# NAND Flash Memory

(AX20NV1G8, AX20NV1G6)

## Features

- Interface
  - Open NAND Flash Interface (ONFI 1.0) compliant
  - x8, x16
- Technology
  - Single-level cell (SLC)
  - 3xnm NAND Process
- Operating Voltage Range
  - VCC: 2.70V – 3.60V
- Operating Temperature Range
  - Industrial: -40°C to 85°C
- Packages
  - 48-pin TSOP (12.0mm x 20mm)
  - 63-ball FBGA (9mm x 11mm)
- Device Signature
  - Manufacturer's ID
  - Device ID
  - Device Parameters (ONFI)
  - Unique ID
- One Time Programmable Area (OTP)
  - One Block (128K + 4K bytes)
- Quality and Reliability
  - Recommended Error Correction Code: 4-bit / 528 bytes of data
  - Data retention: 10 years
  - Endurance (P/E cycles): 100K (Typ.)
  - Block zero (block address 00h) is a valid block when shipped from factory and will remain valid for at least 1K P/E cycles with ECC
- Memory Array Organization
  - x8
    - Page size: 2112 bytes (2048 + 64 bytes)
  - x16
    - Page size: 1056 words (1024 + 32 words)
  - Block size: 64 pages (128K + 4K bytes)
  - Device (Plane) size: 1024 blocks
- Data Protection
  - WP# signal: write protect entire device
- Device Status
  - Ready/busy# (R/B#) signal: hardware method for detecting internal operation completion status
- Advanced Command Set
  - Program page cache mode
  - Read page cache mode
  - Data move (page copy back): Internal data move

## Performance

| Device Operation | Values | Units |
|---|---|---|
| **Read page** | 25.0 (Maximum) | µs |
| **Program Page** | 300.0 (Typical) | µs |
| **Block Erase** | 3.0 (Typical) | ms |
| **Standby** | 10.0 (Typical) | µA |
| **Read page** | 15.0 (Typical) | mA |

# Table of Contents

# General Description

Axia Memory Technology's AX20NV1Gx is a 3.0V 1 Gbit NAND Flash organized as 2112 bytes × 64 pages × 1024 blocks. All read and program operations are performed using a 2112-byte register; allowing data to be transferred to and from the memory array in 2112-bytes increments. The erase operation is implemented in a single block unit (2112 bytes × 64 pages). ECC is required for higher data reliability.

Axia's NAND Flash devices communicate through an asynchronous ONFI 1.0 compatible interface for high-performance I/O operations. It is a multiplexed 8-bit/16-bit bus (I/Ox) to transfer commands, address, and data. There are five control signals used to implement the asynchronous data interface: CE#, CLE, ALE, WE#, and RE#. Additional signals control hardware write protection (WP#) and monitor device status (R/B#). This hardware interface creates a low pin-count device with a standard pinout that remains the same in Axia's NAND device family, enabling future upgrades to higher densities without board redesign.

Additionally, Axia's NAND Flash devices support a copy back function which optimizes management of defective blocks. When a page program operation fails, the data already loaded in the page buffer can be directly programmed to another page inside the same array section without the time-consuming serial data insertion phase.

Also present in Axia's NAND Flash devices is a cache read feature that increases the read throughput. During cache reading, the device loads the new data in a cache register while the previous data is transferred to the I/Os.

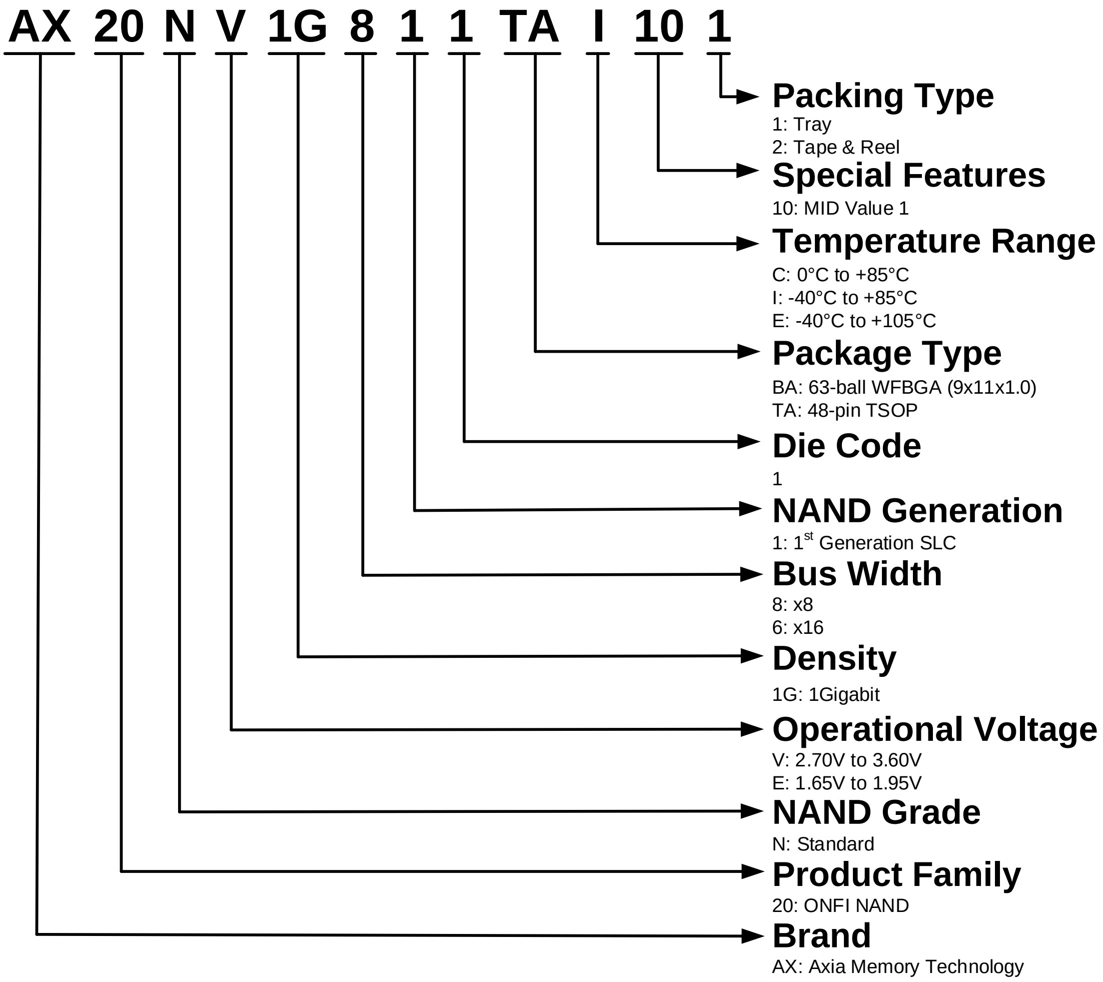
And finally, Axia's NAND Flash devices provide special features listed below in Table 1:

*Table 1: Special Features*

| # | Feature Description | Details |
|---|---|---|
| **1** | One-Time Programmable Area | Size: 1 block (128K + 4K bytes) |
| **2** | Unique Identifier | Size: 16 bytes |

# Ordering Options

The ordering part numbers are firmed by a valid combination of the following options:



AX 20 N V 1G 8 1 1 TA I 10 1

**Packing Type**
1: Tray
2: Tape & Reel

**Special Features**
10: MID Value 1

**Temperature Range**
C: 0°C to +85°C
I: -40°C to +85°C
E: -40°C to +105°C

**Package Type**
BA: 63-ball WFBGA (9x11x1.0)
TA: 48-pin TSOP

**Die Code**
1

**NAND Generation**
1: 1st Generation SLC

**Bus Width**
8: x8
6: x16

**Density**
1G: 1Gigabit

**Operational Voltage**
V: 2.70V to 3.60V
E: 1.65V to 1.95V

**NAND Grade**
N: Standard

**Product Family**
20: ONFI NAND

**Brand**
AX: Axia Memory Technology

## Valid Combinations — Standard

Valid Combinations list includes device configurations currently available. Contact your local sales office to confirm availability of specific valid combinations and to check on newly released combinations.

*Table 2: Valid Combinations List*

| Valid Combinations | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Base Part Number** | **Bus Width** | **Generation** | **Die Code** | **Package type** | **Temperature Range** | **Special Features** | **Packing Type** | **Part Number** |
| **AX20NV1G** | x8 | 1 | 1 | BA, TA | I | 10 | 1, 2 | AX20NV1G811BAI101 |
| | | | | | | | | AX20NV1G811TAI101 |
| | | | | | | | | AX20NV1G811BAI102 |

# Signal Description and Assignment

*Figure 1: Device Pinout*

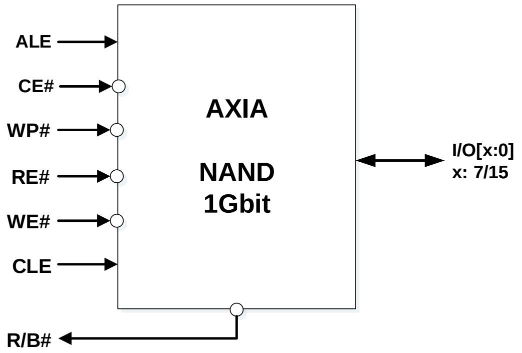
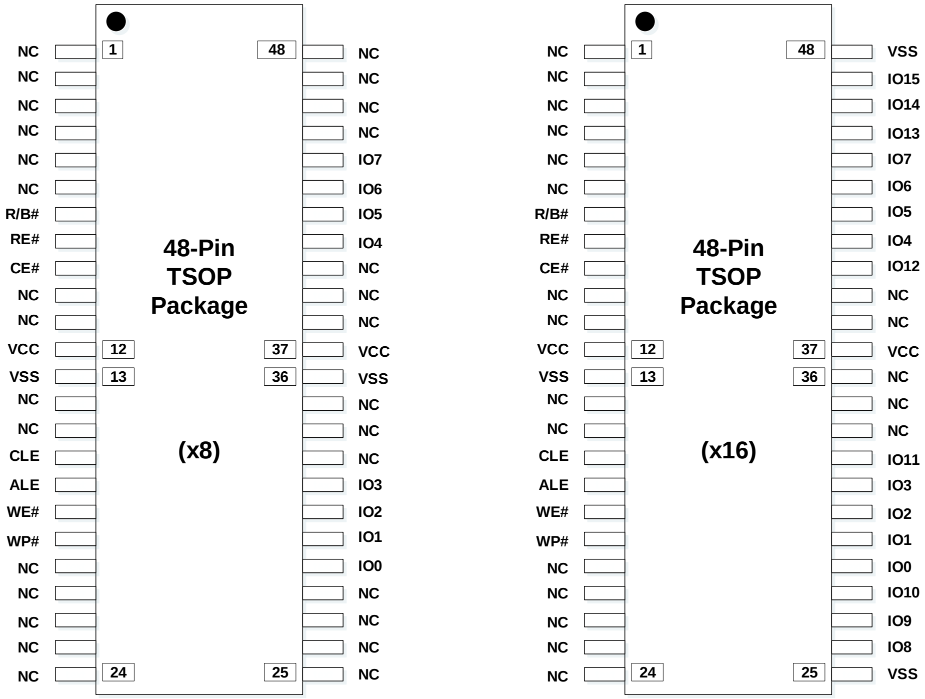


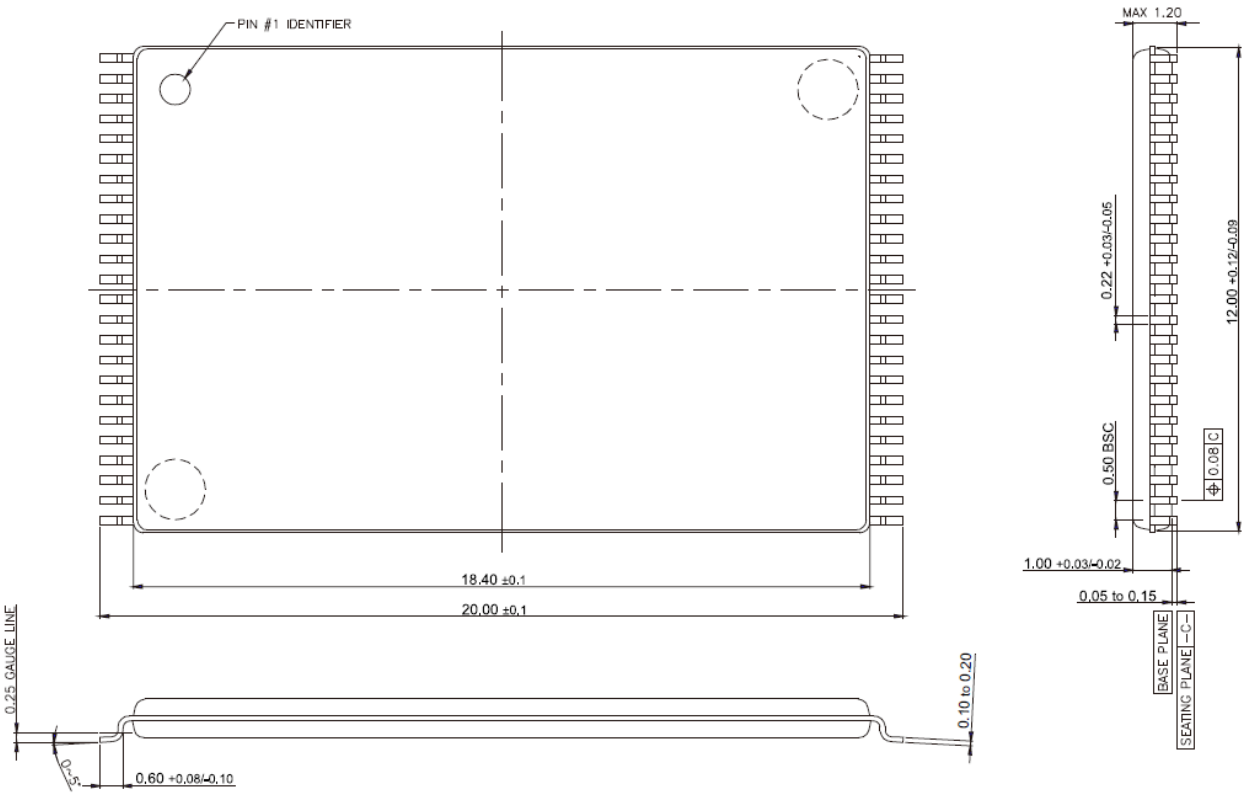
*Table 3: Signal Description*

| Signal | Type | Description |
|---|---|---|
| **CE#** | Input | **Chip enable:** Enables or disables the NAND Flash. |
| **CLE** | Input | **Command latch enable:** Loads a command from x8 - I/O[7:0], x16 - I/O[15:0] into the command register. |
| **ALE** | Input | **Address latch enable:** Loads an address from x8 - I/O[7:0], x16 - I/O[15:0] into the address register. |
| **WE#** | Input | **Write enable:** Transfers commands, addresses, and serial data from the host system to the NAND Flash. |
| **RE#** | Input | **Read enable:** Transfers serial data from the NAND Flash to the host system. |
| **WP#** | Input | **Write protect:** Enables or disables NAND Flash memory array program and erase operations. |
| **R/B#** | Output | **Ready/busy:** An open-drain, active-low output that requires an external pull-up resistor. This signal indicates NAND Flash activity. |
| **I/O [7:0] - x8**<br>**I/O [15:0] - x16** | Input / Output | **Data inputs/outputs:** The bidirectional I/Os transfer address, data, and command information. |
| **$V_{CC}$** | Supply | **$V_{CC}$:** Core and I/O power supply. |
| **$V_{SS}$** | Supply | **$V_{SS}$:** Core and I/O ground supply. |
| **NC** | | **No connect:** NCs are not internally connected. They can be driven ($V_{CC}$/$V_{SS}$) or left unconnected. |
| **DNU** | | **Do not use:** DNUs must be left unconnected. |

# Package Options

## 48-Pin TSOP (Top View)





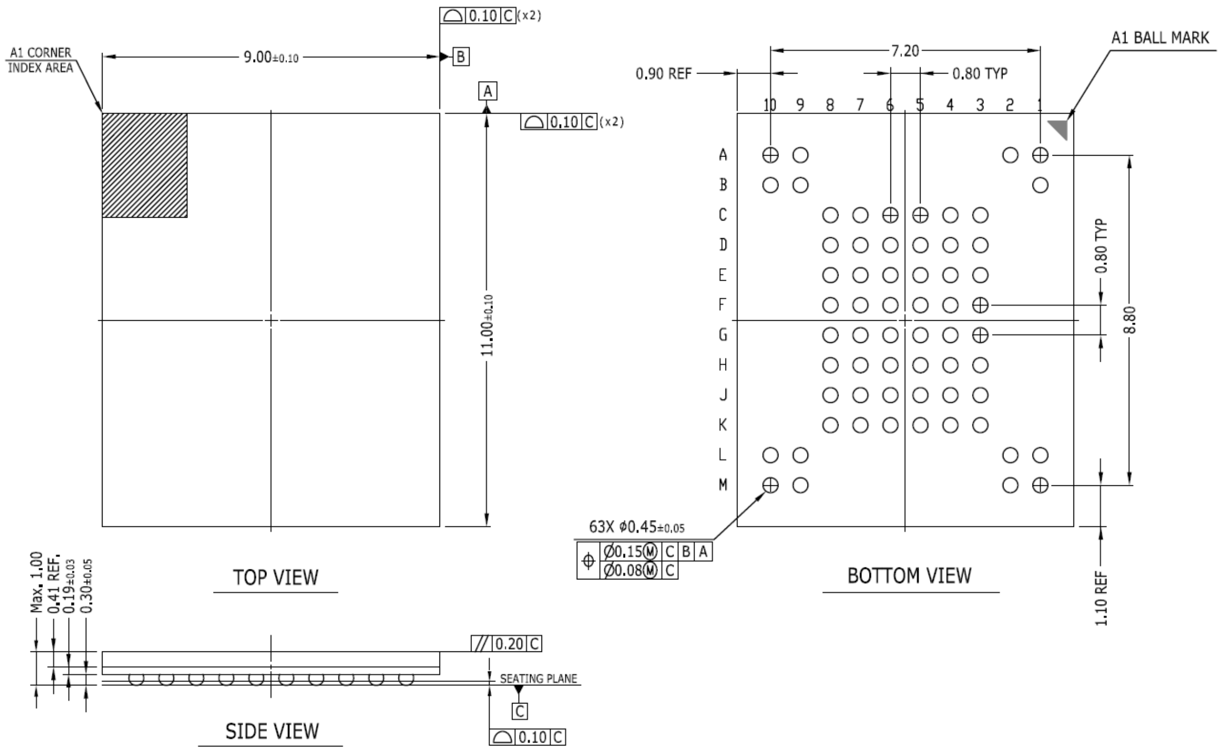
## 63-Ball FBGA (Balls Down, Top View)

(x8)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | NC | NC | | | | | | | NC | NC |
| B | NC | | | | | | | | NC | NC |
| C | | | WP# | ALE | VSS | CE# | WE# | R/B# | | |
| D | | | NC | RE# | CLE | NC | NC | NC | | |
| E | | | NC | NC | NC | NC | NC | NC | | |
| F | | | NC | NC | NC | NC | NC | NC | | |
| G | | | NC | NC | NC | NC | NC | NC | | |
| H | | | NC | IO0 | NC | NC | NC | VCC | | |
| J | | | NC | IO1 | NC | VCC | IO5 | IO7 | | |
| K | | | VSS | IO2 | IO3 | IO4 | IO6 | VSS | | |
| L | NC | NC | | | | | | | NC | NC |
| M | NC | NC | | | | | | | NC | NC |

(x16)

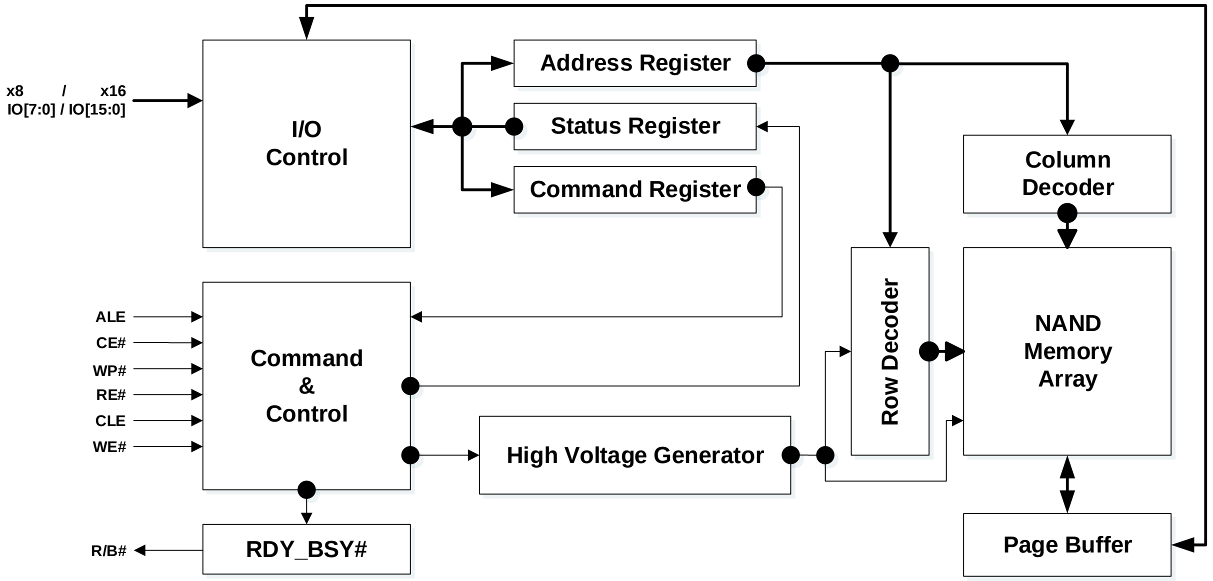
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | NC | NC | | | | | | | NC | NC |
| B | NC | | | | | | | | NC | NC |
| C | | | WP# | ALE | VSS | CE# | WE# | R/B# | | |
| D | | | NC | RE# | CLE | NC | NC | NC | | |
| E | | | NC | NC | NC | NC | NC | NC | | |
| F | | | NC | NC | NC | NC | NC | NC | | |
| G | | | NC | NC | NC | IO13 | IO15 | NC | | |
| H | | | IO8 | IO0 | IO10 | IO12 | IO14 | VCC | | |
| J | | | IO9 | IO1 | IO11 | VCC | IO5 | IO7 | | |
| K | | | VSS | IO2 | IO3 | IO4 | IO6 | VSS | | |
| L | NC | NC | | | | | | | NC | NC |
| M | NC | NC | | | | | | | NC | NC |



# Architecture

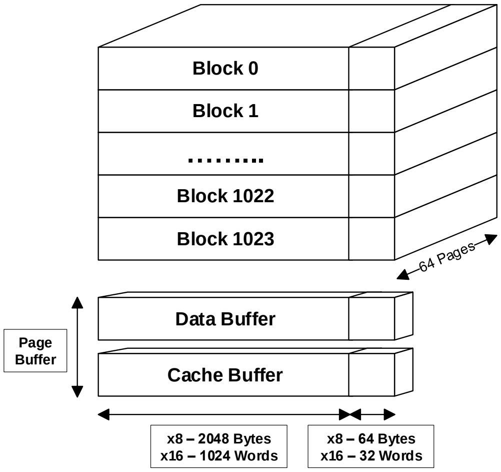
Axia's NAND Flash devices use ONFI 1.0 compatible interface for all operations. Data, commands, and addresses are multiplexed onto the I/O pins. The commands received at the I/Os are latched by a command register and are used to determine the operations the device must perform. The addresses are latched into an address register and sent either to a row decoder to select a row address, or to a column decoder to select a column address. Data is transferred to or from the NAND Flash memory array either in a byte format (x8) or in a word format (x16) through a page buffer which is a combination of a data register and a cache register. During normal page operations, the data and cache registers act as a single register. During cache operations, the data and cache registers operate independently to increase data throughput. The NAND Flash memory array is programmed and read using page-based operations and is erased using block-based operations. The status register reports the status of all operations.

*Figure 2: Functional Block Diagram*



## Memory Array Architecture & Addressing

*Figure 3: Memory Array Architecture*



x8:
Page = 2,048 + 64 bytes = 2,112 bytes
1 block = 64 pages (135,168 bytes)
1Gbit Device = 1024 blocks (138,412,032 bytes)
x16:
Page = 1,024 + 32 words = 1,056 words
1 block = 64 pages (67,54 words)
1Gbit Device = 1024 blocks (69,206,016 words)

*Table 4: Array Addressing Sequence – x8, x16*

**x8:**

| Cycle | IO[7] | IO[6] | IO[5] | IO[4] | IO[3] | IO[2] | IO[1] | IO[0] |
|---|---|---|---|---|---|---|---|---|
| **First** | BYTA7 | BYTA6 | BYTA5 | BYTA4 | BYTA3 | BYTA2 | BYTA1 | BYTA0 |
| **Second** | Logic '0' | Logic '0' | Logic '0' | Logic '0' | BYTA11 | BYTA10 | BYTA9 | BYTA8 |
| **Third** | BA7 | BA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| **Fourth** | BA15 | BA14 | BA13 | BA12 | BA11 | BA10 | BA9 | BA8 |

**x16:**

| Cycle | IO[15:8] | IO[7] | IO[6] | IO[5] | IO[4] | IO[3] | IO[2] | IO[1] | IO[0] |
|---|---|---|---|---|---|---|---|---|---|
| **First** | Logic '0' | WRDA7 | WRDA6 | WRDA5 | WRDA4 | WRDA3 | WRDA2 | WRDA1 | WRDA0 |
| **Second** | Logic '0' | Logic '0' | Logic '0' | Logic '0' | Logic '0' | Logic '0' | WRDA10 | WRDA9 | WRDA8 |
| **Third** | Logic '0' | BA7 | BA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| **Fourth** | Logic '0' | BA15 | BA14 | BA13 | BA12 | BA11 | BA10 | BA9 | BA8 |

Notes:

1. BYTAx: Byte based Column address, WRDAx: Word based Column address, PAx: Page address, BAx: Block address
2. Block address concatenated with page address = actual page address
3. I/O[15:8] are not used during the addressing sequence and must be driven to Logic ‘0’ (Low)
4. 1st and 2nd address cycles form the column Address, whereas 3rd and 4th address cycles form the row address

## Status Register Definition

*Table 5: Status Register Definition*

| Status Register Bits | Name | Function | Default State | Description |
|---|---|---|---|---|
| **SR[7]** | WRPT | WP# based Device Protection | 1 | 0 = Device is protected - WP# is Low<br>1 = Device Is not protected - WP# is High |
| **SR[6]** | RDY | Data Cache Register Ready Status Bit | 1 | 0 = Data Cache is busy - not ready<br>1 = Data Cache is not busy - ready |
| **SR[5]** | ARDY | Memory Array Ready Status Bit | 1 | 0 = Memory Array is busy - not ready<br>1 = Memory Array is not busy - ready |
| **SR[4]** | RSVD | Reserved for Future Use | 0 | Reserved for Future Use |
| **SR[3]** | RSVD | Reserved for Future Use | 0 | Reserved for Future Use |
| **SR[2]** | RSVD | Reserved for Future Use | 0 | Reserved for Future Use |
| **SR[1]** | PS1 | Program Status of the Previous Command (Cache Program) | 0 | 0 = Program was successful<br>1 = Program was not successful |
| **SR[0]** | PES2 | Program and Erase Status of the Current Command | 0 | 0 = Program or Erase was successful<br>1 = Program or Erase was not successful |

Notes:

1. SR[6] - RDY: If set to ‘1’, the device is ready for another command and all other status bits are valid. If cleared to ‘0’, then the last command issued is not yet complete and all other status bits are not valid. When cache operations are in use, this bit indicates when the Cache buffer is ready to accept new data. R/B# follows RDY (SR[5] indicates if the last command was complete).
2. SR[5] – ARDY: If set to ‘1’, all array operations are complete. If cleared to zero, then there is a command being processed or an array operation in progress.
3. SR[1] – PS1: This bit is only valid for cache program operations and shows whether the previous operation was a success or a failure. This bit is not valid until after the second 15h command or the 10h command has been transferred in a Cache program sequence.
4. SR[0] – PES2: This bit is valid for program and erase operations and shows whether the operation was a success or a failure. During cache program operations, this bit is only valid when ARDY is set to ‘1’.

## Identification Definition (Address 00h)

*Table 6: Device Identification Definition (Address 00h)*

| Byte # | | Options | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Values |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Manufacturer's ID | MID | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | ADh |
| **1** | Device ID | 1Gbit, x8, 3.3V | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | F1h |
| | | 1Gbit, x16, 3.3V | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | C1h |
| **2** | Device Characteristics | Internal Device # 1 | | | | | | | 0 | 0 | x8 /x16<br>↓<br>80h/80h |
| | | Internal Device # 2 | | | | | | | 0 | 1 | |
| | | Internal Device # 4 | | | | | | | 1 | 0 | |
| | | Internal Device # 8 | | | | | | | 1 | 1 | |
| | | Cell Type - 2 LEVEL | | | | | 0 | 0 | | | |
| | | Cell Type - 4 LEVEL | | | | | 0 | 1 | | | |
| | | Cell Type - 8 LEVEL | | | | | 1 | 0 | | | |
| | | Cell Type - 16 LEVEL | | | | | 1 | 1 | | | |
| | | Simultaneous Programmed Pages - 1 | | | 0 | 0 | | | | | |
| | | Simultaneous Programmed Pages - 2 | | | 0 | 1 | | | | | |
| | | Simultaneous Programmed Pages - 4 | | | 1 | 0 | | | | | |
| | | Simultaneous Programmed Pages - 8 | | | 1 | 1 | | | | | |
| | | Interleaved Programming - Not Supported | | 0 | | | | | | | |
| | | Interleaved Programming - Supported | | 1 | | | | | | | |
| | | Cache Program - Not Supported | 0 | | | | | | | | |
| | | Cache Program - Supported | 1 | | | | | | | | |
| **3** | Array Architecture | Page Size - 1KB | | | | | | | 0 | 0 | x8 /x16<br>↓<br>1Dh/5Dh |
| | | Page Size - 2KB | | | | | | | 0 | 1 | |
| | | Page Size - 4KB | | | | | | | 1 | 0 | |
| | | Page Size - 8KB | | | | | | | 1 | 1 | |
| | | Spare Area Size - 8 bytes / 512 bytes | | | | | | 0 | | | |
| | | Spare Area Size - 16 bytes / 512 bytes | | | | | | 1 | | | |
| | | Block Size - 64KB | | | 0 | 0 | | | | | |
| | | Block Size - 128KB | | | 0 | 1 | | | | | |
| | | Block Size - 256KB | | | 1 | 0 | | | | | |
| | | Block Size - 512KB | | | 1 | 1 | | | | | |
| | Interface Type | x8 | | 0 | | | | | | | |
| | | x16 | | 1 | | | | | | | |
| | Access Time | 45ns | 0 | | | | 0 | | | | |
| | | 25ns | 0 | | | | 1 | | | | |

| Byte # | | Options | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Values |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Reserved for Future Use | 1 | | | | 0 | | | | |
| | | Reserved for Future Use | 1 | | | | 1 | | | | |

## Identification Definition (Address 20h)

*Table 7: Device Identification Definition (Address 20h)*

| Byte # | Option | Values |
|---|---|---|
| **0** | "O" | 4Fh |
| **1** | "N" | 4Eh |
| **2** | "F" | 46h |
| **3** | "I" | 49h |

## Parameter Page Structure & Values

*Table 8: Parameter Page Structure & Values*

| Byte # | O/M | Description | Values |
|---|---|---|---|
| | | **Revision Information & Features Block** | |
| **0 - 3** | M | Signature "O" "N" "F" "I" | 4Fh, 4Eh, 46h, 49h |
| **4 - 5** | M | Revision Number | 02h, 00h |
| **6 - 7** | M | Features Supported<br>Bit # Value<br>0 1 = Supports 16-bit Data bus<br>1 1 = Supports multiple LUN operations<br>2 1 = Supports non-sequential page programming<br>3 1 = Supports interleaved operations<br>4 1 = Supports odd to even page Copyback<br>5 - 15 0 = Reserved | 14h, 00h |
| **8 - 9** | M | Optional Commands Supported<br>Bit # Value<br>0 1 = Supports Page Cache Program<br>1 1 = Supports Read Cache<br>2 1 = Supports Get/Set features<br>3 1 = Supports Read Status Enhanced<br>4 1 = Supports Copyback<br>5 1 = Supports Read Unique ID<br>6 - 15 0 = Reserved | 33h, 00h |
| **10- 31** | | Reserved (0) | 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h |
| | | **Manufacturer's Information Block** | |
| **32 - 43** | M | Manufacturer's ID (12 ASCII Characters) | 48h, 59h, 4Eh, 49h, 58h, 20h, 20h, 20h, 20h, 20h, 20h, 20h |

| Byte # | O/M | Description | Values |
|---|---|---|---|
| **44 - 63** | M | Device Model (20 ASCII Characters) | 48h, 32h, 37h, 55h, 31h, 47h, 38h, 46h, 32h, 43h, 4Bh, 41h, 2Dh, 42h, 4Dh, 20h, 20h, 20h, 20h, 20h |
| **64** | M | JEDEC Manufacturer's ID | ADh |
| **65 - 66** | O | Date Code | 00h, 00h |
| **67 - 79** | | Reserved (0) | 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h |
| | | **Memory Organization Block** | |
| **80 - 83** | M | Number of Data Bytes per Page (2048 Bytes) | 00h, 08h, 00h, 00h |
| **84 - 85** | M | Number of Spare Bytes per Page (64 Bytes) | 40h, 00h |
| **86 - 89** | M | Number of Data Bytes per Partial Page (512 Bytes) | 00h, 00h, 00h, 00h |
| **90 - 91** | M | Number of Spare Bytes per Partial Page (16 Bytes) | 00h, 00h |
| **92 - 95** | M | Number of Pages per Block (64 Pages) | 40h, 00h, 00h, 00h |
| **96 - 99** | M | Number of Blocks per Logical Unit (LUN) (1024) | 00h, 04h, 00h, 00h |
| **100** | M | Number of Logical Units (LUNs) | 01h |
| **101** | M | Number of address cycles<br>Bit # Value<br>0 - 3 Row Address Cycles<br>4 - 7 Column Address Cycles | 22h |
| **102** | M | Number of Bits per Cell | 01h |
| **103 - 104** | M | Bad Blocks Number per LUN (20) | 20h, 00h |
| **105 - 106** | M | Block Endurance (50K) | 05h, 04h |
| **107** | M | Guaranteed Valid Blocks at Beginning of Device | 01h |
| **108 - 109** | M | Block Endurance for Guaranteed Valid Blocks | 05h, 04h |
| **110** | M | Number of Programs per Page | 04h |
| **111** | M | Partial Programming Attributes<br>Bit # Value<br>0 1 = Partial Page programming has Constraints<br>1 - 3 1 = Reserved (0)<br>4 1 = Partial Page Layout<br>5 - 7 0 = Reserved | 00h |
| **112** | M | Number of Bits ECC Correctability | 04h |
| **113** | M | Number of Interleaved Address Bits<br>Bit # Value<br>0 - 3 Number of Interleaved Address Bits<br>4 - 7 Reserved (0) | 00h |
| **114** | M | Interleaved Operation Attributes<br>Bit # Value<br>0 Overlapped / concurrent Interleaving Support<br>1 1 = No Block Address Restrictions<br>2 1 = Program Cache Supported<br>3 Address Restrictions for Program Cache<br>4 - 7 0 = Reserved | 00h |

| Byte # | O/M | Description | Values |
|---|---|---|---|
| **115 - 127** | | Reserved (0) | 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h |
| **Electrical Parameter Block** | | | |
| **128** | M | I/O Pin Capacitance (10pF) | 0Ah |
| **129 - 130** | M | Timing Mode Support<br>Bit # Value<br>0 1 = Supports Timing mode 0<br>1 1 = Supports Timing mode 1<br>2 1 = Supports Timing mode 2<br>3 1 = Supports Timing mode 3<br>4 1 = Supports Timing mode 4<br>5 1 = Supports Timing mode 5<br>6 - 15 0 = Reserved | 1Fh, 00h |
| **131 - 132** | M | Program Cache Timing Mode Support<br>Bit # Value<br>0 1 = Supports Timing mode 0<br>1 1 = Supports Timing mode 1<br>2 1 = Supports Timing mode 2<br>3 1 = Supports Timing mode 3<br>4 1 = Supports Timing mode 4<br>5 1 = Supports Timing mode 5<br>6 - 15 0 = Reserved | 1Fh, 00h |
| **133 - 134** | M | $t_{PROG}$ Maximum Page Program Time (700 µs) | BCh, 02h |
| **135 - 136** | M | $t_{BERS}$ Maximum Block Erase Time (10000 µs) | 10h, 27h |
| **137 - 138** | M | $t_R$ Maximum Page Read Time (25 µs) | 19h, 00h |
| **139 - 163** | | Reserved (0) | 3Ch, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h |
| **Vendor Block** | | | |
| **164 - 165** | M | Vendor Specific Revision Number | 00h, 00h |

| Byte # | O/M | Description | Values |
|---|---|---|---|
| **166 - 253** | | Vendor Specific | 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h |
| **254 - 255** | M | Integrity CRC | 82h, BCh |
| | | **Redundant Parameter Pages** | |
| **256 - 511** | M | Value of Bytes 0 - 255 | Not Available |
| **512 - 767** | M | Value of Bytes 0 - 255 | Not Available |
| **768 +** | O | Additional Redundant Parameter Pages | Not Available |

# Bus Interface

As mentioned above, the I/O bus on the device is multiplexed. Commands, addresses and data input/output all share the same I/O pins. Commands and addresses are always supplied on I/O[7:0]. Data uses I/O [7:0] for x8 configuration and uses I/O[15:0] for x16 configuration.

The read, program or erase command sequences typically consist of a command input cycle, two or four address input cycles, and one or more data cycles, either input or output.

*Table 9: Device Modes Selection*

| Mode | CE# | CLE | ALE | WE# | RE# | WP# | I/O[x] |
|---|---|---|---|---|---|---|---|
| **Standby** | $V_{IH}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{CC}$ / $V_{SS}$ | $V_{IH}$ / $V_{IL}$ |
| **Command Input** | $V_{IL}$ | $V_{IH}$ | $V_{IL}$ | ↑ | $V_{IH}$ | $V_{IH}$ | Data |
| **Address Input** | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | ↑ | $V_{IH}$ | $V_{IH}$ | Data |
| **Data Input** | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | ↑ | $V_{IH}$ | $V_{IH}$ | Data |
| **Data Output** | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | ↓ | $V_{IH}$ / $V_{IL}$ | Data |
| **Write Protect** | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IL}$ | $V_{IH}$ / $V_{IL}$ |

Notes:

1. $V_{IH}$ = High (Logic ‘1’), $V_{IL =}$ Low (Logic ‘0’)

## Standby

The device enters standby when CE# pin is driven $V_{IH}$ (High). This helps reduce power consumption. In standby mode, all I/Os are tri-stated (High-Z).

**Note:** The device enters standby if CE# goes High and the device is not busy (no program/erase operations in progress).

## Busy

The device enters busy when program, erase or read operations are initiated. The device returns to Standby after the completion of the operation. During busy state, only RESET (FFH) and READ STATUS REGISTER (70h) commands are accepted by the device.

*Table 10: Recommended Signal Selections During Busy*

| Mode | CE# | CLE | ALE | WE# | RE# | WP# | I/O[x] |
|---|---|---|---|---|---|---|---|
| **Busy Period (READ)** | $V_{IL}$ | $V_{IL}$ | $V_{IL}$ | $V_{IH}$ | $V_{IH}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ |
| **Busy Period (Program)** | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ | $V_{IH}$ / $V_{IL}$ |
| **Busy Period (Erase)** | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ / $V_{IL}$ | $V_{IH}$ | $V_{IH}$ / $V_{IL}$ |

Notes:

1. $V_{IH}$ = High (Logic ‘1’), $V_{IL =}$ Low (Logic ‘0’)

## Device Protection (Write Protect WP#)

The write protect# (WP#) signal enables or disables program and erase operations within the Flash device. When WP# is Low, program and erase operations are disabled. When WP# is High, program and erase operations are enabled.

It is recommended that the host drive WP# Low during power-on until VCC is stable to prevent inadvertent program and erase operations.

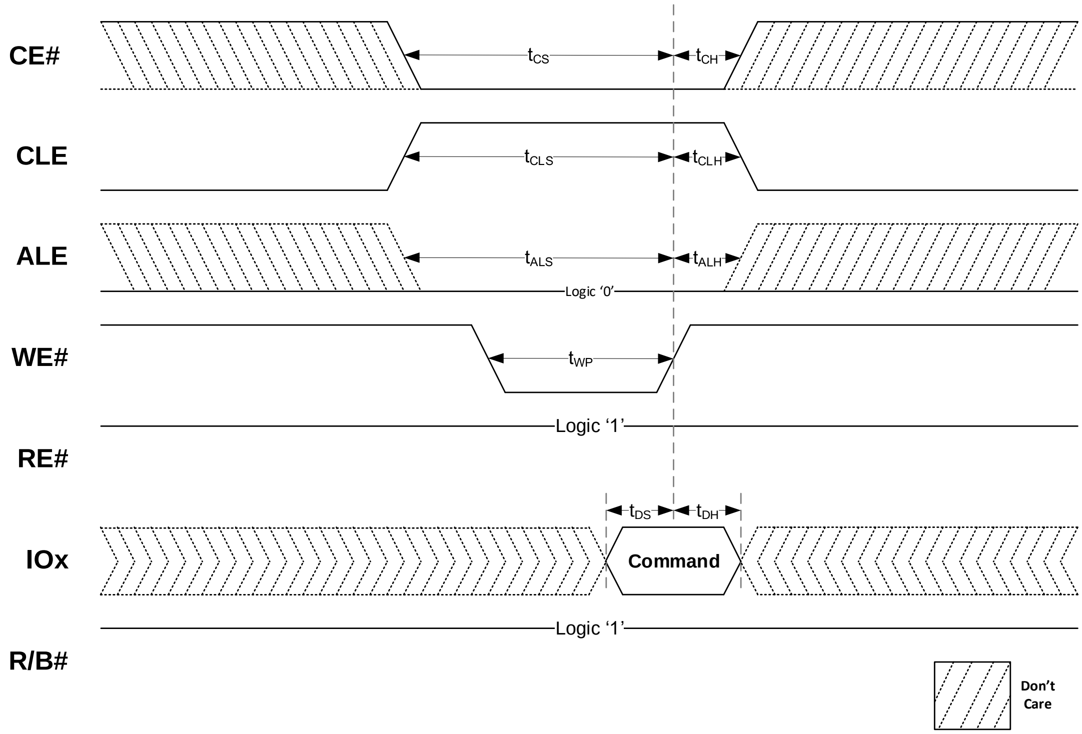
WP# must be transitioned only when the device is in Standby and prior to beginning a command sequence. After a command sequence is complete and the device is ready, WP# can be transitioned. After WP# is transitioned, the host must wait $t_{WW}$ before issuing a new command.

The WP# signal is always an active input, even when CE# is High. This signal should not be multiplexed with other signals.

## Command Input

A command is entered from I/O[7:0] to the command register on the rising edge of WE# when CE# is Low, ALE is Low, CLE is High, and RE# is High. Most commands are ignored if the device is busy (R/B# = 0); however, some commands, including READ STATUS (70h), are accepted. Moreover, for commands that starts a modify operation (program/erase), WP# must be high. For devices with a x16 interface, I/O[15:8] must be written with zeros when a command is issued.

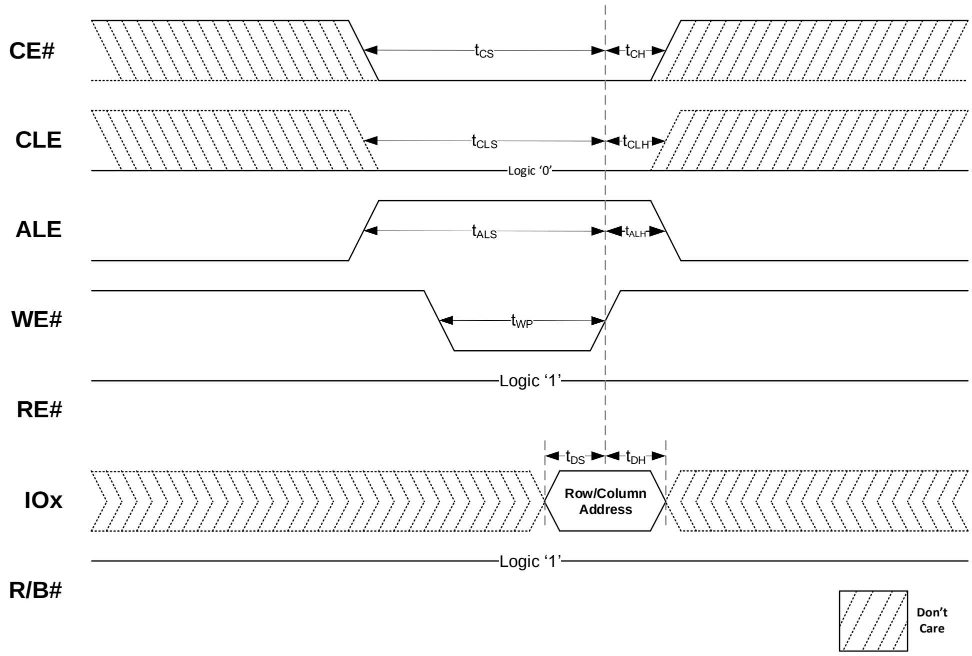
*Figure 4: Command Latch Cycle*



## Address Input

An address is entered to the address register on the rising edge of WE# when CE# is Low, ALE is High, CLE is Low, and RE# is High. Bits that are not part of the address space must be Low (see Array Addressing Sequences). The number of address cycles required depends on the command (refer to the command descriptions to determine addressing requirements). Addresses are input on I/O[7:0] on x8 devices and on I/O[15:0] on x16 devices. Moreover, for commands that start a modify operation (program/erase), WP# must be high.

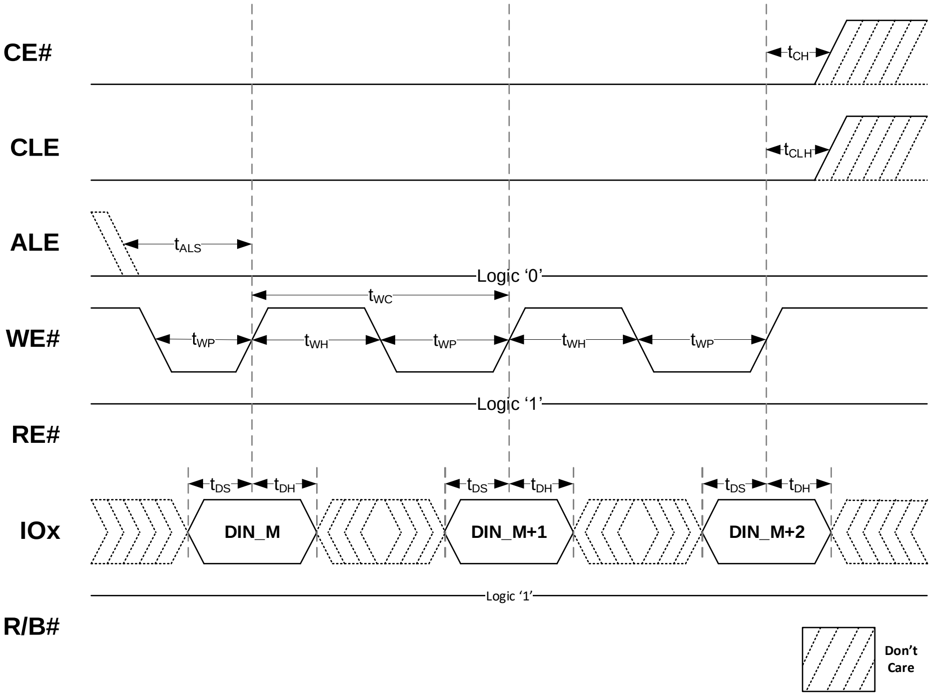
*Figure 5: Address latch Cycle*



## Data Input

Data insertion is serial and timed by WE# cycles. Data is entered to the page buffer (cache register or data register) on the rising edge of WE# when CE# is Low, ALE is Low, CLE is Low, and RE# is High. Data input is ignored if the device is busy (R/B# = 0). Data is input on I/O[7:0] on x8 devices and on I/O[15:0] on x16 devices.

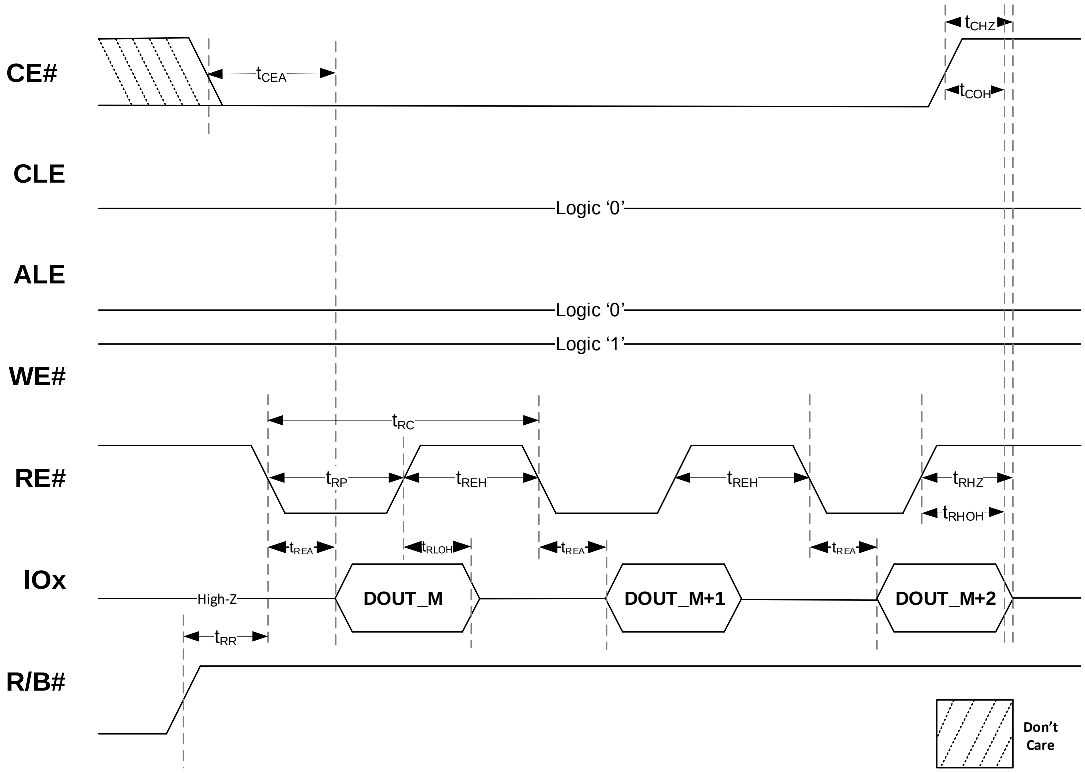
*Figure 6: Data Input Cycle*



## Data Output

Data can only be output if the device is not busy and is in the READ state. Data output is supported following a READ operation from the NAND Flash array. Data is output from the cache register on the falling edge of RE# when CE# is Low, ALE is Low, CLE is Low, and WE# is High. Data is output on I/O[7:0] on x8 devices and on I/O[15:0] on x16 devices.

*Figure 7: Data Output Cycle*



## Command Set

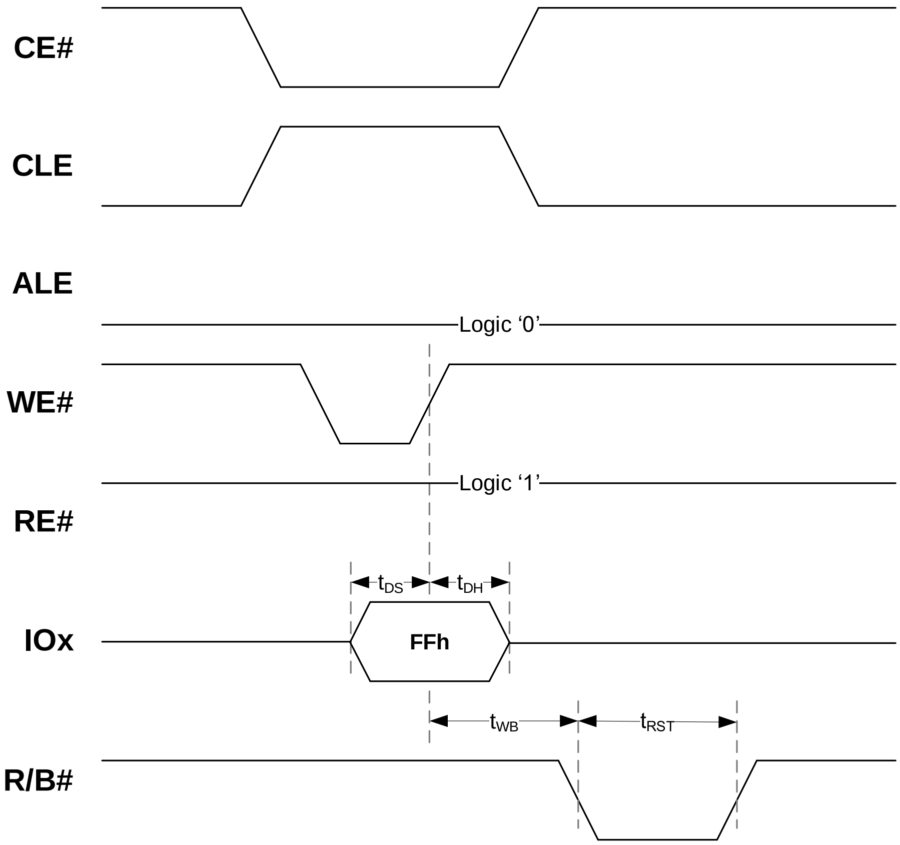
*Table 11: Command Set*

| Operation | Command Cycle #1 | Address Cycles | Data Input Cycles | Command Cycle # 2 | Command Accepted Device Busy |
|---|---|---|---|---|---|
| **Reset Operation** | | | | | |
| **RESET** | FFh | 0 | 0 | N/A | Yes |
| **Identification Operations** | | | | | |
| **READ ID** | 90h | 1 | 0 | N/A | No |
| **READ UNIQUE ID** | EDh | 1 | 0 | N/A | No |
| **READ PARAMETER PAGE** | ECh | 1 | 0 | N/A | No |
| **Status Register Operation** | | | | | |
| **READ STATUS REGISTER** | 70h | 0 | 0 | N/A | Yes |
| **Read Operations** | | | | | |
| **READ MODE** | 00h | 0 | 0 | N/A | No |
| **READ PAGE** | 00h | 4 | 0 | 30h | No |
| **READ PAGE CACHE (Start)** | 31h | 0 | 0 | N/A | No |
| **READ PAGE CACHE (End)** | 3Fh | 0 | 0 | N/A | No |
| **READ PAGE CACHE (Random)** | 00h | 4 | 0 | 31h | No |
| **READ RANDOM DATA** | 05h | 2 | 0 | E0h | No |
| **Program Operations** | | | | | |
| **PROGRAM PAGE** | 80h | 4 | Yes | 10h | No |
| **PROGRAM PAGE CACHE (End)** | 80h | 4 | Yes | 10h | No |
| **PROGRAM PAGE CACHE (Start)** | 80h | 4 | Yes | 15h | No |
| **RANDOM DATA INPUT** | 85h | 2 | Yes | N/A | No |
| **Re-Program Operations** | | | | | |
| **PROGRAM PAGE 2 (RE-PROGRAM)** | 8Bh | 5 | Yes | 10h | No |
| **Erase Operation** | | | | | |
| **ERASE BLOCK** | 60h | 2 | 0 | D0h | No |
| **Data Move Operations (Internal)** | | | | | |
| **READ FOR DATA MOVE** | 00h | 4 | 0 | 35h | No |
| **PROGRAM FOR DATA MOVE** | 85h | 4 | Optional | 10h | No |
| **One-Time Programmable (OTP) Region Operations** | | | | | |
| **OTP REGION ENTRY** | 29h-17h-04h-19h | 0 | 0 | N/A | No |

## RESET Operation (FFh)

The RESET command (FFh) places the NAND Flash device into the standby mode and aborts any command sequence in progress. Read, program, and erase commands can be aborted while the device is in the busy state using the RESET command. The contents of the memory location being programmed, or the block being erased are no longer valid - the data may be partially erased or programmed and is invalid. The command register is cleared and is ready for the next command. The data register and cache register contents are marked invalid. The status register contains the value E0h when WP# is High; otherwise it is written with a 60h value. If the device is already in RESET state a new reset command will not be accepted by the command register. R/B# goes Low for $t_{RST}$ during which the device completes the reset operation.

*Figure 8: RESET Operation*



## READ ID Operation (90h)

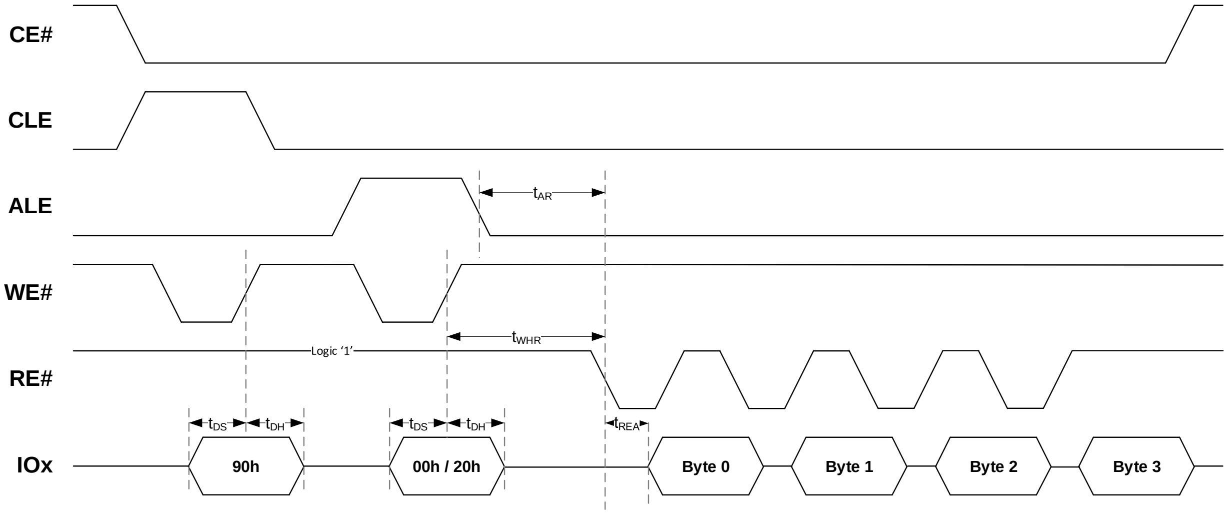
The READ ID (90h) command is used to read the product identification information programmed into the NAND Flash device. This command is accepted when the device is in Standby mode. Writing 90h to the command register puts the device in read ID mode. The device stays in this mode until another valid command is issued.

When the 90h command is followed by a 00h address cycle, the device returns a 4-byte identifier code that includes the manufacturer ID, device configuration, and part-specific information. Reading beyond the four bytes yields indeterminate data.

When the 90h command is followed by a 20h address cycle, the device returns the 4-byte ONFI identifier code. Reading beyond the four bytes yields indeterminate data.

Note: Table 6 provides the ID values.

*Figure 9: READ ID Operation*



## READ UNIQUE ID Operation (EDh)

The READ UNIQUE ID (EDh) command is used to read the ONFI Unique identification information programmed into the NAND Flash device. This command is accepted when the device is in Standby mode. Writing EDh to the command register puts the device in read UNIQUE ID mode. The device stays in this mode until another valid command is issued.

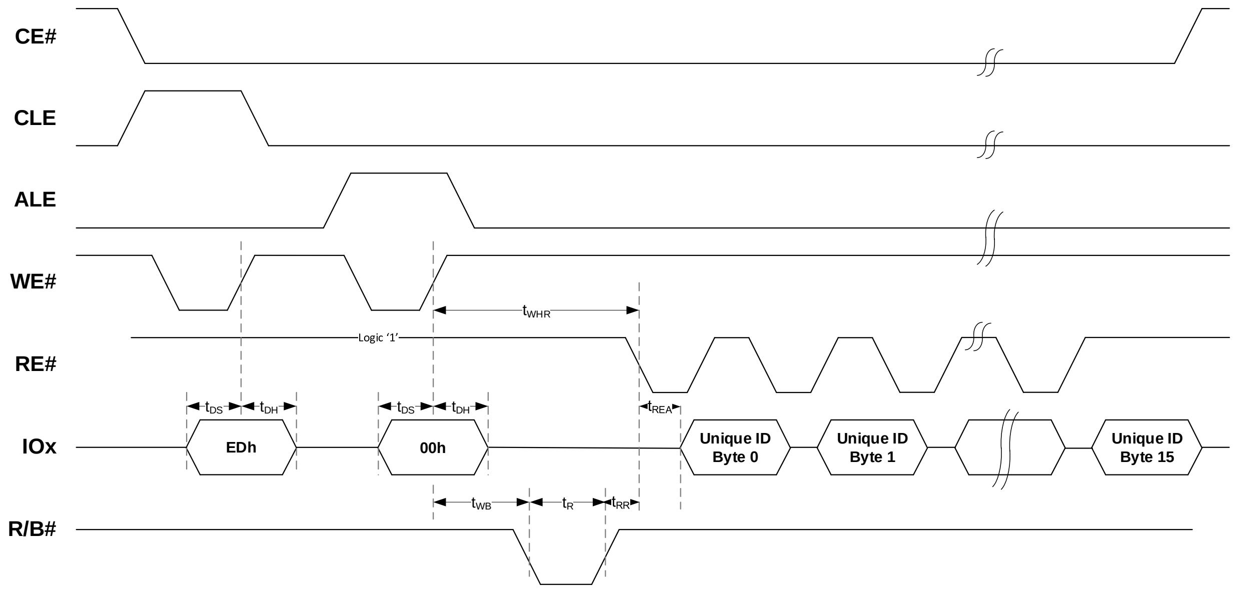
When the EDh command is followed by a 00h address cycle, the device returns 16-bytes of a unique value. The next 16-bytes are the bit-wise complement of the unique value. The host can verify that the Unique ID was read correctly by performing an XOR of the two values; result should be all ones. The host must monitor the R/B# pin or wait for the maximum data transfer time ($t_R$) before reading the Unique ID data.

*Table 12: Unique ID Data Description*

| Bytes | Information |
|---|---|
| 0-15 | Unique ID |
| 16-31 | Unique ID Complement |
| 32-47 | Unique ID |
| 48-63 | Unique ID Complement |
| 64-79 | Unique ID |
| 80-95 | Unique ID Complement |
| 96-111 | Unique ID |
| 112-127 | Unique ID Complement |
| 128-143 | Unique ID |
| 144-159 | Unique ID Complement |
| 160-175 | Unique ID |
| 176-191 | Unique ID Complement |
| 192-207 | Unique ID |
| 208-223 | Unique ID Complement |
| 224-239 | Unique ID |
| 240-255 | Unique ID Complement |

| Bytes | Information |
|---|---|
| **256-271** | Unique ID |
| **272-287** | Unique ID Complement |
| **288-303** | Unique ID |
| **304-319** | Unique ID Complement |
| **320-335** | Unique ID |
| **336-351** | Unique ID Complement |
| **352-367** | Unique ID |
| **368-383** | Unique ID Complement |
| **384-399** | Unique ID |
| **400-415** | Unique ID Complement |
| **416-431** | Unique ID |
| **432-447** | Unique ID Complement |
| **448-463** | Unique ID |
| **464-479** | Unique ID Complement |
| **480-495** | Unique ID |
| **496-511** | Unique ID Complement |

*Figure 10: READ Unique ID Operation*

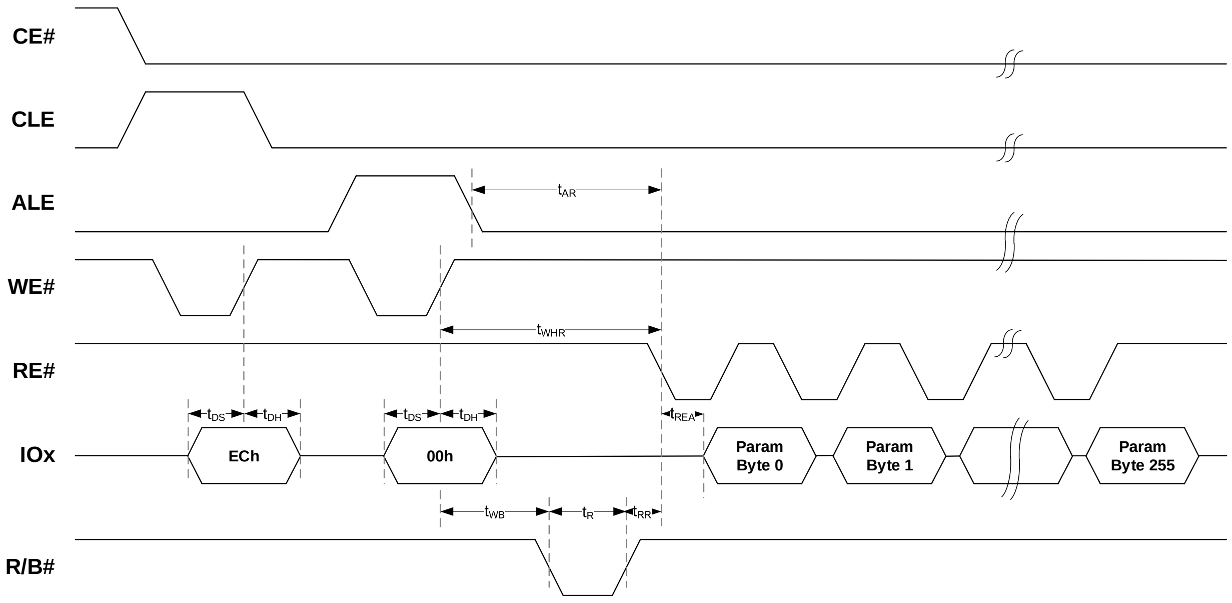


## READ PARAMETER PAGE Operation (ECh)

The READ PARAMETER PAGE (ECh) command is used to read the ONFI parameter page which describes the device's organization, features, timings and other behavioral parameters. These values are static and cannot be altered by the user. The READ PARAMETER PAGE command is accepted by the device when it is in Standby mode.

Writing ECh to the command register puts the device in read parameter page mode. It stays in this mode until another valid command is issued. When the ECh command is followed by an 00h address cycle, the device goes busy for $t_R$. If desired, the RANDOM DATA OUTPUT (05h-E0h) command can be used to change the location of data output. If the READ STATUS (70h) command is used to monitor for command completion, the READ MODE (00h) command must be used to re-enable data output mode.

*Figure 11: Parameter Read Operation*

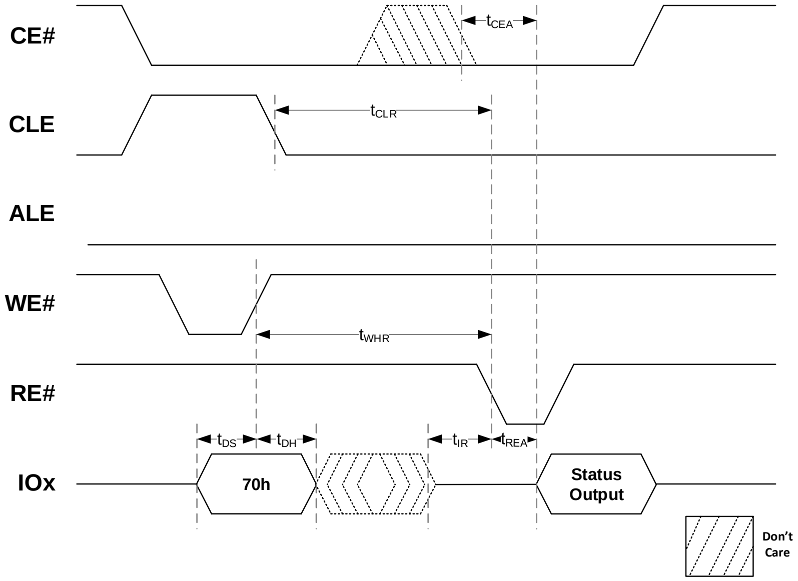


## READ STATUS REGISTER Operation (70h)

The device contains a Status Register which may be read to find out whether read, program or erase operation is completed, and whether the program or erase operation is completed successfully. The device provides its status through its 8-bit status register. After the READ STATUS REGISTER (70h) command is issued, status register output is enabled. The contents of the status register are returned on I/O[7:0] for each data output request (toggle RE#).

When the status register output is enabled, changes in the status register are seen on I/O[7:0] as long as CE# and RE# are Low; it is not necessary to toggle RE# to see the status register update. The command register remains in Status Read mode until other commands are issued to it. Therefore, if the status register is read during a random read cycle, the read command (00h) should be given before starting read cycles.

*Figure 12: Read Status Register Operation*



## READ MODE Operation (00h)

Upon initial device power up, the device defaults to Read mode. This operation is also initiated by writing 00h and 30h to the command register along with four address cycles. Two types of read operations are available: random read and serial page read. The random read mode is enabled when the page address is changed. The repetitive high to low transitions of the RE# clock make the device output the data starting from the selected column address up to the last column address. The device may output random data in a page instead of the consecutive sequential data by writing random data output command. Random data output can be operated multiple times regardless of how many times it is done in a page. Any operation other than read or random data output causes the device to exit read mode.

The READ MODE (00h) command enables data output and disables status output after a READ operation (00h-30h, 00-31h) has been modified with a status operation (70h). This command is accepted by the device when it is ready (RDY = 1, ARDY = 1). It is also accepted by the device during READ PAGE CACHE (31h, 00h-31h) operations (RDY = 1 and ARDY = 0).

## READ PAGE Operation (00h – 30h)

The READ PAGE (00h–30h) command copies a page (x8: 2112 bytes, x16: 1056 words) from the NAND Flash array to its cache register and enables data output. This command is accepted by the device when it is ready (RDY = 1, ARDY = 1).

To read a page from the NAND Flash array, the host must first write the 00h command to the command register, followed by writing 4 address cycles to the address registers, and conclude with writing the 30h command. The device will go busy (RDY = 0, ARDY = 0) for $t_R$ as data is transferred. To determine the progress of the data transfer, the host can monitor the device's R/B# signal or, alternatively, the READ STATUS REGISTER operation (70h) can be used. If the READ STATUS REGISTER operation is used to monitor the progress, the host must disable status output and enable data output by issuing the READ MODE (00h) command when the device gets ready (RDY = 1, ARDY = 1).

As mentioned above, two types of read operations are available: random read and serial page read. During data output the READ RANDOM DATA (05h-E0h) command can be issued if random data output from the page is desired. Otherwise, the data output is sequential.

*Figure 13: Read Page Operation*

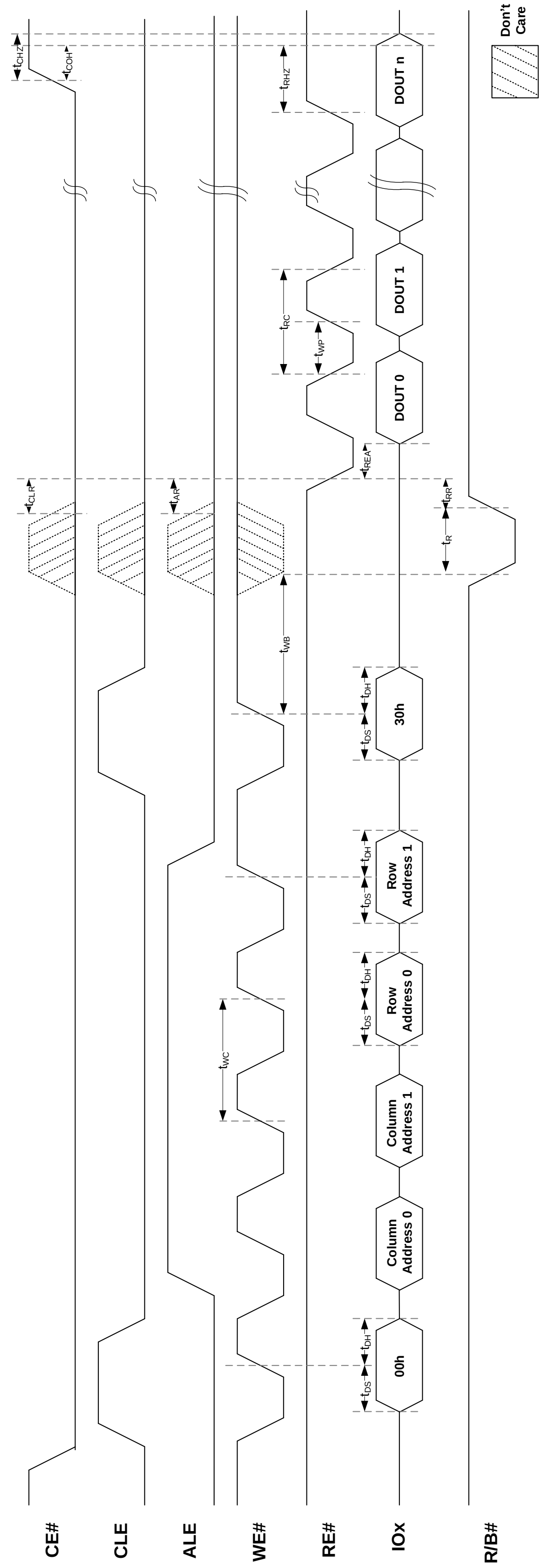
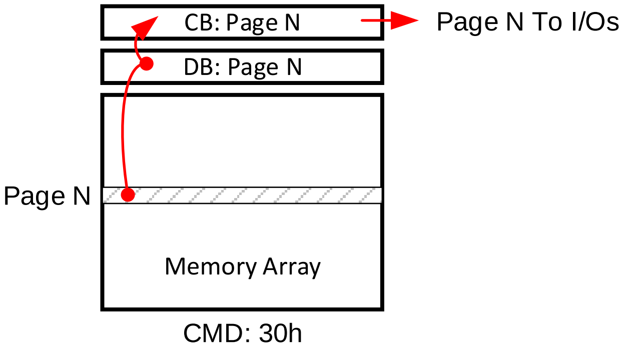


*Figure 14: Read Page Operation Overview*



A data transfer operation from the cell array to the Cache Buffer (CB) via Data Buffer (DB) starts on the rising edge of WE# in the 30h command input cycle (after the address information has been latched in). The device will be in the Busy state during this transfer period.

## READ PAGE CACHE Start Operation (31h)

The READ PAGE CACHE Start (31h) command allows reading a page from the cache register while another page is simultaneously loaded from the Flash array into the data register. A READ PAGE (00h–30h) command must be issued prior to the READ PAGE CACHE Start command. Operationally, the READ PAGE CACHE Start (31h) command reads the next sequential page within a block into the data register (while the previous page is output from the cache register). This command is accepted by the device when it is ready (RDY = 1, ARDY = 1). It is also accepted during READ PAGE CACHE (31h, 00h-31h) operation (RDY = 1 and ARDY = 0).

To issue this command, write 31h to the command register. After this command is issued, R/B# goes Low and the device is busy (RDY = 0, ARDY = 0) for $t_{RBSY}$. After $t_{RBSY}$, R/B# goes High (RDY = 1, ARDY = 0), indicating that the cache register is available and that the specified page is being copied from the NAND Flash array to the data register. At this point, data can be output from the cache register beginning at column address 00h. The READ RANDOM DATA (05h-E0h) command can be used to change the column address of the data being output from the cache register.

The READ PAGE CACHE Start (31h) command must not cross block boundaries. The host can enter the address of the next page to be read from the Flash array. If the host does not enter an address to retrieve, the next sequential page is read. When no more pages are to be read, the final page is copied into the page register by issuing the READ PAGE CACHE End (3Fh) command.

*Figure 15: Read Page Cache Start Operation*

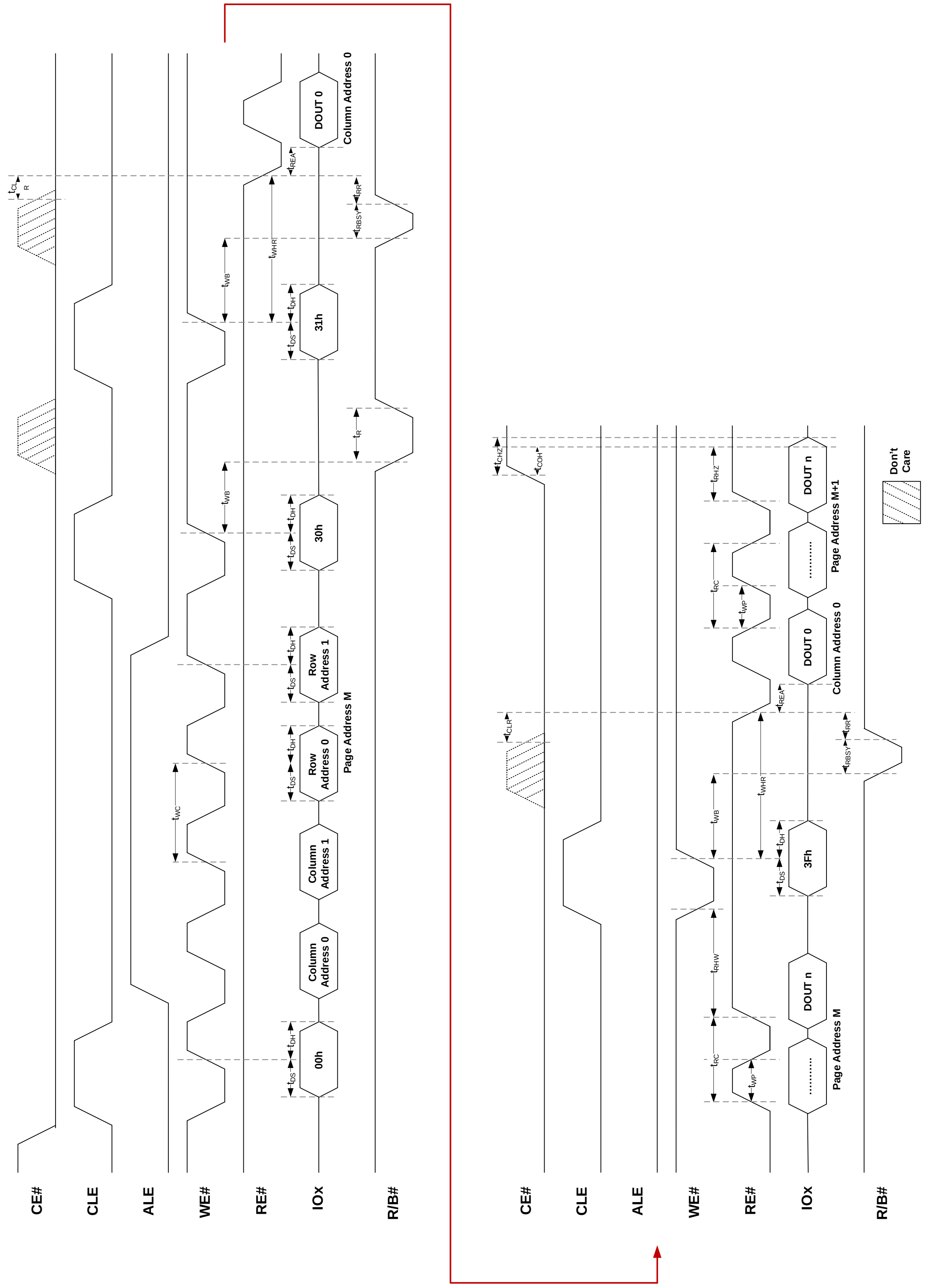
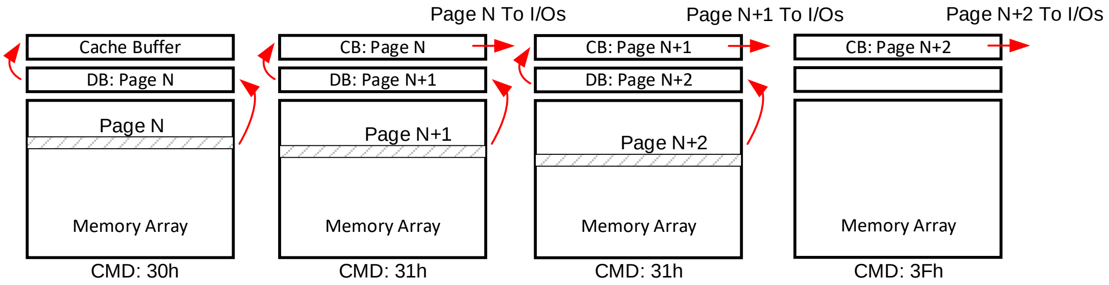


*Figure 16: Read Page Cache Start Operation Overview*



When 31h command is issued to the device, the data content of the next page is transferred to the Data Buffer (DB) during serial output of the Cache Buffer (CB).

- During Normal read, Data is transferred from Page N to CB through DB. During this time period, the device outputs Busy state for $t_R$.
- After the Ready/Busy returns to Ready, 31h command is issued and data is transferred to DB from CB again. This data transfer takes $t_{RBSY}$.
- Data of Page N+1 is transferred to DB while the data of Page N in CB can be read out.
- The 31h command makes data of Page N+1 transfer to CB from DB after the completion of the transfer from cell to DB. The device outputs Busy state for $t_{RBSY}$.
- Data of Page N+2 is transferred to DB while the data of Page N +1 in CB can be read out by RE# clock simultaneously.
- The 3Fh command makes the data of Page N+2 transfer to the CB from the DB after the completion of the transfer to DB. The device outputs Busy state for $t_{RBSY}$.
- Data of Page N+2 in CB can be read out.

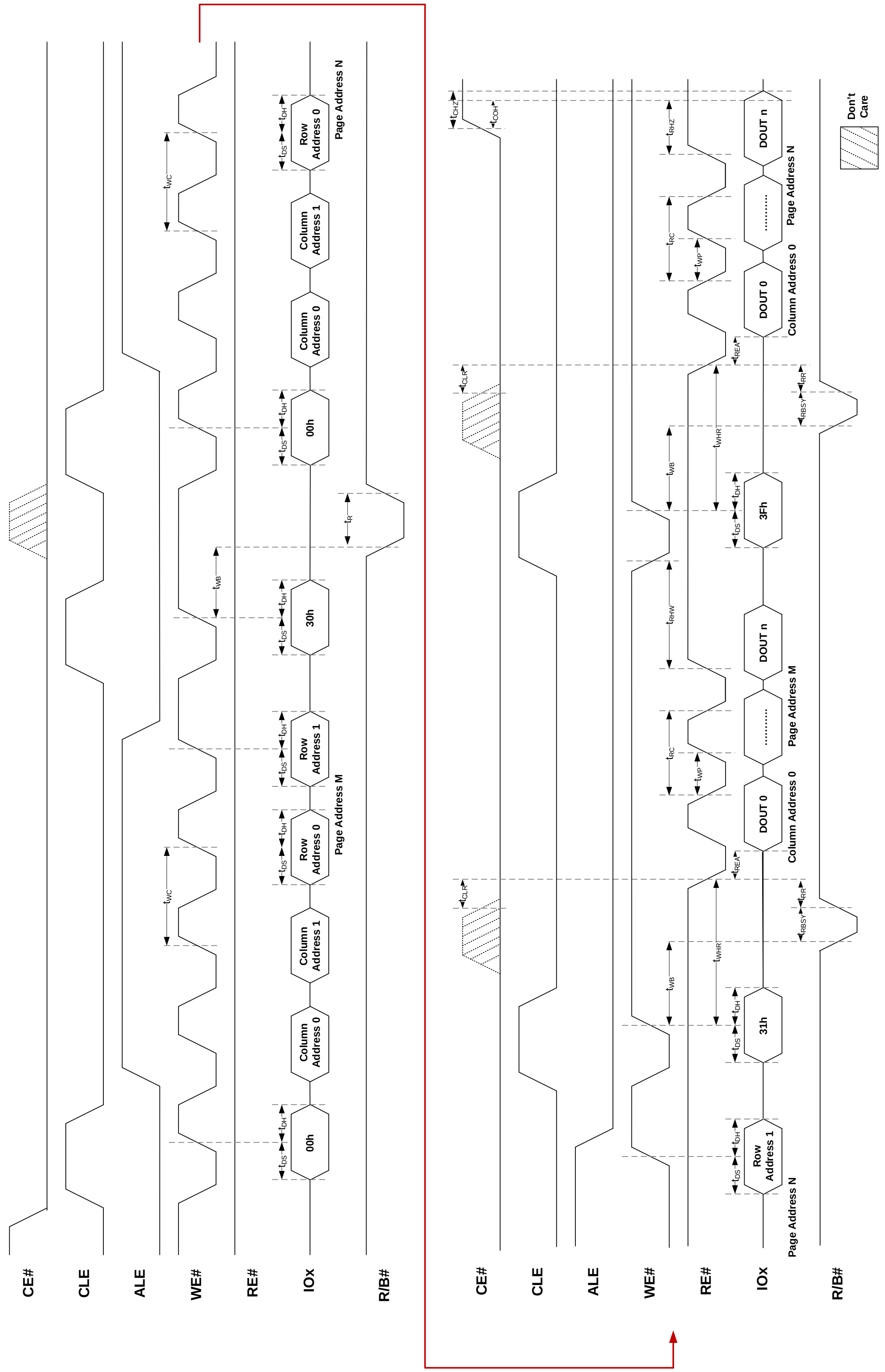
## READ PAGE CACHE Random Operation (00h - 31h)

The READ PAGE CACHE RANDOM (00h-31h) command allows reading a page from the cache register while another page is simultaneously loaded from the Flash array into the data register. However, the page address can be randomly selected (not sequential). A READ PAGE (00h – 30h) command must be issued prior to the READ PAGE CACHE RANDOM command. This command is accepted by the device when it is ready (RDY = 1, ARDY = 1). It is also accepted during READ PAGE CACHE (31h, 00h-31h) operation (RDY = 1 and ARDY = 0).

To issue this command, write 00h to the command register, followed by writing 4 address cycles to the address register and conclude by writing 31h to the command register. Note that the column address in the address specified is ignored. After this command is issued, R/B# goes Low and the device is busy (RDY = 0, ARDY = 0) for $t_{RBSY}$. After $t_{RBSY}$, R/B# goes High (RDY = 1, ARDY = 0) indicating that the cache register is available and that the specified page is being copied from the NAND Flash array to the data register.

The data in the cache register can be read beginning at column address 0. The READ RANDOM DATA (05h-E0h) command can be used to change the column address of the data being output from the cache register.

*Figure 17: Read Page Cache Random Operation*



## READ PAGE CACHE End Operation (3Fh)

The READ PAGE CACHE End (3Fh) command ends the read page cache sequence and copies a page from the data register to the cache register. This command is accepted by the device when it is ready (RDY = 1, ARDY = 1). It is also accepted by the die (LUN) during READ PAGE CACHE (31h, 00h-31h) operations (RDY = 1 and ARDY = 0).

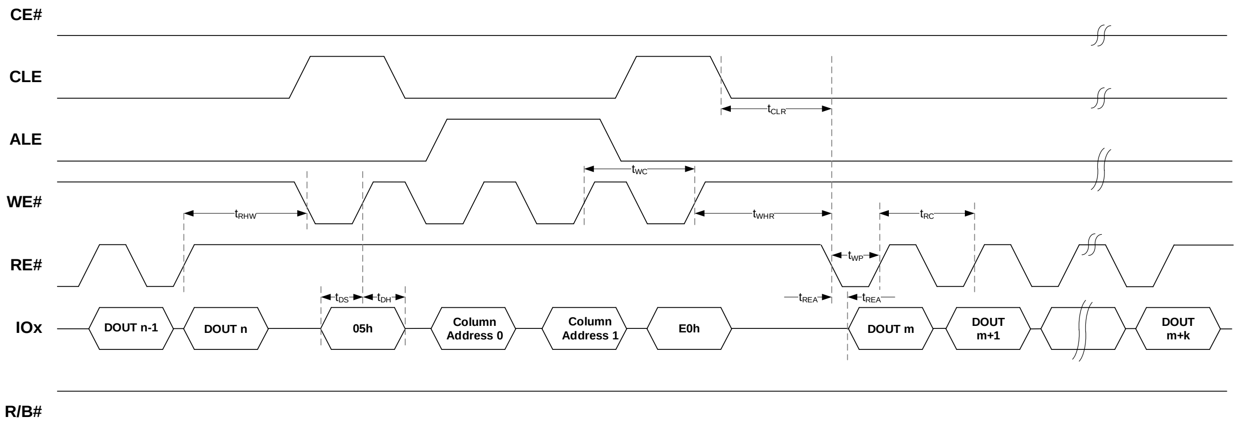
To issue the READ PAGE CACHE End (3Fh) command, write 3Fh to the command register. After this command is issued, R/B# goes LOW and device is busy (RDY = 0, ARDY = 0) for $t_{RBSY}$. After $t_{RBSY}$, R/B# goes HIGH and the device is ready (RDY = 1, ARDY = 1). At this point, data can be output from the cache register, beginning at column address 0. The READ RANDOM DATA (05h-E0h) command can be used to change the column address of the data being output from the cache register.

## READ RANDOM DATA Operation (05h – E0h)

The READ RANDOM DATA (05h-E0h) command changes the column address of the selected cache register and enables data output from the device. This command is accepted when the device is ready (RDY = 1; ARDY = 1) or during CACHE READ operations (RDY = 1; ARDY = 0).

Writing 05h to the command register, followed by two column address cycles containing the column address, followed by the E0h command, puts the device into data output mode. After the E0h command cycle is issued, the host must wait at least $t_{WHR}$ before requesting data output. The device stays in data output mode until another valid command is issued.

*Figure 18: Read Random Operation*



## PROGRAM PAGE Operation / PROGRAM PAGE CACHE End (80h – 10h)

As mentioned in the architecture section, programming is page based. Pages must be programmed sequentially within a block. However, the device also allows multiple partial page programming in a single page program cycle; Partial page programming of consecutive bytes (1 to 2112) or words (1 to 1056) in a single page program cycle. The number of consecutive partial page programming operation within the same page without an intervening erase operation must not exceed 4; for example, 2 times for main array (1time/512byte) and 2 times for spare array (1time/16byte).

The PROGRAM PAGE (80h-10h) command enables the host to input data to a cache register and moves the data from the cache register to the addressed block and page in the Flash array. This command is accepted by the device when it is ready (RDY = 1, ARDY = 1). It is also accepted when the device is busy with a PROGRAM PAGE CACHE (80h-15h) operation (RDY = 1, ARDY = 0). Note that (80h-10h) command is also used to end the program page cache operation.

To program an addressed page in the Flash array, write 80h to the command register. Write 4 address cycles containing the column address and row address. Data input cycles follow. Serial data is input beginning at the column address specified. At any time during the data input cycle the RANDOM DATA INPUT (85h) and PROGRAM FOR DATA MOVE (85h) commands can be issued. When data input is complete, write 10h to the command register. The device will go busy (RDY = 0, ARDY = 0) for $t_{PROG}$ as data is programmed.

The internal program state controller automatically executes the algorithms and timings necessary to program and verify, thereby freeing the system controller for other tasks. To determine the progress of the data transfer, the host can either monitor the device's R/B# signal or, execute the status operation (70h). When the device is ready (RDY = 1, ARDY = 1), the host should check the status of the Program/Erase status (PS1/PES2) bit. The internal program verify detects only errors for "1"s that are not successfully programmed to "0"s.

*Figure 19: Program Page Operation*

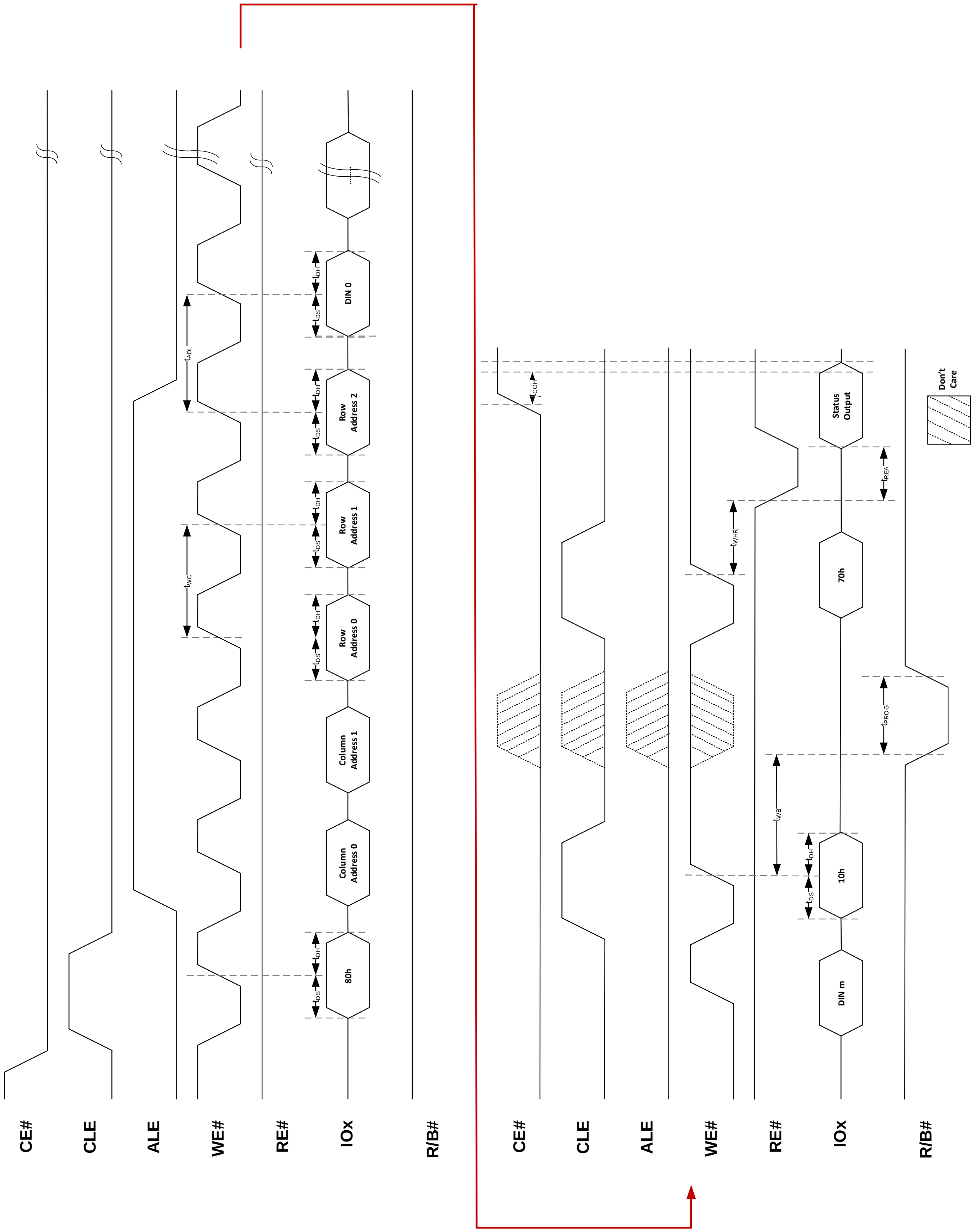
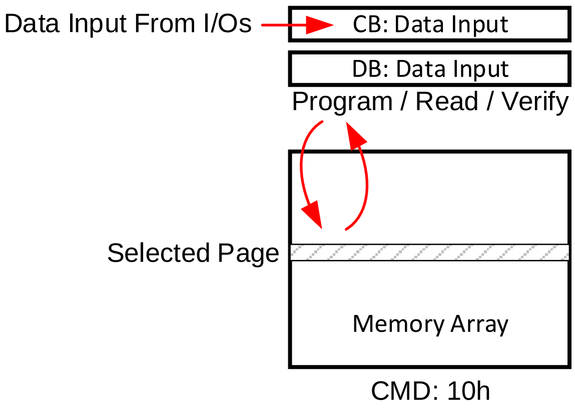


*Figure 20: Program Page Operation Overview*



The data is programmed from the Data Buffer (DB) to the selected page on the rising edge of WE# following input of the "10h" command. After programming, the programmed data is transferred back to the DB to be automatically verified by the device. If the programming does not succeed, the Program/ Verify operation is repeated by the device until success is achieved or until the maximum programming loop count is reached.

## PROGRAM PAGE CACHE Operation (80h – 15h)

PROGRAM PAGE CACHE is used to improve the program throughput by programing data using the cache register. Cache program is available only within a block. The PROGRAM PAGE CACHE (80h-15h) command is an extension of the PROGRAM PAGE (80h-10h) command.

The PROGRAM PAGE CACHE (80h-15h) command enables the host to input data to the cache register; copies the data from the cache register to the data register; then moves the data register contents to the memory array. Once the data is copied to the data register, the cache register is available for additional data using PROGRAM PAGE CACHE (80h-15h) or PROGRAM PAGE (80h-10h) commands. The PROGRAM PAGE CACHE command is accepted by the device when it is ready (RDY =1, ARDY = 1). It is also accepted by the device when it is busy with a PROGRAM PAGE CACHE operation (RDY = 1, ARDY = 0).

To input a page to the cache register, write 80h to the command register. Then write 4 address cycles containing the column address and row address. Data input cycles follow. Serial data is input beginning at the column address specified. At any time during the data input cycle the RANDOM DATA INPUT (85h) and PROGRAM FOR DATA MOVE (85h-10h) commands may be issued. When data input is complete, write 15h to the command register. The device will go busy (RDY = 0, ARDY = 0) for $t_{PBSY}$ to copy data from the cache register to the data register, and then to begin the programming operation.

To determine the progress of the data transfer, the host can either monitor the device's R/B# signal or, execute the READ STATUS REGISTER (70h) command. When the device is busy with a PROGRAM PAGE CACHE operation (RDY = 1, ARDY = 0), the host must check the status of the PS1 bit to ensure previous cache operation was successful. If, after $t_{PBSY}$, the host wants to wait for the PROGRAM PAGE CACHE operation to complete without issuing PROGRAM PAGE (80h-10h) command, the host should monitor ARDY bit in the status register until it is 1. The host should then check the status of the PS1 and PES2 bits for program completion status.

*Figure 21: Program Page Cache Operation*

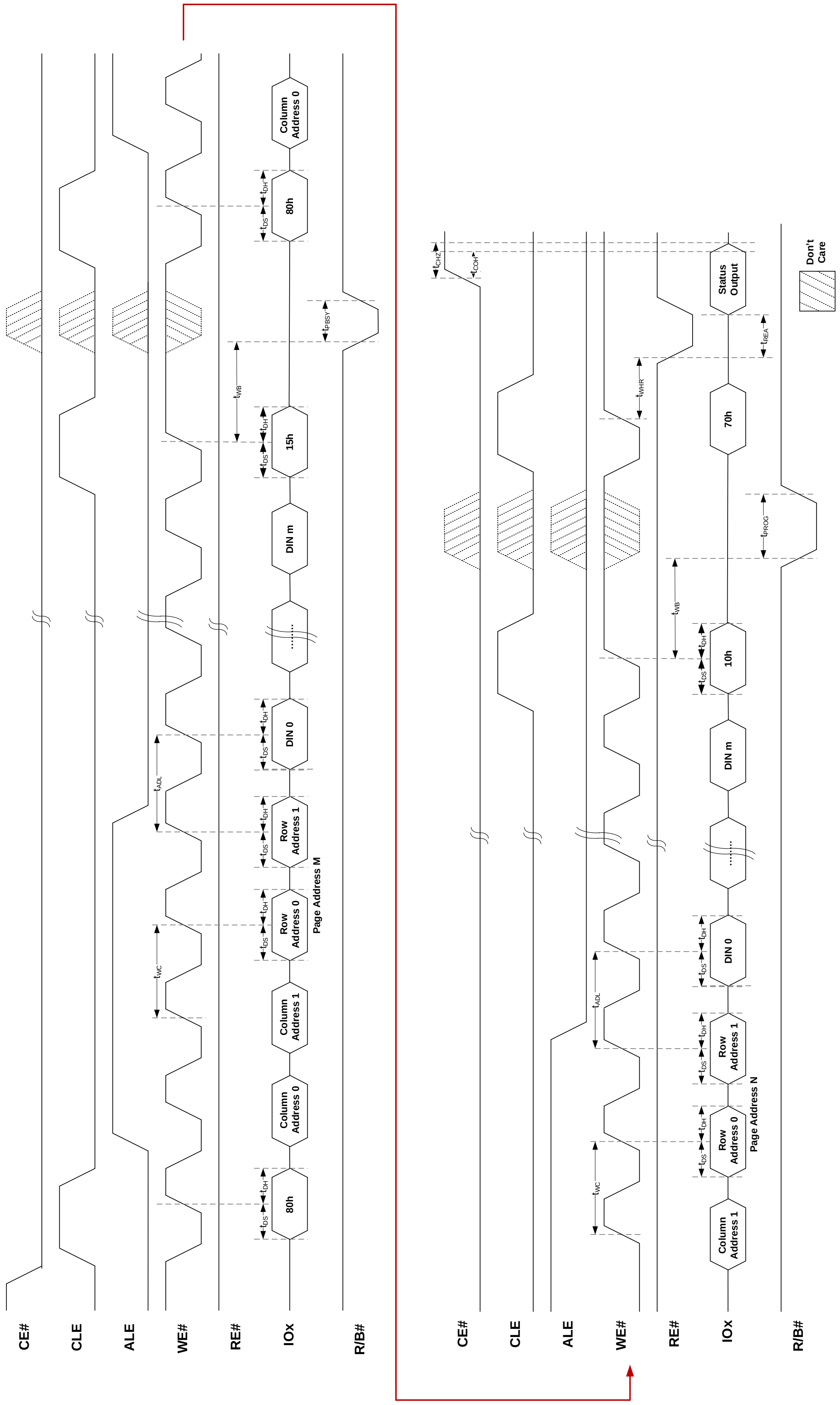
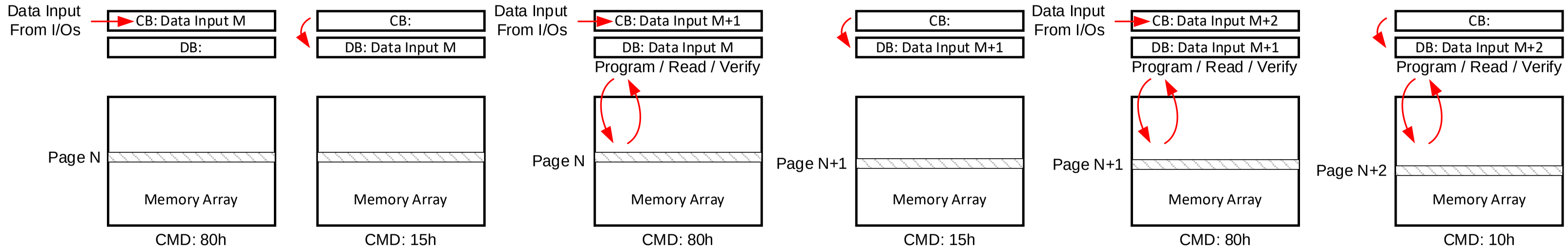


*Figure 22: Program Page Cache Operation Overview*



Issuing the 15h command to the device after serial data input initiates the program operation with Data Buffer (DB). Detailed Program Page Cache Operation steps are as follows:

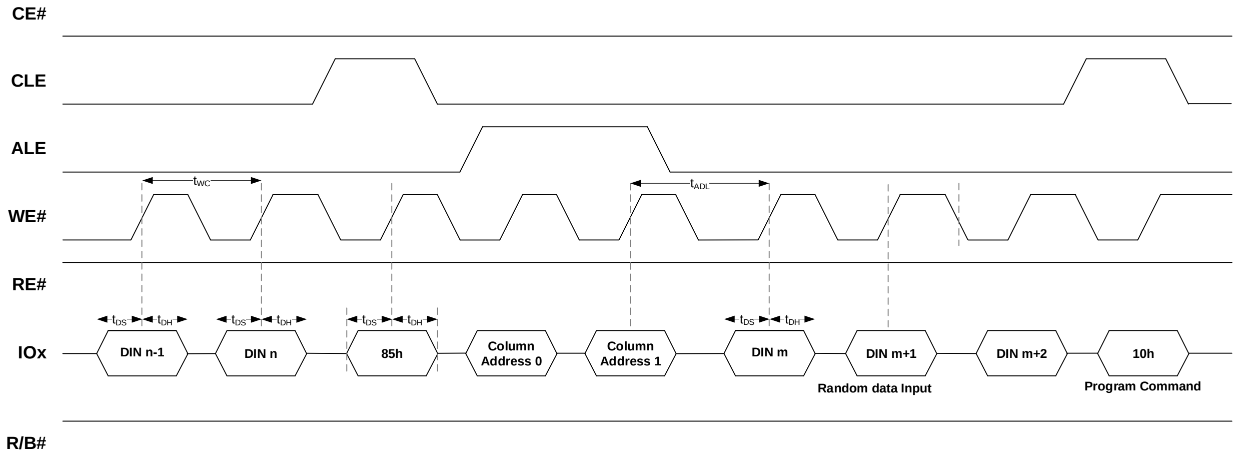
- Data for Page N is input to Cache Buffer (CB).
- Data is transferred to the DB by the 15h command. During the transfer the Ready/Busy outputs Busy State ($t_{PBSY}$).
- Data is programmed to the selected page while the data for page N+1 is input to the CB.
- Issuing the 15h command, the data in the CB is transferred to the DB after the programming of page N is completed. The device output busy state from the 15h command until $t_{PBSY}$.
- Data for Page N+2 is input to the CB while the data of the Page N+1 is being programmed.
- The programming with CB is terminated by the 10h command. When the device becomes Ready, it shows that the internal programming of the Page N+2 is completed. The device output busy state from the 10h command until $t_{PROG}$.

## RANDOM DATA INPUT Operation (85h)

The RANDOM DATA INPUT (85h) command changes the column address  and enables data input. This command is accepted by the device when it is ready (RDY = 1; ARDY = 1) or during PROGRAM PAGE CACHE operations (RDY = 1; ARDY = 0). The RANDOM DATA INPUT (85h) command is allowed after the required address cycles are specified, but prior to the final program command cycle (10h,15h) of the following commands while data input is permitted: PROGRAM PAGE (80h-10h), PROGRAM PAGE CACHE (80h-15h),and PROGRAM FOR DATA MOVE (85h-10h).

Writing 85h to the command register, followed by two column address cycles containing the column address, puts the device into data input mode. After the second address cycle is issued, the host must wait at least $t_{ADL}$ before inputting data. The device stays in data input mode until another valid command is issued.

*Figure 23: Random data Input Operation*



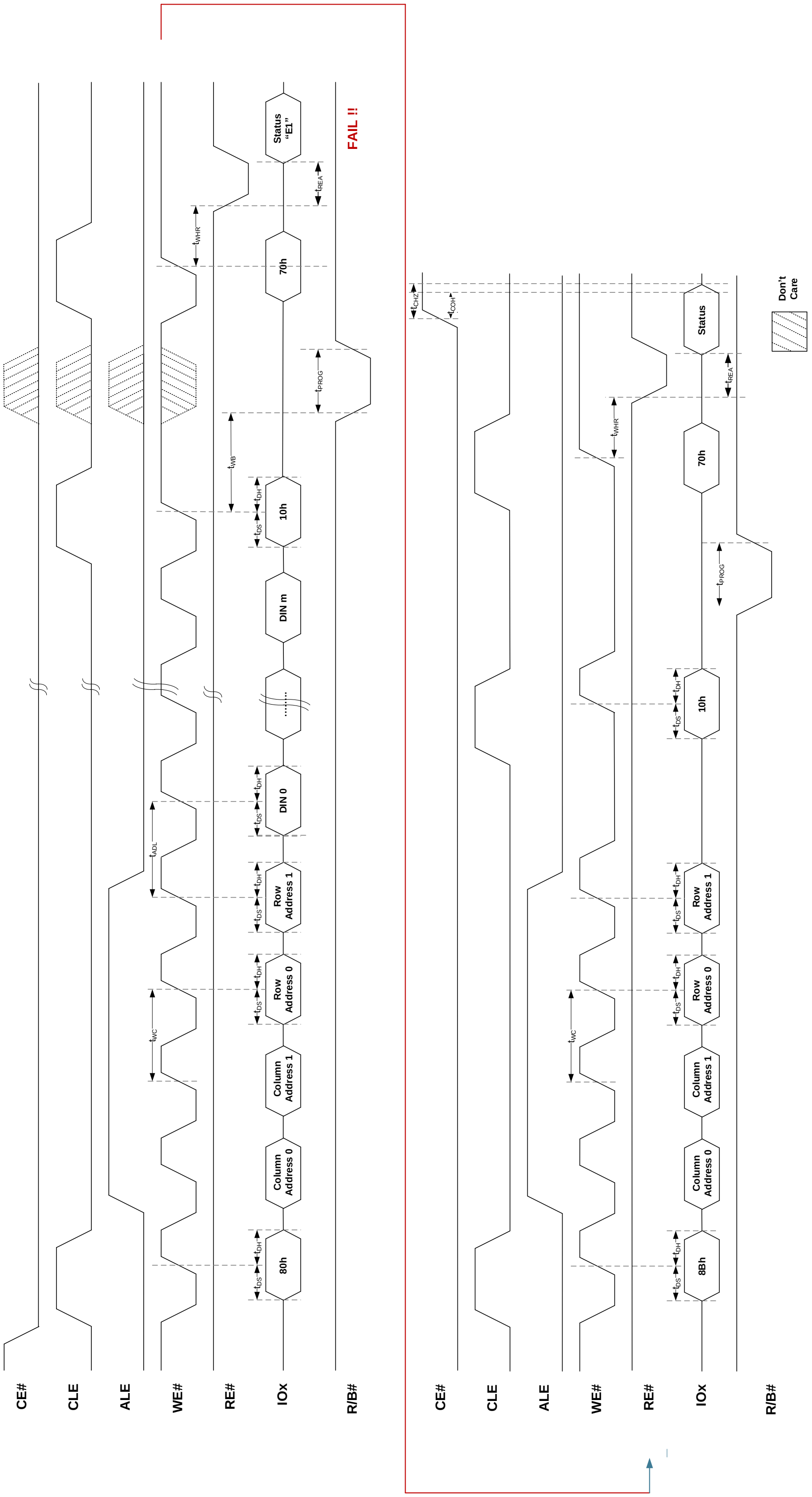
## PROGRAM PAGE 2 (RE-PROGRAM) Operation (8Bh – 10h)

The PROGRAM PAGE 2 (8Bh-10h) command allows re-programming of the same data into a new page if the last PAGE PROGRAM (80h-10h) operation failed. This command is most efficient if the data to be programmed is not changed. However, if the data needs to be altered, data in cycles can be initiated before issuing the program confirm “10h” command.

To re-program the loaded data into a new page address in the Flash array, write 8Bh to the command register. Write 5 address cycles containing the column address and row address of the new page. Data input cycles follow. Serial data is input beginning at the column address specified. At any time during the data input cycle the RANDOM DATA INPUT command can be issued. When data entry is complete, write 10h to the command register. The device will go busy (RDY = 0, ARDY = 0) for $t_{PROG}$ as data is programmed.

To determine the progress of the data transfer, the host can either monitor the device's R/B# signal or, execute the status operation (70h). When the device is ready (RDY = 1, ARDY = 1), the host should check the status of the Program/Erase status (PS1/PES2) bits.

*Figure 24: Program Page 2 (RE-PROGRAM) Operation*



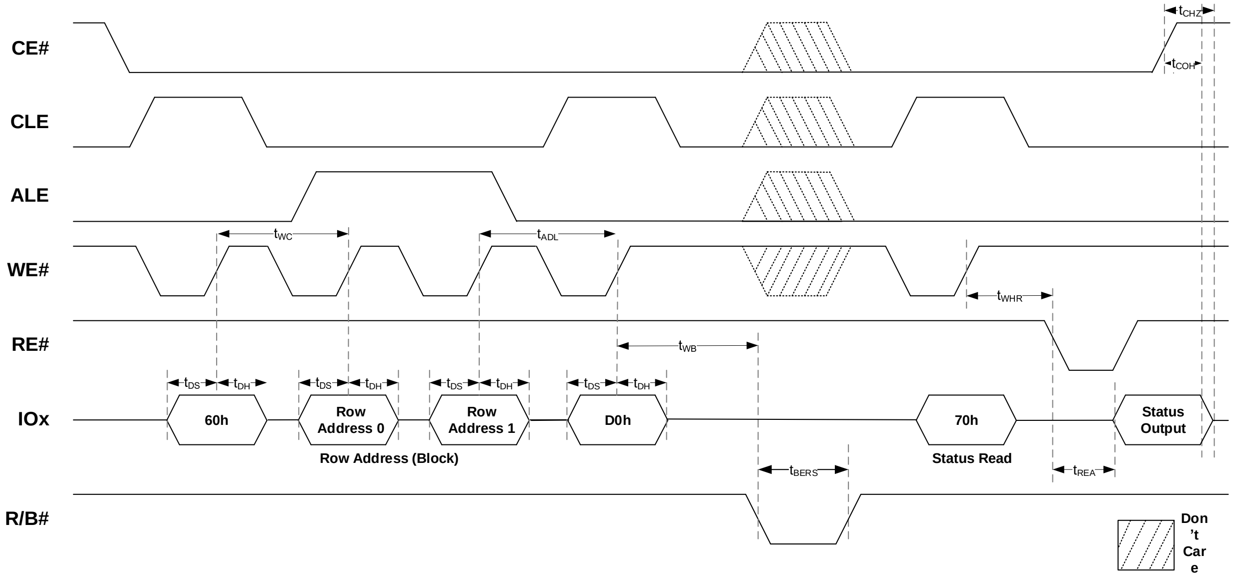
## ERASE BLOCK Operation (60h – D0h)

The erase operation in the device is done on a block basis. The ERASE BLOCK (60h-D0h) command erases the specified block in the NAND Flash array. This command is accepted by the device when it is ready (RDY = 1, ARDY = 1).

To erase a block, write 60h to the command register. Then write two address cycles containing the row address (the page address is not required). Conclude by writing D0h to the command register. The device will go busy (RDY = 0, ARDY = 0) for $t_{BERS}$ while the block is erased.

To determine the progress of the data transfer, the host can either monitor the device's R/B# signal or, execute the READ STATUS REGISTER (70h) command. When the device completes an ERASE Block operation, the host must check the status of the PES2 bit for erase completion status.

*Figure 25: Erase Block Operation*



## READ FOR DATA MOVE Operation (00h – 35h)

The READ FOR DATA MOVE (00h-35h) operation working in conjunction with PROGRAM FOR DATA MOVE (85h-10h) operation provides a very efficient copy-back operation where data stored in one page can be written to another page without utilizing any of the host's resources. It is much faster since loading of the data is not required (if data modifications are not needed).

The READ FOR DATA MOVE (00h-35h) command is functionally identical to the READ PAGE (00h-30h) command, except that 35h is written to the command register instead of 30h.

It is recommended that the host read the data out of the device to verify the data prior to issuing the PROGRAM FOR DATA MOVE (85h-10h) command to prevent the propagation of data errors.

## PROGRAM FOR DATA MOVE Operation (85h – 10h)

The PROGRAM FOR DATA MOVE (85h-10h) command is functionally identical to the PROGRAM PAGE (80h-10h) command, except that 85h is written to the command register.

*Figure 26: Read/Program For Data Move Operation*

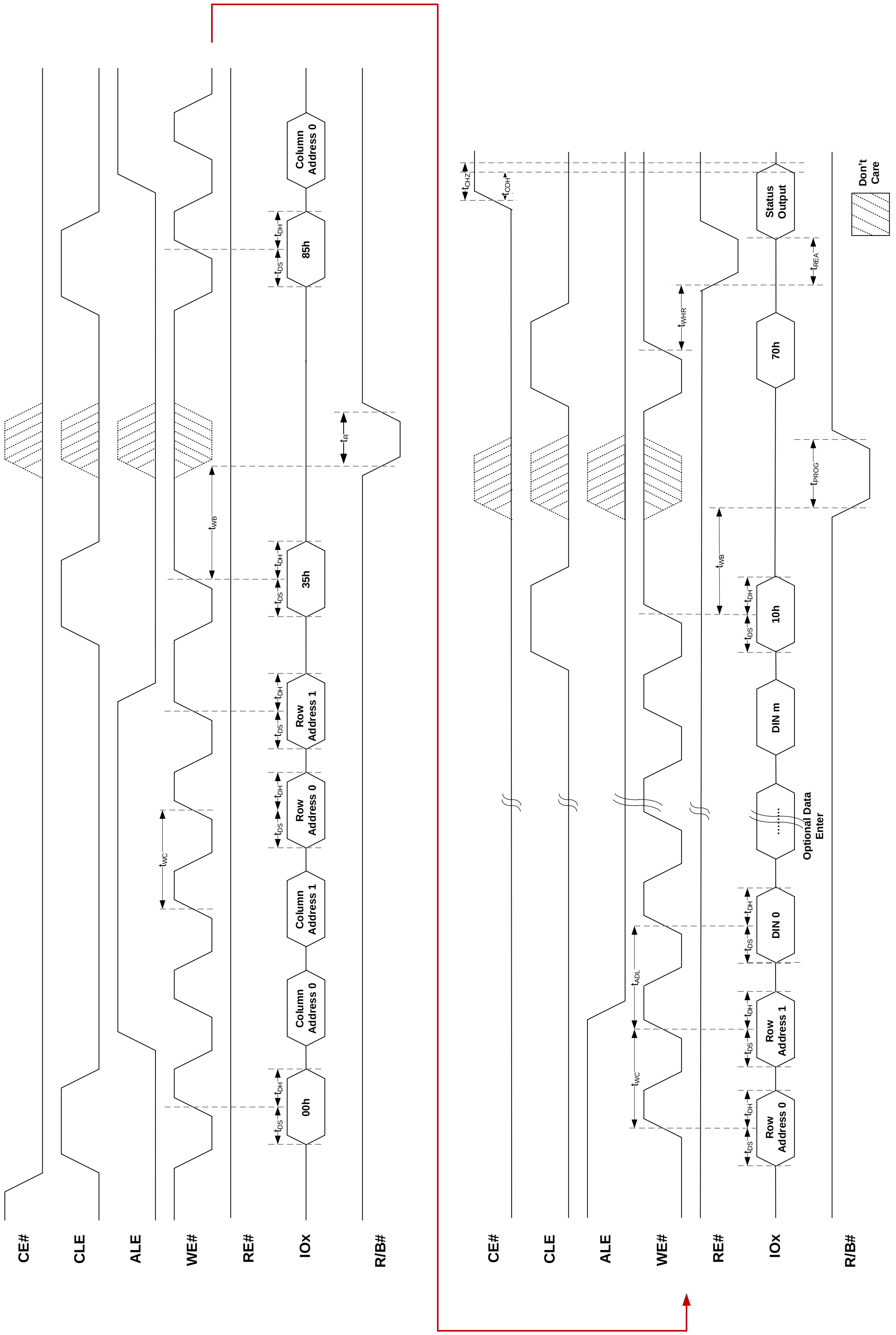
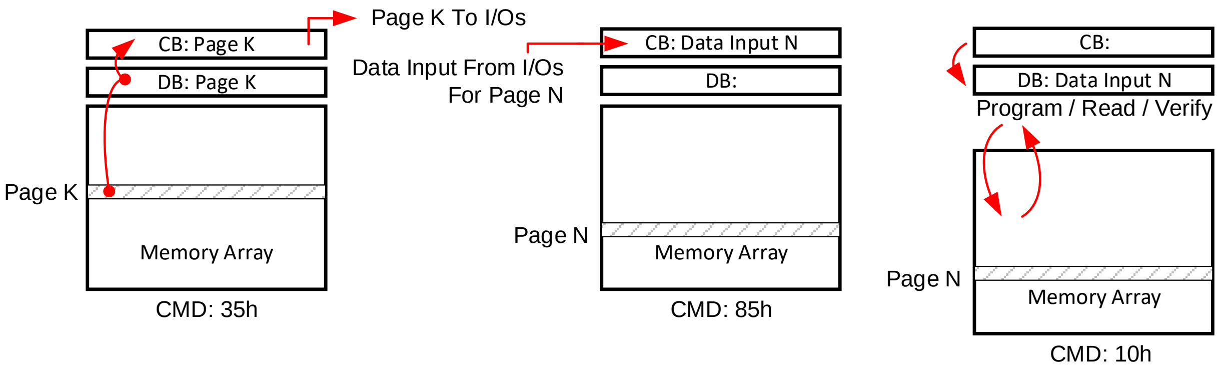


*Figure 27: Read/Program For Data Move Operation Overview*



Data Move operation from Page K to Page N is as following:
1 Data for Page K is transferred to the Cache Buffer (CB) - Read for Data Move.
2 Data for Page K is read out.
3 Data Move to Page address N is input and any data updates are implemented.
4 CB for Page N is transferred to the Data Buffer (DB).
5 By issuing the 10h command, the data in the DB is programmed to Page N – Program for Data Move

## ONE-TIME PROGRAMMABLE (OTP) ENTRY Operation (29h-17h-04h-19h)

The device contains a one-time programmable (OTP) area, which is accessed by writing 29h-17h-04h-19h to the command register. The device is then ready to accept READ PAGE (00h-30h) and PROGRAM PAGE (80h-10h) commands. The OTP area is of a single block size (64 pages), and hence only row addresses between 00h and 3Fh are allowed. The host must issue the Reset command to exit the OTP area and access the normal flash array. The ERASE BLOCK (60h-D0h) command is not allowed in the OTP area.

# Device Initialization

When powering up, the following procedure is required to initialize the device correctly:

- Ramp $V_{CC}$.
- Drive WP# Low during power-up until $V_{CC}$ is stable
- RESET operation is required after $V_{CC}$ ramps up and must be the first command issued. The host must wait 5ms after $V_{CC}$ reaches $V_{CC}$ (minimum) before issuing RESET.
- The device is now initialized and ready for normal operation (after RESET busy time $t_{RST}$ has elapsed - this can be monitored by polling R/B# or issuing the READ STATUS (70h) command).

*Figure 28: Power-On Behavior*

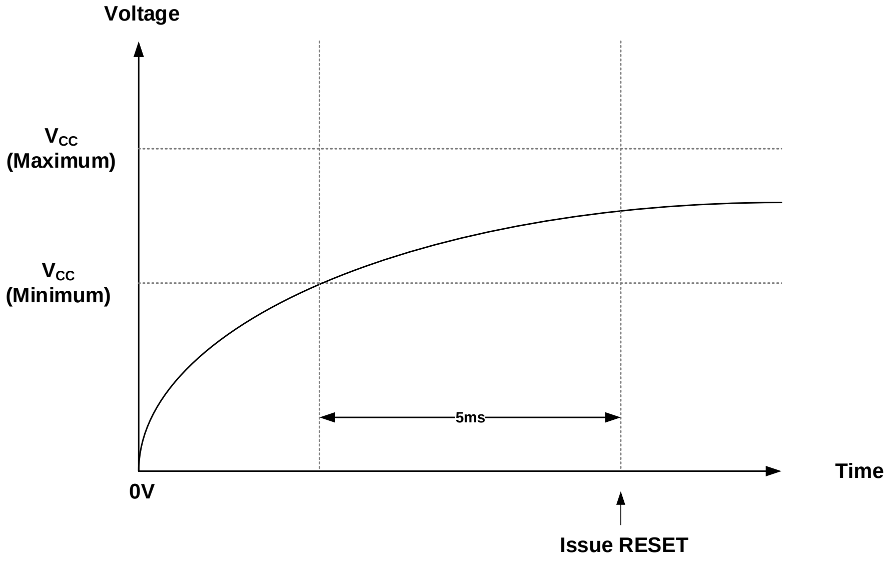
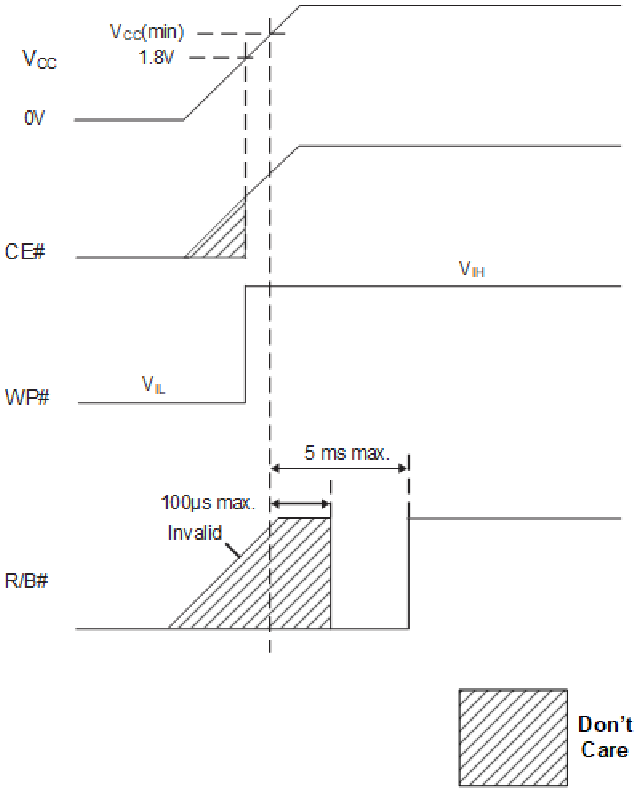


*Figure 29: Power-On Behavior (Continued)*



# Device Power-Down

When powering down, the following procedure is required to initialize the device correctly:

- Ramp down $V_{CC}$.
- Drive WP# Low during power-down before $V_{CC}$ reaches $V_{CC}$ (minimum) and goes below.

*Figure 30: Power-Down Behavior*

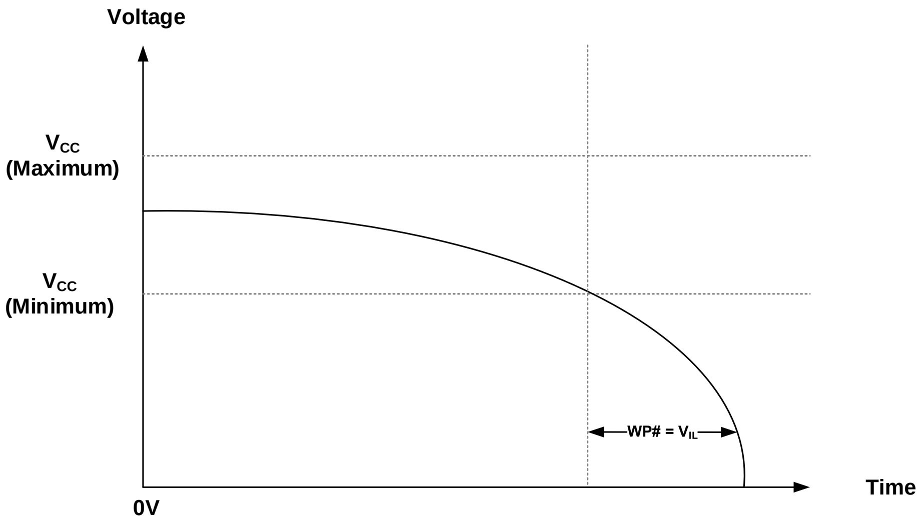
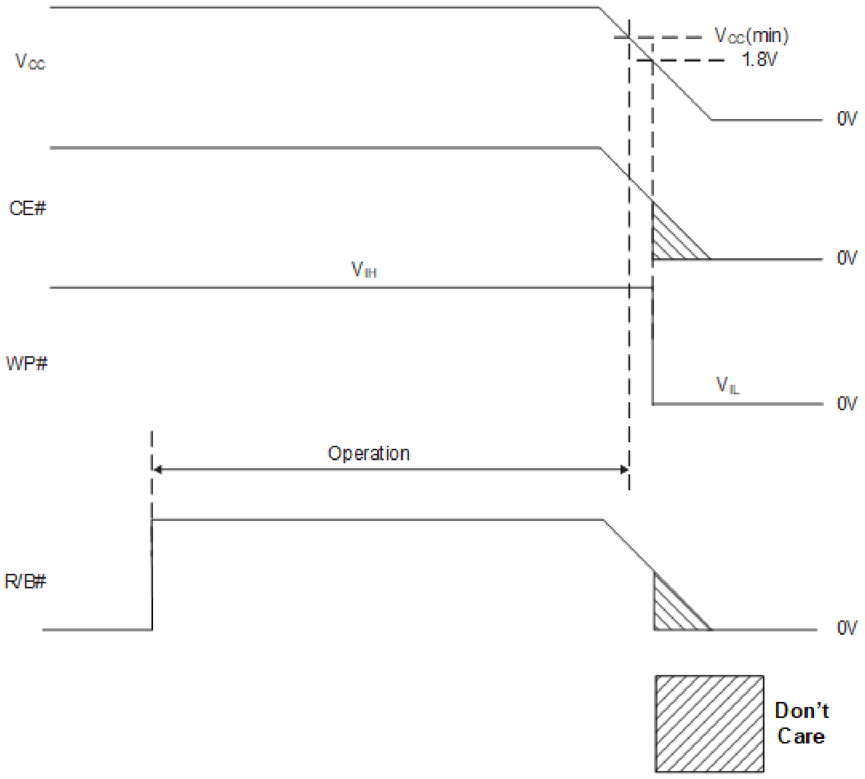


*Figure 31: Power-Down Behavior (Continued)*



# Electrical Specifications

*Table 13: Absolute Maximum Ratings*

| **Voltage on any pin with respect to $V_{SS}$** | | | |
|---|---|---|---|
| **Parameter / Condition** | **Minimum** | **Maximum** | **Units** |
| **Voltage Input** | -0.6 | 4.6 | V |
| **$V_{CC}$ Supply Voltage** | -0.6 | 4.6 | V |
| **Storage Temperature** | -65.0 | 150.0 | °C |
| **Short circuit output current, I/Os** | - | 5.0 | mA |

*Table 14: Recommended Operating Conditions*

| **Parameter / Condition** | | **Minimum** | **Maximum** | **Units** |
|---|---|---|---|---|
| **Operating Temperature** | Industrial | -40.0 | 85.0 | °C |
| **$V_{CC}$ Supply Voltage** | 3.0V | 2.7 | 3.6 | V |
| **$V_{SS}$ Supply Voltage** | | 0.0 | 0.0 | V |

*Table 15: Pin Capacitance*

| **Parameter** | **Test Conditions** | **Symbol** | **Maximum** | **Units** |
|---|---|---|---|---|
| **Input Pin Capacitance** | TEMP = 25°C; f = 1 MHz; $V_{IN}$ = 0V | $C_{IN}$ | 10.0 | pF |
| **Input / Output Pin Capacitance** | TEMP = 25°C; f = 1 MHz; $V_{IN}$ = 0V | $C_{INOUT}$ | 10.0 | pF |

*Table 16: DC Characteristics*

| **Parameter** | **Symbol** | **Test Conditions** | **3.0V Device (2.7V-3.6V)** | | | |
|---|---|---|---|---|---|---|
| | | | **Minimum** | **Typical** | **Maximum** | **Units** |
| **Read Current (Seq)** | $I_{CC1}$ | $t_{RC}$ = 50ns, CE#=$V_{IL}$, $I_{OUT}$=0mA | - | 15.0 | 30.0 | mA |
| **Program Current** | $I_{CC2}$ | | - | 15.0 | 30.0 | mA |
| **Erase Current** | $I_{CC3}$ | | - | 15.0 | 30.0 | mA |
| **Standby Current (TTL)** | $I_{CC4}$ | CE#=$V_{IH}$, WP#=0V/$V_{CC}$ | - | - | 1.0 | mA |
| **Standby Current (CMOS)** | $I_{CC5}$ | CE#=$V_{CC}$-0.2, WP#=0/$V_{CC}$ | - | 10.0 | 50.0 | µA |
| **Input Leakage Current** | $I_{LI}$ | $V_{IN}$=0 to $V_{CC}$ (max) | - | - | ±10.0 | µA |
| **Output Leakage Current** | $I_{LO}$ | $V_{OUT}$=0 to $V_{CC}$ (max) | - | - | ±10.0 | µA |
| **Input High Voltage** | $V_{IH}$ | | 0.8x$V_{CC}$ | - | $V_{CC}$+0.3 | V |
| **Input Low Voltage** | $V_{IL}$ | | -0.3 | - | 0.2x$V_{CC}$ | V |
| **Output High Voltage Level** | $V_{OH}$ | $I_{OH}$ = -100µA | $V_{CC}$-0.4 | - | - | V |
| | | $I_{OH}$ = -400µA | 2.4 | - | - | V |
| **Output Low Voltage Level** | $V_{OL}$ | $I_{OL}$ = 100µA | - | - | - | V |
| | | $I_{OL}$ = 2.1mA | - | - | 0.4 | V |
| **Output Low Current (R/B#)** | $I_{OL}$ (R/B#) | $V_{OL}$=0.1V | - | - | - | mA |
| | | $V_{OL}$=0.4V | 8.0 | 10.0 | - | mA |

*Table 17: Program / Erase Characteristics*

| Parameter | Symbol | 3.0V Device (2.7V-3.6V) | | | |
|---|---|---|---|---|---|
| | | Minimum | Typical | Maximum | Units |
| **Program Time** | $t_{PROG}$ | - | 300.0 | 700.0 | µs |
| **Cache Program Time (Busy Time)** | $t_{PBSY}$ | - | 5.0 | $t_{PROG}$ | µs |
| **Block Erase Time** | $t_{BERS}$ | - | 3.0 | 10.0 | ms |
| **Array Read Time** | $t_R$ | - | - | 25.0 | µs |
| **Cache Read Time** | $t_{RBSY}$ | - | 3.0 | $t_R$ | µs |
| **Number of Partial-Page Programs** | NOP | - | - | 4.0 | cycles |

*Table 18: AC Test Conditions*

| Parameter | Value |
|---|---|
| **Input pulse levels** | 0.0V to $V_{CC}$ |
| **Input rise and fall times** | 5ns |
| **Input and output measurement timing levels** | $V_{CC}/2$ |
| **Output Load** | CL = 50pF |

*Table 19: AC Timing Characteristics*

| Parameter | Symbol | 3.0V Device (2.7V-3.6V) | | |
|---|---|---|---|---|
| | | Minimum | Maximum | Units |
| **CLE Setup time** | $t_{CLS}$ | 12.0 | - | ns |
| **CLE Hold time** | $t_{CLH}$ | 5.0 | - | ns |
| **CE# Setup time** | $t_{CS}$ | 20.0 | - | ns |
| **CE# Hold time** | $t_{CH}$ | 5.0 | - | ns |
| **WE# Pulse width** | $t_{WP}$ | 12.0 | - | ns |
| **ALE Setup time** | $t_{ALS}$ | 12.0 | - | ns |
| **ALE Hold time** | $t_{ALH}$ | 5.0 | - | ns |
| **Data Setup time** | $t_{DS}$ | 12.0 | - | ns |
| **Data Hold time** | $t_{DH}$ | 5.0 | - | ns |
| **Write Cycle time** | $t_{WC}$ | 25.0 | - | ns |
| **WE# High Hold time** | $t_{WH}$ | 10.0 | - | ns |
| **Address to Data Loading time** | $t_{ADL}$ | 70.0 | - | ns |
| **Data Transfer from Cell to Register** | $t_R$ | - | 25.0 | µs |
| **ALE to RE# Delay** | $t_{AR}$ | 10.0 | - | ns |
| **CLE to RE# Delay** | $t_{CLR}$ | 10.0 | - | ns |
| **Ready to RE# Low** | $t_{RR}$ | 20.0 | - | ns |
| **RE# Pulse Width** | $t_{RP}$ | 12.0 | - | ns |
| **WE# High to Busy** | $t_{WB}$ | - | 100.0 | ns |

| Parameter | Symbol | 3.0V Device (2.7V-3.6V) | | |
|---|---|---|---|---|
| | | Minimum | Maximum | Units |
| **Read Cycle Time** | $t_{RC}$ | 25.0 | - | ns |
| **RE# Access Time** | $t_{REA}$ | - | 20.0 | ns |
| **CE# Access Time** | $t_{CEA}$ | - | 25.0 | ns |
| **RE# High to Output Hi-Z** | $t_{RHZ}$ | - | 100.0 | ns |
| **CE# High to Output Hi-Z** | $t_{CHZ}$ | - | 30.0 | ns |
| **CE# High to ALE or CLE Don't care** | $t_{CSD}$ | 10.0 | - | ns |
| **RE# High to Output Hold** | $t_{RHOH}$ | 15.0 | - | ns |
| **RE# Low to Output Hold** | $t_{RLOH}$ | 5.0 | - | ns |
| **CE# High to Output Hold** | $t_{COH}$ | 15.0 | - | ns |
| **RE# High Hold Time** | $t_{REH}$ | 10.0 | - | ns |
| **Output Hi-Z to RE# Low** | $t_{IR}$ | 0.0 | - | ns |
| **RE# High to WE# Low** | $t_{RHW}$ | 100.0 | - | ns |
| **WE# High to RE# Low** | $t_{WHR}$ | 60. | - | ns |
| **Device Resetting Time (Read/Program/Erase)** | $t_{RST}$ | - | 5.0/10.0/500.0 | µs |
| **Write protection time** | $t_{WW}$ | 100.0 | - | ns |

# Error Management

NAND Flash devices have bad blocks that are invalid when shipped from the factory. However, each NAND Flash device specifies a minimum number of valid blocks (NVB) of the total available blocks during the endurance life of the product. An invalid block is one that contains at least one page that has more bad bits than can be corrected by the minimum required ECC. Even though NAND Flash devices contain bad blocks, they can be used reliably in systems provided the systems have bad block management and error-correction algorithms.

Axia's NAND Flash devices are shipped from the factory erased. The factory identifies invalid blocks before shipping by providing a bad block marker (00h to FEh) into the 1st byte in the spare area of the 1st or 2nd page (if the 1st page is Bad). This method is compliant with ONFI Factory Defect Mapping requirements.

System software should check the first spare area location on the first page of each block before performing any program or erase operations on the NAND Flash device, thus creating a bad block table for the whole device.

The following recommendations should be followed to achieve maximum reliability performance:

- Always check the program/erase status bits (PS1, PES2) after a program or erase operation
- Use bad block management and wear-leveling algorithms
- The first block (physical block address 00h) is guaranteed to be valid with ECC when shipped from the factory (1K program/erase cycles)

*Table 20: Error Management Details*

| Description | Requirement |
|---|---|
| **Minimum number of valid blocks (NVM)** | 1004 |
| **Total number of available blocks** | 1024 |
| **First spare area location** | x8: byte 2048, x16: word 1024 |
| **Bad block mark** | x8: 00h - FEh, x16: 0000h - FFFEh |

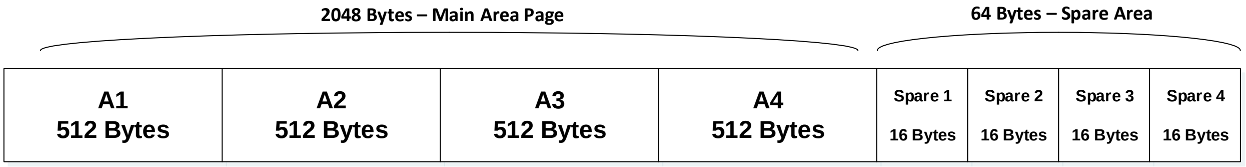
# ECC Management

Digital data stored in SLC NAND Flash is prone to a certain number of errors which must be detected and corrected. Error correction codes (ECC) are widely implemented on the host side which encode data in such a way that a decoder can identify and correct errors in the data.

The most commonly used ECC codes in the industry are Hamming, Reed-Solomon and Bose-Chaudhuri-Hocquenghem (BCH).

Hamming codes are widely used to detect 2-bit errors and correct 1-bit errors. As an example, according to the Hamming ECC principle, a 22-bit ECC is needed to perform a 1-bit correction per 256 bytes. For 512 bytes, a Hamming ECC principle can be used that generates a 24-bit ECC per 528 bytes to perform a 2-bit detection and a 1-bit correction. For 2112-byte page NAND devices, the calculation can be done per 512 bytes, which means a 24-bit ECC per 4096 bits (exactly 3 bytes per 512 bytes). 2112-byte pages are divided into 512-byte (main area page) + 16-byte (spare) chunks. Figure 32 shows how the main and spare page areas can be used to implement the Hamming codes.

*Figure 32: Main Page divided into Chunks*



BCH codes are more popular than Reed-Solomon due to their improved efficiency. However, BCH can only correct 1-bit errors which are more typical in NAND Flash devices. Most new processor designs anticipate coupling with NAND Flash devices and include 4-bit and greater ECC engines within the hardware itself. Axia's recommendation for BCH based ECC is shown in Table 21:

*Table 21: Minimum ECC Requirements*

| Description | Requirement |
|---|---|
| **Minimum required ECC** | 4-bit ECC per 528 bytes of data |

# Product Use Limitations

Axia reserves the right to make changes to the information in this document, and related hardware, software and system (collectively referred to as “**Products**”) without notice. This document supersedes and replaces any and all prior or contemporaneous information or arrangement of any kind relating to the subject matter hereof. This document and any information set forth herein may not be reproduced without the prior written permission from Axia.

**Critical Applications**. Products are not authorized for use in applications in which failure of the Axia component could result, directly or indirectly in death, personal injury, or severe property or environmental damage ("**Critical Applications**"). Axia assumes no liability for Products if used for Critical Applications. Should customer or distributor purchase, use, or sell any Axia component for Critical Applications, customer and distributor shall indemnify and hold harmless Axia and its subsidiaries, subcontractors, and affiliates and the directors, officers, and employees of each against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such Critical Application, whether or not Axia or its subsidiaries, subcontractors, or affiliates were negligent in the design, manufacture, or warning of the Products.

**Automotive Applications**. Products are not authorized for use in automotive applications unless specified by Axia. Axia assumes no liability for Products if used for automotive applications. Distributor and customer shall assume the sole risk and liability for such use and shall indemnify and hold Axia harmless against all claims, costs, damages, and expenses and reasonable attorneys' fees arising out of, directly or indirectly, any claim of product liability, personal injury, death, or property damage resulting directly or indirectly from any use of Products in automotive applications.

**Military Applications**. Products are not authorized for use in military applications, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons).

**Customer Responsibility**. Customers are solely responsible for the design and operation of their systems using Products. It is customer’s sole responsibility to determine whether Products are suited for their system. Customers are required to incorporate good safety design measures and adequate safeguards to eliminate risks of personal injury, death, or severe property or environmental damages that could result from failure of Products. **AXIA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS AND DAMAGES OR LOSSES OCCURING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

The information contained herein is presented only as guidance for Product use. Axia assumes no responsibility for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Products. No license to any intellectual property right is granted by this document, whether expressed or implied. You may not perform compositional, structural, functional or other analysis of Products, or undertake deconvolution or reverse engineering with respect to Products.

**Limited Warranty**

In no event shall Axia and its representatives be liable for any indirect, incidental, punitive, special or consequential damages (including but not limited to loss of use, revenue or profit), whether or not such damages are based on tort, warranty, breach of contract or other legal theory.

In no event shall Axia’s aggregate liability for any breach, warranty, indemnity or other obligation or liability arising out of or in connection with the sale of Products or the use of any Products exceed the purchase price of the particular Product(s) or services with respect to which losses or damages are claimed.

# Revision History

| Revision | Date | Change Summary |
|---|---|---|
| **REV E** | 05/17/2018 | Initial release |
| **REV F** | 04/14/2019 | Added Typ. to Endurance<br>Updated Table 2: Valid Combinations List<br>Updated Figure 35: Power-On Behavior (Continued) and Figure 37: Power-Down Behavior (Continued) |