

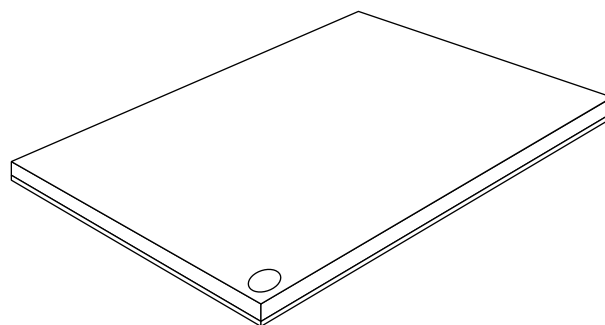
1Gb NAND Flash Memory

MT29F1GxxABB

Features

- Organization
 - Page size x8: 2,112 bytes (2,048 + 64 bytes)
 - Page size x16: 1,056 words (1,024 + 32 words)
 - Block size: 64 pages (128K + 4K bytes)
 - Device size: 1Gb: 1,024 blocks
- READ performance
 - Random READ: 25µs (MAX)
 - Sequential READ: 50ns (MIN)
- WRITE performance
 - PROGRAM PAGE: 250µs (TYP)
 - BLOCK ERASE: 2.0ms (TYP)
- Endurance: 100,000 PROGRAM/ERASE cycles
- Data retention: 10 years
- The first block (block address 00h) is guaranteed to be valid without ECC (up to 1,000 PROGRAM/ERASE cycles)
- VCC: 1.65–1.95V
- Automated PROGRAM and ERASE
- Basic NAND Flash command set
 - PAGE READ, RANDOM DATA READ, READ ID, READ STATUS, PROGRAM PAGE, RANDOM DATA INPUT, PROGRAM PAGE CACHE MODE, INTERNAL DATA MOVE, INTERNAL DATA MOVE with RANDOM DATA INPUT, BLOCK ERASE, RESET
- New commands
 - PAGE READ CACHE MODE
 - READ ID2 (contact factory)
 - READ UNIQUE ID (contact factory)
 - Programmable I/O
 - OTP
 - BLOCK LOCK
- Operation status byte: Provides a software method for detecting:
 - Operation completion
 - Pass/fail condition
 - Write-protect status
- Ready/busy# signal (R/B#): Provides a hardware method of detecting operation completion
- LOCK signal: Protects selectable ranges of blocks

Figure 1: 63-Ball VFBGA x8



- WP# signal: Write-protects the entire device
- Reset required after power-up

Options¹

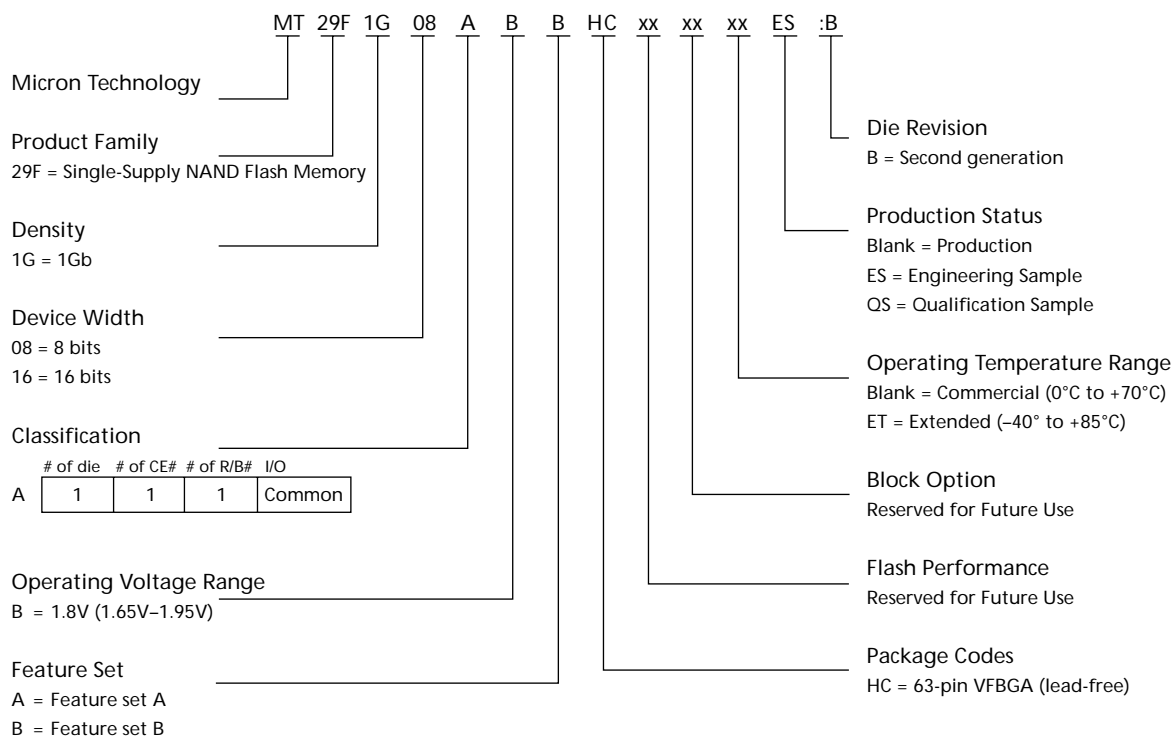
- Configuration
 - x8
 - x16
- Package
 - 63-ball VFBGA
 - 13mm x 10.5mm x 1.0mm
- Operating temperature
 - Commercial temperature (0 to +70°C)
 - Extended temperature (–40°C to +85°C)

Notes: 1. For part numbers and device markings, see Figure 2 on page 2.

Part Numbering Information

Micron NAND Flash devices are available in several configurations and densities.

Figure 2: Part Number Chart



Valid Part Number Combinations

After building the part number from the part numbering chart, verify that the part number is offered and valid by using the Micron Parametric Part Search Web site at www.micron.com/products/parametric. If the device required is not on this list, contact the factory.

Table of Contents

| | |
|---|----|
| Features | 4 |
| Part Numbering Information | 5 |
| Valid Part Number Combinations | 5 |
| General Description | 10 |
| Architecture | 14 |
| Addressing | 14 |
| Memory Mapping | 17 |
| Bus Operation | 19 |
| Control Signals | 19 |
| Commands | 19 |
| Address Input | 19 |
| Data Input | 20 |
| READ | 20 |
| READY/BUSY# | 20 |
| Command Definitions | 23 |
| READ Operations | 24 |
| PAGE READ 00h-30h | 24 |
| RANDOM READ 05h-E0h | 25 |
| PAGE READ CACHE MODE START 31h; PAGE READ CACHE MODE START LAST 3Fh | 25 |
| READ ID 90h | 27 |
| ONFI READ ID | 29 |
| ONFI READ PARAMETER PAGE Operation | 30 |
| READ STATUS 70h | 31 |
| PROGRAM Operations | 33 |
| PROGRAM PAGE 80h-10h | 33 |
| SERIAL DATA INPUT 80h | 33 |
| RANDOM DATA INPUT 85h | 33 |
| PROGRAM PAGE CACHE MODE 80h-15h | 34 |
| INTERNAL DATA MOVE Operations | 36 |
| READ FOR INTERNAL DATA MOVE 00h-35h | 36 |
| INTERNAL DATA MOVE 85h-10h | 36 |
| BLOCK ERASE 60h-D0h | 37 |
| One-Time Programmable (OTP) Area | 38 |
| OTP DATA PROGRAM A0h-10h | 38 |
| OTP DATA PROTECT A5h-10h | 40 |
| OTP DATA READ AFh-30h | 41 |
| BLOCK LOCK Feature | 42 |
| WP# and BLOCK LOCK | 42 |
| UNLOCK 23h-24h | 42 |
| LOCK 2Ah | 44 |
| LOCK-TIGHT 2Ch | 45 |
| BLOCK LOCK READ STATUS 7Ah | 46 |
| RESET Operation | 49 |
| RESET FFh | 49 |
| Programmable Drive Strength | 50 |
| PROGRAMMABLE I/O DRIVE STRENGTH B8h | 50 |
| WRITE PROTECT | 51 |
| Error Management | 53 |
| Electrical Characteristics | 54 |
| VCC Power Cycling | 55 |
| Timing Diagrams | 59 |
| Package Dimensions | 72 |

List of Figures

| | | |
|------------|--|----|
| Figure 1: | 63-Ball VFBGA x8 | 4 |
| Figure 2: | Part Number Chart | 5 |
| Figure 3: | Functional Block Diagram: 1Gb NAND Flash | 10 |
| Figure 4: | Ball Assignment (x8), 63-Ball VFBGA | 11 |
| Figure 5: | Ball Assignment (x16), 63-Ball VFBGA | 12 |
| Figure 6: | Array Organization for MT29F1G08 (x8) | 15 |
| Figure 7: | Array Organization for MT29F1G16 (x16) | 16 |
| Figure 8: | Memory Map (x8) | 17 |
| Figure 9: | Memory Map (x16) | 18 |
| Figure 10: | READY/BUSY# Open Drain | 21 |
| Figure 11: | tFall and tRise | 21 |
| Figure 12: | Iol vs. Rp | 21 |
| Figure 13: | TC vs. Rp | 22 |
| Figure 14: | PAGE READ Operation | 24 |
| Figure 15: | RANDOM DATA READ Operation | 25 |
| Figure 16: | PAGE READ CACHE MODE | 26 |
| Figure 17: | READ ID Operation | 27 |
| Figure 18: | ONFI READ ID Operation | 29 |
| Figure 19: | ONFI READ PARAMETER PAGE Operation | 30 |
| Figure 20: | Status Register Operation | 32 |
| Figure 21: | PROGRAM and READ STATUS Operation | 33 |
| Figure 22: | RANDOM DATA INPUT | 34 |
| Figure 23: | PROGRAM PAGE CACHE MODE Example | 35 |
| Figure 24: | INTERNAL DATA MOVE | 37 |
| Figure 25: | INTERNAL DATA MOVE with RANDOM DATA INPUT | 37 |
| Figure 26: | BLOCK ERASE Operation | 37 |
| Figure 27: | OTP DATA PROGRAM | 39 |
| Figure 28: | OTP DATA PROTECT | 40 |
| Figure 29: | OTP DATA READ Operation | 41 |
| Figure 30: | Flash Array Protected: Inverted Area Bit = 0 | 43 |
| Figure 31: | Flash Array Protected: Invert Area Bit = 1 | 43 |
| Figure 32: | UNLOCK Operation | 44 |
| Figure 33: | LOCK Operation | 45 |
| Figure 34: | LOCK-TIGHT Operation | 45 |
| Figure 35: | PROGRAM/ERASE Issued to Locked or Locked-Tight Block | 46 |
| Figure 36: | LOCKED-TIGHT BLOCKS to LOCKED BLOCKS Operation | 46 |
| Figure 37: | BLOCK LOCK READ STATUS | 47 |
| Figure 38: | BLOCK LOCK Flow Chart | 48 |
| Figure 39: | RESET Operation | 49 |
| Figure 40: | Programmable I/O Drive Strength Command Sequence | 50 |
| Figure 41: | ERASE Enable | 51 |
| Figure 42: | ERASE Disable | 51 |
| Figure 43: | PROGRAM Enable | 52 |
| Figure 44: | PROGRAM Disable | 52 |
| Figure 45: | AC Waveforms During Power Transitions | 55 |
| Figure 46: | COMMAND LATCH Cycle | 59 |
| Figure 47: | ADDRESS LATCH Cycle | 59 |
| Figure 48: | INPUT DATA LATCH Cycle | 60 |
| Figure 49: | SERIAL ACCESS Cycle after READ | 60 |
| Figure 50: | READ STATUS Cycle | 61 |
| Figure 51: | PAGE READ Operation | 61 |
| Figure 52: | READ Operation with CE# "Don't Care" | 62 |
| Figure 53: | RANDOM DATA READ Operation | 62 |
| Figure 54: | PAGE READ CACHE MODE Operation, Part 1 of 2 | 63 |
| Figure 55: | PAGE READ CACHE MODE Operation, Part 2 of 2 | 64 |
| Figure 56: | PAGE READ CACHE MODE Operation without R/B#, Part 1 of 2 | 65 |

| | | |
|------------|--|----|
| Figure 57: | PAGE READ CACHE MODE Operation without R/B#, Part 2 of 2 | 66 |
| Figure 58: | READ ID Operation | 67 |
| Figure 59: | PROGRAM Operation with CE# "Don't Care" | 67 |
| Figure 60: | PROGRAM PAGE Operation | 68 |
| Figure 61: | PROGRAM PAGE Operation with RANDOM DATA INPUT | 68 |
| Figure 62: | INTERNAL DATA MOVE Operation | 69 |
| Figure 63: | PROGRAM PAGE CACHE MODE Operation | 69 |
| Figure 64: | PROGRAM PAGE CACHE MODE Operation Ending on 15h | 70 |
| Figure 65: | BLOCK ERASE Operation | 71 |
| Figure 66: | RESET Operation | 71 |
| Figure 67: | 63-Ball VFBGA Package | 72 |

List of Tables

| | | |
|-----------|--|----|
| Table 1: | Ball Descriptions | 13 |
| Table 2: | Array Addressing: MT29F1G08 (x8) | 15 |
| Table 3: | Array Addressing: MT29F1G16 (x16) | 16 |
| Table 4: | Operational Example (x8) | 17 |
| Table 5: | Operational Example (x16) | 18 |
| Table 6: | Mode Selection | 22 |
| Table 7: | Command Set | 23 |
| Table 8: | Device ID and Configuration Codes | 28 |
| Table 9: | Status Register Bit Definition | 32 |
| Table 10: | BLOCK LOCK Address Cycle Assignments | 44 |
| Table 11: | BLOCK LOCK Status Register Bit Definitions | 46 |
| Table 12: | Status Register Contents After Reset | 49 |
| Table 13: | I/O Drive Strength Settings | 50 |
| Table 14: | Programmable I/O Drive Strength Register READ/WRITE Timing | 51 |
| Table 15: | Absolute Maximum Ratings by Device | 54 |
| Table 16: | Recommended Operating Conditions | 54 |
| Table 17: | DC and Operating Characteristics, VCC = 1.65–1.95V | 56 |
| Table 18: | Valid Blocks | 56 |
| Table 19: | Capacitance | 56 |
| Table 20: | Test Conditions | 56 |
| Table 21: | AC Characteristics – Command, Data, and Address Input | 57 |
| Table 22: | AC Characteristics – Normal Operation | 57 |
| Table 23: | PROGRAM/ERASE Characteristics | 58 |

General Description

The MT29F1G08 and MT29F1G16 are both 1Gb NAND Flash memory devices. NAND Flash technology provides a cost-effective solution for applications requiring high-density, solid-state storage. The MT29F1Gxx devices include standard NAND Flash features as well as new features designed to enhance system-level performance.

The MT29F1Gxx devices use a multiplexed 8- or 16-bit bus (I/O[7:0] or I/O[15:0]) to transfer data, address, and instruction information. The five control signals (CLE, ALE, CE#, RE#, WE#) control the NAND Flash command bus interface protocol. Additional signals control hardware write protection (WP#), monitor the device ready/busy (R/B#) state, and enable BLOCK LOCK functions (LOCK).

This hardware interface creates a low-ball-count device with a standard ball arrangement that is the same from one density to another, enabling future upgrades to higher densities without any board redesign.

MT29F1Gxx devices contain 1,024 erasable blocks. Each block is subdivided into 64 programmable pages. Each page consists of 2,112 bytes (x8), or 1,056 words (x16). The pages are further divided into a 2,048-byte data storage region with a separate 64-byte area on the x8 device; and on the x16 device, separate 1,024-word and 32-word areas. The 64-byte and 32-word areas are typically used for error correction functions.

On-chip control logic automates PROGRAM and ERASE operations to maximize cycle endurance. PROGRAM/ERASE endurance is specified at 100,000 cycles when using appropriate error correction code (ECC) and bad-block-management software.

Figure 1: Functional Block Diagram: 1Gb NAND Flash

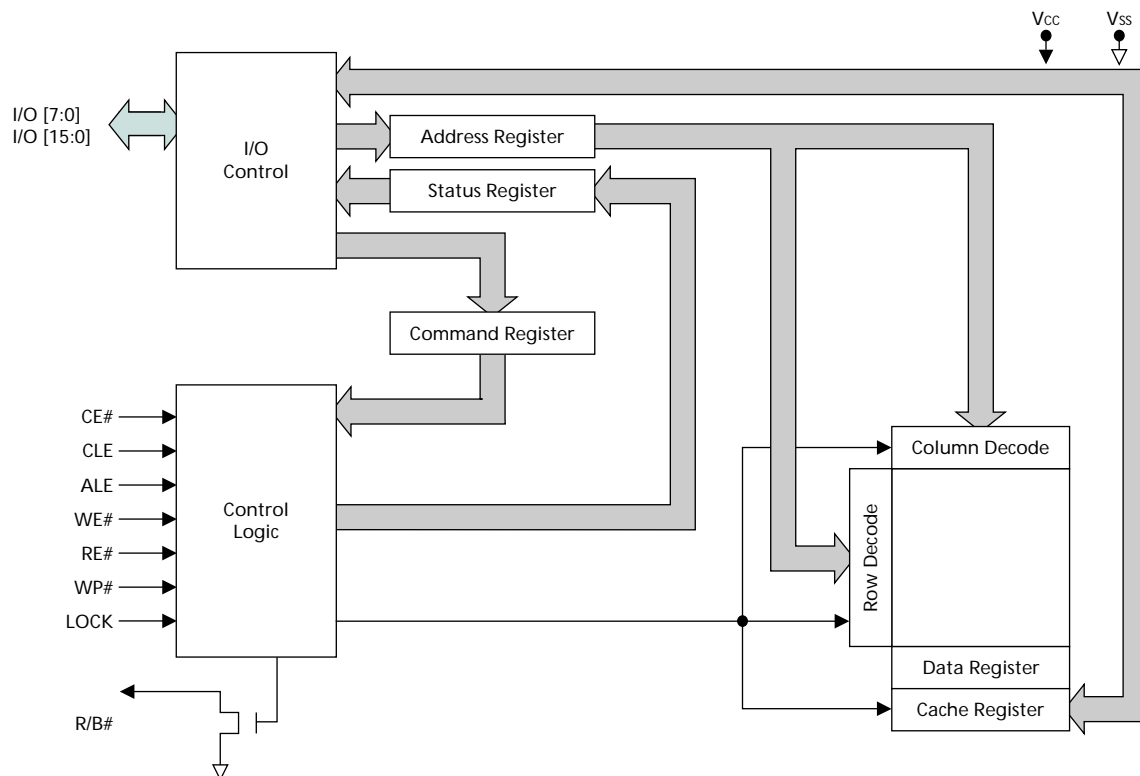
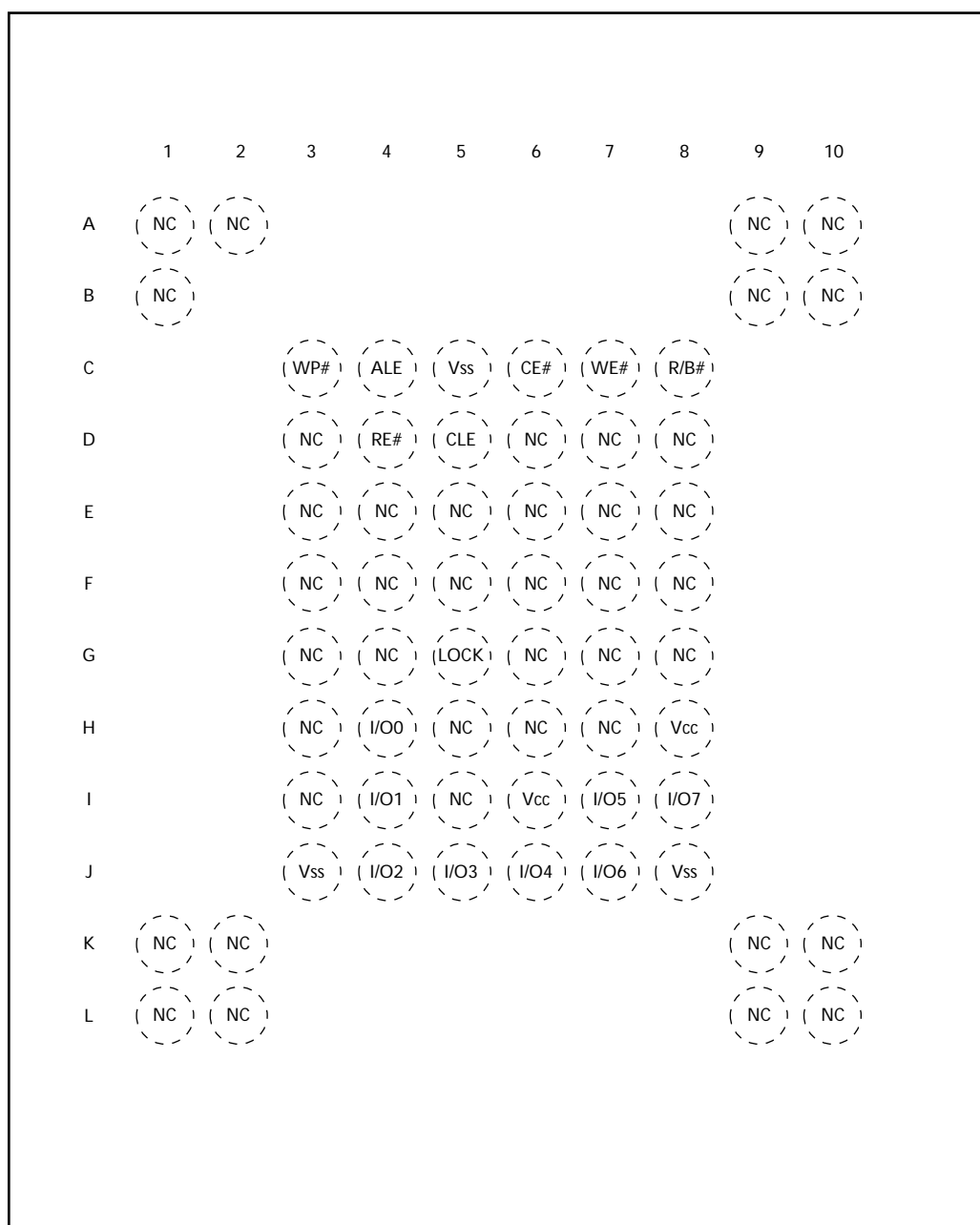


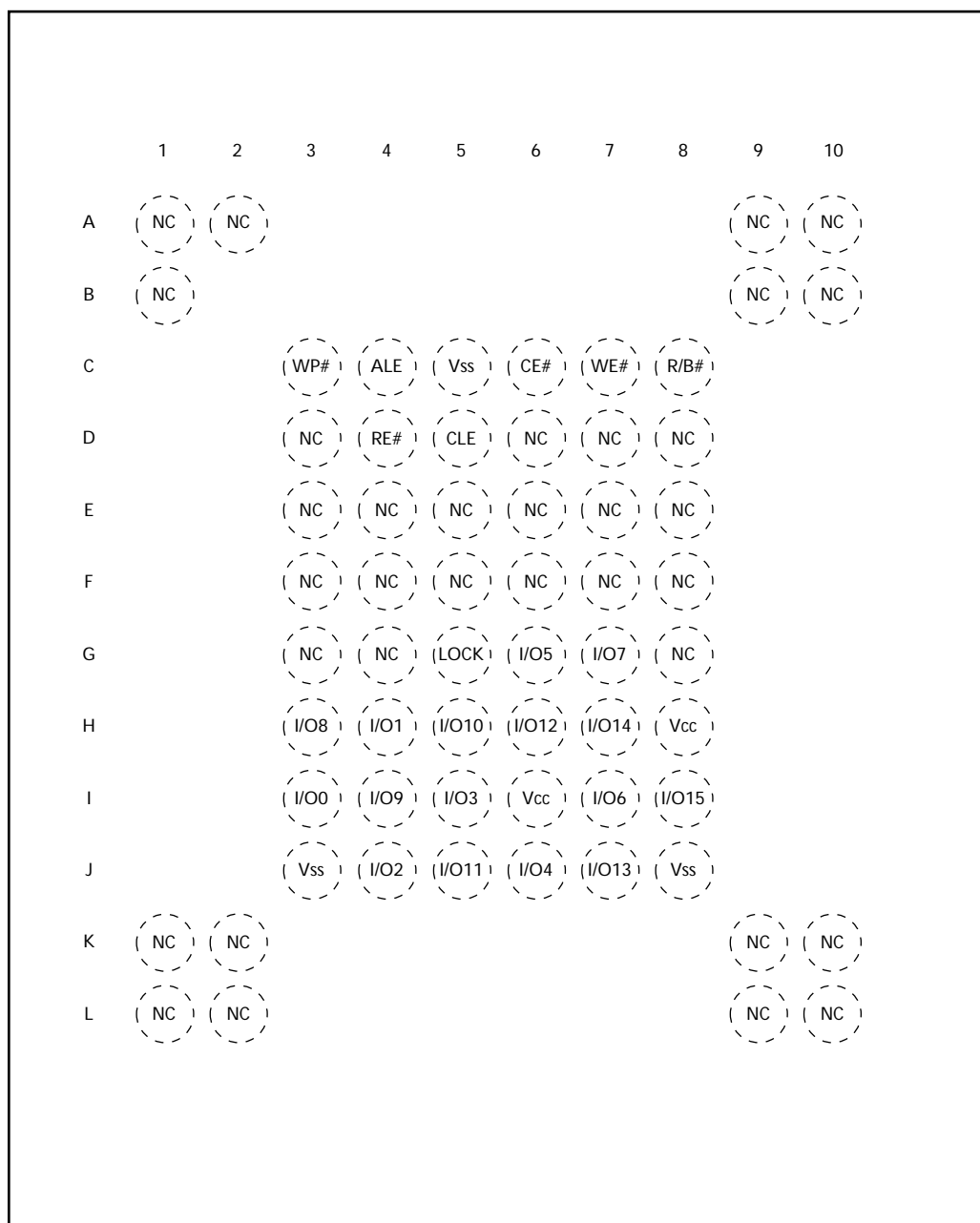
Figure 2: Ball Assignment (x8), 63-Ball VFBGA



Top View, Ball Down

Notes: 1. For package dimensions, see Figure 65 on page 69.

Figure 3: Ball Assignment (x16), 63-Ball VFBGA



Top View, Ball Down

Notes: 1. For package dimensions, see Figure 65 on page 69.

Table 1: Ball Descriptions

| Symbol | Type | Ball Function |
|--|--------|--|
| ALE | Input | Address latch enable: During the time ALE is HIGH, address information is transferred from I/O[7:0] into the on-chip address register. Upon a LOW to HIGH transition on WE#—when address information is not being loaded—the ALE signals should be driven LOW. |
| CE# | Input | Chip enable: Gates transfers between the host system and the NAND Flash device. After the device becomes busy or starts a PROGRAM or ERASE operation, CE# can be de-asserted. See “Bus Operation” on page 16 for additional operational details. |
| CLE | Input | Command latch enable: When CLE is HIGH, information is transferred from I/O [7:0] to the on-chip command register on the rising edge of WE#. When command information is not being loaded, the CLE signals should be driven LOW. |
| LOCK | Input | When LOCK is HIGH during power-up, the BLOCK LOCK function is enabled. To disable BLOCK LOCK, connect LOCK to Vss during power-up, or leave it unconnected (internal pull-down). |
| RE# | Input | Read enable: Gates transfers from the NAND Flash device to the host system. |
| WE# | Input | Write enable: Gates transfers from the host system to the NAND Flash device. |
| WP# | Input | Write protect: Protects against inadvertent PROGRAM and ERASE operations. All PROGRAM and ERASE operations are disabled when WP# is LOW. |
| I/O[7:0] (x8) I/O[15:0] (x16) | I/O | Data inputs/outputs: Bidirectional I/O signals transfer address, data and instruction information. Data is output only during READ operations; at other times the I/O signals are inputs. |
| R/B# | Output | Ready/busy: The ready/busy signal is an open-drain, active-LOW output, that uses an external pull-up resistor. The signal is used to indicate when the chip is processing a PROGRAM or ERASE operation. The signal is also used during READ operations to indicate when data is being transferred from the array into the serial data register. When these operations have completed, the ready/busy signal returns to the high-impedance state. |
| Vcc | Supply | Vcc: The Vcc ball is the power supply. |
| Vss | Supply | Vss: The Vss ball is the ground connection. |
| NC | – | No connect: NC balls are not internally connected. These balls can be driven or left unconnected. |

Architecture

The MT29F1G08 and MT29F1G16 use NAND Flash electrical and command interfaces. Data, commands, and addresses are multiplexed onto the same signals. This provides a memory device with a low ball count.

The internal memory array is accessed on a page basis. When performing READs, a page of data is copied from the memory array into the data register. Once copied to the data register, data is output sequentially, byte by byte on the x8 device, or word by word on the x16 device.

The memory array is programmed on a page basis. After the starting address is loaded into the internal address register, data is sequentially written to the internal data register up to the end of a page. After all page data has been loaded into the data register, array programming is started.

In order to increase programming bandwidth, this device incorporates a cache register. In the cache programming mode, data is first copied into the cache register and then into the data register. Once the data is copied into the data register, programming begins. After the data register has been loaded and programming has started, the cache register becomes available for loading additional data. Loading the next page of data into the cache register takes place while page programming is in process.

The INTERNAL DATA MOVE command also uses the internal cache register. Normally, moving data from one area of external memory to another uses a large number of external memory cycles. By using the internal cache register and data register, array data can be copied from one page and then programmed into another without using external memory cycles.

Addressing

The MT29F1G08 and MT29F1G16 devices do not have dedicated address balls. Addresses are loaded using a 4-cycle sequence as shown in Tables 2 and 3 on pages 12 and 13. Table 2 presents address functions internal to the MT29F1G08 device; Table 3 presents address functions internal to the MT29F1G16. See Figures 6 and 7 on pages 14 and 15 for additional memory mapping and addressing details.

Figure 4: Array Organization for MT29F1G08 (x8)

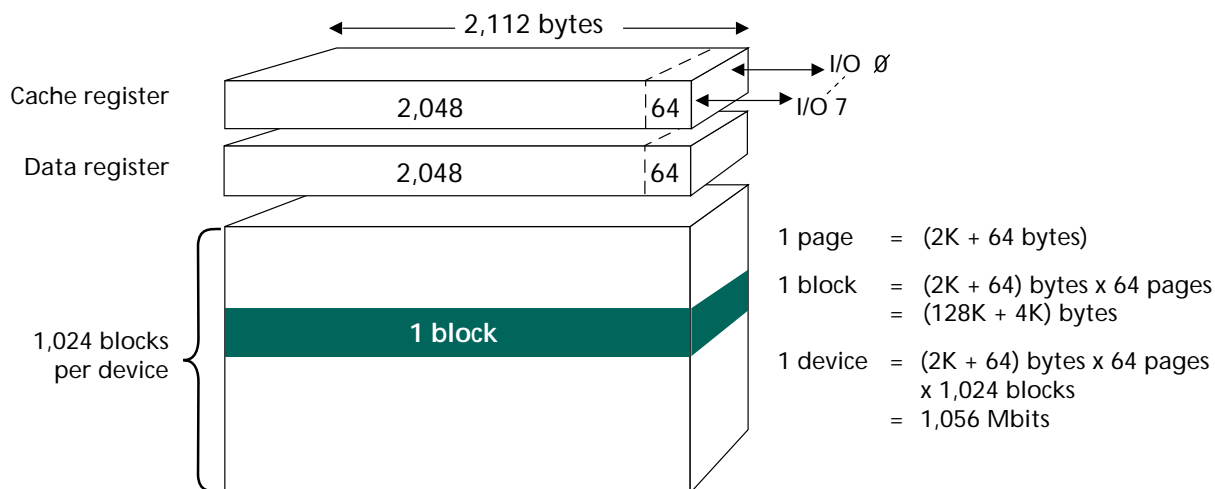


Table 2: Array Addressing: MT29F1G08 (x8)

| Cycle | I/O7 | I/O6 | I/O5 | I/O4 | I/O3 | I/O2 | I/O1 | I/O0 |
|--------|------|------|------|------|------|------|------|------|
| First | CA7 | CA6 | CA5 | CA4 | CA3 | CA2 | CA1 | CA0 |
| Second | LOW | LOW | LOW | LOW | CA11 | CA10 | CA9 | CA8 |
| Third | BA7 | BA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| Fourth | BA15 | BA14 | BA13 | BA12 | BA11 | BA10 | BA9 | BA8 |

- Notes:
1. Block address concatenated with page address = actual page address. CAx = column address; PAx = page address; BAx = block address.
 2. Note that the 12-bit column address is capable of addressing from 0 to 4,095 bytes on a x8 device; however, only bytes 0 through 2,111 are valid. Bytes 2,112 through 4,095 of each page are "out of bounds," do not exist in the device, and cannot be addressed.

Figure 5: Array Organization for MT29F1G16 (x16)

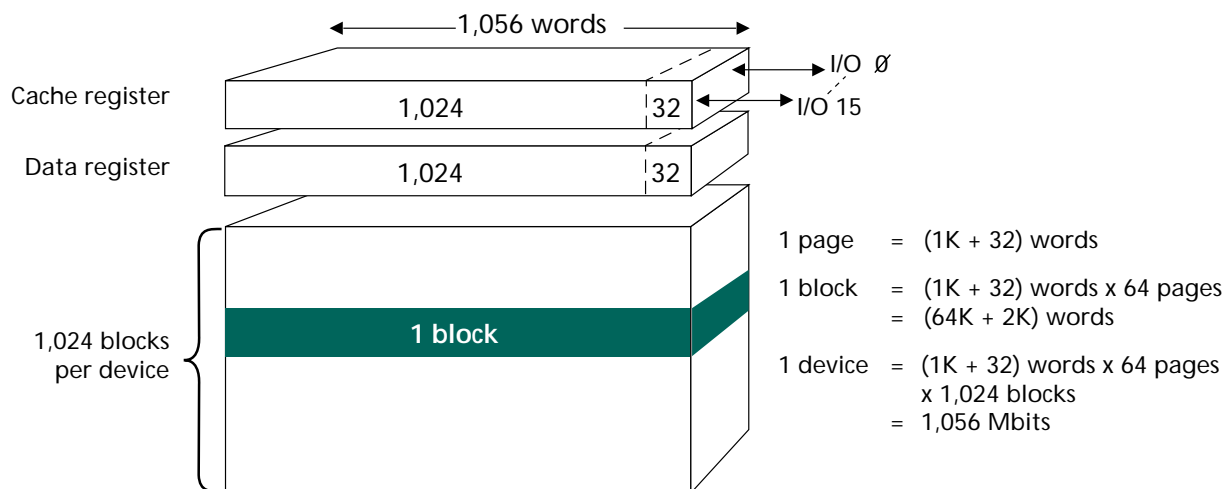


Table 3: Array Addressing: MT29F1G16 (x16)

| Cycle | I/O[15:8] | I/O7 | I/O6 | I/O5 | I/O4 | I/O3 | I/O2 | I/O1 | I/O0 |
|--------|-----------|------|------|------|------|------|------|------|------|
| First | LOW | CA7 | CA6 | CA5 | CA4 | CA3 | CA2 | CA1 | CA0 |
| Second | LOW | LOW | LOW | LOW | LOW | LOW | CA10 | CA9 | CA8 |
| Third | LOW | BA7 | BA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| Fourth | LOW | BA15 | BA14 | BA13 | BA12 | BA11 | BA10 | BA9 | BA8 |

- Notes:
1. Block address concatenated with page address = actual page address. CAx = column address; PAx = page address; BAx = block address.
 2. I/O[15:8] are not used during addressing sequence and should be driven LOW.
 3. Note that the 11-bit column address is capable of addressing from 0 to 2,047 words on a x16 device; however, only words 0 through 1,055 are valid. Words 1,056 through 2,047 of each page are "out of bounds," do not exist in the device, and cannot be addressed.

Memory Mapping

Figure 6: Memory Map (x8)

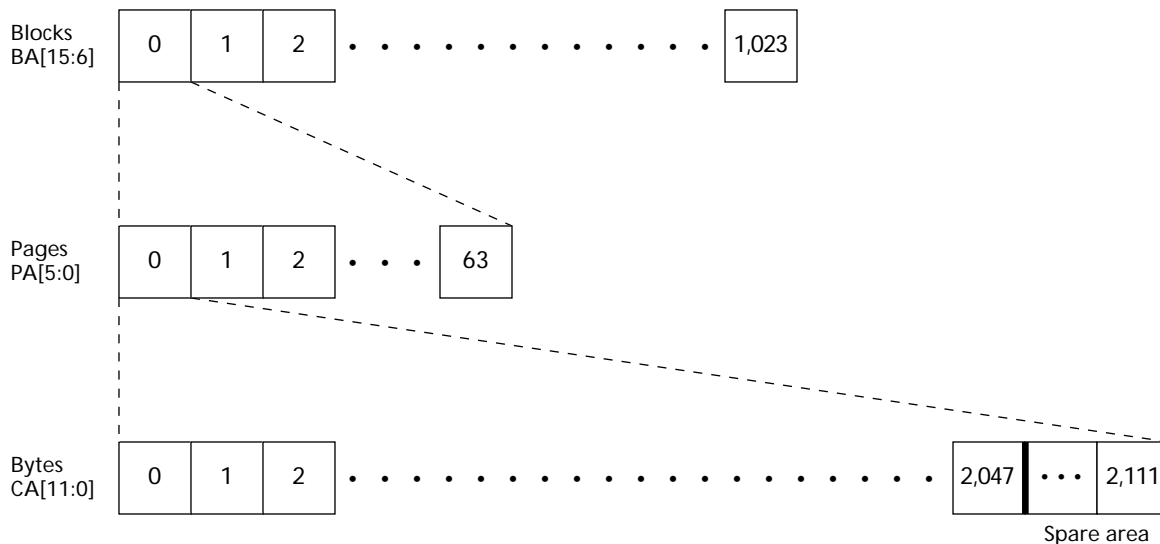


Table 4: Operational Example (x8)

| Block | Page | Min Address in Page | Max Address in Page | Out of Bounds Addresses in Page |
|-------|------|---------------------|---------------------|---------------------------------|
| 0 | 0 | 0x00000000 | 0x0000083F | 0x00000840–0x00000FFF |
| 0 | 1 | 0x00010000 | 0x0001083F | 0x00010840–0x00010FFF |
| 0 | 2 | 0x00020000 | 0x0002083F | 0x00020840–0x00020FFF |
| ... | ... | ... | ... | ... |
| 1,023 | 62 | 0xFFFE0000 | 0xFFFE083F | 0xFFFE0840–0xFFFE0FFF |
| 1,023 | 63 | 0xFFFF0000 | 0xFFFF083F | 0xFFFF0840–0xFFFF0FFF |

- Notes:
- As shown in Table 2 on page 12, the high 4 bits of the second ADDRESS cycle have no assigned address bits. However, these 4 bits must be held LOW during the ADDRESS cycle to ensure that the address is interpreted correctly by the NAND Flash device. These extra bits are accounted for in the second ADDRESS cycle even though they have no address bits assigned to them.
 - Note that the 12-bit column address is capable of addressing from 0 to 4,095 bytes on a x8 device; however, only bytes 0 through 2,111 are valid. Bytes 2,112 through 4,095 of each page are "out of bounds," do not exist in the device, and cannot be addressed.

Figure 7: Memory Map (x16)

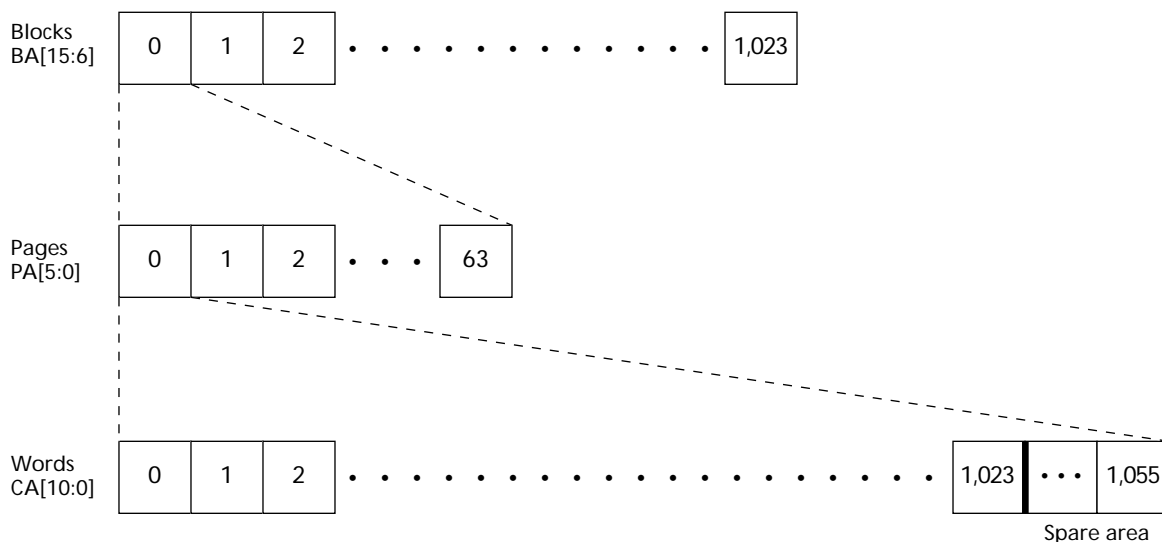


Table 5: Operational Example (x16)

| Block | Page | Min Address in Page | Max Address in Page | Out of Bounds Addresses in Page |
|-------|------|---------------------|---------------------|---------------------------------|
| 0 | 0 | 0x00000000 | 0x0000041F | 0x00000420–0x00000FFF |
| 0 | 1 | 0x00010000 | 0x0001041F | 0x00010420–0x00010FFF |
| 0 | 2 | 0x00020000 | 0x0002041F | 0x00020420–0x00020FFF |
| ... | ... | ... | ... | ... |
| 1,023 | 62 | 0xFFFFE0000 | 0xFFFFE041F | 0xFFFFE0420–0x00020FFF |
| 1,023 | 63 | 0xFFFFF0000 | 0xFFFFF041F | 0xFFFFF0420–0xFFFFFOFFF |

- Notes:
- As shown in Table 3 on page 13, the high 5 bits of ADDRESS cycle 2 have no assigned address bits. However, these 5 bits must be held LOW during the ADDRESS cycle to ensure that the address is interpreted correctly by the NAND Flash device. These extra bits are accounted for in ADDRESS cycle 2 even though they have no address bits assigned to them.
 - Note that the 11-bit column address is capable of addressing from 0 to 2,047 words on a x16 device; however, only words 0 through 1,055 are valid. Words 1,056 through 2,047 of each page are "out of bounds," do not exist in the device, and cannot be addressed.

Bus Operation

The bus on the MT29F1Gxx devices is multiplexed. Data I/O, addresses and commands all share the same balls. I/O[15:8] are used only for data in the x16 configuration. Addresses and commands are always supplied on I/O[7:0].

The command sequence normally consists of a COMMAND LATCH cycle, ADDRESS LATCH cycle and a DATA cycle—either READ or WRITE.

Control Signals

CE#, WE#, RE#, CLE, ALE, LOCK, and WP control NAND Flash READ and WRITE operations.

CE# is used to enable the device. When CE# is LOW and the device is not in the BUSY state, the NAND Flash memory will accept command, data, and address information.

When the device is not performing an operation, CE# is typically driven HIGH and the device enters standby mode. The memory will enter standby if CE# goes HIGH while data is being transferred and the device is not busy. This helps reduce power consumption (see Figure 57 on page 64).

The CE# “Don’t Care” operation enables the NAND Flash to reside on the same asynchronous memory bus as other Flash or SRAM devices. Other devices on the memory bus can then be accessed while the NAND Flash is busy with internal operations. This capability is important for designs that require multiple NAND Flash devices on the same bus. One device can be programmed while another is being read.

A HIGH CLE signal indicates that a COMMAND cycle is taking place. A HIGH ALE signal signifies that an ADDRESS INPUT cycle is occurring.

Commands

Commands are written to the command register on the rising edge of WE# when all of these conditions are met:

- CE# and ALE are LOW
- CLE is HIGH
- the device is not busy

The READ STATUS and RESET commands are different because they can be written to the device while it is busy. Commands are transferred to the command register on the rising edge of WE# (see Figure 26 on page 37).

Commands are input on I/O[7:0] only. For devices with a x16 interface, I/O[15:8] must be written with zeros when issuing a command.

Address Input

Addresses are written to the address register on the rising edge of WE# when all of these conditions are met:

- CE# and CLE are LOW
- ALE is HIGH

Addresses are input on I/O[7:0] only. For devices with a x16 interface, I/O[15:8] must be written with zeros when issuing an address.

Generally, all 4 address cycles are written to the device. An exception is the BLOCK ERASE command, which requires only 2 address cycles (see “BLOCK ERASE Operation” on page 33 for details).

RANDOM DATA INPUT and OUTPUT commands need only column addresses, so only 2 address cycles are required. Refer to the command descriptions to determine the addressing requirements for each command.

Data Input

Data is written to the data register on the rising edge of WE# when these conditions are met:

- CE#, CLE, and ALE are LOW
- the device is not busy

Data is input on I/O[7:0] for x8 devices, and I/O[15:0] on x16 devices. See Figure 46 on page 57 for additional data input details.

READ

After a READ command is issued, data is transferred from the memory array to the data register on the rising edge of WE#. R/B# goes LOW for t_R and transitions HIGH after the transfer is complete. When data is available in the data register, it is clocked out of the part by RE# going LOW (see Figure 12 on page 21 for timing details).

The READ STATUS (70h) command or the READY/BUSY signal can be used to determine when the device is ready (see the READ STATUS command section starting on page 28 for details).

READY/BUSY#

The R/B# output provides a hardware method of indicating the completion of a PROGRAM/ERASE/READ operation. The signal is typically HIGH, and transitions to LOW after the appropriate command is written to the device. The signal's open-drain driver enables multiple R/B# outputs to be OR-tied. The signal requires a pull-up resistor for proper operation. The READ STATUS command can be used in place of R/B#. Typically, R/B# would be connected to an interrupt ball on the system controller (see Figure 8 on page 18).

The combination of R_p and the capacitive loading of the R/B# circuit determine the R/B# rise time. The actual value used for R_p depends on the system timing requirements. Large R_p values delay R/B# significantly. At the 10 percent/90 percent points on the R/B# waveform, rise time is approximately two time constants (TC).

$$TC = R \times C$$

Where $R = R_p$ (resistance of pull-up resistor), and $C =$ total capacitive load.

The R/B# fall time is determined mainly by the output impedance of R/B# and the total load capacitance. Refer to Figures 9 and 10 on page 18, which depict approximate R_p values using a circuit load of 100pF.

The minimum value for R_p is determined by the R/B# output drive capability, the output voltage swing, and VCC.

$$R_p(\text{MIN}, 1.8\text{V part}) = \frac{V_{CC}(\text{MAX}) - V_{OL}(\text{MAX})}{I_{OL} + \Sigma I_L} = \frac{1.85\text{V}}{3\text{mA} + \Sigma I_L}$$

Where ΣI_L is the sum of the input currents of all devices tied to the R/B# pin.

Figure 8: READY/BUSY# Open Drain

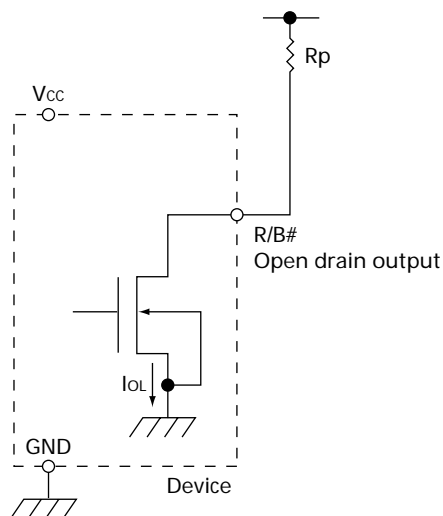
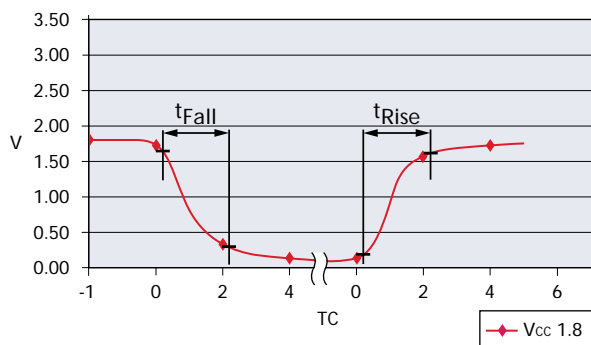
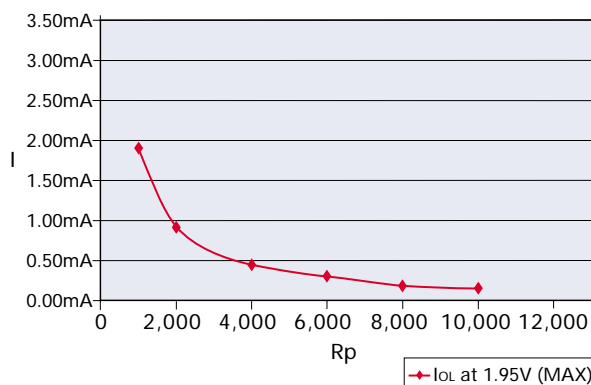


Figure 9: t_{Fall} and t_{Rise}

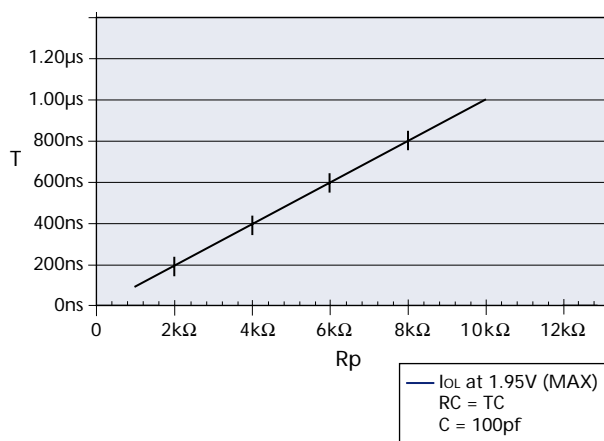








- Notes:
- t_{Fall} and t_{Rise} are calculated at 10 percent and 90 percent points.
 - t_{Rise} is primarily dependent on external pull-up resistor and external capacitive loading.
 - $t_{Fall} \approx 7\text{ns}$ at 1.8V.
 - See TC values in Figure 11 on page 19 for approximate R_p value and TC.

Figure 10: I_{OL} vs. R_p



Note: To calculate R_p value, see page 17.

Figure 11: TC vs. Rp

Table 6: Mode Selection

| CLE | ALE | CE# | WE# | RE# | WP# | Mode | |
|-----|-----|-----|---|---|---------------------|---------------------------------|---------------|
| H | L | L |  | H | X | Read mode | Command input |
| L | H | L |  | H | X | | Address input |
| H | L | L |  | H | H | Write mode | Command input |
| L | H | L |  | H | H | | Address input |
| L | L | L |  | H | H | Data input | |
| L | L | L | H |  | X | Sequential read and data output | |
| L | L | L | H | H | X | During READ (busy) | |
| X | X | X | X | X | H | During PROGRAM (busy) | |
| X | X | X | X | X | H | During ERASE (busy) | |
| X | X | X | X | X | L | Write protect | |
| X | X | H | X | X | 0V/Vcc ² | Standby | |

Notes: 1. Mode selection settings for this table:

H = Logic level HIGH

L = Logic level LOW

X = VIH or VIL

2. WP# should be biased to CMOS HIGH or LOW for standby.

Command Definitions

Table 7: Command Set

| Command | First Cycle | Second Cycle | Valid During Busy | Notes |
|--------------------------------|-------------|--------------|-------------------|-------|
| BLOCK ERASE | 60h | D0h | No | |
| BLOCK LOCK | 2Ah | – | No | |
| BLOCK LOCK READ STATUS | 7Ah | – | No | |
| BLOCK LOCK TIGHT | 2Ch | – | No | |
| BLOCK UNLOCK | 23h-24h | – | No | |
| OTP DATA PROGRAM | A0h | 10h | No | |
| OTP DATA PROTECT | A5h | 10h | No | |
| OTP DATA READ | AFh | 30h | No | |
| PAGE READ | 00h | 30h | No | |
| PAGE READ CACHE MODE START | 31h | – | No | |
| PAGE READ CACHE MODE LAST | 3Fh | – | No | |
| PROGRAM for INTERNAL DATA MOVE | 85h | 10h | No | |
| PROGRAM PAGE | 80h | 10h | No | |
| PROGRAM PAGE CACHE MODE | 80h | 15h | No | |
| PROGRAMMABLE DRIVE STRENGTH | B8h | – | No | |
| RANDOM DATA INPUT | 85h | – | No | 1 |
| RANDOM DATA READ | 05h | E0h | No | 2 |
| READ for INTERNAL DATA MOVE | 00h | 35h | No | |
| READ ID | 90h | – | No | |
| READ ID (ONFI) | 90h | – | No | |
| READ PARAMETER PAGE (ONFI) | ECh | – | No | |
| READ STATUS | 70h | – | Yes | |
| RESET | FFh | – | Yes | |

- Notes: 1. RANDOM DATA INPUT command is limited to use within a single page.
2. RANDOM DATA READ command is limited to use within a single page.

READ Operations

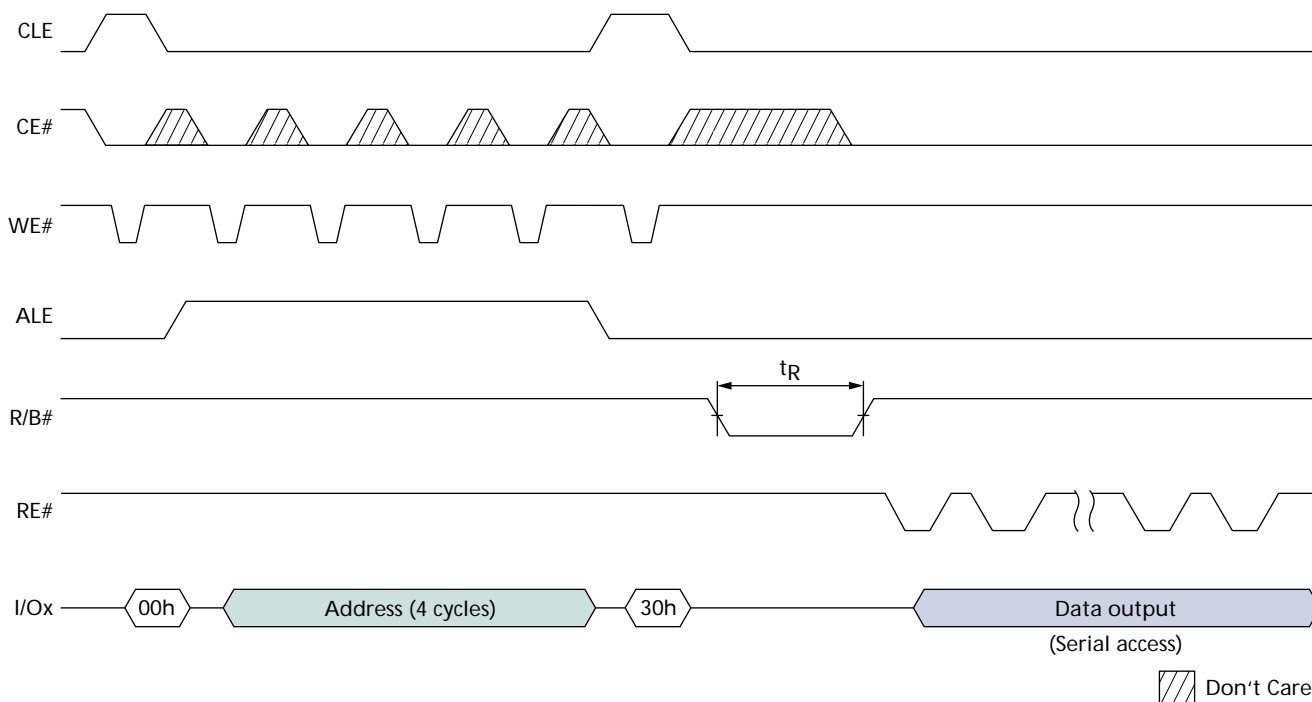
PAGE READ 00h-30h

To enter READ mode, write a 00h command to the device, then specify the starting address via the ADDRESS cycles, and finally, issue the 30h command. At this point, the device enters a busy state while it retrieves data from the NAND Flash array. During this time, the ready/busy status of the device can be monitored using the R/B# or the READ STATUS (70h) command.

The R/B# signal is LOW when the device is busy retrieving data from the NAND Flash array. When R/B# returns to HIGH, data is ready for output. Pulsing the RE# line results in data output on the I/O lines. Note that the first byte or word of data output is that which was specified in the ADDRESS cycle. Each pulse of the RE# signal increases the address counter by one, so additional address cycles are not required when reading sequential data.

If the system does not have a R/B# signal, NAND Flash device status can be monitored by issuing a READ STATUS (70h) command, then reading bit 5 or 6 from the status register (0 = busy; 1 = ready). If the READ STATUS command is used to monitor the data transfer, the user must re-issue the READ (00h) command to initiate data output from the data register. The user can issue 00h only after R/B# goes HIGH or the status register value is E0h. See Figure 54 on page 62 and Figure 55 on page 63 for examples.

Figure 12: PAGE READ Operation



RANDOM READ 05h-E0h

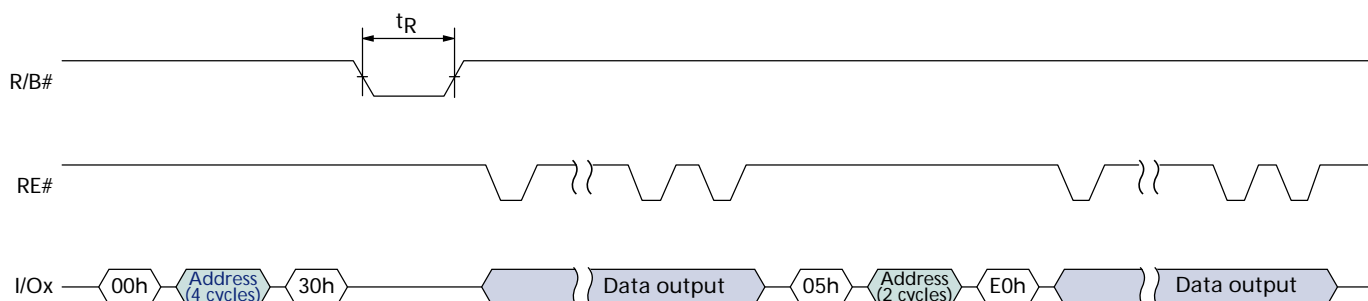
The RANDOM READ command enables the user to specify a new column address so data at single or multiple addresses can be read. The random read mode is enabled after a normal PAGE READ (00h-30h sequence).

Random data can be output after the initial PAGE READ by writing an 05h-E0h command sequence along with the new column address (2 cycles).

The RANDOM READ command can be issued without limit within the page.

Only data on the current page can be read. Pulsing RE# outputs data in the same manner as a serial PAGE READ (see Figure 13).

Figure 13: RANDOM DATA READ Operation



PAGE READ CACHE MODE START 31h; PAGE READ CACHE MODE START LAST 3Fh

Micron NAND Flash devices have a cache register that can be used to increase READ operation speed when accessing sequential pages in a block.

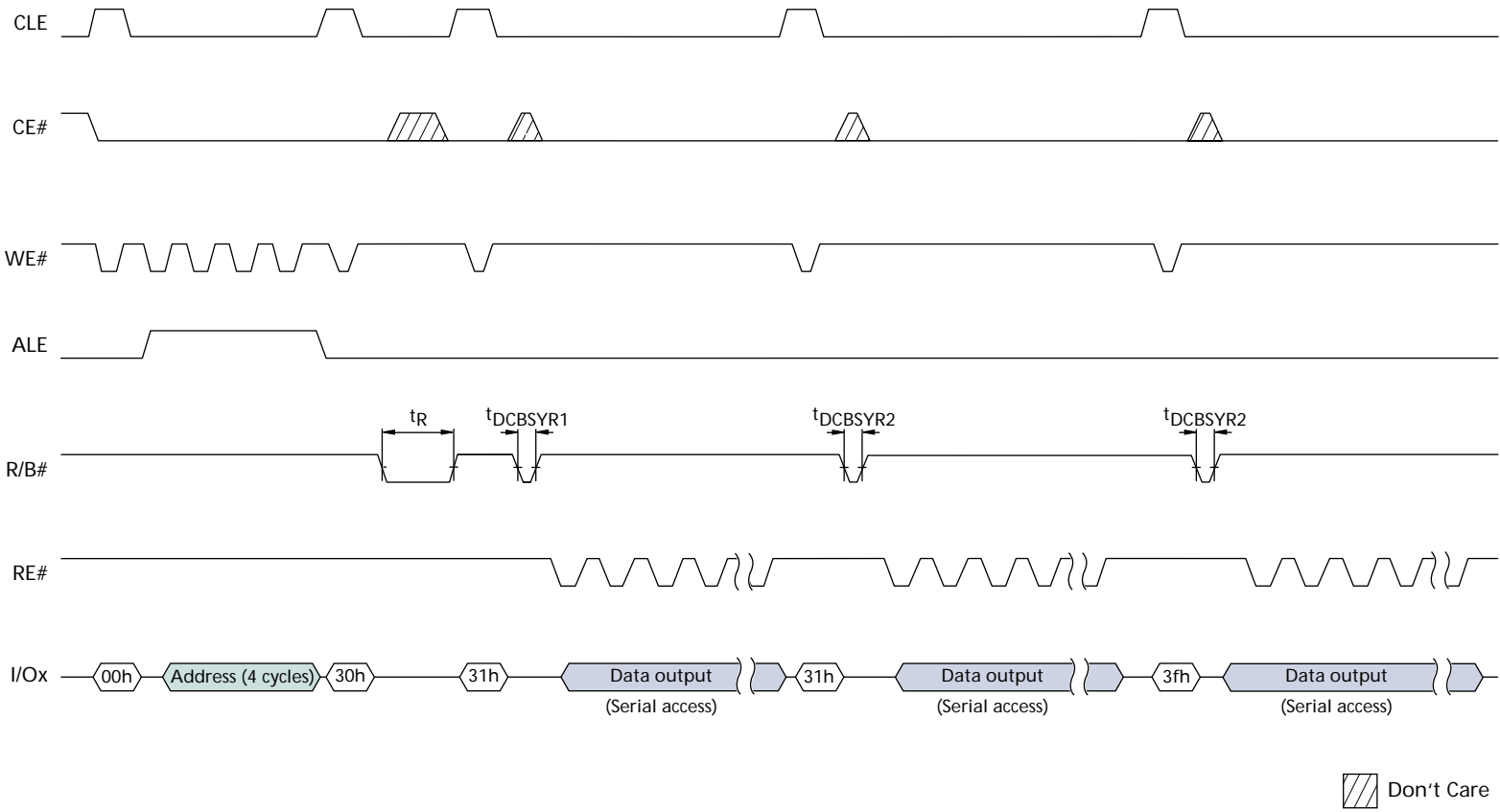
A normal PAGE READ (00h-30h) command sequence is issued (see Figure 14 on page 23 for details). The R/B# signal goes LOW for t_R during the time it takes to transfer the first page of data from the memory to the data register. After R/B# returns to HIGH, the PAGE READ CACHE MODE START (31h) command is latched into the command register. R/B# goes LOW for $t_{DCBSYR1}$ while data is being transferred from the data register to the cache register. When the data register contents are transferred to the cache register, another PAGE READ is automatically started as part of the 31h command. Data is transferred from the memory array to the data register at the same time data is being output (pulsing of RE#) from the cache register. If the total time to output data exceeds t_R , then the PAGE READ is hidden.

The second and subsequent pages of data are transferred to the cache register by issuing additional 31h commands. R/B# will stay LOW up to $t_{DCBSYR2}$. This time can vary, depending on whether the previous memory-to-data-register transfer was completed before issuing the next 31h command. If the data transfer from memory to the data register is not completed before the 31h command is issued, R/B# stays LOW until the transfer is complete.

It is not necessary to output a whole page of data before issuing another 31h command. R/B# will stay LOW until the previous PAGE READ is complete and the data has been transferred to the cache register.

To read out the last page of data, the PAGE READ CACHE MODE START LAST (3Fh) command is issued. This command transfers data from the data register to the cache register without another PAGE READ (see Figure 14 on page 23 for details).

Figure 14: PAGE READ CACHE MODE

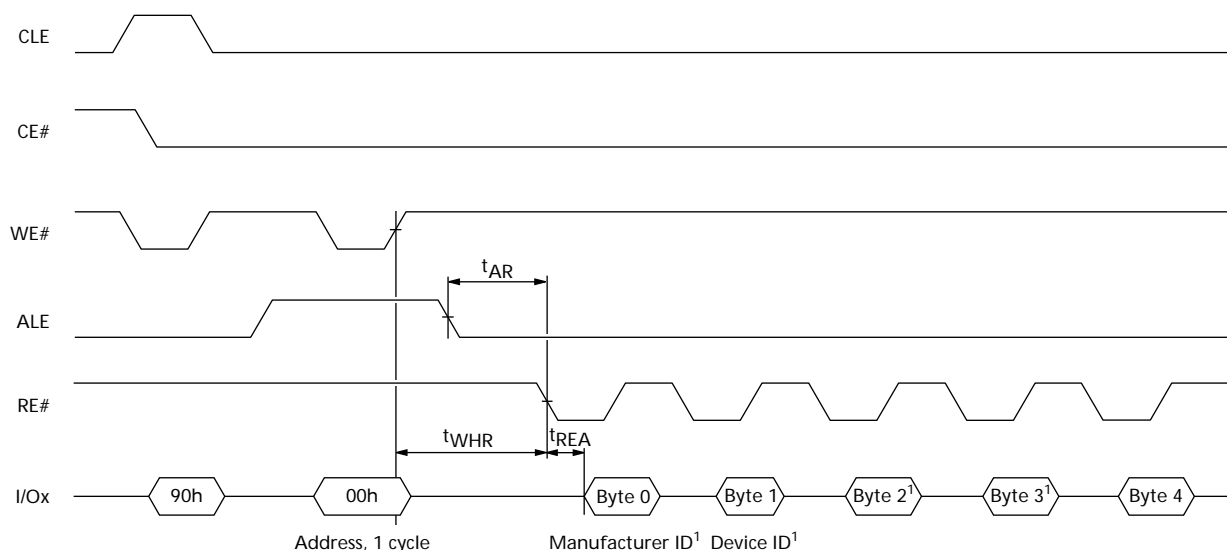


READ ID 90h

The READ ID command is used to read the identifier codes from the MT29F1G08 and MT29F1G16 devices. The READ ID command reads a 5-byte table that includes the manufacturer ID, device configuration, and part-specific information. Table 8 on page 25 shows a complete listing of configuration details.

Issuing a 90h command to the command register and a 00h command to the address register puts the device in read ID mode. The device will remain in this mode until another valid command and address are issued (see Figure 15). If a 90h command is issued without an address, the device will remain in read ID mode.

Figure 15: READ ID Operation



Notes: 1. See Table 8 on page 25 for byte definitions.

Table 8: Device ID and Configuration Codes

| | | Options | I/O7 | I/O6 | I/O5 | I/O4 | I/O3 | I/O2 | I/O1 | I/O0 | Value ¹ | Notes |
|---|--------------|---------------------|------|------|------|------|------|------|------|------|--------------------|-------|
| Byte 0 | | Manufacturer ID | | | | | | | | | | |
| | | Micron | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 2Ch | |
| Byte 1 | | Device ID | | | | | | | | | | |
| MT29F1G08ABB | | 1Gb, x8, 1.8V | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | A1h | |
| MT29F1G16ABB | | 1Gb, x16, 1.8V | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | B1h | |
| Byte 2 | | | | | | | | | | | | |
| Number of die | | 1 | | | | | | | 0 | 0 | 00b | |
| Cell type | | SLC | | | | | 0 | 0 | | | 00b | |
| Number of simultaneously programmed pages | | 1 | | | 0 | 0 | | | | | 00b | |
| Interleaved operations between multiple die | | Not supported (1Gb) | | 0 | | | | | | | 0b | |
| Cache programming | | Supported | 1 | | | | | | | | 1b | |
| Byte value | MT29F1GxxABB | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 80h | |
| Byte 3 | | | | | | | | | | | | |
| Page size | | 2KB | | | | | | | 0 | 1 | 01b | |
| Spare area size (bytes) | | 64 | | | | | | 1 | | | 01b | |
| Block size (w/o spare) | | 128KB | | | 0 | 1 | | | | | 01b | |
| Organization | | x8 | | 0 | | | | | | | 0b | |
| | | x16 | | 1 | | | | | | | 1b | |
| Serial access (MIN) | | 50ns | 0 | | | | 0 | | | | 0xxx0b | |
| Byte value | MT29F1G08ABB | x8 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 95h | |
| | MT29F1G16ABB | x16 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | D5h | |
| Byte 4 | | | | | | | | | | | | |
| Reserved | | | | | | | | | 0 | 0 | 00b | |
| Planes per die | | 1 | | | | | 0 | 0 | | | 00b | |
| Plane size | | 1Gb | | 0 | 0 | 0 | | | | | 000b | |
| Reserved | | | 0 | | | | | | | | 0b | |
| Byte value | MT29F1GxxABB | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00h | |

Notes: 1. b = binary; h = hex.

ONFI READ ID

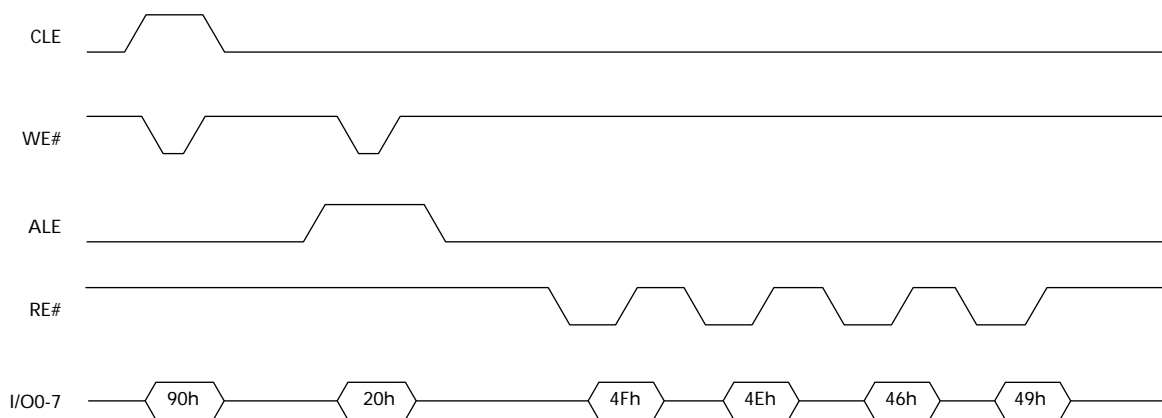
The ONFI READ ID function identifies that the device supports the ONFI specification. If the device supports the ONFI specification, then the ONFI signature will be returned. The ONFI signature is the ASCII encoding of ONFI:

- O = 4Fh
- N = 4Eh
- F = 46h
- I = 49h.

Reading beyond these four values yields indeterminate data. Figure 16 defines the ONFI READ ID behavior and timings.

Issuing a 90h command to the command register and a 20h command to the address register puts the device into ONFI read ID mode. The device will remain in this mode until another valid command and address are issued. If a 90h command is issued without an address, the device will remain in the ONFI read ID mode.

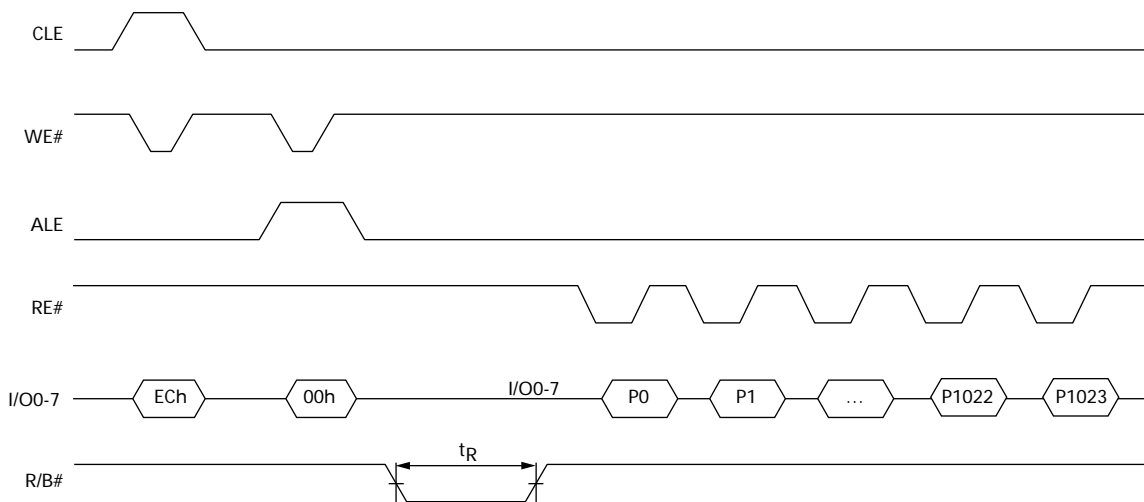
Figure 16: ONFI READ ID Operation



ONFI READ PARAMETER PAGE Operation

The READ PARAMETER PAGE function retrieves the data structure that describes the device organization, features, timings, and other behavioral parameters. Figure 17 defines the READ PARAMETER PAGE behavior.

Figure 17: ONFI READ PARAMETER PAGE Operation



Parameter Page Data Structure Definition

For parameters that span multiple bytes, the least significant byte of the parameter corresponds to the first byte. For example, if bytes 8–9 contain a 16-bit parameter, then bits 7:0 are contained in byte 8.

READ STATUS 70h

The MT29F1G08 and MT29F1G16 devices have an 8-bit status register the software can read during device operation. On the x16 device, I/O[15:8] are “0” when reading the status register. Table 9 on page 29 describes the status register.

After a READ STATUS (70h) command, all READ cycles will be from the status register until a new command is issued. Changes in the status register will be seen on I/O[7:0] as long as CE# and RE# are LOW. It is not necessary to start a new READ cycle to see these changes.

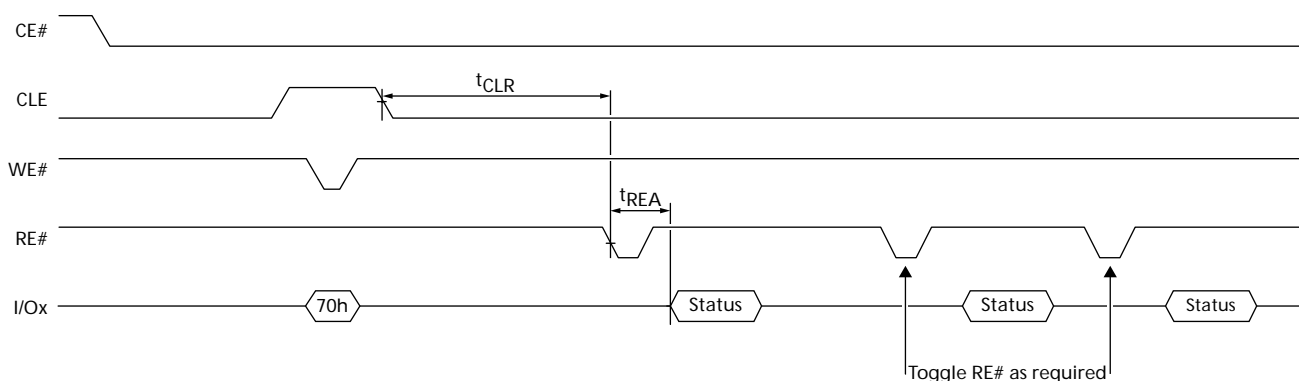
During monitoring of the status register to determine when the t_R (transfer from NAND Flash array to data register) is complete, the READ (00h) command must be re-issued to make the change from STATUS READs to DATA READs. After the READ command has been re-issued, pulsing the RE# line will result in outputting data, starting from the specified column address.

Table 9: Status Register Bit Definition

| SR Bit | Program Page | Program Page Cache Mode | Page Read | Page Read Cache Mode | Block Erase | Definition | Notes |
|--------|---------------|-------------------------|---------------|----------------------|---------------|--|-------|
| 0 | Pass/fail | Pass/fail (N) | – | – | Pass/fail | 0 = Successful PROGRAM/ERASE 1 = Error in PROGRAM/ERASE | |
| 1 | – | Pass/fail (N - 1) | – | – | – | 0 = Successful PROGRAM 1 = Error in PROGRAM | |
| 2 | – | – | – | – | – | 0 | |
| 3 | – | – | – | – | – | 0 | |
| 4 | – | – | – | – | – | 0 | |
| 5 | Ready/busy | Ready/busy | Ready/busy | Ready/busy | Ready/busy | 0 = Busy 1 = Ready | 1 |
| 6 | Ready/busy | Ready/busy cache | Ready/busy | Ready/busy cache | Ready/busy | 0 = Busy 1 = Ready | 2 |
| 7 | Write protect | Write protect | Write protect | Write protect | Write protect | 0 = Protected 1 = Not protected | 3 |
| [15:8] | – | – | – | – | – | 0 | |

- Notes:
- Status register bit 5 is “0” during the actual programming operation. If cache mode is used, this bit will be “1” when all internal operations are complete.
 - Status register bit 6 is “1” when the cache is ready to accept new data. R/B# follows bit 6. See Figure 19 on page 30, and Figure 25 on page 36.
 - Status register bit 7 typically mirrors the status of WP#. However, when BLOCK LOCK is used, status register bit 7 returns “0” if PROGRAM or ERASE operations are performed on a locked block. Additionally, when using the OTP PROGRAM DATA command, status register bit 7 returns “0” if the page is protected. This bit is not modified until the next PROGRAM or ERASE command is issued.

Figure 18: Status Register Operation



PROGRAM Operations

PROGRAM PAGE 80h-10h

Micron NAND Flash devices are inherently page-programmed devices. Pages must be programmed consecutively within a block, from the least significant page address to most significant page address (that is, 0, 1, 2, ..., 63). Random page address programming is prohibited.

Micron NAND Flash devices also support partial-page programming operations. This means that any single bit can only be programmed one time before an erase is required; however, the page can be partitioned so that a maximum of eight programming operations are supported before an erase is required.

SERIAL DATA INPUT 80h

PROGRAM PAGE operations require loading the SERIAL DATA INPUT (80h) command into the command register, followed by the ADDRESS cycles, then the data. Serial data is loaded on consecutive WE# cycles starting at the given address. The PROGRAM (10h) command is written after the data input is complete. The internal control logic automatically executes the proper algorithm and controls all the necessary timing to program and verify the operation. Write verification only detects “1s” that are not successfully written to “0.”

R/B# goes LOW for the duration of array programming time, t_{PROG} . The READ STATUS REGISTER (70h) command and the RESET (FFh) command are the only commands valid during the programming operation. Bit 6 of the status register will reflect the state of R/B#. When the device reaches ready, read bit 0 of the status register to determine if the program operation passed or failed (see Figure 19). The command register stays in read status register mode until another valid command is written to it.

RANDOM DATA INPUT 85h

After the initial data set is input, additional data can be written to a new column address with the RANDOM DATA INPUT (85h) command. The RANDOM DATA INPUT command can be used any number of times in the same page prior to issuing the PAGE WRITE (10h) command. See Figure 20 for the proper command sequence.

Figure 19: PROGRAM and READ STATUS Operation

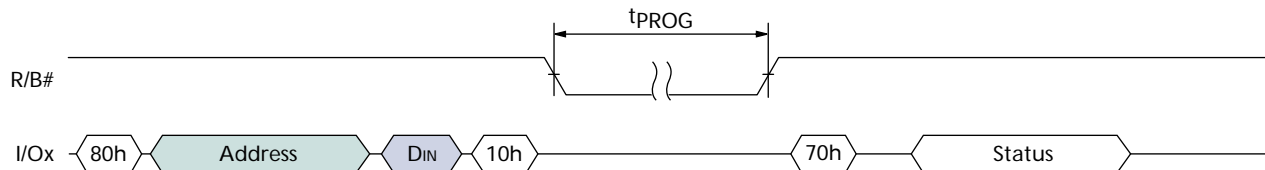
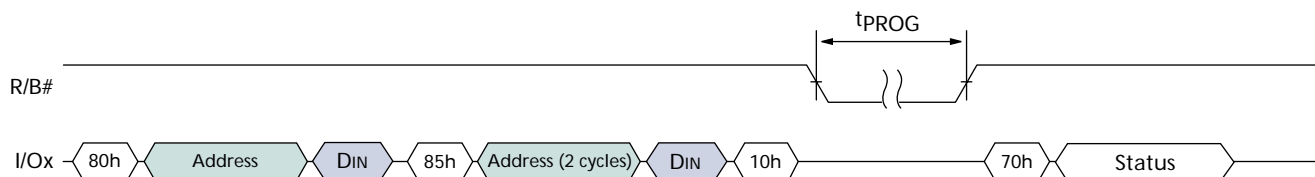


Figure 20: RANDOM DATA INPUT



PROGRAM PAGE CACHE MODE 80h-15h

Cache programming is actually a buffered programming mode of the standard page programming command. Programming is started by loading the SERIAL DATA INPUT (80h) command to the command register, followed by 4 cycles of address, and a full or partial page of data. The data is initially copied into the cache register, and the CACHE WRITE (15h) command is then latched to the command register. Data is transferred from the cache register to the data register on the rising edge of WE#. R/B# goes LOW during this transfer time. After the data has been copied into the data register and R/B# returns to HIGH, memory array programming begins.

When R/B# returns to HIGH, new data can be written to the cache register by issuing another PROGRAM PAGE CACHE MODE command sequence. The time that R/B# stays LOW will be controlled by the actual programming time. The first time through equals the time it takes to transfer the cache register contents to the data register. On the second and subsequent programming passes, transfer from the cache register to the data register is held off until current data register content has been programmed into the array.

The PROGRAM PAGE CACHE MODE command can cross block address boundaries; it must not cross die address boundaries. RANDOM DATA INPUT (85h) commands are permitted with PROGRAM PAGE CACHE MODE operations.

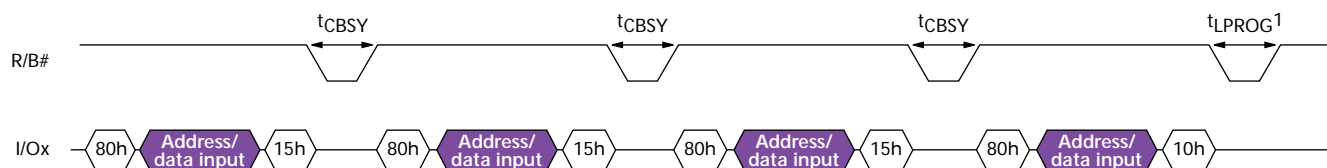
Bit 6 (cache R/B#) of the status register can be read by issuing the READ STATUS (70h) command to determine when the cache register is ready to accept new data. R/B# always follows bit 6.

Bit 5 (R/B#) of the status register can be polled to determine when the actual programming of the array is complete for the current programming cycle.

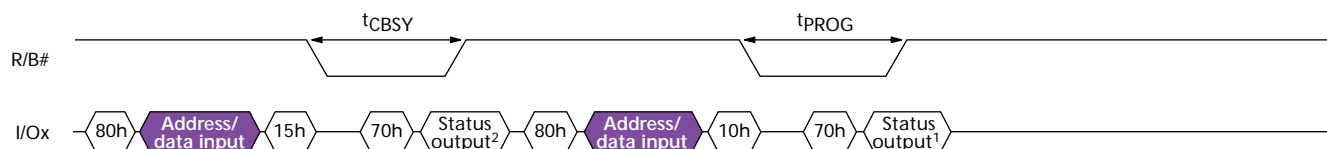
If R/B# is used to determine programming completion, the last page of the program sequence must use the PROGRAM PAGE (10h) command instead of the CACHE PROGRAM (15h) command. If the CACHE PROGRAM (15h) command is used every time, including the last page of the programming sequence, status register bit 5 must be used to determine when programming is complete.

Bit 1 of the status register returns the pass/fail for the previous page when bit 6 of the status register is a "1" (ready state). The pass/fail status of the current PROGRAM operation is returned with bit 0 of the status register when bit 5 of the status register is a "1" (ready state) (see Figure 21 on page 32).

Figure 21: PROGRAM PAGE CACHE MODE Example



A: Without status reads



B: With status reads

- Notes:
1. For definition of t_{LPROG} , see note 3, Table 23 on page 55.
 2. Check I/O[6:5] for internal ready/busy. Check I/O[1:0] for pass/fail. RE# can remain LOW or pulse multiple times after a 70h command.

INTERNAL DATA MOVE Operations

An internal data move requires two command sequences. Issue a READ for INTERNAL DATA MOVE (00h-35h) command first, then the INTERNAL DATA MOVE (85h-10h) command. Data moves are only supported within the die from which data is read.

READ FOR INTERNAL DATA MOVE 00h-35h

This READ command is used in conjunction with the INTERNAL DATA MOVE (85h-10h) command. First, 00h is written to the command register, then the internal source address is written (4 cycles). After the address is input, the READ for INTERNAL DATA MOVE (35h) command writes to the command register. This transfers a page from memory into the cache register. The written column addresses are ignored even though all 4 address cycles are required. The memory device is now ready to accept the INTERNAL DATA MOVE (85h-10h) command.

INTERNAL DATA MOVE 85h-10h

After the READ for INTERNAL DATA MOVE command has been issued and R/B# goes HIGH, the INTERNAL DATA MOVE command can be written to the command register. This command transfers the data from the cache register to the data register and programming of the new destination page begins. After the INTERNAL DATA MOVE command and address sequence are written to the device, R/B# goes LOW while the internal control logic automatically programs the new page. The READ STATUS command and bit 6 of the status register can be used instead of the R/B# line to determine when the WRITE is complete. Bit 0 of the status register indicates if the operation was successful.

The RANDOM DATA INPUT (85h) command can be used during the INTERNAL DATA MOVE command sequence to modify a word or multiple words in the original data. First, data is copied into the cache register using the 00h-35h command sequence; then the RANDOM DATA INPUT (85h) command is written, along with the address of the data to be modified next. New data is input on the external data balls. This copies the new data into the cache register.

When 10h is written to the command register, the original data plus the modified data is transferred to the data register, and programming of the new page is started. The RANDOM DATA INPUT command can be issued as many times as necessary before starting the programming sequence with 10h (see Figure 22 and Figure 23 on page 34 for details).

Because the INTERNAL DATA MOVE operation does not utilize external memory, ECC cannot be used to check for errors before programming the data to a new page. This can lead to a data error if the source page contains a bit error due to charge loss or charge gain. If multiple INTERNAL DATA MOVE operations are performed, these bit errors may accumulate without correction. For this reason, it is highly recommended that systems utilizing the INTERNAL DATA MOVE operation use a robust ECC scheme that can correct two or more bits per sector.

Figure 22: INTERNAL DATA MOVE

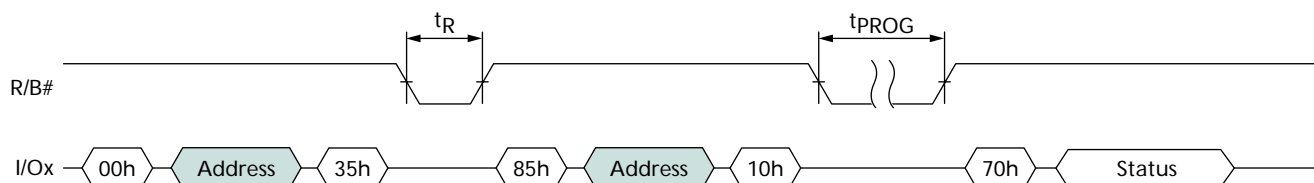
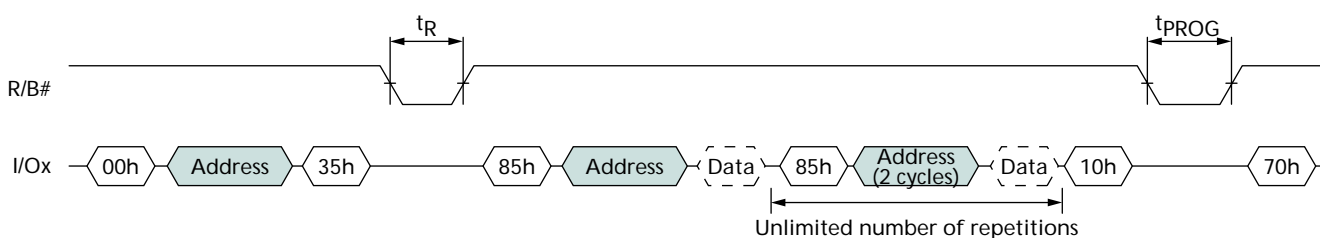


Figure 23: INTERNAL DATA MOVE with RANDOM DATA INPUT



BLOCK ERASE 60h-D0h

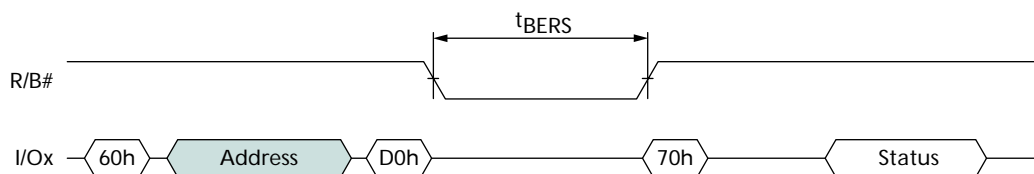
Erasing occurs at the block level. The MT29F1G08 and the MT29F1G16 have 1,024 erase blocks organized as 64 pages per block. The BLOCK ERASE command operates on one block at a time (see Figure 24).

Two cycles of addresses BA[15:6] are required for the x8 device, and 2 cycles of BA[15:6] for the x16 device. Although addresses PA[5:0] (x8) and PA[5:0] (x16) are loaded, they are a “Don’t Care” and are ignored for BLOCK ERASE operations. See Figures 6 and 7 on pages 14 and 15 for addressing details.

The actual BLOCK ERASE command sequence is a two-step process. First, write the ERASE SETUP (60h) command to the command register. Then write 2 cycles of addresses to the device. Next, write the ERASE CONFIRM (D0h) command to the command register. At the rising edge of WE#, R/B# goes LOW and the internal control logic automatically controls the timing and erase-verify operations. R/B# stays LOW for the entire t_{BERS} erase time.

The READ STATUS REGISTER command can be used to check the status of the ERASE operation. When bit 6 = 1, the ERASE operation is complete. Bit 0 indicates a pass/fail condition where 0 = pass. See BLOCK ERASE, and Table 9 on page 29 for details.

Figure 24: BLOCK ERASE Operation



One-Time Programmable (OTP) Area

This Micron NAND Flash device offers a protected, one-time programmable NAND Flash memory area. Ten full pages (2,112 bytes or 1,056 words per page) of OTP data is available on the device, and the entire range is guaranteed to be good. The OTP area is accessible only through the OTP commands. Customers can use the OTP area in any way they desire; typical uses include programming serial numbers or other data for permanent storage.

In Micron NAND Flash devices, the OTP area leaves the factory in an unwritten state (each OTP bit is “1”). Programming or partial-page programming enables the user to program only “0” bits in the OTP area. The OTP area cannot be erased, even if it is not protected. Protecting the OTP area simply prevents further programming of the OTP area.

While the OTP area is referred to as “one-time programmable,” Micron provides a unique way to program and verify data—before permanently protecting it and preventing future changes.

OTP programming and protection are accomplished in two discrete operations. First, using the OTP DATA PROGRAM (A0h-10h) command, an OTP page is programmed entirely in one operation, or in up to four partial-page programming sequences. Programming can occur on other pages within the OTP area in a similar manner. Second, the OTP area is permanently protected from further programming using the OTP DATA PROTECT (A5h-10h) command. The pages within the OTP area can always be read using the OTP DATA READ (AFh-30h) command, whether or not it is protected.

OTP DATA PROGRAM A0h-10h

The OTP DATA PROGRAM (A0h-10h) command is used to write data to the pages within the OTP area. An entire page can be programmed at one time, or a page can be partially programmed up to four times. There is no ERASE operation for the OTP pages.

The OTP DATA PROGRAM enables programming into an offset of an OTP page, using the two bytes of column address (CA[11:0]). The OTP DATA PROGRAM command will not execute if the OTP area has been protected. If the OTP area is protected, the busy time for the OTP DATA PROGRAM operation is ^tOBSY and not ^tPROG.

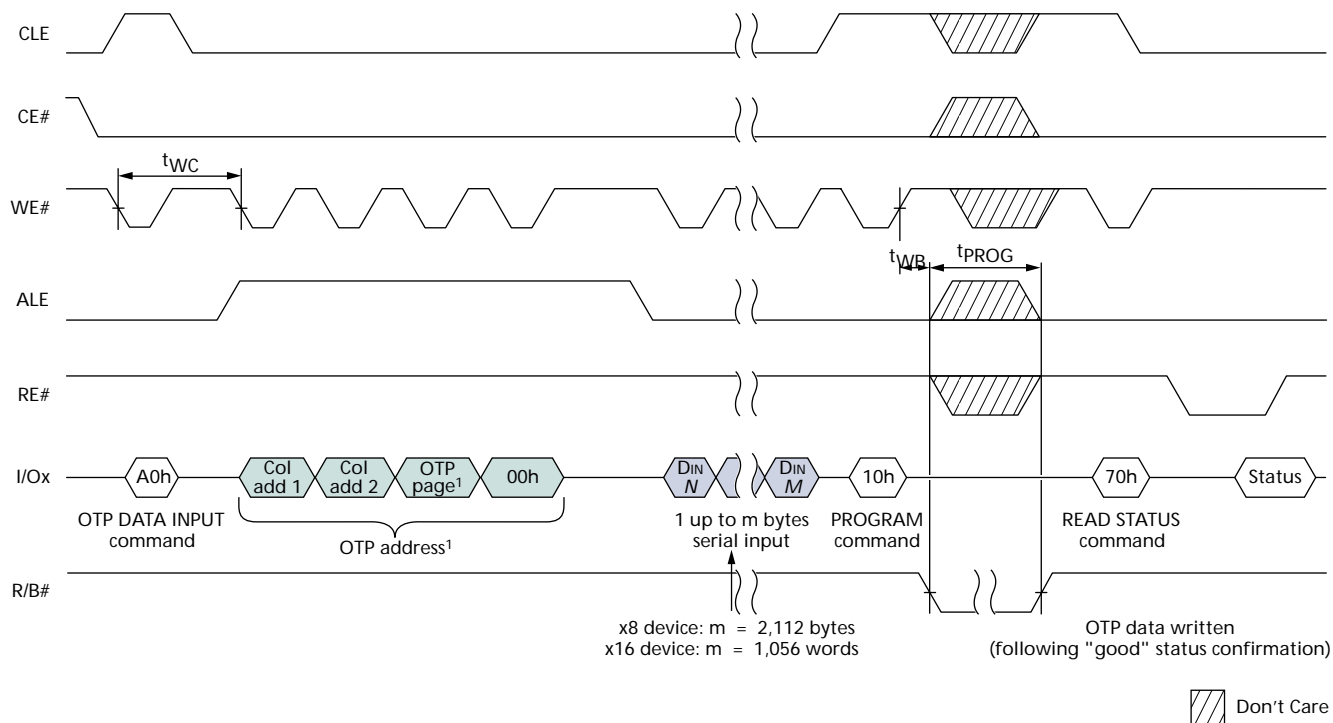
To use the OTP DATA PROGRAM command, issue the A0h command. Issue 4 ADDRESS cycles: the first 2 ADDRESS cycles are the column address, and for the remaining 2 cycles select a page in the 02h–0Bh range. Next, write the data: from 1 to 2,112 bytes (x8 device), or from 1 to 1,056 words (x16 device). After data input is complete, issue the 10h command. The internal control logic automatically executes the proper programming algorithm and controls the necessary timing for programming and verification. Program verification only detects “1”s that are not successfully written to “0”s.

RANDOM DATA INPUT (85h) commands are supported during OTP DATA PROGRAM operations only if the OTP area is unprotected.

R/B# goes LOW during the duration of the array programming time (^tPROG). The READ STATUS (70h) command is the only command valid during the OTP DATA PROGRAM operation. For this operation, bits 5 and 6 of the status register will reflect the state of R/B#. If bit 7 is “0,” then the OTP area has been protected; otherwise, it will be a “1.”

When the device is ready, read bit 0 of the status register to determine if the operation passed or failed (see Table 9 on page 29).

Figure 25: OTP DATA PROGRAM



Notes: 1. The OTP page must be within the 02h–0Bh range.

OTP DATA PROTECT A5h-10h

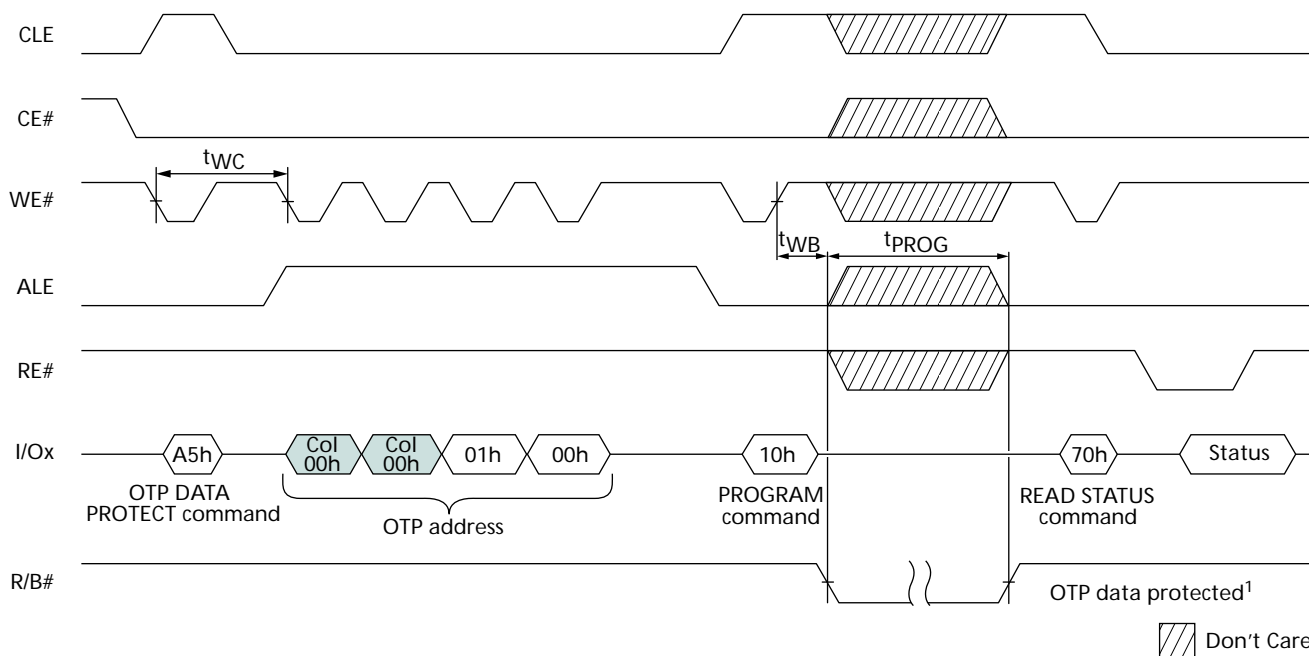
The OTP DATA PROTECT (A5h-10h) command is used to protect all the data in the OTP area. After the data is protected, it cannot be programmed further. When the OTP area is protected, the pages within the area are no longer programmable and cannot be unprotected.

To use the OTP DATA PROTECT command, issue the A5h command. Next, issue the following 4 ADDRESS cycles: 00h-00h-01h-00h. Finally, issue the 10h command.

R/B# goes LOW while the OTP area is being protected. The protect command duration is similar to a normal page programming operation, t_{PROG} . The READ STATUS (70h) command is the only command valid during the OTP DATA PROTECT operation. For this operation, bits 5 and 6 of the status register will reflect the state of R/B#.

When the device is ready, read bit 0 of the status register to determine if the operation passed or failed (see Table 9 on page 29).

Figure 26: OTP DATA PROTECT



Notes: 1. OTP data is protected following "good" status confirmation.

OTP DATA READ AFh-30h

The OTP DATA READ (AFh-30h) command is used to read data from a page within the OTP area. An OTP page within the OTP area is available for reading data whether or not the area is protected.

To use the OTP DATA READ command, issue the AFh command. Next, issue 4 ADDRESS cycles: the first 2 ADDRESS cycles are the column address, and for the remaining 2 cycles, select a page in the range of 02h–0Bh. Finally, issue the 30h command.

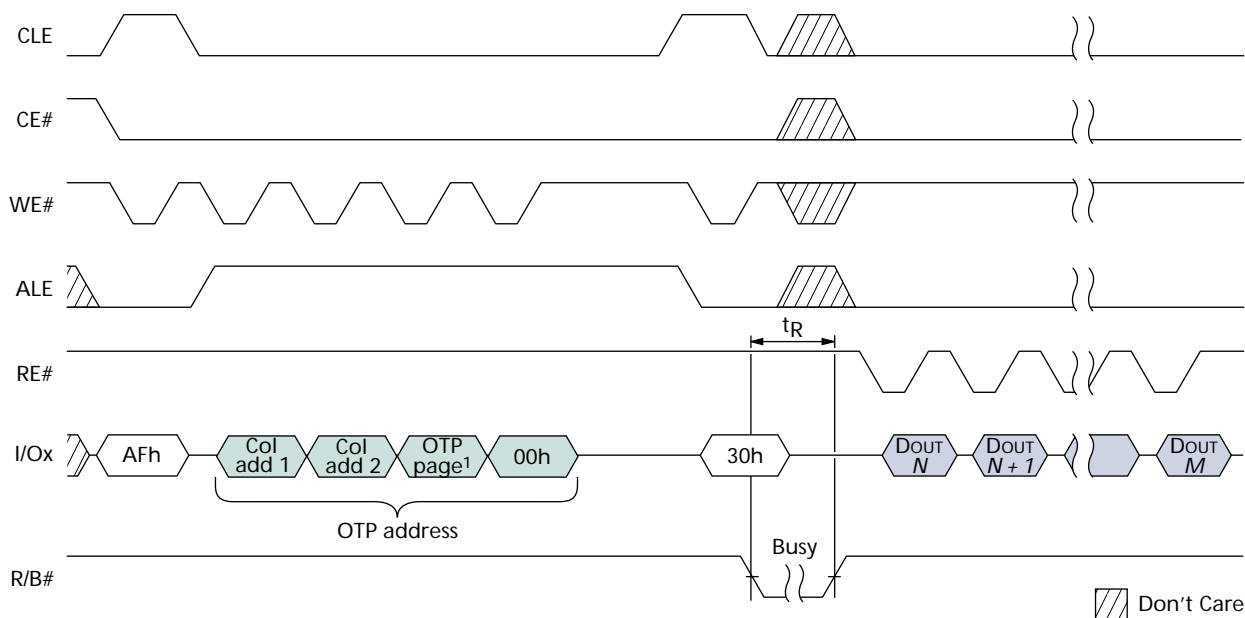
RANDOM DATA READ (05h-E0h) commands are supported during OTP DATA READ operations.

R/B# goes LOW (^tR) while the data is moved from the OTP page to the data register. The READ STATUS (70h) command and the RESET (FFh) command are the only commands valid during the OTP DATA READ operation. For this operation, bits 5 and 6 of the status register will reflect the state of R/B#. For details, refer to Table 9 on page 29.

Normal READ operation timings apply to OTP read accesses (see Figure 27). Additional pages within the OTP area can be selected by repeating the OTP DATA READ command.

Note that if OTP DATA READ is followed by PAGE READ CACHE MODE, a RESET (FFh) must be issued prior to issuing the PAGE READ CACHE MODE command. The maximum RESET time will not exceed 5μs.

Figure 27: OTP DATA READ Operation



Notes: 1. The OTP page must be within the range 02h–0Bh.

BLOCK LOCK Feature

The BLOCK LOCK feature provides the ability to protect the entire device or ranges of blocks from PROGRAM and ERASE operations. Using this BLOCK LOCK feature offers increased functionality and flexibility over using just WP# to prevent PROGRAM and ERASE operations.

BLOCK LOCK features are enabled and disabled at power-on through the use of the LOCK signal. At power-on, if LOCK is LOW, all BLOCK LOCK commands are disabled. However, at power-on, if LOCK is HIGH, the BLOCK LOCK commands are enabled and, by default, all of the blocks on the device are protected, or locked, from PROGRAM and ERASE operations, even if WP# is HIGH.

Before the contents of the device can be modified, the device must first be unlocked. Either a range of blocks or the entire device can be unlocked. PROGRAM and ERASE operations complete successfully only in the block ranges that have been unlocked. Blocks, once unlocked, can be locked again to protect them from further PROGRAM and ERASE operations.

Blocks that are locked can be protected further, or locked tight. When locked tight, the device's blocks can no longer be locked or unlocked until WP# is pulled LOW for more than 100ns. After WP# goes LOW for this period, the entire device is locked from PROGRAM and ERASE operations until unlocked again.

WP# and BLOCK LOCK

When the BLOCK LOCK feature is enabled, it interacts with WP# as follows:

- WP# must be driven HIGH and remain HIGH when UNLOCK and LOCK-TIGHT commands are issued.
- Holding WP# LOW locks all blocks.
- If WP# is held LOW to lock blocks, and then returned to HIGH, a new UNLOCK command must be issued to unlock blocks.

UNLOCK 23h-24h

By default at power-on, if LOCK is HIGH, all of the blocks in the NAND Flash device are locked, meaning that they are protected from PROGRAM and ERASE operations. The UNLOCK (23h) command is used to unlock a range of blocks. Unlocked blocks have no protection and can be programmed or erased.

The UNLOCK command uses two registers—a lower boundary block address register and an upper boundary block address register—and the invert area bit to determine which range of blocks is unlocked. When the invert area bit = 0, the range of blocks within the lower and upper boundary address registers is unlocked. When the invert area bit = 1, the range of blocks outside the boundaries of the lower and upper boundary address registers are unlocked. The lower boundary block address must be less than the upper boundary block address. Figures 28 and 29 on page 40 show examples of how the lower and upper boundary address registers work with the invert area bit.

To unlock a range of blocks, issue the UNLOCK (23h) command followed by the appropriate ADDRESS cycles that indicate the lower boundary block address. Then issue the 24h command followed by the appropriate ADDRESS cycles that indicate the upper boundary block address. The least significant page address bit, PA0, should be set to “1” if setting the invert area bit; otherwise, it should be “0.” The other page address bits should be “0” (see Figure 30 on page 41).

Only one range of blocks can be specified in the lower and upper boundary block address registers. If, after unlocking a range of blocks, the UNLOCK command is again issued, the new block address range determines which blocks are unlocked. The previous unlocked block address range is not retained.

The UNLOCK (23h-24h) command is disabled if LOCK is LOW at power-on or if the device is locked tight (see page 42).

Figure 28: Flash Array Protected: Inverted Area Bit = 0

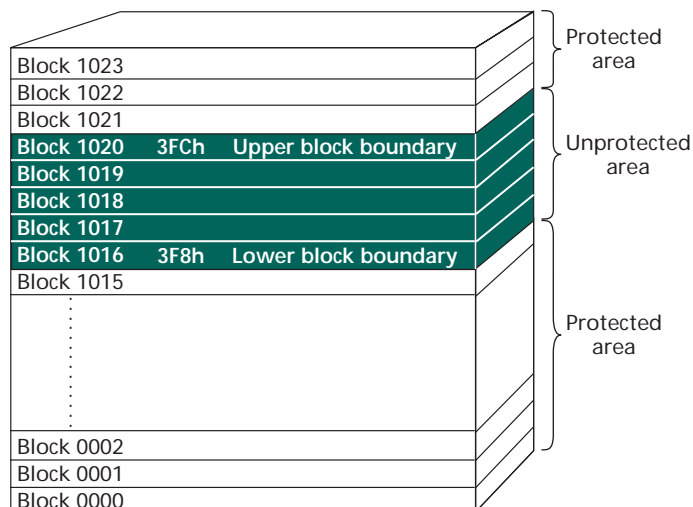


Figure 29: Flash Array Protected: Invert Area Bit = 1

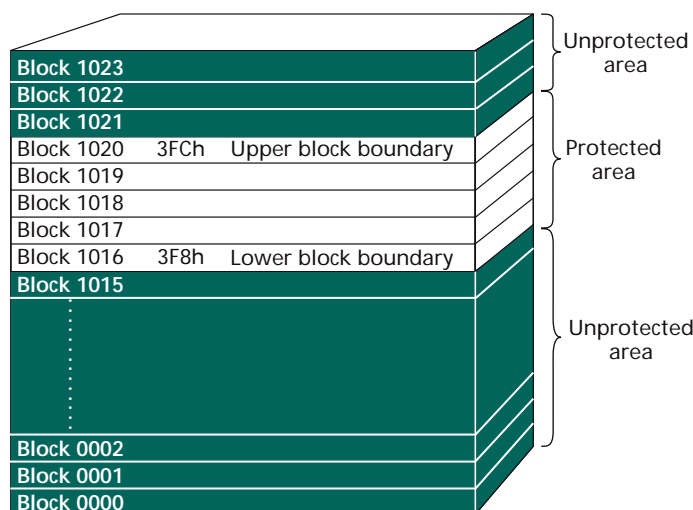


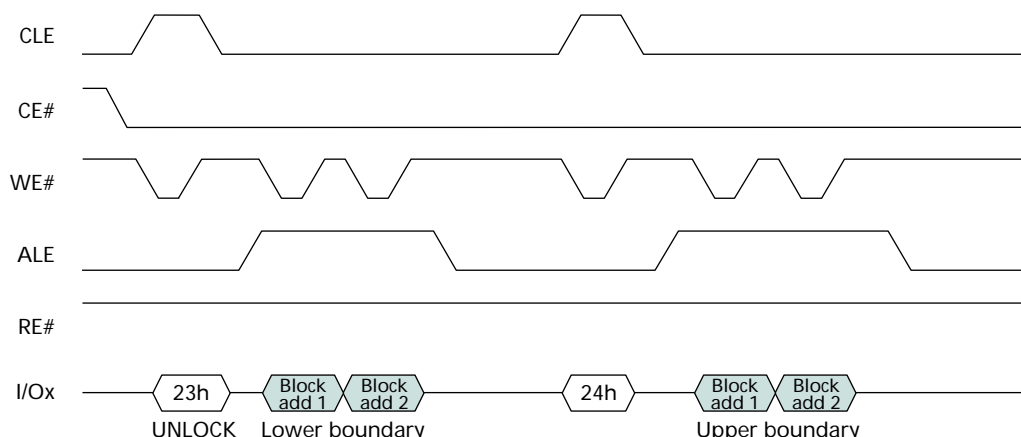
Table 10: BLOCK LOCK Address Cycle Assignments

| ALE Cycle | I/O[15:8] ¹ | I/O7 | I/O6 | I/O5 | I/O4 | I/O3 | I/O2 | I/O1 | I/O0 |
|-----------|------------------------|------|------|------|------|------|------|------|------------------------------|
| First | LOW | BA7 | BA6 | LOW | LOW | LOW | LOW | LOW | Invert Area Bit ² |
| Second | LOW | BA15 | BA14 | BA13 | BA12 | BA11 | BA10 | BA9 | BA8 |

Notes: 1. I/O[15:8] is applicable only for x16 devices.

2. Invert area bit is applicable for 24h command; it can be HIGH or LOW for the 23h command.

Figure 30: UNLOCK Operation



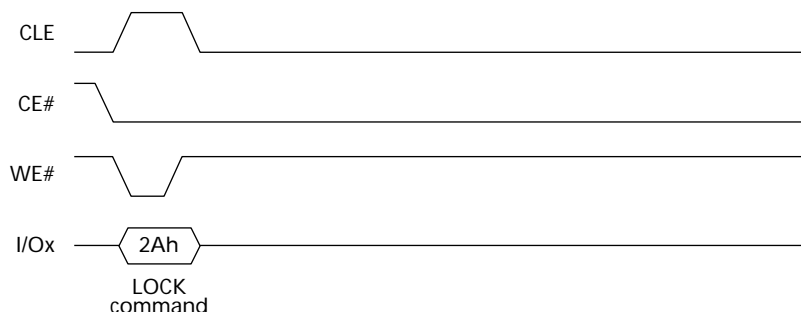
LOCK 2Ah

By default at power-on, if LOCK is HIGH, all of the blocks in the NAND Flash device are locked and protected from PROGRAM and ERASE operations. If portions of the device are unlocked using the UNLOCK (23h) command, they can be locked again using the LOCK (2Ah) command. The LOCK command locks all of the blocks in the device. Locked blocks are write-protected from PROGRAM and ERASE operations.

To lock all of the blocks in the device, issue the LOCK (2Ah) command.

When a PROGRAM or ERASE operation is issued to a locked block, R/B# goes LOW for t_{LBSY}. The PROGRAM or ERASE operation does not complete. The READ STATUS (70h) command reports bit 7 as “0,” indicating that the block is protected.

The LOCK (2Ah) command is disabled if LOCK is LOW at power-on or if the device is locked tight (see page 42).

Figure 31: LOCK Operation


LOCK-TIGHT 2Ch

The LOCK-TIGHT (2Ch) command prevents locked blocks from being unlocked and also prevents unlocked blocks from being locked. When this command is issued, the UNLOCK (23h) and LOCK (2Ah) commands are disabled. This provides an additional level of protection to locked blocks from inadvertent PROGRAM and ERASE operations.

To implement the lock-tight state in all of the locked blocks in the device, verify that WP# is HIGH and then issue the LOCK-TIGHT (2Ch) command.

When a PROGRAM or ERASE operation is issued to a locked block that has also been locked tight, R/B# goes LOW for ^tLBSY. The PROGRAM or ERASE operation does not complete. The READ STATUS (70h) command reports bit 7 as “0,” indicating that the block is protected. PROGRAM and ERASE operations complete successfully to blocks that were not locked at the time the LOCK-TIGHT command was issued.

Once the LOCK-TIGHT command is issued, it cannot be disabled via a software command. The only way to disable the lock-tight status is either to hold WP# LOW for greater than 100ns or to power cycle the device. When the lock-tight status is disabled, all of the blocks become locked, the same as if the LOCK (2Ah) command were issued.

The LOCK-TIGHT (2Ch) command is disabled if LOCK is LOW at power-on.

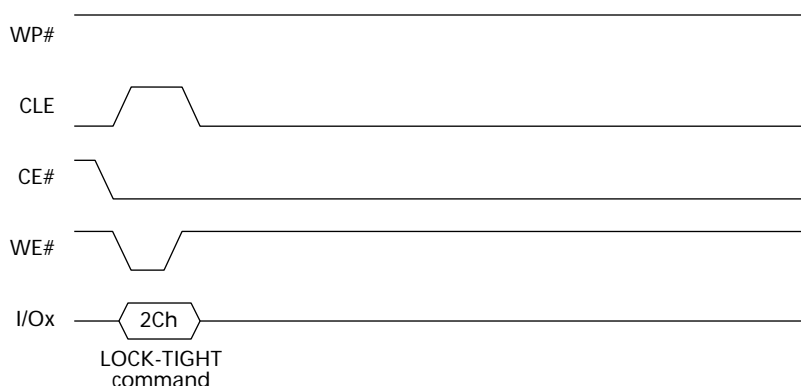
Figure 32: LOCK-TIGHT Operation


Figure 33: PROGRAM/ERASE Issued to Locked or Locked-Tight Block

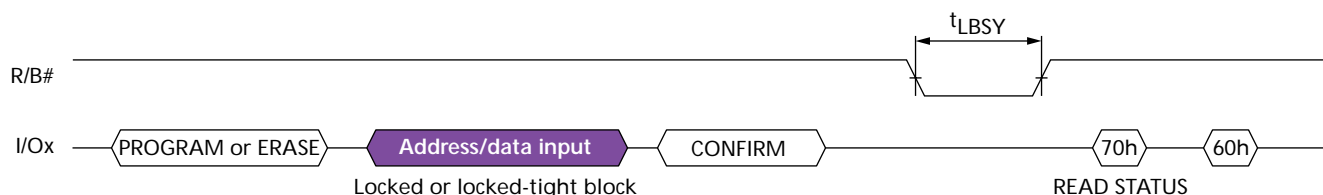
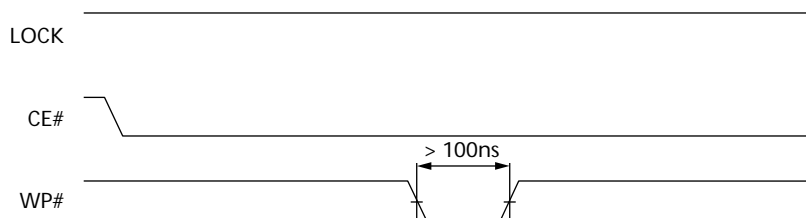


Figure 34: LOCKED-TIGHT BLOCKS to LOCKED BLOCKS Operation



Note: The device ensures exit from lock-tight mode if the WP# pulse is greater than 100ns. The device may exit lock-tight mode if WP# pulse is less than 100ns, however, this is not guaranteed.

BLOCK LOCK READ STATUS 7Ah

The BLOCK LOCK READ STATUS (7Ah) command is used to determine the protection status of individual blocks. The ADDRESS cycles have the same format as shown in Table 10 on page 41; the invert area bit should be set LOW. On the falling edge of RE#, the I/O outputs the block-lock status register which contains the information on the protection status of the block. Table 11 shows how to interpret the block-lock status register bits.

The BLOCK LOCK READ STATUS (7Ah) command is disabled if LOCK is LOW at power-on.

Table 11: BLOCK LOCK Status Register Bit Definitions

| BLOCK LOCK Status Register Definitions | I/O[7:3] | I/O2 (Lock#) | I/O1 (LT#) | I/O0 (LT) |
|---|----------|--------------|------------|-----------|
| Block is locked and device is locked-tight | X | 0 | 0 | 1 |
| Block is locked and device <i>is not</i> locked-tight | X | 0 | 1 | 0 |
| Block is unlocked and device is locked-tight | X | 1 | 0 | 1 |
| Block is unlocked and device <i>is not</i> locked-tight | X | 1 | 1 | 0 |

Figure 35: BLOCK LOCK READ STATUS

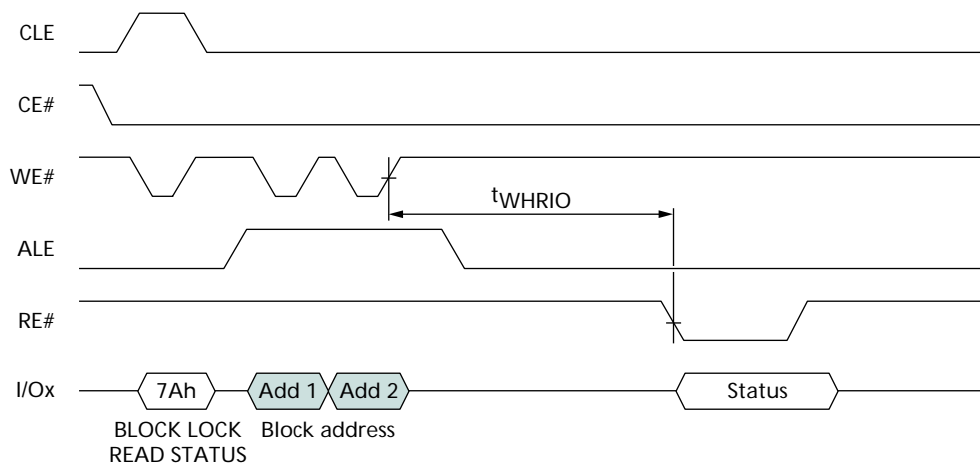
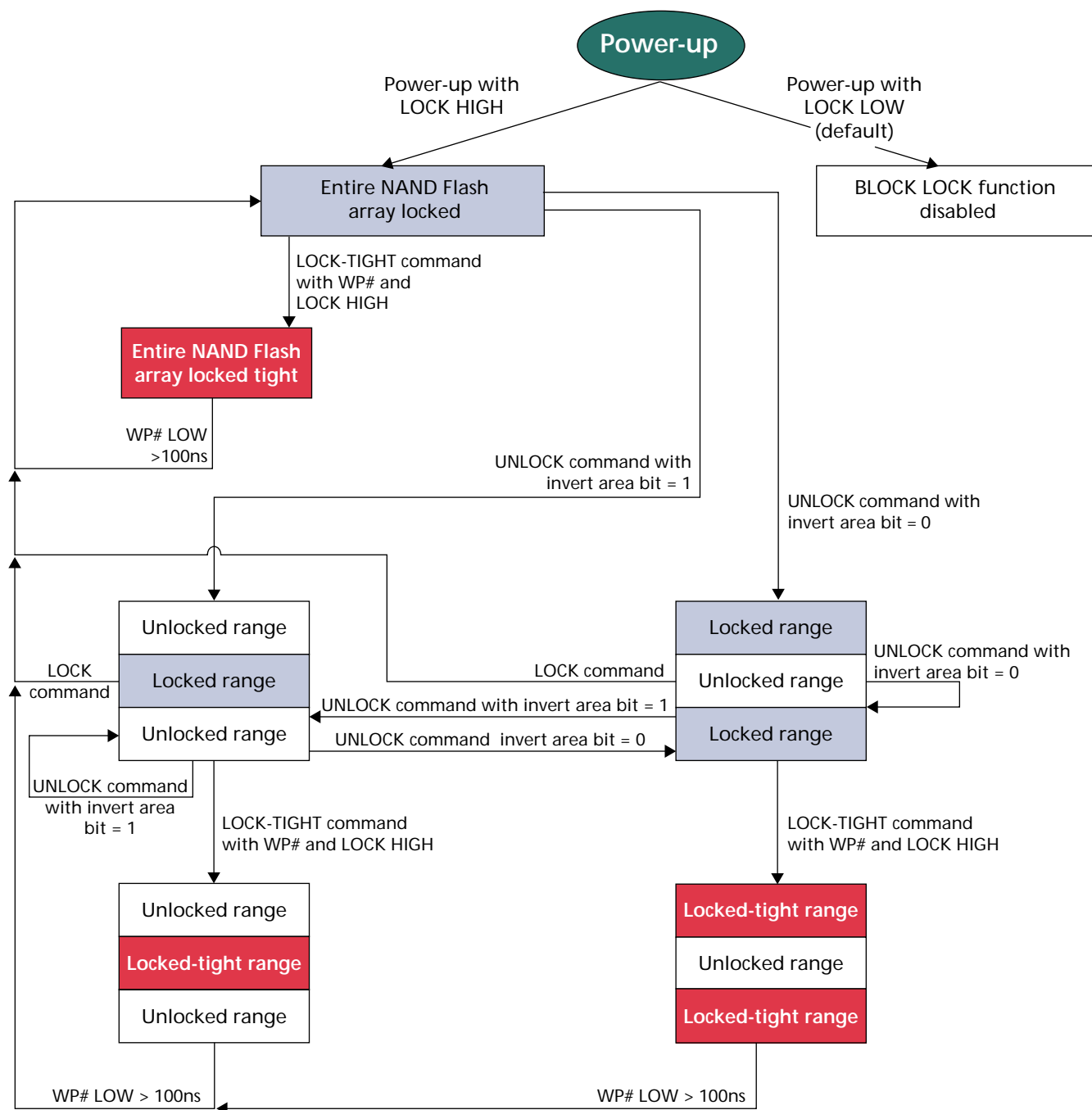


Figure 36: BLOCK LOCK Flow Chart



RESET Operation

RESET FFh

The RESET command is used to put the memory device into a known condition and to abort a command sequence in progress.

READ, PROGRAM, and ERASE commands can be aborted while the device is in the busy state. The contents of the memory location being programmed or the block being erased are no longer valid. The command register is cleared and is ready for the next command.

The status register contains the value E0h when WP# is HIGH; otherwise, it is written with a 60h value. R/B# goes LOW for t_{RST} after the RESET command is written to the command register. See Figure 37 and Table 12 for details.

The RESET command must be issued after power-on and before any other command is issued to the device. The device will be busy for a maximum of 1ms at this time.

Figure 37: RESET Operation

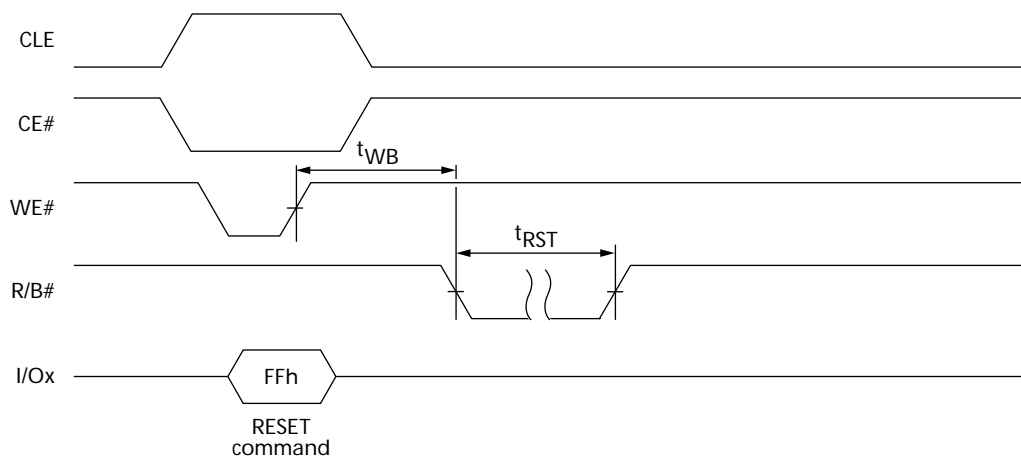


Table 12: Status Register Contents After Reset

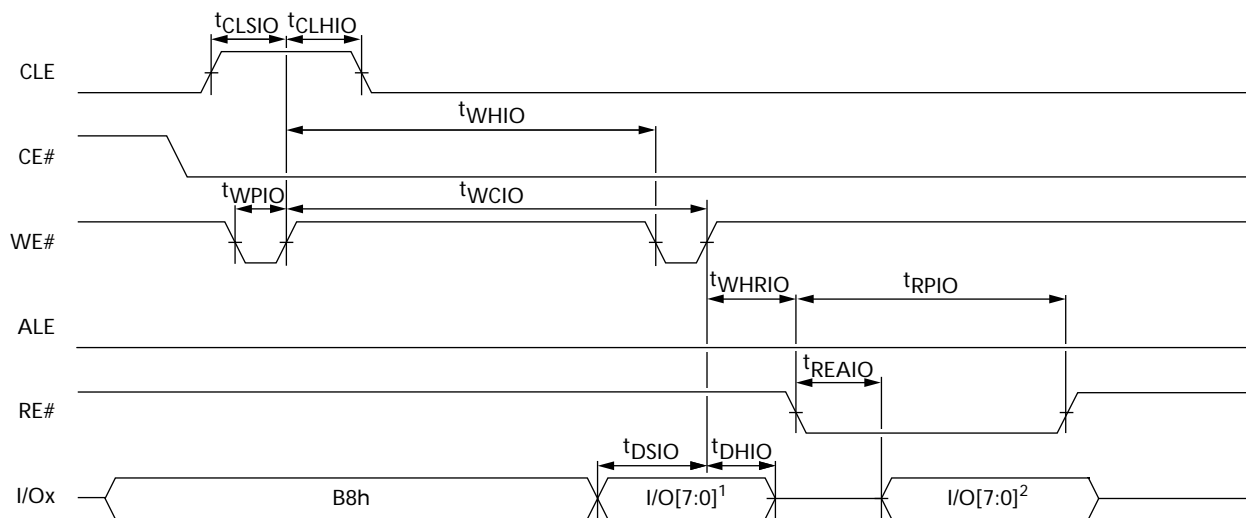
| Condition | Status | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Hex |
|-----------|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-----|
| WP# HIGH | Ready | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | E0h |
| WP# LOW | Ready and write protected | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 60h |

Programmable Drive Strength

PROGRAMMABLE I/O DRIVE STRENGTH B8h

The B8h command is used to change the default I/O drive strength as shown in Figure 38. Drive strength should be selected based on expected memory bus loading. There are four allowable settings for the output drive strength. The settings and the default drive strength are shown in Table 13. The device returns to the default drive strength mode after it is power-cycled. Figure 38 shows how to write and read the drive strength. Refer to Table 14 on page 48 for unique timing parameters associated with the PROGRAMMABLE I/O DRIVE STRENGTH command. Note that the AC timing characteristics documented in Table 21 on page 54 and Table 22 on page 54 may need to be relaxed if the I/O drive strength is not set to “full.”

Figure 38: Programmable I/O Drive Strength Command Sequence



Notes: 1. WRITE operation.
2. READ operation.

Table 13: I/O Drive Strength Settings

| Drive Strength | I/O7 | I/O6 | I/O5 | I/O4 | I/O3 | I/O2 | I/O1 | I/O0 |
|----------------|------|------|------|------|------|------|------|------|
| Full (default) | X | X | X | X | 0 | 0 | X | X |
| Three-quarters | X | X | X | X | 0 | 1 | X | X |
| One-half | X | X | X | X | 1 | 0 | X | X |
| One-quarter | X | X | X | X | 1 | 1 | X | X |

Notes: 1. For WRITE operation, X = “Don’t Care.” For READ operation, X = “Undefined.”
2. Timing parameters shown in Table 21 on page 54 and Table 22 on page 54 represent full drive setting.

Table 14: Programmable I/O Drive Strength Register READ/WRITE Timing

| Parameter | Symbol | Min | Max | Unit | Notes |
|----------------------|-------------|-----|-----|------|-------|
| CLE hold time | t_{CLHIO} | 15 | – | ns | |
| CLE setup time | t_{CLSIO} | 25 | – | ns | |
| Data hold time | t_{DHIO} | 15 | – | ns | |
| Data setup time | t_{DSIO} | 30 | – | ns | |
| RE# access time | t_{REAIO} | – | 250 | ns | |
| RE# pulse width | t_{RPIO} | 250 | – | ns | |
| WRITE cycle time | t_{WCIO} | 100 | – | ns | |
| WE# pulse width HIGH | t_{WHIO} | 50 | – | ns | |
| WE# HIGH to RE# LOW | t_{WHRIO} | 100 | – | ns | |
| WE# pulse width | t_{WPIO} | 50 | – | ns | |

WRITE PROTECT

The WRITE PROTECT feature protects the device against inadvertent PROGRAM and ERASE operations. All PROGRAM and ERASE operations are disabled when WP# is LOW. For WRITE PROTECT timing details, see Figures 39 through 42.

Figure 39: ERASE Enable

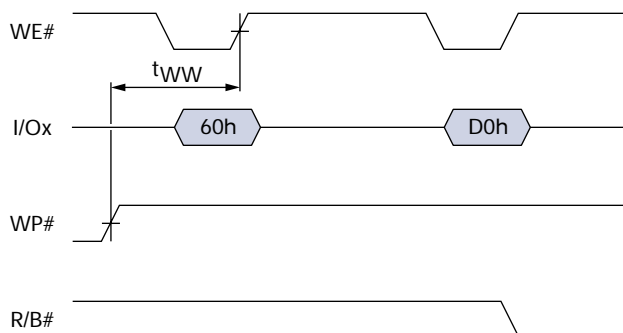


Figure 40: ERASE Disable

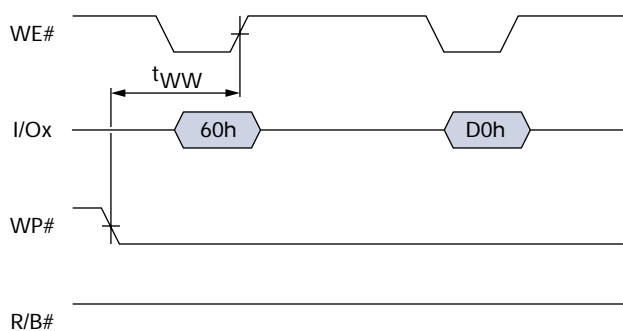


Figure 41: PROGRAM Enable

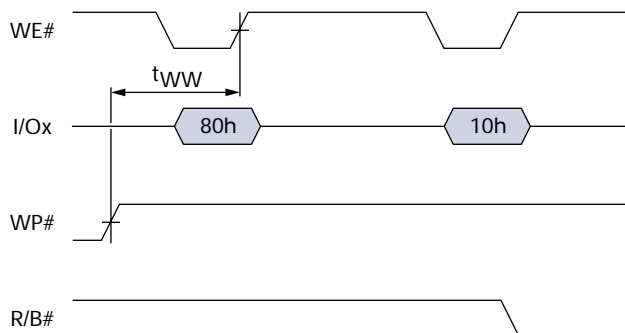
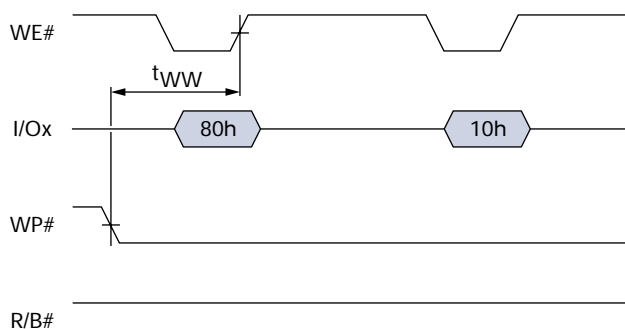


Figure 42: PROGRAM Disable



Error Management

Micron MT29F1Gxx NAND Flash devices are specified to have a minimum of 1,004 valid blocks (NVB) out of 1,024 total available blocks. This means the devices may have blocks that are invalid when they are shipped. An invalid block is one that contains one or more bad bits. Additional bad blocks may develop with use. However, the total number of available blocks will not fall below NVB.

Although NAND Flash memory devices may contain bad blocks, they can be used quite reliably in systems that provide bad-block mapping, replacement, and error correction algorithms. This type of software environment ensures data integrity.

Internal circuitry isolates each block from other blocks, so the presence of a bad block does not affect the operation of the rest of the NAND Flash device.

The first block (physical block address 00h) for each CE# in Micron NAND Flash devices is guaranteed to be free of defects (up to 1,000 PROGRAM/ERASE cycles) when shipped from the factory. This provides a reliable location for storing boot code and critical boot information.

Before NAND Flash devices are shipped from Micron, they are erased. The factory identifies invalid blocks before shipping by programming data other than FFh (x8) or FFFFh (x16) into the first spare location (column address 2,048 for x8 devices, or 1,024 for x16 devices) of the first or second page of each bad block.

System software should check the first spare address on the first and second page of each block prior to performing any erase or formatting operations on the NAND Flash device. A bad-block table can then be created, enabling system software to map around these areas. Factory testing is performed under worst-case conditions. Because blocks marked "bad" may be marginal, it may not be possible to recover this information if the block is erased.

If the NAND Flash device is erased before these operations are performed, system software must determine which blocks are bad by writing and verifying valid information in each memory location in the device. After writing and verifying all locations, the device must be fully erased and checked to verify that each block has erased properly.

Over time, some memory locations may fail to program or erase properly. In order to ensure that data is stored properly over the life of the NAND Flash device, certain precautions must be taken, including:

- Always check status after a WRITE or ERASE operation.
- Under typical use conditions, utilize a minimum of 1-bit ECC for each 528 bytes of data.
- Use a bad-block replacement algorithm.

Electrical Characteristics

Stresses greater than those listed under Absolute Maximum Ratings by Device (see Table 15) 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 above those indicated in the operational sections of this specification is not guaranteed. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

Table 15: Absolute Maximum Ratings by Device

| Device | Symbol | | Min | Max | Unit |
|------------------------------------|------------------|--|------|-------|------|
| MT29F1GxxABB | V _{IN} | Supply voltage on any ball relative to V _{SS} | -0.6 | +2.45 | V |
| MT29F1GxxABB | V _{CC} | | -0.6 | +2.45 | V |
| MT29F1GxxABB | T _{STG} | Storage temperature | -65 | +150 | °C |
| Short circuit output current, I/Os | | | | 5 | mA |

Table 16: Recommended Operating Conditions

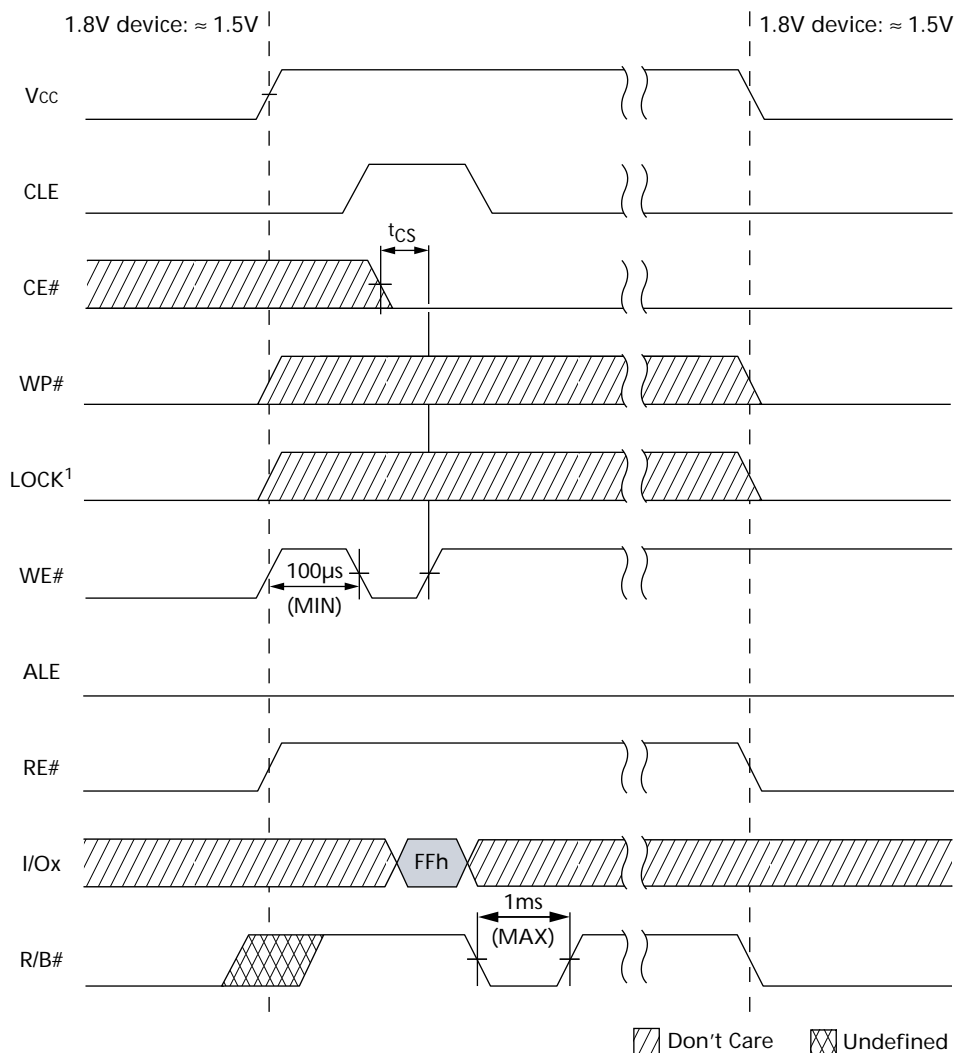
| Parameter/Condition | | Symbol | Min | Typ | Max | Units |
|--------------------------------|--------------|-----------------|------|-----|------|-------|
| Operating temperature | Commercial | T _A | 0 | – | 70 | °C |
| | Extended | T _A | -40 | – | 85 | °C |
| V _{CC} supply voltage | MT29F1GxxABB | V _{CC} | 1.65 | 1.8 | 1.95 | V |
| Supply voltage | | V _{SS} | 0 | 0 | 0 | V |

Vcc Power Cycling

Micron NAND Flash devices are designed to prevent data corruption during power transitions. VCC is internally monitored. (The WP# signal permits additional hardware protection during power transitions.) When VCC reaches 1.5V, a minimum of 100μs should be allowed for the Flash device to initialize before any commands are executed (see Figure 43 for the states of signals during VCC power cycling).

The RESET command must be issued to all CE#s after the NAND Flash device is powered on. Each CE# will be busy for a maximum of 1ms after a RESET command is issued.

Figure 43: AC Waveforms During Power Transitions



Notes: 1. If the system requires the LOCK features to be enabled, then the LOCK signal must be HIGH during power-up. If the LOCK features are to be disabled, then the LOCK signal should be held LOW during power-up.

Table 17: DC and Operating Characteristics, $V_{CC} = 1.65\text{--}1.95\text{V}$

| Parameter | Conditions | Symbol | Min | Typ | Max | Unit |
|---------------------------------------|--|-------------------------|-----------------------------------|-----|-----------------------------------|---------------|
| Sequential READ current | $t_{\text{CYCLE}} = 50\text{ns}$; $\text{CE}\# = \text{VIL}$; $\text{I}_{\text{OUT}} = 0\text{mA}$ | I_{CC1} | – | 10 | 20 | mA |
| PROGRAM current | – | I_{CC2} | – | 10 | 20 | mA |
| ERASE current | – | I_{CC3} | – | 10 | 20 | mA |
| Standby current (TTL) | $\text{CE}\# = \text{V}_{\text{IH}}$; $\text{WP}\# = 0\text{V}/\text{V}_{\text{CC}}$ | I_{SB1} | – | – | 1 | mA |
| Standby current (CMOS) | $\text{CE}\# = \text{V}_{\text{CC}} - 0.2\text{V}$; $\text{WP}\# = 0\text{V}/\text{V}_{\text{CC}}$ | I_{SB2} | – | 10 | 50 | μA |
| Input leakage current | $\text{V}_{\text{IN}} = 0\text{V}$ to V_{CC} | I_{LI} | – | – | ± 10 | μA |
| Output leakage current | $\text{V}_{\text{OUT}} = 0\text{V}$ to V_{CC} | I_{LO} | – | – | ± 10 | μA |
| Input high voltage | I/O [7:0], I/O [15:0], $\text{CE}\#$, CLE , ALE , $\text{WE}\#$, $\text{RE}\#$, $\text{WP}\#$, $\text{R/B}\#$, LOCK | V_{IH} | $0.8 \times \text{V}_{\text{CC}}$ | – | $\text{V}_{\text{CC}} + 0.3$ | V |
| Input low voltage, all inputs | – | V_{IL} | -0.3 | – | $0.2 \times \text{V}_{\text{CC}}$ | V |
| Output high voltage | $\text{I}_{\text{OH}} = -100\mu\text{A}$ | V_{OH} | $\text{V}_{\text{CC}} - 0.1$ | – | – | V |
| Output low voltage | $\text{I}_{\text{OL}} = 100\mu\text{A}$ | V_{OL} | – | – | 0.1 | V |
| Output low current ($\text{R/B}\#$) | $\text{V}_{\text{OL}} = 0.1\text{V}$ | I_{OL} | 3 | 4 | – | mA |

Table 18: Valid Blocks

| Parameter | Symbol | Device | Min | Typ | Max | Unit | Notes |
|--------------------|--------|--------------|-------|-----|-------|--------|-------|
| Valid block number | NvB | MT29F1GxxABB | 1,004 | – | 1,024 | blocks | 1, 2 |

- Notes: 1. Invalid blocks are blocks that contain one or more bad bits. The device may contain bad blocks after shipping. Additional bad blocks may develop over time; however, the total number of available blocks will not drop below NvB during the endurance life of the device. Do not erase or program blocks marked “invalid” by the factory.
2. Block 00h (the first block) is guaranteed to be valid, and does not require error correction for up to 1,000 PROGRAM/ERASE cycles.

Table 19: Capacitance

| Description | Symbol | Device | Max | Unit | Notes |
|--------------------------------|------------------------|--------------|-----|------|-------|
| Input capacitance | C_{IN} | MT29F1GxxABB | 10 | pF | 1, 2 |
| Input/output capacitance (I/O) | C_{IO} | MT29F1GxxABB | 10 | pF | 1, 2 |

- Notes: 1. These parameters are verified in device characterization and are not 100 percent tested.
2. Test conditions: $T_C = 25^\circ\text{C}$; $f = 1\text{ MHz}$; $\text{V}_{\text{IN}} = 0\text{V}$.

Table 20: Test Conditions

| Parameter | Value | Notes |
|--------------------------------|---|-------|
| Input pulse levels | MT29F1GxxABA 0.0V to 1.8V | |
| Input rise and fall times | 5ns | |
| Input and output timing levels | $\text{V}_{\text{CC}}/2$ | |
| Output load | MT29F1GxxABA 1 TTL GATE and $\text{C}_L = 30\text{pF}$ | 1, 2 |

- Notes: 1. Verified on device characterization; not 100 percent tested.
2. Outputs tested at full drive strength.

Table 21: AC Characteristics – Command, Data, and Address Input

| Parameter | Symbol | Min | Max | Unit | Notes |
|----------------------|-----------|-----|-----|------|-------|
| ALE to data start | t_{ADL} | 100 | – | ns | 1 |
| ALE hold time | t_{ALH} | 10 | – | ns | |
| ALE setup time | t_{ALS} | 25 | – | ns | |
| CE# hold time | t_{CH} | 10 | – | ns | |
| CLE hold time | t_{CLH} | 10 | – | ns | |
| CLE setup time | t_{CLS} | 25 | – | ns | |
| CE# setup time | t_{CS} | 25 | – | ns | |
| Data hold time | t_{DH} | 10 | – | ns | |
| Data setup time | t_{DS} | 20 | – | ns | |
| WRITE cycle time | t_{WC} | 45 | – | ns | |
| WE# pulse width HIGH | t_{WH} | 15 | – | ns | |
| WE# pulse width | t_{WP} | 25 | – | ns | |
| WP# setup time | t_{WW} | 30 | – | ns | |

Notes: 1. Timing for t_{ADL} begins in the ADDRESS cycle, on the final rising edge of WE#, and ends with the first rising edge of WE# data output.

Table 22: AC Characteristics – Normal Operation

| Parameter | Symbol | Min | Max | Unit | Notes |
|---|---------------|---------------|--------------------|---------|---------|
| ALE to RE# delay | t_{AR} | 10 | – | ns | 1 |
| CE# access time | t_{CEA} | – | 45 | ns | 1 |
| CE# HIGH to output High-Z | t_{CHZ} | – | 45 | ns | 1,2 |
| CLE to RE# delay | t_{CLR} | 10 | – | ns | 1 |
| CE# HIGH to output hold | t_{COH} | 15 | – | ns | 1 |
| Cache busy in PAGE READ CACHE MODE (first 31h) | $t_{DCBSYR1}$ | – | 3 | μ s | 1 |
| Cache busy in PAGE READ CACHE MODE (next 31h and 3Fh) | $t_{DCBSYR2}$ | $t_{DCBSYR1}$ | 25 | μ s | 1 |
| Output High-Z to RE# LOW | t_{IR} | 0 | – | ns | 1,2 |
| Data transfer from Flash array to data register | t_{R} | – | 25 | μ s | 1 |
| READ cycle time | t_{RC} | 50 | – | ns | 1, 3 |
| RE# access time | t_{REA} | – | 30 | ns | 1 |
| RE# HIGH hold time | t_{REH} | 15 | – | ns | 1 |
| RE# HIGH to output hold | t_{RHOH} | 15 | – | ns | 1 |
| RE# HIGH to WE# LOW | t_{RHW} | 100 | – | ns | 1 |
| RE# HIGH to output High-Z | t_{RHZ} | – | 100 | ns | 1, 2 |
| RE# pulse width | t_{RP} | 25 | – | ns | 1 |
| Ready to RE# LOW | t_{RR} | 20 | – | ns | 1 |
| Reset time (READ/PROGRAM/ERASE/power-up) | t_{RST} | – | 5/10/500/ 1,000 | μ s | 1, 4 |
| WE# HIGH to busy | t_{WB} | – | 100 | ns | 1, 4, 5 |
| WE# HIGH to RE# LOW | t_{WHR} | 80 | – | ns | 1 |

- Notes: 1. AC characteristics may need to be relaxed if I/O drive strength is not set to full.
2. Transition is measured ± 200 mV from steady-state voltage with load. This parameter is sampled and not 100 percent tested.
3. When V_{CC} is less than 1.7V down to 1.65V, t_{RC} MIN is 60ns.
4. If RESET (FFh) command is loaded at ready state, the device goes busy for maximum 5 μ s.
5. Do not issue a new command during t_{WB} , even if R/B# is ready.

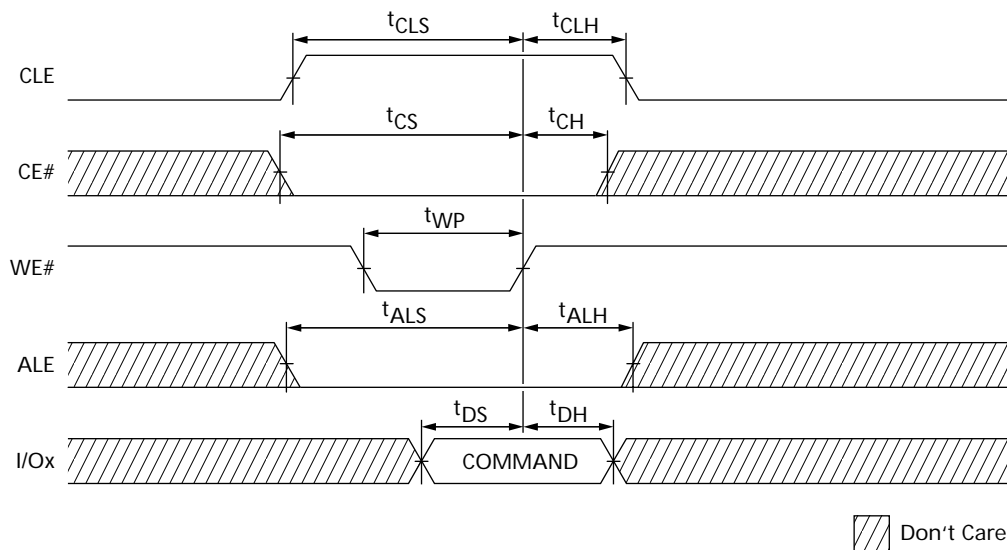
Table 23: PROGRAM/ERASE Characteristics

| Parameter | Symbol | Typ | Max | Unit | Notes |
|--|-------------|-----|-----|---------|-------|
| Number of partial page programs | NOP | – | 8 | cycle | 1 |
| Block erase time | t_{BERS} | 2 | 3 | ms | |
| Busy time for cache program | t_{CBSY} | 3 | 700 | μs | 2 |
| Busy time for PROGRAM ERASE on locked block | t_{LBSY} | | 3 | μs | |
| Busy time for OTP DATA PROGRAM operation if OTP is protected | t_{OBSY} | | 30 | μs | |
| Last page program time | t_{LPROG} | – | – | – | 3 |
| Page program time | t_{PROG} | 250 | 700 | μs | 4 |

- Notes: 1. Eight total to the same page.
2. t_{CBSY} MAX time depends on timing between internal program completion and data in.
3. $t_{LPROG} = t_{PROG}$ (last page) + t_{PROG} (last - 1 page) - command load time (last page) - address load time (last page) - data load time (last page).
4. More than 50 percent of the pages will meet typical t_{PROG} at 1.8V and 25°C.

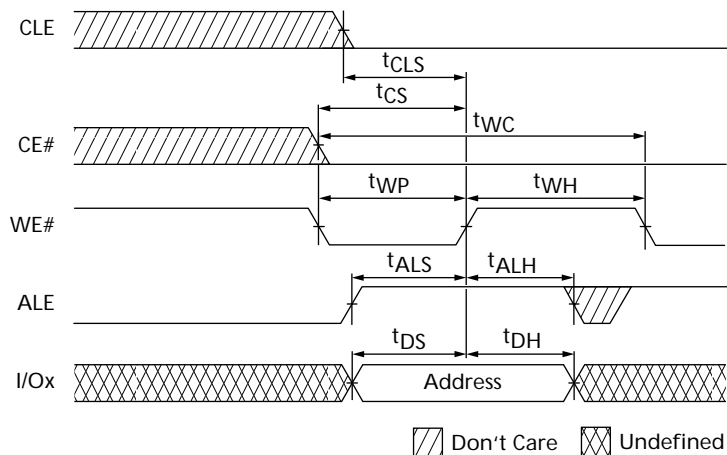
Timing Diagrams

Figure 44: COMMAND LATCH Cycle



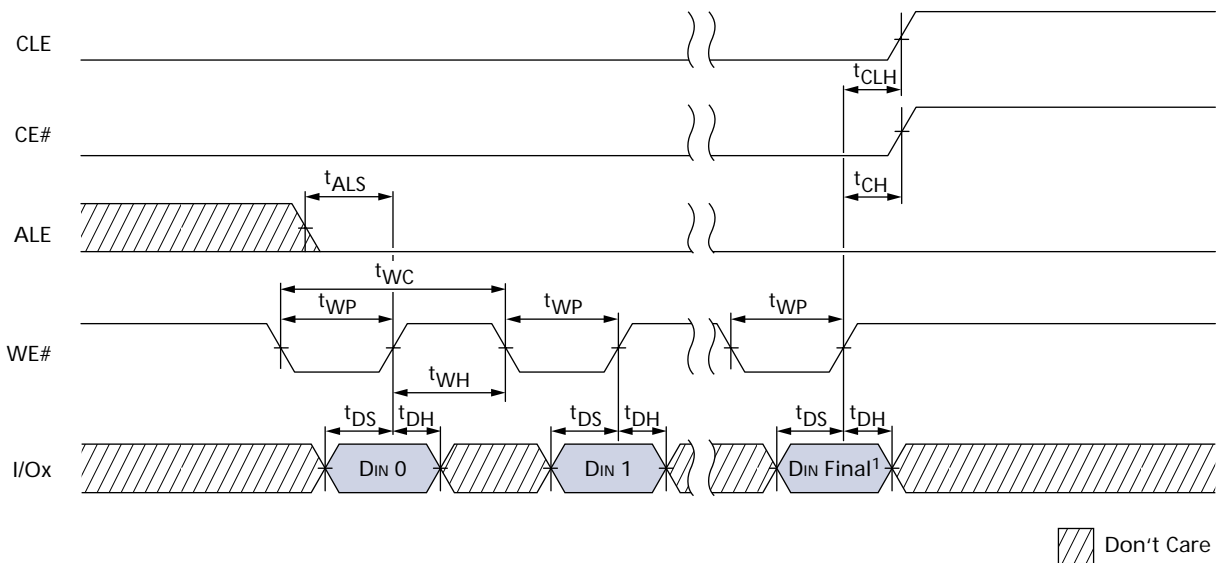
Note: The x16 devices must have I/O[15:8] set to "0."

Figure 45: ADDRESS LATCH Cycle



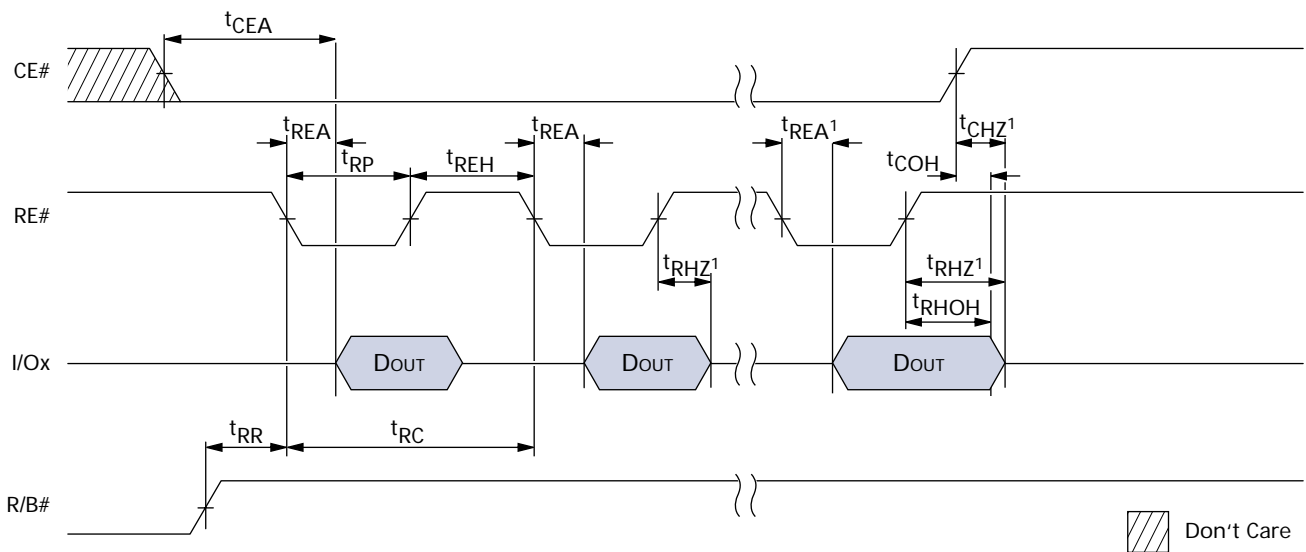
Note: The x16 devices must have I/O[15:8] set to "0."

Figure 46: INPUT DATA LATCH Cycle



Note: DIN Final = 2,112 (x8) or 1,056 (x16).

Figure 47: SERIAL ACCESS Cycle after READ



Note: Transition is measured $\pm 200\text{mV}$ from steady-state voltage with load.
This parameter is sampled and not 100 percent tested.

Figure 48: READ STATUS Cycle

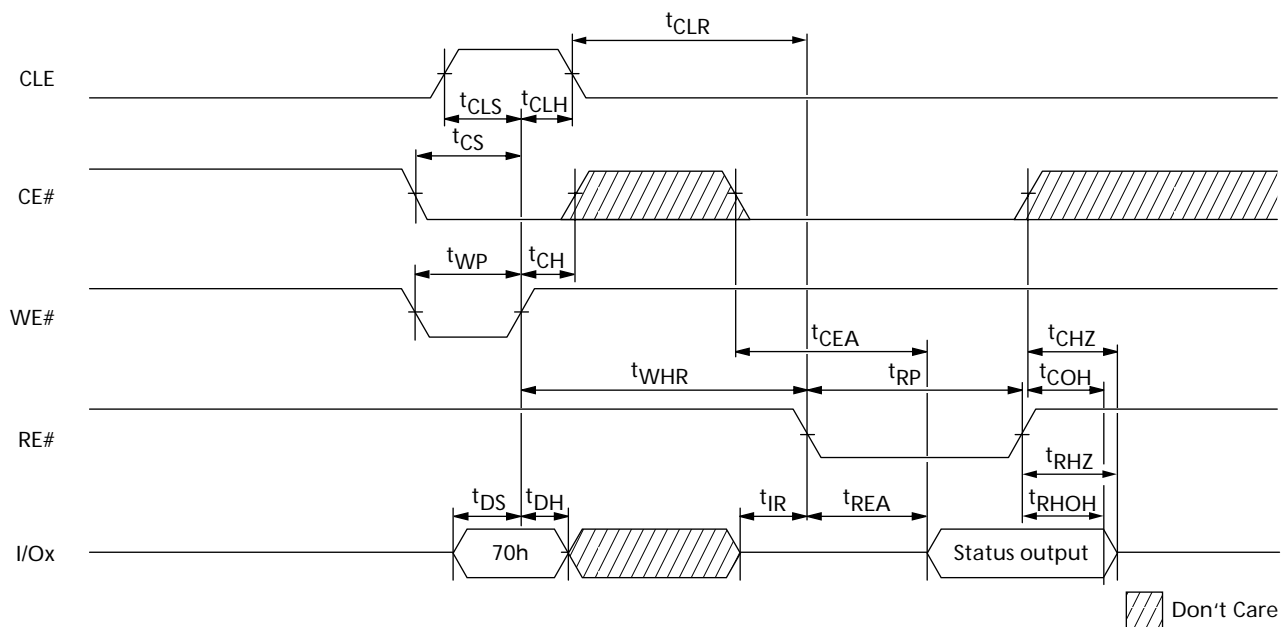


Figure 49: PAGE READ Operation

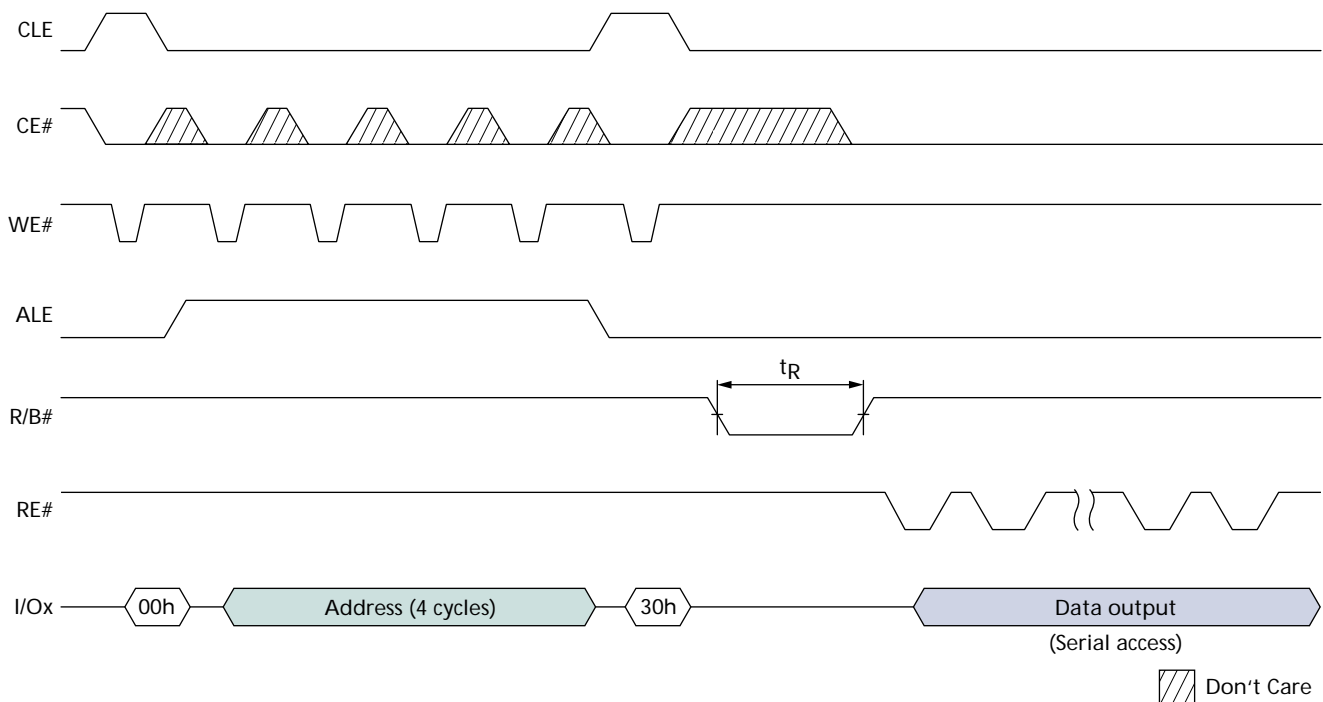


Figure 50: READ Operation with CE# "Don't Care"

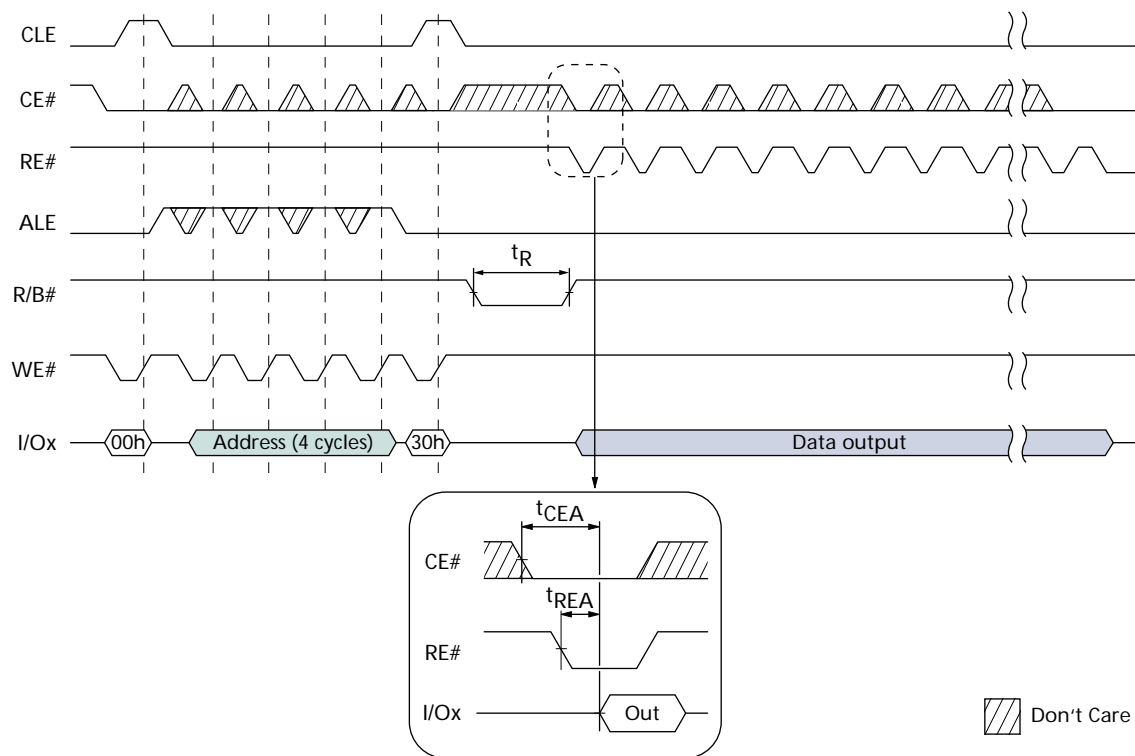


Figure 51: RANDOM DATA READ Operation

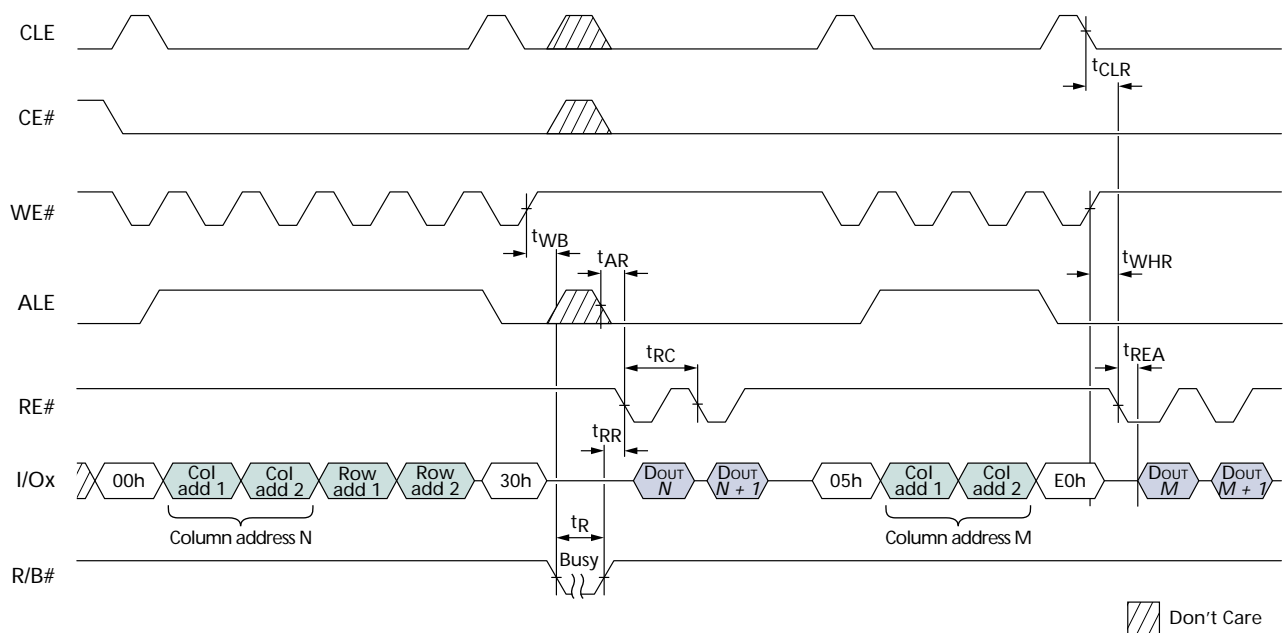


Figure 52: PAGE READ CACHE MODE Operation, Part 1 of 2

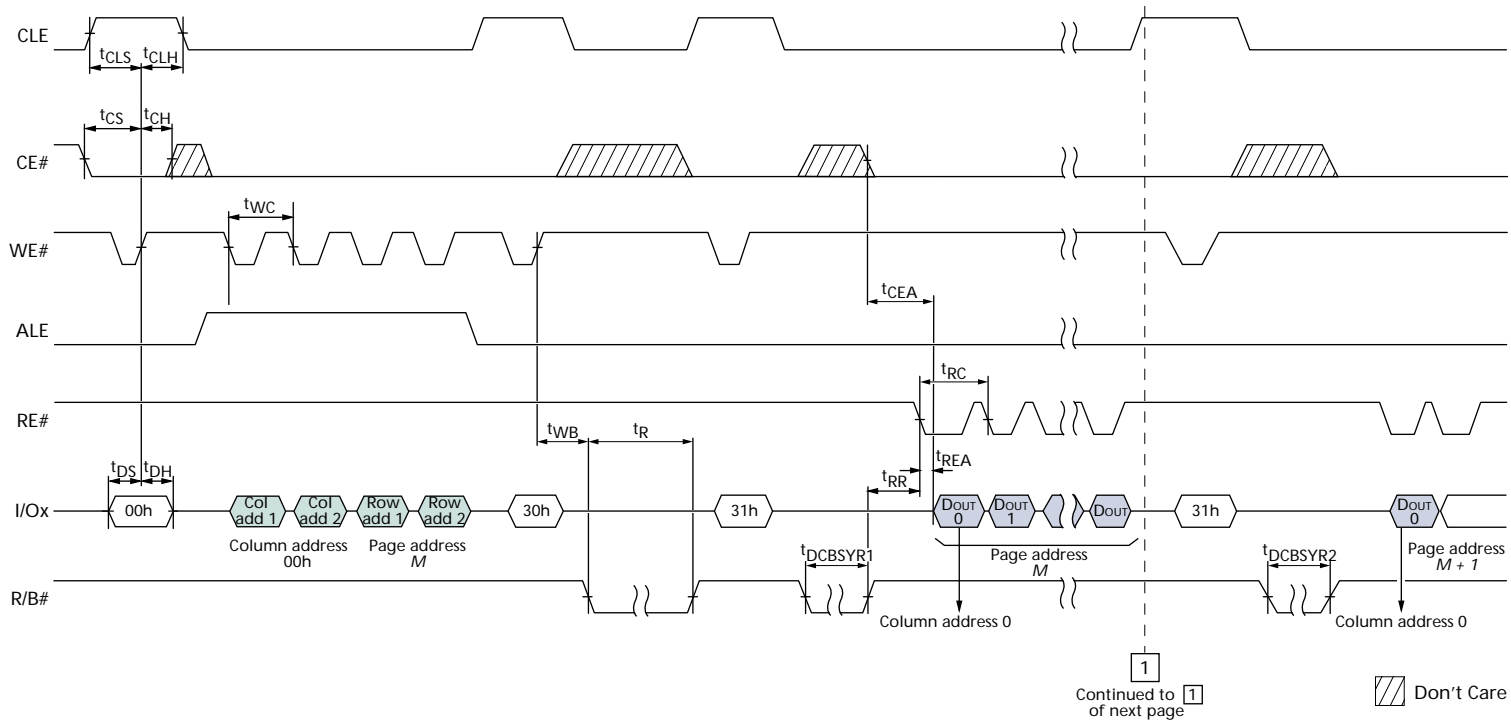


Figure 53: PAGE READ CACHE MODE Operation, Part 2 of 2

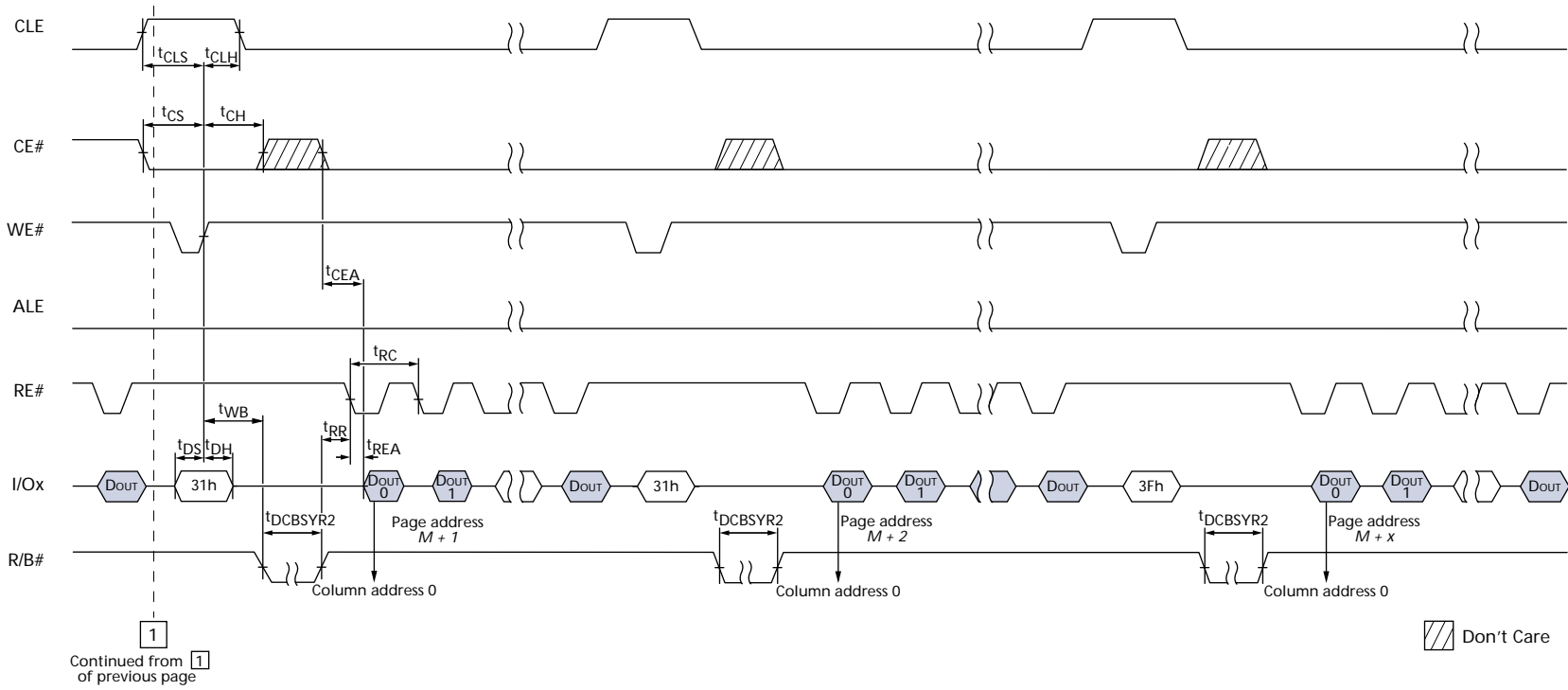


Figure 54: PAGE READ CACHE MODE Operation without R/B#, Part 1 of 2

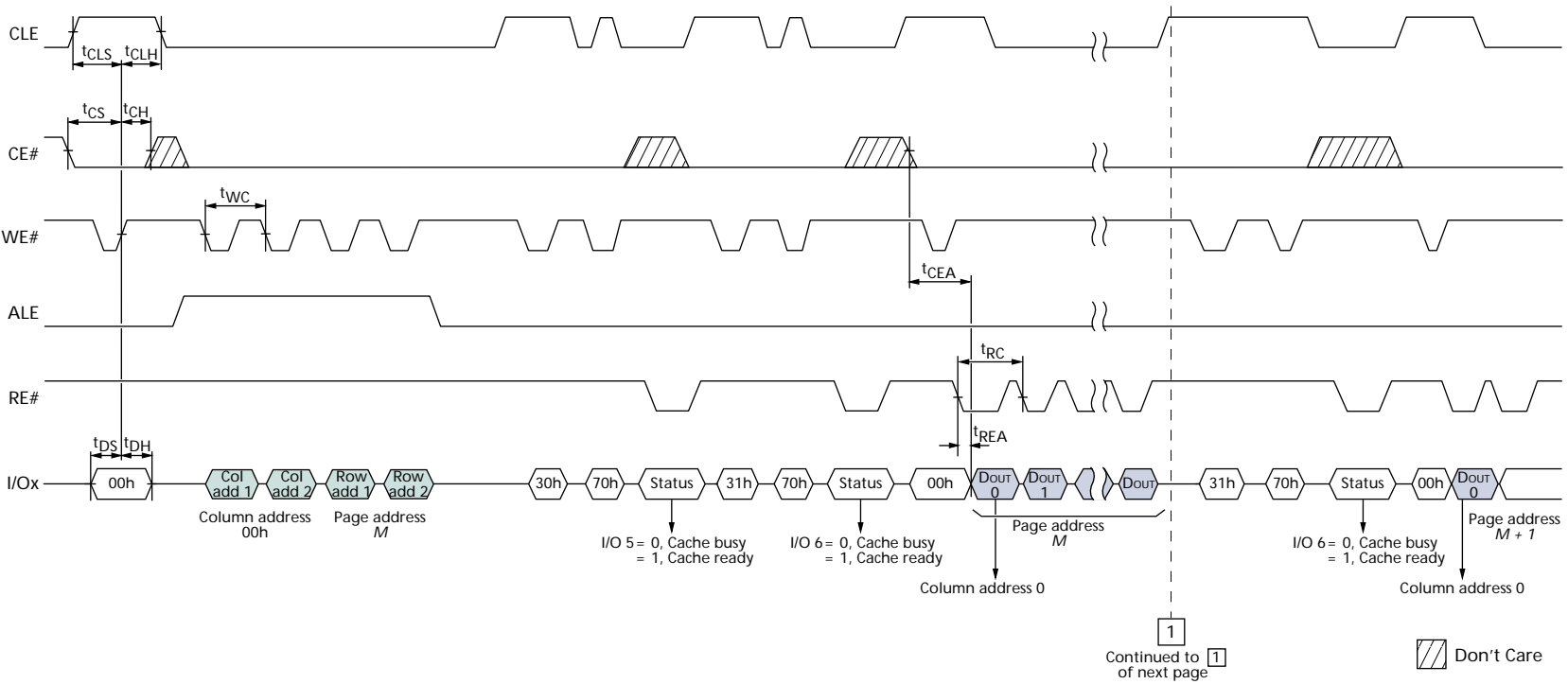


Figure 55: PAGE READ CACHE MODE Operation without R/B#, Part 2 of 2

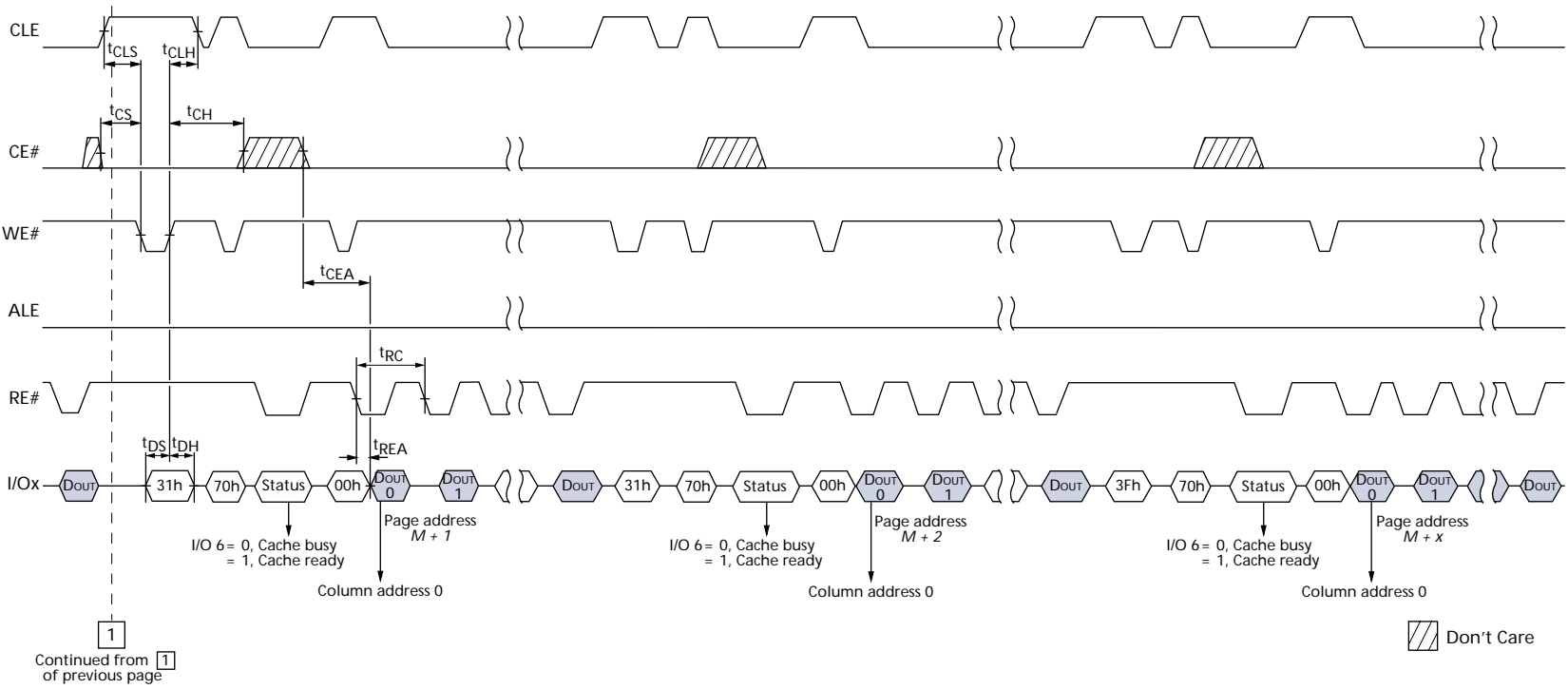
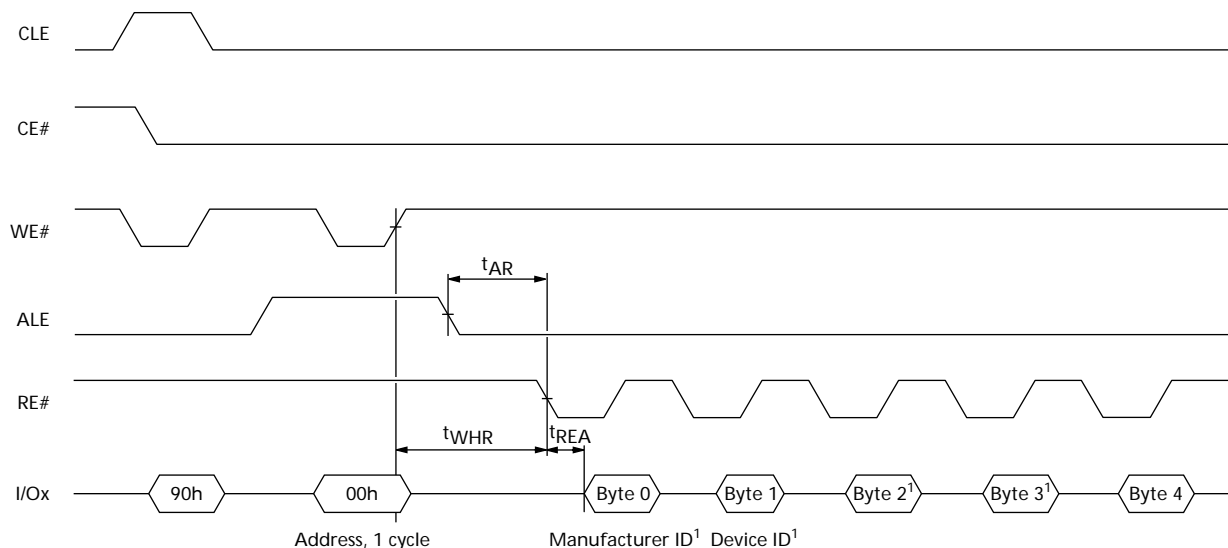


Figure 56: READ ID Operation



Notes: 1. See Table 8 on page 25 for byte definitions.

Figure 57: PROGRAM Operation with CE# "Don't Care"

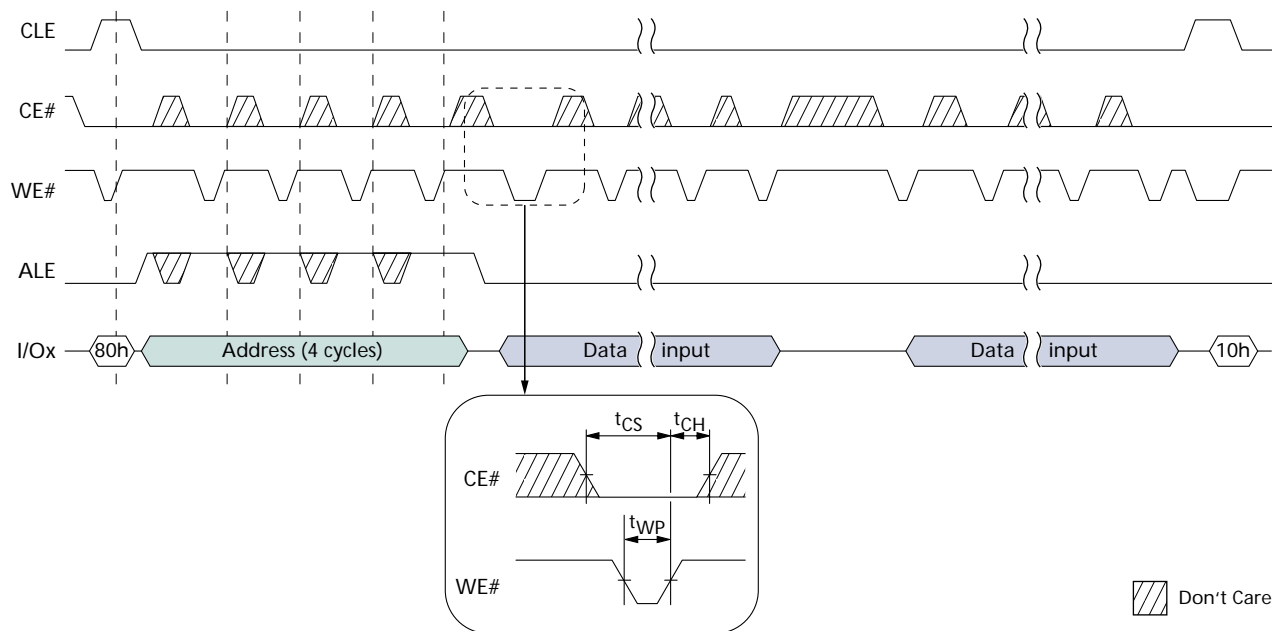


Figure 58: PROGRAM PAGE Operation

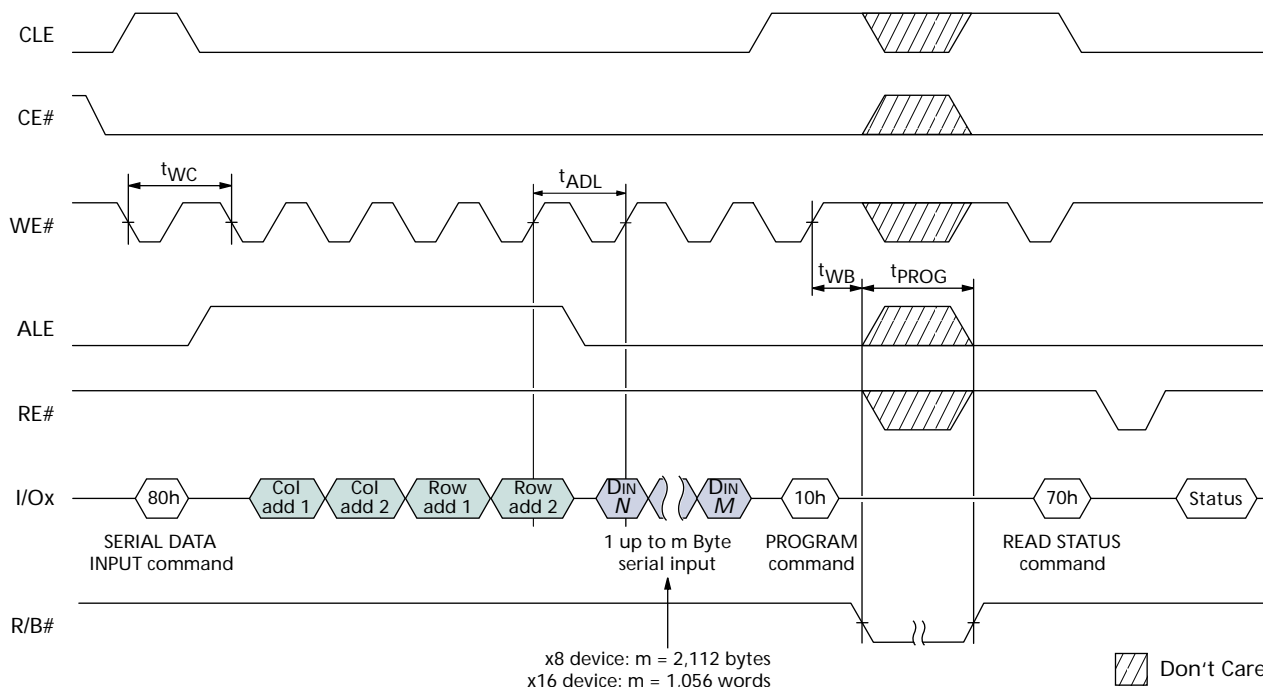


Figure 59: PROGRAM PAGE Operation with RANDOM DATA INPUT

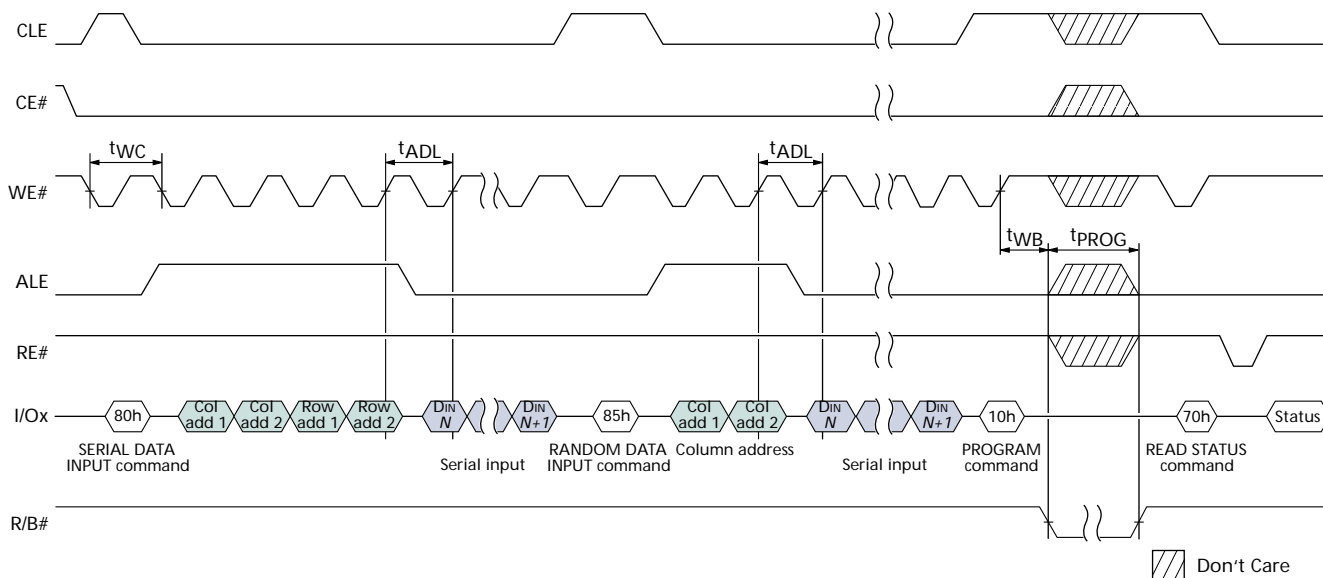


Figure 60: INTERNAL DATA MOVE Operation

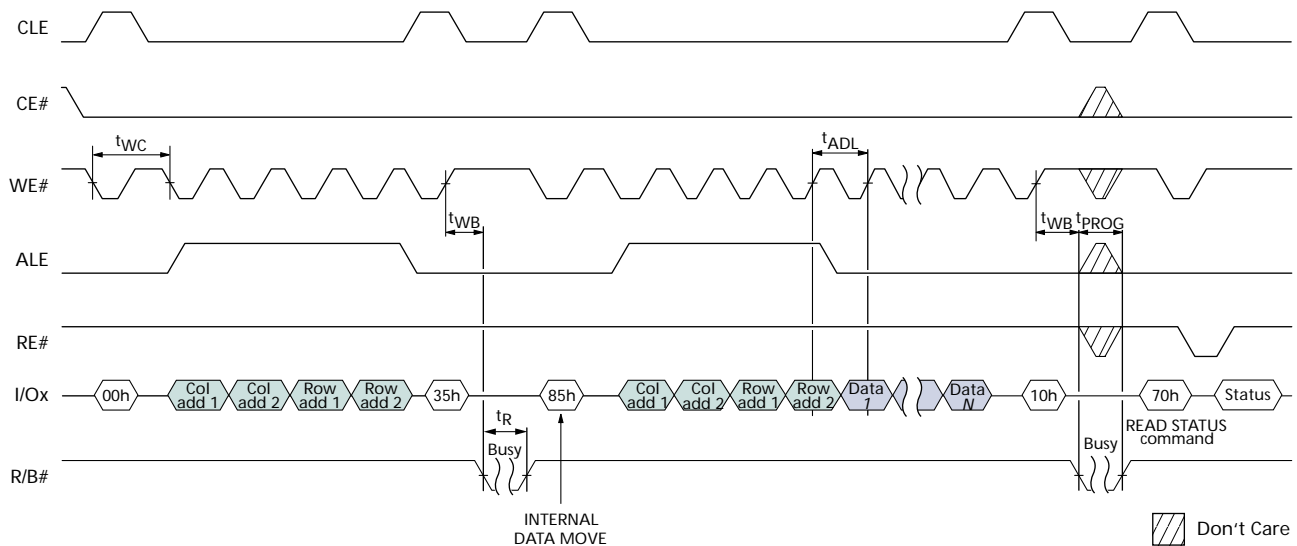


Figure 61: PROGRAM PAGE CACHE MODE Operation

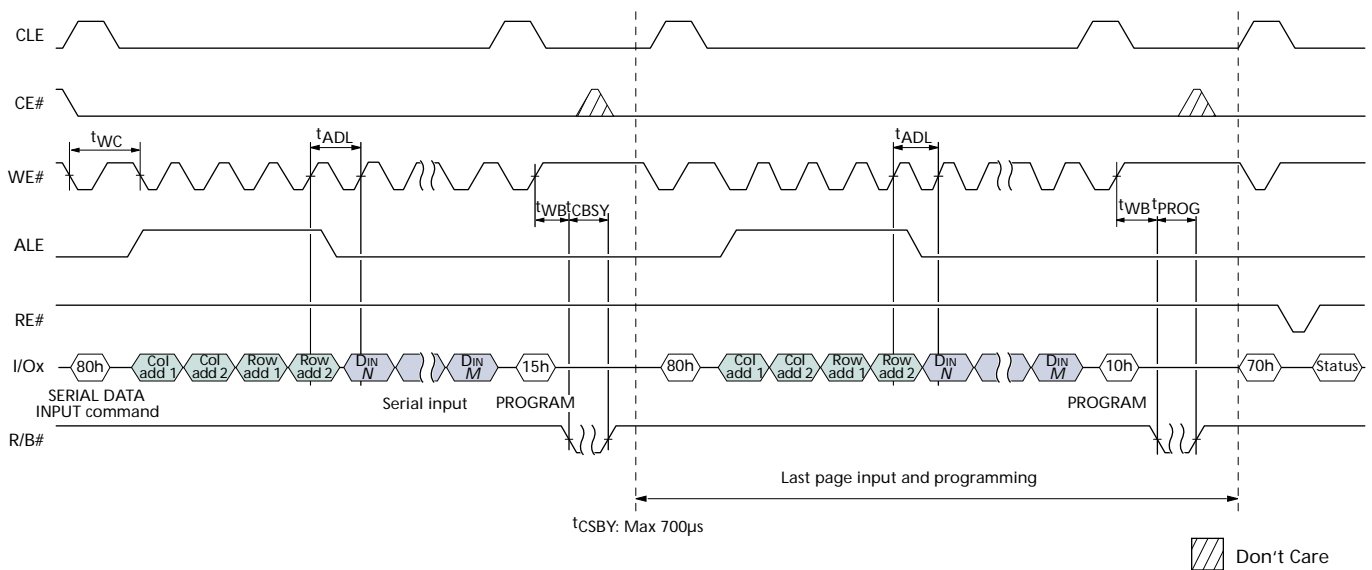


Figure 62: PROGRAM PAGE CACHE MODE Operation Ending on 15h

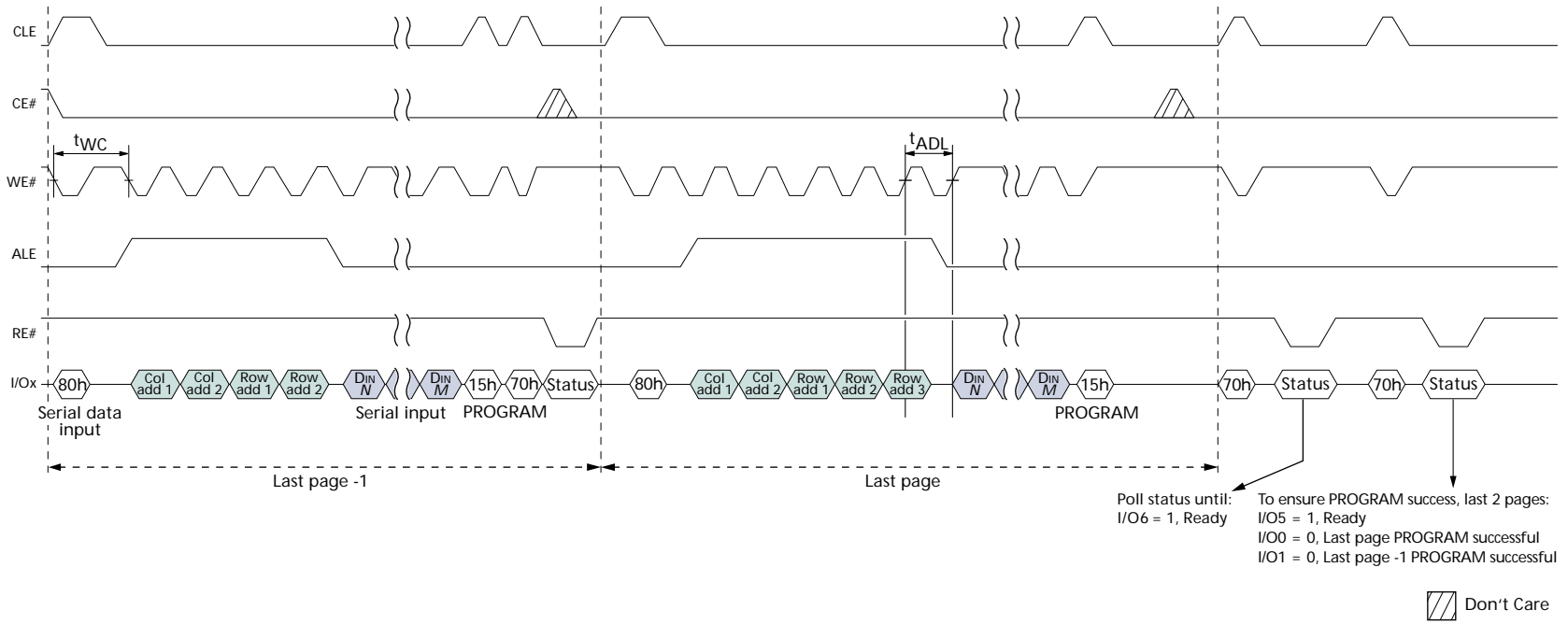


Figure 63: BLOCK ERASE Operation

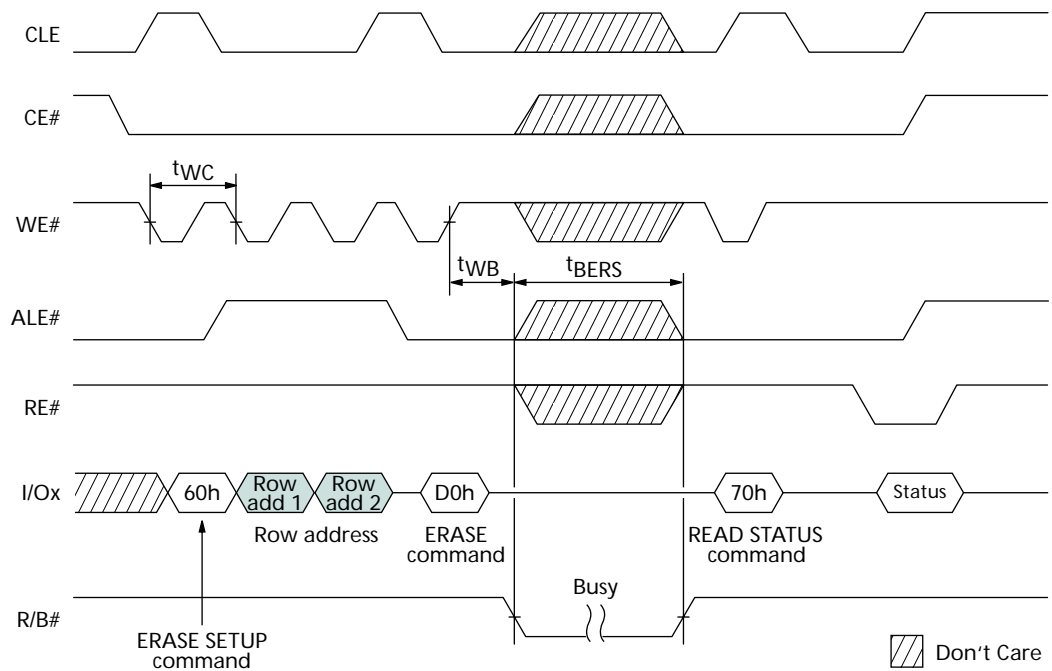


Figure 64: RESET Operation

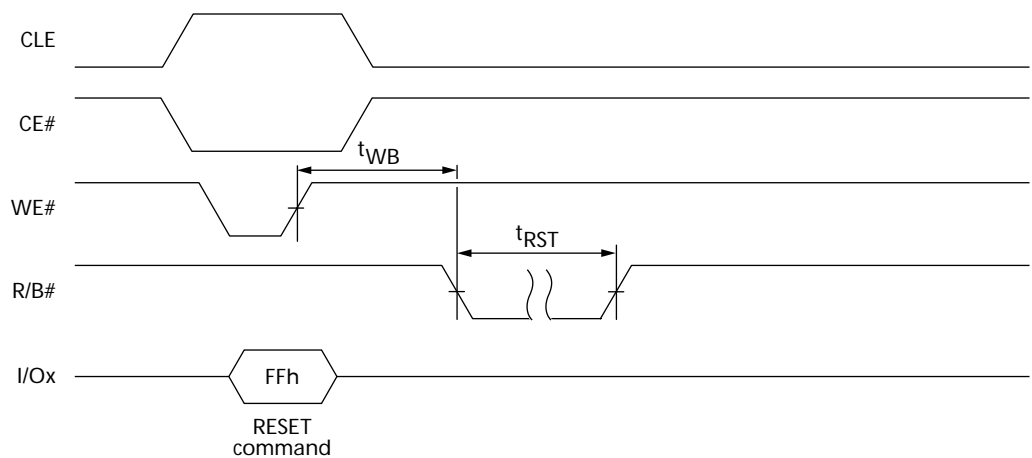


Figure 65: 63-Ball VFPGA Package



Revision History

| | |
|---|-------|
| Rev. E | 1/08 |
| <ul style="list-style-type: none"> • “OTP DATA READ AFh-30h” on page 38: Added comment regarding OTP DATA READ followed by PAGE READ CACHE MODE. | |
| Rev. D | 6/07 |
| <ul style="list-style-type: none"> • “PROGRAM PAGE CACHE MODE 80h-15h” on page 31: Revised last paragraph. | |
| Rev. C | 5/07 |
| <ul style="list-style-type: none"> • Figure 2: Ball Assignment (x8), 63-Ball VFBGA on page 8: Renumbered ball assignments. • Figure 3: Ball Assignment (x16), 63-Ball VFBGA on page 9: Renumbered ball assignments. • Former Figure 10: Time Constants on page 17: Converted figure to equation format. • Former Figure 11: Minimum Rp on page 18: Converted figure to equation format. • “OTP DATA PROGRAM A0h-10h” on page 35: Revised RANDOM DATA INPUT (85h) discussion. • “Error Management” on page 50: Modified second bullet point wording. • VCC Power Cycling and Figure 43: AC Waveforms During Power Transitions on page 52: Changed 10µs to 100µs. • Table 22: AC Characteristics – Normal Operation on page 54: Removed ^tRLOH. | |
| Rev. B | 11/06 |
| <ul style="list-style-type: none"> • “Features” on page 4: Added required RESET after power-up; changed ready/busy pin to ready/busy signal. • Table 22: AC Characteristics – Normal Operation on page 54: Deleted ^tLBSY and ^tOBSY MIN values and notes; changed ^tOBSY (MAX) to 30µs; moved ^tLBSY and ^tOBSY to table 23. • Table 23: PROGRAM/ERASE Characteristics on page 55: Changed ^tPROG (TYP) to 250µs and added note 4. | |
| Rev. A | 10/06 |
| <ul style="list-style-type: none"> • Initial release. | |