# W25N02KVxxIR/U

## 2. FEATURES

### 1.Overview

* 2G-bit (256MB) SLC QspiNAND Flash Memory.
* Combines SPI interface with large NAND capacity.
* Supports code shadowing, XIP execution, and data storage.
* Operates on a single 2.7V–3.6V supply with ultra-low power modes.

### 2.Memory Organization

* 131,072 pages × 2KB each.
* Erase granularity: 128KB block (64 pages).
* Total: 2,048 erasable blocks.

### 3.Interface & Performance

* Standard SPI, Dual, and Quad I/O modes.
* Clock up to 104MHz.
* Effective data rates: 208MHz (Dual), 416MHz (Quad).
* Sequential Read Mode for efficient memory access.
* Transfer speed up to 50MB/s.

### 4.Endurance & Reliability

* 60,000 program/erase cycles.
* 10-year data retention.
* On-chip 8-bit ECC with status reporting.

### 5.Power & Temperature

* 25mA active, 10µA standby, 1µA deep power down.
* Wide operating temperature: -40°C to +85°C.

### 6.Advanced Features

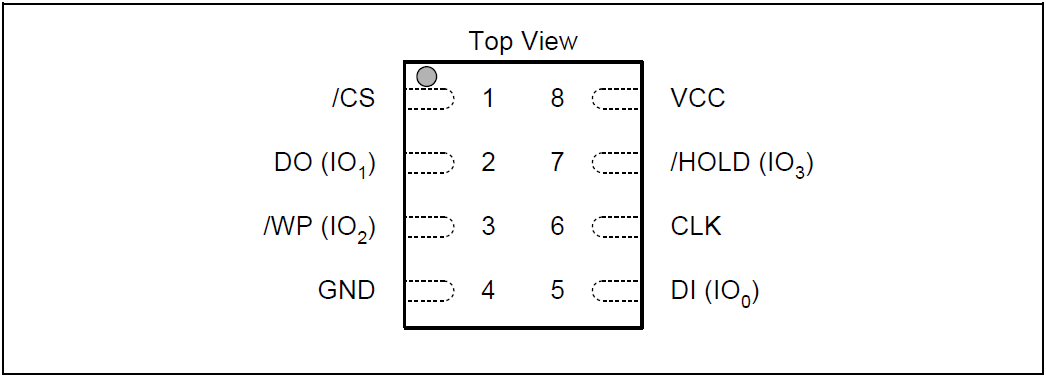
* Software/Hardware write protection.
* Power Supply Lock-Down and OTP security.
* Unique ID and parameter pages.
* 10 OTP pages (2KB each).

### 7.Packaging

* Compact packages: 8-pad WSON, 24-ball TFBGA, 16-pin SOIC.
* Designed for space-constrained applications.

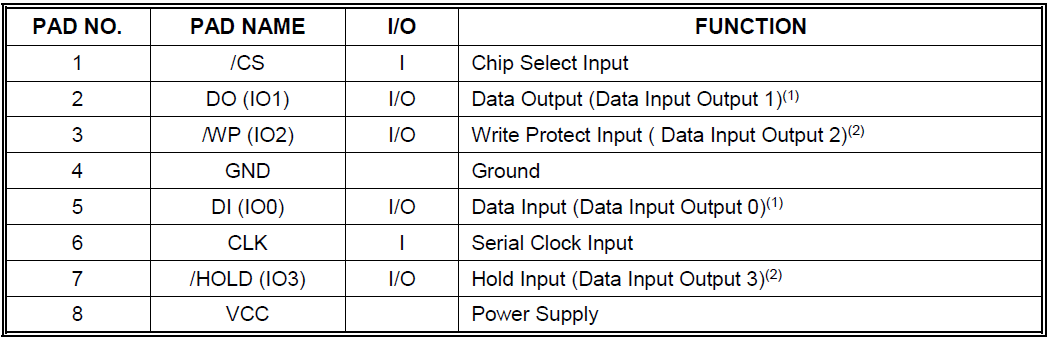
## 3. PACKAGE TYPES AND PIN CONFIGURATIONS

### 3.1 Pad Configuration WSON 8x6-mm



**W25N02KV Pad Assignments, 8-pad WSON 8x6-mm (Package Code ZE)**

### 3.2 Pad Description WSON 8x6-mm



NOTES:

=> IO0 and IO1 are used for Standard and Dual SPI instructions

=> IO0 – IO3 are used for Quad SPI instructions, /WP & /HOLD functions are only available for Standard/Dual SPI.

## 4. PIN DESCRIPTIONS

### 1.Chip Select (/CS)

* The SPI Chip /CS pin enables and disables device operation.
* High = device deselected (DO/IO pins high-Z, low power).
* Low = device active, instructions accepted.
* After power-up, /CS must transition from high to low before a new instruction will be accepted.

### 2.Serial Data Input, Output and IOs (DI, DO and IO0, IO1, IO2, IO3)

* W25N02KV supports Standard, Dual, and Quad SPI operations.
* DI (input) : serially writes instructions, addresses, or data (on rising edge of CLK).
* DO (output) : reads data or status (on falling edge of CLK).
* Dual/Quad SPI : uses bidirectional IO pins for read/write.

### 3.Write Protect (/WP)

* WP-E = 0 ---> Software Protection Mode (only SR-1 can be protected).
* WP-E = 1 ---> Hardware Protection Mode.
* If /WP is tied to GND, all Write/Program/Erase functions are disabled; all registers, memory array, and OTP pages become read-only.

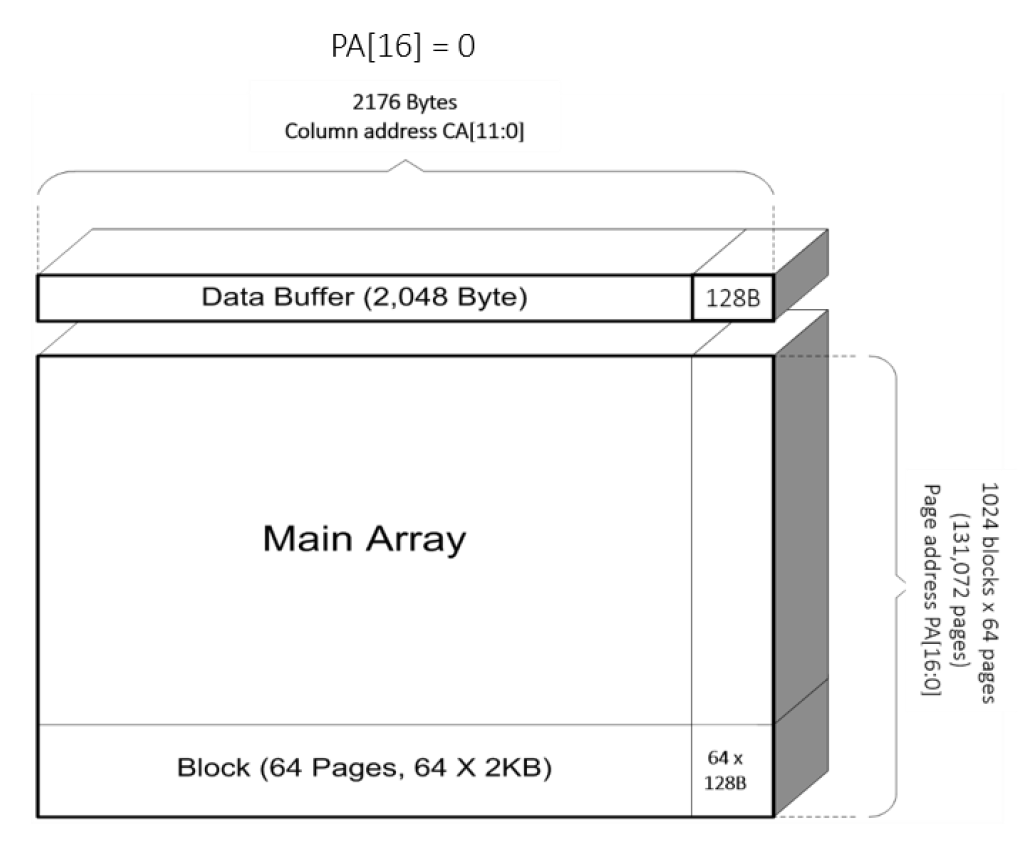
### 4.HOLD (/HOLD)

* Pauses device activity in Standard / Dual SPI.
* When /HOLD and /CS = low ---> DO and CLK signals ignored.
* When /HOLD = high ---> device operation resumes.
* In Quad SPI, /HOLD becomes IO3.

### 5.Serial Clock (CLK)

* Provides the timing for all serial input and output operations.

## 5. BLOCK DIAGRAM(1)

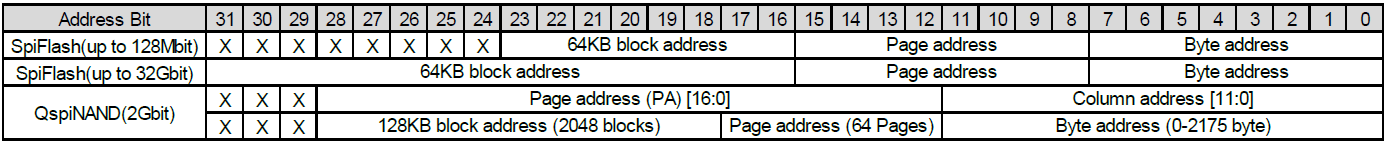


**W25N02KV Flash Memory Architecture and Addressing**

Memory Array Strcuture:

* **Composed of Blocks ( each block has 64 Pages, each page 2048 Bytes ).**
* **Data Buffer ( 2048 Bytes ) is used for page program / read operations.**
* **Each Page has 2176 Bytes of addressable column space ( 2048 B data + 128 B overhead ).**
* **Column address is CA[11:0] ( 0~2175 )**
* **PA[16:0] is the page address (total 131072 pages = 2048 blocks \* 64 pages ).**
* **2KB of main data all in Data Buffer**
* **Additional 128B ( Spare + Parity ) not include 2KB Buffer, But still accessed using field address CA, so the entire page is represented by 2176-byte displacement.**
* **Total 131072 pages, 17 bits (PA[16:0]) are required to locate the page.**

## 5. BLOCK DIAGRAM(2)

****

**W25N02KV Flash Memory Architecture and Addressing**

### SpiFlash (up to 128 Mbit)

* It has smaller capacity (128 Mbit = 16 MB)
* Address bit assignment:
  + [00:07] : Byte address, which determines which in page.
  + [08:15] : Page address.
  + [16:23] : 64 KB Block address.
  + [24:31] : Unused(X) => Because the capacity is small, so many address are not used.

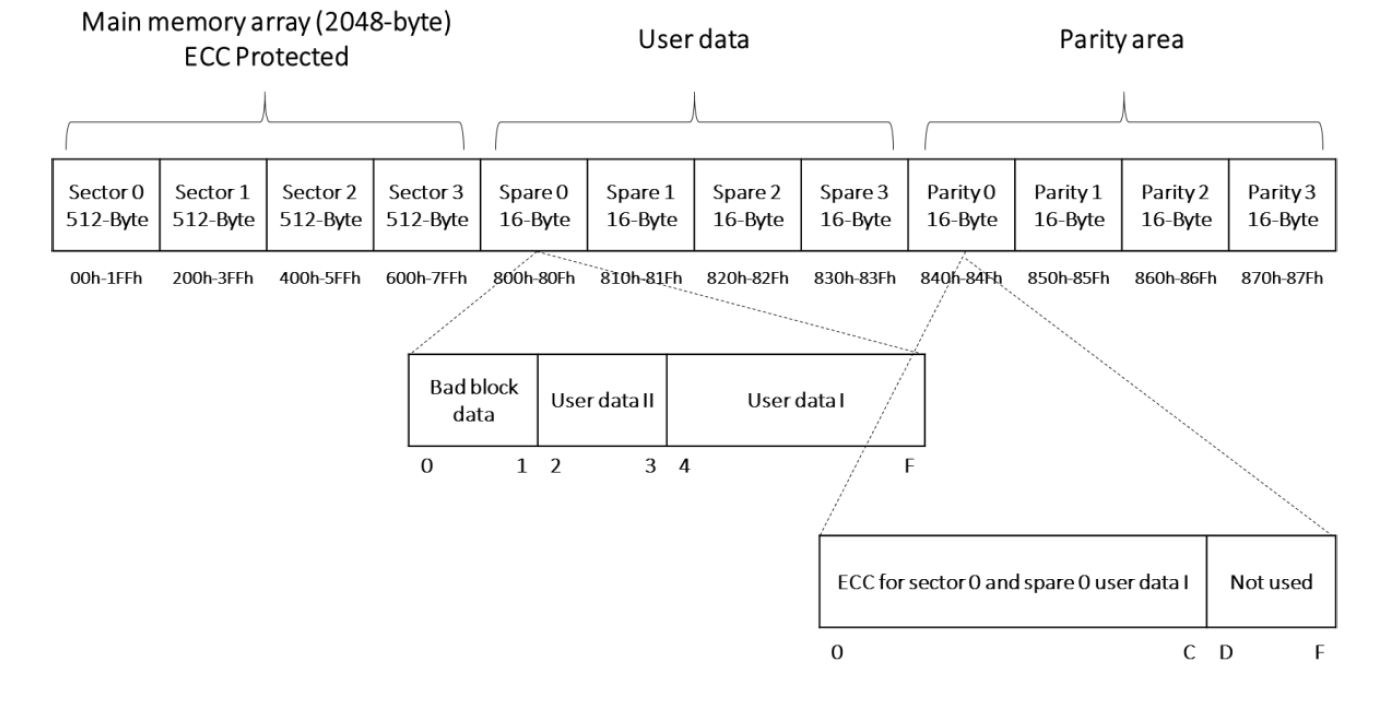
### SpiFlash (up to 32 Gbit)

* It has large capacity ( 32 Gbit = 4GB )
* Address bit assignment
  + [00:07] : Byte address, which determines which in page.
  + [08:15] : Page address.
  + [16:31] : 64 KB Block address.

### QspiNAND(2 Gbit)

* NAND Flash, with a different structure than SPI Flash
* Address bit allocation
* Column address + Page address
* [00:11] : Column address.
* [12:28] : Page address PA[0:16](16 bit) => Page address + block address.
* Byte address + Page address + 128KB block address (2048 blocks)
* [00:11] : Byte address (0 ~ 2175 byte)
* [12:17] : Page address (64 Pages)
* [18:28] : 128KB block address (2048 blocks)

## BLOCK DIAGRAM(3)

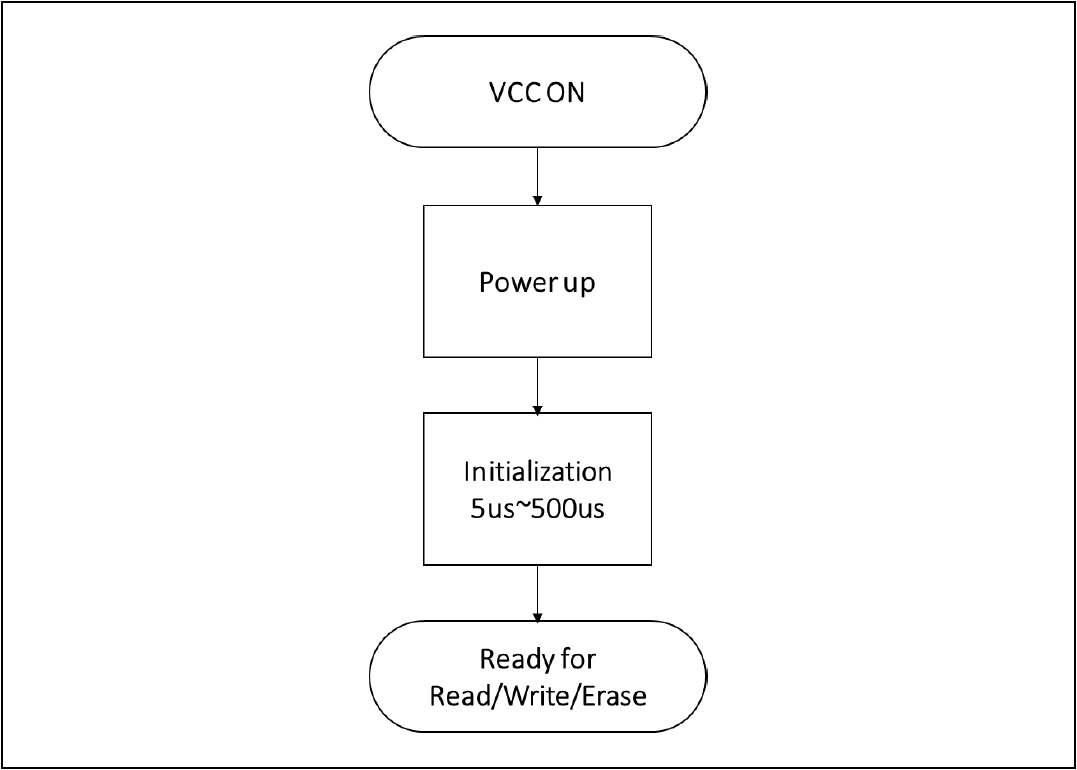


**W25N02KV Flash Memory Architecture and Addressing**

* This page shows the layout of one NAND page in W25N02KV.
* A page has 2176 bytes in total, split as follow:
* Page size = 2176 B : 2048 B data + 64 B spare + 64 B parity
* Data is split into 4 x 512 B sector ; ECC is tracked per sector.
* Spare 0, byte[0] = Bad Block Markter (BBM) ; non-FF typically means the block is bad.
* User metadata (logical page index, wear-level info, CRC, flags, etc.) can go into Spare bytes ( User data I/II ).
* Parity area holds ECC bytes produced/consumed by the device when internal ECC is enabled.
* Column map (must-know):
  + ( 0x000 ~ 0x7FF ) --> Main Data (2KB)
  + ( 0x800 ~ 0x83F (64B) --> Spare0[0] = BBM ) --> Spare
  + ( 0x840 ~ 0x87F (64B) --> ParityN 對應 SectorN ) --> parity (ECC)
* Typical FW Flow:

1. Page cache -> Read cache from desired column range ;
2. In programing, provide data ( + optional spare ), device updates parity ;
3. After read check status / ECC bits ; use BBM to skip bad blocks.

## FUNCTIONAL DESCRIPTIONS(1)



**W25N02KV Flash Memory Operation Diagram**

* Flash Memory Flow:
* VCC ON -> Device recevice power.
* Power up -> Internal circuits begin startup.
* Initialization ( 5µs ~ 500 µs ) -> Device needs this time to stabilize internal logic and register states.
* Ready for Read / Write / Erase -> After initialization, device guarantees correct operation.
* If the initialization delay is skipped or too short:

1. Read commands may return invalid or corrupted data.
2. Write/erase commands may fail or corrupt memory cells.
3. The status register may report wrong values.
4. Device reliability and data retention can be compromised.

## 6. FUNCTIONAL DESCRIPTIONS(2)

### Standard SPI Instructions

* Pins : 4 singals -> CLK、/CS、DI、DO.
* Data Flow:
  + Instructions, addresses, data input via DI on rising edge of CLK.
  + Data/status output via DO on falling edge of CLK.
* Modes supported: Mode 0 (CLK idle low) & Mode 3 (CLK idle high).
* Use case: Basic access, lowest bandwidth.

### Dual SPI Instructions

* Pins: IO0 and IO1 (DI & DO repurposed as bidirectional I/Os).
* Speed: 2 ~ 3 × faster than Standard SPI.
* Instructions : (3Bh), (BBh).
* Use case:
  + Faster reads.
  + loading firmware from Flash into RAM.
  + Supports XIP (Execute-In-Place) for non-speed-critical code.

### Quad SPI Instructions

* Pins : IO0、IO1、IO2、IO3
  + DI/DO -> IO0/IO1、/WP -> IO2、/HOLD -> IO3
* Speed : 4 ~ 6 x faster than Standard SPI.
* Instructions : (6Bh/6Ch)、(EBh/ECh)、(32h/34h).
* Use case:
  + Very fast reads.
  + Suitable for random access acceleration.
  + Commonly used for code shadowing and XIP with higher performance.

### Hold Function

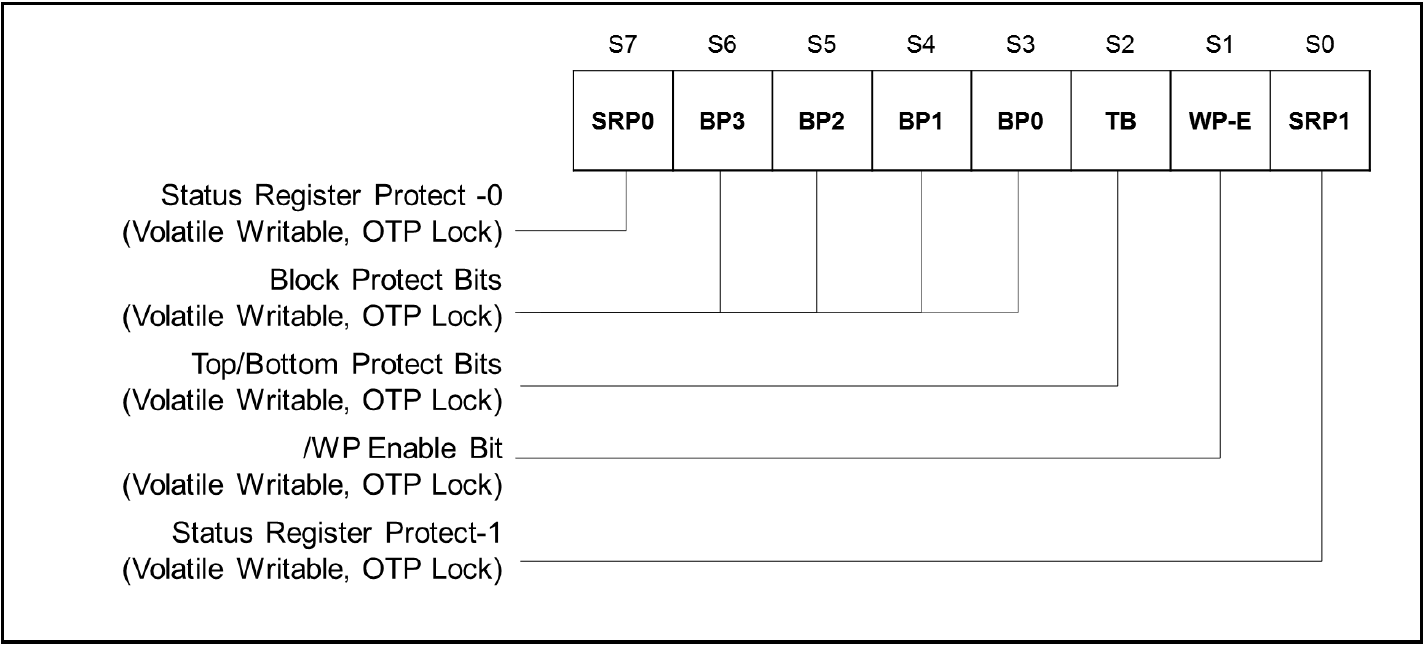
* Available in: Standard SPI & Dual SPI (NOT supported in Quad SPI).
* Behavior:
  + - Must have /CS = Low
  + Trigger timing (/HOLD):
  + CLK(Low) = activates falling edge. | CLK(High) = activates rising edge.
  + During /HOLD:
  + DO = High impedance (Hi-Z). | DI & CLK are ignored. | Device keeps its state.
  + Exiting /HOLD: happens on next opposite edge of /HOLD signal.
* Limitation: In Quad SPI mode, /HOLD pin is reused as IO3 -> no pause support.

### Write Protection

* Write protection features include:
  + Power-related reset ( disable ops when VCC < VCC(min) ).
  + Automatic write-disable after program/erase.
  + Software + hardware protection via /WP pin & SR-1.
  + Lock Down protection for SR-1 (until next power cycle).
  + OTP protection for memory array.
  + Hardware WP via /WP + WP-E bit.
* Power-up behehavior:
  + All operations disable while VCC < VCC(min).
  + After VCC(min) + tVSL ---> still must wait tPUW before enabling program / erase.
  + /CS must track VCC to avoid spurious commands.
* WEL (Write Enable Latch):
  + Default = 0 (Write disable).
  + Must issue Write Enable before program / erase.
  + Auto-cleared to 0 after each operation.
* Software protection:
  + Control via SRP0, SRP1, TB, BP[3:0] bits.
  + Configure partial or full memory as read-only.
* WP-E hardware protection:
  + WP-E = 1 + /WP Low ---> device = read-only, no program/erase.
  + Quad SPI ops also disable when WP-E = 1.

## 7. PROTECTION, CONFIGURATION AND STATUS REGISTERS

### 7.1 Block Protect Bits ( BP3 ~ BP0 ):Volatile Writable, OTP lockable



**Protection Register / Status Register-1 (Address Axh)**

* Three status registers: SR-1 (Protection), SR-2 (Configuration), SR-3 (Status).
* Accessed via Read/Write Status Register instructions with 1-byte address.
* Read Status Register (05h/0Fh) ---> reports flash availability, write enable/disable state, write protection, read mode, OTP lock, program/erase result, ECC usage.
* Write Status Register ---> configures write protection, SW/HW write protection, read mode, ECC enable/disable, OTP lock.
* Write access control ---> determined by SRP0/SRP1 bits, Write Enable instruction, WP-E & /WP pin.
* Status Register-1 (SR-1) contains:
  + SRP0, SRP1 ---> status register protection.
  + BP0-BP3, TB ---> block protection bits.
  + WP-E ---> /WP enable bit.
* Block Protect bits (BP3–BP0, TB): define protected memory range (none, part, all). Default = all protected after power-up.
* If SR1-L = 1 in SR-2 ---> default values locked by OTP.

### 7.1.1 Block Protect Bits ( BP3 ~ BP0 )

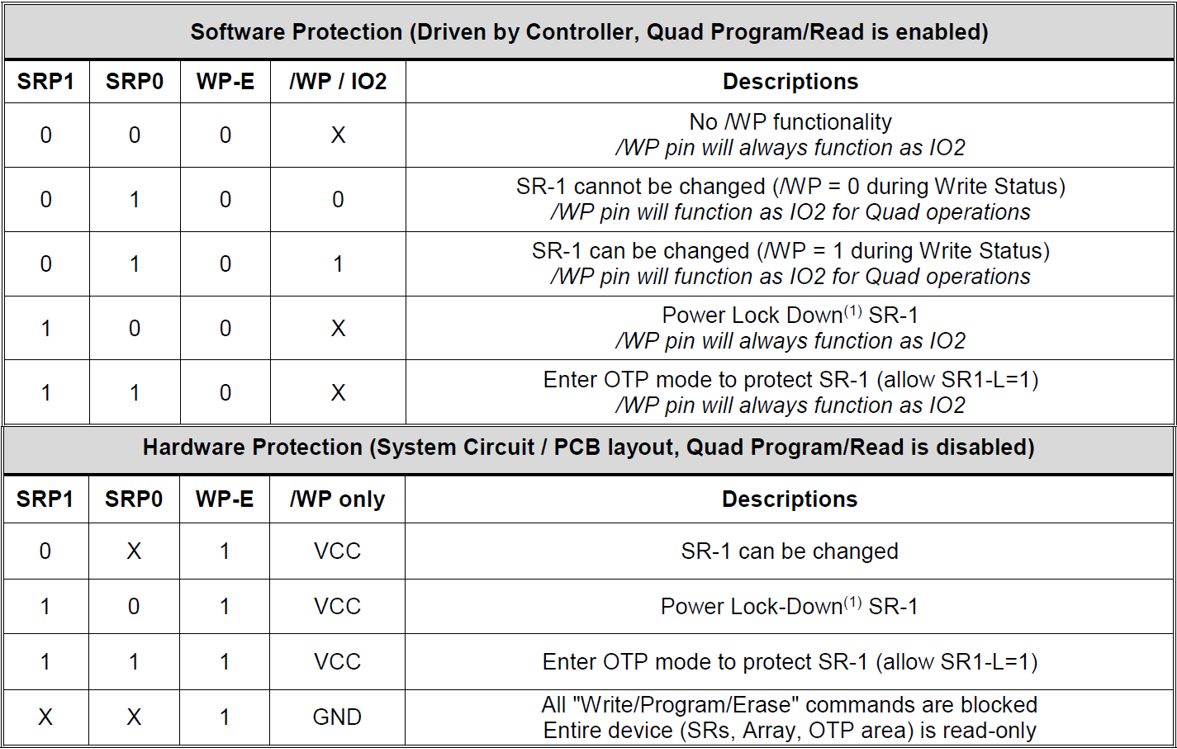
* Locate in SR-1 (S6 ~ S2).
* Control write protection (all, part, or none of memory array).
* Default after power-up:all protected.
* If SR1-L = 1 in SR-2 ---> default = OTP locked values.

### 7.1.2 Write Protection Enable Bit ( WP-E )

* Works with SRP1 & SRP0.
* Control write protection method:
  + Software protection
  + Hardware protection
  + Power supply lock-down
  + OTP protection
* Also affects /WP pin function and Quad SPI enable/disable.
* WP-E = 0 ---> Software mode (default).
* WP-E = 1 ---> Hardware mode, Quad disabled, /WP & /HOLD become dedicated control pins.

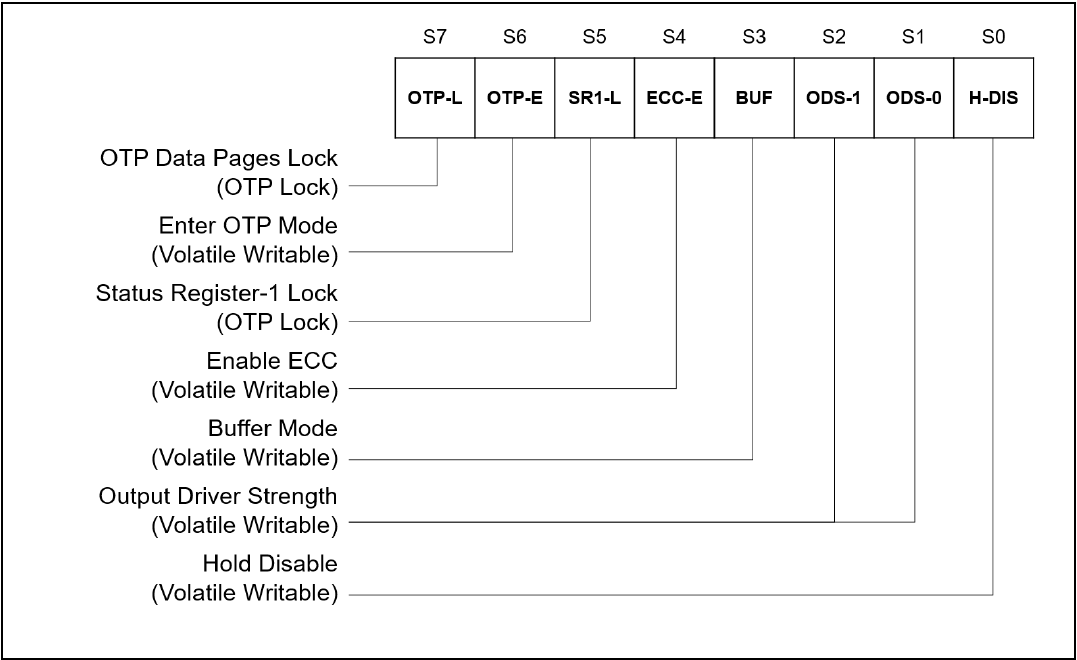
### 7.1.3 Block Protect Bits ( BP3 ~ BP0 )

* Located in SR-1 (S0 & S7).
* Define method of write protection: SW, HW, power lock, OTP.



Note : When SRP1, SRP0 = (1, 0), a power-down, power-up cycle will change SRP1, SRP0 to (0, 0) state.

### 7.2 Configuration Register / Status Register-2 ( Volatile Writable )



**Configuration Register / Status Register-2 (Address Bxh)**

### 7.2.1 One Time Program Lock Bit ( OTP-L ) - OTP lockable

* Locks OTP area permanently.
* OTP area = 10 pages × 2176B each.
* Default = FFh.
* Program only 1 ---> 0, not reversible.

### 7.2.2 Enter OTP Access Mode Bit ( OTP-E ) - Volatile Writable

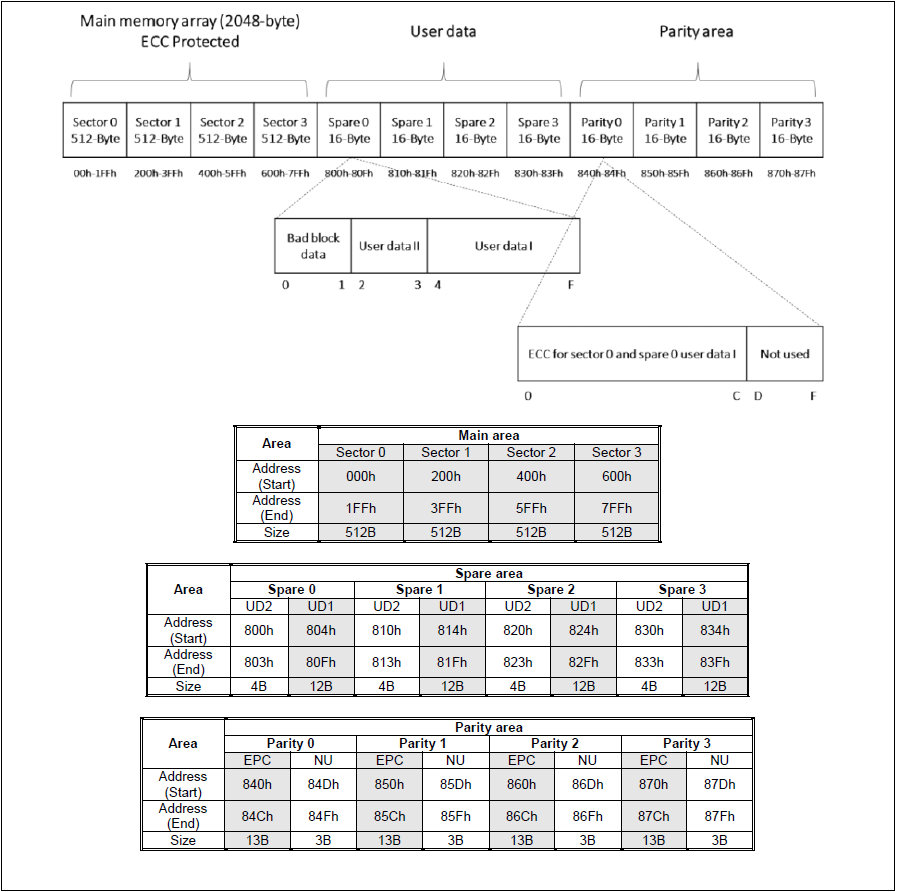
* Must = 1 to read/write OTP area and access Unique ID/Parameter Page.
* Default after reset = 0.

### 7.2.3 Status Register-1 Lock Bit ( SR1-L ) - OTP lockable

* Locks SR-1 values permanently.
* Requires SRP1 = SRP0 = 1 and OTP-E = 1 to set SR1-L = 1.

### 7.2.4 ECC Enable Bit ( ECC-E ) - Volatile Writable

* Built-in ECC, auto-calculated during page program.
* Stores ECC data in extra 64B / page.
* Enabled by default ( ECC-E = 1 ), not cleared by Reset.
* Constraints:
  + ECC protects User Data I ( not User Data II ).
  + Max 4 partial page programs ( NoP = 4 ).
  + Must write sector + spare data together to keep ECC parity valid.



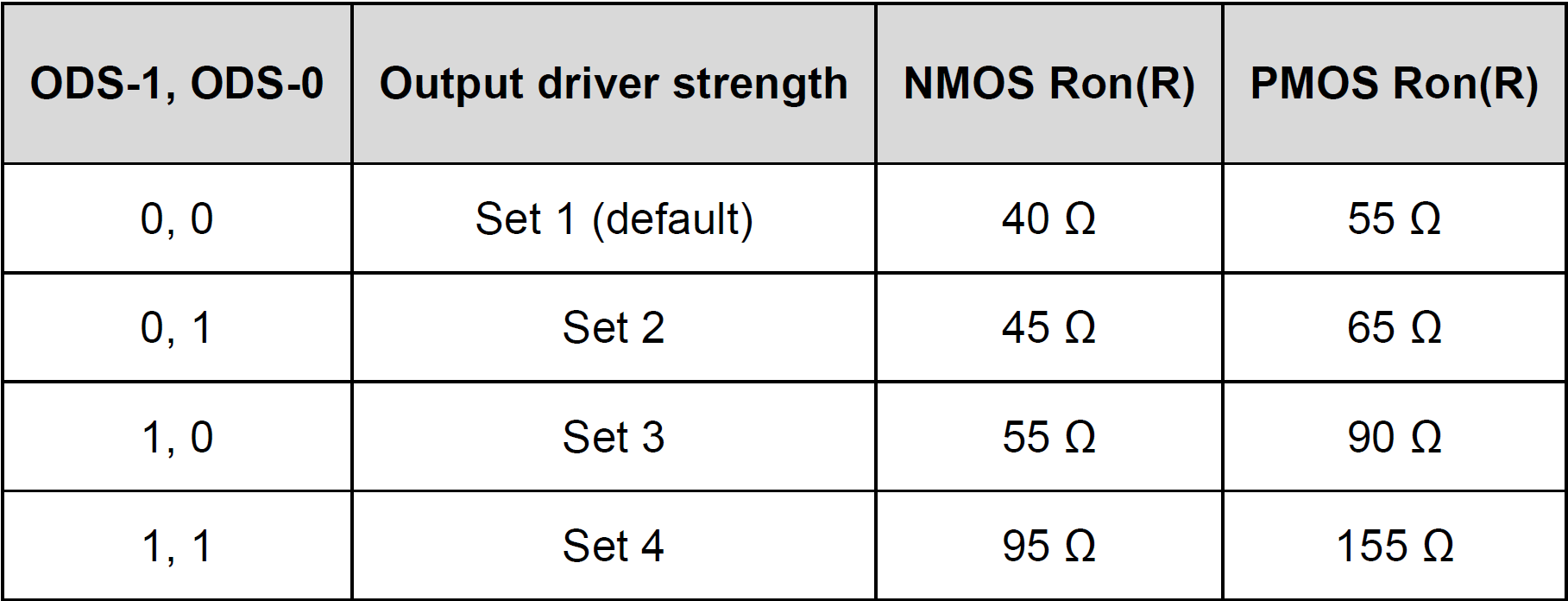
[W25N02KVxxIE datasheet](https://www.winbond.com/hq/support/documentation/?__locale=zh_TW&line=/product/code-storage-flash-memory/index.html&family=/product/code-storage-flash-memory/qspinand-flash/index.html&pno=W25N02KV&category=/.categories/resources/datasheet/)

Notes:

1. UD2 : User Data II
2. UD1 : User Data I
3. EPC : ECC Parity Code
4. NU : Not use

The gray area of the above table is protected by ECC

### 7.2.5 Output Driver Strength ( ODS-1, ODS-0 ) – Volatile Writable

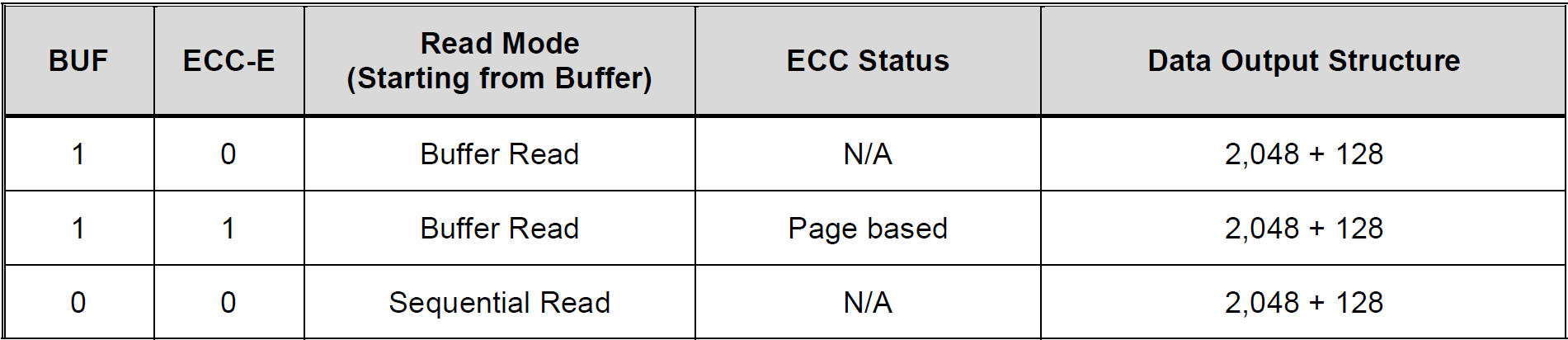


### 7.2.6 Hold Disable ( H-DIS ) - Volatile Writable

* H-DIS = 1 ---> /HOLD function will disable.

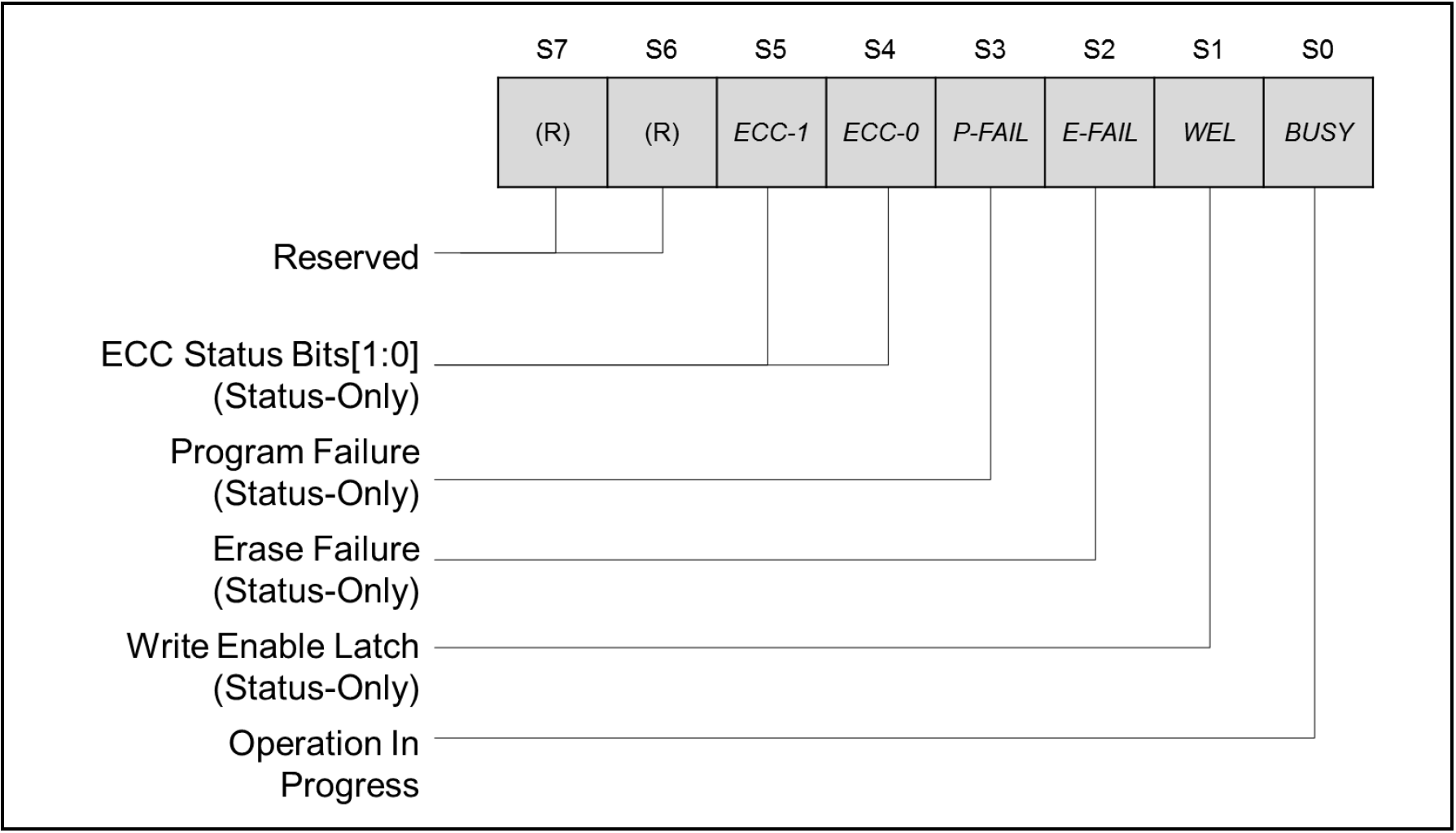
### 7.2.7 Buffer Read/Sequential Read Mode Bit (BUF) - Volatile Writable

* BUF = 1 ( Buffer Read Mode )
  + Needs Column Address
  + Stops at end of buffer (2176B)
  + With ECC-E = 1 ---> ECC page-based protection enabled.
* BUF = 0 ( Sequential Read Mode )
  + Starts at Byte 0, continues page by page.
  + Single read command can dump entire array.
  + ECC is NOT applied ( regardless of ECC-E ).



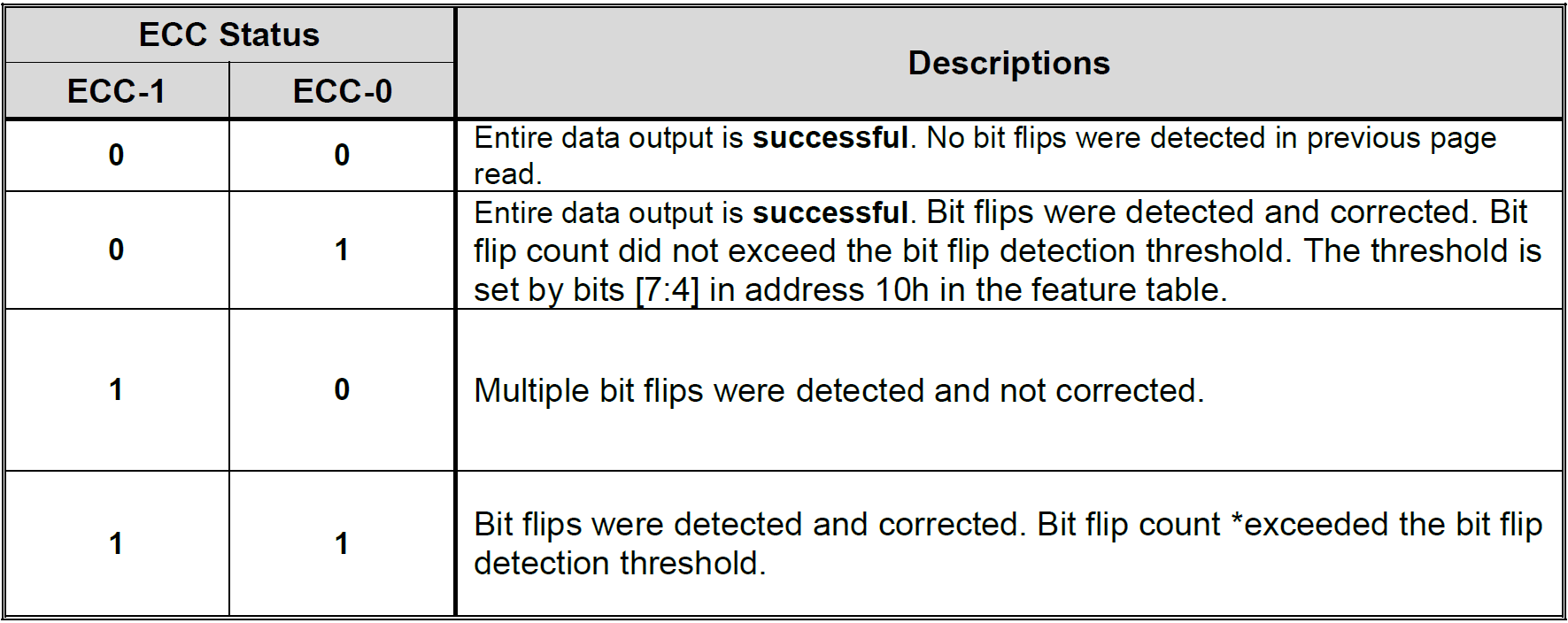
Note : When BUF set to 0, no matter which setting for ECC-E, there is no built-in ECC algorithm to preserve the data integrity.

### 7.3 Status Register-3 (Status Only)



**Status Register-3 (Address Cxh)**

### 7.3.1 Cumulative ECC Status (ECC-1, ECC-0) - Status Only



Note : Bit flip count > BDF setting

* These bits will be cleared to 0 after a power cycle or a RESET command or a Page Data Read Command.
* ECC Status (ECC-1, ECC-0)
  + (0,0) : No Errors
  + (0,1) : OK, corrected errors < threshold
  + (1,0) : Fail, multiple uncorrected errors
  + (1,1) : OK, corrected but exceeded threshold

### 7.3.2 Program Failure ( P-FAIL ) - Status Only

* Indicates if Program operation executed successfully.
* P-FAIL = 0 ---> Success
* P-FAIL = 1 ---> Fail (timeout, or command issued to locked/protected memory/OTP area).

### 7.3.3 Erase Failure ( E-FAIL ) - Status Only

* Indicates if Erase operation executed successfully.
* E-FAIL = 0 ---> Success
* E-FAIL = 1 ---> Fail (timeout, or erase command to locked/protected memory/OTP area).

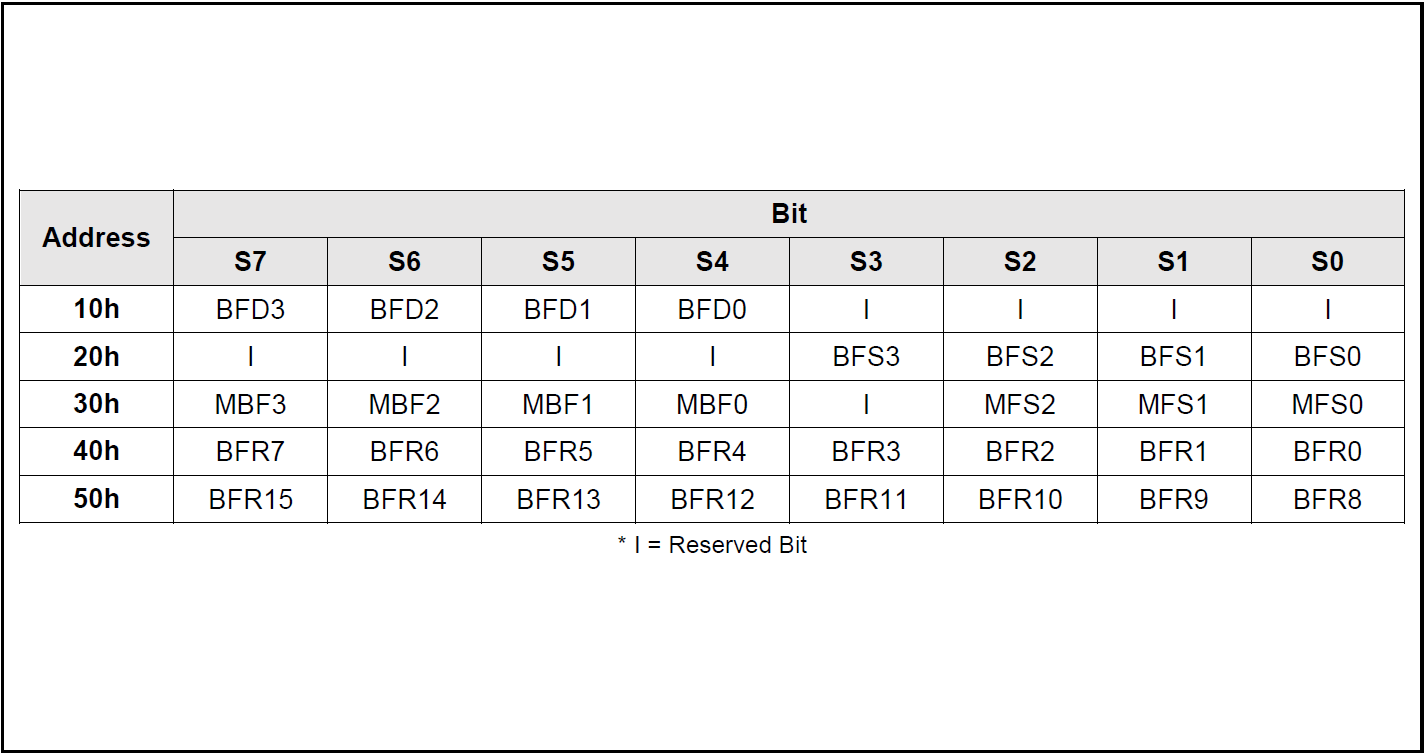
### 7.3.4 Write Enable Latch ( WEL ) - Status Only

* Read-only bit
* WEL = 1 ---> After Write Enable instruction.
* WEL = 0 ---> After power-up/Write Disable/Program/Erase/Page Data Read/Program Execute for OTP.

### 7.3.5 Erase/Program In Progress ( BUSY ) - Status Only

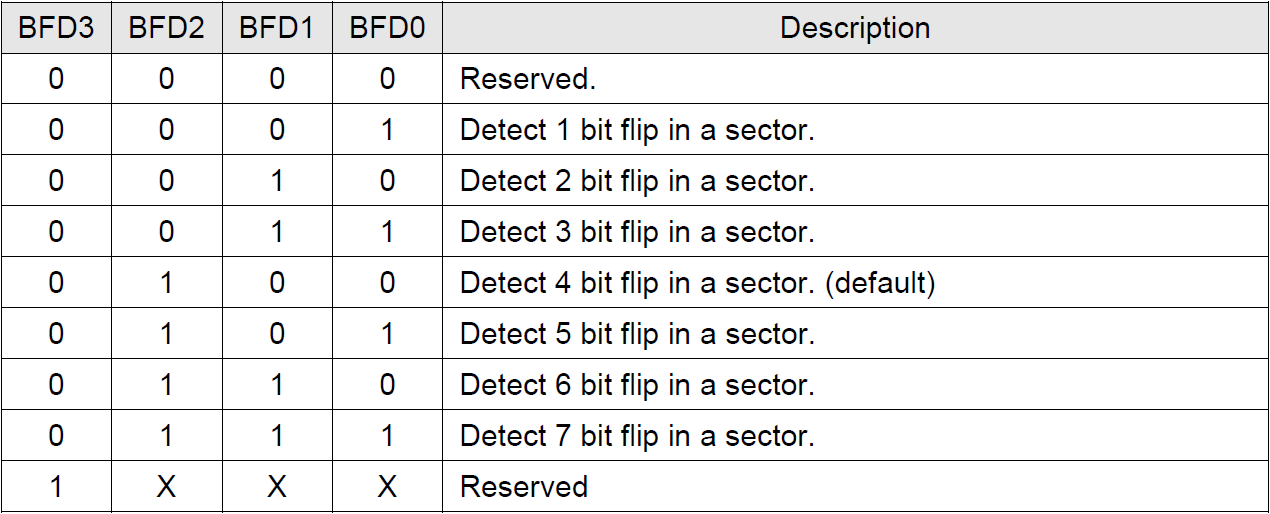
* BUSY = 1 ---> when device is powering up, executing Program/Erase commands.
* BUSY = 0 ---> operation is completed, meaning device ready.
* Device ignores new commands except Read Status Register / Read JEDEC ID.

### 7.4 Extended internal ECC feature registers



**Extended Internal ECC feature registers**

### 7.4.1 ECC Bit Flip Count Detection(BFD) - Volatile Writable



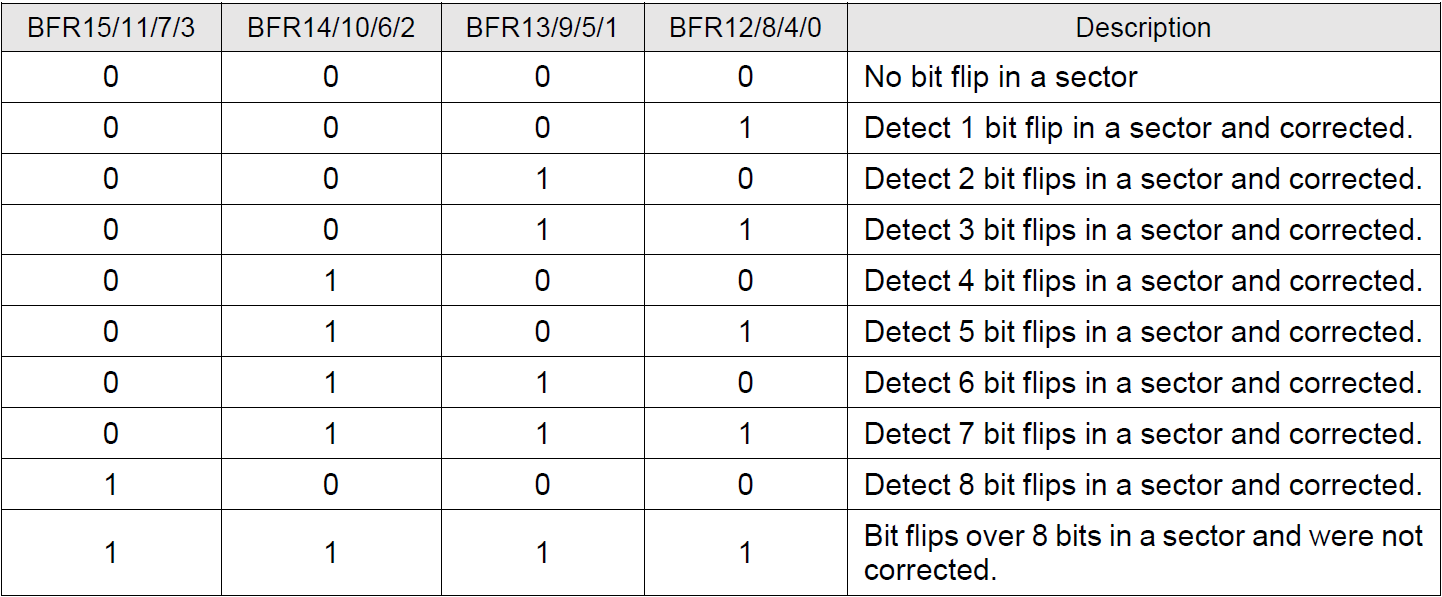
* BFD purpose:Sets error threshold (1~7 bit flips per sector).
* Default:Threshold = 4 bit flip.
* Operation:After Page Data Read with ECC enabled, ECC checks actual bit flips.
* Comparison:If bit flips ≥ threshold, status flags (BFS) are set.
* FW usage:
  + Monitor memory health.
  + Detect weak sectors early.
  + Trigger data migration when threshold exceeded.
  + Mark block as bad if errors are uncorrectable.

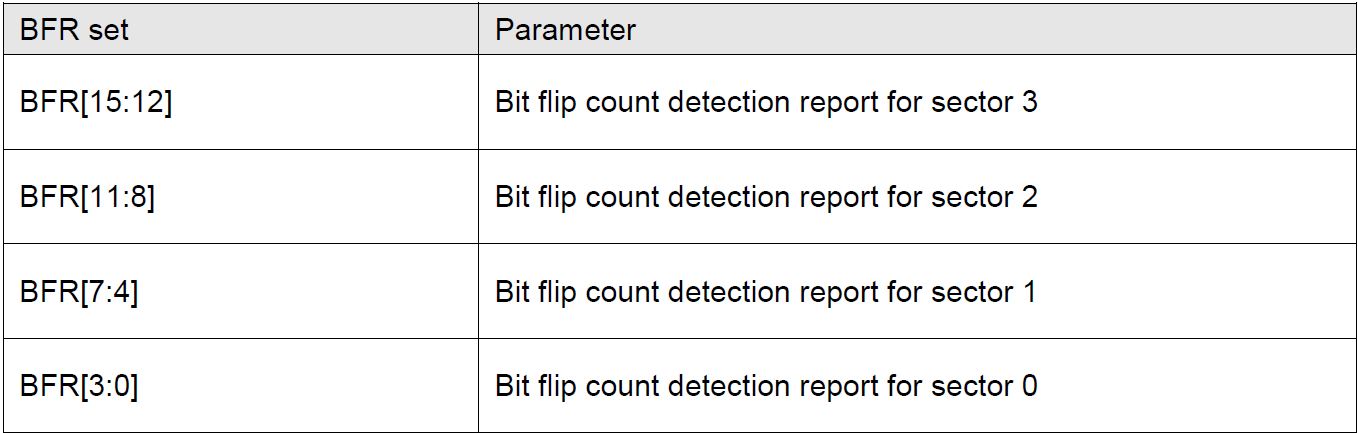
### 7.4.2 ECC Bit Flip Count Detection Status(BFS) - Status Only



* Checks each sector against BFD threshold.
* Reports binary result:
  + 1 = error count >= threshold
  + 0 = error count < threshold
* Sector mapping:
  + BFS3 ---> sector 3
  + BFS3 ---> sector 2
  + BFS3 ---> sector 1
  + BFS3 ---> sector 0
* FW usage:
  + Fast health check of sectors
  + Quick filtering before deeper analysis with BFR

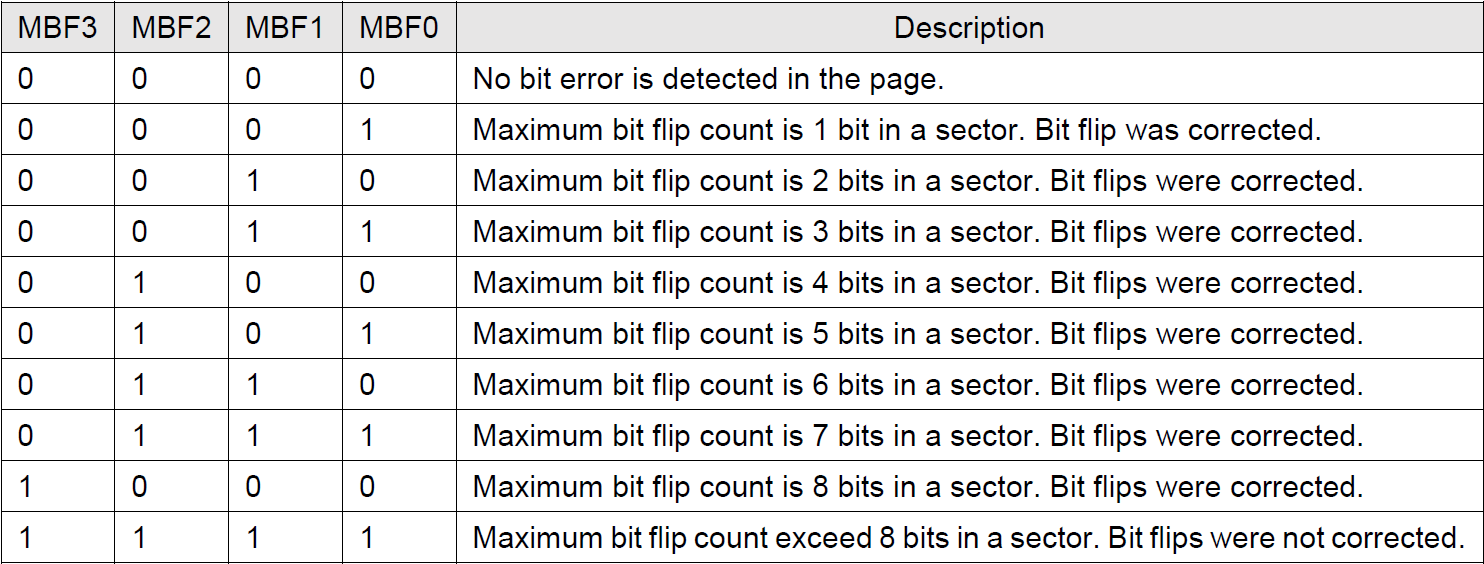
### 7.4.3 ECC Bit Flip Count Report(BFR) - Status Only

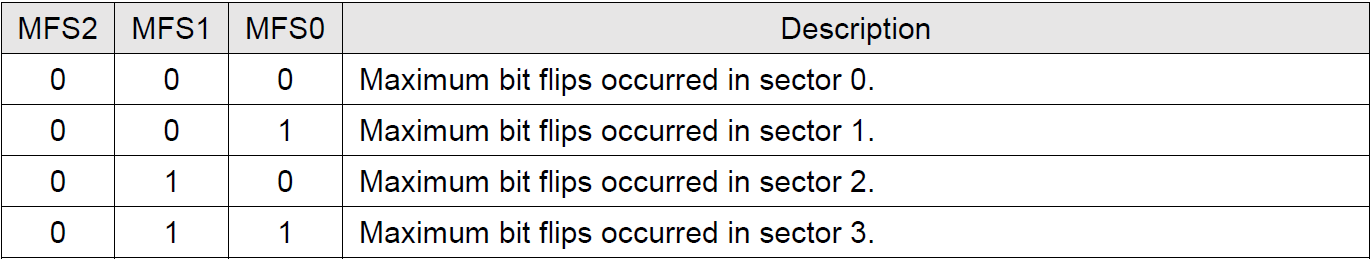




* Reports the exact bit flip count per sector.
* Encoding:
  + 0000 ---> no error
  + 0001 ~ 1000 ---> 1 ~ 8 errors (all corrected by ECC).
  + 1111 ---> 8 errors (uncorrectable).
* Sector mapping:
  + BFR[3:0] ---> sector 0
  + BFR[7:4] ---> sector 1
  + BFR[11:8] ---> sector 2
  + BFR[15:12] ---> sector 3
* FW usage:
  + Detailed error count for reliability analysis
  + <= 4 error ---> sector still usable, monitor closely
  + 5 ~ 8 errors ---> ECC corrected, but migration recommended
  + 8 errors ---> uncorrectable, mark block as Bad Block.

### 7.4.4 ECC Maximum Bit Flip Count Report (MBF, MFS) - Status Only



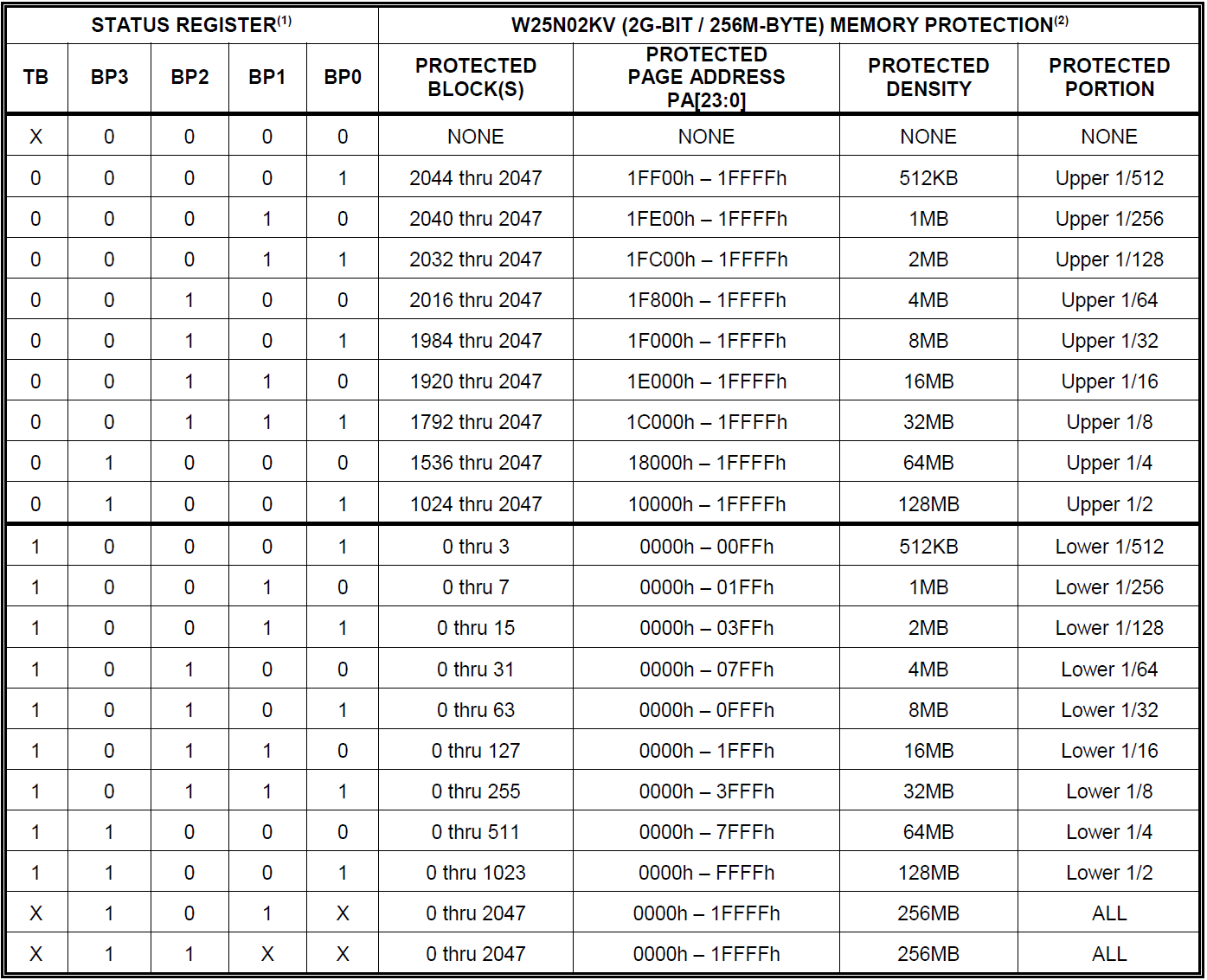


* MBF (Maximum Bit Flip Count)
* Reports the maximum bit flip count in the page.
* Range:
  + 0 ---> no errors
  + 1 ~ 8 ---> errors corrected by ECC
  + 8 ---> uncorrectable
* MFS (Sector Number of Max Error)
* Reports the sector number where the maximum bit flips occurred.
* Encoding:
  + 000 ---> sector 0
  + 001 ---> sector 1
  + 010 ---> sector 2
  + 011 ---> sector 3
* FW usage:
  + Quickly identify the worst sector in the page.
  + Use for reliability evaluation.
  + If MBF > 8 ---> mark block as unreliable (candidate for Bad Block)

### 7.4.5 Reserved Bits – Non Functional

* Reserved bits = not functional, no impact on device.
* Readout may be 0 or 1, should be ignored by FW.
* When writing, set reserved bits = 0.
* Writing has no effect on device behavior.

### 7.4.6 W25N02KV Status Register Memory Protection



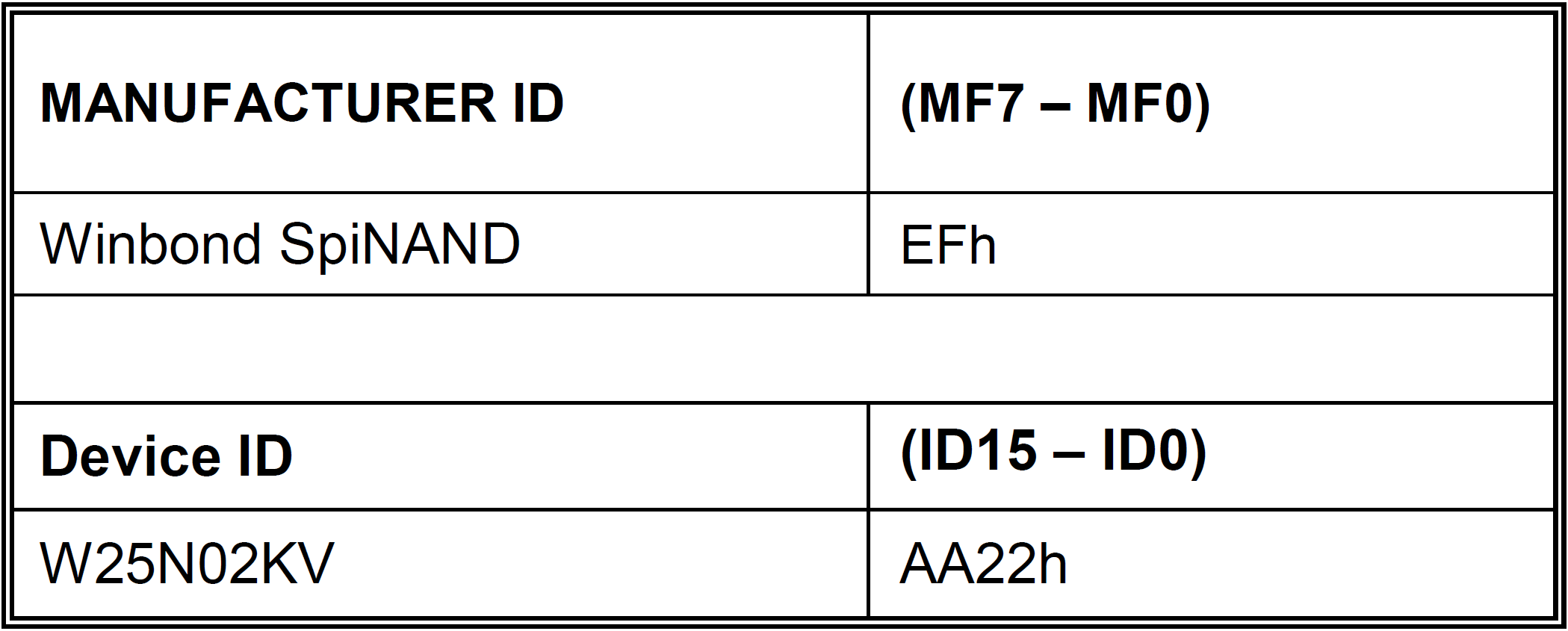
* BP3 ~ BP0 (Block Protect bits)
  + Define protected size : 512KB ---> 256MB
  + More BP bits set = larger protected region
* TB (Top/Bottom bit)
  + TB = 0 ---> protect upper memory blocks
  + TB = 1 ---> protect lower memory blocks
* Protected Portion
  + 1/512, 1/256, 1/128, …, 1/2, ALL.
  + Example:
  + BP0 = 1, others = 0 ---> 512KB protected
  + BP3 ~ BP0 = 1111 ---> all 256MB protected.
* FW usage:
  + Protect bootloader or metadata from accidental erase/program.
  + Selectively lock top or bottom portion depending on system design.
  + Increase BP protection bits as system enters deployment mode for higher safety.

## 8. INSTRUCTIONS

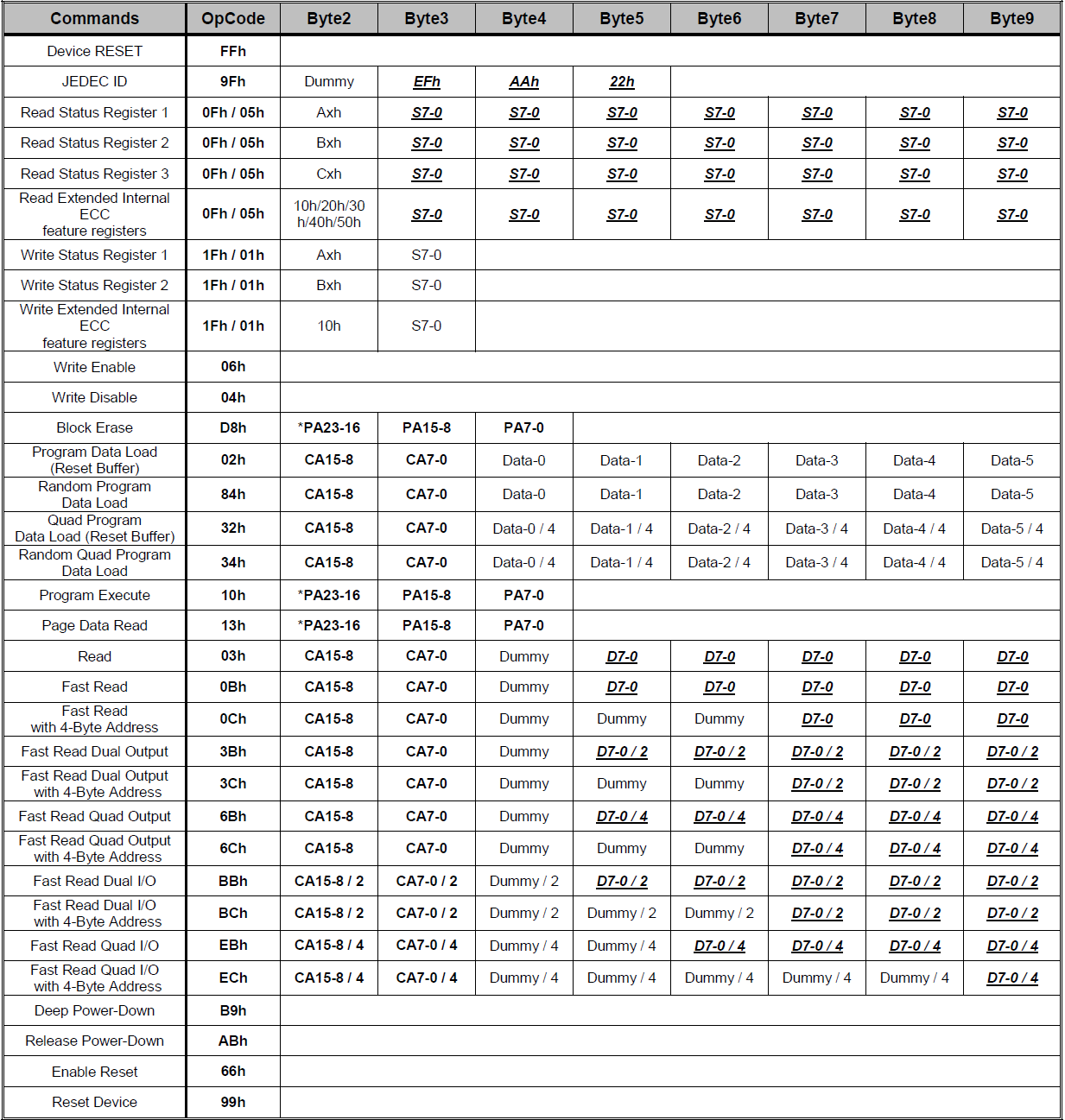
* Instruction System:
  + 31 SPI instructions supported.
  + Start : /CS falling edge ---> instruction code.
  + End : /CS rising edge.
* Transfer Rules:
  + May include address, data, dummy bytes.
  + Read ---> can end at any clock bit.
  + Write / Erase ---> must finish on byte boundaries(8-bit).
* Busy Handling:
  + During Program/Erase/Page Read/OTP ---> BUSY = 1.
  + Only Read Status Register & Read JEDEC ID allowed.
* Device Identification:
  + Manufacturer ID = EFh (Winbond SpiNAND).
  + Device ID = AA22h (W25N02KV).

### 8.1 Device ID and Instruction Set Tables

### 8.1.1 Manufacturer and Device Identification

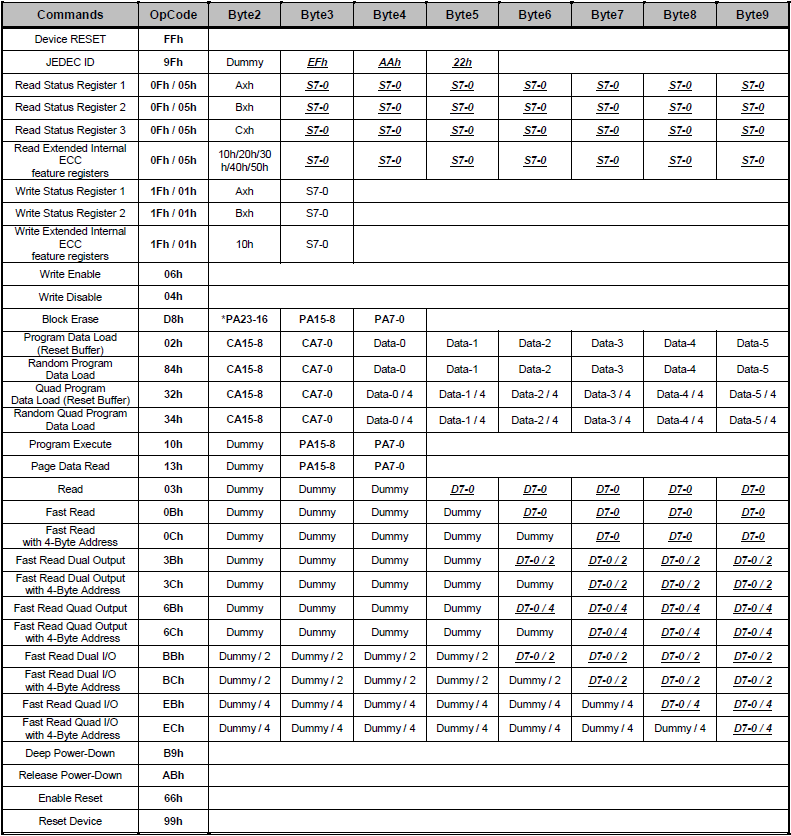


### 8.1.2 Instruction Set Table 1 (Buffer Read, BUF = 1, default)



\*Note : PA[23:17] input would be ignored.

**8.1.2 Instruction Set Table 2 (Sequential Read, BUF = 0, ECC-E = 0)**

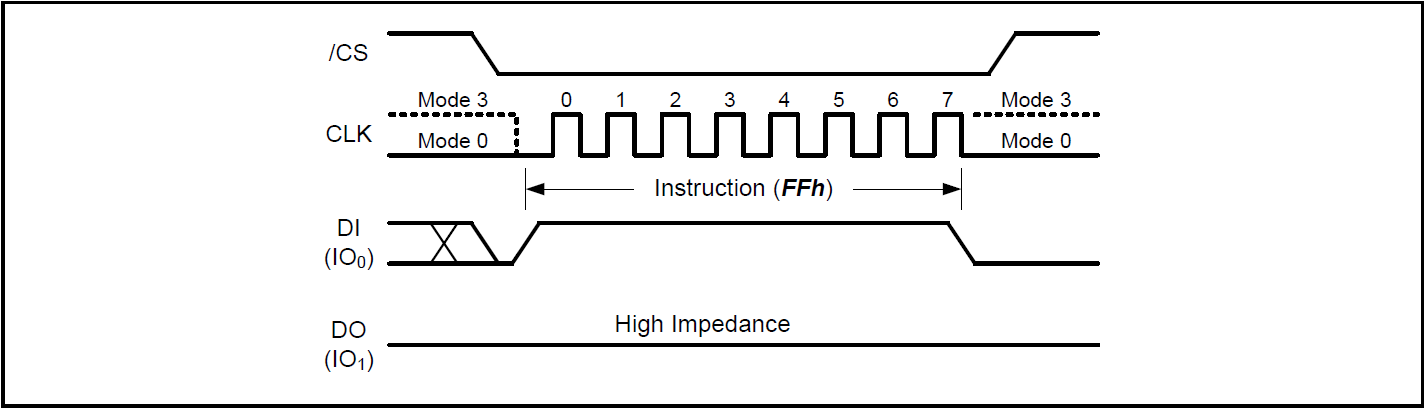


Note :

* Addressing
  + CA = 12 bits ( byte offset in page ).
  + PA = 17 bits ( block + page selection ).
* Registers : SR1 = Axh, SR2 = Bxh, SR3 = Cxh.
* SPI Formats : Dual/Quad mapping for IO0–IO3.
* Protection : WP-E = 1 ---> disable Quad commands.
* Read behavior :
  + Normal read: buffer retained after /CS high.
  + Sequential read: buffer cleared, BUSY = 1, need reload.

### 8.2 Instruction Descriptions

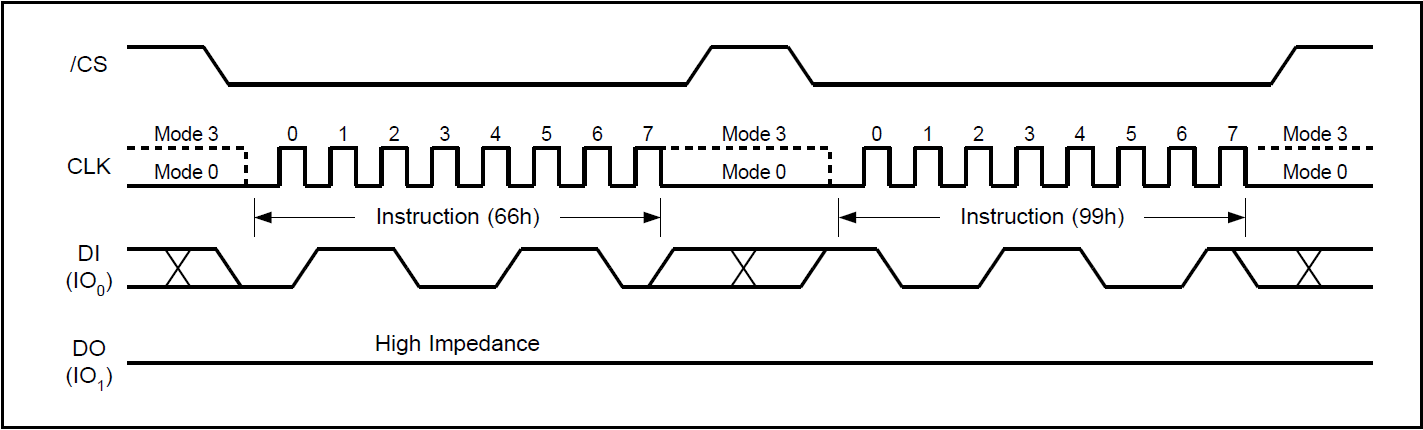
### 8.2.1 Device Reset (FFh), Enable Reset (66h) and Reset Device (99h)



Device Reset Instruction

Device Reset (FFh)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pules ( device expects 8 cycles for the 8-bit command ).
* [DI(IO0)] : Send FFh Command.
* [DO(IO1)] : stays High Impedance ( No data is driven out during the instruction ).
* [/CS] High : Instruction End.



Enable Reset and Reset Instruction Sequence

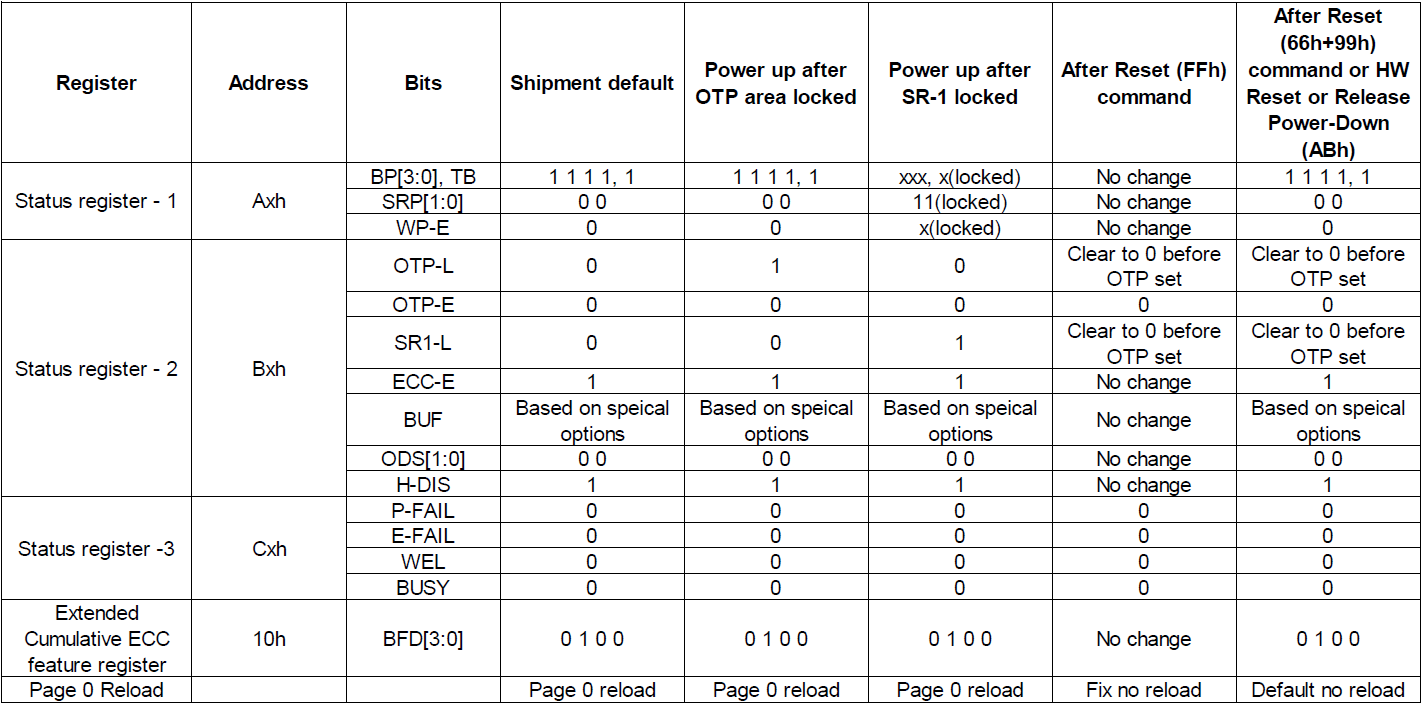
Enable Reset + Reset Device (66h + 99h)

Enable Reset (66h)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pulses ( device expects 8 cycles for the 8-bit command ).
* [DI(IO₀)] : Send 66h Command.
* [DO(IO₁)] : stays High Impedance ( No data is driven out during the instruction ).
* [/CS] High : Instruction End.

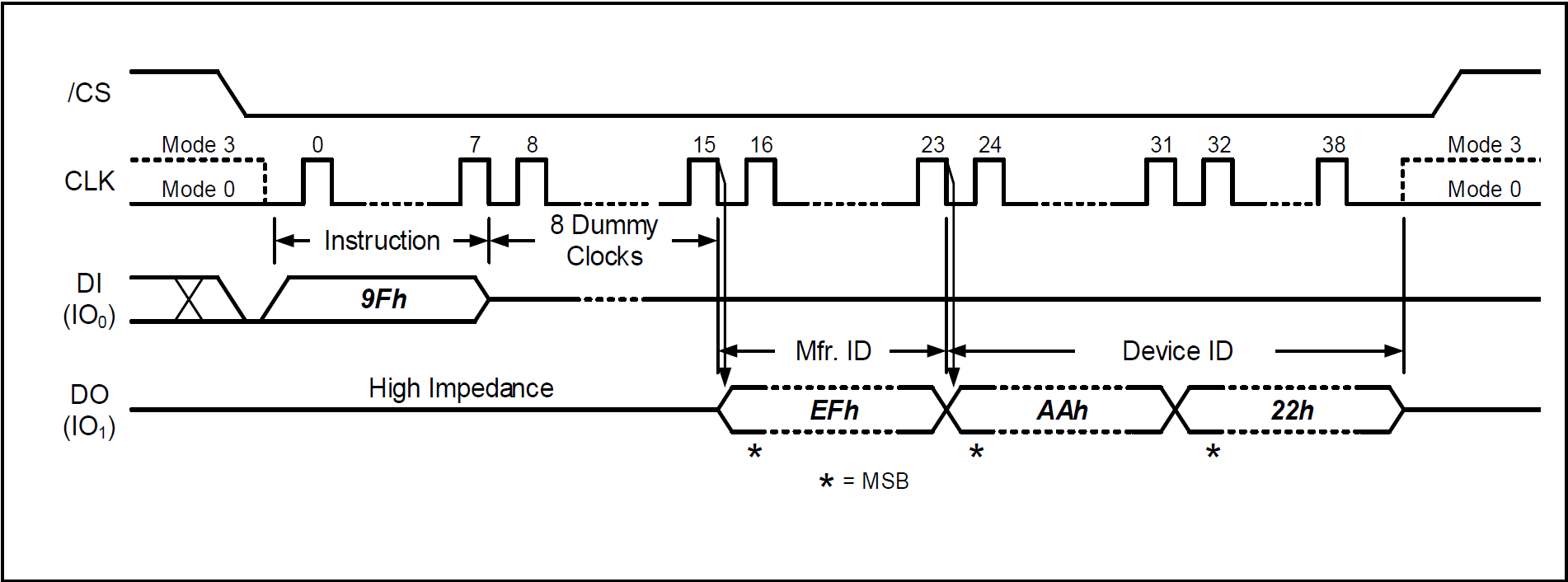
Reset Device (99h)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pulses ( device expects 8 cycles for the 8-bit command ).
* [DI(IO₀)] : Send 99h Command.
* [DO(IO₁)] : stays High Impedance ( No data is driven out during the instruction ).
* [/CS] High : Instruction End.



Default values of the Status Registers after power up and Device Reset

### 8.2.2 Read JEDEC ID (9Fh)



Read JEDEC ID Instruction

* JEDEC ID command (9Fh) is standard for SPI-compatible serial memories.
* Requires sending 8 dummy clocks after command.
* Output sequence:
* Manufacturer ID (1 byte, EFh for Winbond).

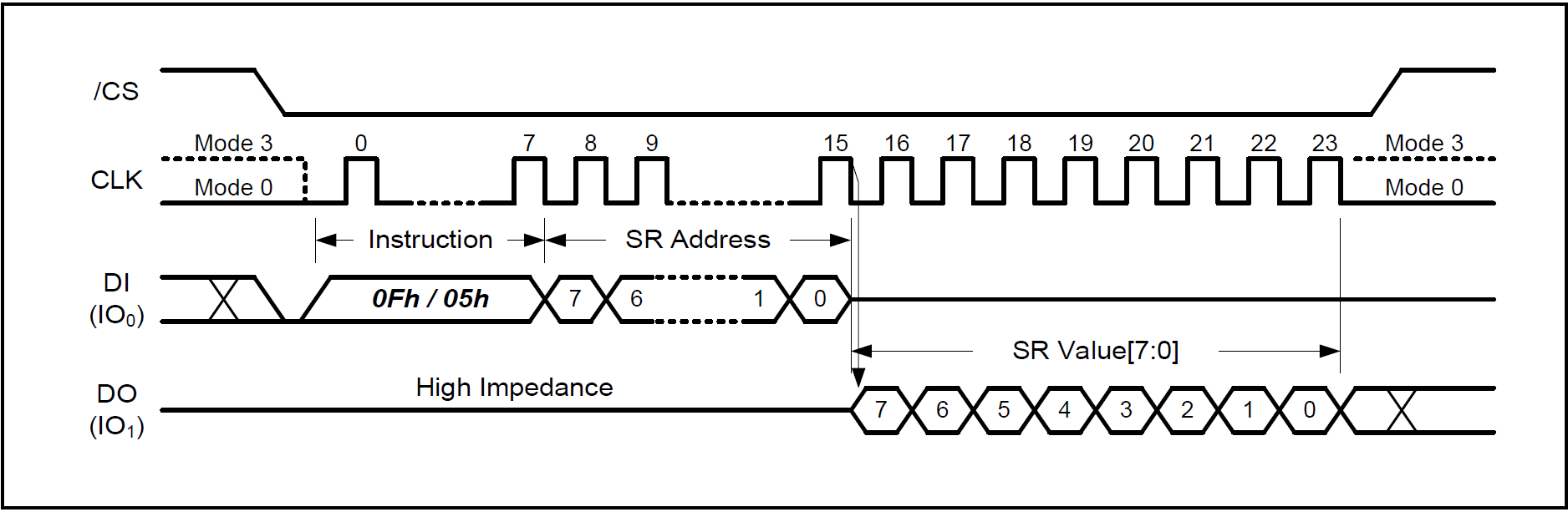
1. Device ID (2 bytes, e.g., AAh, 22h).
2. Data shifted out MSB first on falling edge of CLK.

* Used by FW for device detection, compatibility check, and capacity verification.

Read JEDEC ID (9Fh)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pulses ( device expects 8 cycles for the 8-bit command ).
* [DI(IO₀)] : Send 9Fh Command.
* [DO(IO₁)] :
  + High Impedance during instruction and dummy clocks.
  + After 8 dummy clocks, device outputs Manufacturer ID (EFh for Winbond).
  + followed by 2-byte Device ID (e.g., AAh, 22h).
  + Data is shifted out on falling edge of CLK, MSB first.
* [/CS] High : Instruction End.

### 8.2.3 Read Status Register (0Fh / 05h)



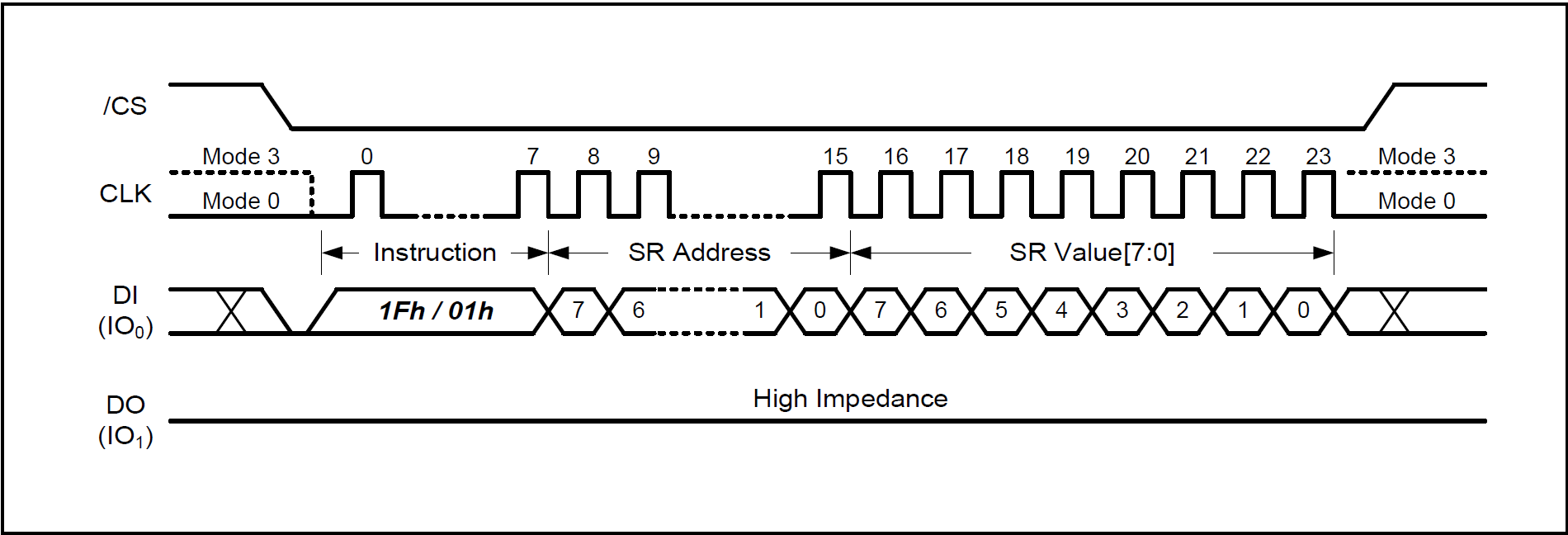
Read Status Register Instruction

* Instructions :
  + 05h (Read Status Register-1)
  + 0Fh (General Read Status Register)
* Requires :
  + Command (05h or 0Fh)
  + 8-bit Status Register Address (Axh, Bxh, Cxh, 10h)
* Output : 8-bit SR value [7:0], MSB first.
* Can be used any time, even during Program, Erase, or Page Read cycles.
* FW usage :
  + Poll BUSY bit to check if device is ready.
  + Read WEL、P-FAIL、E-FAIL、ECC-E、etc.
* Continuous read supported ---> FW can keep clocking for repeated SR values.

Read Status Register (0Fh / 05h)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pulses for the instruction code + 8 pulses for the register address.
* [DI(IO₀)] :
  + Send 0Fh or 05h Command
  + followed by 1-byte Status Register Address.
* [DO(IO₁)] :
  + High Impedance during command and address input.
  + Device outputs the 8-bit Status Register value [7:0], MSB first, on the falling edge of CLK.
  + Output can continue (continuous read) if /CS remains low.
* [/CS] High : Instruction End.

### 8.2.4 Write Status Register (1Fh / 01h)



Write Status Register – 1 / 2 / 3 Instruction

* Purpose: The Write Status Register instruction allows writing specific bits in the Status Registers.
* Writable bits:
  + Status Register – 1 : SRP[1:0]、TB, BP[3:0]、WP-E
  + Status Register – 2 : OTP-L、OTP-E、SR1-L、ECC-E
* Other bits: Read-only, not affected by this instruction.
* Command code: 1Fh (General) or 01h (SR-1).
* Sequence:

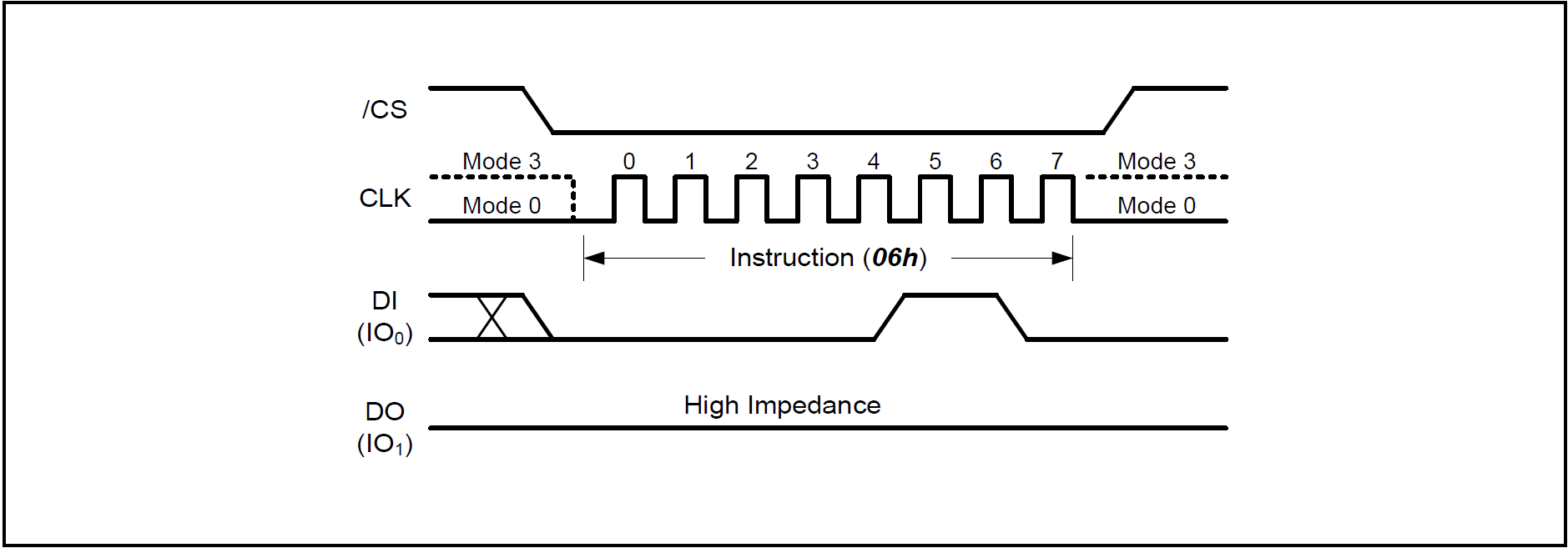
1. Drive /CS low
2. Send instruction code (1Fh/01h)
3. Send 8-bit Status Register Address
4. Send 8-bit Status Register Data (SR Value[7:0])

* Defaults after power-up:
  + BP[3:0] = 1, TB = 1
  + ECC-E = 1, All other writable bits = 0

Write Status Register (1Fh / 01h)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pulses for command, + 8 pulses for SR Address, + 8 pulses for SR Data.
* [DI(IO₀)] :
  + Send 1Fh or 01h Command.
  + Send 1-byte Status Register Address.
  + Send 1-byte Status Register Data (SR Value[7:0]).
* [DO(IO₁)] : High Impedance (no data output).
* [/CS] High : Instruction End.

### 8.2.5 Write Enable (06h)



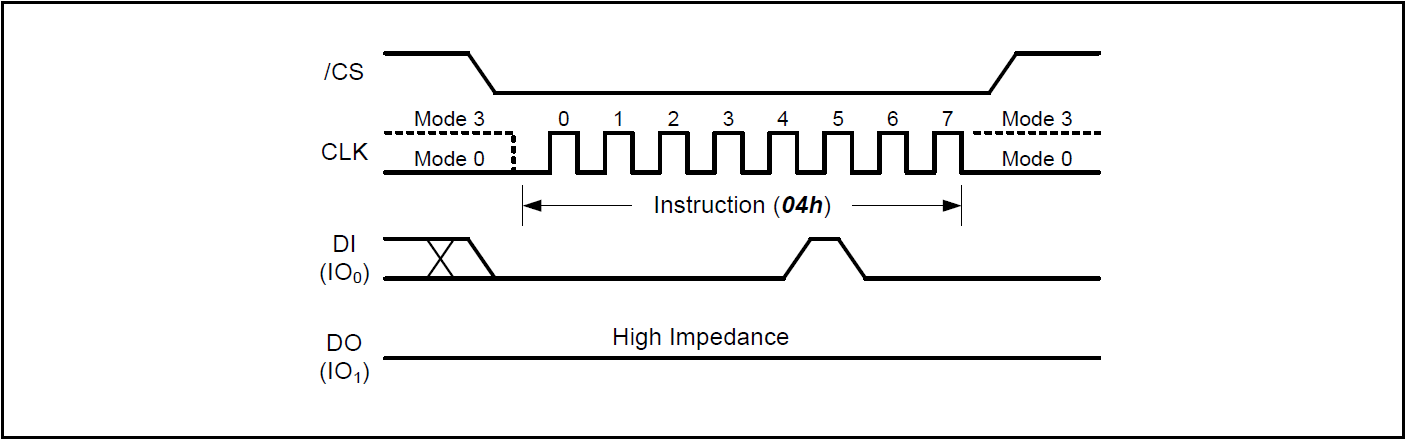
Write Enable Instruction

* Purpose : Sets the Write Enable Latch (WEL) bit in the Status Register to 1.
* Requirement :
* Must be set before every :
  + Load Program Data (02h / 84h / 32h / 34h)
  + Program Execute
  + Block Erase instruction

Write Enable (06h)

* [/CS] Low : Instruction Start.
* [CLK] : Provides 8 clock pulses (8-bit command).
* [DI(IO₀)] : Send 06h Command.
* [DO(IO₁)] : Stays High Impedance (no data output).
* [/CS] High : Instruction End.

### 8.2.6 Write Disable (04h)



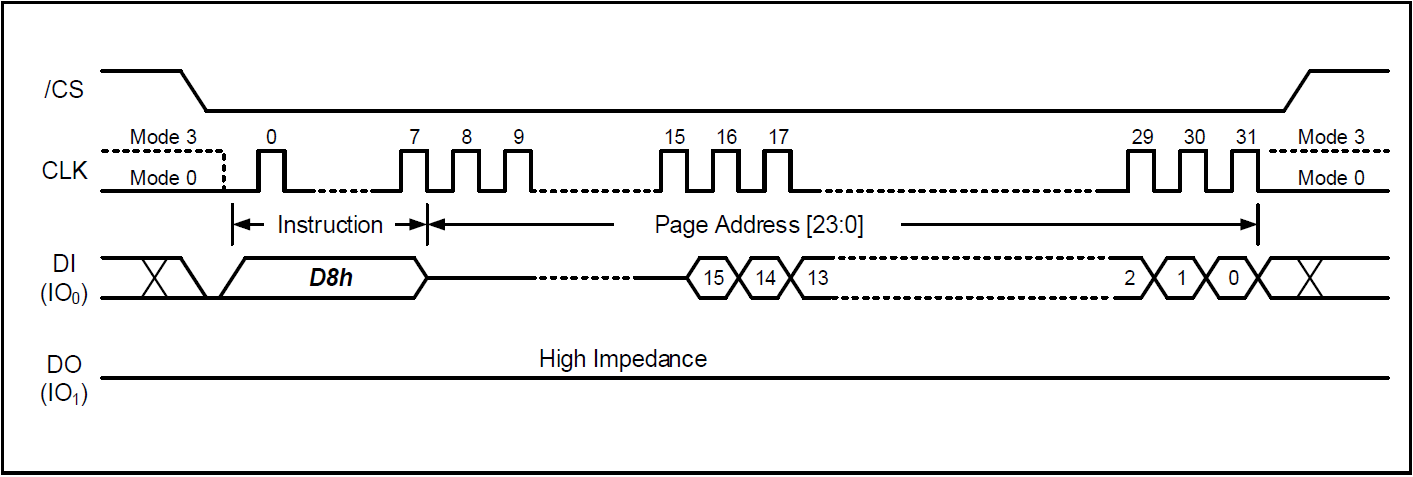
Write Disable Instruction

* Purpose: Resets the Write Enable Latch (WEL) bit in the Status Register to 0.
* Auto reset behavior:
* WEL is automatically cleared after:
  + Power-up
  + Completion of Page Data Read
  + Program Execute
  + Block Erase
  + Reset instructions
* Generally, there's no need to explicitly send 04h, as WEL is automatically cleared after most operations complete.
* However, in a multi-operation sequence, if writes need to be disabled, the firmware can proactively send 04h to prevent accidental writes.

Write Disable (04h)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 clock pulses (8-bit command).
* [DI(IO₀)] : Send 04h Command.
* [DO(IO₁)] : stays High Impedance (no data output).
* [/CS] High : Instruction End.

### 8.2.7 128KB Block Erase (D8h)



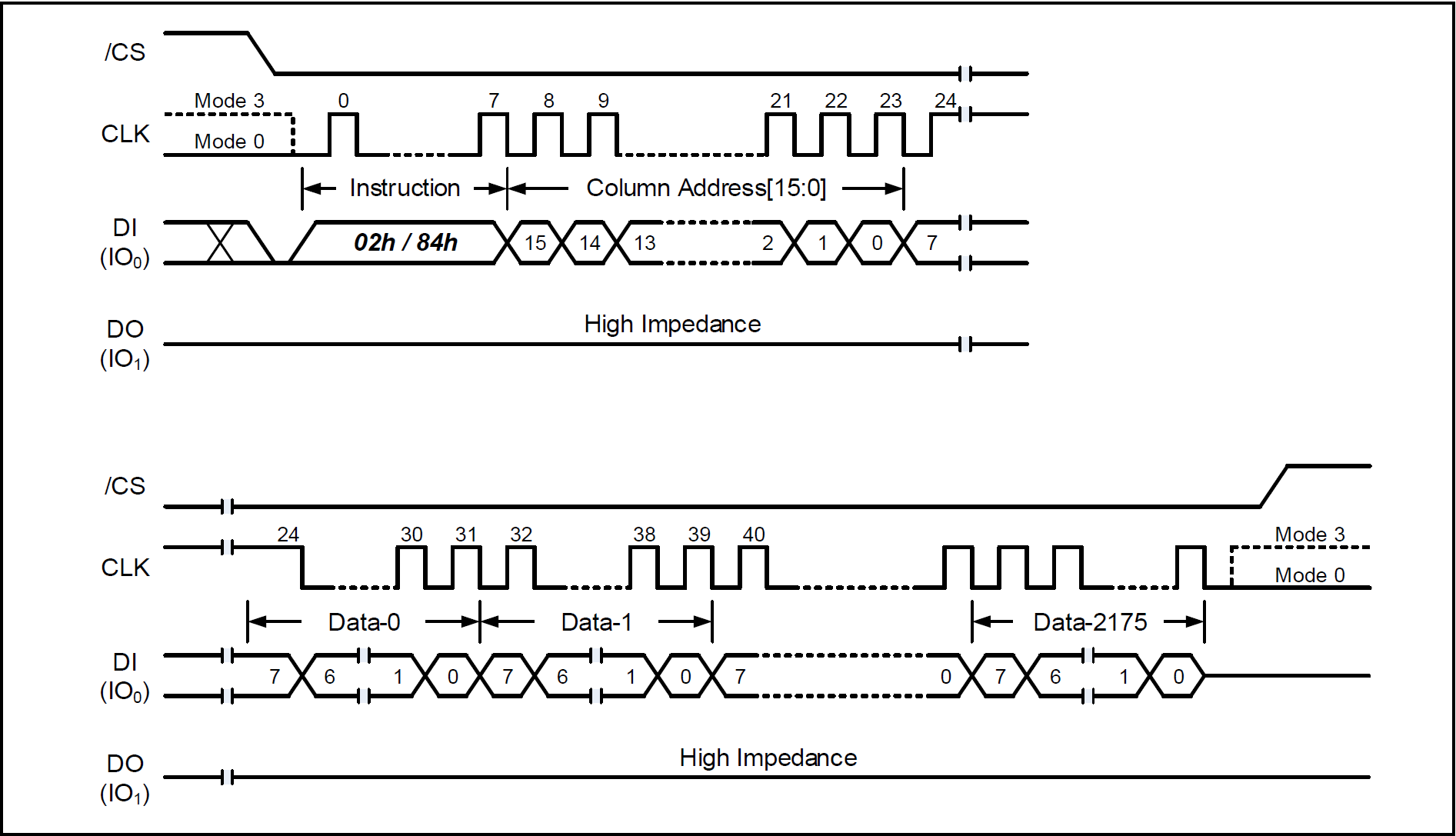
128KB Block Erase Instruction

* Purpose : Erases all memory within a specified block (64 pages = 128KB) to erased state of all 1s (FFh).
* Requirement :
* Must issue Write Enable (06h) first (WEL bit = 1).
* Instruction ignored if block is protected (TB, BP3–BP0 bits).
* Execution :
* Self - timed erase begins after /CS high.
* BUSY bit = 1 during erase, cleared to 0 when done.
* After completion, WEL is automatically cleared to 0.
* FW Notes:
  + During an erase, you can use the Read Status Register (05h/0Fh) to poll BUSY.
  + If /CS is not released correctly after entering the address, the instruction will be invalid.
  + After each Block Erase, must re-execute the Write Enable (06h) command before the next program or erase operation can be performed.
  + Must poll BUSY bit in Status Register until erase is done.
  + Erase will not start if /CS not released properly after last byte.

128KB Block Erase (D8h)

* [/CS] Low : Instruction Start.
* [CLK] : provides 8 cycles for command + 24 cycles for address.
* [DI(IO₀)] :
  + Send D8h Command.
  + Send 24-bit Block Address (A23–A0, any address within block).
* [DO(IO₁)] : stays High Impedance.
* [/CS] High : Instruction End, erase operation begins.

### 8.2.8 Load Program Data (02h) / Random Load Program Data (84h)



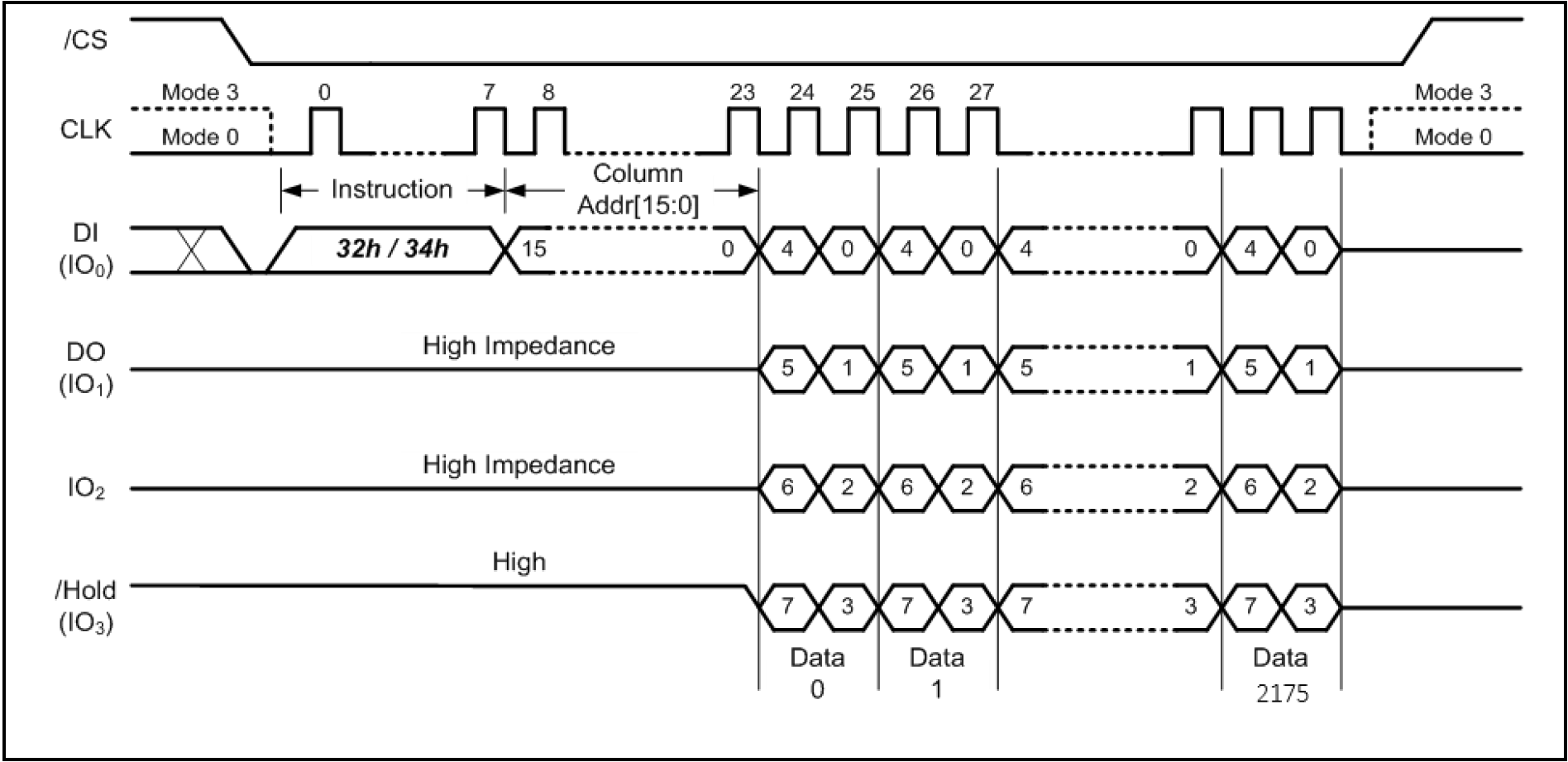
Load / Random Load Program Data Instruction

* (02h) Loads data bytes into Data Buffer, Unused buffer locations are reset to FFh.
* (84h) Updates only specified locations in Data Buffer, Other buffer contents remain unchanged.
* Precondition : Must Write Enable (06h) before this instruction (WEL = 1).
* Max 2,176 bytes per page (main array) / Extra 64 bytes reserved for ECC.
* If ECC-E=1 (enabled) : All 2,176 bytes accepted / ECC parity (64B) will be overwritten by ECC calculation.
* If ECC-E=0 (disabled) : Extra 64B can be used for external ECC or other usage.

Load Program Data (02h) / Random Load Program Data (84h)

* [/CS] Low : Instruction Start
* [CLK] : Provides continuous clock cycles (8 cycles for opcode, 16 cycles for column address, variable cycles for data).
* [DI(IO0)] :
  + Send 02h or 84h Command.
  + Send 16 bit Column Address (CA[15:0]).
  + Send 1 ~ N Data Bytes (up to 2176 Bytes).
* [DO(IO1)] : Stays High Impedance(No data output)
* [/CS] High : Intsruction End.

### 8.2.9 Quad Load Program Data (32h) / Quad Random Load Program Data (34h)



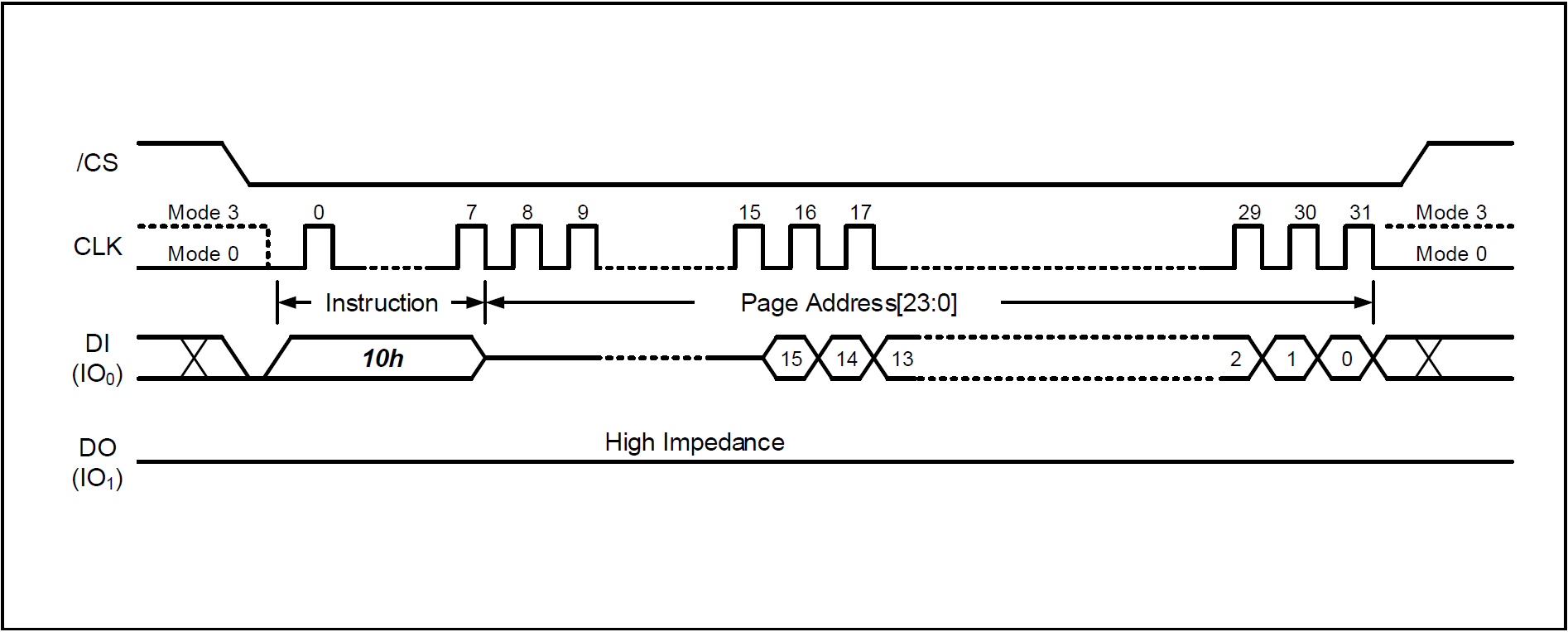
Quad Load / Quad Random Load Program Data Instruction

* (32h) Loads data bytes into Data Buffer, Unused Data Buffer bytes are reset to FFh.
* (34h) Updates only the specified Data Buffer bytes, Other Data Buffer contents remain unchanged.
* Must Write Enable (06h) before this instruction (WEL = 1).
* If WP-E bit = 1 in Status Register ---> all Quad SPI instructions are disabled.
* Max 2176 bytes per page (main array).
* Extra 64 bytes reserved for ECC (same as non-quad mode).
* Same as standard load (02h/84h).
* ECC-E = 1 ---> ECC section (64B) overwritten by ECC calculation.
* ECC-E = 0 ---> Extra 64B free for external ECC or other usage.

Quad Load Program Data (32h) / Quad Random Load Program Data (34h)

* [/CS] Low : Instruction Start.
* [CLK] : Provides continuous clock cycles (8 cycles for opcode, 16 cycles for column address, variable cycles for data).
* [DI(IO0)] : Send 32h or 34h Command.
* [DI(IO0 ~ IO3)] :
* Send 16-bit Column Address (CA[15:0]).
* Send 1 ~ N Data Bytes (quad input, up to 2,176 Bytes).
* [DO(IO1)] : Stays High Impedance (No data output).
* [/CS] High : Instruction End.

### 8.2.10 Program Execute (10h)



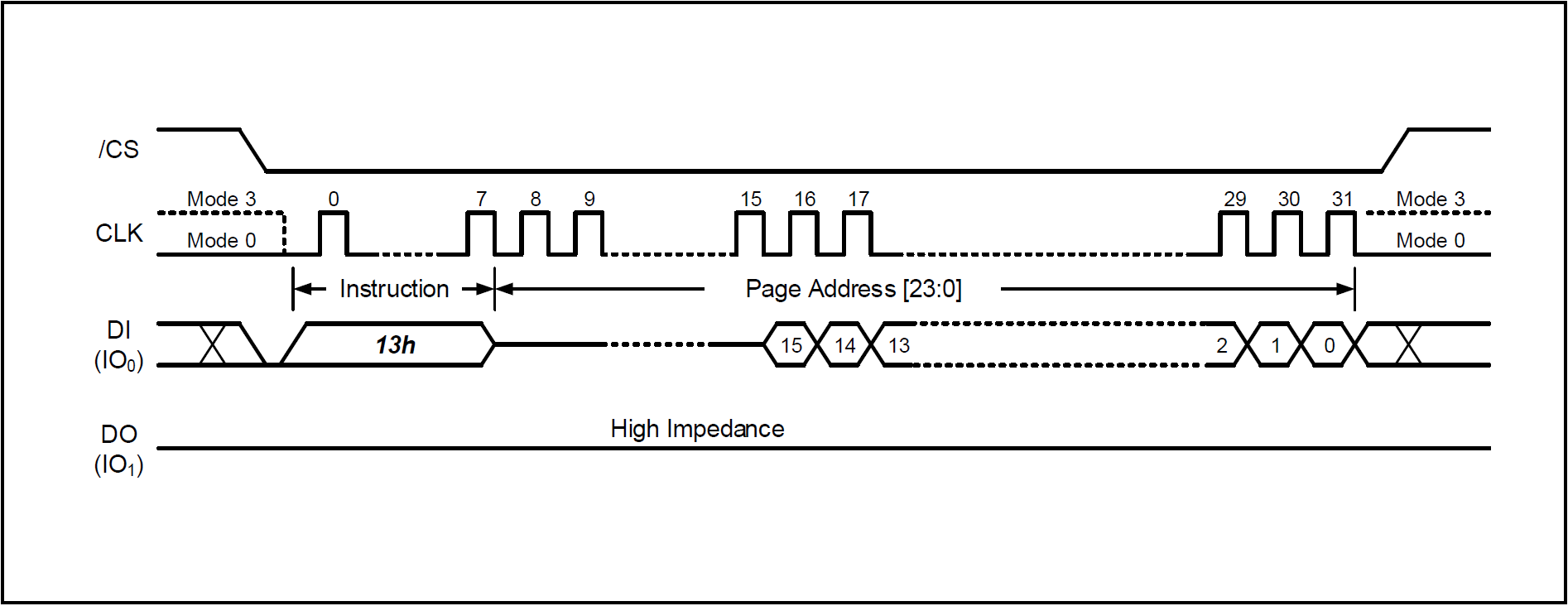
Program Execute Instruction

* Second step (after data loaded into Data Buffer using 02h/84h/32h/34h).
* Programs the contents of Data Buffer (2,176 bytes / 2,048 bytes if ECC enabled) into the physical NAND memory page.
* Pages in a block must be programmed sequentially from lower to higher addresses.
* Out-of-sequence programming is prohibited.
* Cannot cross plane boundaries (data from one plane cannot program another plane).
* Will not execute if the page is protected by Block Protect bits (TB, BP3, BP2, BP1, BP0).
* During program execution, device sets BUSY bit = 1.
* After program finishes, BUSY bit = 0.
* WEL (Write Enable Latch) bit is automatically cleared to 0 after execution.
* Must re-issue Write Enable (06h) before next program.
* Program execution is self-timed, duration = tPP (refer to AC characteristics).
* While busy, Read Status Register instruction may be used to poll BUSY bit.

Program Execute (10h)

* [/CS] Low : Instruction Start.
* [CLK] : Provides 8 cycles for opcode, 24 cycles for page address.
* [DI(IO0)] :
  + Send 10h Command.
  + Send 24-bit Page Address (PA[23:0]).
* [DO(IO1)] : Stays High Impedance (No data output).
* [/CS] High : Instruction End.

### 8.2.11 Page Data Read (13h)



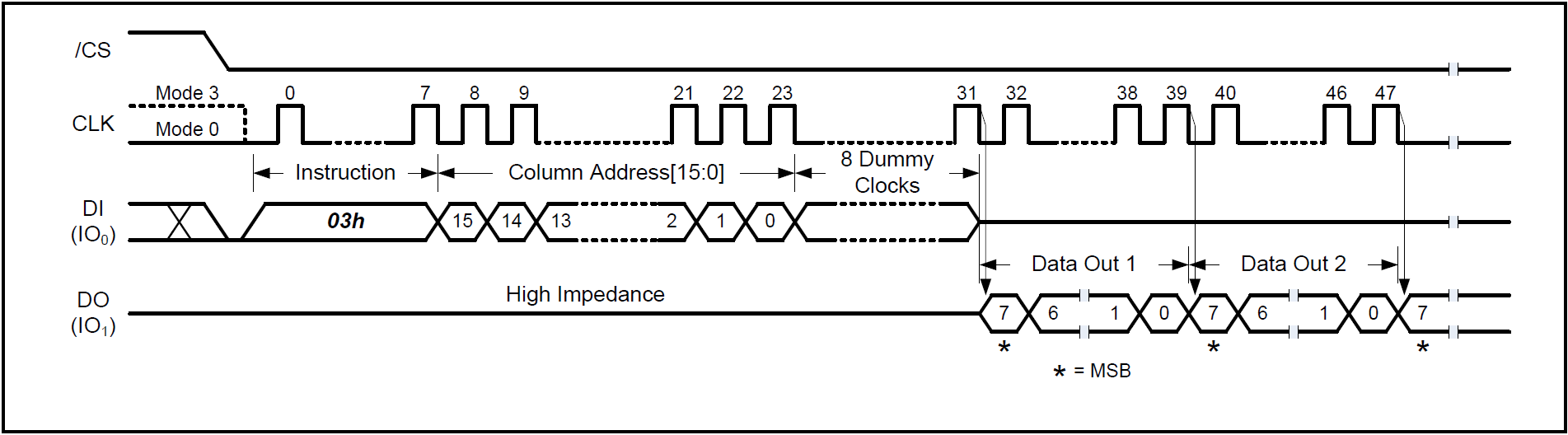
Page Data Read Instruction

* Transfers a full page (2,176 Bytes) from NAND memory array into the Data Buffer.
* After /CS goes High, the device enters a self-timed Page Data Read cycle.
* Status Register Behavior:
* BUSY bit = 1, During operation.
* BUSY bit = 0, After operation completes
* Poll Status Register to check readiness.
* Operation time defined as tRD
* Post-Operation:
* Once data is transferred into the Data Buffer, several Read instructions (e.g., 03h, 0Bh, 6Bh, EBh, etc.) can be used to output the data.

Page Data Read (13h)

* [/CS] Low : Instruction Start.
* [CLK] : Provides 8 cycles for opcode, 24 cycles for page address.
* [DI(IO0)] :
  + Send 13h Command.
  + Send 24-bit Page Address (PA[23:0]).
* [DO(IO1)] : Stays High Impedance (No data output).
* [/CS] High : Instruction End → Device begins tRD cycle.

### 8.2.12 Read Data (03h)



Read Data Instruction (Buffer Read Mode, BUF=1)



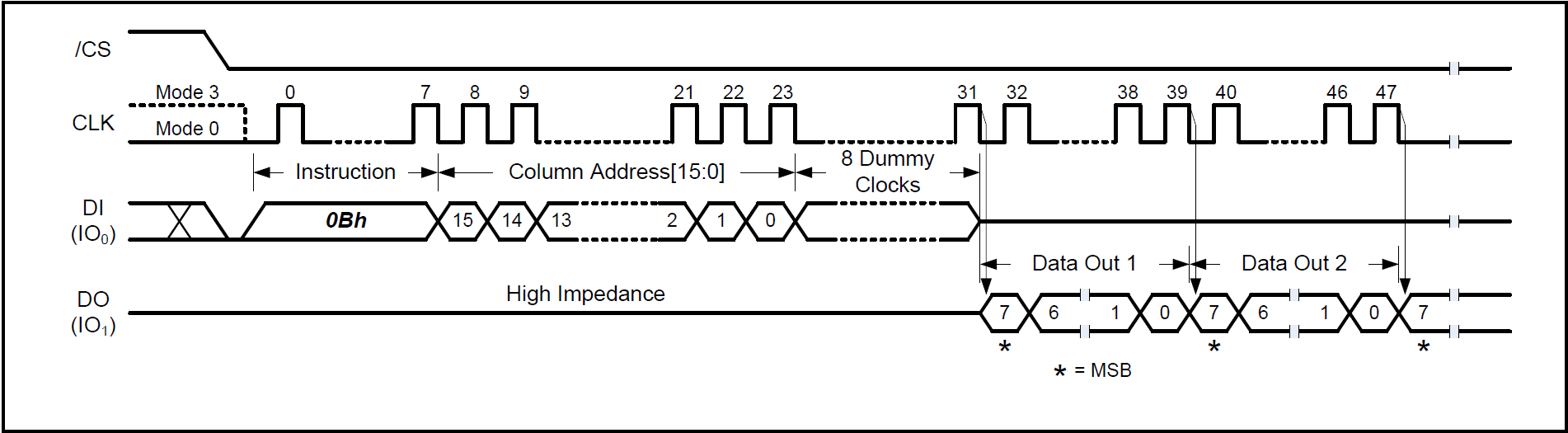
Read Data Instruction (Sequential Read Mode, BUF=0)

* Read Modes:
* Buffer Read Mode (BUF=1):
* Output starts at specified Data Buffer column address.
* Continues until end of Data Buffer (2,176 bytes).
* After last byte, DO goes High-Z.
* Sequential Read Mode (BUF=0):
* Output starts at first byte of Data Buffer.
* Auto-increments through Data Buffer.
* When Data Buffer ends, continues to next memory page
* allows continuous array read.
* Compatible with Winbond NOR SPI flash read sequence.

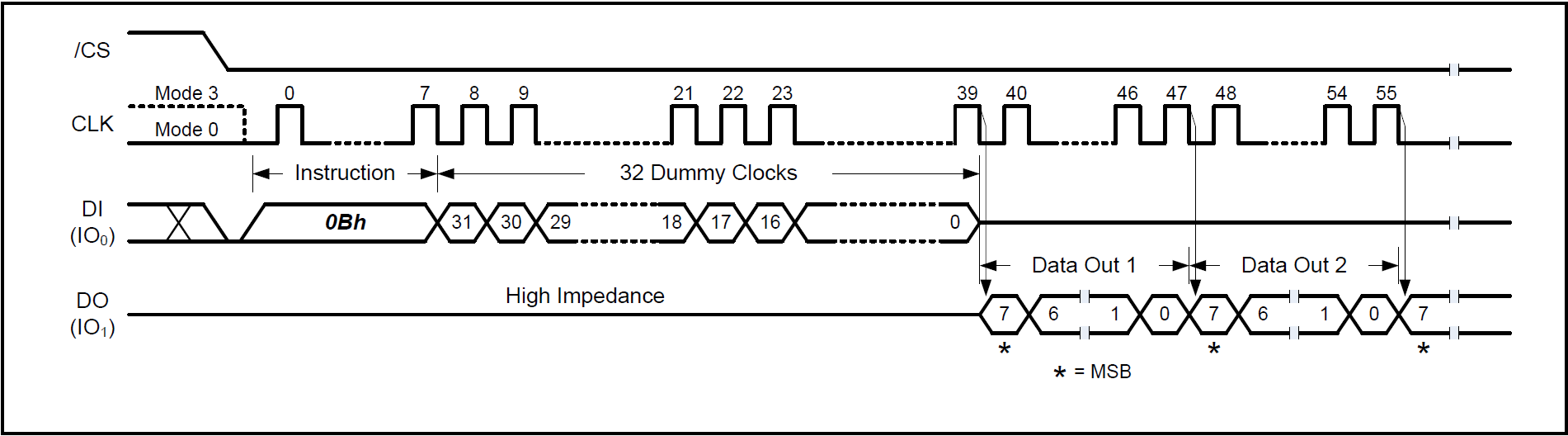
Read Data (03h)

* [/CS] Low : Instruction Start.
* [CLK] : 8 cycles for opcode | 16 cycles column address | Dummy cycles (8 or 24).
* [DI(IO0)] : Send 03h Command | Send 16-bit Column Address (CA[15:0]) | Send 8 dummy clocks (BUF=1) or 24 dummy clocks (BUF=0).
* [DO(IO1)] : Outputs Data sequentially, MSB first | Data auto-increments column address.
* [/CS] High : Instruction End.

### 8.2.13 Fast Read (0Bh)



Fast Read Instruction (Buffer Read Mode, BUF = 1)

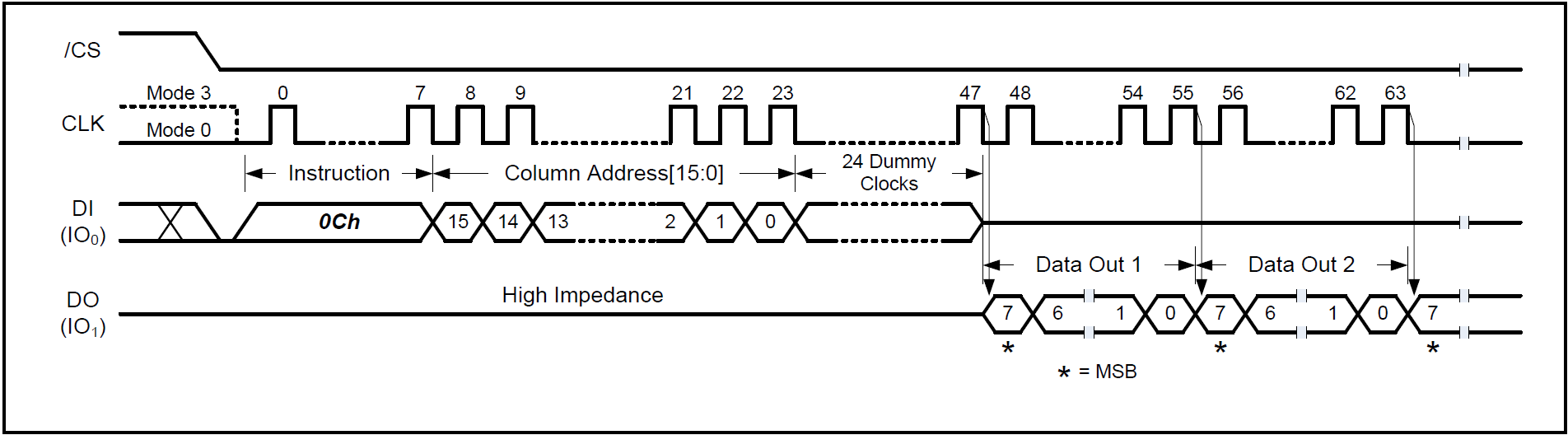


Fast Read Instruction (Sequential Read Mode, BUF = 0)

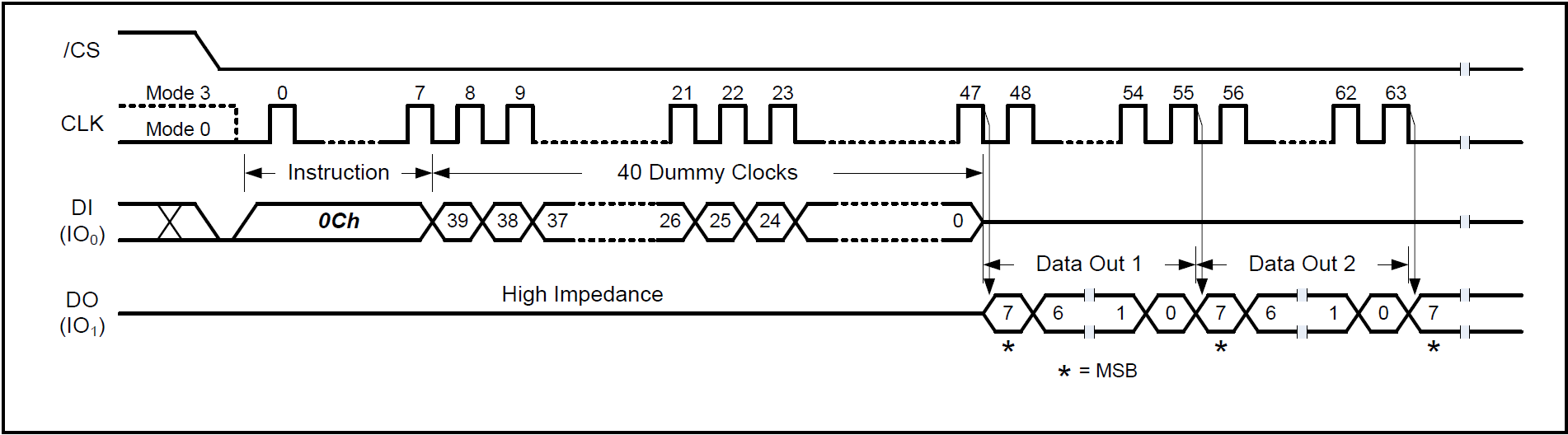
Fast Read (0Bh)

* [/CS] Low : Instruction Start.
* [CLK] :
* Provides 8 cycles for opcode.
* 16 cycles for column address.
* Dummy cycles (8 clocks if BUF = 1, 32 clocks if BUF = 0).
* [DI(IO0)] :
  + Send 0Bh Command.
  + Send 16-bit Column Address (CA[15:0]).
  + Send Dummy Clocks (8 or 32 depending on BUF).
* [DO(IO1)] :
  + Outputs Data sequentially, MSB first.
  + Data auto-increments column address.
* [/CS] High : Instruction End.
* Instruction Code : (0Bh) Fast Read
* Operation:
* Reads 1 or more bytes sequentially from the Data Buffer after executing Page Data Read (13h).
* Data output on DO(IO1), MSB first, at falling edge of CLK.
* Column address auto-increments as data is read out.
* Read Modes:
* Buffer Read Mode ( BUF = 1 )
  + Output starts from Data Buffer column address.
  + Continues until end of Data Buffer (2176 bytes).
  + After last byte, output pin goes High-Z.
  + Needs 8 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
  + Output starts from first byte of Data Buffer.
  + Continues through Data Buffer, then automatically moves to next page.
  + Enables continuous memory array read.
  + Compatible with Winbond NOR SPI flash sequence.
  + Needs 32 dummy clocks before data output.
* I/O Behavior :
* DI(IO0) : opcode + address input.
* DO(IO1) : data output.

### 8.2.14 Fast Read with 4-Byte Address (0Ch)



Fast Read with 4-Byte Address Instruction (Buffer Read Mode, BUF = 1)

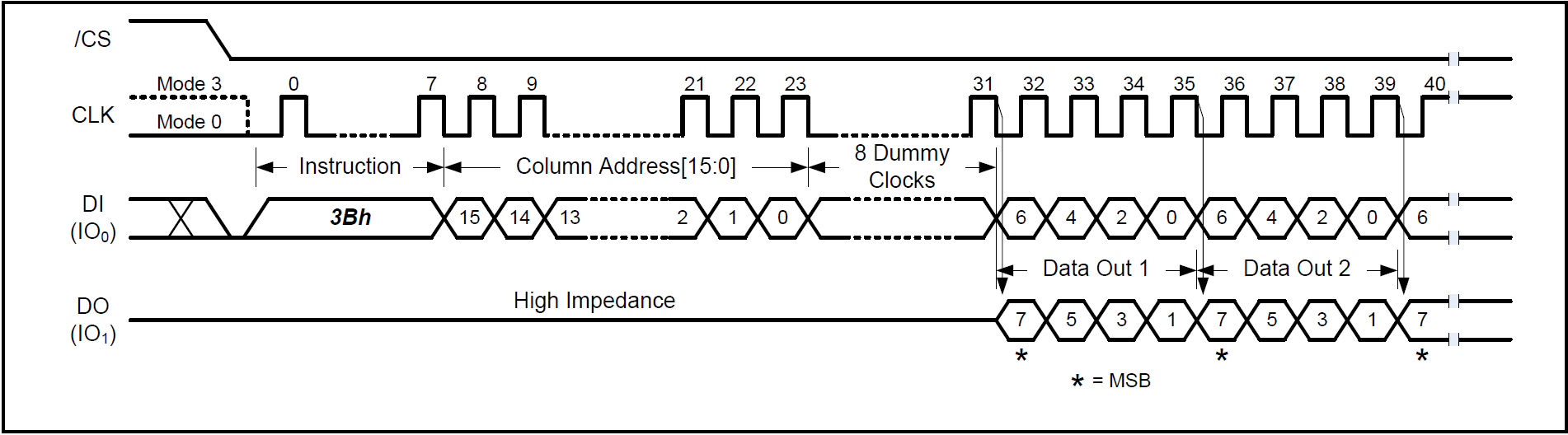


Fast Read with 4-Byte Address Instruction (Sequential Read Mode, BUF=0)

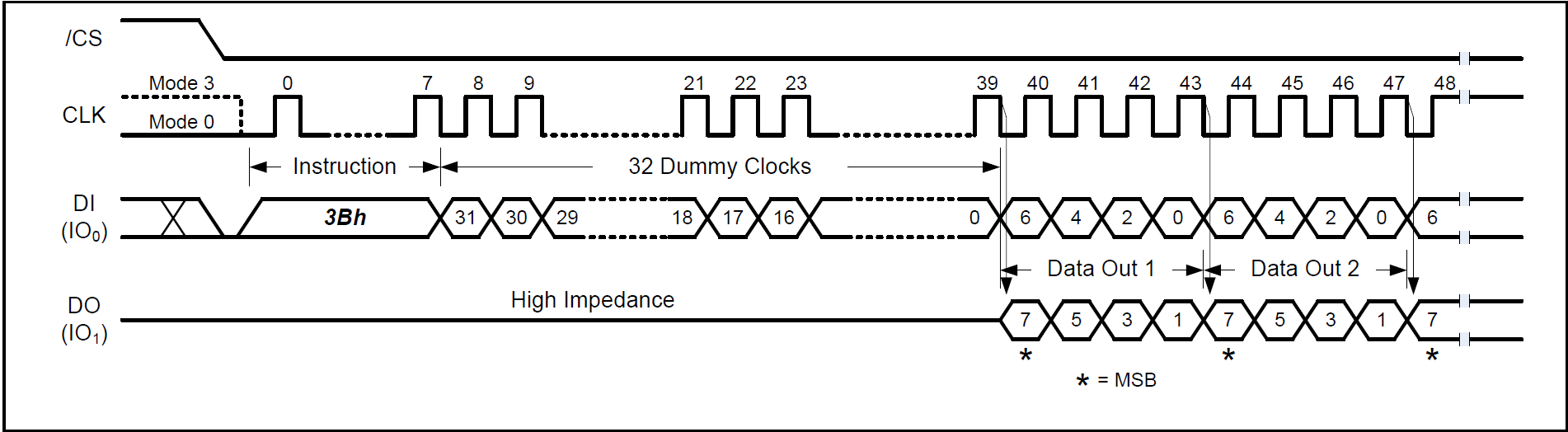
Fast Read with 4-Byte Address (0Ch)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode
  + 16 cycles for column address
  + Dummy cycles (24 clocks if BUF=1, 40 clocks if BUF=0).
* [DI(IO0)] :
  + Send 0Ch Command.
  + Send 16-bit Column Address (CA[15:0]).
  + Send Dummy Clocks (24 or 40 depending on BUF).
* [DO(IO1)] :
  + Outputs Data sequentially, MSB first.
  + Data auto-increments column address.
* [/CS] High : Instruction End.
* Instruction Code : (0Ch) Fast Read with 4-Byte Address
* Operation :
* Reads 1 or more bytes sequentially from the Data Buffer after executing Page Data Read (13h).
* Uses 4-byte addressing for compatibility with large memory systems.
* Data output on DO(IO1), MSB first, at falling edge of CLK.
* Address auto-increments after each byte.
* Read Modes :
* Buffer Read Mode ( BUF = 1 )
  + Output starts from Data Buffer column address.
  + Continues to the end of Data Buffer (2,176 bytes).
  + After last byte, output goes High-Z.
  + Needs 24 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
* Output starts from first byte of Data Buffer.
* Continues through Data Buffer, then automatically to next page.
* Enables continuous memory array read.
* Compatible with Winbond NOR SPI continuous read.
* Needs 40 dummy clocks before data output.
* I/O Behavior :
* DI(IO0) : opcode + column address input.
* DO(IO1) : data output.

### 8.2.15 Fast Read Dual Output (3Bh)



Fast Read Dual Output Instruction ( Buffer Read Mode, BUF = 1 )

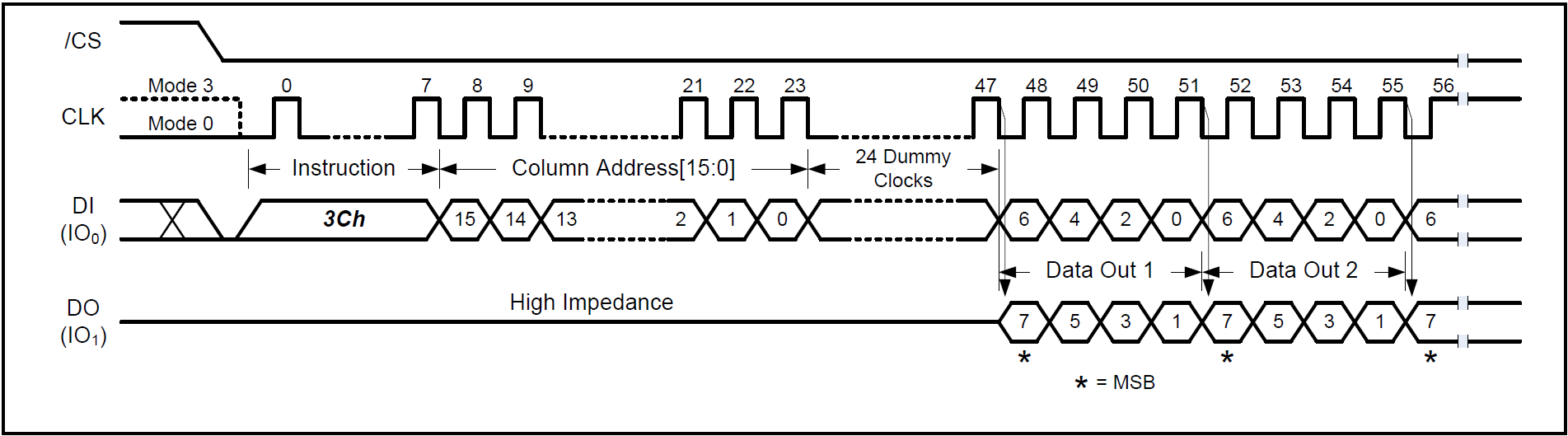


Fast Read Dual Output Instruction ( Sequential Read Mode, BUF = 0 )

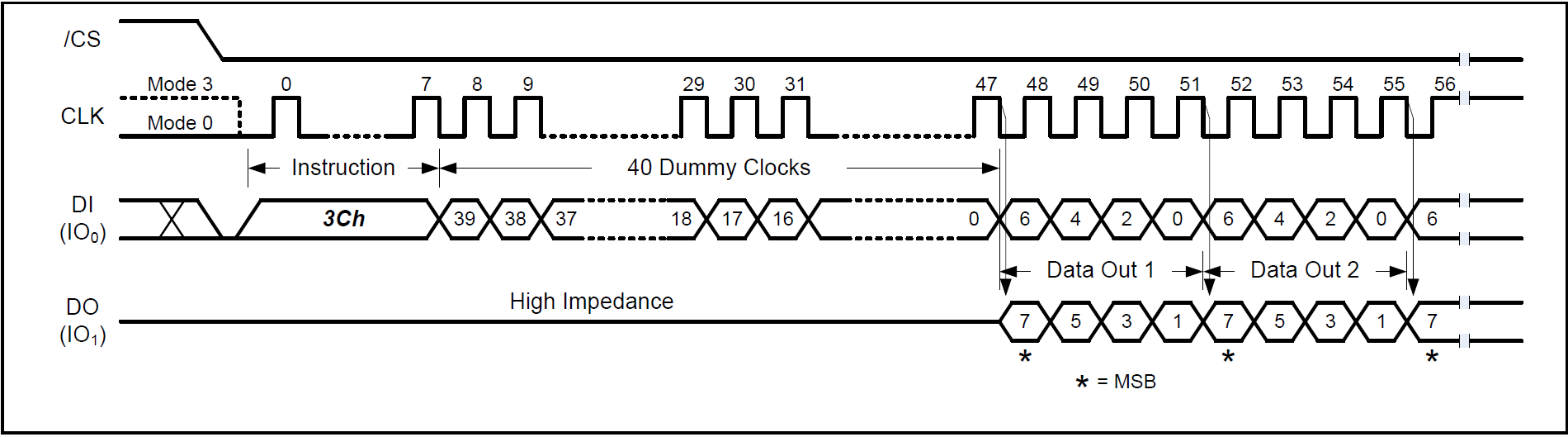
Fast Read Dual Output (3Bh)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode
  + 16 cycles for column address
  + Dummy cycles (8 clocks if BUF=1, 32 clocks if BUF=0).
* [DI(IO0)] :
  + Send 3Bh Command.
  + Send 16-bit Column Address (CA[15:0]).
  + Send Dummy Clocks (8 or 32 depending on BUF).
* [DO(IO0, IO1)] :
  + Outputs Data sequentially (Dual Output), MSB first.
  + Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (3Bh) Fast Read Dual Output
* Operation:
  + Similar to Fast Read (0Bh) but data is output on two lines (IO0, IO1).
  + Provides 2 x data rate compared to standard SPI read.
  + Data is shifted out MSB first at falling edge of CLK.
  + Column address auto-increments as data is read.
* Read Modes :
  + Buffer Read Mode ( BUF = 1 )
  + Starts output from specified Data Buffer column address.
  + Continues until end of Data Buffer (2176 bytes).
  + Output becomes High-Z after last byte.
  + Requires 8 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
  + Starts from first byte of Data Buffer.
  + Continues sequentially, auto-incrementing address.
  + When buffer ends, continues from next memory page (continuous read).
  + Compatible with Winbond NOR SPI flash.
  + Requires 32 dummy clocks before data output.
* I/O Behavior:
  + DI(IO0) : opcode + address input.
  + DO(IO0, IO1) : dual data output.

### 8.2.16 Fast Read Dual Output with 4-Byte Address (3Ch)



Fast Read Dual Output with 4-Byte Address Instruction (Buffer Read Mode, BUF = 1)

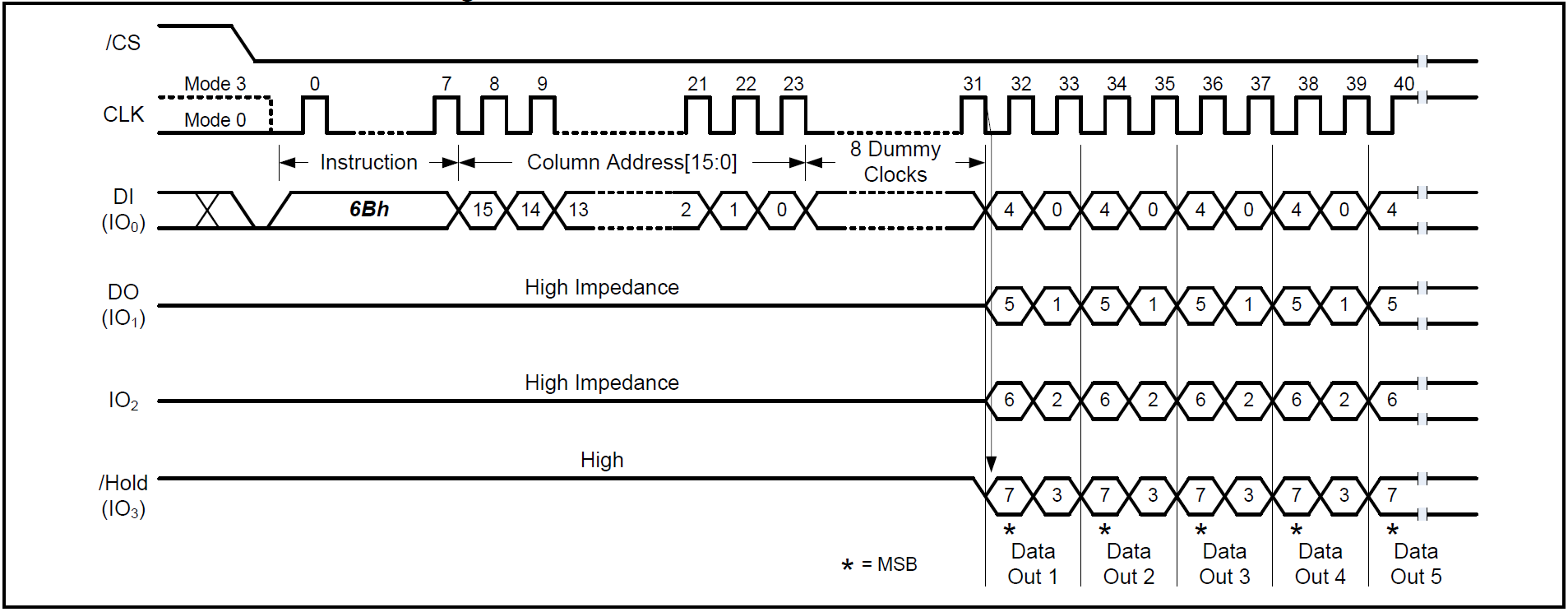


Fast Read Dual Output with 4-Byte Address Instruction (Sequential Read Mode, BUF = 0)

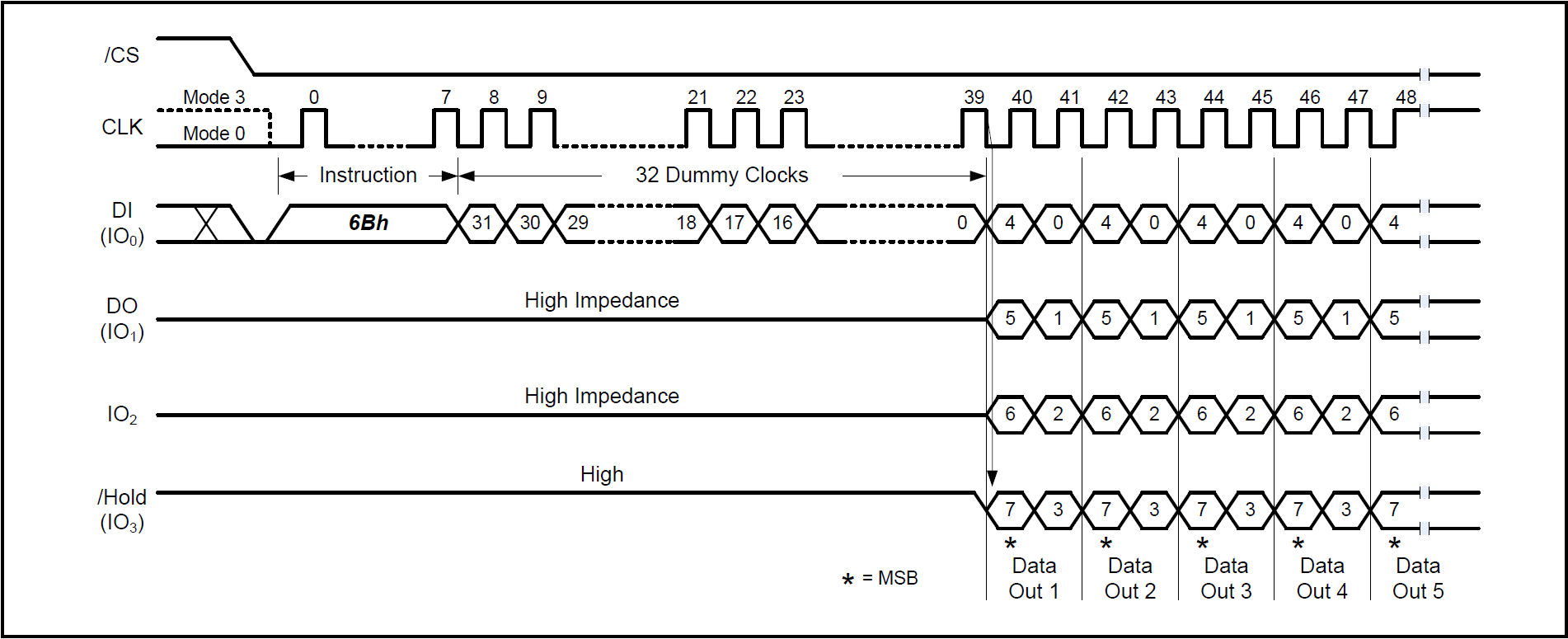
Fast Read Dual Output with 4-Byte Address (3Ch)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode
  + 16 cycles for column address
  + Dummy cycles (24 clocks if BUF=1, 40 clocks if BUF=0).
* [DI(IO0)] :
  + Send 3Ch Command.
  + Send 16-bit Column Address (CA[15:0]).
  + Send Dummy Clocks (24 or 40 depending on BUF).
* [DO(IO0, IO1)] :
  + Outputs Data sequentially (Dual Output), MSB first.
  + Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (3Ch) Fast Read Dual Output with 4-Byte Addres
* Operation :
* Same Fast Read Dual Output (3Bh) but Supports 4-byte addressing.
* Data is output on two lines (IO0, IO1), providing 2 x data rate over standard SPI.
* Data shifted out MSB first at falling edge of CLK.
* Column address auto-increments as data is read.
* Read Modes :
* Buffer Read Mode (BUF = 1)
  + Starts output at specified Data Buffer column address.
  + Continues until end of Data Buffer (2,176 bytes).
  + After last byte, output pins go High-Z.
  + Needs 24 dummy clocks before data output.
* Sequential Read Mode (BUF = 0)
  + Starts from first byte of Data Buffer.
  + Continues through Data Buffer, then next memory page seamlessly.
  + Enables continuous read across full array.
  + Compatible with Winbond NOR SPI flash.
  + Needs 40 dummy clocks before data output.
* I/O Behavior :
* DI(IO0) : opcode + address input.
* DO(IO0, IO1) : dual data output.

### 8.2.17 Fast Read Quad Output (6Bh)



Fast Read Quad Output Instruction (Buffer Read Mode, BUF = 1)

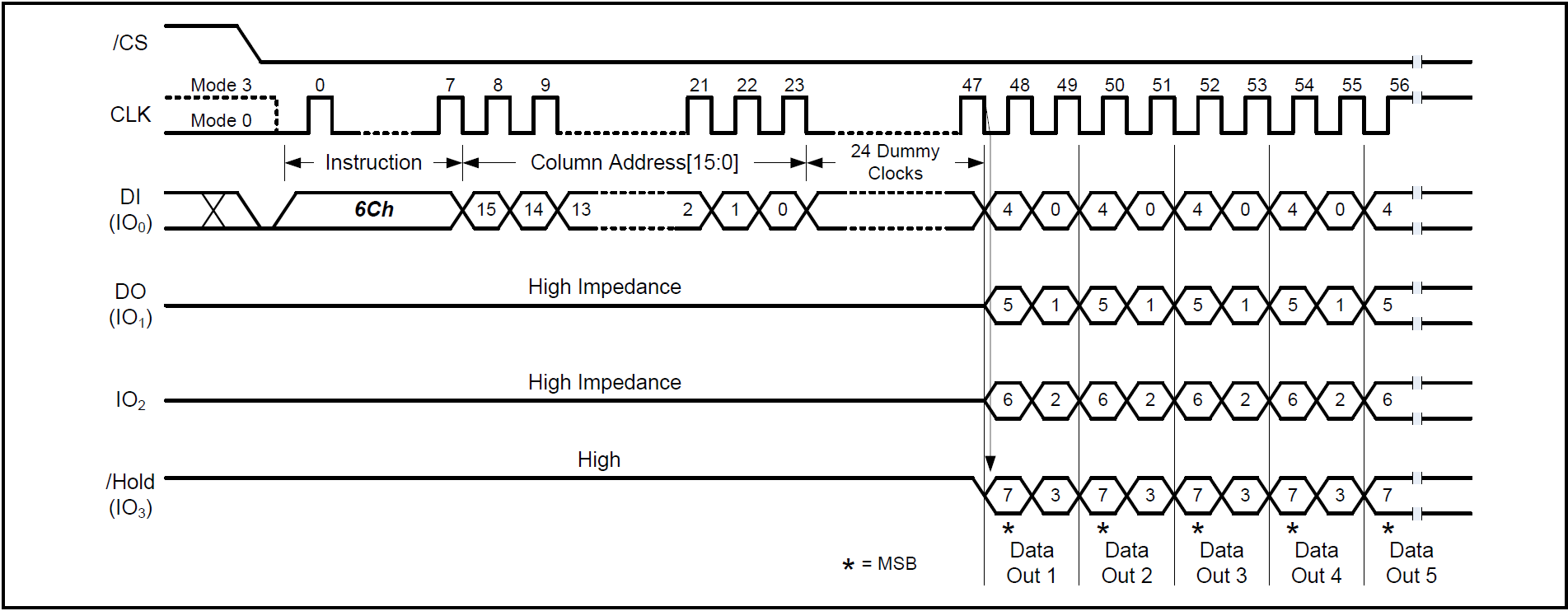


Fast Read Quad Output Instruction (Sequential Read Mode, BUF = 0)

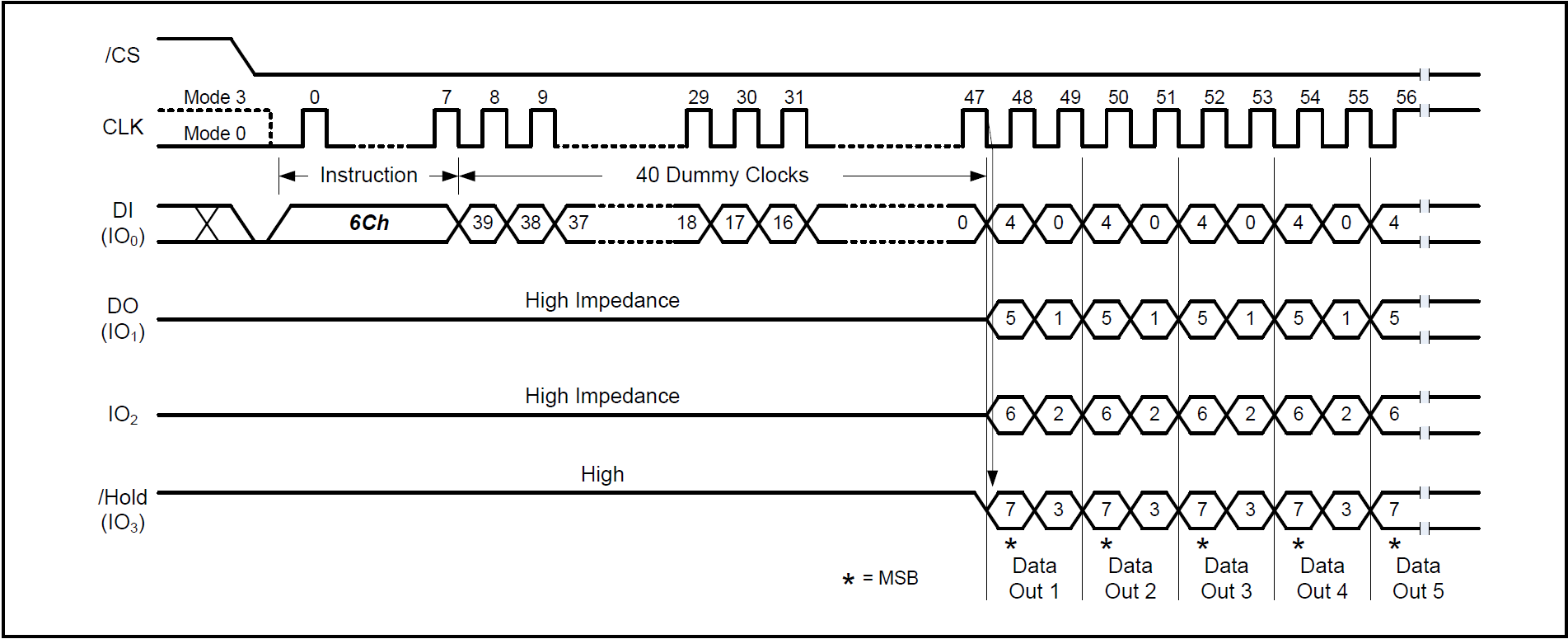
Fast Read Quad Output (6Bh)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode
  + 16 cycles for column address
  + Dummy cycles ( 8 clocks if BUF = 1, 32 clocks if BUF = 0 ).
* [DI(IO0)] :
  + Send 6Bh Command.
  + Send 16-bit Column Address (CA[15:0]).
  + Send Dummy Clocks (8 or 32 depending on BUF).
* [DO(IO0, IO1, IO2, IO3)] :
* Outputs Data sequentially (Quad Output), MSB first.
* Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (6Bh) Fast Read Quad Output
* Operation :
* Provides 4 x data rate compared to standard SPI read.
* Data shifted out MSB first, on falling edge of CLK.
* Column address auto-increments during read.
* Same Fast Read Dual Output (3Bh) but outputs data on 4 lines (IO0, IO1, IO2, IO3).
* Read Modes :
* Buffer Read Mode (BUF = 1)
* Starts output from Data Buffer column address.
* Continues until end of Data Buffer (2176 Bytes).
* After last byte, outputs go High-Z.
* Needs 8 dummy clocks before data output.
* Sequential Read Mode (BUF = 0)
* Starts from first byte of Data Buffer.
* Continues through Data Buffer, then next memory page seamlessly.
* Allows continuous read of entire memory array.
* Compatible with Winbond NOR SPI flash.
* Needs 32 dummy clocks before data output.
* I/O Behavior :
* DI(IO0) : opcode + address input.
* DO(IO1), IO2, IO3, IO0: quad data outputs.
* Data = MSB first.
* Restrictions :
* If WP-E bit = 1 in Status Register ---> this instruction is disabled.

### 8.2.18 Fast Read Quad Output with 4-Byte Address (6Ch)



Fast Read Quad Output with 4-Byte Address Instruction (Buffer Read Mode, BUF = 1)

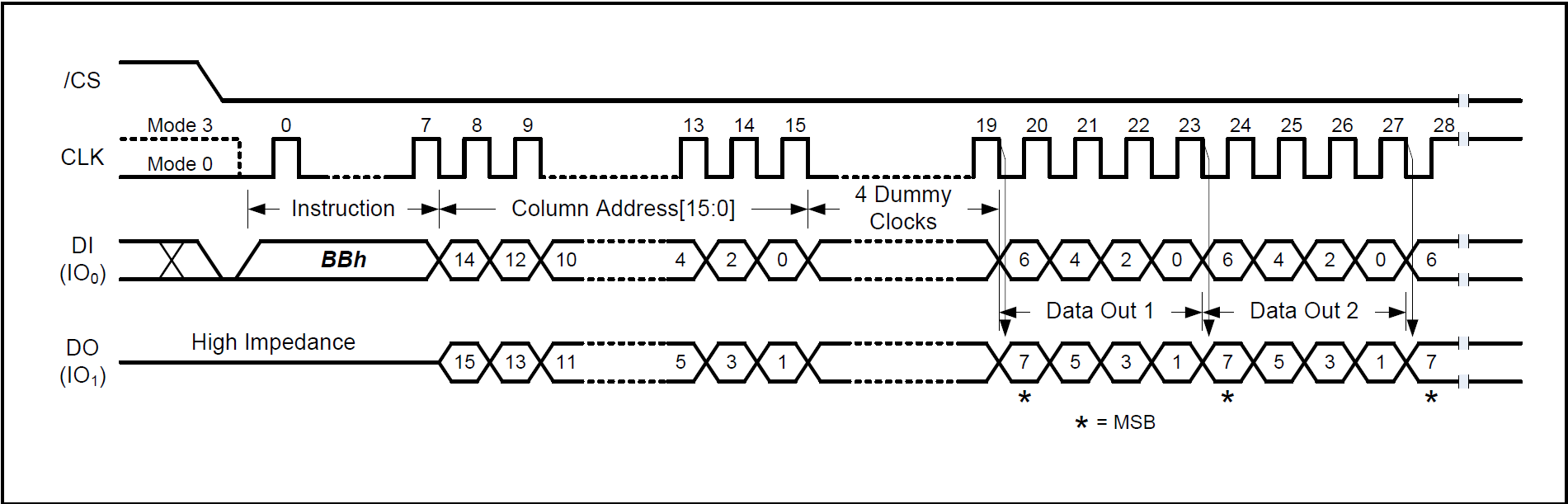


Fast Read Quad Output with 4-Byte Address Instruction (Sequential Read Mode, BUF = 0)

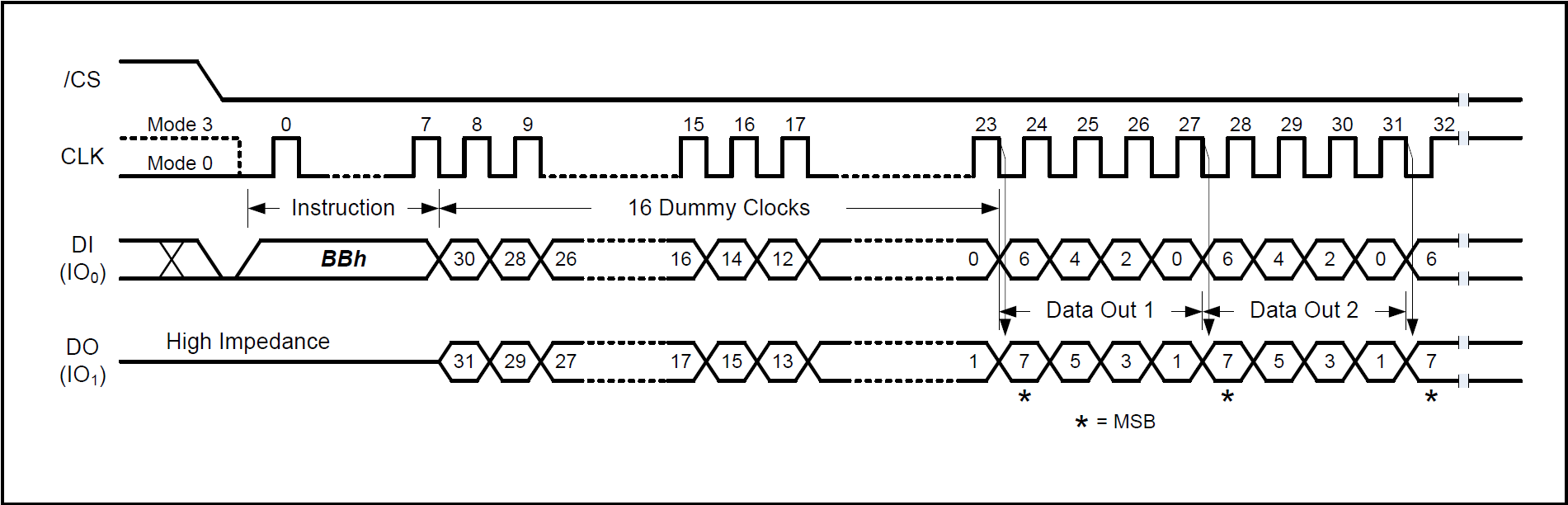
Fast Read Quad Output with 4-Byte Address (6Ch)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode.
  + 16 cycles for column address.
  + Dummy cycles (24 clocks if BUF = 1, 40 clocks if BUF = 0).
* [DI(IO0)] :
  + Send 6Ch Command.
  + Send 16-bit Column Address (CA[15:0]).
  + Send Dummy Clocks (24 or 40 depending on BUF).
* [DO(IO0, IO1, IO2, IO3)] :
  + Outputs Data sequentially (Quad Output), MSB first.
  + Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (6Ch) Fast Read Quad Output with 4-Byte Address
* Operation :
* Same Fast Read Quad Output (6Bh) but Supports 4-byte addressing.
* Outputs data on 4 lines (IO0, IO1, IO2, IO3), providing 4 x data rate vs. standard SPI.
* Data shifted out MSB first, on falling edge of CLK.
* Column address auto-increments as data is read.
* Read Modes :
  + Buffer Read Mode ( BUF = 1 )
  + Starts from specified Data Buffer column address.
  + Reads until end of Data Buffer (2176 Bytes).
  + After last byte, outputs go High-Z.
  + Needs 24 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
  + Starts at first byte of Data Buffer.
  + Continues into next memory page automatically.
  + Enables continuous read across the entire memory array.
  + Compatible with Winbond NOR SPI flash.
  + Needs 40 dummy clocks before data output.
* Restrictions :
* When WP-E bit = 1 in Status Register, Instruction is Disabled.

### 8.2.19 Fast Read Dual I/O (BBh)



Fast Read Dual I/O Instruction ( Buffer Read Mode, BUF = 1 )

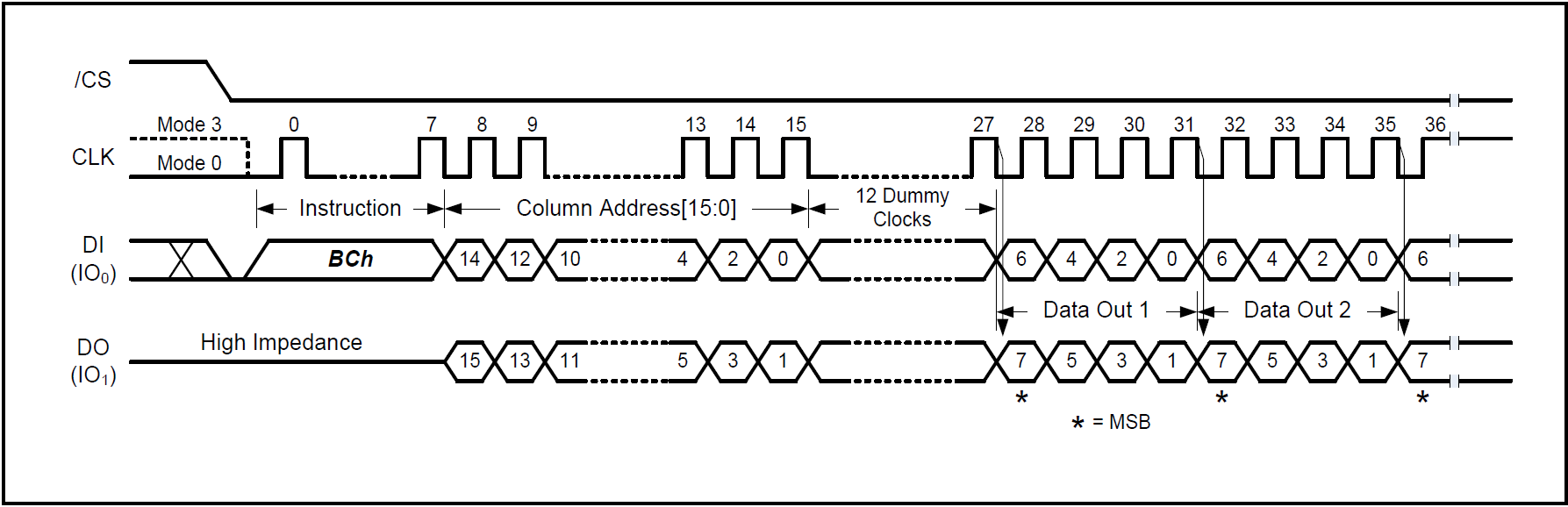


Fast Read Dual I/O Instruction ( Sequential Read Mode, BUF = 0 )

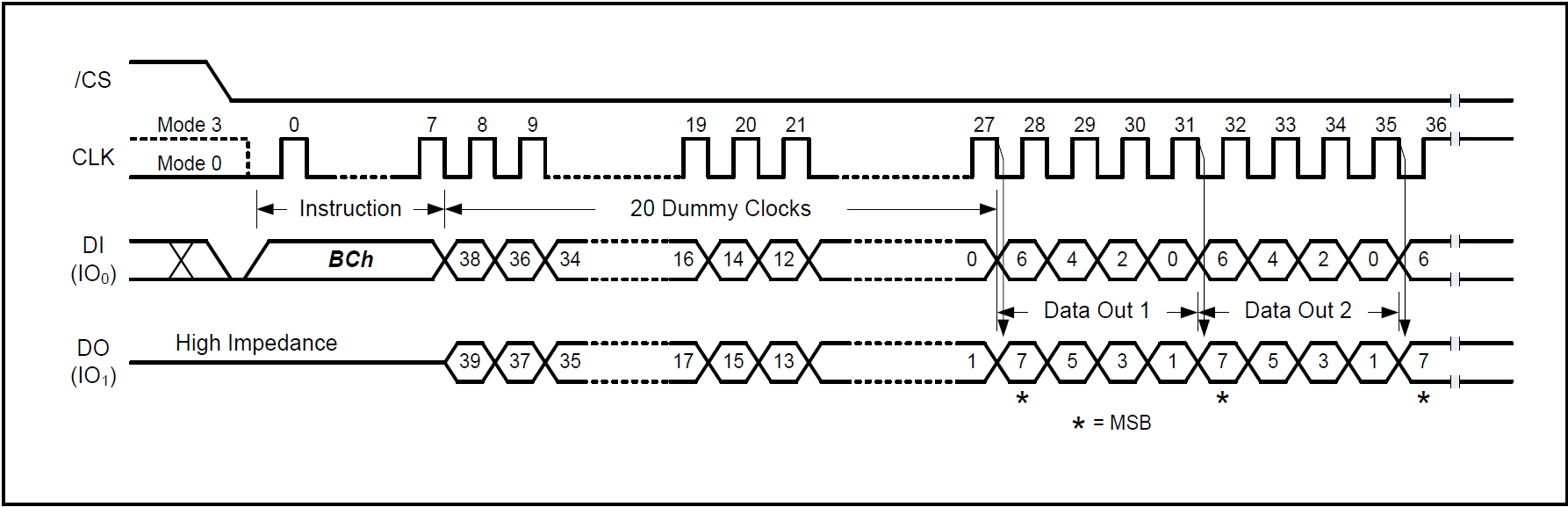
Fast Read Dual I/O (BBh)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode.
  + 16 cycles for column address (Dual I/O).
  + Dummy cycles ( 4 clocks if BUF = 1, 16 clocks if BUF = 0 ).
* [DI(IO0, IO1)] :
  + Send BBh Command.
  + Send 16-bit Column Address (Dual I/O).
  + Send Dummy Clocks ( 4 or 16 depending on BUF ).
* [DO(IO0, IO1)] :
  + Outputs Data sequentially (Dual I/O), MSB first.
  + Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (BBh) Fast Read Dual I/O
* Operation :
  + Same Fast Read Dual Output (3Bh) but with Dual I/O capability:
    - Both address input and data output use two lines (IO0, IO1).
    - Reduces instruction overhead, improves random access.
    - Useful for XIP (eXecute In Place) in Dual SPI applications.
  + Data shifted out MSB first, on falling edge of CLK.
  + Column address auto-increments as data is read.
* Read Modes:
* Buffer Read Mode ( BUF = 1 )
  + Starts output at specified Data Buffer column address.
  + Reads until end of Data Buffer ( 2176 Bytes ).
  + After last byte, outputs go High-Z.
  + Needs 4 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
  + Starts from first byte of Data Buffer.
  + Continues through Data Buffer, then next memory page seamlessly.
  + Enables continuous read across the full array.
  + Compatible with Winbond NOR SPI flash.
  + Needs 16 dummy clocks before data output.

### 8.2.20 Fast Read Dual I/O with 4-Byte Address (BCh)



Fast Read Dual I/O with 4-Byte Address Instruction (Buffer Read Mode, BUF = 1)

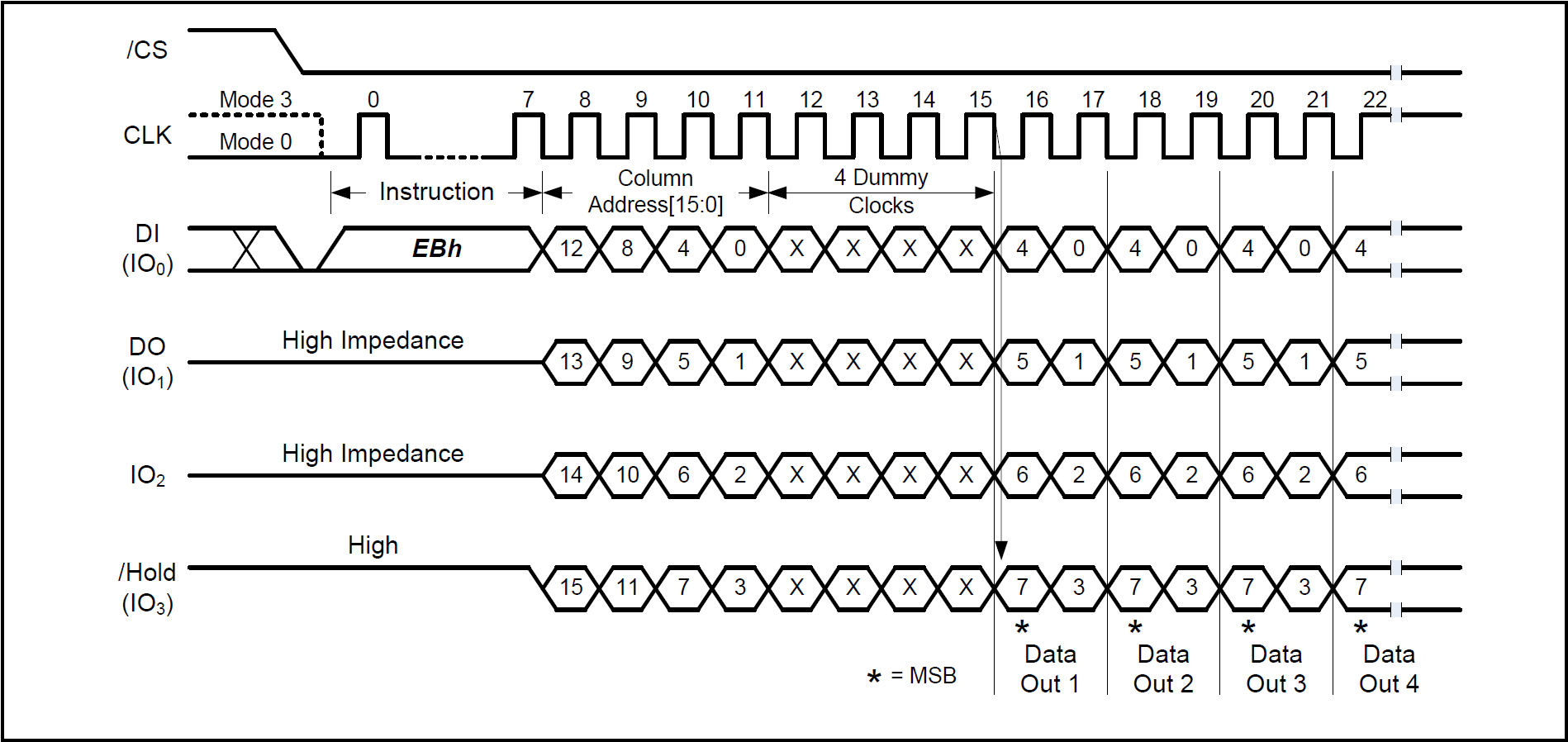


Fast Read Dual I/O with 4-Byte Address Instruction (Sequential Read Mode, BUF = 0)

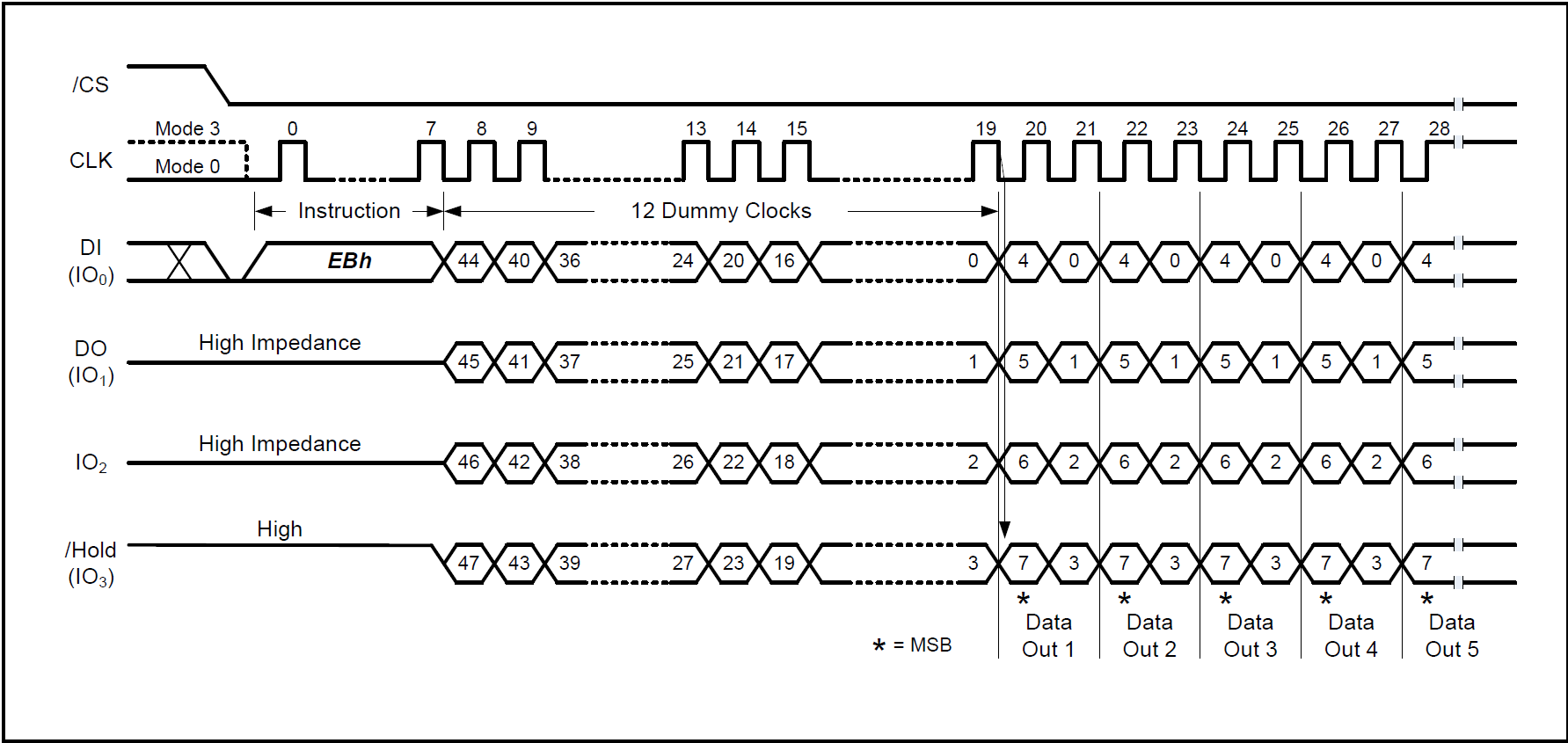
Fast Read Dual I/O with 4-Byte Address (BCh)

* [/CS] Low : Instruction Start.
* [CLK] :
  + Provides 8 cycles for opcode.
  + 16 cycles for column address (Dual I/O).
  + Dummy cycles (12 clocks if BUF = 1, 20 clocks if BUF = 0).
* [DI(IO0, IO1)] :
  + Send BCh Command.
  + Send 16-bit Column Address (Dual I/O).
  + Send Dummy Clocks (12 or 20 depending on BUF).
* [DO(IO0, IO1)] :
  + Outputs Data sequentially (Dual I/O), MSB first.
  + Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (BCh) Fast Read Dual I/O with 4-Byte Address
* Operation :
* Same Fast Read Dual I/O (BBh) but supports 4-byte addressing.
* Both address input and data output use two lines (IO0, IO1).
* Provides reduced overhead, suitable for XIP (Execute In Place) in Dual SPI systems.
* Data shifted out MSB first, on falling edge of CLK.
* Column address auto-increments as data is read.
* Read Modes :
* Buffer Read Mode ( BUF = 1 )
  + Starts output from specified Data Buffer column address.
  + Reads until end of Data Buffer (2176 Bytes).
  + Output pins go High-Z after last byte.
  + Requires 12 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
* Starts output from first byte of Data Buffer.
* Continues through Data Buffer and automatically into next page.
* Enables continuous read through full memory array.
* Compatible with Winbond NOR SPI flash.
* Requires 20 dummy clocks before data output.

### 8.2.21 Fast Read Quad I/O (EBh)



Fast Read Quad I/O Instruction ( Buffer Read Mode, BUF = 1 )

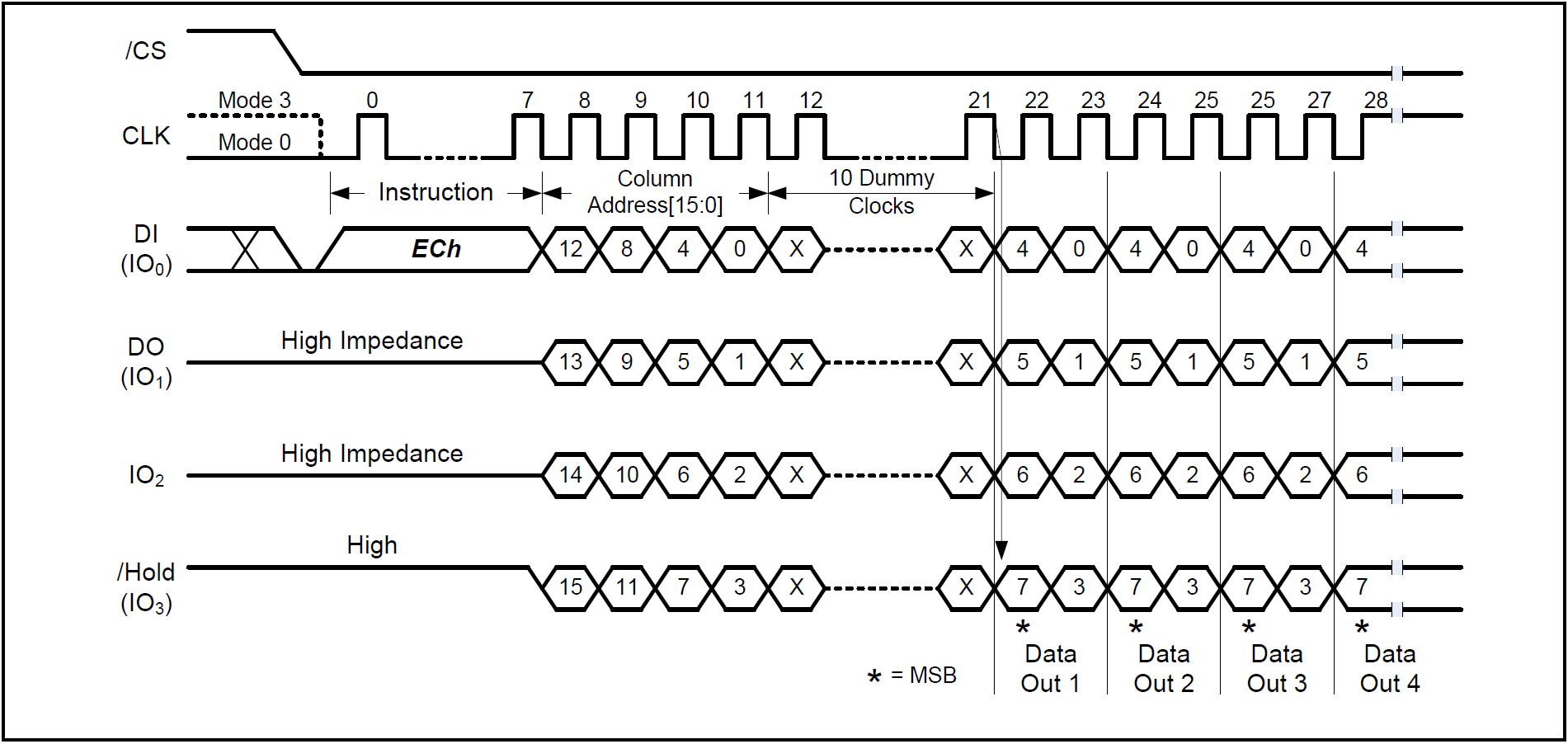


Fast Read Quad I/O Instruction (Sequential Read Mode, BUF =0 )

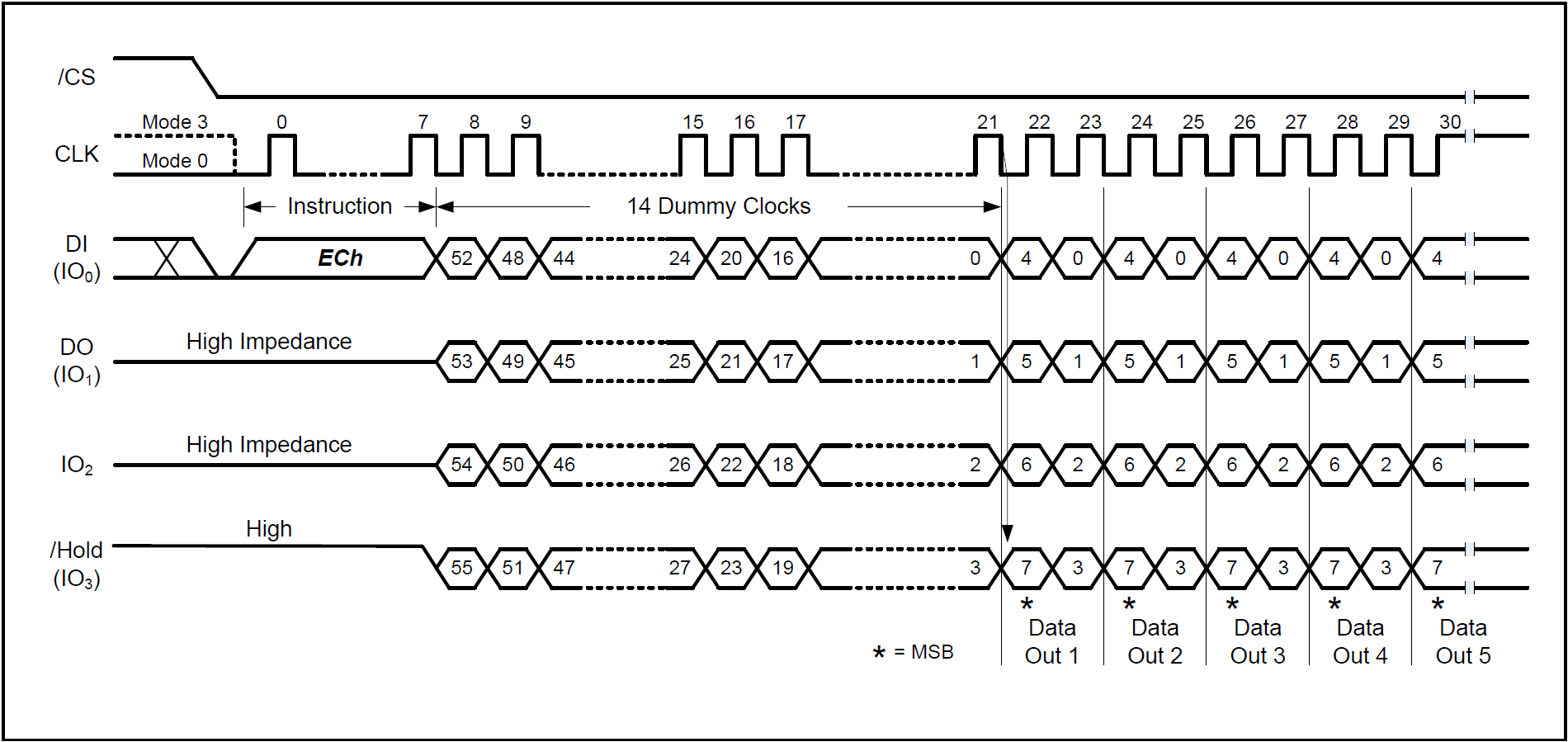
Fast Read Quad I/O (EBh)

* [/CS] Low : Instruction Start.
* [CLK] : Provides 8 cycles for opcode, 16 cycles for column address (Quad I/O), Dummy cycles (4 clocks if BUF = 1, 16 clocks if BUF = 0).
* [DI(IO0 ~ IO3)] : Send EBh Command, Send 16-bit Column Address ( Quad I/O ), Send Dummy Clocks ( 4 or 16 depending on BUF ).
* [DO(IO0 ~ IO3)] :
* Outputs Data sequentially ( Quad I/O ), MSB first
* Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (EBh) Fast Read Quad I/O
* Operation :
* Similar to Fast Read Dual I/O (BBh) but with Quad I/O:
* Both address input and data output use four lines (IO0, IO1, IO2, IO3).
* Significantly reduces instruction overhead, enabling fast random access.
* Useful for XIP (Execute In Place) directly from Quad SPI.
* Data shifted out MSB first, on falling edge of CLK.
* Column address auto-increments during read.
* Read Modes :
* Buffer Read Mode ( BUF = 1 )
  + Starts output from specified Data Buffer column address.
  + Reads until end of Data Buffer (2176 Bytes).
  + After last byte, outputs go High-Z.
  + Requires 4 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
  + Starts from first byte of Data Buffer.
  + Continues through Data Buffer, then automatically to next page.
  + Allows continuous read of entire memory array.
  + Compatible with Winbond NOR SPI flash.
  + Requires 16 dummy clocks before data output.
* I/O Behavior :
* IO0 ~ IO3 : used for both address input and data output.
* Data output = MSB first.
* Restrictions :
* When WP-E bit = 1 in Status Register ---> Instruction Disabled.

### 8.2.22 Fast Read Quad I/O (EBh)



Fast Read Quad I/O with 4-Byte Address Instruction (Buffer Read Mode, BUF = 1)

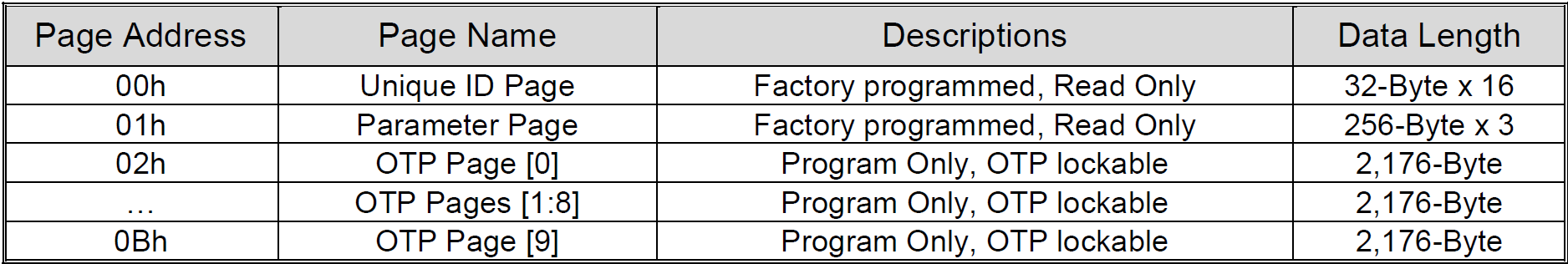


Fast Read Quad I/O with 4-Byte Address Instruction (Sequential Read Mode, BUF = 0)

Fast Read Quad I/O with 4-Byte Address (ECh)

* [/CS] Low : Instruction Start.
* [CLK] : Provides 8 cycles for opcode, 16 cycles for column address (Quad I/O), Dummy cycles (10 clocks if BUF=1, 14 clocks if BUF=0).
* [DI(IO0~IO3)] : Send ECh Command, Send 16-bit Column Address (Quad I/O), Send Dummy Clocks (10 or 14 depending on BUF).
* [DO(IO0~IO3)] : Outputs Data sequentially (Quad I/O), MSB first, Address auto-increments.
* [/CS] High : Instruction End.
* Instruction Code : (ECh) Fast Read Quad I/O with 4-Byte Address
* Operation:
* Similar to Fast Read Quad I/O (EBh) but supports 4-byte addressing.
* Both address input and data output use four lines (IO0, IO1, IO2, IO3).
* Reduced instruction overhead, optimized for XIP (Execute In Place) in Quad SPI systems.
* Data shifted out MSB first, on falling edge of CLK.
* Column address auto-increments during read.
* Read Modes :
* Buffer Read Mode ( BUF = 1 )
  + Starts from specified Data Buffer column address.
  + Reads until end of Data Buffer (2176 Bytes).
  + After last byte, outputs go High-Z.
  + Requires 10 dummy clocks before data output.
* Sequential Read Mode ( BUF = 0 )
  + Starts from first byte of Data Buffer.
  + Continues seamlessly through next memory page.
  + Enables continuous read of full memory array.
  + Compatible with Winbond NOR SPI flash.
  + Requires 14 dummy clocks before data output.
* I/O Behavior :
* IO0 ~ IO3 : used for both address input and data output.
* Data = MSB first.
* Restrictions :
* When WP-E bit = 1 in Status Register ---> Instruction disabled.

### 8.2.23 Accessing Unique ID / Parameter / OTP Pages (OTP-E = 1)



Special Pages:

00h ---> Unique ID Page (Factory programmed, Read Only, 32-Byte ×16)

01h ---> Parameter Page (Factory programmed, Read Only, 256-Byte ×3)

02h ---> OTP Page [0] (Program Only, OTP lockable, 2,176-Byte)

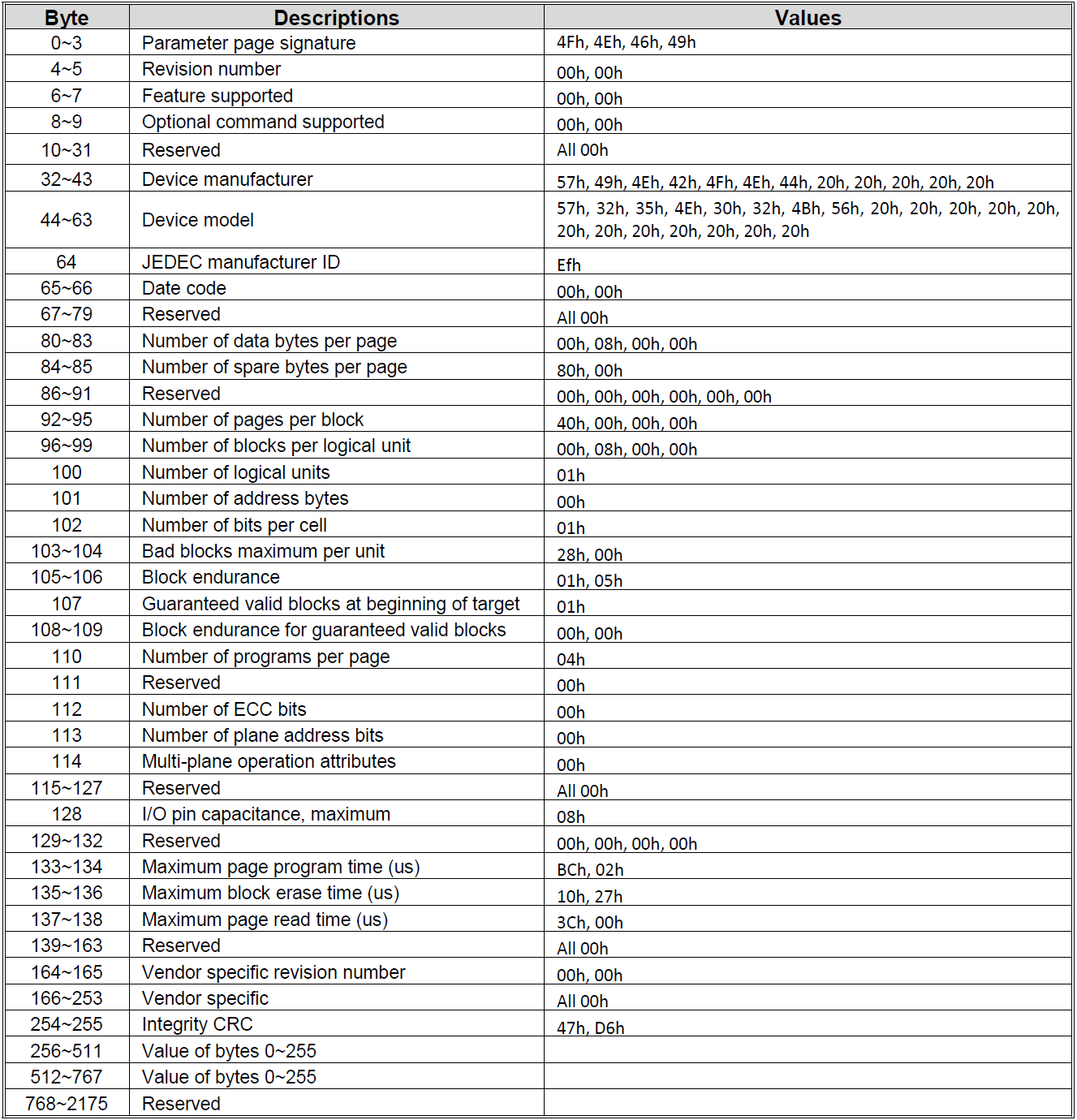
03h ~ 0Ah ---> OTP Pages [1–8] (Program Only, OTP lockable, 2,176-Byte each)

0Bh ---> OTP Page [9] (Program Only, OTP lockable, 2,176-Byte)

OTP Page Access (OTP-E=1)

* [/CS] Low : Instruction Start.
* [CLK] :
* Standard command cycles depending on instruction (13h for Page Data Read, 02h/84h for Program Data Load, 10h for Program Execute).
* [DI(IO0)] :
* Send Command ( 13h for read, 02h/84h for program load, 10h for program execute ).
* Send OTP Page Address (00h ~ 0Bh).
* [DO(IOx)] :
* Data Out (for Read instructions).
* High Impedance when not outputting data.
* [/CS] High : Instruction End.
* Access Control :
  + OTP-E bit = 1 in Status Register-2 => Enter OTP/UID/Parameter access mode.
  + OTP-E bit = 0, Return to main memory Mode.
  + Read Operations:
  + Page Data Read (13h) command with OTP Page address.
  + Load into Data Buffer ( BUSY = 0 ), Can be read using any standard read command (03h, 0Bh).
  + Enabled ECC can protect OTP readout( Additional 64 bytes for parity ).
  + Must follow Buffer Read Mode command structure (CA[15:0] addressing).
* Program & Lock Operations :
  + OTP pages are program-only ( no erase, only change 1 ---> 0 ).
  + Data is loaded via Program Data Load commands, then written using Program Execute with OTP Page address.
  + Once programmed, OTP-L bit can be set to permanently lock the page (no further modification).
  + When ECC is Enabled, parity is stored in an additional 64 bytes.
* SR1-L OTP Lock Operation :
* OTP pages can also be locked by SR1-L bit in Status Register-1.
* Requires:
  + Set OTP-E = 1
  + Send Program Execute (Page address = don’t care)
  + Set SR1-L=1 ---> Permanently locks OTP protection.
  + Once locked, cannot revert.

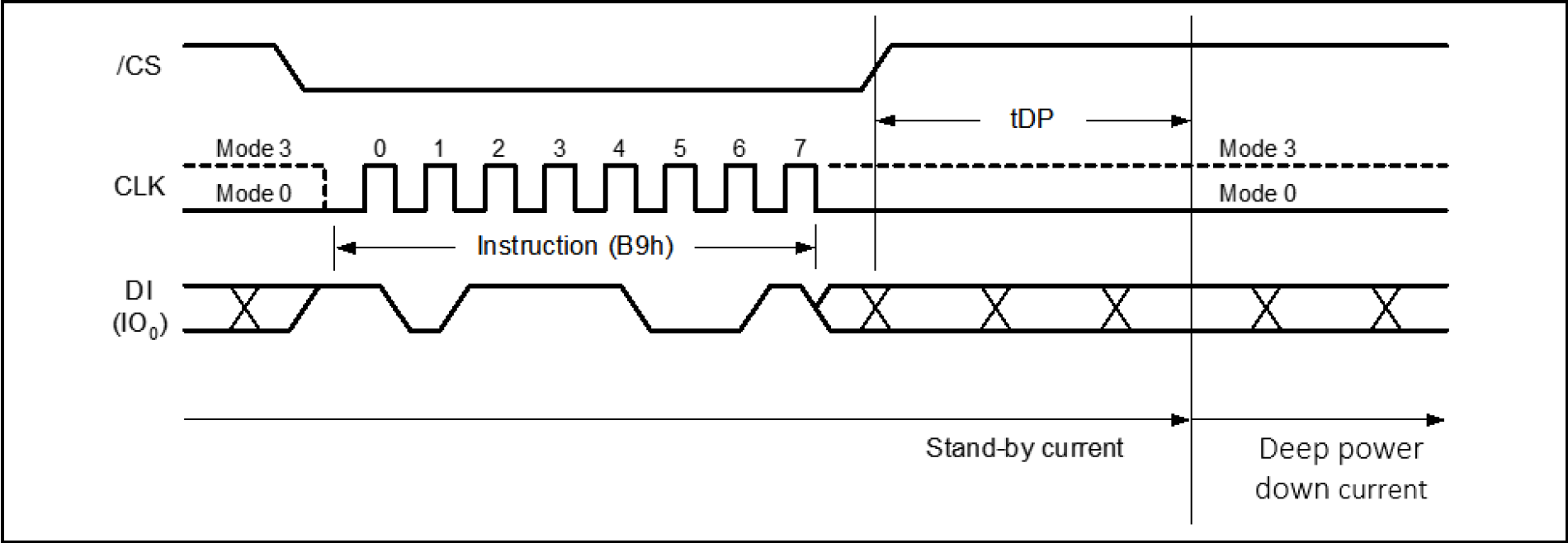
### 8.2.24 Parameter Page Data Definitions



Accessing the Parameter Page :

1. Set OTP-E = 1 ( Switch to special page access mode ).
2. Use Page Data Read (13h) + Page Address = 01h to load the Parameter Page into the Data Buffer.
3. Then use the Read Data (03h / 0Bh …) Command to Read out the Data Buffer.
4. FW parses the first 256 bytes ---> extracts the NAND parameters.

### 8.2.25 Deep Power-Down (B9h)

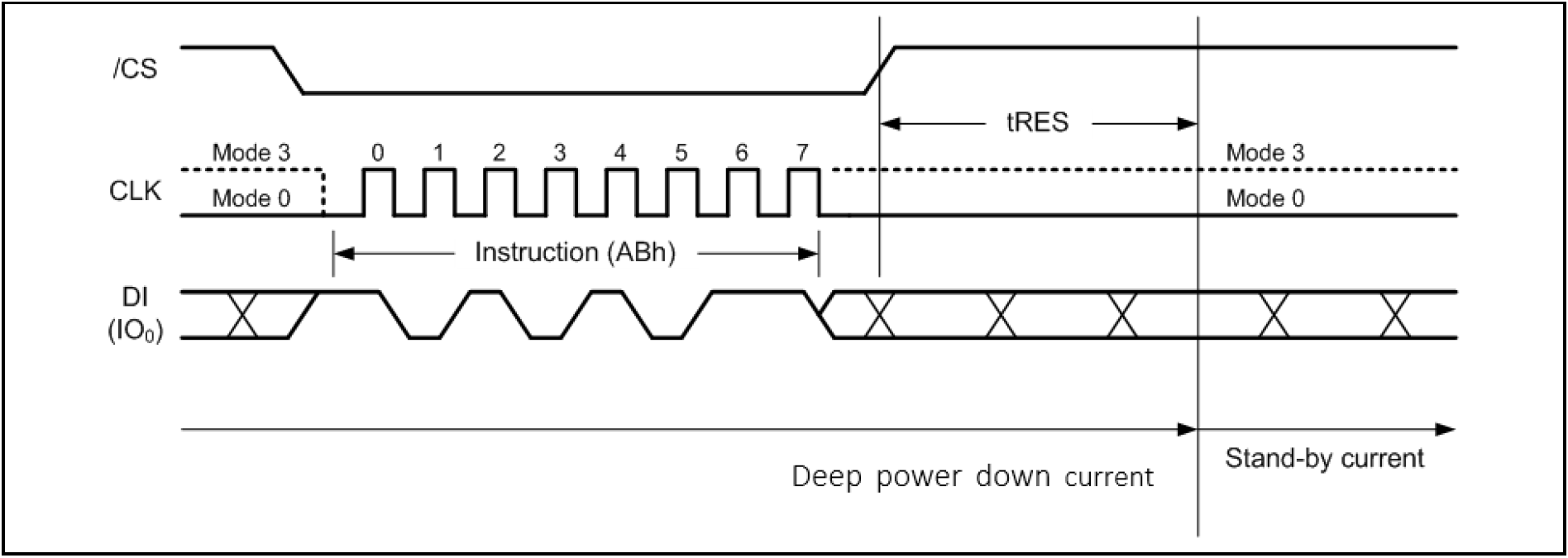


Deep Power-Down Instruction

Deep Power-Down (B9h)

* [/CS] Low : Instruction Start
* [CLK] : 8 clock cycles
* [DI(IO₀)] : Send B9h command
* [DO(IO₁)] : High Impedance
* [/CS] High : Instruction End ---> Device enters Deep Power-Down within tDP
* Instruction Code : B9h
* Purpose:
  + Reduces standby current further than normal standby.
  + Useful for battery-powered systems.
* Behavior in Deep Power-Down :
  + Only a limited set of instructions are recognized:
  + Release from Power-Down / Read Device ID (ABh)
  + Reset (66h + 99h, FFh)
  + All other instructions (including Read Status Register) are ignored.
* Exit Condition :
  + Must issue ABh (Release Power-Down) or Reset ( 66h+99h / FFh ).
  + Device returns to normal standby current ( ICC1 ).
* FW Usage Notes :
  + Ensures maximum write protection (ignores commands except release).
  + Device always powers up in normal standby (ICC1), not deep power-down.

### 8.2.26 Release Power-Down (ABh)



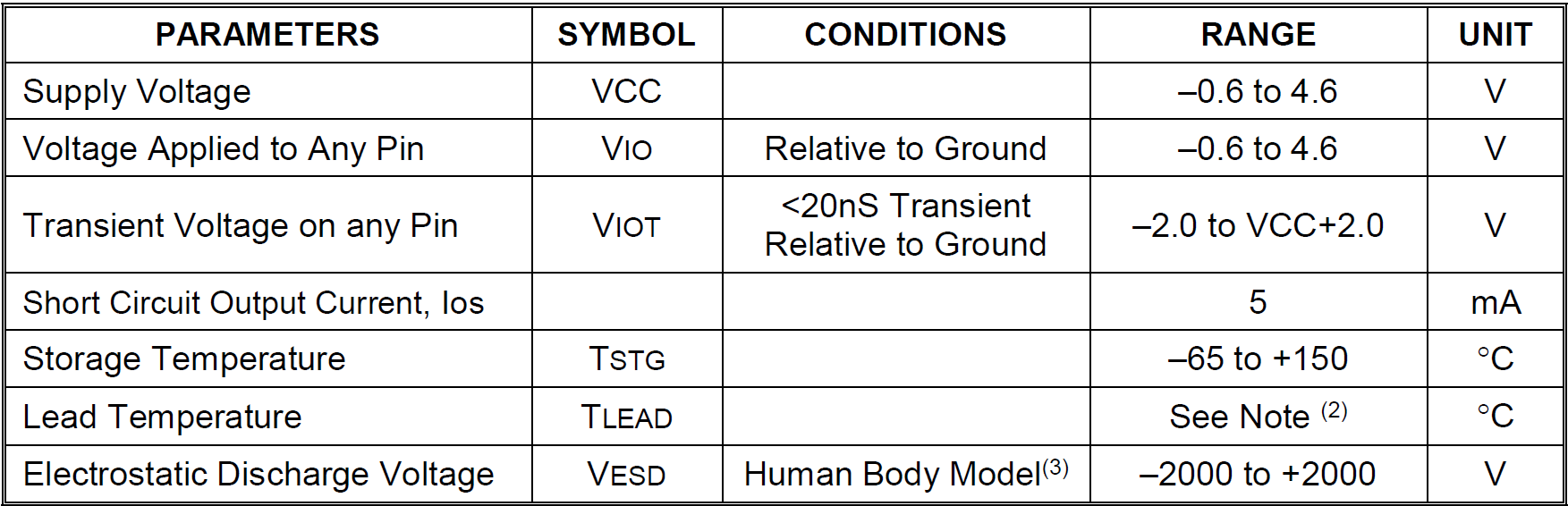
Release Power-down Instruction

Release Power-Down (ABh)

* [/CS] Low : Instruction Start
* [CLK] : 8 clock cycles
* [DI(IO₀)] : Send ABh command
* [DO(IO₁)] : High Impedance
* [/CS] High : Instruction End → Device exits Deep Power-Down after tRES
* Instruction Code : (ABh) Release Power-Down
* Purpose :
  + Used to release device from Deep Power-Down state (entered via B9h).
  + Returns the device to normal standby current (ICC1).
* Behavior After Release :
  + During tRES, /CS must stay high.
  + All Status Register bits reset to default shipment values, except:
  + OTP-L and SR1-L ---> remain unchanged (same as before entering Deep Power-Down).
  + Once tRES expires, all normal commands are accepted.
* FW Usage Notes :
* Required wake-up sequence after B9h.
* Must respect tRES wait time before sending other instructions.
* FW should reconfigure any Status Register settings if needed (since they reset).

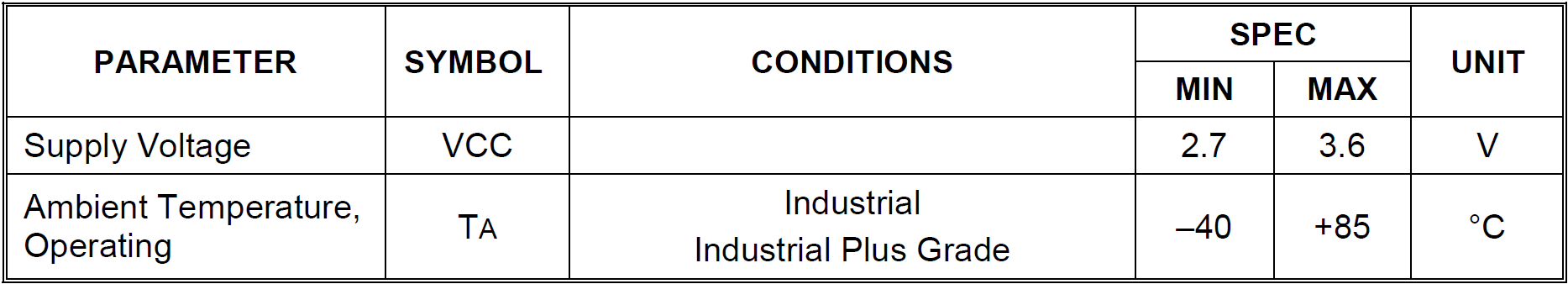
## 9. ELECTRICAL CHARACTERISTICS

### 9.1 Absolute Maximum Ratings



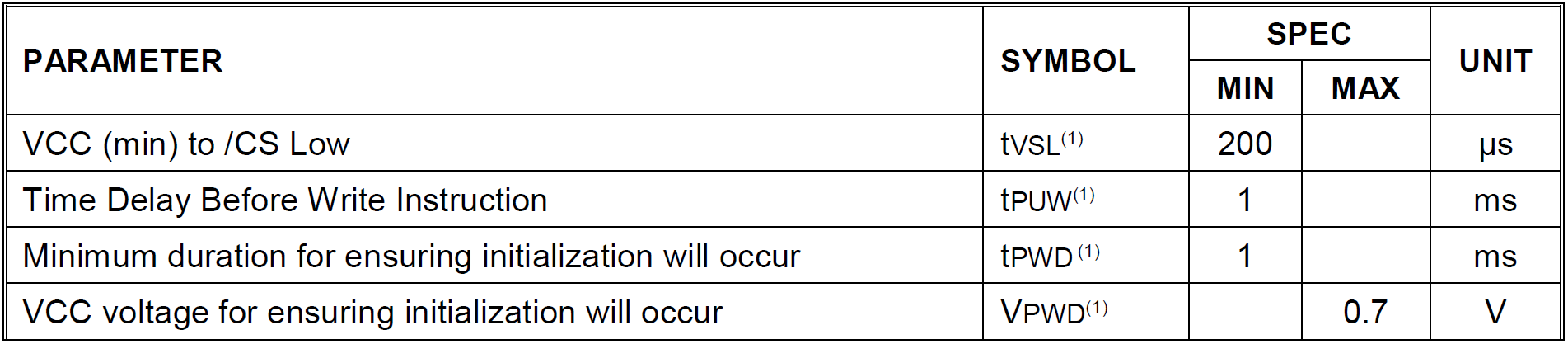
* Defines non-operational limits (not normal working conditions).
* Exceeding these values may permanently damage the device.
* Not directly controlled by FW, but FW must avoid operation when system voltage is outside safe ranges.

### 9.2 Operating Ranges



* Supply Voltage (VCC): 2.7V ~ 3.6V --> FW should check power status before write operations.
* Operating Temperature (TA) : -40°C ~ +85°C --> FW should be tested for reliable function across this range.

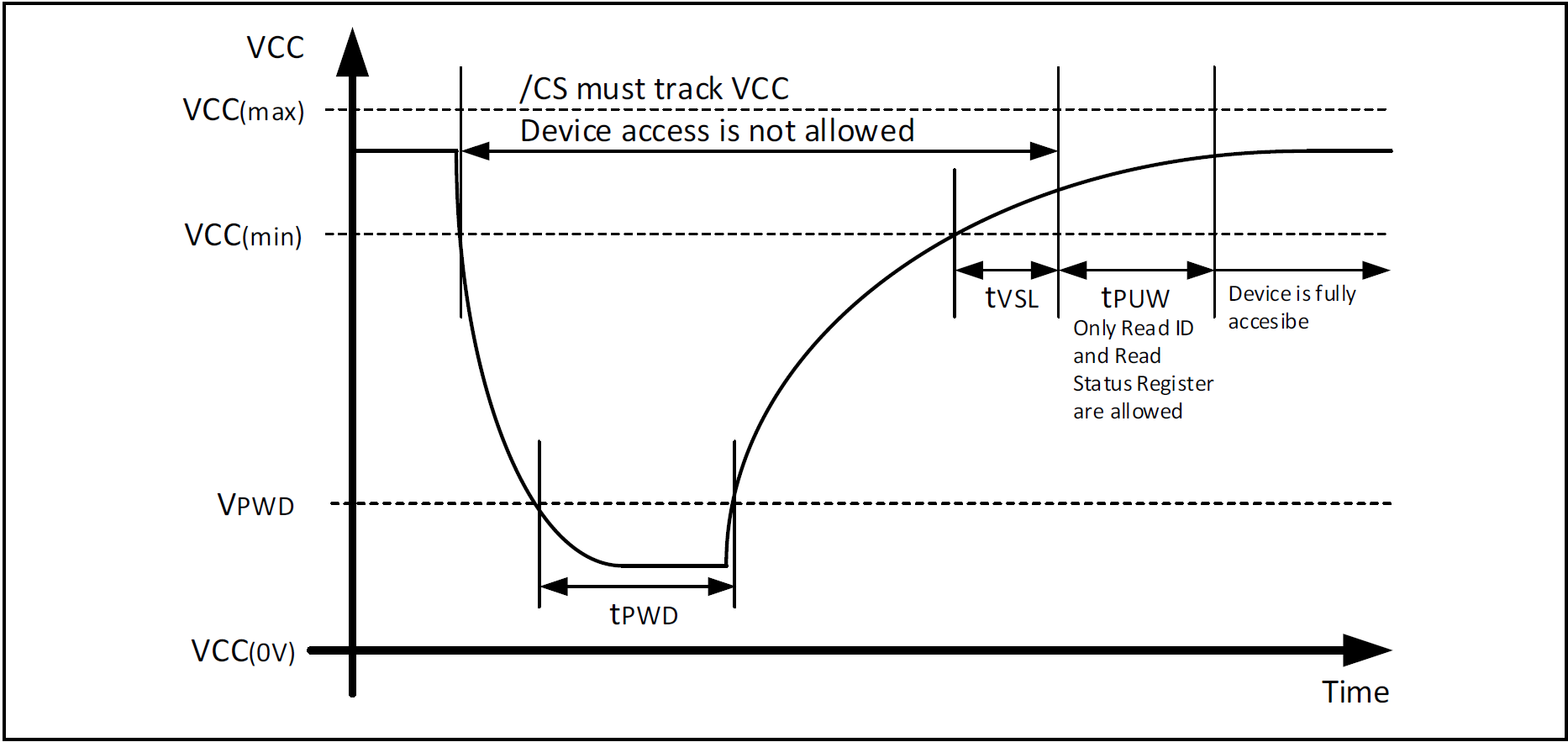
### 9.3 Power-up Power-down Timing Requirements



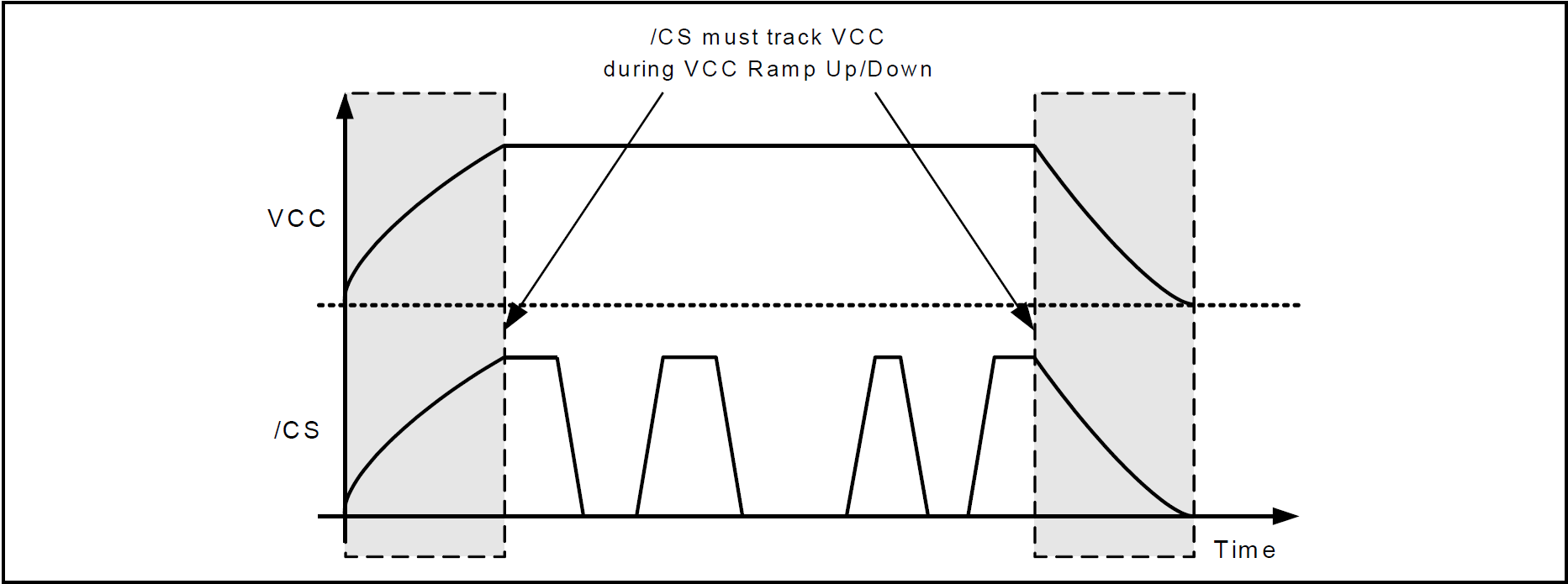
Note : These parameters are characterized only.

/CS must track VCC during power-up and power-down to prevent false commands.

FW must always insert proper delay and status polling before any operation.

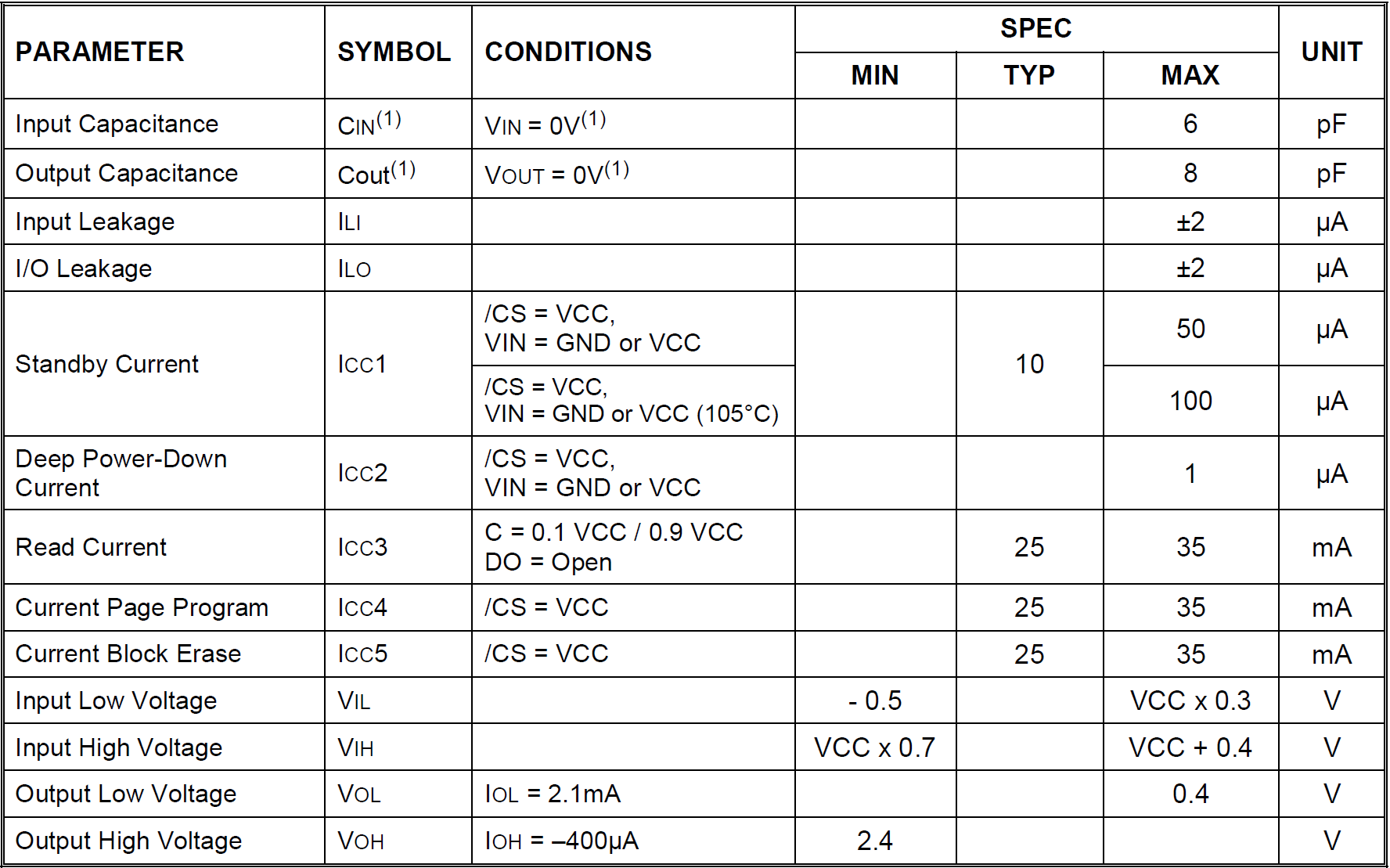


Power-up Timing and Voltage Levels



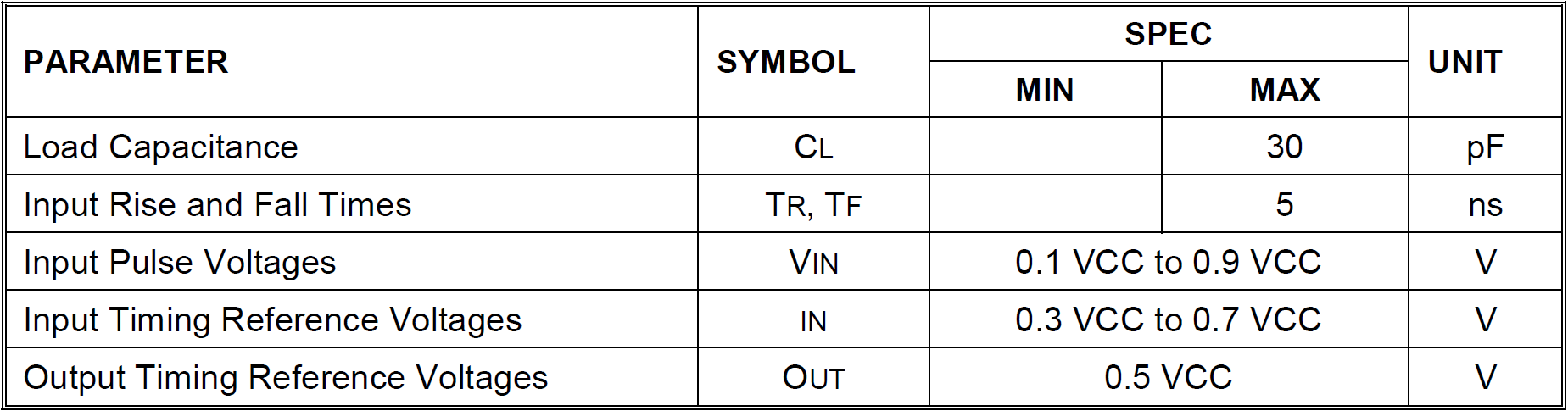
Power-up, Power-Down Requirement

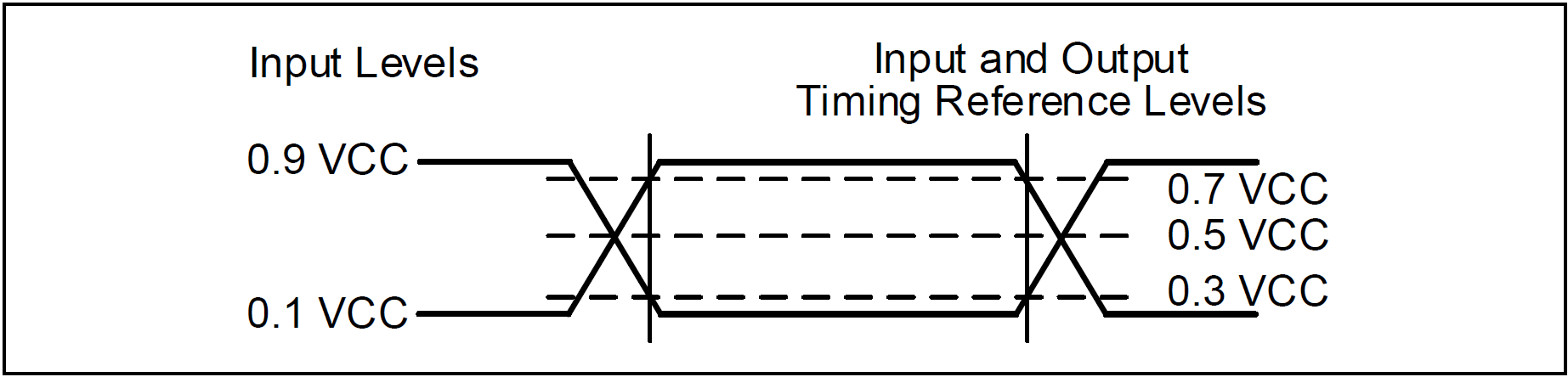
### 9.4 DC Electrical Characteristics



Notes : The typical (TYP) value is tested on sample basis and specified through design and characterization data. TA = 25° C, VCC = 3.0V.

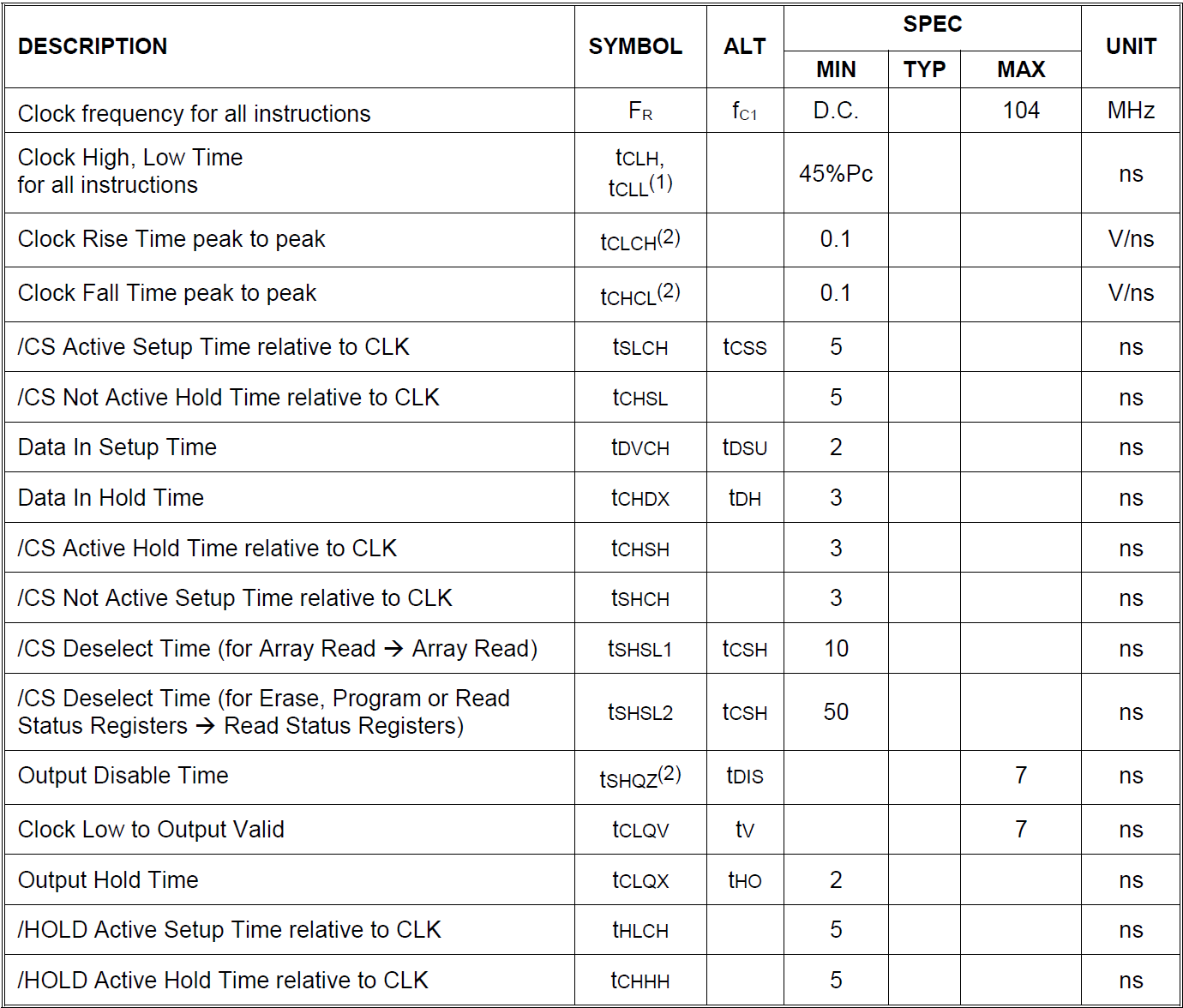
### 9.5 DC Electrical Characteristics



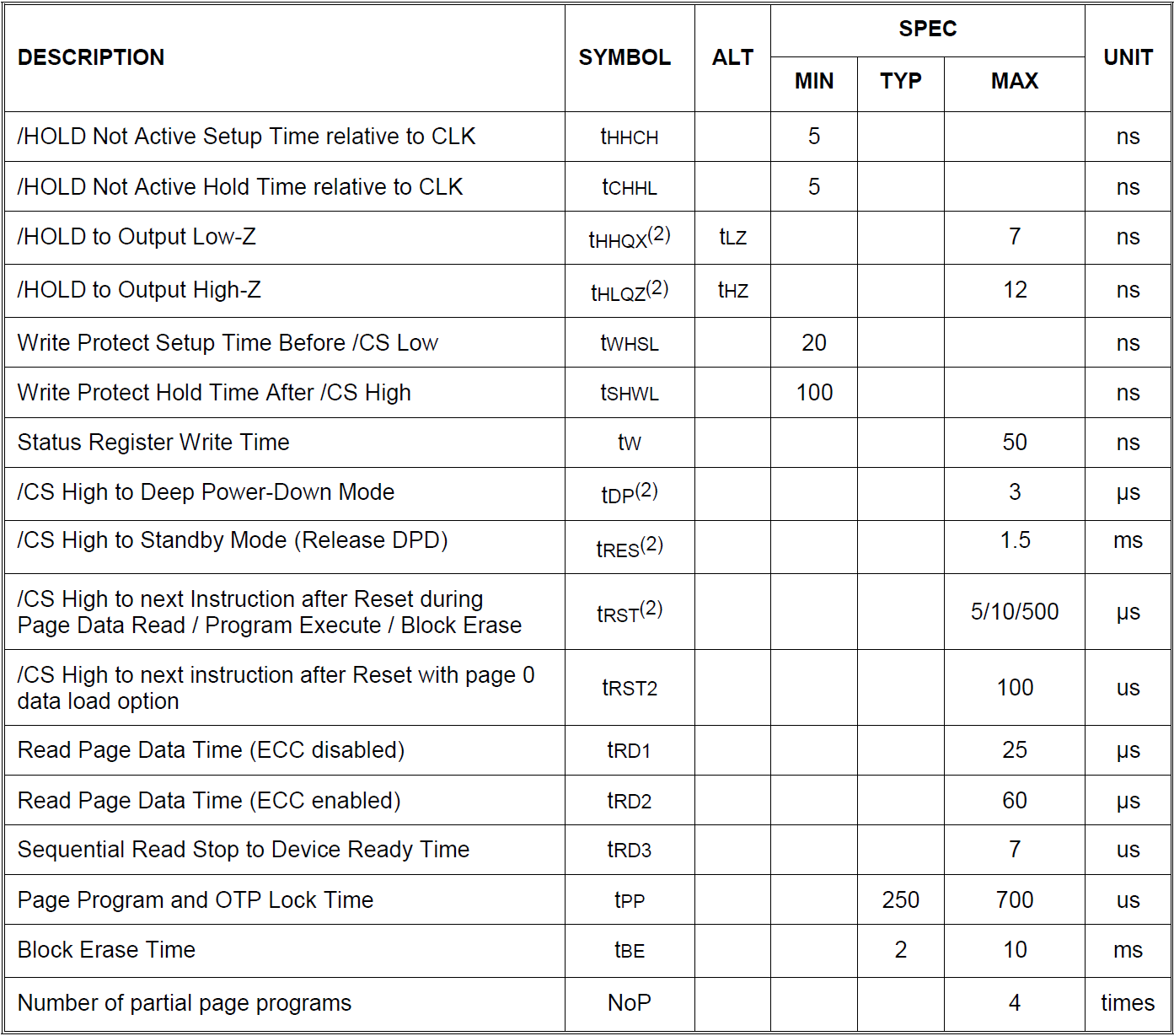


AC Measurement I/O Waveform

### 9.6 AC Electrical Characteristics(3)



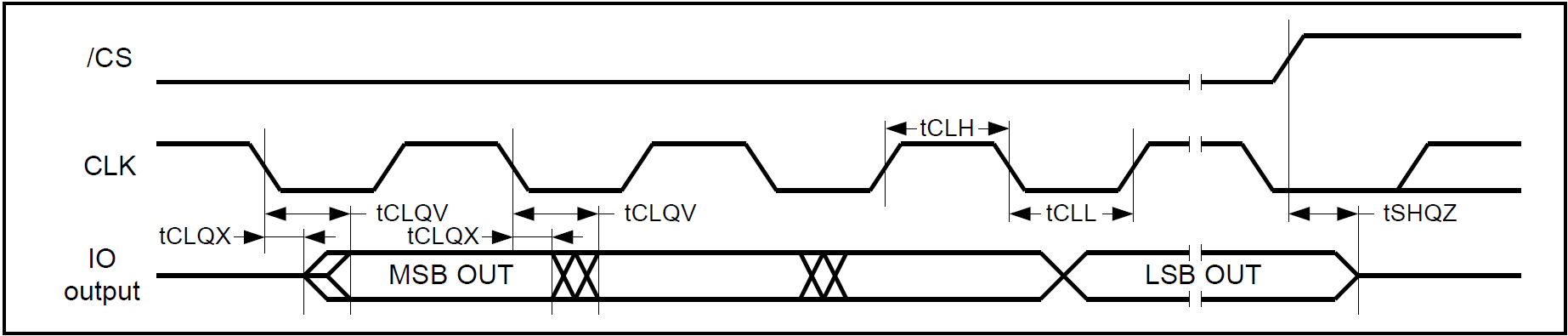
**AC Electrical Characteristics (cont’d)**



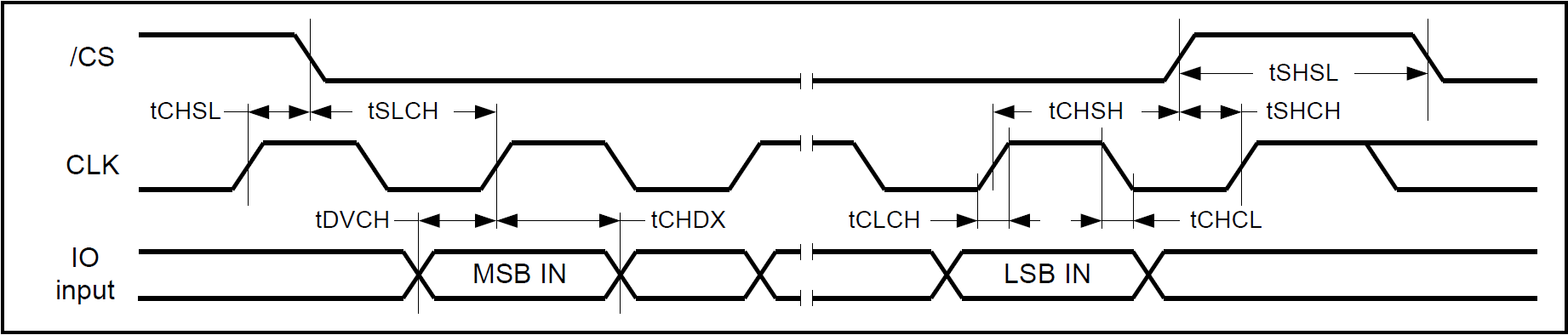
Notes :

1. Clock high + Clock low must be less than or equal to Pc. PC = 1/fc1(max)
2. Value guaranteed by design and/or characterization, not 100% tested in production.
3. The typical (TYP) spec is tested on sample basis and specified through design and characterization data. TA = 25° C, VCC = 3.0V.
4. The product which re-loads page0 data after Reset takes tRST + tRD busy time.
5. AC electrical characteristics is based on default setting of ODS.

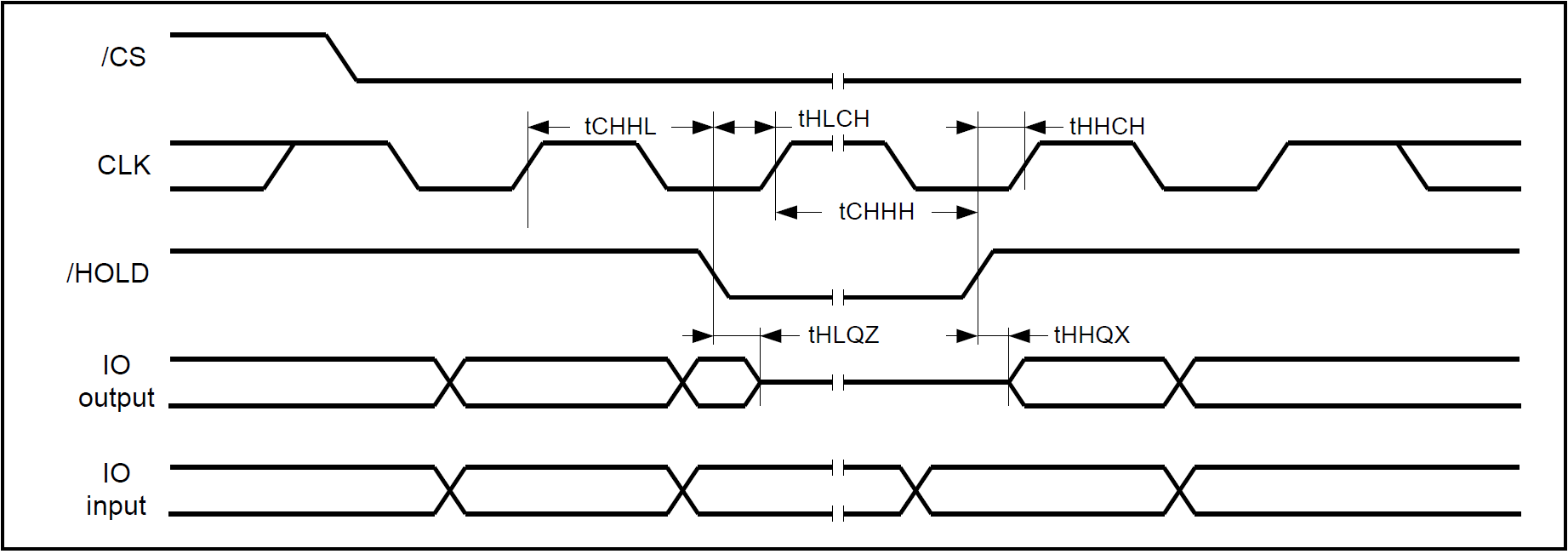
### 9.7 Serial Output Timing



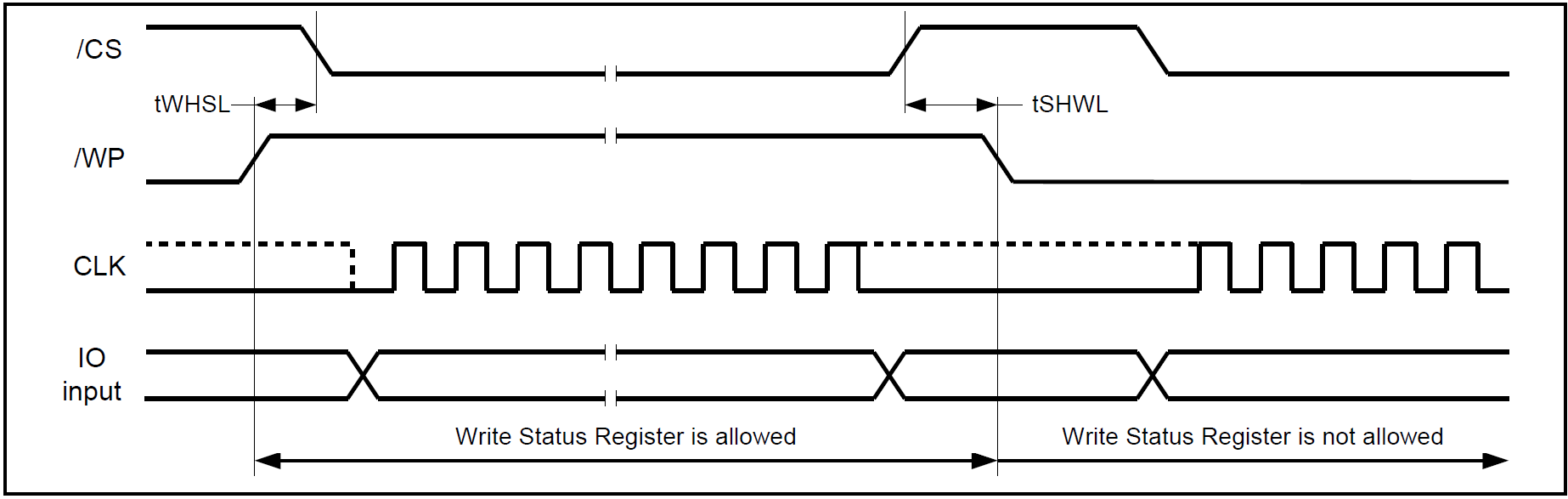
### 9.8 Serial Input Timing



### 9.9 /HOLD Timing



### 9.10 /WP Timing



## 10. INVALID BLOCK MANAGEMENT

### 10.1 Invalid Blocks

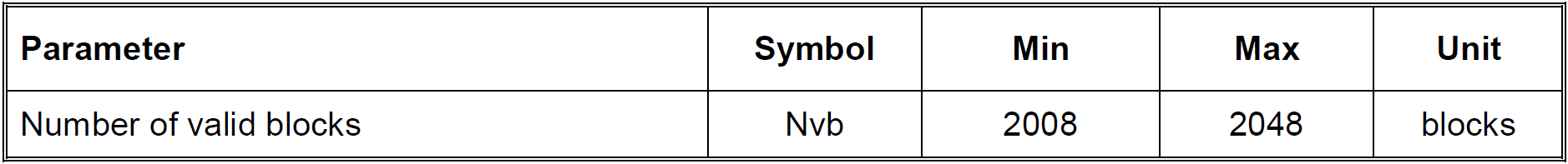
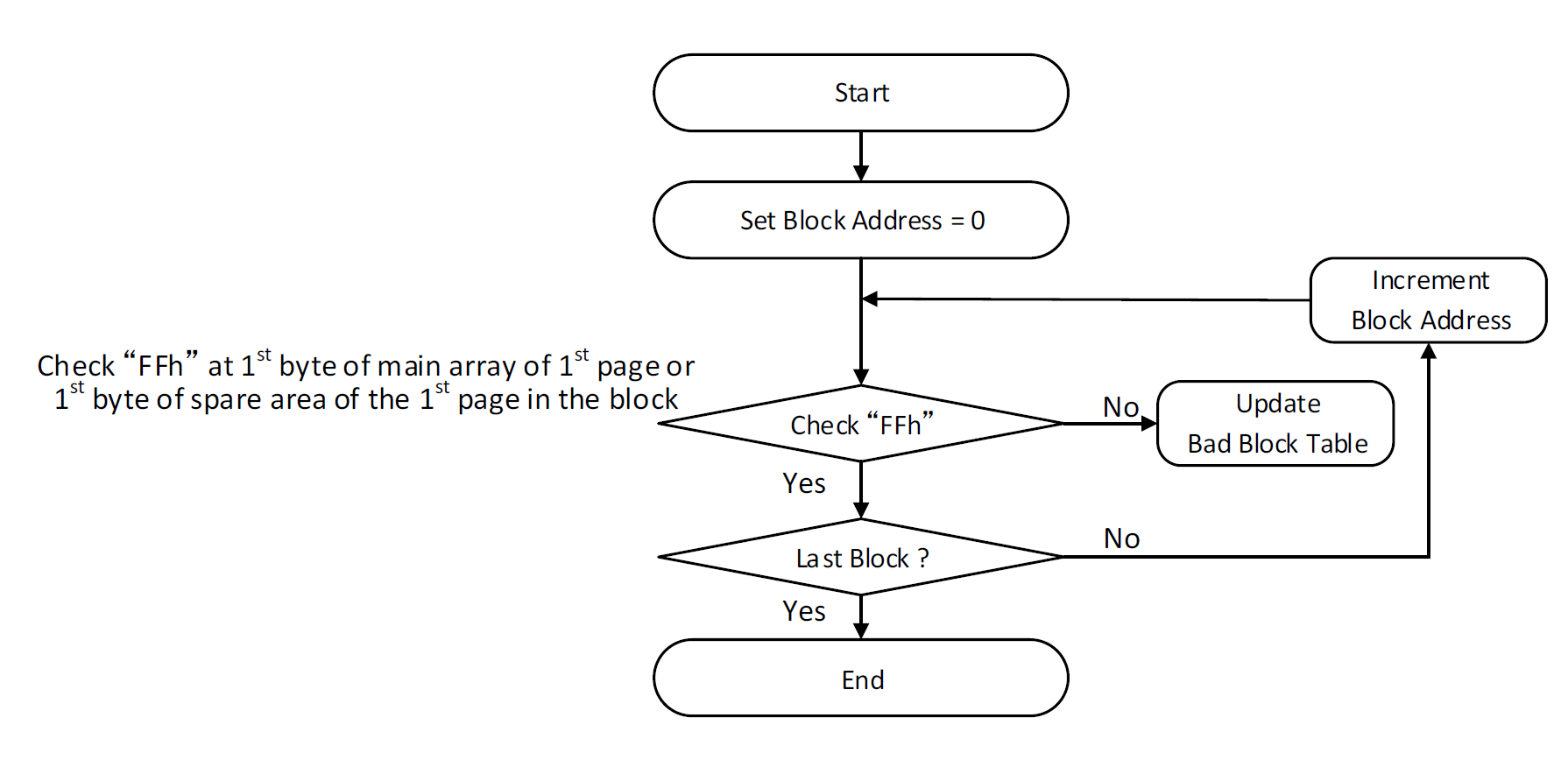


Table 10-1 Valid Block Number

**Factory bad blocks :**

* **The device may ship with some invalid (bad) blocks.**
* **These blocks are permanently marked before leaving the factory.**
* **Runtime bad blocks :**
* **Additional invalid blocks may appear during usage due to wear-out and program/erase cycles.**
* **Definition :**
* **A block is considered invalid if it contains uncorrectable ECC errors.**
* **ECC correction is not enough ---> block must be retired.**
* **Guarantees :**
* **In All 2048 Block have least 2008 (0 ~ 2007) guaranteed valid Blocks.**
* **First 8 blocks ( 0 ~ 7 ) and last 4 blocks ( 2044 ~ 2047 ) are always valid when ECC is enabled.**
* **FW implication :**
* **FW must not assume all blocks are usable.**
* **FW must maintain a Bad Block Table (BBT) and always skip invalid blocks.**

### 10.2 Initial Invalid Blocks

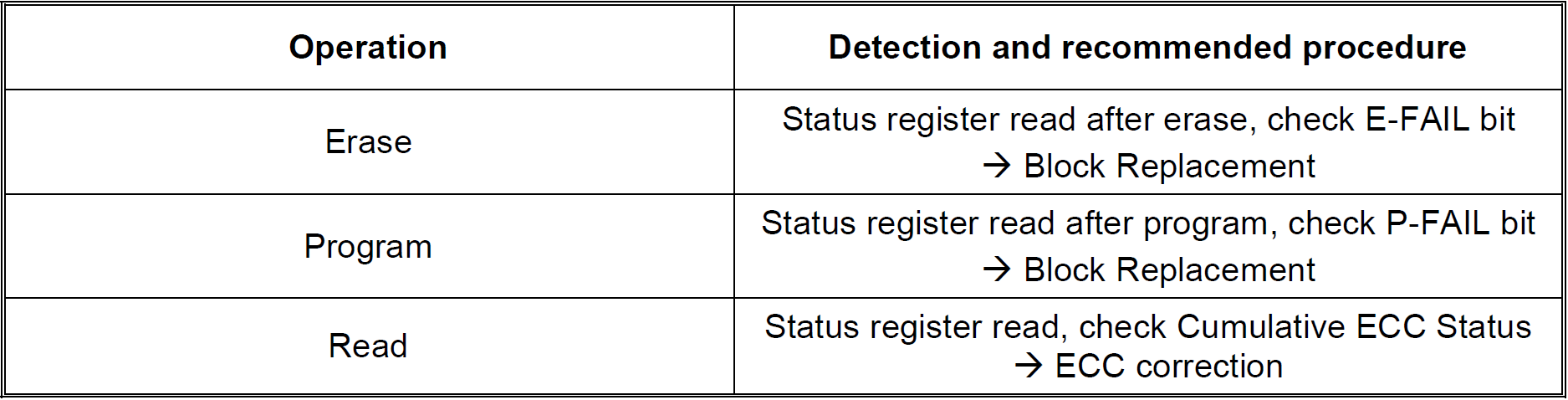


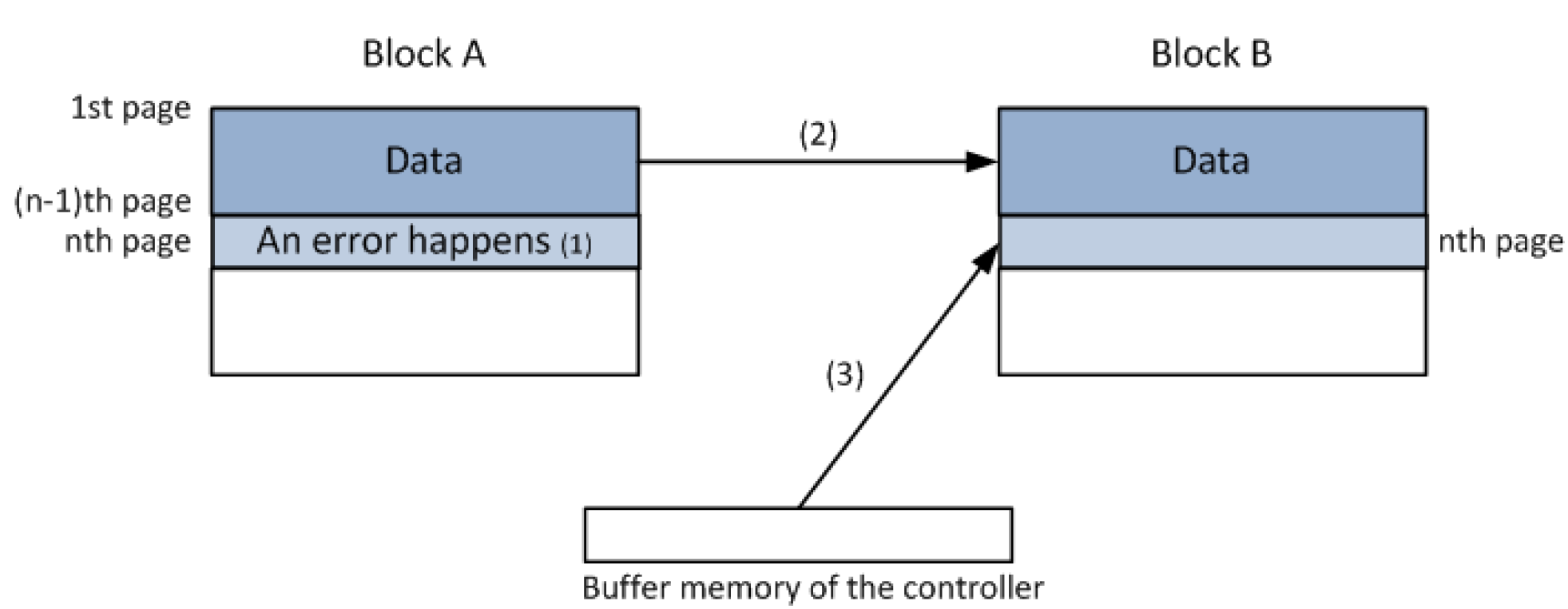
* **Marking Rule for Invalid Blocks (Permanent, Cannot be Erased):**
  + **First Main Array in Page 1 Byte != FFh**
  + **First Spare Area in Page 1 Byte != FFh**
* **FW requirement at startup :**
  + **After Power-up, Must scan all block addresses from 0 ~ 2047.**
  + **Marker != FFh, Add Block Address to Bad Block Table (BBT).**
* **Handling invalid blocks :**
  + **FW must exclude bad blocks from : Erase / Program / Read ( except when verifying marker ).**
* **Updating BBT dynamically:**
  + **If ECC reports an uncorrectable error during runtime, FW should immediately mark the block as invalid and update BBT.**
* **Flow chart implementation :**

1. **Initialize Block Address = 0**
2. **Check marker byte (FFh) in main array or spare area**
3. **If not FFh, Need to update Bad Block Table**
4. **Increment Block Address**
5. **Repeat until last block (2047)**

* **FW strategy:**
  + **Maintain the BBT in RAM for fast access.**
  + **Optionally store BBT in reserved NAND area for persistence.**
  + **Ensure wear-leveling logic is aware of bad blocks.**
  + **FW must handle remapping ---> logical block address (LBA) mapped to physical block, skipping invalid blocks.**

### 10.3 Error in Operation





**Note :**

1. **An error happens in the nth page of block A during program or erase operation.**
2. **Copy the data in block A to the same location of block B which is valid block.**
3. **Copy the nth page data of block A in the buffer memory to the nth page of block B**
4. **Creat and update bad block table for preventing further program or erase to block A**

* **Additional invalid blocks :**
  + **New invalid blocks may appear during lifetime due to program/erase (P/E) cycles.**
  + **FW must dynamically detect them during operation.**
* **Block replacement procedure:**
  + **Detect error in Block A using Status Register.**
  + **Copy valid data from Block A (up to failing page) to Block B (a valid block).**
  + **Copy failing page data from controller buffer to corresponding page in Block B.**
  + **Update Bad Block Table (BBT) to prevent further use of Block A.**
* **FW Implication:**
  + **FW must always check status bits after erase/program/read.**
  + **If failure occurs → perform block migration (copy + remap).**
  + **BBT must be dynamically updated when runtime failures occur.**

### 10.4 Addressing in Program Operation

* **Sequential page programming rule :** 
  + **Pages must be programmed sequentially within a Block.**
  + **Programming must start from the LSB page (page 0) and continue upwards.**
  + **Random page programming is prohibited.(Never)**
* **FW Implication :** 
  + **FW must enforce sequential programming.**
  + **Skipping pages or out-of-order programming can corrupt data.**
  + **Garbage collection (GC Policy) and wear leveling (WEL) must respect sequential programming rule.**