | t ^{AH} | t ^{WLAX} | 45 | – | 50 | – | ns | |
| OE# HIGH to WE# LOW | – | t ^{GHWL} | 0 | – | 0 | – | ns | |
| WE# HIGH to OE# LOW | t ^{OEH} | t ^{WHGL} | 0 | – | 0 | – | ns | |
| Program/erase valid to RY/BY# LOW | t ^{BUSY} | t ^{WHRL} | – | 30 | – | 35 | ns | 2 |
| V _{CC} HIGH to CE# LOW | t ^{VCS} | t ^{VCHEL} | 50 | – | 50 | – | μs | |

- Notes: 1. 70ns becomes 80ns if the 80ns device code is used.
2. Sampled only; not 100% tested.

Figure 14: WE#-Controlled AC Timing

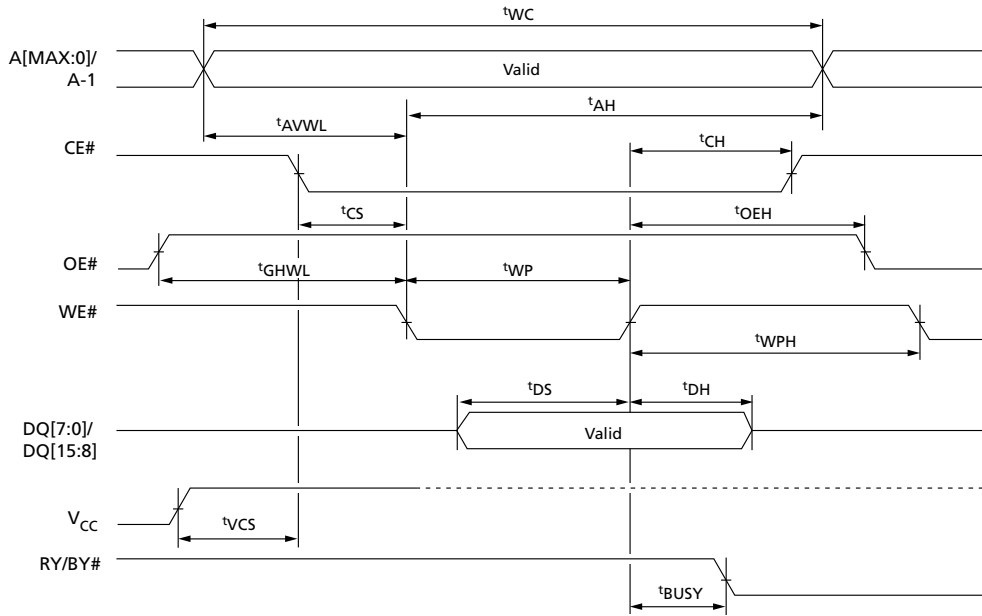
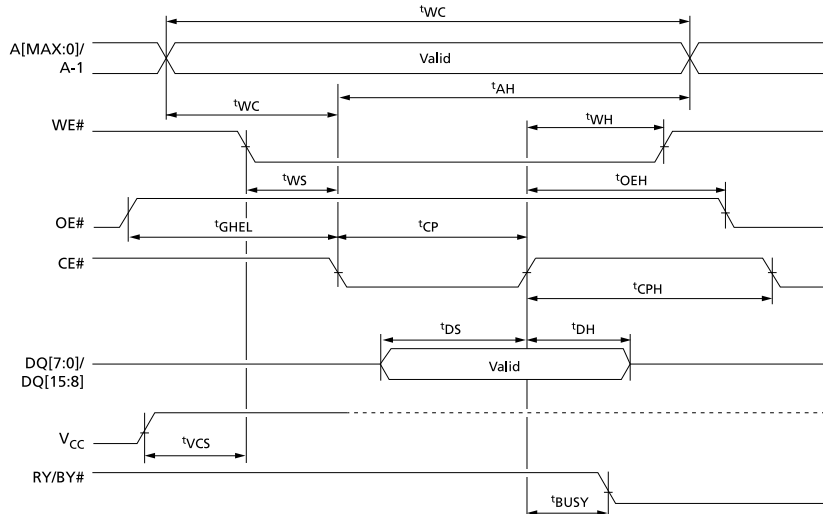


Table 21: CE#-Controlled Write AC Characteristics

| Parameter | Symbol | | 7A/70/80 ² | | 90 | | Unit |
|-------------------------------------|-----------------------------|------------------------------|-----------------------|-----|-----|-----|------|
| | Legacy | JEDEC | Min | Max | Min | Max | |
| Address valid to next address valid | t ^{WC} | t ^{AVAV} | 70 | – | 90 | – | ns |
| WE# LOW to CE# LOW | t ^{WS} | t ^{WLEL} | 0 | – | 0 | – | ns |
| CE# LOW to CE# HIGH | t ^{CP} | t ^{ELEH} | 45 | – | 50 | – | ns |
| Input valid to CE# HIGH | t ^{DS} | t ^{DVEH} | 45 | – | 50 | – | ns |
| CE# HIGH to input transition | t ^{DH} | t ^{EHD_X} | 0 | – | 0 | – | ns |
| CE# HIGH to WE# HIGH | t ^{WH} | t ^{EHWH} | 0 | – | 0 | – | ns |
| CE# HIGH to CE# LOW | t ^{CPH} | t ^{EHEL} | 30 | – | 30 | – | ns |
| Address valid to CE# LOW | t ^{AS} | t ^{AVEL} | 0 | – | 0 | – | ns |
| CE# LOW to address transition | t ^{AH} | t ^{ELAX} | 45 | – | 50 | – | ns |
| OE# HIGH to CE# LOW | – | t ^{GHEL} | 0 | – | 0 | – | ns |
| CE# HIGH to OE# LOW | t ^{OE_H} | t ^{EHGL} | 0 | – | 0 | – | ns |
| Program/Erase valid to RY/BY# LOW | t ^{BUSY} | t ^{EHRL} | – | 30 | – | 35 | ns |
| V _{CC} HIGH to WE# LOW | t ^{VCS} | t ^{VCHWL} | 50 | – | 50 | – | μs |

- Notes: 1. 70ns becomes 80ns if the 80ns device code is used.
2. Sampled only; not 100% tested.

Figure 15: CE#-Controlled AC Timing



Program/Erase Characteristics

Table 22: Program/Erase Times and Endurance Cycles

Notes 1 and 2 apply to the entire table

| Parameter | Min | Typ | Max | Unit | Notes |
|----------------------------------|---------|-----|-----|--------|-------|
| Chip erase | – | 29 | 60 | s | 3 |
| Block erase (64KB) | – | 0.8 | 1.6 | s | 4 |
| Erase suspend latency time | – | 20 | 25 | μs | 4 |
| Program (byte or word) | – | 13 | 200 | μs | 3 |
| Chip program (byte by byte) | – | 26 | 120 | s | 3 |
| Chip program (word by word) | – | 13 | 60 | s | 3 |
| PROGRAM/ERASE cycles (per block) | 100,000 | – | – | cycles | |
| Data retention | 20 | – | – | years | |

- Notes:
1. Typical values measured at room temperature and nominal voltages and for not cycled devices.
 2. Sampled, but not 100% tested.
 3. Maximum value measured at worst case conditions for both temperature and V_{CC} after 100,000 PROGRAM/ERASE cycles.
 4. Maximum value measured at worst case conditions for both temperature and V_{CC} .

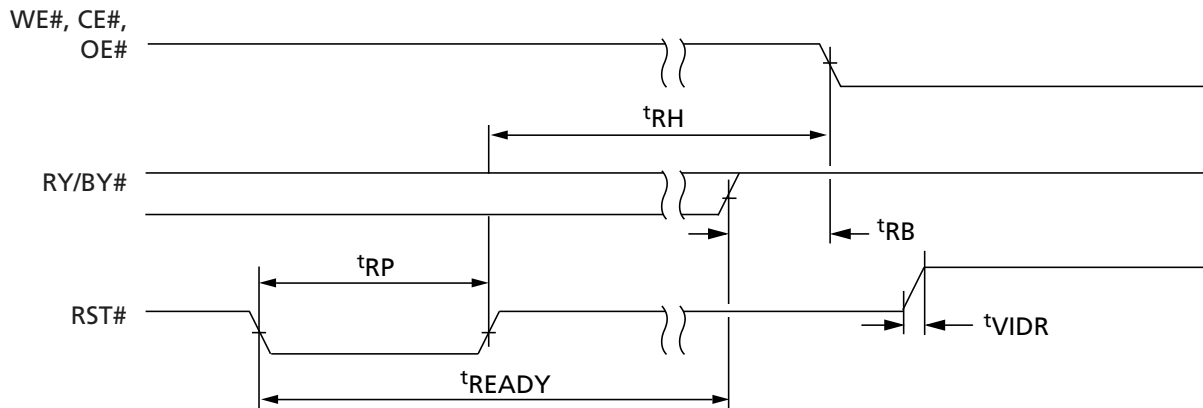
Reset Characteristics

Table 23: Reset/Block Temporary Unprotect AC Characteristics

| Condition/Parameter | | Symbol | | M29W160ET/B | | Unit | Notes |
|--|-----|-------------|--|-------------|-----|---------|-------|
| | | Legacy | JEDEC | 7A/70/80 | 90 | | |
| RST# HIGH to WE# LOW; CE# LOW; OE# LOW | Min | t_{RH} | t_{PHWL} t_{PHEL} t_{PHGL} | 50 | 50 | ns | 1 |
| RY/BY# HIGH to WE# LOW; CE# LOW; OE# LOW | Min | t_{RB} | t_{RHWL} t_{RHEL} t_{RHGL} | 0 | 0 | ns | 1 |
| RST# pulse width | Min | t_{RP} | t_{PLPX} | 500 | 500 | ns | |
| RST# LOW to read mode | Max | t_{READY} | t_{PLYH} | 10 | 10 | μs | 1 |
| RST# rise time to V_{ID} | Min | t_{VIDR} | t_{PHPHH} | 500 | 500 | ns | 1 |

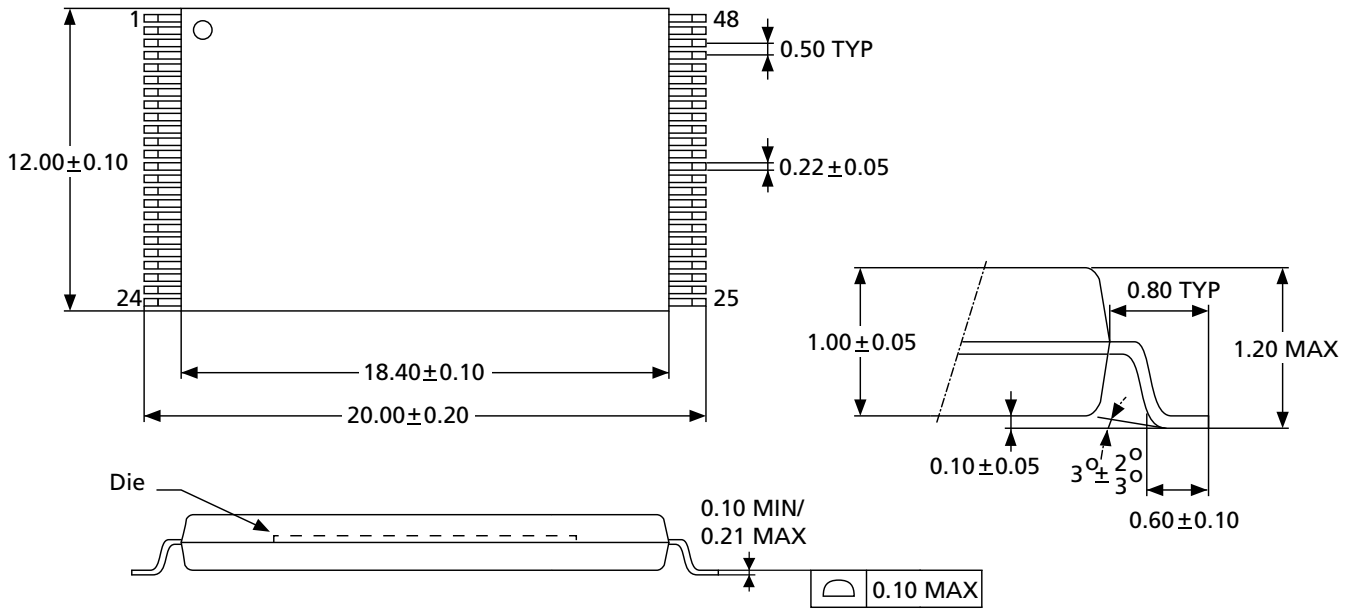
Note: 1. Sampled only; not 100% tested.

Figure 16: Reset/Block Temporary Unprotect AC Waveforms



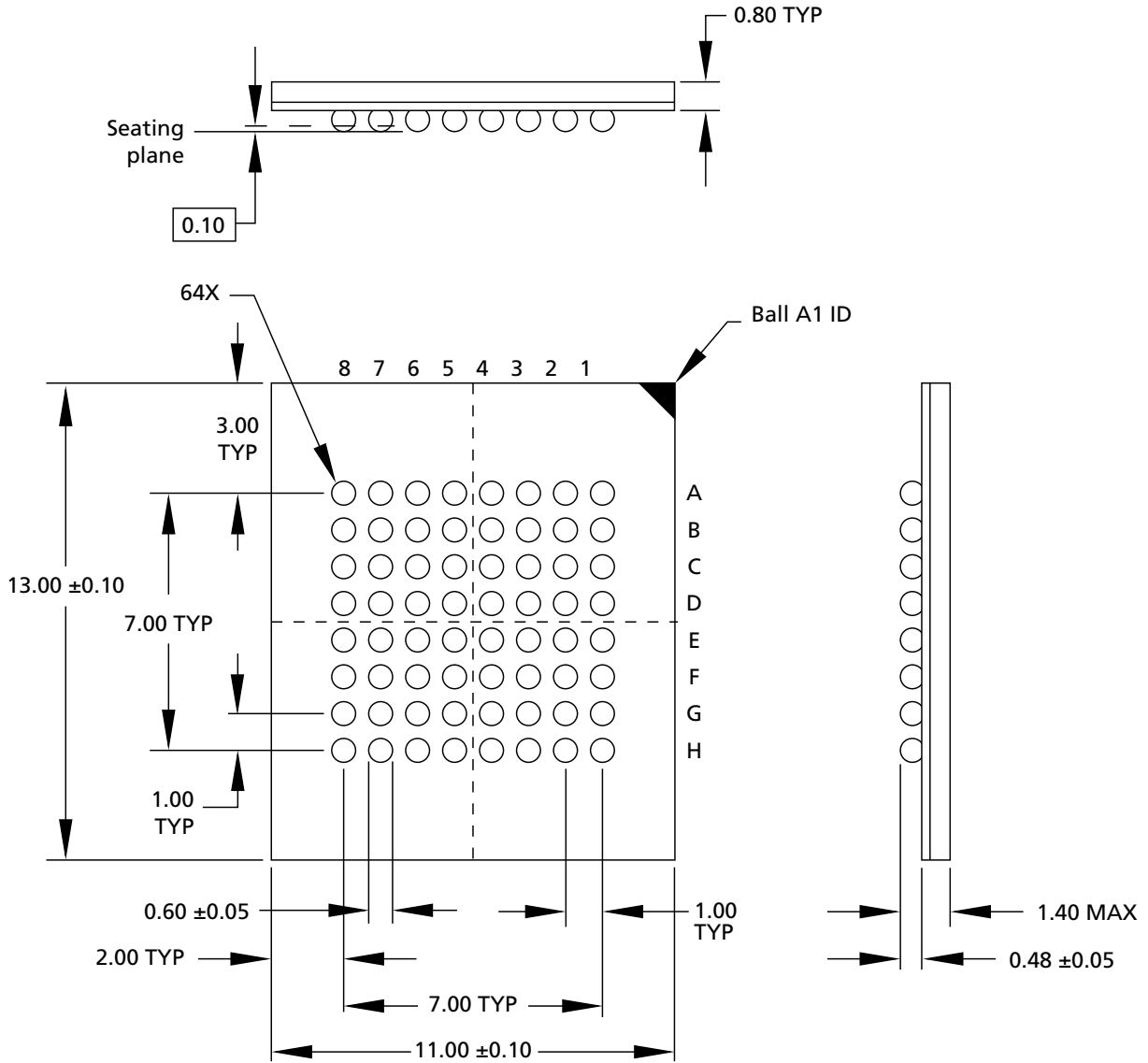
Package Dimensions

Figure 17: 48-Pin TSOP – 12mm x 20mm



Note: 1. All dimensions are in millimeters.

Figure 19: 64-Ball FBGA – 11mm x 13mm



Note: 1. All dimensions are in millimeters.



Revision History

Rev. B – 06/13

- Updates to format, typo fixes

Rev. A – 07/12

- Initial Micron brand release

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
www.micron.com/productsupport Customer Comment Line: 800-932-4992
Micron and the Micron logo are trademarks of Micron Technology, Inc.
All other trademarks are the property of their respective owners.

This data sheet contains minimum and maximum limits specified over the power supply and temperature range set forth herein. Although considered final, these specifications are subject to change, as further product development and data characterization sometimes occur.